



Polibits

ISSN: 1870-9044

polibits@nlp.cic.ipn.mx

Instituto Politécnico Nacional

México

Herrera Lozada, Juan Carlos; González Robles, Juan Carlos; Pérez Romero, Patricia

Modulación por Ancho de Pulsos en Lógica Programable

Polibits, núm. 33, 2006, pp. 25-29

Instituto Politécnico Nacional

Distrito Federal, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=402640446004>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# Modulación por Ancho de Pulsos en Lógica Programable

M. en C. Juan Carlos Herrera Lozada,  
M. en C. Juan Carlos González Robles,  
Ing. Patricia Pérez Romero,  
Profesores del CIDETEC – IPN

## TEORÍA DE LA PWM

**E**l presente trabajo muestra el diseño de una unidad PWM (*Pulse Width Modulation*, Modulación por Ancho de Pulso) en lógica programable. Se analiza particularmente la realización del hardware de un Convertidor Digital – Analógico embebido en un PLD.

## INTRODUCCIÓN

Actualmente, 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 para realizar la conversión; sin embargo, la tendencia es considerar diseños de bajo costo sin sacrificar la versatilidad del módulo convertidor. Una unidad PWM es básicamente un DAC (*Digital- Analog Converter*, Convertidor Digital-Analógico) fácil de implementar y que se puede utilizar en sistemas que no requieran una gran velocidad de muestreo, tales como el control de motores, la variación de voltaje en diversas aplicaciones o, en algunos casos, el manejo de señales acústicas de baja calidad.

Es importante reconocer las diferentes alternativas en la implementación; por lo general, se utiliza algún microcontrolador o driver monolítico con funcionamiento específico; si este fuera el caso, sólo es necesario configurar los registros respectivos para acceder a dicha funcionalidad. En este trabajo se consideran dispositivos de lógica programable, dado que presentan 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 FPGA o CPLD).

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 determina si el dato aplicado a la entrada de la unidad es menor al valor binario del contador, mismo que cambia constantemente. El tiempo de la señal en alto ("1" lógico) depende de la cantidad de pulsos de reloj que se aplique 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 valor de la señal de referencia, denominada en la figura como entrada 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.

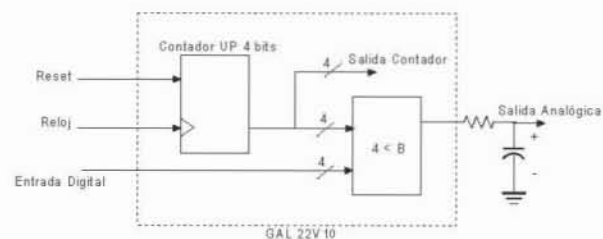


Figura 1. Módulo PWM digital.

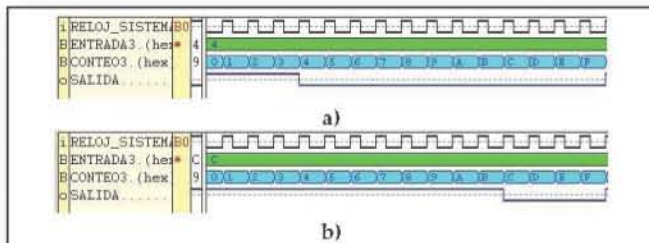


Figura 2. a) Salida PWM para una referencia de valor 4.  
b) Salida PWM para una referencia de valor 12.

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.

## DISEÑO DE LA UNIDAD PWM

Se decidió utilizar un contador ascendente de 4 bits, al igual que un comparador de la misma magnitud, 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:

$$T_{PWM} = (T_{reloj})(2^n) \quad (1)$$

Es posible advertir que si el contador es de 4 bits, se tendrá  $T_{PWM} = 16 T_{reloj}$ , 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 importante cuando se desea realizar una aplicación que requiera sincronizar la señal analógica para controlar un sistema. El programa que modela la unidad PWM en ABEL HDL, se incluye a continuación. El código es fácilmente adaptable a cualquier otra herramienta 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

Alternativamente, en VHDL se puede describir el mismo diseño, como lo muestra la aproximación siguiente:

```
ENTITY pwm IS
PORT(
reloj, reset: in STD_LOGIC;
s3, s2, s1, s0: in STD_LOGIC;
-- entradas a comparar
sal: out STD_LOGIC
);
END pwm;
ARCHITECTURE arch_pwm OF pwm IS
signal cuenta: STD_LOGIC_VECTOR (3 downto 0);
signal entrada: STD_LOGIC_VECTOR (3 downto 0);
begin
entrada(3)<= s3;
entrada(2)<= s2;
entrada(1)<= s1;
entrada(0)<= s0;
--Proceso de comparación y contador
contador: process (entrada, reloj, reset)
begin
if reset='1' then
cuenta <= "0000";
elsif reloj='1' and reloj'event then
cuenta <= cuenta + 1;
if (cuenta < entrada)then
sal<='1';
else
sal<='0';
end if;
else
cuenta <= cuenta;
end if;
end process contador;
END arch_pwm;
```

## DISEÑO DEL FILTRO RC PASABAJAS

Como ya se mencionó, 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 aumentara gradualmente con una frecuencia  $PWM$  aceptable. Considerando una frecuencia de referencia de 60 Hz, con un contador de 4 bits, y sustituyendo en (1), 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:

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

por lo que eligiendo  $C = 1\mu F$ , se obtiene un valor para  $R$  de aproximadamente 42 K $\Omega$  (es posible ajustar a un valor comercial más exacto).

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  $PWM$ , sin considerarlas iguales, tal y como se hizo en la aproximación anterior. En dicha situación será necesario sustituir la frecuencia  $PWM$  por la frecuencia de corte en la (2) o tal vez, hasta considerar un filtro de mayor orden que permita una mayor estabilidad de la señal.

## OSCILADOR EMBEDIDO

Considerando que la frecuencia de la unidad  $PWM$  es fija en dependencia al diseño, se puede proponer un oscilador interno que se acople con una salida RC. 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:

$$f = \frac{1}{2R_1C \left[ \frac{0.405R_2}{R_1 + R_2} \right] + 2R_1C(0.693)} \quad (3)$$

$$\text{Se advierte que con } R_1 = R_2, f = \frac{0.559}{RC} \quad (4)$$

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

$$f = \frac{0.559}{(4.7 \text{ K}\Omega)(100 \mu F)} = 1.1 \text{ Hz} \quad (5)$$

La descripción del oscilador RC se puede incluir en el mismo código de la unidad  $PWM$ .

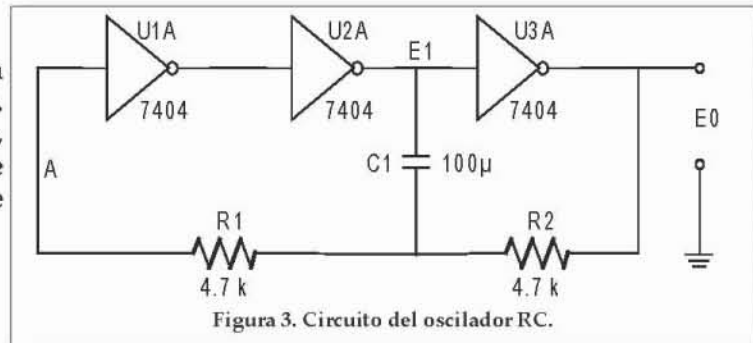


Figura 3. Circuito del oscilador RC.

Se tienen tres entradas fijas (reloj de la intermitencia; A y E1, entradas a los inversores del oscilador RC), tal y como se aprecia en el diagrama anterior.

En el código ABEL que se muestra a continuación, la salida  $EO$  es la entrada de la señal de reloj de la misma unidad  $PWM$ .

```
module led
reloj, A, B, E1, dat3, dat2, dat1, dat0 pin;
salida_pwm pin istype 'com,dc';
C, EO pin 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 de la unidad  $PWM$

## RESULTADOS

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

La unidad fue programada inicialmente sobre un dispositivo 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 4 muestra los resultados de simulación de la unidad diseñada.

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.



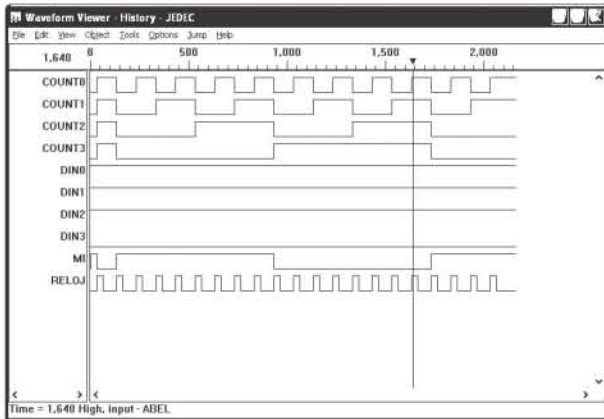


Figura 4. Resultados de simulación para la unidad PWM.

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

La Figura 5, muestra una vista de un 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* Figura 6.

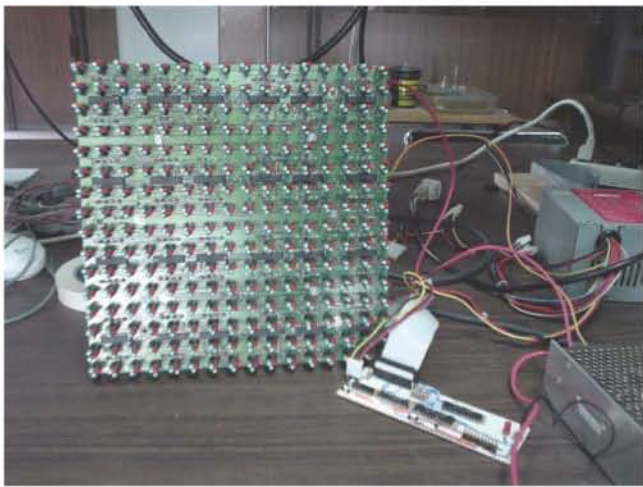


Figura 5. Panel de leds diseñado.

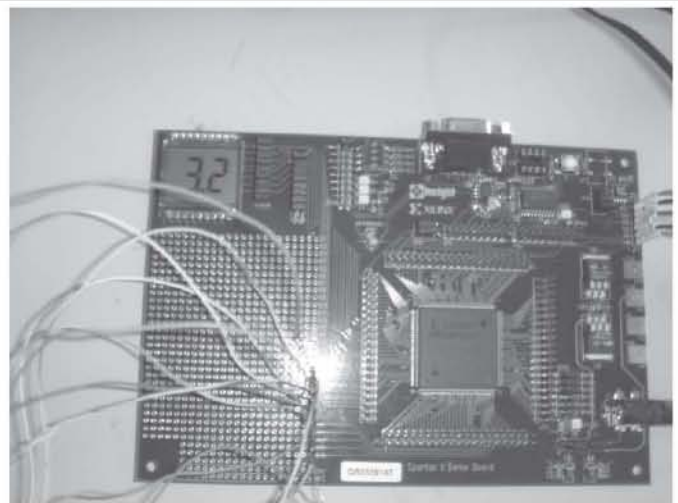


Figura 6. Tarjeta de Desarrollo FPGA Spartan 2 de Xilinx.

### 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 de enseñanza o profesionales que requieran frecuencias de trabajo bajas.

El diseño de convertidores DAC a través de PWM es un recurso ampliamente sustentable en dispositivos de lógica programable. La lógica descrita es muy sencilla y se adapta sin cambios drásticos en el código a cualquier HDL con la alternativa de la arquitectura de los PLD simples, y se hace extensiva para dispositivos de arquitectura avanzada como FPGA y CPLD.

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.

### 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] Wakerly, John. *Digital Design*, Prentice Hall, 2002.
- [4] Palacherla, Amar., *Using PWM to Generate Analog Output*, Application Notes AN538, Microchip Technology Inc, [www.microchip.com](http://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>.