

MODULACIÓN PWM EN FPGA BASADO EN MÁQUINAS DE ESTADO FINITO

PWM modulation in FPGA based on Finite State Machines

RESUMEN

La modulación por ancho de pulsos PWM, consiste en la generación de pulsos que varían en su ciclo útil de acuerdo con el valor de la señal de entrada.

En este artículo se presenta una técnica basada en máquinas de estado finito FSM para modular PWM en dispositivos lógicos programables FPGAs. A la entrada se tiene una señal digital de tres bits y una señal de reloj, y a la salida se obtiene una señal con ciclo útil proporcional al valor de la señal de entrada con una frecuencia igual a 1/9 la frecuencia de la señal de reloj.

La programación de la FSM se realiza en una herramienta CAE de Xilinx, la cual permite generar automáticamente el código VHDL, y finalmente se simula su funcionamiento.

DORA MARÍA BALLESTEROS

Ingeniera Electrónica, M. Sc.

Profesor Asistente

Universidad Militar Nueva Granada

doramaria02@gmail.com

PALABRAS CLAVES: FSM, FPGA, PWM, VHDL.

ABSTRACT

Modulation PWM involves the generation of pulses that vary in their useful cycle according to the value of the input signal.

This article presents a technique based on finite state machines FSM to modulate PWM in FPGAs. At the entrance there is a three-bit digital signal and a clock signal and the output signal is obtained with useful cycle proportional to the value of the input signal with a frequency equal to 1 / 9 the frequency of the signal clock.

The programming of the FSM takes place in a CAE tool of Xilinx, which allows automatically generate VHDL code, and finally simulate their operation.

KEYWORDS: FSM, FPGA, PWM, VHDL.

1. INTRODUCCIÓN

El uso de PWM en el control ha sido ampliamente estudiado en los últimos años, y su implementación tradicionalmente se ha basado en soluciones análogas y microcontroladores.

Recientemente se han utilizado los DSP (procesadores digitales de señal) para la modulación por ancho de pulso pero se ha encontrado la desventaja de la velocidad máxima de operación en esta solución software. Otro campo de estudio para el desarrollo de moduladores PWM, más recientemente estudiado, ha sido la utilización de dispositivos de arreglo de campo programable FPGAs.

En la literatura se encuentran trabajos que utilizan FPGAs de Xilinx, para la generación de PWM trifásicos aplicado a inversores [1], para control de velocidad de motores de inducción utilizando lenguaje VHDL [2], para controladores PID basado en el principio de los moduladores PWM [3], para la realización de controladores SVPWM [4], para control digital realimentado de inversores PWM [5], para PWM de inversores de cascada multinivel [6], para compensación

en inversores PWM [7], entre otros. También se han planteado diseños ASIC como los encontrados en los artículos [8] y [9].

En este trabajo se diseña e implementa un modulador por ancho de pulsos basado en el concepto de máquinas de estado finito y utilizando la herramienta CAE de Xilinx ISE Foundation v8.2. La FPGA seleccionada es una Spartan 3 referencia XC3S200, la cual cuenta con 200.000 compuertas equivalentes.

La señal de entrada de la FPGA es una señal digital de tres bits que ingresa de forma paralela, y la salida es serial correspondiente a la señal modulada PWM. La técnica de modulación PWM seleccionada corresponde al control todo/nada utilizando contadores en la FSM y se presentan dos versiones para la generación de la señal modulada.

Adicionalmente se evalúan los recursos hardware de cada versión, se realiza la simulación para verificar su funcionamiento y se compara la solución propuesta con desarrollos comerciales y académicos ya existentes.

2. MARCO TEÓRICO

2.1. MÁQUINAS DE ESTADO FINITO

Una máquina de estado es un circuito secuencial síncrono que utiliza lógica combinacional y flip-flops, y el valor del “estado” corresponde al valor de las salidas de los flip-flops.

Las máquinas de estado finito, FSM, se dividen en Mealy y Moore. En una máquina de estados Moore, la salida o salidas depende(n) únicamente del valor del estado en que se encuentra; mientras, en una máquina Mealy la salida se ve afectada adicionalmente por el valor de la señal de entrada.

El diagrama de la máquina de estados contiene: los estados, las salidas, las transiciones y las condiciones de transición. Para pasar de un estado a otro es necesario que exista una transición entre los dos en el sentido adecuado y que se cumpla con la condición de la transición.

Una máquina de estados diseñada correctamente, debe cumplir con:

- La suma de las condiciones de salida de cada estado debe ser igual a uno.
- Ninguna combinación de las señales de entrada se puede repetir en dos o más condiciones de salida de un estado.

En el diagrama de la máquina de estados se identifica que es Moore si las salidas están dentro de los estados; y es Mealy si las salidas están en las transiciones.

Un ejemplo de una máquina de estados se presenta en la Figura 1

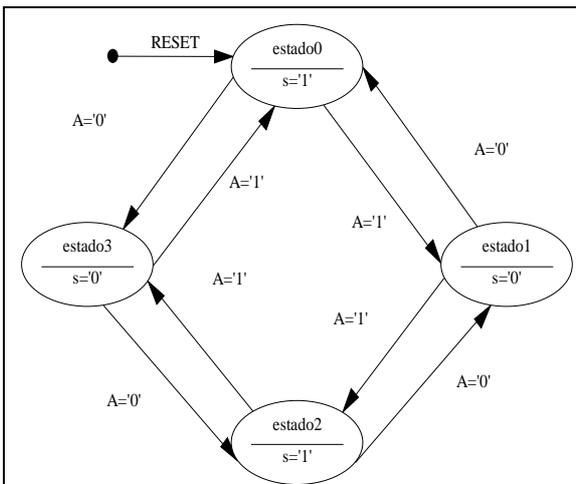


Figura 1. Máquina de estado finita, tipo Moore. Diseño del autor

En la máquina de la figura 1 se tienen cuatro estados, ocho transiciones, una señal de entrada de un bit definida como A y una señal de salida de un bit definida como s.

2.2. MODULACIÓN PWM

En la modulación por ancho de pulsos PWM, se transforma una señal análoga en una señal digital con ancho de pulso proporcional al valor de la señal análoga previamente muestreada. El muestreo de la señal debe ser por lo menos el doble de su ancho de banda, para garantizar la correcta modulación de la señal.

Entre las técnicas tradicionales para modular en PWM se encuentra el control todo/nada, el cual requiere de un contador módulo N, de un comparador con una señal de referencia y un registro. La estructura se presenta en la Figura2.

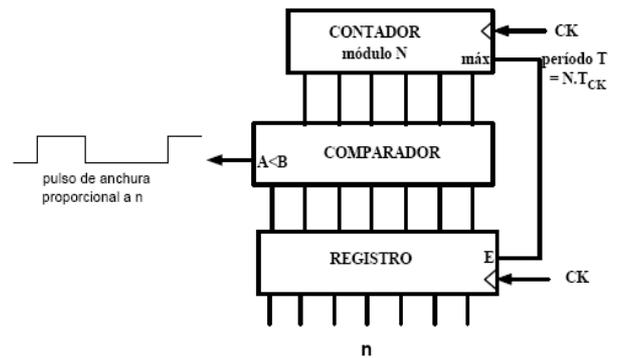


Figura 2. Control todo/nada por conversión número-tiempo. Tomado de [10]

En la Figura 3 se presenta una señal análoga con su correspondiente señal digital PWM, se puede apreciar que a mayor valor análogo de entrada, mayor es el ciclo útil de la señal de salida digital (mayor tiempo alto con respecto al periodo de la señal).

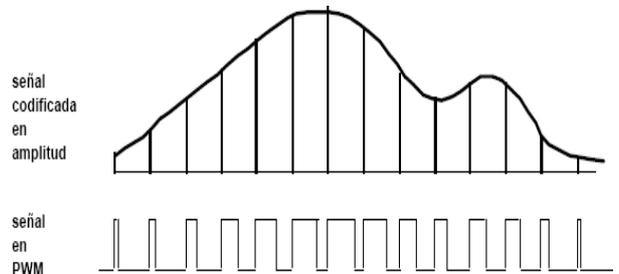


Figura 3. Ejemplo modulación PWM. Señal análoga (arriba) y señal modulada (abajo). Tomado de [10]

3. PWM CON MÁQUINAS DE ESTADO

Las condiciones utilizadas para el modulador PWM son:

- La señal de entrada es una señal digital de tres bits
- La señal de reloj es una señal digital con ciclo útil del 50%
- La señal de salida es digital con ciclo útil de 1/9, 2/9, 3/9,...8/9, dependiendo del valor digital de la señal de entrada. Tabla 1.
- Cada nueve ciclos de la señal de reloj se actualiza la lectura de la señal de entrada de 3 bits.
- El número de estados debe ser el menor posible (menos estados, menos flip-flops).
- La técnica desarrollada para el modulador debe permitir una generalización a un número mayor de bits de conversión de la señal análoga.

Adicionalmente se consideraron las siguientes restricciones:

- El diseño de la FSM debe utilizar recursos menores a los disponibles en una Spartan XC3S200.
- La herramienta CAE utilizada para el gráfico de la máquina de estados, la generación del código VHDL y la simulación del funcionamiento del modulador PWM es ISE Foundation versión 8.3 de Xilinx.

En la Tabla 1 se presenta la variación del ciclo útil del modulador de acuerdo con el valor digital de la señal de entrada. El periodo de cada salida corresponde a nueve veces la señal de reloj.

Entrada	Salida								
	clk 1	clk 2	clk 3	clk 4	clk 5	clk 6	clk 7	clk 8	clk 9
0 0 0	1	0	0	0	0	0	0	0	0
0 0 1	1	1	0	0	0	0	0	0	0
0 1 0	1	1	1	0	0	0	0	0	0
0 1 1	1	1	1	1	0	0	0	0	0
1 0 0	1	1	1	1	1	0	0	0	0
1 0 1	1	1	1	1	1	1	0	0	0
1 1 0	1	1	1	1	1	1	1	0	0
1 1 1	1	1	1	1	1	1	1	1	0

Tabla 1. Entrada y salida del modulador PWM diseñado

3.1. DISEÑO PROPUESTO

Versión 1 (V1):

El diseño de la FSM parte del controlador todo/nada que se utiliza para modular en PWM. En esta caso se utilizan contadores para contar la cantidad de unos consecutivos y ceros consecutivos de la señal modulada, y en vez de utilizar un comparador, se utilizan ramificaciones

paralelas para cada valor digital de la señal de entrada de la Tabla 1.

Adicionalmente se aprovecha la característica que todas las salidas moduladas tienen el primer ciclo de reloj igual a uno y el último ciclo de reloj igual a cero, permitiendo compartir estos dos estados para todos los valores de la señal de entrada digital.

Por ejemplo, para la entrada “000”, se tiene un primer ciclo común igual a uno digital, siete ciclos consecutivos de ceros y un último ciclo común igual a cero digital. Para la entrada “100” se tiene un ciclo común igual a uno, cuatro ciclos consecutivos de unos, tres ciclos consecutivos de ceros, y un ciclo común de cero digital.

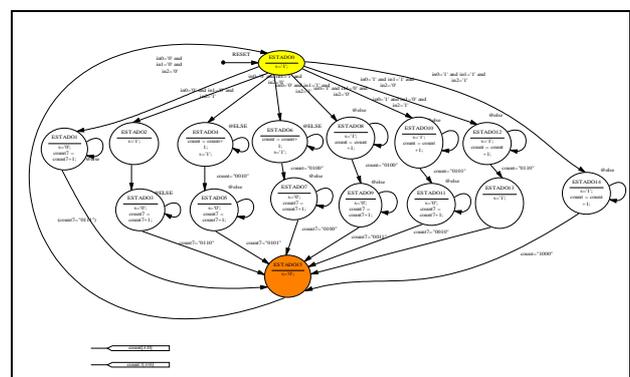


Figura 4. Diagrama de estados de la FSM, versión 1. Diseño del autor

El diseño de la FSM para el caso de tres bits utilizó 16 estados, si se realiza la extrapolación para el caso de 4 bits se utilizarán 32 estados, para 5 bits 64 estados, y así sucesivamente. En términos generales, el número de estados de la FSM de acuerdo con el número de bits de la señal de entrada se rige por:

$$Estados = 2^{n+1} \quad [1]$$

Donde n corresponde al número de bits de la entrada digital.

Se pueden obtener otros diseños para el modulador PWM con el mismo número de bits utilizando contadores de 1's y 0's, por ejemplo una FSM con 8 ramificaciones cada una con 2 estados y en cada estado se cuenta el total de 1's y 0's consecutivos, en cada caso. La desventaja principal de este diseño (que puede ser el obvio en primera instancia), es que al no existir estados en común, se deben colocar transiciones entre el segundo estado de cada ramificación con el primer estado de todas las ramificaciones, aumentando el total de transiciones de la máquina de estados y por consiguiente la lógica combinacional de la misma.

De tal forma que para el caso de 3 bits se necesitarían 81 transiciones, mientras que con el diseño propuesto es

suficiente con 36 transiciones (ahorro del 56% de recursos hardware).

Versión 2 (V2):

Una mejora del diseño consiste en utilizar únicamente dos estados, uno para la generación de 1's y otro para la generación de 0's, y en las transiciones se establece el valor de la señal de entrada con su correspondiente contador de 1's 0's según el caso. Por ejemplo, para la entrada "101" el contador de 1's denominado count llega al valor de 6; mientras que el contador de 0's denominado count7 llega al valor de 3. Todas las combinaciones de la señal de entrada con su respectivo contador se separan por condicionales tipo or. La segunda versión se presenta en la Figura 5.

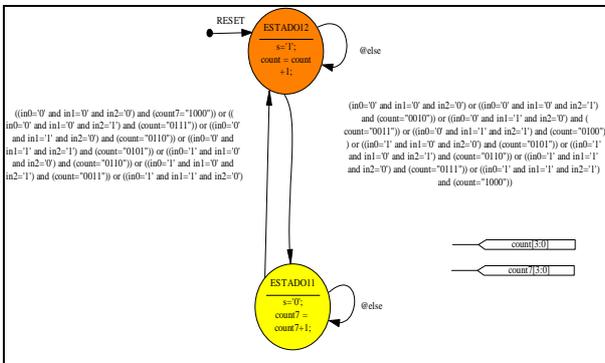


Figura 5. Diagrama de estados de la FSM, versión 2. Diseño del autor

4. RESULTADOS

4.1. SÍNTESIS DEL DISEÑO

Los recursos hardware asignados a la máquina de estados para una FPGA referencia XC3S200, se presentan en la Tabla II, -tomados del reporte automático que genera ISE Foundation al sintetizar la FSM-.

Tabla II. Utilización de recursos (valores estimados)

Recursos	Utilizados		Disponibles	Utilización	
	V1	V2		V1	V2
Número de slices	39	16	1920	2%	2%
Número de flip-flops	25	9	3840	0.6%	0.2%
Número de LUTs de 4 entradas	76	30	3840	2%	1%
Número de IOBs	14	14	173	8%	8%
Número de GCLKs	1	1	8	12%	12%

Comparando las dos versiones de las FSM, se encuentra que la segunda utiliza menos recursos hardware, lo que

permite una implementación de un modulador PWM de mayor número de bits sin necesidad de cambiar de tarjeta, razón por la cual, se escoge este diseño para la presentación de los resultados en este trabajo.

El esquema tecnológico de la máquina de estado versión 2, se presenta en la Figura 6. Cada una de las 30 LUTs que se presentan corresponde a un circuito combinacional de 4 entradas 1 salida; los flip-flops utilizados son de tipo D. Como estructuras macro se tienen: 2 sumadores/restadores, 2 sumadores de 4 bits, 9 registros de desplazamiento y 9 flip-flops.

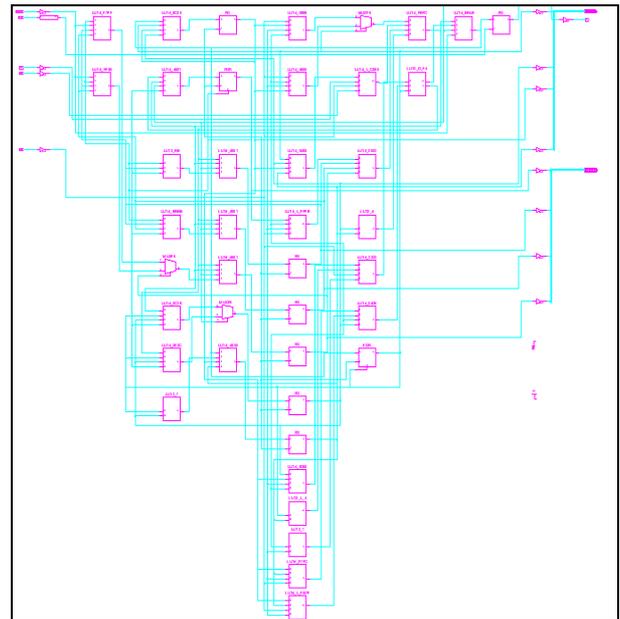


Figura 6. Diagrama esquemático de la máquina de estados diseñada.

Después de la asignación de recursos hardware al diseño de la FSM, la herramienta CAE estima el retardo total del diseño y la frecuencia máxima de operación del mismo. Los resultados obtenidos son:

- Mínimo periodo: 8.05 ns
- Máxima frecuencia: 124.224 MHz
- Mínimo tiempo de arribo de la señal de entrada antes de la señal de reloj: 8.321 ns
- Máximo tiempo de salida requerido después de la señal de reloj: 7.498 ns

Estos resultados corresponden al tiempo en actualizar el estado de la FSM. como en el diseño propuesto se utilizan 9 ciclos de la señal de reloj de entrada, entonces la frecuencia máxima de operación real (frecuencia de muestreo máxima de la señal analoga) es de 14.96 MHz.

4.2. SIMULACIÓN

Utilizando el simulador de máquinas de estado de ISE Foundation versión 8.2 se presenta en la Figura 7 la salida PWM para diferentes valores digitales de la señal de entrada.



Figura 7. Resultados de la simulación de la FSM

De arriba a abajo se presentan los resultados de la señal modulada por ancho de pulso para los valores de la señal de entrada de “001”, “010”, “011”, “100”, “101”, “110” y “111”. Se puede apreciar que a medida que la señal de entrada digital es mayor, el ciclo útil de la señal de salida *s* aumenta. El comportamiento obtenido corresponde al funcionamiento previsto en la Tabla 1. Cada 9 ciclos de la señal de reloj se actualiza el ancho de banda de la señal de salida.

5. DISCUSIÓN

En [11] se presenta un *core* desarrollado por *NASA Office of Logic Design*, el cual consiste en la generación de un *core* para modulación por ancho de pulso con las siguientes características:

- PWM de 16 bits
- Frecuencia máxima: 17.517 M Hz
- Periodo señal modulada: $2^n \cdot T_{clock}$
- Ciclo útil: valor del dato (digital)/ 2^{16}
- Implementado en la FPGA XC4003E
- Recursos utilizados de la FPGA: 16% (33flip-flops)

La utilización del *core* ahorra tiempo de desarrollo del diseñador, pero todos los *core* no están disponibles para todas las referencias de las FPGA existentes, ni se encuentran en todas las plataformas de los fabricantes de estos dispositivos. Comparando el diseño utilizado en el desarrollo del *core* con el propuesto en este trabajo, se concluye que ambos desarrollos permiten frecuencia de muestreo máxima de valor similar, pero la cantidad de recursos de la versión2 es mayor a la obtenida en el *core*.

Un *core* comercial desarrollado por Actel llamado CorePWM [12] utiliza una sencilla máquina de estado finito para controlar el ancho de pulso de salida del *core*. Para utilizarlo se necesita tener instalado en el PC los recursos software Actel Librero IDE y CoreConsole, y tener licenciado el CorePWM.

Se destaca que en ambos casos los *core* utilizan una señal de reloj como referencia de la señal de salida PWM digital, y que la frecuencia de la salida es menor a la frecuencia de la señal de reloj, y ésta depende del número de bits de la señal de entrada o número de ciclos útiles diferentes de la señal de salida. Esta filosofía fue aplicada en el trabajo presentado en este artículo.

Otra opción que tienen los diseñadores de circuitos digitales a la hora de configurar un modulador PWM consiste en la utilización del PicoBlaze de las tarjetas Spartan 3E- Starter Kit de Xilinx [13]. Como desventajas de utilizar el PWM predefinido se tienen: necesidad de utilizar dos ciclos de reloj, programación en lenguaje ensamblador y restricciones de garantía del fabricante relacionadas con el diseño a medida.

En relación con los desarrollos no comerciales, en el ámbito académico se encuentran trabajos relacionados con la generación de PWM en FPGAs basado en máquinas de estado. Ejemplo de ello se tiene el trabajo [14] en el cual la FSM tiene 6 estados y utiliza el 65% de los recursos de una Spartan II XCS2100 (100.000 compuertas equivalentes) y el 10% de I/O con una frecuencia máxima de 3 MHz. Comparando con el diseño de la versión2, ésta tiene un ahorro notable de recursos hardware y frecuencia máxima de operación similar.

6. CONCLUSIONES

Una opción a la hora de diseñar un circuito digital que permita ahorrar tiempo sin depender de soluciones comerciales, que normalmente necesitan de licencia y encarecen los costos del proyecto, consiste en la utilización de herramientas CAE que compilen, simulen, sinteticen e implementen el diseño en un dispositivo lógico-programable. En el caso de circuitos secuenciales, la FSM se convierte en la clave de la solución, y la herramienta CAE es la encargada de generar el código VHDL que se programa en el CPLD o FPGA.

Cuando se necesita un módulo de aplicación específica, puede ser conveniente diseñarlo uno mismo en vez de utilizar un bloque predefinido, ya que algunas veces los *core* trabajan bajo algunas condiciones especiales que no necesariamente se ajustan a las necesidades del usuario. Esto no significa que no se puedan o deban utilizar, si se cuenta con el *core* que se ajusta a los requisitos y está

definido para la tarjeta que se utiliza en el proyecto y se encuentra disponible gratis en la herramienta CAE, es muy conveniente y útil su uso.

7. BIBLIOGRAFÍA

- [1] S Mekhlef y N.A. Rahim. "Xilinx FPGA based three-phase PWM invertir and its application for utility connected PV system", en *Proc. 2002 IEEE Region 10 Conference on computers, communications, control and power engineering*, vol3, pp. 2079-2082.
- [2] D Deng, S Chen, G Joos. "Implementation of PWM pattern generators", en *Proc. Canadian conference on Electrical and computer engineering, 2001*, vol 1, pp. 225-230.
- [3] K Hwu y Y.T. Yau. "Applying a counter-based PWM control scheme to an FPGA-based SR forward converter", en *Proc. 2006 Applied power electronics conference and exposition, APEC'06, Twenty-first annual IEEE*, pp. 5
- [4] Z Yuan, X Fei-peng, Z Zhao-young. "Realization of an FPGA-based space-vector PWM controller", in *Power electronics and motion control conference, 2006, IPEMC'06*, vol1, pp.1-5
- [5] T Sugure, F Takahiro, Y Tomokj. "Variable sampling quasi multirate deadbeat control method for single phase PWM inverter in low carrier frequency", en *Proc. Power conversion Conference, PCC'07*, pp.804-809.
- [6] S.M. Ayob, Z Salom, A Jusoh. "Trapezoidal PWM scheme for cascaded multilevel inverter", en *Proc. First International Power and energy conference, PECon 2006*, pp. 368-372.
- [7] K Rauma, et al. "FPGA based dead time compensation for PWM inverters", en *Proc. European Conference on Power Electronics and applications, 2005*, pp.7
- [8] A.M. Omar, N.A. Rahin, S Mekhilef. "Three-phase synchronous PWM for Flyback converter with power factor correction using FPGA ASIC design", *IEEE Trans. Industrial electronics*, vol. 51, pp. 96-106, Feb. 2004.
- [9] S.J. Kim et al. "ASIC design DTC based speed control of induction motor", en *Proc. IEEE International symposium on industrial electronics, 2001*, vol2, pp. 956-961.
- [10] T Pollán. Capítulo 18: Modulación de anchura de pulsos. Universidad de Zaragoza pp. 160-176. Disponible en: <http://www.unizar.es/euitiz/areas/aretecel/docencia/digitel/Apuntes/dig18.pdf>
- [11] S Thangjan, S Attri, BS Sohi. "Efficient FPGA Implementation of PWM core", presentado en el MAPLD International Conference, Estados Unidos, 2004. Disponible en: http://klabs.org/mapld04/abstracts/abstracts_04.htm
- [12] Actel. Configuring CorePWM using RTL Blocks. Application note. Sep 2006. pp. 1-13. Disponible en: http://www.actel.com/documents/CorePWM_RTL_AN.pdf
- [13] K Chapman. Software Implementation of Pulse Width Modulation (PWM). A reference design using the Spartan 3E Starter Kit. Rep Tec. Xilinx Ltda. 24, May. 2006. Disponible en: http://www.xilinx.com/products/boards/s3estarter/files/s3esk_picoblaze_pwm_control.pdf
- [14] A, Arbit et al. "A DSP-controlled PWM generator using field programmable gate array", en *Proc. 2004 23rd IEEE Convention of Electrical and Electronics Engineers*, Volume , Issue , pp. 325 – 328
- [15] ISE Help. Working with Architecture Wizard IP