

2019

DIGITAL SYNTHESIZER BASED ON FPGA



- Geovanni Santamaria Lopez
- Miguel Maroto Lema

ESPOL

INTRODUCCIÓN

El objetivo de este proyecto es realizar un sistema embebido en la tarjeta DE10 Standard con FPGA SoC que se comporte como un sintetizador digital con el cual podamos tener un control sobre las tres dimensiones 'macro' de un sonido: su altura tonal, su timbre y su intensidad.

Inicialmente crearemos nuestro sistema embebido en QSYS de Quartus, bajo las directrices

indicadas tales como sistema multinúcleos, reloj, memorias y puertos de comunicación correspondientes.

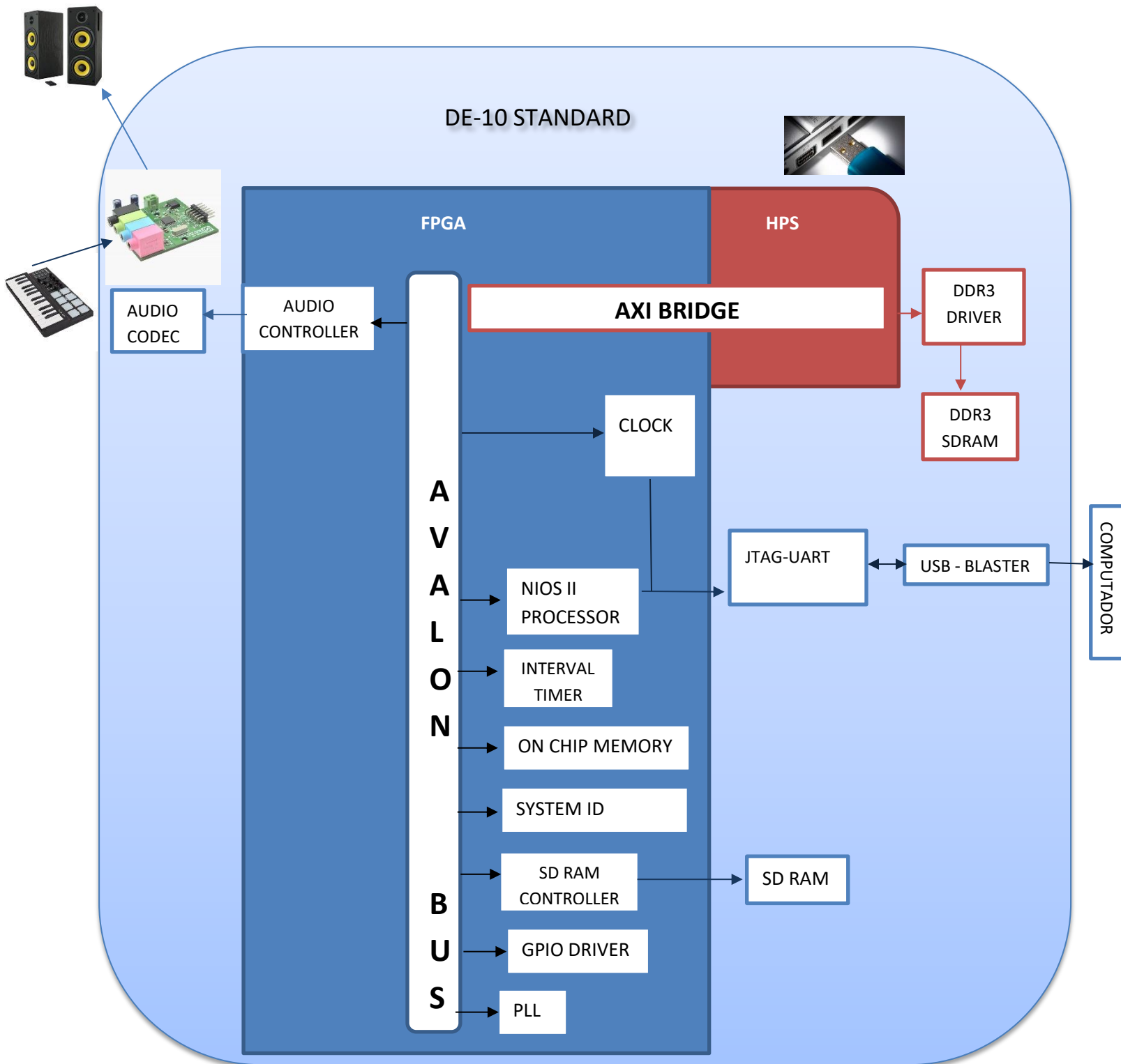
Usando lenguaje C en el entorno de Eclipse de Quartus, programaremos el sistema de tal manera que sea capaz de reconocer todos los drivers de los dispositivos periféricos a utilizar como las entradas de control del sintetizador o las salidas como pueden ser los parlantes.

Los resonadores se llevarán a cabo mediante filtros. La señal del oscilador se filtra y posteriormente se amplifica. Son esos 3 elementos los que definen el camino que sigue el audio. Pero además aparecen el EG y LFO destinados a conseguir sonidos que se mueven, cambian, evolucionan a lo largo del tiempo.

El proceso consiste en que el sistema embebido debe tener como ingreso una señal de entrada la cual serán las notas de un piano utilizando los puertos GPIO para su posterior procesamiento. Adicional a esto se puede almacenar una melodía dependiendo si el usuario lo requiera. Utilizando la tarjeta de desarrollo se procesa la señal analógica y se configura mediante una interfaz de usuario ciertos parámetros como intensidad, volumen, timbre, etc. La interfaz de usuario es emitida a través de una pantalla/monitor mediante un puerto VGA.

El resultado final del sonido producido por todo el sistema que compone el sintetizador podrá ser escuchado a través de unos parlantes estéreo los cuales estarán conectados al puerto de salida de audio de la FPGA. Una tarjeta de memoria SD podría ser utilizada para realizar el almacenamiento de melodías que posteriormente pueden ser reproducidas por el usuario y que serán escuchadas también a través de los altavoces.

ARQUITECTURA DE HARDWARE



DESCRIPCIÓN

CPU1

Este procesador es utilizado para el proceso de recibir y transmitir el audio del sintetizador, además de realizar el almacenamiento de melodías. Se utiliza un NIOS II /f debido a que como se va a trabajar con audio a tiempo real se necesita el procesador más rápido.

Onchip_memory1

Es la memoria dedicada del CPU. En estas se introducirá la memoria de programa del procesador para el correcto funcionamiento del sistema.

TIMER

Debido a que este bloque permite tener contadores (de 64/32 bits), controles para el inicio, paro o restablecimiento del temporizador, modos de conteo, opción para deshabilitar o habilitar la solicitud de interrupción (IRQ), temporizador de vigilancia el cual restablece el sistema cuando el cronómetro llega a cero es indispensable su uso ya que se trabaja con cambios de ondas y señales en el tiempo.

System ID

Este bloque es requerido en cualquier sistema que se esté creando, sobre todo es utilizado ahora porque nuestro sistema es multicore. Gracias a este bloque cuando se descarga software por medio de Eclipse, se puede elegir el procesador que ejecutará el código basándose en función del ID del sistema. También se evita que existan faltas de coincidencia entre configuraciones de hardware y software.

JTAG UART

El núcleo JTAG UART, realiza la comunicación de caracteres realizando transmisiones entre una PC host y un sistema QSYS. El JTAG- UART proporciona una interfaz Avalon que oculta las complejidades de la interfaz JTAG del software integrado programadores.

PLL

Son utilizados 2 bloques PLL uno exclusivo para la sección de audio y otro para los procesadores y el resto del sistema.

AUDIO

Es utilizado junto a los bloques i2c_sda, i2c_scl para poder utilizar los puertos de salida y entrada de audio de la tarjeta de desarrollo

GPIO

Los bloques KEY, 7SEG, LED, SW corresponden a las botoneras, pantalla de 7 segmentos, leds y interruptores que la tarjeta de desarrollo tiene incorporada, estos serán utilizados en diversas funciones, por ejemplo: los interruptores se utilizarán

para seleccionar el tipo de forma de onda del oscilador, y una botonera servirá para realizar la grabación del sonido.

SDRAM

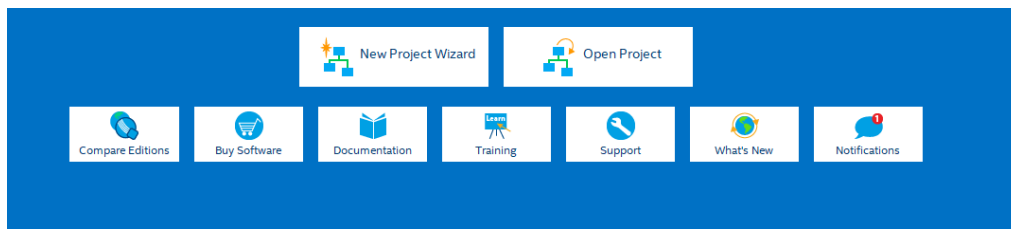
Con este bloque se espera tener un buen rendimiento de memoria para realizar un sistema de procesamiento de sonido, almacenamiento y las comunicaciones de nuestro sistema.

Bus Avalon

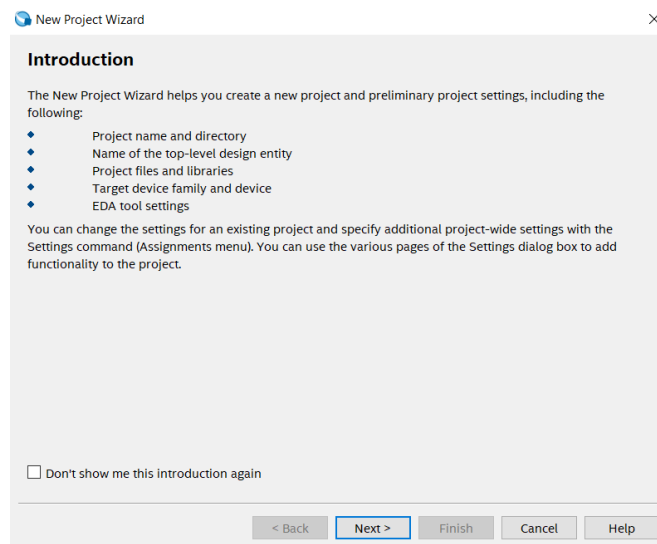
Es utilizado como interfaz de comunicación entre los procesadores, periféricos y distintos bloques de todo el sistema.

PROCEDIMIENTO PARA REALIZAR LA ARQUITECTURA DEL SISTEMA.

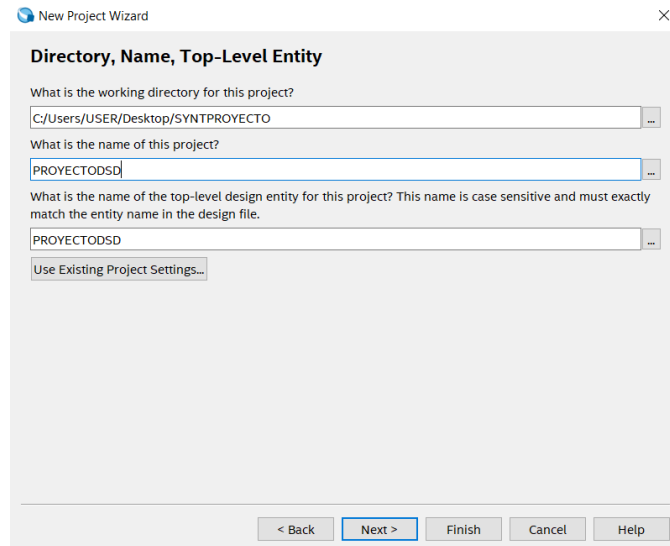
1. Se crea un proyecto nuevo presionando en “New Project Wizard”



2. Presionar “Next” en la ventana que se despliega.

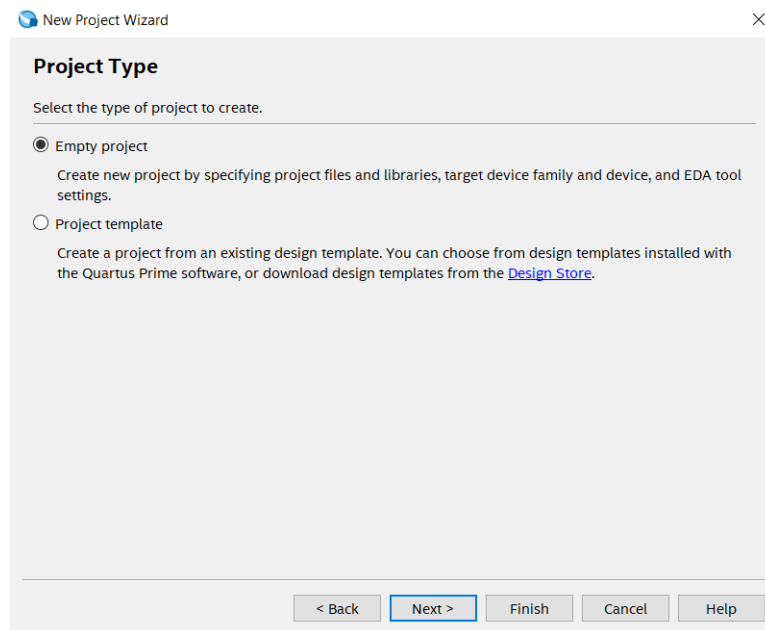


3. Se procede a crear una carpeta con el nombre del proyecto y de igual manera se escribe el nombre del proyecto. Dar clic en “next”.



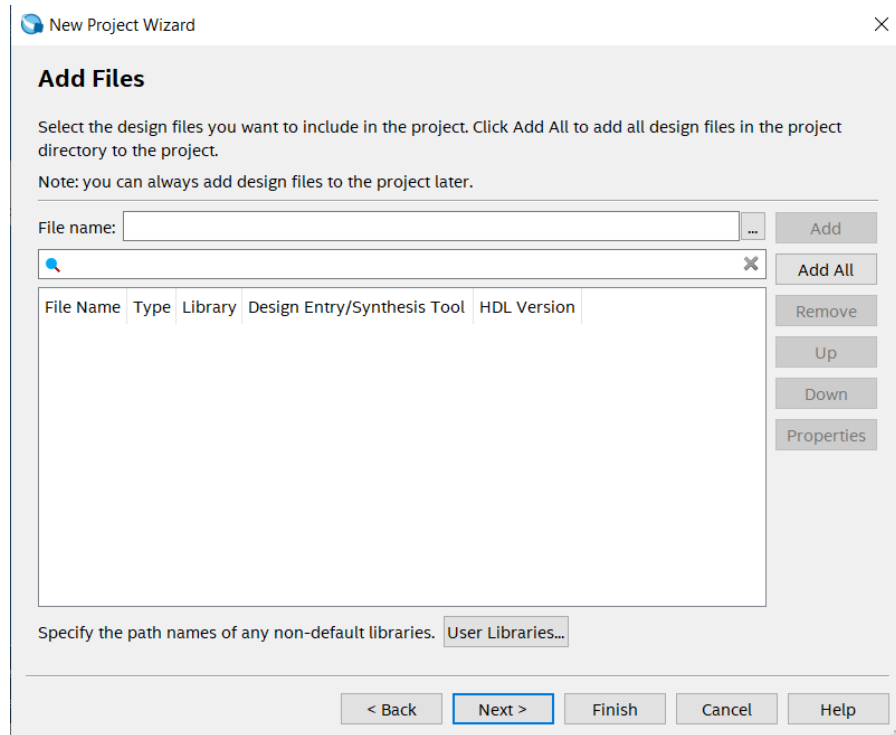
The screenshot shows the 'New Project Wizard' dialog box, specifically the 'Directory, Name, Top-Level Entity' step. The dialog has a title bar with a blue icon and the text 'New Project Wizard'. The main area contains three text input fields and a button. The first field is labeled 'What is the working directory for this project?' and contains the text 'C:/Users/USER/Desktop/SYNTPROYECTO'. The second field is labeled 'What is the name of this project?' and contains the text 'PROYECTODSD'. The third field is labeled 'What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.' and contains the text 'PROYECTODSD'. Below these fields is a button labeled 'Use Existing Project Settings...'. At the bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

4. Seleccionar la creación de un proyecto vacío.

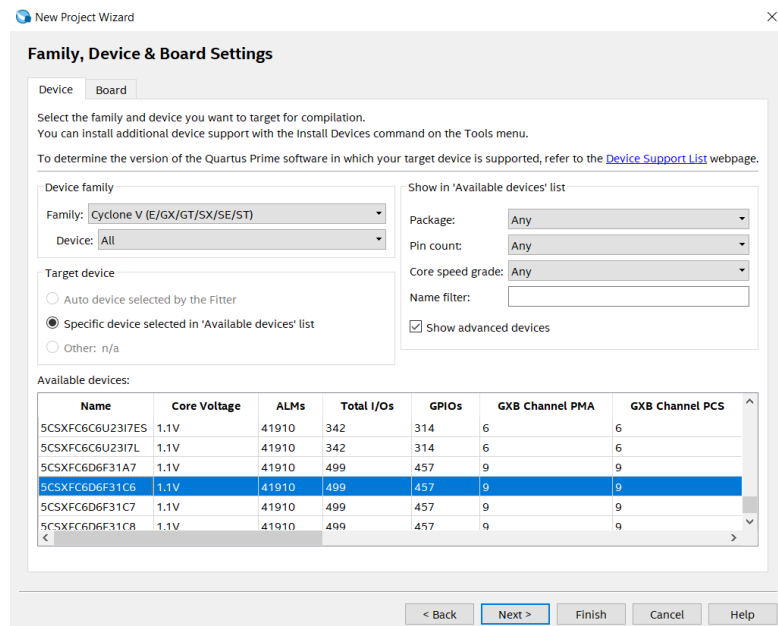


The screenshot shows the 'New Project Wizard' dialog box, specifically the 'Project Type' step. The dialog has a title bar with a blue icon and the text 'New Project Wizard'. The main area contains a section titled 'Project Type' with the instruction 'Select the type of project to create.' Below this are two radio button options. The first option is 'Empty project', which is selected (indicated by a filled radio button). Below it is a description: 'Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.' The second option is 'Project template', which is not selected (indicated by an empty radio button). Below it is a description: 'Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the [Design Store](#).' At the bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

5. Debido a que no se agregarán archivos, se da clic en siguiente.



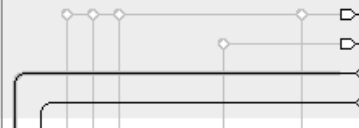

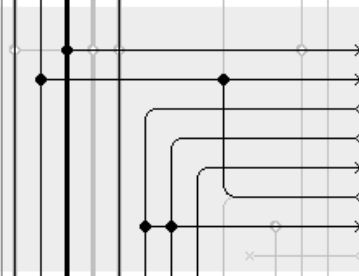

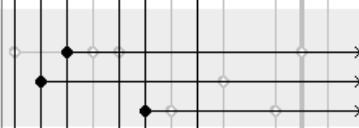

6. En esta ventana seleccionar el dispositivo a utilizar, para este proyecto se utiliza la tarjeta de la familia Cyclone V (E/GX/GT/SX/SE/ST) y el modelo del chip es el **5CSXFC6D6F31C6**. Una vez seleccionada la tarjeta se puede dar clic en finalizar.

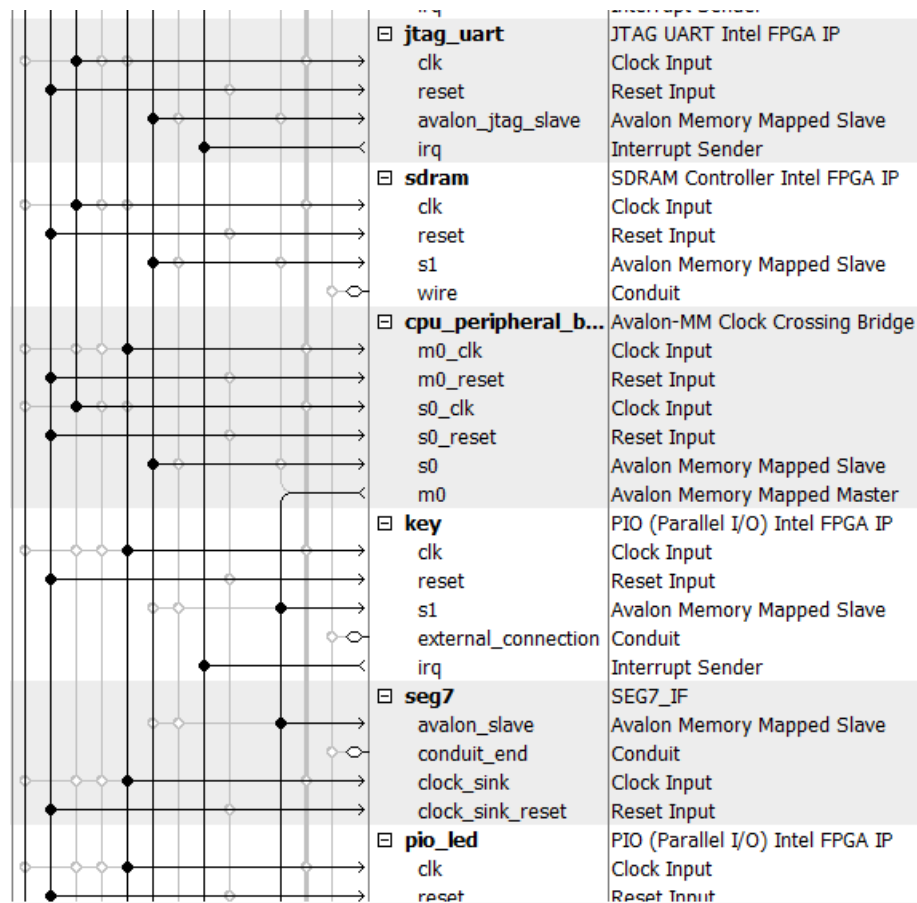


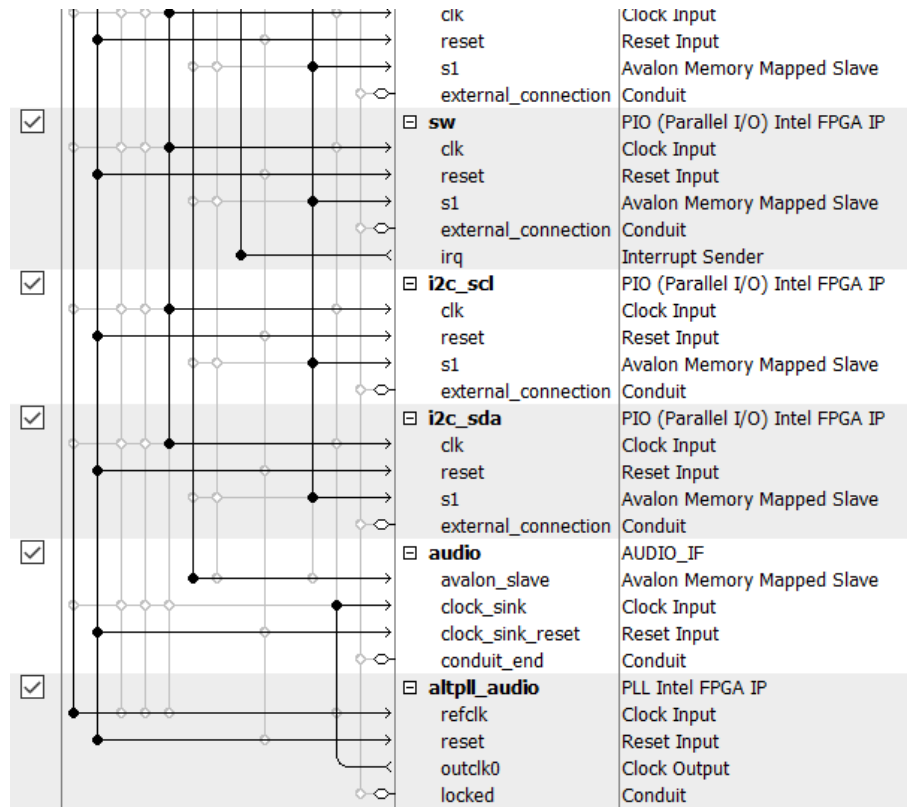
7. Una vez realizados estos pasos se procede a crear la arquitectura del proyecto, esto se realizará abriendo el software Qsys.



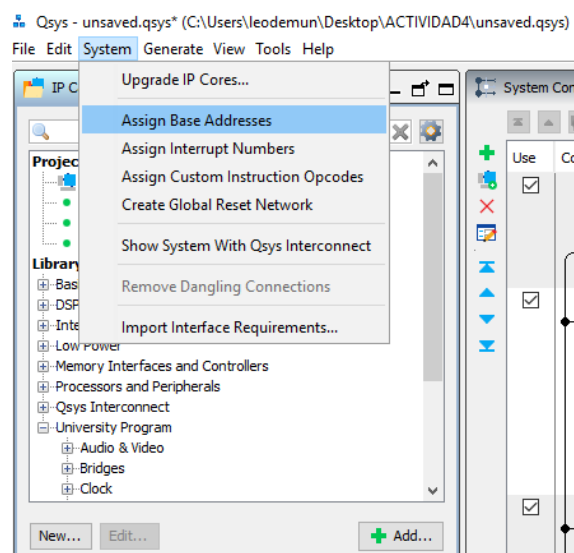
8. En Qsys, se procederá a añadir todos los componentes de las siguientes imágenes, realizando además cada una de las conexiones que se observan para un correcto funcionamiento.

Use	Connections	Name	Description
<input checked="" type="checkbox"/>		clk_50	Clock Source
		clk_in	Clock Input
		clk_in_reset	Reset Input
		clk	Clock Output
		clk_reset	Reset Output
<input checked="" type="checkbox"/>		pll	PLL Intel FPGA IP
		refclk	Clock Input
		reset	Reset Input
		outclk0	Clock Output
		outclk1	Clock Output
		outclk2	Clock Output
		locked	Conduit
<input checked="" type="checkbox"/>		nios2_gen2_0	Nios II Processor
		clk	Clock Input
		reset	Reset Input
		data_master	Avalon Memory Mapped Master
		instruction_master	Avalon Memory Mapped Master
		irq	Interrupt Receiver
		debug_reset_requ...	Reset Output
		debug_mem_slave	Avalon Memory Mapped Slave
		custom_instructio...	Custom Instruction Master
<input checked="" type="checkbox"/>		onchip_memory2	On-Chip Memory (RAM or ROM)...
		clk1	Clock Input
		s1	Avalon Memory Mapped Slave
<input checked="" type="checkbox"/>		sysid_qsys	System ID Peripheral Intel FPGA...
		clk	Clock Input
		reset	Reset Input
		control_slave	Avalon Memory Mapped Slave
<input checked="" type="checkbox"/>		timer	Interval Timer Intel FPGA IP
		clk	Clock Input

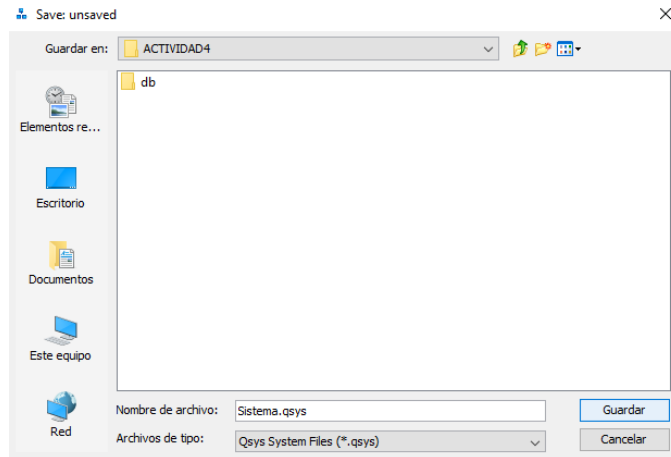




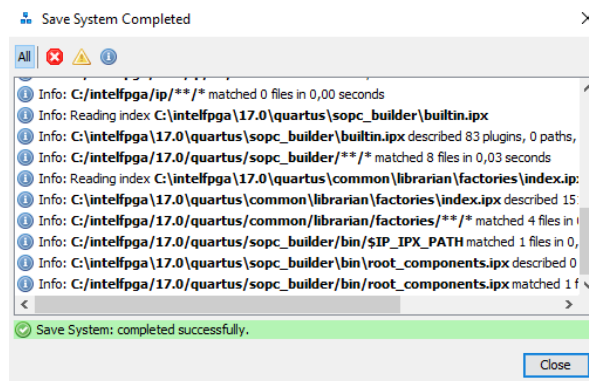
9. AL terminar de realizar las conexiones y configuraciones mostradas se procede a asignar el direccionamiento de direcciones de todos los bloques, esto dirigiéndose al menú **System** y seleccione la opción **Assign Base Addresses**, tal cual se muestra en la siguiente figura.



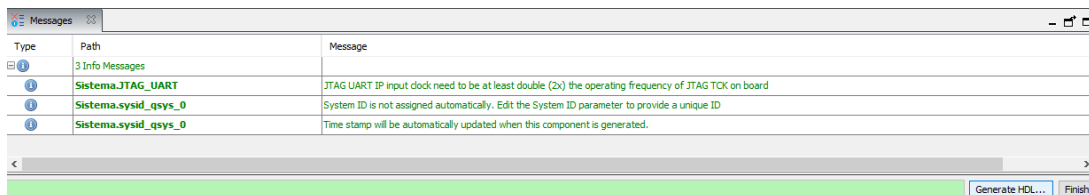
10. Finalmente se procede a guardar el archivo de Qsys con el nombre de preferencia haciendo clic en el menú File la opción “Save As” como se muestra en la siguiente figura.



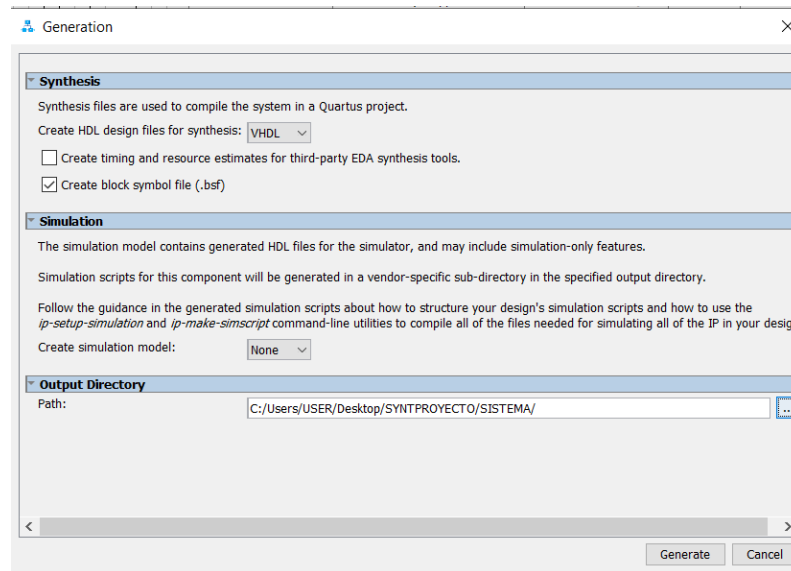
11. Una vez guardado se procederá a mostrar la siguiente ventana, dar clic en close.



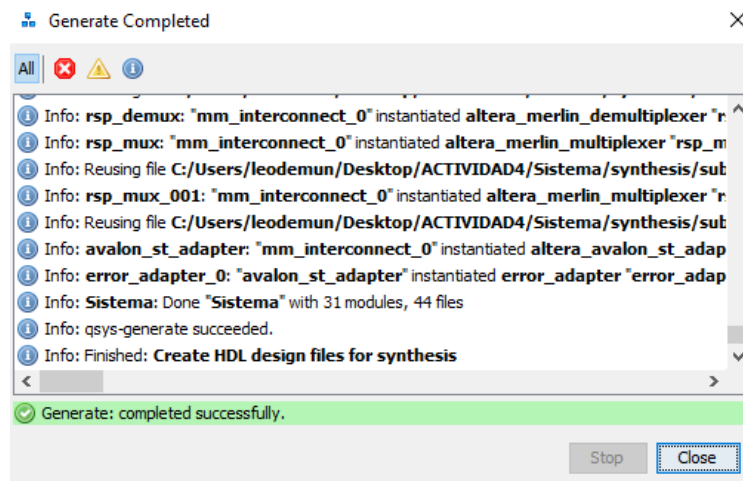
12. Dar clic en **Generate HDL** en la parte inferior derecha de la ventana del Qsys..



13. Una vez finalizada la arquitectura en Qsys se procede a generar el HDL, seleccionar "Create HDL design files for synthesis" VHDL y dar click en Generate. Si este proceso se realiza sin problemas, saldrá un mensaje de que el HDL se ha generado, caso contrario proceda a revisar que las conexiones estén bien realizadas.



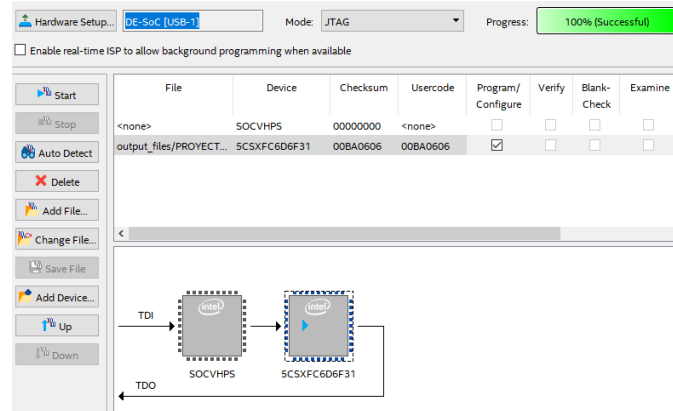
14. Aparecerá esta ventana cuando la generación se haya realizado de manera exitosa.



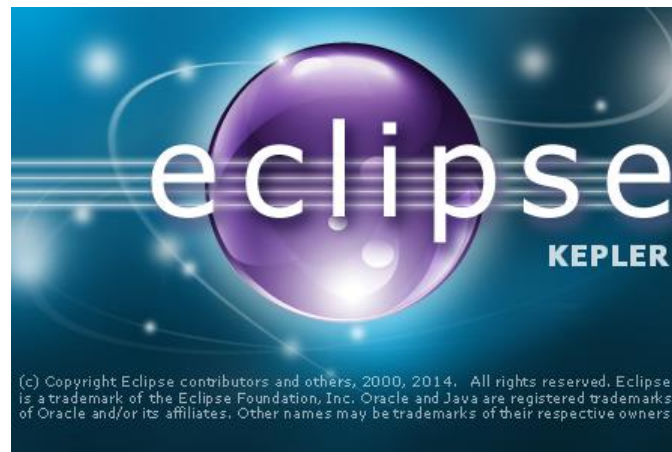
15. Una vez generado el HDL, realizar la compilación del archivo generado, para esto establecer el archivo como más alta jerarquía y proceder a realizar la compilación completa.



16. Finalmente se procede a realizar la carga a la FPGA, para esto dar click en la ventana de programador. Seleccionar la penúltima opción al dar click en auto detect y reemplazar el segundo archivo por el archivo que se encuentra en la carpeta “output files” de su proyecto. Dar click en start y esperar a que la carga se realice satisfactoriamente.



17. Una vez cargado se procede a abrir el eclipse para poder realizar la programación que se requiera para el funcionamiento del proyecto.



18. En Eclipse se procede a seleccionar la carpeta “software” de nuestro proyecto, para posteriormente empezar a crear los archivos que nos permitirán realizar el funcionamiento del proyecto. Primero se debe dar clic en “**New→Nios II Application and BSP from Template**”.
19. En ventana que se abre se debe buscar el archivo .SOPCINFO generado en nuestra carpeta. Una vez seleccionado el archivo, se procede a seleccionar el CPU con el que vamos a trabajar.
20. En la sección de “Project Template” seleccionar la opción “Blank Project”, escribir un nombre al proyecto y dar clic en “Finish”.

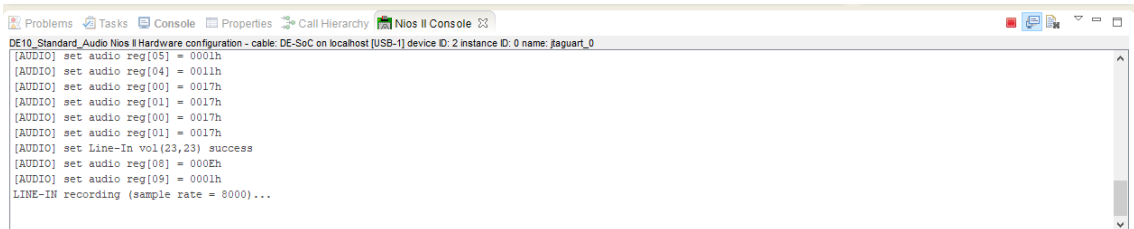
21. Se generarán 2 carpetas una con el nombre del proyecto y otra con el mismo nombre pero finalizando en BSP.
22. En la carpeta del proyecto se debe crear un archivo al que se le debe colocar como nombre main.c
23. En el archivo main.c se debe copiar el código que se encuentra en el archivo "proyecto.txt".

```

178
179
180
181 bool init_audio(AUDIO_FUNC audio_func){
182     bool bSuccess = TRUE;
183     AUDIO_InterfaceActive(FALSE);
184     //
185     if (audio_func == MIC_RECORD){
186         bool bMicBoost;
187         bMicBoost = ui_is_mic_boost();
188         AUDIO_SetInputSource(SOURCE_MIC);
189         AUDIO_DacEnableSoftMute(TRUE);
190         AUDIO_AdcEnableHighPassFilter(FALSE);
191         AUDIO_MicBoost(bMicBoost);
192         AUDIO_MicMute(FALSE);
193         AUDIO_LineInMute(TRUE);
194     }else if (audio_func == LINEIN_RECORD){
195         AUDIO_SetInputSource(SOURCE_LINEIN);
196         AUDIO_DacEnableSoftMute(TRUE);
197         AUDIO_AdcEnableHighPassFilter(FALSE);
198         AUDIO_MicMute(TRUE);
199         AUDIO_LineInMute(FALSE);
200         AUDIO_SetLineInVol(0x17, 0x17); // max 0x1F, min:0; 0x17: 0dB (assume max input is 2.0v rms)
201     }else if (audio_func == LINEOUT_PLAY){
202         AUDIO_DacEnableSoftMute(TRUE);
203         AUDIO_MicBoost(FALSE);
204         AUDIO_MicMute(TRUE);
205         AUDIO_LineInMute(FALSE);
206         AUDIO_DacEnableSoftMute(FALSE);
207         //AUDIO_DacDeemphasisControl(DEEMPHASIS_48K);
208         AUDIO_DacEnableZeroCross(ui_is_dac_zero_cross());
209         AUDIO_SetLineOutVol(LINEOUT_DEFAULT_VOL, LINEOUT_DEFAULT_VOL); // max 7F, min: 30, 0x79: 0 db
210         AUDIO_DacEnableSoftMute(FALSE);

```

24. Se procede a guardar "main.c" y a cargar el proyecto dando click derecho a la carpeta y seleccionar la opción "Build Project". Una vez que se haya compilado el proyecto se procede a dar click en "Run As NIOS II Hardware".
25. Una vez cargado el archivo se podrá utilizar el sintetizador y se observará por consola lo siguiente:



```

DE10_Standard_Audio Nios II Hardware configuration - cable: DE-SoC on localhost [USB-1] device ID: 2 instance ID: 0 name: #aguart_0
[AUDIO] set audio reg[05] = 0001h
[AUDIO] set audio reg[04] = 0011h
[AUDIO] set audio reg[00] = 0017h
[AUDIO] set audio reg[01] = 0017h
[AUDIO] set audio reg[00] = 0017h
[AUDIO] set audio reg[01] = 0017h
[AUDIO] set Line-In vol(23,23) success
[AUDIO] set audio reg[08] = 000Eh
[AUDIO] set audio reg[09] = 0001h
LINE-IN recording (sample rate = 8000)...
```

Instrucciones:

1. Para conectar un piano/instrumento a la FPGA se pueden utilizar las entradas de micrófono o la entrada "Line/in" (color azul) en la parte donde se encuentran las entradas y salidas de audio. Para poder escuchar se recomienda utilizar parlantes o audífonos estéreo.
Si se utiliza la entrada de micrófono (color rosado) se debe tener el switch 1 de la FPGA desactivado (hacia abajo), caso contrario habilitarlo para poder usar la otra entrada de audio.
2. Para grabar el audio se procede a presionar la botonera "Key 3" de la tarjeta y para proceder a reproducir el audio grabado se presiona la botonera "Key 2". Al implementar este sintetizador se pudo realizar un sonido con forma de onda sinusoidal.

Resultados:

Al finalizar el proyecto se pudieron obtener como resultados un sintetizador el cual tenía la capacidad de alterar el sonido de una entrada de audio a un sonido con característica sinusoidal, esto se podía escuchar claramente al momento de realizar la grabación o reproducción del sonido que entraba a la tarjeta de desarrollo. Lo interesante de este sintetizador, es la capacidad de poder realizar esos cambios en el sonido sin alterar la calidad del mismo pudiendo dar la oportunidad de no solo experimentar con un piano, si no también con otros instrumentos o melodías previamente almacenadas.

Conclusiones y recomendaciones:

Conclusiones:

- Se pudo realizar el diseño e implementación de un sintetizador basado en un sistema embebido en una FPGA de manera satisfactoria realizando la síntesis de sonido en una onda sinusoidal.
- La implementación de un espacio de memoria en la FPGA para el almacenamiento de una melodía fue realizado de manera correcta, dándole al usuario la oportunidad de almacenar una melodía tocada en un piano o instrumento conectado a la FPGA.

- Se pudo realizar una correcta arquitectura donde los distintos componentes de la FPGA interactuaban para poder mostrar datos (por ejemplo el tiempo de grabación) o para realizar acciones como lo eran la grabación y reproducción del sonido.
- La FPGA es un dispositivo bastante recomendable para este tipo de sistemas debido a su alta rapidez, se pudo comprobar que utilizando un CPU modelo F (rápido) el sistema pudo correr con fluidez a pesar de la carga que representa el almacenamiento, reproducción y alteración del audio debido a la onda sinusoidal.
- La posibilidad de utilizar una entrada de audio estéreo fue posible gracias a la tarjeta de desarrollo, permitiendo así tener mayores opciones a nivel sonoro.

Recomendaciones:

- Reforzar los conocimientos en lenguaje C ya que son primordiales para el desarrollo de este proyecto.
- Seguir cada uno de los pasos de la guía para evitar fallos de funcionamiento.
- La incorporación de filtros en futuras versiones de este proyecto, se deben realizar con bastante cuidado debido a que en ocasiones los filtros no filtraban el ruido si no más bien el sonido del piano.
- Almacenar el proyecto en distintos lugares antes de realizar cambios.
- Tener una FPGA para poder trabajar en el proyecto.