REVISTA DIGITAL DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES
TELEM@TICA
Revista Telem@tica. Vol. 14. No. 3, septiembre-diciembre, 2015, p.54-67 ISSN 1729-3804

IMPLEMENTACIÓN DE UN MÓDULO IP PARA UN MODELO DECODIFICADOR H.264/AVC SOBRE FPGA.

Laura Quesada del Busto[1], Gustavo Javier Aguirre Soler[2], Osmany Yaunner Nuñez[3], Orlando Landrove Gámez[4]

LACETEL, Instituto de Investigación y Desarrollo de Telelcomunicaciones
Ave. Independencia, #34515, km 14[1/2], Repto 1[ro] de Mayo, Boyeros, La Habana, Cuba, Código Postal: 19200
[1]e-mail: **laura@lacetel.cu**
[2]e-mail: **gustavo.aguirre@lacetel.cu**
[3]e-mail: **osmany@lacetel.cu**
[4]e-mail: **landrove@lacetel.cu**

RESUMEN

El estándar H.264/AVC constituye uno de los formatos de compresión de video más utilizados hoy en día en los sistemas de Televisión Digital Terrestre. En ***LACETEL***, Instituto de Investigación y Desarrollo de Telecomunicaciones se cuenta con un modelo decodificador H.264/AVC implementado en una plataforma FPGA con microprocesador empotrado. Este modelo es completamente funcional pero su velocidad de procesamiento no le permite operar en tiempo real. En este trabajo se insertó al sistema decodificador H.264/AVC un módulo IP para sustituir a la función software que ejecuta la decodificación aritmética binaria de bypass, que es dentro del núcleo CABAC la más frecuentemente invocada. Además, como módulos IP auxiliares, se insertaron un temporizador y un controlador de interrupciones, para evaluar su desempeño. Se utilizaron como herramientas de diseño los softwares: XPS para la implementación de la plataforma hardware y SDK para el diseño del subsistema software, ambos proporcionados por Xilinx. Como resultados se obtuvieron los tiempos de procesamiento al decodificar varios videos en diferentes escenarios de prueba.

PALABRAS CLAVES: Televisión Digital Terrestre, H.264/AVC, módulos IP, XPS, SDK.

ABSTRACT

Nowadays, H.264/AVC is one the most used video compression standards in Terrestrial Digital Television Systems. ***LACETEL***, Research and Development Telecommunications Institute, has implemented an H.264/AVC video decoder model using FPGA technology with an embedded processor. In this paper an IP module was inserted to the H.264/AVC decoder system to replace software function responsible of bypass binary arithmetic decoding. Bypass decoding is the most frequently invoked core block of CABAC's decoding. Also, as auxiliary IP modules, a timer and an interrupt controller were added to evaluate the system performance. The design tools used were XPS for hardware platform implementation and SDK for software subsystem design, both of them provided by Xilinx. As results, the processing times of decoding many H.264/AVC videos were gotten in different test benches.

KEYWORDS: Terrestrial Digital Television, H.264/AVC, IP modules, XPS, SDK.

INTRODUCCIÓN

Las tecnologías de procesamiento, almacenamiento y transmisión de escenas en movimiento representadas por una secuencia de imágenes, son denominadas tecnologías de video. Estas tecnologías pueden implementarse a través de medios electrónicos digitales o analógicos. Las tecnologías de video digitales ofrecen grandes ventajas sobre las analógicas, tales como calidad de imagen e inmunidad a las interferencias, además los videos digitales ocupan un espacio significativamente menor gracias a las diferentes técnicas de compresión existentes. Por estas y otras razones, en las últimas décadas ha ocurrido una evolución de la mayoría de los sistemas de video analógicos a digitales.

La migración hacia la televisión digital es un proceso que se ha estado llevando a cabo en muchos países desarrollados desde la década de los 80. Actualmente nuestro país se encuentra inmerso en este proceso, debido a las múltiples ventajas que este trae consigo. En el sistema de televisión digital los programas son comprimidos antes de ser transmitidos, utilizando alguna de las normas o estándares de compresión actuales: MPEG2, AVS y H.264/AVC.

LACETEL cuenta con un modelo decodificador H.264/AVC implementado en un sistema con microprocesador empotrado en FPGA (siglas de Field Programmable Gate Array). Este modelo es completamente funcional pero su tiempo de procesamiento es elevado, dado que el algoritmo en que se basa está descrito en lenguaje C. Por esta razón no puede ser utilizado por aplicaciones en tiempo real.

En trabajos anteriormente realizados en ***LACETEL***, con vistas al desarrollo del modelo decodificador H.264/AVC, se diseñó el algoritmo VHDL para la implementación de un módulo IP de decodificación de bypass del núcleo CABAC (siglas de Codificación Aritmética Basada en Adaptación de Contexto). Con este módulo IP se pretende sustituir la lógica secuencial del decodificador de bypass por lógica concurrente. El diseño de este módulo hardware forma parte de la metodología definida para ir optimizando el procesamiento del sistema. Esta se fundamenta en rediseñar e insertar aquellos módulos que requieren mayor velocidad a partir de módulos IP. De esta manera y de forma gradual la solución contendrá lógica concurrente en sustitución de lógica secuencial, lo que se espera tribute a un mejor desempeño.

El modelo decodificador H.264/AVC se implementa sobre el FPGA Virtex 5 de Xilinx, donde el microprocesador empotrado PowerPC 440 se encarga de la ejecución del algoritmo del modelo decodificador, el cual está descrito en lenguaje C. Para el desarrollo de herramientas con vista a su optimización se prevé hacer uso de las capacidades que presenta el FPGA Virtex 5, tales como la adición al sistema de Núcleos de Propiedad Intelectual (IP) que permitan un desarrollo híbrido entre software y hardware dentro del mismo entorno. En este trabajo se implementa y se inserta en el sistema el módulo IP decodificador de “bypass”. Como módulos auxiliares también fueron insertados el temporizador Logicore IP XPS Timer/Counter (v1.02a) y el controlador de interrupciones LogiCORE IP XPS InterruptController (v2.01a).

PLATAFORMA DE IMPLEMENTACIÓN

La inserción de los núcleos XPS Timer/Counter y XPS INTC le brinda al sistema la posibilidad de temporizar el procesamiento de cualquiera de sus elementos y la capacidad de utilizar la interrupción como modo de atención del procesador a sus periféricos. El módulo hardware decodificador de bypass se insertó con el objetivo de suplementar a la función software que realiza la decodificación aritmética binaria de bypass.

El núcleo del decodificador CABAC está formado por tres modos de decodificación aritmética binaria: el regular, el de bypass y el de terminación, donde la selección de uno u otro modo depende del elemento de sintaxis que se quiere decodificar. [1]

El decodificador de bypass implementado en VHDL fue diseñado a partir del software de referencia JM 18.6, en el cual la función "**unsigned int** biari_decode_symbol_eq_prob (DecodingEnvironmentPtr dep)" se encarga de la decodificación de bypass, por lo cual la implementación VHDL constituye una adaptación hardware de dicha función.

El Controlador de Interrupciones LogiCORE IP XPS InterruptController (v2.01a), en lo adelante XPS INTC, concentra múltiples entradas de interrupciones provenientes de los dispositivos periféricos del sistema a una única salida de interrupción conectada al procesador del sistema, el procesador PowerPC. Los registros de chequeo, habilitación y reconocimiento de interrupciones son accedidos a través de una interfaz esclava al bus PLB. [2]

Sus principales características son: [2]

- Conexión a 32 bits del Bus PLB 4.6 esclavo en sus configuraciones de 32, 64 y 128 bits de bus de dato.
- Pueden ser configuradas hasta 32 entradas de interrupción.
- Única salida de interrupción.
- Posibilidad de conectar en cascada para proveer mayor número de interrupciones.
- La prioridad entre interrupciones está determinada por su posición en el vector. El bit menos significativo (LSB en este caso el bit 0) es el de mayor prioridad.
- Registro de Habilitación de Interrupción para selectivamente deshabilitar entradas de interrupción individuales.
- Cada entrada puede ser configurada para ser capturada por flanco o por nivel de sensibilidad. Si es por flanco puede ser configurada por subida o caída. Si es por nivel puede ser activa por nivel alto o bajo

El Logicore IP XPS Timer/Counter (v1.02a) es un Módulo Contador/Temporizador (MCT) diseñado para trabajar conectado a través del bus PLB. El mismo presenta las siguientes características: [3]

- Posibilidad de conectarse a un bus de 32 bits.
- Conexión al bus PLB v4.6
- Dos temporizadores programables con capacidad para generación de eventos, interrupción y captura de eventos.

- Salida de Modulación de Ancho de Pulso (PWM).

INSERCIÓN DE MÓDULOS IP EN EL SISTEMA DECODIFICADOR H.264/AVC

Diseño del subsistema hardware

El sistema sobre el cual se implementa el decodificador H.264/AVC está formado por un subsistema hardware y un software. El subsistema hardware se creó con el apoyo de la herramienta XPS del EDK de Xilinx. Las especificaciones del sistema fueron conformadas mediante el empleo del Asistente Constructor de Sistema BSB (en inglés, Base System Builder). En este se seleccionó el bus PLB, la tarjeta de desarrollo ML507, el trabajo con un único procesador para el sistema, el procesador PowerPC, la frecuencia del procesador 400 MHz, la frecuencia del bus PLB 125 MHz, además de la interfaz para la depuración FPGA JTAG.

Los periféricos que conformaron el sistema son:

- Módulo de memoria Bram.
- Módulos para atención del puerto RS232 UART. Se utiliza para reportar actividad del sistema.
- Módulo para memoria DDR-2 de la tarjeta ML507.En la memoria se trabajará con datos dinámicos.
- Módulo Temporizador/Contador.
- Módulo Controlador de Interrupciones.
- Controlador del Bus PLB v46.
- Módulo para el controlador de Compact Flash. En esta memoria se almacenarán los videos de entrada al sistema y la salida decodificada.
- Módulo decodificador de bypass-ip. Este módulo fue creado e importado por un asistente del XPS, y posteriormente fue añadido al sistema, direccionado y conectado al bus PLB.

Todos los periféricos fueron conectados al procesador PowerPC a través de sus interfaces con el bus PLB. Las direcciones físicas correspondientes a cada dispositivo fueron asignadas automáticamente.

Con vistas a la utilización de interrupción como vía para la atención del procesador a varios periféricos se insertó el Controlador de Interrupciones. Para informar al procesador de que una interrupción fue generada, la salida IRQ del controlador de interrupciones fue conectada a la entrada de interrupción externa del PowerPC. Al crear el módulo decodificador de bypass IP se configuraron una serie de opciones que permitieron su funcionamiento tales como: el módulo IP no posee puertos externos; la atención al periférico se estableció por interrupción, conectando la salida FIN del módulo IP a una de las entradas del controlador de interrupciones. La comunicación entre el microprocesador y el periférico se realiza con la ejecución de lecturas y escrituras en registros de 32 bits previamente configurados, en los cuales son mapeadas cada una de las señales internas que el dispositivo posee para su funcionamiento. En el caso del decodificador de bypass IP fueron definidos 11 registros.

Al contar con la implementación VHDL del decodificador de bypass, para la importación del periférico al sistema hardware en el XPS, solo se añadió dicha implementación al módulo IP, además se realizó el mapeo de las señales internas del dispositivo en los registros como se muestra en la figura 1.

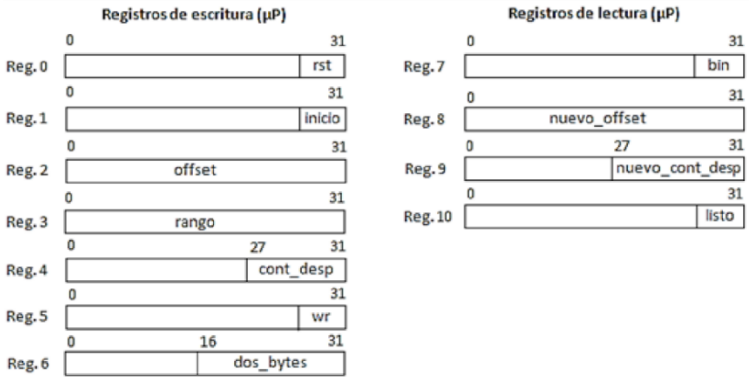


Figura 1: Mapeo de las señales internas en los registros.

El Temporizador/Contador se utilizó en modo de generación, en el cual es generada una señal de interrupción al culminar un conteo programado, la cual fue conectada a una de las entradas del controlador de interrupciones.

En la figura 2 se muestran los bloques que conforman la arquitectura del FPGA para este diseño. La tabla 1 resume la cantidad de recursos del FPGA utilizados en el diseño. Como se puede observar los porcentajes de utilización mostrados son relativamente bajos. Esto permite que en un futuro se puedan insertar más módulos hardware dentro del mismo FPGA.

Tabla 1: Resumen de utilización del FPGA

Lógica	Utilizada	Disponible	Utilización
Número de FFs de slice	4 239	44 800	9 %
Número de LUTs de slice	3 783	44 800	8 %
Número de slices usados	2 658	11 200	23 %
Número de IOBs	213	640	33 %
Memoria total usada (KB)	216	5 318	4 %
Número de BUFIOs	8	80	10 %
Número de PPC 440s	1	1	100 %

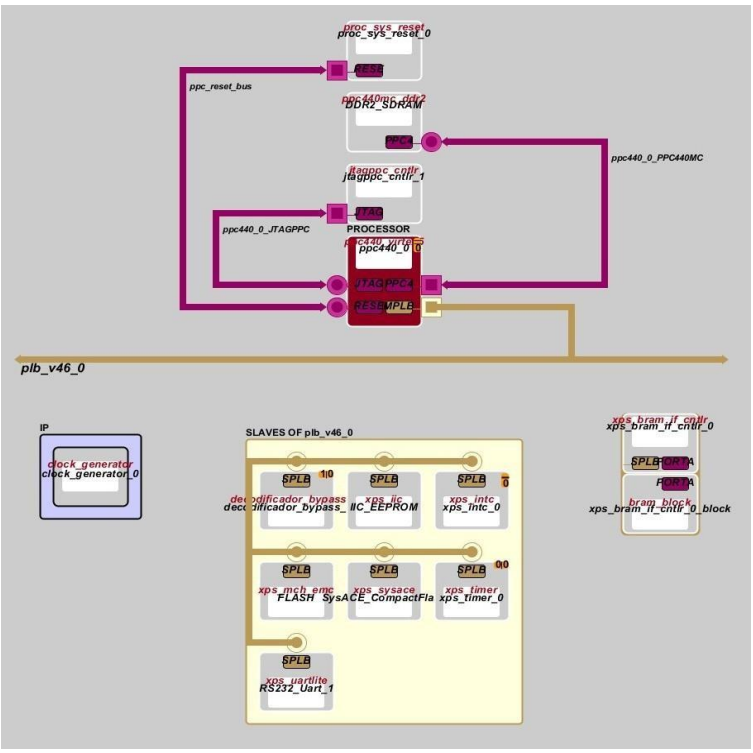


Figura 2: Arquitectura del subsistema hardware

Diseño del subsistema software

Por otra parte el subsistema software se desarrolló en el SDK. La comunicación entre el procesador y los módulos hardware se realiza mediante el empleo de los "drivers". Existen "drivers" de bajo y de alto nivel los cuales están contenidos en el Paquete de Soporte de Tarjeta (BSP, por sus siglas en inglés). El BSP es la forma de vincular los elementos incluidos en la plataforma hardware con la nueva aplicación software creada y se genera automáticamente después exportar o actualizar la plataforma hardware desde el XPS hacia el SDK.

Para el trabajo con el temporizador se emplearon algunas de las funciones de alto nivel declaradas en el archivo de cabecera xtmrctr.h y definidas en diferentes archivos fuentes. Las funciones utilizadas para la configuración del temporizador fueron: XTmrCtr_Initialize, XTmrCtr_SetOptions, XTmrCtr_SetHandler, XTmrCtr_SetResetValue y XTmrCtr_Reset. Con ellas se habilitó el modo de trabajo con interrupción y el modo de recarga automática; además se estableció el 0 como valor de reset, tmr_cnt_isr como función de retorno de interrupción y como tipo de conteo el incremental. La lógica que se siguió para el trabajo con el temporizador consistió en configurarlo al inicio del programa dentro de la función main, y de esta forma tenerlo listo para la posterior temporización de un determinado bloque o función. En ese caso se insertó la función XTmrCtr_Start justo antes de donde inicia lo temporizado y XTmrCtr_GetValue y XTmrCtr_Stop se insertaron donde finaliza. Para obtener el tiempo real que demoró lo que fue temporizado se utilizó la ecuación:

$$tiempo = \frac{valor\_obtenido + \max\_cont * tmr\_overflow}{frec\_bus} \quad (1)$$

Donde *valor_obtenido* es el valor que devuelve la función XTmrCtr_GetValue; *max_cont* es el máximo valor que alcanzan los registros del contador, en este caso FFFFFFFF; *tmr_overflow* es la cantidad de veces que se ha desbordado el temporizador y *frec_bus* es la frecuencia del bus PLB utilizada, en este caso 125 MHz.

La función *tmr_ctr_isr*, se creó para ser utilizada como función de retorno de interrupción correspondiente al temporizador, como se dijo anteriormente. Esta función se utilizó para llevar el conteo de las veces que el temporizador se desborda a través de la variable *tmr_overflow* que fue inicializada en 0 y se incrementa en 1 cada vez que se ejecuta la función.

El controlador de interrupciones se configuró y se puso a trabajar seguidamente con el objetivo de que estuviese listo para atender cualquier solicitud de interrupción proveniente de los dispositivos que tiene conectados. Para la ello se utilizaron las funciones XTmrCtr_Initialize, XTmrCtr_Connect, XTmrCtr_Enable y XTmrCtr_Start. Estas funciones de alto nivel se brindan en el BSP y se encuentran declaradas en el archivo de cabecera xintc.h y definidas en xintc.c.

Otro aspecto que se configuró fue la habilitación y reconocimiento de interrupciones externas en el procesador PowerPC, de tal forma que el procesador sea capaz de detectar las solicitudes de interrupción provenientes del controlador de interrupciones y se encargue de ejecutar las funciones de retorno de interrupción correspondientes según le indique el controlador. Para estas configuraciones se utilizaron las funciones XExc_Init, XExc_RegisterHandler y XExc_mEnableExceptions.

Para la transferencia de datos entre el procesador y el módulo hardware de decodificador de bypass se utilizaron las funciones: DECOD_BYPASS_IP_mWriteReg, la cual lee y devuelve el dato almacenado en el registro especificado del IP, y DECOD_BYPASS_IP_mReadReg, la cual escribe el dato en el registro especificado dentro del módulo hardware. Ambas funciones se brindan en el BSP y se encuentran declaradas en el archivo de cabecera decod_bypass_ip.h y definidas en decod_bypass_ip.c. Para que estas funciones escriban o lean en el periférico correctamente, se deben especificar la dirección base del IP, la cual se brinda en el archivo de cabecera xparameters.h y el desplazamiento en la dirección del registro especifico respecto a la dirección base.

Evaluación del desempeño temporal del modelo decodificador H.264/AVC

Para evaluar el desempeño temporal del decodificador H.264/AVC dentro de un sistema embebido en FPGA se realizaron pruebas para tres diferentes casos, los cuales diferían en la forma de ejecución de la decodificación aritmética de bypass. En cada uno de estos casos se midieron los tiempos de decodificación de tres videos codificados H.264/AVC diferentes en cuanto a formato y tamaño como se muestra en la Tabla 2.

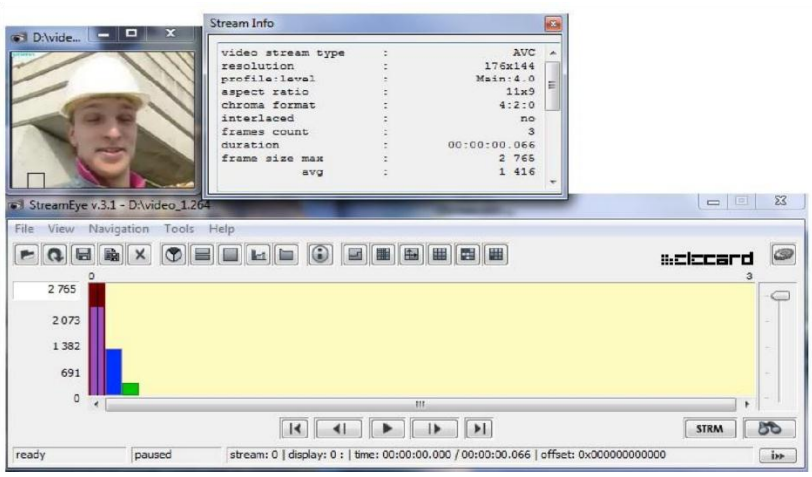
En todas las pruebas la temporización se inicia dentro de la función main, luego de las configuraciones iniciales y justo antes del comienzo de la decodificación. La decodificación termina en la función int

CloseDecoder (), en ese punto fue obtenido el valor del registro de conteo del temporizador y detenida la temporización. Luego se calcula el tiempo real que dicho valor representa.

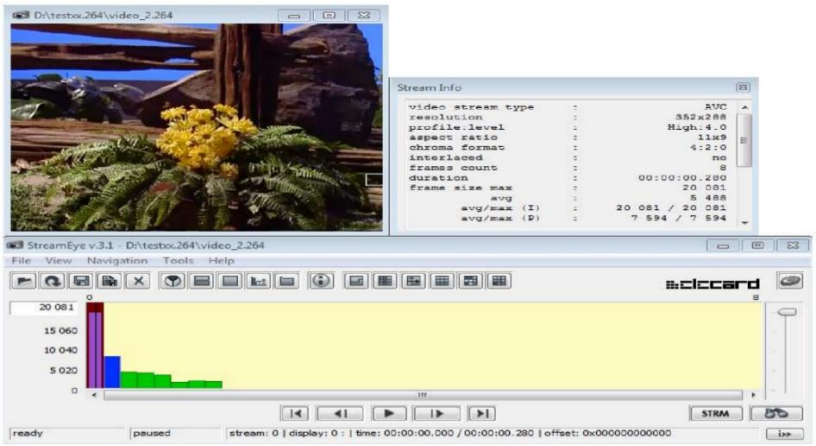
Tabla 2: Videos codificados H.264/AVC.

Video	Resolución en píxeles	Cantidad de cuadros	Perfil
Video_1	176 x 144	3	Principal
Video_2	352 x 240	8	Alto
Video_3	176 x 144	10	Alto 4:2:0

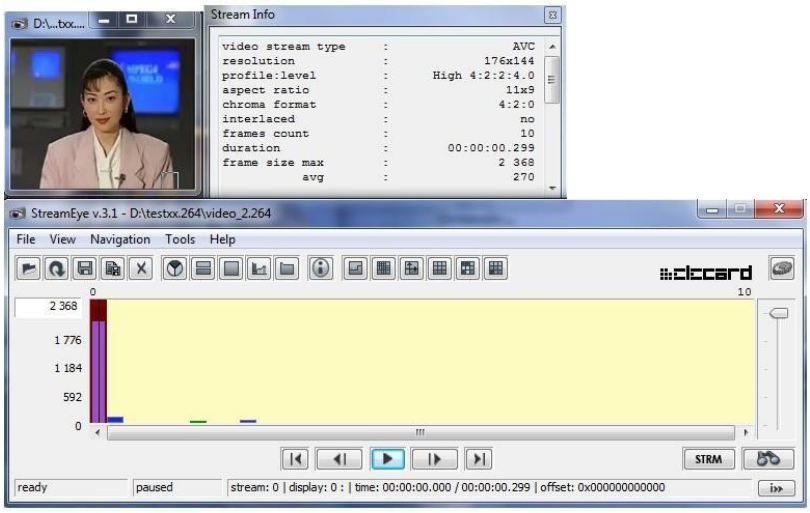
Dado que CABAC solo se admite en los perfiles principales y altos, entre los videos utilizados en las pruebas no se incluyó un video de perfil bajo. Esto no significa que el decodificador obtenido no sea capaz de descomprimir un video de perfil bajo, sino que para este perfil no se obtienen resultados significativos ya que no se ejecuta la decodificación de bypass. Con la utilización del software Elecard se asegura que cada uno de los videos codificados utilizados como fuente de entrada al sistema decodificador cumplan con los requerimientos del estándar H.264/AVC. La Fig. 3 visualiza la estructura de las muestras de video.



a)



b)



c)

Figura 3: Análisis mediante Elecard de videos codificados H.264/AVC. a) Video_1. b) Video_2. c) Video_3

El primer caso evaluado fue la temporización del modelo decodificador H.264/AVC donde la decodificación aritmética de bypass se realiza mediante software, a través de la función: unsigned int biari_decode_symbol_eq_prob (DecodingEnvironmentPtr dep).

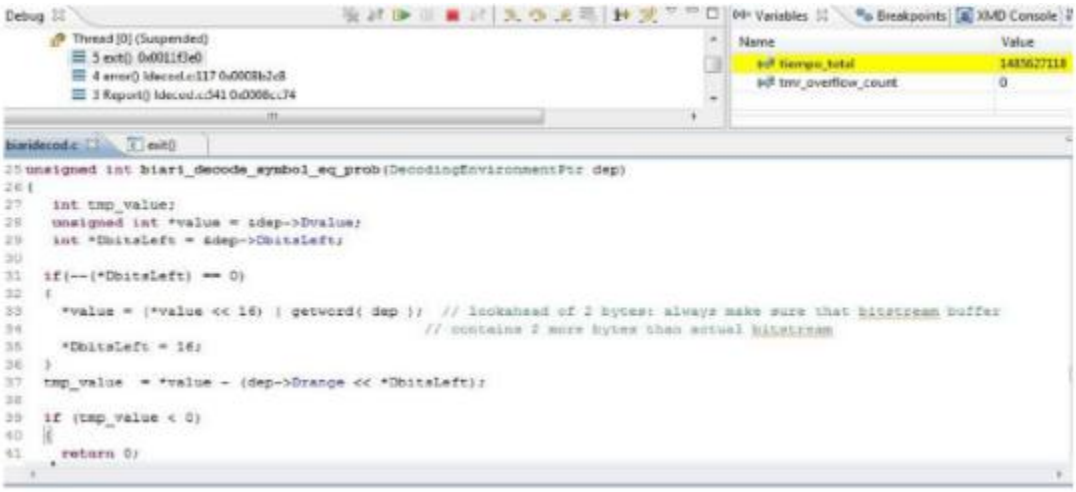


Figura 4: Decodificación de Video_2 para el primer caso.

Tabla 3: Tiempos de decodificación para el primer caso.

Video	Valor del registro	Desbordamiento	Tiempo (s)
Video_1	323839288	NO	2,5907
Video_2	1485627118	NO	11,8850
Video_3	474519877	NO	3,7962

En la figura 4 se visualiza un fragmento de la función biari_decod_symbol_eq_porb, además se muestra el valor leído desde el registro de conteo del temporizador al concluir la decodificación del video_2, guardado en la variable tiempo_total. La Tabla 3 resume los resultados obtenidos en este caso para cada uno de los videos.

En el segundo caso la decodificación aritmética de bypass se realiza mediante el módulo IP decodificador de bypass. La atención al módulo se realiza mediante interrupción. Cuando la interrupción es reconocida por el PowerPC, este ejecuta la subrutina de interrupción correspondiente al dispositivo que la solicitó, en este caso la función utilizada para el módulo decodificador de bypass fue DECOD_BYPASS_IP_Intr_Handler.

En la decodificación de bypass, inicialmente se envían a las entradas offset, rango y cont_desp del módulo, los datos contenidos en las variables DbitsLeft, Drange y Dvalue respectivamente, dentro del puntero dep. Luego se envía el bit de inicio en 1 para que el módulo genere los nuevos valores en sus salidas. La salida fin, cambia de 0 a 1 y provoca que se genere la solicitud de interrupción. Durante la ejecución de la función DECOD_BYPASS_IP_Intr_Handler se lee la salida listo y en el caso que lo requiera se envían al módulo IP los datos correspondientes a las entradas dos_bytes y WR. Cuando se retorna de la interrupción, se leen las salidas nuevo_offset, nuevo_conteo_desp, y se

actualizan DbitsLeft y Dvalue, luego se lee bin que es el valor de retorno de la decodificación de bypass y se resetea la señal de inicio.

En la figura 5 se visualiza un fragmento de la función biari_decod_symbol_eq_porb en el cual se evidencia el envío de los datos hacia los registros del módulo IP decodificador de bypass, además se muestra el valor leído desde el registro de conteo del temporizador al concluir la decodificación del video_2, guardado en la variable tiempo_total. La Tabla 4 resume los resultados obtenidos en este caso para cada uno de los videos.

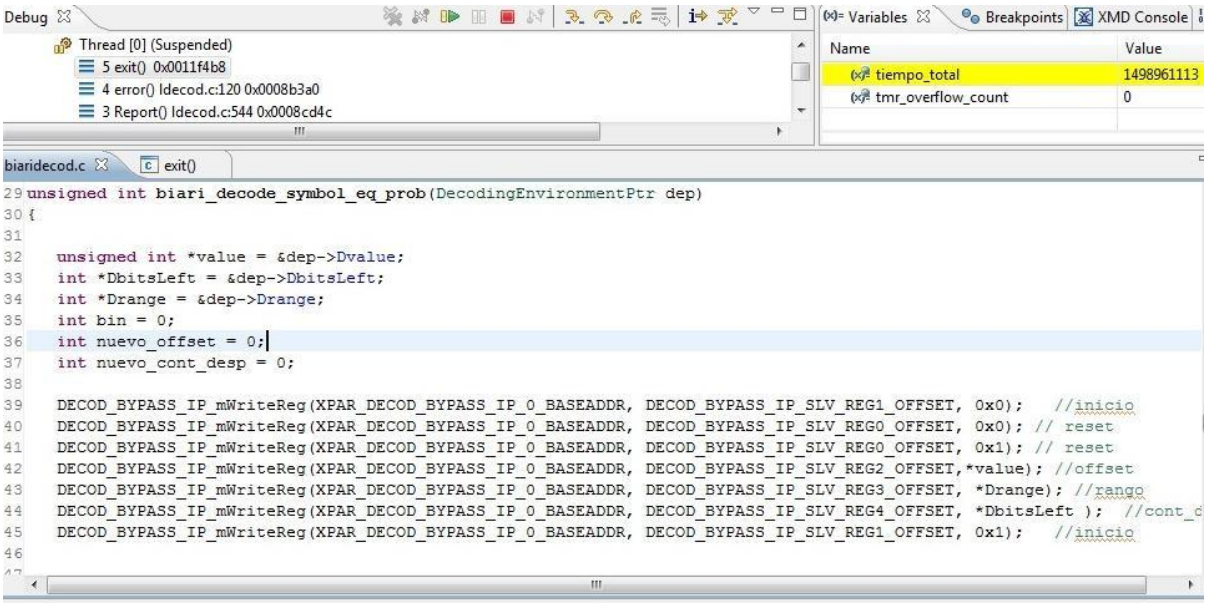


Figura 5: Decodificación de Video_2 para el segundo caso.

Tabla 4: Tiempos de decodificación para el segundo caso.

Video	Valor del registro	Desbordamiento	Tiempo (s)
Video_1	333857008	NO	2,6709
Video_2	1498961113	NO	11,9917
Video_3	486171142	NO	3,8894

En el tercer caso la decodificación aritmética de bypass también se realizó mediante el módulo IP de decodificación de bypass. En este caso se realizan cambios con vistas a aprovechar las ventajas del trabajo en un sistema híbrido hardware/software y reducir los tiempos de procesamiento.

Al importar la implementación VHDL al módulo IP decodificador de bypass se hicieron algunas modificaciones en los registros, tales como el mapeo de varias señales en un mismo registro, además

fue conectada la entrada Bus2IP_Reset al reset de la implementación y la señal de salida fin no se utilizó. En la figura 6 se muestra el nuevo mapeo de las señales en los registros.

Figura 6: Mapeo de las señales internas en los registros modificados.

Estas modificaciones fueron realizadas para reducir la cantidad de ciclos de escritura y lectura a través del bus PLB empleados, puesto que estos son los que mayores retardos provocan en el trabajo con el módulo IP. A pesar de que complejizan obtener los valores de cada una de las señales.

En la figura 7 se visualiza un fragmento de la función biari_decod_symbol_eq_porb en el cual se evidencia el envío y la obtención de los valores de los registros del módulo IP decodificador de bypass, además se muestra el valor leído desde el registro de conteo del temporizador al concluir la decodificación del video_2, guardado en la variable tiempo_total. La Tabla 5 resume los resultados obtenidos en este caso para cada uno de los videos.

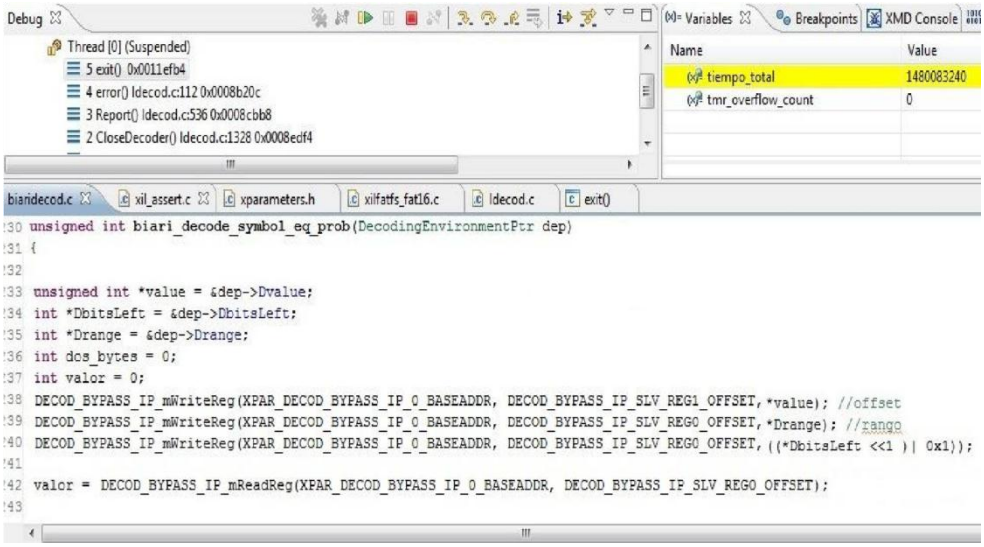


Figura 7: Decodificación de Video_2 para el tercer caso.

Tabla 5: Tiempos de decodificación para el tercer caso.

Video	Valor del registro	Desbordamiento	Tiempo (s)
Video_1	322962782	NO	2,5837
Video_2	1480083240	NO	11,8407
Video_3	475229251	NO	3,8018

De las pruebas presentadas anteriormente, el primer caso es presentado con vistas a ser utilizado como referencia para comparar los valores de los casos posteriores.

Los resultados de estas pruebas muestran que el segundo caso, donde la decodificación de bypass se realiza a través del módulo IP atendido por interrupción, es el que presenta mayores retardos. Esto ocurre porque el proceso que va desde la solicitud de interrupción por parte del periférico hasta la ejecución de la función de atención a la interrupción, incorpora tiempos adicionales al sistema. Además, este caso presenta mayor cantidad de los ciclos de escritura y lectura del procesador al módulo IP a través del bus PLB respecto al tercer caso, lo cual incrementa os tiempos de procesamiento.

La interrupción es un método útil en casos donde el procesador se encuentra realizando otras funciones mientras el periférico está trabajando, pero en este caso no se pueden aprovechar totalmente las ventajas que brinda la interrupción. Esto se debe a que el proceso de decodificación CABAC es altamente secuencial y hasta que los datos de retorno de la decodificación de bypass no se obtengan la decodificación no continúa y el procesador no hace nada más.

A pesar de que este caso se descarta como solución a emplear en la decodificación de bypass es presentado en este trabajo ya que fue el primer sistema que se diseñó, es totalmente funcional y sirvió de base para la obtención del tercer caso.

Por otra parte se observa que para el tercer caso donde la decodificación de bypass se realiza mediante el módulo IP con modificaciones se obtienen tiempos de decodificación menores y similares a los obtenidos en el primer caso. Esto se debe a que se reducen la cantidad de ciclos de lectura y escritura del procesador al periférico, dado a la unión de varias señales dentro de un mismo registro. Además, se obtienen los valores desde el periférico sin recurrir a métodos de atención como la encuesta o la interrupción, lo cual se demuestra que es totalmente válido ya que los tres videos de prueba se decodifican exitosamente.

CONCLUSIONES.

La adición de módulos IP al sistema decodificador H.264/AVC implementado en FPGA le brinda la capacidad de contar con herramientas para optimizar y evaluar el desempeño del decodificador a partir de un desarrollo híbrido entre software y hardware.

La comunicación entre el procesador y el módulo IP de decodificación de bypass se realiza mediante ciclos de lectura y escritura a los registros del módulo a través del bus PLB. La concatenación de varias

señales a un mismo registro permite reducir la cantidad de ciclos empleados, de este modo se disminuyen los tiempos de procesamiento.

El decodificador H.264/AVC presentado todavía no se encuentra listo para brindar servicio en tiempo real. Sin embargo, constituye una plataforma con capacidad de soportar próximas implementaciones y optimizaciones, tales como la inserción de nuevos módulos de forma similar a la presentada en este trabajo.

REFERENCIAS.

1. YAUNNER NÚÑEZ, Osmany. Tesis de Diploma ***LACETEL***: "Propuesta de optimización de un decodificador CABAC". Junio 2014: p. 59-67. Disponible en: http://www.lacetel.cu/eventos/2do-foro-internacional-de-tvd.html

2. XILINX. "LogiCORE IP XPS Interrupt Controller (v2.01a)". Abril 2010.

3. XILINX. "LogiCORE IP XPS Timer/Counter (v1.02a) ". Abril 2010

4. LANDROVE GAMEZ, Orlando. "Modelo funcional de un decodificador H.264/AVC". EAC [online]. 2014, vol.35, n.3, pp. 48-59. ISSN 1815-5928. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59282014000300005&lng=es&nrm=iso