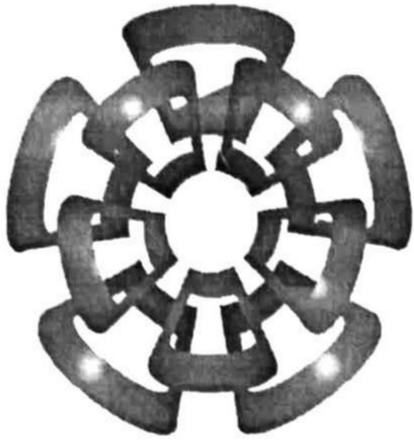


xx(79879.1)



CINVESTAV - IPN

Centro de Investigación y de Estudios Avanzados del IPN
Unidad Guadalajara

Diseño de una tarjeta para codificación de voz "CS-ACELP"



CINVESTAV I.P.N.
SECCION DE INFORMACION
Y DOCUMENTACION

Tesis que presenta
Karen Hernández Rueda

Para obtener el grado de
Maestro en Ciencias

En la especialidad de
Ingeniería Eléctrica

Guadalajara, Jal. Agosto de 1999

CLASIF.:	
ADQUIS.:	Tesis-00
FECHA:	16-03-00
PROCED.:	Depto. Servicios Bibliográficos
	\$

Depto. Servicios Bibliográficos

Diseño de una tarjeta para codificación de voz “CS-ACELP”

Tesis de Maestría en Ciencias Ingeniería Eléctrica

Por:

Karen Hernández Rueda

Ingeniero Electrónico
Instituto Tecnológico y de Estudios Superiores de Occidente
1992-1996

Becario del CONACYT, expediente no. 112946

Directores de Tesis
Dr. Arturo Veloz Guerrero
Dr. José Luis Leyva Montiel

CINVESTAV del IPN Unidad Guadalajara, Agosto de 1999

Agradecimientos ...

A Dios, por estar viva y porque siempre me ha dado la energía necesaria para seguir adelante, la ilusión y la magia para vivir en el mundo que diariamente construimos.

A mis padres, Braulio Hernández Ricárdez y Rosa Rueda Martínez; por el apoyo, la fe y el amor que me han brindado durante todos estos años. Gracias porque siempre me han motivado a seguir adelante y realizar día con día todos mis sueños.

A todos mis compañeros y amigos de la primera y segunda generación; especialmente a Larios, Meza, Bugarin, Núñez, Medina, Maestre, Govea, Iván, Omar, María, Tamar y Jorge Mex. Gracias por su amistad, enseñanzas, comprensión y apoyo incondicional.

A los doctores Arturo Veloz, J. Luis Leyva, Manuel Guzmán y Antonio Ramírez, por su amistad, consejos, enseñanzas, ayuda y comprensión.

A todos los ingenieros del CTS, especialmente a Mike y Moises, por su amistad, orientación y apoyo; por todas las veces que me sacaron de apuros.

A CONACYT y al país mismo, porque hizo posibles mis estudios de maestría mediante una beca.

A todos aquellos que directa e indirectamente contribuyeron en mi formación y especialmente a los que forman parte del CINVESTAV

Karen Hernández

Contenido

Prólogo	1
Capítulo 1. Introducción	2
Capítulo 2. Panorama General	5
2.1 Codificadores de voz	5
2.2 El problema de codificación digital	6
2.2.1 Definición de la razón de información	7
2.2.2 Definición de la calidad de voz	7
2.2.3 Definición de complejidad de codificación	8
2.2.4 Definición de retardo de codificación	8
2.3 Codificadores de forma de onda	8
2.4 Codificadores paramétricos	9
2.5 Codificadores híbridos	10
Capítulo 3. Modelo CS-ACELP	13
3.1 Antecedentes	13
3.2 Descripción del algoritmo	14
3.3 Norma G.729	19
3.3.1 Preprocesamiento	19
3.3.2 Análisis de predicción lineal y cuantificación	19
3.3.3 Ponderación perceptual	23
3.3.4 Análisis del tono	24
3.3.5 Libro fijo(algebraico)	27
3.3.6 Cuantificación de ganancias	29
3.3.7 Decodificación y postprocesamiento	32
3.4 Prueba en la PC	35
3.5 Diseño de la tarjeta usando el algoritmo CS-ACELP	37
3.6 Resultados	39
Capítulo 4. Conclusiones	41
Apéndices	
A. Especificación técnica	44
B. Diagramas esquemáticos	60
C. Lista de materiales	67
D. Diseño físico de la tarjeta.....	71
E. Archivos de fabricación	73
Referencias	78

Prólogo

En este documento se presenta el diseño de una tarjeta basada en un DSP para implementar el algoritmo conocido como CS-ACELP (Conjugate-Structure Algebraic-Code-Excited Linear-Prediction) o la norma G.729 (estandarizado así por la ITU: International Telecommunications Union), con la idea de utilizarlo para transportar voz sobre una red de paquetes.

La iniciativa de diseñar la tarjeta surge por dos razones:

1. La necesidad de transportar voz por una red de datos aprovechando el ancho de banda de la mejor manera posible.
2. Debido a que en la actualidad no se han tenido convenios entre fabricantes y proveedores de servicios de transmisión de datos para transportar voz por una red de datos

El algoritmo fue obtenido de la ITU en código ANSI C (un conjunto de archivos), se ordenó para ser compilado y verificar su funcionamiento en la PC antes de hacer el diseño de la tarjeta. Este codificador se clasifica como codificador híbrido y muestrea a 8Khz tramas de voz de 10 ms; cada muestra se representa en 16bits con lo que se obtiene una velocidad de 128Kbps a su entrada y a su salida, una velocidad de 8Kbps.

También se hace un estudio del algoritmo con el fin de entender su funcionamiento y que sirva de base para otros trabajos futuros, como por ejemplo, analizar si es factible hacer algún cambio en sus tablas de código que permitan reducir el algoritmo según la selección del idioma en que este se aplique.

Capítulo 1

Introducción

Nos encontramos en la era de las comunicaciones digitales, la señal de voz se representa en forma digital para que su transmisión y almacenamiento sean mucho más eficientes. Un tema que ha venido tomando mucha importancia con relación al tratamiento de la señal de voz ha sido la codificación digital de la voz.

La compresión de voz (considerada codificación digital de voz en este documento) es el proceso de remover información redundante en las comunicaciones de voz para reducir el ancho de banda requerido para su transmisión. Esta área ha permitido que los sistemas telefónicos se agrupen con sistemas de redes de cómputo para obtener mayor comodidad y mejores beneficios para la sociedad.

Aunque la codificación de voz utilizada en la telefonía digital convencional tiene muy baja complejidad en cuanto a su implementación, ésta no se considera como codificación óptima porque existe gran redundancia de información. La telefonía digital convencional no es funcional para muchas aplicaciones ya que el ancho de banda utilizado pudiera ser menor sin que esto implique la pérdida de las características principales de la voz. Esto ha motivado el estudio de otras técnicas de codificación.

Las diferentes técnicas de codificación están relacionadas con la manera en que la señal de voz se trata y estas se clasifican como codificadores de forma de onda, paramétrico e híbrido. Estas técnicas están ligadas con la razón de bits que manejan (velocidad de transmisión) y que puede ser alta, media, baja o muy baja [19].

Es importante recalcar que todos los codificadores contienen un grado de pérdida de información, que va desde el error de cuantificación hasta el modelo inexacto del sistema de producción de la voz. El CS-ACELP aunque también pierde parte de la información mantiene la inteligibilidad del mensaje y el reconocimiento de la persona que habla.

El CS-ACELP surgió para aplicaciones de telefonía inalámbrica y actualmente también es usado para otras aplicaciones dentro de la cual se encuentra la codificación de voz y su transporte, en pequeños paquetes a través de una red de datos [5]. En la actualidad no se han negociado convenios entre fabricantes y proveedores de servicios de transmisión de datos para el transporte de voz sobre redes de datos; sin embargo, muchos fabricantes de equipos han desarrollado métodos propios para integrar voz sobre tramas de retransmisión (Frame Relay) o sobre redes de protocolo de transferencia de

internet (IP). El algoritmo CS-ACELP (G.729) es un estándar recomendado por el Frame Relay Forum [22]. Esta organización trabaja junto con los fabricantes de equipos de telecomunicación para asegurar la interoperabilidad de los equipos de los fabricantes.

Por otro lado, la codificación ha tomado gran importancia para enviar información por la red a un costo efectivo y debido a los avances en el área de procesamiento digital de señales, se están buscando métodos para transportar tráfico de voz o video a través de la red. Utilizando el algoritmo CS-ACELP para transportar voz sobre una red de datos, se puede aprovechar el ancho de banda comprimiendo la señal de voz a 8Kbps; esto quiere decir, que un canal clásico telefónico de 64Kbps puede ser usado para llevar hasta 8 llamadas telefónicas.

Sabemos que la ITU cuenta con el algoritmo CS-ACELP (norma G.729) implementado en ANSI C y hasta hace poco (Marzo de 1998) se notificaron de dos implementaciones hechas en hardware por parte de dos organizaciones: FT CNET y NTT [8]. Respecto a la implementación de la organización FT CNET se sabe que fue realizada en un DSP de la quinta generación (TMS320C54x) de 16 bits de punto fijo. Sobre la implementación de la NTT no se tiene información.

El objetivo de esta tesis es diseñar una tarjeta basada en un DSP para el algoritmo CS-ACELP que sirva para su implementación y en el futuro, se tenga una máquina de codificación de voz con una buena calidad, que sea usada para enviar información por una red de datos a un costo efectivo. La idea es utilizar una sola tarjeta de codificación/decodificación de voz en un sistema que dé servicio a varios usuarios en una red de datos; la computadora con esta tarjeta estaría dedicada a controlar los paquetes de voz que reciba de redes global y local. La compresión la haría hacia la red global y la descompresión hacia la red local. Lo anterior se puede visualizar en la siguiente figura.

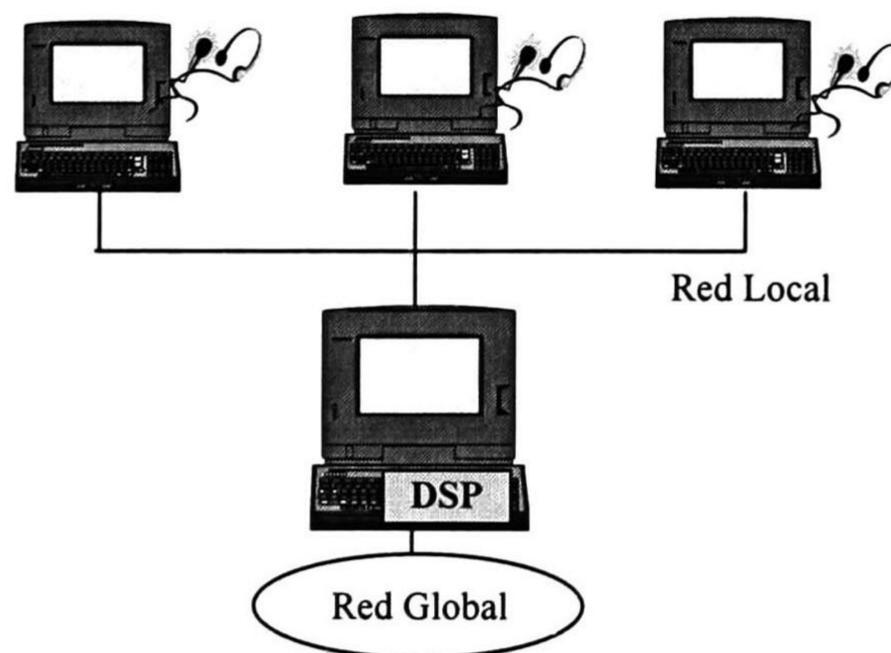


Figura 1. Diagrama con un sistema de compuerta.

La tesis está organizada de la siguiente manera:

En el capítulo 2 se da un breve panorama sobre los diferentes tipos de codificadores de voz y en el capítulo 3 se describe el algoritmo CS-ACELP, sus funciones, las pruebas realizadas en una PC para verificar su funcionamiento; se presentan también las consideraciones de diseño hechas para la elaboración de la tarjeta, basadas en la compilación del código para un DSP. También se describe el funcionamiento de la tarjeta y se presentan resultados. En el último capítulo, se tratan las conclusiones y se proponen algunos trabajos futuros sobre este tema. Al final del documento se agregan apéndices sobre descripción de la tarjeta, sus esquemáticos, la lista de componentes, el diseño físico de la tarjeta y archivos para su fabricación.

Capítulo 2

Panorama General

El tratamiento de la señal de voz de manera digital ha tomado gran importancia en nuestra vida diaria y ha originado estudios conducentes al desarrollo de métodos más precisos en el procesamiento de la señal de voz. Una de las áreas de interés ha sido la codificación.

Uno de los primeros codificadores que se desarrollaron tuvo como fin la seguridad telefónica durante la segunda guerra mundial. Su importancia fue creciendo según las necesidades de transmitir voz a largas distancias usando un ancho de banda menor y obteniendo una buena calidad de voz. Gracias a la aparición de la computadora y la telefonía celular nuevas aplicaciones aparecieron en el mercado, tales como telefonía celular digital, correo de voz, video de teléfonos y documentos de multimedia [21]. Con éstas, surgieron también nuevos codificadores digitales. Las estrategias y técnicas de codificación utilizadas por la mayoría de las instituciones dedicadas al desarrollo y mantenimiento de codificadores, se estandarizaron entre 1987-1996 de acuerdo a algunas aplicaciones específicas (un retardo específico y algunos otros requerimientos como velocidad (bit/s), complejidad y calidad de voz).

En este capítulo se introduce el concepto de codificación digital, el problema que representa y los factores que son considerados para clasificarlos; dando mayor énfasis a los codificadores híbridos porque el algoritmo CS-ACELP pertenece a esta clasificación.

2.1 Codificadores de voz

La representación digital (codificación digital) de la voz juega un papel muy importante en el área de las telecomunicaciones y el almacenamiento de información. Su función principal es representar la voz a través de palabras de códigos (digitales) permitiendo una comunicación robusta sobre un medio ruidoso por medio de algunas técnicas de codificación. Algunas de las aplicaciones que hoy en día tienen estos codificadores de voz son: comunicaciones de satélite móvil, radio móvil celular, multiplexaje de voz/datos para redes públicas y privadas, sistemas de video de teleconferencias, redes de correo de voz, equipos de multiplexaje de circuito digital y software de PC interactivo. Día con día continúan surgiendo más aplicaciones a medida que la tecnología de procesamiento digital de señal se hace más eficiente en la compresión. Las técnicas de codificación digital son evaluadas en términos de la calidad de voz digitalizada, la razón de información en la representación digital, la complejidad del algoritmo de codificación, el retardo introducido por el codificador y la robustez a

los errores de transmisión.

2.2 El problema de codificación digital

Se puede describir un sistema de codificación digital de una manera simple considerando el diagrama a bloques de la Figura 2. Se tiene una señal de entrada X que representa una forma de onda de voz (amplitud continua y tiempo continuo). Esta entra a un codificador para representar la señal de voz en una secuencia de dígitos binarios que, cuando se alimenta al decodificador, resulta en una señal de salida Y (amplitud discreta y tiempo discreto) que es una aproximación de la señal X. Una versión analógica de Y (línea punteada) se obtiene al pasarla por un filtro apropiado.

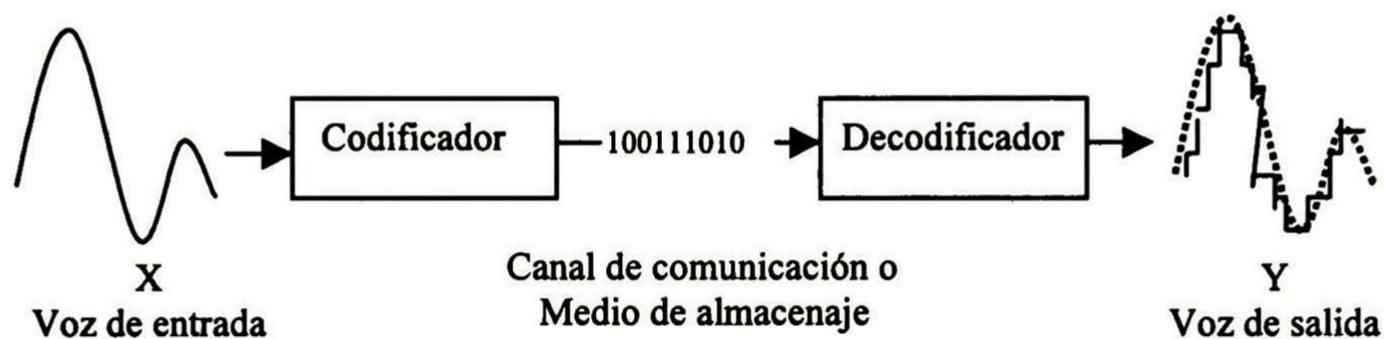


Figura 2. Sistema de codificación digital

La combinación de las operaciones del codificador y el decodificador forman un sistema de codificación, comúnmente conocido como codificador. En un sistema de comunicación de voz, los dígitos binarios atraviesan un canal de transmisión en tiempo real; en un sistema de grabación, por otra parte, los dígitos binarios generados por la señal de voz son almacenados y usados como entradas de un decodificador. En ambos casos, los dígitos binarios (bits), son modulados para obtener una señal (analógica) que pueda enviarse por un canal físico o al medio de almacenaje y esa señal es demodulada para que el decodificador pueda usarla. En general, los bits de la voz codificada están sujetos a errores aleatorios, y para asegurar que los bits que salieron del codificador estén libres de esos errores o que sean mínimos en el decodificador, algunas veces se usan técnicas de codificación para protección de errores [21].

Los mecanismos de modulación y de protección de error controlan el número de bits por segundo que el canal de transmisión físico (o almacenaje) asegura soportar. Este número tiene un límite superior y es conocido como capacidad del canal. En la práctica, los sistemas de voz digital operan con una razón de bits por debajo de esta capacidad, y con la razón de error de bits (varía entre 10^{-6} y 10^{-2} según sea la aplicación) que depende de la razón de bits tan bien como de la eficiencia del proceso de modulación/protección de error.

El problema principal en la codificación de voz digital es lograr una adecuada razón de información (menor número de bits) en la representación digital y la obtención de un nivel aceptable en la calidad de la voz con la mínima complejidad del codificador.

2.2.1 Definición de la razón de información

La razón de bits de un sistema de codificación de voz es medida en bits por segundo (bps), y usualmente se describe en kilobits por segundo (Kbps).

Dependiendo del tipo de codificador, esta razón tiene diferentes formas de expresarse. Por ejemplo, para los codificadores de forma de onda, la razón de bits I se puede expresar como el producto de la razón de muestreo f_s y el número de bits por muestra R : $I = f_s R$. El número de bits por muestra R que se necesita a la salida de un cuantizador de L niveles se define por: $R = \log_2 L$. En los codificadores paramétricos, la razón de bits es una suma de la razón de componentes usados para codificar los parámetros de las señales en el modelo del codificador.

2.2.2 Definición de la calidad de la voz

El desempeño de un sistema de codificación digital puede ser descrito mediante dos medidas de calidad: objetiva y subjetiva. La calidad de la voz se relaciona con la inteligibilidad del mensaje e identificación del hablante.

La medida objetiva de calidad considera: la relación señal a ruido (Signal to Noise Ratio: SNR) que describe qué tan cerca se encuentra la señal de salida Y con respecto a la señal de entrada X (ver Figura 1). Algunas pruebas objetivas se basan en obtener la distancia entre las formas de onda de las señales original y reconstruida.

Con respecto a la medida subjetiva de calidad se puede decir que es algo difícil de obtener; es la que se basa en la opinión y percepción de un grupo de personas. Se consideran dos tipos básicos de prueba subjetiva:

- MOS (Mean Opinion Score), es la más usada para determinar la calidad de la voz de un codificador. Para llevar a cabo estas pruebas se necesita un considerable grupo de hablantes, como voz de entrada y de oyentes, quienes estiman una calificación numérica entre 1 y 5, considerando la calificación de la siguiente forma: 5. Excelente 4. Buena 3. Regular 2. Pobre 1. Mala.
- DTR (Diagnostic Rhyme Test) y MTR (Modified Rhyme Test), son pruebas de rima. Una prueba de rima es un examen de varias opciones en donde los oyentes deben seleccionar la palabra pronunciada de un conjunto pequeño de posibles respuestas. Las pruebas DTR constan de un par de opciones por cada palabra, mientras que las pruebas MRT cuentan con seis opciones distintas.

La técnica más conocida para la medida de la inteligibilidad del mensaje es la prueba Harvard [20]. Esta prueba consiste en transmitir una lista de palabras fonéticamente balanceadas (PB) a través del codificador de voz bajo la prueba, y medir la proporción de palabras percibidas correctamente. La lista es seleccionada de tal manera que cada fonema contenido en la lista tenga la misma probabilidad de ocurrencia como sucede en una conversación normal de voz.

2.2.3 Definición de complejidad de codificación

Los algoritmos sofisticados para codificación digital tienden a ser implementados usando procesadores de señal digital (de propósito general y de propósito especial). Una medida apropiada de complejidad del codificador para estas implementaciones es el número de operaciones de multiplicaciones y sumas requeridas para codificar un segundo de voz (MIPS: millones de instrucciones por segundo) y se consideran básicamente en la parte de codificación porque suele ser mucho más compleja que la parte de decodificación.

2.2.4 Definición del retardo de codificación

El retardo de codificación considera la capacidad de almacenamiento de voz para analizarse, el tiempo tomado para procesar las muestras de voz almacenadas y factores de transmisión. Otro parámetro importante que no siempre se considera, es el retardo de comunicación. Este resulta del total de retardos introducidos en las operaciones del codificador y el decodificador.

En la tabla 1, se hacen comparaciones entre los codificadores de forma de onda, paramétricos e híbridos. Se consideran las definiciones de razón de bits, calidad de la voz y complejidad, entre otros factores.

Tipo de codificador	Razón de bits (Kbps)	Complejidad MIPS	Retardo ms	Calidad	MOS
PCM	64	0.01	0	Alta	>4
ADPCM	32	0.1	0	Alta	
ASBC	16	1	25	Alta	
MP-LPC	8	10	35	Comunicación	>2
SE-LPC	4	100	35	Comunicación	
LPC vocoder	2	1	35	Sintética	<2

Tabla 1. Comparación de los esquemas de codificación de voz

El costo del hardware en la codificación generalmente se incrementa con la complejidad. Sin embargo, con los avances que se tienen en la tecnología con los procesadores de señales (DSP) se tiende a decrementar el costo para un nivel de complejidad dado y, más significativamente, reducir el costo entre las técnicas de la baja y alta complejidad.

Una vez introducido las definiciones de razón de bits, calidad de voz, retardo y complejidad como características de los codificadores o la problemática de codificación, procedemos a describir sus clasificaciones: forma de onda, paramétrico e híbrido.

2.3 Codificadores de forma de onda

Los codificadores de forma de onda intentan reproducir la forma de onda de una señal de entrada después de ser codificada (los códigos obtenidos son usados para

reproducir la forma de onda de la señal). Son generalmente diseñados sin importar la señal de entrada así que pueden ser usados para codificar una amplia variedad de señales. Presentan cierta degradación en presencia del ruido y errores de transmisión. Sin embargo, pueden ser usados efectivamente para velocidades (bit/s) intermedias o mayores a 16 Kbps. PCM (Pulse Code Modulation), APCM (Adaptive PCM), DPCM (Differential PCM) y ADPCM (Adaptive Differential PCM) son algunos de los codificadores que se encuentran en esta categoría.

Aunque los codificadores de forma de onda generan voz de alta calidad y su complejidad no es muy alta, existen aplicaciones como por ejemplo, telefonía celular, radio móvil y almacenamiento; en donde la compresión de información es un factor importante, ya sea por un ancho de banda o por capacidad de almacenamiento restringido. Por esa razón, se usan otras técnicas de compresión como codificación paramétrica e híbrida.

2.4 Codificadores paramétricos (Vocoders)

Son codificadores especializados en la reproducción de la señal de voz; intentan reproducir una señal que suene parecida a la voz original. Este modelo supone que la voz es producida al excitar un sistema lineal, el tracto vocal, por medio de una serie de pulsos periódicos o ruido. Un esquema sencillo es el que se muestra la figura 3. Consiste de un analizador (extrae de la señal original un conjunto de parámetros que representan el modelo de reproducción de la voz) y un sintetizador (usa los parámetros para reproducir una señal de voz sintética). Operan en regiones no mayores a 4.8 Kbps. El LPC-10, Vocoder de canal y Vocoder de formantes son algunos de los codificadores que se encuentran en esta categoría.

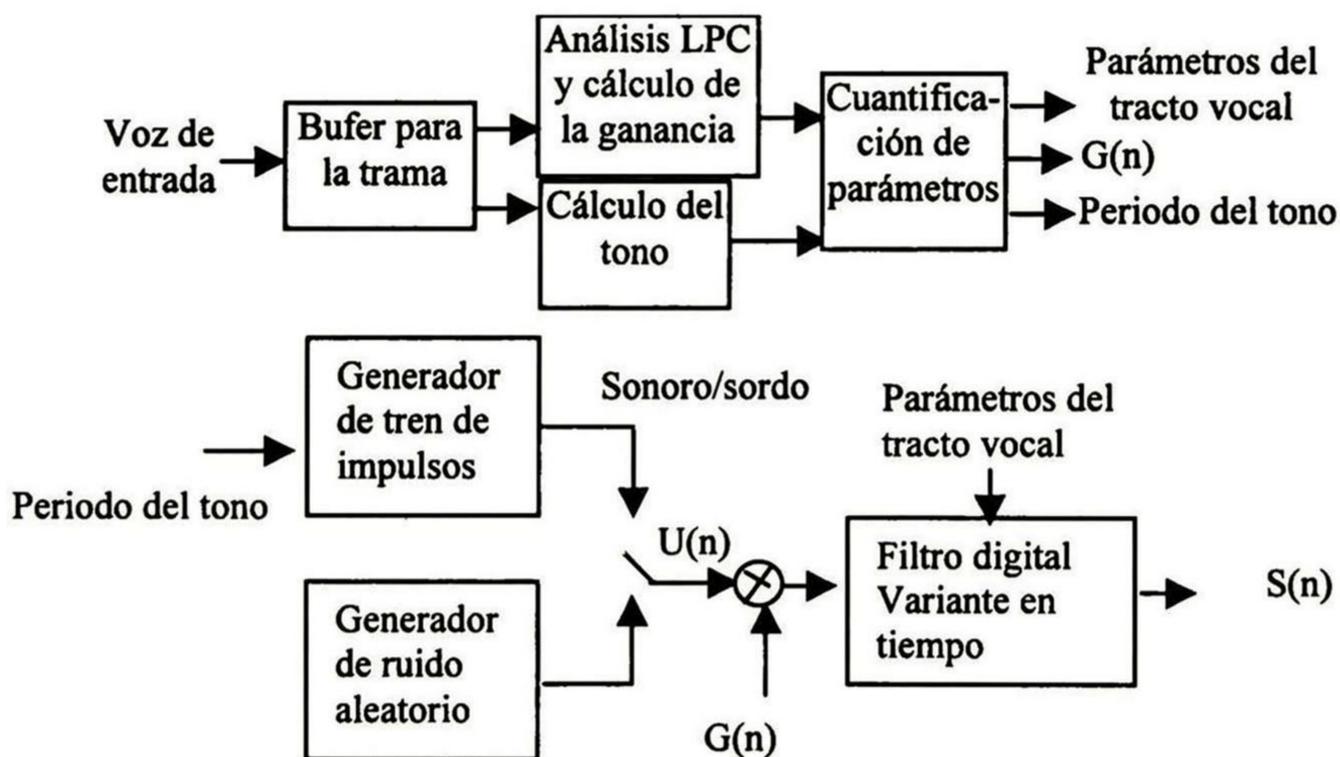


Figura 3. Uno de los modelos usados por los codificadores paramétricos.

Lo que se busca en este codificador es obtener un buen modelo tanto para generar la señal de excitación como para el tracto vocal. A diferencia del codificador de

forma de onda, lo que se busca en este codificador es considerar las características básicas de la señal para reducir la velocidad de transmisión; sin embargo, la calidad de voz obtenida en el proceso de decodificación tiende a ser muy pobre (modelo incorrecto del tracto vocal, de la excitación o de ambos). También toma en cuenta el mecanismo perceptual del oído humano, en el cual la magnitud de las componentes frecuenciales de la señal es más importante que la fase. Por esta razón, la forma de onda de la señal sintetizada puede llegar a ser muy diferente a la forma de onda de la señal original.

2.5 Codificadores híbridos

Como su nombre lo indica, estos codificadores son una combinación de los codificadores de forma de onda y los paramétricos; trabajan a baja velocidad (8Kbps). Surgen de la necesidad de utilizar codificadores a baja velocidad (bit/s) con una buena calidad de voz a la salida; esto es, obtener una señal de voz que sea inteligible y que no suene muy sintética. Han sido desarrollados para incorporar las ventajas que ofrecen los dos codificadores anteriores. Estos codificadores intentarán preservar las partes importantes perceptualmente de la forma de onda de la voz de entrada. El rango de operación en que se ubican es de 4.8Kbps-16Kbps.

APC (Adaptive Predictive Coding), RELP (Residual Excited Linear Predictive) y CELP (Code Excited Linear Prediction) son algunos de estos codificadores que trabajan en el dominio del tiempo y su esquema de trabajo es de predicción lineal (LPC: La idea básica del análisis de predicción lineal es que una muestra de la señal de voz puede ser aproximada como una combinación lineal de muestras de voz anteriores a ésta; minimizando la suma de las diferencias cuadradas, sobre un intervalo finito, entre las muestras de la aproximación y de la señal de voz actual). Las variaciones principales de estos codificadores se enfocan en el tratamiento de la señal de excitación de un filtro variante en el tiempo que origina la señal de voz.

Los estándares APC y RELP, que operan en el rango de 9.6Kbps-16Kbps, están basados en el método de análisis y síntesis (AaS). Sin embargo, en ellos no hay control sobre las distorsiones de la voz reconstruida y debido a su esquema adaptable, los errores acumulados en la trama anterior no son considerados en la trama siguiente. El método de análisis por síntesis (AbS) surge para cubrir estas deficiencias; cada subsistema es optimizado de tal forma que sobre toda la voz sintetizada se introduzca una mínima distorsión. Con este último esquema se cubre el rango de 4.8Kbps-9.6Kbps.

El método de análisis por síntesis (AbS) no es único en la codificación de voz sino que es una técnica general que también es usada en otras áreas de estimación e identificación. Su funcionamiento básico es como sigue. Se supone que la señal puede ser observada o representada de alguna forma (dominio del tiempo o frecuencia). Luego, se supone la forma teórica del modelo de la reproducción de la señal. El modelo tiene un número de parámetros que pueden ser variados para producir diferentes rangos de la señal observable. Para obtener una representación del modelo que sea de igual forma que la del modelo de la señal real, se aplica un procedimiento de prueba y error. Variando los parámetros del modelo de manera sistemática es posible encontrar una

serie de parámetros que puedan producir la señal sintética que se aproxime a la señal real con un mínimo error. Así, cuando se obtiene esta aproximación óptima, se puede decir que los parámetros del modelo son los parámetros de la señal real [19].

La estructura básica del sistema de codificación AbS-LPC se ilustra en la figura 4(a y b). Hay tres subbloques en el modelo que pueden variarse de tal manera que se iguale al modelo real (señal real) para obtener una buena señal de voz sintetizada: 1.- filtro variante en el tiempo 2.- señal de excitación 3.- proceso de minimización basada en un filtro perceptual. La operación básica del esquema se presenta a continuación:

1. Se inicializan los contenidos del filtro LPC y del filtro variante en el tiempo (filtro de tono) para predeterminar valores (usualmente se fijan como cero o ruido aleatorio de bajo nivel).
2. Se almacena una trama de muestras de voz, y usando el análisis LPC se calculan los coeficientes LPC.
3. Se usan los coeficientes LPC calculados y se construye un filtro inverso LPC para calcular el primer residual (error) no cuantificado. Este paso no es necesario si se hace una búsqueda de tono en ciclo cerrado.
4. La trama se divide en subtramas para hacer un análisis eficiente y así, determinar la señal de excitación.
5. Para cada subtrama:
 - a.- Se calculan los parámetros del filtro de tono (predictor de término largo), se puede hacer a ciclo abierto o a ciclo cerrado.
 - b.- El filtro de síntesis LPC y de tono se agrupan en cascada. De esta manera, se determina la mejor excitación secundaria que minimice la diferencia entre la voz sintética generada y la voz real (original).
6. La voz sintética final se genera con la excitación secundaria óptima a través del filtro en cascada pero con todos los valores iniciales de la memoria de los filtros.
7. Se repiten los pasos 2-6 para las siguientes tramas.

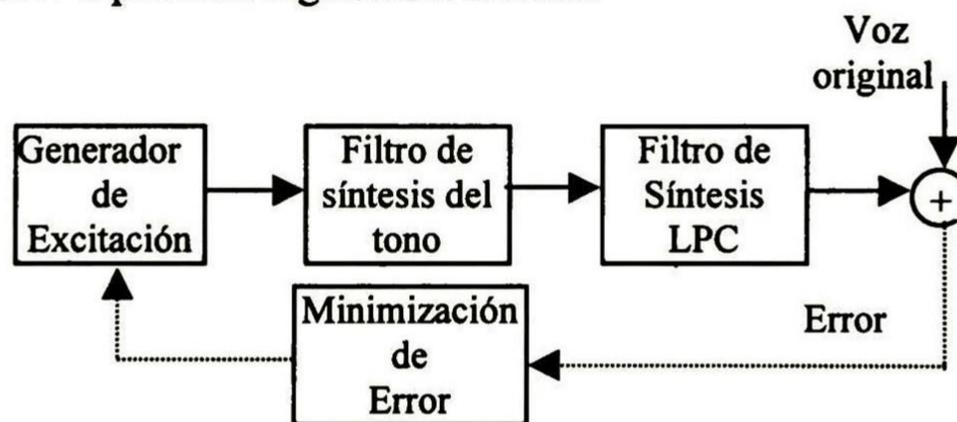


Figura 4a. Diagrama a bloques del codificador AbS-LPC



Figura 4b. Diagrama a bloques del decodificador AbS-LPC

Hay muchas variantes del método de codificación por predicción lineal usando análisis por síntesis (AbS-LPC), la primera variante fue Multi-Pulse LPC (MPLPC); la determinación de las localizaciones de los pulsos y amplitudes se llevan a cabo por un proceso de AbS. Una de las desventajas de este codificador es la gran cantidad de operaciones que realiza. Posteriormente, surgió el RPELPC (Regular-Pulse Excited LPC) como un híbrido del MPLPC y el RELP que fue adoptado para sistemas de radio móvil celular digital en Europa. El esquema más notable es el CELP [6] y es uno de los esquemas AbS-LPC más utilizados para bajas velocidades (bit/s). El procedimiento AbS en el CELP exige mucho cálculo de operaciones y debido a esto, se han estado buscando mejoras. Surge entonces una versión del CELP conocida como VSELP (Vector Sum Excited LPC) [7] que usa una estructura de código de libro condensado para reducir el cálculo de operaciones, el cual fue elegido como estándar celular digital en Norteamérica. También se ha visto que este esquema es bueno para velocidades más altas. Un esquema que se ha estandarizado a 16Kbps es el LD-CELP (codificador de bajo retardo) que se ha reportado ser equivalente o mejor que el ADPCM de 32 Kbps.

Capítulo 3

Modelo CS-ACELP

El codificador de Predicción Lineal Excitado por Código Algebraico con Estructura Conjugada (CS-ACELP: Conjugate-Structure Algebraic-Code-Excited Linear-Prediction) es un algoritmo usado para codificar una señal de voz digital de 128Kbps a 8 Kbps. La señal de voz debe ser representada en muestras de 16 bits (Modulación de Código de Pulso lineal de 16 bits) a una frecuencia de muestreo de 8KHz. La salida del decodificador también debe tener el mismo formato PCM lineal de 16 bits y ésta debe convertirse nuevamente a una señal analógica para que sea audible.

En esta parte del documento se menciona cómo surge el algoritmo CS-ACELP, se describe en forma general y de acuerdo a la recomendación de la ITU para explicar las pruebas realizadas en las PC y la manera en que se diseñó la tarjeta para ser implementado. También se comentan los resultados obtenidos en las pruebas realizadas.

3.1 Antecedentes del modelo CS-ACELP

El codificador CS-ACELP surgió principalmente para aplicaciones en comunicaciones inalámbricas en 1990. En la actualidad, también es utilizado en sistemas de comunicación personales (PCS), sistemas de satélite digital y algunas otras aplicaciones como voz paquetizada y equipo de multiplexaje de circuito.

La Unión Internacional de Telecomunicaciones ya había estandarizado el codificador PCM de 64Kbps, el ADPCM de 32Kbps y el LD-CELP de 16Kbps; el siguiente objetivo fue estandarizar un codificador de 8Kbps. La idea era diseñar un codificador que ofreciera bajo retardo, baja complejidad y una buena calidad de voz a una velocidad de 8Kbps sobre un canal de comunicación ruidoso. Al principio se buscaba que este codificador tuviera una longitud de trama menor a 5 ms pero se modificó a menos de 16 ms porque ningún algoritmo cumplió con esas características; posteriormente, con esta modificación, varios algoritmos participaron en la propuesta. Para el año 1993 fueron sometidos dos algoritmos: el CS-CELP (Conjugate Structure CELP) [1]-[2] y el ACELP (Algebraic CELP) [3].

El ACELP se basa en el uso de un libro de código algebraico. Los libros algebraicos son libros determinísticos en el cual los vectores del libro se determinan por un índice transmitido usando álgebra simple en vez de una búsqueda de tablas. Esta estructura tiene ventajas en términos de almacenamiento, complejidad de búsqueda y robustez; además el procedimiento de búsqueda para la selección del libro de excitación es mucho más rápido y los libros son más pequeños.

El CS-CELP basa su nombre en el hecho de que dos de sus libros (aleatorio y de ganancia) tienen una estructura conjugada. Esto es, la resultante de cada libro está dada como una suma de dos vectores conjugados. Los libros de código de ganancia VQ (Vector Quantization) mejoran su robustez. El libro conjugado consiste de dos sublibros. Cada uno de los sublibros es muy pequeño comparado con el libro simple, por lo que se requiere menos memoria y se reduce la complejidad de búsqueda del libro aleatorio. Además, es posible buscar un libro aleatorio grande si se usa una preselección. Ya que esto reduce requerimientos de memoria, es posible pasar a hacer un libro aleatorio adiestrado.

Aunque ambos algoritmos presentaron una calidad equivalente al ADPCM de 32Kbps bajo condiciones libre de errores, no cumplían todas las características requeridas por la ITU; así que se decidió combinar los mejores aspectos de ambos algoritmos para desarrollar lo que actualmente es el codificador CS-ACELP.

El codificador CS-ACELP fue estandarizado en el año de 1995 basándose en aritmética de punto fijo de 16 bits tras de efectuarse algunas pruebas. Estas pruebas fueron: trabajo con otros estándares de codificación de voz regionales e internacionales, voz de entrada que ha sido corrompida por un ambiente ruidoso y operación sobre un canal de transmisión degradado (incluyendo errores de bits aleatorios y un canal de radio simulado). Los resultados indicaron que este codificador tiene un buen desempeño (alta calidad de voz sobre un canal de radio ruidoso y baja complejidad) y bajo retardo. Trabaja a una velocidad de 8kbps, con un preanálisis de 5ms y con tramas de 10ms. No incluye codificación de canal [4]-[5].

3.2 Descripción del algoritmo

3.2.1 Descripción general del codificador

El codificador de voz CS-ACELP se basa en el modelo de codificación mediante la predicción lineal excitado por código (CELP). En este modelo, la señal decodificada localmente (reconstruida) es comparada con la señal original (muestras de entrada) para seleccionar los parámetros del codificador, de tal manera que el error cuadrático medio ponderado entre la señal original y la reconstruida se minimice.

Básicamente, la voz puede reconstruirse teniendo un filtro de síntesis apropiado y la correspondiente señal de excitación. El algoritmo CS-ACELP utiliza una señal de excitación que es construida por la contribución de dos libros de código: uno adaptable (indica periodicidad en la señal) y otro fijo (no periodicidad).

Este codificador opera con tramas vocales de 10 ms (80 muestras). La trama es dividida en subtramas de 5 ms para tener un mejor seguimiento del tono y la ganancia; además de reducir la complejidad de la búsqueda del libro de código. La figura 5 muestra el principio de codificación (compresión de la señal de voz). La señal de entrada (muestras de 16 bits) se introduce a un bloque de preprocesamiento donde es filtrada y

escalada para evitar componentes indeseables de baja frecuencia y reducir la posibilidad de desbordamiento en la aplicación de punto fijo (lenguaje C).

Posteriormente, esta señal preprocesada se utiliza para hacer todo el análisis requerido en la trama de voz y así, obtener los parámetros (coeficientes de filtros de predicción lineal, ganancias e índices de las tablas de códigos adaptable y fijo) necesarios para la reconstrucción de la misma en el decodificador.

Se efectúa un análisis de predicción lineal (LP) en cada trama de 10 ms para calcular los coeficientes del filtro de síntesis (filtro LP), que es usado en la reconstrucción del modelo del tracto vocal. Estos coeficientes se convierten en pares del espectro de líneas (LSP: L0, L1, L2, L3) por una transformación lineal. Los coeficientes LSP son robustos al ruido y evitan la inestabilidad en los filtros del decodificador. Estos coeficientes se usan cuantificados y no cuantificados en la segunda subtrama; mientras que en la primera subtrama se usan interpolados sin cuantificar. La señal de excitación se selecciona usando el procedimiento de búsqueda basado en el análisis por síntesis y consiste en dos pasos: un ciclo abierto y un ciclo cerrado.

El objetivo en ciclo abierto es estimar una frecuencia fundamental preliminar (tono) por trama, basado en la señal de voz ponderada perceptualmente. Una vez encontrado el tono preliminar (T_{op}) se comienza el ciclo cerrado.

El objetivo del ciclo cerrado es determinar el tono más exacto basándose en el T_{op} del ciclo abierto por medio (búsqueda del índice del libro) de minimización del error ponderado ϵ . Este procedimiento empieza ejercitando todos los índices del libro adaptable tratando de minimizar ϵ . El índice (P0, P1, P2), que produce el mínimo error, es elegido y usado en el siguiente paso. En un segundo paso, el mismo tipo de procedimiento se lleva a cabo sobre el libro fijo (libro algebraico). De este procedimiento, se obtiene el índice del libro fijo (S1, S2, C1, C2). También son obtenidas las ganancias del libro adaptable y fijo basándose en sus índices y son cuantificados vectorialmente usando un libro de estructura conjugada (F1, F2, G1, G2).

Finalmente, se actualizan las memorias de los filtros mediante la señal de excitación así determinada. Los parámetros codificados L_i , P_i , S_i , C_i , F_i y G_i son transmitidos en un flujo de bits (ver la tabla 2). Estos parámetros son la salida de información del CS-ACELP; se envían cada 10ms con un retardo implícito de 15ms.

Parámetro	Palabra de código	Subtrama 1	Subtrama 2	Trama (Total de bits)
Pares del espectro lineal	L0,L1,L2,L3			18
Retardo del libro código adaptable	P1,P2	8	5	13
Paridad del retardo de tono	P0	1		1
Índices del libro fijo	C1,C2	13	13	26
Signo del libro fijo	S1,S2	4	4	8
Ganancias de los libros (fase 1)	F1,F2	3	3	6
Ganancias de los libros (fase 2)	G1,G2	4	4	8

Tabla 2. Asignación de bits del CS-ACELP para una trama de 10ms (80 bits)

3.2.2 Descripción general del decodificador

El decodificador se encarga de recibir el flujo de bits enviado por el codificador para extraer los parámetros requeridos en la reconstrucción de la señal de voz (primero los decodifica y después reconstruye la señal de voz) cada 10 ms; esto se puede ver en la figura 6. Los coeficientes LSP cuantificados se interpolan y se convierten en coeficientes del filtro LP de cada subtrama. Después se realizan los siguientes pasos para cada subtrama:

- Se reconstruye la señal de excitación sumando los vectores de las tablas de código adaptable y fijo (escalados con sus respectivas ganancias).
- Se reconstruye la señal de voz filtrando la excitación obtenida por el filtro de síntesis LP.
- La señal de voz reconstruida pasa por una etapa de postprocesamiento (para mejorarla) y así obtener las muestras de entrada que tuvo el codificador.

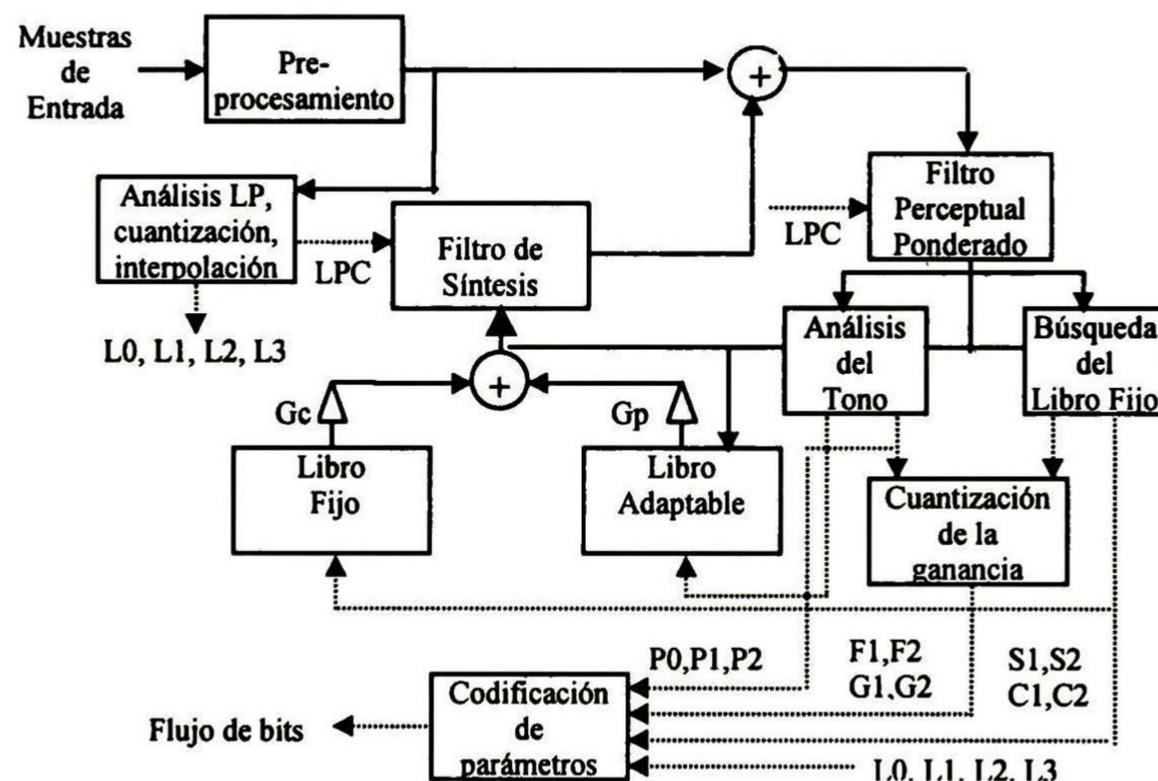


Figura 5. Codificador

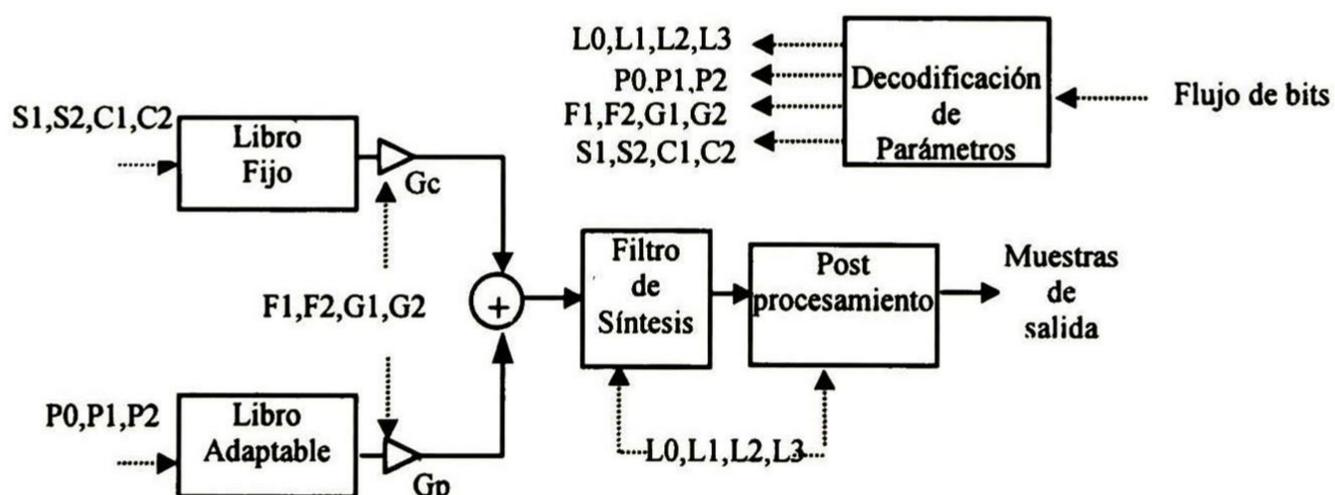


Figura 6. Decodificador

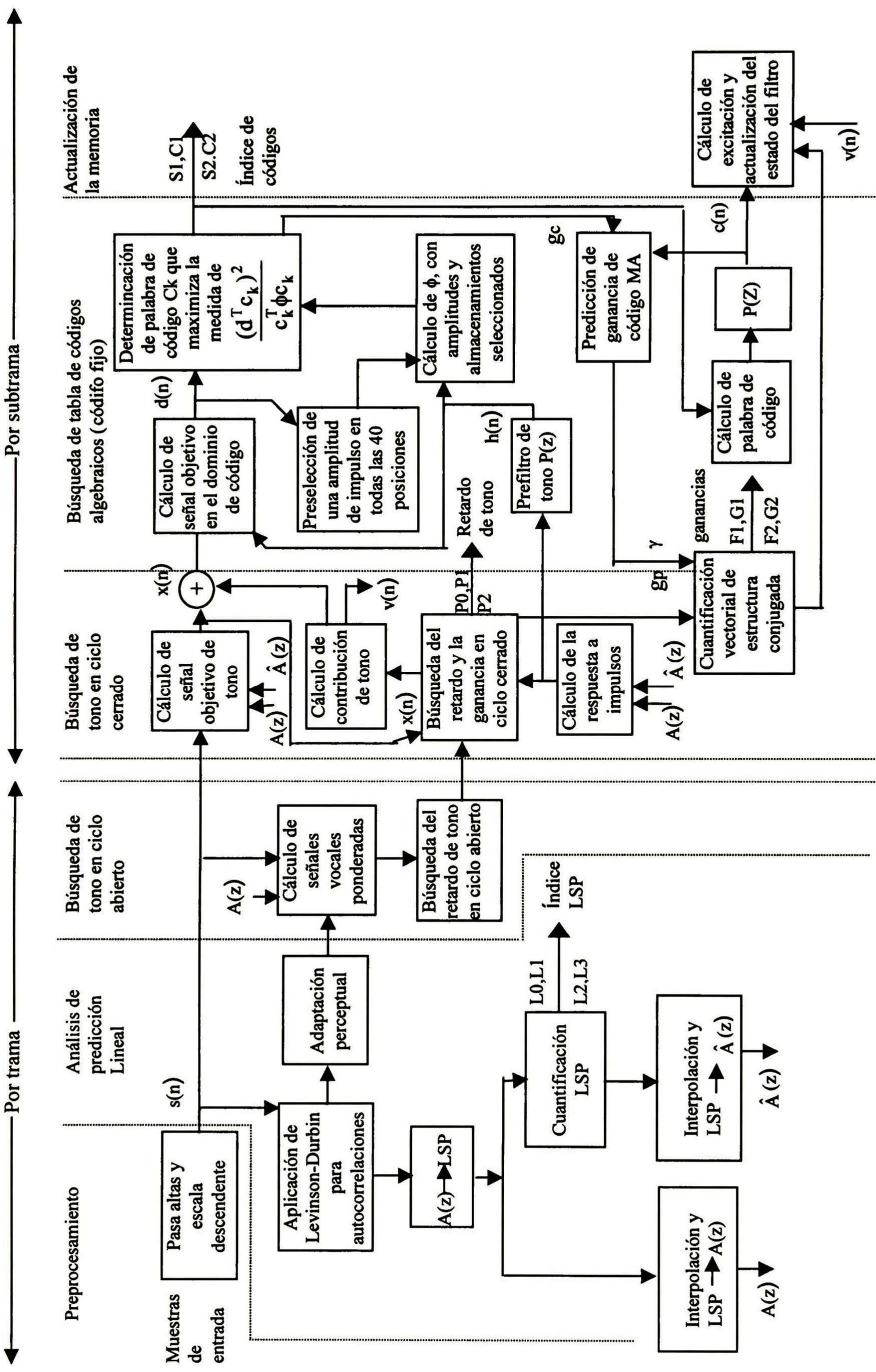


Figura 7. Diagrama a bloques del codificador CS-ACELP

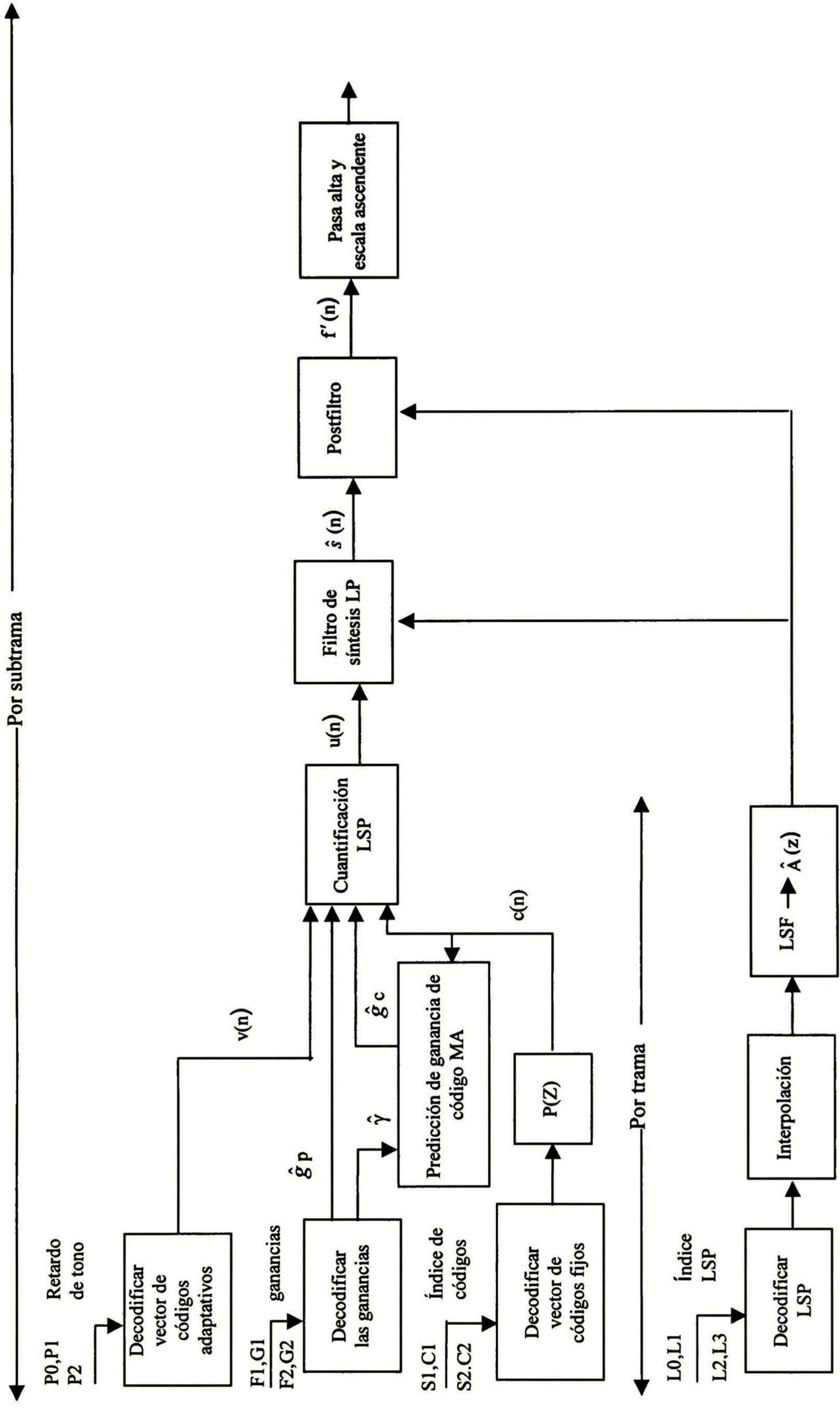


Figura 8. Diagrama a bloques del decodificador CS-ACELP

3.3 Norma G.729

Las funciones que realiza este algoritmo se puede explicar en los siguientes puntos [8] y se pueden ver en las figuras 7 y 8, que corresponden al codificador y decodificador respectivamente:

- Preprocesamiento
- Análisis de predicción lineal y cuantificación
- Ponderación perceptual
- Análisis del tono
- Libro fijo (algebraico)
- Cuantificación de ganancias
- Decodificación y postprocesamiento

3.3.1 Preprocesamiento

Las muestras de entrada (16 bits por muestra) son filtradas con un filtro pasa altas de polos y ceros de segundo orden con una frecuencia de corte de 140Hz; también, son escaladas dividiéndolas entre 2. Como se mencionó en la descripción del codificador, este filtro se usa para atenuar bajas frecuencias o componentes de directa y la división entre dos sirve para prevenir el desbordamiento en la implementación de punto fijo (algoritmo en C). La señal filtrada y escalada se refiere como señal de voz $s(n)$ y será usada en todas las siguientes operaciones del codificador para poder hacer la compresión de la señal de voz.

3.3.2 Análisis de predicción lineal

El objetivo de esta sección es obtener los coeficientes a_i para construir el filtro de síntesis y el filtro perceptual analizando las muestras de voz.

El análisis de predicción lineal (LP) se realiza una vez por trama usando el método de autocorrelación [9] con una ventana asimétrica de 30 ms; el uso de esta ventana produce mejor calidad de voz porque es como un alisador para el filtro LP. Cada 80 muestras, se calculan los coeficientes de autocorrelación de la señal que es pasada por la ventana y se convierten en coeficientes LP usando el algoritmo de Levinson-Durbin [9]. Después los coeficientes LP son transformados a frecuencia del espectro de líneas (LSF) para permitir la cuantificación e interpolación. Luego, los coeficientes LSF interpolados, cuantificados y no cuantificados se reconvierten de nuevo en coeficientes LP para construir los filtros de síntesis y de ponderación para cada subtrama. Los filtros de análisis y síntesis se basan en filtros de predicción lineal (LP) de décimo orden. El filtro de síntesis se define como:

$$\frac{1}{\hat{\lambda}(z)} = \frac{1}{1 + \sum_{i=1}^{10} a_i z^{-i}} \quad (1)$$

donde $\hat{a}_i, i=1, \dots, 10$, representa los coeficientes (cuantificados) de predicción lineal.

A continuación se muestran los pasos requeridos para hacer el análisis.

1.0- Ventana de análisis para seleccionar tramas de voz y cálculo de la autocorrelación:

La ventana de análisis consta de dos partes: la primera es media ventana de Hamming y la segunda un cuarto del ciclo de la función coseno. La ventana está dada por la ecuación:

$$\omega_{lp}(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{399}\right), n = 0, \dots, 199, \\ \cos\left(\frac{2\pi(n-200)}{159}\right), n = 200, \dots, 239. \end{cases} \quad (2)$$

Hay un preanálisis de 5 ms en el análisis LP, lo cual significa que se necesitan 40 muestras de la trama de voz futura. Esto se traduce a un retardo algorítmico extra de 5 ms en la etapa del codificador. El uso de una ventana asimétrica reduce el preanálisis sin afectar la calidad de la señal de voz [10]. La ventana de análisis LP se aplica a 120 muestras de las tramas de voz precedentes, 80 muestras de la trama de voz presente y 40 muestras de la trama de voz futura.

Los coeficientes de la autocorrelación son calculados de la señal de voz que pasa por la ventana:

$$r(k) = \sum_{n=k}^{239} \omega_{lp}(n)s(n)\omega_{lp}(n-k)s(n-k), k = 0, \dots, 10 \quad (3)$$

Con el fin de eludir los problemas aritméticos para las señales de entrada de bajo nivel, el valor $r(0)$ más bajo que puede tomar se limita a 1. También, se produce una expansión de la anchura de banda de 60Hz multiplicando los coeficientes de autocorrelación por:

$$\omega_{lag}(k) = \exp\left[-\frac{1}{2}\left(\frac{2\pi f_o k}{f_s}\right)^2\right], k = 1, \dots, 10 \quad (4)$$

donde $f_o = 60\text{Hz}$ es la expansión de la anchura de banda y $f_s = 8000\text{Hz}$ es la frecuencia de muestreo. La expansión en los coeficientes de autocorrelación reduce la posibilidad de mal condicionamiento en el algoritmo de Levinson-Durbin y también previene resonancias agudas indeseables.

Para reducir la posibilidad de mal condicionamiento debido al filtro que hay en la entrada, el valor $r(0)$ es multiplicado por un factor de corrección de ruido blanco de 1.0001. Los coeficientes de autocorrelación modificados son usados para obtener los coeficientes del filtro LP $a_i, i=1, \dots, 10$. usando el algoritmo de Levinson-Durbin.

2.0- Cuantificación de los coeficientes LSF (Line Spectral Frequencies): Los coeficientes del filtro LP a_i , $i=1, \dots, 10$ son convertidos a frecuencias de espectro de líneas (LSF) usando polinomios de Chebyshev [11]. Se definen dos polinomios de los cuales uno de ellos es simétrico y el otro antisimétrico, donde las raíces de cada uno de ellos se encuentran en el círculo unidad y están alternados entre sí, cada uno de estos polinomios tiene cinco raíces conjugadas y cuando uno vale 1 el otro vale -1 . Los coeficientes LSF se obtienen evaluando cada polinomio en 60 puntos igualmente espaciados entre 0 y π . Cada cambio de signo implica la existencia de una raíz y el intervalo de cambio de signo se divide entonces cuatro veces, para permitir un mejor seguimiento de la raíz. Ya que el cuantificador está basado en cuantificación vectorial (VQ), es más conveniente representar los términos LSF como frecuencias en radianes normalizadas. La manera en que se relacionan estas dos representaciones está dada por:

$$\omega_i = \arccos(q_i), i=1, \dots, 10 \quad (5)$$

donde q_i son los coeficientes LSF en el dominio del coseno y ω_i los coeficientes LSF en el dominio de la frecuencia.

Para mantener el retardo algorítmico tan bajo como sea posible, los coeficientes LP se actualizan cada 10 ms. Sin embargo, casi el espectro de la voz varía lentamente en el tiempo, y una actualización más lenta (por ejemplo, 20 ms) proveyería un mejor cambio entre la representación espectral y la razón de bit. Ya que una alta razón de actualización introduce una fuerte correlación entre los coeficientes de trama en trama, se usa un VQ predictivo. Durante el comienzo esta correlación no es muy fuerte. Para acomodar ambos tipos de correlación el predictor conmuta entre dos nodos, uno representa una fuerte correlación y otro representa una baja correlación. Otra ventaja de usar un bit separado para conmutar es que reduce la longitud del libro de código, reduciendo los requerimientos de almacenamiento. Para limitar la propagación de errores de canal, el predictor está basado en un filtro MA (Moving Average). La longitud de este filtro fue determinada empíricamente usando una base de datos grande [12] y fue encontrado que con un cuarto orden el predictor MA logra un buen compromiso entre el desempeño y la propagación de error.

El cuantificador está organizado de la siguiente forma: se aplica una predicción MA de cuarto orden conmutada para predecir los coeficientes LSF de la trama en curso. La diferencia entre los coeficientes calculados y su predicción se cuantifica mediante un cuantificador vectorial de dos etapas. La primera etapa es una cuantificación vectorial (VQ) de 10 dimensiones que utiliza una tabla de códigos $L1$ de 128 entradas (7 bits). La segunda etapa es una VQ de 10 bits que se aplica como una VQ de división mediante dos tablas de códigos de cinco dimensiones, $L2$ y $L3$, de 32 entradas (5 bits) cada una. El hecho de que la primera etapa sea un poco diferente de la segunda tiene que ver con el efecto de la correlación, para el primer caso algunas veces son fuertes y son tomadas en cuenta pero para el segundo no; sin embargo, reduce los requerimientos de búsqueda y almacenaje.

Para explicar el proceso de cuantificación es conveniente primero describir el proceso de decodificación. Cada valor cuantificado es obtenido de la suma de dos palabras de código, como sigue:

$$\hat{l}_i = \begin{cases} L1_i(L1) + L2_i(L2), i = 1, \dots, 5, \\ L1_i(L1) + L3_{i-5}(L3), i = 6, \dots, 10. \end{cases} \quad (6)$$

donde L1, L2, y L3 son índices del libro de código. Para garantizar que los filtros reconstruidos sean estables, el vector \hat{l}_i se ordena de modo que los elementos adyacentes tengan una distancia mínima d_{\min} [13]. Este proceso de reordenamiento se hace dos veces. La primera vez se establece un valor de 0.0012, y la segunda vez 0.0006. La incorporación de este proceso, en el procedimiento de cuantificación, asegura que cada posible vector \hat{l}_i reconstruido produzca un filtro estable. Después de efectuado el reordenamiento, los coeficientes LSF cuantificados $\hat{\omega}_i^{(m)}$ para la trama presente m se obtiene de la suma ponderada de las salidas del cuantificador precedente $\hat{l}_i^{(m-k)}$ y de la salida actual del cuantificador $\hat{l}_i^{(m)}$

$$\hat{\omega}_i^{(m)} = \left(1 - \sum_{k=1}^4 \hat{p}_{i,k} \right) \hat{l}_i^{(m)} + \sum_{k=1}^4 \hat{p}_{i,k} \hat{l}_i^{(m-k)}, i = 1, \dots, 10 \quad (7)$$

donde $\hat{p}_{i,k}$ son los coeficientes del predictor MA conmutado definido por el libro de código L0, y el índice del libro de código de un bit L0. En el arranque, los valores iniciales de $\hat{l}_i^{(k)}$ están dados por $\hat{l}_i = i\pi / 11$ para todo $k < 0$.

Después de haber calculado $\hat{\omega}_i$, se controla la estabilidad del filtro correspondiente y las resonancias agudas no naturales por medio de la revisión de la propiedad de ordenamiento $0 < \omega_1 < \omega_2 < \dots < \omega_{10} < \pi$. Si no se cumple la propiedad anterior, entonces, las frecuencias son removidas usando un proceso heurístico, que obliga a que haya un espaciamiento de 50Hz entre los coeficientes [13].

El procedimiento para codificar los parámetros LSF, se puede resumir como sigue. Para cada uno de los dos predictores MA se encuentra la mejor aproximación en curso de los coeficientes LSF. La mejor aproximación se define como la única que minimiza el error cuadrático medio ponderada:

$$E_{lsf} = \sum_{i=1}^{10} \omega_i^2 (\omega_i - \hat{\omega}_i)^2 \quad (8)$$

Las ponderaciones enfatizan la importancia relativa de cada LSF (las resonancias espectrales son perceptualmente más importantes). En la literatura se pueden encontrar otros tipos de procedimientos para obtener estos coeficientes.

El vector a cuantificar para la trama actual m se obtiene mediante:

$$l_i^{(m)} = \frac{\left[\omega_i^{(m)} - \sum_{k=1}^4 \hat{p}_{i,k} l_i^{(m-k)} \right]}{\left(1 - \sum_{k=1}^4 \hat{p}_{i,k} \right)}, i = 1, \dots, 10 \quad (9)$$

Se indaga la primera tabla de códigos $L1$, seleccionando la entrada $L1$ que hace mínimo el error cuadrático medio (no ponderado). Posteriormente se indaga la segunda tabla de códigos $L2$, que define la parte inferior de la segunda etapa. El error cuadrático medio (MSE) de la ecuación (8) se calcula, y se selecciona el vector $L2$ que produzca el error más bajo. Usando el vector $L1$ seleccionado de la primera etapa y la parte inferior del vector $L2$ de la segunda etapa, se extrae la parte superior de la segunda etapa de la tabla de códigos $L3$. Se selecciona el vector $L3$ que minimice el MSE ponderado. El vector resultante $l_i, i=1, \dots, 10$ se reordena asegurando una distancia mínima. Este proceso se lleva a cabo para cada uno de los dos predictores MA definidos por $L0$, seleccionando el predictor de MA $L0$ que produzca el menor MSE ponderado.

3.0- Interpolación de los coeficientes LSF: Los coeficientes LP cuantificados (y no cuantificados) se utilizan para la segunda subtrama. Para la primera subtrama, los coeficientes LP cuantificados (y no cuantificados) se obtienen mediante interpolación lineal de los parámetros correspondientes en las subtramas adyacentes. La interpolación tiene lugar en los coeficientes LSF en el dominio del coseno; se realiza en este dominio porque es fácil de implementar.

3.3.3 Ponderación perceptual

La señal de voz ponderación $s_\omega(n)$ en una subtrama se obtiene filtrando la voz a través de un filtro de ponderación perceptual $W(z)$. Este filtro de ponderación perceptual [14] está basado en los coeficientes del filtro LP no cuantificados a_i y se define como:

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)} = \frac{1 + \sum_{i=1}^{10} \gamma_1^i a_i z^{-i}}{1 + \sum_{i=1}^{10} \gamma_2^i a_i z^{-i}} \quad (10)$$

El uso de coeficientes no cuantificados da al filtro de ponderación una mejor igualación del espectro original. Los valores de γ_1 y γ_2 modifican la respuesta de frecuencia del filtro $W(z)$, y con eso, la cantidad de ruido ponderado. Es difícil encontrar valores fijos de γ_1 y γ_2 que den un buen desempeño para las diferentes características de la señal de entrada.

Se efectúa esta adaptación una vez por trama de 10 ms, pero aplicando un procedimiento de interpolación para cada primera subtrama, a fin de suavizar el proceso de adaptación. La forma del espectro se obtiene a partir de un filtro de predicción lineal de segundo orden, como resultado secundario de la recursión de Levinson-Durbin. Los

coeficientes de reflexión k_i se convierten en coeficientes o_i de razón de área logarítmicos (LAR) mediante:

$$o_i = \log \frac{(1+k_i)}{(1-k_i)}, i = 1,2. \quad (11)$$

Los coeficientes LAR son usados porque tienen mejor propiedad de interpolación que los coeficientes de reflexión. Los coeficientes LAR correspondientes a la trama actual de 10 ms sirven para la segunda subtrama. Los coeficientes LAR de la primera subtrama se obtienen mediante interpolación lineal de los parámetros LAR de la trama anterior. La envolvente del espectro se caracteriza como plana ($S=1$) o inclinada ($S=0$). Para cada subtrama esta caracterización se logra aplicando a los coeficientes LAR una función umbral. Para evitar cambios bruscos, se realiza una histéresis tomando en cuenta el valor de S en la subtrama anterior $m-1$,

$$S^{(m)} = \begin{cases} 0, & \text{si } o_1^{(m)} < -1.74 \text{ y } o_2^{(m)} > 0.65 \text{ y } S^{(m-1)} = 1, \\ 1, & \text{si } (o_1^{(m)} > -1.52 \text{ o } o_2^{(m)} < 0.43) \text{ y } S^{(m-1)} = 0, \\ S^{(m-1)} & \text{cualquier otro caso} \end{cases} \quad (12)$$

Si el espectro interpolado se clasifica como plano ($S^{(m)}=1$), los factores de ponderación se establecen en $\gamma_1=0.94$ y $\gamma_2=0.6$. Si el espectro es clasificado como inclinado ($S^{(m)}=0$), entonces $\gamma_1=0.98$ y γ_2 se adapta a la intensidad de las resonancias en el filtro de síntesis LP, pero limitado entre 0.4 y 0.7. En caso de registrarse una fuerte resonancia, el valor de γ_2 se fija más cerca del límite superior. Esta adaptación se hace para reducir la cantidad de ruido no enmascarable en las frecuencias de los formantes. La adaptación de γ_2 se logra usando un criterio heurístico basado en la distancia mínima entre dos coeficientes LSP sucesivos para la subtrama actual.

Los valores de γ_1 y γ_2 para las diferentes condiciones fueron obtenidos a través de muchos experimentos informales usando receptores expertos [8].

3.3.4 Análisis del tono

Como se mencionó en la breve descripción del codificador, en el ciclo abierto se estima un tono T_{op} por trama de 10 ms usando la señal de voz ponderada $s_w(n)$. El libro de código adaptable es usado para representar la periodicidad de la señal de excitación. El vector del libro adaptable seleccionado se representa por un índice, que corresponde a un cierto valor de retardo fraccionario. El valor de retardo fraccionario se busca en el ciclo cerrado basándose en el encontrado en el ciclo abierto (T_{op}).

Para cada subtrama la señal objetivo, $x(n)$, y la respuesta al impulso, $h(n)$, se calculan del filtro de síntesis ponderado.

El proceso para encontrar el valor de retardo fraccionario del libro adaptable es como sigue. Una resolución fraccionaria de 1/3 se utiliza en el rango de $[19 \frac{1}{3}, 84 \frac{2}{3}]$ y los enteros son usados en el rango de 85-143. Fue encontrado que esta elección de resolución provee un buen desempeño de razón de bit. El índice del libro adaptable en la primera subtrama se codifica con 8 bits. En la segunda subtrama, una resolución fraccionaria de 1/3 es utilizada en el rango $[T_1 - 52/3, T_1 + 42/3]$ donde T_1 es la parte entera del retardo del libro adaptable en la primera subtrama. Este intervalo se adapta para los casos en que T_1 sobrepasa los límites del intervalo de retardo. El retardo en la segunda subtrama es diferencialmente codificada con 5 bits. Ya que la estimación del tono en ciclo abierto provee un buen seguimiento del tono, la diferencia de codificación en ambas subtramas no introduce degradaciones notables en la calidad de la señal.

1.0- Estimación del retardo del tono en ciclo abierto: usa una señal de voz ponderada $s_\omega(n)$ y se realiza de la siguiente forma: en el primer paso, se establecen tres máximos de la correlación:

$$R(k) = \sum_{n=0}^{79} s_\omega(n)s_\omega(n-k) \quad (13)$$

Para las tres gamas siguientes: 1) $k= 80, \dots, 143$, 2) $k= 40, \dots, 79$, y 3) $k= 20, \dots, 39$. Los valores $n-k < 0$ de las tramas previas son usados. Los máximos retenidos $R(t_i)$, donde t_i corresponde a los valores de las tres gamas máximas $i=1, \dots, 3$, se normalizan mediante:

$$R'(t_i) = \frac{R(t_i)}{\sqrt{\sum_n s_\omega^2(n-t_i)}}, \quad i=1, \dots, 3. \quad (14)$$

El ganador de las tres correlaciones normalizadas se selecciona favoreciendo aquellos retardos que presenten valores en la gama inferior. Esto se logra ponderando las correlaciones normalizadas correspondientes a los retardos más largos. El recurso de dividir la gama de retardos en tres secciones para favorecer los valores menores tiene por objeto evitar la elección de los múltiplos del tono fundamental.

2.0- Cálculo de la señal objetivo: El residual LP está dado por:

$$r(n) = s(n) + \sum_{i=1}^{10} \hat{a}_i s(n-i), \quad n = 0, \dots, 39. \quad (15)$$

La señal objetivo $x(n)$ para la búsqueda del libro adaptable se calcula filtrando la señal $r(n)$ a través de la combinación del filtro de síntesis $1/\hat{A}(z)$ y el filtro ponderado $A(z/\gamma_1)/A(z/\gamma_2)$. Después de determinar la señal de excitación para la subtrama, el estado inicial de estos filtros se actualizan filtrando la diferencia entre las señales residual y de excitación.

3.0- Búsqueda del libro adaptable: Los parámetros del libro adaptable (o parámetros del tono) son índices correspondientes a un retardo y una ganancia. En la aproximación del

libro adaptable para la implementación del filtro del tono [15], la excitación se repite para retardos menores que la longitud de la subtrama. El uso de retardos fraccionarios durante la etapa de búsqueda hace muy caro este proceso. Por tanto, en la fase de reconocimiento, la excitación se amplía mediante la LP residual para simplificar la búsqueda en ciclo cerrado. Este procedimiento es más simple, y se encontró que produce resultados similares comparado con el uso del libro adaptable para subtramas completas. Es importante notar que una vez determinado el retardo, la aproximación convencional del libro adaptable se usa para generar el vector del libro adaptable.

Para cada subtrama de 5 ms, el retardo se determina usando un análisis a ciclo cerrado que minimice el MSE ponderado. En la primera subtrama el retardo T_1 se encuentra buscando en un rango pequeño (6 muestras) de valores alrededor del retardo T_{op} de ciclo abierto. Para la segunda subtrama, la búsqueda del libro adaptable se realiza alrededor del retardo seleccionado en la primera subtrama para encontrar el retardo óptimo T_2 .

El ciclo cerrado busca minimizar el MSE ponderado entre la señal de voz original y la reconstruida. Esto se lleva a cabo maximizando el término:

$$R(k) = \frac{\sum_{n=0}^{39} x(n)y_k(n)}{\sqrt{\sum_{n=0}^{39} y_k(n)y_k(n)}} \quad (16)$$

donde $x(n)$ es la señal objetivo y $y_k(n)$ es la señal de excitación filtrada anterior en el retardo k (excitación anterior convolucionada con $h(n)$, donde $h(n)$ es la respuesta al impulso del filtro de síntesis ponderado $W(z)/\hat{A}(z)$).

Para determinar T_2 y T_1 cuando el retardo en bucle cerrado entero óptimo es inferior a 85, deben probarse las fracciones alrededor del retardo entero óptimo. La búsqueda del tono fraccionario se realiza interpolando la correlación normalizada de la ecuación (16) para encontrar su valor máximo. La interpolación tiene lugar mediante un filtro FIR (finite-duration impulse response) basado en una función sinc con una ventana de Hamming con truncamiento a ± 11 y completado con ceros a ± 12 . El valor de retardo correspondiente a la correlación máxima se selecciona, y el vector del libro adaptable $v(n)$ se calcula interpolando la señal de excitación anterior $u(n)$ en el retardo entero dado k y la fracción t . El filtro tiene una frecuencia de corte (-3dB) a 3600Hz en el dominio de sobremuestra. Fue encontrado que la acción del filtro pasabajas produce un alisamiento en la señal reconstruida.

Después de determinar el retardo del libro adaptable, la ganancia de este libro g_p se calcula como sigue:

$$g_p = \frac{\sum_{n=0}^{39} x(n)y(n)}{\sum_{n=0}^{39} y(n)y(n)}, \text{ limitado entre } 0 \leq g_p \leq 1.2 \quad (17)$$

donde $x(n)$ es la señal objetivo e $y(n)$ es el vector del libro adaptable $v(n)$ filtrado, obtenido mediante la convolución de $v(n)$ con $h(n)$. El vector de libro adaptable filtrado y escalado se sustrae de $x(n)$ para producir una nueva señal objetivo $x'(n)$.

3.3.5 Libro fijo (algebraico)

El libro fijo utiliza un libro algebraico. Cada vector del libro fijo contiene cuatro pulsos diferentes de cero. Estos pulsos pueden asumir amplitudes o posiciones dada la tabla 3, y son codificados separadamente usando un bit de posición dado en esta tabla.

Sea c_k el vector del libro fijo en el índice k . La palabra de código óptimo es la que maximiza el término:

$$\tau_k = \frac{C_k^2}{\epsilon_k} = \frac{(d^t c_k)^2}{c_k^t \phi c_k}, \quad d = x'^t \phi \quad (18)$$

donde d es el vector de correlaciones entre la señal objetivo $x'(n)$ y la respuesta al impulso, $h(n)$, del filtro de síntesis ponderado, y ϕ es la matriz de correlaciones de $h(n)$.

Pulso	Amplitud	Posiciones	Bits
0	± 1	0,5,10,15,20,25,30,35	1+3
1	± 1	1,6,11,21,26,31,36	1+3
2	± 1	2,7,12,17,22,27,32,37	1+3
3	± 1	3,8,13,18,23,28,33,38 4,9,14,19,24,34,39	1+4

Tabla 3. Estructura del libro fijo.

La estructura del libro se usa para un procedimiento de búsqueda rápido ya que el vector del libro fijo c_k contiene solamente cuatro pulsos diferentes de cero cuyas amplitudes son ± 1 . La búsqueda se desarrolla en cuatro ciclos cercanos, donde la contribución de cada ciclo se va sumando al nuevo pulso.

La correlación en el numerador (18) está dada como:

$$C = a_0 d(m_0) + a_1 d(m_1) + a_2 d(m_2) + a_3 d(m_3) \quad (19)$$

donde m_i es la posición del i -ésimo impulso y a_i es su signo. La energía en el denominador en (18) está dada como:

$$\epsilon = \sum_{i=0}^3 \phi(m_i, m_i) + 2 \sum_{i=0}^2 \sum_{j=i+1}^3 a_i a_j \phi(m_i, m_j) \quad (20)$$

Para simplificar el procedimiento de búsqueda, las amplitudes de los pulsos se determinan previamente cuantificando la señal $d(n)$. Para ello, se establece que la

amplitud de un pulso en determinada posición es igual al signo de $d(n)$ en dicha posición. Esta elección de signos para una combinación de pulsos maximiza el término de la correlación en (18). Por eso, antes de llevar a cabo la búsqueda del libro, se deben realizar los siguientes pasos. Se descomponen primeramente la señal $d(n)$ en su valor absoluto $d'(n)=|d(n)|$ y su signo que caracteriza la preselección de las amplitudes del pulso en cada una de las 40 posibles posiciones. Seguidamente, se modifica la matriz ϕ incorporando la información del signo, es decir: $\phi'(i,i)=\text{sign}[s(i)]\text{sign}[s(j)]\phi(i,j)$ y $\phi'(i,i)=0.5\phi(i,i)$. La correlación ahora en (19) tiene la forma:

$$C = d'(m_0) + d'(m_1) + d'(m_2) + d'(m_3) \quad (21)$$

La energía en (20) está dada:

$$\begin{aligned} \varepsilon/2 = & \phi'(m_0, m_0) + \phi'(m_1, m_1) + \phi'(m_0, m_1) + \phi'(m_2, m_2) \\ & + \phi'(m_0, m_2) + \phi'(m_1, m_2) + \phi'(m_3, m_3) + \phi'(m_0, m_3) \\ & + \phi'(m_1, m_3) + \phi'(m_2, m_3) \end{aligned} \quad (22)$$

Para evaluar las posiciones de todos los pulsos, se necesitan examinar un total de $2^{13}=8192$ combinaciones. Para simplificar aún más el procedimiento de búsqueda, se aplica un método de búsqueda focalizada que provee una buena igualación. Debido a que el último ciclo tiene 16 posibles posiciones, se decidió limitar el número de veces que este ciclo se realiza. Se hace una búsqueda exhaustiva de $N=2^9=512$ veces para encontrar la posición de los cuatro pulsos para todas las combinaciones de las tres posiciones de los pulsos. Con el objetivo de maximizar (21) y manteniendo la contribución relativa de los tres pulsos, es posible hacer una decisión (heurística) para no buscar el cuarto pulso.

Esto se realiza a través de un umbral basado en la correlación C . El umbral se hace en función de la correlación absoluta máxima y la correlación promedio debido a la contribución de los tres primeros pulsos dados:

$$C_{max} = \max[d'(t_0)] + \max[d'(t_1)] + \max[d'(t_2)] \quad (23)$$

donde $\max[d'(t_i)]$ son la máxima de $d'(n)$ en las tres primeras posiciones del curso dado en la tabla 3.

La correlación promedio debido a la contribución de los tres primeros pulsos está dada como:

$$C_{av} = \frac{1}{8} \left(\sum_{n=0}^7 d'(5n) + \sum_{n=0}^7 d'(5n+1) + \sum_{n=0}^7 d'(5n+2) \right) \quad (24)$$

La búsqueda del umbral está dada por:

$$C_{th} = C_{av} + (C_{max} - C_{av})\alpha_t \quad (25)$$

El valor de este umbral se presenta antes de empezar la búsqueda del libro. El último ciclo se busca solamente si la correlación absoluta (debida a los tres pulsos) excede este umbral. El valor α_t controla el número de veces que se debe realizar el último ciclo y se encontró que con un valor de 0.4 se hace una búsqueda exhaustiva. La complejidad de búsqueda se determina por el número de palabras de código que se buscan en una trama de 10 ms (dos subtramas). Para limitar el peor caso de complejidad, el valor de N se limita como N_{max} en las dos subtramas. Para la primera subtrama se limita en 105 y para la segunda subtrama se limita a $180-N_1$ (N_1 el valor de N en la primera subtrama).

La periodicidad en la excitación se da solamente por la contribución del libro adaptable. Las voces de alto tono tendrán más de un pulso de tono en una subtrama. En esa situación, la contribución del libro fijo podrá reducirse al valor de periodicidad, conduciendo a degradar la calidad de voz. Este efecto se puede evitar introduciendo periodicidad en la excitación del libro fijo, filtrando el vector del libro fijo a través de un filtro $P(z) = 1/(1-\beta z^{-T})$, donde T es la parte entera del retardo del tono para la trama actual, y β es la ganancia del libro adaptable. Esta modificación de los vectores del libro fijo se integran en la búsqueda filtrando la respuesta al impulso $h(n)$ con $P(z)$ antes de la búsqueda de éste libro.

3.3.6 Cuantificación de las ganancias

Las ganancias de los libros adaptables y fijos son cuantificadas vectorialmente usando 7 bits. Esta cuantificación vectorial, ahorra 2 bits si se compara con una cuantificación escalar. Se encontró que no introduce ninguna degradación notable en la calidad de la voz comparada con las ganancias sin cuantificar.

La búsqueda del libro de ganancia se realiza minimizando el MSE ponderado entre la voz original y la reconstruida:

$$E_{\omega} = \sum_{n=0}^{39} (x(n) - g_p y(n) - g_c z(n))^2 \quad (26)$$

donde $x(n)$ es el vector objetivo, y $y(n)$ y $z(n)$ son los vectores del libro adaptable y fijo, respectivamente.

El procedimiento se hace de la siguiente forma:

1.0-Predicción de la ganancia del libro fijo: Las ganancias del libro fijo en las tramas adyacentes son correlacionadas. Un medio eficiente para explotar esta redundancia es usar un predictor de ganancia logarítmica de energía [16]. Este predictor de ganancia no solamente ayuda a reducir el rango dinámico de la ganancia del libro fijo, sino que

también hace este rango menos dependiente de las variaciones de los niveles de entrada. El uso de un filtro de media móvil (MA) reduce la propagación de los errores en el canal.

La ganancia del libro fijo puede expresarse como:

$$g_c = \gamma \tilde{g}_c \quad (27)$$

donde \tilde{g}_c es una ganancia de predicción basada en las energías del libro fijo previo, y γ es el factor de corrección, que es codificada para su transmisión.

La energía media de la contribución del libro fijo está dada por:

$$E_c = 10 \log \left(\frac{1}{40} \sum_{n=0}^{39} c^2(n) \right) \quad (28)$$

E_c solamente tiene que ser calculada para aquellos vectores modificados por el filtro de realce de periodicidad $P(z)$. Después de escalar el vector $c(n)$ con la ganancia del libro fijo g_c , la energía del libro fijo escalado es $20 \log g_c + E_c$. Sea $E^{(m)}$ la energía media removida (en dB) de la contribución del libro fijo (escalado) en la subtrama m , dada por:

$$E^{(m)} = 20 \log g_c + E_c - \bar{E}_c \quad (29)$$

donde $\bar{E}_c = 30\text{dB}$ es la energía media de la excitación del libro fijo. La ganancia g_c puede ser expresada como una función de $E^{(m)}$, E_c y \bar{E}_c por:

$$g_c = 10^{(E^{(m)} + \bar{E}_c - E_c) / 20} \quad (30)$$

La ganancia predicha \tilde{g}_c es encontrada prediciendo la energía logarítmica de la contribución actual del libro fijo de la energía logarítmica de las contribuciones previas del libro fijo. La predicción MA de cuarto orden se realiza como sigue. La energía predicha está dada por:

$$\tilde{E}_{(m)} = \sum_{i=1}^4 b_i \hat{U}^{(m-i)} \quad (31)$$

donde $[b_1 \ b_2 \ b_3 \ b_4] = [0.68 \ 0.58 \ 0.34 \ 0.19]$ son los coeficientes de predicción MA, y $\hat{U}^{(m)}$ es la versión cuantificada del error de predicción $U^{(m)}$ en la subtrama m , definida por:

$$U^{(m)} = E^{(m)} - \tilde{E}^{(m)} \quad (32)$$

La ganancia predicha \tilde{g}_c se encuentra reemplazando $E(m)$ por su valor predicho en (30):

$$\tilde{g}_c = 10^{(\tilde{E}^{(m)} + E - E_c)/20} \quad (33)$$

El factor de corrección γ se relaciona con el error de predicción de ganancia por:

$$U^{(m)} = E^{(m)} - \tilde{E}^{(m)} = 20 \log(\gamma) \quad (34)$$

2.0- Búsqueda del libro para cuantificación de ganancia: la ganancia del libro adaptable, g_p , y el factor γ son vectores cuantificados usando un libro de estructura conjugada de dos etapas. El término conjugado se refiere al hecho de que cada vector de entrada se cuantifica como una combinación lineal de ambos libros. Tal estructura reduce requerimientos de cálculo y memoria. La primera etapa consiste de un libro F de dos dimensiones de 3 bits y la segunda etapa consiste de un libro G de dos dimensiones de 4 bits.

El primer elemento en cada libro representa una fracción de la ganancia cuantificada del libro adaptable \hat{g}_p , y el segundo elemento representa una fracción del factor de corrección de ganancia cuantificada del libro fijo $\hat{\gamma}$. Dados los índices del libro i_f e i_g para F y G , respectivamente, la ganancia cuantificada del libro adaptable está dada por:

$$\hat{g}_p = F_1(i_f) + G_1(i_g) \quad (35)$$

y la ganancia cuantificada del libro fijo por:

$$\hat{g}_c = \tilde{g}_c \hat{\gamma} = \tilde{g}_c (F_2(i_f) + G_2(i_g)) \quad (36)$$

Esta estructura conjugada simplifica la búsqueda del libro utilizando un proceso de preselección usando las ganancias óptimas del libro adaptable y fijo, resolviendo para $\partial E_\omega / \partial g_p = 0$ y $\partial E_\omega / \partial g_c = 0$ en (26).

El libro F contiene ocho accesos donde el segundo elemento (correspondiente a g_c) tiene en general valores mayores al primer elemento (correspondiente a g_p). Esta preferencia permite una preselección usando el valor de g_c . En este proceso de preselección, se selecciona un grupo de cuatro vectores cuyos segundos elementos son cercanos a g_c . Similarmente, el libro G contiene dieciséis accesos que tienen una preferencia con el primer elemento (correspondiente a g_p). Se selecciona un grupo de ocho vectores cuyos primeros elementos son cercanos a g_p . Por tanto, para cada libro de código, el 50% de los mejores vectores candidatos son seleccionados. Esto se hace por una búsqueda exhaustiva sobre las $4 \times 8 = 32$ posibilidades, tal que la combinación de los dos índices minimicen el MSE ponderado de (26). El proceso de preselección reduce la complejidad de búsqueda sin ningún cambio alguno comparado con la búsqueda

exhaustiva. Los libros de códigos se entrenaron usando un proceso iterativo donde uno de ellos se optimiza y el otro permanece fijo[8].

3.3.7 Decodificación y postprocesamiento

El proceso de decodificación genera los coeficientes del filtro LP del flujo de bits que éste recibe con el mismo proceso que se usó en el codificador. Entonces para cada subtrama los vectores escalonado del libro fijo y adaptable son filtrados con el filtro de síntesis LP para generar la señal de voz reconstruida en el codificador.

El postprocesamiento consiste de un filtro adaptable [17], un filtro pasa altas y aplicación de un escalamiento ascendente a las señales. El filtro adaptable es la cascada de tres filtros: un filtro de término largo $H_p(z)$, un filtro de término corto $H_f(z)$, y un filtro de compensación de pendiente $H_t(z)$, seguido de un proceso de control de ganancia adaptable. Los coeficientes del filtro son actualizados cada 5 ms por subtrama. El proceso de filtrado se realiza como sigue. Primero, la voz reconstruida $\hat{s}(n)$ se filtra a través de $\hat{A}(z/\gamma_n)$ para producir una señal residual $\hat{r}(n)$. Esta señal se usa para calcular el retardo T y la ganancia g_l del filtro de término largo $H_p(z)$. La señal $\hat{r}(n)$ entonces se filtra a través de un filtro de término largo $H_p(z)$ y del filtro de síntesis $1/[g_f \hat{A}(z/\gamma_d)]$. Finalmente, la señal de salida del filtro de síntesis $1/[g_f \hat{A}(z/\gamma_d)]$ se pasa por el filtro de compensación de inclinación $T_t(z)$ para generar la señal de voz reconstruida filtrada $f(n)$. Luego, se aplica un control de ganancia adaptable para $f(n)$ para igualar la energía de $\hat{s}(n)$. La señal resultante $f'(n)$ se filtra con un filtro pasa altos de 100Hz y se multiplica por dos para producir la señal de salida del decodificador. A continuación se describen los cuatro procesos que intervienen en esta etapa.

1.0-Filtro de término largo: la función del sistema de este filtro está dada por la siguiente ecuación:

$$H_p(z) = \frac{1}{1 + 0.5g_l z^{-T}} \quad (37)$$

donde T es el retardo del tono, y g_l es el coeficiente de ganancia. El limite superior que puede tomar el valor g_l es de 1 (no negativo), y se fija en cero si no muestra una fuerte periodicidad (siendo la ganancia de predicción de término largo inferior a 3dB). El factor de 0.5 controla la cantidad de filtrado del término largo. El retardo y la ganancia del término largo se calculan de la señal residual $\hat{r}(n)$ obtenido de filtrar la señal de voz $\hat{s}(n)$ por $\hat{A}(z/\gamma_n)$, que es el numerador del filtro de término corto.

El retardo del término largo se calcula en un proceso que consiste de dos pasos. En el primer paso se selecciona el mejor entero T_0 en el rango $[\text{int}(T_1)-1, \text{int}(T_1)+1]$, donde $\text{int}(T_1)$ es la parte entera del retardo del tono (transmitido) en la primera subtrama. El uso del retardo de la primera subtrama se usa porque está cerca del retardo del tono

que fue estimado en el ciclo abierto durante la codificación. El mejor retardo entero es el que maximiza la correlación:

$$R(k) = \sum_{n=0}^{39} \hat{r}_0(n) \hat{r}_k(n) \quad (38)$$

donde $\hat{r}_k(n)$ es la señal residual (fraccionaria) del retardo k . En el segundo paso se escoge el mejor retardo fraccionario T con una resolución de $1/8$ alrededor de T_0 . Esto se realiza encontrando el retardo con la más alta correlación pseudonormalizada:

$$R'(k) = \frac{\sum_{n=0}^{39} \hat{r}_0(n) \hat{r}_k(n)}{\sqrt{\sum_{n=0}^{39} \hat{r}_k(n) \hat{r}_k(n)}} \quad (39)$$

Una vez que se encuentra el retardo óptimo T , la correlación correspondiente $R'(T)$ se normaliza con la raíz cuadrada de la energía de $\hat{r}(n)$. El valor cuadrado de esta correlación normalizada se usa para determinar si el filtro de término largo es deshabilitado. Esto se realiza con $g_l=0$ si:

$$\frac{R'(T)^2}{\sum_{n=0}^{39} \hat{r}_0(n) \hat{r}_0(n)} < 0.5 \quad (40)$$

En cualquier otro caso el valor g_l se calcula de:

$$g_l = \frac{\sum_{n=0}^{39} \hat{r}_0(n) \hat{r}_k(n)}{\sum_{n=0}^{39} \hat{r}_k(n) \hat{r}_k(n)}, \text{ limitado por } 0 \leq g_l \leq 1 \quad (41)$$

Se observó que recalculando el retardo del tono con la resolución más alta mejora el desempeño sobre el uso del retardo transmitido. Deshabilitando el filtro de término largo se notó que es provechoso para reducir los efectos que se producen por medios ruidosos [8].

2.0- Filtro de término corto: la función del sistema de éste filtro está dada por la siguiente ecuación:

$$H_f(z) = \frac{1}{g_f} \frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)} = \frac{1}{g_f} \frac{1 + \sum_{i=1}^{10} \gamma_n^i \hat{a}_i z^{-i}}{1 + \sum_{i=1}^{10} \gamma_d^i \hat{a}_i z^{-i}} \quad (42)$$

$$g^{(n)} = 0.85g^{(n-1)} + 0.15G \quad n = 0, \dots, 39 \quad (47)$$

Se utiliza el valor inicial de $g^{(-1)}=1.0$. Seguidamente, para cada nueva subtrama, $g^{(-1)}$ se iguala a $g^{(39)}$ de la subtrama anterior.

Al terminar de estudiar la norma G.729 con el fin de tener un conocimiento básico de su funcionamiento, es decir, saber las partes que lo integran y qué métodos utiliza; procedimos a verificar su funcionamiento en la PC y posteriormente, hacer el diseño de la tarjeta.

3.4 Prueba en la PC

El algoritmo G.729 fue obtenido en ANSI C en punto fijo (implementado por la ITU). El algoritmo consiste de aproximadamente 26 archivos que se integran en dos partes. Una parte constituye el codificador que se utiliza para comprimir voz y otra parte, la constituye el decodificador que se necesita para descomprimir la voz.

Para llevar a cabo la prueba de este algoritmo en la PC, se agruparon los programas para codificar y decodificador archivos de voz con los nombres de coderb y decoderb, respectivamente, en Borland C++. Se ejecutaron bajo DOS.

La prueba del codificador se lleva a cabo como sigue:

coderb archivoentrada flujobits

La prueba del decodificador se lleva a cabo como sigue:

decoderb flujobits archivosalida

Los archivos de entrada y de salida son archivos de datos de muestreo (cada muestra representada en 16 bits). El flujo de bits es un archivo que se genera, el cual contiene los parámetros necesarios para la compresión de la voz y que también es utilizado en el decodificador para descomprimir la voz y obtener el archivo original de salida.

Los sub programas específicos que integran el codificador son los siguientes:

Nombre del archivo	Descripción
acelp_co.c	Buscar tabla de códigos fijos
cod_ld8k.c	Sub programa del codificador
lpc.c	Análisis LP
pitch.c	Búsqueda del tono
pre_proc.c	Preprocesamiento (filtro paso alto y escalamiento)
pwf.c	Cálculo de coeficientes de ponderación perceptual
qua_gain.c	Cuantificador de ganancia
qua_lsp.c	Cuantificador LSP

Los sub programas específicos que integran el decodificador son los siguientes:

Nombre del archivo	Descripción
de_acelp.c	Decodificador tabla de códigos algebraicos
dec_gain.c	Decodificador de ganancias
dec_lag3.c	Decodificar el índice del libro adaptable
dec_ld8k.c	Sub programa del decodificador
lspdec.c	Sub programa de decodificación LSP
post_pro.c	Postprocesamiento (filtro paso alto y escaleamiento)
pst.c	Sub programa de postfiltro

Los sub programas generales comunes para el codificador y decodificador (no se mencionan los archivos de definición de tabla y de variables, los que tienen extensión h):

Nombre del archivo	Descripción
basic_op.c	Operadores básicos
oper_32b.c	Operadores básicos ampliados
bits.c	Sub programas de manipulación de bits
Dspfunc.c	Funciones matemáticas
filter.c	Funciones de filtro
Gainpred.c	Predictor de ganancia
Lpcfunc.c	Sub programas varios relativos al filtro LP
lspgetq.c	Cuantificador LSP
p_parity.c	Cálculo de paridad de tono
pred_lt3.c	Generación del libro adaptable
util.c	Funciones de utilidad

Se utilizaron unos vectores de prueba dados por la ITU para saber si este algoritmo estaba funcionando correctamente y aunque no cubren todo el algoritmo exhaustivamente sirven para verificar la correcta compilación y ejecución del algoritmo en ANSI C.

Los archivos que se introdujeron para probar la funcionalidad del algoritmo fueron comparados con los que la ITU proporcionaba. Dicha comparación se hizo mediante un programa en ANSI C que verificaba el contenido de los archivos. Estos archivos resultaron ser iguales. También se utilizaron algunos archivos de voz en español. El hecho de usar archivos de voz en diferente idioma fue para saber si el algoritmo originaba mejor calidad en alguno de estos. Estos archivos se escuchaban antes de hacer la compresión y después de hacer la descompresión, verificando que ambos se entendieran. A continuación se presentan algunos de estos archivos:

- alghm: señal artificial para verificar ciertas partes del código.
- fixed: señal artificial para verificar la búsqueda del libro.
- lsp: señal artificial para verificar la cuantificación lsp.
- pitch: señal artificial para verificar el predictor de término largo.
- tame: onda senoidal para verificar el proceso de temporización.

overflow: detección de desbordamiento en el sintetizador.
parity: chequeo de paridad.
speech: señal de voz.

En el punto 3.6 se presentan dos diagramas espectrales mostrando el archivo de voz de entrada sin comprimir y el archivo de voz de salida descomprimido.

3.5 Diseño de la tarjeta usando el algoritmo CS-ACELP

Antes de hacer el diseño de la tarjeta se estudiaron algunos DSP de la tercera, quinta y sexta generación para determinar en cual de ellos sería factible hacer su implementación. Se consideró su costo, las instrucciones por segundo que podía ejecutar, que la comunicación fuera full-duplex, el hecho de contar con sus compiladores y posibles modificaciones de diseño en el futuro. Se seleccionó el DSP de la sexta generación.

Una vez que el algoritmo fue probado en la PC, fue necesario compilarlo y usar la instrucción link del DSP de la sexta generación antes de elaborar la tarjeta. El hecho de compilar nos proporciona como salida un archivo que contiene el algoritmo en lenguaje ensamblador y un archivo que es utilizado para hacer la verificación del funcionamiento del algoritmo en el simulador del DSP.

El algoritmo fue compilado por separado y se utilizó el siguiente comando: cl6x -k coder. Una vez compilado, se procedió con el siguiente comando: lnk6x coder -m coder.map -o coder.out -l c:\c6xtools\lib\rts6201.lib.

Es importante mencionar que lo único que se verificó con esto es que el algoritmo se ejecutara en el DSP pero sin verificar que la ejecución fuera correcta.

Se decidió utilizar el TMS320C6201 de punto fijo, su máxima velocidad de operación es de 1600MIPS, lo que facilita poder manejar varios usuarios a la vez.

En la elaboración de la tarjeta fue necesario considerar los siguientes puntos:

- El tiempo de ejecución del algoritmo, el cual fue estimado calculando el número de operaciones requeridas para la ejecución de la compresión y descompresión de 10ms de voz (13000 ciclos, 5000 llamadas, 1500 multiplicaciones, 2500 sumas, 100 divisiones, etc. obteniendo un total de 45500 operaciones). La frecuencia que se utiliza en el DSP es de 200MHz por lo que el tiempo de ejecución estimado es de 500 microsegundos. Con este resultado podemos concluir que esta tarjeta tiene la capacidad de dar servicio hasta 20 usuarios en 10ms.
- El espacio de memoria requerido para almacenar el algoritmo, que se calculó contando el número de líneas de código obtenido después de la compilación (45000 líneas) del algoritmo en el DSP correspondiente. Como se consideraron 2 bytes por línea se obtienen 88Kbytes requeridos para almacenar el algoritmo. De acuerdo a

este resultado se decidió usar una memoria de 128Kbytes considerando 40Kbytes para implementar otros programas requeridos.

- El bus ISA porque es común encontrarlo en las computadoras personales. Se usa una memoria de puerto dual como almacenamiento temporal. A través de esta memoria la PC y el DSP pueden intercambiar datos independientemente. El dimensionamiento de esta memoria se realizó considerando el número de bytes de voz comprimida (10bytes) y de voz a comprimir (160 bytes) y el tiempo requerido para la ejecución del proceso de compresión y descompresión. Así, 1700 bytes son necesarios para mantener 10 usuarios. Sin embargo, considerando la disposición de dispositivos y expansiones futuras, nosotros decidimos usar una memoria de 16Kbytes.

A continuación se muestra un diagrama a bloques que ilustra cómo está constituida la tarjeta:

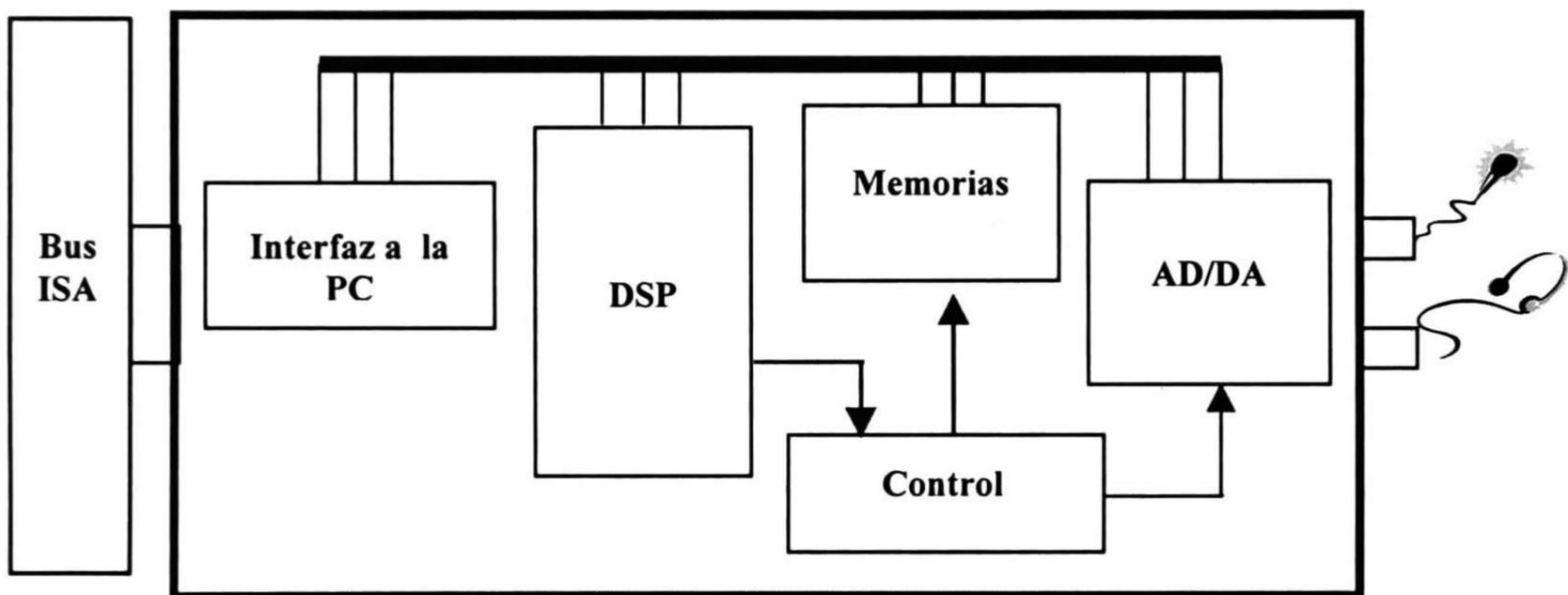


Figura 9. Diagrama a bloques del diseño de la tarjeta

En los apéndices A y B se pueden ver los detalles del diseño de la tarjeta y los diagramas esquemáticos, respectivamente. En los otros apéndices, se puede verificar la lista de materiales utilizados, la manera en que la tarjeta quedó diseñada y finalmente, los archivos de fabricación que se obtuvieron después de la elaboración del PCB.

La manera como debe funcionar la tarjeta de acuerdo al diseño es como sigue (ver figura 10):

- a) Voz a comprimir: la voz de entrada recibida del micrófono se debe filtrar y muestrear, después debe ser almacenada (buffer de entrada) hasta que se hayan completado 80 muestras correspondientes a 10 ms. Después de que las muestras sean comprimidas (decodificador) deben ser almacenadas temporalmente en una memoria de puerto dual (buffer del DSP) y después deben ser enviadas al DSP. El DSP debe interrumpir a la PC, para que la PC tome las 80 muestras de trama y las almacene en un buffer (buffer de la PC). Después de eso, la PC debe construir un flujo de bits con la información recibida para enviarla por la red (paquete transmitido).

- b) **Voz a descomprimir:** en el otro lado, se tendrá una tarjeta igual que hará la función inversa de lo que se hace en el proceso de compresión. El flujo de bits llega y la voz comprimida (trama de voz) es recuperada; (paquete recibido) más tarde es almacenada (buffer de la PC) antes de ser enviada al DSP. También la PC debe interrumpir al DSP para avisarle que llegó información. Cuando el DSP sabe que tiene información para descomprimir, toma los paquetes de voz y los almacena (buffer para el DSP) para posteriormente descomprimirlos. Después que la voz sea descomprimida debe almacenarse en otro buffer (buffer de salida) para ser filtrada y amplificada antes de ser escuchada por la bocina.

El diagrama que ilustra esta explicación aparece a continuación:

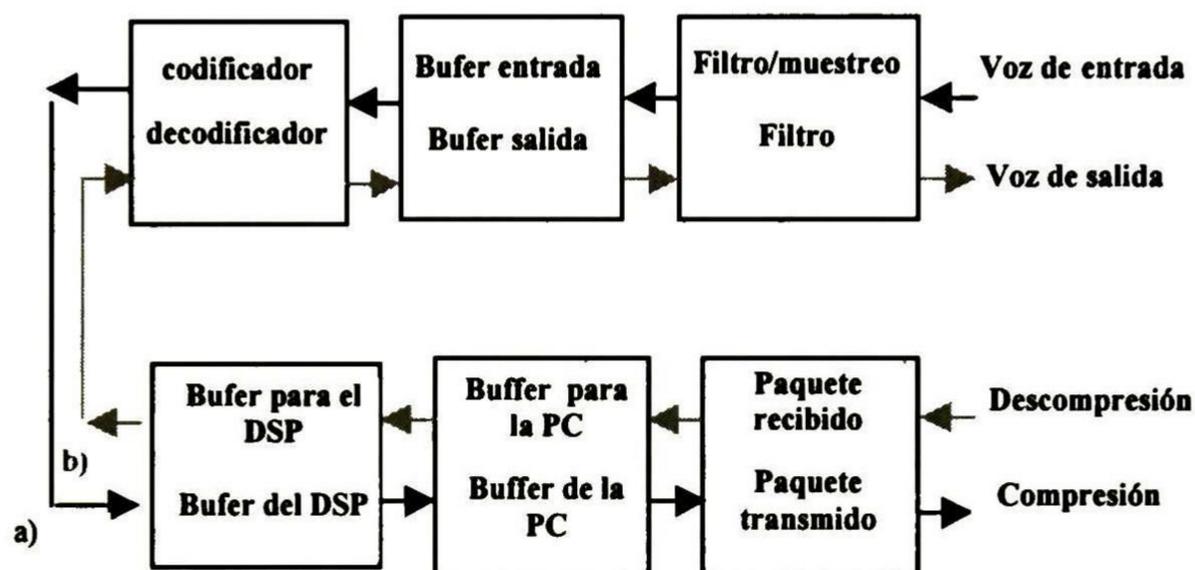


Figura 10.a) Voz comprimida. b) Voz descomprimida.

3.6 Resultados

De las pruebas hechas en la PC se verificó el buen funcionamiento del algoritmo. Se introdujeron varios archivos de voz para comprimir; una vez que el algoritmo de compresión generó un archivo de información comprimida éste se utilizó para descomprimirlo. El tiempo de compresión fue de 10ms. Se escucharon ambos archivos de voz (los originales a comprimir y los descomprimidos) resultando iguales al oído.

Las personas que trabajaban con otros codificadores de voz y los asesores de la tesis, fueron las que escucharon estos archivos.

También se comparó el contenido de estos archivos mediante un programa en C; con el cual se comparó el archivo de voz original contra el reconstruido por el codificador G.729. Los archivos resultaron ser muy diferentes desde el punto de vista numérico. Sin embargo, el oído humano no percibe esta diferencia. Esto se concluye después de verificar las opiniones del grupo de personas que escucharon los archivos. Por otro lado, cuando la comparación se lleva a cabo desde el punto de vista espectral nos damos cuenta que los archivos son muy parecidos en vista de que la distribución de

energía es similar. Esto se puede notar en los espectrogramas que se presentan en la Figura 11 (señal original) y en la Figura 12 (la señal reconstruida).

Respecto al funcionamiento de la tarjeta, lo único que se pudo verificar fue la parte del DSP que va conectada al convertidor TLC32040; se simuló en Maxplus II para saber si se podían leer y escribir los datos del DSP sin problemas de tiempo.

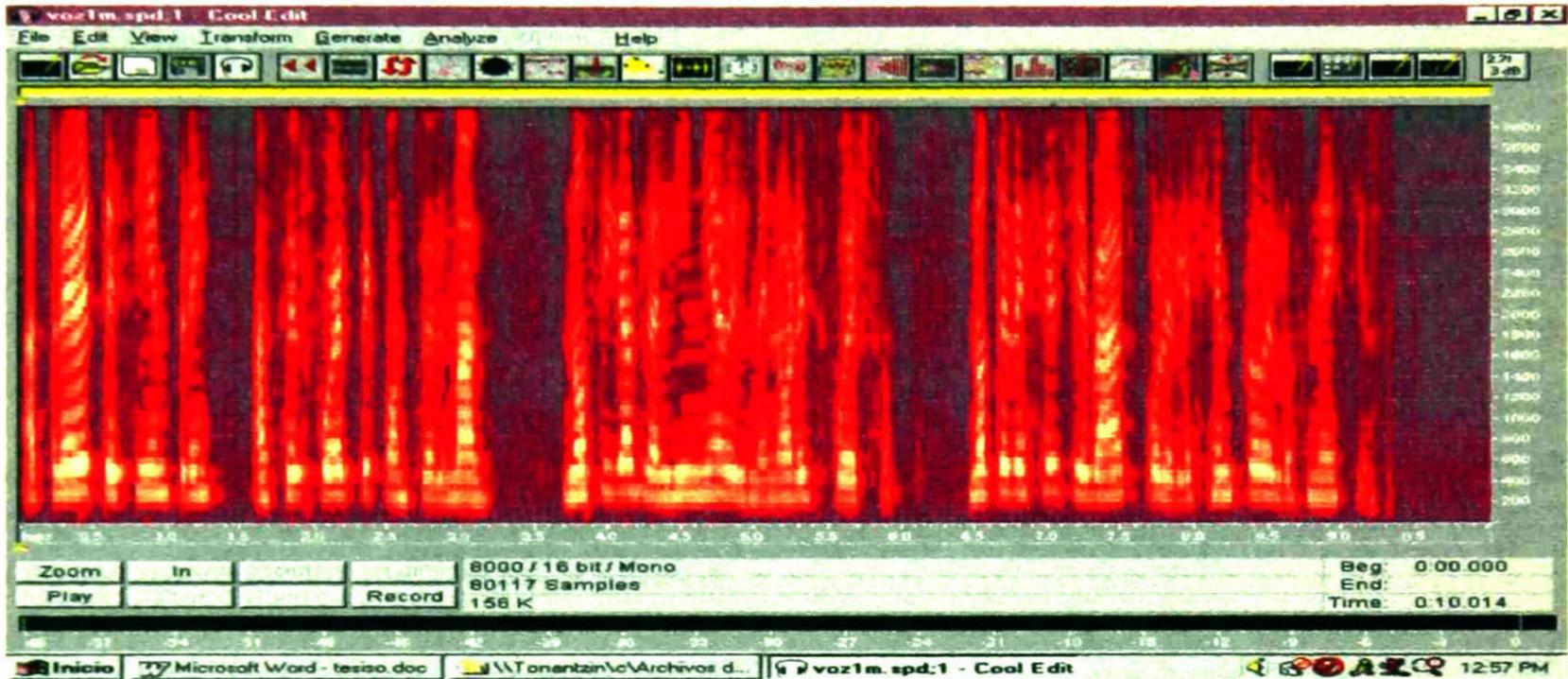


Figura 11. Espectrograma de la señal original

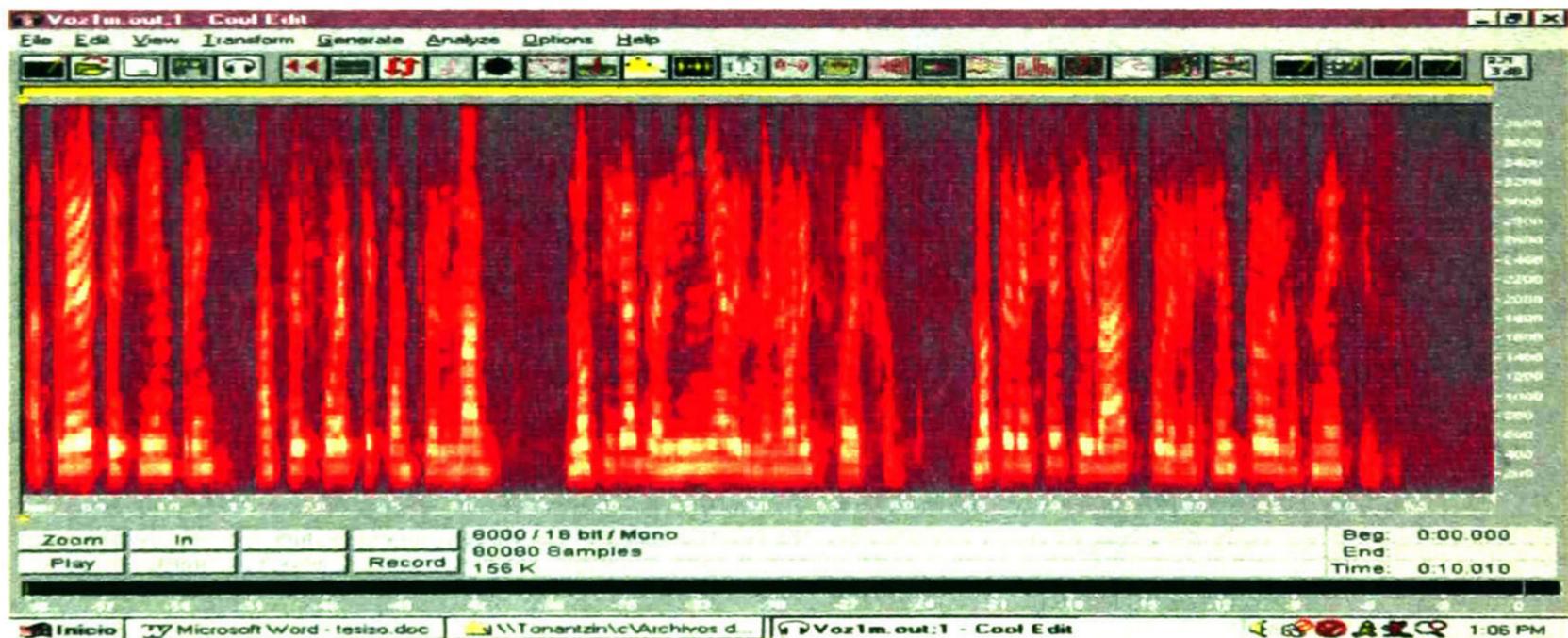


Figura 12. Espectrograma de la señal reconstruida

Según las verificaciones que se hicieron, esta parte del diseño funciona bien. Las demás partes que constituyen la tarjeta no se verificaron porque debía enviarse a fabricar para poder hacer las pruebas debido a que el DSP no se encuentra en ningún simulador.

Capítulo 4

Conclusiones

El algoritmo CS-ACELP fue obtenido de la ITU en ANSI C (un conjunto de varios archivos). El algoritmo se agrupa en dos partes: una para codificar y otra para decodificar; estas partes están constituidas por una serie de archivos que fueron ordenados, compilados y simulados en la PC para verificar su funcionamiento.

Aunque no se programó el algoritmo es bueno mencionar que el hecho de estudiarlo ayuda mucho para verificar su funcionamiento y poder entender a grandes rasgos cómo está integrado el programa y saber si es factible o no hacer algunos cambios que ayuden a su simulación en el DSP.

Se buscó información sobre los DSP de la tercera, quinta y sexta generación para analizar en cuál de ellos se podría hacer la implementación. Se consideró el costo, las instrucciones por segundo que podía ejecutar, posibles modificaciones del diseño en el futuro y el hecho de contar o no con sus compiladores. El DSP TMS320C6201 es el que se seleccionó para el diseño.

Antes de hacer el diseño de la tarjeta, fue necesario considerar cuánto espacio ocuparía el código y qué tiempo de ejecución tomaría para llevarse a cabo. Es por eso, que el algoritmo se tuvo que compilar y simular en el DSP.

Por otro lado, también fue necesario estudiar el DSP que se utilizaría para saber las partes que se debían considerar en el diseño de la tarjeta y poder especificar los requisitos de diseño. En este punto es bueno mencionar que aunque se establecen ciertos requisitos al inicio del proyecto estos pueden cambiar en el desarrollo del mismo por diversos factores.

Una vez terminado el diseño y compilado el algoritmo, se procedió a adquirir los dispositivos que se iban a utilizar (memorias, osciladores, capacitores, etc.); se contactaron diversos distribuidores, nacionales e internacionales. En este punto es bueno mencionar que la selección de dispositivos algunas veces absorbe tiempo porque las cotizaciones se atrasan y algunos de ellos resultan no estar disponibles. Además, los dispositivos no siempre cumplen con todas las características requeridas para el diseño y se debe hacer una ponderación entre los posibles candidatos.

La parte del diseño que conecta al DSP con el convertidor TLC32040 se tomó de las hojas de especificaciones del convertidor. Se simuló en MAXPLUS II para verificar

su funcionamiento. Se notó que el diseño tenía algunos problemas para poder escribir datos del DSP al convertidor, así que se añadió una compuerta adicional para evitar este problema. Puede ser poco notorio pero es importante verificar las conexiones que sugieren los distribuidores para asegurar su funcionamiento.

Posteriormente, se procedió a la elaboración del PCB (Printed Circuit Board) usando Menthor Graphics. En la elaboración del PCB se cubrieron los siguientes puntos: hacer los esquemáticos y símbolos requeridos, hacer las geometrías de los dispositivos, hacer un mapa donde cada símbolo se identificara con su geometría (considerando la cantidad requerida), identificar el número de capas y grosores de líneas que tendría la tarjeta, ordenar los componentes en la tarjeta ayudado por un diagrama de distribución para poder rutear adecuadamente la tarjeta y finalmente, elaborar los archivos necesarios para su fabricación.

Se hizo un calendario para realizar todas las actividades antes mencionadas pero no se lograron terminar en el tiempo previsto. Una de las razones es que no se tenía conocimiento de las herramientas a utilizar y se fue muy optimista en la disponibilidad de aprendizaje. Tampoco se consideraron algunos factores externos como por ejemplo, disponibilidad de licencias y lentitud en la red. Otro punto importante es la falta de experiencia en la coordinación de proyectos.

La tarjeta se diseñó y se dejó lista con los archivos de fabricación para que se mande a fabricar. Sin embargo, existe todavía mucho trabajo que hacer para que se pueda construir el sistema que se mencionó al principio de este documento: tener una PC con una sola tarjeta de compresión y descompresión, que realice estas funciones entre una red local (donde fluya voz descomprimida) y una red global (donde fluya voz comprimida) para que varios usuarios puedan usarla en una red IP o una red FR aprovechando el ancho de banda del sistema.

Dentro de los puntos que faltan por hacer se encuentran los siguientes:

- Simular el algoritmo CS-ACELP en el DSP antes de integrarlo a la tarjeta.
- Modificar la tarjeta para que pueda ser insertada a una PC dedicada (sin convertidores A/D-D/A, sin micrófono y bocina).
- Fabricar la tarjeta diseñada, ensamblarla y verificarla.
- Hacer un programa en la PC para conectar la tarjeta (inicialización, interrupciones, etc).
- Cambiar de bus ISA a bus PCI para aprovechar al máximo el tiempo de transferencia de paquetes de la PC a la tarjeta y viceversa.
- Hacer un programa que se encargue de formar los paquetes de voz que van a enviarse a la red y viceversa.
- Verificar el tiempo de ejecución del algoritmo considerando los retardos que se generen en la red.

El diseño de la tarjeta se presentó para el uso del codificador CS-ACELP pero puede ser usada para cualquier otro tipo de codificador. Se concluye esto porque el

algoritmo se carga en una memoria externa al DSP y aunque se consideró el tiempo de procesamiento de trama para 10 usuarios, puede ser modificado. La tarjeta también puede ser usada como un multicodificador; esto es, como un dispositivo que puede contener varios tipos de codificadores y dependiendo del sistema que se use, seleccionar alguno en particular. Es importante aclarar que para llevar a cabo lo anterior, se tendría que cambiar las dimensiones de las memorias que se consideraron en el diseño.

También se pueden estudiar los libros de código para analizar si es posible reducirlos de acuerdo a un idioma seleccionado.

APÉNDICE A

Especificación Técnica

Tarjeta Tabasco / Codificador de voz CS-ACELP

1. Introducción

Este documento describe la tarjeta electrónica llamada Tabasco. Tabasco es una tarjeta basada en el DSP TMS230C6201 de TI. Su función primaria es la de comprimir y descomprimir las tramas de voz bajo el estándar G.729. Tabasco usa el formato del bus ISA y está diseñada para ser insertada en un "slot" de una PC. Para llevar a cabo la puesta en marcha y corrección de errores de diseño de la tarjeta, se agregaron un convertidor A/D y D/A, un micrófono y una bocina. Estos últimos elementos nos permitirán probar la tarjeta de manera local y verificar que el algoritmo CS-ACELP ha sido implementado en forma correcta.

El propósito final de la tarjeta es el de comprimir y descomprimir la información que le sea enviada a través de la red de datos. La tarjeta ha sido dimensionada para manejar correctamente 10 llamadas telefónicas simultáneas.

Originalmente, el codificador fue diseñado para operar voz digitalizada a 128Kpbs es decir, muestreada a 8Khz considerando 16 bits por muestra. Sin embargo, nuestra implementación usa 14 bits a 8Khz debido a que el convertidor elegido sólo maneja esos números de bits pero aún así, el algoritmo funciona considerando 16 bits por muestras, donde los dos bits menos significativos en la trama se toman como cero.

2. Propósito del documento

Este documento va dirigido a ingenieros de hardware, de software y técnicos encargados de diseñar y mantener este sistema.

3. Documentación de referencia

3.1. Estándares

Recomendación G729

Recomendación G729 anexo B

3.2. Hojas de datos

www.insight-electronics.com

www.itu.com

www.ti.com

TMS320 Floating-Point DSP Assembly Language Tools. User's Guide 1997

TMS320 Floating-Point DSP Code Generation Tools. Getting Started 1997

TMS320 Floating-Point DSP Optimizing C compiler. User's Guide 1997

TMS320C6X Software Tools: getting started guide. Texas Instruments 1997.

TT L Devices

Manual del FPGA

4. Aplicaciones

La figura 1 muestra un ejemplo de una comunicación llevada a cabo con este codificador.

El sistema consta de dos computadoras conectadas a una red para compartir información, dos tarjetas (basadas en DSP) para comprimir/descomprimir la señal de voz, un micrófono y una bocina para que el usuario se comuniquen.

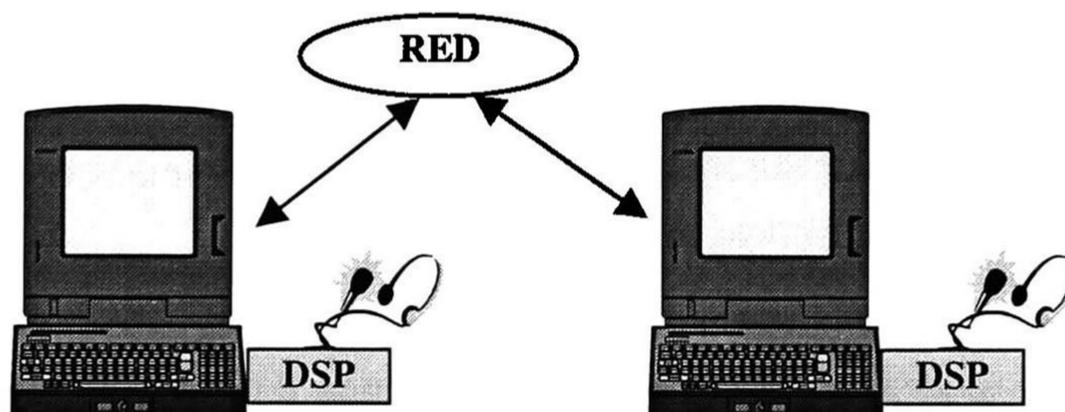


Figure 1. Aplicación típica.

La Figura 2 muestra otro ejemplo de la aplicación que tiene esta tarjeta. A diferencia de la primera aplicación, en ésta sólo se usaría una tarjeta en la computadora conectada a la red externa global. En esta configuración, se hace uso de las tarjetas multimedia para generar las comunicaciones y señales de voz.

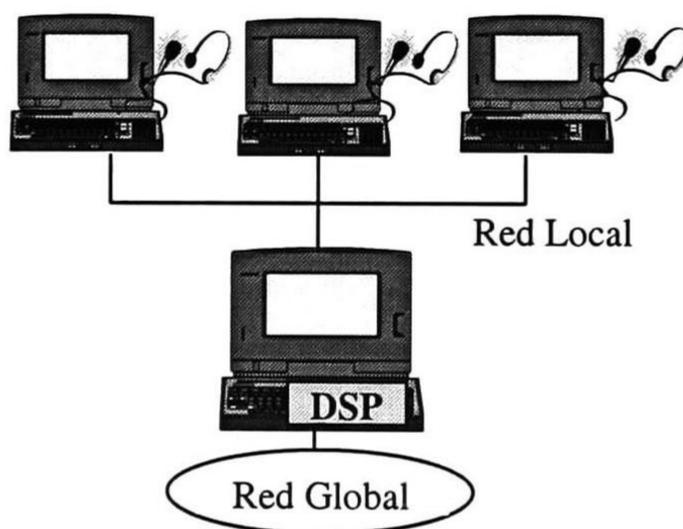


Figura 2. Diagrama con un sistema de compuerta.

La computadora con esta tarjeta estaría dedicada a controlar los paquetes de voz que reciba de ambas redes. La compresión la haría hacia la red global y la descompresión hacia la red local.

5. Funciones básicas de la tarjeta Tabasco.

Función de la tarjeta de compresión/descompresión (Tabasco):

- Compresión

Tarjeta Tabasco / Codificador de voz CS-ACELP

- Filtrar a 4KHz la señal analógica recibida.
 - Hacer un muestreo a 8KHz en 14 bits.
 - Comprimir la señal de voz a 8Kbps.
 - Interfazar con el Bus ISA.
-
- Descompresión
 - Recibir la señal comprimida a través del Bus ISA.
 - Descomprimir la señal de voz.
 - Convertir la información descomprimida a una señal analógica.
 - Filtrar la señal recuperada y amplificarla para su audición en una bocina.

Función de la PC:

- Interface al usuario.
- Conexión a la red de datos.

6. Diseño Físico

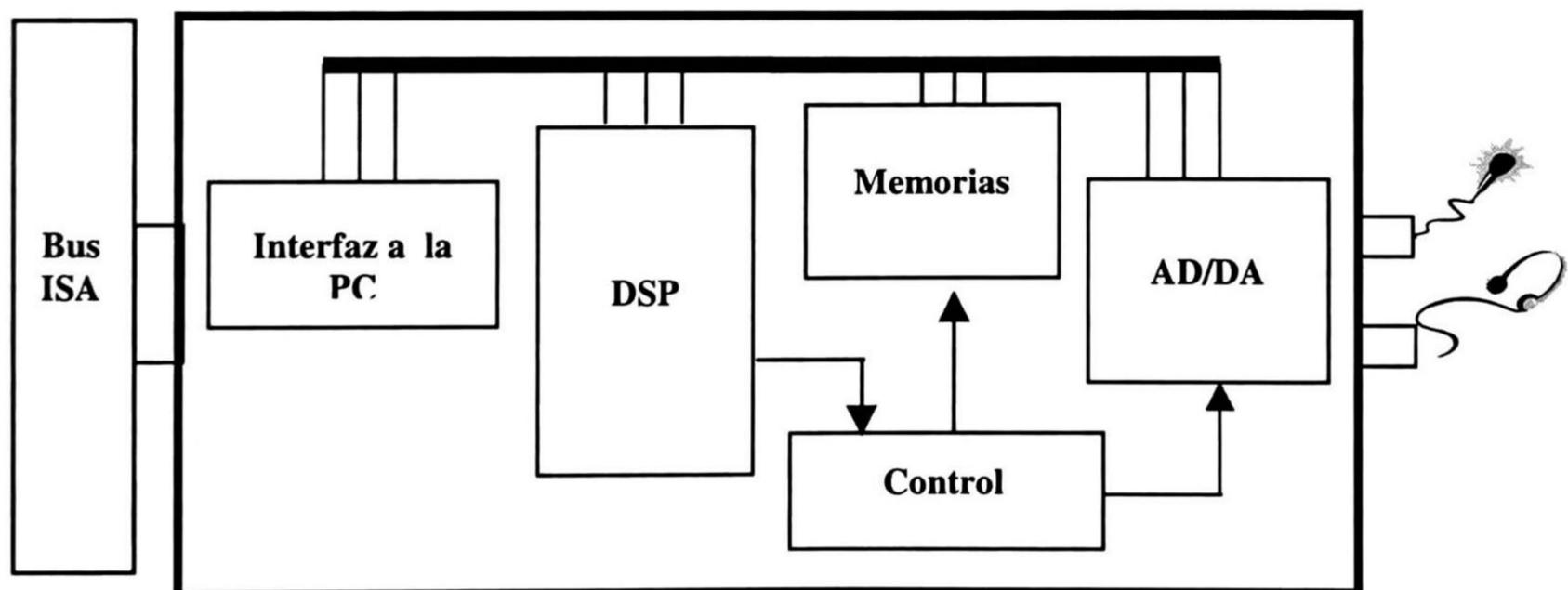
La tarjeta será diseñada en formato AT Bus ISA.

7. Puertos Físicos

La tarjeta tendrá dos conectores tipo Jack 3.5 para conectar un micrófono y una bocina.

8. Arquitectura de la tarjeta

El siguiente esquema muestra el diagrama a bloques de la tarjeta Tabasco:



Tarjeta Tabasco / Codificador de voz CS-ACELP

8.1 Función de cada bloque

A/D-D/A

La función A/D, filtra y convierte en forma digital la señal analógica recibida por el micrófono. La función D/A convierte los datos recibidos de la descompresión a una señal analógica filtrada.

Memorias

Dos tipos de memoria componen este bloque: memoria EPROM para almacenar el programa del DSP y memoria SRAM para almacenamiento temporal de datos. También se utiliza una memoria de puerto dual en la interfaz al bus ISA.

Control

Encargado de decodificar el acceso a memoria de los diferentes dispositivos (RAM, EPROM, Convertidor, Memorial Dual) instalados en la tarjeta.

DSP

Ejecuta la tarea de compresión/descompresión. También controla los dispositivos instalados en la tarjeta.

Interfaz al bus ISA

Controla las entradas y salidas de información de la tarjeta **Tabasco** hacia el procesador central en la PC.

Nota: el dispositivo de control y la interfaz al bus ISA están considerados para ser diseñados en un mismo dispositivo programable.

8.2 Descripción de cada bloque

8.2.1 A/D-D/A

Este dispositivo es un TLC32040 de Texas Instruments que contiene un ADC y un DAC en un sólo chip.

Características:

La resolución del ADC y DAC es de 14 bits.

Velocidad de muestreo variable hasta 19200 muestras por segundo.

Velocidad de conversión programable asíncrona o síncrona.

Interface para puerto serial para el SN74299 (en caso de requerir una interfaz paralela con otro dispositivo).

Filtro pasabanda de entrada para evitar aliasing.

Filtro pasabaja de salida para reconstrucción de la señal.

MSTR_CLK debe ser de 95ns como mínimo para que funcione el convertidor.

Tarjeta Tabasco / Codificador de voz CS-ACELP

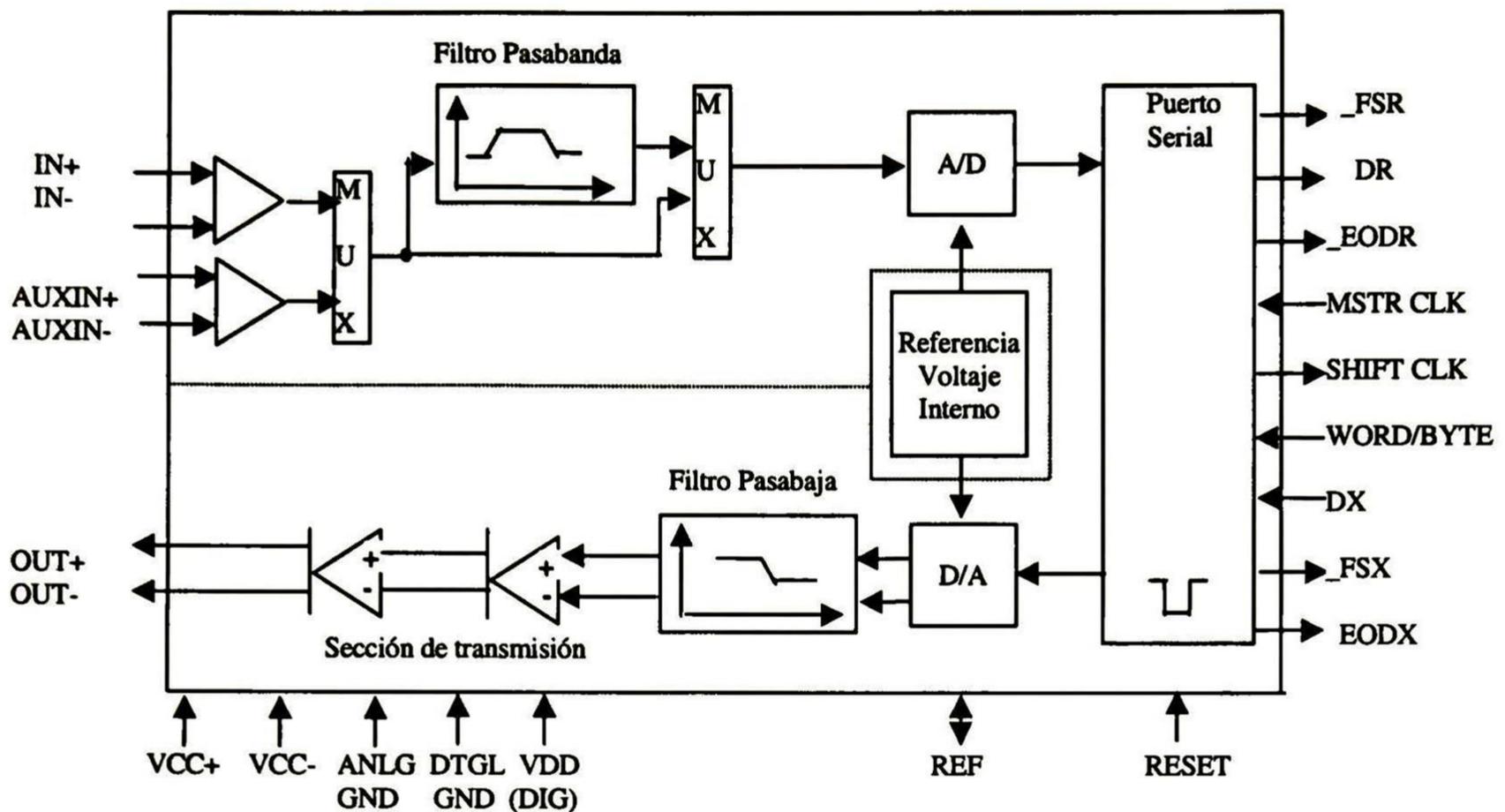


Figura 1. Diagrama a bloques del convertidor TLC32040

Los pines considerados para este diseño (incluyendo las alimentaciones de voltajes analógicos y digitales así como las referencias de tierra) son los siguientes:

DX es usado para recibir los bits de entrada del DAC

DR es usado para transmitir los bits de salida del ADC

IN+ entrada no invertida a la entrada del amplificador analógico del TLC320

¹*MSTR_CLK* es usado para derivar todas las señales del TLC320

_RESET es usado para inicializar registros y controlarlos.

_EODX indica el fin de transmisión de datos.

FSX indica la transmisión sincrona de trama

OUT+ salida no invertida del amplificador analógico

SHIFT_CLK usado para el reloj de la transferencia serial de datos del TLC320.

BYTE es conectado a 5V para hacer la comunicación serial con el DSP.

Entre las señales de *VCC_* y *ANLG* se conecta un diodo de .4V por requerimientos del dispositivo. Para mayor información ver la hoja de datos.

La iniciación de este dispositivo se hace por software, de esta manera se selecciona el modo de operación del convertidor (asíncrono) y sus frecuencias de muestreo (8khz) y de corte (4khz) para este diseño.

8.2.1.1 Conexión con el DSP

La manera en que se encuentra conectado este dispositivo con el DSP se muestra a continuación:

¹ El símbolo “_” al inicio de una señal indica que la señal es activa en bajo.

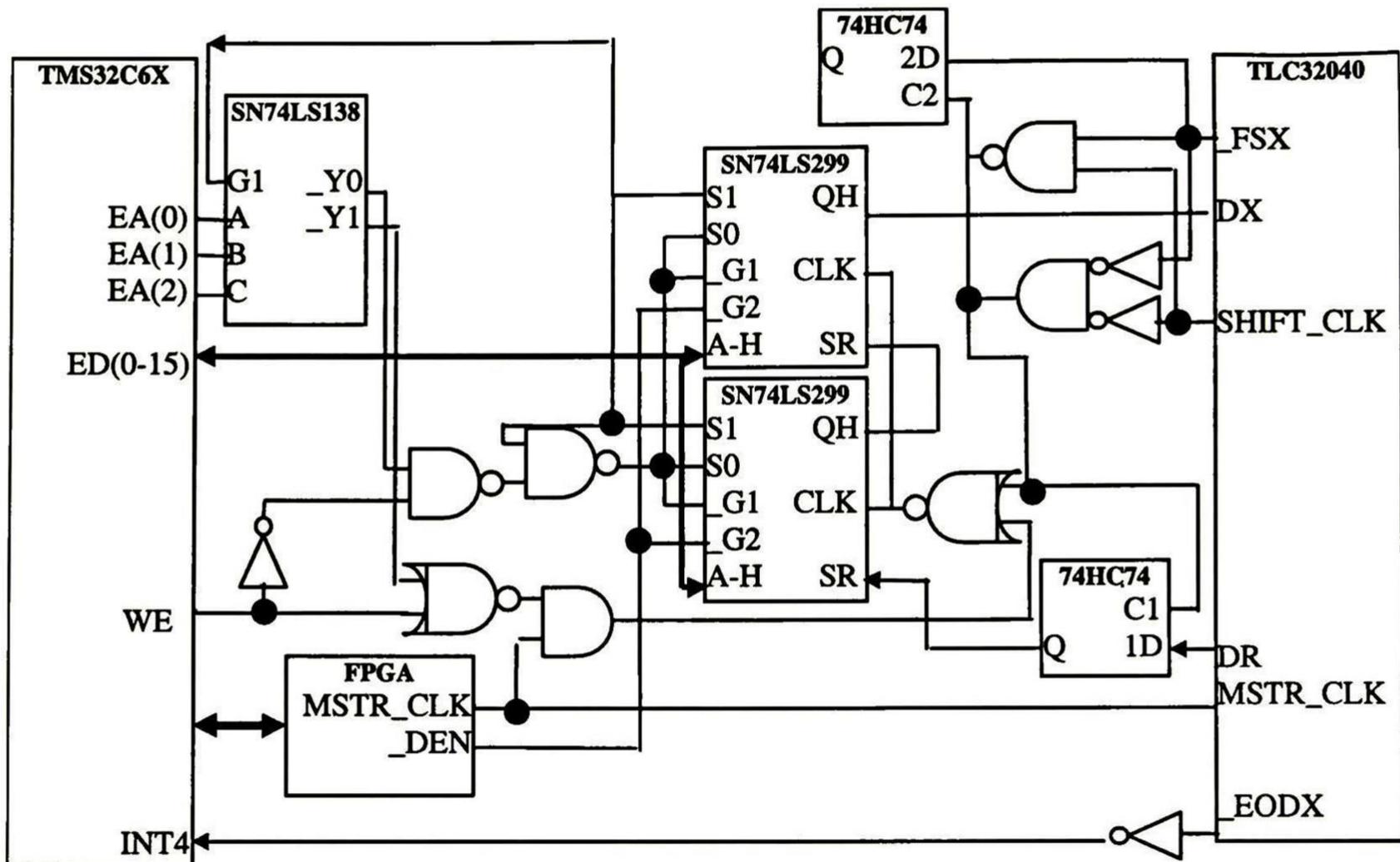


Figura 2. Diagrama a bloques de la conexión TLC32040 con el DSP

El TLC32040 no es conectado directamente al DSP porque su interfaz es serie y el DSP tiene interfaz paralela, así que es necesario adicionar circuitos para poder hacer la conexión. Básicamente se utiliza un 74LS299 recomendado en la hoja de especificación del convertidor para hacer la conversión (serie a paralelo – paralelo a serie).

Las señales requeridas por parte del DSP se muestran en el diagrama son:

- EA(0-2) pines de dirección, sirven para activar las salidas del decodificador 74LS136 para poder hacer las lecturas y escrituras al convertidor.
- ED(0-15) pines de datos, que llevan la información que se requiere leer o escribir del convertidor.
- _WE pin de escritura, se activa en 0 cuando se requiere hacer una escritura hacia el convertidor de lo contrario hace una lectura del convertidor.
- INT4 pin, por medio de este pin se le informa al DSP que tiene un dato para ser leído. Por cada muestra obtenida del convertidor este pin es activado.

Las señales que se utilizan del FPGA para realizar esta interfaz son:

- MSTR_CLK señal de reloj a 25MHz.
- _DEN señal de habilitación para obtener datos del 74LS299. Mientras este en alto, el dispositivo permanece en alta impedancia.

Otros dispositivos que intervienen en esta interfaz:

- 74LS299

Tarjeta Tabasco / Codificador de voz CS-ACELP

Registro de almacenamiento/desplazamiento. Modos de operación: almacenamiento, carga, desplazamiento a la izquierda y a la derecha. Los que se utilizan en la conexión son los dos tipos de desplazamientos y la carga de datos. Opera con salidas habilitadas o en estado de alta impedancia.

- 74LS138

Decodificador de direcciones. Tiene tres entradas de selección (A,B,C) y tres de habilitación de las cuales sólo una es usada (_G2A y _G2B son conectadas a tierras, sólo se usa G1). Sus señales son activadas en bajo.

- 74LS74

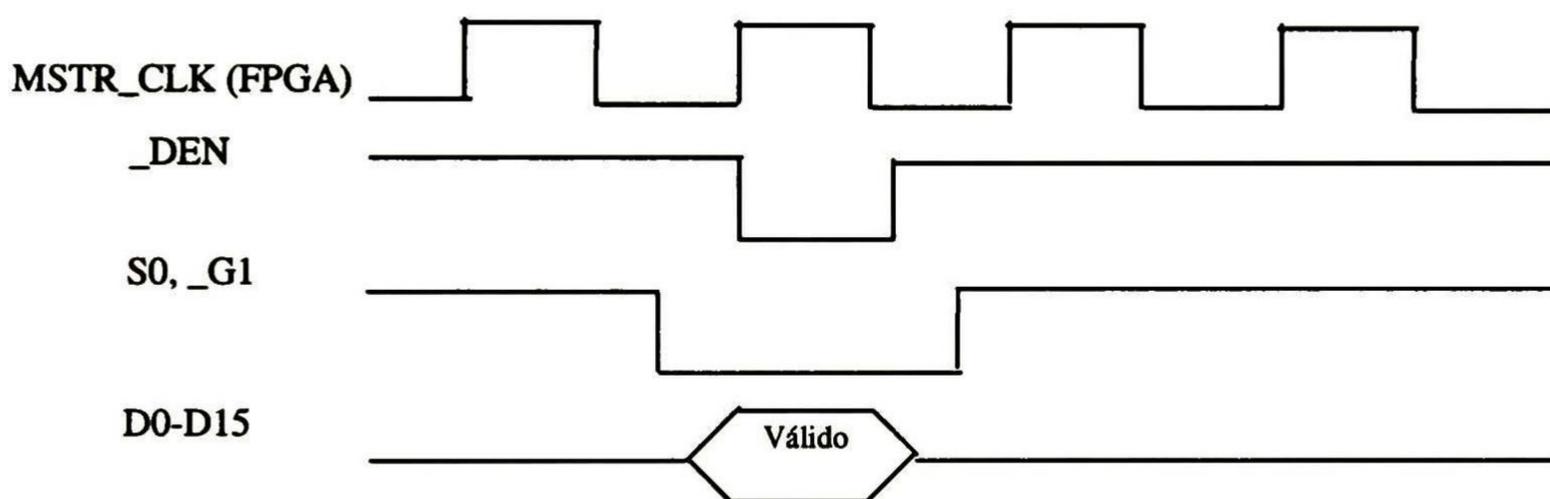
Este chip contiene dos Flipflops tipo D. Sus salidas son activas en los flancos de reloj.

- 74HC04,74HC01,74HC02,74HC08

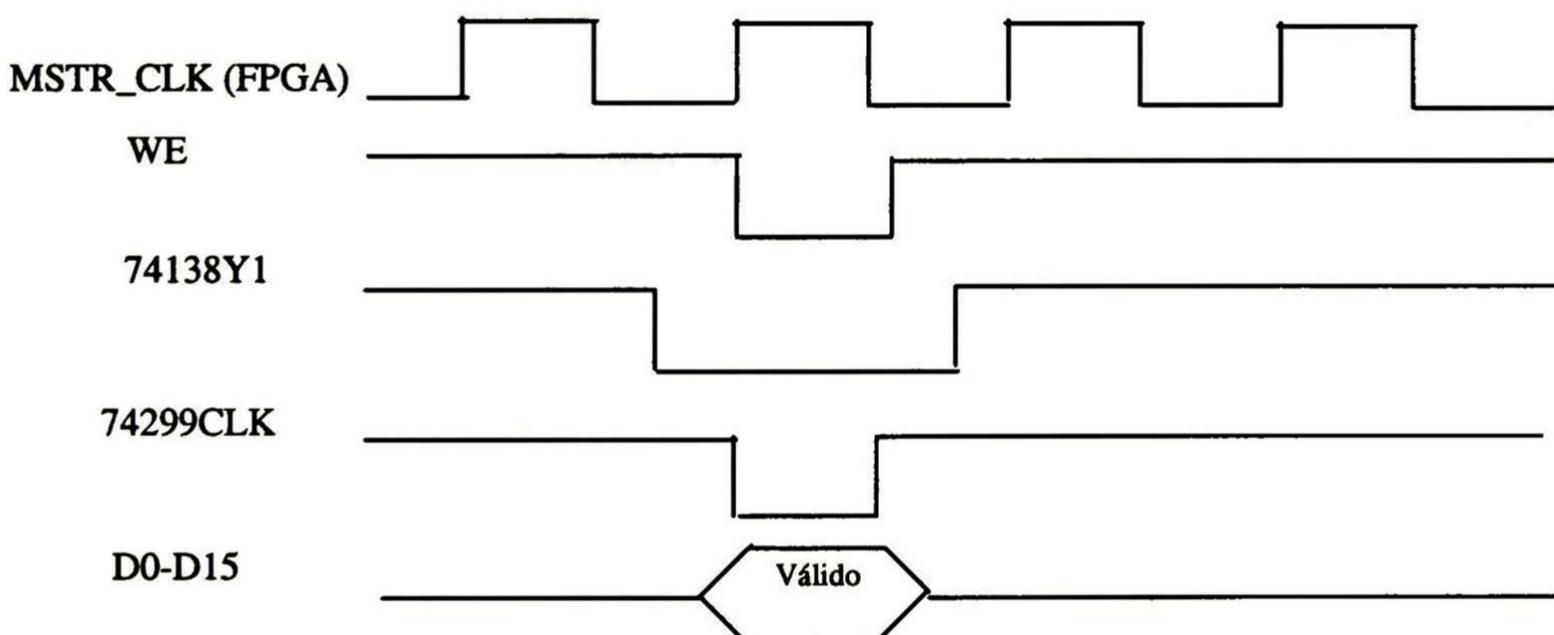
Son dispositivos TTL's que contienen compuertas NOT, NAND, NOR y AND respectivamente.

A continuación se muestran los diagramas de temporización para leer y escribir datos del convertidor:

Temporización para instrucción de lectura:



Temporización para instrucción de escritura:



Tarjeta Tabasco / Codificador de voz CS-ACELP

8.2.1.2 Conexión con micrófono y bocina

El TLC32040 también se encuentra conectado con un micrófono y una bocina para que se pueda llevar a cabo la comunicación con el usuario. El siguiente diagrama muestra las señales que intervienen en la conexión y los dispositivos utilizados:

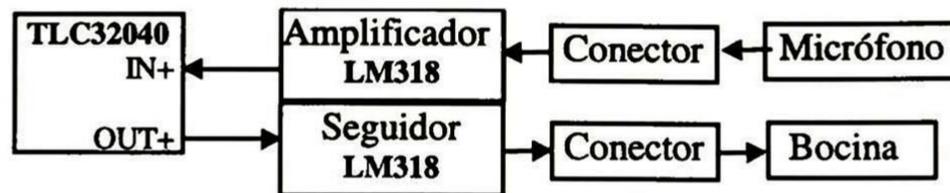


Figura 3. Diagrama a bloques del TLC32040 conectado al micrófono y la bocina

El amplificador y el seguidor utilizados son ajustables por medio de potenciómetros de 200k.

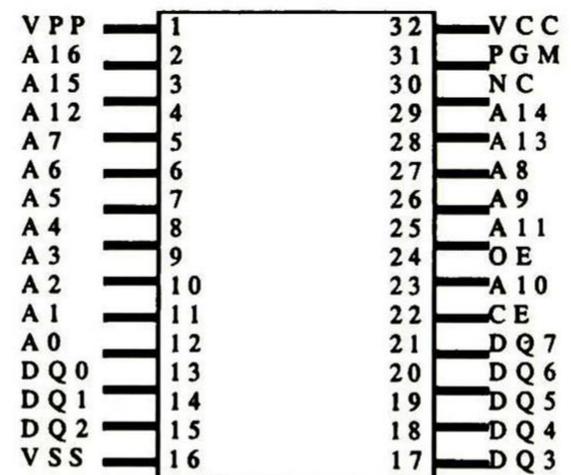
8.2.2 Memorias

8.2.2.1 EPROM

AM27C010 (128K X 8bit) es el número de parte de la EPROM utilizada en esta tarjeta y es fabricada por AMD. El tipo de empaquetado usado es DIP con ventana programable. Como el bus de datos del DSP es de 32 bits se utilizan cuatro memorias de este tipo.

Características:

- Tiempo de acceso 45ns
- Corriente de disipación 20uA
- Voltaje de alimentación 5V
- Alta inmunidad al ruido.
- Entradas y salidas compatibles con CMOS y TTL.
- 17 pines de direcciones (A0-A16).
- 8 pines de datos (DQ0-DQ7).



Aunque esta memoria es muy rápida con respecto a las que existen en el mercado, no alcanza a tener los datos listos al tiempo que el DSP los requiere. Por lo que es necesario agregar tiempos de espera en el DSP.

El programa que ejecuta el DSP se encuentra cargado inicialmente en la memoria EEPROM. Sin embargo, la memoria EEPROM requiere tiempos de espera, lo cual implica agregar retardos en la ejecución de los programas y reducir el número de usuarios que la tarjeta puede atender. Para reducir el tiempo de ejecución, los programas se cargarán durante la etapa de inicialización a la memoria RAM y se ejecutarán desde ella. Otra razón por la que se graba el algoritmo inicialmente en la EEPROM es para que una vez grabado el algoritmo no se pierda la información.

Tarjeta Tabasco / Codificador de voz CS-ACELP

8.2.2.1.1 Conexión con el DSP

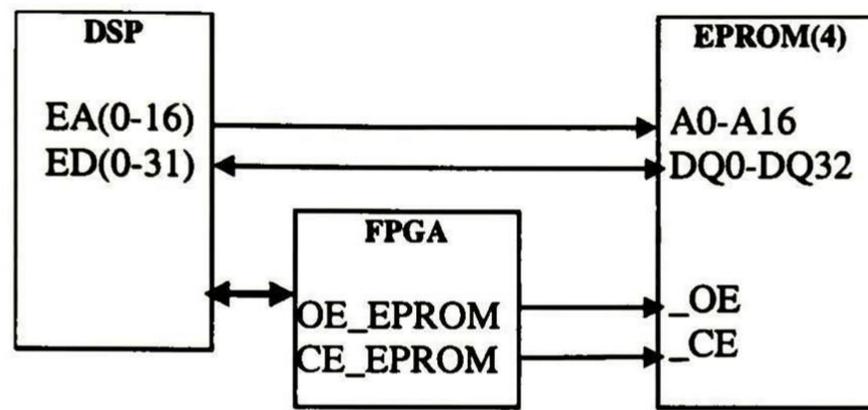


Figura 4. Diagrama a bloques del DSP conectado a las memorias EPROM

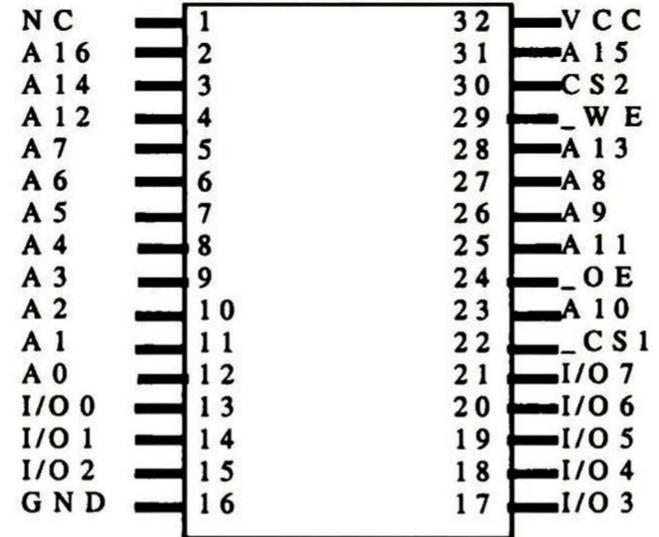
En el diagrama se consideran las cuatro EPROMs como una sola de 128K x 32bits.

8.2.2.2 SRAM

IDT71024 (128K X 8bit) es el número de parte de la memoria RAM estática utilizada en esta tarjeta y es fabricada por Integrated Device Technology (IDT). El tipo de empaquetado es Plastic SOJ. Al igual que la memoria EPROM se requieren cuatro de estas memorias para conectarse al DSP por el ancho del bus de datos (32 bits).

Características:

- Tiempo de acceso 20ns
- Corriente de disipación 140-160 mA
- Voltaje de alimentación 5V+- 0.5V
- Dos líneas de control (CS1, CS2).
- Todas las entradas y salidas son compatibles con dispositivos TTL



8.2.2.2.1 Conexión con el DSP

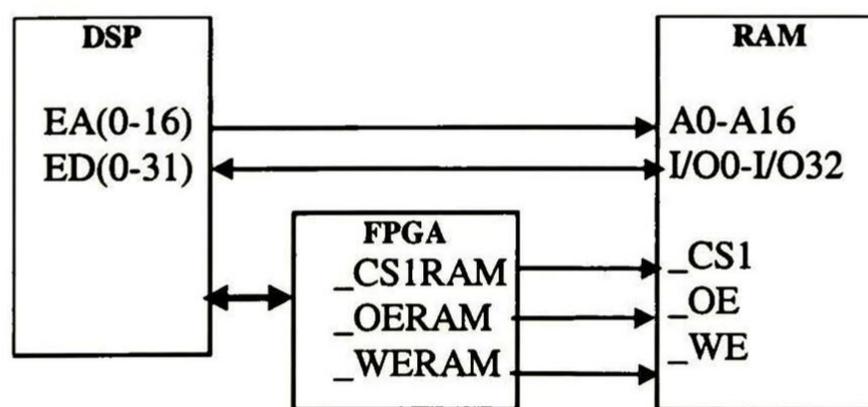


Figura 5. Diagrama a bloques del DSP conectado a las memorias RAM

Tarjeta Tabasco / Codificador de voz CS-ACELP

La señal CS2 se encuentra conectada directamente a 5V para que la selección de escritura o lectura se realicen únicamente con la señal _CS1. Aquí se representan las cuatro RAMs como una sola de 128K x 32bits.

8.2.2.3 MEMORIA DE PUERTO DUAL

IDT7026L15J es una memoria SRAM de 16K x 16 de 15ns de acceso. También es fabricada por IDT. El tipo de empaquetado es PLCC. Se utilizan dos memorias en la tarjeta.

Operación asíncrona.

Con 84 pines.

Compatible con dispositivos TTLs

Baja potencia de disipación 750mW activa

Seis señales VCC y tres de GND

Dos puertos independientes con señales de control, dirección y datos.

Soporta señalización de semáforo entre puertos.

R/WL	R/WR
_UBL	_UBR
_LBL	_LBR
_CEL	_CER
_OEL	_OER
I/O0L-I/O15L	IO0R-I/O15R
_BUSYL	_BUSYR
A13L-A0L	A13R-A0R
_SEML	_SEMR
	M/_S

Esta memoria es usada en la interfaz entre el DSP y el bus ISA. Se utiliza como almacenamiento temporal de datos entre ambas partes mientras cada uno de ellos tiene tiempo de tomarlos para procesarlos independientemente. Maneja dos tipos de paquetes de datos. Paquetes sin comprimir de 160 bytes y paquetes comprimidos de 10 bytes. En la figura 1, se manejan sólo paquetes de datos de 10 bytes y en la figura dos se manejan paquetes de datos tanto de 10 bytes como de 160 bytes. En el primer caso, se manejan sólo paquetes de 10 bytes porque los paquetes de datos que se transmiten de la PC a la tarjeta y viceversa, están comprimidos. En el segundo caso, los paquetes de datos que sean transmitidos de la red local a la computadora que tiene la tarjeta y viceversa, no estarán comprimidos (160bytes) y; los que sean transmitidos de la computadora que tiene la tarjeta a la red global y viceversa, estarán comprimidos (10 bytes).

8.2.2.3.1 Conexión con el DSP y el FPGA

En el siguiente diagrama aparecen todas las líneas que utiliza la memoria de puerto dual para su funcionamiento, las señales que van conectadas el DSP y las que van al FPGA.

Tarjeta Tabasco / Codificador de voz CS-ACELP

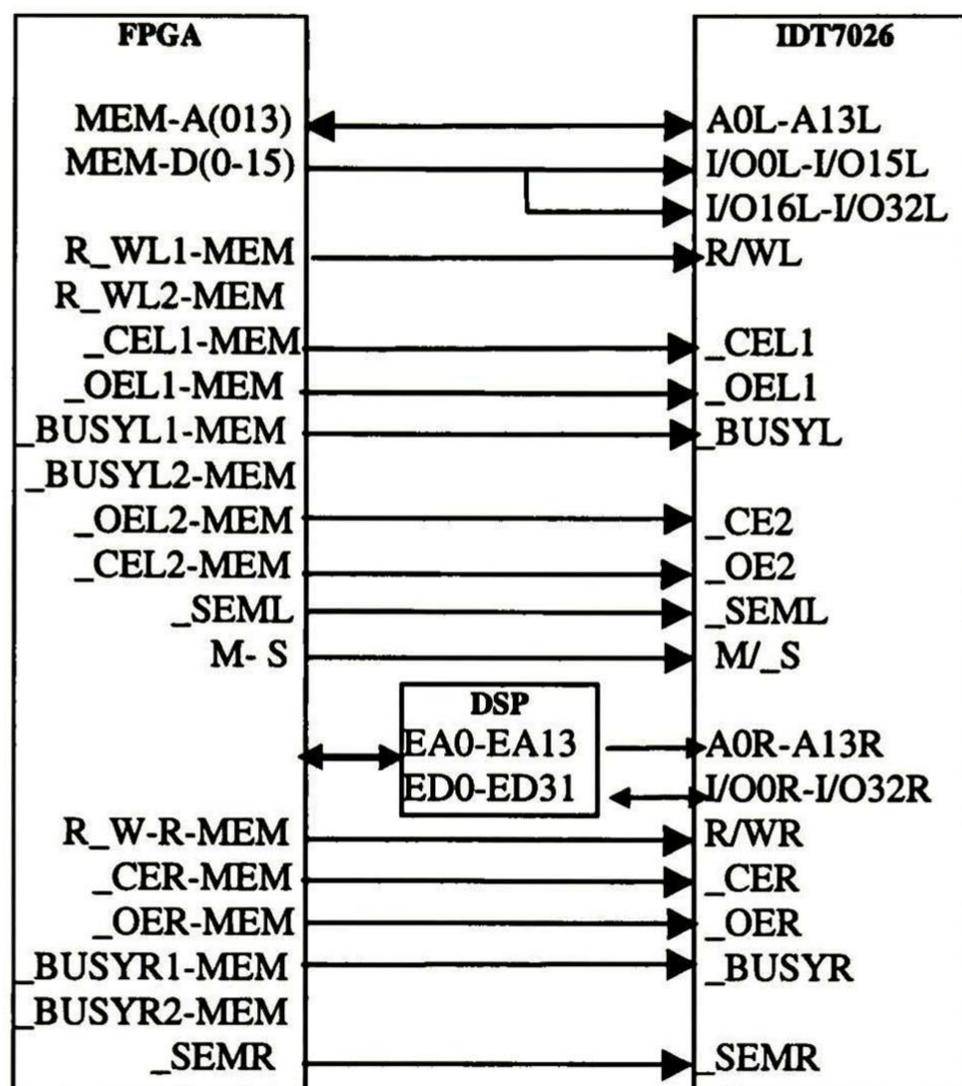


Figura 6. Diagrama a bloques de la memoria dual conectada al FPGA

El FPGA genera las señales que van conectadas a los dos puertos de la memoria dual. Las señales con terminación L corresponden a un puerto (originadas por la PC) y las de terminación R corresponden a otro (originadas por el DSP).

- Puerto L:

Las dos memorias utilizadas de puerto dual se consideran como una sola de 16K x32bits, por lo que solo hay una señal de habilitación _CER y OER. Para leer o escribir en esta memoria se usan las señales RW-R-MEM.

- Puerto R:

Aunque las memorias de puerto dual se consideran como una sola de 16K x32bits, se hacen dos transferencias de 16bits. Por esta razón no se habilitan al mismo tiempo (se consideran dos señales CEL y dos señales OEL) y para leer o escribir en ella se usan dos señales R_WL-MEM.

La situación de acceso al mismo tiempo a una misma dirección de memoria nunca se va a dar. Los datos deben ser intercambiados por paquetes completos. Entonces tanto el bus como el DSP no van a ir a buscar algo que el otro esta a penas escribiendo. Cuando el bus o el DSP termina la escritura de un paquete de datos, es entonces que se interrumpe al interlocutos para avisarle que vaya a buscar la nueva información.

Nota: Las señales _UBL y _LBL de ambas memorias se fijan a cero lógico (conectadas a tierra). La señal BUSY se utiliza para que no se lea o escriba en una misma localidad a la vez.

8.2.3 Control

El dispositivo a utilizar para controlar el acceso al bus por los dispositivos externos al DSP es un FPGA de Xilinx (4010). Será programado para codificar direcciones y activar el dispositivo que necesite el bus de datos del DSP. Los pines de entrada considerados para realizar el control se toman del DSP y se muestran en el siguiente bloque. También se muestran las señales que se generan para indicar el ciclo de lectura o escritura de cada dispositivo externo.

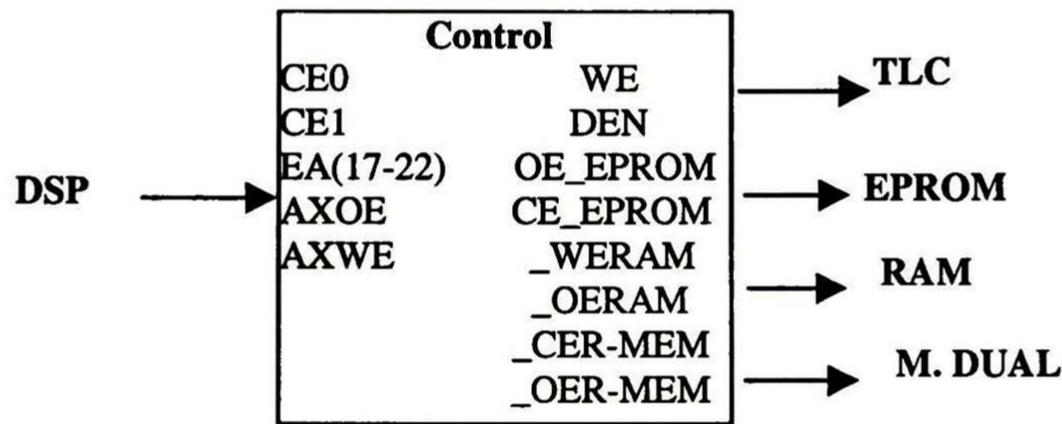


Figura 7. Diagrama a bloques del control

8.2.4 DSP

Es un TMS230C6201 de Texas Instruments de punto fijo. Su máxima velocidad es de 1600 MIPS con un ciclo de reloj de 5ns. Las señales utilizadas en el DSP se muestran en el siguiente diagrama.

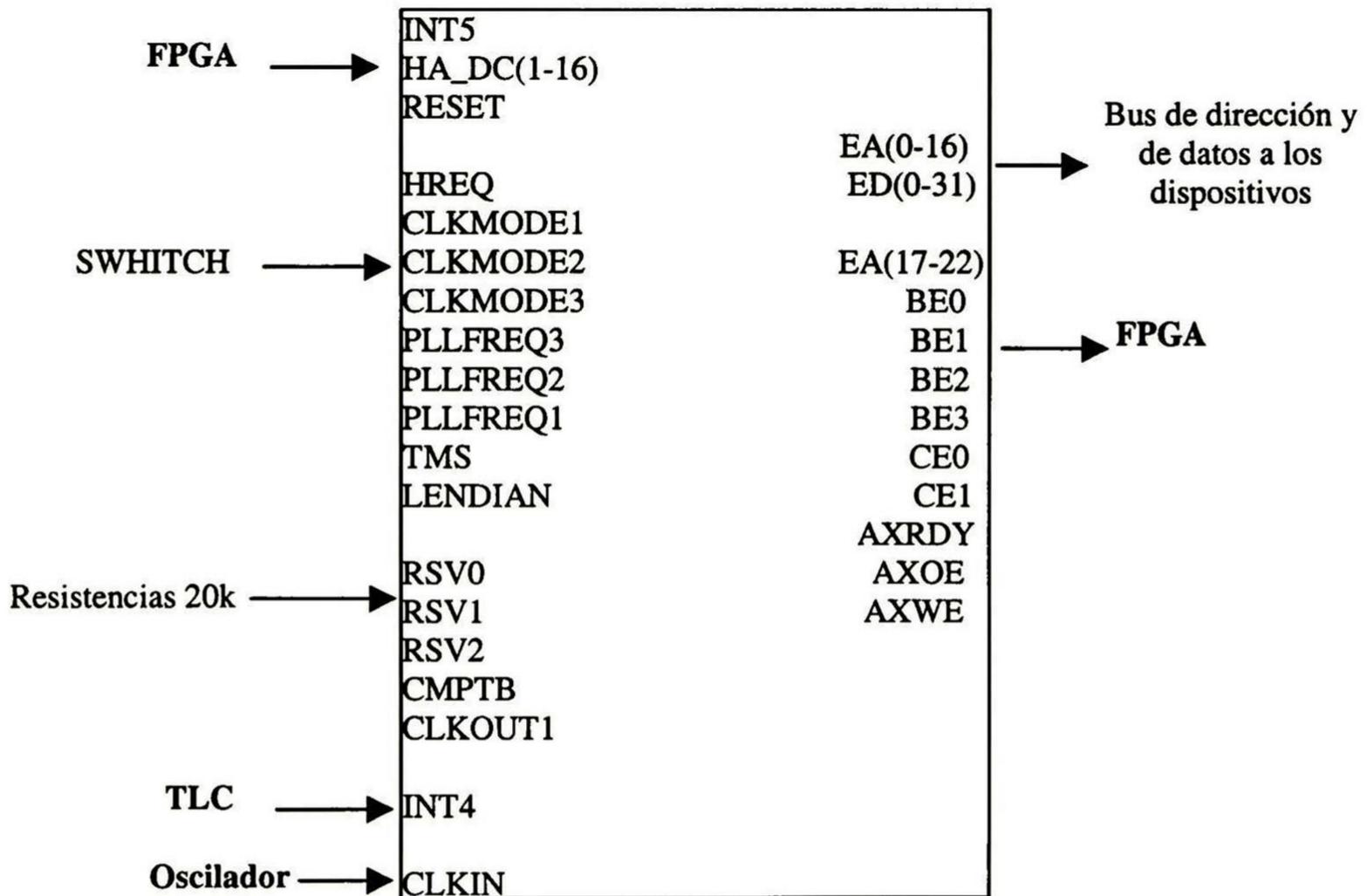


Figura 8. Diagrama a bloques de las señales usadas del DSP

Tarjeta Tabasco / Codificador de voz CS-ACELP

Este DSP tiene dos niveles de voltaje de alimentación: uno de 2.5V y otro de 3.3V, estas alimentaciones son derivadas de dos reguladores de voltajes. HA_DC(16:1) son un bus de datos que pueden ser conectadas a la PC, sin embargo, en este diseño solo se usan para inicializar el DSP y salen del FPGA.

Los demás pines que se encuentran a la izquierda del diagrama HREQ-LENDIAN son fijados mediante un switch y una de las funciones que tiene es seleccionar la frecuencia que manejará el DSP. Los pines que van conectados al valor de resistencia de 20k no son utilizadas y se fijan a un valor para que no se queden sin conectar (ver en hoja de especificación del DSP). El pin INT4 vienen del convertidor y fue explicado en la sección del TLC32040. El RESET sirve para dar inicialización al DSP y todo el sistema que requiera de este y el INT5 sirve para interrumpir al DSP indicándole que la memoria dual tiene un dato para que este lo comprima o lo descomprima.

Respecto a los pines del lado derecho, se puede ver que estos son utilizados (la mayoría) en el FPGA para controlar las entradas y salidas de los datos a cada dispositivo externo.

El mapa de memoria utilizada en este DSP es el mapa 0. CE0 (4Mx8bits) y CE1(16Mx8bits) son considerados para elegir una zona de memoria donde estarán ubicados los dispositivos externos, [hoja de especificación del DSP]. La selección de esta zona de memoria, se considera al inicio de la configuración de la tarjeta dados por los pines HA-DC(1-16).

Dirección		Bytes
000 0000	Espacio Memoria CE0	16M
100 0000	Espacio Memoria CE1	4M
140 0000	Programa Interno RAM	64K
141 0000	Reservado	4M
180 0000	Periférico Interno	4M
1C0 0000	Reservado	4M
200 0000	Espacio Memoria CE2	32M
400 0000	Reservado	1984M
8000 0000	Dato Interno RAM	64K
8001 0000	Reservado	4M
8040 0000	Reservado	2044M
1 0000 0000		

Figura 9. Mapa de memoria 0.

8.2.4 Interfaz al bus ISA

La interfaz la realiza el dispositivo FPGA (4010). Este dispositivo se encarga de controlar la transmisión y/o recepción de voz comprimida y/o descomprimida entre el bus de la PC y la tarjeta Tabasco. En esta interfaz se consideran memorias de puerto dual para el almacenamiento temporal de los datos, encargado del control de acceso al bus del DPS. En el siguiente diagrama se muestran las señales del bus ISA consideradas en el diseño así también como las señales que la interfaz generaría para escribir y leer datos de la memoria dual:

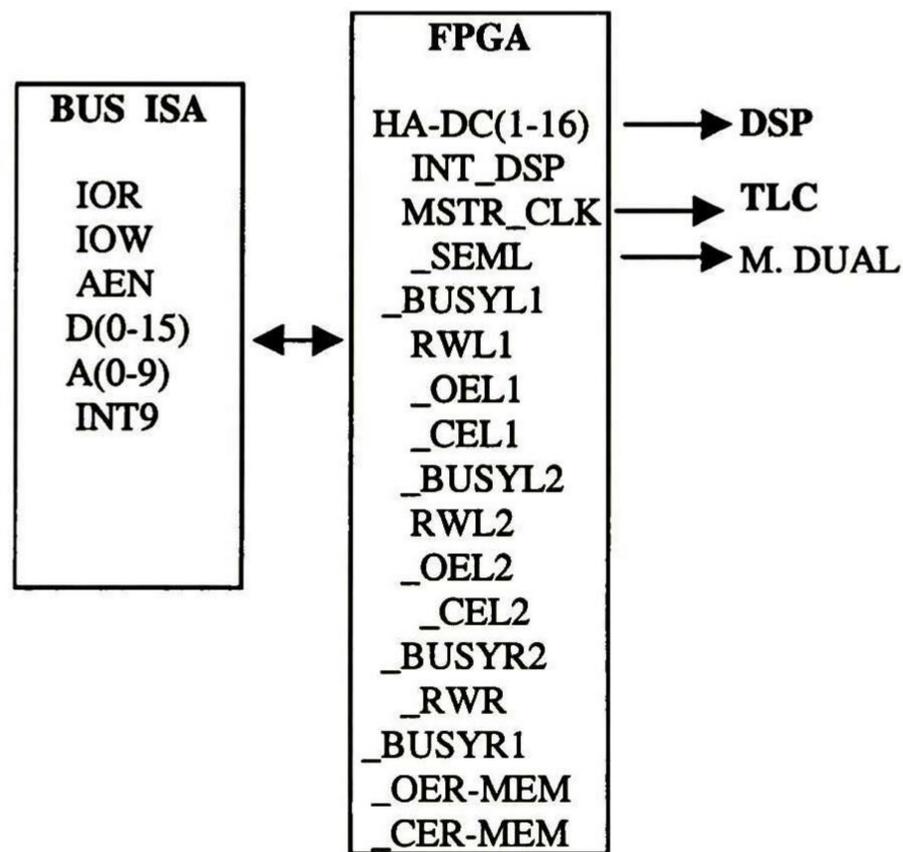


Figura 10. Diagrama a bloques del bus ISA conectado al FPGA

Aunque no se señalaron en el diagrama también se consideran los niveles de alimentación del BUS (+5,-5,+12,-12, GND). Y a la entrada el FPGA para esta interfaz también se consideran líneas que vienen de un switch U34, que se utiliza para seleccionar la dirección del puerto en el que la tarjeta va a trabajar.

Los pines que van al DSP fueron mencionados en la sección anterior, INT_DSP va al pin de entrada INT5 del DSP. El MSTR_CLK es utilizado para que el TLC funcione debidamente, éste utiliza una frecuencia de 25 MHz y es derivado del oscilador que va al DSP. Los demás pines van a la memoria de puerto dual y son utilizados para habilitarla, para tener acceso a determinada parte de la memoria y para ponerlas en estado de espera si es necesario.

El acceso que se va a tener a la PC es por medio del BUS ISA y se va a configurar en forma de puerto. Por eso el uso de las señales IOR e IOW, así también como la señal AEN.

Podemos resumir las funciones del FPGA como se menciona a continuación:

Tarjeta Tabasco / Codificador de voz CS-ACELP

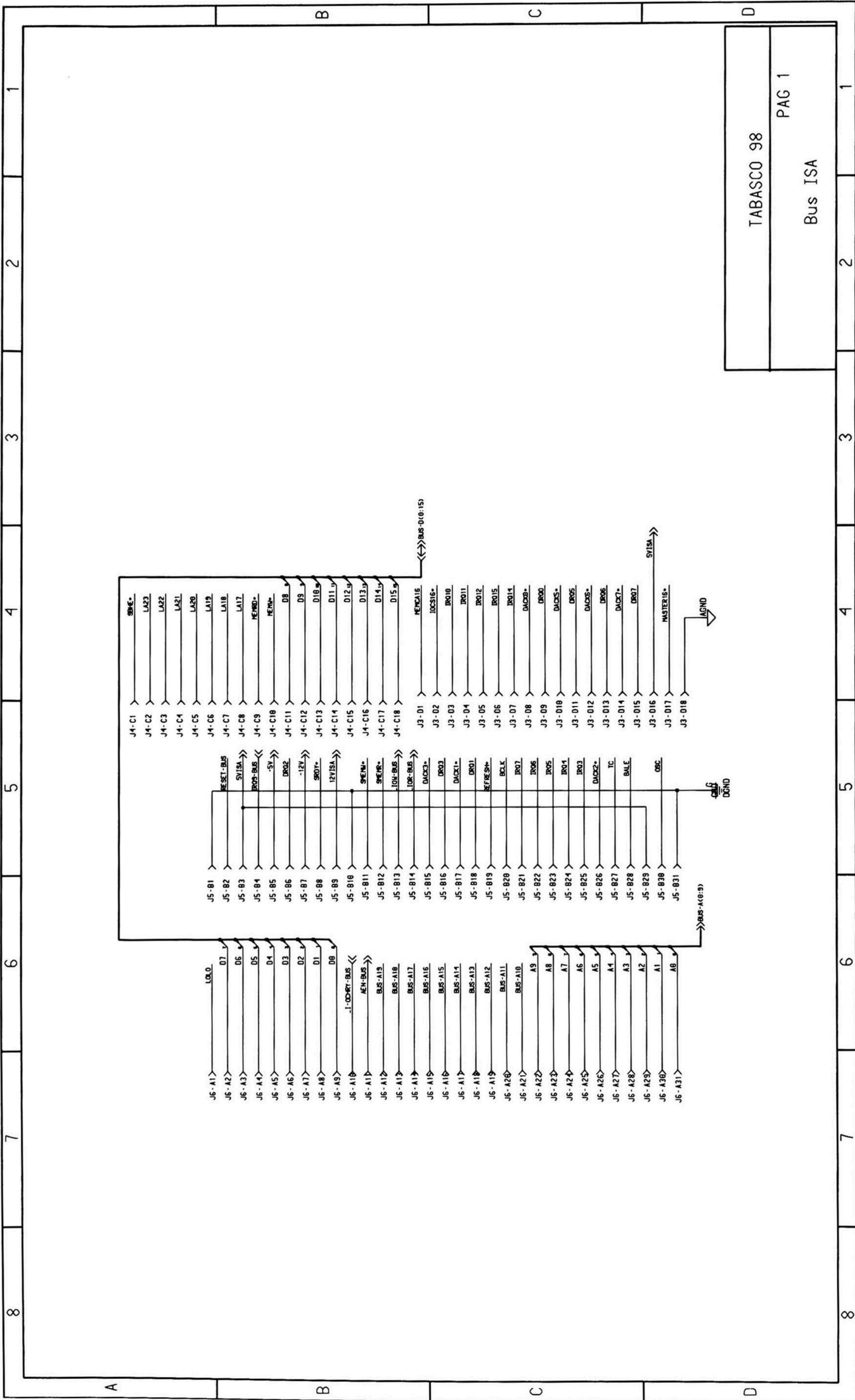
- Configurar el modo de trabajo del DSP.
- Dar reset al DSP y otros componentes que tienen esta función.
- Tener un divisor de frecuencia de 50MHz a 25 MHz para que el TLC32040 funcione.
- Controlar el acceso del bus de datos del DSP que tienen acceso el TLC32040, memorias RAM, memorias PUERTO DUAL.
- Transmitir y controlar el intercambio de datos entre el BUS ISA y el DSP (interrupción al DSP, interrupción a la PC, direcciones y datos a las memorias PUERTO DUAL): interfaz Al BUS ISA.

El sistema utiliza dos memorias (X1765,XC17128) para guardar la configuración del FPGA

Los diagramas esquemáticos se pueden ver en el anexo siguiente.

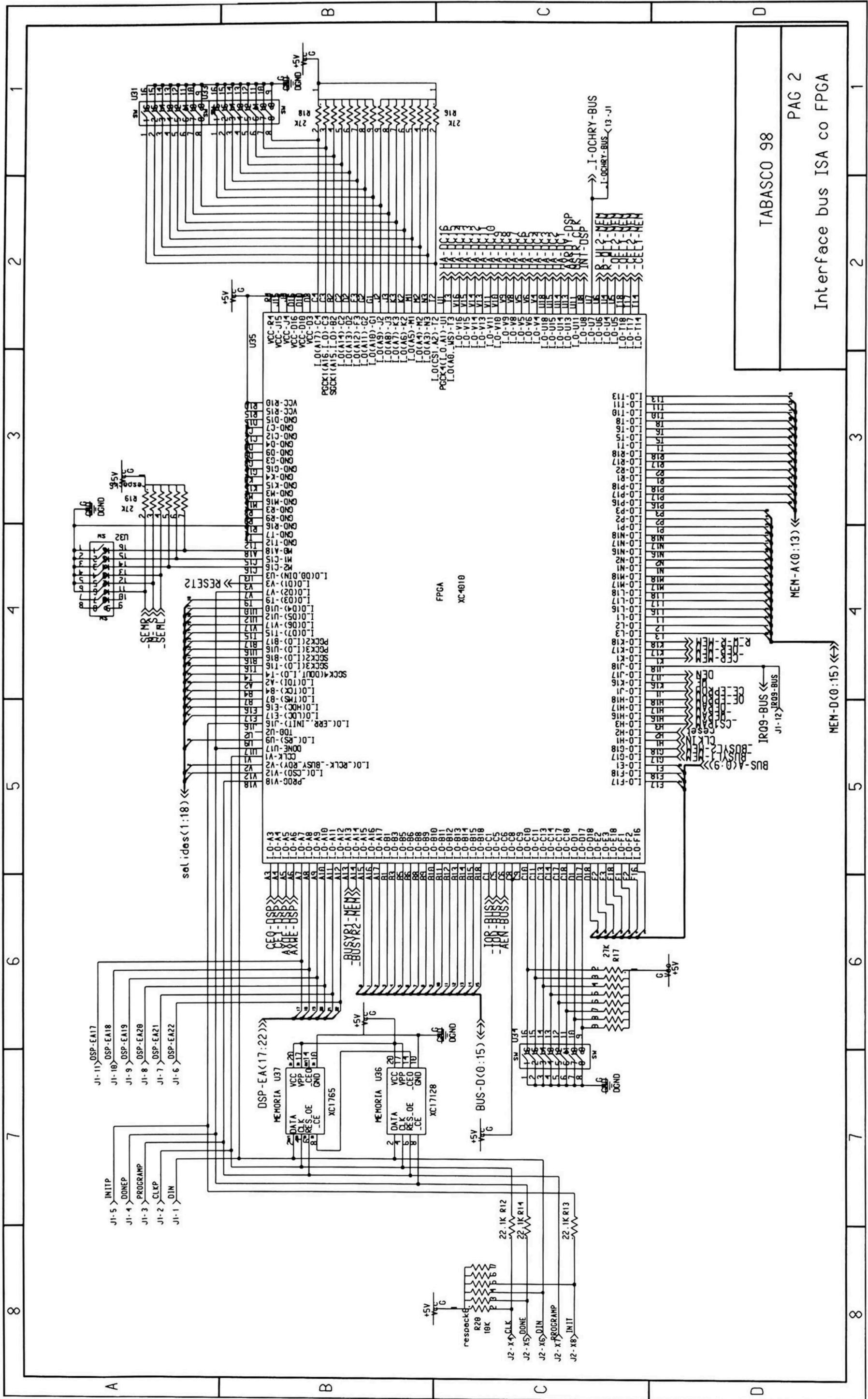
APÉNDICE B

Diagramas Esquemáticos



TABASCO 98
Bus ISA

PAG 1



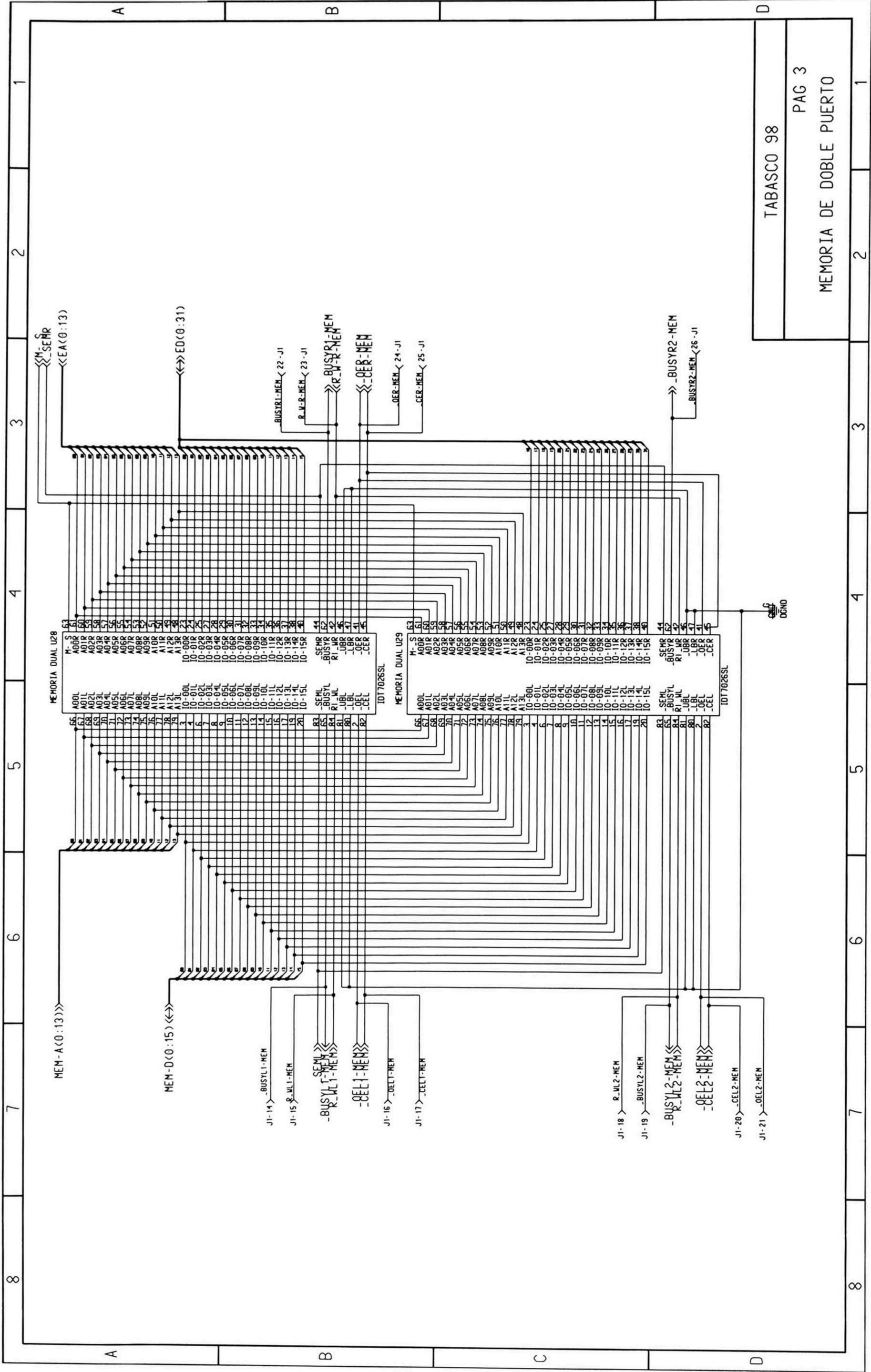
TABASCO 98
Interface bus ISA co FPGA

1 2 3 4 5 6 7 8

1 2 3 4 5 6 7 8

1 2 3 4 5 6 7 8

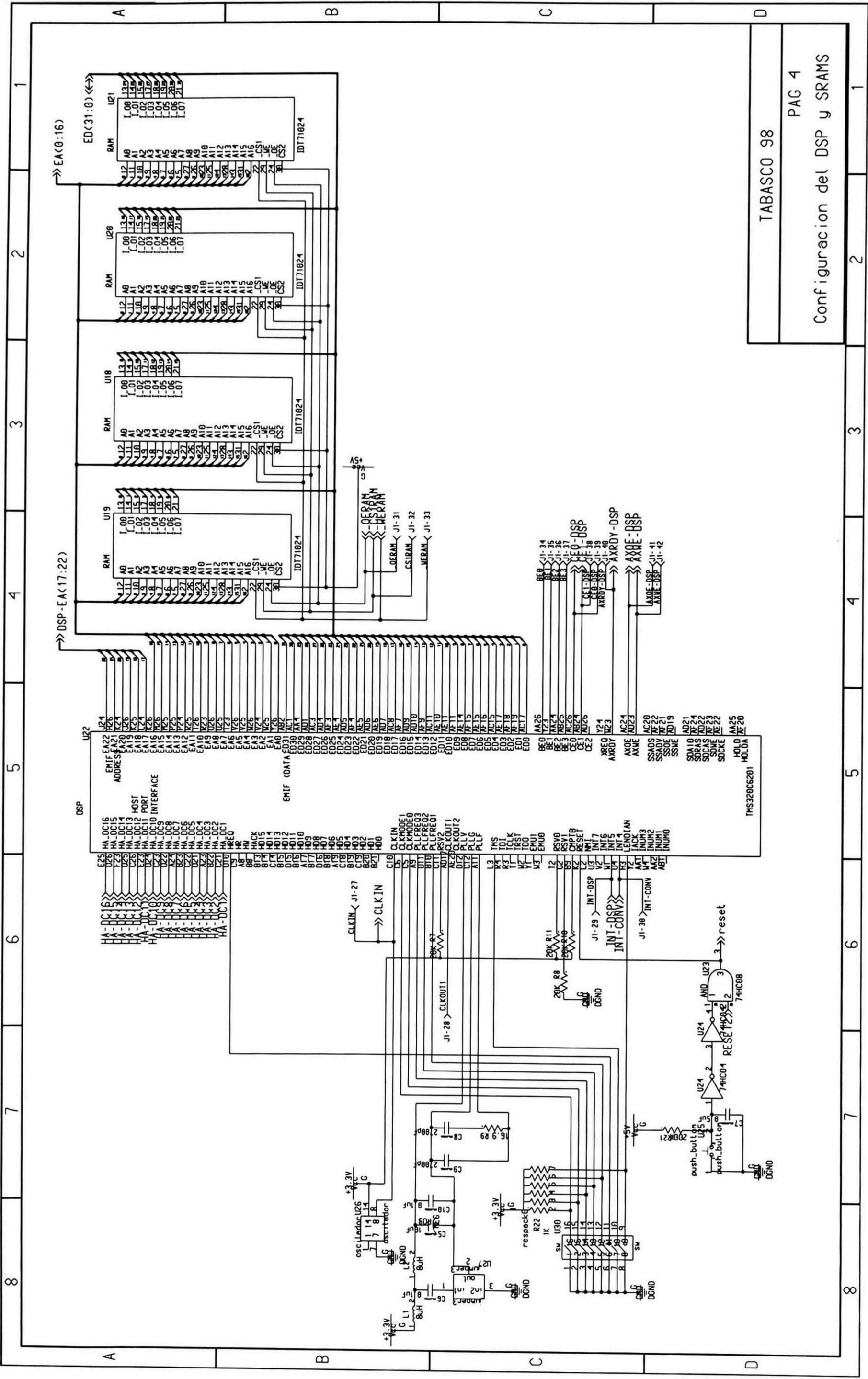
1 2 3 4 5 6 7 8



TABASCO 98

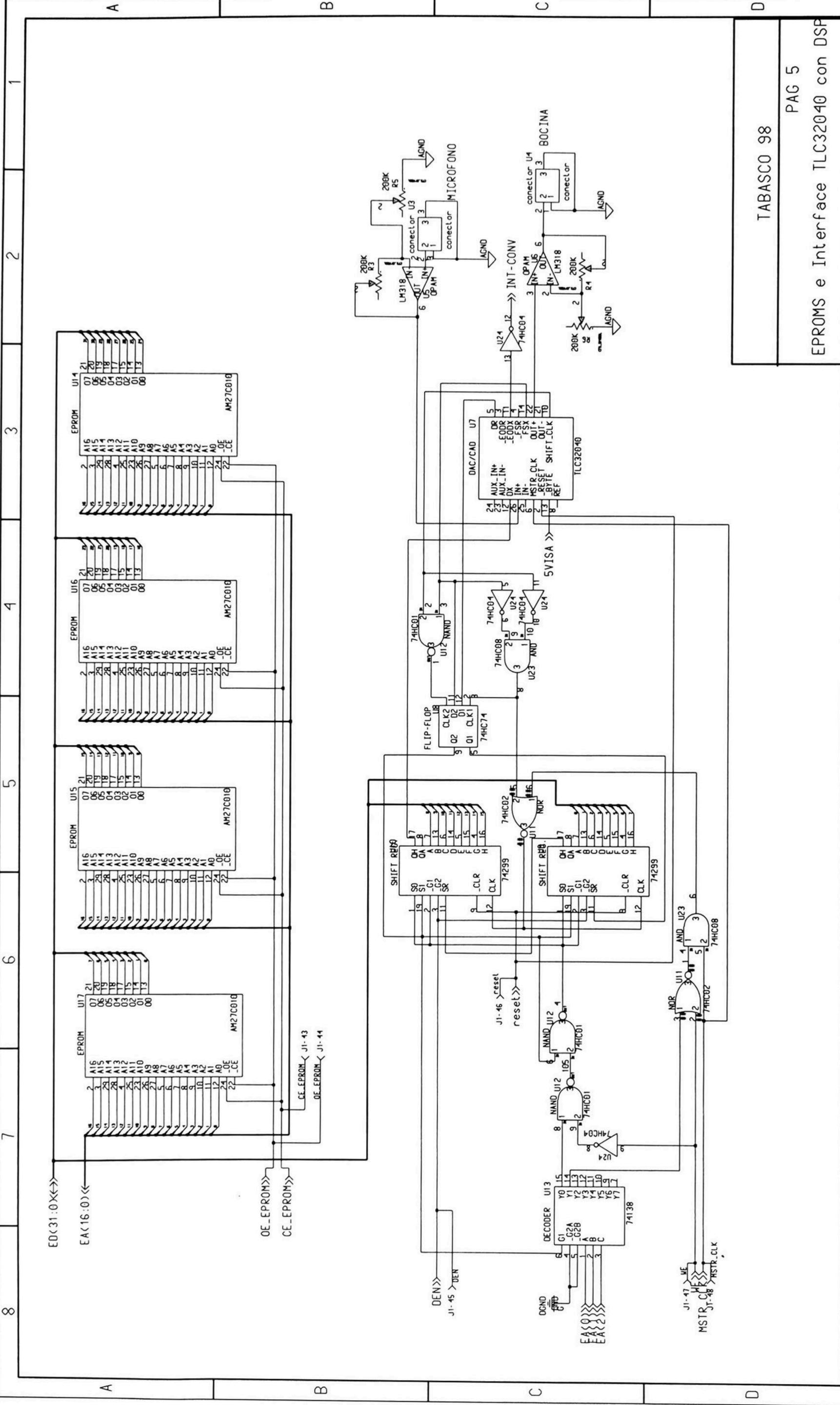
PAG 3

MEMORIA DE DOBLE PUERTO



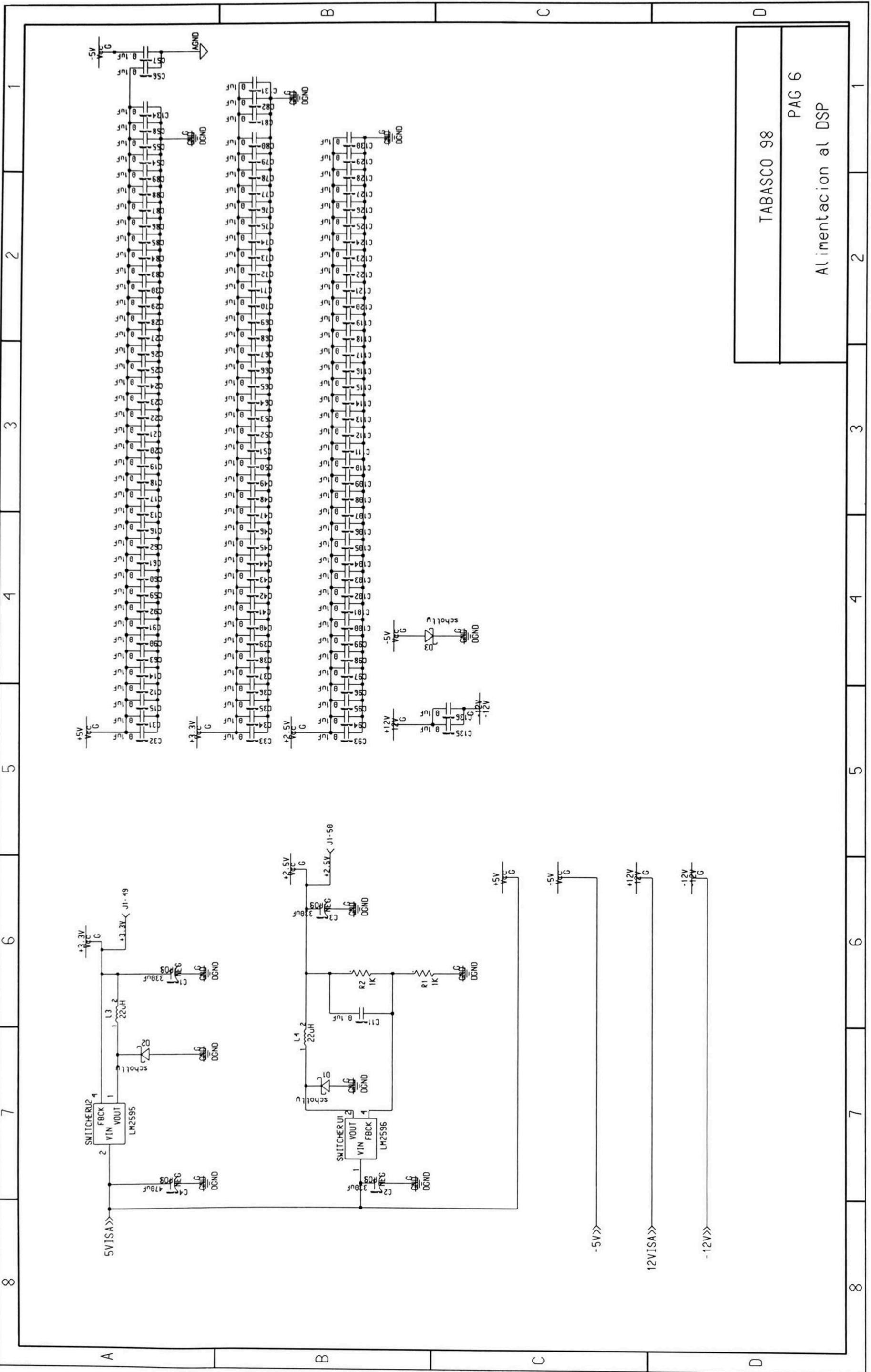
1 2 3 4 5 6 7 8

1 2 3 4 5 6 7 8



1 2 3 4 5 6 7 8

A B C D



TABASCO 98

PAG 6

Alimentacion al DSP

APÉNDICE C

Lista de Materiales

Board Station BOM file

ITEM NUMBER	COMPANY PART NO.	GEOMETRY	COUNT	DESCRIPTION	REFERENCE
1	74138	DIP16	1	DECODER	U13
2	74299	DIP20	2	SHIFT REG.	U9 U10
3	74HC01	DIP14	1	NAND	U12
4	74HC02	DIP14	1	NOR	U11
5	74HC04	DIP14	1	NOT	U24
6	74HC08	DIP14	1	AND	U23
7	74HC74	DIP14	1	FLIP-FLOP	U8
8	AM27C010	DIP32-W	4	EPROM	U14 U15 U1
9	C	c_elect	1	cap_elect	C5
10	C	c_cer	6	CAPACITOR	C6 C7 C8 C11
11	C2	c_elect2	2	cap_elect	C3 C4
12	C3	c_elect3	2	cap_elect	C1 C2
13	CON-XILINX	con-xilinx	1	CONN_PIN	J2
14	IDT7026SL	PLCC84	2	MEMORIA DUAL	U28 U29
15	IDT71024	S032-3	4	RAM	U18 U19 U2
16	ISABUSA	isabusa	1	CONN_PIN	J6
17	ISABUSB	isabusb	1	CONN_PIN	J5
18	ISABUSC	isabusc	1	CONN_PIN	J4
19	ISABUSD	isabusd	1	CONN_PIN	J3
20	L	L	4	INDUCTOR	L1 L2 L3 L4
21	LM318	DIP8	2	OPAM	U5 U6
22	LM2595	TO263	1	SWITCHER	U2
23	LM2596	TO263	1	SWITCHER	U1
24	PRESET	preset2	4	preset	R3 R4 R5 R6
25	R	r14	8	RESISTOR	R7 R8 R9 R10 R11
26	R2	r12	2	RESISTOR	R12 R13 R14
27	TLC32040	PLCC28	1	DAC/CAD	R1 R2
28	TMS320C6201	BGA352	1	DSP	U7
29	XC1765	PLCC20	1	MEMORIA	U22
30	XC4010	PG191	1	FPGA	U37
31	XC17128	PLCC20	1	MEMORIA	U35
32	con-2x25	con-2x25	1	CONN_PIN	U36
33	conector	conector	2	conector	J1
34	jumper3	jumper3	1	jumper3	U3 U4
35	oscilador	oscilador	1	oscilador	U27
36	push_buttom	puch_buttom	1	push_buttom	U26
37	respack6	sip-7	2	respackk6	U25
38	respack8	sip-9	4	respack8	R19 R20
39	schotty	CASE267-03	2	diodo	D1 D2
40	sw	DIP16-SWITCH	5	sw	U30 U31 U32 U33 U34

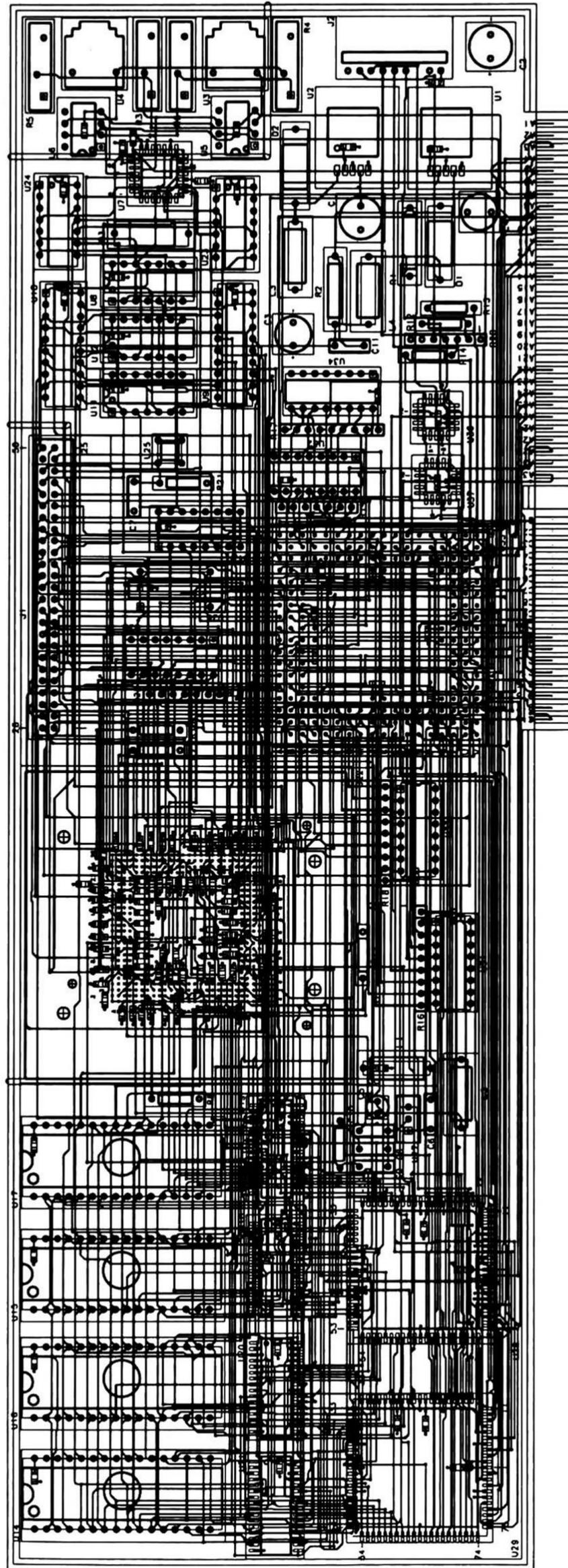
REFERENCE	ITEM_NUMBER	COMPANY PART NO.	GEOMETRY	DESCRIPTION
C1	12	C3	c_elect3	cap_elect
C2	12	C3	c_elect3	cap_elect
C3	11	C2	c_elect2	cap_elect
C4	11	C2	c_elect2	cap_elect

C5	9	C	c_elect	cap_elect
C6	10	C	c_cer	CAPACITOR
C7	10	C	c_cer	CAPACITOR
C8	10	C	c_cer	CAPACITOR
C9	10	C	c_cer	CAPACITOR
C10	10	C	c_cer	CAPACITOR
C11	10	C	c_cer	CAPACITOR
D1	39	schotty	CASE2607-03	diodo
D2	39	schotty	CASE2607-03	diodo
J1	32	con-2x25	con-2x25	CONN_PIN
J2	13	CON-XILINX	con.xilinx	CONN_PIN
J3	19	ISABUSD	isabusd	CONN_PIN
J4	18	ISABUSC	isabusc	CONN_PIN
J5	17	ISABUSB	isabusb	CONN_PIN
J6	16	ISABUSA	isabusa	CONN_PIN
L1	20	L	L	INDUCTOR
L2	20	L	L	INDUCTOR
L3	20	L	L	INDUCTOR
L4	20	L	L	INDUCTOR
R1	26	R2	r12	RESISTOR
R2	26	R2	r12	RESISTOR
R3	24	PRESET	preset2	preset
R4	24	PRESET	preset2	preset
R5	24	PRESET	preset2	preset
R6	24	PRESET	preset2	preset
R7	25	R	r14	RESISTOR
R8	25	R	r14	RESISTOR
R9	25	R	r14	RESISTOR
R10	25	R	r14	RESISTOR
R11	25	R	r14	RESISTOR
R12	25	R	r14	RESISTOR
R13	25	R	r14	RESISTOR
R14	25	R	r14	RESISTOR
R15	38	respack8	sip-9	respack8
R16	38	respack8	sip-9	respack8
R17	38	respack8	sip-9	respack8
R18	38	respack8	sip-9	respack8
R19	37	respack6	sip-7	respack6
R20	37	respack6	sip-7	respack6
U1	23	LM2596	TO263	SWITCHER
U2	22	LM2595	T0263	SWITCHER
U3	33	conector	conector	conector
U4	33	conector	conector	conector
U5	21	LM318	DIP8	OPAM
U6	21	LM318	DIP8	OPAM
U7	27	TLC32040	PLCC28	DAC/CAD
U8	7	74HC74	DIP14	FLIP-DLOP
U9	2	74299	DIP29	SHIFT REG.
U10	2	74299	DIP20	SHIFT REG.
U11	4	74HC02	DIP14	NOR
U12	3	74HC01	DIP14	NAND

U13	1	74138	DIP16	DECODER
U14	8	AM27C010	DIP32-W	EPROM
U15	8	AM27C010	DIP32-W	EPROM
U16	8	AM27C010	DIP32-W	EPROM
U17	8	AM27C010	DIP32-W	EPROM
U18	15	IDT71024	SO32-3	RAM
U19	15	IDT71024	SO32-3	RAM
U20	15	IDT71024	SO32-3	RAM
U21	15	IDT71024	SO32-3	RAM
U22	28	TMS320C6201	BG4352	DSP
U23	6	74HC08	DIP14	AND
U24	5	74HC04	DIP14	NOT
U25	36	push_buttom	push_buttom	push_buttom
U26	35	oscilador	oscilador	oscilador
U27	34	jumper3	jumper3	jumper3
U28	14	IDT7026SL	PLCC84	MEMORIA DUAL
U29	14	IDT7026SL	PLCC84	MEMORIA DUAL
U30	40	sw	DIP16-SWITCH	sw
U31	40	sw	DIP16-SWITCH	sw
U32	40	sw	DIP16-SWITCH	sw
U33	40	sw	DIP16-SWITCH	sw
U34	40	sw	DIP16-SWITCH	sw
U35	30	XC4010	PG181	FPGA
U36	31	XC17128	PLCC20	MEMORIA
U37	29	XC1765	PLCC20	MEMORIA

APÉNDICE D

Diseño Físico de la Tarjeta



APÉNDICE E

Archivos de Fabricación

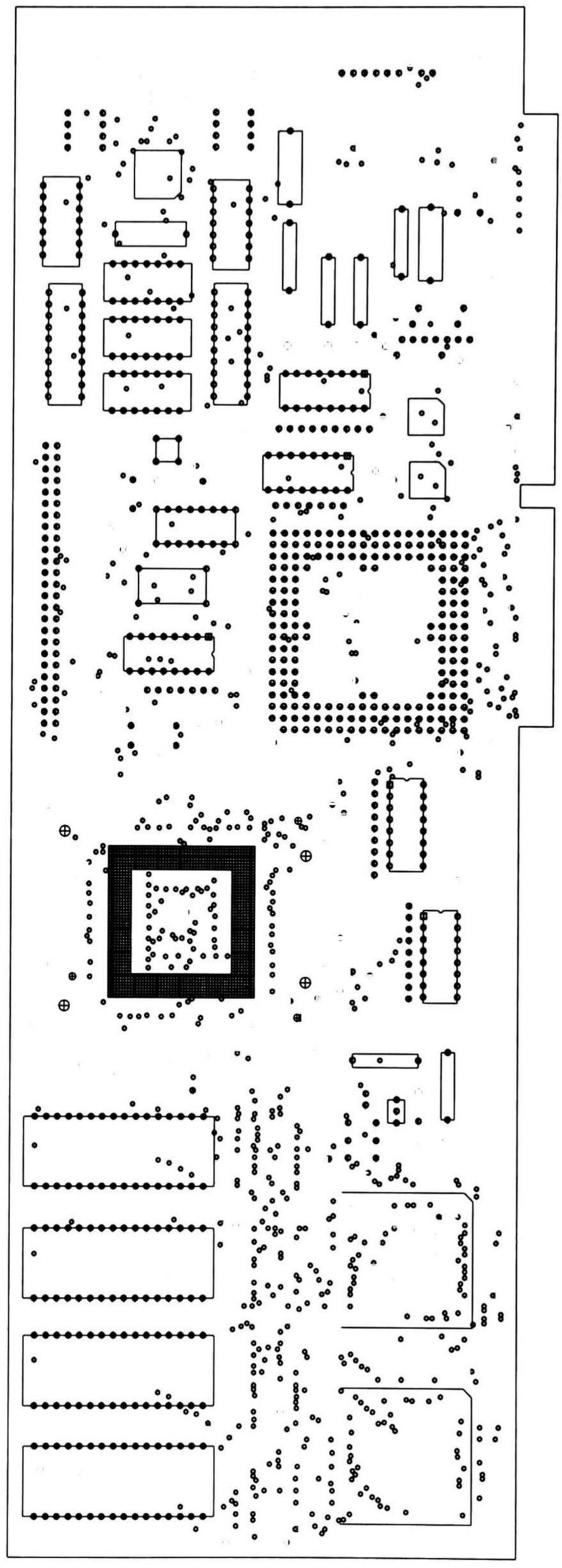
ARTWORK REVISION CONTROL	LAYER QTY
DESCRIPTION	6
SILKSCREEN_2 A/W SHT	1
SOLDER_MASK_1 A/W SHT	2
SIGNAL_1, PAD_1 A/W SHT	3
POWER_1, POWER_2 A/W SHT	4
SIGNAL_2 A/W SHT	5
SIGNAL_3 A/W SHT	6
POWER_3, POWER_4 A/W SHT	7
SIGNAL_4, PAD_2 A/W SHT	8
SOLDER_MASK_2 A/W SHT	9
SILKSCREEN_2 A/W SHT	10
DRILL A/W SHT	11

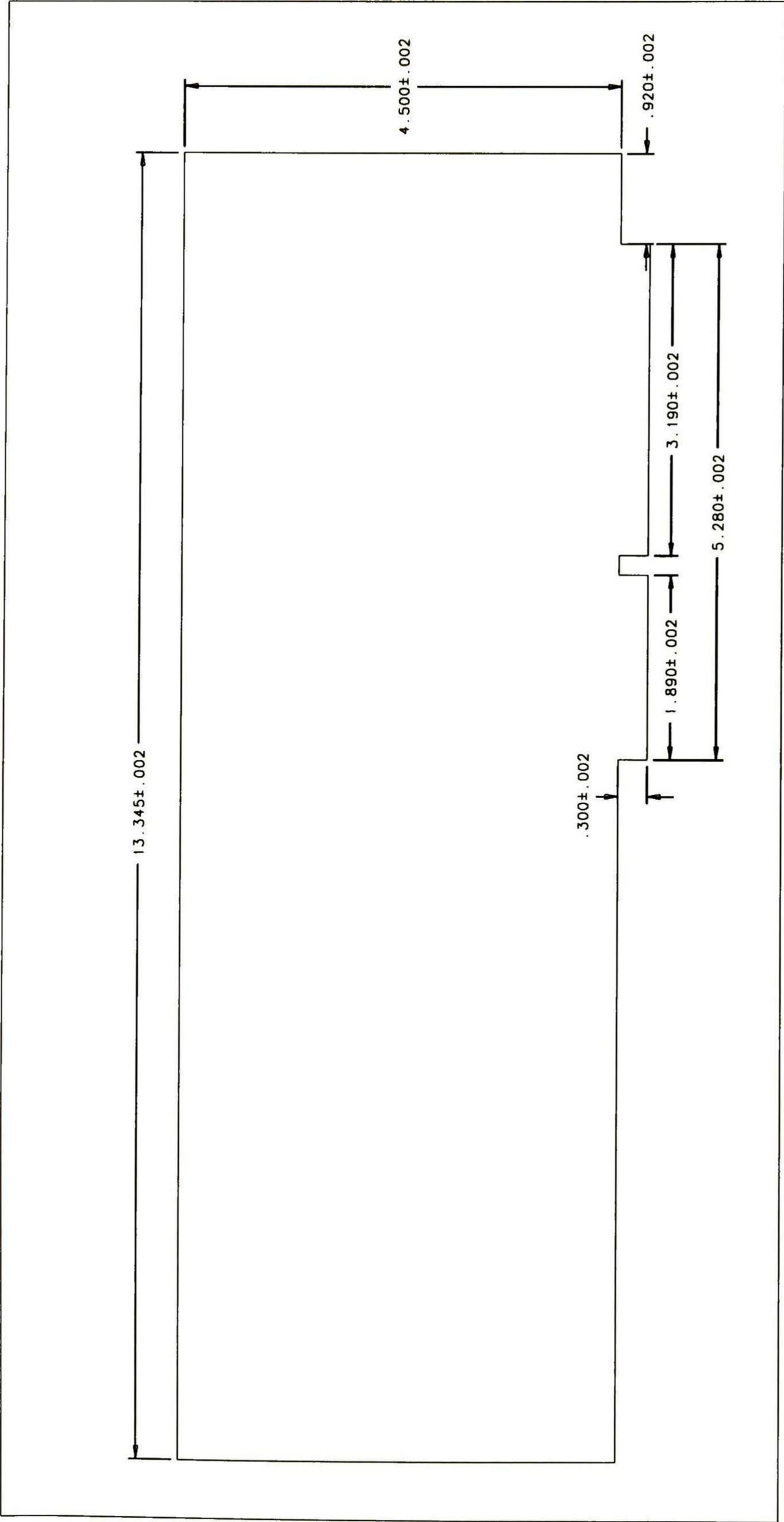
6 LAYER STACKUP

layer 1, sheet 3	solder_mask_1
layer 2, sheet 4	signal_1, pad_1
layer 3, sheet 5	dielectric
layer 4, sheet 6	power_1, power_2
layer 5, sheet 7	dielectric
layer 6, sheet 8	signal_2
	signal_3
	dielectric
	power_3, power_4
	dielectric
	signal_4, pad_2
	solder_mask_2

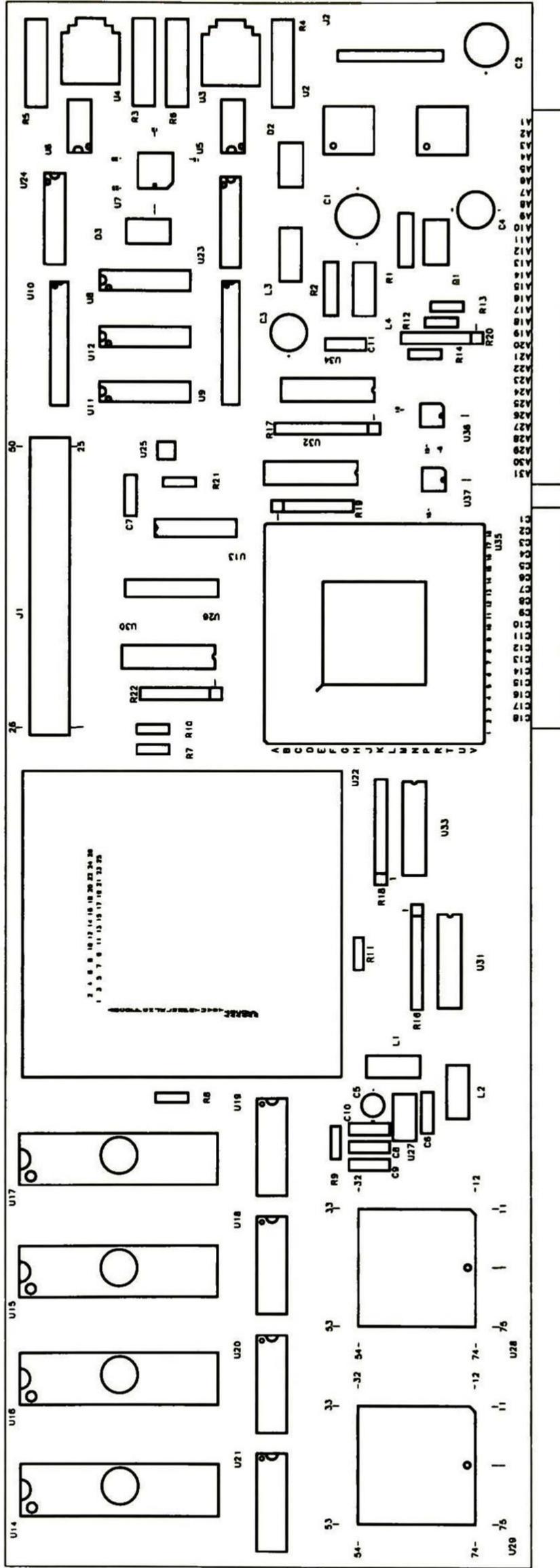
BOARD'S DRILL SCHEDULE

DRILL SYMBOL	DRILL SIZE	COUNT	PLATED	Min/Max
○	.013	791	YES	---
⊞	.019	352	YES	---
⊕	.028	191	YES	---
⊞	.038	5	YES	---
⊞	.04	538	YES	---



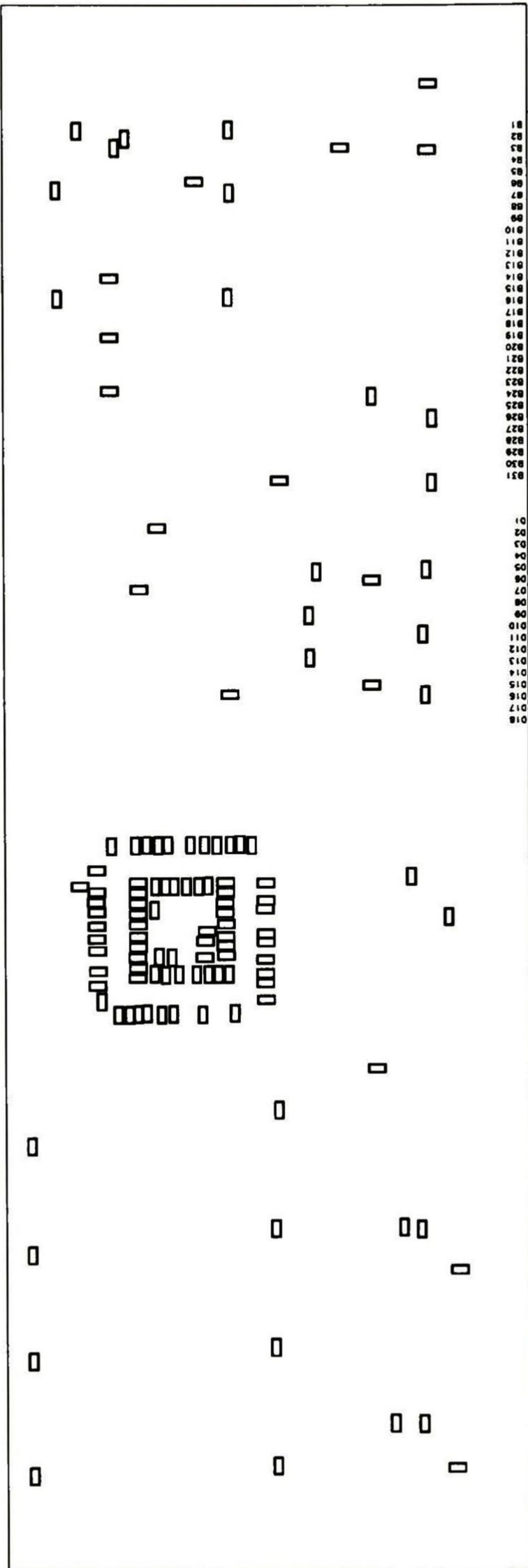


COMPONENTES EN LA CAPA SUPERIOR



COMPONENTES EN LA CAPA INFERIOR

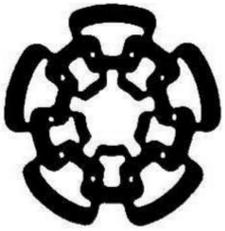
C174 C172 C170
 C155 C153 C152 C150 C148 C146 C144 C142 C140 C138 C136 C134 C132 C130 C128 C126 C124 C122 C120 C118 C116 C114 C112 C110 C108 C106 C104 C102 C100 C98 C96 C94 C92 C90 C88 C86 C84 C82 C80 C78 C76 C74 C72 C70 C68 C66 C64 C62 C60 C58 C56 C54 C52 C50 C48 C46 C44 C42 C40 C38 C36 C34 C32 C30 C28 C26 C24 C22 C20 C18 C16 C14 C12 C10 C8 C6 C4 C2 C0



Referencias

- [1] A. Kataoka, T. Moriya, and S. Hayashi, "An 8 kbit/s speech coder based on conjugate structure CELP", in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, 1993, pp. II-592-II-595.
- [2] A. Kataoka, T. Moriya, and S. Hayashi, "An 8 kbit/s Conjugate Structure CELP Speech Coder", IEEE Trans. Speech Audio Processing, no.6 pp. 401-411, Nov. 1996.
- [3] J. Adoul, P. Mabillean, M. Delprat and S. Morissette, "Fast CELP coding based on algebraic codes", on Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, 1987, pp. 1953-1956.
- [4] M.E Perkins et al. "Characterizing the Subjective Performance of the ITU-T 8Kbps Speech Coding Algorithm" Communications Magazine 1996.
- [5] R. V. Cox. "Three New Speech Coders from the ITU Cover a Range of Applications. Communication magazine 1996.
- [6] M. Schroeder and B. Atal. "Code excited linear prediction: high quality speech at low bit rates". Proc. of ICASSP, pp 937-940, 1985.
- [7] I. Gerson and M. Jasiuk. "Vector sum excited prediction (VSELP) speech coding at 4.8kbit/s". Proc. of Int. Mobile Satellite Conf., pp 678-683, Ottawa, 1990.
- [8] R. Salami, C. Lflamme, J. P Adoul, A. Kataoka, S. Hayashi, T. Moriya, C. Lamblin, B. Massalouz, S. Proust, P. Kroon and Y. Shoham. "Design and Description of CS-ACELP: A Toll Quality 8kbps Speech Coder" IEEE Transactions on Speech and Audio Processing, vol.6. No.2. March 1998.
- [9] J.D Markel and A. H. Gray. " Linear Prediction of Speech" New York: Springer-Verlag, 1976.
- [10] D.Florencio. "Investigating the use of asymmetric windows in CELP vocoders". in Proc. ICASSP, Minneapolis, MN, 1993, pp. II427-II430.
- [11] P. Kabal and R.P. Ramachandran, "The computation of line-spectral frequencies using Chebyshev polynomials ". IEEE Trans. Acoust., Speesh, Signal, Processing, vol. ASSP-27, pp. 247-254, 1979.
- [12] "Implementation and performance of an 8kbps conjugate structure CELP speech coder". In Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, 1993, pp. II-592-II-595.
- [13] ITU-T. "Recommendation G.729 –Coding of speech at 8 kbps using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)". Mar. 1996.
- [14] M.R Schroeder, B. S. Atal, and J. L. Hall. "Optimizing digital speech coders by exploiting masking properties of the human ear". J. Acoust.Soc. Amer., vol 66, pp. 1647-1652,1979.
- [15] W. B. Kleijn, D. J. Krasinki, and R. H. Ketchum. "An efficient stochastically excited linear predictive coding algorithm for high quality low bit rate transmission of speech". Speech Commun., vol.7 pp. 305-316,1988.
- [16] J.H. Chen, R. V. Cox, Y. C Lin, N. Jayant and M. J. Melchner. "A low-delay CELP coder for the CCITT 16 kbps speech coding standard". IEEE J. Select. Areas Commun., vol. 10, pp. 830-849,1992.

- [17] J. H. Chen and A. Gersho. "Adaptive postfiltering for quality enhancement of coded speech". IEEE Trans. Speech Audio Processing, vol. 3, pp. 59-71,1995.
- [18] http://wwwdsp.ucd.ie/speech/tutorial/speech_coding/
- [19] A.M. Kondoz. "Digital Speech: coding for low bit rate communication systems" Editorial Advisory Board, pp. 35-213.
- [20] A. Nejat Ince. "Digital Speech Processing: Speech Coding, Synthesis and Recognition". Editorial Klumer Academic Publishers. PP43-97.
- [21] N.S Jayant. "Digital Communications". Chapter 8: Digital Coding of Sppech.
- [22] The Frame Relay Forum. "A Discussion of Voice over Frame Relay October 1996". www.frforum.com/400.
- [23] Simon Haykin. "Adaptive Filter Theory" Prentice Hall. Information and Systems Sciences Serires. Third Edition 1996.



**CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL IPN
UNIDAD GUADALAJARA**

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, aprobó la tesis: **Diseño de una tarjeta para codificación de voz "CS-ACELP"** de la Srita. Karen Hernández Rueda el día 27 de Agosto de 1999.

Dr. Edgar Sánchez Camperos
Investigador CINVESTAV
CINVESTAV DEL IPN
Guadalajara

Dr. José Luis Leyva Montiel
Investigador CINVESTAV
CINVESTAV DEL IPN
Guadalajara

Dr. Mauricio Lara Barrón
Profesor Adjunto CINVESTAV
CINVESTAV DEL IPN
México, D.F.



CINVESTAV
BIBLIOTECA CENTRAL



SSIT000003839