

DISEÑO DIGITAL CON FPGAS Y MICROPROCESADORES

Digital desig with fpgas and microprocessors

JOSÉ ALEJANDRO FRANCO CALDERÓN*, MYRIAM CATALINA GARZÓN JARAMILLO**

Recibido: 14 de mayo de 2015. Aceptado: 14 de junio de 2015

RESUMEN

El presente artículo describe el diseño y la implementación de un sistema que permite visualizar algunas secuencias y mensajes en una FPGA. El documento muestra el desarrollo de la lógica a implementar en código VHDL para un contador descendente de cuatro (4) dígitos de 9999 a 0000, un cronómetro ascendente que muestra el conteo de minutos y segundos (mm:ss) de 00:00 a 59:59, un mensaje de cuatro letras y un mensaje de al menos 10 caracteres los cuales se muestran de forma dinámica en los cuatro (4) displays siete segmentos de la tarjeta NEXYS2.

Palabras clave: FPGA, NEXYS2, lógica secuencial, lógica combinacional, visualización dinámica.

ABSTRACT

This article describes the design and implementation of a system that displays some sequences and messages on a FPGA. The paper shows the development of logic implemented in VHDL code for a down counter of four (4) digits in 9999-0000, showing an upward counting stopwatch minutes and seconds (mm: ss) 00:00 59 59, a message of four letters and a message of at least 10 characters which are displayed dynamically in the four (4) of the seven-segment displays NEXYS2 card.

Key words: FPGA, NEXYS 2 sequential logic, combinational logic, dynamic display.

I. INTRODUCCIÓN

Los objetivos del sistema están orientados a la visualización en los displays de la tarjeta NEXYS2 un contador descendente 9999 - 0000, un cronómetro ascendente que muestre los minutos y segundos mm:ss de 00:00 - 59:59, un mensaje de 4 letras y un mensaje de al menos 10 caracteres. Implementando el sistema de tal manera que permita al usuario escoger entre las siguientes frecuencias de conteo (1Hz, 2Hz, 10Hz, 20Hz, 100Hz, 200Hz, 1000Hz, 2000Hz) y las siguientes frecuencias de visualización (2Hz, 15Hz, 60Hz, 500Hz, 1KHz, 10KHz, 1MHz, 10MHz).

Adicionalmente se presenta la implementación en las secuencias la opción al usuario para el reseteo

tanto en las secuencias de conteo como en el reinicio del mensaje de texto mostrado.

II. MARCO TEÓRICO

A. Las FPGA's [1]

Las FPGAs (Field Programmable Gate Array) son dispositivos semiconductores que están en una matriz de bloques lógicos configurables (CLBs) conectados a través de interconexiones programables mediante un lenguaje de descripción especializado.

Las FPGAs pueden ser reprogramadas para la aplicación o funcionalidad requerida después de

* Ingeniero Electrónico de la Escuela Colombiana de Ingeniería "Julio Garavito", especialista en diseño de aplicaciones para televisión digital interactiva y en administración de tecnologías de la información para la comunicación virtual de la Universidad Manuela Beltrán, Estudiante de maestría en ingeniería electrónica en la Escuela Colombiana de Ingeniería Julio Garavito. Docente investigador de la facultad de ingeniería adscrito al Grupo de Investigación y Desarrollo en Ingeniería de Sistemas - GIDIS de la Corporación Universitaria Republicana. Correo electrónico: alejing@gmail.com

* Ingeniero Electrónico de la Escuela Colombiana de Ingeniería "Julio Garavito", Estudiante de maestría en ingeniería electrónica en la Escuela Colombiana de Ingeniería Julio Garavito. Correo electrónico: catalina.garzonj@gmail.com

ser fabricada, esta característica la distingue de circuitos integrados de aplicación específica (ASIC), que son fabricados para tareas específicas que no se pueden modificar.

En la lógica programable de las FPGAs se pueden reproducir circuitos tan sencillos como una compuerta lógica o un sistema combinacional o tan complejos como sistemas a implementar en un chip.

B. Lenguaje de descripción HDL [2]

Los lenguajes de descripción HDL son utilizados para detallar la estructura, el funcionamiento, la operación y las conexiones de circuitos electrónicos y/o digitales de alto nivel.

Los lenguajes HDL más utilizados son el VHDL y el Verilog, ambos permiten detallar el funcionamiento y las conexiones del circuito.

El VHDL (Very High speed Integrated Circuits Hardware Description Language) es un lenguaje portable y reusable ya que es independiente de la tecnología o fabricante en la que se implemente el mismo. Las sentencias programadas con este lenguaje no se ejecutan secuencialmente salvo aquellas que están dentro de procedimientos.

Las secciones fundamentales que forman el código VHDL son: Librería (LIBRARY), entidad (ENTITY) y arquitectura (ARCHITECTURE). LIBRARY, contiene el conjunto de librerías donde se halla la definición de datos, estructuras, etc; ENTITY, especifica las entradas y salidas del circuito y ARCHITECTURE, describe el circuito que se quiere implementar, para lo cual existe dos formas de hacerlo la estructural y la comportamental esta última a su vez se divide en flujo de datos y/o algorítmica.

Una ventaja al utilizar este lenguaje es que permite generar la implementación de test-bench para la simulación de diseños.

C. Sistemas combinacionales [3]

Los sistemas combinacionales están formados por un conjunto de compuertas interconectadas cuya salida depende únicamente del cambio en las entradas. Los sistemas combinacionales no tienen memoria. Para la construcción de estos sistemas se utilizan típicamente compuertas NAND o NOR.

D. Sistemas secuenciales

Los sistemas secuenciales son capaces de tener salidas no solo en función a las entradas actuales, sino que también de entradas o salidas anteriores. Los sistemas secuenciales tienen memoria y son capaces de almacenar información a través de sus estados internos.

Estos sistemas normalmente están compuestos por circuitos combinacionales y elementos de memoria como los Flip-Flop SR o tipo D.

E. Visualización dinámica [4]

La visualización dinámica consiste en mostrar porciones de información en distintos intervalos cortos de tiempo. En cada intervalo de tiempo la información es diferente consiguiendo así mostrar toda la información requerida.

El efecto de la visualización dinámica se da cuando el intervalo de tiempo en el que se muestra las porciones de la información es menor al tiempo en que la imagen dura retenida en la retina del ojo humano por lo cual el cerebro procesa la información nueva y la retenida como una sola; este efecto se da debido a lo que se ha denominado como tiempo de retención de la retina, el cual explica que una imagen dura pequeñas fracciones de segundo en la retina.

Gracias al tiempo de retención de la retina es posible proyectar una imagen obturándola a más de 24Hz (poniendo y quitando la imagen más de 24 veces por segundo) y ser percibida por el ojo humano como una imagen fija.

III. SIMULACIÓN

Con el fin de comprobar el diseño de la solución implementada para resolver los problemas planteados se generó una simulación por cada uno de los ejercicios y una del diseño completo.

Nota: Para las simulaciones no se tuvo en cuenta la variación de frecuencia para conteo, desplazamiento ni visualización.

A. Simulación para contador descendente

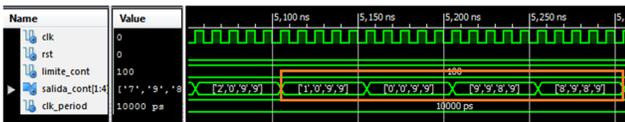


Figura 7. Decenas llegan a cero centenas disminuyen en una unidad.

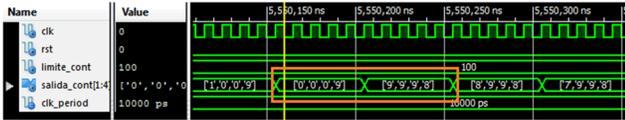


Figura 8. Centenas llegan a cero unidades de mil disminuyen en una unidad.



Figura 9. Unidades de mil llegan a cero se reinicia el conteo desde 9999.

Para comprobar el correcto funcionamiento del cronómetro ascendente se generó un archivo de prueba «Cronometro_tb», se sintetizó y ejecutó.

En la simulación se esperaba que: a la salida del bloque se obtuviera como primer número 00:00 y fuera aumentando de a una unidad en los segundos y luego los minutos, además que la misma fuera en caracteres y mostrara el número correcto, esto se pudo verificar en la simulación como se muestra en la Fig.10. También se esperaba que al recibir el impulso del Rst comenzara de nuevo el cronómetro en 00:00, y al recibir el impulso del Rst2 comenzara el cronómetro a contar desde 59:59 esto también se verificó con la simulación como se muestra en la Fig.11 y Fig. 12. Adicionalmente se verificó que al llegar a 09s aumentará a 10 seg (Fig. 13), que cuando el cronómetro mostrara 59 seg el cronómetro aumentara en 01:00 min (Fig.14), que al llegar a 09:59 min pasara a 10:00 min (Fig. 15) y finalmente que cuando el cronómetro llegara a 59:59 (min : seg) comenzara de nuevo el conteo desde 00:00 (Fig. 16).

C. Simulación para mostrar palabra «tdda».

Para comprobar que la palabra tdda se mostrará y se desplazará correctamente se generó un archivo de prueba «Palabra_tb», se sintetizó y ejecutó.

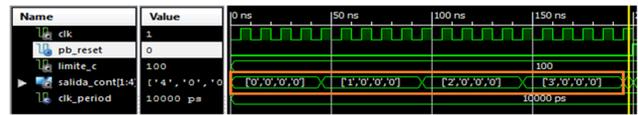


Figura 10. Cronómetro inicia en 00:00 y aumenta de a un seg, el conteo se obtiene en caracteres.

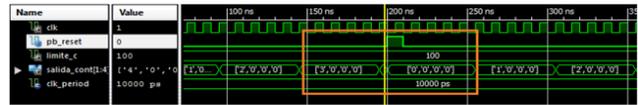


Figura 11. Cronómetro inicia en 00:00 al recibir el impulso de Rst.

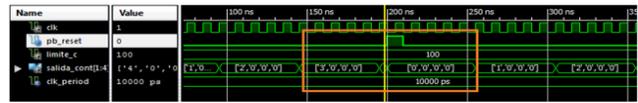


Figura 12. Cronómetro inicia en 59:59 al recibir el impulso del Rst2.

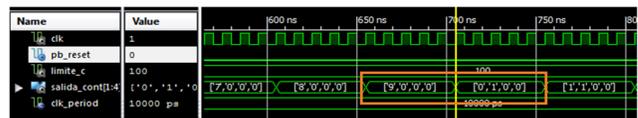


Figura 13. Cronómetro pasa de 00:09 a 00:10 (mm:ss).

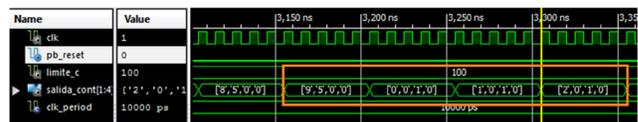


Figura 14. Cronómetro pasa de 00:59 a 01:00 (mm:ss).

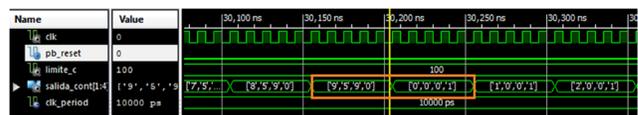


Figura 15. Cronómetro pasa de 09:59 a 10:00 (mm:ss).

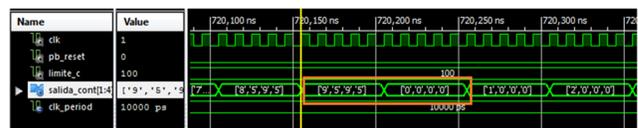


Figura 16. Cronómetro pasa de 59:59 a 00:00 (mm:ss).

Al ejecutar el programa el simulador arrojó lo que se observa en la Fig.17, en la cual se puede ver que la simulación se termina antes del tiempo programado 1000ns, no se realiza el desplazamiento en la palabra y todas las variables quedan en sus valores iniciales, por esto, se revisó el código desarrollado y se encontró con que se estaba evaluando la cadena de la palabra en una posición que no existía, se corrigió el error y se simuló de nuevo.

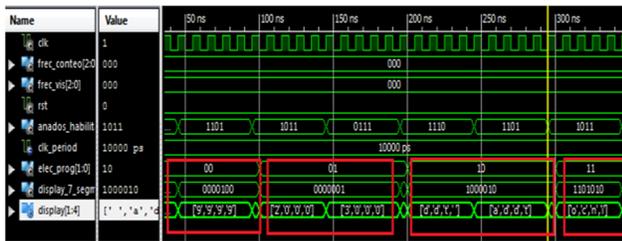


Figura 22. El selector Elec_Prog cambia el programa según se esperaba y se indica en la tabla 1.

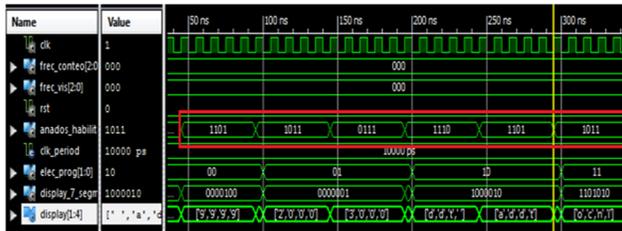


Figura 23. Habilitador de los displays alternados cada cuatro (4) impulsos de tiempo

observar que efectivamente el habilitador se alterna cada cuatro (4) pulsos del reloj.

En la Fig. 24, se puede observar que el decodificador está trabajando adecuadamente, codificando cada letra enviada en in_deco salida del multiplexor el cual genera la variación de la información en intervalos de tiempo cortos con el fin de generar la visualización dinámica.

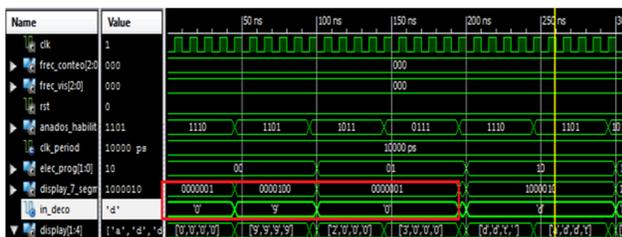


Figura 24. Decodificador de caracteres funcionando adecuadamente.

IV. CÁLCULOS TEÓRICOS

La siguiente formula permite realizar el cálculo de la cantidad de pulsos de reloj que deben contarse para obtener la frecuencia deseada en la conmutación de los displays con el fin de generar la visualización dinámica del sistema, las frecuencias de visualización programadas se muestran en la Tabla 2.

Tabla 2. Frecuencias de visualización dinámica.

Frecuencia (Hz) por cada display	Pulsos conteo
2	6.250.000
15	833.333
60	208.333
2000	25.000
4.000	12.500
40.000	1.250
4.000.000	12,5
40.000.000	1,2

Se parte del hecho de que la tarjeta NEXYS2 maneja un reloj interno a una frecuencia de 50MHz y que cada frecuencia deseada se debe multiplicar por el número de displays en el que se requiere mostrar el mensaje en este caso 4.

$$Pulsos\ visualización = \frac{Frecuencia\ cristal}{Frecuencia\ deseada * 4}$$

Por ejemplo para una Frecuencia de 2Hz se tiene que:

$$Pulsos\ conteo = \frac{50.000.000\ Hz}{2\ Hz * 4}$$

$$Pulsos\ conteo = 6.250.000$$

Para hallar los pulsos de conteo se aplicó la siguiente fórmula:

$$Pulsos\ conteo = \frac{Frecuencia\ cristal}{Frecuencia\ deseada}$$

Por ejemplo para una Frecuencia de 2Hz se tiene que:

$$Pulsos\ conteo = \frac{50.000.000\ Hz}{2\ Hz}$$

$$Pulsos\ conteo = 25.000.000$$

Haciendo uso de la anterior ecuación se hicieron las conversiones a las diferentes frecuencias programadas, los resultados se encuentran en la Tabla 3.

Tabla 3. Frecuencias de conteo.

Frecuencia (Hz)	Pulsos conteo
1	50.000.000
2	25.000.000
10	5.000.000
20	2.500.000
100	500.000
200	250.000
1.000	50.000
2.000	25.000

V. DESARROLLO

El desarrollo integral del presente diseño presupone la sintetización modular de código descrito en VHDL de 4 componentes claves para el desarrollo del mismo, unido con un módulo principal encargado de manejar la multiplexación, dando la libertad al usuario de seleccionar la frecuencia de visualización, de conteo o desplazamiento y/o el componente que requiera observar en los displays, adicionalmente el módulo principal contiene la visualización dinámica en común para todos los componentes. Bajo este contexto se comenzó haciendo el análisis de los primeros componentes que se debían sintetizar en la FPGA, estos son:

1. Contador descendente 9999 a 0000.
2. Cronómetro ascendente 00:00 a 59:59.
3. Palabra «tdda».
4. Mensaje de más de 10 letras.

Los bloques funcionales de dichos componentes los podemos observar en las siguientes figuras:

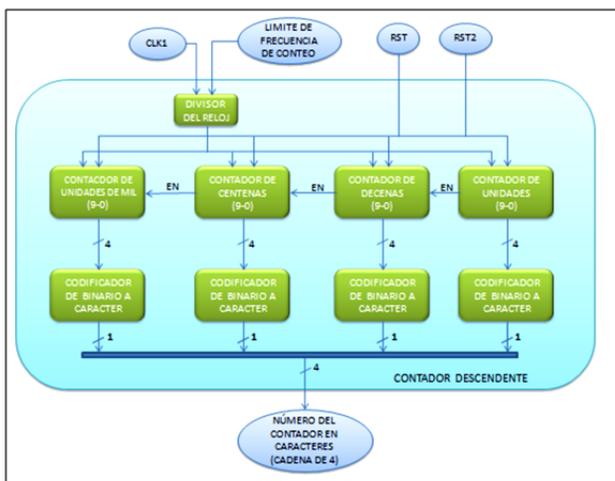


Figura 25. Componente contador descendente.

Como se puede observar en la Fig. 25 para el componente de conteo descendente, se tiene un proceso que se encarga de ir decrementando los contadores correspondientes a las unidades, las decenas, las centenas y las unidades de mil, los cuatro contadores están conectados en cascada para generar el decremento de la cifra cada vez que el contador predecesor llegue a cero (0) excepto en el contador de unidades donde el decremento de la cifra está dada por la velocidad de conteo, esta última se generó teniendo en cuenta el límite de velocidad de conteo asignado por el usuario con el cual se hace el divisor de la frecuencia del reloj principal. La salida que se obtiene de este componente es una cadena de caracteres generada por cuatro codificadores que toman el valor arrojado por cada uno de los contadores y los convierte en carácter asignado a una posición específica de la cadena.

En la Fig. 26 se muestra el proceso que incrementa las unidades de los segundos, las decenas de los segundos, las unidades de los minutos y las decenas de los minutos de cuatro contadores conectados en cascada. En el proceso los contadores aumentan de la siguiente manera: Las decenas aumentan una unidad cada vez que los contadores de unidades ya sea de segundos o minutos lleguen a nueve (9), las unidades de minutos aumenta cada vez que las decenas de segundos lleguen a cinco (5) y finalmente las unidades de segundo aumentan con cada pulso del reloj que llega con una frecuencia configurada según lo indique el límite de conteo asignado por el usuario. A la salida del componente se tiene una cadena de cuatro (4) caracteres compuesta en cada posición

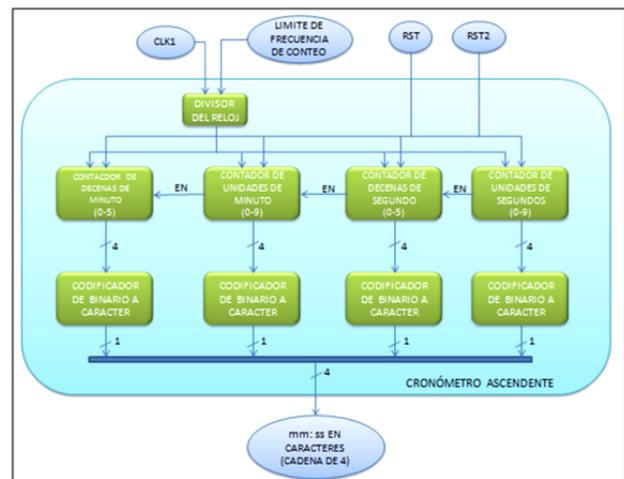


Figura 26. Componente cronómetro ascendente.

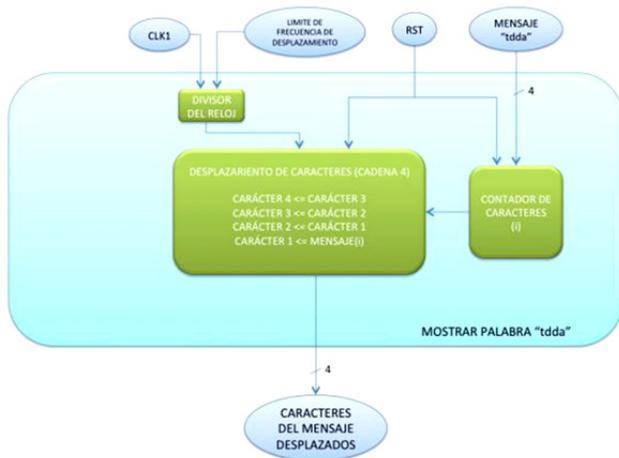


Figura 27. Componente mostrar palabra «tdda».

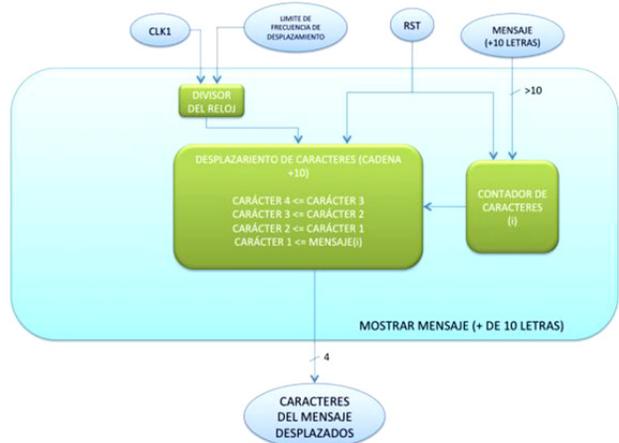


Figura 28. Componente mostrar mensaje de más de 10 letras.

por la codificación del número que tiene la salida actual de cada contador.

Por último en las Fig. 3 y 4 se muestra los procesos que se encargan de ir desplazando los caracteres de una palabra determinada, a la velocidad correspondiente según la división del reloj dada por el límite asignado por el usuario. A la salida de este proceso se tiene una cadena de cuatro (4) caracteres con las letras desplazadas.

Cada componente fue sintetizado como un módulo independiente en VHDL, de tal manera que nos permitiera conectarlo posteriormente con el módulo principal, este último contiene la caracterización de todas las entradas del sistema, adicionalmente la multiplexación según la elección del usuario, el sistema de visualización dinámica y la codificación a los 7 segmentos de los display.

Al realizar el análisis de entradas y salidas del sistema se tiene que:

Tabla 4. Descripción de entradas y salidas del sistema.

Pin puerto	Descripción
Clk	(in) Reloj del sistema
Rst	(in) Reset del sistema hacia el inicio de la secuencia
Rst2	(in) Reset del sistema hacia el final de la secuencia
Elec_Prog[1 a 0]	(in) Switches selector de programa.
Frec_conteo[2 a 0]	(in) Switches selector de frecuencias de conteo.
Frec_vis[2 a 0]	(in) Switches selector de frecuencias de visualización.
Display_7Segmentos[6 a 0]	(out) Vector de salida de los displays 7 segmentos.
Anodos_Habilitador[3 a 0]	(out) Vector de habilitadores de ánodo común para los displays.

El sistema completo se puede observar en la siguiente figura:

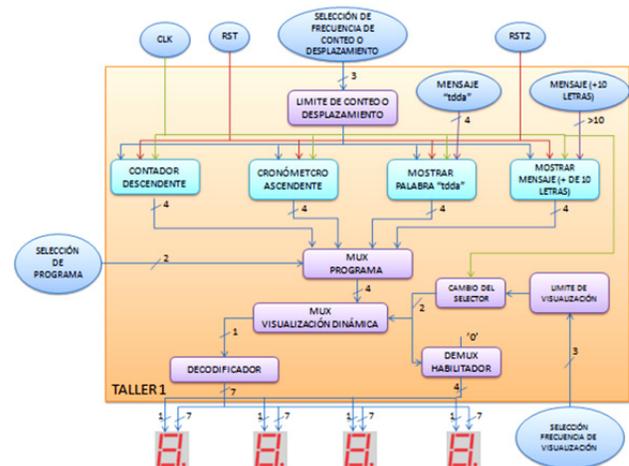


Figura 29. Sistema completo integrado, relación entre entradas, salidas e integración con otros componentes individuales.

Mediante la técnica de diseño estructural en VHDL [5] se procede a sintetizar en la FPGA todos los módulos descritos anteriormente conectan-

do entradas y salidas de los componentes con ayuda del mapa de puertos (Fig. 30).

```

NET "clk" LOC="B8";
NET "Rst" LOC="B18";
NET "Rst2" LOC="D18";

NET "Display_7_Segmentos[0]" LOC="H14";
NET "Display_7_Segmentos[1]" LOC="J17";
NET "Display_7_Segmentos[2]" LOC="G14";
NET "Display_7_Segmentos[3]" LOC="D16";
NET "Display_7_Segmentos[4]" LOC="D17";
NET "Display_7_Segmentos[5]" LOC="F18";
NET "Display_7_Segmentos[6]" LOC="L18";

NET "Anados_Habilitador[3]" LOC="F15";
NET "Anados_Habilitador[2]" LOC="C18";
NET "Anados_Habilitador[1]" LOC="H17";
NET "Anados_Habilitador[0]" LOC="F17";

NET "Frec_conteo[2]" LOC="K18";
NET "Frec_conteo[1]" LOC="H18";
NET "Frec_conteo[0]" LOC="G18";

NET "Frec_vis[2]" LOC="R17";
NET "Frec_vis[1]" LOC="N17";
NET "Frec_vis[0]" LOC="L13";

NET "Elec_Prog[1]" LOC="L14";
NET "Elec_Prog[0]" LOC="K17";
    
```

Figura 30. Asignación de pines FPGA.

Observando el manual de la tarjeta NEXYS2 [6] se puede hacer la siguiente asignación de pines acorde a nuestro sistema caracterizado en la FPGA.

Para efectos de pruebas sobre la FPGA se describe la combinación de switches de entrada del sistema con su correspondiente función dentro del mismo.

Tabla 5. Frecuencias de conteo según switches de la FPGA.

Frecuencias de conteo			Selección
K18	H18	G18	
0	0	0	1Hz
0	0	1	2Hz
0	1	0	10Hz
0	1	1	20Hz
1	0	0	100Hz
1	0	1	200Hz
1	1	0	1000Hz
1	1	1	2000Hz

Tabla 6. Frecuencias de visualización según switches de la FPGA.

Frecuencias de visualización			Selección
R	17	N17	L13
0	0	0	2Hz
0	0	1	15Hz
0	1	0	60Hz
0	1	1	500Hz
1	0	0	1KHz
1	0	1	10KHz
1	1	0	1MHz
1	1	1	10MHz

Tabla 7. Switches de selección de funcionalidad en la FPGA.

Elección de programa		Selección
R	17	L13
0	0	Contador
0	1	Cronómetro
1	0	Mensaje «tdda»
1	1	Palabra + 10

VI. ANÁLISIS DE RESULTADOS

A. Análisis de frecuencia de visualización

En las frecuencias bajas de 0Hz a 15Hz la visualización dinámica no se presenta debido a que la velocidad de conmutación es menor a la velocidad de retención de la pupila.

En la frecuencia de 15Hz los caracteres en los displays son legibles pero hay un parpadeo que incomoda la visión humana, generando mareo.

A partir de los 60Hz los caracteres dejan de parpadear, son legibles pero al movimiento mecánico de la FPGA presenta un comportamiento similar al analizado en la frecuencia de 15Hz.

En el rango de frecuencias de 500Hz hasta 1kHz se tiene una visualización óptima sin parpadeo. El cambio más representativo que se da al moverse de la frecuencia menor a la frecuencia mayor en este rango es el aumento en la intensidad de la iluminación en los displays.

En frecuencias altas como 1MHz y 10MHz la visualización se torna tan rápida que se hace ilegible para el ojo humano.

B. Análisis de frecuencia de conteo y desplazamiento

A partir de frecuencias de desplazamiento de 10Hz la visualización de mensajes se dificulta debido a la velocidad con que los caracteres se trasladan de un display a otro.

Para las secuencias de conteo o cronómetro el cambio más significativo ocurre a la frecuencia más alta en la cual, los dígitos menos significativos se tornan ilegibles al ojo humano debido a la velocidad de incremento o decremento según sea el caso.

VII. CONCLUSIONES

Debido a la versatilidad del lenguaje VHDL se logró hacer conversiones entre enteros y caracteres de una manera fácil, con lo cual se pudo diseñar un decodificador 7 segmentos general para los cuatro módulos del programa.

Se logró comprobar diferentes estados del sistema mediante la simulación generada en archivos test-bench, disminuyendo el tiempo de pruebas del sistema diseñado.

Durante la práctica se generaron 4 desarrollos independientes para cada componente lo cual no era el objetivo final del laboratorio, debido a esto, se tuvo que rediseñar el sistema integrando los cuatro módulos, lo que permitió evidenciar la versatilidad del lenguaje VHDL y la utilidad del diseño por componentes.

Se logró diseñar un contador descendente 9999 - 0000 que se puede visualizar en los displays de la tarjeta NEXYS2.

Se logró diseñar un sistema que permite visualizar en los displays de la tarjeta NEXYS2 un

cronómetro ascendente que muestre los minutos y segundos mm:ss de 00:00 - 59:59.

Se logró diseñar un sistema que permite visualizar en los displays de la tarjeta NEXYS2 un mensaje con la palabra «tdda».

Se logró diseñar un sistema que permite visualizar en los display de la tarjeta NEXYS2 un mensaje de al menos 10 caracteres.

Se logró integrar al sistema completo un arreglo de switches de tal manera que se le permite al usuario escoger entre las siguientes frecuencias de conteo (1Hz, 2Hz, 10Hz, 20Hz, 100Hz, 200Hz, 1000Hz, 2000Hz).

Se logró integrar al sistema completo un botón cuya función consiste en resetear tanto las secuencias de conteo enviando los contadores al valor inicial o final.

Se logró integrar al sistema completo un botón cuya función consiste en mostrar el mensaje desde la primera letra cada vez que sea oprimido.

REFERENCIAS

- [1] XILINXS Inc., «Field Programmable Gate Array (FPGA),» *XILINXS*, 2015.
- [2] S. Noriega, «Introducción al diseño lógico con VHDL,» *VHDL*, 2010.
- [3] R. Araya, «Sistemas Combinacionales y Sistemas Secuenciales.» 2006.
- [4] R. Martínez, «ITT - 327- P-074. Reporte Práctica #5, T1. Visualizador dinámico. 2012.
- [5] J. Soto, «Elementos de diseño estructural.» 2015.
- [6] Digilent, «Digilent NEXYS2 Board Reference Manual,» *Digilent*, 2011.