Ejemplo de diseño FPGA para medidas de máximas frecuencias de operación

Carlos Jiménez Fernández, Pilar Parra Fernández, Carmen Baena Oliva, Manuel Valencia Barrero
Dpto. Tecnología Electrónica
Universidad de Sevilla
Instituto Microelectrónica de Sevilla
IMSE-CNM CSIC/US
Sevilla, España
{cjesus, parra, baena, manolov}@imse-cnm.csic.es

F. Eugenio Potestad Ordóñez
Instituto Microelectrónica de Sevilla
IMSE-CNM CSIC/US
Sevilla, España
potestad@imse-cnm.csic.es

***Abstract*—La mejor forma de prender a diseñar sistemas digitales a nivel RT es haciendo uso de ejemplos prácticos. Además, desde el punto de vista docente, cuanto más prácticos, más atractivos son para los alumnos. Pero para que un diseño sea atractivo, aunque se plantee con una baja complejidad, no es posible realizarlo en una única sesión de prácticas. En esta comunicación se presenta, a modo de demostrador, el diseño a nivel RT y su implementación en FPGA de un sistema digital que utiliza el cifrador de flujo Trivium y sobre el que se hacen medidas de su frecuencia máxima de operación. El diseño de este circuito se realiza en tres sesiones de prácticas de unas dos horas de duración cada una.**

Keywords— VHDL, diseño de sistemas digitales, FPGA, Chipscope.

I. INTRODUCTION

Las asignaturas que enseñan diseño digital en cursos avanzados de titulaciones electrónicas pueden utilizar lenguajes de descripción de hardware para la descripción de circuitos y dispositivos FPGA como tecnología para probar de forma experimental el comportamiento de los circuitos diseñados. Esta metodología tiene muchas ventajas. Una de ellas es la utilización de un único entorno de CAD para el diseño, la verificación y la programación de los dispositivos. Además este entorno de CAD es ofrecido de forma gratuita para uso docente por los fabricantes de FPGA. Otra ventaja es la disponibilidad de placas de desarrollo que incluyen, además de la FPGA, elementos de visualización e interfaz que permiten introducir valores a las entradas y ver los valores de las salidas. Por todo esto la enseñanza del diseño digital con el tándem VHDL-FPGA es una opción con un coste bajo y que, sobre todo, permite la comprobación experimental de los diseños lo que tiene un alto atractivo para los alumnos.

Esta alternativa se está aplicando en la asignatura "Diseño Digital Avanzado" [1], optativa de cuarto curso de la titulación del Grado en Electrónica Industrial impartida en la Escuela Politécnica Superior de la Universidad de Sevilla. En esta asignatura se pretende que el alumno adquiera las competencias más importantes del diseño digital. Los únicos conocimientos previos sobre los que se construye esta asignatura son las asignaturas "Electrónica Industrial" y "Electrónica Digital", ambas de segundo curso.

La asignatura "Electrónica Industrial" es una asignatura del bloque de formación común de la rama industrial, impartida en el primer cuatrimestre del segundo curso. Es la primera asignatura que tienen los alumnos en esta titulación relacionada con la electrónica. Los contenidos de esta asignatura [2] se dividen en dos bloques: un bloque analógico y un bloque digital. En el bloque analógico se estudian las operaciones de amplificación y filtrado utilizando amplificadores operacionales. En el bloque digital se introducen los conceptos básicos de la electrónica digital, desde el álgebra de conmutación hasta el diseño de máquinas de estado, pasando por los conceptos de puertas lógicas y biestables.

La asignatura "Electrónica Digital" es una asignatura obligatoria en el plan de estudios de la titulación. Se imparte en el segundo cuatrimestre del segundo curso. Sus contenidos [3] desarrollan los iniciados en Electrónica Industrial. Incluyen características reales de las puertas lógicas y los biestables, análisis y diseño de circuitos digitales (combinacionales y secuenciales) y subsistemas combinacionales y secuenciales. En los últimos temas de la asignatura se introducen conceptos relacionados con el diseño a nivel RT (estructura de circuitos basadas en unidad de control y unidad de datos), cartas ASM y principios básicos de microprocesadores. Aunque se hacen algunas prácticas de laboratorio utilizando dispositivos FPGA, los diseños se hacen utilizando el entorno de captura de esquemas y no se utilizan lenguajes de descripción de hardware.

Con estos conocimientos previos por parte de los alumnos, en la asignatura "Diseño Digital Avanzado" se enseña a los alumnos la descripción de sistemas digitales usando el lenguaje de descripción de hardware VHDL y la forma de implementarlos sobre dispositivos FPGA (en nuestro caso de Xilinx). La asignatura se plantea de forma muy práctica, de manera que en una sesión de dos horas se combina una parte teoría con otra parte de laboratorio, en la que los alumnos comienzan haciendo el diseño de pequeños circuitos. Conforme avanza la asignatura se reduce el tiempo dedicado a la parte de teoría y los laboratorios van subiendo de complejidad, pero teniendo en cuenta siempre que la complejidad no sea muy grande para que dé tiempo al alumno a completar la práctica en la duración asignada a los laboratorios (en nuestro caso, un máximo de dos horas).

El límite de dos horas parece que obliga a no desarrollar sistemas mínimamente complejos o a relegarlos a una metodología de *aprendizaje basado en proyectos* [4][5]. Sin embargo, una solución para este problema es la reutilización de los diseños o la realización de un diseño en varias sesiones. Para que esta solución sea interesante debe tener varias características: por una parte cada una de las partes debe ser autocontenida; en cada sesión de laboratorio se debe poder conseguir un objetivo de diseño. Las distintas sesiones de laboratorio deben ir construyendo de forma paulatina la funcionalidad del sistema completo. Al diseño del sistema final se puede llegar tanto descomponiéndolo en partes más simples (*metodología del divide y vencerás*) como construyendo de poco a mucho (*metodología bottom-up*).

En esta comunicación presentamos a modo de demostrador un conjunto de tres prácticas que tienen como objetivo el análisis temporal y la medida de la máxima frecuencia de funcionamiento de un sistema digital medianamente complejo, combinando herramientas de CAD y mediciones experimentales. La realización de estas tres prácticas conlleva, además, el manejo de bloques generadores de señales de reloj de los dispositivos FPGA (para obtener relojes de frecuencias altas) y también el manejo de la herramienta Chipscope, que es un analizador lógico empotrado, con el que se pueden visualizar señales internas durante el funcionamiento de un circuito.

La estructura de esta comunicación es como sigue: en el segundo apartado se explica brevemente el diseño a realizar. En el apartado III se detallan los contenidos y los objetivos de diseño de cada una de las sesiones en las que se ha dividido el diseño así como los resultados que se pretenden obtener. Finalmente se extraen algunas conclusiones.

II. DISEÑO A REALIZAR

En este conjunto de prácticas se ha tomado como diseño base el cifrador de flujo Trivium [6]. Los cifradores de flujo son circuitos criptográficos de clave privada que se emplean para el cifrado de información bit a bit. Su función es generar una secuencia pseudo-aleatoria utilizando una clave (secreta) y un vector de inicialización (IV) que puede ser público. El mensaje se cifra haciendo la operación XOR entre el mensaje sin cifrar y la secuencia pseudo-aleatoria generada. En el receptor, el descifrado del mensaje se hace de la misma forma, mediante una operación XOR entre el mensaje cifrado y la misma secuencia pseudo-aleatoria.

Por ello el proceso de cifrado y descifrado es relativamente sencillo e igual en el cifrador y el descifrador. El mismo circuito es utilizado en el proceso de cifrado y en del descifrado. La principal dificultad a la hora de utilizar este tipo de cifradores es que deben sincronizar muy bien su funcionamiento. La variación en un solo ciclo de reloj entre cifrador y descifrador hace que el proceso de descifrado sea incorrecto. También hay que tener en cuenta que cada vez que se reinicie el funcionamiento del cifrador hay que utilizar un nuevo vector de inicialización, ya que la utilización del mismo vector de inicialización crea un importante problema de seguridad.

La estructura de circuito de los cifradores de flujo suele estar basada en registros de desplazamiento con realimentaciones no lineales. El cifrador Trivium es uno de los cifradores de flujo finalistas del Proyecto Europeo eSTREAM [7], cuyo objetivo fue seleccionar nuevas propuestas más seguras de cifradores. El cifrador Trivium está optimizado para su realización en hardware. Consta de tres registros de desplazamiento, que suman un total de 288 bits, y algunas realimentaciones (lineales y no lineales). Al registro con los 288 bits se le suele llamar registro de estado. Requiere una clave (key) de 80 bits, que debe ser secreta y conocida sólo por cifrador y descifrador, y un vector de inicialización (IV), también de 80 bits, que puede ser público. La estructura de circuito se muestra en la Fig. 1. Inicialmente el registro de estado se carga con la clave y el IV y se dejan pasar 1152 ciclos de reloj antes de poder generar una secuencia de salida (key_stream) válida. En el cifrado de un dato, el texto plano que soporta el dato es combinado (con una operación XOR) con la señal key_stream de salida del Trivium generando así la secuencia de texto cifrado. En el descifrado, este texto cifrado es combinado (también con una operación XOR) con la señal key_stream de salida del Trivium recuperando así la secuencia de texto plano original.

III. DESARROLLO DE LAS PRÁCTICAS

Las tres prácticas que se van a desarrollar tienen los siguientes objetivos parciales:

- Práctica 1: Diseño del cifrador Trivium: Verificación funcional, imposición de restricciones temporales y análisis temporal estático.
- Práctica 2: Creación de una máquina de estados que controle la carga de la clave y el IV y el funcionamiento del Trivium. Introducción de un módulo DCM (*Digital Clock Manager*) para generar una señal de reloj con distintas frecuencias. Verificación de la máxima frecuencia de funcionamiento con simulaciones post-route.
- Práctica 3: Implementación del diseño en una placa Nexys4 DDR y utilización de Chipscope para medir experimentalmente la máxima frecuencia de operación.

Para la realización de los laboratorios se utiliza la herramienta Xilinx ISE, versión 14.7, y la placa de Digilent Nexys-4 DDR.

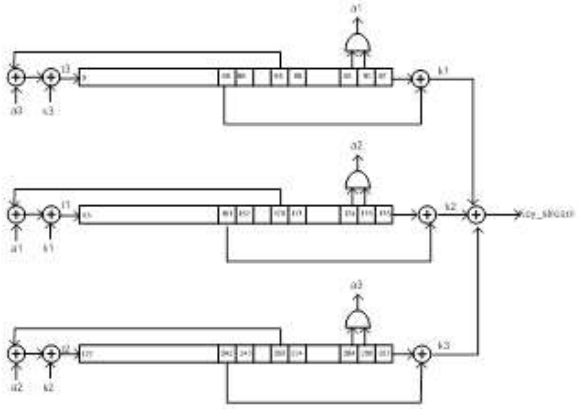


Fig. 1. Estructura de circuito del cifrador Trivium.

A. *Primera práctica: Diseño del cifrador Trivium*

A la vista de la estructura de circuito mostrada en la Fig. 1, el diseño en VHDL del Trivium es bastante sencillo. Va a tener como entradas el reloj (*clk*), una señal de reset asíncrona activa en baja (*reset*) y dos entradas de control: *ce* que va a funcionar como entrada de habilitación de funcionamiento y *ld* que va a controlar la carga de la clave y el vector de inicialización. Para simplificar el diseño, la clave (*key*) y el vector de inicialización (*IV*) van a estar definidos como constantes dentro del cifrador. Como salida tiene la señal *key_stream*, por donde sale la secuencia pseudo-aleatoria generada por el cifrador.

El código va a contener, entre otras cosas:

- La declaración de una señal *state* que contenga el registro de estados (288 bits) y la declaración de las constantes *key* e *IV* con los valores prefijados en la memoria de la práctica:

```
key = X"0F62B5085BAE0154A7FA"
iv  = X"F67A079428CF0AF960C7"
```

- La generación, de forma combinacional y concurrente de unas señales intermedias (a1, a2, a3, k1, k2, k3, t1, t2, t3) y del key_stream (ver Fig. 1):

```
k1 <= state(65) XOR state(92);
k2 <= state(161) XOR state(176);
k3 <= state(242) XOR state(287);

a1 <= state(90) AND state(91);
a2 <= state(174) AND state(175);
a3 <= state(285) AND state(286);

t1 <= k1 XOR a1 XOR state(170);
t2 <= k2 XOR a2 XOR state(263);
t3 <= k3 XOR a3 XOR state(68);

keystream <= k1 XOR k2 XOR k3;
```

- Un proceso síncrono, disparado por el flanco de subida del reloj y con reset asíncrono activo en alto. Cuando el reset esté activo se debe poner a cero todo el registro de estado. Cuando *ce* esté a cero el cifrador debe mantener su estado. Cuando *ce* esté a uno, si *ld* está a uno debe hacer la carga de *key* e *IV*, y si *ld* está a cero debe operar haciendo los desplazamientos.

La carga de la clave y del IV se realiza según las especificaciones del cifrador Trivium [6].

Además del diseño del cifrador, el alumno debe crear un test_bench para su verificación funcional. El test_bench, además de generar las señales para realizar una carga de la clave y el IV, debe hacer funcionar el cifrador durante 1000 ciclos de reloj, pararlo durante 10 ciclos (con la desactivación de la señal *ce*) y hacer que siga funcionando de forma indefinida.

Una vez realizados el diseño del cifrador Trivium y del test_bench, deberá realizar simulaciones y medidas:

- Realización de una simulación funcional.

De la simulación funcional se pide anotar los bits más significativos (en hexadecimal) del registro de estado (señal state) en el tiempo en el momento en el que tras 1000 ciclos de reloj se para el cifrador. Este valor va a servir para conocer la frecuencia máxima de operación.

- Realización de una implementación.

Y análisis de los informes ofrecidos por la herramienta de Xilinx buscando: Recursos consumidos (slices registers, luts usados como lógica, % de ocupación de la FPGA), así como la frecuencia máxima de operación y los tiempos de retraso entre las entradas y el flanco de reloj. También se pide anotar el tiempo desde el flanco de reloj hasta la salida.

- Realización de simulaciones post-route.

El objetivo es conocer el mínimo periodo de reloj del circuito. Para ello se pide realizar una primera simulación post-route con un reloj de 10 ns de periodo. Comprobar si el comportamiento es correcto comparando los valores del registro de estado con los anotados para la simulación funcional. Después se va reduciendo el periodo del reloj hasta que el contenido del registro de estados no coincida con el anotado. Se apunta el mínimo valor del periodo para el que se tiene un funcionamiento correcto, que será utilizado en próximos laboratorios.

- Imposición de restricciones temporales.

Se impone una restricción temporal sobre la señal de reloj de 5 ns de periodo. Se realiza una nueva implementación y se analizan los informes para saber si se ha podido cumplir con dicha restricción. Se vuelve a realizar una simulación post-route y se comprueba el periodo mínimo en el que el funcionamiento es correcto.

Al finalizar esta práctica no sólo se ha realizado el diseño y la verificación funcional del cifrador Trivium, sino que también se han obtenidos datos temporales con simulaciones post-route y análisis temporal estático.

B. Segunda práctica: Utilización de generadores de relojes (DCM)

Esta segunda práctica tiene como objetivo la creación de una máquina de estados que genere señales de entrada para el cifrador Trivium, así como generar la señal de reloj del Trivium mediante un MMCM. Sin embargo, para simplificar el desarrollo del laboratorio, se proporciona a los alumnos el diseño con la máquina de estados.

La máquina de estados que se les proporciona tiene la siguiente funcionalidad: realiza la carga de la clave y del IV, pone en funcionamiento al Trivium y espera que pasen 1023 ciclos de reloj, pone una salida especial a uno durante un ciclo de reloj, y hace que cifrador Trivium siga funcionando de forma continua hasta que se active una señal de reset. Esta salida especial servirá referencia para comprobar si el key_stream generado es correcto o no.

El procedimiento a seguir en este laboratorio es el siguiente:

- Realización de una simulación funcional.

Debe crear un test_bench y realizar una simulación funcional comprobando el correcto funcionamiento del Trivium (realización de la carga, operación y generación de la señal especial).

- Introducir un MMCM para generar el reloj.

Se pide crear un nuevo diseño que tendrá el mismo comportamiento y las mismas entradas y salidas que el circuito con la máquina de estados pero en el que se coloque un bloque de generación de señales de reloj MMCM (*Mixed-Mode Clock Manager*). Por lo tanto en este diseño se copia el código del diseño de la máquina de estados.

El MMCM se inserta creando una nueva fuente del tipo IP, y configurándolo de forma que tenga un reloj de entrada de 100 MHz y un reloj de salida de 200 MHz (*clkfx*). Una vez generado, se incorpora como un componente al circuito con la máquina de estados, haciendo que tanto el reloj del Trivium como el de la máquina de estados sea el reloj de salida del bloque MMCM (*clkfx)*.

- Crear un test_bench y simular funcionalmente.

De nuevo se crea un test_bench y se realiza la simulación. Un punto interesante es la comprobación de que el reloj clkfx tiene una frecuencia de 200 MHz.

- Realizar una simulación post-route.

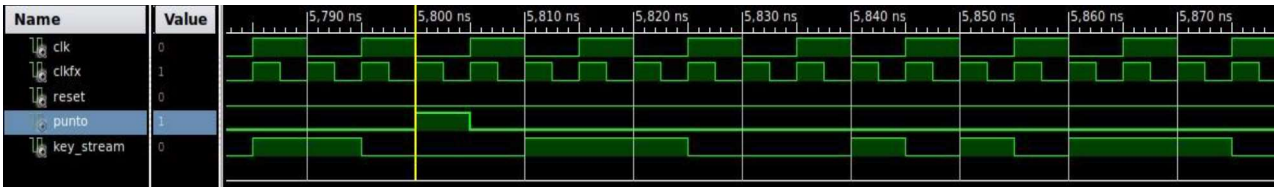
Para poder analizar la máxima frecuencia de funcionamiento desde el punto de vista de simulación, se realiza una implementación del diseño y se realiza una simulación post-route. Pero ahora la frecuencia del reloj no se cambia en el test_bench, sino en el módulo MMCM, de forma que manteniendo la frecuencia del reloj de entrada en 100 MHz, se vaya cambiando la frecuencia de la señal de reloj *clkfx*. Con estas simulaciones se vuelve a obtener un dato de frecuencia máxima de operación del circuito.

- Comprobación de máxima frecuencia externa.

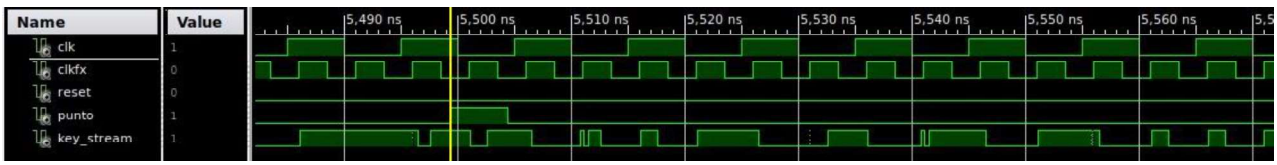
Finalmente se les propone un último análisis. Este análisis es experimental y consiste en visualizar señales de salida de la FPGA en el osciloscopio. Para ello se realiza la implementación del circuito en la placa Nexys-4 DDR y se conecta la salida *key_stream* del cifrador a un osciloscopio. Cambiando la frecuencia del reloj *clkfx* y programando la placa se puede observar la forma de onda y si es visible o no.

Los resultados de esta práctica son varios: por una parte se ha aprendido el manejo de los MMCM para genera internamente señales de reloj de frecuencias tanto inferiores como superiores a la frecuencia del reloj de entrada. Se ha vuelto a medir con simulaciones post-route la máxima frecuencia de funcionamiento y además se ha visto experimentalmente, cómo se degradan las salidas digitales cuando se aumenta la frecuencia de funcionamiento.

En la Fig. 2 se muestran capturas de simulaciones funcional y post-route a para la frecuencia de 200 MHz. Estos son los resultados que se pretenden obtener en esta práctica. Puede



(a)



(b)

Fig. 2. Capturas del funcionamiento (a) funcional, (b) Post-route a 200MHz.

.comprobarse cómo la salida a ambas frecuencias es diferente, lo que implica un mal funcionamiento del cifrador a 200 MHz.

C. *Tercera práctica: Verificación experimental de la máxima frecuencia de operación*

Este laboratorio tiene como objetivo analizar las señales internas del circuito programado en la placa para comprobar experimentalmente la máxima frecuencia de operación del cifrador Trivium. Para ello se va a utilizar la herramienta de Xilinx Chipscope.

Chipscope es un analizador lógico empotrado. Permite la inclusión de puntos de prueba dentro de la propia FPGA para capturar los valores de señales muestreadas con una señal de muestreo. Los datos muestreados se almacenan en un bloque de memoria del dispositivo FPGA. Posteriormente son trasladados al ordenador y visualizados en pantalla.

Este laboratorio parte del último diseño realizado en la práctica anterior. Este diseño incluye un módulo MMCM para cambiar la frecuencia de la señal del reloj.

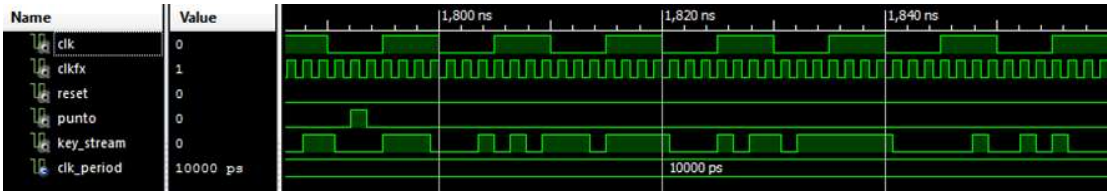
Sobre ese diseño hay que añadir un módulo de Chipscope. Para ello se añade una nueva fuente del tipo "ChipScope Definition and Connection File". La configuración de este módulo consiste básicamente en seleccionar las señales a muestrear, la señal de muestreo y el número de muestras que ve van a almacenar. Toda esta configuración se hace desde un entorno gráfico.

Una vez incluido el módulo Chipscope, los pasos a seguir son:

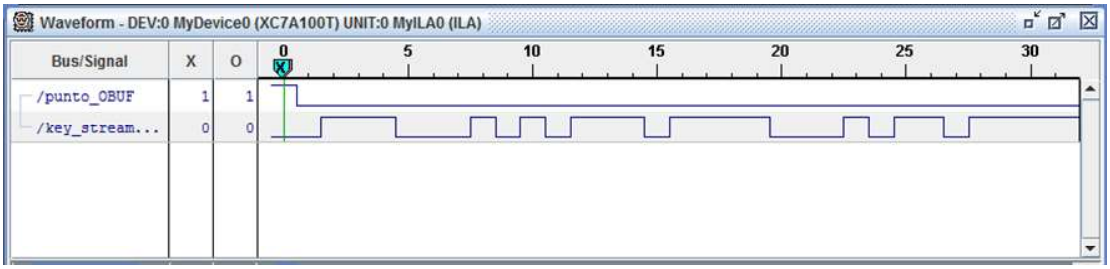
- Programación y verificación con Chipscope

La programación se hace dentro de la misma herramienta de Xilinx ISE usando la opción "Analyze Design Using Chipscope". Este programa abre una nueva interfaz gráfica en la que se pueden configurar las señales a visualizar y el modo de disparo.

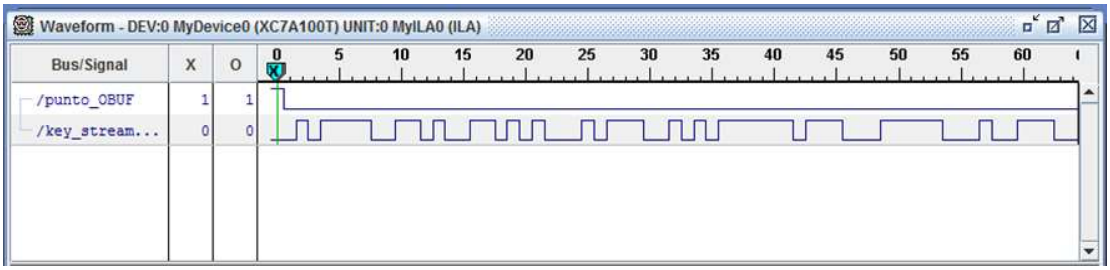
Se configura el disparo para que se produzca cuando la salida punto se ponga a '1'. La verificación del funcionamiento se hace comparando los resultados obtenidos en simulación funcional con los obtenidos mediante Chipscope. Para analizar el comportamiento a distintas frecuencias, en el módulo MCMM se cambia la frecuencia del reloj de salida y se programa la FPGA. En Chipscope se analiza la salida a partir de la puesta a '1' de la señal punto. Por la comparación con las salidas observadas en la simulación funcional puede saberse si



(a)



(b)



(c)

Fig. 3. Capturas del funcionamiento (a) funcional, (b) funcionamiento correcto observado con Chipscope a 500 MHz, (c) funcionamiento incorrecto observado con Chipscope a 600 MHz.

el funcionamiento es correcto o no.

En la Fig. 3 se muestran los resultados de la salida de la simulación funcional y las salidas capturadas por Chipscope a 500 MHz y a 600 MHz. Puede observarse que a 500 MHz la salida es igual a la generada por simulación funcional, mientras que a 600 MHz la salida es totalmente diferente. Con esto se concluye que la frecuencia máxima de operación está comprendida entre los 500 y los 600 MHz.

Un punto importante es la comparación con los resultados obtenidos experimentalmente con los obtenidos mediante simulación. Experimentalmente se obtienen unas frecuencias máximas de funcionamiento mucho mayores que las obtenidas mediante simulación post-route.

IV. Conclusiones

En este artículo se ha propuesto un conjunto de tres prácticas que utilizan diseños VHDL implementados en FPGA para enseñar a los alumnos aspectos relacionados con la generación de relojes en una FPGA y con el cálculo experimental de la frecuencia máxima de operación.

El circuito escogido tiene la suficiente sencillez para que funcione a una frecuencia mayor que la del reloj de entrada, por lo que se utiliza un bloque de generación interno de señales de reloj para generar una frecuencia de reloj más alta que la frecuencia de la señal de entrada.

Los resultados son muy satisfactorios e impactantes para los alumnos, pues comprueban que de forma externa las señales que sólo pueden llegar a una frecuencia muy baja en comparación con la frecuencia máxima a la que funcionan internamente.

Agradecimientos

Este trabajo ha sido parcialmente soportado por los proyectos CESAR (TEC2013-45523-R), INTERVALO (TEC2016-80549-R) y LACRE (CSIC 201550E039).

Referencias

[1] Puede consultarse el proyecto docente en el enlace: http://www.us.es/estudios/grados/plan_201/asignatura_2010034 (accedido en febrero 2018).

[2] Puede consultarse el proyecto docente en el enlace: http://www.us.es/estudios/grados/plan_201/asignatura_2010011 (accedido en febrero 2018).

[3] Puede consultarse el proyecto docente en el enlace: http://www.us.es/estudios/grados/plan_201/asignatura_2010018 (accedido en febrero 2018).

[4] Julio Pastor Mendoza, José Manuel Villadangos Carrizo, Francisco Javier Rodríguez Sánchez, "Experiencias en el aprendizaje basado en proyectos del diseño de sistemas empotrados", TAEE 2016, Sevilla 22-24 de junio.

[5] Alfredo Rosado Muñoz, Manuel Bataller Mompeán y Juan Fco. Guerrero Martínez, "Aprendizaje por Proyectos: Una Aproximación Docente al Diseño Digital Basado en VHDL", Revista Iberoamericana de Tecnologías del Aprendizaje (IEEE-RITA), Vol. 3, Nº 2, Noviembre 2008, pp. 87-95.

[6] C. De Cannière, "Trivium: A stream cipher construction inspired by block cipher design principles," in Proc. Int. Conf. Inf. Secur., 2006, pp. 171–186, doi: 10.1007/11836810_13.

[7] eSTREAM: The ECRYPT Stream Cipher Project. Accessed: Jul. 2017.[Online]. Available: http://www.ecrypt.eu.org/stream/ (accedido en febrero 2018).