

# VHDL: Tendencia Tecnológica para Controladores Digitales

Ismael Millán Páez, Oscar Montiel, Roberto Sepúlveda

## *VHDL: technology trend for digital controllers*

Recibido: abril 05, 2007

Aceptado: junio 20, 2007

Palabras clave: VHDL, control automático

### **Abstract:**

*This article presents an overview of the applications can be developed using VHDL, highlighting all the features that can contribute to the development of new technological developments. Automatic control is an area of science, where VHDL has great potential for application, so in this article will give an overview of the usefulness of this language and Control.*

**Keywords:** VHDL, automatic control



LECTRONIC Design Automation (EDA), es el nombre que se le da a todas las herramientas de hardware y software en el diseño de sistemas electrónicos. Porque no sólo el software es importante, también lo es el hecho de que las computadoras cada día son más veloces y de mayor capacidad de procesamiento, lo cual influye en el proceso de diseño de circuitos electrónicos [1].

El diseño de hardware tiene un problema fundamental, que no existe en desarrollo de software. Este problema es el alto costo del ciclo diseño – desarrollo del prototipo – pruebas – reinicio del ciclo, ya que el costo del prototipo generalmente suele ser bastante elevado. Se impone la necesidad de reducir este ciclo de diseño para no incluir la fase de desarrollo del prototipo más que al final del proceso, evitando la repetición de varios prototipos que es lo que encarece el ciclo. Para ello se introduce la fase de simulación y verificación de circuitos utilizando herramientas EDA, de tal forma que no sea necesario implementar físicamente un

prototipo para comprobar el funcionamiento del circuito. En el ciclo de diseño de circuitos, las herramientas EDA están presentes en todas las fases. Primero en la fase de generación del sistema que puede representarse mediante un diagrama esquemático, a bloques o de flujo.

Se encuentran también en la fase de simulación y comprobación de circuitos, donde diferentes herramientas permiten verificar el funcionamiento del sistema. Estas simulaciones pueden ser de eventos, funcionales, digitales o eléctricas, de acuerdo al nivel de simulación requerido. Después están las herramientas EDA utilizadas en la síntesis y programación de circuitos digitales en dispositivos lógicos programables. Existen, además, las herramientas EDA orientadas a la fabricación de circuitos. En el caso del diseño de hardware estas herramientas sirven para la realización de PCBs (Printed Circuit Boards o placas de circuito impreso), o para desarrollar circuitos integrados de aplicación específica conocidos como ASICs (Application Specific Integrated Circuits). Este ciclo de diseño de hardware se muestra en la Figura 1 [1].

Los lenguajes de descripción de circuitos son aquellos mediante los cuales es posible describir el funcionamiento y estructura de un circuito eléctrico o digital. La descripción puede ser mediante bloques donde se muestra la arquitectura del diseño, o de comportamiento, es decir, se describe el funcionamiento del circuito en vez de especificar los elementos de los que está compuesto.

Hoy en día las aplicaciones electrónicas requieren de lenguajes de programación que sean sencillos y eficientes para la solución de problemas que en la actualidad están impactando al mundo electrónico. Una de las áreas en las cuales hay mucho desarrollo es Control, en la cual la utilización de este lenguaje resulta muy conveniente a la hora de tener requerimientos de programación fácil y velocidad en tiempo de ejecución de la aplicación ya programada. Son objetivos de este artículo mostrar aplicaciones de control que se pueden realizar mediante VHDL, así como motivar el uso de este lenguaje como una opción interesante para el desarrollo tecnológico.

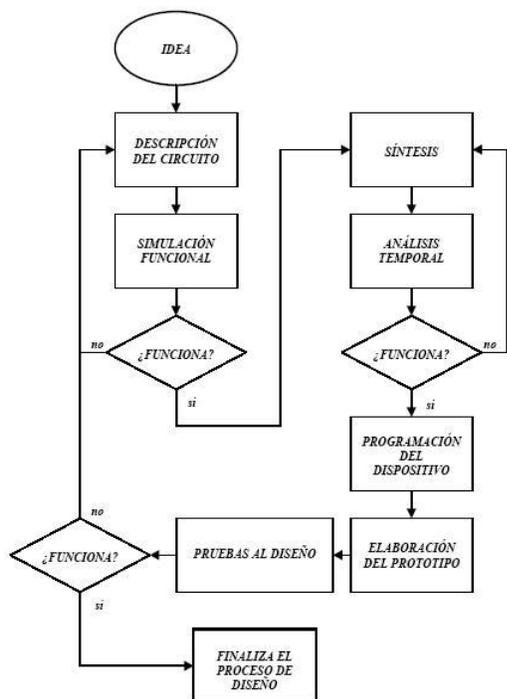


Figura 1. Diagrama de flujo para el desarrollo de sistemas electrónicos.

El lenguaje VHDL está creado específicamente para el diseño de hardware, es decir, podremos implementar con él multitud de circuitos lógicos, tanto combinacionales como secuenciales.

La actividad que se ha generado entorno a VHDL es muy intensa. En muchos países, entre ellos en España, se han creado grupos de trabajo en torno a VHDL. Se realizan reuniones periódicas con presentación de trabajos tanto en Estados Unidos (VIUF, VHDL International User's Forum) como en Europa (VHDL Forum for CAD in Europe), así como el congreso EuroVHDL que se celebra una vez al año. Todas las empresas que se dedican a la microelectrónica han ido paulatinamente adaptándose a VHDL. En Japón tiene una creciente aceptación, a pesar de que tienen su propio lenguaje estándar UDL/I. Siemens ha creado un 'VHDL

Center'. En resumen, el lenguaje está teniendo un impacto muy importante.

## QUÉ ES EL VHDL

VHDL es el acrónimo que representa la combinación de VHSIC y HDL, donde VHSIC es el acrónimo de Very High Speed Integrated Circuit y HDL es a su vez el acrónimo de Hardware Description Language [3]. Los estudios para la creación del lenguaje VHDL (VHSIC HDL) comenzaron en el año 1981, bajo la cobertura de un programa para el desarrollo de Circuitos Integrados de Muy Alta Velocidad (VHSIC), del Departamento de Defensa de los Estados Unidos. En 1983 las compañías Intermetrics, IBM y Texas Instruments obtuvieron la concesión de un proyecto para la realización del lenguaje y de un conjunto de herramientas auxiliares para su aplicación. Finalmente, en el año 1987, el lenguaje VHDL se convierte en la norma IEEE-1076 –como todas las normas IEEE, se somete a revisión periódica, por lo que en 1993 sufrió algunas leves modificaciones.

VHDL se ha diseñado para soportar diferentes metodologías de diseño ('top-down' frente a diseño basado en módulos) así como diferentes tecnologías de diseño (circuitos comerciales, microprocesadores, PLDs, FPGAs, ASICs, etc) con distinta funcionalidad (circuitos combinacionales, síncronos y asíncronos). De este modo, el lenguaje puede usarse por organizaciones con puntos de vista diferentes y satisface las necesidades de diseño distintas que puede tener el diseñador, el centro de diseño de ASICs, la fundición o el vendedor de herramientas CAD. VHDL es un lenguaje usado por ingenieros definido por el IEEE (Institute of Electrical and Electronics Engineers) (ANSI/IEEE 1076-1993) que se usa para diseñar circuitos digitales. Existen métodos para diseñar circuitos son la captura de esquemas (con herramientas CAD) y los diagramas de bloques, pero éstos no son prácticos en diseños complejos. Otros lenguajes para el mismo propósito son Verilog y ABEL [2].

## Cómo se puede describir un circuito en VHDL

La forma de diseñar circuitos en VHDL se divide en tres categorías de acuerdo a su complejidad: funcional, flujo de datos y estructural [1]. Estos tres estilos de diseño se detallan a continuación.

### Funcional

El diseño es un poco más complicado ya que requiere de varias decisiones antes de definir los datos de salida correctos. Por lo que se requiere de una descripción algorítmica del funcionamiento del circuito para facilitar el diseño del sistema. En VHDL esto se obtiene expresando el funcionamiento del diseño mediante una estructura PROCESS la cual se compone de instrucciones secuenciales.

### Flujo de datos

En este estilo el diseño del circuito no es complicado por lo que basta con describir como fluyen los datos a través de la entidad, de las entradas hacia las salidas. La operación del sistema está definida en términos de un conjunto de transformaciones de datos expresadas como instrucciones concurrentes.

### Estructural

Una descripción estructural se utiliza en circuitos de más de una función, hablando en términos de hardware, para realizar la finalidad del sistema. Para ello segmentamos el sistema en subcircuitos o componentes para facilitar el diseño. Cada componente es caracterizado en particular ya sea utilizando una descripción de flujo de datos o de comportamiento. Y a la entidad donde se describen las interconexiones de estos componentes recibe el nombre de descripción estructural.

## Características del lenguaje

El lenguaje VHDL fue creado con el propósito de especificar y documentar circuitos y sistemas digitales utilizando un lenguaje formal. En la práctica se ha convertido, en un gran número de entornos de CAD, en el HDL de referencia para realizar modelos sintetizables automáticamente. Las principales características del lenguaje VHDL se explican en los siguientes puntos:

### Descripción textual normalizada

El lenguaje VHDL es un lenguaje de descripción que especifica los circuitos electrónicos en un formato adecuado para ser interpretado tanto por máquinas como por personas.

### Amplio rango de capacidad descriptiva

El lenguaje VHDL posibilita la descripción del hardware con distintos niveles de abstracción, pudiendo adaptarse a distintos propósitos y utilizarse en las sucesivas fases que se dan en el desarrollo de los diseños.

### Otras ventajas

Además de las ventajas ya reseñadas también es destacable la capacidad del lenguaje para el manejo de proyectos de grandes dimensiones, las garantías que comporta su uso cuando, durante el ciclo de mantenimiento del proyecto, hay que sustituir componentes o realizar modificaciones en los circuitos, y el hecho de que, para muchas organizaciones contratantes, sea parte indispensable de la documentación de los sistemas.

## Componentes principales de un programa en VHDL

En un diseño en VHDL tenemos dos partes principales: la entidad mostrada en la Figura 2; es como una caja negra en la que se definen entradas y salidas pero no se tiene acceso al interior, y es lo que usa cuando se reutiliza un diseño dentro de otro; la arquitectura, que es donde se describe el diseño de la forma que se ha visto antes. Otros elementos del lenguaje son las librerías, paquetes, funciones, etc. La forma de definir una entidad se muestra a continuación [3]:

```
entity suma is
port (A,B : in bit;
SUM, CARRY : out bit);
end suma;
```

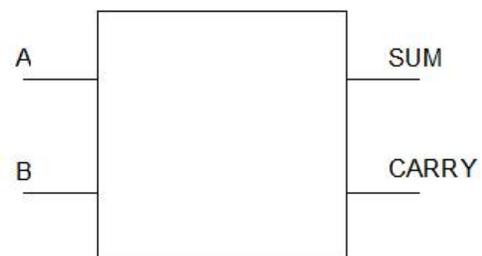


Figura 2. Representación en bloque de la entidad en VHDL.

Enseguida se muestra un programa simple escrito mediante VHDL para ejemplificar el uso de la entidad y la arquitectura.

```

library ieee;
use ieee.std_logic_1164.all;

entity pwm is
    port ( clock      : in  std_logic;
          reset      : in  std_logic;
          state      : in  std_logic;
          pwm_signal  : out std_logic
        );
end pwm;

architecture RTL of pwm is

begin

    gen : process

    begin

        -- Sincronizaci\on del flanco de
        -- subida del reloj (comentario)

        wait until rising_edge( clock );

        -- Reset s\incrono
        --
        if reset = '1' then

            pwm_signal <= '0';

        else

            if state = '1' then

                pwm_signal <= '1';

            else

                pwm_signal <= '0';

            end if;

        end if;

    end if;

    end process gen;

end RTL;

```

El programa anterior realiza la tarea de un PWM, donde se encuentra definida la entidad y la arquitectura de dicho diseño. En la entidad podemos ver que están declaradas las entradas y salidas de nuestro PWM:

**Clock:** que es el reloj del sistema. **Reset:** Si el Reset se encuentra en 1, la salida del PWM se vuelve 0, si el reset esta en nivel bajo el programa se ejecuta.

**State:** Determina si a la salida del PWM se le mandará un 0 ó 1. **Pwm\_signal:** La salida del sistema.

En la arquitectura del diseño podemos ver el funcionamiento del programa: se declara el proceso gen, el programa espera hasta que haya un flanco de subida en el reloj, revisa si reset está en 1 manda a la salida del PWM un 0 y termina el proceso, se vuelve a iniciar el programa esperando de nuevo otro flanco de subida.

Si el reset fue 0 entonces se pregunta si la entrada state es 1, si es así manda un 1 a la salida del pwm, si state fue 0 a la salida del PWM le manda un 0, termina el proceso y el programa vuelve a iniciarse.

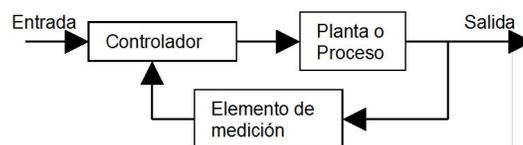
## CONCEPTOS SOBRE CONTROL AUTOMÁTICO

El control automático ha jugado un papel vital en el avance de la ingeniería y de la ciencia. Además de su extrema importancia en vehículos espaciales, en la guía de proyectiles y sistemas de pilotaje de aviones, etc., el control automático se ha convertido en parte importante e integral de los procesos de manufactura e industriales modernos. Por ejemplo, el control automático resulta esencial en operaciones industriales como el control de presión, temperatura, humedad, viscosidad y flujo en las industrias de procesos; maquinado, manejo y armado de piezas mecánicas en las industrias de fabricación entre muchos otros [4].

Dentro de los sistemas de control destacan dos ramas los sistemas de lazo cerrado y de lazo abierto, los cuales se explicarán a continuación para entender un poco más acerca de lo que se puede desarrollar en VHDL.

### Sistemas de control de lazo cerrado

En la Figura 3 se muestra un sistema de control de lazo cerrado donde la señal de salida tiene efecto directo sobre la acción de control.



*Figura 3. Sistema de control de lazo cerrado.*

Un sistema de lazo cerrado es un sistema retroalimentado en donde la señal de error actuante, es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la señal de salida o una función de la señal de salida y sus derivadas), entra al detector o control para reducir el error y llevar la salida del sistema al valor deseado.

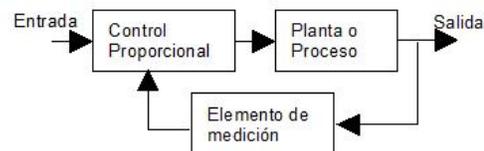


Figura 5. Sistema de un control proporcional de lazo cerrado.

## Sistemas de control de lazo abierto

Los sistemas de control de lazo abierto son sistemas de control en los que la salida no tiene efecto sobre la acción de control. En un sistema de lazo abierto la salida no es realimentada para comparación con la entrada. Un esquema de este tipo de control se ilustra en la Figura 4.

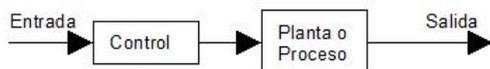


Figura 4. Sistema de control de lazo abierto.

## Controladores

Sabiendo lo que es el control automático podemos describir algunos tipos de control que se pueden desarrollar mediante VHDL para ello se explicará el funcionamiento de cada uno primero y después se relacionarán con el lenguaje de programación.

### Acción de control proporcional

La parte proporcional consiste en el producto entre la señal de error y la constante proporcional. Esta componente toma un papel importante cuando la señal de error es grande, pero su acción se ve mermada con la disminución de dicha señal. Este efecto tiene como consecuencia la aparición de un error permanente, que hace que la parte proporcional nunca llegue a solucionar por completo el error del sistema. Para un control de acción proporcional, la relación entre la salida del controlador  $m(t)$  y la señal de error actuante  $e(t)$  es :

$$m(t) = K_p e(t) \quad (1)$$

Un esquema de un control proporcional se muestra en la Figura 5.

### Acción de control integral

El modo de control Integral tiene como propósito disminuir y eliminar el error en estado estacionario, provocado por el modo proporcional. El error es integrado, lo cual tiene la función de promediarlo o sumarlo por un periodo de tiempo determinado; luego es multiplicado por una constante I. I representa la constante de integración. Posteriormente, la respuesta integral es adicionada al modo Proporcional para formar el control P + I que se muestra en la Figura 6; con el propósito de obtener una respuesta estable del sistema sin error estacionario.

$$m(t) = K_p e(t) + \frac{K_i}{T_i} \int_0^t e(t), \quad (2)$$

donde:

$K_p$ = constante proporcional

$K_i$ = constante integral

$T_i$ = tiempo integral

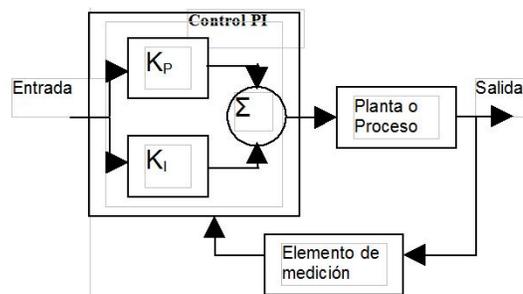


Figura 6. Sistema de un control proporcional-integral de lazo cerrado.

### Acción de control derivativa

El propósito de la acción derivativa es mejorar la estabilidad de lazo cerrado. El mecanismo de inestabilidad puede ser

descrito intuitivamente como sigue: debido a la dinámica del proceso, pasa algún tiempo antes de que la variable de control se note en la salida del proceso. De esta manera, el sistema de control tarda en corregir el error. La acción de un controlador con acción proporcional y derivativa puede ser interpretada como si el control proporcional fuese hecho para predecir la salida del proceso.

#### Acción de control PID

Con estos 3 controles podemos formar un Control PID, en la Figura 7 se muestra un esquema de dicho control, el cual es muy utilizado en la industria por las prestaciones que este tiene, así como la composición del controlador PID la podemos observar en la Figura 8.

$$m(t) = K_p e(t) + K_d T_e \frac{de(t)}{dt} + \frac{K_i}{T_i} \int_0^t e(t), \quad (3)$$

donde:  $K_p, K_i, K_d =$  constantes proporcional, integral, derivativa

$T_i =$  tiempo integral.

$T_d =$  tiempo derivativo.

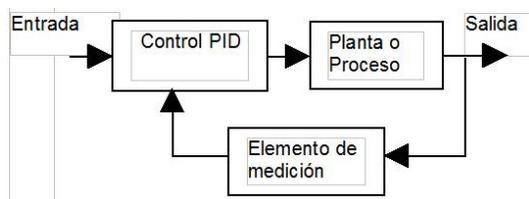


Figura 7. Sistema de un control proporcional-integral-derivativo de lazo cerrado.

Los controladores PID son suficientes para resolver el problema de control de muchas aplicaciones en la industria, particularmente cuando la dinámica del proceso lo permite (en general procesos que pueden ser descritos por dinámicas de primer y segundo orden), y los requerimientos de desempeño son modestos (generalmente limitados a especificaciones del comportamiento del error en estado estacionario y una rápida respuesta a cambios en la señal de referencia).

Los fabricantes proporcionan los controladores PID de variadas formas. Existen sistemas del tipo “stand alone” con capacidad para controlar uno o varios lazos de control. Estos dispositivos son fabricados en el orden de cientos de miles al año. El controlador PID es también un ingrediente

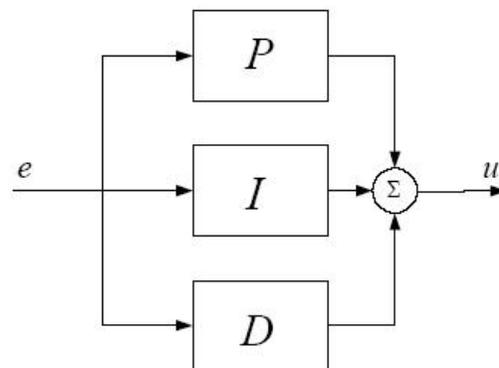


Figura 8. Componentes internos de un control PID.

importante en los sistemas de control distribuido, ya que proporciona regulación a nivel local de manera eficaz. Por otro lado, pueden también venir empotrados, como parte del equipamiento, en sistemas de control de propósito especial, formando así parte integrante de la aplicación.

Su uso extensivo en la industria es tal que el 95% de los lazos de control que existen en las aplicaciones industriales son del tipo PID, de los cuales la mayoría son controladores PI, lo que muestra la preferencia del usuario en el uso de leyes de control muy simples. En general, el usuario no explota todas las características de estos controladores, quizás por falta de una mejor comprensión desde el punto de vista de la teoría de control.

Los controladores PID han sobrevivido a muchos cambios en la tecnología a lo largo de su historia. Desde los antiguos reguladores de Watt, de la época de la revolución industrial, pasando por los controladores neumáticos, los controladores analógicos eléctricos y electrónicos (primero implementados con válvulas y luego con circuitos integrados) hasta los modernos controladores basados en microprocesadores, que proporcionan una mayor flexibilidad debido a su programabilidad. El microprocesador ha tenido una influencia dramática sobre el desarrollo del controlador PID; ha permitido brindar nuevas oportunidades para implementar funciones adicionales como el ajuste automático de parámetros y los cambios de modos de control.

El desarrollo de los sistemas de control PID está también influenciado por el desarrollo en el campo de la comunicación de datos de campo, lo que ha permitido su inserción como módulos importantes en los esquemas de control distribuido. En este sentido, la capacidad de comunicación de

estos dispositivos con otros dispositivos de campo como PLCs y otros sistemas de control de niveles superiores, como es el caso de FPGA en donde se utiliza la programación en VHDL, es una función necesaria en los modernos controladores PID.

Si bien a nivel industrial existen grupos de ingenieros de procesos e instrumentación que están familiarizados con los controladores PID, en el sentido de que llevan una práctica continua de instalación, puesta en marcha y operación de sistemas de control con lazos PID, también es cierto que existe mucho desconocimiento acerca de los detalles involucrados en la construcción de los algoritmos. Por esto la continua búsqueda de lenguajes de programación en los cuales se pueda desarrollar este tipo de aplicaciones es esencial en los avances tecnológicos de esta y cualquier área. Debido a esta constante necesidad de tener controles más eficientes es inevitable el desarrollo de aplicaciones de este tipo en VHDL, ya que es una herramienta de programación muy versátil y fácil de implementar, así como su rápido procesamiento en dispositivos como FPGA's la hacen una excelente opción para la innovación de nuevos controladores.

## APLICACIONES REALIZADAS MEDIANTE VHDL

VHDL se ha aplicado principalmente en el desarrollo de aplicaciones en FPGA, ya que estos dispositivos proporcionan importantes características que los hace una alternativa interesante para realizar nuevas tecnologías o mejorar las ya existentes. Entre las características más importantes de los FPGA se encuentra: la rapidez con la que trabaja la ejecución de los procesos, el costo, la versatilidad de aplicaciones en las cuales se pueden utilizar, su funcionamiento no tan complicado, la integración de circuitos que se pueden hacer dentro de él y la reconfiguración de los programas diseñados en VHDL cada vez que el diseño lo requiera para su funcionamiento óptimo. En VHDL pueden realizarse distintas aplicaciones entre las cuales están:

- Sistemas de control en tiempo real.
- Procesamiento de imágenes.
- Sistemas sensoriales.
- Modulación digital.
- Demodulación digital.

- Filtrado mediante correlaciones.
- Implementación de dispositivos electrónicos.
- En aplicaciones comerciales como:
  - Detección de objetos en vías de ferrocarril.
  - Procesamiento simultaneo de señales sensoriales.
  - Sistemas de posicionamiento en robots móviles.
  - Diseño de videojuegos en 2D.

Un mundo de aplicaciones interesantes se pueden realizar mediante este lenguaje y un área del conocimiento científico que tiene numerosos avances y desarrollo es “Control Automático”, para ello daremos una breve explicación acerca de esta área que es predominante en la tecnología actual.

## Controlador PID con VHDL

Los controladores PID será muy difícil que desaparezcan, porque hoy en día se siguen utilizando en la mayoría de las industrias y la tecnología actual todavía emplea este tipo de controladores, por ello a pesar de que ya se han realizado en muchos lenguajes de programación, sería una opción interesante realizarlo en VHDL, por todas las prestaciones que este lenguaje tiene y también un motivo muy importante sería la implementación en FPGA's que lo hace aún más atractivo. Un desarrollo de este tipo utiliza una estructura parecida a la siguiente:

Primero, hay que determinar el problema a resolver o sistema a controlar, es decir, por ejemplo pudiera ser un motor de CD, al cual se le quiera controlar la velocidad mediante un controlador PID diseñado en VHDL.

Ya teniendo el problema a resolver debemos tomar en cuenta las características del motor, puesto que es muy importante saber las revoluciones por minuto y el funcionamiento del encoder de dicho motor, el cual nos servirá para contar los pulsos y así mediante programación determinar la velocidad real a la que el motor está girando.

Hay que desarrollar un programa en VHDL que consista en los 3 controles (proporcional, integral y derivativo), así como el PWM (modulación por ancho de pulsos), el cual sirve para ampliar o disminuir la frecuencia con la que el motor estará trabajando y así incrementar o decrementar su velocidad.

Un experimento interesante sería utilizar los bloques que se pueden realizar en VHDL y encapsular los controles por separado y así poder usar ya sea un proporcional, un proporcional-integral, un proporcional-derivativo o un PID para observar las distintas reacciones que puede ocasionar la combinación de dichos controladores, este es un ejemplo de muchos que se pueden hacer con este lenguaje.

VHDL cuenta con una gama amplia de herramientas que pueden ayudar a mejorar el funcionamiento de programas o sistemas que con otros lenguajes no se les puede obtener su mayor rendimiento. Por lo tanto es una alternativa vanguardista que contribuye muy sustancialmente en el avance tecnológico de hoy en día, así como los futuros trabajos que se puedan realizar a través de esta herramienta. VHDL siendo relativamente joven tiene un futuro prometedor y un gran potencial en el desarrollo de hardware simulado.

## CONCLUSIONES

El uso de VHDL tiene la siguientes ventajas:

- Disponibilidad pública.
- Independencia de dispositivos y fabricantes.
- Reutilización.
- Diseño jerárquico

VHDL implementado en FPGA's es la excelente combinación de programación de hardware mediante software y velocidad de procesamiento. En la actualidad, VHDL constituye el lenguaje estándar de referencia a nivel internacional. Impulsado originalmente por el DoD de los EEUU, cualquier programa lanzado por alguna de sus agencias oficiales obliga al uso de VHDL en el modelado de los sistemas y en la documentación del proceso de diseño. Este hecho ha motivado que la gran mayoría de empresas y universidades norteamericanas hayan adoptado el lenguaje. Control automático aplicado en VHDL es una opción importante para el desarrollo de controladores más eficientes con mayor velocidad de respuesta y facilidad de diseño. Los controladores PID son muy utilizados actualmente, por ello el desarrollo de estos en VHDL los hacen más atractivos por todas las ventajas que este lenguaje ofrece.

## Bibliografía

- [1] Apuntes “Síntesis y descripción de circuitos digitales utilizando VHDL”, Francisco Javier Torres Valle, Universidad Autónoma de Guadalajara, 2002. Disponible en: [http://www.uag.mx/214/I\\_LENGUAJES\\_DE\\_DESCRIPCION\\_DE\\_HARDWARE.pdf](http://www.uag.mx/214/I_LENGUAJES_DE_DESCRIPCION_DE_HARDWARE.pdf)
- [2] Apuntes de VHDL, 2007 <http://es.wikipedia.org/wiki/VHDL>
- [3] Apuntes “Introducción al diseño con VHDL” Ricardo José Colom Palero, Departamento de ingeniería electrónica, Universidad Politécnica de Valencia, 2001.
- [4] Katsuhiko Ogata, Ingeniería de control moderna, Editorial Prentice Hall, séptima edición 1980.
- [5] Meter J. Ashenden, The VHDL cookbook, Dept. Computer Science University of Adelaide South Australia, 1990
- [6] IEEE, “IEEE Standard VHDL language reference manual”, 1994.
- [7] Apuntes “VHDL Lenguaje para descripción y modelado de circuitos”, Fernando Pardo Carpio, Ingeniería informática, Universidad de Valencia, 1997.

### Acerca del autor o autores

Oscar Montiel y Roberto Sepúlveda son Investigadores del Instituto Politécnico Nacional Centro de Investigación y Desarrollo de Tecnología Digital, Ave. Del Parque No. 1310, Mesa de Otay, Tijuana B.C 22510 TEL: +(664)6231344. Ismael Millán Páez es Alumno de la Maestría en Sistemas Digitales del CITEDI. Correos-e: [millan@citedi.mx](mailto:millan@citedi.mx), [o.montiel@ieee.org](mailto:o.montiel@ieee.org), [r.sepulveda@ieee.org](mailto:r.sepulveda@ieee.org)