PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DISEÑO DE UN MODULADOR FM BASADO EN LA TECNOLOGÍA SOFTWARE-DEFINED RADIO EN FPGA

Tesis para optar el Título de Ingeniero Electrónico

Presentado por:

JORGE LUCIO TONFAT SECLEN

Lima – PERÚ

2008

RESUMEN

La aparición de una gran cantidad de estándares para comunicaciones inalámbricas como WLAN IEEE 802.11, WIMAX, GPRS, Bluetooth, etc. ha aumentado el problema que enfrentan los diseñadores de equipos de telecomunicaciones que requieren cada vez más espacio en sus equipos para la adición de nuevos circuitos que soporten los estándares emergentes. La tecnología *Software-defined radio* (SDR) ha generado la atención de las telecomunicaciones debido a que ofrece una solución al problema actual. Se basa en la idea de llevar el software lo más cerca que se pueda a la antena. Pretende reemplazar a todos los circuitos que realizan la modulación y demodulación por un algoritmo que se ejecute en un procesador de propósito general. Esta característica le da una gran flexibilidad y adaptabilidad ante la aparición de nuevos estándares. Estas dos propiedades son las que se quieren aprovechar para plantear una solución al problema que existe actualmente en las comunicaciones de emergencia en nuestro país. El problema reside en la incompatibilidad de algunos equipos para poder comunicarse debido a diferencias en las bandas de operación y en algunos casos al tipo de modulación empleado. El presente trabajo pretende mostrar una alternativa tecnológica al problema mencionado utilizando la tecnología SDR. La propuesta consiste en realizar un diseño digital basado en FPGA (arreglo de compuertas programables en campo) que sea capaz de realizar la etapa de la modulación y selección de la frecuencia utilizando un código en lenguaje C. Se utilizará el CODEC WM8731 como dispositivo para la adquisición de la señal de audio que será procesada en el FPGA, para ello se utilizará la tarjeta de desarrollo Altera DE2 Development kit como hardware para realizar las pruebas respectivas. Todo el tratamiento de la señal se realizará en banda base para luego ser modulada a la frecuencia respectiva utilizando un sintetizador digital directo. La señal modulada se obtendrá utilizando el conversor digital/análogo ADV7123. Los resultados muestran que es posible realizar un modulador de señales FM dentro de la banda de frecuencias de FM comercial sin utilizar dispositivos dedicados a las telecomunicaciones. El sistema diseñado es capaz de realizar también modulaciones AM con tan sólo una variación del código del procesador.

A Dios, por darme la fuerza y tranquilidad para salir adelante.
A mis padres, Jorge y Mirtha, por todo su amor y apoyo, sin ellos no hubiera podido llegar a ser quien soy hoy.
Al Dr. Carlos Silva, por su apoyo y confianza, por sus consejos y dedicación en el desarrollo de esta tesis y sobre todo por su amistad.
Al Ing. Mario Raffo, por su apoyo y confianza, pues con su ejemplo de esfuerzo y dedicación me enseñó a seguir intentando.
Al Grupo de Microelectrónica de la PUCP: Jorge Benavides, Joel Muñoz, Heiner Alarcón, Héctor Villacorta, César Saldaña, Erick Raygada, Manuel Monge, José Quenta, Jorge Lagos, Walter Calienes, Eva Benites y a todos los demás integrantes actuales, pues fue ahí donde encontré una segunda familia y el lugar donde descubrí el arte de investigación.

A todos, muchas gracias.

ÍNDICE

INTRODUCCIÓN 1
CAPÍTULO 1: LAS COMUNICACIONES INALÁMBRICAS DE EMERGENCIAS EN NUESTRO PAÍS 2
1.1 Variables que limitan el desarrollo de estos equipos 2
1.2 Proceso de comunicación durante una emergencia 3
1.3 Declaración del marco problemático 4
CAPÍTULO 2: TECNOLOGÍA SDR: EQUIPOS DEFINIDOS POR SOFTWARE 5
2.1 Introducción 5
2.2 El concepto de software radio 5
2.3 Problemas de diseño en un equipo SDR 6
2.4 Investigaciones académicas 8
2.5 Aplicaciones comerciales 9
2.6 Beneficios de la tecnología SDR 9
2.7 Síntesis sobre el asunto de estudio 10
2.8 Conceptos Generales acerca de un equipo definido por software 10
2.8.1 Equipos de Comunicaciones inalámbricas 10
2.8.1.1 Comunicaciones inalámbricas comerciales 10
2.8.1.2 Comunicaciones inalámbricas militares 11
2.8.2 Tecnología SDR 11
2.8.2.1 Definición 11
2.8.2.2 Arquitectura 12
2.8.3 FPGA 12
2.8.3.1 Definición 12
2.8.3.2 Arquitectura de un FPGA 12
2.8.3.3 Arquitectura del FPGA EP2C35 de Altera 14
2.9 Modelo Teórico 17
2.10 Definiciones operativas 18
2.10.1 Indicadores Cuantitativos 18
CAPÍTULO 3: DISEÑO DE LA ARQUITECTURA PROPUESTA PARA EL MODULADOR FM 19
3.1 Hipótesis de la investigación 19
3.1.1 Hipótesis principal 19
3.1.2 Hipótesis secundarias 19
3.2 Objetivos de la investigación 20
3.2.1 Objetivo general 20

3.2.2 Objetivos específicos 20
3.3 Desarrollo del hardware del sistema 21
3.3.1 Sintetizador Digital Directo (DDS) 23
3.3.1.1 Acumulador de fase 25
3.3.1.2 Conversor de fase a amplitud 26
3.3.2 Procesador Nios II 31
3.3.3 Interfaz Serial Estándar de Comunicación (JTAG UART) 32
3.3.4 Periférico paralelo de entrada y salida 33
3.3.5 Módulo de Audio 34
3.3.5.1 Sub-módulo de configuración del CODEC 34
3.3.5.2 Sub-módulo de comunicación con el CODEC 36
3.3.6 Módulo controlador de señales de reloj del sistema 36
3.3.7 Memoria RAM embebida 37
3.3.8 Módulo de Video 37
3.4 Desarrollo del software del sistema 38
3.4.1 Capa de abstracción de hardware (HAL) 39
3.4.2 Descripción funcional del código principal 39
CAPÍTULO 4: RESULTADOS DEL DISEÑO PROPUESTO 40
4.1. Resultados de simulación de los módulos independientes 40
4.1.1. Sintetizador digital Directo (DDS) 40
4.1.2. Módulo de Audio 45
4.2. Resultados utilizando la herramienta SignalTap II 49
4.2.1. Sintetizador digital Directo (DDS) 49
4.2.2. Módulo de Audio 51
4.3. Resultados de la implementación en la tarjeta de desarrollo 53
CONCLUSIONES 56
RECOMENDACIONES 57
BIBLIOGRAFÍA 58

INTRODUCCIÓN

En una situación de emergencia debido a un accidente o desastre natural, las comunicaciones son vitales en las primeras horas de la emergencia. De éstas dependerá la calidad de la ayuda que se pueda brindar.

Los equipos de comunicaciones son diseñados para operar en una banda de frecuencias, esto quiere decir que sólo podrán operar en ese rango predeterminado. En una situación de emergencia, organizaciones como defensa civil han elaborado un plan nacional de respuesta inmediata donde se contemplan los canales de comunicación que se deben establecer. El problema está en que no todos van a poder comunicarse a través de estos canales ya que la mayoría de equipos de comunicaciones que se utilizan en el país se obtienen mediante donaciones. Esto impide que se tengan equipos que operen en la misma banda. Esto generará por ejemplo descoordinaciones durante una emergencia entre bomberos y policías.

El presente trabajo pretende mostrar una alternativa tecnológica al problema mencionado utilizando la tecnología SDR (Software-Defined Radio) que permite diseñar equipos de comunicaciones programables por software. Es decir que las características del tipo de comunicación que puede realizar el equipo se define mediante software o hardware reconfigurable.

La propuesta consiste en realizar un diseño digital basado en FPGA que sea capaz de realizar la etapa de la modulación y selección de la frecuencia utilizando un código en lenguaje C. Se utilizará el CODEC WM8731 como dispositivo para la adquisición de la señal de audio que será procesada en el FPGA, para ello se utilizará la tarjeta de desarrollo Altera DE2 Development kit como hardware para realizar las pruebas respectivas. Todo el tratamiento de la señal se realizará en banda base para luego ser modulada a la frecuencia respectiva utilizando un sintetizador digital directo. La señal modulada se obtendrá utilizando el conversor digital/análogo ADV7123.

La selección de señales FM como señales de prueba se debe a que éstas son las que se utilizan en comunicaciones de equipos de emergencia y además permiten hacer las respectivas pruebas de recepción con un equipo común de recepción FM.

CAPÍTULO 1: LAS COMUNICACIONES INALÁMBRICAS DE EMERGENCIAS EN NUESTRO PAÍS

1.1 Variables que limitan el desarrollo de estos equipos

Dentro del marco problemático encontramos algunas variables externas que pueden caracterizar las comunicaciones de emergencia. Estos equipos han sido modificados y actualizados a medida que avanza la tecnología. Han pasado de equipos totalmente analógicos a equipos que utilizan sistemas digitales pero su función no ha sido modificada. En el caso de nuestro país los equipos que utilizan los diversos servicios de emergencias son adquiridos a través de donaciones o a través de adquisiciones del estado.

Para un correcto uso de los equipos de comunicaciones en situaciones de emergencia el Instituto de Defensa Civil (INDECI) ha desarrollado un plan nacional de respuesta inmediata por el Sistema Nacional de Defensa Civil (SINADECI) ante situaciones de emergencia y/o desastre [1] que permite de alguna manera ordenar las comunicaciones que se realicen. Dentro de este plan se considera las frecuencias a las cuales deben comunicarse los equipos además contempla la capacitación de personal calificado para operar los equipos.

Las comunicaciones juegan un papel importante dentro del marco de respuesta ante emergencias o desastres. Es parte del servicio que se tiene que brindar a la sociedad frente a una necesidad como lo es la ayuda ante situaciones de riesgo.

También tenemos que tomar en cuenta que estos equipos necesitan un mantenimiento preventivo periódico por parte de un personal técnico calificado que permita el buen funcionamiento. Esto implica que si la tecnología de los equipos cambia también la capacitación de éstos tiene que actualizarse.

Ningún equipo de comunicaciones puede operar sin las licencias que otorga el Ministerio de Transportes y Comunicaciones que vela que los equipos tengan las características necesarias para operar correctamente sin interferir a otras bandas de frecuencias por ejemplo. Además los equipos de comunicaciones que se utilizan actualmente siguen siendo los de hardware digital. La tendencia actual se centra en los equipos de comunicaciones que se definen por software.

Además de las normas que da el Ministerio de Transportes y Comunicaciones existen también organizaciones mundiales como la ITU (International Telecommunication Union) que dan algunas normas con respecto a equipos por ejemplo en lo que concierne a temas de regulación y seguridad. El tema de comunicaciones de emergencia no puede dejarse de lado ya que es parte importante del desarrollo de las urbes modernas. Cada día aumenta la densidad poblacional y las ciudades crecen rápidamente, esto aumenta la posibilidad de accidentes o desastres de gran magnitud. Tener un sistema de comunicaciones eficiente es la clave para que la ayuda que se les brinde sea óptima.

1.2 Proceso de comunicación durante una emergencia

Cuando sucede un accidente o sucede algún desastre de gran magnitud en la gran mayoría de casos existirá algunas comunicaciones desde el lugar del desastre. Este mensaje será para comunicar a las autoridades competentes sobre la gravedad de la situación. Hasta este punto se pueden haber utilizado varios medios de comunicación Ej. Telefonía fija, telefonía celular, etc. por lo que no es tan crítico.

Luego se producirá comunicaciones desde los centros de emergencia hacia las unidades de apoyo, cabe mencionar que estas comunicaciones por lo general se realizan a través de medios inalámbricos a una frecuencia preestablecida. Por lo general, cada grupo de apoyo posee sus propias frecuencias de comunicaciones esto hace imposible coordinación alguna entre individuos de estos grupos.

Una vez que llegan a la zona de emergencia las personas sólo podrán comunicarse si poseen equipos de comunicaciones capaces de transmitir en la misma banda de frecuencias. Además hay que agregar que sólo se puede utilizar un canal para hacer una comunicación en un tiempo y lugar dado. Esto hace necesario el uso de muchos canales de comunicación que algunas veces es insuficiente dependiendo de la gravedad de la situación.

Un evento de gran magnitud no ajeno a nuestra realidad puede ser los periódicos fenómenos del niño que afectan a nuestro país y los eventos sísmicos debido a nuestra ubicación geográfica. El hecho de no existir una comunicación fluida generará fallas de coordinación o tal vez duplicidad de la ayuda haciendo más dura las labores de rescate.

1.3 Declaración del marco problemático

Las comunicaciones inalámbricas son de gran importancia durante una emergencia. Estas permiten que la ayuda llegue al lugar de la emergencia en forma ordenada y efectiva. Para lograr esto todos los elementos que participan en la ayuda de una emergencia tienen que estar comunicados.

El problema más común que se presenta está en que los equipos de emergencia de distintas localidades poseen distintos equipos de comunicaciones por lo tanto no es posible la comunicación entre estas personas, generando descoordinaciones que pueden disminuir la calidad de la ayuda brindada.

El problema de la incompatibilidad de equipos de comunicaciones en nuestro país pasa también por el tema de adquisición de éstos. La mayoría de equipos que tienen las unidades de bomberos, defensa civil, etc. son gracias a donaciones por lo tanto no hay el cuidado necesario para hacer que los equipos sean compatibles.

CAPÍTULO 2: TECNOLOGÍA SDR: EQUIPOS DEFINIDOS POR SOFTWARE

2.1 Introducción

Las comunicaciones inalámbricas actuales nos dan la ventaja de tener acceso a la información en cualquier lugar y en cualquier momento. Éstas han permitido que las empresas dedicadas a las comunicaciones puedan aumentar rápidamente la cantidad de sus suscriptores sin disminuir la calidad del servicio. En el caso de voz, es comparable al servicio telefónico tradicional. La amplia variedad de tecnologías inalámbricas nos permite contar con una diversidad de servicios, sea de voz o datos, sin embargo ésta a su vez trae consigo una incompatibilidad intrínseca a éstas y problemas de gestión del espectro electromagnético. Este problema ha causado que el centro de atención de las telecomunicaciones se enfoque en las comunicaciones inalámbricas.

En este contexto aparece la tecnología *software-defined radio* (SDR) que permite que un equipo pueda ser configurado vía software y así poder cambiar de banda y de tecnología en el momento y lugar que se requiera.

El presente estudio muestra la evolución de las comunicaciones inalámbricas hasta el contexto actual que ha generado la aparición de la tecnología SDR, como una solución alternativa a los problemas de las comunicaciones actuales.

2.2 El concepto de software radio

El término *software radio* se entiende por funciones de comunicación implementadas por software, es decir la posibilidad de definir por software la interfaz de comunicación que normalmente se refieren al transmisor y receptor del equipo de comunicaciones. Esta característica, de la definición de la interfaz vía software, implica el uso de procesadores de señal digital (DSPs) ó FPGAs para reemplazar el hardware dedicado y poder ejecutar en tiempo real el software necesario.

Estas son algunas otras definiciones de otros autores acerca del *software radio:* [2]

1. Arquitectura flexible de transmisión y recepción, controlado y programado vía software.

2. Procesamiento de señal capaz de reemplazar, cuanto sea posible, las funciones de radio.

3. Transmisor-receptor donde lo siguiente puede ser definido por software:
   - Ancho de banda de canal y frecuencia
   - Modulación y codificación
   - Administración de protocolos.
   - Aplicaciones de usuario

   Estos parámetros pueden ser cambiados por el operador de la red, el proveedor del servicio o el usuario final.

La flexibilidad de un sistema *software radio* radica en su capacidad de operar en ambientes multiservicio, es decir sin estar restringido a un estándar en particular pero con la posibilidad de funcionar en cualquier sistema existente o en uno que aún no ha sido definido. Es importante recalcar que el procesamiento digital de señales no sólo se puede hacer en un procesador DSP, existen también tecnologías como los FPGAs capaces de implementar estas funciones además de otros procesadores de propósito general como el INMOS transputer ó el Intel Pentium/MMX [2].

2.3 Problemas de diseño en un equipo SDR

El desarrollo de un sistema *software radio* implica el logro de estas dos metas [2]:

1. Mover, en el transmisor y receptor, el borde entre el mundo digital y el analógico a la frecuencia de comunicación (RF), usando conversores análogo-digital y digital-análogo, lo más cerca posible a la antena.

2. Reemplazar los circuitos de aplicación específica (ASICs) con DSPs para el procesamiento de señal en banda base, definiendo la mayor cantidad de funciones en software.

Actualmente, los transmisores y receptores de los equipos de radiocomunicaciones están basados en el tradicional esquema superheterodino (Figura 2.1), donde las etapas de RF (radio frecuencia) e IF (frecuencia intermedia) son totalmente analógicas.

Por el contrario un esquema ideal de un receptor SDR tiene una etapa analógica reducida: la antena, el filtro pasabanda y el amplificador de bajo ruido (LNA).

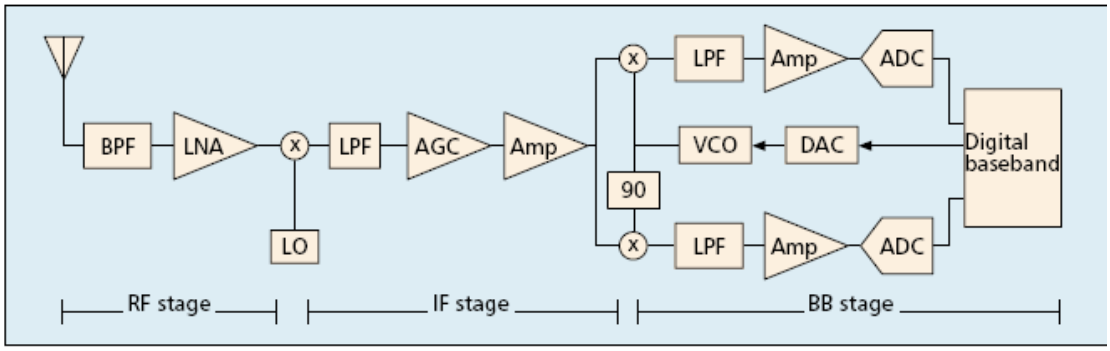


Figura 2.1. Un receptor tradicional superheterodino.[2]

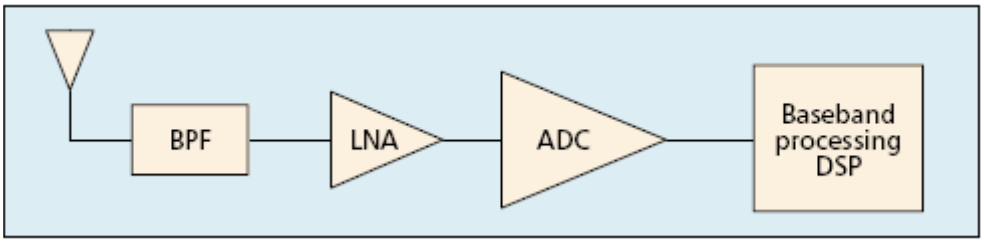


Figura 2.2. Un receptor SDR ideal.[2]

El receptor mostrado en la figura 2.2 es ideal por una serie de aspectos que lo hacen difícil de realizar.

No es razonable el uso de una sola etapa de RF para un sistema multibanda, la única forma de garantizar la capacidad de multibanda es teniendo más etapas RF.

Además, el efecto *"jitter"* o rizado provocado por la conversión A/D directa de RF hace que sea complicado su diseño.

2.4 Investigaciones académicas

El interés por la tecnología SDR no sólo radica en la industria comercial y militar también está en lo académico.

En el Georgia Institute of Technology [3], a través de fondos del programa para el desarrollo económico del estado de Georgia, EE.UU., investigadores están desarrollando un sistema de comunicaciones inalámbricas de alta velocidad usando SDR. El proyecto tiene dos objetivos principales:

1. Desarrollar comunicaciones inalámbricas móviles totalmente adaptables. Esto incluye la coordinación de los cambios en la aplicación, red, acceso y capas físicas de la red cuando ésta cambia de un estándar a otro.

2. Desarrollar un sistema inalámbrico de alto rendimiento. Esto incluye el uso de antenas adaptativas y algoritmos de codificación para combatir los efectos de las variaciones de canal en el tiempo y espacio.

Este sistema de desarrollo SDR incluye RF front-ends, ADCs, DACs, DSPs etc. Éste sirve como un prototipo funcional para la implementación, prueba e integración de esta investigación desarrollada por miembros de este instituto.

En el Laboratorio para Computer Science del Massachussets Institute of Technology (MIT) [3] han desarrollado una computadora de bolsillo multipropósito capaz de combinar las funciones de de un celular, conexión inalámbrica a Internet, un localizador, una radio AM/FM y receptor de TV. Es una de las aplicaciones más innovadoras de SDR. Este dispositivo reemplaza muchos otros dispositivos. Para seleccionar las funciones se realiza de manera similar a la selección de un programa con el mouse con la PC. El equipo en esencia es una PC equipada con una antena y un ADC. El hardware permite la selección de cualquier región de 10 MHz del espectro dentro del rango permitido del dispositivo, baja la señal hasta una frecuencia intermedia (I.F.) y luego la señal es almacenada en la memoria principal de este dispositivo para realizarle su posterior procesamiento digital de señales.

2.5 Aplicaciones comerciales

Actualmente no existen muchas aplicaciones comerciales que mencionar, ya que este concepto recién esta siendo aplicado a estándares comerciales. La empresa Harris Corporation es una de las pocas que ofrece varios equipos de comunicaciones basados en tecnología SDR. Uno de estos es el conjunto de equipos FALCON II, diseñado para su uso en el campo de batalla. Integra equipos que trabajan en las bandas de HF, VHF y UHF. Permite la interoperabilidad entre equipos de comunicaciones para operaciones militares tácticas tierra – tierra, tierra - aire y tierra-satélite.

Figura 2.3. FALCON II: equipo SDR comercial.[4]

2.6 Beneficios de la tecnología SDR

Uno de los principales beneficios es la reducción de costos. Tradicionalmente los equipos de comunicaciones se producían en base a circuitos integrados de aplicación especifica (ASIC) pero estos tienen una alta inversión inicial por lo tanto para hacerlo económicamente viable se deben producir a gran escala. Con la tecnología SDR se pueden reducir los costos ya que se van a reducir la cantidad de ASICs usados en el equipo y la producción a baja escala no es tan costosa. A esto último ha contribuido mucho el uso de FPGAs que ya no se toman como chips para prototipos sino que han llegado a una madurez tal que puede ser usado en productos finales.

Otro aspecto importante es el hecho que el equipo haya sido definido por software implica que si ha ocurrido un error en el diseño (caso cada vez más probable

debido al aumento de la complejidad en los estándares), éste puede ser corregido sin tener que cambiar el hardware.

La alta adaptabilidad al medio de los equipos basados en tecnología SDR le permite prolongar el tiempo de vida que tendrá el equipo. Estándares posteriores a la creación podrían ser insertados en el equipo con una simple actualización de software.

2.7 Síntesis sobre el asunto de estudio

La tecnología SDR se encuentra en su primera etapa y se proyecta como una alternativa de solución al problema de las comunicaciones actuales. Todavía existen muchos obstáculos que resolver pero el avance en este campo de las telecomunicaciones cada vez es mayor. La evolución del concepto de SDR desde su primera aparición en el mundo científico ha tenido algunas modificaciones debido al contexto actual de la tecnología necesaria para la implementación de ésta [5]. Nos podría traer muchos beneficios pero también hay que aclarar que se tiene que pagar un precio por tan alta flexibilidad.

2.8 Conceptos Generales acerca de un equipo definido por software

2.8.1 Equipos de Comunicaciones inalámbricas

2.8.1.1 Comunicaciones inalámbricas comerciales

Existen muchos estándares y protocolos para comunicaciones inalámbricas que nos brindan soluciones a las necesidades de comunicaciones. Esta tendencia ha generado un uso indebido del espectro electromagnético. Existen varios tipos de comunicaciones inalámbricas: conexión punto a punto, punto a multipunto, de jerarquía simple y de jerarquía múltiple.

En la siguiente figura se muestra el contexto actual de las comunicaciones inalámbricas comerciales:

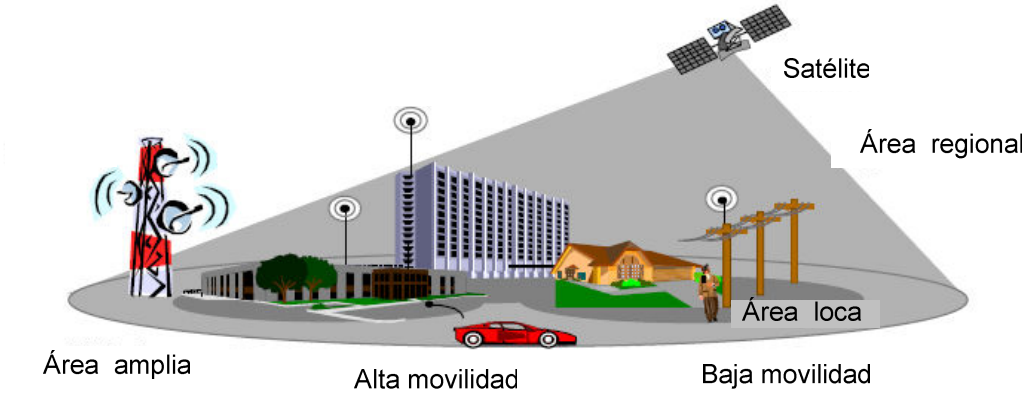


Figura 2.4. Contexto actual de las comunicaciones inalámbricas. [6]

Actualmente tenemos comunicaciones inalámbricas de todo tipo y en todo lugar. Tenemos comunicaciones con satélites para obtener nuestra ubicación, *Global Positioning System* (GPS), también tenemos redes celulares que trabajan con diferentes tecnologías, *Global System of Mobile Communications* (GSM), *Code División Multiple Access* (CDMA). Tampoco debemos olvidarnos de las nuevas tecnologías en redes inalámbricas como Wifi o WiMax. Cada una de estas tiene una ubicación dentro del espectro electromagnético con su respectivo ancho de banda.

2.8.1.2 Comunicaciones inalámbricas militares

Se aplican para las comunicaciones tácticas entre los diferentes elementos de una fuerza militar. Esta comunicación debe ser lo mas rápida y fluida posible pero ya que los equipos no trabajan en la misma banda ni utilizan el mismo protocolo entonces esto resulta algo difícil. Existen tantos protocolos como equipos de comunicaciones existen en las fuerzas armadas.

2.8.2 Tecnología SDR

2.8.2.1 Definición

Todavía no existe un consenso acerca de una definición exacta pero muchos autores concuerdan que tiene que ser una arquitectura flexible de transmisión y recepción controlada y programada por software. Otra característica debe ser que el procesamiento digital de señales debe ser capaz de reemplazar las funciones de

radiocomunicación que tradicionalmente se realizaban con hardware específico. La capacidad para que el equipo pueda ser configurado dinámicamente a distancia no es característica esencial de todos los autores [2] [5].

2.8.2.2 Arquitectura

Concepto original

La arquitectura original planteada pretende que todas las funciones del equipo se realicen mediante procesamiento digital de señales y por lo tanto colocar los conversores digital – analógicos y viceversa lo más cerca posible a la antena.

Nuevas Tendencias

El gran problema de la arquitectura original es su inviabilidad tecnológica, no es posible aún tener los componentes necesarios para tal implementación. Se plantean múltiples soluciones usando componentes de hardware específico que permitan llegar a una aproximación.

2.8.3 FPGA

2.8.3.1 Definición

Por sus siglas en inglés es un arreglo de compuertas lógicas programables en campo capaz de sintetizar funciones lógicas programadas por el usuario final. Puede implementar todo tipo de funciones lógicas combinacionales y secuenciales [7].

2.8.3.2 Arquitectura de un FPGA

Los FPGAs son producidos a base de celdas lógicas interconectadas. Dependerá del fabricante el nombre que se le asigne a esta unidad básica. Estas unidades básicas por lo general están compuestas por flip-flops, multiplexores y LUTs (*look up tables*) que le permiten generar cualquier tipo de función lógica. La estructura típica de un FPGA se muestra en la siguiente figura:

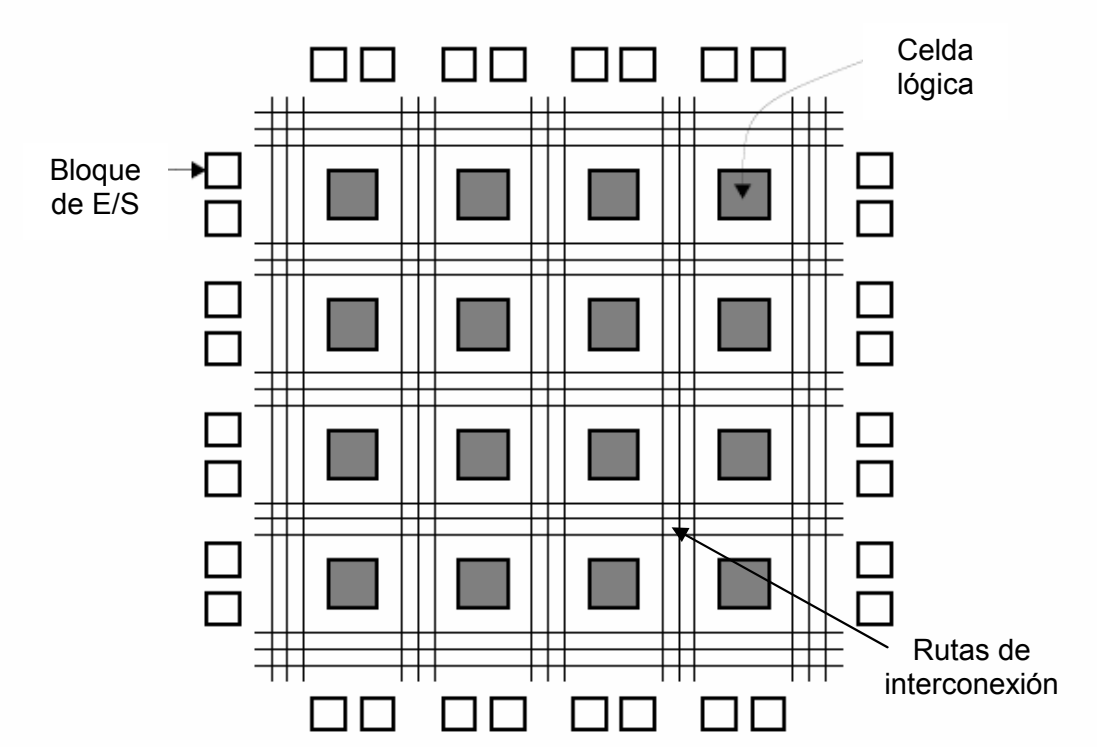


Figura 2.5. Arquitectura genérica de un FPGA. [7]

En la actualidad, los FPGAs también poseen bloques de memoria embebidas en el chip. Esta memoria puede ser de dos tipos:

1. Memoria implementada usando LUTs. Denominada *Distributed RAM* (*Random access memory*) [7].

2. Memoria usando bloques designados para tal fin. Denominada *Block RAM* [7].

Además poseen circuitería especializada en hacer el enrutamiento de las señales internas del FPGA. Existen rutas especiales para el uso de señales importantes en los circuitos digitales como por ejemplo la señal de reloj para disminuir los problemas de llegada de la señal de reloj a destiempo conocido como *clock skew* y violaciones de los tiempos de seteo y retención [7][8].

Los pines de este dispositivo pueden ser configurados como entradas o salidas y todos poseen registros para poder almacenar estos valores. (*Buffered pins*).

Las últimas versiones de estos dispositivos contienen además procesadores embebidos dando como resultado un verdadero sistema en el chip (*System on Chip*).

2.8.3.3 Arquitectura del FPGA EP2C35 de Altera

El FPGA EP2C35 utilizado en el presente trabajo pertenece a la familia Cyclone II que está basada en un proceso de 1.2V, 90-nm SRAM. Posee una alta densidad de elementos lógicos (LEs) que alcanzan los 68416 y contiene memoria RAM embebida que alcanza los 1.1Mbits de capacidad [14].

Los FPGAs de esta familia poseen multiplicadores embebidos de 18 x 18 bits, PLL´s para la administración del reloj del sistema e interfaces externas de alta velocidad para la conexión con memorias SRAM y DRAM. La cantidad de cada uno de estos componentes dependerá del dispositivo elegido [14].

Este FPGA contiene una arquitectura bidimensional compuesta por filas y columnas, entre las cuales se encuentran los bloques de lógica. En el perímetro del dispositivo se encuentran los IOEs (Input Output Elements), que permiten el manejo de las señales de entrada y salida. Las interconexiones de filas y columnas permiten que se transmitan las señales eléctricas entre los distintos componentes. Los arreglos lógicos están organizados en LABs (Logic Array Blocks), que son un conjunto de LEs; bloques de memoria (M4K) y multiplicadores embebidos. La disposición de estos elementos se encuentra en la figura 2.6.

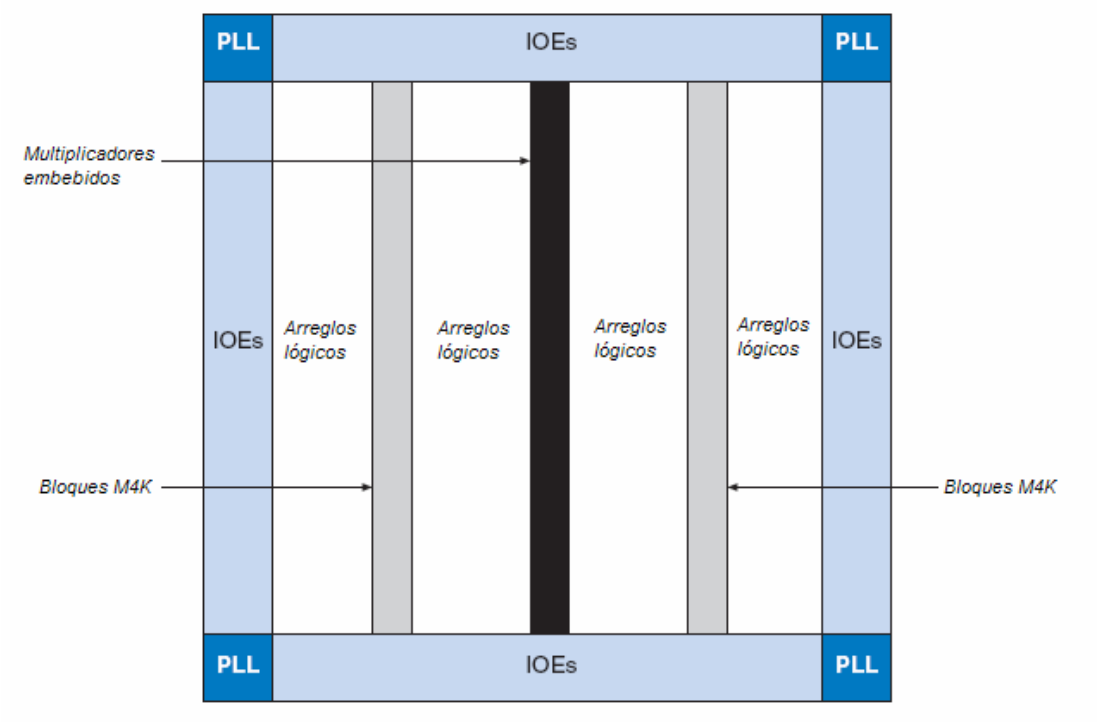


Figura 2.6. Diagrama de bloques del FPGA Cyclone II. [14]

Cada LE está compuesto principalmente de un registro programable y de un LUT (Look up table) de 4 cuatro entradas. En el caso que se desee implementar una función combinacional, la salida del LUT evita pasar por el registro y se conduce directamente a la salida del LE.

El registro programable puede ser configurado como tipo D, T, JK o SR. Además, tiene entradas de data, clock, clock enable y clear. Se puede elegir una de las dos señales de reloj (labclk1 o labclk2) para controlar el registro. Si se elige la señal labclk1, entonces la señal labclkena1 es la que funciona y viceversa. Otra característica importante es que la salida del registro puede retro alimentar al LUT. La estructura de un LE se muestra en la figura 2.7.

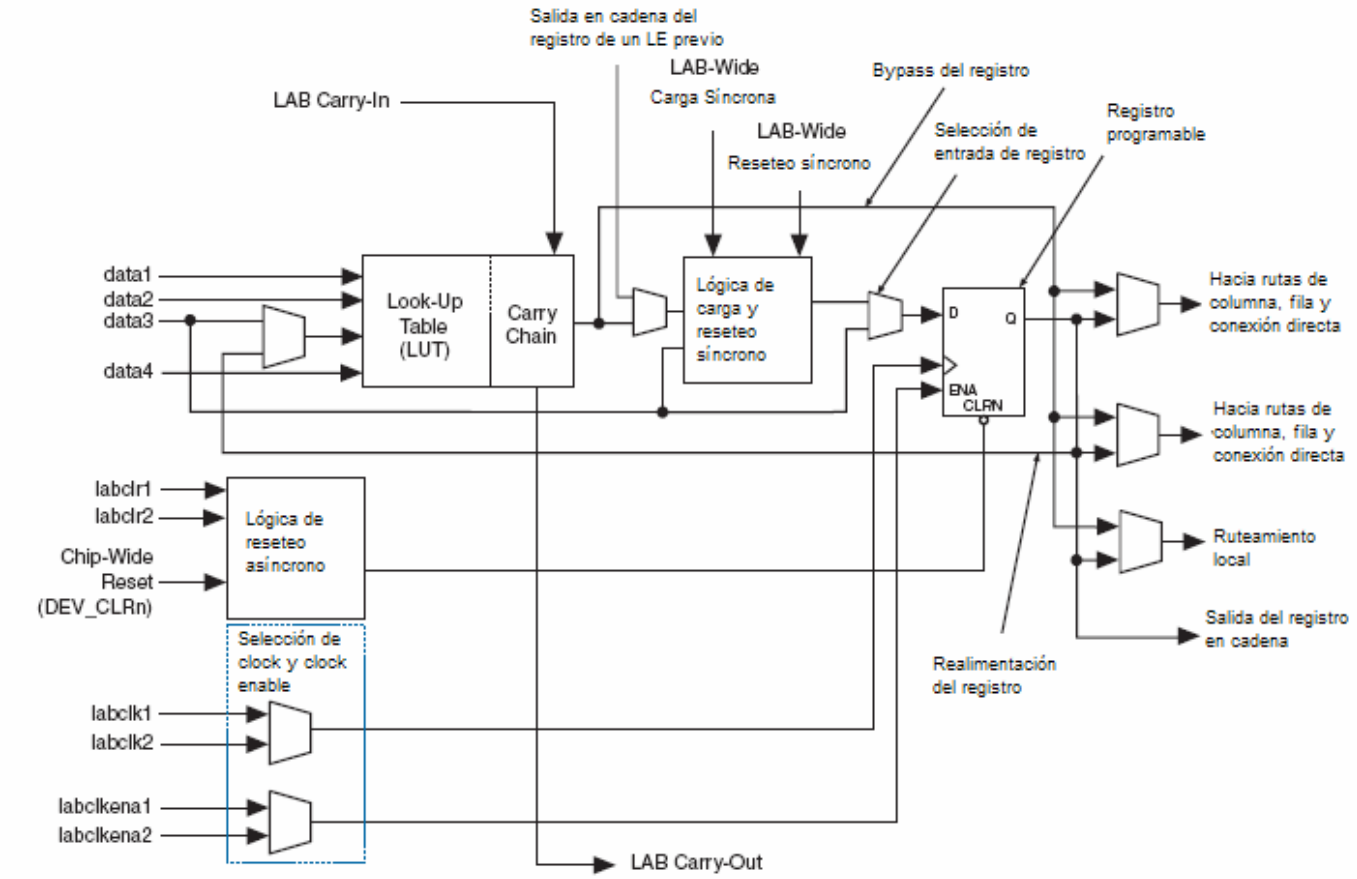


Figura 2.7. Diagrama de bloques de un LE del FPGA Cyclone II. [14]

2.9 Modelo Teórico

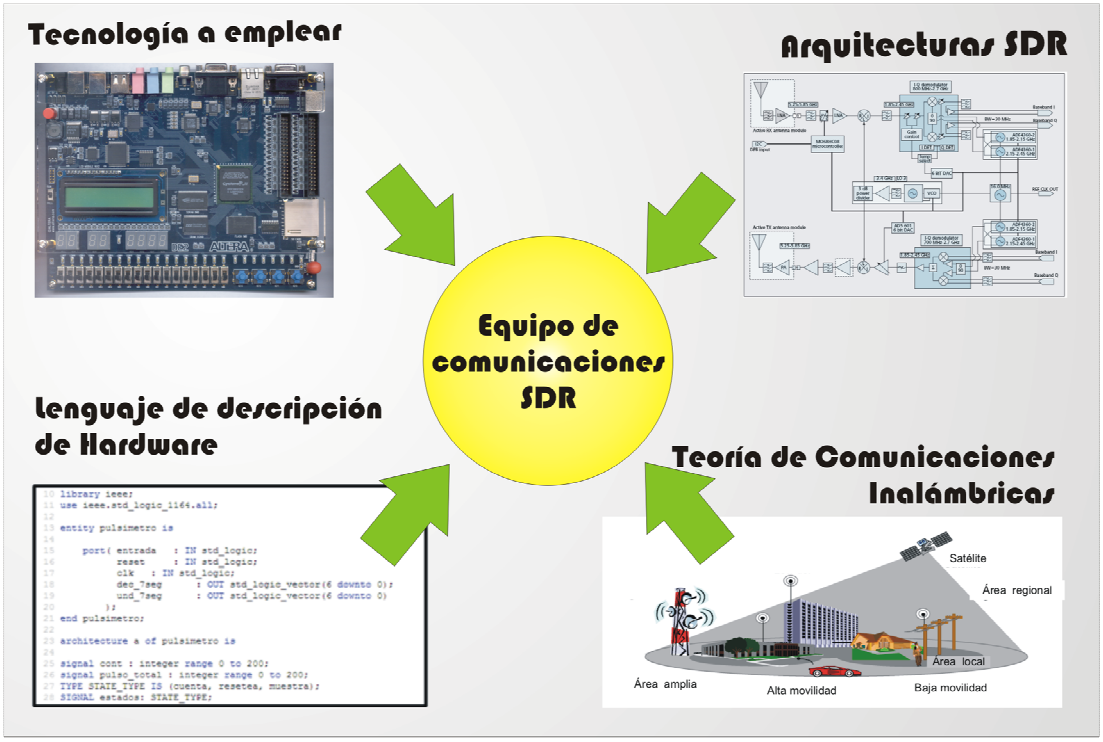


Figura 2.8. Gráfica del Modelo Teórico.

Para poder diseñar un equipo SDR se tiene que tomar en cuenta el desarrollo tecnológico que se viene dando en las comunicaciones inalámbricas, sin esto no se entendería claramente que se tiene que hacer. La teoría de comunicaciones inalámbricas nos va a dar los conceptos necesarios para el desarrollo de un equipo de comunicaciones digital, como por ejemplo las arquitecturas que existen para el desarrollo de estos equipos así como las regulaciones que existen para los estándares actuales de comunicaciones inalámbricas.

Existen arquitecturas definidas para realizar procesos como la modulación/demodulación, la codificación de línea, la multiplexación digital síncrona, el control de errores, etc.

La tecnología SDR posee una arquitectura ideal original de la cual han ido desarrollándose las actuales. Depende de la aplicación que se le quiera dar a la tecnología SDR existirá una arquitectura adecuada.

Toda esta idea de realizar equipos de comunicaciones definidos por software no podría ser realizable sino fuera por la tecnología actual. Los FPGAs se presentan como los dispositivos idóneos para implementar las arquitecturas SDR. Los FPGAs son dispositivos lógicos programables capaces de implementar cualquier función lógica (combinatoria o secuencial) con la propiedad de ser reconfigurado cuando se requiera. De nada serviría tener un dispositivo programable sino tuviera ese alto desempeño que tienen los FPGAs actuales para poder implementar estas arquitecturas SDR que requieren de una alta capacidad de procesamiento. Los FPGAs son configurados por el usuario, pero es éste el que debe decidir como es que los va a configurar.

Es aquí que entra a tomar protagonismo el lenguaje de descripción de hardware que usemos para tal tarea. Existen diversidad de maneras para hacer esto, una de éstas es el VHDL (*Very High Speed Hardware Description Language*). Se requiere el conocimiento de la sintaxis que posee el VHDL y además conocer cuales partes de la sintaxis del VHDL pueden ser implementadas en un FPGA.

2.10 Definiciones operativas

2.10.1 Indicadores Cuantitativos

- **Frecuencia máxima de operación**

La frecuencia máxima de operación nos permitirá saber si es posible la implementación del sistema. Este parámetro es proporcionado por el sintetizador del entorno de descripción de hardware.

Cantidad de LEs: Es un parámetro referido al área utilizada por el sistema diseñado. Esta área para el caso de un FPGA se contabiliza mediante la cantidad de elementos lógicos utilizados ó LEs.

CAPÍTULO 3: DISEÑO DE LA ARQUITECTURA PROPUESTA PARA EL MODULADOR FM

3.1 Hipótesis de la investigación

3.1.1 Hipótesis principal

Dado que actualmente los sistemas de comunicación implementados utilizan equipos que están diseñados para trabajar a una frecuencia definida del espectro electromagnético (E.M.) y además la aparición de nuevas tecnologías de comunicación inalámbrica generan la saturación y el uso indebido del E.M. entonces se plantea el diseño de un modulador FM definido por software que permita programar la frecuencia de operación.

3.1.2 Hipótesis secundarias

1) En la actualidad el desarrollo de los procesadores digitales de señales hace posible la implementación de sistemas de comunicaciones SDR en hardware programable en tiempo real por ello se plantea el diseño de un sistema embebido en el chip (SoC) que verifique esta afirmación utilizando uno de estos dispositivos en este caso un FPGA.

2) La etapa de procesamiento de la señal, que en un equipo de tecnología SDR se realiza mediante software, se puede realizar utilizando hardware flexible que pueda ser reconfigurado. Se propone mostrar una arquitectura que realice el procesado de la señal en tiempo real en un FPGA.

3) Las pruebas se realizarán utilizando tarjetas de desarrollo que permitan mostrar el funcionamiento del bloque de procesado de señal. Para esto se pretenderá utilizar frecuencias que se utilizan en comunicaciones de equipos de emergencias en nuestro país.

3.2 Objetivos de la investigación

3.2.1 Objetivo general

Diseñar un modulador FM SDR que permita la transmisión de una señal analógica a una frecuencia específica. En este caso frecuencias que son utilizadas por equipos de apoyo civil como bomberos.

3.2.2 Objetivos específicos

1) Diseñar el módulo en VHDL que sea capaz de modular una señal analógica de radiocomunicación utilizando datos obtenidos de un computador (sonido pre-grabado) y de un micrófono.

2) Diseñar utilizando un lenguaje de programación C que sea ejecutado por el procesador embebido del sistema y que realice el control del flujo de datos en el sistema y la adaptación de la señal de audio.

3) Realizar pruebas utilizando las bandas de frecuencias que se utilizan en comunicaciones de emergencia haciendo uso de la tarjeta de desarrollo DE2 Altera Development Kit.

3.3 Desarrollo del hardware del sistema

El diseño del hardware del sistema se realizó utilizando la herramienta SoPC Builder del Quartus II [9] y utilizando el lenguaje de descripción de hardware VHDL. Para este sistema se han usado módulos estándar de la herramienta y los que provee el programa universitario de Altera. El esquema general del sistema es el siguiente:

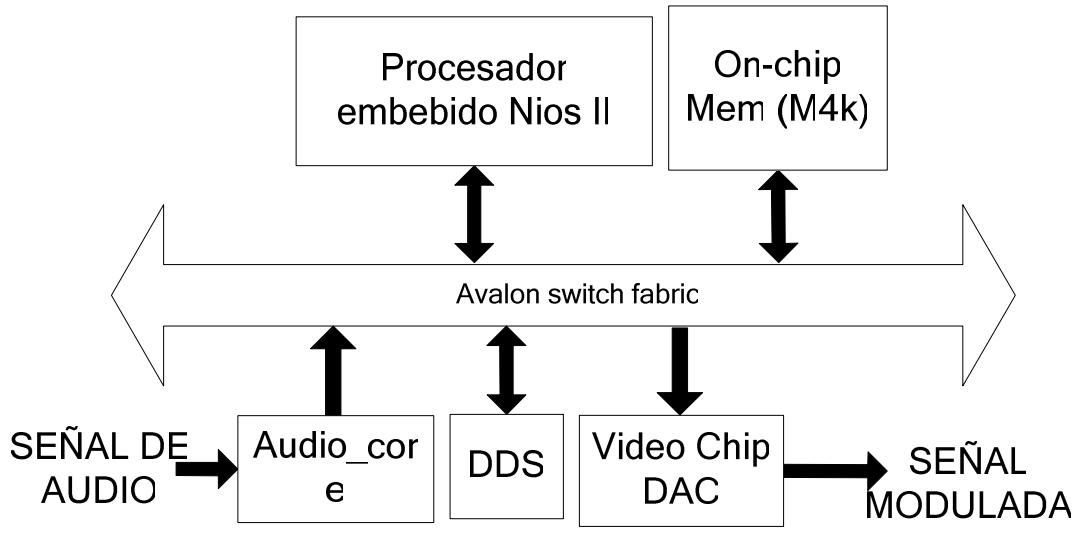


Figura 3.1. Diagrama de bloques del sistema.

La herramienta SoPC builder nos brinda la plataforma para crear el sistema embebido. En esta herramienta encontraremos los módulos necesarios para poder integrar componentes como el procesador Nios II, memorias RAM embebidas, otros periféricos, etc. Pero además de esto también nos permite interconectar los módulos que han sido diseñados con VHDL. Esta herramienta utiliza la arquitectura de un bus propietario de comunicaciones internas para poder interconectar todos los módulos del sistema. Este bus es llamado Avalon Switch Fabric. Posee un conjunto de señales y lógica digital para comunicar los módulos esclavos o maestros, entre los cuales se encuentra la decodificación de la dirección, multiplexación de las señales de datos, generación de estados de espera, arbitraje del bus y controlador de interrupciones.

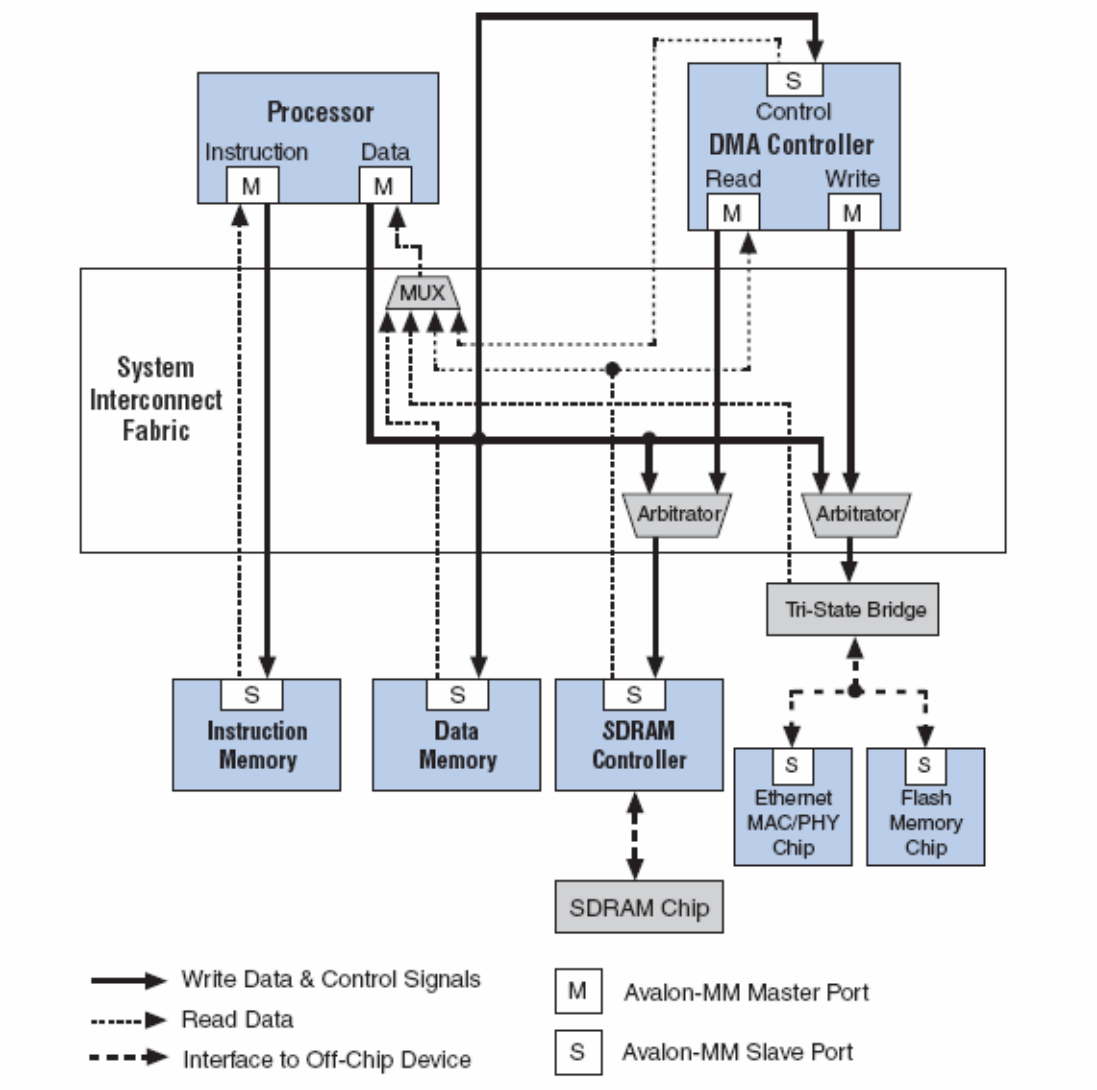


Figura 3.2. Diagrama de bloques del bus del sistema.[9]

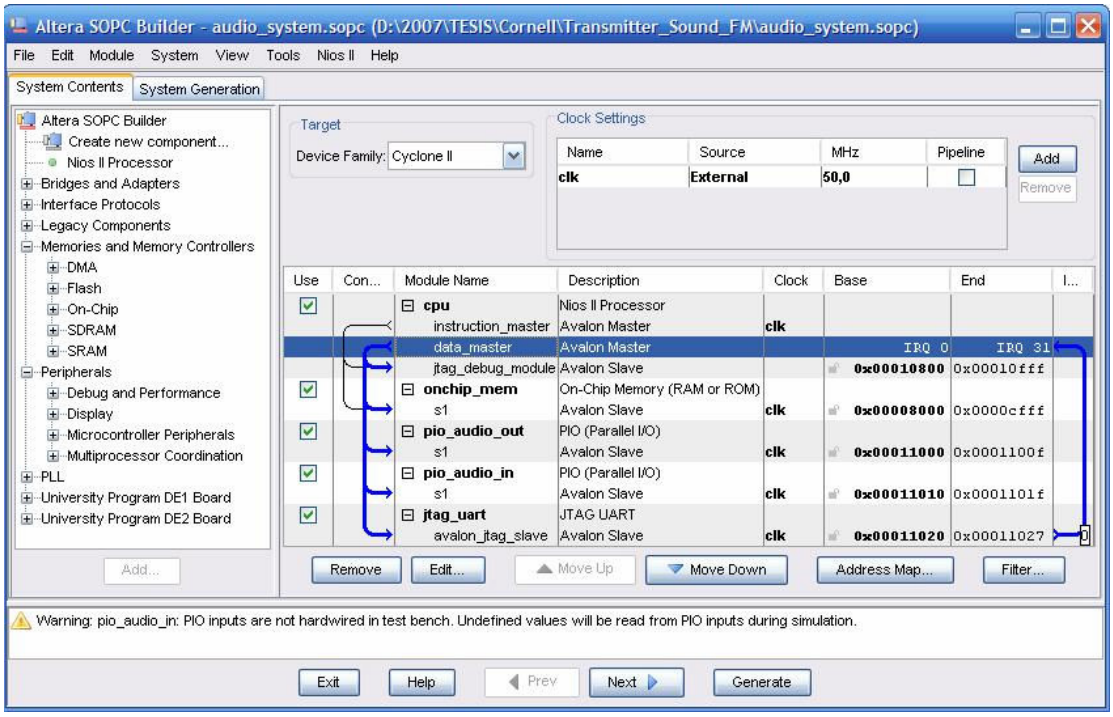


Figura 3.3. Arquitectura del sistema en el entorno del SOPC Builder.

3.3.1 Sintetizador Digital Directo (DDS)

Este módulo nos permite modular en frecuencia la señal de audio y en general nos permite la generación de señales periódicas. Este bloque también es conocido como oscilador controlado numéricamente o NCO. Existen varios métodos para generar una señal analógica como el uso de PLLs. Las ventajas de este método sobre los otros están tanto en el consumo de potencia y el control fino de la frecuencia de salida debido a su naturaleza digital [10]. En este caso se ha realizado un diseño personalizado para trabajar con el ADC del CODEC de audio [12] y el DAC del chip de video [13]. El principio de funcionamiento del DDS es acumular un cambio de fase a una frecuencia mayor o igual al doble de la frecuencia a generar (Principio de Nyquist). Con lo anteriormente mencionado se genera la forma de una onda digitalizada utilizando una conversor de fase a amplitud. Este conversor es una tabla de valores almacenados en una ROM. Con esta tabla podemos generar cualquier tipo de onda periódica. Este módulo va a ser utilizado para generar una señal senoidal de pruebas y para generar la señal portadora del sistema, que en este caso será una señal cuadrada debido a limitaciones del DAC.

A continuación se muestra la estructura diseñada del DDS:

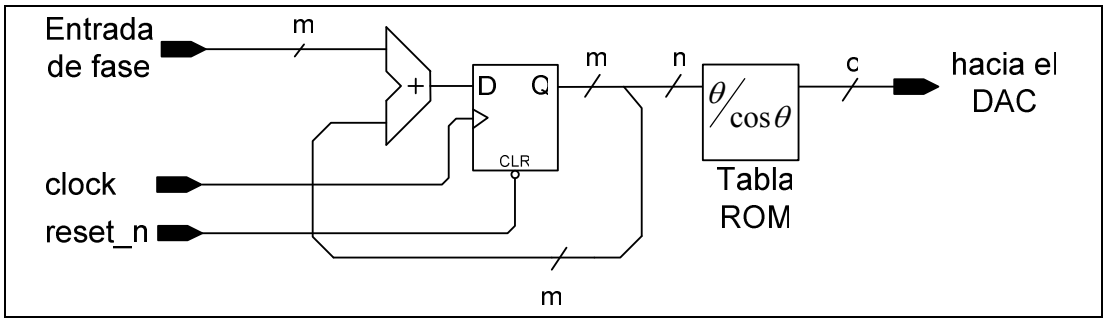


Figura 3.2. Estructura del DDS.

Donde m representa la cantidad de bits de la entrada de fase que permite controlar la frecuencia. n representa los n bits más significativos de m. Es un truncamiento que permite una reducción significativa de la tabla ROM y también genera un problema explicado mas adelante. o representa la cantidad de bits de salida que van hacia la entrada del DAC. De la gráfica anterior se puede deducir que la fórmula para el cálculo de la frecuencia de salida es la siguiente:

$$F_0 = \Delta\theta * \frac{F_{clk}}{2^m} \quad (1)$$

Donde:

- $\Delta\theta$ representa el cambio de fase.
- F_{clk} es la frecuencia de actualización del registro, en este caso la frecuencia del reloj del sistema.
- m es la cantidad de bits del cambio de fase.

El truncamiento de los n primeros bits a la salida del registro van a generar componentes espectrales no deseados a la salida. El peor caso (analizado en [14] y en [15]) aparecerá cuando $\Delta\theta$ sea un múltiplo de 2^{m-n-1} y la magnitud del mayor componente se definirá como:

para $m - n > 4$:

$$WCSM \approx 6.02 * n(dB) \quad (2)$$

A continuación se describirán los módulos principales que componen el DDS.

3.3.1.1 Acumulador de fase

Este pequeño módulo permite generar el incremento de fase que se necesita para generar la señal que deseamos. La salida de este módulo será la entrada de la dirección de la memoria ROM que contiene la tabla de conversión de fase a amplitud. Está compuesto por un sumador de 16 bits y un registro de igual cantidad de bits, posee una señal de reset asíncrono y la entrada del valor de la fase o palabra de ajuste de frecuencia.

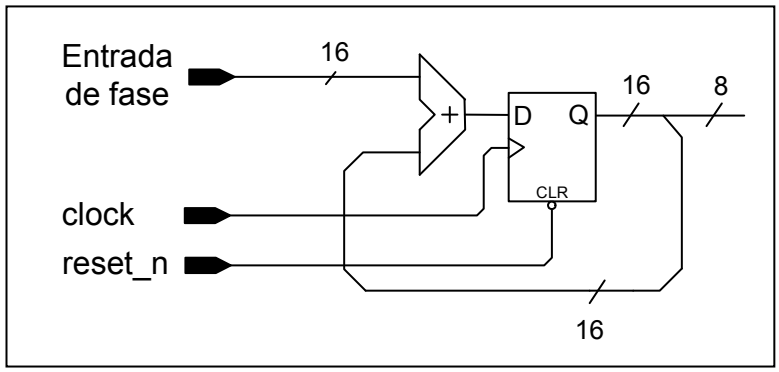


Figura 3.3. Acumulador de fase.

Para entender el funcionamiento de este bloque podemos imaginarnos la onda senoidal como un vector rotando alrededor de un círculo de fase (Figura 3.4) Cada punto designado en la rueda de fase corresponde a un punto equivalente dentro de un periodo de la onda senoidal. A medida que el vector se desplaza por la rueda, el seno del ángulo generado corresponderá a un valor de la onda generada. Una vuelta completa a esta rueda a una velocidad constante corresponde a un periodo completo.

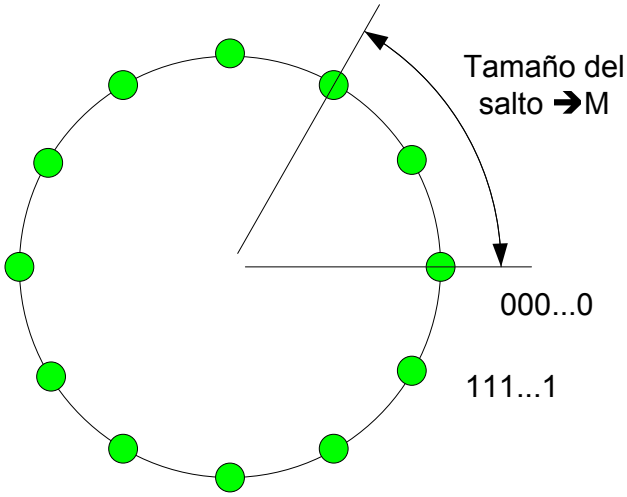


Figura 3.4. Rueda de fase digital.

3.3.1.2 Conversor de fase a amplitud

El conversor de fase a amplitud se basa en una memoria ROM implementada en el FPGA utilizando los bloques de memoria M4K [16] del CYCLONE II. Para crear esta ROM se utilizó el MATLAB para crear una función que generará de manera automática el código en VHDL de la ROM según las recomendaciones de diseño de la compañía Altera para la correcta síntesis de ROMs [17]. En este caso la función que se implementa en esta ROM es la función seno y una onda cuadrada. Para la ROM que implementa la función seno se diseñaron dos arquitecturas distintas. La primera implementa un periodo completo de la función seno (figura 3.6), mientras que el otro sólo implementa un cuarto de la onda (figura 3.7). Con está última, logramos una compresión de ROM de 4 a 1. Para el caso de la onda cuadrada, en realidad no se implementa una ROM sino más bien un comparador de la salida del acumulador de fase truncada con un valor umbral. El valor umbral de este comparador es la mitad del valor máximo que se puede generar este acumulador de fase truncado para generar una onda cuadrada.

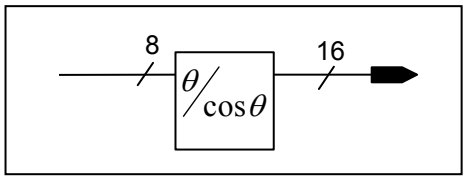


Figura 3.5. Conversor de fase a amplitud.

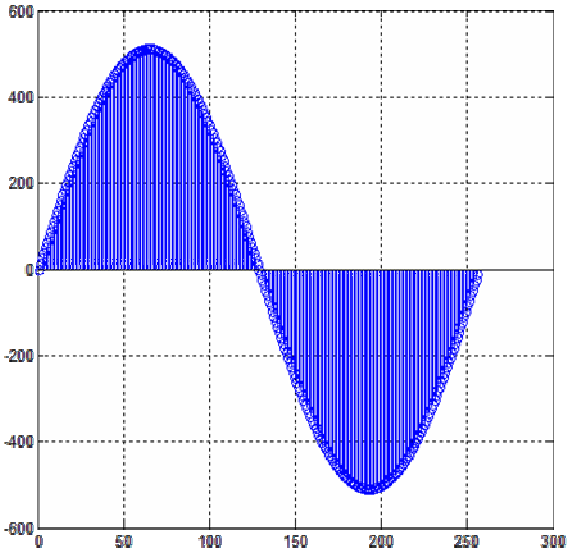


Figura 3.6. Valores para la tabla de periodo completo.

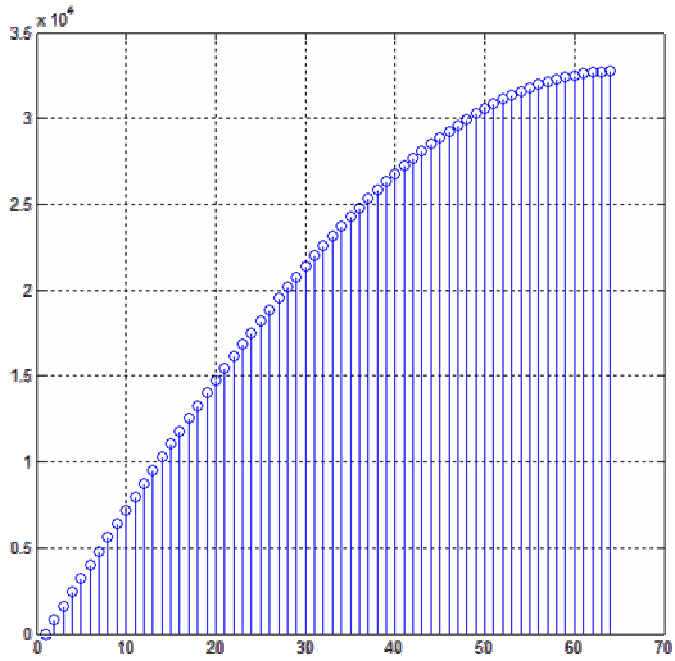
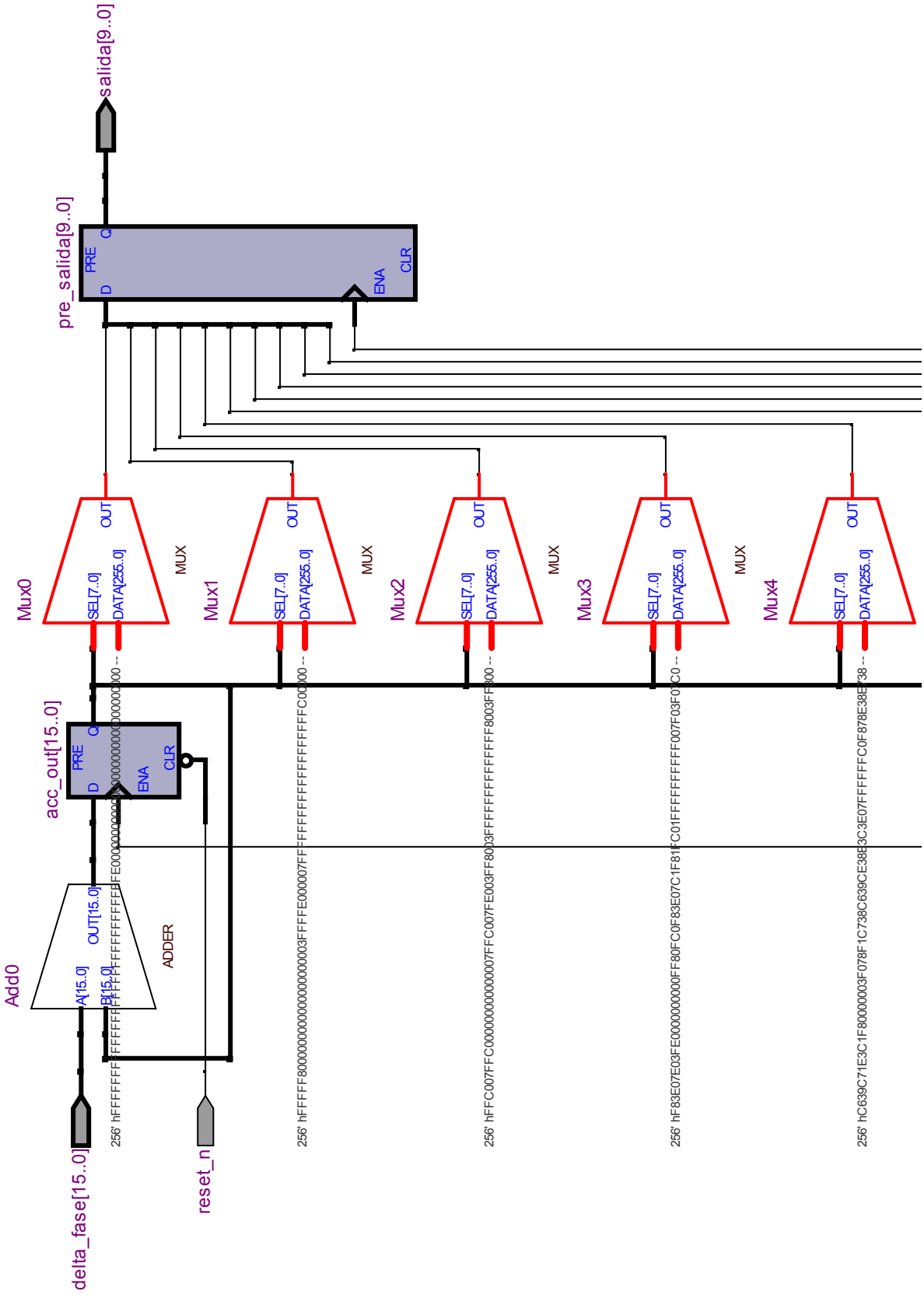


Figura 3.7. Valores para la tabla de arquitectura con técnica de compresión de simetría de cuarto de onda.

La compresión de la ROM provocará que la arquitectura sea modificada levemente, ya que se necesitará lógica combinacional adicional para la generación de la onda completa a partir de sólo un cuarto de la onda. En las figuras 3.8 y 3.9 se muestra la síntesis RTL del Quartus II del DDS para ambas ROMs.

En la figura 3.8 se observan resaltados en color rojo 10 multiplexores que representan la ROM. El sintetizador optó por este tipo de implementación debido a la cantidad de bits que tiene cada valor de la ROM. En el primer caso (figura 3.8) se utilizó 8 bits para representar cada valor de la ROM mientras que en el segundo (figura 3.9) se optó por 16 bits. Hay que mencionar además que un requisito para que el sintetizador pueda inferir una ROM e implementarla en los bloques de memoria M4K es registrar la entrada de dirección de la ROM. Esto se cumple ya que la dirección de la ROM es la salida del registro de acumulación de fase.


salida[9..0]
pre_salida[9..0]
PRE
D
Q
ENA
CLR
Mux0
Mux1
Mux2
Mux3
Mux4
OUT
SEL[7..0]
DATA[255..0]
MUX
acc_out[15..0]
Add0
A[15..0]
B[15..0]
OUT[15..0]
ADDER
delta_fase[15..0]
reset_n

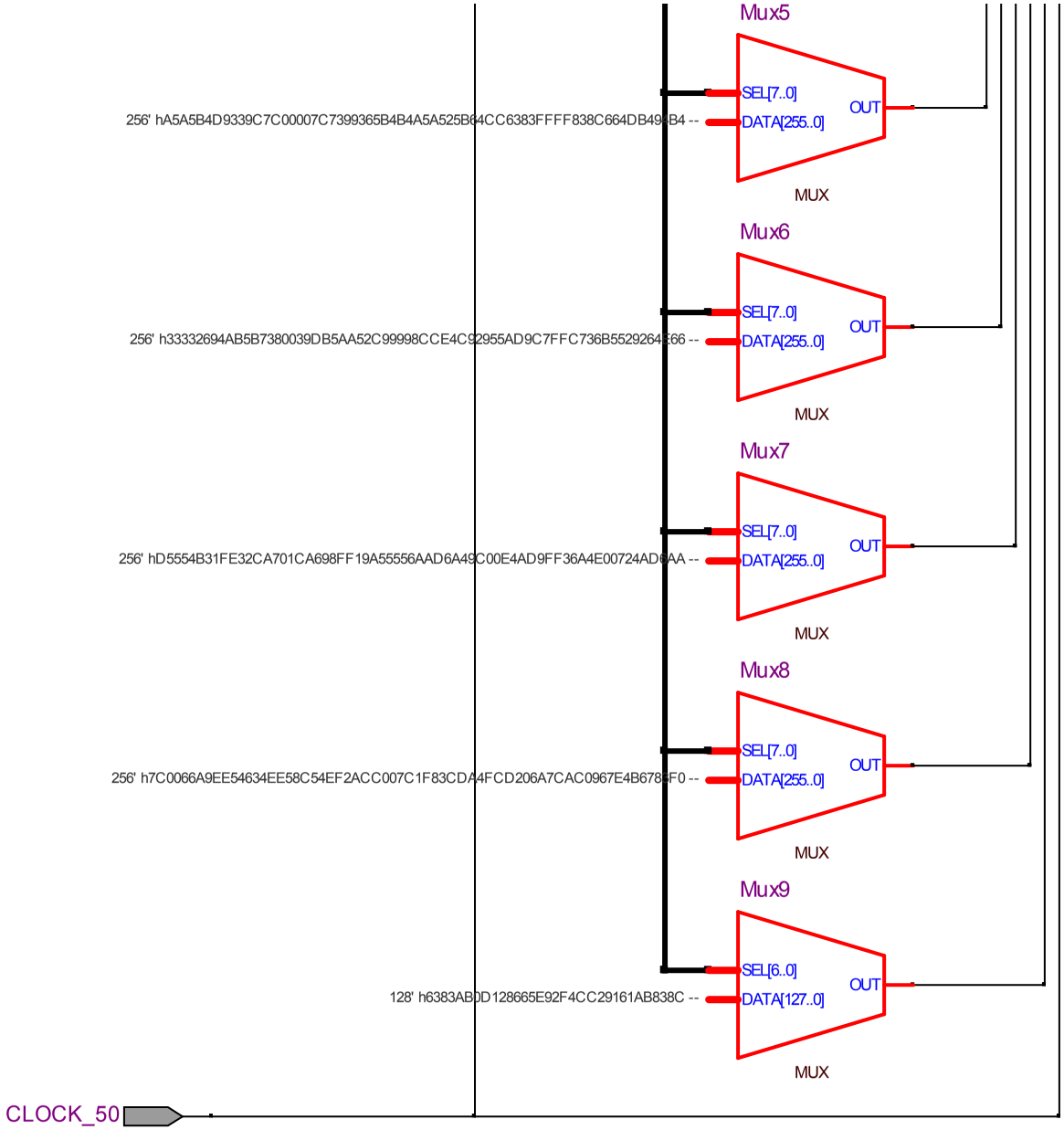


Figura 3.8. Arquitectura DDS para una ROM sin compresión.

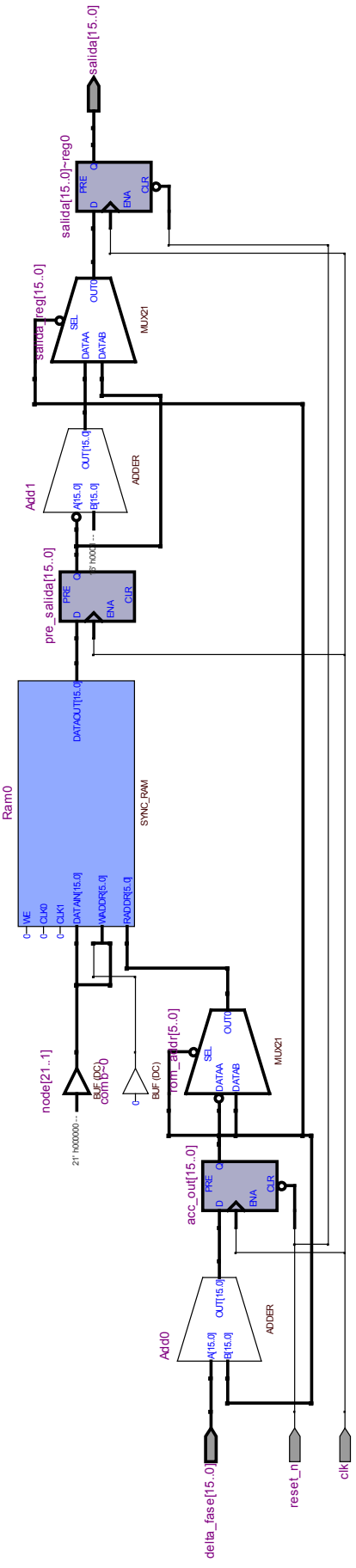


Figura 3.9. Arquitectura DDS para una ROM con compresión de simetría de cuarto de onda.

3.3.2 Procesador Nios II

Este procesador no es un procesador que se encuentre físicamente dentro del FPGA. El procesador Nios II es considerado un "core", es decir que es un procesador realizado en un lenguaje de descripción de hardware y puede ser implementado en la mayoría de FPGAs del fabricante. Es un procesador del tipo RISC - Harvard con un ancho de bus de datos y de direcciones de 32 bits. Muchos de sus parámetros son configurables a través de la herramienta SoPC Builder de Altera [11]. Este core ofrece tres conjuntos de opciones posibles: NiosII/e (económico) que posee el menor consumo en elementos lógicos y alcanza un desempeño de mas de 5 DMIPS (Dhrystone MIPS) @ 50 Mhz [18], NiosII/s (estándar) que posee un consumo medio en elementos lógicos y alcanza un desempeño de mas de 25 DMIPS @ 50 Mhz [18]. Cabe mencionar que además te permite configurar multiplicadores y divisores como co-procesadores y una memoria caché para la memoria de instrucciones. El último modo de configuración es el NiosII/f (rápido) con un alto consumo de elementos lógicos y con un desempeño de más 51 DMIPS @ 50 Mhz [18]. En este caso se utilizó la versión rápida del procesador porque se necesita velocidad en la ejecución de las instrucciones de gestión y control de los dispositivos ya que la velocidad de los datos que provengan del ADC va a ser alta. Se ha deshabilitado las cachés de instrucciones y datos para reducir la complejidad del sistema y facilitar su simulación y verificación.

Tabla 3.1. Características del procesador Nios II.[19]

Característica		Core		
		Nios II/f	Nios II/s	Nios II/e
Objetivo		Rendimiento	Balance rendimiento/ área	Área mínima
Rendimiento[1]	Máx. DMIPS[2]	220	128	31
	f_{MAX}	188	170	201
Bus de datos y direcciones		32 bits	32 bits	32 bits
Segmentación		6 etapas	5 etapas	ninguna
Espacio de direcciones externo		32 Gigabytes	32 Gigabytes	32 Gigabytes
Caché (configurable)	Instrucciones	512 bytes - 64 Kbytes	512 bytes - 64 Kbytes	ninguno
	Datos	512 bytes - 64 Kbytes	Ninguno	ninguno
Área	Cyclone, Cyclone II, Stratix	x < 1800 LEs	x < 1400 LEs	x < 700 LEs
	Stratix II	x < 1050 LEs	x < 950 LEs	x < 500 LEs

Nota:
1. El rendimiento varía dependiendo de la arquitectura del FPGA. Los números mostrados le pertenecen al FPGA Stratix II.
2. DMIPS = Dhrystone MIPS (Utilizando el Benchmark Dhrystone 2.1)

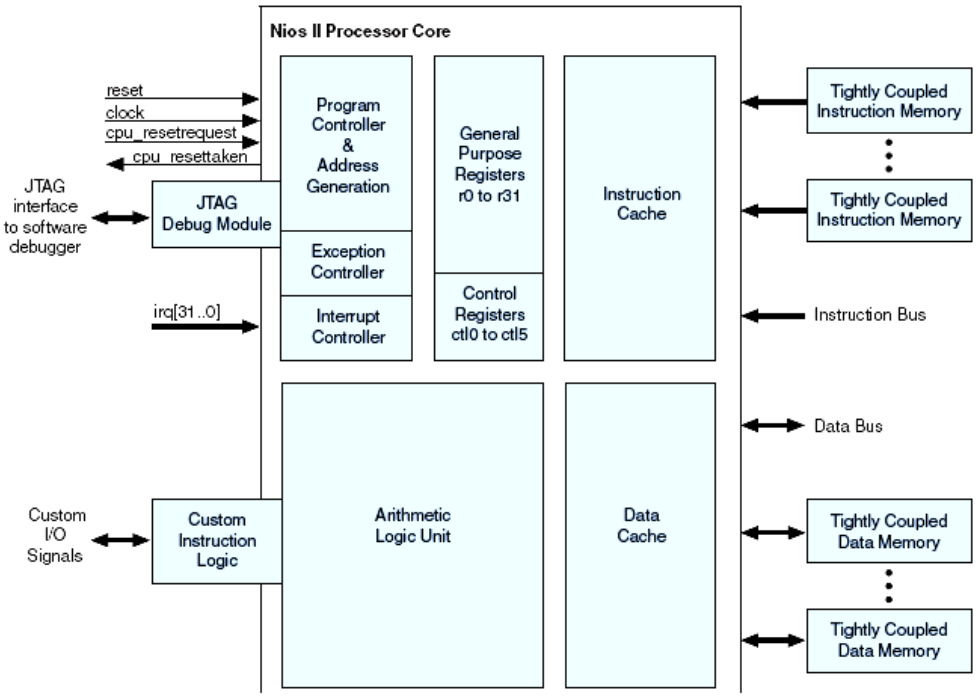


Figura 3.10. Diagrama de bloques del procesador Nios II. [20]

3.3.3 Interfaz Serial Estándar de Comunicación (JTAG UART)

Este periférico nos ayudará a realizar la depuración del código mientras se está en la etapa de pruebas. Realiza una interfaz serial entre el computador y el procesador que nos va a permitir detener la ejecución del procesador y ejecutar línea a línea el código. Además a través de éste podremos tener acceso a los valores de los registros internos del procesador. De esta manera se eliminará también la necesidad de una conexión serial RS-232 para el envío y/o recepción de datos con el computador. Esta conexión se establece a través del protocolo JTAG que en el caso de la tarjeta de desarrollo DE2 está implementada a través de una comunicación USB.

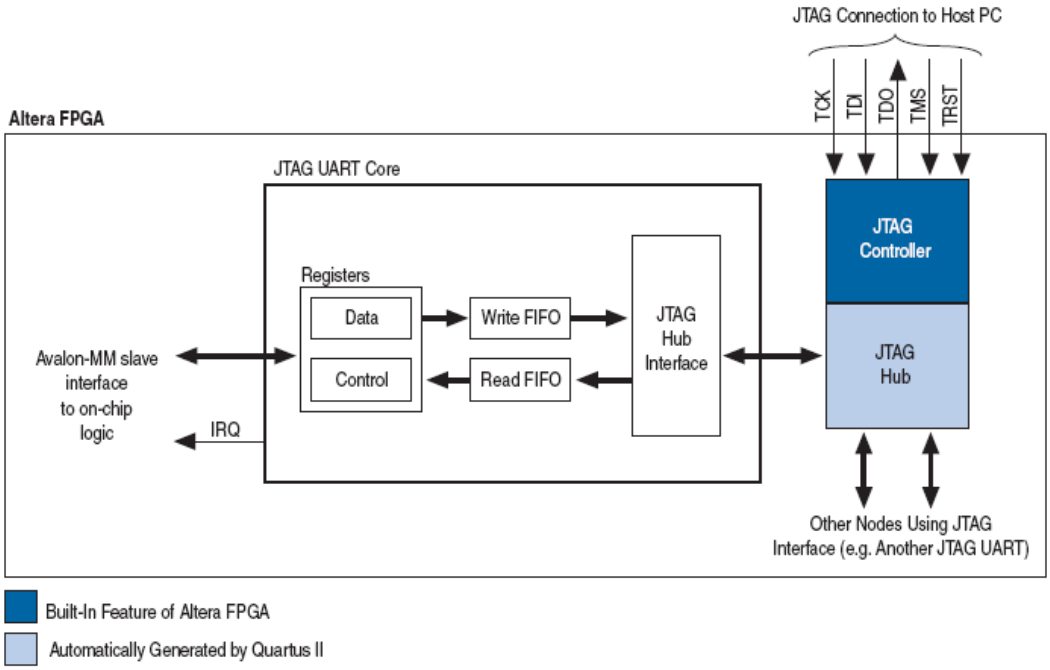


Figura 3.11. Diagrama de bloques del módulo JTAG UART.

3.3.4 Periférico paralelo de entrada y salida

Este periférico nos permitirá conectar el módulo DDS al sistema con el Nios II. Se utilizarán dos de estos módulos, uno de 32 bits configurado como salida que irá conectado a la entrada del módulo DDS y uno como entrada de 16 bits que permitirá capturar los valores del ADC del audio CODEC.

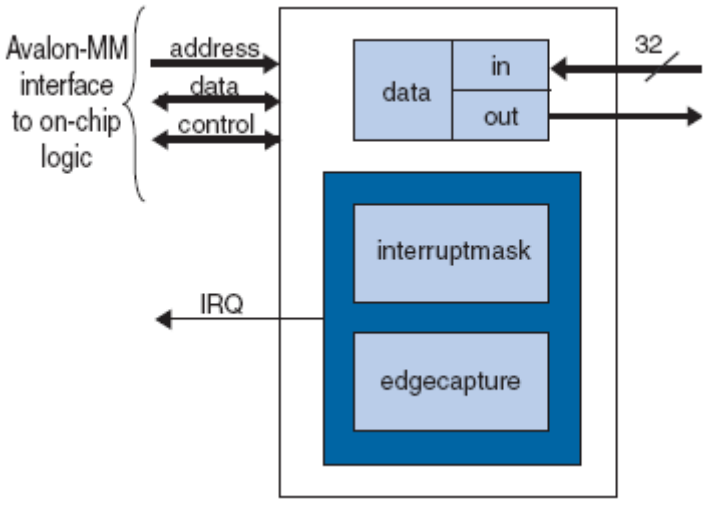


Figura 3.12. Diagrama de bloques del módulo de entrada / salida.

3.3.5 Módulo de Audio

Este módulo nos va a permitir enlazar los puertos del CODEC de audio WM8731 [12] al sistema diseñado. Se compone principalmente de dos sub-módulos, uno encargado de la configuración del dispositivo a través de una comunicación utilizando el protocolo I^2C y un módulo encargado de establecer la comunicación de los datos con el CODEC.

3.3.5.1 Sub-módulo de configuración del CODEC

La función principal de este módulo es la de realizar todos los parámetros de configuración del dispositivo. En la tabla 3.2 se muestra el mapa de registros configurables del CODEC de audio.

Tabla 3.2. Mapa de registros de configuración del CODEC.[12]

	DIRECCIÓN							DATOS								
REG.	B 15	B 14	B 13	B 12	B 11	B 10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
R0	0	0	0	0	0	0	0	LRIN BOTH	LIN MUTE	0	0	LEFT LINE IN VOLUME				
R1	0	0	0	0	0	0	1	RLIN BOTH	RIN MUTE	0	0	RIGHT LINE IN VOLUME				
R2	0	0	0	0	0	1	0	LRHP BOTH	LZCEN	LEFT HEADPHONE VOLUME						
R3	0	0	0	0	0	1	1	RLHP BOTH	RZCEN	RIGHT HEADPHONE VOLUME						
R4	0	0	0	0	1	0	0	0	SIDEATT		SIDE TONE	DAC SEL	BY PASS	INSEL	MUTE MIC	MIC BOOST
R5	0	0	0	0	1	0	1	0	0	0	0	HPOR	DACMU	DEEMPH		ADC HPD
R6	0	0	0	0	1	1	0	0	PWR OFF	CLK OUTPD	OSC PD	OUT PD	DACPD	ADC PD	MIC PD	LINEIN PD
R7	0	0	0	0	1	1	1	0	BCLK INV	MS	LR SWAP	LRP	IWL		FORMAT	
R8	0	0	0	1	0	0	0	0	CLKO DIV2	CLKI DIV2	SR				BOSR	USB/ NORM
R9	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	ACTIVE
R15	0	0	0	1	1	1	1	RESET								

Estos registros son de 16 bits, donde los 7 bits más significativos representan la dirección del registro. El CODEC soporta dos modos de comunicación para la interfase de configuración, el modo compatible con SPI y el modo compatible con I^2C. El modo se selecciona utilizando el pin 25 del circuito integrado (figura 3.13). En este modo el dispositivo se comporta como esclavo. El pin 26 permite escoger entre dos direcciones para el dispositivo.

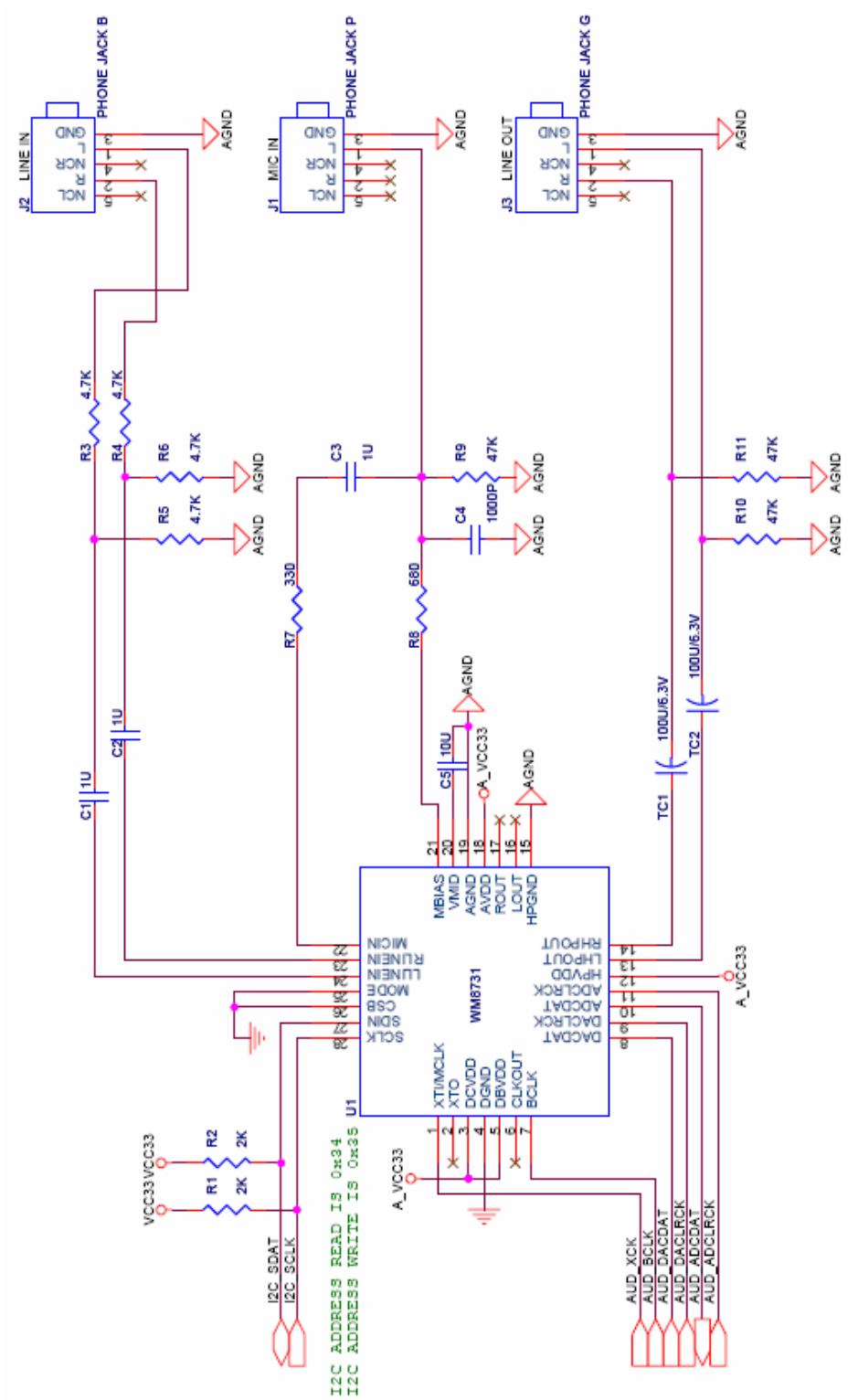


Figura 3.13. Diagrama esquemático de conexiones del CODEC de audio al FPGA en la tarjeta de desarrollo. [21]

El circuito se diseñó utilizando como referencia los circuitos desarrollados por el fabricante de la tarjeta de desarrollo y se basa principalmente en una máquina de estados y un circuito generador de la trama de bits serial I^2C tanto para la señal de sincronía SCL como también la señal de datos serial SDA.

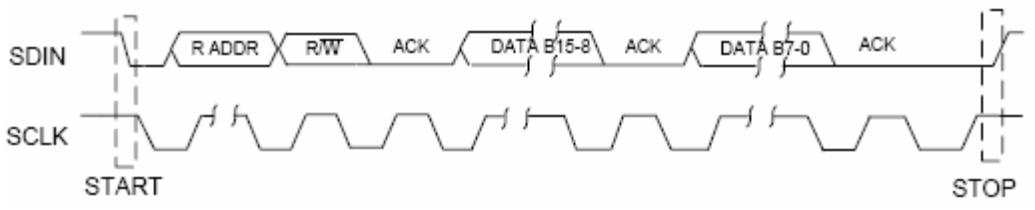


Figura 3.14. Diagrama de tiempos de una comunicación I^2C.[12]

3.3.5.2 Sub-módulo de comunicación con el CODEC

Este módulo es el que permite enviar los datos hacia el DAC y recibir la señal de audio digitalizada del ADC. El CODEC posee 4 modos de operación: justificado a la izquierda, justificado a la derecha, I^2C y DSP. El circuito diseñado opera utilizando el primer modo ya que así se ha configurado en el registro de configuración respectivo. El diseño de este circuito esta basado en los circuitos desarrollados por el fabricante para el control del CODEC de audio.

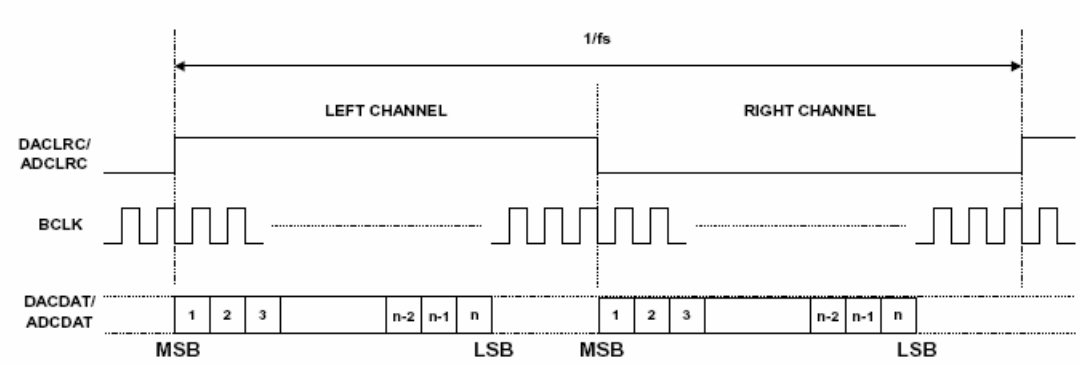


Figura 3.15. Modo de operación justificado a la izquierda. [12]

3.3.6 Módulo controlador de señales de reloj del sistema

Este módulo se encargará de generar las señales de reloj correspondientes para los circuitos integrados de audio y video. Para el CODEC de audio, la señal de reloj esta relacionada con la frecuencia de muestreo. Para obtener una frecuencia de muestreo de 48 Khz se utilizará una señal de reloj de 12.288 Mhz (según

especificación de la hoja de datos del fabricante [12]) obtenida del reloj del sistema. El chip de video necesita una señal de reloj de 25 Mhz. Para lograr estos requerimientos se utilizó uno de los cuatro bloques de PLL que posee el FPGA Cyclone II. Cada uno de estos bloques tiene la posibilidad de generar tres señales de reloj a partir de una sola referencia. La referencia de este bloque será la señal del cristal de 27 Mhz que se encuentra en la tarjeta de desarrollo. Se utilizó la Megafunción de Altera ALTPLL a través de la herramienta MegaWizard Plug-in manager.

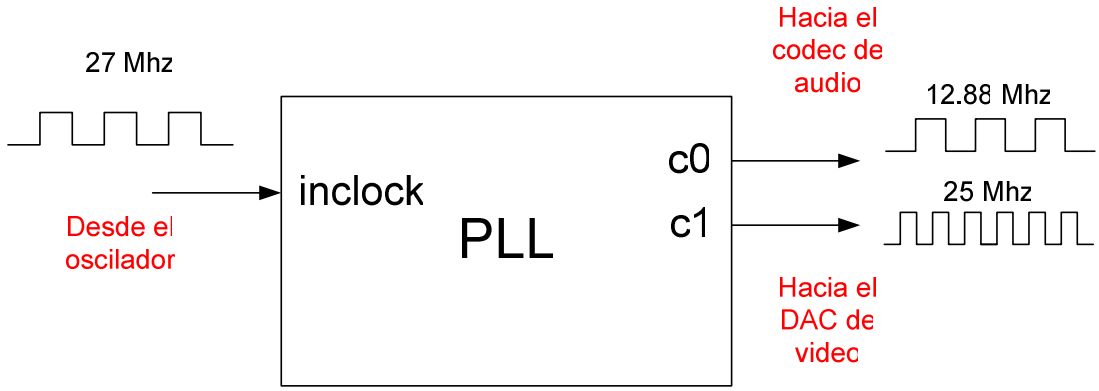


Figura 3.16. PLL implementado.

3.3.7 Memoria RAM embebida

Este módulo nos permitirá implementar una memoria RAM embebida en el FPGA utilizando los bloques M4K. En esta memoria se colocarán las instrucciones y los datos que ejecutará el procesador. El módulo permite implementar la memoria utilizando elementos lógicos o usando los bloques M4K pero se han utilizado éstos últimos debido a la velocidad de lectura y escritura que se requiere para el sistema. Con los bloques M4K se pueden obtener velocidades de hasta 250 Mhz. Este módulo se instancia desde la herramienta SoPC Builder. La comunicación del procesador con esta memoria se realizará a través de dos buses independientes para el intercambio de datos e instrucciones.

3.3.8 Módulo de Video

Este módulo nos permite configurar los parámetros del chip de video. En este caso se desactivarán todos los pines correspondientes al control y sincronía de los

píxeles hacia la pantalla ya que sólo utilizaremos los DACs del mencionado elemento.

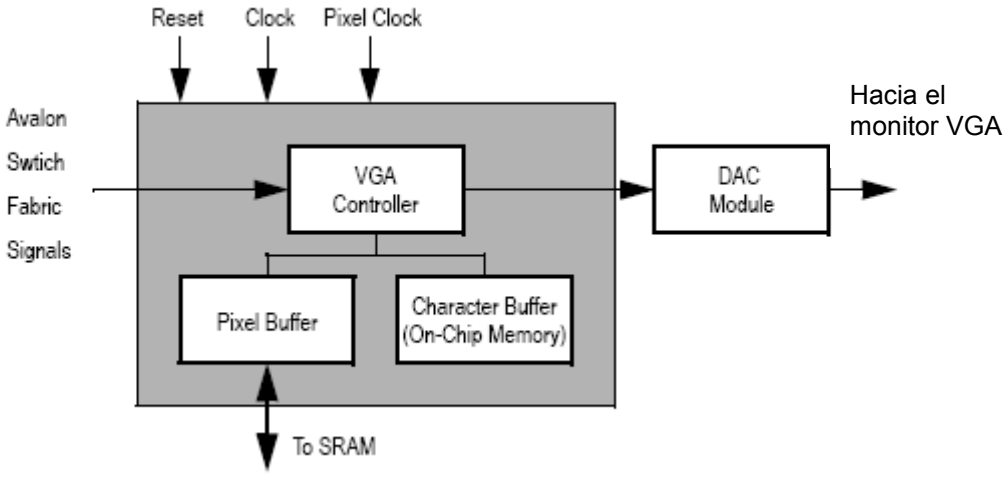


Figura 3.17. Diagrama de bloques del módulo de video.

3.4 Desarrollo del software del sistema

En esta etapa del diseño se utilizó la herramienta Nios II IDE de Altera. Esta herramienta permite gestionar la plataforma necesaria para desarrollar el software del sistema. Utiliza un conjunto de drivers que controlan al hardware.

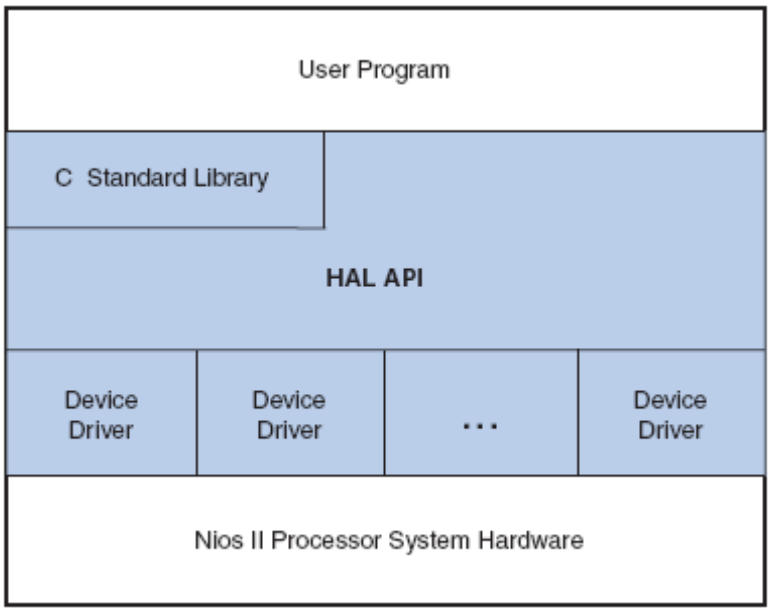


Figura 3.18. Diagrama de bloques de la interfaz hardware-software. [22]

3.4.1 Capa de abstracción de hardware (HAL)

Ésta es la que permite diseñar el software del sistema sin tener una dependencia fuerte sobre el hardware. Permite el uso de funciones de alto nivel (Lenguaje C) para interactuar con el hardware. Estas funciones se basan en el control de los registros que la mayoría de módulos posee.

3.4.2 Descripción funcional del código principal

La función principal del programa que se ejecute en el procesador, será únicamente de gestión y control de flujo de datos. Se utilizará el procesador para la inicialización de los dispositivos externos al FPGA como el CODEC de audio y el chip de video. A continuación se muestra el diagrama de flujo general del lazo principal. Debido a que este programa se ejecuta indefinidamente, se utilizarán algunas opciones que provee el compilador para reducir el tamaño del código ejecutable.

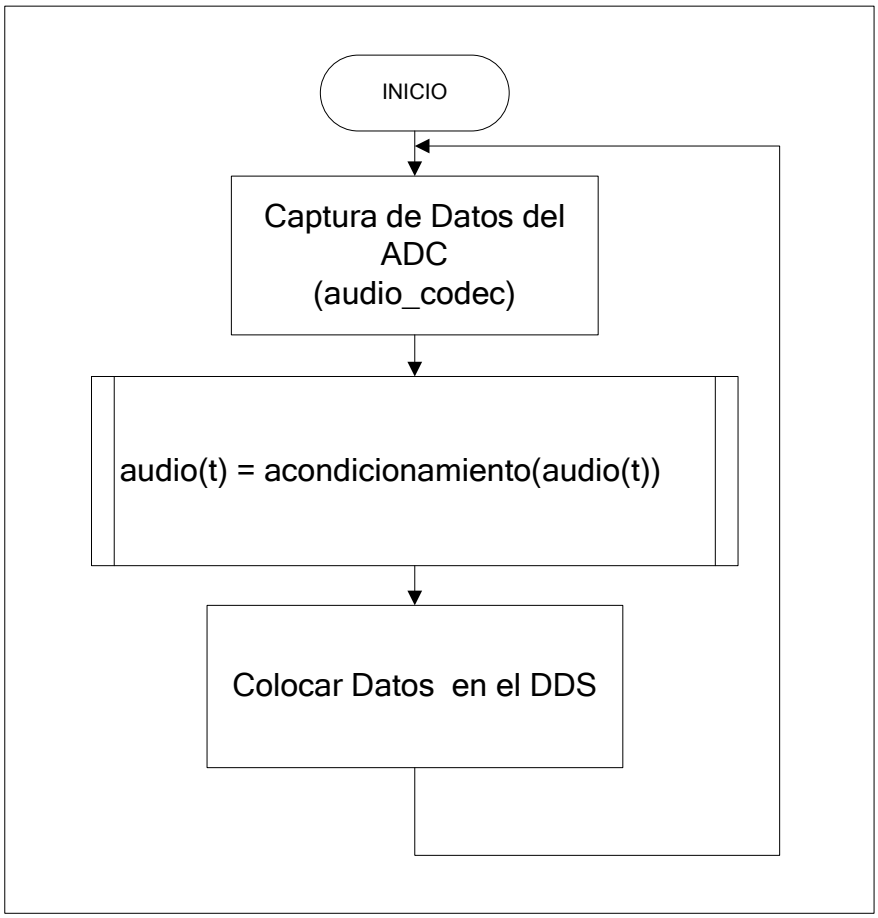


Figura 3.19. Diagrama de flujo del software.

CAPÍTULO 4: RESULTADOS DEL DISEÑO PROPUESTO

Luego de haber estructurado el diseño final del sistema y siguiendo el flujo de diseño en FPGAs, consistente en realizar la descripción de los módulos a utilizar de manera independiente, se procedió con la simulación de los bloques funcionales independientes utilizando el simulador ModelSim y el simulador propio del Quartus II. La interfaz utilizada para hacer la configuración del FPGA fue el USB utilizando el protocolo de comunicación JTAG.

4.1. Resultados de simulación de los módulos independientes

Se realizaron las simulaciones de los módulos diseñados utilizando el simulador propio del Quartus II. Los módulos pertenecientes a la herramienta SoPC Builder no han sido simulados debido a que ya cuentan con una validación de su funcionamiento de parte del fabricante.

4.1.1. Sintetizador digital Directo (DDS)

Se simularon los dos diseños propuestos para este módulo. El primero utilizando una ROM sin compresión y el segundo utilizando una compresión de simetría de cuarto de onda. La compresión utilizando simetría de cuarto de onda consiste en aprovechar la simetría que posee una onda senoidal. Por lo tanto sólo es necesario almacenar en la ROM un cuarto de período de la onda y así obtener una compresión de 4 a 1. La compresión es importante ya que casi el 90 % del área que ocupa un sintetizador digital directo en una implementación digital se debe al tamaño de la ROM.

La descripción de los pines de entrada y salida son los siguientes:

Tabla 4.1. Descripción de las entradas y salidas del módulo DDS. [elaboración propia]

Módulo	**Función**	
DDS	Generación de señal modulada	
Entrada	**Función**	**Cantidad de bits**
reset_n	Señal de reseteo asíncrono del bloque activado en baja.	1

clock_50	Señal sincronía de reloj	1
delta_fase	Indica la variación de la fase.	32
Salida	**Función**	**Cantidad de bits**
salida	Datos con la señal modulada	10

Tabla 4.2. Descripción de los parámetros del módulo DDS. [elaboración propia]

Parámetro	Descripción
ancho_palabra_in	Cantidad de bits de la entrada *delta_fase.*
ancho_palabra_out	Cantidad de bits de la salida *salida.*
bits_nousados	Cantidad de bits que serán descartados por el acumulador de fase.

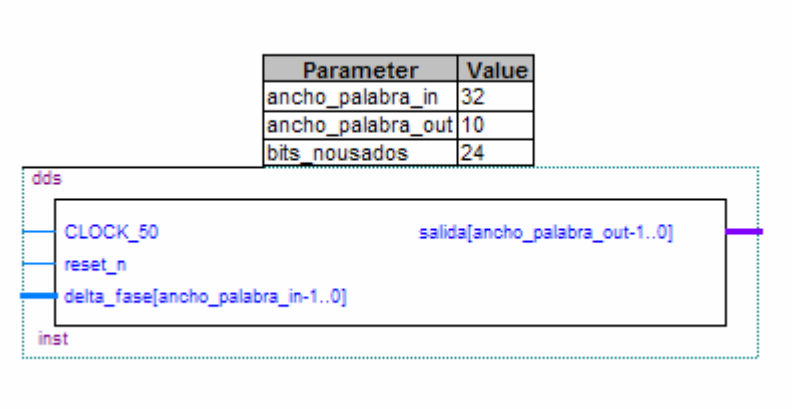


Figura 4.1. Sintetizador digital directo.

Los resultados de consumo de área se muestran en la siguiente tabla:

Tabla 4.3. Resultado de la síntesis del DDS. [elaboración propia]

	Resumen de utilización del FPGA EP2C35F672C6				
Sin compresión	**Utilización de lógica**	**Usada**	**Disponible**	**Utilización (%)**	**Frec. Máx. (Mhz)**
	Total de elementos lógicos	32	33216	<1%	264.48
	Total de pines	44	475	9%	
	Total de bits de memoria (M4K)	2560	483840	<1%	
	Resumen de utilización del FPGA EP2C35F672C6				
Con compresión	**Utilización de lógica**	**Usada**	**Disponible**	**Utilización (%)**	**Frec. Máx. (Mhz)**
	Total de elementos lógicos	49	33216	<1%	170.88
	Total de pines	44	475	9%	
	Total de bits de memoria (M4K)	640	483840	<1%	

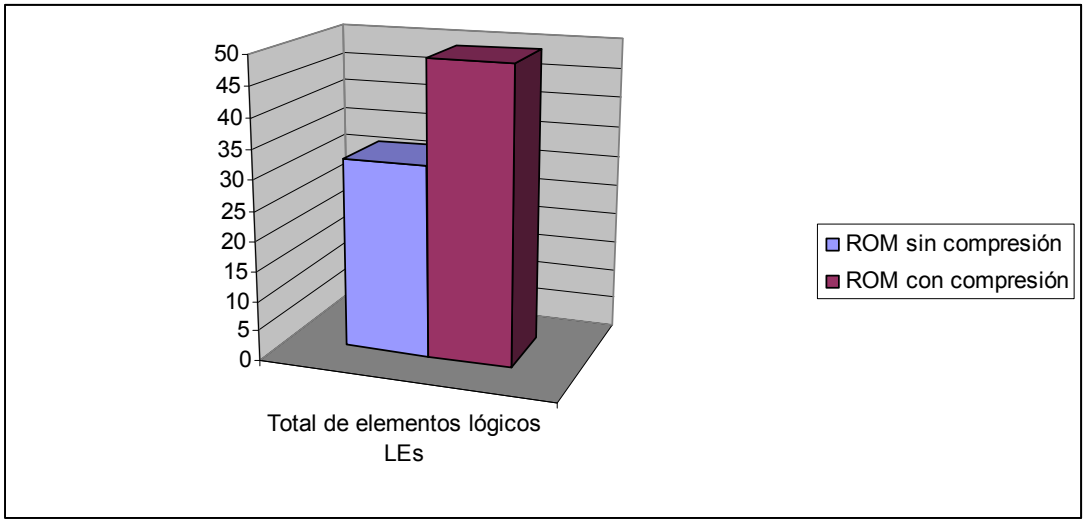


Figura 4.2. Comparación del consumo de elementos lógicos

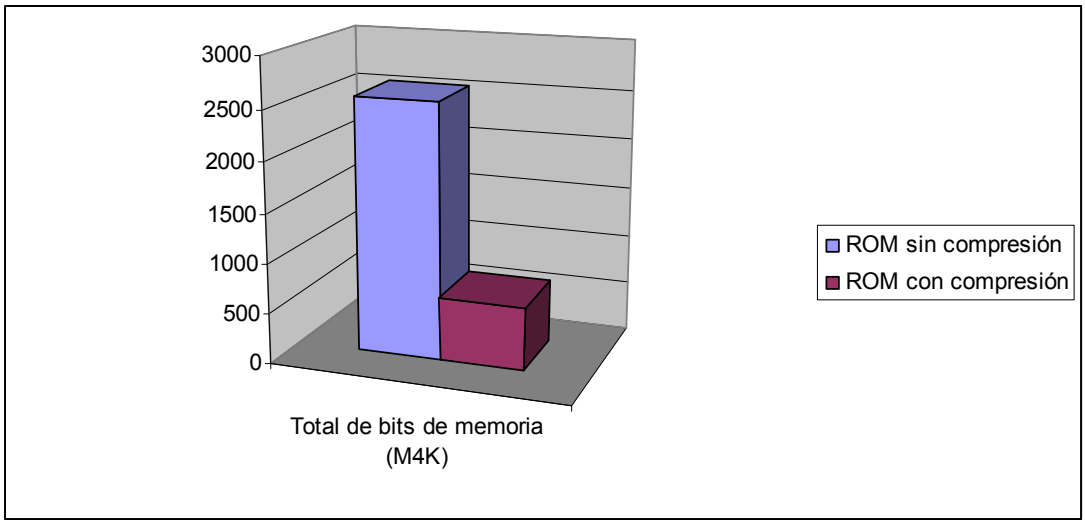


Figura 4.3. Comparación del consumo de bits de memoria (M4K).

Los resultados muestran que existe un gran ahorro de área al utilizar la compresión en la ROM. Esta técnica logró comprimir la ROM en un 75% mientras que la adición de lógica combinacional no es tan significativa.

Los resultados de la simulación se muestran a continuación:

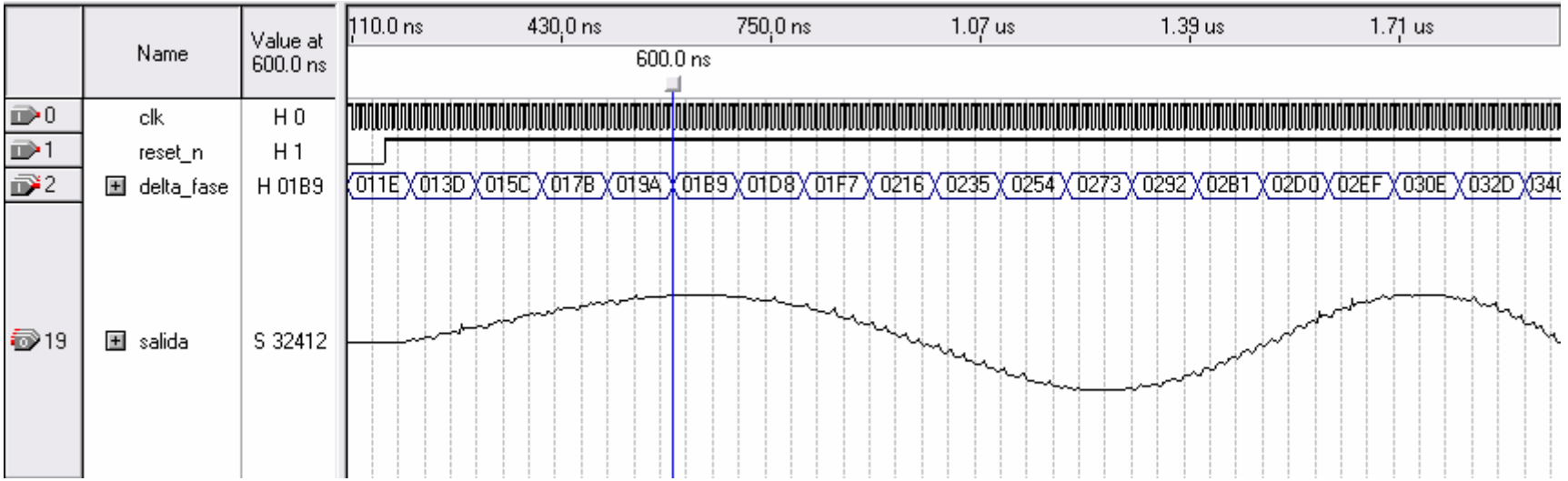


Figura 4.5. Resultados de la simulación del DDS sin compresión.

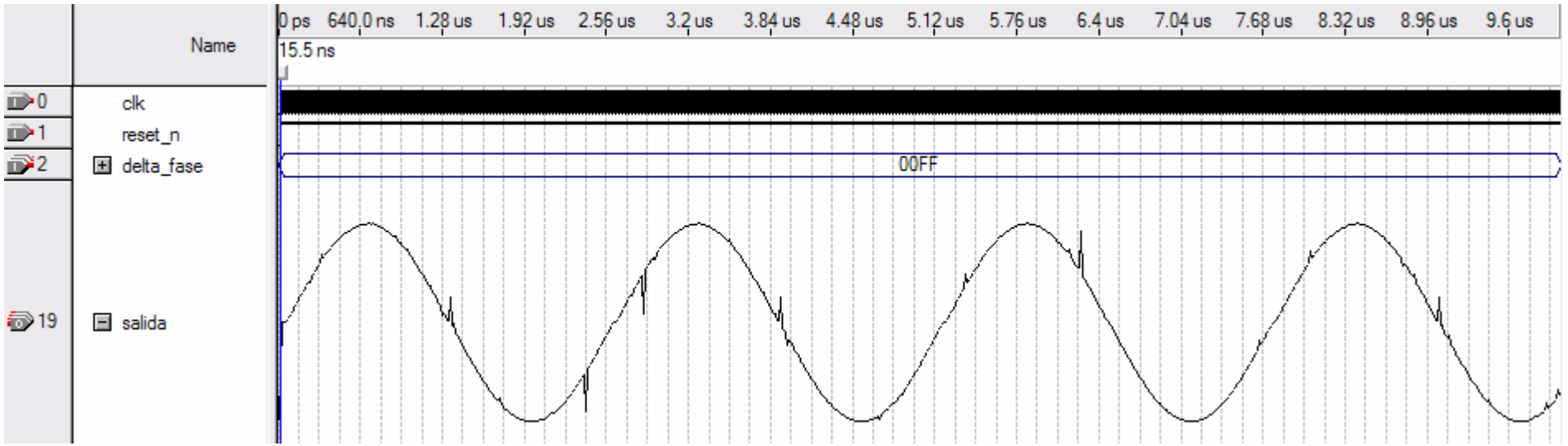


Figura 4.6. Resultados de la simulación del DDS con compresión.

En las figuras 4.5 y 4.6 observamos la generación de una onda senoidal para ambos diseños. Para la simulación del DDS sin compresión de ROM se ha colocado un incremento de fase (delta_fase) que varía constantemente en el tiempo, por lo tanto obtenemos una onda senoidal que va aumentando su frecuencia con respecto al incremento de fase. En realidad lo que debemos observar en la variable (salida) debe ser los valores que retorna la ROM pero el simulador de Quartus II posee la característica de interpretar estos valores y representarlos como una señal analógica. Esto es muy útil porque nos permite observar la correcta síntesis de la onda senoidal. En la simulación del DDS con compresión de ROM se ha utilizado un incremento de fase constante por lo tanto se genera una onda de frecuencia constante.

4.1.2. Módulo de Audio

Este módulo posee dos sub-bloques: el primero que implementa el protocolo I^2C y el segundo que a través de una máquina de estados, envía los valores que se deben grabar en los registros del CODEC de audio a través del sub-módulo I^2C. La velocidad de transmisión I^2C será de 20 Kbps. La función principal de este módulo es la captura del audio que proviene de la línea de entrada del CODEC y almacenarla temporalmente para que luego sea modulada y transmitida.

Tabla 4.4. Descripción de las entradas y salidas del módulo de audio. [elaboración propia]

Módulo	**Función**	
i2c_ctrl	Interfaz con el CODEC de audio.	
Entrada	**Función**	**Cantidad de bits**
iRST_N	Señal de reseteo asíncrono del bloque activado en baja.	1
iCLK	Señal sincronía de reloj.	1
Salida	**Función**	**Cantidad de bits**
I2C_SCLK	Señal de reloj de la comunicación I^2C.	1
I2C_SDAT	Señal de datos de la comunicación I^2C.	1

Tabla 4.5. Descripción de los parámetros del módulo de audio. [elaboración propia]

Parámetro	Descripción
CLK_Freq	Frecuencia a la que trabajará el módulo de audio expresada en hertz.
I2C_Freq	Frecuencia a la que trabajará la comunicación I^2C del módulo expresada en hertz.
LUT_SIZE	Cantidad de posiciones que poseerá la memoria del módulo para almacenar la configuración que se le dará al CODEC.
Dummy_DATA	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
SET_LIN_L	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
SET_LIN_R	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
SET_HEAD_L	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
SET_HEAD_R	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
A_PATH_CTRL	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
D_PATH_CTRL	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento y del código.
POWER_ON	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
SET_FORMAT	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
SAMPLE_CTRL	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
SET_ACTIVE	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.
SET_VIDEO	Representación literal de la posición en la memoria que ayudará a un mejor entendimiento del código.

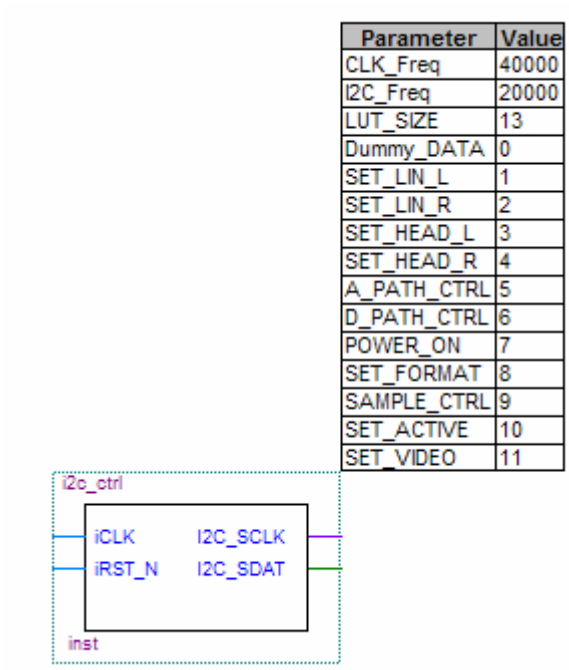


Figura 4.7. Módulo de audio.

Tabla 4.6. Resultado de la síntesis del módulo de audio. [elaboración propia]

Resumen de utilización del FPGA EP2C35F672C6				
Utilización de lógica	**Usada**	**Disponible**	**Utilización (%)**	**Frec. Máx. (Mhz)**
Total de elementos lógicos	95	33216	<1%	264.48
Total de pines	4	475	<1%	
Total de bits de memoria (M4K)	0	483840	0%	

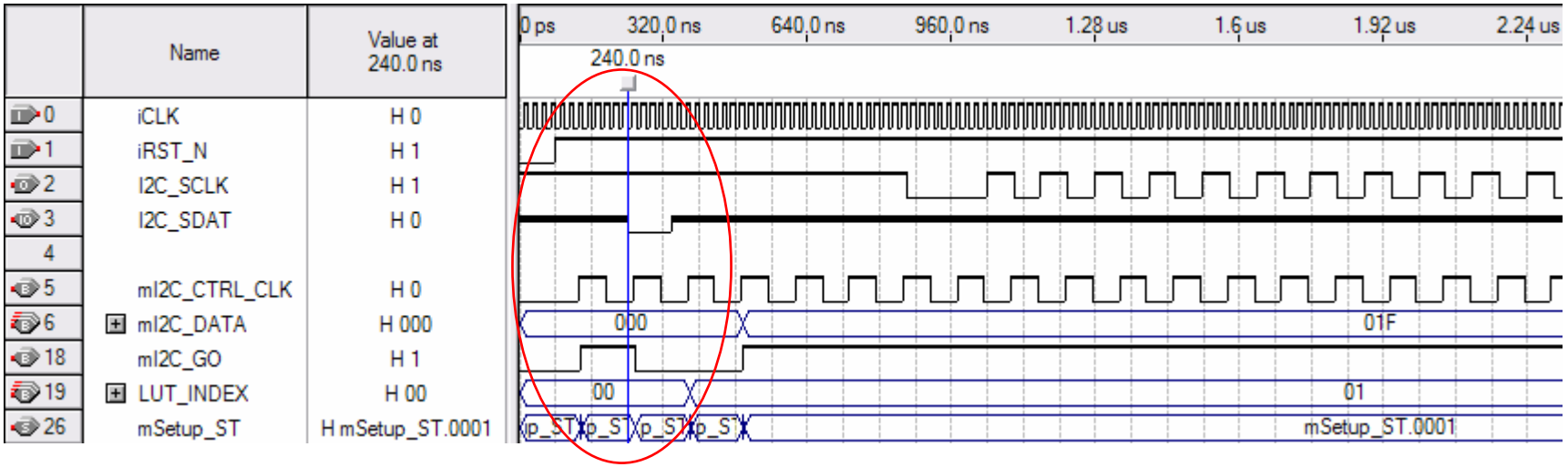


Figura 4.8. Resultados de simulación del módulo de configuración de audio.

En la gráfica anterior se muestra la etapa de configuración de registros del CODEC. Se muestra el envío de las dos primeras palabras de 16 bits hacia el CODEC de audio a través del bus I^2C. La máquina de estados controla el inicio del envío de los bits a través del bus con la señal mI2C_GO. Cada vez que se active esta señal se enviará una trama de 16 bits de datos más 8 bits de la dirección del dispositivo y el tipo de operación a ejecutar (lectura o escritura), esto parte del protocolo I^2C. Los bits que se deben enviar se encuentran almacenados en el registro mI2C_DATA. El registro LUT_INDEX permite seleccionar que trama de bits se va a enviar. Este valor es controlado por la máquina de estados. Según recomendación del fabricante del CODEC la primera vez que se envíe un dato, no es seguro que se ejecute correctamente la operación. Es por esto que el primer dato que se envía es un trama de 16 bits “0”.

4.2. Resultados utilizando la herramienta SignalTap II

El siguiente paso es la comprobación de los módulos independientes ya implementados en la tarjeta de desarrollo. Dentro de este proceso de verificación se utilizó la herramienta SignalTap II para corroborar los resultados de simulación. SignalTap II es un analizador lógico desarrollado por Altera para poder observar los cambios en la señales internas y externas del FPGA. El analizador se añade al proyecto creado en Quartus II, se sintetiza y se implementa en el FPGA. Se le programa la condición para que inicie la captura de las señales seleccionadas y a través del protocolo JTAG se comunica con el computador.

4.2.1. Sintetizador digital Directo (DDS)

El la siguiente figura se muestran los resultados obtenidos de la implementación del módulo DDS. En esta herramienta no se puede hacer la transformación de la salida a una forma de onda analógica pero si podemos obtener otros resultados. Está el hecho de que existe un retardo inicial en la generación de la onda desde el momento en que se resetea el sistema. Esto se debe a los registros internos que posee el módulo que se encuentran en un arreglo del tipo pipeline o segmentado por etapas, que permite obtener un valor nuevo en la salida cada ciclo de reloj. Este hecho no interferirá en el correcto funcionamiento del módulo. Los cambios a la salida siguen siendo constantes y periódicos con respecto a la señal de reloj.

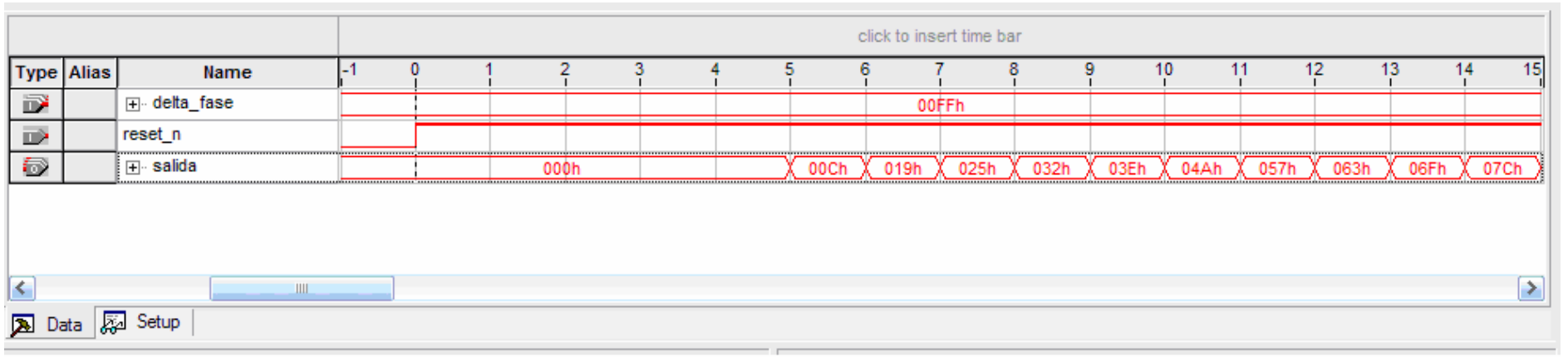


Figura 4.9. Resultados obtenidos del módulo DDS.

4.2.2. Módulo de Audio

En la figura 4.10 se muestran los resultados de la implementación del módulo de audio. Ya no se muestra la etapa de configuración de los registros sino más bien la etapa de recepción de los datos provenientes del ADC del CODEC hacia el FPGA. El módulo se configuró para que recibiera los datos en el modo justificado a la izquierda y con el MSB primero en transmitirse. Dado que el CODEC está configurado como esclavo, las señales AUD_ADCLRCK y AUD_DACLRCK son salidas del FPGA y por lo tanto entradas del CODEC. La señal AUD_ADCDAT es la que recibe los datos del ADC de manera serial sincronizada con la señal AUD_BCK y la señal AUD_ADCLRCK es la que le indica al CODEC cual de los dos canales se va a recibir ya sea el izquierdo o el derecho.

Las otras dos señales (AUD_DACDAT y AUD_DACLRCK) se utilizan cuando se desea enviar alguna señal hacia el DAC del CODEC y poder escucharla en la línea de salida. Estas señales no se utilizarán en la implementación completa del sistema.

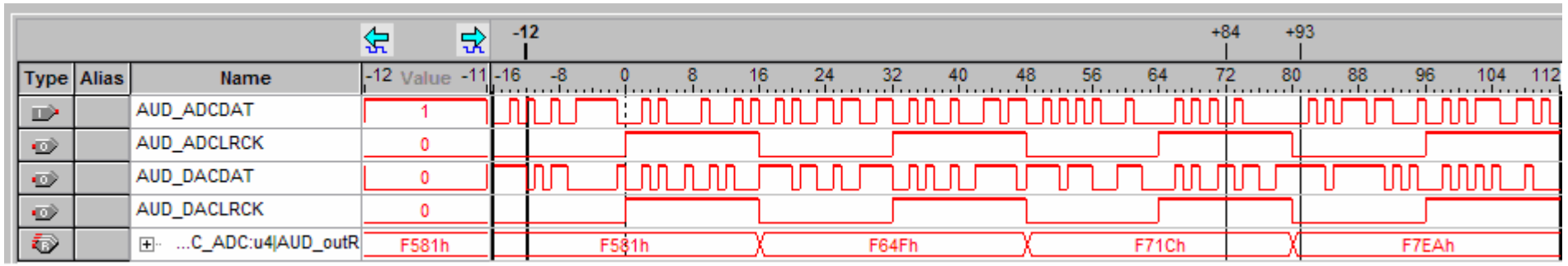


Figura 4.10. Resultados obtenidos del módulo de audio.

4.3. Resultados de la implementación en la tarjeta de desarrollo

El sistema fue implementado en la tarjeta de desarrollo Altera DE2. Existe un parámetro de configuración del sintetizador que nos permite escoger entre una síntesis orientada a la velocidad, al consumo de área o balanceada. Esta síntesis fue realizada utilizando los siguientes parámetros:

- 16 bits de datos del ADC del CODEC de audio.
- 32 bits de la longitud de palabra del acumulador de fase.
- 10 bits de salida del módulo DDS.

Los resultados de consumo de área del FPGA y frecuencia máxima de operación del sistema se muestran en la tabla siguiente:

Tabla 4.6. Resultados luego del análisis y síntesis. [elaboración propia]

Consumo del FPGA EP2C35F672C6						Frec. Máx. (Mhz)
PARAMETRO	"speed"	**Consumo lógico**	**Utilizado**	**Disponible**	**% de utilización**	68.99
		Total de LEs	3543	33216	11%	
		Total de pines	429	475	90%	
		Total Memory bits (M4K)	227840	483840	47%	
		PLLs	1	4	25%	
	"área"	**Consumo lógico**	**Utilizado**	**Disponible**	**% de utilización**	60.25
		Total de LEs	3324	33216	10%	
		Total de pines	429	475	90%	
		Total Memory bits (M4K)	227840	483840	47%	
		PLLs	1	4	25%	
	"balanced"	**Consumo lógico**	**Utilizado**	**Disponible**	**% de utilización**	68.25
		Total de LEs	3328	33216	10%	
		Total de pines	429	475	90%	
		Total Memory bits (M4K)	227840	483840	47%	
		PLLs	1	4	25%	

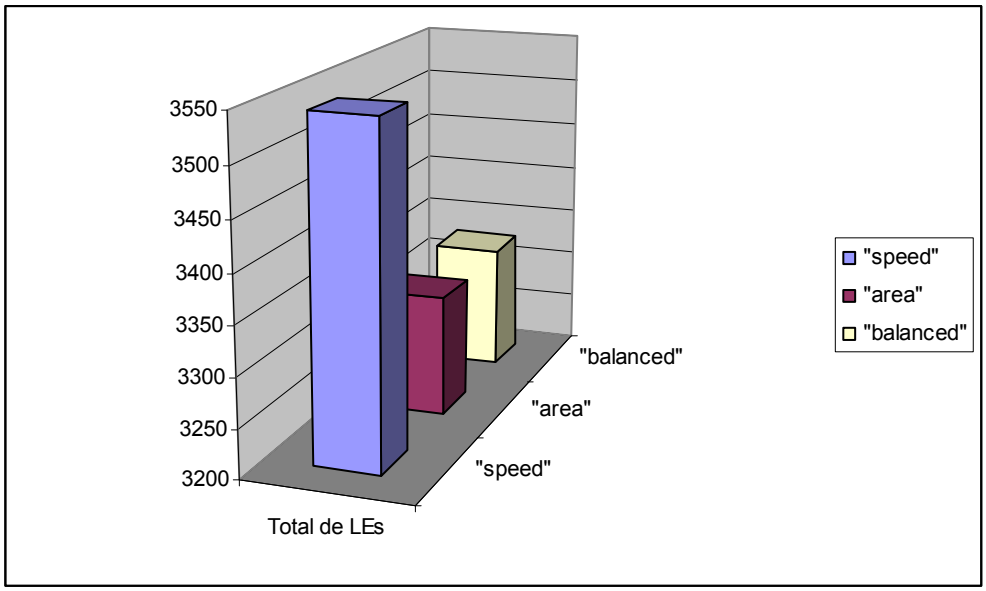


Figura 4.11. Comparación de los resultados de total de LEs para las distintas síntesis.

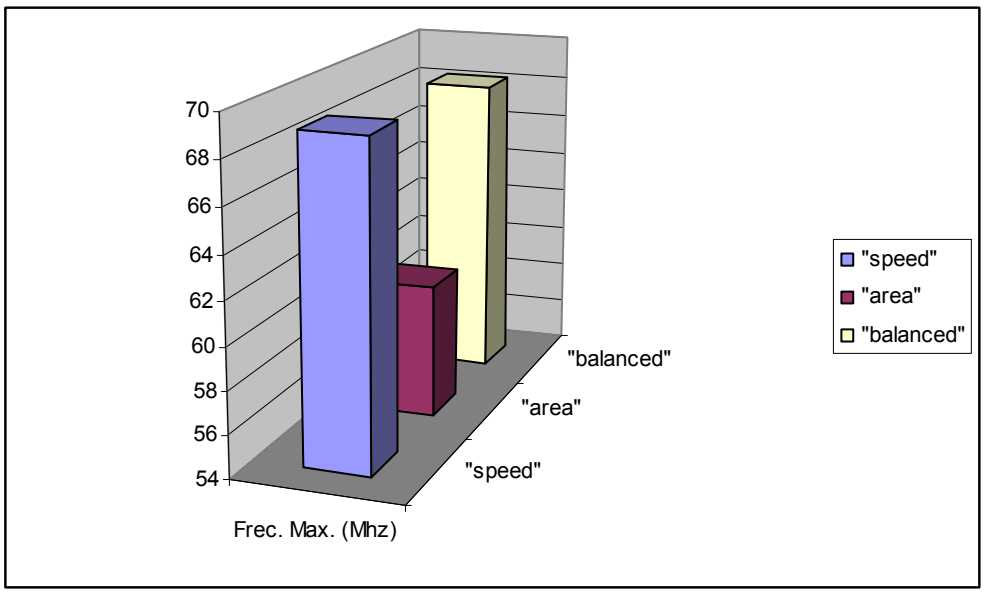


Figura 4.12. Comparación de los resultados de frecuencia máxima de operación para las distintas síntesis.

Los resultados muestran que éste parámetro del compilador altera considerablemente el área y la frecuencia máxima de operación. El “balanced” ofrece un buen compromiso entre consumo de área y frecuencia máxima de operación. El funcionamiento de éste parámetro dentro del sintetizador involucra los algoritmos que posee para la síntesis lógica y su colocación dentro del FPGA.

La figura 4.13 muestra la señal obtenida a la salida de DAC del chip de video. Se obtiene una señal FM modulada con una portadora cuadrada a una frecuencia de 4.5 Mhz. Se puede observar además el desplazamiento en frecuencia que ocurre debido a la señal de audio que esta ingresando a través del canal estéreo del CODEC de audio. Al realizar las pruebas de la implementación se utilizó un receptor de radio FM comercial y se colocó en una frecuencia cercana a los 88 Mhz. Debido al espectro en frecuencia que posee la onda cuadrada, se pudo escuchar la señal modulada proveniente del FPGA en el receptor.

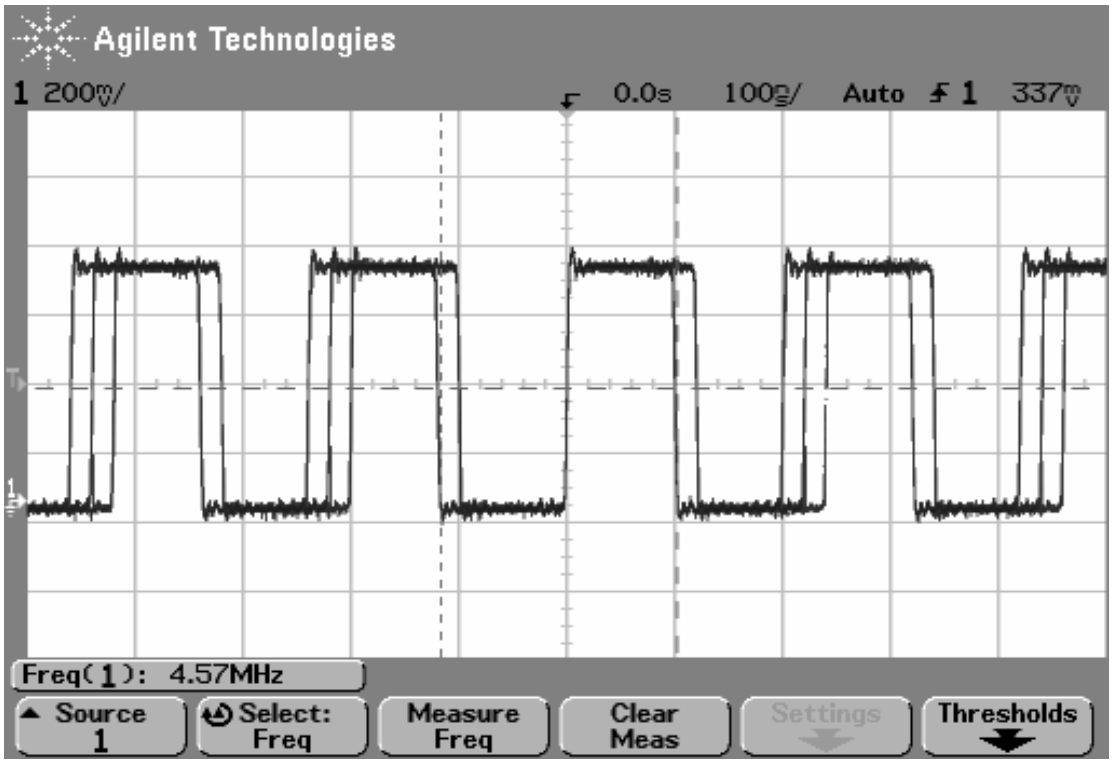


Figura 4.13. Señal FM obtenida de la implementación del sistema.

CONCLUSIONES

- La arquitectura propuesta muestra un reducido consumo de área (14% de LEs totales del FPGA Cyclone II EP2C35F672C6) debido a un diseño más estructural que comportamental. Esto permite un mejor control sobre la síntesis del circuito. Hay que mencionar que el espacio ocupado por los diseños de propiedad intelectual de Altera consumen el 10% de LEs totales del FPGA y sólo el 4% son utilizados por los módulos propios.

- Se obtuvo una reducción significativa de consumo de LEs por parte del DDS ya que se utilizó una técnica de compresión de ROM de simetría de cuarto de onda. El consumo de LEs del módulo DDS se debe en gran medida a la ROM que posee para generar la onda senoidal. Además se sabe que el tamaño de la ROM aumenta exponencialmente a medida que se aumenta la cantidad de bits de la dirección de la ROM.

- Las limitaciones de ancho de banda del conversor digital-analógico DAC ADV7123 no permiten utilizar una señal senoidal como señal portadora para realizar la modulación FM a la frecuencia deseada. Para resolver este problema se utilizó una onda cuadrada de menor frecuencia que pueda modular la señal de audio a la frecuencia deseada utilizando la propiedad de las ondas cuadradas que en su espectro en frecuencia poseen la forma de una señal sampling ($\frac{\sin x}{x}$).
- Al usar una onda cuadrada como señal portadora para realizar la modulación, se generan además de la señal modulada deseada en la frecuencia deseada, otras réplicas en diferentes bandas de frecuencia. Es decir, estamos ocupando casi todo el espectro de frecuencias menores a la frecuencia deseada.

- El uso de un procesador embebido como el Nios II facilita la reconfiguración del sistema utilizando únicamente software. Además la herramienta Nios IDE permite una fácil depuración del sistema.

- El uso de la herramienta SOPC builder de Altera permite una rápida implementación de prototipos de sistemas embebidos complejos que tomarían mucho tiempo en diseñar desde el inicio.

RECOMENDACIONES

Debido a la utilización de una señal cuadrada como onda portadora, la calidad de la señal transmitida no es tan buena. Para mejorar la calidad de la señal de audio modulada se podrían utilizar los filtros digitales que tiene incluido el CODEC de audio. Otra opción es utilizar la tarjeta USRP (Universal Software Radio Peripheral) que provee de los RF front-ends necesarios para hacer una correcta modulación FM. Para utilizarla se debe implementar un módulo de comunicación USB con la tarjeta DE2 para emular los datos que le envía el computador a la tarjeta USRP.

El concepto de software radio se mantiene en el sistema diseñado pero se puede mejorar si es que todo el hardware del sistema es controlado por un procesador. Para lograr esto se necesitan elaborar los drivers necesarios para los módulos diseñados. Se necesita un conocimiento mas profundo del control del procesador sobre el hardware utilizando la plataforma HAL (Hardware Abstraction Layer) que provee el fabricante Altera.

BIBLIOGRAFÍA

[1] INSTITUTO NACIONAL DE DEFENSA CIVIL
1998 Plan de respuesta inmediata por el SINADECI ante situaciones de emergencia y/o desastre. [en línea] [consultado 15/10/2006] <http://www.indeci.gob.pe/planes_proy_prg/p_operativos/p_oper_emerg/p_noe/plan_rpta_inmed_sinadeci.pdf>

[2] BURACCHINI, Enrico.
2000 The Software Radio Concept. *IEEE Communications Magazine.* Septiembre 2000: 138 – 143.

[3] BING, Benny y JAYANT, Nikil.
2002 A Cellphone For All Standards. *IEEE Spectrum*. Mayo 2002: 34 – 39.

[4] HARRIS CORPORATION
2006 FALCON® II Global Family of Products.[en línea] [consultado 27/09/2006] http://www.rfcomm.harris.com/products/tactical-radio-communications/.

[5] MITOLA, Joe.
1995 The Software Radio Architecture. *IEEE Communications Magazine.* Mayo 1995: 26 -38.

[6] DICK, Chris.
2006 Design and Implementation of High-Performance FPGA Signal Processing Datapaths for Software Defined Radios. [en línea] [consultado 27/09/2006] <http://www.xilinx.com/esp/mil_aero/collateral/ProductEngineering/design_implementaion.pdf>

[7] BROWN, Stephen y ROSE, Jonathan.
1996 Architecture of FPGAs and CPLDs: A Tutorial. *IEEE Design and Test of Computers*, Vol. 13, No. 2, pp. 42-57.

[8] ALTERA CORPORATION
2007 Design Recommendations for Altera Devices and the Quartus II Design Assistant. [en línea] [consultado 15/10/2007] <http://www.altera.com/literature/hb/qts/qts_qii51006.pdf>

[9] ALTERA CORPORATION
2007 Quartus II Version 7.2 Handbook Volume 4: SOPC Builder. [en línea] [consultado 15/10/2007] <http://www.altera.com/literature/hb/qts/qts_qii5v4.pdf>

[10] ANALOG DEVICES, INC.
2004 Ask The Application Engineer—33. All About Direct Digital Synthesis. [en línea] [consultado 03/08/2007] <http://www.analog.com/library/analogdialogue/archives/38-08/dds.pdf>

[11] ANALOG DEVICES, INC.
1999 A Technical Tutorial on Digital Signal Synthesis. [en línea] [consultado 03/08/2007]

<http://www.analog.com/UploadedFiles/Tutorials/450968421DDS_Tutorial_rev12-2-99.pdf>

[12] WOLFSON MICROELECTRONICS
2005 Portable Internet Audio CODEC with Headphone Driver and Programmable Sample Rates WM8731 / WM8731L. [en línea] [consultado 09/08/2007] <http://www.wolfsonmicro.com/uploads/documents/WM8731_8731L.pdf>

[13] ANALOG DEVICES
2005 CMOS, 330 MHz. Triple 10-Bit High Speed Video DAC [en línea] [consultado 09/08/2007] <http://www.analog.com/UploadedFiles/Data_Sheets/ADV7123.pdf>

[14] SÓÑORA MENGANA A.
2003 Diseño de un sintetizador utilizando un FPGA. *Memorias del V congreso de la sociedad cubana de bioingeniería*, Habana. Centro de Biofísica Médica. Universidad de Oriente. Santiago de Cuba. Cuba.

[15] PECHANEC Martin.
1997 Output Spectrum of a Direct Digital Synthesizer [en línea] [consultado 26/08/2007] <http://www.geocities.com/CapeCanaveral/5611/dds.html>

[16] ALTERA CORPORATION
2007 Cyclone II Device Handbook, Volume 1. [en línea] [consultado 05/05/2007] <http://www.altera.com/literature/hb/cyc2/cyc2_cii5v1.pdf>

[17] ALTERA CORPORATION
2007 Recommended HDL Coding Styles. [en línea] [consultado 23/02/2007] <http://www.altera.com/literature/hb/qts/qts_qii51007.pdf>

[18] ALTERA CORPORATION
2007 Nios II Performance Benchmarks. [en línea] [consultado 15/10/2007] <http://www.altera.com/literature/ds/ds_nios2_perf.pdf>

[19] SYNPLICITY INCORPORATED
2004 Nios II: An Extremely Versatile Processor. A Technical Newsletter for ASIC and FPGA Designers [en línea] [consultado 02/11/2007] <http://www.synplicity.com/literature/syndicated/pdf/v4_i2/niosII_v4_i2.pdf>

[20] ALTERA CORPORATION
2007 Nios II Processor Reference Handbook. [en línea] [consultado 15/10/2007] <http://www.altera.com/literature/hb/nios2/n2cpu_nii5v1.pdf>

[21] ALTERA CORPORATION
2006 Development and Education Board 2 User Manual. [en línea] [consultado 20/04/2007]

<http://www.altera.com/education/univ/materials/boards/DE2_UserManual.pdf>

[22] ALTERA CORPORATION
2007 Nios II Software Developer´s Handbook. [en línea] [consultado 15/10/2007]
<http://www.altera.com/literature/hb/nios2/n2sw_nii5v2.pdf>

[23] ALTERA CORPORATION
2007 Simulating Nios II Embedded Processor Designs. [en línea] [consultado 01/11/2007]
<http://www.altera.com/literature/an/an351.pdf>