

Pereira, 15 de Febrero del 2010

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA, FÍSICA Y SISTEMAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

COMITÉ DE PROYECTOS

Cordial saludo,

Estoy enviando mi anteproyecto de grado titulado “NÚCLEO IP BLANDO DECODIFICADOR DE MP3” como requisito para mi proyecto de grado el cual quisiera pedirles el favor que valoraran. El proyecto tiene la viabilidad técnica de la ingeniería Ana María y se le han realizado las debidas correcciones.

A modo de aclaración dejo constancia que el título y el contenido de mi anteproyecto se cambio a idioma español, sin embargo se conservan las palabras en inglés que maneja el estándar ISO y que por motivo de estandarización del documento no fueron cambiadas. De igual manera los sitios web que referencio en el documento son confiables puesto que son sitios web oficiales de la empresas Thomson y del IIS, propietarios de la patente MP3 y de sus patentes asociadas.

Muchas gracias por la atención prestada.

Atentamente

JUAN PABLO GÓMEZ GALLEGU CODIGO 4.585.951

FICHA DEL PROYECTO

Título: NÚCLEO IP BLANDO DECODIFICADOR DE MP3

Facultad: Facultad de ingenierías: Eléctrica, electrónica, física y ciencias de la computación.

Estudiantes participantes en el proyecto:

Nombre	Identificación	Contacto	Programa académico
Juan Pablo Gómez Gallego	4'585,951 de Santa Rosa de Cabal	juanwalker@ gmail.com Tel : 3370572	Ingeniería de sistemas y computación

NÚCLEO IP BLANDO DECODIFICADOR DE MP3

JUAN PABLO GÓMEZ GALLEGO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

**FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS
DE LA COMPUTACIÓN**

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA

2010

NÚCLEO IP BLANDO DECODIFICADOR DE MP3

JUAN PABLO GÓMEZ GALLEGU

Anteproyecto de grado presentado como requisito parcial para realizar el proyecto de
grado

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS
DE LA COMPUTACIÓN

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA

2010

TABLA DE CONTENIDO

1	Resumen	1
2	Definición del Problema	2
3	JUSTIFICACIÓN	3
3.1	Técnica	3
3.2	Económica	4
4	OBJETIVOS	5
4.1	Objetivo General	5
4.2	Objetivos Específicos	5
4.2.1	Objetivos del producto	5
5	MARCO REFERENCIAL	6
5.1	Marco Histórico Del MP3	6
5.2	Marco Teórico	7
5.2.1	Psicoacústica	7
5.2.2	Enmascaramientos	7
5.2.3	Estándar PCM	9
5.2.4	Estándar Mpeg-1	10
5.2.5	Sobre la Compresión en Relación 1 a 12	11
5.2.6	Bit Rate	11
5.2.7	Frecuencia de Muestreo	12
5.2.8	Modos del Estándar	12
5.2.9	Archivo MP3	13
5.3	Frame MP3	13
5.3.1.1	Header	14
5.3.1.2	Side Information	15
5.3.1.3	Main Data	23
5.3.1.4	Decodificación	25
5.3.1.4.1	Decodificando la Parte Header	26
5.3.1.4.2	Decodificación del Side Information	26
5.3.1.4.3	Decodificación de los factores de escala	27
5.3.1.4.4	Decodificación de los Datos de Huffman	27
5.3.1.4.5	Recuantización del Espectro	29
5.3.1.4.6	Reordenamiento del Espectro	30

5.3.1.4.7	Procesamiento Stereo	31
5.3.1.4.8	Reducción de Alias	32
5.3.1.4.9	IMDCT (Inversa de la transformada modificada del coseno)	34
5.3.1.4.10	Filterbank Polifásico	35
5.4	SISTEMAS EMBEBIDOS	38
5.5	MARCO LEGAL	39
6	DISEÑO METODOLÓGICO	41
6.1	Definición de la Hipótesis	41
6.2	Metodología de la Investigación	41
6.2.1	Especificación de Requerimientos (Fase 1):	42
6.2.2	Diseño de Alto Nivel (Fase 2):	42
6.2.3	Diseño en Detalle (FASE 3):	42
6.2.4	Implementación (Fase 4):	42
6.2.5	Test Unitario (Fase 5):	43
6.2.6	Integración (Fase 6):	43
6.2.7	Test Operacional del Sistema (Fase 7):	43
6.3	Esquema Temático	45
7	RESULTADOS ESPERADOS	46
8	ESTRATEGIA A UTILIZAR PARA LA TRANSFERENCIA DE RESULTADOS	46
9	CRONOGRAMA	47
10	RECURSOS MATERIALES	49
11	Apéndice	50
11.1	Codificación de Huffman	50
11.2	Glosario	52
11.3	Glosario estándar ISO/IEC 11172-3	55
12	BIBLIOGRAFÍA	57
13	ANEXOS	58
13.1	Cotización de decodificadores de MP3 de núcleo blando	58

Cotización de decodificadores de MP3 de núcleo blando

LISTA DE ILUSTRACIONES

<u>Ilustración 5–1 Tono enmascarador de 1khz</u>	8
<u>Ilustración 5–2 Efectos del pre y post enmascaramiento</u>	9
<u>Ilustración 5–3 Frame MP3</u>	13
<u>Ilustración 5–4 Descripción del header de un frame MP3</u>	14
<u>Ilustración 5–5 Descripción del side information</u>	15
<u>Ilustración 5–6 Las 5 regiones</u>	17
<u>Ilustración 5–7 Composición de un gránulo</u>	23
<u>Ilustración 5–8 Descripción de la técnica bit reservoir</u>	25
<u>Ilustración 5–9 Diagrama de bloques del decodificador MP3</u>	25
<u>Ilustración 5–10 Decodificación de huffman</u>	28
<u>Ilustración 5–11 Proceso de recuantización</u>	30
<u>Ilustración 5–12 Reordenamiento del espectro</u>	31
<u>Ilustración 5–13 Procesamiento de las 8 mariposas</u>	33
<u>Ilustración 5–14 Operación de superposición</u>	34
<u>Ilustración 5–15 Operación de superposición</u>	35
<u>Ilustración 5–16 Ejemplo de organización de un sistema embebido</u>	38
<u>Ilustración 6–1 Metodología de desarrollo VEE</u>	42
<u>Ilustración 12–1 Ejemplo de construcción de un árbol de huffman</u>	51
Ilustración 11–1 Ejemplo de construcción de un árbol de huffman	

LISTA DE TABLAS

Tabla 3–1 Costos de un núcleo ip blando para decodificar MP3 en el mercado	2
Tabla 5–1 Bit rate por capa MPEG	10
Tabla 5–2 Clasificación de las bandas los factores de escala	16
Tabla 5–3 Compresiones de los factores de escala	19
Tabla 5–4 Tipos de bloque²⁰	20
Tabla 5–5 Valores predefinidos de pretab	21
Tabla 5–6 Razones de cambios de los factores de escala ²¹	21
Tabla 5–7 Descripción del side information para bloques largos	22
Tabla 5–8 Descripción del side information para bloques pequeños	23
Tabla 5–9 Coeficientes de las mariposas	33
Tabla 5–10 Países donde están registradas las patentes MP3	39
Tabla 5–11 Costos de licenciamiento de MP3	40
Tabla 9–1 Cronograma del proyecto	48
Tabla 11–1 Recursos del proyecto	49
Tabla 10–1 Recursos del proyecto	

1 RESUMEN

Palabras claves : MP3,MPEG, FPGA, NÚCLEO IP,VHDL¹

La compresión digital de datos de audio es importante debido a las limitaciones de ancho de banda en las redes de datos. El estándar MPEG-1 Layer III, también conocido como MP3, y descrito en el documento ISO/IEC 11172-3², es un códec de audio muy común para la compresión de música con calidad CD₁. Un decodificador de MP3 tiene una estructura compleja y es computacionalmente exigente. El objetivo de esta trabajo de grado es desarrollar un núcleo ip blando que permita decodificar MP3 e implementarlo en un kit de desarrollo de un dispositivo FPGA que cumpla el estándar ISO².

2 DEFINICIÓN DEL PROBLEMA

Hace falta documentación pública que permita la elaboración de un decodificador MP3 basado en hardware. Esto ocurre debido a que el estándar MP3 está protegido por patentes y la poca documentación existente se encuentra en otros idiomas diferentes al

¹ Ver definición en el glosario

² El estándar que describe la codificación de las animaciones y el audio. La parte 3 describe el audio

español, además que por política comercial, los fabricantes de hardware no comparten su conocimiento de ingeniería de manera pública.

A continuación se muestran los costos de algunos fabricantes para la adquisición de una licencia de uso de un decodificador mp3 de núcleo ip blando

Empresa	Valor	Descripción
Aptx.com	60.000 USD	Desarrollo en FPGA
Ingenient.com	75.000 USD	RTL del núcleo ip blando decodificador de MP3. Para Arquitectura ARM
FastVDO.com	50.000 USD	Bit Stream para FPGA

Tabla 2–1 Costos de un núcleo ip blando para decodificar MP3 en el mercado ³

La ausencia de documentación pública para el desarrollo de hardware MP3 restringe el desarrollo de la electrónica de consumo puesto que limita el soporte de audio MP3.

3 JUSTIFICACIÓN

3.1 Técnica

La sofisticación y la complejidad de los sistemas multimedia ha evolucionado al ritmo de la tecnología informática la cual lo ha hecho rápidamente desde mediados de la

³ Datos obtenidos a través de correos y llamadas telefónicas hechas por el autor. Ver anexo

década de 1980. El uso de técnicas de compresión de datos es un requerimiento técnico esencial para la creación de electrónica de consumo, puesto que no sólo permiten manejar una enorme cantidad de de datos sino también mantener un sonido de buena calidad. MPEG (Moving Pictures Experts Group) es un estándar de compresión de audio digital y animación, que puede de manera significativa reducir los requisitos de ancho de banda de transmisión y almacenamiento de datos con baja distorsión. El MP3 (Más conocido como MPEG-1 Audio Layer III) es una técnica avanzada para la compresión de audio digital que por lo general puede conseguir una relación de compresión de 1:12 lo cual significa que una canción de 5 minutos que ocupa más de 50 Mbytes ahora podría ser almacenada en 4 Mbytes, conservando características de alta compresión y de alta calidad. Por esta razón la tecnología MP3 se ha masificado ampliamente en el mundo y se encuentra soportada por fabricantes de electrónica de consumo y de gadgets

Beneficios:

El desarrollo de un núcleo ip blando para decodificar MP3 facilita la portabilidad⁴ a diferentes sistemas de hardware debido a que son rutinas de software que describen hardware reconfigurable, de esa manera las actualizaciones y optimizaciones que se realicen en el código podrán implementarse sin traumatismo en los entornos de producción.

En cuanto a sus posibles usos puede utilizarse en dispositivos móviles tales como celulares, notebooks ,y pdas permitiendo decodificar música sin necesidad de ocupar la CPU implicando un menor consumo de potencia y aumentando la autonomía de los dispositivo. Cualquier aplicación donde el consumo de potencia sea crítico es una buena candidata a tener el decodificador MP3 en hardware, o también aquellas que necesiten realizar decodificaciones en paralelo utilizando una baja potencia.

⁴ Referenciarse al glosario

3.2 Económica

Los núcleos ip blandos es muy apetecidos debido a ofrecen mayor flexibilidad que un núcleo ip en silicona y por lo general son más difíciles de realizar. En la tabla 3-1 se pueden ver los costos que algunos fabricantes tienen por cada licencia, referirse también al anexo para ver la fuente de cada consultas.

Beneficios:

Como es de notarse el costo para realizar la transferencia de tecnología es muy alto, y los proveedores de este tipo de núcleo ip blando son escasos, de realizarse un núcleo ip blando, estándar, para decodificar MP3 representaría un potencial activo de valor en el mercado.

4 OBJETIVOS

4.1 Objetivo General

Desarrollar un núcleo ip blando decodificador de audio MP3 en un dispositivo FPGA con propósito educativo y documental.

4.2 Objetivos Específicos

- Diseñar el decodificador MP3 en alto nivel y de manera detallada.

- Implementar el diseño del decodificador MP3 en la FPGA transcribiendo el código del software dist¹⁰ en lenguaje C a un lenguaje de descripción de hardware.
- Cuantificar los recursos utilizados por el decodificador en la FPGA (Bloques de lógica, memoria, tiempo de procesamiento), la potencia y las latencia.
- Comprobar el cumplimiento del indicador RMS con un valor menor a 8.81×10^{-6} ⁶
- Documentar el proyecto y publicar la información en un wiki internacionalizado.

1 MARCO REFERENCIAL

1.1 Marco Histórico Del MP3⁷

El estándar MPG tuvo sus inicio en el instituto Fraunhofer a cargo del profesor Karlheinz Brandenburg y de su equipo de desarrollo en los años 80, pasarían por algunos años de desarrollo hasta convertirse en un estándar internacional

- En 1987 el instituto Fraunhofer en Alemania comienza a desarrollar el proyecto con código clave EUREKA para broadcasting⁸ de audio digital (DAB)⁷ lo cual es un inicio para la creación del formato MP3. Con el aporte del Profesor Dieter Seitzer de la Universidad de Erlangen, se desarrolló un algoritmo que eventualmente se hizo conocido como el ISO-MPEG Audio Layer-3 standard.
- En enero de 1988 se crea MPEG estableciéndose como un subcomité de ISO/IEC el cual es dirigido por Leonardo Chiariglione y Hiroshi Yoshuda director del JPEG.

⁵ Código fuente anexo al estándar ISO/IEC Coding of Moving Pictures And Associated Audio

⁶ ISO/IEC Coding of Moving Pictures And Associated Audio part 4

⁷ Chad Fogg MPEG, Video Compression Standard, Springer, 1 Edition p 383-391, 1996

⁸ Ver definición en el glosario

- En 1989 Fraunhofer recibe la patente alemana por la creación del MP3.
- En 1992 el códec MP3 se integra dentro del estándar MPEG-I
- En 1993 el estándar es publicado.
- En noviembre de 1996 la patente de MP3 es aceptada dentro de los Estados Unidos.
- En septiembre del 2008 Fraunhofer decide empezar a cobrar royalties por hacer uso de la patente de MP3 todos los desarrolladores de deben pagar tarifas a Fraunhofer.
- En 1999 aparecen los primeros reproductores de MP3

1.1 Marco Teórico

1.1.1 Psicoacústica

Los seres humanos están expuestos a una cantidad extrema de radiación, las ondas se encuentran un espectro de frecuencia que comprende muchísimas frecuencias diferentes. Los órganos son sensibles a una pequeña fracción de estas ondas; La luz visible y los sonidos audibles, en tanto que hay otras ondas de luz como las infra-rojas y las ultravioleta que no son perceptibles por los órganos de los seres humanos. El oído de la mayoría de seres humanos no pueden escuchar frecuencias menores a 20Hz ni mayores a 20kHz⁹, este espectro de frecuencias tiende a disminuirse a medida que el ser humano envejece.

El cerebro humano captura los datos recogidos en el medio por sus 5 sentidos en un tiempo determinado sin embargo el cerebro humano no es capaz de procesar toda esta información por lo tanto debe aplicar filtros mentales. Cuando es reproducido un CD no es posible percibir todo el audio que llega a los oídos así que no hay necesidad de guardar toda la información no audible. El proceso para hacer que ciertas muestras de sonido no sean audibles se llama enmascaramiento. Hay dos tipos de enmascaramiento que el códec de audio necesita manejar el **enmascaramiento simultáneo** y el **enmascaramiento temporal**. Se ha demostrado que el oído humano tiene 24 bandas de frecuencia llamadas bandas críticas. Las frecuencias localizadas dentro de estas denominadas bandas críticas son más difíciles de distinguir por el oído humano.

1.1.2 Enmascaramientos

Suponiendo un caso donde existe un tono dominante 1kHz, 60db presente en una señal de audio. El tono dominante creará un umbral de enmascaramiento que

⁹ Saposhkov M.A Electroacústica, Editorial reverté 1era edición, 1993 p 19

envolverá y no hará audibles las frecuencias en la misma banda. Este fenómeno es conocido como **enmascaramiento simultáneo**.

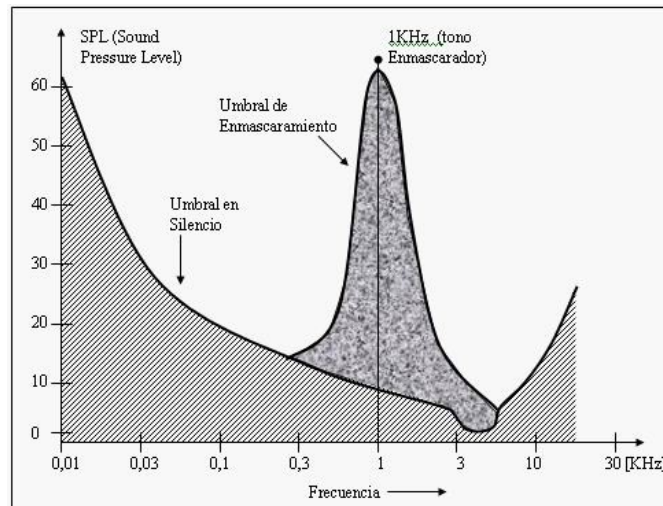


Ilustración 5–1 Tono enmascarador de 1khz

10

Para ilustrar la cuantificación del enmascaramiento simultáneo se debe realizar el siguiente experimento, con una persona aislada de sonidos se debe producir un tono de 1 KHz (enmascarador) a un volumen determinado (ilustración 5-1). Y generar otro sonido de prueba a 1,1 KHz y elevar su volumen hasta que sea justo perceptible e ir variando la frecuencia del tono de prueba.

El enmascaramiento temporal ocurre en el dominio del tiempo. Un tono dominante (enmascarador) envuelve uno más débil (enmascarado) si aparece dentro un pequeño intervalo de tiempo. El umbral de enmascaramiento envolverá las señales débiles previamente y posteriormente a la aparición del enmascarador. La etapa previa tomará 50 ms mientras que la etapa posterior tomará entre 50ms 300 ms dependiendo de la fuerza y duración del enmascarador. (Ilustración 5-2)

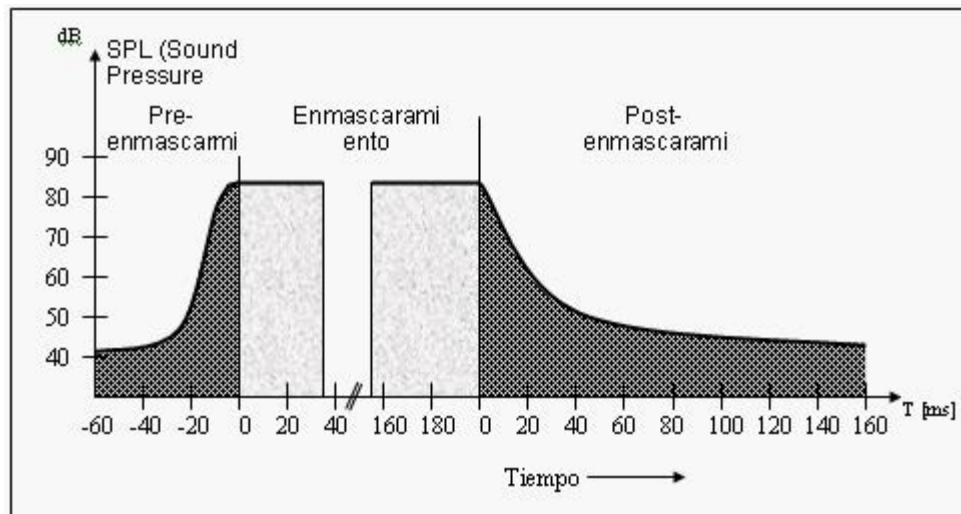


Ilustración 5–2 Efectos del pre y post enmascaramiento

11

Para la determinación de un enmascaramiento temporal se puede realizar el experimento en las mismas condiciones que el anterior, ofrecer un tono enmascarador de 1 KHz a 60 dB y un tono de prueba de 1,1 KHz a 40 dB. El tono de prueba no es audible. Parar el tono enmascarador y, con un pequeño retraso, el de prueba. Aumentar el retraso hasta que se distinga el tono de prueba.

Parte de la codificación del audio en MP3 consiste en calcular el umbral de enmascaramiento y de esta manera filtrar las señales que no serán escuchadas.

1.1.3 Estándar PCM

La modulación por impulsos de codificación es un formato estándar para almacenar o transmitir audio digital sin compresión. Un CD es un ejemplo de medio de

almacenamiento que se adapta el formato PCM. Hay dos variables que son significativas para PCM, la frecuencia de muestreo [Hz] y bit rate [Bit]¹². La frecuencia de muestreo se describe cómo la cantidad de muestras por segundo de grabación. Una frecuencia de muestreo significa una mayor cantidad de frecuencias incluidas. El bit rate representa el tamaño de la palabra digital. Un bit rate alto ofrece una mejor resolución de audio y menor cantidad debido a que la muestras podrán ser determinadas más exactamente. La calidad CD utiliza 44.100 Hz y 16 bits.

1.1.4 Estándar Mpeg-1

La organización internacional para la estandarización (ISO) es una federación que tiene como objetivo facilitar el intercambio de bienes, servicios y conocimiento. MPEG (Moving Picture Experts Group) trabaja junto a ISO desarrollando la codificación y decodificación de audio y video. El desarrollo empezó en 1988 y fue terminado en 1992 produciendo así el estándar

MPEG-1 el cual estaba dividido en tres partes :

1. Primera parte : Sistemas.
2. Segunda parte: Video.
3. Tercera parte : Audio.

El estándar describe como transmitir múltiples señales de audio y video en un solo canal de distribución a una velocidad de hasta 1,5Mbit/s. Para la parte de audio MPEG-1 maneja tres capas de compresión: Layer I, Layer II y Layer III. Siendo la capa tres la de mayor complejidad y menor consumo de ancho de banda sin pérdida de calidad de audio (Ver tabla 5-1)

Codificación	Relación	Bit rate requerido
PCM calidad CD	1:01	1.4 Mbps
Layer I	4:01	384 kbps
Layer II	8:01	192 kbps
Layer III (MP3)	12:01	128 kbps

¹² Ver definición en el glosario

Tabla 5–1 Bit rate por capa MPEG¹³

MPEG-1 Layer III comprime el PCM en una relación de 1 a 12 sin pérdida de calidad apreciable siendo así la capa más compleja pero también la más eficiente. El MPEG-1 Layer III es más conocido como MP3.

1.1.1 Sobre la Compresión en Relación 1 a 12

Debido a que MP3 es un códec que está basado en la capacidad de percepción humana para filtrar la información que no es necesaria, la información descartada en la codificación no puede ser recuperable. Cada banda crítica¹⁴ del oído humano se aproxima a una banda de factor de escala¹⁵. Para cada banda de factor de escala se calcula un umbral de enmascaramiento. Dependiendo del umbral las bandas son escaladas utilizando un factor de escala adecuado, que modifica la amplitud en la muestras de audio, de esta manera reduciendo el ruido de cuantización. Finalmente para hacer la compresión con pérdida el estándar aplica la codificación de Huffman¹⁶.

1.1.2 Bit Rate

El bit rate es una opción de codificación que se debe establecer antes de la codificación. Este representa la cantidad de datos permitidos al codificador para guardar durante cada segundo brindando al usuario la posibilidad de escoger la calidad de los datos codificados. El estándar MP3 permite definir bit rates desde 8kbit/s hasta 320 kbit/s, por defecto se usa 128kbit/s. Un mayor bit rate implica que las muestras se guardaran más precisamente ofreciendo una mejor resolución de audio.

El estándar maneja dos tipos de bit rate: CBR o (Constant Bit rate) y VBR o (Variable Bit rate). Cuando el codificador utiliza CBR cada parte de la canción se guarda utilizando la misma cantidad de bits. Pero debido a que las canciones pueden variar en complejidad, hay momentos de la canciones que tienen más efectos e instrumentos que en otras, por resolución de sonido el bit rate uniforme guarda con la misma cantidad de espacio las partes de la canción que necesita más bits que otra más sencillas que no los necesitan. El bit rate tipo VBR es la solución a este problema permitiendo variar el bit rate según las dinámicas del audio, el usuario dispone cual es el mayor bit rate

13 Raissi Rassol, The theory behind MP3 www.mp3-tech.org/programmer/docs/mp3_theory.pdf, Tabla 4.1

14 Ver definición en el glosario

15 Ver anexo factores de escala

16 Ver en el anexo codificación de huffman

permitido a utilizar por el codificador, sin embargo VBR puede presentar problemas para algunos decodificadores por problemas de sincronización. Por lo tanto para broadcasting es más utilizado el CBR.

1.1.3 Frecuencia de Muestreo

La resolución de audio es dependiente de la frecuencia de muestreo la cual se define como las veces por segundo que una señal es muestreada. Una mayor cantidad de muestras almacenadas mejorará la precisión. El estándar MPEG-1 trabaja con diferentes frecuencias de muestreo: 32kHz, 44.1 kHz y 48 kHz.

1.1.4 Modos del Estándar

Hay 4 tipos de funcionamiento definidos:

1. **Modo Monofónico:** Utiliza un solo canal para codificar todos los datos del audio
2. **Modo Monofónico doble,** Utiliza dos canales de mono independientes de manera forzada, la información codificada en el canal izquierdo y el derecho es la misma. Funcionalmente es idéntico al modo stereo.
3. **Modo Stereo,** Hace referencia al sonido estereofónico, la información se codifica en el canal izquierda y derecho, la señal de audio de un canal está relacionada con la señal del otro canal por lo tanto son dependientes uno del otro para decodificar el audio correctamente.
4. **Modo Joint Stereo :** Este modo considera la correlación y la redundancia entre el canal izquierdo y el derecho; codificándolos en uno solo y pudiéndose obtener una mejor compresión del audio, se utilizadas dos técnicas : **MS Stereo** y **Intensity Stereo**.

La técnica **MS Stereo** es útil cuando los canales están altamente correlacionados. Los canales izquierdos y derechos son transmitidos como la suma y la diferencia de los dos canales respectivamente. Los dos canales son similares la mayor parte del tiempo por lo tanto la señal de suma contiene más información que la señal de diferencia esto permite una mayor eficiencia de compresión en comparación la transmisión de los dos canales de forma independiente.

La técnica **Intensity Stereo** aprovecha que el oído humano es menos agudo para percibir la dirección de ciertas frecuencias de audio. Explotando esta limitación, este tipo de codificación puede reducir la cantidad de datos de audio con una pérdida en la calidad pequeña o nula.

1.1.1 Archivo MP3

Todos los archivos MP3 se encuentran divididos por frames quienes guardan 1152 muestras de audio que representan 26 ms. El bit rate determina el tamaño de cada frame de manera proporcional. El tamaño también depende de la frecuencia de muestreo y se encuentra definido por la ecuación (1)

$$144 \times \text{bitrate} \times \text{frecuencia de muestreo} + \text{padding bytes} \quad (1)$$

El slot adicional se sumará siempre y cuando el bit de padding esté activo.

1.2 Frame MP3

Cada frame comienza con un byte de sincronización y termina luego del siguiente byte de sincronización. El frame está dividido en 5 partes : Header o cabecera, CRC o comprobación de errores, side information, main data o datos del frame y ancillary o información de etiquetado (Ilustración 5–3)

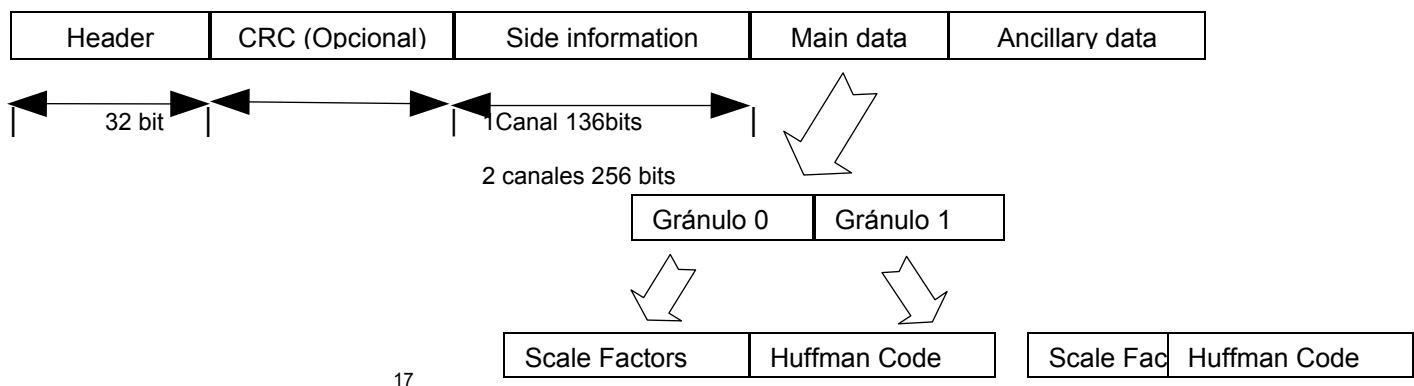


Ilustración 5–3 Frame MP3

1.2.1.1 Header

La cabecera del frame tiene un tamaño de 32 bits(4 bytes). Contiene información sobre el frame. Los primeros doce bits pertenecen a **syncword** y serán siempre 1 y son denominados de sincronización y se utiliza para encontrar el comienzo de cada frame.

17 Sripada praveen, MP3 decoder in theory and practice, Masters Thesis, 2006, p 13

El campo **ID** describe la versión de MPEG. El campo **Layer** describe la capa MPEG utilizada. **Protection Bit** indica si el bitstream está protegido por CRC seguido del header. El campo **Bitrate Index** indica el bit rate. Puede ir desde 32 kbit/s hasta 320 kbit/s. **Sampling frequency** describe la frecuencia de muestreo utilizada soportando 32 kHz, 44.1 kHz y 48 kHz. **Padding bit** representa si los datos están enmarcados con un slot adicional. **Private bit** se utiliza para propósito informativo. Los bit **Mode** seleccionan el modo de reproducción (Monofónico, monofónico doble, stereo o joint-stereo). **Modo extensión** se utiliza para especificar la técnica joint-stereo-mode escogida. **Copyright** indica si el bitstream está amparado por derechos de autor. **Original**, indica si el bitstream es una copia o no. **Emphasis** indica el tipo de énfasis utilizada por ejemplo re-equalizar el sonido después de un dolby. (Ilustración 5–4)

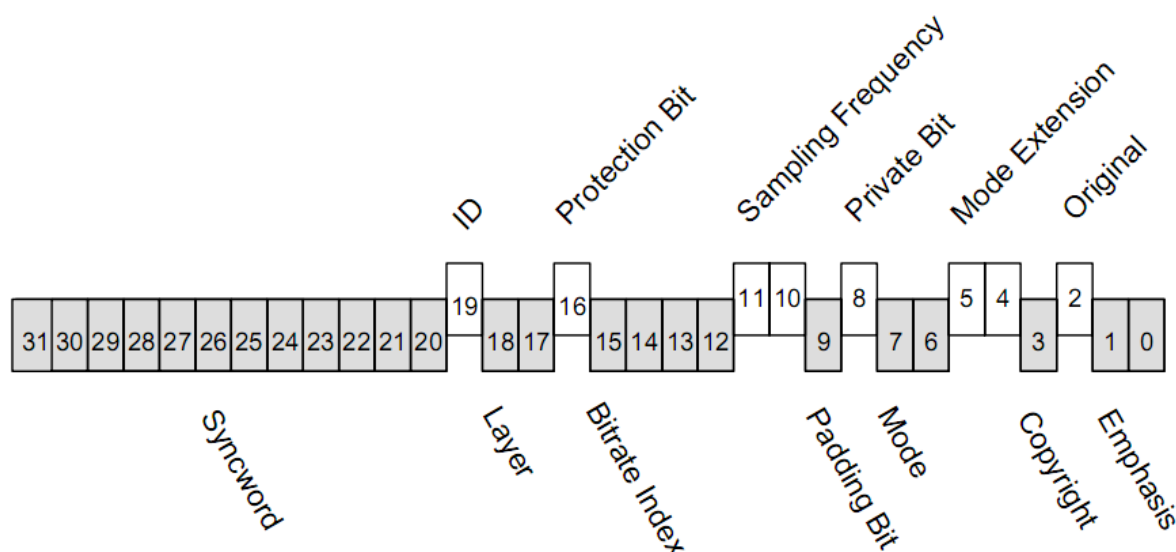


Ilustración 5–4 Descripción del header de un frame MP3¹⁸

1.2.1.2 Side Information

La parte del side information comprende de información necesaria para decodificar la parte main data. Para el modo tipo monofónico tiene un tamaño de 17 bytes, si es tipo dual o stereo su tamaño es de 32 bytes. Toda la información contenida aquí es global durante el proceso de decodificación, la decodificación de huffman requiere la información necesaria para saber que tablas de Huffman se deben utilizar y la información sobre la descompresión de los factores de escala.

¹⁸ Hsiao- Lun Liao Platform-based Hardware/Software CoDesign From Mp3 Decoder Master Thesis, Figure 2-2

La parte Side information también almacena información para localizar la parte main data que utiliza la técnica bit reservoir que será descrita más adelante.

La estructura básica de la parte side information se muestra en la ilustración 5–5

Main_data_begin	private_bit	scfsi	Side_info granule	Side_info granule
-----------------	-------------	-------	-------------------	-------------------

r anexo factores de escalaribe el audioignal
storage med el audio correctamente. Ilustración
5–5 Descripción del side information

19

main_data_begin (9 bits ambos canales)

Este valor es un puntero que apunta a la dirección donde comienza la parte de main data, en MP3 se utiliza una técnica llamada Bit reservoir que permite alojar espacio de manera óptima aprovechando el espacio libre entre dos frame consecutivos.

Private_bits (5 bits un solo canal, 3 bits dos canales)

Se utilizan para uso privado, no serán incluidos para futuras versiones ISO.

Scfsi [Scfi_band] (4 bits un solo canal, 8 bits dos canales)

Especifica si los factores de escala se reutilizan para el gránulo 0 y el gránulo 1. Las bandas de los factores de escala se dividen en grupos de 4. Si tiene un valor de '0' los factores de escala se transmiten para ambos gránulos, si tiene un valor '1' los factores de escala contenidos en el primer gránulo son válidos para el segundo gránulo. En la tabla 5–2 aparece las diferentes bandas de los factores de escala

Scfi_band d	Banda de factor de escala
0	0 – 5
1	6 – 10
2	11 – 15
3	16 – 20

Tabla 5–2 Clasificación de las bandas los factores de escala ¹⁸

Part2_3_lenght[gr] (12 bits un solo canal, 24 bits dos canales)

La cantidad de bits en la parte main_data utilizados por los scale factor y la codificación huffman. Es la suma de part2_lenght (Tamaño de los factores de escala) y part3_lenght el tamaño de los datos generadas en la tabla de huffman

Big_values[gr] (9 bits en un solo canal, 18 bits dos canales)

Las 576 muestras almacenadas en cada gránulo no se codifican con una sola tabla de codificación de huffman. Las frecuencias van desde 0 hasta la frecuencia de Nyquist y se dividen en cinco regiones, la justificación para hacer estas divisiones es utilizar cinco diferentes tablas que permitan mejorar el desempeño de la codificación y una mejor robustez a los errores. En la ilustración 5-6 se describe las divisiones de Huffman.

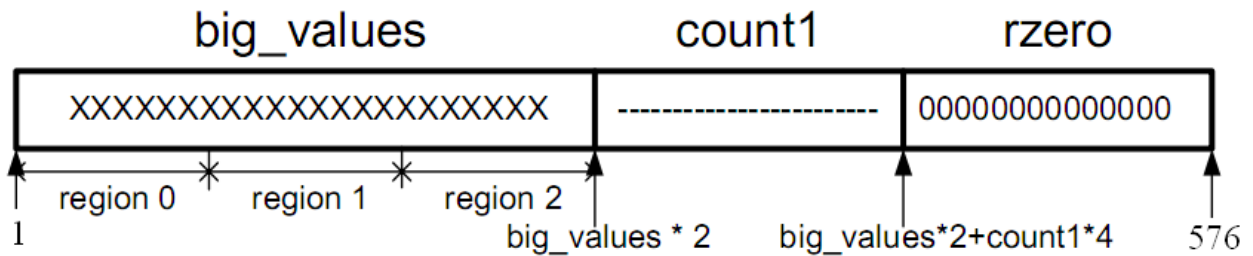


Ilustración 5–6 Las 5 regiones²⁰

- Los resultados van desde la muestra 1 a la muestra 576, cada región está codificada por una tabla de huffman
- Los valores “000” son todos ceros
- Los valores “---” pueden ser -1,0,1
- Los valores “xxx” puede ir desde -8206 hasta 8206, se codifican como duplas

Esta es la descripción de cada una de las partes :

Rzero : Se asume que las frecuencias más altas tiene menores amplitudes y por lo tanto no necesitan ser codificadas por lo tanto la cuantificación de estas frecuencias siempre serán cero. Se codifican como duplas

Count1: Son cuádruplas de valores cuantificadas.

Big values: Se codifican como duplas, los valores normalmente irían desde -15 hasta 15 sin embargo si se necesita de una buena precisión se utiliza el valor 15 o código de ESCAPE que permitirá leer otro número llamado linbits que indicará de 1

²⁰ Hsiao- Lun Liao Platform-based Hardware/Software CoDesign From Mp3 Decoder Master Thesis, Figure 2-5

a 13 cuantos bits deberán leerse y sumarse. Esta región está dividida en tres subregiones

Global_gain[gr] (8bits)

Especifica el tamaño del proceso de cuantización, esto es necesario para el bloque de recuantificación del decodificador.

Scalefac_compress[gr] (4 bits en un solo canal, 8 bits dos canales)

Define el número de bits utilizado para la transmisión de los factores de escala. Un gránulo de 576 muestras puede estar dividido en 12, 17 o 21 bandas de los factores de escala. Un bloque largo ($\text{block_type}=\{0,1,3\}$) particiona el gránulo en 21 bandas, un bloque pequeño ($\text{block_type}=2$) particiona el gránulo en 12 bandas y un bloque mixto particiona el gránulo en 17 bandas. La longitud de los los factores de escala se calcula en dos valores Slen1 y Slen2 .

Para bloques largos ($\text{block_type} = 0,1,3$)

Slen1 : Tamaño de los factores de escala para las bandas de 0 a 10

Slen2 : Tamaño de los factores de escala para las bandas de 11 a 20

Para bloques pequeños ($\text{block_type} = 2$ y $\text{mixed_block_flag} = 0$)

Slen1 : Tamaño de los factores de escala para las bandas de 0 a 6

Slen2 : Tamaño de los factores de escala para las bandas de 6 a 11

Para bloques mixtos ($\text{block_type} = 2$ y $\text{mixed_block_flag} = 1$)

Slen1 : Tamaño de los factores de escala para las bandas de 0 a 7 (Bloques largos) y de 3 a 5 (Bloques pequeños).

Slen2 : Tamaño de los factores de escala para las bandas de 6 a 11 (Bloques pequeños).

Scalefac_compress	Slen1	Slen2
0	0	0
1	0	1
2	0	2
3	0	3
4	3	0
5	1	1
6	1	2
7	1	3
8	2	1
9	2	2
10	2	3
11	3	1
12	3	2
13	3	3
14	4	2
15	4	3

Tabla 5–3 Compresiones de los factores de escala ²¹

windows_switching[gr] (1 bit un solo canal, 2 bits dos canales)

Indica que otro bloque diferente al normal está siendo utilizado. Block_type, mixed_block_flag y subblock_gain están activas.

Block_type[gr] (2 bits un solo canal, 4 bits dos canales)

Este campo indica que tipo de bloque se va a utilizar para un gránulo específico. Vale 00 cuando el tipo de bloque es normal, a continuación en la tabla 5–4 se describen los tipos de boques.

Block_type	Window_type
00	Reserved

01	Start	E s p e c i f i c a l i z a d o
10	3 short windows	
11	End	

Tabla 5–4 Tipos de bloque ²⁰

Mixed_block_flag (1 bits, 2bits)

Indica que frecuencias diferentes son transformadas con diferentes tipos de bloques y solo se utiliza cuando window_switching_flag está activo.

Table_select (10 bits, 20 bits)

Hay 32 tipos de tablas de huffman disponibles para la decodificación. Este valor de 5 bits sirve para codificar 32 valores posibles para cada región. Este valor solamente especifica las tablas para decodificar la partición big_values.

subblock_gain (9 bits, 18 bits)

Esta propiedad solo se utiliza cuando windows_switching está activa y block_type=2. Es una variable de 3 bits que indica la offset de la ganancia desde global_gain para cada bloque pequeño

region0_count (4 bits, 8 bits), **region1_count** (3 bits, 6 bits)

region0_count y region1_count indican los límites de las subregiones de bigvalues. Estos límites están alineados con el particionamiento del espectro. region0_count y

region1_count valen el número de las bandas de los factores de escala para su respectivas subregiones menos 1.

Preflag(1 bit, 2 bits)

Este campo nunca se utiliza para bloques cortos (blocktype=2). Si está activo, los valores se adicionan a los factores de escala y funciona como un acceso directo para las frecuencias altas, en la tabla 5–5 aparecen los valores de pre-tab definidos por el standard

Scalefac_scales	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	18	18
Pretab	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	2	3	3	3	2

Tabla 5–5 Valores predefinidos de pretab²²

Scalefac_scale

Los factores de escala están cuantificados logarítmicamente a una razón de cambio de 2 o 2 de acuerdo a la tabla 5–6:

Scalefac_scale	Step_size
0	2
1	2

Tabla 5–6 Razones de cambios de los factores de escala²¹

Count1table_select (1 bit, 2bits)

²² Ibidem p 30

Dos posibles tablas de huffman están disponibles para la región count1. Este campo especifica que tabla utilizar. La estructura del side information para bloques 0,1 y 3 se describe en la tabla 5–7

Nombre	Un solo canal	Dos canales
Main_data_begin	9	9
Private_bits	5	3
Share	4	4+4
Información para el primer gránulo:		
Part2_3_lenght	12	12 +12
Big_values	9	9 + 9
Global_gain	8	8 + 8
Scalefac_compress	4	4 + 4
Window_switching	1	1 + 1
Para bloques normales:		
Table_select	3*5	3*5+3*5
Region0_count	4	4 + 4
Region1_count	3	3 + 3
Sub total para el primer gránulo	22	44
Preflag	1	2
Scalefac_scale	1	2
Count1table_select	1	2
Subtotal para el primer gránulo	59	118
Subtotal para el segundo gránulo	59	118
Número total de bits	136	256
Número total de bytes	17	32

Tabla 5–7 Descripción del side information para bloques largos²³

Para bloques tipo 2 se describe la estructura del side information en la tabla 5–8:

Nombre	Un solo canal	Dos canales

²³ Sripada Praveen Mp3 decoder in theory and practice, Masters thesis report, 2006 Table 6

<i>Block_type</i>	2	2.+2
<i>Mixed_block_flag</i>	1	1+1
<i>Table selection for two regions</i>	2*5	2*5+2*5
	2*5	3*3+3*3
<i>Subblock_gain</i>		
Subtotal for not normal blocks	22	44

Tabla 5–8 Descripción del side information para bloques especiales²⁴

1.1.1.1 Main Data

Esta parte se encuentra dividida en dos gránulos. Cada gránulo contiene los factores de escala , los datos codificados por el algoritmo de huffman y los datos ancillary. La parte Main Data se describe en la ilustración 5-7.

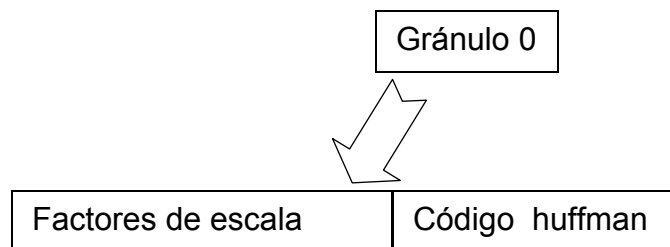


Ilustración 5–7 Composición de un gránulo 25

Factor de escala

El propósito de los factores de escala es reducir el ruido de cuantización. Si las muestras de una banda de factores de escala en particular son escaladas de la manera

²⁴ Ibidem table 7

²⁵ Sripada praveen, MP3 decoder in theory and prectice, Masters Thesis, 2006, p 13

correcta el ruido de cuantización será completamente enmascarado, un factor de escala es transmitido para cada banda.

Código de huffman

La información que describe como decodificar este código se encuentra en el side information, las tres subregiones de big_values se codifican como duplas y se disponen 32 tablas en tanto que la subregión count_1 como cuádruplas y dispone de la tabla A o B. La región rzero no se decodifica porque todos sus valores son ceros.

Técnica bit reservoir

El campo `main_data_begin` es un puntero de 9bits localizado en la parte side information que apunta al comienzo de la parte `main_data`. Este puntero especifica una posición negativa desde la parte header. En otras palabras el codificador solo puede pedir espacio prestado o donado por frames anteriores, no se puede utilizar espacio libres de los siguientes frames.

De manera ilustrativa analicemos basándonos en la ilustración 5-8: Para el Frame 1 `main_data_begin` vale 0 por lo tanto `main_data` sigue inmediatamente después de side information sin embargo queda espacio libre en el Frame 1. Ahora el `main_data` del frame2 ocupará este espacio libre y empezará antes que su header por lo tanto `main_data_begin 2` tendrá un valor negativo.

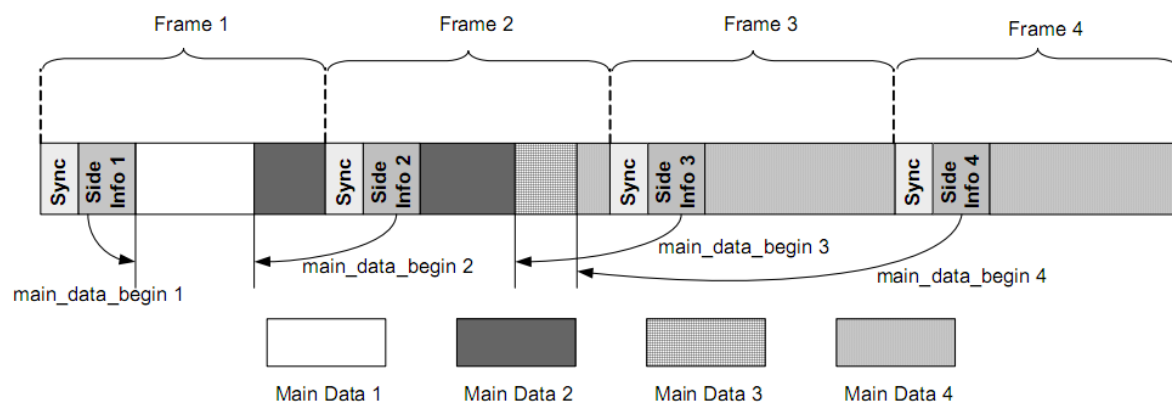


Ilustración 5–8 Descripción de la técnica bit reservoir²⁶

26 Hsiao- Lun Liao Platform-based Hardware/Software CoDesign From Mp3 Decoder Master Thesis, Figure 2-3

1.1.1.2 Decodificación

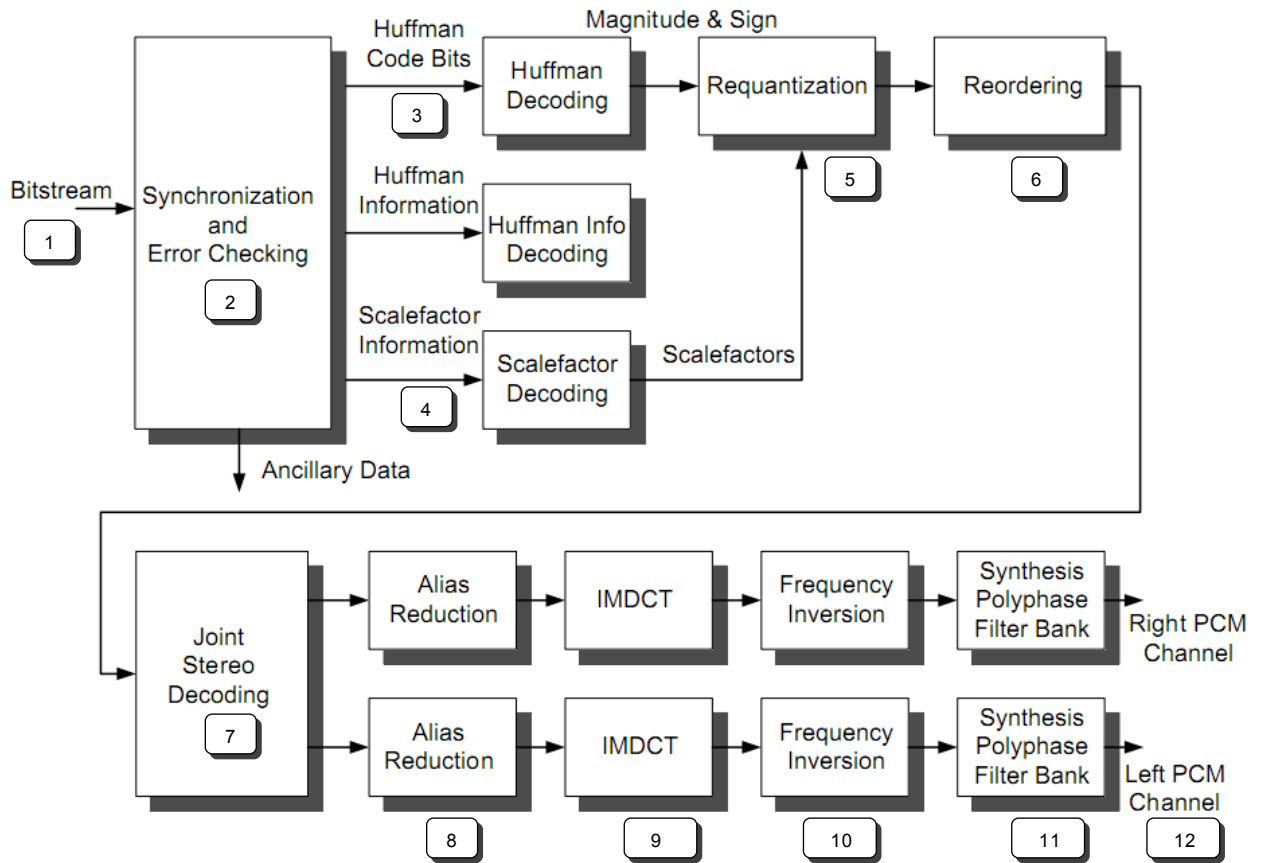


Ilustración 5–9 Diagrama de bloques del decodificador MP3²⁷

1. Se lee el bitsream y se busca el header
2. Se decodifica la parte side information
3. Se decodifica los datos de huffman
4. Se decodifican los factores de escala
5. Se recuantiza el espectro
6. Se reordena el espectro (Si Window_switching=1 y block_type=2)
7. Se realiza el procesamiento JOINT STEREO de ser necesario
8. Se realiza la reducción Alias
9. Se realiza la sintetización utilizando la IMDCT (Inversa modificado del coseno)

²⁷ Ibídem Figure 2-4

10. Se realiza la inversión de la frecuencia
11. Se realiza la sintetización por medio del FILTER BANK polifásico
12. Se generan las muestras PCM.

1.1.1.1.1 Decodificando la Parte Header

Al recibir un nuevo frame se debe localizar el header, los primeros 12 bits son leídos para la sincronización los cuales deben valer '1', aquí disponemos de la información necesaria para calcular el tamaño del frame lo cual es el comienzo del siguiente frame utilizando los campos del bit rate y de la frecuencia de muestreo. Se lee el resto de información contenida en el header necesaria para decodificar el frame. El header puede estar protegido con un dato de 15 bit CRC. Esta protección es opcional por lo tanto puede ser omitida. Cuando está presente el decodificador debe calcular el CRC y compararlo con el del frame si estos no concuerdan el decodificador debe empezar a buscar otra palabra de sincronización.

1.1.1.1.2 Decodificación del Side Information

Se leerán 17 bytes para el modo monofónico o si es tipo dual o stereo se leerán 32 bytes. Aquí encontraremos toda la información necesaria para descomprimir los factores de escala y realizar la decodificación de huffman.

1.1.1.1.3 Decodificación de los factores de escala

Los factores de escala pueden ser o no compartidos, si scfsi vale 0 los factores de escala se transmiten de manera independiente a ambos gránulos. Sin embargo si scfsi vale 1 los factores de escala son utilizados en ambos gránulos.

El número de bits utilizados para codificar los factores de escala se llama part2_lenght y se calculan con las fórmulas 2,3 y 4 dependiendo los tipos de los bloques utilizados en los gránulos.

Para bloques largos Si block_type = 0,1,3 entonces

Slen1 cuantifica 11 bandas. Slen2 cuantifica 10 bandas

$$part2length=11 \times slen1 + 10 \times slen2 \quad (2)$$

Para bloques pequeños Si $block_type = 2$ y $mixed_block_flag = 0$ entonces

Slen1 cuantifica 6 bandas de bloques pequeños. Slen2 cuantifica 6 bandas de bloques pequeños.

Recordemos que cada bloque pequeño contiene 3 bloques superpuestos.

$$part2length=3 \times 6 \times slen1 + 3 \times 6 \times slen2 = 18 \times slen1 + 18 \times slen2 \quad (3)$$

Para bloques mix Si $block_type = 2$ y $mixed_block_flag = 1$ entonces

Slen1 cuantifica 8 bandas de bloques largos y 3 de bloques pequeños.

Slen2 cuantifica 6 bandas de bloques pequeños

Recordemos que cada bloque pequeño contiene 3 bloques.

$$part2length=8 + 3 \times 3 \times slen1 + 6 \times 3 \times slen2 = 17 \times slen1 + 18 \times slen2 \quad (4)$$

1.1.1.1.4 Decodificación de los Datos de Huffman

Se decodifican las tres subregiones pertenecientes a la sección big_values . La decodificación se realiza utilizando las tablas de huffman, las tablas utilizada para la decodificación se guardará en $table_select$. Luego que se hallan decodificado la subregión big_values se siguen con la región $count1$ por el valor $count1table_select$. El fin de la decodificación se da cuanto las 576 muestras se decodifican si hay más bits adicionales a estos 576 muestras se descartan. En la figura 5-10 se describe la decodificación de huffman. Hay 2 métodos para las búsquedas en las tablas de huffman:

Búsqueda de árbol binario: Las tablas de decodificación pueden ser traducidas en árboles binarios cada árbol representa una tabla. Un '0' significa izquierda y un '1' derecha. Una palabra se decodifica totalmente cuando se encuentra una hoja. Las hojas contiene los valores de las líneas espectrales. Este método ahorra espacio sin embargo lleva más tiempo en realizarse.

Búsqueda por tabla: Este método utilizar tablas largas. El tamaño de cada tablas es de 2b donde el decodificador lee 'b' bits. Estos bits se utilizan como índices en la tabla

donde cada entrada contiene el valor de una línea espectral y la información del tamaño real de una palabra de código. Este método ocupad más espacio sin embargo permite para realizar las búsquedas más rápidamente.

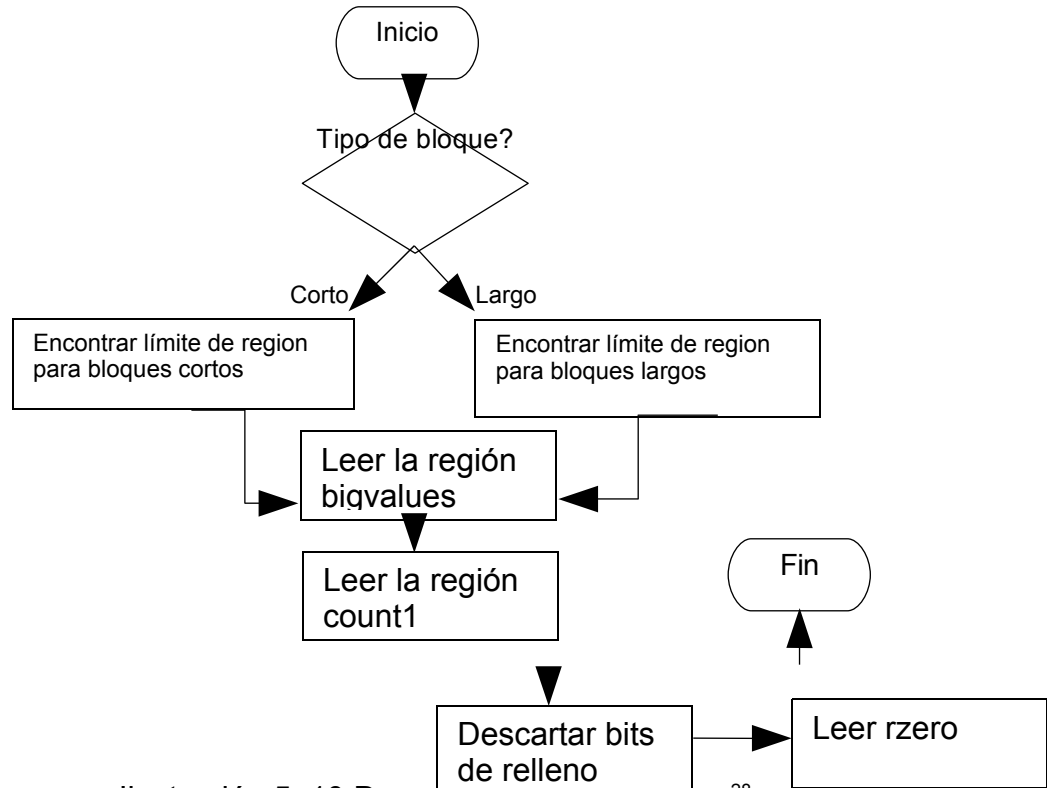


Ilustración 5-10 Decodificación de Huffman²⁸

1.1.1.1.5 Recuantización del Espectro

La cuantización es el proceso de convertir un número real de infinita precisión en un número entero. Esto es lo que se realiza en la codificación, en la decodificación se realiza lo contrario para obtener la líneas de frecuencias. De la decodificación de Huffman se obtiene los 576 enteros correspondientes a las líneas de frecuencias los cuales se recuantizan. La recuantización se realiza de manera separada para los bloques cortos y largos utilizando las leyes de energía descritas en las fórmulas 5 y 8. El escalamiento que se realiza posteriormente se realiza multiplicando los valores por los correspondientes factores de escala los productos se guardan como frecuencias escaladas. La ilustración 5-11 describe este proceso

²⁸ Hsiao- Lun Liao Platform-based Hardware/Software CoDesign From Mp3 Decoder Master Thesis, Figure 2-6

En las ecuaciones a continuación los valores de salida de la decodificación de huffman son denotados como $is[i]$ y las salidas de la recuantización de denotan como $xr[i]$. Los demás parámetros tales como $global_gain$, $subblock_gain$, $scalefac_multiplier$ y $preflag$ se encuentran disponibles en la parte de side information. $Scalefac_l$ y $scalefac_s$ corresponden a los factores de escala para los bloques largos y pequeños respectivamente que se encuentran en la parte main data. $Pretab$ es una tabla predefinida por el estándar ISO/IEC 11172-3.

Las líneas de frecuencias para los bloques largos y cortos están divididos en 23 y 13 bandas de los factores de escala respectivamente. Cada banda tiene su propio factores de escala .

Para bloques cortos para $block_type = 2$

$$xri = sign(is_i) \times is_i^{43 \times 2A \times 2B} \quad (5)$$

$$A = 14 \times global_gain[gr] - 210 - 8 \times subblock_gain[window][gr] \quad (6)$$

$$B = -scalefac_multiplier \times scalefac_s[gr][ch][sfb][window] \quad (7)$$

Para bloques largos para $block_type = \{0,1,3\}$

$$xri = sign(is_i) \times is_i^{43 \times 2C \times 2D} \quad (8)$$

$$C = 14 \times global_gain[gr] - 210 \quad (9)$$

$$D = -scalefac_multiplier \times scalefac_lsfbchgrwindow + preflag[gr] \times pretab[sfb] \quad (10)$$

La fórmula utilizada para la recuantización depende del tipo de bloque, cuando la diferencia de tiempo entre el frame actual y el frame anterior es muy poca se utiliza un bloque largo. Alternativamente si la señal de la subbanda muestra considerable diferencias entre los tiempos de los frames, entonces un bloque pequeño es utilizado.

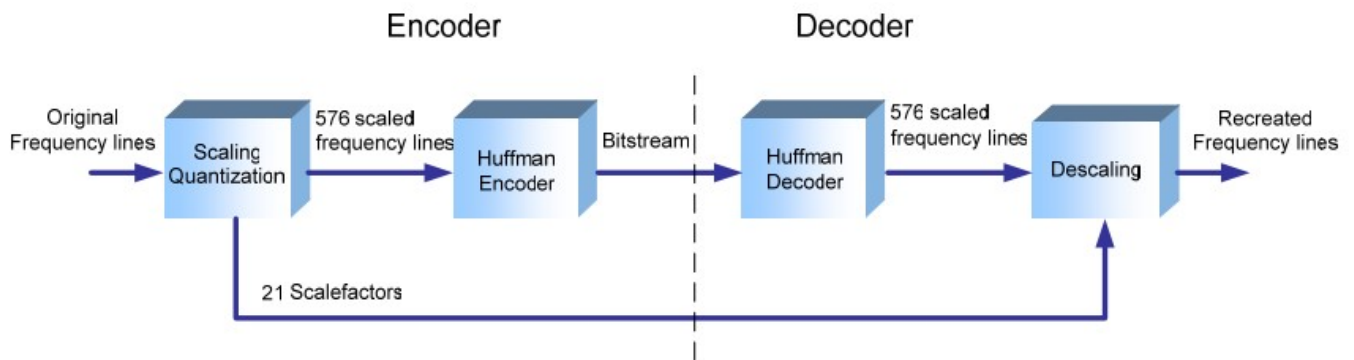


Ilustración 5–11 Proceso de recuantización²⁹

1.1.1.1.6 Reordenamiento del Espectro

Para una codificación efectiva, el reordenamiento es hecho antes de la codificación de huffman para los bloques pequeños. Normalmente la salida del MDCT está organizada por subbandas en frecuencia creciente. El reordenamiento es solo aplicado en bloques pequeños los cuales se encuentran organizados por subbandas luego en bloques y en frecuencias. De manera creciente. El reordenamiento reordena las líneas de frecuencias organizando las líneas de frecuencia primero por subbandas y después por frecuencias. La gráfica 5-12 ilustra el procedimiento de reordenamiento en la decodificación.

²⁹ Hsiao- Lun Liao Platform-based Hardware/Software CoDesign From Mp3 Decoder Master Thesis, Figure 2-7

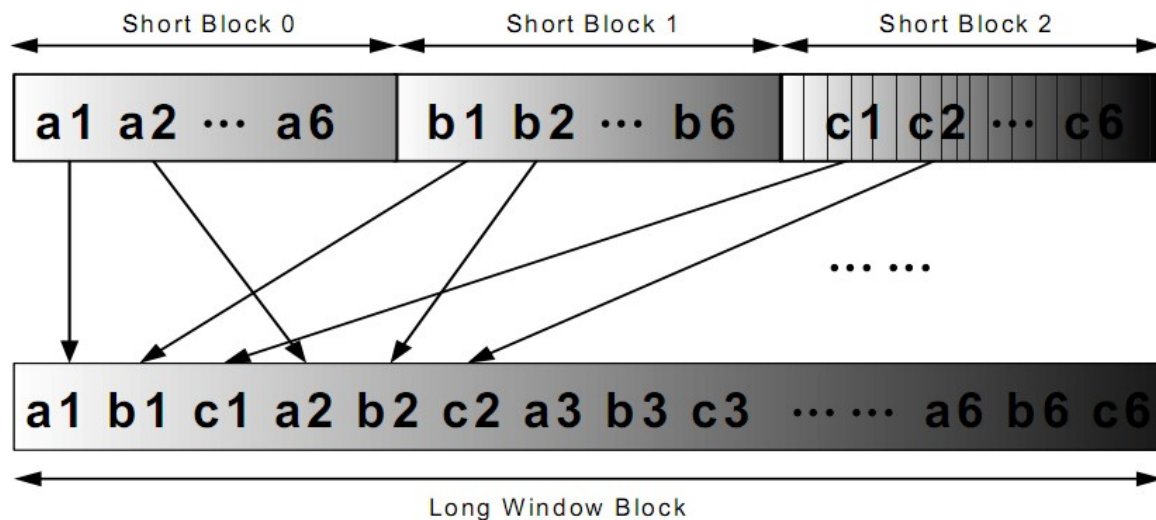


Ilustración 5–12 Reordenamiento del espectro³⁰

1.1.1.1.7 *Procesamiento Stereo*

Luego de realizar la recuantización, si el modo escogido es joint stereo: modo MS stereo o modo stereo intensivo, los 576 valores no vienen originalmente como canal izquierdo o canal derecho, en vez de esto, la información transferida es reconstruida a los canales izquierdo y derecho. El proceso stereo controlado por **mode** y **mode_extension**.

Ms Stereo Mode:

En el modo MS stereo mode los canales izquierdo y derecho son transmitidos como suma “M” y diferencia “S” de ambos canales respectivamente. Este método es recomendable cuando los canales son altamente correlacionados lo que significa que la señal de suma tiene más información y la señal stereo puede ser más comprimida eficientemente que si se usaran dos canales de manera independiente. Los canales pueden ser reconstruidos utilizando la ecuación. MS Stereo no tiene pérdidas.

11 y 12.

$$L_i = M_i + S_i 2^{(11)} \quad R_i = M_i - S_i 2^{(12)}$$

Modo Intensivo:

³⁰ Ibidem Figure 2-8

En el modo stereo intensivo el codificador codifica algunas subbandas de alta frecuencia produciendo una única señal de suma L+R en vez de enviar señales de subbandas de manera independiente a la izquierda “L” y a la derecha “R”. El balance entre la izquierda y la derecha es transmitido en vez de enviar los factores de escala. El decodificar reconstruye la izquierda y la derecha basándose solo en la señal de suma L+R la cual es transmitida en el canal de la izquierda en tanto que el balance es transmitido en reemplazo de los factores de escala (is_possb) en el canal de la derecha.

$$is_ratio = \tan(is_possb \times \pi/12) \quad (13)$$

$$Li = L'i \times is_ratio + is_ratio \quad (14)$$

$$Ri = L'i \times 11 + is_ratio \quad (15)$$

El parámetro is_possb está limitado a valores entre 0 y 6, por lo tanto la función tan() es fácilmente reemplazable por una pequeña tabla de búsqueda

1.1.1.1.8 Reducción de Alias

La reducción de aliasing se realiza para bloques largos. El bloque de antialias reduce el aliasing, creado por la utilización de un filtro ideal de paso incorrecto, y de esta manera se obtendrá una correcta reconstrucción del análisis hecho por el filter bank. Las líneas de frecuencia en el gránulo están ordenadas desde 0 como la frecuencia más baja hasta 576 como la más alta. La reducción de aliasing se hace juntando las líneas de frecuencias por medio de 8 cálculos de mariposa para cada subbanda. Los coeficientes de las mariposas son calculados utilizando la tabla 5-9 y sustituyéndolos por las ecuaciones 16 y 17

$$csi = 11 + ci2 \quad (16)$$

$$cai = ci1 + ci2 \quad (17)$$

I	0	1	2	3	4	5	6	7
ci	-0.6	-0.535	-0.33	-0.185	-0.095	-0.041	-0.00142	-0.0037

Tabla 5–9 Coeficientes de las mariposas³¹

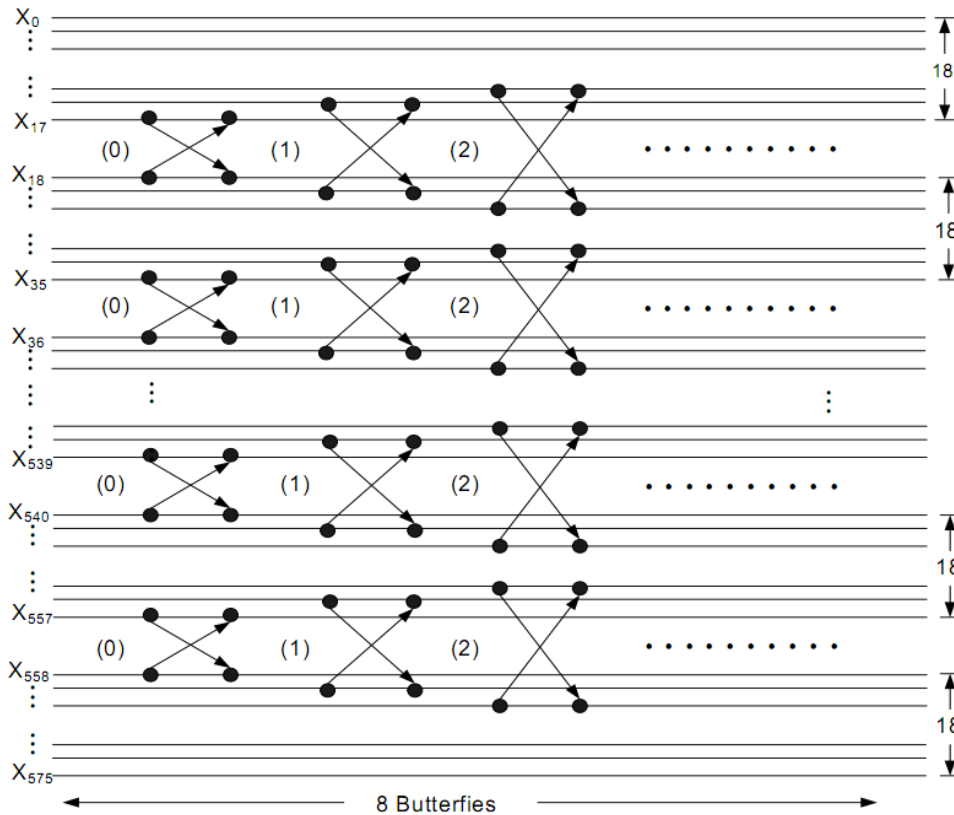
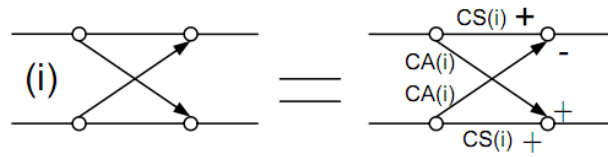


Ilustración 5–13 Procesamiento de las 8 mariposas³²

1.1.1.1.9IMDCT (Inversa de la transformada modificada del coseno)

La IMDCT in cooperación con la síntesis del bloque filter bank convierte muestras en el dominio de la frecuencia $X[k]$ al dominio del tiempo (Líneas de frecuencia) $x[i]$. n se define como el número de muestras a las que se les realiza ventaneo, para bloques pequeños $n=12$, para bloque largos $n=36$.³³ La IMDCT se define por la ecuación 18:

$$i=0\dots n-1$$

$$x_i = k=0n/2-1 X_k \cos \pi 2n 2i+1+n 22k+1 \quad (18)$$

En en el caso de bloques largos como entrada, la IMDCT toma 18 líneas de frecuencia como lectura y entrega 36 valores ($n=36$). En el caso de bloques pequeños como

31 Sripada Praveen MP3 decoder in theory and practice, 2006, table 12

32 Hsiao- Lun Liao Platfrom-based Hardware/Software CoDesign From Mp3 Decoder Master Thesis, Figure 2-9

33 Ibidem 2.3.7 IMDCT

entrada tres transformadas son realizadas lo cual produce 12 valores($n=12$) .El ventaneo clasifica los bloques en 4 tipos: start,short,normal y stop.

Para $n=36$ la primera mitad de la salida (Los primeras 18 valores) superponen (50% de superposición) con la segunda mitad del anterior valor calculado. La segunda mitad se almacena para el próximo cálculo, la superposición es llevada a cabo adicionando los valores de la primera mitad del primer frame con la segunda mitad del segundo frame.

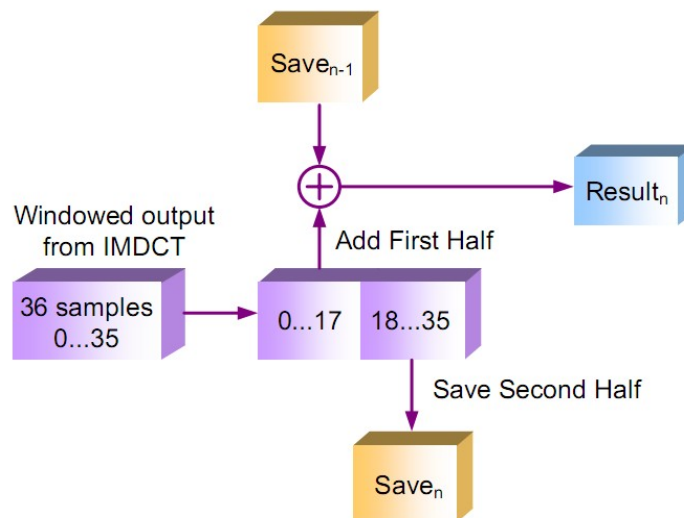


Ilustración 5–14 Operación de superposición³⁴

INVERSIÓN DE FRECUENCIA

Para compensar la inversión de frecuencias en la el filterbank polifásico en cada muestra tiempo de muestra impar de cada subbanda impar se multiplica por 1. Las subbandas están numeradas [0..31] y las muestras de tiempo de cada subbanda [0..17]

1.1.1.1.10 Filterbank Polifásico

El banco de filtro polifásico transforma 32 subbandas de 18 muestras en el dominio del tiempo de cada gránulo en 18 bloques de 32 muestras PCM. La siguiente ilustración describe el algoritmo y el procedimiento de la síntesis del filtro polifásico.

³⁴ Ibidem figure 2-10

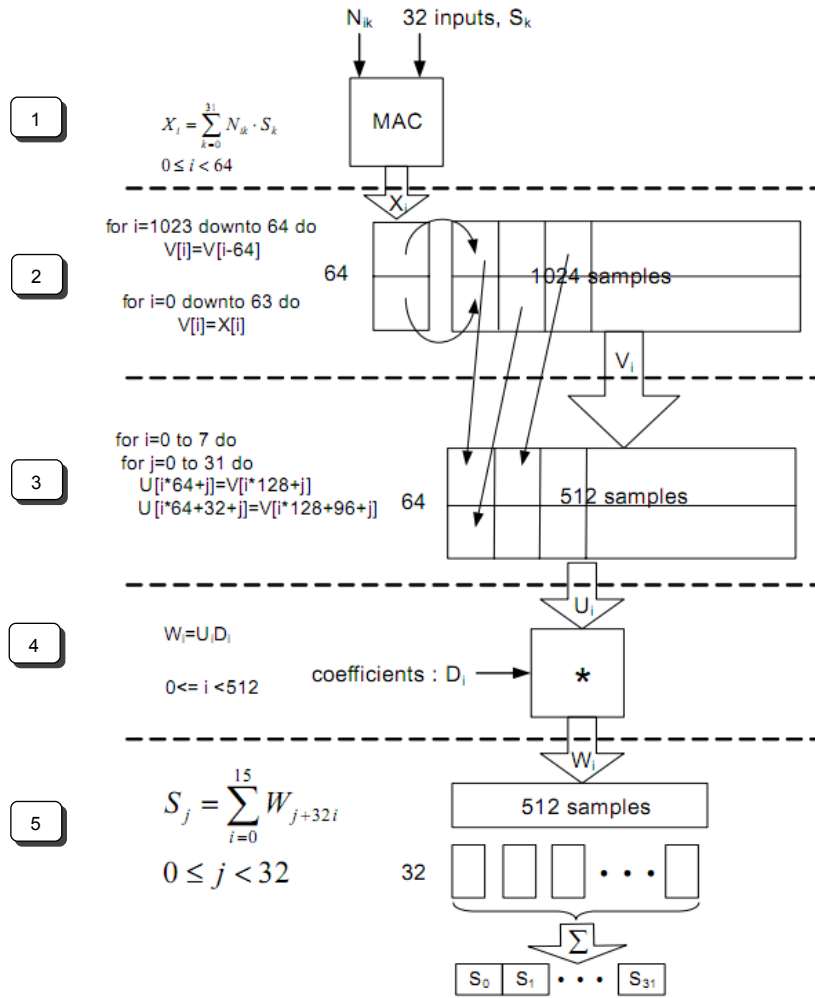


Ilustración 5-15

Operación de superposición³⁵

A continuación se describe el algoritmo utilizado en el banco de filtro :

1. Recepción de las muestras y vectorización $S_k \rightarrow X_i$

Se reciben 32 valores de subbandas los cuales serán transformados en un vector de 64 valores; el vector será almacenado en una fifo con capacidad de 16 vectores (1024 muestras). Para es propósito el estándar define la siguiente ecuación para contruir el vector.

$$X_i = \sum_{k=0}^{31} N_{ik} \cdot S_k \quad (19)$$

$$0 \leq i \leq 64$$

35 Hsiao- Lun Liao Platfrom-based Hardware/Software CoDesign From Mp3 Decoder Master Thesis, Figure 2-11

donde

$N_{i,k} = \cos 16 + i 2k + i \pi 640 \leq i \leq 63, 0 \leq k \leq 31, S_k = \text{Muestras en el dominio del tiempo}$ ⁽²⁰⁾

2. Ingreso de vector en la fifo y desplazamiento $X_i \rightarrow V_i$

Luego de haberse generado un nuevo vector X_i se debe realiza el ingreso a la fifo a través del desplazamiento interno. Los vector en la fifo son denominados como V_i .

for $i=1023$ downto 64 do

$V_i = V_i - 64$

for $i=0$ to 63 do

$V_i = X_i$

3. Contrucción de fifo de 512 muestras $V_i \rightarrow U_i$

Posteriormente se construye a partir de V_i , escojiendo de manera alternada 32 muestras el vector U_i de 64 valores utilizando el siguiente código de desplazamiento.

for $i=0$ to 7 do

for $j=0$ to 31 do

$U_i * 64 + j = V_i * 128 + j$

$U_i * 64 + 32 + j = V_i * 128 + 96 + j$

4. Ventaneo del vector $U_i \rightarrow W_i$

Y es almacenado en la fifo con capacidad de 8 vectores U_i

Ahora vector U_i entra en un tratamiento de ventaneo al ser multiplicando por el coeficiente D_i que referencia a valores definidos por el estándar³⁶ produciendo así el vector W_i

$W_i = U_i * D_i$

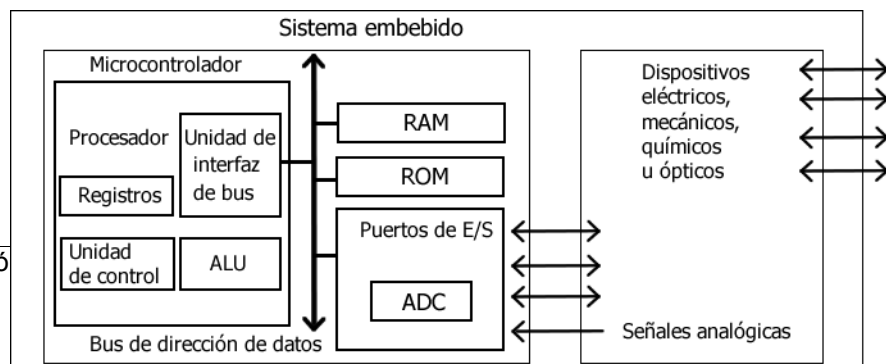
5. Salida final $W_i \rightarrow 32$ muestras PCM

36 ISO/IEC Coding of moving pictures and associated audio for digital storage media, Annex AB p 12

Finalmente el vector W_i se descompone en 16 vectores de 32 muestras que al sumarse representan 32 muestras PCM.

1.1 SISTEMAS EMBEBIDOS

Los sistemas embebidos son dispositivos que poseen una o más microcomputadoras³⁷ en su interior las cuales ejecutan de instrucciones interpretables que han sido grabadas en una memoria de tipo no volátil; El término embebido hacer referencia a algo encapsulado u oculto que no es visible, en la ilustración 5-16 podemos ver un ejemplo de un sistema embebido



37 Ver definició

Ilustración 5–16 Ejemplo de organización de un sistema embebido³⁸

Los componentes comunes de un sistema embebido son CPU, RAM, ROM, puertos de E/S y resto de dispositivos³⁹. La CPU dispone de una ALU (Unidad de lógica aritmética) donde se ejecutan las operaciones matemáticas, de registros utilizados como almacenamiento de alta velocidad para la realización de las operaciones, de interfaces de BUS utilizadas para comunicarse con el resto de dispositivos. La memoria ROM de tipo no volátil servirá para almacenar las instrucciones de software ejecutables y datos constantes. Las memorias temporales como la SRAM³⁶³⁶,SDRAM³⁶ servirán para el almacenamiento de datos volátiles. Los puertos son conexiones física que se encargan de la transferencia de señales entre los dispositivos para manejar las entrada o salida de datos.

1.2 MARCO LEGAL

El instituo Fraunhofer IIS⁴⁰ y Thomson⁴¹ tienen patentadas todas técnicas utilizadas para decodificar y codificar MP3⁴² tanto desde software como hardware. MP3 es un estándar internacional que puede ser utilizado por desarrolladores a nivel mundial sin embargo esto no implica que no se paguen royalties por su uso mientras las patentes estén vigentes. Thompson ha registrado sus patentes en los siguientes países :

Alemania	Francia	Noruega
Austria	Grecia	Portugal
Bélgica	Holanda	Reino Unido
Canada	Irlanda	Rusia
China	Italia	Suecia
Dinamarca	Japón	Suiza

38 W. Valvano Jonathan, Introduccion a los sistemas de microcomputadora embebidos, Cengage Learning Latin America,2004 Figura 1.1

39 Ibidem p 6

40 Instituto Fraunhofer <http://www.iis.fraunhofer.de/EN/index.jsp>

41 Fabricante Thomson <http://www.thomson.net>

42 Patent Portafolio, <http://www.mp3licensing.com/patents/> , 21 de noviembre del 2009

España	Korea	Taiwán
Estados Unidos	Luxemburgo	Ucrania
Filandia	Monaco	Portugal

Tabla 5–10 Países donde están registradas las patentes MP3⁴³

Las patentes MP3 solo son válidas dentro de los países anteriormente mencionados tanto para software, hardware como para medios, se excluyen países no citados anteriormente como Colombia. A continuación veremos la tabla de costos publicada en el sitio web de licenciamiento de MP3 en el capítulo hardware :

Patente MP3 para licenciamiento de software realizado por Fraunhofer IIS-A	
Decodificador	US \$0.75 por unidad
Codificador	US \$1.25 por unidad
Patente MP3 para licenciamiento de software hecho por terceros	
Decodificador	US \$0.75 por unidad
Codificador	US \$1.25 por unidad
Tarifa anual por uso de la patente	US \$15.000 por año

Tabla 5–11 Costos de licenciamiento de MP3⁴⁴

Estas tarifas deberán ser cubiertas por los fabricantes de los codificadores y decodificadores de hardware, los fabricantes de electrónica de consumo que hagan uso de los codificadores o de los decodificadores deberán estar amparados bajo una licencia que cubra sus operaciones, en tanto que los usuarios finales no deberán preocuparse por su uso.

43 Patents, <http://www.mp3licensing.com/patents/index.html>, 21 de noviembre del 2009

44 Royalties rates, <http://mp3licensing.com/royalty/hardware.html>, 21 noviembre del 2009

Los desarrollos no comerciales como trabajos académicos o desarrollos con licenciamiento libre estarán exentos de pagos según lo acordado por los dueños de la patente siempre y cuando se mantengan así bajo esta condición.⁴⁵

2 DISEÑO METODOLÓGICO

2.1 Definición de la Hipótesis

Los decodificadores MP3 desarrollados en software requieren de más alto procesamiento de CPU, de mayor ocupación de RAM y mayor consumo de energía que los desarrollados en hardware.⁴⁶

⁴⁵ FAQ <http://www.mp3licensing.com/help/developers.html>, 21 de noviembre del 2009

⁴⁶ Remitir los términos al glosario

2.2 Metodología de la Investigación

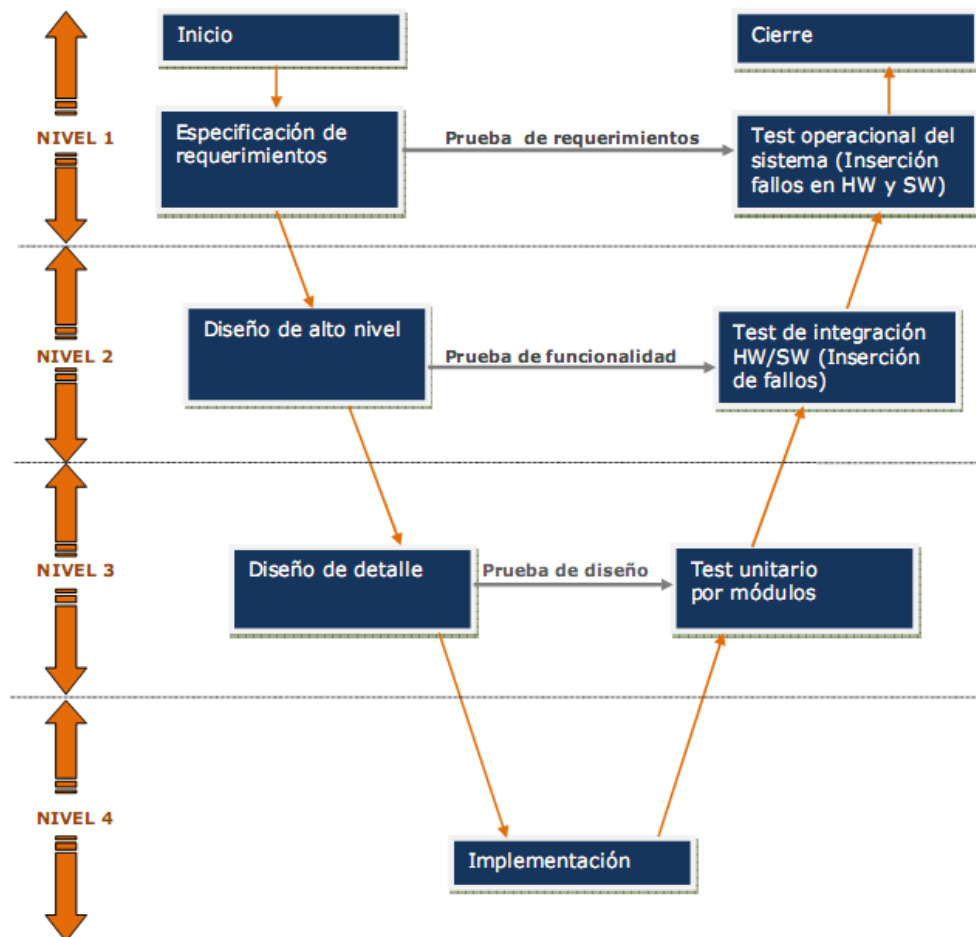


Ilustración 6–1 Metodología de desarrollo VEE⁴⁷

La metodología utilizada para este proyecto será la VEE (Ilustración 6-1) para desarrollar sistemas integrados. La metodología es económica y ágil, ideal trabajar en equipos pequeños de personas; permite que mientras cada etapa del proyecto se desarrolla se prepara paralelamente las pruebas de cada fase del proyecto basadas en indicadores previamente establecidos. La primera parte del ciclo de vida se dedica al desarrollo del producto y la preparación de las pruebas, la segunda parte del ciclo realiza pruebas exhaustivas para realizar las pruebas y corregir los errores. A continuación se explica brevemente cada fase⁴⁸:

⁴⁷Rook Paul, Software Reliability Handbook, First Edition, 1991 Figure 5.1

⁴⁸Ibidem p 113

2.2.1 Especificación de Requerimientos (Fase 1):

Se deben documentar en detalle los diferentes requerimientos del sistema a desarrollar, se describen las trazabilidades de los requerimientos, se define una lista de chequeo para cumplimiento de la trazabilidad. Se utilizan diagramas de caso de uso y de secuencia.

2.2.2 Diseño de Alto Nivel (Fase 2):

El objetivo de esta etapa es obtener un diseño y visión general del sistema. Se documentan diagramas de bloque en alto nivel. Se preparan las n pruebas de funcionalidad.

2.2.3 Diseño en Detalle (FASE 3):

Se detalla cada bloque de la fase anterior, se utilizan diagramas de estado y actividad (UML). Se preparan las evaluaciones sobre los diseños.

2.2.4 Implementación (Fase 4):

Se adelanta el trabajo técnico para desarrollo del sistema integrado, se realizan descripciones de código fuente y documentación técnica para los desarrollares como la utilización de un API.

2.2.5 Test Unitario (Fase 5):

En esta fase se verifica cada módulo HW y SW de forma unitaria, comprobando su funcionamiento adecuado.

2.2.6 Integración (Fase 6):

En esta fase se integran los distintos módulos que forman el sistema. Como en el caso anterior, ha de generarse un documento de pruebas. Por una parte, se debe comprobar en todo el sistema el funcionamiento correcto, y por otra, en caso de tratarse con un sistema tolerante a fallos, debe verificarse que ante la presencia de un fallo persiste el funcionamiento correcto. Se comprueba el cumplimiento de los requisitos establecidos.

2.2.7 Test Operacional del Sistema (Fase 7):

Se realizan las últimas pruebas pero sobre un escenario real, hechas por un usuario utilizando el aplicativo y el hardware, donde serán anotadas una vez más las pruebas realizadas y los resultados obtenidos.⁴⁹

La toma de resultados será realizada utilizando el decodificador de MP3 través de un aplicativo que lea audio de un streaming⁵⁰ en internet o desde un dispositivo de almacenamiento y los pase al decodificador. Para cada módulo del decodificador se realizarán mediciones de latencia de operación, cantidad de memoria ram utilizada y cantidad de bloques de lógica que ocupan de tal manera que sirvan como referentes para las optimizaciones del decodificador.

La parte 4 ISO/IEC 11172 define las pruebas de calidad de audio y video, para audio se definen pruebas de precisión del audio. Para este efecto se dispone de una ecuación denominada RMS

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N (t_i - r_i)^2}$$

t_i corresponde a la muestra de la decodificación que se está probando, r_i es la muestra de sonido utilizada como referencia y N corresponde al número de muestras utilizada. Las condiciones que se deben cumplir para que el decodificador cumpla el estándar deberán ser: RMS debe tener un valor menor a $8,81 \times 10^{-6}$ y la diferencia absoluta entre $t_i - r_i$ no deberá superar en ningún momento el valor de $6,1 \times 10^{-5}$

49 Ibidem p 114-123

50 Ver glosario

2.3 Esquema Temático

INTRODUCCIÓN

CAPITULO 1. INTRODUCCIÓN

CAPÍTULO 2. DESCRIPCIÓN PROCESO DECODIFICACIÓN MP3

El formato MP3

El proceso de decodificación MP3

CAPÍTULO 3. DISEÑO Y DESARROLLO DEL HARDWARE

Diseño y mapa del hardware

Implementación del hardware

Diseño detallado de cada bloque

CAPÍTULO 4. PRUEBAS Y RESULTADOS DE LA DECODIFICACIÓN

Pruebas y análisis generales de la decodificación

Pruebas y análisis del bloque huffman

Pruebas y análisis de bloque IMDCT

Pruebas y análisis del bloque filterbank

Calidad de sonido

CAPÍTULO 5. SOFTWARE DE CONTROL DE DECODIFICADOR

Mapeo de las memorias y la interfaz

Integración del software y el hardware

3 RESULTADOS ESPERADOS

Implementación de un prototipo estable, con una funcionalidad técnica aceptable definida por los indicadores a definir.

Disponer de un wiki con documentación completa de todo el proyecto con toda la documentación técnica .

4 ESTRATEGIA A UTILIZAR PARA LA TRANSFERENCIA DE RESULTADOS

Los resultados se darán a conocer en el wiki del proyecto, la documentación se traducirá totalmente a inglés y portugués de tal manera que pueda ser entendida por personas de diferentes lenguajes.

5 CRONOGRAMA

Semanas	Descripción de actividad	Completado
2	Diseño de los sistemas de software y hardware, actualización y construcción de los indicadores del proyecto. Creación del wiki.	0%
6	Desarrollo, documentación y pruebas de bloque decodificador de Huffman	5,71%
2	Desarrollo, documentación y pruebas de Bloque de recuantización	22,86%

1	Desarrollo, documentación y pruebas de bloque de reordenamiento	28,57%
2	Desarrollo, documentación y pruebas de bloque de procesamiento stereo	31,43%
1	Desarrollo, documentación y pruebas de bloque de Antialias	37,14%
8	Desarrollo, documentación y pruebas de bloque IMDCT	40,00%
6	Desarrollo, documentación y pruebas de Bloque Filterbank	62,86%
1	Realización de pruebas y análisis de los datos del decodificador MP3	80,00%
1	Ajustes al decodificador posterior a las pruebas	85,71%
1	Implantación GNU/uCLinux y desarrollo de software MP3	88,57%
2	Desarrollo de software Integración y hardware.	91,43%
1	Pruebas integración software y hardware. Ajustes finales al proyecto	97,14%
34 semanas		100,00%

Tabla 9–1 Cronograma del proyecto

6 RECURSOS MATERIALES

Recursos	Valor y descripción
KIT DE DESARROLLO ALTERA-DE2-35	Recurso propio del estudiante KIT de desarrollo FPGA. Costo 480 USD
SD CARD 2GB	Recurso propio del estudiante Memoria no volátil para el almacenaje del audio. Costo 15 USD
Estándar ISO/IEC 11172-3	Recurso propio del estudiante Documento que describe el estándar MPEG-I Costo 40 USD
Ruckert Martin - Understanding MP3	Recurso propio del estudiante Libro que describe detalladamente la teoría del estándar MPEG-I Costo 75 USD

Total	610 USD
--------------	----------------

Tabla 10–1 Recursos del proyecto

7 APÉNDICE

7.1 *Codificación de Huffman*

La codificación de huffman hace parte de las conocidas compresiones sin pérdida más utilizadas ya que permite recuperar los datos originales antes de comprimirlos puesto que el método de compresión es inverso al de descompresión. La manera como hace esto es conseguir la razón más óptima entre el número de símbolos de código por símbolos de fuente, pudiéndose representar el código utilizada en un árbol binario.

Pasos para codificar un árbol de huffman⁵¹

El primer paso en la construcción del código de huffman es contar el número de veces que aparece cada caracter de la cadena que se va a codificar.

⁵¹ Sedgewick Robert, Algoritmos en c++, 1 Edición, Addison-Wesley ,1995 p 357

El segundo paso es empezar a construir el árbol de codificación de abajo hacia arriba de acuerdo a las frecuencias. Se deberán crear N nodos por cada frecuencia diferente de 0. Posteriormente seleccionamos los dos nodos con las frecuencias más bajas y los tomamos como hijos, creamos un padre para ellos con una frecuencia igual a la suma de ambos. Ver la figura; luego entre los nodos restantes escogeremos los dos nodos con la menor frecuencia y se vuelve a crear un nuevo nodo de la misma forma, tal como se muestra en el tercer diagrama. De esta manera se van construyendo subárboles más grandes, a la vez que se reduce en cada paso el número de subárboles del bosque. Finalmente solo debe quedar un solo árbol construido.

El tercer paso consiste en obtener el código de Huffman reemplazando las frecuencias por los caracteres correspondientes y considerando el árbol como una ruta de codificación, cada bifurcación a la izquierda corresponde al bit 0 en tanto que la bifurcación a la derecha corresponde al código 1.

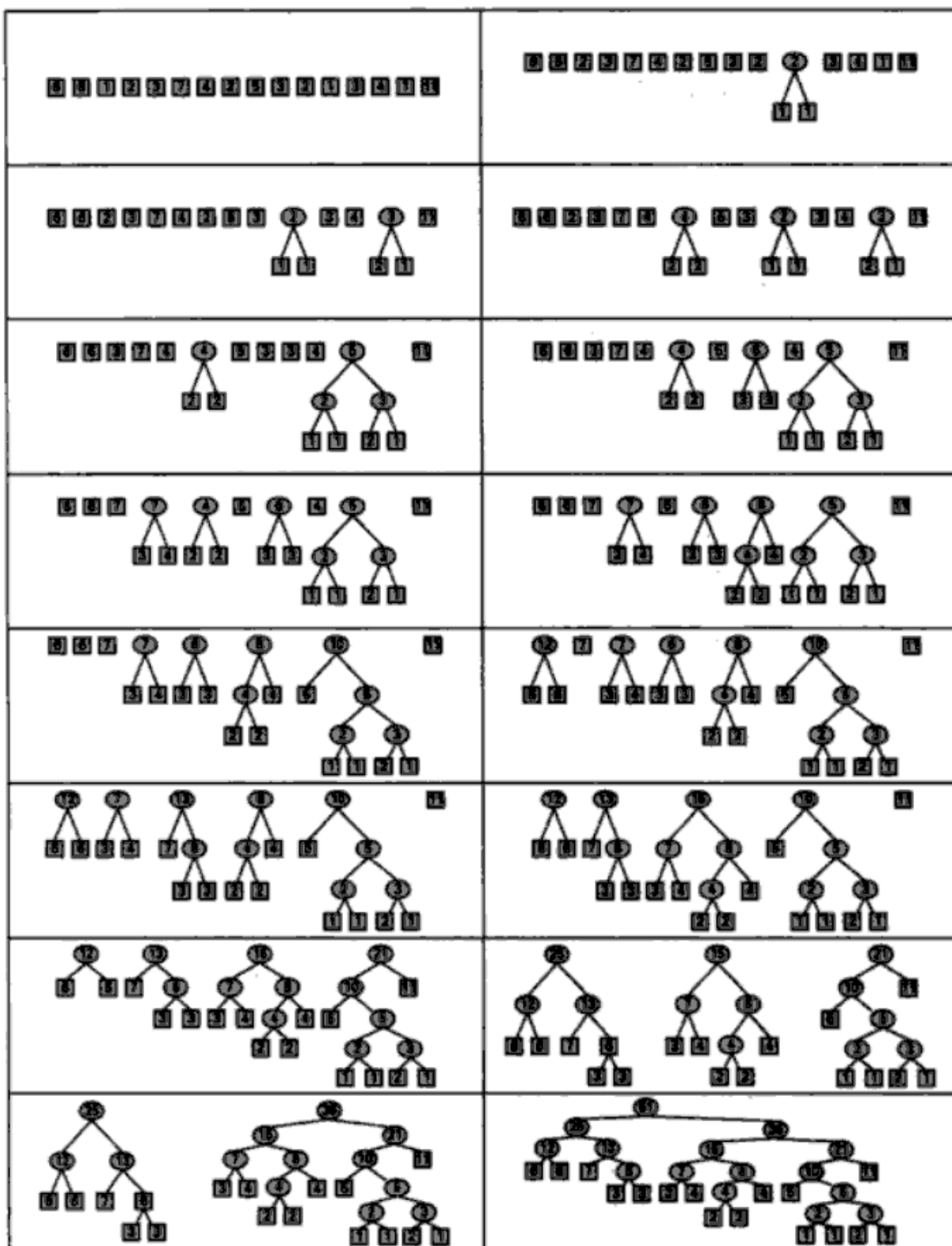


Ilustración 11-1 Ejemplo de construcción de un árbol de huffman

7.2 Glosario

Audio CD: Conocido como el estándar Red Book y fue lanzado en el año de 1980. Permite almacenar audio 2 canales de muestras a 16 bits con frecuencia 44100 Hz⁵³

Banda crítica : Es la máxima de frecuencias que al ser ruido blanco es capaz de enmascarar un sonido puro con la misma intensidad, centrado a la banda del ruido.⁵⁴

Bit rate : En telecomunicación e informática, el término **tasa de bits** (en inglés bit rate) define el número de bits que se transmiten por unidad de tiempo a través de un sistema de transmisión digital o entre dos dispositivos digitales. Significa por lo tanto la velocidad de transferencia de datos.⁵⁵

Broadcasting : Hace referencia a la transmisión para audiencias de señales de audio o de video.

Códec: es la abreviatura de *codificador-decodificador*. Es una tecnología desarrollada en software, hardware o una combinación de ambos que tiene la capacidad de transformar señales analógicas a señales comprimidas o viceversa ⁵⁶

CPU: Siglas del acrónimo Unidad de procesamiento central. Hace referencia al componente de una computadora que procesa las instrucciones y procesa los datos contenidos en los programas de la computadora.⁵⁷

Frecuencia de muestreo : La **tasa o frecuencia de muestreo** es el número de muestras por unidad de tiempo que se toman de una señal continua para producir una señal discreta, durante el proceso necesario para convertirla de analógica en digital. Como todas las frecuencias, generalmente se expresa en hercios (Hz, ciclos por segundo) o múltiplos suyos, como el kilohercio (kHz), aunque pueden utilizarse otras magnitudes.⁵⁸

53 Pohlmann, Ken C.Principles of digital audio ,McGraw Hill,2005,p 306

54 Salesa E. Tratado de audiología,1era edición, Masson, 2005 p 88

55 Pohlmann, Ken C.Principles of digital audio fifth edition,,McGraw Hill,2005,p 800

56 Pohlmann, Ken C.Principles of digital audio fifth edition,,McGraw Hill,2005,p 117

57 Dan Gookin, PC for dummies, 11th edition, Wiley Publishing,2007, p 76

58 Underdahl Keith,Digital video for dummies, 4 edition, p 112

FPGA: (*Field Programmable Gate Array*) is un dispositivo lógico que contiene un arreglo bidimensional de celdas de lógica genéricas de switches programables. Una celda lógica puede ser programada para realizar funciones simples y un switch programable puede ser personalizado para realizar interconexiones entre las celdas lógicas⁵⁹

Gadgets : Un gadget es un dispositivo tecnológico con una función en particular que se ofrece como algo novedoso. Sus diseños son poco comunes y por lo general tiene un diseño más inteligente que los dispositivos que se ofrecen en el mercado.⁶⁰

Núcleo ip : Un núcleo ip es núcleo de unidad, una plantilla de chip reusable que puede servir para propósito generales. Viene tipo silicona o tipo soft. El núcleo ip blando suelen distribuirse como netlist (Representación lógica) o como código el cual puede ser sintetizable y vienen hechos en lenguajes como Verilog o VHDL permitiendo a los programadores poder realizar personalizaciones ⁶¹

ISO : es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.⁶²

MP3 : Acrónimo de siglas que significan (*MPEG-1 Audio Layer 3*)

MPEG: Acrónimo de siglas que significan Moving Picture Experts Group grupo de trabajo de ISO/IEC encargado de desarrollar estándares de codificación de audio y vídeo. Desarrollaron MPEG-1 que definió MP3⁶³

59 Chu Pong P, Fpga Prototyping by verilog examples, first edition, Wiley, 2008, p 15

60 New York Magazine 4 feb 1985, p 14

61 Kamat Rajanish Unleash the system on chip using FPGAs and Handel C, 1 edition, Springer, 2009 p 143

62 Iso about us, <http://www.iso.org/iso/about.htm> , 19/10/2009

Microcomputadora: Corresponde al tipo más pequeño de computadora, consta de varios circuitos integrados incluyendo un microprocesador, memoria y circuitos de interfaz de entrada y salida.⁶⁴

RAM : Acrónimo de Random Access Memory. Es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados.⁶⁵

SDRAM: Acrónimo de DRAM *sincrónica*, es un tipo de memoria que está sincronizada con con el bus de memoria, la SDRAM entrega información a muy alta velocidad utilizando una interfaz con reloj. La SDRAM soluciona el problema de la latencia que sufren las memorias DRAM puesto que sus señales se encuentran sincronizadas con el bus de la motherboard.⁶⁶

SRAM: Acrónimo de static RAM, este tipo de memoria no requiere un ciclo de refresco como las memoria DRAM por lo tanto permite el acceso a la información mucho más rápidamente, su producción es más costosa que otros tipos de DRAM.⁶⁷

Streaming: Hace se refiere a ver u oír un archivo al realizar una conexión con un host sin necesidad de descargarlo completamente en ordenador⁶⁸

VHDL: Es un lenguaje usado por ingenieros definido por el IEEE (Institute of Electrical and Electronics Engineers) (ANSI/IEEE 1076-1993) que se usa para diseñar circuitos digitales.⁶⁹

7.3 Glosario estándar ISO/IEC 11172-3

Análisis filterbank: Filterbank es un codificador que convierte una señal de PCM de banda ancha de audio en un conjunto de muestras subbanda submuestras.

Banda crítica : Parte del dominio espectral que corresponde a una anchura de una franja

63 Pohlmann, Ken C.Principles of digital audio fifth edition,,McGraw Hill,2005,p 674

64Tocci Ronald, Sistemas digitales, octava edición,Pearson education,2003, p 19

65 Dan Gookin, PC fro dummies, 11th edition, Willey Publishing,2007, p 94

66 Mueller Scott, Upgrading and repairing laptops,1 edition, Que, 2003, p 222

67 Mueller Scott, Upgrading and repairing laptops,1 edition, Que, 2003, p 830

68 Topic Michael, Streaming media demystified, 1 edition, McGraw-Hill Telecom 2002, p 10

69 Chu Pong P, Fpga Prototyping by verilog examples, first edition,Wiley,2008, p xxii

Bitrate Variable: Operación donde el bit rate varia con el tiempo durante la decodificación del bitstream

Bitstream: Serie de bits seguidos leídos desde un archivo MPEG.

Canal: Un medio digital que almacena o transporta una corriente de la norma ISO 1117-2.

Compresión: Reducción del número de bits utilizados para representar un elemento de datos.

CRC: Código de redundancia cíclica

Decuantización: Proceso de decodificación de una subbanda que recupera la original cuantificada en valores.

Filterbank : Conjunto de filtros pasa banda que cubre todo el rango de frecuencia de audio.

Frame : Una parte de la señal de audio que corresponde a un número determinado de muestras de audio PCM.

Gránulos [Layer III]: 576 líneas de frecuencia con su correspondiente side information.

Huffman coding: Un método específico para la codificación de la entropía(Incertidumbre)

IMDCT: Inversa de la transformada modificada del coseno.

Layer : Uno de los niveles de la jerarquía del sistema de audio definido por el estandar internacional.

Banda de factor de escala: Un conjunto de línea de frecuencias que son escaladas por un factor de escala

Índice de factor de escala: Código para seleccionar un factor de escala

Factor de escala: Conjunto de valores que se escalan antes de la cuantificación para reducir el ruido al ser enmascarado.

Side information: Información del bitsream necesaria para controlar el decodificador

Subband : Subdivisión de una frecuencia de audio

Syncword: Palabra de 12 bit de un bitsream que identifica el comienzo de un frame.

8 BIBLIOGRAFÍA

- [1] ISO/IEC 11172-3:1993 Information Technology – Coding of moving pictures and associated audio for signal storage media at up to about 1,5 Mbit/s 1993
- [2] Rucker Martin, Understanding MP3, Gvw-Vieweg,2005
- [3] Lagerström, Krister Design and Implementation of an MPEG-1 Layer III Audio Decoder, Master's Thesis : 2001

- [4] Sripada Praveen MP3 DECODER in Theory and Practice, Master Thesis Report,2006
- [5] Le-Tien Thuong FPGA-Based Architecture Of Mp3 Decoding Core For Multimedia Systems:2005
- [6] Liao Hsiao- Lun Platfrom-based Hardware/Software CoDesign From MP3 Decoder, Master Thesis 2005
- [7] Raissi Rassol The Theory behind MP3,2002,
- [8] Fältman Irina A Hardware Implementation of an MP3 Decoder: 2003
- [9] Singh Maninder pLayer-i An Internet Based Muzik Player: 2008,
- [10]Rook Paul, Software Reliability Handbook, First Edition, 1991
- [11]W. Valvano Jonathan, Introduccion a los sistemas de microcomputadora embebidos, Cengage Learning Latin America,2004
- [12]Saposhkov M.A, Electroacústica, Editorial reverté 1era edición,1993
- [13]Sedgewick Robert ,Algoritmos en c++, 1 Edición, Addison-Wesley ,1995
- [14]Layer III: A Flexible Coding Standard – MPEG digital Audio Coding, AES E-Library, Eberlein, 1993
- [15]Jay Gould, Designing flexible high performance embedeed systems, Xilinx inc

1 ANEXOS

1.1 Cotización de decodificadores de MP3 de núcleo blando

Aptx.com De swray@aptx.com 24 de agosto del 2009

Hi Juan Pablo,

We can deliver MP3 and WMA Decode on Virtex4 FX FPGA, but we will have to put engineering resource to this porting work and I suspect it might take at least 6weeks work.

Would you be able to sustain pricing of ~ \$60k? Unfortunately this would be our discount rate and we could not reduce the pricing any more than this?

I look forward to hearing from you.

Best regards,
Stephen

Ingenient.com De Bob.Lee@ingenient.com 20 de agosto del 2009

Hi Pablo,

Although our decoder will work on ARM processor. It is working as OBJ library.

For the budget you have, it is far too low to buy any ipcore, thi ipcore cost 75K. I recommend you to find alternative sources such as the open source solution.

FastVDO.com de contact@fastvdo.com 25 de agosto del 2009

Juan,

We have MP3 decoder that is available for Virtex4 FPGA. We do not have WMA decoder. Let me know if you would like to buy.

The Price fot our decoder will be 50.000 USD

Regards,

~Raj.