

DISEÑO DEL BLOQUE DE PROCESAMIENTO DE UN DECODIFICADOR DE VIDEO MPEG-2

MPEG-2 VIDEO DECODER PROCESSING BLOCK DESIGN

Ing. Rufino R. Cabrera Álvarez

Ingeniero en Telecomunicaciones y Electrónica, **LACETEL**, rufino@lacetel.cu, 6832814

Proceso de despliegue de la TVD en Cuba

La Habana, Noviembre 2013

RESUMEN

La migración hacia la televisión digital es un proceso que se ha estado llevando a cabo en muchos países desarrollados y Cuba no se encuentra al margen de esta realidad. El presente trabajo constituye el inicio de una etapa dentro de la investigación y desarrollo que lleva a cabo **LACETEL** para impulsar la Televisión Digital en Cuba, que tendrá como punto importante el desarrollo de un módulo de propiedad intelectual (módulo IP) para un decodificador de video MPEG-2 estándar desarrollado por el grupo **MPEG (Moving Picture Experts Group)**. El diseño fue dividido en bloques funcionales y realizado sobre un **PIC24 (Peripheral Interface Controller)** de Microchip® en el lenguaje de alto nivel **ANSI C (American National Standard Institute)** haciendo uso de la herramienta MPLAB® IDE. Además se presentan los resultados de la simulación en MPLAB® IDE los cuales muestran el comportamiento esperado bloque a bloque.

Palabras Clave: Televisión Digital, MPEG-2, Decodificador, PIC24, ANCI C.

ABSTRACT

*The migration to digital television is a process that has been taking place in many developed countries and Cuba is not outside of this reality. This paper is the beginning of a stage in the Digital Television research and development lead by **LACETEL** and a MPEG-2 video decoder IP module design is one of the important parts in this stage. The design was split in functional blocks and performed on a PIC24 from Microchip ® using ANSI C language and using MPLAB ® IDE tool and. Also presents the results of simulation in MPLAB ® IDE which show the expected behavior block by block.*

KeyWords: Digital Television, MPEG-2, Decoder, PIC24, ANCI C.

TABLA DE CONTENIDO

1.	INTRODUCCIÓN	1
2.	DECODIFICADOR DE VIDEO MPEG-2	1
2.1	La familia PIC24FJ128GA010. Plataforma del diseño.	3
2.1.1	<i>Temporización del PIC24</i>	3
2.1.2	<i>Comunicación serie en el PIC24</i>	4
2.2	Bloques funcionales del diseño.....	4
2.2.1	Bloque de “Recepción del Bitstream”	5
2.2.2	Bloque de “Cálculo y Procesamiento de los Bloques”	11
3.	CONCLUSIONES	15
4.	REFERENCIAS BIBLIOGRÁFICAS	15
5.	ANEXOS	16
	Anexo 1. Contenido de la trama MPEG-2.	16

1. INTRODUCCIÓN

Durante la digitalización de la televisión se incorporan etapas como la compresión, que permite un mejor aprovechamiento del espectro radioeléctrico y con ello la evolución respecto a la actual televisión analógica convencional.

Existen múltiples procedimientos para la compresión de las fuentes de video y audio empleados en televisión digital, entre ellos destacan: MPEG-2, H.264/AVC y **AVS** (*Audio Video coding Standard*) los cuales superan las deficiencias de sus antecesores para lograr mejores tasas de compresión. MPEG-2 cubre los aspectos y metodologías utilizados en la compresión de las señales de audio y video, así como los procedimientos de multiplexación y sincronismo de estas señales en tramas de programa y de transporte, aunque al ser el primero de estos es el menos eficiente en la compresión pero resulta ser el más sencillo, por lo que nos permitirá alcanzar el *know how* para el futuro desarrollo de H.264/AVC y AVS que alcanzan mayores tasas de compresión. Como en todo proyecto el diseño se dividirá en bloques funcionales para simplificar el trabajo.

Por lo tanto el objetivo general es diseñar el bloque de procesamiento que conforma un decodificador de video MPEG-2 y los objetivos específicos son:

- Fundamentar los referentes teóricos que regulan la decodificación MPEG-2 con las especificaciones descritas en la familia de estándares ISO/IEC 13818¹.
- Seleccionar una interfaz donde implementar el diseño.
- Validar el proyecto según los resultados de las simulaciones.
- Dividir el diseño en bloques para simplificar el trabajo.

2. DECODIFICADOR DE VIDEO MPEG-2

La compresión de video se basa en aprovechar las propiedades de las imágenes y las características perceptuales del sistema visual humano, para ello se siguen los siguientes pasos [1]:

- Eliminación de información irrelevante.

¹ Estándar definido por el **ISO/IEC** (International Standardization Organization/ International Electrotechnical Commission) para el tratamiento de la trama MPEG-2, que incluye 11 partes entre las que se encuentra: Sistema, Video, Audio, etc.

- Redundancia espacial.
- Redundancia temporal.
- Compactación de la energía.

MPEG-2 resume estos pasos de forma muy sencilla:

- Eliminación del blanking digital.
- División en bloques, Transformada Discreta del Coseno, Cuantificación y Codificación entrópica.
- Compensación de movimiento.

Para el diseño de los bloques funcionales del proyecto se cuenta con un PIC24, un PIC33, un CPLD y un FPGA como materiales además de los simuladores MPLAB® IDE, XILINX® ISE y PROTEUS® 7.7 sp2.

Dejando a un lado los simuladores se analizan el resto de los materiales para seleccionar el más idóneo. De ellos se seleccionaron los PICs ya que son programados en lenguaje de alto nivel, por lo que resulta más sencillo de realizar los cálculos que se requieren para la transformada, como son las sumatorias y los cosenos; algo que en **HDL (Hardware Description Language)** resulta complejo. Además el PIC presenta periféricos como los Temporizadores y el Puerto Serie que en un FPGA hay que crearlos. Entre los PICs, fue seleccionado el PIC24 que soporta los requerimientos necesarios para el diseño haciendo uso óptimo de los recursos.

De acuerdo con las tareas de investigación que llevaron al desarrollo del contenido de esta tesis, fueron aplicados los siguientes métodos de investigación:

- **Métodos teóricos:**
 - Método histórico-lógico: Se manifiesta en la revisión bibliográfica de los aspectos teóricos que cubre el estándar ISO/IEC 13818. A través de él, se estudió la evolución desde sus inicios, de los sistemas, formatos y normas de la televisión analógica, hasta llegar al mundo digital y dentro de él, a la compresión de las fuentes de audio y video MPEG-2, que es el objetivo principal de este trabajo.
 - Método de análisis y síntesis: Utilizado para analizar las tablas y gráficas que apoyan esta investigación, y tener elementos que establezcan conclusiones y sugieran recomendaciones a este proyecto.

- Método sistémico: Debido a la acumulación de las experiencias teóricas, en los años que ha evolucionado la TV, se estudiaron la familia de estándares ISO/IEC 13818, la guía de usuario del PIC24, el manual de instrucciones de ANSI C para desarrollar el diseño proyectado.

- **Métodos empíricos:**

- Método de observación: Permiten estudiar los fenómenos observables directamente y contribuyen a confirmar las teorías des-arrolladas por los métodos anteriores dando cumplimiento a los objetivos de este trabajo. Por ejemplo las simulaciones que se realizan de los bloques funcionales.

- Método experimental: Debido a la acumulación de las experiencias teóricas y prácticas se prosigue al diseño del dispositivo, simulando cada bloque para detectar las fallas de diseño, corregirlas y formular una propuesta más adecuada.

- Método de medición: Validar el diseño final comprobando que a la salida del decodificador se obtenga el resultado adecuado para una señal de entrada ofrecida.

2.1 La familia PIC24FJ128GA010. Plataforma del diseño.

El La modificación de 16 bits de la arquitectura de Hardware en todos los dispositivos PIC24 ofrece una amplia gama de mejoras, tales como [2]:

- Bus de datos de 16 bits y de direcciones de 24 bits.
- Seguridad en el acceso por la separación entre la memoria de datos y la de programa con la habilidad de transferir información entre datos y espacios en memoria.
- Direccionado lineal de hasta 8 Mbyte (espacio en memoria) y 64 Kbyte (datos).
- Opera con arreglos de 16 elementos incorporando soporte de pila de software.
- Posee en el hardware un multiplicador de 17x17 para el cálculo de enteros.
- Eficiencia del código: su juego reducido de instrucciones que soporta múltiples modos de direccionamiento y está optimizado para lenguajes de alto nivel como ANSI "C" permitiendo una gran compactación de los programas.
- Rapidez de ejecución: el desempeño operativo de hasta 16 millones de instrucciones por segundo (MIPS) a una frecuencia de trabajo de 32 MHz.

2.1.1 Temporización del PIC24

La familia PIC24FJ128GA de *Microship* incorpora cinco temporizadores internos que pueden ser atendidos por interrupción. El módulo del temporizador 1 presenta un contador de 16 bits que puede ser utilizado para el reloj en tiempo real o como temporizador /contador en intervalos deseados. Este temporizador puede operar en tres modos [2]:

- Temporizador de 16 bits.
- Contador sincrónico de 16 bits.
- Contador asincrónico de 16 bits.

Además soporta las siguientes características [2]:

- Ajustes de pre-escale seleccionables.
- Operaciones mientras el CPU está en modo inactivo o dormido.
- Interrupciones por frente de bajada de una señal externa.

2.1.2 Comunicación serie en el PIC24

La familia PIC24FJ128GA incorpora dos interfaces asincrónicas **UART** (**U**niversal **A**synchronous **R**eceiver **T**ransmitter), dos interfaces sincrónicas **SPI** (**S**erial **P**eripheral **I**nterface) y dos módulos independientes I²C, que soportan los modos de funcionamiento tanto maestro como esclavo. Según las características mostradas en la Tabla I se seleccionó uno de los SPI ya que alcanza las mayores tasas de bitrates lo cual aprovecharemos para que el diseño del primer bloque del decodificador de video cumpla con la definición estándar. [2]

Tabla I: Velocidades de los periféricos de comunicación serie del PIC24FJ128GA.

Periférico		Velocidad
Asincrónica	UART	500 kbps
Sincrónica	I ² C	1 Mbps
	SPI	10 Mbps

2.2 Bloques funcionales del diseño

El diseño que a continuación se describe está sustentado por el uso del lenguaje de

alto nivel C sobre el PIC24FJ128GA010 de la familia PIC24 de Microchip. La Figura 1 representa el esquema general del proyecto realizado. En ella se pueden observar las entradas y salidas globales del diseño.

DATA_MPEG representa la entrada de datos serie o flujo de bits MPEG-2, CLK es la señal de reloj perteneciente al flujo; se asumirá que ambas son entregadas por un dispositivo previo al decodificador encargado del sincronismo. La señal de reloj representa la frecuencia a la que arriban los datos MPEG-2. Las salidas $Y[v][u]$, $Cr[v][u]$ y $Cb[v][u]$ son las tres matrices: luminancia y croma que forman una imagen. La señal PULSO indica al siguiente dispositivo que las imágenes están listas para pro-cesarlas.



Figura. 1 Esquema del diseño con sus entradas y salidas

El esquema general está dividido en varias etapas como se mencionó antes. La Figura 2 muestra las etapas en que se ha fragmentado el diseño.

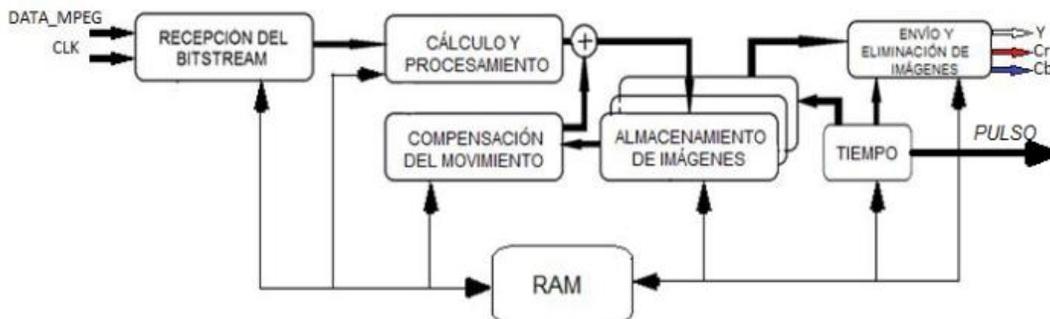


Figura. 2 Diagrama en bloques del decodificador de MPEG-2

En este trabajo se recogen detalladamente el bloque de Procesamiento que comprende los bloques de Recepción del Bitstream y Cálculo y Procesamiento de la señal.

2.2.1 Bloque de "Recepción del Bitstream"

Un codificador MPEG-2 entrega un flujo de bits a su salida con una organización de

alto nivel como puede verse en el diagrama de la Figura 3. Todas las tramas comienzan con una cabecera de secuencia, que es idéntica a la que produciría un codificador MPEG-1 para garantizar su compatibilidad con MPEG-2 y pueden verse en anexo 1. [3]

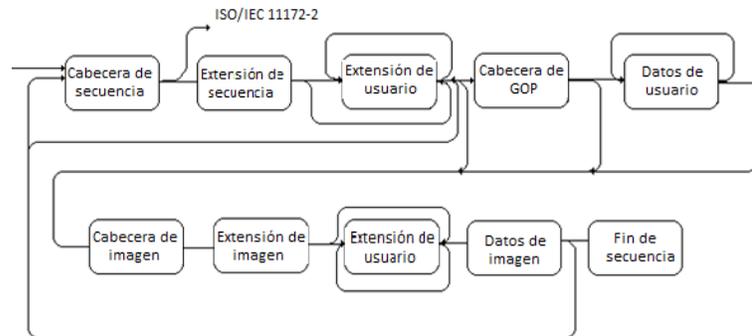


Figura. 1 Organización de alto nivel del bitstream

Para cada imagen el codificador incluye una cabecera, una extensión de codificación y los datos de la imagen (vector de movimiento, coeficientes cuantificados, etcétera). Después de la primera imagen se suceden las demás secuencialmente (el codificador puede insertar o no nuevas cabeceras de secuencia, cabecera de grupo de imágenes y datos de usuario) hasta el fin de la secuencia, que se señala con un código específico.

La cabecera de secuencia contiene información como: el tamaño de las imágenes, dado por su componente horizontal y vertical; la relación de aspecto requerida para la visualización; la velocidad de imagen, que indica la frecuencia a la cual deben mostrarse las mismas. Entre los parámetros que transporta la cabecera de macrobloque destacan: el código de escala de cuantificador, que permite decodificar el factor de escala del mismo y se utiliza hasta que se defina un nuevo valor; el código de movimiento, utilizado en la decodificación de la componente horizontal y vertical de los vectores de movimiento.

Los datos de usuario pueden aparecer después de la extensión de secuencia, tras la cabecera de grupo de imágenes y/o tras la extensión de imágenes. Si aparecen, no afectan al proceso de decodificación y pueden ser ignorados por el decodificador. La cabecera de grupo de imágenes no es obligatoria, es una forma de agrupar conjuntos de imágenes en un nivel jerárquico superior, lo cual es útil sobre todo para la edición de video almacenado. Si aparece esta cabecera, la primera imagen del grupo debe ser siempre tipo I. [3,4]

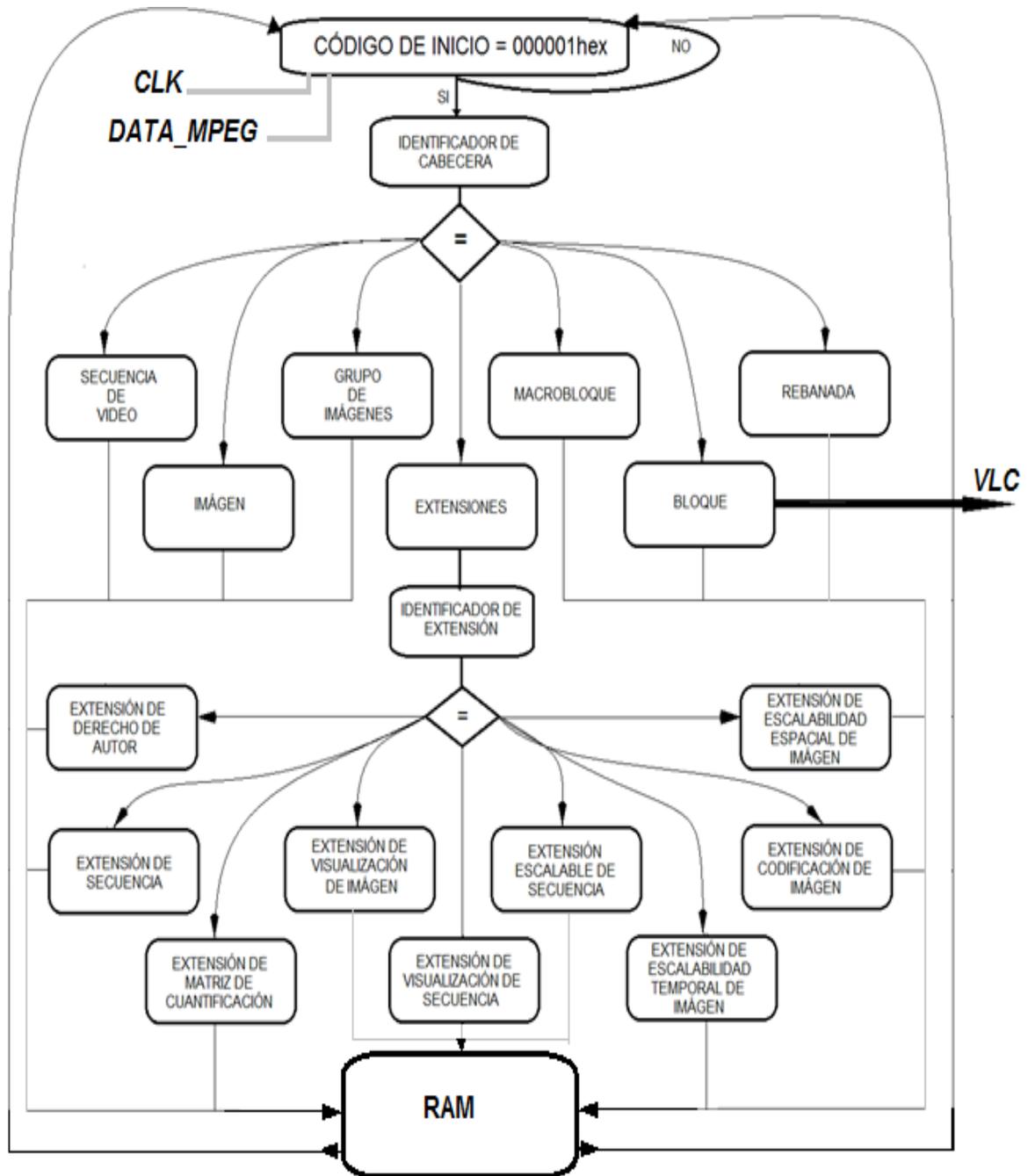


Figura. 2 Diagrama en bloques de la etapa de recepción del bit stream

Para realizar el diseño de este bloque se utiliza la interfaz serie sincrónica (SPI) que es la más adecuada como se puede apreciar en la Tabla I en comparación con el resto de las interfaces que el PIC24 ofrece para operar con video de definición estándar², lo

² En MPEG-2 se plantea que la definición estándar debe cumplir con nivel principal at perfil principal (MP@ML).

que implica una relación de aspecto máxima de 720x480 *píxeles* y una razón binaria máxima de 4Mbps, es válido aclarar que este último modo es el más utilizado para las transmisiones de televisión digital. Las acciones que se realizan para la extracción de los múltiples campos transportados por el flujo se muestran en el diagrama en bloques de la Figura 4 y fue programada en lenguaje C.

Con motivo de la verificación de esta etapa, se introdujo una secuencia de bits que posee los inicios de cabeceras correspondientes al flujo MPEG-2 y mediante la herramienta MPLAB Sim de MPLAB® se puede constatar, como se muestra en la Figura 5 resaltado en rojo, que la función *sequence_header*, adquiere los valores correspondientes a: *horizontal_size_value* y *bit_rate_value*, entre otros. También puede apreciarse, resaltado en azul, que el parámetro *next_bit* almacena el valor del último código de inicio recibido, B3h en este caso, indicando que los campos que se deben adquirir son los mencionados anteriormente.

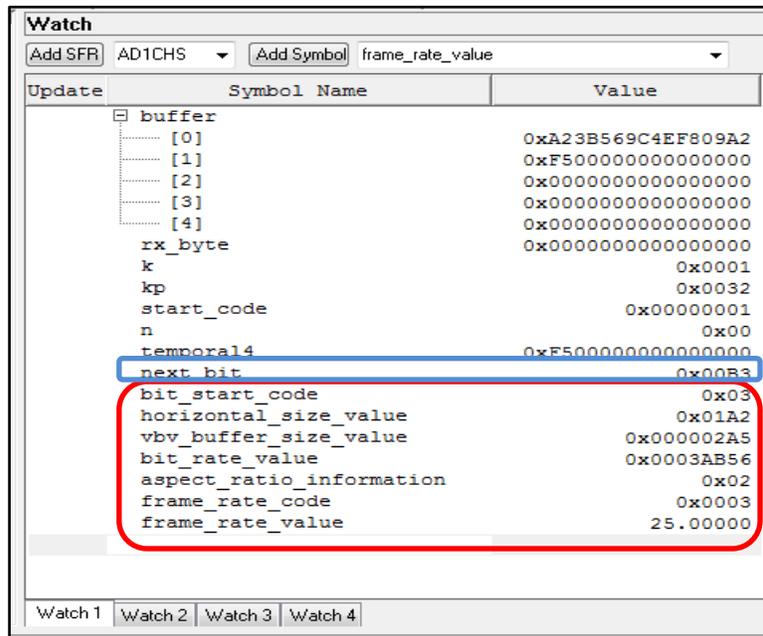


Figura. 3 Ejecución de la función *sequence_header* en MPLAB Sim

En la Figura 6 se muestra, resaltado en azul, la detección de la cabecera de grupo de imágenes (*group_of_pictures_header*) ya que *next_bit* toma el valor B8h y se procede a la extracción de los campos: *time_code*, *close_gopybroken_link*, que transporta la misma, los cuales se resaltan en rojo en dicha figura.

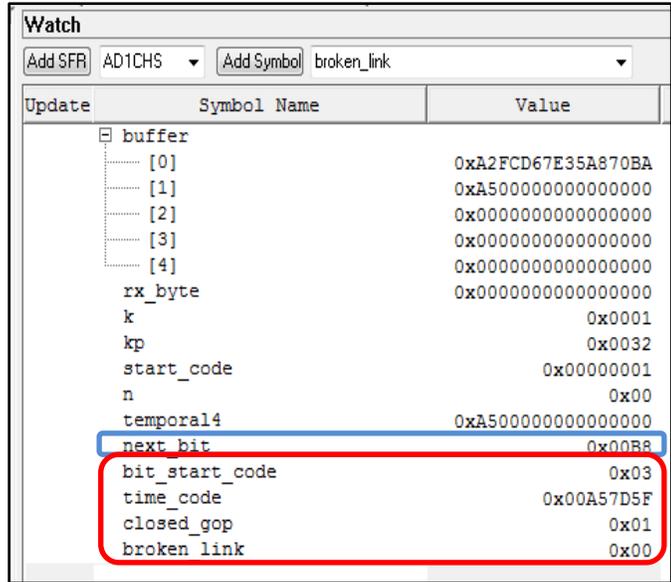


Figura. 6 Ejecución de la función *group_of_pictures_header* en MPLAB Sim

En la Figura 7 se muestra, resaltado en azul, la detección de la cabecera de imagen (*picture_header*) ya que *next_bit* toma el valor 00h y se procede a la extracción de los campos: *temporal_reference*, *picture_coding_type*, *vbv_delay*, entre otros que dicha cabecera posee, los cuales se resaltan en rojo en dicha figura.

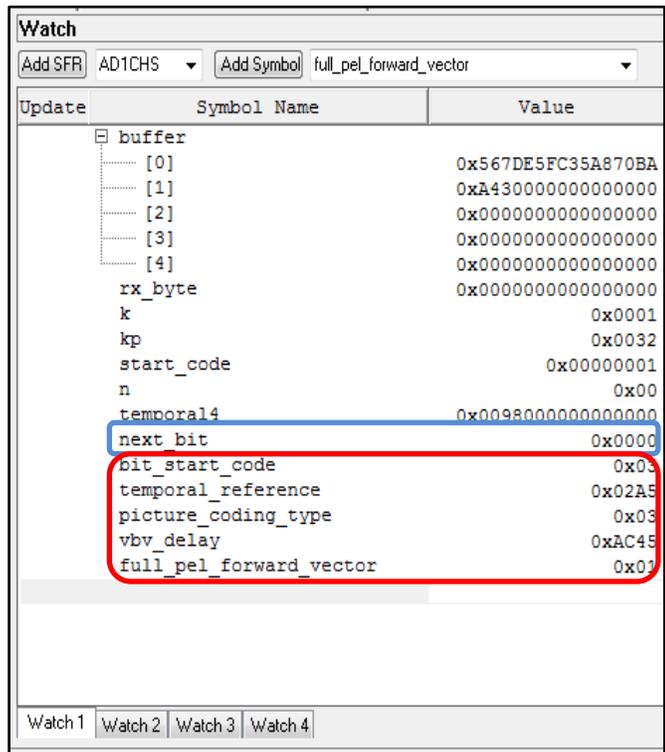


Figura. 7 Ejecución de la función *picture_header* en MPLAB Sim

En la Figura 8 se muestra, resaltado en azul, la detección del inicio de la rebanada (*slice*) ya que *next_bit* toma valor 01h y se procede a la extracción de los campos: *priority_breakpoints* y *reserved_bits*, entre otros que transporta la misma, los cuales se resaltan en rojo en dicha figura.

Update	Symbol Name	Value
	buffer	
	[0]	0x6954035DA54FE567
	[1]	0xF430000000000000
	[2]	0x0000000000000000
	[3]	0x0000000000000000
	[4]	0x0000000000000000
	rx_byte	0x0000000000000000
	k	0x0001
	kp	0x0032
	start_code	0x00000001
	n	0x00
	temporal4	0x0098000000000000
	next_bit	0x0001
	bit_start_code	0x03
	priority_breakpoint	0x46
	reserved_bits	0x3A
	intra_slice	0x01

Figura. 8 Ejecución de la función *slice* en MPLAB Sim

En la Figura 9 se muestra, resaltado en azul, la detección de cada macrobloque (*macroblock*) que transporta el slice porque *macroblock_escape* tiene valor 00000001000b y la extracción de los campos que posee cada uno como son: *mb_width* y *mb_column*, los cuales se resaltan en rojo en dicha figura. Estas especificaciones que presenta la trama pueden verse con mayor precisión en la ISO/IEC 13818-2 donde se presentan tablas que ayudan al esclarecimiento de dicho artículo.

Update	Symbol Name	Value
	buffer	
	[0]	0x56733DA54E7DC908
	[1]	0x0000000000000000
	[2]	0x0000000000000000
	[3]	0x0000000000000000
	[4]	0x0000000000000000
	rx_byte	0x0000000000000000
	k	0x0003
	kp	0x0040
	start_code	0x00000001
	n	0x00
	temporal4	0x0000000000000000
	next_bit	0x0001
	bit_start_code	0x03
	macroblock_escape	0x0008
	mb_column	0x0085
	mb_width	0x03A7

Figura. 9 Ejecución de la función *macroblock* en MPLAB Sim

2.2.2 Bloque de “Cálculo y Procesamiento de los Bloques”

La información de la trama MPEG-2 que sale del codificador de video es procesada para lograr altas tasas de compresión con este estándar. Para ello se divide la imagen en bloques de 8x8 píxeles que son transformados por las ecuaciones (1) y (2) correspondiente a la transformada discreta del coseno para suavizar los contornos de los macrobloques, lo cual es muy útil para no afectar la calidad de la imagen.

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] * \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (1)$$

Donde:

F(u,v) es el bloque de 8x8 transformado

f(x,y) es el bloque de 8x8 de la imagen

N es el orden de la matriz en este caso 8

u, v, x, y = 0, 1, 2, N-1.

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{para } u, v = 0 \\ 1 & \text{en otros casos} \end{cases} \quad (2)$$

Para diseñar el bloque que aplique transformada inversa a este bloque 8x8 píxeles del dominio de la frecuencia al dominio del espacio, es decir, cada F[v][u] es transformado a f[y][x] y se logra a través de la ecuación (3). Los valores obtenidos son saturados según la ecuación (4) para limitarlos al rango de [-256:255].

$$f[y][x] = \frac{2}{N} c(u)d(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F[v][u] * \cos \left[\frac{(2x+1)u\pi}{2N} \right] * \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (3)$$

$$f[y][x] = \begin{cases} -256, & \text{si } f[y][x] \leq -256 \\ f[y][x], & \text{si } -256 < f[y][x] < 255 \\ 255, & \text{si } f[y][x] \geq 255 \end{cases} \quad (4)$$

Los macrobloques pueden ser omitidos cuando sus coeficientes no aportan información, lo que significa que los valores de f[y][x] son cero y no se requiere procesarlos.

MPEG-2 luego de transformar los bloques los somete a un proceso de

cuantificación que consiste en lograr una tasa de bits constante, dividiendo entre un factor conocido como paso de cuantificación (a mayor paso menor número de niveles) lo que hace más notorio las discontinuidades en los contornos de los macrobloques y las imágenes se perciben muy deterioradas. Para disminuir estos errores MPEG-2 define este proceso como se muestran en las ecuaciones (5) y (6).

$$QF'(u, v) = \frac{16F''(u, v)}{(quantiser_scale * W(u, v))} \quad (5)$$

$$QF(u, v) = \begin{cases} QF'(u, v) - \frac{1}{2}; & \text{para } QF'(u, v) > 0 \\ QF'(u, v); & \text{para } QF'(u, v) = 0 \\ QF'(u, v) + \frac{1}{2}; & \text{para } QF'(u, v) < 0 \end{cases} \quad (6)$$

Donde:

$F''(u, v)$: resultado de una saturación en la etapa de cuantificación

$quantiser_scale$: factor de escala.

$W(u, v)$: matriz de cuantificación que es enviada por el receptor, que no siempre es la misma y que se recibe en el campo de extensión de matriz de cuantificación ($quant_matrix_extension$), en caso de no recibirla se utilizan otras dadas por defecto.

Para diseñar el bloque de decuantificación se ha programado en C las siguientes ecuaciones según expresa el estándar ISO/IEC 13818-2. [3]

$$F''[v][u] = \frac{(2 * QF[v][u] + k) * W[v][u] * quantiser_scale}{32} \quad (7)$$

$$k = \begin{cases} 0, & \text{para bloques intra} \\ \text{signo}(F''[v][u]), & \text{para bloques no intra} \end{cases} \quad (8)$$

$$F'[v][u] = \begin{cases} 2047, & \text{para } F''[v][u] > 2047 \\ F''[v][u], & \text{para } -2048 \leq F''[v][u] \leq 2047 \\ -2048, & \text{para } F''[v][u] < -2048 \end{cases} \quad (9)$$

$$sum = \sum_{v=0}^{v \leq 8} \sum_{u=0}^{u \leq 8} F'[v][u] \quad (10)$$

$$F[v][u] = F'[v][u] \text{ para todo } u, v \neq 7 \quad (11)$$

$$F[7][7] = \begin{cases} F'[7][7], & \text{para todo valor impar de sum} \\ \left. \begin{array}{l} F'[7][7] - 1, \text{ si } F'[7][7] \text{ es impar} \\ F'[7][7] + 1, \text{ si } F'[7][7] \text{ es par} \end{array} \right\} \text{ para todo valor par de sum} \end{cases} \quad (12)$$

Para las componentes de DC en las imágenes *intra* se cuantifican diferente a como se ha explicado con anterioridad. El valor de $F[0][0]$ se obtiene según la ecuación (13).

$$F[0][0] = intra_{dc_{mult}} * QF[0][0] \quad (13)$$

Donde:

intra_dc_mult: se extrae de la Tabla II.

Tabla II: Relación entre *intra_dc_precision* e *intra_dc_mult*

Intra_dc_precision	Bits resolución	Intra_dc_mult
0	8	8
1	9	4
2	10	2
3	11	1

El estándar MPEG-2 define dos posibles formas para la transmisión de los coeficientes de los bloques, una es la exploración en zigzag (*zigzag scan*) y la otra es la exploración alternativa (*alternative scan*) como se muestra en la Figura 10. Estas dos exploraciones pueden ser seleccionadas con un bit denominado “exploración alternativa” presente en la extensión de cabecera de imagen. [3]

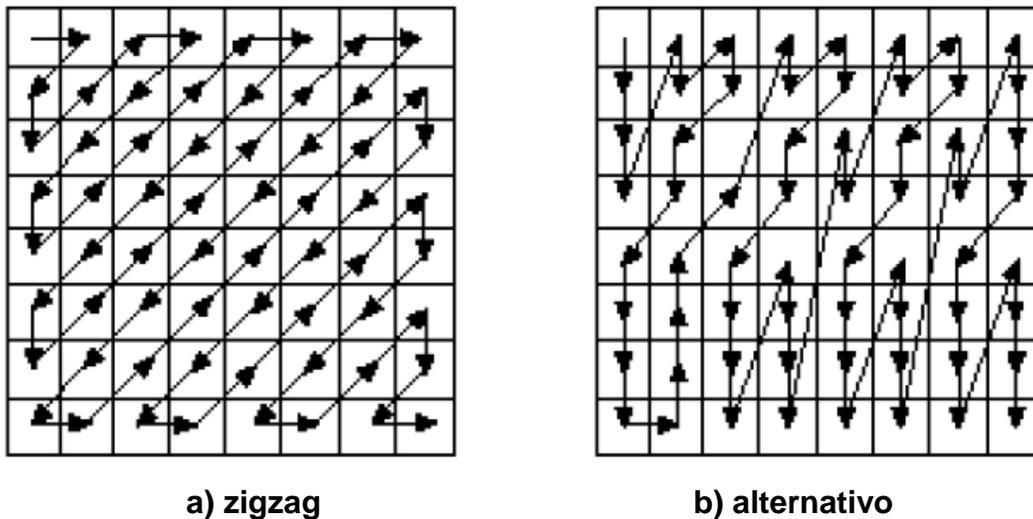


Figura. 10 Exploración definida por MPEG

Para verificar el correcto diseño de este bloque funcional se introdujeron varios arreglos como por ejemplo el que es mostrado a continuación correspondiente a los datos recibidos y debería dar como resultado la matriz de la Figura 11 correspondiente a la matriz trasformada inversamente.

Arreglo:

00001100000000000100101000011000010010110000000100110010011100001110
001110

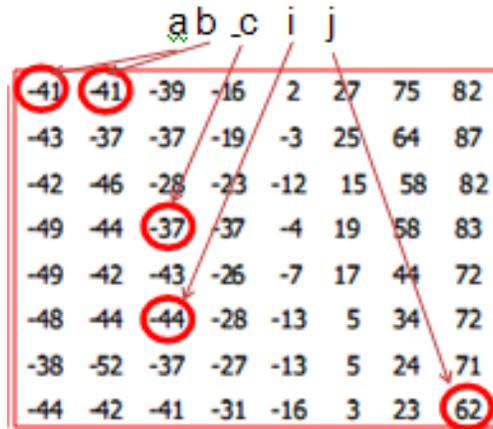


Figura. 11 Bloque procesado y calculado

Debido a que MPLAB Sim no muestra los arreglos bidimensionales en la forma convencional, en la Figura 11 se muestran algunos valores de la matriz $f[y][x]$, resaltados en rojo, para comprobar que los resultados son correctos, dichos valores son: $a = f[0][0]$, $b = f[0][1]$, $_c = f[5][2]$, $i = f[3][2]$, $j = f[7][7]$. Es válido aclarar que los valores se muestran con formato hexadecimal y corresponden a los valores negativos como se muestra en la Figura 12.

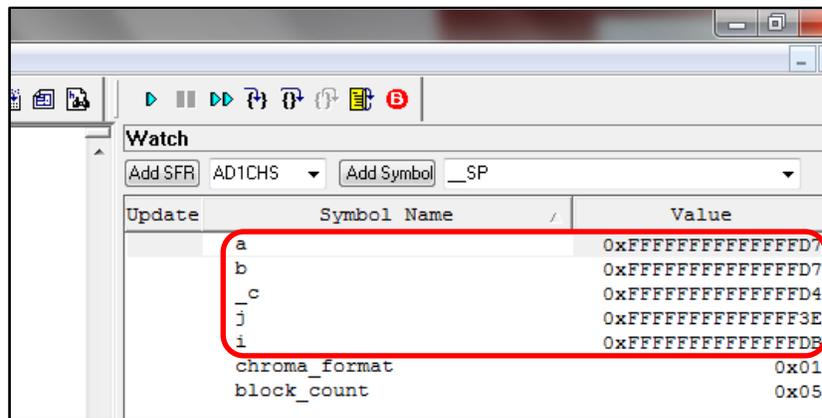


Figura. 12 Ejecución de la función *idct* en MPLAB Sim

3. CONCLUSIONES

El avance hacia la Televisión Digital es una prioridad para la mayoría de los países, Cuba es partícipe de esta revolución mundial en la esfera de las telecomunicaciones y la vía más económica para la migración es la producción a nivel nacional de cajas decodificadoras (set-top-box) asociados a una norma de TVD de las existentes, garantizando la soberanía tecnológica que se desea alcanzar en este sentido.

Una vez comprendida la estructura de la trama MPEG-2 a la salida del codificador y el procesa-miento que requiere la misma para extraer los campos que transporta, según se expone en el estándar ISO/IEC 13818-2, el diseño del decodificador fue dividido en varios bloques funcionales con el fin de reducir su complejidad, de los cuales fueron programados y simulados el subbloque de procesamiento sobre el PIC24FJ128GA010 haciendo uso de la herramientas MPLAB® IDE y PROTEUS® 7.7 sp2, obteniéndose resultados satisfactorios.

El mayor aporte de este trabajo radica en el “know how” derivado del diseño que sienta las bases para el desarrollo de otros estándares de compresión de video más avanzados como H.264/AVC y AVS, los cuales alcanzan mayores tasas de compresión, aunque con la norma de codificación MPEG-2 se logran tasas de compresión aceptables y poca degradación de las imágenes.

4. REFERENCIAS BIBLIOGRÁFICAS

1. Torres Ruiz, F., *Sistemas audiovisuales. Te-levisión analógica y digital*, ed. Ediciones UPC, Cataluña, España, 2000.
2. Microchip Technology Incorporated, *PIC24FJ128GA Family Data Sheet*, M.T., USA, 2006.
3. ISO/IEC13818-2: Information technology-- Generic coding of moving pictures and associated audio information: video, USA, 1995.
4. Garrido González, M., “Arquitectura versátil para la codificación de video multi-estándar: Apor-taciones metodológicas para el diseño de sistemas reutilizables y sistemas en un chip”, Tesis doctoral, Escuela Técnica Superior de Ingenieros en Telecomunicación, Madrid, España, 2004.

5. ANEXOS

Anexo 1. Contenido de la trama MPEG-2.

En la Tabla III se muestra el valor de código de inicio que se transmite después del indicador de inicio 000001h.

Tabla. III Códigos de inicio. [3]

Nombre	Valor de código de inicio (hexadecimal)
Código de inicio de imagen	00
Código de inicio de slice	01 hasta AF
Reservado	B0
Reservado	B1
Código de inicio de datos de usuario	B2
Código de cabecera de secuencia	B3
Código de error de secuencia	B4
Código de inicio de extensión	B5
Reservado	B6
Código de fin de secuencia	B7
Código de inicio de grupo	B8
Códigos de inicio de sistema (ver nota)	B9 hasta FF
NOTA: Los códigos de inicio de sistema están definidos en la ISO /IEC 13818-1 Sistema.	

En caso que el código de inicio sea B5h se debe introducir un identificador de extensión que en la Tabla V se muestra el valor correspondiente a cada extensión definida por la ISO/IEC 13818-2.

Tabla. V Códigos de identificación de extensión de secuencia. [3]

Identificador de extensión de inicio de código (<i>extension start code identifier</i>)	Nombre
0000	Reservado
0001	ID de extensión de secuencia
0010	ID de extensión de secuencia de

	mostrado
0011	ID de extensión de matriz cuantificada
0100	ID de extensión de derecho de autor
0101	ID de extensión de secuencia escalable
0110	Reservado
0111	ID de extensión de mostrado de imagen
1000	ID de extensión de copia de imagen
1001	ID de extensión de escalabilidad espacial de imagen
1010	ID de extensión de escalabilidad temporal de imagen
1011	Reservado
.....	Reservado
1111	Reservado