CAPÍTULO II . SISTEMA DE CONTROL Y GENERADOR DE PWM

2.1. Introducción.

Los rápidos cambios de tecnología en el área de la instrumentación han aumentado dramáticamente la necesidad de diseño de sistemas más sofisticados que permitan la calibración y verificación del equipo, de manera que el ambiente industrial siga avanzando. Los sistemas digitales han llegado a ser la solución y están teniendo un fuerte impacto en el campo de la medición. Especialmente los llamados dispositivos lógicos programables que han venido a revolucionar en diversas áreas, incluyendo la instrumentación.

La utilización de estos dispositivos, entre ellos uno de los más avanzados el FPGA, trae grandes ventajas ya que se caracterizan por facilidad en uso, flexibilidad para adaptarse a diversas aplicaciones y permiten el diseño de equipos con un tamaño más reducido, todo ello a un costo relativamente bajo por el hecho de ser programables.

Para el diseño del sistema controlador y generador de PWM se hizo uso de estos dispositivos tomando en cuenta las ventajas de su uso. El lenguaje de programación usado fue VHDL, ya que es un lenguaje sencillo que permite describir el hardware.

2.2. Introducción al VHDL.

Actualmente nos encontramos rodeados de sistemas electrónicos muy sofisticados, como son los teléfonos celulares, equipos de sonido y computadoras portátiles. Estos ejemplos y otros más, demuestran el crecimiento en el desarrollo tecnológico, el cual ha cambiado nuestras vidas haciéndolas más confortables.

Su tamaño de dimensiones tan pequeñas es lo que caracteriza a estos sistemas, los cuales son incluso más potentes que los que existían hace algunos años atrás. Esto es posible, gracias a la evolución de la tecnología y el diseño de la microelectrónica que ha permitido la realización de sistemas electrónicos digitales complejos en un único circuito integrado de escala de integración progresivamente elevada. Que va desde los circuitos de integración de baja escala (SSI o Small Scale Integration), los de mediana escala (MSI o Médium Scale Integration), los de muy alta escala (VLSI o Very Large Scale Integration,), hasta los de propósito específico ASIC (Application Specific Integrated Circuit) [2].

Actualmente la tendencia en el diseño de circuitos para aplicaciones específicas se basa en la utilización de celdas programables preestablecidas dentro del circuito integrado. Con base en lo anterior, surgen los dispositivos lógicos programables (PLD) cuyo nivel de densidad de integración se ha venido incrementando al paso del tiempo. Desde los Arreglos Lógicos Programables (PAL) hasta los Dispositivos Lógicos Programables Complejos (CPLD) y Arreglos de Compuertas Programables en Campo (FPGA), que han dado como resultado mayor facilidad en el desarrollo de estos circuitos y una disminución en su costo de fabricación.

Para este tipo de diseño de aplicaciones existe un lenguaje de programación que es considerado una de las mejores herramientas para el diseño de sistemas dentro de la industria. El lenguaje de descripción de hardware VHDL que da la posibilidad de integrar aplicaciones digitales de forma fácil y práctica.

2.2.1. Definición y Antecedentes Históricos.

VHDL (Hardware Description Language) es un lenguaje de descripción de hardware orientado al modelado de sistemas digitales. Pertenece a un conjunto de lenguajes especializados en la descripción de componentes de hardware que son la alternativa para

el diseño mediante esquemáticos pero cuentan con las mismas funcionalidades que los lenguajes convencionales de programación.

Este lenguaje permite modelar, verificar y simular sistemas digitales, a través de la simulación por eventos y la metodología *top-down*. Además, los diseños realizados pueden ser traducidos a una implementación real lo que se denomina *sintetizar*. Es decir, la transformación de la descripción en VHDL a los componentes lógicos necesarios para realizar un sistema real.

El lenguaje VHDL nació a principios de los años 80 en el departamento de defensa de los Estados Unidos y fue desarrollado a partir del lenguaje ADA. Más tarde fue adoptado como un estándar por la IEEE (Institute of Electrical and Electronics Engineers) para el año de 1987 (IEEE 1076-87). Seis años después fue revisada y aprobada una nueva versión del estándar con la incorporación de nuevas características y la definición de nuevas librerías, convirtiéndose así en el estándar IEEE 1076-93 [4].

2.2.2. Estructura básica de Diseño.

En general, un diseño VHDL se compone de un conjunto de bloques o módulos, donde cada uno de ellos contiene declaraciones o instrucciones que describen y estructuran el comportamiento del sistema. Dentro de cada módulo se declaran diferentes unidades de diseño como son:

- Entidad
- Arquitectura
- Configuración
- Paquete
- Biblioteca

Entre las principales unidades de diseño están:

Entidad - Es el elemento básico del lenguaje VHDL que define el nombre de un componente y su interfaz de entrada-salida. La interfaz es definida por un conjunto de puertos, mientras que la implementación queda oculta al resto del circuito, como un modelo de *caja negra.* No pueden existir dos entidades con el mismo nombre dentro del diseño. Y su funcionalidad es definida mediante la arquitectura.

Arquitectura – Describe el funcionamiento de la entidad a la que está asociada. Una misma entidad puede contener una o más arquitecturas. Se compone de dos partes: la declarativa donde se incluyen las señales, variables y componentes que se va a emplear, y el cuerpo donde se incluye la implementación de la entidad por medio de instrucciones o sentencias.

En general, dentro de la arquitectura en VHDL la descripción del modelo se puede realizar utilizando los siguientes estilos de programación:

- **Funcional (Behavioral) :** describe el algoritmo que refleja la funcionalidad o el comportamiento de dicho componente de manera secuencial.
- **Flujo de Datos (Data Flow) :** especifica los flujos de datos del sistema y la interconexión entre sus componentes. Es decir, la forma en que los datos se transfieren de una señal a otra. Y se realiza por medio de funciones lógicas, que se ejecutan de forma concurrente.
- **Estructural (Structural) :** esta descripción se basa en modelos lógicos ya establecidos como restadores, sumadores, compuertas, entre otros. Define componentes e interconexiones y resulta muy útil para aprovechar diseños ya compilados y guardados en bibliotecas de componentes para un uso posterior.

Dentro del diseño de VHDL un sistema puede utilizar varios estilos de programación y mezclarlos al describir sus diferentes componentes.

2.2.3…Ventajas de su Uso.

Existen varias ventajas de la utilización de VHDL sobre otros lenguajes de programación entre ellas están:

- Portabilidad del diseño entre diferentes herramientas.
- Mayor facilidad en la verificación y detección de fallos por medio de la compilación y simulación.
- Lenguaje poderoso y actual en la descripción de hardware.
- Lenguaje con semántica orientada a la simulación.
- Estándar reconocido por los fabricantes de dispositivos digitales.
- Permite la utilización de diferentes estilos de descripción de eventos ya sea de manera secuencial o concurrente.
- Permite mezclar en una misma descripción diferentes niveles de abstracción.
- Permite la reutilización de submódulos.
- Lenguaje que permite ser sintetizado en un circuito digital.

Todas estas ventajas hacen de VHDL un lenguaje poderoso y actual, basado en tecnologías contemporáneas. Capaz de ampliar las posibilidades en aplicaciones de sistemas digitales con mayor complejidad.

Esta herramienta es utilizada en la industria en todo el mundo y es la base en el uso de dispositivos lógicos programables. Entre ellos los FPGAs dispositivos muy importantes en el desarrollo de sistemas digitales.

2.3. Introducción al FPGA.

Cuando se habla del diseño de un sistema electrónico siempre surge la necesidad de implementar una parte con hardware. Es ahí cuando las variadas posibilidades se reducen al analizar la relación de costo, flexibilidad, prestaciones y complejidad. Los dispositivos lógicos programables son una excelente oferta para realizar diseños electrónicos digitales mejorando la relación costo-prestaciones.

Por otro lado, reducen asombrosamente el tiempo de implementación del diseño. Haciendo que el costo en la realización del sistema sea muy bajo, por lo que son una buena opción en el diseño de prototipos. De hecho, en la última década han hecho posible que el diseño de hardware sea casi tan flexible y conveniente como el diseño en software.

Entre los dispositivos lógicos programables más recientes y avanzados están los FPGAs (Field Programmable Gate Array) , circuitos con componentes microelectrónicos de última generación que permiten la capacidad de 5000 a 10 millones de compuertas en un solo chip. Estos dispositivos son un arreglo bidimensional de bloques lógicos reprogramables colocados con interconexiones programables, por lo que hacen posible programar la función de los bloques lógicos y las interconexiones entre los mismos incluyendo las entradas y salidas. Esta característica abre la posibilidad de implementar múltiples diseños de sistemas físicos mediante FPGAs.

Además, estos dispositivos tienen la facilidad de ser programados por el usuario. Para lograrlo, el usuario utiliza como herramienta los lenguajes de descripción de hardware, entre los que se encuentran VHDL, Verilog y Handel-C.

2.3.1. Arquitectura básica y funcionamiento del FPGA.

Cuando se habla de la fabricación de dispositivos lógicos programables es indispensable hacer mención de dos conceptos básicos:

- **Funcionalidad completa** que se refiere al hecho de que cualquier función lógica se puede realizar mediante una suma de productos; y
- **Celdas de Funciones Universales** también denominadas como generadores de funciones, que son bloques lógicos configurados para procesar cualquier función lógica, siendo similares en su funcionamiento a una memoria. Estas celdas almacenan los datos de salida del circuito combinacional en vez de implementar físicamente la ecuación booleana [3].

Basados en los conceptos anteriores se puede decir que la arquitectura de un FPGA consiste en arreglos bidimensionales de varias celdas lógicas que se comunican mediante canales de interconexión verticales y horizontales. De hecho, en un FPGA se utilizan Celdas de Funciones Universales en vez de compuertas, como lo hace un CPLD. Cada una de estas celdas son como una memoria en donde se precalcula el resultado y se almacena en la misma. Sus entradas funcionan como un bus de direcciones, y por medio de las diferentes combinaciones el generador selecciona el resultado correcto. Lo anterior reduce el tiempo de propagación comparado con el uso de compuertas lógicas.

Además, un FPGA contiene gran cantidad de celdas lógicas lo que hace posible la implementación de grandes funciones con la utilización de varias celdas lógicas interconectadas en forma de cascada. Esto hace factible la realización de sistemas amplios y complejos dentro del dispositivo.

Por otro lado, la cantidad de interconexiones dentro del dispositivo programable puede variar junto con el número de entradas y salidas. Por lo tanto, las tecnologías utilizadas en la creación de las conexiones varían desde ANTIFUSE utilizada por Ciprés, Actel, Xilinx y QuickLogic; hasta SRAM utilizada por Altera, Lucent Technologies, Atmel, Xilinx y otros.

La estructura básica del FPGA junto con todos sus componentes mencionados anteriormente se muestran en la figura 2.1. Gracias a esta arquitectura los FPGA proporcionan grandes ventajas en el ámbito industrial, además de promover un desarrollo tecnológico futuro.

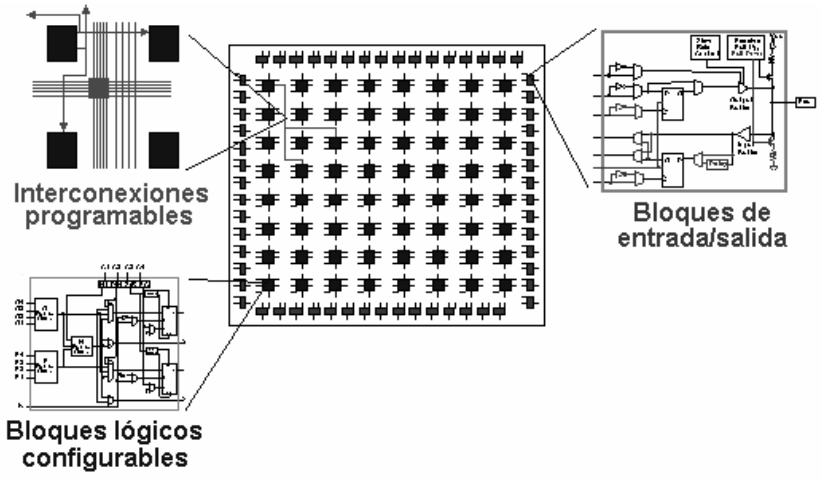

Figura 2.1 Arquitectura básica del FPGA [2].

2.3.2. Ventajas en su Uso.

Entre sus principales ventajas de uso están:

- Flexibilidad en la adaptación a diversas aplicaciones.
- Facilidad de uso.
- Tiempo mínimo de puesta en el mercado.
- Costo relativamente bajo.
- Reducción en el tiempo de diseño.
- Reconfiguración dinámica.
- Desarrollo de sistemas complejos en un solo chip.

Estas ventajas han hecho de los FPGAs una excelente herramienta para el diseño e implementación de sistemas digitales dentro del área industrial. Y no solo dentro de esta área, sino que también han revolucionado dentro de diversas áreas de

investigación y desarrollo como son las redes neuronales, algoritmos genéticos y otras más.

2.4. Descripción del Sistema Controlador y Generador de PWM.

El sistema se conforma de una unidad de control capaz de procesar ciertos valores que vienen del computador y generar una señal PWM, equivalente a los valores de una señal senoidal previamente muestreada y almacenada en una ROM. Este sistema genera dos señales senoidales, las cuales pueden ser manipuladas en su amplitud, frecuencia y fase de acuerdo con los datos programados por el usuario desde el computador.

El diseño de esta unidad de control y del generador de PWM se dividió en dos etapas:

1) Diseño de software mediante la herramienta de lenguaje de programación VHDL.
2) Implementación y sintetización del sistema mediante el dispositivo lógico programable FPGA.

2.4.1. Diseño del Sistema en VHDL.

El sistema controlador y generador de PWM se diseño valiéndose de la poderosa herramienta para diseñar sistemas digitales, VHDL. Las razones principales por la que se eligió esta tecnología en lenguajes de programación son las siguientes:

- Es un lenguaje que permite integrar sistemas digitales en forma fácil y práctica.
- Permite la implementación de sistemas reales mediante la sintetización.
- Es la tendencia actual y futura en el diseño e implementación de aplicaciones en circuitos digitales.

- Abarata los costos en el tiempo de diseño y la implementación del mismo.
- Conjunta los beneficios de los lenguajes de programación comunes con el beneficio de la utilización de esquemáticos para facilitar su diseño.
- Permite la división del diseño del sistema complejo en varios módulos con mucho menor complejidad de programación. De modo que se pueda dar una jerarquía al sistema.
- Brinda la posibilidad de la reutilización de módulos previamente desarrollados y compilados.
- Permite modelar, simular y sintetizar circuitos digitales.
- Es un lenguaje reconocido en todo el mundo como un estándar de la IEEE.
- Ayuda en la localización de fallas durante la compilación y simulación.
- Permite la simulación de eventos concurrentes además de los secuenciales.

Para la creación de cada módulo se hizo uso de algunas unidades de diseño en VHDL. Entre ellas, librerías como la IEEE y la work, además de paquetes como std_logic_1164, numeric_std, std_logic_arith, std_logic_signed los cuales facilitaron y agilizaron el diseño ya que permiten tener acceso a estructuras lógicas predeterminadas ya por el fabricante.

En conjunto con lo anterior, se declararon en cada componente la entidad respectiva y su arquitectura. A su vez, dentro de su arquitectura se utilizó la declaración de procesos, que equivale a una programación secuencial. Es decir, se hizo una descripción funcional utilizando el estilo de programación de comportamiento o behavioral.

Para generar la aplicación deseada, se crearon 20 diferentes componentes o módulos. De los cuales, 7 forman el programa principal y los restantes se encuentran como submódulos. La figura 2.2 muestra el diagrama de los 7 bloques principales que conforman el sistema programado en VHDL.

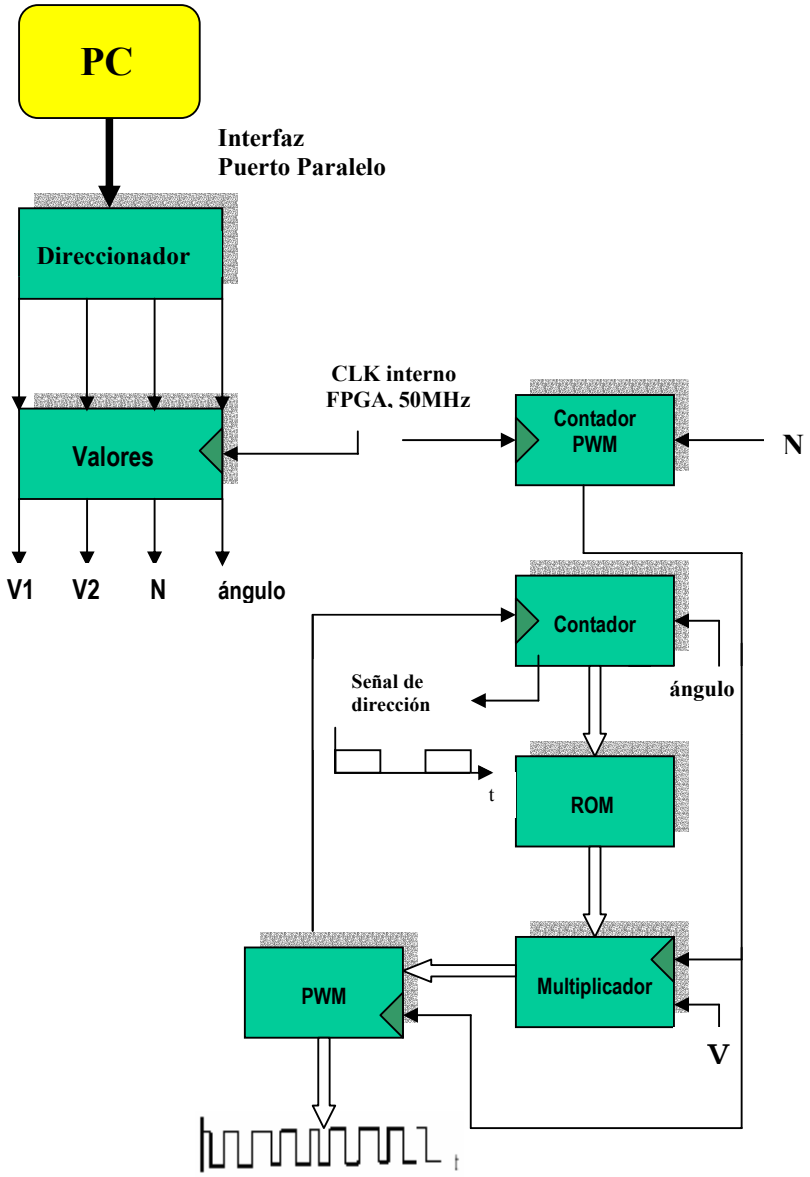

Figura 2.2. Diseño general del Sistema Controlador y Generador de PWM

De acuerdo a la descripción del circuito simulado en el capítulo I (ver pag.5), la etapa de generación de PWM requiere de dos señales que se comparan, una señal senoidal rectificada y una rampa. Dentro del diseño mostrado en la figura 2.2, la primera señal es equivalente al muestreo (50 muestras) de un semiciclo de señal senoidal almacenado en una ROM, mientras que la señal de rampa equivale al contador interno en el módulo de PWM cuya cuenta es de 0 a 100. Además, la comparación entre

ambas señales se realiza dentro del módulo de PWM mediante dos comparadores, uno para el ciclo de trabajo y otro para el periodo.

Por otro lado, los valores desplegados por la ROM equivalen al ciclo de trabajo de la señal PWM, por lo que el ciclo de trabajo varía en proporción a la amplitud de la señal senoidal. Y esta amplitud puede ser modificada al multiplicar cada valor de la ROM por un mismo factor (**V**). De igual manera, la fase (mediante la constante **ángulo**) y la frecuencia (mediante el índice **N**) pueden ser manipuladas, lo que muestra una ventaja sobre el circuito mostrado en capítulo anterior (figura 1.3.a.).

Ahora bien, la descripción paso a paso de este sistema es la siguiente:

Por medio de la interfaz del puerto paralelo de la computadora al FPGA se envían los valores de voltaje (**V1** y **V2**), el índice de frecuencia (**N**) y el ángulo de desfasamiento (**ángulo**), parámetros para generar dos señales de salida deseadas. Estos parámetros son recibidos por el módulo *Direccionador* programado en VHDL específicamente para la comunicación con la interfaz del puerto paralelo. Este módulo se describirá con más detalle en el Capítulo V donde se detalla el diseño y programación de la interfaz.

El bloque *Valores* carga todos los parámetros y los despliega hacia los módulos en los que se utilizarán. El *ContadorPWM*, que esta conectado a la señal del reloj de 50 MHz, recibe la constante de frecuencia (**N**) y realiza la función de un divisor de frecuencia que genera una señal de reloj a la frecuencia con la que funcionará el módulo de *PWM*. A su vez, el módulo de *PWM* genera internamente una señal de reloj que se conecta a los demás módulos para la sincronización del sistema. Por otro lado, el *Contador* , al recibir la señal de reloj del módulo *PWM*, incrementa su salida de 0 a 50, desplegando así los valores de las 50 muestras del semiciclo de la señal senoidal, que se encuentran almacenadas en la *ROM*. Estos valores de las muestras pasan al siguiente módulo (*Multiplicador*) donde son multiplicados por un valor equivalente a la amplitud

(**V**, correspondiente a V1 para la señal 1 y V2 para la señal 2) deseada de la señal. Finalmente el producto, que corresponde al ciclo de trabajo de la señal de PWM, llega al módulo *PWM* donde el periodo es constante a un valor de 100.

Todos los bloques ya mencionados corresponden a la generación de PWM de la primera señal. En el caso de la segunda señal generada, se reutilizan los mismos bloques interconectados igual que en la primera señal. La diferencia es que el *Contador* puede ser cargado al inicio con el valor del ángulo de desfasamiento, para generar así un cambio en la fase de esta señal con respecto a la otra.

En lo que corresponde específicamente a los valores de muestreo que se encuentran almacenados en la ROM, éstos fueron calculados previamente. Y la representación matemática que se utilizó para obtener los valores de las muestras es la siguiente:

$$f(x) = \left[Sen\left(\frac{\pi x}{50} \right) \right](100) + 0.5 \qquad (2.1)$$

Donde X corresponde al número de muestra para el que se desea calcular la función. El valor de 50 es el número total de muestras y 100 es el valor máximo de amplitud correspondiente al valor del periodo de la señal de PWM.

Esta ecuación (2.1) permite obtener el muestro completo de un semiciclo de señal senoidal por medio de 50 muestras. Además, limita el periodo de la señal rectificada a un valor numérico de 100.

La herramienta que se utilizó para calcular las 50 muestras fue Excel, programa permitió realizar los cálculos de manera rápida y exacta. Estos cálculos realizados se muestran en la Tabla 2.1.

X	B = *X* *3.1416/50	C =Sen (B)	=INT(C *100+0.5)
0	0	0	0
1	0.062832	0.06279067	6
2	0.125664	0.12533353	13
3	0.188496	0.18738175	19
4	0.251328	0.24869046	25
5	0.31416	0.30901769	31
6	0.376992	0.36812537	37
7	0.439824	0.42578022	43
8	0.502656	0.4817547	48
9	0.565488	0.53582791	54
10	0.62832	0.58778644	59
11	0.691152	0.63742524	64
12	0.753984	0.68454839	68
13	0.816816	0.72896993	73
14	0.879648	0.77051455	77
15	0.94248	0.80901829	81
16	1.005312	0.84432919	84
17	1.068144	0.87630788	88
18	1.130976	0.90482818	90
19	1.193808	0.92977751	93
20	1.25664	0.95105742	95
21	1.319472	0.96858393	97
22	1.382304	0.98228786	98
23	1.445136	0.99211512	99
24	1.507968	0.99802695	100
25	1.5708	1	100
26	1.633632	0.99802649	100
27	1.696464	0.9921142	99
28	1.759296	0.98228648	98
29	1.822128	0.9685821	97
30	1.88496	0.95105515	95
31	1.947792	0.92977481	93
32	2.010624	0.90482505	90
33	2.073456	0.87630434	88
34	2.136288	0.84432525	84
35	2.19912	0.80901397	81
36	2.261952	0.77050987	77
37	2.324784	0.72896491	73
38	2.387616	0.68454304	68
39	2.450448	0.63741957	64
40	2.51328	0.5877805	59
41	2.576112	0.53582171	54
42	2.638944	0.48174827	48
43	2.701776	0.42577357	43
44	2.764608	0.36811854	37
45	2.82744	0.30901071	31
46	2.890272	0.24868334	25
47	2.953104	0.18737453	19
48	3.015936	0.12532624	13
49	3.078768	0.06278333	6
50	3.1416	-7.3464E-06	0

Tabla 2.1. Tabla de valores de muestreo

Estos valores de muestras son los que se ingresaron en la ROM interna al FPGA correspondientes al ciclo de trabajo del PWM. Claro esta que para modificar la amplitud de la señal fueron multiplicadas, por una valor V que el usuario asigna desde la computadora, antes de ingresarse al módulo de PWM.

Ahora bien, dentro del bloque *ContadorPWM* podemos encontrar otros submódulos. Como lo muestra la figura 2.3, el primer submódulo recibe la constante N que determina el usuario para dividir la frecuencia de entrada. Los siguientes dos submódulos generan el reloj que sincroniza el módulo de *PWM* .

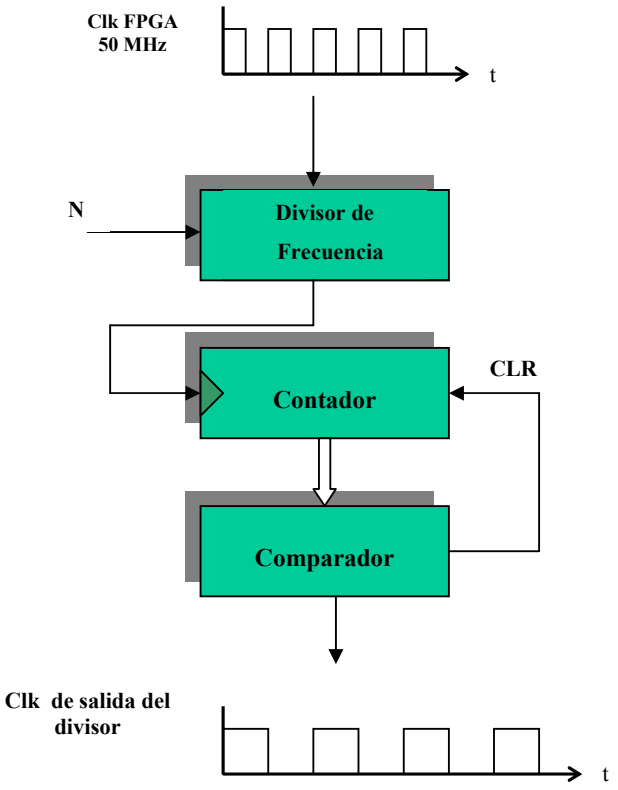

Figura 2.3 Diagrama de la estructura del módulo ContadorPWM

Por otro lado, el módulo de *PWM* también fue creado a partir de varios submódulos. En la figura 2.4, que a continuación se describe, podemos observar los módulos programados para generar la señal en PWM .

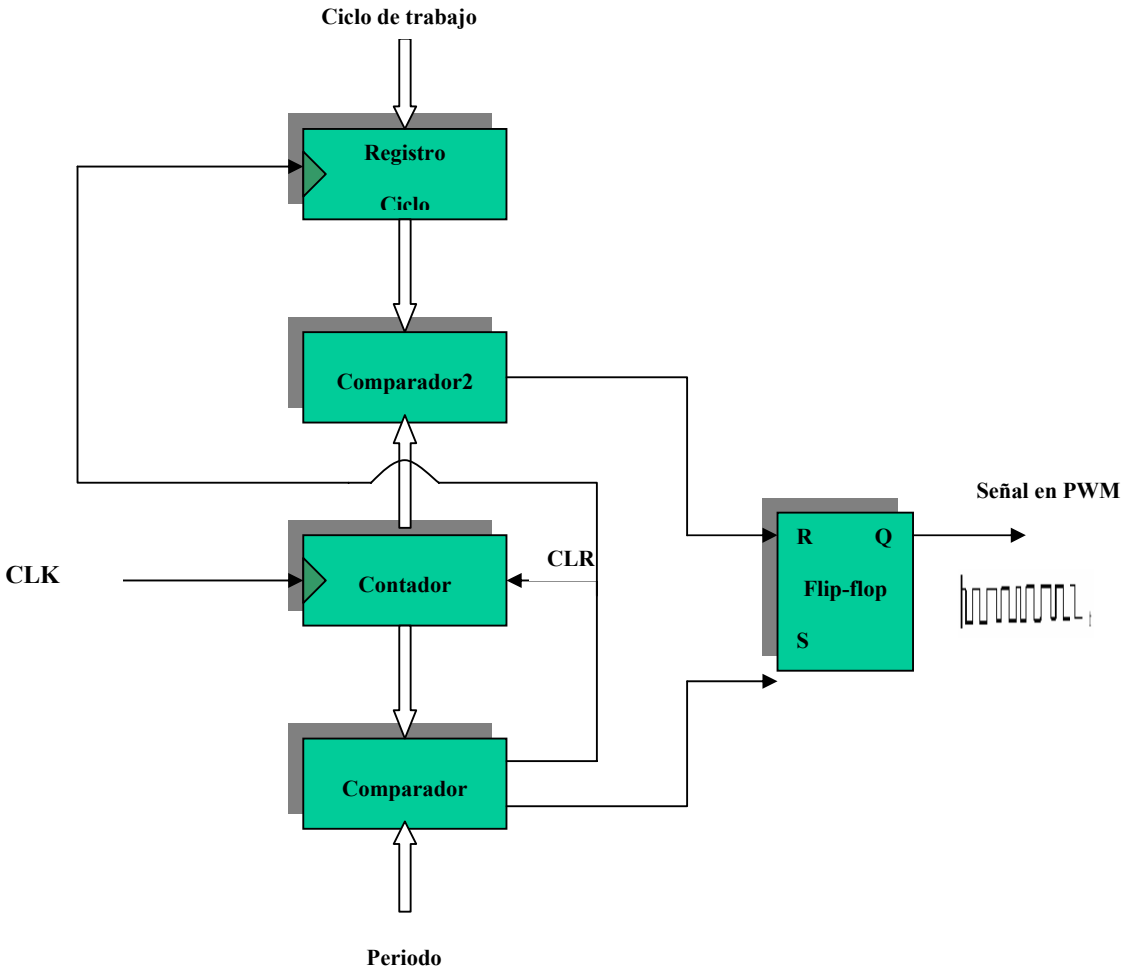

Figura 2.4 Diagrama de los submódulos del PWM

Como entrada al módulo PWM se tiene el valor correspondiente al ciclo de trabajo que varía de acuerdo a la amplitud de la señal senoidal rectificada. Por otro lado, el bloque *Contador* incrementa su cuenta de uno en uno empezando de 1 hasta 100, similar a la generación de una señal rampa con un periodo de valor 100. La

comparación de estas dos señales, por así decirlo, se realiza mediante los dos bloques comparadores y el flip-flop , dando como salida una señal modulada en PWM.

Finalmente, a la salida del sistema se puede observar una señal con modulación en PWM de una señal senoidal rectificada, como se encuentra en la figura 2.5. En la figura 2.5.(a) se muestra la señal senoidal con rectificación completa que corresponde a la señal muestreada y en la parte (b) tenemos la señal de salida modulada ya en PWM.

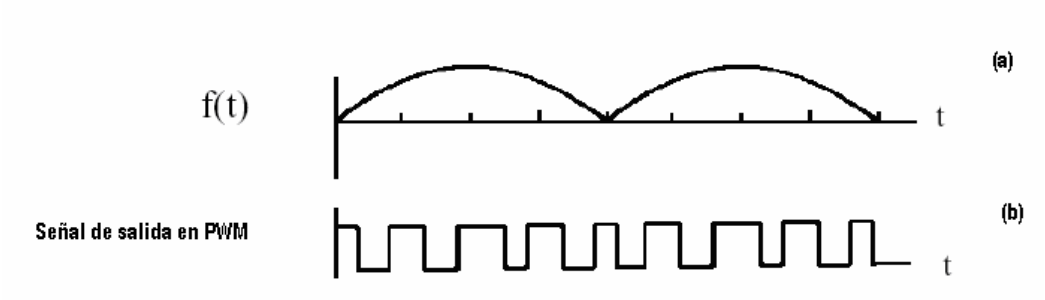

Figura 2.5 Señal senoidal rectificada modulada en PWM

Para llevar a cabo la implementación del diseño, ya descrito en programación VHDL se necesita sintetizarlo. Para realizar esta tarea se hizo uso del FPGA, un dispositivo lógico programable especializado en aplicaciones para sistemas digitales.

2.4.2 Implementación en el FPGA

Cuando se trata del diseño de sistemas digitales para la industria, no basta con crear un software que simule las aplicaciones, es necesario traducir ese software a un circuito real, es decir, a un hardware. El FPGA al ser un dispositivo lógico programable permite la implementación o sintetización del sistema previamente diseñado en software. Por esta razón, fue utilizado para la implementación de sistema de control y generador de PWM.

Aún cuando existen otros dispositivos programables que podían ser usados en la implementación del sistema como los microcontroladores. Sin embargo, se decidió la elección del FPGA porque es una tecnología actual, muy poderosa en la síntesis de circuitos. Además, un solo dispositivo tiene mayor capacidad para expandir el proyecto, si se desea en un futuro. Mientras que los microcontroladores tienen mayores limitaciones por el número de puertos de entradas y salidas. Además en un microcontrolador no es posible generar el periodo de la señal, ya que su programación es secuencial, por lo que tarda mayor tiempo en ejecutarse, lo cual no permitiría tener un número grande de muestras. En cambio, la ventaja del FPGA es que su programación es concurrente permitiendo mayor velocidad al desplegar el muestreo, por lo que se tiene la capacidad de tener mayor número de muestras, dando como resultado una reconstrucción más exacta de una señal senoidal.

Por otro lado, a diferencia de los microcontroladores, la programación de los FPGAs es más sencilla ya que permite la descripción del hardware. De manera que el sistema se puede dividir en módulos, mismos que se usarían si se implementara directamente. Esto hace posible, que el tiempo de diseño e implementación se reduzca considerablemente, comparado con el tiempo que llevaría al usar un microcontrolador o cualquier otro dispositivo.

Existen en la industria diferentes fabricantes de FPGAs como son Altera, Lucent Technologies, Atmel, Xilinx y otros. En este proyecto se utilizó la familia del fabricante Xilinx que corresponde al Spartan2 XC2S200 que es programado con lenguajes de descripción de hardware, cuyo lenguaje de simulación es VHDL.

Para hacer uso de este FPGA se utilizó la tarjeta **Digilent 2SB** que se muestra a continuación en la figura 2.6.

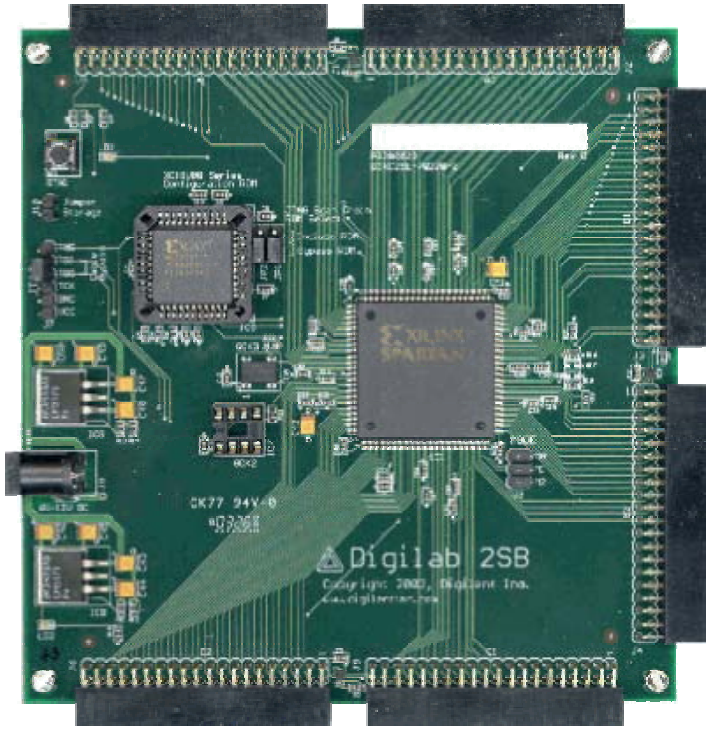

Figura 2.6. Tarjeta D2SB de Xilinx [8]

Cuyas características se pueden encontrar en el anexo bajo el nombre Apéndice A, así como una lista del mapeo de los pines utilizados como salidas y entradas del sistema.

El diagrama general de la estructura interna funcional de esta tarjeta se puede analizar mediante la figura 2.7.

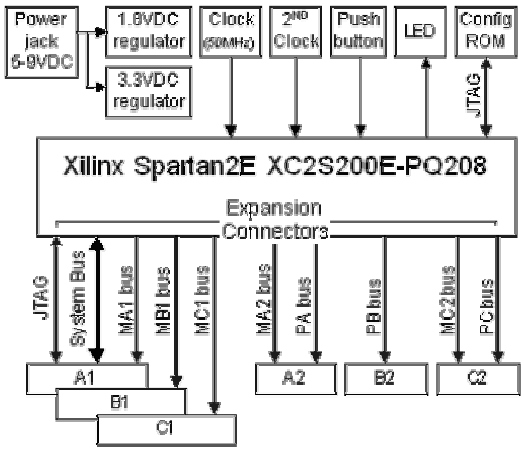

Figura 2.7. Diagrama funcional de la tarjeta D2SB [8]

La tarjeta contiene un reloj que trabaja a la frecuencia de 50 MHz. Este reloj fue utilizado en el diseño del sistema como la señal de sincronización entre los diferentes módulos, a través de un divisor de frecuencia.

Por otro lado, la tarjeta utilizada tiene 143 pines de entradas/salidas de las cuales se utilizaron 4 pines como salidas y 9 como entradas. Entre los pines de salida están 2 pines de las señales de control de dirección del puente H y 2 pines de las dos señales de PWM generadas. Entre las señales de entrada tenemos 9 pines destinados a la interfaz paralela con la computadora , de estos 8 pines son para el dato y el control , y un pin es el strobe que habilita la comunicación por el puerto paralelo.

Para diseñar el sistema en el FPGA primero se hizo uso del programador ACTIVE HDL que permite realizar el diseño del software y comprobar su funcionamiento correcto mediante un simulador. Sin embargo, mediante este programa no es posible implementar el diseño. Por eso se utilizó otra herramienta, el programador de Xilinx *Project Navegator* que es capaz de sintetizar, implementar el diseño y configurar el dispositivo FPGA, es decir, pasar el software a hardware.

Los pasos a seguir al utilizar este programador son los siguientes:

1. Generar un nuevo proyecto que contenga los módulos y las interconexiones internas en el diseño.
2. Sintetizar el programa general que contiene el diseño completo para encontrar advertencias o errores y corregirlos.
3. Una vez corregidos los errores de sintaxis, mapear las señales de entrada y salida del FPGA hacia los pines correspondientes.
4. Implementar el diseño mediante la traducción, mapeo y generación de rutas internas al FPGA.

5. Finalmente se genera el archivo de programación y se configura el dispositivo, es decir, se programa el FPGA.

Siguiendo estos pasos se implementó el sistema de control y generación de PWM, obteniendo como resultado las señales deseadas.

De acuerdo al sistema completo de la Fuente de Voltaje Programable, las señales conseguidas mediante el FPGA son procesadas mediante las siguientes etapas, la de conversión y filtrado. La función de cada una de las etapas en el procesamiento de las señales se describe en el siguiente capítulo.