



Modelo funcional de un decodificador H.264/AVC

Orlando Landrove Gámez

RESUMEN / ABSTRACT

La Televisión Digital Terrestre (TVD-T) es una tecnología en constante desarrollo, con elevada actualidad de acuerdo al Estado del Arte. Su adopción, así como sustitución de la tradicional cadena de transmisión analógica es un hecho en muchos países del mundo, debido a sus múltiples ventajas y correspondencia con otras tecnologías modernas. Cuba, como otros, se encuentra en franco despliegue de la misma, con programas definidos por cada una de las entidades responsables de su apropiada implementación, donde la asimilación de tecnología resulta decisivo para un desarrollo real e independiente en este campo. Entre los bloques básicos que conforman la cadena de Televisión Digital, la compresión y descompresión de video constituyen un elemento esencial, en un escenario donde el ancho de banda para la transmisión es finito y cotizado. H.264/MPEG-4 Parte 10 es actualmente el estándar de compresión de video mayormente empleado para la radiodifusión de Televisión Digital. Los receptores que cumplan con H.264 deben tener la capacidad de decodificar su formato. En este trabajo se describe la utilización de software descrito en lenguaje C para modelar el funcionamiento de un decodificador H.264. Se utiliza un FPGA Virtex 5, de Xilinx, para empotrar un sistema con PowerPC como elemento principal que soportará el modelo propuesto. A través de la Tarjeta de Desarrollo ML507 se demuestra el funcionamiento del sistema. Para esto se utilizan ficheros de video comprimidos con el formato H.264 que a su vez son decodificados obteniéndose ficheros de video en formato descomprimido YCbCr. Estos constituyen el resultado del experimento.

Palabras claves: TVD-T, H.264, YCbCr, FPGA, Lenguaje C, Sistema Embebido, VHDL.

Terrestrial Digital Television. (DTV-T) is a constantly evolving technology with current update according to the State of the Art. Its adoption, as well as replacing the traditional analog transmission chain is a fact in many countries, because of its many advantages and synergy with other modern technologies. Cuba, like others, is already in deployment phase, following scripts defined by entities in charge of its proper implementation, where the assimilation of technologies is crucial for a real and independent development in this field. Among the basic blocks that make up the Digital TV chain, the compression and decompression of video are an essential element in a scenario where the bandwidth for transmission is finite and quoted. H.264/MPEG-4 Part 10 is currently the video compression standard widely used for digital television broadcasting. Receptors that fulfill H.264 must be able to decode the format. This paper described the use of software in C language to model the performance of an H.264 decoder. FPGA Virtex 5 from Xilinx is used to embed a system with PowerPC as the main element that will support the proposed model. Through the ML507 Development Board is realized the system operation. H.264 video format files are used, which in turn are decoded to obtain uncompressed video files using YCbCr format. These become the result of the experiment.

Key words: DTV-T, H.264, YCbCr, FPGA, C Language, Embedded System, VHDL.

Functional Model of an H.264 decoder.

INTRODUCCION

El creciente desarrollo de la tecnología de video digital impone su inclusión en múltiples escenarios actuales, tanto en sistemas integrados de telecomunicaciones, aplicaciones específicas, como en el consumo general de la población a través de numerosos tipos de dispositivos electrónicos que soportan varios de los formatos de video existentes.

En el caso de la Televisión Digital (TVD), la compresión de video digital juega un importante papel en un escenario en que el ancho de banda de transmisión es limitado, por tanto la optimización en su aprovechamiento es determinante para implementar nuevos servicios que demanden grandes volúmenes de información.

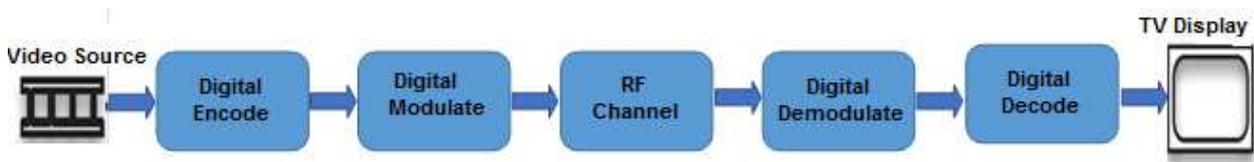


Figura 1. Bloques básicos de una cadena de TVD.

El desarrollo de técnicas, así como la estandarización de complejos algoritmos permiten la transmisión de información de video utilizando menor cantidad de datos, sin afectar la apreciación subjetiva del ojo humano. Algunas de estas técnicas han sido estandarizadas y documentadas para la disposición y total compatibilidad de productos afines en el mercado.

Formatos de Video que más se utilizan en Televisión Digital.

Los formatos de compresión más importantes de acuerdo al alcance y a su utilización son publicados y reconocidos por organizaciones especializadas, tales como la Unión Internacional de Telecomunicaciones¹, Organización Internacional para la Estandarización², Grupo de Experto en Imágenes y Movimiento³. A continuación se comentan los tres estándares que actualmente lideran la compresión de video en la TVD-T.

ISO/IEC 13818-2 (H.262 o MPEG-2 Parte 2), conocido por MPEG-2 para Video. Este formato fue desarrollado en 1994 y es utilizado para transmisiones de TVD-T, TVD-C (Televisión Digital por Cable) y TVD-S (Televisión Digital Satelital) [1]. Toma experiencias similares de MPEG-1, pero añade características como el soporte de video entrelazado, presenta algoritmos de predicción para cuadros presentes, posteriores y bidireccionales, mayor tasa de bits así como una serie de perfiles y niveles para definir aplicaciones específicas dentro del propio estándar. Además MPEG-2 soporta videos de varios tipos de relaciones de aspecto, formatos 4:2:2 y 4:4:4, adaptación frame/field⁴ DCT⁵, entre otros [2]. MPEG-2 parte 2 es todavía usado en muchos escenarios. Muchos operadores, tanto para radiodifusión como para satélites y cables utilizan este formato. La gran mayoría de Set-top boxes⁶, soportan su decodificación.

No obstante, otros formatos más recientes incorporan nuevos algoritmos y técnicas convirtiéndolos en más eficientes, de esa manera algunos como H.264 alcanzan un mayor despliegue y soporte por muchos de los operadores en el mundo. Mientras que otros como AVS, crece entre los consumidores y en el mercado cada vez más.

IEEE Std. 1857™-2013 (AVS1-P2), conocido como AVS para Video. Este códec fue desarrollado por el grupo AVS, en su gran mayoría desarrolladores chinos, totalmente respaldados por su gobierno y en conjunto de muchas universidades e instituciones científicas. Su propósito fue reducir la dependencia de regalías y patentes de estándares extranjeros. En febrero el año 2006 oficialmente fue un estándar nacional para la TVD. En octubre del 2009, la UIT-T para IPTV reconoció a AVS como uno de los tres códec para este servicio, junto a H.264/AVC y MPEG-2 [3]. Asimismo en junio del 2011 se determina que AVS es el único

¹ ITU, por sus siglas en inglés: International Telecommunication Union.

² ISO, por sus siglas en inglés: International Organization for Standardization

³ MPEG, por sus siglas en inglés: Motion Picture Expert Group.

⁴ Frame/field: original en inglés. Cuadro/campo en su traducción al español.

⁵ DCT, por sus siglas en inglés. Discrete Cosine Transform.

⁶ Set-top Box (STB): original en inglés: refiere a receptor digital para servicios de TV, como STB-C, STB-IP, STB-T, STB-S

estándar de video obligatorio para los productos destinados al mercado chino. En noviembre 2012 se lleva a la práctica esta decisión, donde todos los Set-top Box deben soportar de manera obligatoria el códec AVS.

Los desarrolladores de AVS alegan que este códec tiene similar eficiencia que H.264/AVC, empleando un 30% menos de complejidad en su implementación [3]. Además refieren que AVS contiene la mitad de las patentes esenciales que H.264/AVC. Actualmente existen más de 25 fabricantes de chips que adoptaron AVS en sus soluciones, la mayoría de estas en decodificadores. Estos fabricantes se distribuyen entre China continental, Taiwán, EEUU, Europa, Japón y Corea del Sur.

ISO/IEC 14496-10 (H.264/AVC o MPEG-4 Parte 10), conocido por distintas maneras: H.264, MPEG-4 Parte 10 y AVC⁷. Fue desarrollado por el grupo de expertos JVT (Joint Video Team), como resultado de la colaboración conjunta entre MPEG y VCEG⁸, y su primera versión apareció en el año 2003. Su propósito fue optimizar la tasa de bits en comparación con estándares anteriores, sin aumentar demasiado su complejidad para que fuera posible llevarse a la práctica implementaciones del mismo. Actualmente es el códec de video más utilizado por los proveedores de servicio, siendo aprobado por DVB⁹ y ATSC¹⁰. H.264/AVC es cada vez más utilizado en transmisiones de Televisión Digital de Alta Definición (HDTV), servicios de streams de video a través de internet como YouTube, servicios entre teléfonos móviles, por formatos de grabación como DVD entre otros.

Desde el punto de vista de propiedad intelectual, este códec no es libre de usar sin pago de regalías. Implementaciones y reproducción de técnicas para el desarrollo de aplicaciones H.264/AVC están sujetas al pago de licencias por parte de sus desarrolladores. El tema de patentes para este formato es complejo, ya que existen consorcios que se encargan de estos temas legales. Es algo muy importante tener en cuenta para cualquier adopción y despliegue de soluciones empleando H.264/AVC.

Sistema empotrado en FPGA.

El uso de la tecnología FPGA permite responder tanto a necesidades de velocidad de procesamiento, a través del diseño mediante lenguajes de descripción de hardware (Hardware Description Language, HDL), como requerimientos de análisis y cómputo utilizando microprocesadores empotrados en el chip. A través de módulos de propiedad intelectual (Intellectual Property, IP), se conforma un sistema flexible, con capacidad de brindar rapidez y complejidad de análisis.

El sistema que se utiliza en este trabajo está basado en el microprocesador PowerPC 440 (PPC) empotrado en un FPGA de la familia Virtex5 de Xilinx. A través de la tarjeta de Desarrollo ML507, que contiene como elemento principal el FPGA XC5VFX70T, se modela el funcionamiento de un decodificador H.264, el cual basa su hardware en un sistema embebido con PPC y otros periféricos.

CONTENIDO

Para **LACETEL**, Instituto de Investigación y Desarrollo de Telecomunicaciones de Cuba, resulta científicamente atractivo consolidar soluciones referentes a TVD. **LACETEL** tiene, entre otras responsabilidades, la de ser líder en la Transferencia de Tecnología de China a Cuba en el proyecto de despliegue de la TVD en nuestro país. El estudio de códec de video para TVD resulta un tema importante para un futuro en que se logre la independencia de esta tecnología en el país.

H.264/AVC es el códec más utilizado en la actualidad por los consumidores de TVD. Su estudio e investigación abarca varias etapas, desde la búsqueda de información, las pruebas con equipamientos que lo soporten, hasta el acercamiento más profundo al desarrollar módulos propios.

El trabajo describe la concepción de un modelo práctico de un decodificador H.264/AVC. Este modelo presenta los siguientes objetivos:

- Consolidar el estudio y las hipótesis de diseño de los módulos que conforman el formato H.264/AVC.
- Ser un punto de referencia teórico y práctico para los siguientes pasos de optimización de los módulos.
- Constituir la base para una solución apta para la decodificación en tiempo real.

H.264/AVC. Características generales.

⁷ AVC, por sus siglas en inglés. Advanced Video Coding.

⁸ VCEG, por sus siglas en inglés. Video Coding Expert Group

⁹ DVB, por sus siglas en inglés: Digital Video Broadcasting. Estándar de modulación utilizado en TVD.

¹⁰ ATSC, por sus siglas en inglés: Advanced Television System Committee. Estándar de modulación utilizado en TVD.

De manera general, un codificador de video comprime la información en un formato que contenga menos espacio en el momento de ser almacenada o transmitida. El decodificador, de manera inversa, convierte el video comprimido en el formato original, descomprimido. En realidad el formato descomprimido después del proceso de decodificación no es precisamente idéntico al original, debido a las pérdidas durante el proceso de compresión. El término *codec* se debe a la asociación de las palabras en inglés *encoder* y *decoder*.

Una imagen digitalizada está formada por unidades denominados macrobloques. Cada macrobloques contiene información de 16x16 píxeles correspondiente a muestras de luminancia y componentes de color. Los macrobloques a su vez conformarán los llamados *slice*¹¹. Cada campo en el caso de un formato entrelazado, o cada cuadro si se tratase de un formato progresivo se le llamará Imagen. Cada Imagen estará compuesta por uno o más slice. En el caso del video entrelazado, cada cuadro está formado por dos imágenes pertenecientes a dos campos correspondientes: campo par y campo impar. Por otra parte, en el video progresivo, cada cuadro está formado por una sola imagen. Esta imagen, por tanto, contiene la totalidad de los macrobloques del cuadro.

De manera general, en el codificador se realiza una predicción de los macrobloques a partir de macrobloques pertenecientes a slices de referencia. El macrobloque predicho será sustraído al macrobloque que en ese momento se esté procesando, obteniéndose un llamado macrobloque residuo. Este macrobloque, conformado en teoría por la diferencia entre el macrobloque actual y aquel predicho, contendrá menos energía y su tamaño en cantidad de información será menor que cualquiera de los macrobloques que le dieron origen.

No obstante, al macrobloque residuo se les aplica técnicas con el propósito de disminuir la cantidad de información a transmitir o guardar. Se le aplica transformada para separar los datos más importantes de otros redundantes. Además, se cuantifica y se codifica por entropía. De esta manera el dato es mandado hacia el decodificador que a su vez realizará el proceso inverso.

Cuando el dato es codificado, además de mandarse al decodificador, de forma paralela también será objeto del proceso inverso como parte del propio proceso de codificación, con el objetivo de comparar, obtener información del error involucrado y ajustar el mecanismo para la siguiente predicción.

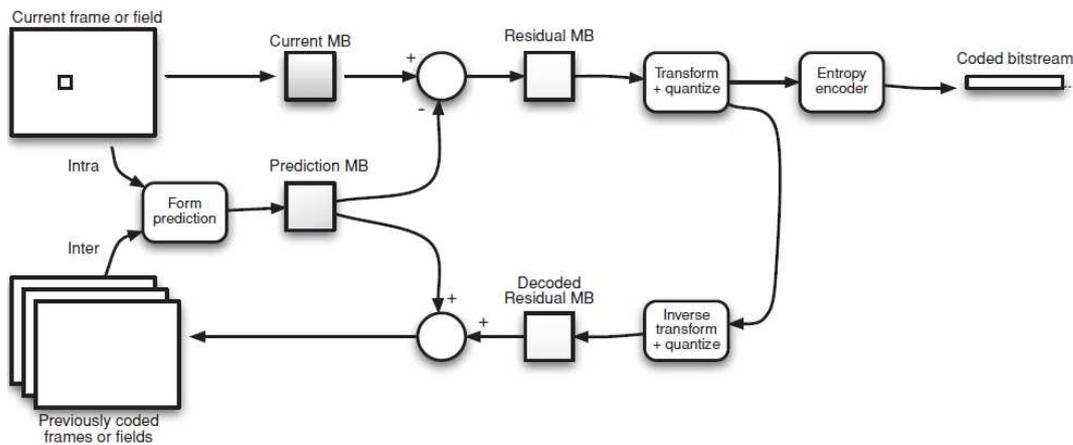


Figura 2. Esquema general de un codificador H.264/AVC [5].

En el decodificador aparecen los bloques de decodificación por entropía, re-escala, transformada inversa, predicción, para reconfigurar el macrobloque residuo. Por otra parte, se predice el macrobloque de referencia de la misma manera en que fue predicho por el codificador. De esta forma, al macrobloque de referencia predicho se le añade la información obtenida del macrobloque residuo, obteniéndose como resultado el macrobloque decodificado, el cual ocuparía un lugar dentro del slice que está siendo decodificando en ese momento.

¹¹ Slice, por su origen en idioma inglés.

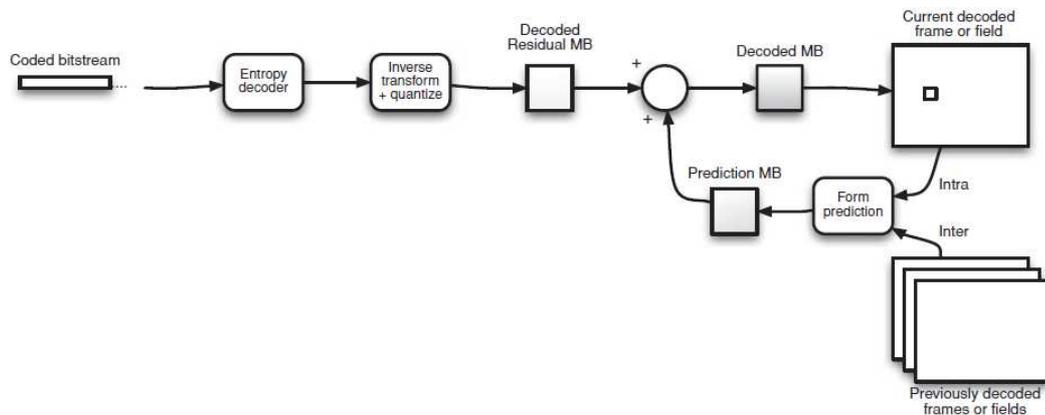


Figura 3. Esquema general de un decodificador H.264/AVC [5].

De manera general, el Estándar *ISO/IEC 14496-10 (H.264/AVC o MPEG-4 Parte 10)* rige las especificaciones que debe tener la cadena de datos codificados para que un decodificador H.264/AVC pueda realizar el proceso. Solo el Bitstream¹² y las Sintaxis quedan sujetos al Estándar, quedando total flexibilidad para el diseño e implementación de técnicas y soluciones siempre que estas cumplan con las especificaciones impuestas para el Bitstream [5].

Actualmente H.264/AVC contiene 21 perfiles y 17 niveles. Los perfiles son el conjunto de algoritmos y herramientas definidos en el Estándar orientados a una aplicación determinada. Los niveles especifican la capacidad de procesamiento que el sistema debe poseer para cubrir la demanda de esa aplicación. De esta manera, es posible para los diseñadores implementar soluciones orientadas a un dúo perfil-nivel. Especificar el perfil y el nivel de una aplicación implica cumplir con aquellos algoritmos y capacidad de procesamiento que el Estándar dicta para dicho dúo. Por tanto, las soluciones serán compatibles de acuerdo al tipo de aplicación con que sea concebida, sin tener que implementarse todo el estándar. Nuevos perfiles y niveles han ido apareciendo debido a la demanda video en alta definición, transmisión de flujos de videos escalables usados en videoconferencias, vigilancia, así como el auge de la Televisión 3D.

Por ejemplo, para que una implementación pueda asegurar que cumple con H.264/AVC en el perfil Principal¹³ esta debe poseer los algoritmos y herramientas definidos en el Estándar para ese perfil. Algunos de ellos son [4]:

- Predicción Intra-trama: 9 modos de predicción para bloques de luminancia de 4x4 píxeles, 4 modos para bloques de luminancia de 16x16, 4 modos para bloques de cromina.
- Número de tramas de referencia para predicción: Menor e igual que 16.
- Tamaño de bloque de compensación de movimiento: 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4.
- Modo de predicción de tramas B: Utilización de 2 vectores movimiento que se codifican. Modo directo independiente, espacial y temporal.
- Compensación de movimiento: Compensación de 1/2 y 1/4 de pixel.
- Transformación y Cuantificación: Transformada de 4x4 píxeles.
- Codificación por Entropía: CAVLC¹⁴, CABAC¹⁵.
- Filtro de Lazo: Procesamiento a bloques de 4x4 píxeles.

¹² Bitstream, por su origen en idioma inglés.

¹³ Main Profile. Original en idioma inglés. Referido a uno de los perfiles definidos en el Estándar.

¹⁴ CAVLC. Por sus siglas en inglés: Context Adaptive Variable Length Coding.

¹⁵ CABAC. Por sus siglas en inglés: Context Adaptive Binary Arithmetic Coding.

PowerPC 440. Características generales

PowerPC 440 (PPC440) es un microprocesador embebido de 32 bit desarrollado por IBM. PPC440 comparte la arquitectura RISC, donde el limitado número de instrucciones permite simplificar la unidad de decodificación. Su diseño escalar, bajo consumo, flexibilidad y capacidad de procesamiento lo sitúan como uno de los microprocesadores importantes en el Estado del Arte de los núcleos embebidos.

Utiliza la tecnología de bus CoreConnect entre los periféricos que conforman el sistema. Los que demanden altas velocidades se interconectarán a través del bus PLB¹⁶ de 128 bit [6].

PowerPC 440 presenta 32KB de cache para instrucciones y la misma cantidad para datos. Su MMU¹⁷ soporta tamaños múltiples de paginados, varios atributos para protección del almacenamiento, así como opciones para el control de acceso. Posibilita interfaces para controlador de alta velocidad, controlador APU¹⁸, e interfaz para conexión de un co-procesador a la medida del usuario. Soporta 16 tipos de interrupciones clasificadas en críticas o no críticas. Soporta 3 temporizadores con una base de tiempo de 64 bits, así como 7 etapas de pipeline [7].

Existen muchos proveedores con herramientas y soluciones completas de desarrollo para PowerPC440, incluyendo compiladores para C/C++, depuradores, modelos funcionales de bus, ambientes co-simuladores hardware/software, modelos para simulación en VHDL y Verilog entre otros.

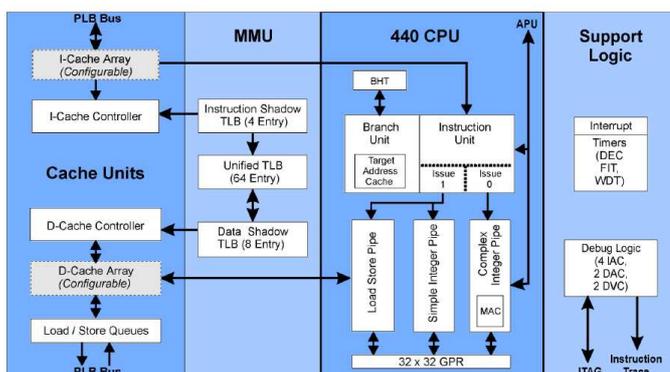


Figura 4. Diagrama en bloques del núcleo PowerPC 440 [7].

H.264/AVC. Referencias. Joint Video Team.

Joint Video Team (JVT) es un grupo experto en codificación de imágenes y video que desarrollaron el estándar H.264/AVC [9]. Como parte del soporte al estándar, fue desarrollado por este grupo versiones de software de referencia. Estas versiones, modeladas en lenguaje C, tiene el propósito de constituir una referencia práctica para el estudio, investigaciones e implementaciones de H.264/AVC.

El software de referencia es una versión libre, y como su nombre indica, es una referencia de la norma. Su uso es aconsejable en las implementaciones tanto en codificación como decodificación, y cumplir los requerimientos del software brinda total compatibilidad con el Estándar.

El presente trabajo consultó la versión JM 18.4 del software de referencia. Esta versión permite ser compilada para ambiente .NET, UNIX, o WINDOWS usando GCC¹⁹.

La versión está organizada para obtener 2 ejecutables: el codificador H.264 y el decodificador H.264. Ambos tienen ficheros fuentes comunes para su compilación, tanto cabeceras (extensión .h) como ficheros de implementación (extensión .c). Sin embargo, es posible la compilación de ambos por separado, obteniéndose su ejecutable correspondiente.

¹⁶ PLB. Por sus siglas en inglés: Processor Local Bus.

¹⁷ MMU. Por sus siglas en inglés: Memory Management Unit.

¹⁸ APU. Por sus siglas en inglés: Auxiliary Processor Unit.

¹⁹ GCC: Por sus siglas en inglés: GNU Compiler Collection.

La solución para el codificador presenta 94 ficheros cabeceras (.h) y 112 ficheros de implementación (.c). La solución del decodificador presenta 39 ficheros cabeceras (.h) y 52 ficheros de implementación (.c). En este trabajo se utilizó Visual Studio 2008 para compilar las fuentes.

El ejecutable del codificador requiere entradas en formato de video descomprimido (extensión .yuv) y brinda a la salida el video codificado en H.264/AVC, con extensión .264. De forma inversa, el ejecutable del decodificador requiere a su entrada un formato de video comprimido, y a su salida brinda el video descomprimido (extensión .yuv).

De esta manera, para la comprobación de los 2 softwares se llevó a cabo el siguiente procedimiento:

1. Codificar un video YUV utilizando el ejecutable codificador.
2. Reproducir el video codificado (formato .264) mediante software visores y players H.264 (VLC, Media Player Classic)
3. Analizar y comprobar la estructura del stream mediante el software analizador de video Elecard Stream Analyzer 2.1.2
4. Decodificar el video utilizando el ejecutable decodificador.
5. Reproducir la salida del decodificador, el video descomprimido en formato YUV con los reproductores correspondientes.

Tanto el video codificado como el video decodificado son propiamente reproducidos por los players de video. Además, mediante el software Elecard es mostrado su correspondencia con el formato H.264/AVC.

```

----- JM 18.4 <PRExt> -----
Input YUV file      : foreman.yuv
Output H.264 bitstream : test.264
Output YUV file     : test_rec.yuv
YUV Format          : YUV 4:2:0
Frames to be encoded : 6
Freq. for encoded bitstream : 30.000
PicInterlace / MbInterlace : 0/0
Transform8x8Mode    : 0
ME Metric for Refinement Level 0 : SAD
ME Metric for Refinement Level 1 : Hadamard SAD
ME Metric for Refinement Level 2 : Hadamard SAD
Mode Decision Metric : Hadamard SAD
Motion Estimation for components : Y
Image format        : 720x480 <720x480>
Error robustness    : 0ff
Search range        : 32
Total number of references : 5
References for P slices : 5
References for B slices <L0, L1> : 5, 1
Sequence type       : I-B-P-B-P <QP: I 28, P 28, B 30>
Entropy coding method : CABAC
Profile/Level IDC    : <77,30>
Motion Estimation Scheme : Fast Full Search
Search range restrictions : none
RD optimized mode decision : used
Data Partitioning Mode : 1 partition
Output File Format    : H.264/AVC Annex B Byte Stream Format
-----

```

Frame	Bit/pic	QP	SnrY	SnrU	SnrU	Time<ms>	MET<ms>	FRM/Fld	Ref
00000<NUB>	176								
00000<IDR>	84872	28	103.517	46.414	48.794	2757	0	FRM	3
00002< P >	20816	28	103.517	46.274	48.643	96108	92073	FRM	2
00001< B >	27704	30	103.517	46.334	48.845	367309	362529	FRM	0
00004< P >	20552	28	103.517	46.201	48.555	189633	185459	FRM	2
00003< B >	27440	30	103.517	46.207	48.517	461429	456538	FRM	0
00005< P >	15176	28	103.517	46.236	48.569	281844	277669	FRM	2

Total Frames:	6								

Figura 5. Codificación de 6 frames de video en H.264/AVC. Software JM 18.4

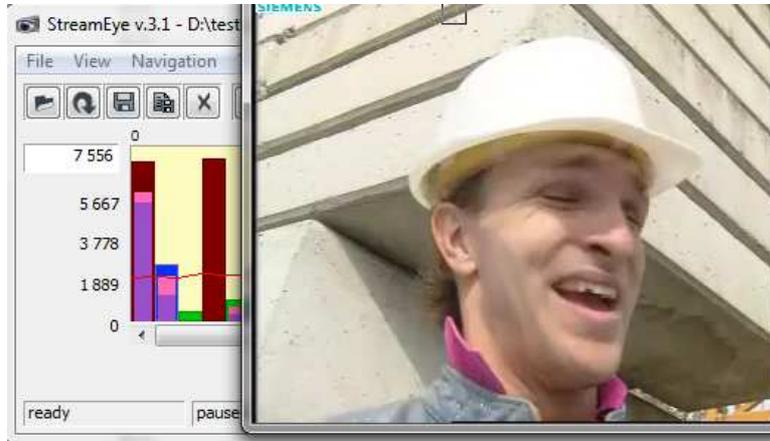


Figura 6. Video codificado a través del software de referencia JM 18.4. Reproductor Eleccard.

A través del software decodificador, se descomprimió el video H.264/AVC obtenido por el software codificador (Figura 7). De esta forma, se recoge a la salida del decodificador la misma secuencia en formato YUV (Figura 8).

```
D:\>ldecod.exe
Setting Default Parameters...
Parsing Configfile decoder.cfg
.....
Parsing error in config file: Parameter Name 'DecodeAllLayers' not recognized.
Parsing error in config file: Parameter Name '=' not recognized.

----- JM 18.4 (PRExt) -----
Input H.264 bitstream           : D:\test.264
Output decoded YUV             : D:\test_dec.yuv
Input reference file           : D:\test_rec.yuv

Profile IDC : 77
Image Format : 720x480 (720x480)
Color Format : 4:2:0 (8:8:8)

POC must = frame# or field# for SNRs to be correct
-----
Frame      POC  Pic#  QP   SnrY   SnrU   SnrU   Y:U:V  Time(ms)
00000<IDR>  0    0    28   0.0000 0.0000 0.0000 4:2:0   179
00002< P >  4    1    28   0.0000 0.0000 0.0000 4:2:0    70
00001< b >  2    2    30   0.0000 0.0000 0.0000 4:2:0   131
00004< P >  8    2    28   0.0000 0.0000 0.0000 4:2:0    60
00003< b >  6    3    30   0.0000 0.0000 0.0000 4:2:0   130
00005< P > 10    3    28   0.0000 0.0000 0.0000 4:2:0    59
-----
Average SNR all frames
SNR Y<dB> : 0.00
SNR U<dB> : 0.00
SNR V<dB> : 0.00
Total decoding time : 0.631 sec (9.509 fps)[6 frm/631 ms]

Exit JM 18 (PRExt) decoder, ver 18.4
Output status file           : log.dec
6 frames are decoded.
```

Figura 7. Decodificación de 6 frames de video en H.264/AVC. Software JM 18.4

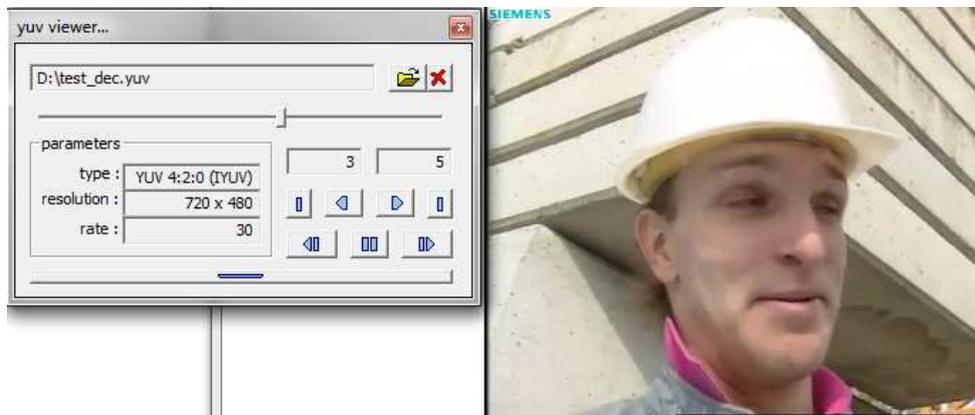


Figura 8. Reproducción del video decodificado mediante el software JM 18.4. Formato YUV.

El procedimiento llevado a cabo demuestra la factibilidad de utilizar el software JM 18.4 como referencia de estudio, guía e implementación de soluciones referentes al códec H.264. Además de la prueba visual, se ha corroborado la secuencia del stream a través del analizador Elecard Stream Analyzer 2.1.2. Este muestra las sintaxis y el formato correcto de la trama codificada.

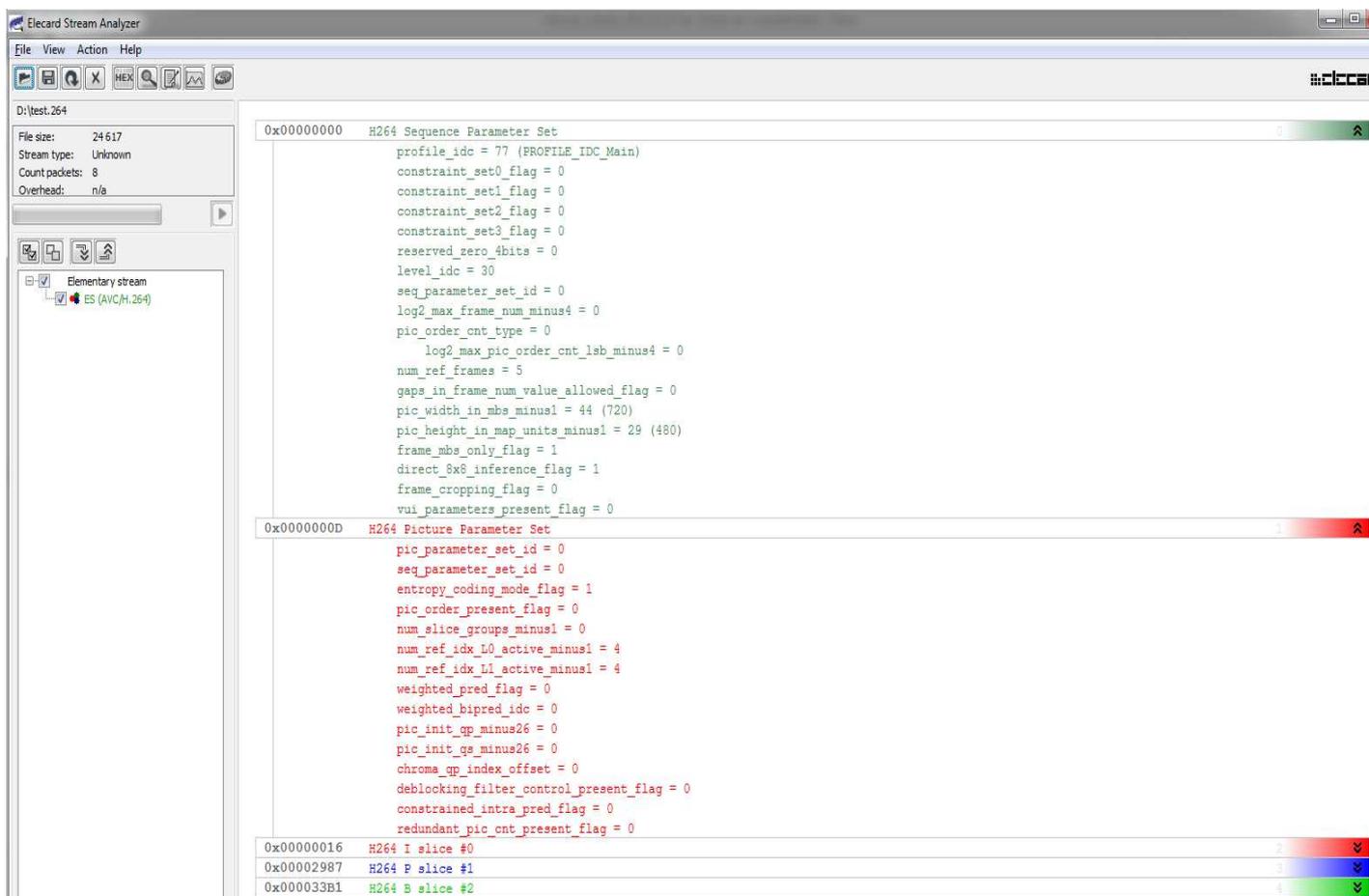


Figura 9. Analizador Elecard 2.1.2 muestra las sintaxis y formato del video codificado por JM 18.4.

Modelo Decodificador H.264/AVC sobre FPGA.

Una de las múltiples ventajas de los sistemas embebidos sobre FPGA es la capacidad de análisis y cómputo a través de softwares en alto nivel que permiten establecer modelos de sistemas en tiempos relativamente breves. Para validar la posibilidad de implementar una solución de un decodificador conforme a la norma H.264 se utilizó un sistema con microprocesador empotrado en un FPGA Virtex 5, de Xilinx. Para ello se utilizó la tarjeta de Desarrollo ML507, que contiene dentro del FPGA un núcleo PowerPC 440.

Para el sistema empotrado se utilizó los siguientes módulos:

- PowerPC440 v1.01a. Microprocesador del sistema y elemento principal.
- Módulo para el controlador de la memoria FLASH de la tarjeta ML507.
- Módulos para atención del puerto RS232. Se utiliza para reportar actividad del sistema.
- Modulo para memoria DDR-2 de la tarjeta ML507. En la memoria se trabajará con datos dinámicos.
- Módulo para el controlador de Compact Flash. En esta memoria se almacenarán los videos de entrada al sistema y la salida decodificada.

De esta manera el sistema en el FPGA queda listo para soportar un software en alto nivel que sirva como modelo para una solución de un decodificador H.264/AVC. Para esto se trabajó con los ficheros fuentes en lenguaje C, JM 18.4. Como parte del proceso, se llegó a una profundidad del funcionamiento del software para poder modificar su flujo, algunas estructuras, eliminar elementos redundantes para el objetivo trazado. A continuación se ejemplifican algunas modificaciones:

- Extracción de ficheros fuentes necesarios para compilar el software para el decodificador H.264/AVC.
- Modificación de cada función para introducir y almacenar ficheros al sistema. Se sustituyeron por nuevas funciones para el sistema de ficheros y algunas API para el controlador de la Compact Flash en el sistema embebido.
- Modificación de las funciones de configuración del sistema. En su lugar se crearon funciones con configuraciones fijas, las cuales para poder ser modificadas será necesario editar los ficheros .h
- Dejar sin acción las opciones de MVC²⁰. El código analiza esas opciones, no válidas en el stream utilizado en este trabajo.
- Modificar algunos tipos de datos de variables tipo char, por variables enteras. Estas variables tomaban valores distintos provocados por la diferencia de las plataformas x86 y PowerPC. Esto provocaba errores en el proceso de decodificación.
- Reducción del número de Sintaxis para la decodificación por Entropía. De esta forma se concentra solamente en aquellas sintaxis que el stream codificado solicitaría. No obstante, a pesar de que el fichero codificado utiliza solamente CABAC, entre las opciones presentes en el software se dejaron tanto las herramientas de CABAC como CAVLC.
- Introducir comentarios en funciones y ficheros para señalar el flujo del software.
- Añadir nuevos ficheros cabeceras (.h) y ficheros de implementación (.c) para complementar las librerías existentes con las nuevas funciones utilizadas por la plataforma empotrada en el FPGA.

Se obtuvo un conjunto de ficheros fuentes con la capacidad de ser soportado por la plataforma PowerPC embebida en el FPGA. Se compiló el sistema y se descargaron los ficheros a la tarjeta. Los pasos para la comprobación del modelo dentro del FPGA fueron los siguientes:

- Almacenar el fichero codificado en H.264/AVC en la Compact Flash de la tarjeta ML507. El fichero es el mismo utilizado como entrada del decodificador compilado en Visual Studio. De esta manera se compara el procesamiento de ambos software decodificadores (software en la PC y el sistema en FPGA) a partir de la misma entrada de video.
- Correr el sistema, verificando su funcionamiento al decodificar el fichero comprimido en la Compact Flash y brindando su salida en un fichero YUV dentro de la propia Compact Flash.
- Retirar la Compact Flash de la tarjeta e insertarla en el puerto de la PC. Verificar la estructura del fichero decodificado por el FPGA y su similitud con aquel decodificado en la PC.
- Analizar el stream a través del software Elecard Stream Analyzer 2.1.2. Reproducir el fichero y verificar visualmente la validez del fichero descomprimido, así como el correcto funcionamiento del modelo decodificador dentro del FPGA.

El fichero obtenido, tiene igual tamaño e idéntica estructura que el fichero obtenido por el software decodificador compilado en Visual Studio (Figura 7). El fichero fue verificado por varios reproductores de formato YUV, visualizando exactamente los mismos frames que los mostrados en la figura 8.

CONCLUSIONES

- El sistema embebido sobre FPGA utilizando microprocesadores embebidos como PowerPC, puede ser utilizado para el soporte de complejos softwares de alto nivel. El resultado obtenido dentro de un FPGA en este trabajo es el mismo que el que se obtiene dentro de una PC convencional.
- Crear un modelo de referencia de un decodificador H.264/AVC es fundamental para el proceso de implementación de soluciones afines. Constituye un elemento importante para planificar futuras soluciones.
- El modelo dentro del sistema embebido independiza la solución del software en la PC y lo ubica en un ambiente indicado para futuras modificaciones y bondades que brinda la tecnología FPGA.

²⁰ MVC: Por sus siglas en inglés: MultiView Video Coding.

- El modelo no está apto para soluciones en tiempo real. Su propósito fundamental es constituir la base teórica y práctica para optimizaciones, modificaciones, relación e intercambio de módulos software por otros de tipo hardware que permitan mayor velocidad de procesamiento. De esta forma, de manera segura y precisa, se avanzará hacia una solución en tiempo real totalmente compatible con la norma H.264/AVC.

REFERENCIAS

1. HANZO, L. CHERRIMAN, P. STREIT.J.: *Video Compression and Communications. Second Edition*. Ed. John Wiley & Sons, Ltd. West Sussex, Gran Bretaña 2007.
2. WANG, YAO: "Video Processing & Communications". Polytechnic Institute of NYU. Brookling, NY11201.
3. GAO, WEN. READERS, CLIFF. WU, FENG. HE, YUN. YU, LU. LU, HANQING. YANG, SHIQIANG. HUANG, TIEJUN. PAN, XINDE.: "AVS- The Chinese Next-Generation Video Coding Standard".
4. HUANG, TIEJUN. MA, SIEWI.: "AVS-The Video Coding Standard for DTV and Other Applications" *China-Cuba Primer Foro Internacional de Televisión Digital*. La Habana, Cuba, marzo 20, 2013.
5. RICHARDSON, IAN E.: *The H.264 Advanced Video Compression Standard. Second Edition*.Ed. John Wiley & Sons, Ltd. Wiltshire, Gran Bretaña, 2010.
6. XILINX: "Virtex-5 FPGA Embedded Processor Block with PowerPC 440 Processor (v1.01a)". Julio 2011.
7. IBM MICROELECTRONICS DIVISION: "The PowerPC 440 Core. A High-performance, superscalar processor core for embedded application". 1999.
8. XILINX: "Virtex-5 FPGA Embedded Processor Block with PowerPC 440 Processor (v1.01a)". Julio 2011.
9. JOINT VIDEO TEAM (JVT) OF ISO/IEC MPEG & ITU-T VCEG: "H.264/14496-10 AVC Reference Software Manual". Junio 2009.

AUTORES

Orlando Landrove Gámez, Ingeniero en Telecomunicaciones y Electrónica. Graduado en 2007. Aspirante a Investigador y Profesor Instructor en la CUJAE, Facultad de Eléctrica. Labora en **LACETEL**, Instituto de Investigación y Desarrollo de Telecomunicaciones. Actualmente realiza investigaciones acerca microprocesadores empotrados en FPGA y Codecs de Video.

Correo Electrónico: landrove@lacetel.cu