

# DESARROLLO E IMPLEMENTACIÓN DE LA INTERFACE SBA PARA UN NÚCLEO PWM DE 16 CANALES INDEPENDIENTES PROGRAMABLES

## DEVELOPMENT AND IMPLEMENTATION OF THE SBA INTERFACE FOR A 16 INDEPENDENT PROGRAMMABLE CHANNELS PWM IP CORE

RENZO BERMÚDEZ Y MIGUEL RISCO

CENTRO DE INVESTIGACIÓN Y DESARROLLO EN INGENIERÍA (CIDI) DE LA FACULTAD DE INGENIERÍA ELECTRÓNICA Y MECATRÓNICA, UNIVERSIDAD TECNOLÓGICA DEL PERÚ, LIMA I

### RESUMEN

Los IP-Cores (Núcleos de Propiedad Intelectual) son para el diseño de hardware lo que las librerías son para la programación de computadoras. Se suelen utilizar en la forma de un circuito discreto integrado, donde la “placa de circuito” es un diseño más grande en ASIC o en FPGA. Un núcleo de propiedad intelectual a menudo adopta la forma de un programa de computadora escrito en el HDL, tales como Verilog, VHDL o SystemC. Idealmente, un IP-Core debe ser totalmente “portable”, es decir, que fácilmente se pueda adaptar a cualquier tecnología de otros proveedores o diferentes métodos de diseño. Los Receptores/Transmisores Asíncronos Universales (UART), las Unidades Centrales de Procesamiento (CPU), los Controladores Ethernet, las Interfaces PCI, son algunos ejemplos de IP-Cores. En este trabajo, se presenta la adaptación de un IPCore PWM de 16 canales a una estructura de bloques independientes similar a los SoC (System on Chip). No se ha implementado un microprocesador como maestro del sistema; en su lugar una máquina de estado compleja administra un bus con la finalidad de ahorrar recursos en la FPGA. Esta máquina de estado compleja, que hace las veces de controlador del sistema, se encuentra dentro de una disposición a la que se le denomina SBA (Simple Bus Architecture) o Arquitectura Simple de Bus, la cual no es más de una simplificación de las señales y reglas que establece la especificación Wishbone. El sistema así integrado permite la configuración de 16 salidas digitales PWM independientes en modo de bajo rizado. Si bien en el ejemplo que se presenta en este trabajo muestra un solo IPCore PWM instanciado, esto no supone un límite. El núcleo PWM implementado no hace uso de recursos específicos o especiales de la FPGA, lo que permite que la cantidad de bloques instanciados pueda crecer tanto como bloques genéricos configurables en la FPGA se encuentren disponibles. Es decir, por cada núcleo instanciado se dispondrá de 16 canales PWM independientes que poseerán una posición de programación específica dentro del mapa de direcciones del SBA.

**Descriptors:** *FPGA, PWM, System on Chip*

### ABSTRACT

*IP Cores (Intellectual Property cores) are for hardware design what libraries are for computer programming. They are typically used in the style and form of a discrete integrated circuit,*

where the “circuit board” is a larger design in ASIC or FPGA. A core intellectual property often takes the form of a software program written in HDL such as Verilog, VHDL or SystemC. Ideally, an IP-Core must be fully portable, meaning that it can be easily adapted to any technology from other suppliers or different design methods. Receivers/Transmitters Universal Asynchronous (UART), Central Processing Units (CPU), Ethernet Controllers, Interfaces PCI are examples of IP-Cores. This paper presents the adaptation of a 16-channel PWM IPCore to a separate brick structure similar to SoC (System on Chip). We did not implement a microprocessor as master of the system, instead a complex state machine runs a bus in order to save resources in the FPGA. This complex state machine that acts as the controller of the system is within a provision which is called SBA (Single Bus Architecture), which is just a simplification of the signals and rules establishing the Wishbone specification. The system thus allows the configuration of 16 independent PWM digital outputs in low ripple mode. While the example presented in this work shows a single PWM IPCore instantiated this is not a limit. The implemented PWM core does not use specific or special resources of the FPGA, which allows that the number of instantiated blocks can grow as much as configurable generic blocks in the FPGA become available. That is, for each instantiated core there will be 16 independent PWM channels that will have specific preset positions within the address map of the SBA.

**Keywords:** FPGA, PWM, System on Chip

## INTRODUCCIÓN

En la actualidad existen muchos circuitos integrados en los que se implementa la modulación por ancho de pulso «PWM» [1]. Los microcontroladores poseen también facilidades para trabajar con señales pwm; pero estos integrados presentan grandes limitaciones a la hora de elaborar proyectos de mayor envergadura, como por ejemplo, las señales de salida pwm están limitadas por la cantidad de pines de dicho integrado y por la cantidad de memoria que presenta el microcontrolador, entre otras. Pensando en estas y en otras desventajas, Actel presenta una solución. Diseña el IP-CorePWM [2], que al implementarse dentro de una FPGA, elimina los problemas previamente mencionados, brindando al mismo tiempo facilidades al programador a la hora de diseñar. Por tratarse de un IP-Core, la “portabilidad” del mismo está garantizada, pudiéndose instanciar el núcleo cuantas veces sea posible, encontrándose como única limitación la cantidad de celdas lógicas disponibles dentro de la FPGA.

## MÉTODO EXPERIMENTAL

### CorePWM v4.1

Es el núcleo desarrollado por Actel, que consiste en un modulador de ancho de pulso multicanal de

propósito general. Tiene dos modos de configuración: modo PWM, de propósito general; y modo DAC, de bajo rizado. Para nuestro propósito, usaremos el segundo modo por su mucho mayor precisión, siempre que se use un filtro pasa-bajo, como el R-C, que permite un mejor aprovechamiento en ancho de banda y rizado que en un algoritmo estándar de PWM. Entre las principales características del núcleo figuran las siguientes:

- Permite obtener hasta 16 salidas digitales PWM independientes, configuradas a través de una interface de registro;
- Todas las salidas son controladas por doble flanco;
- Configuración basada en APB bus; y
- Permite también configurar hasta 16 entradas digitales de tacómetro, a través de una interface de registro.

En la figura 1 se muestran las señales de entrada y de salida del núcleo. En la figura 2 se presentan ejemplos de configuración del núcleo para ambos modos de configuración: PWM1 – PWM4, modo PWM de propósito general, y PWM5, modo DAC de bajo rizado. Se carga el valor 0x3F al respectivo registro del canal que se quiere configurar para obtener a la salida 25% Duty Cycle. Para el 50% Duty Cycle se cargaría 7F (valores desde 0 hasta FF equivalen a 0-100%).

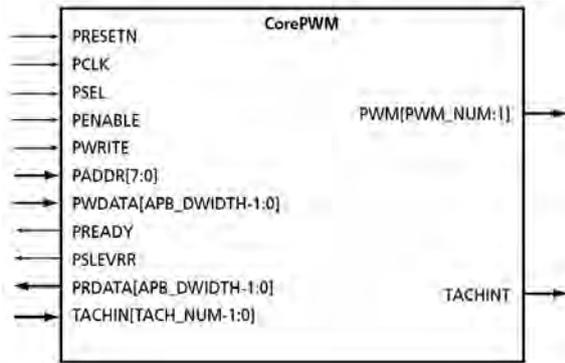


Figura 1: Diagrama de señales E/S.

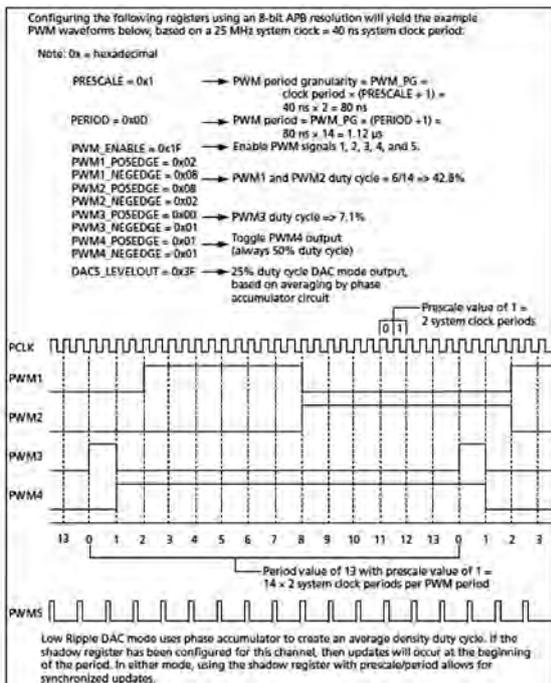


Figura 2: Ejemplos de configuración.

### Hardware de la PEB

ProAsic3E Evolution Board «PEB» es la tarjeta de desarrollo usada en este proyecto, que cuenta con una FPGA A3P31500-PQ208 de 1,5M celdas lógicas y 208 pines. Tiene cuatro bancos de ocho leds cada uno, permitiendo visualizar hasta 32 leds, ideales para este proyecto. Entre otras características, esta tarjeta de desarrollo cuenta con un LCD 2x16, un reloj oscilador de 40 MHz, dos puertos UART, entre otros. En la figura 3 se muestra una imagen de la PEB.

### SBA

El SBA o Simple Bus Architecture permite interconectar diferentes componentes o bloques

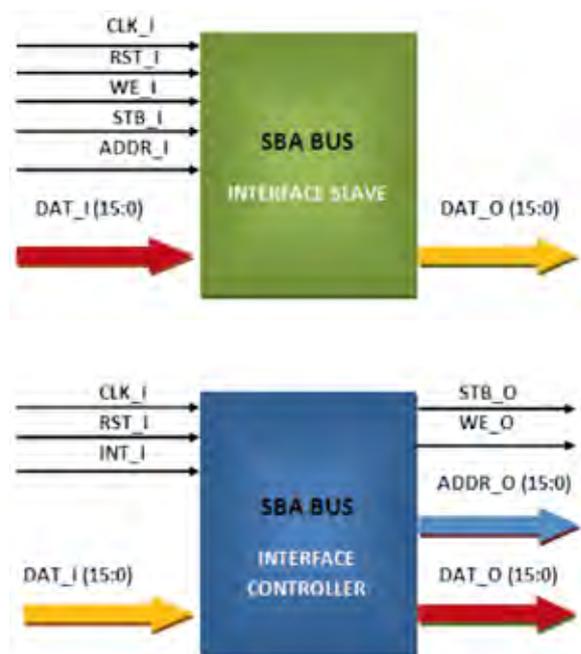


Figura 3: ProAsic3E Evaluation Board «PEB».

entre sí de manera práctica y sencilla, optimizando el tiempo a la hora de realizar los proyectos, sacrificando las complejidades que muestran otras implementaciones con la finalidad de introducir con facilidad el concepto SoC en la FPGA, por lo que tiene un valor educativo inherente.

El SBA pretende ser una interface de propósito general; como tal, define el intercambio de datos entre módulos estándar de IP Cores.

Se divide en tres tipos de bloques principales: maestro o controlador del sistema (SBA Controller), núcleos esclavos y núcleos de apoyo al bus (decodificador de direcciones, adaptadores de bus, generadores de reloj, etc.) (fig. 4).



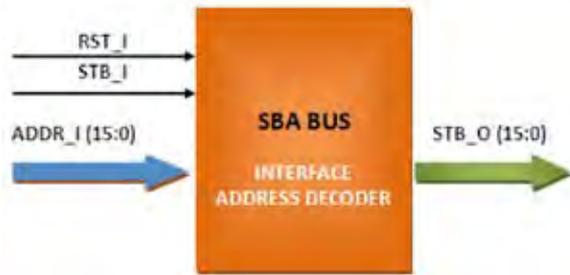


Figura 4: SBA.

Descripción funcional de las señales SBA

- RST\_I: Recibe la señal de reset principal del sistema
- CLK\_I: Recibe la señal de reloj de sincronismo del sistema
- ADDR\_I: Recibe la dirección del SBA Controller
- ADDR\_O: Envía la señal a los demás bloques
- DAT\_I, DAT\_O: Bus de datos
- INT\_I: Solicitud de Interrupción
- WE\_I, WE\_O: Habilitación de Escritura
- STB\_I, STB\_O: Selector de núcleo (Strobe)

Descripción funcional de los bloques SBA

- SBA Controller: Es el bloque principal, pues aquí se asigna una dirección para cada acción a realizar. Se encarga también de decidir el orden de dichas acciones (decide qué se ejecuta primero y qué después). Este direccionamiento se comunica directamente con el Address Decoder.
- Address Decoder: Asigna a cada núcleo esclavo una posición en el mapa de direcciones y habilita dichos núcleos mediante el STROBE (Chip Select).
- Esclavos: Periféricos a usar (componentes, bloques, IP Cores). Envían y reciben datos.

## RESULTADOS Y DISCUSIÓN

Como un ejemplo aplicativo para el CorePWM v4.1, se desarrolló un ejercicio en el cual, configurando los 16 canales PWM, se puede observar en 16 leds respectivamente de la PEB, el efecto de una ola que mediante un generador de rampa proporciona los datos para configurar los canales PWM, que sirven para controlar la intensidad luminosa de los leds.

Vale recalcar que este proyecto se realizó respetando las especificaciones del bus SBA.

En la figura 5 se aprecia cómo están conectados los bloques mediante las señales SBA.

Estructura del sistema descrito en VHDL:

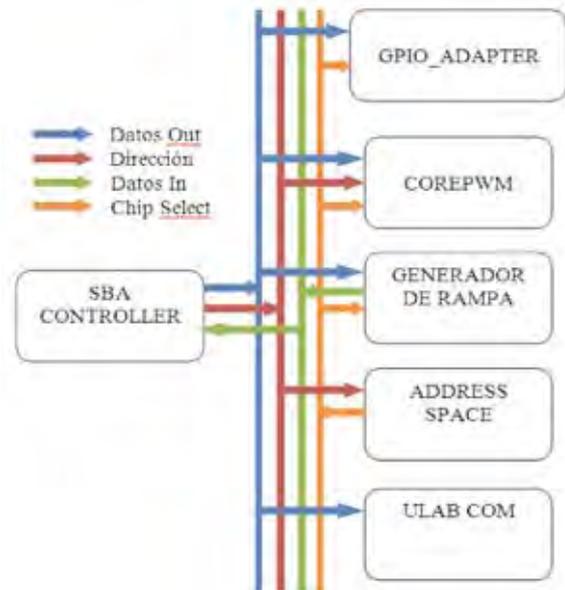


Figura 5: Diagrama de bloques.

Descripción funcional de los bloques:

- SBA Controller: Realiza la administración general del bus. Dirige el orden del flujo de la información según la secuencia determinada por su máquina de estado interna.
- Address Space: Genera el mapeo de las direcciones, y a cada esclavo una línea para su habilitación mediante una señal STROBE (Chip Select).
- Generador de rampa: Entrega los datos para controlar la intensidad de luminosidad de los leds.
- CorePWM: Mediante una dirección y un dato que recibe como entradas, configura un canal de PWM.
- Adaptador GPIO.
- uLAB Com: Permite la comunicación de la PEB con la PC, para poder visualizar en el monitor los datos que han sido enviados por el generador de rampa.

La figura 6 muestra en los últimos dos bancos de leds de la derecha (verdes) el efecto de la ola que se va desplazando. Muestra también en el primer banco de leds (rojos) los datos que envía el generador de rampa.

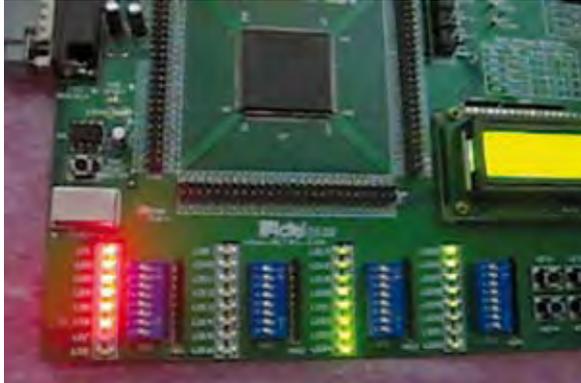


Figura 6: 16 salidas PWM mostrando diferentes intensidades de luminosidad.

### CONCLUSIONES

Después de haber logrado obtener 16 salidas digitales PWM para controlar 16 leds de manera independiente, se plantea como futura aplicación poder controlar leds RGB, manipulando todos los 16.777.216 posibles colores que se pueden obtener.

Una vez logrado controlar los leds RGB, se puede implementar paneles luminosos publicitarios de gran tamaño e instanciando varias veces el CorePWM se puede controlar miles de leds, a diferencia de un microcontrolador que limita la cantidad de salidas.

### AGRADECIMIENTOS

Al profesor Miguel Risco, por su gran apoyo y paciencia durante todo el tiempo que tomó realizar este proyecto, y por aportar con la PEB para seguir con las investigaciones.

### REFERENCIAS

- [1] Definición IP-Core. [En línea] [http://whatis.techtarget.com/definition/0,,sid9\\_gci759036,00.html#](http://whatis.techtarget.com/definition/0,,sid9_gci759036,00.html#).
- [2] Definición PWM, Wikipedia en español: [http://es.wikipedia.org/wiki/Modulaci%C3%B3n\\_por\\_ancho\\_de\\_pulsos](http://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_ancho_de_pulsos).
- [3] CorePWM v4.1 HandBook. [En línea] <https://www.actel.com/techdocs/hb/ip.aspx>.
- [4] M. Risco, Desarrollo de instrumentos basados en instrumentación virtual reconfigurable, ECIPERU 2009, vol. 6, número 2, 22-27.

E-mail: renzobond@hotmail.com