

```
1  //
   -----
2  // Copyright (c) 2005 by Terasic Technologies Inc.
3  //
   -----
4  //
5  // Permission:
6  //
7  //   Terasic grants permission to use and modify this code for use
8  //   in synthesis for all Terasic Development Boards and Altera
   Development
9  //   Kits made by Terasic.  Other use of this code, including the
   selling
10 //   ,duplication, or modification of any portion is strictly
   prohibited.
11 //
12 // Disclaimer:
13 //
14 //   This VHDL/Verilog or C/C++ source code is intended as a
   design reference
15 //   which illustrates how these types of functions can be
   implemented.
16 //   It is the user's responsibility to verify their design for
17 //   consistency and functionality through the use of formal
18 //   verification methods.  Terasic provides no warranty
   regarding the use
19 //   or functionality of this code.
20 //
21 //
   -----
22 //
23 //
24 //   Terasic Technologies Inc
25 //   356 Fu-Shin E. Rd Sec. 1. JhuBei City,
26 //   HsinChu County, Taiwan
27 //   302
28 //
29 //   web: http://www.terasic.com/
30 //   email: support@terasic.com
31 //
   -----
32 //
33 // Major Functions: DE2 Default Bitstream
34 //
35 //
   -----
36 //
37 // Revision History :
38 //
   -----
39 //   Ver   :| Author           :| Mod. Date :| Changes Made:
40 //   V1.0  :| Johnny Chen      :| 05/08/19 :| Initial
   Revision
```

```

41 // V1.1 :| Sean Peng           :| 05/09/30  :|      Changed
    CLOCK, SW, LEDG/R
42 //                                     according to
    Zvonko's requests.
43 // V1.2 :| Johnny Chen         :| 05/11/16  :|      Add to FLASH
    Address FL_ADDR[21:20]
44 // V1.3 :| Johnny Chen         :| 05/12/12  :|      Fixed
    VGA_Audio_PLL Initial Sequence.
45 //
-----
-
46 //
-----
-
47 //                               Modulador Digital FM
48 //
-----
-
49 // Título: FM digital modulator SDR
50 //
-----
-
51 // Autor: Jorge Tonfat
52 //
-----
-
53 // Descripción: Integración de todos los modulos diseñados
54 //
-----
-
55 // Nombre: DE2_Default
56 //
-----
-
57 // Fecha de creación: 20/11/2007
58 // Última modificación: 03/12/2007
59 //
-----
-
60
61
62 module DE2_Default
63 (
64     ///////////////////////////////////      Clock Input
    ///////////////////////////////////
65     CLOCK_27,                               // 27 MHz
66     CLOCK_50,                               // 50 MHz
67     EXT_CLOCK,                             // External Clock
68     ///////////////////////////////////      Push Button
    ///////////////////////////////////
69     KEY,                                     // Pushbutton[3:0]
70     ///////////////////////////////////      DPDT Switch
    ///////////////////////////////////
71     SW,                                     // Toggle Switch[17:0]
72     ///////////////////////////////////      7-SEG Dispaly
    ///////////////////////////////////
73     HEX0,                                   // Seven Segment Digit 0
74     HEX1,                                   // Seven Segment Digit 1
75     HEX2,                                   // Seven Segment Digit 2

```

```

76      HEX3,          // Seven Segment Digit 3
77      HEX4,          // Seven Segment Digit 4
78      HEX5,          // Seven Segment Digit 5
79      HEX6,          // Seven Segment Digit 6
80      HEX7,          // Seven Segment Digit 7
81      //////////////// LED
      ////////////////
82      LEDG,          // LED Green[8:0]
83      LEDR,          // LED Red[17:0]
84      //////////////// UART
      ////////////////
85      UART_TXD,       // UART Transmitter
86      UART_RXD,       // UART Receiver
87      //////////////// IRDA
      ////////////////
88      IRDA_TXD,       // IRDA Transmitter
89      IRDA_RXD,       // IRDA Receiver
90      //////////////// SDRAM Interface
      ////////////////
91      DRAM_DQ,        // SDRAM Data bus 16 Bits
92      DRAM_ADDR,      // SDRAM Address bus 12
      Bits
93      DRAM_LDQM,      // SDRAM Low-byte Data
      Mask
94      DRAM_UDQM,      // SDRAM High-byte Data
      Mask
95      DRAM_WE_N,      // SDRAM Write Enable
96      DRAM_CAS_N,     // SDRAM Column Address
      Strobe
97      DRAM_RAS_N,     // SDRAM Row Address
      Strobe
98      DRAM_CS_N,      // SDRAM Chip Select
99      DRAM_BA_0,      // SDRAM Bank Address 0
100     DRAM_BA_1,      // SDRAM Bank Address 0
101     DRAM_CLK,        // SDRAM Clock
102     DRAM_CKE,        // SDRAM Clock Enable
103     //////////////// Flash Interface
      ////////////////
104     FL_DQ,           // FLASH Data bus 8 Bits
105     FL_ADDR,         // FLASH Address bus 22
      Bits
106     FL_WE_N,         // FLASH Write Enable
107     FL_RST_N,        // FLASH Reset
108     FL_OE_N,         // FLASH Output Enable
109     FL_CE_N,         // FLASH Chip Enable
110     //////////////// SRAM Interface
      ////////////////
111     SRAM_DQ,         // SRAM Data bus 16 Bits
112     SRAM_ADDR,       // SRAM Address bus 18
      Bits
113     SRAM_UB_N,       // SRAM High-byte Data
      Mask
114     SRAM_LB_N,       // SRAM Low-byte Data
      Mask
115     SRAM_WE_N,       // SRAM Write Enable
116     SRAM_CE_N,       // SRAM Chip Enable
117     SRAM_OE_N,       // SRAM Output Enable
118     //////////////// ISP1362 Interface
      ////////////////

```

```

119      OTG_DATA,          // ISP1362 Data bus 16
      Bits
120      OTG_ADDR,          // ISP1362 Address 2 Bits
121      OTG_CS_N,          // ISP1362 Chip Select
122      OTG_RD_N,          // ISP1362 Write
123      OTG_WR_N,          // ISP1362 Read
124      OTG_RST_N,         // ISP1362 Reset
125      OTG_FSPPEED,       // USB Full Speed, 0 =
      Enable, Z = Disable
126      OTG_LSPPEED,       // USB Low Speed, 0 =
      Enable, Z = Disable
127      OTG_INT0,          // ISP1362 Interrupt 0
128      OTG_INT1,          // ISP1362 Interrupt 1
129      OTG_DREQ0,          // ISP1362 DMA Request 0
130      OTG_DREQ1,          // ISP1362 DMA Request 1
131      OTG_DACK0_N,        // ISP1362 DMA
      Acknowledge 0
132      OTG_DACK1_N,        // ISP1362 DMA
      Acknowledge 1
133      //////////////// LCD Module 16X2
      ////////////////
134      LCD_ON,             // LCD Power ON/OFF
135      LCD_BLON,           // LCD Back Light ON/OFF
136      LCD_RW,             // LCD Read/Write
      Select, 0 = Write, 1 = Read
137      LCD_EN,             // LCD Enable
138      LCD_RS,             // LCD Command/Data
      Select, 0 = Command, 1 = Data
139      LCD_DATA,           // LCD Data bus 8 bits
140      //////////////// SD_Card Interface
      ////////////////
141      SD_DAT,             // SD Card Data
142      SD_DAT3,            // SD Card Data 3
143      SD_CMD,             // SD Card Command Signal
144      SD_CLK,             // SD Card Clock
145      //////////////// USB JTAG link
      ////////////////
146      TDI,               // CPLD -> FPGA (data in)
147      TCK,               // CPLD -> FPGA (clk)
148      TCS,               // CPLD -> FPGA (CS)
149      TDO,               // FPGA -> CPLD (data out)
150      //////////////// I2C
      ////////////////
151      I2C_SDAT,           // I2C Data
152      I2C_SCLK,           // I2C Clock
153      //////////////// PS2
      ////////////////
154      PS2_DAT,            // PS2 Data
155      PS2_CLK,            // PS2 Clock
156      //////////////// VGA
      ////////////////
157      VGA_CLK,            // VGA Clock
158      VGA_HS,             // VGA H_SYNC
159      VGA_VS,             // VGA V_SYNC
160      VGA_BLANK,          // VGA BLANK
161      VGA_SYNC,           // VGA SYNC
162      VGA_R,              // VGA Red[9:0]
163      VGA_G,              // VGA Green[9:0]
164      VGA_B,              // VGA Blue[9:0]

```

```

165          // Ethernet Interface
166          //////////////////////////////////
167          ENET_DATA, // DM9000A DATA bus 16Bit:
168          ENET_CMD, // DM9000A Command/Data
169          Select, 0 = Command, 1 = Data
170          ENET_CS_N, // DM9000A Chip Select
171          ENET_WR_N, // DM9000A Write
172          ENET_RD_N, // DM9000A Read
173          ENET_RST_N, // DM9000A Reset
174          ENET_INT, // DM9000A Interrupt
175          ENET_CLK, // DM9000A Clock 25 MHz
176          ////////////////////////////////// Audio CODEC
177          //////////////////////////////////
178          AUD_ADCLRCK, // Audio CODEC ADC LR
179          Clock
180          AUD_ADCDAT, // Audio CODEC ADC Data
181          AUD_DACLCK, // Audio CODEC DAC LR
182          Clock
183          AUD_DACDAT, // Audio CODEC DAC Data
184          AUD_BCLK, // Audio CODEC
185          Bit-Stream Clock
186          AUD_XCK, // Audio CODEC Chip Clock
187          ////////////////////////////////// TV Decoder
188          //////////////////////////////////
189          TD_DATA, // TV Decoder Data bus 8
190          bits
191          TD_HS, // TV Decoder H_SYNC
192          TD_VS, // TV Decoder V_SYNC
193          TD_RESET, // TV Decoder Reset
194          ////////////////////////////////// GPIO
195          //////////////////////////////////
196          GPIO_0, // GPIO Connection 0
197          GPIO_1, // GPIO Connection 1
198          );
199
200          ////////////////////////////////// Clock Input
201          //////////////////////////////////
202          input CLOCK_27; // 27 MHz
203          input CLOCK_50; // 50 MHz
204          input EXT_CLOCK; // External Clock
205          ////////////////////////////////// Push Button
206          //////////////////////////////////
207          input [3:0] KEY; // Pushbutton[3:0]
208          ////////////////////////////////// DPDT Switch
209          //////////////////////////////////
210          input [17:0] SW; // Toggle Switch[17:0]
211          ////////////////////////////////// 7-SEG Dispaly
212          //////////////////////////////////
213          output [6:0] HEX0; // Seven Segment Digit 0
214          output [6:0] HEX1; // Seven Segment Digit 1
215          output [6:0] HEX2; // Seven Segment Digit 2
216          output [6:0] HEX3; // Seven Segment Digit 3
217          output [6:0] HEX4; // Seven Segment Digit 4
218          output [6:0] HEX5; // Seven Segment Digit 5
219          output [6:0] HEX6; // Seven Segment Digit 6
220          output [6:0] HEX7; // Seven Segment Digit 7
221          ////////////////////////////////// LED
222          //////////////////////////////////
223          output [8:0] LEDG; // LED Green[8:0]

```

```

210    output    [17:0] LEDR;                // LED Red[17:0]
211    ///////////////////////////////////////////////////    UART
212    ///////////////////////////////////////////////////
213    output    UART_TXD;                    // UART Transmitter
214    input     UART_RXD;                    // UART Receiver
215    ///////////////////////////////////////////////////    IRDA
216    ///////////////////////////////////////////////////
217    output    IRDA_TXD;                    // IRDA Transmitter
218    input     IRDA_RXD;                    // IRDA Receiver
219    ///////////////////////////////////////////////////    SDRAM Interface
220    ///////////////////////////////////////////////////
221    inout     [15:0] DRAM_DQ;               // SDRAM Data bus 16 Bits
222    output    [11:0] DRAM_ADDR;            // SDRAM Address bus 12
223    Bits
224    output    DRAM_LDQM;                   // SDRAM Low-byte Data
225    Mask
226    output    DRAM_UDQM;                   // SDRAM High-byte Data
227    Mask
228    output    DRAM_WE_N;                   // SDRAM Write Enable
229    output    DRAM_CAS_N;                  // SDRAM Column Address
230    Strobe
231    output    DRAM_RAS_N;                  // SDRAM Row Address
232    Strobe
233    output    DRAM_CS_N;                   // SDRAM Chip Select
234    output    DRAM_BA_0;                   // SDRAM Bank Address 0
235    output    DRAM_BA_1;                   // SDRAM Bank Address 0
236    output    DRAM_CLK;                    // SDRAM Clock
237    output    DRAM_CKE;                    // SDRAM Clock Enable
238    ///////////////////////////////////////////////////    Flash Interface
239    ///////////////////////////////////////////////////
240    inout     [7:0] FL_DQ;                  // FLASH Data bus 8 Bits
241    output    [21:0] FL_ADDR;               // FLASH Address bus 22
242    Bits
243    output    FL_WE_N;                     // FLASH Write Enable
244    output    FL_RST_N;                     // FLASH Reset
245    output    FL_OE_N;                      // FLASH Output Enable
246    output    FL_CE_N;                      // FLASH Chip Enable
247    ///////////////////////////////////////////////////    SRAM Interface
248    ///////////////////////////////////////////////////
249    inout     [15:0] SRAM_DQ;               // SRAM Data bus 16 Bits
250    output    [17:0] SRAM_ADDR;            // SRAM Address bus 18
251    Bits
252    output    SRAM_UB_N;                   // SRAM High-byte Data
253    Mask
254    output    SRAM_LB_N;                   // SRAM Low-byte Data
255    Mask
256    output    SRAM_WE_N;                   // SRAM Write Enable
257    output    SRAM_CE_N;                   // SRAM Chip Enable
258    output    SRAM_OE_N;                   // SRAM Output Enable
259    ///////////////////////////////////////////////////    ISP1362 Interface
260    ///////////////////////////////////////////////////
261    inout     [15:0] OTG_DATA;              // ISP1362 Data bus 16
262    Bits
263    output    [1:0] OTG_ADDR;               // ISP1362 Address 2 Bits
264    output    OTG_CS_N;                     // ISP1362 Chip Select
265    output    OTG_RD_N;                     // ISP1362 Write
266    output    OTG_WR_N;                     // ISP1362 Read
267    output    OTG_RST_N;                    // ISP1362 Reset
268    output    OTG_FSPEED;                   // USB Full Speed, 0 =

```

```

Enable, Z = Disable
253 output      OTG_LSPEED;           // USB Low Speed, 0 =
Enable, Z = Disable
254 input      OTG_INT0;              // ISP1362 Interrupt 0
255 input      OTG_INT1;              // ISP1362 Interrupt 1
256 input      OTG_DREQ0;             // ISP1362 DMA Request 0
257 input      OTG_DREQ1;             // ISP1362 DMA Request 1
258 output      OTG_DACK0_N;          // ISP1362 DMA
Acknowledge 0
259 output      OTG_DACK1_N;          // ISP1362 DMA
Acknowledge 1
260 /////////////////////////////////////////////////// LCD Module 16X2
//////////////////////////////////////
261 inout [7:0] LCD_DATA;              // LCD Data bus 8 bits
262 output      LCD_ON;                // LCD Power ON/OFF
263 output      LCD_BLON;              // LCD Back Light ON/OFF
264 output      LCD_RW;                // LCD Read/Write
Select, 0 = Write, 1 = Read
265 output      LCD_EN;                // LCD Enable
266 output      LCD_RS;                // LCD Command/Data
Select, 0 = Command, 1 = Data
267 /////////////////////////////////////////////////// SD Card Interface
//////////////////////////////////////
268 inout      SD_DAT;                 // SD Card Data
269 inout      SD_DAT3;                // SD Card Data 3
270 inout      SD_CMD;                 // SD Card Command Signal
271 output      SD_CLK;                // SD Card Clock
272 /////////////////////////////////////////////////// I2C
//////////////////////////////////////
273 inout      I2C_SDAT;               // I2C Data
274 output      I2C_SCLK;               // I2C Clock
275 /////////////////////////////////////////////////// PS2
//////////////////////////////////////
276 input      PS2_DAT;                // PS2 Data
277 input      PS2_CLK;                // PS2 Clock
278 /////////////////////////////////////////////////// USB JTAG link
//////////////////////////////////////
279 input      TDI;                    // CPLD -> FPGA (data in)
280 input      TCK;                    // CPLD -> FPGA (clk)
281 input      TCS;                    // CPLD -> FPGA (CS)
282 output      TDO;                   // FPGA -> CPLD (data out)
283 /////////////////////////////////////////////////// VGA
//////////////////////////////////////
284 output      VGA_CLK;                // VGA Clock
285 output      VGA_HS;                 // VGA H_SYNC
286 output      VGA_VS;                 // VGA V_SYNC
287 output      VGA_BLANK;              // VGA BLANK
288 output      VGA_SYNC;               // VGA SYNC
289 output [9:0] VGA_R;                 // VGA Red[9:0]
290 output [9:0] VGA_G;                 // VGA Green[9:0]
291 output [9:0] VGA_B;                 // VGA Blue[9:0]
292 /////////////////////////////////////////////////// Ethernet Interface
//////////////////////////////////////
293 inout [15:0] ENET_DATA;             // DM9000A DATA bus 16Bit:
294 output      ENET_CMD;               // DM9000A Command/Data
Select, 0 = Command, 1 = Data
295 output      ENET_CS_N;              // DM9000A Chip Select
296 output      ENET_WR_N;              // DM9000A Write
297 output      ENET_RD_N;              // DM9000A Read

```

```

298 output      ENET_RST_N;           // DM9000A Reset
299 input       ENET_INT;             // DM9000A Interrupt
300 output      ENET_CLK;             // DM9000A Clock 25 MHz
301 ////////////////////////////////////////////////// Audio CODEC
302 //////////////////////////////////////////////////
302 output      AUD_ADCLRCK;           // Audio CODEC ADC LR
303 Clock
303 input       AUD_ADCDAT;           // Audio CODEC ADC Data
304 output      AUD_DACLCK;           // Audio CODEC DAC LR
305 Clock
305 output      AUD_DACDAT;           // Audio CODEC DAC Data
306 output      AUD_BCLK;             // Audio CODEC
307 Bit-Stream Clock
307 output      AUD_XCK;              // Audio CODEC Chip Clock
308 ////////////////////////////////////////////////// TV Devoder
309 //////////////////////////////////////////////////
309 input [7:0]  TD_DATA;             // TV Decoder Data bus 8
310 bits
310 input       TD_HS;               // TV Decoder H_SYNC
311 input       TD_VS;               // TV Decoder V_SYNC
312 output      TD_RESET;            // TV Decoder Reset
313 ////////////////////////////////////////////////// GPIO
314 //////////////////////////////////////////////////
314 inout [35:0] GPIO_0;             // GPIO Connection 0
315 inout [35:0] GPIO_1;             // GPIO Connection 1
316
317 // LCD ON
318 assign LCD_ON      = 1'b0;
319 assign LCD_BLON    = 1'b0;
320
321 // All inout port turn to tri-state
322 assign DRAM_DQ      = 16'hzzzz;
323 assign FL_DQ        = 8'hzz;
324 assign SRAM_DQ      = 16'hzzzz;
325 assign OTG_DATA     = 16'hzzzz;
326 assign SD_DAT       = 1'bz;
327 assign ENET_DATA    = 16'hzzzz;
328 assign GPIO_0       = 36'hzzzzzzzzzz;
329 assign GPIO_1       = 36'hzzzzzzzzzz;
330
331 wire VGA_CTRL_CLK;
332 wire AUD_CTRL_CLK;
333 wire DLY_RST;
334
335 assign TD_RESET     = 1'b1; // Allow 27 MHz
336 assign AUD_ADCLCK   = AUD_DACLCK;
337 assign AUD_XCK      = AUD_CTRL_CLK;
338
339 //DDS hardware
340 reg [31:0] DDS_accum; // registro del acumulador de fase
341 wire [9:0] sq_out;
342 reg [31:0] DDS_accum_mod;
343 wire [9:0] mod_out;
344
345 wire signed [15:0] modulation;
346 wire signed [31:0] post_modulation;
347 wire [19:0] scaled_mod;
348 //bogus clock
349 wire VGA_CLK_c2;

```



```

350
351   Reset_Delay          r0  (  .iCLK(CLOCK_50),
352                               .oRESET(DLY_RST)    );
353
354   VGA_Audio_PLL        p1  (  .areset(~DLY_RST),
355                               .inclk0(CLOCK_27),
356                               .c0(VGA_CTRL_CLK),
357                               .c1(AUD_CTRL_CLK),
358                               .c2(VGA_CLK_c2) );
359
360
361   I2C_AV_Config         u3  (  // Host Side
362                               .iCLK(CLOCK_50),
363                               .IRST_N(KEY[0]),
364                               // I2C Side
365                               .I2C_SCLK(I2C_SCLK),
366                               .I2C_SDAT(I2C_SDAT) );
367
368   //set up DDS frequency
369   //Use switches to set freq
370
371   AUDIO_DAC_ADC         u4  (  // Audio Side
372                               .oAUD_BCK(AUD_BCLK),
373                               .oAUD_DATA(AUD_DACDAT),
374                               .oAUD_LRCK(AUD_DACLCK),
375                               .iAUD_ADCDAT(AUD_ADCDAT),
376                               // Control Signals
377                               .iSrc_Select(SW[17]),
378                               .iCLK_18_4(AUD_CTRL_CLK),
379                               .IRST_N(DLY_RST),
380                               .modulation(modulation)
381                               );
382
383   audio_system          u5  (
384                               // global signals
385                               .clk(CLOCK_50),
386                               .reset_n(KEY[0]),
387
388                               // the_pio
389                               .in_port_to_the_pio(modulation),
390
391                               // the_pio_audio
392                               .out_port_from_the_pio_audio(
393                               post_modulation)
394                               );
395
396   always @ (posedge CLOCK_50) begin
397       //generacion de una se;al FM con una portadora de 4.5Mhz
398       DDS_accum <= DDS_accum + 32'h16FC7BBC + (post_modulation[
399       31:8]);
400
401   end
402
403   //hook up the ROM table for carrier
404   sqwave sqTable(CLOCK_50, DDS_accum[31:24], sq_out);
405
406   //asignacion del DAC de video como puerto de salida

```

```
407     assign VGA_R = sq_out;
408     assign VGA_SYNC = 1 ;
409     assign VGA_BLANK = 1 ;
410     assign VGA_CLK = CLOCK_50 ;
411     // assign LEDR[15:0] = ~modulation;
412 endmodule
413
414 //////////////////////////////////////
415 // DDS generador de onda cuadrada
416 //////////////////////////////////////
417 module sqwave (clock, address, sq);
418     input clock;
419     input [7:0] address;
420     output [9:0] sq;
421     reg [9:0] sq;
422     always @ (posedge clock)
423     begin
424         sq <= (address<128)? 10'h1fff : 10'h001 ;
425     end
426 endmodule
427
```