

TUTORIAL PARA EL MODELADO DE UN BLOQUE ACELERADOR DE VIDEO UTILIZANDO SYSTEM GENERATOR

TUTORIAL FOR THE MODELING OF A BLOCK VIDEO ACCELARATOR USING SYSTEM GENERATOR

Juan J. Raygoza P.¹, Susana Ortega C.¹, Alí Piña R.¹, Julio C. Sosa S.², A. de la Mora¹
juan.raygoza@cucei.udg.mx / susana.ortega@cucei.udg.mx / ali.pina@red.cucei.udg.mx / jcsosa@ipn.mx /
drdivec@cucei.udg.mx

Recibido: 02 marzo, 2009 / Aceptado: 01 mayo, 2009 / Publicado: 31 diciembre, 2009

RESUMEN. En este artículo se presenta el desarrollo de un sistema de adquisición de video en tiempo real, utilizando las características y ventajas de hardware reconfigurable con la flexibilidad de ser prototipado por medio de herramientas aceleradoras de sistemas DSPs. En este caso se utilizó la herramienta Xilinx System Generator que funciona sobre la plataforma Simulink. La ventaja que ofrece Simulink es el extenso bloque gráfico y el conjunto en conjuntos con las librerías de Xilinx para la implementación del diseño, que permite exportar el sistema a un Pcore en Xilinx Platform Studio. System Generator es una herramienta de diseño de alto nivel para la creación de sistemas DSP en dispositivos reconfigurables FPGAs. Además de proveer una gran abstracción, puede utilizarse para la construcción de diseños hechos a medida implementados con eficiente desempeño y mínima área, en la que se configuran los bloques de video creados en System Generator para posteriormente descargar el diseño en la tarjeta de desarrollo Spartan-3A xc3sd3400a-4fg676. Las implementaciones más importantes en las que se puede utilizar este diseño acelerador de video son las de reconocimiento de patrones y procesamiento de video en tiempo real, con la ventaja de un obtener hardware específico para la aplicación final.

PALABRAS CLAVE: Pcore, FPGA, DSP, Simulink.

ABSTRACT. This article presents the development of a system of video acquisition in real time, using the characteristics and advantages of reconfigurable hardware which has the flexibility of being prototyped through the accelerating tools of the DSP's systems. On this case we used the Xilinx System Generator that works over the Simulink platform. The advantage that Simulink offers is the extensive graphic block of libraries together with the Xilinx libraries for the design implementation, being able to export the system to a Pcore in Xilinx Platform Studio. System Generator it's a high level design tool for the DSP systems creation in FPGA reconfigurable devices. It also provide a great abstraction, it can be used for the construction of custom made designs implemented with efficient performance and minimum area, where the blocks of video created in System Generator are created for later download the design in the Spartan-3A xc3sd3400a-4fg676 development card. The most important implementations, in which we can use this video accelerator design, are those of pattern recognition and video processing in real time, with the advantage to obtain specific hardware for the final application.

KEYWORDS: FPGA, Pcore, DSP, Simulink

Introducción

El desarrollo de aplicaciones de video es un área importante en el procesamiento digital de señales. El mercado de la tecnología multimedia (video e imágenes) se ha convertido en un aspecto de gran relevancia en la vida cotidiana, desde la televisión digital hasta las videoconferencias en teléfonos móviles. Este crecimiento está alimentado por la aparición de nuevos estándares de compresión, encriptación y filtrado que permiten una mayor calidad en la imagen. Por consiguiente se requiere mucho más potencia de

¹ Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI), Blvd. Marcelino García Barragán #1421, Guadalajara, 44430, Jalisco, México - www.cucei.udg.mx

² Escuela Superior de Cómputo (ESCOM) del Instituto Politécnico Nacional, Unidad Profesional Adolfo López Mateos

procesamiento que la de los actuales sistemas DSP, aumentado el riesgo de que éstos queden obsoletos. Por ello es importante encontrar soluciones capaces de manejar el desempeño de aplicaciones de video en tiempo real. [1]

Actualmente muchas de estas soluciones son implementadas en procesadores de propósito general. Sin embargo, existen aplicaciones en las que es obligado satisfacer estrictas restricciones de tiempo, haciendo necesario el empleo de dispositivos de hardware. En general, para desarrollar una aplicación basada en el procesamiento digital de señales a alta velocidad se pueden utilizar como soporte tecnológico tres tipos de soluciones: DSPs, FPGAs o ASICs. Estas soluciones tienen sus ventajas y desventajas.

El diseño basado en ASICs ofrece la solución óptima en cuanto a velocidad de procesado, área de silicio ocupada y bajo consumo [3] [4]. Lamentablemente, los tiempos de desarrollo son muy largos, el riesgo de diseño es muy grande, se requiere una gran tirada de producción para justificar los gastos de fabricación iniciales y sus estructuras carecen de flexibilidad para acomodarse a los continuos cambios relativos a protocolos, estándares o mejoras en la calidad de servicio.

Los microprocesadores de propósito general ofrecen mayor facilidad de diseño y flexibilidad. Sus carencias se deben a que el procesamiento secuencial de cada instrucción (direccionamiento, decodificación, ejecución y actualización de resultado) no permite procesar en paralelo ciertos algoritmos. Además, este procesamiento secuencial redundante en un mayor consumo debido al frecuente acceso a memorias externas.

Por último, los dispositivos reconfigurables FPGAs ofrecen una solución intermedia en cuanto a rendimiento, prestaciones y consumo, aportando flexibilidad de software. Permite diseñar aplicaciones de alto rendimiento con menos riesgo que el que supone el diseño ASIC o procesadores específicos. [2]

Por esta razón la herramienta elegida para presentar una solución posible ha sido el entorno MatLab-Xilinx System Generator, pues se trata de una herramienta que permite crear y verificar diseños de hardware mediante la elección de un conjunto de módulos Simulink que son enlazados de una forma óptima. Los módulos ampliados comprenden desde los componentes más básicos de cada familia de FPGAs soportada, hasta módulos más sofisticados que llevan a cabo tareas de procesamiento de más alto nivel.

Con este entorno, el diseño del sistema hardware-software se lleva a cabo en un nivel muy alto de especificación, accediendo desde éste a la síntesis de dicho sistema de forma automática y sencilla. Además de la posibilidad de una interfaz, para el diseño de microprocesadores *softcore* (Microblaze) capaces de administrar periféricos y dispositivos de E/S.

La estructura del desarrollo de este trabajo es la siguiente: en primer lugar, construir un bloque acelerador de video con los periféricos necesarios para la plataforma base que permita al procesador Microblaze leer y escribir la memoria compartida en el bloque acelerador de video. La siguiente etapa consiste en exportar el bloque generado por System Generator como un Pcore a un proyecto a Xilinx Platform Studio. En la siguiente sección agregamos y configuramos el bloque acelerador de video creado con System Generator. Por último configuramos la *hyperterminal* y descargamos el *bitstream* generado por Xilinx Platform Studio.

2 Plataforma Base.

La Plataforma Base se construye utilizando el Xilinx Platform Studio. Ésta debe incluir las librerías de los periféricos para la tarjeta de desarrollo Spartan-3A DSP 3400 [5] y la tarjeta de video. La **Figura 1** muestra un diagrama a bloques de la Plataforma Base antes de que se haya configurado para una aplicación de video.

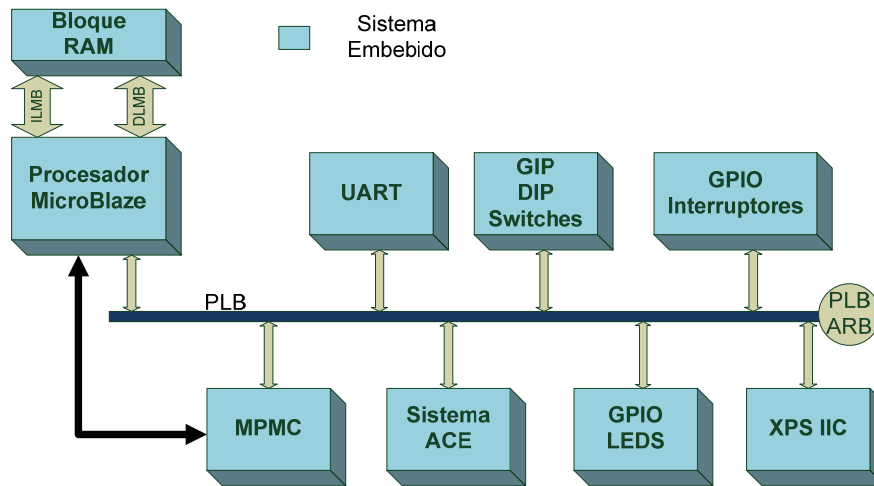


Figura 1. Diagrama de Bloques de la Plataforma Base [5]

El acelerador de video es creado a través del System Generator. Éste aprovecha las ventajas de rendimiento y flexibilidad de la FPGA que automáticamente se conecta a la Plataforma Base utilizando el Platform Studio. En la **Figura 2** se muestra el diagrama a bloques de la plataforma base con la aplicación de video.

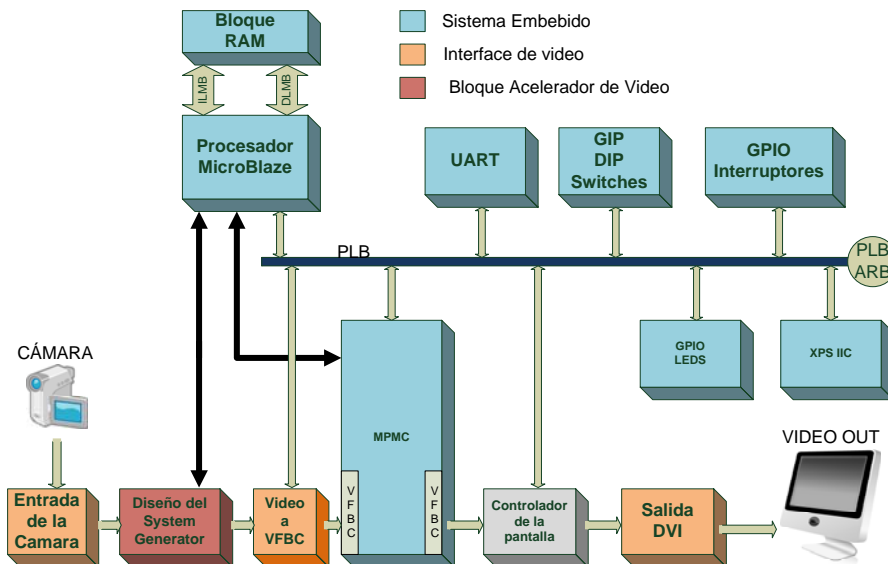


Figura 2. Cámara y video en paralelo con el procesador Microblaze y periféricos [5]

3 Bloque acelerador de video en System Generator.

El procesamiento de video en paralelo se diseñó para una interfaz de la cámara Micron CMOS [6] incluida en la tarjeta de desarrollo Spartan-3A DSP 3400. Esta cámara produce un intervalo serial de video junto con las señales de control de sincronización horizontal y vertical.

En la **Figura 3** se observa el nivel más alto del bloque acelerador de video compuesto por los módulos que procesan en tiempo real los datos provenientes de las señales de control de sincronización horizontal y vertical de la cámara. A continuación se describen los módulos que conforman el sistema:

1. System Generator utiliza los bloques amarillos llamados *gateway* para definir los límites entre el hardware del dispositivo reconfigurable FPGA y el sistema modelo de Simulink. El nivel más alto tiene tres entradas (*vs_in*, *hs_in* y *data_in*) y seis salidas (*vs_out*, *de_out*, *hs_out*, *red_out*, *green_out* and *blue_out*).
2. El bloque del procesador Microblaze añade y configura las interfaces necesarias de software y hardware para exportar el diseño.
3. System Generator genera el Pcore para exportar el diseño al Platform Studio.
4. Los subsistemas encargados del procesamiento de la señal provenientes de la cámara.

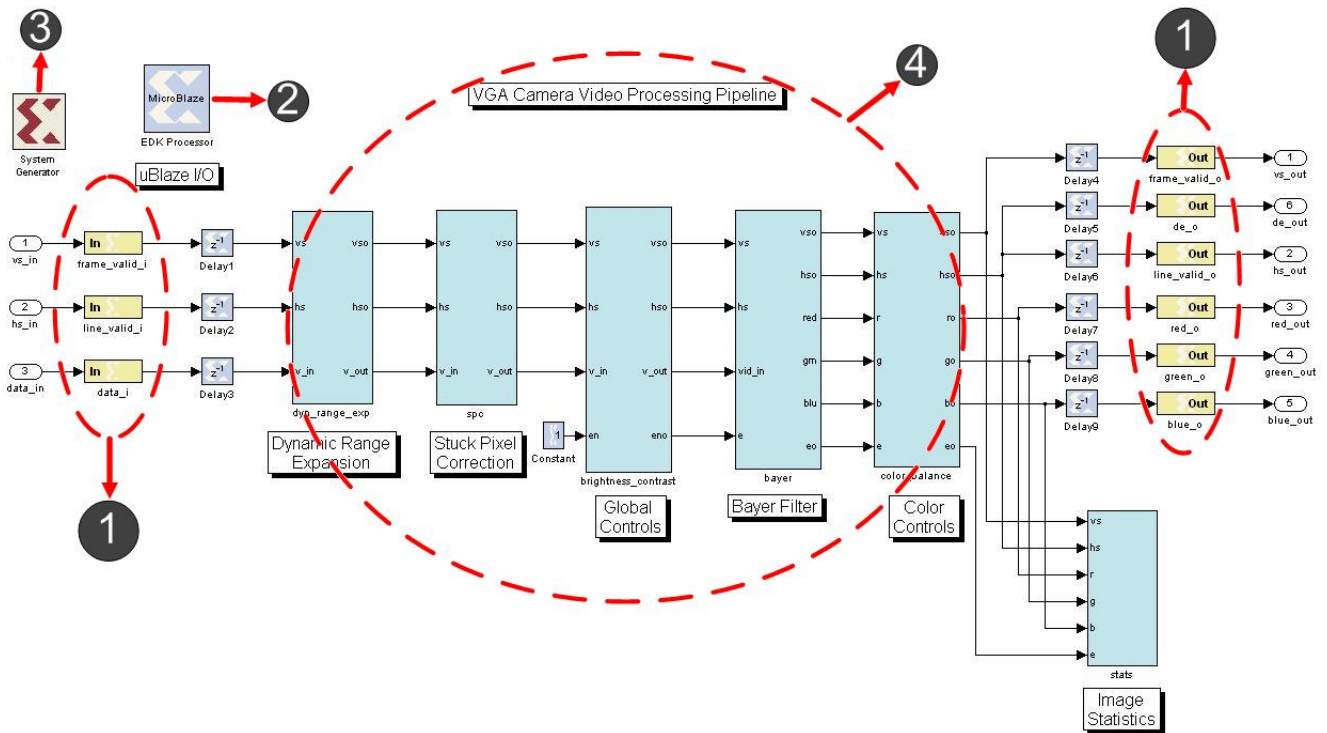


Figura 3. Bloque acelerador de video

Debido a que el bloque acelerador de video comparte registros físicos con el microprocesador de Microblaze es necesario utilizar un bloque de Xilinx capaz de comunicarse con ambos diseños. Por ello, el bloque utilizado es *From Register*, el cual dispone de los datos que son escritos en un bloque de registro correspondiente. Esto se muestra en la **Figura 4**.

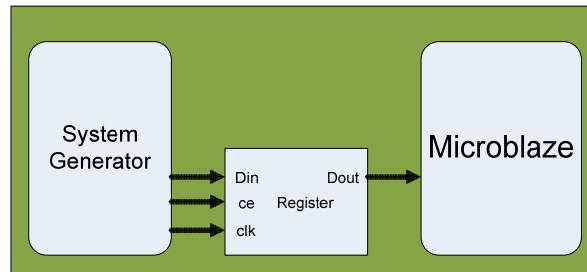


Figura 4. Bloque *From Register*

Los parámetros del bloque deben ser configurados con los siguientes datos:

- El nombre del registro debe ser diferente a todos los demás bloques de memoria que el sistema acelerador de video contenga.
- Se debe especificar un valor inicial de acuerdo con el ancho que el microprocesador de Microblaze contenga.
- El bloque debe inicializarse en el sistema al cual pertenezca; es decir, un bloque *From Register* es propiedad intelectual del sistema local donde haya sido implementado [7].

En la **Figura 5** se observa la ubicación donde el bloque *From Register* tendrá que conectarse.

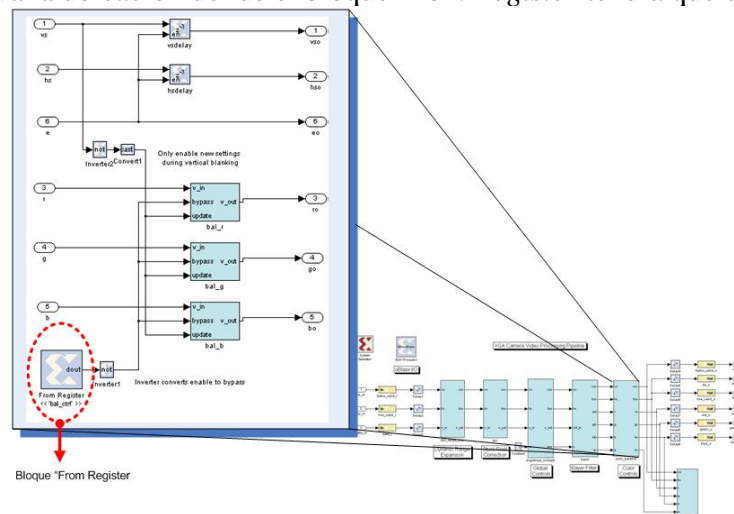


Figura 5. Ubicación del bloque *From Register*

Una de las tareas más importantes en el procesamiento de video e imágenes es la aplicación de diferentes filtros y algoritmos que facilitan una tarea específica; por ejemplo: algoritmo de umbralización, convolución espacial, filtro IIR y FIR, transformada FFT, entre otros. Para facilitar la aplicación de estos filtros y algoritmos, se optó por incorporar un convertidor *rgb2gray*, el cual toma la señal RGB de la cámara y la convierte en escala de grises. En la **Figura 6** se observa el convertidor *rgb2gray*, así como su ubicación en el sistema acelerador de video.

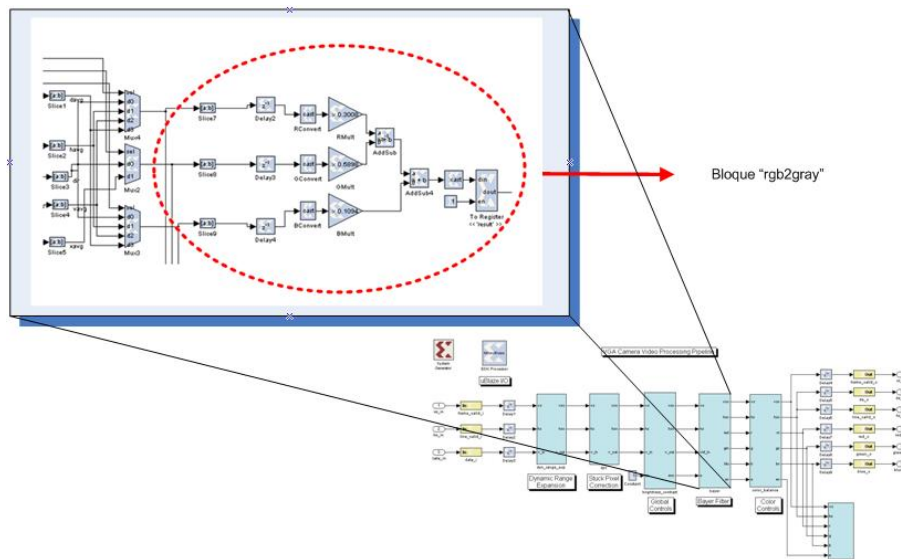


Figura 6. Bloque convertidor de RGB a escala de grises

Hasta este momento el bloque acelerador de video está terminado, así como los *buses* que funcionarán para conectar el diseño del System Generator al Microblaze. Sin embargo, aún no se ha generado el Pcore que se exportará al Platform Studio.

4. Exportando y configurando el Pcore de System Generator a Xilinx Platform Studio.

System Generator ofrece una útil función llamada *Pcore under Development*, la cual impide el almacenamiento en cache del código HDL producido para este Pcore. Esto es muy útil cuando se están generando pruebas, ya que estos Pcores generados son compilados directamente en el Xilinx Platform Studio, evitando la ralentización de la compilación final del archivo que se descargará en la tarjeta de desarrollo Spartan-3A DSP 3400. En la **Figura 7** se presenta el asistente generador de System Generator, el cual generará el Pcore para enseguida exportarlo al Platform Studio. Asimismo se observan las configuraciones del dispositivo, además de la ventana de versión a utilizar.



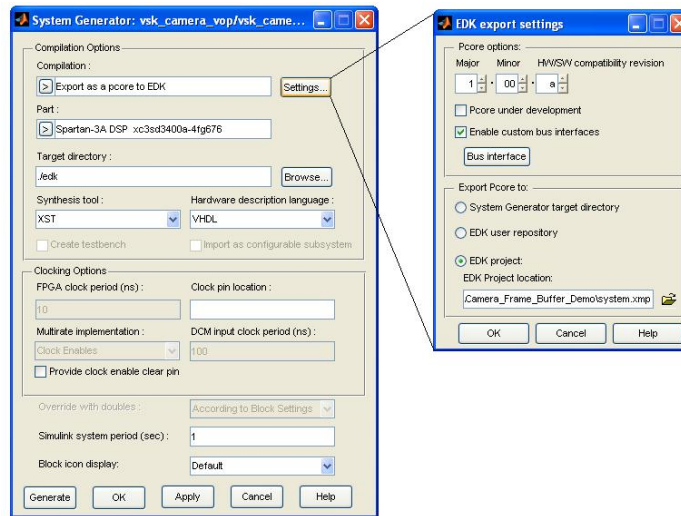


Figura 7. Asistente de generación del Pcore

En seguida se editará el sistema embebido del sistema de video utilizando el Platform Studio para configurar el bloque acelerador de video creado con System Generator. En la Figura 8 se observa el diagrama de bloques que conforma el sistema embebido completo.

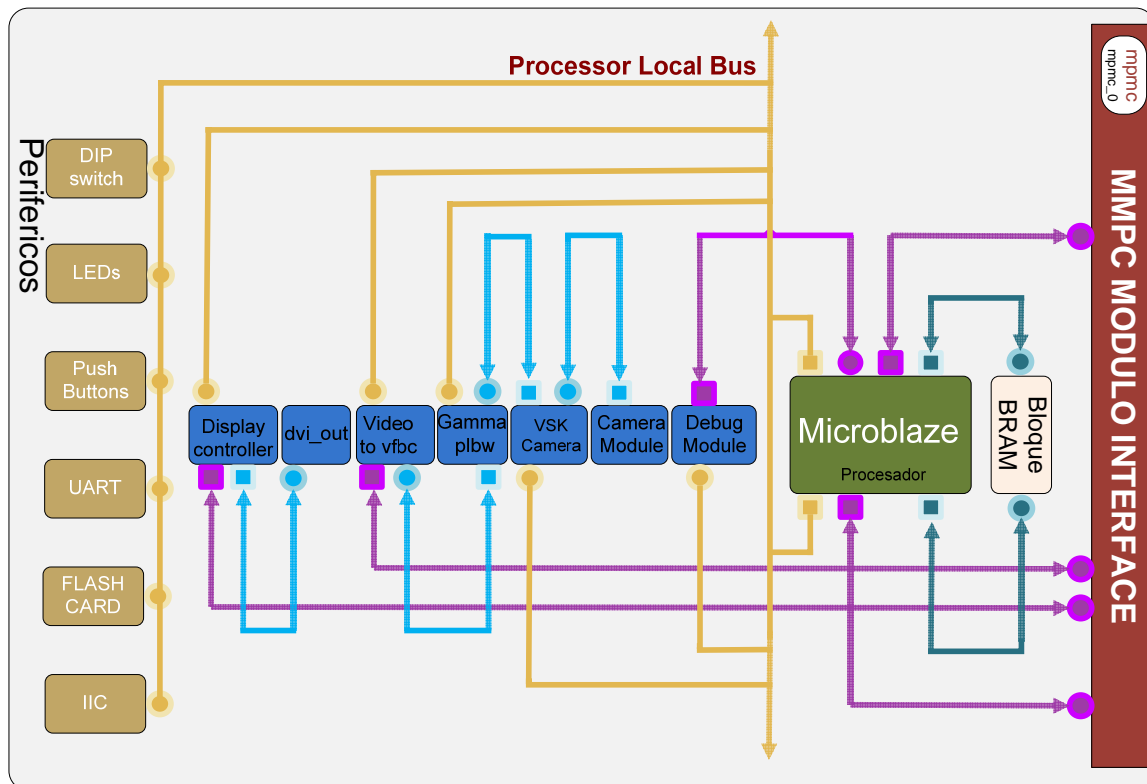


Figura 8. Diagrama a bloques del sistema embebido

5. Conclusiones.

En este artículo se presentó el desarrollo del modelado de un bloque acelerador de video utilizando herramientas aceleradoras de construcción de sistemas DSPs. Dichas herramientas permiten un entorno gráfico interactivo y personalizable para crear sistemas para el procesamiento de señales que, en conjunto con Platform Studio, constituyen un ambiente de gran variedad de herramientas para el desarrollo de sistemas embebidos, soportado en la mayoría de hardware reconfigurable. En definitiva la combinación de implementar un sistema DSP en un hardware reconfigurable proporciona un buen manejo de algoritmos de procesamiento de señales de gran complejidad.

Referencias

1. Tahernia, Omid. (2005) XtremeDSP Selection Guide.
2. Suardfáz Muro, J, Al-Hadithi B. (2006) Tutorial – Introducción a técnicas de procesamiento de imágenes con FPGAs.
3. Majumdar, A.K. (2000) Design of an ASIC for straight line detection in an image pp 128-133.
4. Rizkalla, M.E., Palaniswamy, K., Sinha, A.S.C., El-Sharkawy, M. 2000 ASIC memory design of 2-D median filters.
5. Spartan-3A DSP FPGA Video Starter Kit www.xilinx.com/products/spartan3a/3adsp
6. XtremeDSP Solution FMC-Video Daughter Board www.xilinx.com/products/spartan3a/3adsp
7. “From Register” <http://www.mathworks.com/access/helpdesk/help/toolbox/simulink>