

series48

- 10, 25 & 50W models
- Low cost design
- Rugged & reliable
- Proven operating lifetimes > 45,000 hours



Synrad's **series48** - setting the industry standard

Synrad's original "all-metal" tube technology opened the door for sealed CO₂ lasers in many industrial applications. After 15 years, it remains the industry standard for performance, reliability, long lifetime, and low cost.

The laser tube is a rigid box structure design that provides an ultra-stable platform for the laser resonator. Synrad's 48-series tubes are mounted into an H-shaped heat sink extrusion that also serves as the chassis for the drive electronics. This guarantees that the laser will withstand the harsh requirements of modern industrial environments.

The all-metal sealed tube design and proprietary manufacturing process ensure high gas purity, essential for long operating lifetimes. With over 100,000 Synrad lasers operating worldwide, Synrad is the only CO₂ laser manufacturer that can boast of proven operating lifetimes in excess of 45,000 hours (at which time, a simple and inexpensive gas refill returns the laser to full operation).

Synrad lasers can be operated from 0 - 100% duty cycle, with laser power adjusted using a pulse width modulation (PWM) control. The laser is controlled directly with a TTL signal (rear-mounted BNC connector). Alternatively, a DB-9 connector provides remote access to all laser functions, including laser control, fault conditions monitoring, remote interlock, and failure shutdown options.

All 48-series lasers operate from standard 30VDC power supplies, and can be either fan-cooled (48-1 and 48-2 models) or water-cooled.

With output powers of 10, 25, or 50W, and available in either OEM or keyswitch configurations, Synrad's 48-series lasers are ideal for incorporation into marking, engraving, and small cutting systems, where performance, low price and high reliability are essential.

Synrad, Inc. 4600 Campus Place Mukilteo, Washington 98275 USA
tel 1.425.349.3500 • fax 1.425.349.3667 • email synrad@synrad.com • www.synrad.com

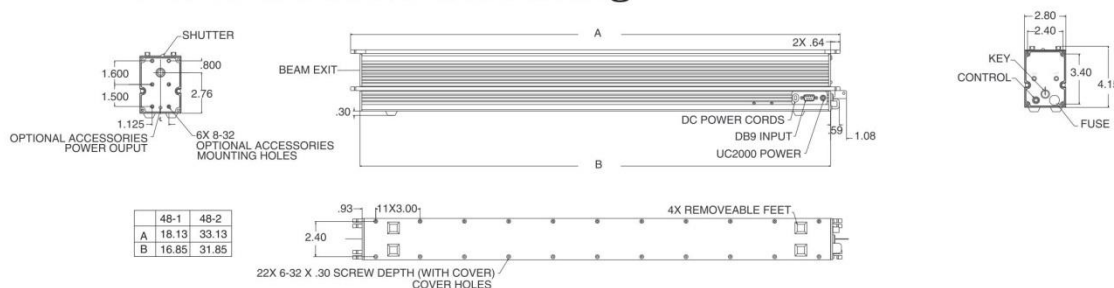
SYNRAD[®]
An Excel Technology Company

• Specifications

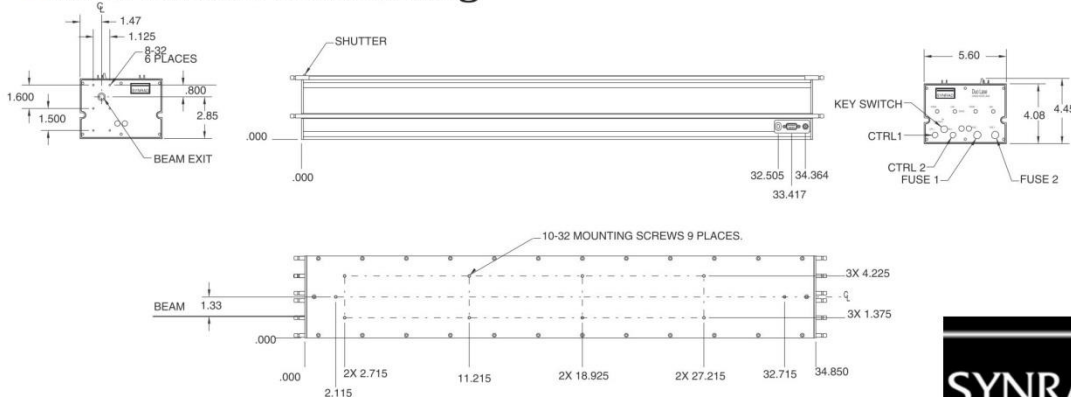
Model	48-1(S)	48-1(S)W	48-2(S)	48-2(S)W	48-5(S)W
Output Power	10W		25W		50W
Mode Quality	TEM ₀₀ , 95% Purity M ² <1.2		TEM ₀₀ , 95% Purity M ² <1.2		TEM ₀₀ , 95% Purity M ² <1.2
Ellipticity	<1.2		<1.2		<1.2
Rise Time	<150µsec		<150µsec		<150µsec
Beam Diameter	3.5mm		3.5mm		3.5mm
Beam Divergence (full angle)	4mR		4mR		4mR
Wavelength	10.57-10.63µm*		10.57-10.63µm*		10.57-10.63µm*
Power Stability, from cold start (guaranteed)	±10		±5%		±5%
Polarization	Linear (Vertical)		Linear (Vertical)		Random
Cooling	Air	Water	Air	Water	Water
Heat Load (max)	300W		500W		800W
Flow Rate, Air	250 CFM x 2	N/A	250 CFM x 4	N/A	N/A
Flow Rate, Water (18-22°C)	N/A	0.5 GPM	N/A	0.8 GPM	1.5 GPM
Input Voltage / Current	30 VDC / 7A		30 VDC / 14A		30 VDC / 28A
Dimensions (in)	16.9 x 2.8 x 4.2		31.9 x 2.8 x 4.2		34.9 x 5.3 x 4.5
(mm)	429 x 71 x 107		810 x 71 x 107		886 x 135 x 114
Weight	9 lbs / 4.1 kg		18 lbs / 8.2 kg		44 lbs / 20 kg

(S) in the model number designates OEM configuration (does not include keyswitch or shutter switch).
 Beam specifications measured at 1/e².
 *Typical. Actual wavelength range may vary from 10.2-10.8µm
 Specifications subject to change without notice.

• 48-1/48-2 Outline & Mounting



• 48-5 Outline & Mounting



Dimensions are in inches
 Additional drawings and specifications at www.synrad.com/48series

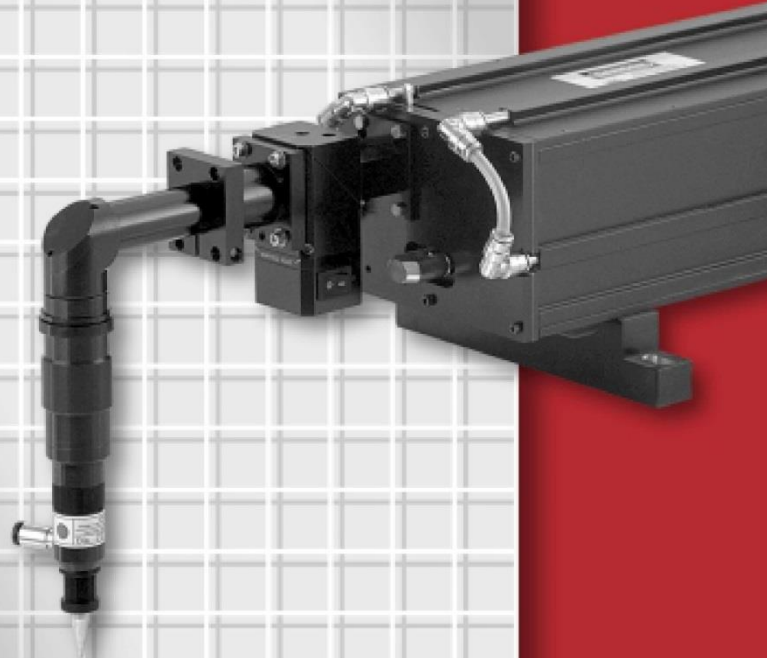


1.25" Series

SOLUTIONS FOR

INDUSTRIAL

BEAM DELIVERY



1.25" SERIES



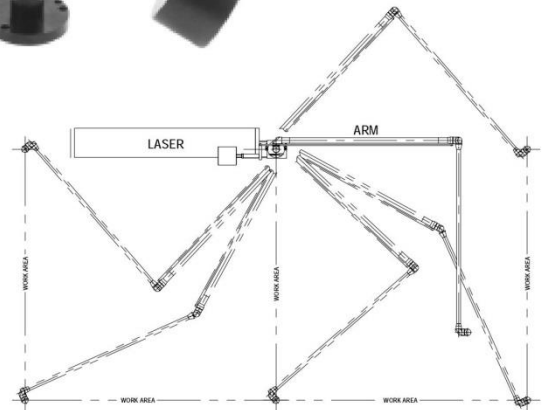
1.25 SERIES INDUSTRIAL ARTICULATED ARM

1.25 Series Industrial Articulated Arm (ATA) is a seven mirror arm consisting of 8 precision ball bearing rotary joints and 7 prealigned right angle mirror knuckles. The length of the arm is designed for reasonable access around the ends of robots or over x-y plotters.

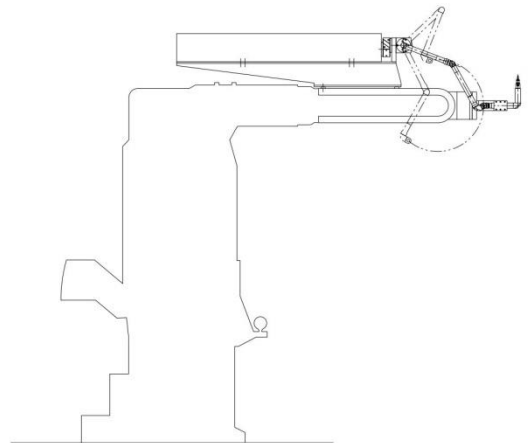
- Tubes are hollow, made of anodized aluminum
- The clear aperture of the arm is available in 15mm or 19mm
- The natural beam divergence of popular small sealed lasers allows use of the 15mm arm in almost all cases if the arm is mounted close to the laser. The Synrad 200 Series requires the 19mm arm
- Arm launching accessories consisting of an adjustable beam bender and mount plate are also available

Model PLATA0119 Articulated Arm

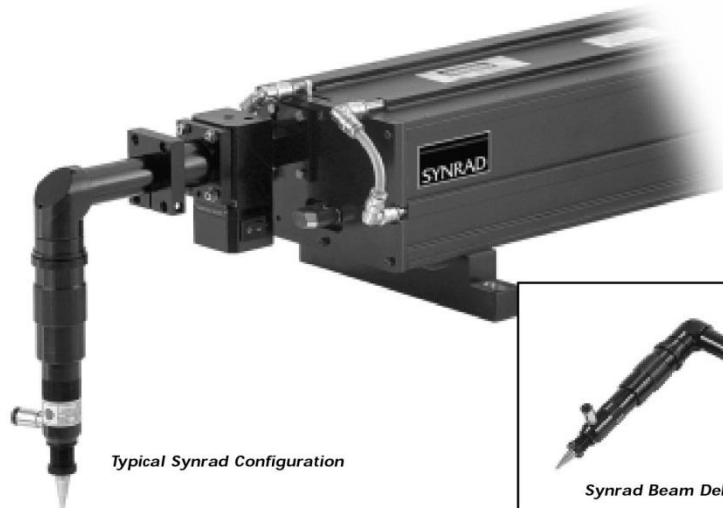
Model PLAAM0001 Articulated Arm Mount



Arm Used To Connect A Laser To A Flat Bed Cutter



Arm Used To Connect A Laser To A Robot Arm



Typical Synrad Configuration



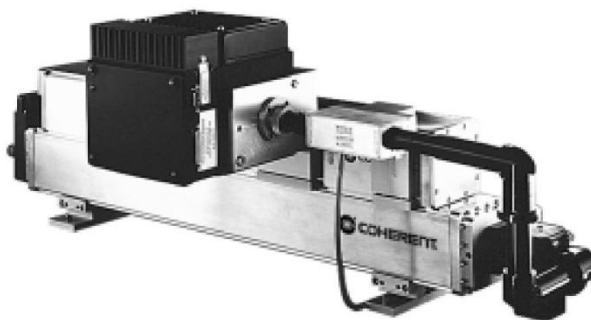
Synrad Beam Delivery System

1.25 SERIES COMPACT BEAM DELIVERY SYSTEMS

1.25 Series Compact Beam Delivery Systems offer a turnkey beam delivery solution for small output Synrad and Diamond sealed CO₂ lasers. Specifically designed for smaller lasers, compact systems offer a significant price advantage over traditional full-size components. Designed and manufactured to the same exacting standards as our full-size components, compact sys-

tems deliver a level of performance, reliability and ease of installation that's unmatched in their class.

Let our team of laser system engineers analyze your compact beam delivery needs. We'll provide a design for a performance guaranteed beam delivery system for your laser machine or new industrial laser-based product.



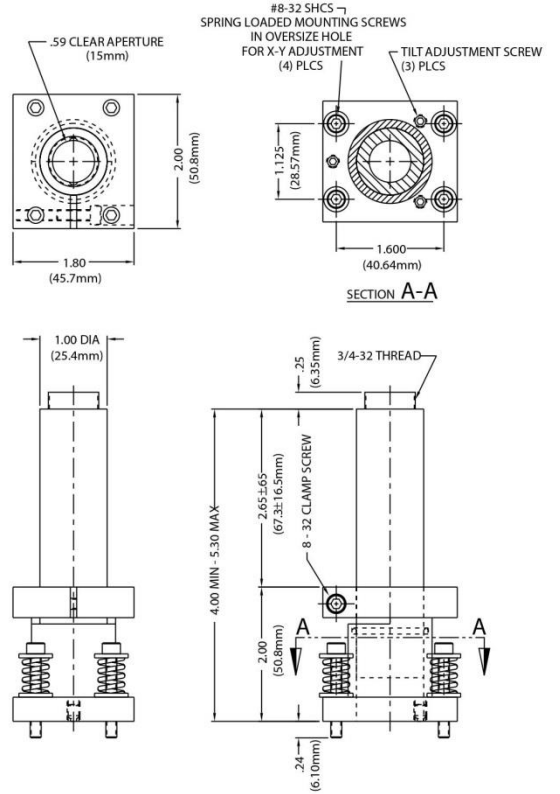
Typical Coherent Configuration



Coherent Beam Delivery Systems



1.25" SERIES



Model PLLMS0008

1.25 SERIES SYNRAD MOUNTING SYSTEMS

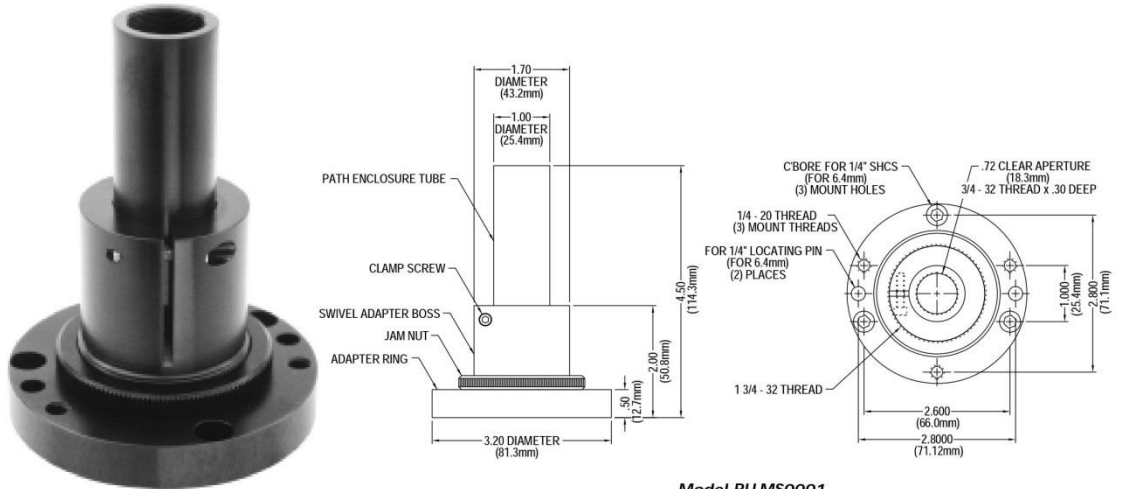
1.25 Series Synrad Mounting Systems are available in two different styles, the PLLMS0004 and PLLMS0008. Both models incorporate an adapter swivel clamp through which a path enclosure tube can be slipped and clamped. The path enclosure tube has 1.30" of travel within the clamp. The clamping feature allows the tube to slide back and forth and can be used as a coarse focus adjust or adjustable length mounting tube. The rectangular clamp mounts with 8 – 32 screws. An "O" ring inside the adapter clamp keeps the beam path sealed. The PLLMS0008 has added adjustable tilt and translation feature.

Model PLLMS0004 Fixed Mounting System (Synrad Lasers)

Model PLLMS0008 Adjustable Mounting System (Synrad Lasers)



Model PLLMS0004



Model PLLMS0001

**1.25 SERIES
COHERENT MOUNTING SYSTEMS**

1.25 Series Coherent Mounting Systems are available in two different styles.

The PLLMS0001 incorporates an adapter boss swivel clamp through which a path enclosure tube can be slipped and clamped. The outside has a 1.75 – 20 thread which will adapt to the flange of many standard lasers or to the adapter ring provided. A 1.75 – 20 jam nut allows the boss to be oriented in the desired direction in the ring or flange. The clamping feature allows the path enclosure tube to slide back and forth (2" travel) and can be used as a coarse focus adjust or adjustable length mounting tube. An "O" ring inside the adapter boss keeps the beam path sealed.

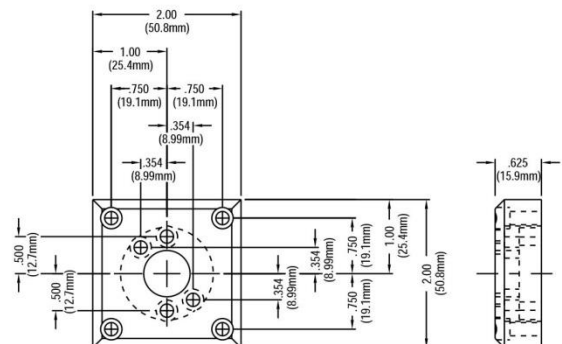
The second style, the PLADP0021, is specifically designed for Mini-Diamond lasers and allows the mounting of beam benders at 45 and 90 degree angles.

Model PLLMS0001 Mounting System (Coherent Diamond Lasers)

Model PLADP0021 Mounting System (Coherent Mini-Diamond Lasers)



Model PLADP0021
(For Mini-Diamond Series)



1.25" SERIES



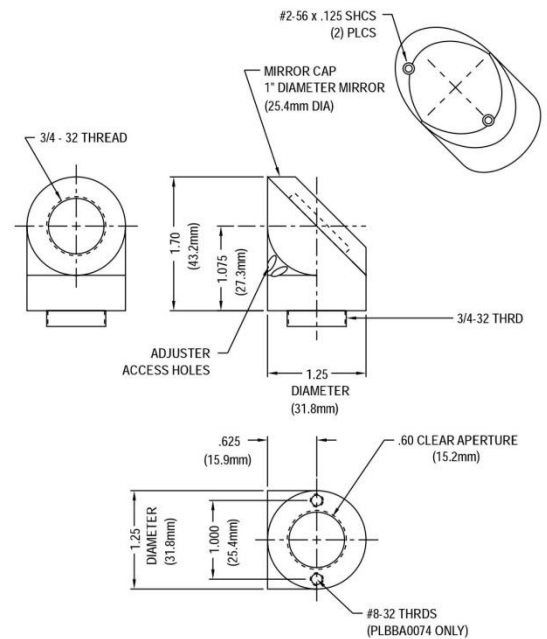
1.25 SERIES BEAM BENDERS

1.25 Series Beam Benders (BBA) are fully adjustable, highly stable 90 degree Beam Benders with a clear aperture of 15mm.

- The adjustment system consists of a fixed plane pivot and two 80 pitch tilt screws. Access to these screws is through two small holes in the body of the bender block.
- A 1.5mm hex wrench can be used to make fine adjustments in the mirror angle of ± 2 degrees
- The face referenced 1.0" diameter, 3mm thick mirror can be removed and replaced with an accuracy of better than .00001 radian
- The mirror is contained in a cap which can be easily removed for cleaning
- Both the input and output faces have a .75 – 32 female thread for mounting the bender or attaching it to other 1.25 Series components
- A male threaded connector is also included
- The PLBBA0074 has the added benefit of two 8 – 32 tapped holes for screw mounting to adapter plates such as the PLADP0021

Model PLBBA0041 Beam Bender

Model PLBBA0074 Beam Bender Screw Mount



Models PLBBA0041/PLBBA0074



PLDPA0003



PLDPA0009

1.25 SERIES DIODE POINTER ASSEMBLIES

1.25 Series Diode Pointers (DPA) contain a ZnSe beam combiner which transmits the CO₂ beam and reflects a red laser diode. The diode can be centered to adjust the near field coincidence and the combiner angle is adjustable for far field coincidence of the diode beam to the CO₂ beam. This eliminates the need for cumbersome IR viewing instruments.

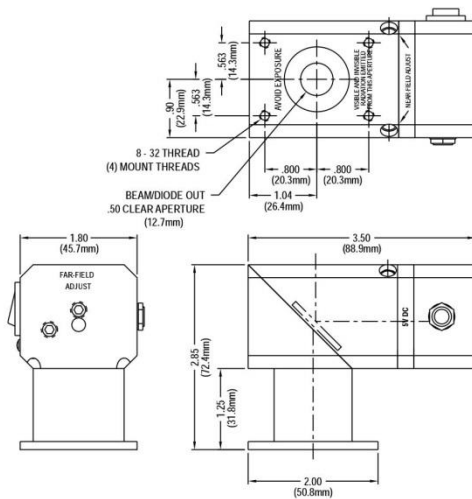
Diode Pointers also provide the important safety benefit of enabling the laser user to clearly monitor the loca-

tion of the CO₂ beam without being exposed to the high-power beam. Diode Pointers can be mounted to many of today's popular sealed CO₂ lasers.

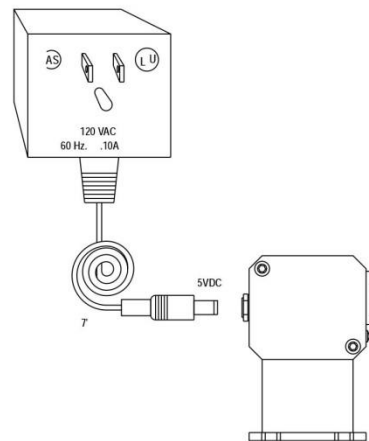
A variety of diode pointers are available with different mounting configurations. Contact a Laser Mech laser system engineer for available models.

Model PLDPA0003 Diode Pointer (Synrad Lasers)

Model PLDPA0009 Diode Pointer With Power Monitor Port (Coherent Mini-Diamond Lasers)



Model PLDPA0003



1.25" SERIES

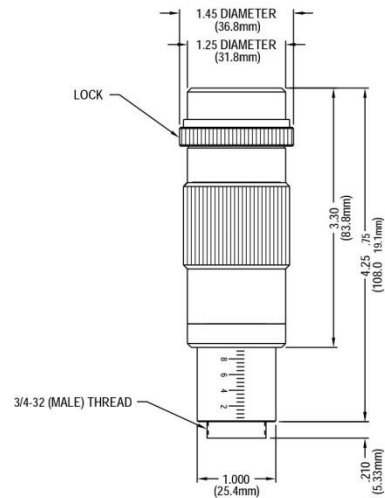
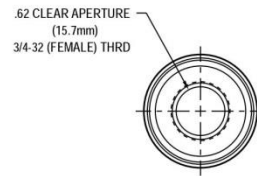


1.25 SERIES VERNIER ADJUST SPACER

1.25 Series Vernier Adjust Spacer (VAS) provides a fine focus adjustment for the lens mounted in the gas jet manifold. Turning the knurled ring advances or retracts the non rotating focus tube .05" per turn. A scale on the focus tube provides a reference of exact axial position. All threads and sliding joints are sealed to prevent dirt from entering the mechanism or beam path.

- The unit is of black anodized aluminum construction
- The full travel of the slide tube is 1.5"
- A lock nut holds the spacer in position
- Each end of the VAS has a .75 – 32 female thread
- A male threaded adapter for attaching it to other 1.25 Series components is included

Model PLVAS0022 Vernier Adjust Spacer 1.25" Series



Model PLVAS0022

1.25 SERIES GAS JET MANIFOLDS

1.25 Series Gas Jet Manifolds (GJM) include:

- A lens holder housing with a .75" diameter lens of the focal length selected premounted in the housing body
- An input adapter with a female .75 – 32 thread to mate to other 1.25 Series components
- A gas inlet fitting for .25" OD tubing to admit gas below the lens
- An adjustable length extension tube with jam nut and tip retainer nut set for the length of lens selected
- A removable gas jet tip

The longer the lens, the more standoff distance is available. Consult individual manifold drawings for exact details on this parameter.

For applications where a gas jet is not required, the tip can simply be removed or a simple lens holder is available to replace the entire gas jet manifold.

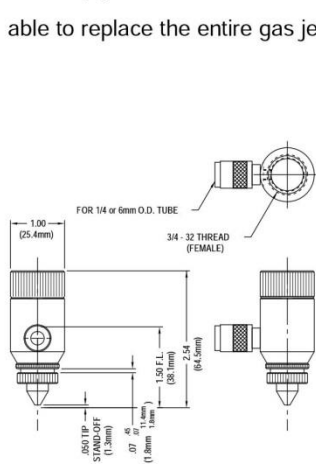


The Gas Jet Tip (GJT) standard orifice is .063", the material is copper and the tip is tapered to .10" diameter. For optimum cutting, blunt-shaped tips and other orifice diameters and materials are available. They can be ordered separately.

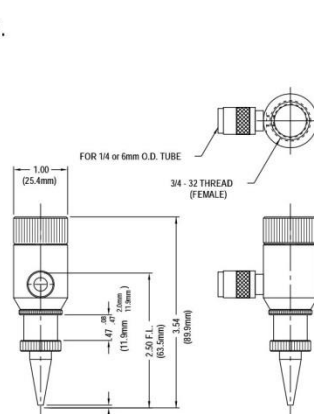
Model PLGJM0102 Gas Jet Manifold 1.5" Focal Length

Model PLGJM0105 Gas Jet Manifold 2.5" Focal Length

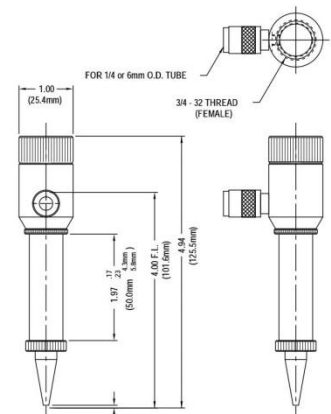
Model PLGJM0107 Gas Jet Manifold 4.0" Focal Length



Model PLGJM0102



Model PLGJM0105



Model PLGJM0107

1.25" SERIES



1.25 SERIES PATH ENCLOSURE TUBES

1.25 Series Path Enclosure Tubes (PET) are anodized aluminum tubes which can be used to support other 1.25 Series components and enclose the beam path. The inside diameter is 18mm and each end has a female .75 – 32 thread with a male threaded connector in one end. This allows several tubes to be connected in series or attached to other components.

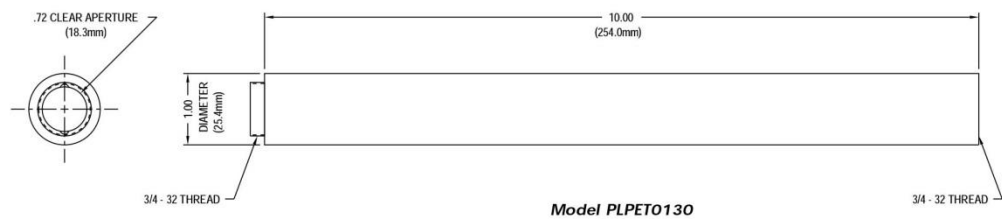
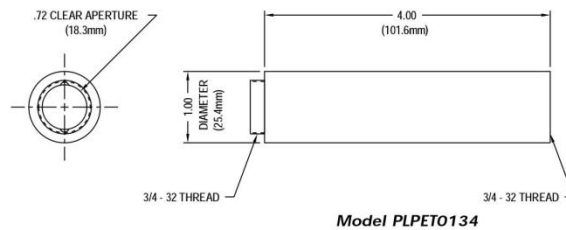
The 1" outside diameter tubes are machined so they

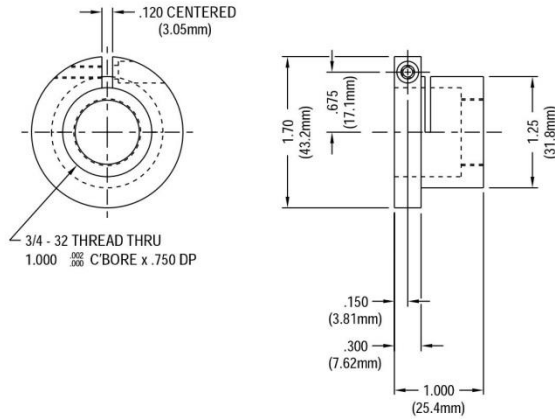
can be clamped in mounting swivel bosses which are part of the system.

The tubes shown are only an example of the variety of PETs available. Consult a Laser Mech laser system engineer for a PET that's right for your system.

Model PLPET0134 Path Enclosure Tube 4"

Model PLPET0130 Path Enclosure Tube 10"





Model PLADC0001



1.25 SERIES ADAPTER CLAMP

1.25 Series Adapter Clamp (ADC) accepts path enclosure tubes which can then be slipped and clamped. The clamping feature allows the tube to slide back and forth and rotate to allow length and angle adjustments between components – such as two beam benders. This allows beam benders to be rotated to any angle desired.

Model PLADC0001 Adapter Clamp 1.25" Series



1.25 SERIES COLLIMATORS

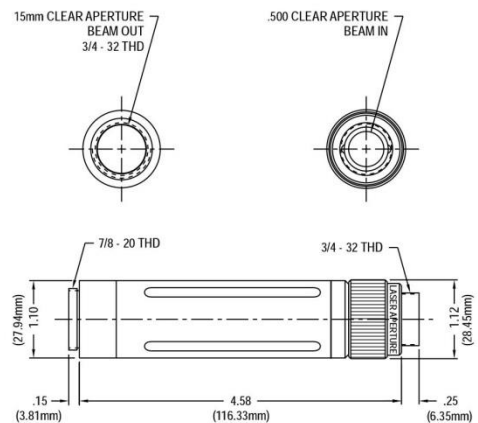
1.25 Series Collimators (COL) are used to expand the beam 1.6X, 2.0X or 2.5X, with an output aperture of 15mm, consistent with the rest of the series. Collimators are used selectively when the delivery system is very close to the laser and the "F" number of the system must be lowered to achieve the smallest spot size possible.

When ordering, make sure the intended exiting beam (input beam dia. x collimator ratio) will be no larger than 14mm, so it can safely clear the 15mm exit aperture.

Model PLCOL0052 1.6X Collimator

Model PLCOL0053 2.0X Collimator

Model PLCOL0054 2.5X Collimator



Model PLCOL0052

1.25" SERIES



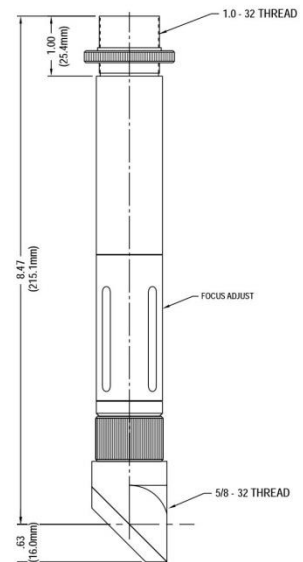
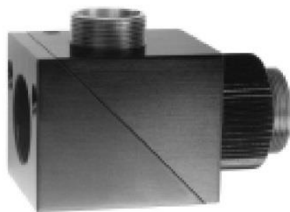
1.25 SERIES BEAM COMBINER BLOCK & OVER THE LENS VIEWER

1.25 Series Beam Combiner Block (BCB) allows the addition of a side viewing port for attachment of a TV camera viewing system or a diode laser coaxial pointing beam. The unit consists of a splitter block and cap for a 1.0" diameter ZnSe beam combiner. The block provides for the proper offset due to the refraction of the laser beam combiner optic. The input port has a .75 - 32 female thread and the exit port has a .795 - 32 female thread with an adapter to .75 - 32 male thread. The viewing port has a .625 - 32 female thread with male adapter included.

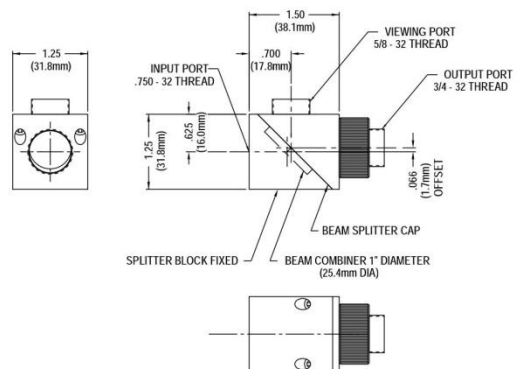
1.25 Series Over The Lens Viewer (OLV) is a lens system which can be used with a 2.5" to 5.0" focal length ZnSe lens. It can also be used with any "C" mount CCD camera and monitor for viewing through the laser focusing lens at the work piece. It mounts to the viewing port of the BCB with a .625 - 32 male adapter.

Model PLBCB0001 Beam Combiner Block

Model PLOLV0024 Over The Lens Viewer



Model PLOLV0024

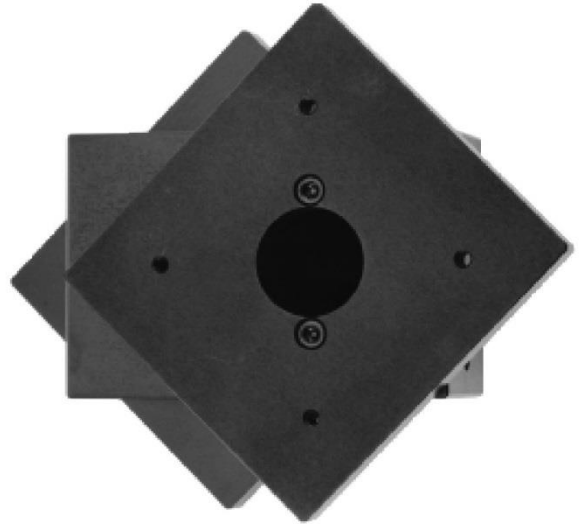


Model PLBCB0001

**1.25 SERIES
CUT QUALITY ENHANCERS**

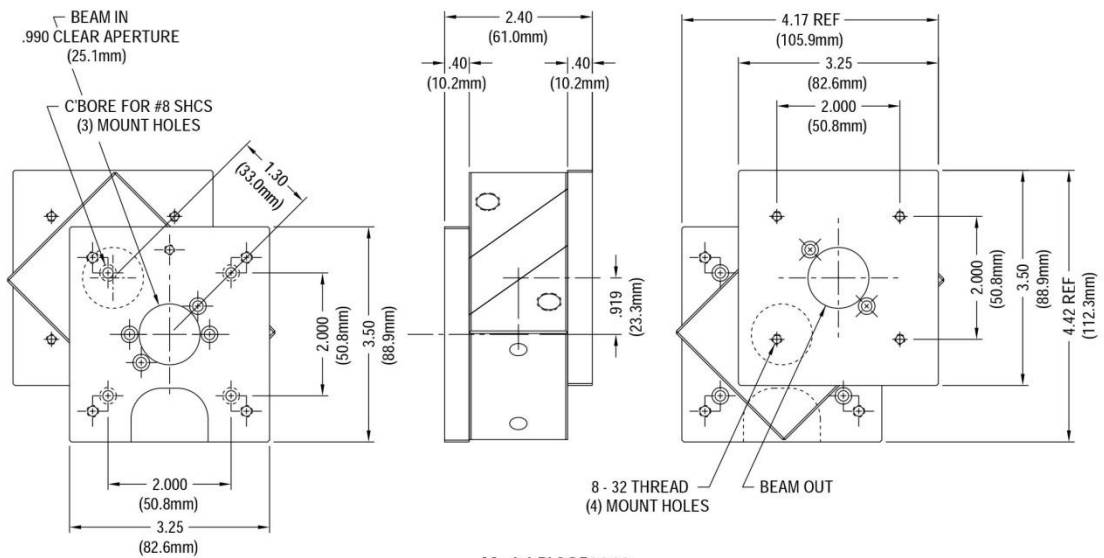
1.25 Series Cut Quality Enhancers (CQE) are used with linearly polarized lasers to produce a circularly polarized beam. When the laser is used for multidirectional cutting processes, a uniform kerf will be obtained. The CQE unit mounts directly to the face of Diamond lasers and provides a surface to remount their standard adapter plate.

The laser mount plate of the unit has three 10 – 32 brass-tipped set screws which provide the reference plane against the front of the laser and are used to adjust the angle of the exiting beam to its original direction. The mirrors contained in the CQE are one 90 degree reflective phase retarder and one zero phase



shift mirror. Both are 1.5" diameter, 4mm thick silicon substrate mirrors.

Model PLCQE0048 25mm Cut Quality Enhancer



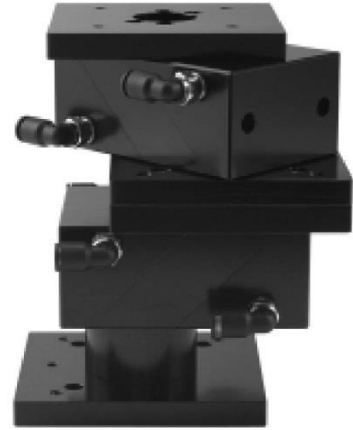
Model PLCQE0048

1.25" SERIES

1.25 SERIES FEEDBACK ISOLATOR

1.25 Series Feedback Isolator (FBI) rejects back reflections off the workpiece, which can upset the laser's intra-cavity conditions, destroying mode quality and power stability. Many process quality problems are traced to this phenomenon.

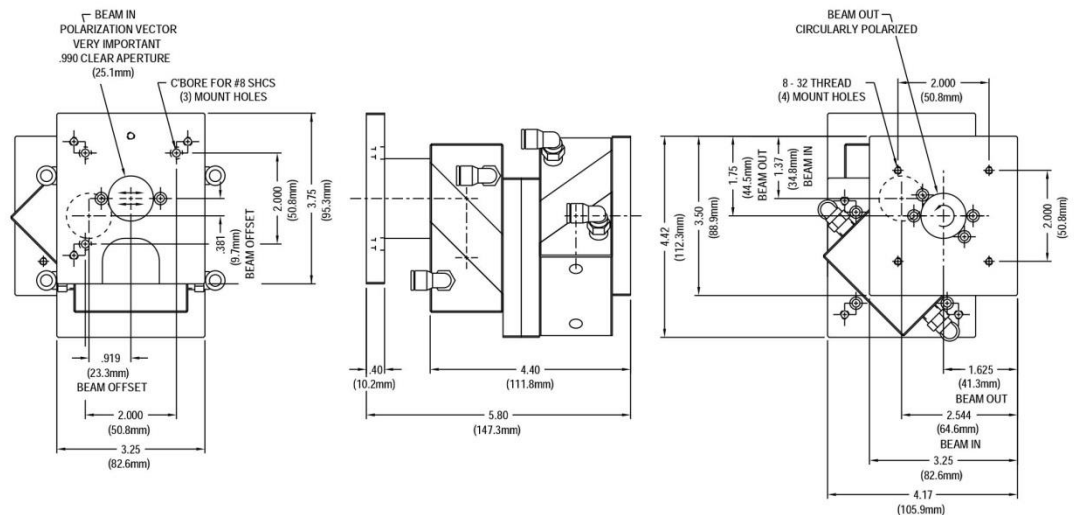
The Feedback Isolator is designed to work with Diamond lasers and it is composed of one feedback isolator optic, two zero-phase shift mirrors and one-phase shifting optic. The Feedback Isolator acts as a polarization analyzer and is located directly in front of the laser output aperture. The S-polarized laser beam from the laser is 98.5 percent reflected by the FBI optic. The beam then enters the second stage of the FBI where it becomes circularly polarized and exits the assembly. Reflections perpendicular to the workpiece will re-enter the circular polarizer where once again they become linearly polarized, except this time the polarization vector is rotated 90 degrees making the



beam P-polarized. The Feedback Isolator absorbs the P-polarized beam thereby preventing it from re-entering the laser cavity.

Water cooling fittings are provided for use with lasers above 200 watts. The aperture of this device even allows it to be used with larger diameter lasers such as the Coherent K-500.

Model PLFBI006 Feedback Isolator 1.25" Series



Model PLFBI006

**1.25 SERIES
TARGETS & ACCESSORIES**

1.25 Series Alignment Tool Kit contains the following:

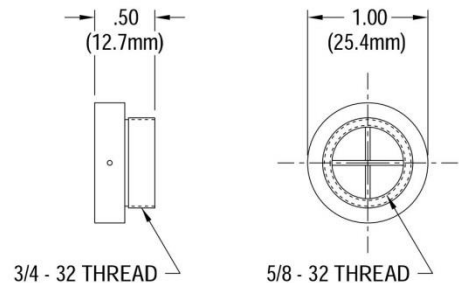
- Beam Bender Target (BBT) has a clear aperture of 9mm and crossed 1mm wires. It inserts into the .625 aperture of the male threaded connector, used to couple the beam bender to the other components. Removing the mirror cap from the bender allows access to the beam from either direction
- Manifold Alignment Target (MAT) has an external .75 – 32 thread for connecting it into any aperture, such as the end of a VAS or PET, and an internal .625 – 32 thread for attaching in place of the gas jet tip retainer nut. The target has crossed 1mm diameter wires and a clear aperture of 15mm
- Wrenches consist of a 1.5mm hex wrench for mirror angle adjusting and a 3/64 hex wrench for removing the mirror cap
- Thermal Paper Cards are used to detect the low power CO₂ beam via a change in color from buff to black

1.25 Series Arm Adapter is used for mounting components to articulated arms.

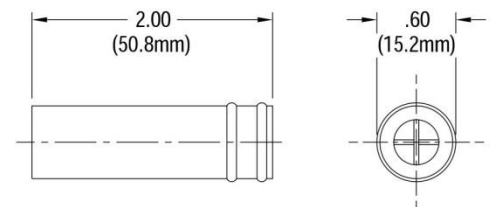
Model PLBBT0008 Beam Bender Target

Model PLMAT0008 Manifold Alignment Target

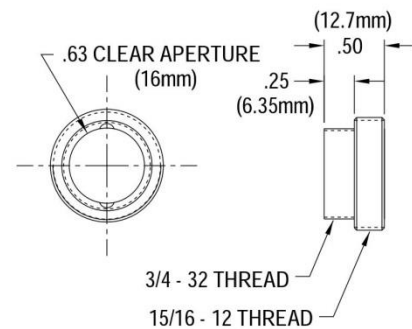
Model PLADB0004 Arm Adapter



Model PLMAT0008



Model PLBBT0008



Model PLADB0004

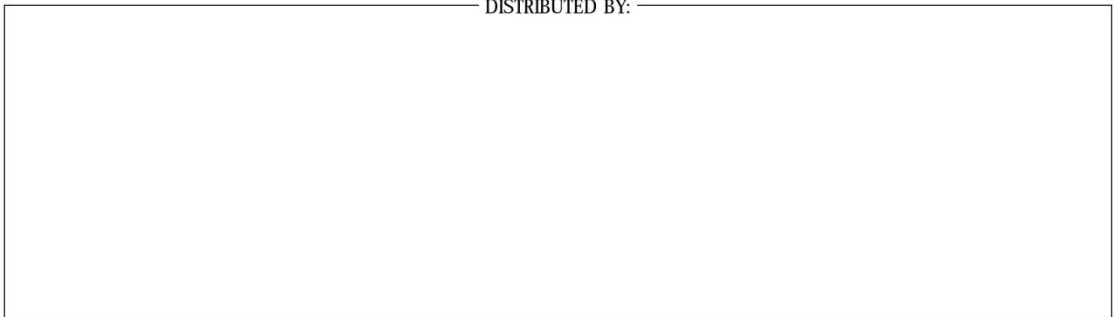


Laser Mechanisms, Inc. specializes in Laser Beam Delivery Components to make the most of your laser.

ALL WAVE LENGTHS • BEAM DIAMETERS TO SIX INCHES • CW POWER TO 25KW

Let our staff of laser system engineers, with a total of over 200 years collective experience, supplement your in-house people. We will provide a design for a performance guaranteed beam delivery system for your special laser machine or new industrial laser based product. We are price effective and qualified to handle everything from the most basic to very sophisticated active sensing beam delivery systems.

— DISTRIBUTED BY: —



24730 Crestview Court • Farmington Hills, MI 48335 USA • Phone: (248) 474-9480 • Fax: (248) 474-9277
In Europe: Phone: +32(0)92 18 70 70 • Fax: +32(0)92 18 70 79
Web: www.lasermech.com • E-Mail: info@lasermech.com

Corporate names and trademarks stated herein are the property of their respective companies.

Precision rolled ball screws





Contents

A Recommendations for selection

SKF – the knowledge engineering company	4
Product overview	6
Technical concepts	8
Introduction to SKF ball screws	8
Basic dynamic load carrying capacity (C_a)	8
Nominal fatigue life L_{10}	8
Service life	8
Equivalent dynamic load	8
Basic static load carrying capacity (C_{0a})	9
Critical rotating speed for screw shafts	9
Permissible speed limit	9
Screw shaft buckling	9
Lubrication	9
Efficiency and back-driving	10
Axial play and preload	10
Static axial stiffness of a complete assembly	10
Materials, heat treatment and coatings	11
Operating temperature	11
Ball screw support bearings	11
Designing the screw shaft ends	11
Critical applications	11
Working environment	11

B Recommendations for assembly

Assembly procedure	12
Storage	12
Alignment	12
Lubrication	12
Removing the nut / assembling the nut on the shaft	12
Wipers assembly	12
Starting-up the screw	12

C Lead precision

Manufacturing precision	14
High precision rolled ball screws	14
Lead precision	15

D Product information

SD/BD/SH miniature screws	16
SDS/BDS/SHS miniature screws in stainless steel	18
SX/BX universal screws	20
Dedicated flanges for SX/BX nuts	22
SND/BND precision screws, DIN standard 69051	24
PND preloaded screws, DIN standard 69051	26
SN/BN precision screws	28
PN preloaded screws	30
SL/TL long lead screws	32
SLT/TLT rotating nut	34
Shaft end combinations	36
Standard machined ends	37
Ball screw support bearings	42
Examples of customized nuts	48
Manufacturing tolerances	49
Calculation formulae	50

E Service range

F Design calculation and inquiry form

Designation system	58
--------------------------	----

G SKF Linear Motion

Rollers screws, electromechanical cylinders and guiding solutions	59
--	----

SKF – the knowledge engineering company

From one simple but inspired solution to a misalignment problem in a textile mill in Sweden, and fifteen employees in 1907, SKF has grown to become a global industrial knowledge leader.



Over the years we have built on our expertise in bearings, extending it to seals, mechatronics, services and lubrication systems. Our knowledge network includes 46 000 employees, 15 000 distributor partners, offices in more than 130 countries, and a growing number of SKF Solution Factories around the world.

Research and development

We have hands-on experience in over forty industries, based on our employees' knowledge of real life conditions. In addition our world-leading experts and university partners who pioneer advanced theoretical research and development in areas including tribology, condition monitoring, asset management and bearing life theory. Our ongoing commitment to research and development helps us keep our customers at the forefront of their industries.



Meeting the toughest challenges

Our network of knowledge and experience along with our understanding of how our core technologies can be combined helps us create innovative solutions that meet the toughest of challenges. We work closely with our customers throughout the asset life cycle, helping them to profitably and responsibly grow their businesses.

Working for a sustainable future

Since 2005, SKF has worked to reduce the negative environmental impact from our own operations and those of our suppliers. Our continuing technology development introduced the SKF BeyondZero portfolio of products and services which improve efficiency and reduce energy losses, as well as enable new technologies harnessing wind, solar and ocean power. This combined approach helps reduce the environmental impact both in our own operations and in our customers'.

SKF Solution Factories make SKF knowledge and manufacturing expertise available locally, to provide unique solutions and services to our customers.

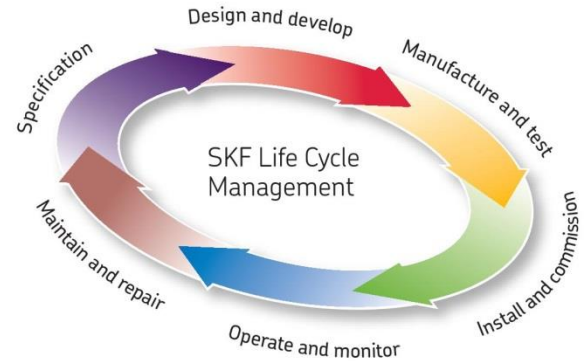


Working with SKF IT and logistics systems and application experts, SKF Authorized Distributors deliver a valuable mix of product and application knowledge to customers worldwide.



Our knowledge – your success

SKF Life Cycle Management is how we combine our technology platforms and advanced services, and apply them at each stage of the asset life cycle, to help our customers to be more successful, sustainable and profitable.



Working closely with you

Our objective is to help our customers improve productivity, minimize maintenance, achieve higher energy and resource efficiency, and optimize designs for long service life and reliability.

Innovative solutions

Whether the application is linear or rotary or a combination of the two, SKF engineers can work with you at each stage of the asset life cycle to improve machine performance by looking at the entire application. This approach doesn't just focus on individual components like bearings or seals. It looks at the whole application to see how each component interacts with the next.

Design optimization and verification

SKF can work with you to optimize current or new designs with proprietary 3-D modeling software that can also be used as a virtual test rig to confirm the integrity of the design.



Bearings

SKF is the world leader in the design, development and manufacture of high performance rolling bearings, plain bearings, bearing units and housings.



Machinery maintenance

Condition monitoring technologies and maintenance services from SKF can help minimize unplanned downtime, improve operational efficiency and reduce maintenance costs.



Sealing solutions

SKF offers standard seals and custom engineered sealing solutions to increase uptime, improve machine reliability, reduce friction and power losses, and extend lubricant life.



Mechatronics

SKF fly-by-wire systems for aircraft and drive-by-wire systems for off-road, agricultural and forklift applications replace heavy, grease or oil consuming mechanical and hydraulic systems.



Lubrication solutions

From specialized lubricants to state-of-the-art lubrication systems and lubrication management services, lubrication solutions from SKF can help to reduce lubrication related downtime and lubricant consumption.



Actuation and motion control

With a wide assortment of products – from actuators and ball screws to profile rail guides – SKF can work with you to solve your most pressing linear system challenges.

Product overview

Screw assembly



SD/BD – SDS/BDS



SH – SHS

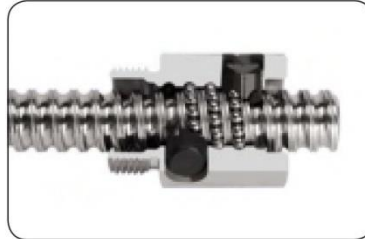


SX/BX



SND/BND/PND, DIN 69051

Type of recirculation



Internal, by inserts
Stainless steel optional ¹⁾



External, by integrated tube
Stainless steel optional ²⁾



Internal, by inserts



Internal, by inserts

Designation	d ₀	P _h	Page
	mm	mm	
SD/BD/SDS/BDS	8	2,5	16
SD/BD/SDS/BDS	10	2	
SD/BD	10	4	18
SD/BD/SDS/BDS	12	2–4–5	
SD/BD/SDS/BDS	14	4	16
SD/BD/SDS/BDS	16	2–5	
SD/BD	16	10	18
SH/SHS	6	2	16
SH	10	3	
SH	12,7	12,7	
SX/BX	20	5	20
SX/BX	25	5–10	
SX/BX	32	5–10	
SX/BX	40	5–10–40	
SX/BX	50	10	
SX/BX	63	10	
SND/BND/PND	16	5–10	24
SND/BND/PND	20	5	
SND/BND/PND	25	5–10	
SND/BND/PND	32	5–10	
SND/BND/PND	40	5–10	
SND/BND/PND	50	10	
SND/BND/PND	63	10	

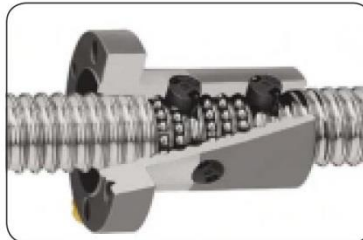
¹⁾ except 10x4 R and 16x10 R
²⁾ 6x2 R only.

Screw assembly



SN/BN/PN

Type of recirculation



Internal, by inserts



SL/TL – SLD/TLD



By faces



SLT/TLT rotating nuts



By faces



Ball screw support bearings FLBU, PLBU, BUF



Complete ball screw assembly with support bearing

Designation	d ₀	P _h	Page
	mm	mm	
SN/BN/PN	16	5	28
SN/BN/PN	20	5	
SN/BN/PN	25	5–10	
SN/BN/PN	32	5–10	
SN/BN/PN	40	5–10	
SN/BN/PN	50	10	
SN/BN/PN	63	10	32
SL/TL	25	20–25	
SL/TL	32	20–32–40	
SLD/TLD	32	32	
SL/TL	40	20–40	
SL/TL	50	50	
SLT/TLT	25	20–25	34
SLT/TLT	32	20–32–40	
SLT/TLT	40	20–40	
SLT/TLT	50	50	
FLBU/PLBU/BUF	16		44
FLBU/PLBU/BUF	20		
FLBU/PLBU/BUF	25		
FLBU/PLBU/BUF	32		
FLBU/PLBU/BUF	40		
FLBU/PLBU/BUF	50		
FLBU/PLBU/BUF	63		

Technical concepts

Introduction to SKF ball screws

This catalogue describes SKF expertise, technology and solutions related to precision rolled ball screws. Thanks to our lengthy experience with manufacturing ball screws and continuous product and process development, SKF provides customers with precision rolled ball screw solutions that fulfil their most demanding applications in terms of efficiency, precision, durability and value.

In many cases, these ball screws can replace ground ball screws, offering a similar level of performance and precision at a lower cost.

The high quality of SKF rolled ball screws is achieved through our dedicated manufacturing processes, including precision rolling and specific heat-treatment.

Ball screws convert rotary motion into linear motion, and vice-versa, and loads are transferred from the screw shaft to the nut through a ball set: in this sense, ball screws relate to general bearing technology. Various types of bearing steel are used to attain the hardness and material fatigue properties required for carrying heavy application loads over extended periods of service. Some bearing concepts such as load ratings, load cycles, nominal and service life, stiffness, speed ratings, lubrication requirements, etc. are explained below to guide customers through the ball screw selection process.

Only basic selection parameters are included in this chapter. To make the very best selection of a ball screw, the designer should

consider critical parameters such as the load cycle, the linear or rotational speed, the rates of acceleration and deceleration, the cycle rate, the environment, the required life, the lead accuracy, the stiffness, and any other special requirements. If in doubt, please consult the SKF ball screw assembly specialists who will assist you in the selection process.

Basic dynamic load carrying capacity (C_a)

The dynamic load rating capacity is used to compute the nominal fatigue life of ball screws. It results from the axial load, constant in magnitude and direction, which acts along the central axis of the ball screw, resulting in the calculated nominal life as defined by ISO of one million revolutions.

With a given combination of nominal diameter and lead, a ball screw's dynamic and static load carrying capacities are determined by the number of ball turns supporting the load.

For each product family, the type and number of circuits generate a specific number of ball turns. For example, the SH type nut with external tube recirculation typically presents 2,5 turns of balls within a circuit. The standard SD type nut has 3 circuits covering 0,9 turns each.

Nominal fatigue life L_{10}

Nominal fatigue life is, according to the ISO definition, the life achieved or exceeded by 90% of a large-enough group of apparently identical ball screws, working under identical conditions (alignment, axially and centrally applied load, speed, acceleration, lubrication, temperature and cleanliness).

The nominal life of a ball screw is the statistical number of revolutions which the ball screw is capable of reaching before the first signs of material fatigue by flaking occur on one of the rolling surfaces.

Service life

The actual life achieved by a specific ball screw before it fails is known as "service life." Failure is due not only to material fatigue by flaking, but also to inadequate lubrication, wear of the recirculation system, corrosion, contamination and, more generally, loss of the functional characteristics required by the application.

Experience acquired with similar applications will help in selecting the right screw to obtain the necessary service life. Structural requirements such as the strength of screw ends and nut attachments should be considered.

To attain L_{10} life performance, a mean working load of up to 60% of C_a (to limit the Hertz pressure at the balls / raceways contacts) and a stroke higher than 4 leads (to avoid false-brinelling which could occur with very short strokes or oscillation movements) are required.

Equivalent dynamic load

The loads acting on the screw can be calculated according to the laws of mechanics if the external forces (e.g. power transmission, work, rotary and linear inertia forces) are known or can be calculated. It is necessary to calculate the equivalent dynamic load.

Radial and moment loads must be taken up by linear bearing systems. It is extremely important to resolve these problems at the earliest possible design stage. These forces are detrimental to the life and the expected performance of the screw (→ fig. 1).

When the load fluctuates during the working cycle, it is necessary to calculate the equivalent dynamic load: this load is defined as the hypothetical load, constant in magnitude and direction, acting axially and centrally on the screw, which if applied, would have the same influence on the screw life as the actual loads which the screw is subjected to.

If misalignment, uneven loading, shocks, etc. cannot be avoided in the application,

Life test bench



they must be taken in account during the sizing of the ball screw.

Their influence on the screw's nominal life can generally be estimated¹⁾.

Basic static load carrying capacity (C_{0a})

Ball screws should be selected considering the basic static load capacity C_{0a} , rather than the basic dynamic load capacity, when they are subjected to continuous or intermittent shock loads while stationary or rotating at very low speed for short periods of time. The permissible load is determined by the permanent deformation caused by the load acting at the contact points.

The static load carrying capacity is, according to ISO standards, the purely axially and centrally applied static load which creates, by calculation, a total (rolling element + threaded surface) permanent deformation equal to 0.0001 times the diameter of the rolling element (→ fig. 2).

A ball screw basic static load rating must be, at a minimum, equal to the product of the maximum axial static load applied and a safety factor "s₀." Past experience with similar applications and requirements of running smoothness and noise level will guide the selection of "s₀"¹⁾.

Critical rotating speed for screw shafts

For this calculation, the shaft is equated to a cylinder, with an external diameter equal to the root diameter of the thread. The formulae use a parameter whose value is dictated by the mounting of the screw shaft, whether it is simply supported or fixed.

As a general rule, the nut is not considered to be a support of the screw shaft. Because of the potential inaccuracies in the mounting of the screw assembly, a safety factor of 0,8 is applied to the calculated critical speed.

Calculations which consider the nut to be a support for the shaft, or which reduce the safety factor, require practical tests and possibly optimization of the design.

Permissible speed limit

The permissible speed limit is the speed which a screw cannot reliably exceed at any time. It is generally the limiting speed of the recirculation system in the nut. It is expressed as the product of maximum rotational speed (in rpm) and the nominal diameter of the screw shaft (expressed in mm).

The speed limits quoted in this catalogue (→ page 48) are the maximum speeds that may be applied for very short periods of time and with optimized running conditions of alignment, light external load and preload with monitored lubrication.

Running a screw continuously at the permissible speed limit may lead to a reduction of the calculated life of the nut mechanism.

Important!

High speed associated with high load requires a large input torque and yields a relatively short nominal life¹⁾.

In the case of high acceleration and deceleration, we recommend either working under a nominal external load or applying a light preload to the nut to avoid internal sliding during reversal of movement.

The preload for screws subjected to high velocity must be calculated to ensure that the rolling elements do not slide¹⁾.

Excessive preload will create an unacceptable increase in the internal temperature.

Screw shaft buckling

The column loading of the screw shaft must be checked when it is subjected to dynamic or static compression loading.

The maximum permissible compressive load is calculated using the Euler formulae, with a safety factor of 3 to 5, depending on the application.

The type of shaft end mounting is critical to select the proper coefficients to be used in the Euler formulae.

When the screw shaft has a single diameter along its total length, the root diameter of the threaded shaft is used for the calculation. When the screw comprises different sections with varying diameters, calculation becomes more complex¹⁾.

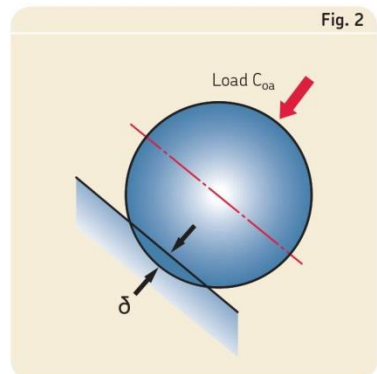
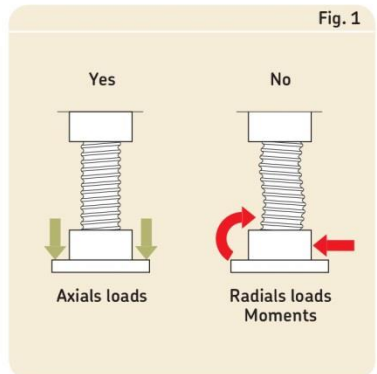
Lubrication

Proper quantities and quality of lubrication must be selected if ball screws are to operate correctly and to maximize their service life.

Greater care is required for operation at high speed, as the lubricant spread on the surface of the screw shaft may be thrown off by centrifugal forces. It is important to monitor this phenomenon during the first run at high speed and, if necessary, to adapt the frequency of re-lubrication or the flow of lubricant, or to select a lubricant with a different viscosity.

Monitoring the steady temperature reached by the nut allows for the optimization of the frequency of re-lubrication or the oil flow rate.

SKF SYSTEM 24 automatic lubrication kit can be adapted to most precision rolled ball screws.



¹⁾ SKF can help you make these calculations with consideration to the actual conditions of service.

Efficiency and back-driving

Screw performance primarily depends on the geometry of the contact surfaces and their finish and the helix angle of the thread. It also depends on the working conditions (load, speed, lubrication, preload, alignment, etc.).

“Direct efficiency” is used to define the input torque required to transform the rotation of one component into the translation of the other. Conversely, “indirect efficiency” is used to define the axial load required to transform the translation of one component into the rotation of the other one. It is also used to define the braking torque required to prevent that rotation.

It is safe to assume that ball screws are reversible or back-driveable under almost all circumstances. A braking mechanism (gear reducers or brake) must be part of the design, if back-driving is to be avoided.

Preload torque

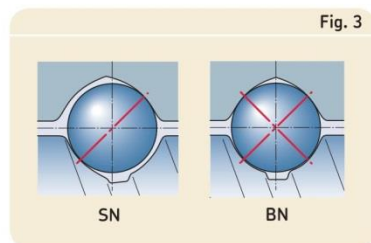
Screws with internal preload exhibit a certain amount of friction torque. This torque still exists when ball screws are not externally loaded. Preload torque is measured with ISO grade 64 oil.

Starting torque

This is the amount of torque required to overcome the following forces to start rotation:

- a the total inertia of all moving parts accelerated by the source of power (including rotational and linear movements);
- b the internal friction of the screw / nut assembly, bearings and associated guiding devices.

In general, the torque required to overcome the inertia (a) is greater than the friction torque (b). The friction coefficient of the high efficiency screw when starting moving (μ_s) is estimated to reach up to double the amount of the dynamic coefficient μ , under normal conditions of usage.



Axial play and preload

SKF products are available with a range of versions of axial play. Standard axial play is intended for transport screws, when the product is not subject to vibrations, high accelerations, and when positioning accuracy under load is not critical (e.g.: SN type).

Reduced play (e.g.: SN type with reduced play) and backlash elimination by oversized balls (e.g.: BN type) are recommended to increase assembly precision (→ fig. 3).

For optimum stiffness and positioning accuracy under load, internally preloaded nuts are recommended (e.g.: PN type) (→ fig. 4). When subjected to external loading, preloaded nuts exhibit a much lower elastic deformation than non-preloaded nuts.

Preload is the amount of force applied to a set of two half-nuts necessary to either press them together or to push them apart with the purpose of eliminating backlash or increasing the stiffness of the assembly. The preload is measured by the value of the preload torque (see explanations in the previous paragraph). For a given amount of preload (expressed in Newton), the friction torque varies with different types of nuts and with the preloading method. The friction torque due to preload is indicated in product tables.

Static axial stiffness of a complete assembly

The static axial stiffness of a complete ball screw assembly is the ratio of the external axial load applied to the system and the axial displacement of the face of the nut in relation to the fixed (anchored) end of the screw shaft. Please see calculation formulae (→ pages 48 to 49).

Nut stiffness: R_n

When a preload is applied to a split nut, the internal play is eliminated. Additionally, the Hertzian elastic deformation increases with increased preload and increased stiffness.

The theoretical elastic deformation at the contact points does not take into account machining inaccuracies, actual sharing of the load between the different contact surfaces, or elasticity of the nut and of the screw shaft. For this reason, the practical stiffness values given in the catalogue are lower than the theoretical values. They are determined by SKF assuming a preload of 8,5% C_a for screws with diameter up to 40 mm, and a preload of 7% C_a for screws with diameter greater than 40 mm, when applying an external axial load centred on the screw shaft and equal to twice the amount of preload.

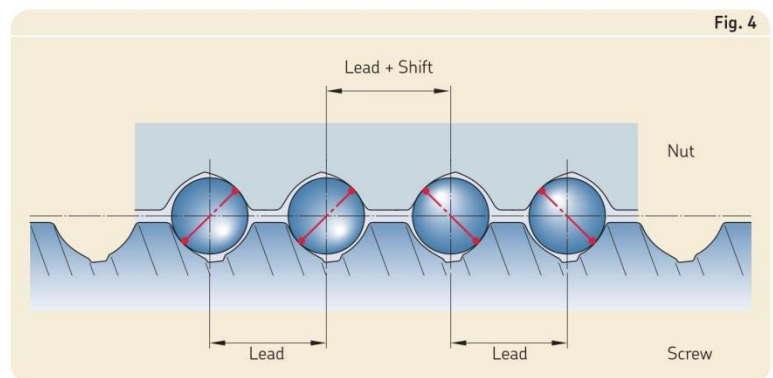
Shaft stiffness: R_s

The elastic deformation of the screw shaft is proportional to its length and inversely proportional to the square of the root diameter.

According to the relative importance of the screw deformation, an excessive increase of nut preload and of the supporting bearings yields a limited increase of stiffness and noticeably increases the preload torque and therefore the running temperature.

Consequently, the preload stated in the catalogue for each screw dimension is optimum and should not be exceeded.

Please see calculation formulae (→ pages 48 to 49).



Materials, heat treatment and coatings

Standard screw shafts are manufactured from carbon steel which is surface hardened by induction. For standard screws, rolling surface hardness is 56 to 60 HRc, depending on diameter (for very small diameter screws, the temperature during the hardening process is slightly lowered to avoid the through-hardening of the screw shaft, therefore resulting in lower surface hardness).

Standard nuts are machined from steel which is through-hardened (100 Cr6–NFA 35.565 or equivalent for diameters ≥ 20 mm, and carbon steel for diameters < 20 mm).

Most stainless steel screws have a surface hardness ranging from 50 to 58 HRc, depending on the type of stainless steel being used and the screw diameter (note the effect of reduced hardening temperature on small diameter screws, as previously mentioned). The load ratings provided in the catalogue are given for standard screws only.

SKF offers various types of surface coating for improved ball screw performance:

- Manganese phosphate coating is standard for the SX/BX universal nuts. This coating can also be applied to most ranges of precision rolled ball screws to improve the resistance to corrosion
- Low friction coating or chrome coating are available on request. Please contact SKF.

Operating temperature

Screws made from standard steel and screws operating under normal loads can operate from -20 to $+110$ °C.

Between 110 °C and 130 °C, SKF must be notified for adaptation of the annealing procedure and for review of the application with hardness below the standard minimum value.

Above 130 °C, steel adapted to the temperature of the application should be selected (100Cr6, special steel, etc.). Please consult SKF for advice.

Operation at high temperatures will lower the steel hardness, alter the thread accuracy, may increase the oxidation of the materials and change the lubricant properties.

Ball screw support bearings

To assist the customer design and machinery assembly process, SKF has developed a range of support bearings specifically designed for ball screws with nominal diameter starting from 16 mm. These support bearings can easily be mounted on the screw shaft ends, following SKF recommendations for ends machining (\rightarrow pages 36 to 41). Three types of support bearings available for fixed axial mounting (FLBU type in pages 42 to 43), for fixed radial mounting (PLBU type in pages 44 to 45) and for pure radial support (BUF type in pages 46 to 47), all fitted with SKF premium bearings, greased and sealed for life. SKF stocks these support bearings for quick delivery.

Designing the screw shaft ends

Generally speaking, when the ends of the screw shaft are specified by the customer's engineering staff, it is their responsibility to check the strength of these ends. However, we offer and recommend a choice of standard machined ends (pages 36 to 41).

Please bear in mind that no dimension on the shaft ends can exceed d_0 . Otherwise, traces of the root of the thread will appear. If the application requires a shaft end with a smooth surface of diameter greater than d_0 , it is advisable to add an additional part attached to the machined shaft end.

A minimum shoulder should be sufficient to maintain the bearing inner ring. Please follow bearing mounting recommendations.

Critical applications

The standard products have been fitted with composite ball recirculation inserts.

If the ball screws are used in severe applications, or if the inserts are used to prevent system collapse (especially in the case of vertical applications), optional steel inserts are available.

For critical applications, SKF also offers optional safety rings for miniature ball screws, and safety nuts for larger ball screws.

In such cases, the customer should consult SKF to define the optimum solution.

Working environment

Our products have not been developed for use in an explosive environment. Consequently, SKF cannot take any responsibility for the use of ball screws in such applications.

¹⁾ SKF can help you with these calculations, taking into account the working conditions.

Assembly procedure

Ball screw assemblies are precision components and should be handled with care to avoid damaging shocks, contamination or corrosion.

Storage

Storage location must ensure that ball screw assemblies are not exposed to contamination, shocks, humidity and other detrimental actions.

When stored out of the shipping crate, ball screw assemblies must lie on wooden or plastic V-shaped blocks and should not be allowed to bounce. The assembly must not be supported on the shelf by the nut body.

During shipping, ball screw assemblies are wrapped in heavy gauge plastic bags, which protect them from foreign material and possible contamination. They should remain wrapped until they are used.

Alignment

After assembly, any radial load or moment loading on the nut will overload some of the contact surfaces, thus significantly reducing the service life (→ **fig. 1**).

SKF linear guidance components should be used to ensure correct alignment and to avoid non-axial loading. The parallelism of the screw shaft with the guiding devices must be checked carefully. If external linear guidance proves impractical, we suggest mounting the nut on trunnions or gimbals, and mounting the screw shaft on self-aligning bearings.

Mounting the screw in tension helps to align it properly and eliminates buckling.

Lubrication

Good lubrication is essential for the proper operation and long term reliability of the ball screw assembly. If necessary, please consult SKF.

Before shipping, the complete ball screw assembly is coated with a protective fluid that dries to a film. This protective film is not a lubricant. Depending on the lubricant selected for the application, it may be necessary to remove the protective film before applying the lubricant in order to eliminate any risk of incompatibility. In such cases, we recommend the following procedure:

- 1 Dip the ball screw assembly into a solvent
- 2 Shake and rotate the assembly to allow the solvent to penetrate
- 3 Remove the assembly from the solvent and allow the solvent to drain.

Removing the nut / assembling the nut on the shaft

Removing the nut from the screw shaft

If possible, do not remove the nut from the shaft, especially for preloaded assemblies. If the nut must be removed from the shaft, i.e. for shaft end machining, check the nut orientation before disassembly.

Never unscrew the nut from the shaft without a mandrel or sleeve to prevent the balls from falling off the nut (→ **fig. 6**).

Once the nut is engaged on the sleeve, use a tie wrap to secure the nut assembly (→ **fig. 5**).

Fitting sleeved nut onto screw shaft

Sleeved nuts should not be removed from the sleeve until final assembly.

- 1 Remove the retaining strap (→ **fig. 5**)
- 2 Check the assembly drawing to confirm the nut orientation
- 3 Hold the sleeve against the ball track of the screw shaft and smoothly engage the ball nut (→ **fig. 6**)

If the sleeve does not cover the diameter next to the ball track (for example, the sleeve bore diameter is smaller than the screw shaft end), then adhesive tape can be used to match the shaft end to the sleeve outer diameter. Otherwise, the sleeve can be held against the unmachined end, if available, with extreme care to prevent the balls from falling off the nut

- 4 Without using force, completely engage the nut in the screw thread, and run the nut to full engagement on the screw shaft.

Wiper assembly

If optional wipers have been ordered, please refer to the fitting instructions enclosed with the shipment.

Starting-up the screw

After the assembly has been cleaned, fitted and lubricated, allow the nut to make several full strokes at low speed (< 50 rpm) and light load (not to exceed 5% of the ball screw dynamic carrying capacity) in order to check the proper positioning of the limit switches or reversing mechanism. Then, normal load and speed can be applied.

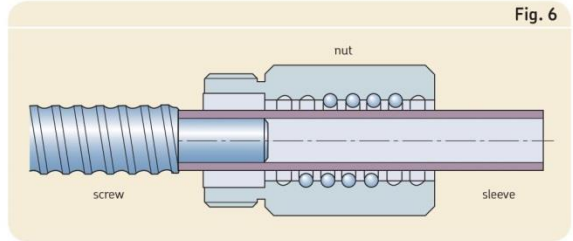
Note:

Instructions for most operations such as fitting a nut onto a screw shaft, a wiper onto a nut, etc. are available in separate sheets delivered with the product. Please refer to them before assembling the screw.

Fig. 5



Fig. 6



B

Notes

A large grid area for taking notes, consisting of a 20x20 grid of small squares. The grid is light blue and occupies most of the lower half of the page.

Lead precision

Manufacturing precision

Generally speaking, the precision indicated defines the lead precision that complies with ISO standards, e.g. G5, G7, etc. (→ **table 1**). Parameters other than lead precision correspond to SKF internal standards, generally based on ISO class 7. If the application requires special tolerances, for example class 5, please specify these requirements in the inquiry.

SKF high precision rolled ball screws

High technology machinery associated with precise control of the cold forming and metallurgical processes results in screw production that virtually offers the same accuracy and performance level of ground ball screws, but at a lower cost (→ **diagram 1**). Standard lead precision is G9, which complies with ISO 286-2:1988. SKF production meets G7 lead precision for screw shaft diameters starting from $d_0 = 20$ mm. On request, SKF can deliver ball screws with G5 lead precision which are in accordance with ISO 3408-3:2006, defined for positioning screws and matching the lead precision of G5 ground ball screws.

Lead precision

Lead precision is measured at 20 °C on the useful stroke l_u . At SKF l_u is the threaded length of the shaft minus twice the length l_e equal to the screw nominal diameter (→ **table 1 and fig. 7**).

Some customer applications require a travel compensation c to account for the effect of operating temperature on the lead precision:

- Standard case with $c = 0$ (→ **fig. 8**)
- Case with specific value of c (→ **fig. 9**).

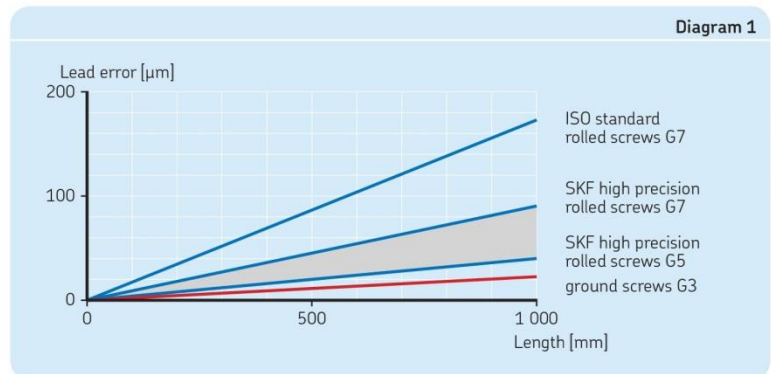
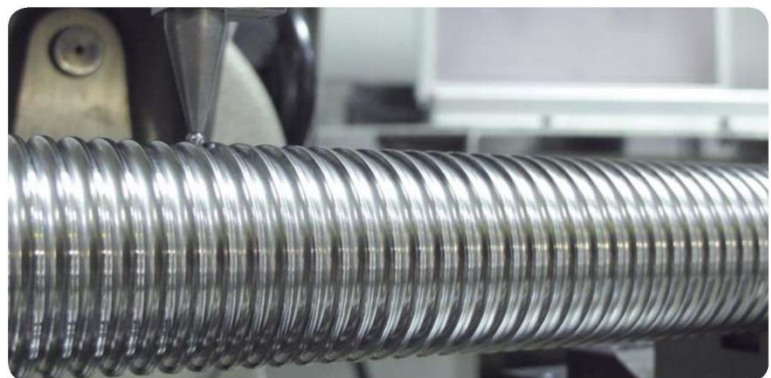


Table 1

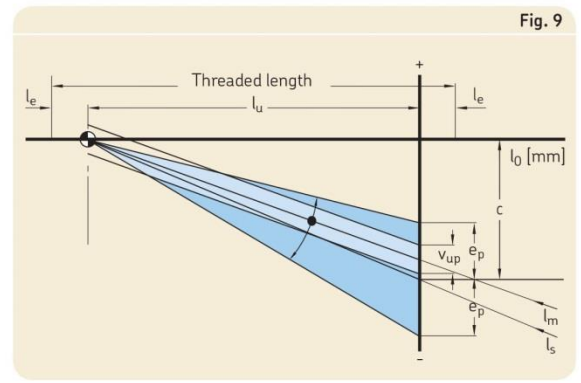
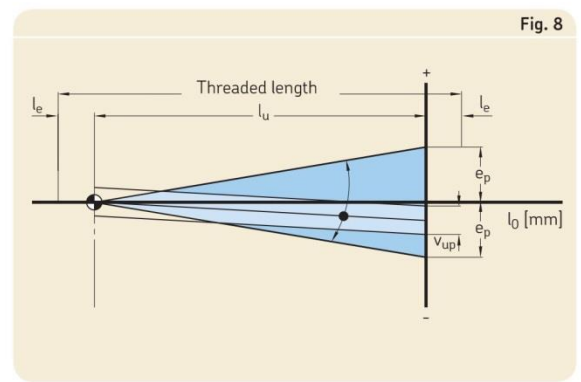
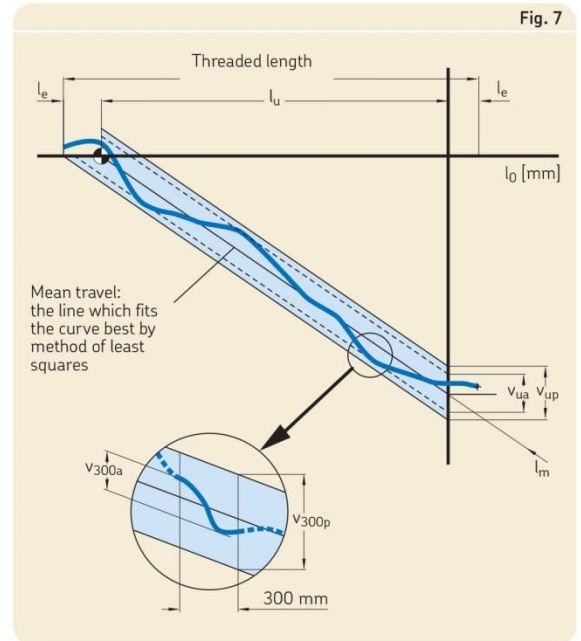
V_{300p} l_u	G5		G7		G9	
	e_p	v_{up}	e_p	v_{up}	e_p	v_{up}
mm	µm					
0	315	23	52	35	130	87
(315) - 400	25	25	57	40	140	100
(400) - 500	27	26	63	46	155	115
(500) - 630	32	29	70	52	175	130
(630) - 800	36	31	80	57	200	140
(800) - 1 000	40	34	90	63	230	155
(1 000) - 1 250	47	39	105	70	260	175
(1 250) - 1 600	55	44	125	80	310	200
(1 600) - 2 000	65	51	150	90	370	230
(2 000) - 2 500	78	59	175	105	440	260
(2 500) - 3 150	96	69	210	125	530	310
(3 150) - 4 000	115	82	260	150	640	370
(4 000) - 5 000	140	99	320	175	790	440
(5 000) - 6 000	170	119	390	210	960	530



Lead precision measurement

Symbols used in figs. 7 to 9

- l_u = useful travel
- l_e = excess travel (no lead precision required)
- l_0 = nominal travel
- l_s = specified travel
- c = travel compensation (difference between l_s and l_0 to be defined by the customer)
- e_p = tolerance over the specified travel
- V = travel variation (or permissible band width)
- V_{300p} = maximum permitted travel variation over 300 mm
- V_{up} = maximum permitted travel variation over the useful travel l_u
- V_{300a} = measured travel variation over 300 mm
- V_{ua} = measured travel variation over l_u



SD/BD/SH miniature screws

Rolled thread miniature ball screw, nut with threaded nose

Features

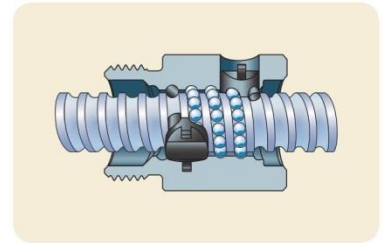
- Nominal diameter from 6 to 16 mm
- Lead from 2 to 12,7 mm
- Recirculation with inserts (SD/BD) or with tube (SH)
- Optional surface coating on shaft and nut
- Optional safety ring¹⁾
- Optional wipers²⁾ except 6×2 R – 10×3 R.

Benefits

- Excellent repeatability with high positioning accuracy
- Smooth running
- Extremely compact nut design with threaded nose for easy assembly
- Backlash elimination by oversized balls on request (BD designation), over maximum length of 1 000 mm.



Standard SD



Recirculation SD/BD



Standard SH

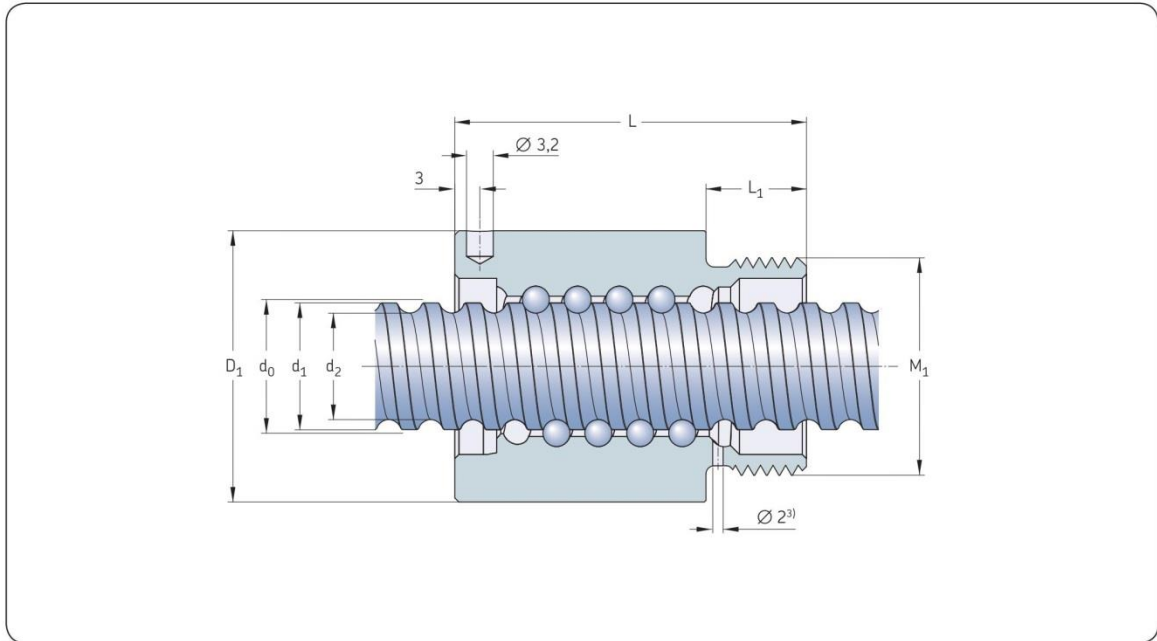


Customised SD

Nominal diameter	Lead (right hand)	Nut		Number of circuits of balls	Std play	Reduced play on request	Inertia	Grease	Weight	Screw			Designation
		Basic load ratings dynamic	static							Mass	Inertia	Grease	
d_0	P_h	C_a	C_{oa}		mm		kgmm^2	cm^3	kg	kg/m	kgmm^2/m	cm^3/m	
6	2	1,9	2,2	1×2,5	0,05	0,02	7,7	0,1	0,025	0,18	0,7	0,7	SH 6×2 R
8	2,5	2,2	2,7	3	0,07	0,03	1,12	0,1	0,025	0,32	2,1	1,1	SD/BD 8×2.5 R
10	2	2,5	3,6	3	0,07	0,03	1,7	0,1	0,03	0,51	5,2	1,4	SD/BD 10×2 R
	3	2,6	3,3	1×2,5	0,07	0,03	2,9	0,3	0,05	0,5	5,1	1,3	SH 10×3 R
	4	4,5	5,5	3	0,07	0,03	2,7	0,3	0,04	0,43	3,8	1,3	SD/BD 10×4 R
12	2	2,9	4,7	3	0,07	0,03	1,5	0,1	0,023	0,67	10	1,7	SD/BD 12×2 R
	4	4,9	6,6	3	0,07	0,03	7	0,4	0,066	0,71	10,8	1,6	SD/BD 12×4 R
	5	4,2	5,4	3	0,07	0,03	5	0,6	0,058	0,71	10,1	1,4	SD/BD 12×5 R
12,7	12,7	6,6	8,9	2×1,5	0,07	0,03	20	1,1	0,15	0,71	16,2	1,6	SH 12,7×12,7 R
14	4	6	9,1	3	0,07	0,03	8	0,6	0,083	1,05	22	1,7	SD/BD 14×4 R
16	2	3,3	6,2	3	0,07	0,03	9,2	0,6	0,1	1,4	39,7	1,7	SD/BD 16×2 R
	5	7,6	10,7	3	0,07	0,03	22,7	0,9	0,135	1,3	33,9	2,1	SD/BD 16×5 R
	10	10,7	17,2	2×1,8	0,07	0,03	24,4	1	0,16	1,21	30,7	1,9	SD/BD 16×10 R

¹⁾ Available for 12×4 R – 12,7×12,7 R – 14×4 R – 16×5 R – 16×10 R

²⁾ It is not possible to supply safety ring and wipers in the same nut



Screw $d_0 \times P_h$ mm	Nut		Without wiper L $\pm 0,3$ mm	With wiper L ₁ mm	Tightening spanner (FACOM) –	Screw length max. mm	d ₂ mm	d ₁ mm	Support bearing Recommended thrust support bearings –	Recommended support pillow block
	D ₁ h10 mm	M ₁ 6g mm								
6x2	16,5	M14x1	20	–	7,5	126-A35	1 000	4,7	6	
8x2,5	17,5	M15x1	23,5	23,5	7,5	126-A35	1 000	6,3	7,6	
10x2	19,5	M17x1	22	22	7,5	126-A35	1 000	8,3	9,5	
10x3	21	M18x1	29	–	9	126-A35	1 000	7,9	9,9	
10x4	21	M18x1	28	33	8	126-A35	1 000	7,4	8,9	
12x2	20	M18x1	20	23,5	8	126-A35	2 000	9,9	11,2	
12x4	25,5	M20x1	34	34	10	126-A35	2 000	9,4	11,3	
12x5	23	M20x1	36	40	10	126-A35	2 000	9,3	11,8	
12,7x12,7	29,5	M25x1,5	50	50	12	126-A35	2 000	10,2	13	
14x4	27	M22x1,5	30	34	8	126-A35	2 000	11,9	13,7	
16x2	29,5	M25x1,5	27	27	12	126-A35	2 000	14,3	15,5	FLBU 16/PLBU 16
16x5	32,5	M26x1,5	42	42	12	126-A35	2 000	12,7	15,2	FLBU 16/PLBU 16
16x10	32	M26x1,5	46	46	12	126-A35	2 000	12,6	15,2	FLBU 16/PLBU 16

SDS/BDS/SHS miniature screws in stainless steel

Rolled thread miniature ball screw, nut with threaded nose

Features

- Nominal diameter from 6 to 16 mm
- Lead from 2 to 5 mm
- Standard lead precision G7 and G9
- Material for shaft and nut is X30Cr13 (similar to AISI 420)
- Balls are made of stainless steel type X105CrMo17 (similar to AISI 440C)¹⁾
- Optional safety ring²⁾
- Optional wipers³⁾ except for SHS 6×2 R.



Standard SDS



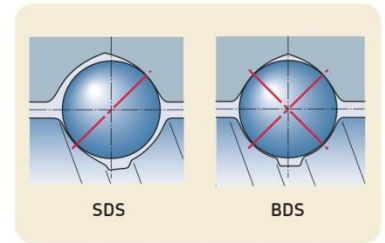
Standard SHS

Benefits

- Excellent repeatability with high positioning accuracy
- Smooth running
- Extremely compact nut design with threaded nose for easy assembly
- Backlash elimination by oversized balls on request (BDS designation), over maximum length of 1 000 mm
- Suitable for long storage periods before customer usage, or for applications with extremely long service life
- Adapted for operation in clean environment.



Customised SDS

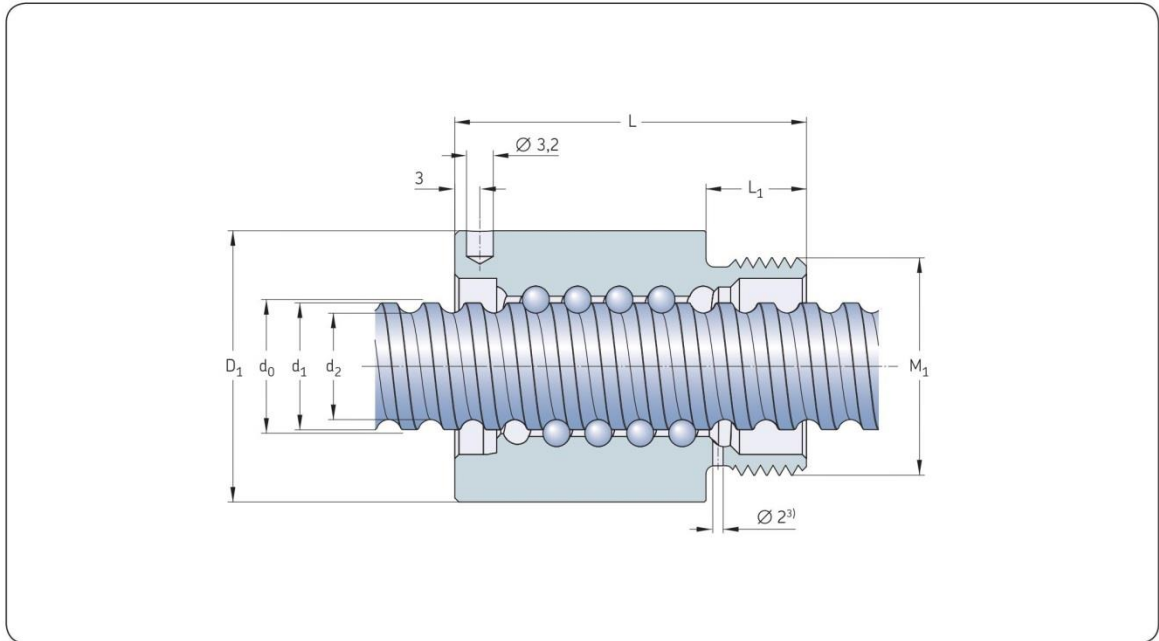


Nominal diameter	Lead (right hand)	Nut		Number of circuits of balls	Std play	Reduced play on request	Inertia	Grease	Weight	Screw			Designation
		Basic load	ratings							Mass	Inertia	Grease	
d_0	P_h	C_a	C_{oa}		mm	mm	kgmm^2	cm^3	kg	kg/m	kgmm^2/m	cm^3/m	
6	2	1,2	1,1	1×2,5	0,05	0,02	7,7	0,1	0,025	0,18	0,7	0,7	SHS 6×2 R
8	2,5	1,4	1,3	3	0,07	0,03	1,12	0,1	0,025	0,32	2,1	1,1	SDS/BDS 8×2,5 R
10	2	1,6	1,8	3	0,07	0,03	1,7	0,1	0,03	0,51	5,2	1,4	SDS/BDS 10×2 R
12	2	1,9	2,3	3	0,07	0,03	1,5	0,1	0,023	0,67	10	1,7	SDS/BDS 12×2 R
	4	3,1	3,3	3	0,07	0,03	7	0,4	0,066	0,71	10,8	1,6	SDS/BDS 12×4 R
	5	2,7	2,7	3	0,07	0,03	5	0,6	0,058	0,71	10,1	1,4	SDS/BDS 12×5 R
14	4	3,8	4,6	3	0,07	0,03	8	0,6	0,083	1,05	22	1,7	SDS/BDS 14×4 R
16	2	2,1	3,1	3	0,07	0,03	9,2	0,6	0,1	1,4	39,7	1,7	SDS/BDS 16×2 R
	5	4,8	5,4	3	0,07	0,03	22,7	0,9	0,135	1,3	33,9	2,1	SDS/BDS 16×5 R

¹⁾ Except for size SDS/BDS 16×5 R using steel type 100 Cr6 (similar to AISI 52100)

²⁾ Available for 12×4 R – 14×4 R – 16×5 R

³⁾ It is not possible to supply safety ring and wipers in the same nut



Screw $d_0 \times P_h$ mm	Nut		Without wiper $L \pm 0,3$ mm	With wiper L_1 mm	Tightening spanner (FACOM) -	Screw length max. mm	d_2 mm	d_1 mm	Support bearing Recommended thrust support bearings -	Recommended support pillow block
	D_1 h_{10} mm	M_1 6g mm								
6x2	16,5	M14x1	20	-	7,5	126-A35	1 000	4,7	6	
8x2,5	17,5	M15x1	23,5	23,5	7,5	126-A35	1 000	6,3	7,6	
10x2	19,5	M17x1	22	22	7,5	126-A35	1 000	8,3	9,5	
12x2	20	M18x1	23,5	23,5	8	126-A35	2 000	9,9	11,2	
12x4	25,5	M20x1	34	34	10	126-A35	2 000	9,4	11,3	
12x5	23	M20x1	40	40	10	126-A35	2 000	9,3	11,8	
14x4	27	M22x1,5	34	34	8	126-A35	2 000	11,9	13,7	
16x2	29,5	M25x1,5	27	27	12	126-A35	2 000	14,3	15,5	FLBU 16/PLBU 16 ⁴⁾
16x5	32,5	M26x1,5	42	42	12	126-A35	2 000	12,7	15,2	FLBU 16/PLBU 16 ⁴⁾ BUF 16 ⁴⁾

⁴⁾ Support bearings with standard steel

SX/BX universal screws

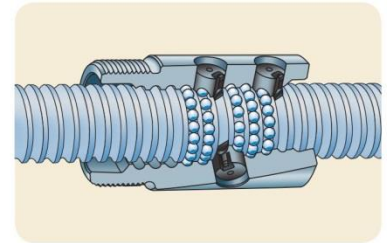
Rolled thread ball screw with recirculation through inserts, nut with threaded nose

Features

- Nominal diameter from 20 to 63 mm
- Lead from 5 to 40 mm
- Standard composite recirculation inserts
- Optional steel recirculation inserts
- Lubrication hole for grease nipple or for SKF SYSTEM 24 automatic lubrication kit
- Phosphate coating on nut
- Optional shaft surface coating
- Optional safety nuts. Please contact SKF for selection and usage of this option
- Optional nut flanges (→ pages 22 to 23)
- Optional wipers.



Standard



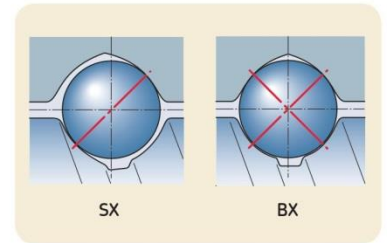
Recirculation

Benefits

- Minimum nut outside diameter and threaded nose for easy assembly
- Nut design well suited and economical for transport screw applications
- Optional steel recirculation inserts can act as a safety device for severe or vertical applications. Please contact SKF for such applications
- Backlash elimination by oversized balls on request (BX designation) for applications with vibrations / changes of direction, over maximum length of 1 000 mm.



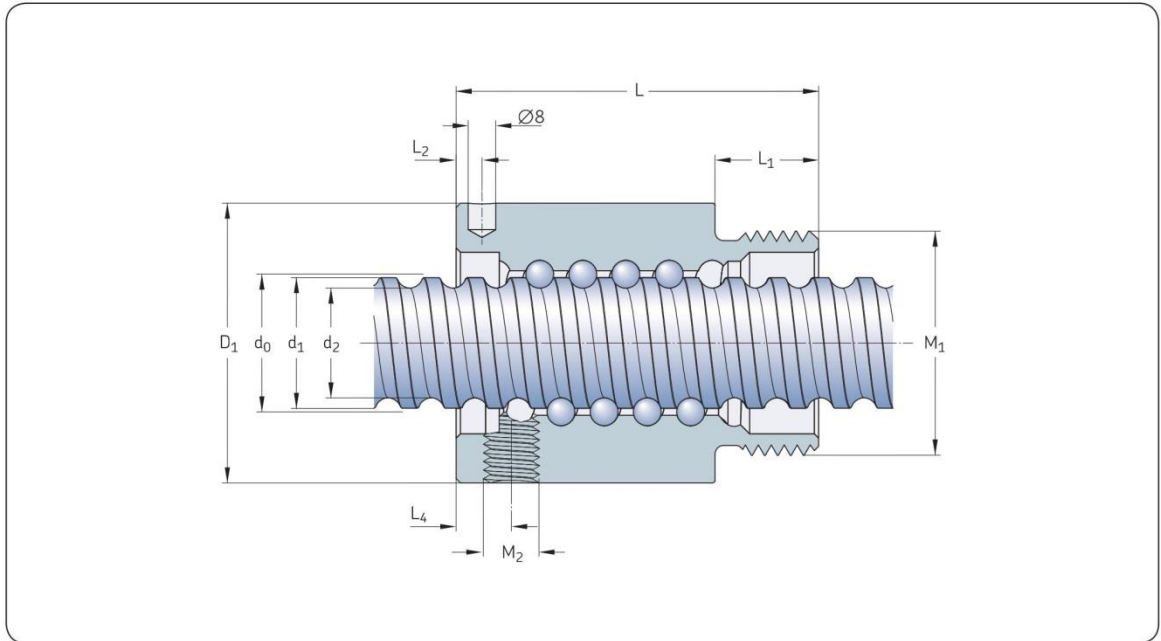
Customised



SX

BX

Nominal diameter	Lead (right hand)	Nut Basic load ratings		Number of circuits of balls	Std play	Reduced play on request	Preload torque zero play T_{pr}	Inertia	Grease	Weight	Screw			Designation
		dynamic	static								Mass	Inertia	Grease	
d_0	P_h	C_a	C_{oa}		mm		Nm	kgmm^2	cm^3	kg	kg/m	kgmm^2/m	cm^3/m	–
mm	mm	kN	–	–	mm		Nm	kgmm^2	cm^3	kg	kg/m	kgmm^2/m	cm^3/m	–
20	5	14	23,8	4	0,1	0,05	0,1	60	1,3	0,24	2	85	2,7	SX/BX 20×5 R
	10	19	37,8	5	0,1	0,05	0,17	125	2,5	0,39	3,3	224	3,4	SX/BX 25×5 R
25	5	23,5	39	4	0,12	0,08	0,23	135	4,6	0,4	3,2	255	3,2	SX/BX 25×10 R
	10	22	51,6	5	0,1	0,05	0,25	230	2,6	0,48	5,6	641	4,4	SX/BX 32×5 R
32	5	27,1	52	4	0,12	0,08	0,32	400	5,9	0,77	5,6	639	3,7	SX/BX 32×10 R
	10	24,3	65,6	5	0,1	0,05	0,34	390	3,3	0,58	9	1 639	5,6	SX/BX 40×5 R/L
40	5	61,5	124,1	5	0,12	0,08	0,64	840	12,4	1,25	8,4	1 437	5	SX/BX 40×10 R
	10	31,3	72,9	2×1,9	0,1	0,05	0,64	1 200	14,4	1,6	8,1	1 330	5,2	SX/BX 40×40 R
50	10	80,4	188,8	6	0,12	0,08	1,02	2 400	19,9	2,4	13,6	3 736	6,3	SX/BX 50×10 R
63	10	91,2	248,3	6	0,12	0,08	1,44	4 620	25,4	3,1	22	9 913	8,1	SX/BX 63×10 R



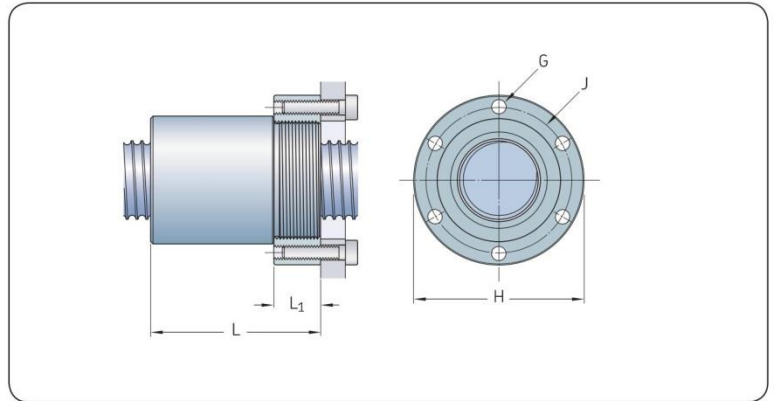
Screw $d_0 \times P_h$	Nut		L	L_1	L_2	L_4	M2 ¹⁾	Tightening spanner	Screw		Support bearing Recommended thrust support bearings	Recommended support pillow block	
	D_1 js13	M_1 6g							length max.	d_2			d_1
mm	mm	mm	mm	mm	mm	mm	mm	mm	mm	mm	mm	mm	
20×5	38	M35×1,5	54	14	8	8	M6×1	HN5	4 700	16,7	19,4	PLBU 20/FLBU 20 ²⁾	BUF 20
25×5	43	M40×1,5	69	19	8	8	M6×1	HN6	4 700	21,7	24,6	PLBU 25/FLBU 25	BUF 25
25×10	43	M40×1,5	84	19	12	12	M6×1	HN6	4 700	20,5	24,6	PLBU 25/FLBU 25	BUF 25
32×5	52	M48×1,5	64	19	8	8	M6×1	HN7	5 700	28,7	31,6	PLBU 32/FLBU 32	BUF 32
32×10	54	M48×1,5	95	19	15	15	M6×1	HN7	5 700	27,8	32	PLBU 32/FLBU 32/FLRBU 3 ³⁾	BUF 32
40×5	60	M56×1,5	65	19	8	8	M6×1	HN9	5 700	36,7	39,6	PLBU 40/FLBU 40	BUF 40
40×10	65	M60×2	105	24	15	13	M8×1	HN9	5 700	34	39,4	PLBU 40/FLBU 40/FLRBU 4 ³⁾	BUF 40
40×40	65	M60×2	121	24	20	48,6	M8×1	HN9	5 700	34,2	38,3	PLBU 40/FLBU 40	BUF 40
50×10	78	M72×2	135	29	15	15	M8×1	HN12	5 700	44	49,7	PLBU 50/FLBU 50/FLRBU 5 ³⁾	BUF 50
63×10	93	M85×2	135	29	15	15	M8×1	HN14	5 700	57	62,8	PLBU 63/FLBU 63	BUF 63

¹⁾ Threaded lubrication hole M2 indexed to ISO thread M₁
²⁾ For high load application, please contact SKF
³⁾ For high load application, use FLRBU type. Please refer to roller screws catalogue for end shaft and support bearings definitions

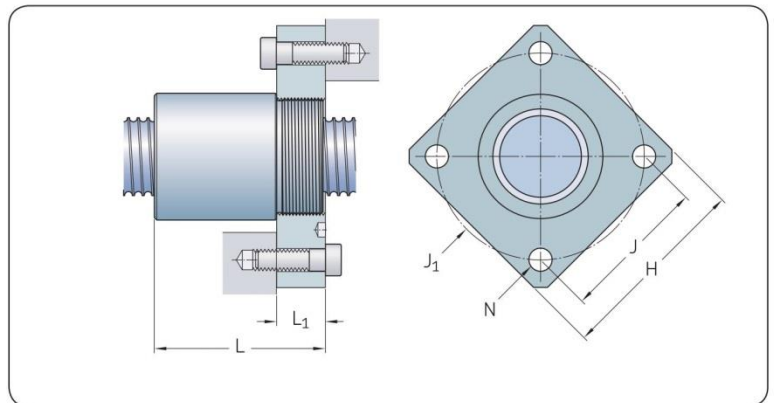
Dedicated flanges for SX/BX nuts



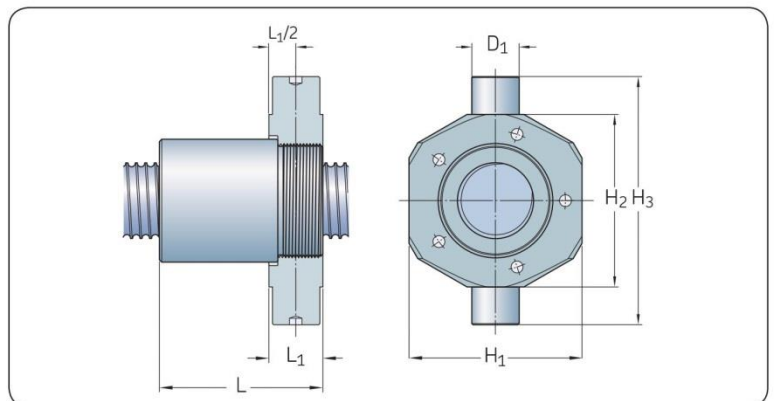
SX nut with round flange (FHRF)



SX nut with square flange (FHSF)



SX nut with trunnions flange (FHTF)



Nominal diameter d_0	Lead P_h	Dimensions					Designation
		L	L_1 h14	G	H h12	J js12	
mm	mm	mm					–
20	5	55	15	M5	52	44	FHRF 20
25	5	70	20	M6	60	50	FHRF 25
	10	85	20	M6	60	50	FHRF 25
32	5	65	20	M6	69	59	FHRF 32
	10	96	20	M6	69	59	FHRF 32
40	5	66	20	M8	82	69	FHRF 40x5
	10	106	25	M10	92	76	FHRF 40x10
	40	122	25	M10	92	76	FHRF 40x10
50	10	136	30	M12	110	91	FHRF 50
63	10	136	30	M12	125	106	FHRF 63

Nominal diameter d_0	Lead P_h	Dimensions					Designation	
		L	L_1 h14	H h14	J js12	J_1		N
mm	mm	mm					–	
20	5	55	15	60	45	63,6	6,6	FHSF 20
25	5	70	20	70	52	73,5	9	FHSF 25
	10	85	20	70	52	73,5	9	FHSF 25
32	5	65	20	80	60	84,8	9	FHSF 32
	10	96	20	80	60	84,8	9	FHSF 32
40	5	66	20	90	70	99	11	FHSF 40x5
	10	106	25	100	78	110,3	13	FHSF 40x10
	40	122	25	100	78	110,3	13	FHSF 40x10
50	10	136	30	120	94	133	15	FHSF 50
63	10	136	30	130	104	147	15	FHSF 63

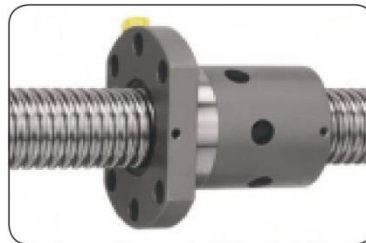
Nominal diameter d_0	Lead P_h	Dimensions					Designation	Glycodur designation GLY PG	
		L	L_1	H_1 js16	H_2 h12	H_3 h12			D_1 h8
mm	mm	mm					–		
20	5	57	17	55	56	80	15	FHTF 20	151710A
25	5	71	21	60	65	97	18	FHTF 25	182015A
	10	86	21	60	65	97	18	FHTF 25	182015A
32	5	68	23	73	73	105	20	FHTF 32	202315A
	10	99	23	73	73	105	20	FHTF 32	202315A
40	5	69	23	85	85	117	20	FHTF 40x5	202315A
	10	108,5	27,5	98	98	140	25	FHTF 40x10	252820A
	40	124,5	27,5	98	98	140	25	FHTF 40x10	252820A
50	10	139	33	120	120	162	30	FHTF 50	303420A
63	10	139	33	135	135	177	30	FHTF 63	303420A

SND/BND precision screws, DIN standard 69051

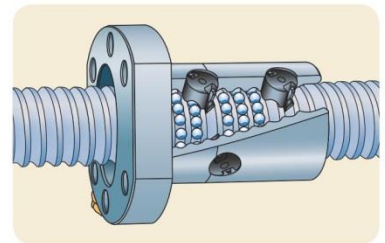
Rolled thread ball screw with recirculation through inserts, DIN nut

Features

- Nominal diameter from 16 to 63 mm
- Lead from 5 to 10 mm
- Standard composite recirculation inserts
- Optional steel recirculation inserts
- Standard lead precision G5, G7 and G9
- Nut ground outside diameter / flange face
- Precision ground nut thread¹⁾
- Lubrication hole for grease nipple or for SKF SYSTEM 24 automatic lubrication kit
- Optional surface coating on shaft and nut
- Optional safety nuts. Please contact SKF for selection and usage of this option
- Optional wipers.



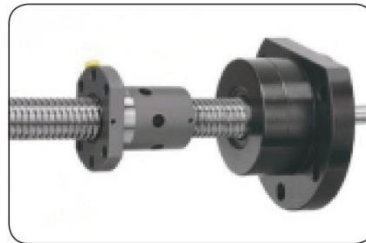
Standard



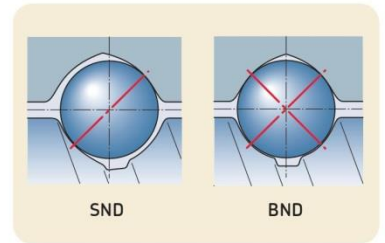
Recirculation

Benefits

- Compact nut / integral flange for easy assembly
- Design well suited for positioning screws. G5 lead precision of ground ball screws
- Optional steel recirculation inserts can act as a safety device for severe or vertical applications. Please contact SKF for such applications
- Backlash elimination by oversized balls on request (BND designation), over maximum length of 1 000 mm.



Assembly with flanged support bearing

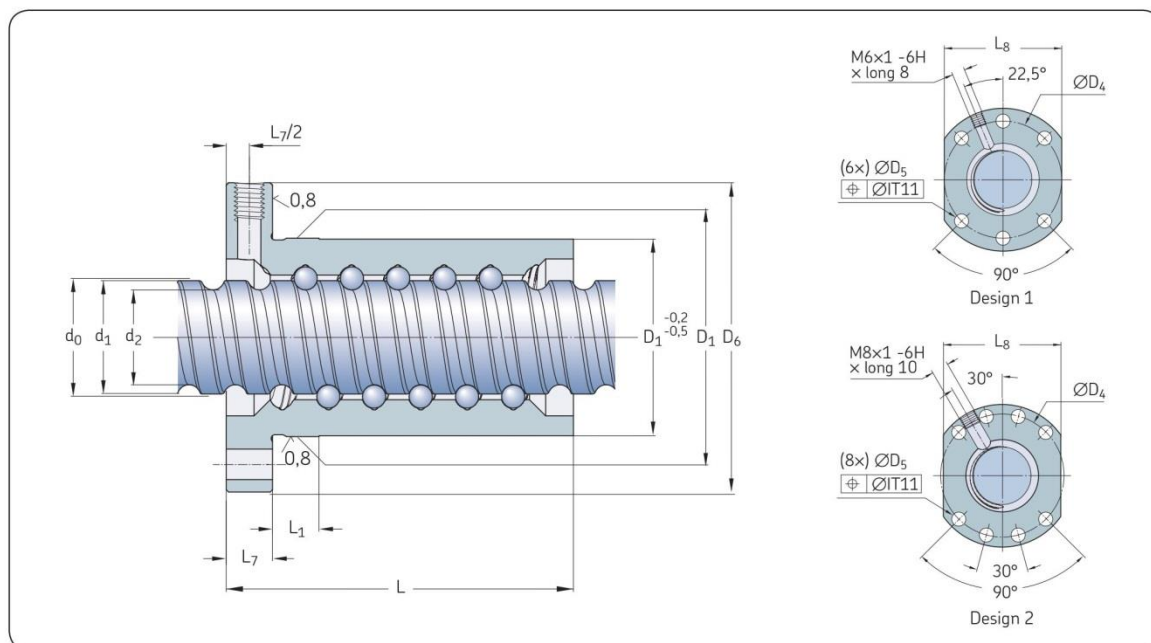


SND

BND

Nominal diameter d_0	Lead (right hand) P_h	Nut Basic load ratings		Number of circuits of balls	Std play	Reduced play on request	Preload torque zero play T_{pr}	Inertia	Grease	Weight	Screw		Grease	Designation
		dynamic C_a	static C_{oa}								Mass	Inertia		
mm	mm	kN		–	mm		Nm	kgmm ²	cm ³	kg	kg/m	kgmm ² /m	cm ³ /m	–
16	5	7,8	10,7	3	0,08	0,05	0,05	40	0,9	0,17	1,3	33	2,1	SND/BND 16×5 R
	10	10,7	17,2	2×1,8	0,07	0,03	0,06	41	1,6	0,18	1,21	30,7	2,1	SND/BND 16×10 R
20	5	11,3	17,9	3	0,1	0,05	0,08	86	1,1	0,24	2	85	2,7	SND/BND 20×5 R
25	5	12,7	22,7	3	0,1	0,05	0,11	117	1,6	0,29	3,3	224	3,4	SND/BND 25×5 R
	10	24,1	39	4	0,12	0,08	0,23	144	4,5	0,38	3,2	255	3,2	SND/BND 25×10 R
32	5	19	41,3	4	0,1	0,05	0,21	364	2,1	0,54	5,6	641	4,5	SND/BND 32×5 R
	10	21,9	39	3	0,12	0,08	0,25	384	4,6	0,58	5,6	639	4,2	SND/BND 32×10 R
40	5	25,6	65,6	5	0,1	0,05	0,36	855	3,1	0,92	9	1 639	5,6	SND/BND 40×5 R/L
	10	63,3	124,1	5	0,12	0,08	0,64	1 010	10,7	1,3	8,4	1 437	5,1	SND/BND 40×10 R
50	10	71,3	157,3	5	0,12	0,08	0,88	2 130	13,1	1,8	13,6	3 736	6,5	SND/BND 50×10 R
63	10	81,5	206,9	5	0,12	0,08	1,23	4 075	16,1	2,4	22	9 913	8,4	SND/BND 63×10 R

¹⁾ Except 16×10 R: nut thread is not ground



Screw	Nut		Design	D ₅ H13	D ₆ h13	L	L ₁	L ₇	L ₈ h13	Screw length max.	d ₂	d ₁	Support bearing Recommended thrust support bearings	Recommended support pillow block
	d ₀ × P _h	D ₁ g6												
mm	mm	mm	-	mm	mm	mm	mm	mm	mm	mm	mm	mm	-	-
16×5	28	38	1	5,5	48	43,5	10	10	40	2 000	12,7	15,2	FLBU 16/PLBU 16	BUF 16
16×10	28	38	1	5,5	48	47	37	10	40	2 000	12,6	15,2	FLBU 16/PLBU 16	BUF 16
20×5	36	47	1	6,6	58	44,5	10	10	44	4 700	16,7	19,4	PLBU 20/FLBU 20	BUF 20
25×5	40	51	1	6,6	62	44,5	10	10	48	4 700	21,7	24,6	PLBU 25/FLBU 25	BUF 25
25×10	40	51	1	6,6	62	75	10	10	48	4 700	20,5	24,6	PLBU 25/FLBU 25	BUF 25
32×5	50	65	1	9	80	51,5	10	12	62	5 700	28,7	31,6	PLBU 32/FLBU 32	BUF 32
32×10	50	65	1	9	80	64	10	12	62	5 700	27,8	32	PLBU 32/FLBU 32	BUF 32
40×5	63	78	2	9	93	58,5	10	14	70	5 700	36,7	39,6	PLBU 40/FLBU 40	BUF 40
40×10	63	78	2	9	93	91	20	14	70	5 700	34	39,4	PLBU 40/FLBU 40/FLRBU 4 ²⁾	BUF 40
50×10	75	93	2	11	110	93	10	16	85	5 700	44	49,7	PLBU 50/FLBU 50/FLRBU 5 ²⁾	BUF 50
63×10	90	108	2	11	125	95	10	18	95	5 700	57	62,8	PLBU 63/FLBU 63	BUF 63

²⁾ For high load application, use FLRBU type. Please refer to roller screws catalogue for end shaft and support bearings definitions

PND preloaded screws, DIN standard 69051

Rolled thread ball screw with recirculation through inserts, DIN nut

Features

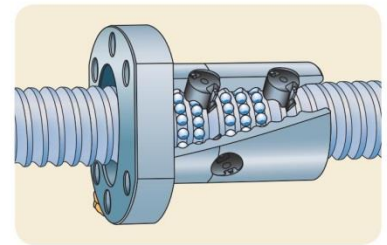
- Nominal diameter from 16 to 63 mm
- Lead from 5 to 10 mm
- Standard composite recirculation inserts
- Optional steel recirculation inserts
- Standard lead precision G5, G7 and G9
- Nut ground outside diameter / flange face
- Precision ground nut thread¹⁾
- Standard preload 7% to 8,5% of ball screw C_a value, depending on ball screw size
- Lubrication hole for grease nipple or for SKF SYSTEM 24 automatic lubrication kit
- Optional surface coating on shaft and nut
- Optional safety nuts. Please contact SKF for selection and usage of this option
- Optional wipers.

Benefits

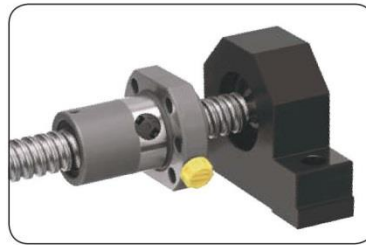
- Compact nut / integral flange for easy assembly
- One-piece nut¹⁾ with internal preload for compactness and optimum rigidity
- Design well suited for positioning screws. G5 lead precision of ground ball screws
- Optional steel recirculation inserts can act as a safety device for severe or vertical applications. Please contact SKF for such applications.



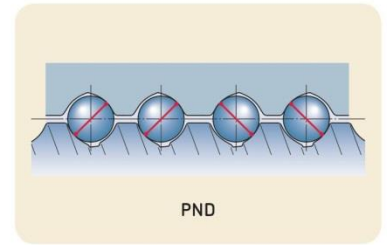
Standard



Recirculation



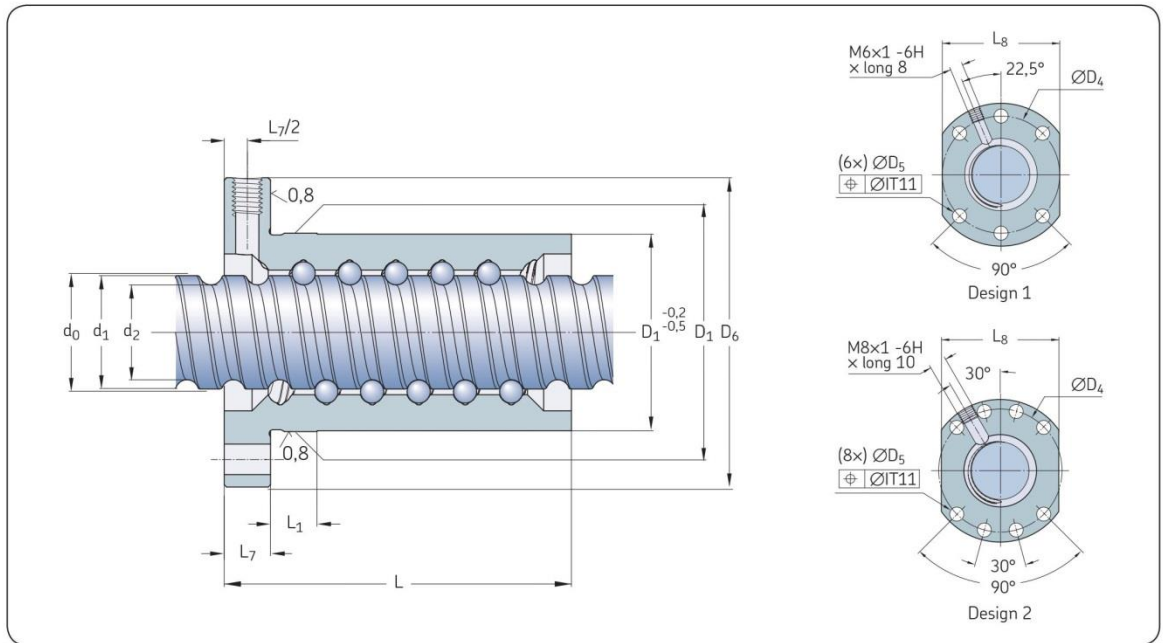
Assembly with pillow block



PND

Nominal diameter	Lead (right hand)	Nut		Number of circuits of balls	Preload torque average T_{pr}	Stiffness R_n	Inertia	Grease	Weight	Screw			Designation
		Basic load ratings dynamic	static							Mass	Inertia	Grease	
d_0	P_h	C_a	C_{oa}							kg/m	kgmm ² /m	cm ³ /m	
mm	mm	kN			Nm	N/μm	kgmm ²	cm ³	kg				
16	5	5,5	7,1	2×2	0,08	147	46	1	0,19	1,3	33	2,1	PND 16×5 R
	10	10,7	17,2	2×2×1,8	0,15	263	56	2,7	0,28	1,21	30,7	1,9	PND 16×10 R ¹⁾
20	5	8	11,9	2×2	0,14	248	91	1,3	0,26	2	85	2,7	PND 20×5 R
25	5	12,7	22,7	2×3	0,28	436	405	2	0,4	3,3	224	3,4	PND 25×5 R
	10	13,3	19,5	2×2	0,3	264	245	4,5	0,53	3,2	255	3,2	PND 25×10 R
32	5	19	41,3	2×4	0,52	734	453	3,2	0,715	5,6	641	3,2	PND 32×5 R
	10	21,9	39	2×3	0,61	490	490	7,6	0,81	5,6	639	4,1	PND 32×10 R
40	5	25,6	65,6	2×5	0,71	968	1 110	4,8	1,3	9	1 639	5,5	PND 40×5 R/L
	10	52,2	99,3	2×4	1,47	793	1 290	15,5	1,8	8,4	1 437	4,9	PND 40×10 R
50	10	71,3	157,3	2×5	2,47	1 222	2 940	27,5	2,6	13,6	3 736	7,9	PND 50×10 R
63	10	81,5	206,9	2×5	3,46	1 448	5 290	26,8	3,2	22	9 913	7,9	PND 63×10 R

¹⁾ Except 16×10 R: nut thread is not ground, double nut design



Screw	Nut									Screw	Support bearing			Recommended
$d_0 \times P_h$	D_1 g6	D_4 js12	Design	D_5 H13	D_6 h13	L	L_1	L_7	L_8 h13	length max.	d_2	d_1	Recommended thrust support bearings	support pillow block
mm	mm									mm			-	
16x5	28	38	1	5,5	48	48	10	10	40	2 000	12,7	15,2	FLBU 16/PLBU 16	BUF 16
16x10	28	38	1	5,5	48	87	77	10	40	2 000	12,6	15,2	FLBU 16/PLBU 16	BUF 16
20x5	36	47	1	6,6	58	50	10	10	44	4 700	16,7	19,4	PLBU 20/FLBU 20	BUF 20
25x5	40	51	1	6,6	62	62	10	10	48	4 700	21,7	24,6	PLBU 25/FLBU 25	BUF 25
25x10	40	51	1	6,6	62	75	10	10	48	4 700	20,5	24,6	PLBU 25/FLBU 25	BUF 25
32x5	50	65	1	9	80	74	10	12	62	5 700	28,7	31,6	PLBU 32/FLBU 32	BUF 32
32x10	50	65	1	9	80	100	10	12	62	5 700	27,8	32	PLBU 32/FLBU 32	BUF 32
40x5	63	78	2	9	93	88	10	14	70	5 700	36,7	39,6	PLBU 40/FLBU 40	BUF 40
40x10	63	78	2	9	93	130	20	14	70	5 700	34	39,4	PLBU 40/FLBU 40/FLRBU 4 ²⁾	BUF 40
50x10	75	93	2	11	110	151	10	16	85	5 700	44	49,7	PLBU 50/FLBU 50/FLRBU 5 ²⁾	BUF 50
63x10	90	108	2	11	125	153	10	18	95	5 700	57	62,8	PLBU 63/FLBU 63	BUF 63

²⁾ For high load application, use FLRBU type. Please refer to roller screws catalogue for end shaft and support bearings definitions

SN/BN precision screws

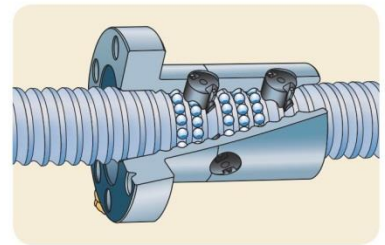
Rolled thread ball screw with recirculation through inserts, cylindrical flange

Features

- Nominal diameter from 16 to 63 mm
- Lead from 5 to 10 mm
- Standard composite recirculation inserts
- Optional steel recirculation inserts
- Standard lead precision G5, G7 and G9
- Nut ground outside diameter / flange face
- Precision ground nut thread
- Lubrication hole for grease nipple or for SKF SYSTEM 24 automatic lubrication kit
- Optional surface coating on shaft and nut
- Optional safety nuts. Please contact SKF for selection and usage of this option
- Optional wipers.



Standard



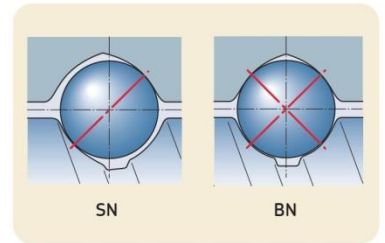
Recirculation

Benefits

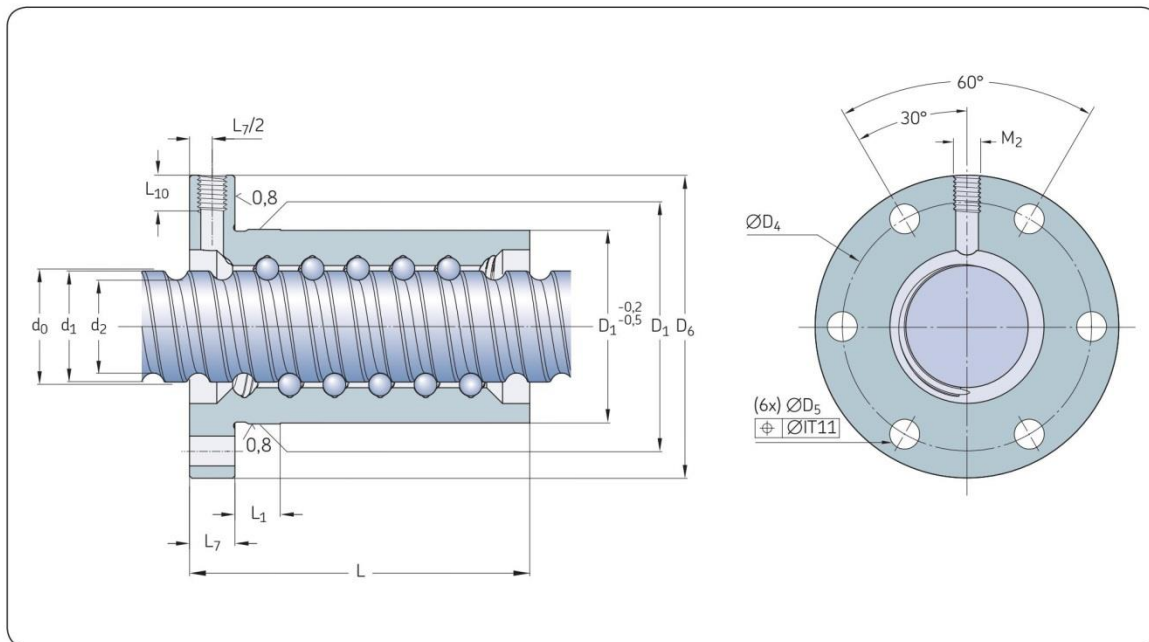
- Economical compact nut / integral flange for easy assembly
- Design well suited for positioning screws. G5 lead precision of ground ball screws
- Optional steel recirculation inserts can act as a safety device for severe or vertical applications. Please contact SKF for such applications
- Backlash elimination by oversized balls on request (BN designation), over maximum length of 1 000 mm.



Customised



Nominal diameter	Lead (right hand)	Nut		Number of circuits of balls	Std play	Reduced play on request	Preload torque zero play T_{pr}	Inertia	Grease	Weight	Screw			Designation
		Basic load dynamic	static								Mass	Inertia	Grease	
d_0	P_h	C_a	C_{oa}								kg/m	kgmm ² /m	cm ³ /m	-
mm	mm	kN		-	mm		Nm	kgmm ²	cm ³	kg				
16	5	7,8	10,7	3	0,08	0,05	0,05	45	0,9	0,18	1,3	33	2,1	SN/BN 16x5 R
20	5	11,3	17,9	3	0,1	0,05	0,08	88	1,2	0,24	2	85	2,7	SN/BN 20x5 R
25	5	12,7	22,7	3	0,1	0,05	0,11	127	1,6	0,28	3,3	224	3,4	SN/BN 25x5 R
	10	24,1	39	4	0,12	0,08	0,23	244	4,5	0,53	3,2	255	3,2	SN/BN 25x10 R
32	5	19	41,3	4	0,1	0,05	0,21	250	2,1	0,4	5,6	641	4,5	SN/BN 32x5 R
	10	21,9	39	3	0,12	0,08	0,25	673	4,6	0,83	5,6	639	4,2	SN/BN 32x10 R
40	5	25,6	65,6	5	0,1	0,05	0,36	495	3,1	0,58	9	1 639	5,6	SN/BN 40x5 R/L
	10	63,3	124,1	5	0,12	0,08	0,64	1 285	10,7	1,4	8,4	1437	5,1	SN/BN 40x10 R
50	10	71,3	157,3	5	0,12	0,08	0,88	1 305	13,1	1,8	13,6	3 736	6,5	SN/BN 50x10 R
63	10	81,5	206,9	5	0,12	0,08	1,23	4 180	16,1	2,25	22	9 913	8,4	SN/BN 63x10 R



Screw	Nut									Screw	Support bearing Recommended thrust support bearings		Recommended support pillow block	
$d_0 \times P_h$	D_1 g6	D_4	D_5 H13	D_6 h13	L	L_1	L_7	L_{10}	M_2 6H	length max.	d_2	d_1		
mm	mm									mm	-			
16×5	28	38	6×5.5	48	43,5	10	10	8	M6	2000	12,7	15,2	FLBU 16 / PLBU 16	BUF 16
20×5	33	45	6×6.6	57	44,5	10	10	8	M6	4700	16,7	19,4	PLBU 20 / FLBU 20	BUF 20
25×5	38	50	6×6.6	62	44,5	10	10	8	M6	4700	21,7	24,6	PLBU 25 / FLBU 25	BUF 25
25×10	43	55	6×6.6	67	75	10	10	8	M6	4700	20,5	24,6	PLBU 25 / FLBU 25	BUF 25
32×5	45	58	6×6.6	70	51,5	10	12	8	M6	5700	28,7	31,6	PLBU 32 / FLBU 32	BUF 32
32×10	54	70	6×9	87	64	10	12	10	M8×1	5700	27,8	32	PLBU 32 / FLBU 32	BUF 32
40×5	53	68	6×6.6	80	58,5	10	14	8	M6	5700	36,7	39,6	PLBU 40 / FLBU 40	BUF 40
40×10	63	78	6×9	95	91	20	14	10	M8×1	5700	34	39,4	PLBU 40 / FLBU 40 / FLRBU 4 ¹⁾	BUF 40
50×10	72	90	6×11	110	99	10	16	10	M8×1	5700	44	49,7	PLBU 50 / FLBU 50 / FLRBU 5 ¹⁾	BUF 50
63×10	85	105	6×11	125	101	10	18	10	M8×1	5700	57	62,8	PLBU 63 / FLBU 63	BUF 63

¹⁾ For high load application, use FLRBU type. Please refer to roller screws catalogue for end shaft and support bearings definitions

PN preloaded screws

Rolled thread ball screw with recirculation through inserts, cylindrical flange

Features

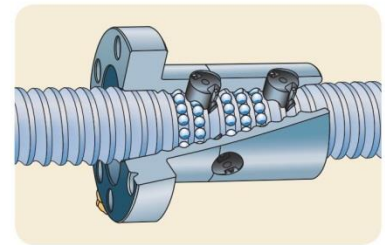
- Nominal diameter from 16 to 63 mm
- Lead from 5 to 10 mm
- Standard composite recirculation inserts
- Optional steel recirculation inserts
- Standard lead precision G5, G7 and G9
- Nut ground outside diameter / flange face
- Precision ground nut thread
- Standard preload 7% to 8,5% of ball screw C_a value, depending on ball screw size
- Lubrication hole for grease nipple or for SKF SYSTEM 24 automatic lubrication kit
- Optional surface coating on shaft and nut
- Optional safety nuts. Please contact SKF for selection and usage of this option
- Optional wipers.

Benefits

- Economical compact nut / integral flange for easy assembly
- One-piece nut with internal preload for compactness and optimum rigidity
- Design well suited for positioning screws. G5 lead precision of ground ball screws
- Optional steel recirculation inserts can act as a safety device for severe or vertical applications. Please contact SKF for such applications.



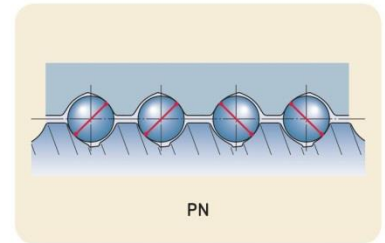
Standard



Recirculation

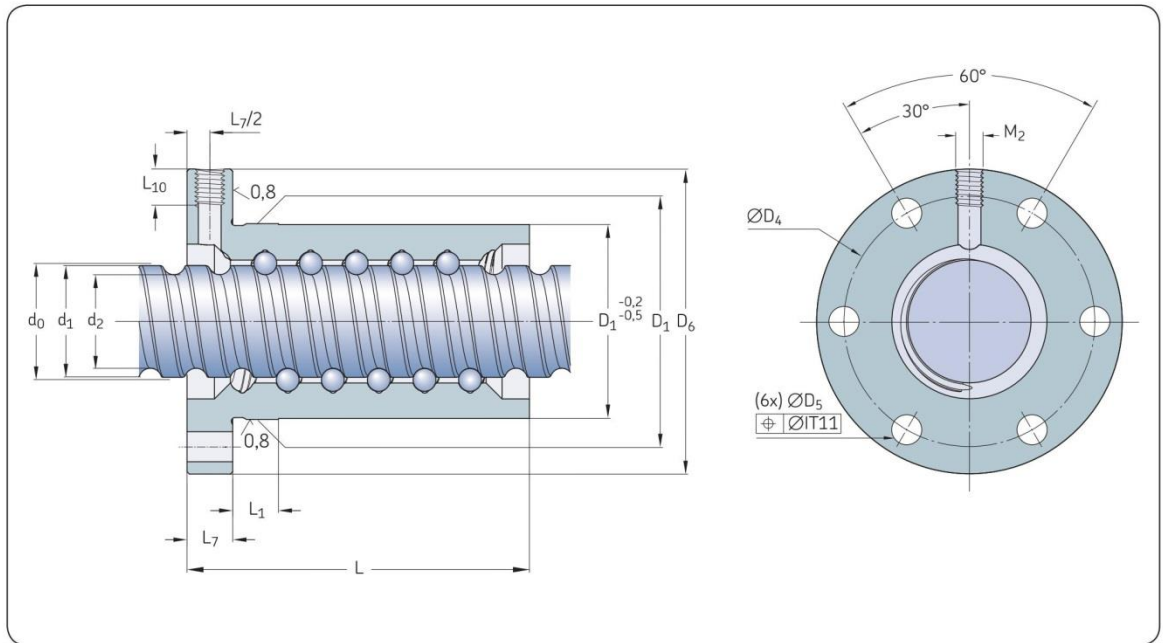


Customised



PN

Nominal diameter	Lead (right hand)	Nut Basic load ratings		Number of circuits of balls	Preload torque average T_{pr}	Stiffness R_n	Inertia	Grease	Weight	Screw			Designation
		dynamic	static							Mass	Inertia	Grease	
d_0	P_h	C_a	C_{oa}							kg/m	kgmm ² /m	cm ³ /m	
mm	mm	kN		–	Nm	N/μm	kgmm ²	cm ³	kg		kgmm ² /m	cm ³ /m	–
16	5	5,5	7,1	2×2	0,08	147	46	1	0,19	1,3	33	2,1	PN 16×5 R
20	5	8	11,9	2×2	0,14	248	91	1,1	0,26	2	85	2,4	PN 20×5 R
25	5	17,7	22,7	2×3	0,28	436	400	2,1	0,39	3,3	224	3,4	PN 25×5 R
	10	13,3	19,5	2×2	0,3	264	245	4,1	0,53	3,2	255	2,8	PN 25×10 R
32	5	19	41,3	2×4	0,52	734	390	3,2	0,5	5,6	641	4,4	PN 32×5 R
	10	21,9	39	2×3	0,61	490	830	7,6	1,13	5,6	639	4,1	PN 32×10 R
40	5	25,6	65,6	2×5	0,71	968	585	4,8	0,74	9	1 639	5,5	PN 40×5 R/L
	10	52,2	99,3	2×4	1,47	793	1 530	14,6	1,8	8,4	1 437	4,9	PN 40×10 R
50	10	71,3	157,3	2×5	2,47	1 222	2 930	27,5	2,6	13,6	3 736	7,9	PN 50×10 R
63	10	81,5	206,9	2×5	3,46	1 448	5 980	26,8	3,2	22	9 913	7,9	PN 63×10 R



Screw	Nut										Screw			Support bearing Recommended thrust support bearings	Recommended support pillow block
$d_0 \times P_h$	D_1 g6	D_4 js12	D_5 H13	D_6 h13	L	L_1	L_7	L_{10}	M_2 6H	length max.	d_2	d_1			
mm	mm										mm			-	
16×5	28	38	6×5,5	48	48	10	10	8	M6	2 000	12,7	15,2	FLBU 16/PLBU 16	BUF 16	
20×5	33	45	6×6,6	57	50	10	10	8	M6	4 700	16,7	19,4	PLBU 20/FLBU 20	BUF 20	
25×5	38	50	6×6,6	62	62	10	10	8	M6	4 700	21,7	24,6	PLBU 25/FLBU 25	BUF 25	
25×10	43	55	6×6,6	67	75	10	10	8	M6	4 700	20,5	24,6	PLBU 25/FLBU 25	BUF 25	
32×5	45	58	6×6,6	70	74	10	12	8	M6	5 700	28,7	31,6	PLBU 32/FLBU 32	BUF 32	
32×10	54	70	6×9	87	100	10	12	10	M8×1	5 700	27,8	32	PLBU 32/FLBU 32	BUF 32	
40×5	53	68	6×6,6	80	88	10	14	8	M6	5 700	36,7	39,6	PLBU 40/FLBU 40	BUF 40	
40×10	63	78	6×9	95	126	20	14	10	M8×1	5 700	34	39,4	PLBU 40/FLBU 40/FLRBU 4 ¹⁾	BUF 40	
50×10	72	90	6×11	110	151	10	16	10	M8×1	5 700	44	49,7	PLBU 50/FLBU 50/FLRBU 5 ¹⁾	BUF 50	
63×10	85	105	6×11	125	153	10	18	10	M8×1	5 700	57	62,8	PLBU 63/FLBU 63	BUF 63	

¹⁾ For high load application, use FLRBU type. Please refer to roller screws catalogue for end shaft and support bearings definitions

SL/TL long lead screws

Rolled thread ball screw for high linear speed

Features

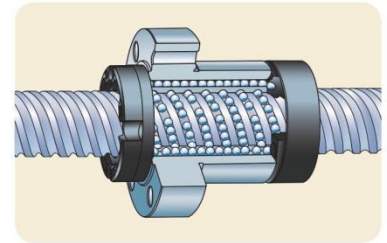
- Nominal diameter from 25 to 50 mm
- Lead from 20 to 50 mm
- Lubrication hole for grease nipple or for SKF SYSTEM 24 automatic lubrication kit
- Standard protection at each end of the nut with composite wipers integrated into recirculation caps (NOWPR)
- Optional double protection at each end of the nut with additional brush wipers fitted into recirculation caps (WPR)
- Optional surface coating on shaft and nut
- Optional safety nuts. Please contact SKF for selection and usage of this option.

Benefits

- High rotational speed up to $nd_0 = 90\,000$, resulting in high linear speed up to 110 m/min
- Nut design well suited for transport and positioning screw applications requiring high velocity such as woodworking, some functions in plastic injection presses, pick-&-place, etc.
- Backlash elimination (TL designation).



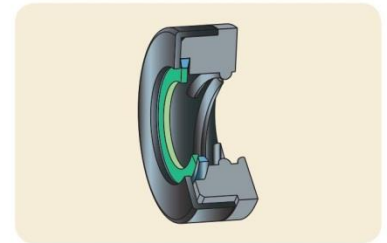
Standard



Recirculation

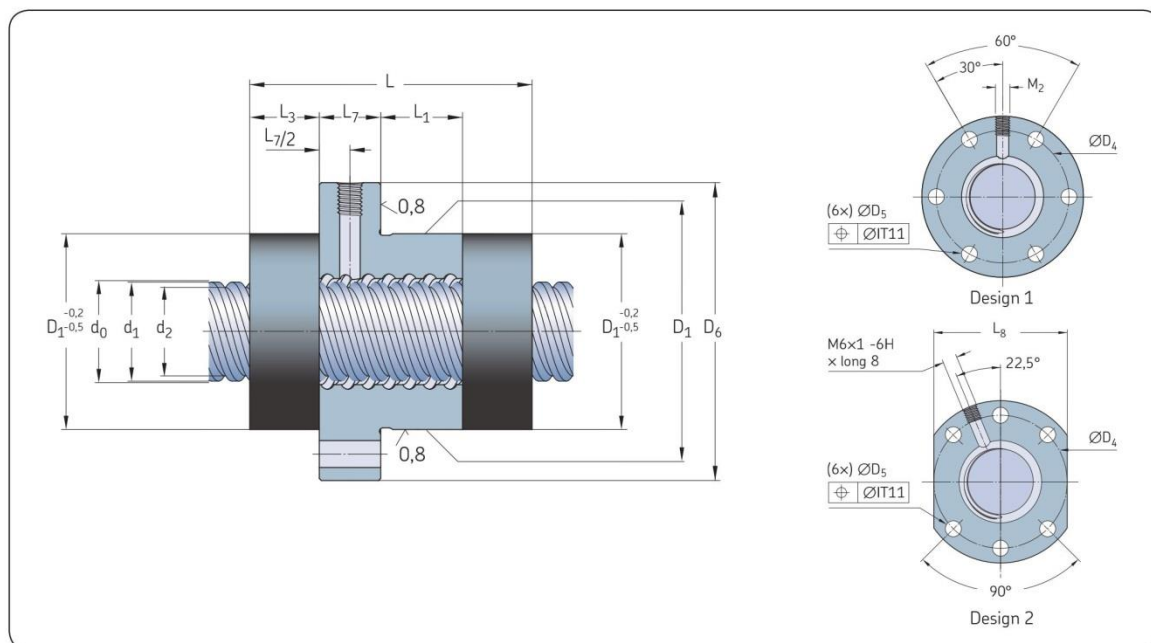


Customised



Optional double protection

Nominal diameter	Lead (right hand)	Nut			TL (with backlash elimination)			Number of circuits of balls	Inertia	Grease	Weight	Screw			Designation
		SL (with play)		Std play	Basic load ratings		Preload torque zero play					Mass	Inertia	Grease	
d_0	P_h	C_a	C_{oa}		C_a	C_{oa}	T_{pr}								
mm	mm	kN		mm	kN		Nm	–	kgmm ²	cm ³	kg	kg/m	kgmm ² /m	cm ³ /m	–
25	20	22,8	51,5	0,08	12,6	25,8	0,04-0,36	4×1,7	480	3	0,57	3,3	215	3,4	SL/TL 25×20 R
	25	22,3	50,6	0,08	12,3	25,3	0,04-0,36	4×1,7	400	3,6	0,66	3,2	210	3,3	SL/TL 25×25 R
32	20	25,4	65,2	0,08	14	32,6	0,05-0,45	4×1,7	550	3,4	0,7	5,1	530	4,4	SL/TL 32×20 R
	32	26,1	69,3	0,08	14,4	34,7	0,05-0,50	4×1,8	450	4,5	0,7	5,4	600	4,3	SL/TL 32×32 R
	40	12,6	29,8	0,08	6,9	14,9	0,05-0,50	4×0,8	515	3	0,65	4,9	490	4,4	SLD/TLD 32×32 R SL/TL 32×40 R
40	20	41,3	128,8	0,08	22,8	64,4	0,05-0,55	4×2,7	1 420	6,6	1,2	8,2	1 380	5,5	SL/TL 40×20 R
	40	51,7	130,5	0,1	28,5	65,3	0,05-0,55	4×1,7	3 300	12,5	2,4	8,1	1 330	5,2	SL/TL 40×40 R
50	50	92,9	235,1	0,12	51,2	117,6	0,1-0,9	4×1,7	6 060	19,4	3,3	13,2	3 560	6,4	SL/TL 50×50 R



Screw	Nut												Screw		Support bearing Recommended thrust support bearings	Recommended support pillow block	
$d_0 \times P_h$	D_1 g9	D_4 js12	Design	D_5 H13	D_6	L	L_1	L_3	L_7	L_8 h13	L_{10}	M_2	length max.	d_2	d_1		
mm	mm	-		mm									mm				
25x20	48	60	1	6x6,6	73	66,8	18	17,6	15	N/A	8	M6	4 700	21,7	24,3	PLBU 25/FLBU 25	BUF 25
25x25	48	60	1	6x6,6	73	78,2	27	18,7	15	N/A	8	M6	4 700	21,5	24,4	PLBU 25/FLBU 25	BUF 25
32x20	56	68	1	6x6,6	80	67,4	18	17,9	15	N/A	8	M6	5 700	27,5	30	PLBU 32/FLBU 32/FLRBU3 ¹⁾	BUF 32
32x32	56	68	1	6x6,6	80	80,3	41	13	15	N/A	8	M6	5 700	28,4	31,1	PLBU 32/FLBU 32/FLRBU3 ¹⁾	BUF 32
32x32	50	g6	65	6x9	80	80,3	41	13	15	62	8	M6	5 700	28,4	31,1	PLBU 32/FLBU 32/FLRBU3 ¹⁾	BUF 32
32x40	53	g6	68	6x6,6	80	54,8	17	12,2	15	N/A	8	M6	5 700	26,9	29,6	PLBU 32/FLBU 32	BUF 32
40x20	63	78	1	6x9	95	87,3	38	18	15	N/A	8	M6	5 700	35,2	37,7	PLBU 40/FLBU 40	BUF 40
40x40	72	90	1	6x11	110	110,8	44	21,6	25	N/A	10	M8x1	5 700	34,2	38,3	PLBU 40/FLBU 40/FLRBU 4 ¹⁾	BUF 40
50x50	85	105	1	6x11	125	134	60	25,5	25	N/A	10	M8x1	5 700	43,5	49,1	PLBU 50/FLBU 50/FLRBU 5 ¹⁾	BUF 50

¹⁾ For high load application, use FLRBU type. Please refer to roller screws catalogue for end shaft and support bearings definitions

SLT/TLT rotating nut

Long lead rolled ball screw with rotating nut

Concept

The main purpose of this solution is to minimize the inertia phenomenon associated with long rotating shafts.

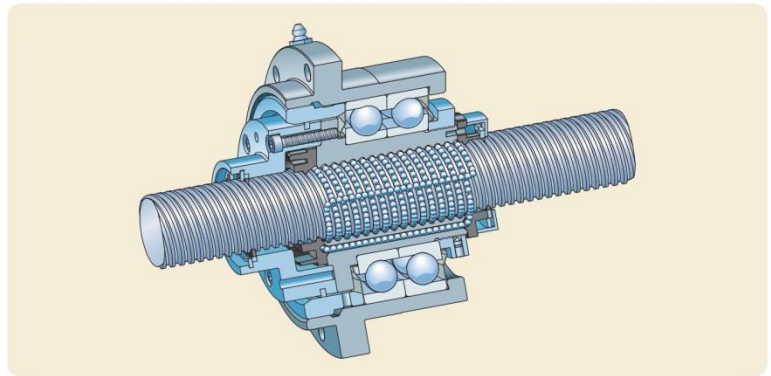
The long lead screw shaft is rigidly fixed to the machine frame. The ball nut, rotating inside a bearing housing and driven via a tension belt, moves along the screw shaft.

The customers are responsible for the sourcing and assembly of the electric motor, belt, pulleys and frame holding the bearing housing.



Features

- Nominal diameter from 25 to 50 mm
- Lead from 20 to 50 mm
- 72 series angular contact bearings are directly mounted on the nut outer diameter
- Bearings are preloaded in back-to-back arrangement in order to fully support the moment created by the belt tension
- 2 Nilos rings protect the bearings against pollution and permit lubrication for life
- Brush wipers are mounted at each end of the nut in the standard configuration for better protection against contamination
- The ball screw assembly is lubricated through a nipple mounted on the housing external diameter in the standard version
- Standard grease is SKF LGMT2. Other lubricants are available on request.

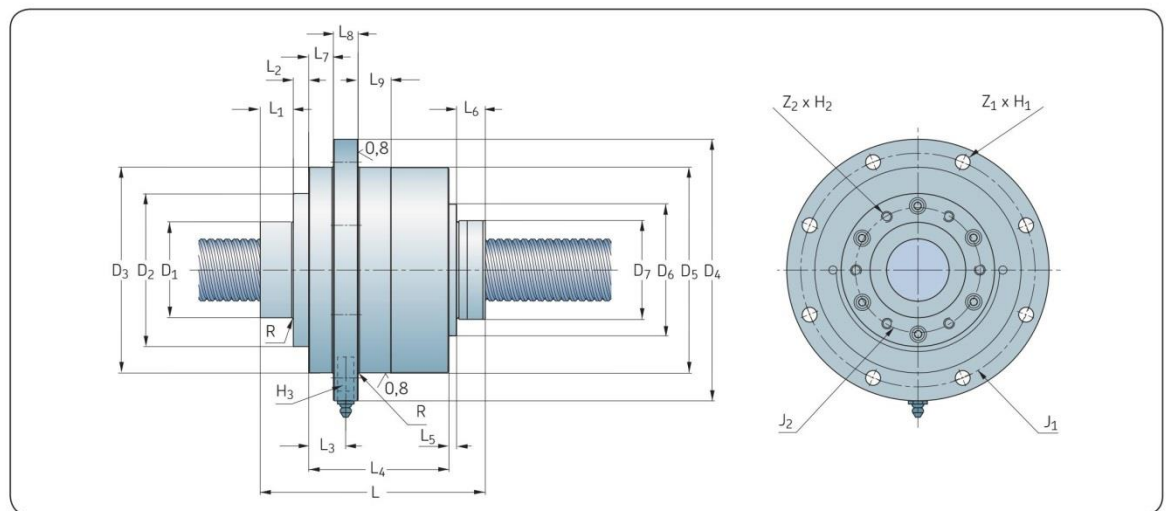


Nominal diameter	Lead (right hand)	Ball screw capacities				Bearing		Rotating nut			Designation	
		SL Basic load ratings		TL Basic load ratings		Basic load ratings		Max transmissible torque	Max transmissible axial load	Inertia with pulley support		Mass
d_0	P_h	C_a	C_{oa}	C_a	C_{oa}	C_a	C_{oa}					–
mm	mm	kN				kN		Nm	kN	kgmm ²	kg	
25	20	39,2	97,0	21,6	48,5	61,8	56	180	68,3	1 012	4,5	SLT/TLT 25×20 R
	25	33,2	80,4	18,3	40,2	61,8	56	180	68,3	1 023	4,6	SLT/TLT 25×25 R
32	20	49,6	141,8	27,3	70,9	78	76,5	209	107	1 935	7,2	SLT/TLT 32×20 R
	32	32,2	88,6	17,7	44,3	78	76,5	209	87,3	1 919	7,1	SLT/TLT 32×32 R
	40	25,3	67,0	13,9	33,5	78	76,5	209	81,7	1 949	7,1	SLT/TLT 32×40 R
40	20	54,2	176,5	29,8	88,3	93,6	91,5	240	116	3 095	7,5	SLT/TLT 40×20 R
	40	51,7	130,5	28,5	65,3	114	118	246	93,3	3 784	8,4	SLT/TLT 40×40 R
50	50	92,9	235,1	51,2	117,6	156	166	803	162	1 1482	15,5	SLT/TLT 50×50 R

Benefits

- High rotational speed up to $nd_0 = 90\,000$, resulting in high linear speed up to 110 m/min
- Compact, easy and simple solution to incorporate into application
- Fixed screw shaft for simplified mounting into application
- Inertia is considerably reduced, for example: 3 800 kgmm² instead of 6 000 kgmm² for a screw shaft 40×40 with 4,5 m stroke
- Lower motor power requirements resulting from lower system inertia
- Backlash elimination (TLT designation).

Screw	Dimensions										
	$d_0 \times P_h$	L	L ₁	L ₂	L ₃	L ₄	L ₅	L ₆	L ₇	L ₈	L ₉
mm											
25×20	121,2	15	12,4	19,9	74	2,9	16,9	12,4	15	15	
25×25	126,3	15	12,4	19,9	74	2,9	22	12,4	15	15	
32×20	132,9	20	3,8	27,5	89	2,2	17,9	20	15	20	
32×32	126,8	20	3,8	27,5	89	2,2	11,8	20	15	20	
32×40	125,9	20	3,8	27,5	89	2,2	10,9	20	15	20	
40×20	136,7	20	9,3	22,5	85	4,7	17,7	15	15	20	
40×40	159,6	47	8,8	19	83	0	20,8	11,5	15	20	
50×50	163,5	20	15,5	25,4	100	4,5	23,5	15,7	20	25	



Screw	Dimensions													
	$d_0 \times P_h$	D ₁	D ₂ h8	D ₃	D ₄	D ₅ g6	D ₆	D ₇	R max.	J ₁	J ₂	Z ₁ × H ₁	Z ₂ × H ₂ × useful depth	H ₃
mm														
25×20	40	72,5	100	133	100	65	48	0,8	116	55	6×Ø9	6×M6×20	M6×1	
25×25	40	72,5	100	133	100	65	48	0,8	116	55	6×Ø9	6×M6×20	M6×1	
32×20	50	82	119,5	150	120	76	56	0,8	135	68	6×Ø9	6×M6×20	M6×1	
32×32	50	82	119,5	150	120	76	50	0,8	135	68	6×Ø9	6×M6×20	M6×1	
32×40	50	82	119,5	150	120	76	53	0,8	135	68	6×Ø9	6×M6×20	M6×1	
40×20	58	93	125	159	125	80	63	0,8	142	75	8×Ø9	6×M6×20	M8×1	
40×40	60	93	137	168	137	N/A	72	1,6	153	80	8×Ø9	6×M6×20	M8×1	
50×50	70	120	170	210	170	110	85	1,6	190	106	8×Ø11	6×M8×30	M8×1	

All tolerances js13 if not specified.

Shaft end combinations

- In the ordering code, shaft ends machining is defined by:
 - One letter for nominal diameter $d_0 < 16$ mm
 - Two letters for nominal diameter $d_0 \geq 16$ mm, detailing the combination of two machined ends (→ designation system **page 54**)
- Machined ends are detailed for nominal diameter < 16 mm (→ **page 37**)
- Machined ends are detailed for nominal diameter ≥ 16 mm (→ **pages 38 to 41**).

S, SA and UA end machining types

^{*)} S and SA: end is machined to thread root diameter d_2 . It is available for all screw shaft nominal diameters (→ **fig. 10**)

^{*)} UA: end is machined to diameter d_3 under induction hardened layer. Any length can be used. UA end machining is available for ball screws with nominal diameter d_0 starting from 16 mm (→ **fig. 10**).

Diameter < 16 mm		Diameter ≥ 16 mm	
Order code	Two machined ends	Order code	Two machined ends
A (without length indication)	cut only	AA (without length indication)	cut only
A (+ length)	cut + annealed		
B	1 + 2	BA	1A + 2A
F ¹⁾	2 + 2	FA ¹⁾	2A + 2A
G ¹⁾	2 + 3	GA ¹⁾	2A + 3A
H	2 + 4	HA	2A + 4A
J	2 + 5	JA	2A + 5A
M	3 + 5	MA	3A + 5A
S ^{*)} (+ length)	end machined to root diameter d_2 , any length	SA ^{*)} (+ length)	end machined to root diameter d_2 , any length
		UA ^{*)} (+ length)	end machined to diameter d_3 under induction hardening, any length
K	keyway	K	keyway
Z	end machined according to customer drawing on request	Z	end machined according to customer drawing on request

¹⁾Attention! This mounting requires the greatest care. Please contact SKF.

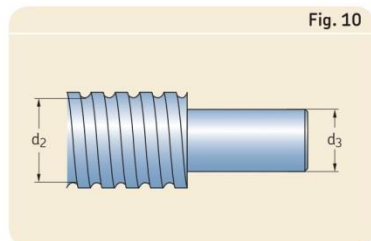


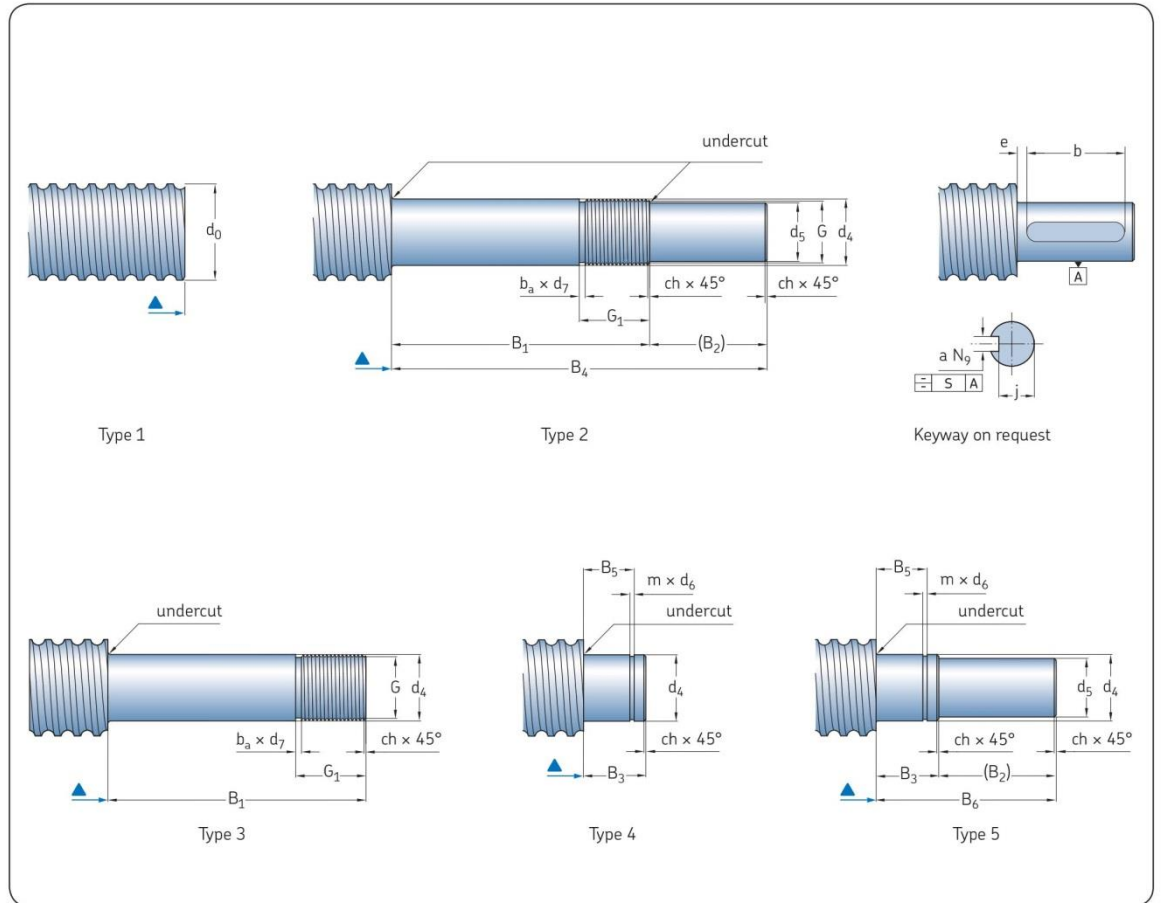
Fig. 10

Dimensions	d_2		d_3	
	mm	mm	mm	mm
6×2	4,7			
8×2,5	6,3			
10×2	8,3			
10×3	7,8			
10×4	7,4			
12×2	9,9			
12×4	9,4			
12×5	9,3			
12,7×12,7	10,2			
14×4	11,9			
16×2	14,3	12		
16×5	12,7	9		
16×10	12,6	9		
20×5	16,7	14		

Dimensions	d_2		d_3	
	mm	mm	mm	mm
25×5	21,7	19		
25×10	20,5	18		
25×20	21,7	19		
25×25	21,5	18		
32×5	28,7	26		
32×10	27,8	25		
32×20	27,4	24		
32×32	28,4	26		
32×40	26,9	24		
40×5	36,7	34		
40×10	34,0	31		
40×20	35,1	32		
40×40	34,2	31		
50×10	44,0	41		
50×50	43,4	40		
63×10	57,0	54		

Standard end machining for nominal diameter < 16 mm

For SD/BD/SH-SDS/BDS/SHS



Dimensions

d_0	d_5 h7	$d_4^{1)}$ js7	B_1 js12	B_2	B_3 js12	B_4 js12	B_5 H11	B_6 js12	G 6g	G_1	m +0,14 0	d_6 h11/ h12	ch	b_a	d_7 h11	a N9	b +0,5 0	e	j	S	Keyway DIN 6885
6	3	4	22	10	7	32	5,4	17	M4x0,7	7	0,5	3,8	0,5	1,2	2,9	-	-	-	-	-	-
8	4	5	24	12	7	36	5,6	19	M5x0,8	7,2	0,7	4,8	0,5	1,2	3,7	-	-	-	-	-	-
10	5	6	26	12	9	38	6,7	21	M6x1	7,5	0,8	5,7	0,5	1,5	4,5	-	-	-	-	-	-
12/12,7	6	8	38	12	10	50	7,8	22	M8x1	12,5	0,9	7,6	0,5	1,5	6,5	2	8	3	4,8	0,1	A2x2x8
14	8	10	40	16	12	56	9	28	M10x1,5	13,3	1,1	9,6	0,5	2,3	7,8	2	10	3	6,8	0,1	A2x2x10

¹⁾ For applications with radial loads on support bearings, please consult SKF for best selection of tolerance on diameter d_4

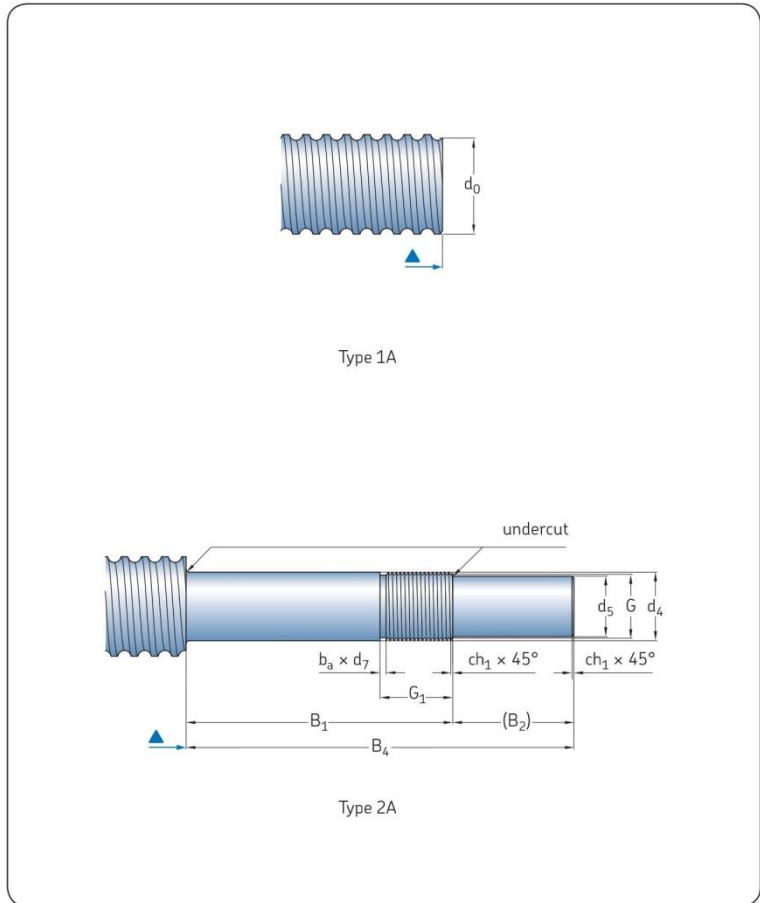
Standard end machining for shaft nominal diameter ≥ 16 mm

For SD/BD-SDS/BDS-SX/BX-SND/BND/PND-SN/BN/PN

Standard shaft ends for ball screws with nominal diameter $d_0 \geq 16$ mm have been developed to fit with the SKF support bearings FLBU, PLBU and BUF.

Support bearing	Machined end type
FLBU	2A or 3A
PLBU	2A or 3A
BUF	4A or 5A

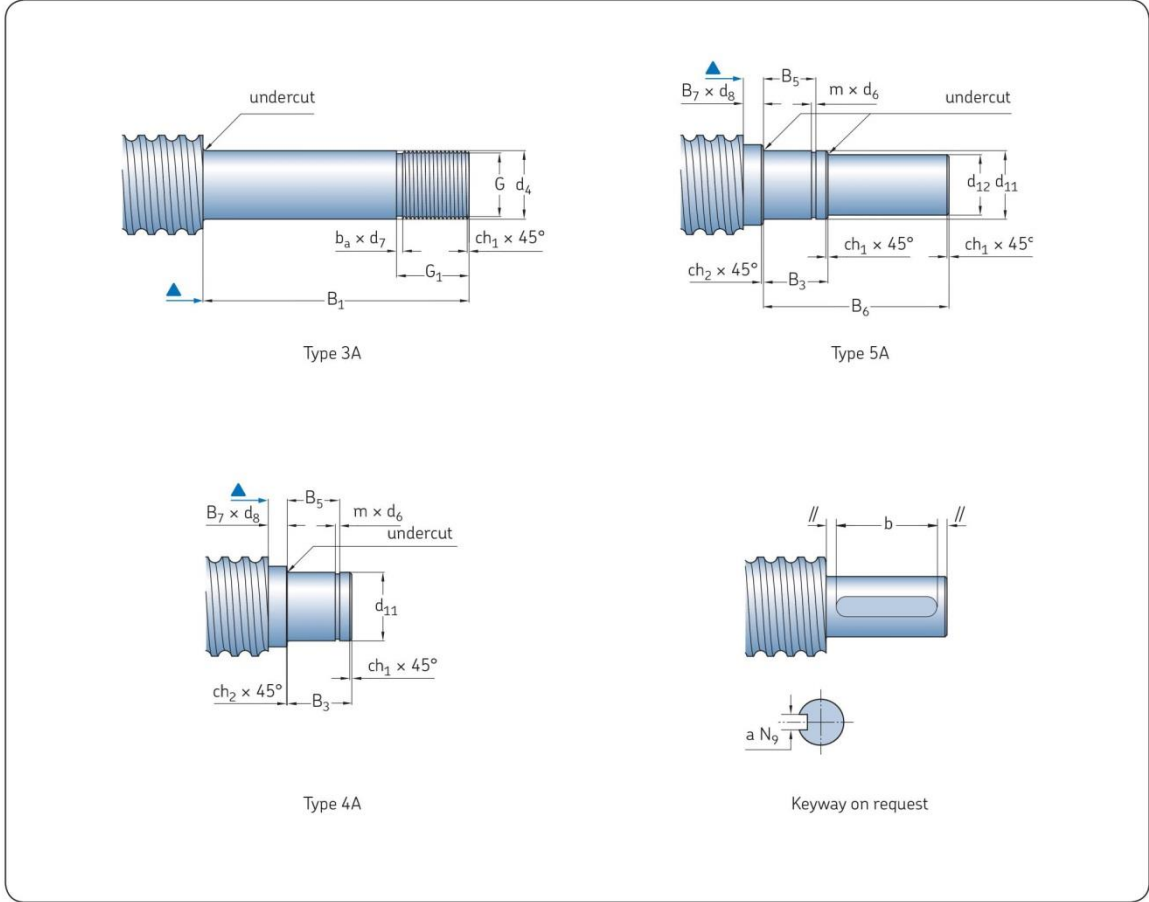
For these types of machined ends, the maximum permissible dynamic load is 75% of the ball screw dynamic load carrying capacity.



Size

d_0 —	d_5 h7	$d_4^{1)}$ h6	d_{11} h6	d_{12} h7	B_1 js12	B_2	B_3 js12	B_4 js12	B_5 H11	B_6 js12	B_7	d_8
mm												
16	8	10	10	8	53	16	13	69	10	29	2	12,5
20	10	12	10	8	58	17	13	75	10	29	2	14,5
25 ¹⁾	15	17	17	15	66	30	16	96	13	46	4,5	20
32 ¹⁾	17	20	17	15	69	30	16	99	13	46	4,5	21,7
40 ¹⁾	25	30	30	25	76	45	22	121	17,5	67	4,5	33,5
50 ¹⁾	30	35	30	25	84	55	22	139	17,5	67	4,5	35,2
63	40	50	45	40	114	65	28	179	20,75	93	3	54

¹⁾ For applications with radial loads on support bearings, please consult SKF for best selection of tolerance on diameter d_4 .



Size										Keyway to DIN 6885	
d_0	G 6g	G_1	m +0,14 0	d_6 h11	h_{12}	ch_1	ch_2	b_a	d_7 h11	$a^{N9} \times l \times b$	
										fixed end (type 2A)	fixed end (type 5A)
mm											
16	M10x0,75	17	1,1	9,6		0,5	0,5	1,2	8,8	A2x2x12	A2x2x12
20	M12x1	18	1,1	9,6		0,5	0,5	1,5	10,5	A3x3x12	A2x2x12
25	M17x1	22	1,1	16,2		0,5	0,5	1,5	15,5	A5x5x25	A5x5x25
32	M20x1	22	1,1	16,2		0,5	0,5	1,5	18,5	A5x5x25	A5x5x25
40	M30x1,5	25	1,6		28,6	1	0,5	2,3	27,8	A8x7x40	A8x7x40
50	M35x1,5	27	1,6		28,6	1	0,5	2,3	32,8	A8x7x45	A8x7x40
63	M50x1,5	32	1,85		42,5	1,5	1	2,3	47,8	A12x8x50	A12x8x50

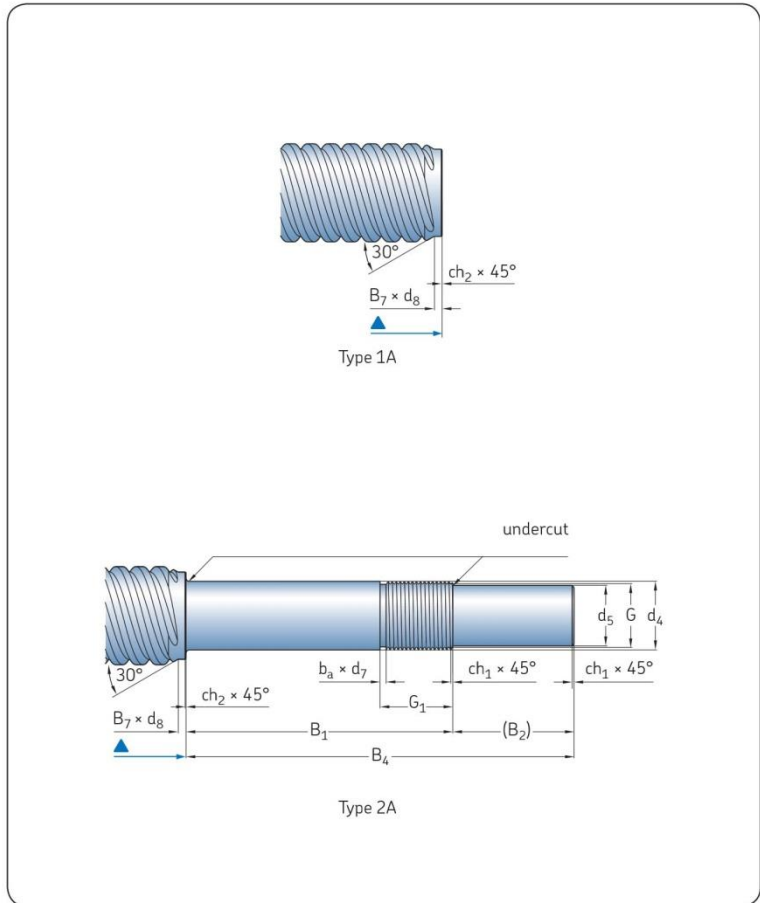
Standard end machining for SL/TL only

Standard shaft ends for SL/TL ball screws have been developed to fit with the SKF support bearings FLBU, PLBU and BUF.

For the SL/TL long lead screw, an additional centering diameter, part of the threaded length, will be machined at both ends of the screw shaft to facilitate the nut assembly.

Support bearing	Machined end type
FLBU	2A or 3A
PLBU	2A or 3A
BUF	4A or 5A

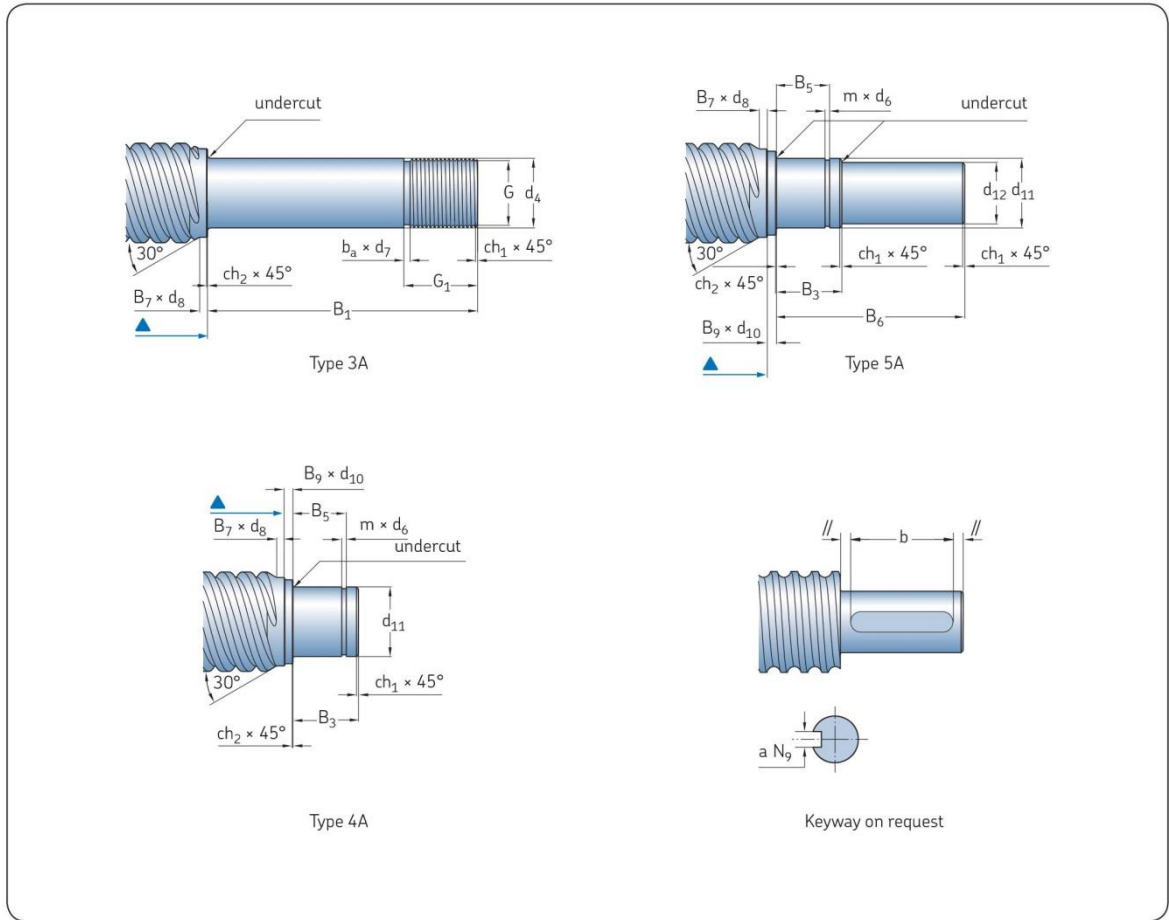
For these types of machined ends, the maximum permissible dynamic load is 75% of the ball screw dynamic load carrying capacity, except for size 50x50 for which the dynamic load must not exceed 40 kN.



Size

d_0	d_5 h7	$d_4^{1)}$ h6	d_{10}	d_{11} h6	d_{12} h7	B_1 js12	B_2	B_3 js12	B_4 js12	B_5 H11	B_6 js12	B_7	B_9	d_8
mm														
25x20	15	17	–	17	15	66	30	16	96	13	46	4,5	0	21,6
25x25	15	17	–	17	15	66	30	16	96	13	46	4,5	0	21,4
32x20	17	20	21,5	17	15	69	30	16	99	13	46	4,5	2	27,3
32x32	17	20	21,5	17	15	69	30	16	99	13	46	4,5	2	28,3
32x40	17	20	21,5	17	15	69	30	16	99	13	46	4,5	2	26,8
40x20	25	30	–	30	25	76	45	22	121	17,5	67	6,5	0	35,1
40x40	25	30	–	30	25	76	45	22	121	17,5	67	6,5	0	34,1
50x50	30	35	37	30	25	84	55	22	139	17,5	67	9	3	43,3

¹⁾ For applications with radial loads on support bearings, please consult SKF for best selection of tolerance on diameter d_4 .



Size	G	G ₁	m	d ₆	h ₁₂	ch ₁	ch ₂	b _a	d ₇	Keyway to DIN 6885	
										a ^{N9} × l × b	fixed end (type 2A)
d ₀ × P _h	6g		+0,14 0	h ₁₁					h ₁₁		
mm											
25×20	M17×1	22	1,1	16,2	–	0,5	0,5	1,5	15,5	A5×5×25	A5×5×25
25×25	M17×1	22	1,1	16,2	–	0,5	0,5	1,5	15,5	A5×5×25	A5×5×25
32×20	M20×1	22	1,1	16,2	–	0,5	0,5	1,5	18,5	A5×5×25	A5×5×25
32×32	M20×1	22	1,1	16,2	–	0,5	0,5	1,5	18,5	A5×5×25	A5×5×25
32×40	M20×1	22	1,1	16,2	–	0,5	0,5	1,5	18,5	A5×5×25	A5×5×25
40×20	M30×1,5	25	1,6	–	28,6	1	0,5	2,3	27,8	A8×7×40	A8×7×40
40×40	M30×1,5	25	1,6	–	28,6	1	0,5	2,3	27,8	A8×7×40	A8×7×40
50×50	M35×1,5	27	1,6	–	28,6	1	0,5	2,3	32,8	A8×7×45	A8×7×40

▲ End of threaded screw length

FLBU ball screw support bearings

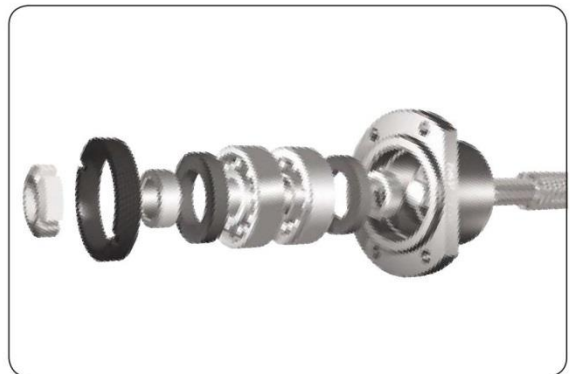
Axially locating flanged housings fitted with SKF angular contact ball bearings

Features

- Precision machined housing made of bur-nished steel
- Two SKF preloaded angular contact ball bearings, 72 or 73 series, in back-to-back arrangement
- Two garter seals
- Standard self-locking Nylstop nut or high precision KMT nut upon request.

Benefits

- Complete support bearing ready to use, simplified application design, easy ordering process
- Quick assembly onto shaft end
- Elimination of most technical risks with bearings and seals assembly
- Support bearing dimensions and load carrying capacity matched to the ball screw characteristics
- Bearings back-to-back assembly with preload for stiff and accurate ball screw positioning
- Greased for life / maintenance-free.

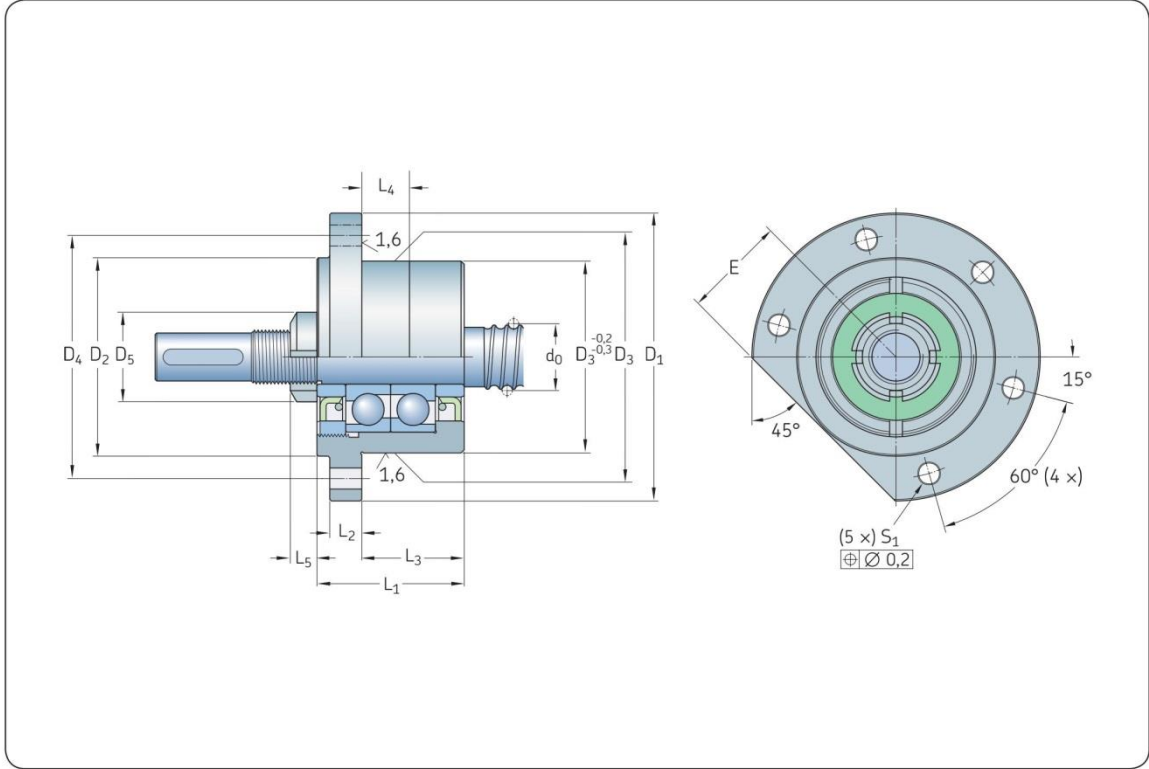


Nominal diameter	Angular contact ball bearing (40°)			SKF bearing designation	Lock nut		High precision nut ³⁾		Tightening torque	Grub screws size	Grub screws tightening torque max.	Flanged support bearing designation
	Basic load rating (axial) dynamic	static	Axial stiffness		Self-locking nut Designation	Hook spanner	Designation	Hook spanner				
d ₀	C _a	C _{0a}										
mm	kN		N/μm	–	–	–	–	Nm	–	Nm	–	–
16	12,2	12,8	play	7200 BECB ¹⁾	CN 70-10	HN 1	KMT 0	HN 2/3	4	M5	4,5	FLBU 16
20	13,3	14,7	125	7201 BEGA ²⁾	CN 70-12	HN 1	KMT 1	HN 3	8	M5	4,5	FLBU 20
25	27,9	31,9	150	7303 BEGA ²⁾	CN 70-17	HN3	KMT 3	HN 4	15	M6	8	FLBU 25
32	24,6	31,9	176	7204 BEGA ²⁾	CN 70-20	HN 4	KMT 4	HN 5	18	M6	8	FLBU 32
40	41,9	59,6	222	7206 BEGA ²⁾	CN 70-30	HN 6	KMT 6	HN 6	32	M6	8	FLBU 40
50	54,5	79,8	250	7207 BEGA ²⁾	CN 70-35	HN 7	KMT 7	HN 7	40	M6	8	FLBU 50
63	128	196,1	353	7310 BEGA ²⁾	CN 70-50	HN 10	KMT 10	HN 10/11	60	M8	18	FLBU 63

¹⁾ No backlash elimination

²⁾ Light preload

³⁾ Optional



D

Screw Support bearing

d ₀	L ₁	L ₂	L ₃	L ₄	Self-locking nut		High precision nut ⁴⁾			D ₂	D ₃ h7	D ₄	S ₁ H13	E	Fixing screws
					L ₅	D ₅	L ₅	D ₅	D ₁						
mm	mm														
16	37	10	22	12	7	18	14	28	76	50	47	63	6,6	26	M6×30
20	42	10	25	12	7,5	21	14	30	76	50	47	63	6,6	27	M6×30
25	46	10	32	18	8,3	28	18	37	90	62	60	76	6,6	32	M6×30
32	49	13	32	18	8,3	32	18	40	90	59	60	74	9	32	M8×40
40	53	16	32	18	11	44	20	49	120	80	80	100	11	44	M10×45
50	59	20	32	18	11	50	22	54	130	89	90	110	13	49	M12×60
63	85	25	43,5	22	11,7	68	25	75	165	124	124	146	13	64	M12×60

⁴⁾ Optional

PLBU ball screw support bearings

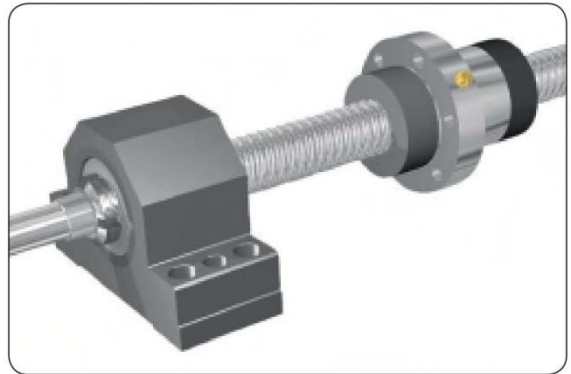
Fixed pillow blocks fitted with SKF angular contact ball bearings

Features

- Precision machined housing made of bur-nished steel
- Precision machined side faces of the housing can be used as reference assembly surfaces for screw alignment
- Two SKF preloaded angular contact ball bearings, 72 or 73 series, in back-to-back arrangement
- Two garter seals
- Standard self-locking Nylstop nut or high precision KMT nut upon request.

Benefits

- Complete support bearing ready to use, simplified application design, easy ordering process
- Quick assembly onto shaft end
- Elimination of most technical risks with bearings and seals assembly
- Support bearing dimensions and load carrying capacity matched to the ball screw characteristics
- Bearings back-to-back assembly with preload for stiff and accurate ball screw positioning
- Good rigidity provided by the base mounting with dowel pins
- Greased for life / maintenance-free.

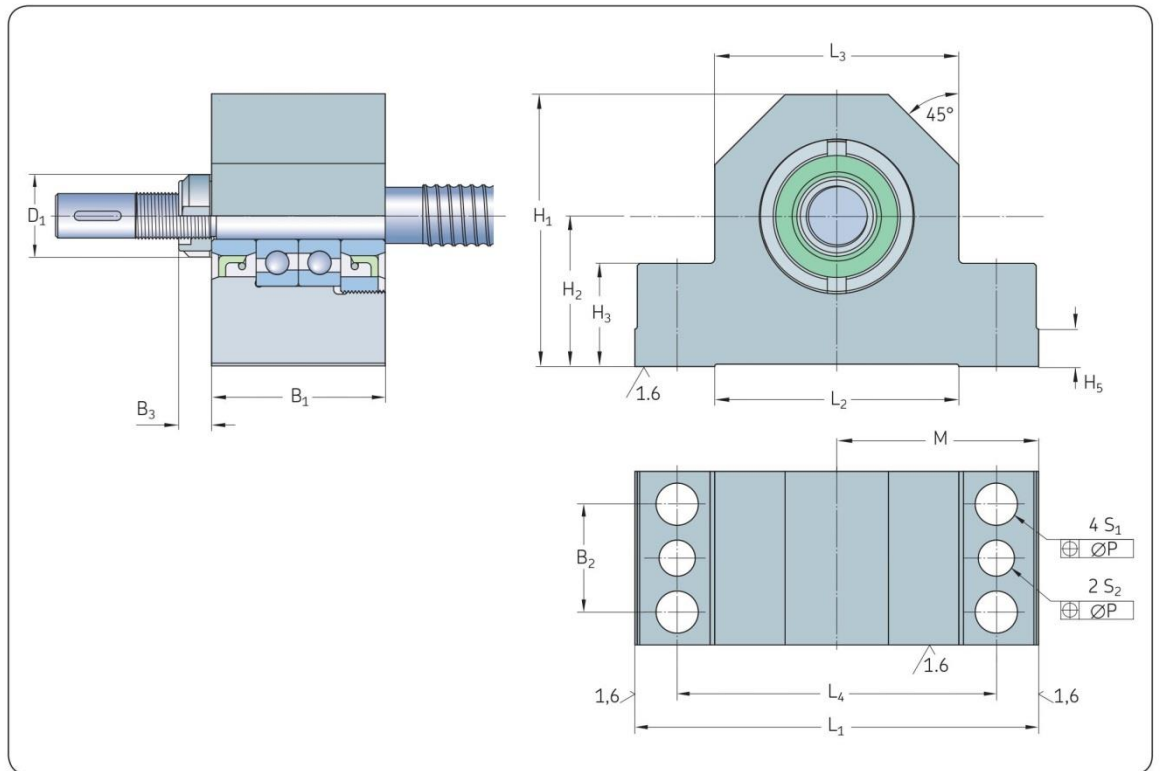


Nominal diameter	Angular contact ball bearing (40°)			SKF bearing designation	Lock nut		High precision nut ³⁾		Tightening torque	Grub screws size	screws tightening torque max.	Pillow block designation
	Basic load rating (axial) dynamic	static	Axial stiffness		Self-locking nut Designation	Hook spanner	Designation	Hook spanner				
d ₀	C _a	C _{0a}	N/μm	–	–	–	–	Nm	–	Nm	–	
16	12,2	12,8	play	7200 BECB ¹⁾	CN 70-10	HN 1	KMT 0	HN 2/3	4	M5	4,5	PLBU 16
20	13,3	14,7	125	7201 BEGA ²⁾	CN 70-12	HN 1	KMT 1	HN 3	8	M5	4,5	PLBU 20
25	27,9	31,9	150	7303 BEGA ²⁾	CN 70-17	HN3	KMT 3	HN 4	15	M6	8	PLBU 25
32	24,6	31,9	176	7204 BEGA ²⁾	CN 70-20	HN 4	KMT 4	HN 5	18	M6	8	PLBU 32
40	41,9	59,6	222	7206 BEGA ²⁾	CN 70-30	HN 6	KMT 6	HN 6	32	M6	8	PLBU 40
50	54,5	79,8	250	7207 BEGA ²⁾	CN 70-35	HN 7	KMT 7	HN 7	40	M6	8	PLBU 50
63	128	196,1	353	7310 BEGA ²⁾	CN 70-50	HN 10	KMT 10	HN 10/11	60	M8	18	PLBU 63

¹⁾ No backlash elimination

²⁾ Light preload

³⁾ Optional



D

Screw Support bearing

d ₀	L ₁	L ₂	L ₃	L ₄	M	B ₁	B ₂	Self-locking nut B ₃	D ₁	High precision nut ⁴⁾ B ₃	D ₁	H ₁	H ₂ js8	H ₃	H ₄	H ₅	S ₁	P	S ₂ H12	Fixing screws	Tapered pin (hardened) or straight pin (DIN6325)	
mm	mm																					
16	86	52	52	68	43	37	23	7,0	18	14	28	58	32	22	15	8	9	0,15	7,7	M8x35	8x40	
20	94	52	60	77	47	42	25	7,5	21	14	30	64	34	22	17	8	9	0,15	7,7	M8x35	8x40	
25	108	65	66	88	54	46	29	8,3	28	18	37	72	39	27	19	10	11	0,20	9,7	M10x40	10x50	
32	112	65	70	92	56	49	29	8,3	32	18	40	77	45	27	20	10	11	0,20	9,7	M10x40	10x50	
40	126	82	80	105	63	53	32	11,0	44	20	49	98	58	32	23	12	13	0,20	9,7	M12x50	10x50	
50	144	80	92	118	72	59	35	11,0	50	22	54	112	65	38	25	12	13	0,20	9,7	M12x55	10x55	
63	190	110	130	160	95	85	40	11,7	68	25	75	130	65	49	35	15	13	0,20	9,7	M12x65	10x65	

⁴⁾ Optional

BUF ball screw support bearings

Axially free pillow blocks fitted with SKF deep groove ball bearing

Features

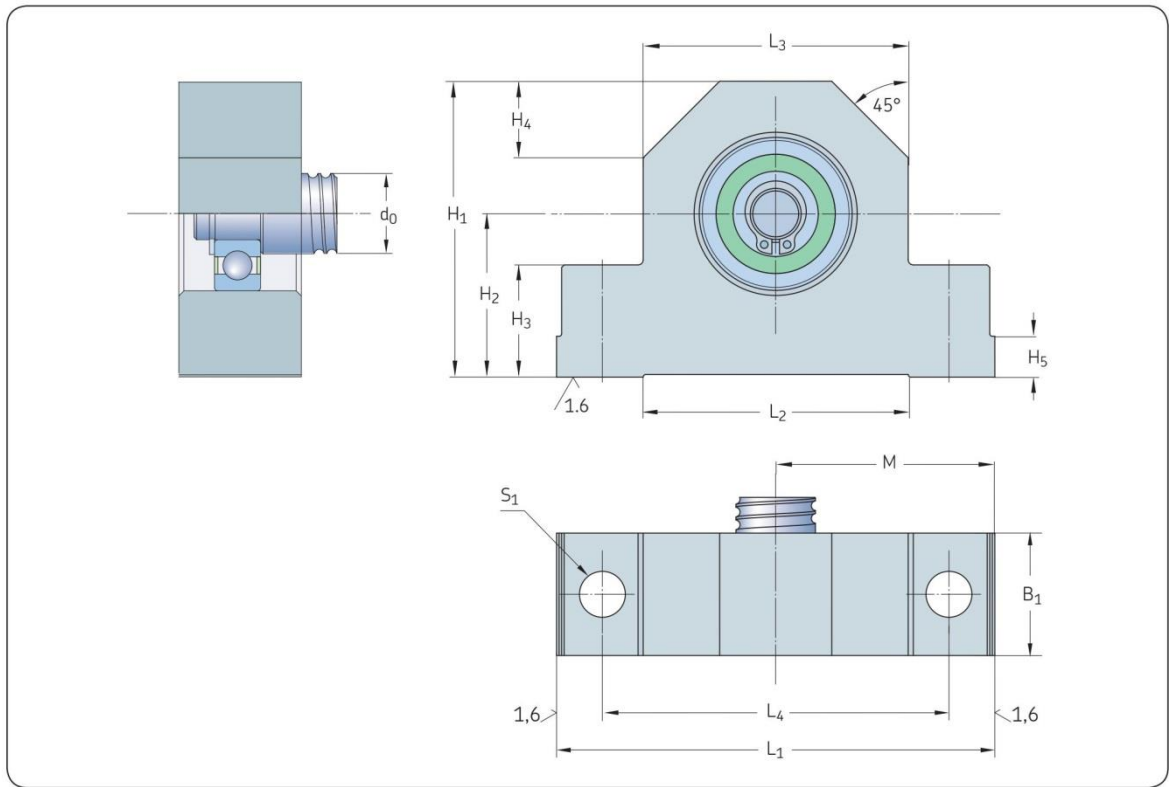
- Precision machined housing made of bur-nished steel
- Precision machined side faces of the housing can be used as reference assembly surfaces for screw alignment
- One SKF deep groove ball bearing of type 62...2RS1
- Bearing is sealed and greased for life
- Retaining ring is supplied with the BUF assembly.

Benefits

- Complete support bearing ready to use, simplified application design, easy ordering process
- Quick assembly onto shaft end
- Elimination of most technical risks with bearings and seals assembly
- Greased for life / maintenance-free.



Nominal diameter	Deep groove ball bearing		SKF bearing designation	Dimensions			Retaining ring (DIN 471)	Pillow block designation
	Basic load rating radial			d	D	B		
d_0	C	C_0		mm	mm	mm		
mm	kN	kN	–	mm	mm	mm	–	–
16	5,07	2,36	6200.2RS1	10	30	9	10×1	BUF 16
20	5,07	2,36	6200.2RS1	10	30	9	10×1	BUF 20
25	9,56	4,75	6203.2RS1	17	40	12	17×1	BUF 25
32	9,56	4,75	6203.2RS1	17	40	12	17×1	BUF 32
40	19,5	11,2	6206.2RS1	30	62	16	30×1,5	BUF 40
50	19,5	11,2	6206.2RS1	30	62	16	30×1,5	BUF 50
63	33,2	21,6	6209.2RS1	45	85	19	45×1,75	BUF 63



Screw Support bearing

Fixing screws

d_0	L_1	L_2	L_3	L_4	M js8	B_1	H_1	H_2 js8	H_3	H_4	H_5	S_1 H12	Fixing screws
mm													-
16	86	52	52	68	43	24	58	32	22	15	8	9	M8x35
20	94	52	60	77	47	26	64	34	22	17	8	9	M8x35
25	108	65	66	88	54	28	72	39	27	19	10	11	M10x40
32	112	65	70	92	56	34	77	45	27	20	10	11	M10x40
40	126	82	80	105	63	38	98	58	32	23	12	13	M12x50
50	144	80	92	118	72	39	112	65	38	25	12	13	M12x55
63	190	110	130	160	95	38	130	65	49	35	15	13	M12x65

Examples of customized nuts



SD rotating nut with flange and bearing journals



SDS nut with integrated trunnions



SN rotating nut with flange and bearing journals

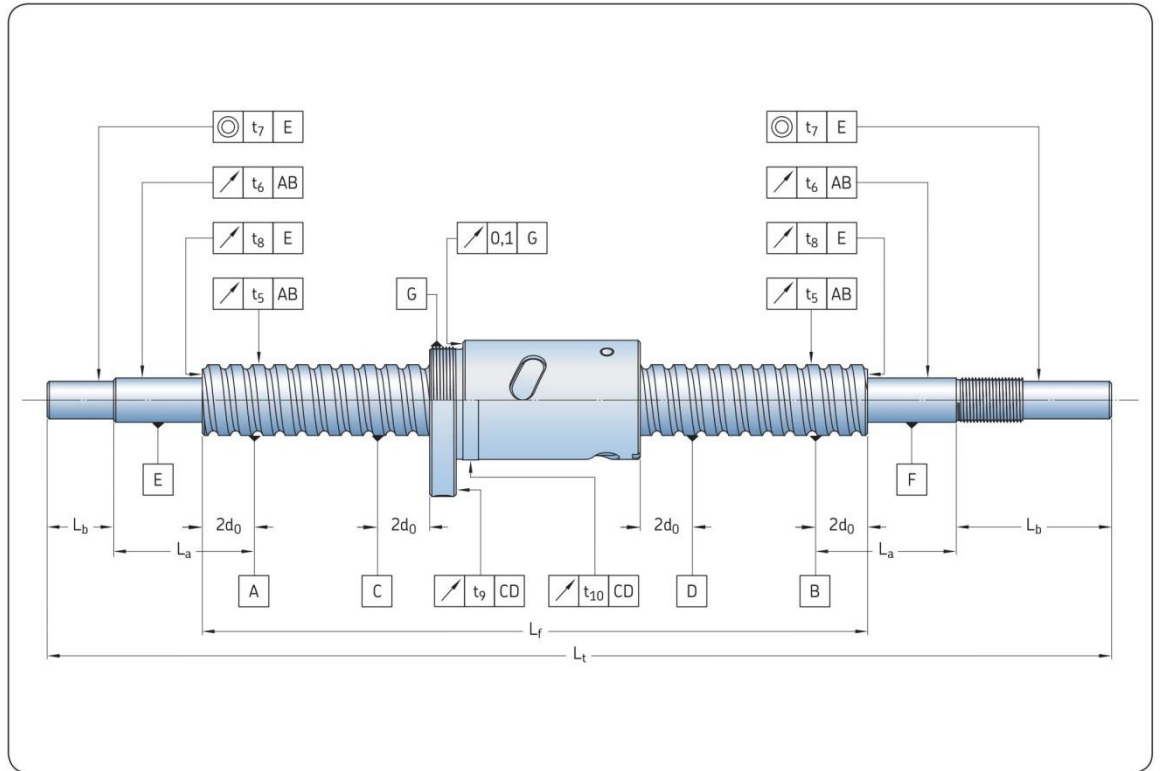


PN nut with customized compact flange



SL nut with customized flange attachment

Manufacturing tolerances



G

Nomineller Durchmesser		Reference lengths		Tolerances						Ratio		Tolerance
d_0		$L_{f\text{ref}}$	$L_{a\text{ref}}$ and $L_{b\text{ref}}$	t_{5p}	t_{6p}	t_{7p}	t_8	t_9	t_{10}	L_f/d_0		t_5
>	≤									>	≤	
mm		μm										
6	12	80	80	40	40	12	6			-	40	80
12	16	160	80	40	40	12	6	20	20	40	60	120
16	20	160	80	40	40	12	6	20	25	60	80	200
20	25	160	125	40	50	16	6	20	25	80	100	320
25	40	315	125	40	50	16	6	25	25			
40	50	315	125	40	50	16	6	25	32			
50	63	630	200	40	63	20	6	25	32			

Value for t_5

- if $L_f \leq L_{f\text{ref}}$ $t_5 = t_{5p}$
- if $L_f > L_{f\text{ref}}$ t_5

Value for t_6

- if $L_a \leq L_{a\text{ref}}$ $t_6 = t_{6p}$
- if $L_a > L_{a\text{ref}}$ $t_6 = (L_a/L_{a\text{ref}}) t_{6p}$

Value for t_7

- if $L_b \leq L_{b\text{ref}}$ $t_7 = t_{7p}$
- if $L_b > L_{b\text{ref}}$ $t_7 = (L_b/L_{b\text{ref}}) t_{7p}$

Calculation formulae

Basic life rating

$$L_{10} = \left(\frac{C_a}{F_m} \right)^3$$

Required load rating

$$C_{req} = F_m (L_{10})^{1/3}_{req}$$

where

L_{10} = life [million revolutions]

C_a = basic dynamic load rating [N]

C_{req} = required dynamic load rating [N]

F_m = cubic mean load [N]

Equivalent mean load

- Duty cycle with step loading

$$F_m = \frac{(F_1^3 L_1 + F_2^3 L_2 + F_3^3 L_3 + \dots)^{1/3}}{(L_1 + L_2 + L_3 + \dots)^{1/3}}$$

where

L_n = load period n (→ diagram 2)

F_n = load during period n (→ diagram 2)

F_n can be a fixed value, or F_n can be calculated using the following formulae for F_m

- Duty cycle with continuous load variation

$$F_m = \frac{F_{min} + 2F_{max}}{3}$$

where

F_{min} = minimum load (→ diagram 3)

F_{max} = maximum load (→ diagram 3)

Critical speed of screw shaft (no safety factor)

$$n_{cr} = 49 \times 10^6 \frac{f_1 d_2}{l^2}$$

where

n_{cr} = critical speed [rpm]

d_2 = root diameter [mm]

l = free length, or distance between the two support bearings [mm]

f_1 = mounting correction factor

0,9 ●● — fixed, free

3,8 ●● — fixed, radial support

5,6 ●● — fixed, fixed

Note: it is generally recommended to apply a safety factor of 0,8 to the calculated value of the critical speed n_{cr} of the screw shaft.

Speed limit of the mechanism (maximal speed applied through very short periods)

With recirculation by inserts / tubes (SD/BD/SH-SDS/BDS/SHS-SX/BX-SND/BND/PND-SN/BN/PN):

$$n d_0 < 50\,000$$

With recirculation through flange (SL/TL-SLD/TLD):

$$n d_0 < 90\,000$$

If $n d_0 > 50\,000$ or $90\,000$ respectively, please consult SKF

where

n = rotational speed [rpm]

d_0 = screw shaft nominal diameter [mm]

Maximum admissible acceleration is

$$4\,000 \text{ rad/s}^2$$

Buckling strength, with safety factor 3

$$F_c = \frac{34 \times 10^3 f_3 d_2^4}{l^2}$$

where

F_c = buckling strength [N]

d_2 = root diameter [mm]

l = free length, or distance between the two support bearings [mm]

f_3 = mounting correction factor

0,25 ●● — fixed, free

2 ●● — fixed, radial support

4 ●● — fixed, fixed

Diagram 2

Equivalent mean load

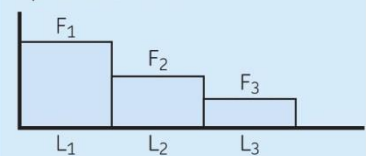
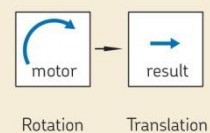


Diagram 3

Equivalent mean load

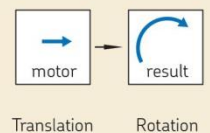


Fig. 11



Rotation Translation

Fig. 12



Translation Rotation

Fig. 13

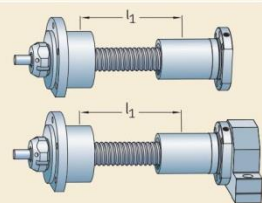
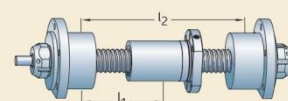


Fig. 14



Theoretical efficiencies

Direct (→ fig. 11)

$$\eta = \frac{1}{1 + \frac{\pi d_0}{P_h} \mu}$$

where

$\mu = 0,0065$ for SH/SHS

$\mu = 0,006$ for SD/BD, SDS/BDS, SX/BX, SND/BND/PND, SN/BN/PN, SL/TL, SLT/TLT

d_0 = nominal diameter of screw shaft [mm]

P_h = lead [mm]

Indirect (→ fig. 12)

$$\eta' = 2 - \frac{1}{\eta}$$

Practical efficiency

$$\eta_p = 0,9 \eta$$

The value 0,9 is an average value between the practical efficiency of a new screw and that of a properly run-in screw.

It should be used for industrial applications in all normal working conditions. For extreme cases, please contact SKF.

Input torque in a steady state

$$T = \frac{F P_h}{2\,000 \pi \eta_p}$$

where

T = input torque [Nm]

F = maximum load of the cycle [N]

P_h = lead [mm]

η_p = practical efficiency

Power requirement in a steady state

$$P = \frac{F n P_h}{60\,000 \eta_p}$$

where

P = power required [W]

n = revolutions per minute [rpm]

Preload torque [Nm]

$$T_{pr} = \frac{F_{pr} P_h}{1\,000 \pi} \left(\frac{1}{\eta_{pr}} - 1 \right)$$

where

T_{pr} = preload torque [N]

F_{pr} = preload [N]

η_{pr} is calculated using $\mu = 0,01$ for preloaded system

Restraining torque (considering a back-driving system)

$$T_B = \frac{F P_h \eta'}{2\,000 \pi}$$

where

T_B = restraining torque [Nm]

F = load [N]

For safety reasons, we use the theoretical indirect efficiency.

Nominal motor torque during acceleration

For a horizontal screw

$$T_t = T_f + T_{pr} + \frac{P_h [F + m_L \mu_f g]}{2\,000 \pi \eta_p} + \dot{\omega} \Sigma I$$

For a vertical screw

$$T_t = T_f + T_{pr} + \frac{P_h [F + m_L g]}{2\,000 \pi \eta_p} + \dot{\omega} \Sigma I$$

where

T_t = nominal torque [Nm]

T_f = torque from friction in support bearings, motors, seals, etc... [Nm]

T_{pr} = preload torque [Nm]

μ_f = coefficient of friction

$\dot{\omega}$ = angular acceleration [rad/s²]

m_L = mass of the load [kg]

g = acceleration of gravity [9,8 m/s²]

$\Sigma I = I_M + I_L + I_S \cdot 10^{-9}$

Nominal braking torque during deceleration

For a horizontal screw

$$T'_t = T_f + T_{pr} + \frac{P_h \eta' [F + m_L \mu_f g]}{2\,000 \pi} + \dot{\omega} \Sigma I$$

For a vertical screw

$$T'_t = T_f + T_{pr} + \frac{P_h \eta' [F + m_L g]}{2\,000 \pi} + \dot{\omega} \Sigma I$$

where

$$I_L = m_L \left(\frac{P_h}{2\pi} \right)^2 \cdot 10^{-6}$$

where

I_M = inertia of motor [kgm²]

I_S = inertia of screw shaft per metre [kgmm²/m]

l = length of screw shaft [mm]

Static axial stiffness of a complete ball screw assembly

$$\frac{1}{R_t} = \frac{1}{R_s} + \frac{1}{R_n} + \frac{1}{R_p}$$

where

R_t = stiffness of a complete assembly [N/μm]

R_s = shaft stiffness [N/μm]

R_n = nut stiffness [N/μm]

R_p = support bearings stiffness [N/μm]

Shaft stiffness

Fixed-free or fixed-radial support

$$R_s = 165 \frac{d_2^2}{l_1} \quad (\rightarrow \text{fig. 13})$$

Fixed-fixed assembly

$$R_s = \frac{165 d_2^2 l_2}{l_1 (l_2 - l_1)} \quad (\rightarrow \text{fig. 14})$$

where

l_1 = distance center of fixed support bearing to center of nut [mm]

l_2 = distance between centers of fixed support bearings

For additional information, please contact SKF.

Calculation example for a ball screw

Description of customer application:

- Ball screw type PND 25 × 5. Ball screw is described (→ page 26): Nut with internal preload, 2 × 3 circuits, dynamic carrying capacity $C_a = 12,7$ kN, and static carrying capacity $C_{0a} = 22,7$ kN
- Screw shaft is horizontally mounted and supported by two support bearings of types PLBU25 and BUF25
- Load cycle as follows:
 - Phase 1: Steady axial load of 3 kN, on travel of 900 mm, with linear speed 100 mm/s, or phase duration of 9 seconds
 - Phase 2: Regular load increase from 3 kN to 7 kN, on travel of 100 mm, with linear speed 10 mm/s, or phase duration of 10 seconds
 - Phase 3: Nut return to initial position, with steady load of 2 kN, on travel of 1 000 mm, with linear speed 100 mm/s, or phase duration of 10 seconds
 - Then period of 31 seconds with zero load, no displacement
 - Operation during 7 hours per day, 5 days per week, 50 weeks per year.

Calculation of equivalent mean load

First, we check that the maximum load from the duty cycle does not create an excessive loading condition that would be detrimental to service life. Please refer to explanation in paragraph "Service life" (→ page 8).

Calculation of basic life rating L_{10}

Fig. 15

Shaft end 2A for PND 25 × 5

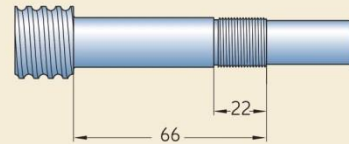
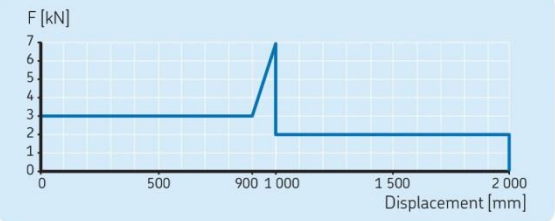
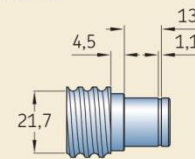


Fig. 16

Shaft end 4A for PND 25 × 5



Maximum application load = 7 kN,
while 60% of $C_a = 60\% \times 12,7 = 7,6$ kN ⇒ OK

$$F_1 = 3000 \text{ N} \quad \text{on } L_1 = 900 \text{ mm}$$

$$F_2 = \frac{3000 + 2 \times 7000}{3} = 5667 \text{ N} \quad \text{on } L_2 = 100 \text{ mm}$$

$$F_3 = 2000 \text{ N} \quad \text{on } L_3 = 1000 \text{ mm}$$

$$F_m = \sqrt[3]{\frac{3000^3 \times 900 + 5667^3 \times 100 + 2000^3 \times 1000}{900 + 100 + 1000}} = 2934 \text{ N}$$

$$L_{10} = \left(\frac{12700}{2934} \right)^3 = 81,1 \text{ millions revolutions}$$

$$\text{Number of nut revolutions per one complete cycle} = (2 \times 1000) / 5 = 400 \text{ revolutions}$$

$$\text{Or } (81,1 \times 10^6) / 400 = 202750 \text{ complete cycles}$$

$$\text{One complete cycle lasts } (9 + 10 + 10 + 31) = 60 \text{ seconds}$$

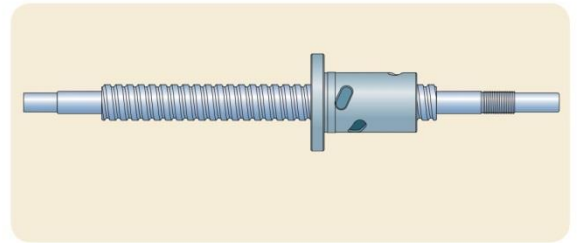
$$\text{Or life rating of } (202750 \times 60) / (3600 \times 7 \times 5 \times 50) = 1,9 \text{ years with 90\% reliability}$$

Critical speed of screw shaft

The critical speed must be checked, especially when the nut travel is long compared to the shaft diameter. Maximum speed during the duty cycle:

$$V_{\max} = \frac{100}{5} \times 60 = 1\,200 \text{ rpm}$$

The shaft threaded length is calculated with considering the total nut travel (1 000 mm), plus the nut length (62 mm), plus extra free length at each shaft end equal to two leads ($2 \times 2 \times 5 = 20$ mm).



Screw assembly is horizontally mounted. End machining is 2A for support bearing PLBU25, and end machining is type 4A for support bearing BUF25. The combination of 2A + 4A ends machining is called HA when ordering the screw (→ page 36).

So total threaded length = 1 082 mm

For end type 2A, with screw nominal diameter $d_0 = 25$ mm, the central axial position of the bearings is calculated with data from pages 38 and 39:

$(B_1 - G_1) / 2 = (66 - 22) / 2 = 22$ mm from the end of the threaded shaft (→ fig. 15).

For end type 4A, with screw nominal diameter $d_0 = 25$ mm, the central axial position of the bearing is calculated with data from pages 38 and 39:

$B_7 + ((B_5 - m) / 2) = 4,5 + ((13 - 1,1) / 2) \approx 11$ mm from the end of the threaded shaft (→ fig. 16).

The free length between the two support bearings is:

$$l = 1\,082 + 22 + 11 = 1\,115 \text{ mm}$$

The root diameter of the threaded shaft is:

$$d_2 = 21,7 \text{ mm (→ page 27 or 36).}$$

Calculation of critical speed:

$$n_{cr} = 49 \times 10^6 \frac{3,8 \times 21,7}{1\,115^2} = 3\,250 \text{ rpm} > V_{\max} \Rightarrow \text{OK}$$

Speed limit

$$n \times d_0 = 1\,200 \times 25 = 30\,000 < 50\,000 \Rightarrow \text{OK}$$

Buckling strength

$$F_c = \frac{34,10^3 \times 2 \times 21,7^4}{1\,115^2} = 12,1 \text{ kN} > F_{\max} = 7 \text{ kN} \Rightarrow \text{OK}$$

Theoretical direct efficiency

$$\eta = \frac{1}{1 + \frac{\pi \times 25}{5} \times 0,006} = 0,914$$

Theoretical indirect efficiency

$$\eta' = 2 - \frac{1}{0,914} = 0,906$$

Practical efficiency

$$\eta_p = 0,9 \times 0,914 = 0,823$$

Input torque in a steady state

$$T = \frac{7\,000 \times 5}{2\,000 \pi \times 0,823} = 6,8 \text{ Nm}$$

Power requirement in a steady state

$$\text{Phase 1: } P = \frac{3\,000 \times 1\,200 \times 5}{60\,000 \times 0,823} = 365 \text{ W}$$

$$\text{Phase 2: } P = \frac{7\,000 \times 1\,200 \times 5}{60\,000 \times 0,823} = 85 \text{ W}$$

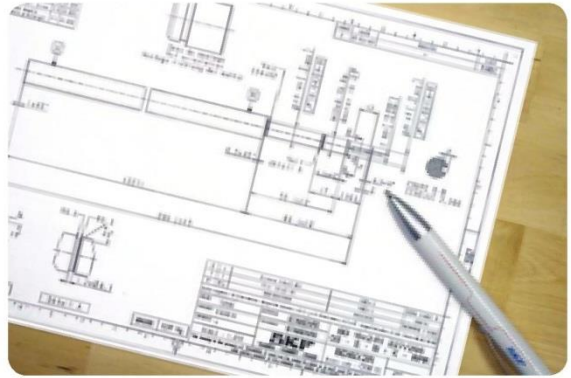
$$\text{Phase 3: } P = \frac{3\,000 \times 1\,200 \times 5}{60\,000 \times 0,823} = 243 \text{ W}$$

E Service range

Service range

Quick service for precision rolled ball screws

For reduced delivery time, SKF operates quick service facilities in Europe and in North America, where standard screw shafts, nuts and accessories are stocked.



Ball screw orders

Customers can order ball screws with the following options:

- Stock items of screws shafts and nuts, without machined ends. Nuts with axial play mounted on screw shaft, or on sleeve. Nuts with backlash elimination or with preload mounted on shaft
- Ball screw assemblies with shaft ends machined according to standard ends as defined in this catalogue
- Ball screw assemblies with shaft ends machined according to customer requirements: In this case, please send a drawing with all dimensional and tolerance requirements, and with all specifications written in English
- Complete ball screw assemblies, including accessories presented in this catalogue. Accessories already mounted on nut or shaft, or delivered separately.

General rules

Delivery time

- From a couple of days to maximum two weeks is possible for orders fulfilling the following conditions

Quantity

- Maximum 5 pieces for types SX/BX – SND/BND/PND – SN/BN/PN – SL/TL – SLD/TLD
- Maximum 15 pieces for types SD/BD/SH

Materials

- Both shaft and nut should be made of standard steel, as stated in the present catalogue

Capabilities

- Standard nuts, including DIN nuts
- Screw shafts machined according to customer drawing
- Backlash elimination by oversized balls available for BD – BX – BND/BN
- Preload available for PND/PN – TL/TLD
- General precision on tolerances ISO IT7 (ISO 3408-3:2006)
- One nut per screw shaft

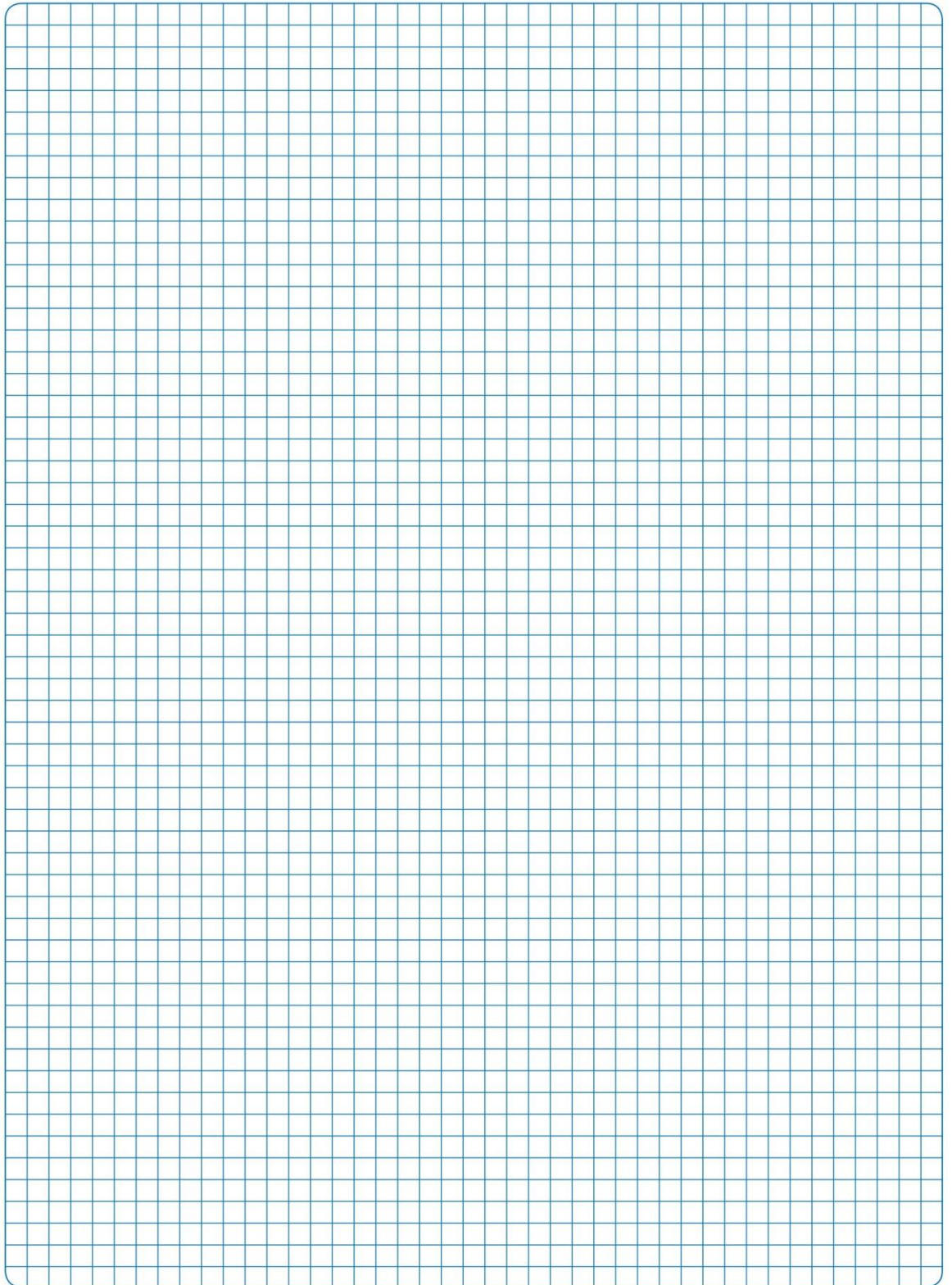
Other conditions for quick delivery

- Rotating nut SLT/TLT types are excluded from this program
- Stainless steel or special treatments, including annealed shaft ends, splines are excluded from this program
- Material certificates, special reports, or orders that require special procedure and approval by the French authorities are excluded for this program.

Available range

Diameter	Lead	Nut types	Lead precision	Accessories
From 6 to 63 mm	From 2 to 50 mm	Cylindrical and flanged nuts with axial play, backlash elimination, or preload, SKF designs or DIN designs	G5 – G7 – G9	Flanges for nuts, and ball screw support bearings

Notes



E

Design calculation and inquiry form

Customer and project information

Company name

Address

Contact name Phone number

Email Website

Project name

Application type

Short description of application
(please attach a sketch if possible)

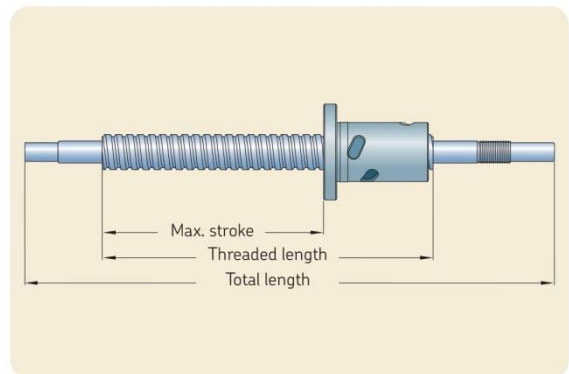
Annual ball screw requirements
and start of production date

Prototype requirements
and suitable delivery date

For existing or modified application,
type of ball screw already used

Ball screw data

Design parameter	Value
Maximum stroke [mm]
Threaded length [mm]
Total length [mm]
Pre-selection of screw shaft nominal diameter d_0 [mm]
Pre-selection of lead P_h [mm]
Pre-selection of nut type
Lead precision grade according to ISO 3408
Pre-selection of axial play, backlash elimination or preload
If axial play is selected, preferred min/max range [μm]
Request for accessories (flanges, support bearings, etc.)
Other pertinent information



Operating conditions

Maximum loads	• Maximum static load or shock load [N]
	• Maximum dynamic load in tension [N]
	• Maximum dynamic load in compression [N]
	• Average linear speed [m/min]
	• Maximum linear speed [m/min]
	• Maximum acceleration [m/s ²]
Lubrication	• Brand name
	• Type
	• Viscosity at average operating temperature [Cst]
Operating temperature	• Minimum [°C]
	• Average [°C]
	• Maximum [°C]
Required service life	• Travel [m]
	• Or revolutions [rev]
	• Or duration [hours]

Duty cycle description

Step	Axial force [N]	Speed, either rotational speed [rpm] or linear speed [m/minute]	Travel [mm]
1
2
3
4
5
Etc.

F

Mounting conditions

Position of the screw	<input type="checkbox"/> Vertical	<input type="checkbox"/> Horizontal
Rotating part	<input type="checkbox"/> Screw	<input type="checkbox"/> Nut
Screw end fixing conditions	<input type="checkbox"/>  <input type="checkbox"/>  <input type="checkbox"/> 	 (fixed, free) (fixed, radial support) (fixed, fixed)

Other pertinent information

.....

.....

.....

.....

Ball screw catalogue, ball screw 2D prints and 3D models are available on www.skf.com
 Please send inquiry form to your SKF sales office. For contact information, please visit www.skf.com

Designation system

Complete rolled ball screw assembly

	SN	32x5 R	330/445	G7	L - HA + K	**/**	WPR
Nut type							
SD = Miniature screw, axial play, recirculation by inserts							
BD = Miniature screw, backlash elimination, recirculation by inserts							
SH = Miniature screw, axial play, recirculation by integrated tube							
SDS = Miniature screw, axial play, stainless steel							
BDS = Miniature screw, backlash elimination, stainless steel							
SHS = Miniature screw, axial play, stainless steel, recirculation by integrated tube							
SX = Universal screw, axial play							
BX = Universal screw, backlash elimination							
SND = Precision screw, axial play, DIN nut							
BND = Precision screw, backlash elimination, DIN nut							
PND = Precision screw, preloaded, DIN nut							
SN = Precision screw, axial play, cylindrical flange							
BN = Precision screw, backlash elimination, cylindrical flange							
PN = Precision screw, preloaded, cylindrical flange							
SL = Long lead screw, axial play							
TL = Long lead screw, backlash elimination							
SLD = Long lead screw, axial play, DIN nut							
TLD = Long lead screw, backlash elimination, DIN nut							
SLT = Rotating nut, axial play							
TLT = Rotating nut, backlash elimination							
Nominal diameter × Lead [mm]							
Hand							
R = Right							
L = Left (on request)							
Threaded length / Total length [mm]							
Lead precision G5, G7, G9							
Nut orientation							
Nut threaded nose or nut flange towards shorter machined end of shaft (S)							
Nut threaded nose or nut flange towards longer machined end of shaft (L)							
In case of identical machining at both shaft ends (-)							
Machined end combination							
See page 36							
Required lengths for AA, SA, UA (both ends)							
See page 36							
Options							
WPR = with wipers							
NOWPR = without wipers							
RING = safety ring (for miniature ball screws only)							
REDPLAY = reduced axial play							

Roller screws, electromechanical cylinders and guiding solutions



Planetary roller screws

The robust screws for long service life through tough conditions
 $d_0 = 8$ to 240 mm
 $P_n = 2$ to 50 mm
 High load carrying capacity
 Ability to survive occasional shock loads
 High reliability, even in hostile environment
 High rotational speed capability
 Beyond the capabilities of ball screws for ultimate driving performance.



Recirculating roller screws

The fine screws for ultimate positioning accuracy
 $d_0 = 8$ to 125 mm
 $P_n = 0,6$ to 5 mm
 Fine resolution for high precision
 High rigidity
 Ideal combination of small lead, high load carrying capacity and axial stiffness for ultra-precision driving solutions.



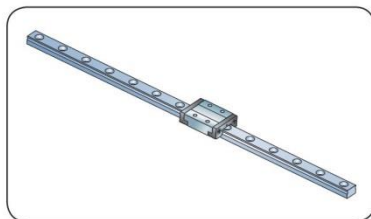
Electromechanical cylinders (EMC and CEMC types)

The high performance electromechanical cylinders incorporate SKF planetary roller screws driven by brushless motors
 Dynamic load capacity up to 450 kN
 Linear speed up to 1,6 m/s
 They are designed for long service life, high acceleration, high force applications and heavy duty cycles
 Compact Electromechanical cylinders (CEMC) deliver a unique combination of design flexibility and powerful actuation within a compact package
 SKF electromechanical cylinders using roller screws are expanding the limits of linear actuators.



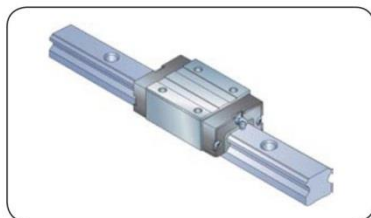
Linear ball bearings

The cost-efficient guiding solution
 Dimensions according to ISO 10285 series 1 and 3, from nominal diameter 5 mm to 80 mm
 Closed and open designs, rigid and self-aligning designs. Integrated lubricant reservoir. Stainless steel option
 Wide range of accessories stocked at SKF such as linear ball bearing housings, ground shafts, shaft supports and shaft blocks
 Load carrying capacity and sealing performance optimized for maximum service life.



Miniature profile rail guides

The compact high precision profile rail guide for space saving and reduced machine weight
 Sizes 7 to 15 with interchangeability to DIN 645-2
 Standard width rail and larger rail for increased moment carrying capacity
 High precision-ground raceways and carriages with two rows of balls offering high load carrying capacity in all four radial directions
 Stainless steel components and factory pre-lubrication for high reliability.



Profile rail guides

The ideal guiding solution in combination with driving by precision rolled ball screws in most machinery
 Rail sizes 15 to 45 with interchangeability to DIN 645-1 and upcoming ISO 12090-1. Variety of carriage types according to industry standards
 High precision profile rail guide with precision-ground raceways and carriages, available with various classes of precision and preload
 4 bearing raceways with X-arrangement providing equal load carrying capacity in all four radial directions, and ability to resist over-turning moments
 Accessories stocked at SKF such as low friction seals, scrapper seals, bellows, lubrication systems, etc.
 Solutions adapted to various types of applications and requirements.



© SKF is a registered trademark of the SKF Group.

© SKF Group 2013

The contents of this publication are the copyright of the publisher and may not be reproduced (even extracts) unless prior written permission is granted. Every care has been taken to ensure the accuracy of the information contained in this publication but no liability can be accepted for any loss or damage whether direct, indirect or consequential arising out of the use of the information contained herein.

Previous catalogues, in which the data deviate from those given here, are rendered invalid. The right is reserved to make changes necessitated by technological developments.

PUB MT/P1 06971/1 EN - August 2013

Certain images used under license from Shutterstock.com

www.skf.com

Detailed Specifications

For user manuals and dimensional drawings, visit the product page resources tab on ni.com.

Last Revised: 2014-12-17 16:19:17.0

Stepper Motors and Encoders



Overview

National Instruments offers a complete stepper motion control solution – including stepper motors, drives, controllers, and software – that is easy to set up, configure, and program. Stepper motors available from NI offer high torque, precision, and easy connectivity to stepper motor drives. Due to their ease of use, simplified control needs, and freedom from expensive feedback requirements, stepper motors are an excellent solution for applications such as machine control, manufacturing test, semiconductor positioning, biomedical machines, and lab automation.

[Back to Top](#)

Application and Technology

Stepper Motors

- NEMA 17, 23, and 34 frame sizes
- Up to 1710 oz-in. (12.1 N · m) holding torque
- 3000 rpm max speed
- 1.8 deg step angle
- Matched with P7000 drives for high performance

Encoders

- 1000 counts/revolution resolution
- NEMA 23 and 34 motor compatibility
- Low profile 1 in. (25.4 mm) height design and easy mounting
- Industrial construction

Hardware

Stepper motors provide very precise, extremely cost-effective motion control. The 2-phase motors inherently move in small, precise, 1.8 degree increments at 200 steps/revolution and are brushless and maintenance-free. Stepping action is simple to control and does not require complicated, expensive feedback devices. National Instruments also offers encoders matched to the motors for applications where position verification is required. Stepper motors are available from NI in three different National Electrical Manufacturers Association (NEMA) frame sizes and with either a single or a dual shaft. The motors provide optimum performance and easy connectivity when matched with the P7000 series stepper drives available from NI.

[Back to Top](#)

Support and Services

System Assurance Programs

NI system assurance programs are designed to make it even easier for you to own an NI system. These programs include configuration and deployment services for your NI PXI, CompactRIO, or Compact FieldPoint system. The NI Basic System Assurance Program provides a simple integration test and ensures that your system is delivered completely assembled in one box. When you configure your system with the NI Standard System Assurance Program, you can select from available NI system driver sets and application development environments to create customized, reorderable software configurations. Your system arrives fully assembled and tested in one box with your software preinstalled. When you order your system with the standard program, you also receive system-specific documentation including a bill of materials, an integration test report, a recommended maintenance plan, and frequently asked question documents. Finally, the standard program reduces the total cost of owning an NI system by providing three years of warranty coverage and calibration service. Use the online product advisors at ni.com/advisor to find a system assurance program to meet your needs.

Calibration

NI measurement hardware is calibrated to ensure measurement accuracy and verify that the device meets its published specifications. To ensure the ongoing accuracy of your measurement hardware, NI offers basic or detailed recalibration service that provides ongoing ISO 9001 audit compliance and confidence in your measurements. To learn more about NI calibration services or to locate a qualified service center near you, contact your local sales office or visit ni.com/calibration.

Technical Support

Get answers to your technical questions using the following National Instruments resources.

- **Support** - Visit ni.com/support to access the NI KnowledgeBase, example programs, and tutorials or to contact our applications engineers who are located in NI sales offices around the world and speak the local language.
- **Discussion Forums** - Visit forums.ni.com for a diverse set of discussion boards on topics you care about.
- **Online Community** - Visit community.ni.com to find, contribute, or collaborate on customer-contributed technical content with users like you.

Repair

While you may never need your hardware repaired, NI understands that unexpected events may lead to necessary repairs. NI offers repair services performed by highly trained technicians who quickly return your device with the guarantee that it will perform to factory specifications. For more information, visit ni.com/repair.

Training and Certifications

The NI training and certification program delivers the fastest, most certain route to increased proficiency and productivity using NI software and hardware. Training builds the skills to more efficiently develop robust, maintainable applications, while certification validates your knowledge and ability.

- **Classroom training in cities worldwide** - the most comprehensive hands-on training taught by engineers.
- **On-site training at your facility** - an excellent option to train multiple employees at the same time.
- **Online instructor-led training** - lower-cost, remote training if classroom or on-site courses are not possible.
- **Course kits** - lowest-cost, self-paced training that you can use as reference guides.
- **Training memberships** and training credits - to buy now and schedule training later.

Visit ni.com/training for more information.

Extended Warranty

NI offers options for extending the standard product warranty to meet the life-cycle requirements of your project. In addition, because NI understands that your requirements may change, the extended warranty is flexible in length and easily renewed. For more information, visit ni.com/warranty.

OEM

NI offers design-in consulting and product integration assistance if you need NI products for OEM applications. For information about special pricing and services for OEM customers, visit ni.com/oem.

Alliance

Our Professional Services Team is comprised of NI applications engineers, NI Consulting Services, and a worldwide National Instruments Alliance Partner program of more than 700 independent consultants and integrators. Services range from start-up assistance to turnkey system integration. Visit ni.com/alliance.

[Back to Top](#)

Detailed Specifications

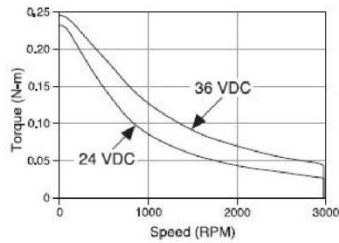
NEMA 17 Motor	
Electrical	
Step angle	1.8 deg
Steps per revolution	200
Angular accuracy	±3%
Phases	2
Industry Standards	
Industrial standards	CE, UR
Sealing standards	IP40
RoHS Compliance	Yes
Physical	
Operating temperature	-20 to 40 °C
Shaft load (20,000 hours at 1,500 rpm)	15 lb (6.8 kg) at shaft center

Radial	
Axial push	6 lb (2.7 kg)
Axial pull	15 lb (6.8 kg)
Recommended heat sink size	10 x 10 x 1/4 in. aluminum plate

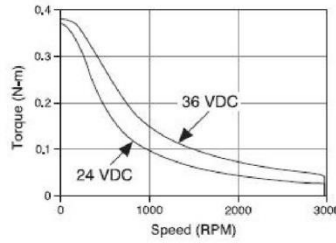
NI Part Number	Manufacturer Part Number	Dual Shaft	Drive	Amps/Phase	Holding Torque oz-in. (N . m)	Rotor Inertia oz-in.-s ² (kg-m ² x 10 ⁻³)	Phase Inductance mH	Phase Resistance Ω ±10%	Detent Torque oz-in. (N . m)	Thermal Resistance °C/watt	Max Speed rpm
780067-01	CTP10ELF10MAA00	no	P70530	1.0	43 (0.30)	0.0005 (0.0040)	7.7	5.25	1.98 (0.014)	6.21	3000
780068-01	CTP10ELF10MMA00	yes									
780069-01	CTP11ELF11MAA00	no		1.1	63 (0.44)	0.0008 (0.0050)	11	5.19	2.55 (0.018)	5.44	
780070-01	CTP11ELF11MMA00	yes									
780071-01	CTP12ELF10MAA00	no		1.0	80 (0.56)	0.0011 (0.0070)	12	6.51	2.97 (0.021)	4.71	
780072-01	CTP12ELF11MAA00	yes									

Torque versus Speed

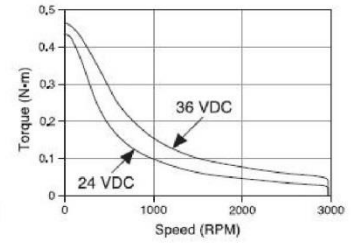
780067-01 and 780068-01
Torque versus Speed at 1.0 A



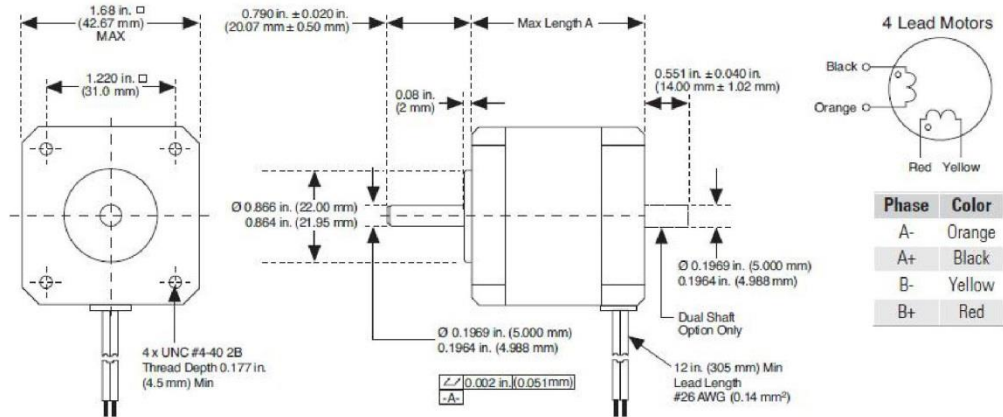
780069-01 and 780070-01
Torque versus Speed at 1.1 A



780071-01 and 780072-01
Torque versus Speed at 1.0 A



Dimensions and Wiring



NI Part Number	Manufacturer Part Number	Dual Shaft	Max Length A in. (mm)	Net Weight lb (kg)
780067-01	CTP10ELF10MAA00	no	1.37 (34.7)	0.441 (0.200)
780068-01	CTP10ELF10MMA00	yes		
780069-01	CTP11ELF11MAA00	no	1.61 (40.9)	0.573 (0.260)
780070-01	CTP11ELF11MMA00	yes		
780071-01	CTP12ELF10MAA00	no	1.92 (48.8)	0.750 (0.340)
780072-01	CTP12ELF11MAA00	yes		

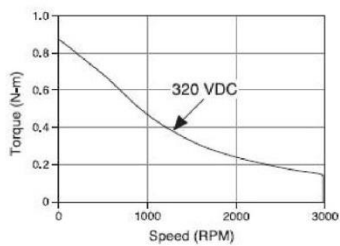
NEMA 23 Motor

Electrical	
Step angle	1.8 deg
Steps per revolution	200
Angular accuracy	±3%
Phases	2
Industry Standards	
Industrial standards	CE, cUR, UR
RoHS Compliance	Yes
Physical	
Operating temperature	-20 to 40 °C
Rated ambient temperature	40 °C
Shaft load (20,000 hours at 1,500 rpm)	
Radial	20 lb (9.1 kg) at shaft center
Axial push	6 lb (2.7 kg)
Axial pull	50 lb (22.7 kg)
Recommended heat sink size	10 x 10 x 1/4 in. aluminum plate
Recommended encoder	780251-01

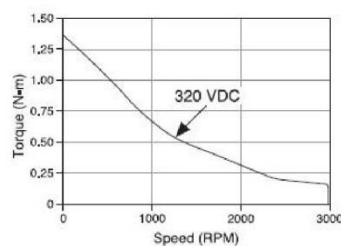
NI Part Number	Manufacturer Part Number	Dual Shaft	Drive	Amps/Phase	Holding Torque oz-in. (N . m)	Rotor Inertia oz-in.-s ² (kg-m ² x10 ⁻³)	Phase Inductance mH	Phase Resistance Ω ±10%	Detent Torque oz-in. (N . m)	Thermal Resistance°C/watt	Max Speed rpm	
780073-01	T21NRLC-LNN-NS-00	no	P70360	0.40	180 (1.27)	0.0034 (0.0248)	209	42.9	2.97 (0.021)	4.64	3000	
780074-01	T21NRLC-LDN-NS-00	yes										
780075-01	T22NRLC-LNN-NS-00	no										
780076-01	T22NRLC-LDN-NS-00	yes		0.46	280 (1.98)	0.0056 (0.0408)	209	41.4	5.95 (0.042)	3.69		
780077-01	T23NRLC-LNN-NS-00	no										
780078-01	T23NRLC-LDN-NS-00	yes										
780079-01	T21NRLH-LNN-NS-00	no	P70530	2.7	180 (1.27)	0.0034 (0.0248)	4.6	0.85	2.97 (0.021)	4.64		
780080-01	T21NRLH-LDN-NS-00	yes										
780081-01	T22NRLG-LNN-NS-00	no										
780082-01	T22NRLG-LDN-NS-00	yes		2.5	280 (1.98)	0.0056 (0.0408)	7.1	1.23	5.95 (0.042)	3.69		
780083-01	T23NRLH-LNN-NS-00	no										
780084-01	T23NRLH-LDN-NS-00	yes										

Torque versus Speed

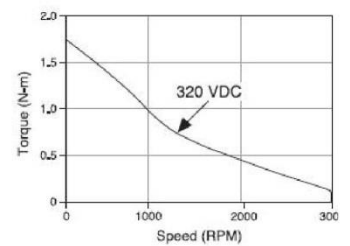
780073-01 and 780074-01
Torque versus Speed at 0.36 A



780075-01 and 780076-01
Torque versus Speed at 0.40 A



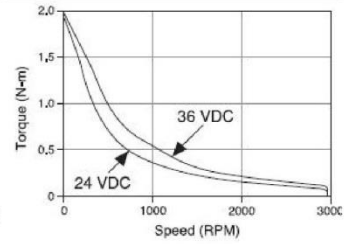
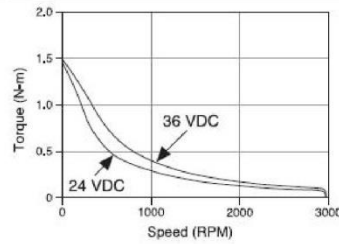
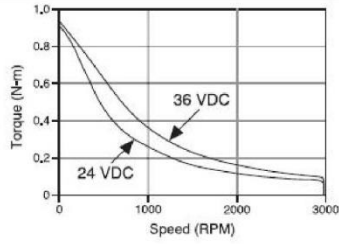
780077-01 and 780078-01
Torque versus Speed at 0.52 A



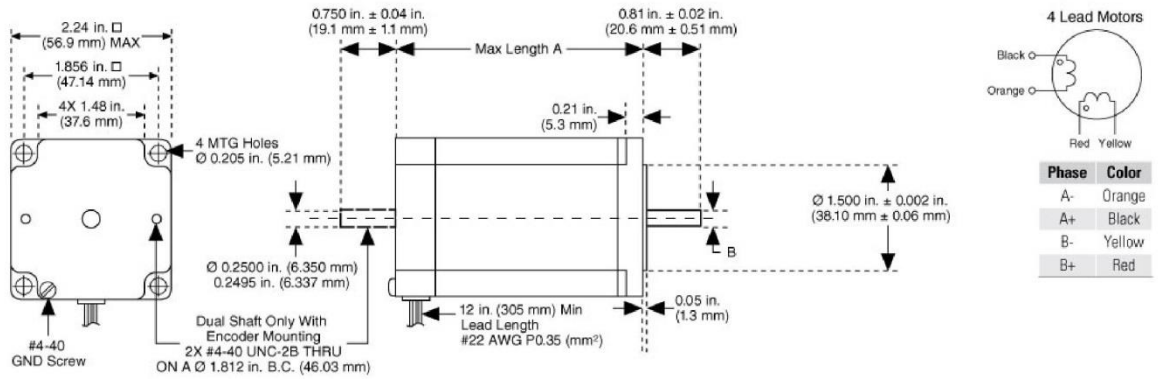
780079-01 and 780080-01
Torque versus Speed at 2.7 A

780081-01 and 780082-01
Torque versus Speed at 2.5 A

780083-01 and 780084-01
Torque versus Speed at 3.0 A



Dimensions and Wiring



NI Part Number	Manufacturer Part Number	Dual Shaft	Max Length A in. (mm)	B Diameter in. (mm)	Net Weight lb (kg)
780073-01	T21NRLC-LNN-NS-00	no	2.21 (56.1)	0.2500 (6.350)	1.6 (0.7)
780074-01	T21NRLC-LDN-NS-00	yes	(56.1)	0.2495 (6.337)	(0.7)
780075-01	T22NRLC-LNN-NS-00	no	3.06 (77.7)	0.2500 (6.350)	2.3 (1.0)
780076-01	T22NRLC-LDN-NS-00	yes	(77.7)	0.2495 (6.337)	(1.0)
780077-01	T23NRLC-LNN-NS-00	no	4.06 (103.1)	0.313 (7.940)	3.2 (1.5)
780078-01	T23NRLC-LDN-NS-00	yes	(103.1)	0.313 (7.940)	(1.5)
780079-01	T21NRLH-LNN-NS-00	no	2.21 (56.1)	0.2500 (6.350)	1.6 (0.7)
780080-01	T21NRLH-LDN-NS-00	yes	(56.1)	0.2495 (6.337)	(0.7)
780081-01	T22NRLG-LNN-NS-00	no	3.06 (77.7)	0.2500 (6.350)	2.3 (1.0)
780082-01	T22NRLG-LDN-NS00	yes	(77.7)	0.2495 (6.337)	(1.0)
780083-01	T23NRLH-LNN-NS00	no	4.06 (103.1)	0.313 (7.940)	3.2 (1.5)
780084-01	T23NRLH-LDN-NS00	yes	(103.1)	0.313 (7.940)	(1.5)

NEMA 34 Motor

Electrical

Step angle	1.8 deg
Steps per revolution	200
Angular accuracy	±3%
Phases	2

Industry Standards

Industrial standards	CE, cUR, UR
RoHS Compliance	Yes

Physical

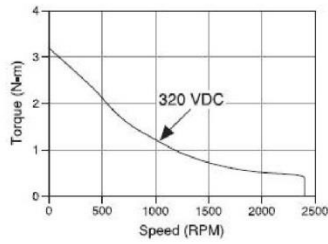
Operating temperature	-20 to 40 °C
-----------------------	--------------

Rated ambient temperature	40 °C
Shaft load (20,000 hours at 1,500 rpm)	
Radial	
N31, N32	65 lb (29.5 kg)
N33	110 lb (49.9 kg)
Axial	
N31, N32, N33	305 lb (138.3 kg)
Recommended heat sink size	10 x 10 x 1/4 in. aluminum plate
Recommended encoder	780252-01

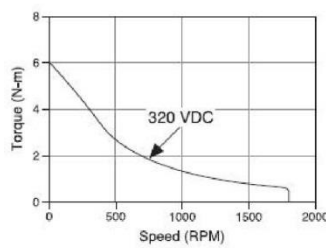
NI Part Number	Manufacturer Part Number	Dual Shaft	Drive	Amps/Phase	Holding Torque oz-in. (N . m)	Rotor Inertia oz-in.-s ² (kg-m ² x 10 ⁻³)	Phase Inductance mH	Phase Resistance Ω ±10%	Detent Torque oz-in. (N . m)	Thermal Resistance °C/watt	Max Speed rpm
780085-01	N31HRLG-LNK-NS-00	no	P70360	0.86	641 (4.52)	0.0202 (0.1430)	138	16.2	18.0 (0.127)	2.65	2400
780086-01	N31HRLG-LEK-M2-00	yes									
780087-01	N32HRLG-LNK-NS-0	no		1.24	1710 (12.08)	0.0567 (0.4000)	144	13.0	54.0 (0.381)	1.61	1800
780088-01	N32HRLG-LEK-M2-00	yes									
780089-01	N33HRLG-LNK-NS-0	no									
780090-01	N33HRLG-LEK-M2-00	yes	P70530	5.5	645 (4.55)	0.0202 (0.1430)	3.5	0.42	18.0 (0.127)	2.65	3000
780091-01	N31HRHJ-LNK-NS-0	no									
780092-01	N31HRHJ-LEK-M2-0	yes		5.1	1195 (8.43)	0.0380 (0.2700)	6.5	0.63	36.0 (0.254)	2.00	3000
780093-01	N32HRHJ-LNK-NS-0	no									
780094-01	N32HRHJ-LEK-M2-0	yes									
780095-01	N33HRHJ-LNK-NS-0	no	5.0	1710 (12.07)	0.0567 (0.4000)	9.0	0.83	54.0 (0.381)	1.61	3000	
780096-01	N33HRHJ-LEK-M2-0	yes									

Torque versus Speed

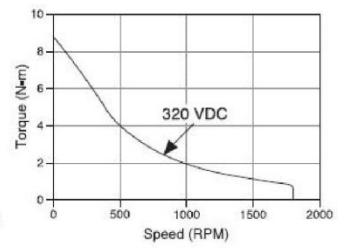
780085-01 and 780086-01
Torque versus Speed at 0.81 A



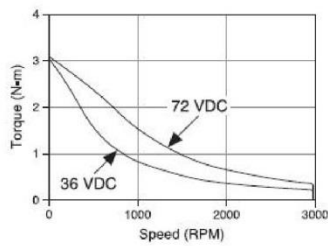
780087-01 and 780088-01
Torque versus Speed at 0.88 A



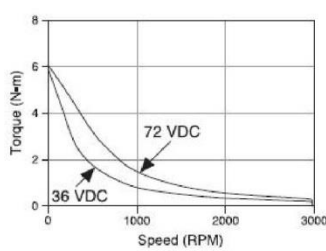
780089-01 and 780090-01
Torque versus Speed at 1.24 A



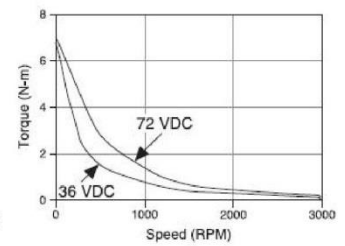
780091-01 and 780092-01
Torque versus Speed at 5 A



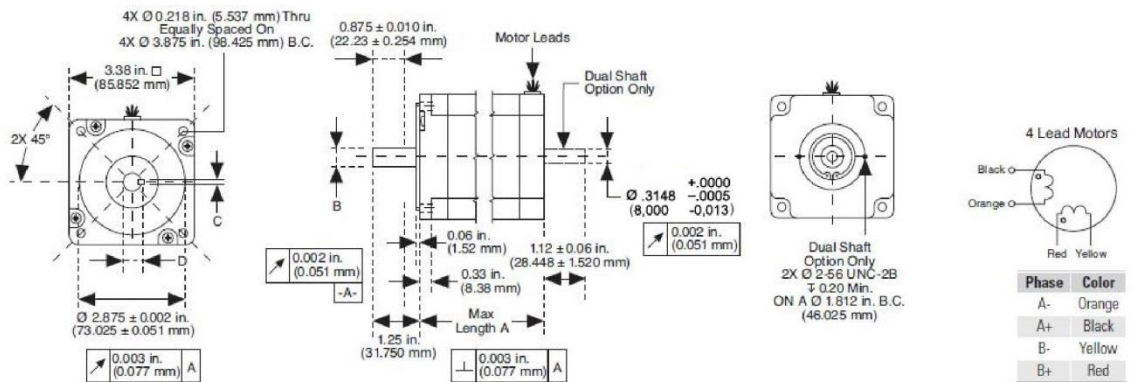
780093-01 and 780094-01
Torque versus Speed at 5 A



780095-01 and 780096-01
Torque versus Speed at 5 A



Dimensions and Wiring



Note: Motor leads are 12.0 in. (304.8 mm) minimum.

NI Part Number	Manufacturer Part Number	Dual Shaft	Max Length A in. (mm)	B max and min in. (mm)	C max and min in. (mm)	D max and min in. (mm)	Net Weight lb (kg)
780085-01	N31HRLG-LNK-NS-00	no	3.13 (79.502)	0.5000 (12.700) 0.4995 (12.687)	0.1250 (3.175) 0.1230 (3.124)	0.555 (14.097) 0.538 (13.665)	5.0 (2.27)
780086-01	N31HRLG-LEK-M2-00	yes	4.65 (118.11)	0.5000 (12.700) 0.4995 (12.687)	0.1250 (3.175) 0.1230 (3.124)	0.555 (14.097) 0.538 (13.665)	8.4 (2.27)
780087-01	N32HRLG-LNK-NS-00	no	6.13 (155.70)	0.6250 (15.875) 0.6245 (15.862)	0.1875 (4.763) 0.1855 (4.712)	0.705 (17.907) 0.688 (17.475)	11.9 (5.39)
780088-01	N32HRLG-LEK-M2-00	yes	3.13 (79.502)	0.5000 (12.700) 0.4995 (12.687)	0.1250 (3.175) 0.1230 (3.124)	0.555 (14.097) 0.538 (13.665)	5.0 (2.27)
780089-01	N33HRLG-LNK-NS-00	no	4.65 (118.11)	0.5000 (12.700) 0.4995 (12.687)	0.1250 (3.175) 0.1230 (3.124)	0.555 (14.097) 0.538 (13.665)	8.4 (2.27)
780090-01	N33HRLG-LEK-M2-00	yes	6.13 (155.70)	0.6250 (15.875) 0.6245 (15.862)	0.1875 (4.763) 0.1855 (4.712)	0.705 (17.907) 0.688 (17.475)	11.9 (5.39)
780091-01	N31HRHJ-LNK-NS-00	no	3.13 (79.502)	0.5000 (12.700) 0.4995 (12.687)	0.1250 (3.175) 0.1230 (3.124)	0.555 (14.097) 0.538 (13.665)	5.0 (2.27)
780092-01	N31HRHJ-LEK-M2-00	yes	4.65 (118.11)	0.5000 (12.700) 0.4995 (12.687)	0.1250 (3.175) 0.1230 (3.124)	0.555 (14.097) 0.538 (13.665)	8.4 (2.27)
780093-01	N32HRHJ-LNK-NS-00	no	6.13 (155.70)	0.6250 (15.875) 0.6245 (15.862)	0.1875 (4.763) 0.1855 (4.712)	0.705 (17.907) 0.688 (17.475)	11.9 (5.39)
780094-01	N32HRHJ-LEK-M2-00	yes	3.13 (79.502)	0.5000 (12.700) 0.4995 (12.687)	0.1250 (3.175) 0.1230 (3.124)	0.555 (14.097) 0.538 (13.665)	5.0 (2.27)
780095-01	N33HRHJ-LNK-NS-00	no	4.65 (118.11)	0.5000 (12.700) 0.4995 (12.687)	0.1250 (3.175) 0.1230 (3.124)	0.555 (14.097) 0.538 (13.665)	8.4 (2.27)
780096-01	N33HRHJ-LEK-M2-00	yes	6.13 (155.70)	0.6250 (15.875) 0.6245 (15.862)	0.1875 (4.763) 0.1855 (4.712)	0.705 (17.907) 0.688 (17.475)	11.9 (5.39)

Encoders for NEMA 23 and NEMA 34 Motors

Electrical

Resolution	1000 counts/revolution
Input voltage	5 V ±10%
Input current	100 mA max (65 mA typical) with no output load
Channel configuration	Quadrature A, B, and Index
Output type	Differential line driver
Noise immunity	Tested to BS EN61000-6-2; BS EN50081-02; BS EN61000-4-2; BS EN61000-4-3; BS EN61000-4-6; BS EN500811
Symmetry	180 deg (±18 deg) electrical
Quadrature phasing	90 deg (±22.5 deg) electrical
Minimum edge separation	67.5 deg electrical
Accuracy	Within 0.017 deg mechanical or 1 arc-minute from true position

Industry Standards

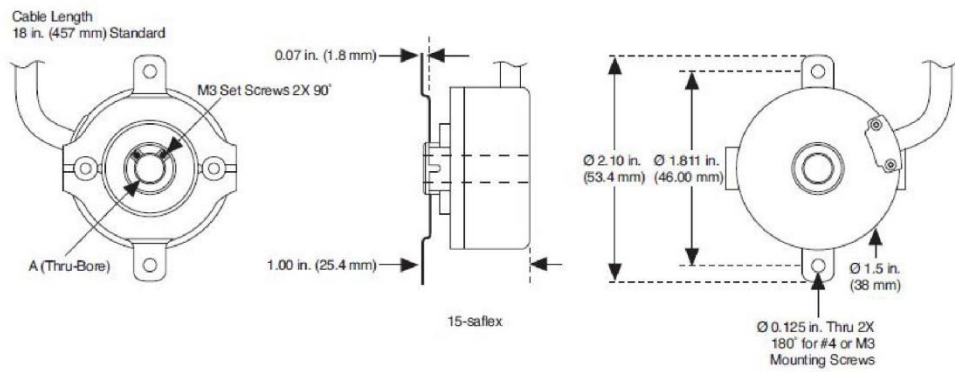
Industrial standards	CE
Sealing standards	IP40
RoHS Compliance	Yes

Physical

Operating temperature	-20 to 85 °C
Model type	Thru-bore
Bore size	1/4 in. (780251-01), 8 mm (780252-01)
Mounting	1.812 in. (46 mm) two-hole flex mount
Maximum frequency	200 kHz
Operating temperature	20 to 85 °C
Max shaft speed	8000 rpm

Bore tolerance	-0.0000 in./+0.0006 in.
User shaft tolerances	
Radial runout	0.008 in. max
Axial endplay	±0.030 in. max
Starting torque	0.300 oz-in. (0.212 N . m)
Moment of inertia	6.7×10^{-5} oz-in.-sec ² (4.8 gm-cm ²)
Max acceleration	1×10^5 rad/sec ²
Weight	3 oz typical
Storage temperature	-25 to 85 °C
Humidity	98% RH noncondensing
Vibration	10 g @ 58 to 500 Hz
Shock	80 g @ 11 ms duration

Dimensions, Wiring and Timing Diagrams



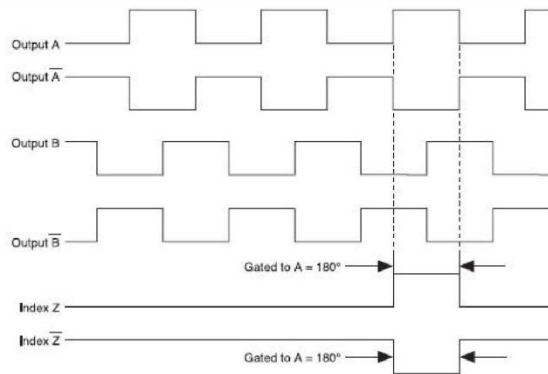
Note: All dimensions have a tolerance of ±0.005 in. or ±0.01 in. unless otherwise specified.

NI Part Number	Manufacturer Part Number	A (Thru-Bore Diameter)
780251-01	15T-01SA-1000-N5RHV-F00-CE	1/4 in., 0.250 in.
780252-02	15T-14SA-1000-N5RHV-F00-CE	8 mm

Wire Description

Pin #	Wire Color	Function
1	Brown	A
2	White	+VDC
3	Yellow	A'
4	Red	B
5	Green	B'
6	Orange	Z
7	Black	COM
8	Blue	Z'

Quadrature Waveform



Glossary

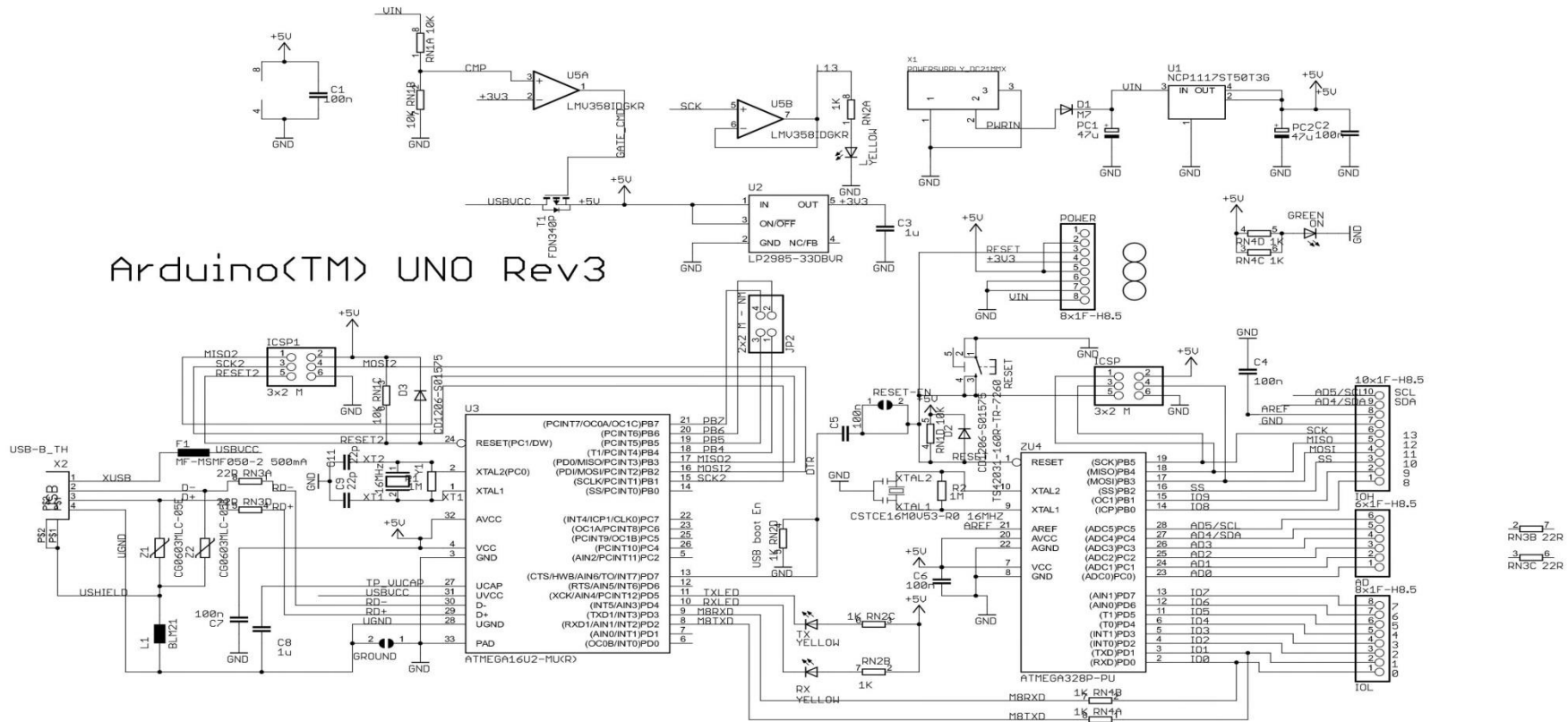
amps/phase	The maximum amount of current allowed through a phase of the stepper motor. Holding torque, the speed versus torque curve, and so on are determined when the motor is excited by this value. The specifications listed in this data sheet are adjusted for the winding configuration.
angular accuracy	A percentage of the step angle that defines the accuracy of each full step.
detent torque (cogging torque)	The amount of torque necessary to rotate the stepper motor one full step when the motor is deenergized.
differential line driver	A type of electrical digital output that can transmit digital data over a long distance. It consists of a complementary pair of digital lines.
electrical symmetry	How close each quadrature channel is to a 50 percent duty cycle when at a constant speed.
holding torque	The amount of torque necessary to rotate the stepper motor one full step (microstepping turned off) when the motor is energized at the rated amps/phase of that motor.
minimum edge separation	Defines in degrees how close (electrically) an edge on channel A can be to an edge on channel B.
NEMA	National Electrical Manufacturers Association (NEMA). NEMA is a U.S.-based association that creates standards for mountings. The NEMA size of a motor defines its shaft size and mounting configuration.
phase inductance	The inductance of each phase of the stepper motor. The specifications listed in this data sheet are already adjusted for the winding configuration.
phases	A wound wire in the stepper motor that is excited with current to produce electromagnetic force. Two or more phases work together by alternating between positively energized, deenergized, and negatively energized states to rotate the stepper motor.
quadrature phasing	The electrical phase shift between channels A and B in a quadrature encoder.
step angle	The distance the motor rotates each full step of the stepper motor. Also defined as 360 degrees divided by the steps per revolution.

[Back to Top](#)

©2010 National Instruments. All rights reserved. CompactRIO, CVI, FieldPoint, LabVIEW, Measurement Studio, National Instruments, National Instruments Alliance Partner, NI, and ni.com are trademarks of National Instruments. The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries. Other product and company names listed are trademarks or trade names of their respective companies. A National Instruments Alliance Partner is a business entity independent from National Instruments and has no agency, partnership, or joint-venture relationship with National Instruments.

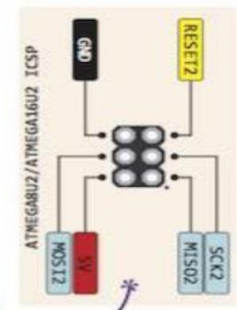
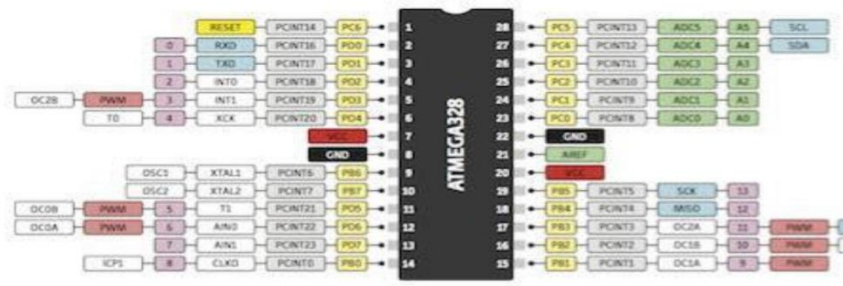
[My Profile](#) | [RSS](#) | [Privacy](#) | [Legal](#) | [Contact NI](#) © 2014 National Instruments Corporation. All rights reserved.

Arduino(TM) UNO Rev3

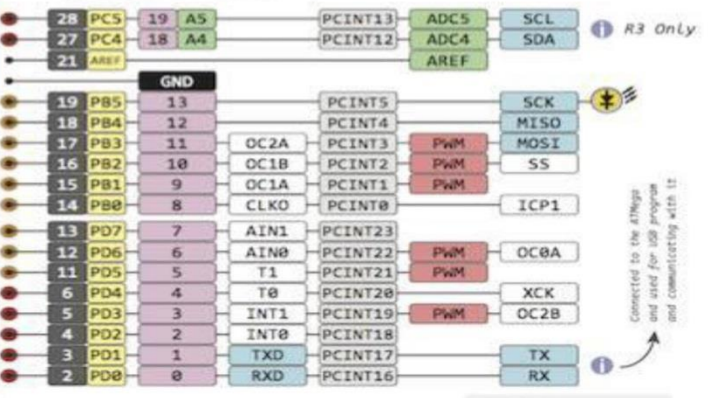
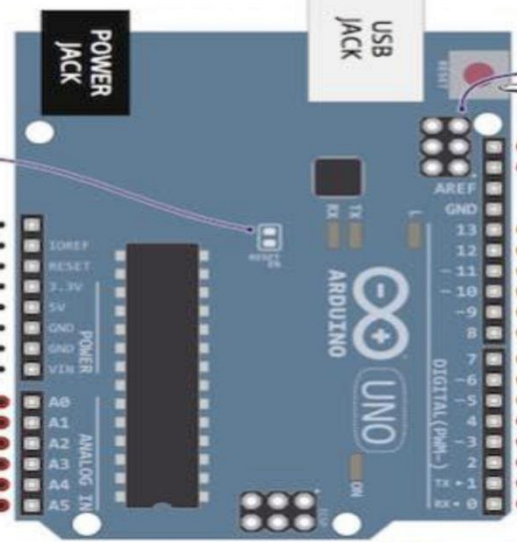
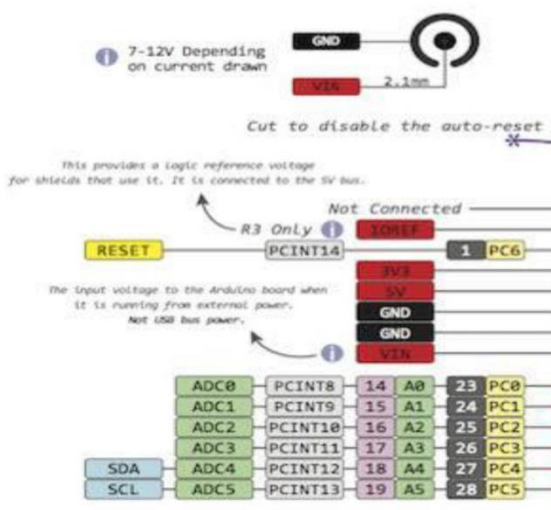


Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark. Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

THE DEFINITIVE ARDUINO UNO PINOUT DIAGRAM



⚠ Absolute max per pin 40mA recommended 20mA
 ⚡ Absolute max 200mA for entire package



www.pjrc.com
 18 FEB 2013
 ver 2 rev 2 - 05.03.2013

Legend:

- Black: GND
- Red: Power
- Green: Control
- Blue: Physical Pin
- Yellow: Port Pin
- Purple: Pin Function
- Light Blue: Digital Pin
- Light Green: Analog Related Pin
- Pink: PWM Pin
- Light Purple: Serial Pin
- Light Blue: I2C
- Red circle: Source Total 150mA

Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



8-bit AVR[®]
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash

ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P

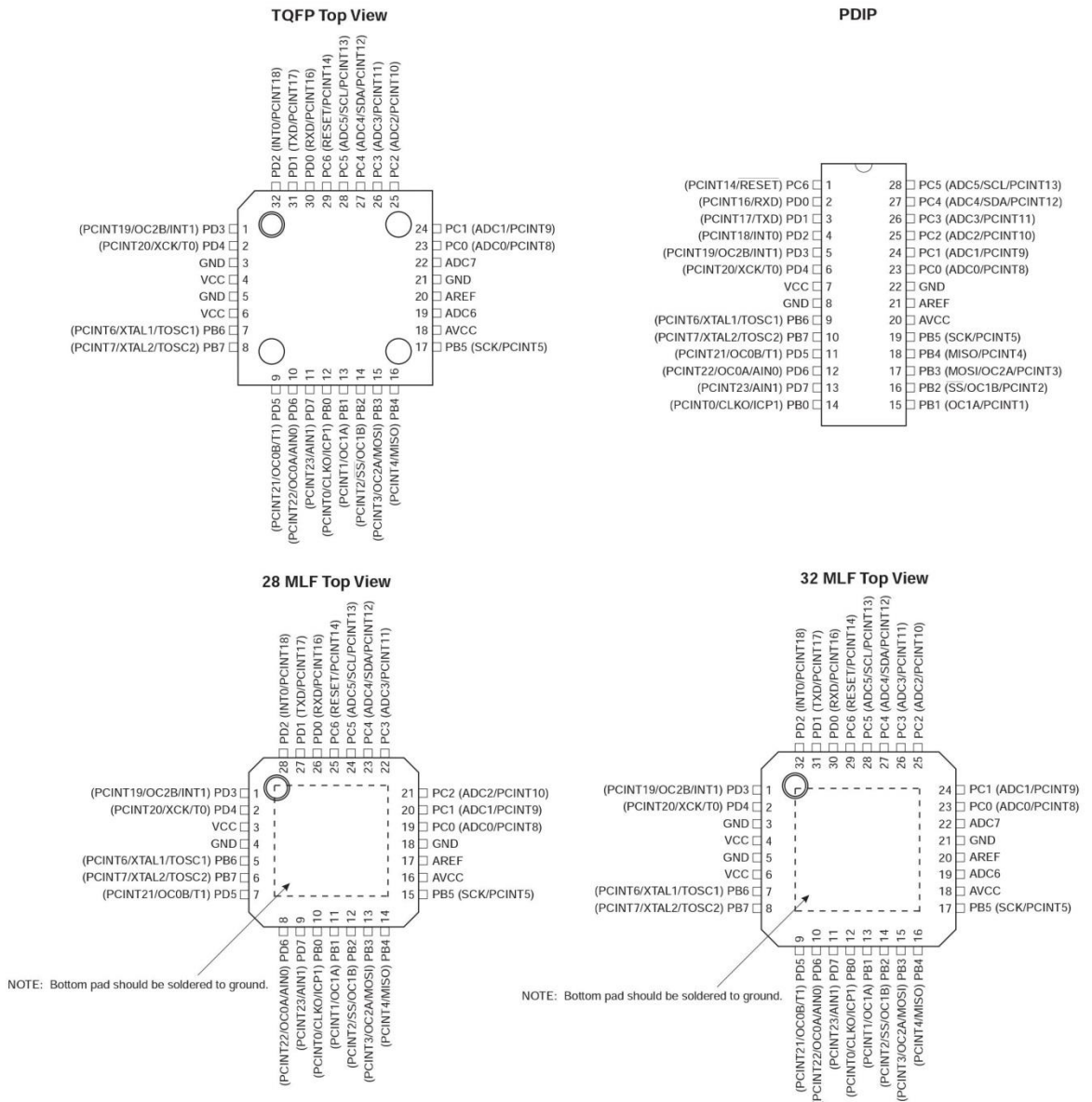
Rev. 8161D-AVR-10/09



ATmega48PA/88PA/168PA/328P

1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "[Alternate Functions of Port B](#)" on page 82 and "[System Clock and Clock Options](#)" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 28-3 on page 318](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "[Alternate Functions of Port C](#)" on page 85.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

ATmega48PA/88PA/168PA/328P

The various special features of Port D are elaborated in ["Alternate Functions of Port D"](#) on page 88.

1.1.7 **AV_{CC}**

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC}.

1.1.8 **AREF**

AREF is the analog reference pin for the A/D Converter.

1.1.9 **ADC7:6 (TQFP and QFN/MLF Package Only)**

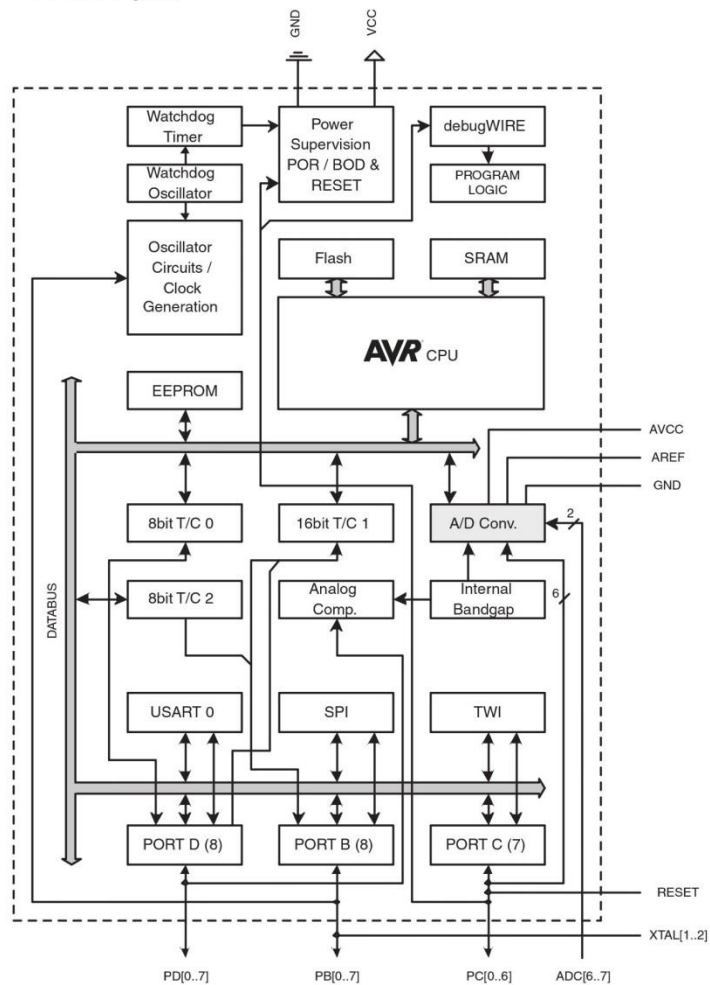
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

2. Overview

The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting

ATmega48PA/88PA/168PA/328P

architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48PA/88PA/168PA/328P provides the following features: 4K/8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48PA/88PA/168PA/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48PA/88PA/168PA/328P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

2.2 Comparison Between ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P

The ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P differ only in memory sizes, boot loader support, and interrupt vector sizes. [Table 2-1](#) summarizes the different memory and interrupt vector sizes for the three devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48PA	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88PA	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168PA	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector

ATmega88PA, ATmega168PA and ATmega328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48PA, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

4. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

5. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

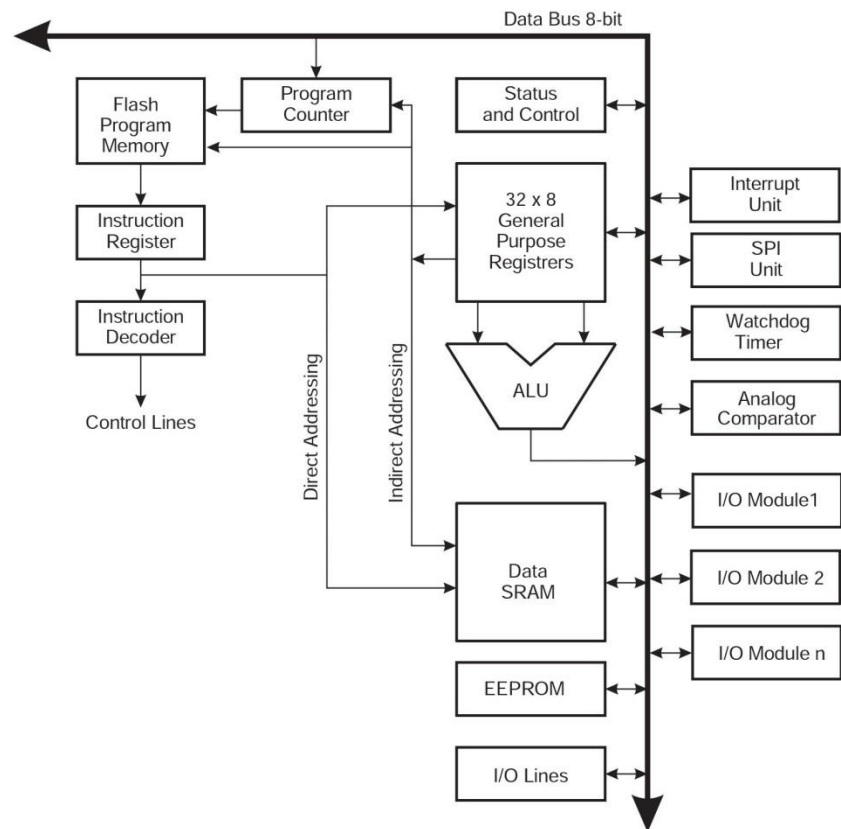
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

6. AVR CPU Core

6.1 Overview

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 6-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typ-

ical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F. In addition, the ATmega48PA/88PA/168PA/328P has Extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

6.2 ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the "Instruction Set" section for a detailed description.

6.3 Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as

specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

6.3.1 SREG – AVR Status Register

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Is useful in BCD arithmetic. See the "Instruction Set Description" for detailed information.

- **Bit 4 – S: Sign Bit, $S = N \oplus V$**

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V. See the "Instruction Set Description" for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The Two's Complement Overflow Flag V supports two's complement arithmetics. See the "Instruction Set Description" for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

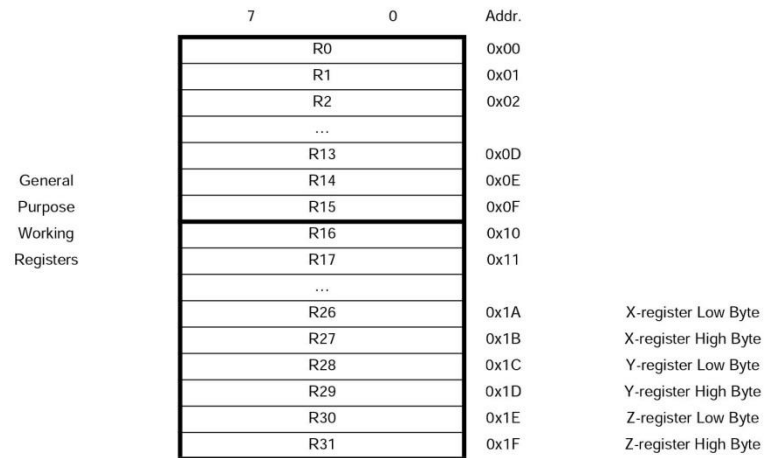
6.4 General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 6-2 shows the structure of the 32 general purpose working registers in the CPU.

Figure 6-2. AVR CPU General Purpose Working Registers



Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 6-2, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y- and Z-pointer registers can be set to index any register in the file.

6.4.1 The X-register, Y-register, and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in [Figure 6-3](#).

Figure 6-3. The X-, Y-, and Z-registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

6.5 Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. Note that the Stack is implemented as growing from higher to lower memory locations. The Stack Pointer Register always points to the top of the Stack. The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. A Stack PUSH command will decrease the Stack Pointer.

The Stack in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. Initial Stack Pointer value equals the last address of the internal SRAM and the Stack Pointer must be set to point above start of the SRAM, see [Table 7-3 on page 18](#).

See [Table 6-1](#) for Stack Pointer details.

Table 6-1. Stack Pointer instructions

Instruction	Stack pointer	Description
PUSH	Decrement by 1	Data is pushed onto the stack
CALL ICALL RCALL	Decrement by 2	Return address is pushed onto the stack with a subroutine call or interrupt
POP	Increment by 1	Data is popped from the stack
RET RETI	Increment by 2	Return address is popped from the stack with return from subroutine or return from interrupt

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

6.5.1 SPH and SPL – Stack Pointer High and Stack Pointer Low Register

Bit	15	14	13	12	11	10	9	8	
0x3E (0x5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
0x3D (0x5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	

6.6 Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock clk_{CPU} , directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 6-4 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clock, and functions per power-unit.

Figure 6-4. The Parallel Instruction Fetches and Instruction Executions

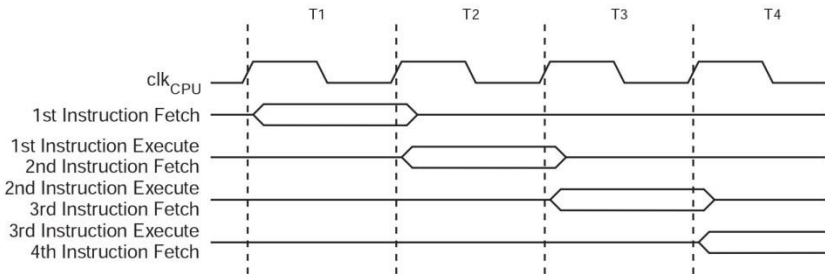
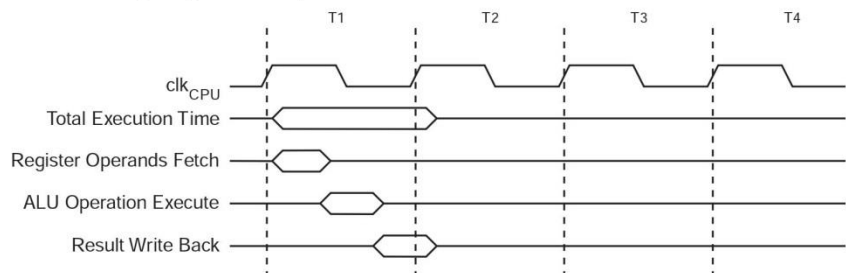


Figure 6-5 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 6-5. Single Cycle ALU Operation



6.7 Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section ["Memory Programming" on page 294](#) for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in ["Interrupts" on page 57](#). The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INTO – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the MCU Control Register (MCUCR). Refer to ["Interrupts" on page 57](#) for more information. The Reset Vector can also be moved to the start of the Boot Flash section by programming the BOOTRST Fuse, see ["Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P" on page 277](#).

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding Interrupt Flag. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding Interrupt Flag(s) will be set and remembered until the Global Interrupt Enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence.

Assembly Code Example

```

in r16, SREG      ; store SREG value
cli              ; disable interrupts during timed sequence
sbi EECR, EEMPE  ; start EEPROM write
sbi EECR, EEPE
out SREG, r16     ; restore SREG value (I-bit)
    
```

C Code Example

```

char cSREG;
cSREG = SREG; /* store SREG value */
/* disable interrupts during timed sequence */
_CLI();
EECR |= (1<<EEMPE); /* start EEPROM write */
EECR |= (1<<EEPE);
SREG = cSREG; /* restore SREG value (I-bit) */
    
```

When using the SEI instruction to enable interrupts, the instruction following SEI will be executed before any pending interrupts, as shown in this example.

Assembly Code Example

```

sei              ; set Global Interrupt Enable
sleep;           ; enter sleep, waiting for interrupt
; note: will enter sleep before any pending interrupt(s)
    
```

C Code Example

```

__enable_interrupt(); /* set Global Interrupt Enable */
__sleep(); /* enter sleep, waiting for interrupt */
/* note: will enter sleep before any pending interrupt(s) */
    
```

6.7.1 Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles the program vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the Program Counter is pushed onto the Stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (two bytes) is popped back from the Stack, the Stack Pointer is incremented by two, and the I-bit in SREG is set.

7. AVR Memories

7.1 Overview

This section describes the different memories in the ATmega48PA/88PA/168PA/328P. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega48PA/88PA/168PA/328P features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

7.2 In-System Reprogrammable Flash Program Memory

The ATmega48PA/88PA/168PA/328P contains 4/8/16/32K bytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 2/4/8/16K x 16. For software security, the Flash Program memory space is divided into two sections, Boot Loader Section and Application Program Section in ATmega88PA and ATmega168PA. See SELFPRGEN description in section "[SPMCSR – Store Program Memory Control and Status Register](#)" on page 292 for more details.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega48PA/88PA/168PA/328P Program Counter (PC) is 11/12/13/14 bits wide, thus addressing the 2/4/8/16K program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in "[Self-Programming the Flash, ATmega48PA](#)" on page 269 and "[Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P](#)" on page 277. "[Memory Programming](#)" on page 294 contains a detailed description on Flash Programming in SPI- or Parallel Programming mode.

Constant tables can be allocated within the entire program memory address space (see the LPM – Load Program Memory instruction description).

Timing diagrams for instruction fetch and execution are presented in "[Instruction Execution Timing](#)" on page 13.

Figure 7-1. Program Memory Map ATmega48PA
Program Memory

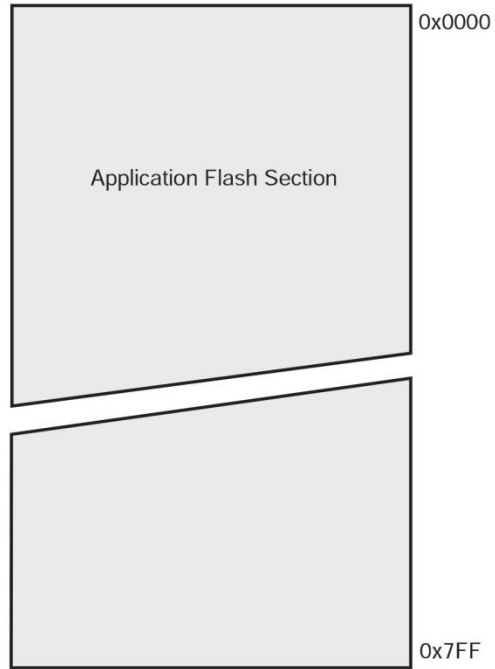
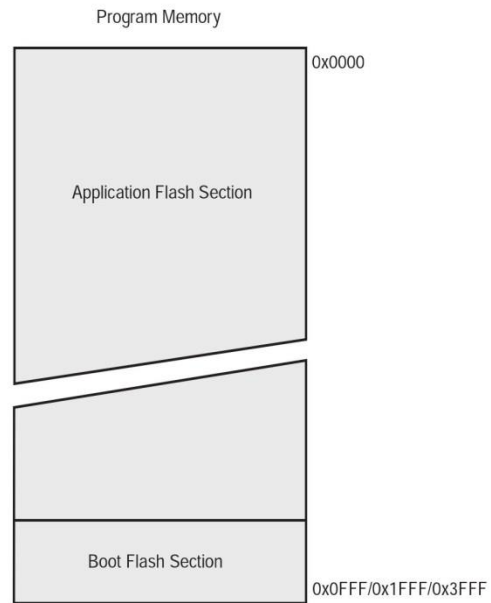


Figure 7-2. Program Memory Map ATmega88PA, ATmega168PA and ATmega328P



7.3 SRAM Data Memory

Figure 7-3 shows how the ATmega48PA/88PA/168PA/328P SRAM Memory is organized.

The ATmega48PA/88PA/168PA/328P is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in the Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

The lower 768/1280/1280/2303 data memory locations address both the Register File, the I/O memory, Extended I/O memory, and the internal data SRAM. The first 32 locations address the Register File, the next 64 location the standard I/O memory, then 160 locations of Extended I/O memory, and the next 512/1024/1024/2048 locations address the internal data SRAM.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers.

The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented.

The 32 general purpose working registers, 64 I/O Registers, 160 Extended I/O Registers, and the 512/1024/1024/2048 bytes of internal data SRAM in the ATmega48PA/88PA/168PA/328P are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 11.

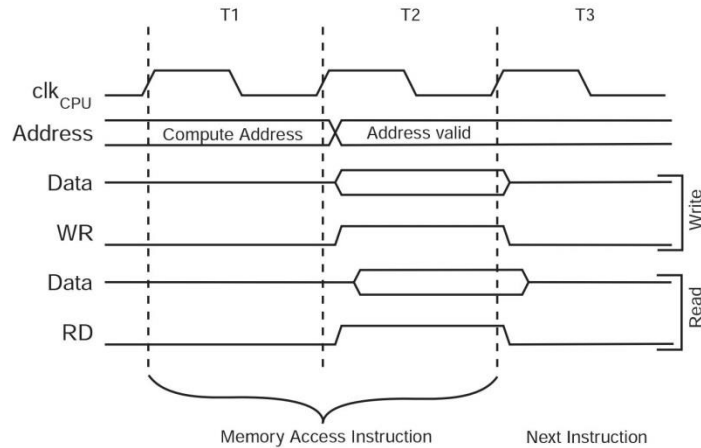
Figure 7-3. Data Memory Map

Data Memory	
32 Registers	0x0000 - 0x001F
64 I/O Registers	0x0020 - 0x005F
160 Ext I/O Reg.	0x0060 - 0x00FF
Internal SRAM (512/1024/1024/2048 x 8)	0x0100
	0x04FF/0x04FF/0x00FF/0x08FF

7.3.1 Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two clk_{CPU} cycles as described in Figure 7-4.

Figure 7-4. On-chip Data SRAM Access Cycles



7.4 EEPROM Data Memory

The ATmega48PA/88PA/168PA/328P contains 256/512/512/1K bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Register, the EEPROM Data Register, and the EEPROM Control Register.

"Memory Programming" on page 294 contains a detailed description on EEPROM Programming in SPI or Parallel Programming mode.

7.4.1 EEPROM Read/Write Access

The EEPROM Access Registers are accessible in the I/O space.

The write access time for the EEPROM is given in Table 7-2. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies, V_{CC} is likely to rise or fall slowly on power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. See "Preventing EEPROM Corruption" on page 20 for details on how to avoid problems in these situations.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.

7.4.2 Preventing EEPROM Corruption

During periods of low V_{CC} , the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage is too low.

EEPROM data corruption can easily be avoided by following this design recommendation:

Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD). If the detection level of the internal BOD does not match the needed detection level, an external low V_{CC} reset Protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.

7.5 I/O Memory

The I/O space definition of the ATmega48PA/88PA/168PA/328P is shown in "[Register Summary](#)" on page 423.

All ATmega48PA/88PA/168PA/328P I/Os and peripherals are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBIS and SBIC instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set section for more details. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48PA/88PA/168PA/328P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

The I/O and peripherals control registers are explained in later sections.

7.5.1 General Purpose I/O Registers

The ATmega48PA/88PA/168PA/328P contains three General Purpose I/O Registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and Status Flags. General Purpose I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

7.6 Register Description

7.6.1 EEARH and EEARL – The EEPROM Address Register

Bit	15	14	13	12	11	10	9	8	
0x22 (0x42)	-	-	-	-	-	-	-	EEAR8	EEARH
0x21 (0x41)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

- **Bits 15..9 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bits 8..0 – EEAR8..0: EEPROM Address**

The EEPROM Address Registers – EEARH and EEARL specify the EEPROM address in the 256/512/512/1K bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 255/511/511/1023. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

EEAR8 is an unused bit in ATmega48PA and must always be written to zero.

7.6.2 EEDR – The EEPROM Data Register

Bit	7	6	5	4	3	2	1	0	
0x20 (0x40)	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..0 – EEDR7.0: EEPROM Data**

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

7.6.3 EECR – The EEPROM Control Register

Bit	7	6	5	4	3	2	1	0	
0x1F (0x3F)	-	-	EPM1	EPM0	EERIE	EEMPE	EEPE	EERE	EECR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	X	X	0	0	X	0	

- **Bits 7..6 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bits 5, 4 – EPM1 and EPM0: EEPROM Programming Mode Bits**

The EEPROM Programming mode bit setting defines which programming action that will be triggered when writing EEPE. It is possible to program data in one atomic operation (erase the old value and program the new value) or to split the Erase and Write operations in two different operations. The Programming times for the different modes are shown in [Table 7-1](#). While EEPE

is set, any write to EEP Mn will be ignored. During reset, the EEP Mn bits will be reset to 0b00 unless the EEPROM is busy programming.

Table 7-1. EEPROM Mode Bits

EEP M1	EEP M0	Programming Time	Operation
0	0	3.4 ms	Erase and Write in one operation (Atomic Operation)
0	1	1.8 ms	Erase Only
1	0	1.8 ms	Write Only
1	1	–	Reserved for future use

• **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEPE is cleared. The interrupt will not be generated during EEPROM write or SPM.

• **Bit 2 – EEMPE: EEPROM Master Write Enable**

The EEMPE bit determines whether setting EEPE to one causes the EEPROM to be written. When EEMPE is set, setting EEPE within four clock cycles will write data to the EEPROM at the selected address. If EEMPE is zero, setting EEPE will have no effect. When EEMPE has been written to one by software, hardware clears the bit to zero after four clock cycles. See the description of the EEPE bit for an EEPROM write procedure.

• **Bit 1 – EEPE: EEPROM Write Enable**

The EEPROM Write Enable Signal EEPE is the write strobe to the EEPROM. When address and data are correctly set up, the EEPE bit must be written to one to write the value into the EEPROM. The EEMPE bit must be written to one before a logical one is written to EEPE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 3 and 4 is not essential):

1. Wait until EEPE becomes zero.
2. Wait until SELFPRGEN in SPMCSR becomes zero.
3. Write new EEPROM address to EEAR (optional).
4. Write new EEPROM data to EEDR (optional).
5. Write a logical one to the EEMPE bit while writing a zero to EEPE in EECR.
6. Within four clock cycles after setting EEMPE, write a logical one to EEPE.

The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a Boot Loader allowing the CPU to program the Flash. If the Flash is never being updated by the CPU, step 2 can be omitted. See ["Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P"](#) on page 277 for details about Boot programming.

Caution: An interrupt between step 5 and step 6 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR or EEDR Register will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the Global Interrupt Flag cleared during all the steps to avoid these problems.

When the write access time has elapsed, the EEPE bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEPE has been set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR Register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEPE bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEAR Register.

The calibrated Oscillator is used to time the EEPROM accesses. [Table 7-2](#) lists the typical programming time for EEPROM access from the CPU.

Table 7-2. EEPROM Programming Time

Symbol	Number of Calibrated RC Oscillator Cycles	Typ Programming Time
EEPROM write (from CPU)	26,368	3.3 ms

The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (e.g. by disabling interrupts globally) so that no interrupts will occur during execution of these functions. The examples also assume that no Flash Boot Loader is present in the software. If such code is present, the EEPROM write function must also wait for any ongoing SPM command to finish.

Assembly Code Example

```
EEPROM_write:
    ; Wait for completion of previous write
    sbic EECR,EEPE
    rjmp EEPROM_write
    ; Set up address (r18:r17) in address register
    out EEARH, r18
    out EEARL, r17
    ; Write data (r16) to Data Register
    out EEDR,r16
    ; Write logical one to EEMPE
    sbi EECR,EEMPE
    ; Start eeprom write by setting EEPE
    sbi EECR,EEPE
    ret
```

C Code Example

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
    /* Wait for completion of previous write */
    while(EECR & (1<<EEPE))
        ;
    /* Set up address and Data Registers */
    EEAR = uiAddress;
    EEDR = ucData;
    /* Write logical one to EEMPE */
    EECR |= (1<<EEMPE);
    /* Start eeprom write by setting EEPE */
    EECR |= (1<<EEPE);
}
```

The next code examples show assembly and C functions for reading the EEPROM. The examples assume that interrupts are controlled so that no interrupts will occur during execution of these functions.

Assembly Code Example

```
EEPROM_read:
    ; Wait for completion of previous write
    sbic EECR,EEPE
    rjmp EEPROM_read
    ; Set up address (r18:r17) in address register
    out EEARH, r18
    out EEARL, r17
    ; Start eeprom read by writing EERE
    sbi EECR,EERE
    ; Read data from Data Register
    in r16,EEDR
    ret
```

C Code Example

```
unsigned char EEPROM_read(unsigned int uiAddress)
{
    /* Wait for completion of previous write */
    while(EECR & (1<<EEPE))
        ;
    /* Set up address register */
    EEAR = uiAddress;
    /* Start eeprom read by writing EERE */
    EECR |= (1<<EERE);
    /* Return data from Data Register */
    return EEDR;
}
```

7.6.4 GPIOR2 – General Purpose I/O Register 2

Bit	7	6	5	4	3	2	1	0	
0x2B (0x4B)	MSB LSB								GPIOR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

7.6.5 GPIOR1 – General Purpose I/O Register 1

Bit	7	6	5	4	3	2	1	0	
0x2A (0x4A)	MSB LSB								GPIOR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

7.6.6 GPIOR0 – General Purpose I/O Register 0

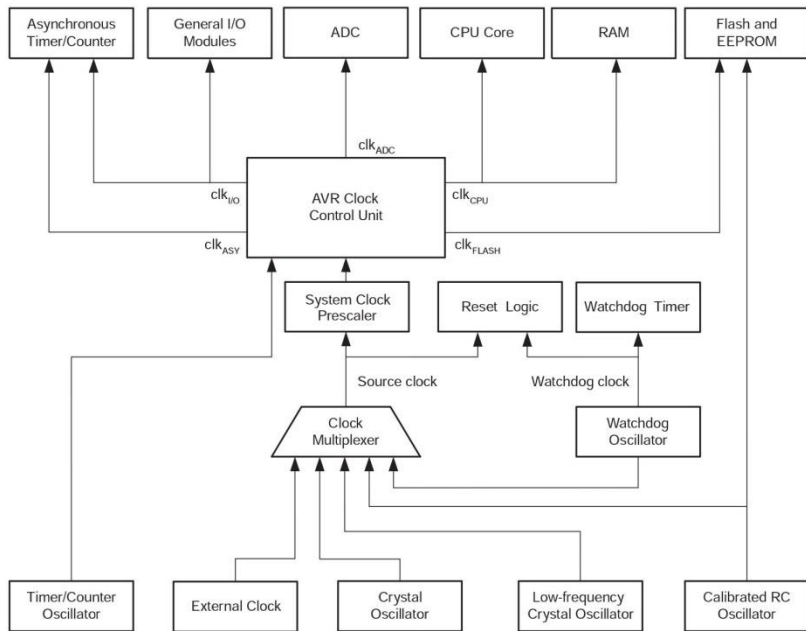
Bit	7	6	5	4	3	2	1	0	
0x1E (0x3E)	MSB LSB								GPIOR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

8. System Clock and Clock Options

8.1 Clock Systems and their Distribution

Figure 8-1 presents the principal clock systems in the AVR and their distribution. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes, as described in "Power Management and Sleep Modes" on page 39. The clock systems are detailed below.

Figure 8-1. Clock Distribution



8.1.1 CPU Clock – clk_{CPU}

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

8.1.2 I/O Clock – $clk_{I/O}$

The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted. Also note that start condition detection in the USI module is carried out asynchronously when $clk_{I/O}$ is halted, TWI address recognition in all sleep modes.

8.1.3 Flash Clock – clk_{FLASH}

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

8.1.4 Asynchronous Timer Clock – clk_{ASY}

The Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external clock or an external 32 kHz clock crystal. The dedicated clock domain allows using this Timer/Counter as a real-time counter even when the device is in sleep mode.

8.1.5 ADC Clock – clk_{ADC}

The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results.

8.2 Clock Sources

The device has the following clock source options, selectable by Flash Fuse bits as shown below. The clock from the selected source is input to the AVR clock generator, and routed to the appropriate modules.

Table 8-1. Device Clocking Options Select⁽¹⁾

Device Clocking Option	CKSEL3..0
Low Power Crystal Oscillator	1111 - 1000
Full Swing Crystal Oscillator	0111 - 0110
Low Frequency Crystal Oscillator	0101 - 0100
Internal 128 kHz RC Oscillator	0011
Calibrated Internal RC Oscillator	0010
External Clock	0000
Reserved	0001

Note: 1. For all fuses "1" means unprogrammed while "0" means programmed.

8.2.1 Default Clock Source

The device is shipped with internal RC oscillator at 8.0MHz and with the fuse CKDIV8 programmed, resulting in 1.0MHz system clock. The startup time is set to maximum and time-out period enabled. (CKSEL = "0010", SUT = "10", CKDIV8 = "0"). The default setting ensures that all users can make their desired clock source setting using any available programming interface.

8.2.2 Clock Startup Sequence

Any clock source needs a sufficient V_{CC} to start oscillating and a minimum number of oscillating cycles before it can be considered stable.

To ensure sufficient V_{CC} , the device issues an internal reset with a time-out delay (t_{TOUT}) after the device reset is released by all other reset sources. "System Control and Reset" on page 46 describes the start conditions for the internal reset. The delay (t_{TOUT}) is timed from the Watchdog Oscillator and the number of cycles in the delay is set by the SUTx and CKSELx fuse bits. The

selectable delays are shown in [Table 8-2](#). The frequency of the Watchdog Oscillator is voltage dependent as shown in ["Typical Characteristics" on page 326](#).

Table 8-2. Number of Watchdog Oscillator Cycles

Typ Time-out ($V_{CC} = 5.0V$)	Typ Time-out ($V_{CC} = 3.0V$)	Number of Cycles
0 ms	0 ms	0
4.1 ms	4.3 ms	512
65 ms	69 ms	8K (8,192)

Main purpose of the delay is to keep the AVR in reset until it is supplied with minimum V_{CC} . The delay will not monitor the actual voltage and it will be required to select a delay longer than the V_{CC} rise time. If this is not possible, an internal or external Brown-Out Detection circuit should be used. A BOD circuit will ensure sufficient V_{CC} before it releases the reset, and the time-out delay can be disabled. Disabling the time-out delay without utilizing a Brown-Out Detection circuit is not recommended.

The oscillator is required to oscillate for a minimum number of cycles before the clock is considered stable. An internal ripple counter monitors the oscillator output clock, and keeps the internal reset active for a given number of clock cycles. The reset is then released and the device will start to execute. The recommended oscillator start-up time is dependent on the clock type, and varies from 6 cycles for an externally applied clock to 32K cycles for a low frequency crystal.

The start-up sequence for the clock includes both the time-out delay and the start-up time when the device starts up from reset. When starting up from Power-save or Power-down mode, V_{CC} is assumed to be at a sufficient level and only the start-up time is included.

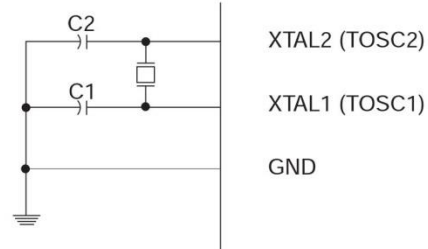
8.3 Low Power Crystal Oscillator

Pins XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an On-chip Oscillator, as shown in [Figure 8-2 on page 29](#). Either a quartz crystal or a ceramic resonator may be used.

This Crystal Oscillator is a low power oscillator, with reduced voltage swing on the XTAL2 output. It gives the lowest power consumption, but is not capable of driving other clock inputs, and may be more susceptible to noise in noisy environments. In these cases, refer to the ["Full Swing Crystal Oscillator" on page 30](#).

C1 and C2 should always be equal for both crystals and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. Some initial guidelines for choosing capacitors for use with crystals are given in [Table 8-3 on page 29](#). For ceramic resonators, the capacitor values given by the manufacturer should be used.

Figure 8-2. Crystal Oscillator Connections



The Low Power Oscillator can operate in three different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3..1 as shown in [Table 8-3 on page 29](#).

Table 8-3. Low Power Crystal Oscillator Operating Modes⁽³⁾

Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 (pF)	CKSEL3..1 ⁽¹⁾
0.4 - 0.9	–	100 ⁽²⁾
0.9 - 3.0	12 - 22	101
3.0 - 8.0	12 - 22	110
8.0 - 16.0	12 - 22	111

- Notes:
1. This is the recommended CKSEL settings for the difference frequency ranges.
 2. This option should not be used with crystals, only with ceramic resonators.
 3. If 8 MHz frequency exceeds the specification of the device (depends on V_{CC}), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8. It must be ensured that the resulting divided clock meets the frequency specification of the device.

The CKSEL0 Fuse together with the SUT1..0 Fuses select the start-up times as shown in [Table 8-4](#).

Table 8-4. Start-up Times for the Low Power Crystal Oscillator Clock Selection

Oscillator Source / Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	CKSEL0	SUT1..0
Ceramic resonator, fast rising power	258 CK	14CK + 4.1 ms ⁽¹⁾	0	00
Ceramic resonator, slowly rising power	258 CK	14CK + 65 ms ⁽¹⁾	0	01
Ceramic resonator, BOD enabled	1K CK	14CK ⁽²⁾	0	10
Ceramic resonator, fast rising power	1K CK	14CK + 4.1 ms ⁽²⁾	0	11
Ceramic resonator, slowly rising power	1K CK	14CK + 65 ms ⁽²⁾	1	00

Table 8-4. Start-up Times for the Low Power Crystal Oscillator Clock Selection (Continued)

Oscillator Source / Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	CKSEL0	SUT1..0
Crystal Oscillator, BOD enabled	16K CK	14CK	1	01
Crystal Oscillator, fast rising power	16K CK	14CK + 4.1 ms	1	10
Crystal Oscillator, slowly rising power	16K CK	14CK + 65 ms	1	11

- Notes:
1. These options should only be used when not operating close to the maximum frequency of the device, and only if frequency stability at start-up is not important for the application. These options are not suitable for crystals.
 2. These options are intended for use with ceramic resonators and will ensure frequency stability at start-up. They can also be used with crystals when not operating close to the maximum frequency of the device, and if frequency stability at start-up is not important for the application.

8.4 Full Swing Crystal Oscillator

Pins XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an On-chip Oscillator, as shown in [Figure 8-2 on page 29](#). Either a quartz crystal or a ceramic resonator may be used.

This Crystal Oscillator is a full swing oscillator, with rail-to-rail swing on the XTAL2 output. This is useful for driving other clock inputs and in noisy environments. The current consumption is higher than the "Low Power Crystal Oscillator" on [page 28](#). Note that the Full Swing Crystal Oscillator will only operate for $V_{CC} = 2.7 - 5.5$ volts.

C1 and C2 should always be equal for both crystals and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. Some initial guidelines for choosing capacitors for use with crystals are given in [Table 8-6 on page 31](#). For ceramic resonators, the capacitor values given by the manufacturer should be used.

The operating mode is selected by the fuses CKSEL3..1 as shown in [Table 8-5](#).

Table 8-5. Full Swing Crystal Oscillator operating modes

Frequency Range ⁽¹⁾ (MHz)	Recommended Range for Capacitors C1 and C2 (pF)	CKSEL3..1
0.4 - 20	12 - 22	011

- Notes:
1. If 8 MHz frequency exceeds the specification of the device (depends on V_{CC}), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8. It must be ensured that the resulting divided clock meets the frequency specification of the device.

Figure 8-3. Crystal Oscillator Connections

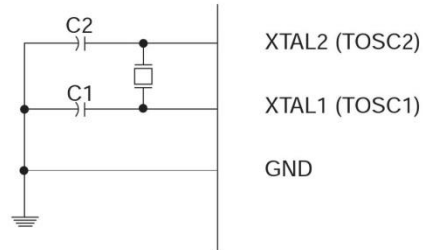


Table 8-6. Start-up Times for the Full Swing Crystal Oscillator Clock Selection

Oscillator Source / Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	CKSEL0	SUT1..0
Ceramic resonator, fast rising power	258 CK	$14CK + 4.1 \text{ ms}^{(1)}$	0	00
Ceramic resonator, slowly rising power	258 CK	$14CK + 65 \text{ ms}^{(1)}$	0	01
Ceramic resonator, BOD enabled	1K CK	$14CK^{(2)}$	0	10
Ceramic resonator, fast rising power	1K CK	$14CK + 4.1 \text{ ms}^{(2)}$	0	11
Ceramic resonator, slowly rising power	1K CK	$14CK + 65 \text{ ms}^{(2)}$	1	00
Crystal Oscillator, BOD enabled	16K CK	14CK	1	01
Crystal Oscillator, fast rising power	16K CK	$14CK + 4.1 \text{ ms}$	1	10
Crystal Oscillator, slowly rising power	16K CK	$14CK + 65 \text{ ms}$	1	11

- Notes:
1. These options should only be used when not operating close to the maximum frequency of the device, and only if frequency stability at start-up is not important for the application. These options are not suitable for crystals.
 2. These options are intended for use with ceramic resonators and will ensure frequency stability at start-up. They can also be used with crystals when not operating close to the maximum frequency of the device, and if frequency stability at start-up is not important for the application.

8.5 Low Frequency Crystal Oscillator

The Low-frequency Crystal Oscillator is optimized for use with a 32.768 kHz watch crystal. When selecting crystals, load capacitance and crystal's Equivalent Series Resistance, ESR must be taken into consideration. Both values are specified by the crystal vendor. ATmega48PA/88PA/168PA/328P oscillator is optimized for very low power consumption, and thus when selecting crystals, see [Table 8-7 on page 32](#) for maximum ESR recommendations on 6.5 pF, 9.0 pF and 12.5 pF crystals

Table 8-7. Maximum ESR Recommendation for 32.768 kHz Crystal Oscillator

Crystal CL (pF)	Max ESR [kΩ] ⁽¹⁾
6.5	75
9.0	65
12.5	30

Note: 1. Maximum ESR is typical value based on characterization

The Low-frequency Crystal Oscillator provides an internal load capacitance at each TOSC pin as specified in the [Table 8-8 on page 32](#).

Table 8-8. Capacitance for Low-Frequency Crystal Oscillator

Device	32 kHz Osc. Type	Cap (Xtal1/Tosc1)	Cap (Xtal2/Tosc2)
ATmega48PA/88PA/168PA/328P	System Osc.	18 pF	8 pF
	Timer Osc.	18 pF	8 pF

The external capacitance (C) needed at each TOSC pin can be calculated by using:

$$C = 2 \cdot CL - C_s$$

where CL is the load capacitance for a 32.768 kHz crystal specified by the crystal vendor and C_s is the total stray capacitance for one TOSC pin.

Crystals specifying load capacitance (CL) higher than the ones given in the [Table 8-8 on page 32](#), require external capacitors applied as described in [Figure 8-2 on page 29](#).

The Low-frequency Crystal Oscillator must be selected by setting the CKSEL Fuses to "0110" or "0111", as shown in [Table 8-10](#). Start-up times are determined by the SUT Fuses as shown in [Table 8-9](#).

Table 8-9. Start-up Times for the Low-frequency Crystal Oscillator Clock Selection

SUT1..0	Additional Delay from Reset (V _{CC} = 5.0V)	Recommended Usage
00	4 CK	Fast rising power or BOD enabled
01	4 CK + 4.1 ms	Slowly rising power
10	4 CK + 65 ms	Stable frequency at start-up
11	Reserved	

Table 8-10. Start-up Times for the Low-frequency Crystal Oscillator Clock Selection

CKSEL3..0	Start-up Time from Power-down and Power-save	Recommended Usage
0100 ⁽¹⁾	1K CK	
0101	32K CK	Stable frequency at start-up

Note: 1. This option should only be used if frequency stability at start-up is not important for the application

8.6 Calibrated Internal RC Oscillator

By default, the Internal RC Oscillator provides an approximate 8.0 MHz clock. Though voltage and temperature dependent, this clock can be very accurately calibrated by the user. See [Table 28-1 on page 317](#) for more details. The device is shipped with the CKDIV8 Fuse programmed. See ["System Clock Prescaler" on page 35](#) for more details.

This clock may be selected as the system clock by programming the CKSEL Fuses as shown in [Table 8-11](#). If selected, it will operate with no external components. During reset, hardware loads the pre-programmed calibration value into the OSCCAL Register and thereby automatically calibrates the RC Oscillator. The accuracy of this calibration is shown as Factory calibration in [Table 28-1 on page 317](#).

By changing the OSCCAL register from SW, see ["OSCCAL – Oscillator Calibration Register" on page 37](#), it is possible to get a higher calibration accuracy than by using the factory calibration. The accuracy of this calibration is shown as User calibration in [Table 28-1 on page 317](#).

When this Oscillator is used as the chip clock, the Watchdog Oscillator will still be used for the Watchdog Timer and for the Reset Time-out. For more information on the pre-programmed calibration value, see the section ["Calibration Byte" on page 298](#).

Table 8-11. Internal Calibrated RC Oscillator Operating Modes

Frequency Range ⁽²⁾ (MHz)	CKSEL3..0
7.3 - 8.1	0010 ⁽¹⁾

Notes: 1. The device is shipped with this option selected.
2. If 8 MHz frequency exceeds the specification of the device (depends on V_{CC}), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8.

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in [Table 8-12 on page 33](#).

Table 8-12. Start-up times for the internal calibrated RC Oscillator clock selection

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	SUT1..0
BOD enabled	6 CK	14CK ⁽¹⁾	00
Fast rising power	6 CK	14CK + 4.1 ms	01
Slowly rising power	6 CK	14CK + 65 ms ⁽²⁾	10
	Reserved		11

Note: 1. If the RSTDISBL fuse is programmed, this start-up time will be increased to 14CK + 4.1 ms to ensure programming mode can be entered.
2. The device is shipped with this option selected.

8.7 128 kHz Internal Oscillator

The 128 kHz internal Oscillator is a low power Oscillator providing a clock of 128 kHz. The frequency is nominal at 3V and 25°C. This clock may be selected as the system clock by programming the CKSEL Fuses to "11" as shown in [Table 8-13](#).

Table 8-13. 128 kHz Internal Oscillator Operating Modes

Nominal Frequency ⁽¹⁾	CKSEL3..0
128 kHz	0011

Note: 1. Note that the 128 kHz oscillator is a very low power clock source, and is not designed for high accuracy.

When this clock source is selected, start-up times are determined by the SUT Fuses as shown in [Table 8-14](#).

Table 8-14. Start-up Times for the 128 kHz Internal Oscillator

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset	SUT1..0
BOD enabled	6 CK	14CK ⁽¹⁾	00
Fast rising power	6 CK	14CK + 4 ms	01
Slowly rising power	6 CK	14CK + 64 ms	10
	Reserved		11

Note: 1. If the RSTDISBL fuse is programmed, this start-up time will be increased to 14CK + 4.1 ms to ensure programming mode can be entered.

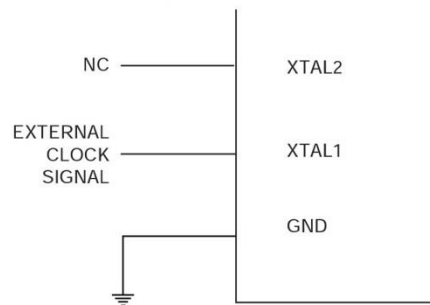
8.8 External Clock

To drive the device from an external clock source, XTAL1 should be driven as shown in [Figure 8-4 on page 34](#). To run the device on an external clock, the CKSEL Fuses must be programmed to "0000" (see [Table 8-15](#)).

Table 8-15. Crystal Oscillator Clock Frequency

Frequency	CKSEL3..0
0 - 20 MHz	0000

Figure 8-4. External Clock Drive Configuration



When this clock source is selected, start-up times are determined by the SUT Fuses as shown in [Table 8-16](#).

Table 8-16. Start-up Times for the External Clock Selection

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	SUT1..0
BOD enabled	6 CK	14CK	00
Fast rising power	6 CK	14CK + 4.1 ms	01
Slowly rising power	6 CK	14CK + 65 ms	10
	Reserved		11

When applying an external clock, it is required to avoid sudden changes in the applied clock frequency to ensure stable operation of the MCU. A variation in frequency of more than 2% from one clock cycle to the next can lead to unpredictable behavior. If changes of more than 2% is required, ensure that the MCU is kept in Reset during the changes.

Note that the System Clock Prescaler can be used to implement run-time changes of the internal clock frequency while still ensuring stable operation. Refer to "[System Clock Prescaler](#)" on page 35 for details.

8.9 Clock Output Buffer

The device can output the system clock on the CLKO pin. To enable the output, the CKOUT Fuse has to be programmed. This mode is suitable when the chip clock is used to drive other circuits on the system. The clock also will be output during reset, and the normal operation of I/O pin will be overridden when the fuse is programmed. Any clock source, including the internal RC Oscillator, can be selected when the clock is output on CLKO. If the System Clock Prescaler is used, it is the divided system clock that is output.

8.10 Timer/Counter Oscillator

ATmega48PA/88PA/168PA/328P uses the same crystal oscillator for Low-frequency Oscillator and Timer/Counter Oscillator. See "[Low Frequency Crystal Oscillator](#)" on page 32 for details on the oscillator and crystal requirements.

ATmega48PA/88PA/168PA/328P share the Timer/Counter Oscillator Pins (TOSC1 and TOSC2) with XTAL1 and XTAL2. When using the Timer/Counter Oscillator, the system clock needs to be four times the oscillator frequency. Due to this and the pin sharing, the Timer/Counter Oscillator can only be used when the Calibrated Internal RC Oscillator is selected as system clock source.

Applying an external clock source to TOSC1 can be done if EXTCLK in the ASSR Register is written to logic one. See "[Asynchronous Operation of Timer/Counter2](#)" on page 155 for further description on selecting external clock as input instead of a 32.768 kHz watch crystal.

8.11 System Clock Prescaler

The ATmega48PA/88PA/168PA/328P has a system clock prescaler, and the system clock can be divided by setting the "[CLKPR – Clock Prescale Register](#)" on page 377. This feature can be used to decrease the system clock frequency and the power consumption when the requirement for processing power is low. This can be used with all clock source options, and it will affect the clock frequency of the CPU and all synchronous peripherals. $clk_{I/O}$, clk_{ADC} , clk_{CPU} , and clk_{FLASH} are divided by a factor as shown in [Table 28-3](#) on page 318.

When switching between prescaler settings, the System Clock Prescaler ensures that no glitches occurs in the clock system. It also ensures that no intermediate frequency is higher than neither the clock frequency corresponding to the previous setting, nor the clock frequency corresponding to the new setting. The ripple counter that implements the prescaler runs at the frequency of the undivided clock, which may be faster than the CPU's clock frequency. Hence, it is not possible to determine the state of the prescaler - even if it were readable, and the exact time it takes to switch from one clock division to the other cannot be exactly predicted. From the time the CLKPS values are written, it takes between $T1 + T2$ and $T1 + 2 * T2$ before the new clock frequency is active. In this interval, 2 active clock edges are produced. Here, $T1$ is the previous clock period, and $T2$ is the period corresponding to the new prescaler setting.

To avoid unintentional changes of clock frequency, a special write procedure must be followed to change the CLKPS bits:

1. Write the Clock Prescaler Change Enable (CLKPCE) bit to one and all other bits in CLKPR to zero.
2. Within four cycles, write the desired value to CLKPS while writing a zero to CLKPCE.

Interrupts must be disabled when changing prescaler setting to make sure the write procedure is not interrupted.

8.12 Register Description

8.12.1 OSCCAL – Oscillator Calibration Register

Bit	7	6	5	4	3	2	1	0									
(0x66)	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">CAL7</td> <td style="padding: 2px;">CAL6</td> <td style="padding: 2px;">CAL5</td> <td style="padding: 2px;">CAL4</td> <td style="padding: 2px;">CAL3</td> <td style="padding: 2px;">CAL2</td> <td style="padding: 2px;">CAL1</td> <td style="padding: 2px;">CAL0</td> </tr> </table>								CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0										
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W									
Initial Value	Device Specific Calibration Value																

- **Bits 7..0 – CAL7..0: Oscillator Calibration Value**

The Oscillator Calibration Register is used to trim the Calibrated Internal RC Oscillator to remove process variations from the oscillator frequency. A pre-programmed calibration value is automatically written to this register during chip reset, giving the Factory calibrated frequency as specified in [Table 28-1 on page 317](#). The application software can write this register to change the oscillator frequency. The oscillator can be calibrated to frequencies as specified in [Table 28-1 on page 317](#). Calibration outside that range is not guaranteed.

Note that this oscillator is used to time EEPROM and Flash write accesses, and these write times will be affected accordingly. If the EEPROM or Flash are written, do not calibrate to more than 8.8 MHz. Otherwise, the EEPROM or Flash write may fail.

The CAL7 bit determines the range of operation for the oscillator. Setting this bit to 0 gives the lowest frequency range, setting this bit to 1 gives the highest frequency range. The two frequency ranges are overlapping, in other words a setting of OSCCAL = 0x7F gives a higher frequency than OSCCAL = 0x80.

The CAL6..0 bits are used to tune the frequency within the selected range. A setting of 0x00 gives the lowest frequency in that range, and a setting of 0x7F gives the highest frequency in the range.

8.12.2 CLKPR – Clock Prescale Register

Bit	7	6	5	4	3	2	1	0									
(0x61)	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">CLKPCE</td> <td style="padding: 2px;">–</td> <td style="padding: 2px;">–</td> <td style="padding: 2px;">–</td> <td style="padding: 2px;">CLKPS3</td> <td style="padding: 2px;">CLKPS2</td> <td style="padding: 2px;">CLKPS1</td> <td style="padding: 2px;">CLKPS0</td> </tr> </table>								CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	CLKPR
CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0										
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W									
Initial Value	0	0	0	0	See Bit Description												

- **Bit 7 – CLKPCE: Clock Prescaler Change Enable**

The CLKPCE bit must be written to logic one to enable change of the CLKPS bits. The CLKPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero. CLKPCE is cleared by hardware four cycles after it is written or when CLKPS bits are written. Rewriting the CLKPCE bit within this time-out period does neither extend the time-out period, nor clear the CLKPCE bit.

- **Bits 3..0 – CLKPS3..0: Clock Prescaler Select Bits 3 - 0**

These bits define the division factor between the selected clock source and the internal system clock. These bits can be written run-time to vary the clock frequency to suit the application requirements. As the divider divides the master clock input to the MCU, the speed of all synchronous peripherals is reduced when a division factor is used. The division factors are given in [Table 8-17 on page 38](#).

ATmega48PA/88PA/168PA/328P

The CKDIV8 Fuse determines the initial value of the CLKPS bits. If CKDIV8 is unprogrammed, the CLKPS bits will be reset to "0000". If CKDIV8 is programmed, CLKPS bits are reset to "0011", giving a division factor of 8 at start up. This feature should be used if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. Note that any value can be written to the CLKPS bits regardless of the CKDIV8 Fuse setting. The Application software must ensure that a sufficient division factor is chosen if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. The device is shipped with the CKDIV8 Fuse programmed.

Table 8-17. Clock Prescaler Select

CLKPS3	CLKPS2	CLKPS1	CLKPS0	Clock Division Factor
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

9. Power Management and Sleep Modes

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The AVR provides various sleep modes allowing the user to tailor the power consumption to the application's requirements.

When enabled, the Brown-out Detector (BOD) actively monitors the power supply voltage during the sleep periods. To further save power, it is possible to disable the BOD in some sleep modes. See "BOD Disable" on page 40 for more details.

9.1 Sleep Modes

Figure 8-1 on page 26 presents the different clock systems in the ATmega48PA/88PA/168PA/328P, and their distribution. The figure is helpful in selecting an appropriate sleep mode. Table 9-1 shows the different sleep modes, their wake up sources BOD disable ability.

Table 9-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes.

Sleep Mode	Active Clock Domains					Oscillators		Wake-up Sources							
	clk _{CPU}	clk _{FLASH}	clk _{IO}	clk _{ADC}	clk _{ASY}	Main Clock Source Enabled	Timer Oscillator Enabled	INT1, INT0 and Pin Change	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other I/O	Software BOD Disable
Idle			X	X	X	X	X ⁽²⁾	X	X	X	X	X	X	X	
ADC Noise Reduction				X	X	X	X ⁽²⁾	X ⁽³⁾	X	X ⁽²⁾	X	X	X		
Power-down								X ⁽³⁾	X				X		X
Power-save					X		X ⁽²⁾	X ⁽³⁾	X	X			X		X
Standby ⁽¹⁾						X		X ⁽³⁾	X				X		X
Extended Standby					X ⁽²⁾	X	X ⁽²⁾	X ⁽³⁾	X	X			X		X

- Notes:
1. Only recommended with external crystal or resonator selected as clock source.
 2. If Timer/Counter2 is running in asynchronous mode.
 3. For INT1 and INT0, only level interrupt.

To enter any of the six sleep modes, the SE bit in SMCR must be written to logic one and a SLEEP instruction must be executed. The SM2, SM1, and SM0 bits in the SMCR Register select which sleep mode (Idle, ADC Noise Reduction, Power-down, Power-save, Standby, or Extended Standby) will be activated by the SLEEP instruction. See Table 9-2 on page 44 for a summary.

If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the Register File and SRAM are unaltered when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

9.2 BOD Disable

When the Brown-out Detector (BOD) is enabled by BODLEVEL fuses, [Table 27-7 on page 296](#), the BOD is actively monitoring the power supply voltage during a sleep period. To save power, it is possible to disable the BOD by software for some of the sleep modes, see [Table 9-1 on page 39](#). The sleep mode power consumption will then be at the same level as when BOD is globally disabled by fuses. If BOD is disabled in software, the BOD function is turned off immediately after entering the sleep mode. Upon wake-up from sleep, BOD is automatically enabled again. This ensures safe operation in case the V_{CC} level has dropped during the sleep period.

When the BOD has been disabled, the wake-up time from sleep mode will be approximately 60 μ s to ensure that the BOD is working correctly before the MCU continues executing code.

BOD disable is controlled by bit 6, BODS (BOD Sleep) in the control register MCUCR, see ["MCUCR – MCU Control Register" on page 44](#). Writing this bit to one turns off the BOD in relevant sleep modes, while a zero in this bit keeps BOD active. Default setting keeps BOD active, i.e. BODS set to zero.

Writing to the BODS bit is controlled by a timed sequence and an enable bit, see ["MCUCR – MCU Control Register" on page 44](#).

9.3 Idle Mode

When the SM2..0 bits are written to 000, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing the SPI, USART, Analog Comparator, ADC, 2-wire Serial Interface, Timer/Counters, Watchdog, and the interrupt system to continue operating. This sleep mode basically halts clk_{CPU} and clk_{FLASH} , while allowing the other clocks to run.

Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow and USART Transmit Complete interrupts. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD bit in the Analog Comparator Control and Status Register – ACSR. This will reduce power consumption in Idle mode. If the ADC is enabled, a conversion starts automatically when this mode is entered.

9.4 ADC Noise Reduction Mode

When the SM2..0 bits are written to 001, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode, stopping the CPU but allowing the ADC, the external interrupts, the 2-wire Serial Interface address watch, Timer/Counter2⁽¹⁾, and the Watchdog to continue operating (if enabled). This sleep mode basically halts $clk_{I/O}$, clk_{CPU} , and clk_{FLASH} , while allowing the other clocks to run.

This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. Apart from the ADC Conversion Complete interrupt, only an External Reset, a Watchdog System Reset, a Watchdog Interrupt, a Brown-out Reset, a 2-wire Serial Interface address match, a Timer/Counter2 interrupt, an SPM/EEPROM ready interrupt, an external level interrupt on INT0 or INT1 or a pin change interrupt can wake up the MCU from ADC Noise Reduction mode.

Note: 1. Timer/Counter2 will only keep running in asynchronous mode, see ["8-bit Timer/Counter2 with PWM and Asynchronous Operation" on page 144](#) for details.

9.5 Power-down Mode

When the SM2..0 bits are written to 010, the SLEEP instruction makes the MCU enter Power-down mode. In this mode, the external Oscillator is stopped, while the external interrupts, the 2-wire Serial Interface address watch, and the Watchdog continue operating (if enabled). Only an External Reset, a Watchdog System Reset, a Watchdog Interrupt, a Brown-out Reset, a 2-wire Serial Interface address match, an external level interrupt on INT0 or INT1, or a pin change interrupt can wake up the MCU. This sleep mode basically halts all generated clocks, allowing operation of asynchronous modules only.

Note that if a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. Refer to ["External Interrupts" on page 70](#) for details.

When waking up from Power-down mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same CKSEL Fuses that define the Reset Time-out period, as described in ["Clock Sources" on page 27](#).

9.6 Power-save Mode

When the SM2..0 bits are written to 011, the SLEEP instruction makes the MCU enter Power-save mode. This mode is identical to Power-down, with one exception:

If Timer/Counter2 is enabled, it will keep running during sleep. The device can wake up from either Timer Overflow or Output Compare event from Timer/Counter2 if the corresponding Timer/Counter2 interrupt enable bits are set in TIMSK2, and the Global Interrupt Enable bit in SREG is set.

If Timer/Counter2 is not running, Power-down mode is recommended instead of Power-save mode.

The Timer/Counter2 can be clocked both synchronously and asynchronously in Power-save mode. If Timer/Counter2 is not using the asynchronous clock, the Timer/Counter Oscillator is stopped during sleep. If Timer/Counter2 is not using the synchronous clock, the clock source is stopped during sleep. Note that even if the synchronous clock is running in Power-save, this clock is only available for Timer/Counter2.

9.7 Standby Mode

When the SM2..0 bits are 110 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter Standby mode. This mode is identical to Power-down with the exception that the Oscillator is kept running. From Standby mode, the device wakes up in six clock cycles.

9.8 Extended Standby Mode

When the SM2..0 bits are 111 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter Extended Standby mode. This mode is identical to Power-save with the exception that the Oscillator is kept running. From Extended Standby mode, the device wakes up in six clock cycles.

9.9 Power Reduction Register

The Power Reduction Register (PRR), see ["PRR – Power Reduction Register" on page 45](#), provides a method to stop the clock to individual peripherals to reduce power consumption. The current state of the peripheral is frozen and the I/O registers can not be read or written. Resources used by the peripheral when stopping the clock will remain occupied, hence the peripheral should in most cases be disabled before stopping the clock. Waking up a module, which is done by clearing the bit in PRR, puts the module in the same state as before shutdown.

Module shutdown can be used in Idle mode and Active mode to significantly reduce the overall power consumption. In all other sleep modes, the clock is already stopped.

9.10 Minimizing Power Consumption

There are several possibilities to consider when trying to minimize the power consumption in an AVR controlled system. In general, sleep modes should be used as much as possible, and the sleep mode should be selected so that as few as possible of the device's functions are operating. All functions not needed should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

9.10.1 Analog to Digital Converter

If enabled, the ADC will be enabled in all sleep modes. To save power, the ADC should be disabled before entering any sleep mode. When the ADC is turned off and on again, the next conversion will be an extended conversion. Refer to ["Analog-to-Digital Converter" on page 250](#) for details on ADC operation.

9.10.2 Analog Comparator

When entering Idle mode, the Analog Comparator should be disabled if not used. When entering ADC Noise Reduction mode, the Analog Comparator should be disabled. In other sleep modes, the Analog Comparator is automatically disabled. However, if the Analog Comparator is set up to use the Internal Voltage Reference as input, the Analog Comparator should be disabled in all sleep modes. Otherwise, the Internal Voltage Reference will be enabled, independent of sleep mode. Refer to ["Analog Comparator" on page 246](#) for details on how to configure the Analog Comparator.

9.10.3 Brown-out Detector

If the Brown-out Detector is not needed by the application, this module should be turned off. If the Brown-out Detector is enabled by the BODLEVEL Fuses, it will be enabled in all sleep modes, and hence, always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to ["Brown-out Detection" on page 48](#) for details on how to configure the Brown-out Detector.

9.10.4 Internal Voltage Reference

The Internal Voltage Reference will be enabled when needed by the Brown-out Detection, the Analog Comparator or the ADC. If these modules are disabled as described in the sections above, the internal voltage reference will be disabled and it will not be consuming power. When turned on again, the user must allow the reference to start up before the output is used. If the reference is kept on in sleep mode, the output can be used immediately. Refer to ["Internal Voltage Reference" on page 49](#) for details on the start-up time.

9.10.5 Watchdog Timer

If the Watchdog Timer is not needed in the application, the module should be turned off. If the Watchdog Timer is enabled, it will be enabled in all sleep modes and hence always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to ["Watchdog Timer" on page 50](#) for details on how to configure the Watchdog Timer.

9.10.6 Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. The most important is then to ensure that no pins drive resistive loads. In sleep modes where both the I/O clock ($clk_{I/O}$) and the ADC clock (clk_{ADC}) are stopped, the input buffers of the device will be disabled. This ensures that no power is consumed by the input logic when not needed. In some cases, the input logic is needed for detecting wake-up conditions, and it will then be enabled. Refer to the section ["Digital Input Enable and Sleep Modes" on page 79](#) for details on which pins are enabled. If the input buffer is enabled and the input signal is left floating or have an analog signal level close to $V_{CC}/2$, the input buffer will use excessive power.

For analog input pins, the digital input buffer should be disabled at all times. An analog signal level close to $V_{CC}/2$ on an input pin can cause significant current even in active mode. Digital input buffers can be disabled by writing to the Digital Input Disable Registers (DIDR1 and DIDR0). Refer to ["DIDR1 – Digital Input Disable Register 1" on page 249](#) and ["DIDR0 – Digital Input Disable Register 0" on page 266](#) for details.

9.10.7 On-chip Debug System

If the On-chip debug system is enabled by the DWEN Fuse and the chip enters sleep mode, the main clock source is enabled and hence always consumes power. In the deeper sleep modes, this will contribute significantly to the total current consumption.

9.11 Register Description

9.11.1 SMCR – Sleep Mode Control Register

The Sleep Mode Control Register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
0x33 (0x53)	-	-	-	-	SM2	SM1	SM0	SE	SMCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..4 Res: Reserved Bits**

These bits are unused in the ATmega48PA/88PA/168PA/328P, and will always be read as zero.

- **Bits 3..1 – SM2..0: Sleep Mode Select Bits 2, 1, and 0**

These bits select between the five available sleep modes as shown in [Table 9-2](#).

Table 9-2. Sleep Mode Select

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby ⁽¹⁾
1	1	1	External Standby ⁽¹⁾

Note: 1. Standby mode is only recommended for use with external crystals or resonators.

- **Bit 0 – SE: Sleep Enable**

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmer's purpose, it is recommended to write the Sleep Enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.

9.11.2 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	-	BODS	BODSE	PUD	-	-	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – BODS: BOD Sleep**

The BODS bit must be written to logic one in order to turn off BOD during sleep, see [Table 9-1 on page 39](#). Writing to the BODS bit is controlled by a timed sequence and an enable bit, BODSE in MCUCR. To disable BOD in relevant sleep modes, both BODS and BODSE must first

be set to one. Then, to set the BODS bit, BODS must be set to one and BODSE must be set to zero within four clock cycles.

The BODS bit is active three clock cycles after it is set. A sleep instruction must be executed while BODS is active in order to turn off the BOD for the actual sleep mode. The BODS bit is automatically cleared after three clock cycles.

- **Bit 5 – BODSE: BOD Sleep Enable**

BODSE enables setting of BODS control bit, as explained in BODS bit description. BOD disable is controlled by a timed sequence.

9.11.3 PRR – Power Reduction Register

Bit (0x64)	7	6	5	4	3	2	1	0	
	PRTWI	PRTIM2	PRTIM0	–	PRTIM1	PRSPI	PRUSART0	PRADC	PRR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 - PRTWI: Power Reduction TWI**

Writing a logic one to this bit shuts down the TWI by stopping the clock to the module. When waking up the TWI again, the TWI should be re initialized to ensure proper operation.

- **Bit 6 - PRTIM2: Power Reduction Timer/Counter2**

Writing a logic one to this bit shuts down the Timer/Counter2 module in synchronous mode (AS2 is 0). When the Timer/Counter2 is enabled, operation will continue like before the shutdown.

- **Bit 5 - PRTIM0: Power Reduction Timer/Counter0**

Writing a logic one to this bit shuts down the Timer/Counter0 module. When the Timer/Counter0 is enabled, operation will continue like before the shutdown.

- **Bit 4 - Res: Reserved bit**

This bit is reserved in ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 3 - PRTIM1: Power Reduction Timer/Counter1**

Writing a logic one to this bit shuts down the Timer/Counter1 module. When the Timer/Counter1 is enabled, operation will continue like before the shutdown.

- **Bit 2 - PRSPI: Power Reduction Serial Peripheral Interface**

If using debugWIRE On-chip Debug System, this bit should not be written to one.

Writing a logic one to this bit shuts down the Serial Peripheral Interface by stopping the clock to the module. When waking up the SPI again, the SPI should be re initialized to ensure proper operation.

- **Bit 1 - PRUSART0: Power Reduction USART0**

Writing a logic one to this bit shuts down the USART by stopping the clock to the module. When waking up the USART again, the USART should be re initialized to ensure proper operation.

- **Bit 0 - PRADC: Power Reduction ADC**

Writing a logic one to this bit shuts down the ADC. The ADC must be disabled before shut down. The analog comparator cannot use the ADC input MUX when the ADC is shut down.

10. System Control and Reset

10.1 Resetting the AVR

During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. For the ATmega168PA, the instruction placed at the Reset Vector must be a JMP – Absolute Jump – instruction to the reset handling routine. For the ATmega48PA and ATmega88PA, the instruction placed at the Reset Vector must be an RJMP – Relative Jump – instruction to the reset handling routine. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa (ATmega88PA/168PA only). The circuit diagram in [Figure 10-1 on page 47](#) shows the reset logic. [Table 28-3 on page 318](#) defines the electrical parameters of the reset circuitry.

The I/O ports of the AVR are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

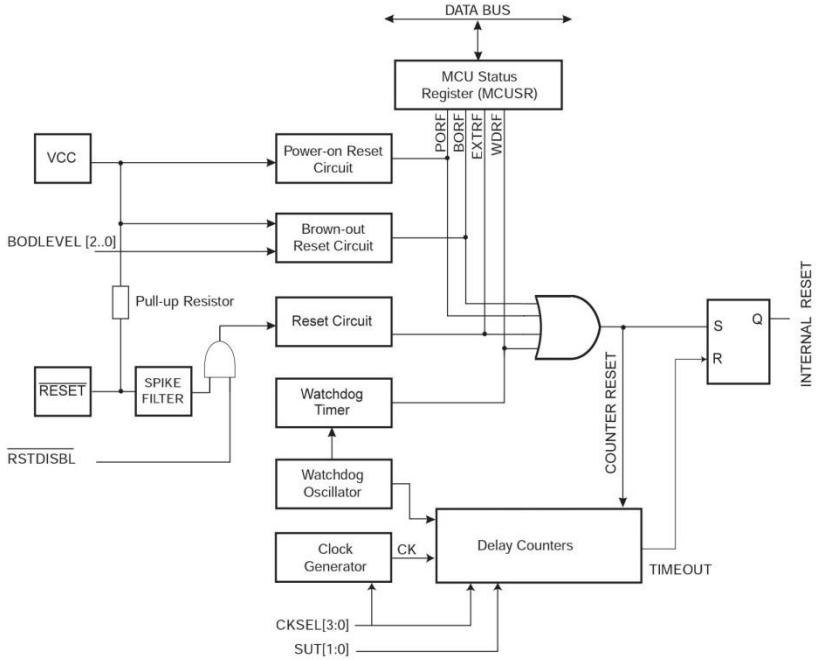
After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the SUT and CKSEL Fuses. The different selections for the delay period are presented in ["Clock Sources" on page 27](#).

10.2 Reset Sources

The ATmega48PA/88PA/168PA/328P has four sources of reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POT}).
- External Reset. The MCU is reset when a low level is present on the \overline{RESET} pin for longer than the minimum pulse length.
- Watchdog System Reset. The MCU is reset when the Watchdog Timer period expires and the Watchdog System Reset mode is enabled.
- Brown-out Reset. The MCU is reset when the supply voltage V_{CC} is below the Brown-out Reset threshold (V_{BOT}) and the Brown-out Detector is enabled.

Figure 10-1. Reset Logic



10.3 Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in "System and Reset Characteristics" on page 318. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the start-up Reset, as well as to detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after V_{CC} rise. The RESET signal is activated again, without any delay, when V_{CC} decreases below the detection level.

Figure 10-2. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC}

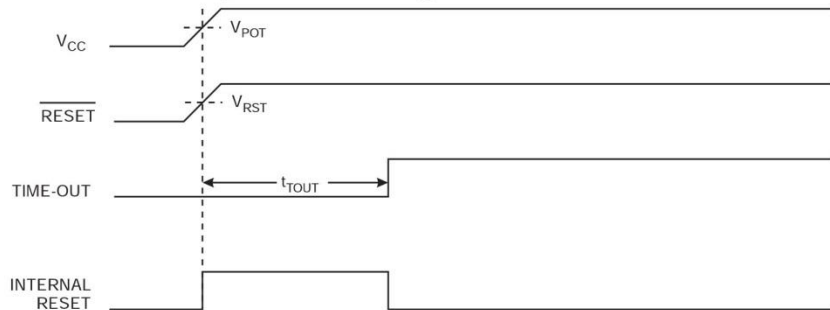
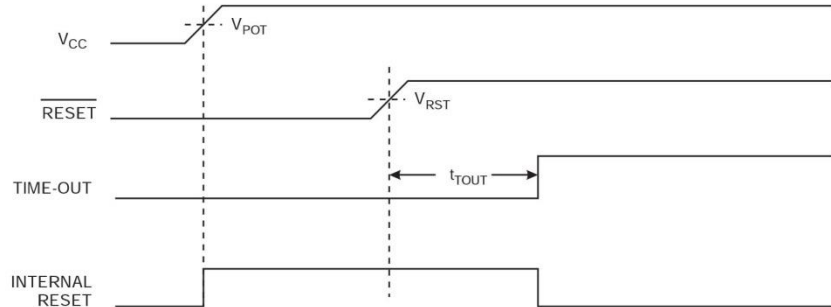


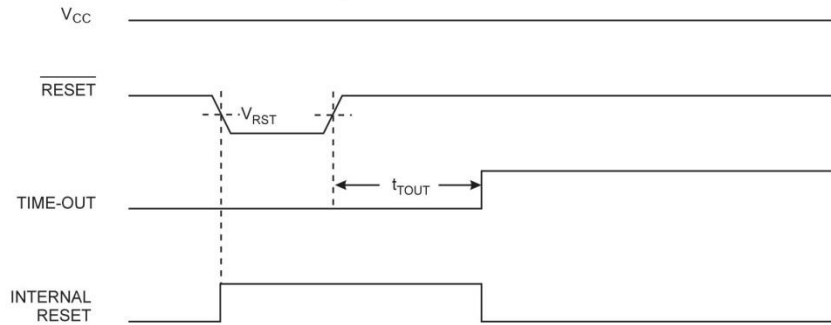
Figure 10-3. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



10.4 External Reset

An External Reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than the minimum pulse width (see “System and Reset Characteristics” on page 318) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage – V_{RST} – on its positive edge, the delay counter starts the MCU after the Time-out period – t_{TOUT} – has expired. The External Reset can be disabled by the RSTDISBL fuse, see Table 27-7 on page 296.

Figure 10-4. External Reset During Operation

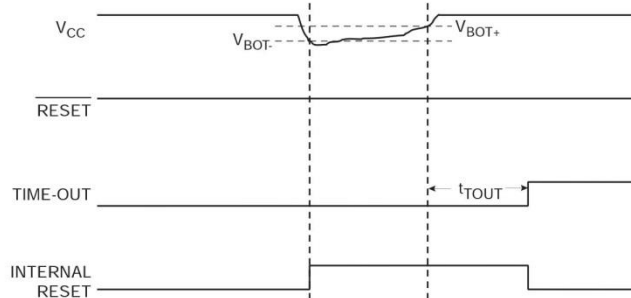


10.5 Brown-out Detection

ATmega48PA/88PA/168PA/328P has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the BODLEVEL Fuses. The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{\text{BOT+}} = V_{\text{BOT}} + V_{\text{HYST}}/2$ and $V_{\text{BOT-}} = V_{\text{BOT}} - V_{\text{HYST}}/2$. When the BOD is enabled, and V_{CC} decreases to a value below the trigger level ($V_{\text{BOT-}}$ in Figure 10-5 on page 49), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level ($V_{\text{BOT+}}$ in Figure 10-5 on page 49), the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in “System and Reset Characteristics” on page 318.

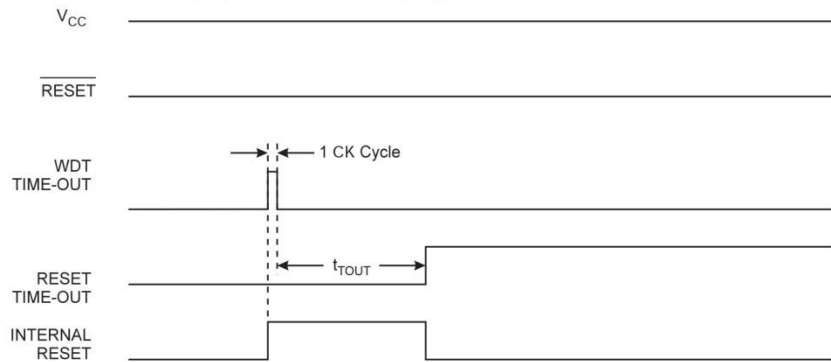
Figure 10-5. Brown-out Reset During Operation



10.6 Watchdog System Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT} . Refer to [page 50](#) for details on operation of the Watchdog Timer.

Figure 10-6. Watchdog System Reset During Operation



10.7 Internal Voltage Reference

ATmega48PA/88PA/168PA/328P features an internal bandgap reference. This reference is used for Brown-out Detection, and it can be used as an input to the Analog Comparator or the ADC.

10.7.1 Voltage Reference Enable Signals and Start-up Time

The voltage reference has a start-up time that may influence the way it should be used. The start-up time is given in "System and Reset Characteristics" on [page 318](#). To save power, the reference is not always turned on. The reference is on during the following situations:

1. When the BOD is enabled (by programming the BODLEVEL [2:0] Fuses).
2. When the bandgap reference is connected to the Analog Comparator (by setting the ACBG bit in ACSR).
3. When the ADC is enabled.

Thus, when the BOD is not enabled, after setting the ACBG bit or enabling the ADC, the user must always allow the reference to start up before the output from the Analog Comparator or

ADC is used. To reduce power consumption in Power-down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-down mode.

10.8 Watchdog Timer

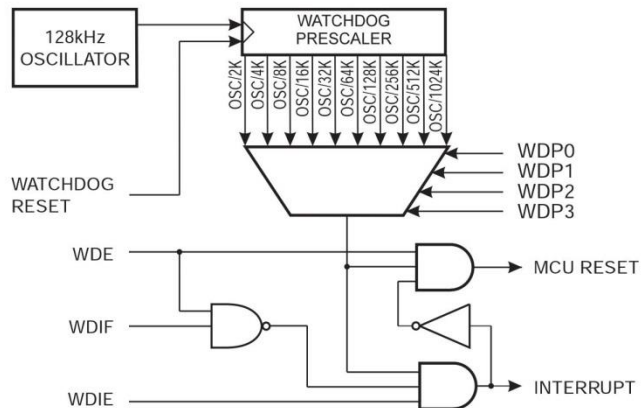
10.8.1 Features

- Clocked from separate On-chip Oscillator
- 3 Operating modes
 - Interrupt
 - System Reset
 - Interrupt and System Reset
- Selectable Time-out period from 16ms to 8s
- Possible Hardware fuse Watchdog always on (WDTON) for fail-safe mode

10.8.2 Overview

ATmega48PA/88PA/168PA/328P has an Enhanced Watchdog Timer (WDT). The WDT is a timer counting cycles of a separate on-chip 128 kHz oscillator. The WDT gives an interrupt or a system reset when the counter reaches a given time-out value. In normal operation mode, it is required that the system uses the WDR - Watchdog Timer Reset - instruction to restart the counter before the time-out value is reached. If the system doesn't restart the counter, an interrupt or system reset will be issued.

Figure 10-7. Watchdog Timer



In Interrupt mode, the WDT gives an interrupt when the timer expires. This interrupt can be used to wake the device from sleep-modes, and also as a general system timer. One example is to limit the maximum time allowed for certain operations, giving an interrupt when the operation has run longer than expected. In System Reset mode, the WDT gives a reset when the timer expires. This is typically used to prevent system hang-up in case of runaway code. The third mode, Interrupt and System Reset mode, combines the other two modes by first giving an interrupt and then switch to System Reset mode. This mode will for instance allow a safe shutdown by saving critical parameters before a system reset.

The Watchdog always on (WDTON) fuse, if programmed, will force the Watchdog Timer to System Reset mode. With the fuse programmed the System Reset mode bit (WDE) and Interrupt

mode bit (WDIE) are locked to 1 and 0 respectively. To further ensure program security, alterations to the Watchdog set-up must follow timed sequences. The sequence for clearing WDE and changing time-out configuration is as follows:

1. In the same operation, write a logic one to the Watchdog change enable bit (WDCE) and WDE. A logic one must be written to WDE regardless of the previous value of the WDE bit.
2. Within the next four clock cycles, write the WDE and Watchdog prescaler bits (WDP) as desired, but with the WDCE bit cleared. This must be done in one operation.

The following code example shows one assembly and one C function for turning off the Watchdog Timer. The example assumes that interrupts are controlled (e.g. by disabling interrupts globally) so that no interrupts will occur during the execution of these functions.

Assembly Code Example⁽¹⁾

```

WDT_off:
; Turn off global interrupt
cli
; Reset Watchdog Timer
wdr
; Clear WDRF in MCUSR
in    r16, MCUSR
andi  r16, (0xff & (0<<WDRF))
out   MCUSR, r16
; Write logical one to WDCE and WDE
; Keep old prescaler setting to prevent unintentional time-out
lds  r16, WDTCR
ori  r16, (1<<WDCE) | (1<<WDE)
sts  WDTCR, r16
; Turn off WDT
ldi  r16, (0<<WDE)
sts  WDTCR, r16
; Turn on global interrupt
sei
ret

```

C Code Example⁽¹⁾

```

void WDT_off(void)
{
    __disable_interrupt();
    __watchdog_reset();
    /* Clear WDRF in MCUSR */
    MCUSR &= ~(1<<WDRF);
    /* Write logical one to WDCE and WDE */
    /* Keep old prescaler setting to prevent unintentional time-out */
    WDTCR |= (1<<WDCE) | (1<<WDE);
    /* Turn off WDT */
    WDTCR = 0x00;
    __enable_interrupt();
}

```

Note: 1. See ["About Code Examples" on page 7](#).

Note: If the Watchdog is accidentally enabled, for example by a runaway pointer or brown-out condition, the device will be reset and the Watchdog Timer will stay enabled. If the code is not set up to handle the Watchdog, this might lead to an eternal loop of time-out resets. To avoid this situation, the application software should always clear the Watchdog System Reset Flag (WDRF) and the WDE control bit in the initialisation routine, even if the Watchdog is not in use.

The following code example shows one assembly and one C function for changing the time-out value of the Watchdog Timer.

Assembly Code Example⁽¹⁾

```
WDT_Prescaler_Change:
; Turn off global interrupt
cli
; Reset Watchdog Timer
wdr
; Start timed sequence
lds r16, WDTCR
ori r16, (1<<WDCE) | (1<<WDE)
sts WDTCR, r16
; -- Got four cycles to set the new values from here -
; Set new prescaler(time-out) value = 64K cycles (~0.5 s)
ldi r16, (1<<WDE) | (1<<WDP2) | (1<<WDP0)
sts WDTCR, r16
; -- Finished setting new values, used 2 cycles -
; Turn on global interrupt
sei
ret
```

C Code Example⁽¹⁾

```
void WDT_Prescaler_Change(void)
{
    __disable_interrupt();
    __watchdog_reset();
    /* Start timed equence */
    WDTCR |= (1<<WDCE) | (1<<WDE);
    /* Set new prescaler(time-out) value = 64K cycles (~0.5 s) */
    WDTCR = (1<<WDE) | (1<<WDP2) | (1<<WDP0);
    __enable_interrupt();
}
```

Note: 1. See "About Code Examples" on page 7.

Note: The Watchdog Timer should be reset before any change of the WDP bits, since a change in the WDP bits can result in a time-out when switching to a shorter time-out period.

10.9 Register Description

10.9.1 MCUSR – MCU Status Register

The MCU Status Register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0				See Bit Description	

- **Bit 7..4: Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 3 – WDRF: Watchdog System Reset Flag**

This bit is set if a Watchdog System Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 2 – BORF: Brown-out Reset Flag**

This bit is set if a Brown-out Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 1 – EXTRF: External Reset Flag**

This bit is set if an External Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 0 – PORF: Power-on Reset Flag**

This bit is set if a Power-on Reset occurs. The bit is reset only by writing a logic zero to the flag.

To make use of the Reset Flags to identify a reset condition, the user should read and then Reset the MCUSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the Reset Flags.

10.9.2 WDTCR – Watchdog Timer Control Register

Bit	7	6	5	4	3	2	1	0	
(0x60)	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	X	0	0	0	

- **Bit 7 - WDIF: Watchdog Interrupt Flag**

This bit is set when a time-out occurs in the Watchdog Timer and the Watchdog Timer is configured for interrupt. WDIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, WDIF is cleared by writing a logic one to the flag. When the I-bit in SREG and WDIE are set, the Watchdog Time-out Interrupt is executed.

- **Bit 6 - WDIE: Watchdog Interrupt Enable**

When this bit is written to one and the I-bit in the Status Register is set, the Watchdog Interrupt is enabled. If WDE is cleared in combination with this setting, the Watchdog Timer is in Interrupt Mode, and the corresponding interrupt is executed if time-out in the Watchdog Timer occurs. If WDE is set, the Watchdog Timer is in Interrupt and System Reset Mode. The first time-out in the

Watchdog Timer will set WDIF. Executing the corresponding interrupt vector will clear WDIE and WDIF automatically by hardware (the Watchdog goes to System Reset Mode). This is useful for keeping the Watchdog Timer security while using the interrupt. To stay in Interrupt and System Reset Mode, WDIE must be set after each interrupt. This should however not be done within the interrupt service routine itself, as this might compromise the safety-function of the Watchdog System Reset mode. If the interrupt is not executed before the next time-out, a System Reset will be applied.

Table 10-1. Watchdog Timer Configuration

WDTON ⁽¹⁾	WDE	WDIE	Mode	Action on Time-out
1	0	0	Stopped	None
1	0	1	Interrupt Mode	Interrupt
1	1	0	System Reset Mode	Reset
1	1	1	Interrupt and System Reset Mode	Interrupt, then go to System Reset Mode
0	x	x	System Reset Mode	Reset

Note: 1. WDTON Fuse set to "0" means programmed and "1" means unprogrammed.

• **Bit 4 - WDCE: Watchdog Change Enable**

This bit is used in timed sequences for changing WDE and prescaler bits. To clear the WDE bit, and/or change the prescaler bits, WDCE must be set.

Once written to one, hardware will clear WDCE after four clock cycles.

• **Bit 3 - WDE: Watchdog System Reset Enable**

WDE is overridden by WDRF in MCUSR. This means that WDE is always set when WDRF is set. To clear WDE, WDRF must be cleared first. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

• **Bit 5, 2..0 - WDP3..0: Watchdog Timer Prescaler 3, 2, 1 and 0**

The WDP3..0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is running. The different prescaling values and their corresponding time-out periods are shown in [Table 10-2 on page 55](#).

Table 10-2. Watchdog Timer Prescale Select

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V _{CC} = 5.0V
0	0	0	0	2K (2048) cycles	16 ms
0	0	0	1	4K (4096) cycles	32 ms
0	0	1	0	8K (8192) cycles	64 ms
0	0	1	1	16K (16384) cycles	0.125 s
0	1	0	0	32K (32768) cycles	0.25 s
0	1	0	1	64K (65536) cycles	0.5 s
0	1	1	0	128K (131072) cycles	1.0 s

Table 10-2. Watchdog Timer Prescale Select (Continued)

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at $V_{CC} = 5.0V$
0	1	1	1	256K (262144) cycles	2.0 s
1	0	0	0	512K (524288) cycles	4.0 s
1	0	0	1	1024K (1048576) cycles	8.0 s
1	0	1	0	Reserved	
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		
1	1	1	1		

11. Interrupts

This section describes the specifics of the interrupt handling as performed in ATmega48PA/88PA/168PA/328P. For a general explanation of the AVR interrupt handling, refer to ["Reset and Interrupt Handling" on page 14](#).

The interrupt vectors in ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P are generally the same, with the following differences:

- Each Interrupt Vector occupies two instruction words in ATmega168PA and ATmega328P, and one instruction word in ATmega48PA and ATmega88PA.
- ATmega48PA does not have a separate Boot Loader Section. In ATmega88PA, ATmega168PA and ATmega328P, the Reset Vector is affected by the BOOTRST fuse, and the Interrupt Vector start address is affected by the IVSEL bit in MCUCR.

11.1 Interrupt Vectors in ATmega48PA

Table 11-1. Reset and Interrupt Vectors in ATmega48PA

Vector No.	Program Address	Source	Interrupt Definition
1	0x000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	PCINT0	Pin Change Interrupt Request 0
5	0x004	PCINT1	Pin Change Interrupt Request 1
6	0x005	PCINT2	Pin Change Interrupt Request 2
7	0x006	WDT	Watchdog Time-out Interrupt
8	0x007	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x008	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x009	TIMER2 OVF	Timer/Counter2 Overflow
11	0x00A	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x00B	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x00C	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x00D	TIMER1 OVF	Timer/Counter1 Overflow
15	0x00E	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x00F	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x010	TIMER0 OVF	Timer/Counter0 Overflow
18	0x011	SPI, STC	SPI Serial Transfer Complete
19	0x012	USART, RX	USART Rx Complete
20	0x013	USART, UDRE	USART, Data Register Empty
21	0x014	USART, TX	USART, Tx Complete
22	0x015	ADC	ADC Conversion Complete
23	0x016	EE READY	EEPROM Ready

Table 11-1. Reset and Interrupt Vectors in ATmega48PA (Continued)

Vector No.	Program Address	Source	Interrupt Definition
24	0x017	ANALOG COMP	Analog Comparator
25	0x018	TWI	2-wire Serial Interface
26	0x019	SPM READY	Store Program Memory Ready

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega48PA is:

```

Address Labels Code           Comments
0x000      rjmp RESET        ; Reset Handler
0x001      rjmp EXT_INT0     ; IRQ0 Handler
0x002      rjmp EXT_INT1     ; IRQ1 Handler
0x003      rjmp PCINT0       ; PCINT0 Handler
0x004      rjmp PCINT1       ; PCINT1 Handler
0x005      rjmp PCINT2       ; PCINT2 Handler
0x006      rjmp WDT           ; Watchdog Timer Handler
0x007      rjmp TIM2_COMPA    ; Timer2 Compare A Handler
0x008      rjmp TIM2_COMPB    ; Timer2 Compare B Handler
0x009      rjmp TIM2_OVF     ; Timer2 Overflow Handler
0x00A      rjmp TIM1_CAPT     ; Timer1 Capture Handler
0x00B      rjmp TIM1_COMPA    ; Timer1 Compare A Handler
0x00C      rjmp TIM1_COMPB    ; Timer1 Compare B Handler
0x00D      rjmp TIM1_OVF     ; Timer1 Overflow Handler
0x00E      rjmp TIM0_COMPA    ; Timer0 Compare A Handler
0x00F      rjmp TIM0_COMPB    ; Timer0 Compare B Handler
0x010      rjmp TIM0_OVF     ; Timer0 Overflow Handler
0x011      rjmp SPI_STC      ; SPI Transfer Complete Handler
0x012      rjmp USART_RXC    ; USART, RX Complete Handler
0x013      rjmp USART_UDRE   ; USART, UDR Empty Handler
0x014      rjmp USART_TXC    ; USART, TX Complete Handler
0x015      rjmp ADC           ; ADC Conversion Complete Handler
0x016      rjmp EE_RDY       ; EEPROM Ready Handler
0x017      rjmp ANA_COMP     ; Analog Comparator Handler
0x018      rjmp TWI          ; 2-wire Serial Interface Handler
0x019      rjmp SPM_RDY     ; Store Program Memory Ready Handler
;
0x01ARESET: ldi   r16, high(RAMEND); Main program start
0x01B      out   SPH,r16      ; Set Stack Pointer to top of RAM
0x01C      ldi   r16, low(RAMEND)
0x01D      out   SPL,r16
0x01E      sei                       ; Enable interrupts
0x01F      <instr> xxx
...      ...      ...      ...

```

11.2 Interrupt Vectors in ATmega88PA

Table 11-2. Reset and Interrupt Vectors in ATmega88PA

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	PCINT0	Pin Change Interrupt Request 0
5	0x004	PCINT1	Pin Change Interrupt Request 1
6	0x005	PCINT2	Pin Change Interrupt Request 2
7	0x006	WDT	Watchdog Time-out Interrupt
8	0x007	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x008	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x009	TIMER2 OVF	Timer/Counter2 Overflow
11	0x00A	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x00B	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x00C	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x00D	TIMER1 OVF	Timer/Counter1 Overflow
15	0x00E	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x00F	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x010	TIMER0 OVF	Timer/Counter0 Overflow
18	0x011	SPI, STC	SPI Serial Transfer Complete
19	0x012	USART, RX	USART Rx Complete
20	0x013	USART, UDRE	USART, Data Register Empty
21	0x014	USART, TX	USART, Tx Complete
22	0x015	ADC	ADC Conversion Complete
23	0x016	EE READY	EEPROM Ready
24	0x017	ANALOG COMP	Analog Comparator
25	0x018	TWI	2-wire Serial Interface
26	0x019	SPM READY	Store Program Memory Ready

- Notes:
1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "[Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P](#)" on page 277.
 2. When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section.

Table 11-3 on page 60 shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 11-3. Reset and Interrupt Vectors Placement in ATmega88PA⁽¹⁾

BOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x001
1	1	0x000	Boot Reset Address + 0x001
0	0	Boot Reset Address	0x001
0	1	Boot Reset Address	Boot Reset Address + 0x001

Note: 1. The Boot Reset Address is shown in [Table 26-7 on page 289](#). For the BOTRST Fuse "1" means unprogrammed while "0" means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88PA is:

```

Address  Labels Code           Comments
0x000      rjmp  RESET           ; Reset Handler
0x001      rjmp  EXT_INT0        ; IRQ0 Handler
0x002      rjmp  EXT_INT1        ; IRQ1 Handler
0x003      rjmp  PCINT0          ; PCINT0 Handler
0x004      rjmp  PCINT1          ; PCINT1 Handler
0x005      rjmp  PCINT2          ; PCINT2 Handler
0x006      rjmp  WDT             ; Watchdog Timer Handler
0x007      rjmp  TIM2_COMPA       ; Timer2 Compare A Handler
0x008      rjmp  TIM2_COMPB       ; Timer2 Compare B Handler
0x009      rjmp  TIM2_OVF         ; Timer2 Overflow Handler
0x00A      rjmp  TIM1_CAPT        ; Timer1 Capture Handler
0x00B      rjmp  TIM1_COMPA       ; Timer1 Compare A Handler
0x00C      rjmp  TIM1_COMPB       ; Timer1 Compare B Handler
0x00D      rjmp  TIM1_OVF         ; Timer1 Overflow Handler
0x00E      rjmp  TIM0_COMPA       ; Timer0 Compare A Handler
0x00F      rjmp  TIM0_COMPB       ; Timer0 Compare B Handler
0x010      rjmp  TIM0_OVF         ; Timer0 Overflow Handler
0x011      rjmp  SPI_STC          ; SPI Transfer Complete Handler
0x012      rjmp  USART_RXC        ; USART, RX Complete Handler
0x013      rjmp  USART_UDRE       ; USART, UDR Empty Handler
0x014      rjmp  USART_TXC        ; USART, TX Complete Handler
0x015      rjmp  ADC              ; ADC Conversion Complete Handler
0x016      rjmp  EE_RDY           ; EEPROM Ready Handler
0x017      rjmp  ANA_COMP         ; Analog Comparator Handler
0x018      rjmp  TWI              ; 2-wire Serial Interface Handler
0x019      rjmp  SPM_RDY         ; Store Program Memory Ready Handler
;
0x01ARESET: ldi   r16, high(RAMEND); Main program start
0x01B      out   SPH,r16          ; Set Stack Pointer to top of RAM
0x01C      ldi   r16, low(RAMEND)
0x01D      out   SPL,r16
0x01E      sei                      ; Enable interrupts
0x01F      <instr> xxx
    
```

ATmega48PA/88PA/168PA/328P

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88PA is:

```
Address Labels Code           Comments
0x000  RESET: ldi    r16,high(RAMEND); Main program start
0x001          out    SPH,r16      ; Set Stack Pointer to top of RAM
0x002          ldi    r16,low(RAMEND)
0x003          out    SPL,r16
0x004          sei                      ; Enable interrupts
0x005          <instr> xxx
;
.org 0xC01
0xC01          rjmp   EXT_INT0      ; IRQ0 Handler
0xC02          rjmp   EXT_INT1      ; IRQ1 Handler
...           ...   ...           ;
0xC19          rjmp   SPM_RDY      ; Store Program Memory Ready Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2K bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88PA is:

```
Address Labels Code           Comments
.org 0x001
0x001          rjmp   EXT_INT0      ; IRQ0 Handler
0x002          rjmp   EXT_INT1      ; IRQ1 Handler
...           ...   ...           ;
0x019          rjmp   SPM_RDY      ; Store Program Memory Ready Handler
;
.org 0xC00
0xC00  RESET: ldi    r16,high(RAMEND); Main program start
0xC01          out    SPH,r16      ; Set Stack Pointer to top of RAM
0xC02          ldi    r16,low(RAMEND)
0xC03          out    SPL,r16
0xC04          sei                      ; Enable interrupts
0xC05          <instr> xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88PA is:

```
Address Labels Code           Comments
;
.org 0xC00
0xC00          rjmp   RESET         ; Reset handler
0xC01          rjmp   EXT_INT0      ; IRQ0 Handler
0xC02          rjmp   EXT_INT1      ; IRQ1 Handler
...           ...   ...           ;
0xC19          rjmp   SPM_RDY      ; Store Program Memory Ready Handler
;
0xC1A  RESET: ldi    r16,high(RAMEND); Main program start
```

ATmega48PA/88PA/168PA/328P

```

0xC1B      out    SPH,r16      ; Set Stack Pointer to top of RAM
0xC1C      ldi    r16,low(RAMEND)
0xC1D      out    SPL,r16
0xC1E      sei                      ; Enable interrupts
0xC1F      <instr> xxx

```

11.3 Interrupt Vectors in ATmega168PA

Table 11-4. Reset and Interrupt Vectors in ATmega168PA

VectorNo.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x0000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x001A	TIMER1 OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART Rx Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, Tx Complete
22	0x002A	ADC	ADC Conversion Complete
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	Store Program Memory Ready

- Notes:
1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P" on page 277.
 2. When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section.

Table 11-5 on page 63 shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 11-5. Reset and Interrupt Vectors Placement in ATmega168PA⁽¹⁾

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot Reset Address + 0x0002
0	0	Boot Reset Address	0x002
0	1	Boot Reset Address	Boot Reset Address + 0x0002

Note: 1. The Boot Reset Address is shown in Table 26-7 on page 289. For the BOOTRST Fuse "1" means unprogrammed while "0" means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168PA is:

```

Address  Labels Code           Comments
0x0000      jmp   RESET           ; Reset Handler
0x0002      jmp   EXT_INT0        ; IRQ0 Handler
0x0004      jmp   EXT_INT1        ; IRQ1 Handler
0x0006      jmp   PCINT0         ; PCINT0 Handler
0x0008      jmp   PCINT1         ; PCINT1 Handler
0x000A      jmp   PCINT2         ; PCINT2 Handler
0x000C      jmp   WDT           ; Watchdog Timer Handler
0x000E      jmp   TIM2_COMP_A    ; Timer2 Compare A Handler
0x0010      jmp   TIM2_COMP_B    ; Timer2 Compare B Handler
0x0012      jmp   TIM2_OVF       ; Timer2 Overflow Handler
0x0014      jmp   TIM1_CAPT      ; Timer1 Capture Handler
0x0016      jmp   TIM1_COMP_A    ; Timer1 Compare A Handler
0x0018      jmp   TIM1_COMP_B    ; Timer1 Compare B Handler
0x001A      jmp   TIM1_OVF       ; Timer1 Overflow Handler
0x001C      jmp   TIM0_COMP_A    ; Timer0 Compare A Handler
0x001E      jmp   TIM0_COMP_B    ; Timer0 Compare B Handler
0x0020      jmp   TIM0_OVF       ; Timer0 Overflow Handler
0x0022      jmp   SPI_STC        ; SPI Transfer Complete Handler
0x0024      jmp   USART_RXC      ; USART, RX Complete Handler
0x0026      jmp   USART_UDRE     ; USART, UDR Empty Handler
0x0028      jmp   USART_TXC      ; USART, TX Complete Handler
0x002A      jmp   ADC           ; ADC Conversion Complete Handler
0x002C      jmp   EE_RDY         ; EEPROM Ready Handler
0x002E      jmp   ANA_COMP       ; Analog Comparator Handler
0x0030      jmp   TWI           ; 2-wire Serial Interface Handler
0x0032      jmp   SPM_RDY       ; Store Program Memory Ready Handler
;
0x0033RESET: ldi    r16, high(RAMEND); Main program start
    
```


ATmega48PA/88PA/168PA/328P

```
0x0034      out    SPH,r16      ; Set Stack Pointer to top of RAM
0x0035      ldi    r16, low(RAMEND)
0x0036      out    SPL,r16
0x0037      sei                      ; Enable interrupts
0x0038      <instr> xxx
...         ...         ...         ...
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168PA is:

Address	Labels	Code	Comments
0x0000	RESET:	ldi r16,high(RAMEND);	Main program start
0x0001		out SPH,r16	; Set Stack Pointer to top of RAM
0x0002		ldi r16,low(RAMEND)	
0x0003		out SPL,r16	
0x0004		sei	; Enable interrupts
0x0005		<instr> xxx	
			;
		.org 0x1C02	
0x1C02		jmp EXT_INT0	; IRQ0 Handler
0x1C04		jmp EXT_INT1	; IRQ1 Handler
...		...	;
0x1C32		jmp SPM_RDY	; Store Program Memory Ready Handler

When the BOOTRST Fuse is programmed and the Boot section size set to 2K bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168PA is:

Address	Labels	Code	Comments
		.org 0x0002	
0x0002		jmp EXT_INT0	; IRQ0 Handler
0x0004		jmp EXT_INT1	; IRQ1 Handler
...		...	;
0x0032		jmp SPM_RDY	; Store Program Memory Ready Handler
			;
		.org 0x1C00	
0x1C00	RESET:	ldi r16,high(RAMEND);	Main program start
0x1C01		out SPH,r16	; Set Stack Pointer to top of RAM
0x1C02		ldi r16,low(RAMEND)	
0x1C03		out SPL,r16	
0x1C04		sei	; Enable interrupts
0x1C05		<instr> xxx	

When the BOOTRST Fuse is programmed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168PA is:

Address	Labels	Code	Comments
			;

```

.org 0x1C00
0x1C00      jmp     RESET           ; Reset handler
0x1C02      jmp     EXT_INT0        ; IRQ0 Handler
0x1C04      jmp     EXT_INT1        ; IRQ1 Handler
...         ...         ...         ;
0x1C32      jmp     SPM_RDY         ; Store Program Memory Ready Handler
;
0x1C33      RESET: ldi    r16,high(RAMEND); Main program start
0x1C34      out     SPH,r16         ; Set Stack Pointer to top of RAM
0x1C35      ldi    r16,low(RAMEND)
0x1C36      out     SPL,r16
0x1C37      sei                      ; Enable interrupts
0x1C38      <instr> xxx

```

11.4 Interrupt Vectors in ATmega328P

Table 11-6. Reset and Interrupt Vectors in ATmega328P

VectorNo.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x0000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x001A	TIMER1 OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART Rx Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, Tx Complete
22	0x002A	ADC	ADC Conversion Complete

Table 11-6. Reset and Interrupt Vectors in ATmega328P (Continued)

VectorNo.	Program Address ⁽²⁾	Source	Interrupt Definition
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	Store Program Memory Ready

- Notes: 1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P" on page 277.
 2. When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section.

Table 11-7 on page 66 shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 11-7. Reset and Interrupt Vectors Placement in ATmega328P⁽¹⁾

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot Reset Address + 0x0002
0	0	Boot Reset Address	0x002
0	1	Boot Reset Address	Boot Reset Address + 0x0002

- Note: 1. The Boot Reset Address is shown in Table 26-7 on page 289. For the BOOTRST Fuse "1" means unprogrammed while "0" means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328P is:

```

Address  Labels Code           Comments
0x0000      jmp  RESET           ; Reset Handler
0x0002      jmp  EXT_INT0        ; IRQ0 Handler
0x0004      jmp  EXT_INT1        ; IRQ1 Handler
0x0006      jmp  PCINT0         ; PCINT0 Handler
0x0008      jmp  PCINT1         ; PCINT1 Handler
0x000A      jmp  PCINT2         ; PCINT2 Handler
0x000C      jmp  WDT           ; Watchdog Timer Handler
0x000E      jmp  TIM2_COMP_A    ; Timer2 Compare A Handler
0x0010      jmp  TIM2_COMP_B    ; Timer2 Compare B Handler
0x0012      jmp  TIM2_OVF       ; Timer2 Overflow Handler
0x0014      jmp  TIM1_CAPT      ; Timer1 Capture Handler
0x0016      jmp  TIM1_COMP_A    ; Timer1 Compare A Handler
0x0018      jmp  TIM1_COMP_B    ; Timer1 Compare B Handler
0x001A      jmp  TIM1_OVF       ; Timer1 Overflow Handler
0x001C      jmp  TIM0_COMP_A    ; Timer0 Compare A Handler
0x001E      jmp  TIM0_COMP_B    ; Timer0 Compare B Handler
    
```

ATmega48PA/88PA/168PA/328P

```

0x0020      jmp    TIM0_OVF      ; Timer0 Overflow Handler
0x0022      jmp    SPI_STC      ; SPI Transfer Complete Handler
0x0024      jmp    USART_RXC    ; USART, RX Complete Handler
0x0026      jmp    USART_UDRE   ; USART, UDR Empty Handler
0x0028      jmp    USART_TXC    ; USART, TX Complete Handler
0x002A      jmp    ADC          ; ADC Conversion Complete Handler
0x002C      jmp    EE_RDY      ; EEPROM Ready Handler
0x002E      jmp    ANA_COMP     ; Analog Comparator Handler
0x0030      jmp    TWI         ; 2-wire Serial Interface Handler
0x0032      jmp    SPM_RDY     ; Store Program Memory Ready Handler
;
0x0033RESET: ldi    r16, high(RAMEND); Main program start
0x0034      out    SPH,r16      ; Set Stack Pointer to top of RAM
0x0035      ldi    r16, low(RAMEND)
0x0036      out    SPL,r16
0x0037      sei                      ; Enable interrupts
0x0038      <instr> xxx
...

```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328P is:

Address	Labels	Code	Comments
0x0000	RESET:	ldi r16,high(RAMEND);	Main program start
0x0001		out SPH,r16	; Set Stack Pointer to top of RAM
0x0002		ldi r16,low(RAMEND)	
0x0003		out SPL,r16	
0x0004		sei	; Enable interrupts
0x0005		<instr> xxx	
;			
.org 0x3C02			
0x3C02		jmp EXT_INT0	; IRQ0 Handler
0x3C04		jmp EXT_INT1	; IRQ1 Handler
...	;
0x3C32		jmp SPM_RDY	; Store Program Memory Ready Handler

When the BOOTRST Fuse is programmed and the Boot section size set to 2K bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328P is:

Address	Labels	Code	Comments
.org 0x0002			
0x0002		jmp EXT_INT0	; IRQ0 Handler
0x0004		jmp EXT_INT1	; IRQ1 Handler
...	;
0x0032		jmp SPM_RDY	; Store Program Memory Ready Handler
;			
.org 0x3C00			
0x3C00	RESET:	ldi r16,high(RAMEND);	Main program start

ATmega48PA/88PA/168PA/328P

```

0x3C01      out    SPH,r16      ; Set Stack Pointer to top of RAM
0x3C02      ldi    r16,low(RAMEND)
0x3C03      out    SPL,r16
0x3C04      sei                      ; Enable interrupts
0x3C05      <instr> xxx

```

When the BOTRST Fuse is programmed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328P is:

```

Address Labels Code          Comments
;
.org 0x3C00
0x3C00      jmp    RESET      ; Reset handler
0x3C02      jmp    EXT_INT0   ; IRQ0 Handler
0x3C04      jmp    EXT_INT1   ; IRQ1 Handler
...
0x3C32      jmp    SPM_RDY    ; Store Program Memory Ready Handler
;
0x3C33  RESET: ldi    r16,high(RAMEND); Main program start
0x3C34      out    SPH,r16    ; Set Stack Pointer to top of RAM
0x3C35      ldi    r16,low(RAMEND)
0x3C36      out    SPL,r16
0x3C37      sei                      ; Enable interrupts
0x3C38      <instr> xxx

```

11.5 Register Description

11.5.1 Moving Interrupts Between Application and Boot Space, ATmega88PA, ATmega168PA and ATmega328P

The MCU Control Register controls the placement of the Interrupt Vector table.

11.5.2 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – IVSEL: Interrupt Vector Select**

When the IVSEL bit is cleared (zero), the Interrupt Vectors are placed at the start of the Flash memory. When this bit is set (one), the Interrupt Vectors are moved to the beginning of the Boot Loader section of the Flash. The actual address of the start of the Boot Flash Section is determined by the BOOTSZ Fuses. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P" on page 277 for details. To avoid unintentional changes of Interrupt Vector tables, a special write procedure must be followed to change the IVSEL bit:

- Write the Interrupt Vector Change Enable (IVCE) bit to one.
- Within four cycles, write the desired value to IVSEL while writing a zero to IVCE.

Interrupts will automatically be disabled while this sequence is executed. Interrupts are disabled in the cycle IVCE is set, and they remain disabled until after the instruction following the write to



IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I-bit in the Status Register is unaffected by the automatic disabling.

Note: If Interrupt Vectors are placed in the Boot Loader section and Boot Lock bit BLB02 is programmed, interrupts are disabled while executing from the Application section. If Interrupt Vectors are placed in the Application section and Boot Lock bit BLB12 is programmed, interrupts are disabled while executing from the Boot Loader section. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P" on page 277 for details on Boot Lock bits.

- **Bit 0 – IVCE: Interrupt Vector Change Enable**

The IVCE bit must be written to logic one to enable change of the IVSEL bit. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. Setting the IVCE bit will disable interrupts, as explained in the IVSEL description above. See Code Example below.

Assembly Code Example

```
Move_interrupts:
    ; Enable change of Interrupt Vectors
    ldi r16, (1<<IVCE)
    out MCUCR, r16
    ; Move interrupts to Boot Flash section
    ldi r16, (1<<IVSEL)
    out MCUCR, r16
    ret
```

C Code Example

```
void Move_interrupts(void)
{
    /* Enable change of Interrupt Vectors */
    MCUCR = (1<<IVCE);
    /* Move interrupts to Boot Flash section */
    MCUCR = (1<<IVSEL);
}
```

12. External Interrupts

The External Interrupts are triggered by the INT0 and INT1 pins or any of the PCINT23..0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0 and INT1 or PCINT23..0 pins are configured as outputs. This feature provides a way of generating a software interrupt. The pin change interrupt PCI2 will trigger if any enabled PCINT23..16 pin toggles. The pin change interrupt PCI1 will trigger if any enabled PCINT14..8 pin toggles. The pin change interrupt PCI0 will trigger if any enabled PCINT7..0 pin toggles. The PCMSK2, PCMSK1 and PCMSK0 Registers control which pins contribute to the pin change interrupts. Pin change interrupts on PCINT23..0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

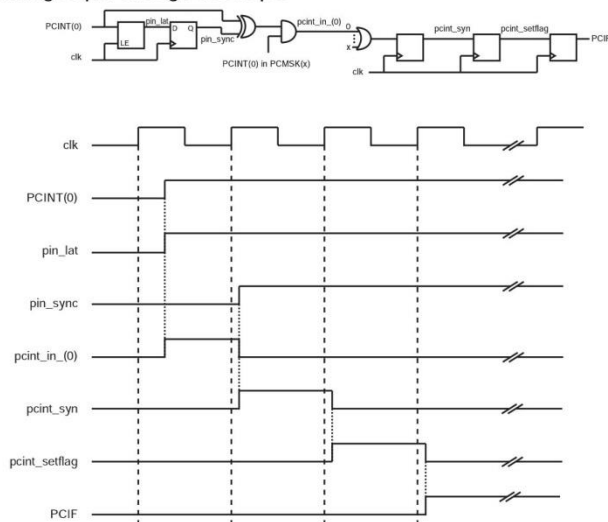
The INT0 and INT1 interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the External Interrupt Control Register A – EICRA. When the INT0 or INT1 interrupts are enabled and are configured as level triggered, the interrupts will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 or INT1 requires the presence of an I/O clock, described in "Clock Systems and their Distribution" on page 26. Low level interrupt on INT0 and INT1 is detected asynchronously. This implies that this interrupt can be used for waking the part also from sleep modes other than Idle mode. The I/O clock is halted in all sleep modes except Idle mode.

Note that if a level triggered interrupt is used for wake-up from Power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the Start-up Time, the MCU will still wake up, but no interrupt will be generated. The start-up time is defined by the SUT and CKSEL Fuses as described in "System Clock and Clock Options" on page 26.

12.1 Pin Change Interrupt Timing

An example of timing of a pin change interrupt is shown in Figure 12-1.

Figure 12-1. Timing of pin change interrupts



12.2 Register Description

12.2.1 EICRA – External Interrupt Control Register A

The External Interrupt Control Register A contains control bits for interrupt sense control.

Bit (0x69)	7	6	5	4	3	2	1	0	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..4 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0**

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 12-1. The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 12-1. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

- **Bit 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 12-2. The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 12-2. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

12.2.2 EIMSK – External Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x1D (0x3D)	-	-	-	-	-	-	INT1	INT0	EIMSK
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..2 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 1 – INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the External Interrupt Control Register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from the INT1 Interrupt Vector.

- **Bit 0 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the External Interrupt Control Register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from the INT0 Interrupt Vector.

12.2.3 EIFR – External Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	-	-	-	-	-	-	INTF1	INTF0	EIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..2 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 1 – INTF1: External Interrupt Flag 1**

When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in EIMSK are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

- **Bit 0 – INTF0: External Interrupt Flag 0**

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in EIMSK are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

12.2.4 PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..3 - Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 2 - PCIE2: Pin Change Interrupt Enable 2**

When the PCIE2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 2 is enabled. Any change on any enabled PCINT23..16 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PC12 Interrupt Vector. PCINT23..16 pins are enabled individually by the PCMSK2 Register.

- **Bit 1 - PCIE1: Pin Change Interrupt Enable 1**

When the PCIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT14..8 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PC11 Interrupt Vector. PCINT14..8 pins are enabled individually by the PCMSK1 Register.

- **Bit 0 - PCIE0: Pin Change Interrupt Enable 0**

When the PCIE0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT7..0 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PC10 Interrupt Vector. PCINT7..0 pins are enabled individually by the PCMSK0 Register.

12.2.5 PCIFR – Pin Change Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	–	–	–	–	–	PCIF2	PCIF1	PCIF0	PCIFR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..3 - Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 2 - PCIF2: Pin Change Interrupt Flag 2**

When a logic change on any PCINT23..16 pin triggers an interrupt request, PCIF2 becomes set (one). If the I-bit in SREG and the PCIE2 bit in PCICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bit 1 - PCIF1: Pin Change Interrupt Flag 1**

When a logic change on any PCINT14..8 pin triggers an interrupt request, PCIF1 becomes set (one). If the I-bit in SREG and the PCIE1 bit in PCICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bit 0 - PCIF0: Pin Change Interrupt Flag 0**

When a logic change on any PCINT7..0 pin triggers an interrupt request, PCIF0 becomes set (one). If the I-bit in SREG and the PCIE0 bit in PCICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

12.2.6 PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23 PCINT22 PCINT21 PCINT20 PCINT19 PCINT18 PCINT17 PCINT16								PCMSK2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..0 – PCINT23..16: Pin Change Enable Mask 23..16**

Each PCINT23..16-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT23..16 is set and the PCIE2 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT23..16 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

12.2.7 PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
(0x6C)	–	PCINT14 PCINT13 PCINT12 PCINT11 PCINT10 PCINT9 PCINT8							PCMSK1
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

This bit is an unused bit in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 6..0 – PCINT14..8: Pin Change Enable Mask 14..8**

Each PCINT14..8-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT14..8 is set and the PCIE1 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT14..8 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

12.2.8 PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7 PCINT6 PCINT5 PCINT4 PCINT3 PCINT2 PCINT1 PCINT0								PCMSK0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..0 – PCINT7..0: Pin Change Enable Mask 7..0**

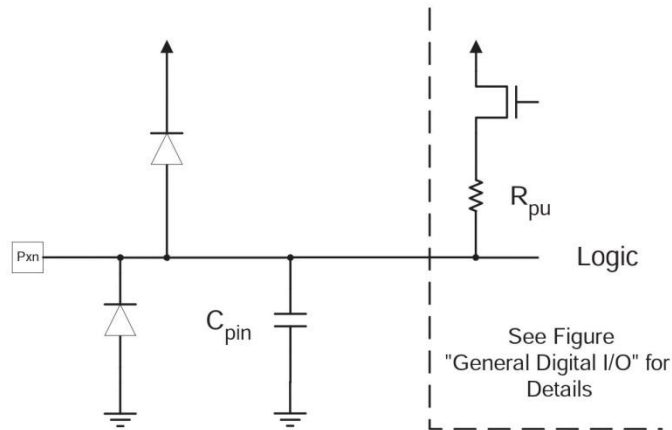
Each PCINT7..0 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is set and the PCIE0 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

13. I/O-Ports

13.1 Overview

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. The pin driver is strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and Ground as indicated in Figure 13-1. Refer to “Electrical Characteristics” on page 313 for a complete list of parameters.

Figure 13-1. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in “Register Description” on page 92.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. However, writing a logic one to a bit in the PINx Register, will result in a toggle in the corresponding bit in the Data Register. In addition, the Pull-up Disable – PUD bit in MCUCR disables the pull-up function for all pins in all ports when set.

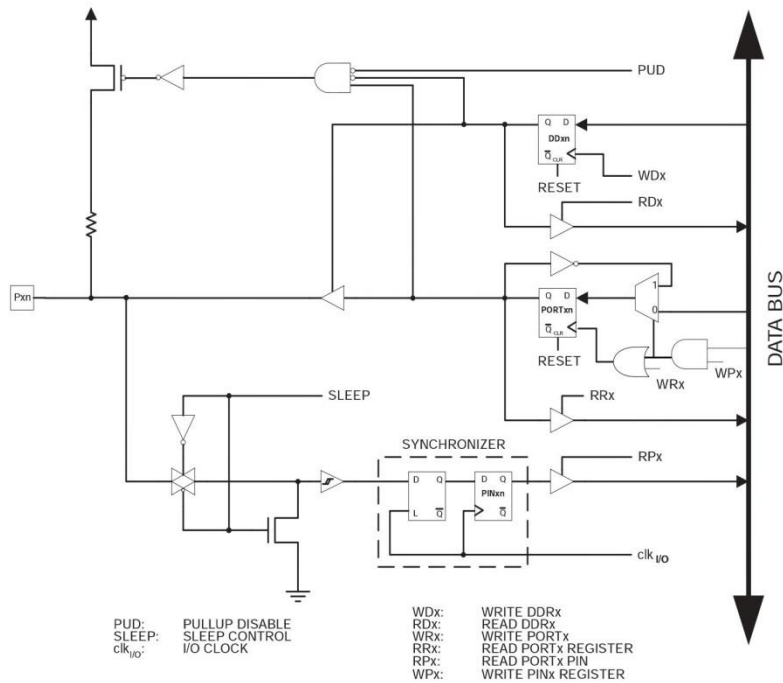
Using the I/O port as General Digital I/O is described in “Ports as General Digital I/O” on page 76. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in “Alternate Port Functions” on page 80. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

13.2 Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 13-2 shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 13-2. General Digital I/O⁽¹⁾



Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. clk_{I/O}, SLEEP, and PUD are common to all ports.

13.2.1 Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in "Register Description" on page 92, the DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

13.2.2 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn. Note that the SBI instruction can be used to toggle one single bit in a port.

13.2.3 Switching Between Input and Output

When switching between tri-state ({DDxn, PORTxn} = 0b00) and output high ({DDxn, PORTxn} = 0b11), an intermediate state with either pull-up enabled ({DDxn, PORTxn} = 0b01) or output low ({DDxn, PORTxn} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedance environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b11) as an intermediate step.

Table 13-1 summarizes the control signals for the pin value.

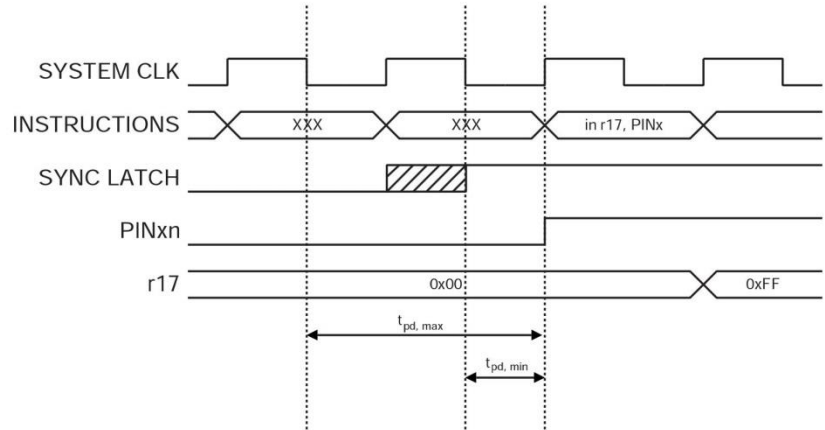
Table 13-1. Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

13.2.4 Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register bit. As shown in Figure 13-2, the PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 13-3 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$ respectively.

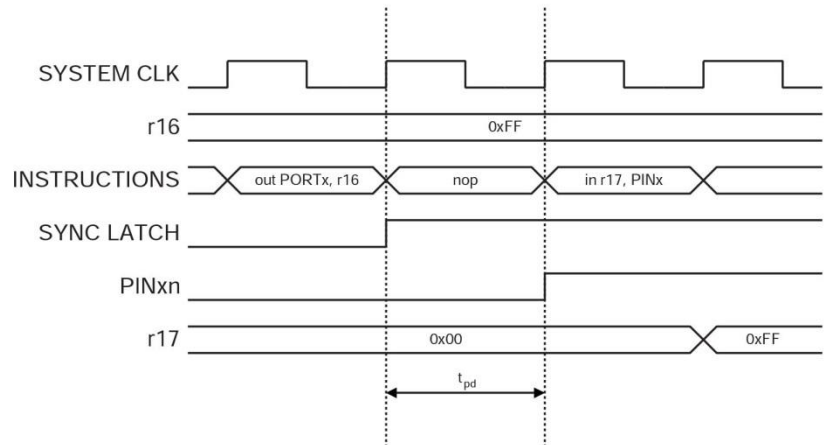
Figure 13-3. Synchronization when Reading an Externally Applied Pin value



Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the "SYNC LATCH" signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn Register at the succeeding positive clock edge. As indicated by the two arrows $t_{pd,max}$ and $t_{pd,min}$, a single signal transition on the pin will be delayed between $\frac{1}{2}$ and $1\frac{1}{2}$ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a nop instruction must be inserted as indicated in Figure 13-4. The out instruction sets the "SYNC LATCH" signal at the positive edge of the clock. In this case, the delay t_{pd} through the synchronizer is 1 system clock period.

Figure 13-4. Synchronization when Reading a Software Assigned Pin Value



The following code example shows how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 7 as input with pull-ups assigned to port pins 6 and 7. The resulting pin values are read back again, but as previously discussed, a nop instruction is included to be able to read back the value recently assigned to some of the pins.

Assembly Code Example⁽¹⁾

```

...
; Define pull-ups and set outputs high
; Define directions for port pins
ldi r16, (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0)
ldi r17, (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0)
out PORTB, r16
out DDRB, r17
; Insert nop for synchronization
nop
; Read port pins
in r16, PINB
...

```

C Code Example

```

unsigned char i;
...
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0);
DDRB = (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
/* Insert nop for synchronization*/
__no_operation();
/* Read port pins */
i = PINB;
...

```

Note: 1. For the assembly program, two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1, 6, and 7, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

13.2.5 Digital Input Enable and Sleep Modes

As shown in [Figure 13-2](#), the digital input signal can be clamped to ground at the input of the Schmitt Trigger. The signal denoted SLEEP in the figure, is set by the MCU Sleep Controller in Power-down mode, Power-save mode, and Standby mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to $V_{CC}/2$.

SLEEP is overridden for port pins enabled as external interrupt pins. If the external interrupt request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in ["Alternate Port Functions" on page 80](#).

If a logic high level ("one") is present on an asynchronous external interrupt pin configured as "Interrupt on Rising Edge, Falling Edge, or Any Logic Change on Pin" while the external interrupt is *not* enabled, the corresponding External Interrupt Flag will be set when resuming from the above mentioned Sleep mode, as the clamping in these sleep mode produces the requested logic change.

13.2.6 Unconnected Pins

If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, float-

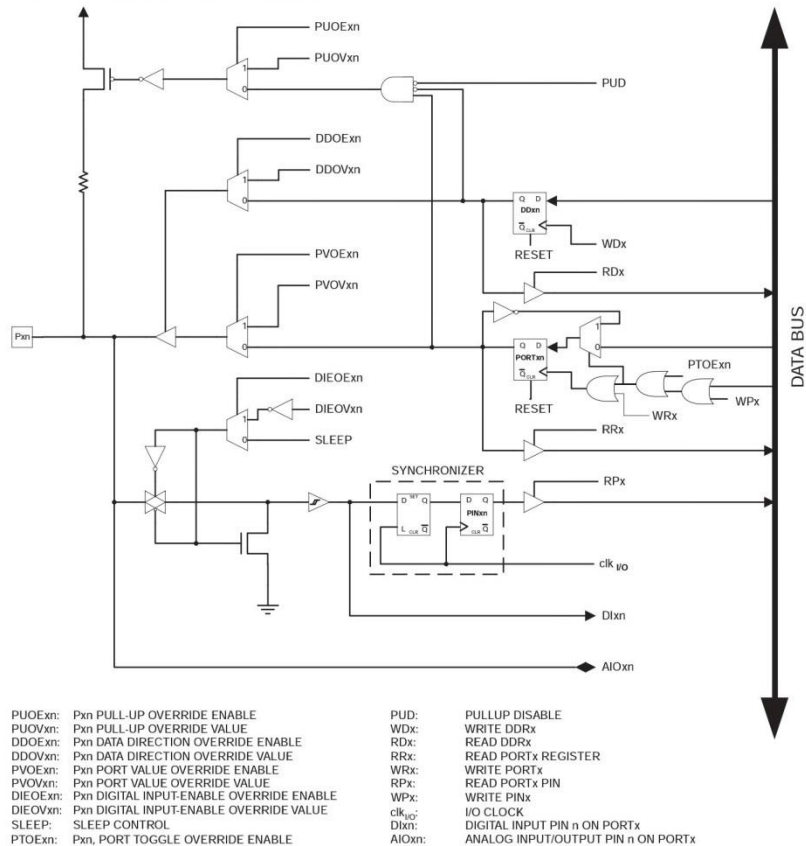
ing inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

13.3 Alternate Port Functions

Most port pins have alternate functions in addition to being general digital I/Os. Figure 13-5 shows how the port pin control signals from the simplified Figure 13-2 on page 76 can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the figure serves as a generic description applicable to all port pins in the AVR microcontroller family.

Figure 13-5. Alternate Port Functions⁽¹⁾



Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. clk_{I/O}, SLEEP, and PUD are common to all ports. All other signals are unique for each pin.

Table 13-2 summarizes the function of the overriding signals. The pin and port indexes from Figure 13-5 on page 80 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

Table 13-2. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUOE	Pull-up Override Enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-up Override Value	If PUOE is set, the pull-up is enabled/disabled when PUOV is set/cleared, regardless of the setting of the DDxn, PORTxn, and PUD Register bits.
DDOE	Data Direction Override Enable	If this signal is set, the Output Driver Enable is controlled by the DDOV signal. If this signal is cleared, the Output driver is enabled by the DDxn Register bit.
DDOV	Data Direction Override Value	If DDOE is set, the Output Driver is enabled/disabled when DDOV is set/cleared, regardless of the setting of the DDxn Register bit.
PVOE	Port Value Override Enable	If this signal is set and the Output Driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the Output Driver is enabled, the port Value is controlled by the PORTxn Register bit.
PVOV	Port Value Override Value	If PVOE is set, the port value is set to PVOV, regardless of the setting of the PORTxn Register bit.
PTOE	Port Toggle Override Enable	If PTOE is set, the PORTxn Register bit is inverted.
DIEOE	Digital Input Enable Override Enable	If this bit is set, the Digital Input Enable is controlled by the DIEOV signal. If this signal is cleared, the Digital Input Enable is determined by MCU state (Normal mode, sleep mode).
DIEOV	Digital Input Enable Override Value	If DIEOE is set, the Digital Input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state (Normal mode, sleep mode).
DI	Digital Input	This is the Digital Input to alternate functions. In the figure, the signal is connected to the output of the Schmitt Trigger but before the synchronizer. Unless the Digital Input is used as a clock source, the module with the alternate function will use its own synchronizer.
AIO	Analog Input/Output	This is the Analog Input/output to/from alternate functions. The signal is connected directly to the pad, and can be used bi-directionally.

The following subsections shortly describe the alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

13.3.1 Alternate Functions of Port B

The Port B pins with alternate functions are shown in [Table 13-3](#).

Table 13-3. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	\overline{SS} (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

The alternate pin configuration is as follows:

- **XTAL2/TOSC2/PCINT7 – Port B, Bit 7**

XTAL2: Chip clock Oscillator pin 2. Used as clock pin for crystal Oscillator or Low-frequency crystal Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

TOSC2: Timer Oscillator pin 2. Used only if internal calibrated RC Oscillator is selected as chip clock source, and the asynchronous timer is enabled by the correct setting in ASSR. When the AS2 bit in ASSR is set (one) and the EXCLK bit is cleared (zero) to enable asynchronous clocking of Timer/Counter2 using the Crystal Oscillator, pin PB7 is disconnected from the port, and becomes the inverting output of the Oscillator amplifier. In this mode, a crystal Oscillator is connected to this pin, and the pin cannot be used as an I/O pin.

PCINT7: Pin Change Interrupt source 7. The PB7 pin can serve as an external interrupt source. If PB7 is used as a clock pin, DDB7, PORTB7 and PINB7 will all read 0.

- **XTAL1/TOSC1/PCINT6 – Port B, Bit 6**

XTAL1: Chip clock Oscillator pin 1. Used for all chip clock sources except internal calibrated RC Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

TOSC1: Timer Oscillator pin 1. Used only if internal calibrated RC Oscillator is selected as chip clock source, and the asynchronous timer is enabled by the correct setting in ASSR. When the

AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pin PB6 is disconnected from the port, and becomes the input of the inverting Oscillator amplifier. In this mode, a crystal Oscillator is connected to this pin, and the pin can not be used as an I/O pin.

PCINT6: Pin Change Interrupt source 6. The PB6 pin can serve as an external interrupt source. If PB6 is used as a clock pin, DDB6, PORTB6 and PINB6 will all read 0.

- **SCK/PCINT5 – Port B, Bit 5**

SCK: Master Clock output, Slave Clock input pin for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB5. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB5. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB5 bit.

PCINT5: Pin Change Interrupt source 5. The PB5 pin can serve as an external interrupt source.

- **MISO/PCINT4 – Port B, Bit 4**

MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a Master, this pin is configured as an input regardless of the setting of DDB4. When the SPI is enabled as a Slave, the data direction of this pin is controlled by DDB4. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB4 bit.

PCINT4: Pin Change Interrupt source 4. The PB4 pin can serve as an external interrupt source.

- **MOSI/OC2/PCINT3 – Port B, Bit 3**

MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB3. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB3. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB3 bit.

OC2, Output Compare Match Output: The PB3 pin can serve as an external output for the Timer/Counter2 Compare Match. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC2 pin is also the output pin for the PWM mode timer function.

PCINT3: Pin Change Interrupt source 3. The PB3 pin can serve as an external interrupt source.

- **\overline{SS} /OC1B/PCINT2 – Port B, Bit 2**

\overline{SS} : Slave Select input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB2. As a Slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB2. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB2 bit.

OC1B, Output Compare Match output: The PB2 pin can serve as an external output for the Timer/Counter1 Compare Match B. The PB2 pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

PCINT2: Pin Change Interrupt source 2. The PB2 pin can serve as an external interrupt source.

- **OC1A/PCINT1 – Port B, Bit 1**

OC1A, Output Compare Match output: The PB1 pin can serve as an external output for the Timer/Counter1 Compare Match A. The PB1 pin has to be configured as an output (DDB1 set

(one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.

PCINT1: Pin Change Interrupt source 1. The PB1 pin can serve as an external interrupt source.

• **ICP1/CLKO/PCINT0 – Port B, Bit 0**

ICP1, Input Capture Pin: The PB0 pin can act as an Input Capture Pin for Timer/Counter1.

CLKO, Divided System Clock: The divided system clock can be output on the PB0 pin. The divided system clock will be output if the CKOUT Fuse is programmed, regardless of the PORTB0 and DDB0 settings. It will also be output during reset.

PCINT0: Pin Change Interrupt source 0. The PB0 pin can serve as an external interrupt source.

Table 13-4 and Table 13-5 on page 85 relate the alternate functions of Port B to the overriding signals shown in Figure 13-5 on page 80. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 13-4. Overriding Signals for Alternate Functions in PB7..PB4

Signal Name	PB7/XTAL2/ TOSC2/PCINT7 ⁽¹⁾	PB6/XTAL1/ TOSC1/PCINT6 ⁽¹⁾	PB5/SCK/ PCINT5	PB4/MISO/ PCINT4
PUOE	$\overline{\text{INTRC}} \cdot \overline{\text{EXTCK}} + \text{AS2}$	$\overline{\text{INTRC}} + \text{AS2}$	$\text{SPE} \cdot \overline{\text{MSTR}}$	$\text{SPE} \cdot \text{MSTR}$
PUOV	0	0	$\text{PORTB5} \cdot \overline{\text{PUD}}$	$\text{PORTB4} \cdot \overline{\text{PUD}}$
DDOE	$\overline{\text{INTRC}} \cdot \overline{\text{EXTCK}} + \text{AS2}$	$\overline{\text{INTRC}} + \text{AS2}$	$\text{SPE} \cdot \overline{\text{MSTR}}$	$\text{SPE} \cdot \text{MSTR}$
DDOV	0	0	0	0
PVOE	0	0	$\text{SPE} \cdot \text{MSTR}$	$\text{SPE} \cdot \overline{\text{MSTR}}$
PVOV	0	0	SCK OUTPUT	SPI SLAVE OUTPUT
DIEOE	$\overline{\text{INTRC}} \cdot \overline{\text{EXTCK}} + \text{AS2} + \text{PCINT7} \cdot \text{PCIE0}$	$\overline{\text{INTRC}} + \text{AS2} + \text{PCINT6} \cdot \text{PCIE0}$	$\text{PCINT5} \cdot \text{PCIE0}$	$\text{PCINT4} \cdot \text{PCIE0}$
DIEOV	$(\text{INTRC} + \overline{\text{EXTCK}}) \cdot \text{AS2}$	$\text{INTRC} \cdot \overline{\text{AS2}}$	1	1
DI	PCINT7 INPUT	PCINT6 INPUT	PCINT5 INPUT SCK INPUT	PCINT4 INPUT SPI MSTR INPUT
AIO	Oscillator Output	Oscillator/Clock Input	–	–

Notes: 1. INTRC means that one of the internal RC Oscillators are selected (by the CKSEL fuses), EXTCK means that external clock is selected (by the CKSEL fuses)

Table 13-5. Overriding Signals for Alternate Functions in PB3..PB0

Signal Name	PB3/MOSI/ OC2/PCINT3	PB2/SS/ OC1B/PCINT2	PB1/OC1A/ PCINT1	PB0/ICP1/ PCINT0
PUOE	SPE · $\overline{\text{MSTR}}$	SPE · $\overline{\text{MSTR}}$	0	0
PUOV	PORTB3 · $\overline{\text{PUD}}$	PORTB2 · $\overline{\text{PUD}}$	0	0
DDOE	SPE · $\overline{\text{MSTR}}$	SPE · $\overline{\text{MSTR}}$	0	0
DDOV	0	0	0	0
PVOE	SPE · MSTR + OC2A ENABLE	OC1B ENABLE	OC1A ENABLE	0
PVOV	SPI MSTR OUTPUT + OC2A	OC1B	OC1A	0
DIEOE	PCINT3 · PCIE0	PCINT2 · PCIE0	PCINT1 · PCIE0	PCINT0 · PCIE0
DIEOV	1	1	1	1
DI	PCINT3 INPUT SPI SLAVE INPUT	PCINT2 INPUT SPI $\overline{\text{SS}}$	PCINT1 INPUT	PCINT0 INPUT ICP1 INPUT
AIO	–	–	–	–

13.3.2 Alternate Functions of Port C

The Port C pins with alternate functions are shown in [Table 13-6](#).

Table 13-6. Port C Pins Alternate Functions

Port Pin	Alternate Function
PC6	$\overline{\text{RESET}}$ (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

The alternate pin configuration is as follows:

- **RESET/PCINT14 – Port C, Bit 6**

RESET, Reset pin: When the RSTDISBL Fuse is programmed, this pin functions as a normal I/O pin, and the part will have to rely on Power-on Reset and Brown-out Reset as its reset sources. When the RSTDISBL Fuse is unprogrammed, the reset circuitry is connected to the pin, and the pin can not be used as an I/O pin.

If PC6 is used as a reset pin, DDC6, PORTC6 and PINC6 will all read 0.

PCINT14: Pin Change Interrupt source 14. The PC6 pin can serve as an external interrupt source.

- **SCL/ADC5/PCINT13 – Port C, Bit 5**

SCL, 2-wire Serial Interface Clock: When the TWEN bit in TWCR is set (one) to enable the 2-wire Serial Interface, pin PC5 is disconnected from the port and becomes the Serial Clock I/O pin for the 2-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation.

PC5 can also be used as ADC input Channel 5. Note that ADC input channel 5 uses digital power.

PCINT13: Pin Change Interrupt source 13. The PC5 pin can serve as an external interrupt source.

- **SDA/ADC4/PCINT12 – Port C, Bit 4**

SDA, 2-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the 2-wire Serial Interface, pin PC4 is disconnected from the port and becomes the Serial Data I/O pin for the 2-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation.

PC4 can also be used as ADC input Channel 4. Note that ADC input channel 4 uses digital power.

PCINT12: Pin Change Interrupt source 12. The PC4 pin can serve as an external interrupt source.

- **ADC3/PCINT11 – Port C, Bit 3**

PC3 can also be used as ADC input Channel 3. Note that ADC input channel 3 uses analog power.

PCINT11: Pin Change Interrupt source 11. The PC3 pin can serve as an external interrupt source.

- **ADC2/PCINT10 – Port C, Bit 2**

PC2 can also be used as ADC input Channel 2. Note that ADC input channel 2 uses analog power.

PCINT10: Pin Change Interrupt source 10. The PC2 pin can serve as an external interrupt source.

- **ADC1/PCINT9 – Port C, Bit 1**

PC1 can also be used as ADC input Channel 1. Note that ADC input channel 1 uses analog power.

PCINT9: Pin Change Interrupt source 9. The PC1 pin can serve as an external interrupt source.

- **ADC0/PCINT8 – Port C, Bit 0**

PC0 can also be used as ADC input Channel 0. Note that ADC input channel 0 uses analog power.

PCINT8: Pin Change Interrupt source 8. The PC0 pin can serve as an external interrupt source.

Table 13-7 and Table 13-8 relate the alternate functions of Port C to the overriding signals shown in Figure 13-5 on page 80.

Table 13-7. Overriding Signals for Alternate Functions in PC6..PC4⁽¹⁾

Signal Name	PC6/ $\overline{\text{RESET}}$ /PCINT14	PC5/SCL/ADC5/PCINT13	PC4/SDA/ADC4/PCINT12
PUOE	RSTDISBL	TWEN	TWEN
PUOV	1	PORTC5 • $\overline{\text{PUD}}$	PORTC4 • $\overline{\text{PUD}}$
DDOE	RSTDISBL	TWEN	TWEN
DDOV	0	SCL_OUT	SDA_OUT
PVOE	0	TWEN	TWEN
PVOV	0	0	0
DIEOE	RSTDISBL + PCINT14 • PCIE1	PCINT13 • PCIE1 + ADC5D	PCINT12 • PCIE1 + ADC4D
DIEOV	RSTDISBL	PCINT13 • PCIE1	PCINT12 • PCIE1
DI	PCINT14 INPUT	PCINT13 INPUT	PCINT12 INPUT
AIO	RESET INPUT	ADC5 INPUT / SCL INPUT	ADC4 INPUT / SDA INPUT

Note: 1. When enabled, the 2-wire Serial Interface enables slew-rate controls on the output pins PC4 and PC5. This is not shown in the figure. In addition, spike filters are connected between the AIO outputs shown in the port figure and the digital logic of the TWI module.

Table 13-8. Overriding Signals for Alternate Functions in PC3..PC0

Signal Name	PC3/ADC3/ PCINT11	PC2/ADC2/ PCINT10	PC1/ADC1/ PCINT9	PC0/ADC0/ PCINT8
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
DIEOE	PCINT11 • PCIE1 + ADC3D	PCINT10 • PCIE1 + ADC2D	PCINT9 • PCIE1 + ADC1D	PCINT8 • PCIE1 + ADC0D
DIEOV	PCINT11 • PCIE1	PCINT10 • PCIE1	PCINT9 • PCIE1	PCINT8 • PCIE1
DI	PCINT11 INPUT	PCINT10 INPUT	PCINT9 INPUT	PCINT8 INPUT
AIO	ADC3 INPUT	ADC2 INPUT	ADC1 INPUT	ADC0 INPUT

13.3.3 Alternate Functions of Port D

The Port D pins with alternate functions are shown in [Table 13-9](#).

Table 13-9. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

The alternate pin configuration is as follows:

- **AIN1/OC2B/PCINT23 – Port D, Bit 7**

AIN1, Analog Comparator Negative Input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.

PCINT23: Pin Change Interrupt source 23. The PD7 pin can serve as an external interrupt source.

- **AIN0/OC0A/PCINT22 – Port D, Bit 6**

AIN0, Analog Comparator Positive Input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.

OC0A, Output Compare Match output: The PD6 pin can serve as an external output for the Timer/Counter0 Compare Match A. The PD6 pin has to be configured as an output (DDD6 set (one)) to serve this function. The OC0A pin is also the output pin for the PWM mode timer function.

PCINT22: Pin Change Interrupt source 22. The PD6 pin can serve as an external interrupt source.

- **T1/OC0B/PCINT21 – Port D, Bit 5**

T1, Timer/Counter1 counter source.

OC0B, Output Compare Match output: The PD5 pin can serve as an external output for the Timer/Counter0 Compare Match B. The PD5 pin has to be configured as an output (DDD5 set (one)) to serve this function. The OC0B pin is also the output pin for the PWM mode timer function.

PCINT21: Pin Change Interrupt source 21. The PD5 pin can serve as an external interrupt source.

- **XCK/T0/PCINT20 – Port D, Bit 4**

XCK, USART external clock.

T0, Timer/Counter0 counter source.

PCINT20: Pin Change Interrupt source 20. The PD4 pin can serve as an external interrupt source.

- **INT1/OC2B/PCINT19 – Port D, Bit 3**

INT1, External Interrupt source 1: The PD3 pin can serve as an external interrupt source.

OC2B, Output Compare Match output: The PD3 pin can serve as an external output for the Timer/Counter0 Compare Match B. The PD3 pin has to be configured as an output (DDD3 set (one)) to serve this function. The OC2B pin is also the output pin for the PWM mode timer function.

PCINT19: Pin Change Interrupt source 19. The PD3 pin can serve as an external interrupt source.

- **INT0/PCINT18 – Port D, Bit 2**

INT0, External Interrupt source 0: The PD2 pin can serve as an external interrupt source.

PCINT18: Pin Change Interrupt source 18. The PD2 pin can serve as an external interrupt source.

- **TXD/PCINT17 – Port D, Bit 1**

TXD, Transmit Data (Data output pin for the USART). When the USART Transmitter is enabled, this pin is configured as an output regardless of the value of DDD1.

PCINT17: Pin Change Interrupt source 17. The PD1 pin can serve as an external interrupt source.

- **RXD/PCINT16 – Port D, Bit 0**

RXD, Receive Data (Data input pin for the USART). When the USART Receiver is enabled this pin is configured as an input regardless of the value of DDD0. When the USART forces this pin to be an input, the pull-up can still be controlled by the PORTD0 bit.

PCINT16: Pin Change Interrupt source 16. The PD0 pin can serve as an external interrupt source.

Table 13-10 and Table 13-11 relate the alternate functions of Port D to the overriding signals shown in Figure 13-5 on page 80.

Table 13-10. Overriding Signals for Alternate Functions PD7..PD4

Signal Name	PD7/AIN1 /PCINT23	PD6/AIN0/ OC0A/PCINT22	PD5/T1/OC0B/ PCINT21	PD4/XCK/ T0/PCINT20
PUOE	0	0	0	0
PUO	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	OC0A ENABLE	OC0B ENABLE	UMSEL
PVOV	0	OC0A	OC0B	XCK OUTPUT
DIEOE	PCINT23 • PCIE2	PCINT22 • PCIE2	PCINT21 • PCIE2	PCINT20 • PCIE2
DIEOV	1	1	1	1
DI	PCINT23 INPUT	PCINT22 INPUT	PCINT21 INPUT T1 INPUT	PCINT20 INPUT XCK INPUT T0 INPUT
AIO	AIN1 INPUT	AIN0 INPUT	–	–

Table 13-11. Overriding Signals for Alternate Functions in PD3..PD0

Signal Name	PD3/OC2B/INT1/ PCINT19	PD2/INT0/ PCINT18	PD1/TXD/ PCINT17	PD0/RXD/ PCINT16
PUOE	0	0	TXEN	RXEN
PUO	0	0	0	PORTD0 • PUD
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	OC2B ENABLE	0	TXEN	0
PVOV	OC2B	0	TXD	0
DIEOE	INT1 ENABLE + PCINT19 • PCIE2	INT0 ENABLE + PCINT18 • PCIE1	PCINT17 • PCIE2	PCINT16 • PCIE2
DIEOV	1	1	1	1
DI	PCINT19 INPUT INT1 INPUT	PCINT18 INPUT INT0 INPUT	PCINT17 INPUT	PCINT16 INPUT RXD
AIO	–	–	–	–

13.4 Register Description

13.4.1 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 4 – PUD: Pull-up Disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01). See “Configuring the Pin” on page 76 for more details about this feature.

13.4.2 PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.3 DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.4 PINB – The Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

13.4.5 PORTC – The Port C Data Register

Bit	7	6	5	4	3	2	1	0	
0x08 (0x28)	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.6 DDRC – The Port C Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x07 (0x27)	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.7 PINC – The Port C Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x06 (0x26)	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

13.4.8 PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.9 DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.10 PIND – The Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

14. 8-bit Timer/Counter0 with PWM

14.1 Features

- Two Independent Output Compare Units
- Double Buffered Output Compare Registers
- Clear Timer on Compare Match (Auto Reload)
- Glitch Free, Phase Correct Pulse Width Modulator (PWM)
- Variable PWM Period
- Frequency Generator
- Three Independent Interrupt Sources (TOV0, OCF0A, and OCF0B)

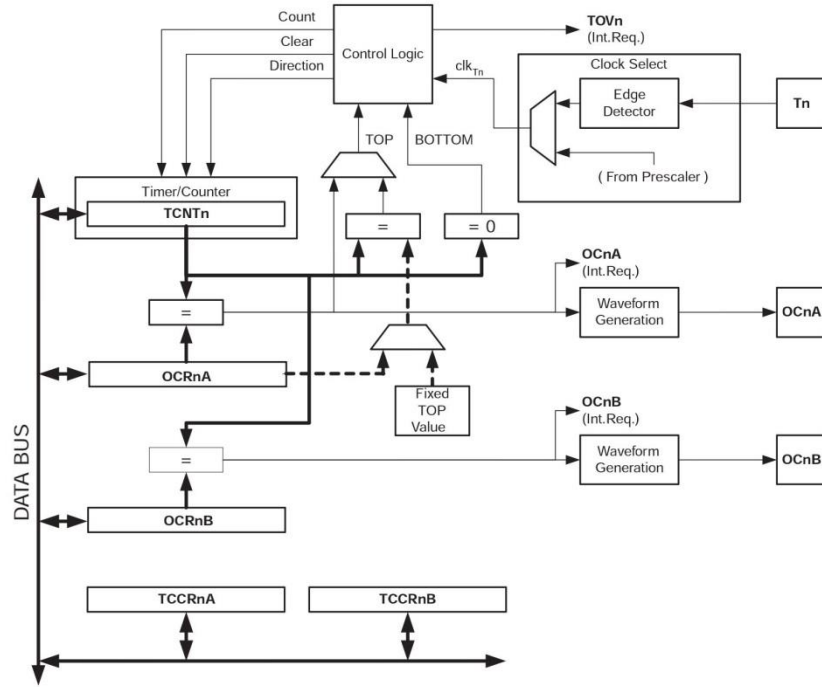
14.2 Overview

Timer/Counter0 is a general purpose 8-bit Timer/Counter module, with two independent Output Compare Units, and with PWM support. It allows accurate program execution timing (event management) and wave generation.

A simplified block diagram of the 8-bit Timer/Counter is shown in [Figure 14-1](#). For the actual placement of I/O pins, refer to "[Pinout ATmega48PA/88PA/168PA/328P](#)" on [page 2](#). CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the "[Register Description](#)" on [page 106](#).

The PRTIM0 bit in "[Minimizing Power Consumption](#)" on [page 42](#) must be written to zero to enable Timer/Counter0 module.

Figure 14-1. 8-bit Timer/Counter Block Diagram



14.2.1 Definitions

Many register and bit references in this section are written in general form. A lower case "n" replaces the Timer/Counter number, in this case 0. A lower case "x" replaces the Output Compare Unit, in this case Compare Unit A or Compare Unit B. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT0 for accessing Timer/Counter0 counter value and so on.

The definitions in [Table 14-1](#) are also used extensively throughout the document.

Table 14-1. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes 0x00.
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR0A Register. The assignment is dependent on the mode of operation.

14.2.2 Registers

The Timer/Counter (TCNT0) and Output Compare Registers (OCR0A and OCR0B) are 8-bit registers. Interrupt request (abbreviated to Int.Req. in the figure) signals are all visible in the Timer Interrupt Flag Register (TIFR0). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK0). TIFR0 and TIMSK0 are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T0 pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{T0}).

The double buffered Output Compare Registers (OCR0A and OCR0B) are compared with the Timer/Counter value at all times. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pins (OC0A and OC0B). See Section "15.7.3" on page 123. for details. The compare match event will also set the Compare Flag (OCF0A or OCF0B) which can be used to generate an Output Compare interrupt request.

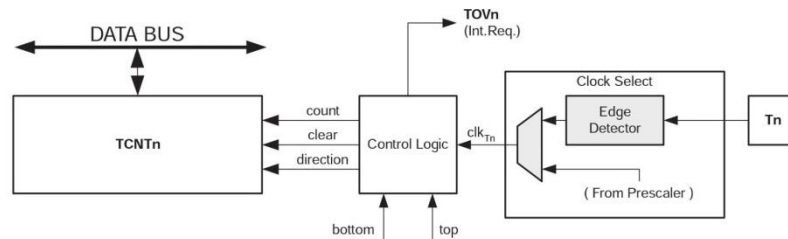
14.3 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the Clock Select logic which is controlled by the Clock Select (CS02:0) bits located in the Timer/Counter Control Register (TCCR0B). For details on clock sources and prescaler, see "Timer/Counter0 and Timer/Counter1 Prescalers" on page 141.

14.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 14-2 shows a block diagram of the counter and its surroundings.

Figure 14-2. Counter Unit Block Diagram



Signal description (internal signals):

count	Increment or decrement TCNT0 by 1.
direction	Select between increment and decrement.
clear	Clear TCNT0 (set all bits to zero).
clk_{Tn}	Timer/Counter clock, referred to as clk_{T0} in the following.
top	Signalize that TCNT0 has reached maximum value.
bottom	Signalize that TCNT0 has reached minimum value (zero).

Depending of the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T0}). clk_{T0} can be generated from an external or internal clock source, selected by the Clock Select bits (CS02:0). When no clock source is selected (CS02:0 = 0) the timer is stopped. However, the TCNT0 value can be accessed by the CPU, regardless of whether clk_{T0} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the Timer/Counter Control Register (TCCR0A) and the WGM02 bit located in the Timer/Counter Control Register B (TCCR0B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC0A and OC0B. For more details about advanced counting sequences and waveform generation, see ["Modes of Operation" on page 99](#).

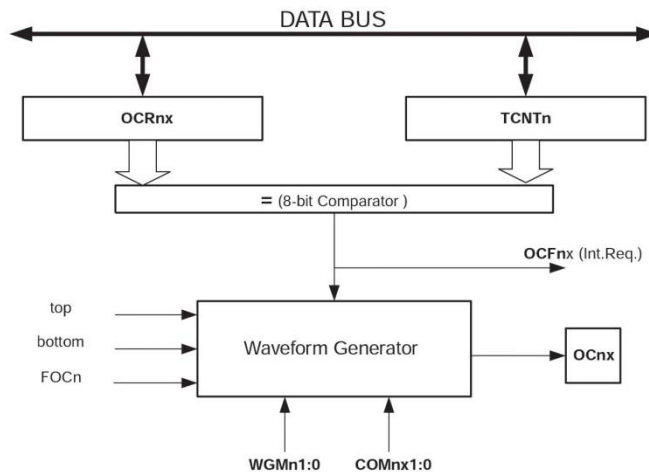
The Timer/Counter Overflow Flag (TOV0) is set according to the mode of operation selected by the WGM02:0 bits. TOV0 can be used for generating a CPU interrupt.

14.5 Output Compare Unit

The 8-bit comparator continuously compares TCNT0 with the Output Compare Registers (OCR0A and OCR0B). Whenever TCNT0 equals OCR0A or OCR0B, the comparator signals a match. A match will set the Output Compare Flag (OCF0A or OCF0B) at the next timer clock cycle. If the corresponding interrupt is enabled, the Output Compare Flag generates an Output Compare interrupt. The Output Compare Flag is automatically cleared when the interrupt is executed. Alternatively, the flag can be cleared by software by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to operating mode set by the WGM02:0 bits and Compare Output mode (COM0x1:0) bits. The max and bottom signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (["Modes of Operation" on page 99](#)).

Figure 14-3 shows a block diagram of the Output Compare unit.

Figure 14-3. Output Compare Unit, Block Diagram



The OCR0x Registers are double buffered when using any of the Pulse Width Modulation (PWM) modes. For the normal and Clear Timer on Compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR0x Compare Registers to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR0x Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR0x Buffer Register, and if double buffering is disabled the CPU will access the OCR0x directly.

14.5.1 Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the Force Output Compare (FOC0x) bit. Forcing compare match will not set the OCF0x Flag or reload/clear the timer, but the OC0x pin will be updated as if a real compare match had occurred (the COM0x1:0 bits settings define whether the OC0x pin is set, cleared or toggled).

14.5.2 Compare Match Blocking by TCNT0 Write

All CPU write operations to the TCNT0 Register will block any compare match that occur in the next timer clock cycle, even when the timer is stopped. This feature allows OCR0x to be initialized to the same value as TCNT0 without triggering an interrupt when the Timer/Counter clock is enabled.

14.5.3 Using the Output Compare Unit

Since writing TCNT0 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT0 when using the Output Compare Unit, independently of whether the Timer/Counter is running or not. If the value written to TCNT0 equals the OCR0x value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT0 value equal to BOTTOM when the counter is downcounting.

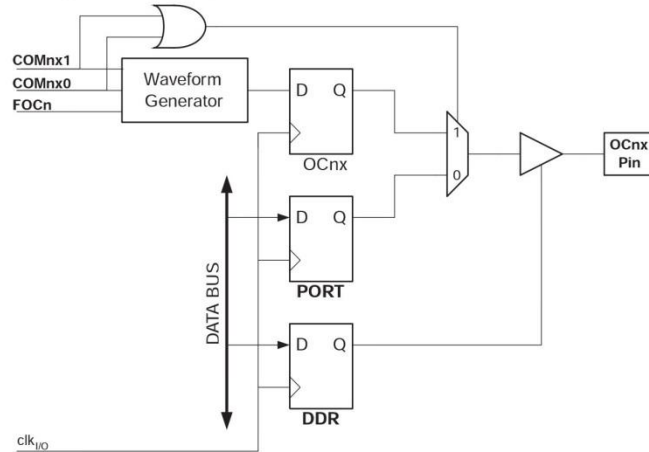
The setup of the OC0x should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC0x value is to use the Force Output Compare (FOC0x) strobe bits in Normal mode. The OC0x Registers keep their values even when changing between Waveform Generation modes.

Be aware that the COM0x1:0 bits are not double buffered together with the compare value. Changing the COM0x1:0 bits will take effect immediately.

14.6 Compare Match Output Unit

The Compare Output mode (COM0x1:0) bits have two functions. The Waveform Generator uses the COM0x1:0 bits for defining the Output Compare (OC0x) state at the next compare match. Also, the COM0x1:0 bits control the OC0x pin output source. [Figure 14-4](#) shows a simplified schematic of the logic affected by the COM0x1:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O port control registers (DDR and PORT) that are affected by the COM0x1:0 bits are shown. When referring to the OC0x state, the reference is for the internal OC0x Register, not the OC0x pin. If a system reset occur, the OC0x Register is reset to "0".

Figure 14-4. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC0x) from the Waveform Generator if either of the COM0x1:0 bits are set. However, the OC0x pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The Data Direction Register bit for the OC0x pin (DDR_OC0x) must be set as output before the OC0x value is visible on the pin. The port override function is independent of the Waveform Generation mode.

The design of the Output Compare pin logic allows initialization of the OC0x state before the output is enabled. Note that some COM0x1:0 bit settings are reserved for certain modes of operation. See Section "14.9" on page 106.

14.6.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM0x1:0 bits differently in Normal, CTC, and PWM modes. For all modes, setting the COM0x1:0 = 0 tells the Waveform Generator that no action on the OC0x Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 14-2 on page 106. For fast PWM mode, refer to Table 14-3 on page 106, and for phase correct PWM refer to Table 14-4 on page 107.

A change of the COM0x1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC0x strobe bits.

14.7 Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM02:0) and Compare Output mode (COM0x1:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM0x1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM0x1:0 bits control whether the output should be set, cleared, or toggled at a compare match (See Section "14.6" on page 98.).

For detailed timing information refer to "Timer/Counter Timing Diagrams" on page 104.

14.7.1 Normal Mode

The simplest mode of operation is the Normal mode ($WGM02:0 = 0$). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 8-bit value ($TOP = 0xFF$) and then restarts from the bottom ($0x00$). In normal operation the Timer/Counter Overflow Flag ($TOV0$) will be set in the same timer clock cycle as the $TCNT0$ becomes zero. The $TOV0$ Flag in this case behaves like a ninth bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the $TOV0$ Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

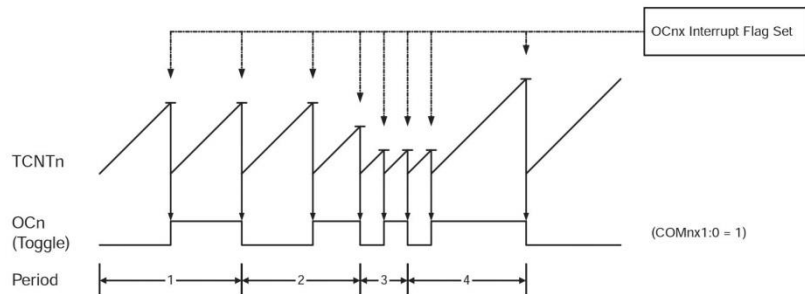
The Output Compare unit can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

14.7.2 Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare or CTC mode ($WGM02:0 = 2$), the $OCR0A$ Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value ($TCNT0$) matches the $OCR0A$. The $OCR0A$ defines the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in Figure 14-5. The counter value ($TCNT0$) increases until a compare match occurs between $TCNT0$ and $OCR0A$, and then counter ($TCNT0$) is cleared.

Figure 14-5. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the $OCF0A$ Flag. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing TOP to a value close to $BOTTOM$ when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to $OCR0A$ is lower than the current value of $TCNT0$, the counter will miss the compare match. The counter will then have to count to its maximum value ($0xFF$) and wrap around starting at $0x00$ before the compare match can occur.

For generating a waveform output in CTC mode, the $OC0A$ output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode ($COM0A1:0 = 1$). The $OC0A$ value will not be visible on the port pin unless the data direction for

the pin is set to output. The waveform generated will have a maximum frequency of $f_{OCO} = f_{clk_I/O}/2$ when OCR0A is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OCnx} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRnx)}$$

The N variable represents the prescale factor (1, 8, 64, 256, or 1024).

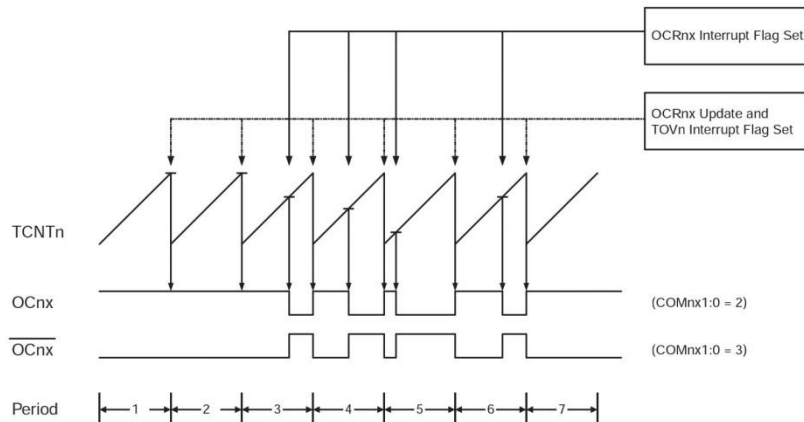
As for the Normal mode of operation, the TOV0 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

14.7.3 Fast PWM Mode

The fast Pulse Width Modulation or fast PWM mode (WGM02:0 = 3 or 7) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to TOP then restarts from BOTTOM. TOP is defined as 0xFF when WGM2:0 = 3, and OCR0A when WGM2:0 = 7. In non-inverting Compare Output mode, the Output Compare (OC0x) is cleared on the compare match between TCNT0 and OCR0x, and set at BOTTOM. In inverting Compare Output mode, the output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that use dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

In fast PWM mode, the counter is incremented until the counter value matches the TOP value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 14-6. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0x and TCNT0.

Figure 14-6. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches TOP. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC0x pins. Setting the COM0x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM0x1:0 to three: Setting the COM0A1:0 bits to one allows the OC0A pin to toggle on Compare Matches if the WGM02 bit is set. This option is not available for the OC0B pin (see [Table 14-6 on page 107](#)). The actual OC0x value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the OC0x Register at the compare match between OCR0x and TCNT0, and clearing (or setting) the OC0x Register at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{clk_I/O}}{N \cdot 256}$$

The N variable represents the prescale factor (1, 8, 64, 256, or 1024).

The extreme values for the OCR0A Register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR0A is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR0A equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM0A1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC0x to toggle its logical level on each compare match (COM0x1:0 = 1). The waveform generated will have a maximum frequency of $f_{OC0} = f_{clk_I/O}/2$ when OCR0A is set to zero. This feature is similar to the OC0A toggle in CTC mode, except the double buffer feature of the Output Compare unit is enabled in the fast PWM mode.

14.7.4 Phase Correct PWM Mode

The phase correct PWM mode (WGM02:0 = 1 or 5) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP and then from TOP to BOTTOM. TOP is defined as 0xFF when WGM2:0 = 1, and OCR0A when WGM2:0 = 5. In non-inverting Compare Output mode, the Output Compare (OC0x) is cleared on the compare match between TCNT0 and OCR0x while upcounting, and set on the compare match while downcounting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase correct PWM mode the counter is incremented until the counter value matches TOP. When the counter reaches TOP, it changes the count direction. The TCNT0 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on [Figure 14-7](#). The TCNT0 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0x and TCNT0.

symmetry around BOTTOM the OCnx value at MAX must correspond to the result of an up-counting Compare Match.

- The timer starts counting from a value higher than the one in OCRnx, and for that reason misses the Compare Match and hence the OCnx change that would have happened on the way up.

14.8 Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{T0}) is therefore shown as a clock enable signal in the following figures. The figures include information on when interrupt flags are set. Figure 14-8 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value in all modes other than phase correct PWM mode.

Figure 14-8. Timer/Counter Timing Diagram, no Prescaling

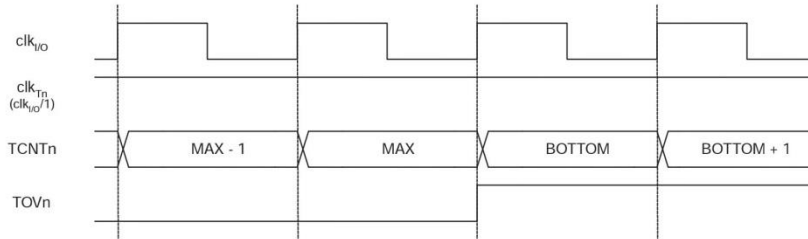


Figure 14-9 shows the same timing data, but with the prescaler enabled.

Figure 14-9. Timer/Counter Timing Diagram, with Prescaler ($f_{clk_{I/O}}/8$)

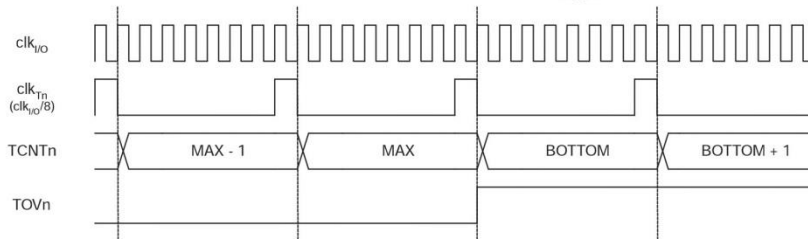


Figure 14-10 shows the setting of OCF0B in all modes and OCF0A in all modes except CTC mode and PWM mode, where OCR0A is TOP.

Figure 14-10. Timer/Counter Timing Diagram, Setting of OCF0x, with Prescaler ($f_{clk_{I/O}}/8$)

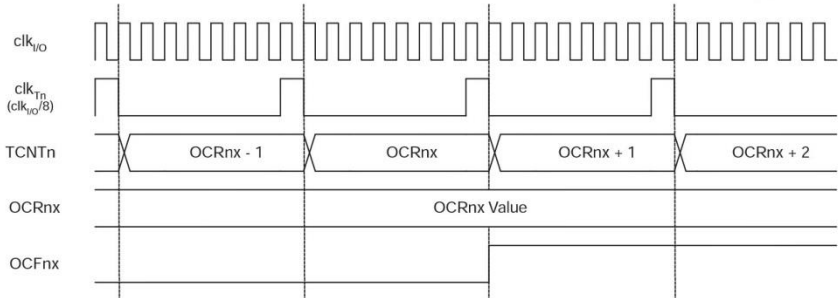
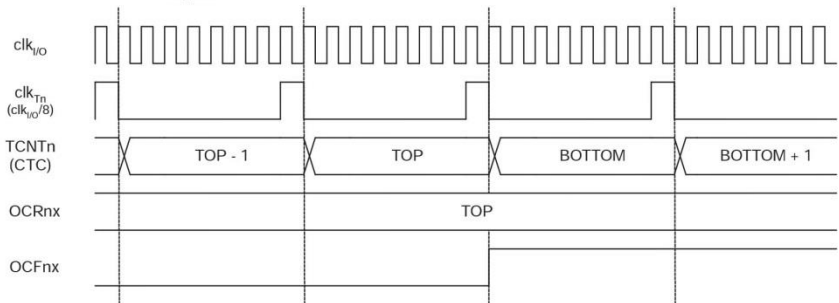


Figure 14-11 shows the setting of OCF0A and the clearing of TCNT0 in CTC mode and fast PWM mode where OCR0A is TOP.

Figure 14-11. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk_{I/O}}/8$)



14.9 Register Description

14.9.1 TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7:6 – COM0A1:0: Compare Match Output A Mode

These bits control the Output Compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0A pin must be set in order to enable the output driver.

When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. Table 14-2 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 14-2. Compare Output Mode, non-PWM Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

Table 14-3 shows the COM0A1:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 14-3. Compare Output Mode, Fast PWM Mode⁽¹⁾

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match, set OC0A at BOTTOM, (non-inverting mode).
1	1	Set OC0A on Compare Match, clear OC0A at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at BOTTOM. See “Fast PWM Mode” on page 101 for more details.

Table 14-4 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 14-4. Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match when up-counting. Set OC0A on Compare Match when down-counting.
1	1	Set OC0A on Compare Match when up-counting. Clear OC0A on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 128 for more details.

• **Bits 5:4 – COM0B1:0: Compare Match Output B Mode**

These bits control the Output Compare pin (OC0B) behavior. If one or both of the COM0B1:0 bits are set, the OC0B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0B pin must be set in order to enable the output driver.

When OC0B is connected to the pin, the function of the COM0B1:0 bits depends on the WGM02:0 bit setting. Table 14-5 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 14-5. Compare Output Mode, non-PWM Mode

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Toggle OC0B on Compare Match
1	0	Clear OC0B on Compare Match
1	1	Set OC0B on Compare Match

Table 14-6 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to fast PWM mode.

Table 14-6. Compare Output Mode, Fast PWM Mode⁽¹⁾

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match, set OC0B at BOTTOM, (non-inverting mode)
1	1	Set OC0B on Compare Match, clear OC0B at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Fast PWM Mode" on page 101 for more details.

Table 14-7 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 14-7. Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match when up-counting. Set OC0B on Compare Match when down-counting.
1	1	Set OC0B on Compare Match when up-counting. Clear OC0B on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 102 for more details.

• **Bits 3, 2 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

• **Bits 1:0 – WGM01:0: Waveform Generation Mode**

Combined with the WGM02 bit found in the TCCR0B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 14-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (see "Modes of Operation" on page 99).

Table 14-8. Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX = 0xFF
2. BOTTOM = 0x00

14.9.2 TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC0A: Force Output Compare A**

The FOC0A bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0A bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0A output is changed according to its COM0A1:0 bits setting. Note that the FOC0A bit is implemented as a strobe. Therefore it is the value present in the COM0A1:0 bits that determines the effect of the forced compare.

A FOC0A strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0A as TOP.

The FOC0A bit is always read as zero.

- **Bit 6 – FOC0B: Force Output Compare B**

The FOC0B bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0B bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0B output is changed according to its COM0B1:0 bits setting. Note that the FOC0B bit is implemented as a strobe. Therefore it is the value present in the COM0B1:0 bits that determines the effect of the forced compare.

A FOC0B strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0B as TOP.

The FOC0B bit is always read as zero.

- **Bits 5:4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 3 – WGM02: Waveform Generation Mode**

See the description in the "[TCCR0A – Timer/Counter Control Register A](#)" on page 106.

- **Bits 2:0 – CS02:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter.

Table 14-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk _{IO} /(No prescaling)
0	1	0	clk _{IO} /8 (From prescaler)
0	1	1	clk _{IO} /64 (From prescaler)
1	0	0	clk _{IO} /256 (From prescaler)
1	0	1	clk _{IO} /1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

14.9.3 TCNT0 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
0x26 (0x46)	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT0 Register blocks (removes) the Compare Match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a Compare Match between TCNT0 and the OCR0x Registers.

14.9.4 OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	OCR0A[7:0]								OCR0A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OCA pin.

14.9.5 OCR0B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
0x28 (0x48)	OCR0B[7:0]								OCR0B
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register B contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OCB pin.

14.9.6 TIMSK0 – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6E)	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 2 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable**

When the OCIE0B bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter Compare Match B interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter Interrupt Flag Register – TIFR0.

- **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Compare Match A interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

- **Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

14.9.7 TIFR0 – Timer/Counter 0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 2 – OCF0B: Timer/Counter 0 Output Compare B Match Flag**

The OCF0B bit is set when a Compare Match occurs between the Timer/Counter and the data in OCR0B – Output Compare Register0 B. OCF0B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0B (Timer/Counter Compare B Match Interrupt Enable), and OCF0B are set, the Timer/Counter Compare Match Interrupt is executed.

- **Bit 1 – OCF0A: Timer/Counter 0 Output Compare A Match Flag**

The OCF0A bit is set when a Compare Match occurs between the Timer/Counter0 and the data in OCR0A – Output Compare Register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to

the flag. When the I-bit in SREG, OCIE0A (Timer/Counter0 Compare Match Interrupt Enable), and OCF0A are set, the Timer/Counter0 Compare Match Interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set, the Timer/Counter0 Overflow interrupt is executed.

The setting of this flag is dependent of the WGM02:0 bit setting. Refer to [Table 14-8, "Waveform Generation Mode Bit Description"](#) on page 108.

15. 16-bit Timer/Counter1 with PWM

15.1 Features

- True 16-bit Design (i.e., Allows 16-bit PWM)
- Two independent Output Compare Units
- Double Buffered Output Compare Registers
- One Input Capture Unit
- Input Capture Noise Canceler
- Clear Timer on Compare Match (Auto Reload)
- Glitch-free, Phase Correct Pulse Width Modulator (PWM)
- Variable PWM Period
- Frequency Generator
- External Event Counter
- Four independent interrupt Sources (TOV1, OCF1A, OCF1B, and ICF1)

15.2 Overview

The 16-bit Timer/Counter unit allows accurate program execution timing (event management), wave generation, and signal timing measurement.

Most register and bit references in this section are written in general form. A lower case "n" replaces the Timer/Counter number, and a lower case "x" replaces the Output Compare unit channel. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT1 for accessing Timer/Counter1 counter value and so on.

A simplified block diagram of the 16-bit Timer/Counter is shown in [Figure 15-1](#). For the actual placement of I/O pins, refer to "[Pinout ATmega48PA/88PA/168PA/328P](#)" on [page 2](#). CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the "[Register Description](#)" on [page 134](#).

The PRTIM1 bit in "[PRR – Power Reduction Register](#)" on [page 45](#) must be written to zero to enable Timer/Counter1 module.

put Compare Units” on page 122. The compare match event will also set the Compare Match Flag (OCF1A/B) which can be used to generate an Output Compare interrupt request.

The Input Capture Register can capture the Timer/Counter value at a given external (edge triggered) event on either the Input Capture pin (ICP1) or on the Analog Comparator pins (See “Analog Comparator” on page 246) The Input Capture unit includes a digital filtering unit (Noise Canceler) for reducing the chance of capturing noise spikes.

The TOP value, or maximum Timer/Counter value, can in some modes of operation be defined by either the OCR1A Register, the ICR1 Register, or by a set of fixed values. When using OCR1A as TOP value in a PWM mode, the OCR1A Register can not be used for generating a PWM output. However, the TOP value will in this case be double buffered allowing the TOP value to be changed in run time. If a fixed TOP value is required, the ICR1 Register can be used as an alternative, freeing the OCR1A to be used as PWM output.

15.2.2 Definitions

The following definitions are used extensively throughout the section:

BOTTOM	The counter reaches the BOTTOM when it becomes 0x0000.
MAX	The counter reaches its MAXimum when it becomes 0xFFFF (decimal 65535).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be one of the fixed values: 0x00FF, 0x01FF, or 0x03FF, or to the value stored in the OCR1A or ICR1 Register. The assignment is dependent of the mode of operation.

15.3 Accessing 16-bit Registers

The TCNT1, OCR1A/B, and ICR1 are 16-bit registers that can be accessed by the AVR CPU via the 8-bit data bus. The 16-bit register must be byte accessed using two read or write operations. Each 16-bit timer has a single 8-bit register for temporary storing of the high byte of the 16-bit access. The same temporary register is shared between all 16-bit registers within each 16-bit timer. Accessing the low byte triggers the 16-bit read or write operation. When the low byte of a 16-bit register is written by the CPU, the high byte stored in the temporary register, and the low byte written are both copied into the 16-bit register in the same clock cycle. When the low byte of a 16-bit register is read by the CPU, the high byte of the 16-bit register is copied into the temporary register in the same clock cycle as the low byte is read.

Not all 16-bit accesses uses the temporary register for the high byte. Reading the OCR1A/B 16-bit registers does not involve using the temporary register.

To do a 16-bit write, the high byte must be written before the low byte. For a 16-bit read, the low byte must be read before the high byte.

The following code examples show how to access the 16-bit Timer Registers assuming that no interrupts updates the temporary register. The same principle can be used directly for accessing the OCR1A/B and ICR1 Registers. Note that when using “C”, the compiler handles the 16-bit access.

Assembly Code Examples⁽¹⁾

```

...
; Set TCNT1 to 0x01FF
ldi r17,0x01
ldi r16,0xFF
out TCNT1H,r17
out TCNT1L,r16
; Read TCNT1 into r17:r16
in r16,TCNT1L
in r17,TCNT1H
...

```

C Code Examples⁽¹⁾

```

unsigned int i;
...
/* Set TCNT1 to 0x01FF */
TCNT1 = 0x1FF;
/* Read TCNT1 into i */
i = TCNT1;
...

```

Note: 1. See "About Code Examples" on page 7.

For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBR", "SBRC", "SBR", and "CBR".

The assembly code example returns the TCNT1 value in the r17:r16 register pair.

It is important to notice that accessing 16-bit registers are atomic operations. If an interrupt occurs between the two instructions accessing the 16-bit register, and the interrupt code updates the temporary register by accessing the same or any other of the 16-bit Timer Registers, then the result of the access outside the interrupt will be corrupted. Therefore, when both the main code and the interrupt code update the temporary register, the main code must disable the interrupts during the 16-bit access.

The following code examples show how to do an atomic read of the TCNT1 Register contents. Reading any of the OCR1A/B or ICR1 Registers can be done by using the same principle.

Assembly Code Example⁽¹⁾

```

TIM16_ReadTCNT1:
    ; Save global interrupt flag
    in r18,SREG
    ; Disable interrupts
    cli
    ; Read TCNT1 into r17:r16
    in r16,TCNT1L
    in r17,TCNT1H
    ; Restore global interrupt flag
    out SREG,r18
    ret
    
```

C Code Example⁽¹⁾

```

unsigned int TIM16_ReadTCNT1( void )
{
    unsigned char sreg;
    unsigned int i;
    /* Save global interrupt flag */
    sreg = SREG;
    /* Disable interrupts */
    _CLI();
    /* Read TCNT1 into i */
    i = TCNT1;
    /* Restore global interrupt flag */
    SREG = sreg;
    return i;
}
    
```

Note: 1. See "About Code Examples" on page 7.

For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

The assembly code example returns the TCNT1 value in the r17:r16 register pair.

The following code examples show how to do an atomic write of the TCNT1 Register contents. Writing any of the OCR1A/B or ICR1 Registers can be done by using the same principle.

Assembly Code Example⁽¹⁾

```
TIM16_WriteTCNT1:
    ; Save global interrupt flag
    in r18,SREG
    ; Disable interrupts
    cli
    ; Set TCNT1 to r17:r16
    out TCNT1H,r17
    out TCNT1L,r16
    ; Restore global interrupt flag
    out SREG,r18
    ret
```

C Code Example⁽¹⁾

```
void TIM16_WriteTCNT1( unsigned int i )
{
    unsigned char sreg;
    unsigned int i;
    /* Save global interrupt flag */
    sreg = SREG;
    /* Disable interrupts */
    _CLI();
    /* Set TCNT1 to i */
    TCNT1 = i;
    /* Restore global interrupt flag */
    SREG = sreg;
}
```

Note: 1. See "About Code Examples" on page 7.
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

The assembly code example requires that the r17:r16 register pair contains the value to be written to TCNT1.

15.3.1 Reusing the Temporary High Byte Register

If writing to more than one 16-bit register where the high byte is the same for all registers written, then the high byte only needs to be written once. However, note that the same rule of atomic operation described previously also applies in this case.

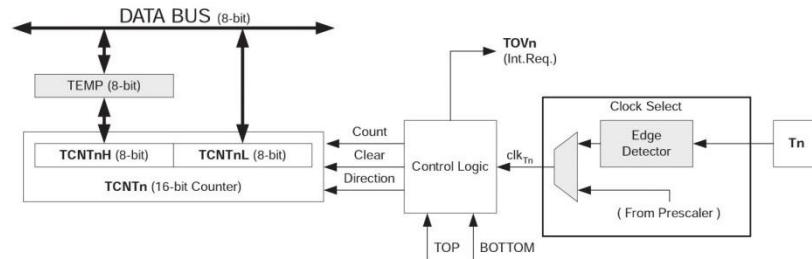
15.4 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the Clock Select logic which is controlled by the *Clock Select* (CS12:0) bits located in the *Timer/Counter control Register B* (TCCR1B). For details on clock sources and prescaler, see "Timer/Counter0 and Timer/Counter1 Prescalers" on page 141.

15.5 Counter Unit

The main part of the 16-bit Timer/Counter is the programmable 16-bit bi-directional counter unit. Figure 15-2 shows a block diagram of the counter and its surroundings.

Figure 15-2. Counter Unit Block Diagram



Signal description (internal signals):

Count	Increment or decrement TCNT1 by 1.
Direction	Select between increment and decrement.
Clear	Clear TCNT1 (set all bits to zero).
clk_{T1}	Timer/Counter clock.
TOP	Signalize that TCNT1 has reached maximum value.
BOTTOM	Signalize that TCNT1 has reached minimum value (zero).

The 16-bit counter is mapped into two 8-bit I/O memory locations: *Counter High* (TCNT1H) containing the upper eight bits of the counter, and *Counter Low* (TCNT1L) containing the lower eight bits. The TCNT1H Register can only be indirectly accessed by the CPU. When the CPU does an access to the TCNT1H I/O location, the CPU accesses the high byte temporary register (TEMP). The temporary register is updated with the TCNT1H value when the TCNT1L is read, and TCNT1H is updated with the temporary register value when TCNT1L is written. This allows the CPU to read or write the entire 16-bit counter value within one clock cycle via the 8-bit data bus. It is important to notice that there are special cases of writing to the TCNT1 Register when the counter is counting that will give unpredictable results. The special cases are described in the sections where they are of importance.

Depending on the mode of operation used, the counter is cleared, incremented, or decremented at each *timer clock* (clk_{T1}). The clk_{T1} can be generated from an external or internal clock source, selected by the *Clock Select* bits (CS12:0). When no clock source is selected (CS12:0 = 0) the timer is stopped. However, the TCNT1 value can be accessed by the CPU, independent of whether clk_{T1} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the *Waveform Generation mode* bits (WGM13:0) located in the *Timer/Counter Control Registers A and B* (TCCR1A and TCCR1B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC1x. For more details about advanced counting sequences and waveform generation, see ["Modes of Operation" on page 125](#).

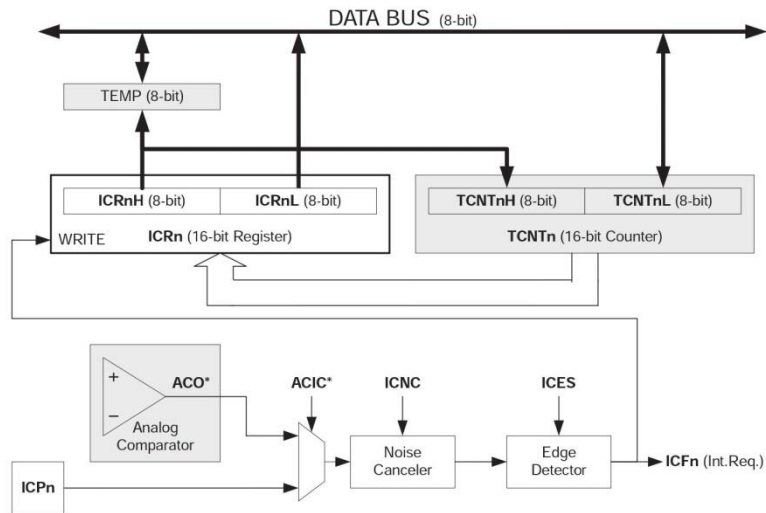
The Timer/Counter Overflow Flag (TOV1) is set according to the mode of operation selected by the WGM13:0 bits. TOV1 can be used for generating a CPU interrupt.

15.6 Input Capture Unit

The Timer/Counter incorporates an Input Capture unit that can capture external events and give them a time-stamp indicating time of occurrence. The external signal indicating an event, or multiple events, can be applied via the ICP1 pin or alternatively, via the analog-comparator unit. The time-stamps can then be used to calculate frequency, duty-cycle, and other features of the signal applied. Alternatively the time-stamps can be used for creating a log of the events.

The Input Capture unit is illustrated by the block diagram shown in Figure 15-3. The elements of the block diagram that are not directly a part of the Input Capture unit are gray shaded. The small "n" in register and bit names indicates the Timer/Counter number.

Figure 15-3. Input Capture Unit Block Diagram



When a change of the logic level (an event) occurs on the *Input Capture pin* (ICP1), alternatively on the *Analog Comparator output* (ACO), and this change confirms to the setting of the edge detector, a capture will be triggered. When a capture is triggered, the 16-bit value of the counter (TCNT1) is written to the *Input Capture Register* (ICR1). The *Input Capture Flag* (ICF1) is set at the same system clock as the TCNT1 value is copied into ICR1 Register. If enabled (ICIE1 = 1), the Input Capture Flag generates an Input Capture interrupt. The ICF1 Flag is automatically cleared when the interrupt is executed. Alternatively the ICF1 Flag can be cleared by software by writing a logical one to its I/O bit location.

Reading the 16-bit value in the *Input Capture Register* (ICR1) is done by first reading the low byte (ICR1L) and then the high byte (ICR1H). When the low byte is read the high byte is copied into the high byte temporary register (TEMP). When the CPU reads the ICR1H I/O location it will access the TEMP Register.

The ICR1 Register can only be written when using a Waveform Generation mode that utilizes the ICR1 Register for defining the counter's TOP value. In these cases the *Waveform Genera-*

tion mode (WGM13:0) bits must be set before the TOP value can be written to the ICR1 Register. When writing the ICR1 Register the high byte must be written to the ICR1H I/O location before the low byte is written to ICR1L.

For more information on how to access the 16-bit registers refer to ["Accessing 16-bit Registers" on page 115](#).

15.6.1 Input Capture Trigger Source

The main trigger source for the Input Capture unit is the *Input Capture pin* (ICP1). Timer/Counter1 can alternatively use the Analog Comparator output as trigger source for the Input Capture unit. The Analog Comparator is selected as trigger source by setting the *Analog Comparator Input Capture* (ACIC) bit in the *Analog Comparator Control and Status Register* (ACSR). Be aware that changing trigger source can trigger a capture. The Input Capture Flag must therefore be cleared after the change.

Both the *Input Capture pin* (ICP1) and the *Analog Comparator output* (ACO) inputs are sampled using the same technique as for the T1 pin ([Figure 16-1 on page 141](#)). The edge detector is also identical. However, when the noise canceler is enabled, additional logic is inserted before the edge detector, which increases the delay by four system clock cycles. Note that the input of the noise canceler and edge detector is always enabled unless the Timer/Counter is set in a Waveform Generation mode that uses ICR1 to define TOP.

An Input Capture can be triggered by software by controlling the port of the ICP1 pin.

15.6.2 Noise Canceler

The noise canceler improves noise immunity by using a simple digital filtering scheme. The noise canceler input is monitored over four samples, and all four must be equal for changing the output that in turn is used by the edge detector.

The noise canceler is enabled by setting the *Input Capture Noise Canceler* (ICNC1) bit in *Timer/Counter Control Register B* (TCCR1B). When enabled the noise canceler introduces additional four system clock cycles of delay from a change applied to the input, to the update of the ICR1 Register. The noise canceler uses the system clock and is therefore not affected by the prescaler.

15.6.3 Using the Input Capture Unit

The main challenge when using the Input Capture unit is to assign enough processor capacity for handling the incoming events. The time between two events is critical. If the processor has not read the captured value in the ICR1 Register before the next event occurs, the ICR1 will be overwritten with a new value. In this case the result of the capture will be incorrect.

When using the Input Capture interrupt, the ICR1 Register should be read as early in the interrupt handler routine as possible. Even though the Input Capture interrupt has relatively high priority, the maximum interrupt response time is dependent on the maximum number of clock cycles it takes to handle any of the other interrupt requests.

Using the Input Capture unit in any mode of operation when the TOP value (resolution) is actively changed during operation, is not recommended.

Measurement of an external signal's duty cycle requires that the trigger edge is changed after each capture. Changing the edge sensing must be done as early as possible after the ICR1 Register has been read. After a change of the edge, the Input Capture Flag (ICF1) must be

cleared by software (writing a logical one to the I/O bit location). For measuring frequency only, the clearing of the ICF1 flag is not required (if an interrupt handler is used).

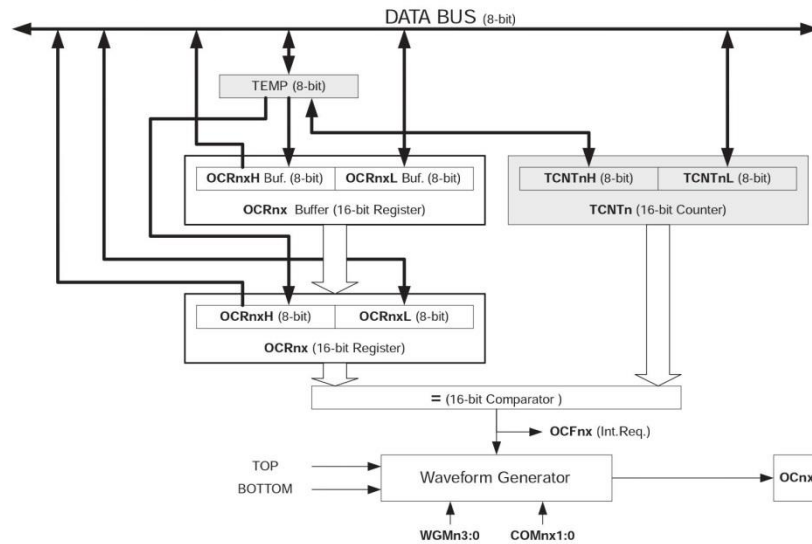
15.7 Output Compare Units

The 16-bit comparator continuously compares TCNT1 with the *Output Compare Register* (OCR1x). If TCNT equals OCR1x the comparator signals a match. A match will set the *Output Compare Flag* (OCF1x) at the next timer clock cycle. If enabled (OCIE1x = 1), the Output Compare Flag generates an Output Compare interrupt. The OCF1x Flag is automatically cleared when the interrupt is executed. Alternatively the OCF1x Flag can be cleared by software by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to operating mode set by the *Waveform Generation mode* (WGM13:0) bits and *Compare Output mode* (COM1x1:0) bits. The TOP and BOTTOM signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (See Section "15.9" on page 125.)

A special feature of Output Compare unit A allows it to define the Timer/Counter TOP value (i.e., counter resolution). In addition to the counter resolution, the TOP value defines the period time for waveforms generated by the Waveform Generator.

Figure 15-4 shows a block diagram of the Output Compare unit. The small "n" in the register and bit names indicates the device number (n = 1 for Timer/Counter 1), and the "x" indicates Output Compare unit (A/B). The elements of the block diagram that are not directly a part of the Output Compare unit are gray shaded.

Figure 15-4. Output Compare Unit, Block Diagram



The OCR1x Register is double buffered when using any of the twelve *Pulse Width Modulation* (PWM) modes. For the Normal and *Clear Timer on Compare* (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR1x Compare Register to either TOP or BOTTOM of the counting sequence. The synchronization

prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR1x Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR1x Buffer Register, and if double buffering is disabled the CPU will access the OCR1x directly. The content of the OCR1x (Buffer or Compare) Register is only changed by a write operation (the Timer/Counter does not update this register automatically as the TCNT1 and ICR1 Register). Therefore OCR1x is not read via the high byte temporary register (TEMP). However, it is a good practice to read the low byte first as when accessing other 16-bit registers. Writing the OCR1x Registers must be done via the TEMP Register since the compare of all 16 bits is done continuously. The high byte (OCR1xH) has to be written first. When the high byte I/O location is written by the CPU, the TEMP Register will be updated by the value written. Then when the low byte (OCR1xL) is written to the lower eight bits, the high byte will be copied into the upper 8-bits of either the OCR1x buffer or OCR1x Compare Register in the same system clock cycle.

For more information of how to access the 16-bit registers refer to "[Accessing 16-bit Registers](#)" on page 115.

15.7.1 Force Output Compare

In non-PWM Waveform Generation modes, the match output of the comparator can be forced by writing a one to the *Force Output Compare* (FOC1x) bit. Forcing compare match will not set the OCF1x Flag or reload/clear the timer, but the OC1x pin will be updated as if a real compare match had occurred (the COM11:0 bits settings define whether the OC1x pin is set, cleared or toggled).

15.7.2 Compare Match Blocking by TCNT1 Write

All CPU writes to the TCNT1 Register will block any compare match that occurs in the next timer clock cycle, even when the timer is stopped. This feature allows OCR1x to be initialized to the same value as TCNT1 without triggering an interrupt when the Timer/Counter clock is enabled.

15.7.3 Using the Output Compare Unit

Since writing TCNT1 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT1 when using any of the Output Compare channels, independent of whether the Timer/Counter is running or not. If the value written to TCNT1 equals the OCR1x value, the compare match will be missed, resulting in incorrect waveform generation. Do not write the TCNT1 equal to TOP in PWM modes with variable TOP values. The compare match for the TOP will be ignored and the counter will continue to 0xFFFF. Similarly, do not write the TCNT1 value equal to BOTTOM when the counter is downcounting.

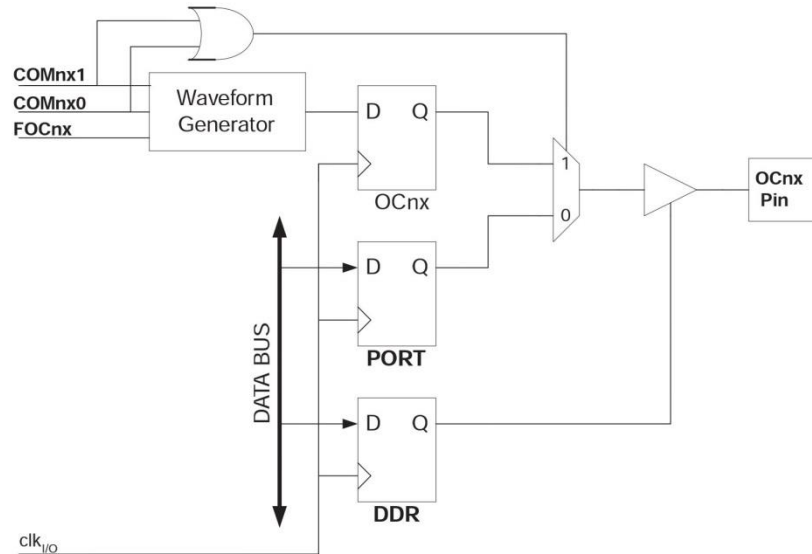
The setup of the OC1x should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC1x value is to use the Force Output Compare (FOC1x) strobe bits in Normal mode. The OC1x Register keeps its value even when changing between Waveform Generation modes.

Be aware that the COM1x1:0 bits are not double buffered together with the compare value. Changing the COM1x1:0 bits will take effect immediately.

15.8 Compare Match Output Unit

The *Compare Output mode* (COM1x1:0) bits have two functions. The Waveform Generator uses the COM1x1:0 bits for defining the Output Compare (OC1x) state at the next compare match. Secondly the COM1x1:0 bits control the OC1x pin output source. Figure 15-5 shows a simplified schematic of the logic affected by the COM1x1:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O Port Control Registers (DDR and PORT) that are affected by the COM1x1:0 bits are shown. When referring to the OC1x state, the reference is for the internal OC1x Register, not the OC1x pin. If a system reset occur, the OC1x Register is reset to "0".

Figure 15-5. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC1x) from the Waveform Generator if either of the COM1x1:0 bits are set. However, the OC1x pin direction (input or output) is still controlled by the *Data Direction Register* (DDR) for the port pin. The Data Direction Register bit for the OC1x pin (DDR_OC1x) must be set as output before the OC1x value is visible on the pin. The port override function is generally independent of the Waveform Generation mode, but there are some exceptions. Refer to Table 15-1, Table 15-2 and Table 15-3 for details.

The design of the Output Compare pin logic allows initialization of the OC1x state before the output is enabled. Note that some COM1x1:0 bit settings are reserved for certain modes of operation. See Section "15.11" on page 134.

The COM1x1:0 bits have no effect on the Input Capture unit.

15.8.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM1x1:0 bits differently in normal, CTC, and PWM modes. For all modes, setting the COM1x1:0 = 0 tells the Waveform Generator that no action on the OC1x Register is to be performed on the next compare match. For compare output actions in the

non-PWM modes refer to [Table 15-1 on page 134](#). For fast PWM mode refer to [Table 15-2 on page 135](#), and for phase correct and phase and frequency correct PWM refer to [Table 15-3 on page 135](#).

A change of the COM1x1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC1x strobe bits.

15.9 Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the *Waveform Generation mode* (WGM13:0) and *Compare Output mode* (COM1x1:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM1x1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM1x1:0 bits control whether the output should be set, cleared or toggle at a compare match (See [Section "15.8" on page 124](#).)

For detailed timing information refer to ["Timer/Counter Timing Diagrams" on page 132](#).

15.9.1 Normal Mode

The simplest mode of operation is the *Normal mode* (WGM13:0 = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 16-bit value (MAX = 0xFFFF) and then restarts from the BOTTOM (0x0000). In normal operation the *Timer/Counter Overflow Flag* (TOV1) will be set in the same timer clock cycle as the TCNT1 becomes zero. The TOV1 Flag in this case behaves like a 17th bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV1 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

The Input Capture unit is easy to use in Normal mode. However, observe that the maximum interval between the external events must not exceed the resolution of the counter. If the interval between events are too long, the timer overflow interrupt or the prescaler must be used to extend the resolution for the capture unit.

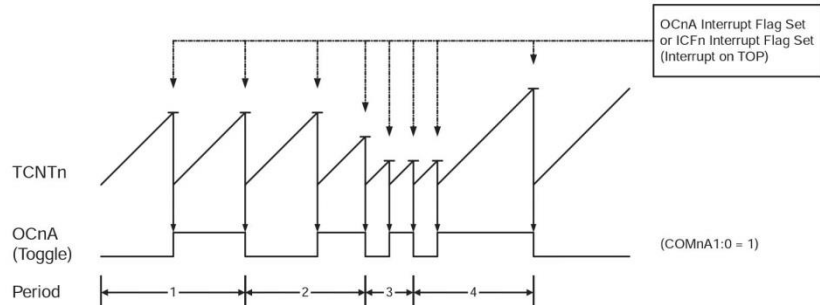
The Output Compare units can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

15.9.2 Clear Timer on Compare Match (CTC) Mode

In *Clear Timer on Compare* or CTC mode (WGM13:0 = 4 or 12), the OCR1A or ICR1 Register are used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT1) matches either the OCR1A (WGM13:0 = 4) or the ICR1 (WGM13:0 = 12). The OCR1A or ICR1 define the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in [Figure 15-6](#). The counter value (TCNT1) increases until a compare match occurs with either OCR1A or ICR1, and then counter (TCNT1) is cleared.

Figure 15-6. CTC Mode, Timing Diagram



An interrupt can be generated at each time the counter value reaches the TOP value by either using the OCF1A or ICF1 Flag according to the register used to define the TOP value. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing the TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR1A or ICR1 is lower than the current value of TCNT1, the counter will miss the compare match. The counter will then have to count to its maximum value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. In many cases this feature is not desirable. An alternative will then be to use the fast PWM mode using OCR1A for defining TOP (WGM13:0 = 15) since the OCR1A then will be double buffered.

For generating a waveform output in CTC mode, the OC1A output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode (COM1A1:0 = 1). The OC1A value will not be visible on the port pin unless the data direction for the pin is set to output (DDR_OC1A = 1). The waveform generated will have a maximum frequency of $f_{OC1A} = f_{clk_I/O} / 2$ when OCR1A is set to zero (0x0000). The waveform frequency is defined by the following equation:

$$f_{OCnA} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRnA)}$$

The N variable represents the prescaler factor (1, 8, 64, 256, or 1024).

As for the Normal mode of operation, the TOV1 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x0000.

15.9.3 Fast PWM Mode

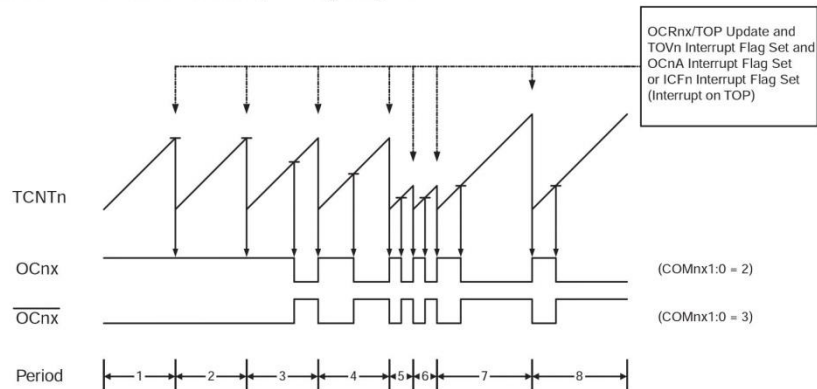
The *fast Pulse Width Modulation* or fast PWM mode (WGM13:0 = 5, 6, 7, 14, or 15) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM options by its single-slope operation. The counter counts from BOTTOM to TOP then restarts from BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the compare match between TCNT1 and OCR1x, and set at BOTTOM. In inverting Compare Output mode output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct and phase and frequency correct PWM modes that use dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), hence reduces total system cost.

The PWM resolution for fast PWM can be fixed to 8-, 9-, or 10-bit, or defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX). The PWM resolution in bits can be calculated by using the following equation:

$$R_{FPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

In fast PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF, or 0x03FF (WGM13:0 = 5, 6, or 7), the value in ICR1 (WGM13:0 = 14), or the value in OCR1A (WGM13:0 = 15). The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 15-7. The figure shows fast PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.

Figure 15-7. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV1) is set each time the counter reaches TOP. In addition the OC1A or ICF1 Flag is set at the same timer clock cycle as TOV1 is set when either OCR1A or ICR1 is used for defining the TOP value. If one of the interrupts are enabled, the interrupt handler routine can be used for updating the TOP and compare values.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the TCNT1 and the OCR1x. Note that when using fixed TOP values the unused bits are masked to zero when any of the OCR1x Registers are written.

The procedure for updating ICR1 differs from updating OCR1A when used for defining the TOP value. The ICR1 Register is not double buffered. This means that if ICR1 is changed to a low value when the counter is running with none or a low prescaler value, there is a risk that the new ICR1 value written is lower than the current value of TCNT1. The result will then be that the counter will miss the compare match at the TOP value. The counter will then have to count to the MAX value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. The OCR1A Register however, is double buffered. This feature allows the OCR1A I/O location

to be written anytime. When the OCR1A I/O location is written the value written will be put into the OCR1A Buffer Register. The OCR1A Compare Register will then be updated with the value in the Buffer Register at the next timer clock cycle the TCNT1 matches TOP. The update is done at the same timer clock cycle as the TCNT1 is cleared and the TOV1 Flag is set.

Using the ICR1 Register for defining TOP works well when using fixed TOP values. By using ICR1, the OCR1A Register is free to be used for generating a PWM output on OC1A. However, if the base PWM frequency is actively changed (by changing the TOP value), using the OCR1A as TOP is clearly a better choice due to its double buffer feature.

In fast PWM mode, the compare units allow generation of PWM waveforms on the OC1x pins. Setting the COM1x1:0 bits to two will produce an inverted PWM and a non-inverted PWM output can be generated by setting the COM1x1:0 bits to three (see [Table on page 135](#)). The actual OC1x value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1x). The PWM waveform is generated by setting (or clearing) the OC1x Register at the compare match between OCR1x and TCNT1, and clearing (or setting) the OC1x Register at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{clk_I/O}}{N \cdot (1 + TOP)}$$

The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1x Register represent special cases when generating a PWM waveform output in the fast PWM mode. If the OCR1x is set equal to BOTTOM (0x0000) the output will be a narrow spike for each TOP+1 timer clock cycle. Setting the OCR1x equal to TOP will result in a constant high or low output (depending on the polarity of the output set by the COM1x1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC1A to toggle its logical level on each compare match (COM1A1:0 = 1). This applies only if OCR1A is used to define the TOP value (WGM13:0 = 15). The waveform generated will have a maximum frequency of $f_{OC1A} = f_{clk_I/O}/2$ when OCR1A is set to zero (0x0000). This feature is similar to the OC1A toggle in CTC mode, except the double buffer feature of the Output Compare unit is enabled in the fast PWM mode.

15.9.4 Phase Correct PWM Mode

The *phase correct Pulse Width Modulation* or phase correct PWM mode (WGM13:0 = 1, 2, 3, 10, or 11) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is, like the phase and frequency correct PWM mode, based on a dual-slope operation. The counter counts repeatedly from BOTTOM (0x0000) to TOP and then from TOP to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the compare match between TCNT1 and OCR1x while upcounting, and set on the compare match while downcounting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has a lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

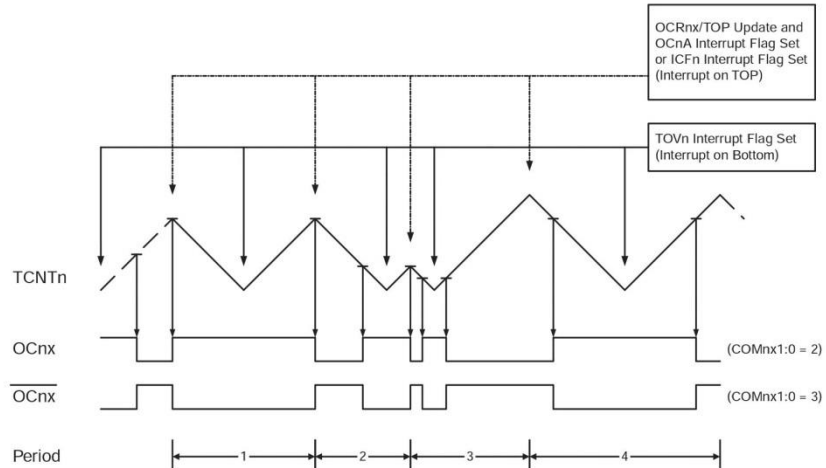
The PWM resolution for the phase correct PWM mode can be fixed to 8-, 9-, or 10-bit, or defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to

0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX). The PWM resolution in bits can be calculated by using the following equation:

$$R_{PCPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

In phase correct PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF, or 0x03FF (WGM13:0 = 1, 2, or 3), the value in ICR1 (WGM13:0 = 10), or the value in OCR1A (WGM13:0 = 11). The counter has then reached the TOP and changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 15-8. The figure shows phase correct PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 value is in the timing diagram shown as a histogram illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.

Figure 15-8. Phase Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV1) is set each time the counter reaches BOTTOM. When either OCR1A or ICR1 is used for defining the TOP value, the OC1A or ICF1 Flag is set accordingly at the same timer clock cycle as the OCR1x Registers are updated with the double buffer value (at TOP). The Interrupt Flags can be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the TCNT1 and the OCR1x. Note that when using fixed TOP values, the unused bits are masked to zero when any of the OCR1x Registers are written. As the third period shown in Figure 15-8 illustrates, changing the TOP actively while the Timer/Counter is running in the phase correct mode can result in an unsymmetrical output. The reason for this can be found in the time of update of the OCR1x Register. Since the OCR1x update occurs at TOP, the PWM period starts and ends at TOP. This

implies that the length of the falling slope is determined by the previous TOP value, while the length of the rising slope is determined by the new TOP value. When these two values differ the two slopes of the period will differ in length. The difference in length gives the unsymmetrical result on the output.

It is recommended to use the phase and frequency correct mode instead of the phase correct mode when changing the TOP value while the Timer/Counter is running. When using a static TOP value there are practically no differences between the two modes of operation.

In phase correct PWM mode, the compare units allow generation of PWM waveforms on the OC1x pins. Setting the COM1x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1x1:0 to three (See [Table on page 135](#)). The actual OC1x value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1x). The PWM waveform is generated by setting (or clearing) the OC1x Register at the compare match between OCR1x and TCNT1 when the counter increments, and clearing (or setting) the OC1x Register at compare match between OCR1x and TCNT1 when the counter decrements. The PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnxPCPWM} = \frac{f_{clk_I/O}}{2 \cdot N \cdot TOP}$$

The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1x Register represent special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR1x is set equal to BOTTOM the output will be continuously low and if set equal to TOP the output will be continuously high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values. If OCR1A is used to define the TOP value (WGM13:0 = 11) and COM1A1:0 = 1, the OC1A output will toggle with a 50% duty cycle.

15.9.5 Phase and Frequency Correct PWM Mode

The *phase and frequency correct Pulse Width Modulation*, or phase and frequency correct PWM mode (WGM13:0 = 8 or 9) provides a high resolution phase and frequency correct PWM waveform generation option. The phase and frequency correct PWM mode is, like the phase correct PWM mode, based on a dual-slope operation. The counter counts repeatedly from BOTTOM (0x0000) to TOP and then from TOP to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the compare match between TCNT1 and OCR1x while upcounting, and set on the compare match while downcounting. In inverting Compare Output mode, the operation is inverted. The dual-slope operation gives a lower maximum operation frequency compared to the single-slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

The main difference between the phase correct, and the phase and frequency correct PWM mode is the time the OCR1x Register is updated by the OCR1x Buffer Register, (see [Figure 15-8](#) and [Figure 15-9](#)).

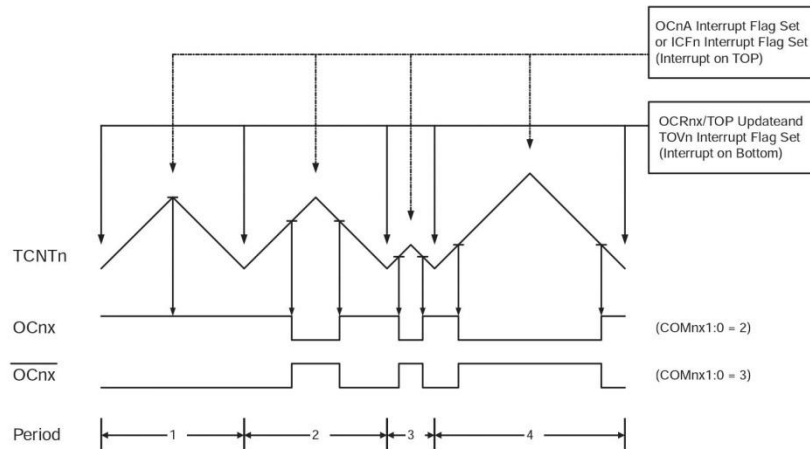
The PWM resolution for the phase and frequency correct PWM mode can be defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and

the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX). The PWM resolution in bits can be calculated using the following equation:

$$R_{PFCPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

In phase and frequency correct PWM mode the counter is incremented until the counter value matches either the value in ICR1 (WGM13:0 = 8), or the value in OCR1A (WGM13:0 = 9). The counter has then reached the TOP and changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct and frequency correct PWM mode is shown on Figure 15-9. The figure shows phase and frequency correct PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.

Figure 15-9. Phase and Frequency Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV1) is set at the same timer clock cycle as the OCR1x Registers are updated with the double buffer value (at BOTTOM). When either OCR1A or ICR1 is used for defining the TOP value, the OC1A or ICF1 Flag set when TCNT1 has reached TOP. The Interrupt Flags can then be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the TCNT1 and the OCR1x.

As Figure 15-9 shows the output generated is, in contrast to the phase correct mode, symmetrical in all periods. Since the OCR1x Registers are updated at BOTTOM, the length of the rising and the falling slopes will always be equal. This gives symmetrical output pulses and is therefore frequency correct.

Using the ICR1 Register for defining TOP works well when using fixed TOP values. By using ICR1, the OCR1A Register is free to be used for generating a PWM output on OC1A. However, if the base PWM frequency is actively changed by changing the TOP value, using the OCR1A as TOP is clearly a better choice due to its double buffer feature.

In phase and frequency correct PWM mode, the compare units allow generation of PWM waveforms on the OC1x pins. Setting the COM1x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1x1:0 to three (See [Table on page 135](#)). The actual OC1x value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1x). The PWM waveform is generated by setting (or clearing) the OC1x Register at the compare match between OCR1x and TCNT1 when the counter increments, and clearing (or setting) the OC1x Register at compare match between OCR1x and TCNT1 when the counter decrements. The PWM frequency when using phase and frequency correct PWM can be calculated by the following equation:

$$f_{OCnxPFCPWM} = \frac{f_{clk_I/O}}{2 \cdot N \cdot TOP}$$

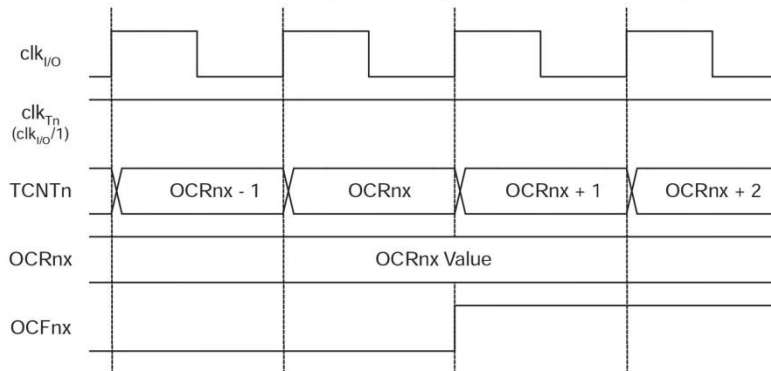
The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1x Register represents special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR1x is set equal to BOTTOM the output will be continuously low and if set equal to TOP the output will be set to high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values. If OCR1A is used to define the TOP value (WGM13:0 = 9) and COM1A1:0 = 1, the OC1A output will toggle with a 50% duty cycle.

15.10 Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{Tn}) is therefore shown as a clock enable signal in the following figures. The figures include information on when Interrupt Flags are set, and when the OCR1x Register is updated with the OCR1x buffer value (only for modes utilizing double buffering). [Figure 15-10](#) shows a timing diagram for the setting of OCF1x.

Figure 15-10. Timer/Counter Timing Diagram, Setting of OCF1x, no Prescaling



[Figure 15-11](#) shows the same timing data, but with the prescaler enabled.

Figure 15-11. Timer/Counter Timing Diagram, Setting of OCF1x, with Prescaler ($f_{clk_I/O}/8$)

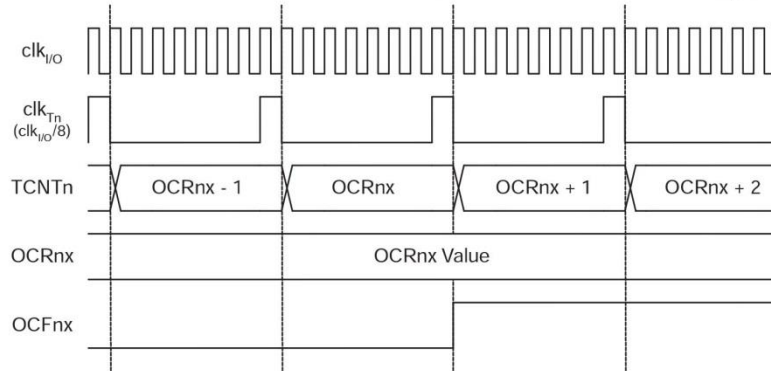


Figure 15-12 shows the count sequence close to TOP in various modes. When using phase and frequency correct PWM mode the OCR1x Register is updated at BOTTOM. The timing diagrams will be the same, but TOP should be replaced by BOTTOM, TOP-1 by BOTTOM+1 and so on. The same renaming applies for modes that set the TOV1 Flag at BOTTOM.

Figure 15-12. Timer/Counter Timing Diagram, no Prescaling

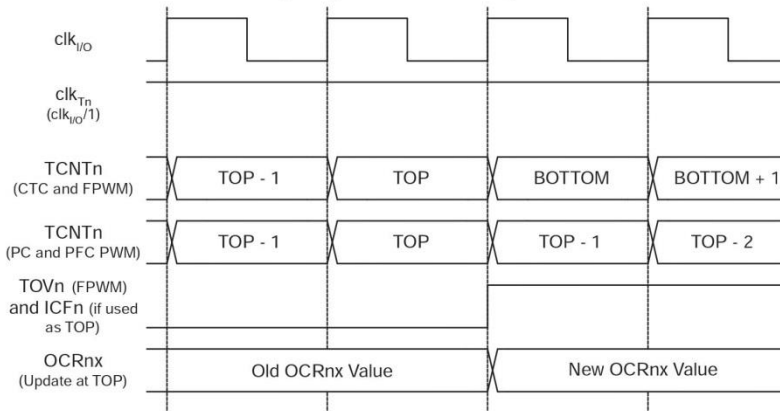
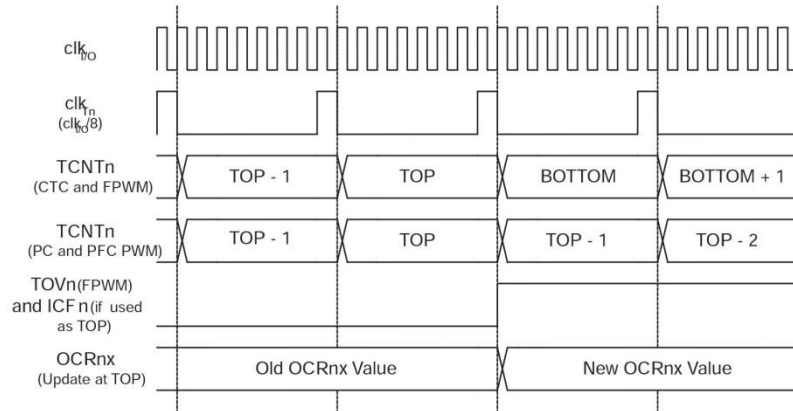


Figure 15-13 shows the same timing data, but with the prescaler enabled.

Figure 15-13. Timer/Counter Timing Diagram, with Prescaler ($f_{clk_{I/O}}/8$)



15.11 Register Description

15.11.1 TCCR1A – Timer/Counter1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A**

- **Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B**

The COM1A1:0 and COM1B1:0 control the Output Compare pins (OC1A and OC1B respectively) behavior. If one or both of the COM1A1:0 bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bit are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the *Data Direction Register (DDR)* bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

When the OC1A or OC1B is connected to the pin, the function of the COM1x1:0 bits is dependent of the WGM13:0 bits setting. Table 15-1 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to a Normal or a CTC mode (non-PWM).

Table 15-1. Compare Output Mode, non-PWM

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on Compare Match.
1	0	Clear OC1A/OC1B on Compare Match (Set output to low level).
1	1	Set OC1A/OC1B on Compare Match (Set output to high level).

Table 15-2 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

Table 15-2. Compare Output Mode, Fast PWM⁽¹⁾

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. In this case the compare match is ignored, but the set or clear is done at BOTTOM. See Section "15.9.3" on page 126. for more details.

Table 15-3 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the phase correct or the phase and frequency correct, PWM mode.

Table 15-3. Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM⁽¹⁾

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 9 or 11: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match when up-counting. Set OC1A/OC1B on Compare Match when downcounting.
1	1	Set OC1A/OC1B on Compare Match when up-counting. Clear OC1A/OC1B on Compare Match when downcounting.

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. See Section "15.9.4" on page 128. for more details.

• **Bit 1:0 – WGM11:0: Waveform Generation Mode**

Combined with the WGM13:2 bits found in the TCCR1B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 15-4. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare match (CTC) mode, and three types of Pulse Width Modulation (PWM) modes. (See Section "15.9" on page 125.).

Table 15-4. Waveform Generation Mode Bit Description⁽¹⁾

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Note: 1. The CTC1 and PWM11:0 bit definition names are obsolete. Use the WGM12:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

15.11.2 TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ICNC1: Input Capture Noise Canceler**

Setting this bit (to one) activates the Input Capture Noise Canceler. When the noise canceler is activated, the input from the Input Capture pin (ICP1) is filtered. The filter function requires four successive equal valued samples of the ICP1 pin for changing its output. The Input Capture is therefore delayed by four Oscillator cycles when the noise canceler is enabled.

- **Bit 6 – ICES1: Input Capture Edge Select**

This bit selects which edge on the Input Capture pin (ICP1) that is used to trigger a capture event. When the ICES1 bit is written to zero, a falling (negative) edge is used as trigger, and when the ICES1 bit is written to one, a rising (positive) edge will trigger the capture.

When a capture is triggered according to the ICES1 setting, the counter value is copied into the Input Capture Register (ICR1). The event will also set the Input Capture Flag (ICF1), and this can be used to cause an Input Capture Interrupt, if this interrupt is enabled.

When the ICR1 is used as TOP value (see description of the WGM13:0 bits located in the TCCR1A and the TCCR1B Register), the ICP1 is disconnected and consequently the Input Capture function is disabled.

• **Bit 5 – Reserved Bit**

This bit is reserved for future use. For ensuring compatibility with future devices, this bit must be written to zero when TCCR1B is written.

• **Bit 4:3 – WGM13:2: Waveform Generation Mode**

See TCCR1A Register description.

• **Bit 2:0 – CS12:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter, see [Figure 15-10](#) and [Figure 15-11](#).

Table 15-5. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk _{IO} /1 (No prescaling)
0	1	0	clk _{IO} /8 (From prescaler)
0	1	1	clk _{IO} /64 (From prescaler)
1	0	0	clk _{IO} /256 (From prescaler)
1	0	1	clk _{IO} /1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

If external pin modes are used for the Timer/Counter1, transitions on the T1 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

15.11.3 TCCR1C – Timer/Counter1 Control Register C

Bit	7	6	5	4	3	2	1	0	
(0x82)	FOC1A FOC1B – – – – –								TCCR1C
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

• **Bit 7 – FOC1A: Force Output Compare for Channel A**

• **Bit 6 – FOC1B: Force Output Compare for Channel B**

The FOC1A/FOC1B bits are only active when the WGM13:0 bits specifies a non-PWM mode. When writing a logical one to the FOC1A/FOC1B bit, an immediate compare match is forced on the Waveform Generation unit. The OC1A/OC1B output is changed according to its COM1x1:0 bits setting. Note that the FOC1A/FOC1B bits are implemented as strobes. Therefore it is the value present in the COM1x1:0 bits that determine the effect of the forced compare.

A FOC1A/FOC1B strobe will not generate any interrupt nor will it clear the timer in Clear Timer on Compare match (CTC) mode using OCR1A as TOP. The FOC1A/FOC1B bits are always read as zero.

15.11.4 TCNT1H and TCNT1L – Timer/Counter1

Bit	7	6	5	4	3	2	1	0	
(0x85)	TCNT1[15:8]								TCNT1H
(0x84)	TCNT1[7:0]								TCNT1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The two *Timer/Counter I/O* locations (TCNT1H and TCNT1L, combined TCNT1) give direct access, both for read and for write operations, to the Timer/Counter unit 16-bit counter. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary High Byte Register (TEMP). This temporary register is shared by all the other 16-bit registers. See Section "15.3" on page 115.

Modifying the counter (TCNT1) while the counter is running introduces a risk of missing a compare match between TCNT1 and one of the OCR1x Registers.

Writing to the TCNT1 Register blocks (removes) the compare match on the following timer clock for all compare units.

15.11.5 OCR1AH and OCR1AL – Output Compare Register 1 A

Bit	7	6	5	4	3	2	1	0	
(0x89)	OCR1A[15:8]								OCR1AH
(0x88)	OCR1A[7:0]								OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

15.11.6 OCR1BH and OCR1BL – Output Compare Register 1 B

Bit	7	6	5	4	3	2	1	0	
(0x8B)	OCR1B[15:8]								OCR1BH
(0x8A)	OCR1B[7:0]								OCR1BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Registers contain a 16-bit value that is continuously compared with the counter value (TCNT1). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC1x pin.

The Output Compare Registers are 16-bit in size. To ensure that both the high and low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit temporary High Byte Register (TEMP). This temporary register is shared by all the other 16-bit registers. See Section "15.3" on page 115.

15.11.7 ICR1H and ICR1L – Input Capture Register 1

Bit	7	6	5	4	3	2	1	0	
(0x87)	ICR1[15:8]								ICR1H
(0x86)	ICR1[7:0]								ICR1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Input Capture is updated with the counter (TCNT1) value each time an event occurs on the ICP1 pin (or optionally on the Analog Comparator output for Timer/Counter1). The Input Capture can be used for defining the counter TOP value.

ATmega48PA/88PA/168PA/328P

The Input Capture Register is 16-bit in size. To ensure that both the high and low bytes are read simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary High Byte Register (TEMP). This temporary register is shared by all the other 16-bit registers. See Section "15.3" on page 115.

15.11.8 TIMSK1 – Timer/Counter1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6F)	-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 5 – ICIE1: Timer/Counter1, Input Capture Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Input Capture interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 57) is executed when the ICF1 Flag, located in TIFR1, is set.

- **Bit 4, 3 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 2 – OCIE1B: Timer/Counter1, Output Compare B Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare B Match interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 57) is executed when the OCF1B Flag, located in TIFR1, is set.

- **Bit 1 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare A Match interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 57) is executed when the OCF1A Flag, located in TIFR1, is set.

- **Bit 0 – TOIE1: Timer/Counter1, Overflow Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Overflow interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 57) is executed when the TOV1 Flag, located in TIFR1, is set.

15.11.9 TIFR1 – Timer/Counter1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 5 – ICF1: Timer/Counter1, Input Capture Flag**

This flag is set when a capture event occurs on the ICP1 pin. When the Input Capture Register (ICR1) is set by the WGM13:0 to be used as the TOP value, the ICF1 Flag is set when the counter reaches the TOP value.

ICF1 is automatically cleared when the Input Capture Interrupt Vector is executed. Alternatively, ICF1 can be cleared by writing a logic one to its bit location.

- **Bit 4, 3 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 2 – OCF1B: Timer/Counter1, Output Compare B Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register B (OCR1B).

Note that a Forced Output Compare (FOC1B) strobe will not set the OCF1B Flag.

OCF1B is automatically cleared when the Output Compare Match B Interrupt Vector is executed. Alternatively, OCF1B can be cleared by writing a logic one to its bit location.

- **Bit 1 – OCF1A: Timer/Counter1, Output Compare A Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register A (OCR1A).

Note that a Forced Output Compare (FOC1A) strobe will not set the OCF1A Flag.

OCF1A is automatically cleared when the Output Compare Match A Interrupt Vector is executed. Alternatively, OCF1A can be cleared by writing a logic one to its bit location.

- **Bit 0 – TOV1: Timer/Counter1, Overflow Flag**

The setting of this flag is dependent of the WGM13:0 bits setting. In Normal and CTC modes, the TOV1 Flag is set when the timer overflows. Refer to [Table 15-4 on page 136](#) for the TOV1 Flag behavior when using another WGM13:0 bit setting.

TOV1 is automatically cleared when the Timer/Counter1 Overflow Interrupt Vector is executed. Alternatively, TOV1 can be cleared by writing a logic one to its bit location.

16. Timer/Counter0 and Timer/Counter1 Prescalers

"8-bit Timer/Counter0 with PWM" on page 94 and "16-bit Timer/Counter1 with PWM" on page 113 share the same prescaler module, but the Timer/Counters can have different prescaler settings. The description below applies to both Timer/Counter1 and Timer/Counter0.

16.1 Internal Clock Source

The Timer/Counter can be clocked directly by the system clock (by setting the CSn2:0 = 1). This provides the fastest operation, with a maximum Timer/Counter clock frequency equal to system clock frequency ($f_{CLK_I/O}$). Alternatively, one of four taps from the prescaler can be used as a clock source. The prescaled clock has a frequency of either $f_{CLK_I/O}/8$, $f_{CLK_I/O}/64$, $f_{CLK_I/O}/256$, or $f_{CLK_I/O}/1024$.

16.2 Prescaler Reset

The prescaler is free running, i.e., operates independently of the Clock Select logic of the Timer/Counter, and it is shared by Timer/Counter1 and Timer/Counter0. Since the prescaler is not affected by the Timer/Counter's clock select, the state of the prescaler will have implications for situations where a prescaled clock is used. One example of prescaling artifacts occurs when the timer is enabled and clocked by the prescaler ($6 > CSn2:0 > 1$). The number of system clock cycles from when the timer is enabled to the first count occurs can be from 1 to N+1 system clock cycles, where N equals the prescaler divisor (8, 64, 256, or 1024).

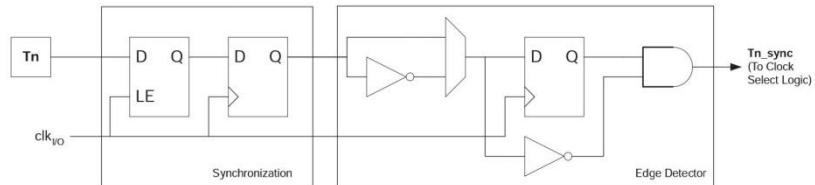
It is possible to use the prescaler reset for synchronizing the Timer/Counter to program execution. However, care must be taken if the other Timer/Counter that shares the same prescaler also uses prescaling. A prescaler reset will affect the prescaler period for all Timer/Counters it is connected to.

16.3 External Clock Source

An external clock source applied to the T1/T0 pin can be used as Timer/Counter clock (clk_{T1}/clk_{T0}). The T1/T0 pin is sampled once every system clock cycle by the pin synchronization logic. The synchronized (sampled) signal is then passed through the edge detector. Figure 16-1 shows a functional equivalent block diagram of the T1/T0 synchronization and edge detector logic. The registers are clocked at the positive edge of the internal system clock ($clk_{I/O}$). The latch is transparent in the high period of the internal system clock.

The edge detector generates one clk_{T1}/clk_{T0} pulse for each positive ($CSn2:0 = 7$) or negative ($CSn2:0 = 6$) edge it detects.

Figure 16-1. T1/T0 Pin Sampling



The synchronization and edge detector logic introduces a delay of 2.5 to 3.5 system clock cycles from an edge has been applied to the T1/T0 pin to the counter is updated.

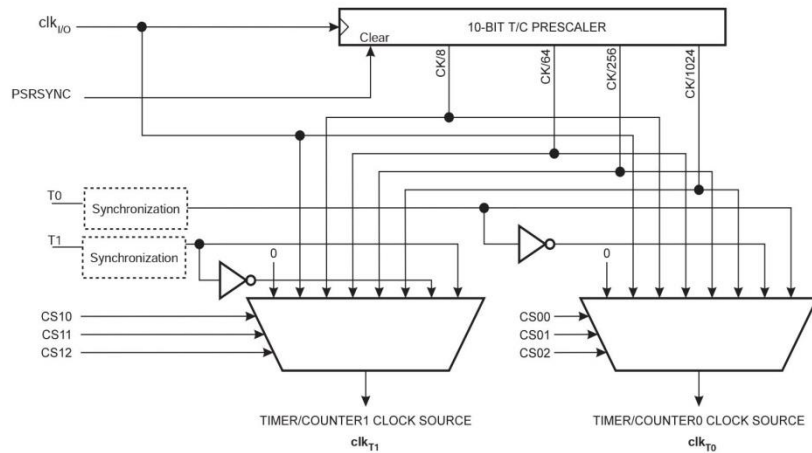
ATmega48PA/88PA/168PA/328P

Enabling and disabling of the clock input must be done when T1/T0 has been stable for at least one system clock cycle, otherwise it is a risk that a false Timer/Counter clock pulse is generated.

Each half period of the external clock applied must be longer than one system clock cycle to ensure correct sampling. The external clock must be guaranteed to have less than half the system clock frequency ($f_{ExtClk} < f_{clk_{I/O}}/2$) given a 50/50% duty cycle. Since the edge detector uses sampling, the maximum frequency of an external clock it can detect is half the sampling frequency (Nyquist sampling theorem). However, due to variation of the system clock frequency and duty cycle caused by Oscillator source (crystal, resonator, and capacitors) tolerances, it is recommended that maximum frequency of an external clock source is less than $f_{clk_{I/O}}/2.5$.

An external clock source can not be prescaled.

Figure 16-2. Prescaler for Timer/Counter0 and Timer/Counter1⁽¹⁾



Note: 1. The synchronization logic on the input pins (T1/T0) is shown in Figure 16-1.

16.4 Register Description

16.4.1 GTCCR – General Timer/Counter Control Register

Bit	7	6	5	4	3	2	1	0	
0x23 (0x43)	TSM	-	-	-	-	-	PSRASY	PSRSYNC	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – TSM: Timer/Counter Synchronization Mode**

Writing the TSM bit to one activates the Timer/Counter Synchronization mode. In this mode, the value that is written to the PSRASY and PSRSYNC bits is kept, hence keeping the corresponding prescaler reset signals asserted. This ensures that the corresponding Timer/Counters are halted and can be configured to the same value without the risk of one of them advancing during configuration. When the TSM bit is written to zero, the PSRASY and PSRSYNC bits are cleared by hardware, and the Timer/Counters start counting simultaneously.

- **Bit 0 – PSRSYNC: Prescaler Reset**

When this bit is one, Timer/Counter1 and Timer/Counter0 prescaler will be Reset. This bit is normally cleared immediately by hardware, except if the TSM bit is set. Note that Timer/Counter1 and Timer/Counter0 share the same prescaler and a reset of this prescaler will affect both timers.

17. 8-bit Timer/Counter2 with PWM and Asynchronous Operation

17.1 Features

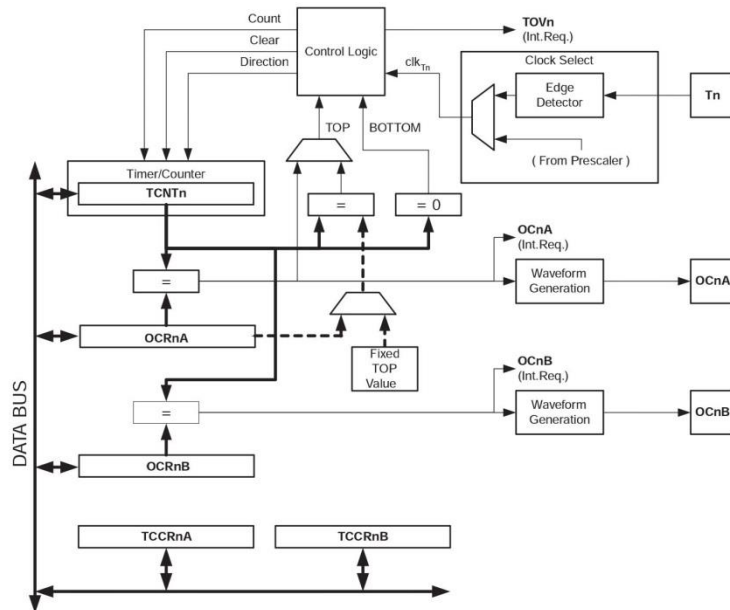
- Single Channel Counter
- Clear Timer on Compare Match (Auto Reload)
- Glitch-free, Phase Correct Pulse Width Modulator (PWM)
- Frequency Generator
- 10-bit Clock Prescaler
- Overflow and Compare Match Interrupt Sources (TOV2, OCF2A and OCF2B)
- Allows Clocking from External 32 kHz Watch Crystal Independent of the I/O Clock

17.2 Overview

Timer/Counter2 is a general purpose, single channel, 8-bit Timer/Counter module. A simplified block diagram of the 8-bit Timer/Counter is shown in Figure 17-1. For the actual placement of I/O pins, refer to "Pinout ATmega48PA/88PA/168PA/328P" on page 2. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the "Register Description" on page 158.

The PRTIM2 bit in "Minimizing Power Consumption" on page 42 must be written to zero to enable Timer/Counter2 module.

Figure 17-1. 8-bit Timer/Counter Block Diagram



17.2.1 Registers

The Timer/Counter (TCNT2) and Output Compare Register (OCR2A and OCR2B) are 8-bit registers. Interrupt request (shorten as Int.Req.) signals are all visible in the Timer Interrupt Flag Register (TIFR2). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK2). TIFR2 and TIMSK2 are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or asynchronously clocked from the TOSC1/2 pins, as detailed later in this section. The asynchronous operation is controlled by the Asynchronous Status Register (ASSR). The Clock Select logic block controls which clock source the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{T2}).

The double buffered Output Compare Register (OCR2A and OCR2B) are compared with the Timer/Counter value at all times. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pins (OC2A and OC2B). See Section "17.5" on page 146. for details. The compare match event will also set the Compare Flag (OCF2A or OCF2B) which can be used to generate an Output Compare interrupt request.

17.2.2 Definitions

Many register and bit references in this document are written in general form. A lower case "n" replaces the Timer/Counter number, in this case 2. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT2 for accessing Timer/Counter2 counter value and so on.

The definitions in Table 17-1 are also used extensively throughout the section.

Table 17-1. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes zero (0x00).
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR2A Register. The assignment is dependent on the mode of operation.

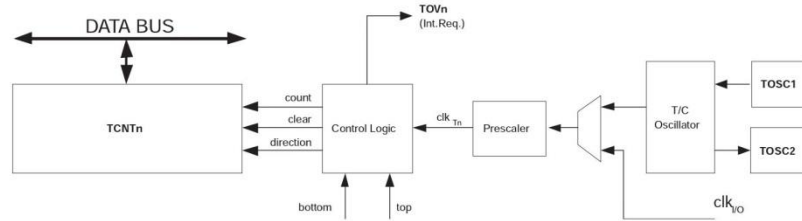
17.3 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal synchronous or an external asynchronous clock source. The clock source clk_{T2} is by default equal to the MCU clock, $clk_{I/O}$. When the AS2 bit in the ASSR Register is written to logic one, the clock source is taken from the Timer/Counter Oscillator connected to TOSC1 and TOSC2. For details on asynchronous operation, see "ASSR – Asynchronous Status Register" on page 164. For details on clock sources and prescaler, see "Timer/Counter Prescaler" on page 156.

17.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 17-2 on page 146 shows a block diagram of the counter and its surrounding environment.

Figure 17-2. Counter Unit Block Diagram



Signal description (internal signals):

count	Increment or decrement TCNT2 by 1.
direction	Selects between increment and decrement.
clear	Clear TCNT2 (set all bits to zero).
clk_{Tn}	Timer/Counter clock, referred to as clk _{T2} in the following.
top	Signalizes that TCNT2 has reached maximum value.
bottom	Signalizes that TCNT2 has reached minimum value (zero).

Depending on the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T2}). clk_{T2} can be generated from an external or internal clock source, selected by the Clock Select bits (CS22:0). When no clock source is selected (CS22:0 = 0) the timer is stopped. However, the TCNT2 value can be accessed by the CPU, regardless of whether clk_{T2} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM21 and WGM20 bits located in the Timer/Counter Control Register (TCCR2A) and the WGM22 located in the Timer/Counter Control Register B (TCCR2B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC2A and OC2B. For more details about advanced counting sequences and waveform generation, see ["Modes of Operation" on page 149](#).

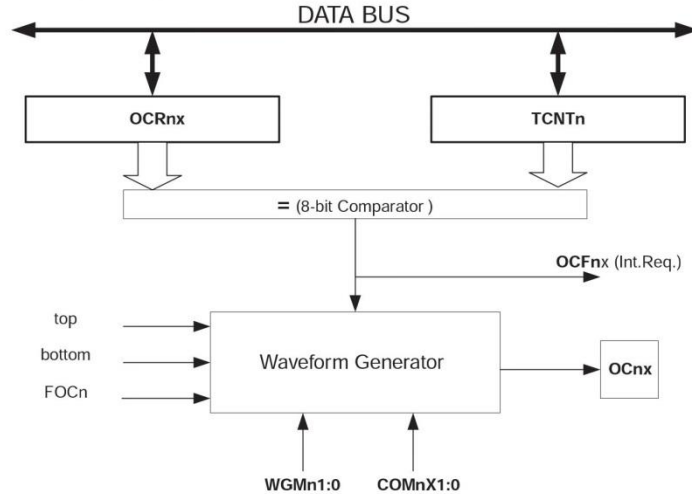
The Timer/Counter Overflow Flag (TOV2) is set according to the mode of operation selected by the WGM22:0 bits. TOV2 can be used for generating a CPU interrupt.

17.5 Output Compare Unit

The 8-bit comparator continuously compares TCNT2 with the Output Compare Register (OCR2A and OCR2B). Whenever TCNT2 equals OCR2A or OCR2B, the comparator signals a match. A match will set the Output Compare Flag (OCF2A or OCF2B) at the next timer clock cycle. If the corresponding interrupt is enabled, the Output Compare Flag generates an Output Compare interrupt. The Output Compare Flag is automatically cleared when the interrupt is executed. Alternatively, the Output Compare Flag can be cleared by software by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to operating mode set by the WGM22:0 bits and Compare Output mode (COM2x1:0) bits. The max and bottom signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (["Modes of Operation" on page 149](#)).

Figure 17-3 shows a block diagram of the Output Compare unit.

Figure 17-3. Output Compare Unit, Block Diagram



The OCR2x Register is double buffered when using any of the Pulse Width Modulation (PWM) modes. For the Normal and Clear Timer on Compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR2x Compare Register to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR2x Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR2x Buffer Register, and if double buffering is disabled the CPU will access the OCR2x directly.

17.5.1 Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the Force Output Compare (FOC2x) bit. Forcing compare match will not set the OCF2x Flag or reload/clear the timer, but the OC2x pin will be updated as if a real compare match had occurred (the COM2x1:0 bits settings define whether the OC2x pin is set, cleared or toggled).

17.5.2 Compare Match Blocking by TCNT2 Write

All CPU write operations to the TCNT2 Register will block any compare match that occurs in the next timer clock cycle, even when the timer is stopped. This feature allows OCR2x to be initialized to the same value as TCNT2 without triggering an interrupt when the Timer/Counter clock is enabled.

17.5.3 Using the Output Compare Unit

Since writing TCNT2 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT2 when using the Output Compare channel, independently of whether the Timer/Counter is running or not. If the value written to TCNT2 equals the OCR2x value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT2 value equal to BOTTOM when the counter is downcounting.

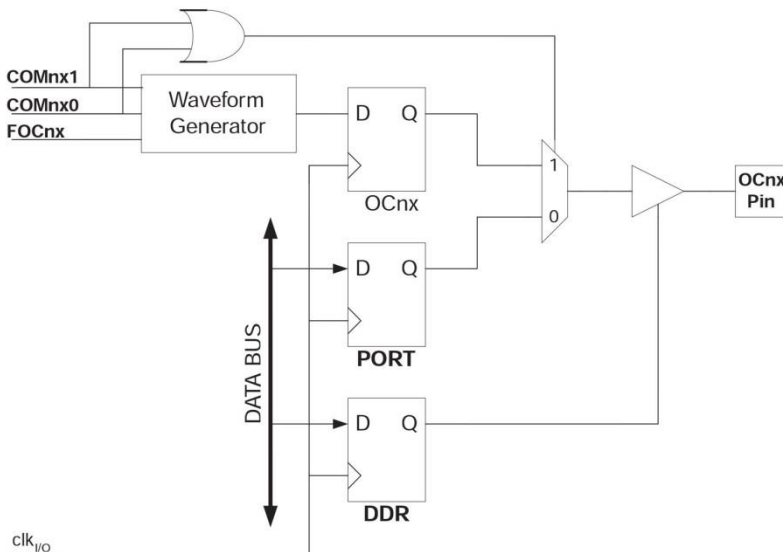
The setup of the OC2x should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC2x value is to use the Force Output Compare (FOC2x) strobe bit in Normal mode. The OC2x Register keeps its value even when changing between Waveform Generation modes.

Be aware that the COM2x1:0 bits are not double buffered together with the compare value. Changing the COM2x1:0 bits will take effect immediately.

17.6 Compare Match Output Unit

The Compare Output mode (COM2x1:0) bits have two functions. The Waveform Generator uses the COM2x1:0 bits for defining the Output Compare (OC2x) state at the next compare match. Also, the COM2x1:0 bits control the OC2x pin output source. [Figure 17-4](#) shows a simplified schematic of the logic affected by the COM2x1:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O Port Control Registers (DDR and PORT) that are affected by the COM2x1:0 bits are shown. When referring to the OC2x state, the reference is for the internal OC2x Register, not the OC2x pin.

Figure 17-4. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC2x) from the Waveform Generator if either of the COM2x1:0 bits are set. However, the OC2x pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The Data Direction Register bit for the OC2x pin (DDR_OC2x) must be set as output before the OC2x value is visible on the pin. The port override function is independent of the Waveform Generation mode.

The design of the Output Compare pin logic allows initialization of the OC2x state before the output is enabled. Note that some COM2x1:0 bit settings are reserved for certain modes of operation. See [Section "17.11" on page 158](#).

17.6.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM2x1:0 bits differently in normal, CTC, and PWM modes. For all modes, setting the COM2x1:0 = 0 tells the Waveform Generator that no action on the OC2x Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to [Table 17-5 on page 159](#). For fast PWM mode, refer to [Table 17-6 on page 159](#), and for phase correct PWM refer to [Table 17-7 on page 160](#).

A change of the COM2x1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC2x strobe bits.

17.7 Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM22:0) and Compare Output mode (COM2x1:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM2x1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM2x1:0 bits control whether the output should be set, cleared, or toggled at a compare match (See [Section "17.6" on page 148](#)).

For detailed timing information refer to ["Timer/Counter Timing Diagrams" on page 153](#).

17.7.1 Normal Mode

The simplest mode of operation is the Normal mode (WGM22:0 = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 8-bit value (TOP = 0xFF) and then restarts from the bottom (0x00). In normal operation the Timer/Counter Overflow Flag (TOV2) will be set in the same timer clock cycle as the TCNT2 becomes zero. The TOV2 Flag in this case behaves like a ninth bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV2 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

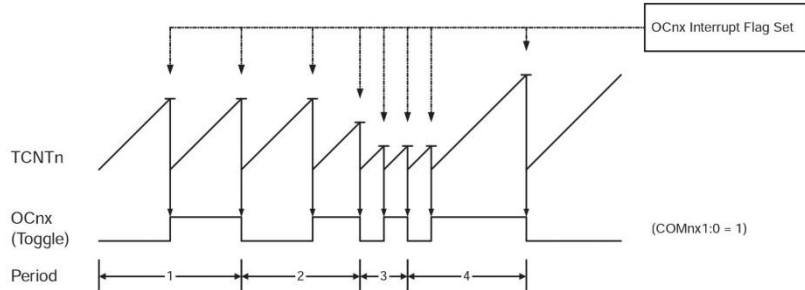
The Output Compare unit can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

17.7.2 Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare or CTC mode (WGM22:0 = 2), the OCR2A Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT2) matches the OCR2A. The OCR2A defines the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in [Figure 17-5](#). The counter value (TCNT2) increases until a compare match occurs between TCNT2 and OCR2A, and then counter (TCNT2) is cleared.

Figure 17-5. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the OCF2A Flag. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR2A is lower than the current value of TCNT2, the counter will miss the compare match. The counter will then have to count to its maximum value (0xFF) and wrap around starting at 0x00 before the compare match can occur.

For generating a waveform output in CTC mode, the OC2A output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode (COM2A1:0 = 1). The OC2A value will not be visible on the port pin unless the data direction for the pin is set to output. The waveform generated will have a maximum frequency of $f_{OC2A} = f_{clk_I/O} / 2$ when OCR2A is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OCnx} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRnx)}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

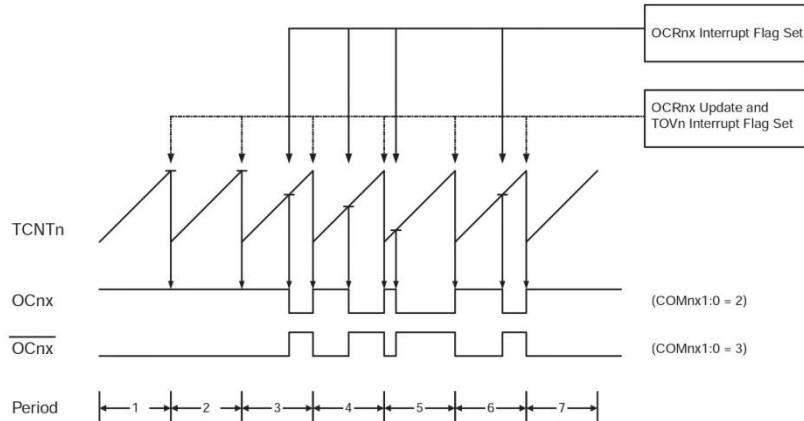
As for the Normal mode of operation, the TOV2 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

17.7.3 Fast PWM Mode

The fast Pulse Width Modulation or fast PWM mode (WGM22:0 = 3 or 7) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to TOP then restarts from BOTTOM. TOP is defined as 0xFF when WGM2:0 = 3, and OCR2A when WGM2:0 = 7. In non-inverting Compare Output mode, the Output Compare (OC2x) is cleared on the compare match between TCNT2 and OCR2x, and set at BOTTOM. In inverting Compare Output mode, the output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that uses dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

In fast PWM mode, the counter is incremented until the counter value matches the TOP value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 17-6. The TCNT2 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT2 slopes represent compare matches between OCR2x and TCNT2.

Figure 17-6. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV2) is set each time the counter reaches TOP. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC2x pin. Setting the COM2x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM2x1:0 to three. TOP is defined as 0xFF when WGM2:0 = 3, and OCR2A when MGM2:0 = 7. (See Table 17-3 on page 158). The actual OC2x value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the OC2x Register at the compare match between OCR2x and TCNT2, and clearing (or setting) the OC2x Register at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{clk_I/O}}{N \cdot 256}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

The extreme values for the OCR2A Register represent special cases when generating a PWM waveform output in the fast PWM mode. If the OCR2A is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR2A equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM2A1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC2x to toggle its logical level on each compare match (COM2x1:0 = 1). The waveform

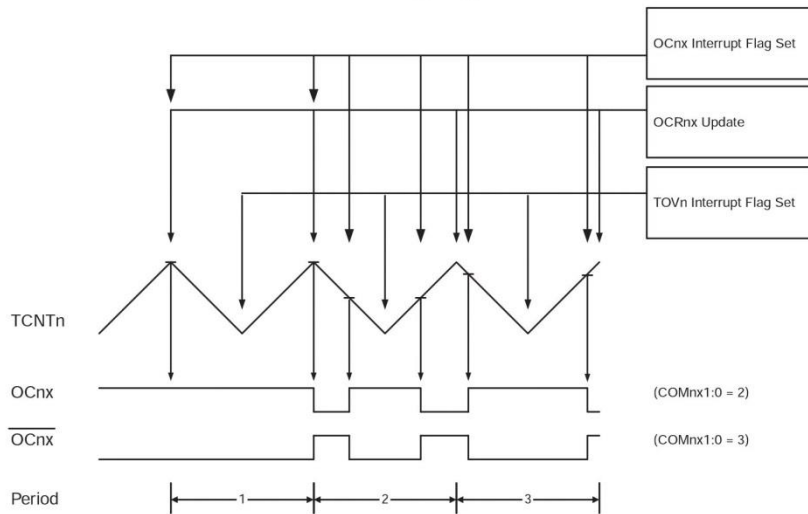
generated will have a maximum frequency of $f_{oc2} = f_{clk_I/O}/2$ when OCR2A is set to zero. This feature is similar to the OC2A toggle in CTC mode, except the double buffer feature of the Output Compare unit is enabled in the fast PWM mode.

17.7.4 Phase Correct PWM Mode

The phase correct PWM mode (WGM22:0 = 1 or 5) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP and then from TOP to BOTTOM. TOP is defined as 0xFF when WGM2:0 = 3, and OCR2A when WGM2:0 = 7. In non-inverting Compare Output mode, the Output Compare (OC2x) is cleared on the compare match between TCNT2 and OCR2x while upcounting, and set on the compare match while downcounting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase correct PWM mode the counter is incremented until the counter value matches TOP. When the counter reaches TOP, it changes the count direction. The TCNT2 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 17-7. The TCNT2 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT2 slopes represent compare matches between OCR2x and TCNT2.

Figure 17-7. Phase Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV2) is set each time the counter reaches BOTTOM. The Interrupt Flag can be used to generate an interrupt each time the counter reaches the BOTTOM value.

In phase correct PWM mode, the compare unit allows generation of PWM waveforms on the OC2x pin. Setting the COM2x1:0 bits to two will produce a non-inverted PWM. An inverted PWM

output can be generated by setting the COM2x1:0 to three. TOP is defined as 0xFF when WGM2:0 = 3, and OCR2A when MGM2:0 = 7 (See Table 17-4 on page 159). The actual OC2x value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by clearing (or setting) the OC2x Register at the compare match between OCR2x and TCNT2 when the counter increments, and setting (or clearing) the OC2x Register at compare match between OCR2x and TCNT2 when the counter decrements. The PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnPCPWM} = \frac{f_{clk_I/O}}{N \cdot 510}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

The extreme values for the OCR2A Register represent special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR2A is set equal to BOTTOM, the output will be continuously low and if set equal to MAX the output will be continuously high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values.

At the very start of period 2 in Figure 17-7 OCn has a transition from high to low even though there is no Compare Match. The point of this transition is to guarantee symmetry around BOTTOM. There are two cases that give a transition without Compare Match.

- OCR2A changes its value from MAX, like in Figure 17-7. When the OCR2A value is MAX the OCn pin value is the same as the result of a down-counting compare match. To ensure symmetry around BOTTOM the OCn value at MAX must correspond to the result of an up-counting Compare Match.
- The timer starts counting from a value higher than the one in OCR2A, and for that reason misses the Compare Match and hence the OCn change that would have happened on the way up.

17.8 Timer/Counter Timing Diagrams

The following figures show the Timer/Counter in synchronous mode, and the timer clock (clk_{T2}) is therefore shown as a clock enable signal. In asynchronous mode, $clk_{I/O}$ should be replaced by the Timer/Counter Oscillator clock. The figures include information on when Interrupt Flags are set. Figure 17-8 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value in all modes other than phase correct PWM mode.

Figure 17-8. Timer/Counter Timing Diagram, no Prescaling

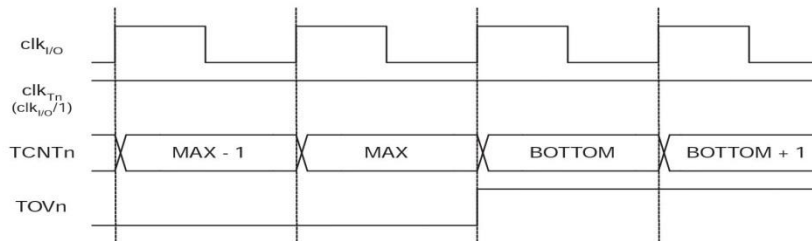


Figure 17-9 shows the same timing data, but with the prescaler enabled.

Figure 17-9. Timer/Counter Timing Diagram, with Prescaler ($f_{clk_{I/O}}/8$)

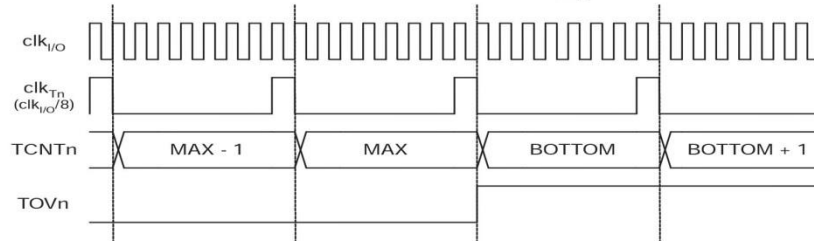


Figure 17-10 shows the setting of OCF2A in all modes except CTC mode.

Figure 17-10. Timer/Counter Timing Diagram, Setting of OCF2A, with Prescaler ($f_{clk_{I/O}}/8$)

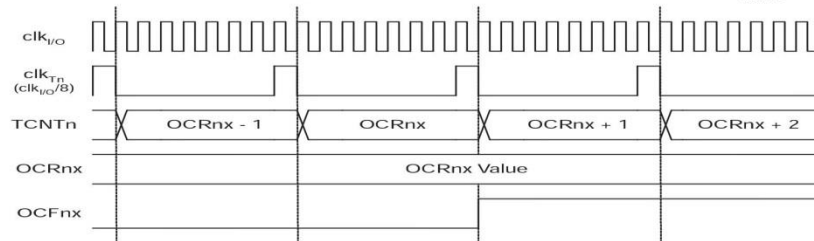
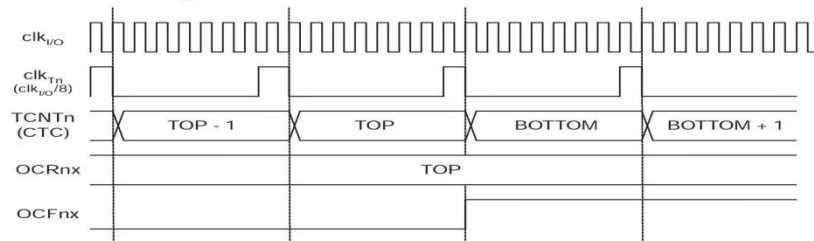


Figure 17-11 shows the setting of OCF2A and the clearing of TCNT2 in CTC mode.

Figure 17-11. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk_{I/O}}/8$)



17.9 Asynchronous Operation of Timer/Counter2

When Timer/Counter2 operates asynchronously, some considerations must be taken.

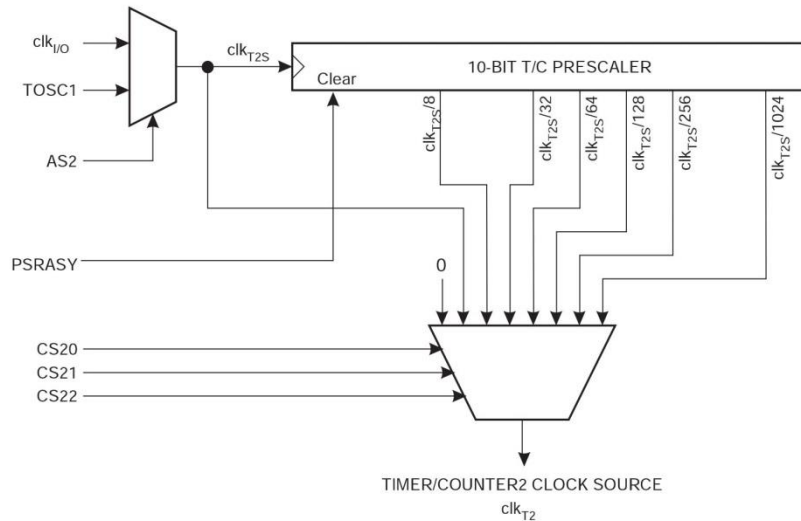
- **Warning:** When switching between asynchronous and synchronous clocking of Timer/Counter2, the Timer Registers TCNT2, OCR2x, and TCCR2x might be corrupted. A safe procedure for switching clock source is:
 - a. Disable the Timer/Counter2 interrupts by clearing OCIE2x and TOIE2.
 - b. Select clock source by setting AS2 as appropriate.
 - c. Write new values to TCNT2, OCR2x, and TCCR2x.
 - d. To switch to asynchronous operation: Wait for TCN2xUB, OCR2xUB, and TCR2xUB.
 - e. Clear the Timer/Counter2 Interrupt Flags.
 - f. Enable interrupts, if needed.
- The CPU main clock frequency must be more than four times the Oscillator frequency.
- When writing to one of the registers TCNT2, OCR2x, or TCCR2x, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each of the five mentioned registers have their individual temporary register, which means that e.g. writing to TCNT2 does not disturb an OCR2x write in progress. To detect that a transfer to the destination register has taken place, the Asynchronous Status Register – ASSR has been implemented.
- When entering Power-save or ADC Noise Reduction mode after having written to TCNT2, OCR2x, or TCCR2x, the user must wait until the written register has been updated if Timer/Counter2 is used to wake up the device. Otherwise, the MCU will enter sleep mode before the changes are effective. This is particularly important if any of the Output Compare2 interrupt is used to wake up the device, since the Output Compare function is disabled during writing to OCR2x or TCNT2. If the write cycle is not finished, and the MCU enters sleep mode before the corresponding OCR2xUB bit returns to zero, the device will never receive a compare match interrupt, and the MCU will not wake up.
- If Timer/Counter2 is used to wake the device up from Power-save or ADC Noise Reduction mode, precautions must be taken if the user wants to re-enter one of these modes: If re-entering sleep mode within the TOSC1 cycle, the interrupt will immediately occur and the device wake up again. The result is multiple interrupts and wake-ups within one TOSC1 cycle from the first interrupt. If the user is in doubt whether the time before re-entering Power-save or ADC Noise Reduction mode is sufficient, the following algorithm can be used to ensure that one TOSC1 cycle has elapsed:
 - a. Write a value to TCCR2x, TCNT2, or OCR2x.
 - b. Wait until the corresponding Update Busy Flag in ASSR returns to zero.
 - c. Enter Power-save or ADC Noise Reduction mode.
- When the asynchronous operation is selected, the 32.768 kHz Oscillator for Timer/Counter2 is always running, except in Power-down and Standby modes. After a Power-up Reset or wake-up from Power-down or Standby mode, the user should be aware of the fact that this Oscillator might take as long as one second to stabilize. The user is advised to wait for at least one second before using Timer/Counter2 after power-up or wake-up from Power-down or Standby mode. The contents of all Timer/Counter2 Registers must be considered lost after a wake-up from Power-down or Standby mode due to unstable clock signal upon start-up, no matter whether the Oscillator is in use or a clock signal is applied to the TOSC1 pin.

- Description of wake up from Power-save or ADC Noise Reduction mode when the timer is clocked asynchronously: When the interrupt condition is met, the wake up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at least one before the processor can read the counter value. After wake-up, the MCU is halted for four cycles, it executes the interrupt routine, and resumes execution from the instruction following SLEEP.
- Reading of the TCNT2 Register shortly after wake-up from Power-save may give an incorrect result. Since TCNT2 is clocked on the asynchronous TOSC clock, reading TCNT2 must be done through a register synchronized to the internal I/O clock domain. Synchronization takes place for every rising TOSC1 edge. When waking up from Power-save mode, and the I/O clock ($clk_{I/O}$) again becomes active, TCNT2 will read as the previous value (before entering sleep) until the next rising TOSC1 edge. The phase of the TOSC clock after waking up from Power-save mode is essentially unpredictable, as it depends on the wake-up time. The recommended procedure for reading TCNT2 is thus as follows:
 - a. Write any value to either of the registers OCR2x or TCCR2x.
 - b. Wait for the corresponding Update Busy Flag to be cleared.
 - c. Read TCNT2.

During asynchronous operation, the synchronization of the Interrupt Flags for the asynchronous timer takes 3 processor cycles plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value causing the setting of the Interrupt Flag. The Output Compare pin is changed on the timer clock and is not synchronized to the processor clock.

17.10 Timer/Counter Prescaler

Figure 17-12. Prescaler for Timer/Counter2



The clock source for Timer/Counter2 is named clk_{T2S} . clk_{T2S} is by default connected to the main system I/O clock $clk_{I/O}$. By setting the AS2 bit in ASSR, Timer/Counter2 is asynchronously clocked from the TOSC1 pin. This enables use of Timer/Counter2 as a Real Time Counter

ATmega48PA/88PA/168PA/328P

(RTC). When AS2 is set, pins TOSC1 and TOSC2 are disconnected from Port C. A crystal can then be connected between the TOSC1 and TOSC2 pins to serve as an independent clock source for Timer/Counter2. The Oscillator is optimized for use with a 32.768 kHz crystal.

For Timer/Counter2, the possible prescaled selections are: $\text{clk}_{T2S}/8$, $\text{clk}_{T2S}/32$, $\text{clk}_{T2S}/64$, $\text{clk}_{T2S}/128$, $\text{clk}_{T2S}/256$, and $\text{clk}_{T2S}/1024$. Additionally, clk_{T2S} as well as 0 (stop) may be selected. Setting the PSRASY bit in GTCCR resets the prescaler. This allows the user to operate with a predictable prescaler.

17.11 Register Description

17.11.1 TCCR2A – Timer/Counter Control Register A

Bit (0xB0)	7	6	5	4	3	2	1	0	TCCR2A
	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7:6 – COM2A1:0: Compare Match Output A Mode

These bits control the Output Compare pin (OC2A) behavior. If one or both of the COM2A1:0 bits are set, the OC2A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC2A pin must be set in order to enable the output driver.

When OC2A is connected to the pin, the function of the COM2A1:0 bits depends on the WGM22:0 bit setting. Table 17-2 shows the COM2A1:0 bit functionality when the WGM22:0 bits are set to a normal or CTC mode (non-PWM).

Table 17-2. Compare Output Mode, non-PWM Mode

COM2A1	COM2A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC2A on Compare Match
1	0	Clear OC2A on Compare Match
1	1	Set OC2A on Compare Match

Table 17-3 shows the COM2A1:0 bit functionality when the WGM21:0 bits are set to fast PWM mode.

Table 17-3. Compare Output Mode, Fast PWM Mode⁽¹⁾

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	WGM22 = 0: Normal Port Operation, OC0A Disconnected. WGM22 = 1: Toggle OC2A on Compare Match.
1	0	Clear OC2A on Compare Match, set OC2A at BOTTOM, (non-inverting mode).
1	1	Set OC2A on Compare Match, clear OC2A at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR2A equals TOP and COM2A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at BOTTOM. See “Fast PWM Mode” on page 150 for more details.

Table 17-4 shows the COM2A1:0 bit functionality when the WGM22:0 bits are set to phase correct PWM mode.

Table 17-4. Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	WGM22 = 0: Normal Port Operation, OC2A Disconnected. WGM22 = 1: Toggle OC2A on Compare Match.
1	0	Clear OC2A on Compare Match when up-counting. Set OC2A on Compare Match when down-counting.
1	1	Set OC2A on Compare Match when up-counting. Clear OC2A on Compare Match when down-counting.

Note: 1. A special case occurs when OCR2A equals TOP and COM2A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 152 for more details.

• **Bits 5:4 – COM2B1:0: Compare Match Output B Mode**

These bits control the Output Compare pin (OC2B) behavior. If one or both of the COM2B1:0 bits are set, the OC2B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC2B pin must be set in order to enable the output driver.

When OC2B is connected to the pin, the function of the COM2B1:0 bits depends on the WGM22:0 bit setting. Table 17-5 shows the COM2B1:0 bit functionality when the WGM22:0 bits are set to a normal or CTC mode (non-PWM).

Table 17-5. Compare Output Mode, non-PWM Mode

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Toggle OC2B on Compare Match
1	0	Clear OC2B on Compare Match
1	1	Set OC2B on Compare Match

Table 17-6 shows the COM2B1:0 bit functionality when the WGM22:0 bits are set to fast PWM mode.

Table 17-6. Compare Output Mode, Fast PWM Mode⁽¹⁾

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Reserved
1	0	Clear OC2B on Compare Match, set OC2B at BOTTOM, (non-inverting mode).
1	1	Set OC2B on Compare Match, clear OC2B at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR2B equals TOP and COM2B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at BOTTOM. See “Phase Correct PWM Mode” on page 152 for more details.

Table 17-7 shows the COM2B1:0 bit functionality when the WGM22:0 bits are set to phase correct PWM mode.

Table 17-7. Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Reserved
1	0	Clear OC2B on Compare Match when up-counting. Set OC2B on Compare Match when down-counting.
1	1	Set OC2B on Compare Match when up-counting. Clear OC2B on Compare Match when down-counting.

Note: 1. A special case occurs when OCR2B equals TOP and COM2B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See “Phase Correct PWM Mode” on page 152 for more details.

• **Bits 3, 2 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

• **Bits 1:0 – WGM21:0: Waveform Generation Mode**

Combined with the WGM22 bit found in the TCCR2B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 17-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (see “Modes of Operation” on page 149).

Table 17-8. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX= 0xFF
2. BOTTOM= 0x00

17.11.2 TCCR2B – Timer/Counter Control Register B

Bit (0xB1)	7	6	5	4	3	2	1	0	TCCR2B
	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	
Read/Write	W	W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC2A: Force Output Compare A**

The FOC2A bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR2B is written when operating in PWM mode. When writing a logical one to the FOC2A bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC2A output is changed according to its COM2A1:0 bits setting. Note that the FOC2A bit is implemented as a strobe. Therefore it is the value present in the COM2A1:0 bits that determines the effect of the forced compare.

A FOC2A strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR2A as TOP.

The FOC2A bit is always read as zero.

- **Bit 6 – FOC2B: Force Output Compare B**

The FOC2B bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR2B is written when operating in PWM mode. When writing a logical one to the FOC2B bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC2B output is changed according to its COM2B1:0 bits setting. Note that the FOC2B bit is implemented as a strobe. Therefore it is the value present in the COM2B1:0 bits that determines the effect of the forced compare.

A FOC2B strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR2B as TOP.

The FOC2B bit is always read as zero.

- **Bits 5:4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 3 – WGM22: Waveform Generation Mode**

See the description in the "TCCR2A – Timer/Counter Control Register A" on page 158.

- **Bit 2:0 – CS22:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter, see Table 17-9 on page 162.

Table 17-9. Clock Select Bit Description

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk _{T2S} /(No prescaling)
0	1	0	clk _{T2S} /8 (From prescaler)
0	1	1	clk _{T2S} /32 (From prescaler)
1	0	0	clk _{T2S} /64 (From prescaler)
1	0	1	clk _{T2S} /128 (From prescaler)
1	1	0	clk _{T2S} /256 (From prescaler)
1	1	1	clk _{T2S} /1024 (From prescaler)

If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

17.11.3 TCNT2 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
(0xB2)	TCNT2[7:0]								TCNT2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT2 Register blocks (removes) the Compare Match on the following timer clock. Modifying the counter (TCNT2) while the counter is running, introduces a risk of missing a Compare Match between TCNT2 and the OCR2x Registers.

17.11.4 OCR2A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
(0xB3)	OCR2A[7:0]								OCR2A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register A contains an 8-bit value that is continuously compared with the counter value (TCNT2). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC2A pin.

17.11.5 OCR2B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
(0xB4)	OCR2B[7:0]								OCR2B
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register B contains an 8-bit value that is continuously compared with the counter value (TCNT2). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC2B pin.

17.11.6 TIMSK2 – Timer/Counter2 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x70)	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2	TIMSK2
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2 – OCIE2B: Timer/Counter2 Output Compare Match B Interrupt Enable**

When the OCIE2B bit is written to one and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match B interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter2 occurs, i.e., when the OCF2B bit is set in the Timer/Counter 2 Interrupt Flag Register – TIFR2.

- **Bit 1 – OCIE2A: Timer/Counter2 Output Compare Match A Interrupt Enable**

When the OCIE2A bit is written to one and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match A interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter2 occurs, i.e., when the OCF2A bit is set in the Timer/Counter 2 Interrupt Flag Register – TIFR2.

- **Bit 0 – TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is written to one and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter2 occurs, i.e., when the TOV2 bit is set in the Timer/Counter2 Interrupt Flag Register – TIFR2.

17.11.7 TIFR2 – Timer/Counter2 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x17 (0x37)	-	-	-	-	-	OCF2B	OCF2A	TOV2	TIFR2
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2 – OCF2B: Output Compare Flag 2 B**

The OCF2B bit is set (one) when a compare match occurs between the Timer/Counter2 and the data in OCR2B – Output Compare Register2. OCF2B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF2B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE2B (Timer/Counter2 Compare match Interrupt Enable), and OCF2B are set (one), the Timer/Counter2 Compare match Interrupt is executed.

- **Bit 1 – OCF2A: Output Compare Flag 2 A**

The OCF2A bit is set (one) when a compare match occurs between the Timer/Counter2 and the data in OCR2A – Output Compare Register2. OCF2A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF2A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE2A (Timer/Counter2 Compare match Interrupt Enable), and OCF2A are set (one), the Timer/Counter2 Compare match Interrupt is executed.

- **Bit 0 – TOV2: Timer/Counter2 Overflow Flag**

The TOV2 bit is set (one) when an overflow occurs in Timer/Counter2. TOV2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV2 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE2A (Timer/Counter2 Overflow Interrupt Enable), and TOV2 are set (one), the Timer/Counter2 Overflow interrupt is executed. In PWM mode, this bit is set when Timer/Counter2 changes counting direction at 0x00.

17.11.8 ASSR – Asynchronous Status Register

Bit	7	6	5	4	3	2	1	0	
(0xB6)	–	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	ASSR
Read/Write	R	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RES: Reserved bit**

This bit is reserved and will always read as zero.

- **Bit 6 – EXCLK: Enable External Clock Input**

When EXCLK is written to one, and asynchronous clock is selected, the external clock input buffer is enabled and an external clock can be input on Timer Oscillator 1 (TOSC1) pin instead of a 32 kHz crystal. Writing to EXCLK should be done before asynchronous operation is selected. Note that the crystal Oscillator will only run when this bit is zero.

- **Bit 5 – AS2: Asynchronous Timer/Counter2**

When AS2 is written to zero, Timer/Counter2 is clocked from the I/O clock, $clk_{I/O}$. When AS2 is written to one, Timer/Counter2 is clocked from a crystal Oscillator connected to the Timer Oscillator 1 (TOSC1) pin. When the value of AS2 is changed, the contents of TCNT2, OCR2A, OCR2B, TCCR2A and TCCR2B might be corrupted.

- **Bit 4 – TCN2UB: Timer/Counter2 Update Busy**

When Timer/Counter2 operates asynchronously and TCNT2 is written, this bit becomes set. When TCNT2 has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCNT2 is ready to be updated with a new value.

- **Bit 3 – OCR2AUB: Output Compare Register2 Update Busy**

When Timer/Counter2 operates asynchronously and OCR2A is written, this bit becomes set. When OCR2A has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that OCR2A is ready to be updated with a new value.

- **Bit 2 – OCR2BUB: Output Compare Register2 Update Busy**

When Timer/Counter2 operates asynchronously and OCR2B is written, this bit becomes set. When OCR2B has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that OCR2B is ready to be updated with a new value.

- **Bit 1 – TCR2AUB: Timer/Counter Control Register2 Update Busy**

When Timer/Counter2 operates asynchronously and TCCR2A is written, this bit becomes set. When TCCR2A has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR2A is ready to be updated with a new value.

- **Bit 0 – TCR2BUB: Timer/Counter Control Register2 Update Busy**

When Timer/Counter2 operates asynchronously and TCCR2B is written, this bit becomes set. When TCCR2B has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR2B is ready to be updated with a new value.

If a write is performed to any of the five Timer/Counter2 Registers while its update busy flag is set, the updated value might get corrupted and cause an unintentional interrupt to occur.

ATmega48PA/88PA/168PA/328P

The mechanisms for reading TCNT2, OCR2A, OCR2B, TCCR2A and TCCR2B are different. When reading TCNT2, the actual timer value is read. When reading OCR2A, OCR2B, TCCR2A and TCCR2B the value in the temporary storage register is read.

17.11.9 GTCCR – General Timer/Counter Control Register

Bit	7	6	5	4	3	2	1	0	
0x23 (0x43)	TSM	–	–	–	–	–	PSRASY	PSRSYNC	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – PSRASY: Prescaler Reset Timer/Counter2**

When this bit is one, the Timer/Counter2 prescaler will be reset. This bit is normally cleared immediately by hardware. If the bit is written when Timer/Counter2 is operating in asynchronous mode, the bit will remain one until the prescaler has been reset. The bit will not be cleared by hardware if the TSM bit is set. Refer to the description of the [“Bit 7 – TSM: Timer/Counter Synchronization Mode”](#) on page 143 for a description of the Timer/Counter Synchronization mode.

18. SPI – Serial Peripheral Interface

18.1 Features

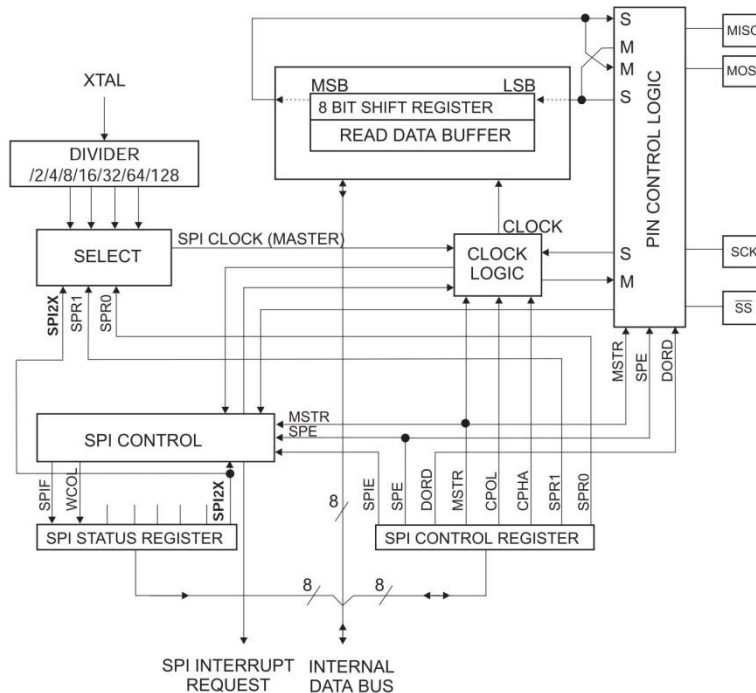
- Full-duplex, Three-wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Seven Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wake-up from Idle Mode
- Double Speed (CK/2) Master SPI Mode

18.2 Overview

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATmega48PA/88PA/168PA/328P and peripheral devices or between several AVR devices.

The USART can also be used in Master SPI mode, see "USART in SPI Mode" on page 204. The PRSPI bit in "Minimizing Power Consumption" on page 42 must be written to zero to enable SPI module.

Figure 18-1. SPI Block Diagram⁽¹⁾



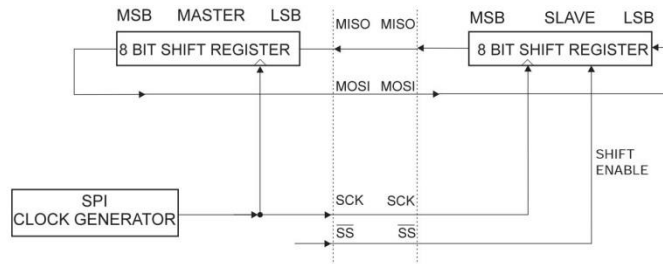
Note: 1. Refer to Figure 1-1 on page 2, and Table 13-3 on page 82 for SPI pin placement.

The interconnection between Master and Slave CPUs with SPI is shown in [Figure 18-2 on page 167](#). The system consists of two shift Registers, and a Master clock generator. The SPI Master initiates the communication cycle when pulling low the Slave Select \overline{SS} pin of the desired Slave. Master and Slave prepare the data to be sent in their respective shift Registers, and the Master generates the required clock pulses on the SCK line to interchange data. Data is always shifted from Master to Slave on the Master Out – Slave In, MOSI, line, and from Slave to Master on the Master In – Slave Out, MISO, line. After each data packet, the Master will synchronize the Slave by pulling high the Slave Select, \overline{SS} , line.

When configured as a Master, the SPI interface has no automatic control of the \overline{SS} line. This must be handled by user software before communication can start. When this is done, writing a byte to the SPI Data Register starts the SPI clock generator, and the hardware shifts the eight bits into the Slave. After shifting one byte, the SPI clock generator stops, setting the end of Transmission Flag (SPIF). If the SPI Interrupt Enable bit (SPIE) in the SPCR Register is set, an interrupt is requested. The Master may continue to shift the next byte by writing it into SPDR, or signal the end of packet by pulling high the Slave Select, \overline{SS} line. The last incoming byte will be kept in the Buffer Register for later use.

When configured as a Slave, the SPI interface will remain sleeping with MISO tri-stated as long as the \overline{SS} pin is driven high. In this state, software may update the contents of the SPI Data Register, SPDR, but the data will not be shifted out by incoming clock pulses on the SCK pin until the \overline{SS} pin is driven low. As one byte has been completely shifted, the end of Transmission Flag, SPIF is set. If the SPI Interrupt Enable bit, SPIE, in the SPCR Register is set, an interrupt is requested. The Slave may continue to place new data to be sent into SPDR before reading the incoming data. The last incoming byte will be kept in the Buffer Register for later use.

Figure 18-2. SPI Master-slave Interconnection



The system is single buffered in the transmit direction and double buffered in the receive direction. This means that bytes to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received character must be read from the SPI Data Register before the next character has been completely shifted in. Otherwise, the first byte is lost.

In SPI Slave mode, the control logic will sample the incoming signal of the SCK pin. To ensure correct sampling of the clock signal, the minimum low and high periods should be:

Low periods: Longer than 2 CPU clock cycles.

High periods: Longer than 2 CPU clock cycles.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK, and \overline{SS} pins is overridden according to [Table 18-1 on page 168](#). For more details on automatic port overrides, refer to ["Alternate Port Functions" on page 80](#).

Table 18-1. SPI Pin Overrides^(Note:)

Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
\overline{SS}	User Defined	Input

Note: See ["Alternate Functions of Port B" on page 82](#) for a detailed description of how to define the direction of the user defined SPI pins.

The following code examples show how to initialize the SPI as a Master and how to perform a simple transmission. `DDR_SPI` in the examples must be replaced by the actual Data Direction Register controlling the SPI pins. `DD_MOSI`, `DD_MISO` and `DD_SCK` must be replaced by the actual data direction bits for these pins. E.g. if MOSI is placed on pin PB5, replace `DD_MOSI` with `DDB5` and `DDR_SPI` with `DDRB`.

Assembly Code Example⁽¹⁾

```
SPI_MasterInit:
; Set MOSI and SCK output, all others input
ldi r17, (1<<DD_MOSI) | (1<<DD_SCK)
out DDR_SPI, r17
; Enable SPI, Master, set clock rate fck/16
ldi r17, (1<<SPE) | (1<<MSTR) | (1<<SPR0)
out SPCR, r17
ret

SPI_MasterTransmit:
; Start transmission of data (r16)
out SPDR, r16
Wait_Transmit:
; Wait for transmission complete
in r16, SPSR
sbrsr16, SPIF
rjmp Wait_Transmit
ret
```

C Code Example⁽¹⁾

```
void SPI_MasterInit(void)
{
    /* Set MOSI and SCK output, all others input */
    DDR_SPI = (1<<DD_MOSI) | (1<<DD_SCK);
    /* Enable SPI, Master, set clock rate fck/16 */
    SPCR = (1<<SPE) | (1<<MSTR) | (1<<SPR0);
}

void SPI_MasterTransmit(char cData)
{
    /* Start transmission */
    SPDR = cData;
    /* Wait for transmission complete */
    while (!(SPSR & (1<<SPIF)))
        ;
}
```

Note: 1. See "About Code Examples" on page 7.

The following code examples show how to initialize the SPI as a Slave and how to perform a simple reception.

Assembly Code Example⁽¹⁾

```
SPI_SlaveInit:
    ; Set MISO output, all others input
    ldi r17, (1<<DD_MISO)
    out DDR_SPI,r17
    ; Enable SPI
    ldi r17, (1<<SPE)
    out SPCR,r17
    ret

SPI_SlaveReceive:
    ; Wait for reception complete
    sbis SPSR,SPIF
    rjmp SPI_SlaveReceive
    ; Read received data and return
    in r16,SPDR
    ret
```

C Code Example⁽¹⁾

```
void SPI_SlaveInit(void)
{
    /* Set MISO output, all others input */
    DDR_SPI = (1<<DD_MISO);
    /* Enable SPI */
    SPCR = (1<<SPE);
}

char SPI_SlaveReceive(void)
{
    /* Wait for reception complete */
    while (!(SPSR & (1<<SPIF)))
        ;
    /* Return Data Register */
    return SPDR;
}
```

Note: 1. See "About Code Examples" on page 7.

18.3 \overline{SS} Pin Functionality

18.3.1 Slave Mode

When the SPI is configured as a Slave, the Slave Select (\overline{SS}) pin is always input. When \overline{SS} is held low, the SPI is activated, and MISO becomes an output if configured so by the user. All other pins are inputs. When \overline{SS} is driven high, all pins are inputs, and the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the \overline{SS} pin is driven high.

The \overline{SS} pin is useful for packet/byte synchronization to keep the slave bit counter synchronous with the master clock generator. When the \overline{SS} pin is driven high, the SPI slave will immediately reset the send and receive logic, and drop any partially received data in the Shift Register.

18.3.2 Master Mode

When the SPI is configured as a Master (MSTR in SPCR is set), the user can determine the direction of the \overline{SS} pin.

If \overline{SS} is configured as an output, the pin is a general output pin which does not affect the SPI system. Typically, the pin will be driving the \overline{SS} pin of the SPI Slave.

If \overline{SS} is configured as an input, it must be held high to ensure Master SPI operation. If the \overline{SS} pin is driven low by peripheral circuitry when the SPI is configured as a Master with the \overline{SS} pin defined as an input, the SPI system interprets this as another master selecting the SPI as a slave and starting to send data to it. To avoid bus contention, the SPI system takes the following actions:

1. The MSTR bit in SPCR is cleared and the SPI system becomes a Slave. As a result of the SPI becoming a Slave, the MOSI and SCK pins become inputs.
2. The SPIF Flag in SPSR is set, and if the SPI interrupt is enabled, and the I-bit in SREG is set, the interrupt routine will be executed.

Thus, when interrupt-driven SPI transmission is used in Master mode, and there exists a possibility that \overline{SS} is driven low, the interrupt should always check that the MSTR bit is still set. If the MSTR bit has been cleared by a slave select, it must be set by the user to re-enable SPI Master mode.

18.4 Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in [Figure 18-3](#) and [Figure 18-4](#) on page 172. Data bits are shifted out and latched in on opposite edges of the SCK signal, ensuring sufficient time for data signals to stabilize. This is clearly seen by summarizing [Table 18-3](#) on page 173 and [Table 18-4](#) on page 173, as done in [Table 18-2](#).

Table 18-2. SPI Modes

SPI Mode	Conditions	Leading Edge	Trailing eDge
0	CPOL=0, CPHA=0	Sample (Rising)	Setup (Falling)
1	CPOL=0, CPHA=1	Setup (Rising)	Sample (Falling)
2	CPOL=1, CPHA=0	Sample (Falling)	Setup (Rising)
3	CPOL=1, CPHA=1	Setup (Falling)	Sample (Rising)

Figure 18-3. SPI Transfer Format with CPHA = 0

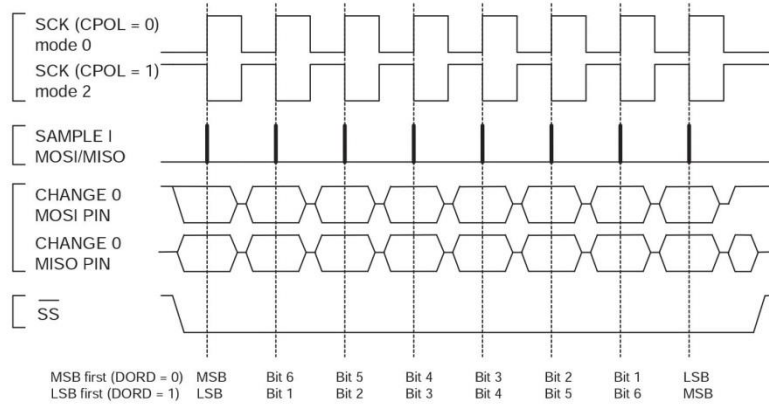
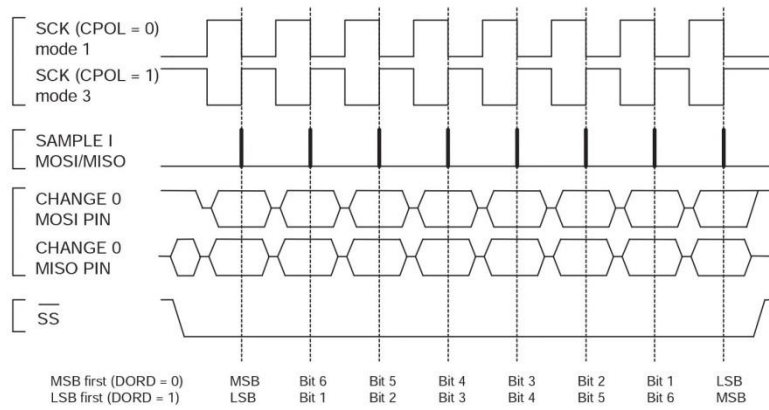


Figure 18-4. SPI Transfer Format with CPHA = 1



18.5 Register Description

18.5.1 SPCR – SPI Control Register

Bit	7	6	5	4	3	2	1	0	
0x2C (0x4C)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIE: SPI Interrupt Enable**

This bit causes the SPI interrupt to be executed if SPIF bit in the SPSR Register is set and the Global Interrupt Enable bit in SREG is set.

- **Bit 6 – SPE: SPI Enable**

When the SPE bit is written to one, the SPI is enabled. This bit must be set to enable any SPI operations.

- **Bit 5 – DORD: Data Order**

When the DORD bit is written to one, the LSB of the data word is transmitted first.

When the DORD bit is written to zero, the MSB of the data word is transmitted first.

- **Bit 4 – MSTR: Master/Slave Select**

This bit selects Master SPI mode when written to one, and Slave SPI mode when written logic zero. If \overline{SS} is configured as an input and is driven low while MSTR is set, MSTR will be cleared, and SPIF in SPSR will become set. The user will then have to set MSTR to re-enable SPI Master mode.

- **Bit 3 – CPOL: Clock Polarity**

When this bit is written to one, SCK is high when idle. When CPOL is written to zero, SCK is low when idle. Refer to [Figure 18-3](#) and [Figure 18-4](#) for an example. The CPOL functionality is summarized below:

Table 18-3. CPOL Functionality

CPOL	Leading Edge	Trailing Edge
0	Rising	Falling
1	Falling	Rising

- **Bit 2 – CPHA: Clock Phase**

The settings of the Clock Phase bit (CPHA) determine if data is sampled on the leading (first) or trailing (last) edge of SCK. Refer to [Figure 18-3](#) and [Figure 18-4](#) for an example. The CPOL functionality is summarized below:

Table 18-4. CPHA Functionality

CPHA	Leading Edge	Trailing Edge
0	Sample	Setup
1	Setup	Sample

- **Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a Master. SPR1 and SPR0 have no effect on the Slave. The relationship between SCK and the Oscillator Clock frequency f_{osc} is shown in the following table:

Table 18-5. Relationship Between SCK and the Oscillator Frequency

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

18.5.2 SPSR – SPI Status Register

Bit	7	6	5	4	3	2	1	0	
0x2D (0x4D)	SPSR								
	SPIF	WCOL	–	–	–	–	–	SPI2X	
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIF: SPI Interrupt Flag**

When a serial transfer is complete, the SPIF Flag is set. An interrupt is generated if SPIE in SPCR is set and global interrupts are enabled. If \overline{SS} is an input and is driven low when the SPI is in Master mode, this will also set the SPIF Flag. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit is cleared by first reading the SPI Status Register with SPIF set, then accessing the SPI Data Register (SPDR).

- **Bit 6 – WCOL: Write COLLision Flag**

The WCOL bit is set if the SPI Data Register (SPDR) is written during a data transfer. The WCOL bit (and the SPIF bit) are cleared by first reading the SPI Status Register with WCOL set, and then accessing the SPI Data Register.

- **Bit 5..1 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 0 – SPI2X: Double SPI Speed Bit**

When this bit is written logic one the SPI speed (SCK Frequency) will be doubled when the SPI is in Master mode (see Table 18-5). This means that the minimum SCK period will be two CPU clock periods. When the SPI is configured as Slave, the SPI is only guaranteed to work at $f_{osc}/4$ or lower.

The SPI interface on the ATmega48PA/88PA/168PA/328P is also used for program memory and EEPROM downloading or uploading. See page 308 for serial programming and verification.

18.5.3 SPDR – SPI Data Register

Bit	7	6	5	4	3	2	1	0	
0x2E (0x4E)	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	Undefined

The SPI Data Register is a read/write register used for data transfer between the Register File and the SPI Shift Register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.

19. USART0

19.1 Features

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous Operation
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Multi-processor Communication Mode
- Double Speed Asynchronous Communication Mode

19.2 Overview

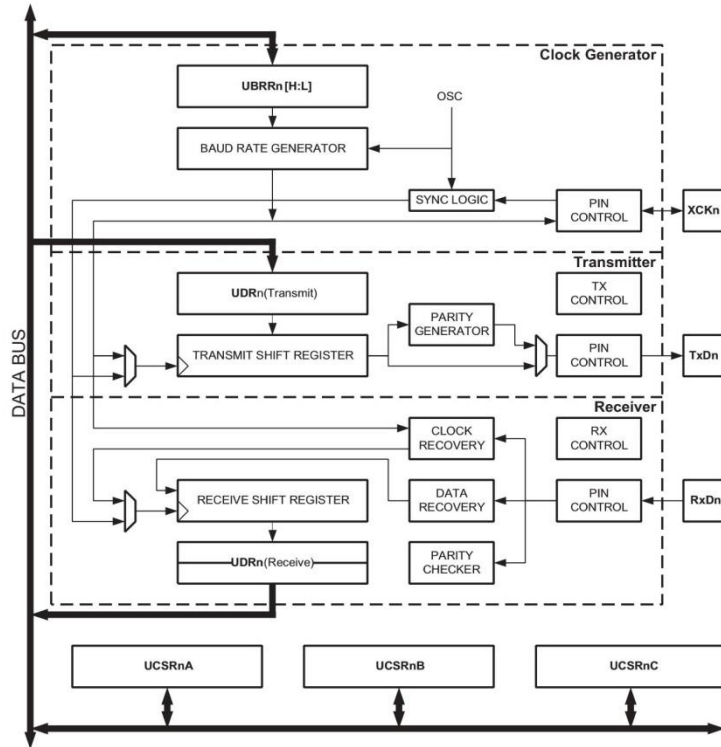
The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device.

The USART0 can also be used in Master SPI mode, see "USART in SPI Mode" on page 204. The Power Reduction USART bit, PRUSART0, in "[Minimizing Power Consumption](#)" on page 42 must be disabled by writing a logical zero to it.

A simplified block diagram of the USART Transmitter is shown in [Figure 19-1 on page 177](#). CPU accessible I/O Registers and I/O pins are shown in bold.

The dashed boxes in the block diagram separate the three main parts of the USART (listed from the top): Clock Generator, Transmitter and Receiver. Control Registers are shared by all units. The Clock Generation logic consists of synchronization logic for external clock input used by synchronous slave operation, and the baud rate generator. The XCKn (Transfer Clock) pin is only used by synchronous transfer mode. The Transmitter consists of a single write buffer, a serial Shift Register, Parity Generator and Control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The Receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery units are used for asynchronous data reception. In addition to the recovery units, the Receiver includes a Parity Checker, Control logic, a Shift Register and a two level receive buffer (UDRn). The Receiver supports the same frame formats as the Transmitter, and can detect Frame Error, Data OverRun and Parity Errors.

Figure 19-1. USART Block Diagram⁽¹⁾



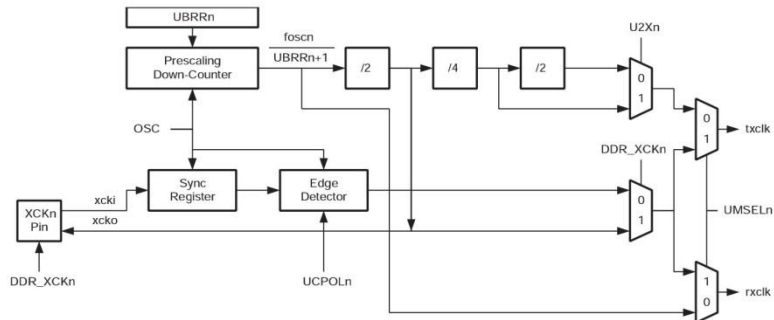
Note: 1. Refer to Figure 1-1 on page 2 and Table 13-9 on page 88 for USART0 pin placement.

19.3 Clock Generation

The Clock Generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation: Normal asynchronous, Double Speed asynchronous, Master synchronous and Slave synchronous mode. The UMSELn bit in USART Control and Status Register C (UCSRnC) selects between asynchronous and synchronous operation. Double Speed (asynchronous mode only) is controlled by the U2Xn found in the UCSRnA Register. When using synchronous mode (UMSELn = 1), the Data Direction Register for the XCKn pin (DDR_XCKn) controls whether the clock source is internal (Master mode) or external (Slave mode). The XCKn pin is only active when using synchronous mode.

Figure 19-2 shows a block diagram of the clock generation logic.

Figure 19-2. Clock Generation Logic, Block Diagram



Signal description:

- txclk** Transmitter clock (Internal Signal).
- rxclk** Receiver base clock (Internal Signal).
- xcki** Input from XCK pin (internal Signal). Used for synchronous slave operation.
- xcko** Clock output to XCK pin (Internal Signal). Used for synchronous master operation.
- fosc** XTAL pin frequency (System Clock).

19.3.1 Internal Clock Generation – The Baud Rate Generator

Internal clock generation is used for the asynchronous and the synchronous master modes of operation. The description in this section refers to [Figure 19-2](#).

The USART Baud Rate Register (UBRRn) and the down-counter connected to it function as a programmable prescaler or baud rate generator. The down-counter, running at system clock (f_{osc}), is loaded with the UBRRn value each time the counter has counted down to zero or when the UBRRnL Register is written. A clock is generated each time the counter reaches zero. This clock is the baud rate generator clock output ($= f_{osc}/(UBRRn+1)$). The Transmitter divides the baud rate generator clock output by 2, 8 or 16 depending on mode. The baud rate generator output is used directly by the Receiver's clock and data recovery units. However, the recovery units use a state machine that uses 2, 8 or 16 states depending on mode set by the state of the UMSELn, U2Xn and DDR_XCKn bits.

Table 19-1 contains equations for calculating the baud rate (in bits per second) and for calculating the UBRRn value for each mode of operation using an internally generated clock source.

Table 19-1. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate ⁽¹⁾	Equation for Calculating UBRRn Value
Asynchronous Normal mode (U2Xn = 0)	$BAUD = \frac{f_{OSC}}{16(UBRRn + 1)}$	$UBRRn = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed mode (U2Xn = 1)	$BAUD = \frac{f_{OSC}}{8(UBRRn + 1)}$	$UBRRn = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master mode	$BAUD = \frac{f_{OSC}}{2(UBRRn + 1)}$	$UBRRn = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps)

BAUD Baud rate (in bits per second, bps)

f_{osc} System Oscillator clock frequency

UBRRn Contents of the UBRRnH and UBRRnL Registers, (0-4095)

Some examples of UBRRn values for some system clock frequencies are found in Table 19-9 (see page 200).

19.3.2 Double Speed Operation (U2Xn)

The transfer rate can be doubled by setting the U2Xn bit in UCSRnA. Setting this bit only has effect for the asynchronous operation. Set this bit to zero when using synchronous operation.

Setting this bit will reduce the divisor of the baud rate divider from 16 to 8, effectively doubling the transfer rate for asynchronous communication. Note however that the Receiver will in this case only use half the number of samples (reduced from 16 to 8) for data sampling and clock recovery, and therefore a more accurate baud rate setting and system clock are required when this mode is used. For the Transmitter, there are no downsides.

19.3.3 External Clock

External clocking is used by the synchronous slave modes of operation. The description in this section refers to [Figure 19-2](#) for details.

External clock input from the XCKn pin is sampled by a synchronization register to minimize the chance of meta-stability. The output from the synchronization register must then pass through an edge detector before it can be used by the Transmitter and Receiver. This process introduces a two CPU clock period delay and therefore the maximum external XCKn clock frequency is limited by the following equation:

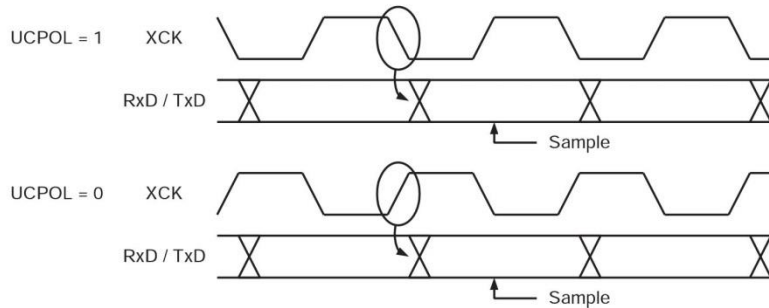
$$f_{XCK} < \frac{f_{OSC}}{4}$$

Note that f_{osc} depends on the stability of the system clock source. It is therefore recommended to add some margin to avoid possible loss of data due to frequency variations.

19.3.4 Synchronous Clock Operation

When synchronous mode is used ($UMSELn = 1$), the XCKn pin will be used as either clock input (Slave) or clock output (Master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input (on RxDn) is sampled at the opposite XCKn clock edge of the edge the data output (TxDn) is changed.

Figure 19-3. Synchronous Mode XCKn Timing.



The UC POLn bit UCRSC selects which XCKn clock edge is used for data sampling and which is used for data change. As [Figure 19-3](#) shows, when UC POLn is zero the data will be changed at rising XCKn edge and sampled at falling XCKn edge. If UC POLn is set, the data will be changed at falling XCKn edge and sampled at rising XCKn edge.

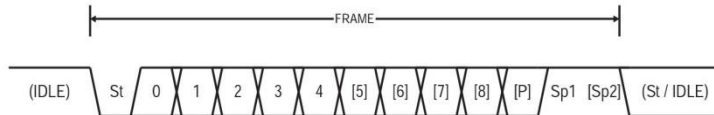
19.4 Frame Formats

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking. The USART accepts all 30 combinations of the following as valid frame formats:

- 1 start bit
- 5, 6, 7, 8, or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit. Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit. If enabled, the parity bit is inserted after the data bits, before the stop bits. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state. Figure 19-4 illustrates the possible combinations of the frame formats. Bits inside brackets are optional.

Figure 19-4. Frame Formats



- St** Start bit, always low.
- (n)** Data bits (0 to 8).
- P** Parity bit. Can be odd or even.
- Sp** Stop bit, always high.
- IDLE** No transfers on the communication line (Rx/Dn or Tx/Dn). An IDLE line must be high.

The frame format used by the USART is set by the UCSZn2:0, UPMn1:0 and USBSn bits in UCSRnB and UCSRnC. The Receiver and Transmitter use the same setting. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and Transmitter.

The USART Character SiZe (UCSZn2:0) bits select the number of data bits in the frame. The USART Parity mode (UPMn1:0) bits enable and set the type of parity bit. The selection between one or two stop bits is done by the USART Stop Bit Select (USBSn) bit. The Receiver ignores the second stop bit. An FE (Frame Error) will therefore only be detected in the cases where the first stop bit is zero.

19.4.1 Parity Bit Calculation

The parity bit is calculated by doing an exclusive-or of all the data bits. If odd parity is used, the result of the exclusive or is inverted. The relation between the parity bit and data bits is as follows:

$$\begin{aligned}
 P_{even} & d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0 \\
 P_{odd} & d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1
 \end{aligned}$$

- P_{even}** Parity bit using even parity
- P_{odd}** Parity bit using odd parity
- d_n** Data bit n of the character

If used, the parity bit is located between the last data bit and first stop bit of a serial frame.

19.5 USART Initialization

The USART has to be initialized before any communication can take place. The initialization process normally consists of setting the baud rate, setting frame format and enabling the Transmitter or the Receiver depending on the usage. For interrupt driven USART operation, the Global Interrupt Flag should be cleared (and interrupts globally disabled) when doing the initialization.

Before doing a re-initialization with changed baud rate or frame format, be sure that there are no ongoing transmissions during the period the registers are changed. The TXCn Flag can be used to check that the Transmitter has completed all transfers, and the RXC Flag can be used to check that there are no unread data in the receive buffer. Note that the TXCn Flag must be cleared before each transmission (before UDRn is written) if it is used for this purpose.

The following simple USART initialization code examples show one assembly and one C function that are equal in functionality. The examples assume asynchronous operation using polling (no interrupts enabled) and a fixed frame format. The baud rate is given as a function parameter. For the assembly code, the baud rate parameter is assumed to be stored in the r17:r16 Registers.

Assembly Code Example⁽¹⁾

```

USART_Init:
    ; Set baud rate
    out  UBRRnH, r17
    out  UBRRnL, r16
    ; Enable receiver and transmitter
    ldi  r16, (1<<RXENn)|(1<<TXENn)
    out  UCSRnB,r16
    ; Set frame format: 8data, 2stop bit
    ldi  r16, (1<<USBSn)|(3<<UCSZn0)
    out  UCSRnC,r16
    ret
    
```

C Code Example⁽¹⁾

```

#define FOSC 1843200 // Clock Speed
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
void main( void )
{
    ...
    USART_Init(MYUBRR)
    ...
}
void USART_Init( unsigned int ubrr)
{
    /*Set baud rate */
    UBRR0H = (unsigned char) (ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    /* Enable receiver and transmitter */
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    /* Set frame format: 8data, 2stop bit */
    UCSR0C = (1<<USBS0)|(3<<UCSZ00);
}
    
```

Note: 1. See "About Code Examples" on page 7.

More advanced initialization routines can be made that include frame format as parameters, disable interrupts and so on. However, many applications use a fixed setting of the baud and control registers, and for these types of applications the initialization code can be placed directly in the main routine, or be combined with initialization code for other I/O modules.

19.6 Data Transmission – The USART Transmitter

The USART Transmitter is enabled by setting the *Transmit Enable* (TXEN) bit in the UCSRnB Register. When the Transmitter is enabled, the normal port operation of the TxDn pin is overridden by the USART and given the function as the Transmitter's serial output. The baud rate, mode of operation and frame format must be set up once before doing any transmissions. If syn-

chronous operation is used, the clock on the XCKn pin will be overridden and used as transmission clock.

19.6.1 Sending Frames with 5 to 8 Data Bit

A data transmission is initiated by loading the transmit buffer with the data to be transmitted. The CPU can load the transmit buffer by writing to the UDRn I/O location. The buffered data in the transmit buffer will be moved to the Shift Register when the Shift Register is ready to send a new frame. The Shift Register is loaded with new data if it is in idle state (no ongoing transmission) or immediately after the last stop bit of the previous frame is transmitted. When the Shift Register is loaded with new data, it will transfer one complete frame at the rate given by the Baud Register, U2Xn bit or by XCKn depending on mode of operation.

The following code examples show a simple USART transmit function based on polling of the *Data Register Empty* (UDREN) Flag. When using frames with less than eight bits, the most significant bits written to the UDRn are ignored. The USART has to be initialized before the function can be used. For the assembly code, the data to be sent is assumed to be stored in Register R16

Assembly Code Example ⁽¹⁾
<pre>USART_Transmit: ; Wait for empty transmit buffer sbis UCSRnA,UDREN rjmp USART_Transmit ; Put data (r16) into buffer, sends the data out UDRn,r16 ret</pre>
C Code Example ⁽¹⁾
<pre>void USART_Transmit(unsigned char data) { /* Wait for empty transmit buffer */ while (!(UCSRnA & (1<<UDREN))) ; /* Put data into buffer, sends the data */ UDRn = data; }</pre>

Note: 1. See "About Code Examples" on page 7.

The function simply waits for the transmit buffer to be empty by checking the UDREN Flag, before loading it with new data to be transmitted. If the Data Register Empty interrupt is utilized, the interrupt routine writes the data into the buffer.

19.6.2 Sending Frames with 9 Data Bit

If 9-bit characters are used (UCSZn = 7), the ninth bit must be written to the TXB8 bit in UCSRnB before the low byte of the character is written to UDRn. The following code examples show a transmit function that handles 9-bit characters. For the assembly code, the data to be sent is assumed to be stored in registers R17:R16.

Assembly Code Example⁽¹⁾⁽²⁾

```

USART_Transmit:
    ; Wait for empty transmit buffer
    sbis UCSRnA,UDREN
    rjmp USART_Transmit
    ; Copy 9th bit from r17 to TXB8
    cbi UCSRnB,TXB8
    sbrc r17,0
    sbi UCSRnB,TXB8
    ; Put LSB data (r16) into buffer, sends the data
    out UDRn,r16
    ret
    
```

C Code Example⁽¹⁾⁽²⁾

```

void USART_Transmit( unsigned int data )
{
    /* Wait for empty transmit buffer */
    while ( !( UCSRnA & (1<<UDREN)) )
        ;
    /* Copy 9th bit to TXB8 */
    UCSRnB &= ~(1<<TXB8);
    if ( data & 0x0100 )
        UCSRnB |= (1<<TXB8);
    /* Put data into buffer, sends the data */
    UDRn = data;
}
    
```

- Notes: 1. These transmit functions are written to be general functions. They can be optimized if the contents of the UCSRnB is static. For example, only the TXB8 bit of the UCSRnB Register is used after initialization.
2. See ["About Code Examples" on page 7](#).

The ninth bit can be used for indicating an address frame when using multi processor communication mode or for other protocol handling as for example synchronization.

19.6.3 Transmitter Flags and Interrupts

The USART Transmitter has two flags that indicate its state: USART Data Register Empty (UDREN) and Transmit Complete (TXCn). Both flags can be used for generating interrupts.

The Data Register Empty (UDREN) Flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty, and cleared when the transmit buffer contains data to be transmitted that has not yet been moved into the Shift Register. For compatibility with future devices, always write this bit to zero when writing the UCSRnA Register.

When the Data Register Empty Interrupt Enable (UDRIEn) bit in UCSRnB is written to one, the USART Data Register Empty Interrupt will be executed as long as UDREN is set (provided that global interrupts are enabled). UDREN is cleared by writing UDRn. When interrupt-driven data transmission is used, the Data Register Empty interrupt routine must either write new data to

UDRn in order to clear UDREn or disable the Data Register Empty interrupt, otherwise a new interrupt will occur once the interrupt routine terminates.

The Transmit Complete (TXCn) Flag bit is set one when the entire frame in the Transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer. The TXCn Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXCn Flag is useful in half-duplex communication interfaces (like the RS-485 standard), where a transmitting application must enter receive mode and free the communication bus immediately after completing the transmission.

When the Transmit Complete Interrupt Enable (TXCIEn) bit in UCSRnB is set, the USART Transmit Complete Interrupt will be executed when the TXCn Flag becomes set (provided that global interrupts are enabled). When the transmit complete interrupt is used, the interrupt handling routine does not have to clear the TXCn Flag, this is done automatically when the interrupt is executed.

19.6.4 Parity Generator

The Parity Generator calculates the parity bit for the serial frame data. When parity bit is enabled (UPMn1 = 1), the transmitter control logic inserts the parity bit between the last data bit and the first stop bit of the frame that is sent.

19.6.5 Disabling the Transmitter

The disabling of the Transmitter (setting the TXEN to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the Transmit Shift Register and Transmit Buffer Register do not contain data to be transmitted. When disabled, the Transmitter will no longer override the TxDn pin.

19.7 Data Reception – The USART Receiver

The USART Receiver is enabled by writing the Receive Enable (RXENn) bit in the UCSRnB Register to one. When the Receiver is enabled, the normal pin operation of the RxDn pin is overridden by the USART and given the function as the Receiver's serial input. The baud rate, mode of operation and frame format must be set up once before any serial reception can be done. If synchronous operation is used, the clock on the XCKn pin will be used as transfer clock.

19.7.1 Receiving Frames with 5 to 8 Data Bits

The Receiver starts data reception when it detects a valid start bit. Each bit that follows the start bit will be sampled at the baud rate or XCKn clock, and shifted into the Receive Shift Register until the first stop bit of a frame is received. A second stop bit will be ignored by the Receiver. When the first stop bit is received, i.e., a complete serial frame is present in the Receive Shift Register, the contents of the Shift Register will be moved into the receive buffer. The receive buffer can then be read by reading the UDRn I/O location.

The following code example shows a simple USART receive function based on polling of the Receive Complete (RXCn) Flag. When using frames with less than eight bits the most significant bits of the data read from the UDRn will be masked to zero. The USART has to be initialized before the function can be used.

Assembly Code Example⁽¹⁾

```
USART_Receive:
    ; Wait for data to be received
    sbis UCSRnA, RXCn
    rjmp USART_Receive
    ; Get and return received data from buffer
    in r16, UDRn
    ret
```

C Code Example⁽¹⁾

```
unsigned char USART_Receive( void )
{
    /* Wait for data to be received */
    while ( !(UCSRnA & (1<<RXCn)) )
        ;
    /* Get and return received data from buffer */
    return UDRn;
}
```

Note: 1. See "About Code Examples" on page 7.
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

The function simply waits for data to be present in the receive buffer by checking the RXCn Flag, before reading the buffer and returning the value.

19.7.2 Receiving Frames with 9 Data Bits

If 9-bit characters are used (UCSZn=7) the ninth bit must be read from the RXB8n bit in UCSRnB **before** reading the low bits from the UDRn. This rule applies to the FEn, DORn and UPEn Status Flags as well. Read status from UCSRnA, then data from UDRn. Reading the UDRn I/O location will change the state of the receive buffer FIFO and consequently the TXB8n, FEn, DORn and UPEn bits, which all are stored in the FIFO, will change.

The following code example shows a simple USART receive function that handles both nine bit characters and the status bits.

Assembly Code Example⁽¹⁾

```

USART_Receive:
    ; Wait for data to be received
    sbis UCSRnA, RXCn
    rjmp USART_Receive
    ; Get status and 9th bit, then data from buffer
    in  r18, UCSRnA
    in  r17, UCSRnB
    in  r16, UDRn
    ; If error, return -1
    andi r18, (1<<FEn) | (1<<DORn) | (1<<UPEn)
    breq USART_ReceiveNoError
    ldi  r17, HIGH(-1)
    ldi  r16, LOW(-1)
USART_ReceiveNoError:
    ; Filter the 9th bit, then return
    lsr  r17
    andi r17, 0x01
    ret
    
```

C Code Example⁽¹⁾

```

unsigned int USART_Receive( void )
{
    unsigned char status, resh, resl;
    /* Wait for data to be received */
    while ( !(UCSRnA & (1<<RXCn)) )
        ;
    /* Get status and 9th bit, then data */
    /* from buffer */
    status = UCSRnA;
    resh = UCSRnB;
    resl = UDRn;
    /* If error, return -1 */
    if ( status & (1<<FEn) | (1<<DORn) | (1<<UPEn) )
        return -1;
    /* Filter the 9th bit, then return */
    resh = (resh >> 1) & 0x01;
    return ((resh << 8) | resl);
}
    
```

Note: 1. See "About Code Examples" on page 7.
 For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBR", "SBRC", "SBR", and "CBR".

The receive function example reads all the I/O Registers into the Register File before any computation is done. This gives an optimal receive buffer utilization since the buffer location read will be free to accept new data as early as possible.

19.7.3 Receive Complete Flag and Interrupt

The USART Receiver has one flag that indicates the Receiver state.

The Receive Complete (RXCn) Flag indicates if there are unread data present in the receive buffer. This flag is one when unread data exist in the receive buffer, and zero when the receive buffer is empty (i.e., does not contain any unread data). If the Receiver is disabled (RXENn = 0), the receive buffer will be flushed and consequently the RXCn bit will become zero.

When the Receive Complete Interrupt Enable (RXCIEn) in UCSRnB is set, the USART Receive Complete interrupt will be executed as long as the RXCn Flag is set (provided that global interrupts are enabled). When interrupt-driven data reception is used, the receive complete routine must read the received data from UDRn in order to clear the RXCn Flag, otherwise a new interrupt will occur once the interrupt routine terminates.

19.7.4 Receiver Error Flags

The USART Receiver has three Error Flags: Frame Error (FEn), Data OverRun (DORn) and Parity Error (UPEn). All can be accessed by reading UCSRnA. Common for the Error Flags is that they are located in the receive buffer together with the frame for which they indicate the error status. Due to the buffering of the Error Flags, the UCSRnA must be read before the receive buffer (UDRn), since reading the UDRn I/O location changes the buffer read location. Another equality for the Error Flags is that they can not be altered by software doing a write to the flag location. However, all flags must be set to zero when the UCSRnA is written for upward compatibility of future USART implementations. None of the Error Flags can generate interrupts.

The Frame Error (FEn) Flag indicates the state of the first stop bit of the next readable frame stored in the receive buffer. The FEn Flag is zero when the stop bit was correctly read (as one), and the FEn Flag will be one when the stop bit was incorrect (zero). This flag can be used for detecting out-of-sync conditions, detecting break conditions and protocol handling. The FEn Flag is not affected by the setting of the USBSn bit in UCSRnC since the Receiver ignores all, except for the first, stop bits. For compatibility with future devices, always set this bit to zero when writing to UCSRnA.

The Data OverRun (DORn) Flag indicates data loss due to a receiver buffer full condition. A Data OverRun occurs when the receive buffer is full (two characters), it is a new character waiting in the Receive Shift Register, and a new start bit is detected. If the DORn Flag is set there was one or more serial frame lost between the frame last read from UDRn, and the next frame read from UDRn. For compatibility with future devices, always write this bit to zero when writing to UCSRnA. The DORn Flag is cleared when the frame received was successfully moved from the Shift Register to the receive buffer.

The Parity Error (UPEn) Flag indicates that the next frame in the receive buffer had a Parity Error when received. If Parity Check is not enabled the UPEn bit will always be read zero. For compatibility with future devices, always set this bit to zero when writing to UCSRnA. For more details see "Parity Bit Calculation" on page 181 and "Parity Checker" on page 189.

19.7.5 Parity Checker

The Parity Checker is active when the high USART Parity mode (UPMn1) bit is set. Type of Parity Check to be performed (odd or even) is selected by the UPMn0 bit. When enabled, the Parity Checker calculates the parity of the data bits in incoming frames and compares the result with the parity bit from the serial frame. The result of the check is stored in the receive buffer together with the received data and stop bits. The Parity Error (UPEn) Flag can then be read by software to check if the frame had a Parity Error.

The UPE_n bit is set if the next character that can be read from the receive buffer had a Parity Error when received and the Parity Checking was enabled at that point (UPM_n1 = 1). This bit is valid until the receive buffer (UDR_n) is read.

19.7.6 Disabling the Receiver

In contrast to the Transmitter, disabling of the Receiver will be immediate. Data from ongoing receptions will therefore be lost. When disabled (i.e., the RXEN_n is set to zero) the Receiver will no longer override the normal function of the Rx_{Dn} port pin. The Receiver buffer FIFO will be flushed when the Receiver is disabled. Remaining data in the buffer will be lost

19.7.7 Flushing the Receive Buffer

The receiver buffer FIFO will be flushed when the Receiver is disabled, i.e., the buffer will be emptied of its contents. Unread data will be lost. If the buffer has to be flushed during normal operation, due to for instance an error condition, read the UDR_n I/O location until the RXC_n Flag is cleared. The following code example shows how to flush the receive buffer.

Assembly Code Example ⁽¹⁾
<pre>USART_Flush: sbis UCSRnA, RXCn ret in r16, UDRn rjmp USART_Flush</pre>
C Code Example ⁽¹⁾
<pre>void USART_Flush(void) { unsigned char dummy; while (UCSRnA & (1<<RXCn)) dummy = UDRn; }</pre>

Note: 1. See "About Code Examples" on page 7.
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

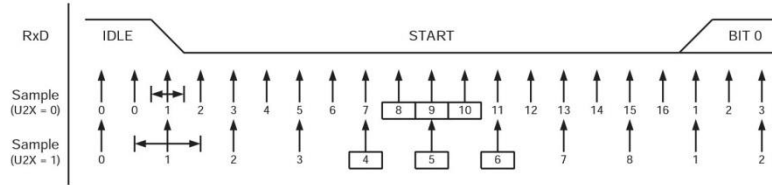
19.8 Asynchronous Data Reception

The USART includes a clock recovery and a data recovery unit for handling asynchronous data reception. The clock recovery logic is used for synchronizing the internally generated baud rate clock to the incoming asynchronous serial frames at the Rx_{Dn} pin. The data recovery logic samples and low pass filters each incoming bit, thereby improving the noise immunity of the Receiver. The asynchronous reception operational range depends on the accuracy of the internal baud rate clock, the rate of the incoming frames, and the frame size in number of bits.

19.8.1 Asynchronous Clock Recovery

The clock recovery logic synchronizes internal clock to the incoming serial frames. Figure 19-5 illustrates the sampling process of the start bit of an incoming frame. The sample rate is 16 times the baud rate for Normal mode, and eight times the baud rate for Double Speed mode. The horizontal arrows illustrate the synchronization variation due to the sampling process. Note the larger time variation when using the Double Speed mode (U2X_n = 1) of operation. Samples denoted zero are samples done when the Rx_{Dn} line is idle (i.e., no communication activity).

Figure 19-5. Start Bit Sampling

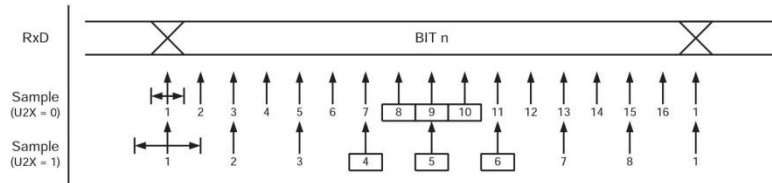


When the clock recovery logic detects a high (idle) to low (start) transition on the RxDn line, the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample as shown in the figure. The clock recovery logic then uses samples 8, 9, and 10 for Normal mode, and samples 4, 5, and 6 for Double Speed mode (indicated with sample numbers inside boxes on the figure), to decide if a valid start bit is received. If two or more of these three samples have logical high levels (the majority wins), the start bit is rejected as a noise spike and the Receiver starts looking for the next high to low-transition. If however, a valid start bit is detected, the clock recovery logic is synchronized and the data recovery can begin. The synchronization process is repeated for each start bit.

19.8.2 Asynchronous Data Recovery

When the receiver clock is synchronized to the start bit, the data recovery can begin. The data recovery unit uses a state machine that has 16 states for each bit in Normal mode and eight states for each bit in Double Speed mode. Figure 19-6 shows the sampling of the data bits and the parity bit. Each of the samples is given a number that is equal to the state of the recovery unit.

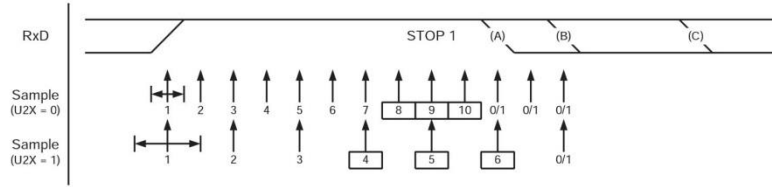
Figure 19-6. Sampling of Data and Parity Bit



The decision of the logic level of the received bit is taken by doing a majority voting of the logic value to the three samples in the center of the received bit. The center samples are emphasized on the figure by having the sample number inside boxes. The majority voting process is done as follows: If two or all three samples have high levels, the received bit is registered to be a logic 1. If two or all three samples have low levels, the received bit is registered to be a logic 0. This majority voting process acts as a low pass filter for the incoming signal on the RxDn pin. The recovery process is then repeated until a complete frame is received. Including the first stop bit. Note that the Receiver only uses the first stop bit of a frame.

Figure 19-7 on page 192 shows the sampling of the stop bit and the earliest possible beginning of the start bit of the next frame.

Figure 19-7. Stop Bit Sampling and Next Start Bit Sampling



The same majority voting is done to the stop bit as done for the other bits in the frame. If the stop bit is registered to have a logic 0 value, the Frame Error (FEn) Flag will be set.

A new high to low transition indicating the start bit of a new frame can come right after the last of the bits used for majority voting. For Normal Speed mode, the first low level sample can be at point marked (A) in Figure 19-7. For Double Speed mode the first low level must be delayed to (B). (C) marks a stop bit of full length. The early start bit detection influences the operational range of the Receiver.

19.8.3 Asynchronous Operational Range

The operational range of the Receiver is dependent on the mismatch between the received bit rate and the internally generated baud rate. If the Transmitter is sending frames at too fast or too slow bit rates, or the internally generated baud rate of the Receiver does not have a similar (see Table 19-2 on page 193) base frequency, the Receiver will not be able to synchronize the frames to the start bit.

The following equations can be used to calculate the ratio of the incoming data rate and internal receiver baud rate.

$$R_{slow} = \frac{(D+1)S}{S-1+D \cdot S_F} \quad R_{fast} = \frac{(D+2)S}{(D+1)S+S_M}$$

- D** Sum of character size and parity size (D = 5 to 10 bit)
- S** Samples per bit. S = 16 for Normal Speed mode and S = 8 for Double Speed mode.
- S_F** First sample number used for majority voting. S_F = 8 for normal speed and S_F = 4 for Double Speed mode.
- S_M** Middle sample number used for majority voting. S_M = 9 for normal speed and S_M = 5 for Double Speed mode.
- R_{slow}** is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate. R_{fast} is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate.

Table 19-2 on page 193 and Table 19-3 on page 193 list the maximum receiver baud rate error that can be tolerated. Note that Normal Speed mode has higher toleration of baud rate variations.

Table 19-2. Recommended Maximum Receiver Baud Rate Error for Normal Speed Mode (U2Xn = 0)

D # (Data+Parity Bit)	R _{slow} (%)	R _{fast} (%)	Max Total Error (%)	Recommended Max Receiver Error (%)
5	93.20	106.67	+6.67/-6.8	± 3.0
6	94.12	105.79	+5.79/-5.88	± 2.5
7	94.81	105.11	+5.11/-5.19	± 2.0
8	95.36	104.58	+4.58/-4.54	± 2.0
9	95.81	104.14	+4.14/-4.19	± 1.5
10	96.17	103.78	+3.78/-3.83	± 1.5

Table 19-3. Recommended Maximum Receiver Baud Rate Error for Double Speed Mode (U2Xn = 1)

D # (Data+Parity Bit)	R _{slow} (%)	R _{fast} (%)	Max Total Error (%)	Recommended Max Receiver Error (%)
5	94.12	105.66	+5.66/-5.88	± 2.5
6	94.92	104.92	+4.92/-5.08	± 2.0
7	95.52	104.35	+4.35/-4.48	± 1.5
8	96.00	103.90	+3.90/-4.00	± 1.5
9	96.39	103.53	+3.53/-3.61	± 1.5
10	96.70	103.23	+3.23/-3.30	± 1.0

The recommendations of the maximum receiver baud rate error was made under the assumption that the Receiver and Transmitter equally divides the maximum total error.

There are two possible sources for the receivers baud rate error. The Receiver's system clock (XTAL) will always have some minor instability over the supply voltage range and the temperature range. When using a crystal to generate the system clock, this is rarely a problem, but for a resonator the system clock may differ more than 2% depending of the resonators tolerance. The second source for the error is more controllable. The baud rate generator can not always do an exact division of the system frequency to get the baud rate wanted. In this case an UBRRn value that gives an acceptable low error can be used if possible.

19.9 Multi-processor Communication Mode

Setting the Multi-processor Communication mode (MPCMn) bit in UCSRnA enables a filtering function of incoming frames received by the USART Receiver. Frames that do not contain address information will be ignored and not put into the receive buffer. This effectively reduces the number of incoming frames that has to be handled by the CPU, in a system with multiple MCUs that communicate via the same serial bus. The Transmitter is unaffected by the MPCMn setting, but has to be used differently when it is a part of a system utilizing the Multi-processor Communication mode.

If the Receiver is set up to receive frames that contain 5 to 8 data bits, then the first stop bit indicates if the frame contains data or address information. If the Receiver is set up for frames with

nine data bits, then the ninth bit (RXB8n) is used for identifying address and data frames. When the frame type bit (the first stop or the ninth bit) is one, the frame contains an address. When the frame type bit is zero the frame is a data frame.

The Multi-processor Communication mode enables several slave MCUs to receive data from a master MCU. This is done by first decoding an address frame to find out which MCU has been addressed. If a particular slave MCU has been addressed, it will receive the following data frames as normal, while the other slave MCUs will ignore the received frames until another address frame is received.

19.9.1 Using MPCMn

For an MCU to act as a master MCU, it can use a 9-bit character frame format (UCSZn = 7). The ninth bit (TXB8n) must be set when an address frame (TXB8n = 1) or cleared when a data frame (TXB = 0) is being transmitted. The slave MCUs must in this case be set to use a 9-bit character frame format.

The following procedure should be used to exchange data in Multi-processor Communication mode:

1. All Slave MCUs are in Multi-processor Communication mode (MPCMn in UCSRnA is set).
2. The Master MCU sends an address frame, and all slaves receive and read this frame. In the Slave MCUs, the RXCn Flag in UCSRnA will be set as normal.
3. Each Slave MCU reads the UDRn Register and determines if it has been selected. If so, it clears the MPCMn bit in UCSRnA, otherwise it waits for the next address byte and keeps the MPCMn setting.
4. The addressed MCU will receive all data frames until a new address frame is received. The other Slave MCUs, which still have the MPCMn bit set, will ignore the data frames.
5. When the last data frame is received by the addressed MCU, the addressed MCU sets the MPCMn bit and waits for a new address frame from master. The process then repeats from 2.

Using any of the 5- to 8-bit character frame formats is possible, but impractical since the Receiver must change between using n and n+1 character frame formats. This makes full-duplex operation difficult since the Transmitter and Receiver uses the same character size setting. If 5- to 8-bit character frames are used, the Transmitter must be set to use two stop bit (USBSn = 1) since the first stop bit is used for indicating the frame type.

Do not use Read-Modify-Write instructions (SBI and CBI) to set or clear the MPCMn bit. The MPCMn bit shares the same I/O location as the TXCn Flag and this might accidentally be cleared when using SBI or CBI instructions.

19.10 Register Description

19.10.1 UDRn – USART I/O Data Register n

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDRn (Read)
	TXB[7:0]								UDRn (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The USART Transmit Data Buffer Register and USART Receive Data Buffer Registers share the same I/O address referred to as USART Data Register or UDRn. The Transmit Data Buffer Register (TXB) will be the destination for data written to the UDRn Register location. Reading the UDRn Register location will return the contents of the Receive Data Buffer Register (RXB).

For 5-, 6-, or 7-bit characters the upper unused bits will be ignored by the Transmitter and set to zero by the Receiver.

The transmit buffer can only be written when the UDREN Flag in the UCSRnA Register is set. Data written to UDRn when the UDREN Flag is not set, will be ignored by the USART Transmitter. When data is written to the transmit buffer, and the Transmitter is enabled, the Transmitter will load the data into the Transmit Shift Register when the Shift Register is empty. Then the data will be serially transmitted on the TxDn pin.

The receive buffer consists of a two level FIFO. The FIFO will change its state whenever the receive buffer is accessed. Due to this behavior of the receive buffer, do not use Read-Modify-Write instructions (SBI and CBI) on this location. Be careful when using bit test instructions (SBIC and SBIS), since these also will change the state of the FIFO.

19.10.2 UCSRnA – USART Control and Status Register n A

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREN	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – RXCn: USART Receive Complete**

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the Receiver is disabled, the receive buffer will be flushed and consequently the RXCn bit will become zero. The RXCn Flag can be used to generate a Receive Complete interrupt (see description of the RXCIEn bit).

- **Bit 6 – TXCn: USART Transmit Complete**

This flag bit is set when the entire frame in the Transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer (UDRn). The TXCn Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXCn Flag can generate a Transmit Complete interrupt (see description of the TXCIEn bit).

- **Bit 5 – UDREN: USART Data Register Empty**

The UDREN Flag indicates if the transmit buffer (UDRn) is ready to receive new data. If UDREN is one, the buffer is empty, and therefore ready to be written. The UDREN Flag can generate a

ATmega48PA/88PA/168PA/328P

Data Register Empty interrupt (see description of the UDRIEn bit). UDREn is set after a reset to indicate that the Transmitter is ready.

- **Bit 4 – FEn: Frame Error**

This bit is set if the next character in the receive buffer had a Frame Error when received. I.e., when the first stop bit of the next character in the receive buffer is zero. This bit is valid until the receive buffer (UDRn) is read. The FEn bit is zero when the stop bit of received data is one. Always set this bit to zero when writing to UCSRnA.

- **Bit 3 – DORn: Data OverRun**

This bit is set if a Data OverRun condition is detected. A Data OverRun occurs when the receive buffer is full (two characters), it is a new character waiting in the Receive Shift Register, and a new start bit is detected. This bit is valid until the receive buffer (UDRn) is read. Always set this bit to zero when writing to UCSRnA.

- **Bit 2 – UPEn: USART Parity Error**

This bit is set if the next character in the receive buffer had a Parity Error when received and the Parity Checking was enabled at that point (UPMn1 = 1). This bit is valid until the receive buffer (UDRn) is read. Always set this bit to zero when writing to UCSRnA.

- **Bit 1 – U2Xn: Double the USART Transmission Speed**

This bit only has effect for the asynchronous operation. Write this bit to zero when using synchronous operation.

Writing this bit to one will reduce the divisor of the baud rate divider from 16 to 8 effectively doubling the transfer rate for asynchronous communication.

- **Bit 0 – MPCMn: Multi-processor Communication Mode**

This bit enables the Multi-processor Communication mode. When the MPCMn bit is written to one, all the incoming frames received by the USART Receiver that do not contain address information will be ignored. The Transmitter is unaffected by the MPCMn setting. For more detailed information see ["Multi-processor Communication Mode" on page 193](#).

19.10.3 UCSRnB – USART Control and Status Register n B

Bit	7	6	5	4	3	2	1	0	
	RXCIE _n	TXCIE _n	UDRIE _n	RXEN _n	TXEN _n	UCSZn ₂	RXB8 _n	TXB8 _n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RXCIE_n: RX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the RXC_n Flag. A USART Receive Complete interrupt will be generated only if the RXCIE_n bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXC_n bit in UCSRnA is set.

- **Bit 6 – TXCIE_n: TX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the TXC_n Flag. A USART Transmit Complete interrupt will be generated only if the TXCIE_n bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXC_n bit in UCSRnA is set.

- **Bit 5 – UDRIEn: USART Data Register Empty Interrupt Enable n**

Writing this bit to one enables interrupt on the UDREn Flag. A Data Register Empty interrupt will be generated only if the UDRIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the UDREn bit in UCSRnA is set.

- **Bit 4 – RXENn: Receiver Enable n**

Writing this bit to one enables the USART Receiver. The Receiver will override normal port operation for the RxDn pin when enabled. Disabling the Receiver will flush the receive buffer invalidating the FEn, DORn, and UPEn Flags.

- **Bit 3 – TXENn: Transmitter Enable n**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxDn pin when enabled. The disabling of the Transmitter (writing TXENn to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the Transmit Shift Register and Transmit Buffer Register do not contain data to be transmitted. When disabled, the Transmitter will no longer override the TxDn port.

- **Bit 2 – UCSZn2: Character Size n**

The UCSZn2 bits combined with the UCSZn1:0 bit in UCSRnC sets the number of data bits (Character SiZe) in a frame the Receiver and Transmitter use.

- **Bit 1 – RXB8n: Receive Data Bit 8 n**

RXB8n is the ninth data bit of the received character when operating with serial frames with nine data bits. Must be read before reading the low bits from UDRn.

- **Bit 0 – TXB8n: Transmit Data Bit 8 n**

TXB8n is the ninth data bit in the character to be transmitted when operating with serial frames with nine data bits. Must be written before writing the low bits to UDRn.

19.10.4 UCSRnC – USART Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bits 7:6 – UMSELn1:0 USART Mode Select**

These bits select the mode of operation of the USARTn as shown in [Table 19-4](#).

Table 19-4. UMSELn Bits Settings

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM) ⁽¹⁾

Note: 1. See "USART in SPI Mode" on page 204 for full description of the Master SPI Mode (MSPIM) operation

• **Bits 5:4 – UPMn1:0: Parity Mode**

These bits enable and set type of parity generation and check. If enabled, the Transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The Receiver will generate a parity value for the incoming data and compare it to the UPMn setting. If a mismatch is detected, the UPEn Flag in UCSRnA will be set.

Table 19-5. UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

• **Bit 3 – USBSn: Stop Bit Select**

This bit selects the number of stop bits to be inserted by the Transmitter. The Receiver ignores this setting.

Table 19-6. USBS Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bit

• **Bit 2:1 – UCSZn1:0: Character Size**

The UCSZn1:0 bits combined with the UCSZn2 bit in UCSRnB sets the number of data bits (Character SiZe) in a frame the Receiver and Transmitter use.

Table 19-7. UCSZn Bits Settings

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

• **Bit 0 – UCPOLn: Clock Polarity**

This bit is used for synchronous mode only. Write this bit to zero when asynchronous mode is used. The UCPOLn bit sets the relationship between data output change and data input sample, and the synchronous clock (XCKn).

Table 19-8. UCPOLn Bit Settings

UCPOLn	Transmitted Data Changed (Output of TxDn Pin)	Received Data Sampled (Input on RxDn Pin)
0	Rising XCKn Edge	Falling XCKn Edge
1	Falling XCKn Edge	Rising XCKn Edge

19.10.5 UBRRnL and UBRRnH – USART Baud Rate Registers

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	UBRRn[11:8]				UBRRnH
	UBRRn[7:0]								UBRRnL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

- **Bit 15:12 – Reserved Bits**

These bits are reserved for future use. For compatibility with future devices, these bit must be written to zero when UBRRnH is written.

- **Bit 11:0 – UBRR11:0: USART Baud Rate Register**

This is a 12-bit register which contains the USART baud rate. The UBRRnH contains the four most significant bits, and the UBRRnL contains the eight least significant bits of the USART baud rate. Ongoing transmissions by the Transmitter and Receiver will be corrupted if the baud rate is changed. Writing UBRRnL will trigger an immediate update of the baud rate prescaler.

19.11 Examples of Baud Rate Setting

For standard crystal and resonator frequencies, the most commonly used baud rates for asynchronous operation can be generated by using the UBRRn settings in [Table 19-9](#). UBRRn values which yield an actual baud rate differing less than 0.5% from the target baud rate, are bold in the table. Higher error ratings are acceptable, but the Receiver will have less noise resistance when the error ratings are high, especially for large serial frames (see "[Asynchronous Operational Range](#)" on page 192). The error values are calculated using the following equation:

$$\text{Error}[\%] = \left(\frac{\text{BaudRate}_{\text{Closest Match}} - 1}{\text{BaudRate}} \right) \cdot 100\%$$

Table 19-9. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies

Baud Rate (bps)	$f_{osc} = 1.0000 \text{ MHz}$				$f_{osc} = 1.8432 \text{ MHz}$				$f_{osc} = 2.0000 \text{ MHz}$			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	25	0.2%	51	0.2%	47	0.0%	95	0.0%	51	0.2%	103	0.2%
4800	12	0.2%	25	0.2%	23	0.0%	47	0.0%	25	0.2%	51	0.2%
9600	6	-7.0%	12	0.2%	11	0.0%	23	0.0%	12	0.2%	25	0.2%
14.4k	3	8.5%	8	-3.5%	7	0.0%	15	0.0%	8	-3.5%	16	2.1%
19.2k	2	8.5%	6	-7.0%	5	0.0%	11	0.0%	6	-7.0%	12	0.2%
28.8k	1	8.5%	3	8.5%	3	0.0%	7	0.0%	3	8.5%	8	-3.5%
38.4k	1	-18.6%	2	8.5%	2	0.0%	5	0.0%	2	8.5%	6	-7.0%
57.6k	0	8.5%	1	8.5%	1	0.0%	3	0.0%	1	8.5%	3	8.5%
76.8k	-	-	1	-18.6%	1	-25.0%	2	0.0%	1	-18.6%	2	8.5%
115.2k	-	-	0	8.5%	0	0.0%	1	0.0%	0	8.5%	1	8.5%
230.4k	-	-	-	-	-	-	0	0.0%	-	-	-	-
250k	-	-	-	-	-	-	-	-	-	-	0	0.0%
Max. ⁽¹⁾	62.5 kbps		125 kbps		115.2 kbps		230.4 kbps		125 kbps		250 kbps	

Note: 1. UBRRn = 0, Error = 0.0%

Table 19-10. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies (Continued)

Baud Rate (bps)	$f_{osc} = 3.6864 \text{ MHz}$				$f_{osc} = 4.0000 \text{ MHz}$				$f_{osc} = 7.3728 \text{ MHz}$			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	95	0.0%	191	0.0%	103	0.2%	207	0.2%	191	0.0%	383	0.0%
4800	47	0.0%	95	0.0%	51	0.2%	103	0.2%	95	0.0%	191	0.0%
9600	23	0.0%	47	0.0%	25	0.2%	51	0.2%	47	0.0%	95	0.0%
14.4k	15	0.0%	31	0.0%	16	2.1%	34	-0.8%	31	0.0%	63	0.0%
19.2k	11	0.0%	23	0.0%	12	0.2%	25	0.2%	23	0.0%	47	0.0%
28.8k	7	0.0%	15	0.0%	8	-3.5%	16	2.1%	15	0.0%	31	0.0%
38.4k	5	0.0%	11	0.0%	6	-7.0%	12	0.2%	11	0.0%	23	0.0%
57.6k	3	0.0%	7	0.0%	3	8.5%	8	-3.5%	7	0.0%	15	0.0%
76.8k	2	0.0%	5	0.0%	2	8.5%	6	-7.0%	5	0.0%	11	0.0%
115.2k	1	0.0%	3	0.0%	1	8.5%	3	8.5%	3	0.0%	7	0.0%
230.4k	0	0.0%	1	0.0%	0	8.5%	1	8.5%	1	0.0%	3	0.0%
250k	0	-7.8%	1	-7.8%	0	0.0%	1	0.0%	1	-7.8%	3	-7.8%
0.5M	-	-	0	-7.8%	-	-	0	0.0%	0	-7.8%	1	-7.8%
1M	-	-	-	-	-	-	-	-	-	-	0	-7.8%
Max. ⁽¹⁾	230.4 kbps		460.8 kbps		250 kbps		0.5 Mbps		460.8 kbps		921.6 kbps	

1. UBRRn = 0, Error = 0.0%

Table 19-11. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies (Continued)

Baud Rate (bps)	$f_{osc} = 8.0000 \text{ MHz}$				$f_{osc} = 11.0592 \text{ MHz}$				$f_{osc} = 14.7456 \text{ MHz}$			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	207	0.2%	416	-0.1%	287	0.0%	575	0.0%	383	0.0%	767	0.0%
4800	103	0.2%	207	0.2%	143	0.0%	287	0.0%	191	0.0%	383	0.0%
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4k	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2k	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8k	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4k	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6k	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8k	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2k	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4k	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250k	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	0	0.0%	1	0.0%	–	–	2	-7.8%	1	-7.8%	3	-7.8%
1M	–	–	0	0.0%	–	–	–	–	0	-7.8%	1	-7.8%
Max. ⁽¹⁾	0.5 Mbps		1 Mbps		691.2 kbps		1.3824 Mbps		921.6 kbps		1.8432 Mbps	

1. UBRRn = 0, Error = 0.0%

Table 19-12. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies (Continued)

Baud Rate (bps)	$f_{osc} = 16.0000$ MHz				$f_{osc} = 18.4320$ MHz				$f_{osc} = 20.0000$ MHz			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%	4	8.5%	10	-1.4%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%	4	0.0%	9	0.0%
0.5M	1	0.0%	3	0.0%	–	–	4	-7.8%	–	–	4	0.0%
1M	0	0.0%	1	0.0%	–	–	–	–	–	–	–	–
Max. ⁽¹⁾	1 Mbps		2 Mbps		1.152 Mbps		2.304 Mbps		1.25 Mbps		2.5 Mbps	

1. UBRRn = 0, Error = 0.0%

20. USART in SPI Mode

20.1 Features

- Full Duplex, Three-wire Synchronous Data Transfer
- Master Operation
- Supports all four SPI Modes of Operation (Mode 0, 1, 2, and 3)
- LSB First or MSB First Data Transfer (Configurable Data Order)
- Queued Operation (Double Buffered)
- High Resolution Baud Rate Generator
- High Speed Operation ($f_{XCKmax} = f_{CK}/2$)
- Flexible Interrupt Generation

20.2 Overview

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) can be set to a master SPI compliant mode of operation.

Setting both UMSELn1:0 bits to one enables the USART in MSPIM logic. In this mode of operation the SPI master control logic takes direct control over the USART resources. These resources include the transmitter and receiver shift register and buffers, and the baud rate generator. The parity generator and checker, the data and clock recovery logic, and the RX and TX control logic is disabled. The USART RX and TX control logic is replaced by a common SPI transfer control logic. However, the pin control logic and interrupt generation logic is identical in both modes of operation.

The I/O register locations are the same in both modes. However, some of the functionality of the control registers changes when using MSPIM.

20.3 Clock Generation

The Clock Generation logic generates the base clock for the Transmitter and Receiver. For USART MSPIM mode of operation only internal clock generation (i.e. master operation) is supported. The Data Direction Register for the XCKn pin (DDR_XCKn) must therefore be set to one (i.e. as output) for the USART in MSPIM to operate correctly. Preferably the DDR_XCKn should be set up before the USART in MSPIM is enabled (i.e. TXENn and RXENn bit set to one).

The internal clock generation used in MSPIM mode is identical to the USART synchronous master mode. The baud rate or UBRRn setting can therefore be calculated using the same equations, see [Table 20-1](#):

Table 20-1. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate ⁽¹⁾	Equation for Calculating UBRRn Value
Synchronous Master mode	$BAUD = \frac{f_{OSC}}{2(UBRRn + 1)}$	$UBRRn = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps)

- BAUD** Baud rate (in bits per second, bps)
- f_{osc}** System Oscillator clock frequency
- UBRRn** Contents of the UBRRnH and UBRRnL Registers, (0-4095)

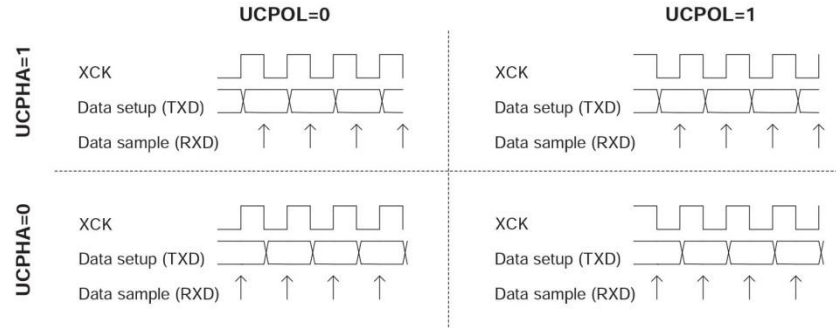
20.4 SPI Data Modes and Timing

There are four combinations of XCKn (SCK) phase and polarity with respect to serial data, which are determined by control bits UCPHAn and UCPOLn. The data transfer timing diagrams are shown in Figure 20-1. Data bits are shifted out and latched in on opposite edges of the XCKn signal, ensuring sufficient time for data signals to stabilize. The UCPOLn and UCPHAn functionality is summarized in Table 20-2. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and Transmitter.

Table 20-2. UCPOLn and UCPHAn Functionality-

UCPOLn	UCPHAn	SPI Mode	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
0	1	1	Setup (Rising)	Sample (Falling)
1	0	2	Sample (Falling)	Setup (Rising)
1	1	3	Setup (Falling)	Sample (Rising)

Figure 20-1. UCPHAn and UCPOLn data transfer timing diagrams.



20.5 Frame Formats

A serial frame for the MSPIM is defined to be one character of 8 data bits. The USART in MSPIM mode has two valid frame formats:

- 8-bit data with MSB first
- 8-bit data with LSB first

A frame starts with the least or most significant data bit. Then the next data bits, up to a total of eight, are succeeding, ending with the most or least significant bit accordingly. When a complete frame is transmitted, a new frame can directly follow it, or the communication line can be set to an idle (high) state.

The UDORDn bit in UCSRNc sets the frame format used by the USART in MSPIM mode. The Receiver and Transmitter use the same setting. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and Transmitter.

16-bit data transfer can be achieved by writing two data bytes to UDRn. A UART transmit complete interrupt will then signal that the 16-bit value has been shifted out.

20.5.1 USART MSPIM Initialization

The USART in MSPIM mode has to be initialized before any communication can take place. The initialization process normally consists of setting the baud rate, setting master mode of operation (by setting DDR_XCKn to one), setting frame format and enabling the Transmitter and the Receiver. Only the transmitter can operate independently. For interrupt driven USART operation, the Global Interrupt Flag should be cleared (and thus interrupts globally disabled) when doing the initialization.

Note: To ensure immediate initialization of the XCKn output the baud-rate register (UBRRn) must be zero at the time the transmitter is enabled. Contrary to the normal mode USART operation the UBRRn must then be written to the desired value after the transmitter is enabled, but before the first transmission is started. Setting UBRRn to zero before enabling the transmitter is not necessary if the initialization is done immediately after a reset since UBRRn is reset to zero.

Before doing a re-initialization with changed baud rate, data mode, or frame format, be sure that there is no ongoing transmissions during the period the registers are changed. The TXCn Flag can be used to check that the Transmitter has completed all transfers, and the RXCn Flag can

be used to check that there are no unread data in the receive buffer. Note that the TXCn Flag must be cleared before each transmission (before UDRn is written) if it is used for this purpose.

The following simple USART initialization code examples show one assembly and one C function that are equal in functionality. The examples assume polling (no interrupts enabled). The baud rate is given as a function parameter. For the assembly code, the baud rate parameter is assumed to be stored in the r17:r16 registers.

Assembly Code Example⁽¹⁾

```
USART_Init:
    clr r18
    out UBRRnH,r18
    out UBRRnL,r18
    ; Setting the XCKn port pin as output, enables master mode.
    sbi XCKn_DDR, XCKn
    ; Set MSPI mode of operation and SPI data mode 0.
    ldi r18, (1<<UMSELn1)|(1<<UMSELn0)|(0<<UCPHAn)|(0<<UCPOLn)
    out UCSRnC,r18
    ; Enable receiver and transmitter.
    ldi r18, (1<<RXENn)|(1<<TXENn)
    out UCSRnB,r18
    ; Set baud rate.
    ; IMPORTANT: The Baud Rate must be set after the transmitter is enabled!
    out UBRRnH, r17
    out UBRRnL, r18
    ret
```

C Code Example⁽¹⁾

```
void USART_Init( unsigned int baud )
{
    UBRRn = 0;
    /* Setting the XCKn port pin as output, enables master mode. */
    XCKn_DDR |= (1<<XCKn);
    /* Set MSPI mode of operation and SPI data mode 0. */
    UCSRnC = (1<<UMSELn1)|(1<<UMSELn0)|(0<<UCPHAn)|(0<<UCPOLn);
    /* Enable receiver and transmitter. */
    UCSRnB = (1<<RXENn)|(1<<TXENn);
    /* Set baud rate. */
    /* IMPORTANT: The Baud Rate must be set after the transmitter is enabled
    */
    UBRRn = baud;
}
```

Note: 1. See "About Code Examples" on page 7.

20.6 Data Transfer

Using the USART in MSPI mode requires the Transmitter to be enabled, i.e. the TXENn bit in the UCSRnB register is set to one. When the Transmitter is enabled, the normal port operation of the TxDn pin is overridden and given the function as the Transmitter's serial output. Enabling the receiver is optional and is done by setting the RXENn bit in the UCSRnB register to one. When the receiver is enabled, the normal pin operation of the RxDn pin is overridden and given the function as the Receiver's serial input. The XCKn will in both cases be used as the transfer clock.

After initialization the USART is ready for doing data transfers. A data transfer is initiated by writing to the UDRn I/O location. This is the case for both sending and receiving data since the transmitter controls the transfer clock. The data written to UDRn is moved from the transmit buffer to the shift register when the shift register is ready to send a new frame.

Note: To keep the input buffer in sync with the number of data bytes transmitted, the UDRn register must be read once for each byte transmitted. The input buffer operation is identical to normal USART mode, i.e. if an overflow occurs the character last received will be lost, not the first data in the buffer. This means that if four bytes are transferred, byte 1 first, then byte 2, 3, and 4, and the UDRn is not read before all transfers are completed, then byte 3 to be received will be lost, and not byte 1.

The following code examples show a simple USART in MSPIM mode transfer function based on polling of the Data Register Empty (UDREN) Flag and the Receive Complete (RXCn) Flag. The USART has to be initialized before the function can be used. For the assembly code, the data to be sent is assumed to be stored in Register R16 and the data received will be available in the same register (R16) after the function returns.

The function simply waits for the transmit buffer to be empty by checking the UDREN Flag, before loading it with new data to be transmitted. The function then waits for data to be present in the receive buffer by checking the RXCn Flag, before reading the buffer and returning the value.

Assembly Code Example⁽¹⁾

```

USART_MSPIM_Transfer:
    ; Wait for empty transmit buffer
    sbis UCSRnA, UDREn
    rjmp USART_MSPIM_Transfer
    ; Put data (r16) into buffer, sends the data
    out UDRn,r16
    ; Wait for data to be received
USART_MSPIM_Wait_RXCn:
    sbis UCSRnA, RXCn
    rjmp USART_MSPIM_Wait_RXCn
    ; Get and return received data from buffer
    in r16, UDRn
    ret

```

C Code Example⁽¹⁾

```

unsigned char USART_Receive( void )
{
    /* Wait for empty transmit buffer */
    while ( !( UCSRnA & (1<<UDREn) ) );
    /* Put data into buffer, sends the data */
    UDRn = data;
    /* Wait for data to be received */
    while ( !(UCSRnA & (1<<RXCn) ) );
    /* Get and return received data from buffer */
    return UDRn;
}

```

Note: 1. See "About Code Examples" on page 7.

20.6.1 Transmitter and Receiver Flags and Interrupts

The RXCn, TXCn, and UDREn flags and corresponding interrupts in USART in MSPIM mode are identical in function to the normal USART operation. However, the receiver error status flags (FE, DOR, and PE) are not in use and is always read as zero.

20.6.2 Disabling the Transmitter or Receiver

The disabling of the transmitter or receiver in USART in MSPIM mode is identical in function to the normal USART operation.

20.7 AVR USART MSPIM vs. AVR SPI

The USART in MSPIM mode is fully compatible with the AVR SPI regarding:

- Master mode timing diagram.
- The UCPOLn bit functionality is identical to the SPI CPOL bit.
- The UCPHAn bit functionality is identical to the SPI CPHA bit.
- The UDORDn bit functionality is identical to the SPI DORD bit.

However, since the USART in MSPIM mode reuses the USART resources, the use of the USART in MSPIM mode is somewhat different compared to the SPI. In addition to differences of the control register bits, and that only master operation is supported by the USART in MSPIM mode, the following features differ between the two modules:

- The USART in MSPIM mode includes (double) buffering of the transmitter. The SPI has no buffer.
- The USART in MSPIM mode receiver includes an additional buffer level.
- The SPI WCOL (Write Collision) bit is not included in USART in MSPIM mode.
- The SPI double speed mode (SPI2X) bit is not included. However, the same effect is achieved by setting UBRRn accordingly.
- Interrupt timing is not compatible.
- Pin control differs due to the master only operation of the USART in MSPIM mode.

A comparison of the USART in MSPIM mode and the SPI pins is shown in [Table 20-3 on page 210](#).

Table 20-3. Comparison of USART in MSPIM mode and SPI pins.

USART_MSPIM	SPI	Comment
TxDn	MOSI	Master Out only
RxDn	MISO	Master In only
XCKn	SCK	(Functionally identical)
(N/A)	\overline{SS}	Not supported by USART in MSPIM

20.8 Register Description

The following section describes the registers used for SPI operation using the USART.

20.8.1 UDRn – USART MSPIM I/O Data Register

The function and bit description of the USART data register (UDRn) in MSPI mode is identical to normal USART operation. See "UDRn – USART I/O Data Register n" on page 195.

20.8.2 UCSRnA – USART MSPIM Control and Status Register n A

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	-	-	-	-	-	UCSRnA
Read/Write	R	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 7 - RXCn: USART Receive Complete**

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the Receiver is disabled, the receive buffer will be flushed and consequently the RXCn bit will become zero. The RXCn Flag can be used to generate a Receive Complete interrupt (see description of the RXCIEn bit).

- **Bit 6 - TXCn: USART Transmit Complete**

This flag bit is set when the entire frame in the Transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer (UDRn). The TXCn Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXCn Flag can generate a Transmit Complete interrupt (see description of the TXCIEn bit).

- **Bit 5 - UDREn: USART Data Register Empty**

The UDREn Flag indicates if the transmit buffer (UDRn) is ready to receive new data. If UDREn is one, the buffer is empty, and therefore ready to be written. The UDREn Flag can generate a Data Register Empty interrupt (see description of the UDRIE bit). UDREn is set after a reset to indicate that the Transmitter is ready.

- **Bit 4:0 - Reserved Bits in MSPI mode**

When in MSPI mode, these bits are reserved for future use. For compatibility with future devices, these bits must be written to zero when UCSRnA is written.

20.8.3 UCSRnB – USART MSPIM Control and Status Register n B

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIE	RXENn	TXENn	-	-	-	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 7 - RXCIEn: RX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the RXCn Flag. A USART Receive Complete interrupt will be generated only if the RXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXCn bit in UCSRnA is set.

- **Bit 6 - TXCIEn: TX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the TXCn Flag. A USART Transmit Complete interrupt will be generated only if the TXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXCn bit in UCSRnA is set.

- **Bit 5 - UDRIE: USART Data Register Empty Interrupt Enable**

Writing this bit to one enables interrupt on the UDREn Flag. A Data Register Empty interrupt will be generated only if the UDRIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the UDREn bit in UCSRnA is set.

- **Bit 4 - RXENn: Receiver Enable**

Writing this bit to one enables the USART Receiver in MSPIM mode. The Receiver will override normal port operation for the RxDn pin when enabled. Disabling the Receiver will flush the receive buffer. Only enabling the receiver in MSPI mode (i.e. setting RXENn=1 and TXENn=0) has no meaning since it is the transmitter that controls the transfer clock and since only master mode is supported.

- **Bit 3 - TXENn: Transmitter Enable**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxDn pin when enabled. The disabling of the Transmitter (writing TXENn to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the Transmit Shift Register and Transmit Buffer Register do not contain data to be transmitted. When disabled, the Transmitter will no longer override the TxDn port.

- **Bit 2:0 - Reserved Bits in MSPI mode**

When in MSPI mode, these bits are reserved for future use. For compatibility with future devices, these bits must be written to zero when UCSRnB is written.

20.8.4 UCSRnC – USART MSPIM Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	-	-	-	UDORDn	UCPHAn	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 7:6 - UMSELn1:0: USART Mode Select**

These bits select the mode of operation of the USART as shown in Table 20-4. See “UCSRnC – USART Control and Status Register n C” on page 197 for full description of the normal USART operation. The MSPIM is enabled when both UMSELn bits are set to one. The UDORDn, UCPHAn, and UCPOLn can be set in the same write operation where the MSPIM is enabled.

Table 20-4. UMSELn Bits Settings

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	Reserved
1	1	Master SPI (MSPIM)

- **Bit 5:3 - Reserved Bits in MSPI mode**

When in MSPI mode, these bits are reserved for future use. For compatibility with future devices, these bits must be written to zero when UCSRnC is written.

- **Bit 2 - UDORDn: Data Order**

When set to one the LSB of the data word is transmitted first. When set to zero the MSB of the data word is transmitted first. Refer to the Frame Formats section page 4 for details.

- **Bit 1 - UCPHAn: Clock Phase**

The UCPHAn bit setting determine if data is sampled on the leading edge (first) or trailing (last) edge of XCKn. Refer to the SPI Data Modes and Timing section page 4 for details.

- **Bit 0 - UCPOLn: Clock Polarity**

The UCPOLn bit sets the polarity of the XCKn clock. The combination of the UCPOLn and UCPHAn bit settings determine the timing of the data transfer. Refer to the SPI Data Modes and Timing section page 4 for details.

20.8.5 USART MSPIM Baud Rate Registers - UBRRnL and UBRRnH

The function and bit description of the baud rate registers in MSPI mode is identical to normal USART operation. See "UBRRnL and UBRRnH – USART Baud Rate Registers" on page 199.

21. 2-wire Serial Interface

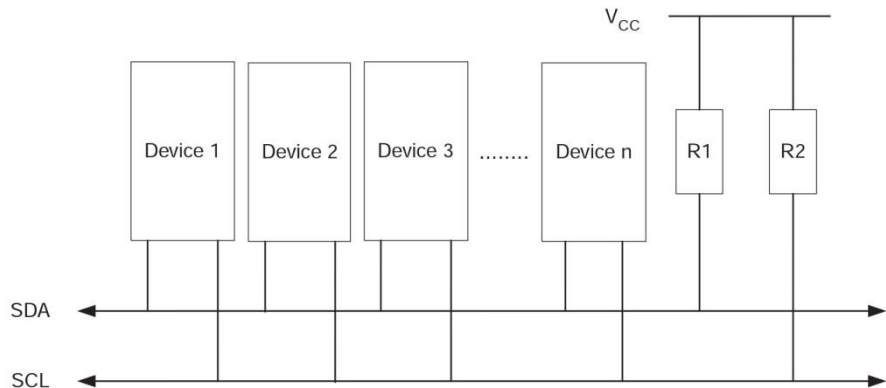
21.1 Features

- Simple Yet Powerful and Flexible Communication Interface, only two Bus Lines Needed
- Both Master and Slave Operation Supported
- Device can Operate as Transmitter or Receiver
- 7-bit Address Space Allows up to 128 Different Slave Addresses
- Multi-master Arbitration Support
- Up to 400 kHz Data Transfer Speed
- Slew-rate Limited Output Drivers
- Noise Suppression Circuitry Rejects Spikes on Bus Lines
- Fully Programmable Slave Address with General Call Support
- Address Recognition Causes Wake-up When AVR is in Sleep Mode
- Compatible with Philips' I²C protocol

21.2 2-wire Serial Interface Bus Definition

The 2-wire Serial Interface (TWI) is ideally suited for typical microcontroller applications. The TWI protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (SCL) and one for data (SDA). The only external hardware needed to implement the bus is a single pull-up resistor for each of the TWI bus lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus contention are inherent in the TWI protocol.

Figure 21-1. TWI Bus Interconnection



21.2.1 TWI Terminology

The following definitions are frequently encountered in this section.

Table 21-1. TWI Terminology

Term	Description
Master	The device that initiates and terminates a transmission. The Master also generates the SCL clock.
Slave	The device addressed by a Master.
Transmitter	The device placing data on the bus.
Receiver	The device reading data from the bus.

The PRTWI bit in ["Minimizing Power Consumption" on page 42](#) must be written to zero to enable the 2-wire Serial Interface.

21.2.2 Electrical Interconnection

As depicted in [Figure 21-1](#), both bus lines are connected to the positive supply voltage through pull-up resistors. The bus drivers of all TWI-compliant devices are open-drain or open-collector. This implements a wired-AND function which is essential to the operation of the interface. A low level on a TWI bus line is generated when one or more TWI devices output a zero. A high level is output when all TWI devices tri-state their outputs, allowing the pull-up resistors to pull the line high. Note that all AVR devices connected to the TWI bus must be powered in order to allow any bus operation.

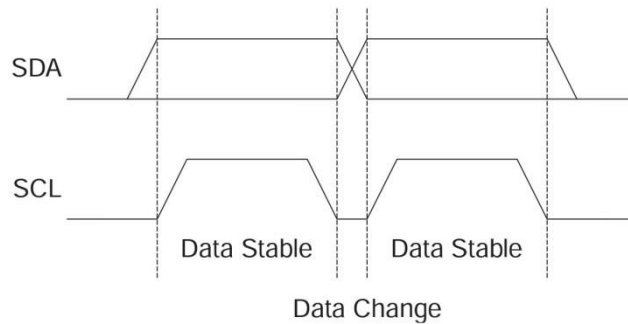
The number of devices that can be connected to the bus is only limited by the bus capacitance limit of 400 pF and the 7-bit slave address space. A detailed specification of the electrical characteristics of the TWI is given in ["2-wire Serial Interface Characteristics" on page 321](#). Two different sets of specifications are presented there, one relevant for bus speeds below 100 kHz, and one valid for bus speeds up to 400 kHz.

21.3 Data Transfer and Frame Format

21.3.1 Transferring Bits

Each data bit transferred on the TWI bus is accompanied by a pulse on the clock line. The level of the data line must be stable when the clock line is high. The only exception to this rule is for generating start and stop conditions.

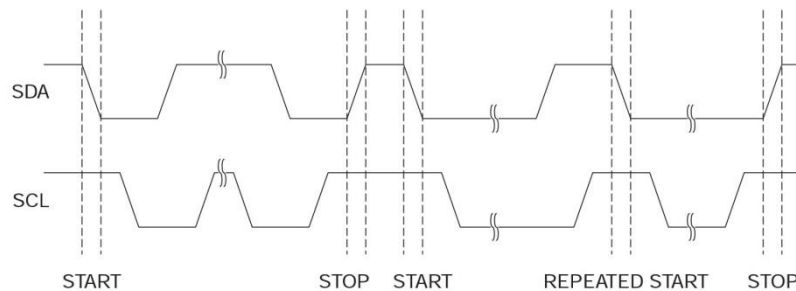
Figure 21-2. Data Validity



21.3.2 START and STOP Conditions

The Master initiates and terminates a data transmission. The transmission is initiated when the Master issues a START condition on the bus, and it is terminated when the Master issues a STOP condition. Between a START and a STOP condition, the bus is considered busy, and no other master should try to seize control of the bus. A special case occurs when a new START condition is issued between a START and STOP condition. This is referred to as a REPEATED START condition, and is used when the Master wishes to initiate a new transfer without relinquishing control of the bus. After a REPEATED START, the bus is considered busy until the next STOP. This is identical to the START behavior, and therefore START is used to describe both START and REPEATED START for the remainder of this datasheet, unless otherwise noted. As depicted below, START and STOP conditions are signalled by changing the level of the SDA line when the SCL line is high.

Figure 21-3. START, REPEATED START and STOP conditions



21.3.3 Address Packet Format

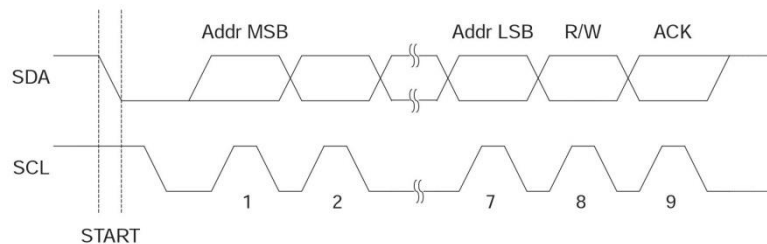
All address packets transmitted on the TWI bus are 9 bits long, consisting of 7 address bits, one READ/WRITE control bit and an acknowledge bit. If the READ/WRITE bit is set, a read operation is to be performed, otherwise a write operation should be performed. When a Slave recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle. If the addressed Slave is busy, or for some other reason can not service the Master's request, the SDA line should be left high in the ACK clock cycle. The Master can then transmit a STOP condition, or a REPEATED START condition to initiate a new transmission. An address packet consisting of a slave address and a READ or a WRITE bit is called SLA+R or SLA+W, respectively.

The MSB of the address byte is transmitted first. Slave addresses can freely be allocated by the designer, but the address 0000 000 is reserved for a general call.

When a general call is issued, all slaves should respond by pulling the SDA line low in the ACK cycle. A general call is used when a Master wishes to transmit the same message to several slaves in the system. When the general call address followed by a Write bit is transmitted on the bus, all slaves set up to acknowledge the general call will pull the SDA line low in the ack cycle. The following data packets will then be received by all the slaves that acknowledged the general call. Note that transmitting the general call address followed by a Read bit is meaningless, as this would cause contention if several slaves started transmitting different data.

All addresses of the format 1111 xxx should be reserved for future purposes.

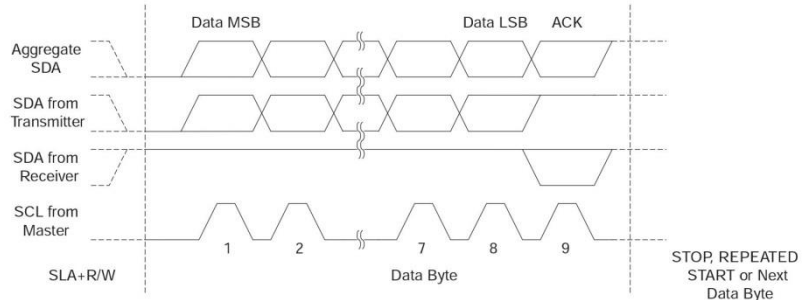
Figure 21-4. Address Packet Format



21.3.4 Data Packet Format

All data packets transmitted on the TWI bus are nine bits long, consisting of one data byte and an acknowledge bit. During a data transfer, the Master generates the clock and the START and STOP conditions, while the Receiver is responsible for acknowledging the reception. An Acknowledge (ACK) is signalled by the Receiver pulling the SDA line low during the ninth SCL cycle. If the Receiver leaves the SDA line high, a NACK is signalled. When the Receiver has received the last byte, or for some reason cannot receive any more bytes, it should inform the Transmitter by sending a NACK after the final byte. The MSB of the data byte is transmitted first.

Figure 21-5. Data Packet Format

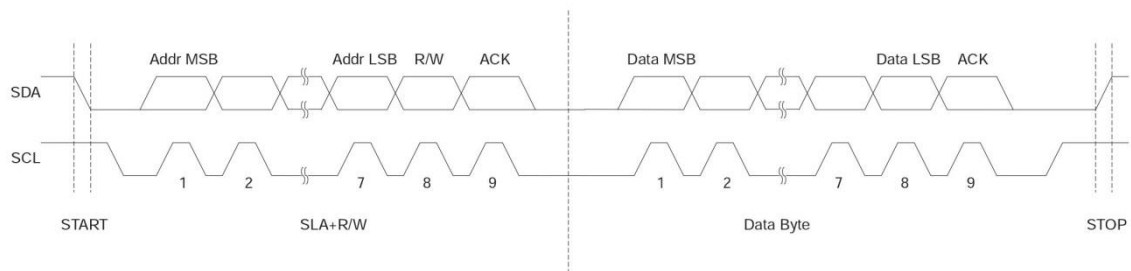


21.3.5 Combining Address and Data Packets into a Transmission

A transmission basically consists of a START condition, a SLA+R/W, one or more data packets and a STOP condition. An empty message, consisting of a START followed by a STOP condition, is illegal. Note that the Wired-ANDing of the SCL line can be used to implement handshaking between the Master and the Slave. The Slave can extend the SCL low period by pulling the SCL line low. This is useful if the clock speed set up by the Master is too fast for the Slave, or the Slave needs extra time for processing between the data transmissions. The Slave extending the SCL low period will not affect the SCL high period, which is determined by the Master. As a consequence, the Slave can reduce the TWI data transfer speed by prolonging the SCL duty cycle.

Figure 21-6 shows a typical data transmission. Note that several data bytes can be transmitted between the SLA+R/W and the STOP condition, depending on the software protocol implemented by the application software.

Figure 21-6. Typical Data Transmission



21.4 Multi-master Bus Systems, Arbitration and Synchronization

The TWI protocol allows bus systems with several masters. Special concerns have been taken in order to ensure that transmissions will proceed as normal, even if two or more masters initiate a transmission at the same time. Two problems arise in multi-master systems:

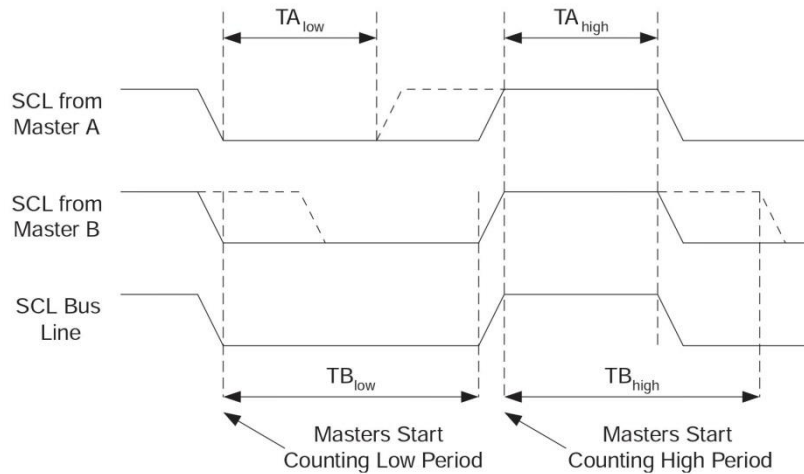
- An algorithm must be implemented allowing only one of the masters to complete the transmission. All other masters should cease transmission when they discover that they have lost the selection process. This selection process is called arbitration. When a contending master discovers that it has lost the arbitration process, it should immediately switch to Slave mode to check whether it is being addressed by the winning master. The fact that multiple

masters have started transmission at the same time should not be detectable to the slaves, i.e. the data being transferred on the bus must not be corrupted.

- Different masters may use different SCL frequencies. A scheme must be devised to synchronize the serial clocks from all masters, in order to let the transmission proceed in a lockstep fashion. This will facilitate the arbitration process.

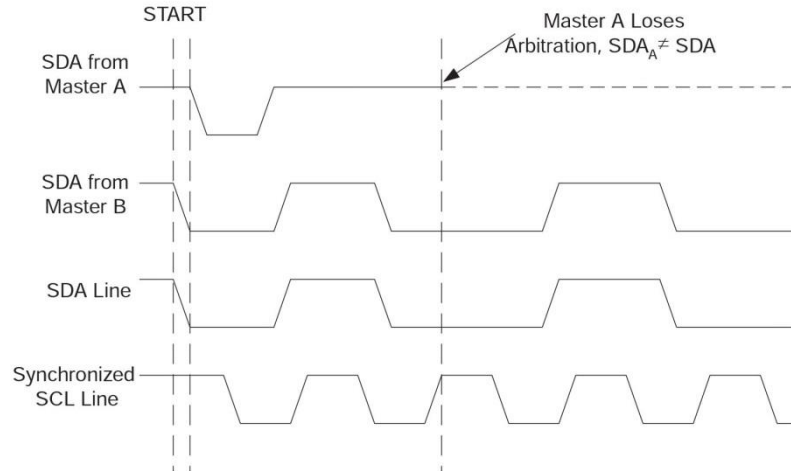
The wired-ANDing of the bus lines is used to solve both these problems. The serial clocks from all masters will be wired-ANDed, yielding a combined clock with a high period equal to the one from the Master with the shortest high period. The low period of the combined clock is equal to the low period of the Master with the longest low period. Note that all masters listen to the SCL line, effectively starting to count their SCL high and low time-out periods when the combined SCL line goes high or low, respectively.

Figure 21-7. SCL Synchronization Between Multiple Masters



Arbitration is carried out by all masters continuously monitoring the SDA line after outputting data. If the value read from the SDA line does not match the value the Master had output, it has lost the arbitration. Note that a Master can only lose arbitration when it outputs a high SDA value while another Master outputs a low value. The losing Master should immediately go to Slave mode, checking if it is being addressed by the winning Master. The SDA line should be left high, but losing masters are allowed to generate a clock signal until the end of the current data or address packet. Arbitration will continue until only one Master remains, and this may take many bits. If several masters are trying to address the same Slave, arbitration will continue into the data packet.

Figure 21-8. Arbitration Between Two Masters



Note that arbitration is not allowed between:

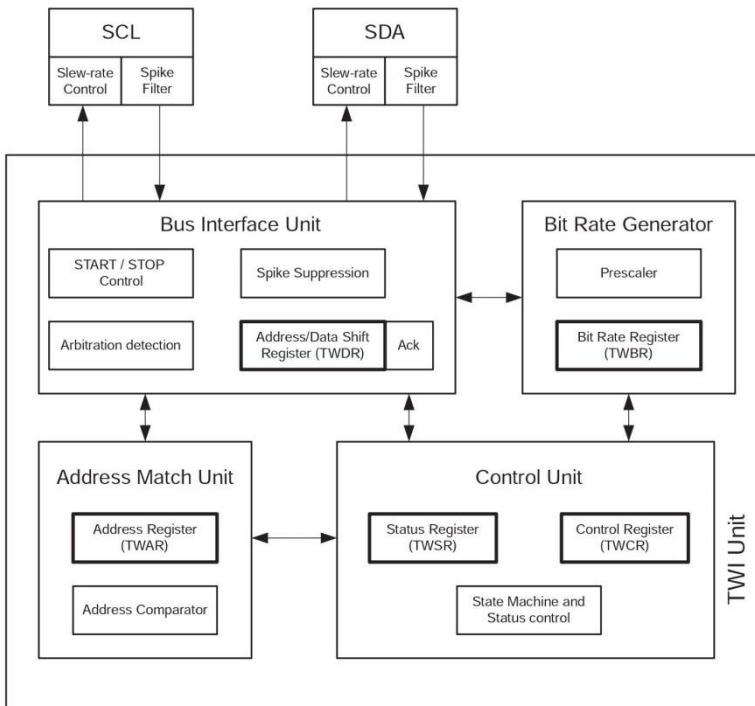
- A REPEATED START condition and a data bit.
- A STOP condition and a data bit.
- A REPEATED START and a STOP condition.

It is the user software's responsibility to ensure that these illegal arbitration conditions never occur. This implies that in multi-master systems, all data transfers must use the same composition of SLA+R/W and data packets. In other words: All transmissions must contain the same number of data packets, otherwise the result of the arbitration is undefined.

21.5 Overview of the TWI Module

The TWI module is comprised of several submodules, as shown in Figure 21-9. All registers drawn in a thick line are accessible through the AVR data bus.

Figure 21-9. Overview of the TWI Module



21.5.1 SCL and SDA Pins

These pins interface the AVR TWI with the rest of the MCU system. The output drivers contain a slew-rate limiter in order to conform to the TWI specification. The input stages contain a spike suppression unit removing spikes shorter than 50 ns. Note that the internal pull-ups in the AVR pads can be enabled by setting the PORT bits corresponding to the SCL and SDA pins, as explained in the I/O Port section. The internal pull-ups can in some systems eliminate the need for external ones.

21.5.2 Bit Rate Generator Unit

This unit controls the period of SCL when operating in a Master mode. The SCL period is controlled by settings in the TWI Bit Rate Register (TWBR) and the Prescaler bits in the TWI Status Register (TWSR). Slave operation does not depend on Bit Rate or Prescaler settings, but the CPU clock frequency in the Slave must be at least 16 times higher than the SCL frequency. Note

that slaves may prolong the SCL low period, thereby reducing the average TWI bus clock period. The SCL frequency is generated according to the following equation:

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 \cdot 2^{\text{(TWBR)}} \cdot (\text{PrescalerValue})}$$

- TWBR = Value of the TWI Bit Rate Register.
- *PrescalerValue* = Value of the prescaler, see [Table 21-7 on page 243](#).

Note: Pull-up resistor values should be selected according to the SCL frequency and the capacitive bus line load. See [Table 28-6 on page 321](#) for value of pull-up resistor.

21.5.3 Bus Interface Unit

This unit contains the Data and Address Shift Register (TWDR), a START/STOP Controller and Arbitration detection hardware. The TWDR contains the address or data bytes to be transmitted, or the address or data bytes received. In addition to the 8-bit TWDR, the Bus Interface Unit also contains a register containing the (N)ACK bit to be transmitted or received. This (N)ACK Register is not directly accessible by the application software. However, when receiving, it can be set or cleared by manipulating the TWI Control Register (TWCR). When in Transmitter mode, the value of the received (N)ACK bit can be determined by the value in the TWSR.

The START/STOP Controller is responsible for generation and detection of START, REPEATED START, and STOP conditions. The START/STOP controller is able to detect START and STOP conditions even when the AVR MCU is in one of the sleep modes, enabling the MCU to wake up if addressed by a Master.

If the TWI has initiated a transmission as Master, the Arbitration Detection hardware continuously monitors the transmission trying to determine if arbitration is in process. If the TWI has lost an arbitration, the Control Unit is informed. Correct action can then be taken and appropriate status codes generated.

21.5.4 Address Match Unit

The Address Match unit checks if received address bytes match the seven-bit address in the TWI Address Register (TWAR). If the TWI General Call Recognition Enable (TWGCE) bit in the TWAR is written to one, all incoming address bits will also be compared against the General Call address. Upon an address match, the Control Unit is informed, allowing correct action to be taken. The TWI may or may not acknowledge its address, depending on settings in the TWCR. The Address Match unit is able to compare addresses even when the AVR MCU is in sleep mode, enabling the MCU to wake up if addressed by a Master.

If another interrupt (e.g., INT0) occurs during TWI Power-down address match and wakes up the CPU, the TWI aborts operation and return to its idle state. If this cause any problems, ensure that TWI Address Match is the only enabled interrupt when entering Power-down⁽¹⁾.

Note: 1. This applies to all device revisions except ATmega88PA revision C or newer.

21.5.5 Control Unit

The Control unit monitors the TWI bus and generates responses corresponding to settings in the TWI Control Register (TWCR). When an event requiring the attention of the application occurs on the TWI bus, the TWI Interrupt Flag (TWINT) is asserted. In the next clock cycle, the TWI Status Register (TWSR) is updated with a status code identifying the event. The TWSR only contains relevant status information when the TWI Interrupt Flag is asserted. At all other times, the TWSR contains a special status code indicating that no relevant status information is avail-

able. As long as the TWINT Flag is set, the SCL line is held low. This allows the application software to complete its tasks before allowing the TWI transmission to continue.

The TWINT Flag is set in the following situations:

- After the TWI has transmitted a START/REPEATED START condition.
- After the TWI has transmitted SLA+R/W.
- After the TWI has transmitted an address byte.
- After the TWI has lost arbitration.
- After the TWI has been addressed by own slave address or general call.
- After the TWI has received a data byte.
- After a STOP or REPEATED START has been received while still addressed as a Slave.
- When a bus error has occurred due to an illegal START or STOP condition.

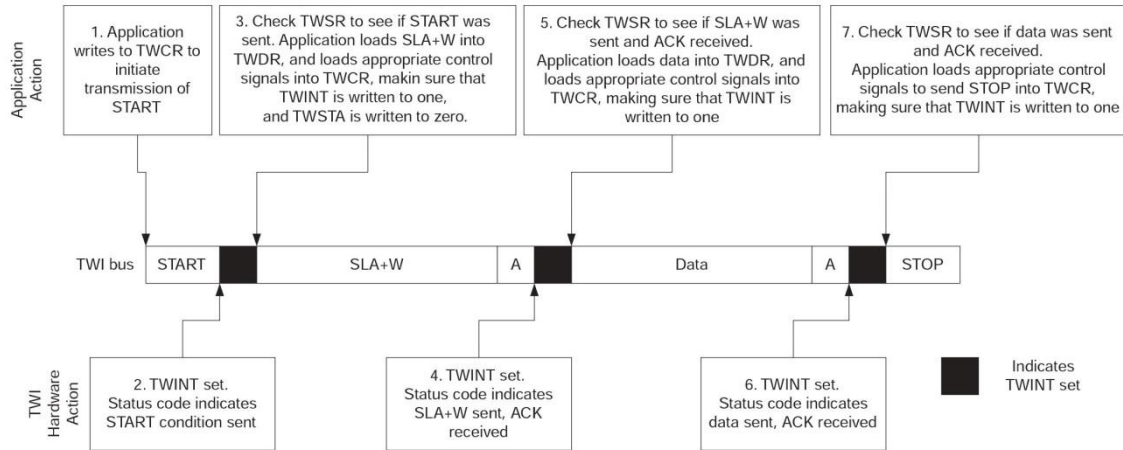
21.6 Using the TWI

The AVR TWI is byte-oriented and interrupt based. Interrupts are issued after all bus events, like reception of a byte or transmission of a START condition. Because the TWI is interrupt-based, the application software is free to carry on other operations during a TWI byte transfer. Note that the TWI Interrupt Enable (TWIE) bit in TWCR together with the Global Interrupt Enable bit in SREG allow the application to decide whether or not assertion of the TWINT Flag should generate an interrupt request. If the TWIE bit is cleared, the application must poll the TWINT Flag in order to detect actions on the TWI bus.

When the TWINT Flag is asserted, the TWI has finished an operation and awaits application response. In this case, the TWI Status Register (TWSR) contains a value indicating the current state of the TWI bus. The application software can then decide how the TWI should behave in the next TWI bus cycle by manipulating the TWCR and TWDR Registers.

Figure 21-10 is a simple example of how the application can interface to the TWI hardware. In this example, a Master wishes to transmit a single data byte to a Slave. This description is quite abstract, a more detailed explanation follows later in this section. A simple code example implementing the desired behavior is also presented.

Figure 21-10. Interfacing the Application to the TWI in a Typical Transmission



1. The first step in a TWI transmission is to transmit a START condition. This is done by writing a specific value into TWCR, instructing the TWI hardware to transmit a START condition. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the START condition.
2. When the START condition has been transmitted, the TWINT Flag in TWCR is set, and TWSR is updated with a status code indicating that the START condition has successfully been sent.
3. The application software should now examine the value of TWSR, to make sure that the START condition was successfully transmitted. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must load SLA+W into TWDR. Remember that TWDR is used both for address and data. After TWDR has been loaded with the desired SLA+W, a specific value must be written to TWCR, instructing the TWI hardware to transmit the SLA+W present in TWDR. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the address packet.
4. When the address packet has been transmitted, the TWINT Flag in TWCR is set, and TWSR is updated with a status code indicating that the address packet has successfully been sent. The status code will also reflect whether a Slave acknowledged the packet or not.
5. The application software should now examine the value of TWSR, to make sure that the address packet was successfully transmitted, and that the value of the ACK bit was as expected. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must load a data packet into TWDR. Subsequently, a specific value must be written to TWCR, instructing the TWI hardware to transmit the data packet present in TWDR. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will

not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the data packet.

6. When the data packet has been transmitted, the TWINT Flag in TWCR is set, and TWSR is updated with a status code indicating that the data packet has successfully been sent. The status code will also reflect whether a Slave acknowledged the packet or not.
7. The application software should now examine the value of TWSR, to make sure that the data packet was successfully transmitted, and that the value of the ACK bit was as expected. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must write a specific value to TWCR, instructing the TWI hardware to transmit a STOP condition. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the STOP condition. Note that TWINT is NOT set after a STOP condition has been sent.

Even though this example is simple, it shows the principles involved in all TWI transmissions. These can be summarized as follows:

- When the TWI has finished an operation and expects application response, the TWINT Flag is set. The SCL line is pulled low until TWINT is cleared.
- When the TWINT Flag is set, the user must update all TWI Registers with the value relevant for the next TWI bus cycle. As an example, TWDR must be loaded with the value to be transmitted in the next bus cycle.
- After all TWI Register updates and other pending application software tasks have been completed, TWCR is written. When writing TWCR, the TWINT bit should be set. Writing a one to TWINT clears the flag. The TWI will then commence executing whatever operation was specified by the TWCR setting.

In the following an assembly and C implementation of the example is given. Note that the code below assumes that several definitions have been made, for example by using include-files.

	Assembly Code Example	C Example	Comments
1	<pre>ldi r16, (1<<TWINT) (1<<TWSTA) (1<<TWEN) out TWCR, r16</pre>	<pre>TWCR = (1<<TWINT) (1<<TWSTA) (1<<TWEN)</pre>	Send START condition
2	<pre>wait1: in r16, TWCR sbrs r16, TWINT rjmp wait1</pre>	<pre>while (!(TWCR & (1<<TWINT))) ;</pre>	Wait for TWINT Flag set. This indicates that the START condition has been transmitted
3	<pre>in r16, TWSR andi r16, 0xF8 cpi r16, START brne ERROR</pre>	<pre>if ((TWSR & 0xF8) != START) ERROR();</pre>	Check value of TWI Status Register. Mask prescaler bits. If status different from START go to ERROR
	<pre>ldi r16, SLA_W out TWDR, r16 ldi r16, (1<<TWINT) (1<<TWEN) out TWCR, r16</pre>	<pre>TWDR = SLA_W; TWCR = (1<<TWINT) (1<<TWEN);</pre>	Load SLA_W into TWDR Register. Clear TWINT bit in TWCR to start transmission of address
4	<pre>wait2: in r16, TWCR sbrs r16, TWINT rjmp wait2</pre>	<pre>while (!(TWCR & (1<<TWINT))) ;</pre>	Wait for TWINT Flag set. This indicates that the SLA+W has been transmitted, and ACK/NACK has been received.
5	<pre>in r16, TWSR andi r16, 0xF8 cpi r16, MT_SLA_ACK brne ERROR</pre>	<pre>if ((TWSR & 0xF8) != MT_SLA_ACK) ERROR();</pre>	Check value of TWI Status Register. Mask prescaler bits. If status different from MT_SLA_ACK go to ERROR
	<pre>ldi r16, DATA out TWDR, r16 ldi r16, (1<<TWINT) (1<<TWEN) out TWCR, r16</pre>	<pre>TWDR = DATA; TWCR = (1<<TWINT) (1<<TWEN);</pre>	Load DATA into TWDR Register. Clear TWINT bit in TWCR to start transmission of data
6	<pre>wait3: in r16, TWCR sbrs r16, TWINT rjmp wait3</pre>	<pre>while (!(TWCR & (1<<TWINT))) ;</pre>	Wait for TWINT Flag set. This indicates that the DATA has been transmitted, and ACK/NACK has been received.
7	<pre>in r16, TWSR andi r16, 0xF8 cpi r16, MT_DATA_ACK brne ERROR</pre>	<pre>if ((TWSR & 0xF8) != MT_DATA_ACK) ERROR();</pre>	Check value of TWI Status Register. Mask prescaler bits. If status different from MT_DATA_ACK go to ERROR
	<pre>ldi r16, (1<<TWINT) (1<<TWEN) (1<<TWSTO) out TWCR, r16</pre>	<pre>TWCR = (1<<TWINT) (1<<TWEN) (1<<TWSTO);</pre>	Transmit STOP condition

21.7 Transmission Modes

The TWI can operate in one of four major modes. These are named Master Transmitter (MT), Master Receiver (MR), Slave Transmitter (ST) and Slave Receiver (SR). Several of these modes can be used in the same application. As an example, the TWI can use MT mode to write data into a TWI EEPROM, MR mode to read the data back from the EEPROM. If other masters are present in the system, some of these might transmit data to the TWI, and then SR mode would be used. It is the application software that decides which modes are legal.

The following sections describe each of these modes. Possible status codes are described along with figures detailing data transmission in each of the modes. These figures contain the following abbreviations:

S: START condition

Rs: REPEATED START condition

R: Read bit (high level at SDA)

W: Write bit (low level at SDA)

A: Acknowledge bit (low level at SDA)

\bar{A} : Not acknowledge bit (high level at SDA)

Data: 8-bit data byte

P: STOP condition

SLA: Slave Address

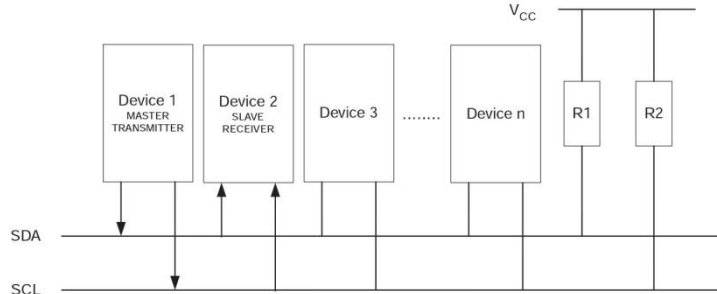
In [Figure 21-12](#) to [Figure 21-18](#), circles are used to indicate that the TWINT Flag is set. The numbers in the circles show the status code held in TWSR, with the prescaler bits masked to zero. At these points, actions must be taken by the application to continue or complete the TWI transfer. The TWI transfer is suspended until the TWINT Flag is cleared by software.

When the TWINT Flag is set, the status code in TWSR is used to determine the appropriate software action. For each status code, the required software action and details of the following serial transfer are given in [Table 21-2](#) to [Table 21-5](#). Note that the prescaler bits are masked to zero in these tables.

21.7.1 Master Transmitter Mode

In the Master Transmitter mode, a number of data bytes are transmitted to a Slave Receiver (see [Figure 21-11](#)). In order to enter a Master mode, a START condition must be transmitted. The format of the following address packet determines whether Master Transmitter or Master Receiver mode is to be entered. If SLA+W is transmitted, MT mode is entered, if SLA+R is transmitted, MR mode is entered. All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 21-11. Data Transfer in Master Transmitter Mode



A START condition is sent by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	1	0	X	1	0	X

TWEN must be set to enable the 2-wire Serial Interface, TWSTA must be written to one to transmit a START condition and TWINT must be written to one to clear the TWINT Flag. The TWI will then test the 2-wire Serial Bus and generate a START condition as soon as the bus becomes free. After a START condition has been transmitted, the TWINT Flag is set by hardware, and the status code in TWSR will be 0x08 (see Table 21-2). In order to enter MT mode, SLA+W must be transmitted. This is done by writing SLA+W to TWDR. Thereafter the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	0	0	X	1	0	X

When SLA+W have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Possible status codes in Master mode are 0x18, 0x20, or 0x38. The appropriate action to be taken for each of these status codes is detailed in Table 21-2.

When SLA+W has been successfully transmitted, a data packet should be transmitted. This is done by writing the data byte to TWDR. TWDR must only be written when TWINT is high. If not, the access will be discarded, and the Write Collision bit (TWWC) will be set in the TWCR Register. After updating TWDR, the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	0	0	X	1	0	X

This scheme is repeated until the last byte has been sent and the transfer is ended by generating a STOP condition or a repeated START condition. A STOP condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	0	1	X	1	0	X

A REPEATED START condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	1	0	X	1	0	X

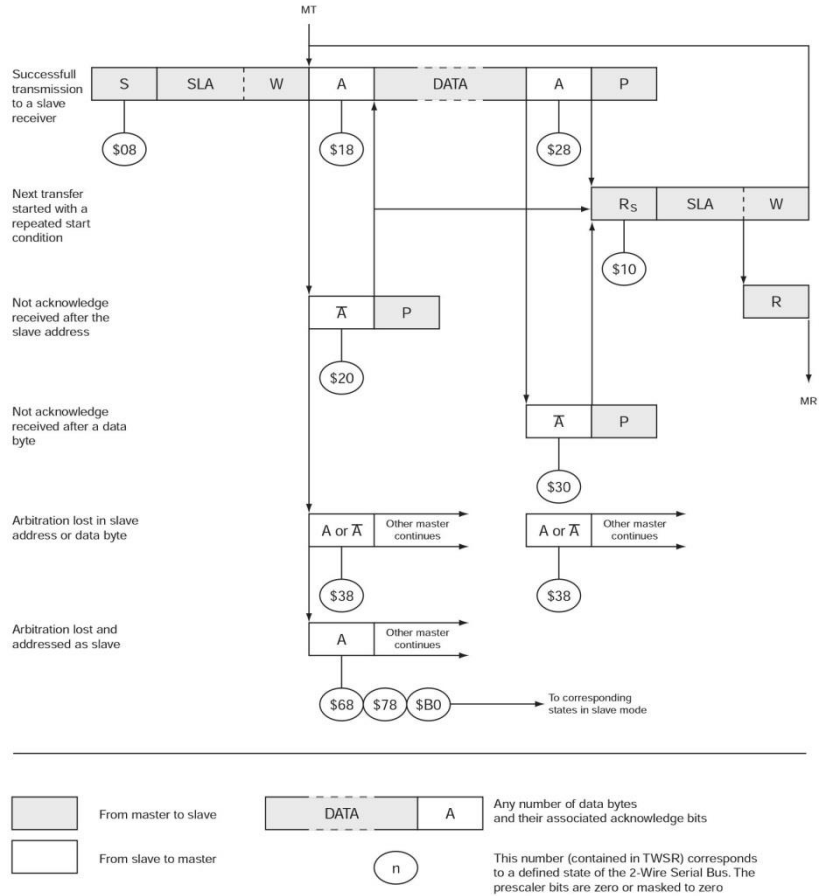
ATmega48PA/88PA/168PA/328P

After a repeated START condition (state 0x10) the 2-wire Serial Interface can access the same Slave again, or a new Slave without transmitting a STOP condition. Repeated START enables the Master to switch between Slaves, Master Transmitter mode and Master Receiver mode without losing control of the bus.

Table 21-2. Status codes for Master Transmitter Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCN				
			STA	STO	TWINT	TWEA	
0x08	A START condition has been transmitted	Load SLA+W	0	0	1	X	SLA+W will be transmitted; ACK or NOT ACK will be received
0x10	A repeated START condition has been transmitted	Load SLA+W or	0	0	1	X	SLA+W will be transmitted; ACK or NOT ACK will be received SLA+R will be transmitted; Logic will switch to Master Receiver mode
		Load SLA+R	0	0	1	X	
0x18	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be received Repeated START will be transmitted STOP condition will be transmitted and TWSTO Flag will be reset STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
		No TWDR action or	1	0	1	X	
		No TWDR action or	0	1	1	X	
0x20	SLA+W has been transmitted; NOT ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be received Repeated START will be transmitted STOP condition will be transmitted and TWSTO Flag will be reset STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
		No TWDR action or	1	0	1	X	
		No TWDR action or	0	1	1	X	
0x28	Data byte has been transmitted; ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be received Repeated START will be transmitted STOP condition will be transmitted and TWSTO Flag will be reset STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
		No TWDR action or	1	0	1	X	
		No TWDR action or	0	1	1	X	
0x30	Data byte has been transmitted; NOT ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be received Repeated START will be transmitted STOP condition will be transmitted and TWSTO Flag will be reset STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
		No TWDR action or	1	0	1	X	
		No TWDR action or	0	1	1	X	
0x38	Arbitration lost in SLA+W or data bytes	No TWDR action or	0	0	1	X	2-wire Serial Bus will be released and not addressed Slave mode entered A START condition will be transmitted when the bus becomes free
		No TWDR action	1	0	1	X	

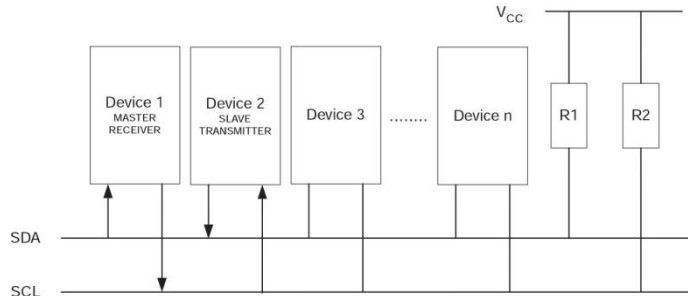
Figure 21-12. Formats and States in the Master Transmitter Mode



21.7.2 Master Receiver Mode

In the Master Receiver mode, a number of data bytes are received from a Slave Transmitter (Slave see Figure 21-13). In order to enter a Master mode, a START condition must be transmitted. The format of the following address packet determines whether Master Transmitter or Master Receiver mode is to be entered. If SLA+W is transmitted, MT mode is entered, if SLA+R is transmitted, MR mode is entered. All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 21-13. Data Transfer in Master Receiver Mode



A START condition is sent by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	1	0	X	1	0	X

TWEN must be written to one to enable the 2-wire Serial Interface, TWSTA must be written to one to transmit a START condition and TWINT must be set to clear the TWINT Flag. The TWI will then test the 2-wire Serial Bus and generate a START condition as soon as the bus becomes free. After a START condition has been transmitted, the TWINT Flag is set by hardware, and the status code in TWSR will be 0x08 (See Table 21-2). In order to enter MR mode, SLA+R must be transmitted. This is done by writing SLA+R to TWDR. Thereafter the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	0	0	X	1	0	X

When SLA+R have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Possible status codes in Master mode are 0x38, 0x40, or 0x48. The appropriate action to be taken for each of these status codes is detailed in Table 21-3. Received data can be read from the TWDR Register when the TWINT Flag is set high by hardware. This scheme is repeated until the last byte has been received. After the last byte has been received, the MR should inform the ST by sending a NACK after the last received data byte. The transfer is ended by generating a STOP condition or a repeated START condition. A STOP condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	0	1	X	1	0	X

A REPEATED START condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	1	0	X	1	0	X

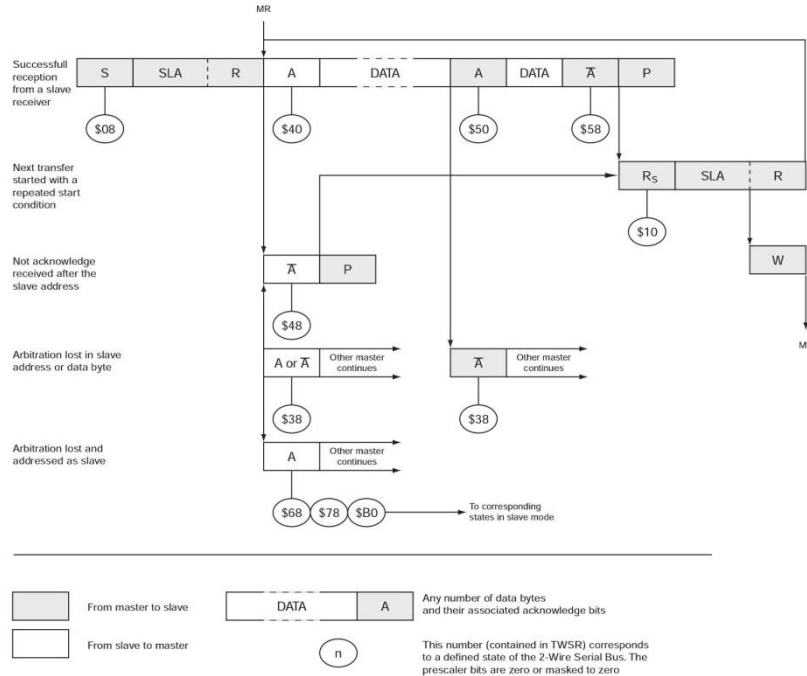
After a repeated START condition (state 0x10) the 2-wire Serial Interface can access the same Slave again, or a new Slave without transmitting a STOP condition. Repeated START enables

the Master to switch between Slaves, Master Transmitter mode and Master Receiver mode without losing control over the bus.

Table 21-3. Status codes for Master Receiver Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0x08	A START condition has been transmitted	Load SLA+R	0	0	1	X	SLA+R will be transmitted ACK or NOT ACK will be received
0x10	A repeated START condition has been transmitted	Load SLA+R or	0	0	1	X	SLA+R will be transmitted ACK or NOT ACK will be received SLA+W will be transmitted Logic will switch to Master Transmitter mode
		Load SLA+W	0	0	1	X	
0x38	Arbitration lost in SLA+R or NOT ACK bit	No TWDR action or	0	0	1	X	2-wire Serial Bus will be released and not addressed Slave mode will be entered A START condition will be transmitted when the bus becomes free
		No TWDR action	1	0	1	X	
0x40	SLA+R has been transmitted; ACK has been received	No TWDR action or	0	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		No TWDR action	0	0	1	1	
0x48	SLA+R has been transmitted; NOT ACK has been received	No TWDR action or	1	0	1	X	Repeated START will be transmitted STOP condition will be transmitted and TWSTO Flag will be reset STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
		No TWDR action or	0	1	1	X	
		No TWDR action	1	1	1	X	
0x50	Data byte has been received; ACK has been returned	Read data byte or	0	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		Read data byte	0	0	1	1	
0x58	Data byte has been received; NOT ACK has been returned	Read data byte or	1	0	1	X	Repeated START will be transmitted STOP condition will be transmitted and TWSTO Flag will be reset STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
		Read data byte or	0	1	1	X	
		Read data byte	1	1	1	X	

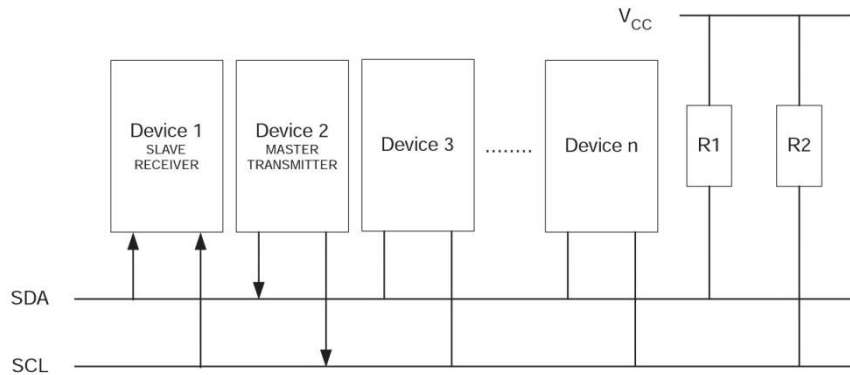
Figure 21-14. Formats and States in the Master Receiver Mode



21.7.3 Slave Receiver Mode

In the Slave Receiver mode, a number of data bytes are received from a Master Transmitter (see Figure 21-15). All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 21-15. Data transfer in Slave Receiver mode



ATmega48PA/88PA/168PA/328P

To initiate the Slave Receiver mode, TWAR and TWCR must be initialized as follows:

TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
value	Device's Own Slave Address							

The upper 7 bits are the address to which the 2-wire Serial Interface will respond when addressed by a Master. If the LSB is set, the TWI will respond to the general call address (0x00), otherwise it will ignore the general call address.

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
value	0	1	0	0	0	1	0	X

TWEN must be written to one to enable the TWI. The TWEA bit must be written to one to enable the acknowledgement of the device's own slave address or the general call address. TWSTA and TWSTO must be written to zero.

When TWAR and TWCR have been initialized, the TWI waits until it is addressed by its own slave address (or the general call address if enabled) followed by the data direction bit. If the direction bit is "0" (write), the TWI will operate in SR mode, otherwise ST mode is entered. After its own slave address and the write bit have been received, the TWINT Flag is set and a valid status code can be read from TWSR. The status code is used to determine the appropriate software action. The appropriate action to be taken for each status code is detailed in [Table 21-4](#). The Slave Receiver mode may also be entered if arbitration is lost while the TWI is in the Master mode (see states 0x68 and 0x78).

If the TWEA bit is reset during a transfer, the TWI will return a "Not Acknowledge" ("1") to SDA after the next received data byte. This can be used to indicate that the Slave is not able to receive any more bytes. While TWEA is zero, the TWI does not acknowledge its own slave address. However, the 2-wire Serial Bus is still monitored and address recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the TWI from the 2-wire Serial Bus.

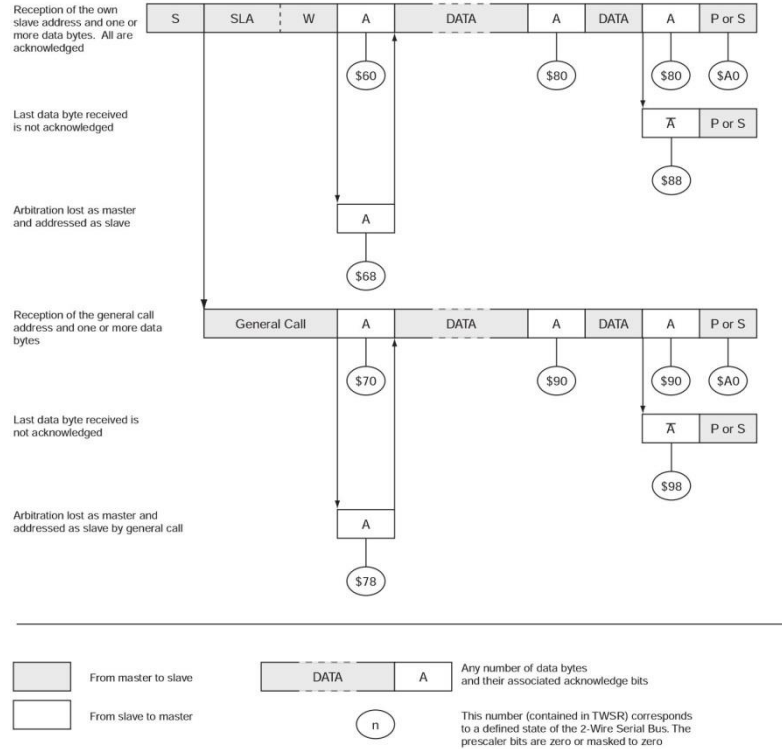
In all sleep modes other than Idle mode, the clock system to the TWI is turned off. If the TWEA bit is set, the interface can still acknowledge its own slave address or the general call address by using the 2-wire Serial Bus clock as a clock source. The part will then wake up from sleep and the TWI will hold the SCL clock low during the wake up and until the TWINT Flag is cleared (by writing it to one). Further data reception will be carried out as normal, with the AVR clocks running as normal. Observe that if the AVR is set up with a long start-up time, the SCL line may be held low for a long time, blocking other data transmissions.

Note that the 2-wire Serial Interface Data Register – TWDR does not reflect the last byte present on the bus when waking up from these Sleep modes.

Table 21-4. Status Codes for Slave Receiver Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0x60	Own SLA+W has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		No TWDR action	X	0	1	1	
0x68	Arbitration lost in SLA+R/W as Master; own SLA+W has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		No TWDR action	X	0	1	1	
0x70	General call address has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		No TWDR action	X	0	1	1	
0x78	Arbitration lost in SLA+R/W as Master; General call address has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		No TWDR action	X	0	1	1	
0x80	Previously addressed with own SLA+W; data has been received; ACK has been returned	Read data byte or	X	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		Read data byte	X	0	1	1	
0x88	Previously addressed with own SLA+W; data has been received; NOT ACK has been returned	Read data byte or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1" Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
		Read data byte or	0	0	1	1	
		Read data byte or	1	0	1	0	
		Read data byte	1	0	1	1	
0x90	Previously addressed with general call; data has been received; ACK has been returned	Read data byte or	X	0	1	0	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned
		Read data byte	X	0	1	1	
0x98	Previously addressed with general call; data has been received; NOT ACK has been returned	Read data byte or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1" Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
		Read data byte or	0	0	1	1	
		Read data byte or	1	0	1	0	
		Read data byte	1	0	1	1	
0xA0	A STOP condition or repeated START condition has been received while still addressed as Slave	No action	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1" Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
			0	0	1	1	
			1	0	1	0	
			1	0	1	1	

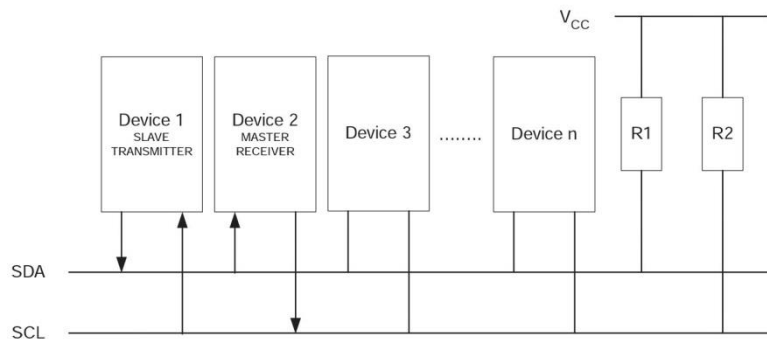
Figure 21-16. Formats and States in the Slave Receiver Mode



21.7.4 Slave Transmitter Mode

In the Slave Transmitter mode, a number of data bytes are transmitted to a Master Receiver (see Figure 21-17). All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 21-17. Data Transfer in Slave Transmitter Mode



To initiate the Slave Transmitter mode, TWAR and TWCR must be initialized as follows:

TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
value	Device's Own Slave Address							

The upper seven bits are the address to which the 2-wire Serial Interface will respond when addressed by a Master. If the LSB is set, the TWI will respond to the general call address (0x00), otherwise it will ignore the general call address.

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
value	0	1	0	0	0	1	0	X

TWEN must be written to one to enable the TWI. The TWEA bit must be written to one to enable the acknowledgement of the device's own slave address or the general call address. TWSTA and TWSTO must be written to zero.

When TWAR and TWCR have been initialized, the TWI waits until it is addressed by its own slave address (or the general call address if enabled) followed by the data direction bit. If the direction bit is "1" (read), the TWI will operate in ST mode, otherwise SR mode is entered. After its own slave address and the write bit have been received, the TWINT flag is set and a valid status code can be read from TWSR. The status code is used to determine the appropriate software action. The appropriate action to be taken for each status code is detailed in [Table 21-5](#). The Slave Transmitter mode may also be entered if arbitration is lost while the TWI is in the Master mode (see state 0xB0).

If the TWEA bit is written to zero during a transfer, the TWI will transmit the last byte of the transfer. State 0xC0 or state 0xC8 will be entered, depending on whether the Master Receiver transmits a NACK or ACK after the final byte. The TWI is switched to the not addressed Slave mode, and will ignore the Master if it continues the transfer. Thus the Master Receiver receives all "1" as serial data. State 0xC8 is entered if the Master demands additional data bytes (by transmitting ACK), even though the Slave has transmitted the last byte (TWEA zero and expecting NACK from the Master).

While TWEA is zero, the TWI does not respond to its own slave address. However, the 2-wire Serial Bus is still monitored and address recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the TWI from the 2-wire Serial Bus.

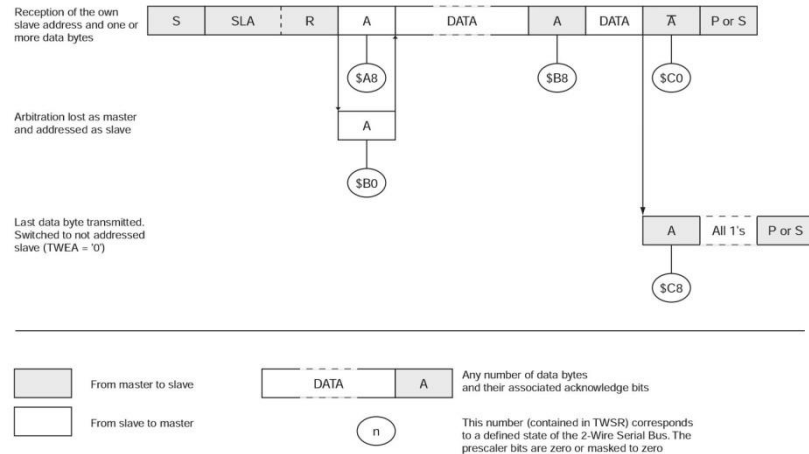
In all sleep modes other than Idle mode, the clock system to the TWI is turned off. If the TWEA bit is set, the interface can still acknowledge its own slave address or the general call address by using the 2-wire Serial Bus clock as a clock source. The part will then wake up from sleep and the TWI will hold the SCL clock low during the wake up and until the TWINT flag is cleared (by writing it to one). Further data transmission will be carried out as normal, with the AVR clocks running as normal. Observe that if the AVR is set up with a long start-up time, the SCL line may be held low for a long time, blocking other data transmissions.

Note that the 2-wire Serial Interface Data Register – TWDR does not reflect the last byte present on the bus when waking up from these sleep modes.

Table 21-5. Status Codes for Slave Transmitter Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0xA8	Own SLA+R has been received; ACK has been returned	Load data byte or	X	0	1	0	Last data byte will be transmitted and NOT ACK should be received Data byte will be transmitted and ACK should be received
		Load data byte	X	0	1	1	
0xB0	Arbitration lost in SLA+R/W as Master; own SLA+R has been received; ACK has been returned	Load data byte or	X	0	1	0	Last data byte will be transmitted and NOT ACK should be received Data byte will be transmitted and ACK should be received
		Load data byte	X	0	1	1	
0xB8	Data byte in TWDR has been transmitted; ACK has been received	Load data byte or	X	0	1	0	Last data byte will be transmitted and NOT ACK should be received Data byte will be transmitted and ACK should be received
		Load data byte	X	0	1	1	
0xC0	Data byte in TWDR has been transmitted; NOT ACK has been received	No TWDR action or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1" Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
		No TWDR action or	0	0	1	1	
		No TWDR action or	1	0	1	0	
		No TWDR action	1	0	1	1	
0xC8	Last data byte in TWDR has been transmitted (TWEA = "0"); ACK has been received	No TWDR action or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1" Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
		No TWDR action or	0	0	1	1	
		No TWDR action or	1	0	1	0	
		No TWDR action	1	0	1	1	

Figure 21-18. Formats and States in the Slave Transmitter Mode



21.7.5 Miscellaneous States

There are two status codes that do not correspond to a defined TWI state, see [Table 21-6](#).

Status 0xF8 indicates that no relevant information is available because the TWINT Flag is not set. This occurs between other states, and when the TWI is not involved in a serial transfer.

Status 0x00 indicates that a bus error has occurred during a 2-wire Serial Bus transfer. A bus error occurs when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. When a bus error occurs, TWINT is set. To recover from a bus error, the TWSTO Flag must set and TWINT must be cleared by writing a logic one to it. This causes the TWI to enter the not addressed Slave mode and to clear the TWSTO Flag (no other bits in TWCR are affected). The SDA and SCL lines are released, and no STOP condition is transmitted.

Table 21-6. Miscellaneous States

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0xF8	No relevant state information available; TWINT = "0"	No TWDR action	No TWCR action				Wait or proceed current transfer
0x00	Bus error due to an illegal START or STOP condition	No TWDR action	0	1	1	X	Only the internal hardware is affected, no STOP condition is sent on the bus. In all cases, the bus is released and TWSTO is cleared.

21.7.6 Combining Several TWI Modes

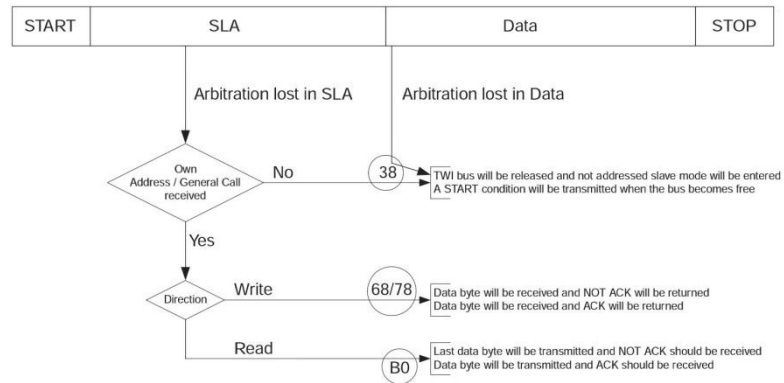
In some cases, several TWI modes must be combined in order to complete the desired action. Consider for example reading data from a serial EEPROM. Typically, such a transfer involves the following steps:

1. The transfer must be initiated.
2. The EEPROM must be instructed what location should be read.
3. The reading must be performed.
4. The transfer must be finished.

- Two or more masters are accessing different slaves. In this case, arbitration will occur in the SLA bits. Masters trying to output a one on SDA while another Master outputs a zero will lose the arbitration. Masters losing arbitration in SLA will switch to Slave mode to check if they are being addressed by the winning Master. If addressed, they will switch to SR or ST mode, depending on the value of the READ/WRITE bit. If they are not being addressed, they will switch to not addressed Slave mode or wait until the bus is free and transmit a new START condition, depending on application software action.

This is summarized in Figure 21-21. Possible status values are given in circles.

Figure 21-21. Possible Status Codes Caused by Arbitration



21.9 Register Description

21.9.1 TWBR – TWI Bit Rate Register

Bit	7	6	5	4	3	2	1	0	
(0xBB)	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0	TWBR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7..0 – TWI Bit Rate Register

TWBR selects the division factor for the bit rate generator. The bit rate generator is a frequency divider which generates the SCL clock frequency in the Master modes. See “Bit Rate Generator Unit” on page 221 for calculating bit rates.

21.9.2 TWCR – TWI Control Register

Bit	7	6	5	4	3	2	1	0	
(0xBC)	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The TWCR is used to control the operation of the TWI. It is used to enable the TWI, to initiate a Master access by applying a START condition to the bus, to generate a Receiver acknowledge, to generate a stop condition, and to control halting of the bus while the data to be written to the bus are written to the TWDR. It also indicates a write collision if data is attempted written to TWDR while the register is inaccessible.

- **Bit 7 – TWINT: TWI Interrupt Flag**

This bit is set by hardware when the TWI has finished its current job and expects application software response. If the I-bit in SREG and TWIE in TWCR are set, the MCU will jump to the TWI Interrupt Vector. While the TWINT Flag is set, the SCL low period is stretched. The TWINT Flag must be cleared by software by writing a logic one to it. Note that this flag is not automatically cleared by hardware when executing the interrupt routine. Also note that clearing this flag starts the operation of the TWI, so all accesses to the TWI Address Register (TWAR), TWI Status Register (TWSR), and TWI Data Register (TWDR) must be complete before clearing this flag.

- **Bit 6 – TWEA: TWI Enable Acknowledge Bit**

The TWEA bit controls the generation of the acknowledge pulse. If the TWEA bit is written to one, the ACK pulse is generated on the TWI bus if the following conditions are met:

1. The device's own slave address has been received.
2. A general call has been received, while the TWGCE bit in the TWAR is set.
3. A data byte has been received in Master Receiver or Slave Receiver mode.

By writing the TWEA bit to zero, the device can be virtually disconnected from the 2-wire Serial Bus temporarily. Address recognition can then be resumed by writing the TWEA bit to one again.

- **Bit 5 – TWSTA: TWI START Condition Bit**

The application writes the TWSTA bit to one when it desires to become a Master on the 2-wire Serial Bus. The TWI hardware checks if the bus is available, and generates a START condition on the bus if it is free. However, if the bus is not free, the TWI waits until a STOP condition is detected, and then generates a new START condition to claim the bus Master status. TWSTA must be cleared by software when the START condition has been transmitted.

- **Bit 4 – TWSTO: TWI STOP Condition Bit**

Writing the TWSTO bit to one in Master mode will generate a STOP condition on the 2-wire Serial Bus. When the STOP condition is executed on the bus, the TWSTO bit is cleared automatically. In Slave mode, setting the TWSTO bit can be used to recover from an error condition. This will not generate a STOP condition, but the TWI returns to a well-defined unaddressed Slave mode and releases the SCL and SDA lines to a high impedance state.

- **Bit 3 – TWWC: TWI Write Collision Flag**

The TWWC bit is set when attempting to write to the TWI Data Register – TWDR when TWINT is low. This flag is cleared by writing the TWDR Register when TWINT is high.

- **Bit 2 – TWEN: TWI Enable Bit**

The TWEN bit enables TWI operation and activates the TWI interface. When TWEN is written to one, the TWI takes control over the I/O pins connected to the SCL and SDA pins, enabling the slew-rate limiters and spike filters. If this bit is written to zero, the TWI is switched off and all TWI transmissions are terminated, regardless of any ongoing operation.

- **Bit 1 – Res: Reserved Bit**

This bit is a reserved bit and will always read as zero.

- **Bit 0 – TWIE: TWI Interrupt Enable**

When this bit is written to one, and the I-bit in SREG is set, the TWI interrupt request will be activated for as long as the TWINT Flag is high.

21.9.3 TWSR – TWI Status Register

Bit (0xB9)	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	TWSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	1	1	1	1	1	0	0	0	

- **Bits 7..3 – TWS: TWI Status**

These 5 bits reflect the status of the TWI logic and the 2-wire Serial Bus. The different status codes are described later in this section. Note that the value read from TWSR contains both the 5-bit status value and the 2-bit prescaler value. The application designer should mask the prescaler bits to zero when checking the Status bits. This makes status checking independent of prescaler setting. This approach is used in this datasheet, unless otherwise noted.

- **Bit 2 – Res: Reserved Bit**

This bit is reserved and will always read as zero.

- **Bits 1..0 – TWPS: TWI Prescaler Bits**

These bits can be read and written, and control the bit rate prescaler.

Table 21-7. TWI Bit Rate Prescaler

TWPS1	TWPS0	Prescaler Value
0	0	1
0	1	4
1	0	16
1	1	64

To calculate bit rates, see ["Bit Rate Generator Unit" on page 221](#). The value of TWPS1..0 is used in the equation.

21.9.4 TWDR – TWI Data Register

Bit (0xBB)	7	6	5	4	3	2	1	0	
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

In Transmit mode, TWDR contains the next byte to be transmitted. In Receive mode, the TWDR contains the last byte received. It is writable while the TWI is not in the process of shifting a byte. This occurs when the TWI Interrupt Flag (TWINT) is set by hardware. Note that the Data Register cannot be initialized by the user before the first interrupt occurs. The data in TWDR remains stable as long as TWINT is set. While data is shifted out, data on the bus is simultaneously shifted in. TWDR always contains the last byte present on the bus, except after a wake up from a sleep mode by the TWI interrupt. In this case, the contents of TWDR is undefined. In the case

of a lost bus arbitration, no data is lost in the transition from Master to Slave. Handling of the ACK bit is controlled automatically by the TWI logic, the CPU cannot access the ACK bit directly.

- **Bits 7..0 – TWD: TWI Data Register**

These eight bits constitute the next data byte to be transmitted, or the latest data byte received on the 2-wire Serial Bus.

21.9.5 TWAR – TWI (Slave) Address Register

Bit	7	6	5	4	3	2	1	0	
(0xBA)	TWAR								
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	0	

The TWAR should be loaded with the 7-bit Slave address (in the seven most significant bits of TWAR) to which the TWI will respond when programmed as a Slave Transmitter or Receiver, and not needed in the Master modes. In multi master systems, TWAR must be set in masters which can be addressed as Slaves by other Masters.

The LSB of TWAR is used to enable recognition of the general call address (0x00). There is an associated address comparator that looks for the slave address (or general call address if enabled) in the received serial address. If a match is found, an interrupt request is generated.

- **Bits 7..1 – TWA: TWI (Slave) Address Register**

These seven bits constitute the slave address of the TWI unit.

- **Bit 0 – TWGCE: TWI General Call Recognition Enable Bit**

If set, this bit enables the recognition of a General Call given over the 2-wire Serial Bus.

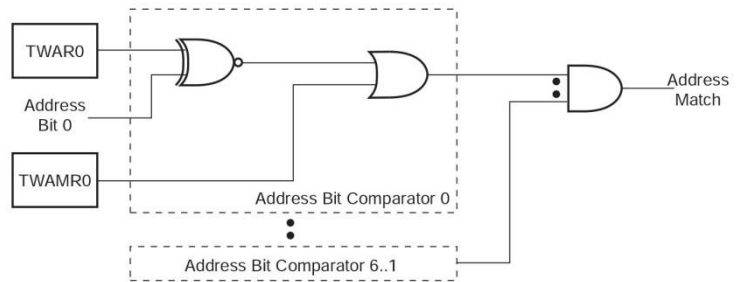
21.9.6 TWAMR – TWI (Slave) Address Mask Register

Bit	7	6	5	4	3	2	1	0	
(0xBD)	TWAMR								
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..1 – TWAM: TWI Address Mask**

The TWAMR can be loaded with a 7-bit Slave Address mask. Each of the bits in TWAMR can mask (disable) the corresponding address bits in the TWI Address Register (TWAR). If the mask bit is set to one then the address match logic ignores the compare between the incoming address bit and the corresponding bit in TWAR. Figure 21-22 shown the address match logic in detail.

Figure 21-22. TWI Address Match Logic, Block Diagram



• **Bit 0 – Res: Reserved Bit**

This bit is an unused bit in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

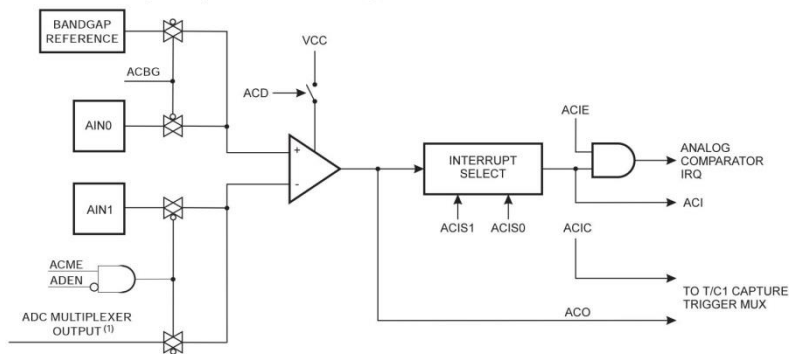
22. Analog Comparator

22.1 Overview

The Analog Comparator compares the input values on the positive pin AIN0 and negative pin AIN1. When the voltage on the positive pin AIN0 is higher than the voltage on the negative pin AIN1, the Analog Comparator output, ACO, is set. The comparator's output can be set to trigger the Timer/Counter1 Input Capture function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user can select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 22-1.

The Power Reduction ADC bit, PRADC, in "Minimizing Power Consumption" on page 42 must be disabled by writing a logical zero to be able to use the ADC input MUX.

Figure 22-1. Analog Comparator Block Diagram⁽²⁾



- Notes:
1. See Table 22-1 on page 247.
 2. Refer to Figure 1-1 on page 2 and Table 13-9 on page 88 for Analog Comparator pin placement.

22.2 Analog Comparator Multiplexed Input

It is possible to select any of the ADC7..0 pins to replace the negative input to the Analog Comparator. The ADC multiplexer is used to select this input, and consequently, the ADC must be switched off to utilize this feature. If the Analog Comparator Multiplexer Enable bit (ACME in ADCSRB) is set and the ADC is switched off (ADEN in ADCSRA is zero), MUX2..0 in ADMUX select the input pin to replace the negative input to the Analog Comparator, as shown in Table 22-1. If ACME is cleared or ADEN is set, AIN1 is applied to the negative input to the Analog Comparator

Table 22-1. Analog Comparator Multiplexed Input

ACME	ADEN	MUX2..0	Analog Comparator Negative Input
0	x	xxx	AIN1
1	1	xxx	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

22.3 Register Description

22.3.1 ADCSRB – ADC Control and Status Register B

Bit (0x7B)	7	6	5	4	3	2	1	0	
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	ADCSRB
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – ACME: Analog Comparator Multiplexer Enable**

When this bit is written logic one and the ADC is switched off (ADEN in ADCSRA is zero), the ADC multiplexer selects the negative input to the Analog Comparator. When this bit is written logic zero, AIN1 is applied to the negative input of the Analog Comparator. For a detailed description of this bit, see "Analog Comparator Multiplexed Input" on page 246.

22.3.2 ACSR – Analog Comparator Control and Status Register

Bit 0x30 (0x50)	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	ACSR
Initial Value	0	0	N/A	0	0	0	0	0	

- **Bit 7 – ACD: Analog Comparator Disable**

When this bit is written logic one, the power to the Analog Comparator is switched off. This bit can be set at any time to turn off the Analog Comparator. This will reduce power consumption in Active and Idle mode. When changing the ACD bit, the Analog Comparator Interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit is changed.

- **Bit 6 – ACBG: Analog Comparator Bandgap Select**

When this bit is set, a fixed bandgap reference voltage replaces the positive input to the Analog Comparator. When this bit is cleared, AIN0 is applied to the positive input of the Analog Comparator. When the bandgap reference is used as input to the Analog Comparator, it will take a

certain time for the voltage to stabilize. If not stabilized, the first conversion may give a wrong value. See "Internal Voltage Reference" on page 49

- **Bit 5 – ACO: Analog Comparator Output**

The output of the Analog Comparator is synchronized and then directly connected to ACO. The synchronization introduces a delay of 1 - 2 clock cycles.

- **Bit 4 – ACI: Analog Comparator Interrupt Flag**

This bit is set by hardware when a comparator output event triggers the interrupt mode defined by ACIS1 and ACIS0. The Analog Comparator interrupt routine is executed if the ACIE bit is set and the I-bit in SREG is set. ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

- **Bit 3 – ACIE: Analog Comparator Interrupt Enable**

When the ACIE bit is written logic one and the I-bit in the Status Register is set, the Analog Comparator interrupt is activated. When written logic zero, the interrupt is disabled.

- **Bit 2 – ACIC: Analog Comparator Input Capture Enable**

When written logic one, this bit enables the input capture function in Timer/Counter1 to be triggered by the Analog Comparator. The comparator output is in this case directly connected to the input capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When written logic zero, no connection between the Analog Comparator and the input capture function exists. To make the comparator trigger the Timer/Counter1 Input Capture interrupt, the ICIE1 bit in the Timer Interrupt Mask Register (TIMSK1) must be set.

- **Bits 1, 0 – ACIS1, ACIS0: Analog Comparator Interrupt Mode Select**

These bits determine which comparator events that trigger the Analog Comparator interrupt. The different settings are shown in Table 22-2.

Table 22-2. ACIS1/ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator Interrupt on Output Toggle.
0	1	Reserved
1	0	Comparator Interrupt on Falling Output Edge.
1	1	Comparator Interrupt on Rising Output Edge.

When changing the ACIS1/ACIS0 bits, the Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR Register. Otherwise an interrupt can occur when the bits are changed.

22.3.3 DIDR1 – Digital Input Disable Register 1

Bit	7	6	5	4	3	2	1	0	
(0x7F)	-	-	-	-	-	-	AIN1D	AIN0D	DIDR1
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..2 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 1, 0 – AIN1D, AIN0D: AIN1, AIN0 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the AIN1/0 pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to the AIN1/0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

23. Analog-to-Digital Converter

23.1 Features

- 10-bit Resolution
- 0.5 LSB Integral Non-linearity
- ± 2 LSB Absolute Accuracy
- 13 - 260 μ s Conversion Time
- Up to 76.9 kSPS (Up to 15 kSPS at Maximum Resolution)
- 6 Multiplexed Single Ended Input Channels
- 2 Additional Multiplexed Single Ended Input Channels (TQFP and QFN/MLF Package only)
- Temperature Sensor Input Channel
- Optional Left Adjustment for ADC Result Readout
- 0 - V_{CC} ADC Input Voltage Range
- Selectable 1.1V ADC Reference Voltage
- Free Running or Single Conversion Mode
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

23.2 Overview

The ATmega48PA/88PA/168PA/328P features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer which allows eight single-ended voltage inputs constructed from the pins of Port A. The single-ended voltage inputs refer to 0V (GND).

The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in [Figure 23-1 on page 251](#).

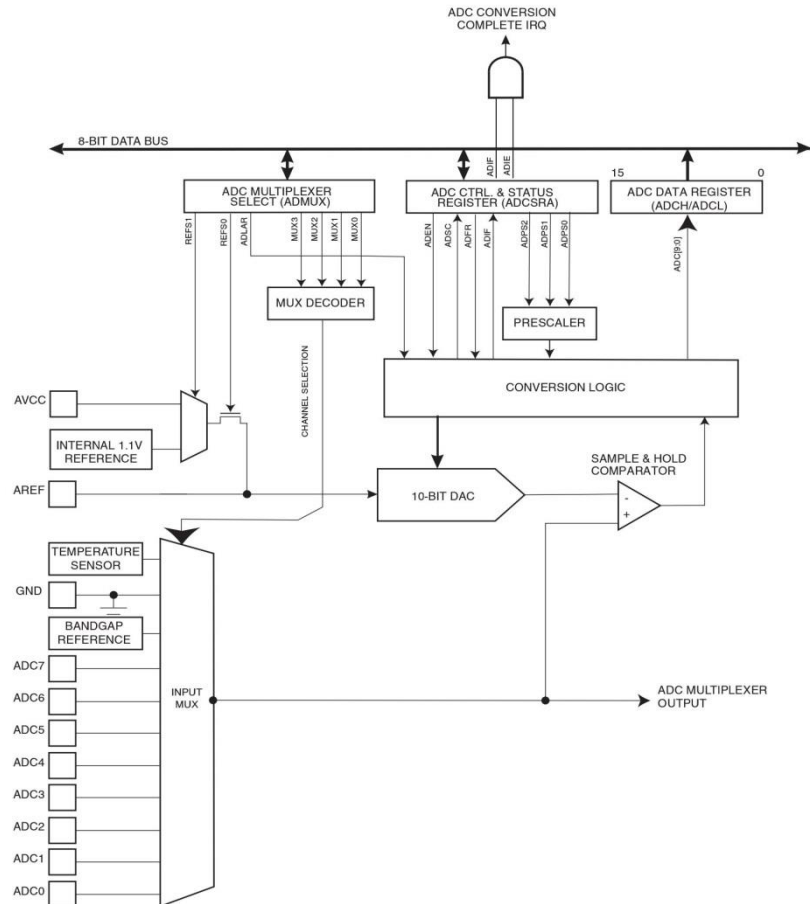
The ADC has a separate analog supply voltage pin, AV_{CC} . AV_{CC} must not differ more than $\pm 0.3V$ from V_{CC} . See the paragraph "[ADC Noise Canceler](#)" on [page 256](#) on how to connect this pin.

Internal reference voltages of nominally 1.1V or AV_{CC} are provided On-chip. The voltage reference may be externally decoupled at the AREF pin by a capacitor for better noise performance.

The Power Reduction ADC bit, PRADC, in "[Minimizing Power Consumption](#)" on [page 42](#) must be disabled by writing a logical zero to enable the ADC.

The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1 LSB. Optionally, AV_{CC} or an internal 1.1V reference voltage may be connected to the AREF pin by writing to the REFSn bits in the ADMUX Register. The internal voltage reference may thus be decoupled by an external capacitor at the AREF pin to improve noise immunity.

Figure 23-1. Analog to Digital Converter Block Schematic Operation,



The analog input channel is selected by writing to the MUX bits in ADMUX. Any of the ADC input pins, as well as GND and a fixed bandgap voltage reference, can be selected as single ended inputs to the ADC. The ADC is enabled by setting the ADC Enable bit, ADEN in ADCSRA. Voltage reference and input channel selections will not go into effect until ADEN is set. The ADC does not consume power when ADEN is cleared, so it is recommended to switch off the ADC before entering power saving sleep modes.

The ADC generates a 10-bit result which is presented in the ADC Data Registers, ADCH and ADCL. By default, the result is presented right adjusted, but can optionally be presented left adjusted by setting the ADLAR bit in ADMUX.

If the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH, to ensure that the content of the Data Registers belongs to the same conversion. Once ADCL is read, ADC access to Data Registers is blocked. This means that if ADCL has been read, and a conversion completes before ADCH is

read, neither register is updated and the result from the conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL Registers is re-enabled.

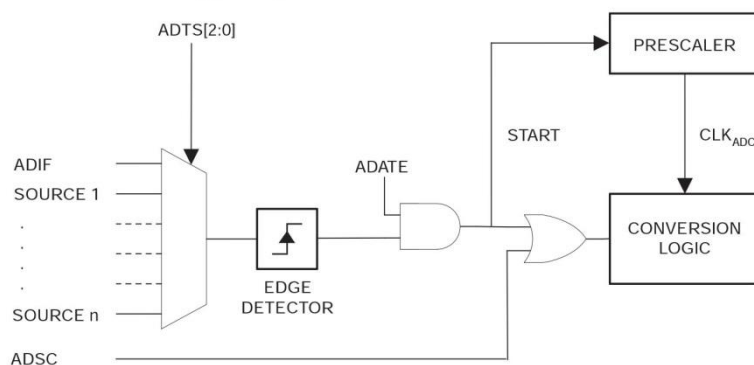
The ADC has its own interrupt which can be triggered when a conversion completes. When ADC access to the Data Registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result is lost.

23.3 Starting a Conversion

A single conversion is started by disabling the Power Reduction ADC bit, PRADC, in “Minimizing Power Consumption” on page 42 by writing a logical zero to it and writing a logical one to the ADC Start Conversion bit, ADSC. This bit stays high as long as the conversion is in progress and will be cleared by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

Alternatively, a conversion can be triggered automatically by various sources. Auto Triggering is enabled by setting the ADC Auto Trigger Enable bit, ADATE in ADCSRA. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB (See description of the ADTS bits for a list of the trigger sources). When a positive edge occurs on the selected trigger signal, the ADC prescaler is reset and a conversion is started. This provides a method of starting conversions at fixed intervals. If the trigger signal still is set when the conversion completes, a new conversion will not be started. If another positive edge occurs on the trigger signal during conversion, the edge will be ignored. Note that an Interrupt Flag will be set even if the specific interrupt is disabled or the Global Interrupt Enable bit in SREG is cleared. A conversion can thus be triggered without causing an interrupt. However, the Interrupt Flag must be cleared in order to trigger a new conversion at the next interrupt event.

Figure 23-2. ADC Auto Trigger Logic

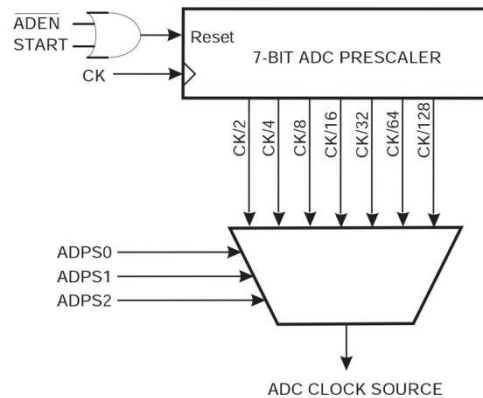


Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a logical one to the ADSC bit in ADCSRA. In this mode the ADC will perform successive conversions independently of whether the ADC Interrupt Flag, ADIF is cleared or not.

If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

23.4 Prescaling and Conversion Timing

Figure 23-3. ADC Prescaler



By default, the successive approximation circuitry requires an input clock frequency between 50 kHz and 200 kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200 kHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100 kHz. The prescaling is set by the ADPS bits in ADCSRA. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a single ended conversion by setting the ADSC bit in ADCSRA, the conversion starts at the following rising edge of the ADC clock cycle.

A normal conversion takes 13 ADC clock cycles. The first conversion after the ADC is switched on (ADEN in ADCSRA is set) takes 25 ADC clock cycles in order to initialize the analog circuitry.

When the bandgap reference voltage is used as input to the ADC, it will take a certain time for the voltage to stabilize. If not stabilized, the first value read after the first conversion may be wrong.

The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 13.5 ADC clock cycles after the start of an first conversion. When a conversion is complete, the result is written to the ADC Data Registers, and ADIF is set. In Single Conversion mode, ADSC is cleared simultaneously. The software may then set ADSC again, and a new conversion will be initiated on the first rising ADC clock edge.

When Auto Triggering is used, the prescaler is reset when the trigger event occurs. This assures a fixed delay from the trigger event to the start of conversion. In this mode, the sample-and-hold takes place two ADC clock cycles after the rising edge on the trigger source signal. Three additional CPU clock cycles are used for synchronization logic.

In Free Running mode, a new conversion will be started immediately after the conversion completes, while ADSC remains high. For a summary of conversion times, see [Table 23-1 on page 255](#).

Figure 23-4. ADC Timing Diagram, First Conversion (Single Conversion Mode)

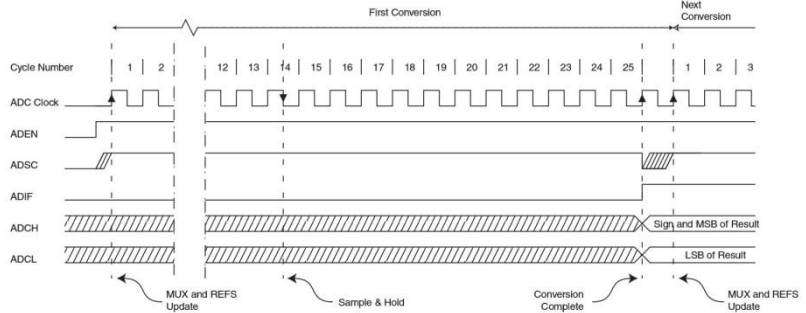


Figure 23-5. ADC Timing Diagram, Single Conversion

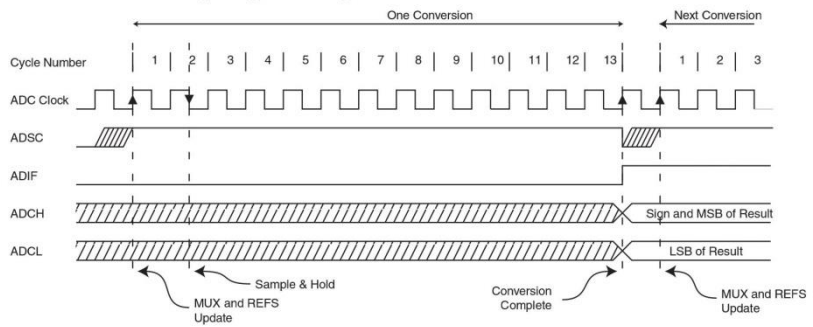
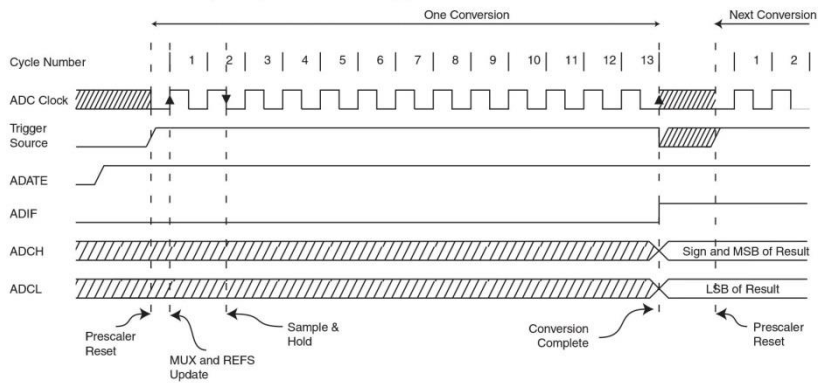


Figure 23-6. ADC Timing Diagram, Auto Triggered Conversion



23.5.1 ADC Input Channels

When changing channel selections, the user should observe the following guidelines to ensure that the correct channel is selected:

In Single Conversion mode, always select the channel before starting the conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the conversion to complete before changing the channel selection.

In Free Running mode, always select the channel before starting the first conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the first conversion to complete, and then change the channel selection. Since the next conversion has already started automatically, the next result will reflect the previous channel selection. Subsequent conversions will reflect the new channel selection.

23.5.2 ADC Voltage Reference

The reference voltage for the ADC (V_{REF}) indicates the conversion range for the ADC. Single ended channels that exceed V_{REF} will result in codes close to 0x3FF. V_{REF} can be selected as either AV_{CC} , internal 1.1V reference, or external AREF pin.

AV_{CC} is connected to the ADC through a passive switch. The internal 1.1V reference is generated from the internal bandgap reference (V_{BG}) through an internal amplifier. In either case, the external AREF pin is directly connected to the ADC, and the reference voltage can be made more immune to noise by connecting a capacitor between the AREF pin and ground. V_{REF} can also be measured at the AREF pin with a high impedance voltmeter. Note that V_{REF} is a high impedance source, and only a capacitive load should be connected in a system.

If the user has a fixed voltage source connected to the AREF pin, the user may not use the other reference voltage options in the application, as they will be shorted to the external voltage. If no external voltage is applied to the AREF pin, the user may switch between AV_{CC} and 1.1V as reference selection. The first ADC conversion result after switching reference voltage source may be inaccurate, and the user is advised to discard this result.

23.6 ADC Noise Canceler

The ADC features a noise canceler that enables conversion during sleep mode to reduce noise induced from the CPU core and other I/O peripherals. The noise canceler can be used with ADC Noise Reduction and Idle mode. To make use of this feature, the following procedure should be used:

- a. Make sure that the ADC is enabled and is not busy converting. Single Conversion mode must be selected and the ADC conversion complete interrupt must be enabled.
- b. Enter ADC Noise Reduction mode (or Idle mode). The ADC will start a conversion once the CPU has been halted.
- c. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the CPU and execute the ADC Conversion Complete interrupt routine. If another interrupt wakes up the CPU before the ADC conversion is complete, that interrupt will be executed, and an ADC Conversion Complete interrupt request will be generated when the ADC conversion completes. The CPU will remain in active mode until a new sleep command is executed.

Note that the ADC will not be automatically turned off when entering other sleep modes than Idle mode and ADC Noise Reduction mode. The user is advised to write zero to ADEN before entering such sleep modes to avoid excessive power consumption.

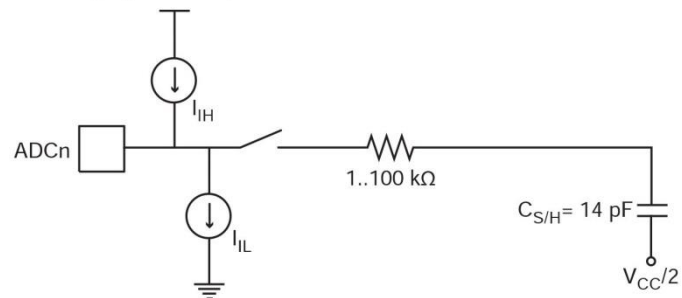
23.6.1 Analog Input Circuitry

The analog input circuitry for single ended channels is illustrated in Figure 23-8. An analog source applied to ADCn is subjected to the pin capacitance and input leakage of that pin, regardless of whether that channel is selected as input for the ADC. When the channel is selected, the source must drive the S/H capacitor through the series resistance (combined resistance in the input path).

The ADC is optimized for analog signals with an output impedance of approximately 10 k Ω or less. If such a source is used, the sampling time will be negligible. If a source with higher impedance is used, the sampling time will depend on how long time the source needs to charge the S/H capacitor, which can vary widely. The user is recommended to only use low impedance sources with slowly varying signals, since this minimizes the required charge transfer to the S/H capacitor.

Signal components higher than the Nyquist frequency ($f_{ADC}/2$) should not be present for either kind of channels, to avoid distortion from unpredictable signal convolution. The user is advised to remove high frequency components with a low-pass filter before applying the signals as inputs to the ADC.

Figure 23-8. Analog Input Circuitry



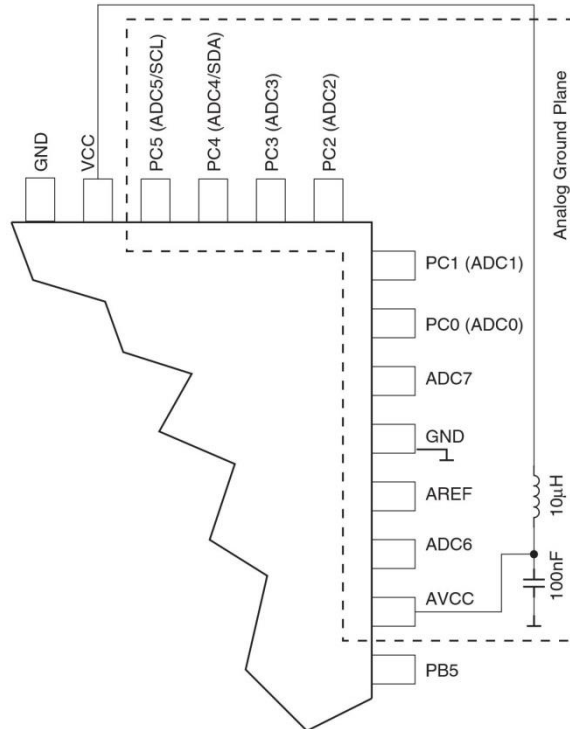
23.6.2 Analog Noise Canceling Techniques

Digital circuitry inside and outside the device generates EMI which might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

- Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.
- The AV_{CC} pin on the device should be connected to the digital V_{CC} supply voltage via an LC network as shown in Figure 23-9.
- Use the ADC noise canceler function to reduce induced noise from the CPU.
- If any ADC [3..0] port pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress. However, using the 2-wire Interface (ADC4

and ADC5) will only affect the conversion on ADC4 and ADC5 and not the other ADC channels.

Figure 23-9. ADC Power Connections



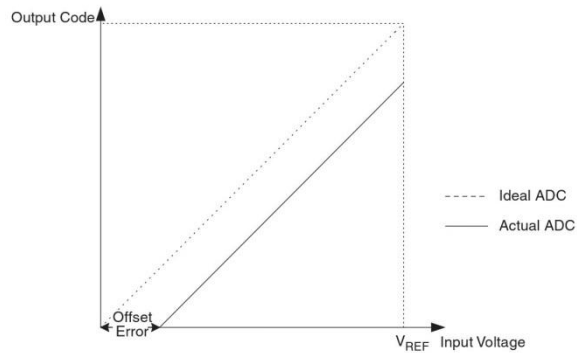
23.6.3 ADC Accuracy Definitions

An n-bit single-ended ADC converts a voltage linearly between GND and V_{REF} in 2^n steps (LSBs). The lowest code is read as 0, and the highest code is read as $2^n - 1$.

Several parameters describe the deviation from the ideal behavior:

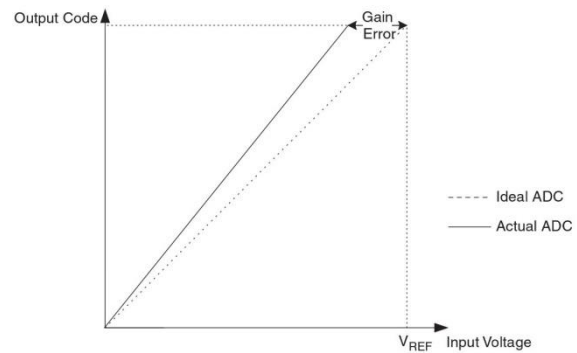
- Offset: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.

Figure 23-10. Offset Error



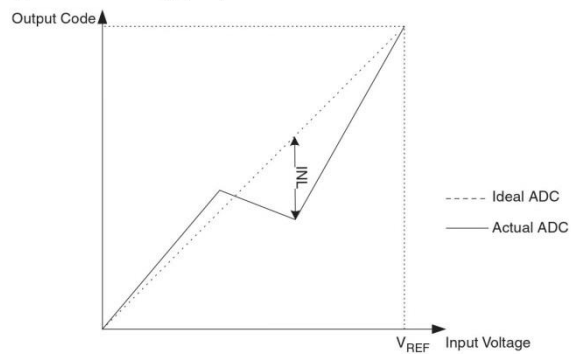
- Gain error: After adjusting for offset, the gain error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition (at 1.5 LSB below maximum). Ideal value: 0 LSB

Figure 23-11. Gain Error



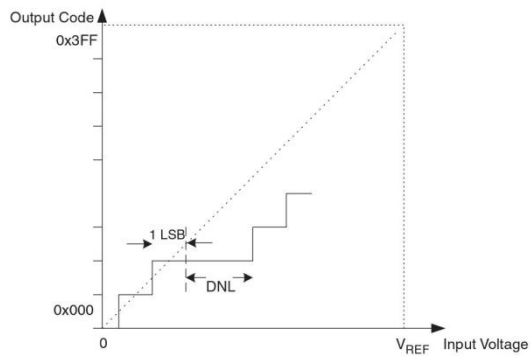
- Integral Non-linearity (INL): After adjusting for offset and gain error, the INL is the maximum deviation of an actual transition compared to an ideal transition for any code. Ideal value: 0 LSB.

Figure 23-12. Integral Non-linearity (INL)



- Differential Non-linearity (DNL): The maximum deviation of the actual code width (the interval between two adjacent transitions) from the ideal code width (1 LSB). Ideal value: 0 LSB.

Figure 23-13. Differential Non-linearity (DNL)



- Quantization Error: Due to the quantization of the input voltage into a finite number of codes, a range of input voltages (1 LSB wide) will code to the same value. Always ± 0.5 LSB.
- Absolute accuracy: The maximum deviation of an actual (unadjusted) transition compared to an ideal transition for any code. This is the compound effect of offset, gain error, differential error, non-linearity, and quantization error. Ideal value: ± 0.5 LSB.

23.7 ADC Conversion Result

After the conversion is complete (ADIF is high), the conversion result can be found in the ADC Result Registers (ADCL, ADCH).

For single ended conversion, the result is

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

where V_{IN} is the voltage on the selected input pin and V_{REF} the selected voltage reference (see [Table 23-3 on page 262](#) and [Table 23-4 on page 263](#)). 0x000 represents analog ground, and 0x3FF represents the selected reference voltage minus one LSB.

23.8 Temperature Measurement

The temperature measurement is based on an on-chip temperature sensor that is coupled to a single ended ADC8 channel. Selecting the ADC8 channel by writing the MUX3..0 bits in ADMUX register to "1000" enables the temperature sensor. The internal 1.1V voltage reference must also be selected for the ADC voltage reference source in the temperature sensor measurement. When the temperature sensor is enabled, the ADC converter can be used in single conversion mode to measure the voltage over the temperature sensor.

The measured voltage has a linear relationship to the temperature as described in [Table 23-2](#). The voltage sensitivity is approximately 1 mV/°C and the accuracy of the temperature measurement is +/- 10°C.

Table 23-2. Temperature vs. Sensor Output Voltage (Typical Case)

Temperature / °C	-45°C	+25°C	+85°C
Voltage / mV	242 mV	314 mV	380 mV

The values described in [Table 23-2](#) are typical values. However, due to the process variation the temperature sensor output voltage varies from one chip to another. To be capable of achieving more accurate results the temperature measurement can be calibrated in the application software. The software calibration requires that a calibration value is measured and stored in a register or EEPROM for each chip, as a part of the production test. The software calibration can be done utilizing the formula:

$$T = \{ [(ADCH \ll 8) | ADCL] - T_{OS} \} / k$$

where ADCn are the ADC data registers, k is a fixed coefficient and T_{OS} is the temperature sensor offset value determined and stored into EEPROM as a part of the production test.

23.9 Register Description

23.9.1 ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – REFS1:0: Reference Selection Bits**

These bits select the voltage reference for the ADC, as shown in [Table 23-3](#). If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.

Table 23-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

- **Bit 5 – ADLAR: ADC Left Adjust Result**

The ADLAR bit affects the presentation of the ADC conversion result in the ADC Data Register. Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the ADLAR bit will affect the ADC Data Register immediately, regardless of any ongoing conversions. For a complete description of this bit, see [“ADCL and ADCH – The ADC Data Register”](#) on [page 265](#).

- **Bit 4 – Res: Reserved Bit**

This bit is an unused bit in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bits 3:0 – MUX3:0: Analog Channel Selection Bits**

The value of these bits selects which analog inputs are connected to the ADC. See [Table 23-4](#) for details. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

Table 23-4. Input Channel Selections

MUX3..0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8 ⁽¹⁾
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V _{BG})
1111	0V (GND)

Note: 1. For Temperature Sensor.

23.9.2 ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 6 – ADSC: ADC Start Conversion**

In Single Conversion mode, write this bit to one to start each conversion. In Free Running mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

When this bit is written to one, Auto Triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the Data Registers are updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a Read-Modify-Write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

- **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.

- **Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits**

These bits determine the division factor between the system clock frequency and the input clock to the ADC.

Table 23-5. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

23.9.3 ADCL and ADCH – The ADC Data Register

23.9.3.1 ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

23.9.3.2 ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	ADC1	ADC0	–	–	–	–	–	–	ADCL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers.

When ADCL is read, the ADC Data Register is not updated until ADCH is read. Consequently, if the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH.

The ADLAR bit in ADMUX, and the MUXn bits in ADMUX affect the way the result is read from the registers. If ADLAR is set, the result is left adjusted. If ADLAR is cleared (default), the result is right adjusted.

- **ADC9:0: ADC Conversion Result**

These bits represent the result from the conversion, as detailed in “ADC Conversion Result” on page 261.

23.9.4 ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	ADCSR B
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 5:3 – Res: Reserved Bits**

These bits are reserved for future use. To ensure compatibility with future devices, these bits must be written to zero when ADCSRB is written.

- **Bit 2:0 – ADTS2:0: ADC Auto Trigger Source**

If ADATE in ADCSRA is written to one, the value of these bits selects which source will trigger an ADC conversion. If ADATE is cleared, the ADTS2:0 settings will have no effect. A conversion will be triggered by the rising edge of the selected Interrupt Flag. Note that switching from a trigger source that is cleared to a trigger source that is set, will generate a positive edge on the

trigger signal. If ADEN in ADCSRA is set, this will start a conversion. Switching to Free Running mode (ADTS[2:0]=0) will not cause a trigger event, even if the ADC Interrupt Flag is set.

Table 23-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

23.9.5 DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
(0x7E)	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – Res: Reserved Bits**

These bits are reserved for future use. To ensure compatibility with future devices, these bits must be written to zero when DIDR0 is written.

- **Bit 5:0 – ADC5D..ADC0D: ADC5..0 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding ADC pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to the ADC5..0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

Note that ADC pins ADC7 and ADC6 do not have digital input buffers, and therefore do not require Digital Input Disable bits.

24. debugWIRE On-chip Debug System

24.1 Features

- Complete Program Flow Control
- Emulates All On-chip Functions, Both Digital and Analog, except RESET Pin
- Real-time Operation
- Symbolic Debugging Support (Both at C and Assembler Source Level, or for Other HLLs)
- Unlimited Number of Program Break Points (Using Software Break Points)
- Non-intrusive Operation
- Electrical Characteristics Identical to Real Device
- Automatic Configuration System
- High-Speed Operation
- Programming of Non-volatile Memories

24.2 Overview

The debugWIRE On-chip debug system uses a One-wire, bi-directional interface to control the program flow, execute AVR instructions in the CPU and to program the different non-volatile memories.

24.3 Physical Interface

When the debugWIRE Enable (DWEN) Fuse is programmed and Lock bits are unprogrammed, the debugWIRE system within the target device is activated. The RESET port pin is configured as a wire-AND (open-drain) bi-directional I/O pin with pull-up enabled and becomes the communication gateway between target and emulator.

Figure 24-1. The debugWIRE Setup

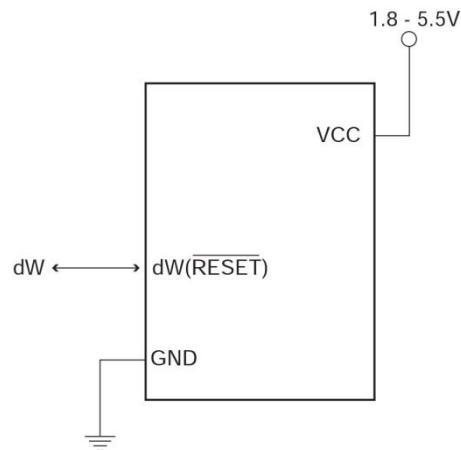


Figure 24-1 shows the schematic of a target MCU, with debugWIRE enabled, and the emulator connector. The system clock is not affected by debugWIRE and will always be the clock source selected by the CKSEL Fuses.

When designing a system where debugWIRE will be used, the following observations must be made for correct operation:

- Pull-up resistors on the dW/(RESET) line must not be smaller than 10kΩ. The pull-up resistor is not required for debugWIRE functionality.
- Connecting the RESET pin directly to V_{CC} will not work.
- Capacitors connected to the RESET pin must be disconnected when using debugWire.
- All external reset sources must be disconnected.

24.4 Software Break Points

debugWIRE supports Program memory Break Points by the AVR Break instruction. Setting a Break Point in AVR Studio® will insert a BREAK instruction in the Program memory. The instruction replaced by the BREAK instruction will be stored. When program execution is continued, the stored instruction will be executed before continuing from the Program memory. A break can be inserted manually by putting the BREAK instruction in the program.

The Flash must be re-programmed each time a Break Point is changed. This is automatically handled by AVR Studio through the debugWIRE interface. The use of Break Points will therefore reduce the Flash Data retention. Devices used for debugging purposes should not be shipped to end customers.

24.5 Limitations of debugWIRE

The debugWIRE communication pin (dW) is physically located on the same pin as External Reset (RESET). An External Reset source is therefore not supported when the debugWIRE is enabled.

A programmed DWEN Fuse enables some parts of the clock system to be running in all sleep modes. This will increase the power consumption while in sleep. Thus, the DWEN Fuse should be disabled when debugWire is not used.

24.6 Register Description

The following section describes the registers used with the debugWire.

24.6.1 DWDR – debugWire Data Register

Bit	7	6	5	4	3	2	1	0	
	DWDR[7:0]								DWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The DWDR Register provides a communication channel from the running program in the MCU to the debugger. This register is only accessible by the debugWIRE and can therefore not be used as a general purpose register in the normal operations.

25. Self-Programming the Flash, ATmega48PA

25.1 Overview

In ATmega48PA, there is no Read-While-Write support, and no separate Boot Loader Section. The SPM instruction can be executed from the entire Flash.

The device provides a Self-Programming mechanism for downloading and uploading program code by the MCU itself. The Self-Programming can use any available data interface and associated protocol to read code and write (program) that code into the Program memory.

The Program memory is updated in a page by page fashion. Before programming a page with the data stored in the temporary page buffer, the page must be erased. The temporary page buffer is filled one word at a time using SPM and the buffer can be filled either before the Page Erase command or between a Page Erase and a Page Write operation:

Alternative 1, fill the buffer before a Page Erase

- Fill temporary page buffer
- Perform a Page Erase
- Perform a Page Write

Alternative 2, fill the buffer after Page Erase

- Perform a Page Erase
- Fill temporary page buffer
- Perform a Page Write

If only a part of the page needs to be changed, the rest of the page must be stored (for example in the temporary page buffer) before the erase, and then be re-written. When using alternative 1, the Boot Loader provides an effective Read-Modify-Write feature which allows the user software to first read the page, do the necessary changes, and then write back the modified data. If alternative 2 is used, it is not possible to read the old data while loading since the page is already erased. The temporary page buffer can be accessed in a random sequence. It is essential that the page address used in both the Page Erase and Page Write operation is addressing the same page.

25.1.1 Performing Page Erase by SPM

To execute Page Erase, set up the address in the Z-pointer, write "00000011" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE in the Z-register. Other bits in the Z-pointer will be ignored during this operation.

- The CPU is halted during the Page Erase operation.

25.1.2 Filling the Temporary Buffer (Page Loading)

To write an instruction word, set up the address in the Z-pointer and data in R1:R0, write "00000001" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The content of PCWORD in the Z-register is used to address the data in the temporary buffer. The temporary buffer will auto-erase after a Page Write operation or by writing the RWWSRE bit in SPMCSR. It is also erased after a system reset. Note that it is not possible to write more than one time to each address without erasing the temporary buffer.

If the EEPROM is written in the middle of an SPM Page Load operation, all data loaded will be lost.

25.1.3 Performing a Page Write

To execute Page Write, set up the address in the Z-pointer, write "00000101" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE. Other bits in the Z-pointer must be written to zero during this operation.

- The CPU is halted during the Page Write operation.

25.2 Addressing the Flash During Self-Programming

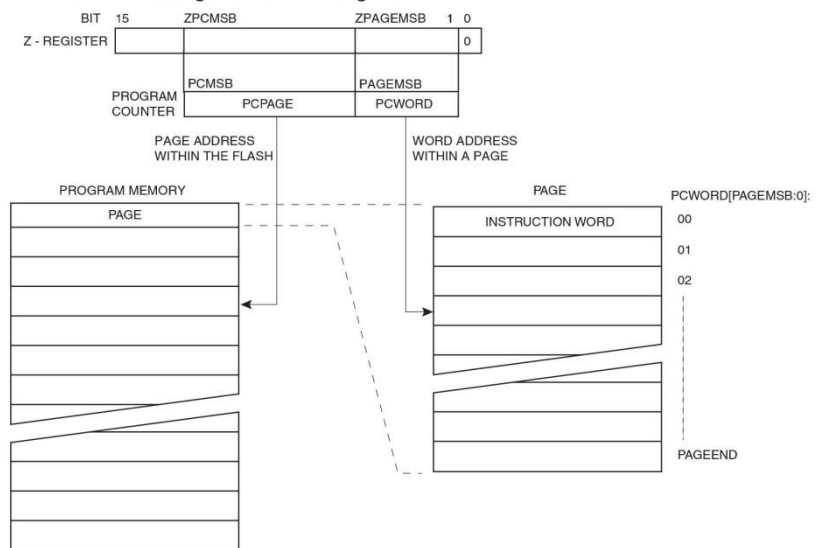
The Z-pointer is used to address the SPM commands.

Bit	15	14	13	12	11	10	9	8
ZH (R31)	Z15	Z14	Z13	Z12	Z11	Z10	Z9	Z8
ZL (R30)	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0
	7	6	5	4	3	2	1	0

Since the Flash is organized in pages (see [Table 27-11 on page 299](#)), the Program Counter can be treated as having two different sections. One section, consisting of the least significant bits, is addressing the words within a page, while the most significant bits are addressing the pages. This is shown in [Figure 26-3 on page 282](#). Note that the Page Erase and Page Write operations are addressed independently. Therefore it is of major importance that the software addresses the same page in both the Page Erase and Page Write operation.

The LPM instruction uses the Z-pointer to store the address. Since this instruction addresses the Flash byte-by-byte, also the LSB (bit Z0) of the Z-pointer is used.

Figure 25-1. Addressing the Flash During SPM⁽¹⁾



Note: 1. The different variables used in [Figure 26-3](#) are listed in [Table 27-11 on page 299](#).

25.2.1 EEPROM Write Prevents Writing to SPMCSR

Note that an EEPROM write operation will block all software programming to Flash. Reading the Fuses and Lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEPE) in the EECR Register and verifies that the bit is cleared before writing to the SPMCSR Register.

25.2.2 Reading the Fuse and Lock Bits from Software

It is possible to read both the Fuse and Lock bits from software. To read the Lock bits, load the Z-pointer with 0x0001 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the BLBSET and SELFPRGEN bits are set in SPMCSR, the value of the Lock bits will be loaded in the destination register. The BLBSET and SELFPRGEN bits will auto-clear upon completion of reading the Lock bits or if no LPM instruction is executed within three CPU cycles or no SPM instruction is executed within four CPU cycles. When BLBSET and SELFPRGEN are cleared, LPM will work as described in the Instruction set Manual.

Bit	7	6	5	4	3	2	1	0
Rd	-	-	-	-	-	-	LB2	LB1

The algorithm for reading the Fuse Low byte is similar to the one described above for reading the Lock bits. To read the Fuse Low byte, load the Z-pointer with 0x0000 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Fuse Low byte (FLB) will be loaded in the destination register as shown below. See [Table 27-5 on page 296](#) for a detailed description and mapping of the Fuse Low byte.

Bit	7	6	5	4	3	2	1	0
Rd	FLB7	FLB6	FLB5	FLB4	FLB3	FLB2	FLB1	FLB0

Similarly, when reading the Fuse High byte (FHB), load 0x0003 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Fuse High byte will be loaded in the destination register as shown below. See [Table 27-5 on page 296](#) for detailed description and mapping of the Extended Fuse byte.

Bit	7	6	5	4	3	2	1	0
Rd	FHB7	FHB6	FHB5	FHB4	FHB3	FHB2	FHB1	FHB0

Similarly, when reading the Extended Fuse byte (EFB), load 0x0002 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Extended Fuse byte will be loaded in the destination register as shown below. See [Table 27-5 on page 296](#) for detailed description and mapping of the Extended Fuse byte.

Bit	7	6	5	4	3	2	1	0
Rd	FHB7	FHB6	FHB5	FHB4	FHB3	FHB2	FHB1	FHB0

Fuse and Lock bits that are programmed, will be read as zero. Fuse and Lock bits that are unprogrammed, will be read as one.

25.2.3 Preventing Flash Corruption

During periods of low V_{CC} , the Flash program can be corrupted because the supply voltage is too low for the CPU and the Flash to operate properly. These issues are the same as for board level systems using the Flash, and the same design solutions should be applied.

A Flash program corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the Flash requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

Flash corruption can easily be avoided by following these design recommendations (one is sufficient):

1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD) if the operating voltage matches the detection level. If not, an external low V_{CC} reset protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.
2. Keep the AVR core in Power-down sleep mode during periods of low V_{CC} . This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the SPMCSR Register and thus the Flash from unintentional writes.

25.2.4 Programming Time for Flash when Using SPM

The calibrated RC Oscillator is used to time Flash accesses. [Table 26-6](#) shows the typical programming time for Flash accesses from the CPU.

Table 25-1. SPM Programming Time⁽¹⁾

Symbol	Min Programming Time	Max Programming Time
Flash write (Page Erase, Page Write, and write Lock bits by SPM)	3.7 ms	4.5 ms

Note: 1. Minimum and maximum programming time is per individual operation.

25.2.5 Simple Assembly Code Example for a Boot Loader

Note that the RWWWSB bit will always be read as zero in ATmega48PA. Nevertheless, it is recommended to check this bit as shown in the code example, to ensure compatibility with devices supporting Read-While-Write.

```

;-the routine writes one page of data from RAM to Flash
; the first data location in RAM is pointed to by the Y pointer
; the first data location in Flash is pointed to by the Z-pointer
;-error handling is not included
;-the routine must be placed inside the Boot space
; (at least the Do_spm sub routine). Only code inside NRWW section can
; be read during Self-Programming (Page Erase and Page Write).
;-registers used: r0, r1, temp1 (r16), temp2 (r17), looplo (r24),
; loophi (r25), spmcrval (r20)
; storing and restoring of registers is not included in the routine
; register usage can be optimized at the expense of code size
;-It is assumed that either the interrupt table is moved to the Boot
; loader section or that the interrupts are disabled.
.equ PAGESIZEB = PAGESIZE*2 ;PAGESIZEB is page size in BYTES, not words
.org SMALLBOOTSTART
Write_page:
; Page Erase
ldi spmcrval, (1<<PGERS) | (1<<SELFPRGEN)
rcallDo_spm

; re-enable the RWW section
ldi spmcrval, (1<<RWSRE) | (1<<SELFPRGEN)
rcallDo_spm

; transfer data from RAM to Flash page buffer
ldi looplo, low(PAGESIZEB) ;init loop variable
ldi loophi, high(PAGESIZEB) ;not required for PAGESIZEB<=256
Wrloop:
ld r0, Y+
ld r1, Y+
ldi spmcrval, (1<<SELFPRGEN)
rcallDo_spm
adiw ZH:ZL, 2
sbiw loophi:looplo, 2 ;use subi for PAGESIZEB<=256
brne Wrloop

; execute Page Write
subi ZL, low(PAGESIZEB) ;restore pointer
sbci ZH, high(PAGESIZEB) ;not required for PAGESIZEB<=256
ldi spmcrval, (1<<PGWRT) | (1<<SELFPRGEN)
rcallDo_spm

; re-enable the RWW section
ldi spmcrval, (1<<RWSRE) | (1<<SELFPRGEN)
rcallDo_spm

; read back and check, optional
ldi looplo, low(PAGESIZEB) ;init loop variable
ldi loophi, high(PAGESIZEB) ;not required for PAGESIZEB<=256
subi YL, low(PAGESIZEB) ;restore pointer

```

```
        sbci YH, high(PAGESIZEB)
Rdloop:
    lpm r0, Z+
    ld r1, Y+
    cpse r0, r1
    rjmp Error
    sbiw loophi:looplo, 1          ;use subi for PAGESIZEB<=256
    brne Rdloop

    ; return to RWW section
    ; verify that RWW section is safe to read
Return:
    in temp1, SPMCSR
    sbrs temp1, RWWSB          ; If RWWSB is set, the RWW section is not ready yet
    ret
    ; re-enable the RWW section
    ldi spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
    rcallDo_spm
    rjmp Return

Do_spm:
    ; check for previous SPM complete
Wait_spm:
    in temp1, SPMCSR
    sbrc temp1, SELFPRGEN
    rjmp Wait_spm
    ; input: spmcrval determines SPM action
    ; disable interrupts if enabled, store status
    in temp2, SREG
    cli
    ; check that no EEPROM write access is present
Wait_ee:
    sbic EECR, EEPE
    rjmp Wait_ee
    ; SPM timed sequence
    out SPMCSR, spmcrval
    spm
    ; restore SREG (to enable interrupts if originally enabled)
    out SREG, temp2
    ret
```

25.3 Register Description

25.3.1 SPMCSR – Store Program Memory Control and Status Register

The Store Program Memory Control and Status Register contains the control bits needed to control the Program memory operations.

Bit	7	6	5	4	3	2	1	0	
0x37 (0x57)	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SELFPRGEN	SPMCSR
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPMIE: SPM Interrupt Enable**

When the SPMIE bit is written to one, and the I-bit in the Status Register is set (one), the SPM ready interrupt will be enabled. The SPM ready interrupt will be executed as long as the SELFPRGEN bit in the SPMCSR Register is cleared. The interrupt will not be generated during EEPROM write or SPM.

- **Bit 6 – RWWSB: Read-While-Write Section Busy**

This bit is for compatibility with devices supporting Read-While-Write. It will always read as zero in ATmega48PA.

- **Bit 5 – Res: Reserved Bit**

This bit is a reserved bit in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 4 – RWWSRE: Read-While-Write Section Read Enable**

The functionality of this bit in ATmega48PA is a subset of the functionality in ATmega88PA/168PA. If the RWWSRE bit is written while filling the temporary page buffer, the temporary page buffer will be cleared and the data will be lost.

- **Bit 3 – BLBSET: Boot Lock Bit Set**

The functionality of this bit in ATmega48PA is a subset of the functionality in ATmega88PA/168PA. An LPM instruction within three cycles after BLBSET and SELFPRGEN are set in the SPMCSR Register, will read either the Lock bits or the Fuse bits (depending on Z0 in the Z-pointer) into the destination register. See ["Reading the Fuse and Lock Bits from Software" on page 271](#) for details.

- **Bit 2 – PGWRT: Page Write**

If this bit is written to one at the same time as SELFPRGEN, the next SPM instruction within four clock cycles executes Page Write, with the data stored in the temporary buffer. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGWRT bit will auto-clear upon completion of a Page Write, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire Page Write operation.

- **Bit 1 – PGERS: Page Erase**

If this bit is written to one at the same time as SELFPRGEN, the next SPM instruction within four clock cycles executes Page Erase. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGERS bit will auto-clear upon completion of a Page Erase, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire Page Write operation.

- **Bit 0 – SELFPRGEN: Self Programming Enable**

This bit enables the SPM instruction for the next four clock cycles. If written to one together with either RWWSRE, BLBSET, PGWRT, or PGERS, the following SPM instruction will have a special meaning, see description above. If only SELFPRGEN is written, the following SPM instruction will store the value in R1:R0 in the temporary page buffer addressed by the Z-pointer. The LSB of the Z-pointer is ignored. The SELFPRGEN bit will auto-clear upon completion of an SPM instruction, or if no SPM instruction is executed within four clock cycles. During Page Erase and Page Write, the SELFPRGEN bit remains high until the operation is completed.

Writing any other combination than "10001", "01001", "00101", "00011" or "00001" in the lower five bits will have no effect.

26. Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P

26.1 Features

- Read-While-Write Self-Programming
- Flexible Boot Memory Size
- High Security (Separate Boot Lock Bits for a Flexible Protection)
- Separate Fuse to Select Reset Vector
- Optimized Page⁽¹⁾ Size
- Code Efficient Algorithm
- Efficient Read-Modify-Write Support

Note: 1. A page is a section in the Flash consisting of several bytes (see [Table 27-11 on page 299](#)) used during programming. The page organization does not affect normal operation.

26.2 Overview

In ATmega88PA, ATmega168PA and ATmega328P, the Boot Loader Support provides a real Read-While-Write Self-Programming mechanism for downloading and uploading program code by the MCU itself. This feature allows flexible application software updates controlled by the MCU using a Flash-resident Boot Loader program. The Boot Loader program can use any available data interface and associated protocol to read code and write (program) that code into the Flash memory, or read the code from the program memory. The program code within the Boot Loader section has the capability to write into the entire Flash, including the Boot Loader memory. The Boot Loader can thus even modify itself, and it can also erase itself from the code if the feature is not needed anymore. The size of the Boot Loader memory is configurable with fuses and the Boot Loader has two separate sets of Boot Lock bits which can be set independently. This gives the user a unique flexibility to select different levels of protection.

26.3 Application and Boot Loader Flash Sections

The Flash memory is organized in two main sections, the Application section and the Boot Loader section (see [Figure 26-2](#)). The size of the different sections is configured by the **BOOTSZ** Fuses as shown in [Table 26-7 on page 289](#) and [Figure 26-2](#). These two sections can have different level of protection since they have different sets of Lock bits.

26.3.1 Application Section

The Application section is the section of the Flash that is used for storing the application code. The protection level for the Application section can be selected by the application Boot Lock bits (Boot Lock bits 0), see [Table 26-2 on page 281](#). The Application section can never store any Boot Loader code since the SPM instruction is disabled when executed from the Application section.

26.3.2 BLS – Boot Loader Section

While the Application section is used for storing the application code, the The Boot Loader software must be located in the BLS since the SPM instruction can initiate a programming when executing from the BLS only. The SPM instruction can access the entire Flash, including the BLS itself. The protection level for the Boot Loader section can be selected by the Boot Loader Lock bits (Boot Lock bits 1), see [Table 26-3 on page 281](#).

26.4 Read-While-Write and No Read-While-Write Flash Sections

Whether the CPU supports Read-While-Write or if the CPU is halted during a Boot Loader software update is dependent on which address that is being programmed. In addition to the two sections that are configurable by the BOOTSZ Fuses as described above, the Flash is also divided into two fixed sections, the Read-While-Write (RWW) section and the No Read-While-Write (NRWW) section. The limit between the RWW- and NRWW sections is given in [Table 26-8 on page 289](#) and [Figure 26-2 on page 280](#). The main difference between the two sections is:

- When erasing or writing a page located inside the RWW section, the NRWW section can be read during the operation.
- When erasing or writing a page located inside the NRWW section, the CPU is halted during the entire operation.

Note that the user software can never read any code that is located inside the RWW section during a Boot Loader software operation. The syntax "Read-While-Write section" refers to which section that is being programmed (erased or written), not which section that actually is being read during a Boot Loader software update.

26.4.1 RWW – Read-While-Write Section

If a Boot Loader software update is programming a page inside the RWW section, it is possible to read code from the Flash, but only code that is located in the NRWW section. During an ongoing programming, the software must ensure that the RWW section never is being read. If the user software is trying to read code that is located inside the RWW section (i.e., by a call/jmp/lpm or an interrupt) during programming, the software might end up in an unknown state. To avoid this, the interrupts should either be disabled or moved to the Boot Loader section. The Boot Loader section is always located in the NRWW section. The RWW Section Busy bit (RWWSB) in the Store Program Memory Control and Status Register (SPMCSR) will be read as logical one as long as the RWW section is blocked for reading. After a programming is completed, the RWWSB must be cleared by software before reading code located in the RWW section. See [Section "26.9.1" on page 292](#). for details on how to clear RWWSB.

26.4.2 NRWW – No Read-While-Write Section

The code located in the NRWW section can be read when the Boot Loader software is updating a page in the RWW section. When the Boot Loader code updates the NRWW section, the CPU is halted during the entire Page Erase or Page Write operation.

Table 26-1. Read-While-Write Features

Which Section does the Z-pointer Address during the Programming?	Which Section can be read during Programming?	CPU Halted?	Read-While-Write Supported?
RWW Section	NRWW Section	No	Yes
NRWW Section	None	Yes	No

Figure 26-1. Read-While-Write vs. No Read-While-Write

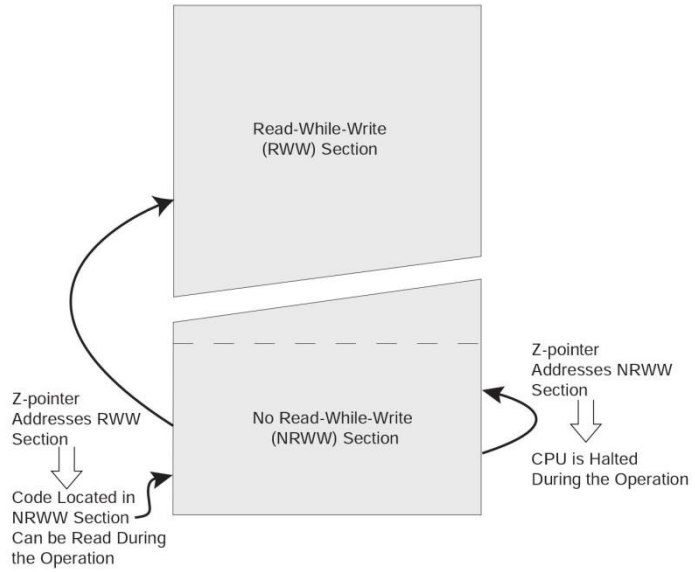
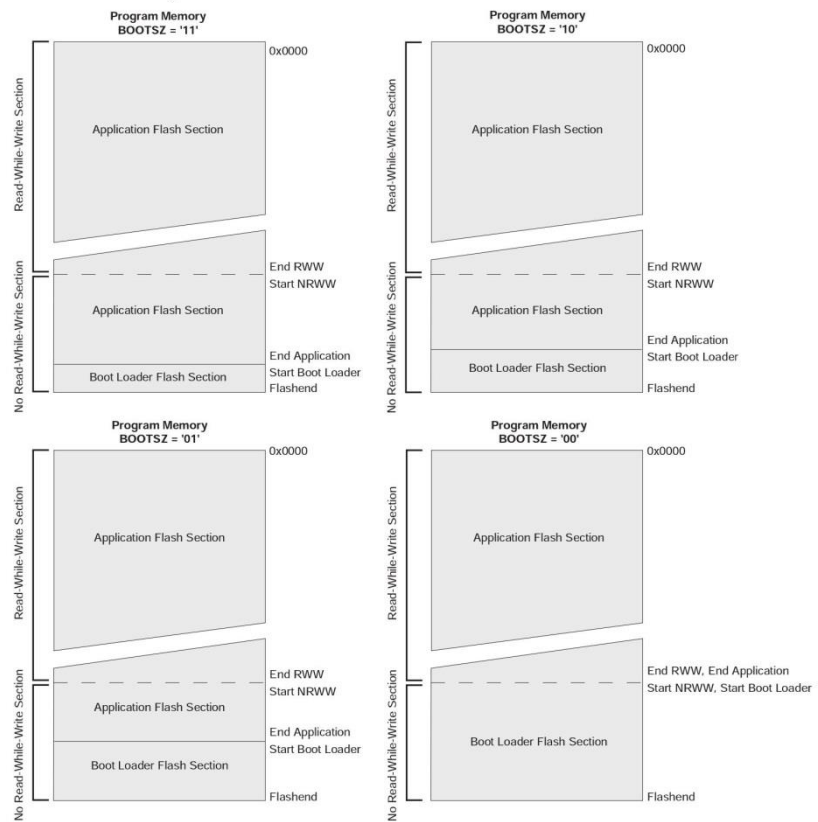


Figure 26-2. Memory Sections



Note: 1. The parameters in the figure above are given in [Table 26-7](#) on page 289.

26.5 Boot Loader Lock Bits

If no Boot Loader capability is needed, the entire Flash is available for application code. The Boot Loader has two separate sets of Boot Lock bits which can be set independently. This gives the user a unique flexibility to select different levels of protection.

The user can select:

- To protect the entire Flash from a software update by the MCU.
- To protect only the Boot Loader Flash section from a software update by the MCU.
- To protect only the Application Flash section from a software update by the MCU.
- Allow software update in the entire Flash.

See [Table 26-2](#) and [Table 26-3](#) for further details. The Boot Lock bits can be set in software and in Serial or Parallel Programming mode, but they can be cleared by a Chip Erase command only. The general Write Lock (Lock Bit mode 2) does not control the programming of the Flash memory by SPM instruction. Similarly, the general Read/Write Lock (Lock Bit mode 1) does not control reading nor writing by LPM/SPM, if it is attempted.

Table 26-2. Boot Lock Bit0 Protection Modes (Application Section)⁽¹⁾

BLB0 Mode	BLB02	BLB01	Protection
1	1	1	No restrictions for SPM or LPM accessing the Application section.
2	1	0	SPM is not allowed to write to the Application section.
3	0	0	SPM is not allowed to write to the Application section, and LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
4	0	1	LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.

Note: 1. "1" means unprogrammed, "0" means programmed

Table 26-3. Boot Lock Bit1 Protection Modes (Boot Loader Section)⁽¹⁾

BLB1 Mode	BLB12	BLB11	Protection
1	1	1	No restrictions for SPM or LPM accessing the Boot Loader section.
2	1	0	SPM is not allowed to write to the Boot Loader section.
3	0	0	SPM is not allowed to write to the Boot Loader section, and LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.
4	0	1	LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.

Note: 1. "1" means unprogrammed, "0" means programmed

26.6 Entering the Boot Loader Program

Entering the Boot Loader takes place by a jump or call from the application program. This may be initiated by a trigger such as a command received via USART, or SPI interface. Alternatively, the Boot Reset Fuse can be programmed so that the Reset Vector is pointing to the Boot Flash start address after a reset. In this case, the Boot Loader is started after a reset. After the application code is loaded, the program can start executing the application code. Note that the fuses cannot be changed by the MCU itself. This means that once the Boot Reset Fuse is programmed, the Reset Vector will always point to the Boot Loader Reset and the fuse can only be changed through the serial or parallel programming interface.

Table 26-4. Boot Reset Fuse⁽¹⁾

BOOTRST	Reset Address
1	Reset Vector = Application Reset (address 0x0000)
0	Reset Vector = Boot Loader Reset (see Table 26-7 on page 289)

Note: 1. "1" means unprogrammed, "0" means programmed

26.7 Addressing the Flash During Self-Programming

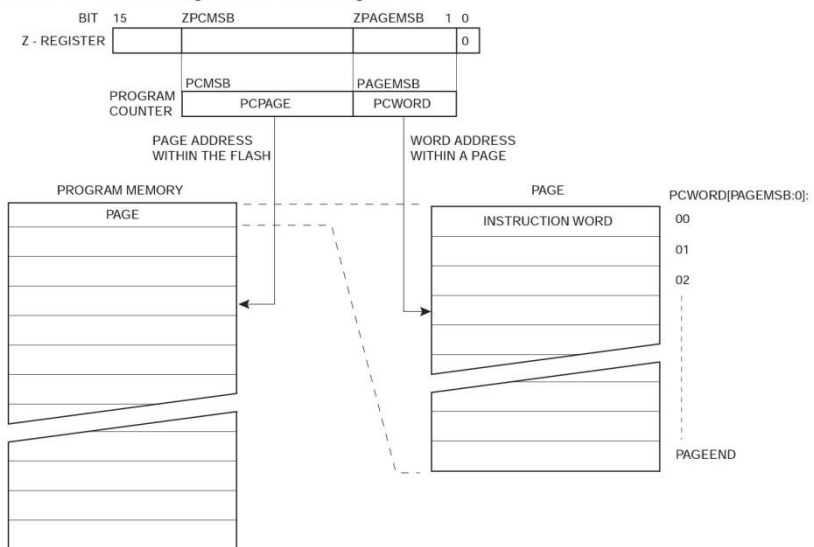
The Z-pointer is used to address the SPM commands.

Bit	15	14	13	12	11	10	9	8
ZH (R31)	Z15	Z14	Z13	Z12	Z11	Z10	Z9	Z8
ZL (R30)	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0
	7	6	5	4	3	2	1	0

Since the Flash is organized in pages (see [Table 27-11 on page 299](#)), the Program Counter can be treated as having two different sections. One section, consisting of the least significant bits, is addressing the words within a page, while the most significant bits are addressing the pages. This is shown in [Figure 26-3](#). Note that the Page Erase and Page Write operations are addressed independently. Therefore it is of major importance that the Boot Loader software addresses the same page in both the Page Erase and Page Write operation. Once a programming operation is initiated, the address is latched and the Z-pointer can be used for other operations.

The only SPM operation that does not use the Z-pointer is Setting the Boot Loader Lock bits. The content of the Z-pointer is ignored and will have no effect on the operation. The LPM instruction does also use the Z-pointer to store the address. Since this instruction addresses the Flash byte-by-byte, also the LSB (bit Z0) of the Z-pointer is used.

Figure 26-3. Addressing the Flash During SPM⁽¹⁾



Note: 1. The different variables used in [Figure 26-3](#) are listed in [Table 26-9 on page 289](#).

26.8 Self-Programming the Flash

The program memory is updated in a page by page fashion. Before programming a page with the data stored in the temporary page buffer, the page must be erased. The temporary page buffer is filled one word at a time using SPM and the buffer can be filled either before the Page Erase command or between a Page Erase and a Page Write operation:

Alternative 1, fill the buffer before a Page Erase

- Fill temporary page buffer
- Perform a Page Erase
- Perform a Page Write

Alternative 2, fill the buffer after Page Erase

- Perform a Page Erase
- Fill temporary page buffer
- Perform a Page Write

If only a part of the page needs to be changed, the rest of the page must be stored (for example in the temporary page buffer) before the erase, and then be rewritten. When using alternative 1, the Boot Loader provides an effective Read-Modify-Write feature which allows the user software to first read the page, do the necessary changes, and then write back the modified data. If alternative 2 is used, it is not possible to read the old data while loading since the page is already erased. The temporary page buffer can be accessed in a random sequence. It is essential that the page address used in both the Page Erase and Page Write operation is addressing the same page. See ["Simple Assembly Code Example for a Boot Loader"](#) on page 286 for an assembly code example.

26.8.1 Performing Page Erase by SPM

To execute Page Erase, set up the address in the Z-pointer, write "X0000011" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE in the Z-register. Other bits in the Z-pointer will be ignored during this operation.

- Page Erase to the RWW section: The NRWW section can be read during the Page Erase.
- Page Erase to the NRWW section: The CPU is halted during the operation.

26.8.2 Filling the Temporary Buffer (Page Loading)

To write an instruction word, set up the address in the Z-pointer and data in R1:R0, write "00000001" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The content of PCWORD in the Z-register is used to address the data in the temporary buffer. The temporary buffer will auto-erase after a Page Write operation or by writing the RWWSRE bit in SPMCSR. It is also erased after a system reset. Note that it is not possible to write more than one time to each address without erasing the temporary buffer.

If the EEPROM is written in the middle of an SPM Page Load operation, all data loaded will be lost.

26.8.3 Performing a Page Write

To execute Page Write, set up the address in the Z-pointer, write "X0000101" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE. Other bits in the Z-pointer must be written to zero during this operation.

- Page Write to the RWW section: The NRWW section can be read during the Page Write.
- Page Write to the NRWW section: The CPU is halted during the operation.

26.8.4 Using the SPM Interrupt

If the SPM interrupt is enabled, the SPM interrupt will generate a constant interrupt when the SELFPRGEN bit in SPMCSR is cleared. This means that the interrupt can be used instead of polling the SPMCSR Register in software. When using the SPM interrupt, the Interrupt Vectors should be moved to the BLS section to avoid that an interrupt is accessing the RWW section when it is blocked for reading. How to move the interrupts is described in "Interrupts" on page 57.

26.8.5 Consideration While Updating BLS

Special care must be taken if the user allows the Boot Loader section to be updated by leaving Boot Lock bit11 unprogrammed. An accidental write to the Boot Loader itself can corrupt the entire Boot Loader, and further software updates might be impossible. If it is not necessary to change the Boot Loader software itself, it is recommended to program the Boot Lock bit11 to protect the Boot Loader software from any internal software changes.

26.8.6 Prevent Reading the RWW Section During Self-Programming

During Self-Programming (either Page Erase or Page Write), the RWW section is always blocked for reading. The user software itself must prevent that this section is addressed during the self programming operation. The RWWSB in the SPMCSR will be set as long as the RWW section is busy. During Self-Programming the Interrupt Vector table should be moved to the BLS as described in "Watchdog Timer" on page 50, or the interrupts must be disabled. Before addressing the RWW section after the programming is completed, the user software must clear the RWWSB by writing the RWWSRE. See "Simple Assembly Code Example for a Boot Loader" on page 286 for an example.

26.8.7 Setting the Boot Loader Lock Bits by SPM

To set the Boot Loader Lock bits and general Lock Bits, write the desired data to R0, write "X0001001" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR.

Bit	7	6	5	4	3	2	1	0
R0	1	1	BLB12	BLB11	BLB02	BLB01	LB2	LB1

See Table 26-2 and Table 26-3 for how the different settings of the Boot Loader bits affect the Flash access.

If bits 5..0 in R0 are cleared (zero), the corresponding Lock bit will be programmed if an SPM instruction is executed within four cycles after BLBSET and SELFPRGEN are set in SPMCSR. The Z-pointer is don't care during this operation, but for future compatibility it is recommended to load the Z-pointer with 0x0001 (same as used for reading the IO_{ck} bits). For future compatibility it is also recommended to set bits 7 and 6 in R0 to "1" when writing the Lock bits. When programming the Lock bits the entire Flash can be read during the operation.

26.8.8 EEPROM Write Prevents Writing to SPMCSR

Note that an EEPROM write operation will block all software programming to Flash. Reading the Fuses and Lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEPE) in the EECR Register and verifies that the bit is cleared before writing to the SPMCSR Register.

26.8.9 Reading the Fuse and Lock Bits from Software

It is possible to read both the Fuse and Lock bits from software. To read the Lock bits, load the Z-pointer with 0x0001 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM

instruction is executed within three CPU cycles after the BLBSET and SELFPRGEN bits are set in SPMCSR, the value of the Lock bits will be loaded in the destination register. The BLBSET and SELFPRGEN bits will auto-clear upon completion of reading the Lock bits or if no LPM instruction is executed within three CPU cycles or no SPM instruction is executed within four CPU cycles. When BLBSET and SELFPRGEN are cleared, LPM will work as described in the Instruction set Manual.

Bit	7	6	5	4	3	2	1	0
Rd	-	-	BLB12	BLB11	BLB02	BLB01	LB2	LB1

The algorithm for reading the Fuse Low byte is similar to the one described above for reading the Lock bits. To read the Fuse Low byte, load the Z-pointer with 0x0000 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Fuse Low byte (FLB) will be loaded in the destination register as shown below. Refer to [Table 27-5 on page 296](#) for a detailed description and mapping of the Fuse Low byte.

Bit	7	6	5	4	3	2	1	0
Rd	FLB7	FLB6	FLB5	FLB4	FLB3	FLB2	FLB1	FLB0

Similarly, when reading the Fuse High byte, load 0x0003 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Fuse High byte (FHB) will be loaded in the destination register as shown below. Refer to [Table 27-7 on page 296](#) for detailed description and mapping of the Fuse High byte.

Bit	7	6	5	4	3	2	1	0
Rd	FHB7	FHB6	FHB5	FHB4	FHB3	FHB2	FHB1	FHB0

When reading the Extended Fuse byte, load 0x0002 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Extended Fuse byte (EFB) will be loaded in the destination register as shown below. Refer to [Table 27-5 on page 296](#) for detailed description and mapping of the Extended Fuse byte.

Bit	7	6	5	4	3	2	1	0
Rd	-	-	-	-	EFB3	EFB2	EFB1	EFB0

Fuse and Lock bits that are programmed, will be read as zero. Fuse and Lock bits that are unprogrammed, will be read as one.

26.8.10 Reading the Signature Row from Software

To read the Signature Row from software, load the Z-pointer with the signature byte address given in [Table 26-5 on page 286](#) and set the SIGRD and SPMEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the SIGRD and SPMEN bits are set in SPMCSR, the signature byte value will be loaded in the destination register. The SIGRD and SPMEN bits will auto-clear upon completion of reading the Signature Row Lock bits or if no LPM instruction is executed within three CPU cycles. When SIGRD and SPMEN are cleared, LPM will work as described in the Instruction set Manual.

Table 26-5. Signature Row Addressing

Signature Byte	Z-Pointer Address
Device Signature Byte 1	0x0000
Device Signature Byte 2	0x0002
Device Signature Byte 3	0x0004
RC Oscillator Calibration Byte	0x0001

Note: All other addresses are reserved for future use.

26.8.11 Preventing Flash Corruption

During periods of low V_{CC} , the Flash program can be corrupted because the supply voltage is too low for the CPU and the Flash to operate properly. These issues are the same as for board level systems using the Flash, and the same design solutions should be applied.

A Flash program corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the Flash requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

Flash corruption can easily be avoided by following these design recommendations (one is sufficient):

1. If there is no need for a Boot Loader update in the system, program the Boot Loader Lock bits to prevent any Boot Loader software updates.
2. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD) if the operating voltage matches the detection level. If not, an external low V_{CC} reset protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.
3. Keep the AVR core in Power-down sleep mode during periods of low V_{CC} . This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the SPMCSR Register and thus the Flash from unintentional writes.

26.8.12 Programming Time for Flash when Using SPM

The calibrated RC Oscillator is used to time Flash accesses. Table 26-6 shows the typical programming time for Flash accesses from the CPU.

Table 26-6. SPM Programming Time⁽¹⁾

Symbol	Min Programming Time	Max Programming Time
Flash write (Page Erase, Page Write, and write Lock bits by SPM)	3.7 ms	4.5 ms

Note: 1. Minimum and maximum programming time is per individual operation.

26.8.13 Simple Assembly Code Example for a Boot Loader

```
;-the routine writes one page of data from RAM to Flash
; the first data location in RAM is pointed to by the Y pointer
; the first data location in Flash is pointed to by the Z-pointer
;-error handling is not included
```

```

;-the routine must be placed inside the Boot space
; (at least the Do_spm sub routine). Only code inside NRWW section can
; be read during Self-Programming (Page Erase and Page Write).
;-registers used: r0, r1, temp1 (r16), temp2 (r17), looplo (r24),
; loophi (r25), spmcrval (r20)
; storing and restoring of registers is not included in the routine
; register usage can be optimized at the expense of code size
;-It is assumed that either the interrupt table is moved to the Boot
; loader section or that the interrupts are disabled.
.equ PAGESIZEB = PAGESIZE*2 ;PAGESIZEB is page size in BYTES, not words
.org SMALLBOOTSTART
Write_page:
; Page Erase
ldi spmcrval, (1<<PGRS) | (1<<SELFPRGEN)
call Do_spm

; re-enable the RWW section
ldi spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
call Do_spm

; transfer data from RAM to Flash page buffer
ldi looplo, low(PAGESIZEB) ;init loop variable
ldi loophi, high(PAGESIZEB) ;not required for PAGESIZEB<=256
Wrloop:
ld r0, Y+
ld r1, Y+
ldi spmcrval, (1<<SELFPRGEN)
call Do_spm
adiw ZH:ZL, 2
sbw loophi:looplo, 2 ;use subi for PAGESIZEB<=256
brne Wrloop

; execute Page Write
subi ZL, low(PAGESIZEB) ;restore pointer
sbci ZH, high(PAGESIZEB) ;not required for PAGESIZEB<=256
ldi spmcrval, (1<<PGWRT) | (1<<SELFPRGEN)
call Do_spm

; re-enable the RWW section
ldi spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
call Do_spm

; read back and check, optional
ldi looplo, low(PAGESIZEB) ;init loop variable
ldi loophi, high(PAGESIZEB) ;not required for PAGESIZEB<=256
subi YL, low(PAGESIZEB) ;restore pointer
sbci YH, high(PAGESIZEB)
Rdloop:
lpm r0, Z+
ld r1, Y+
cpse r0, r1
jmp Error
sbw loophi:looplo, 1 ;use subi for PAGESIZEB<=256
brne Rdloop

; return to RWW section
; verify that RWW section is safe to read
Return:
in temp1, SPMCSR

```



```
sbrs temp1, RWWSB      ; If RWWSB is set, the RWW section is not ready yet
ret
; re-enable the RWW section
ldi  spmcrval, (1<<RWSRE) | (1<<SELFPRGEN)
call Do_spm
rjmp Return

Do_spm:
; check for previous SPM complete
Wait_spm:
in   temp1, SPMCSR
sbrs temp1, SELFPRGEN
rjmp Wait_spm
; input: spmcrval determines SPM action
; disable interrupts if enabled, store status
in   temp2, SREG
cli
; check that no EEPROM write access is present
Wait_ee:
sbic EECR, EEPE
rjmp Wait_ee
; SPM timed sequence
out  SPMCSR, spmcrval
spm
; restore SREG (to enable interrupts if originally enabled)
out  SREG, temp2
ret
```

26.8.14 ATmega88PA Boot Loader Parameters

In [Table 26-7](#) through [Table 26-9](#), the parameters used in the description of the self programming are given.

Table 26-7. Boot Size Configuration, ATmega88PA

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	128 words	4	0x000 - 0xF7F	0xF80 - 0xFFF	0xF7F	0xF80
1	0	256 words	8	0x000 - 0xEFF	0xF00 - 0xFFF	0xEFF	0xF00
0	1	512 words	16	0x000 - 0xDFF	0xE00 - 0xFFF	0xDFF	0xE00
0	0	1024 words	32	0x000 - 0xBFF	0xC00 - 0xFFF	0xBFF	0xC00

Note: The different BOOTSZ Fuse configurations are shown in [Figure 26-2](#) on page 280.

Table 26-8. Read-While-Write Limit, ATmega88PA

Section	Pages	Address
Read-While-Write section (RWW)	96	0x000 - 0xBFF
No Read-While-Write section (NRWW)	32	0xC00 - 0xFFF

For details about these two section, see "NRWW – No Read-While-Write Section" on page 278 and "RWW – Read-While-Write Section" on page 278

Table 26-9. Explanation of Different Variables used in [Figure 26-3](#) and the Mapping to the Z-pointer, ATmega88PA

Variable		Corresponding Z-value ⁽¹⁾	Description
PCMSB	11		Most significant bit in the Program Counter. (The Program Counter is 12 bits PC[11:0])
PAGEMSB	4		Most significant bit which is used to address the words within one page (32 words in a page requires 5 bits PC [4:0]).
ZPCMSB		Z12	Bit in Z-register that is mapped to PCMSB. Because Z0 is not used, the ZPCMSB equals PCMSB + 1.
ZPAGEMSB		Z5	Bit in Z-register that is mapped to PAGEMSB. Because Z0 is not used, the ZPAGEMSB equals PAGEMSB + 1.
PCPAGE	PC[11:5]	Z12:Z6	Program counter page address: Page select, for page erase and page write
PCWORD	PC[4:0]	Z5:Z1	Program counter word address: Word select, for filling temporary buffer (must be zero during page write operation)

Note: 1. Z15:Z13: always ignored
 Z0: should be zero for all SPM commands, byte select for the LPM instruction.
 See "Addressing the Flash During Self-Programming" on page 282 for details about the use of Z-pointer during Self-Programming.

26.8.15 ATmega168PA Boot Loader Parameters

In Table 26-10 through Table 26-12, the parameters used in the description of the self programming are given.

Table 26-10. Boot Size Configuration, ATmega168PA

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	128 words	2	0x0000 - 0x1F7F	0x1F80 - 0x1FFF	0x1F7F	0x1F80
1	0	256 words	4	0x0000 - 0x1EFF	0x1F00 - 0x1FFF	0x1EFF	0x1F00
0	1	512 words	8	0x0000 - 0x1DFF	0x1E00 - 0x1FFF	0x1DFF	0x1E00
0	0	1024 words	16	0x0000 - 0x1BFF	0x1C00 - 0x1FFF	0x1BFF	0x1C00

Note: The different BOOTSZ Fuse configurations are shown in Figure 26-2 on page 280.

Table 26-11. Read-While-Write Limit, ATmega168PA

Section	Pages	Address
Read-While-Write section (RWW)	112	0x0000 - 0x1BFF
No Read-While-Write section (NRWW)	16	0x1C00 - 0x1FFF

For details about these two section, see "NRWW – No Read-While-Write Section" on page 278 and "RWW – Read-While-Write Section" on page 278

Table 26-12. Explanation of Different Variables used in Figure 26-3 and the Mapping to the Z-pointer, ATmega168PA

Variable		Corresponding Z-value ⁽¹⁾	Description
PCMSB	12		Most significant bit in the Program Counter. (The Program Counter is 13 bits PC[12:0])
PAGEMSB	5		Most significant bit which is used to address the words within one page (64 words in a page requires 6 bits PC [5:0])
ZPCMSB		Z13	Bit in Z-register that is mapped to PCMSB. Because Z0 is not used, the ZPCMSB equals PCMSB + 1.
ZPAGEMSB		Z6	Bit in Z-register that is mapped to PAGEMSB. Because Z0 is not used, the ZPAGEMSB equals PAGEMSB + 1.
PCPAGE	PC[12:6]	Z13:Z7	Program counter page address: Page select, for page erase and page write
PCWORD	PC[5:0]	Z6:Z1	Program counter word address: Word select, for filling temporary buffer (must be zero during page write operation)

Note: 1. Z15:Z14: always ignored
 Z0: should be zero for all SPM commands, byte select for the LPM instruction.
 See "Addressing the Flash During Self-Programming" on page 282 for details about the use of Z-pointer during Self-Programming.

26.8.16 ATmega328P Boot Loader Parameters

In Table 26-13 through Table 26-15, the parameters used in the description of the self programming are given.

Table 26-13. Boot Size Configuration, ATmega328P

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	256 words	4	0x0000 - 0x3EFF	0x3F00 - 0x3FFF	0x3EFF	0x3F00
1	0	512 words	8	0x0000 - 0x3DFF	0x3E00 - 0x3FFF	0x3DFF	0x3E00
0	1	1024 words	16	0x0000 - 0x3BFF	0x3C00 - 0x3FFF	0x3BFF	0x3C00
0	0	2048 words	32	0x0000 - 0x37FF	0x3800 - 0x3FFF	0x37FF	0x3800

Note: The different BOOTSZ Fuse configurations are shown in Figure 26-2 on page 280.

Table 26-14. Read-While-Write Limit, ATmega328P

Section	Pages	Address
Read-While-Write section (RWW)	224	0x0000 - 0x37FF
No Read-While-Write section (NRWW)	32	0x3800 - 0x3FFF

For details about these two section, see "NRWW – No Read-While-Write Section" on page 278 and "RWW – Read-While-Write Section" on page 278.

Table 26-15. Explanation of Different Variables used in Figure 26-3 and the Mapping to the Z-pointer, ATmega328P

Variable		Corresponding Z-value ⁽¹⁾	Description
PCMSB	13		Most significant bit in the Program Counter. (The Program Counter is 14 bits PC[13:0])
PAGEMSB	5		Most significant bit which is used to address the words within one page (64 words in a page requires 6 bits PC [5:0])
ZPCMSB		Z14	Bit in Z-register that is mapped to PCMSB. Because Z0 is not used, the ZPCMSB equals PCMSB + 1.
ZPAGEMSB		Z6	Bit in Z-register that is mapped to PAGEMSB. Because Z0 is not used, the ZPAGEMSB equals PAGEMSB + 1.
PCPAGE	PC[13:6]	Z14:Z7	Program counter page address: Page select, for page erase and page write
PCWORD	PC[5:0]	Z6:Z1	Program counter word address: Word select, for filling temporary buffer (must be zero during page write operation)

Note: 1. Z15: always ignored

Z0: should be zero for all SPM commands, byte select for the LPM instruction.

See "Addressing the Flash During Self-Programming" on page 282 for details about the use of Z-pointer during Self-Programming.

26.9 Register Description

26.9.1 SPMCSR – Store Program Memory Control and Status Register

The Store Program Memory Control and Status Register contains the control bits needed to control the Boot Loader operations.

Bit	7	6	5	4	3	2	1	0	
0x37 (0x57)	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SELFPRGEN	SPMCSR
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPMIE: SPM Interrupt Enable**

When the SPMIE bit is written to one, and the I-bit in the Status Register is set (one), the SPM ready interrupt will be enabled. The SPM ready Interrupt will be executed as long as the SELFPRGEN bit in the SPMCSR Register is cleared.

- **Bit 6 – RWWSB: Read-While-Write Section Busy**

When a Self-Programming (Page Erase or Page Write) operation to the RWW section is initiated, the RWWSB will be set (one) by hardware. When the RWWSB bit is set, the RWW section cannot be accessed. The RWWSB bit will be cleared if the RWWSRE bit is written to one after a Self-Programming operation is completed. Alternatively the RWWSB bit will automatically be cleared if a page load operation is initiated.

- **Bit 5 – Res: Reserved Bit**

This bit is a reserved bit in the ATmega48PA/88PA/168PA/328P and always read as zero.

- **Bit 4 – RWWSRE: Read-While-Write Section Read Enable**

When programming (Page Erase or Page Write) to the RWW section, the RWW section is blocked for reading (the RWWSB will be set by hardware). To re-enable the RWW section, the user software must wait until the programming is completed (SELFPRGEN will be cleared). Then, if the RWWSRE bit is written to one at the same time as SELFPRGEN, the next SPM instruction within four clock cycles re-enables the RWW section. The RWW section cannot be re-enabled while the Flash is busy with a Page Erase or a Page Write (SELFPRGEN is set). If the RWWSRE bit is written while the Flash is being loaded, the Flash load operation will abort and the data loaded will be lost.

- **Bit 3 – BLBSET: Boot Lock Bit Set**

If this bit is written to one at the same time as SELFPRGEN, the next SPM instruction within four clock cycles sets Boot Lock bits and Memory Lock bits, according to the data in R0. The data in R1 and the address in the Z-pointer are ignored. The BLBSET bit will automatically be cleared upon completion of the Lock bit set, or if no SPM instruction is executed within four clock cycles.

An LPM instruction within three cycles after BLBSET and SELFPRGEN are set in the SPMCSR Register, will read either the Lock bits or the Fuse bits (depending on Z0 in the Z-pointer) into the destination register. See ["Reading the Fuse and Lock Bits from Software"](#) on page 284 for details.

- **Bit 2 – PGWRT: Page Write**

If this bit is written to one at the same time as SELFPRGEN, the next SPM instruction within four clock cycles executes Page Write, with the data stored in the temporary buffer. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The

PGWRT bit will auto-clear upon completion of a Page Write, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire Page Write operation if the NRWW section is addressed.

- **Bit 1 – PGERS: Page Erase**

If this bit is written to one at the same time as SELFPRGEN, the next SPM instruction within four clock cycles executes Page Erase. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGERS bit will auto-clear upon completion of a Page Erase, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire Page Write operation if the NRWW section is addressed.

- **Bit 0 – SELFPRGEN: Self Programming Enable**

This bit enables the SPM instruction for the next four clock cycles. If written to one together with either RWWSRE, BLBSET, PGWRT or PGERS, the following SPM instruction will have a special meaning, see description above. If only SELFPRGEN is written, the following SPM instruction will store the value in R1:R0 in the temporary page buffer addressed by the Z-pointer. The LSB of the Z-pointer is ignored. The SELFPRGEN bit will auto-clear upon completion of an SPM instruction, or if no SPM instruction is executed within four clock cycles. During Page Erase and Page Write, the SELFPRGEN bit remains high until the operation is completed.

Writing any other combination than "10001", "01001", "00101", "00011" or "00001" in the lower five bits will have no effect.

27. Memory Programming

27.1 Program And Data Memory Lock Bits

The ATmega88PA/168PA/328P provides six Lock bits which can be left unprogrammed ("1") or can be programmed ("0") to obtain the additional features listed in Table 27-2. The Lock bits can only be erased to "1" with the Chip Erase command. The ATmega48PA has no separate Boot Loader section. The SPM instruction is enabled for the whole Flash if the SELFPRGEN fuse is programmed ("0"), otherwise it is disabled.

Table 27-1. Lock Bit Byte⁽¹⁾

Lock Bit Byte	Bit No	Description	Default Value
	7	–	1 (unprogrammed)
	6	–	1 (unprogrammed)
BLB12 ⁽²⁾	5	Boot Lock bit	1 (unprogrammed)
BLB11 ⁽²⁾	4	Boot Lock bit	1 (unprogrammed)
BLB02 ⁽²⁾	3	Boot Lock bit	1 (unprogrammed)
BLB01 ⁽²⁾	2	Boot Lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

Notes: 1. "1" means unprogrammed, "0" means programmed.
2. Only on ATmega88PA/168PA/328P.

Table 27-2. Lock Bit Protection Modes⁽¹⁾⁽²⁾

Memory Lock Bits			Protection Type
LB Mode	LB2	LB1	
1	1	1	No memory lock features enabled.
2	1	0	Further programming of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode. ⁽¹⁾
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Boot Lock bits and Fuse bits are locked in both Serial and Parallel Programming mode. ⁽¹⁾

Notes: 1. Program the Fuse bits and Boot Lock bits before programming the LB1 and LB2.
2. "1" means unprogrammed, "0" means programmed

Table 27-3. Lock Bit Protection Modes⁽¹⁾⁽²⁾, Only ATmega88PA/168PA/328P.

BLB0 Mode	BLB02	BLB01	
1	1	1	No restrictions for SPM or LPM accessing the Application section.
2	1	0	SPM is not allowed to write to the Application section.
3	0	0	SPM is not allowed to write to the Application section, and LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
4	0	1	LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
BLB1 Mode	BLB12	BLB11	
1	1	1	No restrictions for SPM or LPM accessing the Boot Loader section.
2	1	0	SPM is not allowed to write to the Boot Loader section.
3	0	0	SPM is not allowed to write to the Boot Loader section, and LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.
4	0	1	LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.

Notes: 1. Program the Fuse bits and Boot Lock bits before programming the LB1 and LB2.
 2. "1" means unprogrammed, "0" means programmed

27.2 Fuse Bits

The ATmega48PA/88PA/168PA/328P has three Fuse bytes. [Table 27-5](#) - [Table 27-9](#) describe briefly the functionality of all the fuses and how they are mapped into the Fuse bytes. Note that the fuses are read as logical zero, "0", if they are programmed.

Table 27-4. Extended Fuse Byte for ATmega48PA

Extended Fuse Byte	Bit No	Description	Default Value
–	7	–	1
–	6	–	1
–	5	–	1
–	4	–	1
–	3	–	1
–	2	–	1
–	1	–	1
SELFPRGEN	0	Self Programming Enable	1 (unprogrammed)

Table 27-5. Extended Fuse Byte for ATmega88PA/168PA

Extended Fuse Byte	Bit No	Description	Default Value
–	7	–	1
–	6	–	1
–	5	–	1
–	4	–	1
–	3	–	1
BOOTSZ1	2	Select Boot Size (see Table 26-7 on page 289 and Table 26-10 on page 290 for details)	0 (programmed) ⁽¹⁾
BOOTSZ0	1	Select Boot Size (see Table 26-7 on page 289 and Table 26-10 on page 290 for details)	0 (programmed) ⁽¹⁾
BOTRST	0	Select Reset Vector	1 (unprogrammed)

Note: 1. The default value of BOOTSZ[1:0] results in maximum Boot Size. See “Pin Name Mapping” on page 300.

Table 27-6. Extended Fuse Byte for ATmega328P

Extended Fuse Byte	Bit No	Description	Default Value
–	7	–	1
–	6	–	1
–	5	–	1
–	4	–	1
–	3	–	1
BODLEVEL2 ⁽¹⁾	2	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL1 ⁽¹⁾	1	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL0 ⁽¹⁾	0	Brown-out Detector trigger level	1 (unprogrammed)

Note: 1. See Table 28-4 on page 318 for BODLEVEL Fuse decoding.

Table 27-7. Fuse High Byte for ATmega48PA/88PA/168PA

High Fuse Byte	Bit No	Description	Default Value
RSTDISBL ⁽¹⁾	7	External Reset Disable	1 (unprogrammed)
DWEN	6	debugWIRE Enable	1 (unprogrammed)
SPIEN ⁽²⁾	5	Enable Serial Program and Data Downloading	0 (programmed, SPI programming enabled)

Table 27-7. Fuse High Byte for ATmega48PA/88PA/168PA (Continued)

High Fuse Byte	Bit No	Description	Default Value
WDTON ⁽³⁾	4	Watchdog Timer Always On	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed), EEPROM not reserved
BODLEVEL2 ⁽⁴⁾	2	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL1 ⁽⁴⁾	1	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL0 ⁽⁴⁾	0	Brown-out Detector trigger level	1 (unprogrammed)

- Notes:
1. See "Alternate Functions of Port C" on page 85 for description of RSTDISBL Fuse.
 2. The SPIEN Fuse is not accessible in serial programming mode.
 3. See "WDTCR – Watchdog Timer Control Register" on page 54 for details.
 4. See Table 28-4 on page 318 for BODLEVEL Fuse decoding.

Table 27-8. Fuse High Byte for ATmega328P

High Fuse Byte	Bit No	Description	Default Value
RSTDISBL ⁽¹⁾	7	External Reset Disable	1 (unprogrammed)
DWEN	6	debugWIRE Enable	1 (unprogrammed)
SPIEN ⁽²⁾	5	Enable Serial Program and Data Downloading	0 (programmed, SPI programming enabled)
WDTON ⁽³⁾	4	Watchdog Timer Always On	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed), EEPROM not reserved
BOOTSZ1	2	Select Boot Size (see Table 26-7 on page 289, Table 26-10 on page 290 and Table 26-13 on page 291 for details)	0 (programmed) ⁽⁴⁾
BOOTSZ0	1	Select Boot Size (see Table 26-7 on page 289, Table 26-10 on page 290 and Table 26-13 on page 291 for details)	0 (programmed) ⁽⁴⁾
BOTRST	0	Select Reset Vector	1 (unprogrammed)

- Notes:
1. See "Alternate Functions of Port C" on page 85 for description of RSTDISBL Fuse.
 2. The SPIEN Fuse is not accessible in serial programming mode.
 3. See "WDTCR – Watchdog Timer Control Register" on page 54 for details.
 4. The default value of BOOTSZ[1:0] results in maximum Boot Size. See "Pin Name Mapping" on page 300.

Table 27-9. Fuse Low Byte

Low Fuse Byte	Bit No	Description	Default Value
CKDIV8 ⁽⁴⁾	7	Divide clock by 8	0 (programmed)
CKOUT ⁽³⁾	6	Clock output	1 (unprogrammed)
SUT1	5	Select start-up time	1 (unprogrammed) ⁽¹⁾
SUT0	4	Select start-up time	0 (programmed) ⁽¹⁾
CKSEL3	3	Select Clock source	0 (programmed) ⁽²⁾
CKSEL2	2	Select Clock source	0 (programmed) ⁽²⁾
CKSEL1	1	Select Clock source	1 (unprogrammed) ⁽²⁾
CKSEL0	0	Select Clock source	0 (programmed) ⁽²⁾

- Note:
1. The default value of SUT1..0 results in maximum start-up time for the default clock source. See [Table 8-12 on page 33](#) for details.
 2. The default setting of CKSEL3..0 results in internal RC Oscillator @ 8 MHz. See [Table 8-11 on page 33](#) for details.
 3. The CKOUT Fuse allows the system clock to be output on PORTB0. See ["Clock Output Buffer" on page 35](#) for details.
 4. See ["System Clock Prescaler" on page 35](#) for details.

The status of the Fuse bits is not affected by Chip Erase. Note that the Fuse bits are locked if Lock bit1 (LB1) is programmed. Program the Fuse bits before programming the Lock bits.

27.2.1 Latching of Fuses

The fuse values are latched when the device enters programming mode and changes of the fuse values will have no effect until the part leaves Programming mode. This does not apply to the EESAVE Fuse which will take effect once it is programmed. The fuses are also latched on Power-up in Normal mode.

27.3 Signature Bytes

All Atmel microcontrollers have a three-byte signature code which identifies the device. This code can be read in both serial and parallel mode, also when the device is locked. The three bytes reside in a separate address space. For the ATmega48PA/88PA/168PA/328P the signature bytes are given in [Table 27-10](#).

Table 27-10. Device ID

Part	Signature Bytes Address		
	0x000	0x001	0x002
ATmega48PA	0x1E	0x92	0x0A
ATmega88PA	0x1E	0x93	0x0F
ATmega168PA	0x1E	0x94	0x0B
ATmega328P	0x1E	0x95	0x0F

27.4 Calibration Byte

The ATmega48PA/88PA/168PA/328P has a byte calibration value for the Internal RC Oscillator. This byte resides in the high byte of address 0x000 in the signature address space. During reset,

this byte is automatically written into the OSCCAL Register to ensure correct frequency of the calibrated RC Oscillator.

27.5 Page Size

Table 27-11. No. of Words in a Page and No. of Pages in the Flash

Device	Flash Size	Page Size	PCWORD	No. of Pages	PCPAGE	PCMSB
ATmega48PA	2K words (4K bytes)	32 words	PC[4:0]	64	PC[10:5]	10
ATmega88PA	4K words (8K bytes)	32 words	PC[4:0]	128	PC[11:5]	11
ATmega168PA	8K words (16K bytes)	64 words	PC[5:0]	128	PC[12:6]	12
ATmega328P	16K words (32K bytes)	64 words	PC[5:0]	256	PC[13:6]	13

Table 27-12. No. of Words in a Page and No. of Pages in the EEPROM

Device	EEPROM Size	Page Size	PCWORD	No. of Pages	PCPAGE	EEAMSB
ATmega48PA	256 bytes	4 bytes	EEA[1:0]	64	EEA[7:2]	7
ATmega88PA	512 bytes	4 bytes	EEA[1:0]	128	EEA[8:2]	8
ATmega168PA	512 bytes	4 bytes	EEA[1:0]	128	EEA[8:2]	8
ATmega328P	1K bytes	4 bytes	EEA[1:0]	256	EEA[9:2]	9

27.6 Parallel Programming Parameters, Pin Mapping, and Commands

This section describes how to parallel program and verify Flash Program memory, EEPROM Data memory, Memory Lock bits, and Fuse bits in the ATmega48PA/88PA/168PA/328P. Pulses are assumed to be at least 250 ns unless otherwise noted.

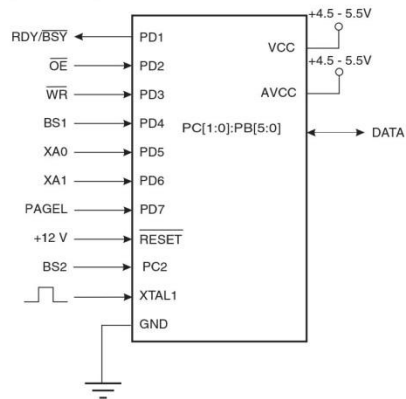
27.6.1 Signal Names

In this section, some pins of the ATmega48PA/88PA/168PA/328P are referenced by signal names describing their functionality during parallel programming, see [Figure 27-1](#) and [Table 27-13](#). Pins not described in the following table are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding is shown in [Table 27-15](#).

When pulsing \overline{WR} or \overline{OE} , the command loaded determines the action executed. The different Commands are shown in [Table 27-16](#).

Figure 27-1. Parallel Programming



Note: $V_{CC} - 0.3V < AV_{CC} < V_{CC} + 0.3V$, however, AV_{CC} should always be within 4.5 - 5.5V

Table 27-13. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/BSY	PD1	O	0: Device is busy programming, 1: Device is ready for new command
\overline{OE}	PD2	I	Output Enable (Active low)
\overline{WR}	PD3	I	Write Pulse (Active low)
BS1	PD4	I	Byte Select 1 ("0" selects Low byte, "1" selects High byte)
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1
PAGEL	PD7	I	Program memory and EEPROM Data Page Load
BS2	PC2	I	Byte Select 2 ("0" selects Low byte, "1" selects 2'nd High byte)
DATA	{PC[1:0]; PB[5:0]}	I/O	Bi-directional Data bus (Output when \overline{OE} is low)

Table 27-14. Pin Values Used to Enter Programming Mode

Pin	Symbol	Value
PAGEL	Prog_enable[3]	0
XA1	Prog_enable[2]	0
XA0	Prog_enable[1]	0
BS1	Prog_enable[0]	0

Table 27-15. XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load Flash or EEPROM Address (High or low address byte determined by BS1).
0	1	Load Data (High or Low data byte for Flash determined by BS1).
1	0	Load Command
1	1	No Action, Idle

Table 27-16. Command Byte Bit Coding

Command Byte	Command Executed
1000 0000	Chip Erase
0100 0000	Write Fuse bits
0010 0000	Write Lock bits
0001 0000	Write Flash
0001 0001	Write EEPROM
0000 1000	Read Signature Bytes and Calibration byte
0000 0100	Read Fuse and Lock bits
0000 0010	Read Flash
0000 0011	Read EEPROM

27.7 Parallel Programming

27.7.1 Enter Programming Mode

The following algorithm puts the device in Parallel (High-voltage) Programming mode:

1. Set Prog_enable pins listed in [Table 27-14 on page 300](#) to "0000", RESET pin to 0V and V_{CC} to 0V.
2. Apply 4.5 - 5.5V between V_{CC} and GND.
Ensure that V_{CC} reaches at least 1.8V within the next 20 μs.

3. Wait 20 - 60 μs, and apply 11.5 - 12.5V to RESET.
4. Keep the Prog_enable pins unchanged for at least 10μs after the High-voltage has been applied to ensure the Prog_enable Signature has been latched.
5. Wait at least 300 μs before giving any parallel programming commands.
6. Exit Programming mode by power the device down or by bringing RESET pin to 0V.

If the rise time of the V_{CC} is unable to fulfill the requirements listed above, the following alternative algorithm can be used.

1. Set Prog_enable pins listed in [Table 27-14 on page 300](#) to "0000", RESET pin to 0V and V_{CC} to 0V.
2. Apply 4.5 - 5.5V between V_{CC} and GND.
3. Monitor V_{CC}, and as soon as V_{CC} reaches 0.9 - 1.1V, apply 11.5 - 12.5V to RESET.

4. Keep the Prog_enable pins unchanged for at least 10µs after the High-voltage has been applied to ensure the Prog_enable Signature has been latched.
5. Wait until V_{CC} actually reaches 4.5 -5.5V before giving any parallel programming commands.
6. Exit Programming mode by power the device down or by bringing RESET pin to 0V.

27.7.2 Considerations for Efficient Programming

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered.

- The command needs only be loaded once when writing or reading multiple memory locations.
- Skip writing the data value 0xFF, that is the contents of the entire EEPROM (unless the EESAVE Fuse is programmed) and Flash after a Chip Erase.
- Address high byte needs only be loaded before programming or reading a new 256 word window in Flash or 256 byte EEPROM. This consideration also applies to Signature bytes reading.

27.7.3 Chip Erase

The Chip Erase will erase the Flash and EEPROM⁽¹⁾ memories plus Lock bits. The Lock bits are not reset until the program memory has been completely erased. The Fuse bits are not changed. A Chip Erase must be performed before the Flash and/or EEPROM are reprogrammed.

Note: 1. The EEPROM memory is preserved during Chip Erase if the EESAVE Fuse is programmed. Load Command "Chip Erase"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS1 to "0".
3. Set DATA to "1000 0000". This is the command for Chip Erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give \overline{WR} a negative pulse. This starts the Chip Erase. RDY/ \overline{BSY} goes low.
6. Wait until RDY/ \overline{BSY} goes high before loading a new command.

27.7.4 Programming the Flash

The Flash is organized in pages, see [Table 27-11 on page 299](#). When programming the Flash, the program data is latched into a page buffer. This allows one page of program data to be programmed simultaneously. The following procedure describes how to program the entire Flash memory:

A. Load Command "Write Flash"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS1 to "0".
3. Set DATA to "0001 0000". This is the command for Write Flash.
4. Give XTAL1 a positive pulse. This loads the command.

B. Load Address Low byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS1 to "0". This selects low address.
3. Set DATA = Address low byte (0x00 - 0xFF).

4. Give XTAL1 a positive pulse. This loads the address low byte.

C. Load Data Low Byte

1. Set XA1, XA0 to "01". This enables data loading.
2. Set DATA = Data low byte (0x00 - 0xFF).
3. Give XTAL1 a positive pulse. This loads the data byte.

D. Load Data High Byte

1. Set BS1 to "1". This selects high data byte.
2. Set XA1, XA0 to "01". This enables data loading.
3. Set DATA = Data high byte (0x00 - 0xFF).
4. Give XTAL1 a positive pulse. This loads the data byte.

E. Latch Data

1. Set BS1 to "1". This selects high data byte.
2. Give PAGESL a positive pulse. This latches the data bytes. (See [Figure 27-3](#) for signal waveforms)

F. Repeat B through E until the entire buffer is filled or until all data within the page is loaded.

While the lower bits in the address are mapped to words within the page, the higher bits address the pages within the FLASH. This is illustrated in [Figure 27-2 on page 304](#). Note that if less than eight bits are required to address words in the page (pagesize < 256), the most significant bit(s) in the address low byte are used to address the page when performing a Page Write.

G. Load Address High byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS1 to "1". This selects high address.
3. Set DATA = Address high byte (0x00 - 0xFF).
4. Give XTAL1 a positive pulse. This loads the address high byte.

H. Program Page

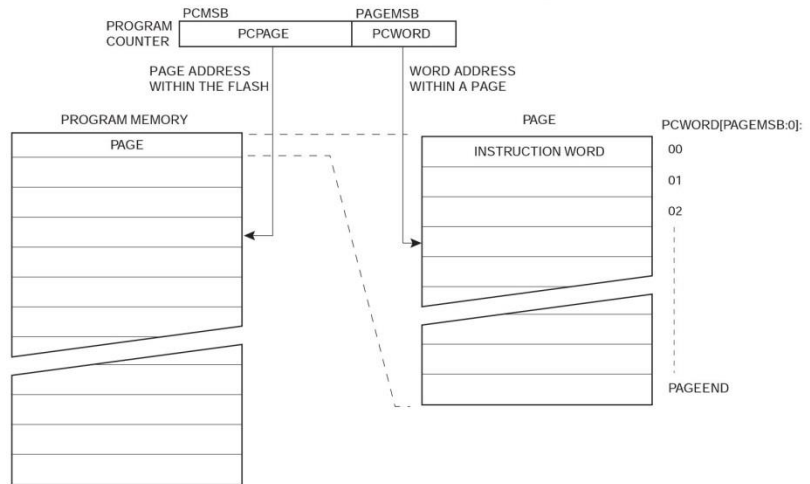
1. Give \overline{WR} a negative pulse. This starts programming of the entire page of data. RDY/ \overline{BSY} goes low.
2. Wait until RDY/ \overline{BSY} goes high (See [Figure 27-3](#) for signal waveforms).

I. Repeat B through H until the entire Flash is programmed or until all data has been programmed.

J. End Page Programming

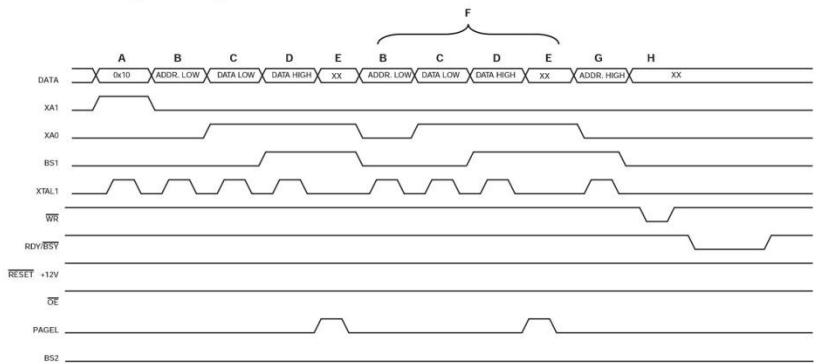
1. Set XA1, XA0 to "10". This enables command loading.
2. Set DATA to "0000 0000". This is the command for No Operation.
3. Give XTAL1 a positive pulse. This loads the command, and the internal write signals are reset.

Figure 27-2. Addressing the Flash Which is Organized in Pages⁽¹⁾



Note: 1. PCPAGE and PCWORD are listed in Table 27-11 on page 299.

Figure 27-3. Programming the Flash Waveforms⁽¹⁾



Note: 1. "XX" is don't care. The letters refer to the programming description above.

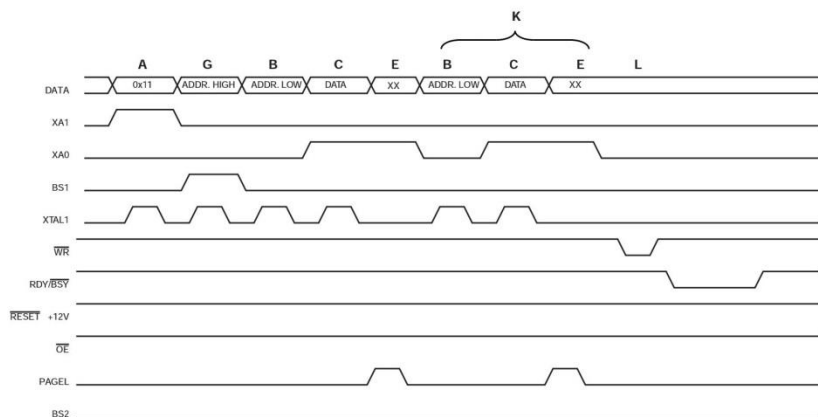
27.7.5 Programming the EEPROM

The EEPROM is organized in pages, see Table 27-12 on page 299. When programming the EEPROM, the program data is latched into a page buffer. This allows one page of data to be programmed simultaneously. The programming algorithm for the EEPROM data memory is as follows (refer to "Programming the Flash" on page 302 for details on Command, Address and Data loading):

1. A: Load Command "0001 0001".
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
4. C: Load Data (0x00 - 0xFF).

5. E: Latch data (give PAGESL a positive pulse).
 - K: Repeat 3 through 5 until the entire buffer is filled.
 - L: Program EEPROM page
1. Set BS1 to "0".
 2. Give \overline{WR} a negative pulse. This starts programming of the EEPROM page. RDY/ \overline{BSY} goes low.
 3. Wait until RDY/ \overline{BSY} goes high before programming the next page (See Figure 27-4 for signal waveforms).

Figure 27-4. Programming the EEPROM Waveforms



27.7.6 Reading the Flash

The algorithm for reading the Flash memory is as follows (refer to "Programming the Flash" on page 302 for details on Command and Address loading):

1. A: Load Command "0000 0010".
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
4. Set \overline{OE} to "0", and BS1 to "0". The Flash word low byte can now be read at DATA.
5. Set BS1 to "1". The Flash word high byte can now be read at DATA.
6. Set \overline{OE} to "1".

27.7.7 Reading the EEPROM

The algorithm for reading the EEPROM memory is as follows (refer to "Programming the Flash" on page 302 for details on Command and Address loading):

1. A: Load Command "0000 0011".
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
4. Set \overline{OE} to "0", and BS1 to "0". The EEPROM Data byte can now be read at DATA.
5. Set \overline{OE} to "1".

27.7.8 Programming the Fuse Low Bits

The algorithm for programming the Fuse Low bits is as follows (refer to "Programming the Flash" on page 302 for details on Command and Data loading):

1. A: Load Command "0100 0000".
2. C: Load Data Low Byte. Bit n = "0" programs and bit n = "1" erases the Fuse bit.
3. Give \overline{WR} a negative pulse and wait for RDY/ \overline{BSY} to go high.

27.7.9 Programming the Fuse High Bits

The algorithm for programming the Fuse High bits is as follows (refer to "Programming the Flash" on page 302 for details on Command and Data loading):

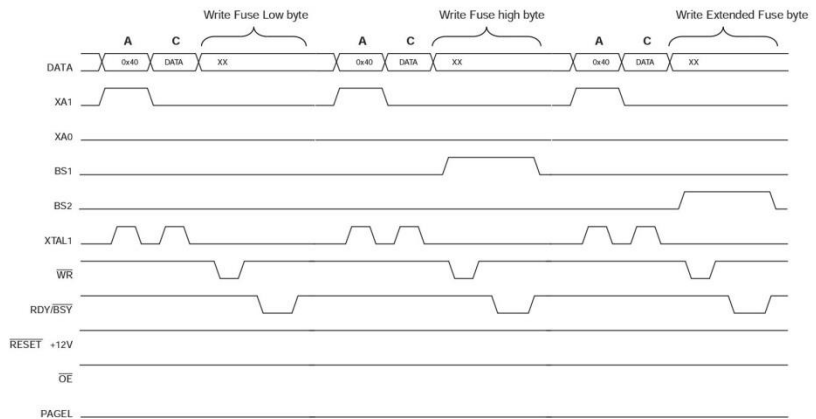
1. A: Load Command "0100 0000".
2. C: Load Data Low Byte. Bit n = "0" programs and bit n = "1" erases the Fuse bit.
3. Set BS1 to "1" and BS2 to "0". This selects high data byte.
4. Give \overline{WR} a negative pulse and wait for RDY/ \overline{BSY} to go high.
5. Set BS1 to "0". This selects low data byte.

27.7.10 Programming the Extended Fuse Bits

The algorithm for programming the Extended Fuse bits is as follows (refer to "Programming the Flash" on page 302 for details on Command and Data loading):

1. 1. A: Load Command "0100 0000".
2. 2. C: Load Data Low Byte. Bit n = "0" programs and bit n = "1" erases the Fuse bit.
3. 3. Set BS1 to "0" and BS2 to "1". This selects extended data byte.
4. 4. Give \overline{WR} a negative pulse and wait for RDY/ \overline{BSY} to go high.
5. 5. Set BS2 to "0". This selects low data byte.

Figure 27-5. Programming the FUSES Waveforms



27.7.11 Programming the Lock Bits

The algorithm for programming the Lock bits is as follows (refer to "Programming the Flash" on page 302 for details on Command and Data loading):

1. A: Load Command "0010 0000".
2. C: Load Data Low Byte. Bit n = "0" programs the Lock bit. If LB mode 3 is programmed (LB1 and LB2 is programmed), it is not possible to program the Boot Lock bits by any External Programming mode.
3. Give \overline{WR} a negative pulse and wait for RDY/\overline{BSY} to go high.

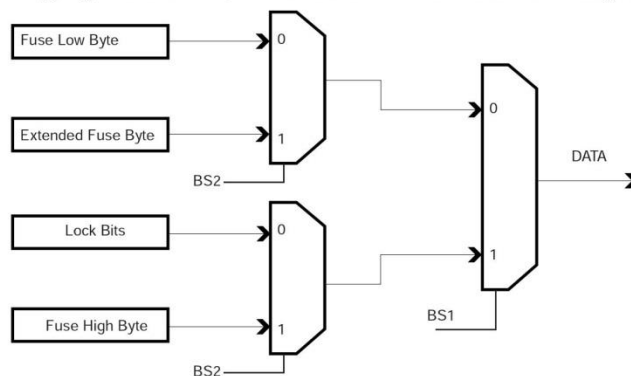
The Lock bits can only be cleared by executing Chip Erase.

27.7.12 Reading the Fuse and Lock Bits

The algorithm for reading the Fuse and Lock bits is as follows (refer to "Programming the Flash" on page 302 for details on Command loading):

1. A: Load Command "0000 0100".
2. Set \overline{OE} to "0", BS2 to "0" and BS1 to "0". The status of the Fuse Low bits can now be read at DATA ("0" means programmed).
3. Set \overline{OE} to "0", BS2 to "1" and BS1 to "1". The status of the Fuse High bits can now be read at DATA ("0" means programmed).
4. Set \overline{OE} to "0", BS2 to "1", and BS1 to "0". The status of the Extended Fuse bits can now be read at DATA ("0" means programmed).
5. Set \overline{OE} to "0", BS2 to "0" and BS1 to "1". The status of the Lock bits can now be read at DATA ("0" means programmed).
6. Set \overline{OE} to "1".

Figure 27-6. Mapping Between BS1, BS2 and the Fuse and Lock Bits During Read



27.7.13 Reading the Signature Bytes

The algorithm for reading the Signature bytes is as follows (refer to "Programming the Flash" on page 302 for details on Command and Address loading):

1. A: Load Command "0000 1000".
2. B: Load Address Low Byte (0x00 - 0x02).
3. Set \overline{OE} to "0", and BS1 to "0". The selected Signature byte can now be read at DATA.
4. Set \overline{OE} to "1".

27.7.14 Reading the Calibration Byte

The algorithm for reading the Calibration byte is as follows (refer to "Programming the Flash" on page 302 for details on Command and Address loading):

1. A: Load Command "0000 1000".
2. B: Load Address Low Byte, 0x00.
3. Set \overline{OE} to "0", and BS1 to "1". The Calibration byte can now be read at DATA.
4. Set \overline{OE} to "1".

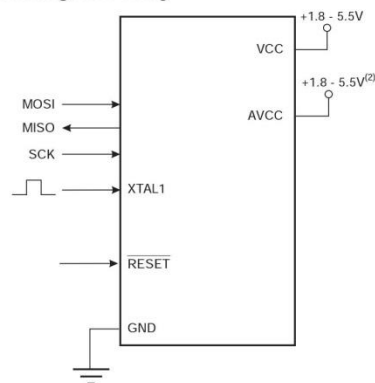
27.7.15 Parallel Programming Characteristics

For characteristics of the Parallel Programming, see "Parallel Programming Characteristics" on page 324.

27.8 Serial Downloading

Both the Flash and EEPROM memory arrays can be programmed using the serial SPI bus while \overline{RESET} is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After \overline{RESET} is set low, the Programming Enable instruction needs to be executed first before program/erase operations can be executed. NOTE, in Table 27-17 on page 309, the pin mapping for SPI programming is listed. Not all parts use the SPI pins dedicated for the internal SPI interface.

Figure 27-7. Serial Programming and Verify⁽¹⁾



- Notes:
1. If the device is clocked by the internal Oscillator, it is no need to connect a clock source to the XTAL1 pin.
 2. $V_{CC} - 0.3V < AV_{CC} < V_{CC} + 0.3V$, however, AV_{CC} should always be within 1.8 - 5.5V

When programming the EEPROM, an auto-erase cycle is built into the self-timed programming operation (in the Serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation turns the content of every memory location in both the Program and EEPROM arrays into 0xFF.

Depending on CKSEL Fuses, a valid clock must be present. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

- Low: > 2 CPU clock cycles for $f_{ck} < 12$ MHz, 3 CPU clock cycles for $f_{ck} \geq 12$ MHz
 High: > 2 CPU clock cycles for $f_{ck} < 12$ MHz, 3 CPU clock cycles for $f_{ck} \geq 12$ MHz

27.8.1 Serial Programming Pin Mapping

Table 27-17. Pin Mapping Serial Programming

Symbol	Pins	I/O	Description
MOSI	PB3	I	Serial Data in
MISO	PB4	O	Serial Data out
SCK	PB5	I	Serial Clock

27.8.2 Serial Programming Algorithm

When writing serial data to the ATmega48PA/88PA/168PA/328P, data is clocked on the rising edge of SCK.

When reading data from the ATmega48PA/88PA/168PA/328P, data is clocked on the falling edge of SCK. See [Figure 27-9](#) for timing details.

To program and verify the ATmega48PA/88PA/168PA/328P in the serial programming mode, the following sequence is recommended (See Serial Programming Instruction set in [Table 27-19 on page 310](#)):

1. Power-up sequence:
Apply power between V_{CC} and GND while \overline{RESET} and SCK are set to "0". In some systems, the programmer can not guarantee that SCK is held low during power-up. In this case, \overline{RESET} must be given a positive pulse of at least two CPU clock cycles duration after SCK has been set to "0".
2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to pin MOSI.
3. The serial programming instructions will not work if the communication is out of synchronization. When in sync, the second byte (0x53), will echo back when issuing the third byte of the Programming Enable instruction. Whether the echo is correct or not, all four bytes of the instruction must be transmitted. If the 0x53 did not echo back, give \overline{RESET} a positive pulse and issue a new Programming Enable command.
4. The Flash is programmed one page at a time. The memory page is loaded one byte at a time by supplying the 6 LSB of the address and data together with the Load Program Memory Page instruction. To ensure correct loading of the page, the data low byte must be loaded before data high byte is applied for a given address. The Program Memory Page is stored by loading the Write Program Memory Page instruction with the 7 MSB of the address. If polling (RDY/BSY) is not used, the user must wait at least t_{WD_FLASH} before issuing the next page (See [Table 27-18](#)). Accessing the serial programming interface before the Flash write operation completes can result in incorrect programming.
5. **A:** The EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. An EEPROM memory location is first automatically erased before new data is written. If polling (RDY/BSY) is not used, the user must wait at least t_{WD_EEPROM} before issuing the next byte (See [Table 27-18](#)). In a chip erased device, no 0xFFs in the data file(s) need to be programmed.
B: The EEPROM array is programmed one page at a time. The Memory page is loaded one byte at a time by supplying the 6 LSB of the address and data together with the Load EEPROM Memory Page instruction. The EEPROM Memory Page is stored by loading the Write EEPROM Memory Page Instruction with the 7 MSB of the address. When using EEPROM page access only byte locations loaded with the Load EEPROM Memory Page instruction is altered. The remaining locations remain unchanged. If polling (RDY/BSY) is

not used, the used must wait at least t_{WD_EEPROM} before issuing the next byte (See [Table 27-18](#)). In a chip erased device, no 0xFF in the data file(s) need to be programmed.

6. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO.
7. At the end of the programming session, \overline{RESET} can be set high to commence normal operation.
8. Power-off sequence (if needed):
Set \overline{RESET} to "1".
Turn V_{CC} power off.

Table 27-18. Typical Wait Delay Before Writing the Next Flash or EEPROM Location

Symbol	Minimum Wait Delay
t_{WD_FLASH}	4.5 ms
t_{WD_EEPROM}	3.6 ms
t_{WD_ERASE}	9.0 ms

27.8.3 Serial Programming Instruction set

[Table 27-19](#) on page 310 and [Figure 27-8](#) on page 312 describes the Instruction set.

Table 27-19. Serial Programming Instruction Set (Hexadecimal values)

Instruction/Operation	Instruction Format			
	Byte 1	Byte 2	Byte 3	Byte 4
Programming Enable	\$AC	\$53	\$00	\$00
Chip Erase (Program Memory/EEPROM)	\$AC	\$80	\$00	\$00
Poll RDY/BSY	\$F0	\$00	\$00	data byte out
Load Instructions				
Load Extended Address byte ⁽¹⁾	\$4D	\$00	Extended adr	\$00
Load Program Memory Page, High byte	\$48	\$00	adr LSB	high data byte in
Load Program Memory Page, Low byte	\$40	\$00	adr LSB	low data byte in
Load EEPROM Memory Page (page access)	\$C1	\$00	0000 000aa	data byte in
Read Instructions				
Read Program Memory, High byte	\$28	adr MSB	adr LSB	high data byte out
Read Program Memory, Low byte	\$20	adr MSB	adr LSB	low data byte out
Read EEPROM Memory	\$A0	0000 000aa	aaaa aaaa	data byte out
Read Lock bits	\$58	\$00	\$00	data byte out
Read Signature Byte	\$30	\$00	0000 000aa	data byte out
Read Fuse bits	\$50	\$00	\$00	data byte out
Read Fuse High bits	\$58	\$08	\$00	data byte out
Read Extended Fuse Bits	\$50	\$08	\$00	data byte out
Read Calibration Byte	\$38	\$00	\$00	data byte out

Table 27-19. Serial Programming Instruction Set (Hexadecimal values) (Continued)

Instruction/Operation	Instruction Format			
	Byte 1	Byte 2	Byte 3	Byte 4
Write Instructions⁽⁶⁾				
Write Program Memory Page	\$4C	adr MSB	adr LSB	\$00
Write EEPROM Memory	\$C0	0000 00aa	aaaa aaaa	data byte in
Write EEPROM Memory Page (page access)	\$C2	0000 00aa	aaaa aa00	\$00
Write Lock bits	\$AC	\$E0	\$00	data byte in
Write Fuse bits	\$AC	\$A0	\$00	data byte in
Write Fuse High bits	\$AC	\$A8	\$00	data byte in
Write Extended Fuse Bits	\$AC	\$A4	\$00	data byte in

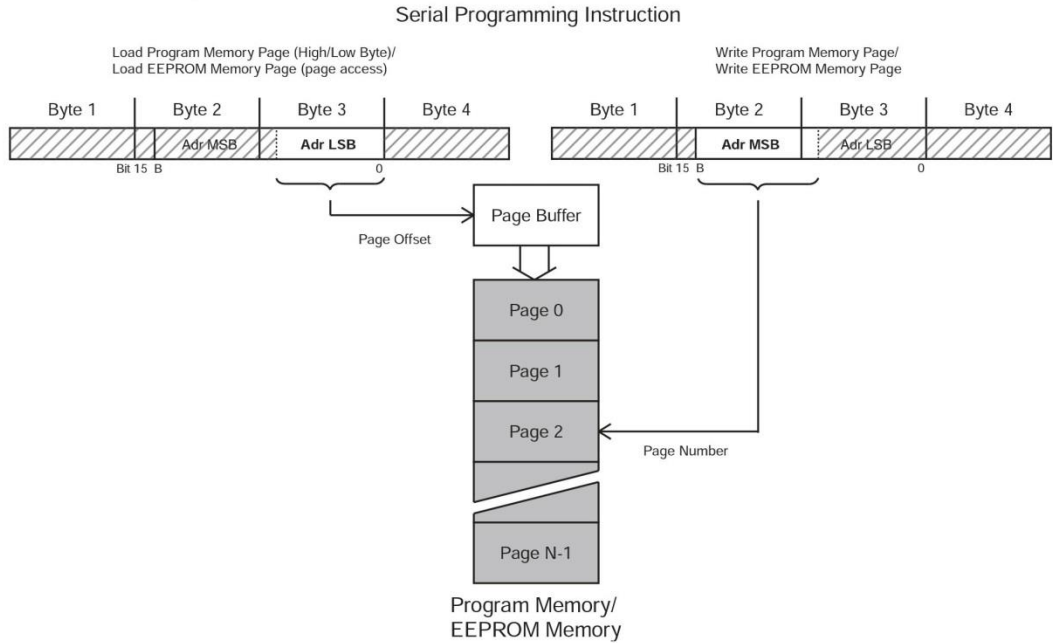
- Notes:
1. Not all instructions are applicable for all parts.
 2. a = address.
 3. Bits are programmed '0', unprogrammed '1'.
 4. To ensure future compatibility, unused Fuses and Lock bits should be unprogrammed ('1').
 5. Refer to the corresponding section for Fuse and Lock bits, Calibration and Signature bytes and Page size.
 6. Instructions accessing program memory use a word address. This address may be random within the page range.
 7. See <http://www.atmel.com/avr> for Application Notes regarding programming and programmers.

If the LSB in RDY/BSY data byte out is '1', a programming operation is still pending. Wait until this bit returns '0' before the next instruction is carried out.

Within the same page, the low data byte must be loaded prior to the high data byte.

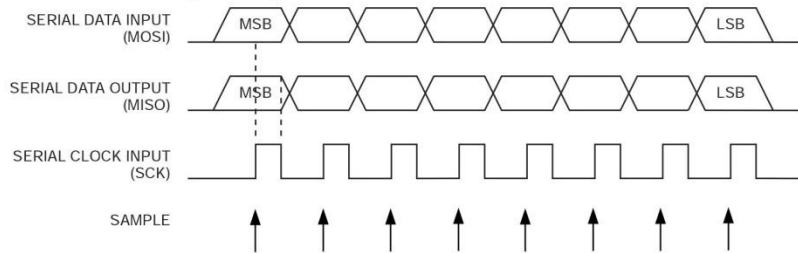
After data is loaded to the page buffer, program the EEPROM page, see [Figure 27-8 on page 312](#).

Figure 27-8. Serial Programming Instruction example



27.8.4 SPI Serial Programming Characteristics

Figure 27-9. Serial Programming Waveforms



For characteristics of the SPI module see "SPI Timing Characteristics" on page 319.

28. Electrical Characteristics

28.1 Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground.....	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins.....	200.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

28.2 DC Characteristics

$T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 1.8V$ to $5.5V$ (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage, except XTAL1 and $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
V_{IH}	Input High Voltage, except XTAL1 and $\overline{\text{RESET}}$ pins	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{IL1}	Input Low Voltage, XTAL1 pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.1V_{CC}^{(1)}$	V
V_{IH1}	Input High Voltage, XTAL1 pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.8V_{CC}^{(2)}$ $0.7V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{IL2}	Input Low Voltage, $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.1V_{CC}^{(1)}$	V
V_{IH2}	Input High Voltage, $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8V - 5.5V$	$0.9V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
V_{IL3}	Input Low Voltage, $\overline{\text{RESET}}$ pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
V_{IH3}	Input High Voltage, $\overline{\text{RESET}}$ pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽³⁾ except $\overline{\text{RESET}}$ pin	$I_{OL} = 20\text{ mA}$, $V_{CC} = 5V$ $I_{OL} = 10\text{ mA}$, $V_{CC} = 3V$			0.9 0.6	V
V_{OH}	Output High Voltage ⁽⁴⁾ except Reset pin	$I_{OH} = -20\text{ mA}$, $V_{CC} = 5V$ $I_{OH} = -10\text{ mA}$, $V_{CC} = 3V$	4.2 2.3			V
I_{IL}	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$, pin low (absolute value)			1	μA
I_{IH}	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$, pin high (absolute value)			1	μA

ATmega48PA/88PA/168PA/328P

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted) (Continued)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
R _{RST}	Reset Pull-up Resistor		30		60	kΩ
R _{PU}	I/O Pin Pull-up Resistor		20		50	kΩ
V _{ACIO}	Analog Comparator Input Offset Voltage	V _{CC} = 5V V _{in} = V _{CC} /2		<10	40	mV
I _{ACLK}	Analog Comparator Input Leakage Current	V _{CC} = 5V V _{in} = V _{CC} /2	-50		50	nA
t _{ACID}	Analog Comparator Propagation Delay	V _{CC} = 2.7V V _{CC} = 4.0V		750 500		ns

- Notes:
- "Max" means the highest value where the pin is guaranteed to be read as low
 - "Min" means the lowest value where the pin is guaranteed to be read as high
 - Although each I/O port can sink more than the test conditions (20 mA at V_{CC} = 5V, 10 mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
ATmega48PA/88PA/168PA/328P:
1] The sum of all I_{OL}, for ports C0 - C5, ADC7, ADC6 should not exceed 100 mA.
2] The sum of all I_{OL}, for ports B0 - B5, D5 - D7, XTAL1, XTAL2 should not exceed 100 mA.
3] The sum of all I_{OL}, for ports D0 - D4, $\overline{\text{RESET}}$ should not exceed 100 mA.
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
 - Although each I/O port can source more than the test conditions (20 mA at V_{CC} = 5V, 10 mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
ATmega48PA/88PA/168PA/328P:
1] The sum of all I_{OH}, for ports C0 - C5, D0 - D4, ADC7, $\overline{\text{RESET}}$ should not exceed 150 mA.
2] The sum of all I_{OH}, for ports B0 - B5, D5 - D7, ADC6, XTAL1, XTAL2 should not exceed 150 mA.
If I_{OH} exceeds the test condition, V_{OH} may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.

28.2.1 ATmega48PA DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ. ⁽²⁾	Max.	Units	
I _{CC}	Power Supply Current ⁽¹⁾	Active 1 MHz, V _{CC} = 2V		0.2	0.5	mA	
		Active 4 MHz, V _{CC} = 3V		1.2	2.5	mA	
		Active 8 MHz, V _{CC} = 5V		4.0	9	mA	
		Idle 1 MHz, V _{CC} = 2V		0.03	0.15	mA	
		Idle 4 MHz, V _{CC} = 3V		0.21	0.7	mA	
		Idle 8 MHz, V _{CC} = 5V		0.9	2.7	mA	
	Power-save mode ⁽³⁾	32 kHz TOSC enabled, V _{CC} = 1.8V			0.75		μA
		32 kHz TOSC enabled, V _{CC} = 3V			0.9		μA
	Power-down mode ⁽³⁾	WDT enabled, V _{CC} = 3V			3.9	8	μA
		WDT disabled, V _{CC} = 3V			0.1	2	μA

- Notes:
- Values with "Minimizing Power Consumption" enabled (0xFF).
 - Typical values at 25°C. Maximum values are characterized values and not test limits in production.
 - The current consumption values include input leakage current.

28.2.2 ATmega88PA DC Characteristics

$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 1.8\text{V}$ to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ. ⁽²⁾	Max.	Units
I_{CC}	Power Supply Current ⁽¹⁾	Active 1 MHz, $V_{CC} = 2\text{V}$		0.2	0.5	mA
		Active 4 MHz, $V_{CC} = 3\text{V}$		1.2	2.5	mA
		Active 8 MHz, $V_{CC} = 5\text{V}$		4.1	9	mA
		Idle 1 MHz, $V_{CC} = 2\text{V}$		0.03	0.15	mA
		Idle 4 MHz, $V_{CC} = 3\text{V}$		0.18	0.7	mA
		Idle 8 MHz, $V_{CC} = 5\text{V}$		0.8	2.7	mA
	Power-save mode ⁽³⁾	32 kHz TOSC enabled, $V_{CC} = 1.8\text{V}$		0.8		μA
		32 kHz TOSC enabled, $V_{CC} = 3\text{V}$		0.9		μA
	Power-down mode ⁽³⁾	WDT enabled, $V_{CC} = 3\text{V}$		3.9	8	μA
		WDT disabled, $V_{CC} = 3\text{V}$		0.1	2	μA

- Notes: 1. Values with "Minimizing Power Consumption" enabled (0xFF).
 2. Typical values at 25°C . Maximum values are test limits in production.
 3. The current consumption values include input leakage current.

28.2.3 ATmega168PA DC Characteristics

$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 1.8\text{V}$ to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ. ⁽²⁾	Max.	Units
I_{CC}	Power Supply Current ⁽¹⁾	Active 1 MHz, $V_{CC} = 2\text{V}$		0.2	0.5	mA
		Active 4 MHz, $V_{CC} = 3\text{V}$		1.2	2.5	mA
		Active 8 MHz, $V_{CC} = 5\text{V}$		4.2	9	mA
		Idle 1 MHz, $V_{CC} = 2\text{V}$		0.03	0.15	mA
		Idle 4 MHz, $V_{CC} = 3\text{V}$		0.2	0.7	mA
		Idle 8 MHz, $V_{CC} = 5\text{V}$		0.9	2.7	mA
	Power-save mode ⁽³⁾	32 kHz TOSC enabled, $V_{CC} = 1.8\text{V}$		0.75		μA
		32 kHz TOSC enabled, $V_{CC} = 3\text{V}$		0.83		μA
	Power-down mode ⁽³⁾	WDT enabled, $V_{CC} = 3\text{V}$		4.1	8	μA
		WDT disabled, $V_{CC} = 3\text{V}$		0.1	2	μA

- Notes: 1. Values with "Minimizing Power Consumption" enabled (0xFF).
 2. Typical values at 25°C . Maximum values are test limits in production.
 3. The current consumption values include input leakage current.

28.2.4 ATmega328P DC Characteristics

$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 1.8\text{V}$ to 5.5V (unless otherwise noted)

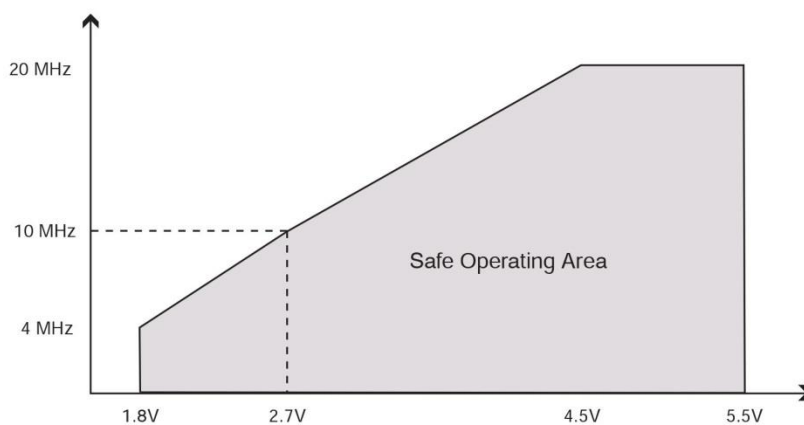
Symbol	Parameter	Condition	Min.	Typ. ⁽²⁾	Max.	Units	
I_{CC}	Power Supply Current ⁽¹⁾	Active 1 MHz, $V_{CC} = 2\text{V}$		0.3	0.5	mA	
		Active 4 MHz, $V_{CC} = 3\text{V}$		1.7	2.5	mA	
		Active 8 MHz, $V_{CC} = 5\text{V}$		5.2	9	mA	
		Idle 1 MHz, $V_{CC} = 2\text{V}$		0.04	0.15	mA	
		Idle 4 MHz, $V_{CC} = 3\text{V}$		0.3	0.7	mA	
		Idle 8 MHz, $V_{CC} = 5\text{V}$		1.2	2.7	mA	
	Power-save mode ⁽³⁾⁽⁴⁾	32 kHz TOSC enabled, $V_{CC} = 1.8\text{V}$			0.8	1.6	μA
		32 kHz TOSC enabled, $V_{CC} = 3\text{V}$			0.9	2.6	μA
	Power-down mode ⁽³⁾	WDT enabled, $V_{CC} = 3\text{V}$			4.2	8	μA
		WDT disabled, $V_{CC} = 3\text{V}$			0.1	2	μA

- Notes:
1. Values with "Minimizing Power Consumption" enabled (0xFF).
 2. Typical values at 25°C . Maximum values are test limits in production.
 3. The current consumption values include input leakage current.
 4. Maximum values are characterized values and not test limits in production.

28.3 Speed Grades

Maximum frequency is dependent on V_{CC} . As shown in Figure 28-1, the Maximum Frequency vs. V_{CC} curve is linear between $1.8\text{V} < V_{CC} < 2.7\text{V}$ and between $2.7\text{V} < V_{CC} < 4.5\text{V}$.

Figure 28-1. Maximum Frequency vs. V_{CC}



28.4 Clock Characteristics

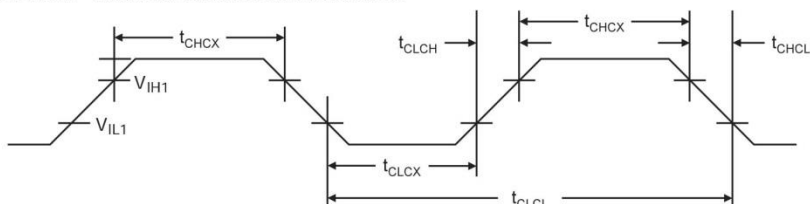
28.4.1 Calibrated Internal RC Oscillator Accuracy

Table 28-1. Calibration Accuracy of Internal RC Oscillator

	Frequency	V _{CC}	Temperature	Calibration Accuracy
Factory Calibration	8.0 MHz	3V	25°C	±10%
User Calibration	7.3 - 8.1 MHz	1.8V - 5.5V	-40°C - 85°C	±1%

28.4.2 External Clock Drive Waveforms

Figure 28-2. External Clock Drive Waveforms



28.4.3 External Clock Drive

Table 28-2. External Clock Drive

Symbol	Parameter	V _{CC} = 1.8 - 5.5V		V _{CC} = 2.7 - 5.5V		V _{CC} = 4.5 - 5.5V		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
$1/t_{CLCL}$	Oscillator Frequency	0	4	0	10	0	20	MHz
t_{CLCL}	Clock Period	250		100		50		ns
t_{CHCX}	High Time	100		40		20		ns
t_{CLCX}	Low Time	100		40		20		ns
t_{CLCH}	Rise Time		2.0		1.6		0.5	μs
t_{CHCL}	Fall Time		2.0		1.6		0.5	μs
Δt_{CLCL}	Change in period from one clock cycle to the next		2		2		2	%

Note: All DC Characteristics contained in this datasheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual silicon.

28.5 System and Reset Characteristics

Table 28-3. Reset, Brown-out and Internal Voltage Characteristics⁽¹⁾

Symbol	Parameter		Min	Typ	Max	Units
V _{POT}	Power-on Reset Threshold Voltage (rising)		1.1	1.4	1.6	V
	Power-on Reset Threshold Voltage (falling) ⁽²⁾		0.6	1.3	1.6	V
SR _{ON}	Power-on Slope Rate		0.01		10	V/ms
V _{RST}	RESET Pin Threshold Voltage		0.2 V _{CC}		0.9 V _{CC}	V
t _{RST}	Minimum pulse width on RESET Pin				2.5	μs
V _{HYST}	Brown-out Detector Hysteresis			50		mV
t _{BOD}	Min Pulse Width on Brown-out Reset			2		μs
V _{BG}	Bandgap reference voltage	V _{CC} =2.7 T _A =25°C	1.0	1.1	1.2	V
t _{BG}	Bandgap reference start-up time	V _{CC} =2.7 T _A =25°C		40	70	μs
I _{BG}	Bandgap reference current consumption	V _{CC} =2.7 T _A =25°C		10		μA

Notes: 1. Values are guidelines only.
2. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling)

Table 28-4. BODLEVEL Fuse Coding⁽¹⁾

BODLEVEL 2:0 Fuses	Min V _{BOT}	Typ V _{BOT}	Max V _{BOT}	Units
111	BOD Disabled			
110	1.7	1.8	2.0	V
101	2.5	2.7	2.9	
100	4.1	4.3	4.5	
011	Reserved			
010				
001				
000				

Notes: 1. V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to V_{CC} = V_{BOT} during the production test. This guarantees that a Brown-Out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 110, 101 and 100.

28.6 SPI Timing Characteristics

See Figure 28-3 and Figure 28-4 for details.

Table 28-5. SPI Timing Parameters

	Description	Mode	Min	Typ	Max	
1	SCK period	Master		See Table 18-5		ns
2	SCK high/low	Master		50% duty cycle		
3	Rise/Fall time	Master		3.6		
4	Setup	Master		10		
5	Hold	Master		10		
6	Out to SCK	Master		$0.5 \cdot t_{sck}$		
7	SCK to out	Master		10		
8	SCK to out high	Master		10		
9	\overline{SS} low to out	Slave		15		
10	SCK period	Slave	$4 \cdot t_{ck}$			
11	SCK high/low ⁽¹⁾	Slave	$2 \cdot t_{ck}$			
12	Rise/Fall time	Slave			1600	
13	Setup	Slave	10			
14	Hold	Slave	t_{ck}			
15	SCK to out	Slave		15		
16	SCK to \overline{SS} high	Slave	20			
17	\overline{SS} high to tri-state	Slave		10		
18	\overline{SS} low to SCK	Slave	20			

- Note:
- In SPI Programming mode the minimum SCK high/low period is:
 - $2 t_{CLCL}$ for $f_{CK} < 12$ MHz
 - $3 t_{CLCL}$ for $f_{CK} > 12$ MHz
 - All DC Characteristics contained in this datasheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual silicon.

Figure 28-3. SPI Interface Timing Requirements (Master Mode)

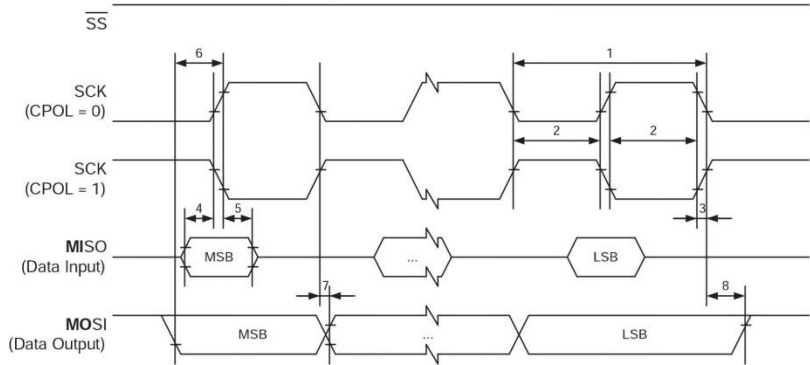
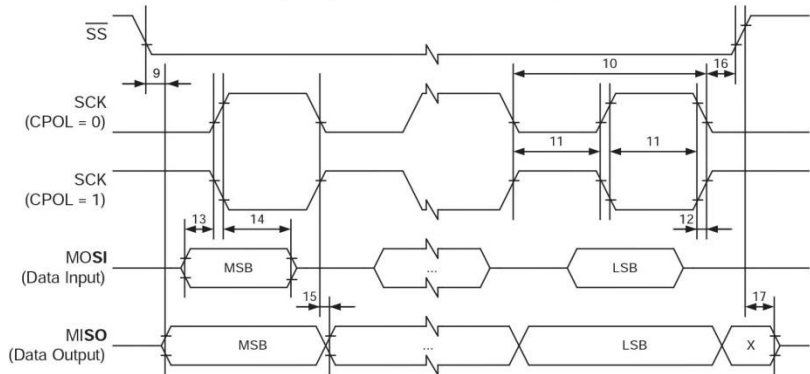


Figure 28-4. SPI Interface Timing Requirements (Slave Mode)



28.7 2-wire Serial Interface Characteristics

Table 28-6 describes the requirements for devices connected to the 2-wire Serial Bus. The ATmega48PA/88PA/168PA/328P 2-wire Serial Interface meets or exceeds these requirements under the noted conditions.

Timing symbols refer to Figure 28-5.

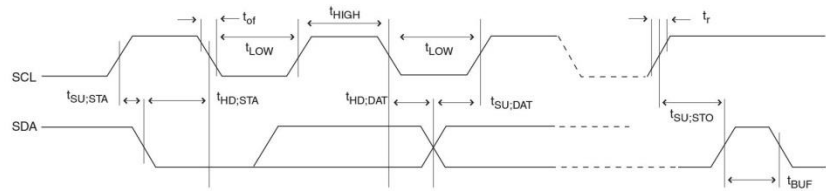
Table 28-6. 2-wire Serial Bus Requirements

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	$0.3 V_{CC}$	V
V_{IH}	Input High-voltage		$0.7 V_{CC}$	$V_{CC} + 0.5$	V
$V_{hys}^{(1)}$	Hysteresis of Schmitt Trigger Inputs		$0.05 V_{CC}^{(2)}$	–	V
$V_{OL}^{(1)}$	Output Low-voltage	3 mA sink current	0	0.4	V
$t_r^{(1)}$	Rise Time for both SDA and SCL		$20 + 0.1C_b^{(3)(2)}$	300	ns
$t_{of}^{(1)}$	Output Fall Time from V_{IHmin} to V_{ILmax}	$10 \text{ pF} < C_b < 400 \text{ pF}^{(3)}$	$20 + 0.1C_b^{(3)(2)}$	250	ns
$t_{SP}^{(1)}$	Spikes Suppressed by Input Filter		0	$50^{(2)}$	ns
I_i	Input Current each I/O Pin	$0.1V_{CC} < V_i < 0.9V_{CC}$	-10	10	μA
$C_i^{(1)}$	Capacitance for each I/O Pin		–	10	pF
f_{SCL}	SCL Clock Frequency	$f_{CK}^{(4)} > \max(16f_{SCL}, 250\text{kHz})^{(5)}$	0	400	kHz
R_p	Value of Pull-up resistor	$f_{SCL} \leq 100 \text{ kHz}$	$\frac{V_{CC} - 0.4V}{3\text{mA}}$	$\frac{1000\text{ns}}{C_b}$	Ω
		$f_{SCL} > 100 \text{ kHz}$	$\frac{V_{CC} - 0.4V}{3\text{mA}}$	$\frac{300\text{ns}}{C_b}$	Ω
$t_{HD;STA}$	Hold Time (repeated) START Condition	$f_{SCL} \leq 100 \text{ kHz}$	4.0	–	μs
		$f_{SCL} > 100 \text{ kHz}$	0.6	–	μs
t_{LOW}	Low Period of the SCL Clock	$f_{SCL} \leq 100 \text{ kHz}$	4.7	–	μs
		$f_{SCL} > 100 \text{ kHz}$	1.3	–	μs
t_{HIGH}	High period of the SCL clock	$f_{SCL} \leq 100 \text{ kHz}$	4.0	–	μs
		$f_{SCL} > 100 \text{ kHz}$	0.6	–	μs
$t_{SU;STA}$	Set-up time for a repeated START condition	$f_{SCL} \leq 100 \text{ kHz}$	4.7	–	μs
		$f_{SCL} > 100 \text{ kHz}$	0.6	–	μs
$t_{HD;DAT}$	Data hold time	$f_{SCL} \leq 100 \text{ kHz}$	0	3.45	μs
		$f_{SCL} > 100 \text{ kHz}$	0	0.9	μs
$t_{SU;DAT}$	Data setup time	$f_{SCL} \leq 100 \text{ kHz}$	250	–	ns
		$f_{SCL} > 100 \text{ kHz}$	100	–	ns
$t_{SU;STO}$	Setup time for STOP condition	$f_{SCL} \leq 100 \text{ kHz}$	4.0	–	μs
		$f_{SCL} > 100 \text{ kHz}$	0.6	–	μs
t_{BUF}	Bus free time between a STOP and START condition	$f_{SCL} \leq 100 \text{ kHz}$	4.7	–	μs
		$f_{SCL} > 100 \text{ kHz}$	1.3	–	μs

Notes: 1. In ATmega48PA/88PA/168PA/328P, this parameter is characterized and not 100% tested.
 2. Required only for $f_{SCL} > 100 \text{ kHz}$.

3. C_b = capacitance of one bus line in pF.
4. f_{CK} = CPU clock frequency
5. This requirement applies to all ATmega48PA/88PA/168PA/328P 2-wire Serial Interface operation. Other devices connected to the 2-wire Serial Bus need only obey the general f_{SCL} requirement.

Figure 28-5. 2-wire Serial Bus Timing



28.8 ADC Characteristics

Table 28-7. ADC Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Resolution			10		Bits
	Absolute accuracy (Including INL, DNL, quantization error, gain and offset error)	$V_{REF} = 4V, V_{CC} = 4V,$ ADC clock = 200 kHz		2		LSB
		$V_{REF} = 4V, V_{CC} = 4V,$ ADC clock = 1 MHz		4.5		LSB
		$V_{REF} = 4V, V_{CC} = 4V,$ ADC clock = 200 kHz Noise Reduction Mode		2		LSB
		$V_{REF} = 4V, V_{CC} = 4V,$ ADC clock = 1 MHz Noise Reduction Mode		4.5		LSB
	Integral Non-Linearity (INL)	$V_{REF} = 4V, V_{CC} = 4V,$ ADC clock = 200 kHz		0.5		LSB
	Differential Non-Linearity (DNL)	$V_{REF} = 4V, V_{CC} = 4V,$ ADC clock = 200 kHz		0.25		LSB
	Gain Error	$V_{REF} = 4V, V_{CC} = 4V,$ ADC clock = 200 kHz		2		LSB
	Offset Error	$V_{REF} = 4V, V_{CC} = 4V,$ ADC clock = 200 kHz		2		LSB
	Conversion Time	Free Running Conversion	13		260	μs
	Clock Frequency		50		1000	kHz
$AV_{CC}^{(1)}$	Analog Supply Voltage		$V_{CC} - 0.3$		$V_{CC} + 0.3$	V
V_{REF}	Reference Voltage		1.0		AV_{CC}	V
V_{IN}	Input Voltage		GND		V_{REF}	V
	Input Bandwidth			38.5		kHz
V_{INT}	Internal Voltage Reference		1.0	1.1	1.2	V
R_{REF}	Reference Input Resistance			32		k Ω
R_{AIN}	Analog Input Resistance			100		M Ω

Note: 1. AV_{CC} absolute min/max: 1.8V/5.5V

28.9 Parallel Programming Characteristics

Table 28-8. Parallel Programming Characteristics, $V_{CC} = 5V \pm 10\%$

Symbol	Parameter	Min	Typ	Max	Units
V_{PP}	Programming Enable Voltage	11.5		12.5	V
I_{PP}	Programming Enable Current			250	μA
t_{DVXH}	Data and Control Valid before XTAL1 High	67			ns
t_{XLXH}	XTAL1 Low to XTAL1 High	200			ns
t_{XHXL}	XTAL1 Pulse Width High	150			ns
t_{XLDX}	Data and Control Hold after XTAL1 Low	67			ns
t_{XLWL}	XTAL1 Low to \overline{WR} Low	0			ns
t_{XLPH}	XTAL1 Low to PAGES high	0			ns
t_{PLXH}	PAGES low to XTAL1 high	150			ns
t_{BVPH}	BS1 Valid before PAGES High	67			ns
t_{PHPL}	PAGES Pulse Width High	150			ns
t_{PLBX}	BS1 Hold after PAGES Low	67			ns
t_{WLBX}	BS2/1 Hold after \overline{WR} Low	67			ns
t_{PLWL}	PAGES Low to \overline{WR} Low	67			ns
t_{BVWL}	BS1 Valid to \overline{WR} Low	67			ns
t_{WLWH}	\overline{WR} Pulse Width Low	150			ns
t_{WLRL}	\overline{WR} Low to RDY/ \overline{BSY} Low	0		1	μs
t_{WLRH}	\overline{WR} Low to RDY/ \overline{BSY} High ⁽¹⁾	3.7		4.5	ms
t_{WLRH_CE}	\overline{WR} Low to RDY/ \overline{BSY} High for Chip Erase ⁽²⁾	7.5		9	ms
t_{XLOL}	XTAL1 Low to \overline{OE} Low	0			ns
t_{BVDV}	BS1 Valid to DATA valid	0		250	ns
t_{OLDV}	\overline{OE} Low to DATA Valid			250	ns
t_{OHDZ}	\overline{OE} High to DATA Tri-stated			250	ns

- Notes:
- t_{WLRH} is valid for the Write Flash, Write EEPROM, Write Fuse bits and Write Lock bits commands.
 - t_{WLRH_CE} is valid for the Chip Erase command.

Figure 28-6. Parallel Programming Timing, Including some General Timing Requirements

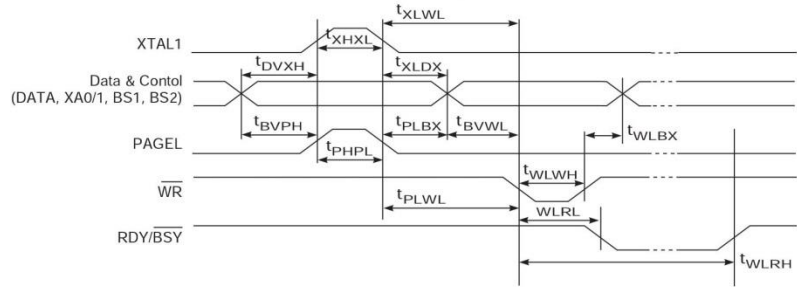
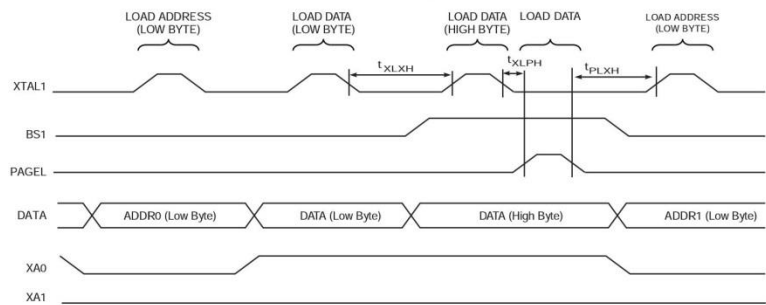
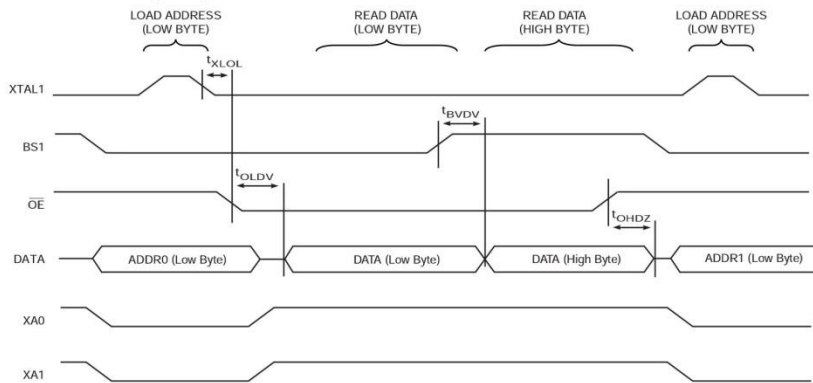


Figure 28-7. Parallel Programming Timing, Loading Sequence with Timing Requirements⁽¹⁾



Note: 1. The timing requirements shown in Figure 28-6 (i.e., t_{DVXH} , t_{XHL} , and t_{XLDX}) also apply to loading operation.

Figure 28-8. Parallel Programming Timing, Reading Sequence (within the Same Page) with Timing Requirements⁽¹⁾



Note: 1. The timing requirements shown in Figure 28-6 (i.e., t_{DVXH} , t_{XHL} , and t_{XLDX}) also apply to reading operation.

29. Typical Characteristics

The following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. A square wave generator with rail-to-rail output is used as clock source.

All Active- and Idle current consumption measurements are done with all bits in the PRR register set and thus, the corresponding I/O modules are turned off. Also the Analog Comparator is disabled during these measurements. The ["ATmega88PA: Supply Current of IO Modules"](#) on page 356 and [page 380](#) shows the additional current consumption compared to I_{CC} Active and I_{CC} Idle for every I/O module controlled by the Power Reduction Register. See ["Power Reduction Register"](#) on page 42 for details.

The power consumption in Power-down mode is independent of clock selection.

The current consumption is a function of several factors such as: operating voltage, operating frequency, loading of I/O pins, switching rate of I/O pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency.

The current drawn from capacitive loaded pins may be estimated (for one pin) as $C_L \cdot V_{CC} \cdot f$ where C_L = load capacitance, V_{CC} = operating voltage and f = average switching frequency of I/O pin.

The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies higher than the ordering code indicates.

The difference between current consumption in Power-down mode with Watchdog Timer enabled and Power-down mode with Watchdog Timer disabled represents the differential current drawn by the Watchdog Timer.

29.1 ATmega48PA Typical Characteristics

29.1.1 Active Supply Current

Figure 29-1. ATmega48PA: Active Supply Current vs. Low Frequency (0.1-1.0 MHz)

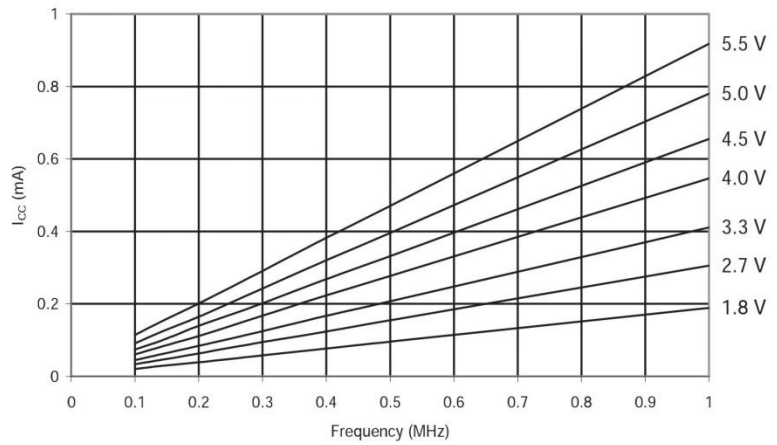


Figure 29-2. ATmega48PA: Active Supply Current vs. Frequency (1-20 MHz)

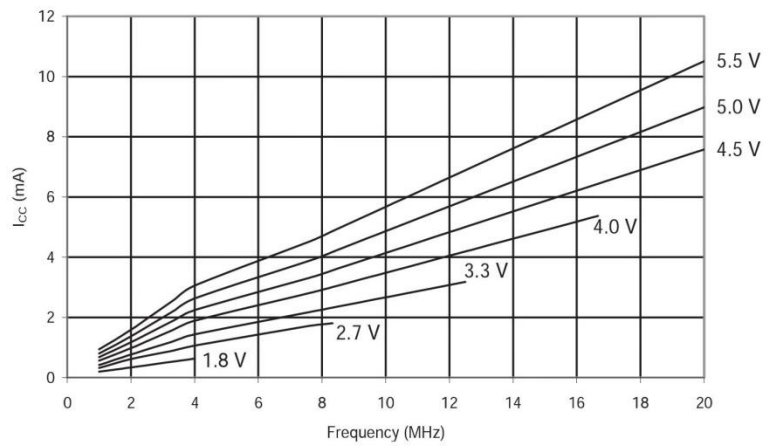


Figure 29-3. ATmega48PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

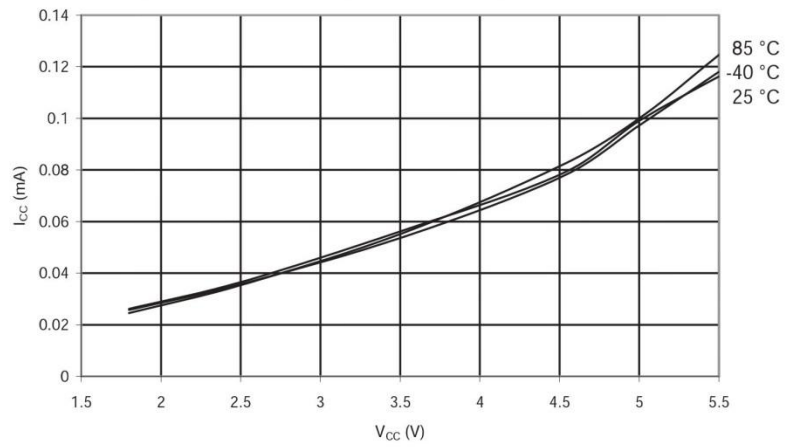


Figure 29-4. ATmega48PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

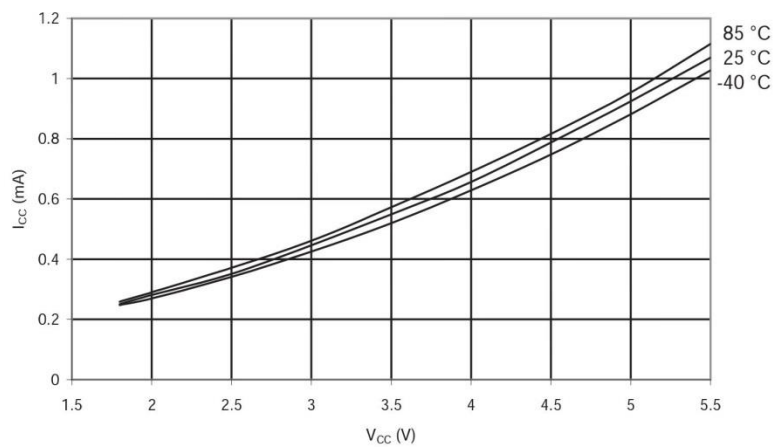
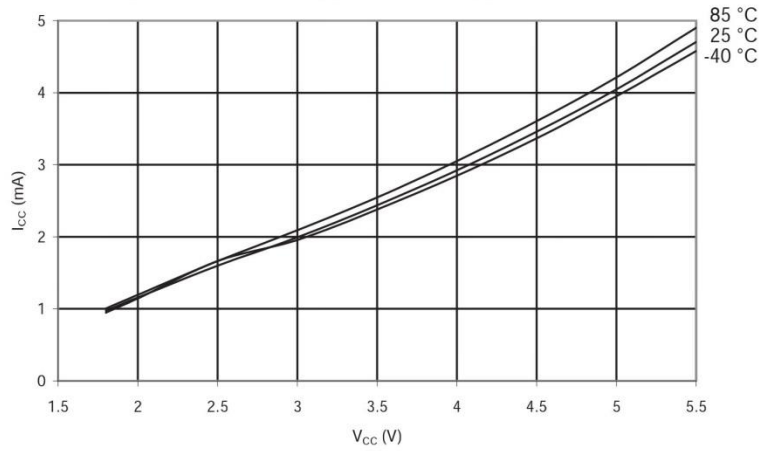


Figure 29-5. ATmega48PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.1.2 Idle Supply Current

Figure 29-6. ATmega48PA: Idle Supply Current vs. Low Frequency (0.1-1.0 MHz)

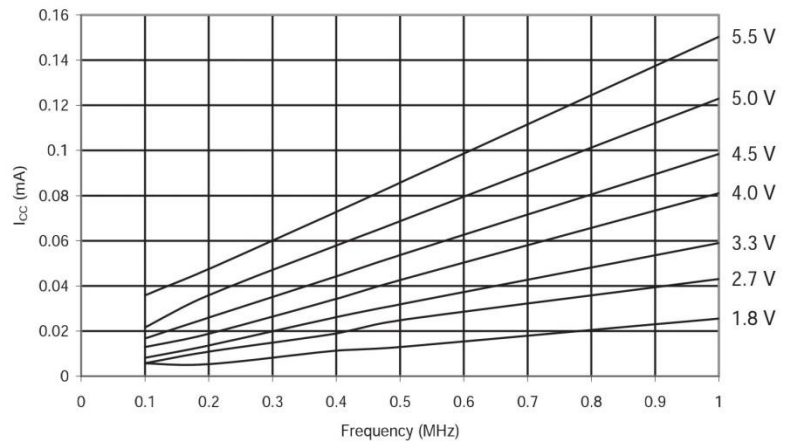


Figure 29-7. ATmega48PA: Idle Supply Current vs. Frequency (1-20 MHz)

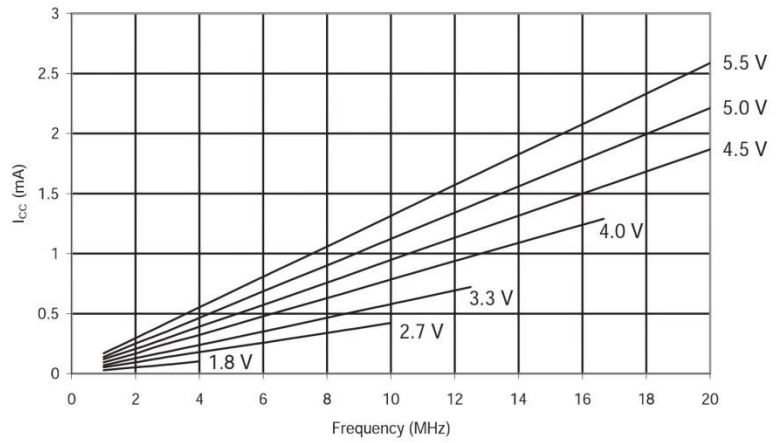


Figure 29-8. ATmega48PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

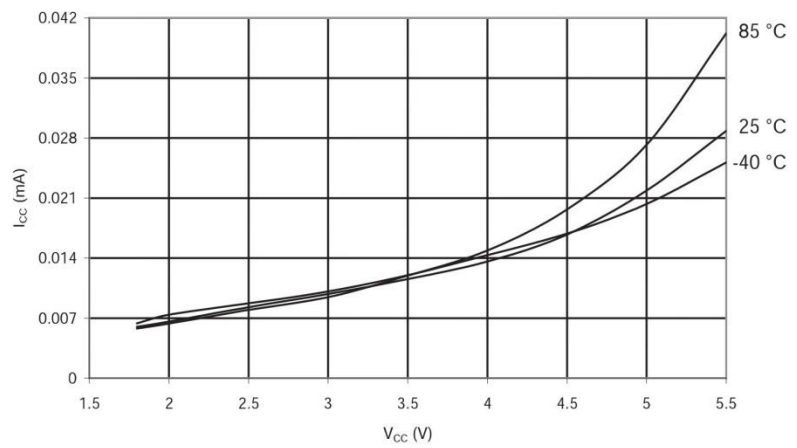


Figure 29-9. ATmega48PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

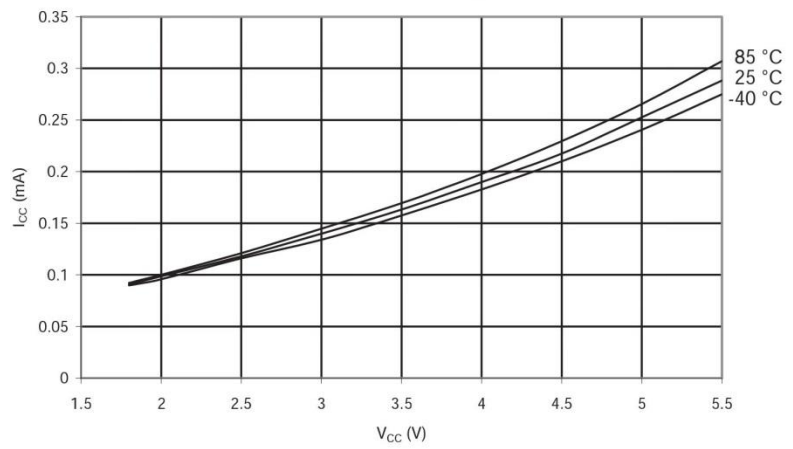
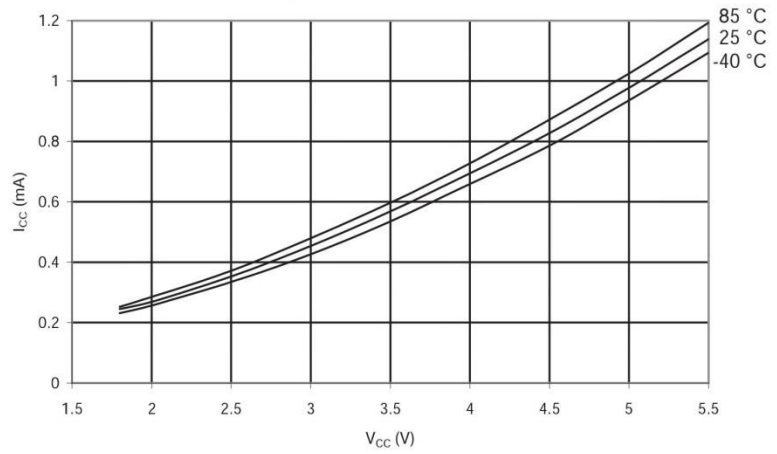


Figure 29-10. ATmega48PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.1.3 ATmega48PA: Supply Current of IO Modules

The tables and formulas below can be used to calculate the additional current consumption for the different I/O modules in Active and Idle mode. The enabling or disabling of the I/O modules are controlled by the Power Reduction Register. See "Power Reduction Register" on page 42 for details.

Table 29-1. ATmega48PA: Additional Current Consumption for the different I/O modules (absolute values)

PRR bit	Typical numbers		
	V _{CC} = 2V, F = 1 MHz	V _{CC} = 3V, F = 4 MHz	V _{CC} = 5V, F = 8 MHz
PRUSART0	2.9 uA	20.7 uA	97.4 uA
PRTWI	6.0 uA	44.8 uA	219.7 uA
PRTIM2	5.0 uA	34.5 uA	141.3 uA
PRTIM1	3.6 uA	24.4 uA	107.7 uA
PRTIM0	1.4 uA	9.5 uA	38.4 uA
PRSPI	5.0 uA	38.0 uA	190.4 uA
PRADC	6.1 uA	47.4 uA	244.7 uA

Table 29-2. ATmega48PA: Additional Current Consumption (percentage) in Active and Idle mode

PRR bit	Additional Current consumption compared to Active with external clock (see Figure 29-1 on page 327 and Figure 29-2 on page 327)	Additional Current consumption compared to Idle with external clock (see Figure 29-6 on page 329 and Figure 29-7 on page 330)
PRUSART0	1.8%	11.4%
PRTWI	3.9%	20.6%
PRTIM2	2.9%	15.7%
PRTIM1	2.1%	11.2%
PRTIM0	0.8%	4.2%
PRSPI	3.3%	17.6%
PRADC	4.2%	22.1%

It is possible to calculate the typical current consumption based on the numbers from Table 29-2 on page 332 for other V_{CC} and frequency settings than listed in Table 29-1 on page 332.

Example

Calculate the expected current consumption in idle mode with TIMER1, ADC, and SPI enabled at V_{CC} = 2.0V and F = 1MHz. From Table 29-2 on page 332, third column, we see that we need to add 11.2% for the TIMER1, 22.1% for the ADC, and 17.6% for the SPI module. Reading from Figure 29-6 on page 329, we find that the idle current consumption is ~0.028 mA at V_{CC} = 2.0V and F = 1MHz. The total current consumption in idle mode with TIMER1, ADC, and SPI enabled, gives:

$$I_{CC\text{total}} \approx 0.028 \text{ mA} \cdot (1 + 0.112 + 0.221 + 0.176) \approx 0.042 \text{ mA}$$

29.1.4 Power-down Supply Current

Figure 29-11. ATmega48PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Disabled)

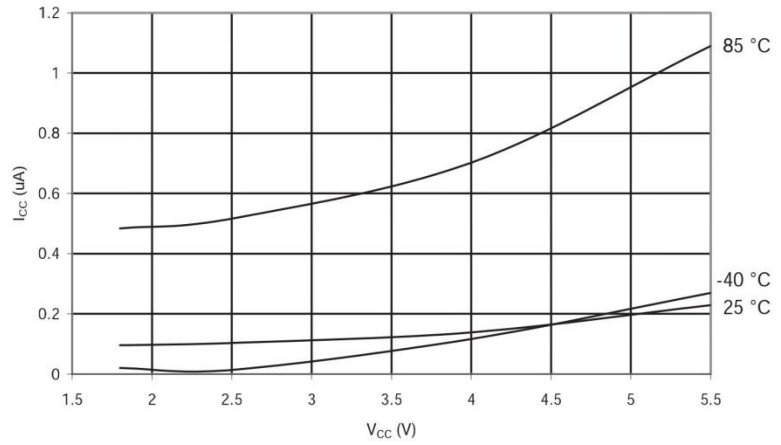
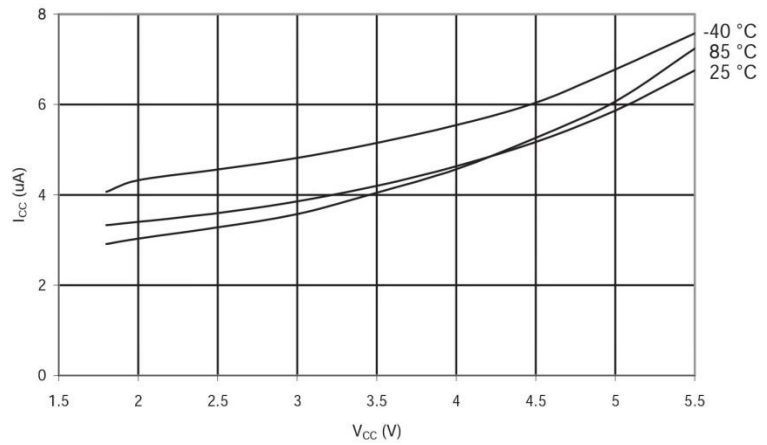
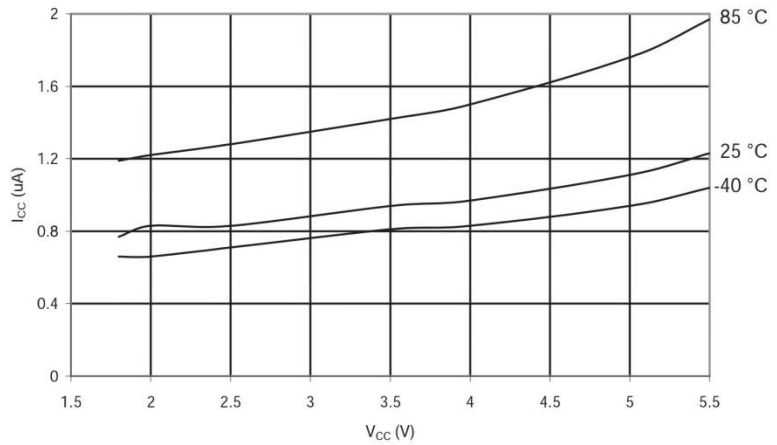


Figure 29-12. ATmega48PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



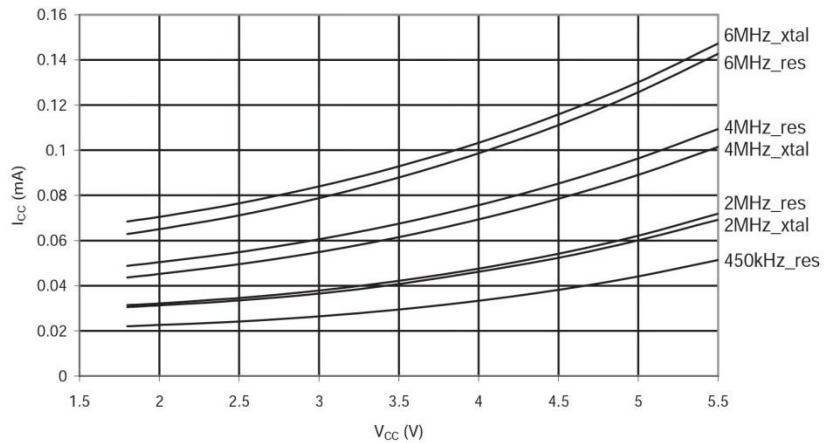
29.1.5 Power-save Supply Current

Figure 29-13. ATmega48PA: Power-Save Supply Current vs. V_{CC} (Watchdog Timer Disabled and 32 kHz Crystal Oscillator Running)



29.1.6 Standby Supply Current

Figure 29-14. ATmega48PA: Standby Supply Current vs. V_{CC} (Watchdog Timer Disabled)



29.1.7 Pin Pull-Up

Figure 29-15. ATmega48PA: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 1.8\text{ V}$)

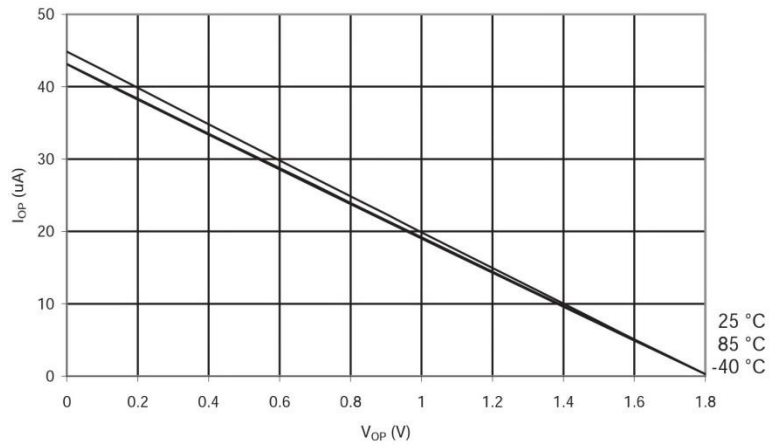


Figure 29-16. ATmega48PA: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 2.7\text{ V}$)

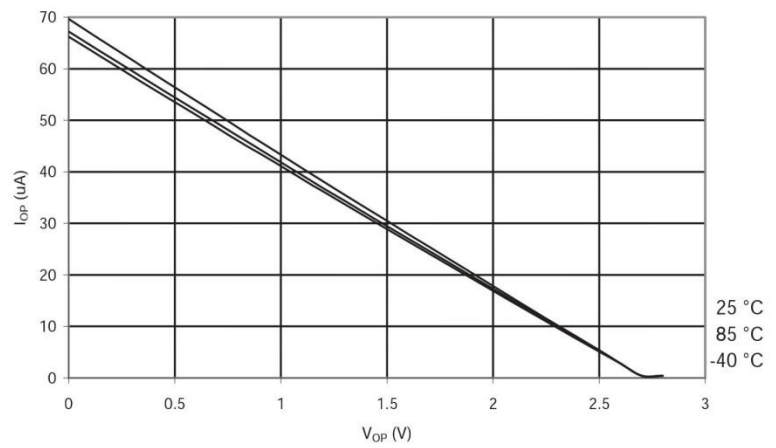


Figure 29-17. ATmega48PA: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 5\text{ V}$)

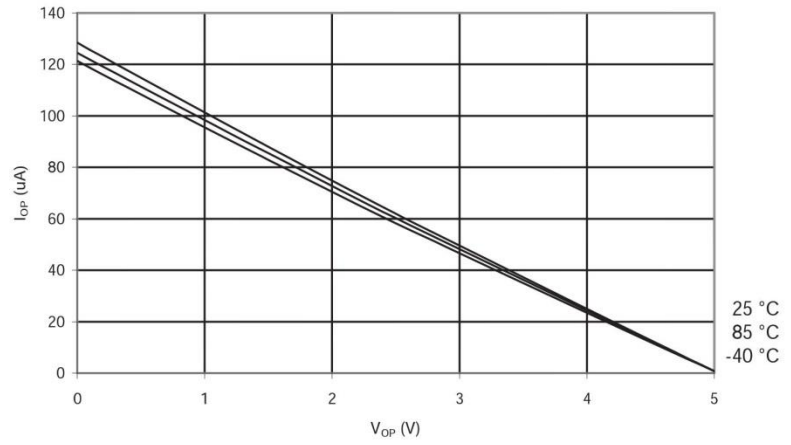


Figure 29-18. ATmega48PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 1.8\text{ V}$)

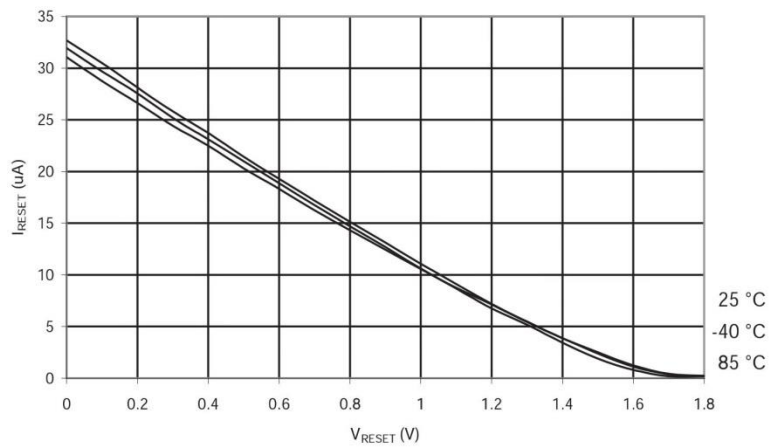


Figure 29-19. ATmega48PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 2.7\text{ V}$)

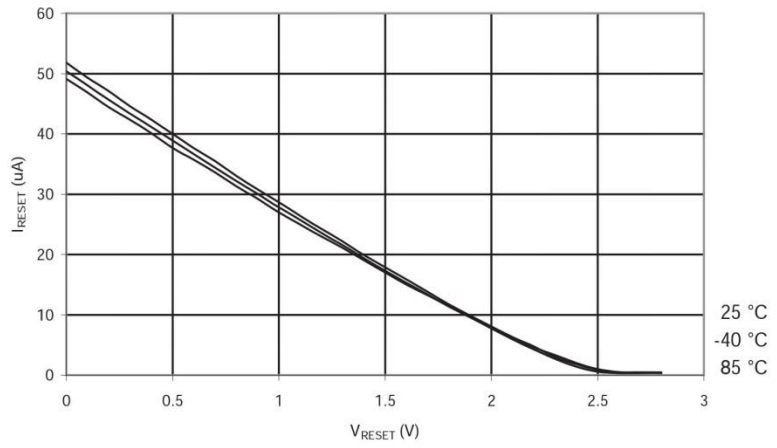
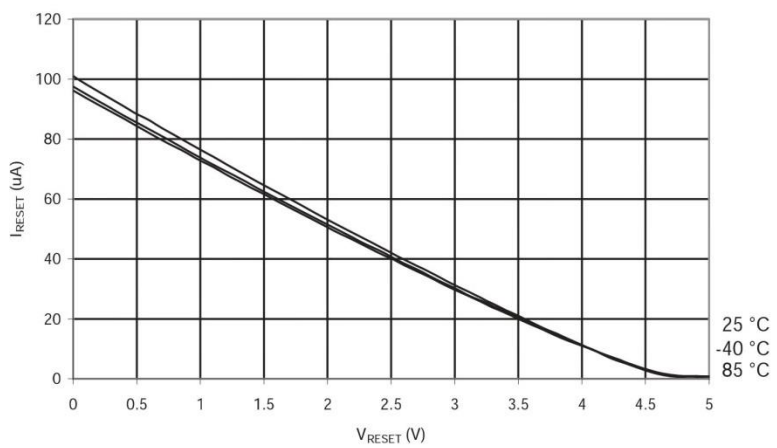


Figure 29-20. ATmega48PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 5\text{ V}$)



29.1.8 Pin Driver Strength

Figure 29-21. ATmega48PA: I/O Pin Output Voltage vs. Sink Current ($V_{CC} = 3\text{ V}$)

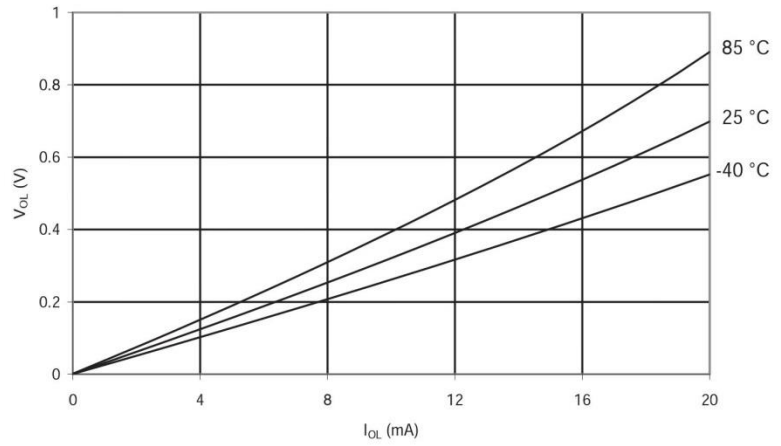


Figure 29-22. ATmega48PA: I/O Pin Output Voltage vs. Sink Current ($V_{CC} = 5\text{ V}$)

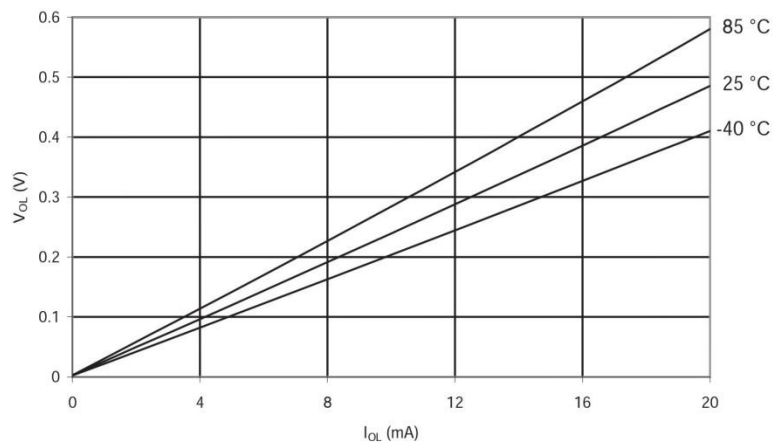


Figure 29-23. ATmega48PA: I/O Pin Output Voltage vs. Source Current ($V_{CC} = 3\text{ V}$)

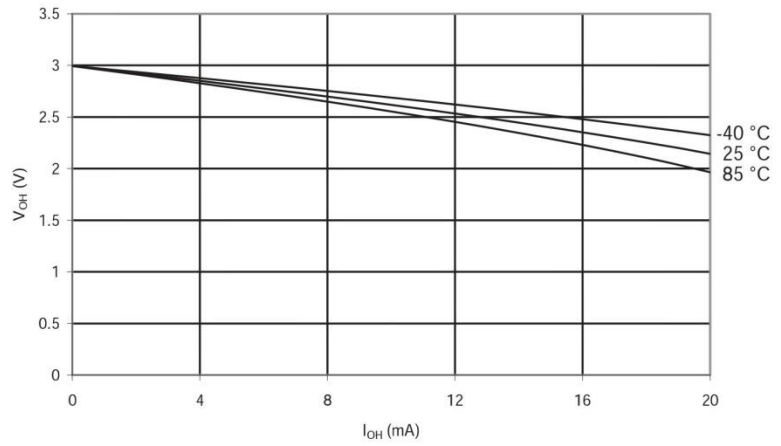
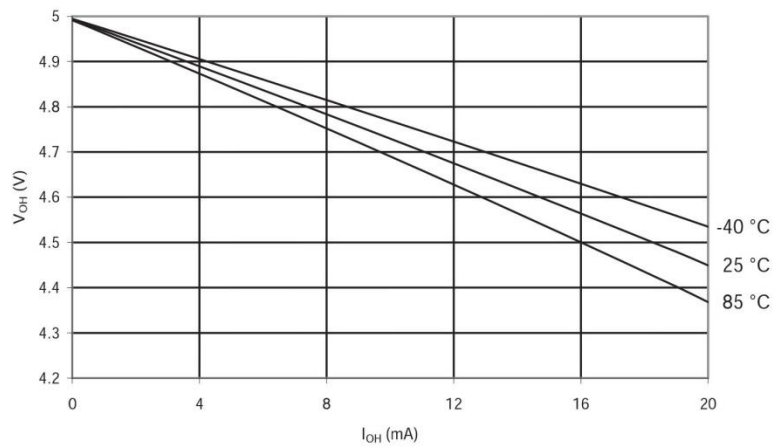


Figure 29-24. ATmega48PA: I/O Pin Output Voltage vs. Source Current ($V_{CC} = 5\text{ V}$)



29.1.9 Pin Threshold and Hysteresis

Figure 29-25. ATmega48PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH} , I/O Pin read as '1')

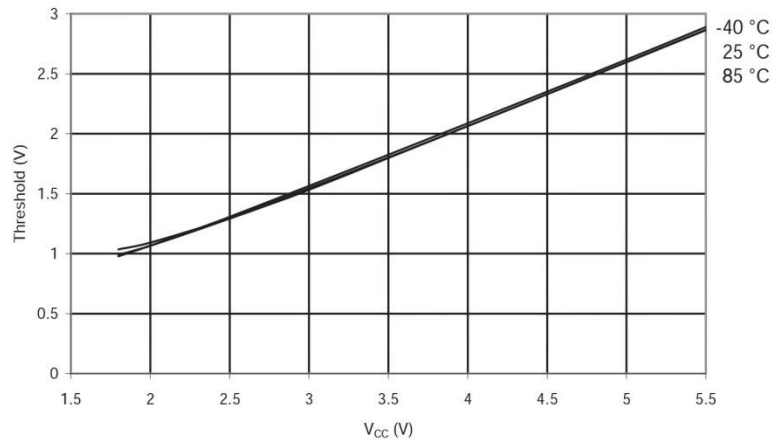


Figure 29-26. ATmega48PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL} , I/O Pin read as '0')

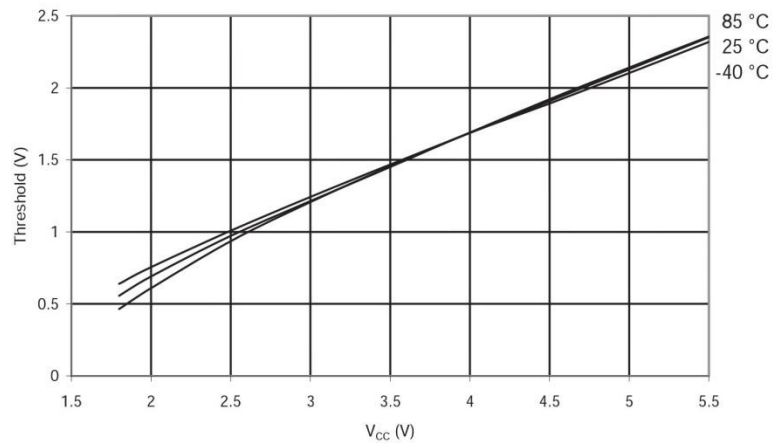


Figure 29-27. ATmega48PA: I/O Pin Input Hysteresis vs. V_{CC}

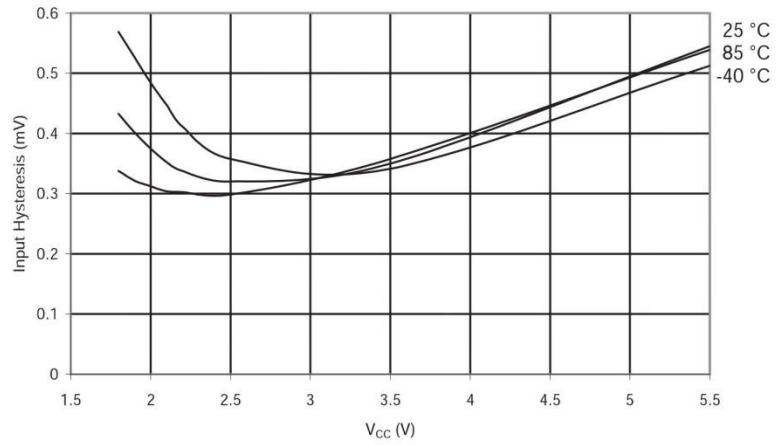


Figure 29-28. ATmega48PA: Reset Input Threshold Voltage vs. V_{CC} (V_{Ihr} I/O Pin read as '1')

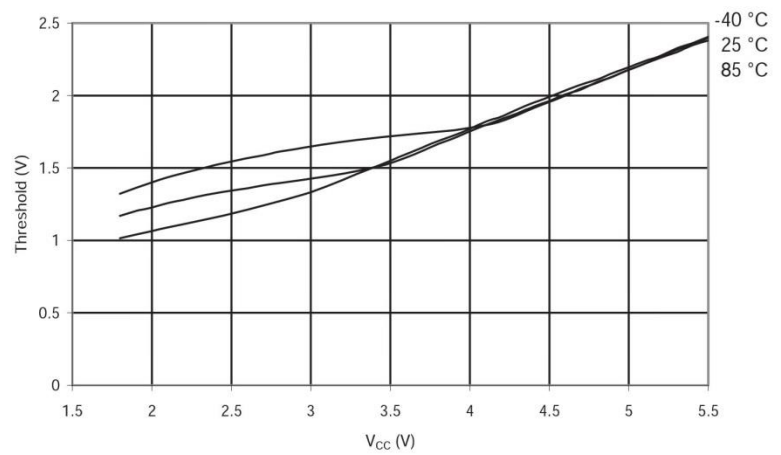


Figure 29-29. ATmega48PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IL} , I/O Pin read as '0')

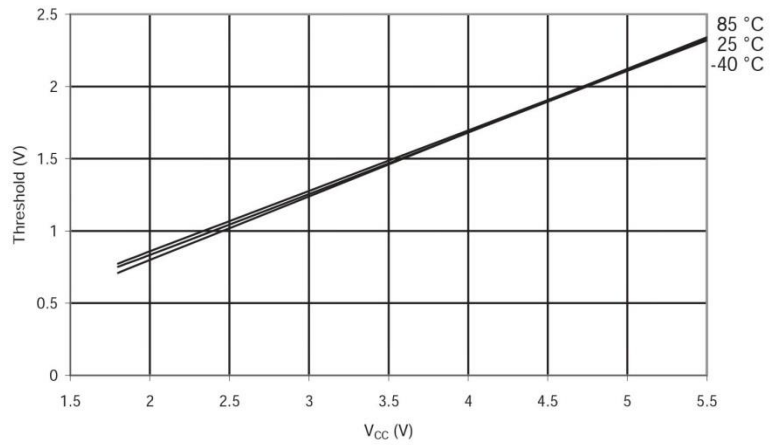
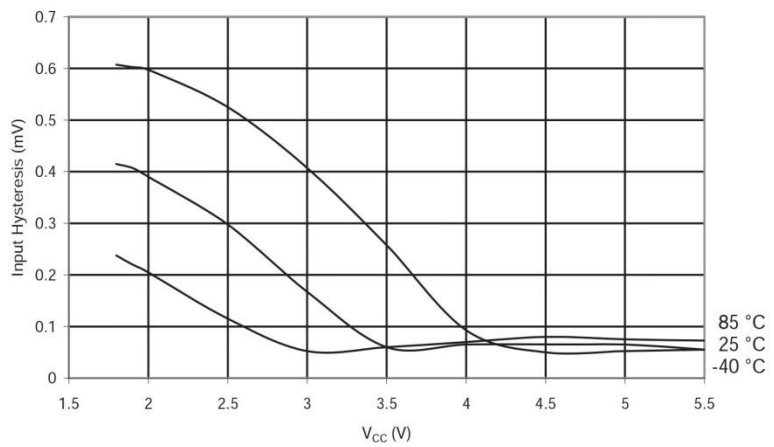


Figure 29-30. ATmega48PA: Reset Pin Input Hysteresis vs. V_{CC}



29.1.10 BOD Threshold

Figure 29-31. ATmega48PA: BOD Thresholds vs. Temperature (BODLEVEL is 1.8 V)

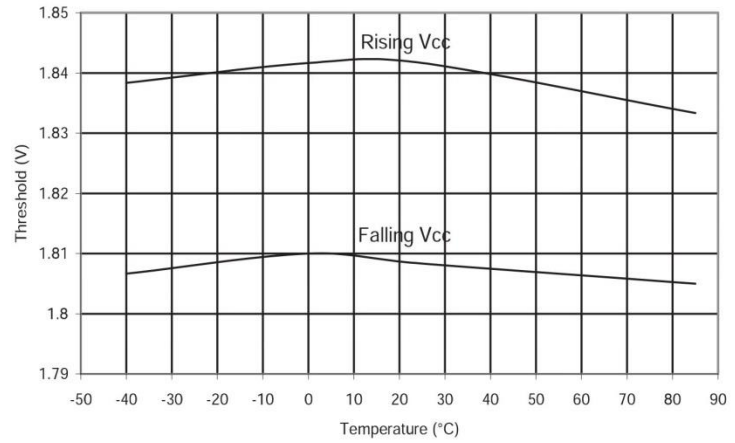


Figure 29-32. ATmega48PA: BOD Thresholds vs. Temperature (BODLEVEL is 2.7 V)

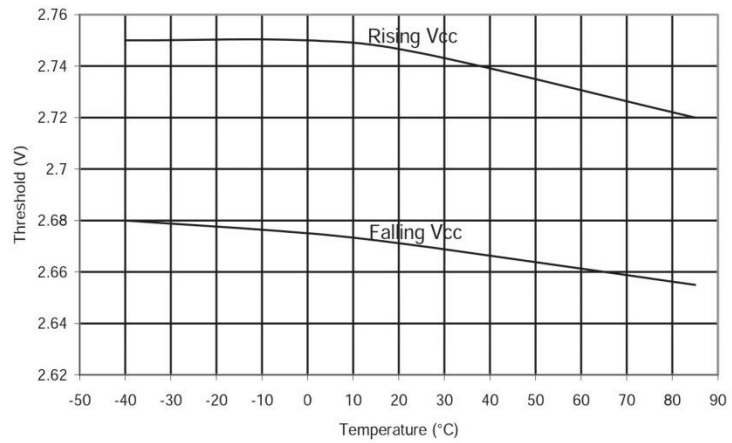
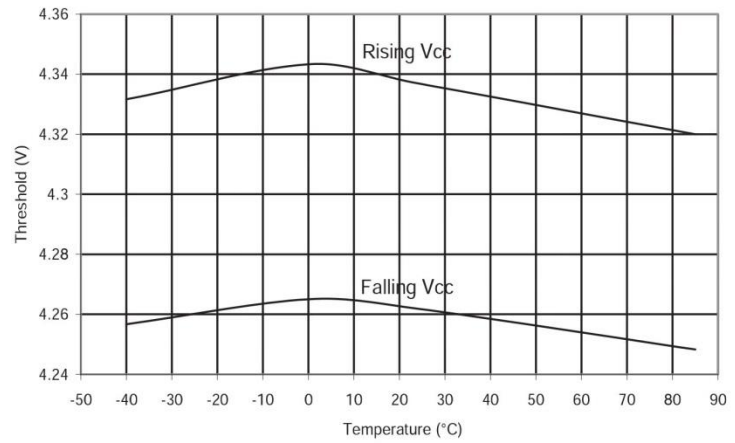


Figure 29-33. ATmega48PA: BOD Thresholds vs. Temperature (BODLEVEL is 4.3 V)



29.1.11 Internal Oscillator Speed

Figure 29-34. ATmega48PA: Watchdog Oscillator Frequency vs. Temperature

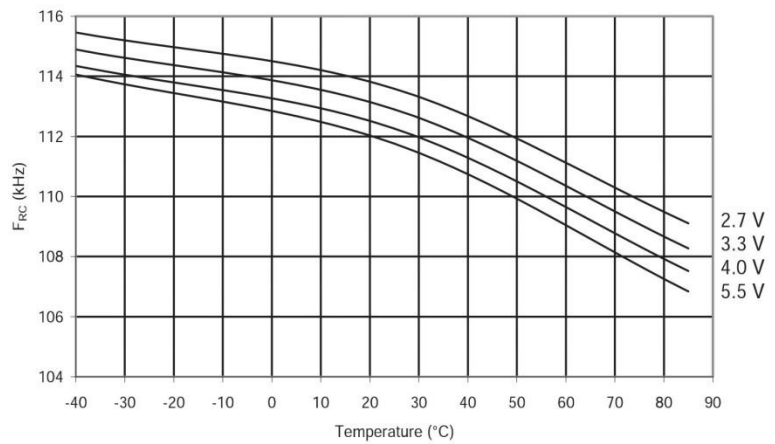


Figure 29-35. ATmega48PA: Watchdog Oscillator Frequency vs. V_{CC}

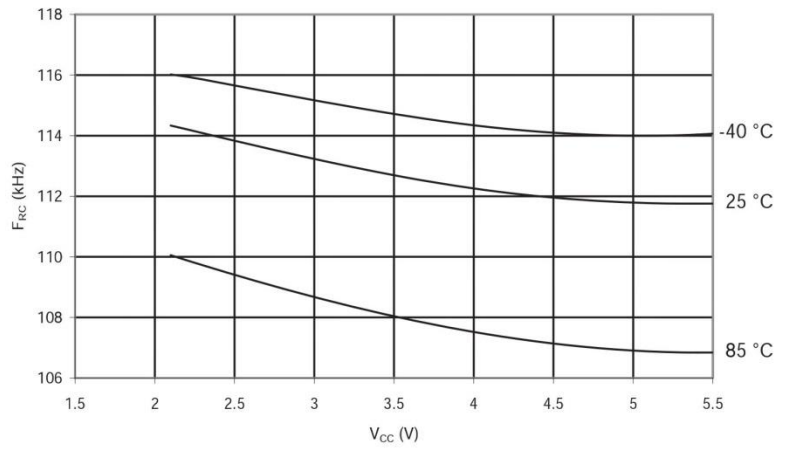


Figure 29-36. ATmega48PA: Calibrated 8 MHz RC Oscillator Frequency vs. V_{CC}

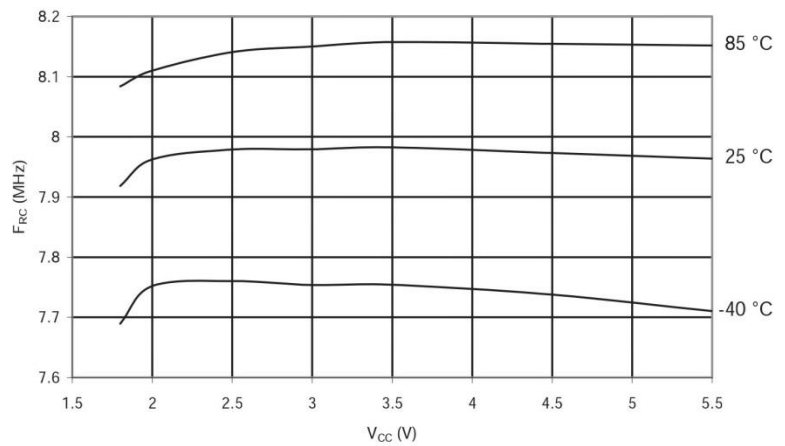


Figure 29-37. ATmega48PA: Calibrated 8 MHz RC Oscillator Frequency vs. Temperature

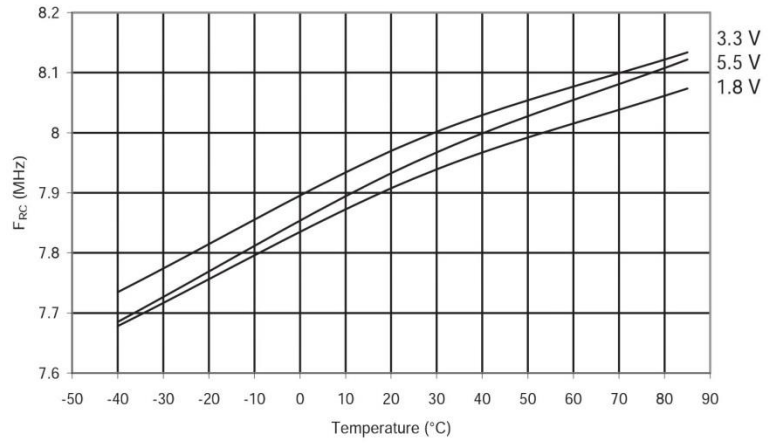
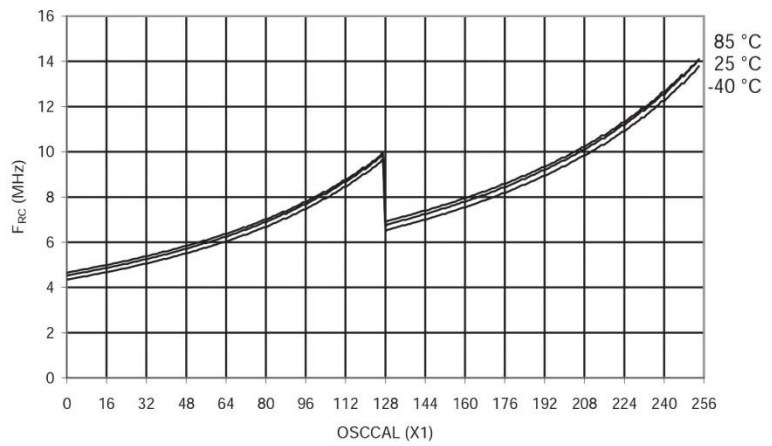


Figure 29-38. ATmega48PA: Calibrated 8 MHz RC Oscillator Frequency vs. OSCCAL Value



29.1.12 Current Consumption of Peripheral Units

Figure 29-39. ATmega48PA: ADC Current vs. V_{CC} (AREF = AV_{CC})

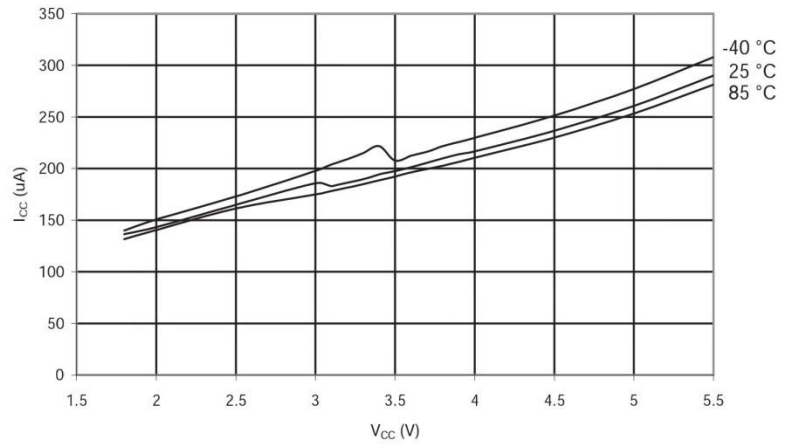


Figure 29-40. ATmega48PA: Analog Comparator Current vs. V_{CC}

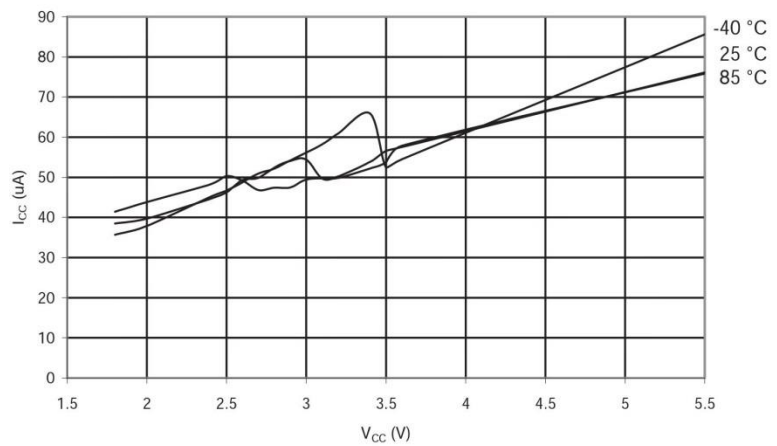


Figure 29-41. ATmega48PA: AREF External Reference Current vs. V_{CC}

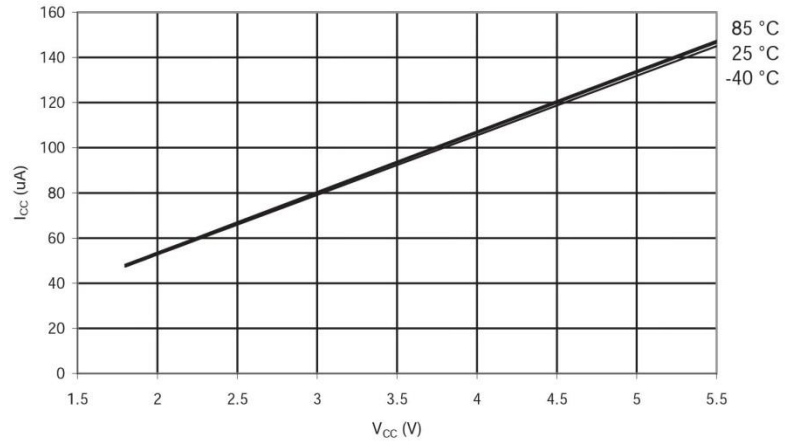


Figure 29-42. ATmega48PA: Brownout Detector Current vs. V_{CC}

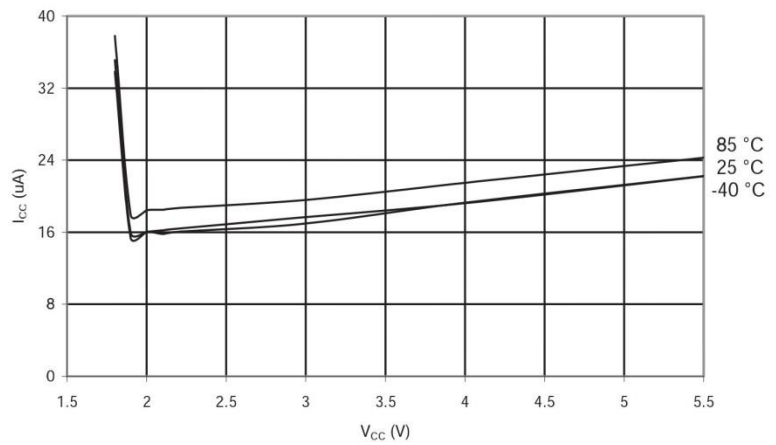
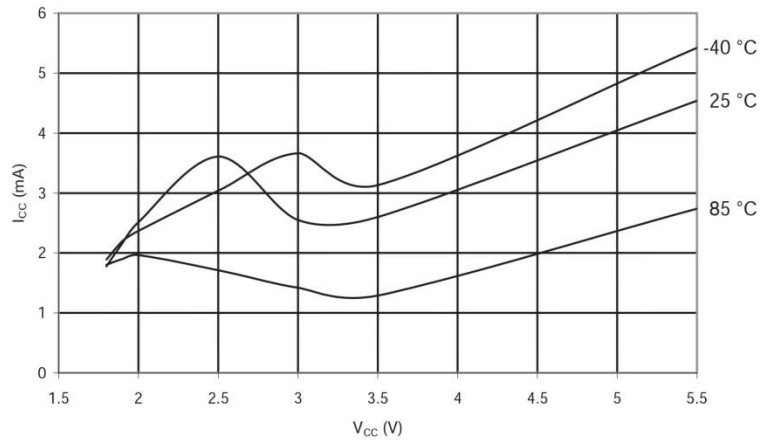


Figure 29-43. ATmega48PA: Programming Current vs. V_{CC}



29.1.13 Current Consumption in Reset and Reset Pulsewidth

Figure 29-44. ATmega48PA: Reset Supply Current vs. Low Frequency (0.1 - 1.0 MHz)

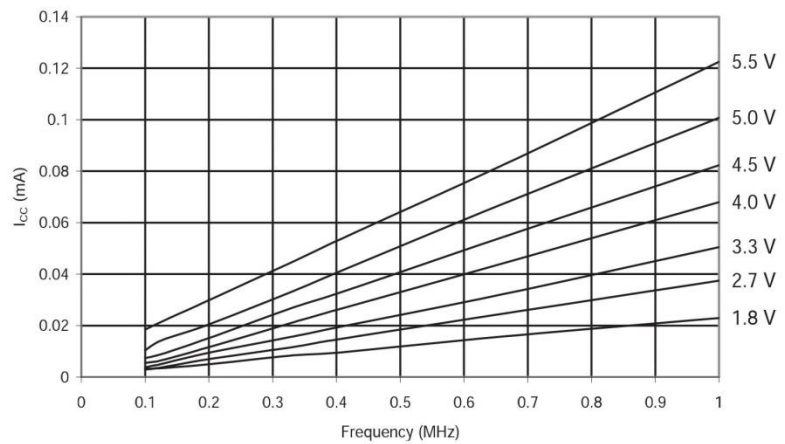


Figure 29-45. ATmega48PA: Reset Supply Current vs. Frequency (1 - 20 MHz)

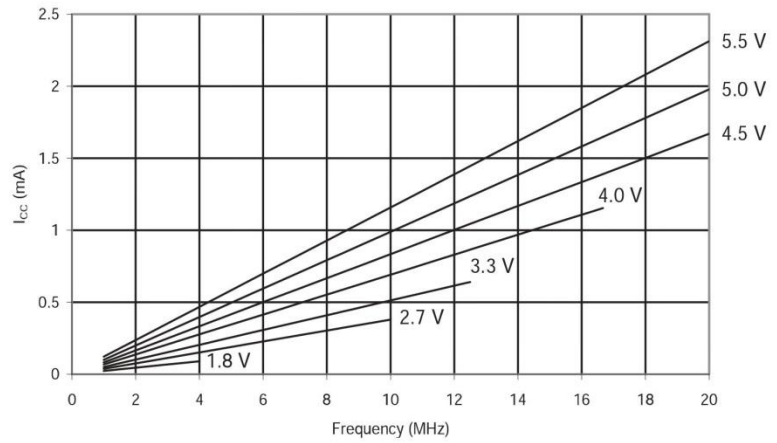
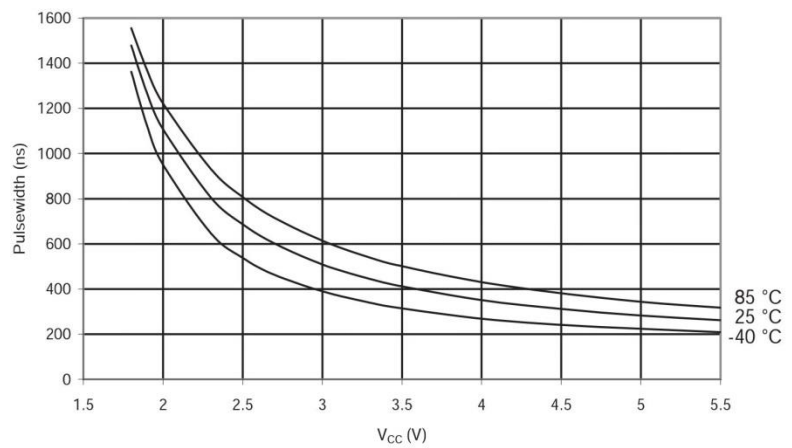


Figure 29-46. ATmega48PA: Minimum Reset Pulse width vs. V_{CC}



29.2 ATmega88PA Typical Characteristics

29.2.1 Active Supply Current

Figure 29-47. ATmega88PA: Active Supply Current vs. Low Frequency (0.1-1.0 MHz)

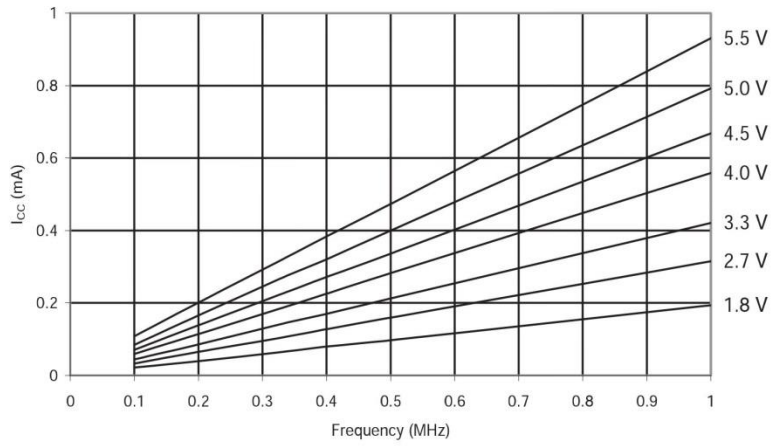


Figure 29-48. ATmega88PA: Active Supply Current vs. Frequency (1 - 20 MHz)

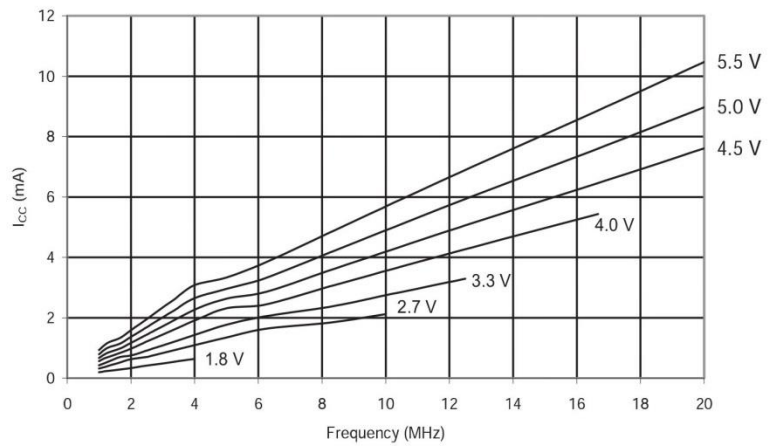


Figure 29-49. ATmega88PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

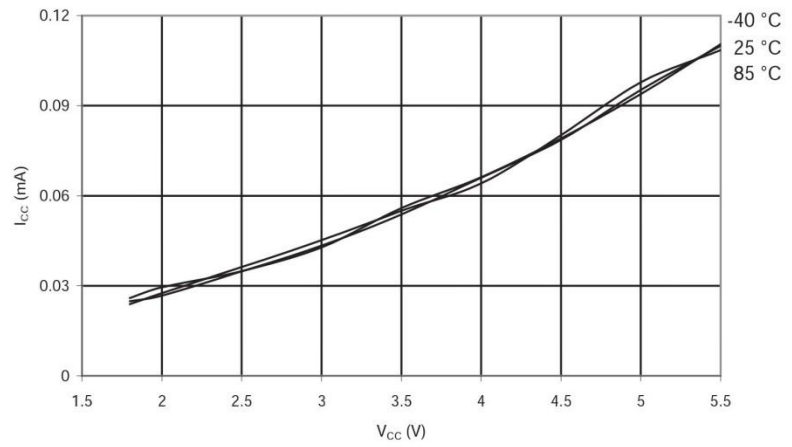


Figure 29-50. ATmega88PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

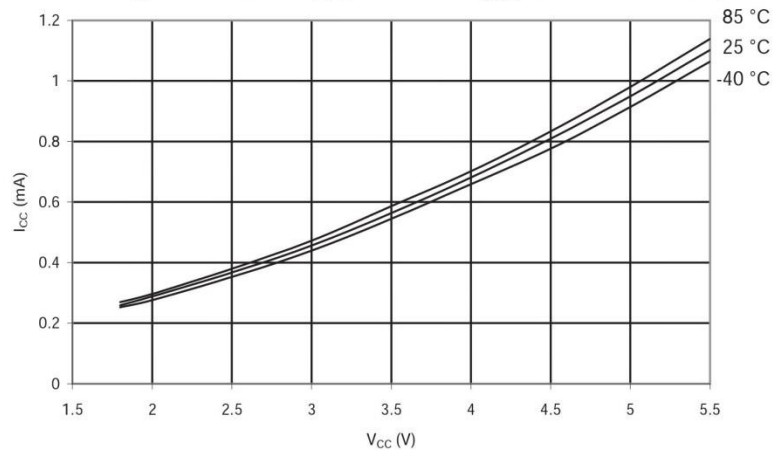
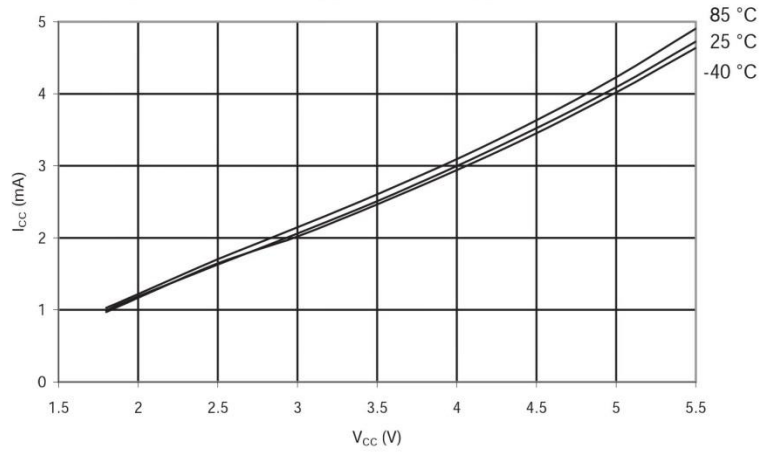


Figure 29-51. ATmega88PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.2.2 Idle Supply Current

Figure 29-52. ATmega88PA: Idle Supply Current vs. Low Frequency (0.1-1.0 MHz)

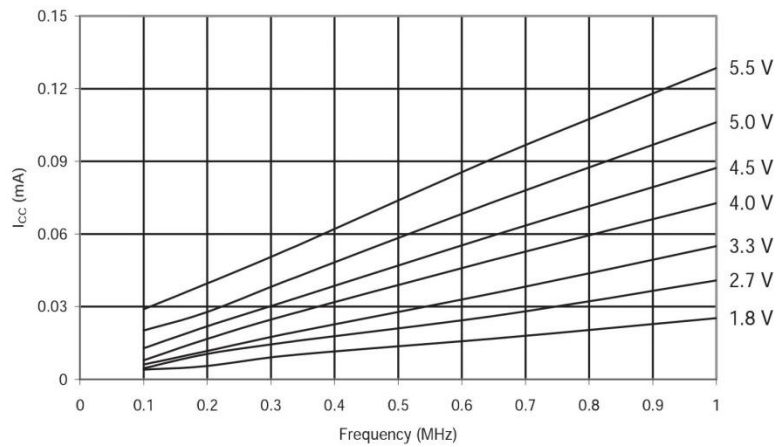


Figure 29-53. ATmega88PA: Idle Supply Current vs. Frequency (1 - 20 MHz)

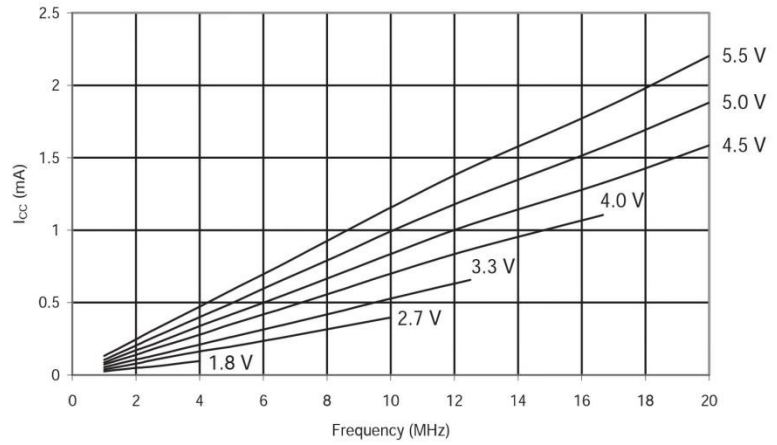


Figure 29-54. ATmega88PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

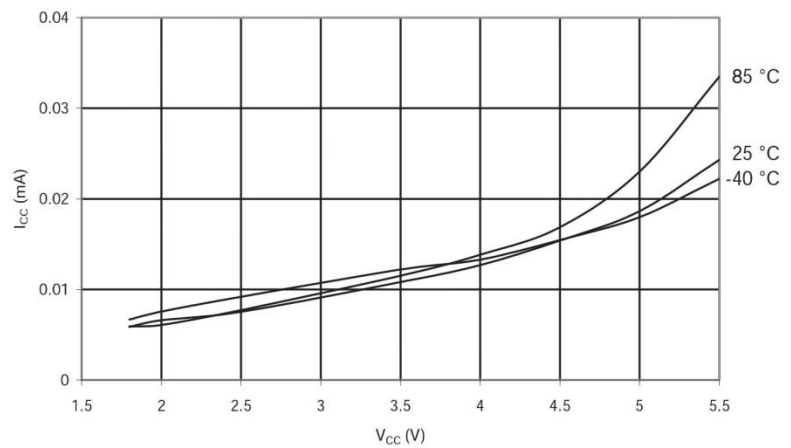


Figure 29-55. ATmega88PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

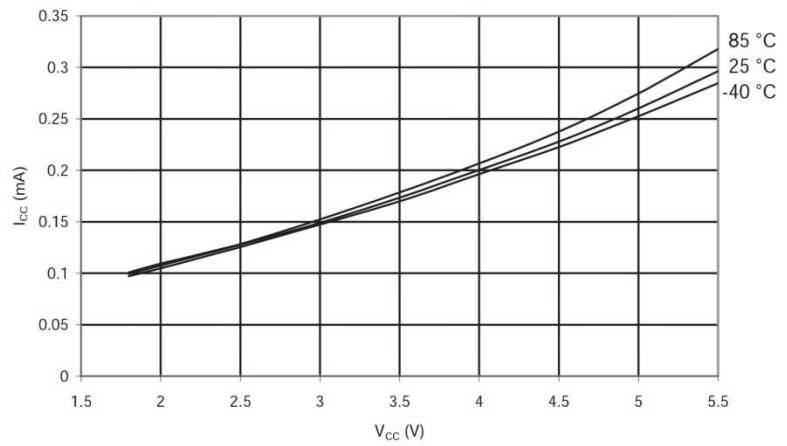
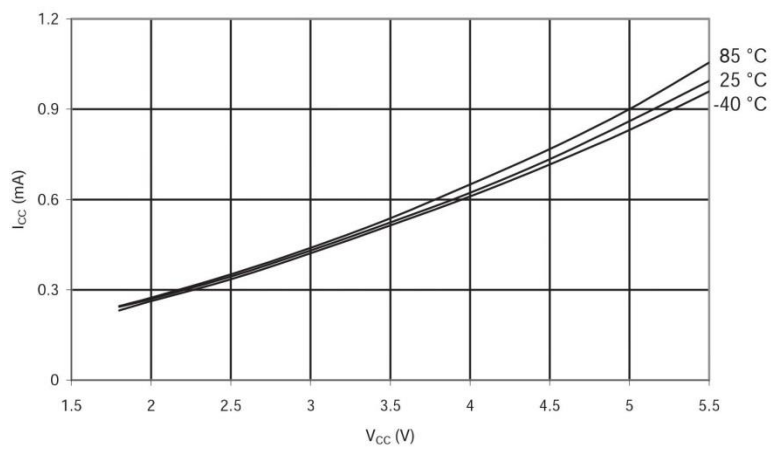


Figure 29-56. ATmega88PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.2.3 ATmega88PA: Supply Current of IO Modules

The tables and formulas below can be used to calculate the additional current consumption for the different I/O modules in Active and Idle mode. The enabling or disabling of the I/O modules are controlled by the Power Reduction Register. See "Power Reduction Register" on page 42 for details.

Table 29-3. ATmega88PA: Additional Current Consumption for the different I/O modules (absolute values)

PRR bit	Typical numbers		
	V _{CC} = 2V, F = 1 MHz	V _{CC} = 3V, F = 4 MHz	V _{CC} = 5V, F = 8 MHz
PRUSART0	3.0 uA	21.3 uA	97.9 uA
PRTWI	6.1 uA	45.4 uA	219.0 uA
PRTIM2	5.2 uA	35.2 uA	149.5 uA
PRTIM1	3.8 uA	25.6 uA	110.0 uA
PRTIM0	1.5 uA	9.8 uA	39.6 uA
PRSPI	5.2 uA	40.0 uA	199.6 uA
PRADC	6.3 uA	48.7 uA	247.0 uA

Table 29-4. ATmega88PA: Additional Current Consumption (percentage) in Active and Idle mode

PRR bit	Additional Current consumption compared to Active with external clock (see Figure 29-47 on page 351 and Figure 29-48 on page 351)	Additional Current consumption compared to Idle with external clock (see Figure 29-52 on page 353 and Figure 29-53 on page 354)
PRUSART0	1.8%	11.4%
PRTWI	3.9%	24.4%
PRTIM2	2.9%	18.6%
PRTIM1	2.1%	13.6%
PRTIM0	0.8%	5.2%
PRSPI	3.5%	21.5%
PRADC	4.2%	26.3%

It is possible to calculate the typical current consumption based on the numbers from Table 29-4 on page 356 for other V_{CC} and frequency settings than listed in Table 29-3 on page 356.

Example

Calculate the expected current consumption in idle mode with TIMER1, ADC, and SPI enabled at V_{CC} = 2.0V and F = 1MHz. From Table 29-4 on page 356, third column, we see that we need to add 13.6% for the TIMER1, 26.3% for the ADC, and 21.5% for the SPI module. Reading from Figure 29-52 on page 353, we find that the idle current consumption is ~0.027 mA at V_{CC} = 2.0V and F = 1MHz. The total current consumption in idle mode with TIMER1, ADC, and SPI enabled, gives:

$$I_{CC\text{total}} \approx 0.027 \text{ mA} \cdot (1 + 0.136 + 0.263 + 0.215) \approx 0.043 \text{ mA}$$

29.2.4 Power-down Supply Current

Figure 29-57. ATmega88PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Disabled)

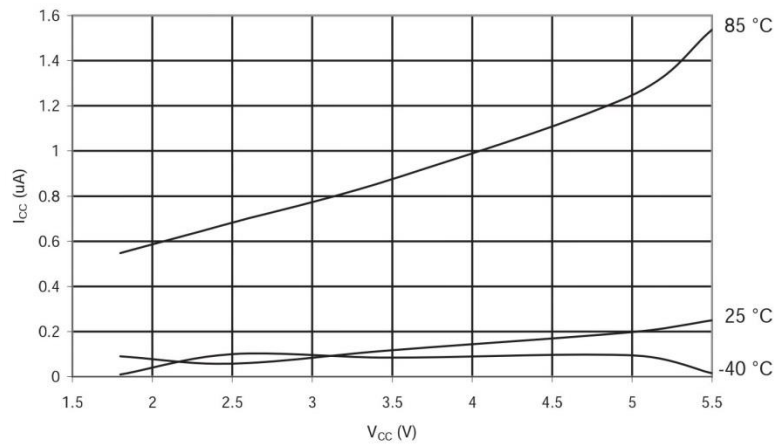
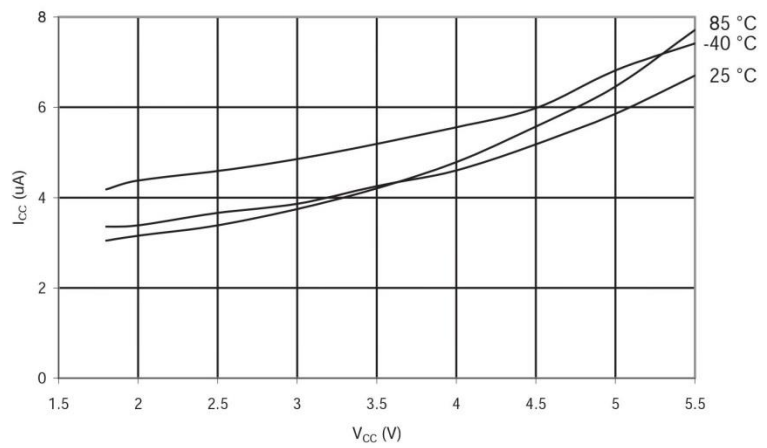
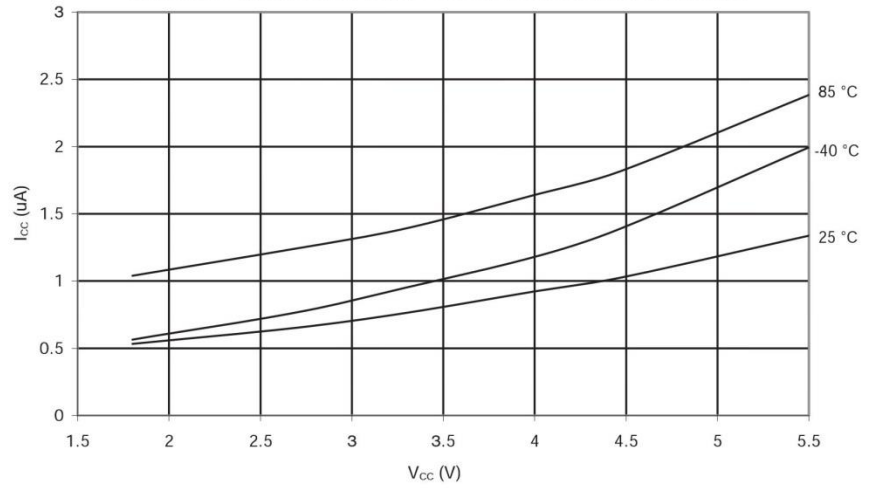


Figure 29-58. ATmega88PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



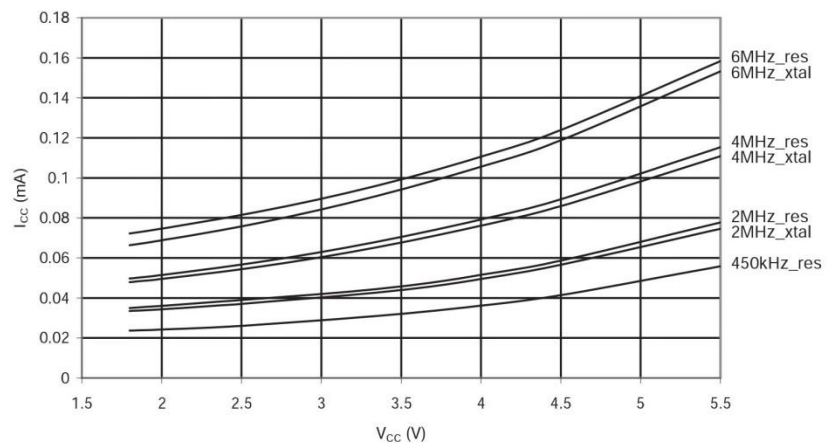
29.2.5 Power-save Supply Current

Figure 29-59. ATmega88PA: Power-Save Supply Current vs. V_{CC} (Watchdog Timer Disabled and 32 kHz Crystal Oscillator Running)



29.2.6 Standby Supply Current

Figure 29-60. ATmega88PA: Standby Supply Current vs. V_{CC} (Watchdog Timer Disabled)



29.2.7 Pin Pull-Up

Figure 29-61. ATmega88PA: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 1.8\text{ V}$)

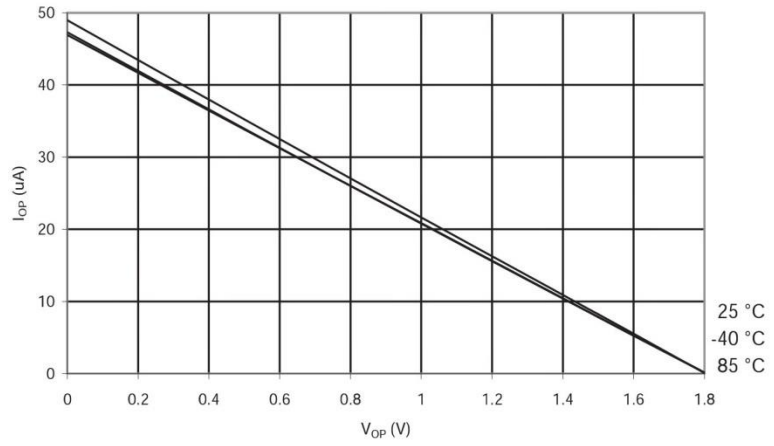


Figure 29-62. ATmega88PA: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 2.7\text{ V}$)

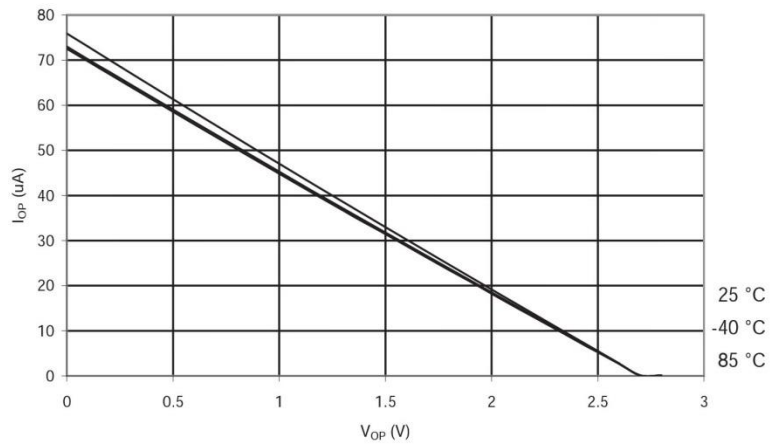


Figure 29-63. ATmega88PA: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 5\text{ V}$)

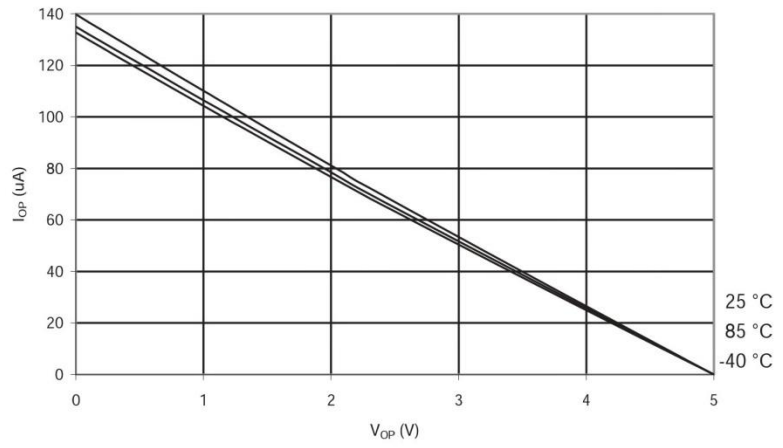


Figure 29-64. ATmega88PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 1.8\text{ V}$)

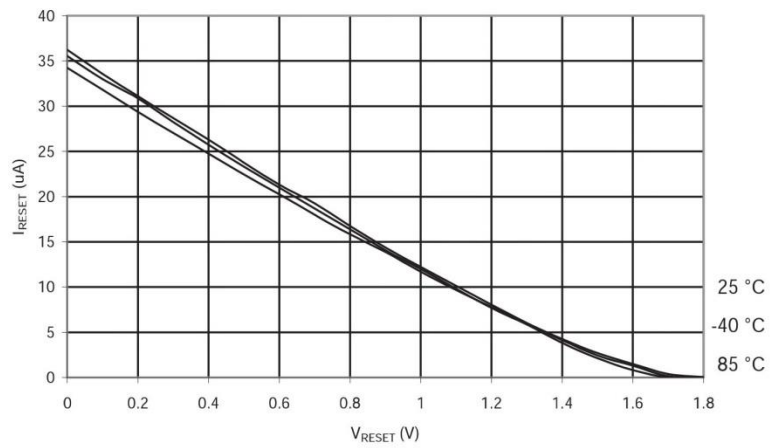


Figure 29-65. ATmega88PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 2.7\text{ V}$)

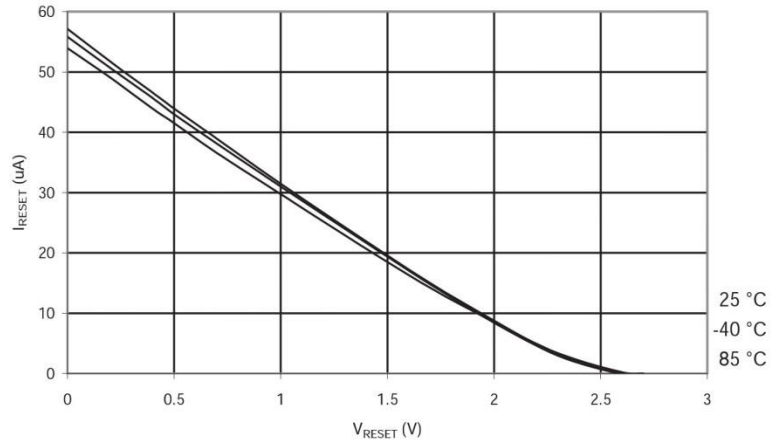
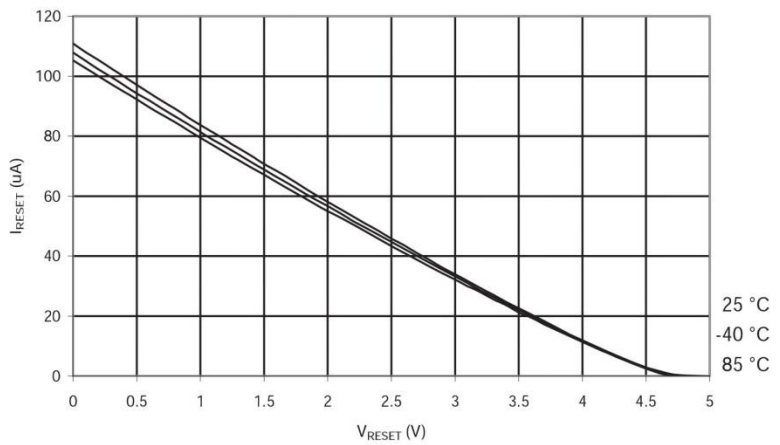


Figure 29-66. ATmega88PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 5\text{ V}$)



29.2.8 Pin Driver Strength

Figure 29-67. ATmega88PA: I/O Pin Output Voltage vs. Sink Current ($V_{CC} = 3\text{ V}$)

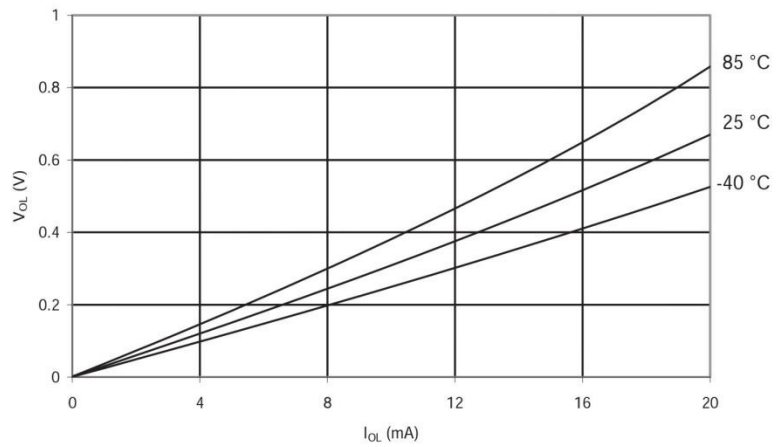


Figure 29-68. ATmega88PA: I/O Pin Output Voltage vs. Sink Current ($V_{CC} = 5\text{ V}$)

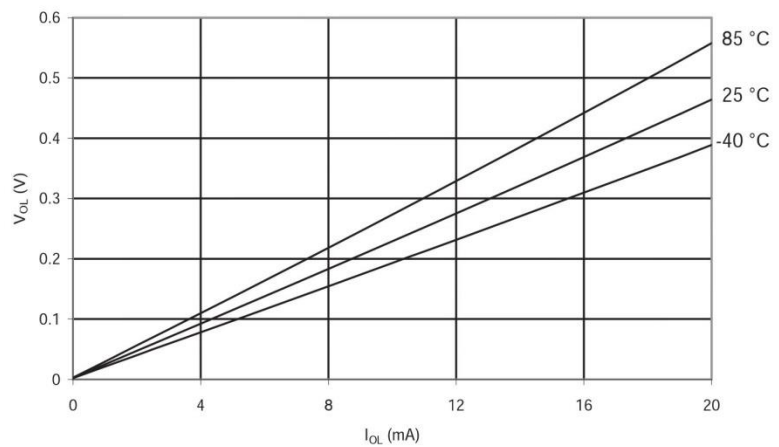


Figure 29-69. ATmega88PA: I/O Pin Output Voltage vs. Source Current ($V_{CC} = 3\text{ V}$)

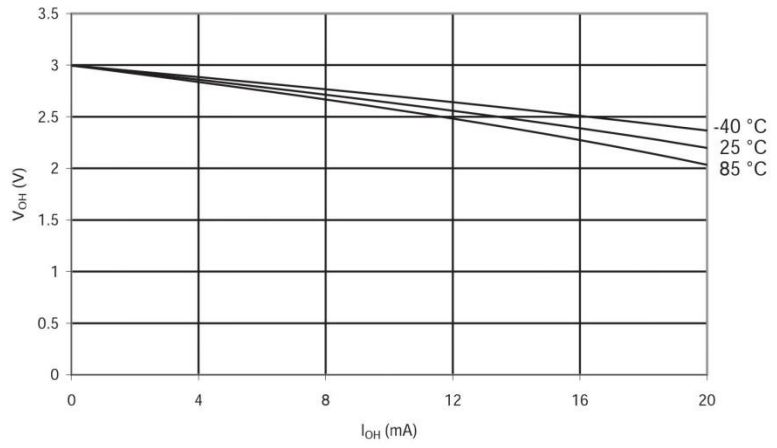
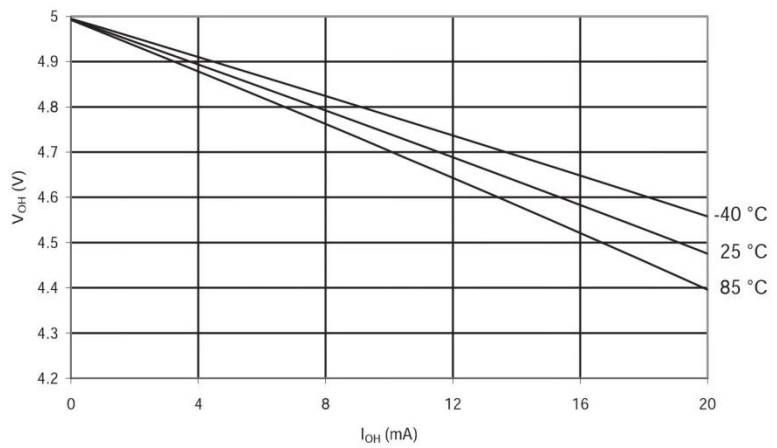


Figure 29-70. ATmega88PA: I/O Pin Output Voltage vs. Source Current ($V_{CC} = 5\text{ V}$)



29.2.9 Pin Threshold and Hysteresis

Figure 29-71. ATmega88PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH} , I/O Pin read as '1')

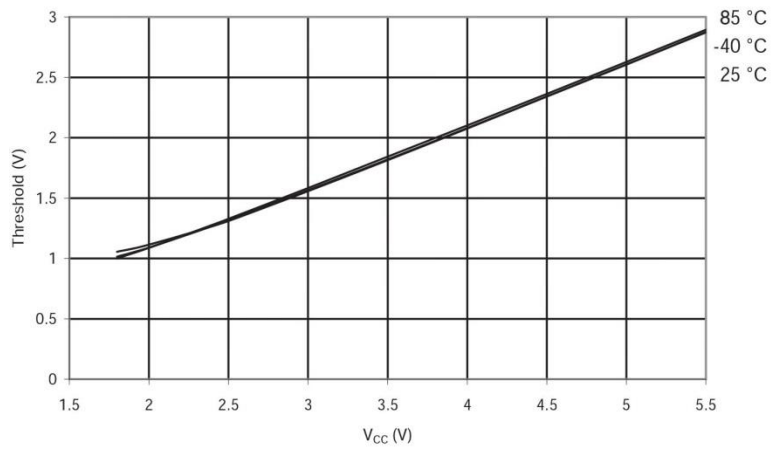


Figure 29-72. ATmega88PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL} , I/O Pin read as '0')

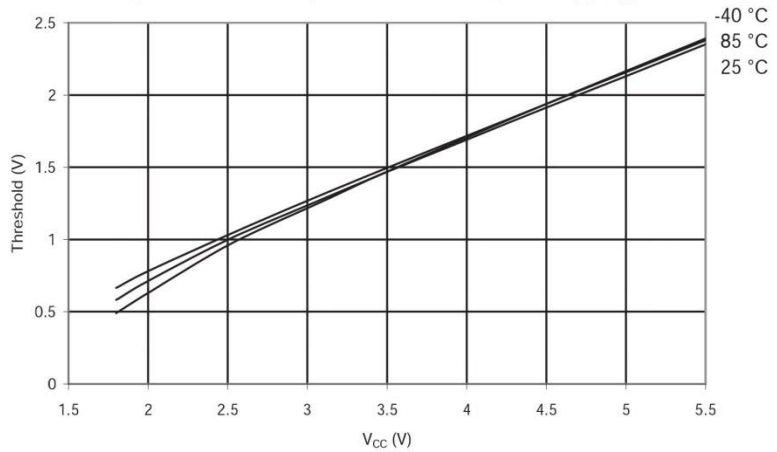


Figure 29-73. ATmega88PA: I/O Pin Input Hysteresis vs. V_{CC}

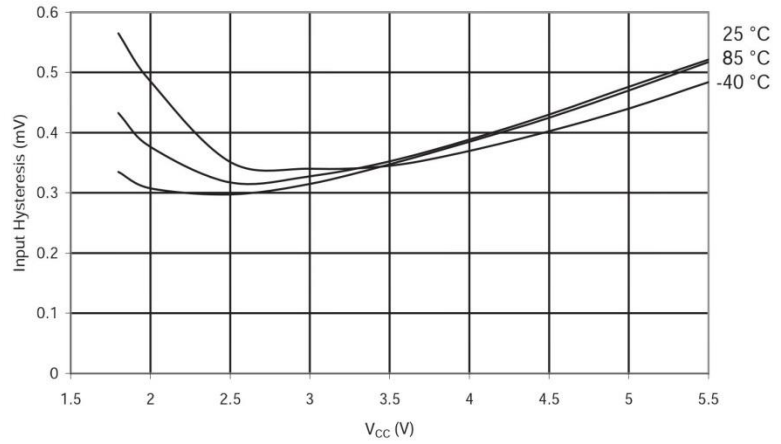


Figure 29-74. ATmega88PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IH} , I/O Pin read as '1')

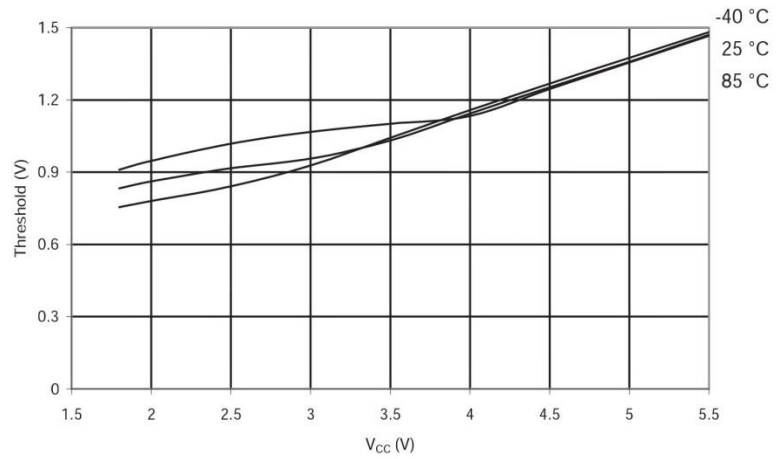


Figure 29-75. ATmega88PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IL} , I/O Pin read as '0')

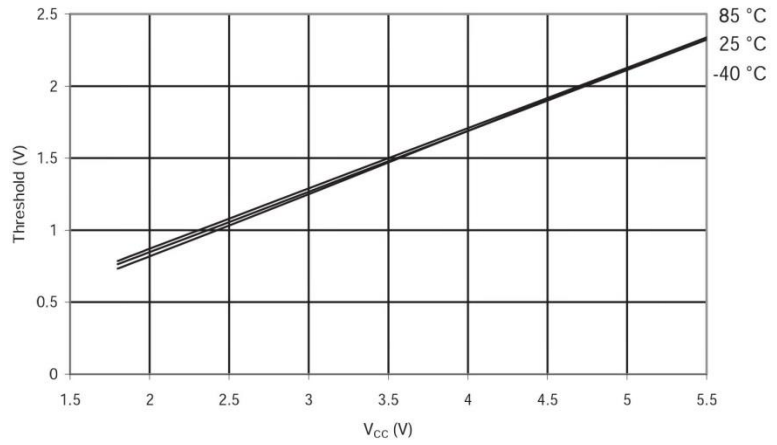
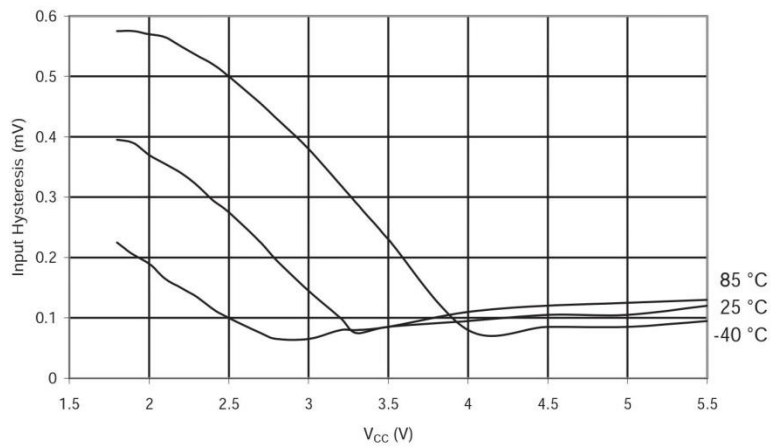


Figure 29-76. ATmega88PA: Reset Pin Input Hysteresis vs. V_{CC}



29.2.10 BOD Threshold

Figure 29-77. ATmega88PA: BOD Thresholds vs. Temperature (BODLEVEL is 1.8 V)

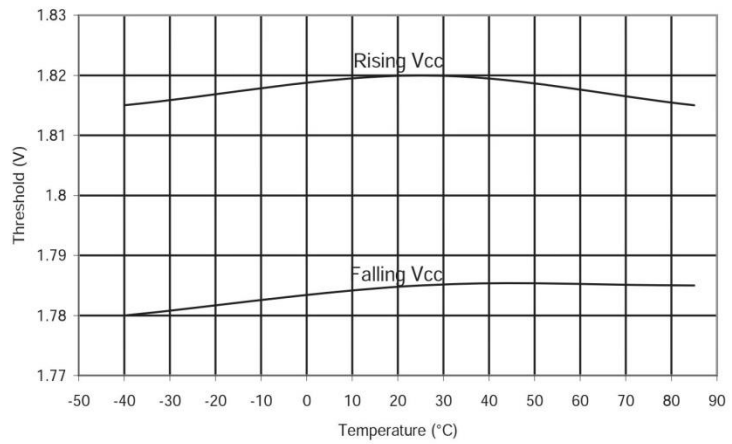


Figure 29-78. ATmega88PA: BOD Thresholds vs. Temperature (BODLEVEL is 2.7 V)

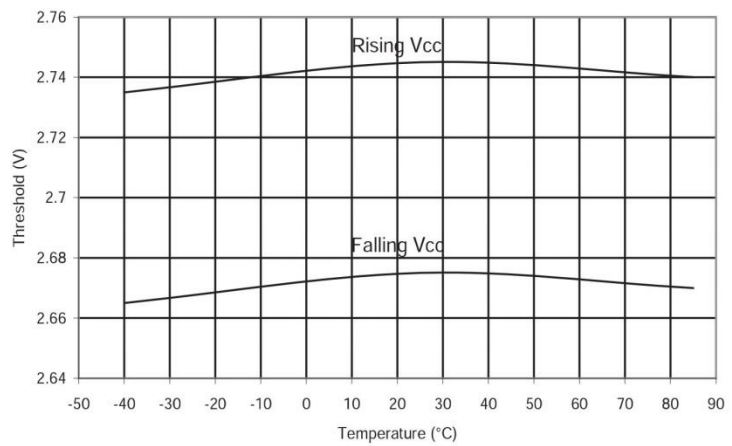
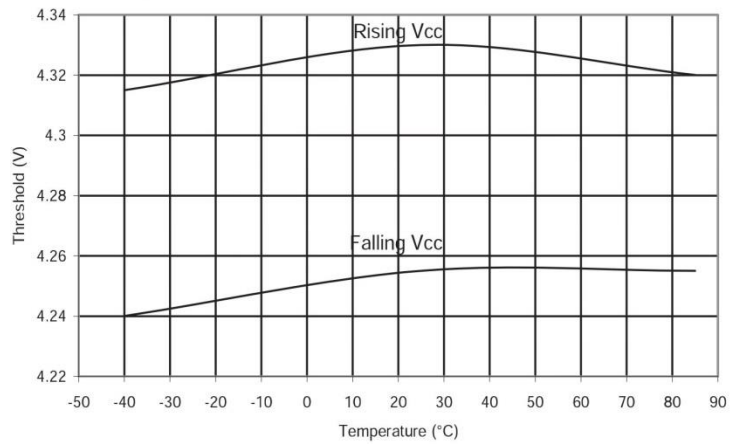


Figure 29-79. ATmega88PA: BOD Thresholds vs. Temperature (BODLEVEL is 4.3 V)



29.2.11 Internal Oscillator Speed

Figure 29-80. ATmega88PA: Watchdog Oscillator Frequency vs. Temperature

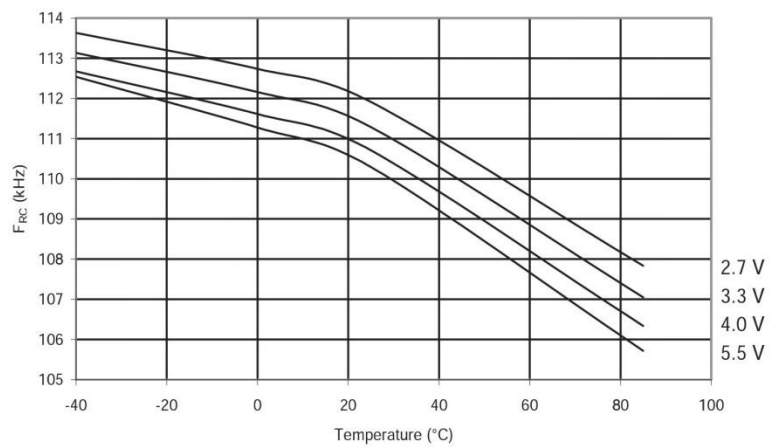


Figure 29-81. ATmega88PA: Watchdog Oscillator Frequency vs. V_{CC}

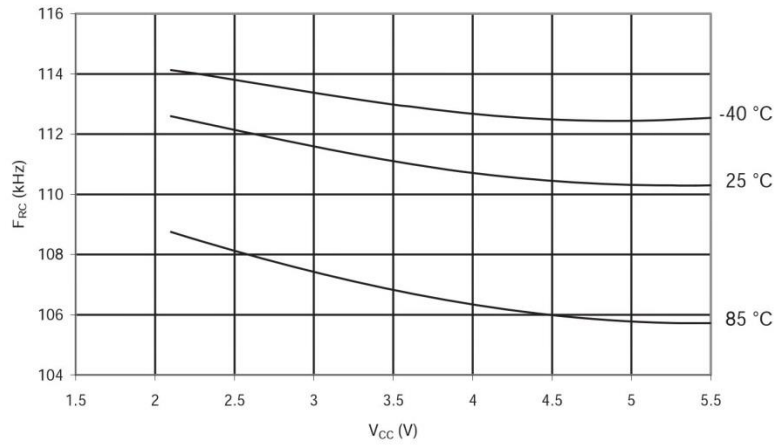


Figure 29-82. ATmega88PA: Calibrated 8 MHz RC Oscillator Frequency vs. V_{CC}

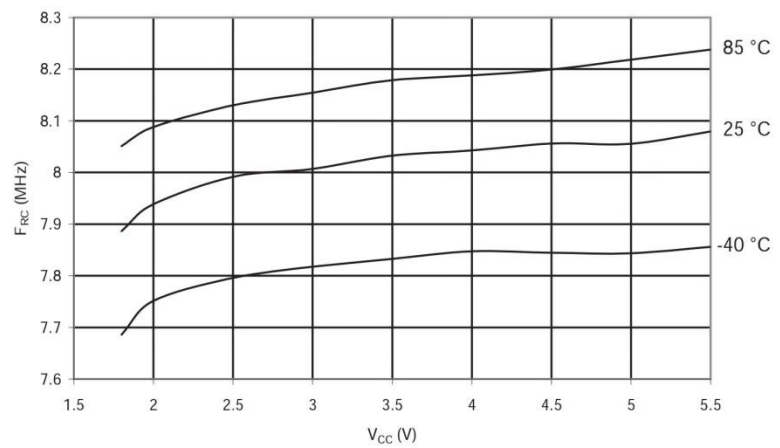


Figure 29-83. ATmega88PA: Calibrated 8 MHz RC Oscillator Frequency vs. Temperature

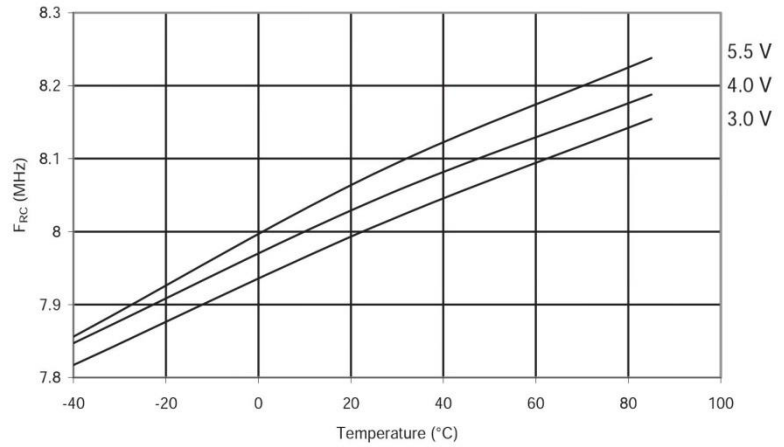
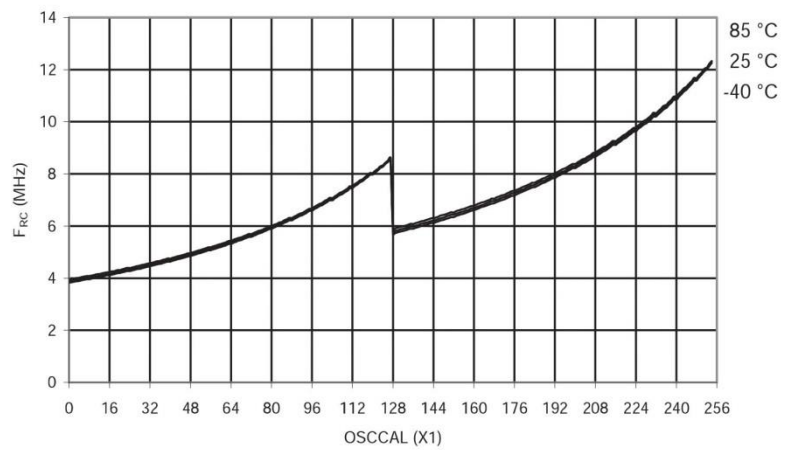


Figure 29-84. ATmega88PA: Calibrated 8 MHz RC Oscillator Frequency vs. OSCCAL Value



29.2.12 Current Consumption of Peripheral Units

Figure 29-85. ATmega88PA: ADC Current vs. V_{CC} (AREF = AV_{CC})

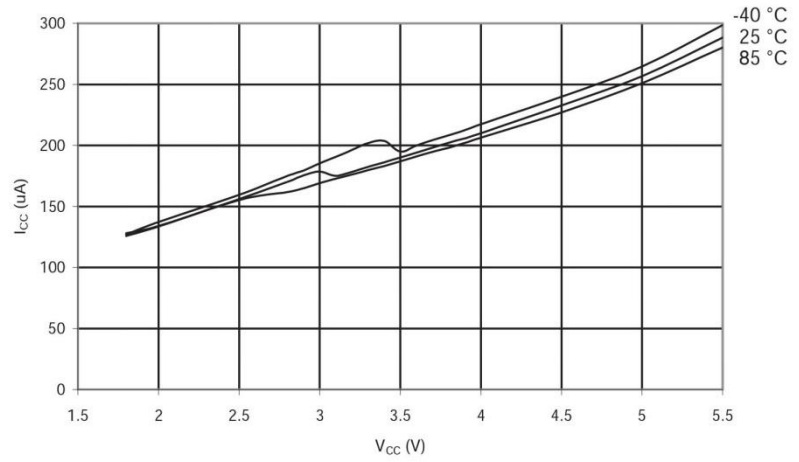


Figure 29-86. ATmega88PA: Analog Comparator Current vs. V_{CC}

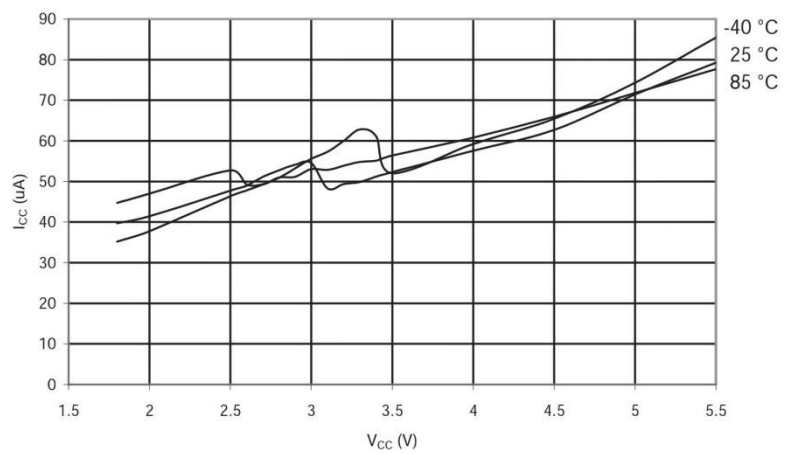


Figure 29-87. ATmega88PA: AREF External Reference Current vs. V_{CC}

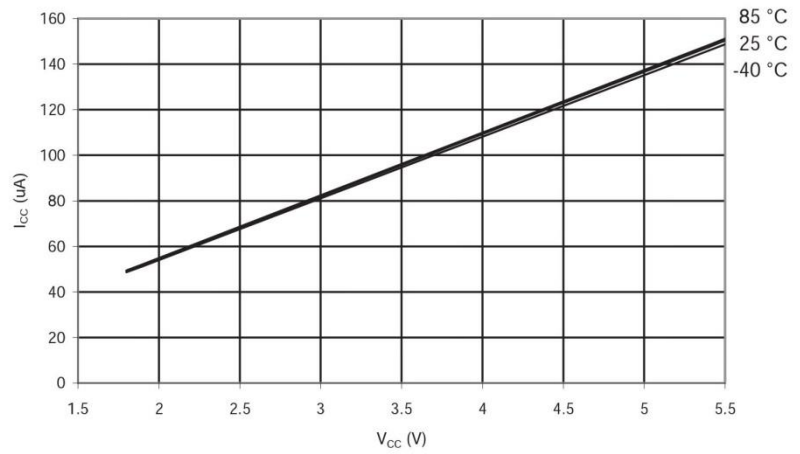


Figure 29-88. ATmega88PA: Brownout Detector Current vs. V_{CC}

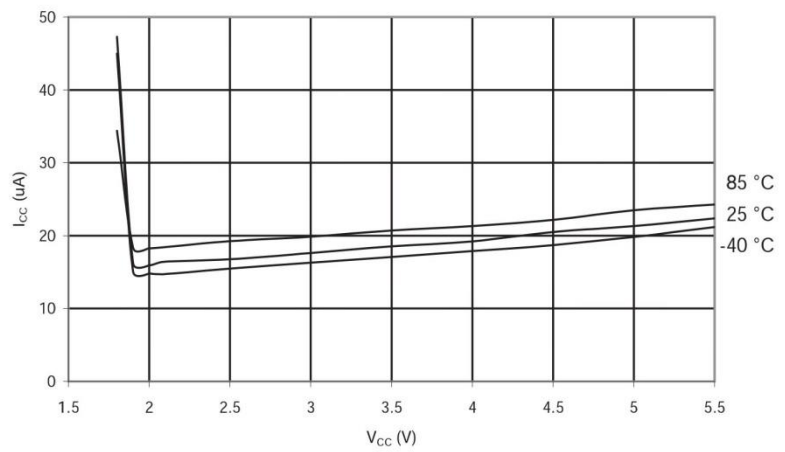
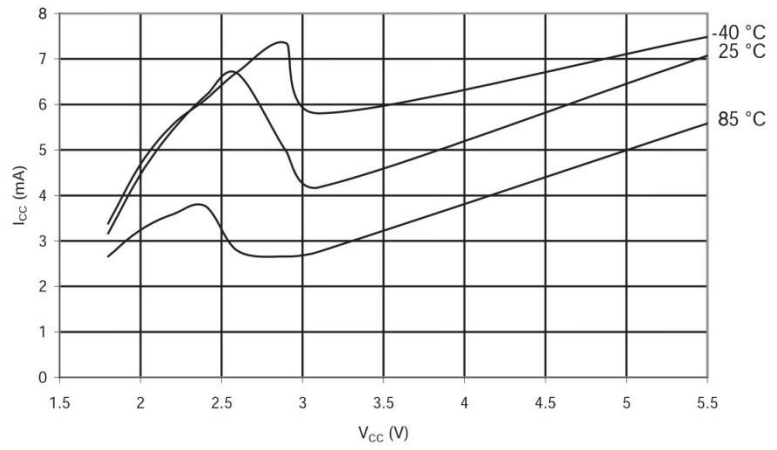


Figure 29-89. ATmega88PA: Programming Current vs. V_{CC}



29.2.13 Current Consumption in Reset and Reset Pulsewidth

Figure 29-90. ATmega88PA: Reset Supply Current vs. Low Frequency (0.1 - 1.0 MHz)

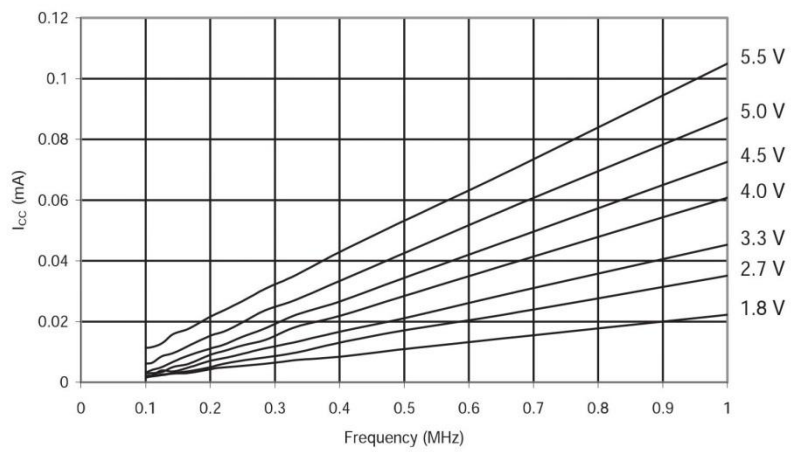


Figure 29-91. ATmega88PA: Reset Supply Current vs. Frequency (1 - 20 MHz)

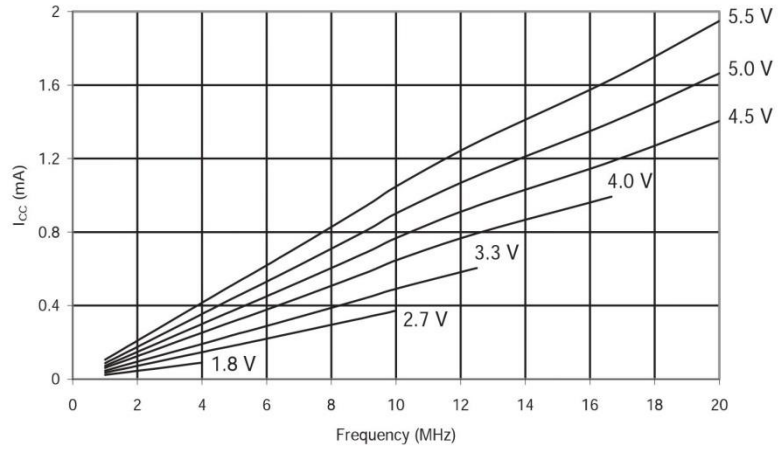
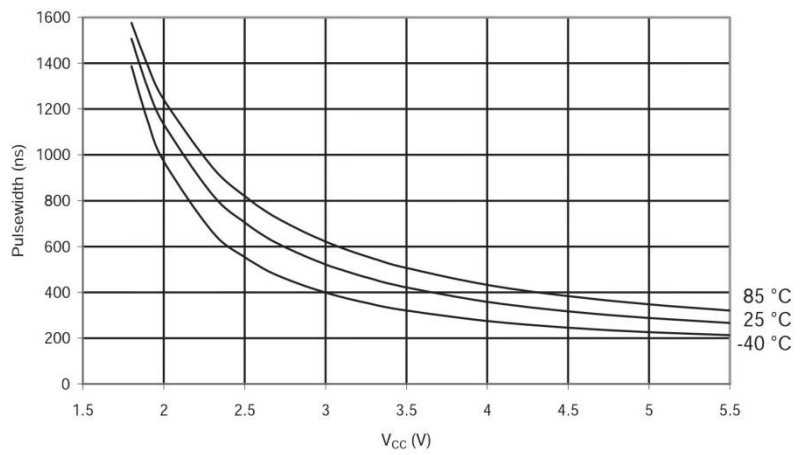


Figure 29-92. ATmega88PA: Minimum Reset Pulse width vs. V_{CC}



29.3 ATmega168PA Typical Characteristics

29.3.1 Active Supply Current

Figure 29-93. ATmega168PA: Active Supply Current vs. Low Frequency (0.1-1.0 MHz)

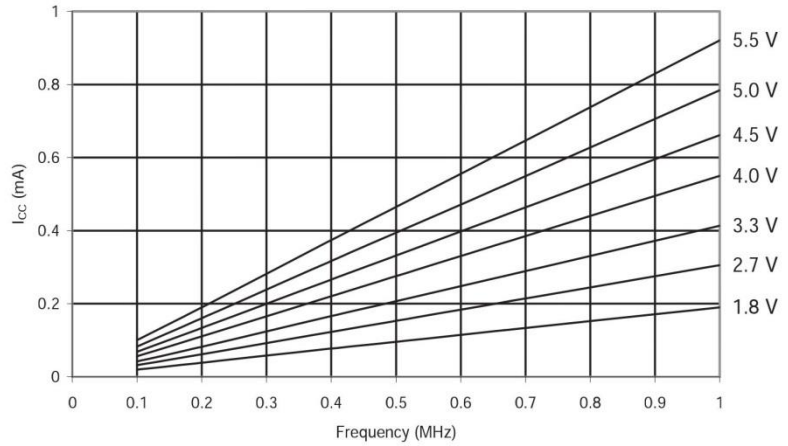


Figure 29-94. ATmega168PA: Active Supply Current vs. Frequency (1-20 MHz)

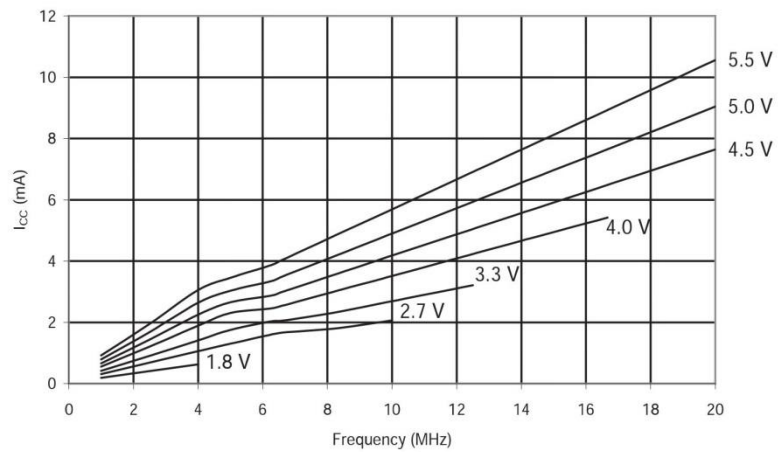


Figure 29-95. ATmega168PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

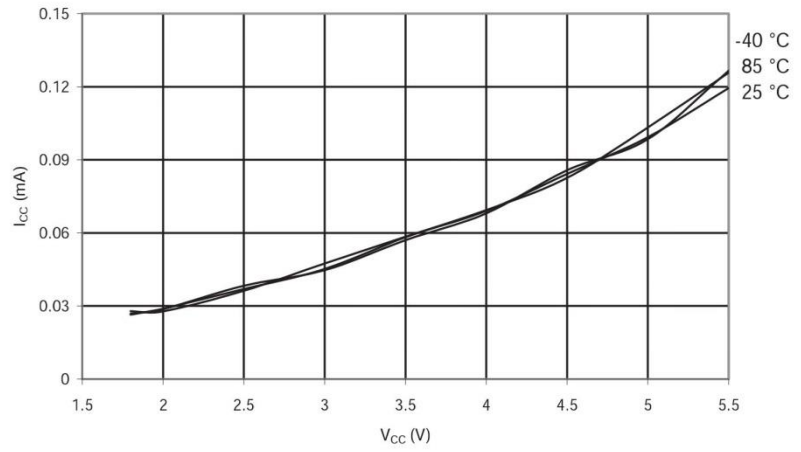


Figure 29-96. ATmega168PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

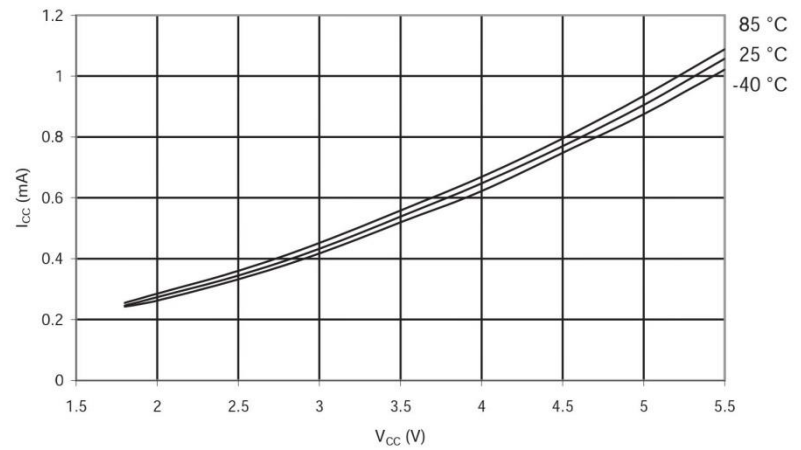
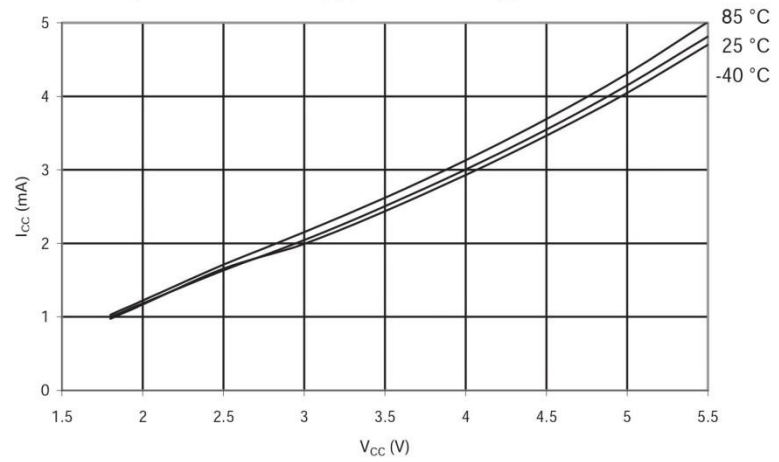


Figure 29-97. ATmega168PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.3.2 Idle Supply Current

Figure 29-98. ATmega168PA: Idle Supply Current vs. Low Frequency (0.1-1.0 MHz)

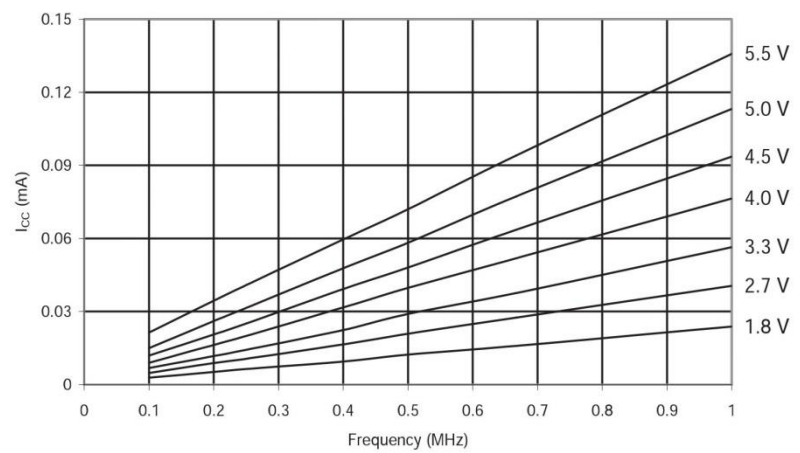


Figure 29-99. ATmega168PA: Idle Supply Current vs. Frequency (1-20 MHz)

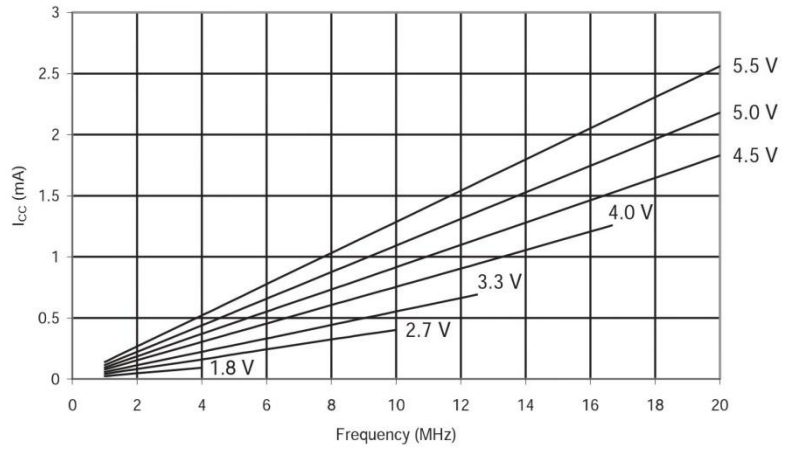


Figure 29-100. ATmega168PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

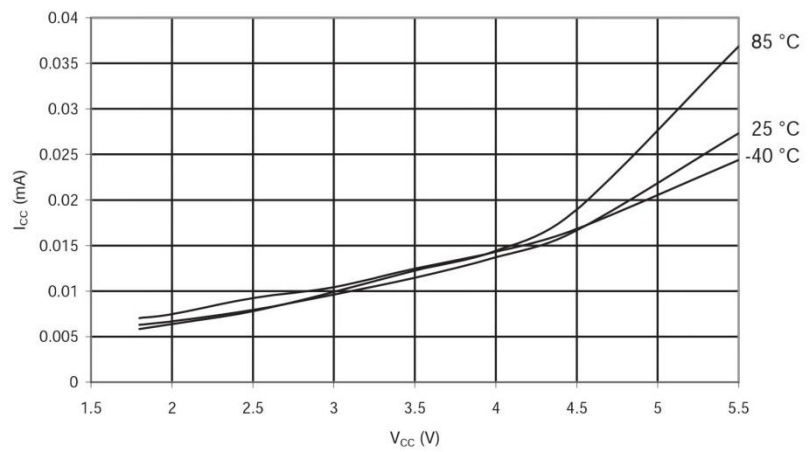


Figure 29-101. ATmega168PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

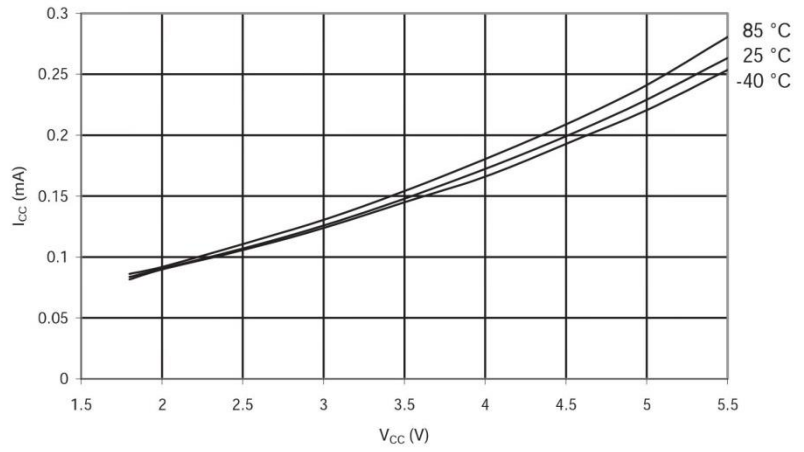
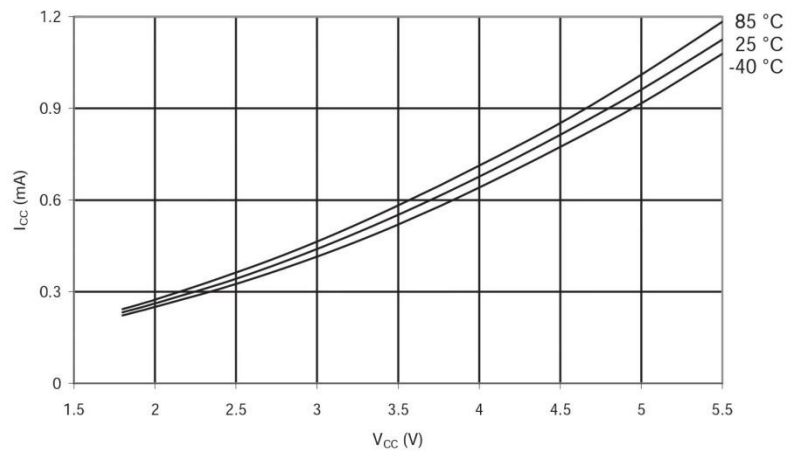


Figure 29-102. ATmega168PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.3.3 ATmega168PA Supply Current of IO Modules

The tables and formulas below can be used to calculate the additional current consumption for the different I/O modules in Active and Idle mode. The enabling or disabling of the I/O modules are controlled by the Power Reduction Register. See "Power Reduction Register" on page 42 for details.

Table 29-5. ATmega168PA: Additional Current Consumption for the different I/O modules (absolute values)

PRR bit	Typical numbers		
	V _{CC} = 2V, F = 1 MHz	V _{CC} = 3V, F = 4 MHz	V _{CC} = 5V, F = 8 MHz
PRUSART0	2.86 μ A	20.3 μ A	52.2 μ A
PRTWI	6.00 μ A	44.1 μ A	122.0 μ A
PRTIM2	4.97 μ A	33.2 μ A	79.8 μ A
PRTIM1	3.50 μ A	23.0 μ A	55.3 μ A
PRTIM0	1.43 μ A	9.2 μ A	21.4 μ A
PRSPI	5.01 μ A	38.6 μ A	111.4 μ A
PRADC	6.34 μ A	45.7 μ A	123.6 μ A

Table 29-6. ATmega168PA: Additional Current Consumption (percentage) in Active and Idle mode

PRR bit	Additional Current consumption compared to Active with external clock (see Figure 29-93 on page 375 and Figure 29-94 on page 375)	Additional Current consumption compared to Idle with external clock (see Figure 29-98 on page 377 and Figure 29-99 on page 378)
PRUSART0	1.5%	8.9%
PRTWI	3.2%	19.5%
PRTIM2	2.4%	14.8%
PRTIM1	1.7%	10.3%
PRTIM0	0.7%	4.1%
PRSPI	2.9%	17.1%
PRADC	3.4%	20.3%

It is possible to calculate the typical current consumption based on the numbers from Table 29-6 on page 380 for other V_{CC} and frequency settings than listed in Table 29-5 on page 380.

Example

Calculate the expected current consumption in idle mode with TIMER1, ADC, and SPI enabled at V_{CC} = 2.0V and F = 1MHz. From Table 29-6 on page 380, third column, we see that we need to add 10.3% for the TIMER1, 20.3% for the ADC, and 17.1% for the SPI module. Reading from Figure 29-98 on page 377, we find that the idle current consumption is -0.027 mA at V_{CC} = 2.0V and F = 1MHz. The total current consumption in idle mode with TIMER1, ADC, and SPI enabled, gives:

$$I_{CCtotal} \approx 0.027 \text{ mA} \cdot (1 + 0.103 + 0.203 + 0.171) \approx 0.040 \text{ mA}$$

29.3.4 Power-down Supply Current

Figure 29-103. ATmega168PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Disabled)

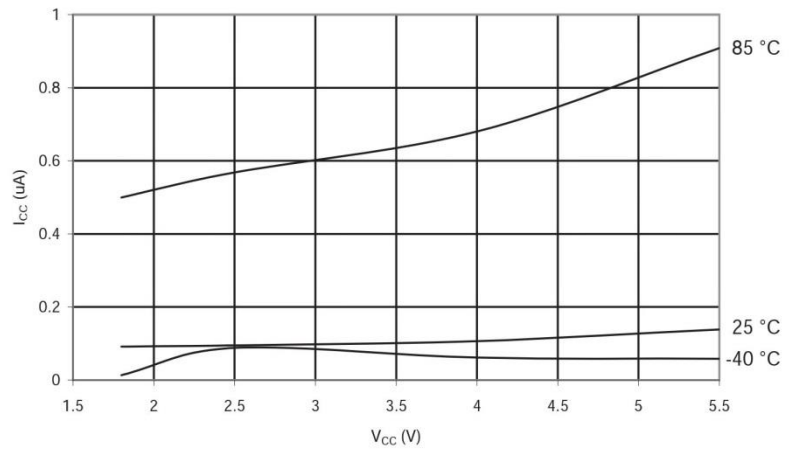
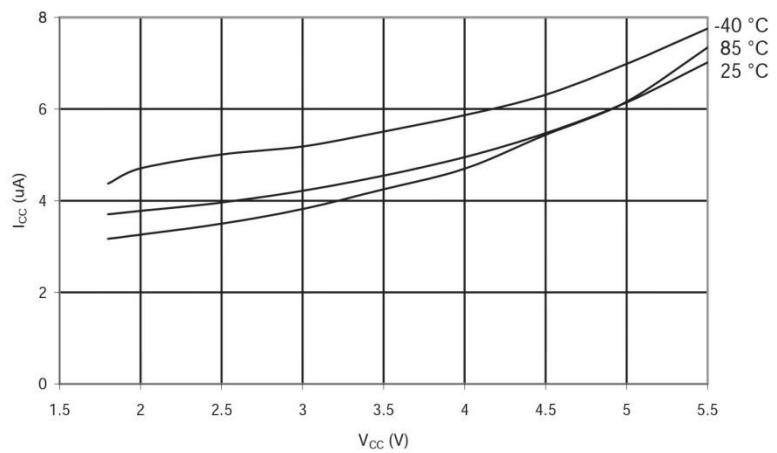
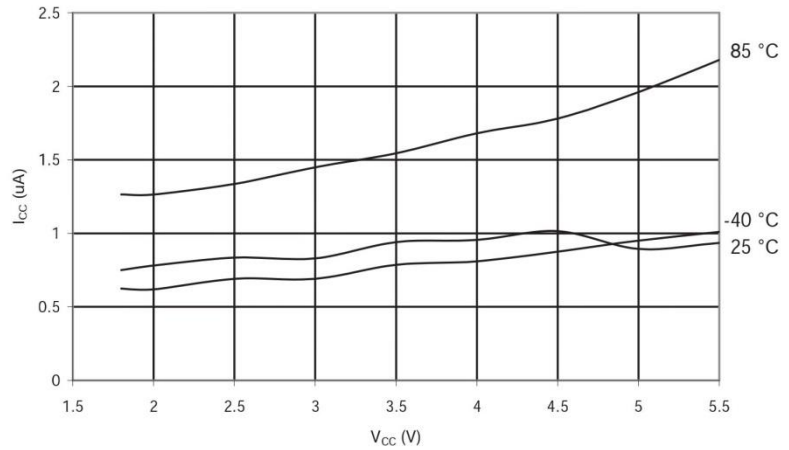


Figure 29-104. ATmega168PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



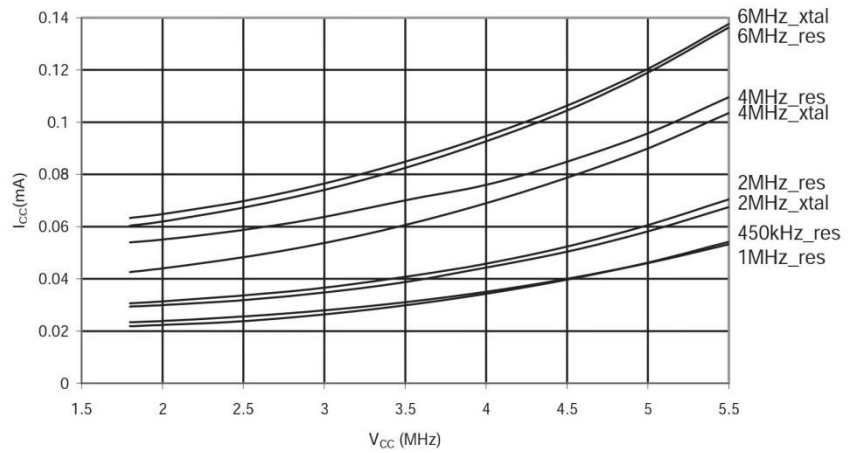
29.3.5 Power-save Supply Current

Figure 29-105. ATmega168PA: Power-Save Supply Current vs. V_{CC} (Watchdog Timer Disabled and 32 kHz Crystal Oscillator Running)



29.3.6 Standby Supply Current

Figure 29-106. ATmega168PA: Standby Supply Current vs. V_{CC} (Watchdog Timer Disabled)



29.3.7 Pin Pull-Up

Figure 29-107. ATmega168PA: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 1.8\text{ V}$)

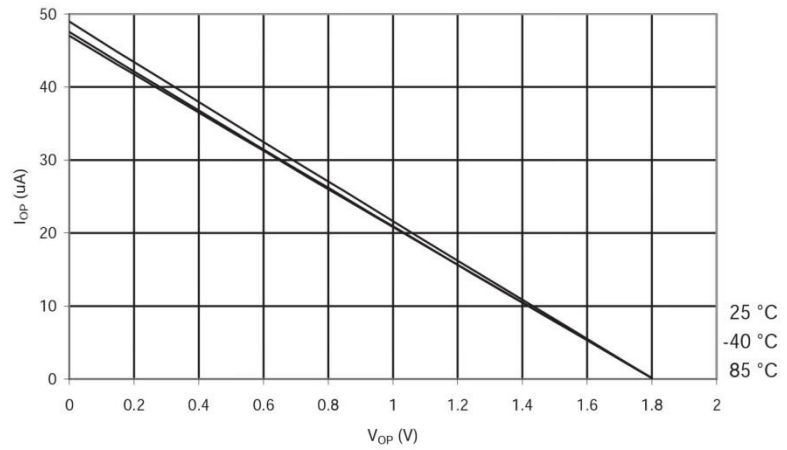


Figure 29-108. ATmega168PA: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 2.7\text{ V}$)

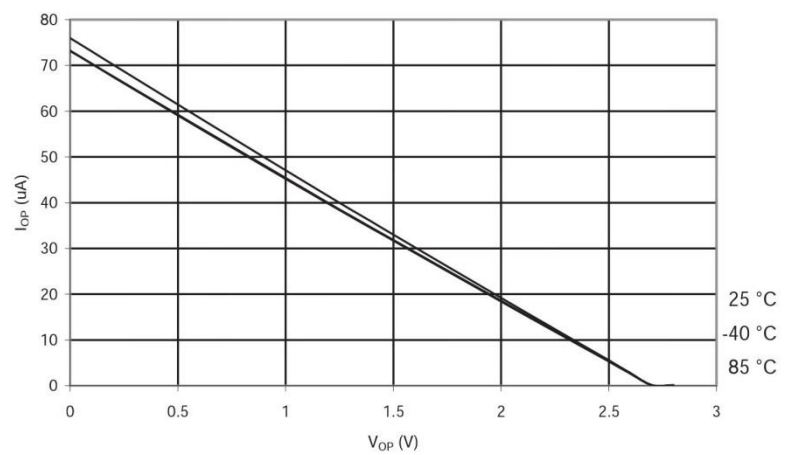


Figure 29-109. ATmega168PA: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 5\text{ V}$)

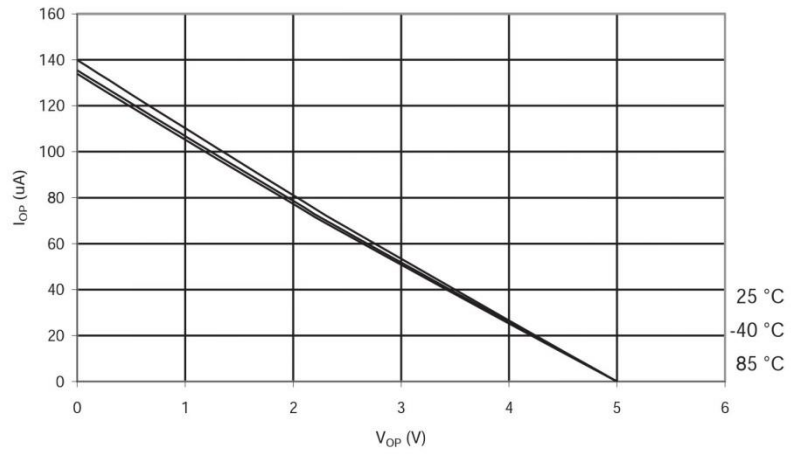


Figure 29-110. ATmega168PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 1.8\text{ V}$)

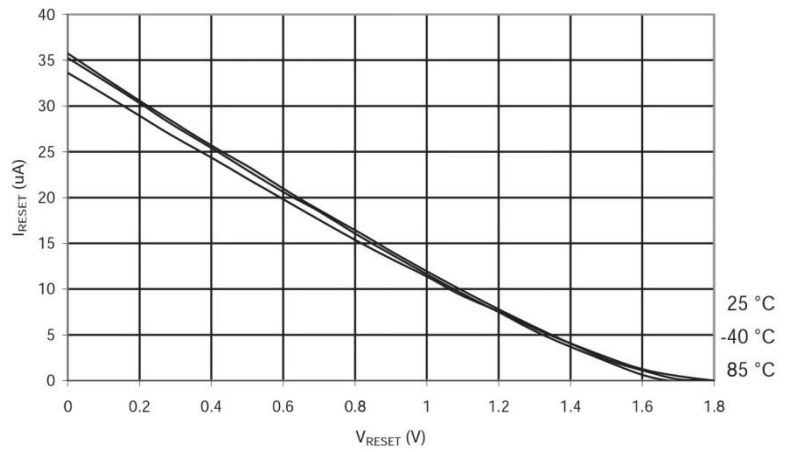


Figure 29-111. ATmega168PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 2.7\text{ V}$)

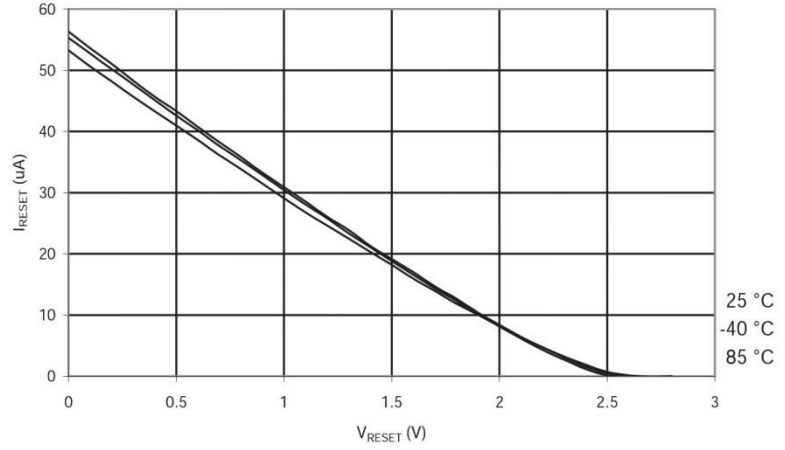
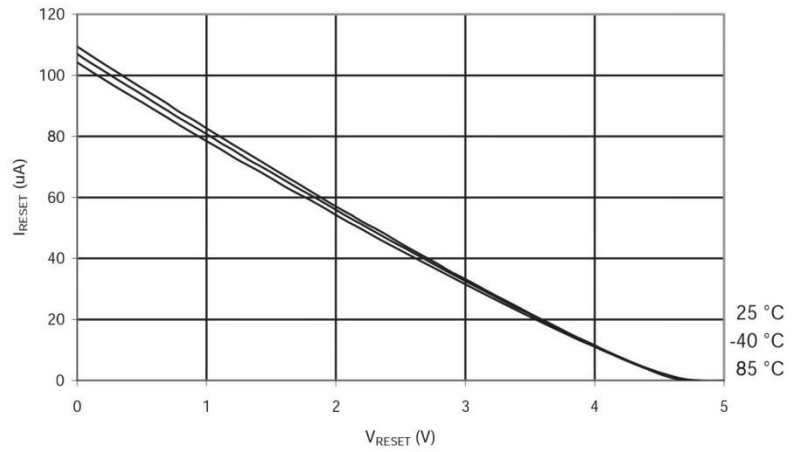


Figure 29-112. ATmega168PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 5\text{ V}$)



29.3.8 Pin Driver Strength

Figure 29-113. ATmega168PA: I/O Pin Output Voltage vs. Sink Current ($V_{CC} = 3\text{ V}$)

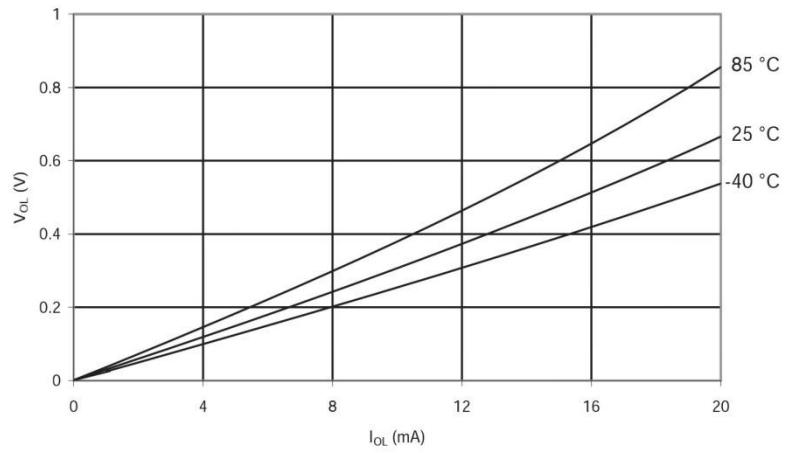


Figure 29-114. ATmega168PA: I/O Pin Output Voltage vs. Sink Current ($V_{CC} = 5\text{ V}$)

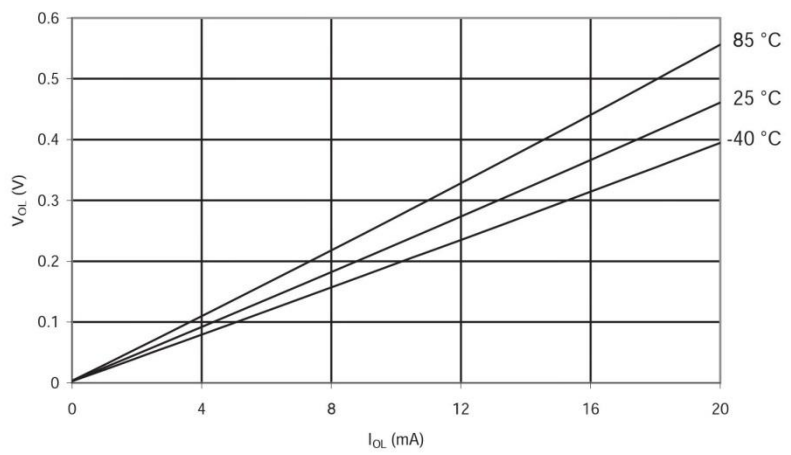


Figure 29-115. ATmega168PA: I/O Pin Output Voltage vs. Source Current ($V_{CC} = 3\text{ V}$)

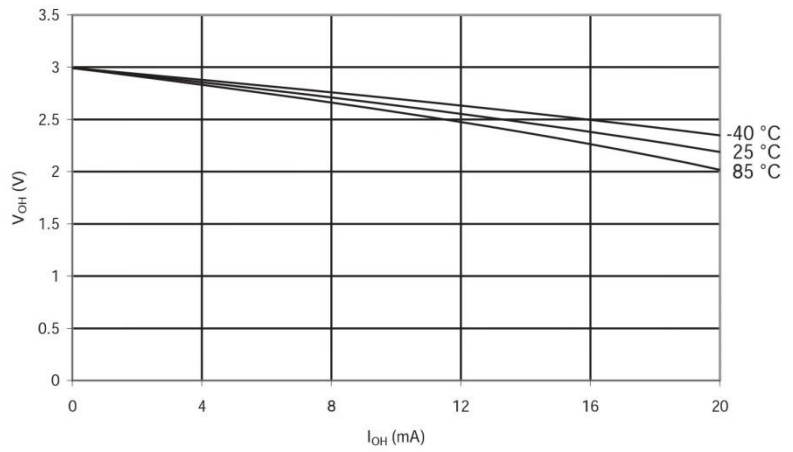
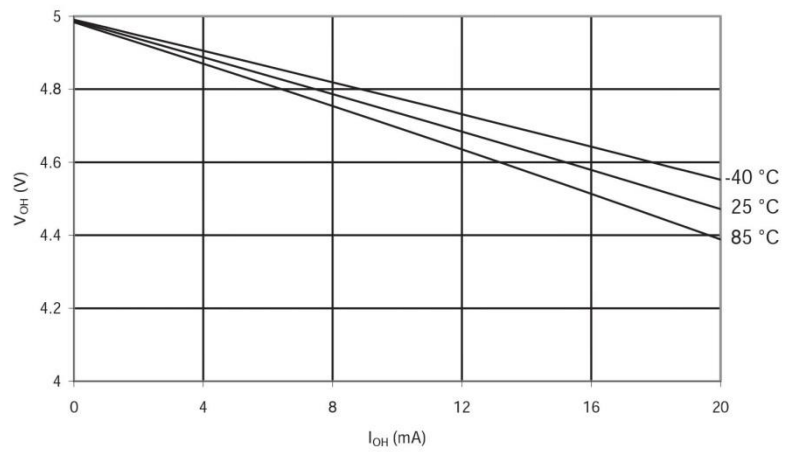


Figure 29-116. ATmega168PA: I/O Pin Output Voltage vs. Source Current ($V_{CC} = 5\text{ V}$)



29.3.9 Pin Threshold and Hysteresis

Figure 29-117. ATmega168PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH} , I/O Pin read as '1')

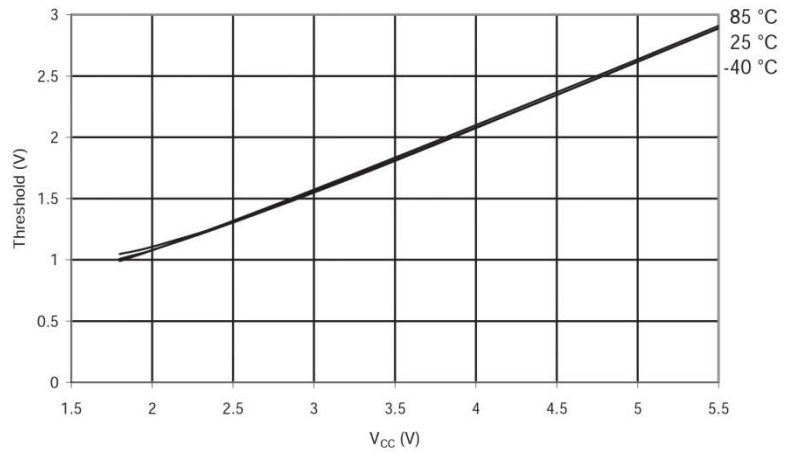


Figure 29-118. ATmega168PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL} , I/O Pin read as '0')

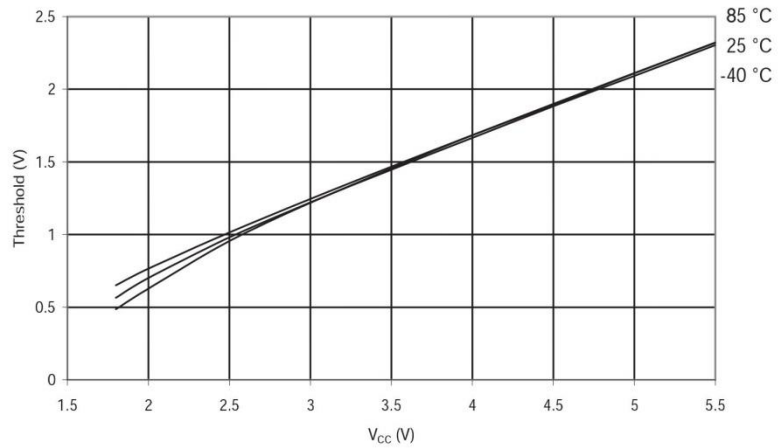


Figure 29-119. ATmega168PA: I/O Pin Input Hysteresis vs. V_{CC}

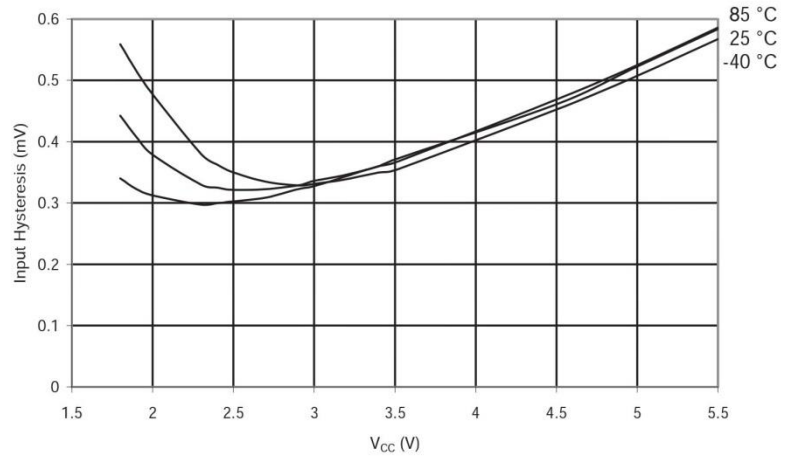


Figure 29-120. ATmega168PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IHr} , I/O Pin read as '1')

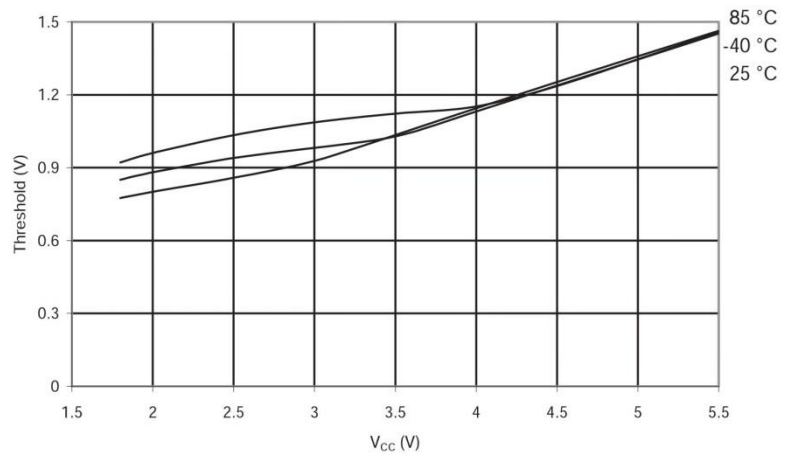


Figure 29-121. ATmega168PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IL} , I/O Pin read as '0')

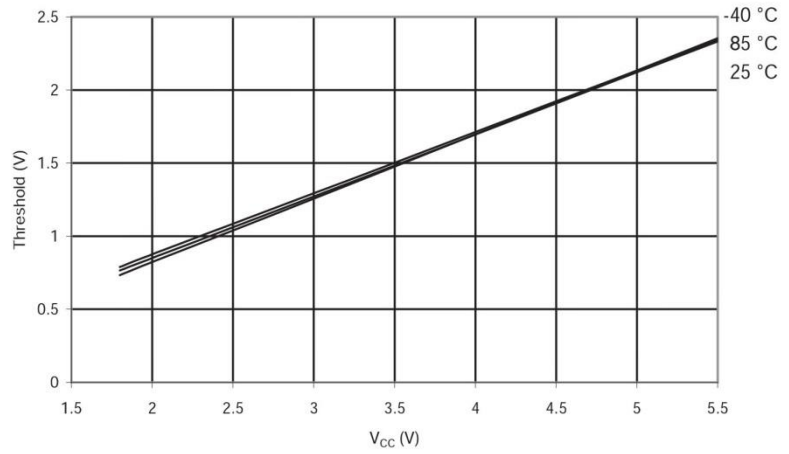
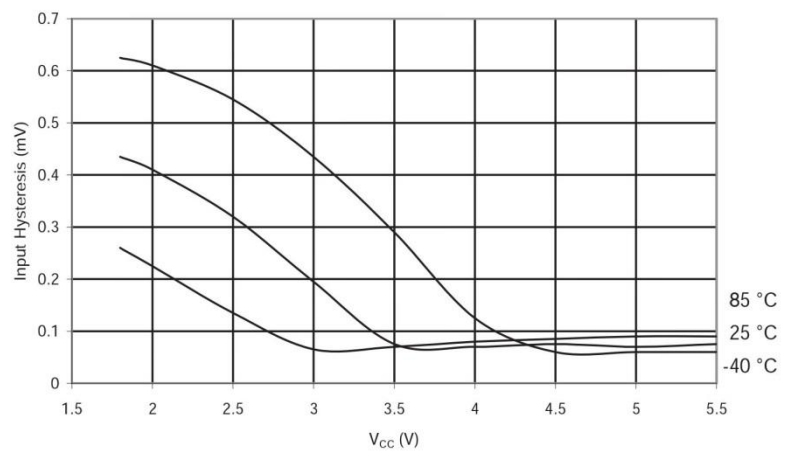


Figure 29-122. ATmega168PA: Reset Pin Input Hysteresis vs. V_{CC}



29.3.10 BOD Threshold

Figure 29-123. ATmega168PA: BOD Thresholds vs. Temperature (BODLEVEL is 1.8 V)

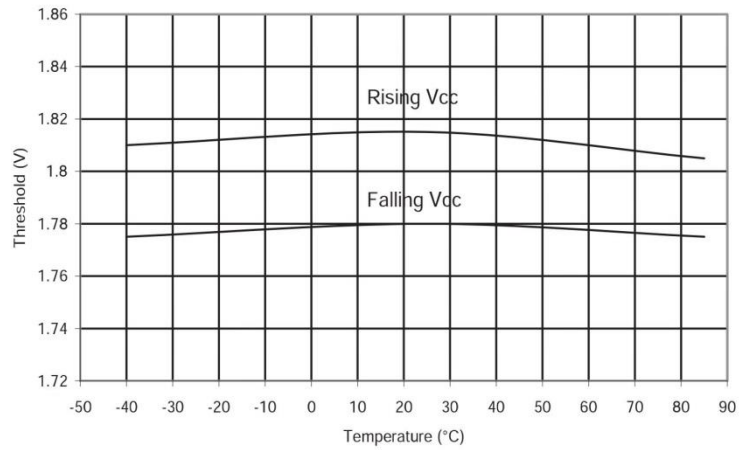


Figure 29-124. ATmega168PA: BOD Thresholds vs. Temperature (BODLEVEL is 2.7 V)

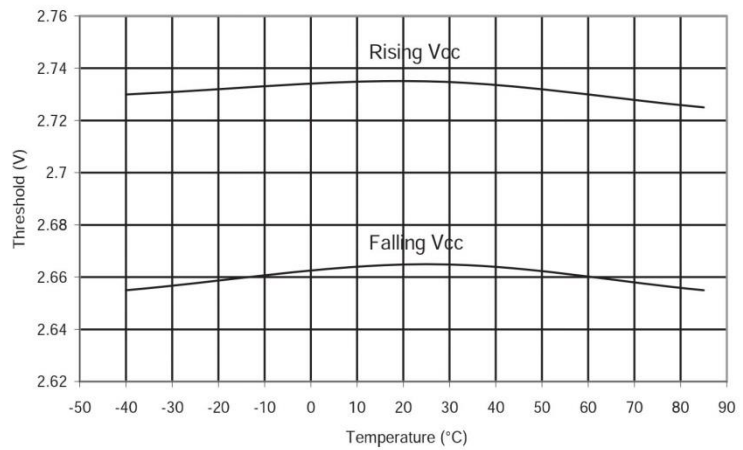
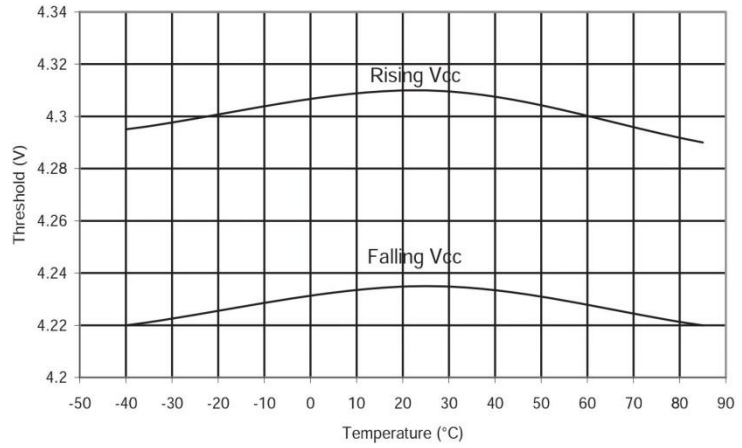


Figure 29-125. ATmega168PA: BOD Thresholds vs. Temperature (BODLEVEL is 4.3 V)



29.3.11 Internal Oscillator Speed

Figure 29-126. ATmega168PA: Watchdog Oscillator Frequency vs. Temperature

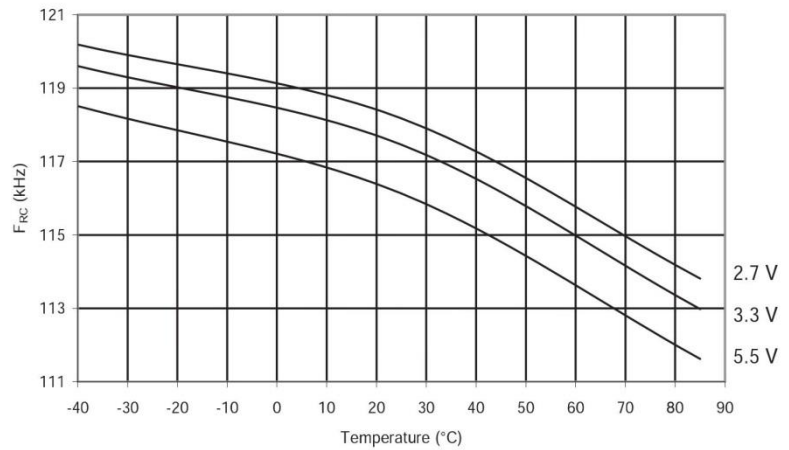


Figure 29-127. ATmega168PA: Watchdog Oscillator Frequency vs. V_{CC}

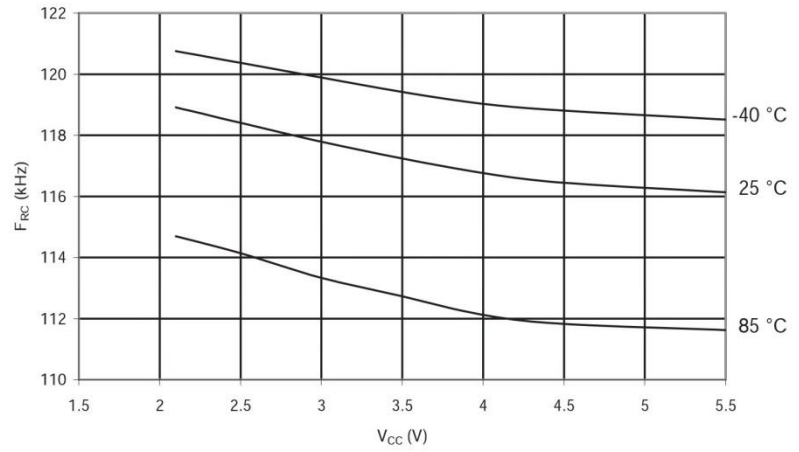


Figure 29-128. ATmega168PA: Calibrated 8 MHz RC Oscillator Frequency vs. V_{CC}

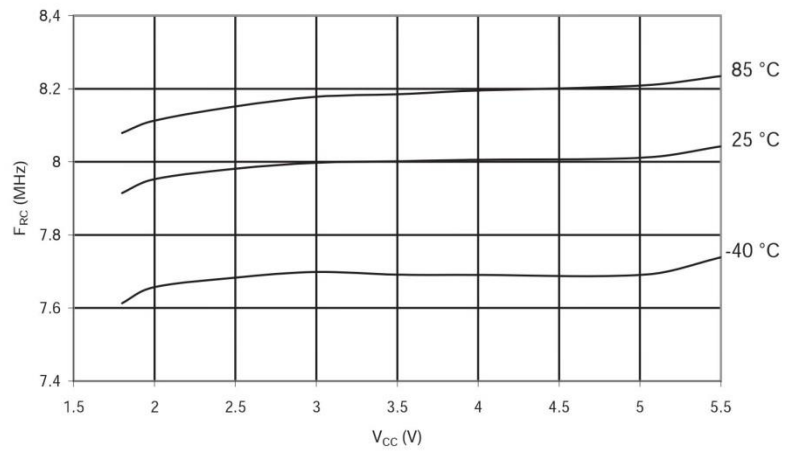


Figure 29-129. ATmega168PA: Calibrated 8 MHz RC Oscillator Frequency vs. Temperature

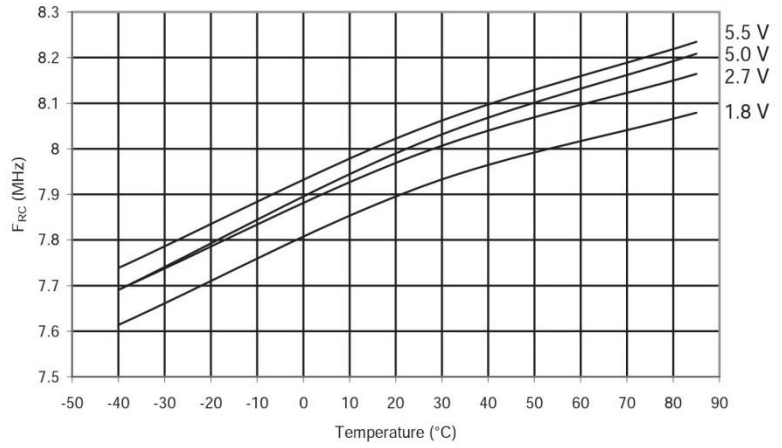
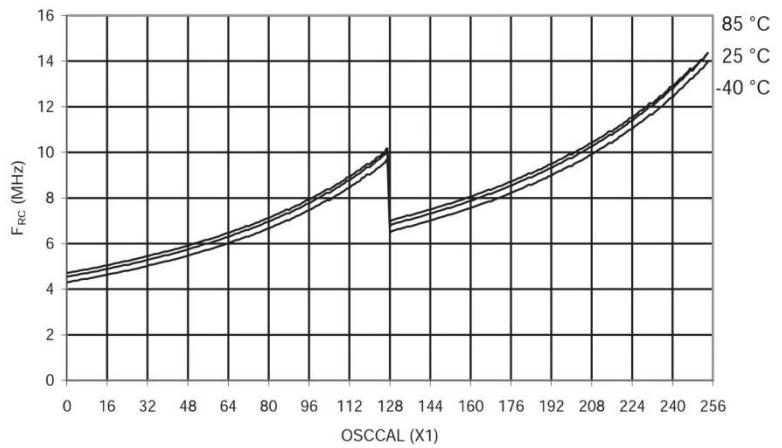


Figure 29-130. ATmega168PA: Calibrated 8 MHz RC Oscillator Frequency vs. OSCCAL Value



29.3.12 Current Consumption of Peripheral Units

Figure 29-131. ATmega168PA: ADC Current vs. V_{CC} (AREF = AV_{CC})

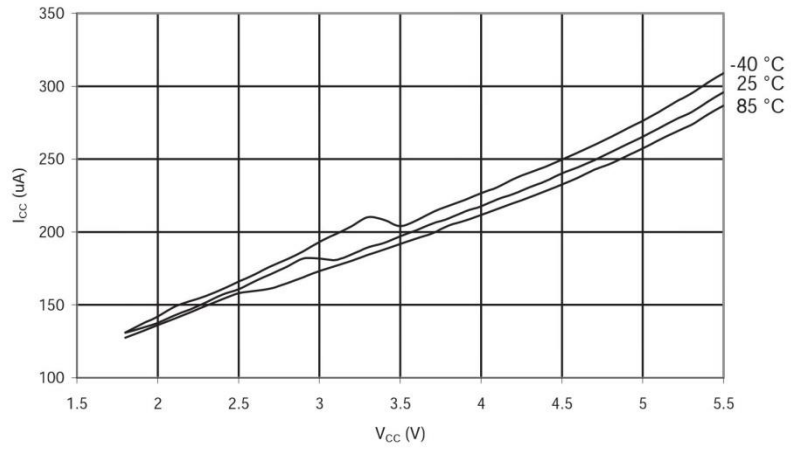


Figure 29-132. ATmega168PA: Analog Comparator Current vs. V_{CC}

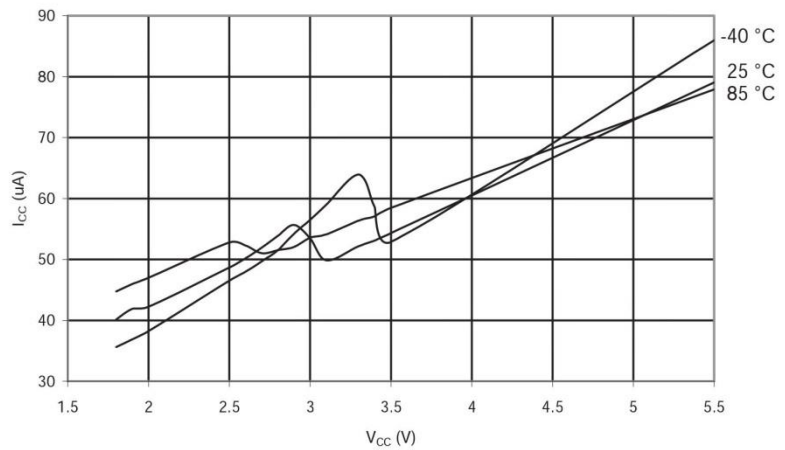


Figure 29-133. ATmega168PA: AREF External Reference Current vs. V_{CC}

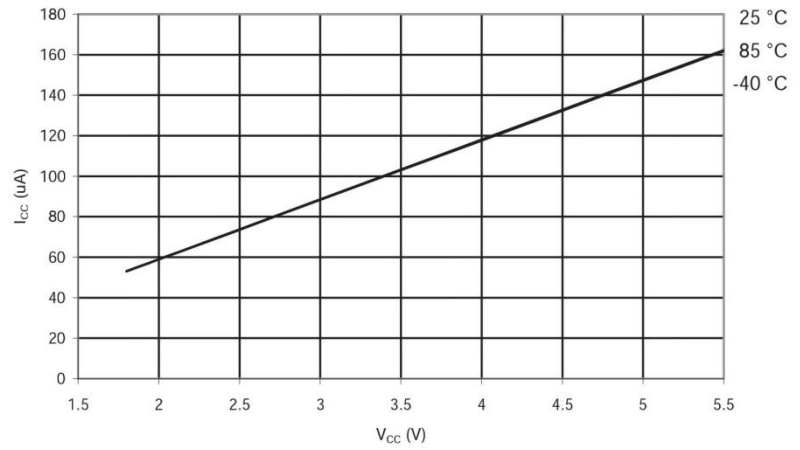


Figure 29-134. ATmega168PA: Brownout Detector Current vs. V_{CC}

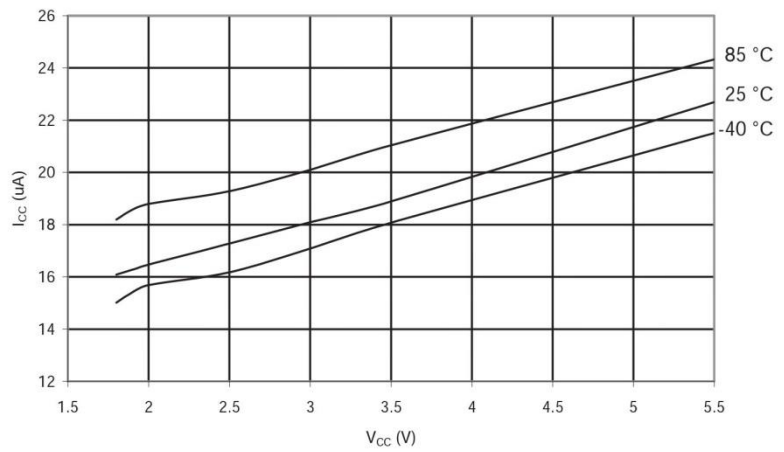
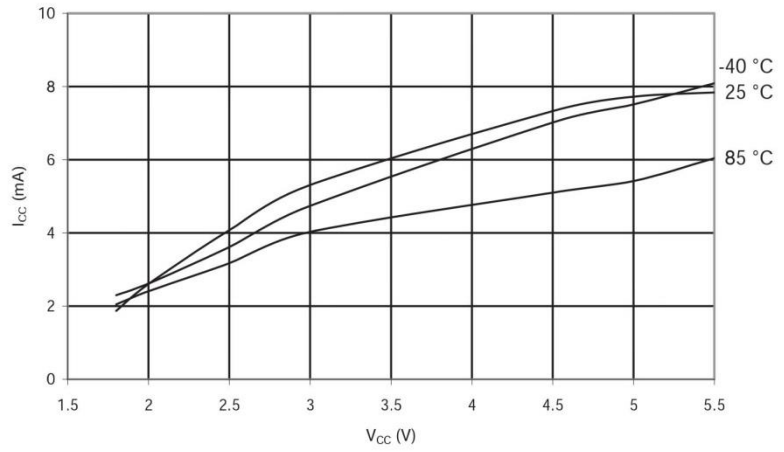


Figure 29-135. ATmega168PA: Programming Current vs. V_{CC}



29.3.13 Current Consumption in Reset and Reset Pulsewidth

Figure 29-136. ATmega168PA: Reset Supply Current vs. Low Frequency (0.1 - 1.0 MHz)

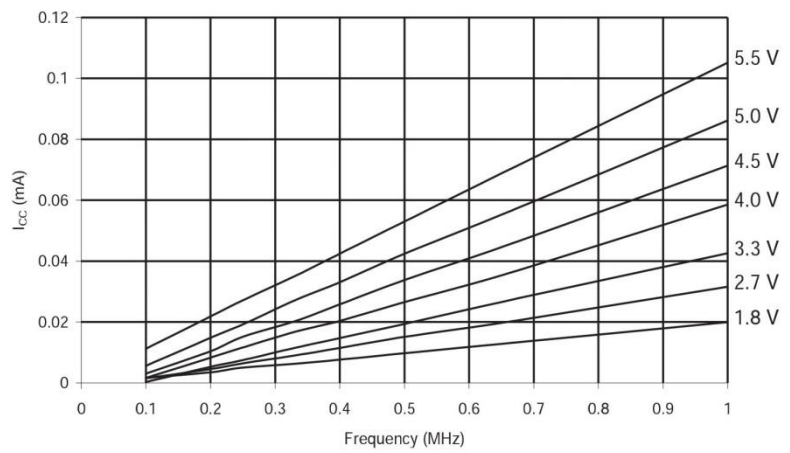


Figure 29-137. ATmega168PA: Reset Supply Current vs. Frequency (1 - 20 MHz)

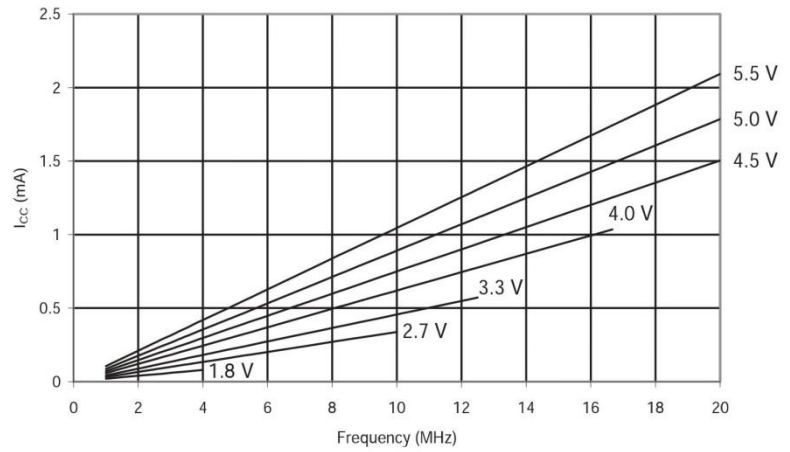
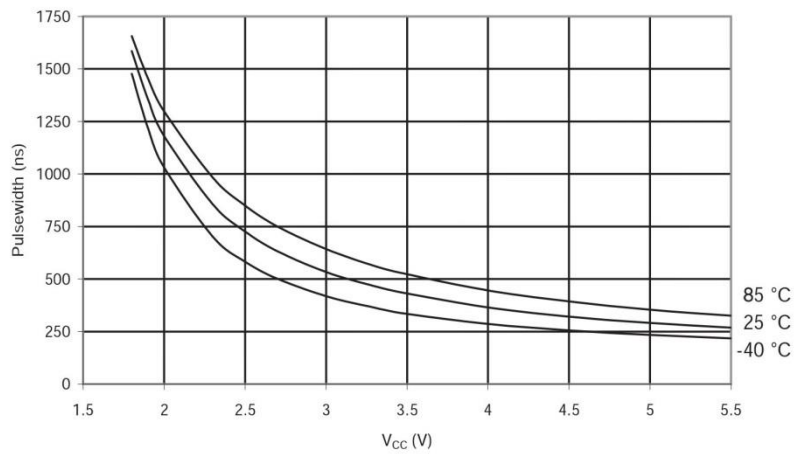


Figure 29-138. ATmega168PA: Minimum Reset Pulse width vs. V_{CC}



29.4 ATmega328P Typical Characteristics

29.4.1 Active Supply Current

Figure 29-139. ATmega328P: Active Supply Current vs. Low Frequency (0.1-1.0 MHz)

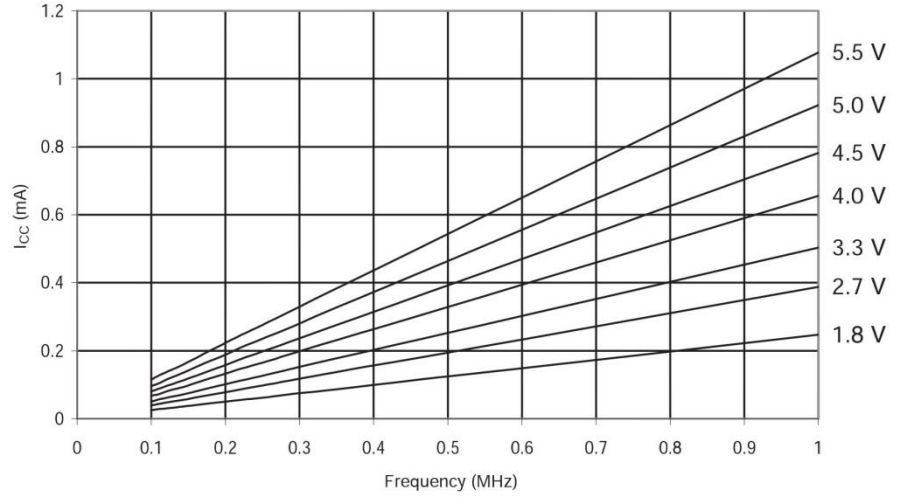


Figure 29-140. ATmega328P: Active Supply Current vs. Frequency (1-20 MHz)

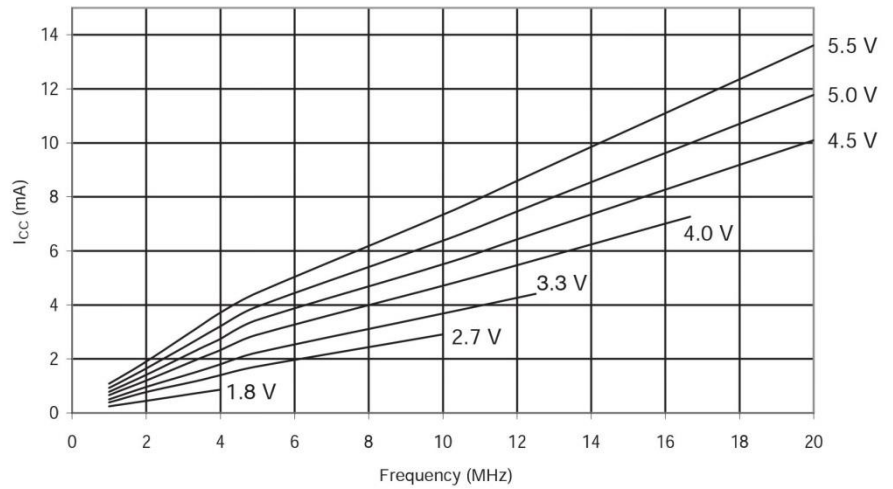


Figure 29-141. ATmega328P: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

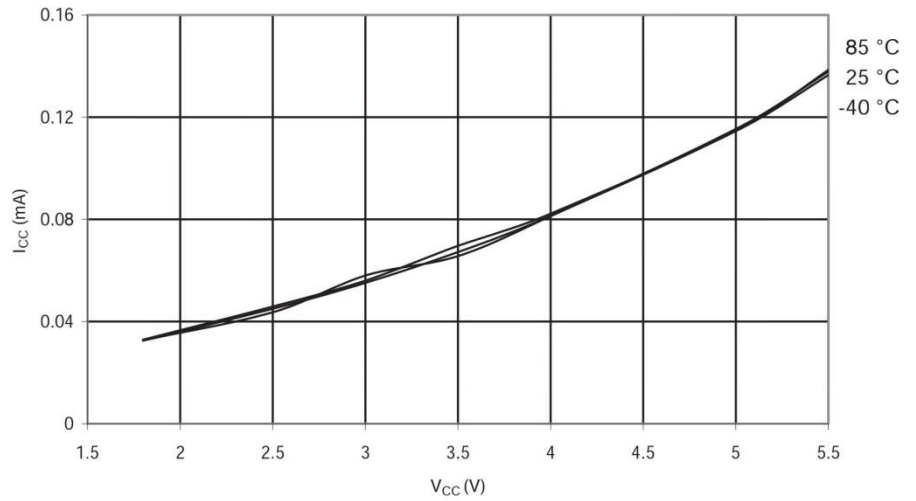


Figure 29-142. ATmega328P: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

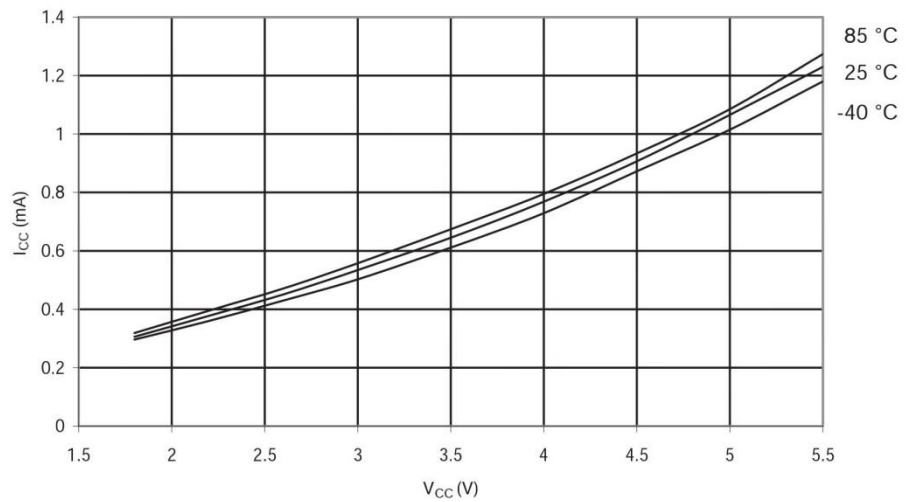
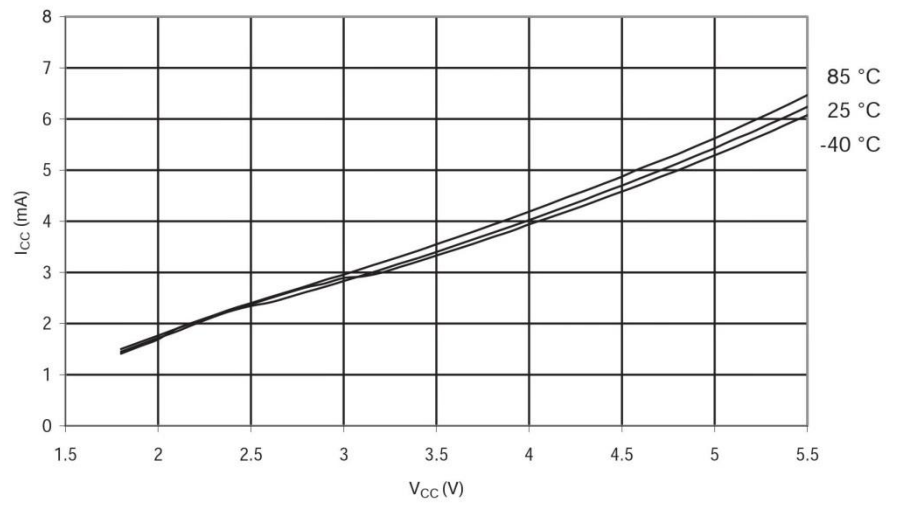


Figure 29-143. ATmega328P: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.4.2 Idle Supply Current

Figure 29-144. ATmega328P: Idle Supply Current vs. Low Frequency (0.1-1.0 MHz)

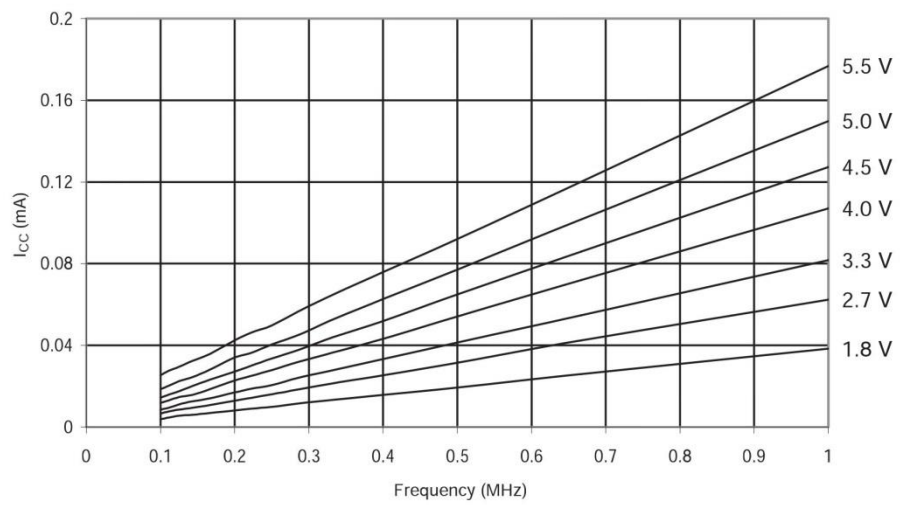


Figure 29-145. ATmega328P: Idle Supply Current vs. Frequency (1-20 MHz)

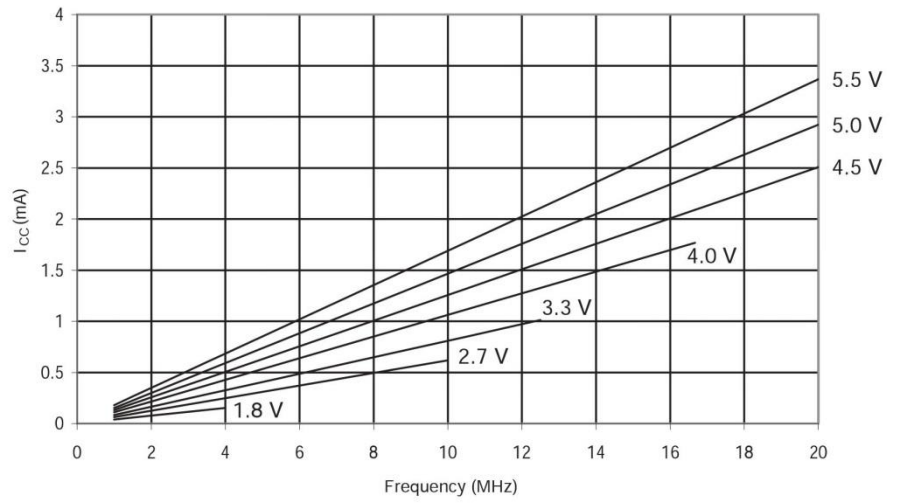


Figure 29-146. ATmega328P: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

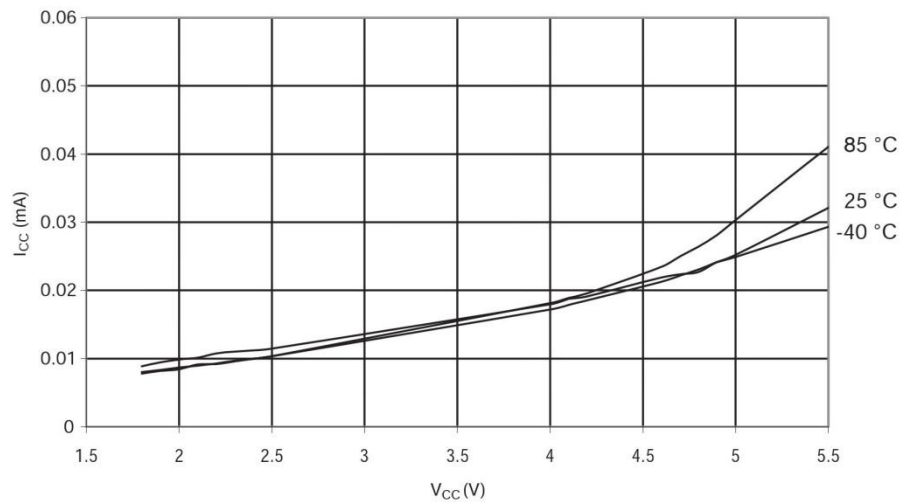


Figure 29-147. ATmega328P: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

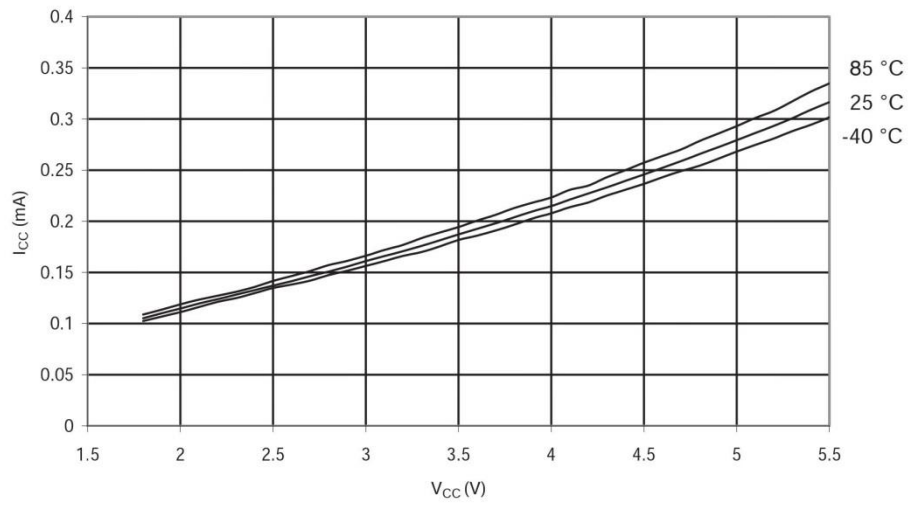
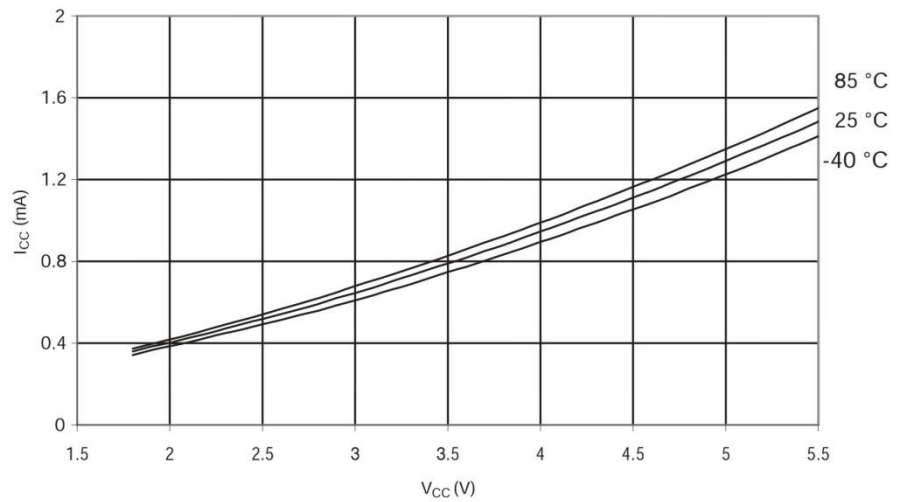


Figure 29-148. ATmega328P: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.4.3 ATmega328P Supply Current of IO Modules

The tables and formulas below can be used to calculate the additional current consumption for the different I/O modules in Active and Idle mode. The enabling or disabling of the I/O modules are controlled by the Power Reduction Register. See "Power Reduction Register" on page 42 for details.

Table 29-7. ATmega328P: Additional Current Consumption for the different I/O modules (absolute values)

PRR bit	Typical numbers		
	V _{CC} = 2V, F = 1MHz	V _{CC} = 3V, F = 4MHz	V _{CC} = 5V, F = 8MHz
PRUSART0	3.20 μA	22.17 μA	100.25 μA
PRTWI	7.34 μA	46.55 μA	199.25 μA
PRTIM2	7.34 μA	50.79 μA	224.25 μA
PRTIM1	6.19 μA	41.25 μA	176.25 μA
PRTIM0	1.89 μA	14.28 μA	61.13 μA
PRSPI	6.94 μA	43.84 μA	186.50 μA
PRADC	8.66 μA	61.80 μA	295.38 μA

Table 29-8. ATmega328P: Additional Current Consumption (percentage) in Active and Idle mode

PRR bit	Additional Current consumption compared to Active with external clock (see Figure 29-139 on page 399 and Figure 29-140 on page 399)	Additional Current consumption compared to Idle with external clock (see Figure 29-144 on page 401 and Figure 29-145 on page 402)
PRUSART0	1.4 %	7.8%
PRTWI	3.0 %	16.6 %
PRTIM2	3.3 %	17.8 %
PRTIM1	2.7 %	14.5 %
PRTIM0	0.9 %	4.8 %
PRSPI	2.9 %	15.7 %
PRADC	4.1 %	22.1 %

It is possible to calculate the typical current consumption based on the numbers from Table 29-8 on page 404 for other V_{CC} and frequency settings than listed in Table 29-7 on page 404.

Example

Calculate the expected current consumption in idle mode with TIMER1, ADC, and SPI enabled at V_{CC} = 2.0V and F = 1MHz. From Table 29-8 on page 404, third column, we see that we need to add 14.5% for the TIMER1, 22.1% for the ADC, and 15.7% for the SPI module. Reading from Figure 29-145 on page 402, we find that the idle current consumption is ~0.055 mA at V_{CC} = 2.0V and F = 1MHz. The total current consumption in idle mode with TIMER1, ADC, and SPI enabled, gives:

$$I_{CCtotal} \approx 0.045 \text{ mA} \cdot (1 + 0.145 + 0.221 + 0.157) \approx 0.069 \text{ mA}$$

29.4.4 Power-down Supply Current

Figure 29-149. ATmega328P: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Disabled)

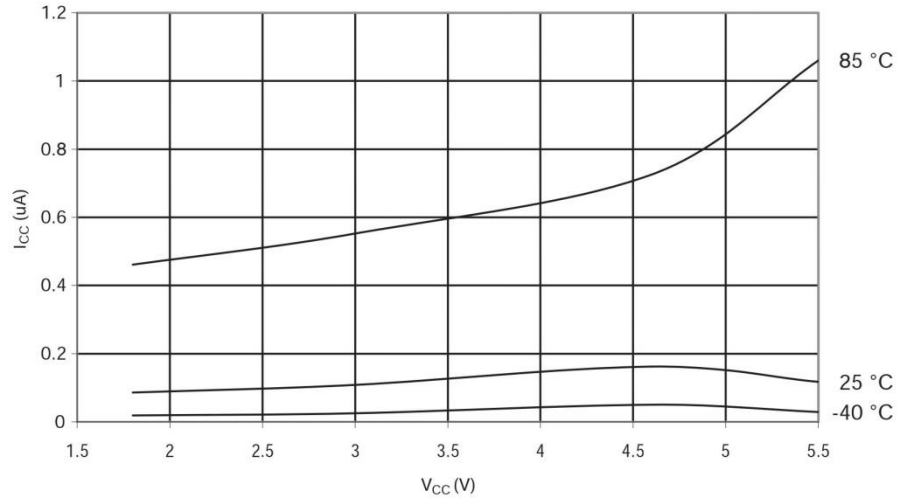
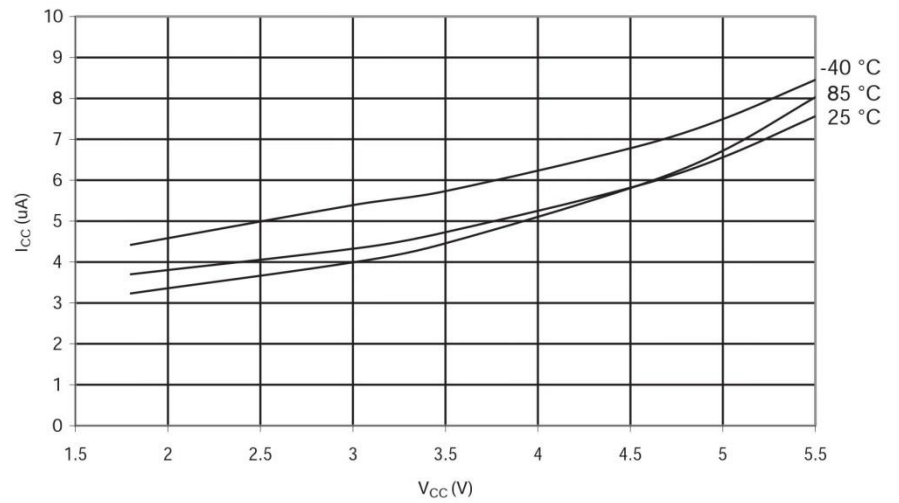
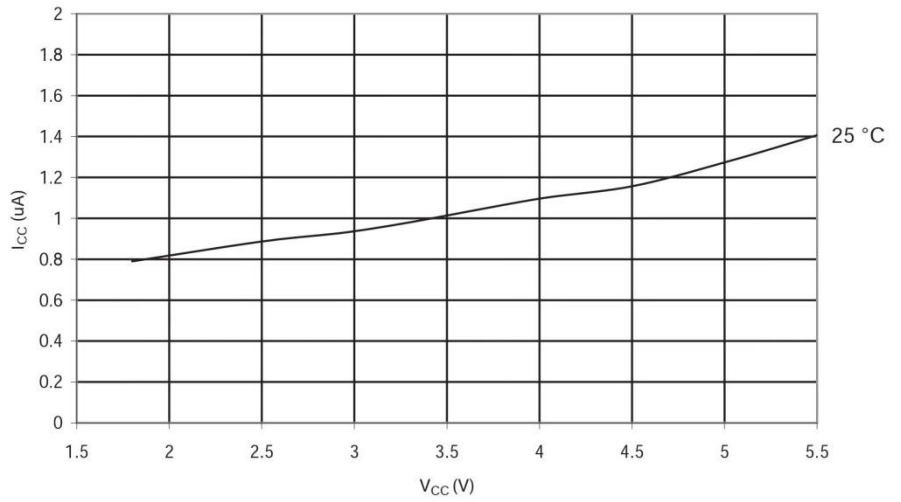


Figure 29-150. ATmega328P: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



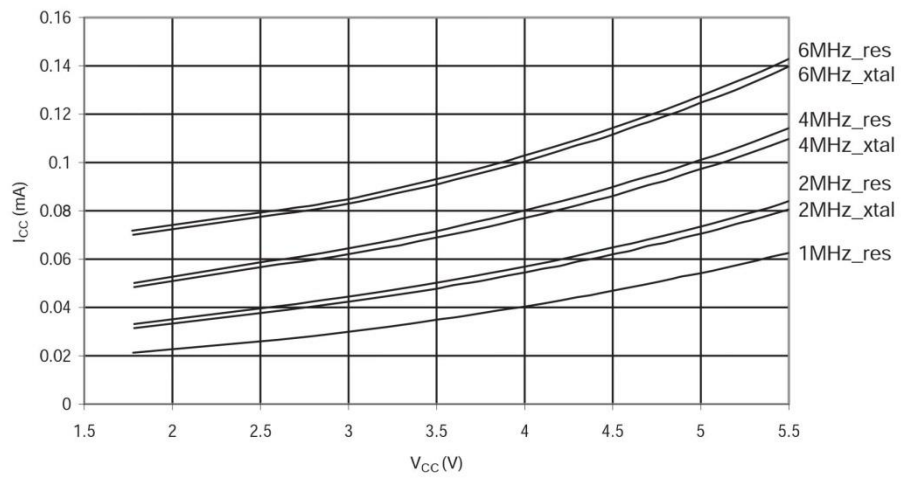
29.4.5 Power-save Supply Current

Figure 29-151. ATmega328P: Power-Save Supply Current vs. V_{CC} (Watchdog Timer Disabled and 32 kHz Crystal Oscillator Running)



29.4.6 Standby Supply Current

Figure 29-152. ATmega328P: Standby Supply Current vs. V_{CC} (Watchdog Timer Disabled)



29.4.7 Pin Pull-Up

Figure 29-153. ATmega328P: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 1.8\text{ V}$)

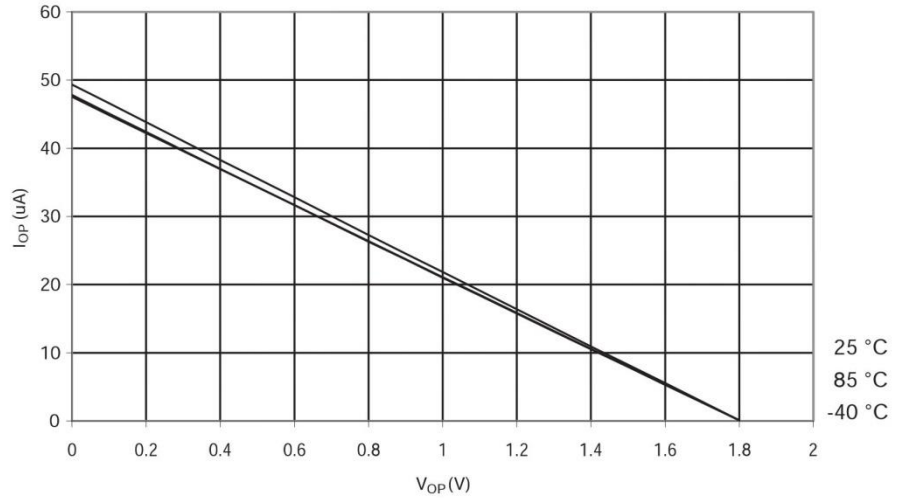


Figure 29-154. ATmega328P: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 2.7\text{ V}$)

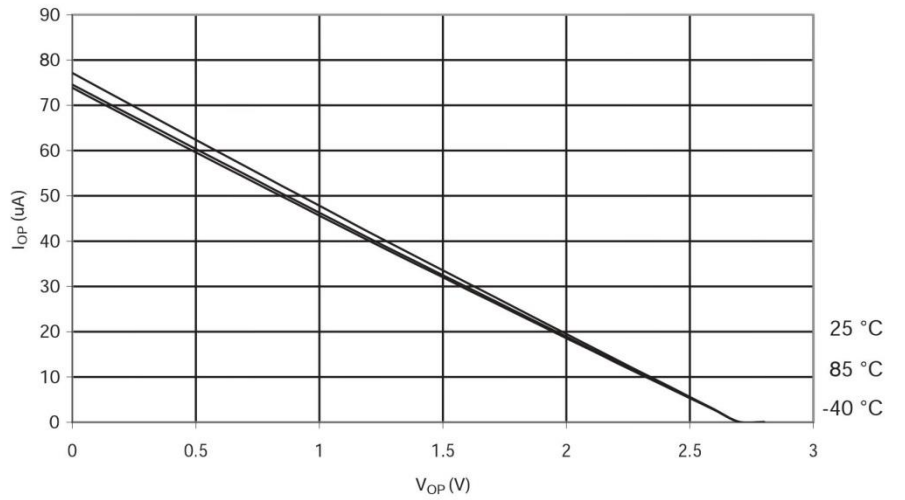


Figure 29-155. ATmega328P: I/O Pin Pull-up Resistor Current vs. Input Voltage ($V_{CC} = 5\text{ V}$)

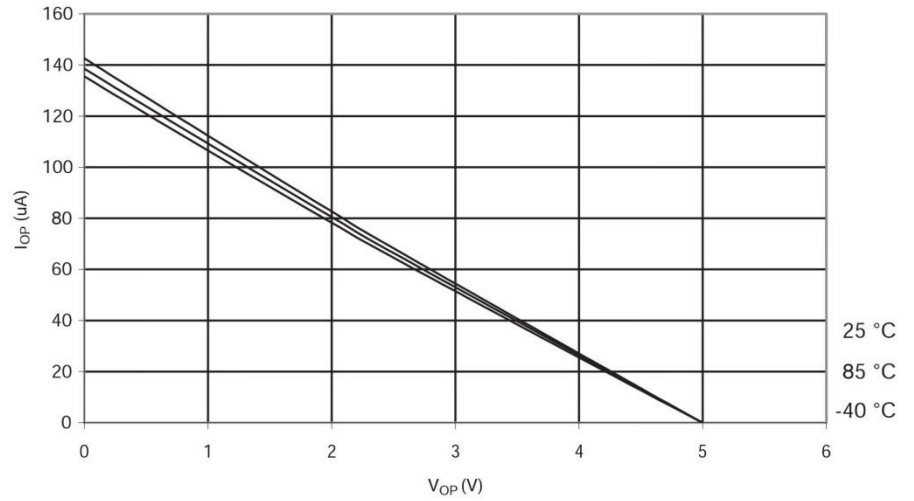


Figure 29-156. ATmega328P: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 1.8\text{ V}$)

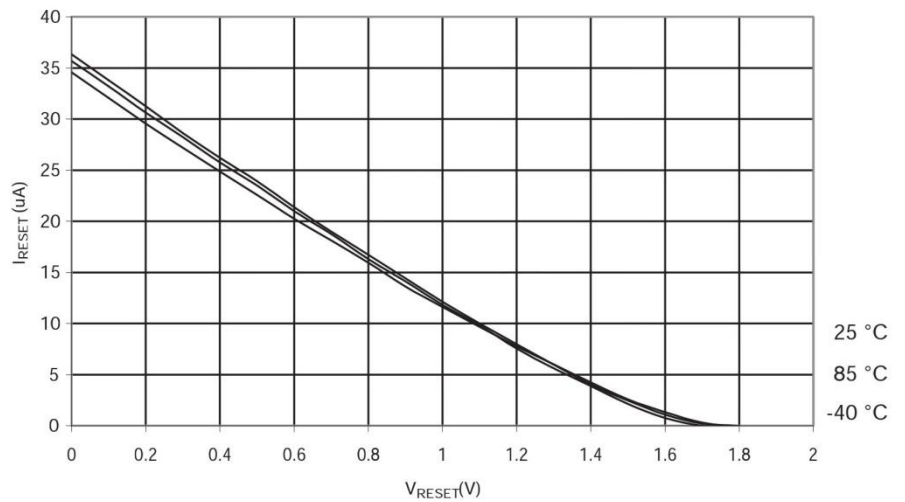


Figure 29-157. ATmega328P: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 2.7\text{ V}$)

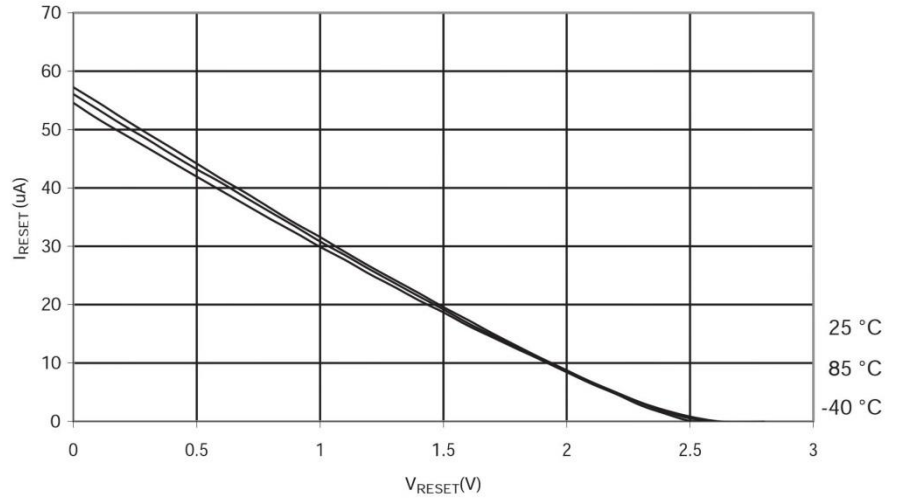
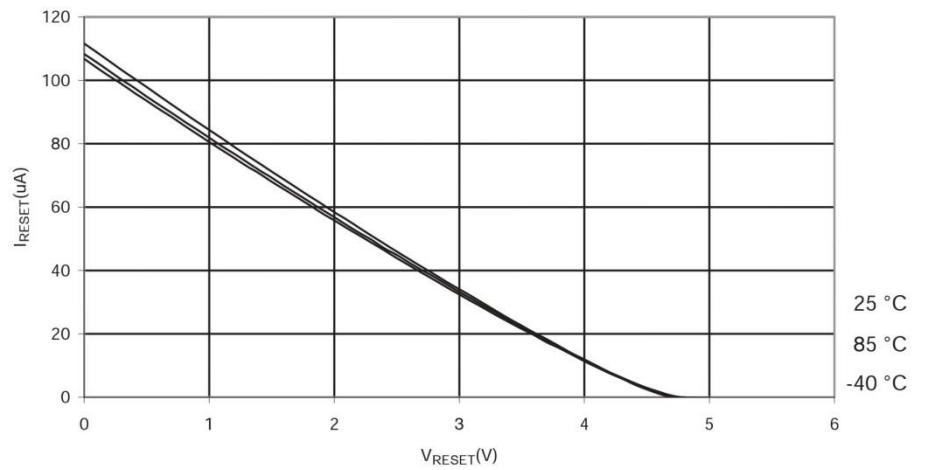


Figure 29-158. ATmega328P: Reset Pull-up Resistor Current vs. Reset Pin Voltage ($V_{CC} = 5\text{ V}$)



29.4.8 Pin Driver Strength

Figure 29-159. ATmega328P: I/O Pin Output Voltage vs. Sink Current ($V_{CC} = 3\text{ V}$)

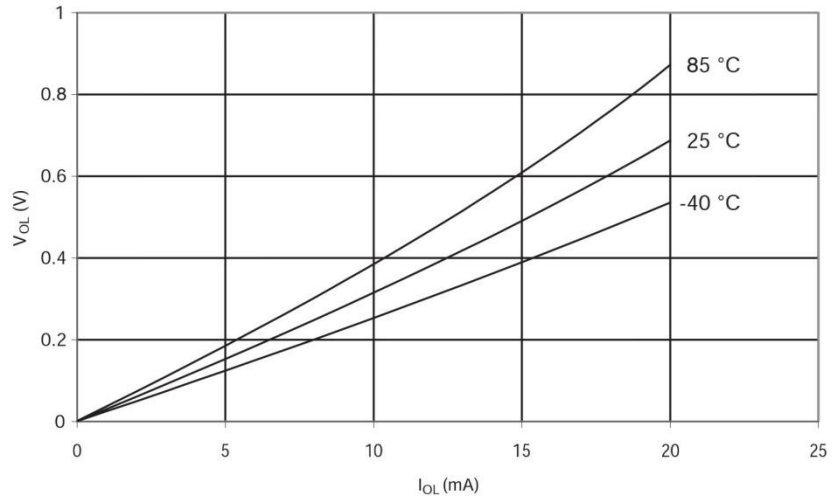


Figure 29-160. ATmega328P: I/O Pin Output Voltage vs. Sink Current ($V_{CC} = 5\text{ V}$)

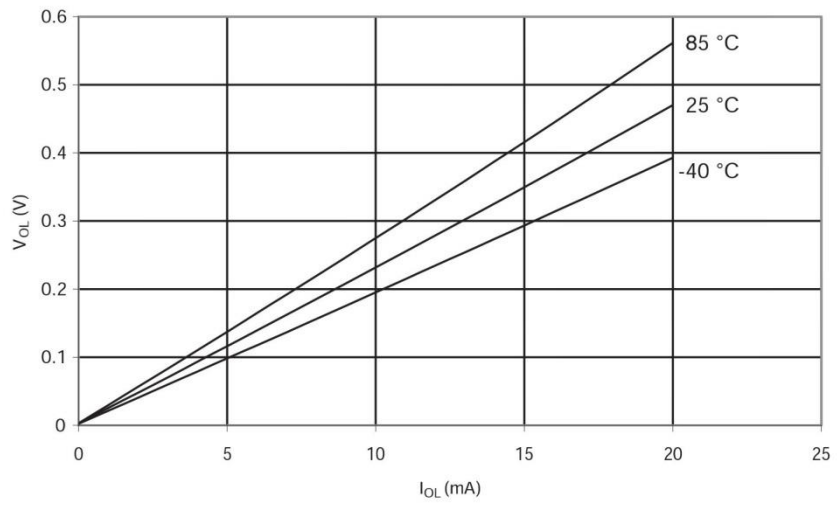


Figure 29-161. ATmega328P: I/O Pin Output Voltage vs. Source Current ($V_{CC} = 3\text{ V}$)

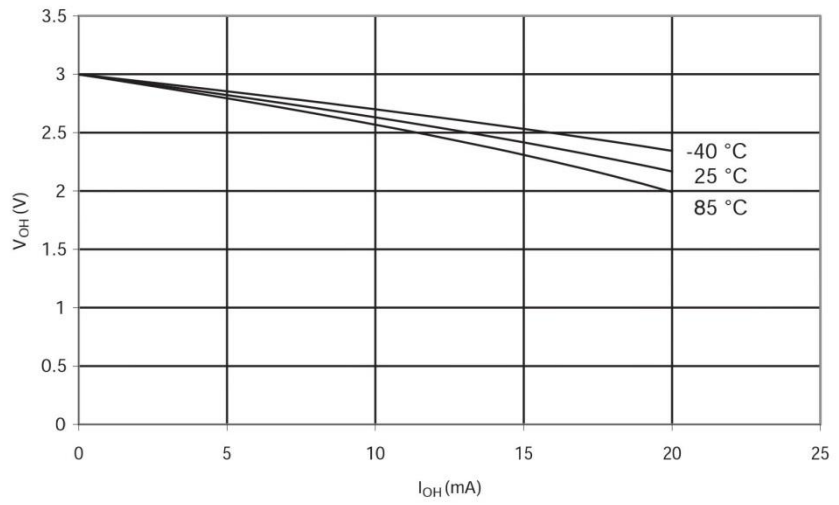
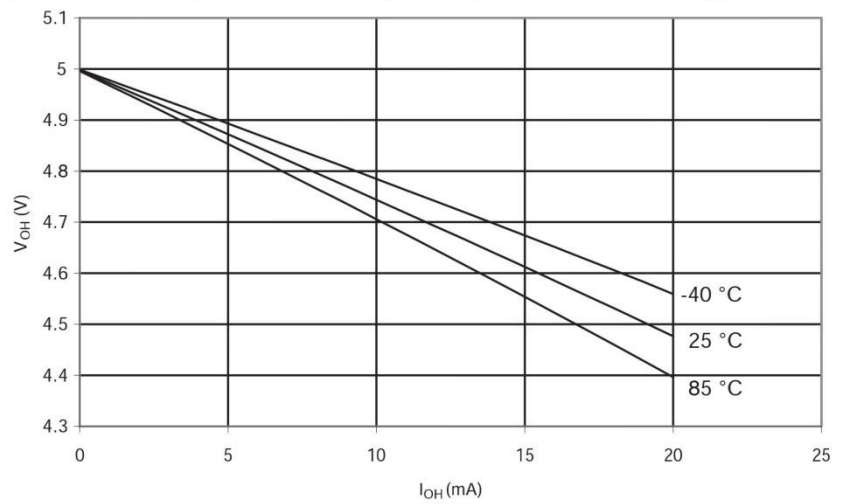


Figure 29-162. ATmega328P: I/O Pin Output Voltage vs. Source Current ($V_{CC} = 5\text{ V}$)



29.4.9 Pin Threshold and Hysteresis

Figure 29-163. ATmega328P: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH} , I/O Pin read as '1')

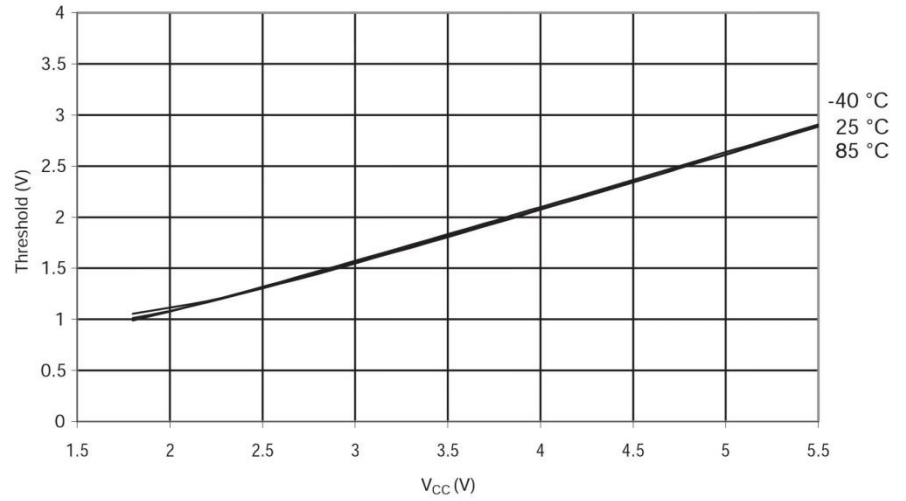


Figure 29-164. ATmega328P: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL} , I/O Pin read as '0')

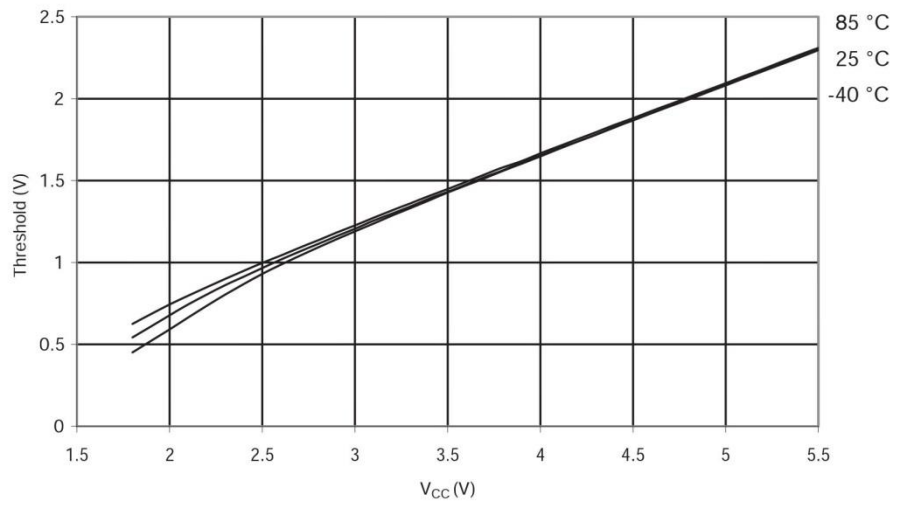


Figure 29-165. ATmega328P: I/O Pin Input Hysteresis vs. V_{CC}

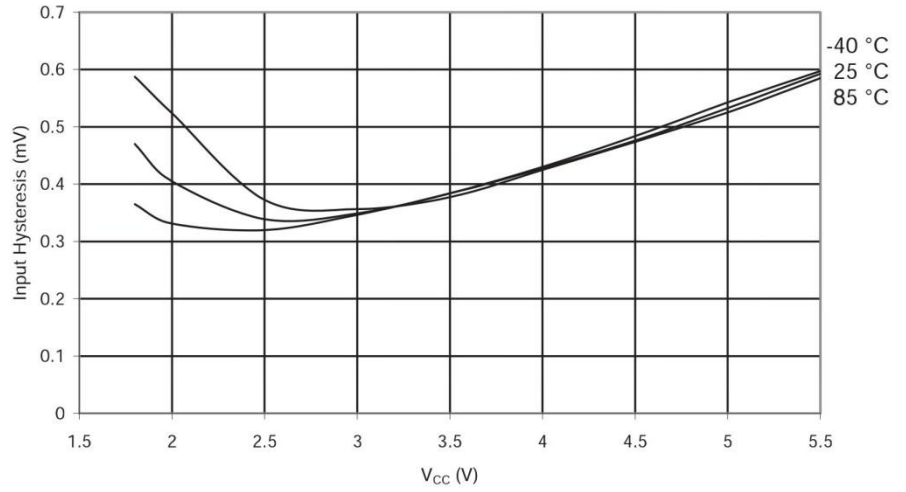


Figure 29-166. ATmega328P: Reset Input Threshold Voltage vs. V_{CC} ($V_{I_{Hr}}$, I/O Pin read as '1')

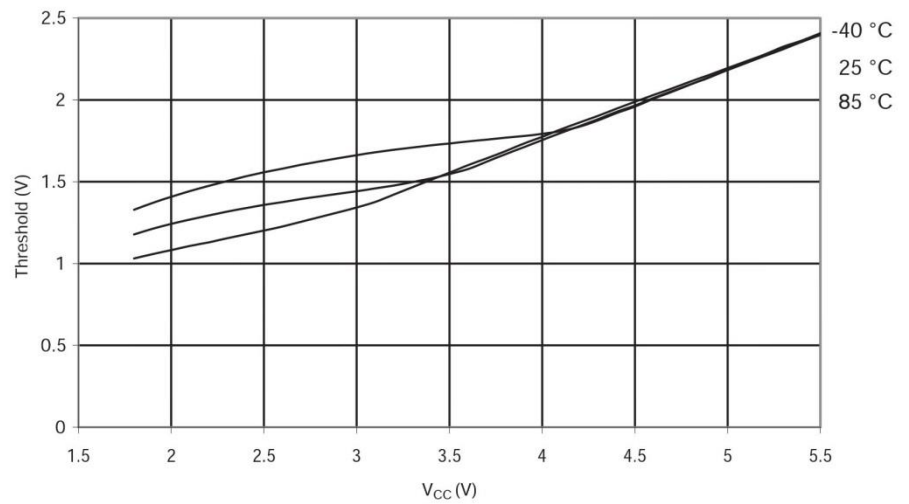


Figure 29-167. ATmega328P: Reset Input Threshold Voltage vs. V_{CC} (V_{IL} , I/O Pin read as '0')

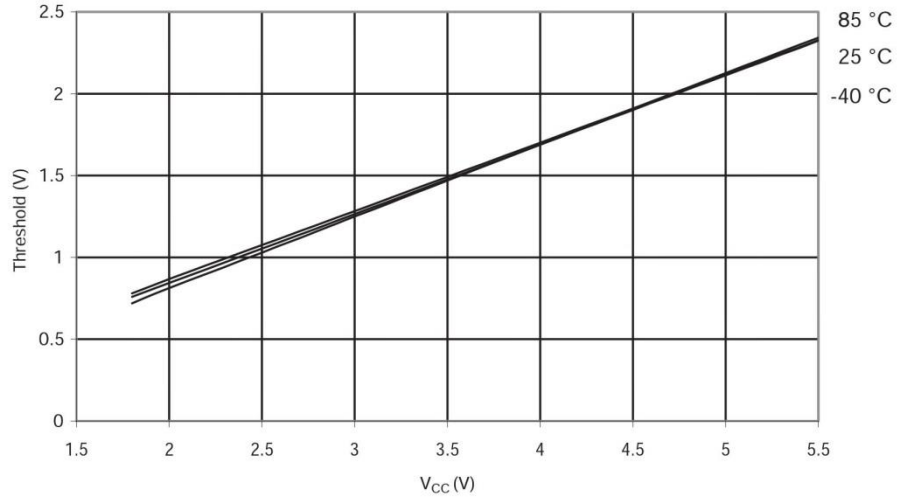
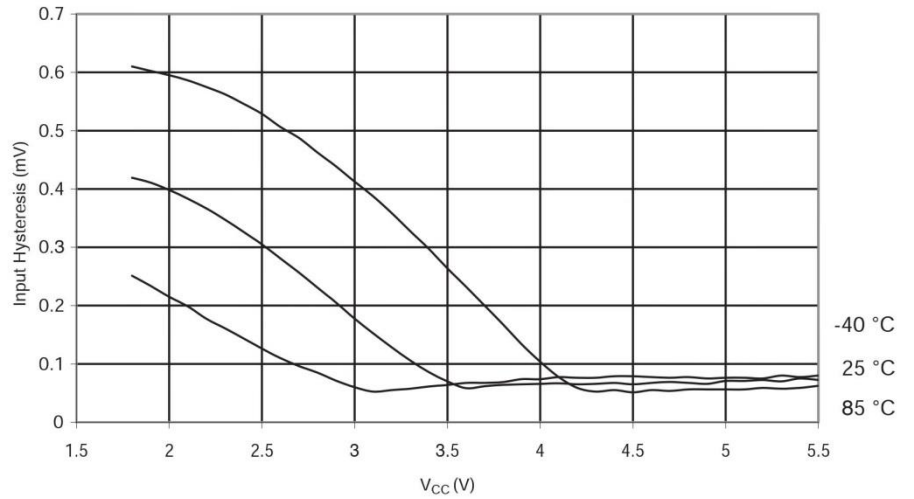


Figure 29-168. ATmega328P: Reset Pin Input Hysteresis vs. V_{CC}



29.4.10 BOD Threshold

Figure 29-169. ATmega328P: BOD Thresholds vs. Temperature (BODLEVEL is 1.8 V)

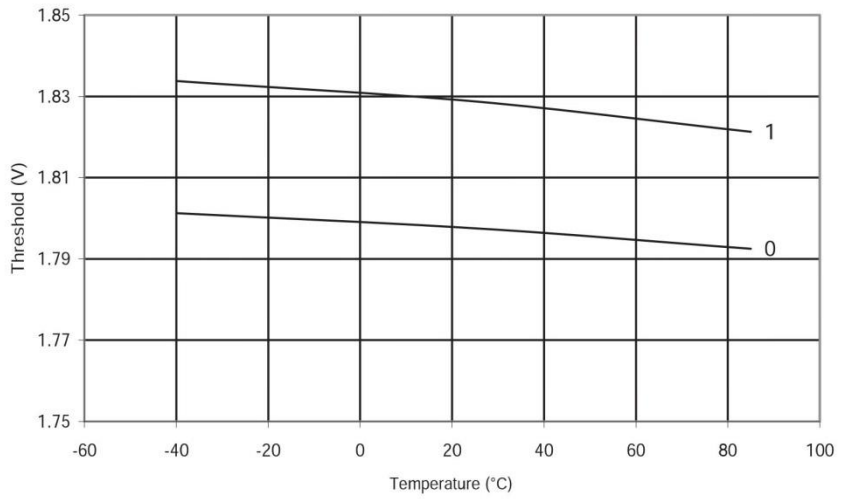


Figure 29-170. ATmega328P: BOD Thresholds vs. Temperature (BODLEVEL is 2.7 V)

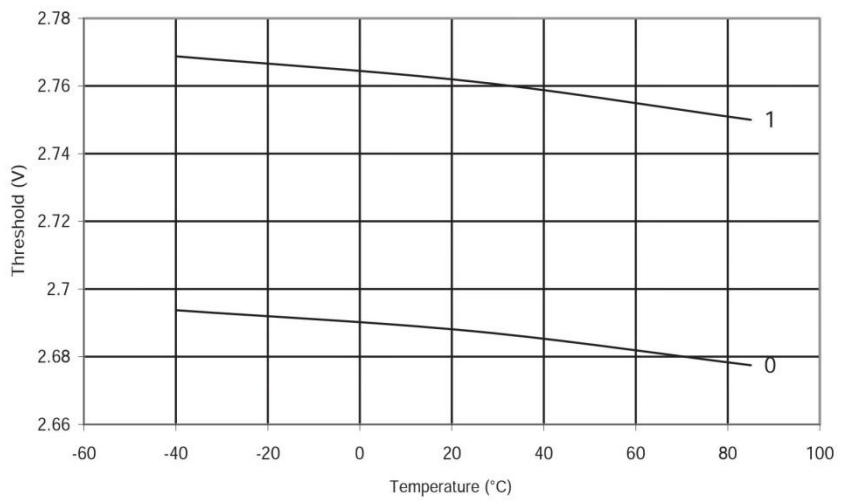
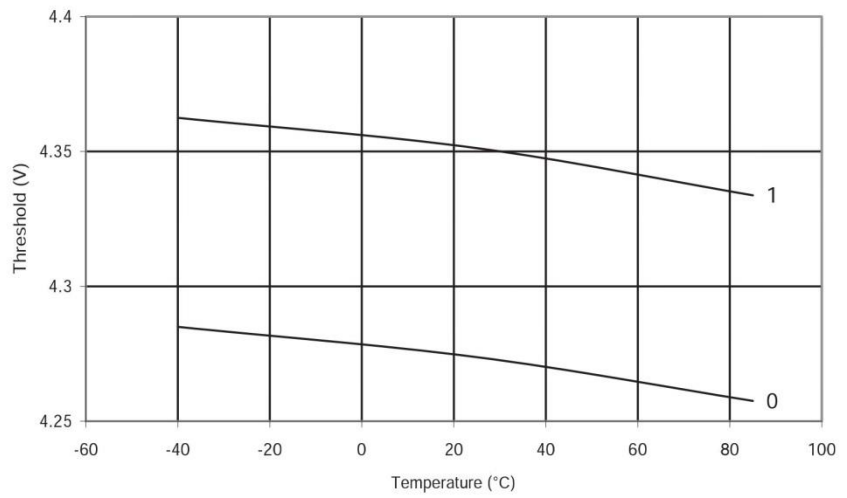


Figure 29-171. ATmega328P: BOD Thresholds vs. Temperature (BODLEVEL is 4.3 V)



29.4.11 Internal Oscillator Speed

Figure 29-172. ATmega328P: Watchdog Oscillator Frequency vs. Temperature

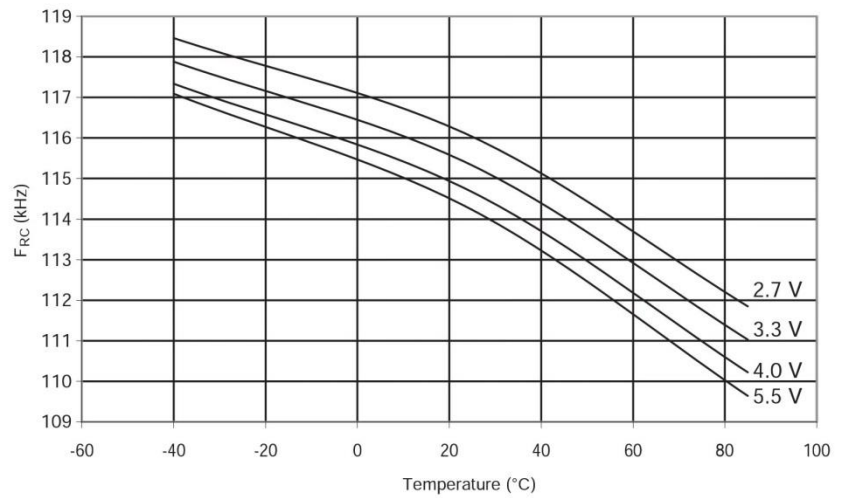


Figure 29-173. ATmega328P: Watchdog Oscillator Frequency vs. V_{CC}

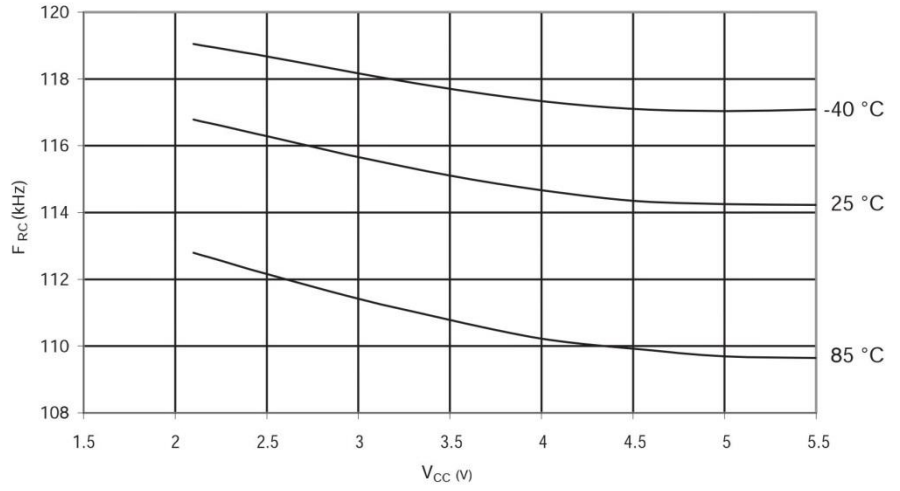


Figure 29-174. ATmega328P: Calibrated 8 MHz RC Oscillator Frequency vs. V_{CC}

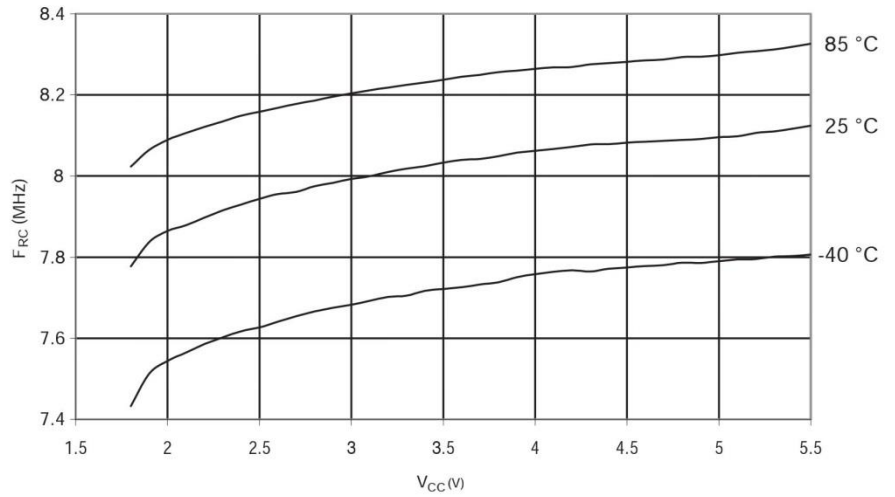


Figure 29-175. ATmega328P: Calibrated 8 MHz RC Oscillator Frequency vs. Temperature

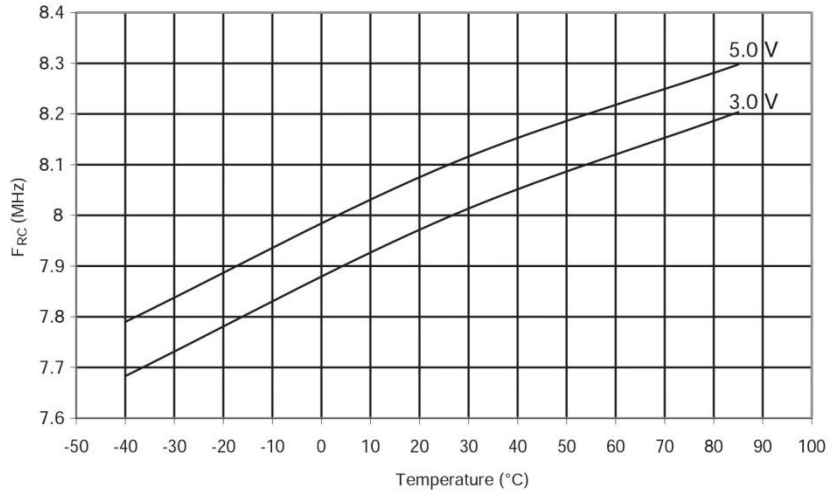
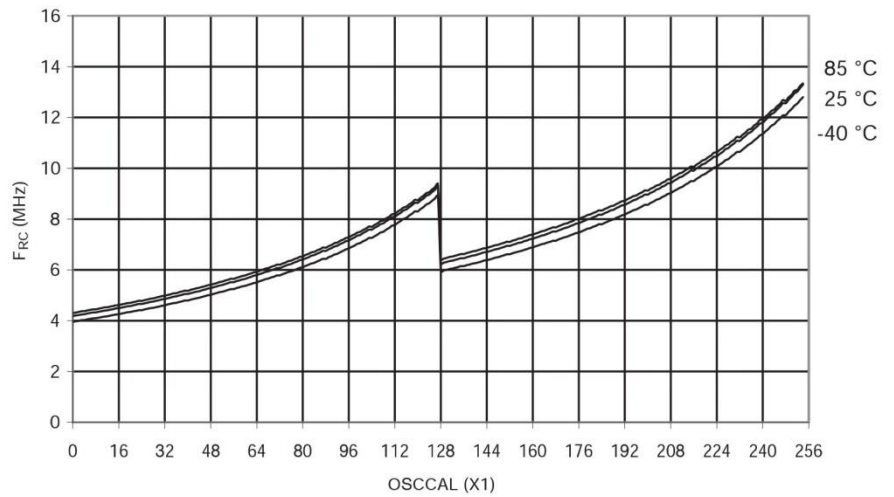


Figure 29-176. ATmega328P: Calibrated 8 MHz RC Oscillator Frequency vs. OSCCAL Value



29.4.12 Current Consumption of Peripheral Units

Figure 29-177. ATmega328P: ADC Current vs. V_{CC} (AREF = AV_{CC})

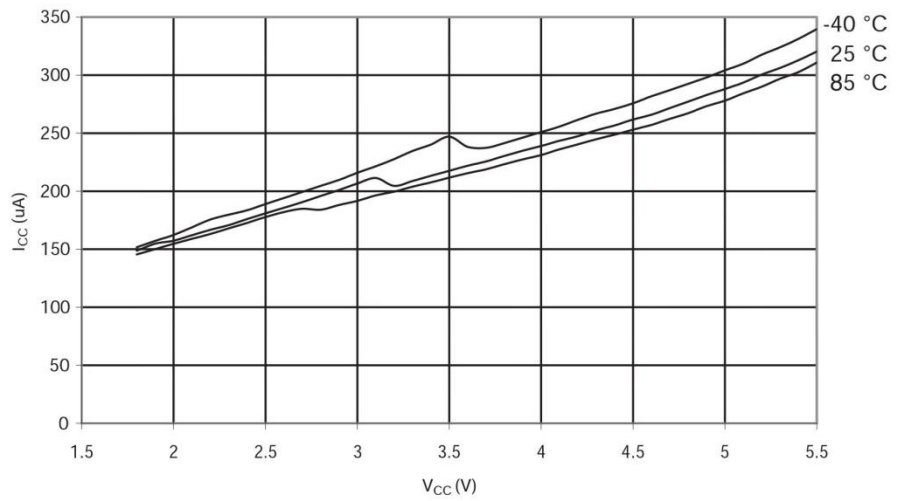


Figure 29-178. ATmega328P: Analog Comparator Current vs. V_{CC}

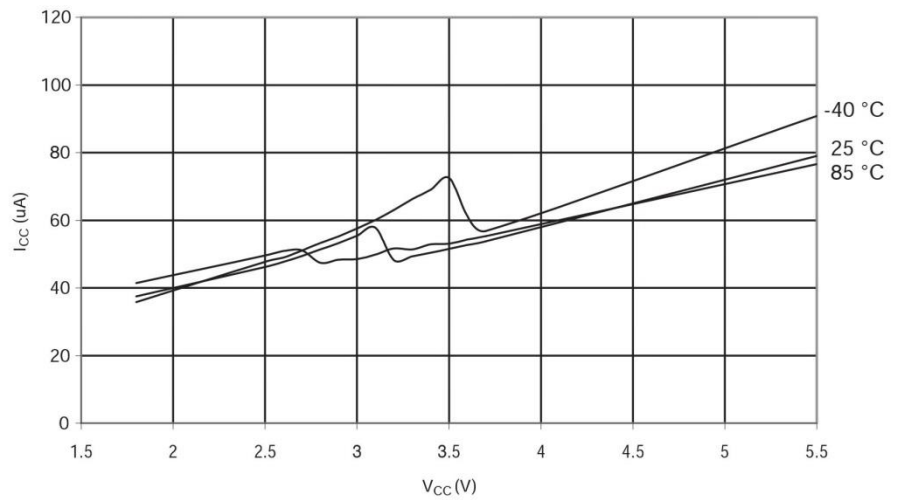


Figure 29-179. ATmega328P: AREF External Reference Current vs. V_{CC}

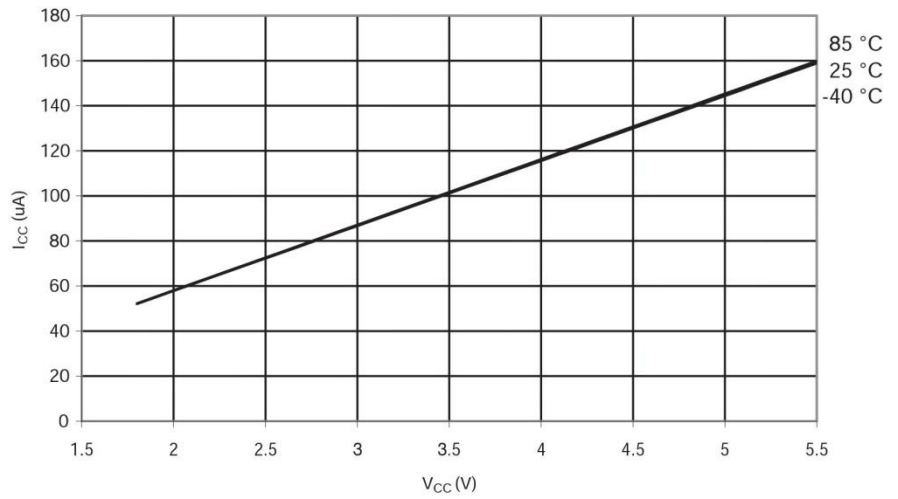


Figure 29-180. ATmega328P: Brownout Detector Current vs. V_{CC}

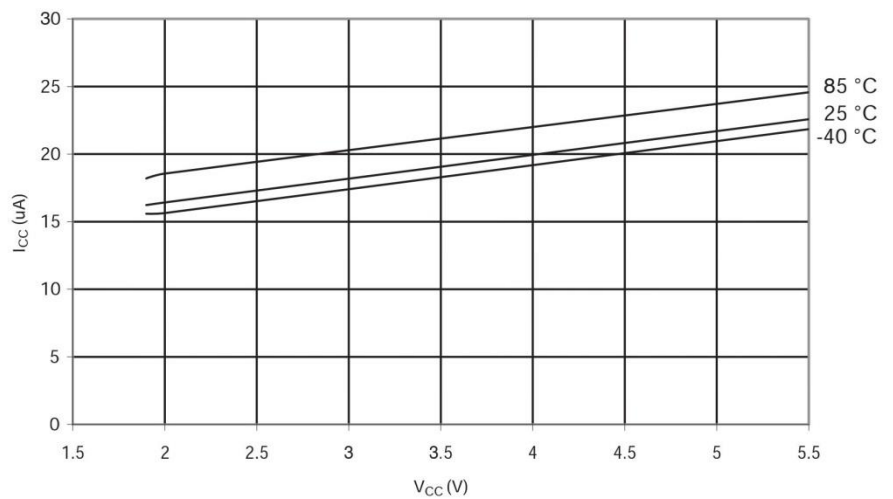
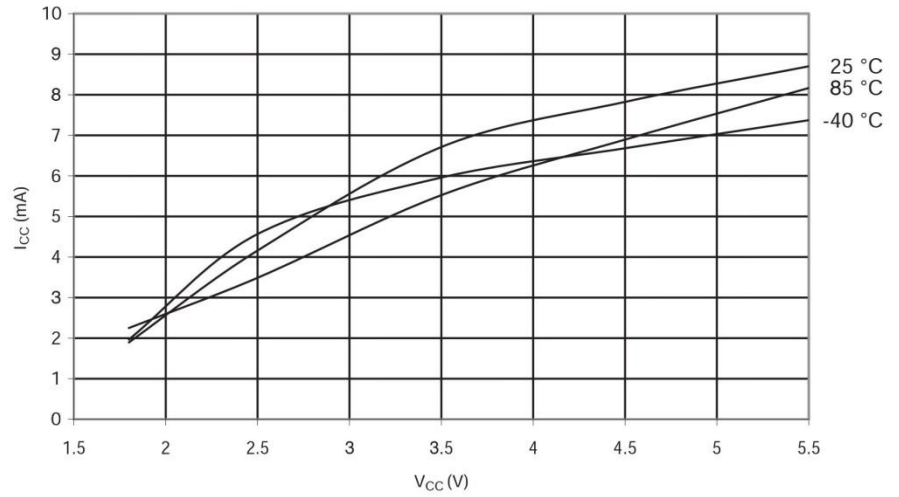


Figure 29-181. ATmega328P: Programming Current vs. V_{CC}



29.4.13 Current Consumption in Reset and Reset Pulsewidth

Figure 29-182. ATmega328P: Reset Supply Current vs. Low Frequency (0.1 - 1.0 MHz)

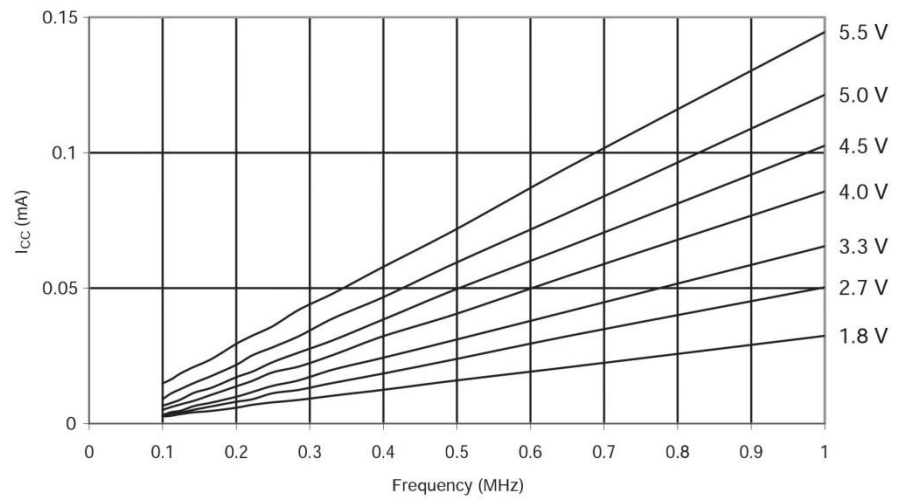


Figure 29-183. ATmega328P: Reset Supply Current vs. Frequency (1 - 20 MHz)

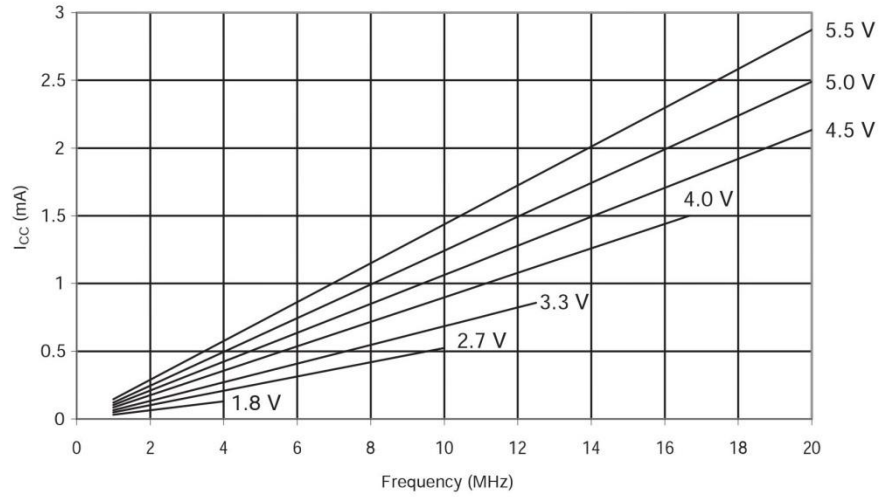
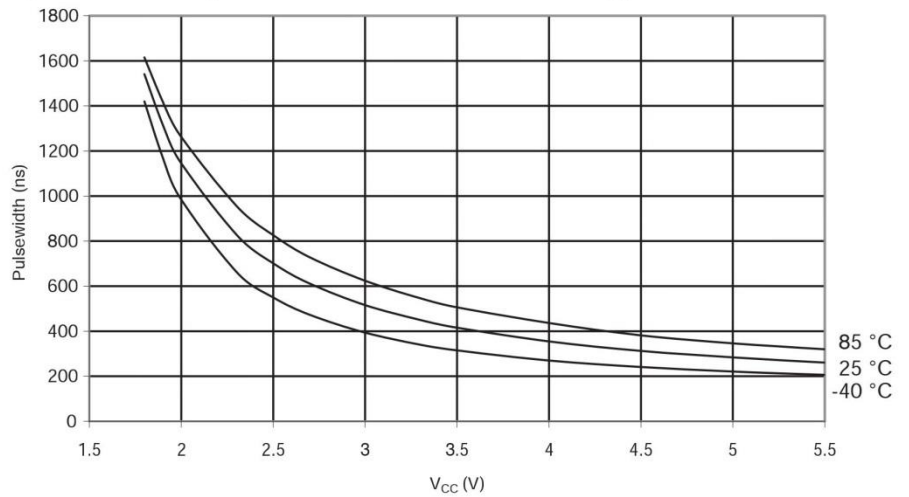


Figure 29-184. ATmega328P: Minimum Reset Pulse width vs. V_{CC}



30. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xFF)	Reserved	-	-	-	-	-	-	-	-	
(0xFE)	Reserved	-	-	-	-	-	-	-	-	
(0xFD)	Reserved	-	-	-	-	-	-	-	-	
(0xFC)	Reserved	-	-	-	-	-	-	-	-	
(0xFB)	Reserved	-	-	-	-	-	-	-	-	
(0xFA)	Reserved	-	-	-	-	-	-	-	-	
(0xF9)	Reserved	-	-	-	-	-	-	-	-	
(0xF8)	Reserved	-	-	-	-	-	-	-	-	
(0xF7)	Reserved	-	-	-	-	-	-	-	-	
(0xF6)	Reserved	-	-	-	-	-	-	-	-	
(0xF5)	Reserved	-	-	-	-	-	-	-	-	
(0xF4)	Reserved	-	-	-	-	-	-	-	-	
(0xF3)	Reserved	-	-	-	-	-	-	-	-	
(0xF2)	Reserved	-	-	-	-	-	-	-	-	
(0xF1)	Reserved	-	-	-	-	-	-	-	-	
(0xF0)	Reserved	-	-	-	-	-	-	-	-	
(0xEF)	Reserved	-	-	-	-	-	-	-	-	
(0xEE)	Reserved	-	-	-	-	-	-	-	-	
(0xED)	Reserved	-	-	-	-	-	-	-	-	
(0xEC)	Reserved	-	-	-	-	-	-	-	-	
(0xEB)	Reserved	-	-	-	-	-	-	-	-	
(0xEA)	Reserved	-	-	-	-	-	-	-	-	
(0xE9)	Reserved	-	-	-	-	-	-	-	-	
(0xE8)	Reserved	-	-	-	-	-	-	-	-	
(0xE7)	Reserved	-	-	-	-	-	-	-	-	
(0xE6)	Reserved	-	-	-	-	-	-	-	-	
(0xE5)	Reserved	-	-	-	-	-	-	-	-	
(0xE4)	Reserved	-	-	-	-	-	-	-	-	
(0xE3)	Reserved	-	-	-	-	-	-	-	-	
(0xE2)	Reserved	-	-	-	-	-	-	-	-	
(0xE1)	Reserved	-	-	-	-	-	-	-	-	
(0xE0)	Reserved	-	-	-	-	-	-	-	-	
(0xDF)	Reserved	-	-	-	-	-	-	-	-	
(0xDE)	Reserved	-	-	-	-	-	-	-	-	
(0xDD)	Reserved	-	-	-	-	-	-	-	-	
(0xDC)	Reserved	-	-	-	-	-	-	-	-	
(0xDB)	Reserved	-	-	-	-	-	-	-	-	
(0xDA)	Reserved	-	-	-	-	-	-	-	-	
(0xD9)	Reserved	-	-	-	-	-	-	-	-	
(0xD8)	Reserved	-	-	-	-	-	-	-	-	
(0xD7)	Reserved	-	-	-	-	-	-	-	-	
(0xD6)	Reserved	-	-	-	-	-	-	-	-	
(0xD5)	Reserved	-	-	-	-	-	-	-	-	
(0xD4)	Reserved	-	-	-	-	-	-	-	-	
(0xD3)	Reserved	-	-	-	-	-	-	-	-	
(0xD2)	Reserved	-	-	-	-	-	-	-	-	
(0xD1)	Reserved	-	-	-	-	-	-	-	-	
(0xD0)	Reserved	-	-	-	-	-	-	-	-	
(0xCF)	Reserved	-	-	-	-	-	-	-	-	
(0xCE)	Reserved	-	-	-	-	-	-	-	-	
(0xCD)	Reserved	-	-	-	-	-	-	-	-	
(0xCC)	Reserved	-	-	-	-	-	-	-	-	
(0xCB)	Reserved	-	-	-	-	-	-	-	-	
(0xCA)	Reserved	-	-	-	-	-	-	-	-	
(0xC9)	Reserved	-	-	-	-	-	-	-	-	
(0xC8)	Reserved	-	-	-	-	-	-	-	-	
(0xC7)	Reserved	-	-	-	-	-	-	-	-	
(0xC6)	UDR0	USART I/O Data Register								195
(0xC5)	UBRR0H	USART Baud Rate Register High								199
(0xC4)	UBRR0L	USART Baud Rate Register Low								199
(0xC3)	Reserved	-	-	-	-	-	-	-	-	
(0xC2)	UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01/UDOR00	UCSZ00/UCPHA0	UCPOL0	197/212
(0xC1)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	196
(0xC0)	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	195

ATmega48PA/88PA/168PA/328P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page	
(0xBF)	Reserved	–	–	–	–	–	–	–	–		
(0xBE)	Reserved	–	–	–	–	–	–	–	–		
(0xBD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	–	244	
(0xBC)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	241	
(0xBB)	TWDR	2-wire Serial Interface Data Register									243
(0xBA)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	244	
(0xB9)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	243	
(0xB8)	TWBR	2-wire Serial Interface Bit Rate Register									241
(0xB7)	Reserved	–	–	–	–	–	–	–	–		
(0xB6)	ASSR	–	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	164	
(0xB5)	Reserved	–	–	–	–	–	–	–	–		
(0xB4)	OCR2B	Timer/Counter2 Output Compare Register B									162
(0xB3)	OCR2A	Timer/Counter2 Output Compare Register A									162
(0xB2)	TCNT2	Timer/Counter2 (8-bit)									162
(0xB1)	TCCR2B	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	161	
(0xB0)	TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	158	
(0xAF)	Reserved	–	–	–	–	–	–	–	–		
(0xAE)	Reserved	–	–	–	–	–	–	–	–		
(0xAD)	Reserved	–	–	–	–	–	–	–	–		
(0xAC)	Reserved	–	–	–	–	–	–	–	–		
(0xAB)	Reserved	–	–	–	–	–	–	–	–		
(0xAA)	Reserved	–	–	–	–	–	–	–	–		
(0xA9)	Reserved	–	–	–	–	–	–	–	–		
(0xA8)	Reserved	–	–	–	–	–	–	–	–		
(0xA7)	Reserved	–	–	–	–	–	–	–	–		
(0xA6)	Reserved	–	–	–	–	–	–	–	–		
(0xA5)	Reserved	–	–	–	–	–	–	–	–		
(0xA4)	Reserved	–	–	–	–	–	–	–	–		
(0xA3)	Reserved	–	–	–	–	–	–	–	–		
(0xA2)	Reserved	–	–	–	–	–	–	–	–		
(0xA1)	Reserved	–	–	–	–	–	–	–	–		
(0xA0)	Reserved	–	–	–	–	–	–	–	–		
(0x9F)	Reserved	–	–	–	–	–	–	–	–		
(0x9E)	Reserved	–	–	–	–	–	–	–	–		
(0x9D)	Reserved	–	–	–	–	–	–	–	–		
(0x9C)	Reserved	–	–	–	–	–	–	–	–		
(0x9B)	Reserved	–	–	–	–	–	–	–	–		
(0x9A)	Reserved	–	–	–	–	–	–	–	–		
(0x99)	Reserved	–	–	–	–	–	–	–	–		
(0x98)	Reserved	–	–	–	–	–	–	–	–		
(0x97)	Reserved	–	–	–	–	–	–	–	–		
(0x96)	Reserved	–	–	–	–	–	–	–	–		
(0x95)	Reserved	–	–	–	–	–	–	–	–		
(0x94)	Reserved	–	–	–	–	–	–	–	–		
(0x93)	Reserved	–	–	–	–	–	–	–	–		
(0x92)	Reserved	–	–	–	–	–	–	–	–		
(0x91)	Reserved	–	–	–	–	–	–	–	–		
(0x90)	Reserved	–	–	–	–	–	–	–	–		
(0x8F)	Reserved	–	–	–	–	–	–	–	–		
(0x8E)	Reserved	–	–	–	–	–	–	–	–		
(0x8D)	Reserved	–	–	–	–	–	–	–	–		
(0x8C)	Reserved	–	–	–	–	–	–	–	–		
(0x8B)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte									138
(0x8A)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte									138
(0x89)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte									138
(0x88)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte									138
(0x87)	ICR1H	Timer/Counter1 - Input Capture Register High Byte									138
(0x86)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte									138
(0x85)	TCNT1H	Timer/Counter1 - Counter Register High Byte									138
(0x84)	TCNT1L	Timer/Counter1 - Counter Register Low Byte									138
(0x83)	Reserved	–	–	–	–	–	–	–	–		
(0x82)	TCCR1C	FOC1A	FOC1B	–	–	–	–	–	–	137	
(0x81)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	136	
(0x80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	134	
(0x7F)	DIDR1	–	–	–	–	–	–	AIN1D	AIN0D	249	
(0x7E)	DIDR0	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	266	

ATmega48PA/88PA/168PA/328P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x7D)	Reserved	–	–	–	–	–	–	–	–	
(0x7C)	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	262
(0x7B)	ADCSRB	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	265
(0x7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	263
(0x79)	ADCH	ADC Data Register High byte								265
(0x78)	ADCL	ADC Data Register Low byte								265
(0x77)	Reserved	–	–	–	–	–	–	–	–	
(0x76)	Reserved	–	–	–	–	–	–	–	–	
(0x75)	Reserved	–	–	–	–	–	–	–	–	
(0x74)	Reserved	–	–	–	–	–	–	–	–	
(0x73)	Reserved	–	–	–	–	–	–	–	–	
(0x72)	Reserved	–	–	–	–	–	–	–	–	
(0x71)	Reserved	–	–	–	–	–	–	–	–	
(0x70)	TIMSK2	–	–	–	–	–	OCIE2B	OCIE2A	TOIE2	163
(0x6F)	TIMSK1	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	139
(0x6E)	TIMSK0	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	111
(0x6D)	PCMSK2	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	74
(0x6C)	PCMSK1	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	74
(0x6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	74
(0x6A)	Reserved	–	–	–	–	–	–	–	–	
(0x69)	EICRA	–	–	–	–	ISC11	ISC10	ISC01	ISC00	71
(0x68)	PCICR	–	–	–	–	–	PCIE2	PCIE1	PCIE0	
(0x67)	Reserved	–	–	–	–	–	–	–	–	
(0x66)	OSCCAL	Oscillator Calibration Register								37
(0x65)	Reserved	–	–	–	–	–	–	–	–	
(0x64)	PRR	PRTWI	PRTIM2	PRTIM0	–	PRTIM1	PRSPI	PRUSART0	PRADC	42
(0x63)	Reserved	–	–	–	–	–	–	–	–	
(0x62)	Reserved	–	–	–	–	–	–	–	–	
(0x61)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	37
(0x60)	WDTCR	WDIF	WDIE	WDP3	WDCE	WDFE	WDP2	WDP1	WDP0	54
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	9
0x3E (0x5E)	SPH	–	–	–	–	–	(SP10) ⁵	SP9	SP8	12
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
0x3C (0x5C)	Reserved	–	–	–	–	–	–	–	–	
0x3B (0x5B)	Reserved	–	–	–	–	–	–	–	–	
0x3A (0x5A)	Reserved	–	–	–	–	–	–	–	–	
0x39 (0x59)	Reserved	–	–	–	–	–	–	–	–	
0x38 (0x58)	Reserved	–	–	–	–	–	–	–	–	
0x37 (0x57)	SPMCSR	SPMIE	(RWWSB) ⁵	–	(RWWRE) ⁵	BLBSET	PGWRT	PGERS	SELFPRGEN	292
0x36 (0x56)	Reserved	–	–	–	–	–	–	–	–	
0x35 (0x55)	MCUCR	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	44/68/92
0x34 (0x54)	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	54
0x33 (0x53)	SMCR	–	–	–	–	SM2	SM1	SM0	SE	40
0x32 (0x52)	Reserved	–	–	–	–	–	–	–	–	
0x31 (0x51)	Reserved	–	–	–	–	–	–	–	–	
0x30 (0x50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	247
0x2F (0x4F)	Reserved	–	–	–	–	–	–	–	–	
0x2E (0x4E)	SPDR	SPI Data Register								175
0x2D (0x4D)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	174
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	173
0x2B (0x4B)	GPOR2	General Purpose I/O Register 2								25
0x2A (0x4A)	GPOR1	General Purpose I/O Register 1								25
0x29 (0x49)	Reserved	–	–	–	–	–	–	–	–	
0x28 (0x48)	OCR0B	Timer/Counter0 Output Compare Register B								
0x27 (0x47)	OCR0A	Timer/Counter0 Output Compare Register A								
0x26 (0x46)	TCNT0	Timer/Counter0 (8-bit)								
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	
0x23 (0x43)	GTCCR	TSM	–	–	–	–	–	PSRASY	PSRSYNC	143/165
0x22 (0x42)	EEARH	(EEPROM Address Register High Byte) ⁵								21
0x21 (0x41)	EEARL	EEPROM Address Register Low Byte								21
0x20 (0x40)	EEDR	EEPROM Data Register								21
0x1F (0x3F)	EECR	–	–	EEDR1	EEDR0	EERIE	EEMPE	EEPE	EERE	21
0x1E (0x3E)	GPOR0	General Purpose I/O Register 0								25
0x1D (0x3D)	EIMSK	–	–	–	–	–	–	INT1	INT0	72
0x1C (0x3C)	EIFR	–	–	–	–	–	–	INTF1	INTF0	72

ATmega48PA/88PA/168PA/328P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x1B (0x3B)	PCIFR	–	–	–	–	–	PCIF2	PCIF1	PCIF0	
0x1A (0x3A)	Reserved	–	–	–	–	–	–	–	–	
0x19 (0x39)	Reserved	–	–	–	–	–	–	–	–	
0x18 (0x38)	Reserved	–	–	–	–	–	–	–	–	
0x17 (0x37)	TIFR2	–	–	–	–	–	OCF2B	OCF2A	TOV2	163
0x16 (0x36)	TIFR1	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	139
0x15 (0x35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0	
0x14 (0x34)	Reserved	–	–	–	–	–	–	–	–	
0x13 (0x33)	Reserved	–	–	–	–	–	–	–	–	
0x12 (0x32)	Reserved	–	–	–	–	–	–	–	–	
0x11 (0x31)	Reserved	–	–	–	–	–	–	–	–	
0x10 (0x30)	Reserved	–	–	–	–	–	–	–	–	
0x0F (0x2F)	Reserved	–	–	–	–	–	–	–	–	
0x0E (0x2E)	Reserved	–	–	–	–	–	–	–	–	
0x0D (0x2D)	Reserved	–	–	–	–	–	–	–	–	
0x0C (0x2C)	Reserved	–	–	–	–	–	–	–	–	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	93
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	93
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	93
0x08 (0x28)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	92
0x07 (0x27)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	92
0x06 (0x26)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	92
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	92
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	92
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	92
0x02 (0x22)	Reserved	–	–	–	–	–	–	–	–	
0x01 (0x21)	Reserved	–	–	–	–	–	–	–	–	
0x00 (0x20)	Reserved	–	–	–	–	–	–	–	–	

- Note:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 2. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
 3. Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
 4. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48PA/88PA/168PA/328P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.
 5. Only valid for ATmega88PA/168PA.

31. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	RdI,K	Add Immediate to Word	$RdH:RdL \leftarrow RdH:RdL + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	RdI,K	Subtract Immediate from Word	$RdH:RdL \leftarrow RdH:RdL - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \wedge Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \wedge K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \wedge (\sim K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \wedge Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
JMP ⁽¹⁾	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL ⁽¹⁾	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if $(Rd = Rr) PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd,Rr	Compare	$Rd - Rr$	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if $(Rr(b)=0) PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRSC	Rr, b	Skip if Bit in Register is Set	if $(Rr(b)=1) PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if $(P(b)=0) PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if $(P(b)=1) PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if $(SREG(s) = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if $(SREG(s) = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BREQ	k	Branch if Equal	if $(Z = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	if $(Z = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	if $(N = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	if $(N = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if $(H = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if $(H = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if $(T = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if $(T = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if $(V = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if $(V = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2

ATmega48PA/88PA/168PA/328P

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, -X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z + 1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z + 1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2

ATmega48PA/88PA/168PA/328P

Mnemonics	Operands	Description	Operation	Flags	#Clocks
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

Note: 1. These instructions are only available in ATmega168PA and ATmega328P.

32. Ordering Information

32.1 ATmega48PA

Speed (MHz)	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20 ⁽³⁾	1.8 - 5.5	ATmega48PA-AU ATmega48PA-MMH ⁽⁴⁾ ATmega48PA-MU ATmega48PA-PU	32A 28M1 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See "[Speed Grades](#)" on page 316.
 4. NiPdAu Lead Finish.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

ATmega48PA/88PA/168PA/328P

32.2 ATmega88PA

Speed (MHz)	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20 ⁽³⁾	1.8 - 5.5	ATmega88PA-AU ATmega88PA-MMH ⁽⁴⁾ ATmega88PA-MU ATmega88PA-PU	32A 28M1 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See "Speed Grades" on page 316.
 4. NiPdAu Lead Finish.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

ATmega48PA/88PA/168PA/328P

32.3 ATmega168PA

Speed (MHz) ⁽³⁾	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20	1.8 - 5.5	ATmega168PA-AU ATmega168PA-MMH ⁽⁴⁾ ATmega168PA-MU ATmega168PA-PU	32A 28M1 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See "Speed Grades" on page 316.
 4. NiPdAu Lead Finish.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

ATmega48PA/88PA/168PA/328P

32.4 ATmega328P

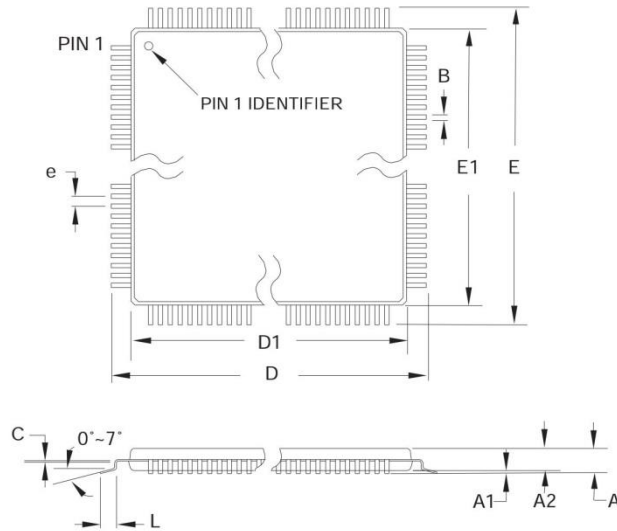
Speed (MHz)	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20 ⁽³⁾	1.8 - 5.5	ATmega328P- AU ATmega328P- MU ATmega328P- PU	32A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See [Figure 28-1 on page 316](#).

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

33. Packaging Information

33.1 32A



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	1.20	
A1	0.05	-	0.15	
A2	0.95	1.00	1.05	
D	8.75	9.00	9.25	
D1	6.90	7.00	7.10	Note 2
E	8.75	9.00	9.25	
E1	6.90	7.00	7.10	Note 2
B	0.30	-	0.45	
C	0.09	-	0.20	
L	0.45	-	0.75	
e	0.80 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-026, Variation ABA.
 2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
 3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

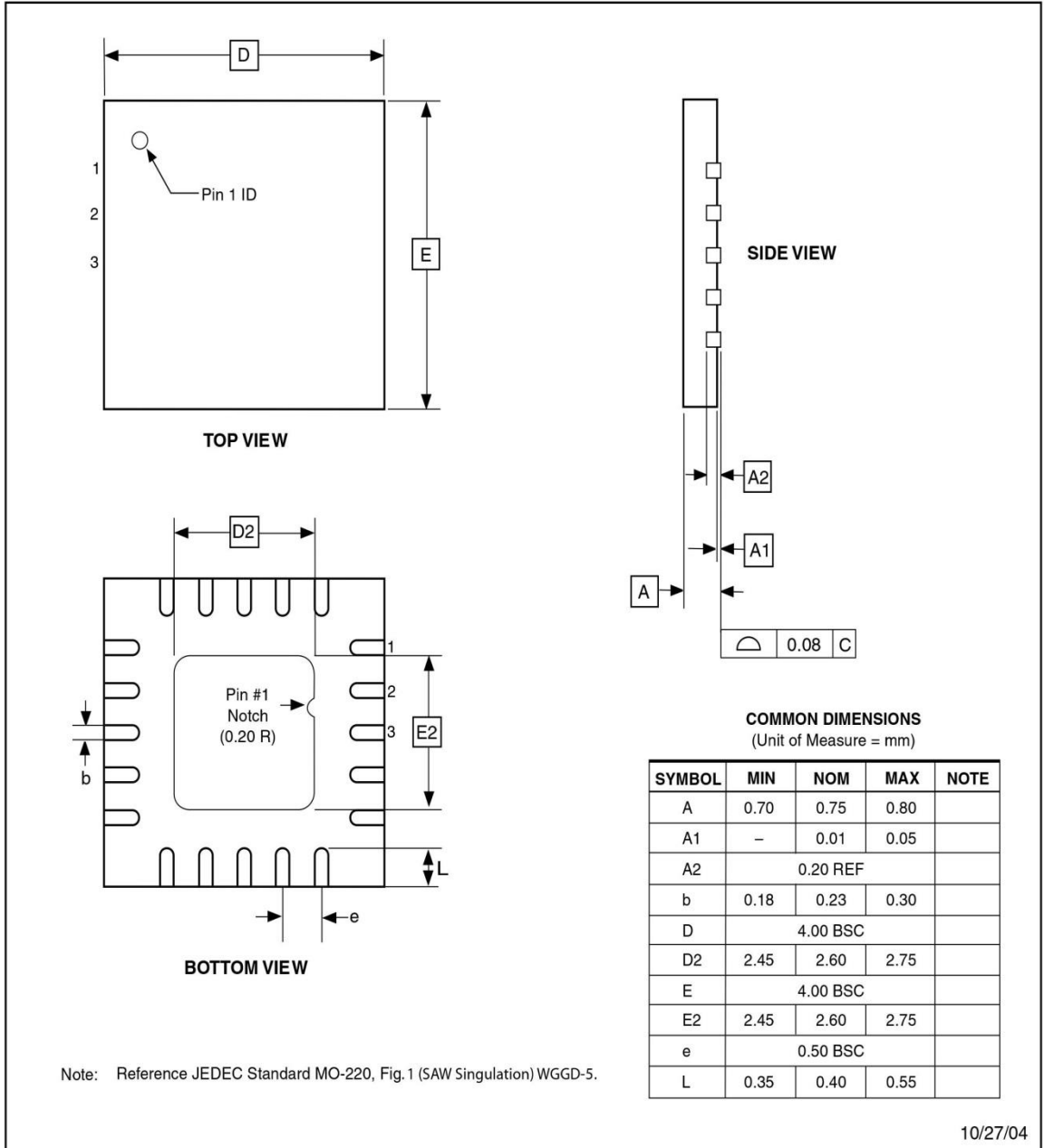
ATMEL 2325 Orchard Parkway
San Jose, CA 95131

TITLE
32A, 32-lead, 7 x 7 mm Body Size, 1.0 mm Body Thickness,
0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)

DRAWING NO.
32A

REV.
B

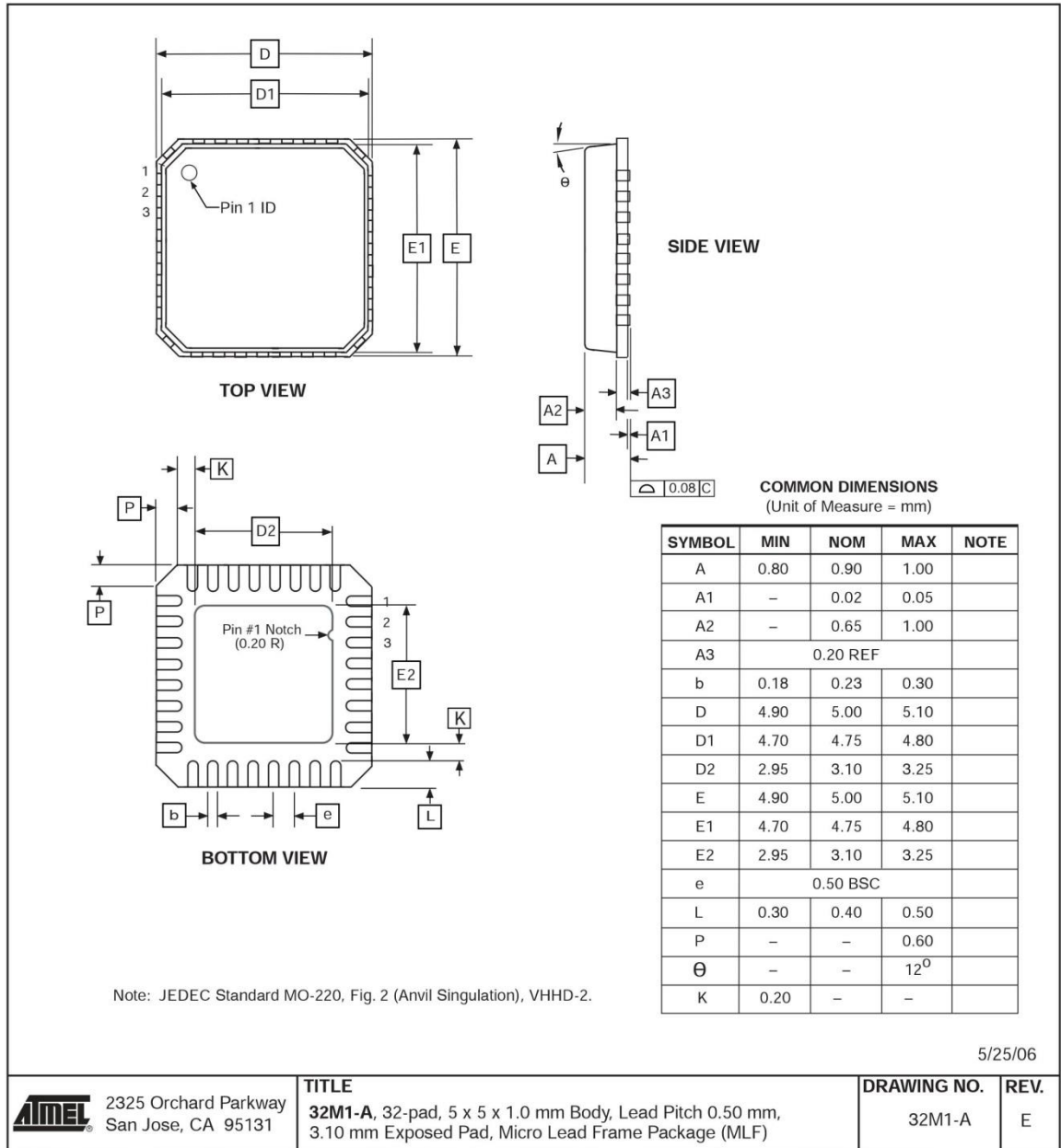
33.2 28M1



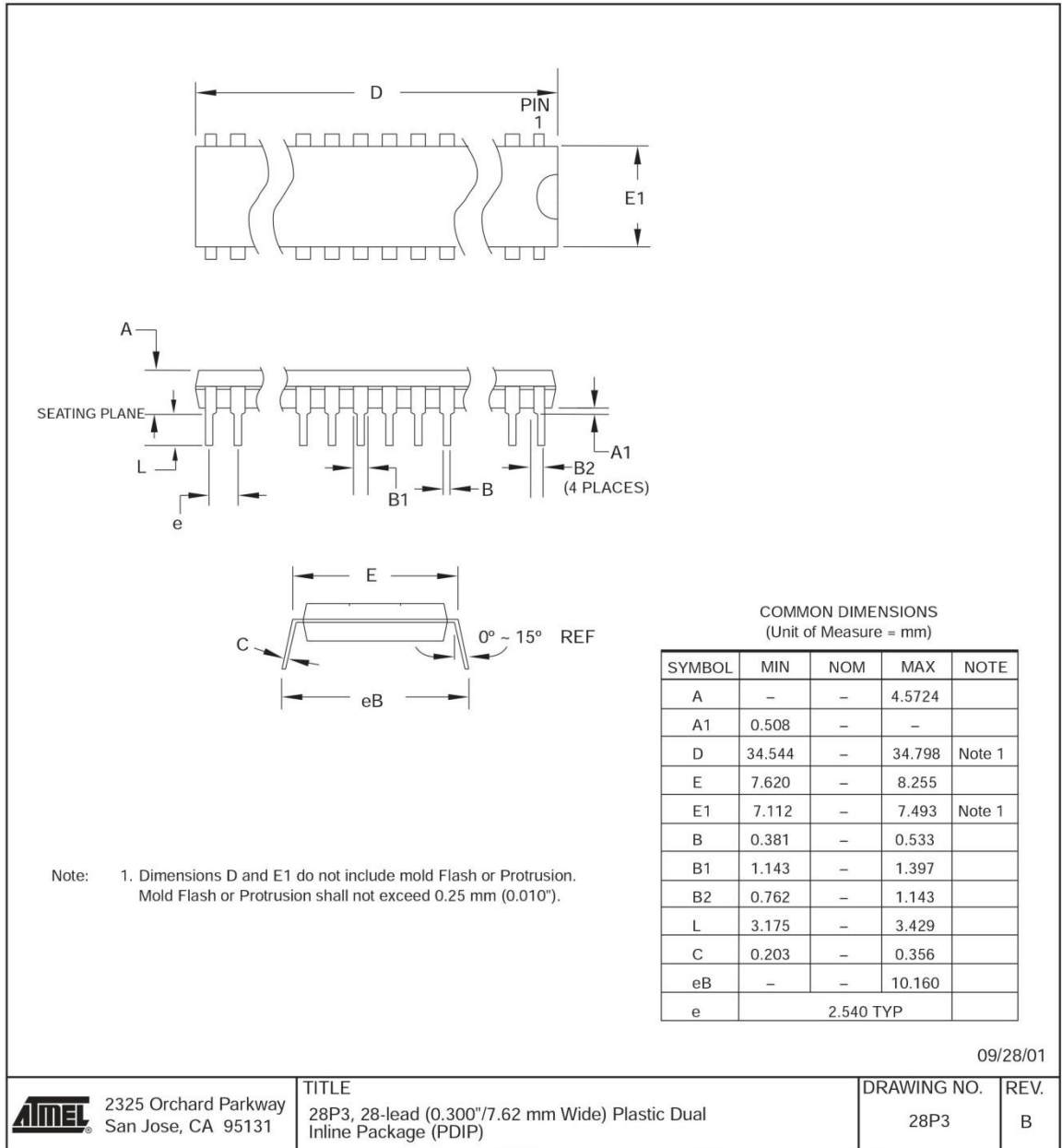
10/27/04

2325 Orchard Parkway	TITLE 20M1, 20-pad, 4 x 4 x 0.8 mm Body, Lead Pitch 0.50 mm,	DRAWING NO.	REV.
		20M1	A

33.3 32M1-A



33.4 28P3



34. Errata**34.1 Errata ATmega48PA**

The revision letter in this section refers to the revision of the ATmega48PA device.

34.1.1 Rev. D

No known errata.

34.2 Errata ATmega88PA

The revision letter in this section refers to the revision of the ATmega88PA device.

34.2.1 Rev. F

No known errata.

34.3 Errata ATmega168PA

The revision letter in this section refers to the revision of the ATmega168PA device.

34.3.1 Rev E

No known errata.

34.4 Errata ATmega328P

The revision letter in this section refers to the revision of the ATmega328P device.

34.4.1 Rev D

No known errata.

34.4.2 Rev C

Not sampled.

34.4.3 Rev B

- **Unstable 32 kHz Oscillator**

- 1. Unstable 32 kHz Oscillator**

The 32 kHz oscillator does not work as system clock.

The 32 kHz oscillator used as asynchronous timer is inaccurate.

Problem Fix/ Workaround

None

34.4.4 Rev A

- **Unstable 32 kHz Oscillator**

- 1. Unstable 32 kHz Oscillator**

The 32 kHz oscillator does not work as system clock.

The 32 kHz oscillator used as asynchronous timer is inaccurate.

Problem Fix/ Workaround

None

35. Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

35.1 Rev. 8161D – 10/09

1. Inserted [Table](#) on page 32, Capacitance for Low-frequency Oscillator.

35.2 Rev. 8161C – 05/09

1. Updated "Features" on page 1 for ATmega48PA/88PA/168PA/328P.
2. Updated "Overview" on page 5 included the [Table 2-1](#) on page 6.
3. Updated "AVR Memories" on page 16 included "Register Description" on page 21 and inserted [Figure 7-1](#) on page 17.
4. Updated "Register Description" on page 44.
5. Updated "System Control and Reset" on page 46.
6. Updated "Interrupts" on page 57.
7. Updated "External Interrupts" on page 70.
8. Updated "Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P" on page 277.
9. Inserted "ATmega168PA DC Characteristics" on page 315.
10. Inserted "ATmega328P DC Characteristics" on page 316.
11. Inserted "ATmega168PA Typical Characteristics" on page 375.
12. Inserted "ATmega328P Typical Characteristics" on page 399.
13. Inserted Ordering Information for "ATmega168PA" on page 432.
14. Inserted Ordering Information for "ATmega328P" on page 433.
15. Inserted "Errata ATmega328P" on page 438.
16. Editing updates.

35.3 Rev. 8161B – 01/09

1. Updated "Features" on page 1 for ATmega48PA and updated the book accordingly.
2. Updated "Overview" on page 5 included the [Table 2-1](#) on page 6.
3. Updated "AVR Memories" on page 16 included "Register Description" on page 21 and inserted [Figure 7-1](#) on page 17.
4. Updated "Register Description" on page 44.
5. Updated "System Control and Reset" on page 46.

6. Updated "Interrupts" on page 57.
7. Updated "External Interrupts" on page 70.
8. Inserted Typical characteristics for "ATmega48PA Typical Characteristics" on page 327.
9. Updated figure names in Typical characteristics for "ATmega88PA Typical Characteristics" on page 351.
10. Inserted "ATmega48PA DC Characteristics" on page 314.
11. Updated Table 28-1 on page 317 by removing the footnote from V_{CC}/User calibration
12. Updated Table 28-7 on page 323 by removing Max value (2.5 LSB) from Absolute accuracy, V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz.
13. Inserted Ordering Information for "ATmega48PA" on page 430.

35.4 Rev. 8161A – 11/08

1. Initial revision (Based on the ATmega48P/88P/168P/328P datasheet 8025F-AVR-08/08).
2. Changes done compared to ATmega48P/88P/168P/328P datasheet 8025F-AVR-08/08:
 - Updated "DC Characteristics" on page 313 with new typical values for I_{CC}.
 - Updated "Speed Grades" on page 316.
 - New graphics in "Typical Characteristics" on page 326.
 - New "Ordering Information" on page 430.

Table of Contents

	Features	1
1	Pin Configurations	2
	1.1 Pin Descriptions	3
2	Overview	5
	2.1 Block Diagram	5
	2.2 Comparison Between ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P 6	
3	Resources	7
4	Data Retention	7
5	About Code Examples	7
6	AVR CPU Core	8
	6.1 Overview	8
	6.2 ALU – Arithmetic Logic Unit	9
	6.3 Status Register	9
	6.4 General Purpose Register File	11
	6.5 Stack Pointer	12
	6.6 Instruction Execution Timing	13
	6.7 Reset and Interrupt Handling	14
7	AVR Memories	16
	7.1 Overview	16
	7.2 In-System Reprogrammable Flash Program Memory	16
	7.3 SRAM Data Memory	18
	7.4 EEPROM Data Memory	19
	7.5 I/O Memory	20
	7.6 Register Description	21
8	System Clock and Clock Options	26
	8.1 Clock Systems and their Distribution	26
	8.2 Clock Sources	27
	8.3 Low Power Crystal Oscillator	28
	8.4 Full Swing Crystal Oscillator	30
	8.5 Low Frequency Crystal Oscillator	32
	8.6 Calibrated Internal RC Oscillator	33

8.7	128 kHz Internal Oscillator	34
8.8	External Clock	34
8.9	Clock Output Buffer	35
8.10	Timer/Counter Oscillator	35
8.11	System Clock Prescaler	35
8.12	Register Description	37
9	Power Management and Sleep Modes	39
9.1	Sleep Modes	39
9.2	BOD Disable	40
9.3	Idle Mode	40
9.4	ADC Noise Reduction Mode	40
9.5	Power-down Mode	41
9.6	Power-save Mode	41
9.7	Standby Mode	41
9.8	Extended Standby Mode	41
9.9	Power Reduction Register	42
9.10	Minimizing Power Consumption	42
9.11	Register Description	44
10	System Control and Reset	46
10.1	Resetting the AVR	46
10.2	Reset Sources	46
10.3	Power-on Reset	47
10.4	External Reset	48
10.5	Brown-out Detection	48
10.6	Watchdog System Reset	49
10.7	Internal Voltage Reference	49
10.8	Watchdog Timer	50
10.9	Register Description	54
11	Interrupts	57
11.1	Interrupt Vectors in ATmega48PA	57
11.2	Interrupt Vectors in ATmega88PA	59
11.3	Interrupt Vectors in ATmega168PA	62
11.4	Interrupt Vectors in ATmega328P	65
11.5	Register Description	68
12	External Interrupts	70

12.1	Pin Change Interrupt Timing	70
12.2	Register Description	71
13	<i>I/O-Ports</i>	75
13.1	Overview	75
13.2	Ports as General Digital I/O	76
13.3	Alternate Port Functions	80
13.4	Register Description	92
14	<i>8-bit Timer/Counter0 with PWM</i>	94
14.1	Features	94
14.2	Overview	94
14.3	Timer/Counter Clock Sources	96
14.4	Counter Unit	96
14.5	Output Compare Unit	97
14.6	Compare Match Output Unit	98
14.7	Modes of Operation	99
14.8	Timer/Counter Timing Diagrams	104
14.9	Register Description	106
15	<i>16-bit Timer/Counter1 with PWM</i>	113
15.1	Features	113
15.2	Overview	113
15.3	Accessing 16-bit Registers	115
15.4	Timer/Counter Clock Sources	118
15.5	Counter Unit	119
15.6	Input Capture Unit	120
15.7	Output Compare Units	122
15.8	Compare Match Output Unit	124
15.9	Modes of Operation	125
15.10	Timer/Counter Timing Diagrams	132
15.11	Register Description	134
16	<i>Timer/Counter0 and Timer/Counter1 Prescalers</i>	141
16.1	Internal Clock Source	141
16.2	Prescaler Reset	141
16.3	External Clock Source	141
16.4	Register Description	143

17	8-bit Timer/Counter2 with PWM and Asynchronous Operation	144
17.1	Features	144
17.2	Overview	144
17.3	Timer/Counter Clock Sources	145
17.4	Counter Unit	145
17.5	Output Compare Unit	146
17.6	Compare Match Output Unit	148
17.7	Modes of Operation	149
17.8	Timer/Counter Timing Diagrams	153
17.9	Asynchronous Operation of Timer/Counter2	155
17.10	Timer/Counter Prescaler	156
17.11	Register Description	158
18	SPI – Serial Peripheral Interface	166
18.1	Features	166
18.2	Overview	166
18.3	\overline{SS} Pin Functionality	171
18.4	Data Modes	171
18.5	Register Description	173
19	USART0	176
19.1	Features	176
19.2	Overview	176
19.3	Clock Generation	177
19.4	Frame Formats	180
19.5	USART Initialization	182
19.6	Data Transmission – The USART Transmitter	183
19.7	Data Reception – The USART Receiver	186
19.8	Asynchronous Data Reception	190
19.9	Multi-processor Communication Mode	193
19.10	Register Description	195
19.11	Examples of Baud Rate Setting	199
20	USART in SPI Mode	204
20.1	Features	204
20.2	Overview	204
20.3	Clock Generation	204
20.4	SPI Data Modes and Timing	205

20.5	Frame Formats	206
20.6	Data Transfer	208
20.7	AVR USART MSPIM vs. AVR SPI	210
20.8	Register Description	211
21	2-wire Serial Interface	214
21.1	Features	214
21.2	2-wire Serial Interface Bus Definition	214
21.3	Data Transfer and Frame Format	216
21.4	Multi-master Bus Systems, Arbitration and Synchronization	218
21.5	Overview of the TWI Module	221
21.6	Using the TWI	223
21.7	Transmission Modes	227
21.8	Multi-master Systems and Arbitration	240
21.9	Register Description	241
22	Analog Comparator	246
22.1	Overview	246
22.2	Analog Comparator Multiplexed Input	246
22.3	Register Description	247
23	Analog-to-Digital Converter	250
23.1	Features	250
23.2	Overview	250
23.3	Starting a Conversion	252
23.4	Prescaling and Conversion Timing	253
23.5	Changing Channel or Reference Selection	255
23.6	ADC Noise Canceler	256
23.7	ADC Conversion Result	261
23.8	Temperature Measurement	261
23.9	Register Description	262
24	debugWIRE On-chip Debug System	267
24.1	Features	267
24.2	Overview	267
24.3	Physical Interface	267
24.4	Software Break Points	268
24.5	Limitations of debugWIRE	268
24.6	Register Description	268

25	<i>Self-Programming the Flash, ATmega48PA</i>	269
25.1	Overview	269
25.2	Addressing the Flash During Self-Programming	270
25.3	Register Description	275
26	<i>Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P</i>	277
26.1	Features	277
26.2	Overview	277
26.3	Application and Boot Loader Flash Sections	277
26.4	Read-While-Write and No Read-While-Write Flash Sections	278
26.5	Boot Loader Lock Bits	280
26.6	Entering the Boot Loader Program	281
26.7	Addressing the Flash During Self-Programming	282
26.8	Self-Programming the Flash	282
26.9	Register Description	292
27	<i>Memory Programming</i>	294
27.1	Program And Data Memory Lock Bits	294
27.2	Fuse Bits	295
27.3	Signature Bytes	298
27.4	Calibration Byte	298
27.5	Page Size	299
27.6	Parallel Programming Parameters, Pin Mapping, and Commands	299
27.7	Parallel Programming	301
27.8	Serial Downloading	308
28	<i>Electrical Characteristics</i>	313
28.1	Absolute Maximum Ratings*	313
28.2	DC Characteristics	313
28.3	Speed Grades	316
28.4	Clock Characteristics	317
28.5	System and Reset Characteristics	318
28.6	SPI Timing Characteristics	319
28.7	2-wire Serial Interface Characteristics	321
28.8	ADC Characteristics	323
28.9	Parallel Programming Characteristics	324
29	<i>Typical Characteristics</i>	326

ATmega48PA/88PA/168PA/328P

29.1ATmega48PA Typical Characteristics	327
29.2ATmega88PA Typical Characteristics	351
29.3ATmega168PA Typical Characteristics	375
29.4ATmega328P Typical Characteristics	399
30 Register Summary	423
31 Instruction Set Summary	427
32 Ordering Information	430
32.1ATmega48PA	430
32.2ATmega88PA	431
32.3ATmega168PA	432
32.4ATmega328P	433
33 Packaging Information	434
33.132A	434
33.228M1	435
33.332M1-A	436
33.428P3	437
34 Errata	438
34.1Errata ATmega48PA	438
34.2Errata ATmega88PA	438
34.3Errata ATmega168PA	438
34.4Errata ATmega328P	438
35 Datasheet Revision History	439
35.1Rev. 8161D – 10/09	439
35.2Rev. 8161C – 05/09	439
35.3Rev. 8161B – 01/09	439
35.4Rev. 8161A – 11/08	440
Table of Contents.....	i



Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2009 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, AVR®, AVR® logo and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

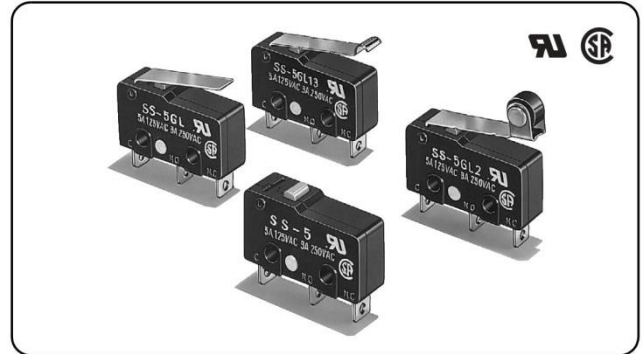
SS

Subminiature Basic Switch

Subminiature Basic Switch Offers High Reliability and Security

- The OMRON's best-selling micro switches of a wide variety from 0.1A to 10.1A.
- A variety of models are available, with operating force ranging from low to high.
- Two split springs ensure a high stability and durability of 30,000,000 operations.

RoHS Compliant




Model Number Legend




- SS-1 2 3 4 5 6
- 1. Ratings**
 - 10 : 250 VAC 10.1A
 - 5 : 125 VAC 5 A
 - 01 : 30 VDC 0.1A
 - 2. Actuator**
 - None : Pin plunger
 - GL : Hinge lever
 - GL111 : Long hinge lever
 - GL13 : Simulated roller lever
 - GL2 : Hinge roller lever
 - GL02 : Hinge roller lever (Roller material: Stainless) heat-resistant
 - 3. Maximum Operating Force (OF)**
 - None : 1.47 N {150 gf}
 - F : 0.49 N {50 gf} (0.1 A, 5 A) Note. These values are for the pin plunger models.
 - E : 0.25 N {25 gf} (0.1 A)
 - 4. Contact form**
 - None : SPDT
 - 2 : SPST-NC
 - 3 : SPST-NO
 - 5. Terminals**
 - None : Solder terminals
 - T : Quick-connect terminals (#110)
 - D : PCB terminals
 - 6. Heat resistance**
 - None : Standard (85°C)
 - T : Heat-resistant (120°C)

List of Models


●Standard Models

Actuator	Terminals	Contact Form	Maximum Operating Force (OF)	Ratings			
				10.1 A	5 A	0.1 A	
Pin plunger 	Solder terminals	SPDT	1.47 N {150 gf}	SS-10	SS-5	SS-01	
		SPST-NC		SS-10-2	SS-5-2	SS-01-2	
		SPST-NO		SS-10-3	SS-5-3	SS-01-3	
	Quick-connect terminals (#110)	SPDT		SS-10T	SS-5T	SS-01T	
		SPST-NC		SS-10-2T	SS-5-2T	SS-01-2T	
		SPST-NO		SS-10-3T	SS-5-3T	SS-01-3T	
	PCB terminals	SPDT		SS-10D	SS-5D	SS-01D	
		SPST-NC		SS-10-2D	SS-5-2D	SS-01-2D	
		SPST-NO		SS-10-3D	SS-5-3D	SS-01-3D	
	Solder terminals	SPDT	0.49 N {50 gf}	-	SS-5-F	SS-01-F	
				-	SS-5-F-2	SS-01-F-2	
				-	SS-5-F-3	SS-01-F-3	
		Quick-connect terminals (#110)		SPDT	-	SS-5-FT	SS-01-FT
				SPST-NC	-	SS-5-F-2T	SS-01-F-2T
				SPST-NO	-	SS-5-F-3T	SS-01-F-3T
		PCB terminals		SPDT	-	SS-5-FD	SS-01-FD
				SPST-NC	-	SS-5-F-2D	SS-01-F-2D
				SPST-NO	-	SS-5-F-3D	SS-01-F-3D
	Solder terminals	SPDT	0.25 N {25 gf}	-	-	SS-01-E	
				-	-	SS-01-E-2	
				-	-	SS-01-E-3	
		Quick-connect terminals (#110)		SPDT	-	-	SS-01-ET
				SPST-NC	-	-	SS-01-E-2T
				SPST-NO	-	-	SS-01-E-3T
PCB terminals	SPDT	-	-	SS-01-ED			
	SPST-NC	-	-	SS-01-E-2D			
	SPST-NO	-	-	SS-01-E-3D			





Separator (Sold Separately), Terminal Connector (Sold Separately) ➔ Refer to "Basic Switch Common Accessories"

Actuator	Terminals	Contact Form	Maximum Operating Force (OF)	Ratings				
				10.1 A	5 A	0.1 A		
 <p>Hinge lever</p>	Solder terminals	SPDT	0.49 N (50 gf)	SS-10GL	SS-5GL	SS-01GL		
		SPST-NC		SS-10GL-2	SS-5GL-2	SS-01GL-2		
		SPST-NO		SS-10GL-3	SS-5GL-3	SS-01GL-3		
	Quick-connect terminals (#110)	SPDT		0.49 N (50 gf)	SS-10GLT	SS-5GLT	SS-01GLT	
		SPST-NC			SS-10GL-2T	SS-5GL-2T	SS-01GL-2T	
		SPST-NO			SS-10GL-3T	SS-5GL-3T	SS-01GL-3T	
	PCB terminals	SPDT			0.49 N (50 gf)	SS-10GLD	SS-5GLD	SS-01GLD
		SPST-NC				SS-10GL-2D	SS-5GL-2D	SS-01GL-2D
		SPST-NO				SS-10GL-3D	SS-5GL-3D	SS-01GL-3D
	Solder terminals	SPDT	0.16 N (16 gf)			-	SS-5GL-F	SS-01GL-F
		SPST-NC				SS-5GL-F-2	SS-01GL-F-2	
		SPST-NO				SS-5GL-F-3	SS-01GL-F-3	
	Quick-connect terminals (#110)	SPDT		0.16 N (16 gf)		-	SS-5GL-FT	SS-01GL-FT
		SPST-NC				SS-5GL-F-2T	SS-01GL-F-2T	
		SPST-NO				SS-5GL-F-3T	SS-01GL-F-3T	
	PCB terminals	SPDT			0.16 N (16 gf)	-	SS-5GL-FD	SS-01GL-FD
		SPST-NC				SS-5GL-F-2D	SS-01GL-F-2D	
		SPST-NO				SS-5GL-F-3D	SS-01GL-F-3D	
	Solder terminals	SPDT	0.08 N (8 gf)			-	-	SS-01GL-E
		SPST-NC				-	-	SS-01GL-E-2
		SPST-NO				-	-	SS-01GL-E-3
	Quick-connect terminals (#110)	SPDT		0.08 N (8 gf)		-	-	SS-01GL-ET
		SPST-NC				-	-	SS-01GL-E-2T
		SPST-NO				-	-	SS-01GL-E-3T
PCB terminals	SPDT	0.08 N (8 gf)			-	-	SS-01GL-ED	
	SPST-NC				-	-	SS-01GL-E-2D	
	SPST-NO				-	-	SS-01GL-E-3D	
 <p>Long hinge lever</p>	Solder terminals		SPDT		0.39 N (40 gf)	SS-10GL111	SS-5GL111	SS-01GL111
			SPST-NC			SS-10GL111-2	SS-5GL111-2	SS-01GL111-2
			SPST-NO			SS-10GL111-3	SS-5GL111-3	SS-01GL111-3
	Quick-connect terminals (#110)		SPDT	0.39 N (40 gf)		SS-10GL111T	SS-5GL111T	SS-01GL111T
			SPST-NC			SS-10GL111-2T	SS-5GL111-2T	SS-01GL111-2T
			SPST-NO			SS-10GL111-3T	SS-5GL111-3T	SS-01GL111-3T
	PCB terminals	SPDT	0.39 N (40 gf)			SS-10GL111D	SS-5GL111D	SS-01GL111D
		SPST-NC				SS-10GL111-2D	SS-5GL111-2D	SS-01GL111-2D
		SPST-NO				SS-10GL111-3D	SS-5GL111-3D	SS-01GL111-3D
	Solder terminals	SPDT			0.12 N (12 gf)	-	SS-5GL111-F	SS-01GL111-F
		SPST-NC				SS-5GL111-F-2	SS-01GL111-F-2	
		SPST-NO				SS-5GL111-F-3	SS-01GL111-F-3	
	Quick-connect terminals (#110)	SPDT		0.12 N (12 gf)		-	SS-5GL111-FT	SS-01GL111-FT
		SPST-NC				SS-5GL111-F-2T	SS-01GL111-F-2T	
		SPST-NO				SS-5GL111-F-3T	SS-01GL111-F-3T	
	PCB terminals	SPDT	0.12 N (12 gf)			-	SS-5GL111-FD	SS-01GL111FD
		SPST-NC				SS-5GL111-F-2D	SS-01GL111-F-2D	
		SPST-NO				SS-5GL111-F-3D	SS-01GL111-F-3D	
	Solder terminals	SPDT			0.06 N (6 gf)	-	-	SS-01GL111-E
		SPST-NC				-	-	SS-01GL111-E-2
		SPST-NO				-	-	SS-01GL111-E-3
	Quick-connect terminals (#110)	SPDT		0.06 N (6 gf)		-	-	SS-01GL111-ET
		SPST-NC				-	-	SS-01GL111-E-2T
		SPST-NO				-	-	SS-01GL111-E-3T
PCB terminals	SPDT	0.06 N (6 gf)	-			-	SS-01GL111-ED	
	SPST-NC		-			-	SS-01GL111-E-2D	
	SPST-NO		-			-	SS-01GL111-E-3D	
 <p>Simulated roller lever</p>	Solder terminals		SPDT		0.49 N (50 gf)	SS-10GL13	SS-5GL13	SS-01GL13
			SPST-NC			SS-10GL13-2	SS-5GL13-2	SS-01GL13-2
			SPST-NO			SS-10GL13-3	SS-5GL13-3	SS-01GL13-3
	Quick-connect terminals (#110)		SPDT	0.49 N (50 gf)		SS-10GL13T	SS-5GL13T	SS-01GL13T
			SPST-NC			SS-10GL13-2T	SS-5GL13-2T	SS-01GL13-2T
			SPST-NO			SS-10GL13-3T	SS-5GL13-3T	SS-01GL13-3T
	PCB terminals	SPDT	0.49 N (50 gf)			SS-10GL13D	SS-5GL13D	SS-01GL13D
		SPST-NC				SS-10GL13-2D	SS-5GL13-2D	SS-01GL13-2D
		SPST-NO				SS-10GL13-3D	SS-5GL13-3D	SS-01GL13-3D
	Solder terminals	SPDT			0.16 N (16 gf)	-	SS-5GL13-F	SS-01GL13-F
		SPST-NC				SS-5GL13-F-2	SS-01GL13-F-2	
		SPST-NO				SS-5GL13-F-3	SS-01GL13-F-3	
	Quick-connect terminals (#110)	SPDT		0.16 N (16 gf)		-	SS-5GL13-FT	SS-01GL13-FT
		SPST-NC				SS-5GL13-F-2T	SS-01GL13-F-2T	
		SPST-NO				SS-5GL13-F-3T	SS-01GL13-F-3T	
	PCB terminals	SPDT	0.16 N (16 gf)			-	SS-5GL13-FD	SS-01GL13-FD
		SPST-NC				SS-5GL13-F-2D	SS-01GL13-F-2D	
		SPST-NO				SS-5GL13-F-3D	SS-01GL13-F-3D	
	Solder terminals	SPDT			0.08 N (8 gf)	-	-	SS-01GL13-E
		SPST-NC				-	-	SS-01GL13-E-2
		SPST-NO				-	-	SS-01GL13-E-3
	Quick-connect terminals (#110)	SPDT		0.08 N (8 gf)		-	-	SS-01GL13-ET
		SPST-NC				-	-	SS-01GL13-E-2T
		SPST-NO				-	-	SS-01GL13-E-3T
PCB terminals	SPDT	0.08 N (8 gf)	-			-	SS-01GL13-ED	
	SPST-NC		-			-	SS-01GL13-E-2D	
	SPST-NO		-			-	SS-01GL13-E-3D	

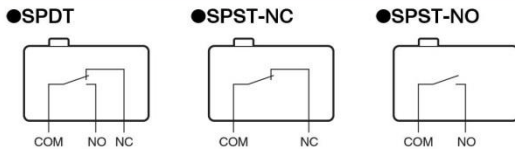
Separator (Sold Separately), Terminal Connector (Sold Separately) ➔ Refer to "Basic Switch Common Accessories"

Actuator	Terminals	Contact Form	Maximum Operating Force (OF)	Ratings		
				10.1 A	5 A	0.1 A
Hinge roller lever 	Solder terminals	SPDT	0.49 N (50 gf)	SS-10GL2	SS-5GL2	SS-01GL2
		SPST-NC		SS-10GL2-2	SS-5GL2-2	SS-01GL2-2
		SPST-NO		SS-10GL2-3	SS-5GL2-3	SS-01GL2-3
	Quick-connect terminals (#110)	SPDT		SS-10GL2T	SS-5GL2T	SS-01GL2T
		SPST-NC		SS-10GL2-2T	SS-5GL2-2T	SS-01GL2-2T
		SPST-NO		SS-10GL2-3T	SS-5GL2-3T	SS-01GL2-3T
	PCB terminals	SPDT		SS-10GL2D	SS-5GL2D	SS-01GL2D
		SPST-NC		SS-10GL2-2D	SS-5GL2-2D	SS-01GL2-2D
		SPST-NO		SS-10GL2-3D	SS-5GL2-3D	SS-01GL2-3D
	Solder terminals	SPDT	0.16 N (16 gf)	-	SS-5GL2-F	SS-01GL2-F
		SPST-NC		-	SS-5GL2-F-2	SS-01GL2-F-2
		SPST-NO		-	SS-5GL2-F-3	SS-01GL2-F-3
	Quick-connect terminals (#110)	SPDT		-	SS-5GL2-FT	SS-01GL2-FT
		SPST-NC		-	SS-5GL2-F-2T	SS-01GL2-F-2T
		SPST-NO		-	SS-5GL2-F-3T	SS-01GL2-F-3T
	PCB terminals	SPDT		-	SS-5GL2-FD	SS-01GL2-FD
		SPST-NC		-	SS-5GL2-F-2D	SS-01GL2-F-2D
		SPST-NO		-	SS-5GL2-F-3D	SS-01GL2-F-3D
	Solder terminals	SPDT	0.08 N (8 gf)	-	-	SS-01GL2-E
		SPST-NC		-	-	SS-01GL2-E-2
		SPST-NO		-	-	SS-01GL2-E-3
	Quick-connect terminals (#110)	SPDT		-	-	SS-01GL2-ET
		SPST-NC		-	-	SS-01GL2-E-2T
		SPST-NO		-	-	SS-01GL2-E-3T
PCB terminals	SPDT	-		-	SS-01GL2-ED	
	SPST-NC	-		-	SS-01GL2-E-2D	
	SPST-NO	-		-	SS-01GL2-E-3D	

●Heat Resistant Models

Actuator	Terminals	Contact Form	Maximum Operating Force (OF)	Ratings		
				10.1 A	5 A	0.1 A
Pin plunger 	Solder terminals	SPDT	1.47 N (150 gf)	SS-10-T	SS-5-T	SS-01-T
	Quick-connect terminals (#110)			SS-10T-T	SS-5T-T	SS-01T-T
	PCB terminals			SS-10D-T	SS-5D-T	SS-01D-T
Hinge lever 	Solder terminals		0.49 N (50 gf)	SS-10GL-T	SS-5GL-T	SS-01GL-T
	Quick-connect terminals (#110)			SS-10GLT-T	SS-5GLT-T	SS-01GLT-T
	PCB terminals			SS-10GLD-T	SS-5GLD-T	SS-01GLD-T
Long hinge lever 	Solder terminals		0.39 N (40 gf)	SS-10GL111-T	SS-5GL111-T	SS-01GL111-T
	Quick-connect terminals (#110)			SS-10GL111D-T	SS-5GL111D-T	SS-01GL111D-T
	PCB terminals			SS-10GL13-T	SS-5GL13-T	SS-01GL13-T
Simulated roller lever 	Solder terminals	0.49 N (50 gf)	SS-10GL13D-T	SS-5GL13D-T	SS-01GL13D-T	
	Quick-connect terminals (#110)		SS-10GL02-T	SS-5GL02-T	SS-01GL02-T	
	PCB terminals		SS-10GL02D-T	SS-5GL02D-T	SS-01GL02D-T	
Hinge roller lever (Roller material: stainless steel) 	Solder terminals	0.49 N (50 gf)	SS-10GL02T-T	SS-5GL02T-T	SS-01GL02T-T	
	Quick-connect terminals (#110)		SS-10GL02D-T	SS-5GL02D-T	SS-01GL02D-T	
	PCB terminals					

Contact Form



Contact Specifications

Item	Model	SS-10 models		SS-5 models	SS-01 models
		Specification	Rivet		Crossbar
Contact	Material	Silver alloy		Silver	Gold alloy
	Gap (standard value)	0.5 mm		0.25 mm	
	Inrush current	NC	20 A max.		1 A max.
NO		15 A max.	10 A max.	1 A max.	
Minimum applicable load (reference value)*		5 VDC 160 mA		5 VDC 1 mA	

* Please refer to "●Using Micro Loads" in "Precautions" for more information on the minimum applicable load.

Separator (Sold Separately), Terminal Connector (Sold Separately) ➔ Refer to "Basic Switch Common Accessories"

Ratings

Model	Item Rated voltage	Resistive load
SS-10 models	250 VAC	10.1 A
SS-5 models	125 VAC	5 A
	250 VAC	3 A
SS-01 models	125 VAC	0.1 A
	30 VDC	0.1 A

Note. The above rating values apply under the following test conditions.
 (1) Ambient temperature: 20±2°C
 (2) Ambient humidity: 65±5%
 (3) Operating frequency: 30 operations/min

Approved Safety Standards

Models shown in the "List of Models" are UL and CSA approved models.

Note. Note that heat resistant models are not standard approved models.

UL (UL1054)/CSA (CSA C22.2 No.55)

Rated voltage	Model	SS-10	SS-5	SS-01
125 VAC		-	5 A	0.1 A
250 VAC		10.1 A	3 A	-
30 VDC		-	-	0.1 A

Consult your OMRON sales representative for specific models with VDE standard approvals.
VDE (EN61058-1)

Rated voltage	Model	SS-10	SS-5
250 VAC		10 A	5 A

Testing conditions: 5E4 (50,000 operations)
 T85 (0°C to 85°C)

Characteristics

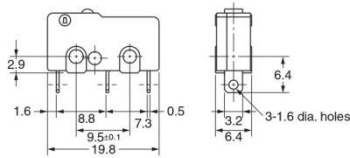
Item	Model	SS-10 models	SS-5 models	SS-01 models
Permissible operating speed		0.1 mm to 1 m/s (for pin plunger models)		
Permissible operating frequency	Mechanical	400 operations/min		
	Electrical	60 operations/min		
Insulation resistance		100 MΩ min. (at 500 VDC with insulation tester)		
Contact resistance (initial value)	OF 1.47 N models	30 mΩ max.		50 mΩ max.
	OF 0.49 N models	-	50 mΩ max.	100 mΩ max.
	OF 0.25 N models	-	-	150 mΩ max.
Dielectric strength *1	Between terminals of the same polarity	1,000 VAC 50/60 Hz for 1 min		600 VAC 50/60 Hz for 1 min
	Between current-carrying metal parts and ground	1,500 VAC 50/60 Hz for 1 min		
	Between each terminals and non-current-carrying metal parts	1,500 VAC 50/60 Hz for 1 min		
Vibration resistance *2	Malfunction	10 to 55 Hz, 1.5 mm double amplitude		
Shock resistance	Durability	OF 1.47 N models	1,000 m/s ² {approx. 100G} max.	
		OF 0.49 N models	500 m/s ² {approx. 50G} max.	
		OF 0.25 N models	500 m/s ² {approx. 50G} max.	
	Malfunction *2	OF 1.47 N models	300 m/s ² {approx. 30G} max.	
		OF 0.49 N models	200 m/s ² {approx. 20G} max.	
Durability *3	Mechanical	10,000,000 operations min. (60 operations/min)	30,000,000 operations min. (60 operations/min)	
	Electrical	50,000 operations min. (30 operations/min)	200,000 operations min. (30 operations/min)	
Degree of protection		IEC IP40		
Degree of protection against electric shock		Class I		
Proof tracking index (PTI)		175		
Ambient operating temperature		-25°C to +85°C (at ambient humidity of 60% max.) (with no icing or condensation)		
Ambient operating humidity		85% max. (for +5°C to +35°C)		
Weight		Approx. 1.6g (pin plunger models)		

Note. The data given above are initial values.

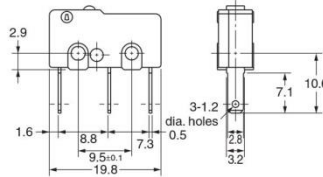
- *1. The values for dielectric strength shown are for models with a Separator (refer to "Micro Switch Common Accessories").
- *2. The values are at Free Position and Total Travel Position values for pin plunger, and Total Travel Position value for lever. Close or open circuit of the contact is 1ms max.
- *3. For testing conditions, consult your OMRON sales representative.

Terminals/Apearances (Unit: mm)

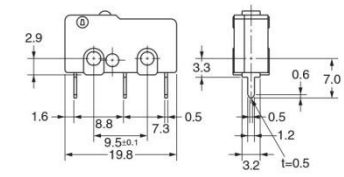
●Solder terminals



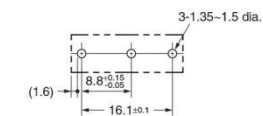
●Quick connect terminals (#110)



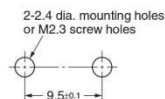
●PCB terminals



<PCB Mounting Dimensions (Reference)>



Mounting Holes (Unit: mm)

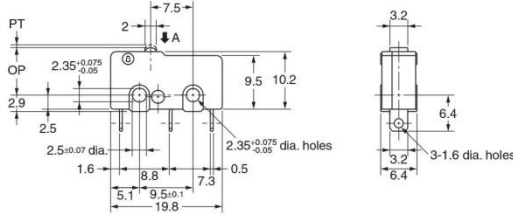
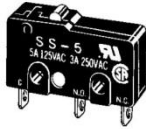


Dimensions (Unit: mm) and Operating Characteristics

The illustrations and drawings are for solder terminals models.

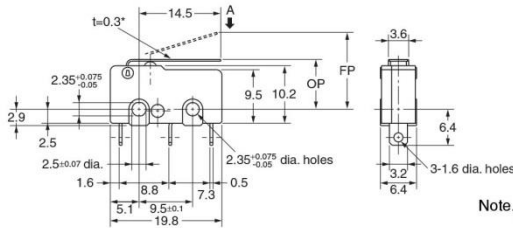
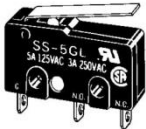
Refer to "Terminals/Appearances" of the previous page for details on models with quick connect terminals (#110) or PCB terminals.

- Pin plunger
- SS-10
- SS-5 (-F)
- SS-01 (-E, -F)



Operating Characteristics		Model	SS-10	SS-5 SS-01	SS-5-F SS-01-F	SS-01-E
Operating Force	OF	Max.	1.47 N {150 gf}	1.47 N {150 gf}	0.49 N {50 gf}	0.25 N {25 gf}
	Releasing Force	RF	Min.	0.25 N {25 gf}	0.25 N {25 gf}	0.04 N {4 gf}
Pretravel	PT	Max.	0.6 mm	0.5 mm	0.5 mm	0.5 mm
	Overtravel	OT	Min.	0.4 mm	0.5 mm	0.5 mm
Movement Differential	MD	Max.	0.12 mm	0.1 mm	0.1 mm	0.1 mm
Operating Position	OP		8.4±0.5 mm			

- Hinge lever
- SS-10GL
- SS-5GL (-F)
- SS-01GL (-E, -F)

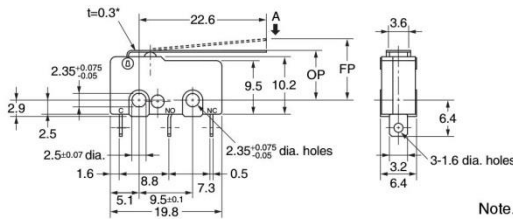
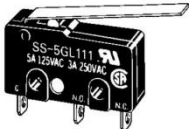


* Stainless-steel lever

Note. The indicated reference values of RF are for cases where the lever weight is not applied to the plunger.

Operating Characteristics		Model	SS-10GL	SS-5GL SS-01GL	SS-5GL-F SS-01GL-F	SS-01GL-E
Operating Force	OF	Max.	0.49 N {50 gf}	0.49 N {50 gf}	0.16 N {16 gf}	0.08 N {8 gf}
	Releasing Force	RF	Min.	0.06 N {6 gf}	0.06 N {6 gf}	0.02 N {2 gf}
Overtravel	OT	Min.	1.0 mm	1.2 mm	1.2 mm	1.2 mm
Movement Differential	MD	Max.	1.0 mm	0.8 mm	0.8 mm	0.8 mm
Free Position	FP	Max.	13.6 mm			
Operating Position	OP		8.8±0.8 mm			

- Long hinge lever
- SS-10GL111
- SS-5GL111 (-F)
- SS-01GL111 (-E, -F)



* Stainless-steel lever

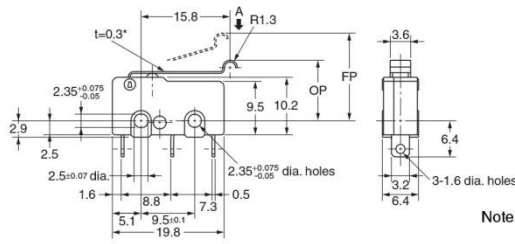
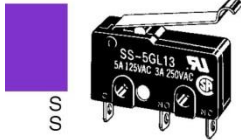
Note. The indicated reference values of RF are for cases where the lever weight is not applied to the plunger.

Operating Characteristics		Model	SS-10GL111	SS-5GL111 SS-01GL111	SS-5GL111-F SS-01GL111-F	SS-01GL111-E
Operating Force	OF	Max.	0.39 N {40 gf}	0.39 N {40 gf}	0.12 N {12 gf}	0.06 N {6 gf}
	Releasing Force	RF	Min.	0.03 N {3 gf}	0.03 N {3 gf}	0.02 N {2 gf}
Overtravel	OT	Min.	1.2 mm	1.2 mm	1.2 mm	1.2 mm
Movement Differential	MD	Max.	1.2 mm	1.2 mm	1.2 mm	1.2 mm
Free Position	FP	Max.	16.8 mm			
Operating Position	OP		8.8±1.5 mm			8.8±2 mm

Note 1. Unless otherwise specified, a tolerance of ±0.4 mm applies to all dimensions.

Note 2. The operating characteristics are for operation in the A direction (↓).

● Simulated roller lever
 SS-10GL13
 SS-5GL13 (-F)
 SS-01GL13 (-E, -F)

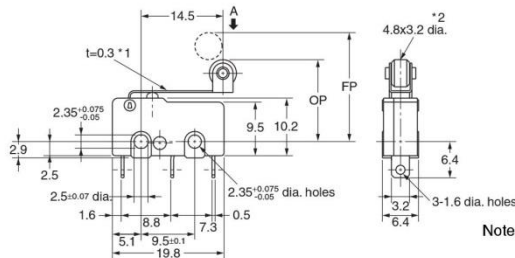


Note. The indicated reference values of RF are for cases where the lever weight is not applied to the plunger.

* Stainless-steel lever

Operating Characteristics		Model	SS-10GL13	SS-5GL13 SS-01GL13	SS-5GL13-F SS-01GL13-F	SS-01GL13-E
Operating Force	OF	Max.	0.49 N {50 gf}	0.49 N {50 gf}	0.16 N {16 gf}	0.08 N {8 gf}
	Releasing Force	RF	Min.	0.06 N {6 gf}	0.02 N {2 gf}	0.01 N {1 gf} (reference value)
Overtravel	OT	Min.	1.0 mm	1.2 mm	1.2 mm	1.2 mm
	Movement Differential	MD	Max.	1.0 mm	0.8 mm	0.8 mm
Free Position	FP	Max.	15.5 mm			
Operating Position	OP		10.7±0.8 mm			

● Hinge roller lever
 SS-10GL2
 SS-5GL2 (-F)
 SS-01GL2 (-E, -F)



Note. The indicated reference values of RF are for cases where the lever weight is not applied to the plunger.

*1. Stainless-steel lever
 *2. Polyacetal resin roller

Operating Characteristics		Model	SS-10GL2	SS-5GL2 SS-01GL2	SS-5GL2-F SS-01GL2-F	SS-01GL2-E
Operating Force	OF	Max.	0.49 N {50 gf}	0.49 N {50 gf}	0.16 N {16 gf}	0.08 N {8 gf}
	Releasing Force	RF	Min.	0.06 N {6 gf}	0.02 N {2 gf}	0.01 N {1 gf} (reference value)
Overtravel	OT	Min.	1.0 mm	1.2 mm	1.2 mm	1.2 mm
	Movement Differential	MD	Max.	1.0 mm	0.8 mm	0.8 mm
Free Position	FP	Max.	19.3 mm			
Operating Position	OP		14.5±0.8 mm			

Note 1. Unless otherwise specified, a tolerance of ±0.4 mm applies to all dimensions.
 Note 2. The operating characteristics are for operation in the A direction (↓).

Precautions

★Please refer to "Common Precautions" for correct use.

Cautions

●Soldering

- Complete the soldering at the iron tip temperature below 350°C within 5 seconds, and do not apply any external force for 1 minute after soldering. Soldering at an excessively high temperature or soldering for more than 5 seconds may deteriorate the characteristics of the Switch.
- Be sure to apply only the minimum required amount of flux. Switch may have contact failures if flux intrudes into the interior of the Switch.
- If the PCB terminal models are soldered in the solder bath, flux will permeate inside the Switch and cause contact failure. Therefore, manually solder the PCB terminal.

Correct Use

●Mounting

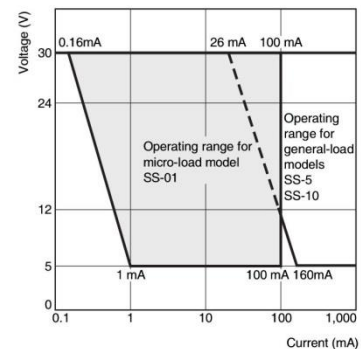
- Use M2.3 mounting screw with plane washers or spring washers to securely mount the Switch. Tighten the screws to a torque of 0.23 to 0.26 N·m (2.3 to 2.7 kgf·cm).
- Mount the Switch onto a flat surface. Mounting on an uneven surface may cause deformation of the Switch, resulting in faulty operation or breakage in the housing.

●Using Micro Loads

Using a model for ordinary loads to open or close the contact of a micro load circuit may result in faulty contact. Use models that operate in the following range. However, even when using micro load models within the following operating range, if inrush current occurs when the contact is opened or closed, it may increase the contact wear and so decrease durability. Therefore, insert a contact protection circuit where necessary. The N-level reference value applies for the minimum applicable load. This value indicates the malfunction reference level for the reliability level of 60% (λ_{60}).

(JIS C5003)

The equation, $\lambda_{60}=0.5 \times 10^{-6}/\text{operation}$ indicates that the estimated malfunction rate is less than $\frac{1}{2,000,000}$ operations with a reliability level of 60%.





• Application examples provided in this document are for reference only. In actual applications, confirm equipment functions and safety before using the product.
• Consult your OMRON representative before using the product under conditions which are not described in the manual or applying the product to nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment, and other systems or equipment that may have a serious influence on lives and property if used improperly. Make sure that the ratings and performance characteristics of the product provide a margin of safety for the system or equipment, and be sure to provide the system or equipment with double safety mechanisms.

Note: Do not use this document to operate the Unit.

OMRON Corporation

ELECTRONIC AND MECHANICAL COMPONENTS COMPANY

Contact: www.omron.com/ecb

Cat. No.B032-E1-13
0812(0207)(O)

SN74LV1T04 Single Power Supply Inverter Gate CMOS Logic Level Shifter

1 Features

- Single-Supply Voltage Translator at 5.0/3.3/2.5/1.8V V_{CC}
- Operating Range of 1.8V to 5.5V
- Up Translation
 - 1.2V⁽¹⁾ to 1.8V at 1.8V V_{CC}
 - 1.5V⁽¹⁾ to 2.5V at 2.5V V_{CC}
 - 1.8V⁽¹⁾ to 3.3V at 3.3V V_{CC}
 - 3.3V to 5.0V at 5.0V V_{CC}
- Down Translation
 - 3.3V to 1.8V at 1.8V V_{CC}
 - 3.3V to 2.5V at 2.5V V_{CC}
 - 5.0V to 3.3V at 3.3V V_{CC}
- Logic Output is Referenced to V_{CC}
- Output Drive
 - 8mA Output Drive at 5V
 - 7mA Output Drive at 3.3V
 - 3mA Output Drive at 1.8V
- Characterized up to 50MHz at 3.3V V_{CC}
- 5V Tolerance on Input Pins
- –40°C to 125°C Operating Temperature Range
- Pb-Free Packages Available: SC-70 (DCK)
 - 2 × 2.1 × 0.65 mm (Height 1.1mm)
- Latch-Up Performance Exceeds 250mA Per JESD 17
- ESD Performance Tested Per JESD 22
 - 2000-V Human-Body Model (A114-B, Class II)
 - 200-V Machine Model (A115-A)
 - 1000-V Charged-Device Model (C101)
- Supports Standard Logic Pinouts
- CMOS Output B Compatible with AUP1G and LVC1G Families

⁽¹⁾ Refer to the V_{IH}/V_{IL} and output drive for lower V_{CC} condition.

2 Applications

- Industrial controllers
- Telecom
- Portable applications
- Servers
- PC and notebooks
- Automotive

3 Description

SN74LV1T04 is a low voltage CMOS gate logic that operates at a wider voltage range for industrial, portable, telecom, and automotive applications. The output level is referenced to the supply voltage and is able to support 1.8V/2.5V/3.3V/5V CMOS levels.

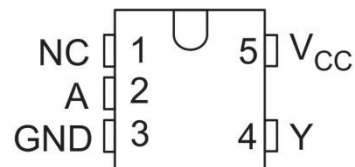
The input is designed with a lower threshold circuit to match 1.8V input logic at $V_{CC} = 3.3V$ and can be used in 1.8V to 3.3V level up translation. In addition, the 5V tolerant input pins enable down translation (e.g. 3.3V to 2.5V output at $V_{CC} = 2.5V$). The wide V_{CC} range of 1.8V to 5.5V allows generation of desired output levels to connect to controllers or processors.

The SN74LV1T04 is designed with current-drive capability of 8 mA to reduce line reflections, overshoot, and undershoot caused by high-drive outputs.

Device Information

ORDER NUMBER	PACKAGE	BODY SIZE
SN74LV1T04DBVR	SOT-23 (5)	2,90mm x 1,60mm
SN74LV1T04DCKR	SC70 (5)	2,00mm x 1,25mm

DCK or DBV PACKAGE
(TOP VIEW)



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

Table of Contents

1 Features	1	4.6 Operating Characteristics	7
2 Applications	1	5 Parameter Measurement Information	8
3 Description	1	5.1 More Product Selection	8
4 Revision History	2	6 Device and Documentation Support	9
4.1 Typical Design Examples	5	6.1 Trademarks	9
4.2 Absolute Maximum Ratings	5	6.2 Electrostatic Discharge Caution	9
4.3 Recommended Operating Conditions	6	6.3 Glossary	9
4.4 Electrical Characteristics	6	7 Mechanical, Packaging, and Orderable	
4.5 Switching Characteristics	7	Information	9

4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (September 2013) to Revision A Page

• Updated V_{CC} values for V_{IH} parameter in the ELECTRICAL CHARACTERISTICS table.	6
--	----------

Changes from Revision A (September 2013) to Revision B Page

• Updated document formatting.	1
-------------------------------------	----------

Function Table

INPUT (Lower Level Input)	OUTPUT (V _{CC} CMOS)
A	Y
H	L
L	H
SUPPLY V _{CC} = 3.3V	
A	Y
V _{IH} (min) = 1.35 V V _{IL} (max) = 0.8 V	V _{OH} (min) = 2.9 V V _{OL} (max) = 0.2 V

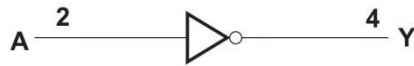


Figure 1. Logic Diagram (NAND Gate)

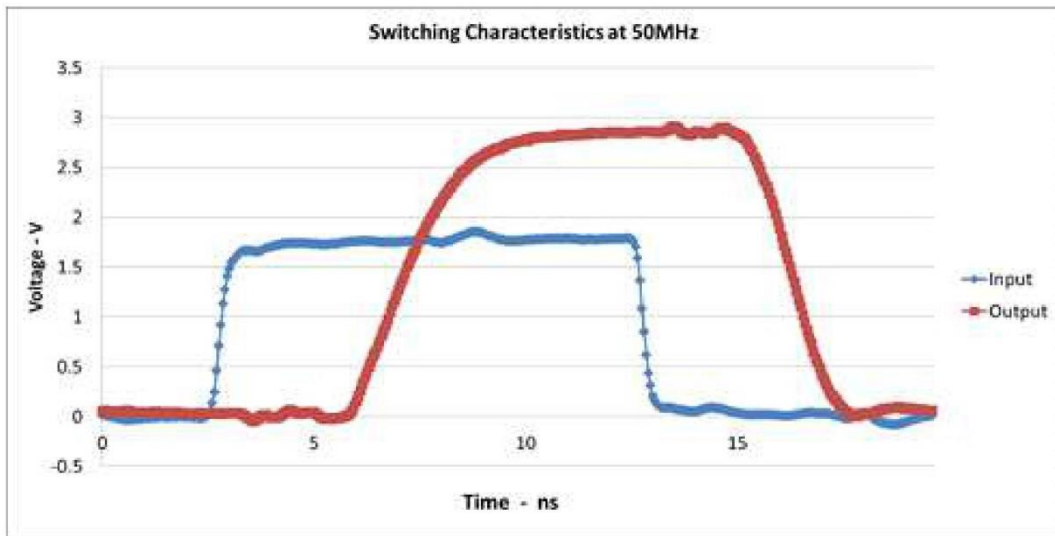


Figure 2. Excellent Signal Integrity (1.8V to 3.3V at 3.3V V_{CC})

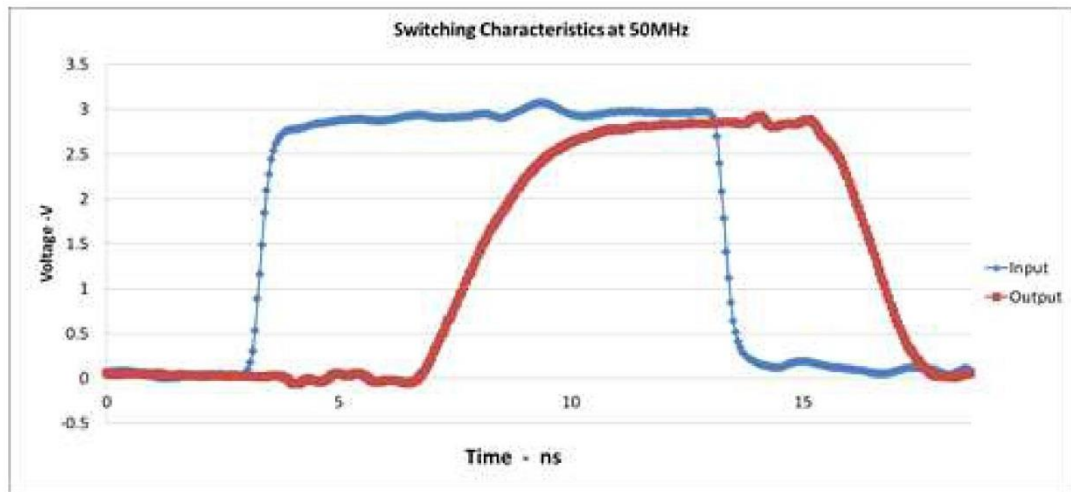


Figure 3. Excellent Signal Integrity (3.3V to 3.3V at 3.3V V_{CC})

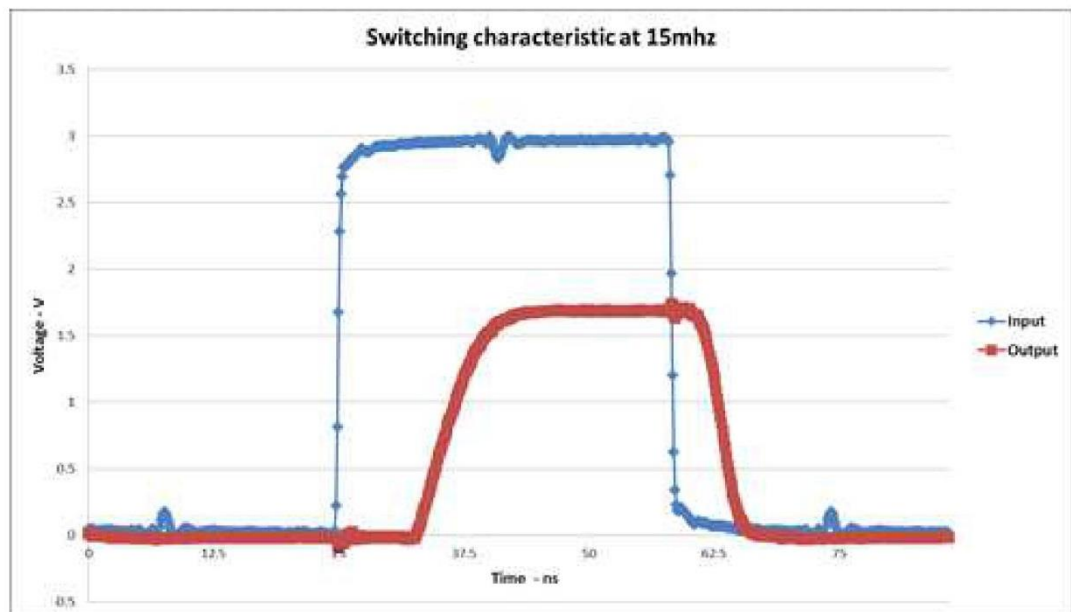


Figure 4. Excellent Signal Integrity (3.3V to 1.8V at 1.8V V_{CC})

4.1 Typical Design Examples

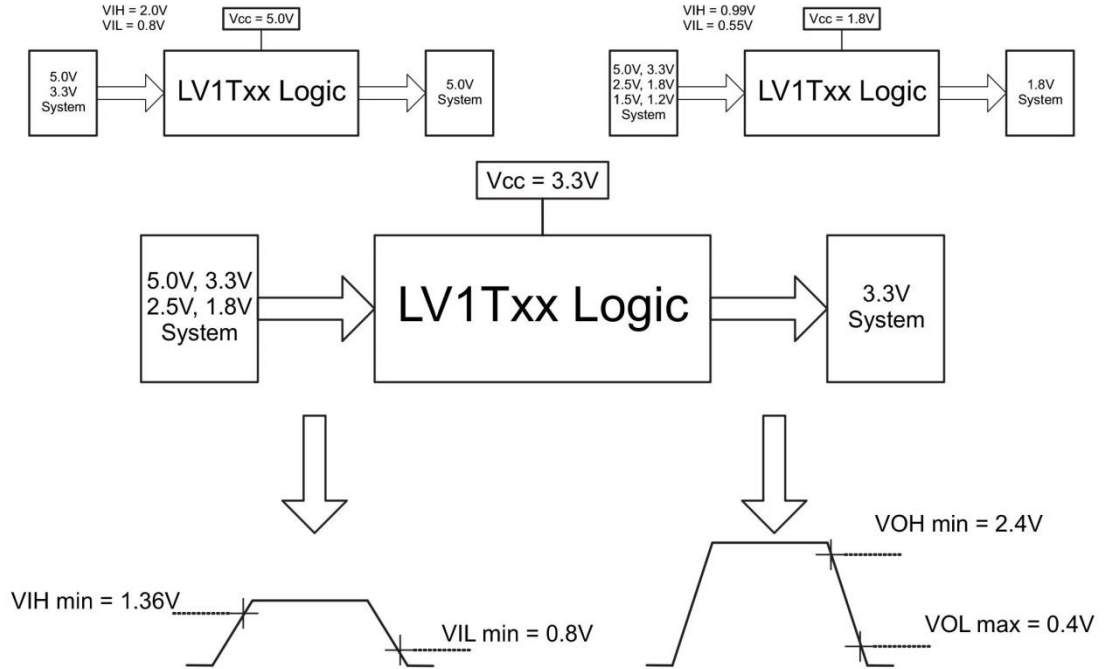


Figure 5. Switching Thresholds for 1.8-V to 3.3-V Translation

4.2 Absolute Maximum Ratings⁽¹⁾

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
V_{CC}	Supply voltage range	-0.5	7.0	V
V_I	Input voltage range ⁽²⁾	-0.5	7.0	V
V_O	Voltage range applied to any output in the high-impedance or power-off state ⁽²⁾	-0.5	4.6	V
	Voltage range applied to any output in the high or low state ⁽²⁾	-0.5	$V_{CC} + 0.5$	V
I_{IK}	Input clamp current	$V_I < 0$	-20	mA
I_{OK}	Output clamp current	$V_O < 0$ or $V_O > V_{CC}$	±20	mA
I_O	Continuous output current		±25	mA
	Continuous current through VCC or GND		±50	mA
θ_{JA}	Package thermal impedance ⁽³⁾	DBV package	206	
		DCK package	252	°C/W
T_{stg}	Storage temperature range	-65	150	°C

(1) Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

(2) The input negative-voltage and output voltage ratings may be exceeded if the input and output current ratings are observed.

(3) The package thermal impedance is calculated in accordance with JESD 51-7.

4.3 Recommended Operating Conditions⁽¹⁾

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT	
V _{CC}	Supply voltage	1.6	5.5	V	
V _I	Input voltage	0	5.5	V	
V _O	Output voltage	0	V _{CC}	V	
I _{OH}	High-level output current	V _{CC} = 1.8 V		-3	mA
		V _{CC} = 2.5 V		-5	
		V _{CC} = 3.3 V		-7	
		V _{CC} = 5.0 V		-8	
I _{OL}	Low-level output current	V _{CC} = 1.8 V		3	mA
		V _{CC} = 2.5 V		5	
		V _{CC} = 3.3 V		7	
		V _{CC} = 5.0 V		8	
Δt/Δv	Input transition rise or fall rate	V _{CC} = 1.8 V		20	ns/V
		V _{CC} = 3.3 V or 2.5V		20	
		V _{CC} = 5.0 V		20	
T _A	Operating free-air temperature	-40	125	°C	

(1) All unused inputs of the device must be held at V_{CC} or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

4.4 Electrical Characteristics

over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	V _{CC}	T _A = 25°C			T _A = -40°C to 125°C		UNIT
			MIN	TYP	MAX	MIN	MAX	
V _{IH}	High-level input voltage	V _{CC} = 1.65 V to 1.8 V	0.94			1.0		V
		V _{CC} = 2.0 V	1.02			1.03		
		V _{CC} = 2.25 V to 2.5 V	1.135			1.18		
		V _{CC} = 2.75 V	1.21			1.23		
		V _{CC} = 3 V to 3.3 V	1.35			1.37		
		V _{CC} = 3.6 V	1.47			1.48		
		V _{CC} = 4.5 V to 5.0 V	2.02			2.03		
V _{IL}	Low-level input voltage	V _{CC} = 1.65 V to 2.0 V				0.58	0.55	V
		V _{CC} = 2.25 V to 2.75 V				0.75	0.71	
		V _{CC} = 3 V to 3.6 V				0.8	0.65	
		V _{CC} = 4.5 V to 5.5 V				0.8	0.8	
V _{OH}	I _{OH} = -20 μA	1.65 V to 5.5 V	V _{CC} - 0.1			V _{CC} - 0.1		V
	I _{OH} = -2.0 mA	1.65 V	1.28			1.21		V
	I _{OH} = -2.3 mA	1.8V	1.5			1.45		V
	I _{OH} = -3 mA	2.3V	2			2		V
	I _{OH} = -3 mA	2.5V	2.25			2.15		V
	I _{OH} = -3.0 mA	3.0 V	2.78			2.7		V
	I _{OH} = -5.5 mA	3.0 V	2.6			2.49		
	I _{OH} = -5.5 mA	3.3 V	2.9			2.8		V
	I _{OH} = -4 mA	4.5 V	4.2			4.1		
	I _{OH} = -8 mA	4.5 V	4.1			3.95		
I _{OH} = -8 mA	5.0 V	4.6			4.5		V	

Electrical Characteristics (continued)

over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	V _{CC}	T _A = 25°C			T _A = -40°C to 125°C		UNIT	
			MIN	TYP	MAX	MIN	MAX		
V _{OL}	I _{OL} = 20 μA	1.65 V to 5.5 V			0.1		0.1	V	
	I _{OL} = 1.9 mA	1.65 V			0.2		0.25		
	I _{OH} = 2.3 mA	2.3V			0.1		0.15		
	I _{OH} = 3 mA				0.15		0.2		
	I _{OL} = 3 mA	3.0 V			0.1		0.15		
	I _{OL} = 5.5 mA				0.2		0.252		
	I _{OL} = 4 mA	4.5 V			0.15		0.2		
I _{OL} = 8 mA				0.3		0.35			
I _I	A input	V _I = 0 V or V _{CC}	0V, 1.8V, 2.5V, 3.3V, 5.5 V		0.12		±1	μA	
I _{CC}	V _I = 0 V or V _{CC} , I _O = 0; open on loading	5.0 V			1		10	μA	
		3.3 V			1		10		
		2.5 V			1		10		
		1.8V			1		10		
ΔI _{CC}	One input at 0.3V or 3.4V, Other inputs at 0 or V _{CC} , I _O = 0	5.5 V			1.35		1.5	mA	
		1.8V			10		10	μA	
C _i	V _I = V _{CC} or GND	3.3 V			2	10	2	10	pF
C _o	V _O = V _{CC} or GND	3.3 V			2.5		2.5		pF

4.5 Switching Characteristics

over recommended operating free-air temperature range (unless otherwise noted) (see Figure 7)

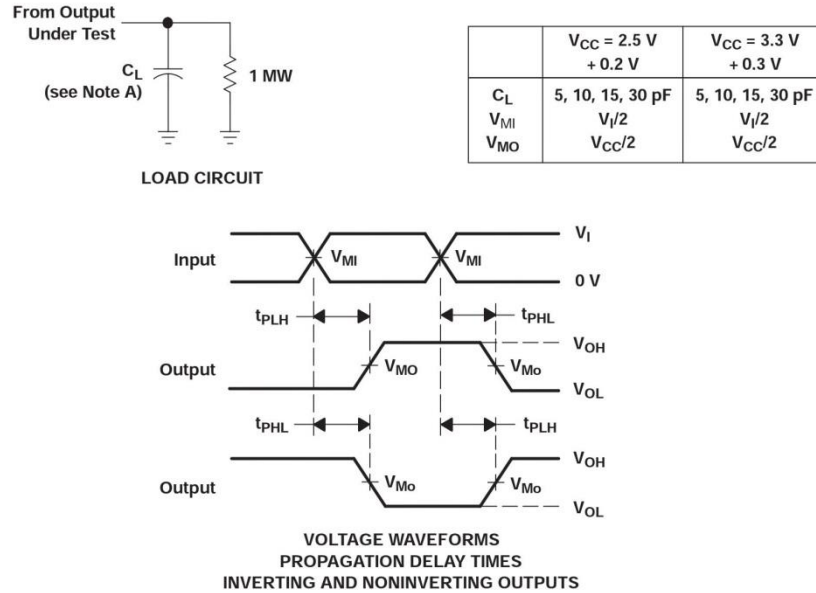
PARAMETER	FROM (INPUT)	TO (OUTPUT)	FREQUENCY (TYP)	V _{CC}	C _L	T _A = 25°C			T _A = -65°C to 125°C			UNIT
						MIN	TYP	MAX	MIN	TYP	MAX	
t _{pd}	Any In	Y	DC to 50 MHz	5.0V	15pF	4	5		4	5	ns	
					30pF	5.5	7.0		5.5	7.0		
				3.3V	15pF	4.8	5		5	5.5	ns	
					30pF	5	5.5		5.5	6.5		
			DC to 25 MHz	2.5V	15pF	6	6.5		7	7.5	ns	
					30pF	6.5	7.5		7.5	8.5		
				DC to 15 MHz	1.8V	15pF	10.5	11		11	12	ns
						30pF	12	13		12	14	

4.6 Operating Characteristics

T_A = 25°C

PARAMETER	TEST CONDITIONS	V _{CC}	TYP	UNIT
C _{pd} Power dissipation capacitance	f = 1 MHz and 10 MHz	1.8 V ± 0.15 V	10	pF
		2.5 V ± 0.2 V	10	
		3.3 V ± 0.3 V	10	
		5.5 V ± 0.5 V	10	

5 Parameter Measurement Information



- NOTES: A. C_L includes probe and jig capacitance.
 B. All input pulses are supplied by generators having the following characteristics: PRR 3 10 MHz, $Z_O = 50 \Omega$, slew rate $\geq 1 \text{ V/ns}$.
 C. The outputs are measured one at a time, with one transition per measurement.
 D. t_{PLH} and t_{PHL} are the same as t_{pd} .

Figure 6. Load Circuit and Voltage Waveforms

5.1 More Product Selection

DEVICE	PACKAGE	DESCRIPTION
SN74LV1T00	DCK, DBV	2-Input Positive-NAND Gate
SN74LV1T02	DCK, DBV	2-Input Positive-NOR Gate
SN74LV1T04	DCK, DBV	Inverter Gate
SN74LV1T08	DCK, DBV	2-Input Positive-AND Gate
SN74LV1T17	DCK, DBV	Single Buffer Gate with 3-state Output
SN74LV1T14	DCK, DBV	Single Schmitt-Trigger Inverter Gate
SN74LV1T32	DCK, DBV	2-Input Positive-OR Gate
SN74LV1T50	DCK, DBV	Single Buffer Gate with 3-state Output
SN74LV1T86	DCK, DBV	Single 2-Input Exclusive-Or Gate
SN74LV1T125	DCK, DBV	Single Buffer Gate with 3-state Output
SN74LV1T126	DCK, DBV	Single Buffer Gate with 3-state Output
SN74LV4T125	RGY, PW	Quadruple Bus Buffer Gate With 3-State Outputs

6 Device and Documentation Support

6.1 Trademarks

All trademarks are the property of their respective owners.

6.2 Electrostatic Discharge Caution



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

6.3 Glossary

[SLYZ022](#) — *TI Glossary*.

This glossary lists and explains terms, acronyms and definitions.

7 Mechanical, Packaging, and Orderable Information

The following packaging information and addendum reflect the most current data available for the designated devices. This data is subject to change without notice and revision of this document.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
SN74LV1T04DBVR	ACTIVE	SOT-23	DBV	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	(NEC3 - NECS)	Samples
SN74LV1T04DCKR	ACTIVE	SC70	DCK	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	(WC3 - WCS)	Samples

⁽¹⁾ The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBsolete: TI has discontinued the production of the device.

⁽²⁾ Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check <http://www.ti.com/productcontent> for the latest availability information and additional product content details.

TBD: The Pb-Free/Green conversion plan has not been defined.

Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.

Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.

Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

⁽³⁾ MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

⁽⁴⁾ There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

⁽⁵⁾ Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "-" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

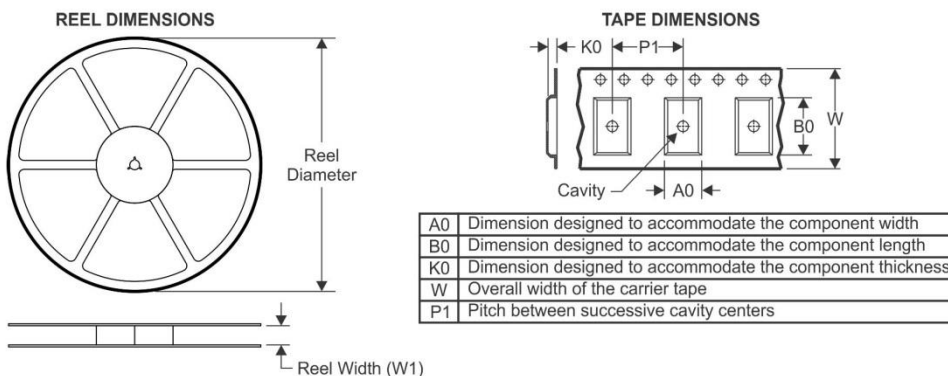
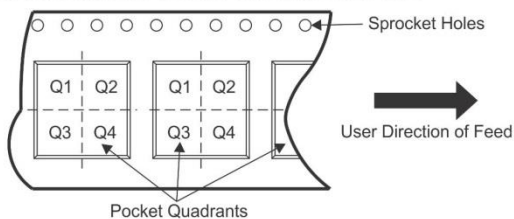
⁽⁶⁾ Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

Addendum-Page 1

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

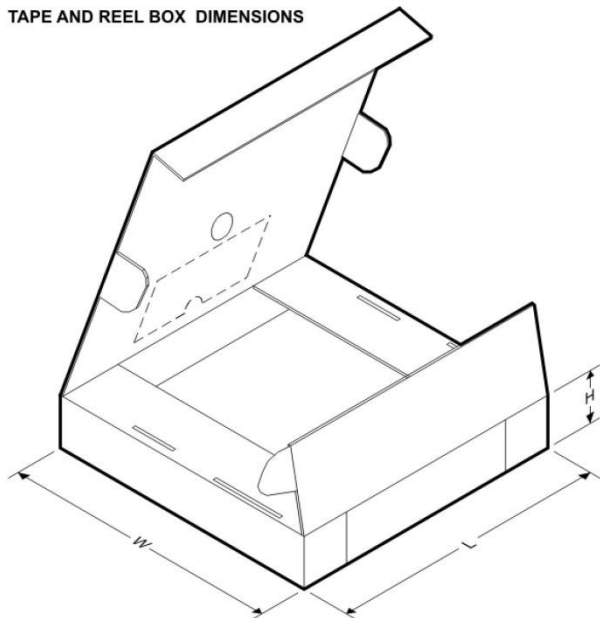
Addendum-Page 2

TAPE AND REEL INFORMATION

QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE


*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
SN74LV1T04DBVR	SOT-23	DBV	5	3000	178.0	9.2	3.3	3.2	1.55	4.0	8.0	Q3
SN74LV1T04DBVR	SOT-23	DBV	5	3000	180.0	8.4	3.23	3.17	1.37	4.0	8.0	Q3
SN74LV1T04DCKR	SC70	DCK	5	3000	180.0	8.4	2.47	2.3	1.25	4.0	8.0	Q3
SN74LV1T04DCKR	SC70	DCK	5	3000	178.0	9.2	2.4	2.4	1.22	4.0	8.0	Q3

TAPE AND REEL BOX DIMENSIONS



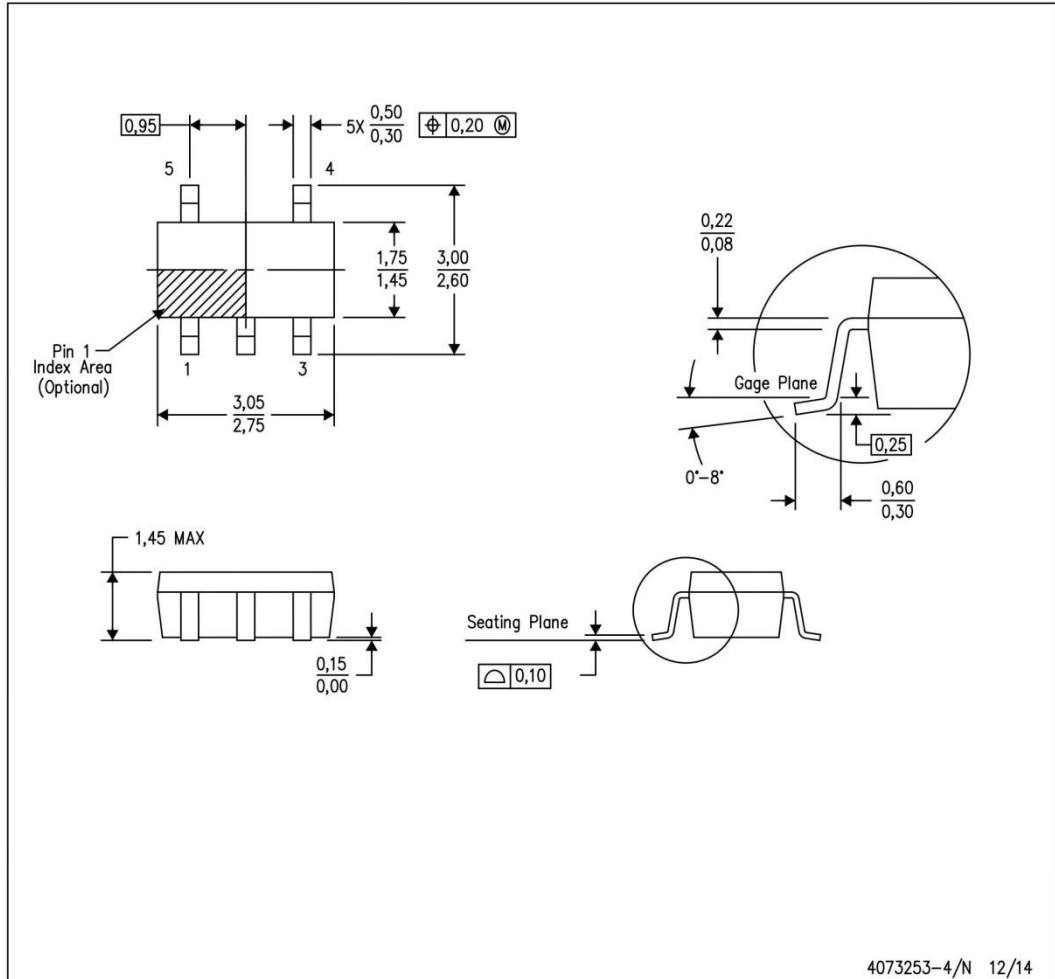
*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
SN74LV1T04DBVR	SOT-23	DBV	5	3000	180.0	180.0	18.0
SN74LV1T04DBVR	SOT-23	DBV	5	3000	202.0	201.0	28.0
SN74LV1T04DCKR	SC70	DCK	5	3000	202.0	201.0	28.0
SN74LV1T04DCKR	SC70	DCK	5	3000	180.0	180.0	18.0

MECHANICAL DATA

DBV (R-PDSO-G5)

PLASTIC SMALL-OUTLINE PACKAGE

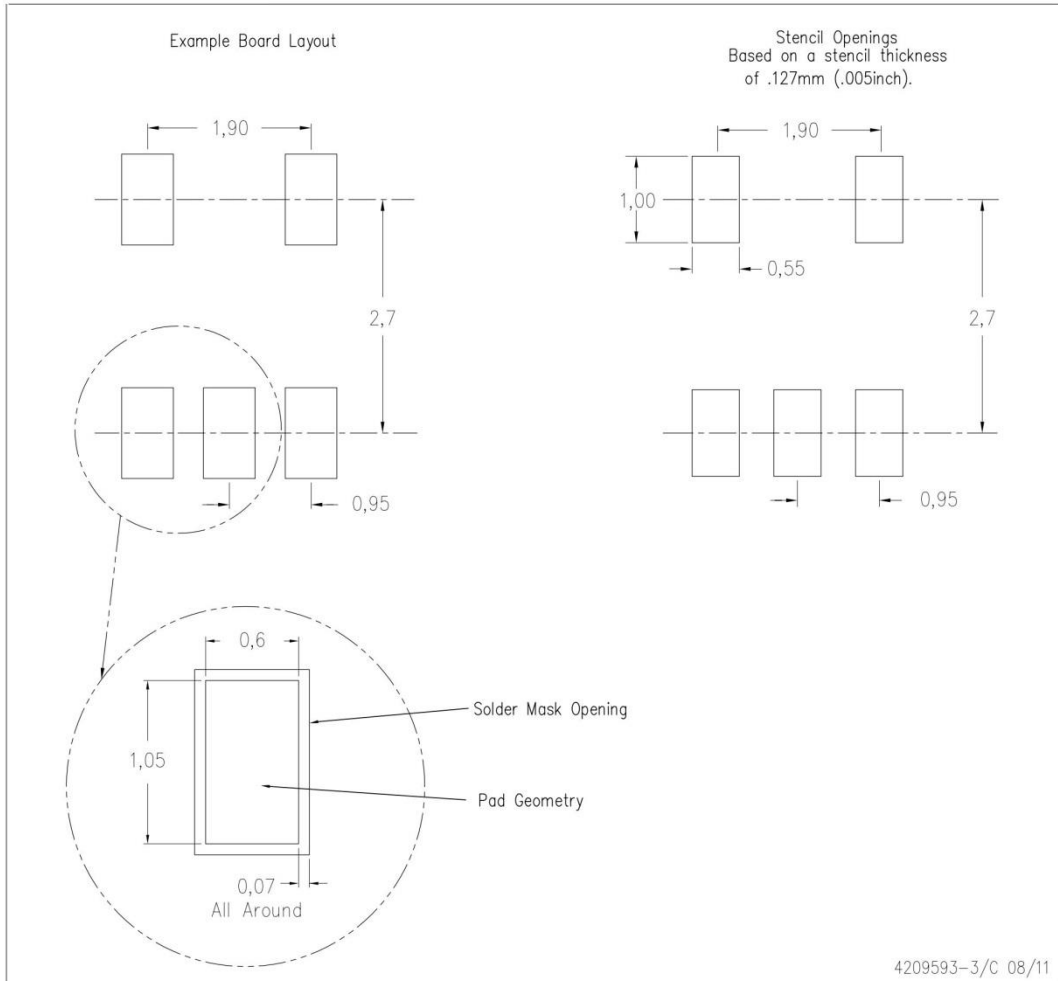


- NOTES:
- A. All linear dimensions are in millimeters.
 - B. This drawing is subject to change without notice.
 - C. Body dimensions do not include mold flash or protrusion. Mold flash and protrusion shall not exceed 0.15 per side.
 - D. Falls within JEDEC MO-178 Variation AA.

LAND PATTERN DATA

DBV (R-PDSO-G5)

PLASTIC SMALL OUTLINE

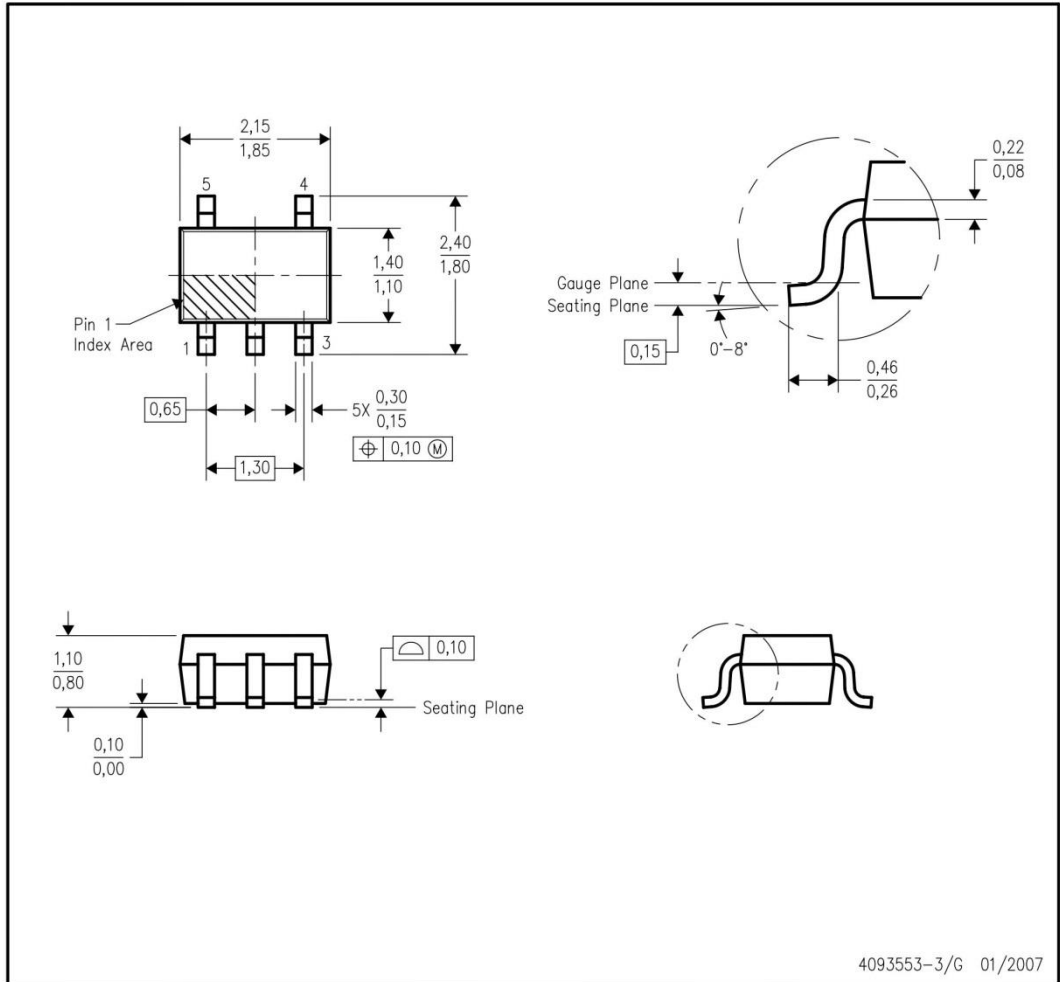


- NOTES:
- All linear dimensions are in millimeters.
 - This drawing is subject to change without notice.
 - Customers should place a note on the circuit board fabrication drawing not to alter the center solder mask defined pad.
 - Publication IPC-7351 is recommended for alternate designs.
 - Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Example stencil design based on a 50% volumetric metal load solder paste. Refer to IPC-7525 for other stencil recommendations.

MECHANICAL DATA

DCK (R-PDSO-G5)

PLASTIC SMALL-OUTLINE PACKAGE

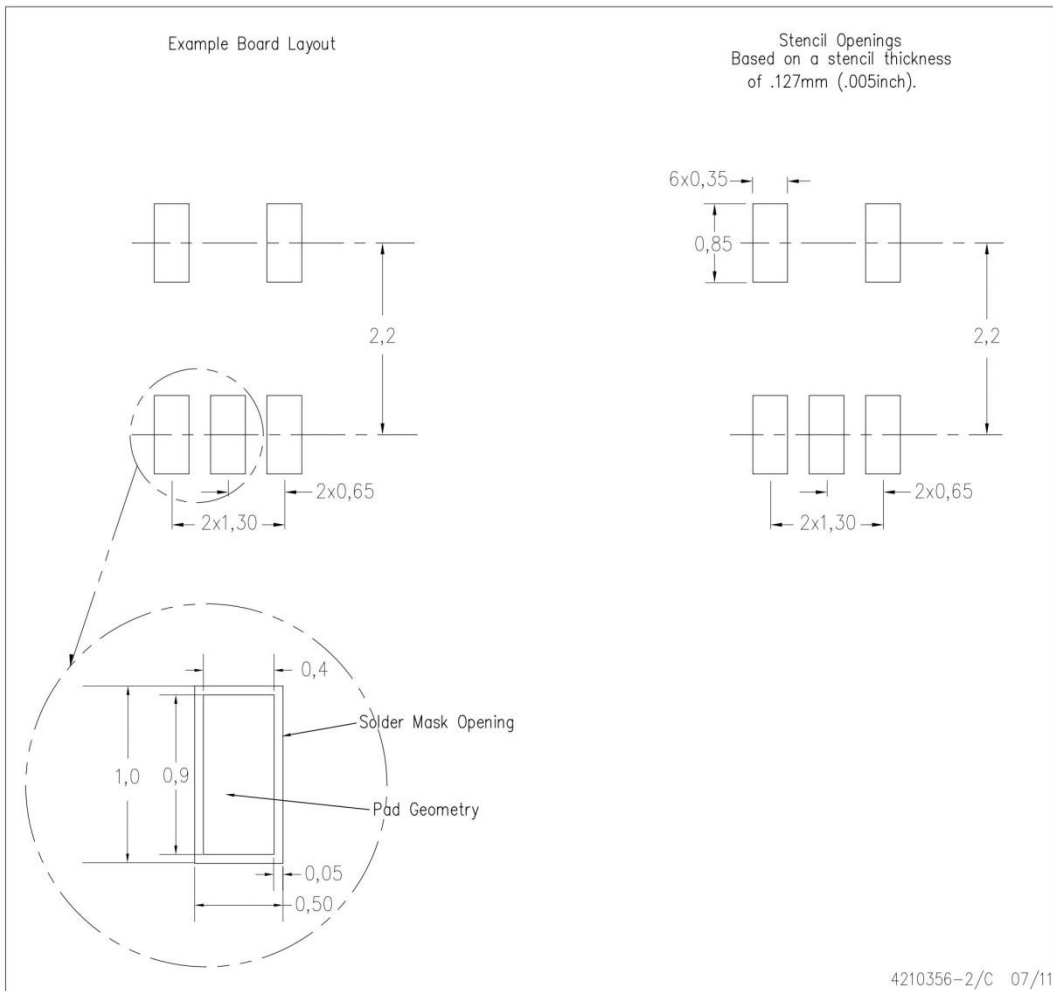


- NOTES:
- A. All linear dimensions are in millimeters.
 - B. This drawing is subject to change without notice.
 - C. Body dimensions do not include mold flash or protrusion. Mold flash and protrusion shall not exceed 0.15 per side.
 - D. Falls within JEDEC MO-203 variation AA.

LAND PATTERN DATA

DCK (R-PDSO-G5)

PLASTIC SMALL OUTLINE



- NOTES:
- All linear dimensions are in millimeters.
 - This drawing is subject to change without notice.
 - Customers should place a note on the circuit board fabrication drawing not to alter the center solder mask defined pad.
 - Publication IPC-7351 is recommended for alternate designs.
 - Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Example stencil design based on a 50% volumetric metal load solder paste. Refer to IPC-7525 for other stencil recommendations.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video
TI E2E Community	e2e.ti.com

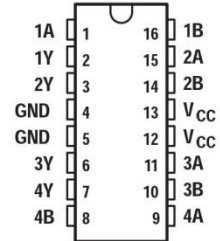
Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2015, Texas Instruments Incorporated

74AC11008 QUADRUPLE 2-INPUT POSITIVE-AND GATE

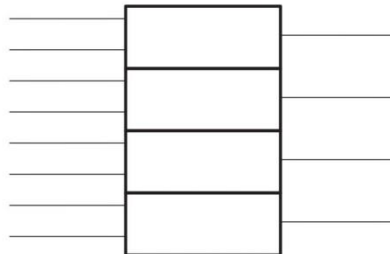
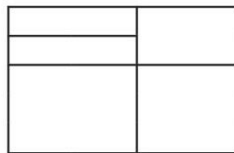
SCAS014C – AUGUST 1987 – REVISED APRIL 1996



D, N, OR PW PACKAGE
(TOP VIEW)



$$= \quad = \quad \overline{\overline{\quad} + \quad}$$



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

EPIC is a trademark of Texas Instruments Incorporated.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

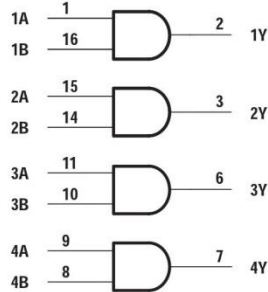


POST OFFICE BOX 655303 DALLAS, TEXAS 75265

74AC11008 QUADRUPLE 2-INPUT POSITIVE-AND GATE

SCAS014C – AUGUST 1987 – REVISED APRIL 1996

logic diagram (positive logic)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage range, V_{CC}	-0.5 V to 7 V	†
Input voltage range, V_I (see Note 1)	-0.5 V to $V_{CC} + 0.5$ V	
Output voltage range, V_O (see Note 1)	-0.5 V to $V_{CC} + 0.5$ V	
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$)	+20 mA	
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$)	+50 mA	
Continuous output current, I_O ($V_O = 0$ to V_{CC})	+50 mA	
Continuous current through V_{CC} or GND	+100 mA	
Maximum power dissipation at $T_A = 55$ °C (in still air) (see Note 2): D package	1.3 W	
..... N package	1.1 W	
..... PW package	0.5 W	
Storage temperature range, T_{stg}	-65 to 150 °C	

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress rating functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES: 1. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.
2. The maximum package power dissipation is calculated using a junction temperature of 150 °C.

 **TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 WOODALL, TEXAS 75265

74AC11008
QUADRUPLE 2-INPUT POSITIVE-AND GATE

SCAS014C – AUGUST 1987 – REVISED APRIL 1996

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	3	5	5.5	V
V_{IH}	High-level input voltage	$V_{CC} = 3\text{ V}$	2.1		V
		$V_{CC} = 4.5\text{ V}$	3.15		
		$V_{CC} = 5.5\text{ V}$	3.85		
V_{IL}	Low-level input voltage	$V_{CC} = 3\text{ V}$		0.9	V
		$V_{CC} = 4.5\text{ V}$		1.35	
		$V_{CC} = 5.5\text{ V}$		1.65	
V_I	Input voltage	0	V_{CC}		V
V_O	Output voltage	0	V_{CC}		V
I_{OH}	High-level output current	$V_{CC} = 3\text{ V}$		-4	mA
		$V_{CC} = 4.5\text{ V}$		-24	
		$V_{CC} = 5.5\text{ V}$		-24	
I_{OL}	Low-level output current	$V_{CC} = 3\text{ V}$		12	mA
		$V_{CC} = 4.5\text{ V}$		24	
		$V_{CC} = 5.5\text{ V}$		24	
Dt/Dv	Input transition rise or fall rate	0	10		ns/V
T_A	Operating free-air temperature	-40		85	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	V_{CC}	$T_A = 25\text{ }^\circ\text{C}$			MIN	MAX	UNIT
			MIN	TYP	MAX			
V_{OH}	$I_{OH} = -50\text{ mA}$	3 V	2.9		2.9		V	
		4.5 V	4.4		4.4			
		5.5 V	5.4		5.4			
	$I_{OH} = -4\text{ mA}$	3 V	2.58		2.48			
		4.5 V	3.94		3.8			
		5.5 V	4.94		4.8			
$I_{OH} = -75\text{ mA}^\dagger$	5.5 V			3.85				
V_{OL}	$I_{OL} = 50\text{ mA}$	3 V		0.1	0.1		V	
		4.5 V		0.1	0.1			
		5.5 V		0.1	0.1			
	$I_{OL} = 12\text{ mA}$	3 V		0.36	0.44			
		4.5 V		0.36	0.44			
		5.5 V		0.36	0.44			
$I_{OL} = 75\text{ mA}^\dagger$	5.5 V			1.65				
I_I	$V_I = V_{CC}$ or GND	5.5 V		+0.1	+1	mA		
I_{CC}	$V_I = V_{CC}$ or GND, $I_O = 0$	5.5 V		4	40	mA		
C_i	$V_I = V_{CC}$ or GND	5 V		3.5		pF		

† Not more than one output should be tested at a time, and the duration of the test should not exceed 10 ms.

74AC11008 QUADRUPLE 2-INPUT POSITIVE-AND GATE

SCAS014C – AUGUST 1987 – REVISED APRIL 1996

switching characteristics over recommended operating free-air temperature range,
 $V_{CC} = 3.3\text{ V} + 0.3\text{ V}$ (unless otherwise noted) (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	$T_A = 25\text{ }^\circ\text{C}$			MIN	MAX	UNIT
			MIN	TYP	MAX			
t_{PLH}	A or B	Y	1.5	6.3	9	1.5	10.2	ns
t_{PHL}			1.5	5.6	7.8	1.5	8.6	

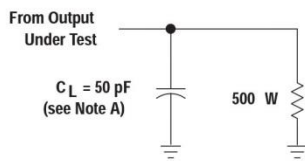
switching characteristics over recommended operating free-air temperature range,
 $V_{CC} = 5\text{ V} + 0.5\text{ V}$ (unless otherwise noted) (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	$T_A = 25\text{ }^\circ\text{C}$			MIN	MAX	UNIT
			MIN	TYP	MAX			
t_{PLH}	A or B	Y	1.5	4.3	6.2	1.5	6.9	ns
t_{PHL}			1.5	5.6	5.9	1.5	6.5	

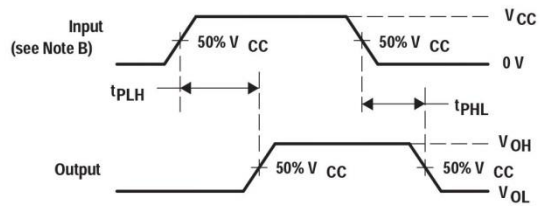
operating characteristics, $V_{CC} = 5\text{ V}, T_A = 25\text{ }^\circ\text{C}$

PARAMETER	TEST CONDITIONS	TYP	UNIT
C_{pd} Power dissipation capacitance per gate	$C_L = 50\text{ pF}, f = 1\text{ MHz}$	29	pF

PARAMETER MEASUREMENT INFORMATION



LOAD CIRCUIT



VOLTAGE WAVEFORMS

NOTES: A. C_L includes probe and jig capacitance.

B. Input pulses are supplied by generators having the following characteristics: PRR

3 1 MHz, $Z_O = 50\text{ } \Omega$, $t_r = 3\text{ ns}$, $t_f = 3\text{ ns}$.

C. The outputs are measured one at a time with one input transition per measurement.

Figure 1. Load Circuit and Voltage Waveforms

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
74AC11008D	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 85	AC11008	Samples
74AC11008DE4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 85	AC11008	Samples
74AC11008N	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type	-40 to 85	74AC11008N	Samples
74AC11008PWLE	OBSOLETE	TSSOP	PW	16		TBD	Call TI	Call TI	-40 to 85		
74AC11008PWR	ACTIVE	TSSOP	PW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 85	AE008	Samples

(1) The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSOLETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check <http://www.ti.com/productcontent> for the latest availability information and additional product content details.

TBD: The Pb-Free/Green conversion plan has not been defined.

Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.

Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.

Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "-" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

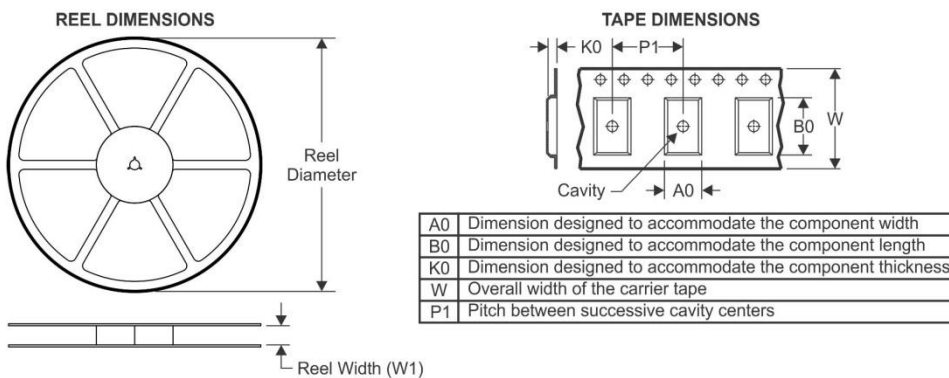
Addendum-Page 1

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

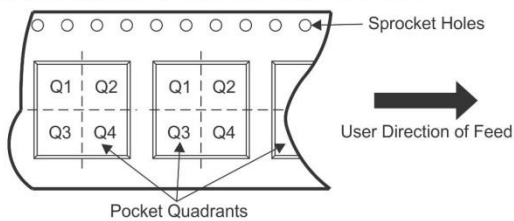
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

Addendum-Page 2

TAPE AND REEL INFORMATION



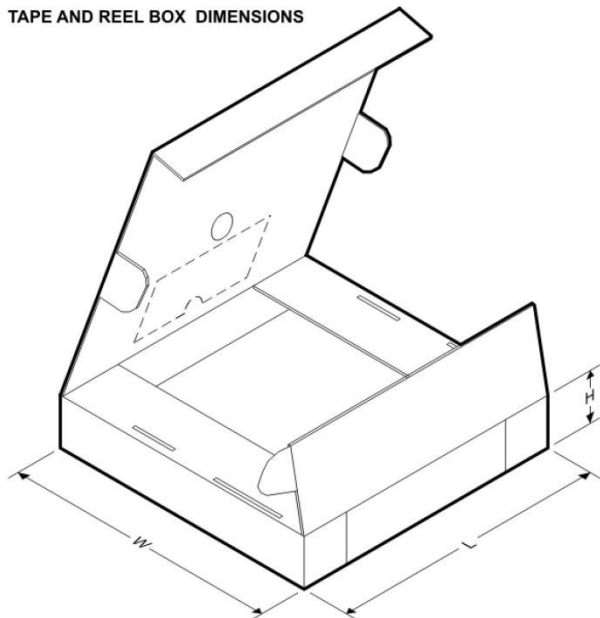
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
74AC11008PWR	TSSOP	PW	16	2000	330.0	12.4	6.9	5.6	1.6	8.0	12.0	Q1

TAPE AND REEL BOX DIMENSIONS



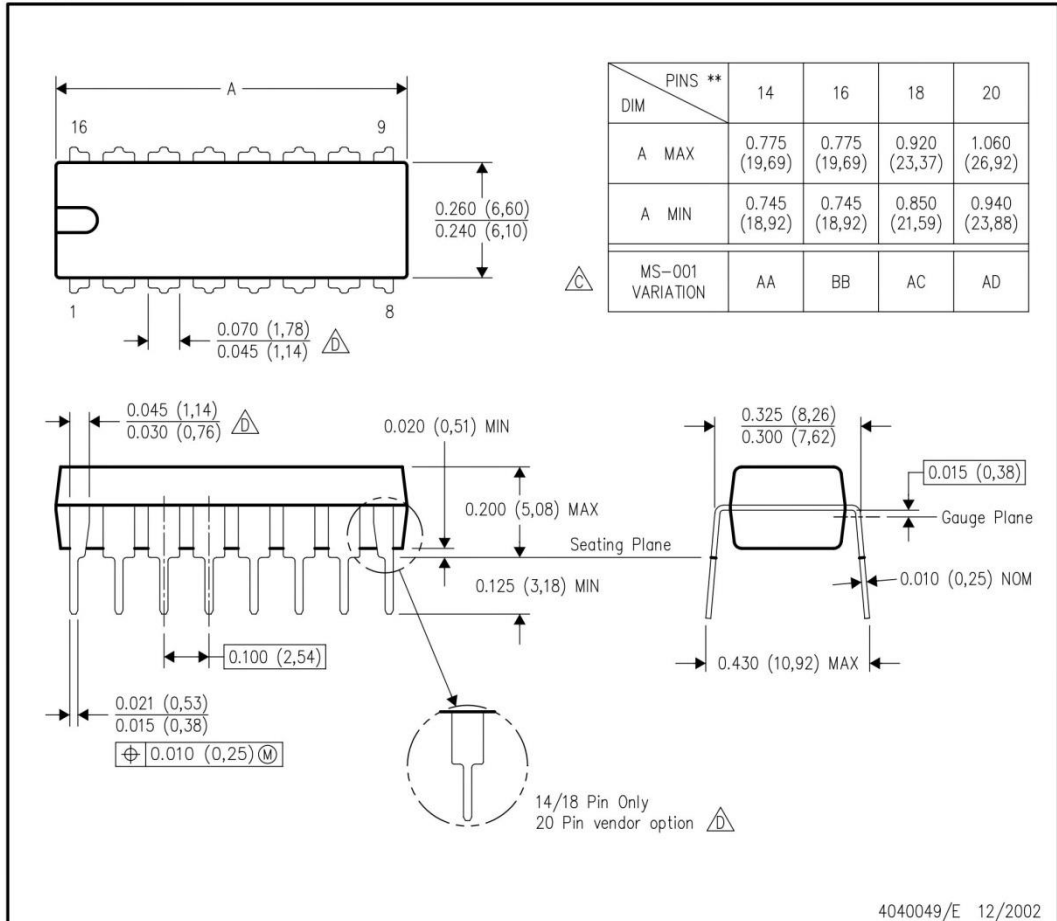
*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
74AC11008PWR	TSSOP	PW	16	2000	367.0	367.0	35.0

MECHANICAL DATA

N (R-PDIP-T**) 16 PINS SHOWN

PLASTIC DUAL-IN-LINE PACKAGE

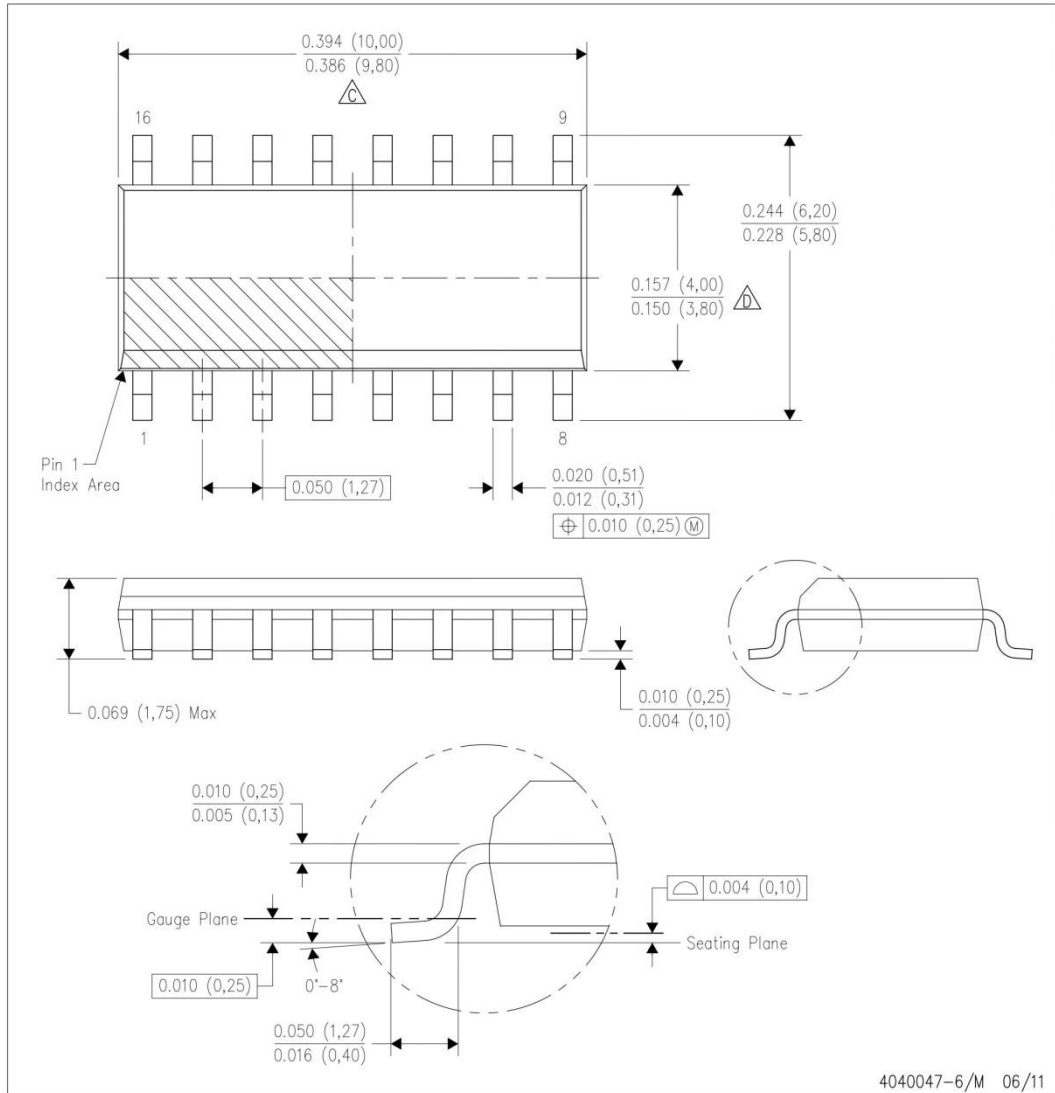


- NOTES:
- A. All linear dimensions are in inches (millimeters).
 - B. This drawing is subject to change without notice.
 - Falls within JEDEC MS-001, except 18 and 20 pin minimum body length (Dim A).
 - The 20 pin end lead shoulder width is a vendor option, either half or full width.



MECHANICAL DATA

D (R-PDSO-G16)

PLASTIC SMALL OUTLINE



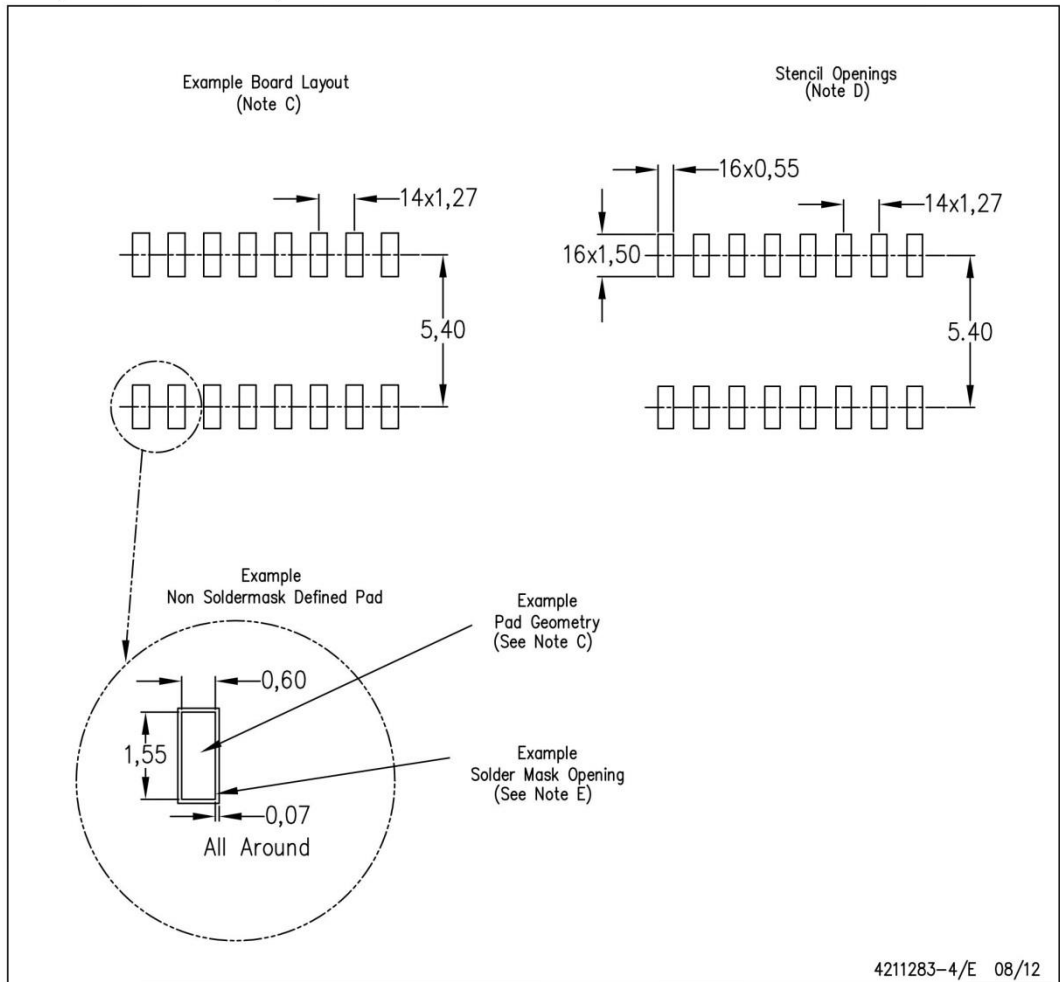
4040047-6/M 06/11

- NOTES:
- A. All linear dimensions are in inches (millimeters).
 - B. This drawing is subject to change without notice.
 -  Body length does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed 0.006 (0,15) each side.
 -  Body width does not include interlead flash. Interlead flash shall not exceed 0.017 (0,43) each side.
 - E. Reference JEDEC MS-012 variation AC.

LAND PATTERN DATA

D (R-PDSO-G16)

PLASTIC SMALL OUTLINE

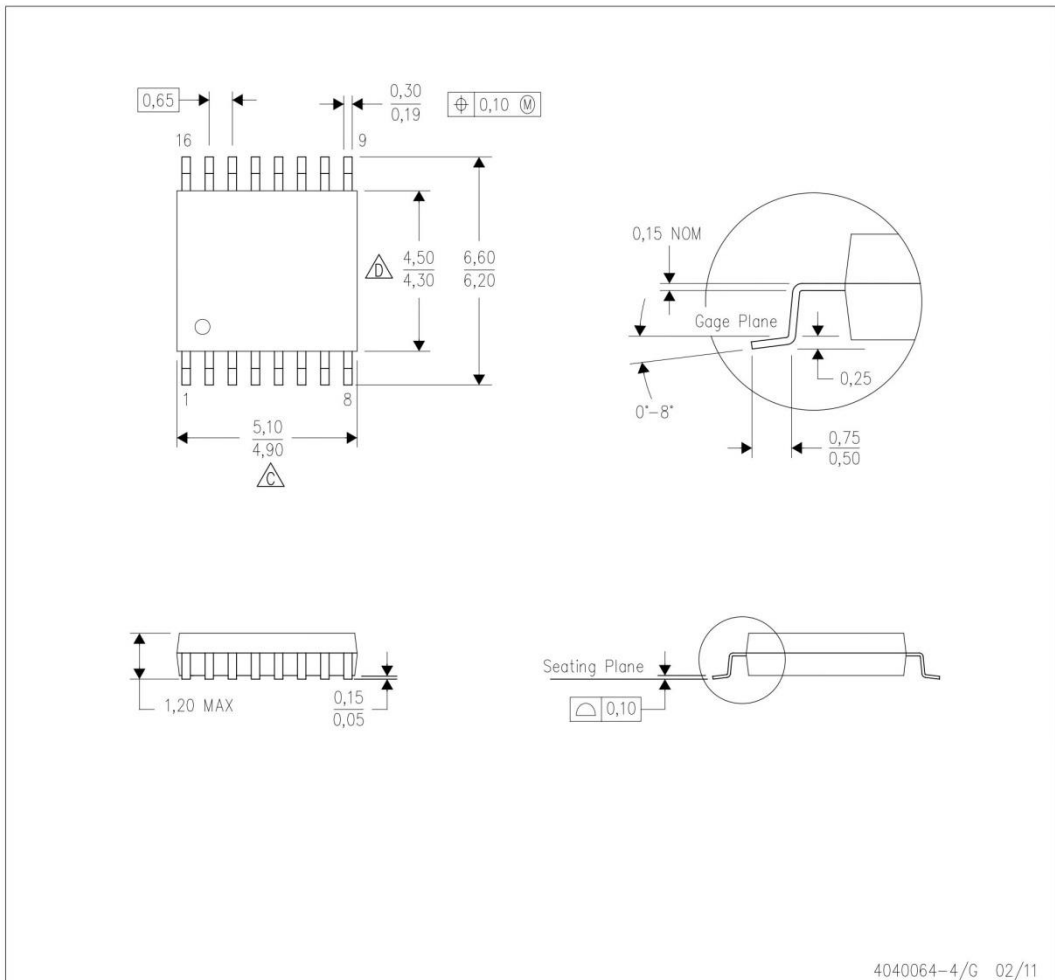




- NOTES:
- All linear dimensions are in millimeters.
 - This drawing is subject to change without notice.
 - Publication IPC-7351 is recommended for alternate designs.
 - Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC-7525 for other stencil recommendations.
 - Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

MECHANICAL DATA

PW (R-PDSO-G16)

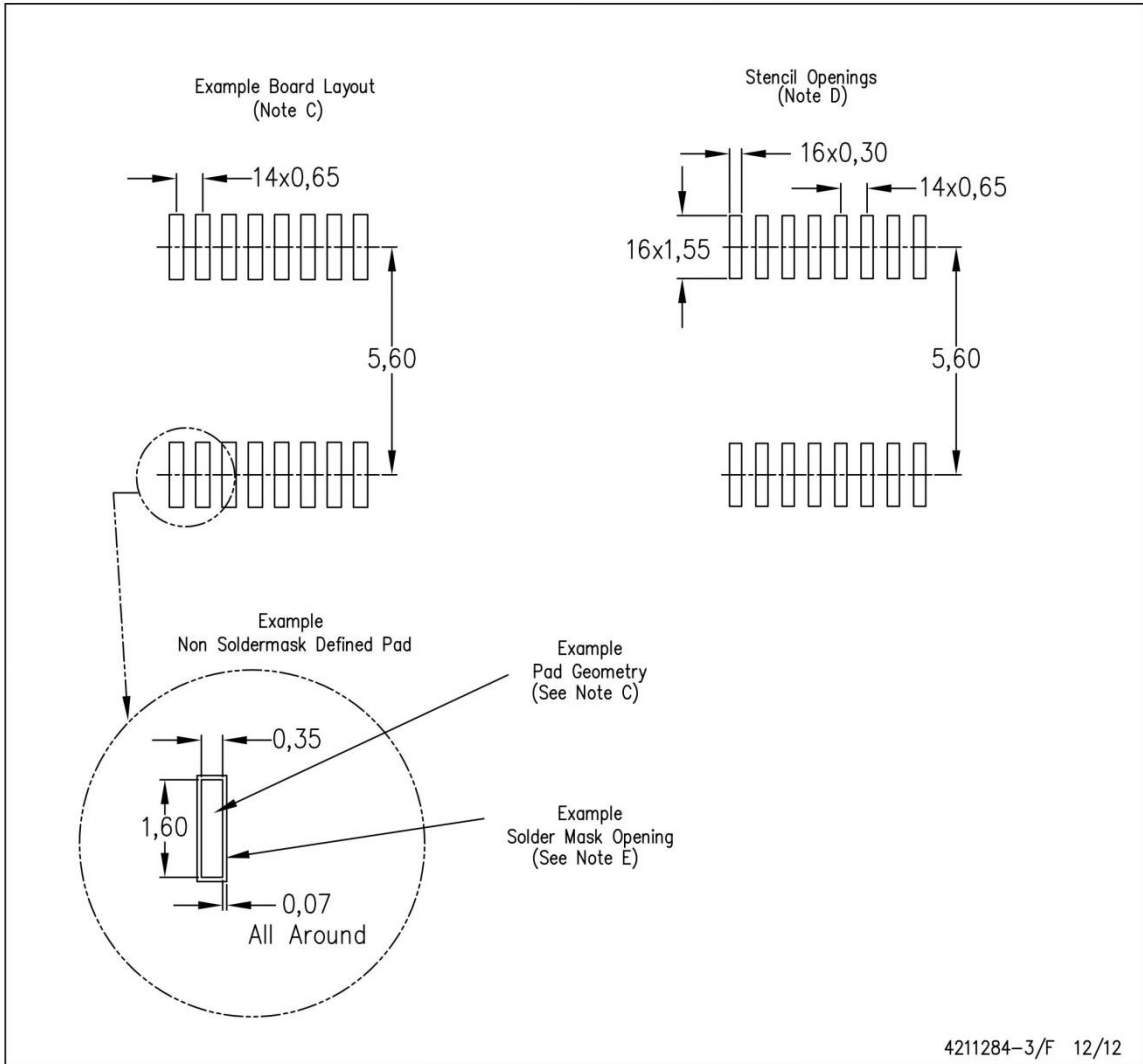
PLASTIC SMALL OUTLINE



- NOTES:
- A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M-1994.
 - B. This drawing is subject to change without notice.
 -  Body length does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed 0,15 each side.
 -  Body width does not include interlead flash. Interlead flash shall not exceed 0,25 each side.
 - E. Falls within JEDEC MO-153

PW (R-PDSO-G16)

PLASTIC SMALL OUTLINE



- NOTES:
- A. All linear dimensions are in millimeters.
 - B. This drawing is subject to change without notice.
 - C. Publication IPC-7351 is recommended for alternate designs.
 - D. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC-7525 for other stencil recommendations.
 - E. Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

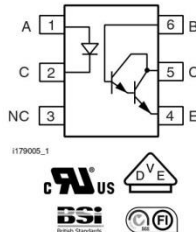
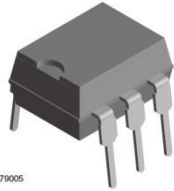
Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com



Optocoupler, Photodarlington Output, High Gain, with Base Connection



FEATURES

- Very high current transfer ratio, 500 % min.
- High isolation resistance, $10^{11} \Omega$ typical
- Standard plastic DIP package
- Compliant to RoHS Directive to 2002/95/EC and in accordance WEEE 2002/96/EC



RoHS COMPLIANT

AGENCY APPROVALS

- UL1577, file no. E52744 system code H
- DIN EN 60747-5-2 (VDE 0884)/DIN EN 60747-5-5 (pending), available with option 1
- BSI IEC60950; IEC60065
- FIMKO

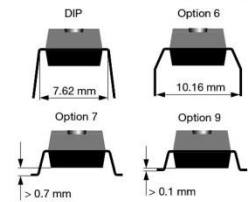
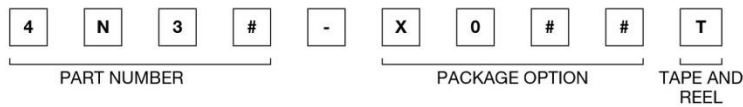
DESCRIPTION

The 4N32 and 4N33 are optically coupled isolators with a gallium arsenide infrared LED and a silicon photodarlington sensor.

Switching can be achieved while maintaining a high degree of isolation between driving and load circuits.

These optocouplers can be used to replace reed and mercury relays with advantages of long life, high speed switching and elimination of magnetic fields.

ORDERING INFORMATION



AGENCY CERTIFIED/PACKAGE	CTR (%)	
	UL, BSI, FIMKO	≥ 500
DIP-6	4N32	4N33
DIP-6, 400 mil, option 6	4N32-X006	-
SMD-6, option 7	4N32-X007T ⁽¹⁾	4N33-X007T ⁽¹⁾
SMD-6, option 9	4N32-X009T ⁽¹⁾	4N33-X009T ⁽¹⁾
VDE, UL, BSI, FIMKO	≥ 500	≥ 500
DIP-6	4N32-X001	4N33-X001
SMD-6, option 7	4N32-X017T	4N33-X017T ⁽¹⁾

Notes

- Additional options may be possible, please contact sales office.
- ⁽¹⁾ Also available in tubes, do not put T on the end.

4N32, 4N33

Vishay Semiconductors

Optocoupler, Photodarlington
Output, High Gain, with Base
Connection



ABSOLUTE MAXIMUM RATINGS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
INPUT				
Reverse voltage		V_R	3	V
Forward current		I_F	60	mA
Power dissipation		P_{diss}	100	mW
Derate linearly	from 55 $^{\circ}\text{C}$		1.33	mW/ $^{\circ}\text{C}$
OUTPUT				
Collector emitter breakdown voltage		BV_{CEO}	30	V
Emitter base breakdown voltage		BV_{EBO}	8	V
Collector base breakdown voltage		BV_{CBO}	50	V
Emitter collector breakdown voltage		BV_{ECO}	5	V
Collector (load) current		I_C	100	mA
Power dissipation		P_{diss}	150	mW
Derate linearly			2	mW/ $^{\circ}\text{C}$
COUPLER				
Total dissipation		P_{tot}	250	mW
Derate linearly			3.3	mW/ $^{\circ}\text{C}$
Isolation test voltage (between emitter	1 s	V_{ISO}	5300	V_{RMS}
Leakage path			7	mm min.
Air path			7	mm min.
Isolation resistance	$V_{IO} = 500\text{ V}$, $T_{amb} = 25\text{ }^{\circ}\text{C}$	R_{IO}	$\geq 10^{12}$	Ω
	$V_{IO} = 500\text{ V}$, $T_{amb} = 100\text{ }^{\circ}\text{C}$	R_{IO}	$\geq 10^{11}$	Ω
Storage temperature		T_{stg}	- 55 to + 150	$^{\circ}\text{C}$
Operating temperature		T_{amb}	- 55 to + 100	$^{\circ}\text{C}$
Lead soldering time ⁽¹⁾	at 260 $^{\circ}\text{C}$		10	s

Notes

- Stresses in excess of the absolute maximum ratings can cause permanent damage to the device. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of this document. Exposure to absolute maximum ratings for extended periods of the time can adversely affect reliability.

⁽¹⁾ Refer to reflow profile for soldering conditions for surface mounted devices (SMD). Refer to wave profile for soldering conditions for through hole devices (DIP).

ELECTRICAL CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
INPUT						
Forward voltage	$I_F = 50\text{ mA}$	V_F		1.25	1.5	V
Reverse current	$V_R = 3\text{ V}$	I_R		0.1	100	μA
Capacitance	$V_R = 0\text{ V}$	C_O		25		pF
OUTPUT						
Collector emitter breakdown voltage ⁽¹⁾	$I_C = 100\text{ }\mu\text{A}$, $I_F = 0$	BV_{CEO}	30			V
Collector base breakdown voltage ⁽¹⁾	$I_C = 100\text{ }\mu\text{A}$, $I_F = 0$	BV_{CBO}	50			V
Emitter base breakdown voltage ⁽¹⁾	$I_C = 100\text{ }\mu\text{A}$, $I_F = 0$	BV_{EBO}	8			V
Emitter collector breakdown voltage ⁽¹⁾	$I_C = 100\text{ }\mu\text{A}$, $I_F = 0$	BV_{ECO}	5	10		V
Collector emitter leakage current	$V_{CE} = 10\text{ V}$, $I_F = 0$	I_{CEO}		1	100	nA
	$I_C = 0.5\text{ mA}$, $V_{CE} = 5\text{ V}$	h_{FE}	13			
COUPLER						
Collector emitter saturation voltage		V_{CEsat}		1		V
Coupling capacitance				1.5		pF

Notes

- Minimum and maximum values are testing requirements. Typical values are characteristics of the device and are the result of engineering evaluation. Typical values are for information only and are not part of the testing requirements.

⁽¹⁾ Indicates JEDEC registered values.



CURRENT TRANSFER RATIO						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Current transfer ratio	$V_{CE} = 10\text{ V}$, $I_F = 10\text{ mA}$	CTR	500			%

SWITCHING CHARACTERISTICS						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Turn-on time	$V_{CC} = 10\text{ V}$, $I_C = 50\text{ mA}$	t_{on}			5	μs
Turn-off time	$I_F = 200\text{ mA}$, $R_L = 180\ \Omega$	t_{off}			100	μs

SAFETY AND INSULATION RATINGS						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Climatic classification (according to IEC 68 part 1)				55/100/21		
Comparative tracking index		CTI	175		399	
V_{IOTM}			8000			V
V_{IORM}			890			V
P_{SO}					700	mW
I_{SI}					400	mA
T_{SI}					175	$^{\circ}\text{C}$
Creepage distance	Standard DIP-6		7			mm
Clearance distance	Standard DIP-6		7			mm
Creepage distance	400 mil DIP-6		8			mm
Clearance distance	400 mil DIP-6		8			mm
Insulation thickness, reinforced rated	per IEC 60950 2.10.5.1		0.4			mm

Note

- As per IEC 60747-5-2, § 7.4.3.8.1, this optocoupler is suitable for "safe electrical insulation" only within the safety ratings. Compliance with the safety ratings shall be ensured by means of protective circuits.

TYPICAL CHARACTERISTICS ($T_{amb} = 25\ ^{\circ}\text{C}$, unless otherwise specified)

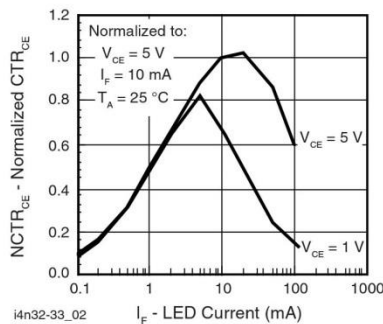


Fig. 1 - Normalized Non-Saturated and Saturated CTR_{CE} vs. LED Current

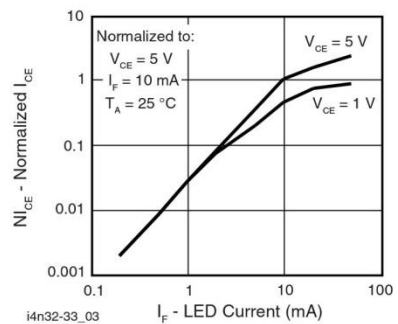


Fig. 2 - Normalized Non-Saturated and Saturated Collector Emitter Current vs. LED Current

4N32, 4N33

Vishay Semiconductors

Optocoupler, Photodarlington
Output, High Gain, with Base
Connection

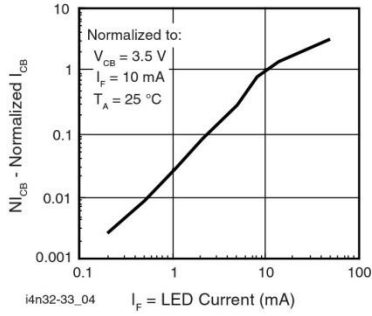


Fig. 3 - Normalized Collector Base Photocurrent vs. LED Current

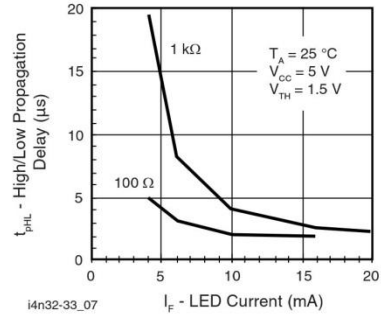


Fig. 6 - High to Low Propagation Delay vs. Collector Load Resistance and LED Current

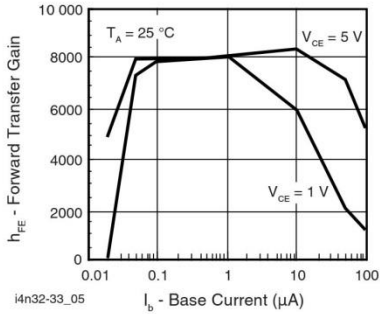


Fig. 4 - Non-Saturated and Saturated h_{FE} vs. Base Current

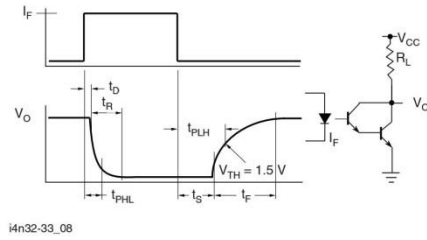


Fig. 7 - Switching Waveform and Switching Schematic

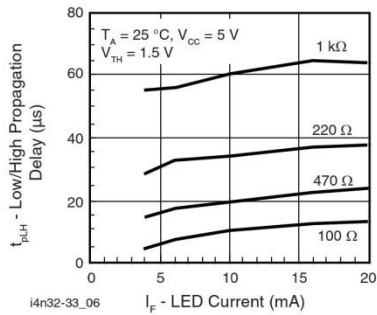


Fig. 5 - Low to High Propagation Delay vs. Collector Load Resistance and LED Current



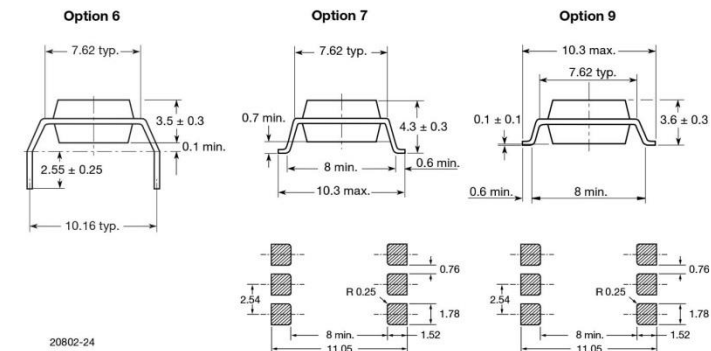
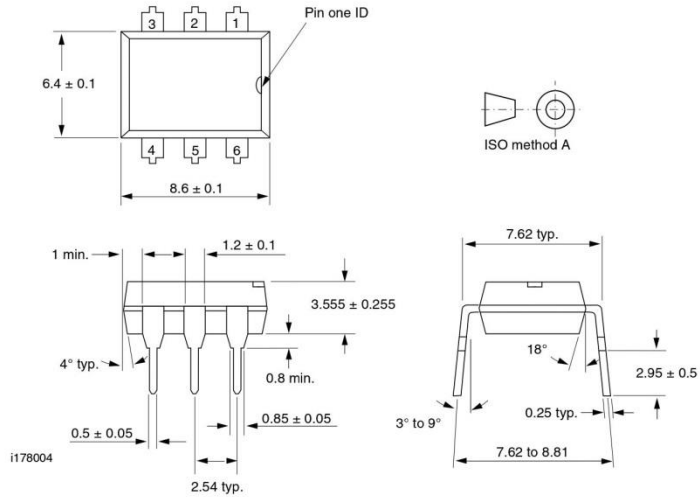
4N32, 4N33

Optocoupler, Photodarlington
Output, High Gain, with Base
Connection

Vishay Semiconductors

PACKAGE DIMENSIONS in millimeters

DIP-6 Package Dimensions



PACKAGE MARKING



Notes

- Example marking for 4N32-X017T.
- Only options 1, and 7 reflected in the package marking.
- The VDE logo is only marked on option 1 parts.
- Tape and reel suffix (T) is not part of the package marking.



Disclaimer

ALL PRODUCT, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, "Vishay"), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Vishay makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Vishay disclaims (i) any and all liability arising out of the application or use of any product, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Vishay's knowledge of typical requirements that are often placed on Vishay products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer's responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application. Parameters provided in datasheets and/or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer's technical experts. Product specifications do not expand or otherwise modify Vishay's terms and conditions of purchase, including but not limited to the warranty expressed therein.

Except as expressly indicated in writing, Vishay products are not designed for use in medical, life-saving, or life-sustaining applications or for any other application in which the failure of the Vishay product could result in personal injury or death. Customers using or selling Vishay products not expressly indicated for use in such applications do so at their own risk. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay. Product names and markings noted herein may be trademarks of their respective owners.

Material Category Policy

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as RoHS-Compliant fulfill the definitions and restrictions defined under Directive 2011/65/EU of The European Parliament and of the Council of June 8, 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (EEE) - recast, unless otherwise specified as non-compliant.

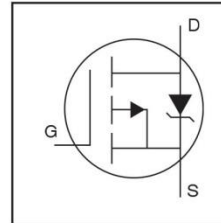
Please note that some Vishay documentation may still make reference to RoHS Directive 2002/95/EC. We confirm that all the products identified as being compliant to Directive 2002/95/EC conform to Directive 2011/65/EU.

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as Halogen-Free follow Halogen-Free requirements as per JEDEC JS709A standards. Please note that some Vishay documentation may still make reference to the IEC 61249-2-21 definition. We confirm that all the products identified as being compliant to IEC 61249-2-21 conform to JEDEC JS709A standards.

IRF4905

HEXFET® Power MOSFET

- Advanced Process Technology
- Ultra Low On-Resistance
- Dynamic dv/dt Rating
- 175°C Operating Temperature
- Fast Switching
- P-Channel
- Fully Avalanche Rated

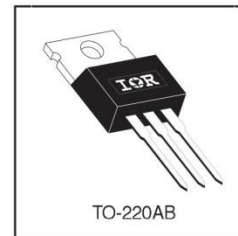


$V_{DSS} = -55V$
$R_{DS(on)} = 0.02\Omega$
$I_D = -74A$

Description

Fifth Generation HEXFETs from International Rectifier utilize advanced processing techniques to achieve extremely low on-resistance per silicon area. This benefit, combined with the fast switching speed and ruggedized device design that HEXFET Power MOSFETs are well known for, provides the designer with an extremely efficient and reliable device for use in a wide variety of applications.

The TO-220 package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 watts. The low thermal resistance and low package cost of the TO-220 contribute to its wide acceptance throughout the industry.



Absolute Maximum Ratings

	Parameter	Max.	Units
$I_D @ T_C = 25^\circ C$	Continuous Drain Current, $V_{GS} @ -10V$	-74	A
$I_D @ T_C = 100^\circ C$	Continuous Drain Current, $V_{GS} @ -10V$	-52	
I_{DM}	Pulsed Drain Current ①	-260	
$P_D @ T_C = 25^\circ C$	Power Dissipation	200	W
	Linear Derating Factor	1.3	W/°C
V_{GS}	Gate-to-Source Voltage	± 20	V
E_{AS}	Single Pulse Avalanche Energy②	930	mJ
I_{AR}	Avalanche Current①	-38	A
E_{AR}	Repetitive Avalanche Energy①	20	mJ
dv/dt	Peak Diode Recovery dv/dt ③	-5.0	V/ns
T_J	Operating Junction and	-55 to + 175	°C
T_{STG}	Storage Temperature Range		
	Soldering Temperature, for 10 seconds	300 (1.6mm from case)	
	Mounting torque, 6-32 or M3 screw	10 lbf•in (1.1N•m)	

Thermal Resistance

	Parameter	Typ.	Max.	Units
$R_{\theta JC}$	Junction-to-Case	---	0.75	°C/W
$R_{\theta CS}$	Case-to-Sink, Flat, Greased Surface	0.50	---	
$R_{\theta JA}$	Junction-to-Ambient	---	62	

IRF4905

International
IOR Rectifier

Electrical Characteristics @ $T_J = 25^\circ\text{C}$ (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions
$V_{(BR)DSS}$	Drain-to-Source Breakdown Voltage	-55	—	—	V	$V_{GS} = 0V, I_D = -250\mu A$
$\Delta V_{(BR)DSS}/\Delta T_J$	Breakdown Voltage Temp. Coefficient	—	-0.05	—	V/ $^\circ\text{C}$	Reference to $25^\circ\text{C}, I_D = -1\text{mA}$
$R_{DS(on)}$	Static Drain-to-Source On-Resistance	—	—	0.02	Ω	$V_{GS} = -10V, I_D = -38A$ ④
$V_{GS(th)}$	Gate Threshold Voltage	-2.0	—	-4.0	V	$V_{DS} = V_{GS}, I_D = -250\mu A$
g_{fs}	Forward Transconductance	21	—	—	S	$V_{DS} = -25V, I_D = -38A$
I_{DSS}	Drain-to-Source Leakage Current	—	—	-25	μA	$V_{DS} = -55V, V_{GS} = 0V$
		—	—	-250	μA	$V_{DS} = -44V, V_{GS} = 0V, T_J = 150^\circ\text{C}$
I_{GSS}	Gate-to-Source Forward Leakage	—	—	100	nA	$V_{GS} = 20V$
	Gate-to-Source Reverse Leakage	—	—	-100	nA	$V_{GS} = -20V$
Q_g	Total Gate Charge	—	—	180	nC	$I_D = -38A$
Q_{gs}	Gate-to-Source Charge	—	—	32	nC	$V_{DS} = -44V$
Q_{gd}	Gate-to-Drain ("Miller") Charge	—	—	86	nC	$V_{GS} = -10V$, See Fig. 6 and 13 ④
$t_{d(on)}$	Turn-On Delay Time	—	18	—	ns	$V_{DD} = -28V$ $I_D = -38A$ $R_G = 2.5\Omega$ $R_D = 0.72\Omega$, See Fig. 10 ④
t_r	Rise Time	—	99	—		
$t_{d(off)}$	Turn-Off Delay Time	—	61	—		
t_f	Fall Time	—	96	—		
L_D	Internal Drain Inductance	—	4.5	—	nH	Between lead, 6mm (0.25in.) from package and center of die contact
L_S	Internal Source Inductance	—	7.5	—		
C_{iss}	Input Capacitance	—	3400	—	pF	$V_{GS} = 0V$ $V_{DS} = -25V$ $f = 1.0\text{MHz}$, See Fig. 5
C_{oss}	Output Capacitance	—	1400	—		
C_{rss}	Reverse Transfer Capacitance	—	640	—		

Source-Drain Ratings and Characteristics

	Parameter	Min.	Typ.	Max.	Units	Conditions
I_S	Continuous Source Current (Body Diode)	—	—	-74	A	MOSFET symbol showing the integral reverse p-n junction diode.
I_{SM}	Pulsed Source Current (Body Diode) ①	—	—	-260		
V_{SD}	Diode Forward Voltage	—	—	-1.6	V	$T_J = 25^\circ\text{C}, I_S = -38A, V_{GS} = 0V$ ②
t_{rr}	Reverse Recovery Time	—	89	130	ns	$T_J = 25^\circ\text{C}, I_F = -38A$
Q_{rr}	Reverse Recovery Charge	—	230	350	nC	$di/dt = -100A/\mu s$ ③
t_{on}	Forward Turn-On Time	Intrinsic turn-on time is negligible (turn-on is dominated by L_S+L_D)				

Notes:

- ① Repetitive rating; pulse width limited by max. junction temperature. (See fig. 11)
- ② Starting $T_J = 25^\circ\text{C}$, $L = 1.3\text{mH}$
 $R_G = 25\Omega$, $I_{AS} = -38A$. (See Figure 12)
- ③ $I_{SD} \leq -38A$, $di/dt \leq -270A/\mu s$, $V_{DD} \leq V_{(BR)DSS}$,
 $T_J \leq 175^\circ\text{C}$
- ④ Pulse width $\leq 300\mu s$; duty cycle $\leq 2\%$.

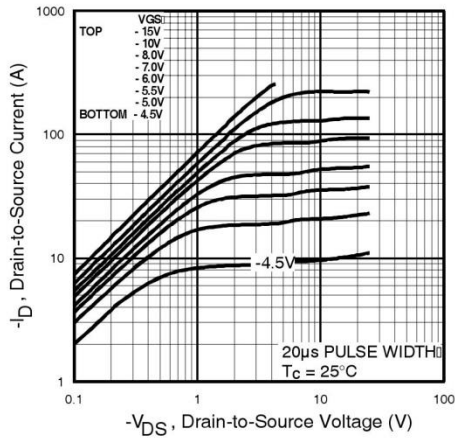


Fig 1. Typical Output Characteristics

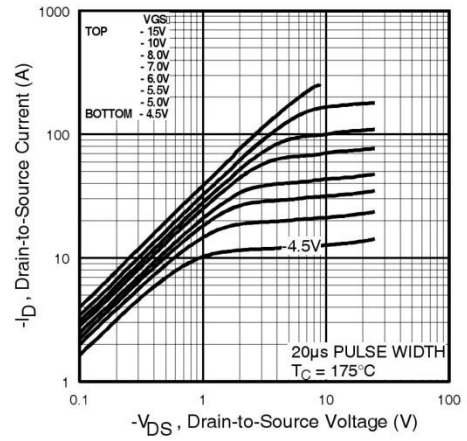


Fig 2. Typical Output Characteristics

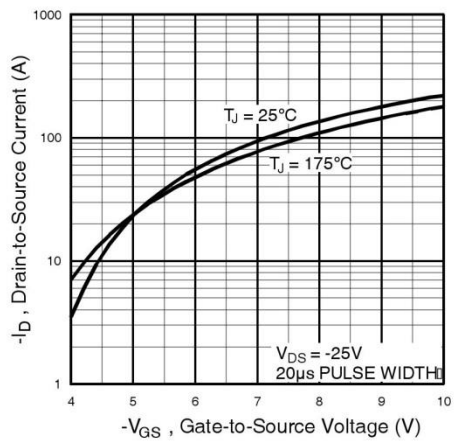


Fig 3. Typical Transfer Characteristics

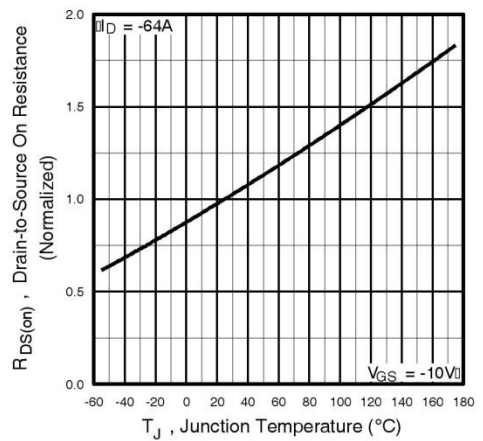


Fig 4. Normalized On-Resistance Vs. Temperature

IRF4905

International
IR Rectifier

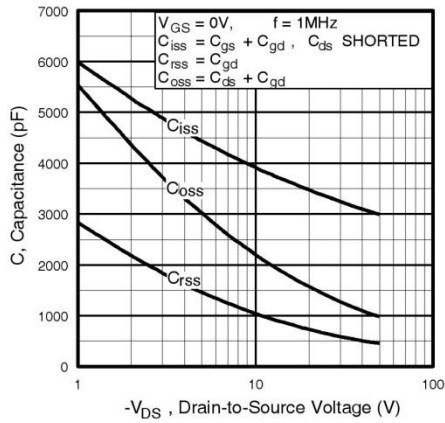


Fig 5. Typical Capacitance Vs. Drain-to-Source Voltage

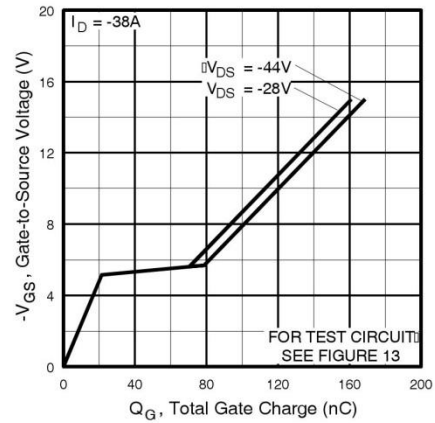


Fig 6. Typical Gate Charge Vs. Gate-to-Source Voltage

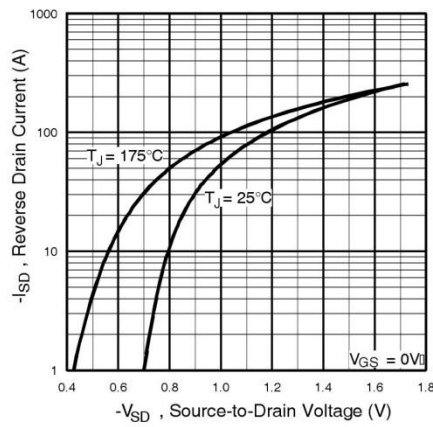


Fig 7. Typical Source-Drain Diode Forward Voltage

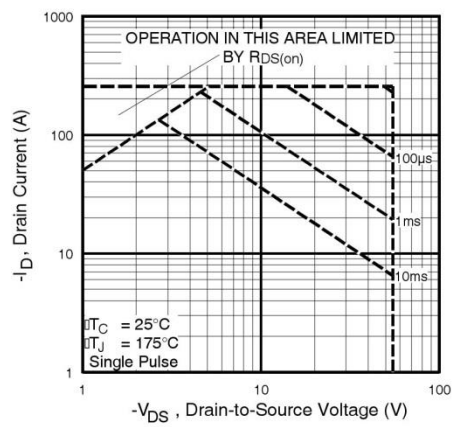


Fig 8. Maximum Safe Operating Area

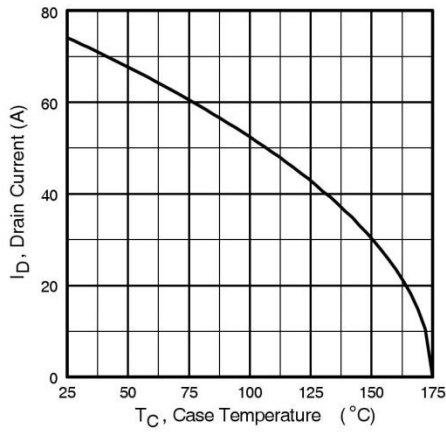


Fig 9. Maximum Drain Current Vs. Case Temperature

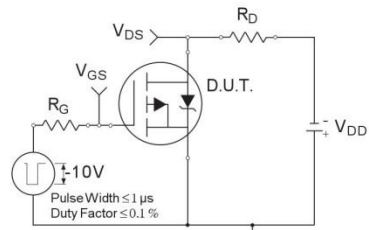


Fig 10a. Switching Time Test Circuit

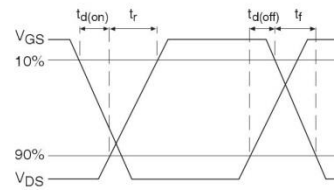


Fig 10b. Switching Time Waveforms

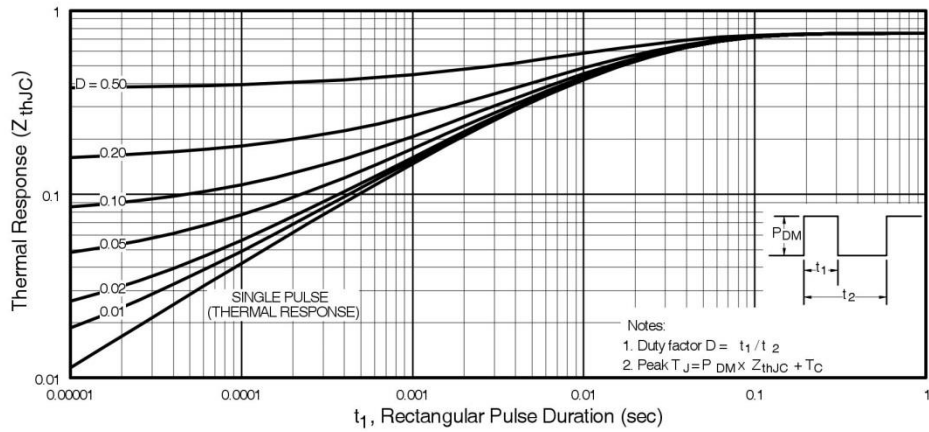


Fig 11. Maximum Effective Transient Thermal Impedance, Junction-to-Case

IRF4905

International
IGOR Rectifier

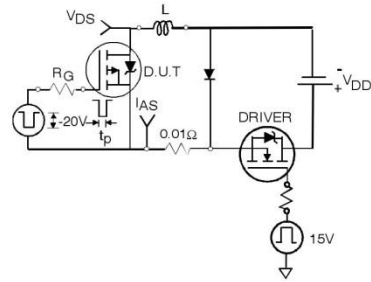


Fig 12a. Unclamped Inductive Test Circuit

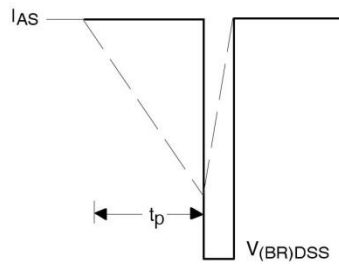


Fig 12b. Unclamped Inductive Waveforms

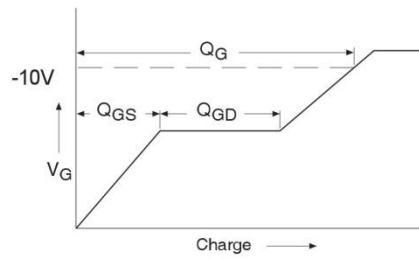


Fig 13a. Basic Gate Charge Waveform

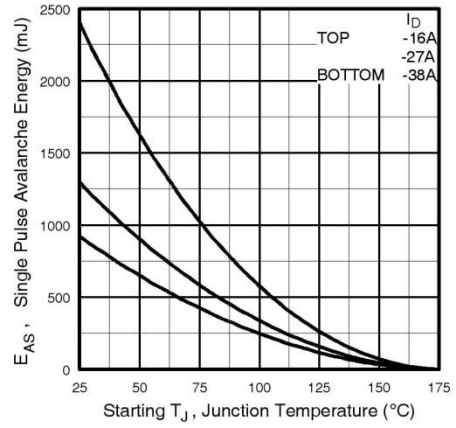


Fig 12c. Maximum Avalanche Energy Vs. Drain Current

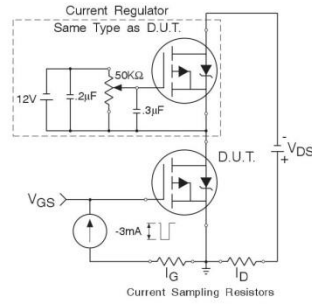
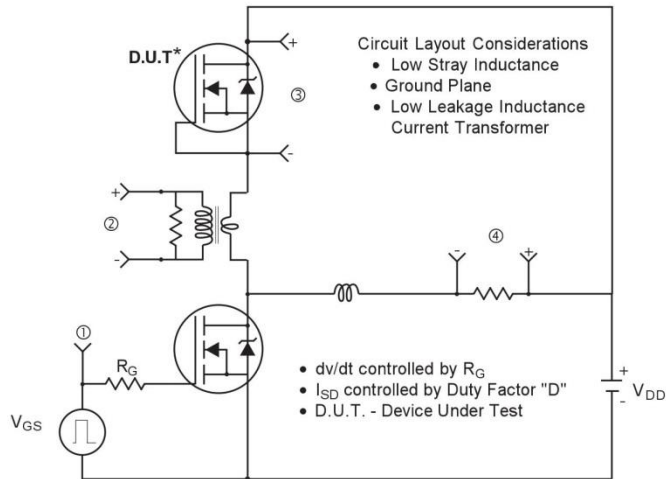
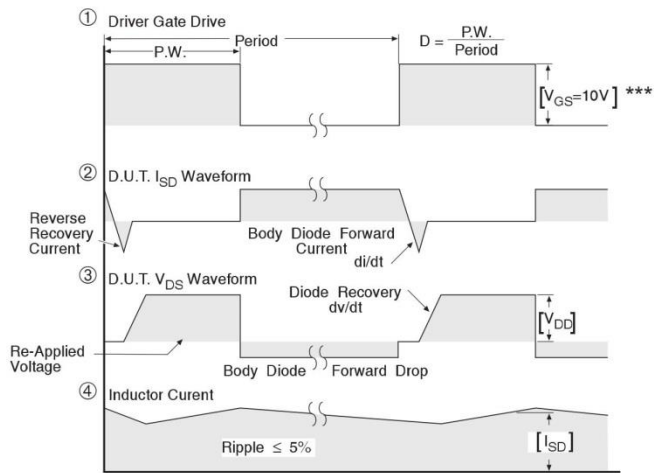


Fig 13b. Gate Charge Test Circuit

Peak Diode Recovery dv/dt Test Circuit



* Reverse Polarity of D.U.T for P-Channel



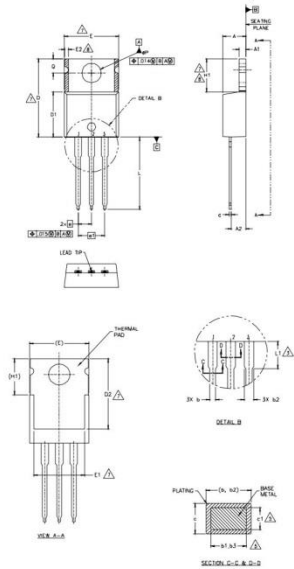
*** $V_{GS} = 5.0V$ for Logic Level and 3V Drive Devices

Fig 14. For P-Channel HEXFETS

IRF4905

TO-220AB Package Outline

Dimensions are shown in millimeters (inches)



- NOTES:
- 1- DIMENSIONING AND TOLERANCING AS PER ASME Y14.5 M- 1994.
 - 2- DIMENSIONS ARE SHOWN IN INCHES (MILLIMETERS).
 - 3- LEAD DIMENSION AND FINISH UNCONTROLLED IN L1.
 - 4- DIMENSION D, D1 & E DO NOT INCLUDE MOLD FLASH. MOLD FLASH SHALL NOT EXCEED .005" (0.127) PER SIDE. THESE DIMENSIONS ARE MEASURED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY.
 - 5- DIMENSION H1, D3 & E1 APPLY TO BASE METAL ONLY. CONTROLLING DIMENSION : INCHES.
 - 7- THERMAL PAD CONTOUR OPTIONAL WITHIN DIMENSIONS E,H1,D2 & E1
 - 8- DIMENSION E2 X H1 DEFINE A ZONE WHERE STAMPING AND SINGULATION IRREGULARITIES ARE ALLOWED.
 - 9- OUTLINE CONFORMS TO JEDEC TO-220, EXCEPT A2 (max.) AND D2 (min.) WHERE DIMENSIONS ARE DERIVED FROM THE ACTUAL PACKAGE OUTLINE.

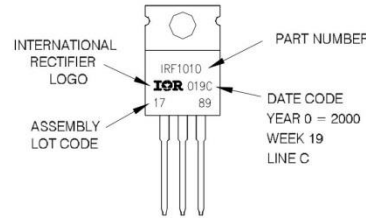
SYMBOL	DIMENSIONS				NOTES
	MILLIMETERS		INCHES		
	MIN.	MAX.	MIN.	MAX.	
A	3.56	4.83	.140	.190	
A1	0.51	1.40	.020	.055	
A2	2.03	2.92	.080	.115	
b	0.38	1.01	.015	.040	5
b1	0.38	0.97	.015	.038	
b2	1.14	1.78	.045	.070	
b3	1.14	1.73	.045	.068	5
c	0.36	0.61	.014	.024	
c1	0.36	0.56	.014	.022	5
D	14.22	16.51	.560	.650	4
D1	8.38	9.02	.330	.355	
D2	11.68	12.88	.460	.507	7
E	9.65	10.67	.380	.420	4,7
E1	6.86	8.89	.270	.350	7
E2	-	0.76	-	.030	8
e	7.54 BSC	-	.300 BSC	-	
e1	6.08 BSC	-	.240 BSC	-	
H1	0.84	6.86	.230	.270	7,8
L	12.70	14.73	.500	.580	
L1	-	6.35	-	.250	3
pp	3.54	4.08	.139	.161	
Q	2.54	3.42	.100	.135	

- LEAD ASSIGNMENTS
- 1- GATE
 - 2- DRAIN
 - 3- SOURCE
- DIODES
- 1- ANODE
 - 2- CATHODE
 - 3- ANODE

TO-220AB Part Marking Information

EXAMPLE: THIS IS AN IRF1010
 LOT CODE 1789
 ASSEMBLED ON VV 19, 2000
 IN THE ASSEMBLY LINE 'C'

Note: 'P' in assembly line position indicates 'Lead - Free'



TO-220AB packages are not recommended for Surface Mount Application.

Note: For the most current drawing please refer to IR website at <http://www.irf.com/package/pkhexfet.html>

Data and specifications subject to change without notice. Qualification Standards can be found on IR's Web site



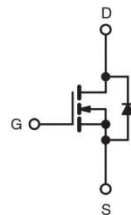
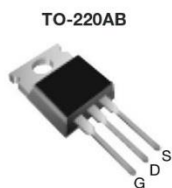
IR WORLD HEADQUARTERS: 233 Kansas St., El Segundo, California 90245, USA Tel: (310) 252-7105
 TAC Fax: (310) 252-7903
 Visit us at www.irf.com for sales contact information. 05/07

Note: For the most current drawings please refer to the IR website at:
<http://www.irf.com/package/>



Power MOSFET

PRODUCT SUMMARY		
V_{DS} (V)	100	
$R_{DS(on)}$ (Ω)	$V_{GS} = 10\text{ V}$	0.16
Q_g (Max.) (nC)	26	
Q_{gs} (nC)	5.5	
Q_{gd} (nC)	11	
Configuration	Single	



N-Channel MOSFET

FEATURES

- Dynamic dV/dt Rating
- Repetitive Avalanche Rated
- 175 °C Operating Temperature
- Fast Switching
- Ease of Paralleling
- Simple Drive Requirements
- Compliant to RoHS Directive 2002/95/EC



DESCRIPTION

Third generation Power MOSFETs from Vishay provide the designer with the best combination of fast switching, ruggedized device design, low on-resistance and cost-effectiveness.

The TO-220AB package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 W. The low thermal resistance and low package cost of the TO-220AB contribute to its wide acceptance throughout the industry.

ORDERING INFORMATION	
Package	TO-220AB
Lead (Pb)-free	IRF530PbF SiHF530-E3
SnPb	IRF530 SiHF530

ABSOLUTE MAXIMUM RATINGS ($T_C = 25\text{ }^\circ\text{C}$, unless otherwise noted)					
PARAMETER	SYMBOL		LIMIT	UNIT	
Drain-Source Voltage	V_{DS}		100	V	
Gate-Source Voltage	V_{GS}		± 20		
Continuous Drain Current	V_{GS} at 10 V	$T_C = 25\text{ }^\circ\text{C}$	14	A	
		$T_C = 100\text{ }^\circ\text{C}$	10		
Pulsed Drain Current ^a	I_{DM}		56		
Linear Derating Factor			0.59	W/ $^\circ\text{C}$	
Single Pulse Avalanche Energy ^b	E_{AS}		69	mJ	
Repetitive Avalanche Current ^a	I_{AR}		14	A	
Repetitive Avalanche Energy ^a	E_{AR}		8.8	mJ	
Maximum Power Dissipation	$T_C = 25\text{ }^\circ\text{C}$		P_D	88	W
Peak Diode Recovery dV/dt ^c			dV/dt	5.5	V/ns
Operating Junction and Storage Temperature Range	T_J, T_{stg}		- 55 to + 175	$^\circ\text{C}$	
Soldering Recommendations (Peak Temperature)	for 10 s		300 ^d		
Mounting Torque	6-32 or M3 screw		10	lbf · in	
			1.1	N · m	

Notes

- Repetitive rating; pulse width limited by maximum junction temperature (see fig. 11).
- $V_{DD} = 25\text{ V}$, starting $T_J = 25\text{ }^\circ\text{C}$, $L = 528\text{ }\mu\text{H}$, $R_g = 25\text{ }\Omega$, $I_{AS} = 14\text{ A}$ (see fig. 12).
- $I_{SD} \leq 14\text{ A}$, $dI/dt \leq 140\text{ A}/\mu\text{s}$, $V_{DD} \leq V_{DS}$, $T_J \leq 175\text{ }^\circ\text{C}$.
- 1.6 mm from case.



* Pb containing terminations are not RoHS compliant, exemptions may apply

IRF530, SiHF530

Vishay Siliconix

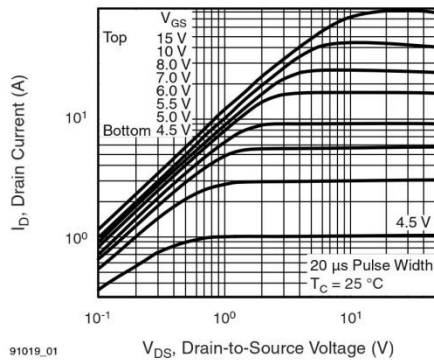
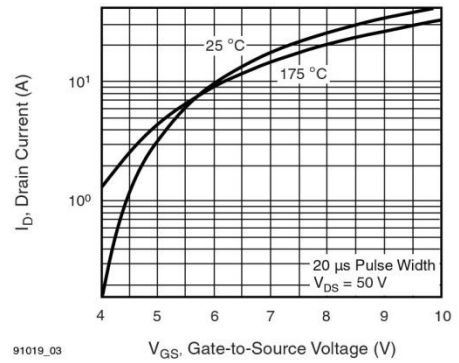
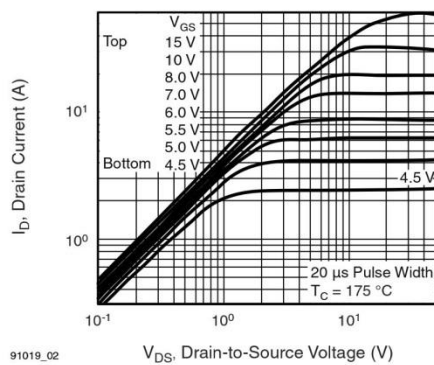
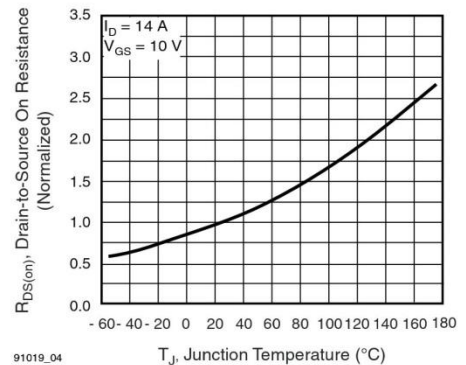


THERMAL RESISTANCE RATINGS				
PARAMETER	SYMBOL	TYP.	MAX.	UNIT
Maximum Junction-to-Ambient	R_{thJA}	-	62	°C/W
Case-to-Sink, Flat, Greased Surface	R_{thCS}	0.50	-	
Maximum Junction-to-Case (Drain)	R_{thJC}	-	1.7	

SPECIFICATIONS ($T_J = 25\text{ }^\circ\text{C}$, unless otherwise noted)						
PARAMETER	SYMBOL	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
Static						
Drain-Source Breakdown Voltage	V_{DS}	$V_{GS} = 0\text{ V}, I_D = 250\text{ }\mu\text{A}$	100	-	-	V
V_{DS} Temperature Coefficient	$\Delta V_{DS}/T_J$	Reference to $25\text{ }^\circ\text{C}$, $I_D = 1\text{ mA}$	-	0.12	-	V/°C
Gate-Source Threshold Voltage	$V_{GS(th)}$	$V_{DS} = V_{GS}, I_D = 250\text{ }\mu\text{A}$	2.0	-	4.0	V
Gate-Source Leakage	I_{GSS}	$V_{GS} = \pm 20\text{ V}$	-	-	± 100	nA
Zero Gate Voltage Drain Current	I_{DSS}	$V_{DS} = 100\text{ V}, V_{GS} = 0\text{ V}$	-	-	25	μA
		$V_{DS} = 80\text{ V}, V_{GS} = 0\text{ V}, T_J = 150\text{ }^\circ\text{C}$	-	-	250	
Drain-Source On-State Resistance	$R_{DS(on)}$	$V_{GS} = 10\text{ V}, I_D = 8.4\text{ A}^b$	-	-	0.16	Ω
Forward Transconductance	g_{fs}	$V_{DS} = 50\text{ V}, I_D = 8.4\text{ A}^b$	5.1	-	-	S
Dynamic						
Input Capacitance	C_{iss}	$V_{GS} = 0\text{ V},$ $V_{DS} = 25\text{ V},$ $f = 1.0\text{ MHz},$ see fig. 5	-	670	-	pF
Output Capacitance	C_{oss}		-	250	-	
Reverse Transfer Capacitance	C_{rss}		-	60	-	
Total Gate Charge	Q_g	$V_{GS} = 10\text{ V}, I_D = 14\text{ A}, V_{DS} = 80\text{ V},$ see fig. 6 and 13 ^b	-	-	26	nC
Gate-Source Charge	Q_{gs}		-	-	5.5	
Gate-Drain Charge	Q_{gd}		-	-	11	
Turn-On Delay Time	$t_{d(on)}$	$V_{DD} = 50\text{ V}, I_D = 14\text{ A}$ $R_g = 12\text{ }\Omega, R_D = 3.6\text{ }\Omega,$ see fig. 10 ^b	-	10	-	ns
Rise Time	t_r		-	34	-	
Turn-Off Delay Time	$t_{d(off)}$		-	23	-	
Fall Time	t_f		-	24	-	
Internal Drain Inductance	L_D	Between lead, 6 mm (0.25") from package and center of die contact 	-	4.5	-	nH
Internal Source Inductance	L_S		-	7.5	-	
Drain-Source Body Diode Characteristics						
Continuous Source-Drain Diode Current	I_S	MOSFET symbol showing the integral reverse p - n junction diode 	-	-	14	A
Pulsed Diode Forward Current ^a	I_{SM}		-	-	56	
Body Diode Voltage	V_{SD}	$T_J = 25\text{ }^\circ\text{C}, I_S = 14\text{ A}, V_{GS} = 0\text{ V}^b$	-	-	2.5	V
Body Diode Reverse Recovery Time	t_{rr}	$T_J = 25\text{ }^\circ\text{C}, I_F = 14\text{ A}, dI/dt = 100\text{ A}/\mu\text{s}^b$	-	150	280	ns
Body Diode Reverse Recovery Charge	Q_{rr}		-	0.85	1.7	μC
Forward Turn-On Time	t_{on}	Intrinsic turn-on time is negligible (turn-on is dominated by L_S and L_D)				

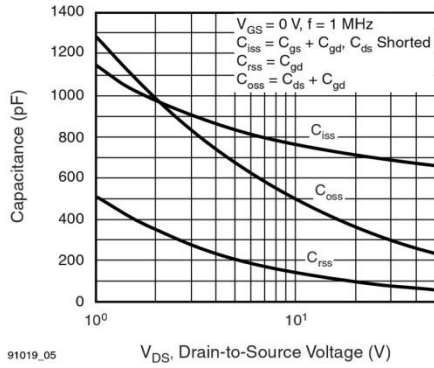
Notes

- a. Repetitive rating; pulse width limited by maximum junction temperature (see fig. 11).
- b. Pulse width $\leq 300\text{ }\mu\text{s}$; duty cycle $\leq 2\%$.

TYPICAL CHARACTERISTICS (25 °C, unless otherwise noted)

Fig. 1 - Typical Output Characteristics, $T_C = 25\text{ °C}$

Fig. 3 - Typical Transfer Characteristics

Fig. 2 - Typical Output Characteristics, $T_C = 175\text{ °C}$

Fig. 4 - Normalized On-Resistance vs. Temperature

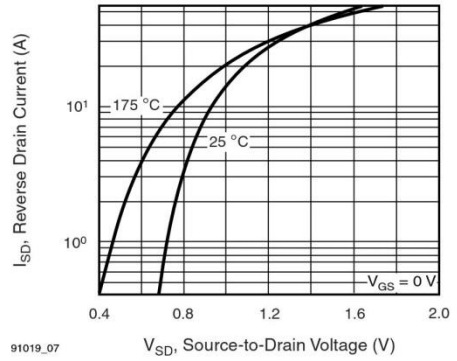
IRF530, SiHF530

Vishay Siliconix



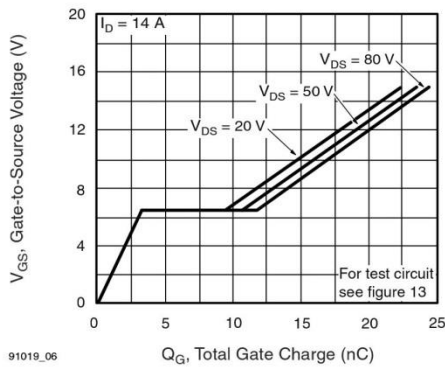
91019_05

Fig. 5 - Typical Capacitance vs. Drain-to-Source Voltage



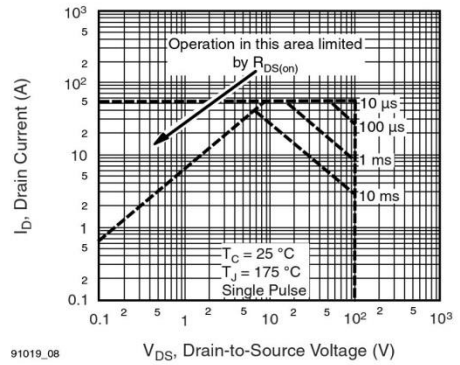
91019_07

Fig. 7 - Typical Source-Drain Diode Forward Voltage



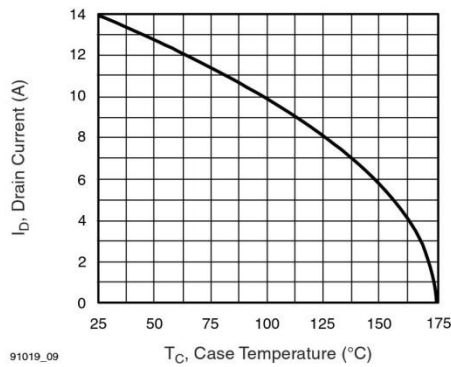
91019_06

Fig. 6 - Typical Gate Charge vs. Gate-to-Source Voltage

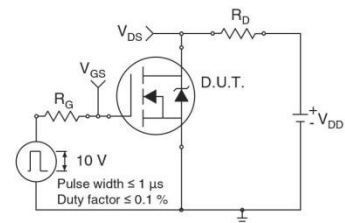
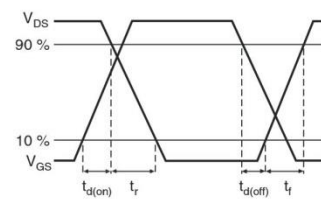
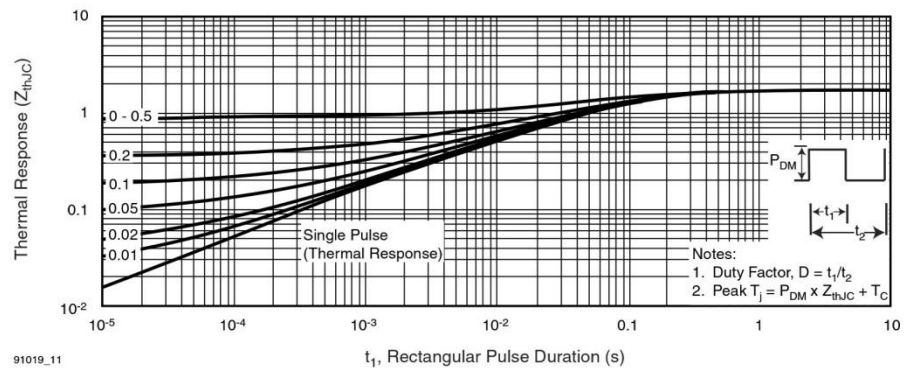


91019_08

Fig. 8 - Maximum Safe Operating Area



91019_09

Fig. 9 - Maximum Drain Current vs. Case Temperature

Fig. 10a - Switching Time Test Circuit

Fig. 10b - Switching Time Waveforms


91019_11

Fig. 11 - Maximum Effective Transient Thermal Impedance, Junction-to-Case

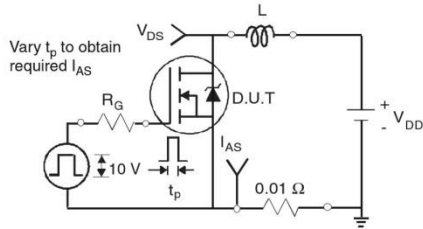


Fig. 12a - Unclamped Inductive Test Circuit

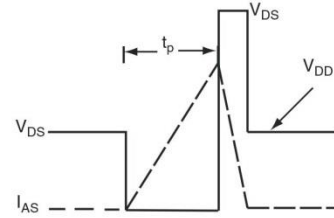


Fig. 12b - Unclamped Inductive Waveforms

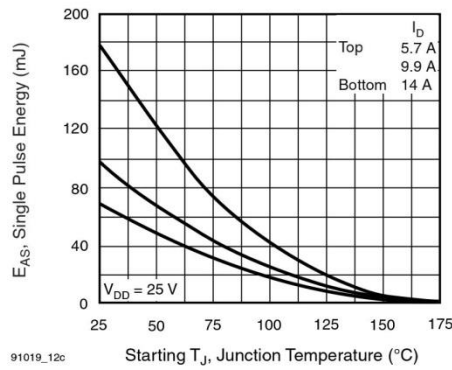


Fig. 12c - Maximum Avalanche Energy vs. Drain Current

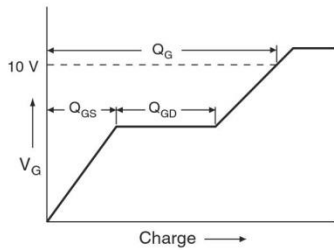


Fig. 13a - Basic Gate Charge Waveform

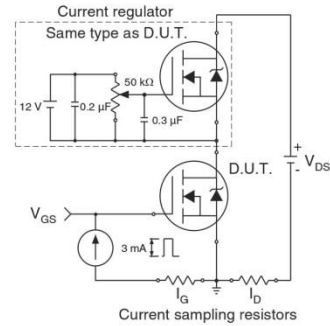
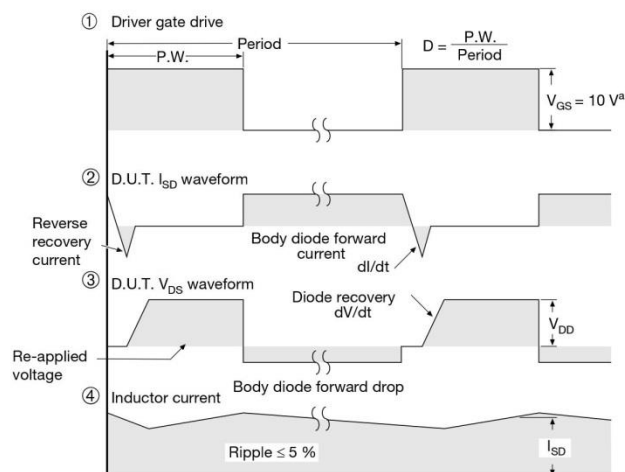
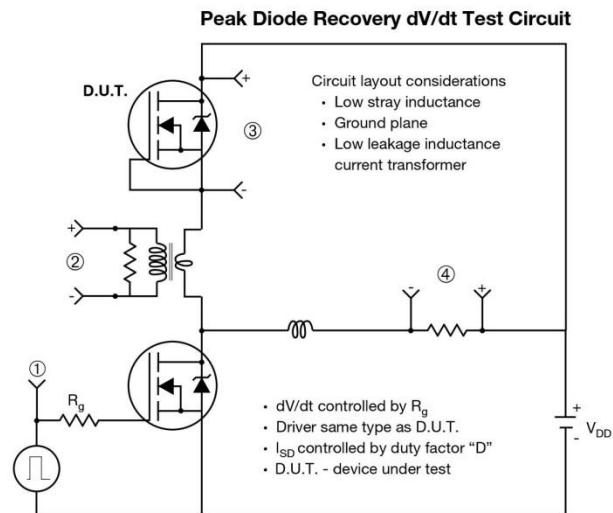


Fig. 13b - Gate Charge Test Circuit



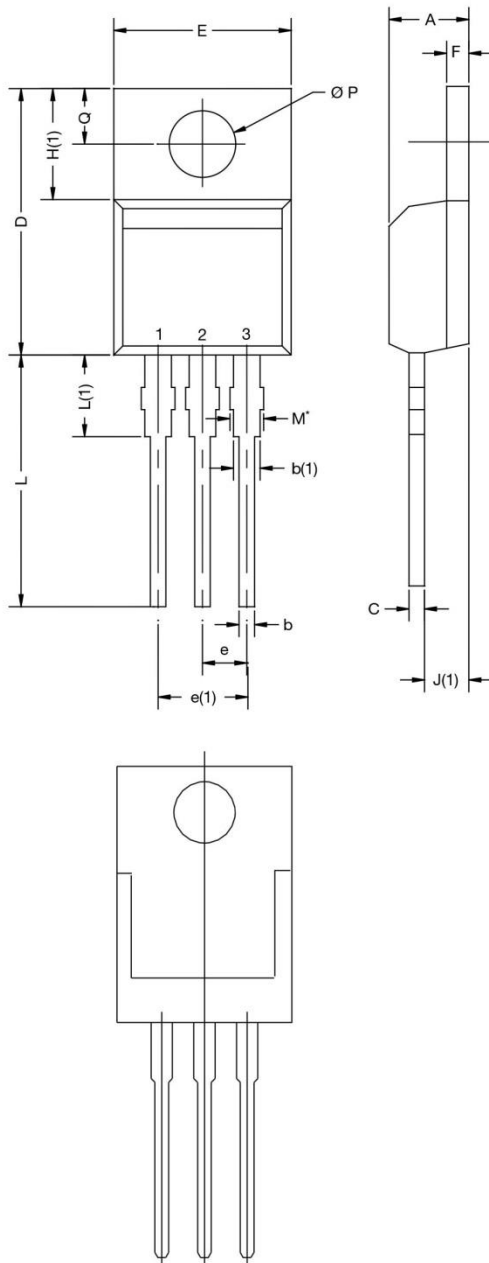
Note
a. $V_{GS} = 5 V$ for logic level devices

Fig. 14 - For N-Channel

Vishay Siliconix maintains worldwide manufacturing capability. Products may be manufactured at one of several qualified locations. Reliability data for Silicon Technology and Package Reliability represent a composite of all qualified locations. For related documents such as package/tape drawings, part marking, and reliability data, see www.vishay.com/ppg?91019.



TO-220-1



DIM.	MILLIMETERS		INCHES	
	MIN.	MAX.	MIN.	MAX.
A	4.14	4.70	0.163	0.185
b	0.69	1.02	0.027	0.040
b(1)	1.14	1.73	0.045	0.068
c	0.36	0.61	0.014	0.024
D	14.33	15.85	0.564	0.624
E	9.96	10.52	0.392	0.414
e	2.41	2.67	0.095	0.105
e(1)	4.88	5.28	0.192	0.208
F	0.43	1.40	0.017	0.055
H(1)	6.10	6.48	0.240	0.255
J(1)	2.41	2.92	0.095	0.115
L	13.36	14.40	0.526	0.567
L(1)	3.33	4.04	0.131	0.159
Ø P	3.53	3.94	0.139	0.155
Q	2.59	3.00	0.102	0.118

ECN: X15-0003-Rev. A, 19-Jan-15
DWG: 6031

- Notes**
- M* = 0.052 inches to 0.064 inches (dimension including protrusion), heatsink hole for HVM
 - Outline conforms to JEDEC® outline TO-220AB with exception of dimension F



Disclaimer

ALL PRODUCT, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, "Vishay"), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Vishay makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Vishay disclaims (i) any and all liability arising out of the application or use of any product, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Vishay's knowledge of typical requirements that are often placed on Vishay products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer's responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application. Parameters provided in datasheets and/or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer's technical experts. Product specifications do not expand or otherwise modify Vishay's terms and conditions of purchase, including but not limited to the warranty expressed therein.

Except as expressly indicated in writing, Vishay products are not designed for use in medical, life-saving, or life-sustaining applications or for any other application in which the failure of the Vishay product could result in personal injury or death. Customers using or selling Vishay products not expressly indicated for use in such applications do so at their own risk. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay. Product names and markings noted herein may be trademarks of their respective owners.

Material Category Policy

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as RoHS-Compliant fulfill the definitions and restrictions defined under Directive 2011/65/EU of The European Parliament and of the Council of June 8, 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (EEE) - recast, unless otherwise specified as non-compliant.

Please note that some Vishay documentation may still make reference to RoHS Directive 2002/95/EC. We confirm that all the products identified as being compliant to Directive 2002/95/EC conform to Directive 2011/65/EU.

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as Halogen-Free follow Halogen-Free requirements as per JEDEC JS709A standards. Please note that some Vishay documentation may still make reference to the IEC 61249-2-21 definition. We confirm that all the products identified as being compliant to IEC 61249-2-21 conform to JEDEC JS709A standards.

B. COTIZACIÓN

Una de las principales motivaciones del presente proyecto de tesis, fue hacer asequible para pequeñas empresas, instituciones e incluso hogares el corte de precisión láser. Por ello, ahora se realiza una cotización de forma aproximada del costo de los componentes que integran la cortadora láser y su fabricación. El precio bruto consiste solamente en el costo de componentes y fabricación, luego se halla el precio que incluye costos de importación (agente aduanero, IGV, derecho de flete, etc) que se aproxima como un incremento del 80% del precio bruto, luego se obtiene el precio comercial que viene a ser el precio al cual se vendería la máquina, el cual considera un margen de ganancia y costos de ingeniería. Todos los precios se encuentran en dólares americanos (USD \$), en el caso de la compra local incluyen IGV y en el caso de importación incluyen costos de envío.

Tabla 4.1 Cotización de Cortadora Láser CNC para acrílicos y madera

Cotización				
Cortadora Láser CNC para acrílicos y madera				
Artículo	Cantidad	Precio Unitario	Precio Total	Origen/Vendedor
Óptica				
Láser Synrad 48-5(S)W	1	\$700	\$700	EEUU/Synrad
Espejos PLBBA0041	1	\$29	\$29	EEUU/Synrad
Enfriador CW3000AG	1	\$198	\$198	EEUU/AliExpress(Internet)
Mecánica				
Barra de Bronce 2m	2	\$15	\$30	Perú/Broncería Espinal
Barra SAE 304 2m	2	\$28	\$56	Perú/Sider Perú
Tornillo de precisión SKF serie SDS/BDS/SHS	2	\$89	\$178	Lulea/SKF del Perú
Otros (Tornillos, Marcos, Corredoras)	X	\$30	\$30	Perú
Electrónica				
Microswitch "heat resistant" SS-01-T	1	\$23	\$23	EEUU/OMRON
Microswitch "standard" SS-01GL	2	\$8	\$16	EEUU/OMRON
Motor de Paso NEMA 23 780079-01	2	\$79	\$158	EEUU/National Instruments
Tarjeta Arduino UNO	1	\$35	\$35	EEUU/Amazon
Fabricación				
Doblado de plancha (Carro eje X e Y)	2	\$20	\$40	Perú/Edipesa Metalmecánica
Fresado (Carro eje X e Y)	2	\$30	\$60	Perú/Edipesa Metalmecánica
Tarjeta Impresa (Inc. Componentes)	1	\$50	\$50	Perú/Bazán
Impresión 3D Cabina PMMA	1	\$240	\$240	Perú/D3SARROLLO 3D
Ensamblaje Global	1	\$50	\$50	Perú/Edipesa Metalmecánica
Programación	1	\$200	\$200	-
Precio Bruto			\$2,093	
Precio con costos de importación (+80%)			\$3,767	
Precio Comercial (+30% Precio Bruto)			\$4,897.62	