



# Prototipado rápido de sistemas de procesamiento de vídeo basados en el VFBC de Xilinx

L. Garcés<sup>1</sup>, S. Sánchez<sup>2</sup>, P. Brox<sup>3</sup>, A. Cabrera<sup>4</sup>

<sup>1</sup> Instituto Superior Politécnico “José Antonio Echeverría”, Cuba, [imgarcess@electronica.cujae.edu.cu](mailto:imgarcess@electronica.cujae.edu.cu)

<sup>2</sup> Instituto de Microelectrónica de Sevilla, España, [santiago@imse-cnm.csic.es](mailto:santiago@imse-cnm.csic.es)

<sup>3</sup> Instituto de Microelectrónica de Sevilla, España, [brox@imse-cnm.csic.es](mailto:brox@imse-cnm.csic.es)

<sup>4</sup> Instituto Superior Politécnico “José Antonio Echeverría”, Cuba, [alex@electronica.cujae.edu.cu](mailto:alex@electronica.cujae.edu.cu)

## RESUMEN / ABSTRACT

El presente trabajo desarrolla módulos hardware para el prototipado rápido de sistemas de procesamiento de vídeo basados en el controlador de memoria para fotogramas de vídeo (VFBC) de Xilinx. Esta implementación permite el almacenamiento de los fotogramas en memoria externa al dispositivo programable, así como su correcto manejo para el diseño de sistemas de procesamiento espacio-temporales utilizando el flujo de diseño basado en modelos de Xilinx System Generator. Los módulos hardware son los encargados de la configuración y control de las interfaces de escritura y lectura del VFBC, además de la manipulación de las señales de sincronismo de vídeo para la interconexión de periféricos de entrada y salida.

El artículo incluye además la descripción de los módulos elaborados así como el análisis de los resultados del empleo de los mismos en el desarrollo de un demostrador de procesamiento temporal de vídeo utilizando un detector de movimiento simple sobre una placa *Spartan-6 SP605 Evaluation Platform*.

Palabras claves: Procesado de vídeo, VFBC, MPMC, Xilinx, FPGA

### *Control modules for video processing systems based on the Xilinx VFBC*

*This paper develops hardware modules for rapid prototyping of video processing systems based on the Xilinx video frame buffer controller (VFBC). This implementation allows the storage of video frames in memory external to the programmable device, as well as its proper handle for designing spatio-temporal processing systems using the Xilinx System Generator model-based design flow. The hardware modules are responsible for the configuration and control of writing and reading VFBC interfaces, as well as the manipulation of video synchronization signals for interconnecting input and output peripherals.*

*The article also include the description of the elaborated modules and the analysis of the results of its use for the development of a temporal video processing demonstrator using a simple motion detector on a Spartan-6 SP605 Evaluation Platform board.*

Key words: Video processing, VFBC, MPMC, Xilinx, FPGA

## INTRODUCCIÓN

Los algoritmos de procesamiento de vídeo han estado condicionados, desde sus inicios, por el empleo de técnicas de ejecución secuencial, sustentadas en procesadores de propósito general y microcomputadoras personales. El avance de las tecnologías de fabricación de circuitos integrados ha permitido el desarrollo de dispositivos lógicos programables de elevadas prestaciones, entre ellos los arreglos de compuertas programables (*FPGA*), que proporcionan una solución alternativa a la implementación eficiente de este tipo de algoritmos. El aumento significativo del tiempo de cálculo en las nuevas técnicas de procesamiento, puede ser compensado asignando tareas de cálculo intensivas al hardware y explotando el paralelismo de los algoritmos <sup>1</sup>.

Muchas operaciones para procesamiento de vídeo están basadas en la modificación de la imagen de acuerdo con los píxeles vecinos en el mismo fotograma (procesado espacial) así como en los fotogramas anteriores (procesado temporal) <sup>2,3</sup>. Este último requisito ha impulsado el desarrollo de diversas interfaces para almacenar y manejar las tramas en memoria, requiriendo de un adecuado diseño para minimizar el consumo de recursos y explotar el paralelismo inherente en los algoritmos de procesamiento <sup>1,2</sup>.

La limitación de recursos de los dispositivos programables en relación con el tamaño que pueden alcanzar los fotogramas, impiden el uso de la memoria interna de los mismos, siendo necesario el uso de otras estructuras de almacenamiento. La memoria externa disponible en muchas placas de desarrollo de *FPGA* es, usualmente, la solución para mantener varios fotogramas de vídeo en el flujo de procesamiento, aunque su uso implique la inclusión de módulos hardware que permitan un acceso correcto a la información <sup>4</sup>.

Los sistemas de procesamiento de vídeo basados en el controlador de fotogramas de vídeo (*VFBC*) utilizan las interfaces de conexión del controlador multipuerto de memoria externa (*MPMC*) de los *FPGA* de Xilinx como un módulo para el manejo de los fotogramas en la memoria <sup>4-6</sup>. El *VFBC* permite la interpretación de la memoria lineal del sistema como un espacio bidimensional para almacenar las tramas de vídeo a procesar, en la que los intercambios de la información no necesitan de la interacción del procesador.

La mayor parte de los diseños basados en el *VFBC* se desarrollan para lograr el sincronismo de las interfaces de entrada y salida conectadas a la placa, y no permiten el procesamiento espacio-temporal de las tramas. Un ejemplo de aplicación del *VFBC* como interfaz de sincronismo se muestra en el kit de desarrollo de vídeo Spartan-3A DSP VSK de Xilinx <sup>7</sup>, utilizando un flujo de vídeo real. Cada demostrador emplea módulos de propiedad intelectual (*IP*) desarrollados en lenguaje de descripción de hardware los cuales interconectan los subsistemas de entrada y de salida con el *MPMC*. La implementación de estos sistemas de procesamiento en placas de desarrollo no dedicadas a aplicaciones de vídeo y, por ende, menos costosas, involucra el cambio de los módulos *IP* de las interfaces de entrada y salida de los demostradores, no siendo estas interfaces genéricas y fácilmente adaptables a cualquier dispositivo.

El sistema de procesamiento descrito en <sup>4</sup> utiliza la metodología de diseño basada en modelos de Matlab - Simulink - Xilinx System Generator (*XSG*) para el desarrollo de las interfaces de control para el *VFBC*. En esta nota de aplicación se realiza una demostración paso a paso de los elementos necesarios para el uso del *VFBC* y su comprobación en hardware utilizando la co-simulación de *XSG*. El sistema está pensado para obtener una imagen desde el espacio de trabajo de Matlab y enviarla al flujo de datos de System Generator para su procesamiento, no teniendo en cuenta las señales de sincronismo presentes en una trama de vídeo real, así como otras señales necesarias para el correcto funcionamiento del *VFBC*.

El presente trabajo expone el desarrollo de un sistema básico de procesamiento de vídeo utilizando el controlador multipuerto de memoria configurado como *VFBC* sobre una placa de desarrollo *Spartan-6 SP605 Evaluation Platform* de Xilinx. El diseño utiliza, además, interfaces genéricas para la implementación, empleando el flujo de diseño basado en modelos de Xilinx System Generator, de sistemas de procesamiento de vídeo sobre placas de desarrollo que admitan el uso del *MPMC* para el acceso a memoria externa. Además se implementa un detector de movimiento simple como demostrador de la aplicación del sistema para el procesamiento temporal de los fotogramas.

## SISTEMA DE PROCESADO DE VÍDEO SOBRE FPGA

Un sistema de procesamiento de vídeo permite la adquisición, almacenamiento, procesamiento y visualización de la señal de vídeo. La necesidad de disponer de sistemas que operen en tiempo real ejecutando transformaciones complejas sobre las señales de vídeo, así como su implementación como sistemas portables con recursos limitados, supone un reto para el diseño microelectrónico<sup>8</sup>.

Un sistema de procesamiento de vídeo implementado sobre dispositivos reconfigurables se compone de un subsistema de entrada o captura de la información, un subsistema de almacenamiento, un subsistema de procesamiento y un subsistema de salida o visualización, como se muestra en la Figura 1.

El flujo de información (*Data Flow* en la Figura 1) transita desde el subsistema de entrada, compuesto generalmente por una o varias señales de vídeo, hasta el subsistema de almacenamiento. Este mecanismo permite realizar un acoplamiento entre la razón de entrada de los datos del vídeo y la frecuencia de procesamiento del sistema.

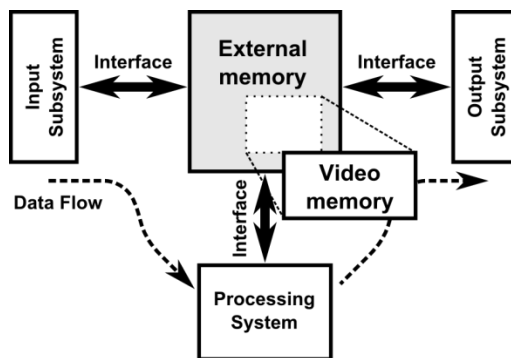


Figura 1: Sistema de procesamiento de vídeo.

El subsistema de almacenamiento incorpora interfaces de control, tanto de señales de entrada y salida, como de direcciones de memoria, en conjunto con un espacio de memoria para almacenar los fotogramas, conocido como memoria de vídeo.

En las placas de desarrollo de Xilinx la memoria externa es manipulada mediante el controlador multipuerto de memoria, MPMC, el cual se encarga del protocolo de comunicación con la misma. Este módulo soporta, además, transferencias de acceso directo a memoria (*DMA*), acelerando el proceso de intercambio de información entre el sistema de procesamiento y la memoria, sin necesidad de la interacción del procesador<sup>6</sup>.

El subsistema de procesamiento obtiene los fotogramas desde la memoria de vídeo y realiza los algoritmos pertinentes, entregando el resultado al subsistema de almacenamiento nuevamente o al subsistema de salida para su visualización o comprobación matemática.

El subsistema de salida asegura que la razón de transferencia de datos sea la requerida por el dispositivo de salida, evitando retrasos en las señales y haciendo que el efecto del refrescamiento del visualizador no sea perceptible.

## SISTEMA DE PROCESADO DE VÍDEO BASADO EN VFBC

El controlador de fotogramas de vídeo (Figura 2) se utiliza como interfaz del subsistema de almacenamiento para el intercambio de información con el MPMC conectado mediante la interfaz de conexión nativa de Xilinx (*NPI Formatter*). El VFBC presenta dos comportamientos posibles, uno como interfaz de entrada y otro como interfaz de salida. Ambos incluyen un módulo de programación compuesto por una cola de comandos para la configuración del dispositivo (*Command FIFO Interface*) y de módulos de control para las interfaces (*Command Fetch* y *Burst Controller*).

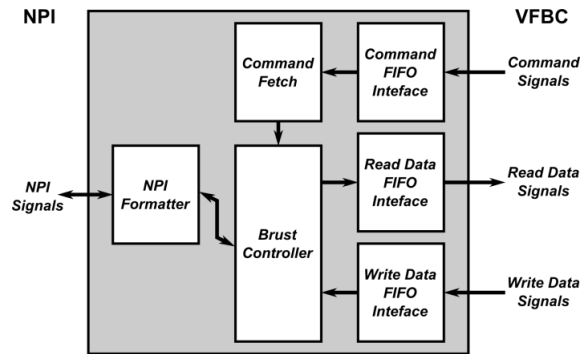


Figura 2: Diagrama de bloques de alto nivel del VFBC <sup>6</sup>.

Las interfaces de escritura (*Write Data FIFO Interface*) y de lectura (*Read Data FIFO Interface*) del VFBC son utilizadas según la configuración previa del dispositivo. Ambas se componen de colas de datos para acelerar el proceso de intercambio de la información con la memoria externa del sistema.

Estas interfaces son conectadas a módulos hardware para la configuración del subsistema de almacenamiento (Figura 3), los cuales pueden ser totalmente autónomos o presentar registros accesibles por el procesador. El módulo de entrada (*VIDEO to VFBC<sub>IN</sub>*) se encarga del control del flujo de datos de entrada, así como del manejo de las señales de sincronismo y la configuración de la interfaz de escritura de datos en memoria. El módulo de lectura (*VFBC<sub>OUT</sub> to Video*) es utilizado para el control del flujo de datos de salida desde la memoria externa. También es empleado para el control del sistema de visualización y la generación de las señales de sincronismo del mismo.

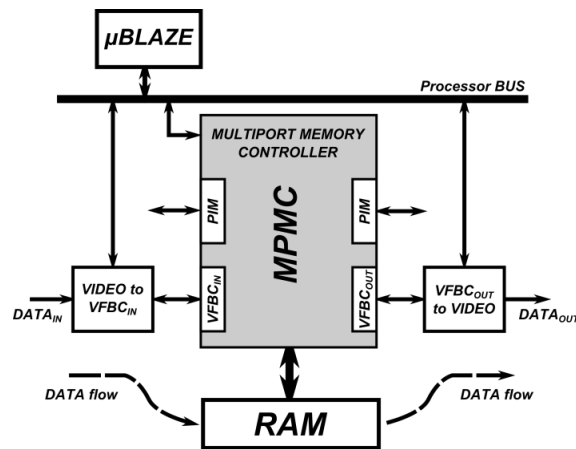


Figura 3: Sistema de procesamiento de vídeo basado en VFBC.

La configuración inicial del sistema requiere de las especificaciones de las tramas de vídeo a recibir. Entre estos parámetros se encuentran: la cantidad de columnas y de filas y el ancho máximo del fotograma, así como la dirección de memoria donde se inicia el almacenamiento. Estos parámetros definen el área bidimensional de la transferencia en memoria (Figura 4).

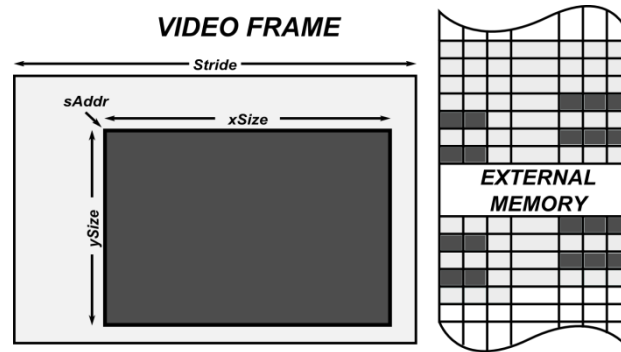


Figura 4: Parámetros de configuración del VFBC.

La cantidad de columnas ( $xSize$  en la Figura 4) representa el número de píxeles en el eje horizontal del fotograma multiplicado por el total de bytes necesarios para representar un píxel, mientras que la cantidad de filas ( $ySize$ ) es el número de píxeles en el eje vertical. La dirección de la localización inicial donde se almacena el fotograma ( $sAddr$ ) se refiere a la dirección que ocupa la celda de memoria externa donde se comienza a almacenar el fotograma de acuerdo con el mapa de memoria del sistema. El ancho total de la imagen ( $Stride$ ) corresponde a la cantidad de columnas máximas que ocupa una trama de video. Este parámetro es muy útil cuando solo se quiere extraer una porción del fotograma ya almacenado. Los parámetros  $xSize$ ,  $Stride$  y  $sAddr$  deben estar alineados en múltiplos de 128 bytes<sup>5,6</sup>.

## INTERFAZ DE CONTROL PARA EL VFBC UTILIZANDO SYSTEM GENERATOR

El control y configuración de las interfaces del VFBC conectadas al controlador de memoria se logra mediante dos módulos para el manejo de las tramas de vídeo. La entrada de un píxel de la trama de vídeo ( $data_m[31:0]$  en la Figura 5) se obtiene en cada ciclo de reloj del periférico de entrada ( $pClk$ ), manejado por los cambios en las señales de sincronismo ( $hSync$  y  $vSync$ ) que indican el inicio de una nueva línea y un nuevo fotograma, respectivamente. A estas señales se les añaden las entradas de control de los módulos: habilitación ( $en$ ), reinicio ( $rst$ ) y reloj ( $clk$ ); que rigen el comportamiento general de los bloques hardware y su sincronización con el resto del sistema.

## MÓDULO DE CONTROL PARA EL VFBC DE ESCRITURA

El módulo de entrada (Figura 5) se compone de cinco bloques principales: el bloque de parámetros de configuración de la trama (**Shared Reg**), el bloque de determinación de la trama activa en memoria (**Active Frame**), el bloque de obtención de los comandos de configuración (**CMD Calc**), el bloque de configuración de la interfaz de entrada (**CMD Conf**) y el bloque de control de escritura de datos (**Data Path**).

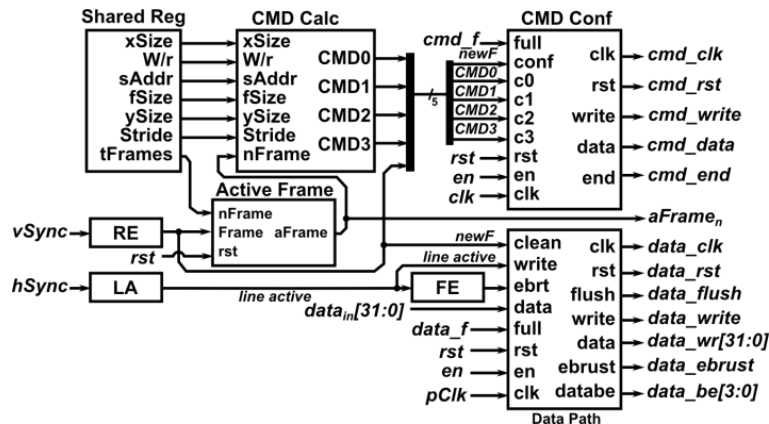


Figura 5: Módulo de configuración del VFBC de entrada.

El bloque de inicialización (*Shared Reg* en la Figura 5) proporciona una conexión directa con el procesador para la inicialización de los parámetros de la transferencia a realizar con la interfaz dada. Estos parámetros incluyen la configuración de la interfaz como de escritura o lectura en memoria (*W/r*), el tamaño total del fotograma (*fSize*) y el total de tramas a almacenar en la memoria (*tFrames*).

El bloque de trama activa (*Active Frame*) es el responsable del desplazamiento de la dirección de inicio del fotograma en memoria. Además provee la referencia de ubicación a los restantes VFBC.

Los parámetros iniciales de configuración son recibidos por el bloque de cálculo de comandos (*CMD Calc*) para realizar el cómputo de las cuatro palabras de configuración del VFBC. Estas palabras son enviadas al bloque de configuración (*CMD Conf*) el cual genera el protocolo de comunicación para la correcta programación del sistema de entrada. Los comandos de configuración son enviados mediante la salida *data* con la activación de la señal *cmd\_write*. Las señales *cmd\_end*, *cmd\_rst* y *cmd\_clk* realizan el control de la interfaz, mientras que la señal *cmd\_f* es devuelta por la interfaz para detener el proceso de envío de comandos. La detección de un frente de subida en la señal *vSync* indica el fin de un fotograma. Esta información es direccionada al bloque de configuración (entrada *conf*) para realizar la reconfiguración de la interfaz, la cual estará lista para escribir en una nueva dirección de memoria.

El bloque de manejo de datos (*Data Path*) controla el flujo de datos de entrada a la interfaz del VFBC. El valor de los píxeles es enviado a través de la señal *data\_wr[31:0]*, a razón de un píxel por pulso de la señal *data\_clk* y controlada por la salida *write* de este bloque, activa mientras se detecte un píxel válido en el fotograma. Las restantes señales (*data\_rst*, *data\_flush*, *data\_ebrust*, *data\_be[3:0]*) realizan el control de la interfaz de entrada del VFBC.

## MÓDULO DE CONTROL PARA EL VFBC DE LECTURA

El módulo de control de la interfaz de lectura agrega modificaciones a los bloques descritos anteriormente. El bloque de parámetros iniciales (*Shared Reg* en la Figura 6) comparte los registros de configuración de la interfaz de escritura, e incorpora el estándar a utilizar en la transferencia de salida (*vStd*), necesario para generar las señales de sincronismo para el próximo elemento de la cadena de procesado. El bloque de trama activa se sustituye por el bloque de referencia de trama (*Ref Frame*). Este recibe la referencia de la posición del VFBC anterior y proporciona al bloque de cálculo de comandos (*CMD Calc*) el desplazamiento en la dirección de inicio del fotograma a leer.

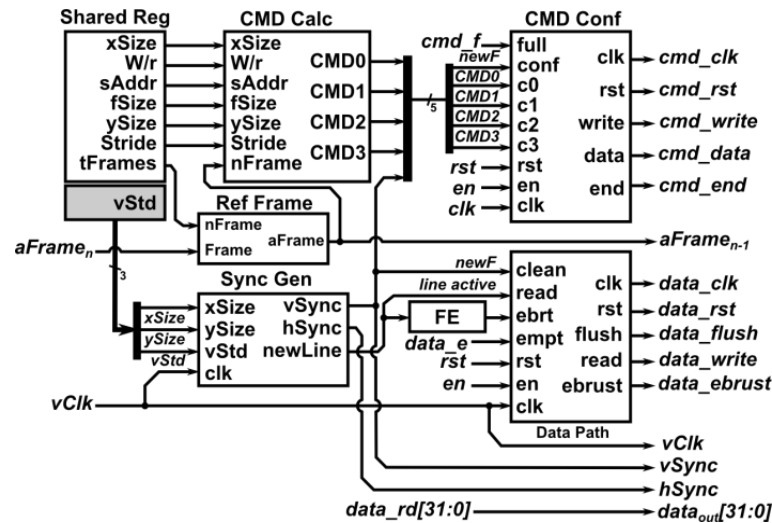


Figura 6: Módulo de configuración del VFBC de salida.

Un bloque nuevo es necesario para la generación de las señales de sincronismo. El bloque de sincronismo (*Sync Gen* en la Figura 6) recibe los parámetros del bloque de inicialización y devuelve las señales de sincronismo (*vSync*, *hSync*) y de habilitación de lectura (*line active*), para el correcto funcionamiento de los restantes bloques del sistema y la posibilidad de interconectar otros periféricos a la salida del VFBC.

El bloque de manejo de datos (*Data Path* en la Figura 6) activa la salida del VFBC (*data\_rd[31:0]*) en cada pulso de la señal *vClk*, mientras la señal *line active* se encuentre en nivel alto.

## RESULTADOS DE IMPLEMENTACIÓN

Usando un flujo de diseño basado en modelos, los módulos anteriores permiten el prototipado rápido de sistemas de procesamiento de vídeo que requieran acceder simultáneamente a información procedente de distintos fotogramas. Los resultados de implementación se obtienen para la placa de desarrollo de Xilinx *Spartan-6 SP605 Evaluation Platform*. La verificación funcional se lleva a cabo realizando co-simulación hardware del diseño desde el entorno de trabajo de Simulink.

El sistema utilizado como demostrador (Figura 7) es un detector de movimiento, compuesto por un módulo de escritura, que recibe los fotogramas de vídeo desde el espacio de trabajo de Matlab, y dos módulos de salida, que devuelven las dos últimos fotogramas almacenados en la memoria externa de la placa de desarrollo. Además, el sistema utiliza un procesador (*µBlaze* en la Figura 7) que incluye 32 KB de memoria interna, una interfaz de comunicación serie asíncrona (*UART*) y el controlador multipuerto de memoria externa (*MPMC*), configurado con tres interfaces para VFBC.

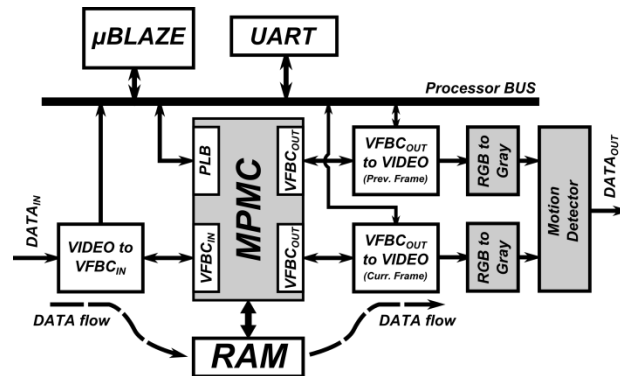


Figura 7: Aplicación para detección de movimiento en una secuencia de vídeo.

La entrada de un nuevo píxel ( $DATA_{IN}$  en la Figura 7), se rige por el reloj de System Generator, añadiendo las señales de sincronismo del mismo. El área de memoria externa donde se almacenan las tramas de vídeo se inicializa con el valor hexadecimal  $0xFFFFFFFF$  para comprobar la correcta transferencia del fotograma. La co-simulación del sistema se realiza para la escritura de cuatro tramas en memoria, habilitando la lectura simultánea de dos fotogramas, tomando como referencia la posición de la interfaz de escritura. Por medio de la transmisión serie se muestra el estado de la configuración de las interfaces y la indicación del funcionamiento de las mismas.

El sistema de procesado utiliza dos bloques de conversión de formato de  $RGB^1$  a escala de grises (**RGB to Gray** en la Figura 7), para luego realizar la estimación de movimiento mediante la diferencia de los píxeles en dos fotogramas de vídeo consecutivos (**Motion Detector**). Los resultados del procesado ( $DATA_{OUT}$ ) son enviados al espacio de trabajo para comprobar la efectividad de la implementación.

Los fotogramas de salida reflejan los cambios ocurridos entre dos fotogramas consecutivos del vídeo de entrada, como muestra la Figura 8. Las áreas oscuras equivalen a píxeles que no sufren cambios detectables por el ojo humano, mientras que las áreas claras corresponden a variaciones bruscas en las imágenes.



Figura 8: Resultado de la detección de movimiento entre dos tramas consecutivas.

La Tabla I muestra el consumo de recursos del sistema. Estos resultados incluyen el área ocupada por el procesador, teniendo en cuenta que la cantidad de recursos del mismo puede variar de acuerdo con los componentes que le sean añadidos. El diseño presenta un consumo de recursos de 7,853 slices<sup>2</sup>, representando un 14.39% del total disponible en el FPGA.

<sup>1</sup> Espacio de color Rojo, Verde y Azul de sus siglas en inglés.

<sup>2</sup> Menor unidad de recursos del FPGA compuesto por tablas de búsquedas y almacenadores.



Tabla I: Consumo de recursos del sistema.

	Spartan-6 SP605 EP	Slices	DSP48A	BRAM
Recursos	Total	54,576	58	116
	Utilizados	7,853	9	44
	Por ciento	14.39%	15.52%	37.93%

Además se utilizan 9 multiplicadores dedicados (*DSP48A*), de ellos, uno en cada bloque de cálculo de las palabras de comandos, para obtener el desplazamiento de la dirección de inicio del fotograma. El consumo de los bloques de memoria (*BRAM*) es de un 37.93%, siendo empleados en los registros compartidos con el procesador, en la memoria local del mismo, así como en el MPMC.

## CONCLUSIONES

El uso del VBFC de Xilinx en conjunto con bloques de configuración y control adecuados, posibilita el desarrollo de sistemas de procesado espacio-temporales donde se requieren más de un fotograma de vídeo para realizar el procesado.

El diseño de módulos para el manejo de las interfaces de controlador de fotogramas sobre Xilinx System Generator disminuye el tiempo de prototipado en la creación de sistemas de procesado de vídeo. El control sobre las señales de sincronismo es necesario para mantener la temporización y la simultaneidad de las distintas etapas en la cadena de procesado, por lo que los módulos de control deben manejar estas señales.

La co-simulación hardware, provista por Xilinx System Generator, permite la comprobación del diseño desarrollado, direccionando el flujo de datos desde el entorno de Matlab hasta la placa y su retorno al espacio de trabajo para la comprobación del sistema y la detección de errores en el mismo.

Los módulos de control pueden ser utilizados en el desarrollo de sistemas de procesado de vídeo sobre cualquier placa de desarrollo que permita el uso del controlador multipuerto de memoria para el manejo de la memoria externa de la misma.

## AGRADECIMIENTOS

Esta investigación ha sido financiada parcialmente por la Agencia Española de Cooperación Internacional para el Desarrollo (AECID) mediante los proyectos PCI D/024124/09, PCI D/030769/10 y PCI A1/039607/11 (<http://www.imse-cnm.csic.es/fortin>)

## REFERENCIAS

1. S. M. QASIM, S. A. ABBASI, AND B. A. ALMASHARY: "An overview of advanced FPGA architectures for optimized hardware realization of computation intensive algorithms," *2009 International Multimedia, Signal Processing and Communication Technologies*, pp. 300–303, 2009.
2. N. LAWAL, B. THORNBERG, AND M. O'NILS: "Address generation for fpga rams for efficient implementation of real-time video processing systems," *International Conference on Field Programmable Logic and Applications, 2005.*, pp. 136–141, 2005.
3. N. LAWAL AND M. O'NILS: "Embedded FPGA memory requirements for real-time video processing applications," *2005 Norchip*, pp. 206–209, 2005.
4. D. PHANTHAVONG AND J. OU: "Integrating a Video Frame Buffer Controller ( VFBC ) in System Generator," *Xilinx Application Notes*, vol. 1136, no. 1.0, pp. 1–32, 2009.
5. XILINX: "Video Frame Buffer Controller v1.0," *Data Sheet: Product Brief*, pp. 1–5, 2007.

6. **XILINX**: “LogiCORE IP Multi-Port Memory Controller v6.06.a,” *Data Sheet: Product Specification*, Xilinx Inc., pp. 1–218, 2012.
7. **XILINX**: “Spartan-3A DSP FPGA Video Starter Kit User Guide,” *Data Sheet: User Guide*, vol. 456, pp. 1–82, 2010.
8. **Z. WENGE AND H. HUIMING**: “FPGA-based video image processing system research,” in *Computer Science and Information Technology (ICCSIT)*, 2010 3rd IEEE International Conference on, 2010, vol. 4, pp. 680–682.

## AUTORES

**Luis Manuel Garcés Socarrás**, Ingeniero en Automática en el 2006, Máster en Sistemas Digitales en el 2011 por el Instituto Superior Politécnico “José Antonio Echeverría”, Profesor Asistente del Departamento de Automática y Computación del ISPJAE, Calle 114 N° 11901 e/ Ciclo vía y Rotonda, CUJAE, Marianao, La Habana, Cuba. Telef: +53 (7) 266 3329. Actualmente desarrolla su tema de investigación pre-doctoral sobre aceleración de algoritmos mediante hardware reconfigurable para el procesado de vídeo.

Dirección: Calle 48 N° 2505 e/ 25 y 27 San Antonio de los Baños. Artemisa. Cuba. CP 32500

Email: [imgarcess@electronica.cujae.edu.cu](mailto:imgarcess@electronica.cujae.edu.cu)

**Santiago Sánchez Solano**, Doctor en Física por la Universidad de Sevilla en 1990, Investigador Científico del CSIC adscrito al Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica (IMSE-CNM-CSIC), Sevilla, España.

Email: [santiago@imse-cnm.csic.es](mailto:santiago@imse-cnm.csic.es)

**Piedad Brox Jiménez**, Doctora en Física por la Universidad de Sevilla en 2009, Investigadora postdoctoral del Programa Nacional Juan de la Cierva adscrita al Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica (IMSE-CNM-CSIC), Sevilla, España.

Email: [brox@imse-cnm.csic.es](mailto:brox@imse-cnm.csic.es)

**Alejandro José Cabrera Sarmiento**, Ingeniero Electricista, Doctor en Ciencias Técnicas por el Instituto Superior Politécnico “José Antonio Echeverría”, Profesor Principal del Departamento de Automática y Computación del ISPJAE, Calle 114 N° 11901 e/ Ciclo vía y Rotonda, CUJAE, Marianao, La Habana, Cuba. Telef: +53 (7) 266 3301.

Email: [alex@electronica.cujae.edu.cu](mailto:alex@electronica.cujae.edu.cu)