

UNIDAD PWM EN LÓGICA PROGRAMABLE Y SU APLICACIÓN EN LA MODULACIÓN DE LA INTENSIDAD DE UN PANEL DE LEDS

Juan Carlos Herrera Lozada, Agustín Cruz Contreras, Israel Rivera Zárate
Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC – IPN)
Av. Té 950, UPIICSA - IPN, Edificio de Graduados 2° piso.
Col. Granjas México, Delegación Iztacalco, México, D. F. C. P. 08400.
Tel (01 55) 5624.2000 Exts. 70263, 70275. Fax: (01 55) 5624.2000 Ext. 70265.
E-mails:
jcrs@ipn.mx, acruz@ipn.mx, irivera@ipn.mx

RESUMEN: El presente trabajo muestra el diseño de una unidad PWM para controlar la intensidad de un panel de leds. Se analiza la conversión de una señal digital a su contraparte analógica a través de un dispositivo de lógica programable.

1. INTRODUCCIÓN

En la actualidad, los convertidores de señal digital a analógica colocados a la salida de un sistema de procesamiento digital se diseñan de forma embebida (interna). Existen diferentes alternativas hardware para realizar la conversión; sin embargo, la tendencia es considerar diseños de bajo coste sin sacrificar la versatilidad del módulo convertidor. Una unidad PWM (Modulación por Ancho de Pulso) es básicamente un DAC (Digital-Analog Converter) fácil de implementar y que se puede utilizar en sistemas que no requieran una gran velocidad de muestreo: control de motores, variación de voltaje en diversas aplicaciones o en algunos casos, manejo de señales acústicas de baja calidad.

Es importante reconocer las diferentes alternativas en la implementación; por lo general, se resuelve utilizando algún microcontrolador o alguno otro driver monolítico con funcionamiento específico. En este trabajo se consideran dispositivos de lógica programable, siendo alternativas de costo reducido (como en el caso de un GAL), con magnífico desempeño y velocidad, reconfigurables y adaptables a aplicaciones simples o complejas (utilizando FPGAs o CPLDs).

2. TEORÍA PWM

La modulación por ancho de pulsos consiste en una técnica que compara una señal de referencia con una señal moduladora de mayor frecuencia, con la intención de que la intersección entre

ambas señales genere los instantes de conmutación de la salida PWM. Una unidad de estas características asigna cierta duración de tiempo en alto o en bajo a un dato digital de n bits que se considera salida de la etapa procesadora. Lo anterior se logra conectando un contador y un circuito comparador. El comparador determinará si el dato aplicado a la entrada de la unidad es menor al valor binario del contador que cambia constantemente. El tiempo de la señal en alto ("1" lógico) depende de la cantidad de pulsos de reloj que se apliquen hasta que el contador presente un dato binario mayor o igual al de la entrada.

A la salida de la unidad PWM es necesario conectar un filtro RC (pasabajas) para determinar el nivel analógico propuesto por la señal de referencia. El módulo PWM básico se muestra en la **figura 1**; también es posible adicionar un registro de datos para mantener estable el dato de la señal de referencia denominada en la figura como *entrada digital*.

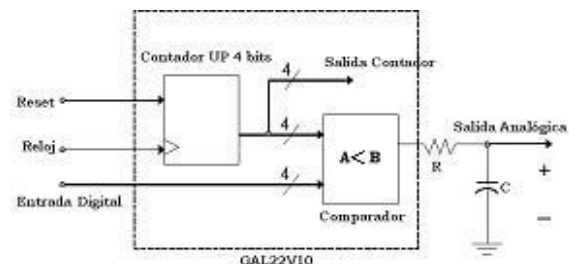


Figura 1. Módulo PWM digital.

En una señal PWM, la frecuencia es constante mientras la duración del ciclo útil varía (de 0% a 100%) de acuerdo a la amplitud de la señal original.

Las **figuras 2a** y **2b**, muestran la proporcionalidad y la duración del ciclo cuando se tienen diferentes entradas de referencia (4 y 12), considerando un

contador de 4 bits. Obsérvese que para una señal con valor 4, el ciclo de la señal PWM dura menos tiempo en alto ("1" lógico) que cuando vale 12. El objetivo de la conversión es que el capacitor del filtro RC entregue un voltaje proporcional a la duración de la parte alta del ciclo de la salida PWM.

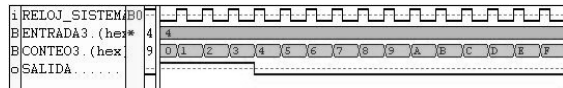


Figura 2a. Salida PWM para una referencia de valor 4.

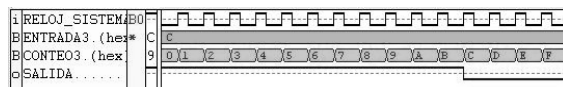


Figura 2b. Salida PWM para una referencia de valor 12.

3. SISTEMA DISEÑADO

La aplicación incluye un controlador para señalización, independiente de la unidad PWM, con un oscilador modelado con inversores dentro del mismo dispositivo GAL. La unidad PWM se implementó en otro GAL y posteriormente se realizaron descripciones compatibles en VHDL y ABEL para la programación de FPGAs.

3.1 Oscilador del Controlador de Señalización

El multivibrador inestable (sin histéresis) para el oscilador, se diseñó a través de compuertas inversoras como se aprecia en la figura 3, cumpliendo la siguiente ecuación:

$$f = \frac{1}{2R_1C \left[\frac{0.405 R_2}{R_1 + R_2} \right] + 2R_1C(0.693)} \quad (\text{eq. 1})$$

Se advierte que con $R_1 = R_2$, $f = \frac{0.559}{RC}$. (eq. 2)

Los valores comerciales para los componentes son: $R = 4.7 \text{ K}\Omega$ y $C = 100 \mu\text{F}$, para

$$f = \frac{0.559}{(4.7 \text{ K}\Omega)(100 \mu\text{F})} = 1.1 \text{ Hz}$$

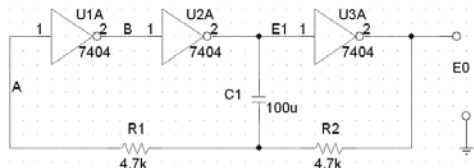


Figura 3. Circuito del oscilador RC.

3.2 Código del Controlador de Señalización

El control de señalización programado en el dispositivo GAL consta de dos partes: la

descripción del oscilador RC explicado con anterioridad y el control en sí.

Fundamentalmente se consideraron dos entradas variables (*equis* y *flecha*) y tres entradas fijas (reloj de la intermitencia; A y E1, como entradas a los inversores del oscilador RC). Las primeras se utilizan para adecuar la señalización y las otras tres forman parte de la lógica que modela el oscilador, tal y como se aprecia en el diagrama anterior.

En el código ABEL que se lista a continuación, la salida *EO* es la entrada de la señal de reloj del contador del controlador. Se modeló una solución con un contador de 60 estados para la intermitencia. Las salidas del control sólo son dos: una línea que prende los leds que forman la *equis* en el panel y otra que prende los correspondientes a la *flecha*. En el código se nombraron como *salida_equis* y *salida_flecha*. Con una lógica OR se solucionó por software la duración doble del ciclo en alto de la señal de reloj, en relación a la del ciclo en bajo.

El circuito no tiene una señal de *reset* dedicada, ya que el código se optimizó hasta lograr una lógica que reinicia el contador en el momento de que cualquiera de las dos variables de entrada es validada.

```

module led
reloj, equis, flecha pin 1, 2, 3
A, E1 pin 8, 9;
salida_equis, salida_flecha pin 20,21
istype 'com,dc';
C, EO pin 16, 17 istype 'com,dc';
B node istype 'com,dc';
cuenta5..cuenta0 pin istype 'reg,dc';
"Definición de constantes
uno = 1;
"Definición de sets
cuenta=[cuenta5..cuenta0];

"Oscilador aproximadamente a 1seg
equations
@carry 8;
cuenta.clk=reloj;
B = !A;
C = !B;
EO = !E1;

"Inicia la descripción del controlador

equations
when !(equis $ flecha) then {salida_equis =
0; salida_flecha = 0; cuenta:=0;}
else when (equis $ flecha) & cuenta==60
then {salida_equis= equis & 1;salida_flecha
= flecha & 1;cuenta:=cuenta;}
else {salida_equis=equis & (cuenta0 #
cuenta1); salida_flecha= flecha & (cuenta0
# cuenta1);cuenta:=cuenta+1;}
END

```

3.3. Diseño de la Unidad PWM

Dado que la unidad PWM se requirió para modular la intensidad del panel (aumentar la intensidad gradualmente), se colocó entre el controlador de señalización y el panel de leds. Es importante denotar que el panel requiere una etapa de potencia para sostener la corriente en todos los leds, este circuito se implementó a través de unos transistores convencionales.

Para la unidad PWM se decidió utilizar un contador ascendente de 4 bits, al igual que un comparador del mismo número de bits, tal y como se indica en la abstracción de la **figura 1**.

El periodo completo de un ciclo PWM es igual al producto del periodo del reloj de la señal de referencia (reloj del sistema) con 2^n , donde n es el número de bits del contador propuesto. Obsérvese la relación siguiente:

$$T_{PWM} = (T_{reloj})(2^n) \quad (\text{eq. 3})$$

Es posible advertir que si el contador es de 4 bits, se tendrá que $T_{PWM} = 16 T_{reloj}$, por lo que al aumentar el número de bits del contador para mejorar la resolución, el periodo PWM tenderá a hacerse más grande reduciendo drásticamente la frecuencia de salida. Lo anterior es sumamente importante cuando se desee realizar una aplicación que sí requiera sincronizar la señal analógica para controlar un sistema. El programa que modela la unidad PWM se lista a continuación. El código es fácilmente adaptable a cualquier otro HDL como Verilog o VHDL.

```
MODULE modpwm
TITLE 'PWM en GAL (DAC)'
reloj pin 1;
reset pin 2;
an3..an0 pin 3..6;
cont3..cont0 pin istype 'reg';
salida pin istype 'com';

"Definición de sets
entd = [an3..an0]; "entrada digital
cuenta = [cont3..cont0]; "contador

equations
cuenta.clk = reloj;
cuenta.ar= reset;

"comparación, contador y entrada digital
when cuenta < entd then salida = 1;
else salida=0;

"Incrementa conteo de forma ascendente
cuenta:= cuenta + 1;
END
```

En el diseño realizado fue necesario adicionar un corrimiento automático para graduar la intensidad del panel, asumiendo un contador ascendente-descendente que de manera natural obliga a que la salida PWM muestre todos los posibles valores de forma ordenada. Con el propósito anterior se utilizó una máquina de estados sencilla; el listado siguiente es sólo una parte del programa escrito, se observa el contador ascendente- descendente automático.

```
state_diagram dir

"Conteo ascendente
state 0:
if cuenta!=15 then 0 with cuenta:=cuenta+1
else 1 with cuenta:=15;

"Conteo Descendente
state 1:
if cuenta!=0 then 1 with cuenta:=cuenta-1
else 0 with cuenta:=0;
```

3.3.1 Diseño del Filtro RC Pasabajos

Como se mencionó con anterioridad, la función del filtro es convertir el dato de salida PWM a su valor analógico proporcional. Para los propósitos del diseño planteado se buscó que la intensidad de los leds del panel aumentará gradualmente con una frecuencia PWM aceptable. Considerando una frecuencia de referencia de 60 Hz, con un contador de 4 bits, y sustituyendo en *eq. 3*, se obtiene una frecuencia PWM de 3.75 Hz, equivalente a un T_{PWM} de 0.26 seg.

La relación de los valores del filtro está dada por:

$$RC = \frac{1}{2\pi F_{PWM}} \quad (\text{eq. 4}),$$

por lo que eligiendo $C = 1\mu\text{F}$, se obtiene un valor para R de aproximadamente 42 $\text{K}\Omega$ (es posible ajustar a un valor comercial más exacto). El oscilador para generar la frecuencia de trabajo para la unidad PWM se diseñó de manera similar al del controlador de señalización.

En el caso particular de esta unidad, se utilizó una frecuencia baja por lo que no fue necesario considerar aspectos de diseño más formales en el filtro. En otros casos se recomienda estimar una frecuencia de corte menor a la frecuencia PWM y no es común considerarlas iguales, tal y como se hizo en la aproximación anterior. En tal situación será necesario sustituir la frecuencia PWM por la frecuencia de corte en la *eq. 4* o tal vez, hasta

considerar un filtro de mayor orden que permita una mejor estabilidad de la señal.

4 RESULTADOS

A continuación se explican brevemente los resultados experimentales, asumiendo dos entidades independientes.

4.1 Controlador de Señalización

Las **figuras 4a** y **4b** muestran los resultados satisfactorios del diseño para el controlador. La visualización de la simulación es global y con acercamiento, respectivamente. Para referencia, los vectores de excitación están definidos en el código ABEL que modela el diseño del controlador sin la unidad PWM. Se aprecia claramente cómo cambia la salida dependiendo de la entrada que se active.

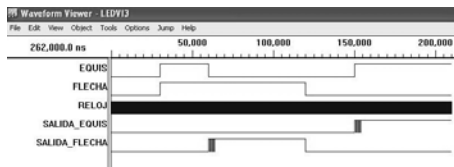


Figura 4a. Resultados de simulación para el controlador de señalización, vista global.

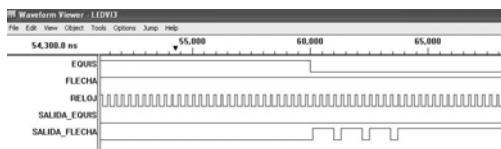


Figura 4b. Resultados de simulación para el controlador de señalización, vista con acercamiento.

El controlador de señalización se probó primeramente en una tablilla de prototipos y posteriormente conectada al panel.

El oscilador interno se acopló apropiadamente a la frecuencia del diseño, obteniendo una señal de reloj estable que puede modificarse sencillamente recalculando los valores de los elementos externos RC.

4.2 Unidad PWM

La unidad fue programada inicialmente sobre un GAL22V10, para resolver la problemática de diseño propuesta. Posteriormente se realizaron algunas pruebas programando sobre un FPGA para complementar los alcances de la técnica abordada. La **figura 5** muestra los resultados de simulación de la unidad diseñada.

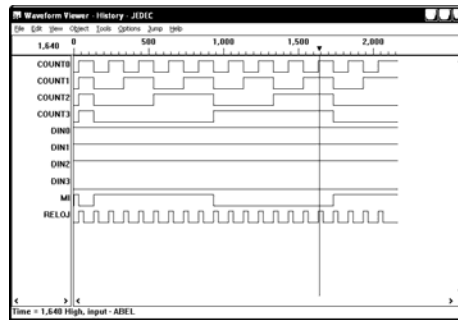


Figura 5. Resultados de simulación para la unidad PWM.

Obsérvese que las señales se exhiben de forma individual y no en conjunto, además, se trata del diseño sin el conteo ascendente – descendente automático para la graduación.

El filtro RC implementado respondió correctamente a la frecuencia de trabajo establecida convirtiendo favorablemente el dato digital a su equivalente analógico.

La **figura 6**, muestra una vista del panel de leds para las pruebas experimentales. Se aprecia el conjunto de dispositivos GAL de la solución completa. Para la configuración física de los dispositivos se utilizó un programador universal estándar y la herramienta de síntesis lógica utilizada para obtener el archivo JEDEC fue *ispEXPERT* de *Lattice Semiconductors*. Para las aproximaciones sobre FPGAs se utilizó *Foundation* de *Xilinx* y dispositivos de la familia *Spartan 2*.

Actualmente es posible apreciar los paneles de señalización en su versión terminal, en la caseta de cobro de la autopista hacia la Ciudad de Cuernavaca.

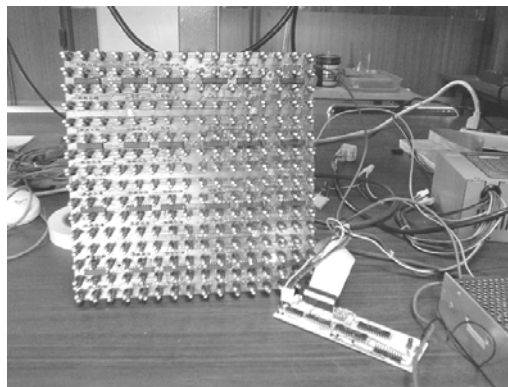


Figura 6. Panel de leds diseñado.

5. CONCLUSIONES

El controlador de señalización implementado resultó muy económico en comparación a una solución con microcontroladores. Se demostró el diseño de un oscilador RC que reduce costo y espacio al ser modelado dentro del mismo código del controlador y que resulta muy versátil en aplicaciones docentes o profesionales que requieran frecuencias de trabajo bajas.

El diseño de DACs a través de PWM es un recurso ampliamente sustentable en dispositivos de lógica programable. La lógica descrita es muy simple y se adapta sin cambios drásticos en el código a cualquier HDL con la alternativa de la arquitectura de los PLDs simples se hace extensiva para dispositivos de arquitectura avanzada como FPGAs y CPLDs.

Los trabajos planteados a futuro implican una comparación detallada con la alternativa de microcontroladores enfocada hacia el área del control. La iniciativa presentada por los dispositivos de lógica programable por sí misma deriva en una mayor versatilidad y desempeño, aunado al diseño contemporáneo de la electrónica digital para actualizar los programas de estudio de la Ingeniería Electrónica y afines.

6. BIBLIOGRAFÍA

[1] Sebastian Michael J., *Application - Specific Integrated Circuits*, Addison Wesley, 2000.

[2] Synopsis, *FPGA Express with Verilog HDL and VHDL*, Reference Manual, Xilinx in line, 2002.

[3] John Wakerly, *Digital Design*, Prentice Hall, 2002.

[4] Palacherla Amar., *Using PWM to Generate Analog Output*, Application Notes AN538, Microchip Technology Inc, www.microchip.com.

[5] Seitzer D., *Electronic Analog to Digital Converters*, Wiley, 1990.

[6] *Ejemplos de ABEL – HDL* en línea, <http://eet.etec.wvu.edu/etec373/ABELExs/>

[7] Van der Spiegel J., *ABEL – HDL Primer*, University of Pennsylvania en línea, <http://www.seas.upenn.edu/ese/rca/software/abel/abel.primer.html>.

[8] R. Katz, *Contemporary Logic Design*, Benjamin/Cummings Publ. Comp., Redwood City, CA, 1998.

[9] D. Van den Bout, *Xilinx FPGA Student Manual*, Prentice Hall, Englewoods Cliff, NJ, 1999

[10] Lattice Semiconductors en línea, *ABEL – HDL Reference Manual*, www.latticesemi.com.

[11] Lattice Semiconductors en línea, *ispEXPERT Reference Manual*, www.latticesemi.com.