

IMPLEMENTACIÓN EN HARDWARE DE LA TRANSFORMADA DCT-2D PARA EL
ESTÁNDAR DE COMPRESIÓN DE VIDEO H.265/HEVC.

JOSÉ DANIEL BOLAÑOS JOJOA

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
GRUPO DE INVESTIGACIÓN DE BIONANOELECTRÓNICA
SANTIAGO DE CALI
08/10/2015

IMPLEMENTACIÓN EN HARDWARE DE LA TRANSFORMADA DCT-2D PARA EL ESTÁNDAR DE COMPRESIÓN DE VIDEO H.265/HEVC.

JOSÉ DANIEL BOLAÑOS JOJOA

Tesis presentada como requisito parcial para obtener el título de Magister en
Ingeniería con Énfasis en Ingeniería Electrónica

Director:

JAIME VELASCO MEDINA, Ph.D

Co-director:

JOHN MICHAEL ESPINOSA, M.Sc.



UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
GRUPO DE INVESTIGACIÓN DE BIONANOELECTRÓNICA
SANTIAGO DE CALI
08/10/2015

TABLA DE CONTENIDO

Capítulo 1	1
INTRODUCCIÓN	
Capítulo 2	4
COMPRESIÓN DE VIDEO DIGITAL	
2.1 TIPOS DE REDUNDANCIA DE LOS DATOS.....	4
2.1.1 REDUNDANCIA ESPACIAL	4
2.1.2 REDUNDANCIA TEMPORAL.....	5
2.1.3 REDUNDANCIA ENTRÓPICA.....	5
2.2 MÉTODOS DE COMPRESIÓN DE IMÁGENES	5
2.2.1 COMPRESIÓN SIN PÉRDIDAS	6
2.2.2 COMPRESIÓN CON PÉRDIDAS	6
2.3 RESOLUCIONES DE LA IMAGEN EN UN VIDEO DIGITAL	6
2.4 CODIFICADOR DE VIDEO HÍBRIDO.....	7
Capítulo 3	11
ESTÁNDAR DE CODIFICACIÓN DE VIDEO DE ALTA EFICIENCIA – H.265/HEVC	
3.1 INTRODUCCIÓN.....	11
3.2 H.265/HEVC	12
3.2.1 REPRESENTACIÓN DE LA IMAGEN EN EL ESPACIO DE COLOR.....	14
3.2.2 PARTICIONAMIENTO DE LA IMAGEN	14
Capítulo 4	20
TRANSFORMADA DCT PARA H.265/HEVC	
4.1 INTRODUCCIÓN.....	20
4.2 TRANSFORMADA DCT -1D.....	21
4.3 TRANSFORMADA IDCT -1D	22
4.4 TRANSFORMADA DCT-2D	23
4.5 TRANSFORMADA IDCT-2D	25

4.6 PROPIEDADES DE LA TRANSFORMADA DCT.....	27
4.6.1 ORTOGONALIDAD	27
4.6.2 NORMALIDAD	27
4.6.3 RECURSIVIDAD.....	27
4.6.4 SIMETRÍA Y ANTI-SIMETRÍA.....	28
4.7 TRANSFORMADA DCT-2D PARA H.265/HEVC	28
Capítulo 5.....	37
DISEÑO HARDWARE DE LA TRANSFORMADA DCT PARA H.265/HEVC	
5.1 INTRODUCCIÓN.....	37
5.2 ESTADO DEL ARTE	38
5.3 DISEÑO HARDWARE DE LA TRANSFORMADA DCT-1D	41
5.3.1 TRANSFORMADA DCT-1D ESTRUCTURADA DE N-PUNTOS.....	45
5.3.2 TRANSFORMADA DCT-1D ESTRUCTURADA Y PIPELINE DE N-PUNTOS.....	51
5.3.3 TRANSFORMADA DCT-1D MODIFICADA DE N-PUNTOS	53
5.4 DISEÑO HARDWARE DE LA TRANSFORMADA DCT-2D	60
5.4.1 BLOQUES DCT-1D DE N-PUNTOS.....	62
5.4.2 BLOQUE DE TRASPOSICIÓN DE MATRICES	65
5.4.3 UNIDAD DE CONTROL DE LA TRANSFORMADA DCT-2D.....	67
5.5 DISEÑO HARDWARE DE LA TRANSFORMADA IDCT-1D	73
5.5.1 TRANSFORMADA IDCT-1D ESTRUCTURADA DE N-PUNTOS	76
5.5.2 TRANSFORMADA IDCT-1D ESTRUCTURADA Y PIPELINE DE N-PUNTOS	80
5.5.3 TRANSFORMADA IDCT-1D MODIFICADA DE N-PUNTOS	82
5.6 DISEÑO HARDWARE DE LA TRANSFORMADA IDCT-2D	86
5.6.1 BLOQUES IDCT-1D.....	88
5.7 DISEÑO HARDWARE DE LA TRANSFORMADA DCT-1D-T	91
5.8 DISEÑO HARDWARE DE LA TRANSFORMADA DCT-2D-T	95
5.8.1 BLOQUES DCT-1D DE MULTIPLES TAMAÑOS	97
5.8.2 BLOQUE DE TRASPOSICIÓN DE MATRICES	102
5.8.3 UNIDAD DE CONTROL DE LA TRANSFORMADA DCT-2D-T	105

5.9 DISEÑO HARDWARE DE LA TRANSFORMADA IDCT-1D-T.....	111
5.10 DISEÑO HARDWARE DE LA TRANSFORMADA IDCT-2D-T	115
5.10.1 BLOQUES IDCT-1D DE MULTIPLES TAMAÑOS.....	116
Capítulo 6	118
SIMULACIÓN FUNCIONAL	
6.1 INTRODUCCIÓN.....	118
6.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT	119
6.2.1 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT-1D	119
6.2.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT-2D	123
6.3 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT INVERSA.....	128
6.3.1 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA IDCT-1D	129
6.3.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA IDCT-2D	132
6.4 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT DE MULTIPLES TAMAÑOS.....	137
6.4.1 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT-1D-T	137
6.4.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT-2D-T	139
6.5 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT INVERSA DE MULTIPLES TAMAÑOS	141
6.5.1 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA IDCT-1D-T.....	141
6.5.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA IDCT-2D-T.....	143
Capítulo 7	145
RESULTADOS DE SÍNTESIS Y DESEMPEÑO	
7.1 INTRODUCCION.....	145
7.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT	146
7.2.1 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT-1D.....	146
7.2.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT-2D.....	147
7.3 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT INVERSA.....	148
7.3.1 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA IDCT-1D.....	148
7.3.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA IDCT-2D.....	150

7.4 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT DE MULTIPLES TAMAÑOS	150
7.4.1 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT-1D-T... 151	
7.4.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT-2D-T... 152	
7.5 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT INVERSA DE MULTIPLES TAMAÑOS	152
7.5.1 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA IDCT-1D-T . 152	
7.5.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA IDCT-2D-T . 153	
7.6 COMPARACION CON OTROS TRABAJOS.....	154
Capítulo 8.....	159
CONCLUSIONES Y TRABAJO FUTURO	
8.1 CONCLUSIONES	159
8.2 TRABAJO FUTURO.....	162
BIBLIOGRAFIA	163

LISTA DE FIGURAS

Figura 2.1 Resoluciones de la imagen en un video digital	7
Figura 2.2 Diagrama de bloques genérico de un codificador de video híbrido	8
Figura 3.1 Diagrama de bloques simplificado del codificador H.265/HEVC	13
Figura 3.2 Imagen dividida en CTUs	15
Figura 3.3 Particionamiento de CTUs en CUs	16
Figura 3.4 Modos de particionamiento de un CU en PUs	17
Figura 3.5 Particionamiento de un CU en TUs	19
Figura 4.1 Factores de escalamiento para las transformadas DCT-2D e IDCT-2D en H.265/HEVC	35
Figura 5.1 Bloque MCM para la transformada DCT-1D-E: a) 4-puntos. b) 8-puntos.....	46
Figura 5.2 Bloque MCM para la transformada DCT-1D-E de 16-puntos	46
Figura 5.3 Bloque MCM para la transformada DCT-1D-E de 32-puntos	47
Figura 5.4 Bloques sumadores de la transformada DCT-1D de 8-puntos.....	48
Figura 5.5 Bloques sumadores de la transformada DCT-1D de 16-puntos.....	48
Figura 5.6 Bloques sumadores de la transformada DCT-1D de 32-puntos.....	49
Figura 5.7 Diagrama de bloques de la transformada DCT-1D-E de 4-puntos.....	49
Figura 5.8 Diagrama de bloques de la transformada DCT-1D-E de 8-puntos.....	50
Figura 5.9 Diagrama de bloques genérico de la transformada DCT-1D-E de N-puntos	50
Figura 5.10 Diagrama de bloques de la transformada DCT-1D-P de 4-puntos.....	51
Figura 5.11 Diagrama de bloques de la transformada DCT-1D-P de 8-puntos.....	52
Figura 5.12 Diagrama de bloques genérico de la transformada DCT-1D-P de N-puntos	53
Figura 5.13 Bloque MCM modificado para la transformada DCT-1D-M de 4 y 8-puntos	54
Figura 5.14 Bloque MCM modificado para la transformada DCT-1D-M de 16-puntos.	55
Figura 5.15 Bloque MCM Modificado para la transformada DCT-1D de 32-puntos.....	55
Figura 5.16 Bloque sumador modificado que genera la salida Y(1) en la transformada DCT-1D-M de 8-puntos	57
Figura 5.17 Bloques sumadores modificados para la transformada DCT-1D-M de 16-puntos.	57
Figura 5.18 Bloques sumadores modificados para la transformada DCT-1D-M de 32-puntos.	58
Figura 5.19 Diagrama esquemático del codificador de video H.265/HEVC	61
Figura 5.20 Diagrama de Bloques Genérico de la Transformada DCT-2D de N-Puntos	61
Figura 5.21 a) Diagrama de bloques genérico de la primera etapa de transformación DCT-1D de N-puntos con redondeo y escalamiento	64
Figura 5.21 b) Diagrama detallado del bloque de redondeo y escalamiento - primera etapa...	64
Figura 5.22 a) Diagrama de bloques genérico de la segunda etapa de transformación DCT-1D de N-puntos con redondeo y escalamiento	64

Figura 5.22 b) Diagrama detallado del bloque de redondeo y escalamiento-segunda etapa	64
Figura 5.23 Diagrama de estados de la FSM de la unidad de control para la transformada DCT-2D-E/DCT-2D-M y DCT-2D-P	70
Figura 5.24 Diagrama de tiempo de la unidad de control de la transformada DCT-2D-E/DCT-2D-M	72
Figura 5.25 Diagrama de tiempo de la unidad de control de la transformada DCT-2D-P de 4-puntos	72
Figura 5.26 Bloques sumadores para la transformada IDCT-1D_E de 8 puntos	77
Figura 5.27 Bloques sumadores para la transformada IDCT-1D-E de 16 puntos	77
Figura 5.28 Bloques sumadores para la transformada IDCT-1D-E de 32 puntos	78
Figura 5.29 Diagrama de bloques de la transformada IDCT-1D-E de 4-puntos	78
Figura 5.30 Diagrama de bloques de la transformada IDCT-1D-E de 8-puntos	79
Figura 5.31 Diagrama de bloques genérico de la transformada IDCT-1D-E de N-puntos	79
Figura 5.32 Diagrama de bloques de la transformada IDCT-1D-P de 4-puntos	80
Figura 5.33 Diagrama de bloques de la transformada IDCT-1D-P de 8-puntos	81
Figura 5.34 Diagrama de bloques genérico de la transformada IDCT-1D-P de N-puntos	82
Figura 5.35 Bloque sumador modificado que genera el resultado $Z(4)$ para la transformada IDCT-1D-M de 8-puntos	83
Figura 5.36 Bloques sumadores modificados para la IDCT-1D-M de 16-puntos.....	83
Figura 5.37 Bloques sumadores modificados para la IDCT-1D-M de 32-puntos.....	84
Figura 5.38 a) Diagrama de bloques del codificador de video H.265/HEVC.	87
Figura 5.38 b) Diagrama de bloques del decodificador de video H.265/HEVC.....	87
Figura 5.38 c) Etapas de procesamiento de la transformada IDCT-2D.....	87
Figura 5.39 a) Diagrama de bloques genérico de la primera etapa de transformación IDCT-1D de N-puntos con redondeo y escalamiento	90
Figura 5.39 b) Diagrama detallado del bloque de redondeo y escalamiento-primera etapa	90
Figura 5.40 a) Diagrama de bloques genérico de la segunda etapa de transformación IDCT-1D de N-puntos con redondeo y escalamiento	90
Figura 5.40 b) Diagrama detallado del bloque de redondeo y escalamiento-segunda etapa ...	90
Figura 5.41 Diagrama de bloques de la transformada DCT-1D-T	93
Figura 5.42 Diagrama de bloques de la DCT-1D de 8-puntos y el bloque de permutación.....	95
Figura 5.43 Diagrama de bloques de la transformada DCT-2D-T.....	96
Figura 5.44 a) Diagrama de bloques de la transformada DCT-1D-T usada en la primera etapa de transformación	101
Figura 5.44 b) MUX 4-1 utilizado en la implementación del bloque R&E -primera etapa.....	101
Figura 5.45 a) Diagrama de bloques de la transformada DCT-1D-T usada en la segunda etapa de transformación	102
Figura 5.45 b) MUX 4-1 utilizado en la implementación del bloque R&E -segunda etapa	102
Figura 5.46 Diagramas de estado de las FSMs de la unidad de control de la transformada DCT-2D-T	109

Figura 5.47 Diagrama de tiempo de la unidad de control de la transformada DCT-2D-T.....	110
Figura 5.48 Diagrama de bloques de la IDCT-1D de 8-puntos y el bloque de permutación. ...	112
Figura 5.49 Diagrama de Bloques de la arquitectura de la transformada IDCT-1D-T	114
Figura 6.1 Simulación y verificación de la transformada DCT-1D-E/DCT-1D-M de 4-puntos.	120
Figura 6.2 Simulación y verificación de la transformada DCT-1D-E/DCT-1D-M de 8-puntos.	120
Figura 6.3 Simulación y verificación de la transformada DCT-1D-E/DCT-1D-M 16-puntos. ..	121
Figura 6.4 Simulación y verificación de la transformada DCT-1D-E/DCT-1D-M 32-puntos. ..	122
Figura 6.5 Bloque de datos de 8x8 pixeles de la imagen "Lenna.bmp"	123
Figura 6.6 Simulación y verificación de la transformada DCT-2D-E/DCT-2D-M de 4-puntos.	124
Figura 6.7 Simulación y verificación de la transformada DCT-2D-E/DCT-2D-M de 8-puntos.	124
Figura 6.8 Simulación y verificación de la transformada DCT-2D-E/DCT-2D-M 16-puntos. ..	125
Figura 6.9 Simulación y verificación de la transformada DCT-2D-E/DCT-2D-M 32-puntos. ..	126
Figura 6.10 Simulación y verificación de la transformada DCT-2D-P de 4-puntos.....	128
Figura 6.11 Simulación y verificación de la transformada IDCT-1D-E/IDCT-1D-M 4-puntos.	129
Figura 6.12 Simulación y verificación de la transformada IDCT-1D-E/IDCT-1D-M 8-puntos.	130
Figura 6.13 Simulación y verificación de la transformada IDCT-1D-E/IDCT-1D-M 16puntos	130
Figura 6.14 Simulación y verificación de la transformada IDCT-1D-E/IDCT-1D-M 32-ptos...	131
Figura 6.15 Simulación y verificación de la transformada IDCT-2D-E/IDCT-2D-M 4-puntos.	133
Figura 6.16 Simulación y verificación de la transformada IDCT-2D-E/IDCT-2D-M 8-puntos.	133
Figura 6.17 Simulación y verificación de la transformada IDCT-2D-E/IDCT-2D-M 16-ptos...	134
Figura 6.18 Simulación y verificación de la transformada IDCT-2D-E/IDCT-2D-M 32-ptos...	135
Figura 6.19 Simulación y verificación de la transformada DCT-1D-T para 4 y 8-puntos.....	137
Figura 6.20 Simulación y verificación de la transformada DCT-1D-T para 16 y 32-puntos. ...	138
Figura 6.21 Simulación y verificación de la transformada DCT-2D-T para 4 y 8-puntos.....	140
Figura 6.22 Simulación y verificación de la transformada IDCT-1D-T para 4 y 8-puntos.	141
Figura 6.23 Simulación y verificación la transformada IDCT-1D-T para 16 y 32-puntos.....	142
Figura 6.24 Simulación y verificación de la transformada IDCT-2D-T para 4 y 8-puntos.	144

LISTA DE TABLAS

Tabla 4.1 Matriz base de la transformada DCT 4x4 para H.265/HEVC.....	30
Tabla 4.2 Matriz base de la transformada DCT 8x8 para H.265/HEVC.....	30
Tabla 4.3 Matriz base de la transformada DCT 16x16 para H.265/HEVC	31
Tabla 4.4 Matriz base de la transformada DCT 32x32 para H.265/HEVC	31
Tabla 5.1 Valores Únicos de la matriz $W_{N/2}$ para la transformada DCT-1D de N-puntos.....	44
Tabla 5.2 Ciclos de latencia inicial de los bloques que conforman la arquitectura de la transformada DCT-2D-E/DCT-2D-M.....	69
Tabla 5.3 Ciclos de latencia inicial de los bloques que conforman la arquitectura de la transformada DCT-2D-P	70
Tabla 5.4 Señales de entrada de la FSM de la unidad de control para la transformada DCT-2D-E/DCT-2D-M y DCT-2D-P	71
Tabla 5.5 Señales de Salida de la FSM de la unidad de control para la transformada DCT-2D-E/DCT-2D-M y DCT-2D-P	71
Tabla 5.6 Tabla de verdad de la unidad de control interna de la arquitectura DCT-1D-T	92
Tabla 5.7 Bits a truncar en el escalamiento y bit de redondeo para la transformada DCT-1D-T usada en la primera etapa de transformación de la DCT-2D-T.....	98
Tabla 5.8 Bits a truncar en el escalamiento y bit de redondeo para la transformada DCT-1D-T usada en la segunda etapa de transformación de la DCT-2D-T.	99
Tabla 5.9 Ciclos de latencia de los bloques que conforman la transformada DCT-2D-T.....	107
Tabla 5.10 Señales de entrada de la unidad de control de la transformada DCT-2D-T	110
Tabla 5.11 Señales de Salida de la Unidad de control de la transformada DCT-2D-T.....	110
Tabla 5.12 Tabla de Verdad de la Unidad de control interna de la arquitectura IDCT-1D-T...	113
Tabla 7.1 Resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas DCT-1D para el estándar H.265/HEVC.....	147
Tabla 7.2 Resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas DCT-2D para el estándar H.265/HEVC.....	148
Tabla 7.3 Resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas IDCT-1D para el estándar H.265/HEVC.....	149
Tabla 7.4 Resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas IDCT-2D para el estándar H.265/HEVC.....	150
Tabla 7.5 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada DCT-1D-T para el estándar H.265/HEVC.....	151
Tabla 7.6 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada DCT-2D-T para el estándar H.265/HEVC.....	152

Tabla 7.7 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada IDCT-1D-T para el estándar H.265/HEVC.....	153
Tabla 7.8 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada IDCT-2D-T para el estándar H.265/HEVC.....	154
Tabla 7.9 Resultados de síntesis y desempeño de las arquitecturas hardware para la transformada DCT-1D y otros trabajos presentados en la literatura	155
Tabla 7.10 Resultados de síntesis y desempeño de las arquitecturas hardware para la transformada DCT-2D de 16-puntos y otro trabajo presentado en la literatura	156
Tabla 7.11 Resultados de síntesis y desempeño de las arquitecturas hardware para la transformada IDCT-2D y otro trabajo presentado en la literatura	157
Tabla 7.12 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada IDCT-2D-T y otro trabajo presentado en la literatura.....	157

Capítulo 1

INTRODUCCIÓN

El procesamiento digital de video es actualmente un tema de investigación y desarrollo tecnológico muy relevante y de gran interés para la comunidad científica, académica e industrial, debido a la gran variedad de aplicaciones que utilizan los desarrollos sobre procesamiento digital de video; incluyendo vigilancia y seguridad, control de tráfico, multimedia y telecomunicaciones, automatización industrial, control y detección, etc. Una de las aplicaciones del procesamiento de video que ha sido ampliamente comercializada es la codificación y decodificación de videos digitales. Esta aplicación ha generado un gran impacto a nivel tecnológico convirtiendo a los codificadores y decodificadores de video digital en una parte fundamental de los dispositivos electrónicos actuales tales como: televisores, reproductores de video DVD y Blu-ray, Smart-phones, computadores, tablets, cámaras digitales y cualquier dispositivo que sea capaz de reproducir videos transmitidos por medio de internet.

El objetivo principal de los codificadores de video es comprimir la cantidad de datos que representan un video digital con el propósito de disminuir el tiempo de transmisión a través de la red. Sin embargo, debido al rápido avance de la tecnología, impulsada por las grandes exigencias de los usuarios, los cuales cada vez desean visualizar vídeos de mayor calidad y resolución, hace que la compresión y transmisión de videos de alta resolución sea un gran reto.

Para transmitir archivos cada vez más grandes como los videos de alta resolución sin disminuir la velocidad de transmisión es necesario realizar una mayor compresión de los datos sin afectar la calidad perceptible de éstos. Para enfrentar este problema y al mismo tiempo homogeneizar comercialmente los codificadores y decodificadores de video se han

desarrollado estándares de compresión de video de alta eficiencia, los cuales usan diferentes algoritmos y técnicas de procesamiento con el fin de reducir el número de bits necesarios para representar el contenido de un vídeo, aprovechando la correlación de los datos y las limitaciones del sistema visual humano (HVS: *Human Visual System*), y de esta forma eliminar los datos redundantes e irrelevantes, respectivamente.

El estándar H.265/HEVC (*High Efficiency Video Coding*) es el estándar de codificación de video más reciente, desarrollado por el grupo de expertos en codificación de video (ISO MPEG) y el grupo de expertos en imágenes en movimiento (ITU-T VCEG).

El principal objetivo en el desarrollo del estándar H.265/HEVC fue duplicar la eficiencia de la codificación con respecto al anterior estándar H.264/AVC (*Advanced Video Coding*), reduciendo a la mitad la tasa de bits a transmitir (*bit-rate*) y manteniendo la misma calidad de video [1] [2].

Entre las diferentes etapas de codificación del estándar H.265/HEVC, una de las más importantes es la etapa de transformación, su finalidad es concentrar la energía de una imagen en tan sólo unos pocos coeficientes numéricos [3]. En este caso, la transformada encargada de realizar este proceso es la Transformada Discreta del Coseno (DCT – Discrete Cosine Transform), la cual es ampliamente utilizada en los estándares de compresión más recientes y es uno de los bloques funcionales que determina en gran medida la eficiencia de compresión de los codificadores de video.

Usualmente los estándares de codificación de video tradicionales seleccionan un tamaño de 4x4 u 8x8 para la transformada DCT-2D con el fin de equilibrar la eficiencia de la codificación y el costo de la implementación. Sin embargo, se ha demostrado que al usar tamaños más grandes de esta transformada se generan mejores resultados en la concentración de la energía, aunque esto produce un incremento exponencial en la complejidad computacional [4]. Con el fin de alcanzar una gran compresión de los datos, el nuevo estándar H.265/HEVC incluye las transformadas DCT-2D de tamaño 16 y 32, obteniendo una alta eficiencia en la codificación de video [5].

Generalmente los diferentes algoritmos y técnicas de procesamiento de un estándar se implementan en software, sin embargo, para procesar videos de alta resolución en tiempo real se requiere una mayor velocidad de procesamiento debido a la gran cantidad de datos. Una de las soluciones más atractivas y eficientes en este tipo de casos es la implementación en hardware de estos algoritmos, permitiendo incrementar la velocidad de procesamiento, reducir el consumo de energía y mejorar el desempeño de los codificadores y decodificadores.

Teniendo en cuenta lo anterior, el principal objetivo de este trabajo es diseñar una arquitectura hardware de alto desempeño y bajo consumo de potencia para la transformada DCT-2D que cumpla con los requerimientos del estándar H.265/HEVC.

En este trabajo se presentan las implementaciones eficientes en hardware de la transformada DCT-2D e IDCT-2D de tamaño 4, 8, 16 y 32. Además se presenta una nueva arquitectura hardware de la transformada DCT-2D directa e inversa que incluye a todos los tamaños, es decir, una arquitectura altamente parametrizable.

Para la implementación en hardware de las transformadas DCT-2D e IDCT-2D se utilizan arquitecturas paralelas basadas en técnicas pipeline, diagramas mariposa y optimizaciones hardware para la reducción de la complejidad, con el fin de obtener diseños con una alta frecuencia de operación y un consumo de área y potencia moderado.

Este trabajo está organizado de la siguiente manera: en el capítulo 2 se presentan algunos conceptos básicos sobre la codificación de video digital. En el capítulo 3 se presenta una breve descripción de algunas características del estándar H.265/HEVC. En el capítulo 4 se presenta una descripción general de la transformada DCT utilizada en el estándar H.265/HEVC. En el capítulo 5 se presentan los diseños hardware de la transformada DCT-2D tanto directa como inversa en donde también se incluye el diseño parametrizable de la transformada DCT-2D de múltiples tamaños. En el capítulo 6 se presentan los resultados de la simulación funcional de cada una de las arquitecturas propuestas. En el capítulo 7 se presentan los resultados de síntesis y desempeño, y se realiza una comparación entre los diseños hardware presentados en este trabajo y otras implementaciones presentes en la literatura. Finalmente en el capítulo 8 se presentan las conclusiones y el trabajo futuro.

Capítulo 2

COMPRESIÓN DE VIDEO DIGITAL

La compresión de video o codificación de video es el proceso que reduce la cantidad de datos necesarios para representar una señal de video digital antes de ser transmitida o almacenada. La operación complementaria, descompresión o decodificación, recupera la señal de video digital a partir de la señal comprimida antes de ser reproducida [6]. El proceso de compresión se realiza eliminando los datos redundantes sin afectar la calidad perceptible del video, con el fin de reducir la cantidad de datos a transmitir o almacenar.

2.1 TIPOS DE REDUNDANCIA DE LOS DATOS

Un dato se considera redundante si éste no se atribuye como información relevante para representar una imagen. Básicamente existen tres tipos diferentes de redundancia en los sistemas de compresión de video: redundancia espacial, redundancia temporal y redundancia entrópica.

2.1.1 REDUNDANCIA ESPACIAL

La redundancia espacial o redundancia *“intra-frame”* surge de la correlación existente entre píxeles adyacentes. Esta redundancia es percibida visualmente cuando píxeles vecinos tienden a poseer valores semejantes. La correlación de los píxeles adyacentes puede ser identificada tanto en el dominio espacial como en el dominio frecuencial. En los codificadores de video híbridos (codificadores que explotan la redundancia espacial y temporal), la etapa que aprovecha la redundancia espacial es llamada *“intra-frame coding”*. En el dominio frecuencial la operación que realiza este proceso es la cuantización. Antes de realizar la cuantización a los datos de una imagen es necesario transformarlos desde el dominio espacial al dominio frecuencial utilizando una transformada bidimensional.

2.1.2 REDUNDANCIA TEMPORAL

La redundancia temporal o redundancia *"inter-frame"* surge de la correlación entre tramas temporalmente próximas. En un video, muchos de los bloques de píxeles no cambian su valor de una trama a otra. Algunos píxeles presentan una pequeña variación debido a cambios de iluminación y otros pueden desplazarse de un sector a otro debido al movimiento de un objeto en una escena del video [7]. En los codificadores de video híbridos la etapa que aprovecha esta redundancia es llamada *"inter-frame coding"*.

2.1.3 REDUNDANCIA ENTRÓPICA

La redundancia entrópica está relacionada con la probabilidad de ocurrencia de dos símbolos codificados. La entropía también se puede considerar como la cantidad de información promedio que contienen los símbolos usados en un video. Los símbolos con menor probabilidad son los que aportan mayor información; por ejemplo, si se considera como sistema de símbolos a las palabras en un texto, palabras frecuentes como "que", "el", "a" aportan poca información, mientras que palabras menos frecuentes como "corren", "niño", "perro" aportan más información. Si de un texto dado borramos un "que", seguramente no afectará la comprensión y se sobreentenderá, no siendo así si borramos la palabra "niño" del mismo texto original. Cuando todos los símbolos son igualmente probables, todos aportan información relevante y la entropía es máxima [8]. Por lo tanto, los codificadores que aprovechan la redundancia entrópica tienen como objetivo transmitir el máximo de información posible por símbolo codificado y, de este modo, representar más información con un número menor de bits. Un codificador de entropía utiliza diferentes técnicas y algoritmos de compresión sin pérdidas para lograr este objetivo.

2.2 MÉTODOS DE COMPRESIÓN DE IMÁGENES

Hay diversas clasificaciones en el campo de la compresión basadas en la aplicación y el tipo de algoritmo utilizado. Sin embargo existen dos tipos básicos de compresión, compresión sin pérdidas y compresión con pérdidas.

2.2.1 COMPRESIÓN SIN PÉRDIDAS

Los algoritmos de compresión de datos sin pérdidas usualmente aprovechan la redundancia entrópica de los datos para representarlos de forma más concisa sin perder información. La compresión sin pérdidas es posible porque la mayoría de los datos del mundo real poseen redundancia entrópica. Existen muchos codificadores para reducir el tamaño de los archivos eliminando esta redundancia de los datos, entre los más conocidos están “*Run-length Coding*” y “*Entropy Coding*”.

2.2.2 COMPRESIÓN CON PÉRDIDAS

En este tipo de compresión algunas pérdidas de información irrelevante son aceptables. Despreciando detalles no esenciales de los datos de entrada es posible ahorrar espacio de almacenamiento. Este tipo de compresión se basa en las limitaciones del sistema visual humano (HVS) para eliminar los datos irrelevantes. Por ejemplo, el ojo humano es más sensible a las variaciones en la luminancia que a las variaciones en el color, por lo cual es posible despreciar algunos de estos datos sin afectar la calidad perceptible de la imagen. Es importante mencionar que en los codificadores que usan este tipo de compresión debe existir un equilibrio entre el grado de compresión y la calidad de la imagen.

2.3 RESOLUCIONES DE LA IMAGEN EN UN VIDEO DIGITAL

La calidad de las imágenes observadas en un vídeo no se basa simplemente en el número de cuadros por segundo o en la forma en que las tramas están compuestas. La cantidad de información en cada trama, o la resolución de la imagen, también es un factor importante. Entre mayor sea la resolución de una imagen mayor será la cantidad de píxeles que la componen. A partir de la Figura 2.1 se puede observar que la resolución de las imágenes en un video digital varía en gran medida para los diferentes tipos de pantalla. Televisores de definición estándar están representadas por el área amarilla con una resolución de 720 x 486 píxeles, mientras que los televisores modernos de alta definición corresponden a las siguientes áreas, ya sea 720p HD (1280x720 píxeles) o 1080p Full HD (1920x1080 píxeles).

Los televisores de ultra alta definición 4K y 8K son la tendencia actual alcanzando resoluciones de 4096 x 2160 píxeles y 8192 x 4608 píxeles, respectivamente [9].

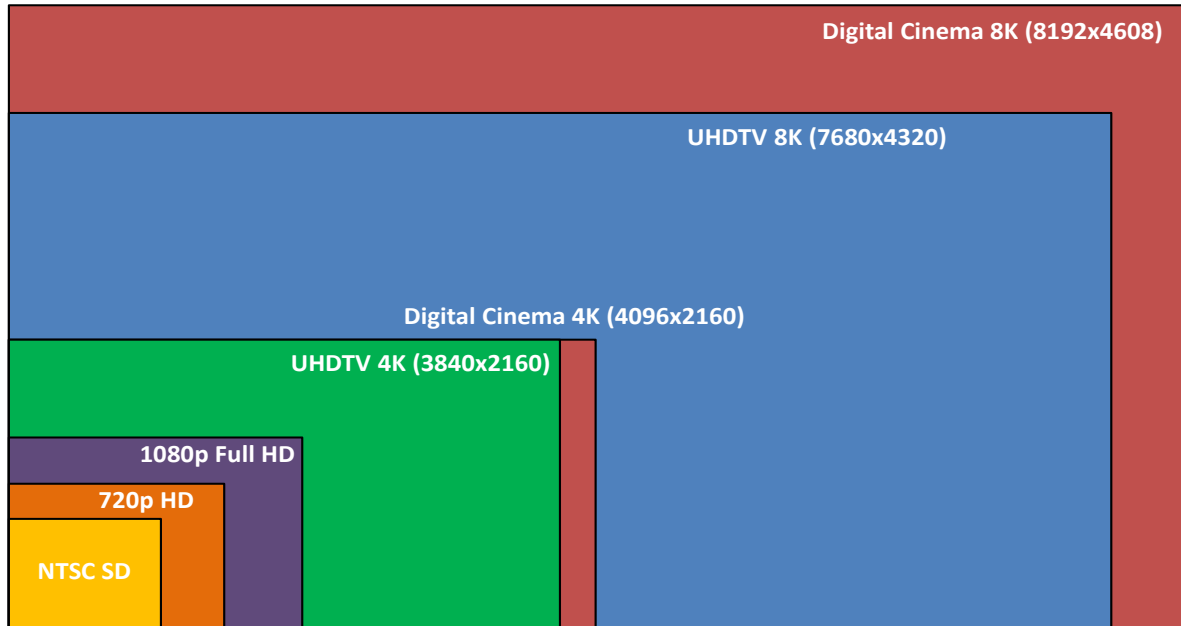


Figura 2.1 Resoluciones de la imagen en un video digital

2.4 CODIFICADOR DE VIDEO HÍBRIDO

El sistema de codificación de vídeo híbrido es el más eficiente de los sistemas de compresión de vídeo actuales. La noción de híbrido proviene del hecho de que se combina la codificación de imágenes fijas y la explotación temporal de imágenes consecutivas en el mismo sistema de compresión. Los sistemas de compresión híbridos aprovechan la redundancia temporal entre las distintas tramas mediante el uso de la estimación y compensación de movimiento, y la redundancia espacial dentro de una misma trama mediante el uso de transformadas bidimensionales. Estas técnicas consiguen reducir a cero el valor de la mayoría de los píxeles, de tal forma que el codificador de entropía usado después de estas etapas de codificación consigue mejores resultados de compresión [10]. La Figura 2.2 muestra el diagrama de bloques genérico de un codificador de video híbrido.

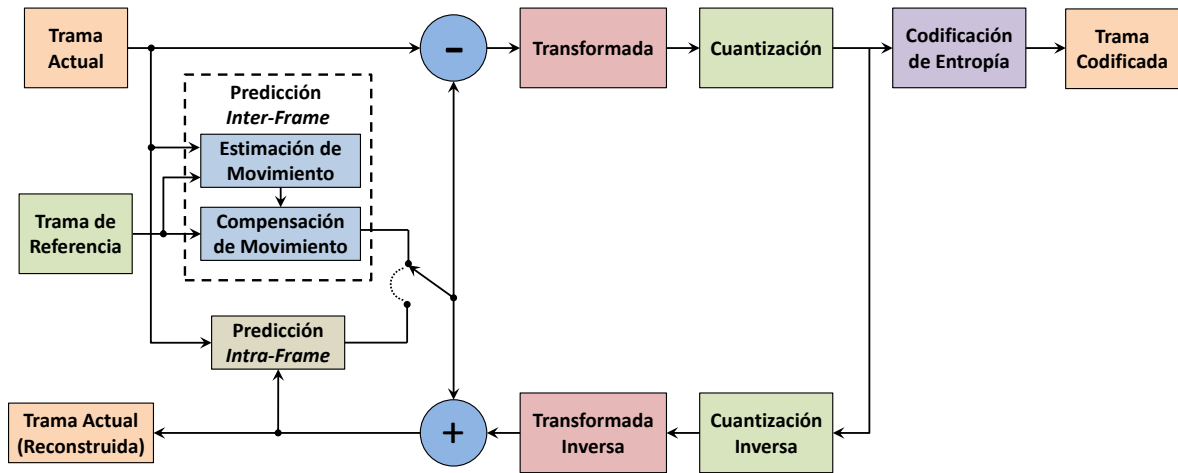


Figura 2.2 Diagrama de bloques genérico de un codificador de video híbrido

En general, la codificación de vídeo híbrida es común por denotar tres tipos de tramas dependiendo de la condición de codificación: tipo I, tipo P y tipo B.

La trama tipo I o “*intra coded frame*”, es una trama de referencia completa, lo que significa que no necesita de otras tramas para ser decodificada, no hace referencia a ninguna otra trama y sólo tiene en cuenta la información contenida en ella misma. Por lo tanto no realiza la predicción “*inter-frame*” y no propaga posibles errores. La predicción utilizada en este tipo de tramas es la predicción “*intra-frame*”. El propósito de la trama tipo I es proporcionar una trama de referencia para las tramas tipo P y también para crear puntos de acceso aleatorio en el flujo de datos. Debido al hecho de que ésta es una trama de referencia completa requiere más bits que los otros tipos de tramas.

La trama tipo P o “*Predicted-Frame*” requiere una trama de referencia previa para ser decodificada. Esta trama puede contener datos de la imagen y/o vectores de movimiento y usa la predicción “*inter-frame*” para su codificación. Además este tipo de tramas sólo hacen referencia a otras tramas I o P anteriores, por lo cual también se pueden utilizar como una trama de referencia para la predicción de tramas [11].

La trama tipo B o "*Bi-predicted Frame*" puede explotar la predicción temporal a partir de tramas de referencias tanto previas como futuras. La trama tipo B realiza la estimación de movimiento dos veces, con respecto a una imagen futura y una imagen pasada, con lo que se obtienen dos vectores de movimiento y dos predicciones por bloque. De esta forma, el codificador puede elegir si utiliza la referencia futura, la pasada o una combinación de ambas. En los recientes esquemas de compresión esta trama también puede ser usada como una trama de referencia.

La tasa de compresión de las tramas tipo P suele ser el doble que las de tipo I, y en el caso de las de tipo B el factor de compresión casi cuadriplica a las de tipo I. Por su parte, las tramas de tipo I no logran obtener una gran compresión pero emplean menos tiempo en su codificación ya que no deben realizar la búsqueda de referencias.

En el codificador híbrido de la Figura 2.2 cada trama se divide en bloques que pueden tener diferentes tamaños. El tamaño de cada bloque depende del tipo de codificador y de la decisión adoptada después de analizar la trama a codificar. Inicialmente, se realiza la predicción "*inter-frame*" e "*intra-frame*" para cada bloque y, después de estos pasos, el codificador toma una decisión acerca de cuál de las dos predicciones presenta mejores resultados. Posteriormente, se lleva a cabo una sustracción entre el bloque original y el bloque previsto generando un residuo.

La predicción "*inter-frame*" tiene como objetivo eliminar la redundancia temporal. El proceso de Estimación de Movimiento dentro de la predicción "*inter-frame*" consiste en buscar la similitud entre la trama actual y las tramas temporalmente adyacentes. Esta redundancia de información se produce debido a que la tasa de muestreo genera muchos bloques con datos redundantes o bloques con pequeños desplazamientos de una trama a otra en un vídeo. Para realizar la predicción "*inter-frame*" debe haber al menos una trama procesada anteriormente, para ser utilizada como referencia, de esta forma sólo se codifica la diferencia entre las dos partes y el vector de movimiento que indica el desplazamiento con respecto a la trama de referencia. El proceso de reconstrucción de la imagen original a partir de estos vectores de movimiento es llamado Compensación de Movimiento. Las tramas generadas a partir de las tramas anteriores hacen referencia a las tramas tipo P y tipo B [12].

En la codificación “*intra-frame*”, se busca eliminar la redundancia espacial. Esta redundancia de información es causada por el hecho de que los píxeles vecinos tienden a tener valores similares y cuanto mayor sea la resolución del vídeo mayor será esta redundancia. La predicción “*intra-frame*” siempre se realiza a la primera trama del vídeo que es una trama tipo I, ya que no depende de tramas previamente codificadas. Como una función de sincronización y reducción de la propagación del error, se insertan tramas tipo I a intervalos constantes de tramas tipo P o tipo B, y todos los bloques dentro de las tramas tipo I se codifican mediante predicción “*intra-frame*”. Por otra parte, también es posible usar la predicción “*intra-frame*” en tramas tipo P o tipo B, ya que en algunos casos, esta predicción logra mejores resultados que la predicción “*inter-frame*” [6].

Después de la etapa de predicción, el residuo se envía al módulo de la transformada, que es responsable de la transformación de los datos desde el dominio espacial al dominio frecuencial. En este dominio, es posible identificar las frecuencias menos relevantes para el sistema visual humano y eliminarlas a través del proceso de cuantización.

La cuantización se lleva a cabo a través de una división entera entre los coeficientes transformados y un factor. Este factor puede ser distinto para cada posición del bloque, siendo mayor en los coeficientes menos importantes. En este proceso se pierde más información cuanto es mayor el factor de cuantización. Para el caso de los valores que después de la cuantización sean cero, estos no podrán recuperarse, ya que se habrá perdido toda la información relativa a ellos. A pesar de ello, las pérdidas no son muy significativas y serán imperceptibles para el sistema visual humano si estos valores hacen parte de los valores de alta frecuencia de la imagen, y el hecho de que sean cero incrementa la tasa de compresión en el codificador de entropía.

Por último, la codificación de entropía tiene como objetivo eliminar la redundancia entrópica, que está relacionada con la forma de representación de los símbolos codificados. La codificación de entropía utiliza varios algoritmos y técnicas para la compresión sin pérdida representando la información con un número menor de bits [7].

Capítulo 3

ESTÁNDAR DE CODIFICACIÓN DE VIDEO DE ALTA EFICIENCIA – H.265/HEVC

3.1 INTRODUCCIÓN

El desarrollo de tecnologías que permiten la captura y visualización de contenidos de vídeo en alta definición ha provocado un incremento del contenido multimedia con altas resoluciones para aplicaciones emergentes. Este crecimiento no solo se ha limitado a la alta definición (720p HD), también ha evolucionado a contenidos de vídeo de ultra alta definición (UHDTV – 8K) con una resolución de 7680×4320 píxeles, que equivale a 16 veces la resolución de 720p HD. Sin duda, este tipo de contenidos de video requiere mayor ancho de banda y una gran capacidad de almacenamiento, que no era posible obtener con los estándares tradicionales de transmisión y almacenamiento. A raíz de este problema fue desarrollado un nuevo estándar de compresión de video H.265/HEVC, el cual supera de manera significativa la eficiencia de compresión de su predecesor H.264/AVC.

El estándar H.265/HEVC es el estándar de codificación de video más reciente, desarrollado por el grupo de expertos en codificación de video (ISO MPEG) y el grupo de expertos en imágenes en movimiento (ITU-T VCEG), con el objetivo de duplicar la eficiencia de compresión con respecto al anterior estándar H.264/AVC, reduciendo a la mitad la tasa de bits a transmitir (*bit-rate*), manteniendo la misma calidad de video y soportando resoluciones de video desde 320×240 píxeles a 7680×4320 píxeles (UHDTV – 8K) [1] [2].

El estándar HEVC es el resultado de una serie de reuniones técnicas en donde se definieron y se mejoraron todas las características del estándar utilizando varios modelos de prueba, HM1 (Octubre/2010) – HM16 (Febrero/2015) los cuales presentan un mejor rendimiento que el

modelo inmediatamente anterior en términos de equilibrio entre complejidad y eficiencia del codificador.

La primera versión de este estándar fue finalizada en Abril del 2013, posteriormente, en Octubre del 2014 se finaliza la segunda versión, en la cual se incluyeron las extensiones RExt (Format Range Extension), SHVC (Scalability Extension) y MV-HEVC (Multi-view Extension). Finalmente, en Febrero del 2015 fue finalizada la tercera y última versión hasta el momento, en donde se incluyó la extensión 3D-HEVC [13].

En esta sección se presenta una breve descripción de algunas de las características del estándar H.265/HEVC que son de interés para este trabajo teniendo en cuenta el modelo HM16 [14].

3.2 H.265/HEVC

El estándar H.265/HEVC está diseñado para lograr múltiples objetivos, entre ellos una mayor eficiencia de compresión, fácil integración de sistemas de transporte, mayor capacidad de recuperación de pérdida de datos, y gran aplicabilidad en el uso de arquitecturas de procesamiento paralelo [1][9].

Como su antecesor, H.265/HEVC usa el modelo de codificador híbrido, sin embargo, en contraste con previos estándares de compresión de video, éste emplea una estructura de particionamiento en árbol flexible o “*quadtree*”, el cual permite usar tamaños más grandes para los bloques de predicción, codificación y transformación. Este estándar también emplea bloques mejorados para la codificación “*intra-frame*” e “*inter-frame*”, un nuevo filtro de bucla y una versión mejorada del algoritmo de codificación aritmética binaria de contexto adaptativo (CABAC: *Context-adaptive binary arithmetic coding*) o codificación de entropía [10][14]. En la Figura 3.1 se presenta el diagrama de bloques simplificado del codificador H.265/HEVC.

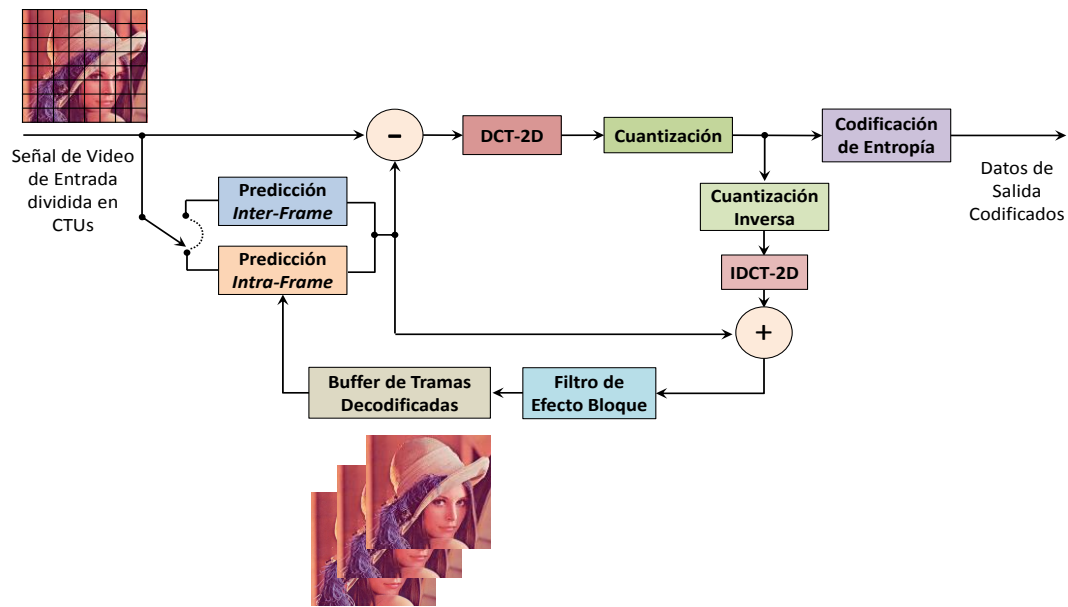


Figura 3.1 Diagrama de bloques simplificado del codificador H.265/HEVC

En el proceso de codificación del estándar H.265/HEVC, cada imagen de entrada es dividida en bloques de datos. La primera imagen de la secuencia de video de entrada es codificada usando solo la predicción “*intra-frame*”. Para la mayoría de las imágenes posteriores o imágenes que se encuentren entre puntos de acceso aleatorio se usa la predicción “*inter-frame*”. La señal residual de la predicción “*intra-frame*” o “*inter-frame*”, la cual es la diferencia entre el bloque de datos original y su predicción, es transformada usando la transformada discreta del coseno bidimensional (DCT-2D). Los datos transformados son escalados, cuantificados y luego codificados por el codificador de entropía. Finalmente los datos codificados son transmitidos junto con la información de predicción para su futura decodificación [1].

No existe un elemento de codificación único en H.265/HEVC que proporcione un significativo mejoramiento en la eficiencia de compresión en relación con los anteriores estándares de codificación de video. Es, más bien, una serie de pequeñas mejoras que se suman obteniendo una ganancia significativa [1].

Algunas de las diversas características implicadas en la codificación de vídeo híbrida del estándar H.265/HEVC son descritas a continuación.

3.2.1 REPRESENTACIÓN DE LA IMAGEN EN EL ESPACIO DE COLOR

Para representar señales de video a color, H.265/HEVC típicamente utiliza el espacio de color YCbCr con muestreo 4:2:0. Este divide el espacio de color en tres componentes llamados Y, Cb y Cr. El componente Y es también llamado luminancia y representa el brillo de la imagen. Los dos componentes de crominancia Cb y Cr representan la medida en que el color se desvía del gris hacia el azul y del gris hacia el rojo, respectivamente. Debido a que el sistema visual humano es más sensible a los cambios de luminancia se utiliza la estructura de muestreo 4:2:0, en la cual cada componente de crominancia tiene un cuarto de las muestras del componente de luminancia. Cada muestra puede representarse con 8 o 10 bits de precisión, siendo 8 el valor típico.

3.2.2 PARTICIONAMIENTO DE LA IMAGEN

El estándar H.265/HEVC está especialmente diseñado para manejar resoluciones de video de hasta 7680x4320 píxeles, la cual es una resolución 75 veces mayor que la resolución usada tradicionalmente para la televisión y 16 veces mayor que la utilizada en la televisión HD.

El hecho de usar tamaños de imagen consecutivas tan grandes implica que los macro-bloques usados por los anteriores estándares de codificación de vídeo procesan un área demasiado pequeña en relación al tamaño total de la imagen. Por esta razón, H.265/HEVC introduce estructuras de particionamiento en árbol o “*quadtree*”, y permite el uso de bloques de hasta 64x64 píxeles, esto es, 16 veces más grandes que los macro-bloques utilizados en estándares anteriores [10].

El término “*quadtree*” es usado para describir una clase de estructura de datos jerárquica que está basada en el principio de descomposición recursiva en el espacio. La representación más común de un “*quadtree*” se basa en la subdivisión sucesiva de la imagen en cuatro cuadrantes de igual tamaño. Si el cuadrante resultante no está compuesto en su totalidad de valores altamente correlacionados, este se sub-divide nuevamente en cuatro sub-cuadrantes de igual tamaño y así sucesivamente hasta obtener la correlación deseada [15].

H.265/HEVC hace uso de estructuras de particionamiento en árbol o “*quadtree*” con el fin de incrementar la eficiencia de codificación. En H.265/HEVC se utilizan varias estructuras de este tipo, las cuales se describen a continuación.

3.2.2.1 Estructuras CTU

En H.265/HEVC las imágenes de entrada se dividen en estructuras cuadradas llamadas “*coding tree units*” (CTUs). El concepto de CTU es equivalente en un amplio sentido al del macro-bloque en estándares anteriores como H.264/AVC. El tamaño de las CTUs está definido por el codificador y puede ser más grande que un macro-bloque tradicional. Para una imagen que posee tres arreglos de muestras, un CTU estará compuesto por un “*coding tree block*” (CTB) de luminancia de tamaño $N \times N$ y sus correspondientes CTBs de crominancia. El tamaño $N \times N$ de un CTB de luminancia se puede elegir como $N = 16, 32$, o 64 muestras, donde los tamaños más grandes permiten una mejor compresión [1][13]. En la Figura 3.2 se muestra un ejemplo de una imagen dividida en CTUs

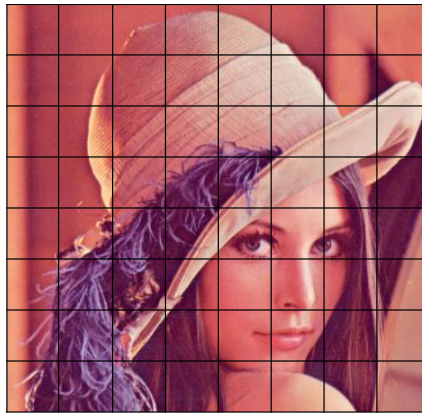


Figura 3.2 Imagen dividida en CTUs

3.2.2.2 Unidad de Codificación CU

Cada CTU puede ser dividida en una o más unidades básicas de codificación, llamadas “*coding units*” (CU), las cuales son estructuras cuadradas que comparten el mismo modo de predicción: “*Intra-frame*”, “*Inter-frame*” o “*Skipped*”. Los bloques CTBs ya sean de luminancia o crominancia pueden ser utilizados directamente como “*coding blocks*” (CBs) o pueden ser

divididos utilizando un particionamiento en árbol o “*quadtree*”, el cual realiza una división recursiva de las estructuras cuadradas en cuatro bloques de tamaños iguales a partir del tamaño del CTB hasta un tamaño mínimo de 8x8 píxeles [1]. La unidad CTU contiene la sintaxis del particionamiento “*quadtree*” en donde se especifica el tamaño apropiado de los bloques CBs teniendo en cuenta las características de la señal de entrada. Este proceso recursivo permite un particionamiento adaptativo al contenido de la imagen, el cual está compuesto por bloques CU cuadrados con dimensiones que varían desde 8x8 hasta 64x64 [14]. La Figura 3.3 muestra un posible particionamiento de 4 CTUs en 34 CUs.

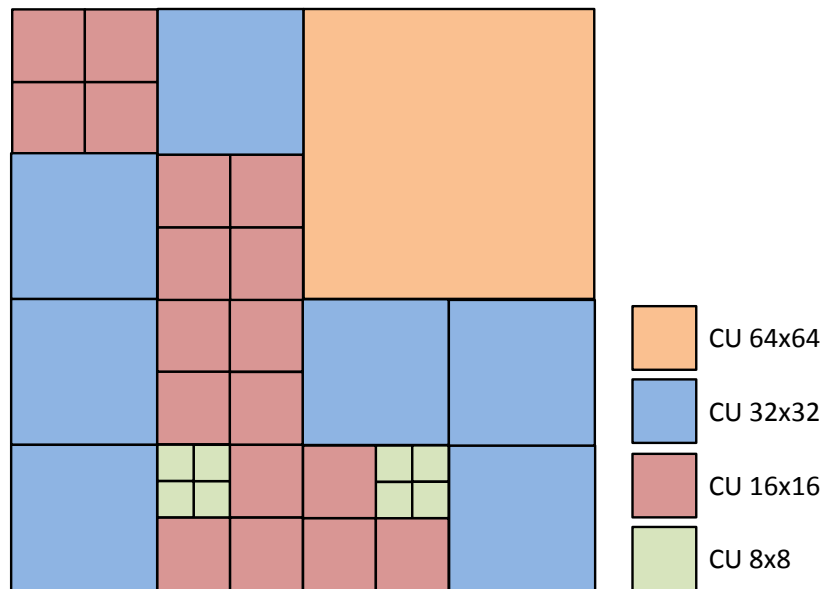


Figura 3.3 Particionamiento de CTUs en CUs

Cada bloque CU no omitido es asignado a uno de los dos modos de predicción: “*Intra-frame*” o “*Inter-frame*”. Un bloque CU omitido se asigna al modo de predicción “*Inter-frame*” sin diferencias en los vectores de movimiento y en la información residual.

3.2.2.3 Unidad de Predicción PU

La unidad de predicción (PU) es una región definida por el particionamiento de la CU, sobre la cual se realiza el mismo tipo de predicción. En general, las PUs contienen la información

relacionada con el proceso de predicción y su tamaño no está restringido a ser cuadrado, lo cual les permite adaptarse a las características de la imagen.

Cuando se ha especificado el modo de predicción “*Intra-frame*”, el tamaño de los bloques de predicción PUs será el mismo que el de los bloques CUs que los contienen para todos los tamaños excepto para el menor tamaño de los CU (8x8), en cuyo caso se realiza una división en cuatro bloques PUs de igual tamaño (4x4) sobre los cuales se realiza su propia predicción “*Intra-frame*”.

Para el caso de la predicción “*Inter-frame*”, cada CU puede ser dividida en una, dos o cuatro PUs. La división en cuatro PUs solo es permitida cuando el tamaño del CU es el mínimo permitido para este tipo de estructura (8x8). Cuando un CU es dividido en cuatro PUs, cada PU abarca un cuadrante del CU. Cuando un CU es dividido en dos PUs existen 6 posibilidades de particionamiento, dos PUs rectangulares simétricas divididas en forma horizontal o vertical y dos PUs rectangulares asimétricas, que son separadas de forma horizontal o vertical a un cuarto o a tres cuartas partes de distancia de la parte superior o de la parte izquierda del CU, lo cual es otra de las principales novedades de H.265/HEVC ya que posibilita la existencia de PUs de tamaño hasta 8x2 píxeles. Esto permite una mejor adaptación a las características de la imagen y, en concreto, a los bordes existentes en ella. [1][10][14]. La Figura 3.4 muestra los ocho tipos de particionamiento que pueden ser usados para definir los PUs dentro de una CU.

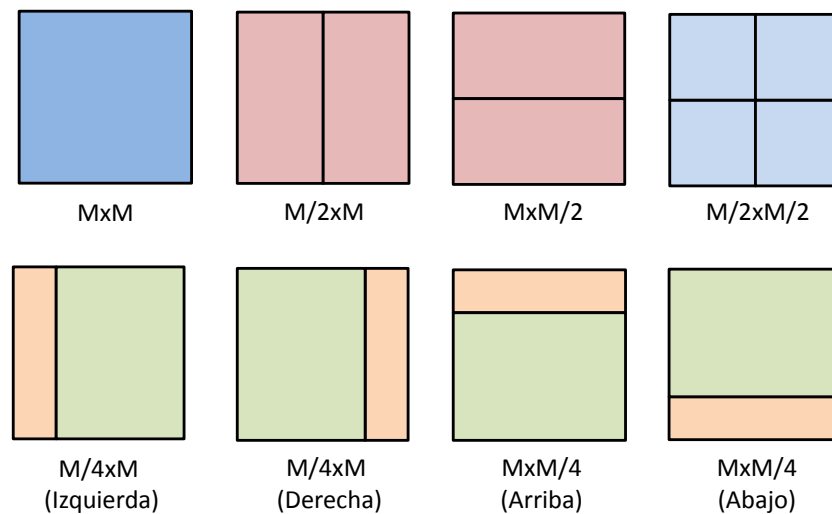


Figura 3.4 Modos de particionamiento de un CU en PUs

3.2.2.4 Unidad de Transformación TU

La unidad de transformación (TU) es una estructura cuadrada definida por el particionamiento de la CU residual, sobre la cual se realizan los procesos de transformación y cuantización. El particionamiento de un CU es incondicional cuando éste supera el tamaño máximo permitido de la TU. Por el contrario, el particionamiento no es llevado a cabo si éste genera bloques de menor tamaño que el mínimo permitido para las estructuras TU. El tamaño de la TU puede oscilar entre los 4x4 y los 32x32 píxeles. Los TUs transfieren los datos resultantes del procesamiento de los PUs hacia el módulo de la transformada, es decir transportan los datos residuales de la predicción “*inter-frame*” e “*intra-frame*”[14].

Para una CU de predicción “*inter-frame*”, las TUs que la componen pueden ser más grandes que las PUs, es decir las TUs pueden contener los límites de las PUs. Sin embargo, en una CU de predicción “*intra-frame*” las TUs no pueden superar los límites de las PUs.

Cada TU está compuesta por un “*transform block*” (TB) de luminancia y los respectivos TBs de crominancia. El tamaño de los TBs de crominancia es igual a la mitad en cada dimensión de los TBs de luminancia, excepto cuando el tamaño del TB de luminancia es 4x4, en cuyo caso un único TB de crominancia es utilizado por la región cubierta por cuatro TBs de luminancia de tamaño 4x4 [1].

La Figura 3.5 muestra CUs de tamaños 32x32, 16x16 y 8x8, los cuales son divididos en TUs de dimensiones 16x16, 8x8 y 4x4. Algunos de los bloques CUs no se subdividen, por lo tanto, los bloques TUs toman el tamaño de los bloques CUs que los contienen teniendo en cuenta que este tamaño no debe superar el máximo tamaño permitido para los bloques TUs que en este caso es de 32x32. La profundidad máxima de división en árbol de las CUs es ajustable y es especificada en la sintaxis de cabecera de la CU.

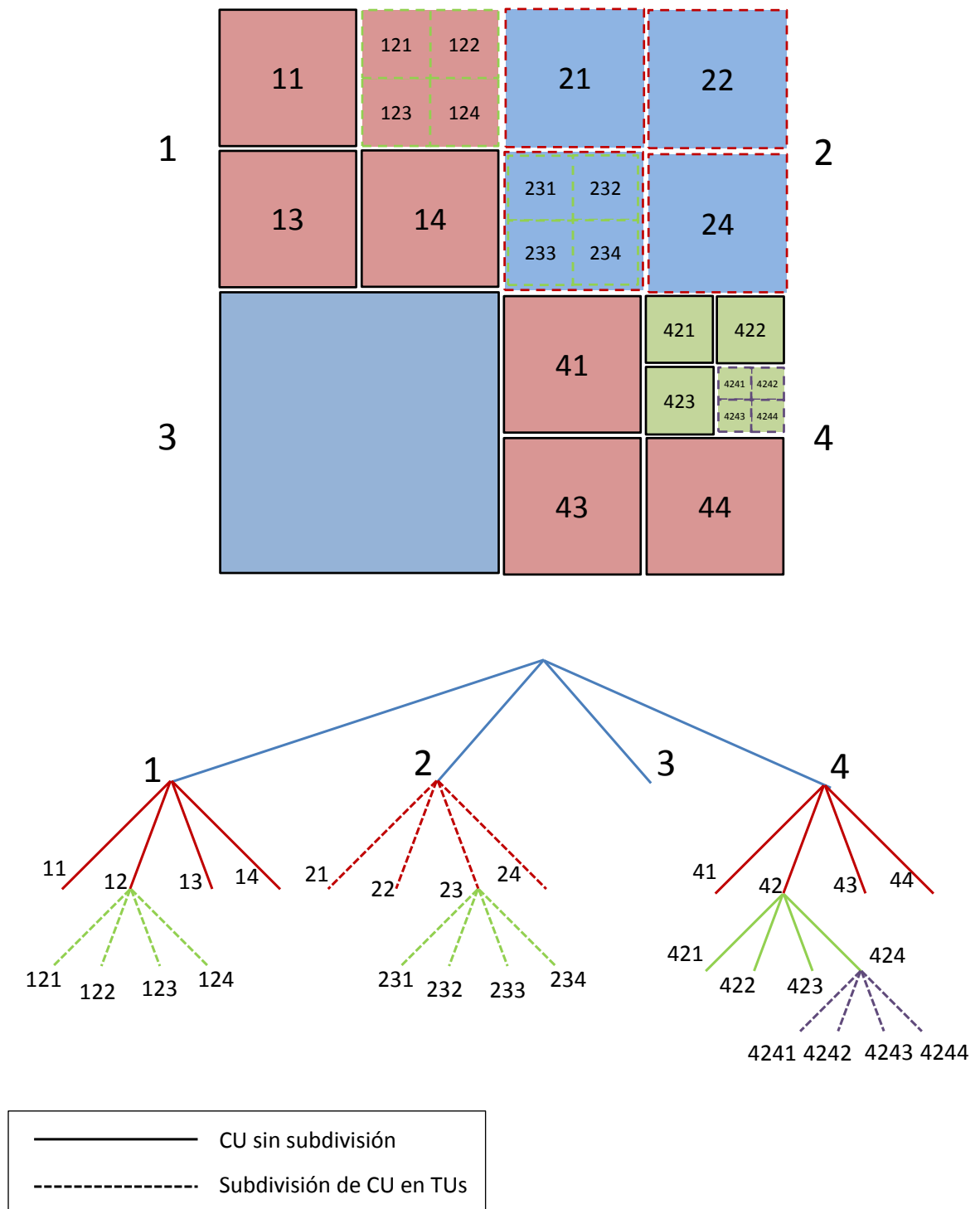


Figura 3.5 *Particionamiento de un CU en TUs*

Capítulo 4

TRANSFORMADA DCT PARA H.265/HEVC

4.1 INTRODUCCIÓN

La transformada discreta del coseno (DCT) es ampliamente utilizada en compresión de imágenes y video debido a sus propiedades inherentes tales como compactación de la energía, separabilidad, simetría y ortogonalidad. El estándar H.265/HEVC emplea la transformada DCT de tamaños 4×4 , 8×8 , 16×16 y 32×32 con el fin de duplicar la compresión de los datos con respecto al estándar H.264/AVC, reduciendo la tasa de bit a transmitir (*bit-rate*) en un rango de 50%, obteniendo la misma calidad perceptual de video y soportando resoluciones de video desde 320×240 píxeles hasta 7680×4320 píxeles (UHDTV – 8K) [1] [2].

A partir del estándar H.264/AVC se han utilizado aproximaciones enteras de las transformadas en lugar de las transformadas de precisión infinita que se usaban en estándares anteriores. Lo anterior con el fin de disminuir la incompatibilidad y asegurar la interoperabilidad entre las implementaciones de codificadores y decodificadores [16]. En el estándar H.264/AVC se especifican matrices base enteras de la transformada DCT de tamaños 4×4 y 8×8 . Las transformadas fueron diseñadas como aproximaciones enteras de la DCT haciendo énfasis en disminuir el número de operaciones aritméticas. Sin embargo estas aproximaciones presentaban grandes variaciones en la norma de los vectores base con respecto a la transformada DCT de precisión infinita. Como consecuencia de esto se especificaron matrices no planas de cuantización por defecto para poder compensar la norma de los vectores base [17].

Durante el desarrollo del estándar H.265/HEVC fueron estudiadas varias aproximaciones de la transformada DCT hasta conseguir un diseño que cumpliera con las propiedades de igualdad de norma, ortogonalidad y simetría de los vectores base de la transformada DCT de precisión infinita.

4.2 TRANSFORMADA DCT -1D

Si consideramos el caso general, matemáticamente una transformada unidimensional se realiza mediante una multiplicación sucesiva de los elementos de un vector de datos, por un conjunto de vectores base, cada uno de los cuales puede considerarse como un conjunto de factores de ponderación. La posterior suma de los valores obtenidos de esta multiplicación complementan las operaciones necesarias que dan lugar a los coeficientes transformados [18][19]. Para el caso de la transformada DCT lo dicho anteriormente puede describirse con la ecuación 4.1.

$$Y_i = \sum_{j=0}^{N-1} C_{i,j} * X_j \quad i = 0, 1, 2, 3, \dots, N-1 \quad (4.1)$$

Donde X_j representa el vector de datos de tamaño N , $C_{i,j}$ representa la matriz base de la transformada DCT-1D de tamaño $N \times N$, y Y_i representa los datos transformados.

La matriz base de la transformada DCT-1D de $N \times N$ elementos está definida por la siguiente ecuación:

$$C_{i,j} = \frac{A}{\sqrt{N}} \cos \left[\frac{\pi}{N} \left(j + \frac{1}{2} \right) i \right] \quad (4.2)$$

Donde,

$$A = \begin{cases} 1, & i = 0 \\ \sqrt{2}, & i > 0 \end{cases} \quad \text{para } i, j = 0, 1, 2, \dots, N-1 \quad (4.3)$$

Otra forma de representar el cálculo de la transformada DCT es usando algebra lineal tal y como se muestra en la ecuación 4.4.

$$\begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{N-2} \\ Y_{N-1} \end{bmatrix} = \begin{bmatrix} C_{0,0} & C_{0,1} & \dots & C_{0,N-2} & C_{0,N-1} \\ C_{1,0} & C_{1,1} & & C_{1,N-2} & C_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{N-2,0} & C_{N-2,1} & \dots & C_{N-2,N-2} & C_{N-2,N-1} \\ C_{N-1,0} & C_{N-1,1} & & C_{N-1,N-2} & C_{N-1,N-1} \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-2} \\ X_{N-1} \end{bmatrix} \quad (4.4)$$

El cálculo de la transformada DCT-1D también puede ser representado por medio de un pseudocódigo como el que se presenta en el algoritmo 1.

Algoritmo 1: Transformada DCT-1D

Entradas: $X = (X_0, X_1, \dots, X_{N-1})^T$ **Constantes:** $C(i, j)$ $0 \leq i < N$; $0 \leq j < N$ **Variables Temporales:** $M_R(i, j)$ $0 \leq i < N$; $0 \leq j < N$ **Salidas:** $Y = (Y_0, Y_1, \dots, Y_{N-1})^T$

1. *for* $i=0$ to $N-1$ *do*
 2. $Y[i] := 0$;
 3. *end for*
 4. *for* $i=0$ to $N-1$ *do*
 5. *for* $j=0$ to $N-1$ *do*
 6. $M_R[i, j] := C[i, j] * X[j]$;
 7. $Y[i] := Y[i] + M_R[i, j]$;
 8. *end for*
 9. *end for*
 10. *Return* (Y)
-

4.3 TRANSFORMADA IDCT -1D

La transformada IDCT-1D realiza el proceso de transformación inversa, recuperando el vector de datos originales a partir del vector de datos transformados. Al igual que la DCT-1D, el cálculo de esta transformada se realiza mediante una multiplicación sucesiva de los elementos de un vector de datos, por un conjunto de valores de la matriz base. En este caso la matriz base de la transformada IDCT-1D hace referencia a la traspuesta de la matriz base de la transformada DCT-1D. Lo anterior puede describirse con la ecuación 4.5.

$$X_i = \sum_{j=0}^{N-1} C_{i,j}^T * Y_j \quad i = 0, 1, 2, 3, \dots, N - 1 \quad (4.5)$$

Donde Y_i representa los datos transformados, $C_{i,j}^T$ representa la traspuesta de la matriz base de la transformada DCT-1D de tamaño $N \times N$ y X_j representa el vector de datos originales de tamaño N .

El cálculo de la transformada DCT-1D también puede ser representado por medio de un pseudocódigo como el que se presenta en el algoritmo 2.

Algoritmo 2: Transformada IDCT-1D

Entradas: $Y = (Y_0, Y_1, \dots, Y_{N-1})^T$ **Constantes:** $C(i, j), C'(i, j) \quad 0 \leq i < N; \quad 0 \leq j < N$ **Variables Temporales:** $M_R(i, j) \quad 0 \leq i < N; \quad 0 \leq j < N$ **Salidas:** $X = (X_0, X_1, \dots, X_{N-1})^T$

```
11. for i=0 to N-1 do
12.    $X[i] := 0;$ 
13. end for
14. for i=0 to N-1 do
15.   for j=0 to N-1 do
16.      $C'[i, j] := C[j, i];$ 
17.   end for
18. end for
19. for i=0 to N-1 do
20.   for j=0 to N-1 do
21.      $M_R[i, j] := C'[i, j] * Y[j];$ 
22.      $X[i] := X[i] + M_R[i, j];$ 
23.   end for
24. end for
25. Return (X)
```

4.4 TRANSFORMADA DCT-2D

Es importante señalar que muchos datos de una imagen tienen una alta correlación entre elementos en más de una dirección, y una transformada unidimensional solamente tendrá en cuenta la correlación entre elementos individuales de un vector de datos, y no entre vectores adyacentes [18][19]. Por esta razón en la compresión de imágenes se hace necesario realizar una transformada bidimensional, la cual tiene en cuenta las correlaciones de todos los datos.

La ecuación 4.6 describe en forma matricial el cálculo de la transformada DCT-2D. Donde D representa la matriz de datos originales, C representa la matriz base de la transformada DCT-1D y Y representa la matriz de datos transformados.

$$Y = CDC^T \quad (4.6)$$

Otra forma de obtener el resultado de la transformada DCT-2D es realizar dos transformaciones DCT-1D consecutivas con una etapa de trasposición intermedia. Este proceso es representado por las ecuaciones 4.7 y 4.8.

$$F_{i,j} = \sum_{j=0}^{N-1} C_{i,j} * D_{j,i} \quad i = 0, 1, 2, 3, \dots, N-1 \quad (4.7)$$

$$Y_{i,j} = \sum_{j=0}^{N-1} C_{i,j} * F_{j,i}^T \quad i = 0, 1, 2, 3, \dots, N-1 \quad (4.8)$$

Donde F representa la matriz resultante de la primera DCT-1D y F^T su traspuesta.

El cálculo de la transformada DCT-2D también puede ser representado por medio de un pseudocódigo como el que se presenta en el algoritmo 3.

Algoritmo 3: Transformada DCT-2D

Entradas: $X(i, j)$ $0 \leq i < N$; $0 \leq j < N$

Constantes: $C(i, j)$ $0 \leq i < N$; $0 \leq j < N$

Variables Temporales: $Aux(i, j), F(i, j), F'(i, j)$ $0 \leq i < N$; $0 \leq j < N$

Salidas: $Y(i, j)$ $0 \leq i < N$; $0 \leq j < N$

1. *for* $i=0$ to $N-1$ *do*
 2. *for* $j=0$ to $N-1$ *do*
 3. $Y[i, j] := 0$;
 4. $F[i, j] := 0$;
 5. $F'[i, j] := 0$;
 6. $Aux[i, j] := 0$;
 7. *end for*
 8. *end for*
 9. *for* $k=0$ to $N-1$ *do*
 10. *for* $i=0$ to $N-1$ *do*
 11. *for* $j=0$ to $N-1$ *do*
 12. $Aux[i, j] := C[i, j] * D[j, k]$;
 13. $F[i, k] := F[i, k] + Aux[i, j]$;
 14. *end for*
 15. *end for*
 16. *end for*
-

```

17. for i=0 to N-1 do
18.   for j=0 to N-1 do
19.     F'[i, j] := F[j, i];
20.   end for
21. end for
22. for k=0 to N-1 do
23.   for i=0 to N-1 do
24.     for j=0 to N-1 do
25.       Aux[i, j] := C[i, j] * F'[j, k];
26.       Y[i, k] := Y[i, k] + Aux[i, j];
27.     end for
28.   end for
29. end for
30. Return (Y)

```

4.5 TRANSFORMADA IDCT-2D

La transformada IDCT-2D se encarga de realizar el proceso de transformación inversa. Esta transformada recupera la matriz de datos originales a partir de matriz de datos transformados. La ecuación 4.9 describe en forma matricial el cálculo de la transformada IDCT-2D. Donde Y representa la matriz de datos transformados, C^T representa la traspuesta de la matriz base de la transformada DCT-1D y D representa la matriz de datos originales.

$$D = C^T Y C \quad (4.9)$$

Al igual que la DCT-2D, La IDCT-2D también es separable. En este caso esta transformada puede ser calculada realizando dos transformaciones IDCT-1D consecutivas con una etapa de trasposición intermedia. Este proceso es descrito por las ecuaciones 4.10 y 4.11.

$$G_{i,j} = \sum_{j=0}^{N-1} C_{i,j}^T * Y_{j,i} \quad i = 0, 1, 2, 3, \dots, N-1 \quad (4.10)$$

$$D_{i,j} = \sum_{j=0}^{N-1} C_{i,j}^T * G_{j,i}^T \quad i = 0, 1, 2, 3, \dots, N-1 \quad (4.11)$$

Donde G representa la matriz resultante de la primera IDCT-1D y G^T su traspuesta.

El cálculo de la transformada IDCT-2D también puede ser representado por medio de un pseudocódigo como el que se presenta en el algoritmo 4.

Algoritmo 4: Transformada IDCT-2D

Entradas: $Y(i, j)$ $0 \leq i < N$; $0 \leq j < N$

Constantes: $C(i, j), C'(i, j)$ $0 \leq i < N$; $0 \leq j < N$,

Variables Temporales: $Aux(i, j), G(i, j), G'(i, j)$ $0 \leq i < N$; $0 \leq j < N$

Salidas: $X(i, j)$ $0 \leq i < N$; $0 \leq j < N$

```

1. for i=0 to N-1 do
2.     for j=0 to N-1 do
3.         Y[i, j] := 0;
4.         F[i, j] := 0;
5.         F'[i, j] := 0;
6.         Aux[i, j] := 0;
7.         C'[i, j] := C[j, i];
8.     end for
9. end for
10. for k=0 to N-1 do
11.     for i=0 to N-1 do
12.         for j=0 to N-1 do
13.             Aux[i, j] := C'[i, j] * Y[j, k];
14.             G[i, k] := G[i, k] + Aux[i, j];
15.         end for
16.     end for
17. end for
18. for i=0 to N-1 do
19.     for j=0 to N-1 do
20.         G'[i, j] := G[j, i];
21.     end for
22. end for
23. for k=0 to N-1 do
24.     for i=0 to N-1 do
25.         for j=0 to N-1 do
26.             Aux[i, j] := C'[i, j] * G'[j, k];
27.             X[i, k] := X[i, k] + Aux[i, j];
28.         end for
29.     end for
30. end for
31. Return (X)

```

4.6 PROPIEDADES DE LA TRANSFORMADA DCT

La transformada DCT posee propiedades que son consideradas útiles tanto en la eficiencia de compresión como en su implementación. A continuación se describen cada una de ellas.

4.6.1 ORTOGONALIDAD

Los vectores base de la matriz base de la transformada DCT son ortogonales, es decir que el producto punto entre un vector base y otro vector base traspuesto es igual a cero. Lo dicho anteriormente puede describirse con la siguiente ecuación:

$$C_i \cdot C_j^T = 0 \text{ para } i \neq j \quad (4.12)$$

La propiedad de ortogonalidad es útil debido a que al obtener coeficientes de transformación no correlacionados se incrementa la eficiencia de compresión del codificador.

4.6.2 NORMALIDAD

Los vectores base de la matriz base de la transformada DCT poseen igual norma, es decir que el producto punto entre un vector base y su traspuesta será igual al producto punto entre otro vector base y su traspuesta. Esto puede describirse con la siguiente ecuación:

$$C_i \cdot C_i^T = C_j \cdot C_j^T \text{ para } i \neq j \quad (4.13)$$

La propiedad de normalidad es de gran utilidad para la simplificación del proceso de cuantización y cuantización inversa ya que elimina la necesidad de matrices de cuantización.

4.6.3 RECURSIVIDAD

Siendo $N = 2^M$ y M un número entero, los elementos de la matriz base de la transformada DCT de tamaño $2^M \times 2^M$ son un subconjunto de los elementos de la matriz base de la

transformada DCT de tamaño $2^{M+1} \times 2^{M+1}$. De forma más específica, los vectores base de una matriz base menor son iguales a la primera mitad de los vectores base pares de la matriz base inmediatamente mayor.

La propiedad de recursividad ofrece la posibilidad de reutilizar bloques funcionales reduciendo los costos de implementación [16].

4.6.4 SIMETRÍA Y ANTI-SIMETRÍA

Los vectores base pares de la matriz base de la transformada DCT poseen la propiedad de simetría, mientras que los vectores base impares poseen la propiedad de anti-simetría. Estas propiedades permiten obtener relaciones trigonométricas con el fin de reducir el número de operaciones aritméticas y así obtener algoritmos rápidos como el algoritmo de Chen [20].

4.7 TRANSFORMADA DCT-2D PARA H.265/HEVC

En el codificador híbrido H.265/HEVC, la transformada DCT se encarga de transformar los datos residuales de la diferencia entre los píxeles de la imagen de entrada y los valores de predicción obtenidos a partir de la predicción “*inter-frame*” o “*intra-frame*”. Los datos residuales son divididos en bloques cuadrados de tamaño $N \times N$ llamados TUs, donde $N = 2^M$ y M es un número entero [21]. Los $N \times N$ datos de los bloques TUs son transformados por medio de la transformada discreta del coseno bidimensional (DCT-2D), la cual concentra la energía de la imagen en unos pocos coeficientes numéricos. Posteriormente, los coeficientes transformados son cuantificados con el fin de eliminar la mayoría de los coeficientes de menor valor, y finalmente el codificador de entropía comprime esta información.

Las matrices base de la transformada DCT-1D usadas en H.265/HEVC son aproximaciones de precisión finita de las matrices base de la transformada DCT-1D de precisión infinita definidas por la ecuación 4.2. El beneficio del uso de aproximaciones de precisión finita en un estándar de codificación de vídeo es que la aproximación a los valores reales de las matrices base de la DCT está especificada por el estándar en lugar de ser dependiente de la implementación. Esto

evita las incompatibilidades entre codificadores y decodificadores causadas por los fabricantes que utilizan diferentes representaciones de punto flotante. Por otro lado, la desventaja de utilizar aproximaciones de precisión finita es que algunas de las propiedades de la transformada DCT discutidas anteriormente pueden no cumplirse. Más específicamente es necesario establecer un equilibrio entre el costo computacional asociado al uso de una mayor cantidad de bits para representar los elementos de la matriz base y el grado en el que algunas de las propiedades de la DCT se cumplan.

Un método sencillo para determinar aproximaciones enteras de los elementos de la matriz base de la transformada DCT es escalar cada elemento con un número de gran magnitud (típicamente con un valor entre 2^5 y 2^{16}) y luego redondear el resultado al entero más cercano. Sin embargo, este enfoque no necesariamente ofrece el mejor rendimiento de compresión.

Las aproximaciones de los elementos de la matriz base de la transformada DCT realizadas en el estándar H.265/HEVC se establecieron de tal forma que se cumplan incondicionalmente las propiedades de recursividad, simetría y anti-simetría de la transformada DCT. Además se estableció un equilibrio entre el número de bits utilizados para representar cada elemento de la matriz base y el cumplimiento de las propiedades de ortogonalidad y normalidad.

Después de una ardua investigación, en H.265/HEVC se decidió representar cada elemento de la matriz base de la transformada DCT con 8 bits (incluyendo el bit de signo) y se determinó un factor de escalamiento igual a $2^{6+(M/2)}$. A partir de la matriz base escalada se realizaron algunos ajustes en donde se mantuvo el valor de 64 para todos los elementos del primer vector base y se modificaron manualmente el resto de los elementos de la matriz base para conseguir un mejor resultado en las propiedades de ortogonalidad y normalidad [16].

Las matrices base de las transformadas DCT de tamaños 4x4, 8x8, 16x16 y 32x32 obtenidas después del proceso descrito anteriormente se presentan en las Tablas 4.1, 4.2, 4.3 y 4.4, respectivamente.

Debido a que los vectores base aproximados tienen casi la misma norma es posible utilizar el mismo factor de escalamiento para todos los coeficientes transformados en la etapa de cuantización y cuantización inversa. Además, este factor es el mismo para todos los tamaños de la transformada DCT, excepto por los desplazamientos a la derecha los cuales si dependen del tamaño de la transformada. Esto elimina las matrices de cuantización de gran tamaño que se utilizaban en estándares previos como H.264/AVC.

Adicionalmente al cumplir con las propiedades de recursividad, es posible implementar las transformadas DCT de mayor tamaño reutilizando bloques de las transformadas de menor tamaño reduciendo los costos de implementación. Además gracias a las propiedades de simetría y anti-simetría es posible obtener un algoritmo rápido para calcular la transformada DCT-1D basado en diagramas mariposa, reduciendo la cantidad de operaciones aritméticas y mejorando el rendimiento de la transformada [21].

Por otra parte, dado que las matrices base de la transformada DCT están escaladas por un factor de $2^{6+(M/2)}$ en comparación con una matriz base de la transformada DCT de precisión infinita, y con el fin de preservar la norma del bloque residual después de realizar la transformada DCT-2D y la transformada IDCT-2D, se establecen factores de escalamiento adicionales, denominados ST1, ST2, SIT1 y SIT2. Estos factores reducen el ancho de bits de los resultados intermedios después de cada transformada DCT-1D o IDCT-1D.

64	64	64	64
83	36	-36	-83
64	-64	64	-64
36	-83	83	-36

Tabla 4.1 Matriz base de la transformada DCT 4x4 para H.265/HEVC

64	64	64	64	64	64	64	64
89	75	50	18	-18	-50	-75	-89
83	36	-36	-83	-83	-36	36	83
75	-18	-89	-50	50	89	18	-75
64	-64	-64	64	64	-64	-64	64
50	-89	18	75	-75	-18	89	-50
36	-83	83	-36	-36	83	-83	36
18	-50	75	-89	89	-75	50	-18

Tabla 4.2 Matriz base de la transformada DCT 8x8 para H.265/HEVC

64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
90	87	80	70	57	43	25	9	-9	-25	-43	-57	-70	-80	-87	-90
89	75	50	18	-18	-50	-75	-89	-89	-75	-50	-18	18	50	75	89
87	57	9	-43	-80	-90	-70	-25	25	70	90	80	43	-9	-57	-87
83	36	-36	-83	-83	-36	36	83	83	36	-36	-83	-83	-36	36	83
80	9	-70	-87	-25	57	90	43	-43	-90	-57	25	87	70	-9	-80
75	-18	-89	-50	50	89	18	-75	-75	18	89	50	-50	-89	-18	75
70	-43	-87	9	90	25	-80	-57	57	80	-25	-90	-9	87	43	-70
64	-64	-64	64	64	-64	-64	64	64	-64	-64	64	64	-64	-64	64
57	-80	-25	90	-9	-87	43	70	-70	-43	87	9	-90	25	80	-57
50	-89	18	75	-75	-18	89	-50	-50	89	-18	-75	75	18	-89	50
43	-90	57	25	-87	70	9	-80	80	-9	-70	87	-25	-57	90	-43
36	-83	83	-36	-36	83	-83	36	36	-83	83	-36	-36	83	-83	36
25	-70	90	-80	43	9	-57	87	-87	57	-9	-43	80	-90	70	-25
18	-50	75	-89	89	-75	50	-18	-18	50	-75	89	-89	75	-50	18
9	-25	43	-57	70	-80	87	-90	90	-87	80	-70	57	-43	25	-9

Tabla 4.3 Matriz base de la transformada DCT 16x16 para H.265/HEVC

[illegible]

Tabla 4.4 Matriz base de la transformada DCT 32x32 para H.265/HEVC

En el estándar H-265/HEVC los valores de ST1, ST2, SIT1 y SIT2 fueron determinados teniendo en cuenta las siguientes restricciones:

1. Todos los factores de escalamiento serán una potencia de dos, con el fin de permitir su implementación mediante desplazamientos a la derecha.
2. Asumiendo una matriz de datos de entrada con todos sus elementos iguales a su valor máximo, el número de bits para representar los resultados de la transformación después de cada una de las dos DCT-1D será igual a 16 (incluyendo el bit de signo). Esto fue considerado como un balance razonable entre precisión y costo de implementación.
3. Dado que las matrices base de la transformada DCT se escalan por un factor de $2^{6+(M/2)}$, la conexión en cascada de las transformadas DCT-1D en la implementación de las transformadas DCT-2D y IDCT-2D ocasiona un factor de escalamiento de $2^{6+(M/2)}$ después de que cada transformada DCT-1D es aplicada tanto a las filas como a las columnas de la imagen de entrada, y de $2^{6+(M/2)}$ después de que cada transformada IDCT-1D es aplicada tanto a las filas como a las columnas de la imagen transformada. En consecuencia para preservar la norma en las transformadas DCT-2D y IDCT-2D será necesario un factor de escalamiento de 2^{-24-2M} .

Después de realizar un análisis teniendo en cuenta el peor de los casos posibles para una imagen de entrada, H.265/HEVC determinó los siguientes valores para ST1, ST2, SIT1 y SIT2:

- Después de la primera etapa de transformación directa DCT-1D: $ST1 = 2^{-(B+M-9)}$
- Después de la segunda etapa de transformación directa DCT-1D: $ST2 = 2^{-(M+6)}$
- Después de la primera etapa de transformación inversa IDCT-1D: $SIT1 = 2^{-7}$
- Después de la segunda etapa de transformación inversa IDCT-1D: $SIT2 = 2^{-(20-B)}$

Donde B representa el *bit depth* de los datos de entrada. Por ejemplo con un *bit depth* = 8, los datos de entrada estarán en un rango de $(-2^B + 1, 2^B - 1)$, requiriendo de B+1 bits para ser representados.

Al usar la representación de 8 bits para los elementos de las matrices base y limitar el número de bits o el ancho de los datos intermedios a 16, todas las operaciones de multiplicación pueden ser calculadas usando multiplicadores de 16 bits o menos, y los acumuladores antes del escalamiento pueden ser implementados con menos de 32 bits para todos los tamaños de la transformada DCT [16][22].

El estándar H.265/HEVC también define algunos valores de compensación o redondeo con el fin de mejorar la precisión de los resultados, estos valores deben sumarse a los datos resultantes después de cada etapa de transformación y antes del proceso de escalamiento mencionado anteriormente.

Los valores de compensación definidos en el estándar H.265/HEVC son los siguientes:

- Después de la primera etapa de transformación directa DCT-1D: $\text{offset1} = 2^{(B+M-10)}$
- Después de la segunda etapa de transformación directa DCT-1D: $\text{offset2} = 2^{(M+5)}$
- Después de la primera etapa de transformación inversa IDCT-1D: $\text{offsetI1} = 2^6$
- Después de la segunda etapa de transformación inversa IDCT-1D: $\text{offsetI2} = 2^{(19-B)}$

El pseudocódigo de las transformadas DCT-2D y IDCT-2D para el estándar HEVC, se muestran en los algoritmos 5 y 6, respectivamente.

Cabe resaltar que en estos algoritmos se incluyen los procesos de redondeo y escalamiento establecidos por el estándar HEVC. Para el cálculo de los factores de redondeo y escalamiento se utiliza la función *SHL (Shift-Left)* que hace referencia a la función de desplazamiento binario hacia la izquierda. Además, es importante tener en cuenta que en estos algoritmos se usan las funciones DCT_1D y IDCT_1D, las cuales hacen referencia a las transformaciones unidimensionales utilizadas en el cálculo de las transformadas DCT-2D y IDCT-2D, respectivamente. El pseudocódigo de estas funciones puede ser determinado a partir de los algoritmos 3 y 4.

Finalmente, en la Figura 4.1 se puede observar un diagrama de bloques en donde se muestran las diferentes etapas de redondeo y escalamiento llevadas a cabo después de cada transformada DCT-1D y IDCT-1D en el proceso de transformación DCT-2D e IDCT-2D, respectivamente.

Algoritmo 5: Transformada DCT-2D para HEVC

Entradas: $X(i, j)$ $0 \leq i < N$; $0 \leq j < N$

Constantes: $offset1, offset2, ST1, ST2$.

Pre-cálculo: $M = \log_2 N$

Variables Temporales: $F(i, j), F'(i, j)$ $0 \leq i < N$; $0 \leq j < N$

Salidas: $Y(i, j)$ $0 \leq i < N$; $0 \leq j < N$

1. $offset1 := SHL(1, B+M-10);$
 2. $ST1 := SHL(1, B+M-9);$
 3. $offset2 := SHL(1, M+5);$
 4. $ST2 := SHL(1, M+6);$

 5. $F := DCT_1D(X);$
 6. *for* $i=0$ *to* $N-1$ *do*
 7. *for* $j=0$ *to* $N-1$ *do*
 8. $F[i, j] := (F[i, j] + offset1) / ST1;$
 9. *end for*
 10. *end for*
 11. *for* $i=0$ *to* $N-1$ *do*
 12. *for* $j=0$ *to* $N-1$ *do*
 13. $F'[i, j] := F[j, i];$
 14. *end for*
 15. *end for*
 16. $Y := DCT_1D(F');$
 17. *for* $i=0$ *to* $N-1$ *do*
 18. *for* $j=0$ *to* $N-1$ *do*
 19. $Y[i, j] := (Y[i, j] + offset2) / ST2;$
 20. *end for*
 21. *end for*
 22. *Return* (Y)
-

Algoritmo 6: Transformada IDCT-2D para HEVC

Entradas: $Y(i, j)$ $0 \leq i < N$; $0 \leq j < N$

Constantes: $offset11$, $offset12$, $SIT1$, $SIT2$.

Pre-cálculo: $M = \log_2 N$

Variables Temporales: $G(i, j)$, $G'(i, j)$ $0 \leq i < N$; $0 \leq j < N$

Salidas: $X(i, j)$ $0 \leq i < N$; $0 \leq j < N$

1. $offset11 := SHL(1, 6)$;
 2. $SIT1 := SHL(1, 7)$;
 3. $offset12 := SHL(1, 19-B)$;
 4. $SIT2 := SHL(1, 20-B)$;
-
5. $G := IDCT_1D(Y)$;
 6. **for** $i=0$ **to** $N-1$ **do**
 7. **for** $j=0$ **to** $N-1$ **do**
 8. $G[i, j] := (G[i, j] + offset11) / SIT1$;
 9. **end for**
 10. **end for**
 11. **for** $i=0$ **to** $N-1$ **do**
 12. **for** $j=0$ **to** $N-1$ **do**
 13. $G'[i, j] := G[j, i]$;
 14. **end for**
 15. **end for**
 16. $X := IDCT_1D(G')$;
 17. **for** $i=0$ **to** $N-1$ **do**
 18. **for** $j=0$ **to** $N-1$ **do**
 19. $X[i, j] := (X[i, j] + offset12) / SIT2$;
 20. **end for**
 21. **end for**
 22. **Return** (X)
-

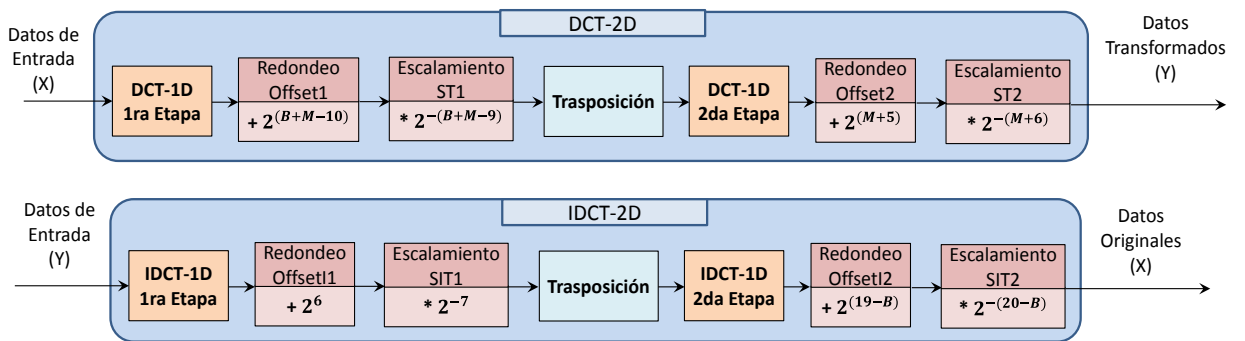


Figura 4.1 Factores de escalamiento para las transformadas DCT-2D e IDCT-2D en H.265/HEVC

Resumiendo todo lo anterior, el diseño de la transformada DCT de precisión finita establecido por el estándar H.265/HEVC presenta las siguientes características.

- Resultados muy cercanos a los presentados por la DCT de precisión infinita
- Norma entre vectores base casi igual.
- Vectores base casi ortogonales.
- Recursividad igual a la DCT de precisión infinita
- Simetría y antisimetría igual a la DCT de precisión infinita
- Representación de 8 bits para los elementos de la matriz base de la transformada
- Numero de bits para los acumuladores de la multiplicación matricial inferior a 32.
- Numero de bits igual a 16 para todos los *buffers* de la transformada.
- Multiplicadores de 16 bits para todos los cálculos internos.

Capítulo 5

DISEÑO HARDWARE DE LA TRANSFORMADA DCT PARA H.265/HEVC

5.1 INTRODUCCIÓN

Usualmente los estándares de codificación de video tradicionales seleccionan un tamaño de 4x4 o 8x8 para la transformada DCT-2D con el fin de equilibrar la eficiencia de codificación y el costo de implementación. El estándar H.265/HEVC utiliza la transformada DCT-2D de tamaños 4x4, 8x8, 16x16 y 32x32 con el fin de duplicar la eficiencia de compresión y aumentar la resolución de video soportada con respecto a estándares anteriores como H.264/AVC. Sin embargo, esto produce un incremento exponencial en la complejidad computacional cuando se implementa en software y un incremento en los costos de área y potencia, cuando se diseña en hardware.

En este capítulo, con el propósito de mitigar las desventajas mencionadas, se presentan tres diseños eficientes en hardware para la transformada DCT-2D tanto directa como inversa, en donde las características principales de los diseños propuestos son su gran velocidad de procesamiento y su disminución en el consumo de área y potencia. Además se presenta una arquitectura hardware parametrizable de la transformada DCT-2D tanto directa como inversa que incluye a todos los tamaños de la transformada en un solo núcleo. Estos diseños fueron implementados en hardware teniendo en cuenta las características de la transformada DCT-2D utilizada en el estándar H.265/HEVC, las cuales fueron mencionadas en el capítulo anterior.

Este capítulo está organizado de la siguiente manera: En la sección 5.2 se presenta el estado del arte de las implementaciones en hardware de las transformadas DCTs en donde se incluyen trabajos tanto de transformadas directas como inversas ya sean unidimensionales o

bidimensionales. En la sección 5.3 se describen tres diseños hardware de la transformada DCT-1D para el estándar H.265/HEVC. En la sección 5.4 se describen los diseños hardware de la transformada DCT-2D, en donde se utilizan los diseños hardware descritos en la sección 5.3. En la sección 5.5 se describen tres diseños hardware para la transformada IDCT-1D para el estándar H.265/HEVC. En la sección 5.6 se describen los diseños hardware de la transformada IDCT-2D, los cuales utilizan las transformadas diseñadas en la sección 5.5. En la sección 5.7 se describe la implementación en hardware de la transformada DCT-1D de múltiples tamaños. En la sección 5.8 se describe el diseño hardware de la transformada DCT-2D de múltiples tamaños en donde se utiliza el diseño descrito en la sección 5.7. En la sección 5.9 se describe el diseño hardware de la transformada IDCT-1D de múltiples tamaños y finalmente, en la sección 5.10 se describe el diseño hardware de la transformada IDCT-2D de múltiples tamaños en donde se utiliza el diseño descrito en la sección 5.9. Todas las arquitecturas hardware mencionadas fueron descritas en VHDL en forma estructural usando Quartus II versión 13.0, y las operaciones de suma y resta utilizadas en los diseños descritos en este capítulo se implementaron usando la mega-función *lpm_parallel_add* de Altera.

5.2 ESTADO DEL ARTE

La transformada DCT ha sido ampliamente utilizada en muchos estándares de compresión de video e imágenes, debido a su gran eficiencia en la compresión de datos, es por tal razón que su implementación tanto en software como en hardware es un tema de investigación de gran precedencia, sin embargo, el nuevo estándar de codificación de video H.265/HEVC emplea nuevos parámetros para el diseño de la transformada DCT tales como, mayor tamaño de la transformada, mayor precisión, diferentes factores de escalamiento, entre otros. Esto hace que el diseño de arquitecturas hardware para la transformada DCT actualmente se enfoque en satisfacer los requerimientos del estándar H.265/HEVC. Debido a la reciente aprobación del estándar H.265/HEVC (Primera versión-Abril/2013, Tercera versión-Febrero/2015) existen pocos trabajos en donde se presenten implementaciones en hardware de la transformada DCT que cumplan con las especificaciones del estándar.

Trabajos recientes se han enfocado en cumplir con los requerimientos del estándar H.265/HEVC basándose en arquitectura diseñadas para estándares anteriores tales como H.264/AVC. Uno de estos trabajos se presenta en [23], el cual describe el diseño de una arquitectura sistólica para la transformada DCT-2D de tamaños 4x4 hasta 32x32. Sin embargo este diseño emplea una gran cantidad de recursos hardware debido al uso de un gran número de multiplicadores y presenta una gran latencia. Otro trabajo con este enfoque se presenta en [24], en donde se describe el diseño de la transformada IDCT para el estándar H.264/AVC y para el estándar H.265/HEVC. Este diseño es específico y aunque disminuye el uso de recursos hardware al reemplazar los multiplicadores por sumas y desplazamientos, no soporta los tamaños 16 y 32 de la transformada IDCT, los cuales hacen referencia a requerimientos importantes del estándar H.265/HEVC.

El cálculo de la transformada DCT puede ser considerado como un problema de multiplicación de matrices, el cual puede ser implementado mediante bloques de multiplicación por constantes (MCM: *Multiple Constant Multiplication*) y unidades de suma. Trabajos recientes presentados en [25][26] utilizan bloques MCM basados en operaciones de suma y desplazamiento en lugar de multiplicadores con el fin de reducir los recursos de hardware y el tiempo de procesamiento. Sin embargo este enfoque puede reducir la frecuencia de operación debido a que el gran número de sumas consecutivas utilizadas incrementan el tiempo de retardo del diseño. En [25] también se propone una arquitectura recortada en donde se truncan los bits menos significativos de los resultados intermedios de la transformada. Esto con el fin de reducir los recursos de hardware utilizando bloques funcionales de menor tamaño pero a costa de un decremento en la precisión y en la capacidad de compresión de la transformada DCT.

Otro de los enfoques empleados en la implementación hardware de la transformada DCT es la reutilización de bloques funcionales mediante la multiplexación de señales. Uno de los trabajos que usa este enfoque es [27], en donde se presenta una arquitectura para la DCT-2D de tamaño 16x16, compuesta por dos bloques DCT-1D y un bloque de transposición basado en un arreglo de registros. Para la implementación de la transformada DCT-1D se utilizan sumas y desplazamientos, y se reutilizan bloques de transformadas DCT-1D de menor tamaño. En [28] se presenta una arquitectura de la transformada IDCT-2D de tamaños 4x4,

8x8, 16x16 y 32x32 en donde se reutilizan los bloques de multiplicación por constantes, los cuales están basados en sumas y desplazamientos. Al igual que el enfoque clásico de separabilidad de la transformada DCT-2D este diseño está compuesto por dos bloques IDCT-1D y un bloque de trasposición basado en 2 memorias RAM de 1 puerto. En [29] se presenta una arquitectura hardware de la IDCT-2D compuesta por un bloque IDCT-1D y un bloque de trasposición. EL bloque IDCT-1D es realimentado después de la trasposición evitando el uso de un segundo bloque IDCT-1D. También, este trabajo presenta una duplicación de los bloques anteriores, obteniendo un procesamiento simultáneo de dos matrices. El enfoque de reutilización de bloques funcionales es un buen método para disminuir el consumo de área, sin embargo esto reduce la velocidad de procesamiento e incrementa la latencia del diseño.

Un enfoque diferente se presenta en [30] y [31], en donde se presentan las arquitecturas hardware de la transformada IDCT-2D de tamaño 16x 16 y 32x32, respectivamente. Los diseños utilizan sumas y desplazamientos en lugar de multiplicadores con el fin de reducir los recursos de hardware. Además se realizan varias agrupaciones y factorizaciones de las ecuaciones de la transformada IDCT con el fin de disminuir el número de operaciones realizadas. Las arquitecturas son puramente combinacionales y están descritas en VHDL comportamental. Sin embargo este enfoque no permite obtener una arquitectura estructurada, lo cual impide el uso de técnicas de diseño para el mejoramiento del desempeño como por ejemplo la técnica de pipeline. Además, este tipo de descripción otorga la libertad al compilador para realizar el mapeo y síntesis, lo cual puede provocar un uso ineficiente de los recursos del dispositivo, en este caso del FPGA.

Otro de los enfoques unifica todos los tamaños de la transformada DCT en una sola arquitectura aprovechando la propiedad de recursividad de la transformada DCT. Esto con el fin de evitar el uso de arquitecturas individuales para cada tamaño de la transformada y así disminuir los recursos de hardware. Sin embargo, este enfoque hace que sea necesario esperar a que el bloque de transformación termine de procesar una matriz antes de recibir la siguiente, eliminando la posibilidad de procesar en paralelo matrices de diferentes tamaños. Algunos trabajos que utilizan este enfoque son [32]-[37].

Otro enfoque utilizado para el diseño de la transformada DCT es el uso de rotaciones de givens y esquemas lifting [38].

5.3 DISEÑO HARDWARE DE LA TRANSFORMADA DCT-1D

En el codificador híbrido H.265/HEVC, la transformada DCT se encarga de transformar los datos residuales resultantes de la diferencia entre los datos de la imagen de entrada y la información de predicción “*Intra/Inter-Frame*”. Los datos residuales son divididos en bloques cuadrados de tamaño $N \times N$ llamados TUs, donde $N = 2^M$ y M es un número entero [21]. Los $N \times N$ datos de los bloques TUs son transformados por medio de la transformada DCT-2D, la cual puede ser implementada en forma separada utilizando una transformación unidimensional sobre cada una de las columnas y posteriormente sobre cada una de las filas de la matriz de datos.

Como se mencionó en el capítulo 4, la transformada DCT-1D, usando álgebra lineal, se puede considerar como la multiplicación entre la matriz base de la transformada DCT-1D y un vector de datos de entrada, tal como se define en la siguiente ecuación:

$$\begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{N-2} \\ Y_{N-1} \end{bmatrix} = \begin{bmatrix} C_{0,0} & C_{0,1} & \dots & C_{0,N-2} & C_{0,N-1} \\ C_{1,0} & C_{1,1} & \dots & C_{1,N-2} & C_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{N-2,0} & C_{N-2,1} & \dots & C_{N-2,N-2} & C_{N-2,N-1} \\ C_{N-1,0} & C_{N-1,1} & \dots & C_{N-1,N-2} & C_{N-1,N-1} \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-2} \\ X_{N-1} \end{bmatrix} \quad (5.1)$$

Donde, $X[0 \dots N-1]$ representa el vector de datos de entrada de tamaño N , $C[0 \dots N-1, 0 \dots N-1]$ representa la matriz base de la transformada DCT-1D de tamaño $N \times N$, y $Y[0 \dots N-1]$ es el vector resultante de tamaño N .

Una de las propiedades de la matriz base de la transformada DCT, mencionadas en el capítulo anterior, es la simetría y antisimetría de sus vectores base pares e impares, respectivamente. Esta propiedad permite reducir el número de operaciones aritméticas mediante la descomposición de las matrices base en matrices de menor tamaño, tal como se muestra a continuación:

$$Y_{even} = \begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ \vdots \\ Y(N-4) \\ Y(N-2) \end{bmatrix} = \begin{bmatrix} C_{0,0} & C_{0,1} & \dots & C_{0,\frac{N}{2}-1} \\ C_{2,0} & C_{2,1} & \dots & C_{2,\frac{N}{2}-1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-4,0} & C_{N-4,1} & \dots & C_{N-2,\frac{N}{2}-1} \\ C_{N-2,0} & C_{N-2,1} & \dots & C_{N-4,\frac{N}{2}-1} \end{bmatrix} \begin{bmatrix} X_0 + X_{N-1} \\ X_1 + X_{N-2} \\ \vdots \\ X_{\frac{N}{2}-2} + X_{N-\frac{N}{2}-1} \\ X_{\frac{N}{2}-1} + X_{N-\frac{N}{2}} \end{bmatrix} \quad (5.2)$$

$$Y_{even} = \begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ \vdots \\ Y(N-4) \\ Y(N-2) \end{bmatrix} = C_{N/2} \begin{bmatrix} a(0) \\ a(1) \\ a(2) \\ \vdots \\ a(N/2-2) \\ a(N/2-1) \end{bmatrix} \quad (5.3)$$

$$Y_{odd} = \begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ \vdots \\ Y(N-3) \\ Y(N-1) \end{bmatrix} = \begin{bmatrix} C_{1,0} & C_{1,1} & \dots & C_{1,\frac{N}{2}-1} \\ C_{3,0} & C_{3,1} & \dots & C_{3,\frac{N}{2}-1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-3,0} & C_{N-3,1} & \dots & C_{N-3,\frac{N}{2}-1} \\ C_{N-1,0} & C_{N-1,1} & \dots & C_{N-1,\frac{N}{2}-1} \end{bmatrix} \begin{bmatrix} X_0 - X_{N-1} \\ X_1 - X_{N-2} \\ \vdots \\ X_{\frac{N}{2}-2} - X_{N-\frac{N}{2}-1} \\ X_{\frac{N}{2}-1} - X_{N-\frac{N}{2}} \end{bmatrix} \quad (5.4)$$

$$Y_{odd} = \begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ \vdots \\ Y(N-3) \\ Y(N-1) \end{bmatrix} = W_{N/2} \begin{bmatrix} b(0) \\ b(1) \\ b(2) \\ \vdots \\ b(N/2-2) \\ b(N/2-1) \end{bmatrix} \quad (5.5)$$

Donde,

$$a(i) = X(i) + X(N-1-i) \quad \text{para } i = 0, 1, 2 \dots \frac{N}{2}-1 \quad (5.6)$$

$$b(i) = X(i) - X(N-1-i) \quad \text{para } i = 0, 1, 2 \dots \frac{N}{2}-1 \quad (5.7)$$

En este caso, $C_{N/2}$ representa la matriz base de la transformada DCT de tamaño inmediatamente menor o en otras palabras, la matriz base de la transformada DCT-1D de $N/2$ -puntos, la cual siempre estará constituida por la primera mitad de los vectores base

pares de la matriz base de la transformada DCT-1D de N-puntos. Por otra parte, $W_{N/2}$ representa una matriz de tamaño $N/2 \times N/2$ la cual está constituida por la primera mitad de los vectores base impares de la matriz base de la transformada DCT-1D de N-puntos y está definida por la siguiente ecuación:

$$W_{N/2}(i, j) = C_N(2i + 1, j), \text{ para } 0 \leq i, j \leq \frac{N}{2} - 1 \quad (5.8)$$

Note que la ecuación 5.3 puede ser calculada usando la misma descomposición recursiva, es decir $C_{N/2}$ puede ser descompuesta obteniendo las matrices $C_{N/4}$ y $W_{N/4}$. Este enfoque puede ser utilizado para todos los tamaños de la transformada DCT-1D.

De lo anterior es posible deducir que la transformada DCT-1D de N-puntos puede ser implementada en hardware utilizando: un bloque de diagrama mariposa parcial, $N/2$ bloques de multiplicación por múltiples constantes (MCM: *Multiple Constant Multiplication*), $N/2$ bloques sumadores, y una transformada DCT-1D de $N/2$ -puntos.

- ❖ **Bloque de diagrama mariposa parcial:** Este bloque se compone $N/2$ sumadores de dos entradas de n bits para generar los valores del vector $a(i)$, y $N/2$ restadores de dos entradas de n bits para generar los valores del vector $b(i)$ de acuerdo con las ecuaciones 5.6 y 5.7, respectivamente.
- ❖ **Bloques MCMs** Estos bloques se encargan de multiplicar cada valor de del vector $b(i)$ por un vector de constantes, el cual está compuesto por los valores únicos de la matriz $W_{N/2}$. Estos valores se pueden observar en la Tabla 5.1.

Cabe resaltar que el cálculo de las ecuaciones 5.3 y 5.5 implica el uso de 340 multiplicadores para el caso de la transformada DCT-1D de 32 puntos. Para evitar el uso de esta gran cantidad de multiplicadores y reducir los costos de área, los bloques MCM se implementan utilizando sumas y desplazamientos.

<i>DCT-1D de N puntos</i>	<i>Valores Únicos de la Matriz $W_{N/2}$</i>
4	83, 36
8	89, 75, 50, 18
16	90, 87, 80, 70, 57, 43, 25, 9
32	90, 90, 88, 85, 82, 78, 73, 67, 61, 54, 46, 38, 31, 22, 13, 4

Tabla 5.1 Valores Únicos de la matriz $W_{N/2}$ para la transformada DCT-1D de N-puntos

- ❖ **Bloques sumadores:** Estos bloques permiten sumar los resultados de los bloques MCMs para generar las salidas impares de la transformada DCT-1D de N-puntos de acuerdo con la ecuación 5.5.
- ❖ **Transformada DCT-1D de N/2-puntos:** Esta transformada procesa los datos del vector $a(i)$ para generar las salidas pares de la transformada DCT-1D de N-puntos de acuerdo con la ecuación 5.3. En este caso, mientras que los datos del vector $b(i)$ son procesados por los bloques MCMs y los bloques sumadores de salida, los datos del vector $a(i)$ son procesados en paralelo por el bloque DCT-1D de N/2-puntos, de esta forma las salidas tanto pares como impares de la transformada DCT-1D de N-puntos son obtenidas en el mismo instante de tiempo.

De acuerdo con las especificaciones del estándar H.265/HEVC, el ancho de los datos de salida después de una etapa de transformación no puede superar los 16 bits, es por esta razón que las salidas de la transformada DCT-1D de N-puntos deben pasar por un proceso de redondeo y escalamiento. Los factores de redondeo y escalamiento dependen del tamaño de la transformada y de la posición en donde se emplea la DCT-1D, es decir en la primera o segunda etapa de transformación; estos factores fueron mencionados en la sección 4.5 del capítulo 4.

La implementación en hardware de la etapa de redondeo y escalamiento no se presenta en esta sección, pero se abordará en detalle en el diseño de la transformada DCT-2D.

A continuación se describen tres diseños de la transformada DCT-1D de N-puntos, los cuales utilizan diferentes técnicas de diseño y diferentes enfoques para la implementación en hardware de los bloques MCMs y los bloques sumadores.

5.3.1 TRANSFORMADA DCT-1D ESTRUCTURADA DE N-PUNTOS

El diseño de la transformada DCT-1D estructurada (DCT-1D-E) se lleva a cabo utilizando los cuatro bloques funcionales mencionados anteriormente. Por lo tanto a continuación solo se describe el diseño de los bloques MCMs y los bloques sumadores, siendo estos dos bloques los que presentan el mayor retardo del sistema y hacen parte de la ruta crítica de este tipo de arquitecturas.

- ❖ **Bloques MCM - DCT-1D-E:** En este caso, estos bloques son implementados teniendo en cuenta el algoritmo Hcub, el cual realiza las multiplicaciones por múltiples constantes utilizando sumas, restas y desplazamientos [39]. El diagrama de bloques del bloque MCM basado en el algoritmo Hcub puede obtenerse desde el generador de bloques MCM disponible en la página web del proyecto Spiral [40].

Para la generación de los bloques MCM se determinó un máximo de dos etapas de sumas y/o restas sucesivas con el fin de minimizar el tiempo de cálculo, y así incrementar la frecuencia de operación.

En la Figura 5.1 se muestra el diagrama esquemático de los bloques MCMs para las transformadas DCT-1D-E de 4-puntos y 8-puntos obtenidos usando el algoritmo Hcub. Los bloques MCMs para las transformadas DCT-1D-E de 16 y 32-puntos se observan en las Figuras 5.2 y 5.3, respectivamente.

En este caso, las operaciones de desplazamiento en los bloques MCMs de la transformada DCT-1D-E no consumen recursos hardware ya que estas son implementadas en forma cableada. Adicionalmente, los valores de las señales de entrada y de salida de los bloques de suma y resta son representados utilizando la mínima cantidad de bits. Esto con el fin de utilizar sumadores y restadores de menor tamaño y así disminuir el consumo de área.

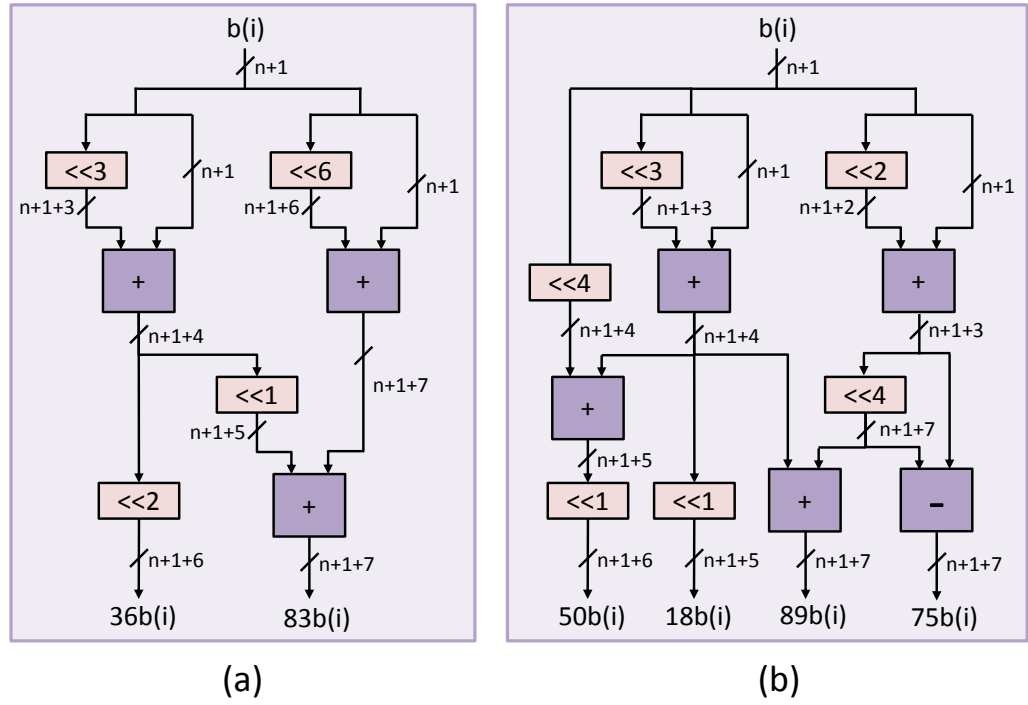


Figura 5.1 Bloque MCM para la transformada DCT-1D-E: a) 4-puntos. b) 8-puntos

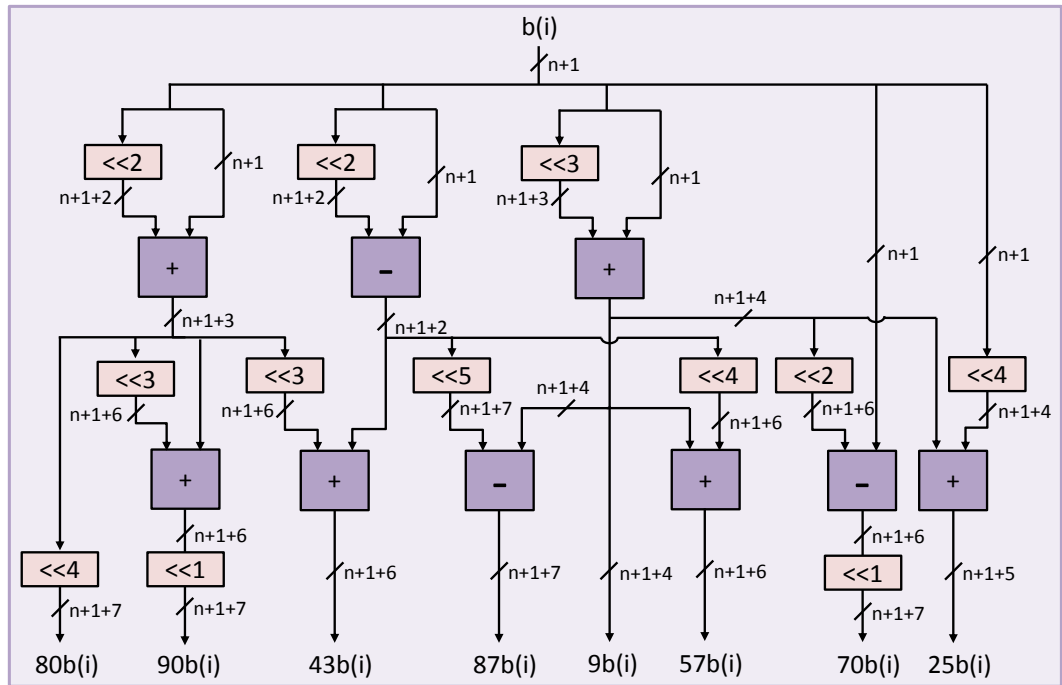


Figura 5.2 Bloque MCM para la transformada DCT-1D-E de 16-puntos

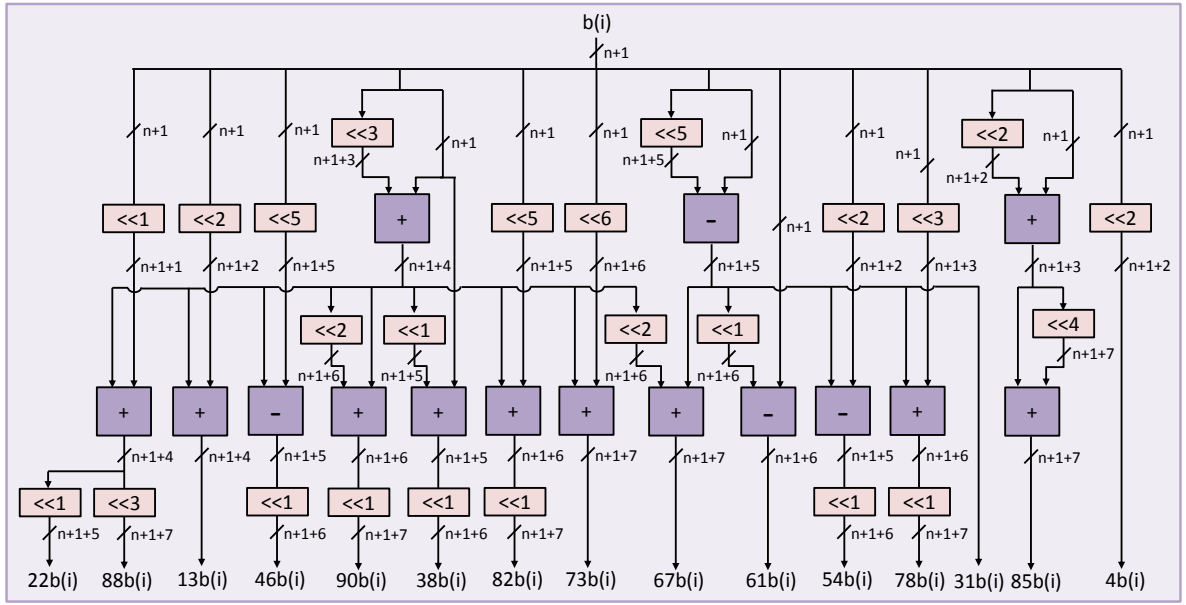


Figura 5.3 Bloque MCM para la transformada DCT-1D-E de 32-puntos

❖ **Bloques Sumadores – DCT-1D-E:** La implementación en hardware de cada bloque sumador de la transformada DCT-1D-E está compuesto de $M-1$ etapas de sumadores y restadores de dos entradas, los cuales están organizados en forma de árbol. El conjunto de $N/2$ bloques sumadores realizan las operaciones de suma y resta necesarias para obtener las salidas impares de la transformada DCT-1D a partir de los resultados de los $N/2$ bloques MCMs de acuerdo con la ecuación 5.5. Cada bloque sumador se encarga de generar una de las salidas impares de la transformada. En la Figura 5.4 se pueden observar los bloques sumadores de salida que generan las salidas $Y(1)$ y $Y(3)$ de la transformada DCT-1D-E de 8-puntos. En la Figura 5.5 se muestra el bloque sumador de salida que generan la salida $Y(15)$ de la transformada DCT-1D-E de 16-puntos y en la Figura 5.6 se muestra el bloque sumador de salida que generan la salida $Y(31)$ de la transformada DCT-1D-E de 32-puntos. En estas figuras los restadores llevan a cabo la operación $A-B$, donde la línea continua y punteada representan las entradas A y B , respectivamente.

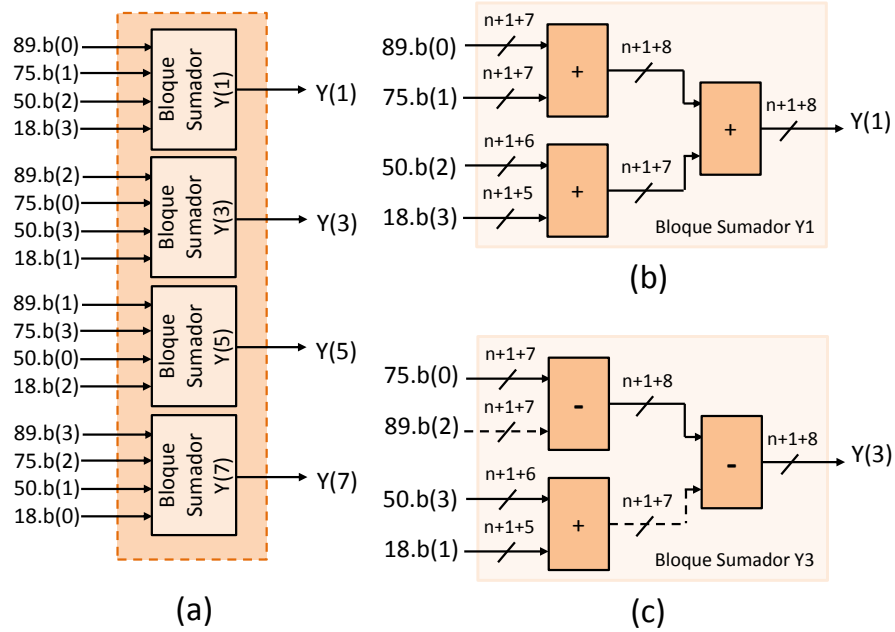


Figura 5.4 a) $N/2$ bloques sumadores de la transformada DCT-1D de 8-puntos. b) Bloque sumador que genera la salida $Y(1)$. c) Bloque sumador que genera la salida $Y(3)$

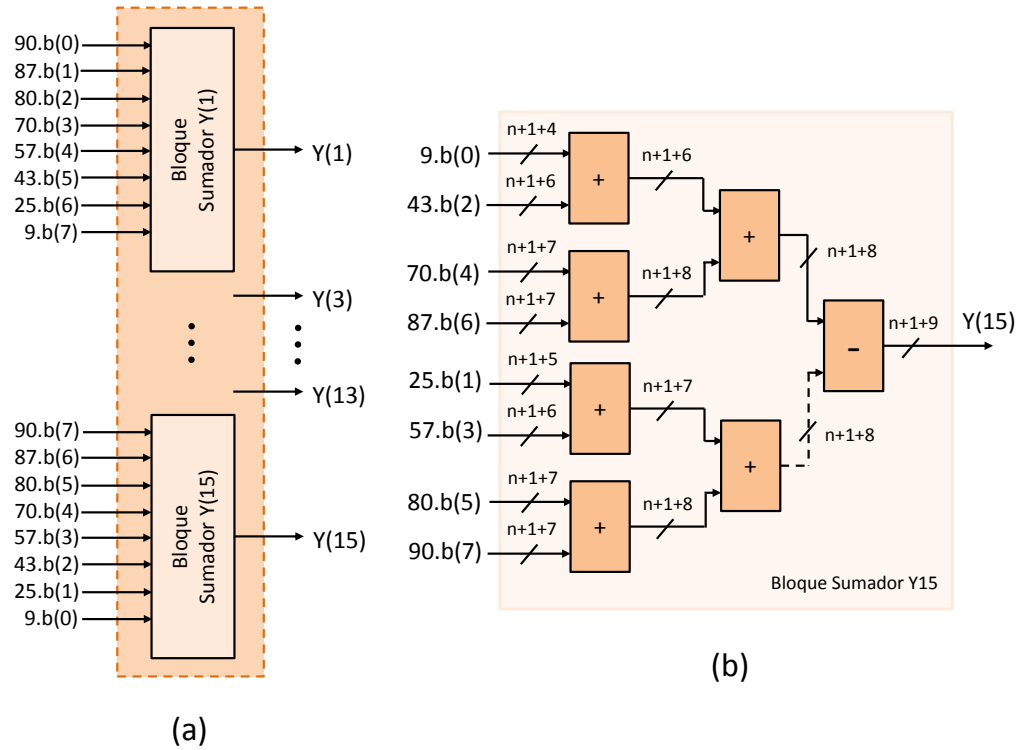


Figura 5.5 a) $N/2$ bloques sumadores de la transformada DCT-1D de 16-puntos. b) Bloque sumador que genera la salida $Y(15)$

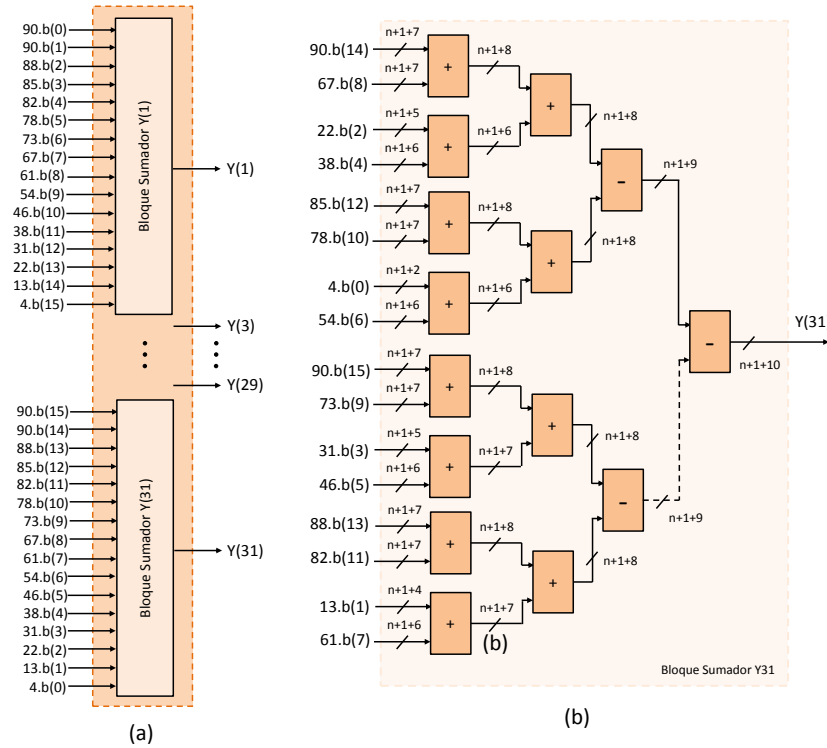


Figura 5.6 a) $N/2$ bloques sumadores de la transformada DCT-1D de 32-puntos. b) Bloque sumador que genera la salida $Y(31)$

Los diagramas de bloques de las transformadas DCT-1D-E de 4 y 8-puntos se presentan en las Figuras 5.7 y 5.8, respectivamente. El diagrama de bloques genérico de la transformada DCT-1D-E de N-puntos se presenta en la Figura 5.9.

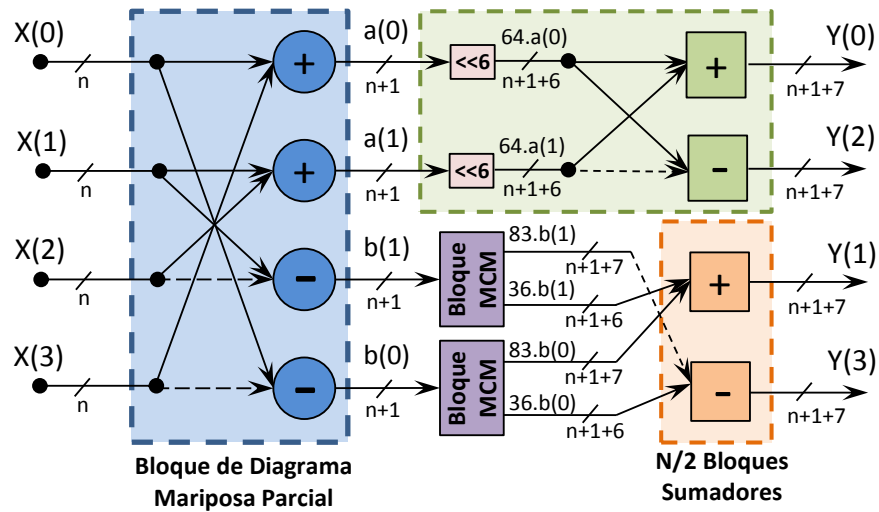


Figura 5.7 Diagrama de bloques de la transformada DCT-1D-E de 4-puntos

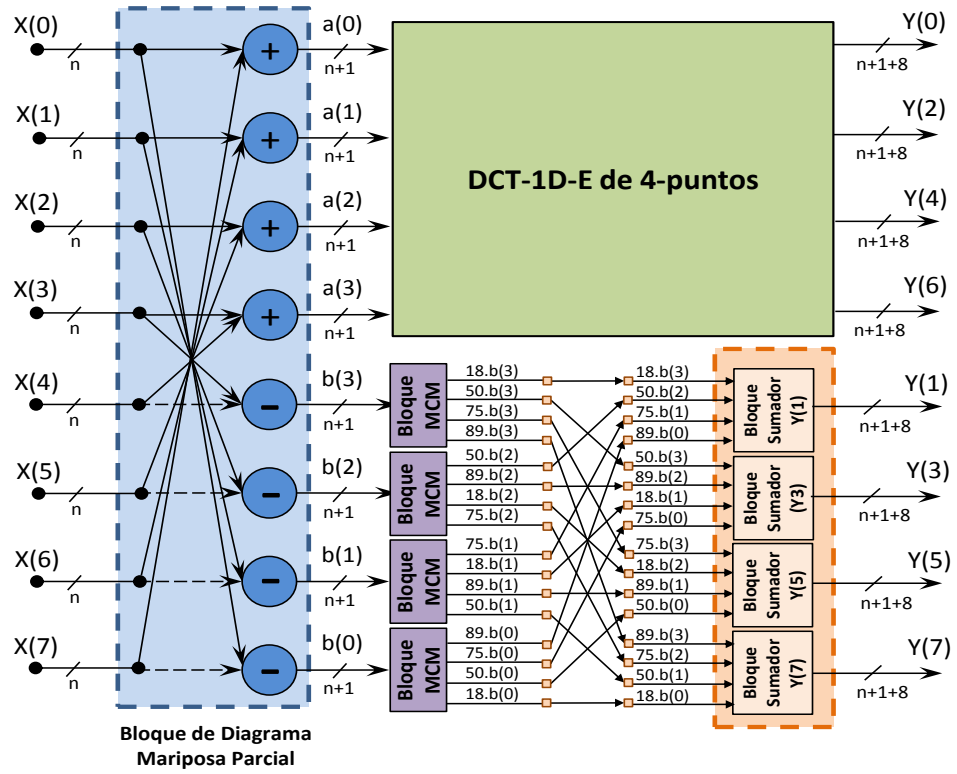


Figura 5.8 Diagrama de bloques de la transformada DCT-1D-E de 8-puntos

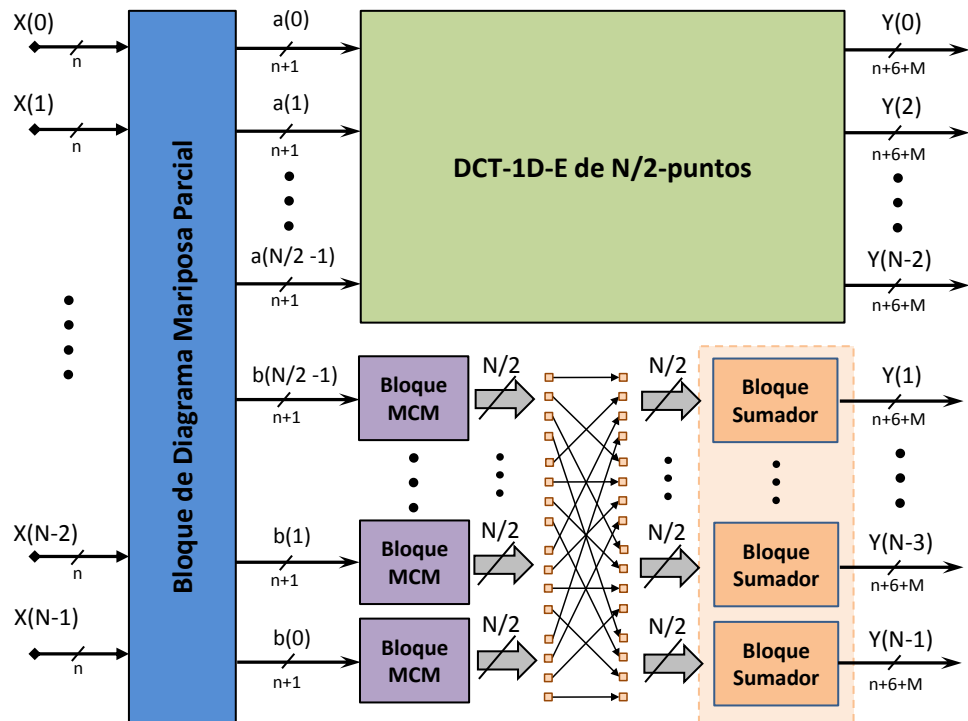


Figura 5.9 Diagrama de bloques genérico de la transformada DCT-1D-E de N -puntos

5.3.2 TRANSFORMADA DCT-1D ESTRUCTURADA Y PIPELINE DE N-PUNTOS

El diseño hardware de la transformada DCT-1D estructurada y pipeline (DCT-1D-P) se basa en el diseño de la transformada DCT-1D-E. Este diseño posee una forma regular, lo cual nos permite realizar fácilmente una segmentación del diseño y de esta forma incluir registros en la entrada y salida de cada etapa de sumas y restas. De ésta manera es posible introducir nuevos datos en cada ciclo de reloj mejorando el desempeño de la arquitectura, es decir, incrementando en gran medida la frecuencia de operación y el “throughput”.

Los diagramas de bloques de las transformadas DCT-1D-P de 4 y 8-puntos se pueden observar en las Figuras 5.10 y 5.11, respectivamente. El diagrama de bloques genérico de la transformada DCT-1D-P de N-puntos se puede observar en la Figura 5.12.

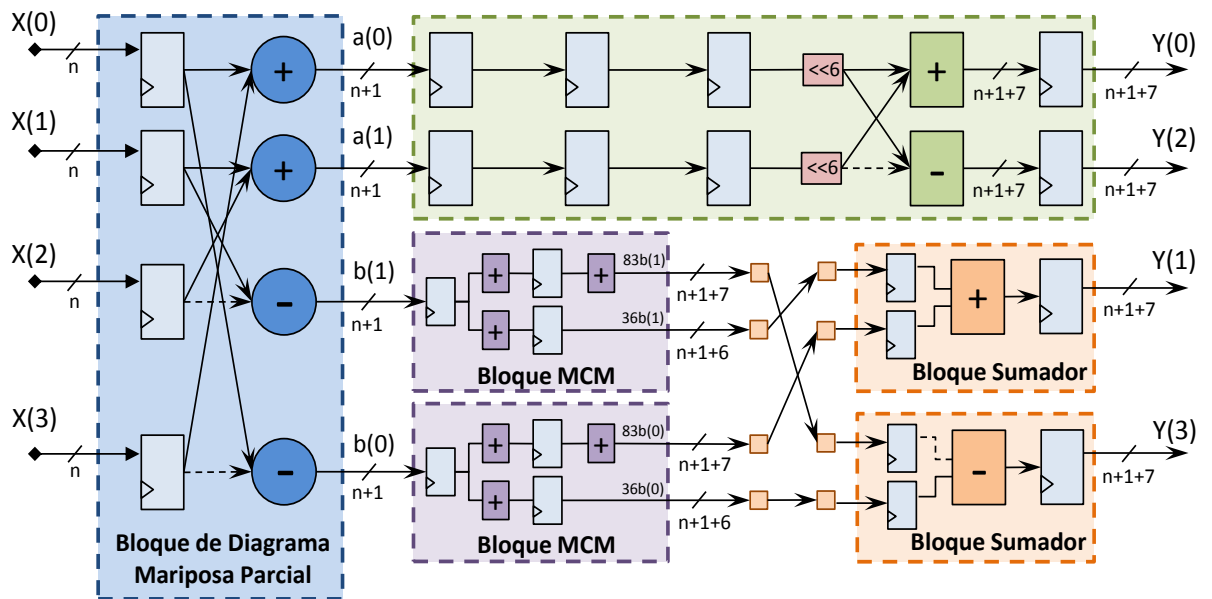


Figura 5.10 Diagrama de bloques de la transformada DCT-1D-P de 4-puntos

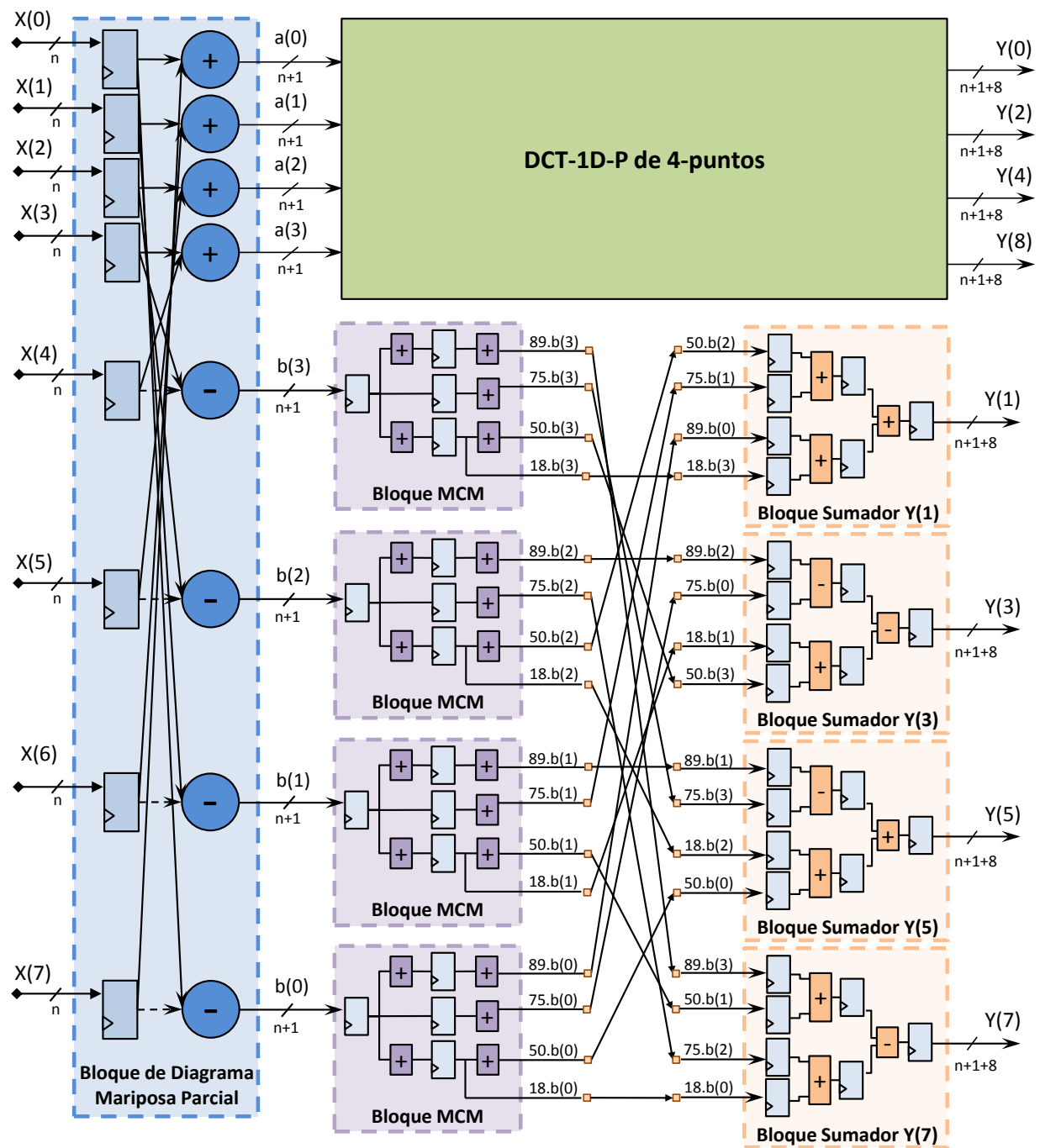


Figura 5.11 Diagrama de bloques de la transformada DCT-1D-P de 8-puntos

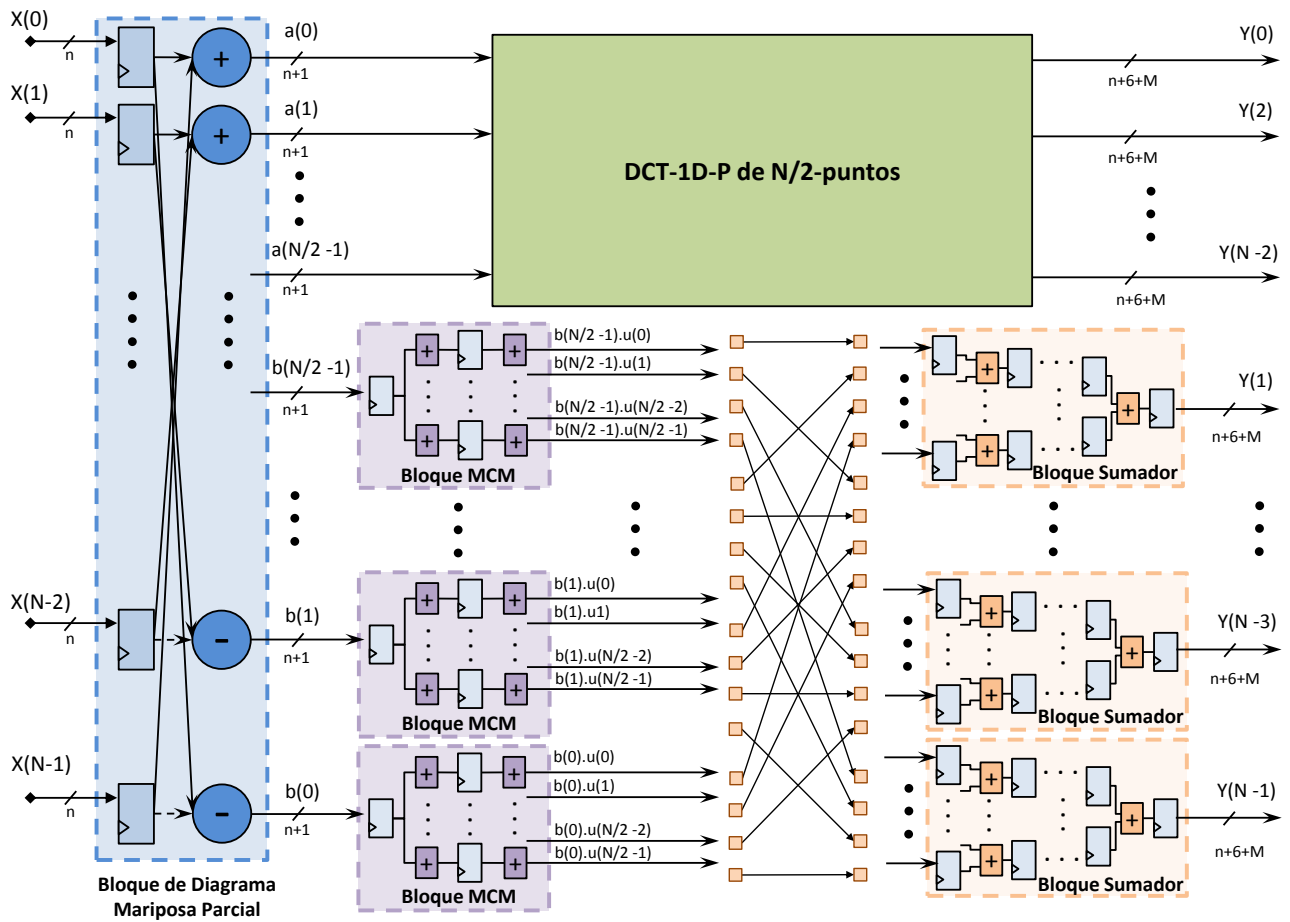


Figura 5.12 Diagrama de bloques genérico de la transformada DCT-1D-P de N-puntos

5.3.3 TRANSFORMADA DCT-1D MODIFICADA DE N-PUNTOS

El diseño de la transformada DCT-1D modificada (DCT-1D-M) está basado en bloques MCM y bloques sumadores modificados, los cuales permiten obtener una reducción en el consumo de área y un incremento en la frecuencia de operación con respecto al diseño descrito previamente.

- ❖ **Bloque MCM - DCT-1D-M:** En este caso, los bloques MCMs son diseñados utilizando un enfoque empírico, el cual usa sumadores de dos y tres entradas con el fin de aprovechar de manera más eficiente los recursos hardware del FPGA al momento del mapeo y síntesis, reduciendo el consumo de área y potencia de la arquitectura. Es importante mencionar que

los bloques MCMs modificados no utilizan restadores ya que estos tienen menor frecuencia de operación y consumen mayor área que los sumadores. Por lo tanto, estos bloques están implementados solamente con sumadores paralelos de 2 y 3 entradas. Al igual que los bloques MCMs utilizados en diseños anteriores se determina un máximo de dos etapas de sumas sucesivas con el fin de minimizar el tiempo de cálculo y así incrementar la frecuencia de operación.

El diagrama esquemático de los bloques MCMs modificados para las transformadas DCT-1D-M de 4 y 8-puntos se observan en la Figura 5.13. Los bloques MCM modificados para las transformadas DCT-1D-M de 16 y 32-puntos se observan en las Figuras 5.14 y 5.15, respectivamente. A partir de estas figuras se puede observar que los bloques MCM modificados requieren menor número de sumadores que los bloques MCM utilizados en la implementación de la transformada DCT-1D-E.

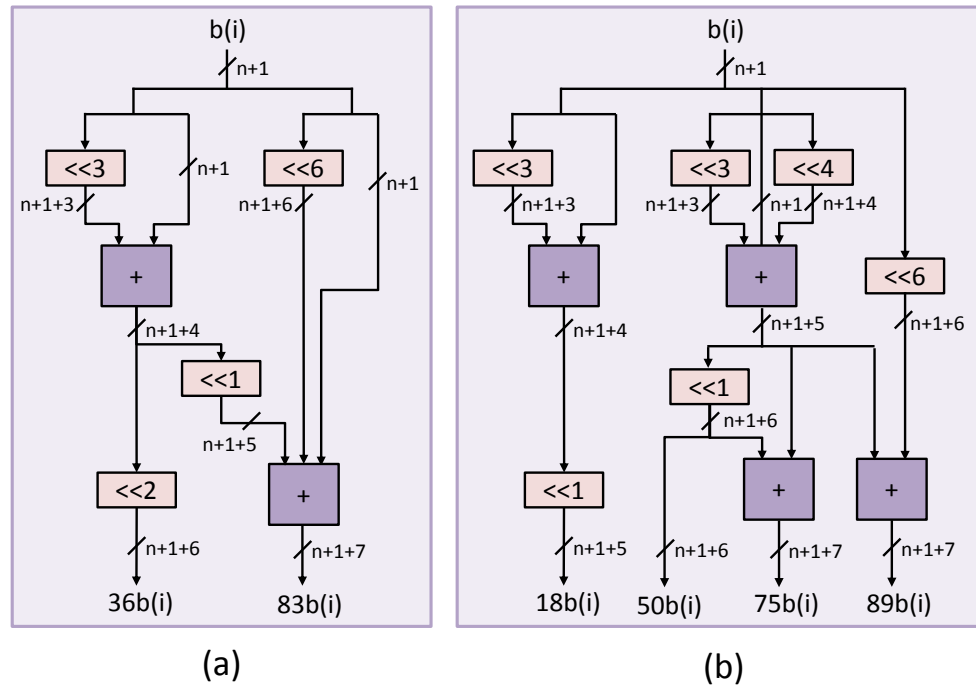


Figura 5.13 Bloque MCM modificado para la transformada DCT-1D-M: a) 4-puntos. b) 8-puntos.

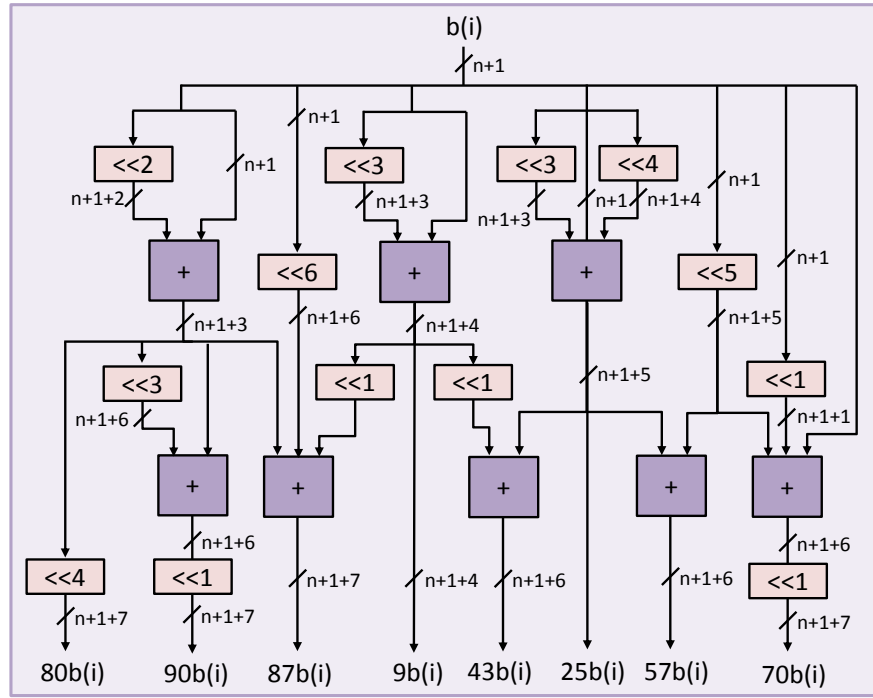


Figura 5.14 Bloque MCM modificado para la transformada DCT-1D-M de 16-puntos.

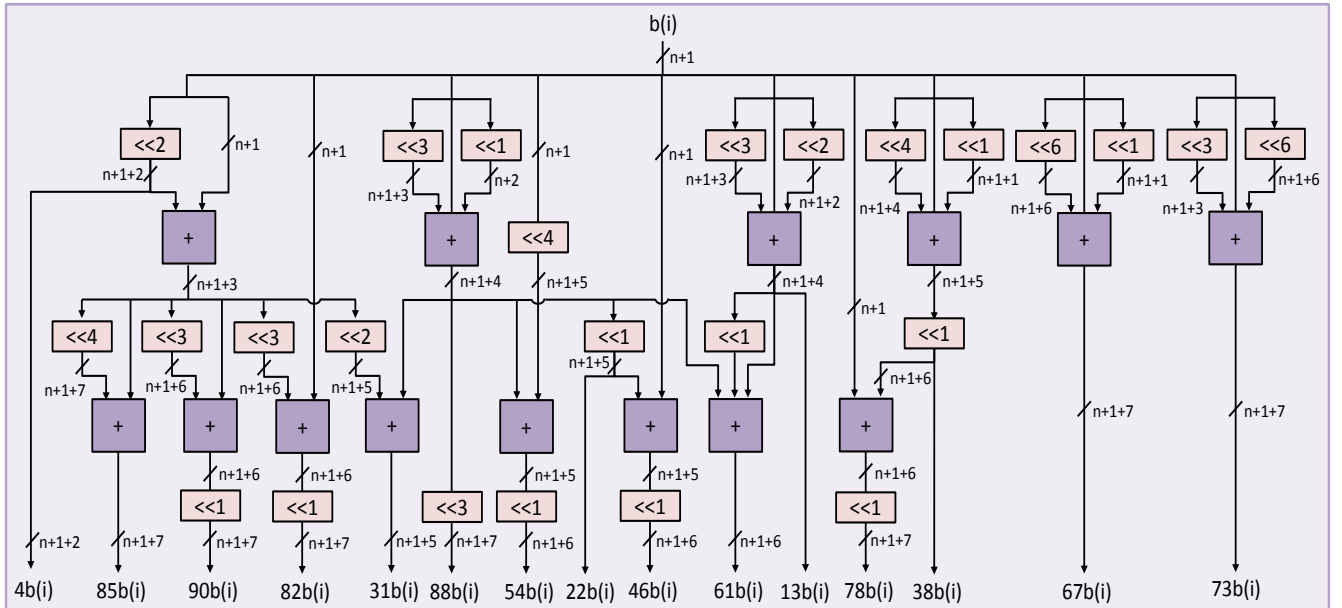


Figura 5.15 Bloque MCM Modificado para la transformada DCT-1D de 32-puntos.

- ❖ **Bloques Sumadores – DCT-1D-M:** Del mismo modo, el diseño de los bloques sumadores modificados utiliza un enfoque empírico, en donde se utilizan sumadores y restadores de dos y tres entradas, tratando en lo posible de utilizar la menor cantidad de restadores debido a que estos consumen mayor área y tienen menor frecuencia de operación.

A diferencia de los bloques sumadores usados en el diseño de la transformada DCT-1D-E, los bloques sumadores modificados no presentan una estructura regular, lo cual impide realizar una segmentación estructurada del diseño para implementar etapas de pipeline. Además estos bloques presentan una gran variabilidad en la organización de los sumadores/restadores para cada uno de los tamaños de la transformada DCT-1D-M. Por lo tanto, los bloques sumadores modificados presentan una estructura común para la transformada DCT-1D-M de 8-puntos, pero tres estructuras diferentes para las transformadas DCT-1D-M de 16 y 32-puntos. Las estructuras de los bloques sumadores para las transformadas DCT-1D-M de 8, 16 y 32-puntos se presentan en las Figuras 5.16, 5.17 y 5.18, respectivamente.

La Figura 5.16 muestra la estructura del bloque sumador modificado que genera la salida $Y(1)$ en la transformada DCT-1D-M de 8-puntos. Esta estructura también puede ser utilizada para generar el resto de salidas impares, para esto es necesario reasignar las entradas del bloque sumador modificado y reconfigurar los bloques funcionales internos ya sea como sumador o restador de tal forma que se obtengan los resultados según la ecuación 5.5.

La Figura 5.17 muestra tres tipos de bloques sumadores modificados para la transformada DCT-1D-M de 16-puntos. La Figura 5.17 a) muestra el bloque sumador modificado para generar la salida $Y(1)$. La Figura 5.17 b) muestra al bloque sumador modificado para generar la salida $Y(3)$. Finalmente, en la Figura 5.17 c) se observa el bloque sumador modificado que genera la salida $Y(7)$. Al igual que en el caso anterior, estas estructuras también pueden ser utilizadas para generar el resto de salidas impares de la transformada DCT-1D-M de 16 puntos. En este caso la estructura del bloque sumador modificado que genera la salida $Y(3)$ también se utiliza para generar las salidas $Y(5)$, $Y(11)$ y $Y(13)$. Del mismo modo, la estructura del bloque sumador modificado que genera la salida $Y(7)$ también se usa para obtener las salidas $Y(9)$ y $Y(15)$.

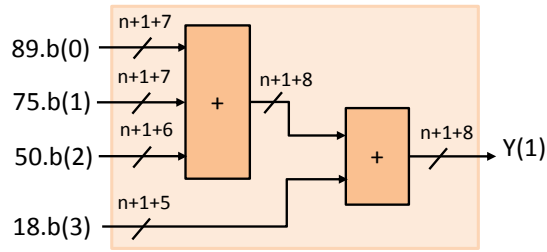


Figura 5.16 Bloque sumador modificado que genera la salida $Y(1)$ en la transformada DCT-1D-M de 8-puntos.

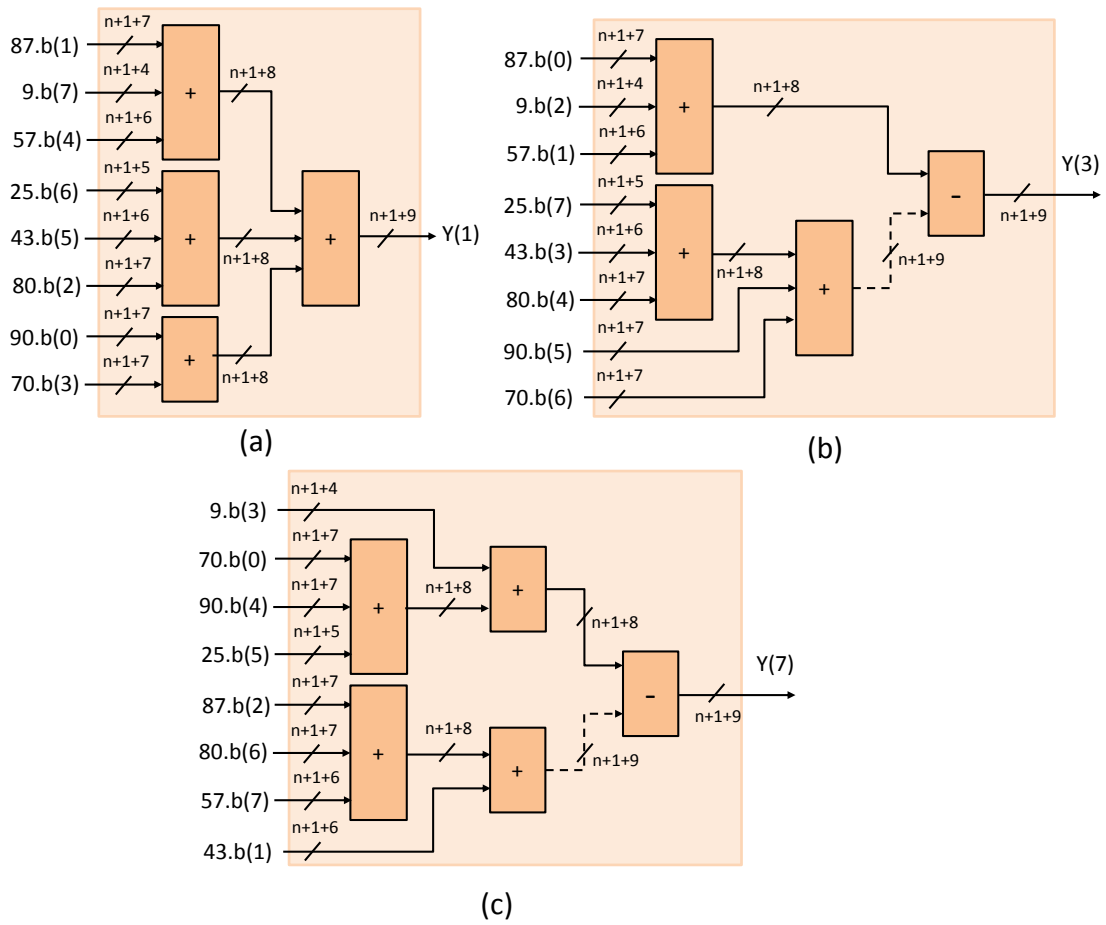


Figura 5.17 Bloques sumadores modificados para la transformada DCT-1D-M de 16-puntos.

a) Bloque sumador modificado que genera la salida $Y(1)$. b) Bloque sumador modificado que genera la salida $Y(3)$. c) Bloque sumador modificado que genera la salida $Y(7)$.

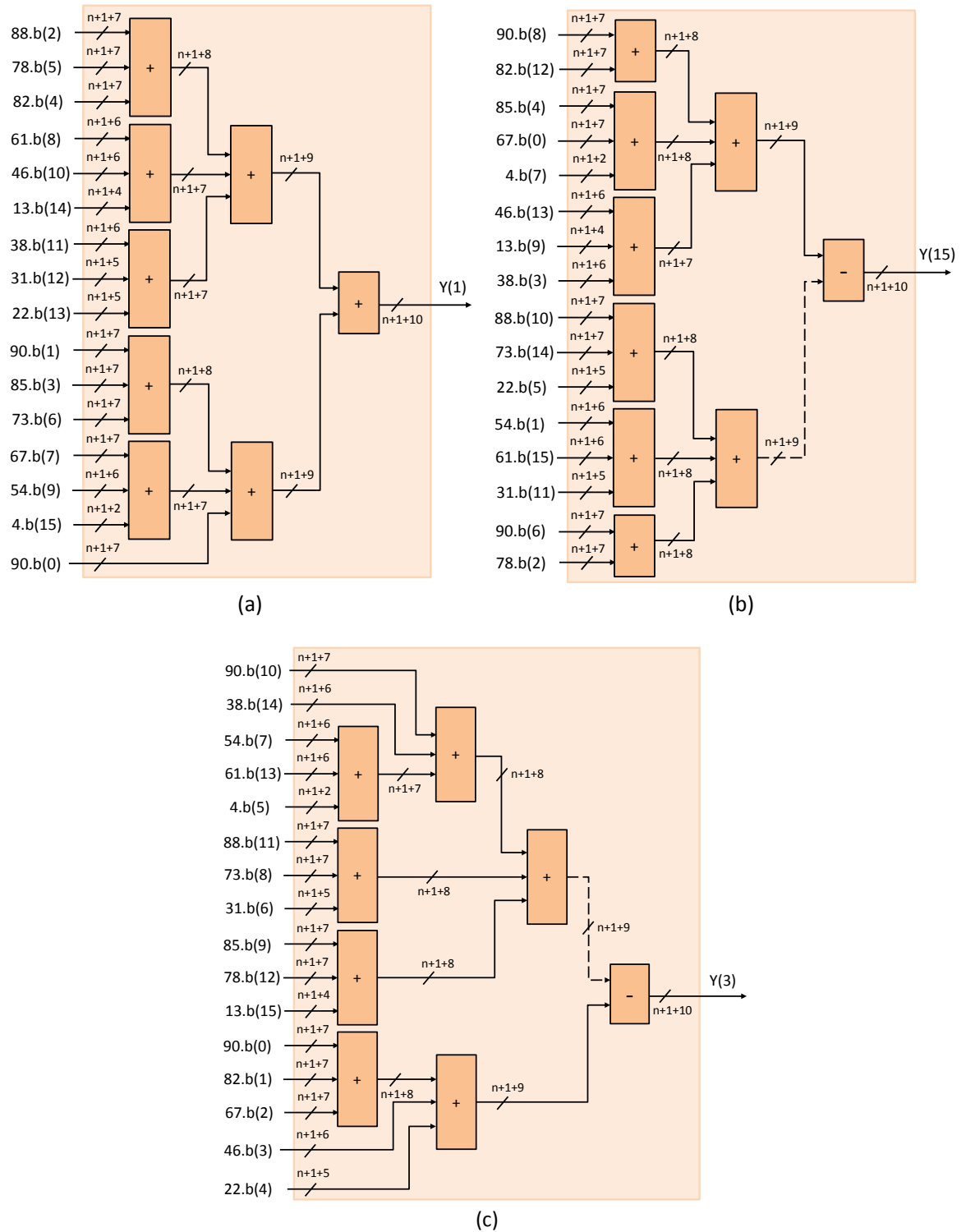


Figura 5.18 Bloques sumadores modificados para la transformada DCT-1D-M de 32-puntos.

a) Bloque sumador modificado que genera la salida $Y(1)$. b) Bloque sumador modificado que genera la salida $Y(15)$. c) Bloque sumador modificado que genera la salida $Y(3)$.

La Figura 5.18 muestra tres tipos de bloques sumadores modificados para la transformada DCT-1D-M de 32-puntos. La Figura 5.18 a) muestra el bloque sumador modificado para generar la salida $Y(1)$. La Figura 5.18 b) representa el bloque sumador modificado que genera la salida $Y(15)$. Finalmente, en la Figura 5.18 c) se observa el bloque sumador modificado que genera la salida $Y(3)$. Como en los casos anteriores, estas estructuras también pueden ser utilizadas para obtener el resto de salidas impares de la transformada DCT-1D-M de 32 puntos. En este caso, la estructura del bloque sumador modificado que genera la salida $Y(1)$ también se utiliza para obtener las salidas $Y(5)$, $Y(11)$, $Y(13)$, $Y(19)$, $Y(27)$ y $Y(29)$. La estructura del bloque sumador modificado que genera la salida $Y(15)$ también se utiliza para las salidas $Y(17)$, $Y(23)$, $Y(25)$ y $Y(31)$. Finalmente, la estructura del bloque sumador modificado que genera la salida $Y(3)$ también se utiliza para generar las salidas $Y(7)$, $Y(9)$ y $Y(21)$.

Desde las anteriores figuras es posible observar que la implementación de los bloques sumadores modificados presenta una distribución balanceada en las señales de salida y entrada de cada sumador y restador, de tal forma que el incremento del tamaño de los datos debido al acarreo de salida sea lo más uniforme posible para ahorrar recursos de hardware.

El diagrama de bloques de la transformada DCT-1D-M de 4-puntos es el mismo que se presenta en la Figura 5.7. En este caso, solo es necesario reemplazar los bloques MCM por los bloques MCM modificados de la Figura 5.13 a).

El diagrama de bloques de la transformada DCT-1D-M de 8-puntos es el mismo que se presenta en la Figura 5.8. En este caso, solo es necesario reemplazar el bloque DCT-1D-E de 4-puntos por el bloque DCT-1D-M de 4-puntos, los bloques MCM por los bloques MCM modificados de la Figura 5.13 b), y utilizar el bloque sumador modificado mostrado en la Figura 5.16.

El diagrama de bloques genérico de la transformada DCT-1D de N-puntos es el mismo que se presenta en la Figura 5.9. En este caso, solo es necesario reemplazar el bloque DCT-1D-E de $N/2$ -puntos por el bloque DCT-1D-M de $N/2$ -puntos, y utilizar los bloques modificados para cada tamaño de la transformada DCT-1D, los cuales se presentan en las Figuras 5.13 - 5.18.

5.4 DISEÑO HARDWARE DE LA TRANSFORMADA DCT-2D

En esta sección se describen los diseños de cada uno de los bloques funcionales que conforman la arquitectura hardware de la transformada DCT-2D para el estándar H.265/HEVC. Teniendo en cuenta la propiedad de separabilidad de la transformada DCT-2D, se llega a la conclusión que su implementación en hardware requiere dos transformadas DCT-1D y un bloque de trasposición de matrices. Además, se deben tener en cuenta los factores de redondeo y escalamiento de los datos de salida después de cada etapa de transformación, los cuales están definidos por el estándar H.265/HEVC y fueron mencionados en el capítulo anterior.

El proceso para calcular la transformada DCT-2D inicia con la transformación de los datos de entrada usando un bloque DCT-1D. Los resultados de la primera transformación son redondeados sumando a cada salida el valor de $2^{(B+M-10)}$ y escalados por un factor de $ST1 = 2^{-(B+M-9)}$, posteriormente se realiza la trasposición de la matriz resultante y se lleva a cabo la segunda etapa de transformación usando un segundo bloque DCT-1D. Finalmente, los resultados de la segunda transformación son redondeados sumando a cada salida el valor de $2^{(M+5)}$ y escalados en un factor de $ST2 = 2^{-(M+6)}$. Todo este proceso se controla por medio de una unidad de control, la cual es diseñada usando una máquina de estados finita (FSM: *Finite State Machine*), un contador y un registro de desplazamiento.

En la Figura 5.19 se muestra el diagrama de bloques del codificador de video híbrido H.265/HEVC y el procesamiento que se realiza la etapa de transformación directa de este codificador.

En la Figura 5.20 se puede observar el diagrama de bloques de la arquitectura hardware de la transformada DCT-2D para el estándar H.265/HEVC. Desde este diagrama es posible diferenciar cada uno de los bloques que componen esta arquitectura, los cuales se describen a continuación.

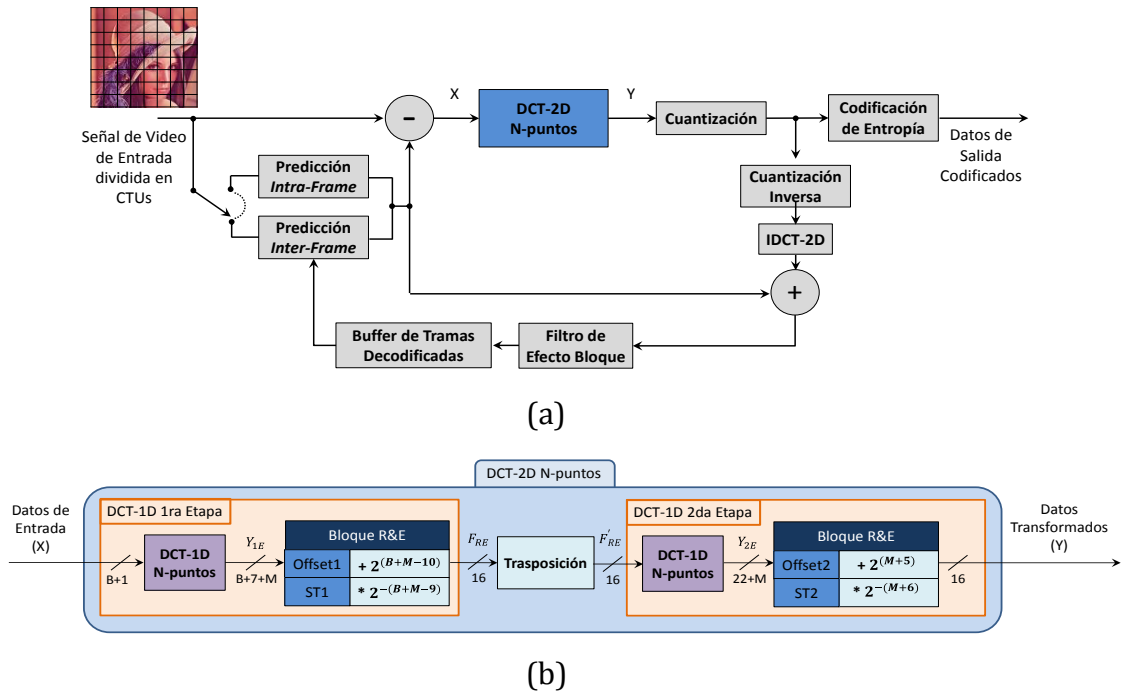


Figura 5.19 a) Diagrama esquemático del codificador de video H.265/HEVC. b) Etapas de procesamiento de la transformada DCT-2D.

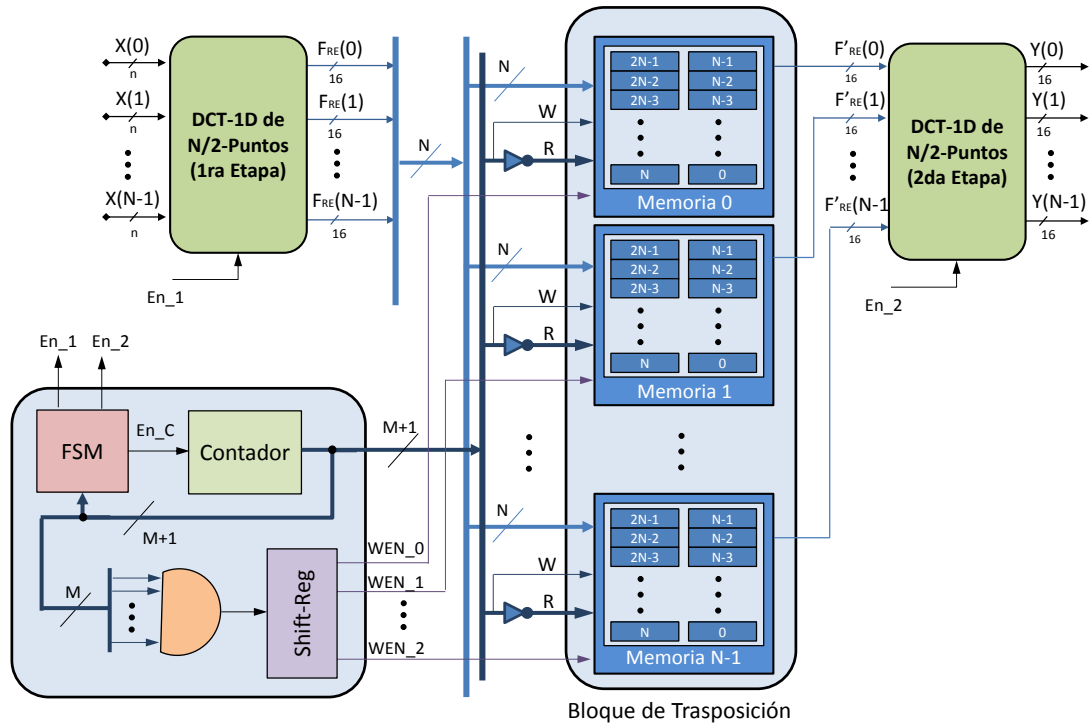


Figura 5.20 Diagrama de Bloques Genérico de la Transformada DCT-2D de N-Puntos para el estándar H.265/HEVC

5.4.1 BLOQUES DCT-1D DE N-PUNTOS

El diseño hardware del bloque DCT-1D utilizado en la primera y segunda etapa de transformación para la transformada DCT-2D es el mismo que se describe en la sección 5.3. Las únicas diferencias entre el bloque DCT-1D de la primera etapa y el bloque de la segunda etapa son el número de bits de las entradas, y los factores de redondeo y escalamiento de las salidas.

Para la primera etapa de transformación, el número de bits de las entradas es igual a $B+1$, el valor de redondeo es igual a $2^{(B+M-10)}$ y el factor de escalamiento es de $2^{-(B+M-9)}$. El factor de escalamiento de la primera etapa de transformación genera datos de salida de 16 bits tal y como se especifica en el estándar HEVC.

La segunda etapa de transformación recibe los datos de salida de la primera etapa después del proceso de redondeo, escalamiento y trasposición. Por lo tanto el número de bits de los datos de entrada de esta etapa siempre será fijo e igual a 16 bits. El valor de redondeo de la segunda etapa de transformación es igual a $2^{(M+5)}$ y el factor de escalamiento es de $2^{-(M+6)}$. Al igual que en la primera etapa de transformación el factor de escalamiento de la segunda etapa de transformación genera datos de salida de 16 bits siguiendo las especificaciones del estándar HEVC.

El redondeo y escalamiento sobre los N datos salida de la primera y segunda etapa de transformación esta descrito por las ecuaciones 5.9 y 5.10, respectivamente:

$$F_{RE}[i] = \frac{Y_{1E}[i] + 2^{(B+M-10)}}{2^{(B+M-9)}}, \quad \text{para } i = 0, 1 \dots N - 1 \quad (5.9)$$

$$Y[i] = \frac{Y_{2E}[i] + 2^{(M+5)}}{2^{(M+6)}}, \quad \text{para } i = 0, 1 \dots N - 1 \quad (5.10)$$

Donde $Y_{1E}[i]$ representa los N datos de salida de la primera etapa de transformación y $F_{RE}[i]$ es el resultado después del redondeo y el truncamiento. $Y_{2E}[i]$ representa los N datos de salida de la segunda etapa de transformación y $Y[i]$ su resultado después del redondeo y el truncamiento.

Siendo $M = \log_2 N$ y $B=n-1$, el número de bits de $Y_{1E}[i]$ es igual a $B+7+M$ y el número de bits de $Y_{2E}[i]$ es igual a $16+7+M-1$, en donde n representa el número de bits de cada una de las N entradas de la transformada DCT-2D.

Teniendo en cuenta las anteriores ecuaciones es posible deducir que el redondeo generará un cambio en el valor de $Y_{1E}[i]$ solo si el bit en la posición $B+M-10$ de $Y_{1E}[i]$ es igual a 1, lo cual es equivalente a sumar $Y_{1E}[i]$ con el valor del bit $Y_{1E}[i]$ ($B+M-10$). Lo mismo ocurre para $Y_{2E}[i]$ y el bit $Y_{2E}[i]$ ($M+5$). En cuanto al escalamiento, éste puede ser implementado mediante desplazamientos a la derecha o simplemente truncando los bits menos significativos del dato. Lo anterior puede ser expresado mediante las siguientes ecuaciones:

$$F_{RE}[i][15..0] = Y_{1E}[i][B+M+6... B+M-9] + Y_{1E}[i][B+M-10], \quad \text{para } i=0, 1, \dots, N-1 \quad (5.11)$$

$$Y[i][15..0] = Y_{2E}[i][M+21... M+6] + Y_{2E}[i][M+5], \quad \text{para } i=0, 1, \dots, N-1 \quad (5.12)$$

El diagrama esquemático de la arquitectura hardware del bloque DCT-1D empleado en el diseño de la transformada DCT-2D para el estándar H-265/HEVC es el mismo que se presentó en la sección anterior, solo que en este caso es necesario incluir el diseño hardware de las etapas de redondeo y escalamiento, tal y como se muestra en las Figuras 5.21 y 5.22.

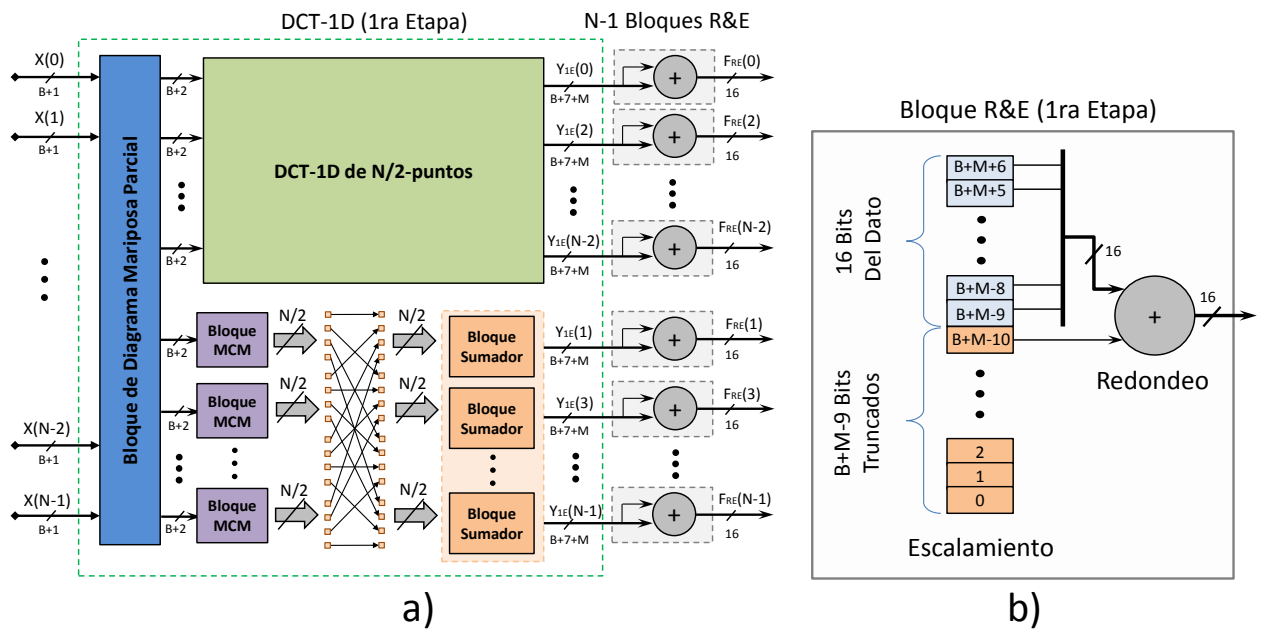


Figura 5.21 a) Diagrama de bloques genérico de la primera etapa de transformación DCT-1D de N-puntos con redondeo y escalaamiento. b) Diagrama detallado del bloque de redondeo y escalaamiento - primera etapa

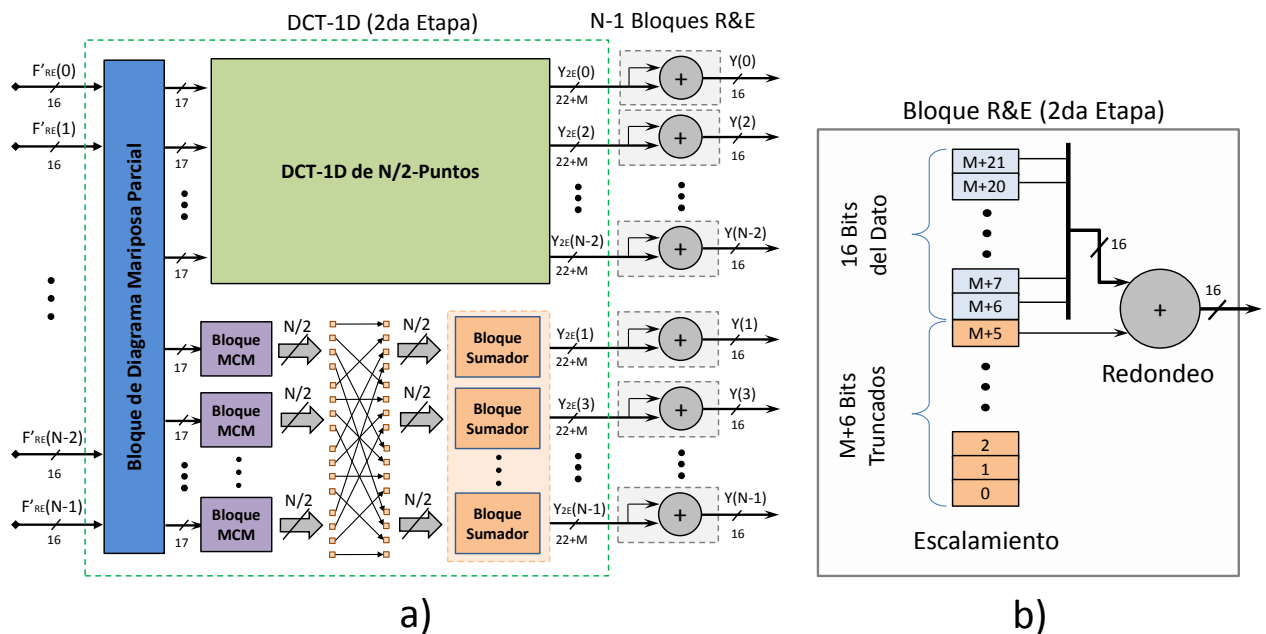


Figura 5.22 a) Diagrama de bloques genérico de la segunda etapa de transformación DCT-1D de N-puntos con redondeo y escalaamiento. b) Diagrama detallado del bloque de redondeo y escalaamiento - segunda etapa

5.4.2 BLOQUE DE TRASPOSICIÓN DE MATRICES

La gran versatilidad que ofrece el diseño hardware permite reconfigurar bloques funcionales de acuerdo a las necesidades. El diseño del bloque de transformación de matrices para la transformada DCT-2D es un ejemplo de ello, ya que para su implementación se utilizan N memorias RAM modificadas de dos puertos. Estas memorias fueron diseñadas de tal forma que el número de datos a almacenar sea mayor al número de datos a leer.

Cada memoria tiene 2 posiciones de escritura y cada una de ellas esta subdividida en N posiciones de 16 bits, es decir $2N$ posiciones de memoria de 16 bits en total, donde N es el número de entradas de la transformada. En este caso, las memorias poseen el doble de la capacidad necesaria para el proceso de trasposición con el fin de llevar a cabo una segunda trasposición consecutiva sin ciclos de latencia intermedios, logrando un procesamiento continuo de los datos.

En cuanto a las señales de control, a cada memoria ingresan una señal de habilitación de escritura de un bit, una señal de dirección de escritura de un bit y una señal de dirección de lectura de $M+1$ bits.

Las direcciones de lectura y escritura de cada memoria se generan mediante un contador ascendente de $M+1$ bits, que cuenta desde cero hasta $2N-1$. La dirección de escritura es generada por el bit más significativo (MSB) del contador, de este modo cuando el contador genere un valor entre 0 y $N-1$, la dirección de escritura será "0" y cuando el contador genere un valor entre N y $2N-1$, la dirección de escritura será "1". Por otra parte, la dirección de lectura de $M+1$ bits es generada por los $M+1$ bits del contador negados, esto debido a que las posiciones de memoria están organizadas en forma descendente.

Cada una de las N memorias es capaz de almacenar un vector columna de N datos transformados de 16 bits provenientes de la primera transformada DCT-1D en solo un ciclo de reloj. Además, gracias a su configuración de dos puertos modificados, cada memoria es capaz de leer cada uno de estos N datos en forma individual de tal forma que se obtenga la trasposición de la matriz transformada.

El proceso que realiza el bloque de trasposición de matrices para obtener una matriz traspuesta consta de los siguientes pasos:

- 1) Las N señales “Wen” habilitan la escritura de una de las N memorias a la vez en cada ciclo de reloj permitiendo almacenar cada vector columna de la matriz de datos transformados en una memoria diferente. Si tenemos en cuenta que el contador inicialmente tiene un valor de “ N ”, este generará una dirección de escritura igual a “1”, Por lo tanto, los N datos de los N vectores columna serán almacenado en las posiciones $2N-1, 2N-2, \dots, N+1, N$ de cada una de las N memorias. Este proceso toma N ciclos de reloj.
- 2) Posteriormente, se lee cada uno de los N datos de los N vectores columna en forma individual, es decir se toma un dato desde cada una de las N memorias obteniendo un vector fila de N datos. Si tenemos en cuenta que después de N ciclos de reloj el contador tendrá un valor de “0”, este generará una dirección de lectura igual a “ $2N-1$ ”, por lo tanto el primer vector fila estará compuesto de los datos almacenados en las posiciones $2N-1$ de cada una de las N memorias. Al siguiente ciclo de reloj el valor del contador será “1” generando una dirección de lectura igual a “ $2N-2$ ”, por lo tanto el siguiente vector fila estará compuesto por los datos almacenados en las posiciones $2N-2$ de cada una de las N memorias. Este proceso se repite hasta obtener los N vectores fila de la matriz traspuesta.
- 3) El proceso de lectura de datos conlleva otros N ciclos de reloj tiempo en el cual es posible almacenar un segundo conjunto de N vectores columna de una segunda matriz de datos en la segunda mitad de las N memorias de transposición. En este lapso de tiempo el contador genera valores entre 0 y $N-1$, lo cual genera una dirección de escritura de “0”. por lo tanto, los N datos de cada uno de los N vectores columna de la segunda matriz de entrada serán almacenados en las posiciones $N-1, N-2, \dots, 1, 0$ de cada una de las N memorias. Este proceso toma N ciclos de reloj que corresponden a los mismos N ciclos de reloj que se emplean en la lectura de los datos de la primera matriz traspuesta.
- 4) Posteriormente, se lee cada uno de los N datos de los N vectores columna de la segunda matriz en forma individual. Si tenemos en cuenta que después de N ciclos de reloj el contador nuevamente tendrá un valor de “ N ”, este generará una dirección de lectura igual

a “N-1”, por lo tanto el primer vector fila de la segunda matriz traspuesta estará compuesto por los datos almacenados en las posiciones N-1 de cada una de las N memorias. Al siguiente ciclo de reloj el valor del contador será “N+1” generando una dirección de lectura igual a “N-2”, por lo tanto el siguiente vector fila estará compuesto por los datos almacenados en las posiciones N-2 de cada una de las N memorias. Este proceso se repite hasta obtener los N vectores fila de la segunda matriz traspuesta tiempo en el cual es almacenada una nueva matriz de datos en la primera mitad de las N memorias de trasposición.

5.4.3 UNIDAD DE CONTROL DE LA TRANSFORMADA DCT-2D

La unidad de control de la arquitectura hardware de la transformada DCT-2D para el estándar H.265/HEVC está conformada por un contador de $M + 1$ bits, un registro de desplazamiento a la derecha de N bits, un circuito lógico, y una FSM.

- **Contador:** El contador de $M+1$ bits genera las direcciones escritura y de lectura de las N memorias del bloque de trasposición. El contador debe ser cargado con un valor inicial, el cual es generado por la FSM. Este valor hace que las direcciones de lectura y escritura iniciales de las memorias del bloque de trasposición se sincronicen con el final de la primera etapa de transformación. Es decir el contador toma el valor N cuando la primera DCT-1D genera el primer vector columna de datos transformados, esto genera una dirección de escritura igual a “1” y de esta forma se da comienzo al proceso de trasposición.
- **Registro de Desplazamiento:** El registro de desplazamiento a la derecha de N bits genera la señal de control “Wen” de cada una de las N memorias del bloque de trasposición. Estas señales habilitan la escritura de una de las N memorias a la vez en cada ciclo de reloj permitiendo almacenar cada vector columna de la matriz de datos transformados en una memoria diferente. La habilitación de la primera memoria del bloque de trasposición se debe realizar justo después de finalizar la primera etapa de transformación, este proceso se lleva a cabo activando la señal Wen_0, la cual es generada por el bit más significativo del registro de desplazamiento. Posteriormente el registro de desplazamiento desplazará este

dato un bit a la derecha cada ciclo de reloj generando la activación de las señales de control de escritura $Wen_1 \dots Wen_{N-1}$.

- **Circuito lógico:** Este circuito se encarga de generar la señal de entrada serial del registro de desplazamiento. Esta señal hace que después de un flanco de reloj el bit más significativo del registro de desplazamiento tome el valor de '1' y de esta forma se inicia la habilitación de las memorias del bloque de trasposición. La señal serial de entrada del registro de desplazamiento solo debe activarse 1 vez cada N ciclos de reloj, esto debido a dos razones, la primera es que solo una de las N memorias del bloque de trasposición puede estar activa en cada ciclo de reloj y la segunda razón es que después de N ciclos de reloj el bloque de trasposición finaliza la escritura de una matriz de datos transformados y continua con la escritura de una nueva matriz de datos transformados en la otra de mitad de las posiciones de memoria, esto hace necesario que el proceso de habilitación de las memorias de trasposición se reinicie después de N ciclos de reloj. En los instantes en que el circuito lógico debe activar la señal de entrada serial del registro de desplazamiento, el valor del contador es igual a $N-1$ y $2N-1$, esto indica que el circuito lógico debe generar un '1' cuando los M bits menos significativos del contador sean "1". Teniendo en cuenta lo anterior el circuito lógico que genera esta señal se implementa utilizando simplemente una compuerta AND de M bits, tomando como entradas los M bits menos significativos del contador.
- **FSM:** La FSM de la unidad de control de la transformada DCT-2D controla la habilitación de los dos bloques DCT-1D y determina el valor inicial del contador. Este proceso es llevado a cabo teniendo en cuenta el número de ciclos de latencia inicial de los bloques DCT-1D y del bloque de trasposición. Estos valores varían de acuerdo al tipo de transformada DCT-1D que se vaya a utilizar en la implementación de la transformada DCT-2D. A las transformadas DCT-2D basadas en bloques DCT-1D-E, DCT-1D-M y DCT-1D-P las llamaremos DCT-2D-E, DCT-2D-M y DCT-2D-P, respectivamente. Teniendo en cuenta lo anterior se proponen dos FSMs:
- ❖ **FSM para la Transformada DCT-2D-E/DCT-2D-M:** Esta FSM genera las diferentes señales de control teniendo en cuenta el número de ciclos de latencia inicial de cada bloque funcional, los cuales se presentan en la Tabla 5.2. En este caso se incluyeron etapas de registros en las entradas y salidas de la transformada DCT-1D-E/DCT-1D-M.

En la Tabla 5.2 el número de ciclos de latencia esta dado de forma genérica para cada uno de los bloques, donde $M = \log_2 N$, y N es el número de datos de entrada de la transformada.

Bloque	Latencia Inicial en ciclos
1ra DCT-1D	2
Trasposición	$N+2$
2da DCT-1D	2

Tabla 5.2 Ciclos de latencia inicial de los bloques que conforman la arquitectura de la transformada DCT-2D-E/DCT-2D-M

Teniendo en cuenta los datos presentados en la tabla anterior, se concluye que el bloque de trasposición debe almacenar el primer vector de datos transformados después de dos ciclos de latencia. En este instante el valor del contador debe ser de N , por lo tanto el valor inicial que se carga en el contador al inicio del proceso de transformación debe ser igual a N menos los ciclos de latencia inicial del primer bloque DCT-1D-E/DCT-1D-M, es decir $N-2$.

Después de almacenar los N vectores transformados, ocupando la mitad de las posiciones de cada una de las N memorias de trasposición, el valor del contador será de "0", de esta forma se inicia la lectura de los datos en forma descendente desde la posición $N-1$ hasta la posición N . Los vectores traspuestos de N datos se obtienen leyendo uno de los N datos almacenados en las N memorias de trasposición a la vez en cada ciclo de reloj. La lectura de los datos desde las memorias de trasposición requiere de 2 ciclos de reloj, esto debido a que las memorias de trasposición poseen registros internos en las entradas y salidas. Por lo tanto, el segundo bloque DCT-1D-E/DCT-1D-M debe activarse dos ciclos después de iniciar la lectura del primer vector traspuesto. Esto implica que el cambio de estado debe realizarse un ciclo antes, instante en el cual el valor del contador es igual a "1". A partir de este momento todas las señales de control están sincronizadas y el procesamiento de los datos se realiza de forma continua y sin ciclos de latencia intermedios.

- ❖ **FSM para la Transformada DCT-2D-P:** Al igual que la otra FSM, esta genera las diferentes señales de control teniendo en cuenta el número de ciclos de latencia inicial de cada bloque funcional, los cuales se presentan en la Tabla 5.3.

Bloque	Latencia Inicial en ciclos
1ra DCT-1D	M+3
Trasposición	N+2
2da DCT-1D	M+3

Tabla 5.3 Ciclos de latencia inicial de los bloques que conforman la arquitectura de la transformada DCT-2D-P

Teniendo en cuenta los datos presentados en la tabla anterior, se puede concluir que en este caso el bloque de trasposición debe almacenar el primer vector de datos transformados después de M+3 ciclos de latencia. En este instante el valor del contador debe ser de N, Por lo tanto el valor inicial del contador debe ser igual a $N-(M+3)$.

En cuanto a la habilitación del segundo bloque DCT-1D-P, se realiza en el mismo instante que se determinó en la FSM para la transformada DCT-2D-E/DCT-2D-M.

El diagrama de estados de las FSMs para la DCT-2D-E/DCT-2D-M y DCT-2D-P se observa en la Figura 5.23. En este caso el diagrama de estados de las dos FSMs es el mismo, pero se debe tener en cuenta que la condición ($C=1$) que genera el cambio de estado y a su vez la habilitación del segundo bloque DCT-1D ocurre en instantes diferentes en cada caso. Lo anterior debido a que el valor inicial del contador es de N-2 para la DCT-2D-E/DCT-2D-M y de $N-(M+3)$ para la DCT-2D-P.

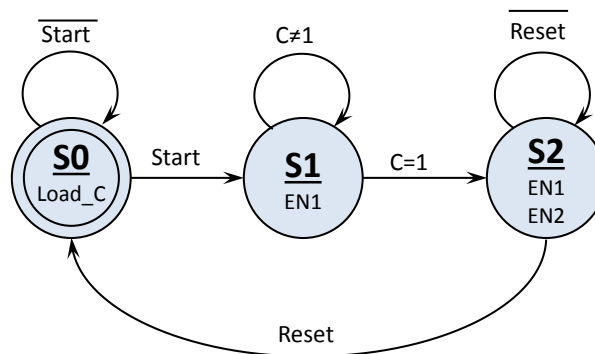


Figura 5.23 Diagrama de estados de la FSM de la unidad de control para la transformada DCT-2D-E/DCT-2D-M y DCT-2D-P

Las señales de entrada de la FSM y su funcionalidad están especificadas en la Tabla 5.4. Las señales de salida de la FSM y su funcionalidad se presentan en la Tabla 5.5.

SEÑAL	DESCRIPCIÓN
Start	Inicia el proceso de la transformada DCT-2D
Reset	Reinicia el proceso de la transformada DCT-2D
C	Indica el valor del contador

Tabla 5.4 Señales de entrada de la FSM de la unidad de control para la transformada DCT-2D-E/DCT-2D-M y DCT-2D-P

SEÑAL	DESCRIPCIÓN
EN1	Activa el primer bloque DCT-1D
EN2	Activa el segundo bloque DCT-1D
Load_C	Carga el valor inicial del contador

Tabla 5.5 Señales de Salida de la FSM de la unidad de control para la transformada DCT-2D-E/DCT-2D-M y DCT-2D-P

Las señales de control generadas por la unidad de control para la transformada DCT-2D-E/DCT-2D-M de 4-puntos se pueden observar en el diagrama de tiempo de la Figura 4.24. De igual forma las señales de control generadas por la unidad de control para la transformada DCT-2D-P de 4-puntos se pueden observar en el diagrama de tiempo de la Figura 5.25.

En las Figuras 5.24 y 5.25 las líneas punteadas de color azul distinguen los ciclos de latencia inicial de la primera transformada DCT-1D, las líneas punteadas de color rojo, distinguen los 4 ciclos de reloj usados para almacenar la matriz de datos transformados. Los dos ciclos de latencia de color amarillo hacen referencia a los ciclos de latencia inicial que presenta la memoria en el proceso de lectura de datos. Finalmente las líneas punteadas de color verde representan los ciclos de latencia inicial de la segunda transformada DCT-1D. A partir de este momento el sistema empieza a arrojar vectores de datos transformados cada ciclo de reloj, consiguiendo un procesamiento continuo de los datos y sin ciclos de latencia intermedios. Este proceso continúa hasta que se active la señal de *Reset* la cual reinicia todo el proceso.

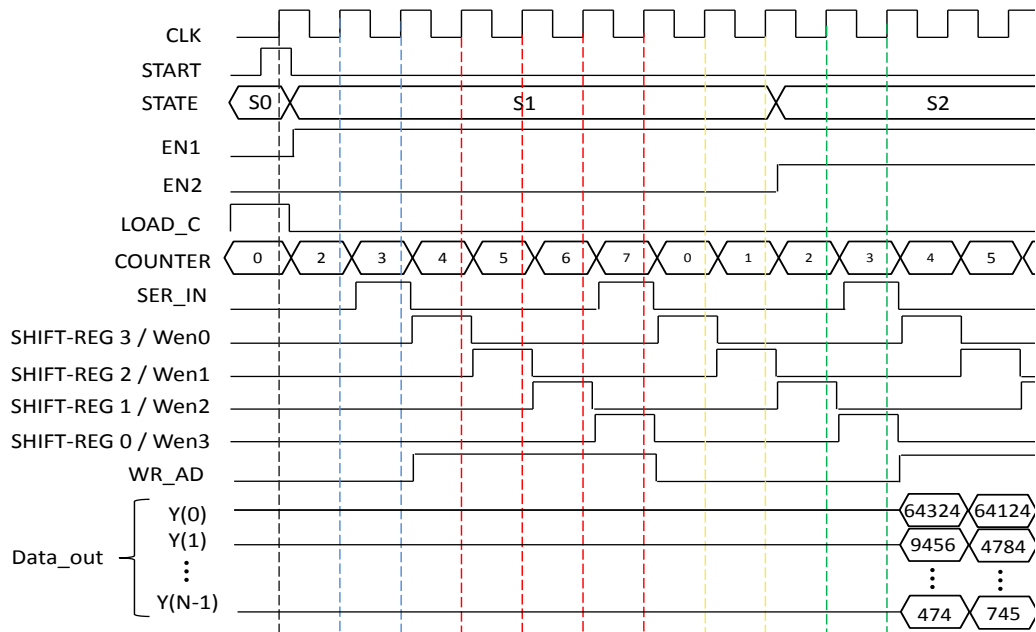


Figura 5.24 Diagrama de tiempo de la unidad de control de la transformada DCT-2D-E/DCT-2D-M de 4-puntos.

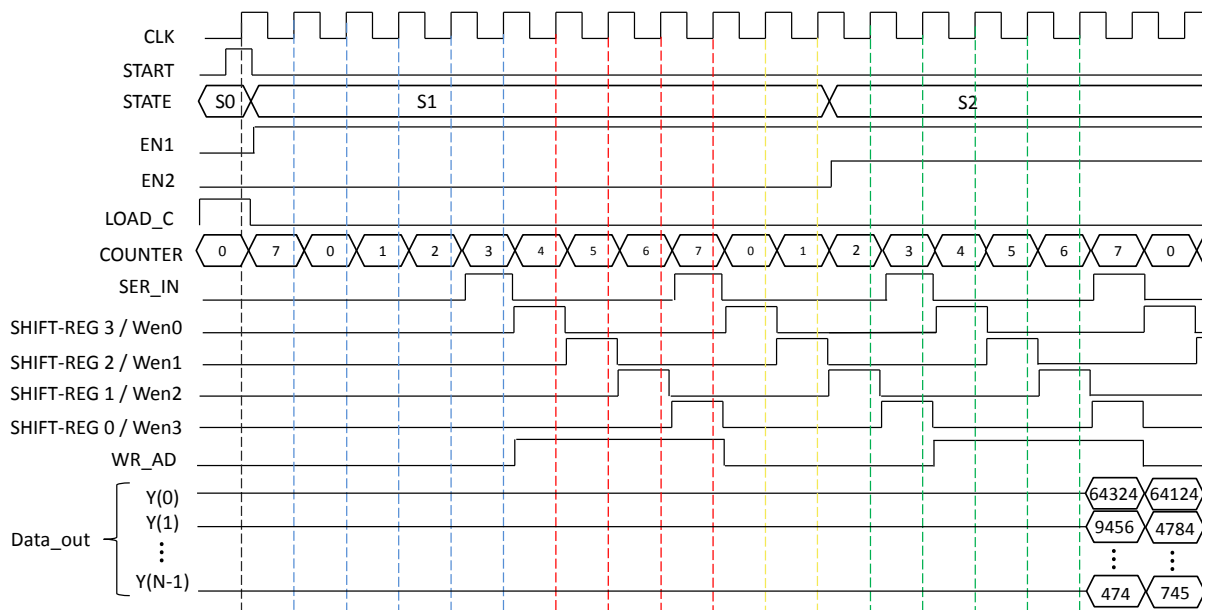


Figura 5.25 Diagrama de tiempo de la unidad de control de la transformada DCT-2D-P de 4-puntos

5.5 DISEÑO HARDWARE DE LA TRANSFORMADA IDCT-1D

La transformada IDCT-1D se puede considerar como la multiplicación entre la matriz base de la transformada DCT-1D traspuesta y un vector de datos transformados tal y como se define en la siguiente ecuación:

$$\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-2} \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} C_{0,0} & C_{0,1} & \cdots & C_{0,N-2} & C_{0,N-1} \\ C_{1,0} & C_{1,1} & & C_{1,N-2} & C_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{N-2,0} & C_{N-2,1} & \cdots & C_{N-2,N-2} & C_{N-2,N-1} \\ C_{N-1,0} & C_{N-1,1} & & C_{N-1,N-2} & C_{N-1,N-1} \end{bmatrix}^T \begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{N-2} \\ Y_{N-1} \end{bmatrix} \quad (5.13)$$

Donde, $Y[0 \dots N-1]$ representa el vector de datos transformados de tamaño N , $C[0 \dots N-1, 0 \dots N-1]^T$ representa la traspuesta de la matriz base de la transformada DCT-1D de tamaño $N \times N$, y $X[0 \dots N-1]$ es el vector de datos originales de tamaño N .

Al igual que en la transformada DCT-1D directa, en la transformada IDCT-1D también es posible explotar las propiedades de simetría y anti-simetría de la matriz base de la transformada DCT con el fin de reducir el número de operaciones aritméticas mediante la descomposición de las matrices base en matrices de menor tamaño, tal como se describe a continuación:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N/2 - 2) \\ X(N/2 - 1) \end{bmatrix} = \begin{bmatrix} Z(0) + Z(N/2) \\ Z(1) + Z(N/2 + 1) \\ Z(2) + Z(N/2 + 2) \\ \vdots \\ Z(N/2 - 2) + Z(N - 2) \\ Z(N/2 - 1) + Z(N - 1) \end{bmatrix} \quad (5.14)$$

$$\begin{bmatrix} X(N/2) \\ X(N/2 + 1) \\ X(N/2 + 2) \\ \vdots \\ X(N - 2) \\ X(N - 1) \end{bmatrix} = \begin{bmatrix} Z(N/2 - 1) - Z(N - 1) \\ Z(N/2 - 2) - Z(N - 2) \\ Z(N/2 - 3) - Z(N - 3) \\ \vdots \\ Z(1) - Z(N/2 + 1) \\ Z(0) - Z(N/2) \end{bmatrix} \quad (5.15)$$

Donde,

$$\begin{bmatrix} Z(0) \\ Z(1) \\ Z(2) \\ \vdots \\ Z(N/2 - 2) \\ Z(N/2 - 1) \end{bmatrix} = C_{N/2}^T \begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ \vdots \\ Y(N - 4) \\ Y(N - 2) \end{bmatrix} \quad (5.16)$$

$$\begin{bmatrix} Z(N/2) \\ Z(N/2 + 1) \\ Z(N/2 + 2) \\ \vdots \\ Z(N - 2) \\ Z(N - 1) \end{bmatrix} = W_{N/2} \begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ \vdots \\ Y(N - 3) \\ Y(N - 1) \end{bmatrix} \quad (5.17)$$

En este caso, el vector de datos transformados $Y=[Y(0), Y(1), \dots, Y(N-1)]$ es el vector de datos de entrada de la transformada IDCT-1D; el vector $X=[X(0), X(1), \dots, X(N-1)]$ es el resultado de la transformada IDCT-1D, $C_{N/2}^T$ es la traspuesta de la matriz base de la transformada DCT de tamaño $N/2$ y $W_{N/2}$ es una matriz de tamaño $N/2 \times N/2$, la cual está definida por la siguiente ecuación:

$$W_{N/2}(i, j) = C_N^T(i, 2j + 1), \text{ para } 0 \leq i, j \leq \frac{N}{2} - 1 \quad (5.18)$$

Note que la ecuación (5.16) puede ser calculada usando la misma descomposición recursiva, obteniendo las matrices $C_{N/4}^T$ y $W_{N/4}$. Este enfoque puede ser utilizado para todos los tamaños de la transformada IDCT-1D.

De lo anterior es posible deducir que la transformada IDCT-1D de N-puntos puede ser implementada en hardware utilizando: $N/2$ bloques MCMs, $N/2$ bloques sumadores, un bloque IDCT-1D de $N/2$ -puntos y un bloque de diagrama mariposa parcial.

❖ **Bloques MCMs:** Resolviendo la ecuación 5.18 se obtiene la misma matriz de valores únicos $W_{N/2}$ que se describió en el diseño de la DCT-1D de N-puntos. Por lo tanto, para el diseño de la

transformada IDCT-1D de N-puntos es posible utilizar los mismos bloques MCMs del diseño de la transformada DCT-1D de N-puntos. En este caso, estos bloques se encargan de multiplicar cada una de las entradas impares del vector de datos de entrada $Y=[Y(1), Y(3), \dots, Y(N-1)]$ por el vector de constantes compuesto por los valores únicos de la matriz $W_{N/2}$.

- ❖ **Bloques sumadores:** Los bloques sumadores suman los resultados de los bloques MCM para generar los resultados $Z(N/2) \dots Z(N-1)$ de acuerdo con la ecuación 5.17.
- ❖ **Transformada IDCT-1D de N/2-puntos:** Esta transformada procesa los datos pares del vector $Y=[Y(0), Y(2), \dots, Y(N-2)]$ para generar los resultados $Z(0) \dots Z(N/2-1)$ de acuerdo con la ecuación 5.16. En este caso, mientras que los datos impares del vector de datos transformados son procesados por los bloques MCMs y los bloques sumadores de salida, los datos pares son procesados en paralelo por el bloque IDCT-1D de N/2-puntos, de esta forma los valores del vector $Z=[Z(0), Z(1), \dots, Z(N-1)]$ son obtenidos en el mismo instante de tiempo.
- ❖ **Bloque de diagrama mariposa parcial:** Este bloque se compone N/2 sumadores de dos entradas de 23 bits que generan las salidas $X(0) \dots X(N/2-1)$ de acuerdo con la ecuación 5.14, y N/2 restadores de dos entradas de 23 bits que generan las salidas $X(N/2) \dots X(N-1)$ de acuerdo con la ecuación 5.15.

Los bloques sumadores de salida y el bloque de diagrama mariposa parcial, son los mismos bloques que se utilizaron en el diseño de la transformada DCT-1D de N-puntos. En este caso, estos bloques no tienen en cuenta el “*carry-out*” cuando el ancho de los datos resultantes es mayor o igual a 23 bits. Lo anterior es debido a que en el peor de los casos los valores resultantes no superan el valor del coeficiente DC multiplicado por 64, esto es equivalente a un incremento de solo 6 bits en el resultado a diferencia de la transformada DCT-1D de N-puntos que genera un incremento de M+6 bits en los datos de salida.

Al igual que en el diseño de la transformada DCT-1D, los datos de salida deben pasar por un proceso de redondeo y escalamiento con el fin de generar de 16 bits tal y como lo especifica el estándar H.265/HEVC. La implementación en hardware de la etapa de redondeo y

escalamiento no se presenta en esta sección, pero se abordará en detalle en el diseño de la transformada IDCT-2D.

A continuación se describen tres diseños de la transformada IDCT-1D de N-puntos, los cuales utilizan diferentes técnicas de diseño y diferentes enfoques para la implementación en hardware de los bloques MCMs y los bloques sumadores.

5.5.1 TRANSFORMADA IDCT-1D ESTRUCTURADA DE N-PUNTOS

El diseño hardware de la transformada IDCT-1D estructurada de N-puntos (IDCT-1D-E) se lleva a cabo utilizando los cuatro bloques funcionales mencionados anteriormente. Los bloques MCM de este diseño presentan la misma estructura de los bloques MCM utilizados en el diseño de la transformada DCT-1D-E, los cuales fueron descritos en la sección 5.3.1 de este capítulo. La única diferencia está en el tamaño de los datos de entrada de los bloques, en este caso son 16 bits en lugar de $n+1$ bits.

Los bloques sumadores de este diseño tienen la misma organización en forma de árbol de los bloques utilizados en el diseño de la transformada DCT-1D-E. El conjunto de $N/2$ bloques sumadores realizan las operaciones de suma y resta para obtener los resultados $Z(N/2) \dots Z(N-1)$, en donde cada bloque sumador se encarga de generar uno de estos valores.

En la Figura 5.26 se pueden observar los bloques sumadores que generan los resultados $Z(4)$ y $Z(5)$ de la transformada IDCT-1D-E de 8-puntos. En la Figura 5.27 se observa el bloque sumador que genera el resultado $Z(15)$ de la transformada IDCT-1D-E de 16-puntos, y en la Figura 5.28 se observa el bloque sumador que genera el resultado $Z(31)$ de la transformada IDCT-1D-E de 32-puntos.

En estas figuras los restadores llevan a cabo la operación $A-B$, en donde la línea continua y punteada representan las entradas A y B, respectivamente.

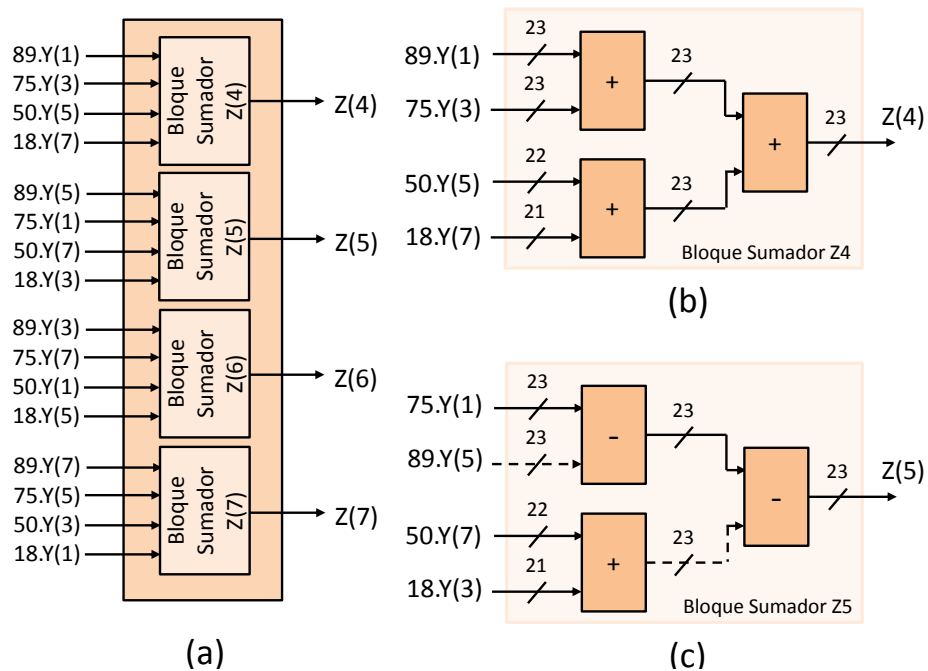


Figura 5.26 a) $N/2$ Bloques sumadores para la transformada IDCT-1D-E de 8 puntos. b) Bloque sumador que genera el resultado Z(4). c) Bloque sumador que genera el resultado Z(5).

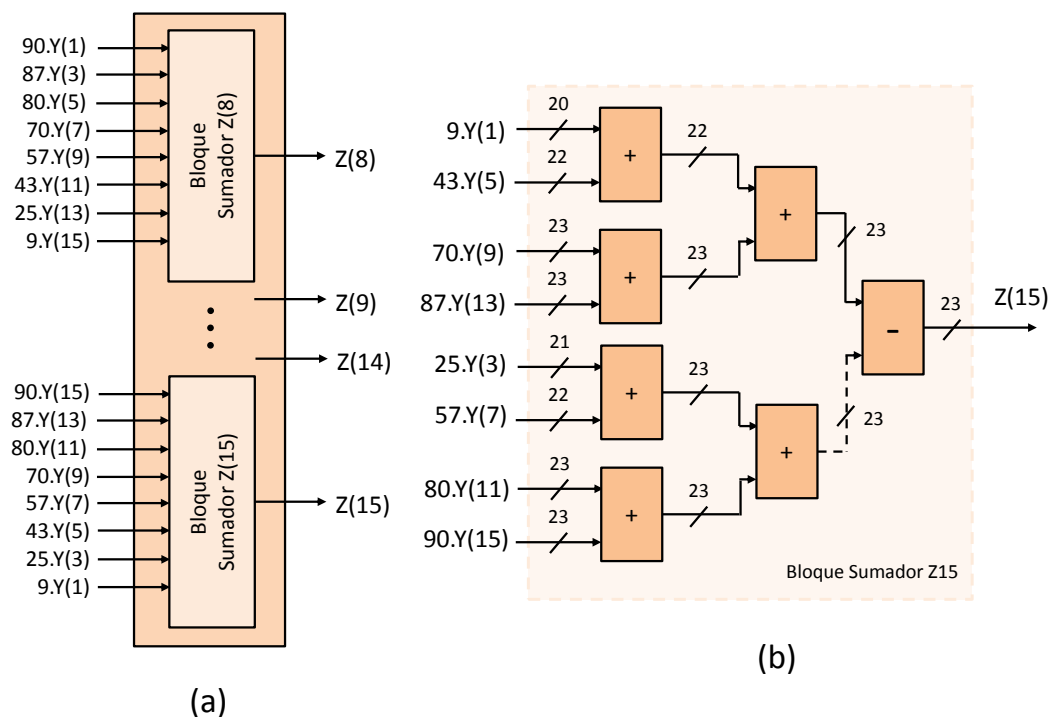


Figura 5.27 a) $N/2$ Bloques sumadores para la transformada IDCT-1D-E de 16 puntos. b) Bloque sumador que genera el resultado Z(15).

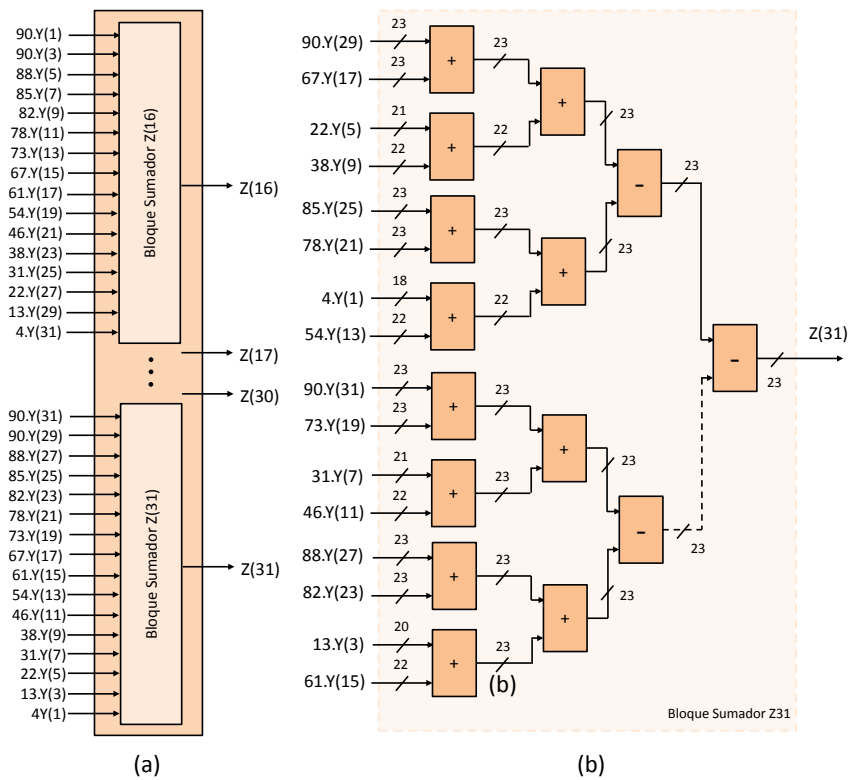


Figura 5.28 a) $N/2$ Bloques sumadores para la transformada IDCT-1D-E de 32 puntos. b) Bloque sumador que genera el resultado $Z(31)$

Los diagramas de bloques de las transformadas IDCT-1D-E de 4 y 8-puntos se presentan en las Figuras 5.29 y 5.30, respectivamente. El diagrama de bloques genérico de la transformada IDCT-1D-E de N-puntos se presenta en la Figura 5.31.

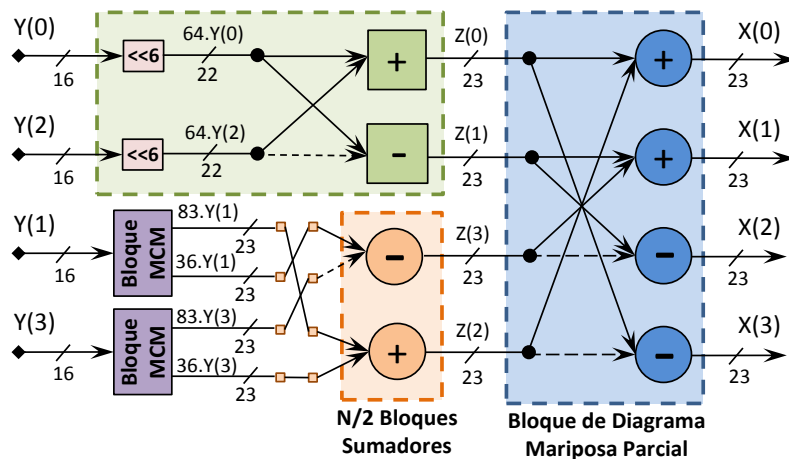


Figura 5.29 Diagrama de bloques de la transformada IDCT-1D-E de 4-puntos

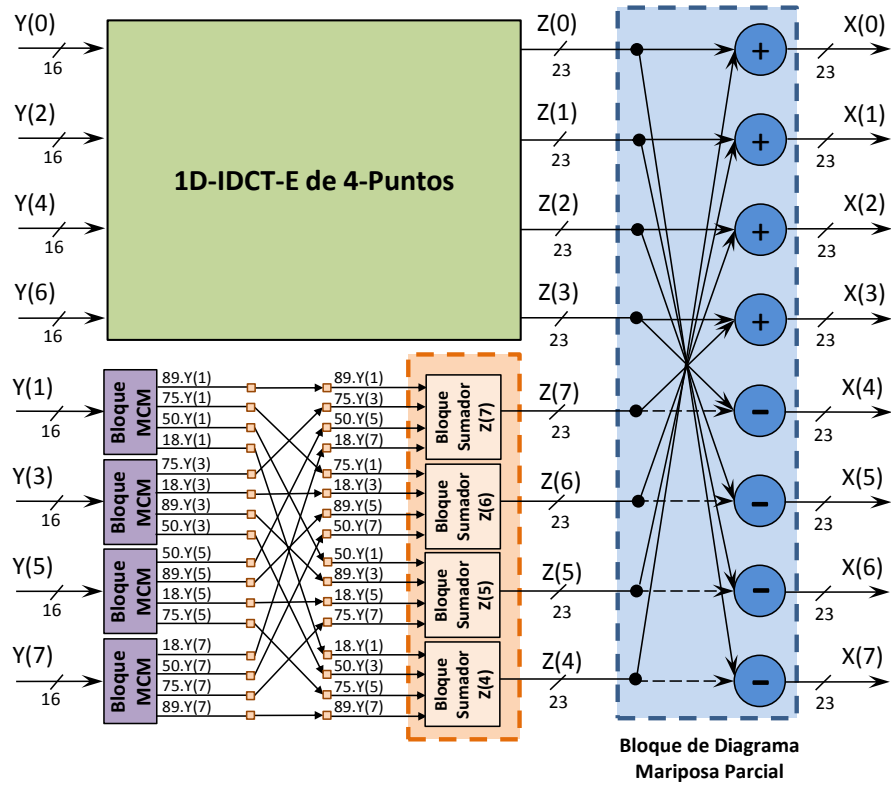


Figura 5.30 Diagrama de bloques de la transformada IDCT-1D-E de 8-puntos

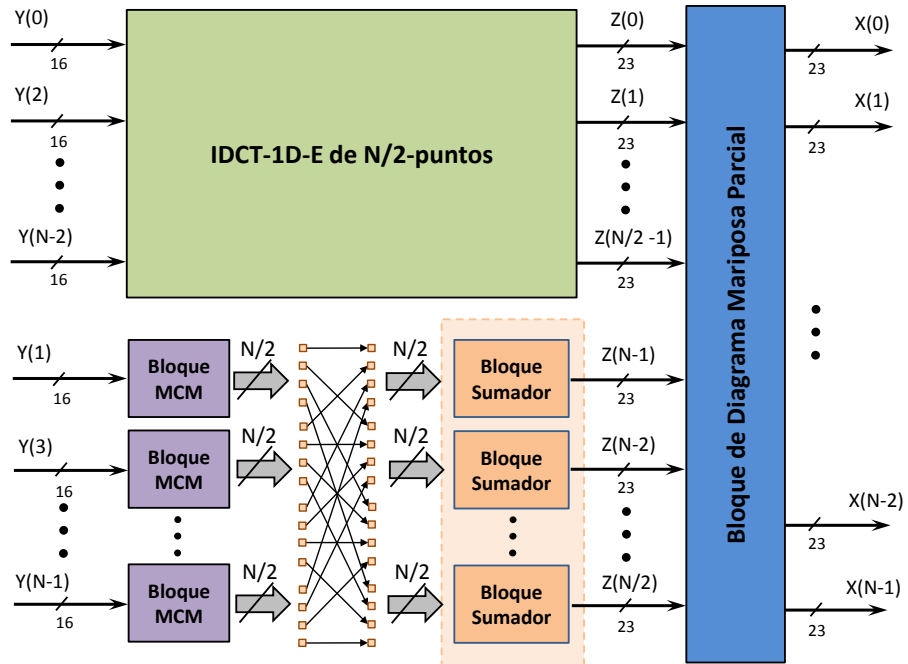


Figura 5.31 Diagrama de bloques genérico de la transformada IDCT-1D-E de N -puntos

5.5.2 TRANSFORMADA IDCT-1D ESTRUCTURADA Y PIPELINE DE N-PUNTOS

El diseño hardware de la transformada IDCT-1D estructurada y pipeline (IDCT-1D-P) se basa en el diseño de la transformada IDCT-1D-E que al igual que la DCT-1D-E, posee una forma regular, esto permite realizar fácilmente una segmentación del diseño y de esta forma incluir registros en la entrada y salida de cada etapa de sumas y restas. De ésta manera es posible introducir nuevos datos en cada ciclo de reloj mejorando el desempeño de la arquitectura, es decir, incrementando en gran medida la frecuencia de operación y el “throughput”.

Los diagramas de bloques de las transformadas IDCT-1D-P de 4 y 8-puntos se pueden observar en las Figuras 5.32 y 5.33, respectivamente. El diagrama de bloques genérico de la transformada IDCT-1D-P de N-puntos se puede observar en la Figura 5.34.

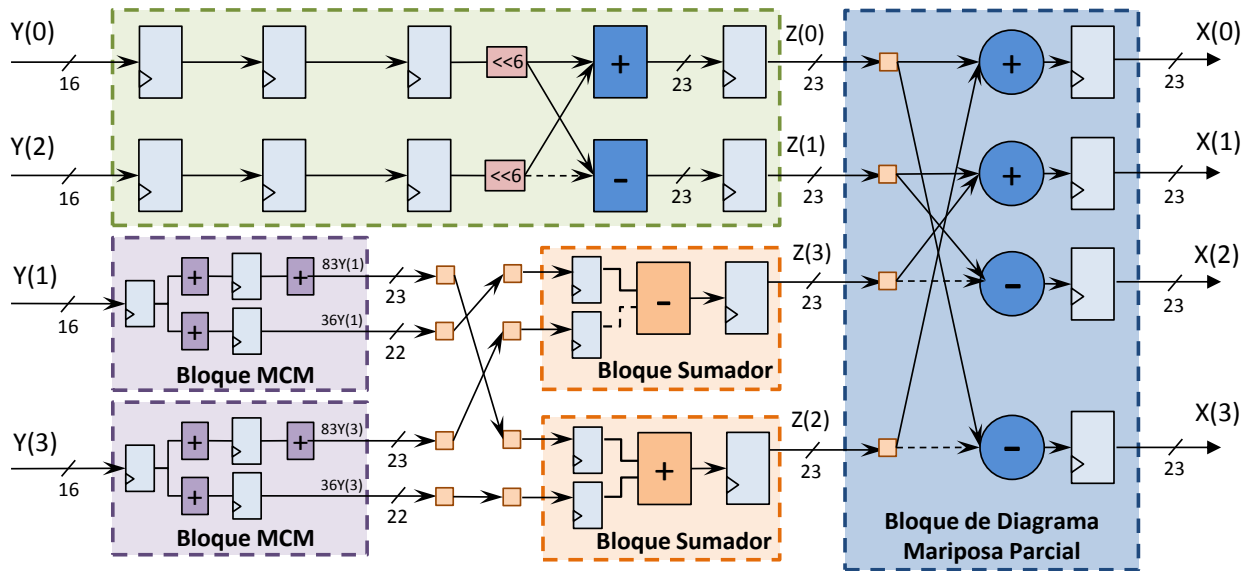


Figura 5.32 Diagrama de bloques de la transformada IDCT-1D-P de 4-puntos

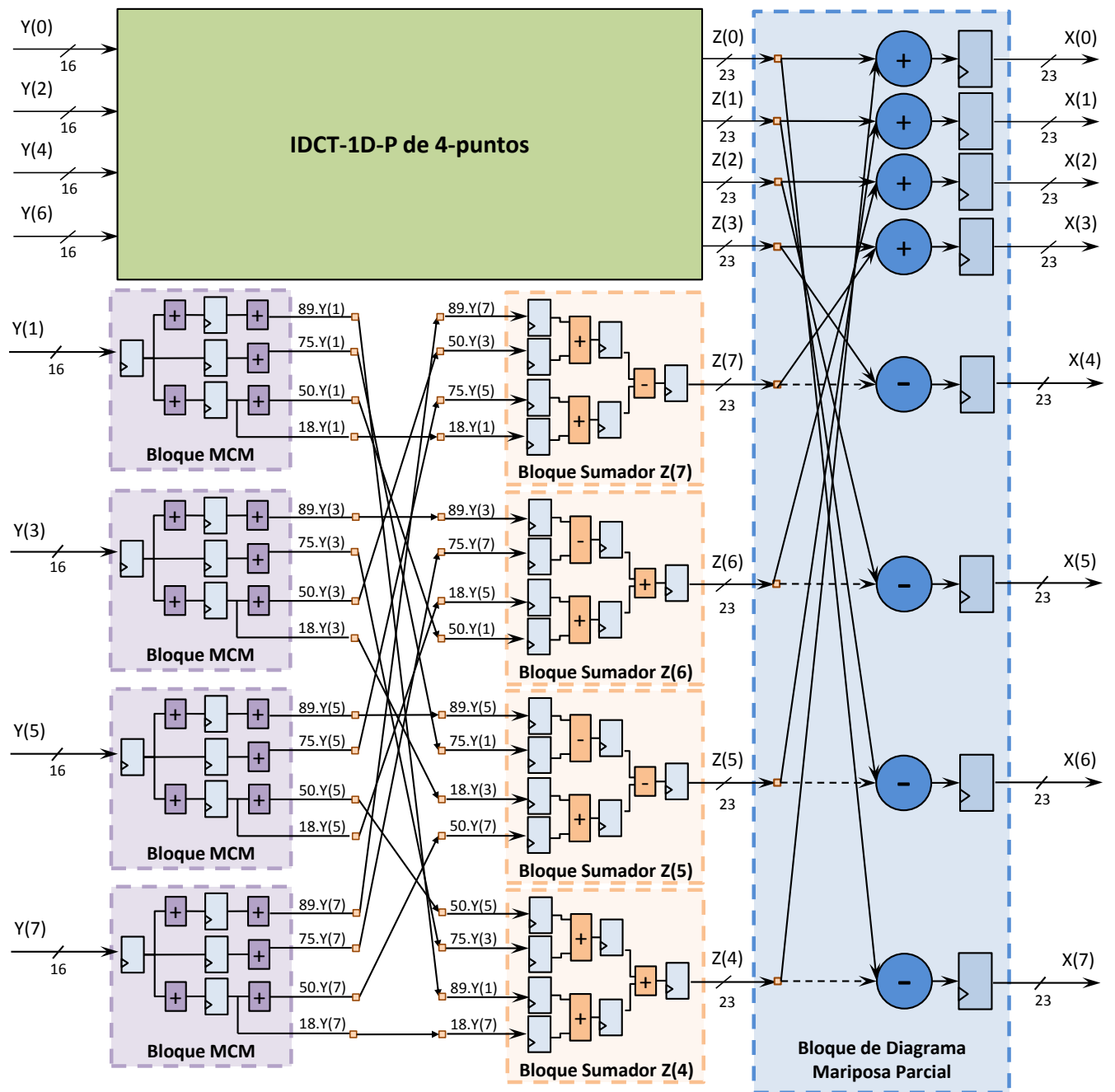


Figura 5.33 Diagrama de bloques de la transformada IDCT-1D-P de 8-puntos

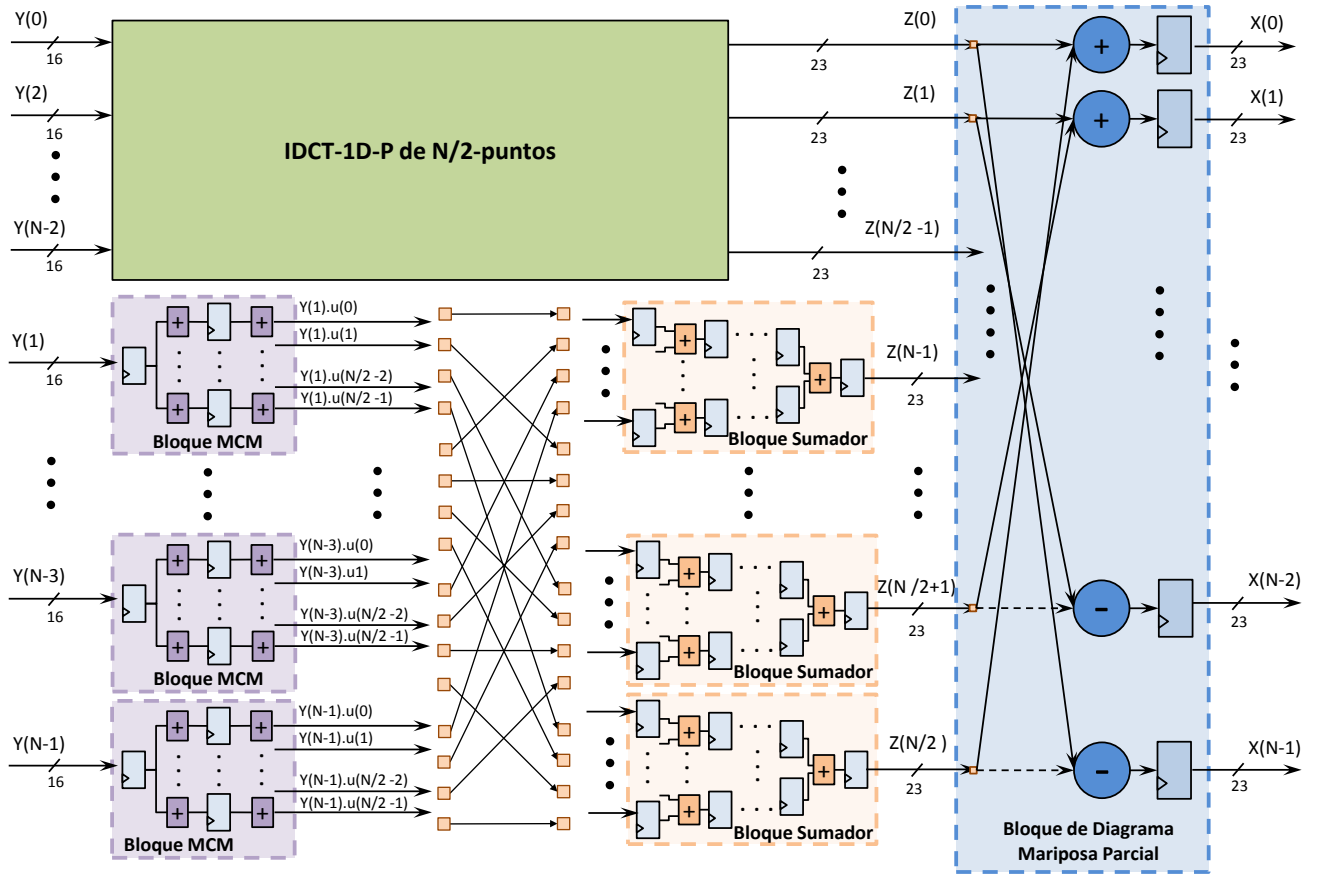


Figura 5.34 Diagrama de bloques genérico de la transformada IDCT-1D-P de N-puntos

5.5.3 TRANSFORMADA IDCT-1D MODIFICADA DE N-PUNTOS

El diseño de la transformada IDCT-1D modificada (IDCT-1D-M) se basa en los bloques MCM y sumadores modificados usados en el diseño de la transformada DCT-1D-M, los cuales fueron descritos en la sección 5.3.3. Es importante tener en cuenta que en este caso, los datos de entrada de los bloques MCM son de 16 bits, y que los bloques sumadores modificados no tienen en cuenta el carry-out cuando el ancho de los datos es mayor o igual a 23 bits. Las estructuras de algunos bloques sumadores modificados para la transformada IDCT-1D-M de 8, 16 y 32-puntos se presentan en las Figuras 5.35, 5.36 y 5.37, respectivamente.

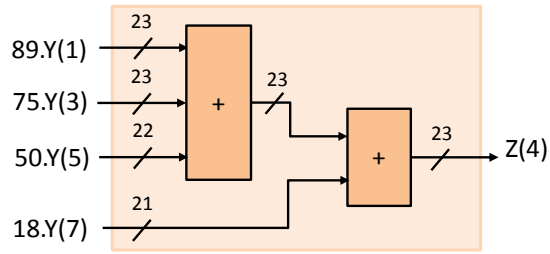


Figura 5.35 Bloque sumador modificado que genera el resultado $Z(4)$ para la transformada IDCT-1D-M de 8-puntos.

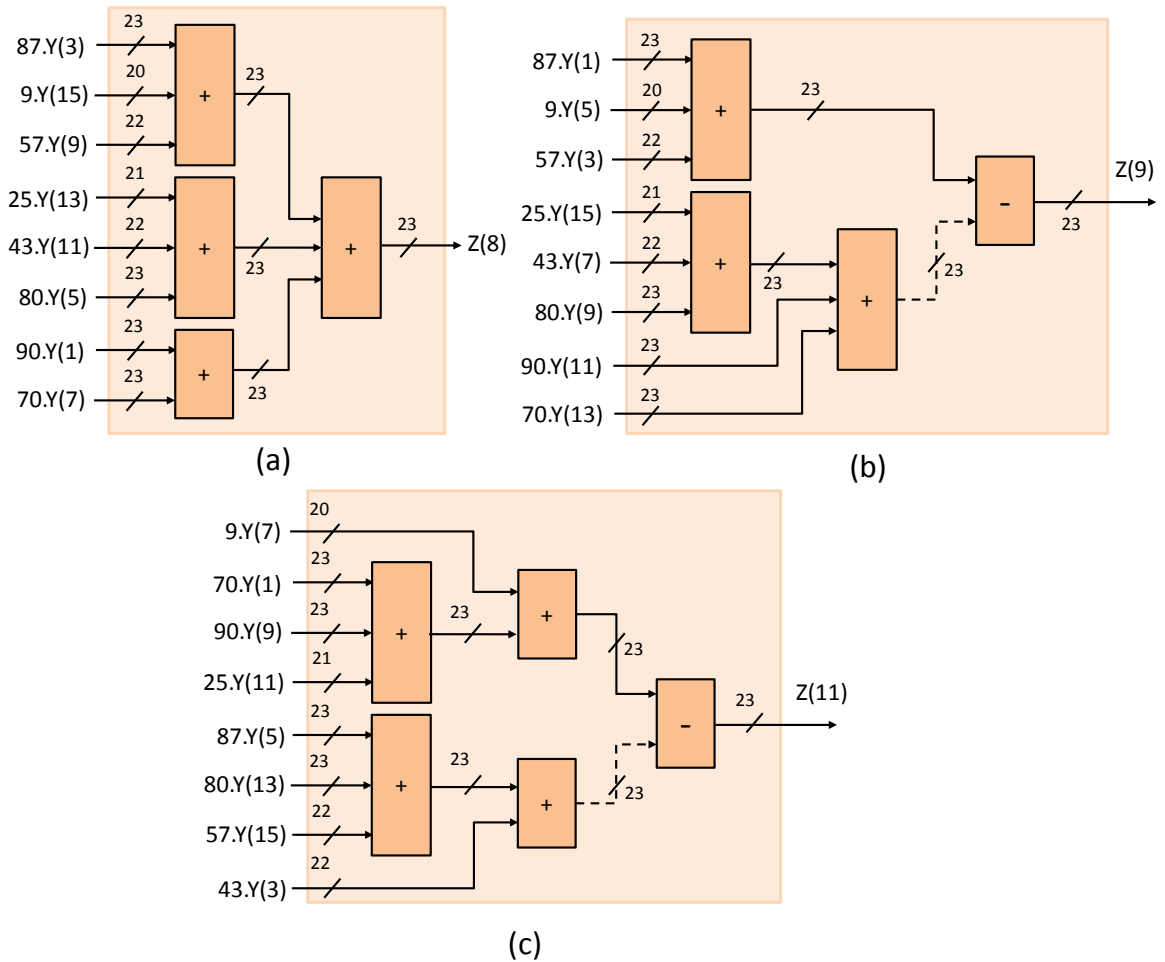


Figura 5.36 Bloques sumadores modificados para la transformada IDCT-1D-M de 16-puntos.

a) Bloque sumador modificado que genera el resultado $Z(8)$. b) Bloque sumador modificado que genera el resultado $Z(9)$. c) Bloque sumador modificado que genera el resultado $Z(11)$.

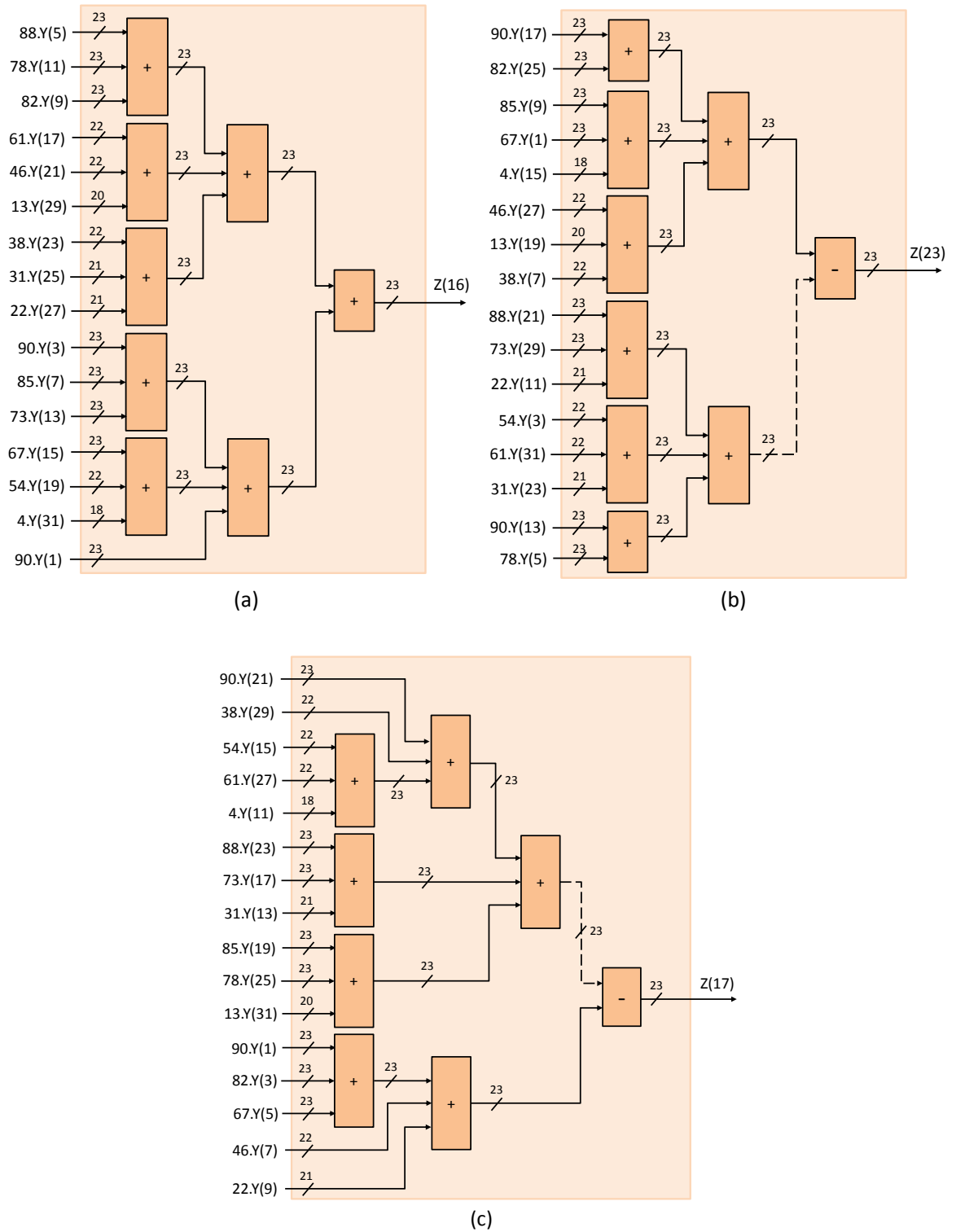


Figura 5.37 Bloques sumadores modificados para la transformada IDCT-1D-M de 32-puntos.

a) Bloque sumador modificado que genera el resultado $Z(16)$. b) Bloque sumador modificado que genera el resultado $Z(23)$. c) Bloque sumador modificado que genera el resultado $Z(17)$.

La Figura 5.35 muestra la estructura del bloque sumador modificado que genera el resultado $Z(4)$ de la transformada DCT-1D-M de 8-puntos. Esta estructura también puede ser utilizada para generar los resultados $Z(5)$, $Z(6)$ y $Z(7)$, para esto es necesario reasignar las entradas del bloque sumador modificado y reconfigurar los bloques funcionales internos ya sea como sumador o restador de tal forma que se obtengan los resultados según la ecuación (5.17).

La Figura 5.36 muestra tres tipos de bloques sumadores modificados para la transformada IDCT-1D-M de 16-puntos. La Figura 5.36 a) muestra el bloque sumador modificado que genera el resultado $Z(8)$. La Figura 5.36 b) muestra al bloque sumador modificado que genera el resultado $Z(9)$. Finalmente, en la Figura 5.36 c) se observa el bloque sumador modificado que genera el valor $Z(11)$. Al igual que en el caso anterior, estas estructuras también pueden ser utilizadas para generar el resto de valores de la segunda mitad del vector Z para la transformada IDCT-1D-M de 16 puntos. En este caso la estructura del bloque sumador modificado que genera el resultado $Z(9)$ también se utiliza para generar los valores $Z(10)$, $Z(13)$ y $Z(14)$. Del mismo modo, la estructura del bloque sumador modificado que genera el resultados $Z(11)$ también se usa para obtener los valores $Z(12)$ y $Z(15)$.

La Figura 5.37 muestra tres tipos de bloques sumadores modificados para la transformada IDCT-1D-M de 32-puntos. La Figura 5.37 a) muestra el bloque sumador modificado que genera el resultado $Z(16)$. La Figura 5.37 b) muestra al bloque sumador modificado que genera el resultado $Z(23)$. Finalmente, en la Figura 5.37 c) se observa el bloque sumador modificado que genera el valor $Z(17)$. En este caso, la estructura del bloque sumador modificado que genera el resultado $Z(16)$ también se utiliza para obtener los valores $Z(18)$, $Z(21)$, $Z(22)$, $Z(25)$, $Z(29)$ y $Z(30)$. La estructura del bloque sumador modificado que genera la salida $Z(23)$ también se utiliza para obtener los resultados $Z(24)$, $Z(27)$, $Z(28)$ y $Z(31)$. Finalmente, la estructura del bloque sumador modificado que genera la salida $Z(17)$ también se utiliza para obtener los resultados $Z(19)$, $Z(20)$ y $Z(26)$.

El diagrama de bloques de la transformada IDCT-1D-M de 4-puntos es el mismo que el presentado en la Figura 5.29, donde los bloques MCM corresponden a los bloques MCM modificados de la Figura 5.13 a).

El diagrama de bloques de la transformada IDCT-1D-M de 8-puntos es el mismo que el presentado en la Figura 5.30. En este caso, solo es necesario reemplazar el bloque IDCT-1D-E de 4- puntos por el bloque IDCT-1D-M de 4-puntos, los bloques MCM por los bloques MCM modificados de la Figura 5.13 b), y utilizar el bloque sumador modificado mostrado en la Figura 5.35.

El diagrama de bloques genérico de la transformada IDCT-1D-M de N-puntos es el mismo que se presentó en la Figura 5.31. En este caso, solo es necesario reemplazar el bloque IDCT-1D-E de N/2-puntos por el bloque IDCT-1D-M de N/2-puntos, y utilizar los bloques MCM modificados y los bloques sumadores modificados para cada tamaño de la transformada IDCT-1D-M, los cuales se presentan en las Figuras 5.13 - 5.15 y 5.35 – 5.37, respectivamente.

5.6 DISEÑO HARDWARE DE LA TRANSFORMADA IDCT-2D

En esta sección se describen los diseños de cada uno de los bloques que conforman la arquitectura hardware de la transformada IDCT-2D para el estándar H.265/HEVC. Al igual que la transformada DCT-2D, la IDCT-2D posee la propiedad de separabilidad, esto permite implementarla en hardware utilizando dos transformadas IDCT-1D y un bloque de trasposición de matrices. También es importante tener en cuenta los procesos de redondeo y escalamiento sobre los datos de salida después de cada etapa de transformación, los cuales están definido por el estándar H.265/HEVC y fueron mencionados en el capítulo anterior.

Los factores de escalamiento tanto para la DCT-2D como para la IDCT-2D aseguran que el ancho de los datos de salida después de una etapa de transformación no sea mayor a 16 bits. Esto permite utilizar el mismo bloque de trasposición tanto para el diseño de la transformada DCT-2D como para el diseño de la IDCT-2D. Además, teniendo en cuenta que los ciclos de latencia inicial de las transformadas unidimensionales tanto directas como inversas son los mismos es posible utilizar la misma unidad de control para los dos diseños.

El proceso para calcular la transformada IDCT-2D consiste en realizar inicialmente la transformación inversa de los datos de entrada usando un bloque IDCT-1D. Los resultados de

la primera transformación son redondeados sumando a cada salida el valor de 2^6 y escalados por un factor de $SIT1 = 2^{-7}$, posteriormente se realiza la trasposición de la matriz resultante y se lleva a cabo la segunda etapa de transformación inversa usando un segundo bloque IDCT-1D. Finalmente, los resultados de la segunda transformación son redondeados sumando a cada salida el valor de $2^{(19-B)}$ y escalados en un factor de $SIT2 = 2^{-(20-B)}$.

En la Figura 5.38 se muestra el diagrama de bloques de un codificador y un decodificador HEVC; y el proceso que se realiza en la etapa de transformación inversa de estos sistemas.

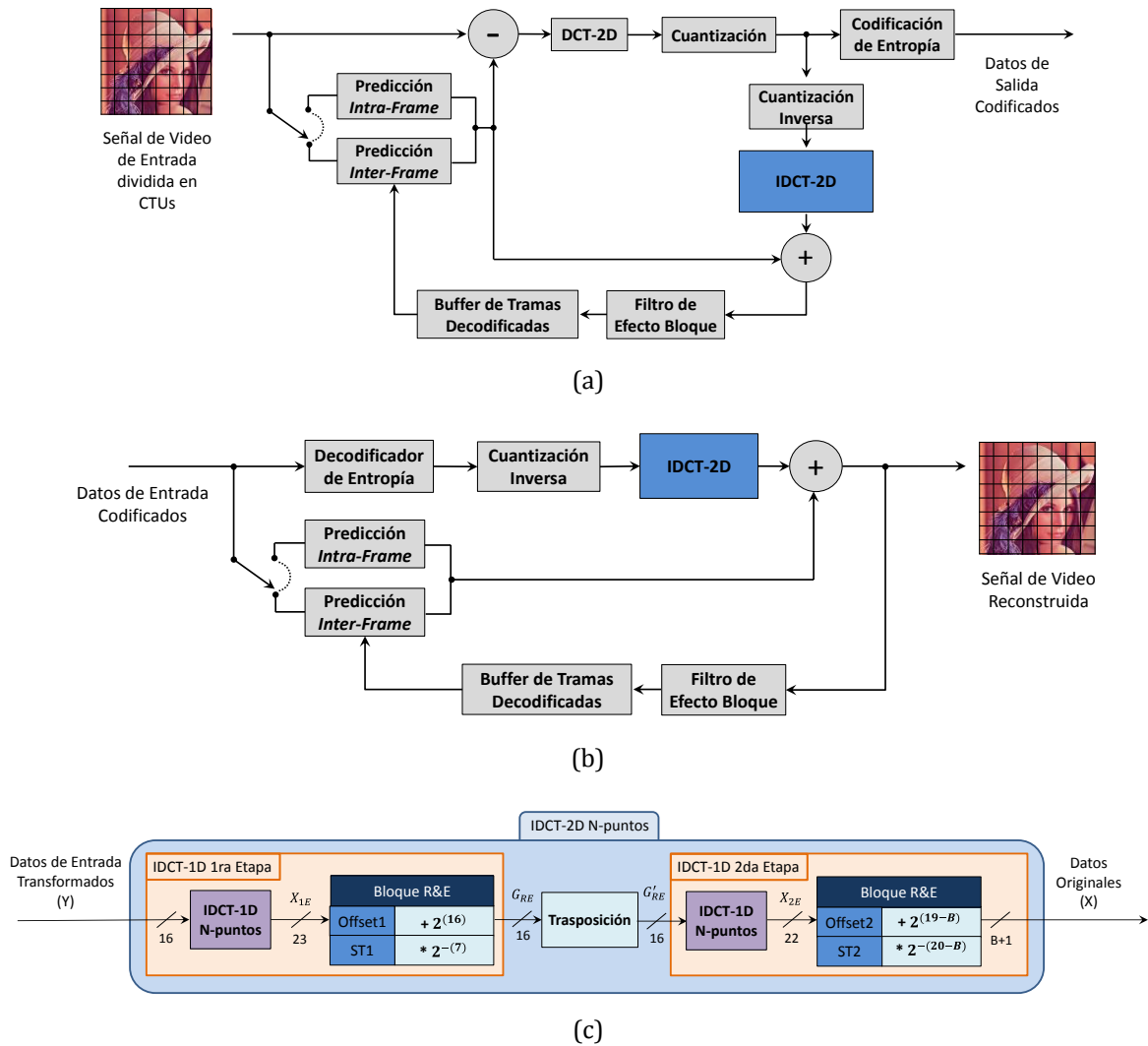


Figura 5.38 a) Diagrama de bloques del codificador de video H.265/HEVC. b) Diagrama de bloques del decodificador de video H.265/HEVC. c) Etapas de procesamiento de la transformada IDCT-2D.

El diagrama de bloques de la arquitectura hardware de la transformada IDCT-2D para el estándar H.265/HEVC es el mismo de la transformada DCT-2D, el cual se mostró en la Figura 5.20. Sin embargo, en este caso, las entradas son datos de 16 bits, las salidas son datos de $B+1$ bits y los bloques utilizados en la primera y segunda etapa de transformación son bloques IDCT-1D. También, se debe tener en cuenta que los factores de redondeo y escalamiento son diferentes.

5.6.1 BLOQUES IDCT-1D

La arquitectura hardware del bloque IDCT-1D utilizado en la primera y segunda etapa de transformación de la transformada IDCT-2D es la misma que se describe en la sección 5.5; las únicas diferencias entre el bloque IDCT-1D de la primera etapa y el de la segunda etapa son el redondeo y el factor de escalamiento.

Para el primer caso, el valor de redondeo es igual a 2^6 y el factor de escalamiento es de 2^{-7} . El factor de escalamiento de la primera etapa de transformación inversa genera datos de 16 bits, tal como se especifica en el estándar HEVC. Además, este factor restringe los valores en un rango dinámico de 15 bits con el fin de asegurar que los errores introducidos por el proceso de cuantización y cuantización inversa no generen un incremento en el tamaño de los datos, es decir que los datos no superen el rango dinámico de 16 bits. Para la segunda etapa de transformación inversa el valor de redondeo es igual a $2^{(19-B)}$ y el factor de escalamiento es de $2^{-(20-B)}$.

Por otra parte, los bloques MCMs incrementan el ancho de los datos en un valor máximo de 7 bits, es decir las salidas pueden ser datos de hasta 23 bits, y los bloques sumadores y el bloque de diagrama mariposa parcial no incrementan el ancho de los datos porque ellos no tienen en cuenta el carry de salida. Lo anterior implica que para obtener datos de salida del mismo ancho que los datos originales es necesario llevar a cabo un escalamiento de $2^{-(22-B)}$ en la segunda etapa de transformación inversa. Entonces, para cumplir con las especificaciones del estándar se desprecia el bit más significativo de las entradas y salidas de los bloques MCM de la segunda transformada IDCT-1D. Lo anterior es posible porque las entradas de la segunda

trasformada IDCT-1D tienen un bit de más debido al sobre-escalamiento de la primera etapa de transformación inversa. Además, el bit más significativo de las salidas de la segunda transformada IDCT-1D se puede despreciar teniendo en cuenta que en el peor de los casos los datos de salida no superan al valor DC multiplicado por 64, lo cual implica un incremento de solo 6 bits. De esta forma es posible llevar a cabo los procesos de redondeo y escalamiento con los factores establecidos por el estándar H.265/HEVC.

El proceso de redondeo y escalamiento efectuado en las salidas de la primera y la segunda etapa de transformación inversa se puede expresarse mediante las ecuaciones 5.19 y 5.20, respectivamente.

$$G_{RE}[i] = \frac{X_{1E}[i] + 2^{(6)}}{2^{(7)}}, \quad \text{para } i = 0, 1 \dots N - 1 \quad (5.19)$$

$$X[i] = \frac{X_{2E}[i] + 2^{(19-B)}}{2^{(20-B)}}, \quad \text{para } i = 0, 1 \dots N - 1 \quad (5.20)$$

Donde $X_{1E}[i]$ representa los N datos de salida de la primera etapa de transformación inversa y $G_{RE}[i]$ es el resultado después del redondeo y el truncamiento. $X_{2E}[i]$ representa los N datos de salida de la segunda etapa de transformación inversa y $X[i]$ su resultado después del redondeo y el truncamiento.

Teniendo en cuenta las anteriores ecuaciones es posible deducir que el redondeo generará un cambio en el valor de $x_{1E}[i]$ solo si el bit en la posición 6 de $X_{1E}[i]$ es igual a 1, lo cual es equivalente a sumar $X_{1E}[i]$ con el valor del bit $X_{1E}[i]$ (6). Lo mismo ocurre para $X_{2E}[i]$ y el bit $X_{2E}[i]$ (19-B). En cuanto al escalamiento, éste puede ser implementado mediante desplazamientos a la derecha o simplemente truncando los bits menos significativos del dato. Lo anterior puede ser expresado mediante las siguientes ecuaciones:

$$G_{RE}[i][15..0] = X_{1E}[i][22... 7] + X_{1E}[i][6], \quad \text{para } i=0, 1, \dots, N-1 \quad (5.21)$$

$$X[i][B...0] = X_{2E}[i][20... 20-B] + X_{2E}[i][19-B], \quad \text{para } i=0, 1, \dots, N-1 \quad (5.22)$$

El diagrama de bloques genérico de la arquitectura hardware de los bloques IDCT-1D empleados en el diseño de la transformada IDCT-2D para el estándar H-265/HEVC se muestran en las Figuras 5.39 y 5.40.

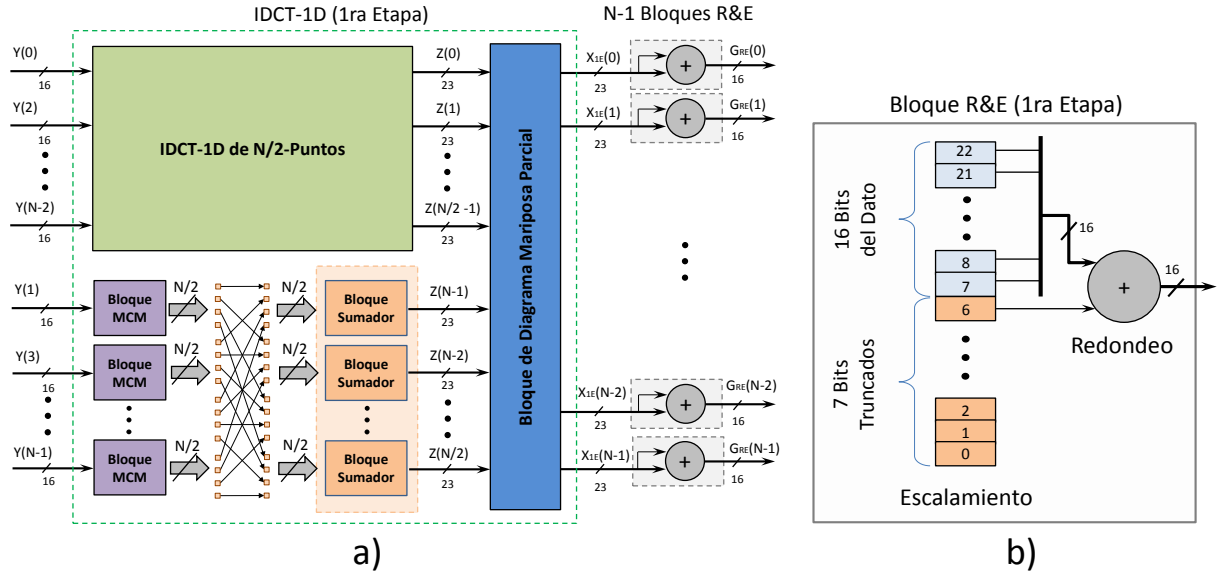


Figura 5.39 a) Diagrama de bloques genérico de la primera etapa de transformación IDCT-1D de N-puntos con redondeo y escalamiento. b) Diagrama detallado del bloque de redondeo y escalamiento - primera etapa

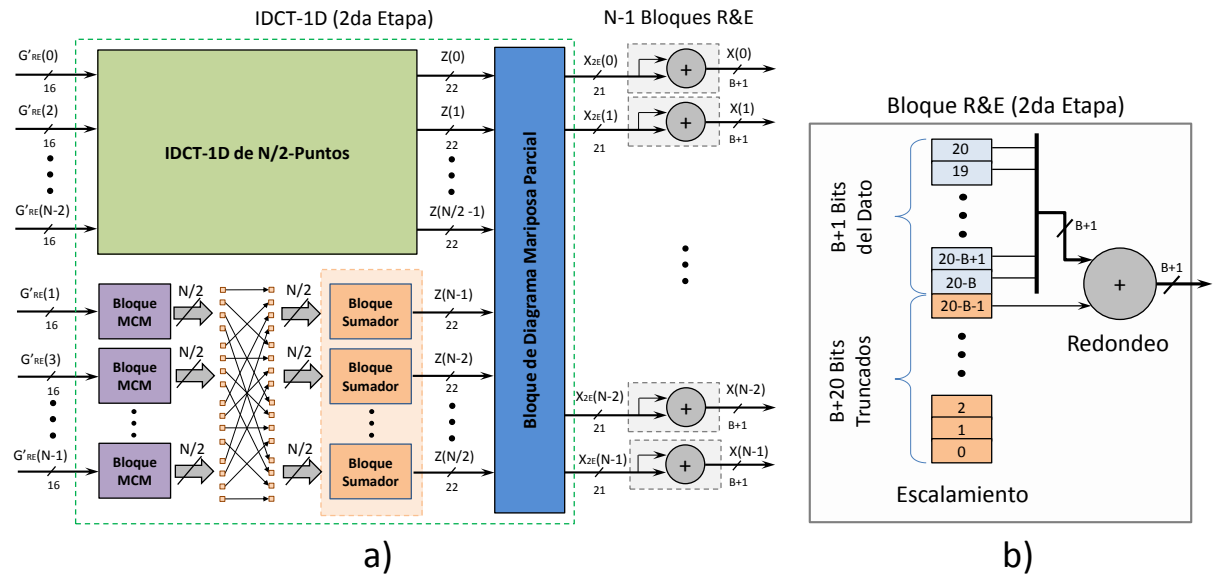


Figura 5.40 a) Diagrama de bloques genérico de la segunda etapa de transformación IDCT-1D de N-puntos con redondeo y escalamiento. b) Diagrama detallado del bloque de redondeo y escalamiento - segunda etapa

5.7 DISEÑO HARDWARE DE LA TRANSFORMADA DCT-1D DE MÚLTIPLES TAMAÑOS

El diseño de la transformada DCT-1D de múltiples tamaños (DCT-1D-T) integra todos los tamaños de la transformada DCT-1D para el estándar H.265/HEVC en un solo núcleo. Lo anterior con el fin de evitar el uso de una arquitectura hardware para cada uno de los tamaños de la transformada DCT-1D, y así disminuir el consumo de recursos de hardware. Esta arquitectura es una solución factible para aplicaciones en donde no se requiera un alto desempeño y/o altas frecuencias de operación.

El diseño de la transformada DCT-1D-T aprovecha la propiedad de recursividad de la transformada DCT, donde una matriz base de tamaño mayor contiene a la matriz base de menor tamaño. En forma general podemos decir que una transformada DCT-1D de N-puntos utiliza dentro de sus bloques funcionales una transformada DCT-1D de $N/2$ -puntos. Teniendo en cuenta lo anterior, se puede concluir que el diseño de la transformada DCT-1D de 32-puntos contiene todos los tamaños de la transformada DCT utilizadas en el estándar H.265/HEVC, es decir que el diseño de la transformada DCT-1D-T de múltiples tamaños se basa en esta consideración.

El diseño de la transformada DCT-1D-T debe habilitar y deshabilitar los diferentes bloques DCT-1D. Por esta razón, es conveniente un diseño estructural en donde la activación o desactivación de las diferentes etapas pueda realizarse fácilmente.

La arquitectura que mejor se acopla a estas necesidades es la transformada DCT-1D-P de 32-puntos, la cual fue descrita en la sección 5.3.2. A partir de este diseño es posible activar o desactivar etapas de registros con el fin de evitar el procesamiento innecesario de datos y la generación de datos erróneos en la salida. Además, es necesario re-direccionar las entradas hacia las diferentes secciones del diseño de acuerdo al tamaño de la transformada seleccionado. Este proceso se realiza utilizando multiplexores de dos entradas y una salida.

Sin embargo, es importante tener en cuenta que el ancho de los datos difiere de una sección a otra. Por lo tanto al re-direccionar los datos de entrada hacia los diferentes bloques DCT-1D es necesario añadir bits en la parte alta. Estos bits deben tomar el valor del bit más significativo del dato con el fin de preservar el signo.

Las señales de habilitación de los registros y multiplexores se controlan mediante una unidad de control interna que tiene 2 bits de entrada y 7 bits de salida. La señal de entrada “Size” de dos bits codificada como 00, 01, 10 y 11 permite seleccionar los tamaños 32, 16, 8 y 4-puntos de la transformada DCT-1D-T, respectivamente. Esta señal controla las señales EN4, EN8, EN16, EN32, MUX4, MUX8 y MUX16 las cuales permiten habilitar o deshabilitar los respectivos registros y multiplexores para procesar los datos de entrada de acuerdo con el tamaño de transformada seleccionado. El diseño hardware de este bloque está basado en la Tabla 5.6.

<i>EN</i>	<i>Size1</i>	<i>Size0</i>	<i>EN4</i>	<i>EN8</i>	<i>EN16</i>	<i>EN32</i>	<i>MUX4</i>	<i>MUX8</i>	<i>MUX16</i>
1	0	0	1	1	1	1	0	0	0
1	0	1	1	1	1	0	0	0	1
1	1	0	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1	0	0

Tabla 5.6 Tabla de verdad de la unidad de control interna de la arquitectura DCT-1D-T

El diagrama de bloques de la arquitectura hardware de la transformada DCT-1D-T para el estándar H.265/HEVC se muestra en la Figura 5.41. En esta figura se observan las diferentes secciones de la arquitectura que son habilitadas por las señales de control EN32, EN16, EN8 y EN4. Estas señales habilitan los registros correspondientes a cada tamaño de la transformada según los colores usados en la nomenclatura ubicada en la parte superior derecha de la figura. Por otra parte, las señales que controlan los multiplexores tales como MUX16, MUX8 y MUX4 seleccionan los datos de entrada que ingresan a cada sección de acuerdo con el tamaño de la transformada seleccionado. Los bloques controlados por estas señales también están diferenciados por medio de colores. Por ejemplo, al seleccionar el tamaño 4-puntos (*size*=“11”), la señal MUX4 habilita el paso de 4 entradas hacia el bloque DCT-1D de 4-puntos, las cuales provienen directamente de las entradas de la arquitectura. De igual forma, la señal EN4 se activa para habilitar los registros de las diferentes etapas de pipeline de la

transformada DCT-1D de 4-puntos. Para el caso del tamaño 8-puntos ($size="10"$), la señal MUX8 habilita el paso de 8 entradas hacia el bloque DCT-1D de 8-puntos, las cuales provienen directamente de las entradas de la arquitectura. En este caso, se activan las señales EN8 y EN4 para habilitar los registros de las diferentes etapas de pipeline, tanto de la DCT-1D de 8-puntos como de la DCT-1D de 4-puntos. El proceso anterior también ocurre al seleccionar el tamaño 16-puntos ($size="01"$) en donde es necesario habilitar los registros de las transformadas DCT-1D de 16, 8 y 4 puntos. Finalmente, para el tamaño 32-puntos no es necesario re-direccionar las entradas por lo tanto todas las señales de control de los multiplexores se mantienen en '0'. Además, es necesario activar todos los registros de las etapas de pipeline mediante las señales EN32, EN16, EN8 y EN4.

Cabe resaltar que en el diagrama de bloques de la Figura 5.41 se han omitido las etapas de registros intermedios de cada uno de los tamaños de la transformada, los cuales también se habilitan o deshabilitan por medio de las señales EN32, EN16, EN8 y EN4.

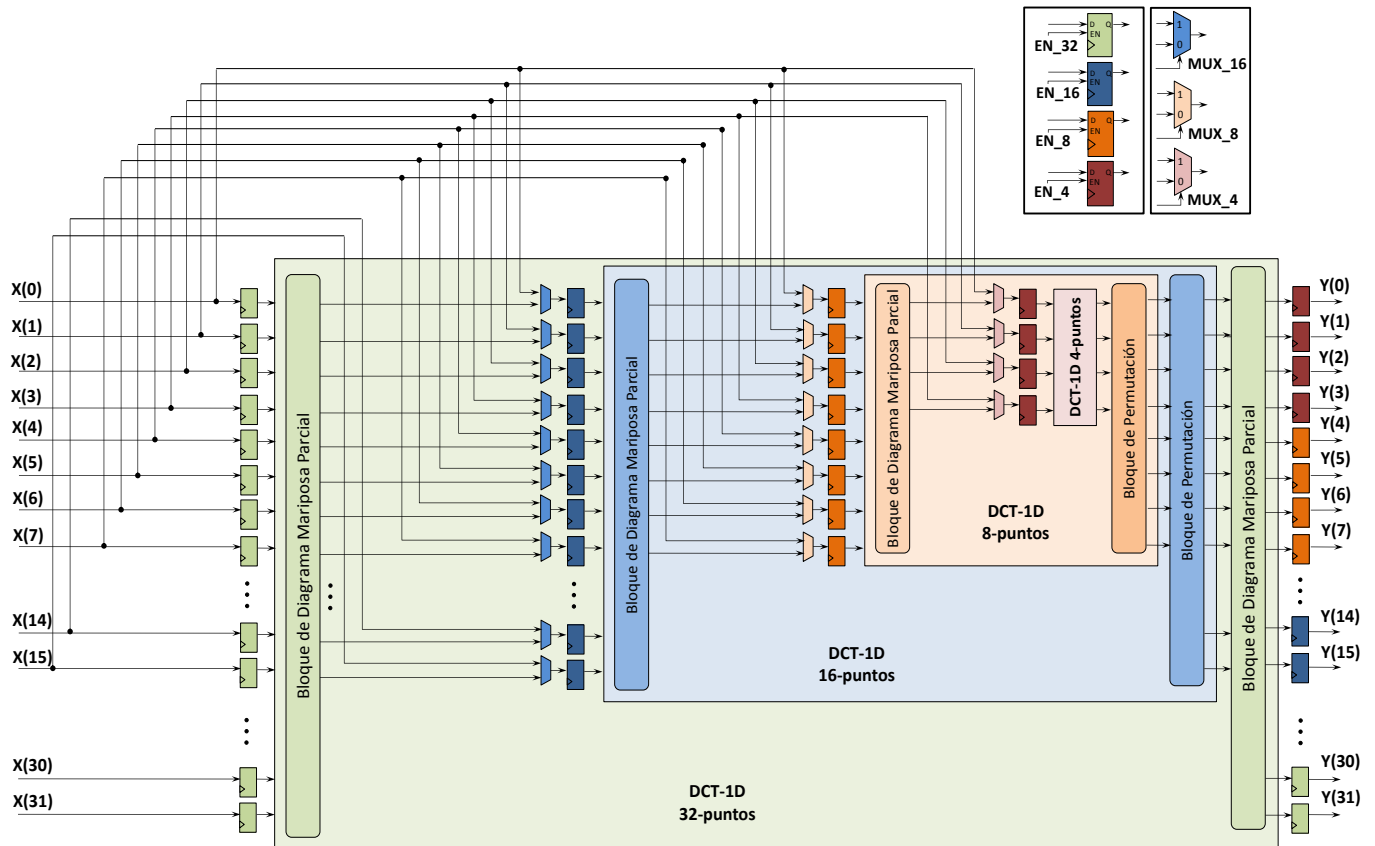


Figura 5.41 Diagrama de bloques de la arquitectura hardware de la transformada DCT-1D-T

Es importante tener en cuenta que los datos de salida de las transformadas DCT-1D para tamaños inferiores a 32-puntos corresponden a los datos en las posiciones pares del vector de datos de entrada de la transformada de tamaño inmediatamente mayor. Es decir, el orden de las salidas de cada uno de los bloques DCT-1D varía de acuerdo al tamaño de la transformada seleccionado. Por ejemplo, al seleccionar el tamaño 4-puntos (*size*="11"), los resultados del bloque DCT-1D de 4-puntos se asignan a las salidas Y(0)...Y(3). Sin embargo, si se selecciona el tamaño 8-puntos (*size*="10"), los resultados del bloque DCT-1D de 4-puntos hacen referencia a las salidas pares del bloque DCT-1D de 8-puntos, por lo tanto estos resultados se asignan a las salidas Y(0), Y(2), Y(4) y Y(6). Lo anterior también se aplica para los otros tamaños de la transformada DCT-1D-T.

Teniendo en cuenta lo anterior es necesario utilizar un bloque de reordenamiento o permutación. Este bloque se encarga de seleccionar el orden correcto de las salidas de cada uno de los bloques DCT-1D de acuerdo con el tamaño de la transformada seleccionado. El diseño hardware de los bloques de permutación es implementado usando MUX 2-1, los cuales son controlados por las señales EN8, EN16 y EN32.

El bloque de permutación para el bloque DCT-1D de 8 puntos tiene 2 vectores de 8 datos de entrada. En el primer vector, las salidas del bloque DCT-1D de 4-puntos se asignan a las posiciones 0...3, mientras que las posiciones 4...7 son generadas por los bloques sumadores de la transformada DCT-1D de 8-puntos. En el segundo vector, las salidas del bloque DCT-1D de 4-puntos se asignan a las posiciones pares 0, 2, 4 y 6, mientras que las posiciones impares 1, 3, 5 y 7 son generadas por los bloques sumadores de la transformada DCT-1D de 8-puntos. En este caso, los bloques MUX 2-1 están controlados por la señal EN8, de esta forma si se selecciona el tamaño 4-puntos (*size*="11"), la señal EN8 se desactiva y habilita el primer vector de datos, por el contrario, si se selecciona el tamaño 8-puntos (*size*="10"), la señal EN8 se activa y habilita el segundo vector de datos. Lo mismo ocurre con los bloques de permutación de los bloques DCT-1D de 16 y 32 puntos, los cuales se controlan por medio de las señales EN16 y EN32, respectivamente. El diagrama de bloques del bloque DCT-1D de 8-puntos utilizando el bloque de permutación se muestra en la Figura 5.42.

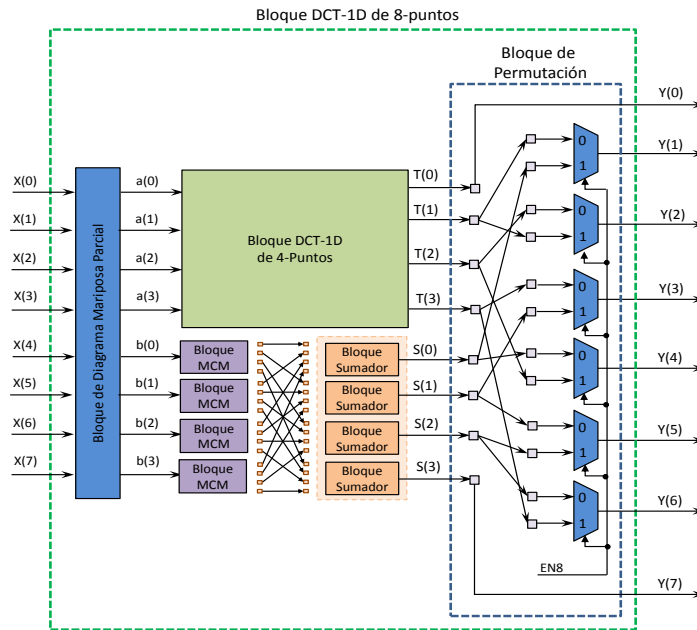


Figura 5.42 Diagrama de bloques del bloque DCT-1D de 8-puntos y el bloque de permutación.

Al igual que en los diseños anteriores, las salidas de la transformada DCT-1D-T deben pasar por un proceso de redondeo y escalamiento. En este caso, los factores de redondeo y escalamiento dependen del tamaño de la transformada y en donde se emplea, primera o segunda etapa de transformación. Entonces, es necesario un bloque adicional para seleccionar los diferentes factores de redondeo y escalamiento de acuerdo con el tamaño de transformada seleccionado. La implementación en hardware de este bloque no se presenta en esta sección, pero se abordará en detalle en el diseño de la transformada DCT-2D de múltiples tamaños.

5.8 DISEÑO HARDWARE DE LA TRANSFORMADA DCT-2D DE MÚLTIPLES TAMAÑOS

Al igual que en los diseños anteriores, la arquitectura hardware de la transformada DCT-2D de múltiples tamaños (DCT-2D-T) se compone de dos bloques DCT-1D y un bloque de trasposición de matrices. Además, como se mencionó anteriormente, es necesario incluir un bloque de redondeo y escalamiento después de cada etapa de transformación, el cual selecciona el factor de redondeo y escalamiento de acuerdo con el tamaño de transformada seleccionado.

El proceso para calcular la transformada DCT-2D-T se inicia con la selección del tamaño de la transformada, luego se realiza la transformación de los datos de entrada mediante un bloque DCT-1D-T, el cual habilita o deshabilita sus etapas de registros y los multiplexores con el fin de redirigir los datos de entrada correspondientes al tamaño de transformada seleccionado. Los resultados de la primera transformación son redondeados, escalados, traspuestos y posteriormente transformados mediante un segundo bloque DCT-1D-T. Finalmente los resultados de la segunda transformación son redondeados y escalados obteniendo como resultado datos de 16 bits. Todo este proceso se controla por medio de una unidad de control, la cual está compuesta por una máquina de estados finita (FSM), un contador y un registro de desplazamiento con entrada y salida paralela.

El diagrama de bloques de la arquitectura hardware de la transformada DCT-2D-T para el estándar H.265/HEVC se muestra en la Figura 5.43.

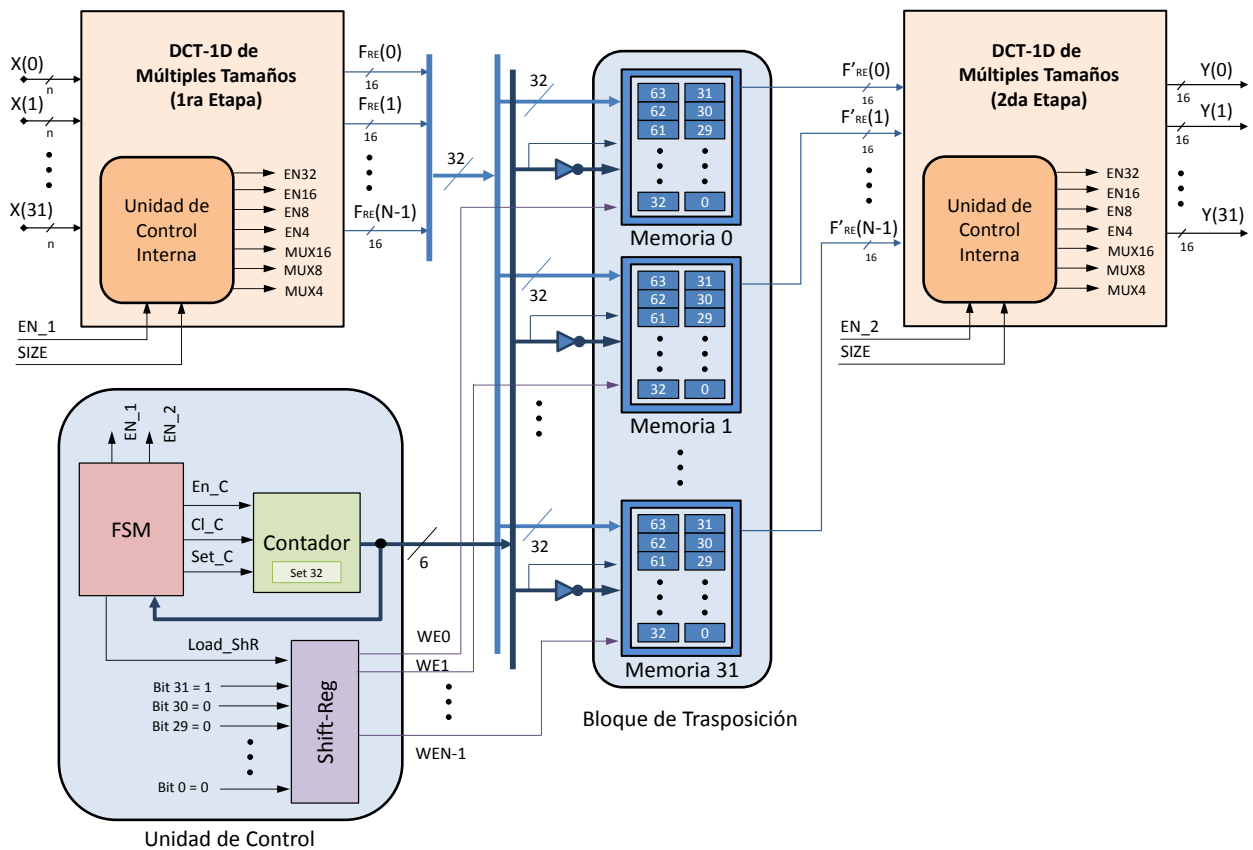


Figura 5.43 Diagrama de bloques de la arquitectura hardware de la transformada DCT-2D-T.

A continuación se describen los diseños de cada uno de los bloques que conforman la arquitectura hardware de la transformada DCT-2D-T para el estándar H.265/HEVC.

5.8.1 BLOQUES DCT-1D DE MÚLTIPLES TAMAÑOS

El diseño hardware del bloque DCT-1D-T utilizado en la primera y segunda etapa de transformación de la DCT-2D-T es el mismo que se describe en la sección 5.7. Los bloques de la primera y segunda etapa, solo difirieren en el ancho de los datos de entrada y en la etapa de redondeo y escalamiento.

Para el primer caso, el ancho de los datos de entrada es igual a $B+1$, el valor de redondeo es igual a $2^{(B+M-10)}$ y el factor de escalamiento es de $2^{-(B+M-9)}$. En secciones anteriores se mencionó que el proceso de redondeo es equivalente a realizar la suma entre el dato de salida y su bit $B+M-10$, y que el escalamiento consiste en despreciar los $B+M-9$ bits menos significativos de la salida. Teniendo en cuenta lo anterior y con un valor de $B=8$, el ancho de los datos de entrada será de 9. El proceso de redondeo consiste en realizar una suma entre la salida y su bit $M-2$, y el escalamiento se obtiene despreciando los $M-1$ bits menos significativos del resultado. Si se tiene en cuenta que la arquitectura de la transformada DCT-1D-T está basada en la arquitectura de la transformada DCT-1D-E de 32 puntos, el valor de M será igual a 5. De esta forma, el redondeo se obtiene sumando la salida y su bit 3, y el escalamiento se obtiene despreciando los 4 bits menos significativos del resultado.

Es importante mencionar que la arquitectura de la transformada DCT-1D de múltiples tamaños re-direcciona las entradas hacia los diferentes bloques DCT-1D internos de menor tamaño por medio de multiplexores, los cuales habilitan el paso de las entradas de acuerdo con el tamaño de la transformada seleccionado.

Cada bloque DCT-1D interno de menor tamaño tiene un bit de más en los datos de entrada con respecto al bloque DCT-1D inmediatamente anterior. Esto es debido a que el bloque de diagrama mariposa parcial de cada uno de los bloques DCT-1D incrementa el ancho de los datos en un bit, el cual corresponde al carry-out/borrow-out de los sumadores y restadores

del diagrama mariposa. Por lo tanto al re-direccionar los datos de entrada hacia los diferentes bloques DCT-1D es necesario añadir bits en la parte alta. Estos bits deben tomar el valor del bit más significativo del dato con el fin de preservar el signo.

Teniendo en cuenta lo anterior, los valores de redondeo y escalamiento mencionados anteriormente solo serán válidos si se selecciona el tamaño 32-puntos, ya que para tamaños menores se generarían errores en los resultados debido a que los valores de los datos de entrada no estarían dentro del rango para el cual fueron calculados los valores de redondeo y escalamiento. Lo anterior produciría un escalamiento excesivo de los datos de salida generando errores en los resultados. Por otra parte, no es posible reducir el factor de escalamiento, ni reducir la cantidad de bits a truncar ya que según el estándar H.265/HEVC se deben obtener datos de salida de 16 bits. Sin embargo, en el caso de las transformadas de tamaños menores a 32-puntos es posible despreciar algunos de los bits más significativos de los datos de salida, los cuales fueron adicionados en el re-direccionamiento de los datos de entrada y de esta forma reducir el número de bits a truncar en la parte menos significativa. Esto evita errores producidos por el factor de escalamiento y al mismo tiempo mantiene el ancho de los datos de salida en 16 bits en la salida. Es importante resaltar que el número de bits truncados tanto en la parte más significativa como en la parte menos significativa de los datos de salida depende del tamaño de la transformada seleccionado. Del mismo modo, la posición del bit del dato de salida que se suma para obtener el redondeo también varía de acuerdo con este tamaño.

Los bits truncados en el proceso de escalamiento y el bit que se debe sumar en el proceso de redondeo para la transformada DCT-1D-T utilizada en la primera etapa de transformación de acuerdo con el tamaño seleccionado se pueden observar en la Tabla 5.7.

<i>Tamaño 1ra DCT-1D</i>	<i>Bits truncados en el Escalamiento</i>	<i>Redondeo</i>
4	Y(19..17) , Y(0)	Y(16..1)+Y(0)
8	Y(19..18) , Y(1..0)	Y(17..2)+Y(1)
16	Y(19) , Y(2..0)	Y(18..3)+Y(2)
32	Y(3..0)	Y(19..4)+Y(3)

Tabla 5.7 Bits a truncar en el escalamiento y bit de redondeo para la transformada DCT-1D-T usada en la primera etapa de transformación de la DCT-2D-T.

En cuanto a la segunda etapa de transformación, el valor del redondeo es igual a $2^{(M+5)}$ y el factor escalamiento es de $2^{-(M+6)}$. Tomando $M=5$, el redondeo es equivalente a realizar una suma entre el dato de salida y su bit 10, y el truncamiento es equivalente a truncar los 11 bits menos significativos del dato de salida. Al igual que en el caso anterior, los valores de redondeo y escalamiento pueden generar errores en los datos de salida cuando se seleccionan tamaños de la transformada menores a tamaño 32-puntos. Por lo tanto, también es necesario truncar bits tanto en la parte más significativa como en la parte menos significativa de los datos de salida según el tamaño de transformada seleccionado. Del mismo modo la posición del bit del dato de salida que se suma para obtener el redondeo también varía de acuerdo con este tamaño. En la Tabla 5.8, se especifican los bits a truncar y el bit de redondeo para la transformada DCT-1D-T utilizada en la segunda etapa de transformación de la transformada DCT-2D-T de acuerdo con el tamaño seleccionado.

<i>Tamaño 2da DCT-1D</i>	<i>Bits Despreciados en el Truncamiento</i>	<i>Redondeo</i>
4	Y(26..24) , Y(7..0)	Y(23..8)+Y(7)
8	Y(26..25) , Y(8..0)	Y(24..9)+Y(8)
16	Y(26) , Y(9..0)	Y(25..10)+Y(9)
32	Y(10..0)	Y(26..11)+Y(10)

Tabla 5.8 Bits a truncar en el escalamiento y bit de redondeo para la transformada DCT-1D-T usada en la segunda etapa de transformación de la DCT-2D-T.

A partir del diagrama de bloques de la arquitectura hardware de la transformada DCT-1D-T que se presentó en la Figura 5.41 es posible observar que los datos de salida Y(0)-Y(3) pueden hacer parte del vector de datos de salida de los bloques DCT-1D de 4, 8, 16 o 32-puntos. Teniendo en cuenta lo anterior, estos cuatro datos de salida pueden presentar uno de los cuatro tipos de escalamiento y redondeo de acuerdo con el tamaño de la transformada seleccionado. Por su parte, los datos de salida Y(4)-Y(7) pueden hacer parte del vector resultante de los bloques DCT-1D de 8, 16 o 32-puntos, por lo tanto pueden presentar uno de los tres tipos de escalamiento y redondeo correspondientes a estos tamaños. A su vez, los datos de salida Y(8)-Y(15) pueden hacer parte del vector resultante de los bloques DCT-1D de 16 o 32-puntos, por lo tanto pueden presentar uno de dos tipos de escalamiento y redondeo. Finalmente los datos de salida Y(16)-Y(31) solo hacen parte del vector resultante del bloques

DCT-1D de 32-puntos y solo presentan el escalamiento y redondeo correspondiente a este tamaño.

Teniendo en cuenta lo anterior, los bloques de redondeo y escalamiento de la primera y segunda etapa de transformación son implementados usando multiplexores 4-1, 3-1 y 2-1, los cuales se encargan de seleccionar el tipo de escalamiento de acuerdo al tamaño seleccionado para los datos de salida $Y(0)$ - $Y(3)$, $Y(4)$ - $Y(7)$ y $Y(8)$ - $Y(15)$, respectivamente. En cuanto a las salidas $Y(16)$ – $Y(31)$, estas solo presentan un tipo de escalamiento por lo cual no es necesario usar multiplexores.

Los datos de entrada de los multiplexores corresponden a los datos de salida de la transformada DCT-1D-T escalados de acuerdo a los diferentes tipos de escalamiento posibles para cada caso. Cabe resaltar que estos datos tienen un bit de más en la parte menos significativa, el cual corresponde al bit que se suma en la etapa de redondeo.

La selección de cada uno de los tipos de escalamiento se realiza usando dos bits de control, para el caso de los MUX 4-1 y 3-1 y 1 bit de control para el MUX 2-1. Para el primer caso los bits de control puede tomar los valores 00, 01, 10 o 11 para seleccionar el escalamiento correspondiente a los tamaños 32, 16, 8 o 4 puntos de la transformada DCT-1D-T, respectivamente. Cabe resaltar que en el MUX 3-1 se debe omitir el escalamiento correspondiente al tamaño 4-puntos ya que los datos que se conectan a este bloque no presentan este tipo de escalamiento. Para el caso del MUX 2-1, la señal de control de 1 bit puede tomar el valor de 1 o 0, para seleccionar el escalamiento correspondiente a los tamaños 32 o 16 puntos de la transformada DCT-1D-T, respectivamente.

Si tenemos en cuenta el diseño de la transformada DCT-1D-T, presentado en la sección 5.7, es posible concluir que los bits de control Size1 y Size0 de la transformada DCT-1D-T también pueden ser utilizados como los bits de selección de los multiplexores en el diseño de los bloques de redondeo y escalamiento. En este caso se toman Size1 y Size0 para controlar la selección en los MUX 4-1 y 3-1, y el bit menos significativo Size 0 para controlar la selección en los MUX 2-1.

Después de realizar la selección del tipo de escalamiento para cada salida de acuerdo con el tamaño seleccionado se procede a realizar el proceso de redondeo en donde se realiza la suma entre el dato de salida del multiplexor y su bit menos significativo obteniendo un resultado de 16 bits. Este proceso se realiza para cada uno de los datos de salida sin importar el tamaño seleccionado.

El diagrama de bloques de la arquitectura hardware de la transformada DCT-1D-T usada en la primera etapa de transformación, y el bloque de redondeo y escalamiento se muestran en la Figura 5.44 a). El diagrama de bloques detallado de uno de los MUX 4-1 utilizado en la implementación del bloque de redondeo y escalamiento de la primera etapa de transformación se muestra en la Figura 5.44 b).

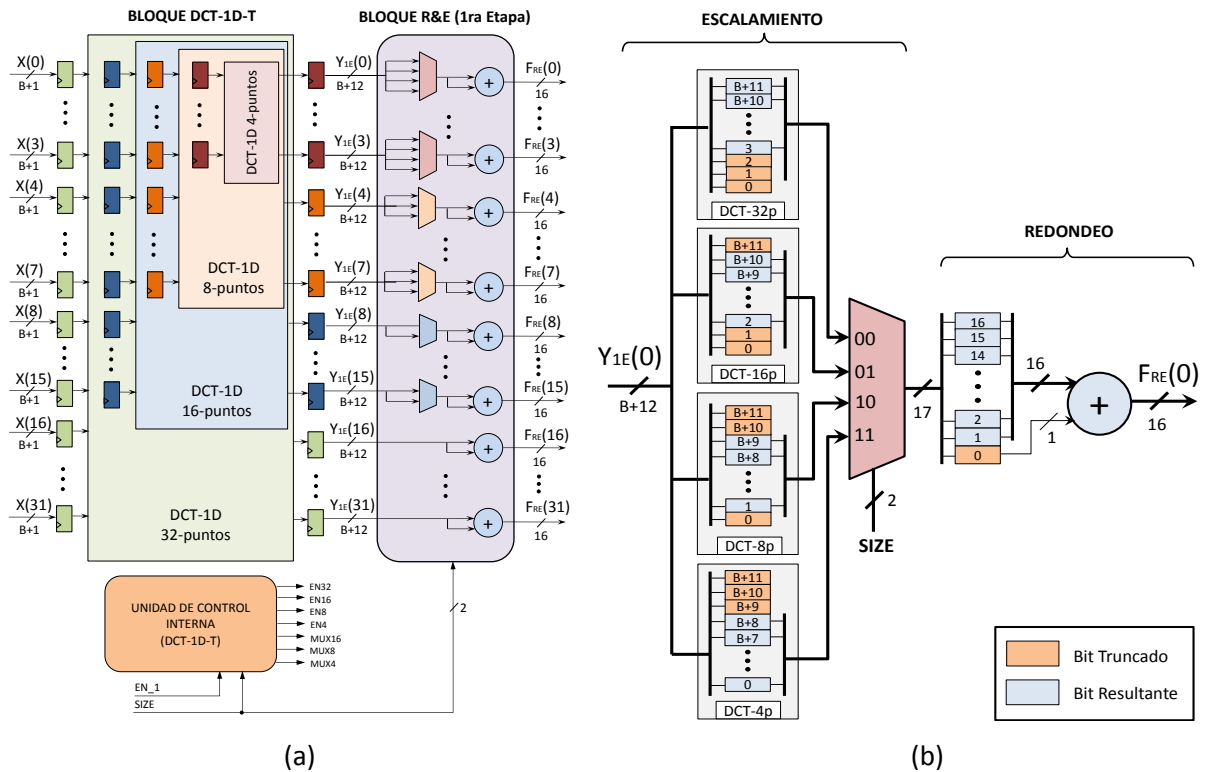


Figura 5.44 a) Diagrama de bloques de la transformada DCT-1D-T usada en la primera etapa de transformación. b) MUX 4-1 utilizado en la implementación del bloque R&E -primera etapa

El diagrama de bloques de la arquitectura hardware de la transformada DCT-1D-T usada en la segunda etapa de transformación, y el bloque de redondeo y escalamiento se muestran en la Figura 5.45 a). El diagrama de bloques detallado de uno de los MUX 4-1 utilizado en la implementación del bloque de redondeo y escalamiento de la segunda etapa de transformación se muestra en la Figura 5.45 b).

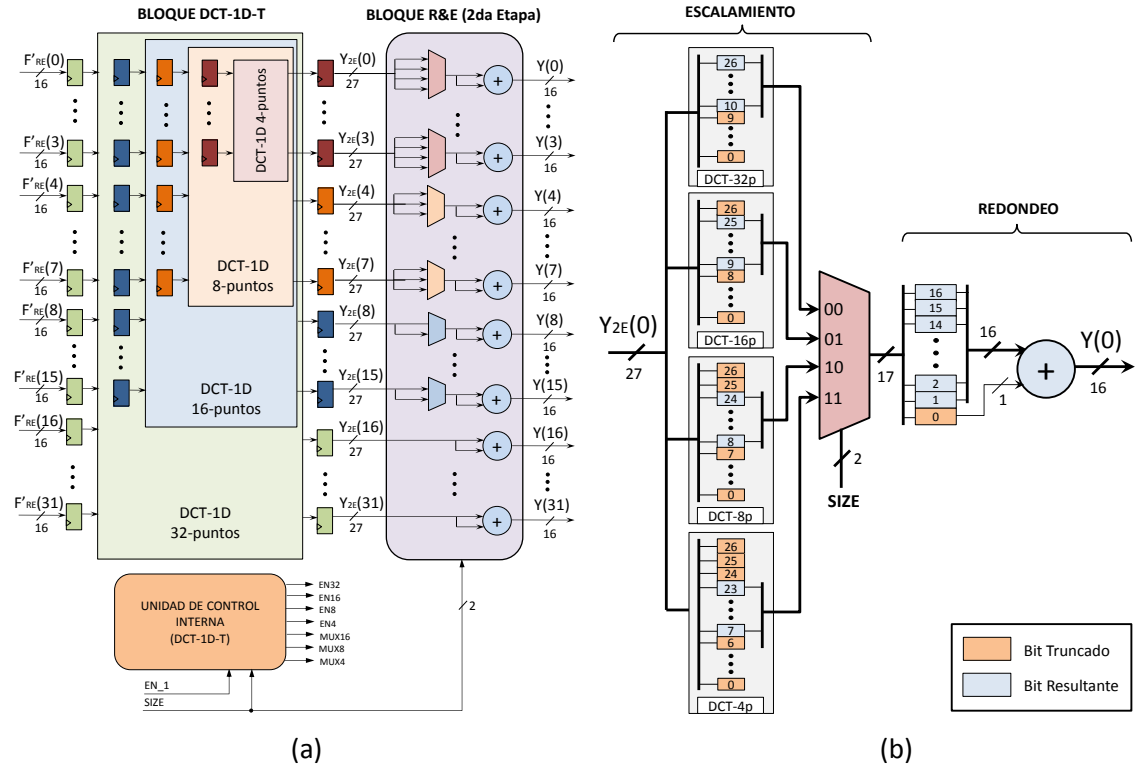


Figura 5.45 a) Diagrama de bloques de la transformada DCT-1D-T usada en la segunda etapa de transformación. b) MUX 4-1 utilizado en la implementación del bloque R&E -segunda etapa

5.8.2 BLOQUE DE TRASPOSICIÓN DE MATRICES

El bloque de trasposición de matrices de la transformada DCT-2D-T está compuesto de 32 memorias RAM modificadas de dos puertos. Cada memoria tiene 2 posiciones de escritura y cada posición se encuentra subdividida en 32 posiciones de 16 bits, es decir 64 posiciones de memoria de 16 bits en total. En cuanto a las señales de control, a cada memoria ingresan una

señal de habilitación de escritura de un bit, una señal de dirección de escritura de un bit y una señal de dirección de lectura de 6 bits.

Las direcciones de lectura y escritura de cada memoria se generan mediante un contador ascendente de 6 bits, que cuenta desde 0 hasta 63. La dirección de escritura es generada por el bit más significativo (MSB) del contador, de este modo cuando el contador genere un valor entre 0 y 31, la dirección de escritura será "0" y cuando el contador genere un valor entre 32 y 63, la dirección de escritura será "1". Por otra parte, la dirección de lectura de 6 bits es generada por los 6 bits del contador negados, esto debido a que las posiciones de memoria están organizadas en forma descendente.

Cada una de las 32 memorias es capaz de almacenar un vector columna de 32 datos transformados de 16 bits provenientes de la primera transformada DCT-1D-T en solo un ciclo de reloj. Además, gracias a su configuración de dos puertos modificados, cada memoria es capaz de leer cada uno de estos 32 datos en forma individual de tal forma que se obtenga la trasposición de la matriz transformada.

A diferencia del bloque de trasposición de la transformada DCT-2D, este bloque realiza la trasposición de las matrices de datos transformados teniendo en cuenta el tamaño de la transformada seleccionado (N). El proceso que realiza el bloque de trasposición de matrices de la transformada DCT-2D-T consta de los siguientes pasos:

- 1) N de Las 32 señales "WE" habilitan la escritura de N de las 32 memorias, una a la vez en cada ciclo de reloj, permitiendo almacenar cada uno de los N vectores columna de la matriz de datos transformados en una memoria diferente. Si tenemos en cuenta que al iniciar el proceso de trasposición el contador tiene un valor de "32", este generará una dirección de escritura igual a "1", Por lo tanto, los N datos de los N vectores columna serán almacenados en las posiciones 63, 62,..., 63-(N-1) de cada una de las N memorias habilitadas. Este proceso toma N ciclos de reloj.
- 2) Posteriormente, se lee cada uno de los N datos de los N vectores columna, en forma individual, es decir se toma un dato desde cada una de las N memorias habilitadas

obteniendo un vector fila de N datos. Si tenemos en cuenta que después de N ciclos de reloj el contador se borra generando una salida igual a "0", este generará una dirección de lectura igual a "63", por lo tanto el primer vector fila estará compuesto de los datos almacenados en las posiciones 63 de cada una de las N memorias habilitadas. Al siguiente ciclo de reloj el valor del contador será "1" generando una dirección de lectura igual a "62", por lo tanto el siguiente vector fila estará compuesto por los datos almacenados en las posiciones 62 de cada una de las N memorias. Este proceso se repite hasta obtener el último vector fila el cual estará compuesto por los datos almacenados en las posiciones 63-($N-1$) de cada una de las N memorias habilitadas.

- 3) El proceso de lectura de datos conlleva otros N ciclos de reloj tiempo en el cual es posible almacenar un segundo conjunto de N vectores columna de una segunda matriz de datos en la segunda posición de escritura de las N memorias habilitadas. En este lapso de tiempo el contador genera valores entre 0 y $N-1$, lo cual genera una dirección de escritura de "0". por lo tanto, los N datos de cada uno de los N vectores columna de la segunda matriz de entrada serán almacenado en las posiciones 31, 30,..., 31-($N-1$) de cada una de las N memorias habilitadas. Este proceso toma N ciclos de reloj que corresponden a los mismos N ciclos de reloj que se emplean en la lectura de los datos de la primera matriz traspuesta.
- 4) Posteriormente, se lee cada uno de los N datos de los N vectores columna de la segunda matriz en forma individual. Si tenemos en cuenta que después de N ciclos de reloj el contador se carga nuevamente con el valor "32", este generará una dirección de lectura igual a "31", por lo tanto el primer vector fila de la segunda matriz traspuesta estará compuesto de los datos almacenados en las posiciones 31 de cada una de las N memorias. Al siguiente ciclo de reloj el valor del contador será "1" generando una dirección de lectura igual a "30", por lo tanto el siguiente vector fila estará compuesto por los datos almacenados en las posiciones 30 de cada una de las N memorias. Este proceso se repite hasta obtener el último vector fila el cual estará compuesto por los datos almacenados en las posiciones 31-($N-1$) de cada una de las N memorias habilitadas. Mientras transcurre este proceso es posible almacenar una nueva matriz de datos en la primera mitad de las N memorias habilitadas. De esta forma se obtiene un procesamiento continuo de los datos sin generar ciclos de latencia intermedios.

5.8.3 UNIDAD DE CONTROL DE LA TRANSFORMADA DCT-2D DE MÚLTIPLES TAMAÑOS

La unidad de control de la arquitectura hardware de la transformada DCT-2D-T para el estándar H.265/HEVC está conformada por un contador, un registro de desplazamiento a la derecha con entrada y salida paralela, y dos FSMs.

- **Contador:** El contador de 6 bits genera las direcciones de lectura y escritura de las 32 memorias RAM de dos puertos del bloque de trasposición. En este caso, el contador debe ser cargado con un valor inicial, el cual es generado por una de las FSMs. Este valor hace que las direcciones de lectura y escritura iniciales de las memorias del bloque de trasposición se sincronicen con el final de la primera etapa de transformación. Es decir, el contador es igual a 32 cuando la primera DCT-1D genera el primer vector columna de la matriz de datos transformados, esto genera una dirección de escritura igual a “1” y de esta forma se da comienzo al proceso de trasposición.

Por otra parte, después de terminar la escritura de la primera matriz de datos transformados, es decir después de N ciclos de reloj, se debe borrar el contador. Este proceso es controlado por una de las FSMs mediante la activación de la señal “CL_C”, esto genera una dirección de escritura igual a “0” y de esta forma se inicia con la escritura de una segunda matriz transformada en la segunda mitad de las memorias del bloque de trasposición. Del mismo modo, después de N ciclos de reloj es necesario volver a cargar el valor 32 en el contador, lo anterior se realiza activando la señal “SET_C”. Este proceso cambia la dirección de escritura de las memorias de trasposición de 1 a 0 y de 0 a 1 cada N ciclos de reloj y mantiene las direcciones de lectura en un rango desde 31 hasta 31-(N-1) y desde 63 hasta 63-(N-1).

- **Registro de Desplazamiento:** El registro de desplazamiento a la derecha con entrada y salida paralela de 32 bits genera la señal de control “WE” de cada una de las 32 memorias de trasposición. Estas señales habilitan la escritura de solo N de las 32 memorias, habilitando una a la vez en cada ciclo de reloj, esto permite almacenar cada uno de los N vectores columna de la matriz de datos transformados en una memoria diferente.

La habilitación de la primera memoria del bloque de transposición se debe realizar justo después de finalizar la primera etapa de transformación, este proceso se lleva a cabo cargando un valor inicial en el registro de desplazamiento el cual es ingresado por medio de la entrada paralela de 32 bits y es cargado cuando la FSM activa la señal “LOAD_SH”. Este valor solo tendrá un ‘1’ en el bit más significativo de sus 32 bits, de esta forma la señal “WE_0” habilitara la escritura de la primera memoria de trasposición y en el siguiente flanco de reloj el primer vector transformado de N datos será escrito en las posiciones 63, 62,..., 63-(N-1) de esta memoria. Luego el bit más significativo del registro de desplazamiento se desplaza a la derecha, esto genera un ‘1’ en la señal “WE_1” la cual habilita la escritura de la segunda memoria de trasposición, en donde se almacena el segundo vector transformado. Este proceso se repite cada ciclo de reloj activando solo una de las N memorias de forma consecutiva hasta almacenar los N vectores transformados.

Teniendo en cuenta que el bloque de trasposición procesa datos continuamente es necesario reiniciar la activación de la señales WE después de N ciclos de reloj, por la tanto después de este lapso de tiempo se debe cargar nuevamente el valor inicial del registro de desplazamiento.

Es importante mencionar que al controlar el instante en el que se carga el dato inicial del registro de desplazamiento se garantiza que solo se habilitaran las N memorias necesarias de las 32 memorias del bloque de trasposición, evitando el almacenamiento de datos inválidos en las otras memorias.

- **FSMs:** La unidad de control de la transformada DCT-2D-T tiene dos FSMs, la primera para controlar la habilitación de los dos bloques DCT-1D-T, y la segunda para determinar el instante en que se cargan los datos iniciales del contador y del registro de desplazamiento.

La sincronización de la señales de control de los bloques de transformación y del bloque de trasposición de la transformada DCT-2D-T se realiza teniendo en cuenta sus ciclos de latencia inicial, los cuales varían de acuerdo al tamaño de la transformada seleccionado. Los ciclos de latencia inicial de los bloques que conforman la arquitectura hardware de la transformada DCT-2D-T se presentan en la Tabla 5.9.

Tamaño DCT-2D	Ciclos de latencia Inicial		
	<i>1ra DCT-1D-T</i>	<i>Trasposición</i>	<i>2da DCT-1D-T</i>
4	5	6	5
8	6	10	6
16	7	18	7
32	8	34	8

Tabla 5.9 Ciclos de latencia de los bloques que conforman la transformada DCT-2D-T.

- ❖ **FSM para el control de habilitación de los bloques DCT-1D-T:** La FSM que controla la habilitación de los bloques DCT-1D-T, habilita el primer bloque DCT-1D justo después de la activación de la señal de inicio (*Start*), manteniéndose en el estado S1 hasta que el contador tome el valor “32”. Lo anterior indica que el proceso de trasposición ha comenzado y es necesario pasar al siguiente al estado S2, en donde se realiza un nuevo ciclo de espera. En este caso la FSM se mantiene en este estado hasta que el proceso de trasposición finalice y el primer vector de datos traspuesto sea leído, esto ocurre cuando el contador es igual a “1”, en este instante la FSM pasa al estado S3 en donde se habilita el segundo bloque DCT-1D. A partir de este momento la FSM mantiene habilitados los dos bloques DCT-1D-T hasta que se reinicie el proceso.

- ❖ **FSM para el control del contador y el registro de desplazamiento:** Esta FSM determina el momento en que se deben activar las diferentes señales de control con el fin de cargar los valores iniciales del contador y del registro de desplazamiento. Para esto, la FSM debe tener en cuenta los 4 posibles valores de latencia inicial de los bloques que conforman la arquitectura de la DCT-2D-T. Lo anterior debido a que el control de las direcciones y el instante de habilitación de escritura de las memorias del bloque de trasposición dependen directamente de estos valores. Inicialmente, la FSM habilita el contador activando la señal “EN_C”, de esta forma el contador incrementa su valor cada ciclo de reloj. El valor del contador nos permite determinar el instante en que finaliza la primera etapa de transformación, de esta forma cuando el valor del contador es igual al número de ciclos de latencia inicial de la transformada DCT-1D-T menos dos, la FSM pasa a un estado en donde se activan las señales “SET_C” y “LOAD_SH”, esto implica que en el siguiente flanco reloj el contador se cargará con el valor 32 y el registro de desplazamiento se cargará con el valor inicial (b“100...00”), respectivamente. Lo anterior da inicio al proceso de trasposición. Después de N-2 ciclos de

reloj, es decir cuando el valor del contador sea igual a $32+(N-2)$, la FSM pasa a un estado en donde se activan las señales “CL_C” y “LOAD_SH”, esto implica que en el siguiente flanco de reloj se borrará el contador y se cargará nuevamente el valor inicial en el registro de desplazamiento. El próximo cambio de estado se realiza cuando el valor del contador es igual a $N-2$, instante en el cual la FSM regresa al estado en donde se activan las señales “SET_C” y “LOAD_SH”. A partir de este punto se inicia un proceso cíclico en donde el contador se carga en forma intercalada con el valor de 32 y 0, y el registro de desplazamiento se carga con el valor inicial. Este proceso se realiza cada N ciclos de reloj hasta que se reinicie el sistema, lo cual permite seleccionar nuevamente un tamaño para la transformada DCT-2D-T, e iniciar nuevamente el cálculo.

En la Figura 5.46 se muestran los diagramas de estado de las dos FSM que componen la unidad de control de la arquitectura hardware de la transformada DCT-2D de múltiples tamaños. Las señales de entrada de las dos FSMs y su funcionalidad están especificadas en la Tabla 5.10. Del mismo modo, las señales de salida de las dos FSMs y su funcionalidad se muestran en la Tabla 5.11.

Las señales de control generadas por la unidad de control para la transformada DCT-2D-T se pueden observar en el diagrama de tiempo de la Figura 5.47. En esta figura, las líneas punteadas de color azul resaltan los ciclos de latencia inicial de la primera transformada DCT-1D-T, las líneas punteadas de color rojo, resaltan los 4 ciclos de reloj usados para almacenar la matriz de datos transformados. Los dos ciclos de latencia de color morado hacen referencia a los ciclos de latencia inicial que presenta la memoria en el proceso de lectura de datos. Finalmente las líneas punteadas de color verde representan los ciclos de latencia inicial de la segunda transformada DCT-1D-T. A partir de este momento el sistema empieza a arrojar vectores de datos transformados cada ciclo de reloj, consiguiendo un procesamiento continuo de los datos y sin ciclos de latencia intermedios. Este proceso continúa hasta que se active la señal de *Reset* la cual reinicia todo el proceso.

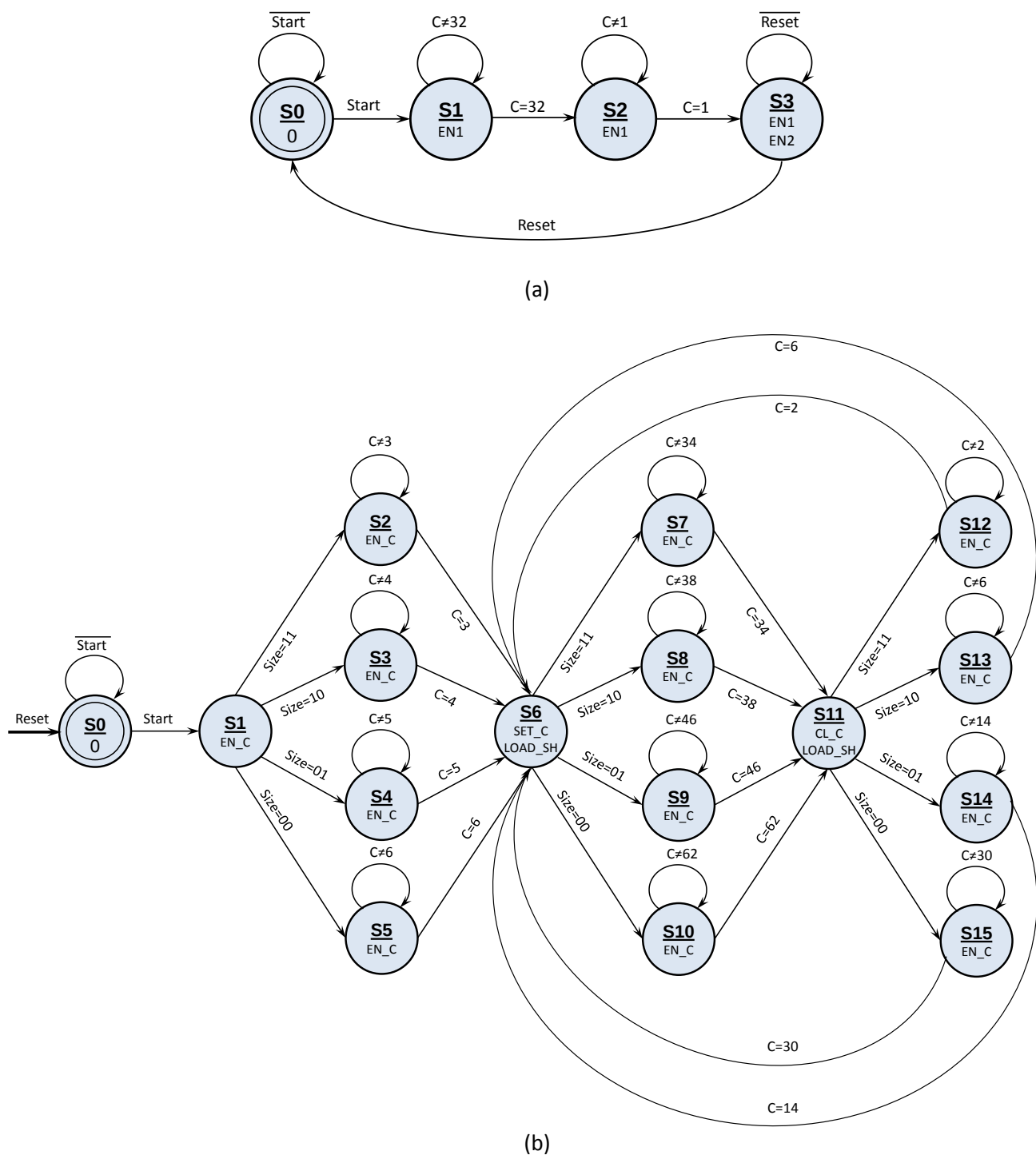


Figura 5.46 Diagramas de estado de las FSMs de la unidad de control de la transformada DCT-2D-T. a) FSM que controla la habilitación de los bloques DCT-1D-T. b) FSM que controla el contador y el registro de desplazamiento.

SEÑAL	DESCRIPCIÓN
Start	Inicia el cálculo de la transformada DCT-2D
Reset	Reinicia el cálculo de la transformada DCT-2D
Size	Selecciona el tamaño de transformada DCT-2D
C	Valor del contador

Tabla 5.10 Señales de entrada de la unidad de control de la transformada DCT-2D-T

SEÑAL	DESCRIPCIÓN
EN_1	Activa el primer bloque DCT-1D
EN_2	Activa el segundo bloque DCT-1D
EN_C	Habilita el contador
SET_C	Posiciona al contador en el valor 32
CL_C	Borra el contador
LOAD_SH	Carga un valor constante en el registro de desplazamiento

Tabla 5.11 Señales de Salida de la Unidad de control de la transformada DCT-2D-T

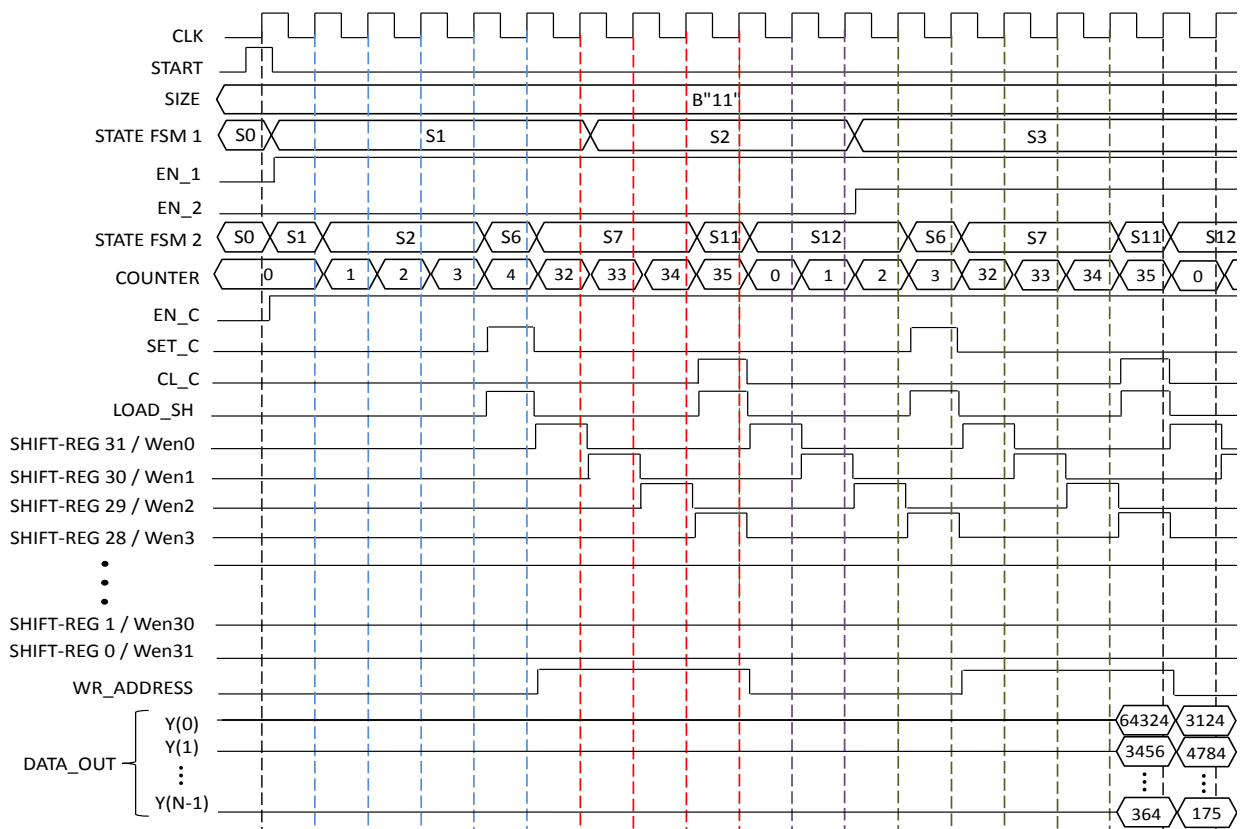


Figura 5.47 Diagrama de tiempo de la unidad de control de la transformada DCT-2D-T.

5.9 DISEÑO HARDWARE DE LA TRANSFORMADA IDCT-1D DE MÚLTIPLES TAMAÑOS

El diseño de la transformada IDCT-1D de múltiples tamaños (IDCT-1D-T) integra todos los tamaños de la transformada IDCT-1D para el estándar H.265/HEVC en un solo núcleo, la cual puede ser utilizada tanto en el codificador como en el decodificador HEVC.

El diseño de la transformada IDCT-1D de múltiples tamaños se basa en el diseño de la transformada IDCT-1D-P de 32 puntos el cual fue descrito en la sección 5.5.2. A partir de este diseño es posible habilitar y deshabilitar las etapas de registros de los diferentes tamaños de transformada IDCT-1D que se encuentran dentro de la arquitectura de la transformada IDCT-1D-P de 32 puntos. De esta forma es posible calcular cualquiera de los tamaños de la transformada IDCT utilizada tanto en el codificador como en el decodificador H.265/HEVC.

Es importante tener en cuenta que el orden de los datos de entrada de la transformada IDCT-1D varía de acuerdo al tamaño de la transformada seleccionada. Por ejemplo, al seleccionar el tamaño 4-puntos (*size*="11"), el bloque IDCT-1D de 4-puntos toma a $Y(0) \dots Y(3)$ como datos de entrada. Sin embargo, si se selecciona el tamaño 8-puntos (*size*="10"), el bloque IDCT-1D de 8-puntos toma a $Y(0) \dots Y(7)$ como datos de entrada, donde los datos en las posiciones pares $Y(0)$, $Y(2)$, $Y(4)$ y $Y(6)$ son procesados por el bloque IDCT-1D de 4-puntos. Lo anterior también se aplica para los otros tamaños de la transformada IDCT-1D-T.

Teniendo en cuenta lo anterior es necesario utilizar un bloque de reordenamiento o permutación. Este bloque se encarga de seleccionar el orden correcto de las entradas de cada uno de los bloques IDCT-1D de acuerdo con el tamaño de la transformada seleccionada. El diseño hardware de los bloques de permutación es implementado usando MUX 2-1, los cuales son controlados por las señales EN8, EN16 y EN32. El diagrama de bloques del bloque IDCT-1D de 8-puntos utilizando el bloque de permutación se muestra en la Figura 5.48.

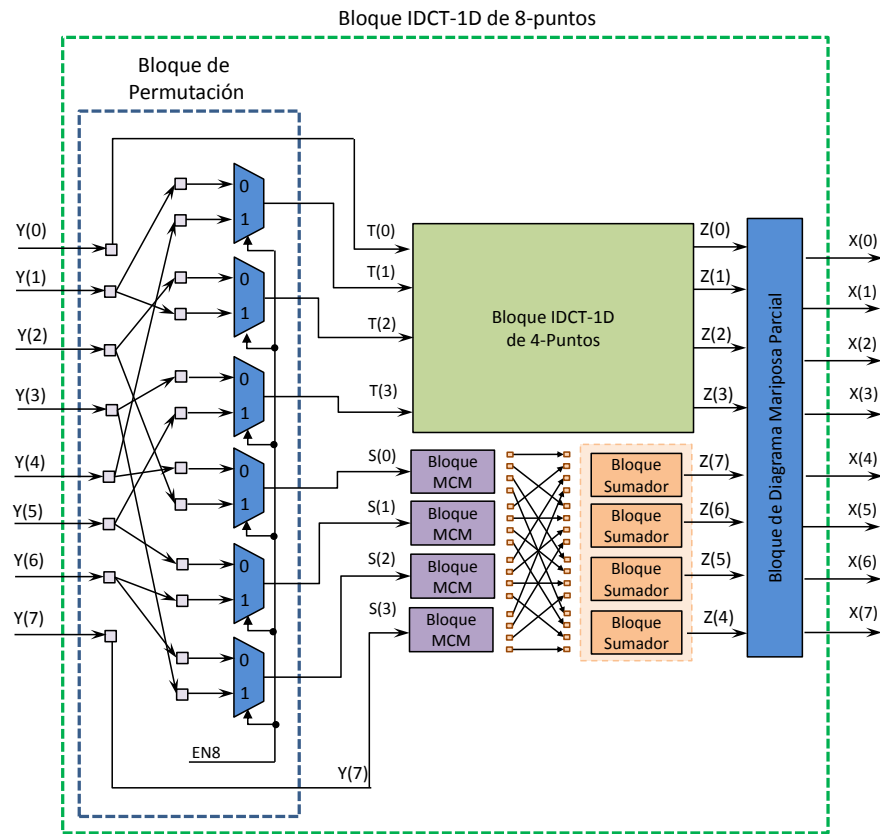


Figura 5.48 Diagrama de bloques del bloque IDCT-1D de 8-puntos y el bloque de permutación.

Por otra parte, las salidas de esta transformada pueden re-direccionarse desde el final de cada bloque IDCT-1D hacia la salida total del diseño. Teniendo en cuenta lo anterior, los datos de salida $X(0) \dots X(3)$ de la transformada IDCT-1D de múltiples tamaños pueden provenir de las transformadas IDCT-1D de tamaño 4, 8, 16 o 32-puntos. Por lo tanto, son necesarios 4 multiplexores 4-1 de 23 bits para seleccionar las salidas correspondientes de acuerdo al tamaño de la transformada seleccionado. Del mismo modo, Las salidas $X(4) \dots X(7)$ pueden provenir de las transformadas IDCT-1D de tamaño 8, 16 o 32-puntos. Por lo tanto son necesarios 4 multiplexores 3-1 de 23 bits para re-direccionar estos 4 datos. En cuanto a las salidas $X(8) \dots X(15)$, estas pueden provenir de las transformadas IDCT-1D de tamaño 16 o 32-puntos, siendo necesarios 8 multiplexores de 2-1 de 23 bits para re-direccionar estos 8 datos. Finalmente, las salidas $X(16) \dots X(31)$ solo pueden provenir de la transformada IDCT-1D de 32-puntos, por lo tanto estos datos no deben ser re-direccionados.

Es importante tener en cuenta que, a diferencia del diseño de la transformada DCT-1D-T, en el diseño de la transformada IDCT-1D-T no es necesario re-direccionar las entradas, esto es debido a que los bloques IDCT-1D se encuentran organizados en forma paralela.

Las señales de habilitación de los registros y los multiplexores se controlan mediante una unidad de control interna que tiene 2 bits de entrada y 6 bits de salida. La señal de entrada “Size” de dos bits codificada como 00, 01, 10 y 11 permite seleccionar los tamaños 32, 16, 8 y 4-puntos de la transformada IDCT-1D-T, respectivamente. Esta unidad habilita o deshabilita las señales EN4, EN8, EN16, EN32 y MUX(1..0), las cuales permiten habilitar o deshabilitar los respectivos registros y multiplexores para procesar los datos de entrada de acuerdo con el tamaño de transformada seleccionado. El diseño hardware de este bloque está basado en la Tabla 5.12.

<i>EN</i>	<i>Size1</i>	<i>Size0</i>	<i>EN4</i>	<i>EN8</i>	<i>EN16</i>	<i>EN32</i>	<i>MUX(1)</i>	<i>MUX(0)</i>
1	0	0	1	1	1	1	0	0
1	0	1	1	1	1	0	0	1
1	1	0	1	1	0	0	1	0
1	1	1	1	0	0	0	1	1

Tabla 5.12 Tabla de Verdad de la Unidad de control interna de la arquitectura IDCT-1D-T

El diagrama de bloques de la arquitectura hardware de la transformada IDCT-1D-T para el estándar H.265/HEVC se muestra en la Figura 5.49. En esta figura se observan las diferentes secciones de la arquitectura que son habilitadas por las señales de control EN32, EN16, EN8 y EN4. Estas señales habilitan los registros correspondientes a cada tamaño de la transformada según los colores usados en la nomenclatura ubicada en la parte superior izquierda de la figura. Por otra parte, las señales MUX(1..0) seleccionan el dato de salida correspondiente al tamaño de la transformada seleccionado. Los bloques controlados por estas señales corresponden a MUX 4-1, MUX 3-1 y MUX 2-1, los cuales también se encuentran diferenciados por medio de colores.

De esta forma, al seleccionar el tamaño 4-puntos (Size = “11”), la señal EN4 habilita los registro de las etapas de pipeline de la IDCT-1D de 4-puntos. De igual forma, la señal MUX(1..0) toma el valor “11” habilitando la selección de las salidas provenientes del bloque

IDCT-1D de 4-puntos en los MUX 4-1. Para el caso del tamaño 8-puntos (Size="10"), las señales EN4 y EN8 habilitan los registros de las etapas de pipeline de los bloques IDCT-1D de 4 y 8-puntos. De igual forma, la señal MUX(1..0) toma el valor "10" habilitando la selección de las salidas provenientes del bloque IDCT-1D de 8-puntos tanto en los MUX 4-1 como en los MUX 3-1. Para el caso del tamaño 16-puntos (Size="01"), las señales EN4, EN8 y EN16 habilitan los registros de las diferentes etapas de pipeline de los bloques IDCT-1D de 4, 8 y 16-puntos. De igual forma, la señal MUX(1..0) toma el valor "01" habilitando la selección de las salidas provenientes del bloque IDCT-1D de 16-puntos en los MUX4-1, MUX3-1 y MUX2-1. Finalmente, al seleccionar el tamaño 32-puntos (Size="00"), todos los registros de la arquitectura son habilitados mediante las señales EN32, EN16, EN8 y EN4. Además, la señal MUX(1..0) toma el valor "00" habilitando la selección de las salidas provenientes del bloque IDCT-1D de 32-puntos en los MUX4-1, MUX3-1 y MUX2-1. Estas salidas corresponden a la primera mitad del vector de datos de salida de esta transformada, la otra mitad de las salidas no requieren re-direccionamiento.

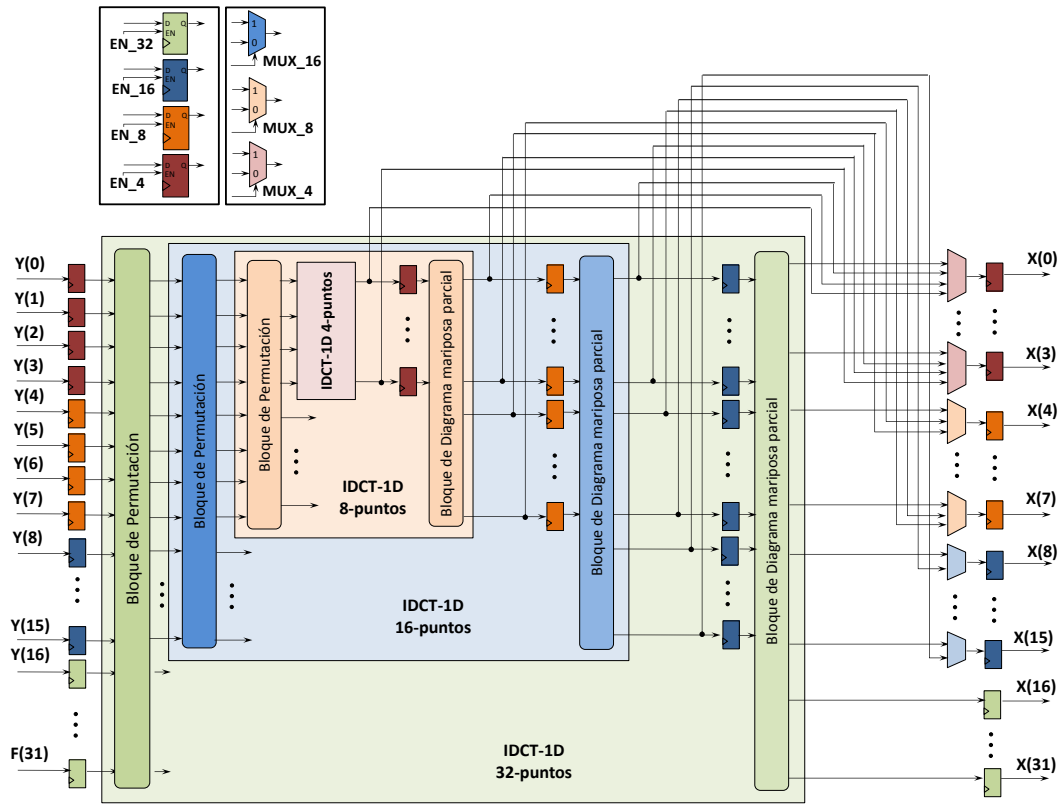


Figura 5.49 Diagrama de Bloques de la arquitectura hardware de la transformada IDCT-1D-T

Cabe resaltar que en el diagrama de bloques de la Figura 5.49 se han omitido las etapas de registros intermedios de cada uno de los tamaños de la transformada IDCT, las cuales también se habilitan o deshabilitan por medio de las señales EN32, EN16, EN8 y EN4 según sea el caso.

Al igual que en los diseños anteriores, las salidas de la transformada IDCT-1D-T deben pasar por un proceso de redondeo y escalamiento. A diferencia de la transformada DCT-1D, los factores de redondeo y escalamiento de la transformada IDCT-1D-T no dependen del tamaño de la transformada seleccionada. Lo anterior es debido a que los bloques sumadores de salida y el bloque de diagrama mariposa parcial no tienen en cuenta el carry de salida, por lo tanto no se presenta un incremento en el número de bits de los datos después de cada una de estas etapas. Esto genera datos de salida con igual número de bits para todos los tamaños de la transformada IDCT-1D.

La implementación en hardware del bloque no se presenta en esta sección, pero se abordará en detalle en el diseño de la transformada IDCT-2D de múltiples tamaños.

5.10 DISEÑO HARDWARE DE LA TRANSFORMADA IDCT-2D DE MÚLTIPLES TAMAÑOS

Al igual que en los diseños anteriores, la arquitectura hardware de la transformada IDCT-2D de múltiples tamaños (IDCT-2D-T) se compone de dos bloques IDCT-1D y un bloque de trasposición de matrices.

El proceso para calcular la transformada IDCT-2D-T inicia con la selección del tamaño de la transformada IDCT-2D, luego se realiza la transformación inversa de los datos de entrada mediante un bloque IDCT-1D-T, el cual habilita o deshabilita sus etapas de registros y multiplexores con el fin de calcular y redirigir los datos de salida correspondientes al tamaño de transformada seleccionada. Los resultados de la primera transformación inversa son redondeados, escalados, traspuestos, y posteriormente transformados mediante un segundo

bloque IDCT-1D-T. Finalmente los resultados de la segunda transformación son redondeado y escalados obteniendo los datos de entrada originales de B+1 bits. de la transformada DCT-2D.

Teniendo en cuenta que los datos intermedios tanto de la transformada DCT-2D-T como de la transformada IDCT-2D-T son datos de 16 bits, es posible utilizar el mismo bloque de trasposición que se utilizó en el diseño de la transformada DCT-2D-T. Además ya que los ciclos de latencia inicial de las transformadas unidimensionales tanto directas como inversas son los mismos, también es posible utilizar la misma unidad de control.

El diagrama de bloques de la arquitectura hardware de la transformada IDCT-2D-T es el mismo de la transformada DCT-2D-T que se presentó en la Figura 5.43. Sin embargo, en este caso, las entradas son datos de 16 bits, las salida son datos de B+1 bits y los bloques utilizados en la primera y segunda etapa de transformación son bloques IDCT-1D-T.

El diseño de los bloques IDCT-1D-T utilizados en el diseño de la transformada IDCT-2D-T para el estándar H.265/HEVC se describen a continuación.

5.10.1 BLOQUES IDCT-1D DE MULTIPLES TAMAÑOS

La arquitectura hardware del bloque IDCT-1D-T utilizado en la primera y segunda etapa de transformación es el mismo que se describe en la sección 5.9. Los bloques de la primera y segunda etapa, solo difirieren en el ancho de los datos de entrada, y en la etapa de redondeo y escalamiento.

Para el primer caso el valor de redondeo es igual a 2^6 y el factor de escalamiento es de 2^{-7} . Por lo tanto, el redondeo es equivalente a realizar la suma entre el dato de salida y su bit 6 y el escalamiento consiste en despreciar los 7 bits menos significativos del dato de salida. Para el segundo caso el valor de redondeo es igual a $2^{(19-B)}$ y el factor escalamiento es de $2^{-(20-B)}$. Con un valor de B=8, el proceso de redondeo es equivalente a realizar la suma entre la salida y su bit 11 y el escalamiento se obtiene truncando los 12 bits menos significativos del resultado.

Por otra parte, los bloques sumadores de salida y el bloque de diagrama mariposa parcial de la transformada IDCT-1D, no tienen en cuenta el carry-out cuando el ancho de los datos es mayor o igual a 23 bits. Lo anterior debido a que en el peor de los casos los valores resultantes no superan al valor del coeficiente DC multiplicado por 64. Por lo tanto todas las salidas del bloque IDCT-1D-T tendrán el mismo número de bits para cualquier tamaño seleccionado.

Teniendo en cuenta lo anterior, se concluye que el proceso redondeo y escalamiento de los bloques IDCT-1D-T no depende del tamaño seleccionado. Por lo tanto, es posible utilizar los bloques de redondeo y escalamiento de los bloques IDCT-1D de la transformada IDCT-2D de 32-puntos, los cuales fueron descritos en la sección 5.6.1.

Los diagramas de bloques de los bloques de redondeo y escalamiento utilizados en la primera y segunda etapa de transformación inversa de la transformada IDCT-2D-T son los mismos que se presentaron en la Figura 5.39 b) y Figura 5.40 b), respectivamente. Estos bloques realizan el redondeo y escalamiento de los datos de salida de la arquitectura IDCT-1D-T presentada en la Figura 5.49, la cual es utilizada tanto en la primera como en la segunda etapa de transformación inversa.

Capítulo 6

SIMULACIÓN FUNCIONAL

6.1 INTRODUCCIÓN

En el capítulo anterior, se describieron cada una de las arquitecturas hardware propuestas para la transformada DCT utilizada en el estándar H.265/HEVC, tanto unidimensionales como bidimensionales, así como sus respectivas transformadas inversas. También, se presentó una arquitectura hardware parametrizable de la transformada DCT-2D tanto directa como inversa que incluye a todos los tamaños de la transformada en un solo núcleo.

En éste capítulo, se presentan la simulación funcional de cada uno de los diseños propuestos, la cual fue realizada utilizando el software Modelsim, y verificada usando Matlab.

Este capítulo está organizado de la siguiente manera. Inicialmente en la sección 6.2 se presenta la simulación funcional de los diseños de la transformada DCT que incluye a las transformadas DCT-1D-E, DCT-1D-M, DCT-1D-P para el caso de las transformadas unidimensionales y a las transformadas DCT-2D-E, DCT-2D-M, DCT-2D-P para el caso de las transformadas bidimensionales. En la sección 6.3 se presenta la simulación funcional de los diseños de la transformada DCT inversa que incluye a las transformadas IDCT-1D-E, IDCT-1D-M, IDCT-1D-P para el caso de las transformadas unidimensionales inversas y a las transformadas IDCT-2D-E, IDCT-2D-M, IDCT-2D-P para el caso de las transformadas bidimensionales inversas. En la sección 6.4 se presenta la simulación funcional de la transformada DCT de múltiples tamaños unidimensional y bidimensional. Finalmente, en la sección 6.5 se presenta la simulación funcional de la transformada DCT inversa de múltiples tamaños unidimensional y bidimensional.

6.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT

En esta sección se presenta la simulación funcional de cada uno de los diseños propuestos para las transformadas DCT-1D y DCT-2D. Es importante tener en cuenta que los resultados de simulación de las transformadas DCT-1D-E y DCT-1D-M son exactamente iguales. Por su parte la transformada DCT-1D-P presenta los mismos resultados numéricos que las otras dos, pero en este caso el cálculo de los resultados requiere de más ciclos de reloj, ya que esta transformada posee varias etapas de pipeline. Lo anterior también se aplica para el caso de las transformadas DCT-2D.

6.2.1 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT-1D

Las simulaciones de las transformadas DCT-1D se llevan a cabo ingresando 4 vectores de entrada, donde cada vector consta de N datos de 9 bits, siendo N el tamaño de la transformada. Cabe resaltar que en este caso los datos de entrada son números positivos en un rango entre 0 y 255, por lo tanto el bit más significativo de estos datos siempre será igual a '0'.

Los resultados de la simulación funcional son verificados en Matlab, en este caso se calculan los resultados de la transformada DCT-1D teniendo en cuenta el Algoritmo 1, el cual fue presentado en el capítulo 4, sección 4.2. También, es importante tener en cuenta el proceso de redondeo y escalamiento que se debe realizar sobre los datos de salida de esta transformada. En este caso, este proceso se lleva a cabo utilizando los factores de redondeo y escalamiento para la primera etapa de transformación DCT-1D, los cuales fueron descritos en la sección 4.7 del capítulo 4.

Los resultados de las simulaciones funcionales y su verificación usando Matlab para las transformadas DCT-1D-E/DCT-1D-M de 4, 8, 16 y 32-puntos se observan en las Figura 6.1, 6.2, 6.3 y 6.4, respectivamente.

En cuanto a los resultados de simulación de las transformadas DCT-1D-P de 4, 8, 16 y 32-puntos, estos son los mismos que se presentan en las Figuras 6.1, 6.2, 6.3 y 6.4, con la diferencia de que en estos casos los resultados son obtenidos después de 5, 6, 7 y 8 ciclos de reloj, respectivamente.

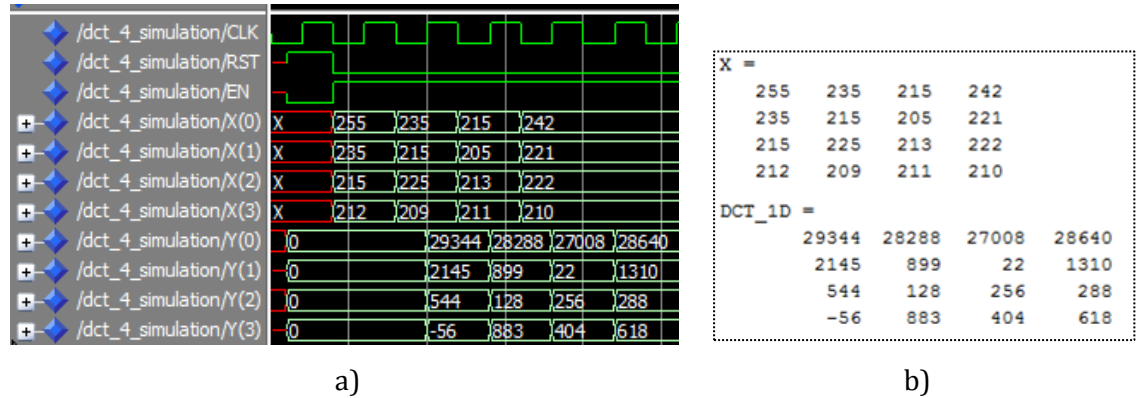


Figura 6.1 a) Resultados de la simulación funcional de la transformada DCT-1D-E/DCT-1D-M de 4-puntos.

b) Verificación de los resultados usando Matlab.

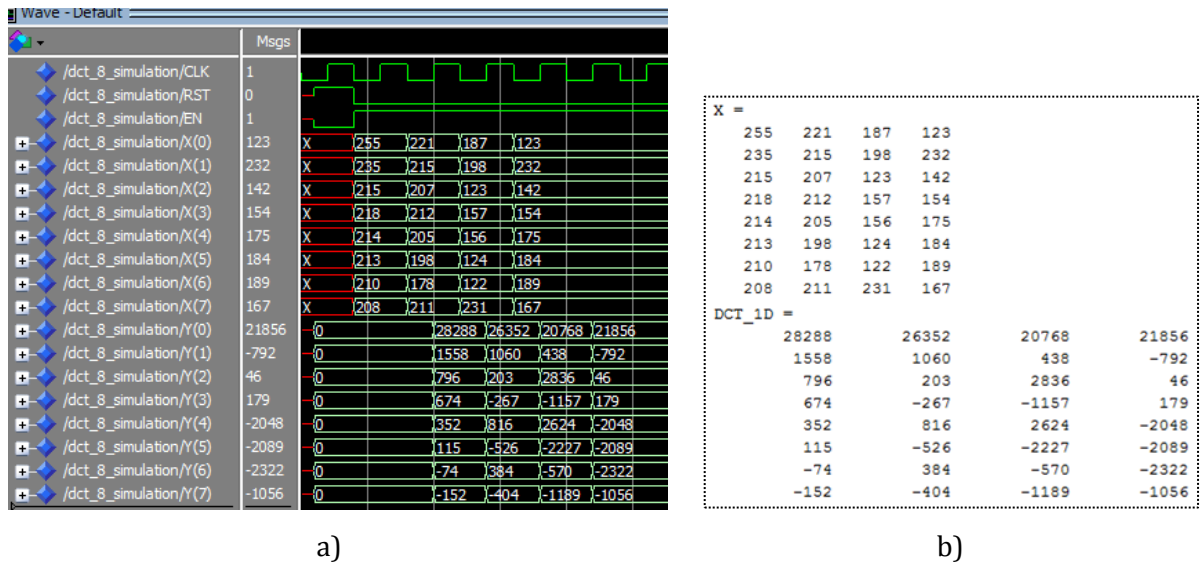
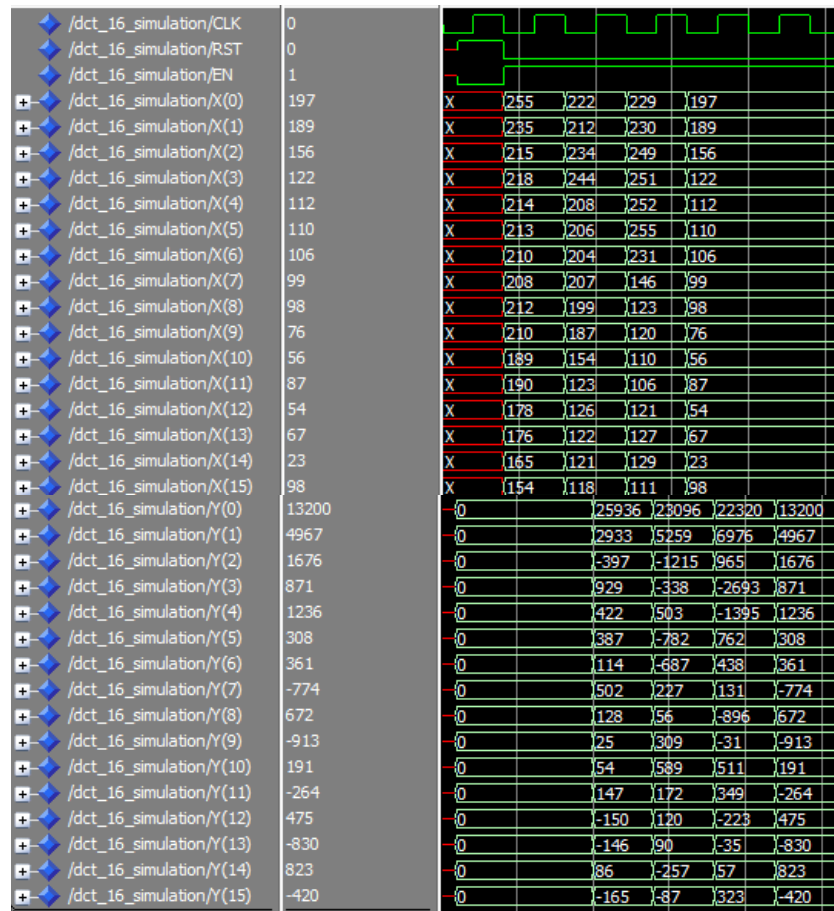


Figura 6.2 a) Resultados de la simulación funcional de la transformada DCT-1D-E/DCT-1D-M de 8-puntos.

b) Verificación de los resultados usando Matlab.



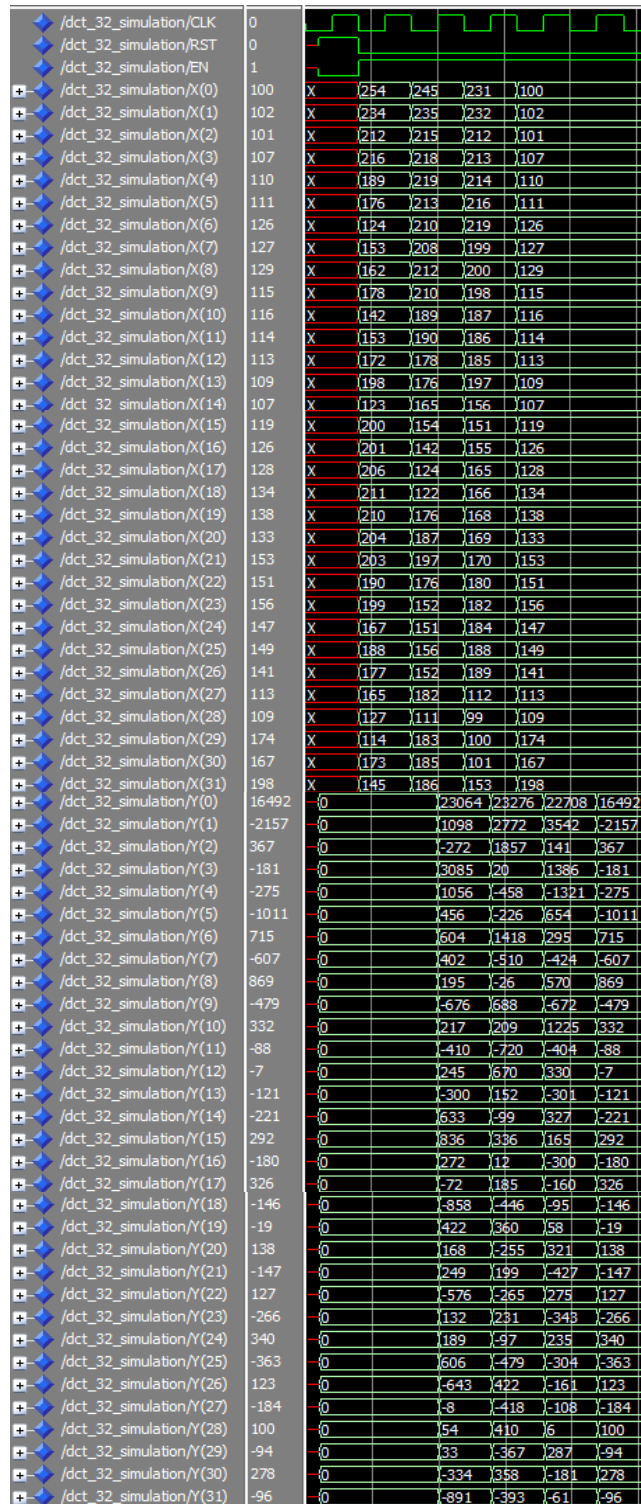
a)

X =				DCT_1D =			
255	222	229	197	25936	23096	22320	13200
235	212	230	189	2933	5259	6976	4967
215	234	249	156	-397	-1215	965	1676
218	244	251	122	929	-338	-2693	871
214	208	252	112	422	503	-1395	1236
213	206	255	110	387	-782	762	308
210	204	231	106	114	-687	438	361
208	207	146	99	502	227	131	-774
212	199	123	98	128	56	-896	672
210	187	120	76	25	309	-31	-913
189	154	110	56	54	589	511	191
190	123	106	87	147	172	349	-264
178	126	121	54	-150	120	-223	475
176	122	127	67	-146	90	-35	-830
165	121	129	23	86	-257	57	823
154	118	111	98	-165	-87	323	-420

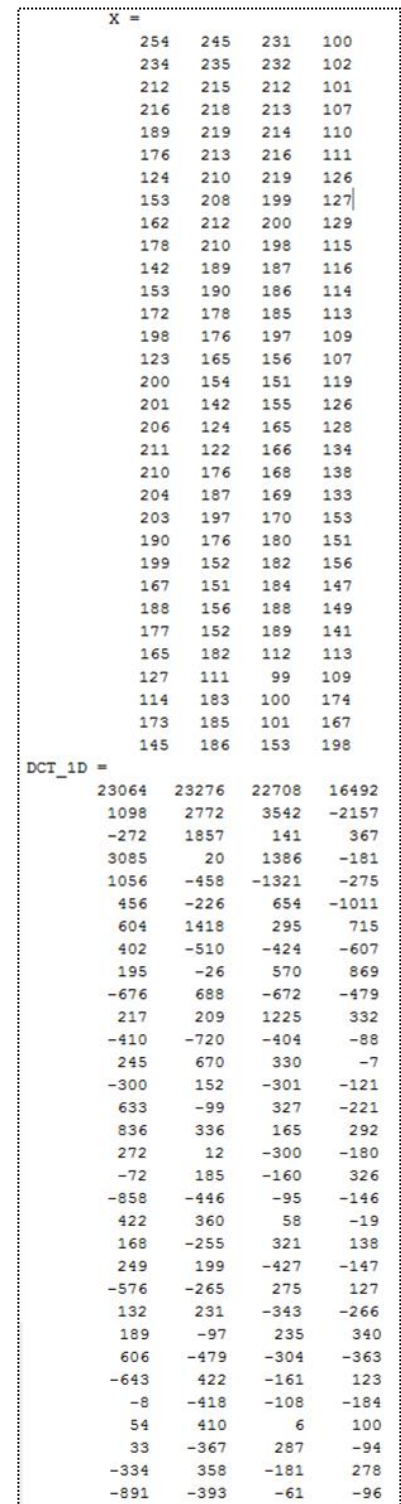
b)

Figura 6.3 a) Resultados de la simulación funcional de la transformada DCT-1D-E/DCT-1D-M de 16-puntos.

b) Verificación de los resultados usando Matlab.



a)



b)

Figura 6.4 a) Resultados de la simulación funcional de la transformada DCT-1D-E/DCT-1D-M de 32-puntos.

b) Verificación de los resultados usando Matlab.

6.2.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT-2D

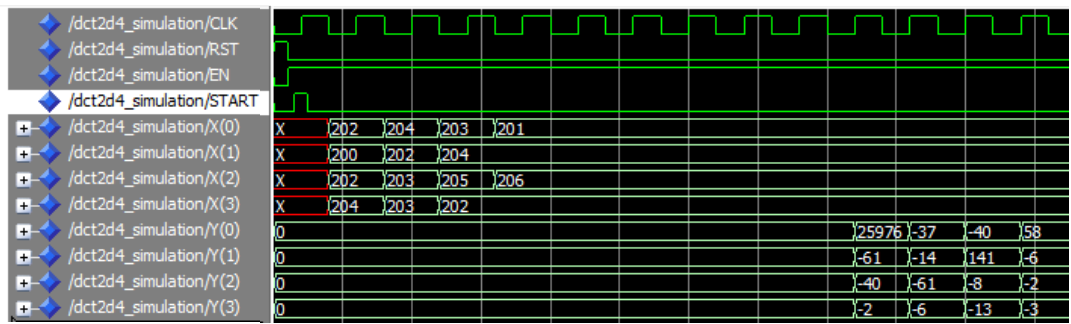
Las simulaciones de las transformadas DCT-2D se llevan a cabo usando matrices de $N \times N$ datos de 9 bits, donde N es el tamaño de la transformada. En este caso, estas matrices son obtenidas a partir de una imagen real. En la Figura 6.5 se muestra un ejemplo de una matriz de datos de entrada de tamaño 8×8 obtenida desde la imagen "Lenna.bmp". Del mismo modo que en la simulación de las transformadas DCT-1D, los datos de entrada de las simulaciones de las transformadas DCT-2D son números positivos en un rango entre 0 y 255.



Figura 6.5 Bloque de datos de 8x8 pixeles de la imagen "Lenna.bmp"

Los resultados de la simulación funcional son verificados en MATLAB, en este caso se calculan los resultados de la transformada DCT-2D teniendo en cuenta el Algoritmo 5, el cual fue presentado en la sección 4.7 del capítulo 4. En este algoritmo también se tienen en cuenta los factores de redondeo y escalamiento de la primera y de la segunda etapa de transformación DCT-1D.

Los resultados de las simulaciones funcionales y su verificación usando Matlab para las transformadas DCT-2D-E/DCT-2D-M de 4, 8, 16 y 32-puntos se observan en las Figura 6.6, 6.7, 6.8 y 6.9, respectivamente.



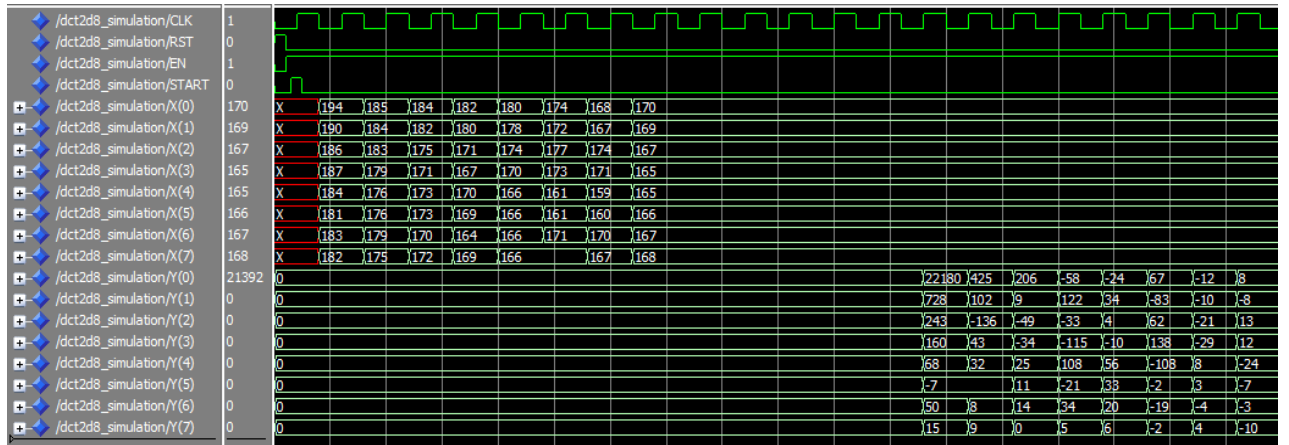
a)

dct4 =	X =				DCT2D =			
64	64	64	64	202	204	203	201	25976
83	36	-36	-83	200	202	204	204	-37
64	-64	-64	64	202	203	205	206	-40
36	-83	83	-36	204	203	202	202	58

b)

Figura 6.6 a) Resultados de la simulación funcional de la transformada DCT-2D-E/DCT-2D-M de 4-puntos.

b) Verificación de los resultados usando Matlab.



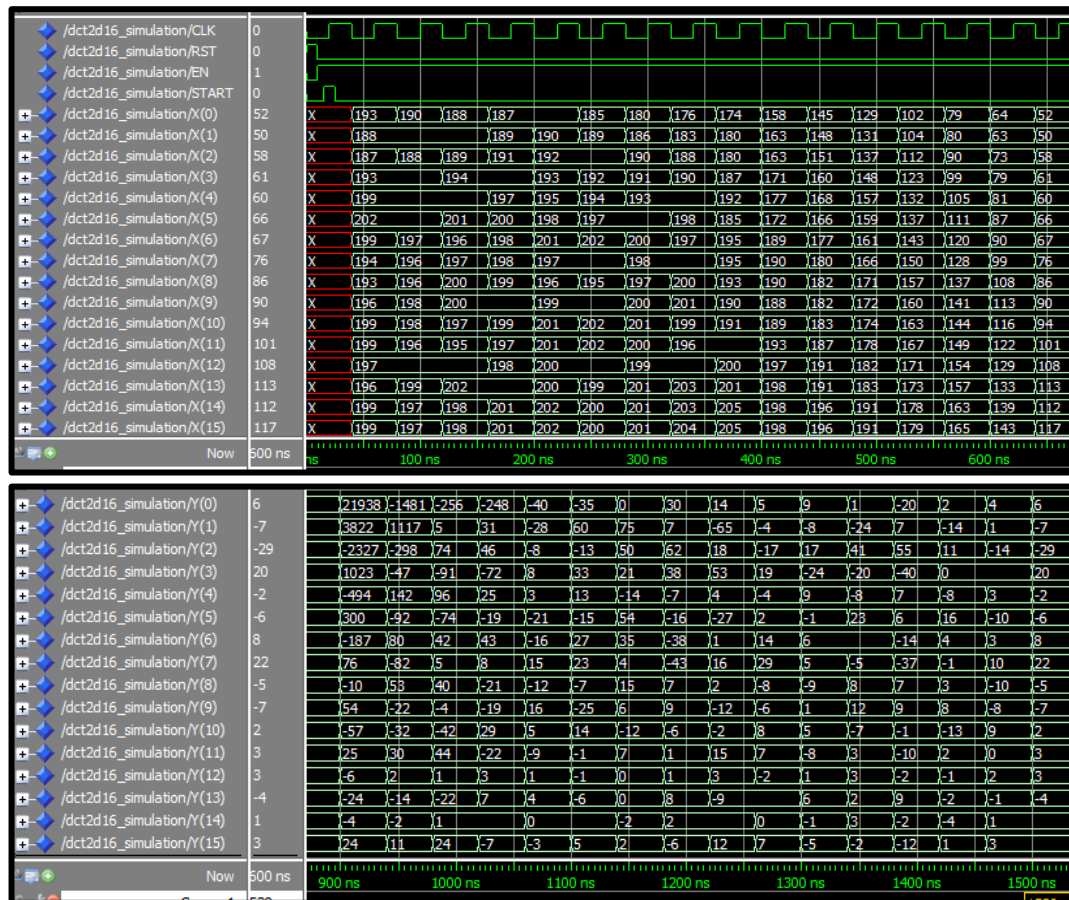
a)

X =	DCT2D =							
194	185	184	182	180	174	168	170	22180
190	184	182	180	178	172	167	169	425
186	183	175	171	174	177	174	167	206
187	179	171	167	170	173	171	165	-58
184	176	173	170	166	161	159	165	-24
181	176	173	169	166	161	160	166	67
183	179	170	164	166	171	170	167	-12
182	175	172	169	166	166	167	168	8

b)

Figura 6.7 a) Resultados de la simulación funcional de la transformada DCT-2D-E/DCT-2D-M de 8-puntos.

b) Verificación de los resultados usando Matlab.



a)

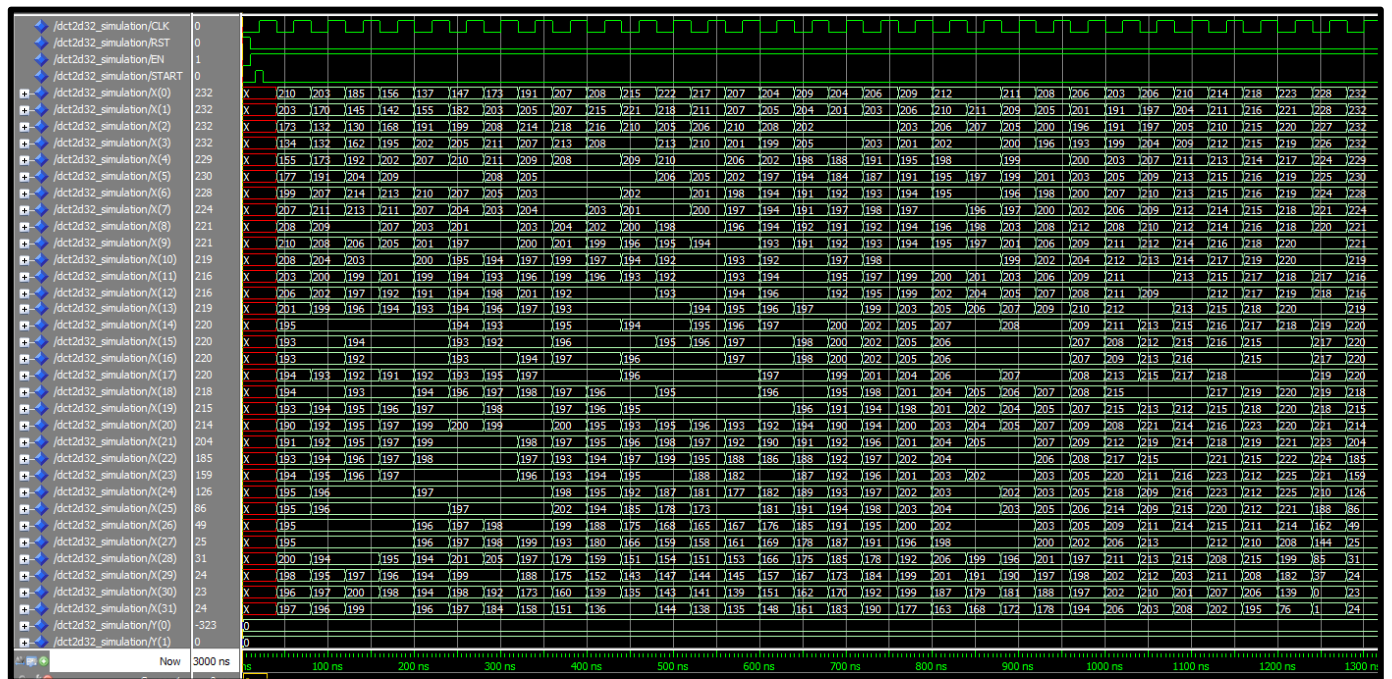
Xd =															
193	190	188	187	187	185	180	176	174	158	145	129	102	79	64	52
188	188	188	189	190	189	186	183	180	163	148	131	104	80	63	50
187	188	189	191	192	192	190	188	180	163	151	137	112	90	73	58
193	193	194	194	193	192	191	190	187	171	160	148	123	99	79	61
199	199	199	197	195	194	193	193	192	177	168	157	132	105	81	60
202	202	201	200	198	197	197	198	185	172	166	159	137	111	87	66
199	197	196	198	201	202	200	197	195	189	177	161	143	120	90	67
194	196	197	198	197	197	198	198	195	190	180	166	150	128	99	76
193	196	200	199	196	195	197	200	193	190	182	171	157	137	108	86
196	198	200	200	199	199	200	201	190	188	182	172	160	141	113	90
199	198	197	199	201	202	201	199	191	189	183	174	163	144	116	94
199	196	195	197	201	202	200	196	196	193	187	178	167	149	122	101
197	197	197	198	200	200	199	199	200	197	191	182	171	154	129	108
196	199	202	202	200	199	201	203	201	198	191	183	173	157	133	113
199	197	198	201	202	200	201	203	205	198	196	191	178	163	139	112
199	197	198	201	202	200	201	204	205	198	196	191	179	165	143	117

DCT2D =															
21938	-1481	-256	-248	-40	-35	0	30	14	5	9	1	-20	2	4	6
3822	1117	5	31	-28	60	75	7	-65	-4	-8	-24	7	-14	1	-7
-2327	-298	74	46	-8	-13	50	62	18	-17	17	41	55	11	-14	-29
1023	-47	-91	-72	8	33	21	38	53	19	-24	-20	-40	0	0	20
-494	142	96	25	3	13	-14	-7	4	-4	9	-8	7	-8	3	-2
300	-92	-74	-19	-21	-15	54	-16	-27	2	-1	23	6	16	-10	-6
-187	80	42	43	-16	27	35	-38	1	14	6	6	-14	4	3	8
76	-82	5	8	15	23	4	-43	16	29	5	-5	-37	-1	10	22
-10	53	40	-21	-12	-7	15	7	2	-8	-9	8	7	3	-10	-5
54	-22	-4	-19	16	-25	6	9	-12	-6	1	12	9	8	-8	-7
-57	-32	-42	29	5	14	-12	-6	-2	8	5	-7	-1	-13	9	2
25	30	44	-22	-9	-1	7	1	15	7	-8	3	-10	2	0	3
-6	2	1	3	1	-1	0	1	3	-2	1	3	-2	-1	2	3
-24	-14	-22	7	4	-6	0	8	-9	-9	6	2	9	-2	-1	-4
-4	-2	1	1	0	0	-2	2	0	0	-1	3	-2	-4	1	1
24	11	24	-7	-3	5	2	-6	12	7	-5	-2	-12	1	3	3

b)

Figura 6.8 a) Resultados de la simulación funcional de la transformada DCT-2D-E/DCT-2D-M de 16-puntos.

b) Verificación de los resultados usando Matlab.



a)

Figura 6.9 a) Resultados de la simulación funcional de la transformada DCT-2D-E/DCT-2D-M de 32-puntos.

TCT2D =																																
25256	907	-754	497	-392	149	-128	85	-11	91	-14	51	-13	39	2	31	-18	9	18	-3	-10	-5	-1	5	-5	6	-1	4	7	7	-5	1	
-559	-501	376	-448	-29	-200	-19	-24	35	-2	68	-16	32	-17	-2	-11	12	0	-21	-15	-3	-17	-17	-12	1	-5	15	6	9	6	14	3	
230	86	-349	-127	-310	-127	8	6	14	73	45	78	23	51	24	-15	1	-25	-2	6	3	5	-1	-5	0	5	5	8	7	3	5		
350	-1141	408	-648	180	-78	87	63	91	75	87	55	38	35	20	35	-3	21	0	3	4	3	5	3	-1	6	-6	15	1	11	1	8	
-492	522	-417	166	-33	64	119	65	69	54	36	20	15	13	14	25	21	13	-3	-13	-3	-1	9	5	3	-3	1	1	-1				
142	-364	172	-223	83	-22	87	67	139	54	122	61	93	37	72	12	30	-2	2	-10	10	-7	0	-3	1	-1	5	8	4	10	7	0	
-384	684	-399	365	-71	166	100	103	108	105	58	74	18	30	23	19	-6	-16	-33	-22	-13	-8	-7	-8	0	-5	8	-3	6	1	7	0	
393	-452	326	-119	103	154	30	149	74	47	38	38	7	3	-25	4	-24	14	-2	18	-17	-3	-20	-9	-2	2	3	-6	4	-6	5	1	
-243	306	-189	101	30	67	90	68	62	10	49	-11	4	-29	7	-23	-1	-35	-5	-20	4	-12	-2	-7	5	0	10	2	6	3	7	2	
224	-247	215	-46	68	141	-28	141	-70	72	-16	20	-56	-3	-40	7	-42	-44	-50	-35	-17	-11	-8	-9	-2	14	6	12	0	12	6	12	
-133	242	-109	80	88	-62	97	-48	27	16	-14	18	-27	-21	-15	-20	-20	-33	-16	-19	-8	-19	-16	-16	-6	9	-2	10	-10	10	-3	10	
174	-241	163	-40	11	77	13	32	4	9	-11	-1	-44	-28	-19	-15	-31	-12	-30	-9	-18	-8	-24	-9	-13	-5	-8	-7	-9	-5	0	0	
-147	196	-54	46	97	-61	79	-68	51	-53	-3	-50	-4	-54	-9	-52	-2	-45	7	-35	7	-25	5	-17	1	-9	-8	-3	-8	-1	-7	0	
105	-131	36	47	-80	125	-87	68	-72	4	-34	-7	-46	-9	-39	-24	-32	-18	-24	-13	-10	-2	-4	4	3	4	8	5	6	8	2		
-98	137	-36	-38	93	-91	74	-56	36	-57	-11	-58	-7	-32	-39	-27	-31	-11	-19	-11	-15	1	10	14	7	15	10	22	8	5	-1		
64	-50	0	62	-97	108	-79	59	-51	-8	-40	-35	-10	-36	-18	-31	-7	3	15	19	1	16	10	15	13	6	14	0	15	-5	5	-3	
-15	44	31	-56	103	-86	84	-51	13	-31	-18	-16	-19	-9	-15	-8	5	-1	21	-2	24	0	22	6	9	6	-3	9	-8	7	-8	0	
58	-15	0	85	-90	115	-70	52	-48	16	-32	3	-23	-5	-10	3	-5	-2	-9	4	-2	10	5	8	3	3	4	-1	6	-2	1	-2	
-10	19	39	-78	114	-97	84	-47	5	-21	-10	-5	-25	-6	-17	-4	-15	1	-4	3	6	3	-1	2	0	9	0	-9	-8	8	-2	3	
-7	35	-40	89	-99	106	-62	27	0	-18	-7	-20	-4	-39	14	-38	6	-16	4	-9	5	-3	0	1	7	-6	9	-6	5	-4	6	-6	
9	4	58	-69	103	-77	45	-8	-27	25	-35	7	-24	16	-29	18	-13	3	3	3	5	-1	7	-1	-3	7	-15	5	-12	5	-9	4	
31	6	-20	82	-74	67	-34	-7	15	-35	20	-15	12	-11	18	-11	13	4	2	7	5	6	1	4	0	-8	2	-12	0	-9	5	-6	
13	-12	71	-72	96	-66	41	-8	-5	11	-15	-2	-9	13	-25	21	-11	9	-2	2	-1	3	7	0	0	-2	2	-9	4	-6	1	-9	2
8	5	-26	48	-53	38	-19	3	8	-13	3	-14	13	-19	13	-20	12	-6	18	-1	8	-4	4	-4	4	-3	2	-6	2	-6	2	-1	
2	-12	27	-41	42	-21	8	5	-5	10	-15	21	-23	22	-16	11	-8	7	-4	0	4	-1	2	2	-1	2	-5	7	-7	7	-2	1	
-20	30	-35	41	-27	14	2	-7	3	-6	23	-28	32	-22	15	-3	6	-10	-7	0	-11	7	-6	0	1	0	4	-12	11	-6	5	3	
14	-28	14	-25	5	-4	-8	5	-13	12	-19	36	-36	22	-12	9	-11	11	-4	2	4	-2	-5	-2	-3	3	2	7	-9	6	1	1	
-21	21	-23	12	-10	5	2	-4	9	-7	13	-25	22	-32	19	-21	14	-18	10	-8	0	-4	3	-4	0	0	4	-1	3	2	1	2	
-11	-17	13	-10	6	-1	-1	6	-11	16	-18	22	-19	20	-20	-16	-16	13	-10	10	-3	3	-3	0	0	-1	1	-2	-1	-2	1	-2	
-1	13	-6	13	-5	1	1	-12	10	-21	21	-13	18	-9	18	-13	16	-7	10	-5	10	-3	4	1	-1	-1	-2	0	-4	0	0	-1	
8	-1	12	-2	11	-8	3	7	-6	14	-11	2	-2	14	-17	18	-13	14	-9	7	-12	8	0	2	1	4	0	-5	5	-2	0	-1	
4	-7	1	-3	-4	4	-2	-3	9	-11	3	-6	3	-8	7	-13	9	-5	17	-6	10	-4	2	-1	1	-1	-2	2	-3	4	-2	0	

b)

Figura 6.9 b) Verificación de los resultados de simulación de la transformada DCT-2D-E/DCT-2D-M de 32 puntos usando Matlab.

En cuanto a los resultados de simulación de las transformadas DCT-2D-P de 4, 8, 16 y 32-puntos, estos son los mismos que se presentaron en las figuras anteriores, con la diferencia que en estos casos los resultados son obtenidos después de 16, 22, 32 y 50 ciclos de reloj, respectivamente.

Los resultados de la simulación funcional de la transformada DCT-2D-P de 4puntos incluyendo las señales de control que intervienen en el proceso de transformación se presenta en la Figura 6.10. En esta figura se pueden observar los resultados de la primera etapa de transformación, transposición y segunda etapa de transformación, los cuales están resaltados con un recuadro azul, naranja y rojo, respectivamente.

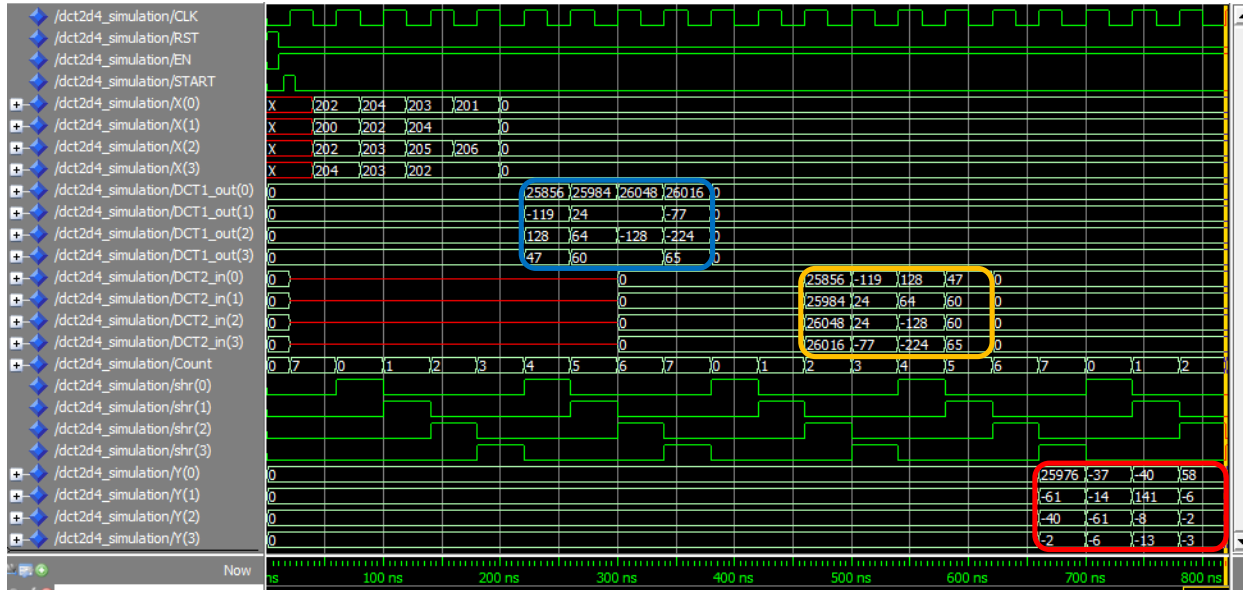


Figura 6.10 Resultados de la simulación funcional de la transformada DCT-2D-P de 4-puntos.

6.3 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT INVERSA

En esta sección se presenta la simulación funcional de los diseños propuestos para las transformadas IDCT-1D y IDCT-2D. Es importante tener en cuenta que los resultados de simulación de las transformadas IDCT-1D-E y IDCT-1D-M son exactamente iguales. Por su parte la transformada IDCT-1D-P presenta los mismos resultados numéricos que las otras dos, pero en este caso el cálculo de los resultados requiere de más ciclos de reloj, ya que esta transformada posee varias etapas de pipeline. Lo anterior también se aplica para el caso de las transformadas IDCT-2D.

6.3.1 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA IDCT-1D

Las simulaciones de las transformadas IDCT-1D se llevan a cabo usando como datos de entrada a los 4 vectores de N datos obtenidos en las simulaciones de las transformada DCT-1D. Lo anterior con el fin de obtener en la salida de la transformada IDCT-1D los datos originales que ingresaron como entradas en la transformada DCT-1D. Cabe resaltar que en este caso los datos de entrada son de 16 bits.

Los resultados de simulación son verificados en MATLAB, en este caso se calculan los resultados de la transformada IDCT-1D teniendo en cuenta el Algoritmo 2, el cual fue presentado en el capítulo 4, sección 4.3. También, es importante tener en cuenta el proceso de redondeo y escalamiento que se debe realizar sobre los datos de salida de esta transformada. En este caso, este proceso se lleva a cabo utilizando los factores de redondeo y escalamiento para la segunda etapa de transformación IDCT-1D, los cuales fueron descritos la sección 4.7 del capítulo 4.

Los resultados de las simulaciones funcionales y su verificación usando Matlab para las transformadas IDCT-1D-E/IDCT-1D-M de 4, 8, 16 y 32-puntos se observan en las Figura 6.11, 6.12, 6.13 y 6.14, respectivamente.

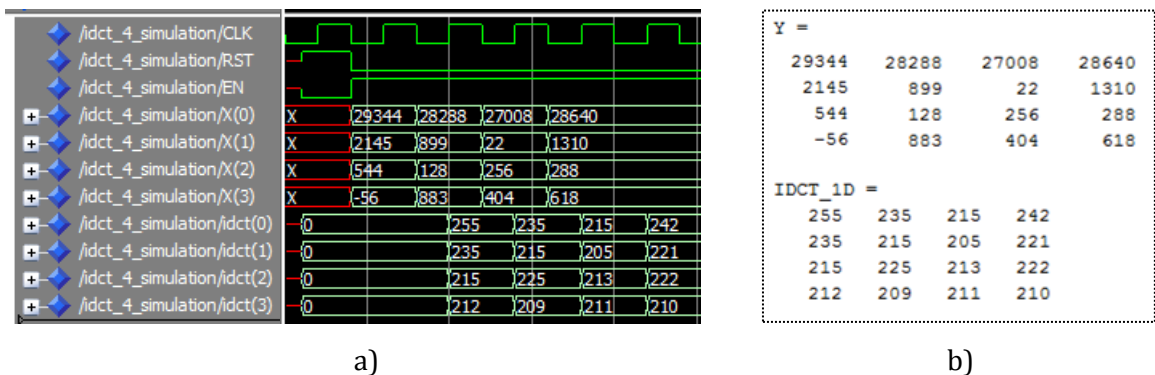
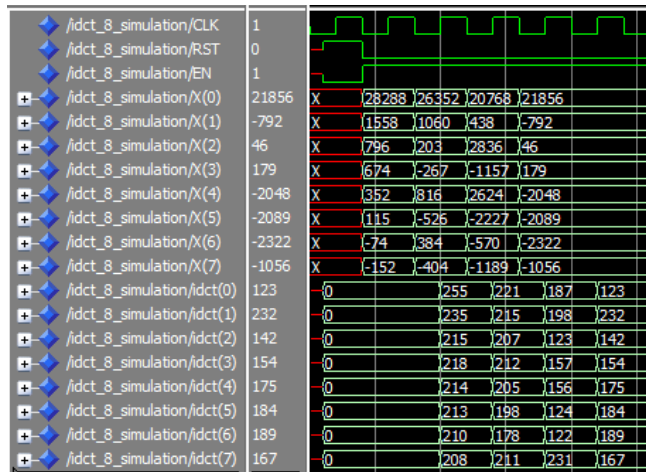
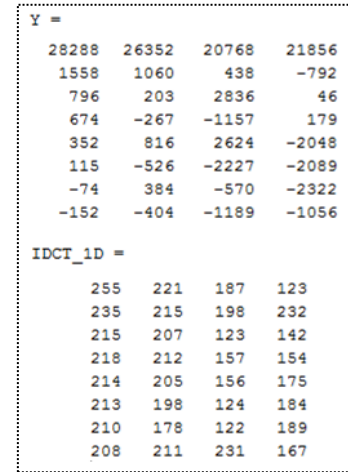


Figura 6.11 a) Resultados de la simulación funcional de la transformada IDCT-1D-E/IDCT-1D-M de 4-puntos. b) Verificación de los resultados usando Matlab.

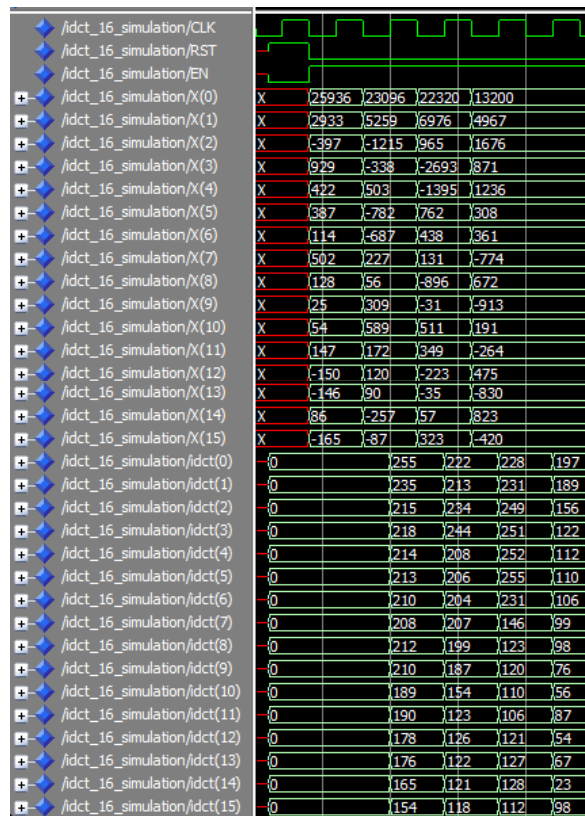


a)

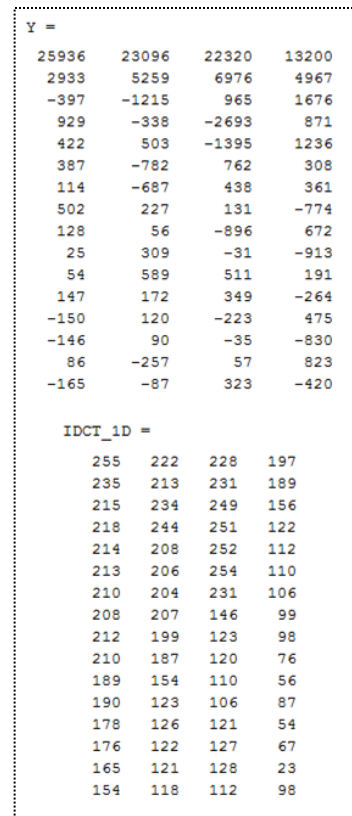


b)

Figura 6.12 a) Resultados de la simulación funcional de la transformada IDCT-1D-E/IDCT-1D-M de 8-puntos. b) Verificación de los resultados usando Matlab.

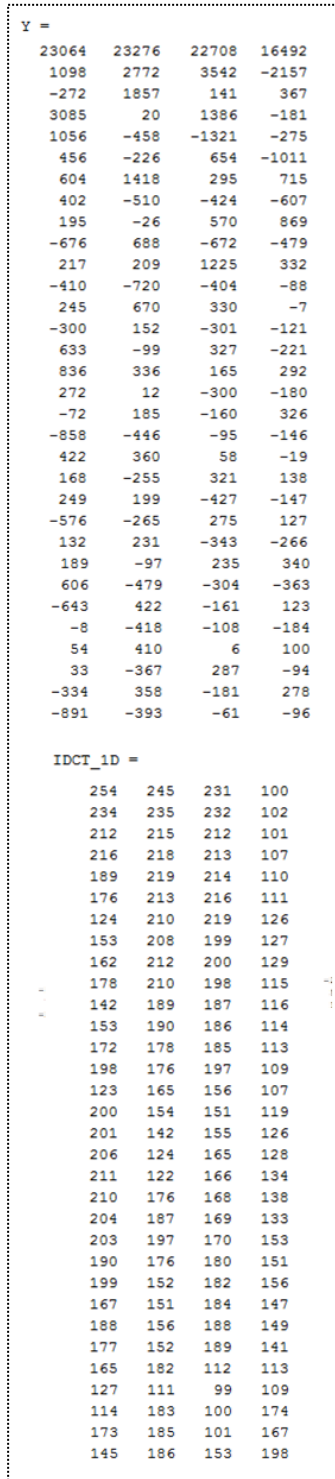
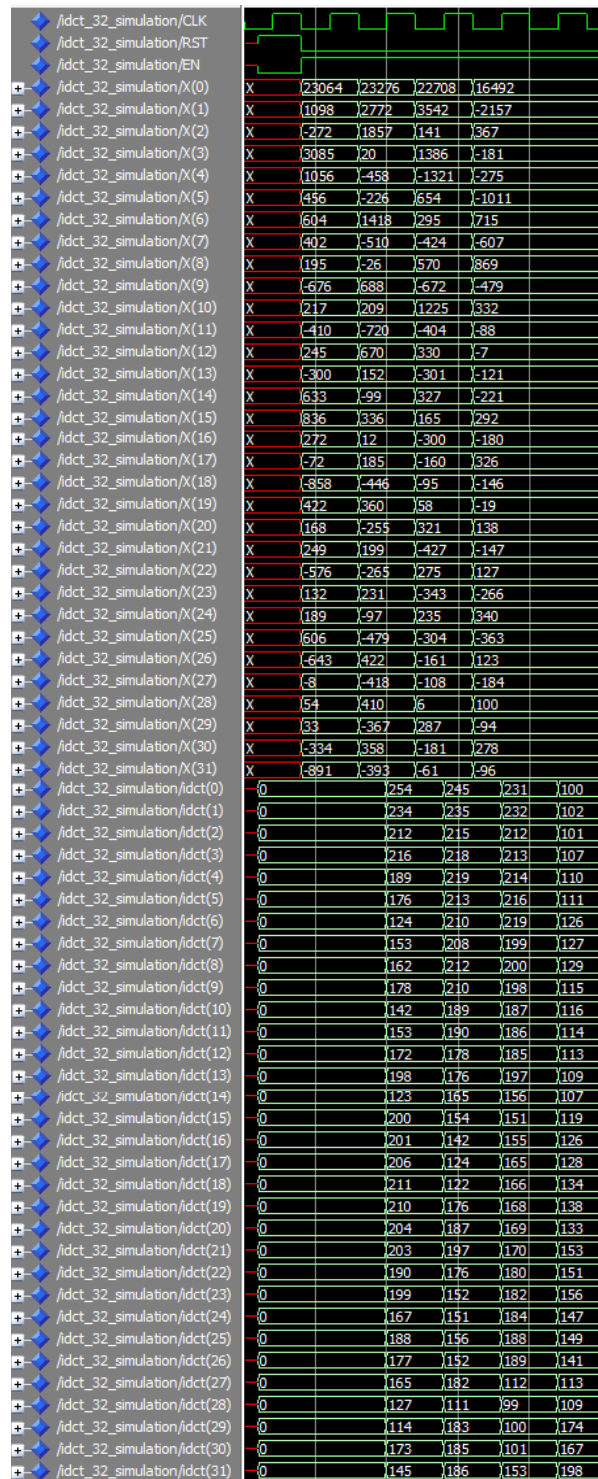


a)



b)

Figura 6.13 a) Resultados de la simulación funcional de la transformada IDCT-1D-E/IDCT-1D-M de 16-puntos. b) Verificación de los resultados usando Matlab.



a)

b)

Figura 6.14 a) Resultados de la simulación funcional de la transformada IDCT-1D-E/IDCT-1D-M de 32-puntos.

b) Verificación de los resultados usando Matlab.

En cuanto a los resultados de simulación de las transformadas IDCT-1D-P de 4, 8, 16 y 32-puntos, estos son los mismos que se presentaron en las figuras anteriores, con la diferencia de que en estos casos los resultados son obtenidos después de 5, 6, 7 y 8 ciclos de reloj, respectivamente.

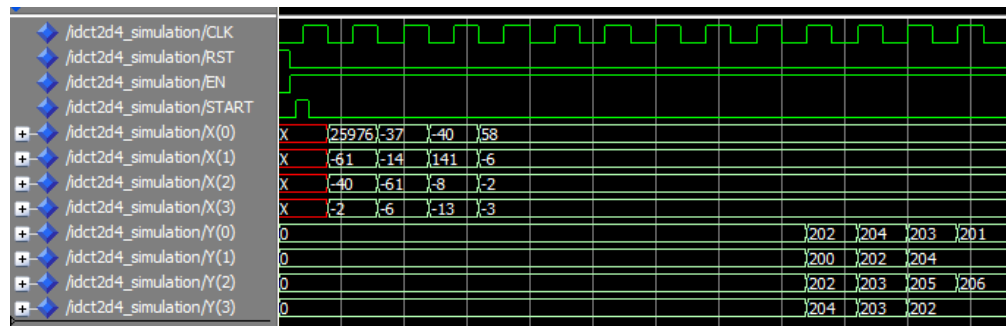
6.3.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA IDCT-2D

La simulación de las transformadas IDCT-2D se lleva a cabo usando como datos de entrada a las matrices de $N \times N$ datos resultantes, obtenidas en las simulaciones de las transformadas DCT-2D. Lo anterior con el fin de obtener en la salida de la transformada IDCT-2D los datos originales que ingresaron como entradas en la transformada DCT-2D.

Los resultados de simulación son verificados en MATLAB, en este caso se calculan los resultados de la transformada IDCT-2D teniendo en cuenta el Algoritmo 6, el cual fue presentado en la sección 4.7 del capítulo 4. En este algoritmo también se tienen en cuenta los factores de redondeo y escalamiento de la primera y de la segunda etapa de transformación IDCT-1D.

Los resultados de las simulaciones funcionales y su verificación usando Matlab para las transformadas IDCT-2D-E/DCT-2D-M de 4, 8, 16 y 32-puntos se observan en las Figura 6.15, 6.16, 6.17 y 6.18, respectivamente.

En cuanto a los resultados de simulación de las transformadas IDCT-2D-P de 4, 8, 16 y 32-puntos, estos son los mismos que se presentan en las Figura 6.15, 6.16, 6.17 y 6.18, con la diferencia que en este caso los resultados son obtenidos después de 16, 22, 32 y 50 ciclos de reloj, respectivamente.



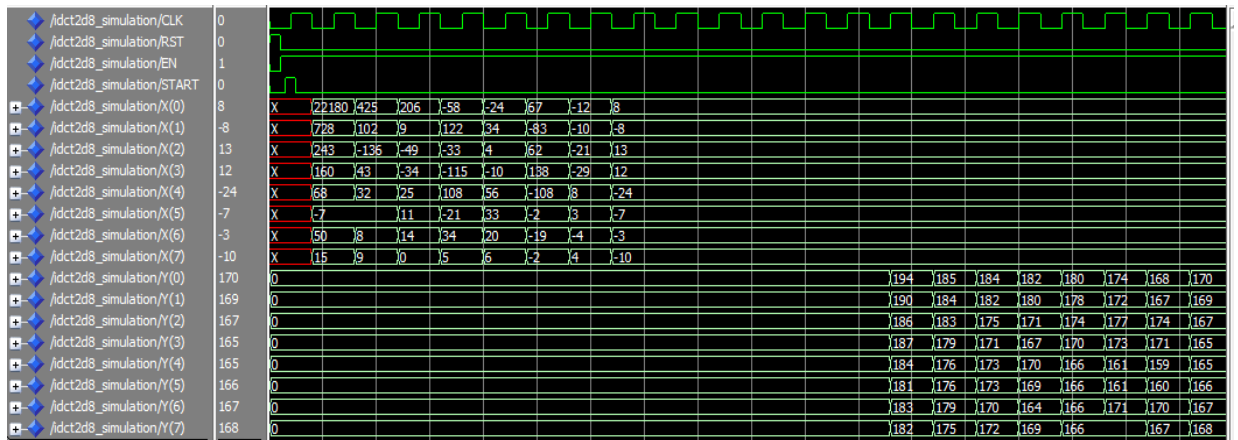
a)

Y =	25976	-37	-40	58	IDCT2D =	202	204	203	201
	-61	-14	141	-6		200	202	204	204
	-40	-61	-8	-2		202	203	205	206
	-2	-6	-13	-3		204	203	202	202

b)

Figura 6.15 a) Resultados de la simulación funcional de la transformada IDCT-2D-E/IDCT-2D-M de 4-puntos.

b) Verificación de los resultados usando Matlab.



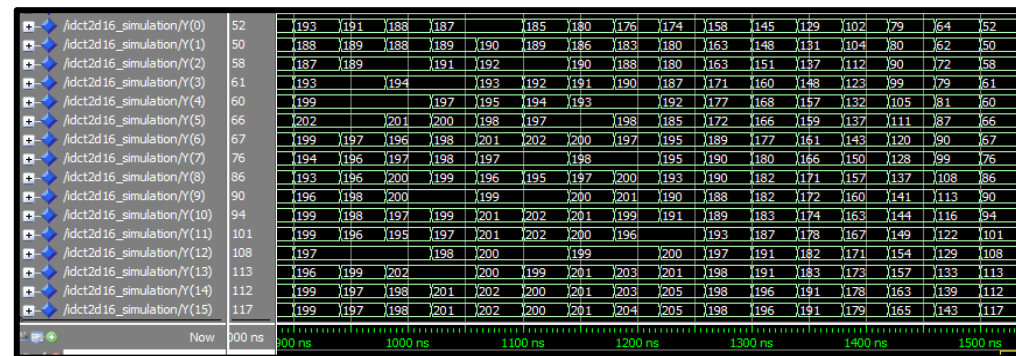
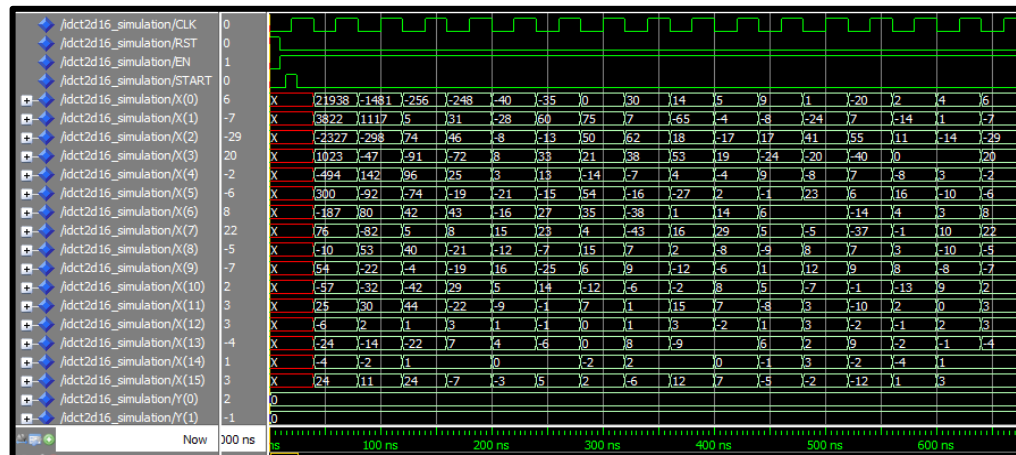
a)

Y =	22180	425	206	-58	-24	67	-12	8	IDCT2D =	194	185	184	182	180	174	168	170
	728	102	9	122	34	-83	-10	-8		190	184	182	180	178	172	167	169
	243	-136	-49	-33	4	62	-21	13		186	183	175	171	174	177	174	167
	160	43	-34	-115	-10	138	-29	12		187	179	171	167	170	173	171	165
	68	32	25	108	56	-108	8	-24		184	176	173	170	166	161	159	165
	-7	-7	11	-21	33	-2	3	-7		181	176	173	169	166	161	160	166
	50	8	14	34	20	-19	-4	-3		183	179	170	164	166	171	170	167
	15	9	0	5	6	-2	4	-10		182	175	172	169	166	166	167	168

b)

Figura 6.16 a) Resultados de la simulación funcional de la transformada IDCT-2D-E/IDCT-2D-M de 8-puntos.

b) Verificación de los resultados usando Matlab.



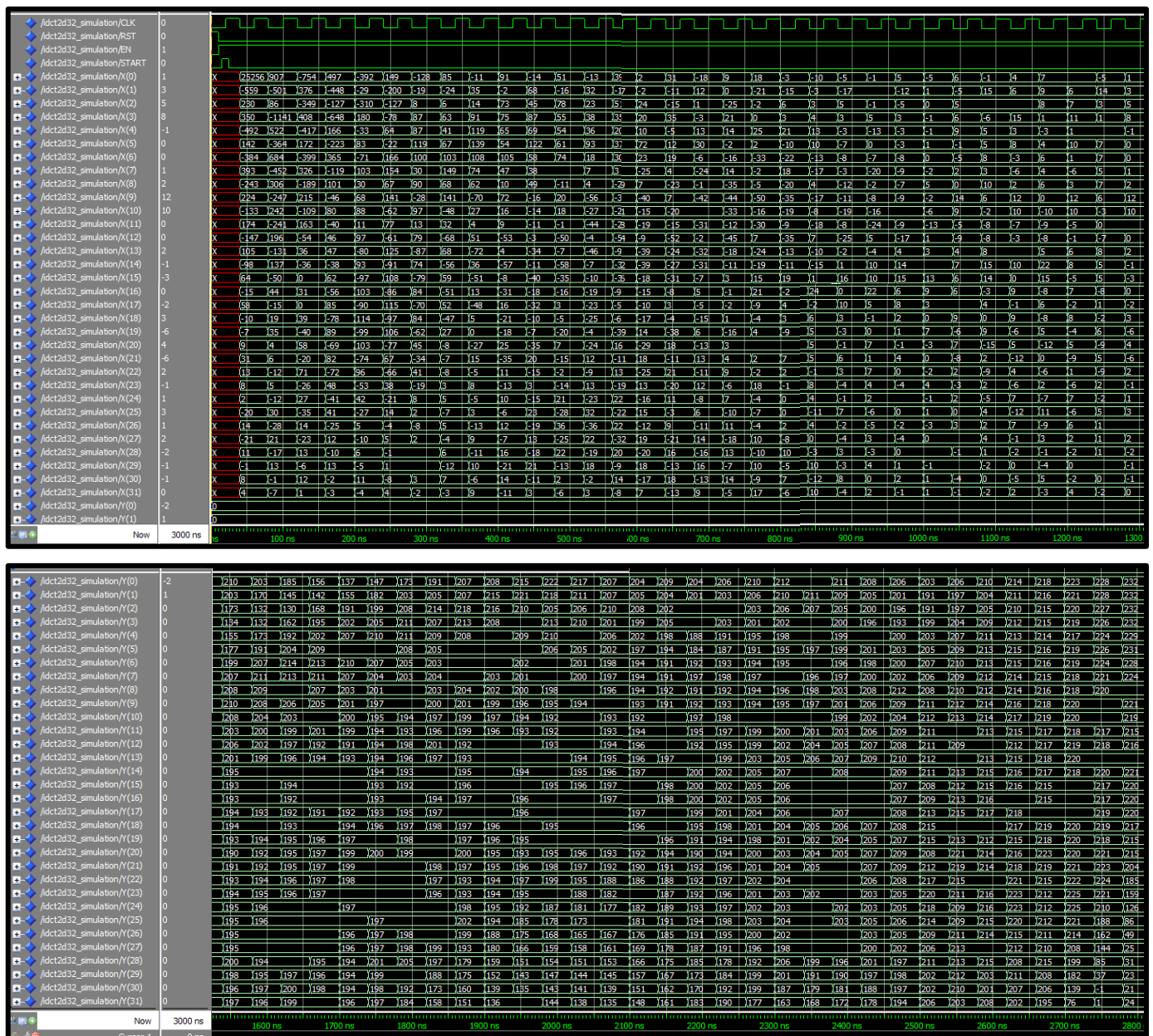
a)

Y =	21938	-1481	-256	-248	-40	-35	0	30	14	5	9	1	-20	2	4	6
	3822	1117	5	31	-28	60	75	7	-65	-4	-8	-24	7	-14	1	-7
	-2327	-298	74	46	-8	-13	50	62	18	-17	17	41	55	11	-14	-29
	1023	-47	-91	-72	8	33	21	38	53	19	-24	-20	-40	0	0	20
	-494	142	96	25	3	13	-14	-7	4	-4	9	-8	7	-8	3	-2
	300	-92	-74	-19	-21	-15	54	-16	-27	2	-1	23	6	16	-10	-6
	-187	80	42	43	-16	27	35	-38	1	14	6	6	-14	4	3	8
	76	-82	5	8	15	23	4	-43	16	29	5	-5	-37	-1	10	22
	-10	53	40	-21	-12	-7	15	7	2	-8	-9	8	7	3	-10	-5
	54	-22	-4	-19	16	-25	6	9	-12	-6	1	12	9	8	-8	-7
	-57	-32	-42	29	5	14	-12	-6	-2	8	5	-7	-1	-13	9	2
	25	30	44	-22	-9	-1	7	1	15	7	-8	3	-10	2	0	3
	-6	2	1	3	1	-1	0	1	3	-2	1	3	-2	-1	2	3
	-24	-14	-22	7	4	-6	0	8	-9	6	2	9	-2	-4	1	1
	-4	-2	1	1	0	0	-2	2	2	0	-1	3	-2	-4	1	1
	24	11	24	-7	-3	5	2	-6	12	7	-5	-2	-12	1	3	3
IDCT2D =	193	191	188	187	187	185	180	176	174	158	145	129	102	79	64	52
	188	189	188	189	190	189	186	183	180	163	148	131	104	80	62	50
	187	189	191	192	190	188	180	163	151	137	112	90	72	58		
	193	194	193	192	191	190	187	171	160	148	123	99	79	61		
	199	199	197	195	194	193	192	177	168	157	132	105	81	60		
	202	202	201	200	198	197	198	185	172	166	159	137	111	87	66	
	199	197	196	198	201	202	200	197	195	189	177	161	143	120	90	67
	194	196	197	198	197	198	198	195	190	180	166	150	128	99	76	
	193	196	200	199	196	195	197	200	193	190	182	171	157	137	108	86
	196	198	200	199	200	201	190	188	182	172	160	141	113	90		
	199	198	197	199	201	202	201	199	191	189	183	174	163	144	116	94
	199	196	195	197	201	202	200	196	193	187	178	167	149	122	101	
	197	198	198	200	199	199	200	197	191	182	171	154	129	108		
	196	199	202	200	199	201	203	201	198	191	183	173	157	133	113	
	199	197	198	201	202	200	201	203	198	196	191	178	163	139	112	
	199	197	198	201	202	200	201	204	205	198	196	191	179	165	143	117

b)

Figura 6.17 a) Resultados de la simulación funcional de la transformada IDCT-2D-E/IDCT-2D-M de 16-puntos.

b) Verificación de los resultados usando Matlab.



a)

Figura 6.18 a) Resultados de la simulación funcional de la transformada IDCT-2D-E/IDCT-2D-M de 32-puntos.

Y =

25256	907	-754	497	-392	149	-128	85	-11	91	-14	51	-13	39	2	31	-18	9	18	-3	-10	-5	-1	5	-5	6	-1	4	7	7	-5	1
-559	-501	376	-448	-29	-200	-19	-24	35	-2	68	-16	32	-17	-2	-11	12	0	-21	-15	-3	-17	-17	-12	1	-5	15	6	9	6	14	3
230	86	-349	-127	-310	-127	8	6	14	73	45	78	23	51	24	-15	1	-25	-2	6	3	5	-1	-5	0	5	5	5	8	7	3	5
350	-1141	408	-648	180	-78	87	63	91	75	87	55	38	35	20	35	-3	21	0	3	4	3	5	3	-1	6	-6	15	1	11	1	8
-492	522	-417	166	-33	64	87	41	119	65	69	54	36	20	10	-5	13	14	25	21	13	-3	-13	-3	-1	9	5	3	-3	1	1	-1
142	-364	172	-223	83	-22	119	67	139	54	122	61	93	37	72	12	30	-2	2	-10	10	-7	0	-3	1	-1	5	8	4	10	7	0
-384	684	-399	365	-71	166	100	103	108	105	58	74	18	30	23	19	-6	-16	-33	-22	-13	-8	-7	-8	0	-5	8	-3	6	1	7	0
393	-452	326	-119	103	154	30	149	74	47	38	38	7	3	-25	4	-24	14	-2	18	-17	-3	-20	-9	-2	2	3	-6	4	-6	5	1
-243	306	-189	101	30	67	90	68	62	10	49	-11	4	-29	7	-23	-1	-35	-5	-20	4	-12	-2	-7	5	0	10	2	6	3	7	2
224	-247	215	-46	68	141	-28	141	-70	72	-16	20	-56	-3	-40	7	-42	-44	-50	-35	-17	-11	-8	-9	-2	14	6	12	0	12	6	12
-133	242	-109	80	88	-62	97	-48	27	16	-14	18	-27	-21	-15	-20	-20	-33	-16	-19	-8	-19	-16	-16	-6	9	-2	10	-10	10	-3	10
174	-241	163	-40	11	77	13	32	4	9	-11	-1	-44	-28	-19	-15	-31	-12	-30	-9	-18	-8	-24	-9	-13	-5	-8	-7	-9	-5	0	0
-147	196	-54	46	97	-61	79	-68	51	-53	-3	-50	-4	-54	-9	-52	-2	-45	7	-35	7	-25	5	-17	1	-9	-8	-3	-8	-1	-7	0
105	-131	36	47	-80	125	-87	68	-72	4	-34	-7	-46	-9	-39	-24	-32	-18	-24	-13	-10	-2	-4	4	3	4	8	8	5	6	8	2
-98	137	-36	-38	93	-91	74	-56	36	-57	-11	-58	-7	-32	-39	-27	-31	-11	-19	-11	-15	1	10	14	14	7	15	10	22	8	5	-1
64	-50	0	62	-97	108	-79	59	-51	-8	-40	-35	-10	-36	-18	-31	-7	3	15	19	1	16	10	15	13	6	14	0	15	-5	5	-3
-15	44	31	-56	103	-86	84	-51	13	-31	-18	-16	-19	-9	-15	-8	5	-1	21	-2	24	0	22	6	9	6	-3	9	-8	7	-8	0
58	-15	0	85	-90	115	-70	52	-48	16	-32	3	-23	-5	-10	3	-5	-2	-9	4	-2	10	5	8	3	3	4	-1	6	-2	1	-2
-10	19	39	-78	114	-97	84	-47	5	-21	-10	-5	-25	-6	-17	-4	-15	1	-4	3	6	3	-1	2	0	9	0	9	-8	8	-2	3
-7	35	-40	89	-99	106	-62	27	0	-18	-7	-20	-4	-39	14	-38	6	-16	4	-9	5	-3	0	1	7	-6	9	-6	5	-4	6	-6
9	4	58	-69	103	-77	45	-8	-27	25	-35	7	-24	16	-29	18	-13	3	3	3	5	-1	7	-1	-3	7	-15	5	-12	5	-9	4
31	6	-20	82	-74	-34	-7	15	-35	20	-15	12	-11	18	-11	13	4	2	7	5	6	1	4	0	-8	2	-12	0	-9	5	-6	
13	-8	7	-72	96	-66	41	-8	-5	11	-15	-2	-9	13	-25	21	-11	9	-2	2	-1	3	7	0	-2	2	-9	4	-6	1	-9	2
8	5	-26	48	-53	38	-19	3	8	-13	3	-14	13	-19	13	-20	12	-6	18	-1	8	-4	4	-4	-3	2	-6	2	-6	2	-1	
2	-12	27	-41	42	-21	8	5	-5	10	-15	21	-23	22	-16	11	-8	7	-4	0	4	-1	2	2	-1	2	-5	7	-7	7	-2	1
-20	30	-35	41	-27	14	2	-7	3	-6	23	-28	32	-22	15	-3	6	-10	-7	0	-11	7	-6	0	1	0	4	-12	11	-6	5	3
14	-28	14	-25	5	-4	-8	5	-13	12	-19	36	-36	22	-12	9	-11	11	-4	2	4	-2	-5	-2	-3	3	2	7	-9	6	1	1
-21	21	-23	12	-10	5	2	-4	9	-7	13	-25	22	-32	19	-21	14	-18	10	-8	0	-4	3	-4	0	0	4	-1	3	2	1	2
11	-17	13	-10	6	-1	-1	6	-11	16	-18	22	-19	20	-20	16	-16	13	-10	10	-3	3	-3	0	0	-1	1	-2	-1	-2	1	-2
-1	13	-6	13	-5	1	1	-12	10	-21	21	-13	18	-9	18	-13	16	-7	10	-5	10	-3	4	1	-1	-1	-2	0	-4	0	0	-1
8	-1	12	-2	11	-8	3	7	-6	14	-11	2	-2	14	-17	18	-13	14	-9	7	-12	8	0	2	1	-4	0	-5	5	-2	0	-1
4	-7	1	-3	-4	4	-2	-3	9	-11	3	-6	3	-8	7	-13	9	-5	17	-6	10	-4	2	-1	1	-1	-2	2	-3	4	-2	0

IDCT2D =

210	203	185	156	137	147	173	191	207	208	215	222	217	207	204	209	204	206	209	212	212	211	208	206	203	206	210	214	218	223	228	232	
203	170	145	142	155	182	203	205	207	215	221	218	211	207	205	204	201	203	206	210	211	209	205	201	191	197	204	211	216	221	228	232	
173	132	130	168	191	199	208	214	218	216	210	205	206	210	208	202	202	202	203	206	207	205	200	196	191	197	205	210	215	220	227	232	
134	132	162	195	202	205	211	207	213	208	208	213	210	201	199	205	205	203	201	202	202	200	196	193	199	204	209	212	215	219	226	232	
155	173	192	202	207	210	211	209	208	208	209	210	210	206	202	198	188	191	195	198	198	199	199	200	203	207	211	213	214	217	224	229	
177	191	204	209	209	209	208	205	205	205	205	206	205	202	197	194	184	187	191	195	197	199	201	203	205	209	213	215	216	219	225	230	
199	207	214	213	210	207	205	203	203	203	202	202	201	198	194	191	192	193	194	195	195	196	198	200	207	210	213	215	216	219	224	228	
207	211	213	211	207	204	203	204	204	203	201	201	200	197	194	191	197	198	197	196	197	196	197	200	202	206	209	212	214	215	218	221	224
208	209	207	207	203	201	201	203	204	202	200	198	198	196	194	192	191	192	194	196	198	203	208	212	208	210	212	214	216	218	220	221	
210	208	206	205	201	197	197	200	201	199	196	195	194	194	193	191	192	193	194	195	197	201	206	209	211	212	214	216	218	220	220	221	
208	204	203	203	200	195	194	197	199	197	194	192	192	193	192	192	198	198	198	198	199	202	204	212	213	214	217	219	220	220	219		
203	200	199	201	199	194	193	196	199	196	193	192	192	193	194	194	195	197	199	200	201	203	206	209	211	211	213	215	217	218	217	216	
206	202	197	192	191	194	198	201	192	192	192	193	193	194	196	196	192	195	199	202	204	205	207	208	211	209	209	212	217	219	218	216	
201	199	196	194	193	194	196	197	193	193	193	193	194	195	196	197	197	199	203	205	206	207	209	210	212	212	213	215	218	220	220	219	
195	195	195	195	195	194	193	193	195	195	194	194	195	196	197	197	200	202	205	207	207	208	208	209	211	213	215	216	217	218	219	220	
193	193	194	194	194	193	192	192	196	196	196	196	195	196	197	197	198	200	202	205	206	206	206	206	207	208	212	215	216	215	215	217	220
193	193	192	192	192	193	193	194	197	197	196	196	196	197	197	198	200	202	205	206	206	206	206	207	209	213	216	216	215	215	217	220	
194	193	192	191	192	193	195	197	197	197	196	196	196	196	197	197	199	201	204	206	206	207	207	208	213	215	217	218	218	218	219	220	
194	194	193	193	193	194	196	197	198	197	196	196	195	195	195	196	195	195	198	201	204	205	206	207	208	215	215	215	217	219	220	219	
193	194	195	196	197	197	198	198	197	196	195	195	195	195	195	196	191	194	198	201	202	204	205	207	215	213	212	215	218	220	218	215	
190	192	195	197	199	200	199	199	200	195	193	195	196	193	192	194	190	194	200	203													

6.4 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT DE MULTIPLES TAMAÑOS

En esta sección se presenta la simulación funcional de las transformadas DCT-1D-T y DCT-2D-T, las cuales pueden calcular cualquiera de los tamaños de la transformada DCT-1D y DCT-2D, respectivamente.

6.4.1 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT-1D-T

La simulación de la transformada DCT-1D-T se lleva a cabo usando 4 vectores de entrada, uno para cada tamaño de la transformada. En este caso estos vectores hacen referencia a los mismos vectores de entrada que se usaron en las simulaciones de las transformadas DCT-1D. Lo anterior con el fin de obtener los mismos resultados, pero en este caso a partir de solo una arquitectura para la transformada DCT-1D, la cual incluye a todos los tamaños.

El resultado de la simulación funcional de la transformada DCT-1D-T para los tamaños 4 y 8-puntos se observa en la Figura 6.19. Del mismo modo, la simulación funcional de la transformada DCT-1D-T para los tamaños 16 y 32-puntos se observa en la Figura 6.20.

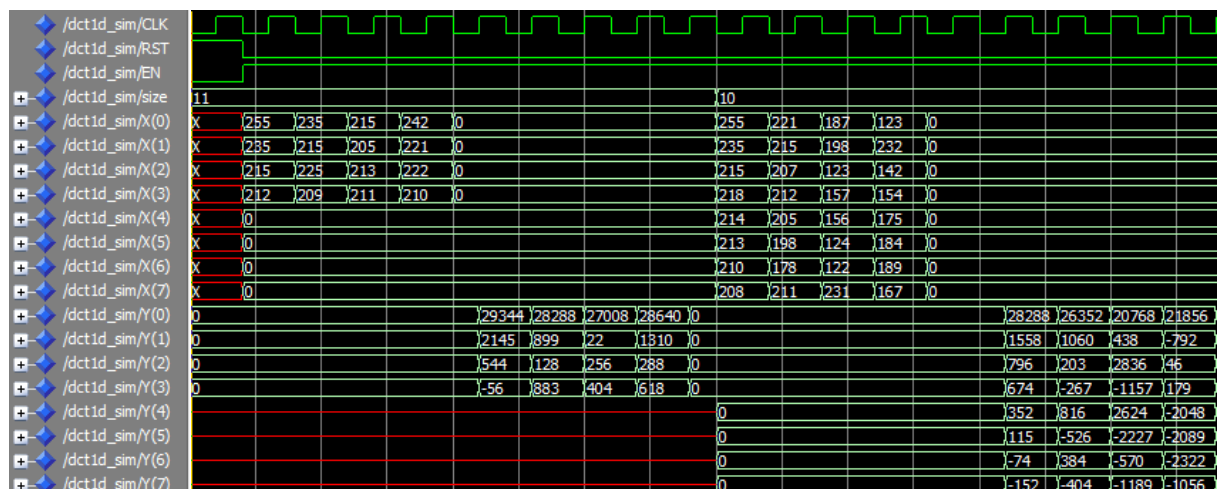


Figura 6.19 Resultados de la simulación funcional de la transformada DCT-1D-T para los tamaños 4 y 8-puntos.

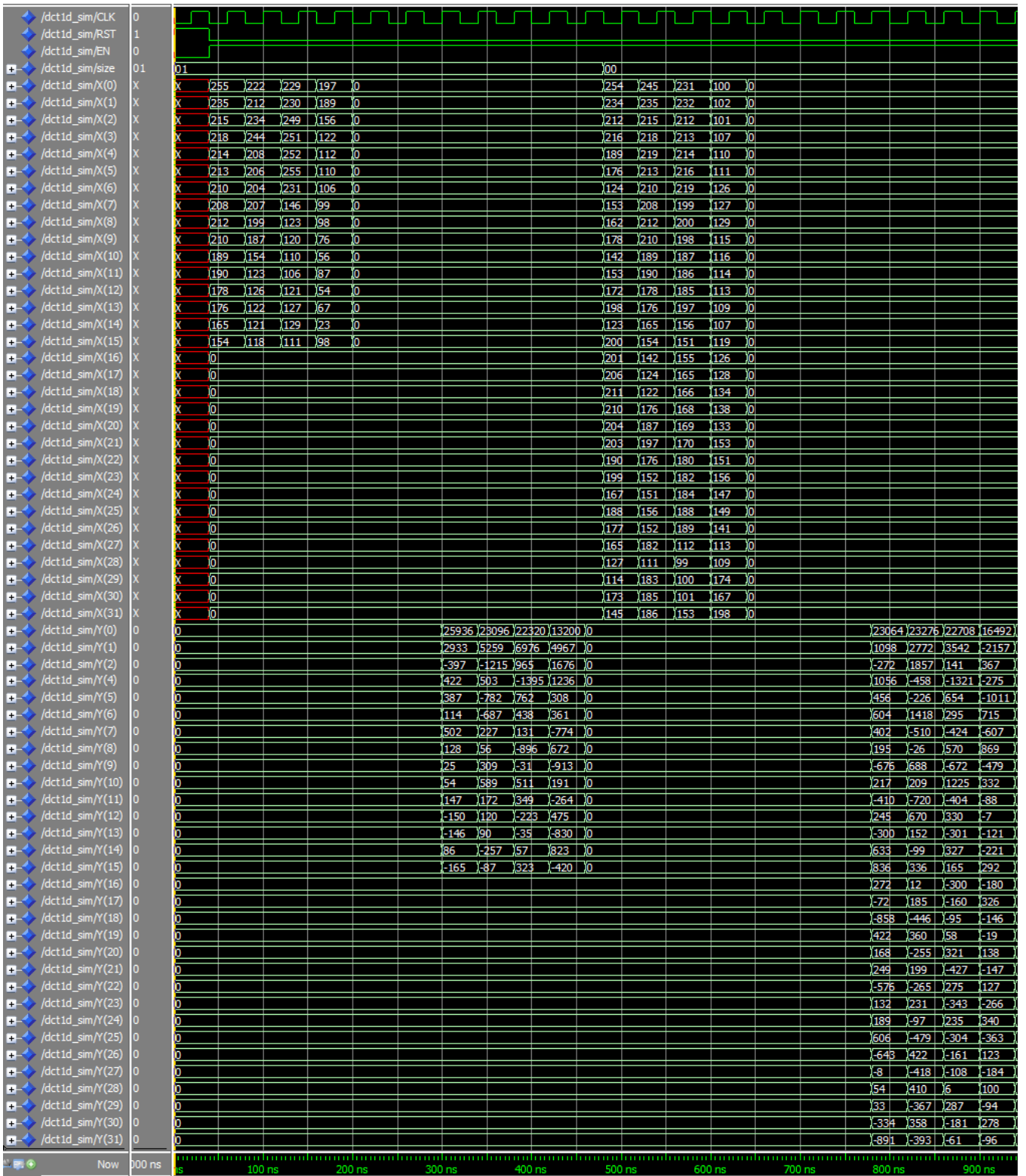


Figura 6.20 Resultados de la simulación funcional de la transformada DCT-1D-T para los tamaños 16 y 32-puntos.

En cuanto a la verificación de los resultados, ésta no se presenta en las anteriores figuras ya que los resultados obtenidos son los mismos que se obtuvieron en las simulaciones de las transformadas DCT-1D, por lo tanto, estos resultados ya fueron verificados anteriormente y no es necesario realizar una nueva verificación.

6.4.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT-2D-T

La simulación de la transformada DCT-2D-T se lleva a cabo usando 4 matrices de entrada, una para cada tamaño de la transformada. En este caso estas matrices hacen referencia a las mismas matrices de entrada que se usaron en las simulaciones de las transformadas DCT-2D. Lo anterior con el fin de obtener los mismos resultados, pero en este caso a partir de solo una arquitectura para la transformada DCT-2D, la cual incluye a todos los tamaños.

El resultado de la simulación funcional de la transformada DCT-2D-T para los tamaños 4 y 8-puntos se observa en la Figura 6.21. En esta figura se incluyen los resultados de cada uno de los bloques que conforman esta arquitectura. El recuadro azul resalta los resultados del primer bloque DCT-1D-T. Por su parte, el recuadro amarillo resalta los resultados del bloque de trasposición y finalmente, el recuadro rojo resalta los resultados de la segunda DCT-1D-T.

En la Figura 6.21, también se presentan algunas de las señales de control que intervienen en el cálculo de la transformada DCT-2D-T tales como: las señales de habilitación de los bloques DCT-1D-T, las señales de habilitación de escritura de las memorias del bloque de trasposición, la señal “size” que permite seleccionar el tamaño de la transformada, y finalmente, el valor del contador.

Al igual que en el caso anterior, en este caso no se presenta la verificación de los resultados, debido a que estos ya fueron verificados anteriormente.

En cuanto a los resultados de simulación de la transformada DCT-1D-T para los tamaños 16 y 32, estos hacen referencia a los resultados que fueron presentados en las Figura 6.8 y 6.9.

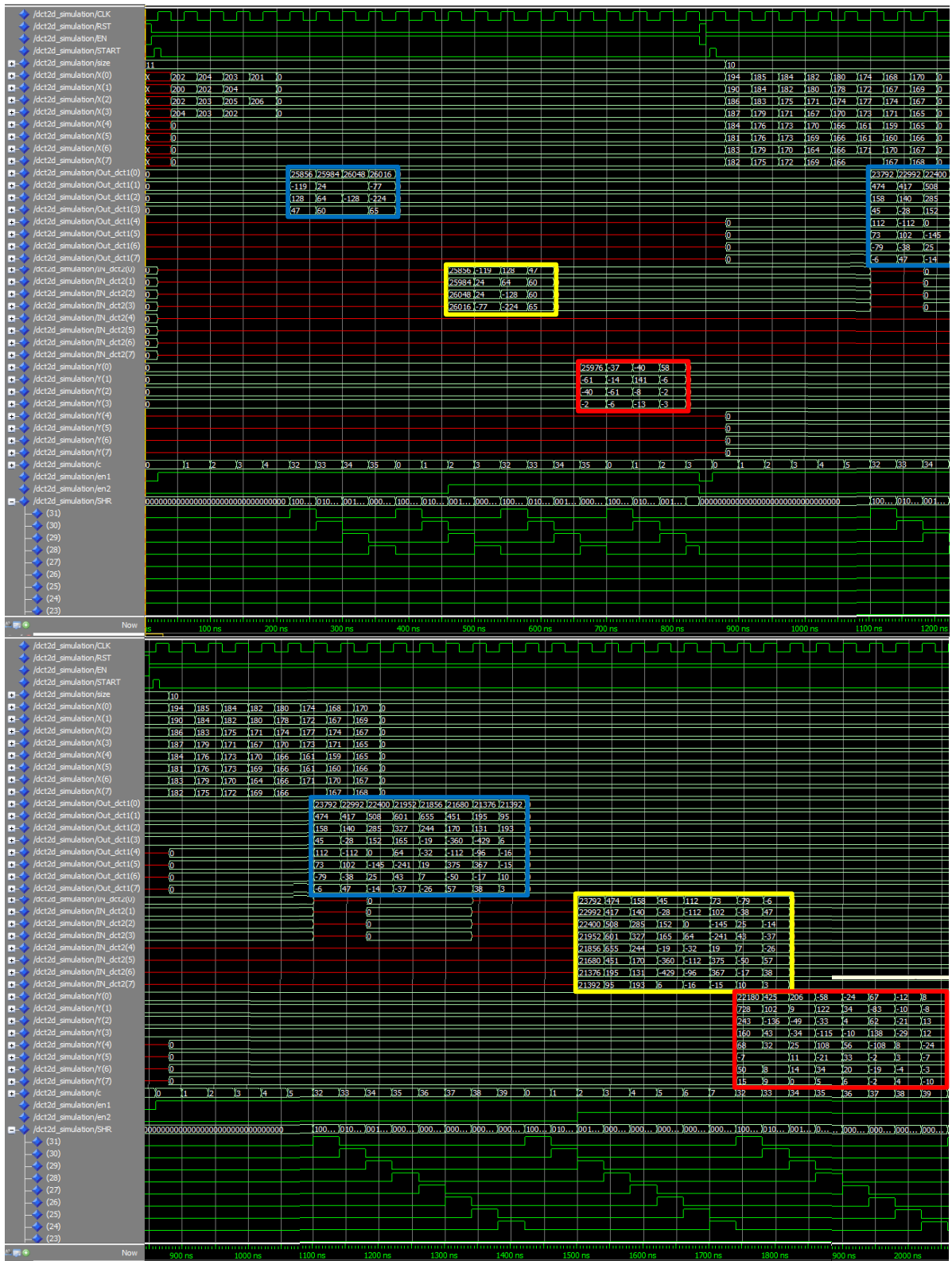


Figura 6.21 Resultados de la simulación funcional de la transformada DCT-2D-T para los tamaños 4 y 8-puntos.

6.5 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA DCT INVERSA DE MULTIPLES TAMAÑOS

En esta sección se presenta la simulación funcional de las transformadas IDCT-1D-T y IDCT-2D-T, las cuales pueden calcular cualquiera de los tamaños de la transformada IDCT-1D y IDCT-2D, respectivamente.

6.5.1 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA IDCT-1D-T

La simulación de la transformada IDCT-1D-T se lleva a cabo usando 4 vectores de entrada, los cuales también hacen referencia a los vectores de datos transformados que se obtuvieron en la simulación de la transformada DCT-1D-T. Lo anterior con el fin de obtener los mismos resultados que se obtuvieron en las simulaciones de las transformadas IDCT-1D, lo cual es equivalente a obtener en la salida a los vectores de datos originales.

El resultado de la simulación funcional de la transformada IDCT-1D-T para los tamaños 4 y 8-puntos se observa en la Figura 6.22. Del mismo modo, la simulación funcional de la transformada IDCT-1D-T para los tamaños 16 y 32-puntos se observa en la Figura 6.23.

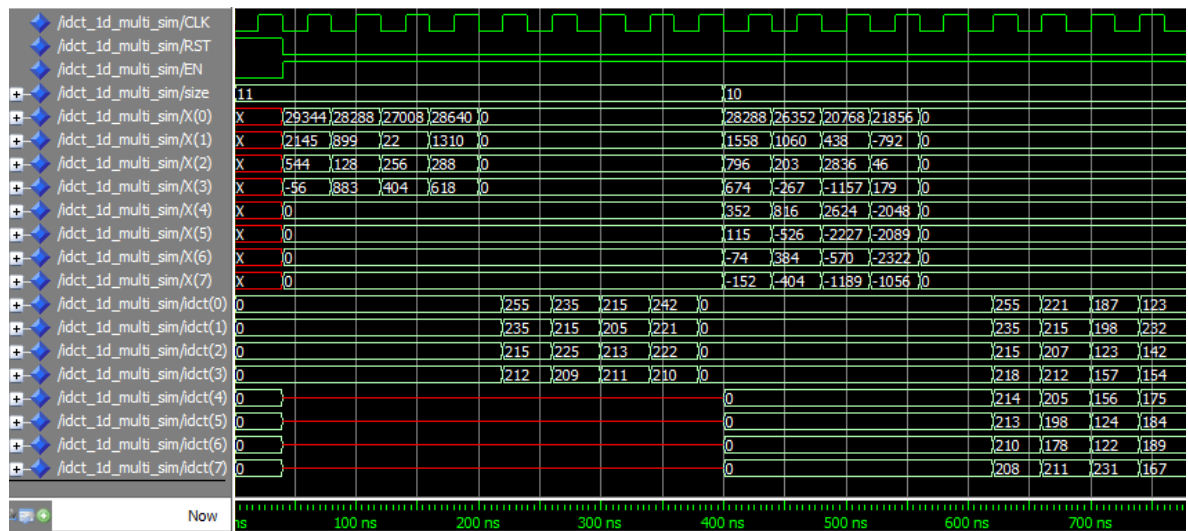


Figura 6.22 Resultados de la simulación funcional de la transformada IDCT-1D-T para los tamaños 4 y 8-puntos.

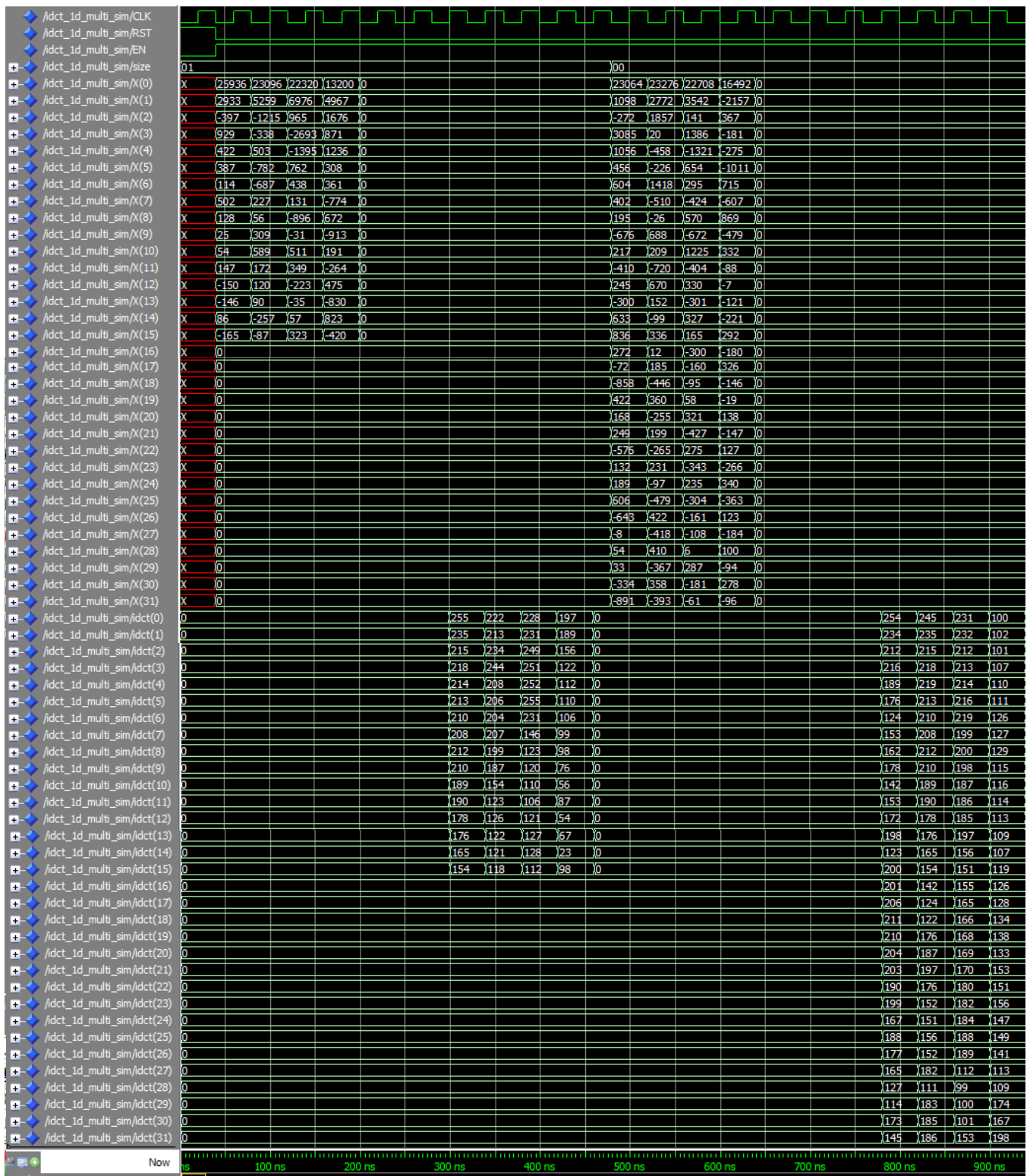


Figura 6.23 Resultados de la simulación funcional de la transformada IDCT-1D-T para los tamaños 16 y 32-puntos.

En cuanto a la verificación de los resultados, esta no se presenta en las anteriores figuras ya que los resultados obtenidos son los mismos que se obtuvieron en las simulaciones de las transformadas DCT-1D, por lo tanto, estos resultados ya fueron verificados anteriormente y no es necesario realizar una nueva verificación.

6.5.2 SIMULACIÓN FUNCIONAL DE LA TRANSFORMADA IDCT-2D-T

La simulación de la transformada IDCT-2D-T se lleva a cabo usando 4 matrices de entrada, una para cada tamaño de la transformada. En este caso estas matrices hacen referencia a las matrices de datos transformados obtenidos en la simulación de la transformada DCT-2D-T. Lo anterior con el fin de obtener los mismos resultados que se obtuvieron en las simulaciones de las transformadas IDCT-2D, lo cual es equivalente a obtener en la salida a las matrices de datos originales.

El resultado de la simulación funcional de la transformada IDCT-2D-T para los tamaños 4 y 8-puntos se observa en la Figura 6.24. En esta figura se incluyen los resultados de cada uno de los bloques que conforman esta arquitectura. El recuadro azul resalta los resultados del primer bloque IDCT-1D-T. Por su parte, el recuadro amarillo resalta los resultados del bloque de trasposición y finalmente, el recuadro rojo resalta los resultados de la segunda IDCT-1D-T.

En la Figura 6.24, también se presentan algunas de las señales de control que intervienen en el cálculo de la transformada IDCT-2D-T tales como: las señales de habilitación de los bloques IDCT-1D-T, las señales de habilitación de escritura de las memorias del bloque de trasposición, la señal “size” que permite seleccionar el tamaño de la transformada, y finalmente, el valor del contador.

En cuanto a los resultados de simulación de la transformada DCT-1D-T para los tamaños 16 y 32, estos hacen referencia a los resultados que fueron presentados en las Figura 6.17 y 6.18.

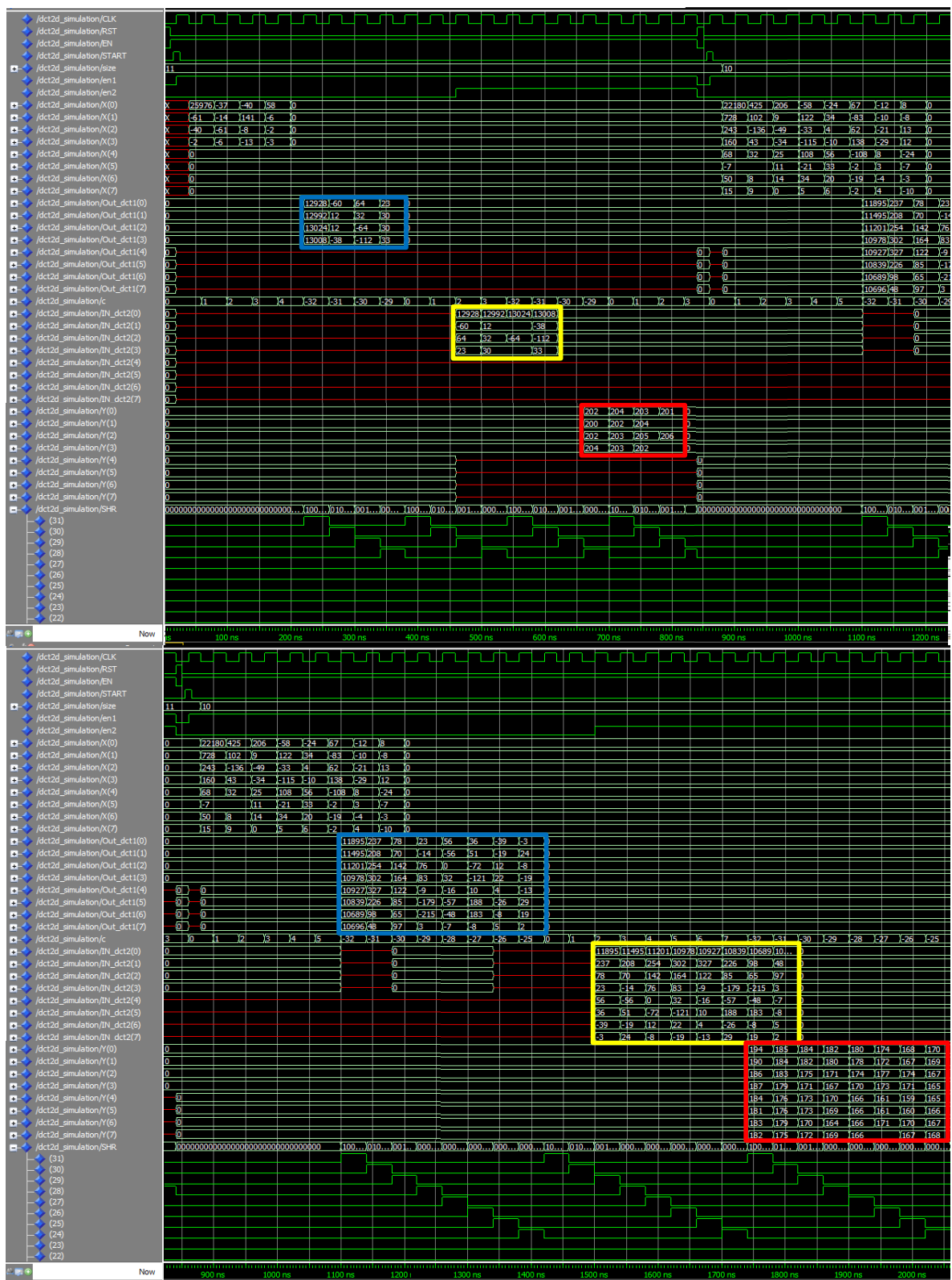


Figura 6.24 Resultados de la simulación funcional de la transformada IDCT-2D-T para los tamaños 4 y 8-puntos.

Capítulo 7

RESULTADOS DE SÍNTESIS Y DESEMPEÑO

7.1 INTRODUCCIÓN

En éste capítulo, se presentan los resultados de síntesis y desempeño para cada uno de los diseños propuestos, los cuales fueron sintetizados sobre el FPGA Stratix V 5SGXMABN3F45I4 usando Quartus II versión 13.0. Los resultados de velocidad se obtuvieron sin tener en cuenta restricciones de síntesis y están basados en el modelo lento a 900mV y a 85°C. Además, se presenta una comparación entre los resultados de síntesis y desempeño de algunas de las arquitecturas propuestas y otros trabajos presentados en la literatura.

El Throughput obtenido para cada una de las arquitecturas propuestas corresponde al producto entre la frecuencia máxima de operación y el número de datos procesados en cada ciclo de reloj, y está definido en unidades Gbps (Giga-samples per second).

Cabe resaltar que el FPGA Stratix V 5SGXMABN3F45I4 posee un total de 359.200 ALMs y 54'067.200 bits de memoria. Teniendo en cuenta estos valores, el lector puede determinar de forma más clara el porcentaje de elementos lógicos utilizados en los diseños propuestos.

Este capítulo está organizado de la siguiente manera. Inicialmente en la sección 7.2 se presentan los resultados de síntesis y desempeño de los diseños de la transformada DCT que incluye a las transformadas DCT-1D-E, DCT-1D-M, DCT-1D-P para el caso de las transformadas unidimensionales y a las transformadas DCT-2D-E, DCT-2D-M, DCT-2D-P para el caso de las transformadas bidimensionales. En la sección 7.3 se presentan los resultados de síntesis y desempeño de los diseños de la transformada DCT inversa que incluye a las transformadas IDCT-1D-E, IDCT-1D-M, IDCT-1D-P para el caso de las transformadas unidimensionales y a las transformadas IDCT-2D-E, IDCT-2D-M, IDCT-2D-P para el caso de las

transformadas bidimensionales. En la sección 7.4 se presentan los resultados de síntesis y desempeño de la transformada DCT de múltiples tamaños unidimensional y bidimensional. En la sección 7.5 se presentan los resultados de síntesis y desempeño de la transformada DCT inversa de múltiples tamaños unidimensional y bidimensional. Finalmente, en la sección 7.6 se presenta una comparación entre los resultados de síntesis y desempeño de las arquitecturas propuestas en este trabajo y otras presentadas en la literatura.

7.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT

En esta sección se presentan los resultados de síntesis y desempeño de cada uno de los diseños propuestos para las transformadas DCT-1D y DCT-2D teniendo en cuenta los requerimientos del estándar H.265/HEVC.

7.2.1 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT-1D

Tal como se mencionó en capítulos anteriores, las transformadas DCT-1D se utilizan como bloques funcionales en la arquitectura hardware de la transformada DCT-2D. Teniendo en cuenta lo anterior, en esta sección se presentan los resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas DCT-1D utilizadas en la primera y segunda etapa de transformación de la DCT-2D.

Las transformadas DCT-1D utilizadas tanto en la primera como en la segunda etapa de transformación generan datos de salida de 16 bits, pero difieren en cuanto al número de bits de los datos de entrada. Para el primer caso las entradas son datos de $n=9$ bits y para el segundo caso son datos de $n=16$ bits.

Los resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas DCT-1D para el estándar H.265/HEVC se presentan en la Tabla 7.1.

Para el caso de las transformadas DCT-1D de 32-puntos utilizadas en la segunda etapa de transformación (datos de entrada de 16 bits), no es posible obtener los resultados de velocidad debido a que estos diseños exceden el número de pines disponibles en el FPGA.

<i>Diseño</i>	<i>n (IN)</i>	<i>N</i>	<i>ALMs</i>	<i>Registros</i>	<i>Latencia (Ciclos)</i>	<i>Frecuencia (MHz)</i>	<i>Throughput (Gbps)</i>
DCT-1D-E	9	4	108	92	2	226,09	0,90
		8	417	198	2	157,06	1,26
		16	1554	408	2	141,7	2,27
		32	5664	826	2	99,18	3,17
	16	4	171	132	2	192,01	0,77
		8	624	264	2	150,63	1,21
		16	2304	528	2	113,61	1,82
		32	8177	1056	2	*	*
DCT-1D-P	9	4	133	288	5	630,12	2,52
		8	487	973	6	558,66	4,47
		16	1751	3472	7	460,41	7,37
		32	6373	13041	8	431,78	13,82
	16	4	212	439	5	575,71	2,30
		8	752	1458	6	505,05	4,04
		16	2660	5132	7	439,95	7,04
		32	9776	18731	8	*	*
DCT-1D-M	9	4	96	92	2	231,37	0,93
		8	357	198	2	152,56	1,22
		16	1305	408	2	125,74	2,01
		32	4474	826	2	96,54	3,09
	16	4	153	132	2	192,75	0,77
		8	547	264	2	142,15	1,14
		16	1920	528	2	113,19	1,81
		32	6552	1056	2	*	*

* Pines insuficientes para síntesis

Tabla 7.1 Resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas DCT-1D para el estándar H.265/HEVC

7.2.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT-2D

Los resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas DCT-2D para el estándar H.265/HEVC se presentan en la Tabla 7.2. Cabe resaltar que estos diseños procesan N datos de entrada de 9 bits y generan N datos de salida de 16 bits, donde N es el tamaño de la transformada.

<i>Diseño</i>	<i>N</i>	<i>ALMs</i>	<i>Registros</i>	<i>Bits de Memoria</i>	<i>Latencia (Ciclos)</i>	<i>Frecuencia (MHz)</i>	<i>Throughput (Gbps)</i>
DCT-2D-E	4	283	242	512	10	191,42	0,77
	8	1046	486	2048	14	148,32	1,19
	16	3865	983	8192	22	113,7	1,82
	32	13768	1964	32768	38	87,83	2,81
DCT-2D-P	4	350	762	512	16	419,11	1,68
	8	1243	2528	2048	22	369,14	2,95
	16	4423	8826	8192	32	273	4,37
	32	15909	30613	32768	50	213,9	6,84
DCT-2D-M	4	253	242	512	10	203,17	0,81
	8	913	487	2048	14	138,06	1,10
	16	3236	979	8192	22	113,84	1,82
	32	10954	1964	32768	38	89,1	2,85

Tabla 7.2 Resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas DCT-2D para el estándar H.265/HEVC

7.3 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT INVERSA

En esta sección se presentan los resultados de síntesis y desempeño de cada uno de los diseños propuestos para las transformadas IDCT-1D y IDCT-2D teniendo en cuenta los requerimientos del estándar H.265/HEVC.

7.3.1 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA IDCT-1D

Esta sección presenta los resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas IDCT-1D utilizadas en la primera y segunda etapa de transformación de la arquitectura hardware de la transformada IDCT-2D.

Las transformadas IDCT-1D utilizadas tanto en la primera como en la segunda etapa de transformación procesan datos de entrada de 16 bits, pero difieren en cuanto al número de bits de los datos de salida. Para el primer caso las salidas son datos de n=16 bits y para el segundo caso son datos de n=9 bits.

Los resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas IDCT-1D para el estándar H.265/HEVC se presentan en la Tabla 7.3.

Para el caso de las transformadas IDCT-1D de 32-puntos utilizadas en la primera etapa de transformación (datos de salida de 16 bits), no es posible obtener los resultados de velocidad debido a que estos diseños exceden el número de pines disponibles en el FPGA.

<i>Diseño</i>	<i>n (OUT)</i>	<i>N</i>	<i>ALMs</i>	<i>Registros</i>	<i>Latencia (Ciclos)</i>	<i>Frecuencia (MHz)</i>	<i>Throughput (Gsps)</i>
IDCT-1D-E	16	4	177	132	2	195,27	0,78
		8	624	264	2	143,6	1,15
		16	2249	528	2	106,8	1,71
		32	7712	1056	2	*	*
	9	4	149	100	2	207,9	0,83
		8	538	200	2	147,97	1,18
		16	1883	400	2	117,45	1,88
		32	6568	800	2	89,22	2,86
IDCT-1D-P	16	4	217	440	5	570,78	2,28
		8	780	1453	6	453,72	3,63
		16	2695	4982	7	440,92	7,05
		32	8799	16893	8	*	*
	9	4	186	384	5	586,51	2,35
		8	677	1297	6	474,61	3,80
		16	2333	4514	7	482,86	7,73
		32	7997	15419	8	392,93	12,57
IDCT-1D-M	16	4	159	132	2	196,43	0,79
		8	549	264	2	143,66	1,15
		16	1869	528	2	114,34	1,83
		32	6191	1056	2	*	*
	9	4	133	100	2	208,07	0,83
		8	476	200	2	153,66	1,23
		16	1616	400	2	118,85	1,90
		32	5530	800	2	99,44	3,18

* Pines insuficientes para síntesis

Tabla 7.3 Resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas IDCT-1D para el estándar H.265/HEVC

7.3.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA IDCT-2D

Los resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas IDCT-2D para el estándar H.265/HEVC se presentan en la Tabla 7.4. Cabe resaltar que estos diseños procesan N datos de entrada de 16 bits, y generan N datos de salida de 9 bits.

<i>Diseño</i>	<i>N</i>	<i>ALMs</i>	<i>Registros</i>	<i>Bits de Memoria</i>	<i>Latencia (Ciclos)</i>	<i>Frecuencia (MHz)</i>	<i>Throughput (Gsps)</i>
IDCT-2D-E	4	319	254	480	10	200,44	0,80
	8	1134	503	1920	14	145,1	1,16
	16	3937	1011	7680	22	110,14	1,76
	32	13646	1956	30720	38	90,15	2,88
IDCT-2D-P	4	399	878	480	16	401,77	1,61
	8	1428	2861	1920	22	340,02	2,72
	16	4937	9660	7680	32	292,48	4,68
	32	16834	32506	30720	50	222,97	7,14
IDCT-2D-M	4	286	251	480	10	203,87	0,82
	8	1003	487	1920	14	151,17	1,21
	16	3399	1001	7680	22	115,17	1,84
	32	11533	1980	30720	38	94,46	3,02

Tabla 7.4 Resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas IDCT-2D para el estándar H.265/HEVC

7.4 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT DE MULTIPLES TAMAÑOS

En esta sección se presentan los resultados de síntesis y desempeño de las arquitecturas hardware DCT-1D-T y DCT-2D-T, las cuales pueden calcular cualquiera de los tamaños establecidos por el estándar H.265/HEVC para las transformadas DCT-1D y DCT-2D, respectivamente.

7.4.1 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT-1D-T

Esta sección presenta los resultados de síntesis y desempeño de la transformada DCT-1D-T, la cual es utilizada en la primera y segunda etapa de transformación de la arquitectura hardware de la transformada DCT-2D-T.

Las transformadas DCT-1D-T utilizadas tanto en la primera como en la segunda etapa de transformación generan datos de salida de 16 bits, pero difieren en cuanto al número de bits de los datos de entrada. Para el primer caso las entradas son datos de $n=9$ bits y para el segundo caso son datos de $n=16$ bits.

Los resultados de síntesis y desempeño de las arquitecturas hardware de las transformadas DCT-1D-T para el estándar H.265/HEVC se presentan en la Tabla 7.5.

Para el caso de la transformada DCT-1D-T utilizada en la segunda etapa de transformación (datos de entrada de 16 bits), no es posible obtener los resultados de velocidad debido a que este diseño excede el número de pines disponibles en el FPGA.

<i>Diseño</i>	<i>n (IN)</i>	<i>ALMs</i>	<i>Registros</i>	<i>Frecuencia (MHz)</i>	<i>Size</i>	<i>Latencia (Ciclos)</i>	<i>Throughput (Gbps)</i>
DCT-1D-T	9	6651	12231	389,71	4	5	1,56
					8	6	3,12
					16	7	6,24
					32	8	12,47
	16	9597	17811	*	4	5	*
					8	6	*
					16	7	*
					32	8	*

Tabla 7.5 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada DCT-1D-T para el estándar H.265/HEVC

7.4.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT-2D-T

Los resultados de síntesis y desempeño de la transformada DCT-2D-T para el estándar H.265/HEVC se presentan en la Tabla 7.6. Cabe resaltar que este diseño procesa N datos de entrada de 9 bits, y genera N datos de salida de 16 bits, donde N depende del tamaño de la transformada seleccionado.

<i>Diseño</i>	<i>ALMs</i>	<i>Registros</i>	<i>Bits de Memoria</i>	<i>Frecuencia (MHz)</i>	<i>Size</i>	<i>Latencia (Ciclos)</i>	<i>Throughput (Gbps)</i>
DCT-2D-T	17044	30173	32768	175,19	4	16	0,70
					8	22	1,40
					16	32	2,80
					32	50	5,61

Tabla 7.6 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada DCT-2D-T para el estándar H.265/HEVC

7.5 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA DCT INVERSA DE MULTIPLES TAMAÑOS

En esta sección se presentan los resultados de síntesis y desempeño de las arquitecturas hardware IDCT-1D-T y IDCT-2D-T, las cuales pueden calcular cualquiera de los tamaños establecidos por el estándar H.265/HEVC para las transformadas IDCT-1D y IDCT-2D, respectivamente.

7.5.1 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA IDCT-1D-T

Esta sección presenta los resultados de síntesis y desempeño de la transformada IDCT-1D-T, la cual es utilizada en la primera y segunda etapa de transformación de la arquitectura hardware de la transformada IDCT-2D-T.

Las transformadas IDCT-1D-T utilizadas tanto en la primera como en la segunda etapa de transformación procesan datos de entrada de 16 bits, pero difieren en cuanto al número de bits de los datos de salida. Para el primer caso las salidas son datos de n=16 bits y para el segundo caso son datos de n=9 bits.

Los resultados de síntesis y desempeño de la transformada IDCT-1D-T para el estándar H.265/HEVC se presentan en la Tabla 7.7.

Para el caso de la transformada IDCT-1D-T utilizada en la primera etapa de transformación (datos de salida de 16 bits), no es posible obtener los resultados de velocidad debido a que este diseño excede el número de pines disponibles en el FPGA.

<i>Diseño</i>	<i>n (OUT)</i>	<i>ALMs</i>	<i>Registros</i>	<i>Frecuencia (MHz)</i>	<i>Size</i>	<i>Latencia (Ciclos)</i>	<i>Throughput (Gbps)</i>
IDCT-1D-T	16	8993	16893	*	4	5	*
					8	6	*
					16	7	*
					32	8	*
	9	8363	15419	336,47	4	5	1,35
					8	6	2,69
					16	7	5,38
					32	8	10,77

Tabla 7.7 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada IDCT-1D-T para el estándar H.265/HEVC

7.5.2 RESULTADOS DE SÍNTESIS Y DESEMPEÑO DE LA TRANSFORMADA IDCT-2D-T

Los resultados de síntesis y desempeño de la transformada IDCT-2D-T para el estándar H.265/HEVC se presentan en la Tabla 7.8. Cabe resaltar que este diseño procesa N datos de entrada de 16 bits, y genera N datos de salida de 9 bits, donde N depende del tamaño de la transformada seleccionado.

<i>Diseño</i>	<i>ALMs</i>	<i>Registros</i>	<i>Bits de Memoria</i>	<i>Frecuencia (MHz)</i>	<i>Size</i>	<i>Latencia (Ciclos)</i>	<i>Throughput (Gps)</i>
IDCT-2D-T	17615	31952	30720	184,88	4	16	0,74
					8	22	1,48
					16	32	2,96
					32	50	5,92

Tabla 7.8 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada IDCT-2D-T para el estándar H.265/HEVC

7.6 COMPARACION CON OTROS TRABAJOS

En esta sección se presenta una comparación entre los resultados de síntesis y desempeño de algunas de las arquitecturas propuestas en este trabajo y otros trabajos presentados en la literatura. En primer lugar se realiza la comparación entre las arquitecturas hardware de la transformada DCT-1D y los trabajos presentados en [12] y [25]. Los diseños presentados en [12] fueron sintetizados sobre el mismo dispositivo que los diseños propuestos, por lo tanto, es posible realizar una comparación objetiva entre los dos trabajos. Por otra parte, los diseños presentados en [25] no son sintetizados sobre el mismo dispositivo, debido a esto y con el fin de realizar una comparación justa entre los diseños propuestos y los diseños presentados en [25], se rediseñaron las arquitecturas propuestas usando los mismos bloques funcionales que se describieron en capítulos anteriores pero reemplazando los bloques MCM por las unidades de suma y desplazamiento (SAU: Shift-Add Units) presentadas en [25].

En [12] también se presentan los resultados de síntesis y desempeño del algoritmo de referencia de la transformada DCT-1D para HEVC, los cuales también se tienen en cuenta en esta comparación con el fin de obtener una medida de la eficiencia de los diseño propuestos.

Los diseños presentados en [12] y [25] no utilizan etapas de pipeline, por lo tanto la arquitectura hardware DCT-1D-P no se tiene en cuenta en este caso.

Los resultados de síntesis y desempeño de los diseños propuestos para las transformadas DCT-1D y los diseños presentados en [12] y [25] se presentan en la Tabla 7.9.

<i>N</i>	<i>Diseño</i>	<i>ALMs</i>	<i>Dec. en Área (%)</i>	<i>Frecuencia (Mhz)</i>	<i>Inc. en Frecuencia (%)</i>	<i>Pixeles /Ciclo</i>	<i>Throughput (Gsp/s)</i>
4	Algoritmo de Referencia [12]	130	-	144.51	-	4	0,58
	1D-DCT [12]	128	1,5	128.92	-10,8	4	0,52
	DCT-1D-E usando SAU [25]	108	16,9	187.27	29,6	4	0.75
	DCT-1D-E	108	16,9	226.09	56,5	4	0,90
	DCT-1D-M	96	26,2	231,37	60,1	4	0,93
8	Algoritmo de Referencia [12]	564	-	107.45	-	8	0,86
	1D-DCT [12]	454	19,5	93.65	-12,8	8	0,75
	DCT-1D-E usando SAU [25]	387	31,4	141.7	31,9	8	1.13
	DCT-1D-E	417	26,1	157.06	46,2	8	1.26
	DCT-1D-M	357	36,7	152,56	42,0	8	1.22
16	Algoritmo de Referencia [12]	2053	-	76.61	-	16	1,22
	1D-DCT [12]	1304	36,5	79.11	3,3	16	1,26
	DCT-1D-E usando SAU [25]	1488	27,5	108.97	42,2	16	1.74
	DCT-1D-E	1555	24,3	123.92	61,8	16	1,98
	DCT-1D-M	1304	36,5	125.74	64,1	16	2,01
32	Algoritmo de Referencia [12]	7500	-	52.66	-	32	1,68
	1D-DCT [12]	4058	45,9	67.44	28,1	32	2,16
	DCT-1D-E usando SAU [25]	5525	26,3	91.77	74,3	32	2.93
	DCT-1D-E	5664	24,5	99.18	88,3	32	3,17
	DCT-1D-M	4128	45,0	96.54	83,3	32	3,09

Tabla 7.9 Resultados de síntesis y desempeño de las arquitecturas hardware para la transformada DCT-1D y otros trabajos presentados en la literatura

A partir de la Tabla 7.9 se puede observar que los diseños propuestos presentan mayor frecuencia de operación que los trabajos presentados en [12] y [25]. En cuanto al área, la arquitectura de la transformada DCT-1D-M es la que presenta mejores resultados, esto indica que los bloques MCM modificados consumen menor área y tienen mayor frecuencia de operación que los bloques SAU presentados en [25].

A continuación se comparan las arquitecturas hardware de la transformada DCT-2D de 16-puntos con el trabajo presentado en [27]. Los resultados de síntesis y desempeño de estos diseños se presentan en la Tabla 7.10.

	<i>FPGA</i>	<i>LUTs</i>	<i>Flip-Flops</i>	<i>Slices</i>	<i>Lat. Inicial</i>	<i>Latencia Intermedia</i>	<i>Frecuencia (MHz)</i>	<i>Throughput (Gbps)</i>
16x16 DCT [27]	Virtex 6	4724	5454	1371	16	16	250	2.0
		<i>ALMs</i>	<i>Regs</i>	<i>Mem.</i>				
DCT-2D-E 16p	Stratix V	3865	983	8192	22	0	113,7	1,82
DCT-2D-P 16p		4423	8826	8192	32	0	273	4,37
DCT-2D-M 16p		3236	979	8192	22	0	113,84	1,82

Tabla 7.10 Resultados de síntesis y desempeño de las arquitecturas hardware para la transformada DCT-2D de 16-puntos y otro trabajo presentado en la literatura

El trabajo presentado en [27] no fue sintetizado sobre el mismo dispositivo que los diseños propuestos en este trabajo, por lo tanto no es posible realizar una comparación objetiva. Sin embargo, en términos generales se puede concluir que la arquitectura hardware de la transformada DCT-2D de 16-puntos tiene mayor frecuencia de operación y *throughput*. En este caso, para calcular el *throughput* se tuvieron en cuenta la latencia inicial y la latencia intermedia del diseño. El trabajo presentado en [27] procesa una matriz de 16x16 datos de entrada cada 16 ciclos de reloj, mientras que las arquitecturas propuestas en este trabajo procesan matrices de 16x16 datos en forma consecutiva, por lo tanto los diseños propuestos tienen el doble de la capacidad de procesamiento que el diseño presentado en [27].

A continuación se comparan las arquitecturas hardware de la transformada IDCT-2D con el trabajo presentado en [32]. Los resultados de síntesis y desempeño de estos diseños se presentan en la Tabla 7.11. El diseño presentado en [32] no usa etapas de pipeline por lo tanto la arquitectura de la transformada IDCT-2D-P no se tiene en cuenta en este caso.

Los diseños presentados en [32] se sintetizaron sobre el mismo dispositivo que los diseños propuestos en este trabajo, por lo tanto es posible hacer una comparación objetiva.

<i>N</i>	<i>Diseño</i>	<i>ALMs</i>	<i>Regs</i>	<i>Bits de Memoria</i>	<i>Frecuencia (Mhz)</i>	<i>Inc. en Frecuencia (%)</i>
4	IDCT-2D [32]	425	391	0	116,43	-
	IDCT-2D-E	319	254	480	200,44	72,15
	IDCT-2D-M	286	251	480	203,87	75,10
8	IDCT-2D [32]	1478	1373	0	94,49	-
	IDCT-2D-E	1134	503	1920	145,1	53,56
	IDCT-2D-M	1003	487	1920	151,17	59,99
16	IDCT-2D [32]	4755	4,793	0	76,76	-
	IDCT-2D-E	3937	1011	7680	110,14	43,49
	IDCT-2D-M	3399	1001	7680	115,17	50,04
32	IDCT-2D [32]	16594	18234	0	68,12	-
	IDCT-2D-E	13646	1956	30720	90,15	32,34
	IDCT-2D-M	11533	1980	30720	94,46	38,67

Tabla 7.11 Resultados de síntesis y desempeño de las arquitecturas hardware para la transformada IDCT-2D y otro trabajo presentado en la literatura

A partir de la tabla anterior se puede observar que las arquitecturas hardware de las transformadas IDCT-2D-E y IDCT-2D-M tienen mayor frecuencia de operación de los diseños presentados en [32]. Además, los diseños propuestos consumen menor cantidad de ALMs y Registros pero consumen una mayor cantidad de bits de memoria.

A continuación se comparan las arquitecturas hardware de la transformada IDCT-2D-T con el trabajo presentado en [32]. Los resultados de síntesis y desempeño de estos diseños se presentan en la Tabla 7.12.

<i>Diseño</i>	<i>ALMs</i>	<i>Registros</i>	<i>Bits de Memoria</i>	<i>Frecuencia (MHz)</i>
Multisize IDCT-2D [32]	17340	18409	0	63,25
IDCT-2D-T	17615	31952	30720	184,88

Tabla 7.12 Resultados de síntesis y desempeño de la arquitectura hardware de la transformada IDCT-2D-T y otro trabajo presentado en la literatura.

A partir de la tabla anterior se puede observar que el diseño de la transformada IDCT-2D-T utiliza más registros que el diseño presentado en [32], lo anterior es debido a que la arquitectura propuesta es completamente pipeline, lo cual permite obtener una frecuencia de operación 3 veces mayor a la del diseño presentado en [32].

Capítulo 8

CONCLUSIONES Y TRABAJO FUTURO

8.1 CONCLUSIONES

Este trabajo presenta tres implementaciones en hardware para la transformada DCT-2D directa e inversa, de 4, 8, 16 y 32-puntos. Las características principales de los diseños propuestos son: alta velocidad de procesamiento y bajo consumo de área. También, se presenta una arquitectura hardware altamente parametrizable desde el punto de vista del número de puntos para la transformada DCT-2D directa e inversa. Todos los diseños presentados en este trabajo fueron implementados en hardware cumpliendo con los requerimientos del estándar H.265/HEVC.

Las arquitecturas hardware de las transformadas DCT-1D de N-puntos son implementadas utilizando: un bloque de diagrama mariposa parcial, $N/2$ bloques MCMs (MCM: *Multiple Constant Multiplication*), $N/2$ bloques sumadores, y una transformada DCT-1D de $N/2$ -puntos.

Los bloques MCMs son implementados utilizando sumas y desplazamientos, en donde se utilizan solo dos etapas de sumas sucesivas con el fin de minimizar el tiempo de cálculo, y así incrementar la frecuencia de operación.

Las operaciones de desplazamiento en los bloques MCMs no consumen recursos hardware ya que estas son implementadas en forma cableada. Adicionalmente, los datos de entrada y salida de los bloques de suma y resta son representados utilizando la mínima cantidad de bits, lo cual disminuye el consumo de área sin afectar la precisión de los resultados.

Los bloques MCMs y los bloques sumadores fueron modificados utilizando un enfoque o aproximación empírico, en donde se utiliza la menor cantidad de restadores y se usan

sumadores paralelos de 3 entradas. Esto permite reducir el consumo de área de las arquitecturas hardware de las transformadas DCT.

Adicionar etapas de registros en la entrada y salida de cada bloque de suma y cada bloque de resta de las arquitecturas hardware de las transformadas DCT-1D permite procesar nuevos datos en cada ciclo de reloj mejorando el desempeño de la arquitectura, es decir, incrementando la frecuencia de operación y el *throughput*.

Los diseños hardware de la transformada DCT-2D/IDCT-2D son implementadas teniendo en cuenta la propiedad de separabilidad de la transformada DCT, lo cual permite diseñar una arquitectura basada en dos etapas de transformación DCT-1D/IDCT-1D y un bloque transposición intermedio.

El diseño hardware del bloque de transposición se implementa utilizando N memorias RAMs modificadas de dos puertos. Estas memorias son diseñadas de tal forma que el número de datos a almacenar es mayor que el número de datos a leer. Cada memoria tiene 2 sectores de escritura y cada sector esta subdividido en N posiciones de 16 bits, es decir la capacidad total es 2N posiciones de memoria de 16 bits, donde N es el número de entradas o puntos de la transformada. En este caso, las memorias tienen el doble de la capacidad necesaria para el proceso de trasposición con el fin de realizar una segunda trasposición sin ciclos de latencia intermedios. Lo anterior permite que se procesen datos continuamente incrementando el *throughput* del diseño.

El diseño hardware de la transformada DCT-1D parametrizable para múltiples tamaños (DCT-1D-T) integra todos los tamaños de la transformada DCT-1D para el estándar H.265/HEVC en un solo núcleo. Este diseño evita el uso de una arquitectura hardware específica para cada uno de los tamaños de la transformada DCT-1D y disminuye el consumo de recursos de hardware. Esta arquitectura es una solución adecuada para aplicaciones que requieren usar un mínimo consumo de hardware.

El diseño de la transformada DCT-1D-T/IDCT-1D-T aprovecha la propiedad de recursividad de la transformada DCT, donde la matriz base de mayor tamaño contiene a la matriz base de

menor tamaño. Teniendo en cuenta lo anterior, el diseño hardware de la transformada DCT-1D-T/IDCT-1D-T se basa en el diseño de la transformada DCT-1D/IDCT-1D de 32-puntos, la cual usa las transformadas DCT/IDTC de 16, 8 y 4-puntos.

La simulación funcional de cada uno de los diseños propuestos fue realizada utilizando el software Modelsim, y los resultados fueron verificados usando Matlab. Es decir, los resultados obtenidos en Modelsim fueron los mismos que se obtuvieron en Matlab. Esto demuestra que la precisión de la transformada DCT no fue afectada al implementar las diversas técnicas de diseño utilizadas en este trabajo.

Los diseños hardware son descritos en VHDL en forma estructural y sintetizados sobre el FPGA Stratix V 5SGXMABN3F45I4 usando Quartus II versión 13.0. Los resultados de velocidad no tienen en cuenta restricciones de síntesis y están basados en el modelo lento a 900mV y a 85°C.

Los resultados de síntesis y desempeño para las arquitecturas basadas en pipeline presentan la mayor frecuencia de operación y *throughput*. Por otra parte, los diseños hardware basados en bloques MCMs modificados y bloques sumadores modificados presentan el menor consumo de área.

La arquitectura de la transformada DCT-2D-T presenta una reducción del 28,6% en ALMs, 41,6% en registros y 32,8% en bits de memoria con respecto a una implementación independiente de cada una de las transformadas DCT-2D de tamaños 4, 8, 16 y 32-puntos. De igual forma, la arquitectura de la transformada IDCT-2D-T presenta una reducción del 33,9% en ALMs, 43,7% en registros y 32,8% en bits de memoria.

En general, las arquitecturas hardware descritas en éste trabajo presentan menor consumo de área, y mayor frecuencia de operación y *throughput* que otras arquitecturas presentadas en la literatura. Lo anterior nos permite concluir que los diseños propuestos son más eficientes y presentan mejor desempeño que trabajos previos.

8.2 TRABAJO FUTURO

El trabajo futuro será orientado a diseñar en hardware otras arquitecturas para la transformada DCT-2D, las cuales consuman la menor cantidad de recursos hardware para ser usadas en sistemas embebidos.

El trabajo futuro también será orientado a diseñar en hardware otros bloques funcionales del estándar H.265/HEVC tales como: cuantización, codificador de entropía, filtro de efecto bloque, predicción inter-frame, etc. Lo anterior con el fin de implementar en hardware un procesador de video basado en el estándar H.265/HEVC.

BIBLIOGRAFIA

- [1]. G. Sullivan, J. Ohm, W. Han and T. Wiegand. "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions on Circuits and systems for Video Technology, Vol. 22, No. 12, December 2012.
- [2]. F. Bossen, B. Bross, K. Suhling and D. Flynn, "HEVC Complexity and Implementation Analysis". IEEE Transactions on Circuits and systems for Video Technology, Vol. 22, No. 12, December 2012.
- [3]. R. Jeske, J. de Souza Jr, G. Wrege, R. Conceição, M. Grellert, J. Mattos, L. Agostini. "Low Cost and High Throughput Multiplierless Design of a 16 Point 1-D DCT of the New HEVC Video Coding Standard". IEEE. 2012.
- [4]. M. Winken, P. Helle, D. Marpe, H. Schwarz, and T. Wiegand. "Transform Coding in the HEVC Test Model". 18th IEEE International Conference on Image Processing. 2011.
- [5]. S. Shen, W. Shen, Y. Fan, X. Zeng. "A Unified 4/8/16/32-Point Integer IDCT Architecture for Multiple Video Coding Standards". IEEE International Conference on Multimedia and Expo. 2012.
- [6]. I. Richardson. "The H.264 Advanced Video Compression Standard". 2a.Ed. John Wiley & Sons. 2010.
- [7]. L. Agostini. "Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas a Compressão de Vídeo Segundo o Padrão H.264/AVC". Tesis Doctoral (Doutorado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre. 2007.
- [8]. "Entropia (información)". [http://es.wikipedia.org/wiki/Entrop%C3%ADa_\(informaci%C3%B3n\)](http://es.wikipedia.org/wiki/Entrop%C3%ADa_(informaci%C3%B3n)).
- [9]. P. Gajjala. "Efficient HEVC Loss-less Coding using Sample Based Angular Intra Prediction (SAP)". Tesis de Maestría (Master of Science in Electrical Engineering). University of Texas at Arlington. 2013.
- [10]. A. Diaz. "Transcodificación de Vídeo H.264/HEVC". Tesis de Maestría. Universidad de Castilla-La Mancha. 2011.
- [11]. M. Flå. "Quality Evaluations and algorithmic Improvement of the next Generation Video Coding – HEVC". Tesis de Maestría (Master of science in Electronics). Norwegian University of Science and Technology. 2011.
- [12]. R. Jeske. "Otimizações Algorítmicas e Desenvolvimento Arquitetural para as DCTs do HEVC". Tesis de Maestría. Universidad Federal de Pelotas. 2013.
- [13]. Fraunhofer Heinrich Hertz Institute, Web page: <https://hevc.hhi.fraunhofer.de/>
- [14]. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 – HM12: High Efficiency Video Coding (HEVC) Test Model 16 Improved Encoder Description Update 2. Document: JCTVC-N1002. 20th Meeting: Geneva, 10 – 18 Feb. 2015.
- [15]. H. Samet, "The quadtree and related hierarchical data structures", Comput. Survey, vol. 16, no. 2, pp. 187–260, Jun. 1984.

- [16]. M. Budagavi, A. Fuldseth, G. Bjøntegaard, V. Sze, M. Sadafale. "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard". IEEE Journal of Selected Topics in Signal Processing, Vol. 7, No. 6, Dec. 2013.
- [17]. H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low complexity transform and quantization in H.264/AVC". IEEE Transaction on Circuits and systems for Video Technology, vol. 13, no. 7, pp. 598–603, Jul. 2003.
- [18]. J. D. Bolaños-Jojoa, "Implementación en Hardware de las Transformadas rápidas de Walsh, Hadamard y Walsh-Hadamard Ordenada usando FPGAs". Tesis de Pregrado en Ingeniería Electrónica, Universidad del Valle, Cali – Colombia, 2012.
- [19]. D. Salomon, G. Motta, "Handbook of Data Compression", 5th ed. Springer-Verlag London Limited 2010.
- [20]. W. H. Chen, C. H. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," IEEE Transaction on Communications, vol. 25, no. 9, pp. 1004–1009, September 1977.
- [21]. J. D. Bolaños-Jojoa and J. Velasco-Medina, "Efficient Hardware Design of N-point 1D-DCT for HEVC". IX IEEE Southern Programmable Logic Conference, SPL, Buenos Aires, Argentina, Nov 2014.
- [22]. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 - CE10: Core transform design for HEVC. Document: JCTVC-G495. 7th Meeting: Geneva, CH, 21-30 November, 2011.
- [23]. T. Dias, N. Romazy and L. Sousazy, "High Performance Multi-Standard Architecture for DCT Computation in H.264/AVC High Profile and HEVC Codecs", IEEE Conference on Design and Architectures for Signal and Image Processing (DASIP), 2013
- [24]. M. Martuza and K. Wahid. "A Cost Effective Implementation of 8x8 Transform of HEVC from H.264/AVC". 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). 2012.
- [25]. P. K. Meher, S. Y. Park, B. K. Mohanty, K. S. Lim, and C. Yeo. "Efficient Integer DCT Architectures for HEVC". IEEE Transactions on Circuits and systems for Video Technology. 2013.
- [26]. S. Y. Park , P. K. Meher. "Flexible Integer DCT Architectures for HEVC". IEEE International Symposium on Circuits and Systems (ISCAS). 2013.
- [27]. A. Edirisuriya and A. Madanayake, R. J. Cintra, F. M. Bayer. "A Multiplication-free Digital Architecture for 16x16 2-D DCT/DST Transform for HEVC". 27th IEEE Convention of Electrical and Electronics Engineers in Israel. 2012.
- [28]. Y. Ziyou, H. Weifeng, H. Liang, H. Guanghui and M. Zhigan, "Area and Throughput Efficient IDCT/IDST Architecture for HEVC Standard", IEEE International Symposium on Circuits and Systems (ISCAS). 2014.
- [29]. L. Chieh, H. Wen, C. Yung, and C. Yuan, "Low-Cost Video Transform for HEVC", 4th IEEE International Conference on Information Science and Technology (ICIST). 2014
- [30]. R. Jeske, J. de Souza Jr, G. Wrege, R. Conceição, M. Grellert, J. Mattos, L. Agostini. "Low Cost and High Throughput Multiplierless Design of a 16 Point 1-D DCT of the New HEVC

- Video Coding Standard". VIII IEEE Southern Conference on Programmable Logic (SPL). 2012.
- [31]. R. Conceição, J. C. de Souza, R. Jeske, M. Porto, J. Mattos, L. Agostini, "Hardware Design for the 32x32 IDCT of the HEVC Video Coding Standard", 26th IEEE Symposium on Integrated Circuits and Systems Design (SBCCI), 2013.
 - [32]. R. Conceição, J. C. de Souza, R. Jeske, M. Porto, B. Zatt, L. Agostini, "Power Efficient and High Throughput Multi-Size IDCT Targeting UHD HEVC Decoders" IEEE International Symposium on Circuits and Systems (ISCAS), 2014.
 - [33]. W. Zhao, T. Onoye and T. Song, "High-Performance Multiplierless Transform Architecture for HEVC", IEEE International Symposium on Circuits and Systems (ISCAS), 2013.
 - [34]. H. Liang, H. Weifeng, Z. Hui, M. Zhigang, "A Cost Effective 2-D Adaptive Block Size IDCT Architecture for HEVC Standard", IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS), 2013.
 - [35]. S. Shen, W. Shen, Y. Fan, X. Zeng, "A Unified 4/8/16/32-Point Integer IDCT Architecture for Multiple Video Coding Standards". IEEE International Conference on Multimedia and Expo (ICME). 2012.
 - [36]. A. Mostafa, M. Martuza, K. Wahid, "A Variable Block-Size Design of HEVC Transform Using Intra-Coefficient Sharing", 26th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2013.
 - [37]. J. Zhu, Z. Liu, D. Wang, "Fully Pipelined DCT/IDCT/Hadamard Unified Transform Architecture for HEVC Codec", IEEE International Symposium on Circuits and Systems (ISCAS), 2013 .
 - [38]. A. Ahmed, M. Awais, M. Maurizio, G. Masera, "VLSI Implementation of 16-point DCT for H.265/HEVC using Walsh Hadamard Transform and Lifting Scheme", 14th IEEE International Multitopic Conference (INMIC), 2011.
 - [39]. Y. Voronenko, M. Püschel, "Multiplierless Constant Multiple Multiplication", ACM Trans. on Algorithms, vol. 3, no. 2, May 2007.
 - [40]. Software/Hardware Generation of DSP Algorithms. [Online]. Available: <http://www.spiral.net/>