



Escuela Colombiana de Ingeniería Julio Garavito  
Programa de Ingeniería de Sistemas

Trabajo de grado:  
Migración del Hardware proyecto Oberon 2013 de  
Spartan 3 a Spartan 3E

Presentado por:  
Cesar Alexander Bonilla Hernández

Propuesta de Trabajo de Grado para optar por el título de  
Ingeniero de Sistemas

Tutor:  
Ing. Gerardo Ospina Hernández



# Contents

<b>1</b>	<b>OBJETIVO</b>	<b>5</b>
1.1	Planteamiento del problema . . . . .	5
1.2	Justificación . . . . .	5
1.3	Objetivo general . . . . .	6
1.4	Objetivo Especificos . . . . .	6
1.5	Área de aplicación del producto resultado del proyecto . . . . .	6
<b>2</b>	<b>MARCO TEÓRICO</b>	<b>7</b>
2.1	Proyecto Oberon 2013 . . . . .	7
2.2	Dispositivos lógicos programables. . . . .	10
2.3	Memoria RAM . . . . .	12
2.4	Osciladores . . . . .	13
<b>3</b>	<b>AMBIENTE DE DESARROLLO</b>	<b>15</b>
3.1	Ambiente de desarrollo FPGA Spartan 3 y Spartan 3E . . . . .	15
3.2	Ambiente de desarrollo diseño y construcción de hardware . . . . .	16
3.3	Ambiente de simulación de hardware . . . . .	18
3.4	Herramientas utilizadas . . . . .	18
<b>4</b>	<b>RESULTADOS</b>	<b>19</b>
<b>5</b>	<b>CONCLUSIONES</b>	<b>25</b>
<b>6</b>	<b>TRABAJOS FUTUROS</b>	<b>27</b>



# Chapter 1

## OBJETIVO

### 1.1 Planteamiento del problema

El proyecto Oberon 2013 es un Sistema Operativo concebido a partir del hardware hasta llegar a las aplicaciones, con el objetivo de comprender el funcionamiento de los Sistemas Operativos y sus componentes para luego desarrollar nuevos sistemas o permitir la intección de otros más complejos.

El sistema fue creado en un Kit de Desarrollo Spartan 3, con características propias en los dispositivos y en los conectores de entrada y salida como puertos PS2, puerto para conexiones seriales y puerto RGB, entre otros, y en la integración de memorias y chips para ejecutar tareas específicas.

La continua evolución de la tecnología ha llevado a que muchos de los productos queden fuera del mercado y se varíen, por tanto, las condiciones para procesar los desarrollos, obligando así al cambio de hardware en los proyectos y su reemplazo por los dispositivos actuales.

Para este caso en particular, se reemplaza el Spartan 3 por el Spartan 3E. Una diferencia entre los dos Kits de Desarrollo radica en la memoria y en la distribución de los pines, hecho que exige una nueva revisión y adaptar los cambios correspondientes a fin de que el Proyecto conserve los objetivos iniciales.

### 1.2 Justificación

El Proyecto Spartan3 se desarrolla para mantener el Sistema Operativo Oberon que facilita el estudio y aprendizaje de los conceptos básicos de los Sistemas

Operativos y la comprensión de sistemas con mayor complejidad, o permitir el desarrollo de sistemas más pequeños, como los Sistemas Embebidos, en los que no se cargan todos los módulos sino solamente los necesarios para una determinada función.

### 1.3 Objetivo general

Migrar el Proyecto Oberon 2013 de Spartan 3 hacia Spartan 3E conservando el funcionamiento del Sistema Operativo que allí se describe.

### 1.4 Objetivo Especificos

1. Diseñar y construir una PCB que conecte el módulo bluetooth, el mouse y la MicroSD a la tarjeta Spartan 3 para probar el funcionamiento del Sistema Oberon 2013.
2. Diseñar y construir una PCB que conecte el módulo bluetooth, el mouse y la MicroSD a la tarjeta Spartan 3E.
3. Revisar y modificar el sistema de direccionamiento de la memoria para la migración de SRAM a SDR SDRAM.

### 1.5 Área de aplicación del producto resultado del proyecto

La aplicación del Proyecto Oberon 2013 en la academia es con el fin de estudiar el Sistema Operativo para entender cómo se integran los diversos componentes, cómo es la unión de la parte lógica y el hardware en el cual está operando, de tal manera que, posteriormente, se puedan entender sistemas más complejos, como los desarrollados por las grandes empresas y la implementación de otros nuevos con el mejoramiento de algunas funcionalidades.

Otro campo en donde se utiliza el Proyecto Oberon 2013 es en los Sistemas Embebidos que debido a sus restricciones en memoria y procesamiento, no soportan grandes Sistemas Operativos, en cambio, el software del proyecto al ser modular permite que no se tenga que usar todo el sistema, sino que se implemente las partes que se requieran para las pruebas y aplicaciones se realizarán.

# Chapter 2

## MARCO TEÓRICO

### 2.1 Proyecto Oberon 2013

El Proyecto Oberon es el diseño de un Sistema Operativo, de un compilador y de un computador [1], y fue implementado en el año 2013. Niklans Wirth y Paul Reed elaboran la descripción y re-implementación del Proyecto Oberon en una FPGA Spartan 3.

El Sistema Operativo Oberon es un sistema modular creado en la década de los 80 [2]. Su nombre fue escogido en un pequeño homenaje a la sonda espacial Voyager por su singular esfuerzo y planificación de la investigación. En primera instancia, se observan los lenguajes de programación del momento. Se examina Modula-2 que no cumple con todas las características deseadas, lo que motivó a Niklans Wirth y Jürg Gutknecht a idear un nuevo lenguaje que pudiera ser aplicado, estudiado y entendido. Este lenguaje, aunque de alto nivel de abstracción, no depende directamente de la máquina en la que se encuentra [1].

Surge, entonces, el sistema preliminar Oberon en 1987 y desde ese momento se publica; luego, en 1991 [1], se escribe un manual y una guía. En 1988 se diseña e instrumentaliza el computador en un microprocesador NS32032 de la National Semiconductor [3], con base en el lenguaje creado.

Con el tiempo, el avance tecnológico hace que el microprocesador quede obsoleto y se deje de fabricar. Para solucionar el problema, algunos montaron el lenguaje sobre plataformas con núcleos adaptados a distintos procesadores y se obtiene Linux Native Oberon (LNO) [4] pero sacrificando parte de lo establecido en el proyecto original.

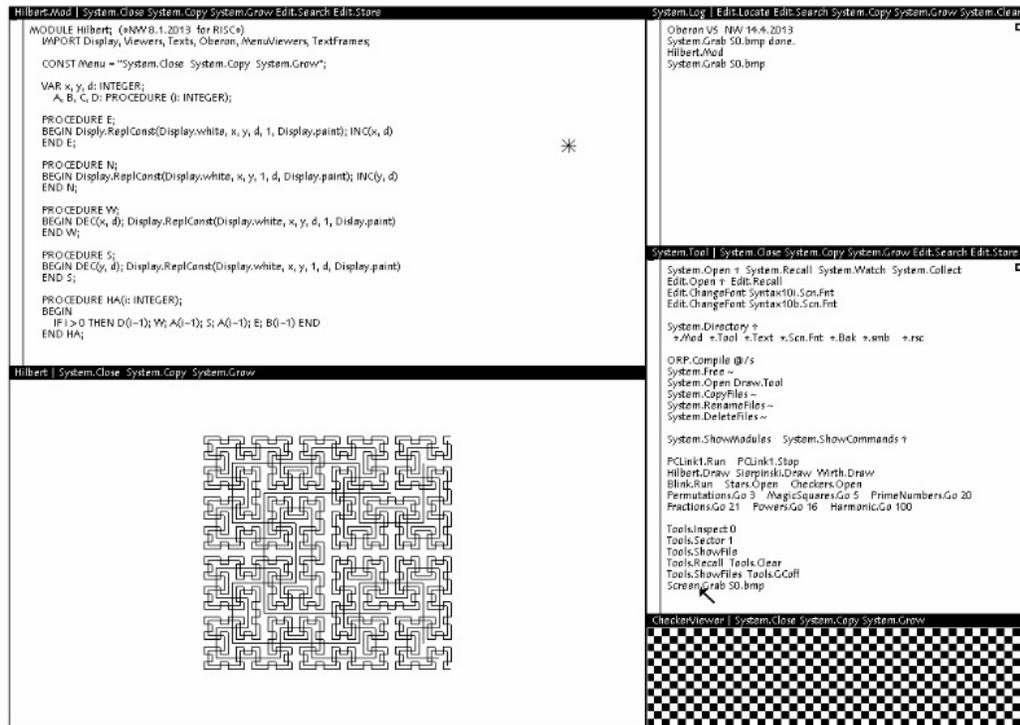


Figure 2.1: Vista Sistema Oberon [1]

En el Proyecto Oberon 2013 se retoman los objetivos originales, se implementa un procesador en una CPU RISC en un Kit de Desarrollo Spartan 3, para poder estudiar el sistema desde el hardware hasta las aplicaciones descritas en el lenguaje Oberon [1].

Los conceptos que se exponen en el Proyecto Oberon son: Vistas, Comandos, tareas y carga dinámica, con una estructura modular donde las interfaces son declaradas explícitamente, lo que facilita la observación de las interdependencias; además, algunos de los módulos se refieren directamente a los manejadores de los dispositivos, originando un acceso directo a los periféricos, lo que se denomina ‘módulos de comando’ (command modules) o ‘herramientas modulares’ (tool modules) [1]. Los módulos son archivos ejecutables en los que se pueden compilar y ejecutar determinadas tareas como, por ejemplo, el ‘módulo Hilbert’, de la Figura 2.1.

El Sistema cuenta con un hardware RISC que puede describirse como la conformación de dispositivos de entrada y de salida, con un sistema de memoria y uno de procesamiento lógico-matemático denominado ALU, en donde se realizan las operaciones del procesador y cuenta con las 16 instrucciones que

0	MOV	a, n	$R.a := n$	
1	LSL	a, b, n	$R.a := R.b \leftarrow n$	(shift left by n bits)
2	ASR	a, b, n	$R.a := R.b \rightarrow n$	(shift right by n bits with sign extension)
3	ROR	a, b, n	$R.a := R.b \text{ rot } n$	(rotate right by n bits)
4	AND	a, b, n	$R.a := R.b \& n$	logical operations
5	ANN	a, b, n	$R.a := R.b \& \sim n$	
6	IOR	a, b, n	$R.a := R.b \text{ or } n$	inclusive or
7	XOR	a, b, n	$R.a := R.b \text{ xor } n$	exclusive or
8	ADD	a, b, n	$R.a := R.b + n$	integer arithmetic
9	SUB	a, b, n	$R.a := R.b - n$	
10	MUL	a, b, n	$R.a := R.b \times n$	
11	DIV	a, b, n	$R.a := R.b \text{ div } n$	
12	FAD	a, b, c	$R.a := R.b + R.c$	floating-point arithmetic
13	FSB	a, b, c	$R.a := R.b - R.c$	
14	FML	a, b, c	$R.a := R.b \times R.c$	
15	FDV	a, b, c	$R.a := R.b / R.c$	

Figure 2.2: Instrucciones [1]

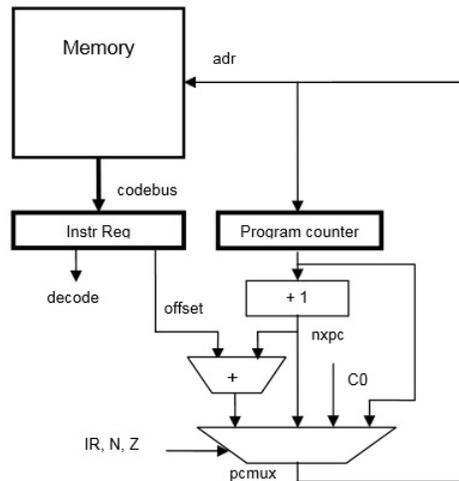


Figure 2.3: Diagrama Procesador [1]

aparecen en la Figura 2.2. Para el manejo de las instrucciones se dispone de un procesador (Diagrama de la Figura 2.3) donde se puede ver que los datos de salida son ingresados a una estructura de 16 registros, que conforman el 'Banco de Registros'. Allí las instrucciones son procesadas en forma combinatorial, excepto la multiplicación y la división que son ejecutadas por medio de ciclos [1].

Para efectuar la comunicación, el Sistema Oberon utiliza una serie de Buses para transportar información a través de diferentes protocolos como el R232, el SPI y el Serial [1]. (Figura 2.4)

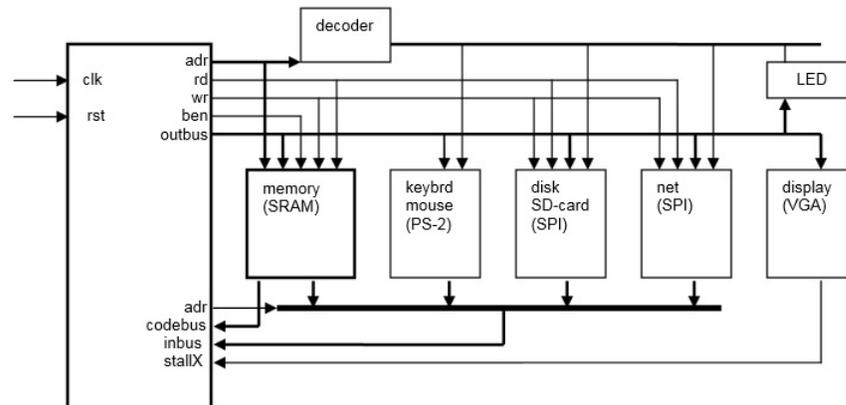


Figure 2.4: Estructura de buses [1]

## 2.2 Dispositivos lógicos programables.

Son dispositivos que permiten al usuario definir funciones lógicas después de ser construidos. Para tal fin existen varios tipos pero se hará referencia a dos de ellos: arreglos de lógica programable y matriz de células lógicas. [5, 6]

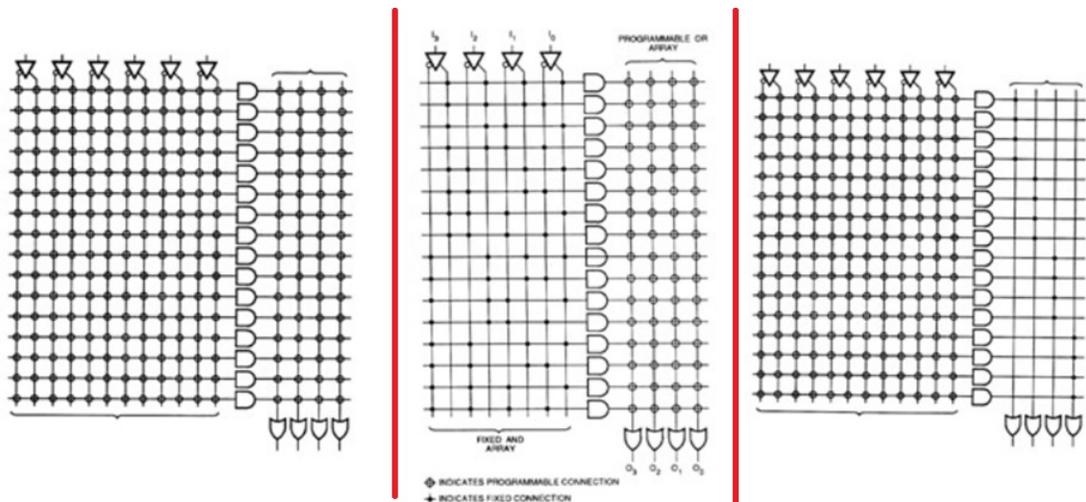


Figure 2.5: Arreglos lógica programable [7]

El primero posee las compuertas AND y OR y en él se programan las conexiones que se dirigirán hacia ellas. Pero no todos son programables en su totalidad. En algunos sólo se pueden programar las conexiones a las compuertas AND y se mantienen fijas las OR (PAL) lo que genera mayor rapidez a

los dispositivos al disminuir su versatilidad, mientras que en otros es posible programar los dos tipos de compuertas (PLA) [7]. Por último, existen dispositivos que permiten la programación de las compuertas OR y quedan fijas las compuertas AND (PROM). En la Figura 2.5, en el lado izquierdo, se ilustra el dispositivo con el cual se pueden programar las compuertas AND y OR; en el centro, con el que se pueden programar las compuertas OR, dejando fijas las AND; con el de la derecha, sólo se programan las compuertas AND.

El otro tipo de dispositivo contiene más módulos programables como los de entrada y salida (IOBs), los Bloques Lógicos Configurables (CLBs), las líneas de interconexión programables. Los módulos se conectan entre sí formando redes (Nets) a través de las líneas de interconexión, que transmiten señales entre módulos, en tanto que los IOBs proporcionan las conexiones hacia los pines externos del circuito y sirven de puente para la lógica interna [6].

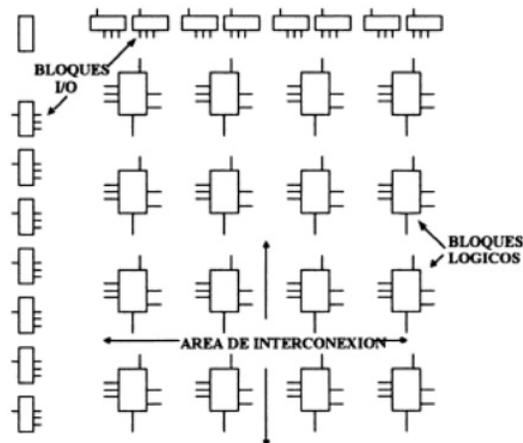


Figure 2.6: Arreglos lógica programable [7]

La programación se lleva a cabo por medio de unas matrices de células de memoria cuya función es la de controlar las uniones que se producen al interior del Chip. La célula de memoria estática se emplea para la configuración por su alta resistencia al ruido y es habilitada por el transistor de paso que incluye. De esta manera cuando la memoria va a leer solamente permite esta función y que no haya escrituras [6].

Los Bloques Lógicos configurables suministran los elementos para que los usuarios construyan los distintos diseños. Cada Bloque contiene una sección de lógica combinatorial, un elemento de almacenaje, un sistema de control y un enrutamiento interno. Posee también múltiples entradas de propósito general

y una entrada especial para el reloj. Además, contiene varias salidas que se enlazan al sistema de interconexión [6].

## 2.3 Memoria RAM

La Memoria RAM es un elemento que almacena información. Su nombre se debe a la denominación en inglés ‘Random Access Memory’ o ‘Memoria de Acceso Aleatorio’, cuya principal característica es la de poder suministrar la información allí almacenada más rápidamente que los dispositivos de almacenamiento de masa, como los discos rígidos, pero que, al contrario de éstos, la información es volátil porque al desconectar la memoria de su fuente de alimentación se pierden los datos [8,9].

El Almacenamiento se realiza en bloques de ocho bits que se denominan bytes. En la mayoría de los casos la información se guarda en múltiplos de ocho bit; por ejemplo, 2 bytes corresponden a 16 bites. Los bloques de bytes se llaman palabras y la capacidad de memoria se establece en términos de los bytes que puede guardar. [10]

Para comunicarse con el exterior, la memoria cuenta con una serie de conectores, a saber: un conector, conformado por múltiples hilos, para la entrada de los datos a almacenar; un conector por donde van a salir los datos, con múltiples hilos también; otro, con la respectiva dirección para guardar o leer los datos; un control de lectura y uno más de escritura. [10]

En la escritura específica se efectúa una transferencia de afuera hacia dentro de la memoria, mientras que en la lectura específica es de dentro hacia afuera. Los pasos a seguir en el proceso de transmitir la información hacia la memoria son: definir la dirección de la memoria que será afectada; aplicar la información en las líneas de entrada; activar el control de escritura. Y para leer, se define la dirección hacia la cual se quiere acceder y se habilita el control de lectura. [10]

Las Memorias RAM son de varios tipos, entre los cuales figuran: FPM DRAM o Fast Page Mode DRAM; EDO RAM o Salida Extendida de Datos; SDRAM o DRAM de Enlace Síncrono; DDRAM; SRAM o Static RAM; SDRAM Synchronous Dynamic RAM. [11]

La Memoria tipo SRAM permite el almacenamiento directo de datos, el acceso a cada posición con el seleccionador y guardar la información por tiempo

indefinido en tanto se mantenga la alimentación, lo que se consigue en virtud del biestable construido con mofest. Esta memoria consta de tres estados: reposo, lectura y escritura. [9]

La Memoria tipo SDRAM trabaja de manera sincronizada al reloj del sistema y se adapta a esta acción para no tener tiempos de [8], proceso que se consigue al momento de trabajar por ráfagas de lectura y escritura haciendo uso de direcciones similares o iguales [9]. Este tipo de Memoria, a pesar de tener mayor capacidad de almacenamiento, no dispone de un sistema de autorrefrescamiento con la consecuente pérdida de datos o su no legibilidad y la falla del sistema que se ejecuta.

## 2.4 Osciladores

El oscilador es un dispositivo mediante el cual una señal DC se transforma en una de tipo AD, con una frecuencia determinada. [12]

Por lo general, todo oscilador está conformado por los siguientes componentes: fuente de oscilación, amplificador y una retroalimentación. Con relación al primero de ellos se cuenta con varias opciones tales como circuitos RLC, cristales y circuitos integrados. [12]

Los Circuitos RLC son de tipo pasivo y la oscilación se produce en el momento de un cambio en la entrada o sucede una realimentación. Por tanto, solamente disponen de condensadores, bobinas y resistencias. Los cristales, generalmente son de cuarzo y tienen una determinada frecuencia que depende de la forma en que sean cortados. Finalmente, existen circuitos integrados que generan la señal AC con una frecuencia determinada. [12]

En algunos casos, a estos circuitos se les adiciona una etapa mediante la cual se recorta la onda y se reduce su amplitud con el fin de acoplarla a la entrada TTL o CMOS de algunos circuitos en los que debe ser de 3.3 a 5 voltios, sin la parte negativa, y poderse aplicar, por ejemplo, en los relojes de los sistemas digitales.



# Chapter 3

## AMBIENTE DE DESARROLLO

### 3.1 Ambiente de desarrollo FPGA Spartan 3 y Spartan 3E

La descripción del hardware del proyecto se elaboró en el ambiente de la Aplicación denominada ISE web junto con la compilación y la implementación en el Kit de desarrollo empleado. La interfaz de ISE web y la de la programación del Kit de desarrollo corresponden a las Figuras 3.1 y Figura 3.2.

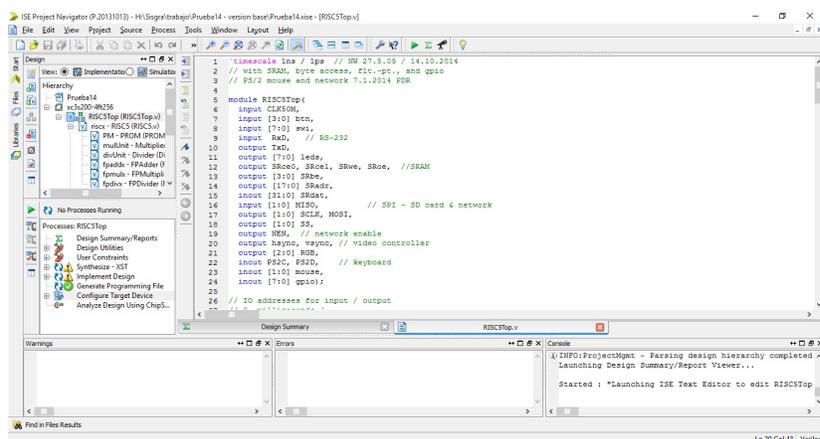


Figure 3.1: Interfaz de ISE web

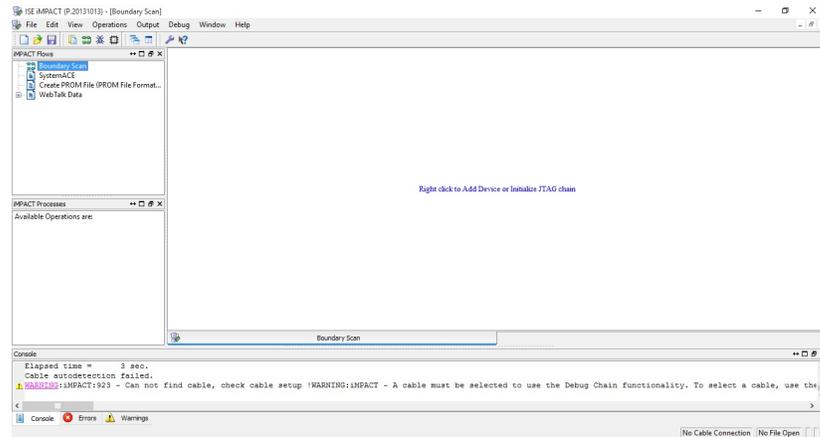


Figure 3.2: Interfaz de programación

## 3.2 Ambiente de desarrollo diseño y construcción de hardware

El diseño de las PCB se procesa dentro del ambiente de la Aplicación Kicad que facilita la construcción de su esquema o mapa; a continuación, a los componentes se les asignan los encapsulados que se usarán. Por último, el cableado en el que se definen las pistas de cobre y su implementación.

La Interfaz de Kicad se desarrolla según el esquema de cableado, empaquetamiento y diseño de PCB que se muestra en las Figuras 3.3, 3.4, 3.5 y 3.6:

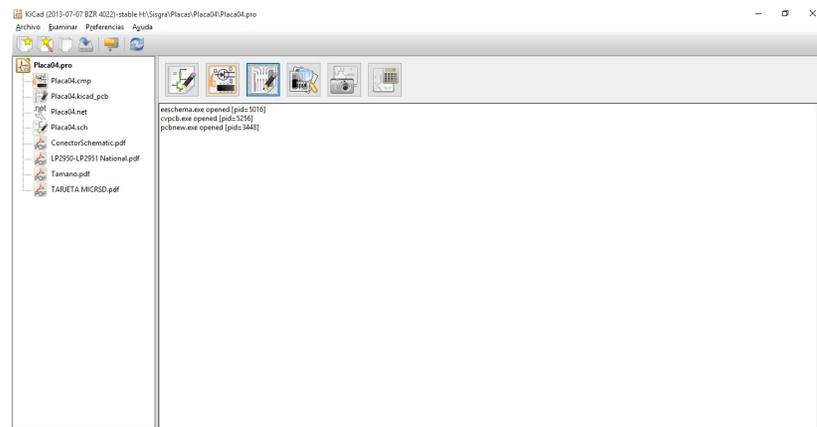


Figure 3.3: Interfaz de Kicat

### 3.2. AMBIENTE DE DESARROLLO DISEÑO Y CONSTRUCCIÓN DE HARDWARE17

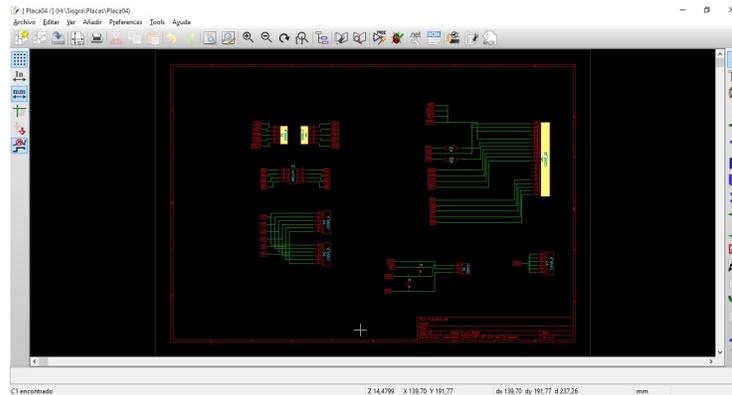


Figure 3.4: Interfaz de Kicad:esquema del cableado

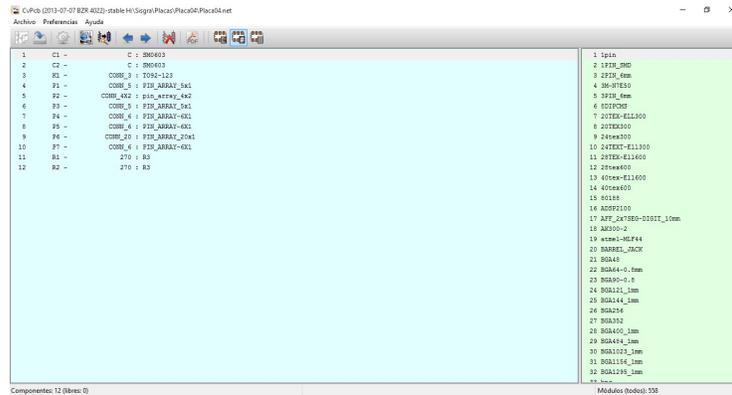


Figure 3.5: Interfaz de Kicad:empaquetamiento

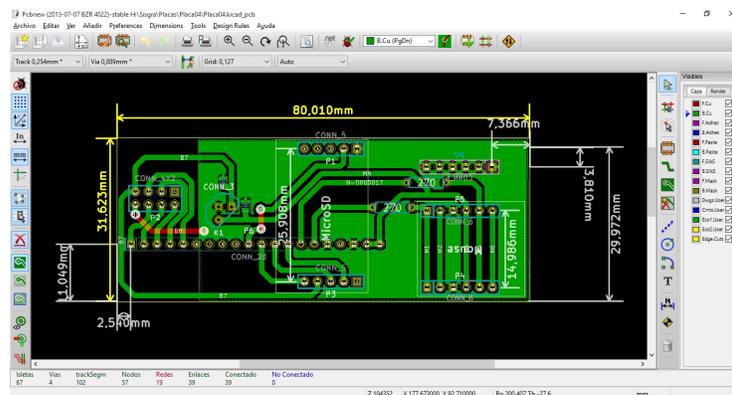


Figure 3.6: Interfaz de Kicad:diseño de PCB

### 3.3 Ambiente de simulación de hardware

La Aplicación Multisim va a permitir las simulaciones del nuevo hardware de elementos físicos como resistencias, condensadores, bobinas, transistores y circuitos operacionales, entre otros, al tiempo que da lugar a la validación del hardware y la observación de las posibles fallas que pueden ocurrir durante la implementación de los circuitos.

El esquema siguiente, Figura 3.7, corresponde a la Interfaz Multisim:

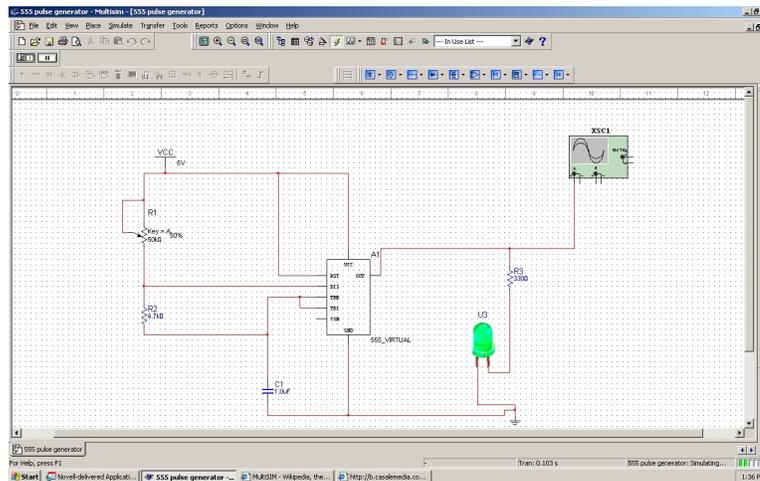


Figure 3.7: Interfaz de Multisim

### 3.4 Herramientas utilizadas

La naturaleza del Proyecto exige el empleo de varias clases de herramientas que a continuación se mencionan:

1. Kit Spartan 3E.
2. 3. Kicad: para el desarrollo de PCB. En ella se comienza la definición del circuito desde el esquemático hasta llegar al diseño de la tarjeta.
3. 4. ISE Design Suite 14.7: permite el diseño de los circuitos que se implementarán en la Spartan 3E para pasar el programa sin necesidad de otra aplicación.
4. Multisim: Simulación de hardware.

# Chapter 4

## RESULTADOS

Dentro de los resultados obtenidos del proyecto es preciso mencionar el diseño y construcción de los PCB que integran los componentes necesarios para el uso de la FPGA Spartan 3 y los correspondientes PCB para la conexión de la FPGA Spartan 3E. Estos son los diseños e implementaciones:

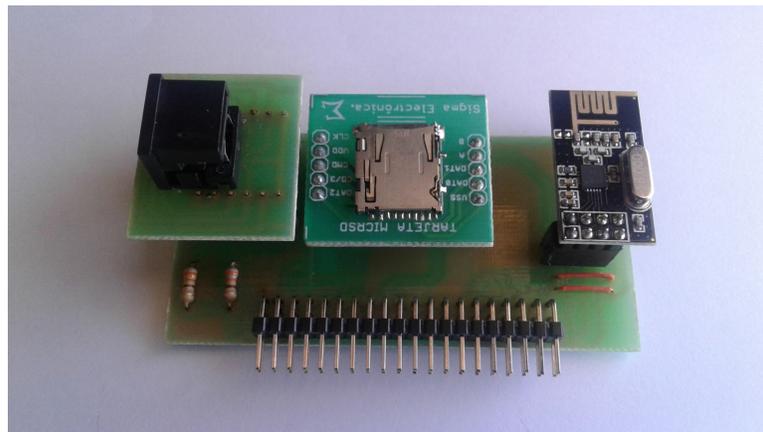


Figure 4.1: Conexión periféricos Spartan 3

La Figura 4.1 ilustra la conexión del mouse, la SDcard y el transmisor bluetooth a la Spartan 3. Después de verificar el correcto funcionamiento del sistema operativo en su entorno nativo, se crearon dos PCB con el objeto de conectar el circuito impreso en la Spartan 3 a la Spartan 3E, como se ilustra en la Figura 4.2.

Durante el proceso de invención de la interfaz del sistema operativo, junto con la memoria, se analizaron dos posibilidades: la primera, extraer completamente el módulo de memoria y colocar el nuevo controlador, pero con el



Figure 4.2: Conexión periféricos Spartan 3E

inconveniente de que el nivel de acoplamiento entre los módulos del sistema operativo y la memoria es muy alto. La segunda, colocar un módulo externo para el manejo de la memoria a causa de las nuevas especificaciones que se presentan al cambiar la forma de direccionamiento y la necesidad de un circuito de autorrefresco.

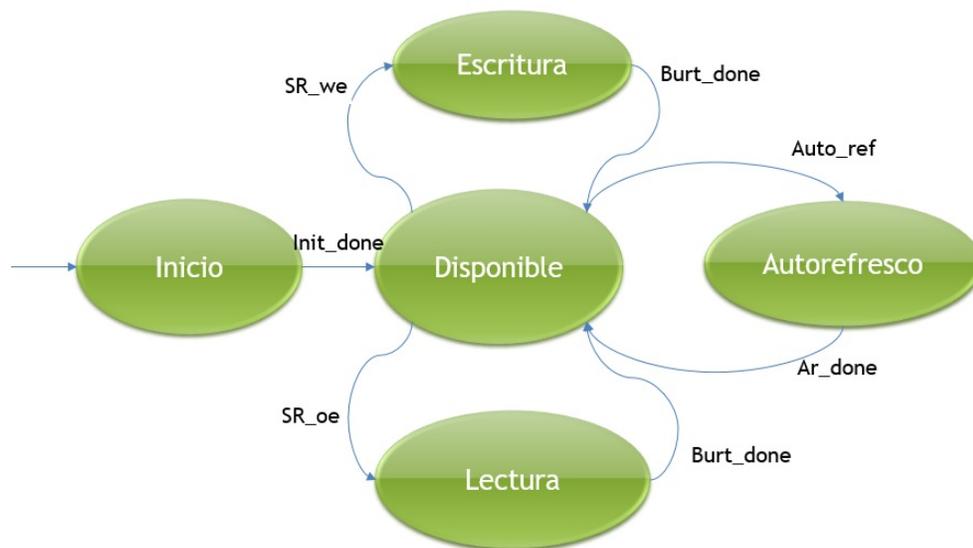


Figure 4.3: Máquina de estados

Este módulo es una Máquina de Estados que controla el reloj del sistema operativo e integrar las nuevas especificaciones, a saber: inicialización, disponible,

lectura, escritura y autorrefresco. En el estado de inicialización se debe esperar a que la memoria finalice esta etapa para poder iniciar el sistema operativo y continuar al disponible. En este estado el sistema puede realizar las peticiones y verificar el Circuito de Autorrefresco, dando prioridad a esta operación.

Cuando no hay necesidad de refrescar la memoria, el sistema operativo puede leer y escribir pero solamente después de que le sea ratificada la finalización de la operación, razón por la cual se congela el reloj del sistema.

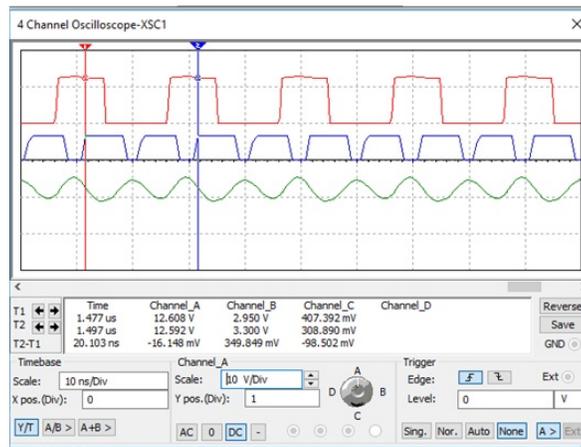


Figure 4.4: Señales de la simulación de 100 MHz

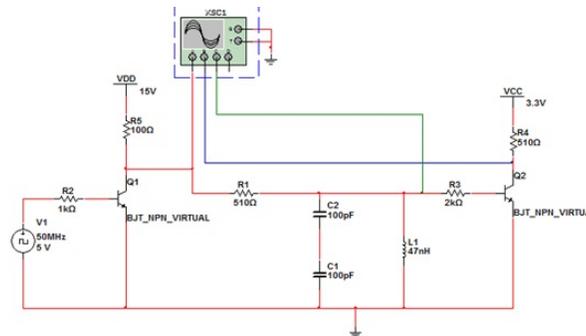


Figure 4.5: Circuito del filtro y transistor para la señal de 100 MHz

Al probar la memoria que está en el Kit de Desarrollo de la Spartan 3E se advierte que el oscilador indispensable para el uso de la memoria no se encuentra disponible porque la frecuencia debe estar entre 100 MHz y 133MHz, mientras que el contenido en el Kit es de 50MHz. Para resolver este problema

es preciso construir un oscilador externo.

Se examinaron, entonces, estas opciones: un oscilador de chip y otro por descomposición de la señal. Para el segundo se tomó la señal cuadrada y se pasó a través de un filtro de 100MHz con el propósito de obtener el armónico de la señal. Sin embargo, después de utilizar un transistor para convertir la señal de AC a señal cuadrada, la amplitud del armónico adquirida no contaba con la suficiente potencia para la activación del transistor.

La simulación de la Figura 4.4 y el Circuito de la Figura 4.5 se percibe cómo se obtendría la frecuencia de 100MHz a partir de una señal cuadrada.

La primera opción, el oscilador de chip, es de tecnología LVPECL que no es reconocida en la entrada de la FPAG porque las compuertas son de tipo CMOS. Se necesita, entonces, un circuito de acople, (como el que a continuación aparece) el cual permite mover la tensión referencia de 2.5 a 0 voltios y cambiar la amplitud de la señal de 800mV a 3.3 voltios. Este circuito se encuentra sometido a pruebas.

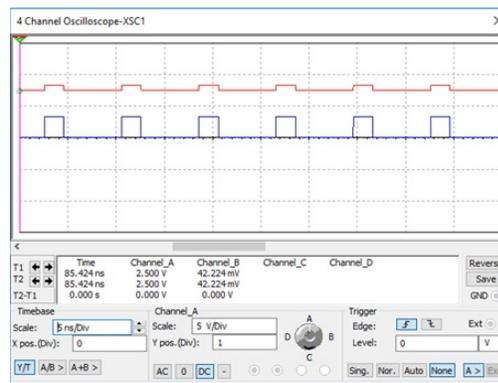


Figure 4.6: Señales de la simulación de 125 MHz

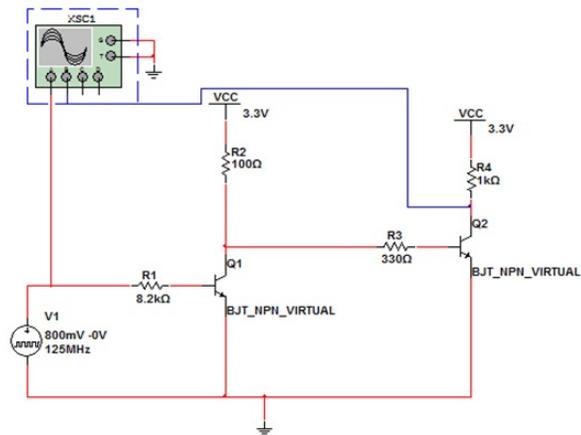


Figure 4.7: Circuito del filtro y transistor para la señal de 125 MHz



## Chapter 5

# CONCLUSIONES

En cuanto a los objetivos planteados se logró la construcción e implementación de las PCB para la conexión de los circuitos a la FPGA Spartan 3, en la cual se ejecutó el Sistema Operativo estando en hardware nativo, verificando el uso de los dispositivos que lo conformaban y su correcto funcionamiento.

Para la migración a FPGA Spartan 3E se logró la construcción de la PCB de acceso al puerto FX-100, luego, se diseñaron los demás impresos dando conexión al hardware externo, en el cual se verifico haciendo que la información fluyera entre el dispositivo y la FPGA.

Por otra parte, se diseñó un circuito de acople entre el sistema descrito para la operación del Sistema Operativo y la memoria física que se encuentra en el kit de desarrollo, en donde se implementó una máquina de estados que controla el reloj del software, al tiempo que maneja el circuito de autorefresco requerido en la operación de la memoria, evidenciando que la memoria necesita un oscilador mayor al que trae el kit de desarrollo.



## Chapter 6

# TRABAJOS FUTUROS

Debido al objetivo planteado para el proyecto, quedan algunas mejoras que se pueden realizar en cuanto al Sistema Operativo Oberon, para ampliar el campo de estudio, algunas son: el diseño de un segundo núcleo y la creación de un sistema de arbitramento para poder acceder a los dispositivos periféricos sin que hayan problemas de replicación de datos o conflictos de entrada.

Por otra parte comenzar a ampliar los dispositivos que pueden conectarse al Sistema Operativo y que están disponibles en el kit de desarrollo de la FPGA Spartan 3E como por ejemplo el puerto Ethernet.



# Bibliography

- [1] J. G. Niklaus Wirth, *Project Oberon The Design of an Operating System, a Compiler, and a Computer*. 2013.
- [2] J. G. Niklaus Wirth, *The Oberon System*. Institut für Informatik Fachgruppe Computer-System, 1988.
- [3] N. W. Martin Reiser, *Programming in Oberon*. ACM Press, 1992.
- [4] ethoberon, “Native oberon operating system,” 2003.
- [5] M. Mano and G. Sánchez, *Diseño digital*. Pearson Educación, 2003.
- [6] J. Repullo, M. Buendía, and A. Valdés, *FPGAs. Circuitos de lógica programable*. Manuales y materiales didácticos, Editum. Ediciones de la Universidad de Murcia, 1995.
- [7] J. A. V. Repullo, *Circuitos programables Pal*. Manuales y materiales didácticos, Editum., Ediciones de la Universidad de Murcia, 1995.
- [8] L. Rodríguez, *El Gran Libro del PC Interno*. EL GRAN LIBRO DE, Marcombo, 2007.
- [9] C. Viejo, *Fundamentos de electrónica digital*. Ediciones Paraninfo. S.A., 2005.
- [10] M. Mano and M. Ciletti, *Digital Design*. Prentice hall design series, Pearson Prentice-Hall, 2007.
- [11] L. Rodríguez, *Ampliar, configurar y reparar su PC*. Marcombo, 2006.
- [12] W. Tomasi, G. Hernández, and V. Pozo, *Sistemas de comunicaciones electrónicas*. Pearson Educación, 2003.