

**Tesis para obtener el título de:**  
Ingeniero en Mecatrónica

## **Descripción de Hardware para un sistema de control en un FPGA**

**Presenta:**

Iván Manuel García Hernández

**Asesor de tesis:** M.C. Rafael Fernando González Zárate

**Co-Asesor de tesis:** Dr. Hiram Netzahualcóyotl García Lozano

**Asesor externo:** M.C. Felipe Santiago Espinosa

**AGOSTO 2014**

## **Dedicatorias**

A mis padres y hermanos, gracias por el apoyo incondicional durante toda mi vida.

Para el tío Juan, te extrañamos.

Para ti Cap, nada pides y todo lo das.

## **Agradecimientos**

Al profesor Rafa por permitirme trabajar con él y los buenos ratos de risa que nos brinda a todos. <http://instantrimshot.com/>

Al profesor Hiram por sus grandes consejos, por ser mi tutor durante toda la carrera y por su gran disposición al momento de hacerme las observaciones necesarias. Muchas gracias.

Al profesor Felipe por brindarme tiempo para este trabajo, su colaboración fue vital para que se llevara a cabo.

A Mercedes que me ayudó en la redacción.

A mis amigos Jacobo, Gaspar, Temo y Eric por su amistad en los años de la universidad.

A toda la familia Hernández Marín por abrirme las puertas de su casa cuando iba a Huajuapán y Oaxaca y tratarme como un integrante más.

**“Comprender a los demás es sabio, comprenderse a uno mismo es estar iluminado. El que vence a los otros es fuerte, pero el que se vence a sí mismo es poderoso”.**

**Lao Tsé.**

## Resumen

La teoría de control permite aplicar diversas técnicas a distintos sistemas con el fin de alcanzar un desempeño deseado. Esto brinda ventajas como son: Ahorro de combustibles, reducción de costos de fabricación, entre otros.

La tendencia actual de los sistemas de control se enfoca principalmente hacia los sistemas digitales, tomando el nombre de controladores digitales. Los controladores digitales brindan una capacidad de diseño que favorece la descripción de técnicas de control complejas. Dos factores importantes a considerar en estos sistemas son el tiempo de discretización y la velocidad de procesamiento. El primero es la capacidad que tiene el sistema para obtener una representación digital de la señal de entrada y así poder procesarla. Y el segundo indica la velocidad con que el sistema puede ejecutar operaciones a dicha señal con el fin de obtener una salida que concuerde con la acción de control diseñada. La velocidad de ejecución estará determinada por el tipo de implementación que se realice.

Para este fin, se han desarrollado sistemas digitales que efectúan una programación en forma secuencial, es decir, que sólo pueden ejecutar una operación a la vez. Este tipo de programación tiene su desventaja al momento de considerar el tiempo de ejecución del programa, ya que entre más operaciones se tenga mayor es el tiempo de retraso entre cada iteración. Existe otro tipo de implementación, de forma concurrente, que ofrece un mejor aprovechamiento del tiempo de ejecución del programa. La implementación en forma concurrente permite ejecutar operaciones del programa al mismo tiempo. Actualmente se encuentran sistemas digitales, desarrollados por diversos fabricantes, cada uno adaptándose a distintas necesidades y tipos de programación. Uno de estos sistemas se conoce como los FPGA's. Los cuales brindan la posibilidad de desarrollar diseños digitales tanto en secuencial como en concurrente.

En este trabajo se realizará el diseño de un sistema de control PID de propósito general en una FPGA utilizando un lenguaje de descripción de hardware (LDH). Para ello se hará uso de una tarjeta Spartan 3E de Xilinx, la cual cuenta (entre otras cosas) con un convertidor Analógico-Digital, un convertidor Digital-Analógico y un LCD de 16x2.

## Abstract

Control theory allows to apply diverse techniques to different systems in order to reach a wished performance. This offers advantages such as: Saving fuel, reducing manufacturing costs, among others.

The current trend of the control systems primarily focus on digital systems, taking the name of digital controllers. Digital controllers provide a design capacity that favors the description of complex control techniques. Two important factors to consider in these systems are the time of discretization and the speed of processing. The first one is the capacity that has the system to obtain a digital representation of the input signal so we can process it. And the second one indicates the speed with which the system can execute operations to the mentioned signal in order to obtain an output that matches the action with the designed control. Execution speed will be determined by the type of implementation that is realized.

For this end, there have developed digital systems that effect a programming in sequential form, that is, that only one operation can be executed simultaneously. This type of programming has his disadvantage to the moment to consider the time of execution of the program, because with more operations have greater time delay between each iteration. There exists another type of implementation, concurrently, providing better exploitation of the execution time of the program. The concurrently implementation allows executing program operations simultaneously. They are currently available such systems, developed by different manufacturers, each adapted to different needs and types of programming. One of these systems is known as the FPGA's. These systems provide the opportunity to develop digital designs both sequentially and concurrently.

In this work there will be realized of a general purpose PID control system in a FPGA using a hardware description language (HDL). For this we will use a Spartan 3E Xilinx, which features (among other things) an Analogical - digital convertor, a Digital - analogical convertor and a LCD of 16x2.

# Índice general

Capítulo 1: Introducción.....	10
1.1 Sistemas de control .....	10
1.2 Teoría de control clásico .....	12
1.3 Diseño de sistemas de control .....	14
1.3.1 Compensador PD (Proporcional-derivativo) .....	14
1.3.2 Compensador PI (Proporcional-integral) .....	14
1.3.3 Compensador PID (Proporcional-integral-derivativo).....	15
1.4 Controladores analógicos y digitales.....	15
1.4.1 Controladores analógicos.....	16
1.4.2 Controladores digitales .....	17
Capítulo 2: Dispositivos lógicos programables.....	18
2.1 Estados lógicos .....	18
2.2 Operaciones y funciones lógicas .....	19
2.3 Dispositivos lógicos programables .....	21
2.4 Lenguajes de descripción de hardware (LDH).....	25
2.4.1 VHDL.....	25
Capítulo 3: Convertidores Analógico-Digital y Digital-Analógico .....	28
3.1 Señales analógicas.....	28
3.2 Señales digitales .....	29
3.3 Discretización de señales .....	30
3.3.1 Muestro y filtrado .....	30
3.3.2 Retención del valor muestreado .....	31
3.4 Conversión Analógico-Digital .....	32
3.4.1 Convertidor Analógico-Digital paralelo (Flash) .....	32
3.4.2 Convertidor Analógico-Digital de doble pendiente.....	33
3.4.3 Convertidor Analógico-Digital por aproximaciones sucesivas .....	34
3.5 Convertidor Digital- Analógico .....	35
3.5.1 Convertidor Digital-Analógico con ponderación binaria.....	35
3.5.2 Convertidor Digital-Analógico en escalera R/2R .....	36
3.6 Protocolos de comunicación. ....	37

Capítulo 4: Diseño del módulo en VHDL .....	38
4.2 Interface SPI .....	39
4.3 Interface de usuario .....	48
4.3 Módulo de control.....	56
Capítulo 5: Verificación del módulo en VHDL .....	60
5.1 Implementación del módulo SPI .....	60
5.2 Verificación del módulo de usuario .....	62
5.3 Verificación del módulo de control.....	63
Capítulo 6: Conclusiones y trabajos a futuro .....	66
Referencias.....	67
ANEXOS .....	68

## Índice de imágenes

Figura 1.1 Entrada y salida de un elevador .....	11
Figura 1.2 Sistema de control de lazo cerrado .....	12
Figura 1.3 Diagrama a bloques de un compensador PID .....	15
Figura 1.4 Diagrama de un amplificador operacional .....	16
Figura 1.5 Diagrama a bloques de un controlador digital .....	17
Figura 2.1 Rango de voltajes de un circuito digital .....	19
Figura 2.2 Función lógica combinacional de cuatro entradas y una salida .....	20
Figura 2.3 Modelo de un sistema combinacional .....	21
Figura 2.4 Diagrama a bloques de un PLA .....	22
Figura 2.5 Diagrama a bloques de un PAL .....	22
Figura 2.6 Ejemplo de estructura y cableado CPLD .....	23
Figura 2.7 Diagrama de bloques de la estructura de un FPGA .....	24
Figura 2.8 Representación gráfica y sintaxis de una entidad .....	26
Figura 3.1 Ejemplo de señal analógica.....	28
Figura 3.2 Muestreo de una señal analógica.....	29
Figura 3.3 Una señal analógica y su aproximación en “escalera” .....	30
Figura 3.4 Ilustración de muestreo y retención .....	31
Figura 3.5 Función básica de un CAD .....	32
Figura 3.6 CAD flash de 3 dígitos.....	33
Figura 3.7 CAD de doble pendiente de 8 dígitos .....	33
Figura 3.8 CAD por aproximaciones finitas de 4 dígitos.....	34
Figura 3.9 CDA con ponderación binaria.....	36
Figura 3.10 CDA en escalera R/2R .....	36
Figura 3.11 Sistema SPI para un maestro y tres esclavos .....	37
Figura 4.1 Spartan 3E desde una vista superior.....	38
Figura 4.2 Comportamiento de la interface SPI .....	39
Figura 4.3 Circuito de captura analógica.....	40
Figura 4.4 Protocolo de comunicación del Pre-Amplificador.....	42
Figura 4.5 Diagrama de tiempos del Pre-Amplificador .....	42

Figura 4.6 Proceso de conversión de un valor analógico .....	44
Figura 4.7 Diagrama del circuito del CDA .....	44
Figura 4.8 Protocolo de 24 bits para el CDA .....	45
Figura 4.9 Entidad del módulo SPI .....	46
Figura 4.10 Diseño de la máquina de estados finitos para los dispositivos SPI.....	47
Figura 4.11 Diagrama de conexión entre el FPGA y el LCD.....	48
Figura 4.12 Diagrama del encoder .....	49
Figura 4.13 Comportamiento de las señales A y B del encoder.....	50
Figura 4.14 Comparación de las señales antes y después del filtro.....	51
Figura 4.15 Entidad para la codificación de la representación del voltaje de entrada .....	52
Figura 4.16 Representación del voltaje de entrada.....	52
Figura 4.17 Código empleado para la representación del voltaje de entrada.....	53
Figura 4.18 Entidad destinada al control del encoder .....	53
Figura 4.19 Filtro aplicado al encoder.....	54
Figura 4.20 Entidad del error del sistema.....	54
Figura 4.21 Entidad de la interface de usuario .....	55
Figura 4.22 Configuración en paralelo para un controlador PID .....	57
Figura 4.23 Entidad Fub9 .....	57
Figura 4.24 Bits destinados para la parte entera y parte fraccionaria del error .....	58
Figura 4.25 Entidad PID.....	58
Figura 4.26 Código de la ley de control PID .....	59
Figura 5.1 Funcionamiento del CDA .....	60
Figura 5.2 Funcionamiento del módulo SPI.....	61
Figura 5.3 Implementación del módulo de usuario .....	62
Figura 5.4 Simulación del módulo de control .....	63
Figura 5.5 Incremento de la componente iT.....	64
Figura 5.6 Adaptación de los bits del CDA.....	64
Figura 5.7 Circuito externo para la adaptación del CDA .....	65
Figura 6.1 Descripción de hardware para un sistema de control .....	66

# Capítulo 1: Introducción

## 1.1 Sistemas de control

Desde épocas remotas el ser humano ha tenido la necesidad de manipular el comportamiento de distintos sistemas con los que se ha relacionado (controlar). Un sistema es un conjunto de partes interactuando entre sí para realizar una tarea en específico. Los griegos empezaron a controlar sistemas hacia el año 300 a.C. (un reloj de agua inventado por Tesebio). El primer sistema considerado de control fue el regulador centrífugo de James Watt, con el cual controló la velocidad de una máquina de vapor mejorando su desempeño. En este capítulo se hará una descripción de los conceptos básicos empleados en la teoría de control para el diseño de controladores.

Un sistema de control puede considerarse como un sistema que produce una salida como respuesta a una entrada. La entrada representa el valor deseado y la salida es el valor real en el que se encuentra el sistema. Por ejemplo, cuando una persona utiliza un elevador debe presionar un botón para seleccionar el piso al cual quiere ir, esto se puede considerar como la señal de entrada [1]. El elevador comenzará a moverse hacia el piso seleccionado a una velocidad y precisión diseñada, esto puede considerarse la salida. Dos características que hacen que la salida actual no sea igual a la deseada (entrada) son:

- Limitantes del sistema: Las entidades físicas no pueden cambiar su estado de manera instantánea (un estado puede ser la velocidad o la posición, entre otras). El estado cambia de acuerdo al sistema y la manera en que adquiere o disipa la energía. A esta característica del sistema se le conoce como *respuesta transitoria*. La importancia de esta característica radica en conocer los efectos que ocasiona. En el caso del elevador, una respuesta transitoria lenta haría que los usuarios se impacienten, y una respuesta muy rápida podría ocasionarles lesiones.

- Aproximación a la respuesta deseada: Volviendo al caso del elevador, la aproximación a la respuesta deseada ocurre cuando se está por llegar al piso seleccionado. Durante la nivelación del elevador con respecto al piso, se tiene una diferencia conocida como *error en estado estable*. El elevador tiene que estar lo suficientemente nivelado con el piso para que el usuario pueda salir de él. El error en estado estable es cuantificable, lo que permite ser analizado y diseñar una acción de control correctiva.

De acuerdo a lo anterior, la respuesta total de un sistema puede ser considerada como sigue:

$$\text{Respuesta total} = \text{Respuesta libre} + \text{Respuesta forzada}$$

La respuesta libre describe la forma en que el sistema adquiere o disipa energía. Dicha respuesta debe decaer a cero a medida que pasa el tiempo (estabilidad del punto de equilibrio). En caso de un sistema que no presente esta característica (sistema inestable), podría llevarlo a su autodestrucción. En el ejemplo del elevador, la inestabilidad podría provocar que éste choque contra el piso o contra el techo.

La respuesta forzada es aquella que sólo es influenciada por la entrada (condiciones iniciales nulas). En la figura 1.1 se muestra la relación de la posición con respecto al tiempo de un elevador.

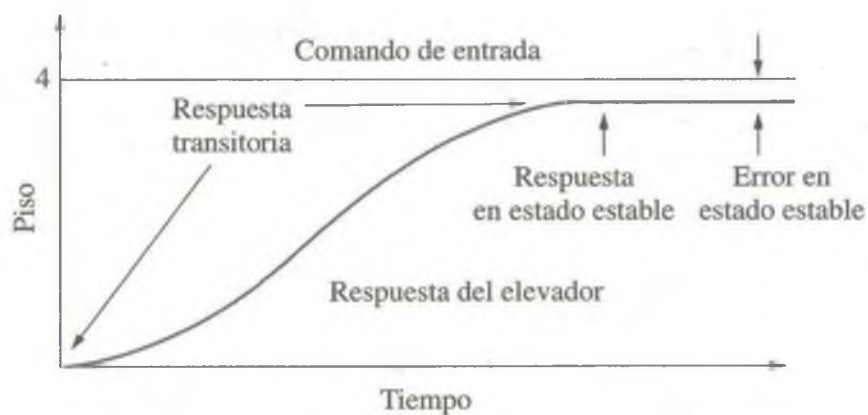


Figura 1.1 Entrada y salida de un elevador

Como se puede apreciar, el sistema requiere de un tiempo determinado para cambiar de la posición inicial a la posición deseada.

El área de la ingeniería que permite estudiar un sistema, modelarlo y manipularlo se conoce como ingeniería de control. En la figura 1.2 se ilustra, en forma general, un sistema de control.

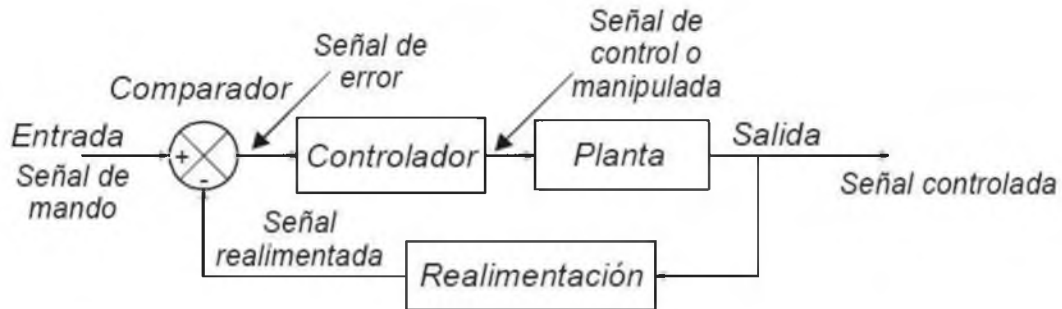


Figura 1.2 Sistema de control de lazo cerrado

Aquí, la *entrada* es una señal la cual indica el valor al que se desea llevar el sistema [2]. La resta entre la entrada y el valor actual del sistema da como resultado la *señal de error*. El *controlador* es la parte del sistema que se encarga de llevar el estado al valor de referencia, utilizando alguna estrategia de control en específico. La parte física del sistema que se desea controlar, la cual puede ser un horno, un vehículo o un motor, se conoce como *planta*. La *señal de control* está definida como la salida generada por el controlador para que el sistema esté en el valor de referencia. Finalmente la *salida* es una señal que representa el estado actual del sistema. Esta se emplea para realizar una operación de *retroalimentación* (para generar el error).

## 1.2 Teoría de control clásico

Con el fin de controlar un sistema es necesario una representación matemática de la dinámica del sistema (relación entrada/salida). Una ecuación diferencial puede describir la relación entrada/salida de un sistema por medio de razones de cambio. Por ejemplo en un

sistema eléctrico, utilizando las leyes de Kirchhoff, se puede obtener una expresión matemática que describa el comportamiento en el tiempo de la corriente (razón de cambio de la carga) con respecto a los voltajes de entrada.

Un método operacional, usado principalmente para solucionar ecuaciones diferenciales lineales, es la transformada de Laplace. Con esta herramienta es posible representar la salida, la entrada y los sistemas, como entidades separadas relacionadas mediante términos algebraicos en términos de la variable compleja  $s$ .

En teoría de control clásico se usan las denominadas *funciones de transferencia*. Una función de transferencia se define como el cociente de la transformada de Laplace de la salida y la transformada de Laplace de la entrada bajo la suposición de que todas las condiciones iniciales son cero. Esta función de transferencia se considera un modelo matemático del sistema exclusivamente para ecuaciones lineales invariantes en el tiempo. La representación de una función de transferencia es la siguiente:

$$\text{Función de transferencia} = G(s) = \frac{Y(s)}{X(s)},$$

donde  $Y(s)$  es la transformada de Laplace de la salida y  $X(s)$  es la transformada de Laplace de la entrada.

En una función de transferencia se involucran dos términos fundamentales: polos y ceros. Los polos de una función de transferencia son:

- Los valores  $s$  de la transformada de Laplace que vuelven infinita la función de transferencia.
- Cualquier raíz del denominador común a las raíces del numerador.

Los polos afectan considerablemente la respuesta transitoria determinando el comportamiento del sistema.

Los ceros de una función de transferencia son:

- Los valores de la variable de la transformada de Laplace  $s$  que hace que la función de transferencia se convierta en cero.

- Cualesquiera raíces del numerador de la función de transferencia que sean comunes con las raíces del denominador.

### 1.3 Diseño de sistemas de control

Para mejorar algunas características de los sistemas tales como la velocidad de respuesta o el error en estado estable se puede utilizar un *compensador* [2]. Este es un dispositivo que se inserta con el fin de compensar las deficiencias del sistema original. Existen distintos tipos de compensadores, cada uno aplica una acción llamada ley de control.

#### 1.3.1 Compensador PD (Proporcional-derivativo)

Este tipo de compensador emplea la razón de cambio del error del sistema. Esto permite conocer la tendencia, y por lo tanto, aplicar una acción de control correctiva más adecuada. El compensador PD tiene la siguiente función de transferencia.

$$G_c(s) = Kp(1 + T_d s),$$

donde  $Kp$  es la ganancia proporcional y  $T_d$  el tiempo derivativo. Este compensador mejora la respuesta en estado transitorio y afecta el error en estado estable, sólo si el error varía con respecto del tiempo. La acción derivativa mantiene el error al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce.

#### 1.3.2 Compensador PI (Proporcional-integral)

Este tipo de compensador aumenta en uno el orden del sistema, mejorando el error en estado estable. El orden se refiere al grado del denominador de la función de transferencia después de la cancelación de los factores comunes con el numerador. El compensador PI está caracterizado por la siguiente función de transferencia:

$$G_c(s) = Kp(1 + \frac{1}{T_i s}),$$

donde  $Kp$  es la ganancia proporcional y  $Ti$  el tiempo integral. El compensador PI puede provocar que el sistema sea menos estable o incluso inestable, por lo tanto los valores de  $Kp$  y  $Ti$  se deben elegir cuidadosamente para que se obtenga la respuesta requerida. [3]

### 1.3.3 Compensador PID (Proporcional-integral-derivativo)

Este compensador tiene las ventajas de los compensadores PD y PI en la misma acción de control. La figura 1.3 muestra el diagrama a bloques de este compensador,

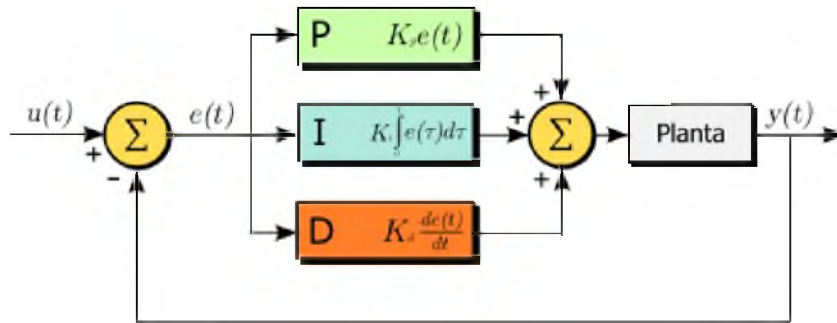


Figura 1.3 Diagrama a bloques de un compensador PID

donde la función de transferencia del compensador es la siguiente:

$$G_c(s) = Kp(1 + \frac{1}{T_i s} + T_d s)$$

Cada una de las estrategias de control; proporcional, integral y derivativa, tiene un peso en la suma final, el cual viene dado por los parámetros que con los que se hayan diseñado.

## 1.4 Controladores analógicos y digitales

Desde el año 1960 la teoría de control ha ocupado un papel muy importante en el avance de la ingeniería. Las distintas aplicaciones en las que es posible aplicar los métodos

el control le permitieron convertirse en una ingeniería interdisciplinaria (en conjunto con la electrónica y computación principalmente).

Con el desarrollo de los dispositivos electrónicos y de las computadoras, se fueron realizando controladores cada vez más sofisticados. Inicialmente los controladores fueron implementados por medio de dispositivos eléctricos, mecánicos y electromecánicos. Para sistemas electrónicos se utilizaron amplificadores operacionales, capacitores, inductores, entre otros, lo que dio origen a los controladores analógicos. El desarrollo de los sistemas digitales a bajo costo y con gran capacidad de procesamiento hizo posible la implementación de estas leyes de control por medio de una computadora, esto dio inicio al control digital.

### 1.4.1 Controladores analógicos

Los controladores analógicos pueden ser implementados por medio de amplificadores operacionales. En la figura 1.4 se ilustra el diagrama de un amplificador operacional.

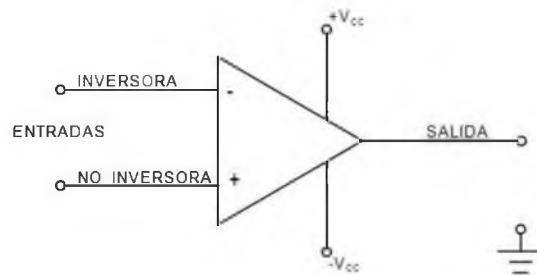


Figura 1.4 Diagrama de un amplificador operacional.

Este dispositivo, como su nombre lo indica, es capaz emular operaciones matemáticas tales como la suma, resta, integración y derivación, a fin de construir las funciones de transferencia de los controladores antes mencionados.

Los controladores analógicos resultan ser una opción muy viable para realizar controladores que no requieran de una ley de control muy compleja.

## 1.4.2 Controladores digitales

Los controladores digitales permiten implementar leyes de control más versátiles desde simples controles proporcionales hasta controles con desempeño óptimo. La figura 1.5 ilustra el diagrama a bloques de un control digital.



Figura 1.5 Diagrama a bloques de un controlador digital.

Debido a que la salida de la planta y la entrada de referencia siempre son señales analógicas, para utilizar un sistema de control digital se requieren de sistemas que permitan la conversión de señales de tiempo continuo a tiempo discreto (convertidores analógico digital, CAD) y viceversa (convertidor digital analógico, CDA). El controlador será implementado en una computadora donde estos datos serán procesados para generar una señal de control digital. Esta señal debe ser regresada a su formato original para que la ley de control pueda ser utilizada por la planta.

En los sistemas de control digital no es posible implementar la transformada de Laplace debido a que esta herramienta es exclusiva para sistemas con señales continuas. Existe una herramienta análoga empleada en sistemas con señales discretas, esta herramienta es conocida como transformada Z.

Actualmente la creciente capacidad de procesamiento y memoria en los sistemas digitales ha permitido a los controladores tener una amplia gama de dispositivos para su implementación, como son los dsPICs, y los FPGAs. Estos últimos sobresalen de los demás debido a su desempeño en cuanto a capacidad de procesamiento, velocidad y flexibilidad de diseño.

# Capítulo 2: Dispositivos lógicos programables

Algunos dispositivos que se emplean en la vida diaria están basados en circuitos digitales. Ejemplos de estos son los relojes digitales, las computadoras personales, entre otros. Antes de 1960, los circuitos digitales ya eran encapsulados en circuitos integrados (CI). En esta época su construcción era por medio de componentes individuales, por lo tanto los sistemas eran muy voluminosos. A partir de ese año la tecnología empleada en el desarrollo y fabricación de dispositivos digitales evolucionó de forma significativa. Los circuitos integrados son fabricados sobre una oblea de silicio y se colocan en el interior de un encapsulado de tamaño reducido.

## 2.1 Estados lógicos

La teoría base sobre la cual los sistemas digitales son construidos se conoce como lógica booleana [4]. La lógica es la parte del razonamiento que nos indica que ciertas proposiciones son verdaderas si éstas cumplen con condiciones dadas. Estas proposiciones se clasifican en: falsas (estado lógico bajo, 0) y verdaderas (estado lógico alto, 1). Para representar cada nivel lógico se usan voltajes, diferentes para cada tecnología. La figura 2.1 ilustra el rango general de variación para los niveles “alto” y “bajo”.

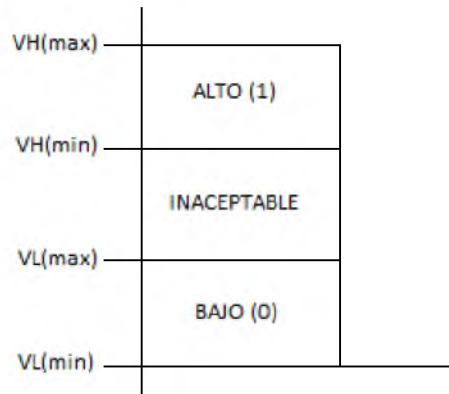


Figura 2.1 Rango de voltajes de un circuito digital.

El valor máximo del nivel bajo se representa mediante  $V_L(\max)$  y el voltaje mínimo mediante  $V_L(\min)$ , cualquier voltaje dentro de este rango se va a considerar como estado “bajo”. De la misma manera los valores comprendidos entre  $V_H(\max)$  y  $V_H(\min)$  van a ser considerados como el estado “alto”. Los valores que se ubican entre  $V_H(\min)$  y  $V_L(\max)$  no van a ubicarse en ninguno de los dos estados.

## 2.2 Operaciones y funciones lógicas

En el año 1850 el matemático y lógico George Boole desarrolló un sistema matemático para formular proposiciones. Este sistema está basado en símbolos de manera que se pueden escribir y resolver similar al álgebra. Este sistema es conocido hoy en día como álgebra de Boole. El álgebra de Boole está basado en 3 operaciones lógicas básicas: NOT, AND, OR, (existen más operadores, pero son derivados de estas tres). Estas operaciones requieren de variables de entrada y producen una única respuesta para cada determinado conjunto de condiciones.

La función NOT cambia el nivel de la variable de entrada a su nivel lógico opuesto. La función AND genera un nivel alto, si y sólo si todas sus entrada se encuentran en un nivel lógico alto. Por último la función OR genera un nivel alto, cuando cualquiera de sus entradas tiene un nivel lógico alto. A continuación se muestra la tabla de verdad para las

operaciones lógicas OR, AND y NOT. Las cuales indican la salida en respuesta a una entrada.

Entradas		Salida
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Entradas		Salida
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Entrada	Salida
A	X
0	1
1	0

Tablas de verdad para las operaciones OR, AND y NOT respectivamente.

Las operaciones lógicas nos permiten crear funciones lógicas combinacionales y secuenciales. En la figura 2.2 se muestra el ejemplo de una función lógica combinacional de cuatro entradas y una salida.

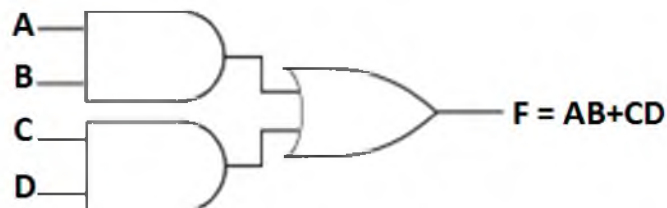


Figura 2.2 función lógica combinacional de cuatro entradas y una salida.

El circuito está formado por 2 operaciones AND y una operación OR. La salida es única y exclusivamente dependiente de las entradas por esto se conoce como combinacional (combinación de las entradas).

En los sistemas secuenciales, el valor de las salidas, no sólo es función del valor de las entradas, sino también de la “historia” o “secuencia” previa por la que han atravesado dichas entradas [8]. Esto se describe en la siguiente figura.

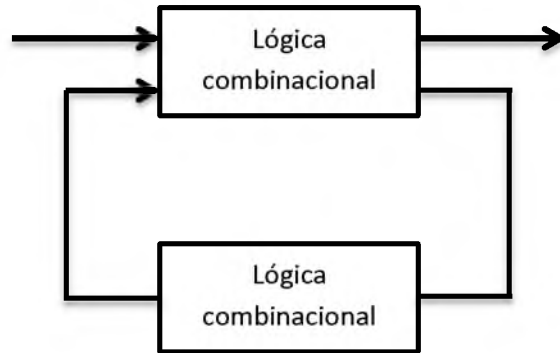


Figura 2.3 Modelo de un sistema secuencial.

## 2.3 Dispositivos lógicos programables

El desarrollo de los circuitos integrados hizo posible manufacturar unidades que tuvieran diversos componentes dentro de un espacio compacto. Esto dio paso a los tres principales tipos de circuitos integrados que existen: Los CI de función específica, los dispositivos lógicos programables (DLP) y los CI diseñados a la medida.

Los CI de función específica son diseñados para cumplir con una tarea en particular. Cuentan con un número pequeño de componentes y siempre realizan la misma función.

Los DLP son circuitos que el usuario puede configurar para el diseño de diferentes circuitos lógicos. Una ventaja que ofrecen estos dispositivos es que pueden configurarse en repetidas ocasiones, permitiendo al diseñador implementar diferentes circuitos en el mismo hardware.

Finalmente los CI diseñados a la medida son aquellos CI que el usuario diseña desde cero, es decir, desde la función que va a realizar, como también la tecnología apropiada para su implementación. Una desventaja de estos CI son los altos precios de producción, ya que si no va a ser implementado en un producto fabricado masivamente, el costo de producción de un único CI resulta ser elevado.

Existen diferentes tipos de DLP y se clasifican dependiendo la forma en que están ordenados sus elementos internos. A continuación se describirán algunos tipos de DLP.

- PLA(programmable logic array). Está conformado por una matriz AND y una matriz OR, ambas programables. La idea base de este dispositivo es de que las funciones lógicas se pueden realizar en suma de productos. La figura 2.4 muestra la arquitectura de una PLA con 3 variables de entrada.

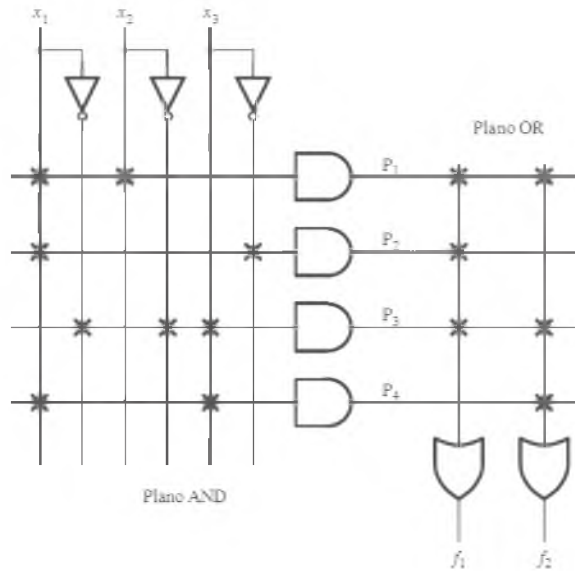


Figura 2.4 Diagrama de bloques de una PLA

- PAL (programmable array logic). Estos fueron desarrollados para mejorar las fallas de los PLA, de modo que eran menos costosos y su producción era mas simple. La figura 2.5 ilustra un diagrama de bloques de una PAL.

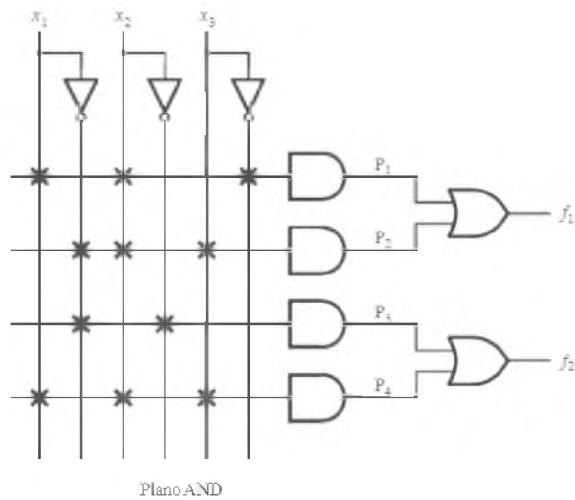


Figura 2.5 Diagrama de bloques de una PAL

- Los CPLD son básicamente versiones más grandes de los PLD, funcionando bajo el mismo principio de interconexión programada. Pasando al final por una macrocelda (circuitos basados en elementos básicos de memoria) que permite agregar la opción de retroalimentación a la matriz de interconexión. Los CPLD contienen una arquitectura limitada ya que cuenta con pines de entrada, salida y reloj específicos, limitando la flexibilidad del diseño. La figura 2.6 ilustra la estructura y cableado de un ejemplo de CPLD.

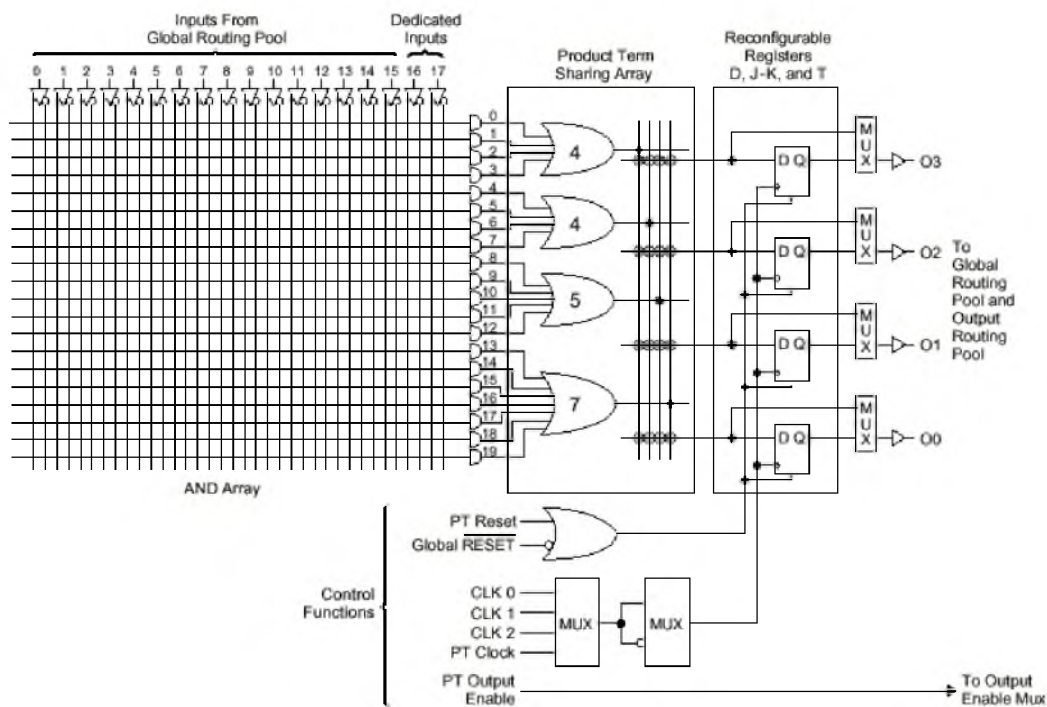


Figura 2.6 Ejemplo de estructura y cableado CPLD [4]

- FPGA (Fiel programmable gate array). Los FPGA surgieron como la evolución de los CPLD. Estos contienen una matriz de interconexión de mayor en comparación con los CPLD. El tipo de arquitectura es la principal diferencia entre ambos dispositivos. Los dispositivos FPGA consisten en tres elementos configurables: bloques lógicos configurables (CLB), bloques de entrada y salida (IOB) y canales de comunicación (CC). Los CLB se

comunican entre ellos y con las terminales de entrada/salida (E/S) por medio de alambros llamados canales de comunicación. En la figura 2.7 Se muestra el diagrama a bloques de una FPGA.

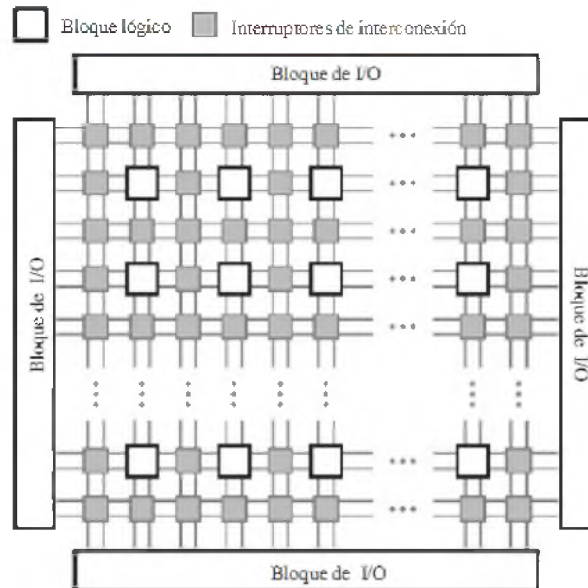


Figura 2.7 Diagrama de bloques de la estructura de un FPGA

Los bloques lógicos se ubican en un arreglo bidimensional, mientras que los canales de enrutamiento se ubican de manera horizontal y vertical entre filas y columnas. Mediante programación se activan interruptores que permiten que los bloques lógicos se conecten de distintas maneras y así se puedan formar las funciones lógicas. Para realizar un circuito en un FPGA, cada función debe ser lo suficientemente pequeña como para encajar en un solo bloque lógico. Se utiliza un lenguaje de descripción de hardware para determinar la manera en se configuraran las conexiones entre los bloques lógicos de la FPGA.

Debido a la demanda de nuevos dispositivos que ofrecieran a los diseñadores de controladores un rendimiento más elevado, surgieron empresas dedicadas al desarrollo y producción de dispositivos electrónicos. Xilinx, empresa Estadounidense fundada en 1984 y es uno de los principales proveedores a nivel mundial de FPGA.

## 2.4 Lenguajes de descripción de hardware (LDH)

Un lenguaje de programación está diseñado para elaborar procesos que pueden ser realizados por una computadora u otro dispositivo digital. Se basan en procedimientos empleando una escritura específica, posteriormente después de concluir el código se procede a compilarlo para verificar su funcionamiento y ser implementado en el dispositivo. Los lenguajes de descripción de hardware son similares a los lenguajes de programación típicos, con la diferencia de que estos realizan una descripción del comportamiento del circuito en lugar de un programa. Existen distintos tipos de LDH, como son el Verilog, ABEL y el VHDL, cada uno tiene una sintaxis definida. Sin embargo el VHDL ha tenido mayor difusión, debido a que facilita el diseño de circuitos.

### 2.4.1 VHDL

Cada fabricante de dispositivos de lógica programable desarrolla su propio LDH, que sólo puede ser empleado en la tecnología que esta compañía desarrolla. Un LDH que puede ser utilizado en todos los DLP es el VHDL. Este LDH está basado en el estándar de IEEE Std 1076-1987 y IEEE Std 1076-1993. VHDL contiene un amplio campo de aplicación, que va desde modelado para simulación de circuitos, hasta la síntesis automática de circuitos.

Las principales ventajas del desarrollo de circuitos en VHDL son:

- Notación formal: La programación en VHDL cuenta con una notación que permite su uso en cualquier diseño electrónico.
- Disponibilidad pública: VHDL no está sometido a patente o marca registrada alguna.
- Independencia tecnológica de diseño: VHDL se diseñó para soportar diversas tecnologías de diseño.
- Capacidad descriptiva en distintos niveles de abstracción: El proceso de diseño consta de varios niveles de detalle, desde la especificación hasta la implementación final.

- Reutilización del código: El uso de VHDL como lenguaje estándar permite reutilizar los códigos en diversos diseños, sin importar que hayan sido generados para una tecnología e implementación en particular.

VHDL ofrece la posibilidad de desarrollar un sistema de manera esquemática, haciendo subsistemas que se relacionan por medio de enlaces.

La disponibilidad de herramientas basadas en computadoras influyó enormemente en el proceso de diseño de sistemas digitales. La descripción de un circuito en VHDL está compuesta por tres elementos:

- Librerías: Contienen elementos que se utilizan en los circuitos digitales, tales como funciones lógicas, operadores matemáticos, tipos de datos, entre otros.
- Entidades: Tiene como objetivo modelar la interface de un circuito con el exterior mediante sus entradas y salidas. La figura 2.8 muestra la representación gráfica de una entidad y a la derecha la declaración en VHDL de la misma.

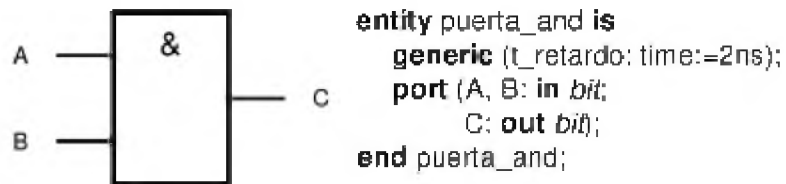


Figura 2.8 Representación gráfica y sintaxis de una entidad

- Arquitecturas: Es la descripción del funcionamiento de la entidad. Las soluciones para un circuito lógico no son únicas, por lo que se pueden tener diversas arquitecturas que funcionen para la misma tarea.

Una de las principales diferencia entre un programa y una descripción de hardware es que el primero se realiza de manera secuencial, es decir, existe un orden para la ejecución de instrucciones, mientras que el segundo se realiza de manera concurrente, lo que indica que

todas sus partes están activadas de manera simultánea. La siguiente tabla ilustra la ejecución de una suma de manera secuencial y de manera concurrente.

TIEMPO	EJECUCION					
	SECUENCIAL			CONCURRENTE		
	A	B	C	A	B	C
0	0	0	0	0	0	0
1	1	0	0	1	4	0
2	1	4	0	1	4	5
3	1	4	5			

Tabla comparativa de una suma de manera concurrente y secuencial

De manera secuencial primero deben asignarse los valores de A y B, para que C pueda obtener un valor. La operación se da por medio de sentencias con un orden establecido. Este mismo ejemplo funcionaria de manera diferente en un circuito concurrente, cuando transcurre una unidad de tiempo los valores de A y B cambian al mismo tiempo, y la operación se lleva a cabo en una unidad de tiempo menos que en la operación secuencial.

# Capítulo 3: Convertidores Analógico-Digital y Digital-Analógico

En la naturaleza los distintos fenómenos físicos cambian en forma continua. Por ejemplo a lo largo de un día la temperatura no varía de 20 a 25 grados de manera instantánea, sino que va tomando valores de forma continua hasta llegar de la temperatura inicial a la final. Debido a que los circuitos digitales sólo pueden ocupar dos estados, se debe construir una interface para que las señales continuas (analógicas) sean operacionalmente compatibles con los circuitos digitales. Esta interface se realiza mediante una conversión de señales analógicas a señales digitales. En este capítulo se hace una descripción de los procesos de conversión de Analógico-Digital (CAD) y Digital-Analógico (CDA).

## 3.1 Señales analógicas

Una señal analógica, en un intervalo continuo de tiempo, se define como la señal cuya amplitud puede adoptar valores continuos. En la figura 3.1 se ilustra la forma de una señal analógica.

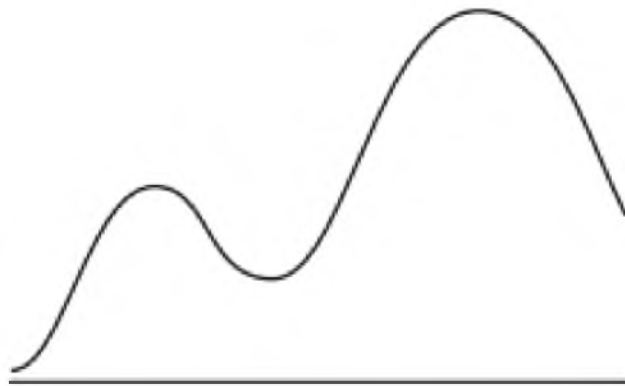


Figura 3.1 Ejemplo de señal analógica

Este tipo de señal puede ser representada por medio de una función matemática continua en la que es variable su amplitud y periodo en función del tiempo.

### 3.2 Señales digitales

Las señales digitales son aquellas que sólo están definidas en instantes de tiempo (que en nuestro caso se conoce como periodo de muestreo) y cuya amplitud se representa mediante magnitudes cuantificadas, esto quiere decir que el dato muestreado se va a representar mediante un valor discreto. La principal ventaja que tienen las señales digitales es de que el procesamiento de éstas es más eficiente. A continuación, se ilustra el proceso de muestreo de una señal analógica.

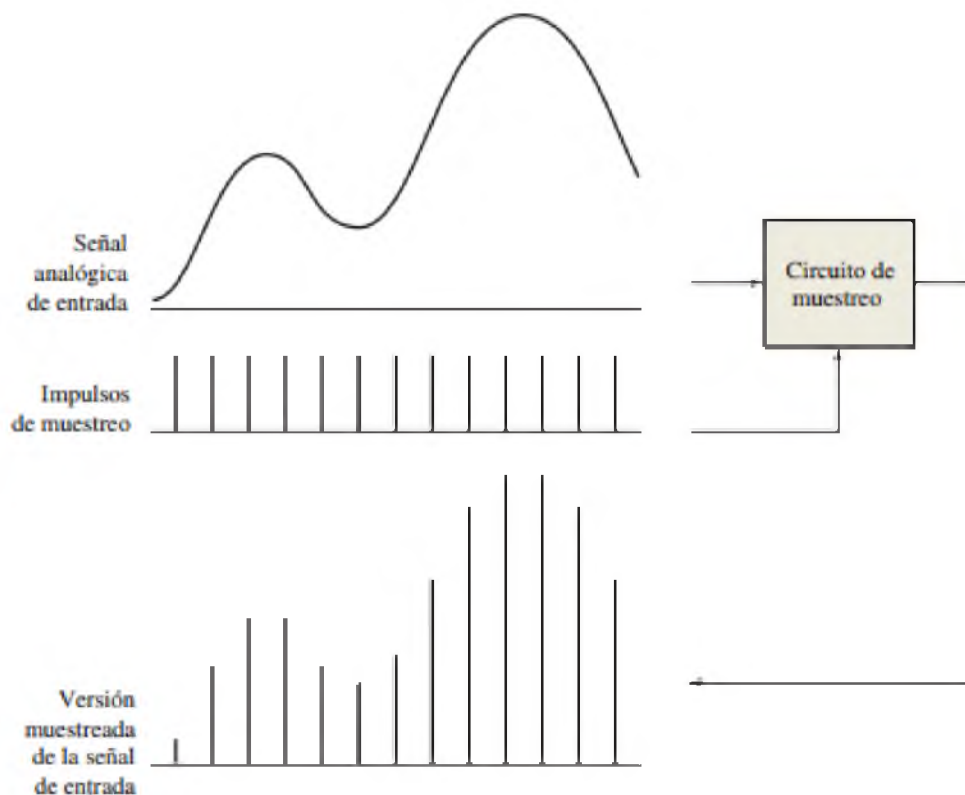


Figura 3.2 Muestreo de una señal analógica

El muestreo convierte la señal analógica en una serie de impulsos, cada impulso representa el valor de la amplitud de la señal en ese tiempo.

### 3.3 Discretización de señales

En forma general el proceso consiste en convertir una señal que tiene valores continuos a una señal con valores discretos. La figura 3.3 ilustra esta discretización para una señal senoidal.

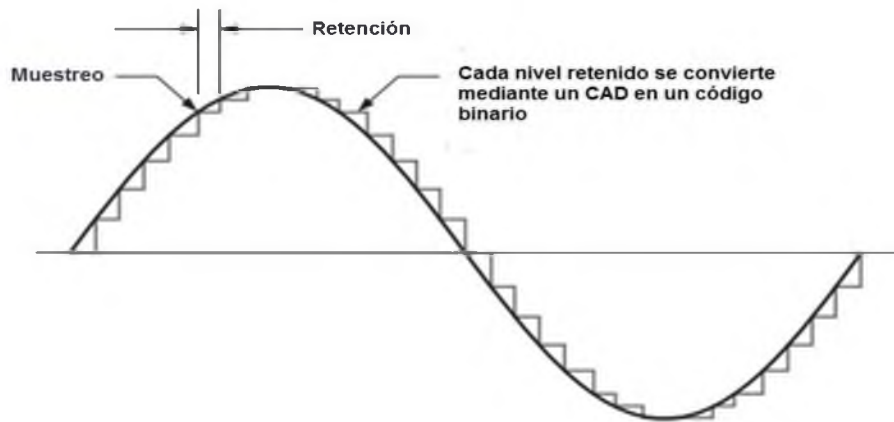


Figura 3.3 Una señal analógica y su aproximación en “escalera”

#### 3.3.1 Muestro y filtrado

En el muestreo se busca definir una forma de onda tomando los suficientes valores discretos en puntos determinados. Mientras más muestras se tomen se podrá definir de manera más precisa la señal. Existen ciertos criterios que deben cumplirse al momento de muestrear una señal:

- **Teorema de muestreo.** Cualquier señal analógica puede ser representada por una sucesión de señales senoidales de distintas amplitudes y frecuencias que sumadas dan como resultado la señal original (series de Fourier). Al momento de muestrear una señal se establece una frecuencia de muestreo. El teorema de muestreo indica

que para poder recuperar la señal original, la frecuencia de muestreo debe ser por lo menos dos veces mayor que la componente de mayor frecuencia.

- **Necesidad de filtrado.** Antes de muestrear una señal se debe establecer valor que será usado para eliminar las frecuencias armónicas que se encuentren por encima de este valor. Este valor es conocido como frecuencia de Nyquist. Por medio de frecuencia de Nyquist se implementa un filtro para eliminar los armónicos que excedan dicha frecuencia.
- **Aplicación.** La aplicación indica un punto de referencia para la selección de la frecuencia de muestreo. Por ejemplo, en las aplicaciones de sonido es común usar frecuencias de 32 KHz, 44.1 KHz o 48 KHz. El teorema de muestreo indica que la frecuencia de muestreo debe ser 2 veces mayor a la frecuencia máxima de sonido. Al utilizar 44.1 KHz es posible muestrear señales de 22 KHz, lo que supera a los 20Khz que comúnmente es lo que escucha el oído humano.

### 3.3.2 Retención del valor muestreado

Después del filtrado y muestreo de la señal, es necesario mantener el valor muestreado hasta el siguiente periodo de muestreo, a lo que se le llama retención. Esto permite que el CAD tenga el tiempo suficiente para procesar dicho valor. La figura 3.4 ilustra la operación de muestreo y retención de una señal. [9]

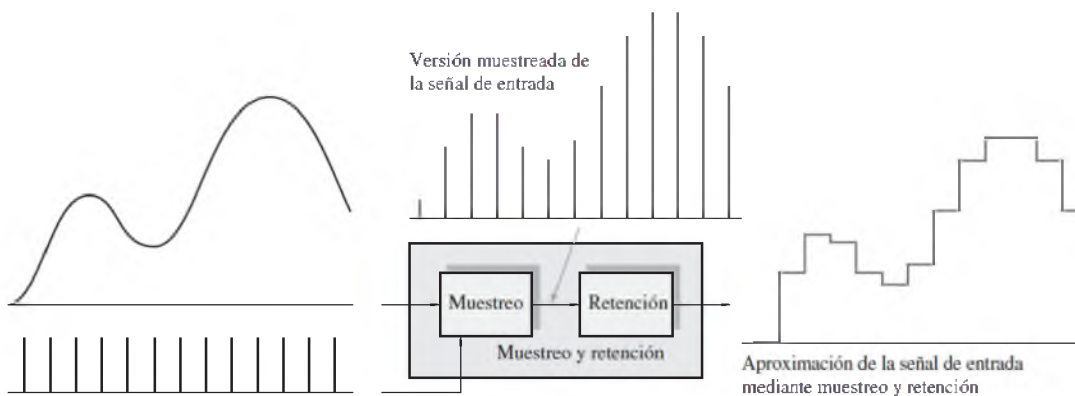


Figura 3.4 Ilustración de muestreo y retención.

### 3.4 Conversión Analógico-Digital

La conversión Analógico-Digital es el proceso donde la salida de un circuito de muestreo y retención se representa por medio de un código binario, tal como se muestra en la figura 3.5.

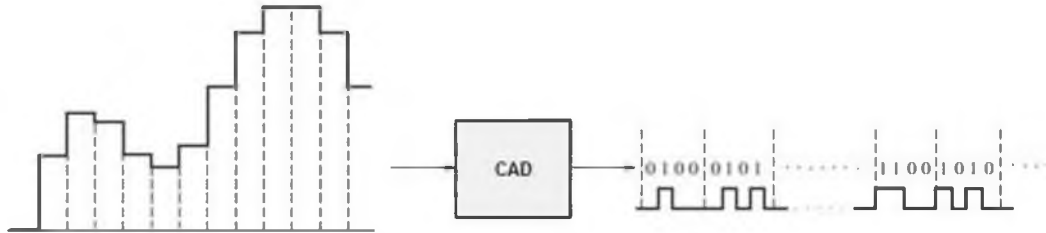


Figura 3.5 Función básica de un CAD.

Al proceso de conversión de un valor analógico a un código binario se conoce como *cuantificación*. Cuantos más bits se utiliza para la cuantificación, más exacta será la representación (resolución del CAD). Existen distintos tipos de CAD, a continuación se describirán lo más utilizados.

#### 3.4.1 Convertidor Analógico-Digital paralelo (Flash)

Este método de conversión consiste en una serie de comparadores basados en amplificadores operacionales. Por medio de una red de resistencias se generan distintos niveles de voltaje. Cada amplificador operacional compara el voltaje de entrada con un nivel de referencia. Si el voltaje de entrada es mayor o igual al voltaje de referencia se produce un estado “alto” que se agrega al codificador. Esto se ilustra en la figura 3.6.

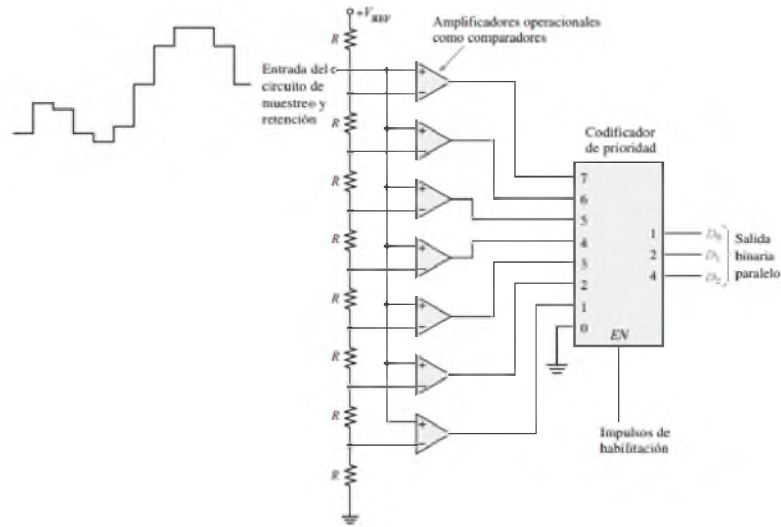


Figura 3.6 CAD flash de 3 dígitos.

Una ventaja de este CAD es su rápido periodo de conversión. Una desventaja es el gran número de comparadores que requiere para un número mayor de bits.

### 3.4.2 Convertidor Analógico-Digital de doble pendiente

El funcionamiento de este CAD consiste en utilizar un contador el cual se incrementa con respecto al tiempo. La figura 3.7 ilustra el diagrama de un CAD de doble pendiente.

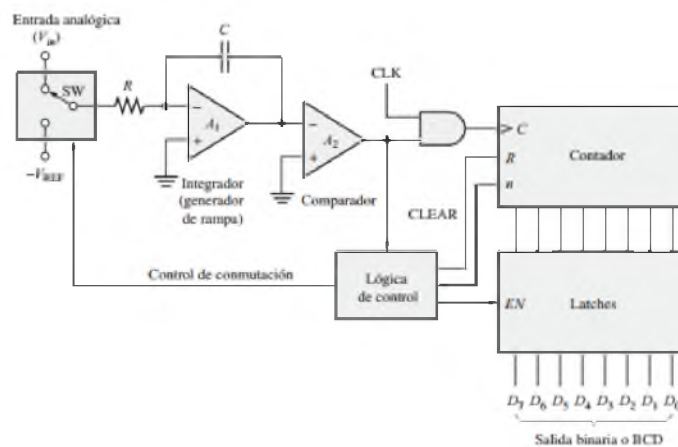


Figura 3.7 CAD de doble pendiente de 8 dígitos.

Una tensión constante en la entrada ( $V_{in}$ ) produce una corriente en la resistencia  $R$ , y en el condensador  $C$ . El condensador comenzará a cargarse lo que generará una rampa lineal negativa de tensión en  $A1$ . El contador incrementará hasta alcanzar un valor específico, reiniciando el contador y conmutando el *switch* hacia el voltaje de referencia ( $-V_{ref}$ ). En este instante el condensador estará cargado a un nivel de tensión negativa ( $-V$ ) proporcional a la tensión analógica de entrada. Posteriormente se genera una rampa positiva cuyo valor inicial es  $-V$  en consecuencia de la descarga del condensador. El contador avanzará a medida que el condensador se descargue. Cuando la tensión a la salida de  $A1$  sea 0, el conmutador de  $A2$  conmutará en estado bajo, lo que inhabilita el reloj del contador, concluyendo el ciclo de conversión. El valor del contador es almacenado y será proporcional a  $V_{in}$ .

### 3.4.3 Convertidor Analógico-Digital por aproximaciones sucesivas

Este CAD tiene un tiempo de conversión más rápido que el de doble pendiente pero más lenta que el tipo flash. Una ventaja que presenta es que el tiempo de conversión es el mismo sin importa la magnitud del voltaje de entrada. La figura 3.8 muestra un diagrama a bloques de un CAD por aproximaciones sucesivas de 4 dígitos.

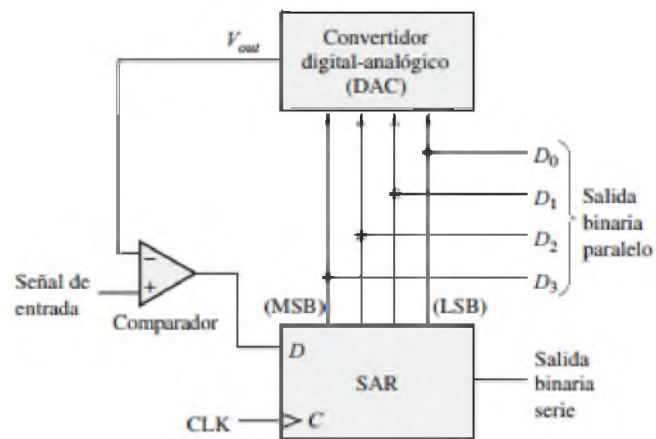


Figura 3.8 CAD por aproximaciones finitas de 4 dígitos.

Este tipo de convertidor emplea un convertidor digital-analógico, un registro de aproximaciones sucesivas (SAR) y un comparador. Los bits del CAD son habilitados de manera sucesiva, a partir del más significativo al menos significativo. Cada que se habilita un bit el comparador produce una señal que indica si el voltaje de entrada es mayor o menor que la salida del CDA. Si la salida del CDA es mayor el comparador se mantiene en estado 'bajo', haciendo que el bit en el registro permanezca en '0'. Si la salida es menor, el bit en el registro pasa a tener '1'. La operación se realiza con el bit más significativo (MSB), después con el segundo más significativo. El ciclo termina cuando todos los bits del CDA han sido aplicados.

### 3.5 Convertidor Digital- Analógico

La conversión digital-analógica es una parte importante de los sistemas de procesamiento de señales digitales. Algunas de las principales características de un Convertidor Digital-Analógico son las siguientes:

- Resolución. Es el número máximo de bits de salida digital. Por ejemplo un CDA de 4 bits, tiene una resolución de  $2^4$  (0 entre 15). [17]
- Precisión. Es la tolerancia de medida y define los límites de los errores cometidos [16]. Suele expresarse en términos de porcentaje de la tensión de salida. Por ejemplo si la salida máximo es de 10V y la precisión es  $\pm 0.1\%$ , entonces el error máximo para cualquier valor de salida será 10mV.

A continuación se describirán dos de los CDA más utilizados.

#### 3.5.1 Convertidor Digital-Analógico con ponderación binaria

Este método de conversión consiste en una red resistiva donde los valores de las resistencias equivalen a un número de los bits de entrada del código digital (las resistencias son múltiplos de  $2R$ ). En cada una de las resistencias puede o no circular corriente. La figura 3.9 muestra el diagrama de un CDA con ponderación binaria de 4 dígitos. El voltaje de salida será igual a

$$V_{out} = I_f R_f = \left( \frac{V}{R} + \frac{V}{2R} + \frac{V}{4R} + \frac{V}{8R} \right) R_f$$

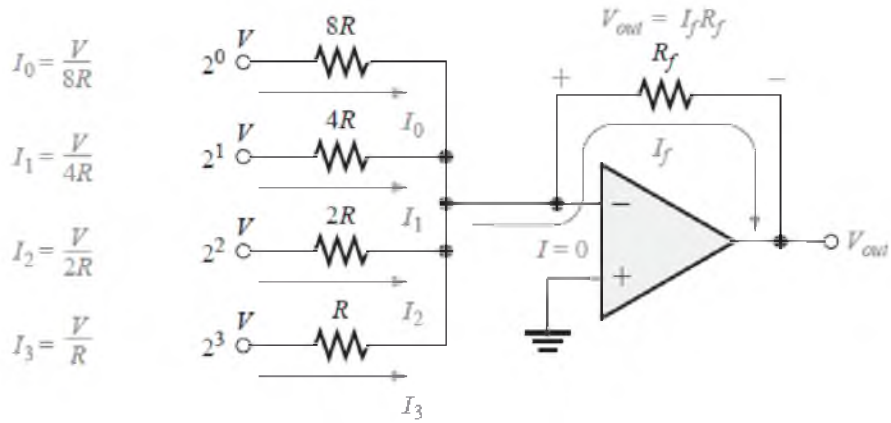


Figura 3.9 CDA con ponderación binaria.

Una desventaja de este tipo de convertidor es que el valor de las resistencias debe ser estrictamente múltiplo, por ejemplo un CDA de 8 bits requiere resistencias cuyo valor sean de R a 128R en pasos ponderados.

### 3.5.2 Convertidor Digital-Analógico en escalera R/2R

La desventaja del método anterior se resuelve con el de ponderación utilizando una red en escalera R/2R. En este método sólo se requieren dos valores de resistencias. La figura 3.10 ilustra un diagrama de este tipo de convertidor.

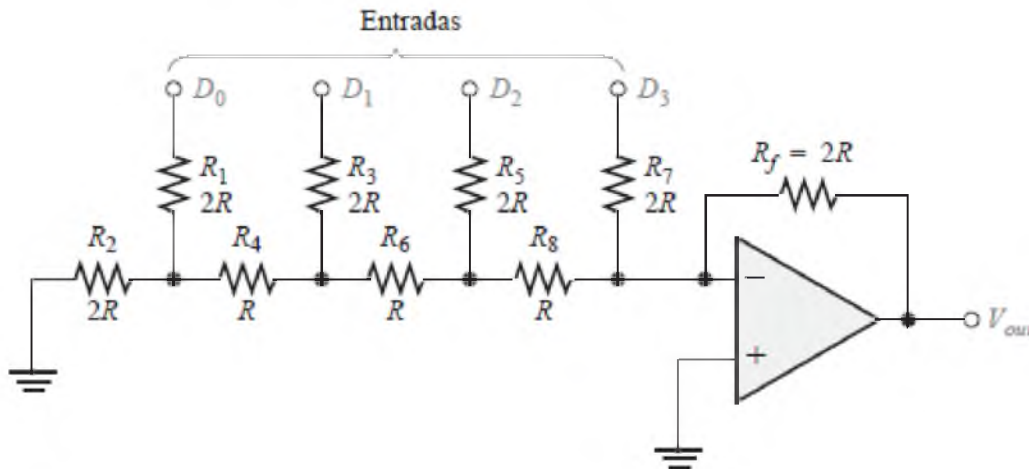


Figura 3.10 CDA en escalera R/2R

### 3.6 Protocolos de comunicación.

Los datos obtenidos en los sistemas de conversión deben ser enviados entre los diferentes dispositivos utilizados. Por lo tanto es necesario tener una forma (protocolo) de comunicación entre ellos, para garantizar un correcto envío y recepción de datos.

El protocolo SPI (del inglés *Serial Peripheral Interface*) consiste en la transmisión de bits por medio de la sincronización de un reloj de forma serial, esto es, bit por bit. El dispositivo que envía información recibe el nombre de maestro (*master*) y los dispositivos que reciben la información son nombrados esclavos (*slave*)

El protocolo SPI está constituido por cuatro señales:

- CLK. Es la señal de reloj que indica los pulsos de sincronización para la transmisión de datos.
- MOSI. Señal que va del maestro al esclavo.
- MISO. Señal que va del esclavo al maestro.
- SS. Esta señal es usada para la activación del protocolo o la selección de un esclavo (en caso de tener más de uno).

La figura 3.11 muestra el diagrama para un sistema de comunicación SPI de un maestro y tres esclavos.

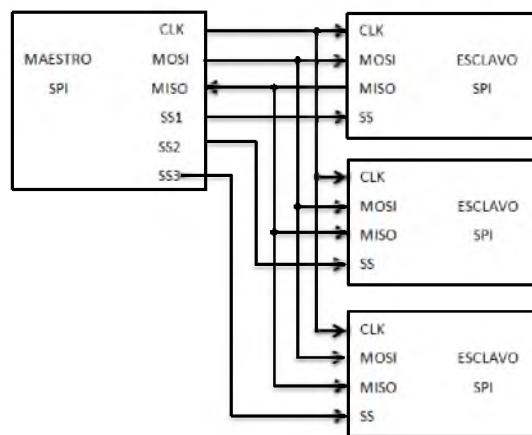


Figura 3.11 Sistema SPI para un maestro y tres esclavos.

# Capítulo 4: Diseño del módulo en VHDL

En este capítulo se presenta el diseño para un sistema de control PID digital en un FPGA utilizando VHDL. El sistema diseñado consta de tres partes; la interface SPI (Habilita de periféricos CDA, CDA y Pre-Amplificador), la interface de usuario y el módulo de control.

Existen numerosas plataformas para el diseño de circuitos lógicos en FPGA. Una alternativa en el mercado es la tarjeta Spartan 3E, la cual cuenta con un precio a la fecha de \$1937.00. La figura 4.1 muestra una tarjeta Spartan 3E desde una vista superior.



Figura 4.1 Spartan 3E desde una vista superior.

Algunos de los componentes de la Spartan 3E starter kit board son:

- FPGA de Xilinx XC3500E Spartan.
- Pantalla LCD 16x2.
- Puerto PS/2.
- Oscilador de 50 MHz.
- 2 canales de entrada para convertidor analógico/digital (CAD) por comunicación SPI con pre-amplificador de ganancias programables.
- 4 salidas para convertidor digital/analógico (CDA) por comunicación SPI.
- Botones rotacionales con encoder.
- 8 Leds discretos.

Esta tarjeta cuenta con los periféricos necesarios para este proyecto.

## 4.2 Interface SPI

La estructura interna de la interface SPI consiste básicamente en 4 estados: El estado “INICIO” el cual deshabilita todos los periféricos a fin de que durante su configuración no se produzca una activación no deseada. El estado “AMP” configura la ganancia del Pre-Amplificador. Este dispositivo requiere ser configurado una sola vez. Posteriormente se tiene una secuencia infinita entre el estado “CAD” y el estado “CDA”. Es importante resaltar que sólo puede estar activo un dispositivo a la vez, ya que ocupan el mismo bus de datos. La figura 4.2 muestra la relación entre los 4 estados del módulo SPI.

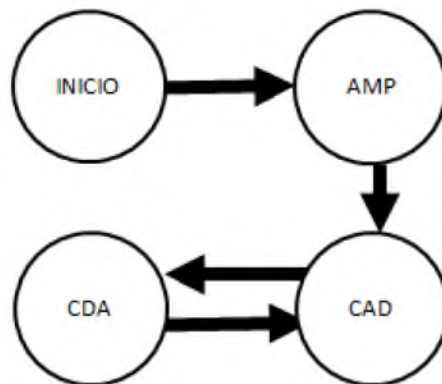


Figura 4.2 Comportamiento de la interface SPI.

## Circuito de captura analógica

La tarjeta spartan 3E tiene integrado dos componentes que son utilizados para aplicaciones donde se requiere usar señales analógicas. El CDA LT1407A y el Pre-Amplificador de ganancia programable LT6912. Un amplificador de voltaje es un dispositivo que amplifica el voltaje en su entrada por una ganancia determinada. La ganancia del amplificador LT6912 es programada por el usuario, y su selección está en relación al rango de voltajes de la aplicación. En este circuito el FPGA es el encargado de activar y desactivar los dispositivos (*maestro*). El circuito de captura analógica se ilustra en la figura 4.3.

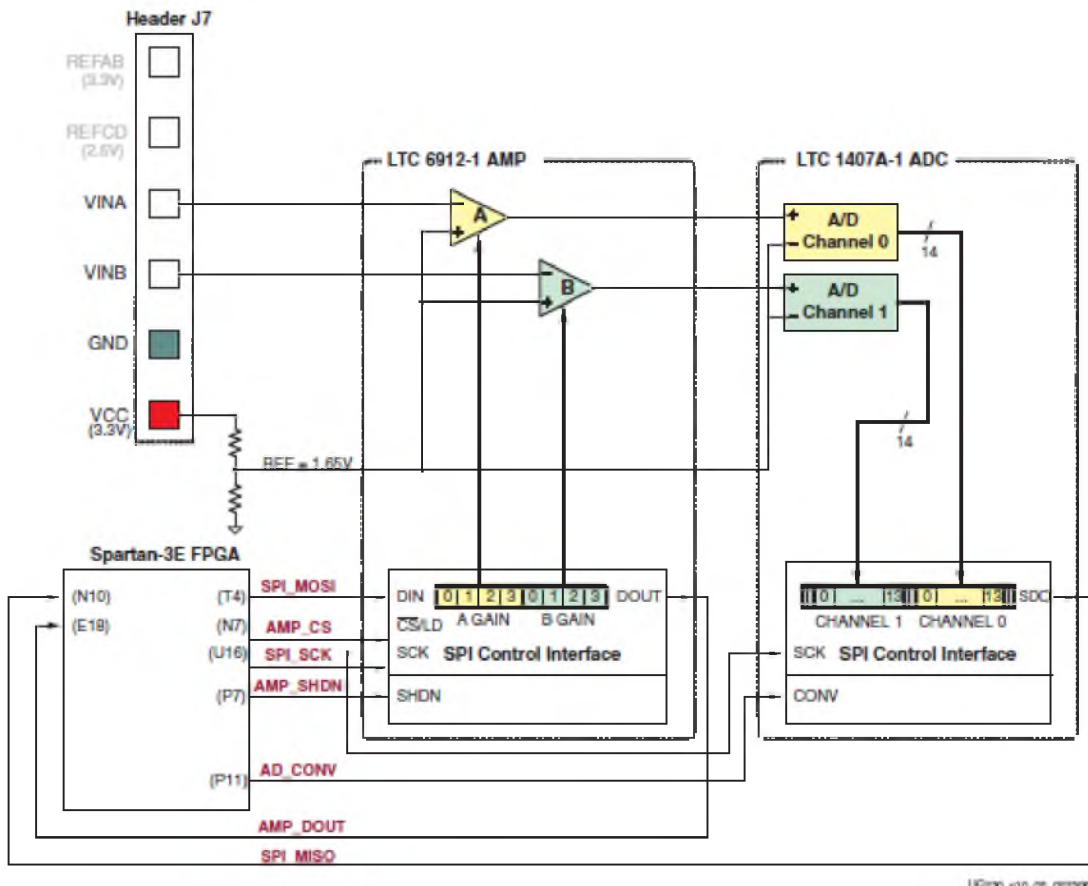


Figura 4.3 Circuito de captura analógica [10]

## Pre-Amplificador

Los valores seleccionados para la ganancia deben estar en relación al voltaje de entrada. Debido a que en este diseño el rango máximo de voltaje es de 0.4V a 2.9V, se configuró con el valor “0001” equivalente a una ganancia de “-1”. La siguiente tabla muestra las diferentes configuraciones para el pre-amplificador.

Ganancia	Configuración	Voltaje mínimo	Voltaje máximo
0	“0000”		
-1	“0001”	0.4V	2.9V
-2	“0010”	1.025V	2.275V
-5	“0011”	1.4V	1.9V
-10	“0100”	1.525V	1.775V
-20	“0101”	1.5875V	1.7125V
-50	“0110”	1.625V	1.675V
-100	“0111”	1.6375V	1.6625V

### Configuraciones de la ganancia del Pre-Amplificador

Las señales que permiten la comunicación de datos entre los dispositivos son utilizadas por el FPGA en su función de maestro. A continuación se indica el funcionamiento de cada una de estas señales.

Señal	FPGA pin	Dirección	Descripción
SPI MOSI	T4	FPGA-> CAD	Datos en serie para la el Pre-Amplificador
AMP_CS	N7	FPGA-> AMP	Bit de selección. Activo en bajo
SPI_SCK	U16	FPGA-> AMP	Señal de reloj
AMP_SHDN	P7	FPGA<-AMP	Reset
AMP_DOUT	E18	FPGA<-AMP	Datos en serie. Configuración

### Señales que se involucran en el protocolo de comunicación

En el protocolo SPI empleado por los dispositivos, el reloj indica el orden y la velocidad de transmisión de cada dato. El Pre-Amplificador consta de dos canales de recepción

(Amplificador A, Amplificador B), cada uno puede programarse con una ganancia independiente. Este protocolo consta de 8 bits, 4 bits determinan la ganancia del amplificador A y los otro 4 bits los del amplificador B. En la figura 4.4 se ilustra la secuencia de bits requerida para el protocolo.

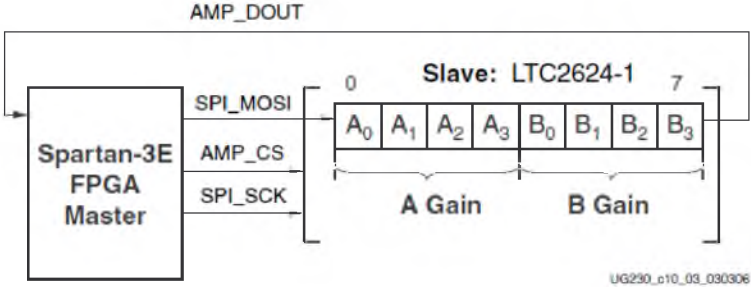


Figura 4.4 Protocolo de comunicación del Pre-Amplificador.

En el diseño propuesto para el PID sólo se requiere de un canal para la conversión (se seleccionó el canal 'A'). La palabra de 8 bits con la cual se configuró el Pre-Amplificador debe ser "10000000". Al enviar "0000" el canal B queda desabilitado.

En la entidad destinada a la interface SPI se debe considerar el diagrama de tiempos de cada dispositivo. En este diagrama se especifica el orden en que se interpretan las señales. La figura 4.5 se presenta el diagrama de tiempos necesario para el protocolo del Pre-Amplificador.

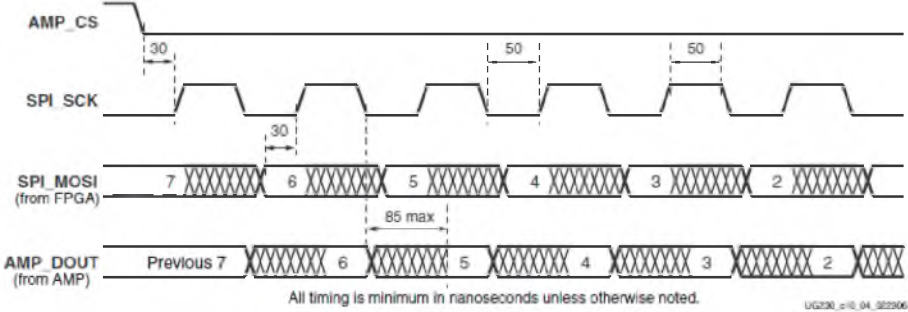


Figura 4.5 Diagrama de tiempos del Pre-Amplificador

## Convertidor Analógico-Digital (CAD)

De acuerdo con el circuito de captura analógica, la salida del Pre-Amplificador está conectada al CAD LT1407, el cual provee de dos canales (canal 0, canal 1) de 14 bits para la conversión. El rango máximo del CAD es de  $\pm 1.25V$ , centrado en  $1.65V$ , es decir el voltaje de trabajo de este dispositivo es de  $0.4V$  a  $2.9V$ . La representación obtenida está basada en la siguiente ecuación.

$$D[13:0] = Ganancia * \frac{(V_{in}-1.65V)}{1.25V} * 8192,$$

donde *Ganancia* es la ganancia programada previamente en el Pre-Amplificador,  $V_{in}$  es el Voltaje analógico en la entrada y  $D[13:0]$  es el vector en donde se guardará la representación de  $V_{in}$ .

De la misma manera que para el Pre-Amplificador, es necesario emplear señales para la transferencia de información entre el FPGA y el CAD. Dichas señales se presentan y describen en la siguiente tabla.

Señal	FPGA pin	Dirección	Descripción
SPI_SCK	U16	FPGA->CAD	Señal de reloj
AD_CONV	P11	FPGA->CAD	Bit para inicio de conversión. Activo en alto
SPI_MISO	N10	FPGA<-CAD	Datos en serie. Presenta la representación digital del muestreo de un valor analógico en complemento a dos.

### Señales involucradas en el ADC.

El protocolo SPI de comunicación requerido para el CAD consta de 34 ciclos de reloj, de los cuales los primeros 17 están destinados para la conversión del canal 0 y los restantes para el canal 1 (figura 4.6). La señal AD\_CONV da inicio a la conversión y los bits de D0 a D13 se mostraran en la señal SPI\_MISO en cada ciclo de reloj.

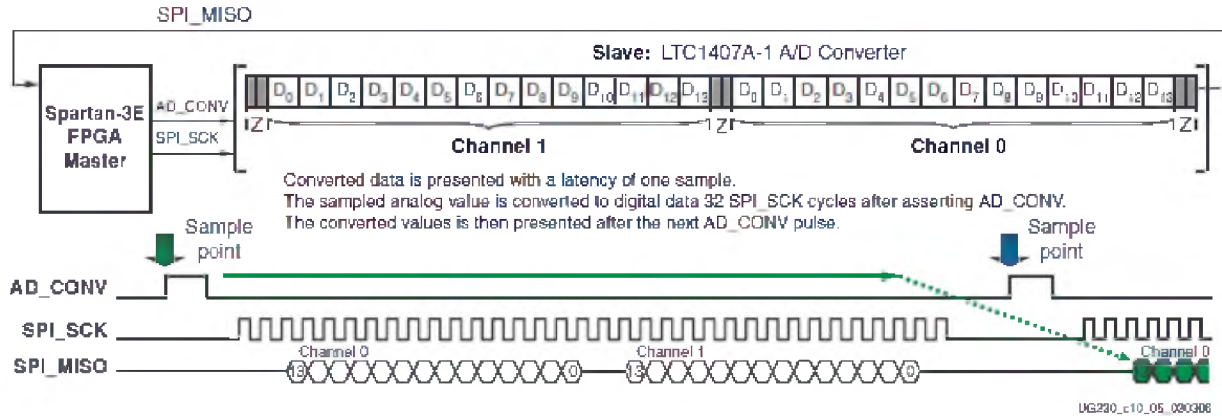


Figura 4.6 Proceso de conversión de un valor analógico.

Para el diseño de este sistema se hará caso omiso del canal 1, pero es necesario que se cumplan ambas conversiones para volver a convertir los voltajes a la entrada.

### Circuito de conversión digital (CDA)

El CDA incorporado en la tarjeta spartan 3E consta de 4 canales. El voltaje de referencia para los canales A y B es de 3.3V, y para el canal C y D es de 2.5V. La figura 4.7 presenta el diagrama del CDA en relación con el FPGA.

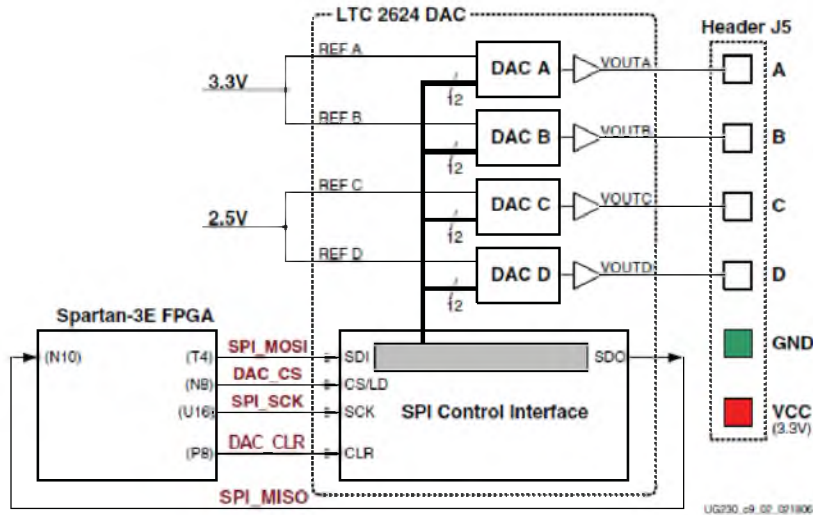


Figura 4.7 Diagrama del circuito del CDA.

El CDA funciona bajo dos esquemas; uno de 24 bits y otro de 32 bits. Se optó por el esquema de 24 bits, debido a que el tiempo de conversión en esta aplicación es importante, esto es porque al utilizar el esquema de 24 bits se reduce la conversión en 8 ciclos de reloj en relación al de 32 bits. El orden de la transferencia de los bits se muestra en la figura 4.8.

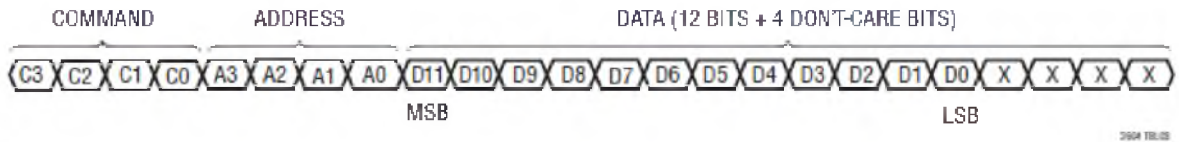


Figura 4.8 Protocolo de 24 bits para el CDA.

Existen tres conjuntos de bits que intervienen en el CDA: Los bits de COMMAND, ADDRESS y DATA. Los bits de COMMAND son los que determinan la acción que ejecutará el CDA. La tabla 4.4 indica las distintas combinaciones y acciones para el CDA.

Configuración	Descripción
0001	Actualiza el canal seleccionado
0011	Escribir y actualizar el canal seleccionado
0100	Apagar canal 'n'
1111	No hay operación

#### Configuraciones para los bits COMMAND.

La configuración COMMAND para este diseño es "0011", es decir, escribir y actualizar el canal seleccionado en el mismo proceso de conversión. Los bits de "ADDRESS" seleccionan el canal (o canales) que se activa para el convertidor. En la siguiente tabla se muestran las combinaciones requeridas para cada configuración.

Configuración	Canal
0000	CDA A
0001	CDA B
0010	CDA C
0011	CDA D
1111	Todos

#### Configuraciones de ADDRESS

Como sólo se requiere un canal, la configuración utilizada es “0000” (para seleccionar el canal A pero pudo haberse utilizado cualquier otro)

Los bits DATA determinan la magnitud del voltaje de salida del convertidor. Para determinar el valor, se aplica la siguiente formula:

$$V_{out} = \frac{D[11:0]}{4096} * V_{referencia},$$

donde  $V_{out}$  es el voltaje de salida del CDA,  $D[11:0]$  es un vector de 12 bits que determina el valor del voltaje de salida que se desea obtener, y  $V_{referencia}$  es voltaje de referencia fijo para ese canal. De acuerdo con el voltaje de referencia del canal A, el rango de voltajes que se tiene será de 0V a 3.3V.

### Módulo VHDL para la interface de SPI

Se diseñó el módulo para la interface SPI en forma de una máquina de estados finitos. Internamente se cuentan los ciclos de reloj requeridos para el protocolo de cada dispositivo (CAD, CDA y Pre-Amplificador), de igual manera se hace una condicional para tener activado únicamente el dispositivo deseado. La figura 4.9 muestra el diseño de la entidad del módulo SPI. [11]

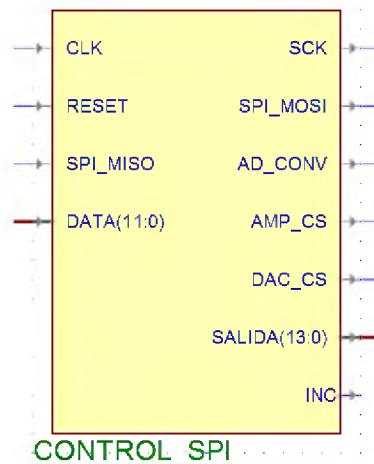


Figura 4.9 Entidad del módulo SPI

La máquina de estados finitos programada en el control de los dispositivos SPI se muestra en la siguiente figura:

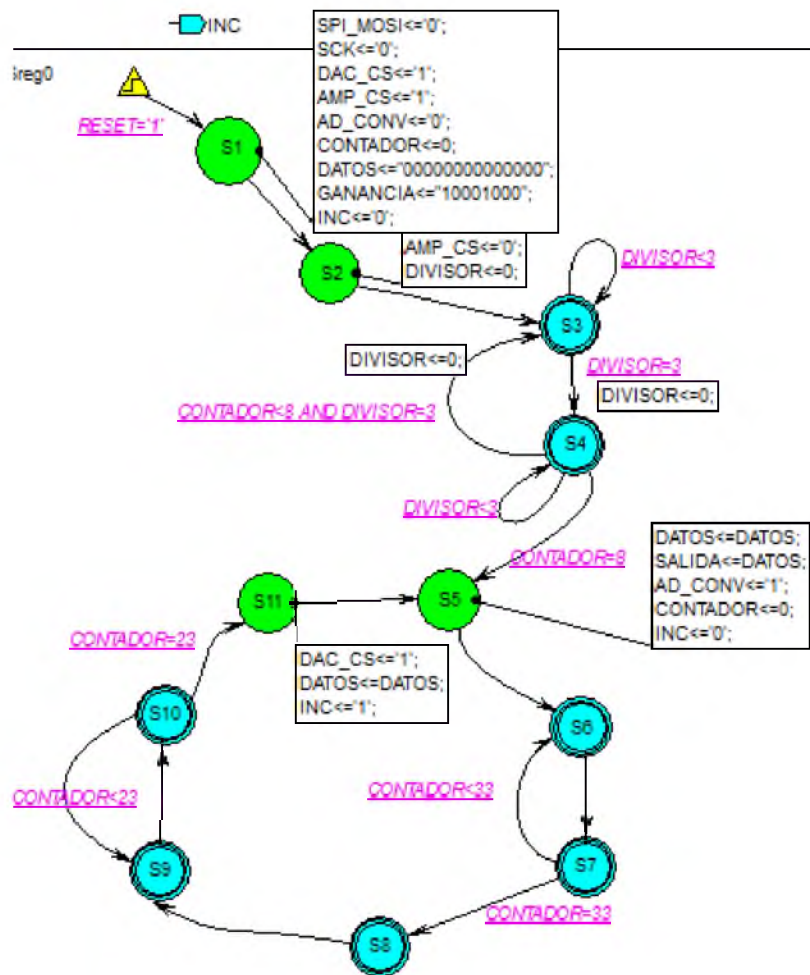


Figura 4.10 Diseño de la máquina de estados finitos para los dispositivos SPI.

En el estado S1 se espera a la activación de los dispositivos SPI (CAD, CDA y el Pre-Amplificador). Durante el estado S2 se activa el Pre-Amplificador (y se mantienen desactivados el CDA y el CAD). Los estados S3 y S4, contienen el protocolo de comunicación requerido para configurar el Pre-Amplificador. Se utiliza un contador que determina el número de bits que se han enviado al dispositivo. Cuando el contador llega a 8, el protocolo llega a su fin pasando al estado siguiente (S5) y desactivando el Pre-Amplificador. El estado S5 activa el CAD y pone en cero el contador. Los estados S6 y S7 contienen el protocolo para el CAD. Este consta de 66 ciclos de reloj, 33 para S6 y 33 para

S7. El estado S6 captura el dato de la conversión correspondiente al ciclo de reloj. Cuando el protocolo llega a su fin se desactiva el CAD. En S8 se activa el CDA reestableciendo el contador y las variables empleadas para el protocolo. Los estados S9 y S10 ejecutan el protocolo para la comunicación del CDA el cual consta de 48 ciclos de reloj (24 ciclos para el estado S9 y 24 para el estado S10). El último estado S11 se encarga de desactivar el CDA y reestablecer las variables para iniciar con el nuevo ciclo de conversión.

### 4.3 Interface de usuario

La interface de usuario tiene como principal objetivo ser un medio por el cual se pueda ingresar la referencia deseada y mostrar información desde el sistema. Para este fin el encoder y los botones se utilizan como medio para ingresar la información del controlador y de la referencia; el LCD despliega los datos del CAD, el error del sistema y la referencia.

#### Pantalla de cristal líquido (LCD)

Un LCD (*liquid crystal display*) es un dispositivo utilizado en sistemas electrónicos para la presentación de datos. El LCD de la Spartan 3E contiene 2 renglones y es capaz de representar 16 caracteres por renglón. Este dispositivo es controlado con una interface de 7 bits. La figura 4.11 muestra el diagrama de conexión del FPGA y el LCD. [12]

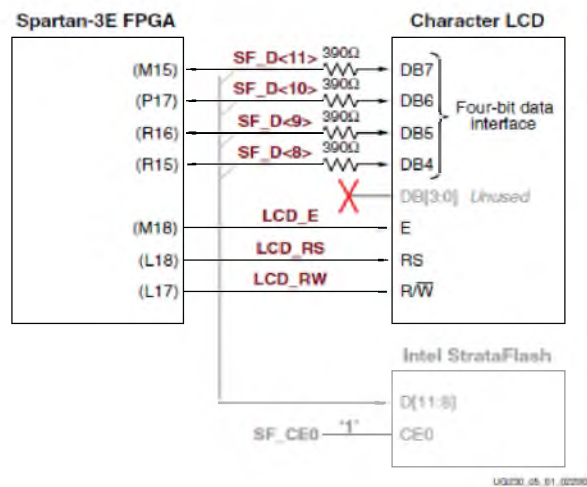


Figura 4.11 Diagrama de conexión entre la FPGA y el LCD

El módulo del LCD recibirá del FPGA las cadenas de bits necesarios para la inicialización y configuración. La siguiente tabla describe las señales requeridas para el uso de este LCD.

Nombre de la señal	Pin	Función	
SF_D <11>	M15	Data bit DB7	Compartida con los pines Strataflash SF_D <11:8>
SF_D <10>	P7	Data bit DB6	
SF_D <9>	R16	Data bit DB5	
SF_D <8>	R15	Data bit DB4	
LCD_E	M18	Pulso de habilitación de lectura/escritura 0 Deshabilitado 1 Habilitador para la operación de Lectura/Escritura	
LCD_RS	L18	Selección de registro 0 Registro de instrucción durante operaciones de escritura. 1 Datos para operaciones de lectura o escritura	
LCD_RW	L17	Control de lectura/escritura 0 Escritura, El LCD acepta datos 1 Lectura, El LCD manda datos	

Señales involucradas en el módulo del LCD

### Encoder

El principio básico de este encoder es el de una leva conectada a un eje que se utiliza para operar 2 interruptores. Cuando el encoder se ubica en una posición estacionaria ambos interruptores permanecen cerrados. En la siguiente figura se muestra el diagrama desde una vista superior.

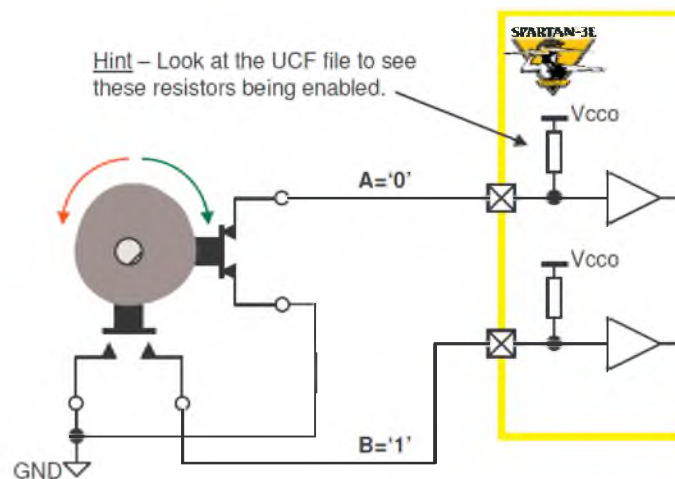
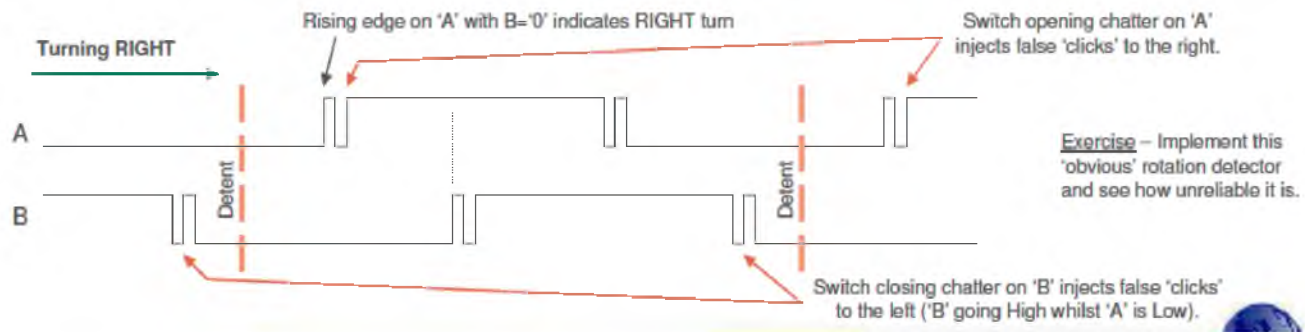


Figura 4.12 Diagrama del encoder

La leva no se ubica centrada, por lo tanto, dependiendo la dirección de giro se activará primero el interruptor A (si el giro es a la derecha) o el interruptor B (si el giro es a la izquierda). Si la rotación es continua, un interruptor será cerrado antes que el otro. La figura 4.14 muestra el comportamiento de las señales lógicas con respecto al giro hacia la derecha del encoder.



4.13 Comportamiento de las señales A y B del encoder

Debido a que este sistema mecánico está propenso a rebotes, los interruptores envían señales con ruido. Una manera de eliminar este efecto es mediante un filtro. Esto se logra con la detección de un sólo primer cambio de la señal y haciendo caso omiso a toda actividad posterior sobre la misma señal hasta que el otro interruptor también cambie de estado.

La siguiente figura muestra una comparación entre el comportamiento de ambas señales, antes y después del filtro.

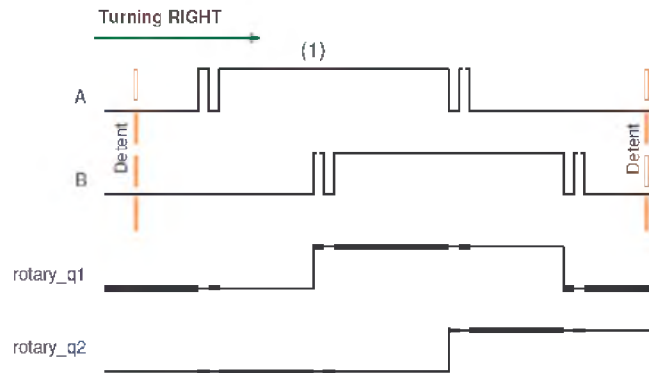


Figura 4.14 Comparación entre las señales antes y después del filtro.

La señal 'A' y 'B' son las señales originales que proporciona el encoder. Las señales rotary\_q1 y rotary\_q2 son las señales proporcionadas a partir del filtro de la señal. Las partes sombreadas representa la retención de la señal que hace el filtro. Aunque la señal que se obtiene cuenta con un retraso su comportamiento más estable.

En este filtro la señal rotary\_q1 es igual a '1' cuando A está en '1' y B también está en '1' y rotary\_q1 será '0' cuando A está en '0' y B en '0', recordando el estado actual en cualquier otra condición (realizando la función XNOR entre las señales A y B). En la señal rotary\_q2 se establece un '1' cuando la señal A está en estado '0' y B en estado '1', finalmente rotary\_q2 es igual a '0' cuando A está en '1' y B es igual a '0' (realizando la función XOR entre las señales A y B). [13]

### Módulo en VHDL para la interface de usuario

El diseño para la interface de usuario consta de 4 entidades: la primera es la entidad de codificación de la representación del voltaje de entrada, la segunda es la entidad con el filtro del encoder la tercera es la entidad para el cálculo del error y finalmente la cuarta entidad realiza la configuración del LCD.

En la siguiente figura se muestra el diagrama de la entidad de codificación de la representación del voltaje de entrada.

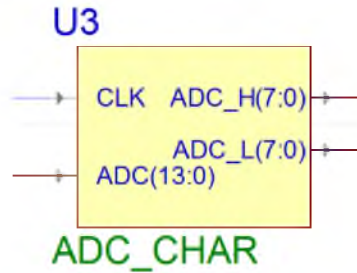


Figura 4.15 Entidad para la codificación de la representación del voltaje de entrada

El CAD de la tarjeta spartan 3E entrega el valor del voltaje de entrada en complemento a 2, el cual es un formato que se utiliza para representar números negativos en binario. Esto es muy útil cuando se requiere trabajar con valores de entrada negativos, pero como en nuestro sistema sólo se requieren valores positivos se debe programar una sentencia que convierta el valor de complemento a 2 a su valor absoluto. Por medio de esta operación es posible utilizar los 14 bits para la representación analógica, es decir, la representación del voltaje analógico es de 16383 valores ( $2^{14}$ ). Este proceso se presenta en la figura 4.16.

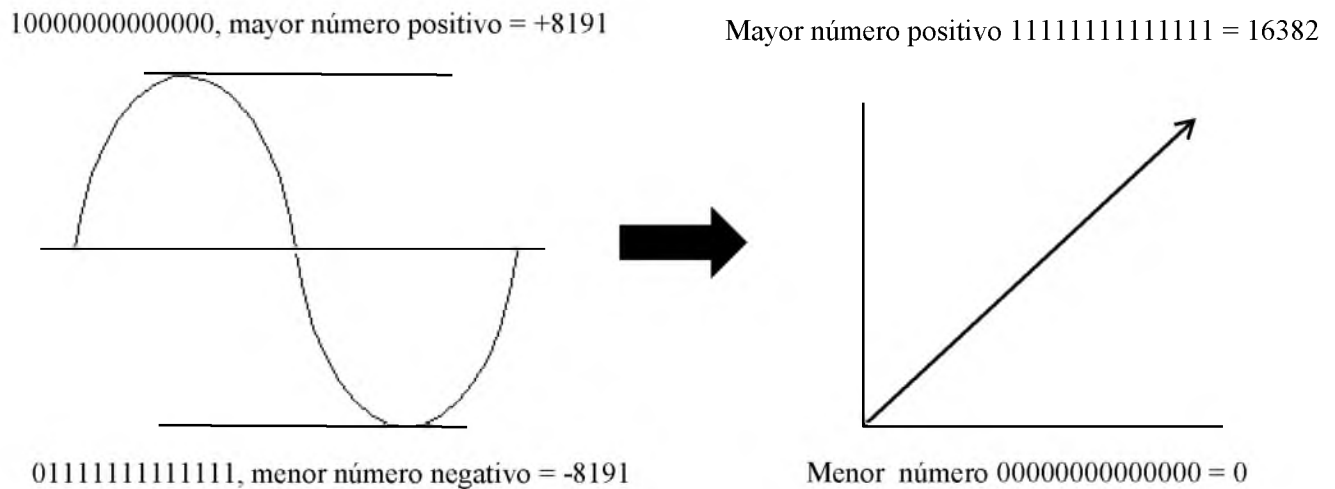


Figura 4.16 Representación del voltaje de entrada

La codificación consiste en aplicar un inversor a todos los valores del voltaje de entrada excepto al más significativo. La figura 4.17 muestra la descripción de hardware utilizada para codificar el complemento a 2 en un valor absoluto.

```

DATOS<="000000000000000";
DATOS(13)<= ADC(13);
DATOS(12)<= NOT ADC(12);
DATOS(11)<=NOT ADC(11);
DATOS(10)<=NOT ADC(10);
DATOS(9)<=NOT ADC(9);
DATOS(8)<=NOT ADC(8);
DATOS(7)<=NOT ADC(7);
DATOS(6)<=NOT ADC(6);
DATOS(5)<=NOT ADC(5);
DATOS(4)<=NOT ADC(4);
DATOS(3)<=NOT ADC(3);
DATOS(2)<=NOT ADC(2);
DATOS(1)<=NOT ADC(1);
DATOS(0)<=NOT ADC(0);

```

Figura 4.17 Código empleado para la representación del voltaje de entrada

El valor absoluto se divide en dos partes: la parte alta que consta de 8 bits representando la parte entera y la parte baja que consta de 8 bits representando la parte fraccionaria del voltaje de entrada.

La entidad destinada al control del encoder se basa en el diagrama de la figura 4.18. Esta entidad debe obtener una representación de la referencia que siga el mismo formato que el CAD, es decir, 8 bits para la parte entera y 8 para la parte fraccionaria.

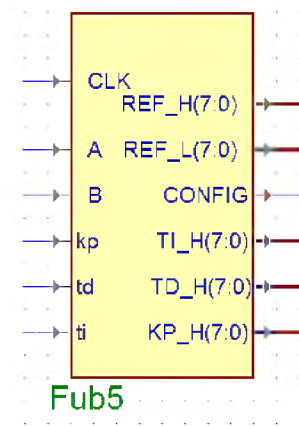


Figura 4.18 Entidad destinada al control del encoder.

En esta entidad también se programó el filtro requerido para mejorar la señal del encoder (figura 4.19). [13]

```
rotary_filter: process(clk)
begin
  if clk'event and clk='1' then

    rotary_in <= rotary_b_in & rotary_a_in;

    case rotary_in is

      when "00" => rotary_q1 <= '0';
                   rotary_q2 <= rotary_q2;

      when "01" => rotary_q1 <= rotary_q1;
                   rotary_q2 <= '0';

      when "10" => rotary_q1 <= rotary_q1;
                   rotary_q2 <= '1';

      when "11" => rotary_q1 <= '1';
                   rotary_q2 <= rotary_q2;

      when others => rotary_q1 <= rotary_q1;
                    rotary_q2 <= rotary_q2;

    end case;

  end if;
end process rotary_filter;
```

Figura 4.19 Filtro aplicado al encoder.

A partir de la representación del voltaje de entrada y de la referencia se puede obtener el error del sistema. Se diseñó una entidad destinada al cálculo del error. Para esta operación se realiza una resta entre el valor de referencia y el valor del voltaje de entrada, obteniendo así la parte entera, la parte fraccionaria y el signo del error (figura 4.20).

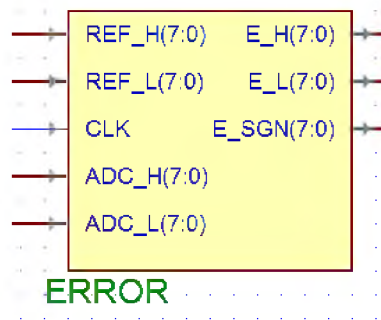


Figura 4.20 Entidad de error del sistema.

Finalmente los valores de referencia, error y la representación del voltaje de entrada son enviadas a una entidad que mostrará estos valores en el LCD. El funcionamiento de esta

entidad está basada en la configuración que provee el manual de usuario de la Spartan 3E (Figura 4.21).

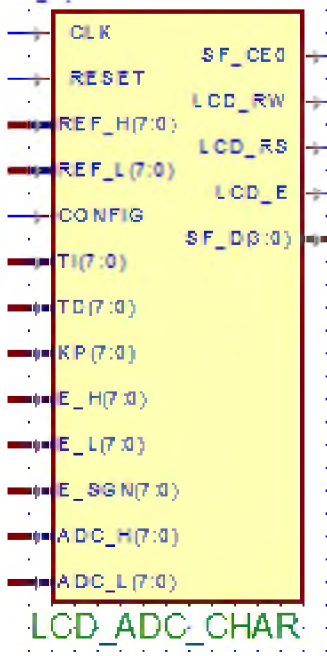


Figura 4.21 Entidad de la interface de usuario

### 4.3 Módulo de control

Para el diseño de este módulo se hace la suposición que el usuario cuenta con un diseño para un controlador proporcional-integral-derivativo en el tiempo discreto, es decir, se conocen los coeficientes numéricos en la siguiente función de transferencia de pulso.

$$\frac{U(z)}{E(z)} = Kp \left[ 1 + \frac{T}{T_i(1 - z^{-1})} + T_d \frac{(1 - z^{-1})}{T} \right]$$

Si definimos

$$a = Kp$$

$$b = \frac{Kp * T}{T_i}$$

$$c = \frac{kp * T_d}{T}$$

Entonces, la función de transferencia de pulso tiene la siguiente forma

$$\frac{U(z)}{E(z)} = a + \frac{b}{(1 - z^{-1})} + c(1 - z^{-1})$$

Existen distintas formas de construir una realización para un controlador PID. En general la realización de funciones de transferencia de pulso significa determinar la configuración física para la combinación apropiada de las operaciones aritméticas y de almacenamiento [14]. Como en este caso se va a realizar una descripción de hardware, la configuración más adecuada del controlador se encuentra en la figura 4.22.

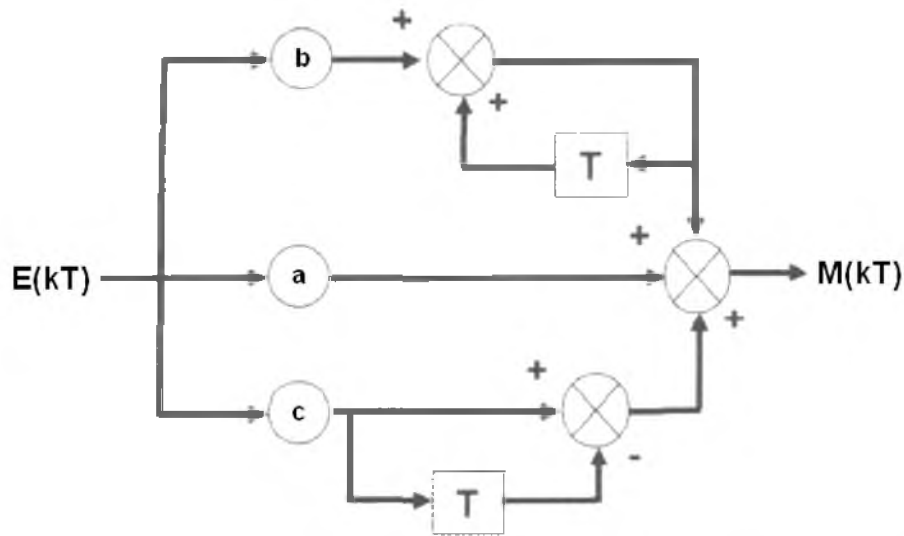


Figura 4.22 Configuración en paralelo para un controlador PID

El módulo de control consiste en dos entidades: Fub9 (figura 4.23) se encarga de convertir el error en formato de 16 bits (8 para la parte entera y 8 para la parte fraccionaria) a un formato de 12 bits (ya que la salida del sistema es el CDA de 12 bits), y la entidad PID (figura 4.23) que contiene programada la ley de control.



Figura 4.23 Entidad Fub9

El formato para la composición del vector es de *punto fijo* es el siguiente. Los 4 bits más significatos representan la parte entera y los 8 bits menos significativos representan la parte fraccionario tal se muestra en la figura 4.24.

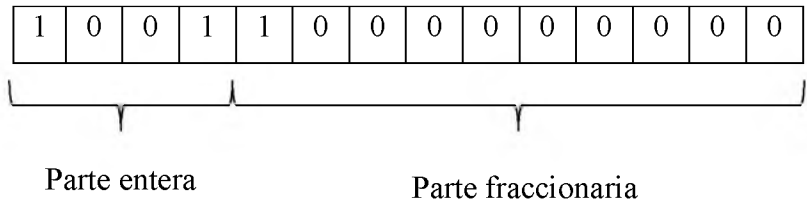


Figura 4.24 Bits destinados para la parte entera y parte fraccionaria del error.

La entidad PID contiene una ley de control tipo PID mencionado anteriormente. La salida de esta entidad está conectada a la interface SPI que se encarga de habilitar el CDA. Los 12 bits que indican el voltaje de salida serán calculados en esta entidad.

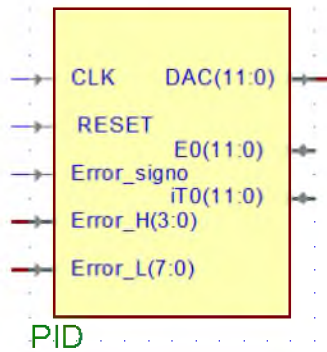


Figura 4.25 Entidad PID

Se eligió una programación secuencial en esta entidad, empleando variables que se actualizan en cada sentencia y no en sincronización con el reloj. Esto facilita y ahorra procesos ya que las variables pueden ser utilizadas para calcular los términos. La figura 4.26 ilustra el código empleado para el controlador PID. [15]

		<code>Error:=Error_H&amp;Error_L;</code>
		<code>iT_doble:=b*Error;</code>
Término integral	{	<code>iT:=iT_doble(17 downto 6);</code>
		<code>iT:=iT+iT0;</code>
Término derivativo	{	<code>dT_doble:=c*(Error-E0);</code>
		<code>dT:=dT_doble(17 downto 6);</code>
Término proporcional	{	<code>Kp_doble:=a*Error;</code>
		<code>Kp:=Kp_doble(17 downto 6);</code>
		<code>uT:=Kp+dT+iT;</code>
		<code>iT0&lt;=iT;</code>
		<code>E0&lt;=Error;</code>

Figura 4.26 Código de la ley de control PID.

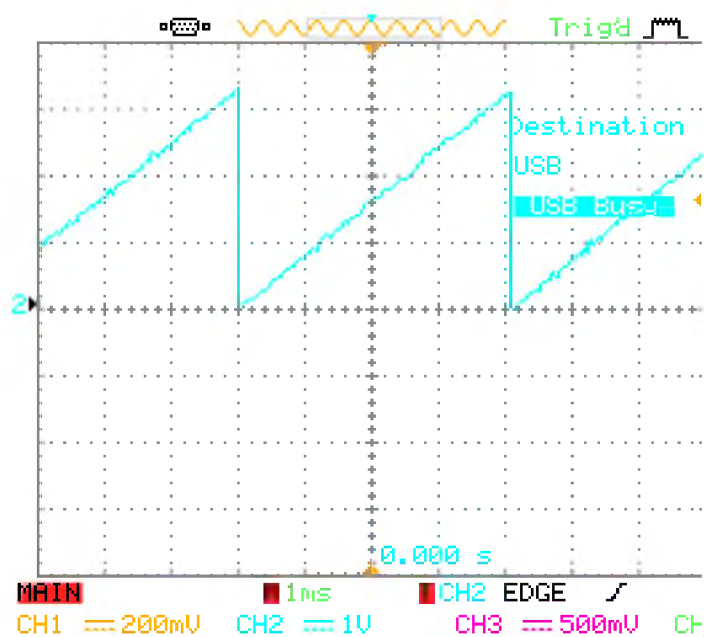
Primero se realiza una concatenación para tener el error en un sólo vector de bits. Posteriormente se realiza el cálculo de los términos integral, derivativo y proporcional. Finalmente se suman los 3 términos, se suman y se envían a la interface SPI.

# Capítulo 5: Verificación del módulo en VHDL

En este capítulo se verificará el funcionamiento de los módulos programados en el capítulo 4. El software de síntesis fue el *ISE Desing tools 14.7* de la compañía de Xilinx. La verificación consistió en 3 partes: La verificación de los periféricos SPI, la interface de usuario y del módulo de control.

## 5.1 Implementación del módulo SPI

Para la verificación de este módulo se funcionamiento del CDA generando una salida de voltaje tipo rampa a través del canal A, tal como se muestra en la figura 5.1. Se implementó un contador de 12 bits que en cada conversión incrementaba y al llegar al máximo (12 bits) se reiniciaba al valor 0. El contador escribía los datos de conteo en los bits de DATA del CDA y esto da como resultado la señal tipo rampa



5.1 Funcionamiento del CDA.

Para la implementación del CAD y CDA juntos, se utilizó un generador de funciones, un osciloscopio y un multímetro. El generador de funciones se configuró para proporcionar una señal tipo seno con una amplitud de 2 Volts y una frecuencia de 5 Hz. La figura 5.2 muestra la verificación del módulo SPI. La señal proporcionada por el generador de funciones se envía al dispositivo CAD y posteriormente el módulo obtiene la representación binaria del voltaje de entrada. Esta representación es enviada al CDA para que la señal de salida proporcional a la señal de entrada.

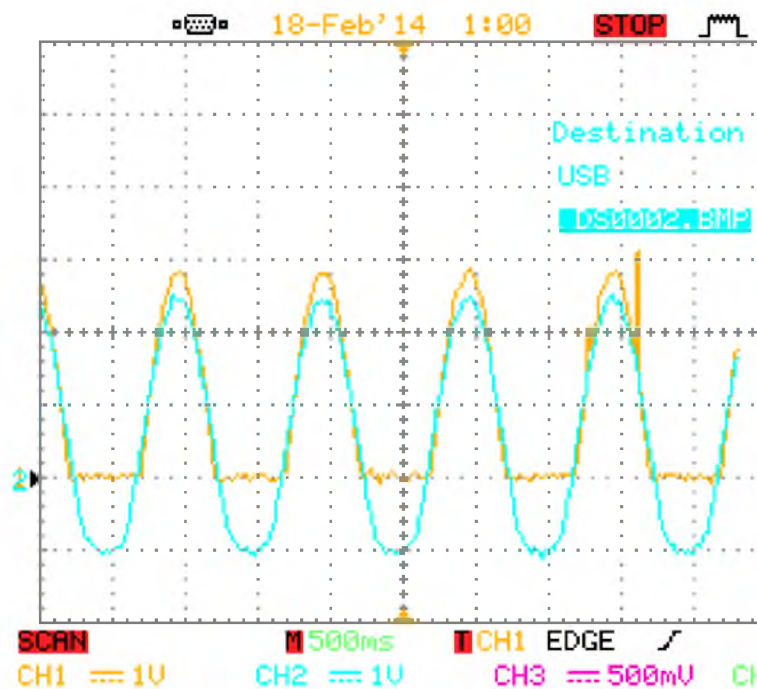


Figura 5.2 Funcionamiento del módulo SPI

La señal de entrada es mostrada en color azul, y la señal de convertidor CDA es la mostrada en color anaranjado. Se puede apreciar que la señal de CDA no llega a estar a cero, como la señal de entrada, esto se debe a que el rango mínimo de entrada al CAD es de 0.4V. Por lo tanto, los voltajes inferiores a 0.4V no pueden ser convertidos.

## 5.2 Verificación del módulo de usuario

La implementación del módulo de usuario consiste en la visualización de los valores de la referencia (ref), estado actual (act) y error del sistema (error) en el LCD. En la figura 5.3 se muestra una fotografía del funcionamiento del LCD con el código implementado.



Figura 5.3 Implementación del módulo de usuario

Los valores correspondientes omiten una entrada en el CAD, mostrando el valor de referencia 1.6V (que en la codificación realizada muestra 5.0V ~ 5.1V) posteriormente la referencia inicial es de 5 pero se puede apreciar que el encoder ha sido rotado lo que varía la referencia hasta el valor de 4.7. Finalmente el error correspondiente es de -0.4 (4.7 - 5.1).

### 5.3 Verificación del módulo de control

El módulo de control se verificó por medio de la simulación con un error constante en el sistema. El error supuesto es de 7.0 (0111 00000000) desde t=0 hasta el término de la simulación. En la siguiente figura se muestra el resultado del procesamiento del error.

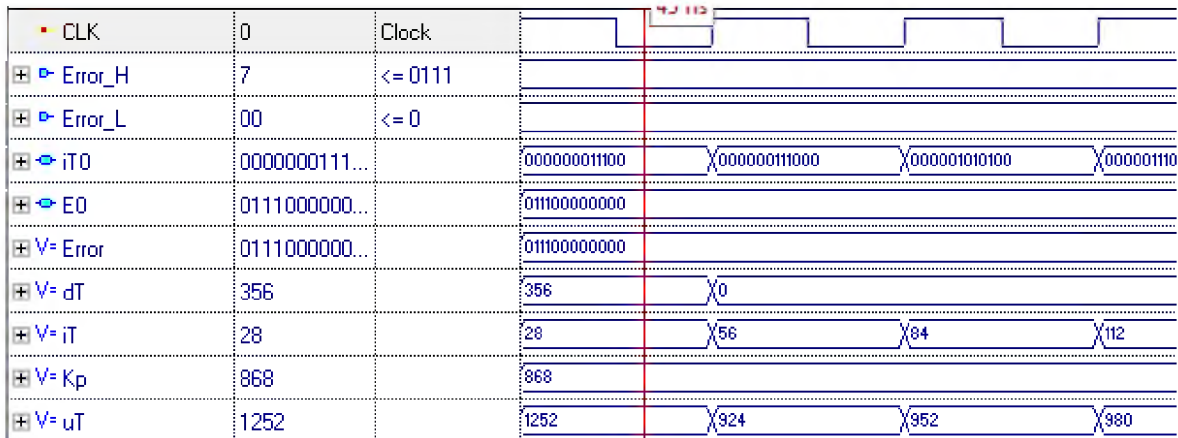


Figura 5.4 Simulación del módulo de control

La señal de Error\_H y Error\_L contienen el valor del error en su parte entera y en su parte fraccionaria, respectivamente. Como ya se mencionó en el capítulo 4, la salida del controlador (uT) está formado por tres componentes: La componente derivativa (dT), la componente integral (iT) y la componente proporcional (Kp). En este caso en particular la componente Kp siempre tendrá el mismo valor debido a que en todos los periodos de tiempo el error permanecerá constante (entregando el valor de 868 para el caso del error en 7.0). La componente derivativa sólo tendrá una aportación a uT ya que en el primer ciclo de reloj el error predeterminado es 0, pero al realizar la actualización a 7.0 se obtiene una diferencia lo que origina un cambio en la componente dT. Para el segundo ciclo de reloj el error vuelve a tener el valor de 7 y por lo tanto no se obtiene un cambio entre el error anterior y el error actual, dejando a la componente dT sin aportación para los siguientes ciclos de reloj. Finalmente la componente iT aumenta conforme el error permanezca. Se

puede apreciar en la figura 5.5 que en cada ciclo de reloj la componente  $iT$  aumenta (58, 64 y 112). Hasta llegar a un nivel máximo permitido por el CDA.

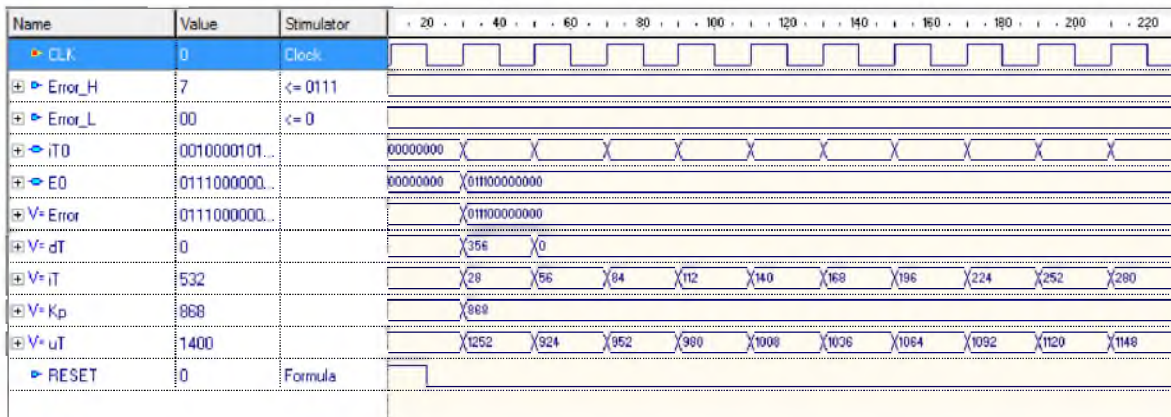


Figura 5.5 Incremento de la componente  $iT$ .

La salida del controlador ( $uT$ ) está conectada al módulo SPI el cual entrega un valor al CDA de 12 Bits. En este caso se consideró el cero de referencia como 1000 0000 0000, es decir la mitad del vector de 12 bits. Así el controlador tiene la posibilidad de entregar valores negativos a la planta. La siguiente figura muestra la ideología que se toma para esta adaptación.

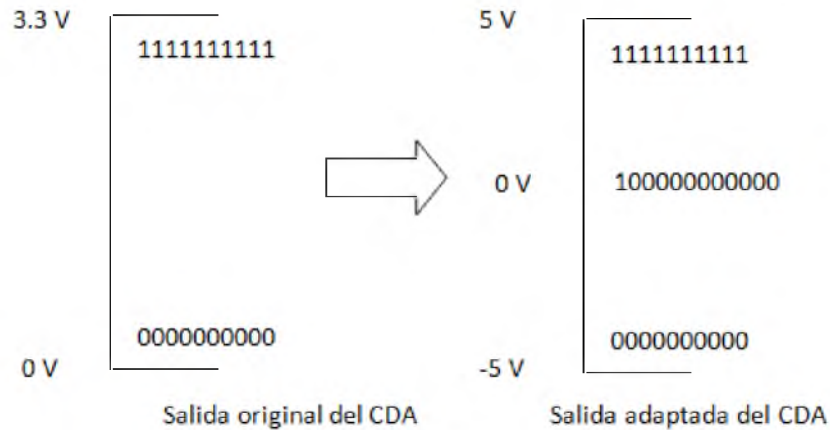


Figura 5.6 Adaptación de los bits del CDA

Mediante esta adaptación se pueden manejar 0000000000 como valor máximo para números negativos y 1111111111 como el valor máximo para números positivos. Es

importante mencionar que esta adaptación es gracias a un circuito externo a la Spartan 3E. El circuito está basado en amplificadores operacionales (Figura 5.7)

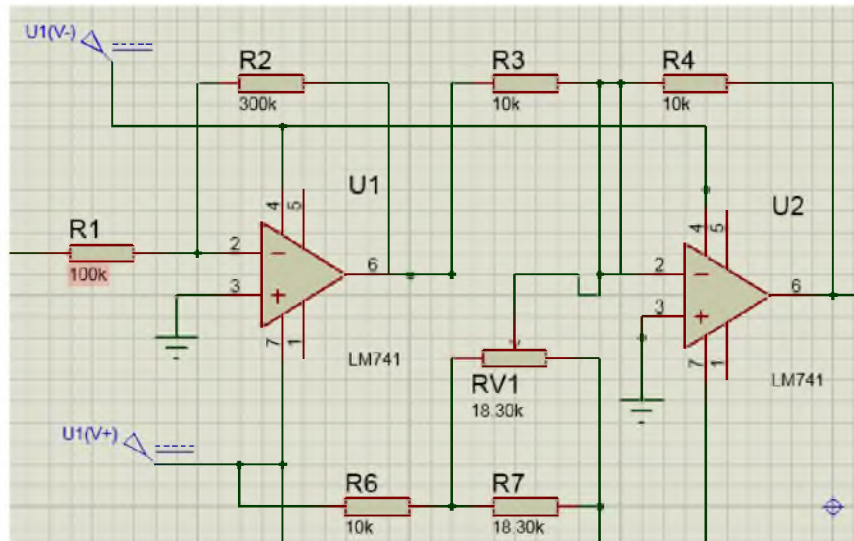


Figura 5.7 Circuito externo para la adaptación del CDA.

Este circuito no fue implementado físicamente debido a que no se realizará ninguna implementación física real.

# Capítulo 6: Conclusiones y trabajos a futuro

Este trabajo permitió codificar un sistema de control PID mediante descripción de hardware utilizando VHDL. El resultado obtenido fue una entidad destinada a sistemas de control que operan con voltaje de 0 a 10V en la entrada y -5V a 5V a la salida (Figura 6.1).

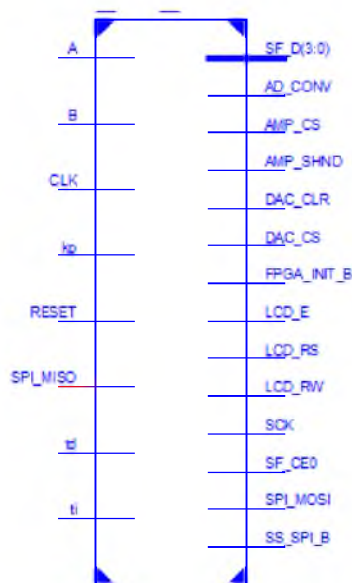


Figura 6.1 Descripción de hardware para un sistema de control

La descripción de hardware para este sistema cuenta con 8 entradas (2 entradas del encoder, 1 entrada del reloj de sincronización, 3 entradas para los parámetros del controlador, 1 entrada para los datos del CAD y 1 señal de RESET) y 14 salidas (4 salidas para controlar el LCD y 10 salidas para el módulo de SPI). El diseño fue ideado para poder ser extendido a sistemas multivariable, esto quiere decir, que se tendrá la posibilidad de agrupar varios controladores dentro de un mismo FPGA, cada uno trabajando con sus parámetros independientes a los demás.

## Referencias

- [1] Norman N. Sistemas de control para ingeniería. Tercera edición. 2010.
- [2] Katsuhiko O. Ingeniería de control moderna. Tercera edición. 1998. Pag. 60.
- [3] Kuo B. Sistemas de control automático. Séptima edición. 1996.
- [4] Maxinez D. VHDL el arte de programar sistemas digitales. Primera reimpresión. 2003.
- [5] Pardo F. VHDL lenguaje para síntesis y modelado de circuitos. Primera reimpresión.
- [6] Stephen B. Fundamentos de lógica con diseño VHDL. Segunda edición. 2006.
- [7] Serafin A. Diseño de sistemas digitales con VHDL. Primera edición. 2006.
- [8] J.M Angulo. Electrónica digital moderna. Tercera edición. 1998. Pag. 60.
- [9] Thomas L. Floyd. Fundamentos de sistemas digitales. Novena edición. 2006.
- [10] 1996. Xilinx. Spartan-3E Starter Kit Board User Guide. 2006.
- [11] Ken C. Picoblaze ADC & DAC for Spartan 3E
- [12] Ken C. Picoblaze LCD for Spartan
- [13] Ken C. Picoblaze Rotary encoder for Spartan 3E
- [14] 3E Katsuhiko O. Sistema de control en tiempo discreto. Segunda edición.
- [15] Eduardo G. Compilación C CCs y simulación PROTEUS para microcontroladores Plc. Primera edición. 2008
- [16] Antonio C. Instrumentación industrial. 1998
- [17] Robert F. Amplificadores operacionales y circuitos integrados lineales. 1993

## **ANEXOS**