UNIVERSIDAD LA SALLE

ESCUELA DE INGENIERIA

INCORPORADA A LA U.N.A.M.

"CODIFICACION DE SEÑALES DE VIDEO"

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO-ELECTRICISTA

CON ESPECIALIDAD EN ELECTRONICA Y COMUNICACIONES

PRESENTA

MANUEL GERARDO RAGGI GONZALEZ

Asesor de Tesis: M. en C. Guillermo Aranda Pérez.

México, D.F. 1992.

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

CODIFICACION DE SEÑALES DE VIDEO

1. INTRODUCCION

Es posible que nadie se imagine la magnitud de la revolución de la información. En general, la gente solo ve la punta del iceberg, sin embargo, los avances en la integración electrónica han logrado que bajen los precios de los circuitos y así mismo, de los sistemas digitales, y esto representa una solución a muchos problemas del alto costo de fabricación que se tenían antes en los equipos.

Un gran número de personas encuentran conveniente gran cantidad de sistemas digitales, entre ellos, el sistema de cajeros automáticos de los bancos, los cuales, al introducir la tarjeta magnética, es leída inmediatamente para obtener su identificación. Otros sistemas son, la transferencia de fondos vía telefónica, la televisión de alta definición (tanto en imagen como en sonido), CD o discos compactos (*compact discs*), correo electrónico y comunicaciones por computadora.

Tradicionalmente las universidades llevaban cursor arduos de comunicaciones analógicas, con algunos sistemas de comunicación digital al final de los cursos; hoy día, se trata de impartir un curso de comunicaciones digitales al principio. Esto se esta realizando no solo por la universalidad de sus aplicaciones, sino porque, en muchos casos, las comunicaciones digitales son más sencillas de analizar que su contra parte analógica.

Por el gran número de aplicaciones que han encontrado los sistemas digitales, solo se tocaran algunos puntos:

1. Para empezar, se hará mención de los elementos que componen un sistema digital (Capitulo 1), así mismo se hara una comparación entre los sistemas analógicos y los sistema digitales (Capitulo 1), posteriormente se mencionaran algunos códigos simples (Capitulo 1) y también se tocaran códigos nuevos en el área de comunicaciones como son los códigos aritméticos y los modelos estadísticos (Capitulo 1).

2. En el Capítulo 2 se describira que es codificador de fuente, los puntos más importanes para muestrear una señal, los errores que puede sufrir la señal al ser muestreada, y brevemente se tocara a la módulación más utilizada para las señales de los sistemas digitales, que la modulación PCM. Dentro de los moduladores PCM se veran los cuantificadores seriales y paralelos, y al mismo tiempo se describira el código Gray, para pasar a la demodulación ó decodificación de los sistemas PCM. Se verán también lo que son los cuantificadores no uniformes, sus ventajas y desventajas sobre los cuantificadores uniformes y terminara el capítulo con el proceso de expansión/compresión para los cuantificadores no uniformes.

3. Se describirá al ejemplo más típico para la utilización de señales de video digitales, que es la Televisión Digital (Capitulo 3), se revisara una tabla cronológica de los eventos más importantes en los últimos 100 años, los cuales han dado lugar a la creación de los sistemas de Televisión Digital, y en especial al sistema de Televisión de Alta Definición (HDTV), después de revisar la tabla cronológica, se describira rapidamente lo que es la Televisión Analógica, esto con el fin de tener claros ciertos puntos importantes que estan relacionados tanto con la Televisión Analógica como con la Televisión Digital, con estos puntos claros, se describira como se realiza el muestreo de la señal de Televisión Digital, con cuantos niveles de cuantificación se codifica. Este tema es particularmente interesante, porque compara brevemente la coficación compuesta y la codificación por componentes y da lugar para hablar de la Norma 4:2:2 para la codificación de señales de video.

4. Gracias a los puntos anteriores se describiran algunas ventajas e inconvenientes de un sistema de Televisión Digital y dará lugar a mencionar la capacidad de memoria necesaria para un sistema de Televisión Digital.

5. Un tema que a primera vista podría parecer que no se relaciona con la Televisión Digital, es el la Transformada del Coseno Discreta (DCT), sin embargo, por la gran importancia que tiene en los sistemas digitales de codificación y análisis de espectro en tiempo real, se dan las bases teóricas, aunque esta teoría podría parecer que se aleja de los sistemas de Televisión, será de suma importancia en partes posteriores del mismo capítulo.

6. Muchas veces en conversaciones referentes al Procesamiento Digital de Señales (DSP), no se toca lo referente al DSP para Video, sin embargo, el DSP para Video es aplicable en cadenas de Televisión, sobre todo en los equipos de efectos de video digital (DVE), también es muy aplicable a los sistemas de comunicación, por sus ventajas de calidad de imagen y su puede aplicar hasta en los equipos electrónicos de consumo, como sería con las video grabadoras digitales.

7. Por la necesidad que se tiene de procesar imágenes de video en tiempo real y de tener sistemas de comunicación que trabajen de una forma confiable, se han diseñado procesadores de video digital en forma de circuitos integrados, un ejemplo es el realizado por la compañia NEC.

8. Uno de los puntos más importantes y más controvertidos del video digital es la gran cantidad de espacio que ocupa, por lo que una parte del capítulo 3 esta dedicada a la compresión de las imágenes de video, como es que trabaja y los estandars que se han creado para este problema.

9. Teniendo como base un sistema digital de comunicaciones, se proveera al sistema de seguridad adicional (Capítulo 4), se explicara brevemente las bases para codificar-decodificar un canal, así como también se explicaran las bases y algunos sistemas importartes de encriptación-desencriptación, como el estandar de encriptación de datos (DES).

10. Por último se explicara brevemente dos sitemas de comunicacón digital de video para satélites, los cuales utilizan un sistemas de encriptación-desencriptación de datos basados en el DES, el sistema Video Cipher II (VCII) y el sistema B-MAC.

1.1 INTRODUCCION E HISTORIA

Al principio las primeras formas de comunicación eran los sonidos generados por las cuerdas vocales de los humanos y de los animales, y solo eran recibidos por el oído. Cuando se quiso aumentar la distancia, el sentido de la vista se utilizó para sustituir al oído. Por ejemplo, dos mil años A.C., los Griegos utilizaban una especie de telégrafo utilizando antorchas para comunicarse. Diferentes combinaciones con antorchas y la posición de estas se utilizaba para representar las letras del alfabeto Griego. Estas señales con antorchas, representan el primer ejemplo de comunicación de datos. Mucho después, el sonido del tambor fue utilizado para comunicarse a grandes distancias, una vez más utilizando el sentido del oído. Los incrementos de distancia eran ahora posibles, ya que los sonidos del tambor son más fácilmente reconocibles a grandes distancias, comparándolos con la voz humana.

En el siglo XVIII, la comunicación por letras se había perfeccionado utilizando las banderas de semáforo. Estas banderas, como con las antorchas Griegas, se valían de la visión para recibir la señal. Esto por supuesto, limitaba la transmisión a gran distancia.

En 1753, Charles Morrison, un cirujano Inglés, sugirió un sistema de transmisión eléctrico utilizando un alambre para cada letra del alfabeto. Un sistema utilizando una esfera y papel, con letras impresas para la recepción.

En 1835, Samuel Morse, empezó a experimentar con el telégrafo, como lo conocemos hoy día. Dos años después, en 1837, el telégrafo fue inventado por Samuel Morse, en los Estados Unidos, y al mismo tiempo por Sir Charles Wheatstone, en la Gran Bretaña. El primer telegrama público

fue enviada en 1844, y por tanto comunicación eléctrica se inició.

Los esquemas de comunicación son esencialmente digitales por naturaleza. Esto en sí es cierto, solo en el caso de que se utilicen un número limitado de mensajes. Estos sistemas se dejaron de usar hasta que Alexander Graham Bell inventó el teléfono en 1876, el cual es un sistema eléctrico analógico. Después de este inventó pareció que las comunicaciones analógicas dejaron totalmente atrás a las comunicaciones digitales.

Tomó cerca de cien años en cerrarse el círculo de nuevo, ya que, en 1976, en muchas áreas tradicionalmente consideradas como analógicas, las comunicaciones digitales empezaron a sustituirlas. Esta explosión de interés por las comunicaciones digitales fue posible gracias a los revolucionarios avances en computación y en componentes de estado sólido.

Las aplicaciones comerciales en comunicaciones digitales empezaron en 1962. El sistema de transmisión T1, introducido en ese año por la compañía Bell System, marcó el comienzo de una revolución comercial para lo digital. Para el final de ese año, cerca de 250 sistemas de comunicación digital habían sido instalados. Para mediados de 1976, el número excedió los 3 millones, y solo se había raspado la superficie.

Para mediados de los 80's, cuando las computadoras estaban celebrando su 40avo. aniversario, siendo aún joven la tecnología de estado sólido, las redes digitales controladas por computadora, ya estaban disponibles comercialmente. La información que se tenía, había alcanzado un nivel de madurez, el cual tendría profunda influencia en muchas fases de la vida humana. Las comunicaciones de acceso

instantáneo, sistemas cronométricos digitales para automóviles, aviones comerciales, se habían vuelto una realidad.

La humanidad tenía un insaciable apetito por los datos. Las computadoras personales en el hogar se estaban volviendo comunes. La velocidad en la que se están incrementado estos sistemas es inaudita.

Tomó cerca de 20 siglos el ir de comunicaciones con antorchas a las comunicaciones eléctricas digitales de datos. Tomó solo 20 años el ir de transmisiones primitivas eléctricas de datos hasta sistemas avanzados de alta velocidad para procesos en comunicación. El final aún no se escribe.

1.2 ELEMENTOS DE UN SISTEMA DIGITAL

Para comenzar, veamos en forma global el sistema actual de comunicaciones. En la Figura 1.1, se ve un diagrama de bloques típico de un sistema de transmisión y recepción digital. Este diagrama es comprensible, aunque no todos los bloques pueden estar presentes en un sistema práctico. En esta sección se describirá brevemente cada bloque, y en capítulos posteriores se ampliará la información.

El codificador de Fuente (*source encoder*), trabaja con una o más señales analógicas, produciendo un tren de símbolos. Estos símbolos pueden ser binarios (1's y 0's) o pueden ser miembros de un conjunto de más de dos elementos. Con varios canales usados para comunicar a más de una fuente a la vez, el codificador de fuente debe de poseer un multiplexor. Como ya se ha dicho, la entrada al codificador de fuente es una señal en el tiempo, s(t), un sistema de comunicación de datos, en realidad, comienza con una señal digital (Por Ejemplo, la salida obtenida al presionar las teclas de una computadora).

A medida que la comunicación por medios eléctricos va sustituyendo al escrito, la seguridad para este medio se ha ido incrementando. Se debe de asegurar que solo el receptor interesado, entienda el mensaje, y solo el transmisor autorizado lo envié. Solo la "*Encriptación*" da tal seguridad. A medida que existan cada vez más transmisores y receptores sofisticados, y existan computadoras más grandes y veloces, el reto para la seguridad en las comunicaciones se incrementa.

El canal codificador provee otro tipo de seguridad diferente en la comunicación. Incrementa la eficiencia y/o decrementa los efectos de los errores de transmisión. Cuando

se introduce ruido en los canales de comunicación, existe la posibilidad de interpretar un símbolo por otro en el receptor. Se pueden reducir los efectos de este tipo de errores dándole al mensaje una estructura en forma de redundancia. En forma más simple, se requiere que se repita el mensaje. El proceso entero se conoce como corrección de error hacia adelante (*forward Error Correction*). La utilización de códigos correctos permiten una corrección de error, sin necesidad de que el receptor pregunte por mayor información al transmisor.

La salida del codificador de señal es una señal digital compuesta por símbolos. Por ejemplo, en un sistema binario, la salida será un tren de 1's y 0's. Un canal eléctrico transmite señales solo en forma de ondas eléctricas. Este es un punto importante. No hay que olvidar en pensar que una señal digital no puede ser transmitida sin formato. Por ejemplo, si se utiliza un canal de audio para transmitir el mensaje "10101", se tendrían que decir cinco palabras para crear una onda de audio. Cuando se dice la primera palabra "uno", se transmite una onda analógica, que corresponde a los sonidos particulares de esa palabra. Para este caso, se transmite una señal digital, utilizando ondas analógicas. Esto puede parecer como un retroceso en el progreso, y de hecho, lo es. Para transmitir la señal analógica, se estará remplazando a una señal digital, enviando esa señal digital utilizando ondas analógicas y convirtiendo las ondas analógicas a señales digitales en el receptor, posteriormente cambiando la señal digital a analógica. Este proceso posee algunas ventajas sobre el sistema puramente analógico, como la reducción de ciertos tipos de distorsión en un ambiente con mucho ruido.

Regresando ahora al modulador de portadora (*Carrier Modulation*), su propósito es el de producir ondas analógicas que corresponden a los símbolos discretos de la entrada.

Ya se ha dicho que la encriptación es el medio para obtener transmisiones seguras de receptores o transmisores no autorizados. En ese sentido, la encriptación produce una secuencia de símbolos que son claramente distinguibles solo por el receptor autorizado. También se provee de seguridad adicional mediante el uso de técnicas de *Spread Spectrum*. Uno de los propósitos del *Spread Spectrum* es el de prevenir que oyentes sin autorización distingan los símbolos, de tal forma que el oyente confunda la señal con ruido de bandas cercanas.

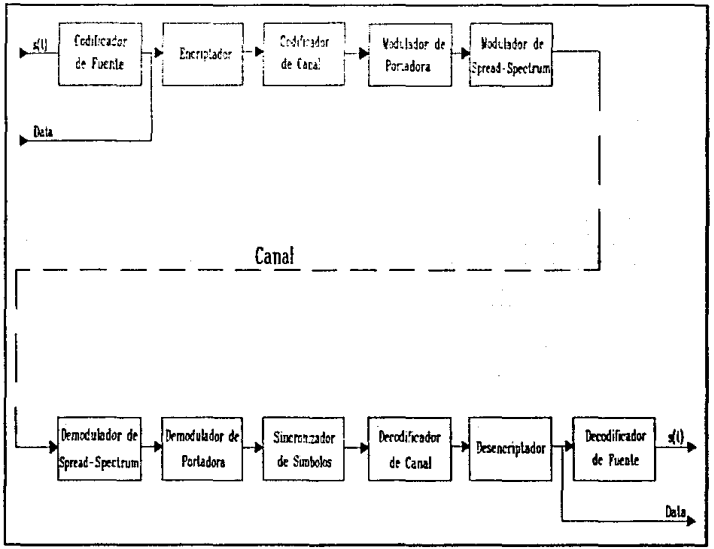


Figura 1.1

DIAGRAMA A BLOQUES TIPICO DE UN SISTEMA DIGITAL DE TRANSMISION Y RECEPCION.

Revisando la segunda parte de la Figura 1.1, se puede observar que el receptor es simplemente una imagen en espejo del transmisor. La única variación con respecto a los bloques, comparándolos uno a uno, corresponde al modulador de la portadora en el transmisor, que se reemplaza por dos bloques en el receptor: el demodulador de portadora y sincronizador de símbolos (*Symbol Synchronizer*). Una vez más, las ondas analógicas son reproducidas en el receptor, y es crítico que la señal total sea dividida correctamente en segmentos correspondientes a cada símbolo para cada mensaje. Esta división es la función del sincronizador.

1.3 COMPARACION ENTRE COMUNICACION ANALOGICA Y DIGITAL

Antes de analizar las ventajas de un sistema sobre el otro, se debe de poner en claro algunos conceptos básicos. El sistema de comunicación analógico, es el que transmite señales analógicas, estas señales en el tiempo, pueden ser continuas o tomadas en ciertos intervalos. Si las señales analógicas son muestreadas en tiempo, uno podría pensar como que los muestras son una lista de números, que serán transmitidos. Esta lista se considera como de números analógicos, la cual puede tomar un infinito número de valores con ciertos límites definidos. El sistema no es del todo digital todavía. Se le refiere como un sistema discreto en tiempo o sistema muestreado. Si los valores muestreados ahora son referidos a un conjunto discreto (por ejemplo, enteros), el sistema será digital. La clave de este concepto es que, los valores muestreados no pueden tomar cualquier valor fuera de un rango discreto.

Muchos sistemas son combinaciones híbridas, de sistemas digitales y analógicos. Por ejemplo, a medida que los ojos revisan esta hoja, el sistema psicológico esta operando como un receptor analógico, mientras que se buscan graduaciones de imágenes en cualquier parte de la hoja. Pero la forma básica de comunicación es digital, desde el momento que uno se programa mentalmente a mirar un número finito de señales -los alfanuméricos más un número limitado de símbolos griegos y matemáticos-. En un nivel superior, se buscan palabras del diccionario de comunicación -un conjunto posiblemente de entre 30,000 palabras-. Siguiendo más allá, si se vuelve a ver la frase anterior, "un conjunto posiblemente de entre 30,000 palabras", probablemente se reciba el mensaje, aunque alguna palabra no sea parte del diccionario. Una persona esta disponible a recibir

correctamente el mensaje, aunque la comunicación sea de tipo digital. De hecho, si la palabra "posiblemente" no fuera parte del diccionario, o se hubiese tenido un error en la recepción, el mensaje sería entendido.

Con esto, se pueden examinar algunas de las ventajas y desventajas de la comunicación digital, comparándola con la analógica, y mencionando solo las mayores ventajas y desventajas, se tiene:

- Ventajas de la Comunicación Digital:
    1. Los errores pueden ser corregidos.
    2. Es posible manipular la señal (por ejemplo, encriptarla).
    3. Es posible un rango dinámico mayor (diferencias entre el mayor y menor valor).

- Desventajas de la Comunicación Digital:
    1. Requieren un ancho de banda mayor que los establecidos como estandares.
    2. Necesitan de sincronización.

La mayor ventaja de la comunicación digital radica en la capacidad de corrección de errores que tiene, este concepto es básico. A lo largo de todas las posibles formas de ondas de las señales que se transmiten, el receptor puede reconocer un error cuando se presenta. En ocaciones, este reconocimiento es suficiente, en otras, el error es corregido automáticamente en el receptor.

La Figura 1.2, contrastan las diferencias entre un sistema analógico y un sistema digital. Hay que notar que en los sistemas analógicos, los amplificadores aparecen a lo largo del camino de la transmisión. Cada amplificador introduce cierta ganancia, pero sin embargo amplifica tanto a la señal como a cualquier ruido extra que exista a lo

largo del camino de transmisión.

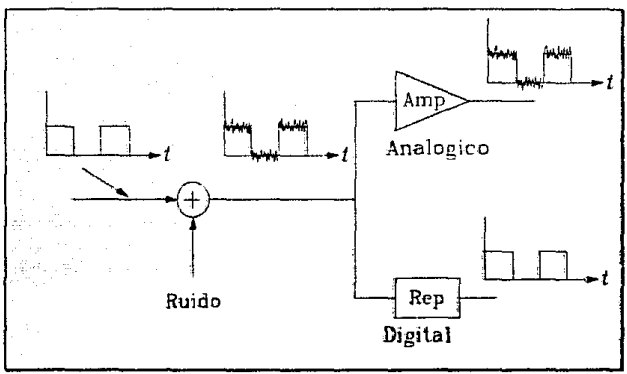


Figura 1.2

CONTRASTES ENTRE UN SISTEMA ANALOGICO Y UN SISTEMA DIGITAL

La parte inferior de la Figura 1.2, muestra un sistema de comunicación digital. Hay que notar que los amplificadores del sistema analógico son reemplazados por repetidores regenerativos. Los repetidores no solamente realizan la función de amplificación, sino también "limpian" la señal.

Se ha ilustrado, las dos señales usando señales de "banda base unipolar" (*Unipolar Baseband*). La entrada de la señal puede tomar solo uno de dos posibles valores, 0 o 1. Por lo que el repetidor solo decide cual de los dos valores es, en un cierto intervalo, luego reproduce el valor exacto para transmitirlo al siguiente paso en el canal.

La segunda ventaja de un sistema de comunicación digital, radica en el hecho que se esta trabajando con números pequeños en vez de ondas. Estos números pueden ser

manipulados por circuitos lógicos simples, y si es necesario, por un microprocesador. Las operaciones complejas pueden realizarse fácilmente en función de completar un proceso de la señal o por seguridad en la transmisión. Comparándola, las operaciones analógicas, necesitarían un hardware más complejo.

La tercera ventaja es el hecho de su rango dinámico. Tradicionalmente la superficie de grabación de los discos fonográficos, tienen un rango dinámico muy limitado, los sonidos muy fuertes requieren variaciones extremas en la superficie de las ranuras, y es muy difícil para la aguja seguir estas variaciones. Las grabaciones digital en cambio, no sufren por este problema, ya que todas las amplitudes de sus valores, o son muy chicas, o son muy grandes y son transmitidas usando el mismo conjunto limitado de señales.

Sin embargo, no todo es perfecto, existen desventajas en las comunicaciones digitales. Los sistemas digitales generalmente requieren más ancho de banda que los analógicos. Por ejemplo, un solo canal de voz transmitido usando un canal de AM, utiliza menos de 5K de ancho de banda, transmitiendo la misma señal usando técnicas digitales, se usará por lo menos 4 veces más.

Otra desventaja, en el sistema de comunicación digital, es de tener que poseer un sincronizador, ya que es importante para el sistema, conocer cuando empieza y acaba cada símbolo, y el poder asociar cada símbolo con la transmisión correcta. Algunas personas podrían argumentar, que los sistemas analógicos, en su forma más simple, son olvidados, es más, un solo canal de voz en AM, esta demasiado sobrado; la frecuencia de portadora o el ancho de banda, pueden ser reducidos y aún se seguirá entendiendo el mensaje.

1.4 ALGUNOS CODIGOS SIMPLES

Un código es un conjunto de reglas que asignan una palabra código a cada mensaje de un diccionario aceptable de mensajes. Las palabras código deben de consistir de símbolos de un alfabeto aceptable. En este trabajo se presenta especial atención a las palabras código consistentes de números binarios, y el diccionario de mensajes serán todas las letras del alfabeto con algunas funciones de control.

1.4.1 CODIGO BAUDOT

Este código fue uno de los primeros, y hoy día obsoleto, este código era utilizado en las máquinas teletipo. Este código aún encuentra limitadas aplicaciones con los radio amateurs y algunos sistemas de comunicación para sordos, como también para telegrafía. El código Baudot asigna un número binario de 5-bits a cada letra del alfabeto. Por lo que solo existen 32 distintos números binarios formados con 5-bits (ya que $2^5 = 32$), este código no provee mucha flexibilidad para trabajar con el. De hecho, las 26 letras mayúsculas del alfabeto, el cambio de línea y el retorno de carro dejan lugar a solo tres palabras clave para el resto de los símbolos, incluyendo a los números. Este código tan primitivo evita este inconveniente, proveyendolo de un cambio de instrucciones. Una palabra código toma todas las palabras siguientes en el modo de "cambio" o "figuras" (*shift mode*), mientras otra palabra clave regresa el sistema al modo de "letras". Substrayendo estas dos instrucciones de cambio, restan 30 posibles palabras en cada modo, o un total de 60 palabras en el diccionario. Esto es suficiente para el alfabeto en mayúsculas, 10 dígitos, los símbolos comunmente usados (como $, # y el %), el cambio de línea, el retorno de carro, y el

tradicionalmente Teletipo Bell. Este código se ilustra en la Tabla 1.1:

CODIGO BAUDOT

MODO DE LETRAS	MODO DE FIGURAS	PALABRA CODIGO
Blanco	Blanco	00000
E	3	00001
Cambio de Línea	Cambio de Línea	00010
A	-	00011
Espacio	Espacio	00100
S	Campana	00101
I	8	00110
U	7	00111
Retorno de Carro	Retorno de Carro	01000
D	$	01001
R	4	01010
J	'	01011
N	,	01100
F	!	01101
C	:	01110
K	(	01111
T	5	10000
Z	"	10001
L	)	10010
W	2	10011
H	#	10100
Y	6	10101
P	0	10110
Q	1	10111
O	9	11000
B	?	11001
G	&	11010
Cambiar al Modo de Figuras	Cambiar al Modo de Figuras	11011
M	.	11100
X	/	11101
V	;	11110
Cambiar al Modo de Letras	Cambiar al Modo de Letras	11111

TABLA 1.1

1.4.2 CODIGO ESTANDAR AMERICANO PARA INTERCAMBIO DE INFORMACION (ASCII)

Este código se ha vuelto el estandar para comunicaciones digitales de alfabeto de símbolos individuales. Este código es también utilizado para comunicaciones de pequeña distancia, como es la comunicación del teclado de la computadora al procesador. El código base consiste de palabras códigos de una longitud de 7-bits, lo

que provee un diccionario de 128 palabras ($2^7 = 128$). Esto es suficiente para el alfabeto completo (en mayúsculas y minúsculas), más un número razonable de dígitos de control. Normalmente se añade un octavo bit como bit de paridad o bit para detección de errores.

CODIGO ASCII

				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
b_1	b_2	b_3	b_4									
0	0	0	0		NUL	DLE	SP	0	@	P	'	p
0	0	0	1		BS	CAN	(	8	H	X	h	x
0	0	1	0		EOT	DC4	$	4	D	T	d	t
0	0	1	1		FF	FS	,	<	L	/	l	
0	1	0	0		STX	DC2	"	2	B	R	b	r
0	1	0	1		LF	SUB	*	:	J	Z	j	z
0	1	1	0		ACK	SYN	&	6	F	V	f	v
0	1	1	1		SO	RS	.	>	N	^	n	
1	0	0	0		SOH	DC1	!	1	A	Q	a	q
1	0	0	1		HT	EM	)	9	I	Y	i	y
1	0	1	0		ENQ	NAK	%	5	E	U	e	u
1	0	1	1		CR	GS	-	=	M	[	m	}
1	1	0	0		ETX	DC3	#	3	C	S	c	s
1	1	0	1		VT	ESC	+	;	K	]	k	{
1	1	1	0		BEL	ETB	'	7	G	W	g	w
1	1	1	1		SI	US	/	?	O		o	DEL

TABLA 1.2

Si se examina con cuidado esta tabla se observa que los códigos están arreglados de una forma muy sistemática. Por ejemplo, supóngase que exista alguna aplicación en donde no se requiera ningún símbolo de control o ninguna de las letras minúsculas, se puede observar que el resto de los símbolos se encuentra en las cuatro columnas en el centro. Examinándolo con cuidado se observa que se puede desechar el bit más significativo b_6, de cada palabra código, para cada uno de los 64 posibles mensajes.

Alternativamente, si no se necesitan símbolos de control ni letras mayúsculas, examinando la tabla, indicaría cual es el bit que sería desechado.

En la Tabla 1.3 se muestran algunos de los símbolos de control más utilizados:

Simbolos de Control

SOH	Start of Heading	LF	Line Feed
STX	Start of Text	VT	Vertical Tab
ETC	End of Text	FF	Form Feed
EOT	End of Transmission	CR	Carriage Return
BS	Backspace	CAN	Cancel
HT	Horizontal Tab	SUB	Substitute

TABLA 1.3

1.4.3 CODIGO SELECTRIC

Este es uno de los muchos códigos especializados utilizados frecuentemente en el pasado. La máquina de escribir Selectric fue el estandar de la industria mucho antes de las máquinas de escribir electrónicas. El código, históricamente hablando, es bastante interesante, ya que fue configurado para el mecanismo de operación de las máquinas de escribir. La máquina de escribir Selectric, y muchas impresoras similares, usan un código de 7-bits para controlar la posición en la impresión de la esfera. Esto permite 128 distintos símbolos, de los cuales solo 88 son utilizados. Esto parecería ser ineficiente, y de hecho, el código fue diseñado para el mecanismo de operación de la esfera de las máquinas de escribir. Uno de los bits en el código de control, cambia la operación (rota la esfera 180 grados). De los otros 6 bits, 4 controlan la rotación y los otros 2, la inclinación. En las máquinas de escribir Selectric puede definirse cualquier letra o símbolo, presionando una o más de las siete teclas. Los controles como el espacio y el de siguiente línea, se manejan completamente separados del código de símbolos.

				CODIGO SELECTRIC							
			T5	0	0	0	0	1	1	1	1
			T1	0	0	1	1	0	0	1	1
			T2	0	1	0	1	0	1	0	1
S	R2A	R2	R1								
0	0	0	0	-	b	w	9				
0	0	0	1	y	h	s	0	/	1	o	4
0	0	1	0								
0	0	1	1								
0	1	0	0	q	k	i	6	,	c	a	8
0	1	0	1	p	e	'	5	;	d	r	7
0	1	1	0	=	n	.	2	f	u	v	3
0	1	1	1	j	t	1/2	z	g	x	m	l
1	0	0	0		B	W	(				
1	0	0	1	Y	H	S	)	7	L	O	S
1	0	1	0								
1	0	1	1								
1	1	0	0	Q	K	I		,	C	A	*
1	1	0	1	P	E	"	%	:	D	R	&
1	1	1	0	+	N	.	@	F	U	V	#
1	1	1	1	J	T	1/4	Z	G	X	M	

TABLA 1.4

Del código de 7-bits se presentan cuatro que controlan 16 filas de la tabla y tres que controlan 8 columnas.

La conversión de un código a otro (por ejemplo de ASCII a Selectric) es trivial utilizando circuitos integrados. Tan solo requiere una tabla, la cual puede ser programada en una memoria de solo lectura (*ROM*).

1.5 CODIGOS ARITMETICOS Y MODELOS ESTADISTICOS

Muchos de los métodos de compresión de datos más comunes, utilizados hoy día, caen en dos campos: esquemas basados en diccionarios o modelos estadísticos. En el mundo de los micro-sistemas, técnicas de compresión basados en datos basados en diccionario son más populares, hasta el momento. Sin embargo, combinando un código aritmético con técnicas de modelado más poderosas, métodos estadísticos para compresión de datos, se pueden obtener mejores resultados. Aquí se presenta como combinar códigos aritméticos con métodos de modelado para obtener rangos de compresión impresionantes.

1.5.1 TERMINOS DE ENTENDIMIENTO

En general, la compresión de datos opera tomando "símbolos" de un "texto" de entrada y procesandolos y escribiendo "códigos" a un archivo comprimido. Para un mayor entendimiento, aquí se presentan los símbolos como bytes, pero pueden ser fácilmente tomados, como pixels, números de 80 bits de punto flotante o caracteres EBCDIC entre otros. Para ser efectivo un esquema de compresión de datos se necesita tener la facilidad para transformar el archivo comprimido en una copia idéntica del texto de entrada. Es útil si el archivo comprimido es menor al de entrada.

Los sistemas de compresión basados en diccionarios operan reemplazando grupos de símbolos del texto de entrada con un código de longitud arreglada o diseñada. Un ejemplo bien conocido de una técnica basada en diccionario, es la compresión LZW, la cual opera reemplazando cadenas de caracteres "strings" de longitud ilimitada con códigos que usualmente están entre el rango de 9 a 16 bits.

Los métodos estadísticos de compresión de datos toman una

aproximación totalmente diferente: operan codificando símbolos en cada instante de tiempo, estos símbolos son codificados en códigos de salida, la longitud en la cual varían se basa en la probabilidades de frecuencia de aparición de cada símbolo. A bajas probabilidades de los símbolos, se utilizan más bits para codificarlos, y a altas probabilidades, se utilizan menos bits.

En la práctica, la línea divisoria entre métodos estadísticos y de diccionarios no es tan distinta. Algunos esquemas no pueden ser puestos tan fácilmente en un campo o en el otro, y siempre existen esquemas híbridos, los cuales utilizan partes de ambas técnicas. Sin embargo, los métodos discutidos en esta sección, utilizan únicamente códigos aritméticos para realizar solo compresión estadística.

1.5.2 COMO TRABAJAN LOS CODIGOS ARITMETICOS

Solo en los últimos 10 años ha sido perfectamente demostrado el sustituir el código Huffman por un código aritmético. El código aritmético sobrepasa la idea de reemplazar un símbolo de entrada con un código específico (Código Huffman). En vez de esto, se toma un tren de símbolos (*stream*) de entrada y son reemplazados con un número de punto flotante sencillo en la salida.

La salida de un proceso de codificación aritmética es un número sencillo menor a 1 y mayor o igual a 0. Este sencillo número puede ser igualmente decodificado para crear el tren exacto de símbolos para su construcción. Para construir el número de salida, el símbolo a codificar tiene un conjunto de probabilidades asignadas a él. Por ejemplo, si se quisiera codificar el mensaje aleatorio "random" JUAN PEREZ, se tendría una distribución de probabilidades como se muestra en la Tabla 1.5.

TABLA 1.5

CARACTER	PROBABILIDAD
Espacio	1/10
A	1/10
E	2/10
J	1/10
N	1/10
P	1/10
R	1/10
U	1/10
Z	1/10

Una vez que las probabilidades de los caracteres son conocidas, se les necesita asignar un rango a lo largo de la "línea de probabilidad", que normalmente va de 0 a 1. No importa cual carácter sea asignado a que segmento, a medida que sea hecho de la misma forma tanto para el codificador y decodificador. Los 9 caracteres utilizados en el ejemplo se verán como en la Tabla 1.6.

TABLA 1.6

CARACTER	PROBABILIDAD	RANGO
Espacio	1/10	0.00 - 0.10
A	1/10	0.10 - 0.20
E	2/10	0.20 - 0.40
J	1/10	0.40 - 0.50
N	1/10	0.50 - 0.60
P	1/10	0.60 - 0.70
R	1/10	0.70 - 0.80
U	1/10	0.80 - 0.90
Z	1/10	0.90 - 1.00

Cada carácter es asignado a una porción entre 0 y 1 que corresponde a su probabilidad de aparición. Hay que notar que todos los caracteres abarcan todo el rango, sin incluir al mayor número. Así es que la letra Z en realidad va del rango de 0.90 - 0.9999.....

La parte más significativa de un mensaje en código aritmético pertenece al primer símbolo a codificar. Cuando codificamos el mensaje JUAN PEREZ, el primer símbolo es la J. Para que el primer carácter sea decodificado correctamente, el

mensaje en código final deberá ser un número mayor o igual que 0.40 y menor que 0.50. Para codificar éste número, debemos mantenernos en el rango en el cual puede caer el número. Así es que después de que el primer carácter es codificado, el rango inferior es 0.40 y el superior es 0.50.

Después de que el primer carácter es codificado, sabemos que el rango para nuestro número de salida se encuentra entre el mayor y menor número. Durante el resto del proceso de codificación, cada símbolo nuevo a ser codificado caerá entre las restricciones de nuestro número de salida. El siguiente número a ser codificado, U, esta entre los rangos de 0.80 y 0.90. Si este fuera el primer número de nuestro mensaje, pondríamos estos valores como los rangos superior e inferior. Pero U es el segundo carácter, así es que a U le corresponden los rangos de entre 0.80 a 0.90 en el nuevo subrango de entre 0.40 y 0.50. Esto significa que el nuevo número a codificar tendrá que caer en alguna parte de entre el 80% y 90% del rango ya establecido. Aplicando esta lógica, nuestro nuevo número estará restringido entre el rango de 0.48 y 0.49. El algoritmo para completar cualquier mensaje de cualquier longitud esta dado por el Esquema 1.1. La Tabla 1.7, muestra este proceso llegando a la conclusión del mensaje escogido. Así es que el valor menor final es 0.4815063476, codificando el mensaje JUAN PEREZ, utilizando el presente esquema de codificación.

ESQUEMA 1.1

```
Set Menor := 0.0;
Set Mayor := 1.0;
While (existan símbolos de entrada) do
     get (un símbolo de entrada);
     rango_cod := Mayor - Menor;
     Mayor := Menor + rango_cod * rango_May(símbolo)
     Menor := Menor + rango_cod * rango_Men(símbolo)
End While
Resultado := Menor.
```

TABLA 1.7

NUEVO CARACTER	VALOR MENOR	VALOR MAYOR
	0.0	1.0
J	0.4	0.5
U	0.48	0.49
A	0.481	0.482
N	0.4815	0.4816
Espacio	0.58150	0.48151
P	0.481506	0.481507
E	0.4815062	0.4815064
R	0.48150634	0.48150636
E	0.481506344	0.481506348
Z	0.4815063476	0.4815063480

Dado este esquema de codificación, es relativamente fácil ver como operara el proceso de decodificación. Se encuentra el primer símbolo del mensaje, revisando que rango le pertenece del espacio del código. A causa de que 0.4815063476 cae entre 0.4 y 0.5, se sabe que el primer carácter es una J. Así es que removemos la J del número codificado. También conocemos el menor y mayor rango de J, así es que removemos sus efectos revirtiendo el proceso que la puso. Primero, se substrae el menor valor de J del número 0.4815063476, lo que da por resultado 0.0815063467. Luego se divide por el rango de J, que es 0.1, y se obtiene 0.815063467. Ahora se calcula en donde cae el número, y se observa que esta en el rango de la letra U. Así se continua hasta que ya no existan números.

El algoritmo para decodificar los siguientes números se ve como en el Esquema 1.2. Hay que notar que se han ignorado algunos de los inconvenientes, como es el problema de que ya

no existan más símbolos a la izquierda para decodificar. Esto se puede manejar, codificando un símbolo especial de EOF (End Of File), o llevando la cuenta de la longitud del tren a lo largo del mensaje codificado. La decodificación del mensaje JUAN PEREZ, se verá como se muestra en la Tabla 1.8.

ESQUEMA 1.2

```
get (número_codificado)
Do
  find (símbolo que se encuentre en el rango codificado)
  output (símbolo encontrado)
  rango := men_valor(símbolo) - may_valor(símbolo)
  número_codificado := número_codificado - men_valor(símbolo)
  número_codificado := número_codificado / rango
Until (no existan más símbolos)
```

TABLA 1.8

NUMERO CODIF.	SIMB. SALIDA	MENOR	MAYOR	RANGO
0.4815063474	J	0.4	0.5	0.1
0.815063476	U	0.8	0.9	0.1
0.15063476	A	0.1	0.2	0.1
0.5063476	N	0.5	0.6	0.1
0.063476	Espacio	0.0	0.1	0.1
0.63476	P	0.6	0.7	0.1
0.3476	E	0.2	0.4	0.2
0.738	R	0.7	0.8	0.1
0.38	E	0.2	0.4	0.2
0.9	Z	0.9	1.0	0.1
0.0				

En suma, el proceso de codificación consiste simplemente en estrechar el rango de números posibles con cada nuevo símbolo. El nuevo rango es proporcional a una probabilidad predefinida a cada símbolo. El proceso de decodificación es inverso: el rango es expandido en proporción a la probabilidad de cada símbolo extraído.

1.5.3 METODOS PRACTICOS

El proceso de codificación y decodificación de un tren de símbolos, usando códigos aritméticos no es tan complicado. A primera vista, parece ser completamente impráctico. Muchas computadoras soportan números de punto flotante arriba de 80 bits. ¿Esto puede implicar que se tiene que volver a empezar después de codificar de 10 a 15 símbolos? ¿Se requiere un procesador extra de punto flotante? ¿Podrían, máquinas con diferentes procesadores de punto flotante, comunicarse utilizando códigos aritméticos?

Mientras más se analiza esto, uno se da cuenta de que la mejor opción es utilizar códigos aritméticos utilizando procesadores matemáticos estandars de 16 y 32 bits. El procesador matemáticos de punto flotante no es indispensable, sin embargo puede ayudar. En vez de esto, se utiliza un incremento en los esquemas de transmisión en donde la dimensión arreglada de estados enteros de variables reciben nuevos bits a la entrada y los regresa cambiados a la salida, formando un simple número, el cual puede ser tan largo como el número de bits disponibles en el almacenamiento medio de la computadora.

En la sección previa, se ha mostrado como trabaja el algoritmo, manteniendo un número entre un rango (los números superior e inferior posibles). Cuando empieza el algoritmo, el número inferior es puesto en 0.0 y el superior en 1.0. Para trabajar con matemáticas enteras, debemos cambiar el 1.0 por 0.999 o a 0.11111 en forma binaria.

Para guardar estos números en registros enteros, primero debemos de darles una justificación, la parte decimal se debe encontrar en el lado izquierdo de la palabra. Después cargamos tantos valores iniciales altos y bajos como el tamaño de la

palabra que estemos trabajando. Una realización utilizando 16 bits, es que el valor alto sea de 0xFFFF, y el menor 0. Sabemos que el valor mayor continua con FF's indefinidamente, y el menor continua con 0's, así es que se puede cambiar esos bits extra en cualquier momento o cuando sean necesitados.

Si imaginamos el ejemplo pasado de JUAN PEREZ, en un registro de 5 dígitos, el equivalente decimal de nuestro arreglo (*setup*) se verá como en el Esquema 1.3(a). Para encontrar el rango de nuestros nuevos números, necesitamos aplicar el algoritmo de codificación de la sección anterior. Primero se calcula el rango entre los valores superior e inferior. La diferencia entre los dos registros será 100,000 no 99,999, porque asumiendo que el registro mayor tenga un infinito número de 9's sumados a él, se necesita aumentar la diferencia de cálculo. Se calcula el nuevo valor superior, utilizando el algoritmo de la sección anterior: mayor = menor + rango_may(símbolo).

En este caso, el rango mayor es 0.5, el cual da un valor mayor de 50,000. Antes de guardar este valor, necesitamos decrementarlo (debido a los dígitos asignados al valor del entero), así es que el nuevo valor mayor es 49,999. El cálculo del menor valor sigue el mismo procedimiento, con un resultado final de 40,000. El mayor y menor valor ahora se ven como en el Esquema 1.3(b).

Hasta este punto, los dígitos más significativos del valor mayor y del valor menor, ya están mencionados. Debido a la naturaleza del algoritmo, el mayor y menor valor pueden continuar acercandose entre sí, sin llegar a tocarse en ninguna ocasión. Aún más, una vez que los dígitos más significativos son marcados, ya no cambiaran. Podemos sacar ese dígito como el primer dígito de nuestro número codificado. Esto se hace cambiando ambos números, el mayor y menor, por un dígito, y cambiando a un 9 el último dígito del mayor valor.

A medida que este proceso continua, el mayor y menor valor continúan acercandose mutuamente, luego, cambiando los dígitos en palabras codificados. El proceso para el mensaje JUAN PEREZ, se observa como en el Esquema 1.3(c). Hay que notar que después de que todas las letras han sido acomodadas, es necesario cambiar dos dígitos, tanto del mayor y del menor valor, para terminar la palabra de salida.

ESQUEMA 1.3

(a)

Mayor : 99999
Menor : 00000

(b)

Mayor : 49999 (999...)
Menor : 40000 (000...)

(c)

	MAYOR	MENOR	RANGO	SALIDA ACUMULADA
Estado Inicial	99999	00000	100000	
Codificar J(0.4-0.5)	49999	40000		
Cambiar el 4	99999	00000	100000	0.4
Codificar U(0.8-0.9)	89999	80000		0.4
Cambiar el 8	99999	00000	100000	0.48
Codificar A(0.1-0.2)	19999	10000		0.48
Cambiar el 1	99999	00000	100000	0.481
Codificar N(0.5-0.6)	59999	50000		0.481
Cambiar el 5	99999	00000	100000	0.4815
Codificar Espacio(0-0.1)	09999	00000		0.4815
Cambiar el 0	99999	00000	100000	0.48150
Codificar P(0.6-0.7)	69999	60000		0.48150
Cambiar el 6	99999	00000	100000	0.481506
Codificar E(0.2-0.4)	39999	20000	20000	0.481506
Codificar R(0.7-0.8)	73999	70000		0.481506
Cambiar el 7				

1.5.4 UNA COMPLICACION

El esquema 1.3 mostrado trabaja muy bien para codificar un mensaje por una serie de incrementos. Existen suficientes retenciones durante los cálculos con enteros de doble precisión, para asegurar que el mensaje será codificado

correctamente. Sin embargo, existe la posibilidad de una pérdida por precisión bajo ciertas circunstancias.

En el proceso de codificar una palabra que tiene un "string" de 0's o 9's en él, el mayor y menor valor convergen lentamente en un valor, pero los dígitos más significativos no se tocaran inmediatamente. Por ejemplo, el valor mayor y menor podrían verse como en el Esquema 1.4(a). Hasta este punto, el rango calculado será solamente de un solo dígito de longitud, lo cual significa que la palabra de salida no tendrá la suficiente precisión para ser codificada. Aún peor, después de algunas iteraciones, el mayor y menor valor podría verse como en el Esquema 1.4(b).

Hasta este punto, los valores son permanentemente alargados. El rango entre el mayor y menor valor se ha vuelto tan pequeño que cualquier cálculo dará siempre el mismo valor. Aún cuando, los dígitos más significativos de ambas palabras no son iguales, el algoritmo no cambiara el dígito.

En el algoritmo original, si el dígito más significativo del valor mayor y del valor menor marcados, los cambiamos, para prevenir el subdesbordamiento de memoria (*underflow*), se necesita aplicar una segunda prueba después de marcar los dígitos mientras estén en números adyacentes. Si el valor mayor y menor están a parte, se debe probar si el segundo dígito más significativo del mayor valor es un 0, y el segundo dígito del menor valor es un 9. Si es así, estamos en camino de provocar un subdesbordamiento de memoria y se necesita tomar otras acciones.

Cuando ocurre el subdesbordamiento de memoria es desastroso, y se sale adelante con una simple operación de cambio diferencial. En vez de cambiar el dígito más significativo fuera de la palabra, solamente se borra el segundo dígito del mayor y menor valor y se cambia el resto de

los dígitos a la izquierda para dejar espacio libre. Los dígitos más significativos permanecen en su lugar. Es entonces donde se debe mandar un contador de subdesbordamiento de memoria, para recordar que hemos tirado un dígito, y no se está seguro de donde va a terminar en 0 o en 9. Esta operación es mostrada en el Esquema 1.4(c).

Después de recalcular cada operación, si los dígitos más significativos no son marcados, se pueden checar los dígitos en el subdesbordamiento de memoria. Si están presentes, se cambian y se incrementa el contador.

Cuando el dígito más significativo converge finalmente a un simple valor, primero se saca el valor. Luego, se calculan todos los dígitos descartados por subdesbordamiento de memoria. Los dígitos del subdesbordamiento de memoria serán 9's o 0's, dependiendo en donde convergen los valores mayores y menores del mayor y menor valor. En una implementación de este algoritmo, el contador de subdesbordamiento de memoria tendería a contar cuantos 1's y 0's hay para descartarlos.

ESQUEMA 1.4

(a)

MAYOR: 700004
MENOR: 699995

(b)

MAYOR: 700000
MENOR: 699999

(c)

	Antes	**Después**
MAYOR	40344	43449
MENOR	39810	38100
SUBDESBORDAMIENTO	0	1

1.5.5 DECODIFICADOR

En el proceso "ideal" de decodificación, se tiene todo el número de entrada para trabajar, así es que el algoritmo tiene que "dividir el número codificado por la probabilidad del símbolo". En la práctica, no se puede realizar la operación tal cual, ya que el número podría ser de billones de bits de longitud. Así como en el proceso de codificación, la decodificación puede operar usando números enteros de 16 o 32 bits para los cálculos.

En vez de mantener dos números, el mayor y el menor, el decodificador tiene que mantener tres números enteros. Los primeros dos, el mayor y el menor, corresponden exactamente a el valor mayor y menor del codificador. El valor del código siempre estará entre el mayor y menor valor. Mientras más se acerquen, nuevos cambios en operaciones se llevaran a cabo, y el mayor y menor valor se removerán del código.

El mayor y menor valor en el decodificador corresponden exactamente a los valores (mayor y menor) usados por el codificador. Estos serán actualizados después de cada símbolo, así como fueron usados en el codificador, y deberán tener exactamente el mismo valor. Realizando la misma prueba de comparación en el dígito superior del mayor y menor valor, el decodificador sabe cuando cambiar a un nuevo dígito en el código de entrada. La misma prueba de subdesbordamiento de memoria es realizada óptimamente en circuito cerrado con el codificador.

Con el algoritmo ideal, es posible determinar donde se encuentran los símbolos codificados, simplemente encontrando las probabilidades que posee el valor actual del código. En el algoritmo entero matemático, las cosas son un poco más complicadas: la probabilidad escalar esta determinada por la

diferencia entre el mayor y menor valor, así es que en vez de que el rango se encuentre entre 0.0 y 1.0, estará entre dos posibles contadores enteros de 16 bits. La probabilidad será determinada por donde el presente valor del código se encuentre dentro del rango. Si uno fuese a dividir el menor valor entre el valor mayor del menor + 1, se obtendría la probabilidad para el símbolo actual.

Tempranamente, se percibe como cada carácter "pertenece" a una probabilidad en la escala de 0.0 a 1.0. Para realizar un código aritmético utilizando números enteros, su valor es calculado, restado un valor menor y un valor mayor, contados a lo largo del rango entero desde 0 hasta el contador máximo.

1.5.6 MODELANDO

La necesidad de predecir la probabilidad de los símbolos en los datos de entrada es inherente a la naturaleza del código aritmético. El principio de este tipo de códigos es el de reducir el número de bits necesarios para codificar un carácter con respecto a sus probabilidades de aparición se van incrementando. Así es que, sí por ejemplo, la letra "E" representa el 25% de los datos de entrada, solo tomara 2 bits para codificarse. Si la letra "Z" representara solo el 0.1% de los datos de entrada, tomara hasta 10 bits para codificarse. Si el modelo no esta generando las probabilidades debidamente, podría tomar 10 bits para representar la letra "E" y solo 2 para representar la letra "Z" causando un expansión de datos en vez de compresión.

La segunda condición es que el modelo necesita hacer las predicciones que se derivan de una distribución uniforme. El mejor modelo es hacer estas predicciones, ya que serán mejores los rangos de compresión. Por ejemplo, un modelo podría crearse asignando de 256 posibilidades una distribución

uniforme, por lo que cada uno tendría una probabilidad de 1/256. Este modelo crearía un archivo de salida exactamente del mismo tamaño que el de entrada, ya que cada símbolo se llevaría los mismos 8 bits para ser codificado. Solamente si se encuentran las probabilidades correctas, derivadas de una distribución uniforme, se puede reducir el número de bits, para ser comprimido. Por supuesto, el incremento de las probabilidades debe de reflejar la realidad, así como se prescribio como primera condición.

Se puede ver que la probabilidad de ocurrencia de un símbolo dado dentro de un tren, esta arreglado, pero no es del todo cierto. Dependiendo del modelo usado, la probabilidad de un carácter puede cambiar en un bit. Por ejemplo, cuando se comprime un programa en lenguaje C, la probabilidad de un carácter de cambio de línea, puede ser de 1/40. Esta probabilidad puede ser determinada, examinando todo el texto y dividiendo el número de ocurrencia del carácter por el número total de caracteres. Pero si se utiliza una técnica de modelado, que busca un solo carácter, la probabilidad cambiara bastante. En ese caso, si el primer carácter es un "}", la probabilidad de una línea se eleva a ½. Esta técnica improvisada de modelado llega a dar mejores resultados de compresión, aún si ambos modelos se les calcularan sus probabilidades.

1.5.7 MODELO DE CONTEXTO FINITO

El tipo de modelado que se presenta es referido como de "contexto finito". El cual se basa en una idea sencilla, la cual dice: la probabilidad de cada símbolo se calcula basado en el contexto de cuando aparece cada símbolo. En todos los ejemplos antes mencionados, el contexto consiste en nada más que símbolos. El "orden" del modelo se refiere al número de símbolos previos que hacen crecer el contexto.

El modelo más simple de contexto finito será un modelo de orden-0, en el cual, la probabilidad de cada símbolo es independiente de cualquier símbolo anterior. En función de realizar este modelo, se necesita una tabla sencilla, conteniendo la frecuencia para cada símbolo que pueda ser encontrado en el tren de entrada. Para un modelo de orden-1, se deben de tener 256 tablas de valores diferentes de frecuencia, porque se necesita un grupo separado de conteo para cada contexto posible. Aún más, un modelo de orden-2, necesita tener 65,536 tablas diferentes de contexto.

1.5.8 MODELO ADAPTABLE

Siguiendo esta lógica, si aumenta el orden del modelo, los rangos de compresión tienden a aumentar también. Por ejemplo, la probabilidad de aparición de la letra "U" en este capítulo, puede que sea de alrededor de un 5%, pero si el carácter anterior es una letra "Q", la probabilidad se eleva a un 95%. Siendo capaces de predecir los caracteres con altas probabilidades, se reduce el número de bits necesarios, por lo que contextos más largos nos da la posibilidad de hacer mejores predicciones.

Desafortunadamente, a medida que el modelo se incrementa linealmente, la memoria consumida por el modelo crece exponencialmente. Con un modelo de orden-0, el espacio consumido por las estadísticas puede ser tan pequeño como 256 bytes. Pero si el modelo crece a orden 2 o 3, aún el model más sencillo consumirá cientos de Kilobytes.

Un modo convencional de compresión de datos es el de leer todos los símbolos y después realizar el modelo estadístico. Una segunda lectura es realizada para codificar los datos. Las estadísticas son entonces utilizadas para comprimir los datos,

de modo que el decodificador tenga una copia de ellos. Esto tendrá problemas muy serios, si la estadística del modelo consume mayor espacio que los datos a ser comprimidos.

La solución es el de realizar una compresión "adaptable", en la cual el compresor y el decompresor empiecen con el mismo modelo. El compresor codifica un símbolo usando el modelo existente, luego actualiza el modelo para el siguiente símbolo. El decompresor, de igual manera, decodifica un símbolo usando el modelo existente, y con ello actualiza el modelo. A lo largo, el algoritmo, actualiza el modelo, y opera de forma idéntica en el compresor y decompresor, el proceso opera perfectamente sin tener que pasar a una tabla estadística para el compresor y decompresor.

El lugar donde se sufre con compresión adaptable es en el costo de actualizar el modelo. Cuando se actualiza el conteo para un símbolo particular utilizando un código aritmético, el código actualizado tiene el potencial de costo para actualizar los conteos acumulativos para todos los otros símbolos también; teniendo que codificar con un promedio de 128 operaciones aritméticas para cada símbolo codificado o decodificado.

Debido al alto costo tanto en memoria como en tiempo de procesador central (*CPU*), modelos adaptables de mayor orden, se han vuelto prácticos solo en los últimos 10 años. Esto es un tanto irónico, si el costo de espacio de disco y la memoria descienden, así también descenderá el costo de compresión de datos. A medida que los costos siguen descendiendo, será posible implementar programas más eficaces que sean prácticos.

1.5.9 MODELOS DE MAYOR ORDEN

Un modelo de orden 0, no toma en consideración los símbolos previos de un archivo texto, cuando calcula las probabilidades para el símbolo. Mirando el carácter previo en el archivo texto, o el "contexto", se pueden predecir los siguientes símbolos.

Cuando se usan modelos de orden 2 o 3, un problema es que, en un modelo de orden arreglado, cada carácter debe tener una probabilidad finita diferente de 0, para poder ser codificada, para cuando esta aparezca. Esto es realizado mandando un código especial de "Escape". Para el contexto previo o REQ, se puede mandar el conteo del código de Escape a 1, y todos los demás símbolos a un conteo de 0. Para la primera vez, el carácter "U" seguido de un REQ, tendrá que emitir un código de Escape, seguido por el código de la "U" en un contexto diferente. Durante la actualización del modelo, se sigue que: se puede incrementar el conteo para la "U" en el contexto del REQ a 1, dando una probabilidad de ½. La siguiente vez que aparezca, sera codificado en 1 bit, con un incremento de probabilidad, el número de bits ira disminuyendo para cada aparición.

La pregunta obvia es: ¿Qué se utiliza como contexto "*fallback*" después de emitir un código de Escape? En el Modelo-2, si en el contexto de orden-3 se genera un código de Escape, el siguiente contexto será de orden-2. Esto significa que la primera vez que el contexto REQ es utilizado, una "U" sera codificada y un código de Escape será generado. Siguiendo esto, el algoritmo del Modelo-2, regresa al modelo de orden-2 y trata de codificar el carácter "U", usando el contexto EQ. Esto continua en descenso hasta el contexto de orden-0. Si el código de Escape sigue siendo generado en el orden-0, se tendrá que ir a un contexto especial de orden (-1). El

contexto -1 es puesto para inicializaciones para tener un conteo de 1 para cada símbolo. Este nunca es actualizado, por lo que se garantiza que estará disponible para codificar cada símbolo.

2. CAPITULO I "CODIFICACION"

2.1 CODIFICADOR DE FUENTE

El diagrama a bloques de la Figura 1.1, contiene un bloque denominado "Codificador de Fuente", el cual no es necesario, si se empieza con una señal digital, como ocurre con la salida de una computadora. Es necesario cuando la señal original viene en forma analógica. Esto es, ya que nuestro sistema de comunicación, solo puede transmitir señales digitales, es necesario, convertir nuestro sistema de comunicación en un sistema digital. Esta transformación es el trabajo del "Codificador de Fuente".

Se empezara hablando del "*muestreo*", la cual es una técnica para transformar del eje continuo del tiempo a un número de puntos discretos.

2.1.1 MUESTREO

Si se requiere convertir una señal de información analógica a una señal digital, se requieren dos tipos distintos de operaciones. Primero, el eje del tiempo es discretizado; así la lista de resultados debe de reformatearse, de tal manera que cada número sea escogido de un alfabeto discreto.

EL TEOREMA DE MUESTREO.- El cambio del eje continuo del tiempo a un eje discreto es acompañado por un muestreo en tiempo. El *teorema de muestreo*, conocido también como teorema de Shannon o teorema de Kotelnikov, dice que, si la transformada de Fourier de una función del tiempo es cero para $f>f_m$ y los valores de la función del tiempo son conocidos para $t=nT_s$ (para todos los intervalos de n), entonces

la función del tiempo es conocida para todos los valores de t con la condición de las muestras estén lo suficientemente cerca una de otra. Esta restricción es que $T_s < 1/2f_m$. Realizado de esta manera, $s(t)$, solo se puede determinar de una secuencia de valores equidistantes en tiempo. El límite superior de T_s, $1/2f_m$, es conocido como el *rango de muestreo de Nyquist*.

El límite superior de T_s se puede expresar de la misma manera para obtener el recíproco de T_s obteniendo así la frecuencia de muestreo, expresada por $f_s=1/T_s$. Entonces las restricciones son:

$$f_S > 2f_m$$

Esto es, la frecuencia de muestreo debe ser al menos, el doble de la frecuencia máxima de la señal a ser muestreada. Por ejemplo, si una señal de voz tiene 3kHz como frecuencia máxima, el muestreo debe ser por lo menos de 6,000 muestras por segundo, para cumplir con las condiciones del teorema de muestreo.

Antes de ir más adelante, se puede observar que el espacio entre los puntos de muestreo es inversamente relativo a la frecuencia máxima, f_m. Si la función varia con rapidez, la cercanía de las muestras permitirán reproducir esa función nuevamente.

Existen al menos tres aproximaciones para probar el teorema. En este trabajo solo se presenta una prueba, la cual solo requiere de un conocimiento básico en teoría de modulación de amplitud.

2.1.2 PRUEBA AL TEOREMA DE MUESTREO

La figura 2.1 muestra un tren de pulsos multiplicando a la señal original, $s(t)$. Si el tren de pulsos consistiera de pulsos muy angostos, uno podría decir, que la salida de la multiplicación es una versión muestreada de la señal original. En realidad, la salida depende no solo de los valores muestreados de la entrada, sino también de un pequeño rango de valores alrededor cada punto de muestreo. Esta teoría no requiere estos valores extra, los cuales representan información adicional. Sin embargo, los sistemas normalmente muestrean cerca de un pequeño rango de tiempo alrededor de los valores actuales de muestreo. A medida que se prueba el teorema, resultara obvio que la función a multiplicar no necesita de pulsos cuadrados perfectos. De hecho, la función puede ser cualquier función periódica.

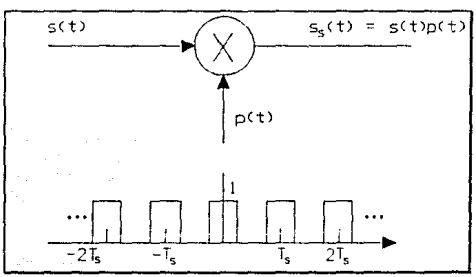


FIGURA 2.1

Señal Original Multiplicada por un Tren de Pulsos

Multiplicando $s(t)$ por $p(t)$ como se muestra en la Figura 2.1, es en realidad una forma de entrada en tiempo. Esto puede ser visto como una apertura y cierre de una compuerta o un interruptor (switch).

La meta es el mostrar que la señal original se puede recobrar de la señal muestreada, $s_s(t)$. Esto se logra, primeramente observando la Transformada de Fourier de $s_s(t)$. El teorema de muestreo requiere que se asuma que $s(t)$ no posee energía arriba de la frecuencia de f_m. La Transformada de Fourier de $s(t)$, $S(f)$, corta todo a la frecuencia f_m. La Figura 2.2 muestra en forma representativa el corte de la Transformada de Fourier.

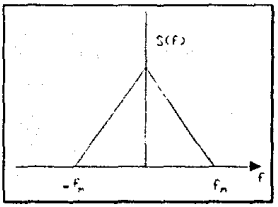


FIGURA 2.2

Frecuencias de Corte Obtenida con la Transformada de Fourier

Puesto que el tren de pulso a ser multiplicado se asumió periódico, puede ser expandido en una serie de Fourier. La función p(t) es una función par, por lo que se puede expresar como una serie trigonométrica utilizando solamente los términos del coseno. Por lo que:

$$s_s(t) = s(t)\,p(t)$$

$$s(t)\left[a_0 + \sum_{n=1}^{\infty} a_n \cdot 2\pi n f_s t\right] \qquad 2.1$$

$$a_0 s(t) + \sum_{n=1}^{\infty} a_n \cdot s(t) \cdot \cos\left(2\pi n f_s t\right)$$

donde:

$$f_s = 1/T_s$$

La meta es separar el primer término, el cual es proporcional al original $s(t)$. Entonces, por los efectos de multiplicar por una constante se puede amplificar o atenuar la señal.

Cada término de la sumatoria de la ecuación 2.1 representa una señal de AM, donde la información de la señal es $s(t)$ y la frecuencia de portadora es nf_s. La frecuencia contenida se encuentra en el centro de la frecuencia de portadora. Se puede ahora encontrar la Transformada de Fourier $s_s(t)$; ilustrada en la Figura 2.3. La forma central en el origen es la transformada de $a_0 s(t)$, y las diferentes versiones representan las transformaciones de varios términos de modulación. Se observa que los diferentes términos no se encimaran con los otros (overlap) para una $f_s > 2f_m$. Pero es muy simple la condición dada en el teorema de muestreo. Por lo que diferentes términos ocupan diferentes bandas de frecuencia, pueden ser separados utilizando filtros lineales. Un filtro paso bajos con un frecuencia de corte de f_m puede ser utilizado para recobrar el término $a_0 s(t)$.

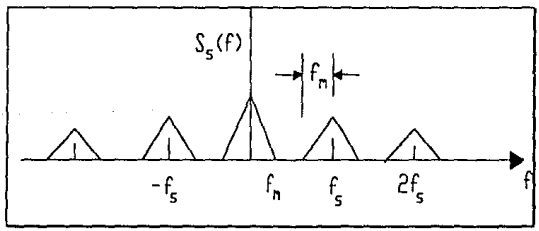


FIGURA 2.3

Frecuencia Portadora al Centro y otros Terminos de Modulación

2.1.3 ERRORES EN EL MUESTREO

El teorema de muestreo indica que $s(t)$ se puede recobrar de las muestras. Si se realiza un muestreo físicamente, resultan errores de tres fuentes. *Errores de Redondeo* ocurren cuando los valores múltiples del muestreo tienen que redondearse en un sistema de comunicación. Los *Errores por Truncamiento* ocurren si el muestreo se realiza en un tiempo limitado. Esto es, el teorema de muestreo requiere que las muestras sean tomadas para todo momento en un intervalo infinito, y cada muestra es utilizada para reconstruir el valor de la señal original en un momento en particular. En un sistema real, la señal es tomada para un intervalo de tiempo. Se puede definir un error de función, como la diferencia entre la función reconstruida y la función original, permitiendo levantar los límites dependiendo de la magnitud de este error de función. Estos límites involucran sumatorias en función del tiempo en que no se toman muestras.

Un tercer error resulta si la velocidad de muestreo no es lo suficientemente alta. Esta situación puede ser intencional o accidental. Por ejemplo, si la señal en tiempo original posee, una transformada de Fourier, con una aproximación asintótica a cero y con frecuencias crecientes, se define a la frecuencia máxima más allá de la señal que se requiera. En función de minimizar este problema, la señal de entrada es normalmente filtrada por un filtro pasa bajos antes de ser muestreada. Por otro lado, se puede diseñar un sistema con un rango de muestreo lo suficientemente grande para anticipar una señal de alta frecuencia inesperada. En este caso, el error causado por muestreo demasiado lento, se le conoce como "*aliasing*", el nombre fue deducido de le hecho que las altas frecuencias se disfracen en forma de frecuencia menores. Este es el mismo fenómeno que ocurre si un dispositivo de rotación es visto como una secuencia de

tramas individuales. Si el dispositivo de rotación incrementa su velocidad, un punto es alcanzado, donde la velocidad angular percibida empieza a decrecer. Eventualmente, el dispositivo alcanza una velocidad donde aparenta estar detenido. Incrementos futuros, en la velocidad, aparentan que el rotor se mueve en dirección contraria.

El análisis de éste tipo de errores es más fácilmente realizado en el dominio de la frecuencia. Antes de hacer esto, la Figura 2.4 se ilustra en función del tiempo. La Figura muestra una señal senoidal a una frecuencia de 3 Hz. Supongase que se muestra esta señal senoidal a 4 muestras por segundo. El teorema de muestreo nos dice que la velocidad mímina de muestreo para recobrar la señal es de 6 muestras por segundo, para este caso, así es que con solo 4 muestras por segundo, no es sufiencietemente rápido el muestreo. Estas 4 muestras por segundo son la misma cantidad de muestras que resultarian de una señal senoidal de 1 Hz, como también lo muestra la grafica. La señal de 3 Hz esta disfrazada como una señal de 1 Hz.

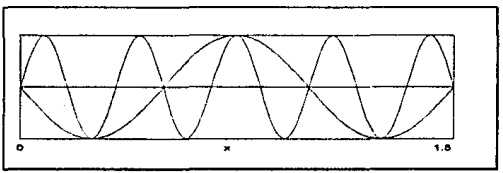


FIGURA 2.4

Frecuencia de 3 Hz disfrazada como una Señal de 1 Hz

La Transformada de Fourier de la señal muestreada es la repetición periódica de la Transformada de Fourier de la Señal original. Si la señal original tiene sus componentes

de frecuencia a solo 1.5 veces de la velocidad de muestreo, estas componentes se confundirian con las frecuencias de las frecuencias vecinas. Esto es muestrado en la Figura 2.4, donde la señal de 3 Hz cae dentro de la señal de 1 Hz.

La Figura 2.5 ilustra el caso donde una señal es muestrada por un tren de pulso ideal. Hay que notar que la transformada de la salida del filtro paso bajos (FPB) no es la mismo que la transformada de la señal original. Si el filtro de salida se expresa como $s_0(t)$, el error puede ser definido como:

$$e(t) = s_0(t) - s(t) \qquad \text{Eq. 2.2}$$

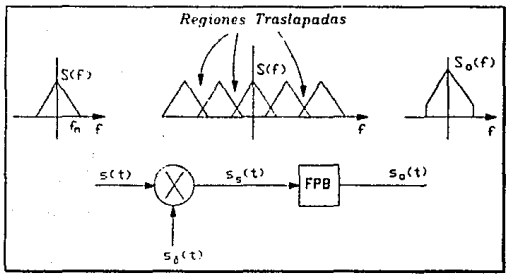


FIGURA 2.5

Si obtenemos la Transformada de Fourier de ambos lados de la ecuación 2.2 tenemos:

$$E(f) = S_0(f) - S(f)$$
$$= s(f - f_s) + S(f + f_s) \text{ para } f < f_m$$

Hay que notar que si $S(f)$ estuviese limitada para frecuencias menores a $f_s/2$, el error de transformación sería cero. No asumiendo una forma especifica para $S(f)$, no

podriamos extrapolar este ejemplo. En general, se prodrían colocar varias regiones dependiendo de la magnitud de la función de error, basado en las propiedades de $S(f)$ para $f>f_s/2$.

2.2 PULSE CODE MODULATION (PCM)

PCM es una técnica para redondear amplitudes de muestreo de una señal. Esta es la segunda de dos operaciones requeridas para cambiar una señal analógica a una señal digital. La operación de redondeo es conocida como *cuantificación* y el error de redondeo es conocida como *ruido de cuantificación*. Esto se puede observar de una manera muy simple, tomando la señal analógica original y aproximandola a una función de escalera, donde los pesos de cada escalón son los valores permitidos por el redondeo (*round-off*). La Figura 2.6 muestra un ejemplo de una señal analógica y su aproximación por escaleras. Mientras mayor sea el número de niveles utilizados, la función de escalera será más aproximada a la señal original. El número de niveles utilizados, determina la resolución de la señal.

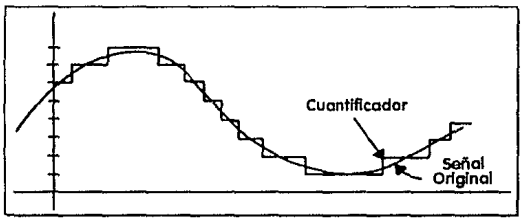


FIGURA 2.6

Digitalización de una Señal Analógica a Digital

Cada muestra evaluada es redondeada a su apropiado nivel de cuantificación, se necesita transmitir solo la información necesaria para que el receptor sepa que nivel que se le transmite. La técnica PCM codifica los niveles en número binarios y transmite el código binario correspondiente dependiendo del nivel de redondeo. Esto es,

por ejemplo, si se tienen ocho niveles de cuantificación, los valores serán codificados en números binarios de 3 bits.

El número binario resultante puede ser transmitido usando una gran variedad de técnicas. En donde en una de ellas, por ejemplo, se pueden transmitir los unos como pulsos positivos y pulsos con valor cero representando los ceros.

La Figura 2.7 representa una señal analógica, la cual es muestreada en nueve intervalos. También en la misma Figura se muestra el tren de pulsos resultantes.

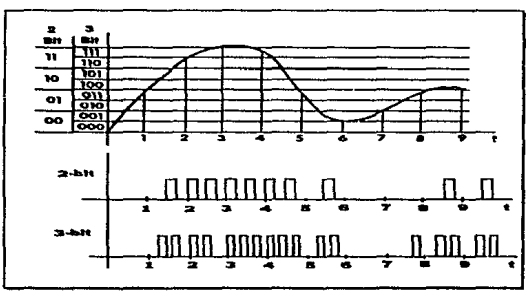


FIGURA 2.7

Cuantificación de una Señal Analógica

2.2.1 MODULADORES PCM

Un modulador PCM no es otra cosa que un convertidor Analógico-Digital. El convertidor primero muestrea la señal y cuantifica cada valor.

La Figura 2.8 ilustra un cuantificador de 3 bits. En este caso se asume que los valores han sido normalizados para estar entre 0 y 1. La Figura 2.8(a) muestra el rango de

valores en ocho regiones. A cada una de estas regiones se les asigna un número binario de 3 bits. Se escogen ocho niveles, debido a que 8 es una potencia de 2. Todas las combinaciones binarias con 3 bits son utilizadas, teniendo con esto mayor eficiencia. La Figura 2.8(b) ilustra el proceso de cuantificación como una relación funcional entre la entrada y la salida.

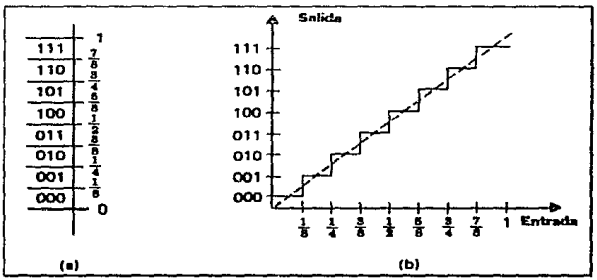


FIGURA 2.8

Muestreo de la Señal

Si se examinan estos número binarios a lo largo de las ordenadas, se nota que el primer bit es igual a 1 para la parte alta y 0 para la mitad inferior. La cual oscila con un periodo igual al rango total. El siguiente bit se alterna con un periodo igual a la mitad del rango y es igual a 1 para la mitad superior de cada mitad y 0 para la mitad inferior de su correspondiente segmento. Este patrón continua con cada bit sucesivo, dividiendo la región en 2 e indicando que mitad de la nueva subregión se encuentra dentro del valor.

Existen 3 formas para los cuantificadores:

1. Cuantificadores Contadores (Counting Quantizers), cuentan serialmente a través de cada nivel de cuantificación.
2. Cuantificadores Seriales (Serial Quantizers), generan una palabra código, bit por bit. Esto es, empiezan con el bit más significativo y terminan con el menos significativo.
3. Cuantificadores Paralelos (Parallel Quantizers), generan todos los bits de una palabra código simultáneamente.

2.2.2 CUANTIFICADORES CONTADORES (COUNTING QUANTIZERS)

La Figura 2.9 ilustra un cuantificador contador. El generador de rampa empieza en cada punto de muestreo y el contador binario comienza simultáneamente. La salida del sample-and-hold es una aproximación de escalera, con escalones que empiezan en el valor de muestreo a través del intervalo de muestreo. Una señal típica es mostrada en la Figura 2.10, la duración en tiempo de la rampa y la duración del contador, T_s, es proporcional al valor de muestreo. Esto es porque la inclinación de la rampa se mantiene constante. Si la frecuencia del reloj es tal que el contador tiene suficiente tiempo para contar hasta su número más alto (todos 1's), tendrá una duración de rampa correspondiente a la máxima muestra posible, así el final de la cuenta correspondera a los niveles de cuantificación.

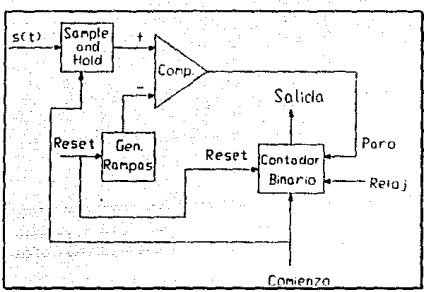


FIGURA 2.9

Cuantificador

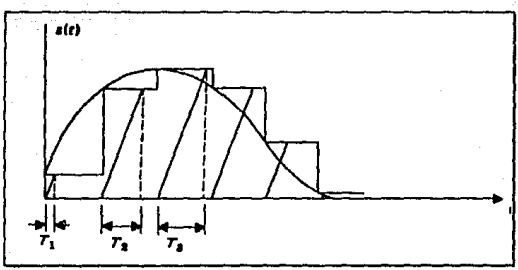


FIGURA 2.10

Contador

2.2.3 CUANTIFICADORES SERIALES (*SERIAL QUANTIZERS*)

Los cuantificadores seriales dividen sucesivamente la ordenada en dos regiones. Primeramente dividen el eje en 2 mitades y observan si el muestreo se encuentra en la mitad superior o inferior. El resultado de esta observación genera el bit más significativo de la palabra código.

La mitad de la región en donde se encuentra el muestreo es subdividido en otras dos regiones, y se realiza nuevamente una comparación, generando con esto el siguiente bit; el proceso continua de esta manera un número de veces igual al número de bits a codificar.

En la Figura 2.11 se muestra un diagrama en bloques de este codificador de 3 bits para entradas entre el rango de 0 y 1. Los figuras con forma de diamante son los comparadores. Estos comparan la entrada para ciertos valores arreglados, dando una salida; si la entrada excede los valores permitidos dará otra salida arreglada. El diagrama de bloques indica estas dos posibilidades como dos posibles caminos de salida denominados *SI* y *NO*. Esta misma figura muestra una palabra código de 3 bits y un rango entre los valores de 0 y 1V. Si el rango de entrada de los valores de muestra de la señal no son 0 y 1, la señal puede ser normalizada (cambiada y después amplificada o atenuada) para que tenga valores entre este rango. Si más o menos bits son requeridos, el bloque de comparación apropiado, puede ser añadido o removido según sea el caso.

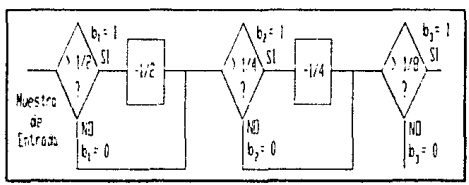


FIGURA 2.11

Cuantificador Serial

Hay que notar que si b_1 es el primer bit del valor de código muestreado, es conocido entonces, como el bit más significativo denominado msb (most significant bit). b_3 es el tercer y último bit del código y es conocido como el bit

menos significativo lsb (least significant bit). La razón de esta terminología es que el peso asociado a b_1 es 2^2, o 4, mientras que el peso asociado a b_3 es 2^0 o 1.

2.2.4 CUANTIFICADORES PARALELOS (*PARALLEL QUANTIZERS*)

Los cuantificadores paralelos (ó codificadores tipo flash) son los más rápidos en operación, ya que desarrollan todos los bits de la palabra clave, simultáneamente. También son los más complejos, ya que requieren un número de comparadores que es solo uno menos que el número de niveles de cuantificación. Esto se ilustra a continuación (Figura 2.12) utilizando solo 3 bits de codificación, como un ejemplo.

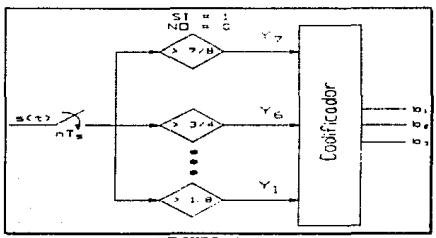


FIGURA 2.12
Cuantificadores Paralelos

La Figura 2.12 muestra un diagrama de bloques de un codificador de 3 bits. El bloque etiquetado "codificador" recibe la salida de siete comparadores. Esto es un simple circuito lógico combinatorio. Si todas las siete salidas son 1 (*SI*), la salida del codificador es 111, debido a que el valor de muestra tiene que ser mayor a 7/8. Si los comparadores de salida del primero al sexto son 1 y la salida del séptimo es 0, el código de salida es 110, debido a que la muestra se encuentra entre los rangos de 6/8 y 7/8.

Así se continua con todos los niveles y finalmente, si todas las salidas de los comparadores se encuentran en estado bajo, la muestra tiene que ser menor a $\frac{1}{8}$, así es que la salida del codificador será 000. Debido a que solo 8 de las 128 (2^7) posibles salidas del codificador son legales, y las otras 120 representan posibilidades ilegales, el circuito lógico combinatorio contiene un alto porcentaje de condiciones sin importancia, y el diseño se simplifica.

Mientras que el cuantificador serial toma ventaja de la estructura de los números binarios, contando en secuencia, el cuantificador paralelo no requiere de esta estructura. De hecho, el código para las regiones de cuantificación pueden ser asignadas de cualquier manera. Un problema con la asignación secuencial es que la transmisión de errores causan una reconstrucción no uniforme de los errores. Un error en el bit msb causa un error mayor que el lsb.

En 1947, F. Gray, quien se encontraba trabajando con códigos electrónicos, inventó un "código binario de reflexión", en donde todos los números adyacentes difieren en solo un bit de posición (Tabla 2.1).

Dígito	Binario	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

TABLA 2.1

Un cambio de uno de estos dígitos causa solo un cambio en un solo bit. Este código es fácilmente implementado en la lógica de los codificadores flash. También puede ser utilizado en otros tipos de cuantificadores. En el cuantificador por conteo, simplemente se varia la secuencia de conteo. En el cuantificador serial, las operaciones de decisión son seguidas de un simple circuito combinatorio para convertir la secuencia a un código Gray.

2.2.5 DECODIFICADORES PCM

Cambiemos nuestra atención por el momento para revisar la conversión de una señal digital a analógica. Esto es realizado por un convertidor digital-analógico (convertidor D/A o DAC). Para realizar esta conversión, es necesario asociar un valor a cada palabra código binaria. Si la palabra código representa una región de valores, el valor se escogen normalmente como el centro de la región. Si la conversión A/D se realizó como se mencionó anteriormente, la operación inversa será la de asignar un valor a cada bit de posición.

Como por ejemplo, si tenemos una palabra binaria de 4-bits, y suponemos que la señal analógica original se encuentra entre el rango de 0 y 1 V. y se utilizó un código secuencial. Se puede observar que la conversión a valores analógicos, se trata de una conversión de números binarios a decimales. Esto es, por ejemplo, el código 1101 representa el número 13, así es que se convierte como 13/16 + 1/32 = 27/32. La adición de 1/32 se toma del final de 1/16 a la mitad.

La Figura 2.13 ilustra un circuito conceptual de como se realizaría la conversión. Si un 1 apareciera como el bit más significativo, msb, una batería de 1/2 V se conectaría al circuito. El segundo bit conectaría una batería de 1/4 V., y así sucesivamente.

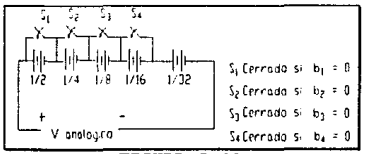


FIGURA 2.13
Decodificadores

El decodificar ideal de la figura anterior es análogo al cuantificador serial, por el hecho de que cada bit es asociado con un componente en particular del mismo valor.

Un decodificar más complejo resulta cuando se intenta una operación de conteo. La Figura 2.14 muestra un decodificador de conteo. Un reloj alimenta simultáneamente a un generador de escalera y a un contador binario. La salida del contador binario es comparado con a entrada binaria digitalizada. Cuando ocurre una igualdad, el generador de escalera se detiene. La salida del generador es tomada hasta que la siguiente muestra sea llevada a cabo. Al final de la aproximación por escalera se coloca un filtro paso bajas para recobrar la señal original.

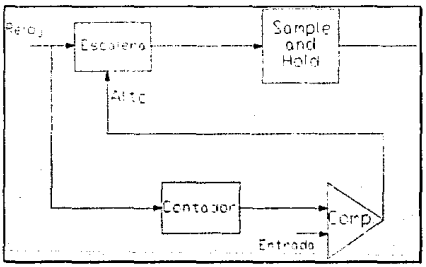


FIGURA 2.14

Diagrama a Bloques de un Decodificador

2.2.6 CUANTIFICACION NO UNIFORME

La Figura 2.15 ilustra el proceso de cuantificación como una función de la entrada contra la salida.

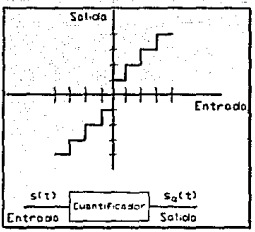


FIGURA 2.15

Cuantificación Uniforme

En la Figura 2.15, el rango de muestreo se ha dividido en regiones de cuantificación, cada una del mismo tamaño. Esto es, por ejemplo, si se tienen 3-bits de cuantificación, se divide el rango completo en ocho regiones iguales (2^3). Debido a que todas las regiones son del mismo tamaño, se denomina entonces, *cuantificación uniforme*.

Bajo ciertas circunstancias, resulta ventajoso utilizar intervalos de diferentes dimensiones, esto es, se puede remplazar la gráfica anterior, con la siguiente (Figura 2.16):

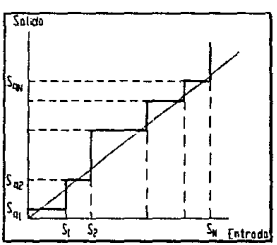


FIGURA 2.16

Cuantificación No Uniforme

La gráfica tiene la propiedad que el espacio que existen entre los niveles de cuantificación no es uniforme, y los niveles de salida no se encuentran en el centro de cada intervalo.

Si por ejemplo, se considera una pieza musical, donde la señal varia entre -2 y +2 Volts, suponiendo 3-bits de cuantificación uniforme, por consiguiente todos los voltajes entre 0 y 1/2 Volt serán codificados en la misma palabra código, 100, que corresponde al valor de salida reconstruido de 1/4. Del mismo modo, todas las muestras que se encuentren entre 1.5 y 2 Volts se les asignara la palabra código, 111, que corresponde al valor de salida reconstruido de 7/4. Durante algunas piezas, donde por periodos muy largos la señal no exceda 1/2 Volt, existirá gran perdida de la definición musical. La cuantificación provee de la misma resolución para niveles superiores e inferiores, más aún, el oído humano es menos sensitivo a variaciones en niveles más altos. La respuesta del oído humano es no lineal. Sería deseable que se utilizaran pequeños escalones en niveles más bajos y pasos mucho mayores en niveles más altos.

Como una justificación alternativa para utilizar cuantificación no uniforme, supóngase que la señal tiene un mayor porcentaje de tiempo en niveles bajos que en los altos. Sería preferible el proveer de mayor resolución a los niveles más bajos que a los altos. Puesto que la señal pasa menos tiempo en los niveles superiores, el promedio de cuantificación de error será mucho menor si se utiliza esta aproximación.

La cuantificación de error o ruido de cuantificación, es una medida de la efectividad del esquema de cuantificación.

El promedio de cuantificación de error es una función de las regiones de cuantificación, los valores redondeados y la densidad de probabilidad de los valores muestreados. La Figura 2.17 muestra un ejemplo de una función de densidad de probabilidad que se asemeja a una probabilidad de densidad de Gauss. Si se divide en ocho regiones iguales, indicadas desde s_0 hasta s_8, los niveles de redondeo serán aproximados a cada una de las regiones.

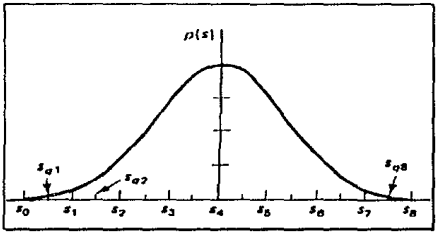


FIGURA 2.17

Densidad de Probabilidad de la Cuantificación del Error

Si, en adición, para especificar los niveles de cuantificación, una complicada optimización resulta y un sistemas más complejo se requerirá para implementar los resultados. En realidad, los cuantificadores resultaran muy complejos, y cada cuantificador de nivel requerirá de comparadores separados. Por esta razón, y también para que este tipo de sistemas sea aplicable a cualquier tipo de señal de entrada, sistemas por debajo de lo óptimo son utilizados.

2.2.7 COMPANDING

La forma más común de cuantificadores no uniformes es llamada *companding*. El nombre es derivado de las palabras del inglés, "compressing-expandig". Este proceso se ilustra en la Figura 2.18:

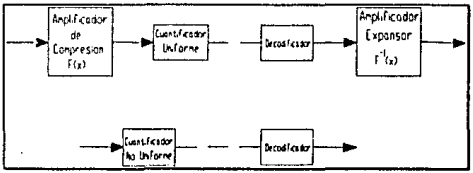


FIGURA 2.18

Diagrama a Bloques del Proceso de Compresión-Expansión

En donde la señal original es comprimida utilizando un dispositivo no lineal. La señal comprimida es entonces cuantificada uniformemente. Siguiendo las demás etapas de transmisión, la señal viaja hasta que debe de ser expandida utilizando una función no lineal, la cual es inversa a la utilizada en la compresión.

Antes de cuantificar, la señal es distorcionada por una función muy similar mostrada en la Figura 2.19. Esta operación comprime los valores extremos de la señal, mientras realza los valores pequeños, las funciones logarítmicas son utilizadas para poder observar valores muy pequeños y muy grandes en el mismo eje. Si la señal analógica es la entrada de este compresor, y la salida es cuantificada uniformemente, el resultado es el equivalente a cuantificar con escalones que empiezan muy pequeños y se van incrementando de tamaño para señales más altas. Esto es mostrado en misma Figura 2.19. Se ha dividido la salida de este compresor en ocho regiones de cuantificación. Esta

función es utilizada para trasladar los límites de esta región a la abscisa, la cual representa la señal de entrada no comprimida. Hay que notar que la región del eje s, empieza con pasos muy pequeños y se va incrementando a medida que se incrementan los valores de s.

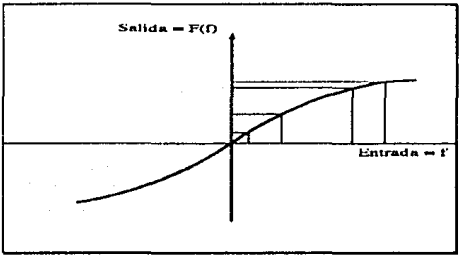


FIGURA 2.19

Función Distorcionada

En función de realizar algunas mejoras en hardware, es bueno acordar algunos estandars para compresión. Esto es, si se tiene alguna esperanza de fabricar un codificador PCM con compresor, es necesario acordar algunas formulas estandars para compresión.

La aplicación más común de la compresión es en la transmisión de voz. Los Estados Unidos y Japón han adoptado una curva estandar de compresión conocida como "μ-law". Europa ha adoptado una ley diferente, pero muy similar, conocida como "A-law".

La formula de compresión de ley μ es la siguiente:

$$F(s) = sgn(s)\frac{\ln(1+\mu|s|)}{\ln(1+\mu)}$$

Esta función esta graficada para algunos valores de μ en la Figura 2.20. El parámetro μ, define el ángulo de curvatura de la función. Un valor típico es de μ = 255.

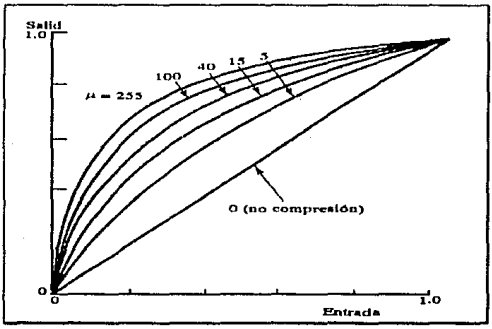


FIGURA 2.20

Gráfica de Ley μ

Se deben de examinar todas las posibles implementaciones de la codificación para μ-255 de 8-bits. Una aproximación es el de simular un sistema no lineal que siga la curva μ-255 de entrada/salida, después, colocar los valores muestreados del sistema y cuantificar uniformemente la salida utilizando un convertidor A/D de 8-bits. Otra aproximación para la curva μ-255 es la de hacerla por secciones lineales como se observa en la Figura 2.21.

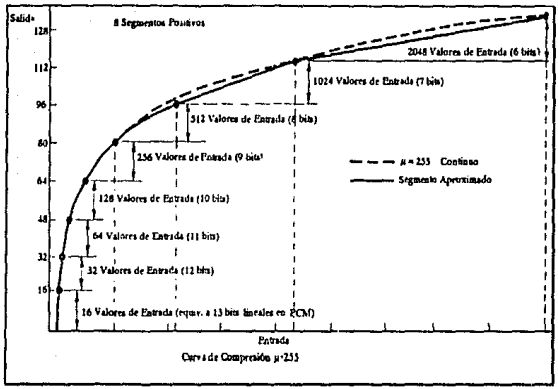


FIGURA 2.21

Curva de Compresión μ

Hay que notar que se tiene la curva aproximada con ocho segmentos lineales. Si se divide la región positiva de salida en ocho segmentos iguales, se dividirá la región de entrada en ocho segmentos desiguales.

Con cada uno de estos segmentos, se cuantifican los valores de las muestras utilizando 4 bits. Así, cada una de estas ocho regiones será dividida en 16 subregiones, para un total de 128 regiones en cada lado del eje. Así es que se tienen en total 256 o 2^8 regiones, los cuales corresponden a 8 bits de cuantificación. La técnica específica en mandar una muestra, es mandarla en un código de 8 bits de la siguiente manera (Tabla 2.2):

# de bits	Descripción
1	Da la polaridad de la muestra: 1 para positivos, 0 para negativos.
3	Identifica en que región cae la muestra.
4	Identifica el nivel de cuantificación para cada muestra en esa región.

TABLA 2.2

La relación logarítmica entre la ley μ-255 estriba en el hecho de los ochos segmentos. Cada segmento en el eje de entrada es del doble de ancho que el segmento a su izquierda. La resolución del primer segmento a la derecha del origen cubrirá todos los intervalos en en eje de entrada, el cual es 1/16 del largo total. Esto es, la resolución de las muestras en un intervalo en particular es el mismo que el utilizando por una cuantificación uniforme utilizando un A/D de 8 bits. La resolución en la región justo a la izquierda de esta será la misma que una cuantificación uniforme de 9-bits. De la misma manera, cada vez que se mueve uno a la izquierda, cada región tiene la resolución de un cuantificador uniforme con más bits de cuantificación que el que le precedió. Esto se puede observar en la Figura 2.22.

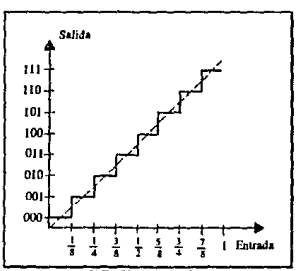


FIGURA 2.22

Relación Logaritmica de la Ley μ

3. CAPITULO II TELEVISION DIGITAL"

3. 1 TV DIGITAL ASPECTOS GENERALES

En los últimos años, se han observado grandes cambios en las imágenes que llegan al receptor de TV. Estos cambios (especialmente en los video clips, spots publicitarios, cabeceras de programas y programas elaborados con técnicas de postproducción complejos) han sido posibles al poder digitalizar la señal de vídeo y por lo tanto, usar dicha tecnología en los respectivos equipos de producción. Así, la gente se ha acostumbrado a ver fotogramas que se pliegan, letras y objetos desplazándose a lo largo y ancho de la pantalla, superposición de varias figuras generadas con personajes surgidos de la realidad, yuxtaposiciones de diseños gráficos con animaciones en tiempo real, "collages" de formas varias y objetos, imágenes sintéticas creadas con computadoras, etc., cuyo límite sólo está condicionado por la imaginación o capacidad artística y creativa de la persona humana, que tiene la posibilidad así de hacer realidad sus fantasías visuales y por tanto transformar sus sueños en realidad.

La imagen digital y de síntesis ha creado una dimensión suplementaria donde no tienen validez las leyes de Newton. El espacio y el tiempo se rigen por leyes completamente nuevas que escapan a nuestra percepción habitual y nos transportan a otros mundos en los que se mezclan la fantasía con la realidad.

También, como usuarios de los vídeos domésticos, se han comprobado los efectos mosaico, zoom, imagen sobre imagen (*"Picture in Picture"*, PIP), etc. también posibles al incorporar tecnología digital en el proceso de la señal de vídeo.

A nivel profesional, y por citar tan sólo algunos aspectos, se ha mejorado enormemente la estabilidad y sincronización de la señal de TV (correctores de base de tiempo, "*Time Base Corrector*", (TBC) y sincronizadores), la transcodificación de programas de distintas normas, la posibilidad de generar efectos e imágenes y la edición y postproducción. Todo esto gracias a incorporar las técnicas digitales, propias del mundo informático, a la señal de vídeo.

3.1 HISTORIA DE LA TELEVISION DIGITAL

Muchos de los sistemas que se desarrollaron en un principio eran imperfectos, antieconómicos, pero lo cierto es que sin ellos no se habría llegado al desarrollo de las sistemas actuales. Ello permitió, asimismo, que se avanzara en otros campos relacionados con la TV, como ocurrió con el desarrollo que tuvo lugar en la década de los 40´s, respecto al conocimiento del comportamiento del ojo humano en la visión de los colores, lo cual sirvió de punto de partida para los sistemas de TV en color vigentes hoy día (NTSC, PAL, SECAM).

Desde el año 1884, en que el joven Paul Nipkow inventó el sistema de exploración mecánica de la imagen con célula fotoeléctrica (y que por falta de medios técnicos no pudo ser utilizado en la práctica hasta 1923), hasta nuestros días, cercanos ya a la última década del siglo XX, en que son habituales las expresiones o siglas HDTV (sistema de TV en alta definición), satélites DBS (satélites de radiodifusión directa), sistemas MAC (sistema de transmisión de la señal de TV multiplexada en componentes y en forma analógica), el hombre no ha cesado de investigar para cada vez exigir más perfección y una mayor calidad tanto en el campo de la TV como en cualquier otro de la ingeniería, biología, química, medicina, etc.

Paul Nipkow fue el primero en comprender que la transmisión de una imagen debe hacerse analizándola en elementos discretos, transmitirla, obtenerla de nuevo en el receptor en elementos discretos y dejar que sea el ojo humano el que finalmente la sintetice, obteniendo así la sensación de imagen visual completa debido a las propias características de funcionamiento de nuestro órgano visual.

En la Tabla 3.1, y sin ánimo de pretender dar una lista exhaustiva, exponemos un resumen de las hechos más interesantes ocurridos desde el año 1862 y que condujeron al estado actual de los sistemas de TV. La mayoría de todos estos avances técnicos han estado condicionados lógicamente por los progresos obtenidos en el campo de obtención de imágenes (tubos de cámara).

TABLA 3.1

FECHA	AUTOR	PAIS	HECHO
1862	Caselli	Italia	Primeros Experimentos de Reproducción de Imágenes a distancia.
1868	Hittorf	Alemania	Descubrimiento de los rayos Catódicos.
1873	Maxwell	Inglaterra	Se establecen las ecuaciones generales del campo electromagnetico.
1881	Selecq	Francia	Inventa el Teletroscopio.
1884	Nipkow	Rusia-Alemania	Inventa el disco de Nipkow.
1897	Braun	Alemania	Inventa el Osciloscopio Catódico.
1903	Korn	Alemania	Transmisión de Imágenes por un sistema telegráfico.
1923	Baird	Inglaterra	Realizó prácticamente el disco de Nipkow.
1923	Jenkins	USA	Transmisión de Imágenes a 200 Km.
1923	Zworykin	Rusia-USA	Inventa el Iconoscopio.
1924	Baird	Inglaterra	Inventa el Receptor de TV doméstico.
1925	Baird	Inglaterra	Primeras Transmisiones de TV de una habitación a otra.
1926	Belin	Francia	Recepción de TV con sistema de espejos acoplado a un Tubo de Osciloscopio.
1926	Takayanagi	Japón	Transmisión a 40 líneas y 14 tramas/seg.
1926	Baird	Inglaterra	Primera demostración pública de un transmisor de TV.
1927	Ives	USA	Transmisión de Imágenes en moviemiento, por cable.
1927	Baird	Inglaterra	Transmisión de Imágenes en Movimiento.
1928	Baird	Inglaterra	Guarda Información de Imágenes en un disco.
1928	Baird	Inglaterra	Demostración de la TV a color.
1928		USA	Emisiones Experimentales de TV.
1928	Mihaly	Alemania	Transmisión de TV por cable.
1928	Korlous	Alemania	Recepción en pantalla de 70*75 cm.
1929	Zworykin	Rusia-USA	Inventa Cinescopio.
1929		Alemania	Emisiones Experimentales de TV.
1929		España	Pruebas Experimentales de TV.
1929		Inglaterra	Transmisión simultanea de Imágenes y Sonido.
1930	Zworykin	Rusia-USA	Demostración en Laboratorio del Iconoscopio.
1930	Baird	Inglaterra	Recepción en pantalla de 60*150 cm.
1932		Inglaterra	Emisiones Experimentales de TV.
1934	Shoenberg	Alemania	Obtiene Imágenes con un tubo de Emitrón.
1934		Alemania	Inaguración del Primer Estudio de TV.

1935	Mandel	Francia	Emisiones de TV desde la Torre Eiffel.
1935		Francia	Primer Estudio de TV.
1935		Alemania	Inaguración del Servicio de TV.
1935	EMI	Inglaterra	Descubrimiento del Sistema TV de Exploración Electrónica
1936	FCC	USA	TV de 441 líneas.
1936	Takayanagi	Japón	TV de 441 líneas Electrónicas.
1936			TV al alcance práctico.
1936	BBC	Inglaterra	Primeras Transmisión de TV.
1936	BBC	Inglaterra	Primer Servicio Mundial de TV.
1937		Inglaterra	Abandono del Sistema de Baird.
1937	BBC	Inglaterra	Transmisión con Sistema EMI de 405 líneas.
1937	BBC	Inglaterra	Primera Transmisión de Exteriores.
1938		URSS	Inaguración del Servicio Regional de TV.
1939	Iams y Roase	USA	Inventan el Tubo de Camara Orticón.
1939		USA	Inagura el Servicio Regional de TV.
1940	CBS	USA	Propuesta Primaria para un Sistema de Normas de TV a Color.
1945	RCA	USA	Descubre el Sistema de Transmisión de TV de 3 vías.
1945	Clark	Inglaterra	Realiza las Bases Teóricas para colocar Satélites Geoestacionarios.
1946	CBS	USA	Pruebas satisfactorias de la TV a color.
1949		USA	Inaguración de la Primera Emisira de TV en cadena
1949	RCA	USA	Descubre el Sistema de Transmisión de TV por puntos.
1950	Gabor	Hugría-Inglaterra	Inventa la Pantalla plana para la TV.
1950	RCA	USA	Inventa el Tubo de Camara Vidicón de Trisulfuro de Antimonio.
1953		USA	Entrada del Sistema NTSC.
1957		URSS	Pone en orbita al Sputnik 1.
1958	Bell	Inglaterra	Inventa el Tubo de Camara Vidicón de Diodo de Silicio.
1959	France	Francia	Inventa el Sistema SECAM.
1962	Philips	Holanda	Inventa el Tubo de Camara Plumbicon.
1962		USA	Primeras Transmisiones de TV desde USA vía Telstar a Europa.
1963	Brunc	Alemania	Inventa el Sistema PAL.
1963		USA	Lanza el Primer Satélite Geoestacionario para Comunicaciones, Snyncom 1.
1964		Inglaterra	Cambia la definición de 405 a 625 líneas.
1964		USA	Lanza Snyncom C.
1967	Wescon	USA	Inventa el Sistema Captador de Imágenes por CCD.
1968	NHK	Japón	Primeras Investigaciones de la HDTV.
1970			Se crea la Red Intelsat.
1970	RCA	USA	Crea el CCD de 180*180 pixels.
1972	Toshiba	Japón	Inventa el Tubo de Camara Calnicon.
1972		Inglaterra	Inventa el Teletexto.
1973	BBC	Inglaterra	Primeras Emisiones de Teletexto.
1973	Hitachi	Japón	Inventa el Tubo de Camara Saticon.
1974	TDF	Francia	Se Inician las Emisiones de Teletexto.
1974	Matsushita	Japón	Inventa el Tubo de Camara Newvicon.

1977		Europa	Se Crea el EUTELSAT.
1978	NHK	Japón	Se Inician la Pruebas del HDTV.
1981		Europa	Se Crea la Norma MAC.
1982	CCITT	Internacional	Se Crea la Recomendación 601 para la TV Digital.
1983			Primera Camara de color de CCD.
1985	Matsushita	Japón	Primer Modelo en Operación de una Pantalla Plana.
1985	UER	Europa	Se Crea la Norma C-MAC.
1985	Thomson TDF FR3	Francia	Inaguración del primer Estudio de Producción de TV Digital.
1986	NHK	Japón	Orbita del Primer Satélite de Radiodifusión directa BS-2.
1987		Alemania	Orbita del Primer Satélite Europeo de Radiodifusión.
1988	TDF	Francia	Orbita del Satélite Europeo de Radiodifusión.
1988	Sociedad Europea de Satélites.	Europa	Orbita del Primer Satélite Privado de Radiodifusión.

3.3 TELEVISION ANALOGICA

La televisión utiliza varios principios de transmisión utilizados en el radio (AM y FM), aunque el sistema de televisión no solo se puede transmitir en forma remota sino también por cable. El gran paso en el desarrollo de la televisión fue el establecer un protocolo para convertir o codificar la información en forma de una señal eléctrica.

Los canales estandars para transmitir televisión fueron fijados en el año de 1945 por la Comisión Federal de Comunicaciones (FCC) en los Estados Unidos, donde se les asignaron los números del 2 al 13 ahora usados en América para la televisión comercial y que se encuentran en la denominada banda VHF (*Very High Frecuency*), existe otra banda de canales menos utilizados en México para la transmisión de televisión (14 a 83), a esta banda se le conoce como la banda UHF (*Ultra High Frecuency*).

El sistema de transmisión conocido como NTSC (National Television Systems Commitee) fue adoptado en 1953 y actualmente se utiliza para todos los televisores. Este comité también desarrolló los estandares utilizados en la televisión de blanco y negro.

En el caso de una estación de televisión de tipo comercial, el área de servicio es aproximadamente de 25 a 75 millas en todas direcciones desde el transmisor y con linea de vista directa. La radiación transmitida se encuentra en forma de dos ondas portadoras de radiofrequencia moduladas con la información deseada. La información de la imagen es transmitida en amplitud modulada (AM) y la señal del sonido es transmitida en frecuencia modulada (FM).

3.3.1 IMAGEN DE LA TELEVISION ANALOGICA

El sistema de televisión funciona en forma análoga a una cinta de una película de cine, el sistema es lo suficiente rápido como para crear la ilusión de movimiento proyectando varios cuadros fijos en períodos de tiempo muy pequeños. Una imagen esta conformada por un grupo de pequeñas áreas de luz y sombra. Esto se puede apreciar mejor si se observa con detenimiento la pantalla de un televisor o si se utiliza una lente para observar la imagen y así apreciar los puntos individuales que la forman. Estos puntos se les conoce como pixels (*PIcture ELementS*, elementos de imagen), y contienen la información visual de la escena que se observa, si estos son transmitidos y reproducidos en el mismo grado de intensidad de luz o sombra en su posición adecuada, la imagen será reproducida.

3.3.1.1 BARRIDO HORIZONTAL Y VERTICAL

La pantalla del televisor es recorrida por un haz de electrones que se emite desde el caño del cinescopio del mismo y barre la pantalla. En su camino el haz de electrones va mandando la información visual dependiendo de la posición en que se encuentre.

La formación de imágenes en la pantalla de una televisión es diferente a la manera en que se forma la imagen en una fotografía, en esta última se forma toda de una sola vez al entrar la luz y grabar la escena en la película, en cambio en el televisor la imagen es reconstruida una linea tras otra y cuadro sobre cuadro. La forma en que se realiza este barrido de la pantalla es similar a la forma en que se leen las palabras en un texto escrito, comenzando de la esquina superior izquierda y terminando en la esquina inferior derecha. Este mismo sistema de barrido es utilizado

en la cámara del transmisor para efectuar el muestreo de la imagen tomada.

La secuencia que se emplea para el muestreo de la imagen es el siguiente:

a) El haz de electrones hace un barrido a lo largo de una línea horizontal de la pantalla, cubriendo en su camino todos los puntos de esta línea.

b) Al final de cada línea el haz de electrones regresa rápidamente al extremo izquierdo de la pantalla y vuelve a comenzar a recorrer la siguiente línea horizontal. A este retorno del haz de electrones al otro extremo de la pantalla se le llama retraso. Ninguna información de la imagen es enviada durante este periodo de tiempo, por esta razón los retraso deben de ser muy rápidos por que de otra manera se perdería información de la imagen.

c) Cuando el haz es regresado al extremo izquierdo de la pantalla es movido ligeramente hacia abajo, de tal manera que al comenzar a mandar la información no se encime en la línea anterior, esto se efectúa con el sistema de barrido vertical.

El número de líneas barridas en una imagen completa debe ser grande, de tal forma que se puedan incluir la mayor cantidad de pixels en la pantalla del televisor. El estándar que se utiliza en el sistema de televisión actual es de 525 líneas por cada cuadro de una imagen. Esta cantidad de líneas es la óptima para que el ancho de banda del canal transmitido sea de 6 MHz.

El haz de electrones que barre la pantalla va bajando lentamente mientras realiza el barrido de la imagen; este movimiento vertical es necesario para que no se encimen una línea de la imagen sobre otra. Por esta razón la frecuencia de barrido vertical es mucho menor que la del barrido horizontal. En este caso la velocidad de barrido vertical es de 30 Hz, es decir la mitad de la frecuencia de la línea de alimentación (60 Hz).

3.3.1.2 MOVIMIENTO DE LA IMAGEN

Así como es necesario transmitir todos los puntos de la imagen en la pantalla del televisor por medio del barrido, es también necesario presentar la imagen al ojo humano en tal forma que cualquier movimiento en la escena aparezca como un cambio suave y continuo. El sistema de la televisión es muy parecido en este aspecto al de una película de cine.

En el sistema de cine se muestran aproximadamente 24 cuadros por segundo. La impresión de una imagen vista por el ojo humano persiste por una fracción de tiempo después de que esta ha sido removida. Por esta razón, si muchas imágenes se presentan durante este intervalo de tiempo el ojo las integrará y entonces tendrá la impresión de ver las imágenes al mismo tiempo. Como los elementos son desplegados en la pantalla en rápida sucesión aparecen al ojo como una imagen completa.

Para crear la ilusión de movimiento un número suficiente de imágenes deben de ser mostradas durante cada segundo. Este efecto se puede lograr teniendo más de 16 imágenes por segundo.

El proceso que se utiliza en la televisión para reproducir el movimiento en una escena, no solamente descompone la imagen en muchos puntos individuales, sino que la escena es barrida en forma tan rápida que da la ilusión de movimiento. En vez de una velocidad de 24 cuadros por segundo como en el cine, la velocidad utilizada en T.V. es de 30 veces por segundo. Esta velocidad da una continuidad en el movimiento bastante apropiada.

Sin embargo la velocidad de repetición de 30 imágenes por segundo no es lo suficientemente rápida para evitar el problema del parpadeo a los niveles de luz encontrados, por lo que cada cuadro de la pantalla es dividido en dos partes, de tal modo que 60 imágenes son presentadas al ojo durante cada segundo. Para poder obtener este efecto se entrelazan las líneas horizontales en dos grupos, uno formado por las líneas pares y otro por las líneas impares, cada uno de estos grupos de líneas pares o impares se denominan campos.

Esta velocidad de repetición es lo suficientemente rápida como para eliminar el parpadeo y se adecúa con la frecuencia de la línea de 60 Hz. En paises de Europa donde la frecuencia de la línea es de 50 Hz, se usan 25 cuadros por campo.

15750 líneas son barridas en 1 segundo, por lo tanto la frecuencia de 15750 Hz es la velocidad a la cual el haz completa su ciclo de movimiento de derecha a izquierda y de nuevo hasta la derecha.

Esto indica que las frecuencias que puede contener la imagen pueden estar en el orden de MHz, si en la imagen existieran mayor número de líneas el barrido debería de hacerse más rápido de modo que se mantuviera la cantidad de cuadros por segundo necesarios. En el sistema utilizado actualmente solamente se emplean 525 líneas, lo que da una

frecuencia máxima dentro de la imagen de aproximadamente 4 MHz, si se incluyeran más líneas el ancho de banda del canal de televisión se incrementaría a más de 6 MHz, sin embargo hoy en día se ha comenzado a introducir la televisión de alta definición, la cual posee una mayor cantidad de líneas y mucha mejor resolución, aunque también emplea un mayor ancho de banda que los canales normales.

Cuando el haz de electrones barre la pantalla, este barrido debe de estar correctamente sincronizado con el de la cámara a fin de que corresponda la posición tomada con la mostrada en el televisor, para mantener al transmisor y al receptor sincronizados deben ser enviados una serie de pulsos de sincronización. Los pulsos de sincronización deben de ser transmitidos como parte de la señal enviada al receptor, sin embargo esta se manda en el momento que se hace el pulso de retraso, durante el cuál no se envía información alguna de la imagen. Se envía un pulso de sincronía horizontal al final de cada línea horizontal, y un pulso de sincronía vertical al terminar de barrerse un campo y comenzar el retraso vertical, como resultado de esto, el barrido del transmisor y el receptor se encuentran sincronizados.

Sin la sincronización vertical en los campos, la imagen reproducida en el receptor no se mantendría verticalmente, y se vería la imagen moverse hacía arriba o abajo en forma continua, por otra parte si no existiera buena sincronía horizontal se vería que la imagen se inclina hacia la derecha o izquierda dividiéndose en segmentos diagonales. De esto se tiene que los pulsos de sincronía deben de tener una frecuencia igual que el barrido horizontal, es decir, 15,750 Hz. La cantidad de cuadros mostrados en la pantalla es de 30 por segundo pero la frecuencia de barrido vertical es de 60 Hz y la frecuencia de los pulsos de sincronización es por tanto la misma.

3.3.1.3 SEÑAL DE COLOR

El sistema que se utiliza en la televisión a color es análogo al de la televisión en blanco y negro, la diferencia es que se le agrega la información del color a la imagen en cantidades de rojo, verde y azul. Cuando la imagen es barrida en el tubo de la cámara, se producen señales separadas de verde, azul y rojo de la escena. Los filtros ópticos de color separan los colores dentro de la cámara. Sin embargo para transmitir en un canal estandar de 6 MHz las señales roja, azul y verde se combinan en una sola señal en conjunto con la de brillo. Especificamente las dos señales transmitidas son:

1. Señal de brillo o luminancia: Contiene únicamente variaciones de brillo de la información de la imagen, incluyendo detalles como una señal monocromática. La señal de brillo se utiliza para reproducir la imagen en blanco y negro. Esta señal se le conoce como señal Y.

2. Señal de croma o crominancia: Contiene la información de color. Esta señal es transmitida como modulación en una subportadora a 3.58 MHz para todas las estaciones. Por tanto 3.58 MHz es la frecuencia de color. Esta generalmente se le conoce como la señal C.

En el receptor de color la señal de brillo se combina con la del color para recobrar los tonos originales de rojo, verde y azul de la señal de video. La pantalla del receptor de televisión tiene fósforos que producen fluorescencia roja, verde y azul. Todos los colores pueden ser producidos con mezclas de estos tres colores.

El los receptores monocromáticos, la señal Y reproduce la imagen en blanco y negro. La subportadora de 3.58 MHz no

se utiliza. En este caso, los 3.58 MHz son filtrados fuera de la señal de video, para prevenir interferencia con la imagen monocromática. Esto permite que los sistemas monocromáticos y de color sean totalmente compatibles. Cuando un programa es televisado en color, la imagen es reproducida en color por los receptores de color, mientras que los receptores monocromáticos muestran la imagen en blanco y negro. Lo que es más, los programas televisados en blanco y negro son reproducidos en blanco y negro por ambos receptores. El tubo de imagen de colores puede también reproducir blanco combinando rojo, verde y azul.

3.3.2 EL CANAL DE TRANSMISION

El grupo de frecuencias asignadas por el FCC (en los Estados Unidos, pero las mismas normas se usan en toda América) para transmitir se le denomina canal. Cada estación de televisión tiene 6 MHz de ancho de banda colocado dentro de una de las siguientes bandas de transmisión:

54 a 88 MHz para canales de banda baja de VHF del 2 al 6
174 a 216 MHz para canales de banda alta de VHF del 7 al 13
470 a 890 MHz para canales de UHF del 14 al 83

En todas estas bandas cada canal de televisión es de 6 MHz de ancho de banda. Por ejemplo el canal 3 de televisión va desde 60 MHz a 66 MHz.

3.3.2.1 MODULACION DE VIDEO

El ancho de banda del canal de 6 MHz se necesita principalmente para la portadora de la señal de imagen. La portadora es modulada en amplitud por la señal de video con un amplio rango de frecuencias de aproximadamente 4 MHz, correspondientes a los detalles mas pequeños en la pantalla.

3.3.2.2 MODULACION DE CROMA

Para transmisiones en color, la señal de croma de 3.58 MHz contiene la información de color. Esta señal se combina con la de brillo para formar una sola señal de video que module la portadora de imagen para transmitir al receptor.

3.3.2.3 EL SONIDO DE FM

También incluido dentro de el canal de 6 MHz se encuentra la portadora de la señal de sonido de la imagen, a esta se le conoce como sonido asociado. La portadora de sonido es una señal de FM modulada por las frecuencias de audio en un rango de 50 hasta 15,000 Hz. Este rango de frecuencias de audio es el mismo para todas las estaciones comerciales de FM en la banda de 88 a 108 MHz. En la señal de sonido de la TV la máxima variación de frecuencia de la portadora es de ±25 kHz para una modulación del 100%. Esta variación es menor que el ±75 kHz de la modulación del 100% en la banda comercial de FM. Sin embargo el sonido en la televisión.

Debe notarse que el AM es mejor para la señal de imagen por que los "fantasmas" que aparecen en la pantalla del el televisor resultado de recepción múltiple de varias direcciones es menos obvia. Con AM los "fantasmas" aparecen fijos pero con FM estos se moverían en la pantalla.

3.3.2.4 CANALES DE TELEVISION

A cada estación de televisión se le asigna un canal de 6 MHz para transmisión de señal de imagen de AM y su señal de sonido de FM.

Ya que la frecuencia de la portadora de imagen debe ser mucho mayor que la frecuencia de modulación de video de 4 MHz, los canales de televisión son asignados en la banda de VHF de 30 a 300 MHz y la banda de UHF entre la banda de 300 a 3,000 MHz. Los canales de televisión pueden considerarse dentro de tres grupos: los cinco canales de banda baja de VHF (2-6), siete canales de banda alta (7-13) y 70 canales de UHF

(14-83). Las frecuencias entre estas bandas de transmisión de televisión son usadas por otros servicios de radio.

El número de canales disponibles para transmisión de televisión en una localidad depende de se población, variando desde un canal en una ciudad pequeña hasta 12 estaciones para ciudades grandes, incluyendo canales de VHF y UHF. La mayoría de las ciudades tienen al menos un canal reservado para televisión educativa no comercial.

Un canal puede ser utilizado por muchas estaciones, pero deben estar lo suficientemente distantes para minimizar la interferencia entre ellos. Deben de estar separadas por 170 a 220 millas para canales de VHF o 155 a 205 millas para canales de UHF. Aquellas estaciones que usan canales adyacentes en frecuencia, como los canales 3 y 4, se llaman estaciones adyacentes. Para minimizar la interferencia entre estas, las estaciones de canales adyacentes deben tener entre ellas al menos 60 millas para canales de VHF o 55 millas para estaciones de UHF. Sin embargo, canales consecutivos en número pero no adyacentes en frecuencia, tal como los canales 4 y 5, 6 y 7, o canales 13 y 14 pueden ser asignados en una misma área.

3.4 MUESTREO DE LA SEÑAL DE TV

Para la señal de TV que tiene un ancho de banda de 5.5 MHz (sistemas B y G), se debe utilizar una frecuencia mínima de muestreo de 11 MHz, siempre que la señal se haga pasar por un filtro paso bajos de frecuencia de corte abrupta antes y después del muestreo, para separar las bandas laterales de la señal de banda base. Una serie de consideraciones, que posteriormente se mencionarán, llevó a fijar (Recomendación 601 del CCITT) como frecuencia de muestreo de la señal de TV, 13,5 MHz para la luminancia y 6.75 MHz para las señales componentes de color.

Las razones que llevaron a elegir la frecuencia de muestreo de 13.5 MHz para la luminancia fueron:

- Hay sistemas en los que el ancho de banda es de 6MHz (L/Secam y K/Secam) y según el teorema del muestreo, como mínimo la frecuencia de muestreo debe ser de 2*6 = 12 MHz.

- Es deseable y conveniente que el número de muestras por línea sea idéntico para todas, facilitando así la estandarización de las memorias digitales. Ello implica que la frecuencia de muestreo debe ser un múltiplo entero de la frecuencia de línea (n muestras por línea). En una línea PAL con esta frecuencia de muestreo, se consigue que haya un número entero de muestras (por el offset de un cuarto de línea).

Al tener las mismas muestras en cada línea, no habrá problemas al utilizar eventualmente muestras de líneas antecedentes o precedentes. Este tipo de muestreo se denomina

"ortogonal" y es idéntico en todas las líneas, campos y cuadros, Figura 3.1.

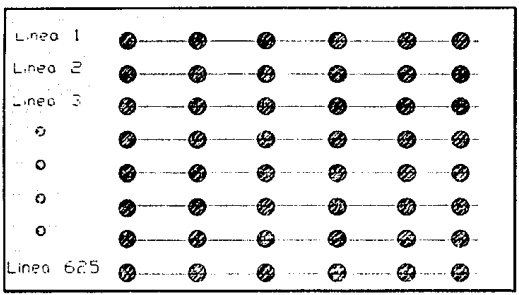


FIGURA 3.1

Estructura de muestreo estática en el espacio y en el tiempo

- Para poder obtener una frecuencia de muestreo universal, debería ser un múltiplo entero de las frecuencias de línea de los sistemas existentes actualmente utilizados en todo el mundo (en el sistema de 625 líneas/cuadro es 15.625 1/s y en el de 525 es de 15734.26573 1/s). El mínimo común múltiplo de ambas frecuencias es aproximadamente 2.25 MHz. Como es un valor inferior a los 12 MHz que se requieren según Nyquist, se elige el valor de 6*2.25 MHz = 13.5 MHz que corresponde aproximadamente a 858 veces la frecuencia de línea del sistema NTSC (525 líneas) y 864 veces la frecuencia de línea del sistema PAL y SECAM (625 líneas).

Según la Recomendación 601 del CCITT:

En cada línea del sistema de NTSC (525 líneas) se toman 858 muestras y 864 en el sistema PAL (625 líneas) para la señal de luminancia. Para las señales diferencia de color se

toman 429 muestras en el sistema NTSC y 432 en el PAL. Estas se van tomando coincidiendo con las muestras impares de la luminancia, es decir que en una línea, se formarían muestras de luminancia y coincidiendo con las muestras impares (1, 3, 5, 7, etc.) se toman las correspondientes a las señales diferencia de color. A continuación se cuantifican y codifican y se transmiten mediante un sistema de multiplexado temporal. Así, suponiendo que se esta en la primera línea, la manera de tomar las muestras es:

```
pixel 1: Y, (R - Y), (B - Y)
pixel 2: Y
pixel 3: Y, (R - Y), (B - Y)
pixel 4: Y
etc.
```

Estas muestras, se transmitirán:

Y, (R - Y), Y, (B - Y), Y, (R - Y), etc.

Lógicamente, aquí no se tiene en cuenta la reducción de la velocidad binaria, según la redundancia que existe en la señal de TV (por la que omitiendo cierto número de muestras podremos reconstruir toda la información) ni es el objetivo exponer el muestreo "sub-Nyquist" al usar filtros digitales bidimensionales que permiten mantener la resolución horizontal y vertical en decremento de la definición diagonal u oblicua en la que parece ser que el ojo es menos sensible, para poder evitar la pérdida de la definición de color que produce al disminuir el número de muestras que se transmiten.

En la Figura 3.2 puede verse el muestreo que se produce en las tres direcciones (dos espaciales x e y más una temporal z) de una señal de TV, según la norma 4:2:2.

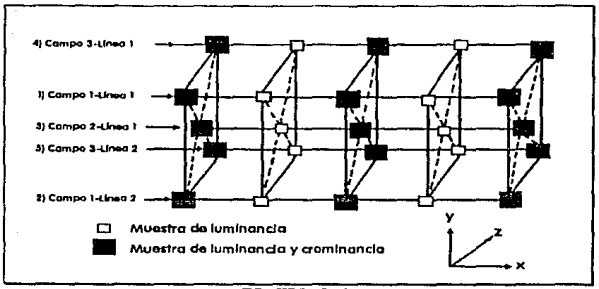


FIGURA 3.2

En realidad, no se muestrea completamente toda la línea (suprimir la parte del borrado y sincronismo, ya que es una información que puede regenerase fácilmente al ser idéntica en todas las líneas) de manera que el número de muestras por período activo de línea para ambos sistemas sea el mismo (720 muestras para la luminancia y 360 para las señales de color). Así se facilitaría la utilización del mismo equipo para ambos sistemas. La diferencia entre el número de octetos o muestras efectivamente transmitidos y el correspondiente a la información de vídeo activos (1728 - 1440 = 288) se emplea para las señales de referencia temporal e identificación necesarias para definir completamente la señal de TV.

3.5 CODIFICACION DE LA SEÑAL

Después de muestrear la señal, debe codificarse, es decir que a cada impulso o muestra se le asigne un código. Según la modulación por pulsos codificados (PCM, "*Pulse Code Modulation*") a cada valor de la muestra se le asigna un código formado por varios bits. Para ello, antes la señal debe cuantificarse.

En la Figura 3.3 se puede observar el diagrama de bloques de un sistema básico PCM.

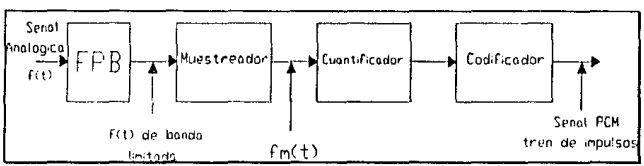


FIGURA 3.3

Diagrama a Bloques de un sistema PCM

3.5.1 CUANTIFICACIÓN

La cuantificación es un proceso no lineal en virtud del cual se limita el número de valores distintos que pueden tomar las amplitudes de la señal muestreada. Para ello se divide la amplitud total de la señal (valor pico a pico) en un número determinado de partes de forma que cada valor muestra pueda asignarse a una de ellas, y éstas estén a su vez representadas por un código de varios bits.

Así pues, si n es número de bits, habrá N niveles de cuantificación ($N = 2^n$), quedando la señal dividida en un número de partes p igual a $N - 1$.

Los pulsos de muestreo se asignarán aproximadamente a cada uno de esos niveles según sobrepasen o no el nivel medio existente entre ellos.

La diferencia que existe entre la señal muestreada y la cuantificada, se traduce como error de cuantificación (ε^{-2}). Este error es propio del cuantificador, existiendo una zona de incertidumbre al pasar de un escalón al siguiente, con lo que se producen fluctuaciones aleatorias en mayor o menor grado. Este error se considera como un ruido (ruido de cuantificación), debido precisamente a que es una distorsión producida de forma aleatoria, ya que en general es imprevisible la señal que entra al cuantificador. Lógicamente cuantos más niveles haya menor aproximación existirá y por lo tanto se tendrá menos ruido.

Para evaluar el valor exacto del ruido de cuantificación, se parte de las siguientes hipótesis:

1) La cuantificación es uniforme. Es decir, que la amplitud entre los niveles de cuantificación es constante (b).

2) El ruido de cuantificación tiene una función densidad de probabilidad uniforme ε igual a $1/b$ en el intervalo $(-b/2, b/2)$.

Entonces:

$$\varepsilon^{-2} = \int_{\frac{-b}{2}}^{\frac{b}{2}} \varepsilon^2 \frac{1}{b} d\varepsilon = \frac{b^2}{12}$$

El valor eficaz del ruido de cuantificación es pues de ± b / $(12)^{½}$.

Como se ve, el valor instantáneo de la señal que se muestra es independiente.

Como una aproximación, se puede decir que el error de cuantificación máximo que se obtiene es de:

± ½ de la amplitud de la señal / (N - 1)

Por ejemplo, en el caso de cuantificar una señal analógica entre 0.3 y 1 Volt (Figura 3.4), con un sistema de 3 bits, es decir ocho niveles de cuantificación, el máximo error obtenido será de ±½ de (1- 0.3)/7 = ± 50 mV que es un error bastante grande.

En el caso de ser un sistema de 8 bits, el error obtenido será de ± 1.37 mV, que ya es aceptable:

n = 8; N = 2^8 = 256; error máximo = ± ½ de (1 - 0.3)/255 = ± 1.37 mV.

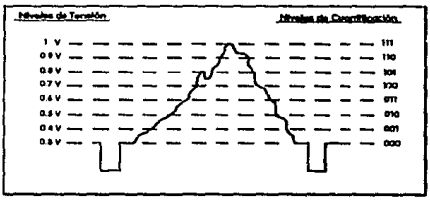


FIGURA 3.4

Cuantificación de una Señal utilizando 3 bits

3.5.2 APLICACION A LA SEÑAL DE TV

Para poder determinar el número de niveles de cuantificación y por lo tanto el número de bits necesarios por muestra, se debe saber el valor máximo permitido de ruido. Partiendo de un valor aceptable para la señal de video de 45 dB como relación S/N (Señal a Ruido, "*Signal To Noise*"), se obtiene que el valor efectivo máximo de ruido permitido es de 3.93 mV:

45 = 20 log Vpico señal TV/V ruido = 20 log 0.7/Vruido

Vruido = 0.7/178 = 3.93 mV

Haciendo el análisis del error obtenido en una señal analógica en diente de sierra, que se digitaliza y luego vuelve a recuperarse en forma analógica, y comparando el resultado obtenido con una señal de TV (monocroma, señal de Barras), se obtiene las siguientes conclusiones:

1) Relación S/N en comparación con una señal con el mismo nivel pico a pico de ruido aleatorio:

   S/N = (6 * n + 18) dB

2) Relación S/N (valor eficaz del ruido) de una señal monocroma de vídeo, codificada entre el nivel de negro y pico de blanco (0.7 V):

   S/N = (6 * n + 10.79) dB

3) Relación S/N, para una señal codificada de barras de color saturadas al 100%, con una excursión entre el fondo de sincronismo y el pico de croma:

   S/N = (6 * n + 5.87) dB

En la Tabla 3.2 se compara los distintos valores de la relación S/N obtenida, dependiendo del número de bits utilizados en la codificación y de la señal de video.

No. de Bits	No. de Niveles	Relación S/N en dB del ruido equivalente pico a pico (monocroma)	Relación S/N RMS (monocroma)	Relación S/N RMS en dB (color)
1	2	24	16.79	11.87
2	4	30	22.79	17.87
3	8	36	28.79	23.87
4	16	42	34.79	29.87
5	32	48	40.79	35.87
6	64	54	46.79	41.87
7	128	60	53.79	47.87
8	256	66	58.79	53.87
n	2n	6n + 18	6n + 10.79	6n + 5.87

TABLA 3.2

Como ya se ha dicho, un valor aceptable de S/N es 45 dB, utilizando la fórmula S/N = 6**n* + 10.79, resulta que el número *n* debe ser mayor o igual que 6 bits, quedando el número de niveles de cuantificación en $2^6 = 64$, que equivale a pasos de 100% / 63 = 1.59%.

Debido a que cuando hay una variación de los niveles de negro de las señales R, G, B mayor a 0.5% (por ejemplo al ajustar los niveles de negro en una cámara) ésta se aprecia visiblemente (ya que aparece un negro colcreado), se ve la necesidad de aumentar el número de bits para poder obtener niveles menores del 0.5%. Así mismo, se aprecia en imágenes con transiciones suaves de luminancia que usando 6 bits, aparece el fenómeno denominado "falso contorno" según el cual esa transición suave aparece visualmente como delimitada por zonas. En el caso de una señal en diente de sierra codificada a 8 bits, se obtendrían 256 niveles de luminancia desde el

negro hasta el blanco con transiciones suaves en las que a efectos visuales no podría decirse donde se inician o terminan cada uno de dichos niveles. Si esa codificación se hace por ejemplo a 4 bits, se obtienen 16 niveles distintos de luminancia, pero en donde sí son perceptibles el inicio y fin de cada uno de ellos, dando la sensación de estar viendo unas "barras". Esto se denomina "falso contorno", fenómeno que tiene lugar justo hasta los 6 bits. Así pues, se escoge el número de 8 bits con lo cual se obtienen 256 posibles niveles de cuantificación y pasos de 0.39%. La relación S/N que se obtiene finalmente (usando la misma fórmula anterior) es de:

$$(6 * 8 + 10.79) = 58.79 \text{ dB}$$

Las conclusiones que se deducen de esta fórmula, son:

- A mayor número de bits se obtiene una mejor relación señal/ruido.

- El término constante de 10.8 depende del valor efectivo del ruido de cuantificación (tal como se ha visto anteriormente).

Según la Recomendación 601 del CCITT:

- Se obtienen 220 niveles de cuantificación, correspondiendo el nivel de negros al 16 y el nivel de blanco de cresta al 235, para la señal de luminancia. El resto de los niveles (del 0 al 16 y del 235 al 255) es una reserva para posibles sobremodulaciones que eventualmente puedan darse (Figura 3.5).

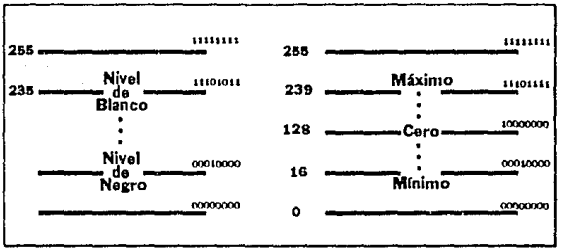


FIGURA 3.5

Niveles de Cuantificación para una Señal de Video

- Para las señales diferencia de color, hay 224 niveles de cuantificación en la parte central de la escala de cuantificación, correspondiendo la señal cero al nivel 128. (Figura 3.5)

- La forma de codificación es PCM, con cuantificación uniforme y 8 bits por muestra, tanto para la señal de luminancia como para las señales de diferencia de color.

3.6 CODIFICACION COMPUESTA O POR COMPONENTES

Cuando apareció la TV en color, la premisa más importante en la que se basó fue en la de mantener una compatibilidad (directa e inversa) con el sistema de transmisión monocromo usado hasta entonces, manteniendo el mismo ancho de banda. Así, la información de color se ubicó en el mismo espectro que ocupaba la luminancia, en su parte alta, mediante una subportadora de color. Para que hubiera la mínima interferencia posible entre ambas informaciones, se eligió el valor de dicha subportadora de color de forma que sus bandas laterales se imbricaran en los espacios de Gray, (como se ha mencionado ya antes en el Capitulo 1), en los que teóricamente la información de luminancia allí era mínima, reduciendo a su vez la anchura de la banda de crominancia.

Esto permitió una transición fácil, sin molestias, del sistema monocromo al de color en los distintos paises.

Con la aparición de equipos basados en tecnología digital y con el aumento de la posibilidad de poder manipular dicha señal de TV (en especial en los centros de producción y postproducción de programas) se han visto agravados los inconvenientes que presentan los actuales sistemas de TV, que básicamente son:

- En la práctica existe una interferencia mutua entre la información de luminancia y la de color, conocida como "cross luminance" (interferencia en los receptores blanco y negro, al tratarse parte del color como información de brillo) y "cross color" (produce "*moiré*" en la zona de altas frecuencias de la luminancia, al tratar como color información de brillo). Todo ello se produce como consecuencia de no

poder separar totalmente ambas informaciones (brillo y color) en los decodificadores.

- Limitación de la resolución horizontal debido a la presencia de información de color en la parte alta de la luminancia.

- La información de color tiene en estos sistemas un ancho de banda reducido (1.3 MHz).

Por todo ello y por otras razones que posteriormente se mencionarán, se manejó la posibilidad de poder usar otros sistemas de transmisión en los cuales no hubiera esa imbricación de ambas informaciones en el mismo espectro, sino que fueran informaciones totalmente separadas. Así el proceso de señal (muestreo y codificación) puede aplicarse tanto a la señal compuesta (PAL, NTSC, SECAM), como a las componentes (señales diferencia de color: R - Y, B - Y o I, Q).

Naturalmente, existen ventajas e inconvenientes en usar uno u otra codificación, las cuales se resumen para ambos casos.

3.6.1 CODIFICACION COMPUESTA

Las ventajas que se obtienen usando esta forma de codificación, son:

- Puede implantarse fácilmente en estudios analógicos, con lo cual constituye una facilidad en la fase de transición de los estudios analógicos a digitales, ya que la evolución de uno a otro sistema será de forma paulatina.

- En PAL y NTSC la codificación de señal compuesta implica una sola vía, por ejemplo en el mezclador, en lugar de tres como ocurría en el otro caso.

Los inconvenientes que aparecen son:

- Antes de transmitir la señal y una vez digitalizada, se debe volver a convertir en analógica, con lo cual no es un método apropiado para tener solamente una única norma mundial estandar, ya que se obtendría nuevamente señales de los distintos sistemas PAL, SECAM y NTSC.

- En edición, se continúa teniendo los problemas propios de la señal PAL o NTSC, es decir, el ciclo de 8 ó 4 campos respectivamente.

- Al existir varios conversores A/D y D/A se introducen perturbaciones y degradaciones de la señal.

3.6.2 CODIFICACION POR COMPONENTES

Las ventajas que existen en este caso son:

- Es un método compatible ya que todos los sistemas de TV en color parten de las señales de luminancia y diferencia de color, con lo cual puede existir un intercambio de dichas señales digitales, dejando un único bloque de conversión a la norma nacional correspondiente al final de la cadena de producción, antes del transmisor. Existe la posibilidad de obtener una norma mundial uniforme, salvo para las frecuencias de trama, por lo que se simplifica la producción y el intercambio de programas al poder usar en todo el mundo muchos elementos de equipos comunes en los estudios.

- Al realizar la grabación de las señales componentes en cinta magnética, se evitarían los problemas de edición propios de las señales compuestas PAL y NTSC. El único ciclo a tener en cuenta sería el de dos campos, debido al entrelazado de la señal de TV (igual que en blanco y negro).

- Se trata por separado la luminancia de la crominancia, por lo que ya no es necesario el decodificador/codificador que introduce un empeoramiento de la señal, disminuyendo la calidad de la misma.

La desventaja que existe en este método respecto al anterior es que ahora en vez de tener que procesar una sola señal (la compuesta) son tres las que hay que tratar.

Es evidente que con los últimos desarrollos tecnológicos en el campo del vídeo, este sistema tiene más ventajas que el anterior, teniendo en cuenta que se mejora la calidad. La utilización creciente del proceso digital de señal en los equipos de vídeo, la aparición de los equipos de grabación ENG basados en el proceso de las señales de componentes y la adopción de la norma MAC europeo para la radiodifusión directa por satélite, así como para la transmisión de televisión con definición extendida, dan a este sistema un futuro prometedor.

3.7 NORMA 4:2:2

La Comisión Consultiva Internacional de Telefonía y Telegrafía (CCITT), en su Recomendación 601 y en el informe 629-2, citado anteriormente, propone que se adopten los siguientes parámetros, cuyo resumen se puede observar en la Tabla 2, como base para las normas de codificación digital para estudios de TV, para ambos sistemas (625 y 525 líneas). Puede decirse que se trata de una norma de TV digital (un estandar de TV digital por componentes), que fue aceptada por la UER, la SMPTE y la OIRT, por lo que, sin duda alguna, se trata de una norma mundial.

Los parámetros que se definen hacen referencia a la codificación digital de las señales en componentes (Y, R-Y, B-Y o R, G, B), y son los siguientes (Tablas 3.3, 3.4, 3.5 y Figuras 3.6, 3.7 y 3.8).

- Señales que se codifican
- Frecuencia de muestreo
- Número de muestras/línea completa
- Estructura de muestreo
- Sistema de Codificación y número de bits/muestra
- Número de muestras/línea activa digital
- Número de niveles de cuantificación y su correspondencia con la señal de video
- Anchos de banda necesarios
- Tipos de Filtrado
- Relación entre la línea activa digital y la referencia analógica de sincronismo

Señales Codificadas	Luminancia y diferencia de color (derivadas de las señales primarias con corrección de gama)
Frecuencia de Muestreo	13.5 MHz (luminancia) 6.75 MHz (diferencia de color)
No. de Muestras de luminancia/ número total de líneas	864 (para sistemas 625/50) 858 (para sistemas 525/60)
Estructura de Muestreo	Ortogonal: repetitiva en cada línea, en cada trama y en cada imagen; las muestras de diferencia de color coinciden mutuamente en la ubicación y también con la 1a., 3a., 5a., etc. muestra de la señal de luminancia en cada línea.
Forma de Codificación	PCM con cuantificación uniforme, binaria positiva para la luminancia, binaria con desplazamiento para la diferencia de color, 8 bits por muestra en cada caso.
Correspondencia entre los niveles de la señal de vídeo y los niveles de Cuantificación	La señal de luminancia ocupa 220 niveles de cuantificación, correspondiendo el negro al nivel 16 en una escala de 0 a 255 y el blanco al nivel 235 en una escala de 0 a 255. Cada señal de diferencia de color ocupa 224 niveles en la parte central de la escala de cuantificación, correspondiendo la señal cero al nivel 128 en una escala de 0 a 255.
No. de Muestras por línea activa digital	720 (luminancia) y 360 (diferencia de color)
Requisitos de la Banda de Paso	Canal de luminancia nominalmente plano hasta 5,5 MHz como mínimo, y atenuado con 12 dB como mínimo a 6.75 MHz. Canales de diferencia de color nominalmente planos hasta 2.75 MHz como mínimo, y con una atenuación de 12 dB como mínimo a 3.375 MHz.
Información sobre Filtros	Los filtros analógicos necesarios para la conversión A/D necesitan una atenuación de supresión de banda superior a 45 dB a 8 MHz, para la señal de luminancia, y a 4 MHz para la señal de diferencia de color. Las características espectrales de la señal de diferencia de color serán conformadas por un filtro lento de corte progresivo insertado en el codificador compuesto y en los monitores de imagen. La igualación (sin x = x), es necesaria, sólo se aplica al final de la cadena digital, donde la señal se convierte a analógica.
Relación entre la línea activa digital y la referencia analógica de sincronismo	Véase la Figura 3.6.

TABLA 3.3

La expresión "norma 4:2:2", se refiere a la relación de las frecuencias de muestreo de las señales de luminancia y diferencia de color, que como es sabido, es el doble para la Y que para R-Y y B-Y. El origen de esta expresión proviene de cuando se estaba investigando sobre la televisión digital y se trabajaba con señales compuestas. Se eligió entonces como frecuencia de muestreo la de cuatro veces la subportadora de color (17.72 MHz para los sistemas de 625 líneas PAL y 14.32 MHz para los de 525 NTSC). Posteriormente, cuando se propuso la codificación por componentes y tras la investigación y estudios correspondientes en los que se decidió el uso de la frecuencia de 13.5 MHz para la luminancia y 6.75 MHz para las señales de diferencia de color, se le dió el nombre de "norma 4:2:2" en vez de 13.5:6.75:6.75 por ser más corto y cómodo.

Sistema		
	525 líneas 60 campos muestras/µs	625 líneas 50 campos muestras/µs
Duración del intervalo entre el origen del tiempo O_H y el principio de la Línea Activa	122/9.037	132/9.778
Duración de la Línea Activa	720/53.33	720/53.33
Duración del relleno Activo.	16/1.185	12/0.889
Total	858/63.555	864/64

TABLA 3.4

Valores de las señales de vídeo
$Y' = 0.299\ R' + 0.587\ G' + 0.114\ B'$
$C'_P = 0.713\ (R' - Y')$
$C'_B = 0.564\ (B' - Y')$

TABLA 3.5

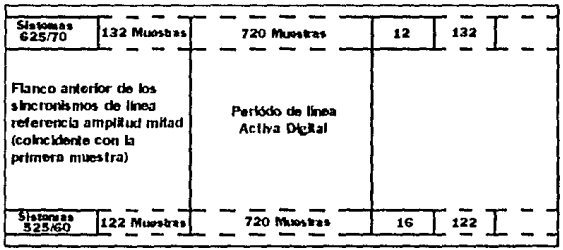


FIGURA 3.6

Relación entre la línea Activa Digital y la Referencia Analógica de Sincronismo

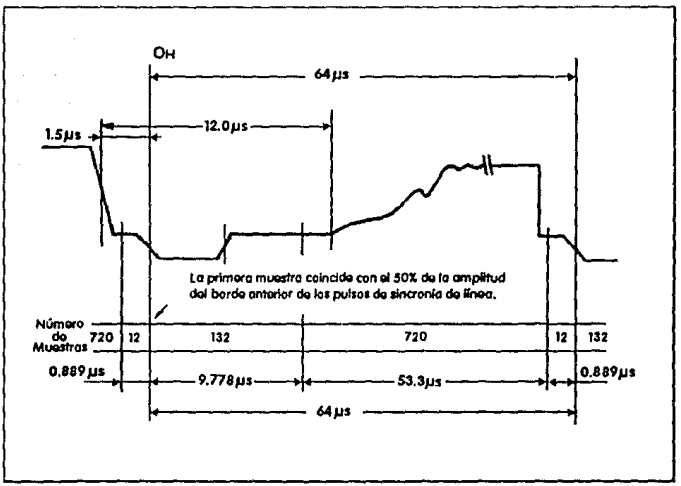


FIGURA 3.7

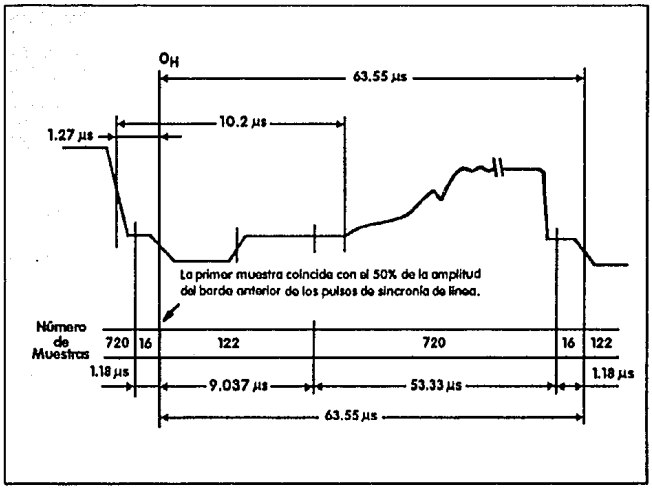

0H
63.55 µs
10.2 µs
1.27 µs
La primer muestra coincide con el 50% de la amplitud
del borde anterior de los pulsos de sincronía de línea.
Número
de
Muestras
720
16
122
720
16
122
1.18 µs
9.037 µs
53.33 µs
1.18 µs
63.55 µs

FIGURA 3.8

3.8 VENTAJAS E INCONVENIENTES DE UN SISTEMA DE TV DIGITAL

Es posible que en un futuro próximo solamente se trabaje con señal de vídeo digital. Los equipos de TV que ofrecen los fabricantes hoy día, no sólo trabajan interiormente de forma digital, sino que sus salidas y/o entradas también son para señales digitales de TV.

Las ventajas que un sistema de TV digital puede tener provienen del hecho de que la información está en una señal digital en vez de analógica. Ello implica que:

- Hay más inmunidad al ruido y a las interferencias.
- Se puede memorizar.
- Puede manipularse y procesarse.
- Aumente la fiabilidad de los equipos (se les puede incorporar sistemas de prueba y detección de fallas).

Esto permite obtener más copias en grabación de la misma señal con una mejor calidad que en caso de ser analógica y la posibilidad de ser procesada y manipulada, con lo cual se obtiene multitud de efectos que serían imposibles obtener con la señal analógica. Así la señal de TV, una vez digitalizada y almacenada en una memoria digital, pueda ser escrita y leída a distinta velocidad, comprimida, leída en orden distinto al escrito, etc.

En cuanto a los equipos de producción con tecnología digital, son por ahora físicamente más grandes que sus equivalentes analógicos y con un consumo mayor. Por ejemplo, se puede citar que el mezclador digital del estudio de Rennes (Francia) tiene un consumo de 2kW cuando su equivalente analógico consume 1kW.

La tecnología digital lleva asociados una serie de problemas como son el ruido de cuantificación, interferencias entre bits adyacentes, etc.

El gran inconveniente desde el punto de vista de transmisión es, sin lugar a dudas, el gran ancho de bando o velocidad binaria a que da lugar la digitalización de la señal de TV y que hace imposible su transmisión al no existir canales normalizados con esta capacidad.

La Comunidad Europea, dentro del programa RACE, está investigando sobre la RDSI, Red Digital de Servicios Integrados (ISDN Integrated Services Digital Networks) de banda ancha, según la cual un único conductor (fibra óptica) sirve para transportar cualquier clase de información (telefonía, TV, videotexto, telefax, etc.). Para ello es necesario poder reducir la velocidad binaria de la señal de televisión; diversas empresas que están investigando a fondo este tema aseguran que es posible reducirla de 216 Mbits/s (para un sistema de 625 líneas y según la Recomendación 601 de la CCITT) a unos 68 Mbits/s, sin pérdida notable de calidad, explotando la redundancia temporal que posee la señal de TV y transmitiendo sólo las diferencias que existen de las muestras respecto a las circundantes (tridimensionalmente, es decir teniendo en cuenta las dos dimensiones espaciales, vertical y horizontal y una temporal).

Existen técnicas para reducir el ancho de banda de la señal de TV, basándose en el hecho de que la información existente en las imágenes es altamente redundante, tanto en el espacio como en el tiempo, ya que la información de un elemento de imagen estadísticamente se parece a la de los elementos adyacentes (espacio) y a la de los elementos de las imágenes anteriores y posteriores (tiempo). A esto añadimos

el hecho de que el órgano visual humano es menos sensible a la información de color (necesitamos menos definición que para la información de brillo) ya los objetos en movimiento.

Cabe mencionar que el único sistema que existe para explotar las técnicas de reducción de información de la TV en HDTV es el sistema MUSE (japonés), el cual logra que una señal de vídeo con un ancho de banda inicial de 30 MHz se reduzca a 8.1 MHz. Para ello utilizan técnicas de muestreo sub-Nyquist, obteniendo una señal capaz de representar 1125 líneas verticales , alrededor de 1900 horizontales, 60 campos con entrelazado de 2:1 y con una relación de espectro de 5:3.

Por su parte, Europa y los Estados Unidos, trabajan con proyectos ya bastante avanzados (especialmente en Europa mediante el proyecto EUREKA 95) investigando un sistema propio. Europa quiere poner en marcha un sistema de televisión de alta definición con motivo de los Juegos Olímpicos de 1992 en Barcelona. También se ha de utilizar, aunque de forma parcial, durante los Campeonatos Mundiales de Atletismo.

Respecto a la situación de los Estados Unidos, en el mes de mayor de 1989, un grupo de empresas electrónicas, solicitó al Congreso que aportara fondos por valor aproximado de unos 50,000 millones de dólares para acelerar las investigaciones y el desarrollo de su propio sistema de televisión de alta definición.

3.9 CAPACIDAD DE MEMORIA

La idea básica de la TV digital es en su concepción muy sencilla: a cada elemento de imagen se le asigna o asocia un valor digital que representa a la luminancia y crominancia asociada, de forma que posteriormente ese valor pueda ser almacenada en una memoria. Una vez almacenados todos los elementos de imagen correspondientes a un cuadro o campo, se puede procesar o manipular.

El volumen de información que existe en un cuadro o imagen es enorme. Este ha sido precisamente la razón que ha provocado un lento avance de la tecnología de la imagen digital a lo largo de tanto tiempo.

Un cuadro completo ocupa un período de 40 milisegundos y consta de dos campos entrelazados, cada uno de 312.5 líneas. Cada una de ellas consta de más de Setecientos elementos de imagen o pixels, que proporcionan información de brillo, saturación y matiz o tinte.

Las muestras se toman a razón de 13.5 millones por segundo para la luminancia, 6.75 millones por segundo para las señales de diferencia de color y a cada una de ellas se le asocia un valor binario de 8 bits.

Algunos datos interesantes (del sistema PAL) y fácilmente deducibles (resumidos en la Tabla 3.6) mediante simples operaciones aritméticas son:

- Número de bytes a transmitir en una línea: 864 + (2 * 432) = 1728.

- En un segundo se tienen: 1728 bytes * 625 líneas * 25 Hz = 27 millones de bytes o muestras = 27 Mbytes = 216 Mbits.

- Número de bytes de vídeo activos = 720 + (2 * 360) = 1440.

En los sistemas de 625 líneas hay 575 con información de imagen. Como es conveniente tratar las dos medias líneas de la parte superior e inferior de la imagen como líneas enteras, este número se incrementa a 576. Así pues:

- En un segundo, se tienen: 1440 * 576 * 25 = 20´736,000 bytes o muestras activas, es decir 165,9 Mbits. Esta sera la memoria necesaria para poder almacenar un segundo.

- Memoria necesaria para almacenar un cuadro: 165.9 Mbits/25 = 6.636 Mbits.

- Para poder almacenar un minuto de un determinado programa se necesita una capacidad de: 60 * 165.9 Mbits = 9954 millones de bits = 9.954 Gbits = 1.25 Gbytes.

Número de octetos que se transmiten en 1 línea	1,728
Velocidad de transmisión binaria	27 Mbytes
Número de bytes de vídeo activo	1,440
Velocidad de transmisión (muestras activas)	165.9 Mbits
Capacidad de memoria para almacenar 1 cuadro	6.636 Mbits
Capacidad de memoria para almacenar 1 segundo	165.9 Mbits
Capacidad de memoria para almacenar 1 minuto	1.25 Gbytes

TABLA 3.6

Actualmente las grabadoras de vídeo de estado sólido (SSVR), utilizan memorias dinámicas de acceso aleatorio DRAM que mejoran y ofrecen una mayor flexibilidad y seguridad que los discos duros, debido entre otras cosas a que no poseen elementos mecánicos y el acceso es casi instantáneo a cualquier trama (*frame*), no existe prácticamente deterioro por la reproducción de imágenes, etc. Sus ventajas vienen condicionadas por su elevado costo, lo cual reduce enormemente, por ahora, su posible utilización, siendo privilegio solamente de empresas con gran capacidad económica. Su principal uso es lógicamente el campo de postproducción, edición profesional y generación de efectos especiales.

En las aplicaciones de postproducción que se usan hoy día, las capacidades de almacenamiento están en torno a los 90´s, es decir 2250 cuadros. Actualmente las grabadoras de vídeo de estado sólido incorporan tarjetas de memoria alimentadas con DRAM de 1 Mbit (por ejemplo un grupo de cuatro tarjetas proporcionan 2147.5 Mbits de memoria) con una capacidad para varios grupos de tarjetas (con seis grupos de tarjetas de memoria se dispone de una capacidad para 12.885 Gbits).

Están a punto de incorporarse a estos equipos, DRAM de 4 Mbits, con lo cual la capacidad de almacenaje se extenderá a más de 5 minutos, desplazando a medios de grabación mecánica en bastantes aplicaciones.

3.10 DCT -TRANSFORMADA DEL COSENO DISCRETA-

A través de los años se han ido desarrollando técnicas y algoritmos cada vez más poderosos para el estudio de las señales. Un ejemplo típico, es la Transformada Discreta del Coseno (DCT); la cual, resulta ser de gran importancia, sobre todo en el campo del Procesamiento Digital del Señales (*"Digital Signal Processing"*, DSP), por lo que es más utilizada en áreas como la electrónica en comunicaciones que en campos como la física.

A pesar de que la DCT fue desarrollada a partir de la Transformada Rápida de Fourier (*"Fast Fourier Transform"*, FFT), se han desarrollado algoritmos reales y recursivos, lo que han dado por resultado un algoritmo más eficiente y poderoso para el cálculo del DCT.

Las transformadas son en particular, transformaciones integrales y son usadas primordialmente para la reducción de la complejidad de problemas matemáticos.

La DCT en particular, tiene 4 expresiones para calcularse, las cuales son:

DCT-I:

$$x^{c(1)}(m) := \sqrt{\frac{2}{N-1}}\, k_m \sum_n k_n\, x(n)\, \cos\left[\frac{m\,n}{N}\right] \qquad m = 0, \ldots, N$$

DCT-II:

$$X^{c(2)}(m) := \sqrt{\frac{2}{N}}\, k_m \sum_n x(n) \cos\left[\frac{(2n+1)\,m}{2N}\right] \qquad m = 0, \ldots, N-1$$

DCT-III:

$$X^{c(3)}(m) := \sqrt{\frac{2}{N}}\, k_m \sum_n k_n\, x(n) \cos\left[\frac{(2m+1)\,n}{2N}\right] \qquad m = 0, \ldots, N-1$$

DCT-IV:

$$X^{c(4)}(m) := \sqrt{\frac{2}{N}}\, k_m \sum_n x(n) \cos\left[\frac{(2m+1)\,(2n+1)}{4N}\right] \qquad m = 0, \ldots, N-1$$

Existen varias aproximaciones computacionales para calcular el DCT, las cuales se clasifican como DCT vía FFT, "*sparse matrix factors*", algoritmo de DIT ("*Decimation in Time*"), algoritmo de DIF ("*Decimation in Frequency*"), DCT vía otras transformadas (Hartley, Walsh, Karhunen-Loeve, etc.) y otras. De entre todos estos algoritmos no se podría seleccionar uno para considerarlo como superior sobre los otros, pues se debe de tomar en cuenta el número de adiciones y multiplicaciones (tanto reales como complejas), la estructura de las gráficas de flujo, los mapas de índices de entrada contra salida y la recursividad. A continuación se presenta una tabla con las ventajas y desventajas de estos algoritmos.

ALGORITMO	VENTAJAS	DESVENTAJAS
Vía N puntos	Fácil de Implementar utilizando rutinas de FFT.	Lento.
FFT Recursivo	Rápido y Recursivo.	Indice de Mapas Muy Complejo.
Sparse Factors	Razonablemente Rápido.	Indice de mapas muy complejo, no recursivo.
DIT (Decimation in Frequency)	Rápido, Decimación en Tiempo Recursiva.	Indice de Mapas Complejo.
DIF (Decimation in Time)	Rápido, Decimación en Frecuencia Recursiva.	Indice de Mapas Complejo.
Vía WHT (Walsh-Hartley Transform)	Facilmente Implementable, Indice de Mapas Sencillo.	Necesita conversión de matrices, lento para N > 16.
Vía DHT (Discrete Harltley Transform)	Fácil de Implementar utilizando rutinas de FFT.	Lento.
PFA	No restringido a grado 2.	Indice de Mapas en Extremo Complejos.
Multiplexión	Rápido, Recursivo e Indice de Mapas Sencillo.	Requiere de cambios por la multiplexión.
Rotores	Utiliza una sola unidad de proceso, buena estructura, Indice de Mapas Sencillo.	Demasiado Lento.

De los algoritmos de la tabla anterior se ha utilizado una modificación del algoritmo DCT-II y del IDCT-II, los cuales son rápidos para N < 256.

DCT-II:

$$Xu_u := \sqrt{\frac{2}{n}}\, C_u \sum_k Xk_k \cos\left[\frac{(2k+1)u}{2n}\right]$$

IDCT-II

$$Xk_u := \sqrt{\frac{2}{n}} \sum_k C_k\, Xu_k \cos\left[\frac{(2u+1)k}{2n}\right]$$

Una ventaja substancial de todos los algoritmos anteriores es su facil desarrollo bidimensional partiendo de

la transformada unidimensional seguida de una parte recursiva. El algoritmo más eficiente para calcular la DCT bidimensional consiste en descomponer en bloques la DCT y posteriormente es calculada por la Transformada Discreta de Fourier ("*Discrete Fourier Transform*", DFT) bidimensional. Otra ventaja que presenta la DCT bidimensional es su bajo porcentaje de error, a continuación se muestra una gráfica comparativa con otras transformadas Figura 3.8.

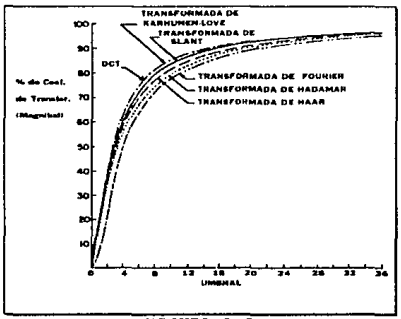


FIGURA 3.8

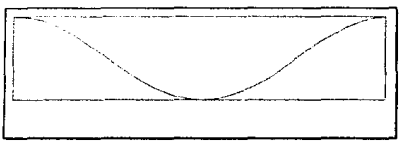
Comparación entre algunas Transformadas

Recientemente con el desarrollo de circuitos integrados VLSI (Very Large Scale of Integration) se han desarrollado algunas transformadas en estos circuitos, y en particular, algunos fabricantes han desarrollado sus integrados con la DCT, pues encontraron que como tiene gran relación con otras transformadas y posee algoritmos muy rápidos, puede trabajar en los circuitos y dar resultados en tiempo real. Por lo que la DCT bidimensional se ha tomado ya como un estandar internacional para el procesamiento de imagenes.

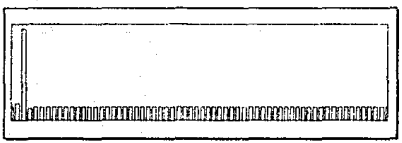
Algunas de las aplicaciones directas en las que se ha utilizado la DCT son las siguientes:

- Filtrado de Señales
- Codificación de Voz
- Codificación de Video
- Compresión de Datos
- Clasificación Topográfica
- Reconocimiento de Patrones
- Codificación de Señales de HDTV
- Transmisiones Progresivas de Imágenes
- Análisis de Superficies

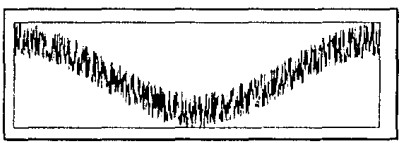
A continuación se muestran algunos ejemplos obtenidos con el algoritmo del DCT:



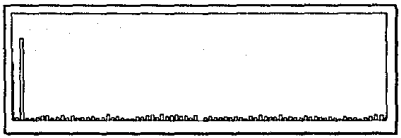
Gráfica en Tiempo de
una Señal Cosenoidal
FIGURA 3.9



Gráfica DCT de la
Señal Cosenoidal
FIGURA 3.10

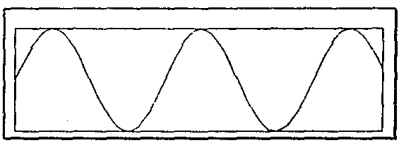


Gráfica en Tiempo de
Señal Cosenoidal + Ruido
FIGURA 3.10

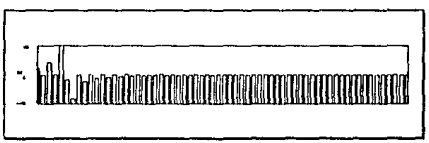


Gráfica DCT de la
Señal Cosenoidal + Ruido
FIGURA 3.11

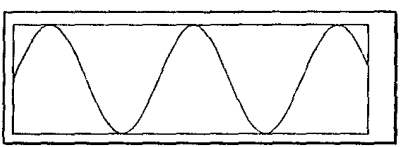
RESULTADOS OBTENIDOS CON LA IDCT-II:



SEÑAL ORIGINAL (GRAFICA EN TIEMPO)
FIGURA 3.13



SEÑAL TRANSFORMADA CON DCT-II
FIGURA 3.14



SEÑAL ANTITRANSFORMADA CON IDCT-II
FIGURA 3.15

3.11 PROCESAMIENTO DIGITAL DE SEÑALES PARA VIDEO

Avances en dispositivos de tecnologías de semiconductores están marcando lo que se conoce como tiempo-real en Procesamiento Digital de Señales (DSP) para señales de video en forma práctica. Recientemente en 1975, un equipo experimental enfocado a la compresión y decompresión de señales de TV en tiempo real utilizando técnicas de DSP desarrollo dos equipos llamados "*racks plus*" y "*chest high gabinets*" para las tramas de memoria de tamaño completo ("*frame memories*"). Cada trama de memoria es una unidad construida en esa época por NEC Central Research Laboratories, utilizando 2000 chips de 1-Kbit de memoria -lo más grande en ese tiempo. Ahora, debido a que existen mejores condiciones de fabricación, están disponibles en el mercado, chips de 1 Mbit de memoria y circuitos DSP sofisticados, realizados en LSI. El DSP para video se ha vuelto una realidad, esta encontrando aplicaciones en áreas como las cadenas televisivas, comunicaciones y aún en aparatos electrónicos de uso diario.

La tecnología en DSP fue aplicada primeramente a señales de voz. Múltiples simulaciones por computadora de variadas operaciones utilizando DSP, fueron iniciadas en los 50's, y operaciones en tiempo real en hardware fueron introducidas en los 60's.

El ancho de banda para las señales de voz es muy angosta, suficiente para que las operaciones con DSP se puedan realizar en tamaños razonables de hardware a un precio razonable. La calidad telefónica para el habla tiene un ancho de banda de 3.1-kHz y se muestrea típicamente a una frecuencia de 8-kHz, lo que da como resultado 125 μs de periodo de muestreo. Esto producue un número muy cómodo de operaciones para procesamiento de señales en circuitos lógicos, aún en los años 60's.

La memoria requerida para señales de voz es razonablemente pequeña. Con una frecuencia de muestreo de 8-kHz, y una cuantificación (bits por muestra) de 8, guardar 10 ms de habla requerirá solo de 640 bits de memoria. Esto también era realizable en los 60's.

3.12 EL COSTO DEL PROCESAMIENTO DIGITAL DE SEÑALES PARA VIDEO

Las señales de video tienen un ancho de banda mayor que el de las señales de habla, por lo que requiere mayor velocidad para las operaciones de DSP, como también una capacidad de memoria mayor. Las señales de televisión, por ejemplo, tienen un ancho de banda de 4-MHz, y se muestrean típicamente a una frecuencia de 14.3-MHz. El intervalo de muestreo es de solo 70 ns -menor que una milésima de los intervalos para las señales con voz-, el cual es comparable con el tiempo de retardo de propagación t_{pd} (propagation delay time) para una compuerta lógica de los años 60's; solo una cantidad limitada de operaciones DSP de tiempo real eran posibles entonces. Es más, los requerimientos de memoria hacían difícil, más no imposible, el aplicar tecnología de DSP para señales de video. Aún para simulaciones con computadora no había mejora para tiempo real, el DSP de señales de video consumian gran cantidad de tiempo y memoria, por lo que solo eran utilizadas en áreas muy limitadas.

Los progresos recientes en componentes de estado sólido en dispositivos lógicos de alta velocidad, mayor capacidad de memorias y sofisticados procesadores de señales digitales, han cambiado dramáticamente la situación. Ahora el hardware, de dimensiones razonables, está disponible a un costo razonable, un rango mucho más ancho para funciones de DSP están disponibles para señales de video [1].

3.13 APLICACIONES

De las muchas señales de video diferentes que existen, las señales de TV son las más comunes. Para un mayor entendimiento de como el Procesamiento Digital de Señales puede ser aplicado a estas señales, será de gran ayuda si recordamos los estandars de TV [2]. Los standards para TV a color son los NTSC, PAL y SECAM. En cada uno de éstos sistemas, una trama de video consiste en dos campos. Las líneas de barrido para el segundo campo caen entre las líneas del primer barrido Figura 3.16. Esto es llamado "entrelazado" y estos sistemas son llamados como sistemas de TV entrelazados.

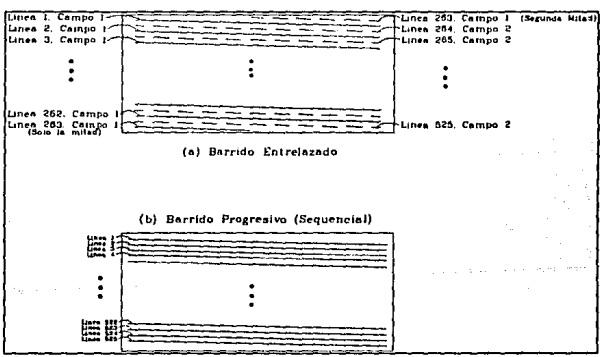


FIGURA 3.16

En el sistema NTSC, existen 30 tramas (cuadros) por segundo y 525 líneas por trama; en los sistemas PAL y SECAM, existen 25 tramas por segundo y 625 líneas por trama. Entrelazando partes del ancho de banda de transmisión a la mitad. En sistemas de TV a color, la señal de luminancia y

la señal de croma están combinadas en la transmisión. Esta señal compuesta tiene un ancho de banda de 4-MHz.

3.13.1 CADENAS DE TELEVISION

Las cadenas de televisión utilizan señales analógicas de radio. Pero antes de que sea una señal radiada, el programa de TV debe ransmitirse desde el estudio hasta la estación transmisora. Esta transmisión puede ser tanto analógica como digital. La calidad requerida para la señal radiada debe ser muy alta. La transmisión en forma analógica inevitablemente degrada la calidad, debido a la no linealidad y el ruido inherente en los canales analógicos.

Las transmisiones digitales ofrecen un alta calidad para las señales de video. Para que esto pueda ser realizado, la señal de TV debe convertirse a un formato digital con suficientes bits para cubrir el rango dinámico de la señal y el ancho de banda.

Digitalizando una señal de video compuesta con un muestreo de 8-bits, a una frecuencia de muestreo de 13.5-MHz, producirá un rango de alrededor de 100-Mb/s. La comunicación en la red actual no siempre provee canales con un ancho de banda tan grande con un costo razonable para su transmisión. Aún más, para realizar transmisiones digitales a un costo competitivo, con transmisiones analógicas, se requiere una reducción en la velocidad de transmisión (*"bit rate"*). No es de sorprenderse, que se empleen codificadores digitales para reducir este rango.

Un sistema de codificación digital con la calidad requerida para las cadenas televisivas, es el conocido como HO-DPCM (*"Higher Order-Differential Pulse Code Modulation"*) [3]. En el sistema HO-DPCM, la señal de video compuesta es convertida a una señal digital, la cual es comprimida

utilizando un código de predicción. En los códigos de predicción, la señal generada representa la diferencia que existe entre la señal de entrada y la señal predicha. La señal de diferencia posee un rango dinámico mucho menor que la señal de entrada, por lo que se puede transmitir con una menor velocidad de transmisión ("*bit rate*"). Utilizando el sistema HO-DPCM, se reduce el rango de transmisión requerida hasta un 50% -hasta 45-Mb/s o menos- reteniendo la alta calidad de video. Los estandars americanos para una señal codificada a 45-Mb/s están en discusión. Muchos otros codificadores algorítmicos, incluyendo el DPCM y el DCT, se están estudiando.

Otras aplicaciones con DSP para cadenas televisivas son los equipos de DVE ("*Digital Video Effect*" o Efectos de Video Digital) y los convertidores de Televisión Estandar ("*Television Standards Converter*", TSC) [4]. Estos requieren de un procesamiento complicado, el cual se realiza con DSP. El DVE ha sido desarrollado para producir escenas atractivas en programas de TV. Bien conocidas son el escalamiento del tamaño de un objeto, rotación en 3D y la variación de la perspectiva de una imagen. Tecnología con DSP es indispensable para realizar estos efectos en tiempo real.

Los DVE consisten de una función procesada de la imagen y una trama de memoria, implementando efectos de video tridimensionales -usualmente combinaciones de operaciones como posición, escalamiento y rotación- en operaciones de matriz de cuatro por cuatro para cada elemento de la imagen [5] (pel).

Las operaciones de DVE en tiempo-real, las operaciones matriciales pueden ser reducidas a las siguientes operaciones simplificadas:

$$X = (A*x + B*y + C) / (P*x + Q*y + K) \qquad \text{Eq. 3.1}$$

$$Y = (D*x + E*y + F) / (P*x + Q*y + K) \qquad \text{Eq. 3.2}$$

donde:

"x" e "y" son los elementos de video (pel) originales en las direcciones horizontal y vertical, respectivamente.

De "A" a "Q" son constantes

"X" e "Y" son las direcciones de los elementos de video (pel) a determinar.

Calculando estas sencillas ecuaciones y realizándolo en una trama de memoria es casi imposible hacerlo con circuitos analógicos. Aún más, el DSP es esencial para generar sofisticados efectos de video.

La necesidad de convertidores de barrido en televisión ("*Television Scan Converters*", TSC), es la segunda necesidad de las cadenas de Televisión. Esta necesidad se ha incrementado con el crecimiento en el ancho mundo de la programación y las transmisiones por satélite. Los TSC son necesarios para convertir el número de líneas de barrido y el número de tramas por segundo del sistema original de TV a aquellos del sistema blanco y negro. Algunos métodos analógicos han sido utilizados por muchos años, porque son fáciles de implementar en tiempo real, pero métodos digitales han demostrado su superioridad en calidad gracias a los avances en las memorias digitales en línea y los controles de caída (delay). Los TSC digitales realizan interpolaciones y funciones decimales para tramas y líneas con una trama de memoria y un filtro digital.

3.13.2 EL PROCESAMIENTO DIGITAL DE SEÑALES PARA COMUNICACIONES

En las cadenas de televisión se requiere que la degradación por transmisión del video sea virtualmente invisible. En comunicaciones, por otro lado, se requiere que la calidad de la imagen pueda variar, dependiendo de la aplicación, lo más ventajoso desde el punto de vista de costo contra beneficio, mientras más sean las aplicaciones de las comunicaciones de video como en teleconferencias y los video teléfonos. El, movimiento completo de video e imágenes de video estáticas son usadas, de acuerdo a la aplicación.

El movimiento completo en video crea la ilusión de conversación real en teleconferencias, pero produce incrementos en los costos de transmisión. Aún más, las técnicas de codificación digital son indispensables para este tipo de aplicación. Para obtener alta eficiencia de compresión, los algoritmos digitales para codificarlos son más complejos, también se requieren mayores memorias, lo que implica que el hardware es mayor y más costoso.

Avances en tecnologías LSI han sido cruciales en el proceso de reducir el rango de transmisión del video requerido para las teleconferencias (Figura 3.17). Este rango de transmisión abarca desde los 64 kb/s hasta los 2 Mb/s. En este rango, varias tecnologías en DSP son utilizadas para codificar video. Para trasmitir en los canales primarios a un rango de 1.5 Mb/s en los países alineados a los estandars americanos y de 2 Mb/s en los países alineados a los estandars europeos, códigos predictivos de entre tramas ("*interframe*") son usadas para compresión. "*Part-1 codec*" [6], que es el estandar de la CCITT para un rango de transmisión de 2 Mb/s, aplica una simple técnica de DPCM ("*Digital Pulse Code Modulation*") y

un código predictivo de entre trama. "*Part-3 codec*" [6], que es de 1.5 Mb/s, aplica un código de predicción de compensación de movimiento con de entre tramas.

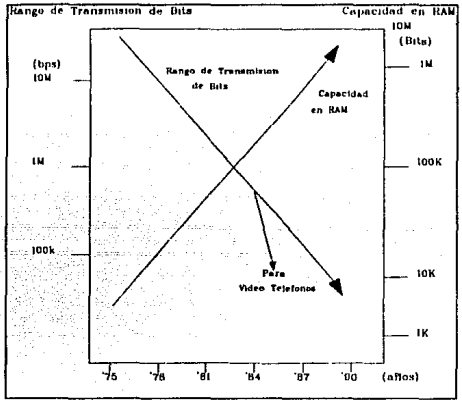


FIGURA 3.17

Para rangos subprimarios (p*64 kb/s; p = 1 hasta 30) de codificación de video, un método con código híbrido, en donde el código DCT (Discrete Cosine Transform) [7] es aplicado a lo largo de la predicción compensada del movimiento de entre tramas [8], es ahora discutida para su estandarización en la CCITT [9]. En los códigos predictivos de entre trama, la diferencia entre dos tramas consecutivas es calculada y codificada. (Una trama en memoria es requerida para guardar la señal anterior). En el código predictivo de compensación del movimiento entre tramas, la diferencia entre las tramas es calculada tomando en consideración los desplazamientos de objetos, lo cual permite futuras implementaciones y una reducción en el rango de transmisión (bit rate).

El código DCT (ya antes mencionado en este mismo Capítulo), es utilizado para reducir la redundancia por espacio contenida en una imagen. La DCT es una transformada ortogonal que transforma las señales de video espaciales al dominio de la frecuencia mediante la siguiente ecuación:

$$X(k)=\frac{2}{N}\cdot c(k)\sum x(m)\cdot\cos\left[(2m+1)\right] \qquad \text{Eq.3.3}$$

para:

k = 0,1,...,N - 1

donde:

$$c(k)=\frac{1}{\sqrt{2}} \qquad (k = 0)$$

$$= 1 \qquad (k = 1,2,...,N - 1)$$

donde:

$x(m)$ son las muestras de la señal de video y $X(k)$ son las señales transformadas. La operación DCT inversa, usada para reconstruir las señales de video espaciales, está dada por una ecuación similar. Para la CCITT el rango de codificación sub primario de diferencia de señales de entre trama esta dividido en sub blocks de 8 por 8 ó de 16 por 16 pels. La DCT es aplicada independientemente a sub bloques individuales en ejes verticales y horizontales. Los componentes resultantes del DCT son normalmente en frecuencias más bajas. Aún más, las componentes de menor frecuencia son codificadas al principio y, en muchos casos, las componentes de alta frecuencia son descartados. Este tipo de proceso puede ser llevado a cabo usando únicamente tecnologías de tipo DSP.

3.13.3 ARTICULOS ELECTRONICOS DE CONSUMO

Por muchos años, las tecnologías de video digital han estado restringidas por ser demasiado costosas para aplicaciones en artículos electrónicos de consumo de uso cotidiano, pero sin embargo los progresos recientes en las tecnologías LSI han cambiado las cosas. La razón principal para incorporar el video digital es el de mejorar la calidad de la señal de video.

Los receptores de TV convencional tienen los siguientes modos de degradación de la calidad de la señal de video: (1) Radiaciones de color en baja frecuencia en patrones muy finos y repetitivos; (2) El llamado "*Line flicker*" a 30-Hz en NTSC y 25-Hz para PAL y SECAM en áreas laterales con cambios bruscos de color; (3) Resolución vertical y líneas de estructura limitadas, particularmente en pantallas largas con sharp beam spots; (4) Resolución horizontal limitada para aproximarse a 330 líneas de TV; y (5) El fenómeno de fantasmas y nieve.

La degradación en el punto 1 es causada por imperfecciones en la señal de separación de luminancia/crominancia. Las degradaciones por los puntos 2 y 3 son causadas principalmente por el entrelazado. Las degradaciones por los puntos 3 y 4 son causadas por limitaciones de ancho de banda y la degradación en el punto 5 es el resultado de imperfecciones en los sistemas de transmisión.

Un sin número de tecnologías han sido desarrolladas para solucionar estos problemas. Las técnicas de filtros *Comb* son usadas en conjunto con los Televisores convencionales para reducir la interferencia por cruzamiento de color (degradación número 1) e incrementar el ancho de banda de la luminancia (puntos 3 y 4). Un filtro *Comb*

consiste en un conjunto de líneas de desvanecimiento exactamente al tiempo de la sincronía horizontal para una línea de video y adición/substracción de circuitos, separa la señal de color de la señal de luminancia (Figura 3.18). Dichos filtros pueden ser implementados en circuitos analógicos, y son ampliamente usados en los TV actuales. Sin embargo estos filtros permiten algo de interferencia por cruzamiento de color.

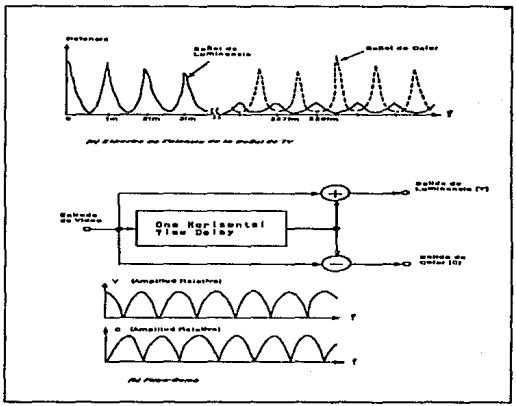


FIGURA 3.18

El proceso de entre tramas -como la detección de movimiento- es efectiva para nuevas innovaciones en la calidad de la señal de video, pero estas técnicas requieren de memorias muy largas para las tramas y un proceso muy complicado. Las tecnologías analógicas no pueden manejar este tipo de procesos, así es que las tecnologías digitales deben de ser utilizadas.

Para reducir la degradación por entrelazado (puntos 2 y

3), es posible convertir el entrelazado, a un barrido progresivo con los receptores de TV usando un proceso tridimensional que es análogo al proceso de separación par Y/C. Mientras la conversión es llevada a cabo en los receptores, no existe la necesidad de modificar las transmisiones estandars de TV.

Nuevos estandars de TV para futuras implementaciones de la calidad de la señal de video son ahora discutidas en la CCITT, entre estas propuestas, se encuentran los sistemas de televisión de definición extendida (EDTV); para estos sistemas, los sistemas de TV convencional son parcialmente modificados y la compatibilidad con los sistemas estandars de TV se mantiene [12,13]. Las modificaciones incluyen adiciones en las componentes de alta frecuencia en porciones no utilizadas del espectro estandar de la señal en frecuencia. Una pantalla más ancha con una relación de 16:9 de largo por ancho -comparada con el rango de 4:3 de los sistemas actuales- esta también en discusión.

Para servicios directos de transmisión en satélite, el sistema MAC ("*Multiplexed Analog Component*") ha sido desarrollado en Europa [14]. En este sistema, los modos de interferencia ya antes mencionados, son reducidos transmitiendo las señales de luminancia y crominancia separadamente en una forma secuencial después del tiempo de compresión. Tales avances solo son posibles a través del procesamiento digital del video.

Los televisores de alta definición (HDTV) serán introducidos hasta la siguiente generación de los estandars de televisión. Este tiene una relación de 16:9 de largo por ancho y más de 1000 línea de barrido en una trama, lo cual es aproximadamente cinco veces más información, que los actuales sistemas de TV. La información contenida dentro de la HDTV es tan larga que no cabe en el ancho de banda de los

canales convencionales de TV. Una solución a esto es la compresión y descompresión del ancho de banda, lo cual requiere de mayor número de operaciones complejas de DSP.

3.14 VLSI PARA SEÑALES DE VIDEO

Como ya se menciono anteriormente, un procesador de una señal de video es un componente para el DSP de video. Para procesamiento de señales de video en tiempo real, las señales deben de ser procesadas durante un periodo extremadamente corto. Aún más, el procesador de video debe de tener una capacidad muy alta de procesamiento. En suma, muchos diferentes y algunas veces, procesos irregulares son necesarios, esto requiere de mayores procesadores flexibles, que son, en general, difíciles de implementar. Para sobre llevar esta difícil situación, un gran número de propuestas han sido hechas; para señales de video en procesadores específicos .

3.15 RENDIMIENTOS REQUERIDOS PARA EL PROCESAMIENTO DE LAS SEÑALES DE VIDEO

El número requerido de multiplicaciones por pel para las funciones básicas utilizadas en el procesamiento de video, van del rango de uno a veinticinco (Tabla 3.9). Asumiendo que esos pels son muestreados a una frecuencia de 14.3-MHz, el número de multiplicaciones por segundo requeridas por un procesador de señal VLSI, es el producto de la frecuencia de muestreo y el número total de las funciones a multiplicar por muestra. Por ejemplo, un DCT (ya antes mencionado en este mismo Capítulo), basado en un codificador de compensación de movimiento de entre tramas compuesto de un detector de movimiento, DCT, DCT inverso, y un (*inner loop filtro*), debe de realizar cerca de mil millones de multiplicaciones por segundo.

TABLA 13.9

MULTIPLICACIONES REQUERIDAS PARA PROCESOS DE VIDEO TIPICOS

FUNCION	MULTIPLICACIONES POR SEGUNDO
Multiplicación Simple	1
L2 Cálculo Normal	1
3 * 3 Filtro Espacial	9
Búsqueda en Arboles de 10 Estados Binarios con Patrones Marcados (Vector Cuantificado con 1024 Vectores)	10
2D Transformada Matricial (DCT) (8 * 8 Kernels)	16
Búsqueda en Arboles de 3 Estados Octales con Patrones Marcados (Vector de Detección de Movimiento)	25

Dos aproximaciones han sido propuestas para los chips VSLI de procesamiento de las señales de video. El primero es realizar una función típica de procesamiento de la señal, así como un filtro de Respuesta de Pulsos Finitos ("Finite

Impulse Response", FIR) o una unidad DCT, con un control lógico amplio en el chip (Tabla 3.10). Esta aproximación ofrece operaciones a una alta velocidad, pero pierde flexibilidad. Implementaciones en esta aproximación incorporadas en el chip de múltiples unidades aritméticas, corresponde a las cajas del diagrama funcional de bloques. Un chip con un filtro FIR, por ejemplo, contiene 64 multiplicadores paralelos para un filtrado de 8 * 8 [17]. El chip opera a 20-MHz para realizar un filtrado dimensión en tiempo real.

TABLA 3.10

CHIPS FUNCIONALES VLSI PARA PROCESAMIENTO DE SEÑALES DE VIDEO

FUNCION	FUENTE	REFERENCIA
Chip FIR con rango de video	INMOS	16
Chip para filtrado de 8 * 8	Lógica LSI	17
Chip DCT de 16 * 16	Bellcore	18
Chip de Separación de Luminancia/Crominancia	NEC	19

La segunda aproximación es implementar un procesador de señal VSLI bajo control de un software. Este software ofrece una gran facilidad y gran versatilidad. Sus desventajas son que disminuye su velocidad. A causa de los requerimientos de alta velocidad para el procesamiento de la señal de video, solo unos pocos chips han sido realizados para este propósito. Para el procesamiento de toda la imagen, un chip de flujo de datos y un chip para imagen han sido diseñados [22,23]. Sin embargo, sus áreas de principal aplicación son limitadas a unidades funcionales del procesamiento de la señal de video, así como en los filtros FIR, con un número pequeños de derivaciones ("*taps*") o 8/16 puntos DCT, los ciclos de procesamiento de alta velocidad en estos chips, permiten un procesamiento directo de un barrido ("*raster*") de la señal de video sin la necesidad de la velocidad de conversión de la entrada muestreada.

Recientemente, chips de procesadores programables bajo software de la señal de video han sido desarrollados como un elemento del proceso en una configuración de multiprocesadores. Ejemplos de esto son los Módulos de Proceso de la Señal de Video (VSPM) y el DSP-1 [24,25,26]. Ambos chips procesadores fueron diseñados para manejar la señal de video en un buffer de video.

3.15.1 UN EJEMPLO DE NEC

Los progresos en las tecnologías VLSI son notables, especialmente en lo concerniente a la densidad de integración con respecto al defasamiento de las compuertas. Recientes circuitos de procesamiento de señales de vídeo, VLSI, emplean procesos paralelos, donde múltiples unidades aritméticas son activadas simultáneamente.

El Procesador de Señales de Video de NEC (VSP) es un sistema de proceso paralelo compuesto de múltiples VSPM [24,25]. La señal de una trama de video es divida en varias subtramas de la misma. Un VSPM es asignado a la señal de video para cada subtrama. La técnica conocida como "*overlap-save*" elimina las discontinuidades en la señal de video después del procesamiento [27]. En este técnica, la subtrama de entrada es más larga que la subtrama de salida. Los pels alrededor de las áreas adyacentes son guardados en diferentes VSPM. A causa de este proceso en paralelo, las aproximaciones son asignadas en el VSPM para procesar la señal de video para una subtrama en vez de a la trama completa, lo que ocasiona que los procesos complejos sean más sencillos. (Figura 3.19)

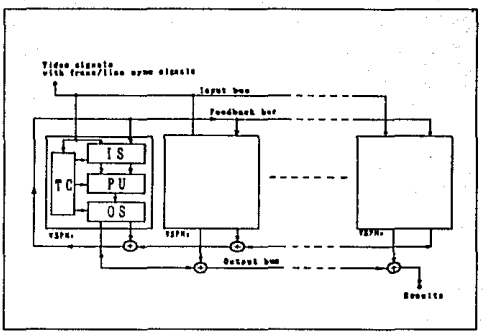

Video signals
with frame/line sync signals
Input bus
Feedback bus
IS
TC
PU
OS
VSPM1
VSPM2
VSPMn
Output bus
Results

FIGURA 3.19

3.16 IMPLEMENTACIONES FUTURAS

El DSP para señales de video ha encontrado un sin número de aplicaciones, y seguirá encontrando e incrementandose el número de áreas de aplicación. Un ejemplo es la proliferación de sistemas de transmisión digital de alta velocidad (la red digital de servicios integrados ISDN o RDSI) y los dispositivos de almacenamiento masivo (memorias de disco ópticos). En otras áreas se incluyen aplicacione a las video grabadoras digitales (VCR), sistemas computacionales de tiempo real, los video juegos, y los sistemas de reconocimiento de imágenes.

Las aplicaciones descritas anteriormente -compresión de video HO-DPCM, conversiones estandars de televisión, y los equipos de efectos de video digital- requieren aproximadamente de entre 100 a 500 MOPS (Millones de Operaciones Por Segundo), mientras más compresión compleja de video se requiera, la velocidad de operación puede ser hasta de 1 GOPS (Giga de Operaciones por Segundo). El reconocimiento de imágenes requiere una velocidad de operación de mayores magnitudes que las de compresión y las gráficas computarizadas de tiempo real requieren aún mayor velocidad de operación (Figura 3.20).

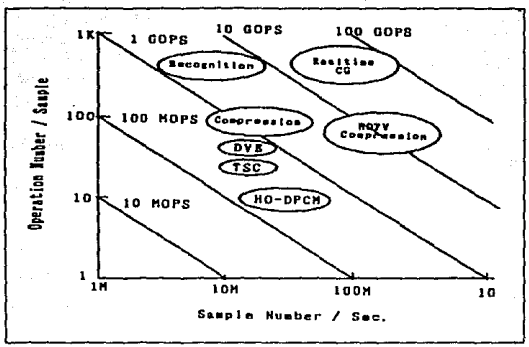


FIGURA 3.20

Para la HDTV, una velocidad de entre 5 y 10 veces mayor será requerida. Operaciones de esta magnitud y complejidad son normalmente realizadas con un gran equipo y a un precio muy elevado. Algunas de estas operaciones aún son imposibles de implementar con equipos de tamaños considerables.

El progreso en las arquitecturas de procesamiento digital de señales y las tecnologías en los dispositivos en semiconductores son la solución a este problema. En adición a las ventajas que traen las arquitecturas de procesamiento paralelo ya descritas, se pueden obtener beneficios de la arquitectura llamada "*pipeline*" o de cadena. Esta arquitectura de multi etapas es utilizada para nuevos incrementos en las velocidades de operación. En dispositivas de memorias, arquitecturas de doble ó triple puerto son ampliamente utilizadas en muchas aplicaciones de video. Esta arquitectura permite conversiones de entrada/salida con mayor facilidad.

Los progresos en los dispositivos continúan en una

rápida carrera (Figura 3.21). El nivel de integración - número de transistores por chip- para las DRAMs se han incrementado en un factor de cuatro cada tres años. Recientemente, sin embargo, progresos en la integración han sido aún más acelerados, por lo menos en dispositivos experimentales. Los progresos en integrar los procesadores digitales de señales han sido también constantes y se espera que continúe en la década de los 90's.

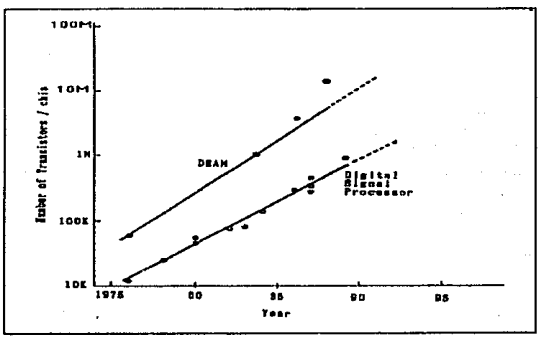


FIGURA 3.21

3.17 COMPRESION DE VIDEO

La codificación digital de vídeo se encuentra ahora al alcance de cualquier persona. Así como los discos compactos ("*compact discs*") han revolucionado la industria del audio, la nueva tecnología promete muchas aplicaciones para el video, en las que se puede contar con herramientas y avances para discos lasers ("*laser discs*"), cámaras electrónicas, video teléfonos, video conferencias, herramientas para vídeo interactivo en computadoras personales y workstations, transmisiones por cable y satélite, y televisores de alta definición (HDTV). A pesar de tener la tecnología digital para audio en los '80s, muchas de las aplicaciones para video se vieron limitadas por falta de compresión de datos. Después de todo, el ancho de banda del audio es de 20 kHz, la cual trasladada a un sistema digital resulta en un rango de 1.4 Megabits por segundo para sonido stereo de alta fidelidad. En cambio para señales de video se requieren rangos más grandes, de alrededor de 10 Mb/s para obtener una calidad comparable a las transmisiones normales de video y más de 100 Mb/s para señales de HDTV.

Aún cuando se manejan imágenes fijas, se necesita gran cantidad de datos para representar y guardar la información de video. Por ejemplo, una imagen de color con una resolución de 1000 por 1000 elementos o pixels a 24-bits cada elemento, ocupara 3 megabytes de espacio si no se le aplica un proceso de compresión, por lo que si se ve desde el punto de vista computacional, esta información no cabría en un solo diskette, los cuales solo pueden almacenar 1.2 o 1.44 Mbytes de información.

Mientras tanto, para facilitar el crecimiento de la industria, se han desarrollado tres estandars: (1) para imágenes en movimiento, (2) para imágenes fijas y (3) para video conferencias. Al mismo tiempo se desarrollaron varios

conjuntos de circuitos integrados para los tres propósitos anteriores.

3.17.1 COMO TRABAJA

Los métodos de compresión se basan tanto en la redundancia de los datos como en la no linealidad de la visión humana. Esta última explota las correlaciones espaciales para imágenes fijas y la correlación espacio/tiempo para imágenes en movimiento. La compresión en espacio es conocida como de "intra tramas" (*intra-frame*), mientras que la compresión en tiempo es llama de "entre tramas" (*inter-frame*). Generalmente, los métodos que tienen un alto rango de compresión para video (10:1 a 50:1 para imágenes fijas y de 50:1 a 200:1 para video en movimiento) pierden mucha información en la reconstrucción de los datos, por lo que los datos de salida distan mucho de los datos de entrada originales.

Existen métodos con menos perdidas, pero sus rangos de compresión son bastante menores, no mayores de 3:1. Por lo que estas técnicas solo son utilizadas en aplicaciones muy específicas, como en imágenes médicas, esto es, si por ejemplo, si ciertos algoritmos, con pérdidas, se llegasen a utilizan en radiografía, pueden causar una incorrecta interpretación y alterar el diagnóstico médico. Por otra parte, en los sectores, comercial e industrial, se prefiere utilizar algoritmos que puedan tener pérdidas, porque se gana espacio y ancho de banda para las comunicaciones.

Estos algoritmos de gran cantidad de pérdidas, explotan normalmente aspectos del sistema visual humano. Esto es, el ojo es mucho más receptivo a detalles finos en la señal de luminancia (o brillo) que en la señal de crominancia (señal de color). Consecuentemente, la señal de luminancia es

usualmente muestreada a una mayor resolución, (por ejemplo, en señales de TV convencional, la resolución digital de una señal de luminancia muestreada, es de 720 por 480 pixels, mientras que la señal de color es solo de 360 por 240 pixels). En segundo lugar, al codificador (o compresor) que representa a la señal de luminancia se le asignan más bits (o un rango dinámico mayor) que a la señal de crominancia.

También, el ojo humano es menos sensitivo a señales con mayor frecuencia que a señales con menor frecuencia espacial. En suma, si una imagen en un monitor de 13 pulgadas de una computadora personal, estuviese formado por señales alternativas de blanco y negro, el ojo humano solo vería una señal uniforme de gris en vez del patrón de señales alternadas. Esta deficiencia es explotada codificando los coeficientes de alta frecuencia con menos bits y las componentes de baja frecuencia con mayor cantidad de bits.

Todas estas técnicas se suman para realizar algoritmos más poderosos con menores pérdidas. En pruebas realizadas, reconstruyendo imágenes codificadas con rangos de compresión de 20:1, es difícil distinguirlas de las originales. Hoy día, datos de video pueden estar en los rangos de compresión de 100:1, y al momento de descomprimirlas, poseen calidad muy cercana al video analógico original.

3.17.2 ESTANDARS

La falta de estandars en este campo trajo consigo un lento crecimiento de la tecnología y de sus aplicaciones. Se han propuesto tres estandars para video digital, el primero es el JPEG (*Joint Photographic Experts Group*) para imágenes fijas, el segundo es la Recomendación H.261 para video conferencias del CCITT (*Comité Consultivo Internacional para*

Telefonía y Telegrafía), y el tercero es para compresión de imágenes en movimiento y almacenamiento medio digital (DSM o *Digital Storage Media*) de MPEG (*Moving Pictures Experts Group*).

La JPEG propuso un algoritmo para la codificación de imágenes fijas, el cual fue desarrollado por un grupo de investigación bajo los auspicios de la ISO (*International Standards Organization*). El grupo llegó a un acuerdo en 1987 y el algoritmo es actualmente la recomendación 10918 de la ISO. Una idea general del algoritmo es la siguiente: comprime en una aproximación una línea base con pérdidas y luego, con otra aproximación con menos pérdidas también lo calcula, siendo las dos funciones independientes y utilizando diferentes técnicas de codificación de una línea base a otra.

El algoritmo JPEG de línea base cae dentro de la codificación de imágenes basadas en transformadas. Una imagen de color puede ser representada en sistemas de diferentes colores. Esto es ampliamente utilizado hoy día, ya sea como, R-G-B (los tres colores primarios, Rojo, Verde y Azul), en la industria de la computación; Y-U-V (Y para luminancia, U y V para la diferencia de las señales de color, y Y-R y Y-B), en la industria de la televisión; y C-M-Y-K (Cyan, Magenta, Yellow, Black) en la industria de la pintura. Donde en cada sistema, las partes de color constitutivas son llamadas componentes. Esto es, tres componentes en sistema R-G-B y cuatro en el sistema C-M-Y-K.

Cada componente de imagen original en el codificador y decodificador JPEG es dividido en bloques que de 8 por 8 pixels, representado en la Figura 3.22:

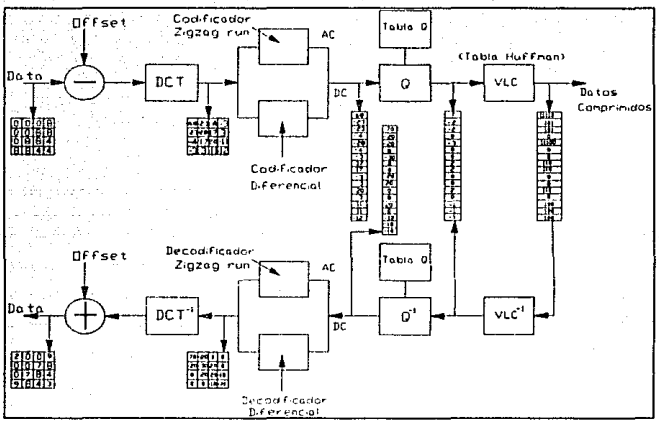


FIGURA 3.22

Cada bloque es transformado utilizando la Transformada Discreta del Coseno (DCT) bidimensional (la cual ya se explicó con anterioridad en este mismo capítulo) con un sistema de bloques de 8 por 8.

Los 64 coeficientes resultantes, calculados como un arreglo en 2-D de 8 por 8, representa la frecuencia contenida en el bloque de entrada. Los coeficientes de la DCT en la parte superior izquierda del arreglo miden la energía de la frecuencia cero o de la señal de corriente directa (DC). (Por ejemplo, si la imagen original de 8 por 8 tiene un valor constante, el único término es el corriente directa). Las otras 63 localidades son los coeficientes de corriente alterna (AC); dando una fuerza relativa en términos de la señal con incrementos en las frecuencias horizontales de la señal, de izquierda a derecha, y en términos del incremento de la frecuencia vertical de arriba hacia abajo.

El siguiente paso, es que se cuantifiquen los coeficientes de la DCT. Los pasos de cuantificación varían dependiendo de la frecuencia y de la cantidad de componentes. La dependencia de la frecuencia se refleja en el hecho que los coeficientes de alta frecuencia son menos importantes que los de baja frecuencia, por lo que son cuantificados con mayores incrementos. En adición, un componente individual puede tener su propia tabla de cuantificación. En el algoritmo realizado por la JPEG, se permiten hasta cuatro tablas de cuantificación.

En cuantificaciones siguientes, los coeficientes son reordenados en un arreglo unidimensional a partir del arreglo bidimensional, siguiendo una ruta en zigzag. De esta manera, los coeficientes cuantificados son "aproximadamente" arreglados en orden ascendente en frecuencia.

Lo siguiente, es que los coeficientes de DC y AC sean codificados, utilizando una codificación de Huffman y con llaves de diferentes parámetros. La codificación de Huffman es bien conocida ya que reduce el número de bits utilizados para representar los datos, sin perder información. Los coeficientes de DC son codificados diferencialmente, así que los coeficientes de DC del bloque anterior de 8 por 8 del mismo componente son utilizados para predecir los coeficientes de DC del siguiente bloque de 8 por 8, y la diferencia entre los dos términos de DC es codificada. La tabla de código Huffman para los términos de DC se base en las diferencias de los valores.

La codificación en zigzag de los coeficientes de AC son codificados primeramente con un código llamado *run-length*. Este proceso reduce cada bloque de 8 por 8, de los coeficientes de la DCT en un número de eventos. Cada evento representa un coeficiente no cero y un número de procedencia de coeficiente cero. Puesto que los coeficientes de

frecuencia alta son más comunmente cero, el código de Huffman hace posible que estos eventos se guarden con una alta eficiencia de compresión.

El algoritmo de base de línea de la JPEG, provee dos tablas, una para los coeficientes de DC y la otra para los de AC. En la decodificación JPEG, el algoritmo de codificación, simplemente se corre en modo inverso, por lo que es comunmente descrito como un algoritmo simétrico.

Para video teléfono la recomendación H.261 de la CCITT, especifica un método de comunicación, el cual es normalmente llamado estandar p*64, porque la velocidad para los datos en el canal de comunicación es de *p* veces 64 kilobits por segundo, donde *p* es un número positivo entero menor o igual a 32. Para p=1, siendo una baja calidad de la señal de video, puede utilizarse en teléfonos y puede ser transmitida a 64-kb/s en línea. Si p=32, es una señal de video de alta calidad y para video conferencias puede ser transmitida a más de 2-Mb/s en línea.

El estandar especifica la organización e interpretación de los bits transmitidos así que dos codificador-decodificador (codecs) de diferentes compañías puedan sostener una sesión de video conferencias. Un codificador de la CCITT es más complicado que un codificador de la JPEG, sin embargo, se pueden distinguir bloques con funciones similares, así como el DCT y el cuantificador. El decodificador de la CCITT, es, sin embargo, menos complejo que su codificador (Figuras 3.24 y 3.25).

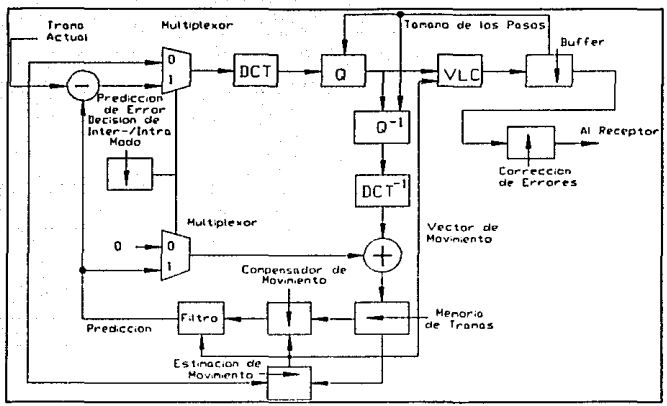


FIGURA 3.24

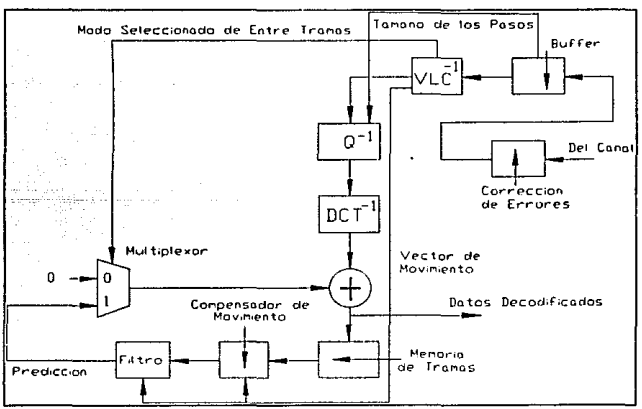


FIGURA 3.25

El codificador CCITT es híbrido, porque combina codificación por transformadas -basado en la DCT- con codificación de predicción, en donde un bloque en la trama actual es pronosticado de la trama anterior utilizando una retroalimentación. (En contraste, el algoritmo de la JPEG opera básicamente en lazo abierto y es restaurado al final de cada imagen). Es esta predicción de entre trama la que da por resultado un rango de compresión mayor.

También, en vez de utilizar solamente un código de predicción basado en las diferencias entre la actual trama y la imagen reconstruida en la memoria de trama, el Estandar H.261 de la CCITT tiene una especificación opcional para compensación de movimiento. Esto incrementa la eficiencia en el código de predicción.

3.17.2.1 ESTANDAR PARA IMAGENES EN MOVIMIENTO

Como el Estandar H.261 para video conferencias, el Estandar MPEG es un algoritmo para compresión de imágenes en movimiento con modos de intra y entre tramas. A comparación del H.261, la velocidad de los datos no debe de exceder 1.5 Mb/s (aunque están en desarrollo trabajos para incrementar la velocidad de transferencia de datos). Hoy día algunos ejemplos, los cuales utilizan partes de este algoritmo son los compact discs, DAT (Digital Audio Tape) y los Discos Duros de Computadora.

Los Estandars par el MPEG no han terminado su trabajo y las especificaciones para el audio permanecen en espera. Para una visión general, la funcionalidad del diagrama en bloques del codificador del H.261 es aplicable, sin embargo, las especificaciones de cuantificación y de estimación de movimiento/compensación son diferentes.

Recientemente una segunda etapa para crear un estandar del MPEG-2 se ha puesto en marcha, el cual propone una eficiente codificación arriba de los 10Mb/s con elevados resultados de calidad de imagen.

3.17.3 SOLUCIONES EN SILICIO

Muchas aplicaciones de video digital requieren implementaciones de bajo costo en silicio, para que puedan ser liberadas al mercado. Al final de 1990, se anunciaron muchos circuitos integrados los cuales comprimían imágenes y video, muchos de los cuales no seguían los estandars, y solo unos pocos seguían los Estandars JPEG y H.261.

Algunas soluciones para el JPEG fueron ofrecidas primeramente por C-Cube Microsystems Inc., en San José, Calif.; después por LSI Logic Corp., en Milpitas, Calif. Los cuales se basaban en un solo circuito integrado, el CL550, el cual seguía una versión primaria del JPEG. Estas compañías actualmente trabajan en diseñar un sistema que siga las nuevas especificaciones del JPEG. Por otra parte, la compañía LSI Logic´s también combinó dos circuitos integrados, el L64735 que es un procesador DCT, y el L64745 que es un cuantificador JPEG, esta implementación redujo los costos.

Las arquitecturas de C-Cube y LSI Logic son similares, ambas implementaciones, las dos con el CL550 y la versión de dos circuitos integrados de LSI, implementan el algoritmo JPEG, pero varían en la resolución de la imagen, el número de componentes, los niveles de cuantificación y las tablas de Huffman utilizadas.

Los productos para el H.261 fueron creados por LSI Logic y por Graphics Communications America Ltd. (CGA). LSI

anunció en Septiembre de 1990 la construcción utilizando un sistema de siete circuitos integrados en bloques, para codificar/decodificar video en movimiento, CGA anunció lo mismo solo que basado en un sistema de doce circuitos integrados.

La línea de LSI Logic consiste en cuatro procesadores y tres codificadores/decodificadores. Los procesadores son el L64720 para estimación de movimiento, el L64720 para el cálculo del DCT, el L64740 para cuantificación y el L64760 para decisión de intra/inter trama. Los otros tres circuitos integrados son el L64715, para corrección de errores, el L64750, para la codificación H.261, y el L64751, para decodificación H.261.

4. CAPITULO III "SEGURIDAD EN COMUNICACIONES"

Con excepción de una larga lista de aplicaciones militares, la seguridad en las comunicaciones es un evento relativamente reciente. En comunicaciones analógicas no militares, muchas personas han expresado estar molestas con ayanamientos en las comunicaciones por cable (por ejemplo, el llamado "*tapping*" o cruzamiento de los teléfonos), las intercepciones de las transmisiones por ondas (por ejemplo, la recepción de una señal de microondas en una trasmisión telefónica o el recibir una señal codificada de un satélite privado).

La llegada de las comunicaciones digitales viene ofreciendo seguridad, que primeramente era para uso exclusivamente militar, ahora se encuentra en el mercado de consumo. Las empresas están substituyendo por transmisiones electrónicas de datos lo que antes se mandaba por servicio postal o servicio de mensajería. El movimiento de datos es importante en llevar a cabo transacciones financieras, el de transferir el dinero electronicamente y el de computar el pago de los cheques. La seguridad es importante no solo para prevenir a personal no autorizado de obtener la señal, sino también para prevenir una alteración de los datos.

Existen varios géneros para garantizar la seguridad de un mensaje. El método más antiguo es el de confinar la señal a una transmisión por cable y limitar el acceso físico al canal. Un segundo método es el de enmascarar la señal, de tal forma, que personal no autorizado, no sea capaz de diferenciar la señal del ruido de fondo. El tercer método es el de encubrir o enmascarar los datos reales usando técnicas de codificación, por lo que el mensaje será incomprensible al personal no autorizado, aún cuando reciban correctamente la señal.

4.1 CRIPTOLOGIA

El reto de la criptología es el de cambiar un mensaje de forma tal, que solo el receptor interesado puede comprenderlo. Esto debe ser realizado en forma económica tanto para el transmisor como para el receptor. Al mismo tiempo debe de resultar muy difícil (y caro en tiempo y/o equipo) para personal no autorizado, el recibir y comprender el mensaje.

Datos muy sensible en el tiemp son ahora, distribuidos exclusivamente por transmisiones electrónicas, mientras en el pasado, el servicio postal y el servicio de entrega inmediata eran utilizados. Esto hacía que la privacia se pusiera en una crisis considerable. Los sistemas tienen que volverse más y más complejos a medida que el personal no autorizado tenga acceso a equipo menos costoso y más sofisticado. Estamos envueltos en una guerra tecnológica, altamente sofisticada, en la cual no se ve un fin próximo. Para cada avance en el de proveer seguridad a la transmisión, existen avances comparables en el arte de "*romper los códigos*". Se estan creando nuevos problemos de seguridad con las nuevas tecnologías, por ejemplo, se requiere de cierta experiencia y pericia para poder captar las transmisiones por cable, sin embargo, no se requiere de una conexión sofistada para captar una señal de satélite, uno simplemente necesita localizar la transmisión con un receptor sin necesidad de dejar huella alguna. Esto, sumado con otras consideraciones, dan por resultado que en las transmisiones se utilicen sistemas criptológicos más complejos.

Poco después de la Segunda Guerra Mundial, el interés y la intriga del público se volcó sobre algunas formas elementales de criptología dejadas para juegos de niños, los cuales podían unirse a un club, en donde mensajes

codificados eran enviados entres los miembros del club. Estos mensajes utilizaban una forma muy sencilla de encriptación, donde cada letra de una palabra era intercambiada por otra de acuerdo a una llave secreta, por ejemplo, cada vez que la letra A era encontrada, era escrita como una letra M. Esta correspondencia de uno a uno o código de *substitución* es aún vista en libros de adivinanzas, donde una operación matemática es remplazada por letras y el jugador debe decodificar el mensaje asosiando un número entero para cada letra. Una desventaja para este tipo de código es que el tamaño del segmento de entrada es siempre el tamaño del segmento de salida, por lo que es relativamente fácil decodificar el mensaje utilizando aproximaciones relativas de frecuencia, por ejemplo, la letra más comunmente utilizada es la "E". Ocurrencias regulares de palabras de tres letras son: "LOS", "LAS", POR", y así sucesivamente.

Algunos códigos de substitución son aleatorios o randoms, esto es, el encriptador decide la substitución para cada símbolo de manera aleatoria. En tales casos, el receptor interesado necesita la tabla entera de equivalencias para descifrar el código. Otros códigos son más sistemáticos, por ejemplo, se le puede sumar 2 a cada letra del alfabeto, así es que la A se convierte en C, la B en D, y así sucesivamente, en este caso, el receptor solo necesita la regla de substitución.

Una segunda técnica elemental de codificación es la *transposición*, donde el orden de los símbolos es permutada. Como un ejemplo, examinemos la siguiente frase:

COMO UN SIMPLE EJEMPLO

Si se permuta por parejas, el resultado es:

OCOM NU ISPMEL JEMELPO

Generalmente permutaciones más largas son utilizadas. La técnica de transposición puede ser combinada con la técnica de substitución mencionada anteriormente. Los criptologístas experimentados tienen pocos problemas con estos códigos, en especial si utilizan una computadora como ayuda. En la mayoría de los casos, para descifrar un código, se empieza por realizar un análisis de frecuencia de los símbolos.

Mayores niveles de encriptación utilizan varias combinaciones de permutaciones en un patrón bien definido. Por ejemplo, si se permuta el mismo mensaje anterior, alternando parejas y trios, se obtiene:

OCUM OS NPIMEL EEJPMOL

De la misma manera, se pueden alternar diferentes algorítmos de substitución. Por ejemplo, para los primeros 100 símbolos se puede utilizar un alfabeto con substitución donde la letra A sea la C, y para los siguientes 100 símbolos se puede utilizar otro diferente donde por ejemplo la A se convierta en Z. Todos estos problemas son grandes obstáculos para un jugador amateur, pero un verdadero espia puede romper todas estas técnicas combinadas, especialmente cuando se tiene acceso a una computadora.

4.2 "SPREAD SPECTRUM"

Una meta de los sistemas de comunicación es el de transmitir la máxima cantidad de información con la menor probabilidad de error en un canal con un ancho de banda mínimo y utilizando la menor señal de potencia. Este es un sistema muy ambicioso, y de hecho, no es posible satisfacer todos estos requerimientos simultáneamente.

La técnica llamada "*Spread Spectrum*" utiliza el mínimo ancho de banda, esto es, intencionalmente utiliza un ancho de banda por lo menos 10 veces el mínimo requerido para mandar la señal de información. Si esto es realizado apropiadamente, la señal transmitida se ve, para una persona no autorizada, como ruido de bandas anchas. Como una ventaja adicional, el ancho puede reducir la probabilidad de error, como por ejemplo, en bandas anchas de FM su relación señal a ruido es mayor que en bandas angostas, por lo que probabilidad de error es mayor. Sin embargo, se debe entender que una señal de FM de banda ancha no es considerada como una señal de *spread spectrum*, para ser considerada una señal de *spread spectrum*, el ancho de la frecuencia debe depender de la señal de banda base.

Supongamos que se empieza con una señal binaria con ruido de banda ancha. Esta señal varía aleatoriamente entre 1 y 0 y tiene un espectro de frecuencia cuyo ancho es proporcional a la velocidad con que se envia. Si se toma una señal que contenga información de banda angosta, y se modula con un ruido, como se muestra en la Figura 4.1:

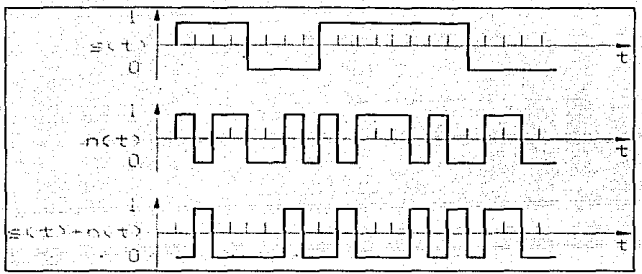


FIGURA 4.1

la acción binaria en mod. 2 es utilizada para la modulación, así es que la suma 1 + 1 = 0 (siendo una operación lógica con OR-Exclusivos). Esto es, cuando el ruido binario de alta frecuencia es 1, el resultado de la modulación es un bit inverso al de la señal de información original. Cuando la señal de ruido es 0, no ocurre inversión. La señal modulada es de banda ancha con un ancho de banda aproximadamente igual al del ruido. Realmente, este ancho de banda es la suma del ancho de banda de la señal de ruido y del ancho de banda de la señal de información.

El efecto de la modulación es el de invertir bloques de la banda ancho del ruido, por ejemplo, si la banda ancha de ruido tiene una velocidad de 1 Mbps (Mega bit por segundo), y la señal tiene una velocidad de 1 Kbps, se estará dividiendo la señal de ruido en bloques de 1,000 bits. Si el bit de información es 0, el bloque de ruido es transmitido sin modificaciones. Si el bit de información es 1, cada bit en el bloque de ruido asociado es invertido. Puesto que la frecuencia del ruido es mucho mayor que la frecuencia de la información, la secuencia de bits resultantes parecerá ser aleatoria.

La señal original puede ser recobrada volviendo a modular con la misma señal de ruido (OR-Exclusivos). Esto es, puesto que la adición binaria de información con los efectos de inversión por la señal de ruido se repetirán, una segunda modulación traerá consigo la señal original.

Se puede notar que la señal de ruido es random. ¿Cómo es posible entonces, repetir la operación en el receptor? Supongamos que en vez de ser una señal aleatoria, el ruido sea pseudoaleatorio o PN (PN por *Pseudorandom Noise*). La secuencia PN, es un ciclo generado utilizando una secuencia de inicialización (en 1) y secuencia en silencio (en 0), también, la secuencia posee muchas de las propiedades del ruido de banda ancha. Cuando el código PN es utilizado para modulación, el proceso es conocido como *secuencia directa PN*.

Cualquier forma de modulación digital puede ser ensanchada, pero normalmente se escoge PKS, debido a la amplitud constante de este tipo de modulación. Es generalmente más difícil para receptores no autorizados detectar la señal de información.

Se define el proceso de ganancia del *spread spectrum*, Gp, como la relación señal a ruido de la salida del demodulador contra la relación señal a ruido de la entrada del demodulador.

Una explicación para esta ganancia es que el receptor multiplica la señal recibida reconstruyendo la secuencia PN. Esta secuencia, el espectro de ruido recibido, será reconstruida en la señal original. La señal a ser reconstruida es pasada a través de un filtro paso bajo.

4.3 CODIGO MULTIPLE POR DIVISION DE TIEMPO (CDMA)

El concepto de secuencia directa *spread spectrum* puede ser extendida para proveer una técnica de multiplexaje, muy diferente a la de frecuencia o a la división de tiempo multiplexado. Esta técnica es conocida como Acceso Múltiple por División de Tiempo (*CDMA*).

Se empieza con un sistema de dos canales como se muestra en la primera parte de la Figura 4.2. Las señales de ruido $n_1(t)$ y $n_2(t)$, representan dos diferentes secuencias de ruido PN, al mismo rango de señal. Se asume que este rango es mucho mayor que la señal $s_1(t)$ y $s_2(t)$. Tambiém se asume que fue realizada por secuencia-directa para *spread spectrum*. Ya que las dos señales moduladas son sumadas simultáneamente en la transmisión a través del canal; se espera que el esquema para separarlas en el receptor sea sencillo. Las dos señales moduladas, $s_1(t)n_1(t)$ y $s_2(t)n_2(t)$, se traslapan tanto en tiempo como en frecuencia y las técnicas convencionales de tiempo y frecuencia para compuertas no pueden ser utilizadas. Pero se puede tomar ventaja en que la secuencia de las dos señales ruidosa es aproximadamente no correlacionada una con la otra.

El resultado de demultiplexar se encuentra ilustrado en la Figura 4.2:

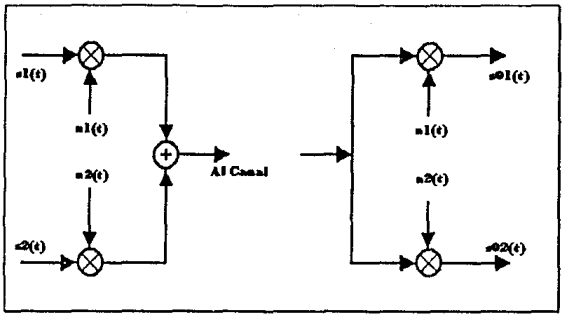


FIGURA 4.2

La señal en la salida del primer multiplexaje esta dada por:

$$s_{01}(t) = s_1(t)n^2_1(t) + s_2(t)n_1(t)n_2(t)$$

Ya que $n^2 1(t)$ es unitario, la función puede reducirse a:

$$s_1(t) + s_2(t)n_1(t)n_2(t)$$

Debido a que las dos funciones de ruido se encuentran esencialmente no correlacionadas, su producto es una señal de ruido a la frecuencia de cada una, respecto a la señal de ruido original. Aún más, el segundo término de la ecuación representa una señal de *spread spectrum* con un amplio ancho de banda, donde el primer término tiene un ancho de banda mucho menor. Un filtro paso bajas puede ser utilizado para reducir el ruido.

El sistema CDMA puede ser extendido a más de dos señales, pero el peligro de cruzamiento se incrementa cuando se aumentan el número de señales.

4.4 MANEJO POR LLAVES

Es obvio que la distribución de llaves es un problema considerable en los sistemas encriptados. Este tipo de sistemas puede ser comparado con una llave de combinación. Muchas personas entienden el algoritmo, pero si la llave no es conocida, de muy poco puede servir entenderlo. Si alguna persona no autorizada se aprende la llave, la seguridad del código es destruida.

Las llaves deben ser distribuidas por cualquier sistema seguro. Normalmente los medios para distribuir las llaves son lentos, por lo que las llaves son distribuidas mucho tiempo antes de que el mensaje sea transmitido.

El sistema por llaves puede ser diseñado para que opere con múltiples llaves. Por ejemplo, supóngase que la llave sea de 64 bits de longitud y es calculada de una adición en mod. 2 (OR- Exclusivos) de dos secuencias de 64 bits, cada una de las secuencias puede ser enviada por diferentes vías, y una persona debe de recibir las dos secuencias para poder realizar la adición y de esa manera calcular la llave correctamente.

4.5 ORIGINALIDAD Y AUTENTICIDAD

Es de gran importancia tanto la privacía como la autenticidad. Pero ¿Cómo podemos asegurar que el receptor conoce quién esta mandando el mensaje? Si realmente la comunicación digital es tan segura que evita las copias, también se necesita un equivalente digital de la firma para evitar falsificaciones, esta firma debe de ser única y segura, debe de ser lo suficientemente protegida para eliminar la posibilidad de errores o de intromisiones.

Un simple ejemplo de esta técnica de autenticidad y originalidad es cuando, se transfieren fondos de una computadora de cajero automático a las manos de la persona. La autenticidad es en este caso es, cuando se le asigna a cada persona un código secreto. La persona debe de introducir el código correcto a la máquina si desea obtener dinero. Para seguridad adicional, la clave en si misma, no es suficiente, por lo que tambieén es requerida la tarjeta, por lo que este tipo de seguridad requiere de dos identificaciones, una de ellas, muy difícil de duplicar (la tarjeta magnética).

Una segundo técnica para autenticidad es utilizada en la distribución de tiempos en computadora, donde a las personas con cuentas activas se les asigna una llave o password, que es requerido para obtener acceso a la computadora. La seguridad en la terminal, de la persona que solicito la entrada, es asegurada, mientras no aparezca el password en pantalla o si se escriben símbolos diferentes sobre la palabra, para que de esta forma otras personas no puedan distinguir la llave correcta.

4.6 DATA SCRAMBLING -REVOLVIENDO DATOS-

El término "*scrambling*" o revolviendo es aplicado a todos aquellos sistemas que reordenan o permutan la secuencia de sus datos. Esta operación normalmente procede a la encriptación, ya que un tren de datos aleatorios poseen algunas propiedades deseables. Si el tren de datos originales posee cadenas muy largas de 1's o 0's, se puede llegar a perder la sincronía. Otro tipo de sistemas, particularmente los sistemas bifásicos, pueden dar condiciones falsas para ciertas secuencias repetidas de datos. Dando aleatoriedad a los datos se reducen estos problemas.

Una simple forma de aleatorizar los datos, consiste en permutar los datos de entrada. La Figura 4.3 ilustra un sistema muy simple que permuta los datos en bloques de 8 bits.

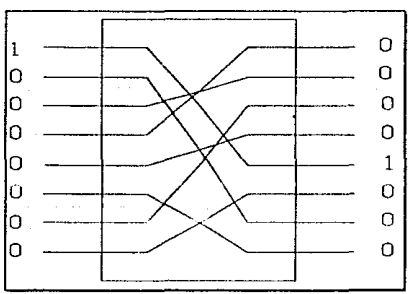


FIGURA 4.3

Este sistema es sumamente sencillo y como es esperado, tiene un sin número de inconvenientes. Uno de ellos, es que, puede ser descubierta la permutación, si en la secuencia de

los datos de entrada solo existe un 1. Esto rápidamente revela las interconexiones para el personal no autorizado.

Algunos sistema más complejos utilizan cambios o corrimientos en los registros para realizar operaciones de convolución de los datos de entrada. La operación matemática realizada por el sistema es la de dividir la secuencia de la información de entrada por un generador de polinomios. Los bits en la secuencia de salida son los coeficientes de los cocientes de esta división.

Existe ventajas en estandarizar este tipo de algoritmos. La CCITT (Comisión Consultiva Internacional de Telefonía y Telegrafía) ha recomendado estandars, uno de ellos es la recomendación V.27, la cual se ilustra en la Figura 4.4:

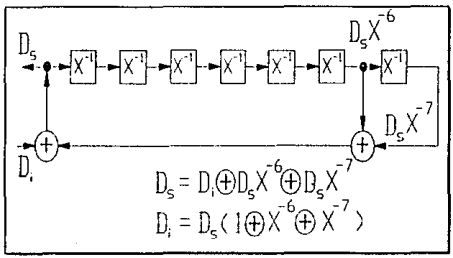


FIGURA 4.4

El generador de polinomios para este sistema es calculado analizando los tiempos de retraso en cada bloque (cada bloque X^{-1} representa un retraso de un bit de periodo), y esta dado por:

$$1 + X^{-6} + X^{-7}$$

El resultado es que los datos transmitidos son

aleatorizados de entre una secuencia de $2^7 - 1$, o 127 bits. Todos los sumadores de la figura son sumadores en mod. 2.

4.6 CODIGOS DE BLOQUES CONTRA TRAMAS

Las técnicas de encriptación puede ser catalogadas en dos áreas: encriptación por bloques y encriptación por tramas.

En la encriptación por bloques, el texto limpio (*plain text*) es dividido en dos bloques de dimensiones específicas. Cada uno de los bloques es encriptado separadamente, esto es, si un bloque es de N bits de longitud, existen 2^N posibles formas de encriptarlo. Por supuesto que el algoritmo que mapea un bloque de texto limpio a un bloque de texto codificado (*Cipher Text*) debe de ser único. De la misma forma, también debe de ser capaz de regresar un texto encriptado a su forma original. Normalmente la longitud del bloque de texto encriptado es de la misma dimensión que la del texto limpio.

Los bloques codificados pueden ser visto como códigos digitales de substitución, en donde existe un diccionario de 2^N palabras y cada uno puede ser reemplazado al final por un bloque de texto limpio.

La codificación por tramas es generada bit a bit a medida que el texto limpio es generado. Esto va realizando una adición al tren de datos con una trama llave, la cual puede ser independiente de la trama de texto limpio o puede ser calculada a partir del mismo.

4.7 SISTEMAS DE LLAVES PUBLICOS

La transmisión de una llave de un receptor a otro es un término absurdo si uno desea sistemas de comunicación seguros. No sería maravilloso que hubiese una forma de marcar el teléfono de alguien, que estuviese conectado al sistema telefónico normal, y sin realizar ningún arreglo previo con el receptor, ¿Mandar un mensaje con un sistema seguro? El sistema de código de llaveS públicas provee esta capacidad para el acceso aleatorio en comunicaciones.

La diferencia entre un sistema público y un sistema no-público, es que en el sistema público, la llave original esta reemplaza por dos llaves, una utilizada por el transmisor y la otra por el receptor. Asumiendo que K_1 sea la llave utilizada para la encriptación, en función de desencriptar, una llave, K_2, es necesaria. Además las dos llaves deben de estar relacionadas, es crítico que un receptor no autorizado que obtenga una de las llaves no obtenga de un método práctico la segunda. Esto es, la llave podría estar incluida dentro de un directorio telefónico. Si una fuente desea transmitir a un receptor, debe de buscar la llave para ese receptor dentro del directorio y escoger una llave complementaria para la transmisión, utilizando alguna regla única para esa fuente. Una persona no autorizado podría conocer la regla para el transmisor, pero sin saber cual es el del receptor, no le sera fácil decodificar el mensaje.

Si cada persona genera un par de llaves que sean complementarias, esto es, una de las llaves es utilizada para encriptar y la otra para desencriptar, siendo la llave para encriptar conocida, no sera fácil desencriptar el mensaje si no se conoce la otra llave. Ahora, si la llave para codificar es una llave que se encuentra en un directorio y la llave para desencriptar es una llave

secreta, cualquier persona que desee mandarle un mensaje utilizara la llave del directorio, por lo que la llave para encriptar es conocida como llave pública y la llave para desencriptar es conocida como llave privada.

Cualquier número en cualquier base puede ser utilizada como llave. Por ejemplo, si se escoge el número 121 de base 10, y se utiliza un código binario, se puede reescribir el número como 1111001, el mensaje se puede dividir en grupos de 7-bits y añadir la llave a cada grupo. De la misma manera, se puede reemplazar esta llave de número escalar con una llave vectorial. La primera parte del mensaje será añadida al primer elemento del vector, el segundo al segundo elemento y así sucesivamente.

Como ejemplo, supóngase que un número es generado de dos variables aleatorias, $\mathbf{x}_1$ y $\mathbf{x}_2$, y por una constante, a, por lo que:

$$K = a^{x1x2}$$

Si **A** conoce el número aleatorio $\mathbf{x}_1$ y **B** conoce el número $\mathbf{x}_2$, y ambos de esto números son necesarios para calcular la llave; si ambos números aparecieran en un directorio, cualquier persona no autorizada podría calcular la llave. En vez de esto, si a^{x1} es asociado a la persona **A** y a^{x2} es asociado a la persona **B**, si **A** quiere mandar un mensaje a **B**, el código de **B** (a^{x2}) es buscado en un directorio, entonces **A** lo eleva a una potencia $\mathbf{x}_1$, por lo que el número secreto solo es conocido por **A**, esto forma al final la llave a^{x1x2}. Una persona no autorizada podría tener acceso a a^{x1} y a^{x2} del directorio, pero para formar la llave se requerirá de realizar la siguiente operación:

$$K = a^{\mathbf{x}_1 \log_a(a^{\mathbf{x}2})}$$

Esto involucra cálculos con logaritmos, los cuales no son difíciles. Sin embargo, si todas las operaciones se realizan en mod. q, donde q es un número primo, los logaritmos resultan entonces más difíciles de realizar.

Funciones de este tipo son conocidas como funciones de una dirección (*one-way functions*). Un simple ejemplo de esto es el siguiente polinomio:

$$y = \sum_{n} a_n x^n$$

Conociendo **x**, es fácil encontrar **y**, pero conociendo **y**, de ninguna manera es fácil encontrar **x**.

4.8 METODO DE HAMMING

Desde los primeros días de la computación y la comunicación de datos, los diseñadores han tenido que luchar contra los errores en las transmisiones, por ejemplo, los datos transmitidos es un canal con ruido, como una línea telefónica, en donde algunos bits son cortados por fenómenos externos. En adición, los datos almacenados en cierto tipo de memoria de computadoras ocasionalmente pierden el valor de un bit. En otros casos, si los datos, no son de alguna forma, transmitidos correctamente, son captados mal. Si un error no es detectado, el resultado puede ser catastrófico - simplemente hay que imaginar que los datos representan alguna transacción de algún banco o de algún cajero automático-.

Por la necesidad de tener confianza en el almacenamiento y en la calidad de la información, se han ido desarrollando, a través de los años, varios métodos para solucionar este problema. Introduciendo una cierta cantidad de redundancia en los datos, un error puede ser detectado en el receptor. Esto es, supóngase que se desea transmitir datos en bytes de 8-bits. Transmitiendo cada byte dos veces, el receptor puede comparar los correspondientes bits y los que discrepen serán erróneos. Por ejemplo, comparando los dos bytes transmitidos:

```
11010011
11011011
```

Estos dos bytes difieren en el quinto bit de izquierda a derecha, por lo que indican un error. De esta forma el receptor no tiene idea de cual de los dos valores es el correcto (0 o 1), por lo que tiene que pedir una retransmisión. Utilizando triple redundancia, la

probabilidad de error se puede eliminar como se muestra a continuación:

```
11010011
11010011
11011011
```

Bajo este esquema, cada byte es transmitido tres veces, y volvemos a notar que existe una discrepancia en el quinto bit. Este bit aparece como 0 en dos de los bytes transmitidos, por lo que se puede asumir con un alto grado de confianza, de que el bit correcto es 0, y una retransmisión no será necesaria. Esta capacidad de autocorrección es enormemente utilizada en muchas situaciones, así como en comunicaciones vía satélite, en donde la propagación de la señal se lleva un tiempo prolongado.

El problema con lo anterior, es que es ineficiente. Con triple redundancia, solo el 33 por ciento de la capacidad del canal es utilizada para los datos actuales. La utilización del código de Hamming es un método ampliamente utilizado para obtener el mismo o aún, hasta mejores resultados. Un subset o subconjunto de bits en cada palabra es asignada a traslapar grupos, y un "*bit de paridad*" (*check bit*) es asignado a cada grupo. Esto permite la detección/corrección de errores en el código recibido con mucha mayor eficiencia del canal.

4.8.1 TEORIA DEL CODIGO DE HAMMING

Para ilustra este procedimiento, supongamos que se desea transmitir un mensaje de 4-bits a través de un canal ruidoso, también se desea poder ser capaz de detectar y corregir cualquier bit erróneo durante la transmisión. Para esto será necesario crear tres grupos de chequeo, y asignar

tres de los bits del mensaje a cada grupo. Nombrando al mensaje y a los bits de chequeo de la siguiente manera:

```
P1 P2 M3 P4 M5 M6 M7
```

Donde P1, P2 y P4 son los bits de chequeo o de paridad. M3, M5, M6 y M7 son los bits del mensaje. Antes de transmitir la palabra código de 7-bits, se les debe de asignar algún valor a los bits de chequeo. Esto es realizado de la siguiente manera: P1 es asignado al chequeo impar de M3, M5 y M7. P2 es asignado al chequeo impar de M3, M6 y M7. Finalmente, P4 es asignado al chequeo impar de M5, M6 y M7. Por ejemplo si tomamos la palabra código:

```
M3 M5 M6 M7
1  0  1  1
```

Debido a que el OR-Exclusivo (XOR) entre M3, M5 y M7 es 0, P1 será 1. El grupo de 4-bits, incluyendo el mismo bit de chequeo, tiene que tener chequeo impar. Por la misma razón, P2 es 0 y P4 es 1. La palabra código completa a ser transmitida será entonces:

```
P1 P2 M3 P4 M5 M6 M7
1  0  1  1  0  1  1
```

Hay que notar que el bit de chequeo P1 también representa la paridad de todo el mensaje para posiciones de 2^1 bit presente, así como en las posiciones 3, 5 y 7 (011, 101, 111 en binario). P2 es la paridad para las posiciones de 2^2 bits, y P4 es la paridad para las posiciones de 2^4 bits. La ubicación de los bits de chequeo es para reforzar la estructura conceptual. En realidad, cualquier orden de bits es aceptada, siempre y cuando el transmisor y el receptor tengan el mismo orden.

Como se puede observar, los bits adicionales de chequeo proveen la suficiente redundancia para detectar y corregir cualquier bit erróneo. Cuando el receptor tiene todos, los 7 bits de la palabra código, la agrupación de bits se repite, y la paridad de cada grupo de 4-bits es revisada. Para P1, los bits de mensaje M3, M5 y M7 son revizados con OR-Exclusivos (XOR); el mismo proceso ocurre para los grupos de P2 y P4.

Si uno o más de los tres grupos no poseen chequeo impar, ha ocurrido un error en la transmisión. Suponiendo que la palabra código recibido sea 1010011, la cual difiere de la palabra código transmitido, los bits del grupo P1 son 1101, del grupo P2 son 0111, y del grupo P4 son 0011. El OR-Exclusivo de los 4 bits en cada grupo es 0, 0 y 1 respectivamente. Como uno de los grupos, P4, no tiene chequeo impar, se ha detectado un error.

También se posee toda la información necesaria para localizar la posición del bit erróneo y corregirlo. Los tres bits de paridad calculados por el receptor están ordenados en orden descendente, de izquierda a derecha, por un número de grupo, P4, luego P2 y finalmente P1. Arreglados de esta manera, los bits son también llamados "*syndrome*". En el ejemplo anterior, el *syndrome* es 100 binario, el cual representa el número cuatro (4) decimal. Es frecuente que el cuarto bit (contando desde izquierda a derecha), sea el bit erróneo, simplemente hay que invertir, para corregir ese bit, el cuarto bit normalmente es un bit de chequeo, por lo que se puede observar que el método de Hamming trabaja para cualquier bit en una palabra código, no importando que sea del mensaje o de chequeo.

El método de Hamming puede ser adaptado para trabajar para palabras de mensaje más largas. La eficiencia se obtiene en términos de los bits de chequeo contra los bits

del mensaje, pero la probabilidad de ocurrencia de error en una palabra código se incrementa. Por el otro lado, si se utilizan un mayor número de bits de chequeo, un número mayor de bits erróneos en cada palabra pueden ser detectados y corregidos.

4.8.2 CODIFICACION Y DECODIFICACION

Los métodos utilizados en diseños lógicos para incrementar los bits de chequeo a un mensaje, incluyen circuitos generados de paridad (Figura 4.5) y tablas de direcciones (como en los ROM). En el primer caso, las compuertas OR-Exclusivas son utilizadas para generar el chequeo impar de un conjunto de 2-bits, la salidas es de nuevo parada por otro OR-Exclusivo para calcular la paridad de un grupo de 4-bits. Los bits de paridad o chequeo son así transmitidos con el mensaje.

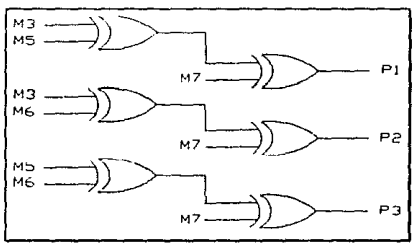


FIGURA 4.5

Para un generador de chequeo de bits de revisión por tablas, los bits del mensaje son presentados como direcciones de memoria. Los datos en cada localidad de memoria son la combinación apropiada de los bits de chequeo para una palabra del mensaje. De nuevo, si se supone un

mensaje de 4-bits, y tres bits de chequeo para corrección de errores, la tabla de memoria requiere cuatro bits de dirección de memoria y 3-bits de palabra, o 16 palabras por 3-bits de memoria. Así como los bits de chequeo producidos por los OR-Exclusivos de una red, los bits de chequeo producidos por la tabla son transmitidos con los bits del mensaje.

El método tradicional para decodificar los datos recibidos es con circuitos OR-Exclusivos, para 4-bits de mensaje y 3-bits de chequeo, como se muestra en la Figura 4.6:

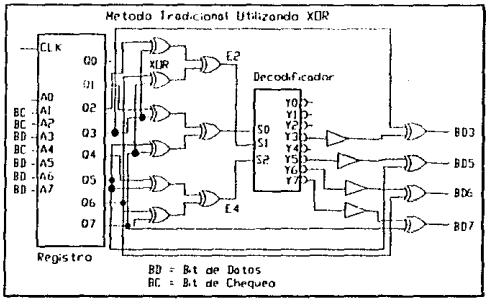


FIGURA 4.6

Los 7 bits de la palabra código son presentados en paralelo a los OR-Exclusivos. Estos son divididos en tres grupos, que corresponden a los tres bits de chequeo. El chequeo impar de cada uno de los 4 bits en el grupo es calculado, y el *syndrome* resultante es pasado al decodificador. El decodificador puede ser considerado como el detector de errores. Si un error es localizado por el *syndrome*, una de las salidas, de Y1 a Y7, será correcta. Si el *syndrome* es 0, Y0, será correcta, por lo que no existirá ningún error.

Las salidas del decodificador son utilizadas para generar la lógica de corrección, la cual consiste en una compuerta OR-Exclusiva para cada bit del mensaje. El bit de chequeo no requiere de ser corregido, mientras se utilice internamente en el receptor, esta es la razón por lo que no existe conexión a las salidas Y1, Y2 y Y4 del decodificador.

Normalmente, las compuertas OR-Exclusivas pasan el mensaje sin cambiarlo. En algunos casos, el decodificador toma una de las salidas, Y3, Y5, Y6 o Y7, causando que la compuerta invierta el bit del mensaje recibido y corrigiendo el error automáticamente.

Las soluciones alternativas para decodificar, similares a una tabla en ROM, tiene la misma simplicidad de diseño. Para el caso del decodificador, el ROM necesita un direccionamiento de 7-bits y 4-bits de palabra o 128 palabras por 4-bits de ROM (Figura 4.7).

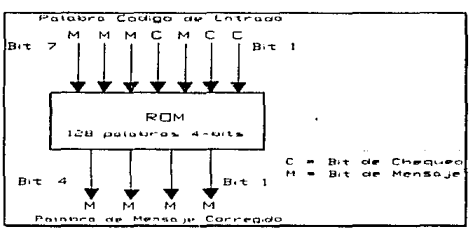


FIGURA 4.7

En esos casos, cuando la palabra código es correcta, la respuesta es limpiada. Solo existen 16 posibles palabras de mensaje (2^4) y 16 diferentes palabras código de corrección. La palabra de mensaje es guardada en cada una de esas 16 direcciones del ROM. Cuando una palabra código de 7-bits es

presentada como entrada, la palabra de mensaje será llevado inmediatamente a la salida.

Por ejemplo, si la palabra código válida recibida es 1011011, es reasignada a las direcciones de entrada de la ROM. Los datos guardados en esa localidad son 1011, que son los 4 bits del mensaje puestos en esa localidad, los datos en la localidad X=X para 16 palabras código correctas.

Los restantes 112 (2^7 - 2^4) localidades de ROM corresponden a palabras código erróneas. Dado un conjunto de 16 posibles palabras código correctas, y asumiendo que solo ocurre un error en un solo bit, la palabra código solo puede cambiar en siete diferentes maneras. Esto significa que existen 16*7=112 posibles palabras código erróneas, porque debido a la redundancia en el, una palabra código recibida, que contenga un solo error, posee una palabra código correcta.

Se debe de almacenar algunas de estas 112 localidad que automáticamente corregirán una porción del mensaje de una palabra código. Si cada palabra código errónea es una dirección en una ROM, entonces los datos almacenados en esas localidades deberán ser las palabras código correctas. En otras palabras, los datos almacenados en Y=X, donde X representa la palabra código correcta.

Suponiendo que la palabra código 1010011 es recibida y presentada en las localidades de entrada de la ROM, esto significara que la palabra código será errónea, los datos almacenados en la localidad 1010011 serán la palabra código correcta, 1011, sin los bits de chequeo o paridad.

4.8.3 BENEFICIOS DEL CODIGO DE HAMMING

Considerando cada permutación por adelantado, es posible programar una ROM para realizar la decodificación en el receptor. El diseño basado en ROM es mucho más sencillo que utilizando circuitos OR-Exclusivos, y para palabras pequeñas, igual de rápido.

Aún más, los sistemas basados en ROM pueden ser extendidos para manejar mensajes y palabras código más largas. Con palabras código más largas, la eficiencia de los bits del mensaje contra los bits de chequeo se incrementa, la posibilidad de ocurrencia de un error de una palabra código dada, se incrementa. Los diseños basados en ROM son capaces de tener diferentes códigos, los cuales pueden detectar y/o corregir más errores en una palabra código, utilizando más bits de chequeo. En este caso, una ROM de diferentes dimensiones es substituida y programada, en el caso de circuitos OR-Exclusivos, un diseño completo es normalmente necesario.

4.9 ESTANDAR DE ENCRIPTACION DE DATOS

Se han presentado ya algunas herramientas para encriptar un mensaje. En general, estos métodos son tan sencillos, que no representaría algún problema que una persona no autorizada interceptara el mensaje y lo decodificara correctamente.

Desafortunadamente, si se complican más los esquemas, se gastara más tiempo desarrollando, configurando e implementando los sistemas en los equipos. Esto da lugar a que se adopten sistemas estandars. El uso de un estandar permite que el hardware sea independientemente desarrollado.

El Estandar de Encriptación de Datos (*DES Data Encription Standard*) fue desarrollado por IBM y certificado por la Agencia Nacional de Seguridad (NSA). Fue adoptada en 1976 como un estandar federal y aprobada por la Agencia Nacional de Estandars (NBS) en 1977. Estos estandars han reducido el hardware necesario, de hecho, se han implementado en circuitos integrados por muchos fabricantes.

El DES puede ser configurado tanto para bloques como para tramas. La única diferencia es la clave de cada una, la cual debe ser guardada para prevenir que sea descifrado por personas no autorizadas.

Cuando el DES es utilizado para la encriptación de bloques, siendo la llave de 64 bits de largo, 56 bits representan la secuencia del código de seguridad y el resto, 8 bits, son utilizados como bits de paridad. Cada uno de los bits de paridad se encuentra en secuencias de 8 bits.

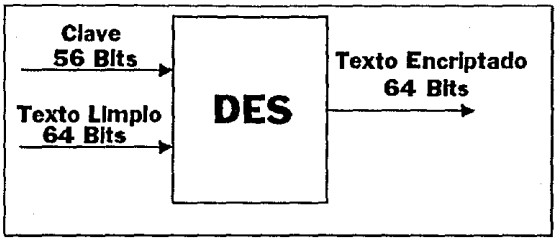


FIGURA 4.8

El uso del DES para encriptar bloques es mostrada en la Figura 4.8. En este modo, el DES es conocido como un libro electrónico de codificación (*ECB Electronic Code Book*), por lo que cada bloque puede ser visto como un código de substitución, donde se substituye cada texto limpio de 64 Bits por una secuencia particular encriptada de 64 bits. La operación es configurada por lo que un cambio de solo un bit en la llave o clave produciría un error que se propagaría en todo el texto encriptado, de manera que causaría que cada bit tenga aproximadamente un 50% de ser cambiado. Si la llave incorrecta es utilizada, un promedio del 50% de los bits encriptados serían erróneos.

En el modo de bloques, el mismo texto limpio siempre resulta el mismo texto encriptado, siempre y cuando no se cambie la llave. El DES realiza esta operación configurando un sistema complejo fuera de los bloques. Cada Construcción de Bloque Estandar (*SSB Standard Building Block*), consiste de una llave de 48 bits (derivada de la llave de entrada de 56 bits), la cual opera dentro del texto limpio de 64 bits. Los 64 bits de texto limpio están particionados en dos mitades, y la llave opera dentro de cada una de estas mitades para proveer la secuencia de salida. Esto es ilustrado en la Figura 4.9.

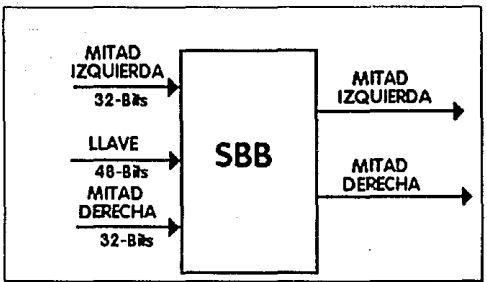


FIGURA 4.9

16 SSB son colocados en cascada para formar el sistema de encriptación por bloques. Una operación similar se lleva a cabo en el receptor para desencriptar, como se muestra en las Figuras 4.10 y 4.11.

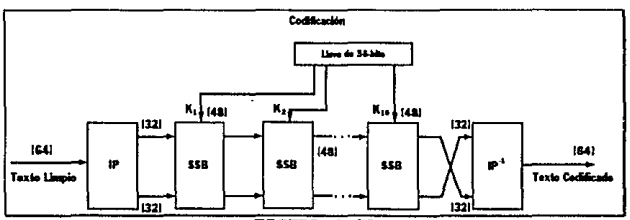
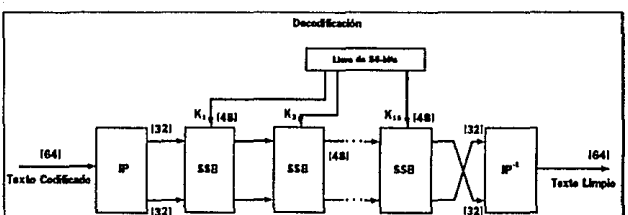


FIGURA 4.10



FIGRUA 4.11

La operación básica ECB puede ser implementada reconfigurando el sistema de tal forma que bloques de entrada idénticos no resulten en bloques encriptados iguales. Esto se realiza colocando una especie de "interferencia de entre bloques". Esto es, se tienen varios bloques interactuando entre si para proveer al sistema de un tipo de respuesta (transient) lo que se puede considerar como una memoria del sistema. Este modo es conocido como Encadenamiento de Código de Bloques (*CBC Cipher Block Chaining*). Esto es implementado si se provee de una retroalimentación, ya que se le puede sumar a cada bloque de mensaje limpio, la operación anterior del bloque codificado. Hay que notar que un error en la transmisión se propagará a los demás bloques, produciendo una probabilidad de error mayor. El modo CBC es usado generalmente para la comprobación (authentication) de un mensaje.

El DES puede ser configurado para proveer un tren codificado. En este modo, el DES, genera una trama de bits, la cual es sumada al texto limpio en forma de bit a bit.

Una una configuración particular de tramas es la conocida como Codificación por Retroalimentación (*CFB Cipher Feedback*), como se ilustra en la Figura 4.12, K es un número entre 1 y 64. Estos K bits son sumados a K bits del texto limpio, para producir K bits del texto encriptado. Estos bits del texto encriptado son regresados al bloque de entrada (shift register) para formar los K-ésimos bits menos significativos. De esta manera, la llave de secuencia es generada en bloques de K bits y es dependiente de los datos de entrada. Si ocurren errores en la transmisión, se propagarán a través de K bits, causando errores en un cierto número de bits de salida.

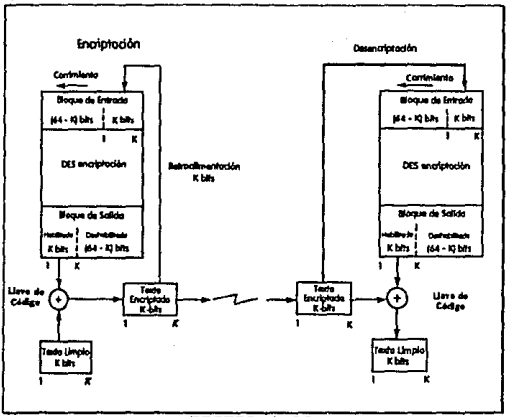


FIGURA 4.12

Una segunda forma de codificación por tramas, es la retroalimentación de la salida (*OFB Output feedback*). La diferencia entre la OFB y la CFB es que la codificación de la trama es independiente de los datos. El sistema es ilustrado en la Figura 4.13.

Hay que notar que es el bloque de salida el cual es retroalimentado. Así es que la trama de código (*Keystream*) y no el texto encriptado es retroalimentado para generar los nuevos bits de la trama de código, esto también es conocido como modo de código de autocódigo.

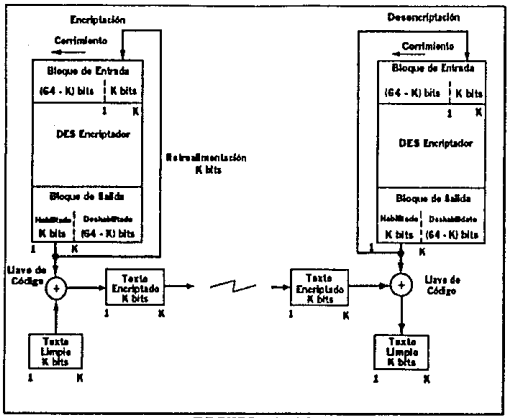


FIGURA 4.13

Además de la configuración del DES, la operación depende de una llave de 56 bits. Existen 2^{56} o aproximadamente $7 * 10^{16}$ posibles llaves. Esto es un número muy grande, así es que, una búsqueda exhaustiva en computadora tardaría demasiado tiempo. Sin embargo, a medida que las computadoras se vuelven más rápidas y más sofisticadas, será necesario incrementar éste número. De hecho, los circuitos integrados DES pueden estar colocados en cascada para incrementar la longitud de la llave. Si dos sistemas se encuentran en cascada, la llave efectiva sera de 112 bits de longitud.

El algoritmo DES es una herramienta matemática, no un circuito integrado, sistema de computación o algún otro dispositivo de hardware. Esta herramienta matemática ha sido implementada en hardware (el T7000A de AT&T *Digital Encryptor Processor*).

Contestando a la pregunta ¿Que es el DES?. Debemos empezar por aclarar que solo se mencionaran los aspectos básicos; el documento original del DES, se puede obtener del FIPS 46, 76 y 81 de la Oficina de Prensa del Gobierno de los Estado Unidos de Norteamérica.

4.9.1 DESARROLLO

La era de la computación trajo consigo el uso de las computadoras a las instituciones bancarias y comerciales. Inevitablemente se origino el crimen computacional. Esto era un problema, ya que una persona con suficientes conocimientos y una terminal de computadora podría transferir fondos a su propia cuenta y cargarlo a la cuenta de cualquier otra persona, o retirar dinero de cualquier cajero automático.

La compañía IBM se dio cuenta de esta situación y en 1960 formó un grupo de investigación para desarrollar un código de seguridad para proteger la información. En 1971, un código llamado "***LUCIFER***" fue desarrollado, el cual fue vendido a LLoyds en Londres, para que fuera utilizado con cajeros automáticos IBM.

4.9.2 LUCIFER

El sistema Lucifer fue un éxito, pero tenia algunas debilidades, por lo que IBM pasó otros tres años refinando y fortificando el sistema Lucifer. El código fue analizado una y otra vez por expertos en criptología, por lo que ya en 1974 estaba listo para ponerse al mercado.

Por el mismo tiempo la NBS (National Bureau of Estandars), la cual era responsable, desde 1965, de desarrollar estandars para los equipos de computación del Gobierno, inició un estudio sobre la seguridad computacional. La NBS vio la necesidad de un método de encriptación y solicitó un algoritmo capaz de realizar esta tarea, esto fue entre Mayo de 1973 y Agosto de 1974. El algoritmo debería estar realizado con el propósito de almacenar y transmitir información clasificada.

En respuesta a esta solicitud, IBM mandó su código LUCIFER. Este código consiste en un algoritmo bastante complejo desarrollado en un circuito integrado. Básicamente, la llave código va en una serie de ocho cajas "S". El código LUCIFER inicial tenía una llave de 128 bits. Antes de mandar su código a la NBS, IBM acortó esta llave removiendo más de la mitad de los bits.

4.9.3 PARTICIPACION DE LA NSA

La NSA (National Security Agency), tomó un interés enorme en el proyecto Lucifer, pues le había dado una mano a IBM en el proceso y desarrollo de las estructuras de las cajas "S". Era la primera vez que la NSA competía en su propio país.

Por años, la NSA había dependido de la información internacional, ya que monitoreaba las comunicaciones como transacciones del petróleo del Medio Este, las transacciones económicas de Latino América, Europa y el Lejano Oriente, también se hacia cargo de los mensajes de la inteligencia militar y la de los diplomáticos (ya que encriptaban sus mensajes utilizando técnicas muy rudimentarias), como también mucha de la información de países Comunista provenientes de países no Comunistas era recogida por la NSA y descifrada. Ahora el desarrollo de un sistema de alta seguridad en la encriptación de datos causaría a la NSA serios problemas.

4.9.4 PROBLEMAS Y CAMBIOS

Encuentros de la NSA y de IBM resultaron en reducir la llave de seguridad de 128 bits a solo 56, y el de clasificar ciertos detalles de la clave de las ocho cajas "S".

La NBS pasó su clave a la NSA para su análisis. La NSA certificó que el algoritmo estaba libre de cualquier debilidad matemática o estadística y lo recomendó como el mejor candidato para ser utilizado como Estandar Nacional de Encriptación de Datos (DES). Esta sugestión fue recibida con escepticismo. Pues ¿Acaso la clave era lo suficientemente larga para prevenir que alguna compañía penetrara o rompiera la clave, y lo suficientemente corta para los rompe códigos de la NSA? ¿Acaso existe un truco matemático (clasificado) que permita a la NSA romper rápidamente el código?

La Agencia no aceptaba del todo las cajas "S", e insistían en que cierto detalles deberían ser totalmente clasificados, la razón para ello era muy simple. Desde que el DES este comercialmente disponible y sea vendido en el extranjero, la NSA estará permitiendo que los extranjeros utilicen un código irrompible. La debilidad del diseño dentro del código permitiría a la Agencia el penetrar cada canal de comunicaciones y cada banco de datos que utilicen el DES. Los rompe códigos de la NSA debían asegurarse que la NSA pudiera romper el código. Como resultado, se levantaron compromisos burocráticos. Las partes de las cajas "S" del código se reforzaron, y la llave, la cual dependía de los usuarios, se reforzó igualmente.

Expertos en computación argumentaron, que sería posible construir una computadora utilizando un millón de circuitos integrados especiales de búsqueda, la cual sería capaz de probar un millón de posibles soluciones por segundo, y que

en 72,000 segundos (20 horas), todas las posibles combinaciones serían realizadas. Existiría por lo tanto un 50% de probabilidad que en solo 10 horas se rompiera el código (con 56 bits, existen 2^{56} combinaciones). Por otra parte, una computadora de tal tipo costaría $20'000,000 de dólares y sería construida en 5 años, lo que equivaldría a $10,000 dólares diarios. Si se utilizaran 24 horas para romper el código, cada código promedio costaría $5,000 dólares, por lo que no era economicamente rentable, pero a medida que la tecnología aumenta, los costos descenderían, por lo que estos costos podrían ser divididos por factores de 10 o incluso hasta de 100.

4.9.5 EL LUCIFER ORIGINAL

¿Qué hubiera ocurrido si la llave de 128-bits del Lucifer Original se pusiera en consideración? Existirían 2^{128} posibles soluciones. Esto es igual a 34.028237×10^{37}, o lo que es lo mismo un 34 seguido de 37 ceros. Este número es astronómicamente grande y es incomprensible para la mayoría de las personas. Si se probaran un trillón de soluciones por segundo, tomaría 34×10^{25} segundos o alrededor de 1.08×10^{19} años para romper el código. Esto es demasiado tiempo. Se calcula que la existencia del universo es de 2.6×10^{10} años (26 billones de años). Por lo que el código Lucifer de IBM, a la fecha, es probabilísticamente hablando, irrompible.

4.9.6 ACEPTACION DEL DES

En Junio 15 de 1977, el Estandar de Encriptación de Datos (DES) es oficialmente aceptado por el gobierno de los Estado Unidos de Norteamérica. Hoy día es utilizado, por la compañía HBO con el sistema Video Cipher II, entre otras. Con incrementos en la velocidad de las computadoras, nuevas tecnologías y precios más bajos, la seguridad de los códigos esta desapareciendo lentamente. Algunos autores le dan a lo más de 5 a 10 años, pero la mayoría concuerdo en que será menos de ese tiempo. La llegada del Video Cipher II, ha enfocado mayor atención hacia el DES y tarde o temprano, alguien romperá el código. Se sabe que el 9 de Octubre de 1986 el sistema Video Cipher II fue roto (pirateado), tan solo tomó 1 año y medio romper el código. Para principios de 1992 se advirtio a todos los subscriptores que el sistema Video Cipher II iba a cambiar para fines del año al nuevo sistema Video Cipher II+, sin embargo a mediados del año, compañias en los Estados Unidos anunciaron su circuito integrado capaz de romper este código.

4.9.7 SELECCION DEL ESTANDAR DE ENCRIPTACION DE DATOS

El DES especifica un algoritmo a ser implementado en un dispositivo electrónico y utilizado para encriptar y proteger los datos de computadora. Las publicaciones con respecto a este estandar [29] proveen una descripción completa del algoritmo matemático para encriptación y desencriptación de un código de información binaria.

La encriptación de datos convierten los datos a un tren de datos ininteligibles llamados código (Cipher). Desencriptar un código ocasiona que los datos regresen a su forma original. El algoritmo descrito en este estandar especifica tanto la operación de encriptado como la desencriptación, las cuales están basadas en números binarios llamados llaves. La llave consiste en 64 dígitos binarios ("0s" y "1s"), los cuales 56 de ellos son utilizados directamente por el algoritmo y los 8 restantes son utilizados para detección de errores.

Los códigos de datos binarios pueden ser protegidos, encriptandolos, utilizando el algoritmo DES en conjunto con una llave. La llave es generada de tal forma que los 56 bits utilizados directamente por el algoritmo son aleatorios o random y los 8 bits para detección de errores son colocados como paridad de cada llave impar de 8-bits o byte, pues existe un número impar de "1s" en cada 8-bits o byte. Cada miembro de un grupo de usuarios autorizados deberá tener la llave que fue utilizada para encriptar los datos. Esta llave, permite a cualquiera de los miembros, descifrar cualquier recepción encriptada. El algoritmo de encriptación de datos especifica que este, será conocido por todos aquellos que utilicen el estandar. La única clave escogida para utilizarse en una aplicación en particular da como resultado una encriptación de datos única. Seleccionando una

clave diferente para el código, produce un conjunto de códigos diferentes. La seguridad de la encriptación de datos dependen en la seguridad que se le provea a la llave utilizada para encriptar y desencriptar los datos.

Los datos solo pueden ser recobrados utilizando la misma llave que fue utilizada para la encriptación. Receptores no autorizados, que conozcan el algoritmo, pero no posean la llave correcta, no podrían deducir el algoritmo original de datos. Sin embargo, todo aquel que posea la llave y el algoritmo, puede descifrar el código y obtener los datos originales. Un algoritmo estandar, el cual esta basado en una llave de seguridad, provee las bases para intercambiar información por computadora encriptada.

4.9.8 MODOS ALTERNATIVOS DE UTILIZAR EL DES

La "Guía para la Implementación y Utilización el Estandar de Datos NBS" [30], describe dos diferentes formas de utilizar el algoritmo. Bloques de datos que contengan 64-bits pueden ser introducidos a un dispositivo donde se generará un bloque encriptado de 64-bits con una llave. Esto es llamado "*Electronic Codebook*"(*ECB*).

Alternativamente, este dispositivo puede ser utilizado como un generados binario de números aleatorios o random, los cuales son combinados con los datos limpios (sin encriptar) (del 1°. al 64°. bit) utilizando un "OR-Exclusivo". Para asegurar que el dispositivo de encriptación y desencriptación se encuentran sincronizados, sus entradas son colocadas a los primeros 64 bits del código que son transmitidos o recibidos. El segundo modo de utilizar el algoritmo de encriptación es conocido como "*Cipher Feedback*" (*CFB*).

El modo *ECB* genera bloques de un código de 64 bits. El *CFB* genera un código de la misma dimensión que el texto de entrada. Cada bloque de código es independiente de todos los demás cuando el *ECB* es utilizado, mientras que cada byte (conjunto de bits) del código, depende directamente de los 64 bits de código anterior cuando es utilizado en *CFB*.

El algoritmo de encriptación especifica una transformación de los valores de los 64-bits en otro único valor de 64-bits, basado en una variable de 56-bits. Si se utilizan completos los 64-bits de entrada y si los 56-bits variables son escogidos al azar, ninguna otra técnica que no sea tratando todas las posibles combinaciones garantizará encontrar la llave correcta. Existiendo más de 70´000,000,000,000,000 (70 mil billones o 70 cuatrillones en

la expresión americana o $7*10^{16}$) de posibles llaves de 56-bits, por lo que derivar una llave en particular de este modo es extremadamente largo. Más aún, si la llave es cambiada frecuentemente, el riesgo de que se derive la llave se disminuye. Sin embargo, los usuarios deben de estar prevenidos de que es posible deducir teóricamente la llave en pocos intentos (con un bajo por ciento de probabilidad de éxito dependiendo del número de llaves tratadas), y deben de ser cuidadosos al cambiar la clave tan seguido como sea posible. Los usuarios deben de cambiar la clave y deben de darle un alto nivel de seguridad en función de minimizar los riesgos potenciales de recepción de datos no autorizados.

4.9.9 METODOS DE ENCRIPTACION DE DATOS

4.9.9.1 METODOS BASICOS

La encriptación es una transformación de datos de su forma original a un código ininteligible. Dos formas básicas de transformación son utilizadas: la *permutación* y la *substitución*. En la Transformación de la Permutación se cambia el orden de los símbolos. En la Transformación por Substitución, los símbolos son remplazados por otros. Durante la permutación, los símbolos retienen sus identidades, pero pierden su posición. Durante la substitución, los símbolos retienen sus posiciones, pero pierden sus identidades originales.

El conjunto de reglas para una transformación en particular puede ser expresada en forma de un algoritmo. Transformaciones básicas pueden ser combinadas para formar una transformación más compleja. En un sistema de computación, los símbolos de los datos son grupos de uno o más dígitos binarios (1s y 0s) llamados bits. Un grupo de bits se le denomina byte. En aplicaciones computacionales, la encriptación por medio de la permutación reordena los bits de los datos. La encriptación por substitución remplaza un bit con otro bit o un byte con otro.

4.9.9.2 CODIGO DE BLOQUES

Un código que sea producido simultáneamente transformando un grupo de bits en un grupo codificado de bits es llamado *código de bloques* (*block cipher*). En general, estos grupos son de las mismas dimensiones.

4.9.9.3 CODIFICACION DE PRODUCTOS

Combinando las transformaciones básicas de permutación y substitución, produciendo una transformación más compleja es denominada *codificador de producto* (*product cipher*). Si las operaciones de permutación y de substitución son aplicadas a un bloque de datos, el código resultante es llamado *bloque codificado de producto* (*block product cipher*).

4.9.10 ALGORITMO DE ENCRIPTACION DE DATOS

4.9.10.1 INTRODUCCION

El algoritmo es diseñado para encriptar y desencriptar bloques de datos que contengan 64-bits bajo una llave de 64-bits. La desencriptación debe de estar acompañada de la misma llave usada para encriptación, pero con el esquema de direcciones de la llave alterada, así que el proceso de desencriptar sea inverso al proceso de encriptar.

El bloque a ser encriptado es sujeto a un permutación inicial, IP, y posteriormente a un computo complejo de llave dependiente, y finalmente, a una permutación que es la inversa de la permutación inicial (IP^{-1}). El computo de llave dependiente puede ser definido facilmente en términos de una función "*f*", llamada la función código, y una función "*KS*", llamada de programa llave. Una descripción de la computación esta dada, primero por detalles de como es que el algoritmo es utilizado para encriptación. Segundo, el uso del algoritmo para desencriptar. Y finalmente, una definición de la función de código "*f*" dada en términos de las funciones primarias, las cuales son llamadas funciones de selección "S_i" y finalmente por la función de permutación "P".

La siguiente notación es conveniente: dados dos bloques (L y R) de bits, LR denota el bloque consistente de los bits de L seguido por los bits de R. Puesto que la concatenación es asociativa, B_1, B_2, ..., B_8, por ejemplo, denota el bloque consistente de bits de B_1, seguido por los bits de B_2... seguido por los bits de B_8.

4.9.10.2 ENCRIPTACION

Un esquema de la encriptación computacional esta dada en la Figura 4.14:

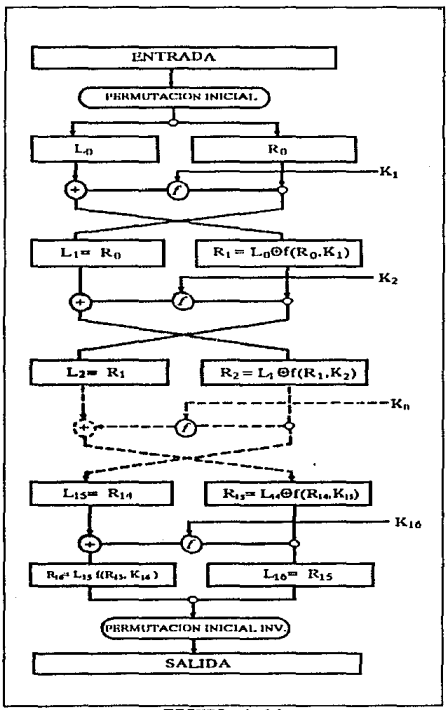


FIGURA 4.14

Los 64-bits a encriptar del bloque de entrada, son sujetos a la siguiente permutación, llamada permutación inicial, IP:

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	17	11	3
61	53	45	37	29	21	13	5
63	55	47	37	31	23	15	7

Esto es, la permutación de entrada tiene el bit 58 como primer bit, el bit 50 como segundo, y así sucesivamente, hasta el bit 7 que es el último bit. La permutación del bloque de entrada es así la entrada de la computación compleja de la llave dependiente. La salida de ese computo es denominado presalida (preoutput), posteriormente es sometido a la siguiente permutación, IP^{-1}, la cual es la inversa de la permutación inicial:

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Esto es, la salida de el algoritmo tiene el bit 40 de la presalida como primer bit, el bit 8 como segundo bit, y así sucesivamente hasta el bit 25 que el último bit.

El computo que utiliza el bloque de permutación de entrada para producir el bloque de presalida, consiste, pero para un intercambio final de bloques, de 16 iteraciones de un cálculo que es descrito posteriormente en términos de la

función de código *f*, la cual opera en dos bloques (uno de 32 bits y el otro de 48 bits) y produce un bloque de 32 bits.

Permitiendo que los 64 bits del bloque de entrada se definan como un bloque L de 32-bits, seguido por el bloque R también de 32-bits. Usando la notación ya antes definida, el bloque de entrada es, entonces, LR.

Permitiendo a K ser un bloque de 48 bits escogido de una llave de 64-bits. Entonces, la salida L´R´, de la interacción con la entrada LR, será definida por:

$$L' = R$$
$$R' = L \text{ xor } f(R,K) \qquad \text{Eq. 4.1}$$

donde xor denota una adición bit a bit en mod. 2.

Como ya se había dicho anteriormente, la entrada de la primera iteración del cálculo es el bloque de entrada permutado. Si L´R´ son las salidas de la decimosexta iteración, entonces, R´L´ será el bloque de presalida. Para cada iteración, un diferente bloque K de bits de llave es escogido de los 64-bits llave designados para la llave (KEY).

Con más notación, se puede describir las iteraciones del cálculo en mayor detalle. Si KS es una función donde toma n valores enteros entre 1 y 16 y toma de la llave (KEY) de entrada un bloque de 64-bits y produce de salida un bloque K_n de 48-bits, la cual es una selección permutada de bits de la llave (KEY), entonces:

$$K_n = KS(n,KEY) \qquad \text{Eq. 4.2}$$

donde:

K_n es determinado por las 48 distintas posiciones del bit de la llave KEY.

KS es llamada la llave de programa porque el bloque K, utilizado en la n-ésima iteración en la ecuación 4.1, es el bloque K_n determinado en la ecuación 4.2.

Como antes, si el bloque de entrada permutado es LR, y L_0 y R_0 son, respectivamente, L y R, y L_n y R_n son, respectivamente, L´ y R´ de la ecuación 4.1, donde L y R son, respectivamente L_{n-1} y R_{n-1} y K es K_n; esto es, cuando n se encuentra entre 1 y 16 entonces:

$$L_n = R_{n-1}$$
$$R_n = L_{n-1} \text{ xor } f(R_{n-1}, K_n) \qquad \text{Eq. 4.3}$$

El bloque de presalida será entonces: $R_{16}L_{16}$.

4.9.10.3 DESENCRIPTACION

La permutación IP^{-1} aplicada al bloque de presalida es la inversa de la permutación inicial, IP, aplicada a la entrada. Después, de la ecuación 4.1, se deriva que:

$$R = L'$$
$$L = R' \text{ xor } f(L',K) \qquad \text{Eq. 4.4}$$

Consecuentemente, para desencriptar, solo es necesario el aplicar el mismo algoritmo a el bloque del mensaje encriptado, tomando en cuenta de que cada iteración en el cálculo, y el mismo bloque de bits de llave, K, es utilizado durante la desencriptación, como fue utilizado en los bloque de encriptación. Utilizando la misma notación, esto puede ser expresado por las ecuaciones:

$$R_{n-1} = L_n$$
$$L_{n-1} = R_n \text{ xor } f(L_n,K_n) \qquad \text{Eq. 4.5}$$

donde $R_{16}L_{16}$ es el bloque de entrada permutado para los cálculos de desencriptación, L_0R_0 es el bloque de presalida. Esto es, para el cálculo de desencriptación, con $R_{16}L_{16}$ como entrada permutada, K_{16} es utilizado en la primera iteración, K_{15} en la segunda y así sucesivamente, hasta K_1 utilizado en la decimosexta iteración.

4.9.10.4 LA FUNCION DE CODIGO F

El modelo de cálculo de $f(R,K)$ esta dado en la Figura 4.15:

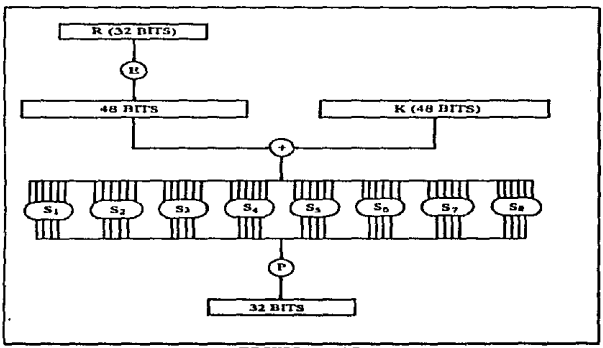


FIGURA 4.15

donde E denota una función la cual toma bloques de 32 bits de entrada y arroja bloques de 48 bits de salida. También E, siendo de 48 bits en la salida, es escrito como 8 bloques de 6 bits cada uno, los cuales son obtenidos, seleccionando los bits de la entrada en orden, de acuerdo a lo siguiente:

Tabla de selección de E-Bits

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

De lo anterior, los primeros tres bits de E(R) son los bits en las posiciones 32, 1 y 2 de R mientras que los dos últimos bits de E(R) son los bits en las posiciones 32 y 1.

Cada una de las funciones de selección únicas S_1, S_2,... S_8, toman bloques de 6 bits como entrada y arrojan bloques de 4 bits de salida. Esto es ilustrado en la Tabla 4.1 conteniendo a la función de selección S_1:

Función de Selección S_1

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

TABLA 4.1

Donde se comparan el número de Columnas (0-15), contra el número de Filas (0-3).

Si S_1 es la función definida en la Tabla 4.1 y B es un bloque de 6 bits, entonces $S1_1(B)$ esta determinada como sigue:

El primero y el último bit de B representa, en base 2, un número entre 0 y 15. Donde ese número es *j*. En la tabla 4.1, si se revisa la i-ésima fila contra la j-ésima columna, se observara que todos los número se encuentran entre 0 y 15, representados en bloques de 4-bits. Ese bloque es la salida $S_1(B)$ de S_1 para la entrada B. Por ejemplo, para la entrada, 011101, la fila es 01 (la cual es la fila 1), y la columna es determinada por 1101 (la cual es la columna 13). En la Fila 1, Columna 13, aparece un 5, así es que la salida será 0101.

La función de permutación, P, arroja 32-bits en la salida, permutando 32-bits del bloque de entrada. Así es que la función queda definida de la siguiente forma:

Función de Permutación P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

La salida P(L) para la Función P, definida por la Tabla 4.1, es obtenida de la entrada L, tomando el decimosexto bit de L, como primer bit de P(L), el séptimo bit como el segundo bit de P(L) y así sucesivamente hasta que el décimo quinto bit de L siendo el trigésimo segundo bit de P(L).

Ahora, siendo de S_1, ..., S_8 ocho distintas funciones de selección, P una función de permutación y E la función definida anteriormente, para definir $f(R,K)$, primero se definen los bloques B_1, ..., B_8 de 6 bits cada uno, para lo cual:

$$B_1B_2...B_8 = K \text{ xor } E(R) \qquad \text{Eq. 4.6}$$

El bloque f(R,K) es definido como:

$$P(S_1(B_1)S_2(B_2...S_8(B_8))) \qquad \text{Eq. 4.7}$$

Donde, K xor E(R) es primeramente dividido entre los 8 bloques, como lo indica la ecuación 4.6. Después cada B_i es tomado como entrada a S_i y los 8 bloques $S_1(B_1), S_2(B_2)$, ..., $S_8(B_8)$ cada uno de 4 bits se unen en un solo bloque de 32 bits, la cual forma la entrada para P. La salida, ecuación 4.7, es la salida de la función f para las entradas R y K.

4.9.11 FUNCIONES PRIMARIAS PARA EL ALGORITMO DE ENCRIPTACION DE DATOS

La elección de las funciones primarias KS, S_1, ..., S_8, y P es crítico para darle fuerza a la encriptación. Las funciones primarias S_1, ..., S_8 son:

S_1

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

S_2

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

S_3

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

S_4

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

S_5

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

S_6

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

S_7

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

S_8

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

La función primaria P es:

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Llamado por K_n, para 1 = n = 16, de un bloque de 48 bits en la ecuación 4.2. Una vez más, para describir *KS*, es suficiente el describir el cálculo de K_n de la Llave para n=1,2,...,16. Ese cálculo es ilustrado en la Figura 4.16:

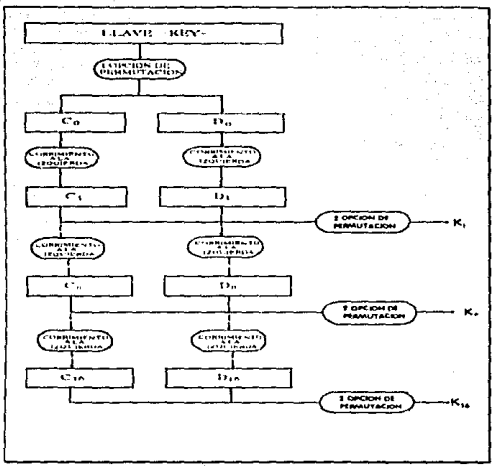


FIGURA 4.16

Para completar la definición de KS, es suficiente con describir las dos opciones de permutación, como el corrimiento a la izquierda. Un bit en cada 8-bits de la llave (KEY), puede ser utilizado para detección de errores en la generación, distribución o almacenamiento. Los bits 8, 16, ..., 64 son utilizados para asegurar el chequeo impar de cada byte.

Opción de Permutación 1 (PC-1)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

TABLA 4.2

La Tabla 4.2 ha sido dividido en dos partes, con la primera parte se determina como los bits de C_0 son escogidos, y la segunda parte determina como los bits de D_0 son escogidos. Los bits de la Llave son numerados del 1 al 64. Los bits de C_0 son, respectivamente, 57, 49, 41, ..., 44, 36 de la llave, y los bits de D_0 son 63, 55, 47, ..., 12 y 4.

Con C_0 y D_0 definidos, se pueden definir como los bloques C_n y D_n son obtenidos a partir de C_{n-1} y D_{n-1} respectivamente, para n = 1, 2, ..., 16, para completar C_n, se utilizan los siguientes bloques con corrimientos a la izquierda:

Número de Iteración	Número de Cambios a la Izquierda
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Por ejemplo, C_3 y D_3 se obtienen de C_2 y D_2, respectivamente, por dos corrimientos a la izquierda, y C_{16} y D_{16} son obtenidos de C_{15} y D_{15} por un solo corrimiento a la izquierda. En todos los casos, un solo corrimiento a la izquierda significa una rotación de los bits un lugar a la izquierda, así es que, después de un corrimiento a la

izquierda, los bits en las 28 posiciones quedarán 2, 3, ..., 28 y al final 1.

La segunda opción de permutación se determina de la siguiente manera:

Opción de Permutación 2 (PC-2)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Además, el primer bit de K_n es el décimo cuarto bit de C_nD_n, el segundo bit es el décimo séptimo, y así sucesivamente, y el cuadragésimo séptimo será el bigésimo noveno, y el cuadragésimo octavo bit será el trigésimo segundo.

4.10 VIDEO CIPHER II -VCII-

Video Cipher II es el sistema actualmente utilizado por HBO y Cinemax, entre otros. El sistema Video Cipher II que se muestra en la Figura 4.17 tiene las siguientes ventajas:

1. Encriptación de Video. Esto es realizado invirtiendo la polaridad del video y moviendo el nivel de color a una posición no estandar de la señal.
2. Un tren de datos de 88-bits y un nivel de color (3.58 MHz), en vez de pulso de sincronía. No se necesita portadora de sonido de 4.5 MHz.

Los 88 bits son utilizados para dos canales de audio, control de programación, información de sincronía, sistemas de seguridad y como canal auxiliar de datos.

El sistema es muy similar en principio a los sistemas estandars de satélite, los cuales utilizan un código de 24 bits para sincronía y audio, excepto que un código de 88-bits en el Video Cipher provee mayor flexibilidad en áreas de audio, control y seguridad. El tren de datos de 88-bits contiene encriptado el audio utilizando el Estandar de Encriptación de Datos (DES). Los dos canales de audio se encuentran filtrados y digitalizados. Cada muestra digital posee una secuencia binaria aleatoria y bits para corrección de errores, estos generados por el algoritmo DES. Los bits encriptados de audio podrían parecer completamente aleatorios, el formato del código para corrección de errores permite desencriptar correctamente cada uno de los bits individualmente, y si existe un doble error, se corrige por interpolación. (Por ejemplo, si un segmento de la señal se sabe que debe de ser "plano" y presenta algunos pulsos, estos son ignorados, y un voltaje que es la mitad entre la muestra anterior y la siguiente es insertada en ese lugar). Esto mejora la relación de señal a ruido del audio para

todas las señales de RF y asegura mayor calidad. El desencriptor debe de poseer la llave correcta del DES para poder desencriptar el audio. La falta de un portadora de sonido de 4.5 MHz trae consigo un fuerte interferencia en los receptores de TV en formato NTSC (Figura 4.17), y permite que la potencia total sea utilizada para el video. Arriba de 2 dB de relación señal a ruido es utilizado para todos los niveles de señal de los sistemas de video.

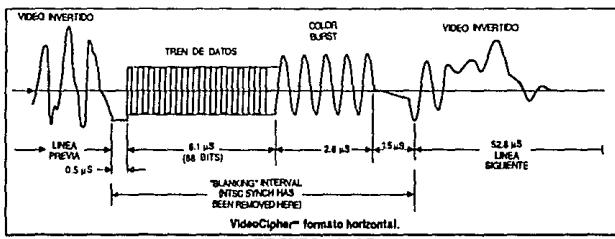


FIGURA 4.17

4.10.1 DIRECCIONES DE CONTROL

Los sistemas de control y seguridad se encuentran también encriptados. Como ya se ha mencionado, las llaves DES son 72 cuatrillones ($72*10^{15}$), y sin la llave apropiada, muchas de estas combinaciones deberán probarse para intentar descifrarlo. Una organización de multi llaves y métodos de distribución son utilizados. Cada desencriptor tiene una dirección única y un número de llaves DES.

En función de recibir programación codificada para un cierto periodo de tiempo, (por ejemplo, un mes o un millón de ciclos), el desencriptor debe de recibir el mensaje del satélite que contenga la llave mensual y el estado de servicio. Este mensaje es transmitido sobre el canal de

control a cada uno de los desencriptores, y se encuentra encriptado con la llave del desencriptor. Un desencriptor específico guarda esta información, y ningún otro desencriptor puede utilizar esta información sin la llave apropiada.

Cada programa es encriptado con una llave diferente. Mensualmente, después de cada billón de periodos, mensajes son transmitidos a los desencriptores autorizados, utilizando una llave diferente para cada programa, y una sola transmisión puede autorizar o no autorizar automáticamente a muchos desencriptores al mismo tiempo. Nuevos subscriptores pueden ser autorizados, y aquellos que no paguen sus cuentas serán desconectados, en adición, los subscriptores que soliciten diferentes programas pueden ser activados o desactivados. Esto puede realizarse para 600,000 desencriptores por hora. Los controles de mensajes y el corrector de errores, si no son recibidos correctamente, son ignorados hasta el siguiente mensaje. Además, los desencriptores tienen memoria no volátil en caso de pérdida de energía.

Un desencriptor robado puede ser desactivado, y no puede ser reutilizado hasta que el satélite lo active. Las direcciones de los desencriptores y las llaves, se encuentran encriptadas bajo el sistema DES, si son descubiertos y desencriptados, pueden ser sustituidos fácilmente por otros. Puesto que un desencriptor posee muchas llaves únicas guardadas en el, no se requiere de alguna modificación especial.

4.10.2 OTRAS VENTAJAS DEL VIDEO CIPHER II

Cincuenta y seis arreglos de programación, en cualquier

combinación pueden ser acomodadas utilizando el video Cipher II. Esta información se encuentra en la llave del mensaje mensual. Además, cada desencriptor solo recibe la programación en arreglos que se encuentren autorizados.

Utilizando la opción de arreglos, se pueden desconectar ciertos programas en ciertas áreas, simplemente acomodando los arreglos no autorizados en esas áreas. Cada desencriptor posee ciertas coordenadas basadas en los códigos postales, entonces, un programa que tenga seleccionados ciertos arreglos no autorizados para cierta área, desconectara todos los desencriptores para esa área. De la misma forma, programas con clasificación PG, X ó R, pueden ser acomodados en arreglos no autorizados, para evitar que los niños puedan verlos.

Teletexto, mensajes personales, mensajes de emergencia pueden ser manejados en el sistema de datos del VCII.

4.11 SISTEMA B-MAC

Scientific Atlanta desarrollo el sistema conocido como "B-MAC". Este es un nuevo formato de transmisión (las siglas MAC son el estandar para Multiplexión de Componentes Analógicos) que utiliza el multiplexaje por división de tiempo (TDM) de la señal analógica de luminancia y de los componentes de crominancia. Este sistema tiene algunas ventajas tecnológicas. Para métodos de transmisión por satélite, utilizando FM, y una señal de espectro de NTSC, no se aprovecha al máximo el canal de FM desde el punto de vista de relación señal a ruido. Esto causa destellos que aparecen en ciertas áreas de la imagen. También, la relación señal a ruido es menor para la crominancia debido la distribución del ruido en el canal de FM. El sistema B-MAC utiliza técnicas TDM para sobre llevar estas dificultades.

4.11.1 TEORIA BASICA

La multiplexión por división de tiempo se base en mandar porciones de información por los canales en una secuencia de tiempo. Por ejemplo, un sistema TDM puede mandar la información contenida en una señal NTSC en varios grupos en una secuencia de tiempo. La sincronía puede ser enviada, seguida de la información de crominancia y, luego, la luminancia. Los datos pueden ser almacenados, combinados, y convertidos a video. En adición, datos multiniveles (para sistemas de seguridad, direcciones, sonido stereo, teletexto, etc.) puede ser enviado con la información de sincronía. Las señales NTSC se encuentran en multiplexión por división de frecuencia (FDM). Toda la información es enviada al mismo tiempo (crominancia, luminancia, audio, etc.) en diferentes frecuencias, por ejemplo, la portadora de video, subportadora de color y la subportadora de sonido. Puesto que las subportadoras son eliminadas en un sistema

TDM, la intermodulación entre componentes no es un problema, puesto que los componentes no se encuentran al mismo tiempo.

Puesto que se requiere la misma velocidad para la transmisión de la información (programa de TV), el hecho de que se utilicen transmisiones secuenciales de los varios componentes, es necesario un tiempo de compresión de los componentes, como se muestra en la Figura 4.18:

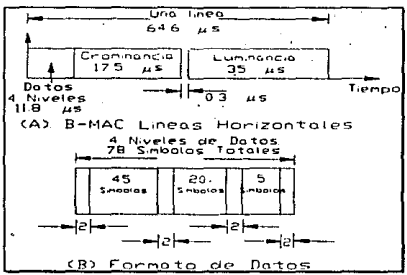


FIGURA 4.18

En el canal del satélite, es necesario un ancho de banda para poder realizar esto. La compresión en tiempo es un cambio de tiempo para el ancho de banda. Por ejemplo, si se pone una grabación al doble de velocidad, el audio se dobla en frecuencia, y la grabación tarda solo la mitad en ejecutarse. Este proceso es también utilizado en la duplicación de cintas para adelantar la producción. Este es un concepto obvio para todos aquellos que han tocado un disco o una cinta a una velocidad mayor de la que fue grabado.

Otro beneficio del sistema B-MAC es que la señal de crominancia se encuentra en base banda, por lo que la relación del canal de crominancia se mejora, transmitiendo

solo la componente de crominancia por línea, ya sea R-Y o B-Y, el tiempo de transmisión de la crominancia se reduce un 50%. Esto hace que el almacenamiento de la información de crominancia necesaria en el receptor se pueda realizar utilizando métodos analógicos o digitales. La señal de crominancia es filtrada en un ancho de banda restringido de 2 MHz o menos. Las componentes R-Y o B-Y se encuentran en bandas limitadas, y muestras de datos alternados son eliminados para reducir la velocidad de los datos a unas 7 muestras por microsegundo. Una velocidad de 14.31818 MHz es utilizada, puesto que es lo mejor que se puede escoger para una transmisión NTSC (ya que es 4 veces 3.579545). Para video digital, una velocidad de reloj de 13.5 MHz es recomendada para las componentes codificadas. Esto es, después de descartar muestras alternadas, da 7.16 muestras por microsegundo. Las muestras R-Y y B-Y son guardadas en una memoria de 385-bytes. Leyendo la memoria a 7.16 MHz * 3 = 21.48 MHz, por lo que la señal en tiempo fue comprimida a 17.5 microsegundos. Utilizando un método similar, la señal de luminancia es comprimida a 35 microsegundos. Esto deja 11.5 microsegundos para los datos.

La velocidad para los datos, audio y sincronía utilizada es de 1.86 megabits por segundo. Los pulsos de los datos son de 2 o 4 niveles durante el intervalo de barrido. Realmente 1.573 megabits/segundo son provistos durante el intervalo de barrido horizontal, y los seis canales digitales de audio toman 1.51 megabits por segundo. Los 62.5 Kilobits restantes son utilizados para un canal de datos; este canal de datos es encriptado y controlado por la cadena de televisión. Los canales de audio no utilizados puede ser utilizados como canales de datos. Los seis canales de audio se encuentran en sonido digital Dolby y utilizan 251.7 Kilobits/segundo, incluyendo códigos de error. La frecuencia de respuesta es de 20 Hz a 18 KHz con un ancho de banda de 30 dB. Los canales de audio pueden ser encriptados y

desencriptados separadamente.

Los intervalos verticales contienen todo el control de datos, el cual se sincroniza con los datos en 4 niveles en las líneas horizontales. Las primeras líneas (de la 1 a la 8) llevan datos de control para la sincronía y datos para recuperar el reloj. La sincronía se encuentra solo en una línea del intervalo de barrido vertical y permite recibir arriba de 1 dB de relación de portadora a ruido. Las líneas 9 al 13 contienen información de teletexto con 40 caracteres ASCII por cada línea.

4.11.2 MODOS DE OPERACION DEL B-MAC

El sistema B-MAC también provee más de 256 millones de direcciones o alrededor de 1 millón por hora con redundancia. Los decodificadores contienen múltiples direcciones para diferentes programas. El audio y los datos pueden ser desencriptados utilizando el DES, con llaves y códigos cambiando cuatro veces por segundo.

La encriptación del video esta acompañada por un proceso de corrimiento en tiempo para cada línea. El tamaño del paquete de datos digitales puede variar, mandando más datos en una línea que en la línea anterior. Esto también retrasa o avanza el empiezo de cada línea, como se observa en la Figura 4.19:

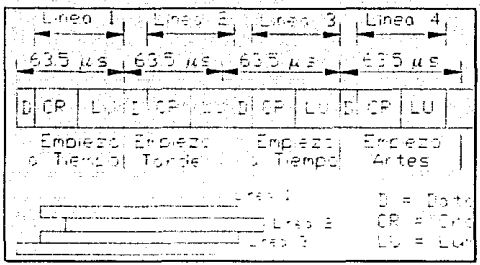


FIGURA 4.19

Ningún dato es omitido -simplemente es mandado o guardado para después-. Esto efectivamente encripta la imagen. Para efectuar la desencriptación, se revierte el proceso. Puesto que la transmisión solamente es corrida, no existe pérdida en la calidad del video.

El sistema B-MAC tiene una ventaja tecnológica en función del ruido. Los umbrales del sistema están definidos como la relación de portadora a ruido en el cual los demoduladores FM utilizados en el sistema producen 100 "clicks" o ciclos de error por segundo, con un sistema de frecuencia intermedia con un ancho de banda de 24 MHz (±1 dB). Arriba del umbral, la señal de crominancia del sistema B-MAC es de 8.3 dB mejor que en el sistema B-NTSC. La función de ruido en tres de los seis canales de audio es excelente con una relación de la portadora a ruido menor a 6.7 dB con una pérdida total de 4.7 dB.

CONCLUSION

Esta es un época de transición y de constante evolución tecnológica, especialmente en el campo de la Televisión, que pretende ofrecer una mejor calidad de vida en el tiempo de ocio, pero que por su continua evolución dificulta enormemente el proceso de formación personal, que día a día están al frente de estos nuevos sistemas o equipos, cada vez más sofisticados, con mayores posibilidades y con un tiempo de vida cada vez menor, debido al efecto de la obsolescencia, lo cual repercute así mismo en los planes de inversiones de todas las empresas dedicadas al campo profesional de la Televisión. Es, sin embargo, un reto difícil y a la vez apasionante que se debe de afrontar con valentía y optimismo.

Se podría pensar que introducir un sistema de Televisión Digital en el mercado significaría alterar todos los sistemas ya establecidos de Televisión convencional y cambiar todos los receptores de Televisión por nuevos modelos. Sin embargo, recientes trabajos, sobre todo en Japón y en los Estados Unidos, han hechos esfuerzos por realizar sistemas compatibles de Televisión Digital con la Televisión Convencional, a pesar de que no han tenido un éxito del 100%, se sigue trabajando y desarrollando sistemas mejorados. Se espera que para antes del año 2000 se tengan sistemas de Televisión Digital transmitiendose en estos dos paises.

No hay que perder de vista que el sistema típico de Televisión Digital es la Televisión de Alta Definición (HDTV). Sin embargo, no es la única solución para la Televisión Digital, ya que también se encuentra el Sistema de Televisión de Definición Extendida (EDTV), el cual podría ser un paso intermedio entre los sistemas convenciones de Televisión y el sistema de Televisión de Alta Definición.

La DCT es una herramienta poderosa y aún cuando debe su origen a la transformada de Fourier, ésta ha pasado a sustituirla para gran cantidad de aplicaciones; por su gran adaptabilidad y versatilidad. La DCT puede ser utilizada en múltiples áreas, sin embargo, son pocas las áreas en donde se aplica, entre estas se encuentra la electrónica en comunicaciones, donde encontró una gran aceptación.

Por su fácil aplicación, la DCT se puede implantar computacionalmente en máquinas pequeñas y aún así trabajar en tiempo real. Por ello es más práctico utilizarla en lugar de otro tipo de transformadas, por lo que resulta ideal para trabajarla en equipos DVE con ayuda de una computadora. Otra caracteristica importante es que tiene un bajo porcentaje de error, lo que es de gran importancia para aplicaciones que necesitan un alto grado de precisión (mayor al 99%). Por esto no es de sorprenderse que se utilice en los sistema JPEG y MPEG, porque además de que trabaja en tiempo real, se utiliza para calcular los valores de frecuencia de una señal.

Otro punto importante que hay que recordar, es que las operaciones más sencillas en el video digital, con procesamiento digital de señales, requieren de millones de operaciones por segundo, por lo que llegar a mayores niveles de integración en los circuitos integrados es vital para realizar estas operaciones. Al mismo tiempo se deben de buscar alternativas para el procesamiento digital del video. Entre las alternativas más factibles se encuentra el de realizar arquitecturas tipo "pipeline" o de cadena y el de realizar arquitecturas paralelas.

Uno de los puntos más importantes y más controvertidos del video digital es la gran cantidad de espacio que ocupa, debido a que una señal de video digital ocupa mucho más espacio que su contraparte analógico. Esto es tanto en espacio para guardarla, como en el ancho de banda para transmitirla, por lo que la compresión de las imágenes de video es fundamental. A partir de este punto, se han creado una serie de estandars, los mas importantes son: (1) el estandar para imágenes fijas creado por la JPEG, (2) el estandar para video conferencias creado por la CCITT y (3) el estandar para imágenes en movimiento, creado por la MPEG. Todos estos estandars tienen sus similitudes y diferencias, entre las similitudes se encuentra que todos ellos realizan una transformada en frecuencia utilizando la DCT, y entre sus diferencias, todos los estandars tienen sus propios algoritmos, lo que es una desventaja enorme, pues uno esperaría que un sistema de compresión de video funcionara en todas las formas posibles.

Otra desventaja es que ningún estandar ha resuelto eficazmente el problema de la compresión de video, pues los algoritmos más poderosos han codificado y decodificado imágenes con un rango de 20:1 sin pérdidas. Aún así, el espacio que ocupan es todavía demasiado grande, por lo que mayores rangos de compresión son necesarios, de 100:1 o mayores. A pesar de que se han reportado logros de compresión con rangos de 100:1, se ha perdido la calidad que posee la imagen de video digital, por lo que es necesario la creación de nuevas técnicas de compresión, como también la necesidad de herramientas más poderosas para este propósito.

Una solución a este problema, ha sido el de crear circuitos integrados que realicen el procesamiento digital del video. Algunas compañías como C-Cube, LSI Logic y Graphics Communication America, han realizado procesadores de video en tarjetas compatibles con PC´s. Sin embargo estas tarjetas son aún muy caras (alrededor de $10,000. USD) y solo traen implementado un solo algoritmo, ya sea el JPEG o el H.261 de la CCITT. Otra desventaja es que utilizan circuitos integrados de propósito específico desarrollados por estas compañías y por tanto no se encuentran en el mercado comercial, por lo que se tiene monopolizado el mercado de circuitos integrados que posean estos algoritmos. Una ventaja de estas tarjetas, es que la mayoría traen integrada una arquitectura paralela, lo que resuelve en parte el problema de trabajar en tiempo real, pero traen consigo otros problemas, como son, los de sincronización.

Un problema similar se ha presentado en la encriptación de la información de video digital. Generalmente la forma de encriptar el video digital se realiza a través del sistema conocido como Video Cipher II, el cual utiliza el algoritmo estandar de encriptación de datos (DES), sin embargo AT&T es la única compañía que realiza este algoritmo es forma de circuito integrado (el T7000A), y son pocas las compañías a las cuales se les vende este circuito.

El DES es el sistema estandar más seguro que se tiene para la encriptación de datos de video digital, pues posee una llave de 56 bits, lo que da un rango de 2^{56} ($7.2*10^{16}$) convinaciones para la llave. A pesar de esto, se ha tenido noticia de que se ha violado esta llave en múltiples

ocasiones. Reportándose por primera vez en Octubre de 1986, y por segunda ocasión en Junio de 1992, por lo que la seguridad de los datos utilizando este sistema se pone en discusión.

Poseer un sistema estandar de encriptación de datos lo suficientemente seguro para que el sistema sea inviolable es todavía una necesidad básica. Sin embargo, gracias a los avances de la ciencia y la tecnología se crean nuevos sistemas, cada vez más complejos y supuestamente más seguros, pero también se crean sistemas capaces de violar estos sistemas, por lo que aún no se tiene un sistema inviolable.

GLOSARIO

A/D	Convertidor Analógico-Digital
AC	Correinte Alterna
AM	Amplitud Modulada
CBC	"*Cipher Block Chaining*"
CCITT	Comisión Consultiva Internacional de Telefonía y Telegrafía
CD	Disco Compacto "Compact Disc"
CDMA	"*Code Division Multiple Access*"
CFB	"*Cipher Feedback*"
CGA	"*Graphics Communications Ameria Ltd.*"
codecs	Codificador-Decodificador
COMPANDING	Abreviación de: "*COMP*ressing-exp*ANDING*"
D/A o DAC	Convertidor Digital-Analógico
DAT	"*Digital Audio Tape*"
DC	Corriente Directa
DCT	Transformada del Coseno Discreta
DES	Estandar de Encriptación de Datos
DFT	Transformada Discreta de Fourier
DIF	"*Decimation in Frequency*"
DIT	"*Decimation in Time*"
DPCM	"*Digital Pulse Code Modulation*"
DRAM	Dinamic RAM
DSM	"*Digital Storage Media*"
DSP	Procesamiento Digital de Señales "*Digital Signal Processor*"
DVE	Efectos de Video Digital "*Digital Video Effect*"
ECB	"*Electronic Code Book*"
EDTV	Sistema de Televisión de Definición Extendida
FDM	Multiplexaje por División de Frecuencias
FEC	Corrección de Error hacia Adelante "*Forward Error Correction*"
FFT	Transformada Rápida de Fourier
FIPS	"*Federal Information Processing Standards*"
FIR	Filtro de Respuesta de Pulso Finito
FM	Frecuencia Modulada
FPB	Filtro Paso Bajos
HDTV	Televisión de Alta Definición "*High Definition Television*"
HO-DPCM	"*Higher Order-Differential Pulse Code Modulation*"
IDCT	Transformad Inversa del Coseno
ISO	"*International Standards Organization*"
JPEG	"*Joint Photographic Experts Group*"
lsb	Bit menos significativo

LSI	"*Large Scale of Integration*"
MAC	"*Multiplexed Analog Component*"
MOPS	Millones de Operaciones por Segundo
MPEG	"*Moving Pictures Experts Group*"
msb	Bit más significativo
NBS	"*National Bureau of Standards*"
NSA	"*National Security Agency*"
NTSC	"*National Television System* "
OFB	"*Output Feedback*"
OIRT	Organización Internacional de Radio y Televisión
overflow	Desbordamiento de Memoria
PAL	Sistema Europeo de 625 líneas
PCM	Modulación de Código de Pulsos "*Pulse Code Modulation*"
PIP	Imagen sobre Imagen "*Picture in Pincture*"
pixels	Abreviación de: "*PI*Cture *EL*ement*S*"
PN	Ruido Pseudo Aleatorio "*Pseudorandom Noise*"
PSK	Técnica para la transmisión digital de señales utilizando señales senoidales de diferentes fases "*Phase Shift Keying*"
RAM	Memoria de Acceso Aleatorio "*Random Access Memory*"
RDSI o ISDN	Red Digital de Servicios Integrados
ROM	Memoria de Solo Lectura "*Read Only Memory*"
S/N o SNR	Relación Señal a Ruido
SSB	"*Standard Building Block*"
SSVR	Video Grabadoras de Estado Sólido
TBC	Correctores de Base de Tiempos
TDM	Multiplexaje por División de Tiempo
TSC	"*Television Standars Converter*"
underflow	Subdesbordamiento de Memoria
VCII	"*Video Cipher II*"
VCR	Video Grabadora Digital
VLSI	"*Very Large Scale of Integration*"
VSP	Procesador de Señales de Video

BIBLIOGRAFIA

[01] K. Niwa, (1987); "Progress in Digital Video Technologies", *IEEE International Symposium on Circuits and Systems. Pre-Symposium Workshop on Digital Video*.

[02] K. B. Benson, ed. (1986); "*Television Engineering Handbook*", McGraw-Hill.

[03] K. Iinuma, et al., (1975); "Intername Coding for 4-MHz Color Television Signals", *IEEE Trans. Communications*, Vo.. COM-23, pp. 1461-66, Dec.

[04] A. D. Stally, (1973); "A Field Rate Converter Using Digital Techniques", *8th International TV Symposium*, May.

[05] D. F. Rogers et al. (1976); "Mathematical Elements for Computer Graphics", McGraw-Hill.

[06] CCITT Rec. H120, (1984); *CCITT*, SG XV.

[07] N. Ahmed et al., (1974); "Discrete Cosine Transform", *IEEE Trans. on Comput*. C-23, pp. 90-93.

[08] T. Ishiguro et al., (1982); "Television Bandwidth Compression Transmission by Motion-Comprensated Interframe Coding", *IEEE Commun. Mag.*, 10, pp. 24-30.

[09] R. Plompen et al., (1988); "Motion Video Coding in CCITT SGXV-The Video Source Coding", *Proc. of IEEE GLOBECOM '88*, 31-2, Nov.

[10] L. S. Golding et al., (1971); "Frequency Interleaved Sampling of a Color Television Signal", *IEEE Trans. on Commun.*, COM-19, No. 6, pp. 972-79.

[11] N. Susuki et al., (1984); "Picture Enhancement for NTSC TV Signals Utilizing Digital Signal Processings", *1984 IEEE International Conf. Consumer Electronics* (ICCE 84), pp. 116-17, June.

[12] T. Fukinuki et al., (1984); "Extended Definition TV Fully Compatible with Existing Standards", *IEEE Trans. on Commun.*, COM-32, No. 8, Aug.

[13] T. Rzeszewski, (1983); "A Compatible High-Definition Television System", *BSTJ*, Vol. 62, No. 7, pp. 2091-2111, Sept.

[14] K. Lucas, (1985); "B-Mac: A Transmission Standard for Pay DBS", *SMPTE Journal*, pp. 1166-72, Nov.

[15] T. Hatada et al, (1980); "Psychophysical Analysis of the 'Sensation of Reality' Induced by a Visual Wide-Field Display", *SMPTE Journal*, Vol. 89, pp. 560-69, Aug.

[16] M. H. Yassaie, (1986); "A High Performance Cascadable CMOS Transversal Filter and its Application to Signal Processing", *VLSI Signal Processing II*, IEEE Press.

[17] P. A. Ruetz et al., (1988); "A Chip Set for Real-Time 20-MHz DSP", *IEEE ICASSP '88*.

[18] T. C. Chen et al., (1988); "VLSI Implementation of a 16 x 16 DCT", *IEEE ICASSP '88*.

[19] T. Miyazaki et al., (1989); "A Single Chip Chrominance/Luminance Separator Based on a Silicon Compiler", to appear in *IEEE ICASSP '88*.

[20] T. Temma et al., (1984); "Data Flow Processor Chip for Image Processing", *IEEE Trans. on Electron Devices*, ED-32, No. 9, Sept.

[21] A. Kanuma et al., (1986); "A 20 MHz 32b Pipelined CMOS Image Processor", IEEE ISSCC '86 (1986). [22] T. Mori et al., "A Microprogrammable Real-time Image Processor", *IEEE ISSCC '86*.

[23] T. Beige et al., (1988); "A 20 nsec CMOS DSP Core for Video Signal Processing", *Tech. Dig. of ISSCC '88*.

[24] H. Harasaki et al., (1988); "A Single Board Video Signal Processor Module Employing Newly Developed LSI Devices", *IEEE J. Selected Areas in Communications*, Vol. 6, No. 3, April.

[25] M. Yamashina et al., (1987); "A Microprogrammable Real-time Video Signal Processor (VSP) LSI", *IEEE J. Solid-state Circuits*, Vol. SC-22, No. 6, Dec.

[26] K. Kaneko et al., (1987); "A 50 ns DSP with Parallel Processing Architecture", *IEEE ISSCC '87*.

[27] D. E. Dudgeon et al., (1984); "*Multidimensional Digital Signal Processing*", Prentice-Hall.

[28] K. Niwa, T. Araseki & T. Nishitani, (1990); "Digital Signal Processing for Video", *IEEE Circuits and Devices Mag.*, Vol. 6, No. 1, Jan.

[29] Federal Information Processing Standards (*FIPS*) Publication 46.

[30] Federal Information Processing Standards (*FIPS*) Publication 74.

[31] Ahmed, N., Natarajan, T. & Rao, K. R. (1974); *IEEE T.Cm.*; Vol. COM-23.

[32] Blair-Benson, K. & Fink, D.G., (1990). *HDTV: Advanced Television for the 1990s*. McGraw-Hill, New York.

[33] Boeck, W., (1989); *Televión Digital*, BRT.

[34] Cho, N. I. & Lee, S. U., (1990); *IEEE T.C.S.*; Vol.38, No.3

[35] González, R.C. & Wintz, P., (1986). *Digital Image Processing*. Addison Wesley, Reading, Mass.

[36] Hou, H. S., (1987); ; IEEE Trans. Acoust., *Speech, Signal Processing*; Vol. ASSP-35.

[37] Jauset, J., (1989); *Mundo Electrónico.*, Diciembre, pags. 111-122.

[38] Lathi, B. P., (1974); "*Introducción a la Teoría y Sistemas de Comunicación*"; Limusa. México.

[39] Luther, A. C., (1988). *Interactive Digital Video (DVI)*. IEEE Spectrum, September.

[40] Pratt, W. K., (1991). *Digital Image Processing*. John Wiley and Sons., New York.

[41] Raggi González, M. G. & Figueroa Nazuno, J., (1991). Transformada Discreta del Coseno: Fundamentos e Implementación. *XXXIV Congreso Nacional de Física*. México, D.F. 21-25, Oct.

[42] Rao, K. R. & Yip, P., (1990); "*Discrete Cosine Transform*"; Academic Press, INC.

[43] Vetterli, M. & Nussbaumer, H., (1984); *Signal Process*; Vol. 6.

[44] Wagh, M. D. & Ganesh, H., (1980); *IEEE T.Cp.*, Vol. C-29.

[45] Hamming, R.W. (1950). "*Error Detecting and Error Correcting Codes*", Bell System Technical Journal, vol. 29, 147-160.

[46] Kohavi, Zvi. (1978). "*Switching and Finite Automata Theory*", Second Edition. New York: McGraw-Hill.

[47] White, B. (1989). *Hamming-Code Decoding*, Dr Dobb's Journal, 52-56.