

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE INGENIERÍA**

### **DISEÑO E IMPLEMENTACIÓN DE FILTROS FIR CON PARÁMETROS VARIABLES EN TIEMPO REAL UTILIZANDO LA VENTANA DE KAISER**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y TELECOMUNICACIONES**

**OLEAS VILLALBA FAUSTO GERARDO  
QUINTANA HERMOSA WILSON HERNÁN**

**DIRECTOR: ING. MARÍA SOLEDAD JIMÉNEZ  
CO DIRECTOR: ING. FERNANDO VÁSQUEZ**

**Quito, febrero de 2002**

## **AGRADECIMIENTO**

A la Ing. María Soledad Jiménez por la acertada conducción del presente Proyecto de Titulación,  
al Ing. Fernando Vásquez por la colaboración permanente, a nuestros amigos por su voz de apoyo incondicional brindada.

## DEDICATORIA

A mis padres que con su infinito amor,  
supieron guiarme en la personificación de mi vida.

A mis hermanos Mónica, Bertha y Edgar que  
comprendieron el valor de este reto y estimularon  
cada paso dado al objetivo trazado.

*Fausto*

## DEDICATORIA

*A Dios Todopoderoso por la vida, salud y fe que otorga a toda mi familia.*

*A mis padres y hermanos por el apoyo y amor incondicional.*

*A la Ing. María Soledad Jiménez por su acertada dirección.*

*Al Ing. Fernando Vásquez por su enseñanza y amistad.*

*A mis amigos,..... no será la última.*

*A todos mis compañeros que sin saber que me apoyaban,.....lo hacían.*

*Gracias Total.*

*Wilson Hernán Quintana Hermosa*

## PRESENTACIÓN

A principio de los años cincuenta el tratamiento de señales se hacía generalmente con sistemas analógicos que se realizaban con circuitos electrónicos o incluso con dispositivos mecánicos. Aunque los computadores digitales estaban ya disponibles el costo era alto y de capacidad limitada.

Las primeras aplicaciones del tratamiento digital de señales no se podían realizar en tiempo real: a menudo eran necesarios minutos o incluso horas de tiempo de computador para procesar solo algunos segundos de datos. El tratamiento de señales mediante computadoras digitales ofrecía tremendas ventajas de flexibilidad, el computador se utilizaba para simular un sistema de tratamiento de señales analógicas.

El descubrimiento y la posterior proliferación de los microprocesadores preparó el terreno para las realizaciones de bajo costo de los sistemas de tratamiento de señales en tiempo discreto, los primeros microprocesadores eran demasiado lentos para implementar en tiempo real la mayoría de los sistemas de procesamiento digital de señales.

A mediados de los ochenta la tecnología de circuitos integrados había avanzado hasta el nivel de permitir la realización de microcomputadores en punto fijo y en punto flotante con arquitecturas especialmente diseñadas para realizar algoritmos de tratamiento de señales en tiempo discreto.

El presente Proyecto de Titulación recoge todos los beneficios que brinda en la actualidad un procesador digital de señal para la implementación de filtros digitales en tiempo real; la utilización de estos filtros o el reemplazo por los filtros analógicos ya existentes en sistemas más complejos, ayudará a obtener como resultado un mayor rendimiento en el objetivo final de cada sistema que involucre el filtrado de señales en tiempo real.

## RESUMEN

La rápida evolución de los computadores y los microprocesadores digitales junto con algunos importantes desarrollos teóricos, como el algoritmo de la Transformada rápida de Fourier (FFT), fueron la causa de un importante desplazamiento hacia las tecnologías digitales naciendo así el campo del tratamiento digital de señales.

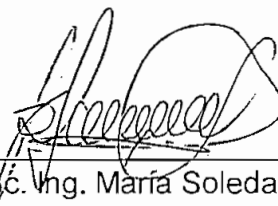
Con la llegada de los Procesadores Digitales de Señal se abre un nuevo campo para el diseñador de sistemas, ya que se puede implementar sistemas que requieren una gran capacidad de proceso con un costo mucho menor que cualquiera de las soluciones existentes.

Los métodos de diseño de filtros FIR son relativamente simples, pero no proporcionan un buen control sobre los parámetros de los mismos, por lo cual generalmente éstos son diseñados utilizando métodos de prueba y error hasta encontrar el filtro que satisfaga las características deseadas.

Un método interesante para solucionar el problema antes mencionado ha sido desarrollado por Kaiser y permite diseñar filtros FIR con características predefinidas.

## CERTIFICACIÓN

Certifico que el siguiente trabajo fue realizado por Oleas Villalba Fausto Gerardo y Quintana Hermosa Wilson Hernán, bajo mi supervisión.



---

MSc. Ing. María Soledad Jiménez  
DIRECTOR DE PROYECTO



---

Ing. Fernando Vásquez  
CO DIRECTOR DE PROYECTO

# CONTENIDO

<b>RESUMEN</b> .....	v
<b>PRESENTACIÓN</b> .....	vi
<b>LISTA DE FIGURAS</b> .....	x
<b>LISTA DE TABLAS</b> .....	xiii
<b>CAPÍTULO 1. CONCEPTOS GENERALES</b> .....	1
1.1 <b>FILTROS DIGITALES</b> .....	1
1.1.1 <b>FILTROS DIGITALES vs. FILTROS ANALÓGICOS</b> .....	3
1.1.2 <b>VENTAJAS DE LOS FILTROS DIGITALES</b> .....	5
1.1.3 <b>DESVENTAJAS DE LOS FILTROS DIGITALES</b> .....	6
1.2 <b>MICROPROCESADORES DSP</b> .....	7
1.2.1 <b>PROCESAMIENTO DIGITAL DE SEÑALES</b> .....	7
1.2.2 <b>PROCESADOR DIGITAL DE SEÑAL (DSP)</b> .....	8
1.2.2.1 <b>Arquitectura típica de un DSP</b> .....	11
1.2.2.2 <b>Áreas de desarrollo del DSP</b> .....	12
1.2.2.3 <b>Familia de Microprocesadores ADSP-2100</b> .....	13
1.3 <b>MODULO DE DESARROLLO PARA EL ADSP 2181</b> .....	17
<b>CAPÍTULO 2. DISEÑO DE FILTROS FIR</b> .....	21
2.1 <b>INTRODUCCIÓN</b> .....	21
2.1.1 <b>PARÁMETROS DE LOS FILTROS DIGITALES</b> .....	22
2.1.1.1 <b>Parámetros en el Dominio del Tiempo</b> .....	25
2.1.1.2 <b>Parámetros en el Dominio de la Frecuencia</b> .....	27
2.1.2 <b>FILTROS DIGITALES FIR e IIR</b> .....	30
2.1.3 <b>COMENTARIOS SOBRE LOS FILTROS FIR e IIR</b> .....	32
2.2 <b>MÉTODOS DE DISEÑO</b> .....	33
2.2.1 <b>DETERMINACIÓN DE LAS ESPECIFICACIONES DE UN                 FILTRO EN TIEMPO DISCRETO</b> .....	35
2.2.2 <b>MÉTODO DE VENTANAS PARA FILTROS FIR</b> .....	38



2.2.2.1	Propiedades de las ventanas comúnmente utilizadas.....	42
2.2.2.2	Incorporación de la fase lineal generalizada.....	47
2.2.2.3	Fase lineal de un filtro pasa bajo .....	49
2.2.2.4	Fase lineal de un filtro pasa alto .....	50
2.2.2.5	Fase lineal de un filtro pasa banda .....	51
2.2.2.6	Fase lineal de un filtro rechaza banda.....	51
2.3	<b>MÉTODO DE KAISER</b> .....	52
2.3.1	RELACIÓN DE UN FILTRO PASA BAJO CON LOS FILTROS FUNDAMENTALES .....	55
2.3.2	EJEMPLOS DE DISEÑO DE FILTROS DIGITALES MEDIANTE LA VENTANA DE KAISER.....	60
2.3.2.1	Diseño de un filtro pasa bajo.....	60
2.3.2.2	Diseño de un filtro pasa alto .....	64
2.3.2.3	Diseño de un filtro pasa banda .....	65
2.3.2.4	Diseño de un filtro rechaza banda.....	65
2.3.3	RELACION DE LA VENTANA DE KAISER CON OTRAS VENTANAS .....	68
<b>CAPÍTULO 3. IMPLEMENTACION DE FILTROS FIR</b> .....		70
3.1	<b>CONSIDERACIONES INICIALES</b> .....	70
3.1.1	ETAPAS EN EL DISEÑO E IMPLEMENTACIÓN DE LOS FILTROS DIGITALES .....	73
3.1.1.1	Especificación de las propiedades deseadas del Sistema .....	74
3.1.1.2	Aproximación de las propiedades mediante un sistema causal en tiempo discreto.....	86
3.1.1.3	Realización del Sistema .....	90
3.2	<b>EL MICROPROCESADOR ADSP-2181</b> .....	94
3.2.1	DESCRIPCIÓN GENERAL .....	95
3.2.2	ARQUITECTURA INTERNA .....	97
3.2.3	UNIDAD ARITMÉTICA LÓGICA.....	99

3.2.4	MULTIPLICADOR ACUMULADOR .....	103
3.2.5	UNIDAD DE DESPLAZAMIENTO .....	106
3.2.6	GENERADORES DE DIRECCIONES DE DATOS.....	111
3.2.7	SECUENCIADOR DE PROGRAMA.....	113
3.3	<b>EL EZ – KIT LITE .....</b>	<b>116</b>
3.3.1	DESCRIPCIÓN DEL HARDWARE EZ – KIT LITE .....	116
3.3.2	ESPECIFICACIONES DE LA TARJETA EZ-Lab.....	121
3.3.2.1	Conectores .....	122
3.3.2.2	Switches .....	123
3.3.2.3	Indicadores .....	123
3.3.2.4	Conectores de Expansión .....	124
3.3.3	DESCRIPCIÓN DEL SOFTWARE DEL EZ – KIT LITE..	125
3.4	<b>PROGRAMAS DESARROLLADOS PARA LOS FILTROS FIR.....</b>	<b>129</b>
 <b>CAPÍTULO 4. RESULTADOS Y APLICACIONES .....</b>		<b>135</b>
4.1	<b>EXPOSICION DE LOS RESULTADOS .....</b>	<b>135</b>
4.1.1	RESULTADO DEL FILTROS PASA BAJO.....	136
4.1.2	RESULTADO DEL FILTROS PASA ALTO .....	141
4.1.3	RESULTADO DEL FILTROS PASA BANDA .....	144
4.1.4	RESULTADO DEL FILTROS RECHAZA BANDA .....	146
4.2	<b>ANALISIS DE LOS RESULTADOS .....</b>	<b>149</b>
4.3	<b>APLICACIONES .....</b>	<b>153</b>
 <b>CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>159</b>
<b>BIBLIOGRAFIA .....</b>		<b>163</b>
<b>ANEXOS .....</b>		<b>165</b>

## LISTA DE FIGURAS

---

Figura 1.1.	Idea general de un filtro.....	2
Figura 1.2.	Proceso de filtrado de una señal analógica.....	3
Figura 1.3.	Arquitectura básica de los DSP .....	11
Figura 1.4.	Áreas de desarrollo de los DSP .....	13
Figura 1.5.	Áreas afines al Procesamiento Digital de Señales.....	14
Figura 1.6.	Diagrama de bloques del ADSP-2181 .....	18
Figura 2.1.	Parámetros de los filtros .....	23
Figura 2.2.	Parámetros para la evaluación de las características en el dominio del tiempo.....	26
Figura 2.3.	Respuestas de frecuencia comunes.....	28
Figura 2.4.	Parámetros para la evaluación del desempeño en el dominio de la frecuencia.....	29
Figura 2.5.	Respuesta impulsiva .....	31
Figura 2.6.	Sistema básico para el filtrado en tiempo discreto de señales en tiempo continuo .....	34
Figura 2.7.	Especificaciones de un filtro pasa bajo .....	36
Figura 2.8.	Funciones típicas $H_d(e^{j\theta})$ y $W(e^{j(\omega-\theta)})$ .....	40
Figura 2.9.	Módulo de la transformada de Fourier de una ventana rectangular ( $M=7$ ).....	41
Figura 2.10.	Ventanas comúnmente utilizadas .....	44
Figura 2.11.	Transformadas de Fourier con $M = 50$ de las ventanas.....	46
Figura 2.12.	Ventanas de Kaiser para $\beta = 0, 3, 6$ y $M = 20$ .....	53
Figura 2.13.	Transformación de un filtro pasa bajo a pasa alto.....	56
Figura 2.14.	Diagrama de bloques de la inversión espectral .....	57
Figura 2.15.	Parámetros de variación de un filtro pasa banda.....	58
Figura 2.16.	Parámetros de variación de un filtro rechaza banda.....	59
Figura 2.17.	Funciones de respuesta del ejemplo de diseño de un filtro pasa bajo mediante la ventana de Kaiser.....	63
Figura 3.1.	Sistema básico para el filtrado en tiempo discreto de señales en tiempo continuo .....	74

Figura 3.2.	Ingreso del factor de rizado mediante un teclado al DSP.....	79
Figura 3.3.	Rango y determinación de los parámetros de un filtro pasa bajos .....	80
Figura 3.4.	Ingreso de las frecuencias de corte, supresión y muestreo .....	
▪	mediante un teclado al DSP.....	81
Figura 3.5.	Parámetros de variación de un filtro pasa alto.....	83
Figura 3.6.	Parámetros de variación de un filtro pasa banda .....	84
Figura 3.7.	Parámetros de variación de un filtro rechaza banda .....	85
Figura 3.8.	Sistema de procesamiento en tiempo real sobre el que se implementa el filtro no recursivo .....	90
Figura 3.9.	Diagrama esquemático del sistema a implementarse.....	91
Figura 3.10.	Distribución física del teclado mediante el cual se ingresa los parámetros del filtro .....	93
Figura 3.11.	Matriz de display LCD .....	93
Figura 3.12.	Arquitectura interna del microprocesador ADSP-2100.....	95
Figura 3.13.	Arquitectura interna del ADSP-2181 .....	96
Figura 3.14.	Diagrama de bloques de la ALU .....	100
Figura 3.15.	Diagrama de bloques del MAC .....	104
Figura 3.16.	Diagrama de bloques de la unidad de desplazamiento.....	107
Figura 3.17.	Diagrama de bloques de un generador de direcciones de datos.....	112
Figura 3.18.	Secuenciador de programas ADSP-2181 .....	114
Figura 3.19.	Diagrama físico de la tarjeta EZ-Lab .....	117
Figura 3.20.	Diagrama de bloques del AD1847 .....	119
Figura 3.21.	Interfaz serial entre el ADSP-2181 y el AD1847.....	120
Figura 3.22.	Selección de las señales de entrada .....	122
Figura 3.23.	Arreglo de un conector tipo header de 50 pines.....	124
Figura 3.24.	Diagrama de Flujo del programa en Matlab 5.3.....	130
Figura 3.25.	Diagramas de flujo. ....	131
Figura 4.1.	Error porcentual de los coeficientes cuantificados y sin cuantificar para el ejemplo (Filtro pasa bajo) caso no crítico.. .....	136
Figura 4.2.	Características teóricas de un filtro pasa bajo de orden $M = 19$ (caso no crítico). .....	137
Figura 4.3.	Características experimentales de un filtro pasa bajo de orden $M = 19$ (caso no crítico). .....	138

Figura 4.4.	Características teóricas de un filtro pasa bajo de orden $M = 41$ y atenuación de 80 dB (caso crítico). .....	140
Figura 4.5.	Características experimentales de un filtro pasa bajo de orden $M = 41$ y atenuación 80dB (caso crítico). .....	141
Figura 4.6.	Error porcentual de los coeficientes cuantificados y sin cuantificar para el ejemplo (Filtro pasa alto).....	141
Figura 4.7.	Características teóricas de un filtro pasa alto de orden $M = 79$ .....	142
Figura 4.8.	Características experimentales de un filtro pasa alto de orden $M = 79$ .	143
Figura 4.9.	Error porcentual de los coeficientes cuantificados y sin cuantificar para el ejemplo (Filtro pasa banda) .....	144
Figura 4.10.	Características teóricas de un filtro pasa banda de orden $M = 37$ .....	147
Figura 4.11.	Características experimentales de un filtro pasa banda de orden $M = 37$ .....	145
Figura 4.12.	Error porcentual de los coeficientes cuantificados y sin cuantificar para el ejemplo (Filtro rechaza banda).....	147
Figura 4.13.	Características teóricas de un filtro rechaza banda de orden $M = 63$ ...	148
Figura 4.14.	Características experimentales de un filtro rechaza banda de orden $M = 63$ .....	148
Figura 4.15.	Ampliación de los lóbulos secundarios de la respuesta impulsiva a) teórica y b) experimental .....	150
Figura 4.16.	Efecto de cuantificación de la respuesta al impulso en los ceros de $H(z)$ .....	152
Figura 4.17.	Magnitud de las respuestas de baja y alta frecuencia en filtros Crossover de dos vías .....	156
Figura 4.18.	Implementación complementaria de filtros crossover de dos vías .....	157
Figura 4.19.	Sistema acústico consistente de un altoparlante y un micrófono .....	158

## LISTA DE TABLAS

---

<b>Tabla 1.1.</b> Unidades funcionales de la Arquitectura ADSP-21xx .....	15
<b>Tabla 2.1.</b> Comparación de las ventanas comúnmente utilizadas.....	47
<b>Tabla 3.1.</b> Formulas para el diseño de filtros digitales .....	87
<b>Tabla 3.2.</b> Fórmulas comunes para los filtros básicos para el diseño .....	87
<b>Tabla 3.3.</b> Respuesta de frecuencia (coeficientes del filtro).....	87
<b>Tabla 3.4.</b> Operaciones que realiza cada unidad computacional de la ADSP-2181 ....	88
<b>Tabla 3.5.</b> Descripción de las funciones de las banderas de la ALU .....	101
<b>Tabla 3.6.</b> Fuentes de los registros de entrada y salida de la ALU.....	102
<b>Tabla 3.7.</b> Entradas y Salidas del Multiplicador - Acumulador.....	106
<b>Tabla 3.8.</b> Salida del arreglo del desplazador como función del código de control C y la señal HI/LO.....	110
<b>Tabla 3.9.</b> Fuentes de entrada y salida de la sección de desplazamiento.....	111
<b>Tabla 3.10.</b> Registros I, M y L de los DAGs .....	112
<b>Tabla 3.11.</b> Configuración de JP1 y tamaño de las EPROMs .....	118
<b>Tabla 3.12.</b> Nombre de los pines de expansión .....	125

# CAPÍTULO 1

## CONCEPTOS GENERALES

---

La complejidad y la capacidad de los *chips* de tratamiento digital de señales han crecido exponencialmente desde principios de los 80's y no muestra signos de detenerse. A medida que las técnicas de integración se vayan desarrollando progresivamente, se implementarán sistemas de procesamiento de señales con bajo costo, tamaño miniaturizado y bajo consumo de potencia.

En consecuencia, la importancia del procesamiento de señales en tiempo discreto continuará creciendo casi con toda certeza y a la vez promoviendo avances revolucionarios en algunas áreas de aplicación. Un área notable es las telecomunicaciones en el que las técnicas de procesamiento digital de señales, la tecnología, avances de los microprocesadores y la transmisión por fibra óptica se combinan para cambiar la naturaleza de los sistemas de comunicaciones. Cambios similares se pueden esperar en muchas otras áreas de la tecnología.

### 1.1 FILTROS DIGITALES

Aunque el procesamiento de señales es un campo dinámico y rápidamente creciente, sus fundamentos están bien establecidos. Con un tratamiento coherente de la teoría de los sistemas lineales en tiempo discreto, *las técnicas de filtrado*, muestreo de señales y el análisis de Fourier en tiempo discreto; se puede tener el conocimiento necesario para apreciar el amplio rango de aplicaciones que se pueden ejecutar y las bases para contribuir a los futuros desarrollos en el campo de la tecnología.

En general, cualquier sistema que seleccione ciertas frecuencias de una señal con respecto a otras se denomina *filtro*. La función de un filtro es la de quitar las partes no deseadas de una señal (como por ejemplo el ruido), o extraer las partes

deseadas de la señal tales como ciertas componentes dentro de un rango de frecuencias.

En la figura 1.1 se ilustra la idea básica de un filtro.



Figura 1.1 Idea general de un filtro [15]

Hay dos clases principales de filtros: *analógicos* y *digitales*. Éstos son muy diferentes en cuanto a su aspecto físico y a la forma cómo trabajan.

Los *filtros analógicos* utilizan circuitos electrónicos analógicos formados por componentes, como: resistencias, condensadores y amplificadores operacionales necesarios para producir los requerimientos de filtrado. Éstos tipos de filtros se utilizan en aplicaciones como: reducción del ruido, perfeccionamiento de señales de video, ecualizadores gráficos en sistemas de alta fidelidad, y en muchas otras áreas.

Un *filtro digital* utiliza un procesador digital para realizar cálculos numéricos con los valores de las muestras de la señal. El procesador podría ser un computador de propósito general como por ejemplo un PC, o un procesador especializado como un DSP (*Digital Signal Processor*, Procesador Digital de Señales).

La señal de entrada analógica debe primero ser muestreada y luego digitalizada usando un ADC (*Analog to Digital Converter*, Conversor Analógico-Digital), obteniendo como resultado números binarios, los cuales representan los valores de las muestras sucesivas que corresponden a la señal de entrada, éstos valores son transferidos hacia el procesador, el cual realiza los cálculos numéricos.



Éstos cálculos típicamente involucran multiplicaciones de los valores de entrada por constantes y las sumas entre dichos productos. Al final los valores obtenidos del procesador son convertidos nuevamente en señal de forma analógica usando un DAC (*Digital to Analog Converter*, Conversor Digital-Analógico).

En un filtro digital, las señales están representadas por una secuencia de números, en lugar de valores de voltaje o corriente. En la figura 1.2 se indica el arreglo básico de dicho sistema.

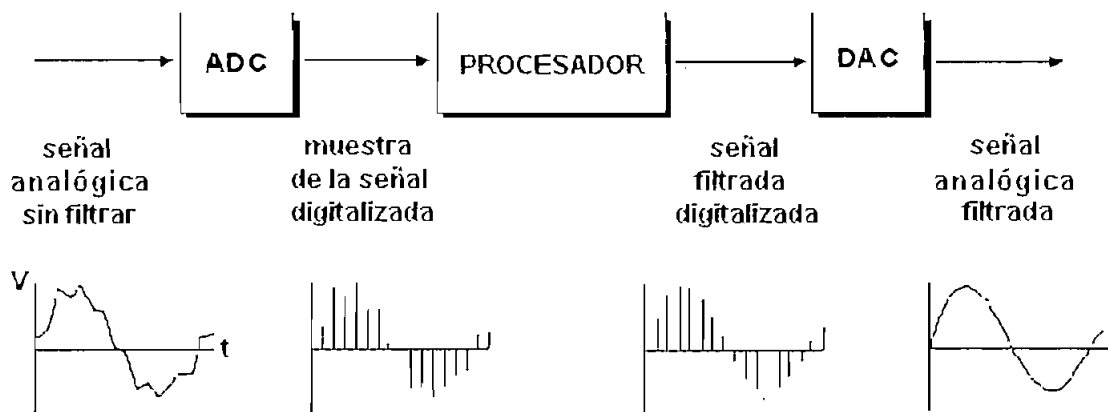


Figura 1.2 Proceso de filtrado de una señal analógica [15]

Los filtros digitales son una parte muy importante de los DSP. De hecho, su extraordinario desempeño es una de las razones más importantes por las cuales los DSP han llegado a ser tan populares.

### 1.1.1 FILTROS DIGITALES vs. FILTROS ANALÓGICOS

Los filtros digitales son utilizados para dos propósitos generales:

1. La *separación de señales* que han sido combinadas.
2. La *restauración de señales* que han sido distorsionadas de alguna manera.

Los filtros analógicos también llamados *electrónicos* pueden ser utilizados para éstos mismos propósitos, sin embargo, los filtros digitales pueden alcanzar resultados superiores.

La *separación de la señal* es utilizada cuando ésta ha sido contaminada con interferencia, ruido u otras señales. Por ejemplo el ECG (*ElectroCardioGram*, Electrocardiograma) es un aparato que sirve para medir la actividad eléctrica del corazón de un bebé mientras está en el útero de su madre, la señal obtenida probablemente podría ser influenciada por la respiración y los latidos del corazón de la madre. Mediante la utilización de un filtro se podría separar éstas señales para ser analizadas individualmente.

Se utiliza la *restauración de una señal* cuando se produce de alguna manera distorsión de la misma. Por ejemplo, en una grabación de sonido realizada con equipos ineficaces, mediante el filtrado se podría representar de mejor manera la señal. Éstos problemas podrían ser resueltos con filtros analógicos o digitales.

Los filtros analógicos son bajos en costo, rápidos, y tienen un *rango dinámico*<sup>1</sup> grande en amplitud y frecuencia. Los filtros digitales, en comparación, son inmensamente superiores en cuanto a la rapidez en la transición que pueden alcanzar. Por ejemplo un filtro digital pasa bajo puede tener una ganancia DC constante de  $1 \pm 0.0002$  a una frecuencia de 1000 hertz, y una ganancia menor que 0.0002 para frecuencias superiores a 1001 hertz. La transición ocurre dentro de tan sólo 1 hertz.

Los filtros digitales pueden alcanzar mejores tiempos de ejecución que los filtros analógicos. Esto hace una diferencia dramática en cuanto al filtrado de señales, ya que con filtros analógicos el problema radica en las limitaciones en cuanto al manejo de los elementos electrónicos, tal como la exactitud y estabilidad de las resistencias y condensadores.

---

<sup>1</sup> Un elemento puede producir un rango constante en Amplitud entre valores máximos y mínimos, al que se le denomina Rango Dinámico, este valor es medido en Decibelios

En comparación, los filtros digitales son tan eficientes que frecuentemente se ignora el rendimiento de los mismos.

### 1.1.2 VENTAJAS DE LOS FILTROS DIGITALES

Los filtros digitales tienen las siguientes ventajas frente a los filtros analógicos:

- Los filtros digitales son *fáciles de diseñar*, probar e implementar en computadoras de propósito general o en estaciones de trabajo.
- *Son programables por software*, ésta operación es determinada por un programa almacenado en la memoria del procesador. Esto significa que los filtros digitales pueden ser cambiados fácilmente sin afectar el *hardware*. Los filtros analógicos pueden ser cambiados únicamente rediseñando la circuitería del filtro.
- *No son afectados por cambios en la temperatura o humedad*, los filtros digitales son extremadamente estables con respecto a éstos factores, en cambio los filtros analógicos contienen elementos activos que son dependientes de la temperatura.
- *Son manejables con señales de baja frecuencia*, ya que los filtros digitales son implementados mediante tecnologías DSP, es decir por *software*, no existe inconveniente en trabajar con señales de baja frecuencia. Esto no ocurre con los filtros analógicos en los cuales se produce inestabilidad.
- *Pueden manejar combinaciones complejas*, gracias a la velocidad de los procesadores DSP se puede implementar por *software* filtros en paralelo o cascada afectando en forma mínima el tiempo de respuesta.
- *Alta precisión*, limitada por los errores de redondeo y truncamiento, en la aritmética empleada en el DSP.

- *Intervalo dinámico.* En filtros analógicos aparecen elementos electrónicos que limitan el rango dinámico, (por lo general la limitación es en el límite inferior), produciendo saturación con la alimentación. En cambio en los filtros digitales es fijado por el número de bits que representa la secuencia y los errores de redondeo.
- *Respuesta dinámica.* El ancho de banda del filtro digital está limitado por la *frecuencia de muestreo*<sup>1</sup>, mientras que en los filtros analógicos con componentes activos suelen estar restringidos por los amplificadores operacionales.
- Son *versátiles*, los filtros digitales tienen la habilidad de procesar señales de varias maneras, esto incluye la capacidad de adaptarse a los cambios en las características de la señal.
- Son *inmunes al ruido*, en filtros analógicos la utilización de dispositivos electrónicos incrementan la posibilidad de tener la presencia de anomalías dentro del sistema, así como es el ruido.
- Muy *bajo costo*, en un filtro analógico para poder obtener los resultados deseados se debe cambiar los dispositivos electrónicos que lo conforman, incrementando su costo.

### 1.1.3 DESVENTAJAS DE LOS FILTROS DIGITALES

Las desventajas que pueden afectar a los filtros digitales se originan básicamente de inconvenientes de aproximación que se obtengan en cada uno de los diferentes métodos matemáticos de diseño; dependiendo de cada tipo de

---

<sup>1</sup> El intervalo de tiempo T entre dos muestras sucesivas se denomina período de muestreo o intervalo de muestreo y su recíproco  $1/T = f_m$  se llama velocidad de muestreo o frecuencia de muestreo.

filtro y de las características deseadas, un determinado método se adaptará de mejor manera que otro.

Las desventajas que se generen dependerán de la elección e implementación adecuada de éstos métodos, es decir, las desventajas que acarree un determinado método no necesariamente lo ocasionará otro.

- Por ejemplo para determinados filtros no es posible aprovechar las ventajas de la Transformada Rápida de Fourier en la implementación, ya que para esto es necesario un número de puntos finitos.
- Otros filtros alcanzan una magnífica respuesta en amplitud a expensas de un comportamiento no lineal en fase.

## 1.2 MICROPROCESADORES DSP

Uno de los motivos principales de la aceptación de los *chips* DSP en la actualidad, es la gran variedad de herramientas de desarrollo proporcionados por los principales fabricantes de éstos. (*Texas Instruments, Analog Devices, Motorola, etc.*)

### 1.2.1 PROCESAMIENTO DIGITAL DE SEÑALES

El Procesamiento Digital de Señales es una de las áreas más poderosas que puede disponer la ciencia e ingeniería actualmente. Existe un amplio conjunto de aplicaciones en donde son de mucha importancia el *hardware* y el *software* del procesamiento de señales, aplicaciones que van desde sistemas militares altamente especializados hasta llegar a la simple electrónica de bajo costo. El procesamiento de señales se ha beneficiado siempre de una relación estrecha entre la teoría, las aplicaciones y el desarrollo de las tecnologías.

El creciente número de aplicaciones y la demanda de algoritmos de sofisticación van de la mano con el avance de la tecnología de dispositivos para el procesamiento de señales. Se estima que la capacidad de proceso de los microprocesadores DSP para tratamiento de señales se incrementará en un factor de 200 o más en los próximos diez años <sup>[14]</sup>.

Antes de los años 60, la tecnología para el procesamiento de señales era casi exclusiva de los elementos analógicos. Pero la rápida evolución de los computadores y los microprocesadores digitales junto con algunos importantes desarrollos teóricos como el algoritmo de la FFT<sup>1</sup> (*Fast Fourier Transform*, Transformada Rápida de Fourier) fueron la causa de un importante desplazamiento hacia las tecnologías digitales, naciendo así el campo del procesamiento digital de señales. Un aspecto importante del procesamiento digital de señales es que se basa en el procesamiento de secuencias de muestras.

### 1.2.2 PROCESADOR DIGITAL DE SEÑAL (DSP)

Actualmente los DSP se están convirtiendo en elementos muy comunes en el diseño electrónico, sustituyendo en algunas aplicaciones a los microprocesadores y microcontroladores.

Principalmente se encuentran DSP en circuitos relacionados con las telecomunicaciones, sistemas de audio y en algoritmos avanzados de control de motores. Por ejemplo, se puede encontrar DSP como integrantes de las siguientes aplicaciones:

- Tarjetas con múltiples puertos seriales en servidores para proveedores de acceso a Internet.

---

<sup>1</sup> La transformada rápida de Fourier es simplemente un algoritmo rápido para la evaluación numérica de integrales de Fourier desarrollado en los laboratorios de IBM, y su importancia radica en la rapidez de cálculo conseguida en muchos tipos de aplicaciones como: ecualización y filtrado en equipos de audio / video en tiempo real, comunicaciones, etc.

- Generadores de eco.
- Compresión de voz en telefonía móvil.
- Decodificación de canales en telefonía celular GSM<sup>1</sup> (*Global System for Mobile communication*, Sistema Global para la comunicación Móvil).
- Filtros complejos de sonido.

Los DSP son sistemas programables que permiten implementar muchos tipos de aplicaciones en función de las posibilidades del sistema y, por supuesto, de las habilidades del programador.

Desde el punto de vista de la arquitectura interna, se puede decir que un DSP es un microprocesador optimizado internamente para realizar los cálculos necesarios para implementar algoritmos de procesamiento de señal. Ésta optimización se consigue mediante algunos aspectos principales:

- Implementación de operaciones por *software*.
- Instrucciones poco comunes que ejecutan varias operaciones en un solo ciclo.
- Modos de direccionamiento especiales.
- Memoria de programa "ancha", con más de 8 bits.

La implementación de algunas operaciones mediante *software* consigue mejorar la velocidad media de cálculo, que se da en MIPS (*Millones de Instrucciones Por Segundo*); normalmente, las instrucciones que se implementan son aquellas que se usan más a menudo.

---

<sup>1</sup> Sistema Global para la comunicación Móvil (GSM). El desarrollo del GSM comenzó en 1982 cuando se formó un grupo de estudio llamado inicialmente *Group Special Mobile*. El grupo tenía que estudiar y desarrollar un sistema celular público de *Pan-European* en el rango de 900 MHz.

Aunque es posible usar diferentes tipos de microprocesadores rápidos de propósito general para procesamiento de señales, éstos no son muy utilizados ya que el diseño resulta complicado de implementarlo y muy costoso de fabricar. En contraste los DSP tienen una arquitectura que simplifica el diseño y reduce el costo de las aplicaciones.

La impresionante capacidad de cálculo de un DSP puede utilizarse también para implementar algoritmos digitales de control (por ejemplo, un controlador PID<sup>1</sup> (*Proporcional Integral Derivativo*)) y, si el costo lo permite, otras labores realizadas tradicionalmente por microprocesadores.

Las clases de algoritmos utilizados en el procesamiento de señales pueden ser mejoradas al apoyarse en una arquitectura de computador. Para realizar el procesamiento digital de señales de una manera eficaz, el microprocesador debe tener las características siguientes:

- Unidades computacionales rápidas y flexibles.
- Permitir el flujo de datos hacia y desde las unidades computacionales.
- Precisión extendida y rango dinámico en las unidades computacionales.
- Generadores de dirección duales.
- Eficiente secuenciamiento de programas y mecanismos de lazo.

---

<sup>1</sup> Es una de las formas más comúnmente utilizadas en sistemas industriales en la que combina las acciones de control y permite mejorar la calidad de la respuesta tanto transitoria como en estado estable.



### 1.2.2.1 Arquitectura típica de un DSP

Los DSP abandonan la arquitectura clásica de *Von Neumann*, en la que datos y programas están en la misma zona de memoria, y apuestan por la denominada "Arquitectura *Harvard*" como se observa en la figura 1.3.

En una arquitectura *Harvard* existen bloques de memoria físicamente separados para datos y programas. Cada uno de éstos bloques de memoria se direcciona mediante buses separados (tanto de direcciones como de datos), e incluso es posible que la memoria de datos tenga distinta longitud de palabra que la memoria de programa (como ocurre en ciertos microcontroladores).

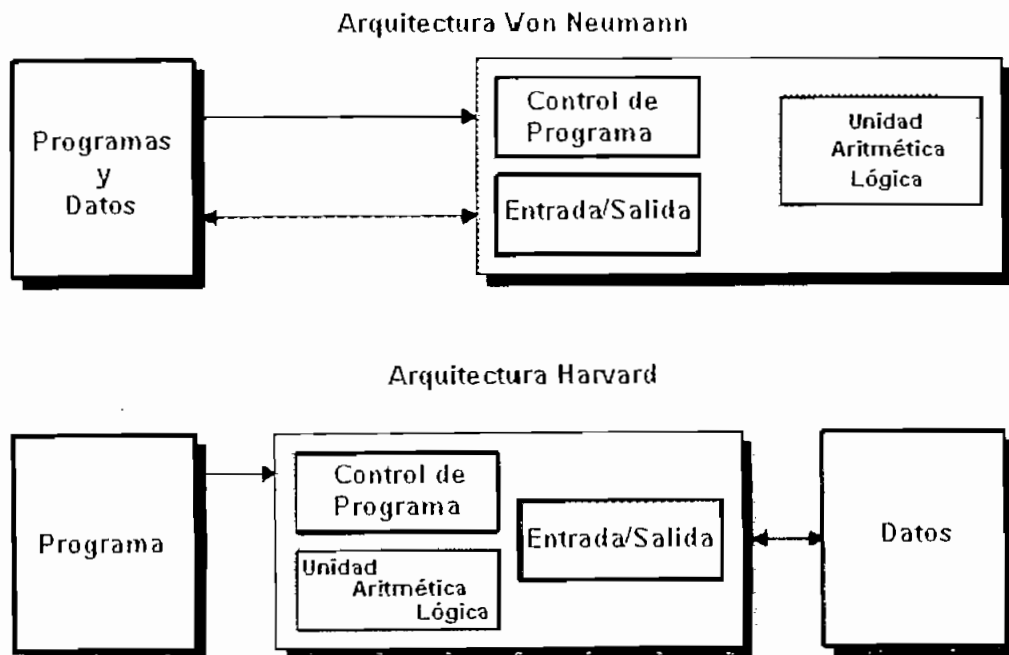


Figura 1.3 Arquitectura básica de los DSP. [15]

Con este diseño se consigue acelerar la ejecución de las instrucciones, ya que el sistema puede ejecutar simultáneamente la lectura de datos de la instrucción "n" y comenzar a decodificar la instrucción "n+1", disminuyendo el tiempo total de ejecución de cada instrucción.

Normalmente en los DSP se usa una arquitectura *Hardvard* modificada con 3 buses: uno de programa y dos de datos, lo cual permite que la Unidad Central de Procesos lea una instrucción y dos operandos a la vez ( pero no dos posiciones de memoria a la vez, para lo que hace falta una RAM (*Random Access Memory*, Memoria de Acceso Aleatorio) de doble puerto). En el proceso de señales, las operaciones con 2 operandos son muy comunes, motivo por el cual se hace ésta modificación.

También es bastante común encontrar un solapamiento entre la ROM (*Read Only Memory*, Memoria Solo de Lectura) y la RAM de datos, de modo que se puede usar parte de la ROM de programa para almacenar coeficientes y leerlos en la RAM de datos, sin tener que usar instrucciones específicas para leer los datos almacenados en ROM, como ocurre en algunos microcontroladores.

### 1.2.2.2 Áreas de desarrollo del DSP

La figura 1.4 ilustra algunas áreas de aplicación de los DSP los cuales han sufrido un continuo desarrollo, tales como: las comunicaciones, imágenes médicas, radar y sonar, reproducción de música de alta fidelidad, etc. Cada una de éstas áreas han originado una tecnología DSP con sus propios algoritmos, matemáticas y técnicas especializadas.

El Procesamiento Digital de Señales se distingue de otras áreas por el tipo de datos que utiliza: *señales*. En la mayoría de casos, éstas señales son originadas como datos censados del mundo real tales como: vibraciones sísmicas, imágenes visuales, ondas de sonido, etc.

En general el DSP son las matemáticas, los algoritmos, y las técnicas usadas para manipular éstas señales después de que han sido convertidas en un formato digital. Esto incluye una gran variedad de metas, tales como: perfeccionamiento de imágenes visuales, reconocimiento y generación de lenguaje, compresión de datos para almacenamiento y transmisión, etc.

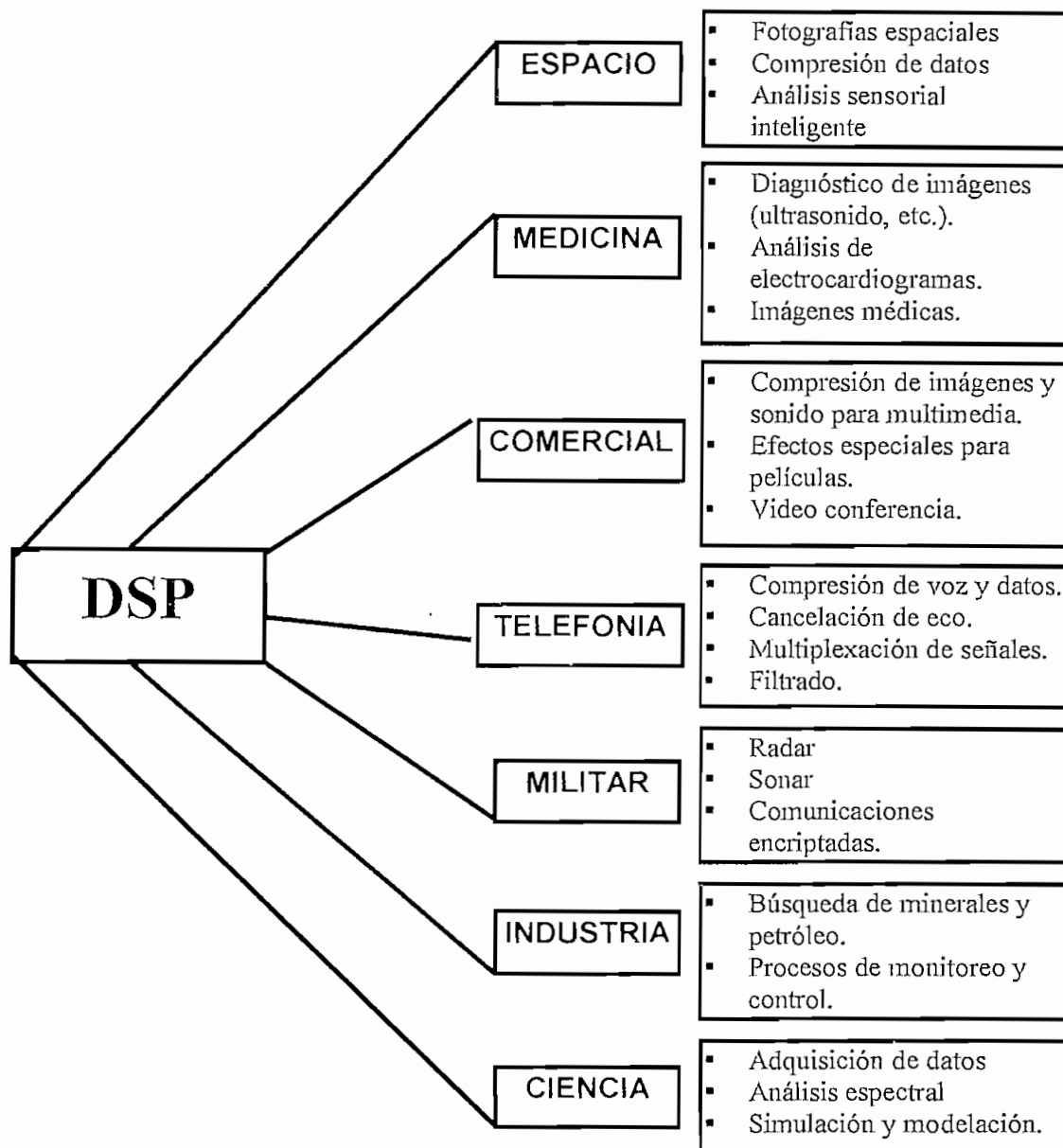


Figura 1.4 Áreas de desarrollo de los DSP [14]

En la figura 1.5 se puede observar la interrelación que existe entre el Procesamiento Digital de Señales y otras disciplinas técnicas que se encuentran bien definidas, pero juntas han permitido desarrollarse con mayor rapidez.

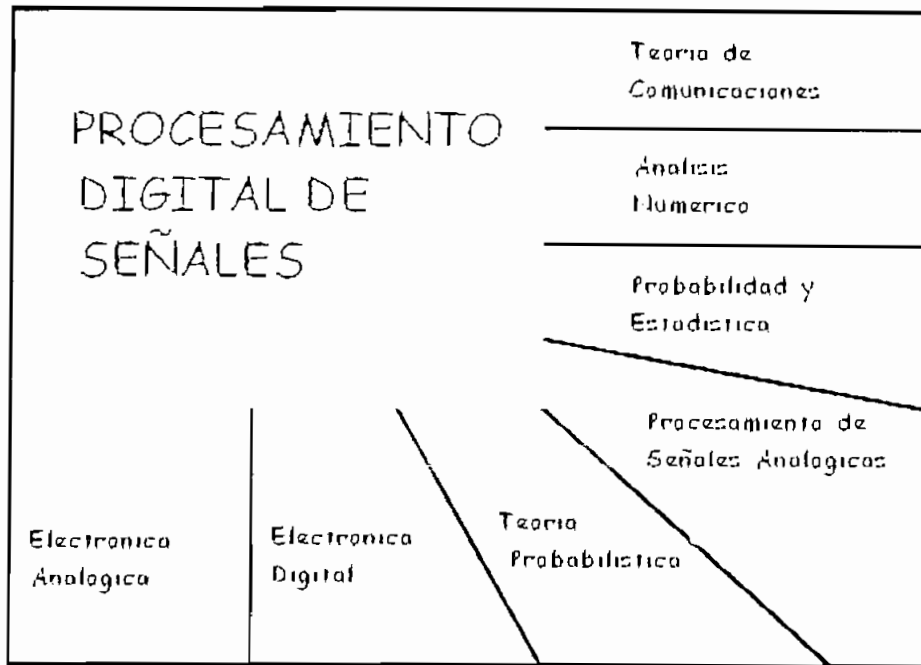


Figura 1.5 Áreas afines al Procesamiento Digital de Señales. [14]

### 1.2.2.3 Familia de Microprocesadores ADSP-2100 <sup>[1]</sup>

La familia ADSP-2100 es una colección de microprocesadores programables de un solo *chip*, que tienen en común una arquitectura base ADSP-21XX optimizada para el procesamiento digital de señales y otras aplicaciones de procesamiento numérico de alta velocidad.

Las familias de procesadores difieren principalmente en el tipo de periféricos que se aumentan en la arquitectura base. Tales como memoria, temporizador, puerto(s) serial(es), y puertos paralelos que están disponibles en los diferentes miembros de la familia. Además, los procesadores ADSP-21msp58/59 incluyen un interfaz analógico para conversión de señales en la banda de voz.

La tabla 1.1 indica las principales unidades funcionales de la arquitectura ADSP-21xx, y muestra las funciones que se incluyen en cada uno de los procesadores.

CARACTERÍSTICAS	2101	2103	2105	2115	2111	2171	2173	2181	2183	21msp58
Unidad Aritmética Lógica	+	+	+	+	+	+	+	+	+	+
Acumulador Multiplicador	+	+	+	+	+	+	+	+	+	+
<i>Shifter</i>	+	+	+	+	+	+	+	+	+	+
Generadores de Direcciones de Datos	+	+	+	+	+	+	+	+	+	+
Secuenciador de Programas	+	+	+	+	+	+	+	+	+	+
Memoria de Datos RAM	1K	1K	512	512	1K	2K	2K	16K	16K	2K
Memoria de Programa RAM	2K	2K	1K	1K	2K	2K	2K	16K	16K	2K
Temporizador	+	+	+	+	+	+	+	+	+	+
Puerto Serial 0	+	+	---	+	+	+	+	+	+	+
Puerto Serial 1	+	+	+	+	+	+	+	+	+	+
Interfaz Puerto <i>Host</i>	---	---	---	---	+	+	+	---	---	+
Puerto DMA	---	---	---	---	---	---	---	+	+	---
Interfaz Analógico	---	---	---	---	---	---	---	---	---	+
Fuente de Voltaje	7V	3.3V	5V	5V	5V	5V	3.3V	5V	3.3V	5V
Velocidad instrucciones MIPS	20	10	13.8	20	20	33	20	33	33	26

TABLA 1.1 Unidades Funcionales de la Arquitectura ADSP - 21XX [1]

- *Unidades Computacionales.* Todos los procesadores de la familia ADSP-2100 contienen tres unidades computacionales independientes:
  - Una ALU (*Arihtmetic / Logic Unit*, Unidad Aritmética Lógica).
  - Un MAC (*Multiplier / Accumulator*, Multiplicador Acumulador) y;
  - Un *Shifter* o Unidad de Desplazamiento.

Las unidades computacionales procesan datos de 16-bit directamente y también proveen soporte de *hardware* para cálculos de multiprecisión.

- *Generadores de Dirección de Datos & Secuenciador de Programa.* Los dos generadores de dirección de datos permiten al procesador establecer

direcciones simultáneas para producir operandos dobles. El secuenciador y los generadores de dirección de datos mantienen las unidades computacionales en continuo funcionamiento, maximizando de ésta manera el *throughput*<sup>1</sup>.

- *Memoria.* La familia ADSP-2100 utiliza la arquitectura modificada de *Harvard* en la que la memoria de datos almacena datos, y la memoria de programa almacena datos y programas. Toda la familia de procesadores ADSP-2100 contienen una RAM que comprende una parte del espacio para la memoria de programas y otra parte para la memoria de datos. La velocidad de la memoria permite al procesador extraer dos operandos (uno de memoria de datos y otro de memoria de programa) y una instrucción (de memoria de programa) en un solo ciclo de máquina.
- *Interfaz analógico* Los procesadores ADSP-21msp58/59 incluyen una circuitería para procesamientos analógicos y digitales de señales. Ésta circuitería incluye un conversor analógico - digital, un conversor digital - analógico, filtros analógicos y digitales, y un interfaz paralelo al centro del procesador. La arquitectura ADSP-21XX exhibe un alto grado de paralelismo, hechos a la medida de los requerimientos de los DSP. En un solo ciclo de máquina el ADSP -21XX puede:
  - Generar la siguiente dirección de programa.
  - Sacar la siguiente instrucción.
  - Actualizar uno o dos punteros de dirección de datos.
  - Ejecutar un operación.

En ese mismo ciclo los procesadores que tienen las unidades funcionales pertinentes también pueden:

---

<sup>1</sup> Se conoce como *throughput* a la velocidad de procesamiento de datos.

- a. Recibir y / o transmitir datos vía puerto(s) serial(es).
  - b. Recibir y / o transmitir datos vía el interfaz puerto *Host*.
  - c. Recibir y / o transmitir datos vía los puertos DMA. (*Direct Memory Access*, Acceso Directo a Memoria)
  - d. Recibir y / o transmitir datos vía el interfaz analógico.
- *Temporizador* Tiene un temporizador - contador programable con 8-bit el cual provee una generación de interrupciones periódicas.
  - *Puertos Seriales* Los puertos seriales (S\_PORTS) proveen un interfaz serial completo con el *hardware* para la compresión y expansión de datos. Cada S\_PORTS podría generar un reloj interno programable o aceptar un reloj externo. El SPORT0 incluye una opción de multicanal.
  - *Interfaz Host Puerto* El Interfaz *Host* - Puerto (HIP) permite la conexión directa a un procesador *Host*.
  - *Puertos DMA* (*Direct Memory Access*, Acceso Directo a Memoria). Los puertos DMA internos (IDMA) y los puertos DMA de *Byte* (BDMA) en los ADSP-2181 proveen eficientes transferencias de datos hacia y desde la memoria interna.

### 1.3 MÓDULO DE DESARROLLO PARA EL ADSP 2181 <sup>[1]</sup>

El ADSP-2181 de *Analog Devices* está sustentado por un módulo de desarrollo de bajo costo denominado *EZ-KIT Lite*. Este módulo incluye una tarjeta de evaluación (EZ-Lab) y un paquete de desarrollo de *software*.

## DESCRIPCIÓN GENERAL

El ADSP-2181 es un microprocesador creado especialmente para Procesamiento Digital de Señales y otras aplicaciones que necesiten de procesamiento numérico de alta velocidad. Fabricado con tecnología CMOS, el ADSP-2181 opera con una sola fuente de alimentación de + 5 VDC; ejecuta 33 MIPS, en el que todas las instrucciones requieren un solo ciclo de procesador.

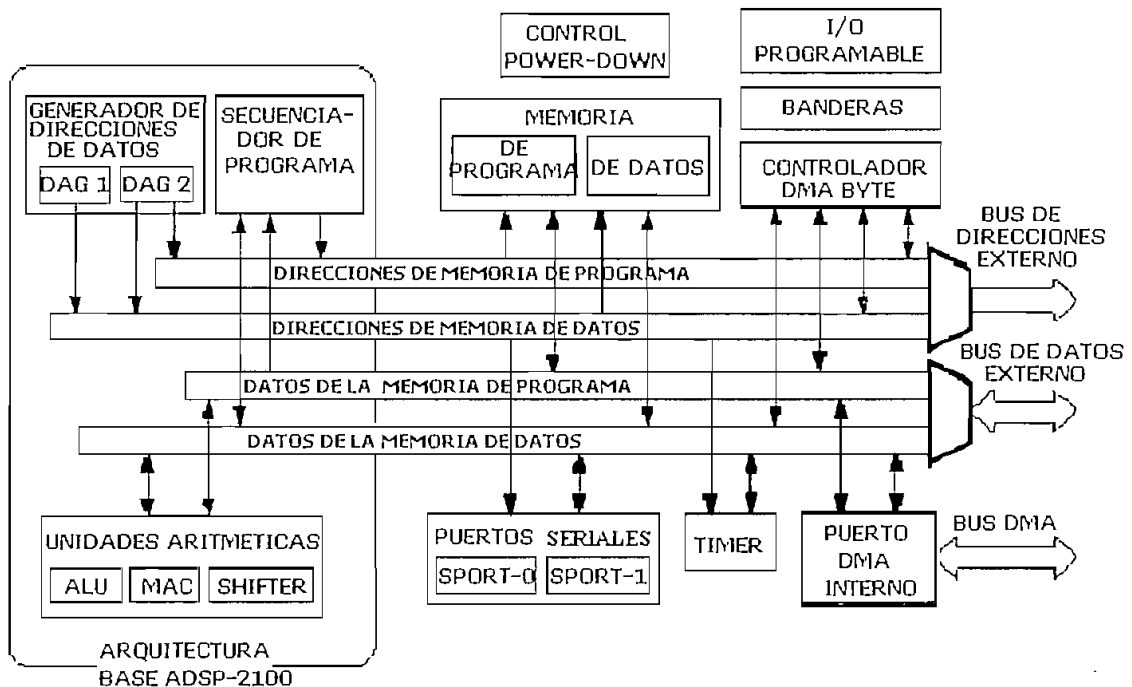


Figura 1.6 Diagrama de Bloques del ADSP-2181 [1]

El ADSP-2181 combina la arquitectura base de la familia ADSP-2100 (tres unidades computacionales, generadores de dirección de datos y secuenciador de programas) con dos puertos seriales, un puerto DMA interno de 16 bits, un puerto de byte DMA, un temporizador programable, banderas I/O, interrupciones y memoria de datos y de programas; como se ilustra en la figura 1.6



El EZ-KIT Lite es el módulo de desarrollo para el ADSP-2181, consta de un equipo de *hardware* y de *software* que ofrece un desarrollo completo para la familia ADSP-21XX. El EZ-KIT Lite incluye las siguientes características:

- Un ADSP-2181 de 33 MIPS
- Puerto de sonido estéreo AD 1847 (CODEC)
- Interfaz RS-232 para conexión con el PC.
- *Software* de Control para *Windows* 3.1
- Teclas de usuario
- EPROM de 256 Kbytes
- Conectores de expansión
- *Jumpers* configurables por el usuario
- Programas Demos DSP

Esta tarjeta puede trabajar sola o conectarse al pÓrtico RS-232 de un computador, para que un programa monitor que se ejecuta en el ADSP-2181, conjuntamente con un programa *Host* en el PC interactúen para almacenar programas de usuario y examinar los contenidos de la memoria interna del procesador.

En la memoria EPROM se graba el programa monitor, ésta memoria se encuentra alojada en un zÓcalo, la cual puede ser removida para insertar una nueva memoria que contenga códigos de programa desarrollados por el usuario.

El paquete de desarrollo de *software* proporciona el soporte necesario para el desarrollo de aplicaciones DSP. A más del programa *Host* y el programa monitor se incluyen utilitarios que son:

- Un *Assembler*, reúne los códigos fuente y los módulos de datos de la mejor manera como soporte la sintaxis de alto nivel del *set* de instrucciones.
- Un *Linker*, enlaza separadamente módulos ensamblados.
- Un *PROM Splitter*, este módulo lee el *linker* de salida y genera archivos compatibles con el programador PROM.
- Un Simulador, ejecuta un interactivo, simulador del nivel de instrucciones de la configuración de *hardware* descrita por el Sistema Constructor.

# CAPÍTULO 2

## DISEÑO DE FILTROS FIR

---

El diseño de filtros digitales es una de las áreas más importantes en el diseño electrónico. Aunque los libros de diseño de filtros analógicos lo muestran como un proceso simple, el diseño de filtros requiere de un conocimiento avanzado de matemáticas y una comprensión entera del proceso que va a ser afectado por el filtro.

Las herramientas de procesamiento digital de señales han hecho posible el reemplazo de filtros analógicos por filtros digitales en aplicaciones que requieren flexibilidad. Éstas aplicaciones incluyen audio, telecomunicaciones, control numérico de procesos, instrumentación, robótica, entre otras. Los filtros digitales son una parte muy importante de los DSP.

### 2.1. INTRODUCCIÓN

Los filtros son una clase de Sistemas LTI <sup>[5], [11] [12]</sup> (*Linear Time Invariant*, Lineales e Invariantes en el Tiempo), muy importantes. En general, cualquier sistema que modifique ciertas frecuencias con respecto a otras se denomina *filtro*.

El diseño de filtros requiere realizar las siguientes etapas:

1. Especificación de las propiedades deseadas,
2. Aproximación de las especificaciones mediante un sistema en tiempo discreto, y
3. Realización del sistema.

La primera es altamente dependiente de la aplicación y la tercera de la tecnología utilizada para la realización. En términos prácticos, el filtro deseado se realiza utilizando cómputo digital y se emplea para filtrar una señal en tiempo continuo mediante muestreo periódico seguido por una conversión analógica digital. Por este motivo se suelen denominar filtros digitales a los filtros en tiempo discreto.

### 2.1.1 PARÁMETROS DE LOS FILTROS DIGITALES

Como se muestra en la figura 2.1, el filtro lineal tiene una *respuesta impulsiva*, una *respuesta de paso* y una *respuesta de frecuencia*. Cada una de éstas respuestas contienen información completa acerca del filtro, pero de una forma diferente. Si se especifica una de las tres formas de respuesta, la otras dos son fijadas y podrían ser calculadas directamente. Todas éstas representaciones son importantes, porque describen cómo el filtro puede reaccionar bajo diferentes circunstancias.

La salida de un sistema cuando la entrada es un impulso se denomina *respuesta impulsiva*. De la misma manera, *la respuesta paso* es la salida cuando la entrada es una función paso (también llamada borde, o una respuesta de borde). Así como la función paso es la integral de la función impulso, la respuesta paso es la integral de la respuesta impulsiva. Existe dos maneras de encontrar la respuesta paso:

1. Alimentando con una forma de onda paso al filtro
2. Integrando la respuesta impulsiva

La respuesta de frecuencia puede ser encontrada tomando la DFT<sup>1</sup> (*Discrete Fourier Transform*, Transformada Discreta de Fourier) de la respuesta impulsiva

---

<sup>1</sup> DFT. La Transformada Discreta de Fourier es una secuencia, no una función de una variable continua y corresponde a muestras equiespaciadas en frecuencia de la Transformada de Fourier de la señal.

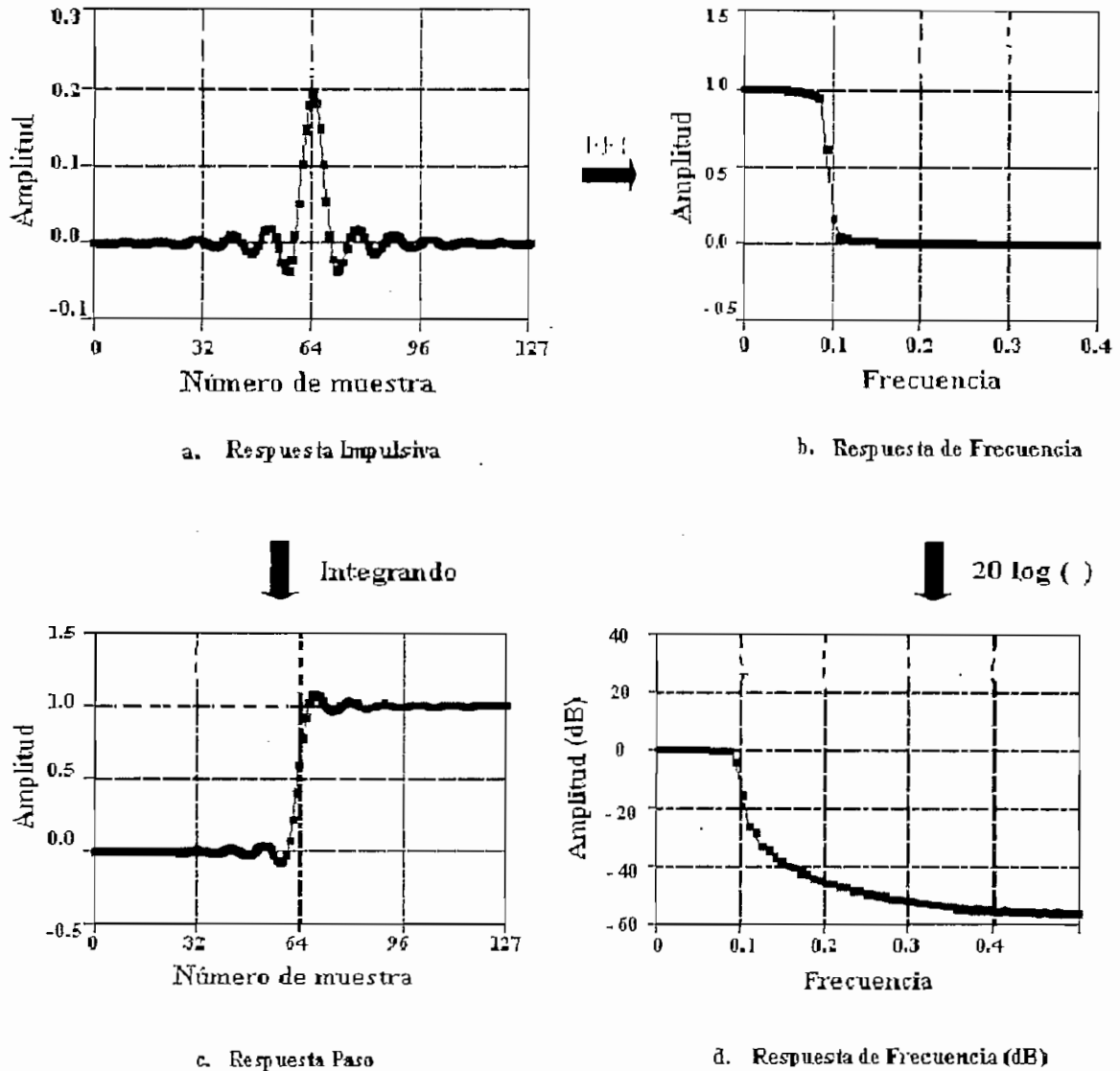


Figura 2.1. Parámetros de los filtros. [14]

Se puede dibujar la respuesta de frecuencia (amplitud) en escala lineal o en escala logarítmica (decibelios), estas respuestas de frecuencia se pueden observar en la figura 2.1 (b) y (d) respectivamente. La escala lineal es mejor para indicar el rizado de la banda de paso y el factor de *roll-off*<sup>1</sup>, mientras que la escala en decibelios es mejor para indicar la atenuación de la banda de supresión

<sup>1</sup> El factor de *roll-off* es una característica del filtro que indica como es la caída de la banda de transición. Permite evaluar el desempeño de un filtro digital en el dominio de la frecuencia.

(parámetros que serán detallados en la sección 2.2.1).

La palabra: *dominio*, es un término extensamente usado en Procesamiento de Señales. Por ejemplo una señal que use el tiempo como variable independiente, se dice estar en el *dominio del tiempo*. De la misma manera la señal que utilice la *frecuencia* como variable independiente, resulta estar en el *dominio de la frecuencia*. Igualmente, las señales que usen la *distancia* como parámetro independiente están en el *dominio espacial* (la distancia es una medida del espacio). El tipo de parámetro en el eje horizontal es el dominio de la señal.

Las señales de entrada y de salida de un filtro están en el dominio del tiempo. Esto es debido a que las señales son usualmente originadas al sacar una muestra o *sampling*<sup>1</sup> a intervalos iguales de tiempo. Ésta no es la única forma de *sampling*. Otra forma común es muestreando a intervalos iguales en espacio. Muchos otros dominios son posibles; sin embargo, el tiempo y el espacio son los más comunes.

Las maneras más comunes de contener la información en señales son:

- La información representada en el dominio del tiempo; y
- La información representada en el dominio de la frecuencia.

La información representada en el dominio del tiempo describe cuándo algo ocurre y cuál es la amplitud de ese suceso. Por ejemplo si se analizara la emisión de luz del sol, cada muestra de la señal indica que está sucediendo en ese instante, y el nivel del evento. Si una señal luminosa ocurre, la señal directamente provee la información en el tiempo de lo que ocurrió, su duración, el desarrollo sobre el tiempo, etc.

---

<sup>1</sup> **Teorema de *Sampling* o del Muestreo:** Si una señal continua compuesta de frecuencias menores que  $f$  son muestreadas a frecuencias  $2f$  o mayores, toda la información contenida en la señal continua se encontrará en la señal muestreada. Frecuentemente es llamado Teorema de *Shannon* o Teorema de *Nyquist*.

Cada muestra contiene información que es interpretada sin referencia de alguna otra muestra. Aun si se tiene una sola muestra de la señal, se puede saber algo acerca de esa medida.

La información representada en el dominio de la frecuencia es más indirecta. Por ejemplo un vaso del vino golpeado con la uña podría vibrar, produciendo un zumbido; el péndulo de un reloj de caja gira de un lado a otro; las estrellas y los planetas giran en su propio eje. Midiendo la frecuencia, fase, y amplitud de éstos movimientos periódicos, se podría tener información de sistemas que produzcan este tipo de movimiento. Si se toma una muestra de sonido que produjo el golpe a un vaso de vino. La frecuencia fundamental y los armónicos de la vibración periódica pueden indicar la masa y la elasticidad del material.

La respuesta paso describe cómo la información representada en el dominio del tiempo está siendo modificada por el sistema. En contraste, la respuesta de frecuencia muestra cómo la información representada en el dominio de la frecuencia está siendo cambiada. Éstas distinciones son completamente críticas en el diseño de filtros debido a que no es posible optimizar un filtro para ambas aplicaciones. Un buen rendimiento en el dominio del tiempo produce un pobre rendimiento en el dominio de la frecuencia, y viceversa. Al diseñar un filtro para eliminar el ruido de una señal ECG (información representada en el dominio del tiempo), la respuesta de paso es el parámetro importante, y la respuesta de frecuencia es de poco interés. Al diseñar un filtro digital para un audífono (con la información en el dominio de la frecuencia), la respuesta de frecuencia es muy importante, mientras que la respuesta de paso no importa.

#### **2.1.1.1 Parámetros en el Dominio del Tiempo**

Las respuestas paso, impulso y de frecuencia contienen idéntica información, sólo que en arreglos diferentes. La respuesta paso es útil en análisis en el dominio del tiempo porque iguala la forma de ver como la información está contenida en las señales.

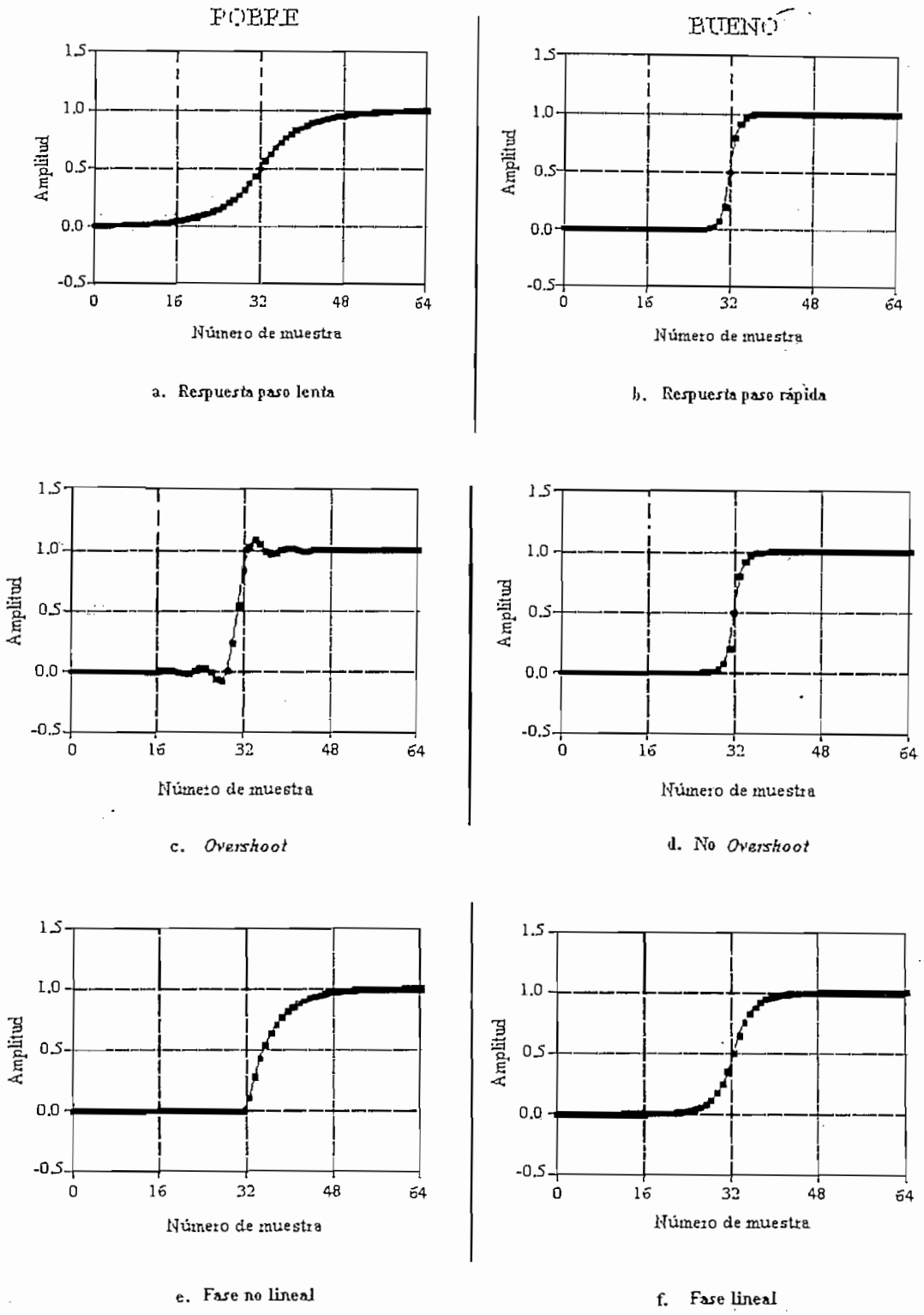


Figura 2.2. Parámetros para la evaluación de las características en el dominio del tiempo.[14]



Por ejemplo si se desea analizar una señal de origen desconocido la primera cosa que se debe hacer es dividir la señal en regiones de características similares. Algunas de las regiones podrían ser uniformes; otras tendrían grandes picos de amplitud o contendrían ruido. Ésta segmentación se logra por identificación de los puntos que separan las regiones. La función paso es la manera más pura de representar una división entre dos regiones distintas. Esto puede indicar cuando un evento empieza, o cuando un evento termina.

Los parámetros de la respuesta paso que son importantes en el diseño de filtros se indican en la figura 2.2. Para distinguir los eventos en una señal, la duración de la respuesta paso debe ser más corta que los espacios de los eventos. Es por esto que la respuesta paso debe ser lo más rápida como sea posible, a ésta velocidad de transición se conoce como *risetime*, este parámetro se ilustra en la figura (a) y (b).

Las figuras (c) y (d) hacen referencia a un parámetro que mide el nivel de rizado u *overshoot* en la respuesta paso. Se debe generalmente eliminar el *overshoot* ya que la amplitud de las muestras cambian en la señal; esto es una distorsión básica de la información contenida en el dominio del tiempo.

Finalmente, a menudo se desea que la mitad superior de la respuesta paso sea simétrica con la mitad inferior, como ilustra en los literales (e) y (f). Esta simetría es llamada **fase lineal**, debido a que la forma de la respuesta de fase es una línea recta.

### 2.1.1.2 Parámetros del Dominio de la Frecuencia

En el diseño de filtros selectivos en frecuencia, las características deseadas de un filtro se especifican como función de la variable frecuencia " $\omega$ " denominada respuesta de frecuencia. Se denomina respuesta de frecuencia a  $H(\omega)$  dado que define la respuesta del sistema en el dominio de la frecuencia, análogamente,  $|H(\omega)|$  es la respuesta de amplitud y  $\Theta(\omega)$  es la respuesta de fase del sistema.

La Figura 2.3 indica las cuatro respuestas de frecuencia básicas. El propósito de éstos filtros es permitir pasar algunas frecuencias sin alterarlas, mientras se bloquean completamente otras.

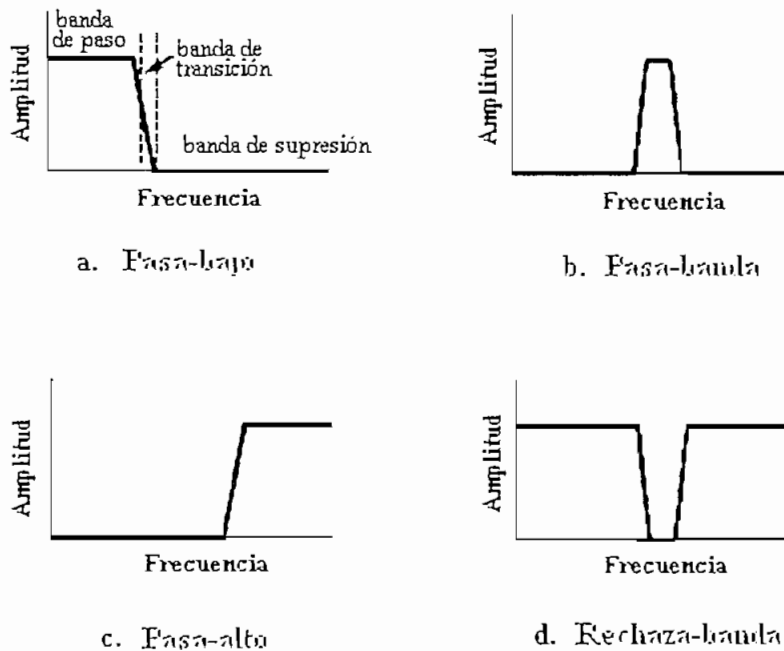


Figura 2.3 Respuestas de frecuencia comunes. [14]

La *banda de paso* se refiere a las frecuencias que permite pasar el filtro, mientras que la *banda de supresión*, *eliminada* o *de parada* contiene las frecuencias que se bloquean o retienen. En medio de estas dos bandas se encuentra la *banda de transición*. Un factor de *roll-off* rápido significa que la banda de transición es muy estrecha. El punto que separa la banda de paso y la banda de transición se llama *cut off frequency*, *frecuencia de corte*.

En el diseño de filtros analógicos la frecuencia de corte, usualmente se define en donde la amplitud se reduce a 0.707 ( -3dB ). Los filtros digitales son los menos estandarizados, y es común ver niveles de amplitud del 99%, 90%, 70.7%, y 50% definidos para frecuencias de corte<sup>[14]</sup>.

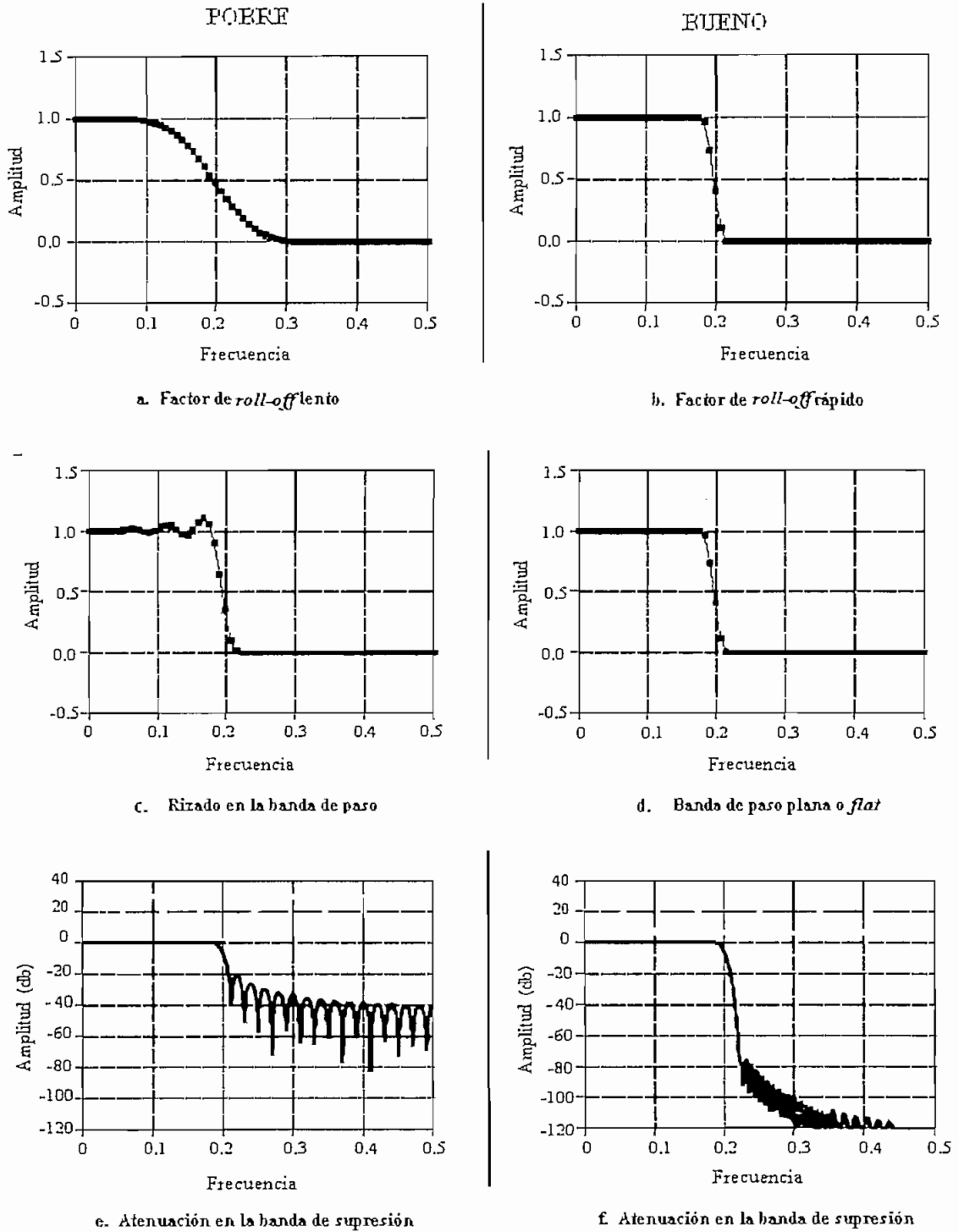


Figura 2.4 Parámetros para la evaluación del desempeño en el dominio de la frecuencia. [14]

La figura 2.4 indica tres parámetros que miden el rendimiento de un filtro en el dominio de la frecuencia. Al separar en cortos espacios de frecuencias, el filtro

debería tener un rápido *roll-off*, como se ilustra en (a) y (b); no debe haber ninguna ondulación en la banda de paso, como muestra en (c) y (d); es necesario tener una buena atenuación en la banda de supresión, como se observa en (e) y (f).

### 2.1.2 FILTROS DIGITALES FIR e IIR

La mejor manera de implementar un filtro digital es *convolucionando* <sup>[5, 7]</sup> la señal de entrada con la respuesta impulsiva del filtro. Cuando la respuesta impulsiva es usada de ésta manera, el diseño de los filtros tiene un nombre especial: **Filtros de Kernel**. Otra manera de diseñar filtros digitales, es mediante **Recursión**. <sup>[5, 7]</sup>

Para encontrar la respuesta impulsiva de un filtro recursivo, simplemente se alimenta de un impulso en la entrada. Las respuestas impulsivas de filtros recursivos están compuestas por sinusoides que decaen exponencialmente en amplitud. En principio, esto hace que su respuesta impulsiva sea infinitamente larga.

Sin embargo, la amplitud eventualmente cae por debajo del ruido *round-off*<sup>1</sup> del sistema, y las muestras que quedan pueden ser ignoradas. A causa de ésta característica, los Filtros Recursivos también son llamados **Filtros IIR (Infinite Impulse Response, Respuesta Impulsiva Infinita)**. En comparación, los filtros que son realizados por convolución son llamados **Filtros FIR (Finite Impulse Response, Respuesta Impulsiva Finita)**.

Un filtro digital es un sistema LTI, el cual pueden ser clasificado en Sistemas FIR e IIR como se ilustra en la figura 2.5, dependiendo si su respuesta impulsiva es finita o infinita.

---

<sup>1</sup> El Ruido *Round Off* es el error causado por redondeo de los cálculos matemáticos para los niveles de cuantización más cercanos.

Los filtros FIR también llamados *no recursivos* o *de convolución*, tienen una respuesta impulsiva  $h[n]$  que se extiende sólo sobre un intervalo de tiempo finito, es decir  $0 \leq n \leq M$ , y es igual a cero para otros valores.

$$\{h_0, h_1, h_2, \dots, h_M, 0, 0, 0, 0, \dots\}$$

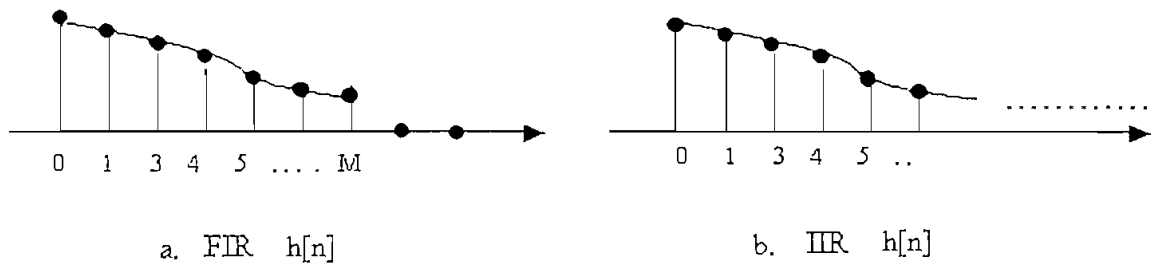


Figura 2.5 Respuesta impulsiva (a) Sistema FIR y (b) Sistema IIR [7]

donde  $M$  indica el *orden del filtro*.

Los coeficientes de la respuesta impulsiva  $h[n]$  son conocidos como *coeficientes del filtro* o *pesos del filtro*.

La salida de un filtro puede expresarse como una convolución finita, es decir:

$$y(n) = \sum_{m=0}^M h(m)x(n-m) \quad (2.1a)$$

$$y(n) = h_0 x(n) + h_1 x(n-1) + \dots + h_M x(n-M) \quad (2.1b)$$

donde  $x$  representa la secuencia de entrada a ser filtrada,  $h$  son los coeficientes del filtro FIR, e  $y$  representa la secuencia de salida filtrada.

Los filtros IIR tienen una respuesta impulsiva  $h[n]$  de duración infinita, definida sobre el intervalo infinito  $0 \leq n \leq \infty$ . La ecuación de un filtro IIR se representa por:

$$y(n) = \sum_{m=0}^{\infty} h(m)x(n-m) \quad (2.2)$$

### 2.1.3 COMENTARIOS SOBRE LOS FILTROS FIR E IIR

La elección entre filtros FIR e IIR depende de la importancia que tengan para el problema de diseño las ventajas de cada tipo. Por ejemplo, los Filtros IIR, tiene la ventaja de que se puede diseñar una gran variedad de filtros selectivos en frecuencia utilizando fórmulas cerradas; es decir una vez que se ha especificado el problema en términos que sean apropiados para un determinado método de aproximación, se calcula el orden del filtro que cumplirá las especificaciones y se obtienen los coeficientes (o los polos y ceros) del filtro mediante sustitución directa en un sistema de ecuaciones de diseño. Ésta simplicidad de procedimiento de diseño hace que se puedan diseñar filtros IIR de forma manual si fuera necesario, y producir programas de computador no iterativos.

Éstos métodos se limitan a filtros selectivos en frecuencia y solo se permite especificar el módulo de la respuesta en frecuencia. Si se desean otras formas del módulo, o es necesario aproximar una respuesta de fase o retardo de grupo predeterminado, se requiere un procedimiento algorítmico.

Por el contrario, los filtro FIR pueden tener una fase lineal (generalizada) de forma precisa. Sin embargo, no existen ecuaciones de diseño en forma cerrada para filtros FIR. Aunque el método de ventanas es sencillo de aplicar, puede ser necesaria alguna iteración para cumplir unas determinadas especificaciones. Además, el método de ventanas y la mayor parte de los procedimientos algorítmicos permiten la posibilidad de aproximar características de respuesta en frecuencia arbitrarias, con solo un poco más de dificultad de la necesaria para diseñar filtros pasa bajo.

Adicionalmente el problema de diseño de filtros FIR está mucho más controlado que el diseño de filtros IIR, debido a la existencia de teoremas de optimización para filtros FIR aplicable en un amplio rango de situaciones prácticas.

Las cuestiones económicas deben ser también tenidas en cuenta al realizar filtros, éstas cuestiones se miden generalmente en términos de complejidad del *hardware*, área del *chip* o velocidad computacional. Éstos factores están relacionados mas o menos directamente con el orden del filtro necesario para cumplir una determinada especificación.

Dejando de lado las consideraciones sobre la fase, generalmente es cierto que una determinada especificación de la respuesta de amplitud se puede concebir de forma más eficiente empleando un filtro IIR. Sin embargo, en muchos casos la fase lineal que se puede conseguir utilizando un filtro FIR, puede compensar el costo adicional.

Los filtros FIR pueden ser tan eficientes como los IIR, además existen microcomputadoras orientadas al tratamiento digital de señales cuyas capacidades aritméticas están diseñadas para acumular sumas de productos como los que aparecen en los filtros FIR.

Por tanto, hay varios compromisos a considerar en el diseño de un filtro. En definitiva, la elección final se basará muy a menudo en juicios de Ingeniería sobre la formulación de las especificaciones, el método de realización del filtro, y los procedimientos computacionales y *software* disponible para realizar el diseño.

## 2.2 MÉTODOS DE DISEÑO

El problema del diseño de filtros digitales es el de construir la función de transferencia de un filtro que cumpla con especificaciones que garanticen un buen desempeño en el dominio de la frecuencia.

La entrada para un método de diseño de un filtro es el establecimiento de especificaciones deseadas; y, la salida es el vector formado por los coeficientes de respuesta de la función impulsiva finita  $h = [h_0, h_1, \dots, h_{N-1}]$  en el caso de filtros FIR o los coeficientes de los vectores del numerador y denominador  $b = [b_0, b_1, \dots, b_M]$ ,  $a = [1, a_1, \dots, a_M]$  en el caso de filtros IIR.

Las dos principales ventajas de los filtros FIR son la fase lineal y la garantía de estabilidad debido a la ausencia de polos. Su desventaja potencial es que requiere de sus especificaciones bien definidas que pueden afectar la longitud de los filtros, consecuentemente incrementando su costo computacional.

Las principales ventajas de los filtros IIR son su bajo costo computacional y su implementación eficiente en cascada de secciones de segundo orden. Su principal desventaja es su potencial inestabilidad introducida cuando la cuantización de los coeficientes desplaza los polos fuera del círculo unitario. Para filtros IIR, la fase lineal no puede ser realizada exactamente sobre el intervalo total de *Nyquist*, pero esto puede ser logrado aproximadamente sobre la banda de paso relevante del filtro, por ejemplo, usando diseños de filtros de *Bessel*.

Cuando se utiliza un filtro en tiempo discreto para realizar el tratamiento en tiempo discreto de señales en tiempo continuo empleando una configuración como la que se indica en la figura.2.6, tanto las especificaciones del filtro en tiempo discreto como en tiempo continuo se suelen dar en el dominio de la frecuencia.

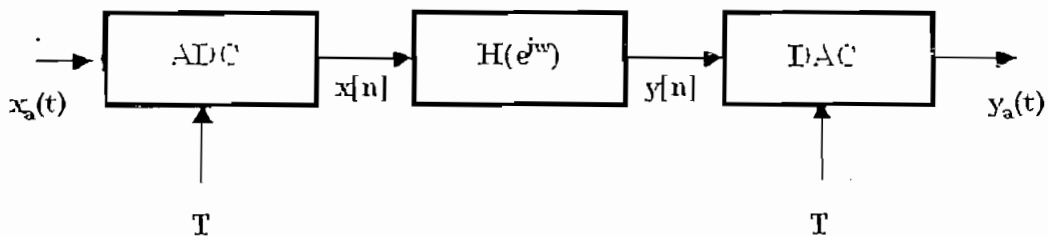


Figura 2.6 Sistema básico para el filtrado en tiempo discreto de señales en tiempo continuo.



Esto es particularmente cierto para el caso de filtros selectivos en frecuencia como los filtros pasa bajo, pasa banda y pasa alto. Si se usa un sistema lineal e invariante en el tiempo como el de la figura 2.6, el sistema completo se comporta como un sistema en tiempo continuo lineal e invariante con el tiempo cuya respuesta en frecuencia es:

$$H_{eff}(j\Omega) = \begin{cases} H(e^{j\Omega T}) & |\Omega| < \pi/T \\ 0 & |\Omega| > \pi/T \end{cases} \quad (2.3)$$

Para convertir las especificaciones del filtro en tiempo continuo en especificaciones del filtro en tiempo discreto, se utiliza la relación  $\omega = \Omega T$  donde  $\Omega$  es la frecuencia normalizada. Es decir,  $H(e^{j\omega})$  se especifica sólo en un periodo mediante la ecuación:

$$H(e^{j\omega}) = H_{eff}(j\frac{\omega}{T}), \quad |\omega| < \pi \quad (2.4)$$

donde:

$\omega$  = Variable de frecuencia.

$T$  = Período de muestreo

### 2.2.1 DETERMINACIÓN DE LAS ESPECIFICACIONES DE UN FILTRO EN TIEMPO DISCRETO

En un filtro pasa bajo en tiempo discreto, el cual se va a utilizar para filtrar una señal en tiempo continuo por medio de la configuración básica de la figura 2.6, se busca que el sistema completo de la figura tenga las siguientes propiedades para una determinada frecuencia de muestreo  $f_m$ . (El periodo de muestreo es  $T = 1 / f_m$ ).

1. La ganancia  $|H_{eff}(j\Omega)|$  debe diferir de la unidad en un factor de  $\pm \delta_1$  en la banda  $0 \leq \Omega \leq \omega_p$ ; en donde  $\delta_1$  es el factor de rizado en la banda de paso y  $\omega_p$  es la frecuencia de paso.
2. La ganancia no debe ser mayor que  $\delta_2$  en la banda de frecuencias  $\omega_s \leq \Omega$ ; donde  $\delta_2$  es el factor de rizado en la banda de supresión y  $\omega_s$  es la frecuencia de supresión.

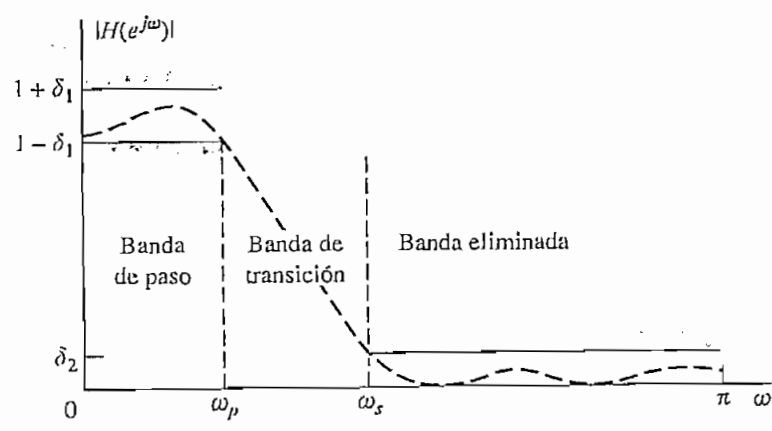
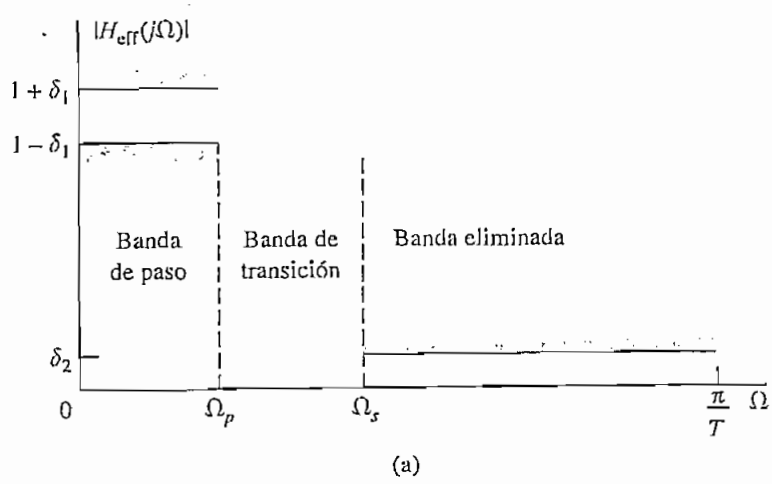


Figura 2.7 Especificaciones de un filtro pasa bajo [5]

- a) Especificaciones de la respuesta de frecuencia del sistema de la figura 2.6.
- b) Especificaciones del filtro discreto de la figura 2.6.

Por tanto en este caso, la ganancia ideal en la banda de paso es la unidad. La ganancia en la banda de paso varía entre  $(1+\delta_1)$  y  $(1-\delta_1)$ , y la ganancia en la

banda de supresión varía entre 0 y  $\delta_2$ . Es habitual expresar las máximas ganancias en la banda de paso y en la banda de supresión en decibelios.

- o Ganancia ideal de la banda de paso en decibelios =  $20 \log_{10}(1) = 0$  dB
- o Máxima ganancia de la banda de paso en decibelios =  $20 \log_{10}(1+\delta_1)$
- o Máxima ganancia de la banda eliminada en decibelios =  $20 \log_{10}(\delta_2)$ .

Para una frecuencia de muestreo  $f_m$ , la ganancia del sistema completo es cero por encima de  $\Omega = 2\pi(f_m / 2)$ .

La Figura 2.7(b) muestra el esquema de tolerancias del filtro en tiempo discreto. Es similar al de la Figura 2.7(a), salvo porque se dibuja en función de la frecuencia normalizada ( $\omega = \Omega T$ ), y sólo es necesario dibujarlo en el intervalo  $0 \leq \omega \leq \pi$ , ya que el resto se puede deducir de las propiedades de simetría (suponiendo que  $h[n]$  es real) y de la periodicidad de  $H(e^{j\omega})$ . A partir de la Ecuación (2.4) se puede deducir que en la banda de paso el módulo de la respuesta en frecuencia debe aproximarse a la unidad con un error menor que  $\pm \delta_1$ , es decir:

$$(1 - \delta_1) \leq |H(e^{j\omega})| \leq (1 + \delta_1), \quad |\omega| \leq \omega_p \quad (2.5)$$

siendo  $\delta_1$  y  $\omega_p = 2\pi \cdot f_p \cdot f_m$  radianes. La otra banda de aproximación es la banda de supresión en la que la respuesta en amplitud se debe aproximar a cero con un error menor que  $\delta_2$ ; es decir:

$$|H(e^{j\omega})| \leq \delta_2, \quad \omega_s \leq |\omega| \leq \pi \quad (2.6)$$

$\delta_2$  y  $\omega_s = 2\pi \cdot f_s \cdot f_m$  radianes. La frecuencia de corte de la banda de paso  $\omega_p$  y la frecuencia de corte de la banda eliminada  $\omega_s$ , se dan en unidades de frecuencia

angular normalizada, o equivalentemente en unidades de fase en el plano  $z$ . Para aproximarse de ésta forma al filtro pasa bajo ideal con un sistema realizable que proporciona una banda de transición no nula ( $\omega_s - \omega_p$ ) en la que el módulo de la respuesta varíe suavemente de la banda de paso a la banda de supresión. La curva discontinua de la Figura 2.6(b) es la respuesta de amplitud de un sistema que cumple las especificaciones dadas.

### 2.2.2 MÉTODO DE VENTANAS PARA FILTROS FIR

Los filtros FIR están restringidos casi totalmente a realizaciones en tiempo discreto. En consecuencia, las técnicas de diseño de filtros FIR se basan en aproximar directamente la respuesta de frecuencia deseada del sistema en tiempo discreto; la mayoría de técnicas de aproximación de la respuesta de amplitud de un sistema FIR asumen una restricción de fase lineal.

El método más simple de diseño de filtros FIR se denomina *método de ventanas*.

Este método empieza generalmente con una respuesta de frecuencia deseada ideal que se puede representar como:

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n] e^{-j\omega n} \quad (2.7)$$

Siendo  $h_d[n]$  la correspondiente secuencia de respuesta al impulso, que se puede expresar en función de  $H_d(e^{j\omega})$  como:

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d[e^{j\omega}] e^{j\omega n} d\omega \quad (2.8)$$

Muchos sistemas se definen de forma idealizada mediante respuestas de frecuencia constantes por tramos con discontinuidades en los límites de las bandas. Como resultado, la respuesta al impulso de éstos sistemas es no causal

e infinitamente larga. La forma más directa de obtener una aproximación FIR causal a éstos sistemas es truncar la respuesta ideal.

La ecuación (2.7) puede verse como el desarrollo en series de *Fourier* de la respuesta de frecuencia periódica  $H_d(e^{j\omega})$  donde los valores de la secuencia  $h_d[n]$  juegan el papel de coeficientes del desarrollo en serie de *Fourier*. Por tanto, la aproximación de un filtro ideal mediante el truncamiento de la respuesta al impulso ideal es idéntica al tema de la convergencia del desarrollo en series de *Fourier*.

La forma más simple de obtener un filtro FIR causal a partir de  $h_d[n]$  es definir un nuevo sistema con respuesta al impulso  $h[n]$  dado por :

$$h[n] = \begin{cases} h_d[n] & 0 \leq n \leq M \\ 0 & \text{en el resto} \end{cases} \quad (2.9)$$

De forma mas general,  $h[n]$  se puede representar como el producto de la respuesta al impulso deseada  $h_d[n]$  y una ventana de longitud finita  $w[n]$ , es decir:

$$h[n] = h_d[n] w[n] \quad (2.10)$$

donde, para el truncamiento simple que indica la ecuación (2.9), la ventana aplicada se denomina *ventana rectangular*.

$$w[n] = \begin{cases} 1 & 0 \leq n \leq M \\ 0 & \text{en el resto} \end{cases} \quad (2.11)$$

Utilizando el teorema de modulación<sup>[2]</sup> o enventanado se puede ver que:

$$H[e^{j\omega}] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d[e^{j\omega}] W(e^{j(\omega-\theta)}) d\theta \quad (2.12)$$

Es decir,  $H(e^{j\omega})$  es la convolución periódica de la respuesta en frecuencia ideal deseada con la transformada de Fourier de la ventana. Por tanto, la respuesta de frecuencia  $H(e^{j\omega})$  será una versión "distorsionada" de la respuesta deseada  $H_d(e^{j\omega})$ . La figura 2.8(a) muestra funciones típicas  $H_d(e^{j\omega})$  y  $W(e^{j(\omega-\theta)})$  como se usarían en la ecuación (2.12).

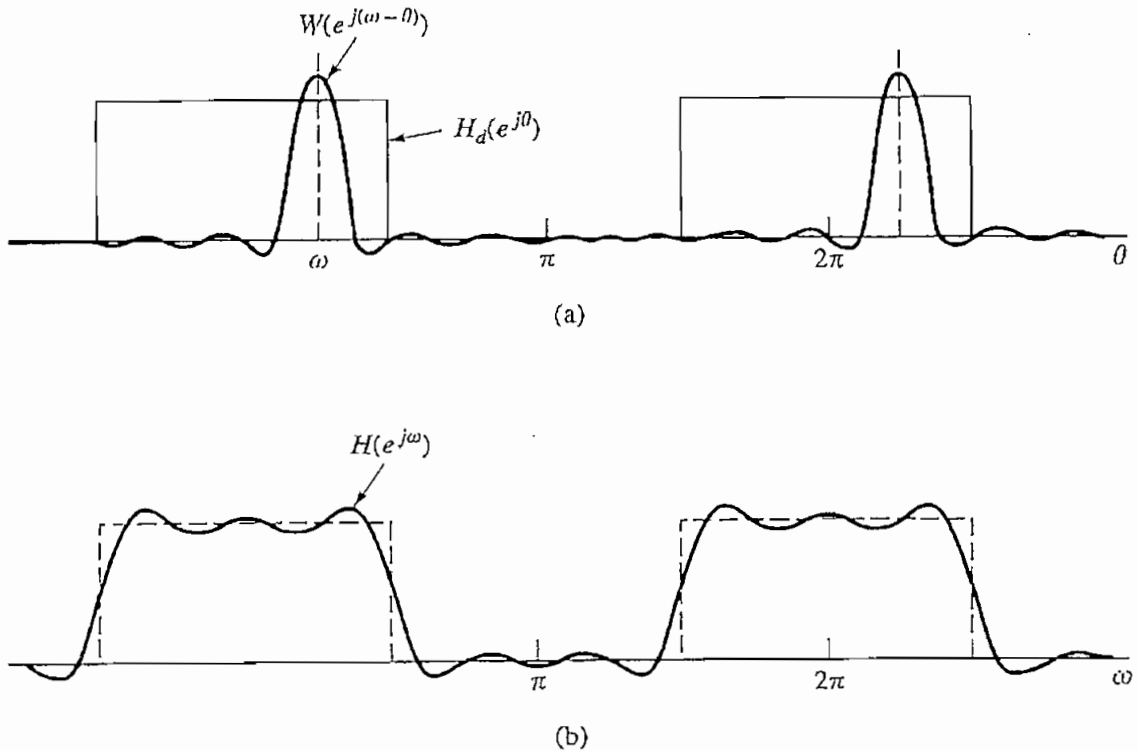


Figura 2.8 (a). Proceso de convolución implicado por el truncamiento de la respuesta impulso ideal (b). Aproximación típica resultante del enventanado de la respuesta al impulso ideal

Si  $\omega[n] = 1$  para todo  $n$  (sin trunca)  $W(e^{j\omega})$  es un tren periódico de impulsos de periodo  $2\pi$  y por tanto  $H(e^{j\omega}) = H_d(e^{j\omega})$ . Ésta interpretación sugiere que si se escoge  $\omega[n]$  de forma que  $W(e^{j\omega})$  se concentre en una banda estrecha de frecuencias alrededor de  $\omega = 0$ , entonces  $H(e^{j\omega})$  se parecerá mucho a  $H_d(e^{j\omega})$  excepto donde ésta última cambie muy abruptamente.

En consecuencia, la selección de la ventana está gobernada por el deseo de que la duración de  $\omega[n]$  sea tan corta como sea posible para minimizar los cálculos

necesarios en la realización del filtro, pero de forma que  $W(e^{j\omega})$  se aproxime a un pulso, es decir, que esté altamente concentrado en frecuencia de forma que la convolución de la ecuación (2.12) reproduzca fielmente la respuesta en frecuencia deseada.

Éstos dos requerimientos entran en conflicto como puede verse en el caso de la ventana rectangular de la ecuación 2.11 donde:

$$W(e^{j\omega}) = \sum_{n=0}^M e^{-j\omega n} = \frac{1 - e^{j\omega(M+1)}}{1 - e^{-j\omega}} = e^{-j\omega \frac{M}{2}} \frac{\text{sen}\left[\frac{\omega(M+1)}{2}\right]}{\text{sen}\left(\frac{\omega}{2}\right)} \quad (2.13)$$

La figura 2.9 muestra el módulo de la función  $\frac{\text{sen}[\omega(M+1)/2]}{\text{sen}(\omega/2)}$  para el caso de  $M = 7$ . Nótese que  $W(e^{j\omega})$  para la ventana rectangular tiene fase lineal generalizada debido a la simetría de la respuesta de frecuencia. A medida que  $M$  crece, el ancho del "lóbulo principal" decrece. El lóbulo principal se define generalmente como la región comprendida entre los primeros cruces por cero a ambos lados del origen.

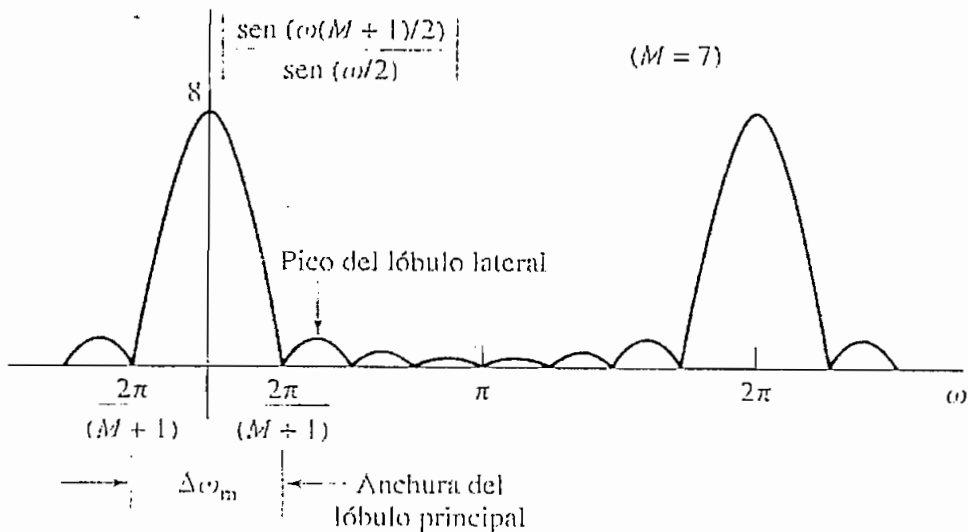


Figura 2.9 Módulo de la transformada de Fourier de una ventana rectangular ( $M = 7$ ) [5]

Para la ventana rectangular, el ancho del lóbulo principal es  $\Delta_{\omega} \approx 4\pi (M+1)$ . Sin embargo, para el caso de la ventana rectangular los lóbulos laterales son altos y de hecho, a medida que  $M$  crece las amplitudes de pico del lóbulo principal y de los lóbulos secundarios crecen de forma que el área encerrada en cada lóbulo permanece constante mientras que su anchura crece con  $M$ . Por tanto, cuando  $W(e^{j(\omega-\theta)})$  se "desliza" por la discontinuidad de  $H_d(e^{j\theta})$  con  $\omega$  creciente, la integral de  $W(e^{j(\omega-\theta)}) H_d(e^{j\theta})$  oscilará cuando cada lóbulo lateral de  $W(e^{j(\omega-\theta)})$  pase por la discontinuidad.

Como el área de cada lóbulo permanece constante al crecer  $M$ , las oscilaciones ocurren más rápidamente, pero no disminuyen su amplitud al aumentar  $M$ .

En la teoría de Series de Fourier,<sup>[3]</sup> es un hecho bien conocido que ésta convergencia no uniforme, denominada *fenómeno de Gibbs*<sup>1</sup>, se puede moderar mediante el uso de un truncamiento menos abrupto del desarrollo en series de Fourier. Si la ventana desciende suavemente hacia cero en cada extremo, la altura de los lóbulos laterales se puede reducir, pero a expensas de una mayor anchura del lóbulo principal y por tanto de una transición más ancha en la discontinuidad.

### 2.2.2.1 Propiedades de las ventanas comúnmente utilizadas

La figura 2.10 muestra algunas de las ventanas comúnmente utilizadas. Éstas ventanas se definen mediante las siguientes ecuaciones:

*Rectangular*

$$w[n] = \begin{cases} 1 & 0 \leq n \leq M \\ 0 & \text{en el resto} \end{cases} \quad (2.14 \text{ a})$$

---

<sup>1</sup> El comportamiento oscilatorio cerca del límite de la banda de paso del filtro se denomina fenómeno de *Gibbs*



Bartlett (triangular)

$$w[n] = \begin{cases} \frac{2n}{M} & 0 \leq n \leq M/2 \\ 2 - \frac{2n}{M} & M/2 \leq n \leq M \\ 0 & \text{en el resto} \end{cases} \quad (2.14 \text{ b})$$

Hanning

$$w[n] = \begin{cases} 0,5 - 0,5 \cos\left(\frac{2\pi n}{M}\right) & 0 \leq n \leq M \\ 0 & \text{en el resto} \end{cases} \quad (2.14 \text{ c})$$

Hamming

$$w[n] = \begin{cases} 0,54 - 0,46 \cos\left(\frac{2\pi n}{M}\right) & 0 \leq n \leq M \\ 0 & \text{en el resto} \end{cases} \quad (2.14 \text{ d})$$

Blackman

$$w[n] = \begin{cases} 0,42 - 0,5 \cos\left(\frac{2\pi n}{M}\right) + 0,08 \cos\left(\frac{4\pi n}{M}\right) & 0 \leq n \leq M \\ 0 & \text{en el resto} \end{cases} \quad (2.14 \text{ e})$$

Por conveniencia, la figura 2.10 muestra éstas ventanas como funciones de una variable continua. Sin embargo, como se indica en las ecuaciones (2.14) la secuencia de ventana se especifica solo para valores enteros de  $n$ .

Las ventanas definidas en las ecuaciones (2.14) se utilizan habitualmente tanto para análisis espectral como para el diseño de filtros FIR. Tienen la deseable propiedad de que sus transformadas de Fourier se concentran alrededor de  $\omega = 0$  además, su expansión funcional es sencilla de forma que se pueden calcular fácilmente.

La transformada de Fourier de la **Ventana de Bartlett** se puede expresar como el producto de las transformadas de Fourier de ventanas rectangulares, mientras que para las transformadas de Fourier de las demás ventanas se pueden expresar como sumas de transformadas de Fourier de la ventana rectangular dada en la ecuación (2.13) desplazadas en frecuencia.

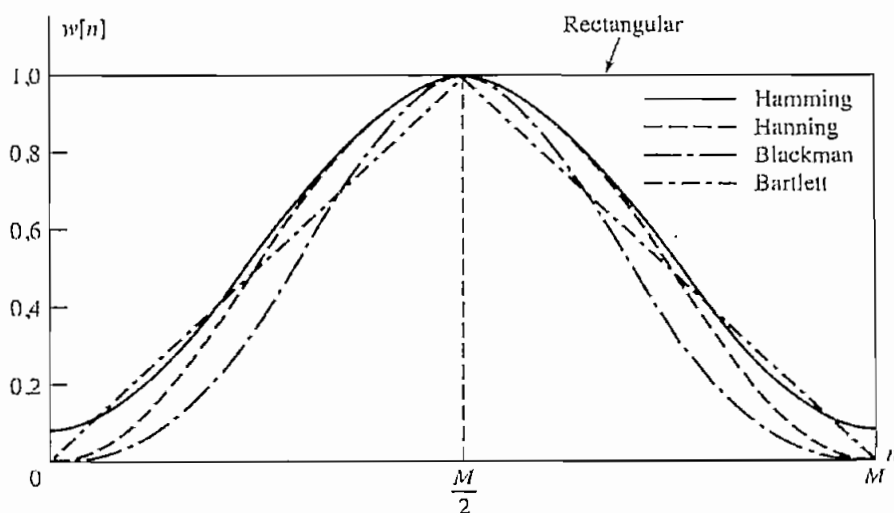
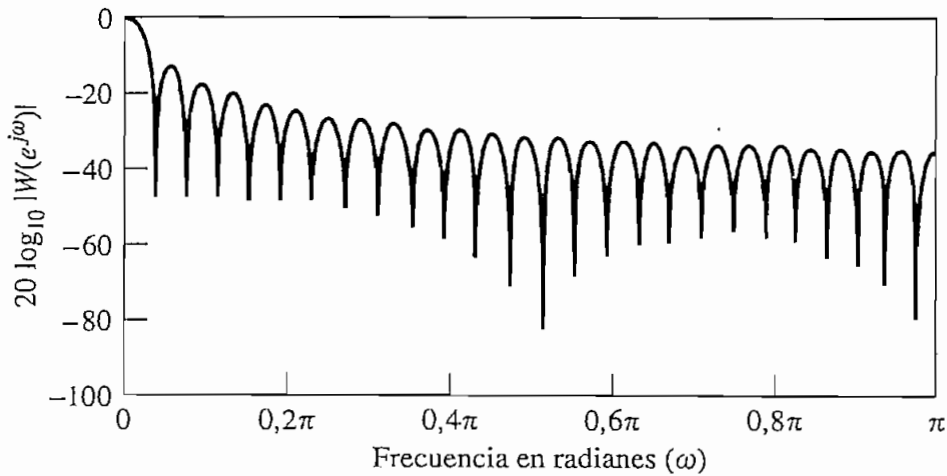


Figura 2.10 Ventanas comúnmente utilizadas. [5]

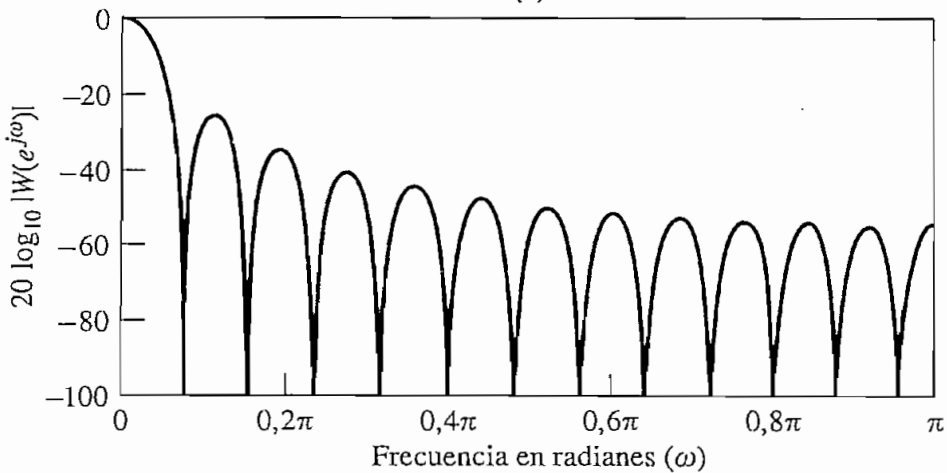
La figura 2.11 muestra la función  $20 \log_{10} |w(e^{j\omega})|$  para cada una de las ventanas interiores,  $M = 50$ . Puede verse claramente que la ventana rectangular es la que tiene el lóbulo principal más estrecho, y por tanto, para una longitud determinada,

es la que producirá en  $H(e^{j\omega})$  transiciones más abruptas en cada discontinuidad de  $H_d(e^{j\omega})$ . Sin embargo, el primer lóbulo está solo 13 dB por debajo del pico del lóbulo principal, lo que produce oscilaciones de  $H(e^{j\omega})$  de tamaño considerable en los alrededores de las discontinuidades de  $H_d(e^{j\omega})$ .

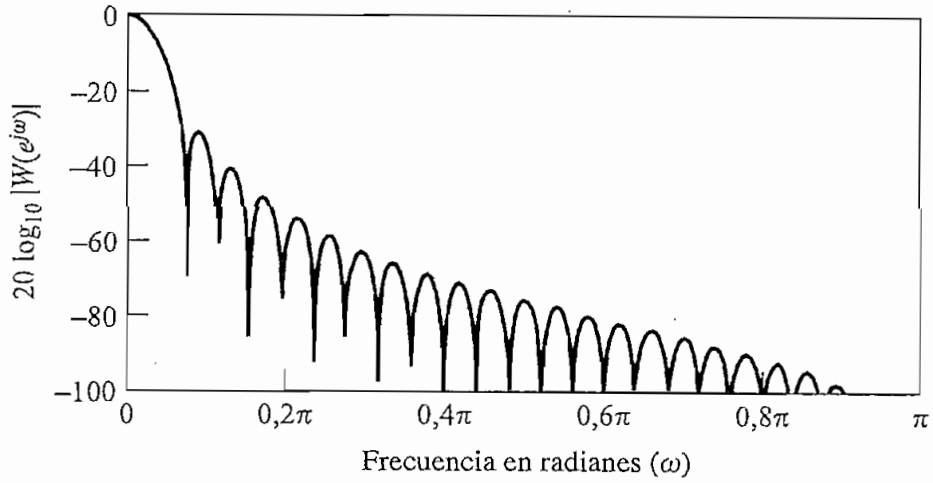
La tabla 2.1 que compara las ventanas de las ecuaciones (2.14) muestra cuando los extremos de las ventanas caen a cero suavemente, como ocurre con las ventanas de **Hamming**, **Hanning**, **Blackman**, los lóbulos laterales (segunda columna) reducen grandemente su amplitud. Sin embargo, se paga el precio de un lóbulo principal mucho más ancho (tercera columna) y por tanto transiciones más anchas en las discontinuidades de  $H_d(e^{j\omega})$ .



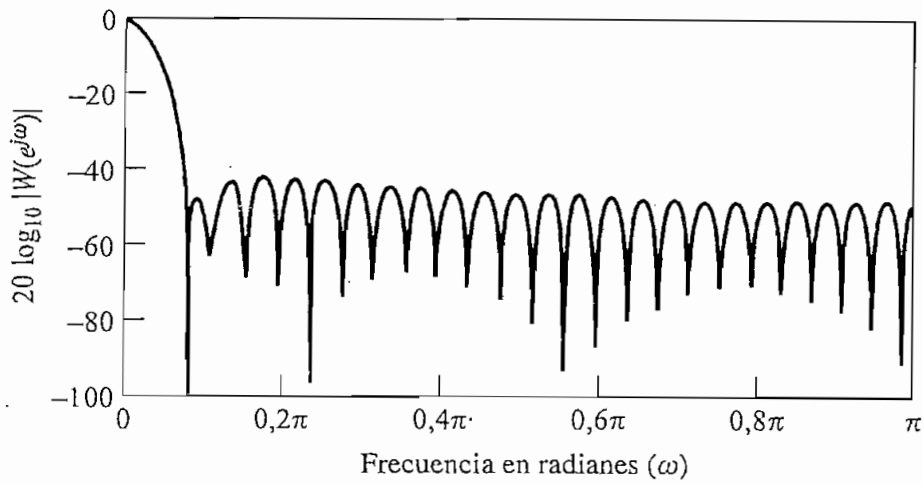
(a)



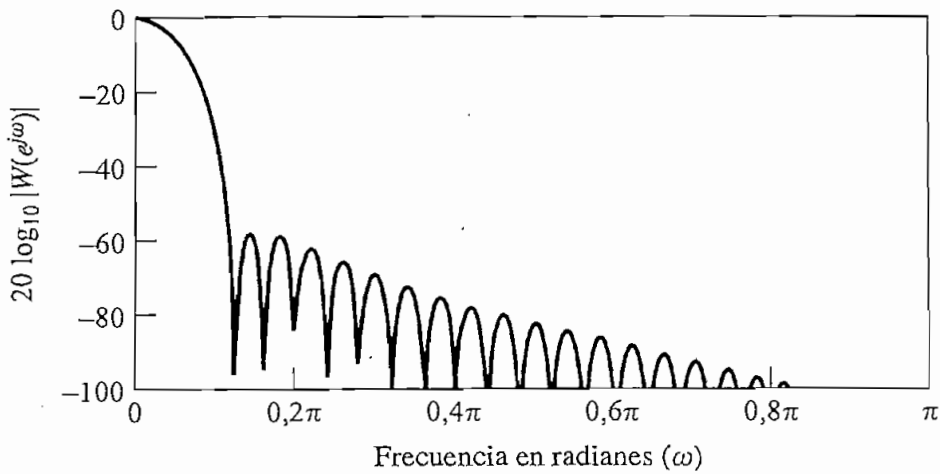
(b)



(c)



(d)



(e)

Figura 2.11. Transformadas de Fourier con  $M = 50$  (Logaritmos del módulo) de las ventanas [5]  
 a) Rectangular b) Bartlett c) Hanning d) Hamming e) Blackman

Tipo de ventana	Amplitud de pico del lóbulo lateral (relativo)	Anchura aproximada del lóbulo principal	Error de aproximación de pico $20 \log_{10} \delta$ (dB)	Ventana de Kaiser equivalente $\beta$	Anchura de transición de la ventana equivalente
Rectangular	-13	$4\pi/(M+1)$	-21	0	$1.81\pi/M$
Bartlett	-25	$8\pi/M$	-25	1.33	$2.37\pi/M$
Hanning	-31	$8\pi/M$	-44	3.86	$5.01\pi/M$
Hamming	-41	$8\pi/M$	-53	4.86	$6.27\pi/M$
Blackman	-57	$12\pi/M$	-74	7.04	$9.19\pi/M$

Tabla 2.1 Comparación de las ventanas comúnmente utilizadas [5].

### 2.2.2.2 Incorporación de la fase lineal generalizada

Todas las ventanas de las ecuaciones (2.14) incorporan una fase lineal, concretamente éstas ventanas tienen la propiedad de que:

$$w[n] = \begin{cases} w[M-n] & 0 \leq n \leq M \\ 0 & \text{en el resto} \end{cases} \quad (2.15)$$

Es decir, son simétricas con respecto al punto  $M/2$ ; como resultado, las transformadas de Fourier son de la forma:

$$W(e^{j\omega}) = W_e(e^{j\omega}) e^{-j\omega M/2} \quad (2.16)$$

siendo  $W_e(e^{j\omega})$  una función real y par de  $\omega$ , como se señaló en la ecuación (2.13). La convención de la ecuación (2.15) conduce en general a filtros causales y si la respuesta al impulso deseada es también simétrica con respecto a  $M/2$ , es decir, si  $h_d[M-n] = h_d[n]$ , la respuesta al impulso enventanada poseerá también esa simetría y la respuesta en frecuencia resultante tendrá también fase lineal generalizada. Es decir:

$$H(e^{j\omega}) = A_e(e^{j\omega}) e^{-j\omega M/2} \quad (2.17)$$

siendo  $A_e(e^{j\omega})$  una función real y par de  $\omega$ . De la misma forma, si la respuesta al impulso deseada es antisimétrica con respecto a  $M/2$  es decir, si  $h_d[M-n] = -h_d[n]$ , la respuesta al impulso enventanada será también antisimétrica con respecto a  $M/2$ , y la respuesta en frecuencia resultante tendrá una fase lineal generalizada con un desplazamiento de fase constante de 90 grados. Es decir:

$$H(e^{j\omega}) = jA_o(e^{j\omega}) e^{-j\omega M/2} \quad (2.18)$$

siendo  $A_o(e^{j\omega})$  una función real e impar de  $\omega$ .

Aunque los resultados anteriores son directos si se considera el producto de la ventana simétrica por la respuesta al impulso simétrica o antisimétrica deseada, es útil considerar la representación en el dominio de la frecuencia. Si se supone que  $h_d[M-n] = h_d[n]$ . Entonces:

$$H_d(e^{j\omega}) = H_e(e^{j\omega}) e^{-j\omega M/2} \quad (2.19)$$

donde  $H_e(e^{j\omega})$  es real y par.

Si la ventana es simétrica, se puede sustituir las ecuaciones (2.16) y (2.19) en la (2.12) con lo que se obtiene:

$$H(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_e(e^{j\theta}) e^{-j\theta M/2} W_e(e^{j(\omega-\theta)}) e^{-j(\omega-\theta)M/2} d\theta \quad (2.20)$$

unos cambios simples en los factores de fase producen

$$H(e^{j\omega}) = A_e(e^{j\omega}) e^{-j\omega M/2} \quad (2.21)$$

siendo:

$$A_e(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_e(e^{j\theta}) W_e(e^{j(\omega-\theta)}) d\theta \quad (2.22)$$

por tanto, el sistema resultante tiene fase lineal generalizada y además la función real  $A_e(e^{j\omega})$  es el resultado de realizar la convolución periódica de las funciones reales  $H_e(e^{j\omega})$  y  $W_e(e^{j\omega})$ .

El comportamiento detallado de la convolución de la ecuación (2.22) determina la respuesta en amplitud del filtro resultante del enventanado.

La cuarta columna de la tabla 2.1 muestra el error de aproximación de pico en dB para las ventanas de las ecuaciones (2.14). Puede verse claramente que las ventanas con menores lóbulos laterales producen mejores aproximaciones a la respuesta ideal en las discontinuidades. Además, la tercera columna que muestra la anchura del lóbulo principal sugiere que se pueden conseguir regiones de transición más estrechas aumentando  $M$ . Por tanto, mediante la selección de la forma y la duración de la ventana se puede controlar las propiedades del filtro FIR resultante. Sin embargo, ensayar con diversos tipos de ventanas y de longitudes mediante prueba y error no es una forma muy satisfactoria de diseñar filtros. Afortunadamente, Kaiser (1974) ha desarrollado una formalización simple del método de ventanas.

### 2.2.2.3 Fase lineal de un filtro pasa bajo

La respuesta de frecuencia deseada es:

$$H_{lp}(e^{j\omega}) = \begin{cases} e^{-j\omega N/2} & |\omega| < \omega_c \\ 0 & \omega_c < |\omega| \leq \pi \end{cases} \quad (2.23)$$

donde el factor de fase lineal generalizada se ha incorporado en la definición del filtro pasa bajo ideal. La correspondiente respuesta al impulso ideal es:

$$h_{hp}[n] = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{-j\omega M/2} e^{j\omega n} d\omega = \frac{\text{sen}[\omega_c(n - M/2)]}{\pi(n - M/2)} \quad (2.24)$$

para  $-\infty < n < \infty$ . Se puede demostrar fácilmente que  $h_{hp}[M - n] = h_{hp}[n]$ , de modo que si se utiliza una ventana simétrica en la ecuación:

$$h[n] = \frac{\text{sen}[\omega_c(n - M/2)]}{\pi(n - M/2)} w[n] \quad (2.25)$$

resultará un sistema con fase lineal.

**2.2.2.4 Fase lineal de un filtro pasa alto.**

El filtro pasa alto ideal con fase lineal generalizada tiene como respuesta de frecuencia:

$$H_{hp}(e^{j\omega}) = \begin{cases} 0 & |\omega| < \omega_c \\ e^{-j\omega M/2} & \omega_c < |\omega| \leq \pi \end{cases} \quad (2.26)$$

La correspondiente respuesta al impulso se puede obtener evaluando la transformada inversa de  $H_{hp}(e^{j\omega})$ , o bien se puede observar que:

$$H_{hp}(e^{j\omega}) = e^{-j\omega M/2} - H_{lp}(e^{j\omega}) \quad (2.27)$$

donde  $H_{lp}(e^{j\omega})$  es la fase lineal generalizada de un filtro pasa bajo, dada en la ecuación (2.24).

Por lo tanto  $h_{hp}[n]$  es:

$$h_{hp}[n] = \frac{\text{sen}[\pi(n - M/2)]}{\pi(n - M/2)} - \frac{\text{sen}[\omega_c(n - M/2)]}{\pi(n - M/2)} \quad -\infty < n < \infty \quad (2.28)$$



### 2.2.2.5 Fase lineal de un filtro pasa banda.

El filtro pasa banda ideal con fase lineal generalizada tiene como respuesta de frecuencia:

$$H_{bp}(e^{j\omega}) = \begin{cases} 0 & |\omega| < \omega_{c1} \cup \omega_{c2} < |\omega| < \pi \\ e^{-j\omega M/2} & \omega_{c1} < |\omega| \leq \omega_{c2} \end{cases} \quad (2.29)$$

La correspondiente respuesta al impulso se puede obtener evaluando la transformada inversa de  $H_{bp}(e^{j\omega})$ , o bien se puede observar que:

$$H_{bp}(e^{j\omega}) = e^{-j\omega M/2} - H_{lp}(e^{j\omega}) - H_{hp}(e^{j\omega}) \quad (2.30)$$

donde  $H_{lp}(e^{j\omega})$  y  $H_{hp}(e^{j\omega})$  es la fase lineal generalizada de un filtro pasa bajo y pasa alto dados en las ecuaciones 2.24 y 2.28 respectivamente.

Por lo tanto  $h_{bp}[n]$  es:

$$h_{bp}[n] = \frac{\text{sen}[\omega_{c2}(n - M/2)]}{\pi(n - M/2)} - \frac{\text{sen}[\omega_{c1}(n - M/2)]}{\pi(n - M/2)} \quad -\infty < n < \infty \quad (2.31)$$

### 2.2.2.6 Fase lineal de un filtro rechaza banda.

El filtro rechaza banda ideal con fase lineal generalizada tiene como respuesta de frecuencia:

$$H_{rb}(e^{j\omega}) = \begin{cases} e^{-j\omega M/2} & |\omega| < \omega_{c1} \cup \omega_{c2} < |\omega| < \pi \\ 0 & \omega_{c1} < |\omega| \leq \omega_{c2} \end{cases} \quad (2.32)$$

La correspondiente respuesta al impulso se puede obtener evaluando la transformada inversa de  $H_{rb}(e^{j\omega})$ , o bien se puede observar que:

$$H_{rb}(e^{j\omega}) = e^{-j\omega M/2} - H_{bp}(e^{j\omega}) \quad (2.33)$$

donde  $H_{bp}(e^{j\omega})$  es la fase lineal generalizada de un filtro pasa banda, dada en la ecuación (2.31).

Por lo tanto  $h_{rb}[n]$  es:

$$h_{rb}[n] = \frac{\text{sen}[\pi(n-M/2)]}{\pi(n-M/2)} - \frac{\text{sen}[\omega_{c2}(n-M/2)]}{\pi(n-M/2)} + \frac{\text{sen}[\omega_{c1}(n-M/2)]}{\pi(n-M/2)} \quad -\infty < n < \infty \quad (2.34)$$

### 2.3 MÉTODO DE KAISER

El compromiso entre anchura del lóbulo principal y área de los lóbulos laterales se puede cuantificar buscando la función de ventana que esté concentrada de forma máxima alrededor de  $\omega = 0$  en el dominio de la frecuencia. Este asunto se ha considerado en profundidad en una serie de artículos clásicos de *Slepian* (1961). La solución que se encontró en estos trabajos utiliza funciones de onda esferoidales que son difíciles de calcular y por tanto inadecuadas para el diseño de filtros. Sin embargo, Kaiser descubrió que se puede formar una ventana cuasi óptima utilizando la función de *Bessel* modificada de primer especie, que es mucho más sencilla de calcular. La ventana de Kaiser se define como:

$$w[n] = \begin{cases} \frac{I_0 \left[ \beta \sqrt{1 - \left[ \frac{(n-\alpha)}{\alpha} \right]^2} \right]}{I_0(\beta)} & 0 \leq n \leq M \\ 0 & \text{en el resto} \end{cases} \quad (2.35)$$

Siendo  $\alpha = M/2$  e  $I_0(\cdot)$  la función de *Bessel* modificada de primera clase y de orden 0.

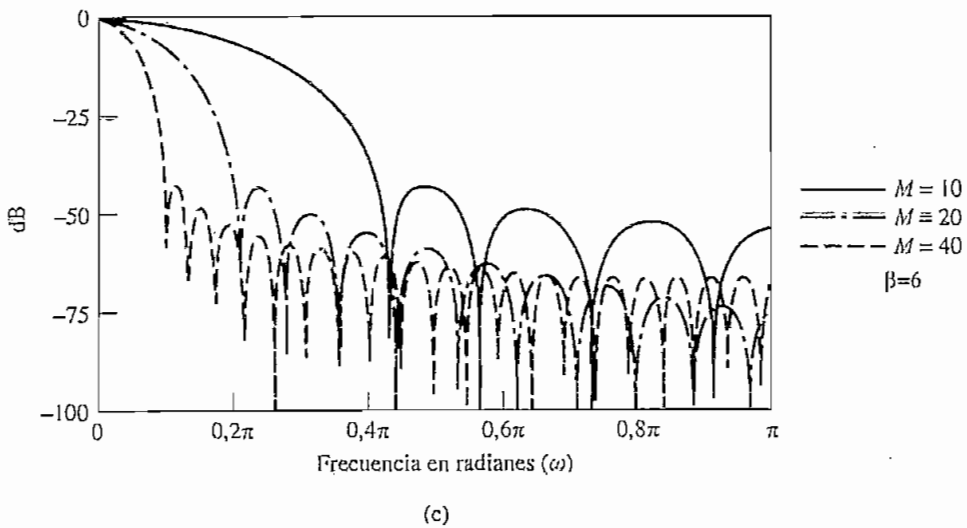
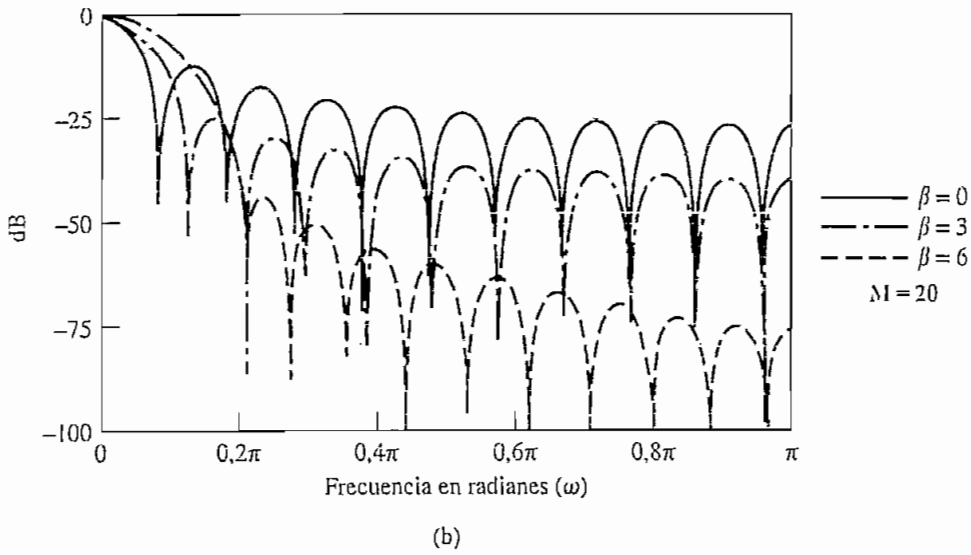
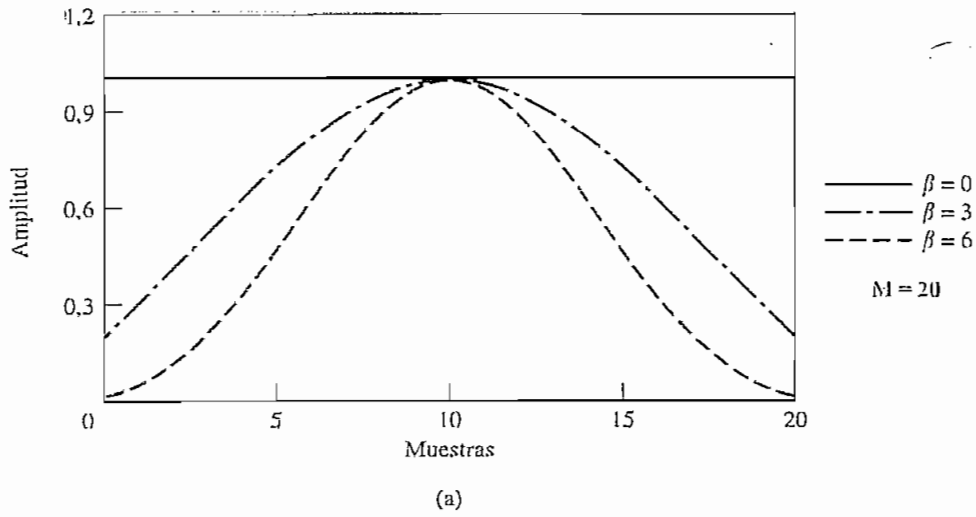


Figura 2.12 a) Ventanas de Kaiser para  $\beta = 0, 3, 6$  y  $M = 20$ . b) Transformadas de Fourier correspondientes a las ventanas de a). c) Transformadas de Fourier de las ventanas de Kaiser con  $\beta = 6$  y  $M = 10, 20, 40$ . [5]

A diferencia de las otras ventanas de las ecuaciones (2.14), la ventana de Kaiser tiene dos parámetros: la longitud  $(M + 1)$  y el parámetro  $\beta$ . Variando  $(M + 1)$  y  $\beta$  se puede ajustar la longitud y la forma de la ventana ajustándose a su vez el compromiso entre amplitud de los lóbulos laterales y anchura del lóbulo principal.

La figura 2.12 (a) muestra envolventes continuas de ventanas de Kaiser de longitud  $M+1 = 21$  para  $\beta = 0$ ,  $\beta = 3$  y  $\beta = 6$ . Al revisar la ecuación (2.35) se observa que el caso de  $\beta = 0$  se reduce a la ventana rectangular. La figura 2.12 (b) muestra las correspondientes transformadas de Fourier de las ventanas de la figura 2.12 (a). si se analiza la figura 2.12 (c) se puede concluir que variando los parámetros  $M+1$  y  $\beta$  se logra ajustar el filtro a requerimientos deseados.

Si la ventana desciende a cero mas suavemente, los lóbulos laterales de la transformada de Fourier disminuyen pero el lóbulo principal se hace más ancho. La figura 2.12 (c) muestra que si aumenta  $M$  manteniendo  $\beta$  constante, el lóbulo principal reduce su anchura pero no se modifica la amplitud de los lóbulos principales.

De hecho, Kaiser obtuvo, mediante amplias experimentaciones numéricas, una pareja de fórmulas que permiten al diseñador de filtros predecir los valores de  $M$  y  $\beta$  necesarios para cumplir una determinada especificación de selectividad de frecuencia. La figura 2.12 es también representativa del comportamiento típico que se obtiene utilizando la ventana de Kaiser.

Manteniendo fijo el factor de atenuación  $\delta$ , la frecuencia de corte de la banda de paso  $\omega_p$  del filtro pasa bajos se define como la máxima frecuencia para la que  $|H(e^{j\omega})| \geq 1 - \delta$ . La frecuencia de corte de la banda de supresión  $\omega_s$  se define como la mínima frecuencia para la que  $|H(e^{j\omega})| \leq \delta$ . Por tanto la anchura de la región de transición es:

$$\Delta\omega = \omega_s - \omega_p \quad (2.36)$$

Para la aproximación del filtro pasa bajo. Definiendo:

$$A = -20 \log_{10} \delta \text{ [dB]} \quad (2.37)$$

Kaiser determinó empíricamente que el valor de  $\beta$  necesario para cumplir un valor especificado de  $A$  está dado por:

$$\beta = \begin{cases} 0.1102(A - 8.7) & A > 50 \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21) & 21 \leq A \leq 50 \\ 0.0 & A < 21 \end{cases} \quad (2.38)$$

Sin olvidar que el caso de  $\beta = 0$  es la ventana rectangular para la que  $A = 21$ ; además, Kaiser descubrió que para cumplir unos valores especificados de  $A$  y  $\Delta\omega$ ,  $M$  debe satisfacer:

$$M = \frac{A - 8}{2.285 * \Delta\omega} \quad (2.39)$$

La ecuación (2.39) permite predecir el valor de  $M$  con una precisión de  $\pm 2$  para un amplio margen de valores de  $\Delta\omega$  y  $A$ . Por tanto, con estas fórmulas el método de diseño basado en la ventana de Kaiser casi no requiere iteraciones o prueba y error<sup>1</sup>.

### 2.3.1 RELACIÓN DEL FILTRO PASA BAJOS CON LOS FILTROS FUNDAMENTALES

Los filtros pasa alto, pasa banda y rechaza banda pueden ser diseñados a partir del filtro pasa bajo, es por esta razón que el análisis y la discusión acerca del diseño de filtros digitales se centraliza en los filtros pasa bajo.

<sup>1</sup> El diseño de filtros digitales IIR (*Chebyshev*, *Butterworth* y *Elípticas*) se realiza mediante métodos de aproximaciones basados en iteraciones, los cuales deben ser asistidos por un computador.

El método utilizado para convertir un filtro *pasa bajo* en un *pasa alto* se denomina *inversión espectral*.

La figura 2.13 ilustra el proceso de inversión espectral. Existen dos elementos que deben ser alterados en un filtro pasa bajo para convertirlo en pasa alto. Primero, se debe cambiar el signo de cada muestra del filtro pasa bajo y segundo, se debe sumar una muestra en el centro de simetría.

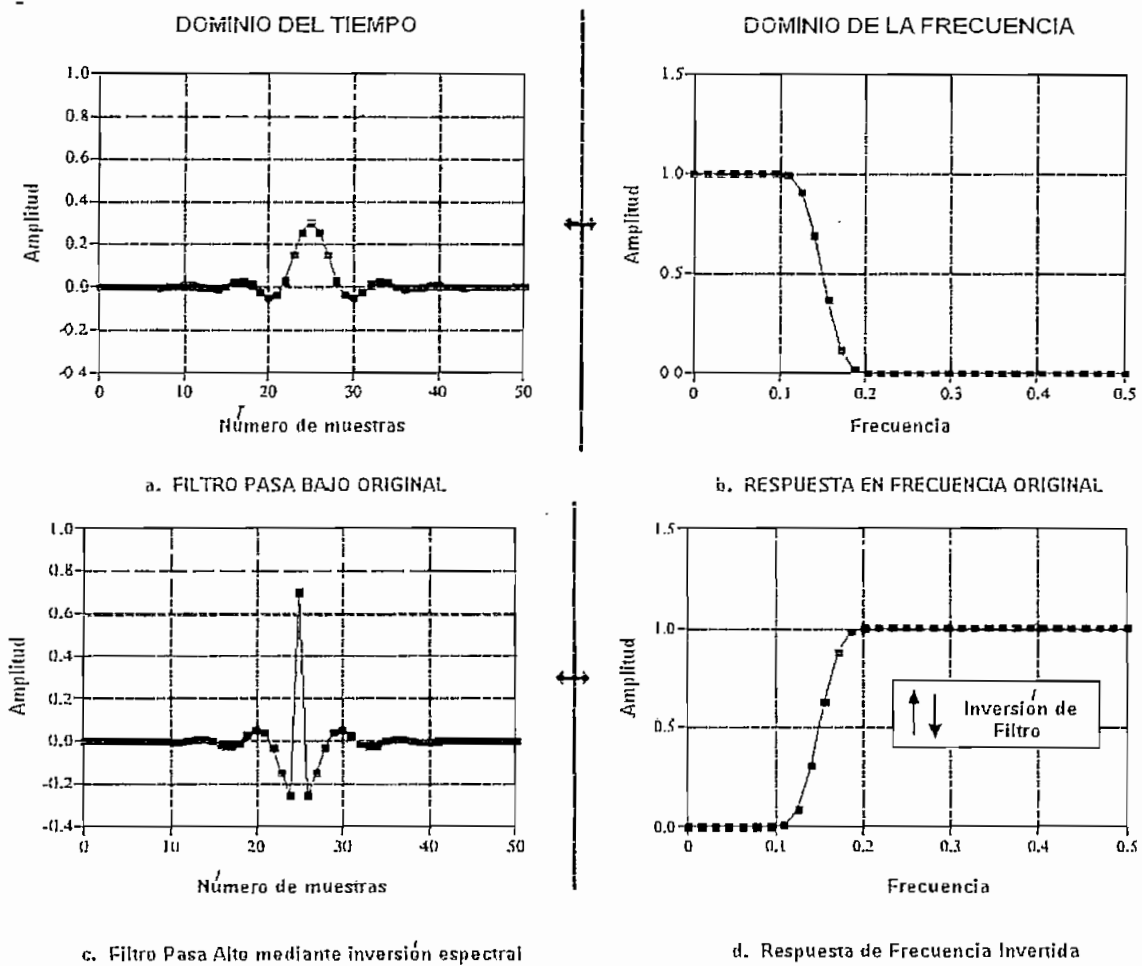


FIGURA 2.13. Transformación de un filtro pasa bajo a pasa alto [14]

La figura 2.14 ilustra en un diagrama de bloques como la modificación de estos dos elementos en el dominio del tiempo resultan en una inversión del espectro de frecuencia.

La señal de entrada en la figura 2.14 (a), es aplicada hacia  $x[n]$  mediante dos sistemas en paralelo. Uno de estos sistemas es un filtro pasa bajo con una respuesta impulsiva dada por  $h[n]$ , mientras que el otro sistema no altera en nada a la señal, teniendo por lo tanto como respuesta impulsiva una función delta  $\delta[n]$ .

La salida  $y[n]$  es igual a la salida del sistema pasa todo menos la salida del sistema pasa bajo. Ya que los componentes de baja frecuencia son restados de la señal original, solo los componentes de alta frecuencia aparecen en la salida.

En la figura 2.14 (b), el filtro pasa alto está dado por  $\delta[n] - h[n]$ , esto se obtiene cambiando el signo de todas las muestras del filtro pasa bajo y luego sumando una de las muestras en el centro de simetría.

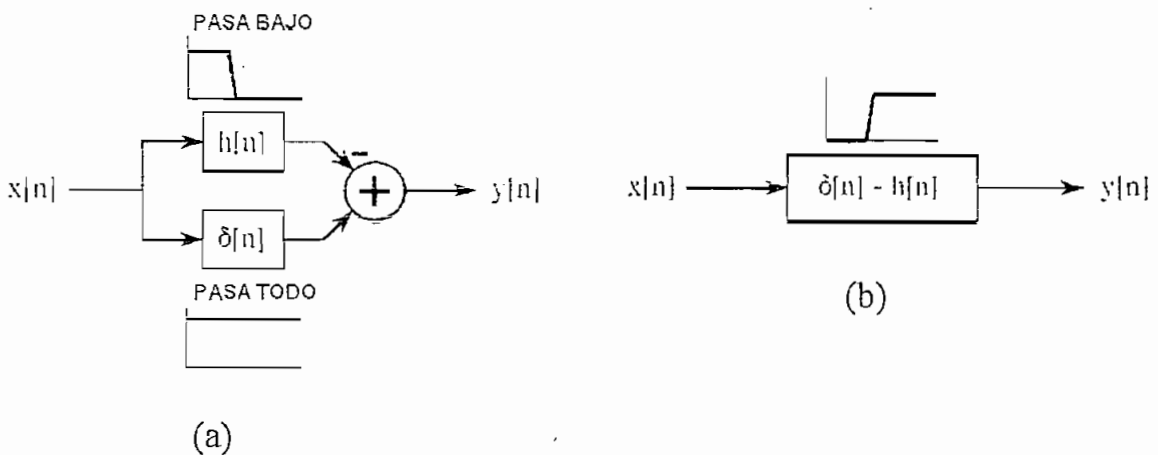


FIGURA 2.14 Diagrama de bloques de la inversión espectral [14]

Por otra parte, mediante el método de la ventana de Kaiser, los pasos pueden ser fácilmente modificados para el diseño de filtros pasa alto, pasa banda y rechaza banda, a partir de un filtro pasa bajo.

Para filtros **pasa alto**, las funciones de la frecuencia de paso  $f_p$  y la frecuencia de corte  $f_s$  son intercambiadas, por lo que, los únicos cambios en los pasos de diseño son la definición del parámetro  $\Delta\omega$  (variación de la banda de transición) como  $\Delta\omega = \omega_p - \omega_s$ , y el uso de la respuesta impulsiva  $h_{hp}[n]$  (obtenida mediante

inversión espectral) para este tipo de filtros. La respuesta impulsiva  $h_{hp}[n]$  pasa alto es:

$$h_{hp} = w[n] \left[ \delta[n-\alpha] - \frac{\sin(\omega_c(n-\alpha))}{\pi(n-\alpha)} \right] \quad (2.40)$$

El primer término de esta ecuación puede ser simplificado por  $w[n] \delta[n-\alpha] = w[\alpha] \delta[n-\alpha] = \delta[n-\alpha]$  debido a que  $w[\alpha] = 1$ . Por lo que, para el diseño de este tipo de filtros la respuesta impulsiva es:

$$h_{hp} = \delta[n-\alpha] - w[n] \frac{\sin(\omega_c(n-\alpha))}{\pi(n-\alpha)} \quad (2.41)$$

Para el mismo valor de  $\omega_c$ , los filtros pasa bajo y pasa alto son complementarios.

Para filtros **pasa banda**, las especificaciones que gobiernan este tipo de filtros son diferentes si se compara con las que caracterizan a los filtros pasa bajo. Como se observa en la figura 2.15, para filtros pasa banda se tiene dos bandas de supresión y de igual manera dos bandas de transición. Para el diseño final se deberá tener un único valor de banda de transición; para lo cual, se tomará el ancho de banda más pequeño de las dos bandas de transición.

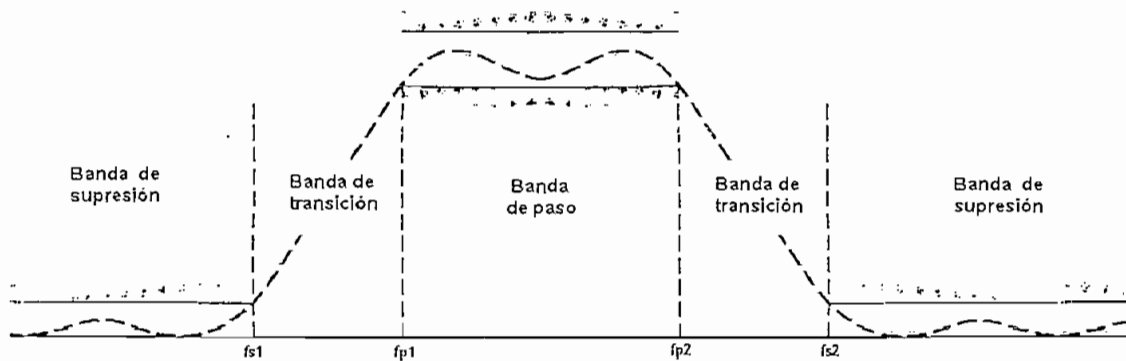


FIGURA 2.15 Parámetros de variación de un filtro pasa banda.

$$\Delta\omega = \min(\Delta\omega_1, \Delta\omega_2)$$



donde:

$$\Delta\omega_1 = \omega_{p1} - \omega_{s1}$$

$$\Delta\omega_2 = \omega_{s2} - \omega_{p2}$$

El filtro pasa banda al poseer dos bandas de transición, se observa la aparición de una frecuencia de corte adicional si se compara con un filtro pasa bajo, por lo que se tendrá que calcular tanto la  $\omega_{c1}$  como la  $\omega_{c2}$ , definidas por las ecuaciones siguientes:

$$\omega_{c1} = \frac{\omega_{p1} + \omega_{s1}}{2} \qquad \omega_{c2} = \frac{\omega_{p2} + \omega_{s2}}{2} \qquad (2.42)$$

Una vez que cada una de las frecuencias de corte ( $\omega_{c1}$  y  $\omega_{c2}$ ) y los parámetros de la ventana ( $M, \beta$ ) son calculados, la respuesta impulsiva pasa banda puede ser definida para  $n = 0, 1, \dots, M$ :

$$h_{bp} = w[n] \left[ \frac{\sin(\omega_{c2}(n-\alpha))}{\pi(n-\alpha)} - \frac{\sin(\omega_{c1}(n-\alpha))}{\pi(n-\alpha)} \right] \qquad (2.43)$$

Donde  $h[\alpha] = (\omega_{c2} - \omega_{c1}) / \pi$ .

Los filtros **rechaza banda** tienen dos bandas de paso y dos bandas de transición como se indica en la figura 2.16. De igual manera, que los filtros pasa bajo, se deberá tener un único valor de banda de transición; para lo cual, se tomará el ancho de banda más pequeño de las dos bandas de transición.



FIGURA 2.16 Parámetros de variación de un filtro rechaza banda.

$$\Delta\omega = \min(\Delta\omega_1, \Delta\omega_2)$$

donde:  $\Delta\omega_1 = \omega_{s1} - \omega_{p1}$   $\Delta\omega_2 = \omega_{p2} - \omega_{s2}$

Al igual que los filtros pasa banda, poseen dos frecuencias de corte  $\omega_{c1}$  y  $\omega_{c2}$ , definidas por las ecuaciones siguientes:

$$\omega_{c1} = \frac{\omega_{p1} + \omega_{s1}}{2} \quad \omega_{c2} = \frac{\omega_{p2} + \omega_{s2}}{2} \quad (2.44)$$

Una vez que cada una de las frecuencias de corte ( $\omega_{c1}$  y  $\omega_{c2}$ ) y los parámetros de la ventana ( $M, \beta$ ) son calculados, la respuesta impulsiva pasa banda puede ser definida para  $n = 0, 1, \dots, M$ :

$$h_{rb} = w[n] \left[ \frac{\sin(\omega_{c1}(n-\alpha))}{\pi(n-\alpha)} - \frac{\sin(\omega_{c2}(n-\alpha))}{\pi(n-\alpha)} \right] \quad (2.45)$$

Donde  $h[\alpha] = (\omega_{c1} - \omega_{c2}) / \pi$ .

## 2.3.2 EJEMPLOS DE DISEÑOS DE FILTROS DIGITALES MEDIANTE LA VENTANA DE KAISER

### 2.3.2.1. Diseño de un filtro pasa bajo

Con el uso de la ventana de Kaiser es directo diseñar un filtro FIR pasa bajo que cumpla unas determinadas especificaciones. El procedimiento es como sigue:

1. Primero deben establecerse las especificaciones. Esto significa seleccionar los valores deseados de  $f_p$  y  $f_s$  expresados en Hertz; y el máximo error de aproximación tolerable. Utilizando el método de ventanas, el filtro resultante tendrá el mismo error de pico en la banda de paso y en la banda de supresión  $\delta$ .

Para el ejemplo:

- $f_p = 4000$  Hz                      frecuencia de paso
- $f_s = 6000$  Hz                      frecuencia de supresión
- $f_m = 20000$  Hz                    frecuencia de muestreo
- $\delta_1 = 0.01$

$\delta_2 = 0.001$ , como el método de diseño de la ventana implica que  $\delta_1 = \delta_2$ , se debe hacer que  $\delta = 0.001$

2. Encontrar la frecuencia de corte del filtro pasa bajo ideal. Debido a la simetría de la aproximación en la discontinuidad de  $H_d(e^{j\omega})$ , sería

$$\omega_p = \frac{f_p}{f_m} * 2\pi = \frac{4000}{20000} * 2\pi = 0.4\pi$$

$$\omega_s = \frac{f_s}{f_m} * 2\pi = \frac{6000}{20000} * 2\pi = 0.6\pi$$

$$\omega_c = \frac{\omega_p + \omega_s}{2} = \frac{0.4\pi + 0.6\pi}{2} = 0.5\pi$$

3. Determinar los parámetros de la ventana de Kaiser, para lo cual se tiene que calcular

$$\Delta\omega = \omega_s - \omega_p = 0.6\pi - 0.4\pi = 0.2\pi$$

$$A = -20 \log_{10} \delta = 60 \text{ [dB]}$$

Al sustituir estas dos cantidades en las ecuaciones (2.38) y (2.39) se obtiene los valores de  $\beta$  y  $M$

$$\beta = 5.653$$

$$M = 37$$

4. La respuesta al impulso del filtro se calcula mediante las ecuaciones (2.31) y (2.35), y se obtiene:

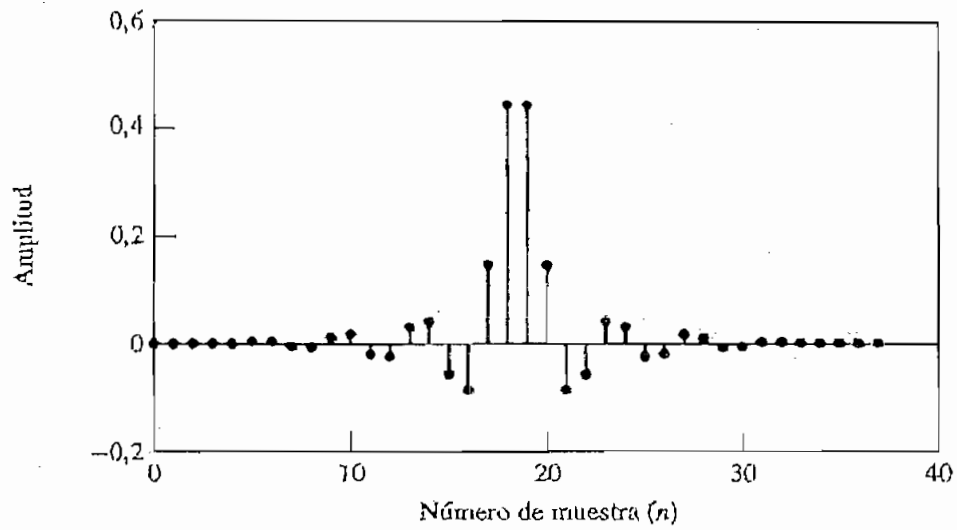
$$h[n] = \begin{cases} \frac{\text{sen} \omega_c (n - \alpha) * I_0 \left[ \beta \left( 1 - \left[ \frac{(n - \alpha) - \alpha}{M} \right]^2 \right)^{1/2} \right]}{\pi (n - \alpha) I_0(\beta)} & 0 \leq n \leq M \\ 0 & \text{en el resto} \end{cases}$$

siendo  $\alpha = M / 2 = 37/2$ . Como  $M = 37$  es un entero impar, el sistema de fase lineal resultante será de tipo II<sup>[2]</sup>. Las características de la respuesta del filtro se muestran en la figura 2.17. La figura 2.17 (a), que muestra la respuesta al impulso, ilustra la simetría característica de un sistema de tipo II. La figura 2.17 (b), que muestra el algoritmo de la respuesta de amplitud en dB, indica que  $H(e^{j\omega})$  es cero en  $\omega = \pi$ , o lo que es lo mismo, que  $H(z)$  es cero en  $z = -1$ , condición que debe cumplir un sistema FIR de tipo II. La figura 2.17 (c) muestra el error de aproximación en la banda de paso y en las bandas de supresión. La función de este error se define como:

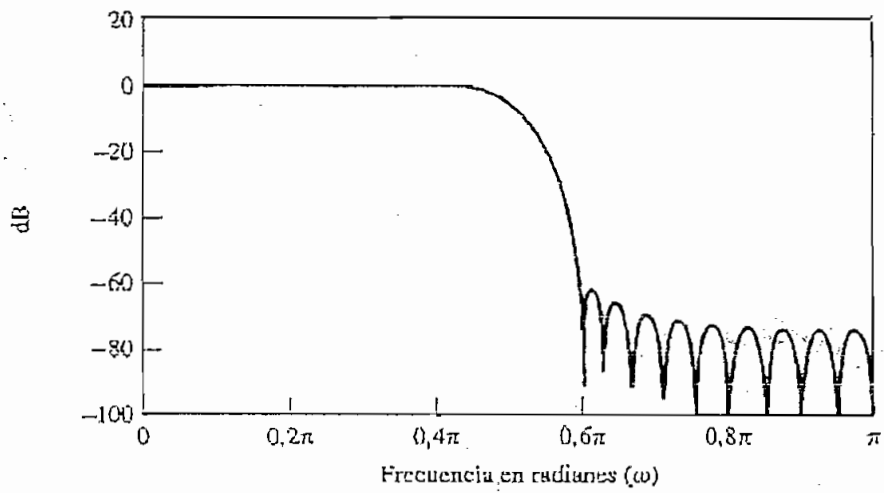
$$E_d(\omega) = \begin{cases} 1 - A_e(e^{j\omega}) & 0 \leq \omega \leq \omega_p \\ 0 - A_e(e^{j\omega}) & \omega_s \leq \omega \leq \pi \end{cases} \quad (2.46)$$

El error no se define en la región de transición,  $0.4\pi < \omega < 0.6\pi$ . Nótese la simetría del error de aproximación, y también que el error de aproximación de pico es ligeramente mayor que  $\delta = 0.001$ . Si se aumenta  $M$  hasta 38 se obtiene un filtro de tipo I con  $\delta = 0.0008$ .

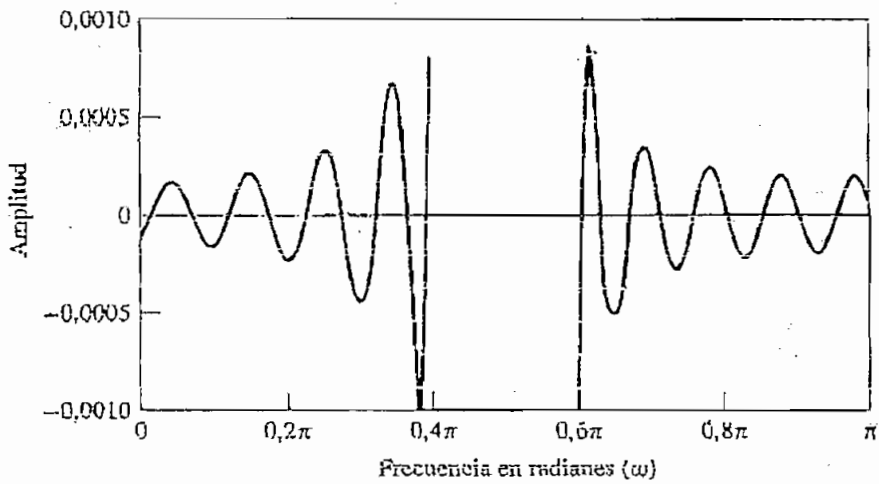
Finalmente, puede observarse que no es necesario dibujar la fase o el retardo de grupo, ya que se sabe que la fase es lineal y el retardo es de  $M = 18,5$  muestras.



(a)



(b)



(c)

Figura 2.17 Funciones de respuesta del ejemplo de diseño de un filtro pasa bajo mediante la ventana de Kaiser. a) Respuesta al impulso ( $M=37$ ). b) Logaritmo del módulo. c) Error de aproximación. [5]

### 2.3.2.2. Diseño de un filtro pasa alto

Diseñar un filtro pasa alto que cumpla con los siguientes parámetros:

$$f_s = 900 \text{ Hz}$$

$$f_p = 1000 \text{ Hz}$$

$$f_m = 3200 \text{ Hz}$$

$$\delta = 0,0001$$

Normalizando la frecuencia de paso y de supresión:

$$\omega_s = \frac{f_s}{f_m} * 2\pi = \frac{900}{3200} * 2\pi = 0,5625\pi$$

$$\omega_p = \frac{f_p}{f_m} * 2\pi = \frac{1000}{3200} * 2\pi = 0,6250\pi$$

La frecuencia de corte del filtro pasa alto ideal, sería :

$$\omega_o = \frac{\omega_p + \omega_s}{2} = \frac{0,6250\pi + 0,5625\pi}{2} = 0,59375\pi$$

Los parámetros de la ventana de Kaiser se calculan:

$$\Delta\omega = \omega_p - \omega_s = 0,6250\pi - 0,5625\pi = 0,0625\pi$$

La atenuación en la banda de supresión correspondiente es:

$$A = -20 \log_{10} \delta = 80 \text{ [dB]}$$

Al sustituir estas dos cantidades en las ecuaciones (2.38) y (2.39) se obtiene los valores de  $\beta$  y  $M$

$$\beta = 7.85626$$

$$M = 160$$

La respuesta al impulso del filtro se calcula mediante la ecuación (2.41); esto es:

$$h_{hp} = \delta[n - \alpha] - w[n] \frac{\sin(\omega_c(n - \alpha))}{\pi(n - \alpha)}$$

### 2.3.2.3. Diseño de un filtro pasa banda

Diseñar un filtro pasa banda que cumpla con los siguientes parámetros:

$$f_{s1} = 500 \text{ Hz}$$

$$f_{p1} = 700 \text{ Hz}$$

$$f_{p2} = 1000 \text{ Hz}$$

$$f_{s2} = 1200 \text{ Hz}$$

$$f_m = 3600 \text{ Hz}$$

$$\delta = 0,01$$

Normalizando las frecuencias de paso y de supresión :

$$\omega_{s1} = \frac{f_{s1}}{f_m} * 2\pi = \frac{500}{3200} * 2\pi = 0.3125\pi$$

$$\omega_{p1} = \frac{f_{p1}}{f_m} * 2\pi = \frac{700}{3200} * 2\pi = 0.4375\pi$$

$$\omega_{p2} = \frac{f_{p2}}{f_m} * 2\pi = \frac{1000}{3200} * 2\pi = 0.6250\pi$$

$$\omega_{s2} = \frac{f_{s2}}{f_m} * 2\pi = \frac{1200}{3200} * 2\pi = 0.75\pi$$

Las frecuencias de corte del filtro pasa banda ideal, serían:

$$\omega_{c1} = \frac{\omega_{p1} + \omega_{s1}}{2} = \frac{0.3125\pi + 0.4375\pi}{2} = 0.375\pi$$

$$\omega_{c2} = \frac{\omega_{p2} + \omega_{s2}}{2} = \frac{0.6250\pi + 0.75\pi}{2} = 0.6875\pi$$

Los parámetros de la ventana de Kaiser se calculan:

$$\Delta\omega_1 = \omega_{p1} - \omega_{s1} = 0.4375\pi - 0.3125\pi = 0.125\pi$$

$$\Delta\omega_2 = \omega_{s2} - \omega_{p2} = 0.75\pi - 0.6250\pi = 0.125\pi$$

La atenuación en la banda de supresión es:

$$A = -20 \log_{10} \delta = 40 \text{ [dB]}$$

Al sustituir estas dos cantidades en las ecuaciones (2.38) y (2.39) se obtiene los valores de  $\beta$  y  $M$

$$\beta = 3.3953$$

$$M = 36$$

La respuesta al impulso del filtro se calcula mediante la ecuación (2.43); esto es:

$$h_{bp} = w[n] \left[ \frac{\sin(\omega_{c2}(n-\alpha))}{\pi(n-\alpha)} - \frac{\sin(\omega_{c1}(n-\alpha))}{\pi(n-\alpha)} \right]$$

#### 2.3.2.4. Diseño de un filtro rechaza banda

Diseñar un filtro rechaza banda que cumpla con los siguientes parámetros:

$$f_{p1} = 3000 \text{ Hz}$$



$$f_{s1} = 4000 \text{ Hz}$$

$$f_{s2} = 11000 \text{ Hz}$$

$$f_{p2} = 12000 \text{ Hz}$$

$$f_m = 32000 \text{ Hz}$$

$$\delta = 0,003162$$

Normalizando las frecuencias de paso y de supresión:

$$\omega_{p1} = \frac{f_p}{f_m} * 2\pi = \frac{3000}{32000} * 2\pi = 0.1875\pi$$

$$\omega_{s1} = \frac{f_s}{f_m} * 2\pi = \frac{4000}{32000} * 2\pi = 0.25\pi$$

$$\omega_{s2} = \frac{f_s}{f_m} * 2\pi = \frac{11000}{32000} * 2\pi = 0.6875\pi$$

$$\omega_{p2} = \frac{f_p}{f_m} * 2\pi = \frac{12000}{32000} * 2\pi = 0.75\pi$$

Las frecuencias de corte del filtro rechaza banda ideal, serían:

$$\omega_{c1} = \frac{\omega_{p1} + \omega_{s1}}{2} = \frac{0.1875\pi + 0.25\pi}{2} = 0.21875\pi$$

$$\omega_{c2} = \frac{\omega_{p2} + \omega_{s2}}{2} = \frac{0.75\pi + 0.6875\pi}{2} = 0.71875\pi$$

Los parámetros de la ventana de Kaiser se calculan:

$$\Delta\omega_1 = \omega_{s1} - \omega_{p1} = 0.25\pi - 0.1875\pi = 0.0625\pi$$

$$\Delta\omega_2 = \omega_{p2} - \omega_{s2} = 0.75\pi - 0.6875\pi = 0.0625\pi$$

La atenuación en la banda de supresión es:

$$A = -20 \log_{10} \delta = 50 \text{ dB}$$

Al sustituir estas dos cantidades en las ecuaciones (2.38) y (2.39) se obtiene los valores de  $\beta$  y  $M$

$$\beta = 4.53351$$

$$M = 94$$

La respuesta al impulso del filtro se calcula mediante la ecuación (2.45); esto es:

$$h_{rb} = w[n] \left[ \frac{\sin(\omega_{c1}(n-\alpha))}{\pi(n-\alpha)} - \frac{\sin(\omega_{c2}(n-\alpha))}{\pi(n-\alpha)} \right]$$

### 2.3.3 RELACIÓN DE LA VENTANA DE KAISER CON OTRAS VENTANAS

El principio básico del método de diseño de la ventana es truncar la respuesta al impulso ideal con una ventana de longitud finita. El correspondiente efecto en el dominio de la frecuencia es que la respuesta en frecuencia ideal se convoluciona con la transformada de Fourier de la ventana.

Si el filtro ideal es un filtro pasa bajo, la discontinuidad de la respuesta en frecuencia se suaviza a medida que el lóbulo principal de la transformada de Fourier de la ventana atraviesa la discontinuidad en el proceso de convolución.

En una primera aproximación se puede decir que la anchura de la banda de transición resultante está determinada por la anchura del lóbulo principal de la transformada de Fourier de la ventana, y el rizado de la banda de paso y de la banda de supresión están determinados por sus lóbulos laterales. Como el rizado de la banda de paso y la banda de supresión están producidos por la integración de los lóbulos laterales de la ventana simétrica dichos rizados son aproximadamente iguales.

Además, es una buena aproximación suponer que las máximas desviaciones en la banda de paso y en la banda de supresión no dependen de  $M$  y sólo se pueden cambiar modificando la forma de la ventana utilizada. Esto se ilustra en la fórmula de Kaiser del parámetro  $\beta$ , ecuación (2.38), que es independiente de  $M$ .

En la tabla 2.1 se observa algunos aspectos que permiten relacionar la ventana de Kaiser con algunas ventanas comúnmente utilizadas. La quinta columna de la tabla muestra ciertos valores para el parámetro  $\beta$  de la ventana de Kaiser, los mismos que permiten tener un pico de error de aproximación similares a otras ventanas.

## CAPÍTULO 3

### IMPLEMENTACIÓN DE FILTROS FIR

---

La implementación de los filtros FIR consiste principalmente en un algoritmo basado en sumas y multiplicaciones aplicadas al lenguaje ensamblador del Procesador Digital de Señales ADSP-2181<sup>[1,3]</sup> el cual permite optimizar el tiempo de procesamiento en relación a programas de generación de código como C o *Ptolemy*<sup>1)</sup>. El algoritmo implementado es usado como plantilla de los diferentes filtros digitales no recursivos (de respuesta finita al impulso o FIR), en alguna de sus formas básicas: pasa bajo, pasa alto, pasa banda o rechaza banda.

Los filtros digitales pueden ser diseñados fácilmente usando una gran variedad de *software* como el *QEDesign 2000*<sup>2)</sup> o el *FDAS (Filter Design and Analysis System)*<sup>3)</sup>, ambos programas costosos. Pero en el presente proyecto, los filtros se diseñan y simulan basándose en la versión para estudiantes, MATLAB 5.3, y en el Laboratorio Matemático MATHCAD 4.0. Una vez establecidos los límites y características de los filtros pueden llevarse al lenguaje ensamblador del DSP.

#### 3.1 CONSIDERACIONES INICIALES

En el estudio de los filtros digitales, se ha supuesto que tanto las variables como los coeficientes de los filtros pueden tomar cualquier valor; ahora bien, la realización de estos sistemas se efectúa con dispositivos digitales de cálculo, es decir con microprocesadores DSP o computadores, en los que el almacenamiento de los valores de las señales se realiza con números finitos de dígitos.

---

<sup>1)</sup> C o *Ptolemy* Son lenguajes ensambladores de programación que estudian los modelos, simulación y diseño de sistemas en tiempo real.

<sup>2)</sup> *QEDesign 2000* Es un paquete de *software* que da soporte al diseño de filtros FIR e IIR, trabaja con hasta 8,192 coeficientes y ordenes de hasta 160. Su costo aproximado redondea los 4200 USD. [Fuente: [www.mds.com](http://www.mds.com)].

<sup>3)</sup> *FDAS* Sistema encargado del diseño, análisis y simulación de los Filtros digitales.

La operación de transformación de los valores de variables y constantes con un número infinito de cifras a un número concreto de bits se denomina *cuantificación*. Este proceso es imprescindible en el tratamiento de la información de los DSP, produciendo errores que deben ser analizados y acotados.

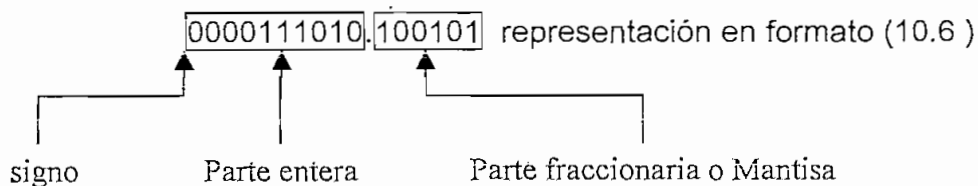
En la implementación de los filtros digitales aparecen tres procesos de cuantificación:

1. Cuantificación de la señal de entrada
2. Cuantificación de los coeficientes de los filtros
3. Errores en el redondeo y truncamiento de las operaciones

### Efectos de truncamiento y redondeo

En el análisis de error de cuantificación hay que tener en cuenta la forma de representar los valores en el microprocesador, según se haga en punto fijo o en punto flotante<sup>1</sup> (el microprocesador ADSP-2181 trabaja con representaciones en punto fijo de 16 bits) y asimismo, también tendrá influencia la forma de realizar la conversión de un número cualquiera equivalente a infinitos dígitos, a un número finito de dígitos, según sea por *truncamiento* o por *redondeo*.

En general la representación de los números en los procesadores se realiza en base 2, es decir en representación binaria, en donde al trabajar en punto fijo se utiliza un número determinado de dígitos para la parte entera y otro para la parte decimal, por ejemplo un registro de 16 bits de ADSP-2181 podría contener:



<sup>1</sup> Anexo 1. Representaciones y formatos numéricos.

La forma de representar los números negativos también tiene su importancia según se haga por signo y magnitud, por complemento de dos o por complemento de uno. Tanto en punto fijo como flotante se considera que la parte fraccionaria o mantisa tiene  $n$  bits, entonces los números se representan en saltos de  $2^{-n}$ , es decir la mínima diferencia entre dos representaciones distintas es de  $2^{-n}$ .

Dado un número infinito de dígitos, se entiende por **truncamiento** del mismo la operación de tomar los “ $b$ ” primeros bits más significativos, despreciando el resto, mientras que su **redondeo** es tomar la representación del valor más próximo al mismo. En definitiva, el efecto de redondeo y truncamiento es introducir un error cuyo valor depende del número de bits originales respecto al número de bits después de la cuantificación; para efectos prácticos se puede asumir que un número original  $x$  está representado por una secuencia infinita de bits, mientras que el truncamiento con  $b$  bits está representado por  $Q(x)$  de la siguiente manera:

$$\begin{array}{ll} x = a_0 \cdot a_{-1} a_{-2} a_{-3} a_{-4} a_{-5} a_{-6} a_{-7} a_{-8} \dots\dots\dots & \text{número original} \\ Q(x) = a_0 \cdot a_{-1} a_{-2} a_{-3} a_{-4} a_{-5} & \text{truncamiento de } x \end{array}$$

El error de truncamiento viene dado por :  $E_t = Q(x) - x$

Y su rango de variación es:  $-2^{-b} < E_t < 0$

En valor absoluto el error máximo por redondeo está dado por:

$$-2^{-b-1} < E_t < 2^{-b+1}$$

para valores positivos y negativos en formato de complemento de dos.

### Cuantificación de señales

En la conversión analógica / digital de una señal para su tratamiento con el microprocesador además del proceso de muestreo hay que tener en cuenta el de cuantificación, efecto que es debido a la limitación del número de cifras de los registros en los que se almacenan sus valores.

## Cuantificación de los coeficientes

Igual que ocurre con los valores de las muestras de la señal, en el proceso de implementación de un filtro, los parámetros que lo definen sufren una cuantificación que puede afectar, en algunas ocasiones considerablemente a las prestaciones del sistema.

Las consecuencias prácticas de la cuantificación de los coeficientes son:

- Los filtros con poco ancho de banda son muy sensibles a la cuantificación.
- El sobremuestreo (*oversampling*)<sup>1</sup> incrementa la sensibilidad a la cuantificación.
- La estructura de implementación del filtro influye mucho.

## Cuantificación de resultados de operaciones

A efectos de cuantificación, la operación de mayor interés es la multiplicación ya que el resultado de ésta ocupa doble número de cifras que los factores, por lo que para almacenarlo o realizar otras operaciones habrá que reducir el número de sus cifras.

### 3.1.1 ETAPAS EN EL DISEÑO E IMPLEMENTACIÓN DE LOS FILTROS DIGITALES

El diseño de filtros requiere realizar las siguientes etapas:

1. Especificaciones de las propiedades deseadas del sistema.

---

<sup>1</sup> **Sobremuestreo** (*oversampling*) Técnica que se emplea en el procesamiento digital de la señal, con el fin de eliminar los espurios que se generan cuando se trabaja cerca del extremo del espectro.

2. Aproximación de las propiedades mediante un sistema en tiempo discreto.
3. Realización del Sistema.

La primera etapa es altamente dependiente de la aplicación deseada y la tercera de la tecnología usada para la realización. Aunque estos tres pasos no son independientes, el más importante es el segundo ya que éste describe el método que se empleará para el diseño del filtro. En términos prácticos, el filtro deseado se realiza utilizando una programación dependiente de la tecnología mediante cómputo digital.

### 3.1.1.1 Especificación de las propiedades deseadas del sistema

Cuando se utiliza un filtro digital discreto para realizar el tratamiento de señales en tiempo continuo, se emplea la configuración usada en la figura 3.1, (configuración que fue analizada en la Sección 2.2), en donde las especificaciones del filtro en tiempo discreto se suelen dar en el dominio de la frecuencia. Esto es particularmente cierto para el caso de los filtros selectivos en frecuencia como los filtros pasa bajo, pasa banda y pasa alto.

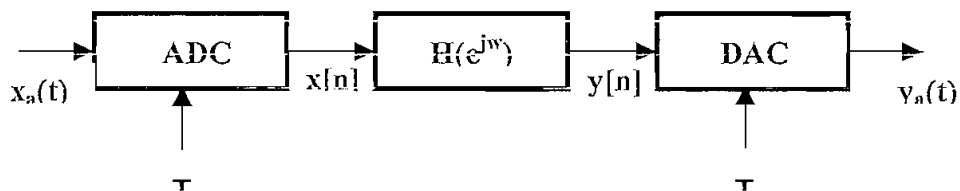
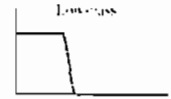


Figura 3.1 Sistema básico para el filtrado en tiempo discreto de señales en tiempo continuo.

Este proyecto considera el diseño e implementación de filtros FIR en tiempo discreto que se van a utilizar para filtrar una señal en tiempo continuo, empleando la configuración básica de la figura 3.1. Los parámetros del filtro son ingresados por medio de un teclado al DSP, de tal forma que con estos valores mediante programación se encuentre la respuesta al sistema deseado.



Para el diseño de filtros digitales, es necesario conocer los parámetros que gobiernan a cada uno de los tipos de filtros, con el objetivo de facilitar la comprensión del método de diseño a utilizarse.



### a) FILTRO PASA BAJO

En el diseño e implementación de filtros digitales, los filtros pasa bajo son los más importantes en relación a los demás filtros fundamentales (pasa alto, pasa banda y rechaza banda), ya que; como se analizó en el capítulo 2 (sección 2.3.2) a partir de los coeficientes de un filtro pasa bajo se puede encontrar los coeficientes de un filtro pasa alto, pasa banda o rechaza banda. Es por esta razón que los filtros pasa bajo tendrán un análisis más detallado, en tanto que, en los demás filtros únicamente se analizarán los parámetros y los métodos de conversión.

Los parámetros de un filtro pasa bajo son:

■ Frecuencia de paso	$f_p$	[hertz]
■ Frecuencia de supresión	$f_s$	[hertz]
■ Frecuencia de muestreo	$f_m$	[hertz]
■ Nivel de rizado en la banda de paso	$\delta_p$	
■ Nivel de rizado en la banda de supresión	$\delta_s$	

Debido al carácter simétrico de los coeficientes resulta que los rizados tanto en la banda de paso  $\delta_p$  como en la supresora  $\delta_s$  se mantienen, evidenciando la aplicación del efecto *Gibbs*. Este fenómeno no desaparece con la longitud del filtro. Es decir que para el diseño de filtros digitales mediante la utilización de ventanas se cumple que el rizado del filtro es el valor mínimo entre  $\delta_p$  y  $\delta_s$

$$\delta = \min(\delta_p, \delta_s)$$

Para determinar el rango de variación de los diferentes parámetros es necesario tener en cuenta las siguientes limitaciones:

- Errores de cuantización (característica propia del DSP) los cuales permiten el truncamiento o redondeo, en la medida que el error sea aceptable.
- Limitaciones propias de los parámetros (característica dada en la teoría de diseño).

### ***Parámetros de variación del factor de rizado***

El factor de rizado es una característica de los filtros, que determina las variaciones máximas de las ondulaciones tanto en la banda de paso como en la de supresión en los filtros pasa bajo. En un filtro ideal el rizado debe ser "*plano*", es decir de valor cero (el valor cero es ideal, no representa un valor a considerarse en el diseño); es por esto que el mejor rizado será aquel que tenga el menor valor positivo.

Debido a que el microprocesador ADSP-2181 trabaja con 16 bits, el *valor mínimo* de rizado que se puede representar es  $\delta = 0.0001$ , este valor permite el menor error de truncamiento en el sistema.

El formato numérico (0.16) permite la obtención máxima de números fraccionarios ya que posee 16 bits de mantisa, el valor mínimo que puede representar es  $2^{-16} = 0.0000152588.....$  y los saltos entre dos representaciones también está dado por este valor.

*Representación binaria*

*Valor cuantificado*

*Formato (0.16)*

0000000000000000  
0000000000000001  
0000000000000010  
0000000000000011

0.0000000000  
0.0000152588  
0.0000305176  
0.0000457764

0000000000000100	0.0000610352
0000000000000101	0.0000762939
0000000000000110	0.0000915527
<b>0000000000000111</b>	<b>0.0001068115</b>
0000000000001000	0.0001220703
0000000000001001	0.0001373291
0000000000001010	0.0001525875
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....

Así, la representación en formato 0.16 de  $\delta = 0.0001$  es:

0000000000000111

MANTISA

$\delta = 0.00010681152$

Obteniéndose un error de truncamiento igual a:

$$E_t = Q(x) - x$$

$$E_t = 0.00010681152 - 0.0001$$

$$E_t = 0.0000068115$$

El valor de atenuación expresado en decibelios [dB] es:

$$\text{Atenuación [dB]} = -20 \log(0.0001) = 80 \text{ [dB]}$$

En general, la manera más común de representar el factor de rizado es en decibelios, por lo cual, mediante las fórmulas de Kaiser (ecuación 2.31) el factor de rizado se relaciona con la atenuación en [dB] mediante:

$$A = -20 \log(\delta) \text{ [dB]} \quad (A \geq 0)$$

de donde el *limite inferior* de A ( $A = 0$ ); hace que el factor de rizado no puede ser mayor que 1, ya que para valores mayores el valor de la atenuación es negativa.

Por otra parte en la ecuación (2.33) se determinó el orden del filtro  $M$  mediante la ecuación:

$$M = \frac{A - 8}{2.285 * \Delta\omega}$$

Por lo que, para que el orden del filtro tome valores positivos se debe cumplir que:

$$A - 8 > 0$$

$$A > 8 \quad [\text{dB}]$$

El rango de variación de la atenuación en la banda de supresión que se utilizará en el presente Proyecto de Titulación está limitado por propiedades de diseño en el método de Kaiser (sección 2.3) y por limitaciones que posee el módulo de desarrollo EZ-Kit Lite, este rango es:

$$20 < A \leq 60 \quad [\text{dB}]$$

El rango de valores previamente establecidos entre los que varía la atenuación (valores en decibelios), son ingresados por medio de un teclado al DSP en pasos de 1 o 10; es decir, desde 20 [dB], 21 [dB], ..... hasta 60 [dB] almacenados en la memoria del microprocesador.

En la figura 3.2 se ilustra el ingreso de la atenuación al DSP, una vez establecido el tipo de filtro que se va a realizar, se selecciona el parámetro que se va a ingresar, mediante la tecla de PARÁMETROS; posteriormente mediante las flechas (ARRIBA-ABAJO) se ajusta al valor requerido.

Cuando se desea pasar de un valor menor como por ejemplo 20 [dB] hasta un valor mayor como 60 [dB], o viceversa, evitando pulsar constantemente las flechas (ARRIBA-ABAJO) se puede realizar saltos (en incremento de 10); ajustando la tecla RESOLUCIÓN a dicho factor, lo que permitirá que las teclas (ARRIBA-ABAJO) sigan la siguiente secuencia 20 [dB], 30 [dB], .....60 [dB].

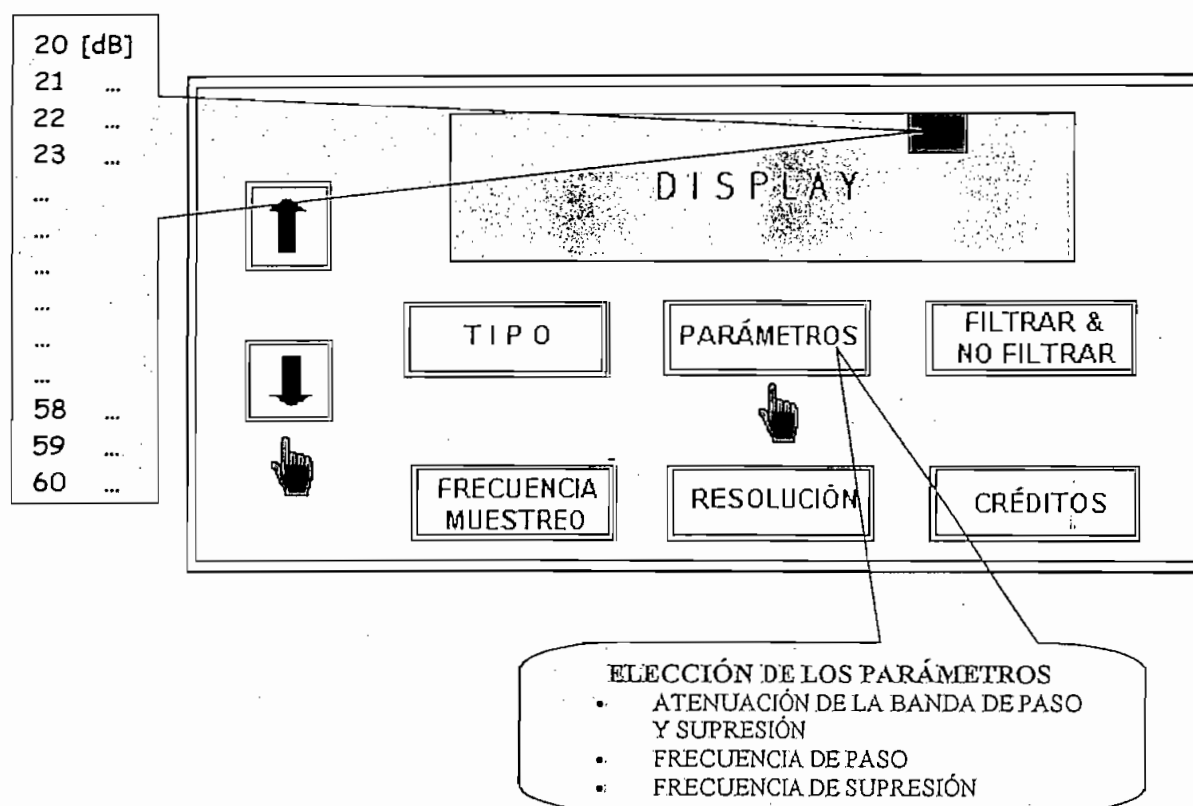


FIGURA 3.2 Ingreso del factor de rizado mediante un teclado al DSP.

La determinación de los parámetros entre los que varía la frecuencia de paso y supresión dependen en definitiva de la aplicación a la que es sometida el filtro, es por esto que los parámetros no son definidos, pero sí sus limitaciones. Los valores establecidos para la implementación y diseño son elegidos mediante el rango de frecuencias más comúnmente utilizado en diferentes aplicaciones, es por esto se ha asumido como valor máximo 20000 Hz.

La frecuencia de muestreo es un parámetro muy importante en el diseño de filtros, ya que determina la frecuencia máxima de *Nyquist*, mediante el teorema del muestreo. La frecuencia de muestreo está limitada por los valores de frecuencia que origina el cristal del conversor digital-analógico cuyos valores son: 5.5125 kHz, 6.6125 kHz, 8 kHz, 9.6 kHz, 11.025 kHz, 16 kHz, 18.9 kHz, 22.05 kHz, 27.42857 kHz, 32 kHz, 33.075 kHz, 37.8 kHz, 44.1 kHz y 48 kHz.

Como la frecuencia de muestreo máxima es 48000 [Hz], la ganancia del sistema completo es cero por encima de los 24000 [Hz], debido al convertor digital-analógico (DAC) ideal, cuyo fundamento es el teorema del muestreo.

La figura 3.3 ilustra la determinación de los parámetros de un filtro pasa bajo.

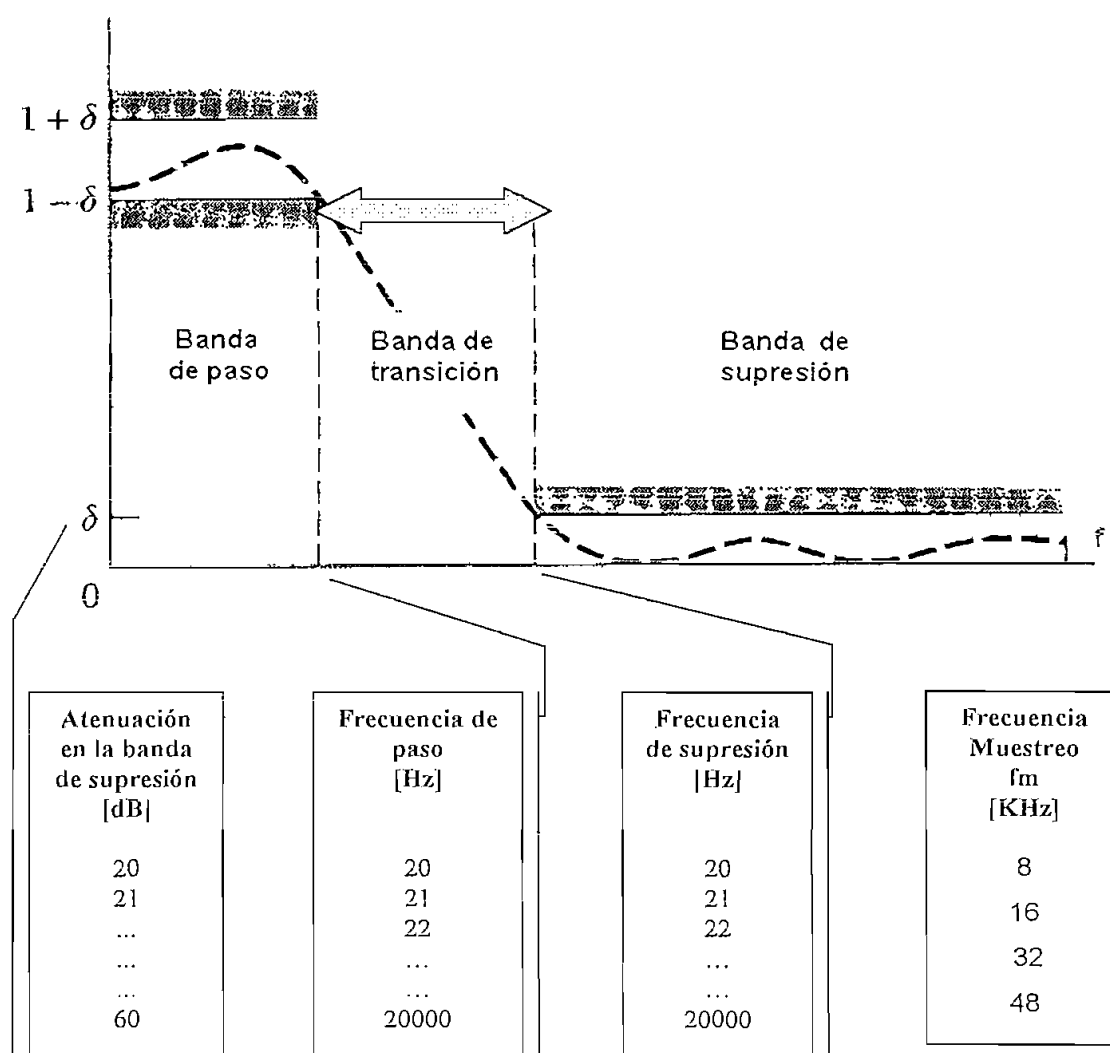


Figura 3.3 Rango y determinación de los parámetros de un filtro pasa bajo

Otro factor importante que se debe tener en cuenta es el factor de *roll-off*, este valor está representado en la ecuación (2.30) mediante la diferencia entre la frecuencia de supresión y la de corte

$$\Delta\omega = \omega_s - \omega_p$$

el factor de *roll-off*,  $\Delta\omega$  afecta al orden del filtro, mediante la ecuación (2.33)

$$M = \frac{A - 8}{2.285 * \Delta\omega}$$

para evitar el desbordamiento en el registro en donde se almacena el orden del filtro, la atenuación y el factor de *roll-off* tendrán un papel importante para la determinación de este parámetro. Si se selecciona un valor bajo de atenuación el factor de *roll-off* que se podrá obtener será pequeño

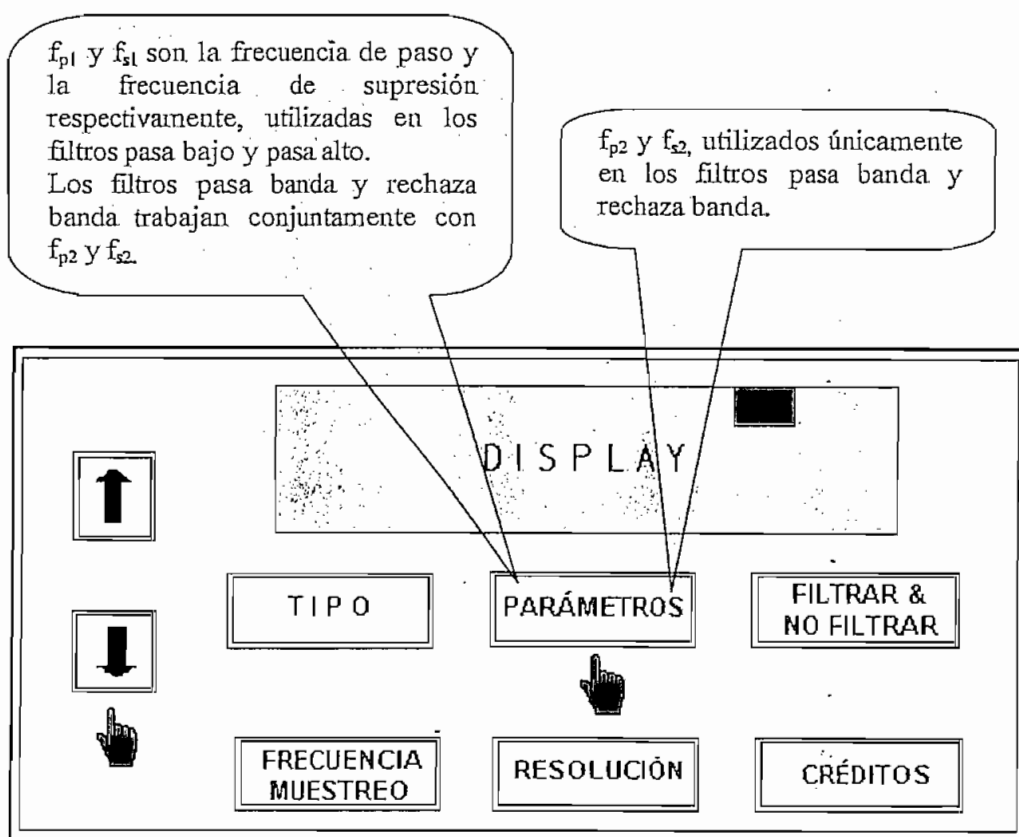


FIGURA 3.4 Ingreso de las frecuencias de corte, supresión y muestreo mediante un teclado al DSP.

El ingreso de estos parámetros se los realiza de forma similar a la atenuación, es decir una vez que se ha seleccionado el filtro pasa bajo mediante la tecla TIPO

(figura 3.4), se escoge el parámetro a ingresar (atenuación, frecuencia de corte, frecuencia de supresión) y mediante las flechas o por medio de saltos se ajusta el valor deseado. En la sección 3.1.1.3 se tiene un análisis más detallado de la utilización del teclado.

Una vez establecidos los parámetros y de acuerdo a las especificaciones establecidas en la sección 2.2.1, el sistema completo debe tener las siguientes propiedades para determinada frecuencia de muestreo  $f_m$ .

1. La ganancia  $|H_{\text{eff}}(j\Omega)|$  debe diferir de la unidad en un factor de  $\pm \delta_p$  en la banda  $0 \leq \Omega \leq 2\pi f_p$ . Por tanto, la ganancia ideal en la banda de paso es la unidad y varía entre  $(1 + \delta_p)$  y  $(1 - \delta_p)$ . Es decir, en la banda de paso el valor absoluto de la respuesta en frecuencia debe aproximarse a la unidad con un error menor que  $\pm \delta_p$ , siendo  $\omega_p$  la frecuencia de paso normalizada expresada en radianes.

$$(1 - \delta_p) \leq |H(e^{j\omega})| \leq (1 + \delta_p), \quad |\omega| \leq \omega_p \quad (3.1)$$

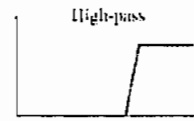
2. La ganancia no debe ser mayor que  $\delta_s$  en la banda de frecuencias  $\Omega \geq 2\pi f_s$ , por lo que la ganancia en la banda de supresión varía entre 0 y  $\delta_s$ . Es decir, en la banda de supresión la respuesta en amplitud debe aproximarse a cero con error menor que  $\delta_s$ , siendo  $\omega_s$  la frecuencia de supresión normalizada expresada en radianes.

$$0 \leq |H(e^{j\omega})| \leq \delta_s, \quad \omega_s \leq |\omega| \leq \pi \quad (3.2)$$

3. Como la frecuencia de muestreo es  $f_m$ , la ganancia del sistema completo es cero por encima de la frecuencia de Nyquist  $F_{\text{nyquist}} \leq \left(\frac{f_m}{2}\right)$ .



## b) FILTRO PASA ALTO



En la figura 3.5 se señalan los parámetros de un filtro pasa alto son:

- Frecuencia de supresión  $f_s$  [hertz]
- Frecuencia de paso  $f_p$  [hertz]
- Frecuencia de muestreo  $f_m$  [hertz]
- Factor de rizado en las bandas de paso y supresión  $\delta$

Cuyas consideraciones y límites de los parámetros mantienen el análisis establecido en los filtros pasa bajo.

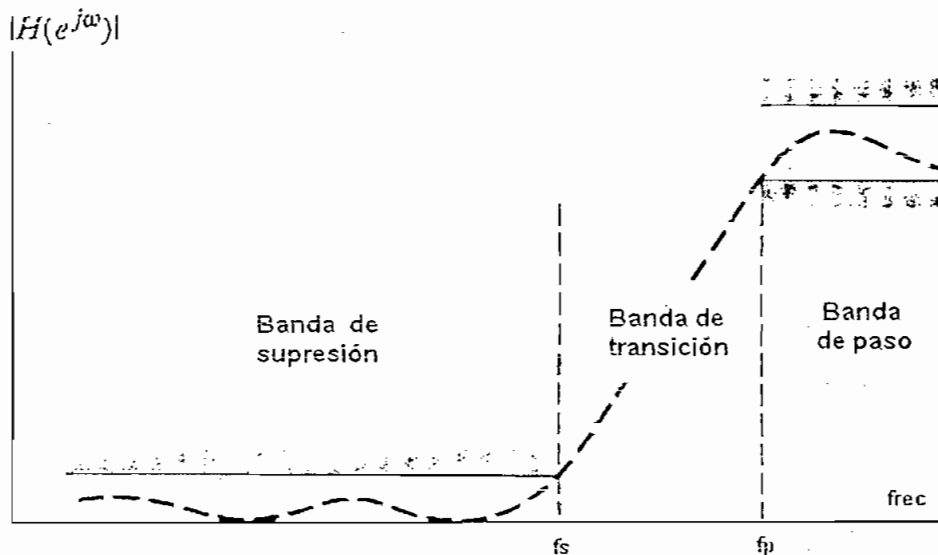


FIGURA 3.5 Parámetros de variación de un filtro pasa alto.



### c) FILTRO PASA BANDA

Los coeficientes de los filtros pasa banda se los puede obtener a partir de un filtro pasa bajo, mediante la conversión determinada en la sección 2.3.2. Los parámetros de un filtro pasa banda son:

- Frecuencia de supresión (1)  $f_{s1}$  [hertz]
- Frecuencia de paso (1)  $f_{p1}$  [hertz]
- Frecuencia de paso (2)  $f_{p2}$  [hertz]
- Frecuencia de supresión (2)  $f_{s2}$  [hertz]
- Frecuencia de muestreo  $f_m$  [hertz]
- Factor de rizado en las bandas de paso y supresión  $\delta$

Las consideraciones y limitaciones establecidas en los filtros pasa bajo se aplican de la misma forma para los filtros pasa banda. En la figura 3.6 se ilustra los parámetros de un filtro pasa banda.

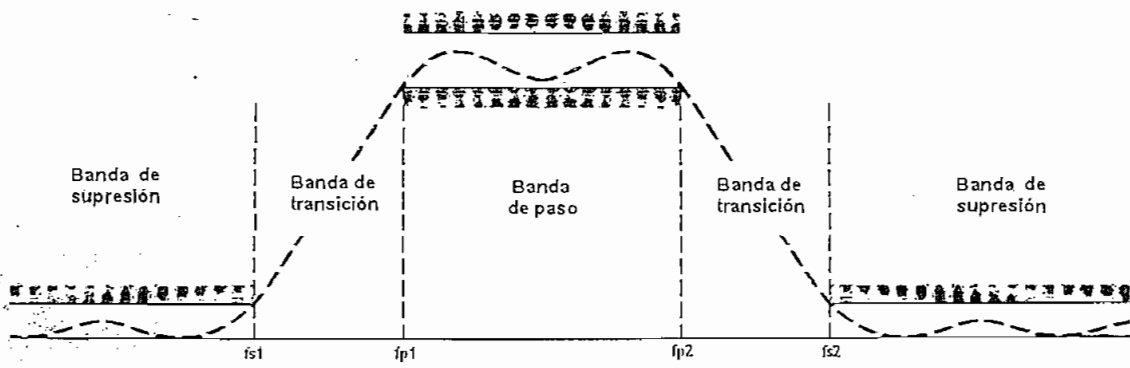
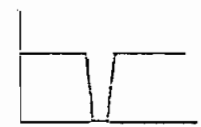


FIGURA 3.6 Parámetros de variación de un filtro pasa banda.



### d) FILTRO RECHAZA BANDA

Los coeficientes de los filtros rechaza banda se los puede obtener a partir de un filtro pasa banda, mediante la conversión establecida en la ecuación (2.37).

$$h_{rb} = \delta[n-\alpha] - \left[ \frac{\sin(\omega_{c2}(n-\alpha))}{\pi(n-\alpha)} - \frac{\sin(\omega_{c1}(n-\alpha))}{\pi(n-\alpha)} \right] * \frac{I_0 \left[ \beta(1 - [(n-\alpha)/\alpha]^2)^{1/2} \right]}{I_0(\beta)}$$

Los parámetros de un filtro rechaza banda son:

- Frecuencia de paso (1)  $f_{p1}$  [hertz]
- Frecuencia de supresión (1)  $f_{s1}$  [hertz]
- Frecuencia de supresión (2)  $f_{s2}$  [hertz]
- Frecuencia de paso (2)  $f_{p2}$  [hertz]
- Frecuencia de muestreo  $f_m$  [hertz]
- Factor de rizado en las bandas de paso y supresión  $\delta$

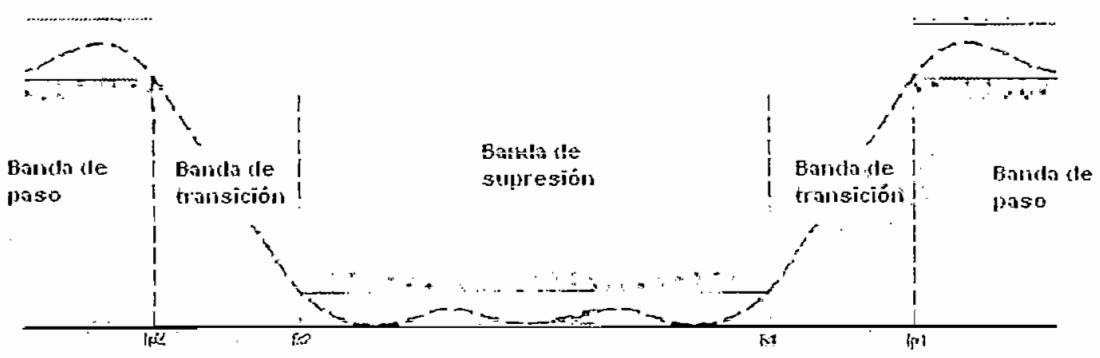


FIGURA 3.7 Parámetros de variación de un filtro rechaza banda.

Las consideraciones y limitaciones establecidas en los filtros pasa bajo se aplican de la misma forma para los filtros rechaza banda. En la figura 3.7 se ilustra los parámetros de un filtro pasa banda.

### 3.1.1.2 Aproximación de las propiedades mediante un sistema causal en tiempo discreto.

Una vez establecido el conjunto de especificaciones, el siguiente paso es determinar la función de transferencia del sistema lineal en tiempo discreto cuya respuesta en frecuencia esté dentro de las tolerancias preestablecidas. Es decir, se trata de un problema de aproximación funcional.

En el diseño de filtros IIR se requiere la aproximación mediante una función racional de  $z$ , mientras que el diseño de filtros FIR implica aproximaciones polinómicas.

Entre 1966 a 1974, Kaiser descubrió que se puede formar una aproximación cuasi-óptima utilizando la función de Bessel modificada de primera clase.

La ventana de Kaiser es un método de aproximación polinómica que utiliza procedimientos en forma cerrada, es decir se basa en la sustitución de las especificaciones de diseño en fórmulas establecidas, por lo que a diferencia de otras técnicas casi no requiere de procesos iterativos o de prueba y error.

De hecho, Kaiser obtuvo mediante amplias experimentaciones numéricas, un conjunto de fórmulas que permiten al diseñador de filtros predecir los valores del orden del filtro y más factores necesarios para cumplir determinadas especificaciones en frecuencia.

En la sección 2.3 se estableció las fórmulas para el diseño de filtros digitales mediante el método de aproximación de la ventana de Kaiser, las cuales son resumidas a continuación.<sup>[5]</sup>

PASA BAJO	PASA ALTO	PASA BANDA	RECHAZA BANDA
$\omega_p = \frac{f_p}{f_m} * 2\pi$	$\omega_s = \frac{f_s}{f_m} * 2\pi$	$\omega_{s1} = \frac{f_s}{f_m} * 2\pi$	$\omega_{p1} = \frac{f_p}{f_m} * 2\pi$
$\omega_s = \frac{f_s}{f_m} * 2\pi$	$\omega_p = \frac{f_p}{f_m} * 2\pi$	$\omega_{p1} = \frac{f_p}{f_m} * 2\pi$	$\omega_{s1} = \frac{f_s}{f_m} * 2\pi$
*****	*****	$\omega_{p2} = \frac{f_p}{f_m} * 2\pi$	$\omega_{s2} = \frac{f_s}{f_m} * 2\pi$
*****	*****	$\omega_{s2} = \frac{f_s}{f_m} * 2\pi$	$\omega_{p2} = \frac{f_p}{f_m} * 2\pi$
$\Delta\omega = \omega_s - \omega_p$	$\Delta\omega = \omega_p - \omega_s$	$\Delta\omega_1 = \omega_{p1} - \omega_{s1}$	$\Delta\omega_1 = \omega_{s1} - \omega_{p1}$
*****	*****	$\Delta\omega_2 = \omega_{p2} - \omega_{s2}$	$\Delta\omega_2 = \omega_{p2} - \omega_{s2}$

Tabla 3.1 Fórmulas para el diseño de filtros digitales

COMUN PARA LOS CUATRO TIPOS DE FILTROS
$\omega_c = \frac{\omega_p + \omega_s}{2}$
$A = -20 \log(\delta)$
$M = \frac{A - 8}{2.285 * \Delta\omega} + 1 \quad (\text{impar})$
$\alpha = \frac{M - 1}{2}$
$w[n] = \frac{I_0 \left[ \beta \sqrt{1 - \left[ \frac{(n - \alpha)}{\alpha} \right]^2} \right]}{I_0(\beta)}$

Tabla 3.2 Fórmulas comunes para los filtros básicos para el diseño

PASA BAJO	$h[n] = \frac{\text{sen}[\omega_c \cdot (n - \alpha)]}{\pi \cdot (n - \alpha)} \cdot w[n]$
PASA ALTO	$h[n] = \delta[n - \alpha] - \frac{\text{sen}[\omega_c \cdot (n - \alpha)]}{\pi \cdot (n - \alpha)} \cdot w[n]$
PASA BANDA	$h[n] = \frac{\text{sen}[\omega_{c2} \cdot (n - \alpha)] - \text{sen}[\omega_{c1} \cdot (n - \alpha)]}{\pi \cdot (n - \alpha)} \cdot w[n]$
RECHAZA BANDA	$h[n] = \delta[n - \alpha] - \frac{\text{sen}[\omega_{c2} \cdot (n - \alpha)] - \text{sen}[\omega_{c1} \cdot (n - \alpha)]}{\pi \cdot (n - \alpha)} \cdot w[n]$

Tabla 3.3 Respuesta impulsiva para cada tipo de filtro

Para la implementación en tiempo real de los filtros digitales FIR, se utiliza el paquete EŽ-KIT Lite ADŠP-2181<sup>[4]</sup> de *Analog Devices*. Este es un microcomputador optimizado para procesamiento digital de señales. La arquitectura del ADSP-2181 está integrada por tres unidades computacionales, las cuales a su vez definen el tipo de operaciones que ésta realiza.

UNIDADES COMPUTACIONALES	TIPO DE OPERACIONES QUE REALIZA
ALU	Operaciones aritméticas ( <i>suma, resta, negación, incremento, decremento</i> ) y Operaciones lógicas ( <i>AND, OR, XOR, NOT</i> ).
MAC	Multiplicaciones y multiplicaciones con sumas.
UNIDAD DE DESPLAZAMIENTO	Desplazamiento de bits para normalizar e indicar diferentes formatos numéricos.

TABLA 3.4 Operaciones que realiza cada unidad computacional de la ADSP-2181<sup>[3]</sup>

De tal forma que si se analiza las fórmulas establecidas para el diseño de filtros digitales mediante la ventana de Kaiser, se observa que a más de sumas, restas y multiplicaciones que se las puede realizar directamente con el ADSP-2181, requieren de operaciones como divisiones, funciones exponenciales, funciones logarítmicas y funciones de Bessel, todas estas operaciones no se las puede realizar directamente en el ADSP-2181. La sección 3.3 corresponde a la programación desarrollada para los filtros, en la cual se analizará algunas operaciones como la división, logaritmos y exponenciales que se pueden obtener mediante subrutinas pre-programadas por Analog Devices. Pero la función de Bessel  $I_0(x)$  de primera clase y de orden cero necesita de evaluaciones y aproximaciones, por lo que puede ser escrita mediante series de expansión de Taylor de la siguiente forma:

$$I_0(x) = \sum_{k=0}^{\infty} \left[ \frac{(x/2)^k}{k!} \right]^2 \quad (3.4)$$

La ventana de Kaiser requiere la evaluación de  $I_0(x)$  sobre el rango del argumento  $0 \leq x \leq \alpha$ . Para la evaluación de la función para cualquier valor de  $x$  se efectúa una subrutina que está basada en recursiones, las cuales evalúan las series de potencias por almacenamiento de algunos términos en la expansión.

Se define el sumatorio parcial de las series como:

$$S_n = \sum_{k=0}^n \left[ \frac{(x/2)^k}{k!} \right]^2 \quad (3.5)$$

Ésta es inicializada para  $S_0 = 1$  y satisface la recursión, para  $n \geq 1$ :

$$S_n = S_{n-1} + D_n \quad \text{donde} \quad D_n = \left[ \frac{(x/2)^n}{n!} \right]^2 \quad (3.6)$$

Entonces,  $D_n$  satisface una recursión para  $n \geq 1$ :

$$D_n = \left[ \frac{x}{2n} \right]^2 D_{n-1} = T_n^2 D_{n-1} \quad \text{donde} \quad T_n = \frac{x}{2n} \quad (3.7)$$

Y ésta es inicializada para  $D_0=1$ . Las iteraciones se detienen cuando los términos de la sucesión llegan a ser más pequeños que los términos acumulados  $S_n$ , que es, cuando:

$$\frac{D_n}{S_n} = \frac{S_n - S_{n-1}}{S_n} < \varepsilon \quad (3.8)$$

Donde  $\varepsilon$  es un número pequeño, que puede ser en el orden de  $\varepsilon=10^{-9}$ .

### 3.1.1.3 Realización del sistema

Un sistema de procesamiento en tiempo real generalmente consiste de un conversor Analógico-Digital y Digital-Analógico, un microprocesador (en este caso ADSP-2181). Las etapas las podemos observar en la figura 3.8. El ADSP-2181 posee un CODEC-1847<sup>[4]</sup> (Codificador-Decodificador) con el que se implementan los conversores. El DSP (donde se implementan los filtros) se comunica con el CODEC usando un puerto del procesador (el procesador ADSP-2181 y la configuración del CODEC serán analizados en la sección 3.2).

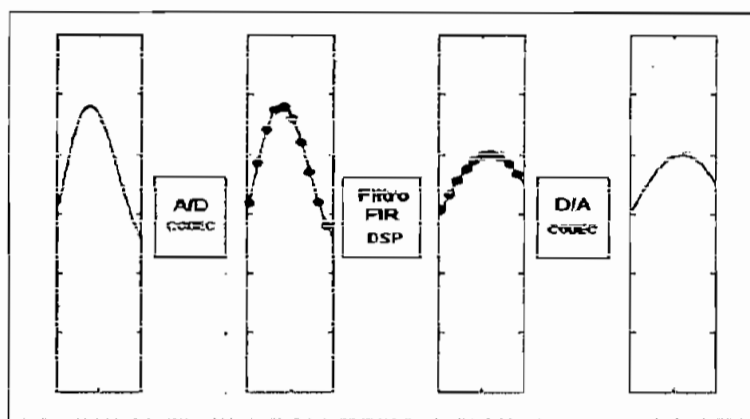


FIGURA 3.8. Sistema de procesamiento en tiempo real sobre el que se implementa el filtro no recursivo FIR

Básicamente el sistema a implementarse (en *hardware*) constará de: un teclado, un display, módulo de desarrollo EZ-KIT Lite ADSP-2181, señal a filtrar y la señal filtrada. En la figura 3.9 se representará el diagrama esquemático del sistema a implementarse.

#### TECLADO

El teclado es la parte física por medio de la cual serán ingresados los parámetros del filtro al microprocesador ADSP-2181.

<sup>1</sup> MM74C923 Anexo 4 determina las características eléctricas, mecánicas, físicas del decodificador de teclado.

Es preciso resaltar los elementos secundarios que son necesarios en la utilización del teclado, como son el decodificador de teclados y el eliminador de rebotes, estos dos elementos se los puede encontrar en el circuito integrado 74C922. El CMOS MM74C922<sup>1</sup> es un decodificador de hasta 16 teclas que provee todo lo necesariamente lógico para decodificar un arreglo de *switches*, además provee de un circuito interno de eliminación de rebotes mediante un simple capacitor externo.



Es preciso resaltar los elementos secundarios que son necesarios en la utilización del teclado, como son el decodificador de teclados y el eliminador de rebotes, estos dos elementos se los puede encontrar en el circuito integrado 74C922. El CMOS MM74C922<sup>1</sup> es un decodificador de hasta 16 teclas que provee todo lo necesariamente lógico para decodificar un arreglo de *switches*, además provee de un circuito interno de eliminación de rebotes mediante un simple capacitor externo.

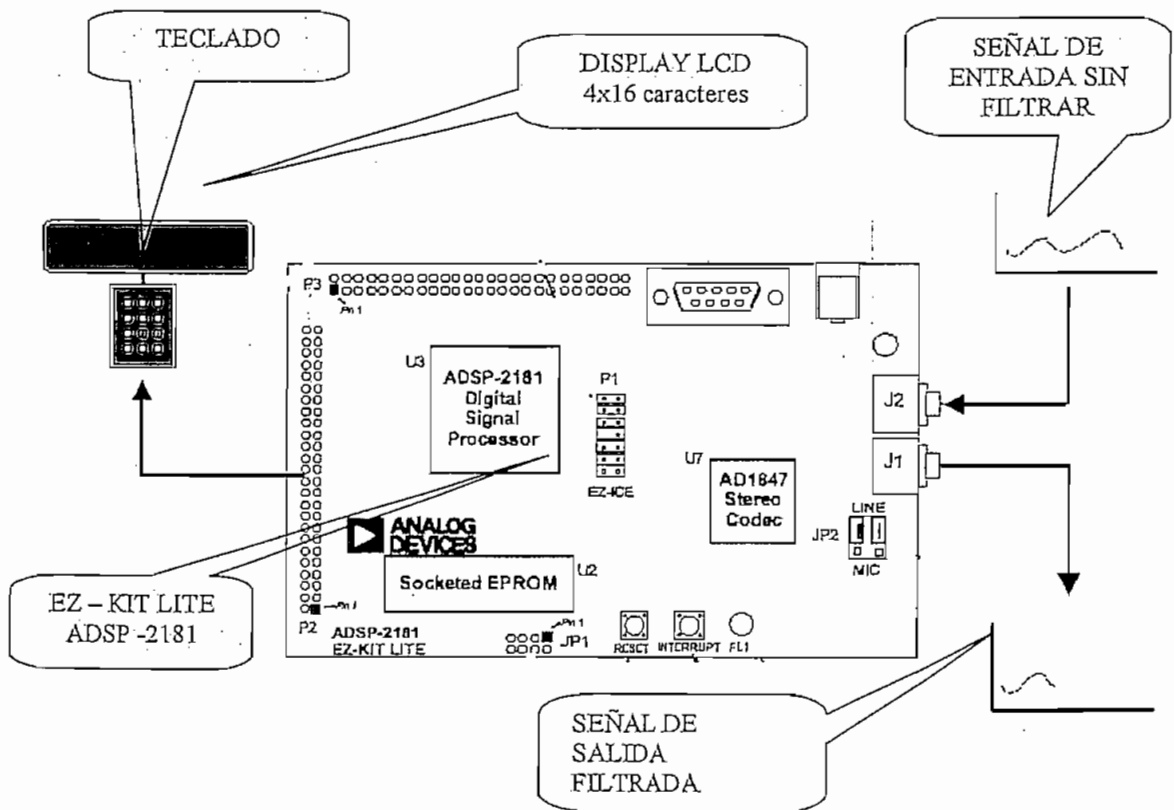




Figura 3.9. Diagrama esquemático del sistema a implementarse.

El teclado consta de una configuración de pulsadores o switches cuya función específicamente es el ingreso de los parámetros del filtro a diseñar.

<sup>1</sup> CMOS MM74C922 en el Anexo 4 se determina las características mecánicas, eléctricas y físicas del decodificador de teclado

<b>PARÁMETROS</b>	PARÁMETROS	Permite escoger los parámetro del filtro a implementarse como: atenuación de las bandas de paso y supresión en [dB], y las frecuencias de paso y supresión. En el caso de un filtro pasa banda o rechaza banda permite ingresar los dos valores de frecuencia de paso, así como de la frecuencia de supresión.
<b>TIPQ</b>	TIPO	Permite escoger el tipo de filtro (pasa bajo, pasa alto, pasa banda y rechaza banda)
<b>RESOLUCIÓN</b>	RESOLUCIÓN	Permite realizar saltos (en pasos de 1, 10, 100 ó 1000) entre los valores del parámetro seleccionado con el fin de evitar las pulsaciones continuas de ARRIBA/ABAJO, en el caso que el salto sea considerable.
	ARRIBA	Incrementa el valor del parámetro seleccionado en pasos de 1, 10, 100 ó 1000.
	ABAJO	Decrementa el valor del parámetro seleccionado en pasos de 1, 10 100 ó 1000
<b>FRECUENCIA MUESTREO</b>	FRECUENCIA DE MUESTREO	Permite seleccionar un valor de frecuencia de muestreo entre: 8, 16, 32 o 48 KHz.
<b>FILTRAR &amp; NO FILTRAR</b>	FILTRAR & NO FILTRAR	Permite filtrar o no filtrar la señal de entrada, para fines de comparación entre las dos señales.
<b>CRÉDITOS</b>	CRÉDITOS	Pantalla de presentación y los nombres de los autores del presenta proyecto.

La distribución física del teclado se puede observar en la figura 310.

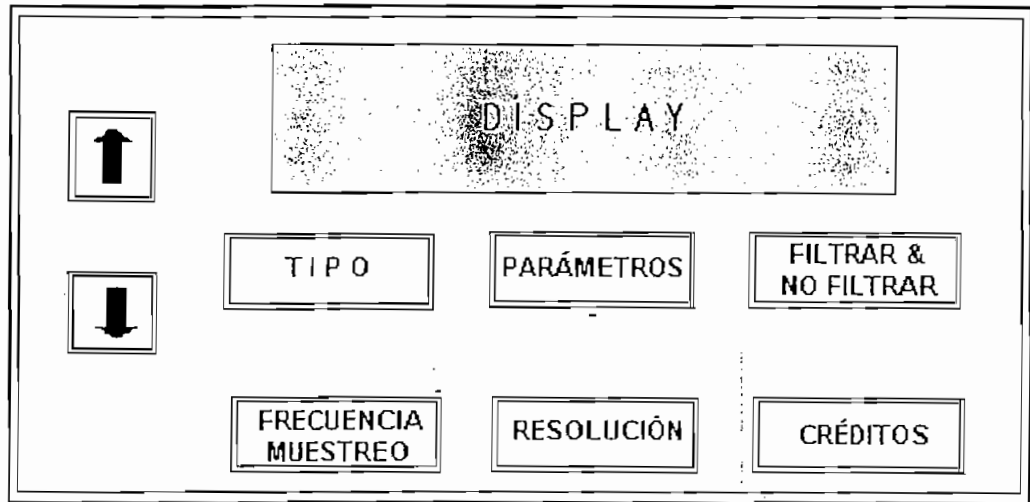


FIGURA 3.10 Distribución física del teclado mediante el cual se ingresa los parámetros del filtro.

### DISPLAY

El *display* es un elemento físico de salida de información, el cual permite visualizar los datos que son ingresados por medio del teclado (parámetros del filtro).



Figura 3.11 *Display* de LCD de matriz ( 4 líneas x 16 caracteres )

El *display* utilizado es un LCD de matriz ( 4 líneas x 16 caracteres ).

En el Anexo 4 se encuentra información mas detallada sobre las especificaciones del *display* mencionado.

## **MÓDULO DE DESARROLLO EZ KIT Lite ADSP-2181<sup>[2]</sup>**

El EZ-KIT Lite es el módulo que realiza todo el trabajo en la implementación de los filtros. El *software* del EZ-KIT Lite incluye algunos utilitarios como son: un *Assembler*, un *Linker*, un *Simulador*, un *Prom Splitter*, un Programa *Host*, el archivo de descripción del EZ-Lab y varios archivos con ejemplos y demostraciones.

Para ejecutar este *software* se necesita de ciertos requisitos que debe cumplir un computador:

- PC 386 (o superior) con un disco duro, drive de alta densidad, tarjeta de video color, monitor VGA, y un mínimo de 2 MB extendidas en RAM.
- MB de espacio libre en el disco para instalar el *software*.
- MS-DOS 3.1 o mayor.
- *Microsoft Windows* 3.1 o superior.

## **3.2 EL MICROPROCESADOR ADSP-2181<sup>[1]</sup>**

El ADSP-21xx es una familia de procesadores programables de un solo *chip*, los cuales utilizan la arquitectura modificada de *Harvard* con buses separados para datos e instrucciones.

El ADSP-2100 es un microprocesador que pertenece a dicha familia en donde los buses se extienden fuera del *chip*, por lo que requieren de memorias externas de datos y programas, además contiene tres unidades computacionales independientes: una unidad aritmética lógica, un multiplicador acumulador y una unidad de desplazamiento. En la figura 3.12 se indica la arquitectura interna de

ADSP-2100. Estas unidades computacionales procesan directamente 16 bits de datos y son usados para cálculos de precisión múltiple.

### 3.2.1 DESCRIPCIÓN GENERAL

El ADSP-2181 de *Analog Devices* es un microprocesador de un solo *chip* optimizado para procesamiento digital de señales y otras aplicaciones de procesamiento numérico de alta velocidad.

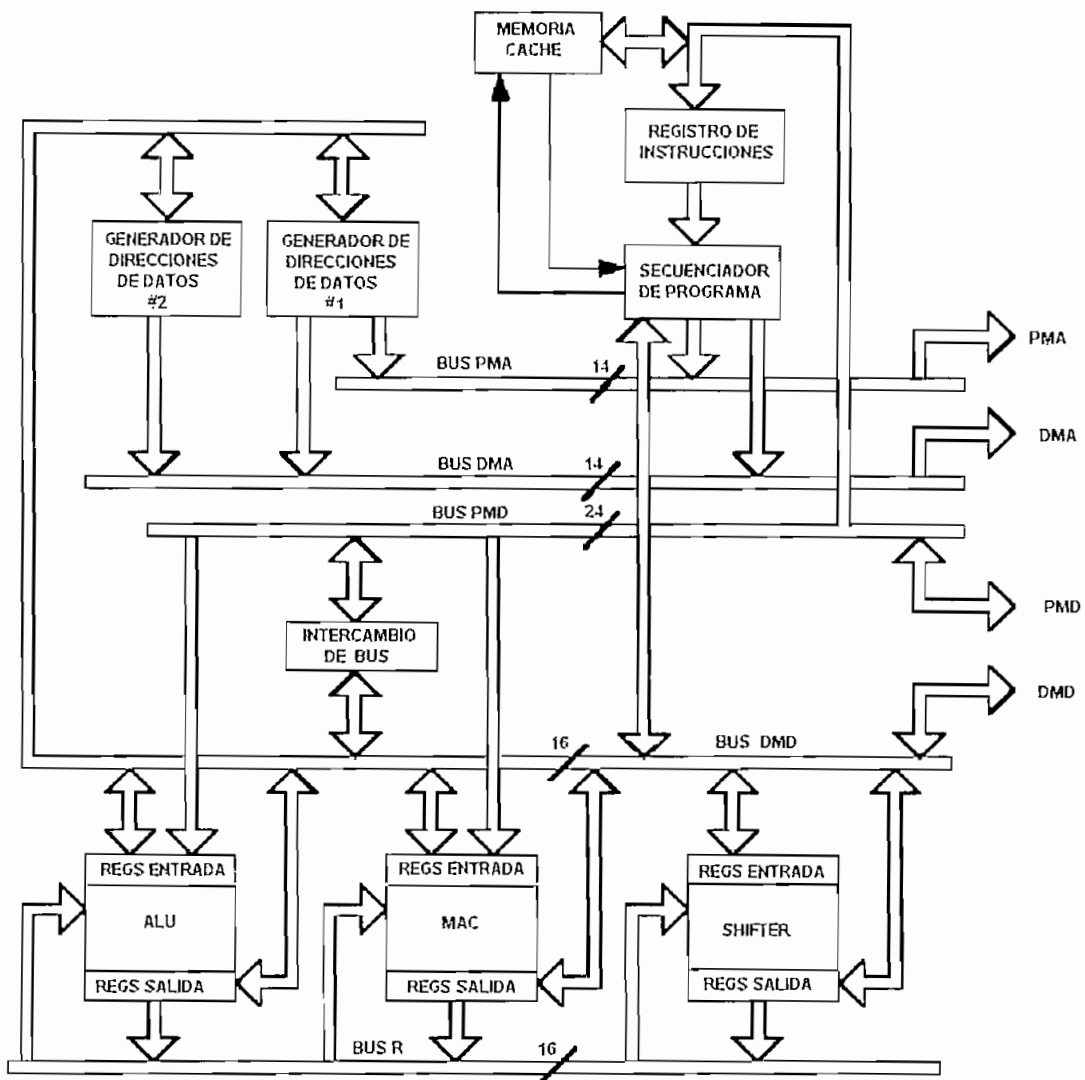


Figura 3.12 Arquitectura Interna del Microprocesador ADSP-2100 [1]

Este microprocesador combina la arquitectura base de la familia ADSP-2100, con dos puertos seriales, un puerto interno de 16 bits DMA, un puerto externo de 1 Byte (8bits) BDMA, un temporizador programable, banderas de estado accesibles al usuario y líneas de interrupciones, como se ilustra en la figura 3.13.

Toda esta circuitería puede operar en modo de bajo consumo de potencia, es decir con una sola fuente de alimentación de +5 VDC para satisfacer las necesidades de consumo en equipos portátiles con baterías. Fabricado con tecnología CMOS, el ADSP-2181 opera con un ciclo de instrucción de 30 ns y ejecuta 33 MIPS, en el que todas las instrucciones requieren un solo ciclo del procesador.

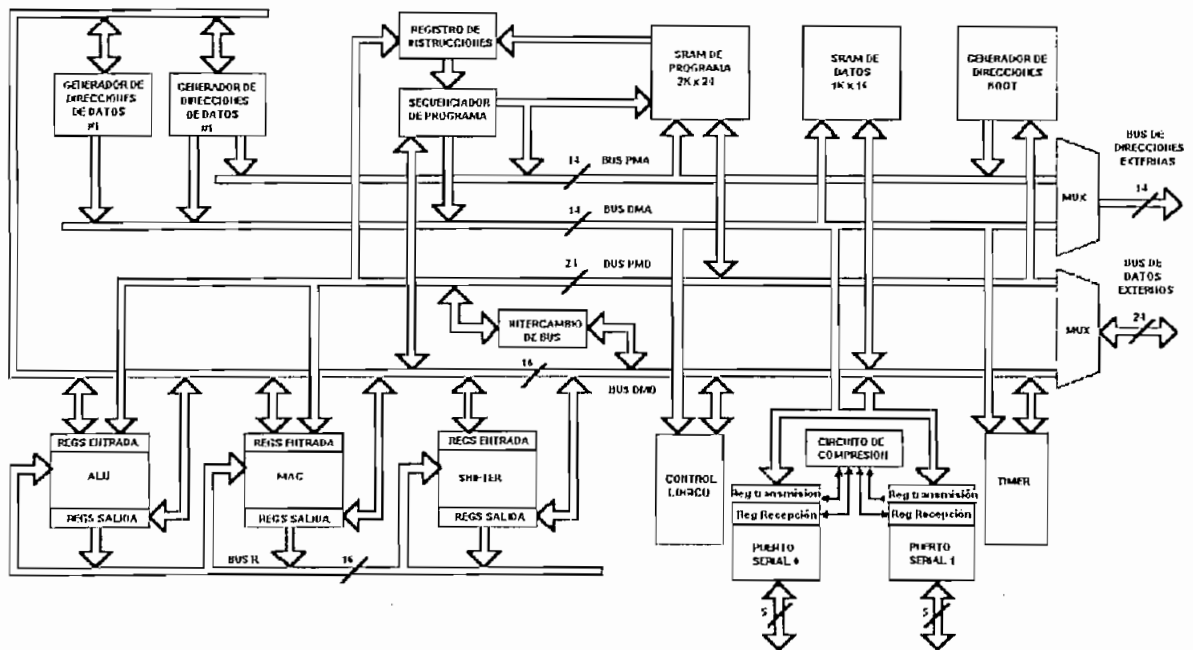


Figura 3.13 Arquitectura Interna del ADSP-2181 [1]

Tiene 80 Kbytes de memoria RAM configuradas de la siguiente manera: 16.000 palabras de 16 bits para memoria de datos y 16.000 palabras de 24 bits para memoria de programas. El microprocesador ADSP-2181 soporta nuevas instrucciones que incluyen manipulaciones de bit, nuevas constantes de la ALU,

nuevas instrucciones de multiplicación, transferencia de memorias y control de interrupciones.

### 3.2.2 ARQUITECTURA INTERNA

El ADSP-2181 es un procesador de 16 bits de punto fijo y admite formatos de datos enteros y fraccionarios en complemento de dos. El ADSP-2181 comparte los elementos básicos de *hardware* de la arquitectura interna de la familia ADSP-21xx. Estos elementos son:

1. ALU (*Arithmetic–Logic Unit*, Unidad Aritmética Lógica)
2. MAC (*Multiplier-Accumulator*, Multiplicador – Acumulador)
3. Unidad de desplazamiento o *Shifter*
4. DAGs (*Data Address Generators*, Generadores de Direcciones de Datos)
5. Secuenciador de Programas

Se obtiene una transferencia de datos eficiente con el uso de cinco (5) buses internos:

- Bus PMA (*Program Memory Address*, Direcciones de Memoria de Programa)
- Bus PMD (*Program Memory Data*, Datos de Memoria de Programas)
- Bus DMA (*Data Memory Address*, Direcciones de Memoria de Datos)
- Bus DMD (*Data Memory Data*, Datos de Memoria de Datos)
- Bus R (*Result*, Resultados)

Las tres unidades computacionales (ALU, MAC, *Shifter*) contienen registros de entrada y de salida los cuales son accesibles desde el bus interno DMD. Las operaciones computacionales generalmente toman sus operandos desde la entrada de los registros y cargan el resultado dentro de los registros de salida. Los registros actúan como un punto intermedio para los datos entre la memoria y la circuitería computacional.

Dos generadores de direcciones de datos y un potente secuenciador de programas permiten el uso eficiente de las unidades computacionales. Los generadores de direcciones de datos proveen de direcciones de memoria cuando la memoria de datos transfiere hacia o desde los registros de entrada y salida. El DAG1 puede generar direcciones hacia la memoria de datos, mientras que el DAG2 genera direcciones hacia la memoria de datos o la memoria de programas. Los dos DAGs son independientes y permiten el acceso simultáneo de datos almacenados en la memoria de programas y en memoria de datos.

Los buses PMA y DMA son usados internamente por las direcciones asociadas con la memoria de Programas y de Datos. Los buses PMD y DMD son usados por los datos asociados con los espacios de memoria. Estos dos pares de buses son multiplexados fuera del *chip* hacia direcciones externas y buses de datos externos.

El bus PMA tiene una extensión de 14 bits permitiendo el acceso directo hacia las 16K palabras de instrucciones combinadas entre códigos y datos. El bus PMD tiene una extensión de 24 bits alojando los 24 bits de instrucciones. El bus DMA con extensión de 14 bits permite el acceso directo hacia las 16K palabras de datos. El bus DMD tiene una extensión de 16 bit y se encarga de dar una ruta para los contenidos de cualquier registro en el procesador a ser transferido hacia cualquier otro registro o hacia una memoria de datos externa.

El bus R, es un bus interno que sirve para transferir directamente los resultados entre las diferentes unidades computacionales. Las direcciones de memoria de datos se obtienen desde dos fuentes: un valor absoluto especificado en el código



de instrucciones (direccionamiento directo) o la salida de un DAG (direccionamiento indirecto). Solo el direccionamiento indirecto es soportado para datos sacados desde memoria de programas.

### 3.2.3 UNIDAD ARITMÉTICA LÓGICA

La ALU permite la ejecución de funciones aritméticas y lógicas. Las funciones aritméticas son sumas, restas, negación, incremento, decremento, y valor absoluto. Las funciones lógicas son AND, OR, XOR (OR exclusivo) y NOT. La figura 3.14 muestra el diagrama de bloques de la ALU.

La ALU tiene una extensión de 16 bits; consta de dos puertos de entrada (X) e (Y) cada uno de 16 bits, y un puerto de salida R. La ALU acepta una señal de acarreo (CI) la cual corresponde al bit de acarreo (*carry bit*) del registro de estado de la aritmética del procesador (registro ASTAT).

El puerto de entrada (X) de la ALU puede aceptar datos desde dos fuentes: los archivos del registro (AX) o del bus de resultados R. El bus R conecta los registros de salida de todas las unidades computacionales, permitiendo que sean usados como operandos de entrada. El archivo de registros (AX) está constituido de dos registros (AX0, AX1). Estos registros (AX) son de lectura y escritura desde el bus DMD. El puerto de entrada (Y) de la ALU también puede aceptar datos desde dos fuentes: el archivo de registros (AY) y el registro de realimentación de la ALU también llamado registro (AF).

El archivo de registros (AY) está formado de dos registros (AY0 y AY1). Estos registros son de lectura y escritura desde el bus DMD y solo de escritura desde el bus PMD.

La ALU genera seis señales de estado también llamadas banderas, las cuales permiten establecer el correcto funcionamiento de la unidad aritmética lógica. Las banderas de la ALU con sus respectivas funciones son descritas a continuación:

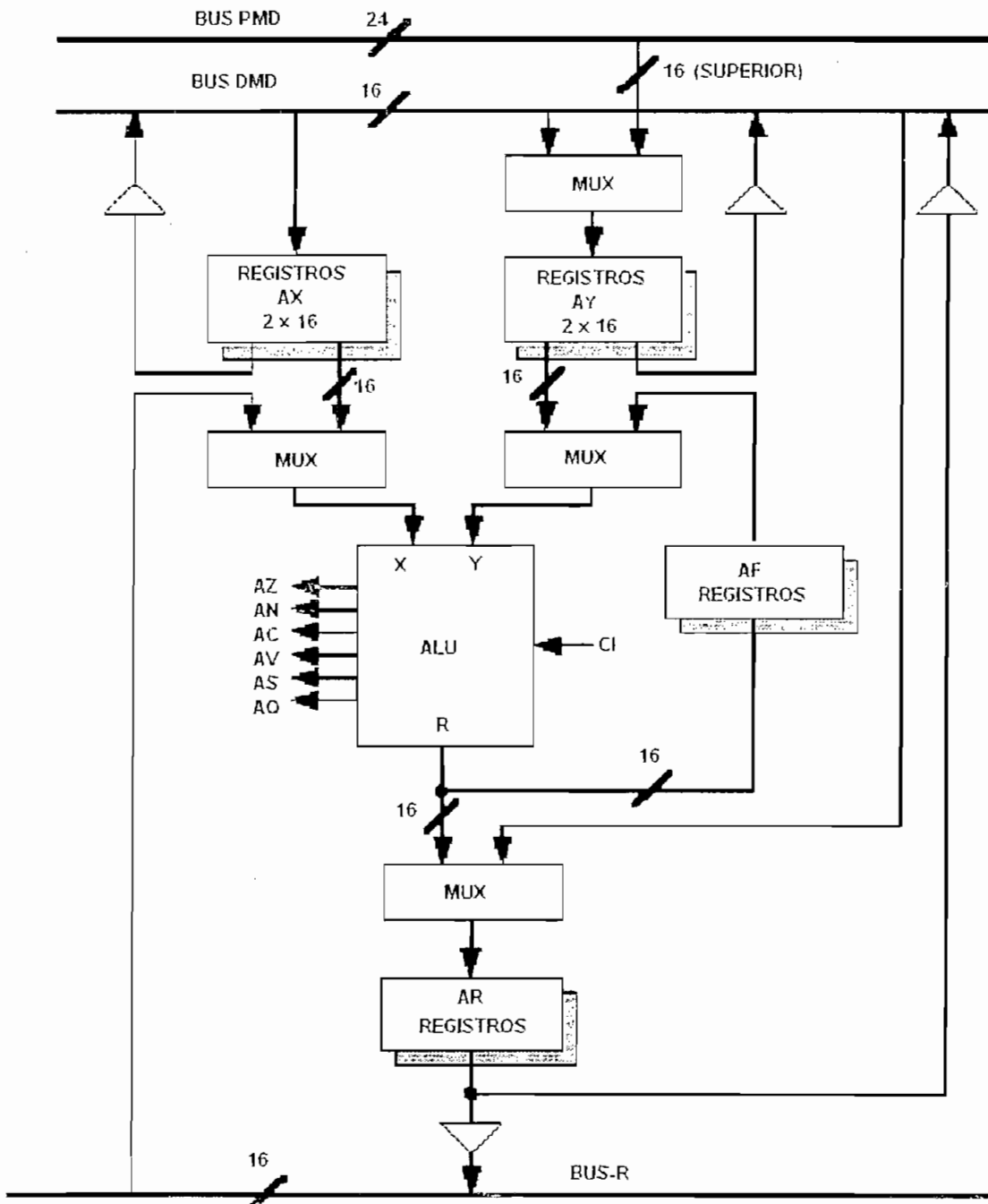


Figura 3.14 Diagrama de Bloques de la ALU [1]

Todas las señales de estado aritméticas son almacenadas dentro de los registros de estado aritmético (ASTAT) al final de un ciclo.

BANDERA	NOMBRE	DEFINICIÓN
AZ	Cero	Es una NOR lógica de todos los bits del registro de resultados de la ALU. Es verdadero si las salidas de la ALU son iguales a cero.
AN	Negativo	Signo del bit de resultado de la ALU. Es verdadero si las salidas de la ALU están en negativo.
AV	Desbordamiento	OR-Exclusiva de las salidas del bit de acarreo de los dos sumandos más significativos. Verdadero si la ALU se desborda.
AC	Acarreo	Salida de bit de acarreo de los sumandos más significativos.
AS	Signo	Signo del bit de el puerto de entrada X de la ALU. Afectada solo por la instrucción ABS.
AQ	Cuociente	Bit de cuociente generado solo por las instrucciones DIVS y DIVQ.

**Tabla 3.5** Descripción de las funciones de las banderas de las ALU [1]

La salida de la ALU es almacenada dentro del registro (AF) o del registro de resultados de la ALU (AR). El registro (AF) es un registro interno de la ALU el cual permite que los resultados sean usados directamente como entradas del registro (Y). El registro (AR) puede manejar ambos buses el DMD y el R.

Cualquiera de los registros asociados con la ALU puede ser de lectura y escritura en un mismo ciclo de máquina. Los registros son leídos al iniciar el ciclo y escritos al finalizar el mismo. Por lo tanto, cuando se lee un registro, se está leyendo el valor almacenado al final del ciclo anterior. Un nuevo valor escrito en un registro no puede ser leído hasta iniciar el siguiente ciclo. Esto permite que un registro de entrada proporcione un operando para la ALU al empezar el ciclo, este registro que puede ser actualizado con el siguiente operando de la memoria al final del mismo. Esto también permite que el registro de resultados sea almacenado en memoria y luego actualizarlo con un nuevo resultado en un mismo ciclo.

En general las fuentes de los registros de entrada y de salida de la ALU son indicadas a continuación:

FUENTE PARA EL PUERTO DE ENTRADA (X)	FUENTE PARA EL PUERTO DE ENTRADA (Y)	DESTINO PARA EL PUERTO DE SALIDA (R)
AX0, AX1	AY0, AY1	AR
AR	AF	AF
MR0, MR1, MR2		
SR0, SR1		

Tabla 3.6 Fuentes de los registros de entrada y salida de la ALU [1]

MR0, MR1 y MR2 son registros de resultados del multiplicador / acumulador; SR0 y SR1 son registros de resultado del *shifter*.

La ALU contiene un banco duplicado de registros AR, AX, AF y AY, también señalados en la figura 3.14. Solo uno de estos bancos es accesible a la vez. La selección del banco primario o secundario es controlado por el bit 0 en el modo de registros de estado del procesador (MSTAT). Si el bit es 0, el banco primario es seleccionado, si es 1, se selecciona el banco secundario.

Las funciones aritméticas y lógicas de la ALU son listadas a continuación:

$R = X + Y$	Suma los operandos X e Y
$R = X + Y + Cl$	Suma los operandos X e Y, y el valor del bit de acarreo
$R = X - Y$	Resta el operando Y del X
$R = X - Y + Cl - 1$	Resta el operando Y del X con "debe" ( <i>borrow</i> )
$R = Y - X$	Resta el operando X del Y
$R = Y - X + Cl - 1$	Resta el operando X del Y con "debe" ( <i>borrow</i> )
$R = -X$	Negación del operando X (complemento de dos)
$R = -Y$	Negación del operando Y (complemento de dos)
$R = Y + 1$	Incremento del operando Y
$R = Y - 1$	Decremento del operando Y

R = PASS X	Pasa el operando X al resultado sin cambiarlo
R = PASS Y	Pasa el operando Y al resultado sin cambiarlo
R = 0 (PASS 0)	Coloca un cero en el resultado
R = ABS X	Valor absoluto del operando X
R = X AND Y	Función lógica AND de los operandos X e Y
R = X OR Y	Función lógica OR de los operandos X e Y
R = X XOR Y	Función lógica XOR de los operandos X e Y
R = NOT X	NOT lógico del operando X ( complemento de uno )
R = NOT Y	NOT lógico del operando Y ( complemento de uno )

### 3.2.4 MULTIPLICADOR ACUMULADOR

El MAC provee de una rápida velocidad de multiplicación, multiplicación con sumas acumulativas, multiplicación con restas acumulativas, saturación y funciones de encerado o borrado. El multiplicador tiene dos puertos de entrada de 16 bits (X) e (Y), y un puerto de salida del producto de 32 bits llamado puerto (P). El producto de 32 bits es transferido a un sumador / restador de 40 bits el cual suma o resta producto de salida del multiplicador con el contenido del registro de resultados de la multiplicación también llamado registro (MR) ó a su vez transfiere el nuevo producto directamente hacia el registro (MR). El registro (MR) tiene una capacidad de 40 bits, los cuales están divididos en tres pequeños registros: (MR0) y (MR1) con una extensión de 16 bits y (MR2) que tiene 8 bits de extensión. En la figura 3.15 se indica detalladamente el diagrama de bloques de el MAC.

El sumador / restador tiene mucho más que 32 bits permitiendo el desbordamiento inmediato en operaciones de multiplicación / acumulación. El bit de estado de desbordamiento en la multiplicación (MV) es seteado cuando el acumulador se ha desbordado sobre los 32 bits de límite, esto es, cuando hay bits significativos (no de signo) en los ocho bits superiores del registro MR (basados en la aritmética de complemento de dos).

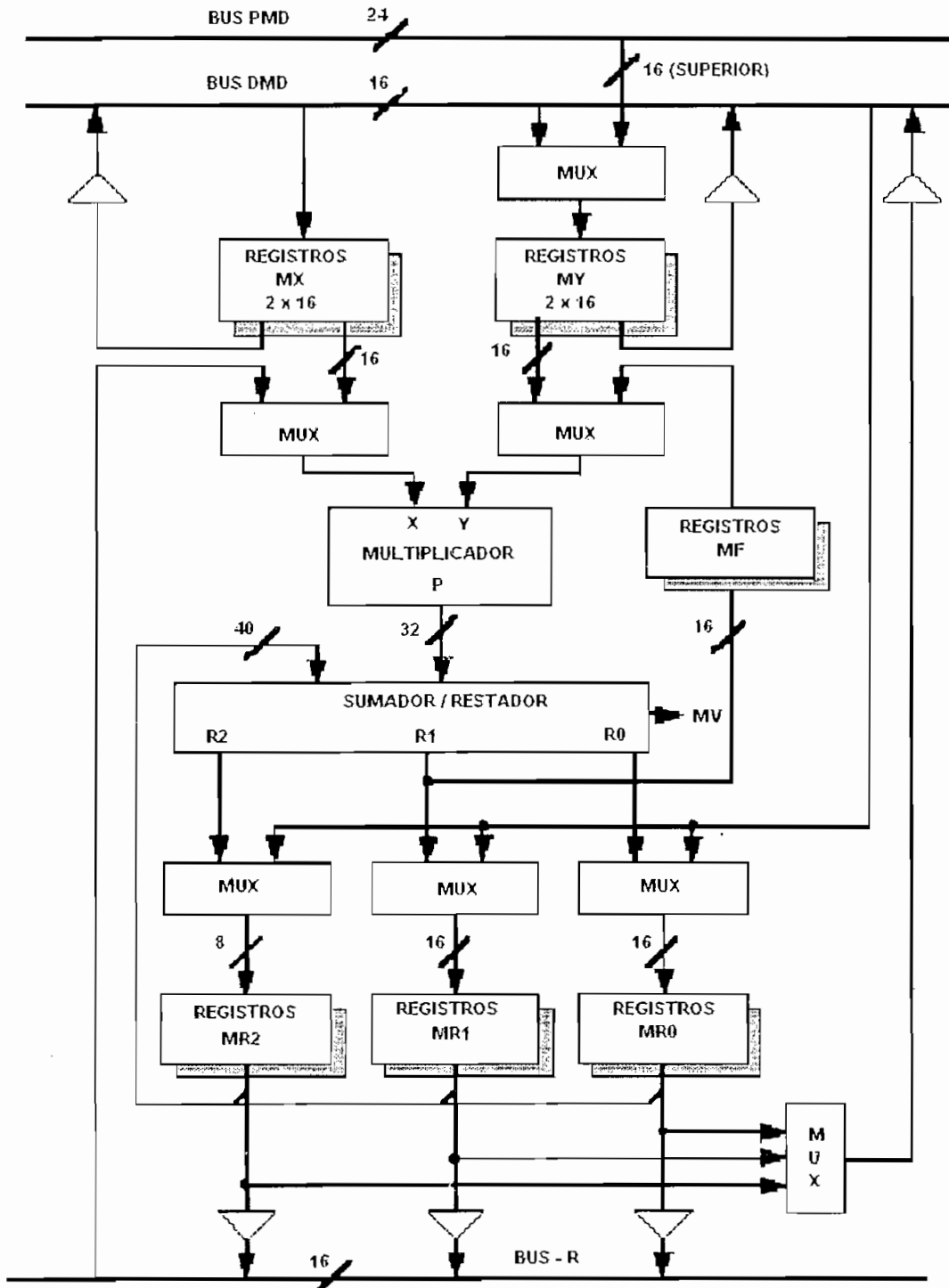


Figura 3.15 Diagrama de Bloques de el MAC. [1]

Los registros de entrada y salida de la MAC son similares a las de la ALU. El puerto de entrada (X) puede aceptar datos desde los registros (MX) o desde cualquier otro registro sobre el bus R. El bus R conecta los registros de salida de

todas las unidades computacionales, permitiendo que sean usados directamente como operandos de entrada. Los archivos del registro (MX) están formados de dos registros menores, (MX0 y MX1). Estos registros pueden ser leídos o escritos desde el bus DMD. El puerto de entrada (Y) puede aceptar datos desde los registros (MY) o desde el registro (MF). El registro (MY) está formado de dos registros menores: (MY0 y MY1), estos registros pueden ser escritos o leídos desde el bus DMD y tan solo escritos desde el bus PMD.

La salida del sumador / restador se dirige hacia el registro (MF) o hacia los registros (MR). El registro (MF) es un registro de realimentación el cual permite que los bits MSB del registro de resultados (bit 16 al 31) sean usados directamente como entrada del multiplicador (Y), en el ciclo siguiente. Los 40 bits del registro (MR) que se dirigen hacia el sumador / restador están divididos en tres secciones (MR2, MR1 y MR0). Cada uno de estos registros se los puede cargar directamente desde el bus DMD y su salida puede dirigirse hacia el bus DMD o el bus R.

Cualquiera de los registros asociados con la MAC pueden ser de lectura y escritura en un mismo ciclo de máquina. Estos registros son leídos al inicio de un ciclo y pueden ser escritos al final del mismo ciclo. Por lo tanto, cuando se lee un registro, se está leyendo los valores cargados al final de un ciclo anterior. Un nuevo valor escrito en un registro, no puede ser leído hasta el inicio del ciclo siguiente. Esto permite que un registro de entrada provea un operando a el MAC al inicio de un ciclo y pueda ser actualizado con el siguiente operando de la memoria al final del mismo. Esto además permite que un registro de resultados sea almacenado en memoria y sea actualizado con un nuevo resultado en un mismo ciclo de máquina.

El MAC contiene un banco duplicado de registros, solo un banco de registros puede ser activado a la vez. La selección del registro del banco primario o secundario es controlado por el bit 0 en el registro de modo de estado (MSTAT). Si el bit es 0, se selecciona el banco primario, si el bit es 1 se selecciona el banco secundario.

Las funciones que pueden desarrollarse en el MAC son las siguientes:

$X * Y$	Multiplica los operandos X e Y
$MR + X * Y$	Multiplica los operandos X e Y, y añade el resultado al registro MR
$MR - X * Y$	Multiplica los operandos X e Y, y resta el resultado del registro MR
0	Setea el resultado del registro MR a cero

Las fuentes posibles de las entradas y salidas del MAC son [1]:

FUENTE PARA EL PUERTO DE ENTRADA (X)	FUENTE PARA EL PUERTO DE ENTRADA (Y)	DESTINO PARA EL PUERTO DE SALIDA (R)
MX0, MX1	MY0, MY1	MR (MR2, MR1, MR0)
AR	MF	MF
MR0, MR1, MR2		
SR0, SR1		

Tabla 3.7 Entradas y Salidas del Multiplicador-Acumulador

### 3.2.5 UNIDAD DE DESPLAZAMIENTO

La unidad de desplazamiento también llamada *shifter*, proporciona una completa gama de funciones de desplazamiento para entradas de 16 bits, con salidas de hasta 32 bits. Estas funciones incluyen desplazamientos aritméticos, desplazamientos lógicos, normalización y denormalización. Estas funciones básicas pueden ser combinadas eficientemente para implementar un control de formatos numéricos, que incluyen representaciones en punto flotante. La figura 3.16 muestra el diagrama de bloques del *shifter*.

La sección del *shifter* puede ser dividida dentro de los siguientes componentes: el arreglo del desplazador, la lógica OR/PASS, el detector de exponente y la lógica del comparador de exponente.



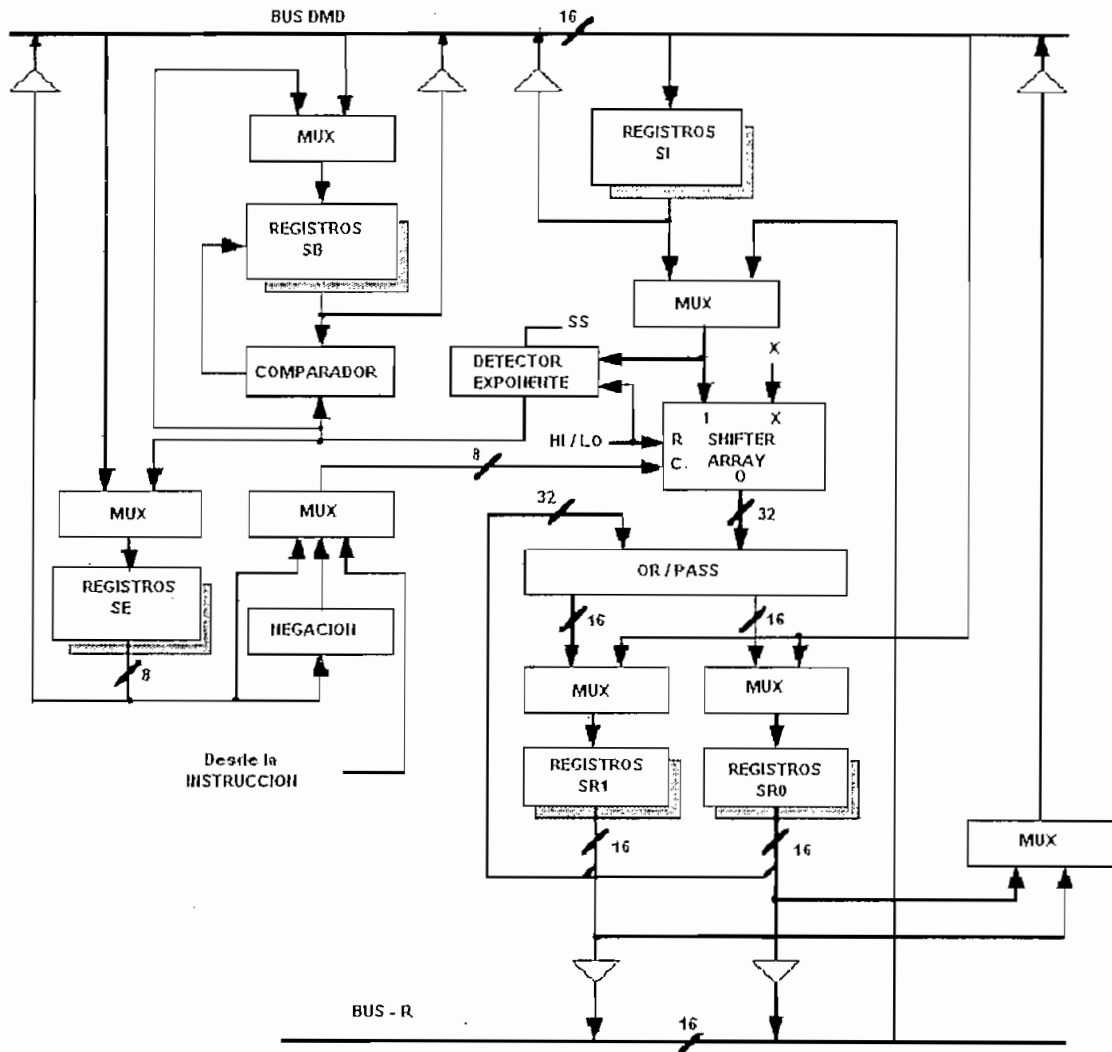


Figura 3.16 Diagrama de Bloques de la Unidad de Desplazamiento [1]

El arreglo del desplazador está formado por un desplazador de barril de 16x32. Este acepta una entrada de 16 bits y la puede colocar en cualquier campo cuya salida es de 32 bits, en un solo ciclo de máquina. Esto da lugar a 49 posibles posiciones dentro del campo de 32 bits. La ubicación de las entradas de 16 bits está determinada por un código de control (C) y una señal de referencia (HI / LO).

El registro SI (*Shifter Input*, Entrada del desplazador) provee la entrada al arreglo del desplazador y al detector de exponente. El registro (SI) tiene una extensión de 16 bits y puede ser leído o escrito desde el bus DMD. El arreglo del

desplazador y el detector de exponente también pueden aceptar como entradas los registros (AR), (AS) o el (MR), por medio del bus R.

El registro SR (*Shifter Result*, Resultado del desplazador) con extensión de 32 bits puede ser dividido dentro de dos secciones de 16 bits, el (SR0) y el (SR1). Estos registros pueden ser almacenados desde el bus DMD o pueden ser salidas hacia el bus DMD o el bus R. El registro (SR) es realimentado hacia la lógica OR/PASS para permitir operaciones de desplazamiento con doble precisión.

El registro SE (*Shifter Exponent*, Desplazador de Exponentes) tiene una extensión de 8 bits y mantiene el exponente durante las operaciones de normalización y desnormalización. Este registro puede almacenar y leer los 8 bits menos significativos del bus DMD.

El registro SB (*Shifter Block*, Bloque de desplazamiento) es importante en operaciones de bloques de punto flotante donde se mantiene el valor del exponente del bloque. El registro (SB) tiene 5 bits y mantiene el valor del exponente de bloque más reciente. Este registro se puede almacenar y leer los 5 bits menos significativos del bus DMD en complemento de dos.

La sección de desplazamiento contiene un banco duplicado de registros, solo uno de ellos puede ser activado a la vez. El banco adicional puede ser activado para combinaciones extremadamente rápidas. La selección del registro del banco primario o secundario es controlado por el bit 0 en el registro de modo de estado (MSTAT). Si el bit es 0, se selecciona el banco primario, si el bit es 1 se selecciona el banco secundario.

El desplazamiento de la entrada está determinado por el código de control (C) y por la referencia de la señal HI / LO. El código de control es un valor de 8 bits con signo, el cual indica la dirección y el número de lugares que va a ser desplazado en la entrada. Los códigos positivos indican un desplazamiento a la izquierda y los códigos negativos indican el desplazamiento a la derecha. El código de control se lo puede obtener de tres fuentes: el contenido de el registro

(SE), el contenido negado del registro (SE) o un valor inmediato desde la instrucción.

La señal HI/LO determina el punto de referencia para el desplazamiento. En el estado HI, todos los desplazamientos (la mitad superior del registro de salida SR) tienen como referencia a (SR1), y con el estado LO, todos los desplazamientos (la mitad inferior del registro de salida SR) tienen como referencia a (SR0). Las señales de referencia HI/LO son seleccionadas cada vez que el *shifter* es usado.

El desplazador completa todos los bits hacia la derecha del valor de entrada con ceros en el registro de salida, y todos los bits hacia la izquierda son llenados con un bit de extensión (X). El bit de extensión puede ser alimentado por tres fuentes posibles dependiendo de la instrucción que sea ejecutada. Estas tres fuentes son: el bit más significativo de la entrada, el bit (AC) del registro de estado aritmético (ASTAT) ó el cero.

La lógica OR/PASS permite que las secciones desplazadas de un número de precisión múltiple sean combinadas en una sola cantidad. Cuando PASS es seleccionado, la salida del arreglo del desplazador se pasa directamente al registro (SR) sin ser modificado. Cuando OR es seleccionado, el arreglo del desplazador se suma lógicamente con los contenidos actuales de el registro (SR).

En la tabla 3.8 se muestra la salida del arreglo del desplazador como función del código de control (C) y de la señal HI/LO.

El conjunto de caracteres ABCDEFGHIJKLMNPR representa el operando de entrada (16 bits) al arreglo del desplazador y (X) corresponde al bit de extensión anteriormente mencionado. Por ejemplo un desplazamiento con un código de control igual a -3 y con referencia HI, provoca que la entrada se desplace 3 lugares a la derecha (lugares que han sido llenados con el bit de extensión (X) ) tomando como referencia el registro SR1.

CÓDIGO DE CONTROL		SALIDA DEL ARREGLO DEL DESPLAZADOR
REFERENCIA HI	REFERENCIA LO	
+16 a +127	+32 a +127	00000000 00000000 00000000 00000000
+15	+31	R0000000 00000000 00000000 00000000
+14	+30	PR000000 00000000 00000000 00000000
+13	+29	NPR00000 00000000 00000000 00000000
+12	+28	MNPR0000 00000000 00000000 00000000
+11	+27	LMNPR000 00000000 00000000 00000000
+10	+26	KLMNPR00 00000000 00000000 00000000
+9	+25	JKLMNPRO 00000000 00000000 00000000
+8	+24	IJKLMNPR 00000000 00000000 00000000
+7	+23	HIJKLMNP R0000000 00000000 00000000
+6	+22	GHIJKLMN PR000000 00000000 00000000
+5	+21	FGHIJKLM NPR00000 00000000 00000000
+4	+20	EFGHIJKL MNPR0000 00000000 00000000
+3	+19	DEFGHIJK LMNPR000 00000000 00000000
+2	+18	CDEFGHIJ KLMNPR00 00000000 00000000
+1	+17	BCDEFGHI JKLMNPRO 00000000 00000000
0	+16	ABCDEF GH IJKLMNPR 00000000 00000000
-1	+15	XABCDEFG HIJKLMNP R0000000 00000000
-2	+14	XXABCDEFG GHIJKLMN PR000000 00000000
-3	+13	XXXABCDE FGHIJKLM NPR00000 00000000
-4	+12	XXXXABCD EFGHIJKL MNPR0000 00000000
-5	+11	XXXXXABC DEF GHIJK LMNPR000 00000000
-6	+10	XXXXXXAB CDEFGHIJ KLMNPR00 00000000
-7	+9	XXXXXXXA BCDEFGHI JKLMNPRO 00000000
-8	+8	XXXXXXXX ABCDEFGH IJKLMNPR 00000000
-9	+7	XXXXXXXX XABCDEFG HIJKLMNP R0000000
-10	+6	XXXXXXXX XXABCDEFG GHIJKLMN PR000000
-11	+5	XXXXXXXX XXABCDE FGHIJKLM NPR00000
-12	+4	XXXXXXXX XXXXABCDEF GHIJKL MNPR0000
-13	+3	XXXXXXXX XXXXABC DEF GHIJK LMNPR000
-14	+2	XXXXXXXX XXXXXAB CDEFGHIJ KLMNPR00
-15	+1	XXXXXXXX XXXXXXA BCDEFGHI JKLMNPRO
-16	0	XXXXXXXX XXXXXXX ABCDEFGH IJKLMNPR
-17	-1	XXXXXXXX XXXXXXX XABCDEFG HIJKLMNP
-18	-2	XXXXXXXX XXXXXXX XXABCDEFG GHIJKLMN
-19	-3	XXXXXXXX XXXXXXX XXXABCDE FGHIJKLM
-20	-4	XXXXXXXX XXXXXXX XXXXABCDEF GHIJKL
-21	-5	XXXXXXXX XXXXXXX XXXXABC DEF GHIJK
-22	-6	XXXXXXXX XXXXXXX XXXXXAB CDEFGHIJ
-23	-7	XXXXXXXX XXXXXXX XXXXXXA BCDEFGHI
-24	-8	XXXXXXXX XXXXXXX XXXXXXX ABCDEFGH
-25	-9	XXXXXXXX XXXXXXX XXXXXXX XABCDEFG
-26	-10	XXXXXXXX XXXXXXX XXXXXXX XXABCDE
-27	-11	XXXXXXXX XXXXXXX XXXXXXX XXXABCDE
-28	-12	XXXXXXXX XXXXXXX XXXXXXX XXXXABCDEF
-29	-13	XXXXXXXX XXXXXXX XXXXXXX XXXXABC
-30	-14	XXXXXXXX XXXXXXX XXXXXXX XXXXXXAB
-31	-15	XXXXXXXX XXXXXXX XXXXXXX XXXXXXA
-32 a -128	-16 a -128	XXXXXXXX XXXXXXX XXXXXXX XXXXXXX

Tabla 3.8 Salida del arreglo del desplazador como función del código de control (C) y de la señal HI/LO. [1]

Las fuentes de entrada y salida de la sección de desplazamiento son las siguientes [1]:

FUENTE DE ENTRADA PARA EL DESPLAZADOR	FUENTE DE SALIDA DEL DESPLAZADOR
SI	SR (SR0, SR1)
AR	
MR0, MR1, MR2	
SR0, SR1	

**Tabla 3.9** Fuentes de entrada y salida de la sección de desplazamiento [1]

Los registros MR0, MR1 y MR2 pertenecen a los resultados del multiplicador / acumulador; SR0 y SR1 son registros de resultado de la unidad de desplazamiento.

### 3.2.6 GENERADORES DE DIRECCIONES DE DATOS

El ADSP-2181 contiene dos generadores de direcciones de datos (DAGs) totalmente independientes, estas unidades permiten controlar el movimiento de los datos en el procesador de modo que la memoria de datos y de programas puedan ser accedidas simultáneamente. El DAG1 genera únicamente direcciones de memoria de datos, en tanto que el DAG2 puede generar ambas direcciones de datos y programas.

La figura 3.17 indica el diagrama de bloques del generador de direcciones de datos. Existen tres archivos de registros en un DAG: el archivo de registros de modificación (M), el archivo de registros de indexados (I), y el archivo de registros de longitud (L). Cada uno de estos archivos contienen cuatro registros de 14 bits los cuales pueden ser leídos y escritos por medio del bus DMD.

La localidad de memoria apuntada por un registro I puede ser accedida por la instrucción DM(I,M), en la cuál luego de la operación de lectura (o escritura) de la localidad de memoria, el contenido del registro M involucrado en la operación se añade a la dirección almacenada en I.

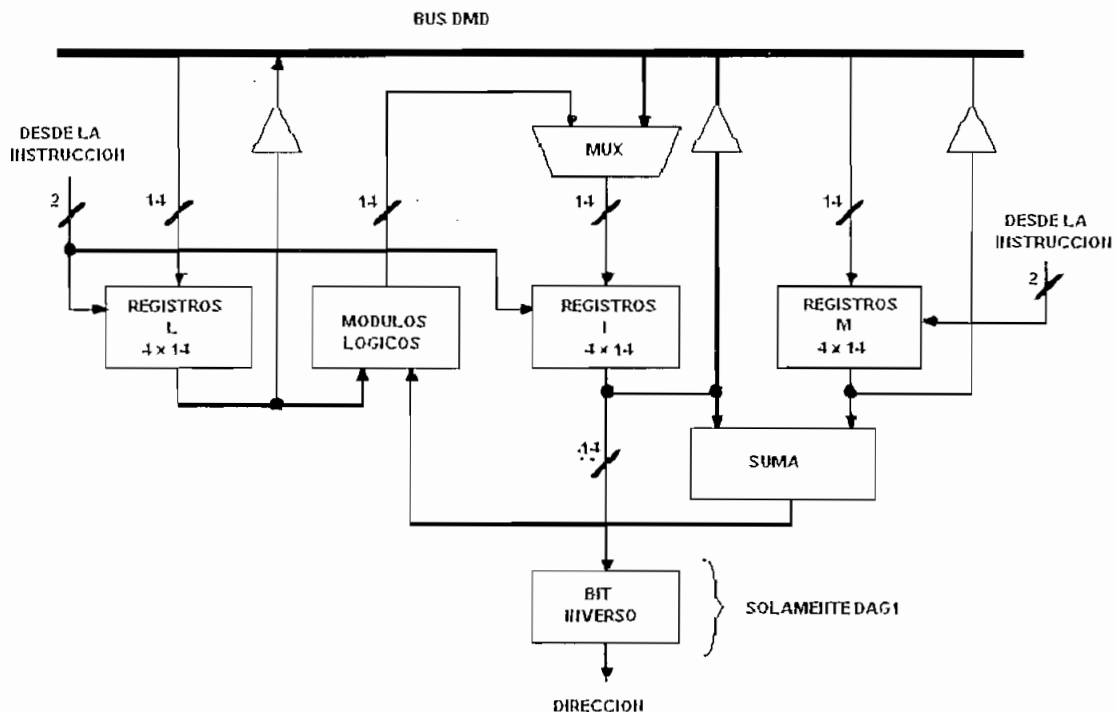


Figura 3.17 Diagrama de Bloques de un Generador de Direcciones de Datos [1]

<b>DAG 1</b>	I0, I1, I2, I3 M0, M1, M2, M3 L0, L1, L2, L3	Tipo de registros I, M y L en el DAG1 que se utilizan para acceder a la memoria de datos (DM)
<b>DAG 2</b>	I4, I5, I6, I7 M4, M5, M6, M7 L4, L5, L6, L7	Tipo de registros I, M y L en el DAG2 que se utilizan para acceder tanto a la memoria de datos (DM) como a la memoria de programas (PM).

Tabla 3.10 Registros I, M y L de los DAGs [1]

Por ejemplo, la instrucción:

$$MR1 = DM (I2, M3)$$

Almacena en el registro MR1 el contenido de la localidad de memoria apuntada por I2 y luego añade el contenido del registro M3 a la dirección almacenada en I2.

Si  $M3 = 1$ , entonces  $I2$  apuntará la siguiente localidad de memoria, o si  $M3 = -1$ ,  $I2$  apuntará la localidad de memoria anterior.

La elección de los registros  $I$  y  $M$  es independiente dentro de cada DAG; es decir, cualquier registro del conjunto  $I0-I3$  puede ser modificado por cualquier registro del conjunto  $M0-M3$ , pero no puede ser modificado por los registros del DAG2 ( $M4-M7$ ), esto se ilustra en la tabla 3.10. Los registros  $L$  son utilizados para determinar el tipo de direccionamiento que se utiliza con los registros  $I$ . Para direccionamiento lineal, la lógica de módulo (Figura 3.17) se deshabilita seteando el correspondiente registro  $L$  a cero. Para el direccionamiento de *buffers* circulares (direccionamiento circular), los DAGs ejecutan modificaciones de dirección de módulo inicializando el registro  $L$  con la longitud del *buffer*.

### 3.2.6 SECUENCIADOR DE PROGRAMA

El secuenciador de programa genera las direcciones de las instrucciones, ejecuta saltos condicionales e incondicionales, llamadas y retorno de subrutinas, procesamiento automático de interrupciones y otros mecanismos de control de flujo del programa.

El secuenciador de programa es manejado por el registro de instrucciones, el cual almacena la instrucción que esta siendo ejecutada. Mientras el procesador está ejecutando una instrucción, el secuenciador de programa realiza una traída previa (*pre-fetch*) de la siguiente instrucción de programa. La lógica de selección de la siguiente dirección en el secuenciador genera una dirección de memoria de programa de una de las siguientes fuentes:

- La salida del Sumador del Contador de Programa
- La pila del Contador de Programa (*PC Stack*)
- El registro de instrucciones

- El controlador de interrupciones.

El circuito de selección de la dirección siguiente escoge una de estas fuentes, basado en las entradas de: el registro de instrucciones, la lógica de condición, el comparador de lazos y el controlador de interrupciones. La dirección escogida para la siguiente instrucción es llevada entonces hacia el bus PMA. La figura 3.18 ilustra el diagrama de bloques del secuenciador de programa.

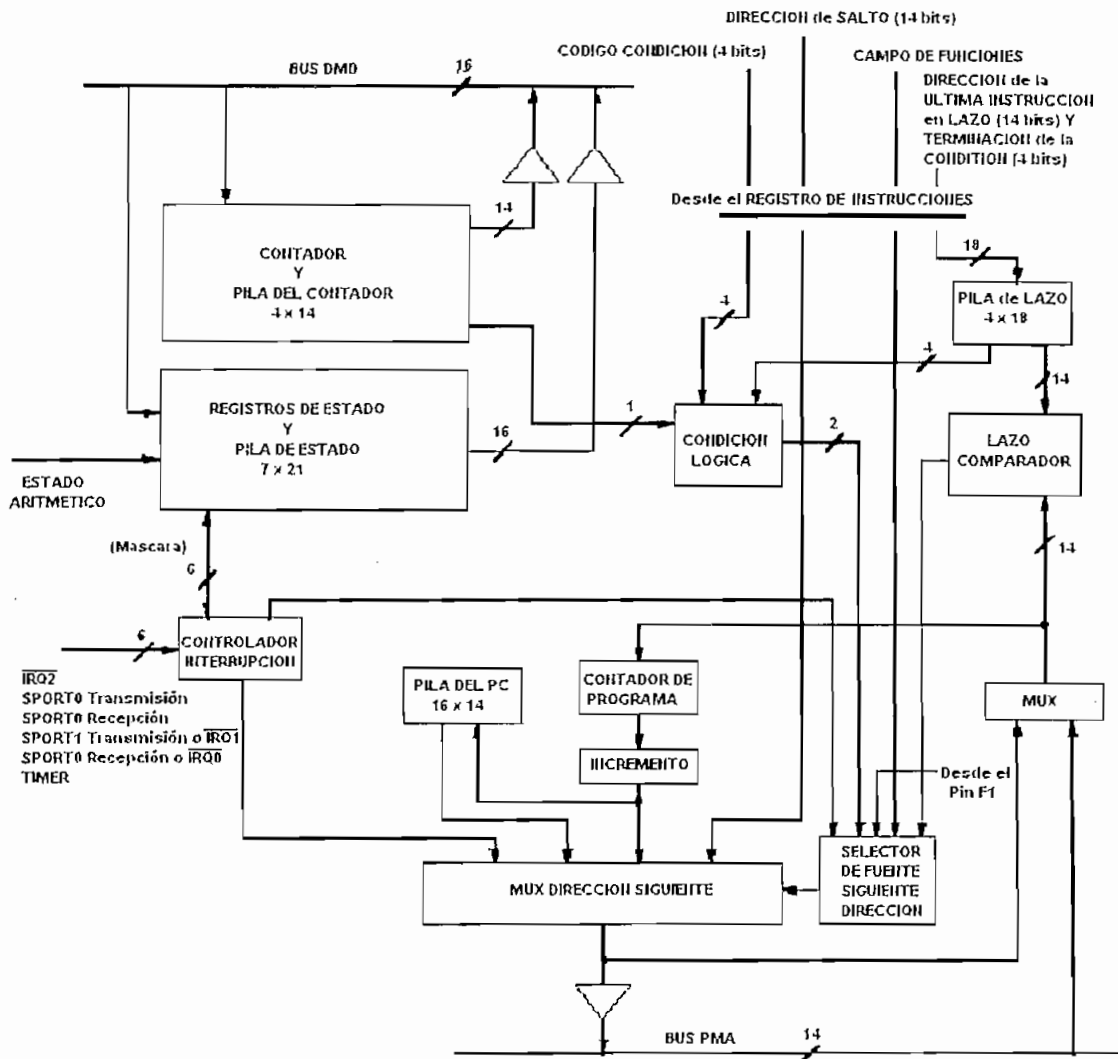


Figura 3.18 Secuenciador de Programas ADSP-2181 [1]

La salida del Sumador del Contador de Programa se selecciona como la fuente para la siguiente dirección si el flujo de programa es secuencial. Igual situación



ocurre si no se realiza un salto condicional o si se termina un lazo del tipo DO UNTIL.

En estos casos la salida del sumador del Contador de Programa se maneja hacia el bus PMA y luego hacia el Contador de Programa al comenzar el siguiente ciclo.

La pila del Contador de Programa (*PC Stack*) se utiliza como la fuente para la siguiente dirección cuando se ejecuta un retorno de subrutina o un retorno de interrupción, o en los casos de retorno al inicio de un lazo *DO UNTIL*.

El registro de instrucciones proporciona la siguiente dirección cuando se lleva a cabo un salto indirecto. En este caso la dirección del salto (de 14 bits) está contenida en la palabra de instrucción.

El controlador de interrupciones entrega la siguiente dirección de memoria de programa cuando se atiende una interrupción. Al reconocer una instrucción válida el procesador salta a la localización correspondiente al vector de interrupciones de la interrupción solicitada. La sección del secuenciador de programa contiene seis registros de estado:

- Registro de Estado Aritmético (ASTAT)
- Registro de Estado de Pila (SSTAT)
- Registro de Estado de Modo (MSTAT)
- Registro de Control de Interrupciones (ICNTL)
- Registro de Enmascaramiento de Interrupciones (IMASK)
- Registro de Forzamiento y Eliminación de Interrupciones (IFC)

Los contenidos de los registros de estado ASTAT, MSTAT e IMASK son almacenados en la Pila de Estado cuando el procesador responde a una interrupción. Estos valores son leídos de la Pila de Estado luego de que se retorna de la subrutina de interrupción con la instrucción RTI. Los registros ICNTL, IMASK e IFC son utilizados para configurar las interrupciones. En el ANEXO 2 se tiene todas las características y el detallado de los registros de estado.

### 3.3 EL EZ-KIT LITE <sup>[2]</sup>

Uno de los motivos principales de la aceptación de los *chips* DSP en la actualidad, es la gran variedad de herramientas de desarrollo proporcionadas por los principales fabricantes de éstos (*Analog Devices, Texas Instruments, Motorola, etc.*).

El EZ-KIT Lite es una plataforma de desarrollo de bajo costo del ADSP-2181 de *Analog Devices*. Este módulo tiene una versión de *software* que es totalmente funcional para el desarrollo de operaciones de DSP básicos. Con este *software*, se puede crear sistemas de gran desempeño basados en el ADSP-2181 con características sofisticadas.

#### 3.3.1 DESCRIPCIÓN DEL *HARDWARE* EZ-KIT Lite

El *hardware* EZ-KIT Lite consiste de un circuito impreso cuyas dimensiones son 8.9 cm x 14 cm, ensambladas dentro de este circuito impreso se encuentran: un procesador digital de señales ADSP-2181, una memoria EPROM, un codificador / decodificador AD1847, varios circuitos de soporte y conectores.

La tarjeta es un sistema completo de procesamiento digital de señales diseñado para demostrar los beneficios del procesador de señales ADSP-2181. Ésta puede ser utilizada como plataforma para el desarrollo de nuevas aplicaciones para el ADSP-2181, el diagrama físico de la tarjeta se observa en la Figura 3.19.

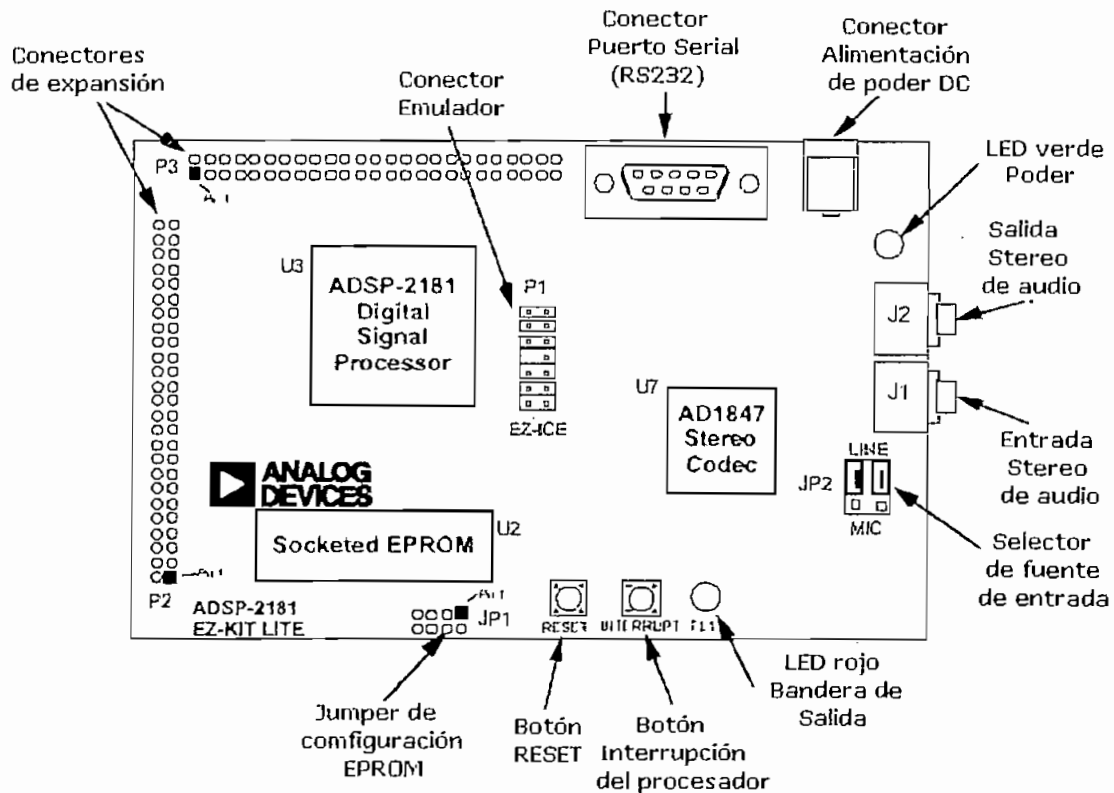


Figura 3.19 Diagrama físico de la tarjeta EZ-Lab. [2]

La tarjeta del EZ-KIT Lite es un ejemplo de una mínima implementación de un procesador ADSP-2181. La EPROM es conectada al procesador vía el puerto BDMA. Este interfaz usa solamente 8 de las 24 líneas de datos para los datos de *carry data* (D8 hasta D15). Ocho de los espacios de datos de líneas sobrantes (D16 hasta D23) son usadas para proveer bits de direcciones adicionales. Esto permite al ADSP-2181 direccionar hasta 32 Mbits (4Mbytes) de memoria.

La tarjeta posee un zócalo destinado para alojar diferentes tipos de memorias EPROM las cuales van desde 256 Kbits hasta 8 Mbits. JP1 provee una manera de ajustar las funciones de los pines 3 y 30 del zócalo como sea requerido para los diferentes tamaños de las EPROMs.

JP1 permite a la tarjeta EZ-KIT Lite ADSP-2181 ser configurada por uno de los seis diferentes tipos de EPROMs. Como la arquitectura de la tarjeta lo permite, ésta puede aceptar otro tipo de memoria como una 27C512 o 27C010, si una de

estas memorias es instalada en el zócalo U2, tendrá necesariamente que cambiar las conexiones en JP1.

Las conexiones son realizadas verticalmente entre bloques. El par de bloques entre cada número constituye la posición del *jumper* asociado con cada número. En la tabla 3.11 se muestran las conexiones y las memorias EPROMs que pueden ser utilizadas con el EZ-KIT Lite.

JP1				MEMORIA	TAMAÑO
1	2	3	4		
○ ○	○ ○	○ ○	□ ○	27C256	32 Kbyte
○ ○	○ ○	○ ○	□ ○	27C512 27C010	64 Kbyte 128Kbyte
○ ○	○ ○	○ ○	□ ○	27C020 27C040 27C080	256 Kbyte 512 Kbyte 1 Mbyte

Tabla 3.11 Configuración de JP1 y Tamaño de las EPROMs [2].

El *codec* AD1847 está encargado de la codificación y decodificación de las señales analógicas de entrada y salida, así como del envío de señales digitalizadas desde y hacia el ADSP-2181 a través de un bus serial. El AD1847 funciona con una sola fuente de alimentación +5VDC.

Utilizando cristales externos, las frecuencias de muestreo que soporta varían desde 5.5 kHz a 48 kHz. La figura 3.20. muestra un diagrama de bloques del AD1847.

El *codec* incluye dos conversores A/D (ADCs) y dos conversores D/A (DACs). Las entradas a los ADC pueden ser seleccionadas desde cuatro pares de entradas de señales analógicas las cuales son seleccionadas mediante *software*, la ganancia de cada canal estéreo puede ser programado antes del ingreso a los ADCs.

Las salidas de cada DAC las efectúa a través de un interfaz serial el cual es conectado al SPORT0 del ADSP-2181 en el EZ-LAB *board*, este puerto se configura en modo multicanal de 32 canales y datos de 16 bits.

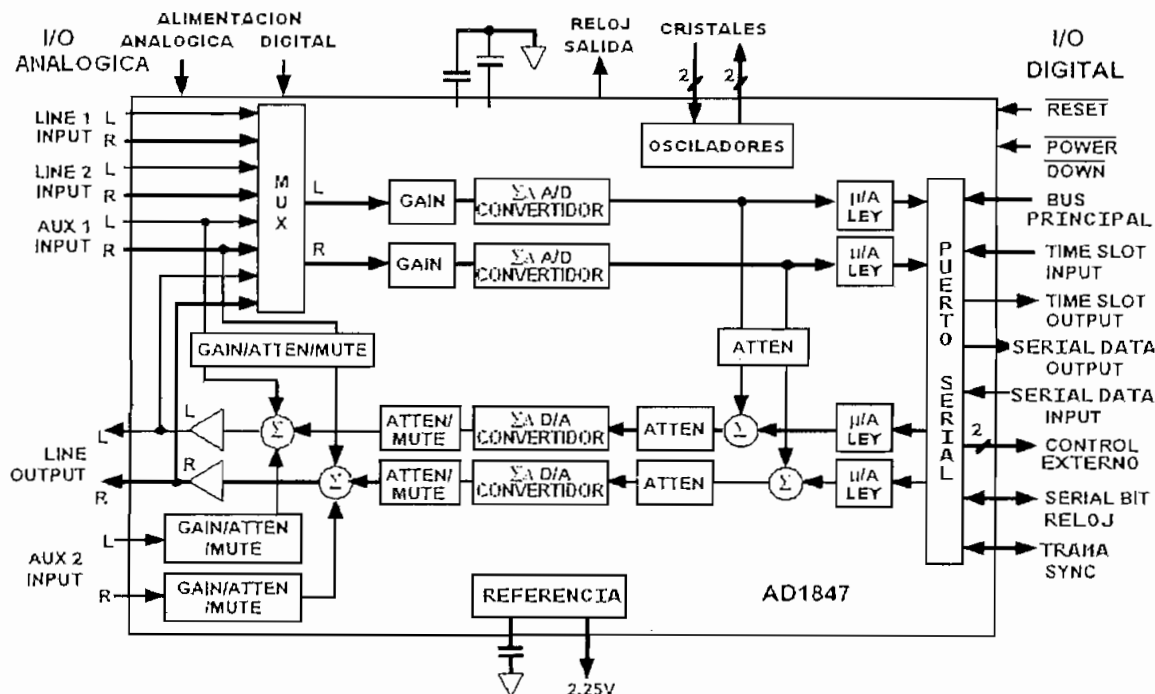


FIGURA 3.20. Diagrama de bloques del AD1847 [2]

El AD1847 también se configura para 32 canales (32 *slots* de tiempo por trama de sincronización, en el Registro de Información Miscelánea). Una señal de sincronización de trama inicia la transferencia de los 32 canales. El AD1847 proporciona las señales de reloj (SCLK) y de sincronización de trama para el ADSP-2181. En la Figura 3.21 se muestra como está configurado el interfaz serial entre el ADSP-2181 y el AD1847 en el EZ-Lab.

El *codec* puede aceptar cuatro tipos de formato de datos: PCM (*Pulse Code Modulated*, Códigos Pulso Modulados) lineal de 16 bits en complemento de dos, PCM lineal de 8 bits codificados de acuerdo al estándar *A-law* y datos de 8 bits codificados de acuerdo al estándar  $\mu$ -*law*.

Los ADCs incorporan un modulador de cuarto orden. Para prevenir el *aliasing* de la entrada analógica se requiere únicamente un filtro pasivo de un solo polo (circuito RC) ya que los ADCs tienen una tasa de sobremuestreo igual a 64 veces la frecuencia de muestreo que se esté utilizando y filtros de diezmado digital con una frecuencia de corte igual a  $0.40 \cdot f_m$ .

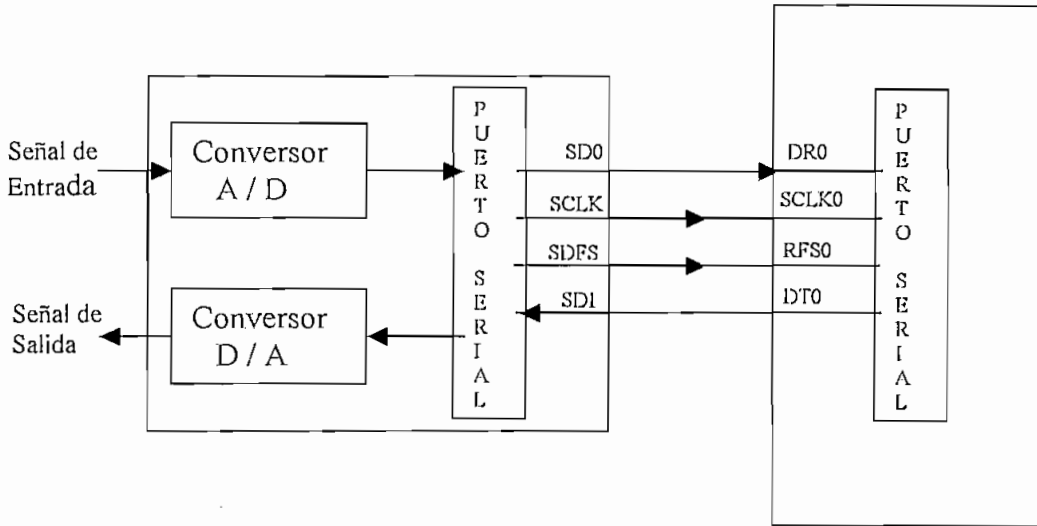


Figura 3.21. Interfaz serial entre el ADSP-2181 y el AD1847 [2]

Los DACs contienen un atenuador programable y un filtro de interpolación digital pasa bajo. El atenuador permite el control independiente de cada canal DAC desde 0 dB a  $-94.5$  dB en pasos de 1.5 dB. Las salidas de los DACs son filtradas analógicamente por una combinación de capacitores y filtros analógicos. Estos filtros remueven toda la componente de alta frecuencia por lo que no requiere de componentes externos adicionales.

Una salida en nivel de línea es disponible desde el *codec*. Para otro nivel de salida como por ejemplo audífonos o parlantes es necesario la implementación de una circuitería externa. Cada canal de salida puede ser independientemente silenciado.

El AD1847 opera con dos cristales externos, XTAL1 (opera a una frecuencia de 16.9344 MHz) y XTAL2 (opera a una frecuencia de 24.576 MHz). Estos dos cristales externos proporcionan frecuencias de muestreo de 5.5125 kHz, 6.6125

kHz, 8 kHz, 9.6 kHz, 11.025 kHz, 16 kHz, 18.9 kHz, 22.05 kHz, 27.42857 kHz, 32 kHz, 33.075 kHz, 37.8 kHz, 44.1 kHz y 48 kHz.

El puerto serial 1 (SPORT1) es utilizado para comunicarse con el computador a través de un interfaz RS-232, en el cual los pines *Flag In* y *Flag Out* del procesador llevan los datos de transmisión y recepción.

U1 es un dispositivo lógico que contiene seis inversores. Dos de estos inversores son usados para proporcionar los estados de *power-on*, *reset* y no-rebotes del accionamiento del botón *reset*. Otros dos inversores son usados para no-rebotes del accionamiento del botón de interrupción. Un quinto inversor es usado para manejar el LED rojo (D1), el sexto inversor no es usado.

El puerto IDMA sobre el DSP no es usado en la tarjeta EZ-KIT Lite, pero todas las señales de este puerto se encuentran disponibles en el conector P3.

### 3.3.2 ESPECIFICACIONES DE LA TARJETA EZ-Lab

<b>Procesador:</b>	ADSP-2181KS-133 Ejecuta instrucciones a una velocidad de 33 MHz.
<b>Interfaz Analógico:</b>	CODEC <i>estéreo</i> AD1847.
<b>Entradas Analógicas:</b>	Un par <i>estéreo</i> de 2V AC RMS acopladas a nivel de línea. Un par <i>estéreo</i> de 20mV AC RMS acoplada a entradas de un micrófono.
<b>Salidas Analógicas:</b>	Un par <i>estéreo</i> de 1V AC RMS acopladas a nivel de línea.
<b>Alimentación:</b>	8 a 10V DC 300 mA

**Ambientales:** 0 a 7° centígrados.  
10 a 90 % de humedad relativa (no condensación).

### 3.3.2.1 Conectores

J1 es un conector *jack estéreo* de 1/8 pulgadas (3.5 mm). Éste *jack* es usado para conducir las señales de audio hacia el nivel de línea o al micrófono dentro de la tarjeta.

La tarjeta acepta señales de entrada estéreo (canal izquierdo y canal derecho) de hasta 2V RMS acopladas a nivel de línea o señales estéreo de hasta 20 mV RMS de un micrófono. La selección de los niveles de la señal de entrada (de línea o micrófono) se realiza configurando los *jumbers* en el conector JP2, tal como se muestra en la Figura 3.22.



**Figura 3.22.** Selección de las señales de entrada. [2]

a) Señal en nivel de línea. b) Señal de micrófono.

J2 es un *jack estéreo* de 1/8 pulgadas (3.5 mm). Éste *jack* es usado para obtener señales de audio hacia el nivel de línea desde la tarjeta. J3 es un conector hembra D-Sub de 9 pines. Éste es usado para comunicaciones con un computador *host* usando señales de nivel RS-232 y protocolos seriales asincrónicos. J4 es un *jack* para un *plug* cilíndrico de 5.5 mm. Éste es usado para la alimentación de poder de la tarjeta. El pin central del *jack* es de 2 mm de diámetro y recubierto por el lado negativo de la fuente de poder.



JP1 es un conector *header* de ocho pines. Éste puede ser usado para configurar la tarjeta para memorias EPROMs de otro tamaño. JP2 es un conector *header* de seis pines. Éste es usado para configurar la entrada *jack* J1 para otro nivel de línea o micrófono de entrada.

Los *jumpers* pueden ser usados para conectar estos pines para una de las salidas del amplificador del micrófono o para la salida del filtro de entrada del nivel de línea.

P1 es un conector *header* de 14 pines usado para conectar a un ADSP-2181 el emulador EZ-ICE ®. El pin 7 puede ser removido para propósitos de seguridad. P2 y P3 son conectores *header* de 50 pines, estos conectores pueden ser usados para acceder a señales del ADSP-2181 para expansión o propósitos de prueba.

### 3.3.2.2 Switches

S1 es el *switch* del botón de *reset*. Accionando este botón causa un estado de *reset* al procesador ADSP-2181 y al CODEC AD1847 y permaneciendo hasta que sea soltado. Los *switches* de salida tienen la característica de no rebotes para prevenir múltiples transiciones esperadas en el rebote del contacto mecánico.

S2 es el *switch* del botón de interrupción. Pulsando este botón causa al procesador ADSP-2181 el recibimiento de una interrupción de entrada IRQE. Esto puede causar al procesador el ejecutar la interrupción IRQE manipulando el *software* si la interrupción es habilitada y el vector de interrupción IRQE esta colocada. El *switch* de salida es electrónicamente sin rebotes para prevenir múltiples interrupciones esperadas al rebote del contacto mecánico.

### 3.3.2.3 Indicadores

D1 es un diodo emisor de luz rojo el cual es controlado por la salida FL1 del procesador ADSP-2181. El *software* puede controlar el estado de este indicador por escritura de un registro interno.

D2 es un diodo emisor de luz verde el cual esta activo cuando la tarjeta se encuentra energizada.

### 3.3.2.4 Conectores de expansión

P2 y P3 son lugares para conectores *header* de 50 pines los cuales proveen acceso a señales de la ADSP-2181 para propósitos de expansión o de prueba.

Los números de los pines sobre estos conectores son arreglados como indica la figura 3.23.

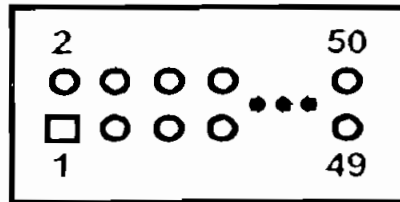


Figura 3.23. Arreglo de un conector tipo *header* de 50 pines

Las señales disponibles en estos pines son mostradas a continuación:

P2 NUMERO DE PIN	NOMBRE DE LA SEÑAL	P2 NUMERO DE PIN	NOMBRE DE LA SEÑAL
1	A0	2	A1
3	A2	4	A3
5	A4	6	A5
7	A6	8	A7
9	A8	10	A9
11	A10	12	A11
13	A12	14	A13
15	D0	16	D1
17	D2	18	D3
19	D4	20	D5
21	D6	22	D7
23	D8	24	D9
25	D10	26	D11
27	D12	28	D13
29	D14	30	D15

31	D16	32	D17
33	D18	34	D19
35	D20	36	D21
37	D22	38	D23
39	<u>WR</u>	40	<u>RD</u>
41	<u>IOMS</u>	42	<u>BMS</u>
43	<u>DMS</u>	44	<u>CMS</u>
45	<u>PMS</u>	46	<u>BR</u>
47	<u>BGH</u>	48	<u>BG</u>
49	VCC	50	GND
P3 NUMERO DE PIN	NOMBRE DE LA SEÑAL	P3 NUMERO DE PIN	NOMBRE DE LA SEÑAL
1	GND	2	IAD0
3	IAD1	4	IAD2
5	IAD3	6	IAD4
7	IAD5	8	IAD6
9	IAD7	10	IAD8
11	IAD9	12	IAD10
13	IAD11	14	IAD12
15	IAD13	16	IAD14
17	IAD15	18	GND
19	<u>IACK</u>	20	IAL
21	<u>IS</u>	22	<u>IWR</u>
23	<u>IRD</u>	24	GND
25	PF0	26	PF1
27	PF2	28	PF3
29	PF4	30	PF5
31	PF6	32	PF7
33	FL0	34	FL1
35	FL2	36	CLKOUT
37	<u>RESET</u>	38	<u>IRQL0</u>
39	IRQL1	40	<u>IRQ2</u>
41	<u>PWD</u>	42	PWDACK
43	<u>CODECDIS</u>	44	TXD0
45	TFS0	46	RFS0
47	RXD0	48	SCK0
49	VCC	50	GND

Tabla 3.12 Nombre de los pines de expansión. [2]

### 3.3.3 DESCRIPCIÓN DEL SOFTWARE DEL EZ-KIT Lite

El *software* de desarrollo de la familia ADSP-2100 es un *set* completo de herramientas que soporta el ADSP-2181.

En esta sección se procederá a explicar de una manera concreta y clara las funciones que generan cada una de las partes que conforman el sistema del *software* de desarrollo para el EZ-KIT Lite:

#### a) **EL ASSEMBLER**

Los programas que se desarrollan para el ADSP-2181 se escriben en un archivo de texto, se puede correr el *Assembler* el cual procesa todas las instrucciones en lenguaje ensamblador del archivo fuente a uno o varios archivos objeto. La extensión por defecto de los archivos de texto es ".dsp" (nombre de archivo.dsp). A partir de éste, el *assembler* crea un archivo con la extensión ".obj" (nombre de archivo.obj).

El *Assembler* se invoca desde MS-DOS con el siguiente comando:

```
asm21 programa -2181
```

En este ejemplo, el archivo de texto creado es de nombre programa.dsp . El *switch* -2181 determina el *assembler* para aceptar instrucciones específicas del ADSP-2181. el *assembler* crea archivo o archivos código objeto apropiados. El *assembler* crea un archivo objeto con la extensión .obj.

#### b) **EL LINKER**

El *Linker* crea un archivo ejecutable desde los módulos objeto creado por el *assembler*. El *Linker* se ejecuta desde MS-DOS como se muestra en el siguiente ejemplo:

```
ld21 programa -a adsp2181 -e prog
```

El archivo de descripción de la arquitectura **adsp2181.ach** es especificado con el parámetro -a y el archivo ejecutable es especificado como prog.exe con el parámetro -e. Existe un número de otros *switches* que son usados para crear una

tabla de símbolos, creando un archivo de mapeo y un archivo que especifica los archivos objeto indirectamente (especifica un archivo donde el archivo contiene una lista de todos los archivos .obj para ser enlazados).

El archivo **adsp2181.ach** es proporcionado con el EZ-KIT Lite, este archivo describe las características del *hardware* utilizado con el ADSP-2181 (en este caso el EZ-Lab). Éste es utilizado por el *Linker* y el Simulador para señalar inconsistencias entre el *software* y el uso del *hardware*.

### c) EL SIMULADOR

El simulador permite correr el código de programa en un ambiente de simulación para probar el *software* desarrollado sin necesidad de utilizar un sistema de *hardware*. El paso de simulación es usado para marcar con seguridad el *software* de trabajo antes de correr éste sobre el *hardware*.

A veces un problema puede surgir cuando se carga el *software* en el *hardware* y no se puede trabajar. Sin la verificación de la operación del programa en el simulador, el diagnóstico puede no ser seguro pues el defecto producido puede estar relacionado con el *hardware* o *software*.

Una vez verificada la operación del *software* en el simulador se procede a descargar el código al *hardware* y si sigue sin trabajar, esto es consecuencia de una falla propia del *hardware*. El simulador es invocado como se muestra:

```
sim2181 -a adsp2181 -e prog
```

Este comando inicia el simulador del ADSP-2181, simulando el programa prog.exe en el *hardware*. Si el simulador no encuentra alguno de estos archivos, se debe chequear el directorio.

```
sim2181 -a c:\adi_dsp\21xx\lib\adsp2181 -e prog
```

#### d) **EL PROM SPLITTER**

Una vez que se ha verificado que su *software* desarrollado trabaja sin problemas se puede formatear los archivos ejecutables de manera que puedan ser programados en una EPROM. La EPROM entonces puede ser insertada en el zócalo correspondiente de la tarjeta para correr el programa. El Prom *Splitter* puede ser invocado como se muestra en el siguiente ejemplo:

```
spl21 prog progrom -loader -2181
```

En este caso se toma el archivo ejecutable prog.exe y se crea un archivo PROM llamado progrom.bnm . El formato PROM por defecto es el formato de grabación *Motorola S*. Se puede especificar el formato de grabación *Intel Hex* con el parámetro `-i`.

Este archivo puede ser descargado al programador PROM para programar una EPROM la cual puede ser insertada en el zócalo de la tarjeta EZ-KIT Lite.

Cuando se alimenta o cuando se presiona el botón *reset* en la tarjeta, los contenidos de la EPROM son automáticamente cargados en la memoria del programa interno y memoria de datos de la ADSP-2181 y empieza la ejecución de la codificación.

#### e) **EL PROGRAMA HOST**

El Programa *Host* EZ-KIT Lite es un programa de aplicación basado en Windows que utiliza el modelo *Windows Graphical User Interface Conventions*. Éste es un programa de aplicación que usa la comunicación con la tarjeta EZ-LAB.

Con esto se puede correr en el EZ-KIT Lite programas demostrativos, cargar / descargar programas y contenidos de la memoria de datos, descargar programas DSP de usuario y ejecutar programas DSP de usuario

## f) EL PROGRAMA EZLD.COM

El programa ezld.com es un programa de dominio público que se encuentra disponible en la página web de Analog Devices (<http://www.analog.com>). Este programa desarrolla las mismas funciones que el Programa *Host* descrito anteriormente con la única diferencia que el programa ezld.com lo realiza bajo un ambiente de MS-DOS.

### 3.4 PROGRAMAS DESARROLLADOS PARA LOS FILTROS FIR

En el proceso de diseño e implementación de filtros digitales FIR se produjeron dos etapas de programación:

1. Programación<sup>1</sup> en *MATLAB* 5.3 previa a la implementación, es decir la determinación de los valores teóricos. Estos valores están respaldados por los cálculos obtenidos en *Mathcad* 4.0.

En la figura 3.24 se ilustra el diagrama de flujo del programa en *Matlab* mediante el cual se obtiene la respuesta al impulso y la respuesta en frecuencia, a más de las características de los filtros digitales FIR.

2. Programación<sup>2</sup> en lenguaje ensamblador del procesador ADSP-2181, en donde se determina todos los valores propuestos en el diseño de *Kaiser* (frecuencias normalizadas, frecuencia central, factor de *roll-off*, orden del filtro, funciones de *Bessel*, coeficientes del filtros, etc).

---

<sup>1</sup> El programa desarrollado en *Matlab* 5.3 únicamente sirve como referencia o elemento de comparación para los valores obtenidos en la DSP. En el **Anexo 3** y en el **Manual de usuario** se detalla la programación y las subrutinas utilizadas, así como los comandos usados en el diseño de filtros digitales FIR mediante la ventana de *Kaiser*.

<sup>2</sup> El ADSP-2181 es un procesador de alta precisión que permite determinadas operaciones matemáticas en su programación (sumas, restas, multiplicaciones). Es por esta razón que se utiliza aproximaciones numéricas y subrutinas preestablecidas para representar otro tipo de operaciones. En el **Manual de usuario** se adjunta el programa fuente (lenguajes ensamblador), a más de subrutinas y aproximaciones utilizadas en la realización del programa.

En la figura 3.24 se ilustra el diagrama de bloques utilizado en el programa de implementación de filtros FIR mediante la ventana de *Kaiser* en *Matlab* 5.3.

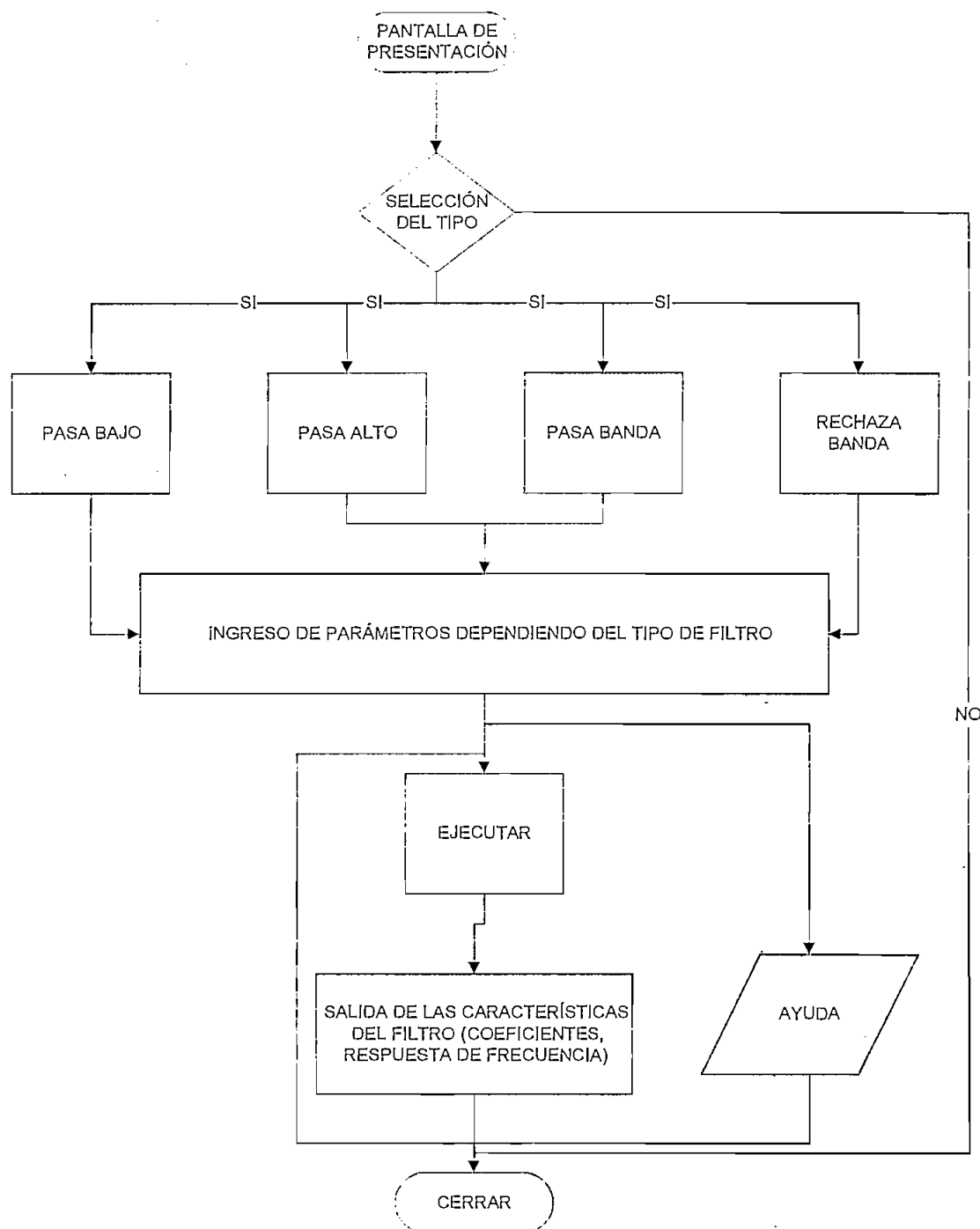


Figura 3.24 Diagrama de Flujo del programa en *Matlab* 5.3, mediante el cual se obtiene los valores teóricos que van a ser comparados con el DSP.



En la figura 3.25 se ilustra el diagrama de bloques utilizado en el programa de implementación de filtros FIR mediante la ventana de *Kaiser* en la ADSP-2181..

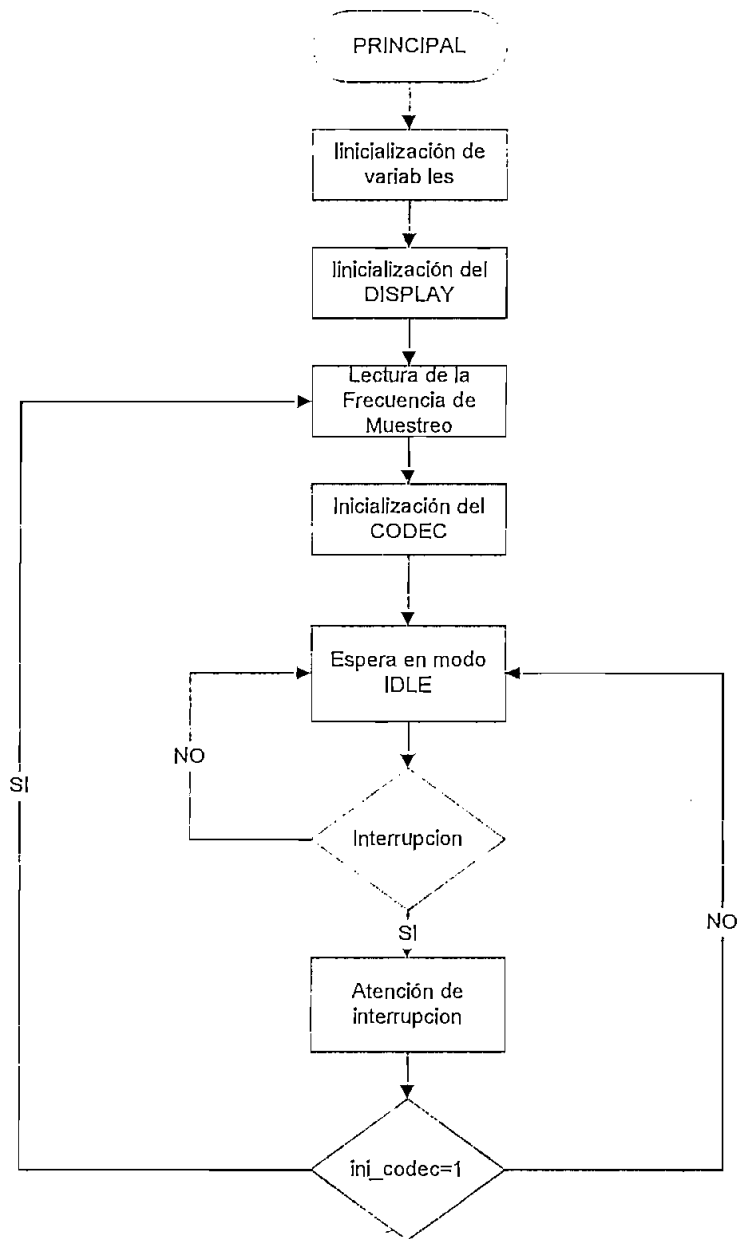


Figura 3.25 (a) Diagrama de Flujo. Programa principal implementado en la ADSP-2181.

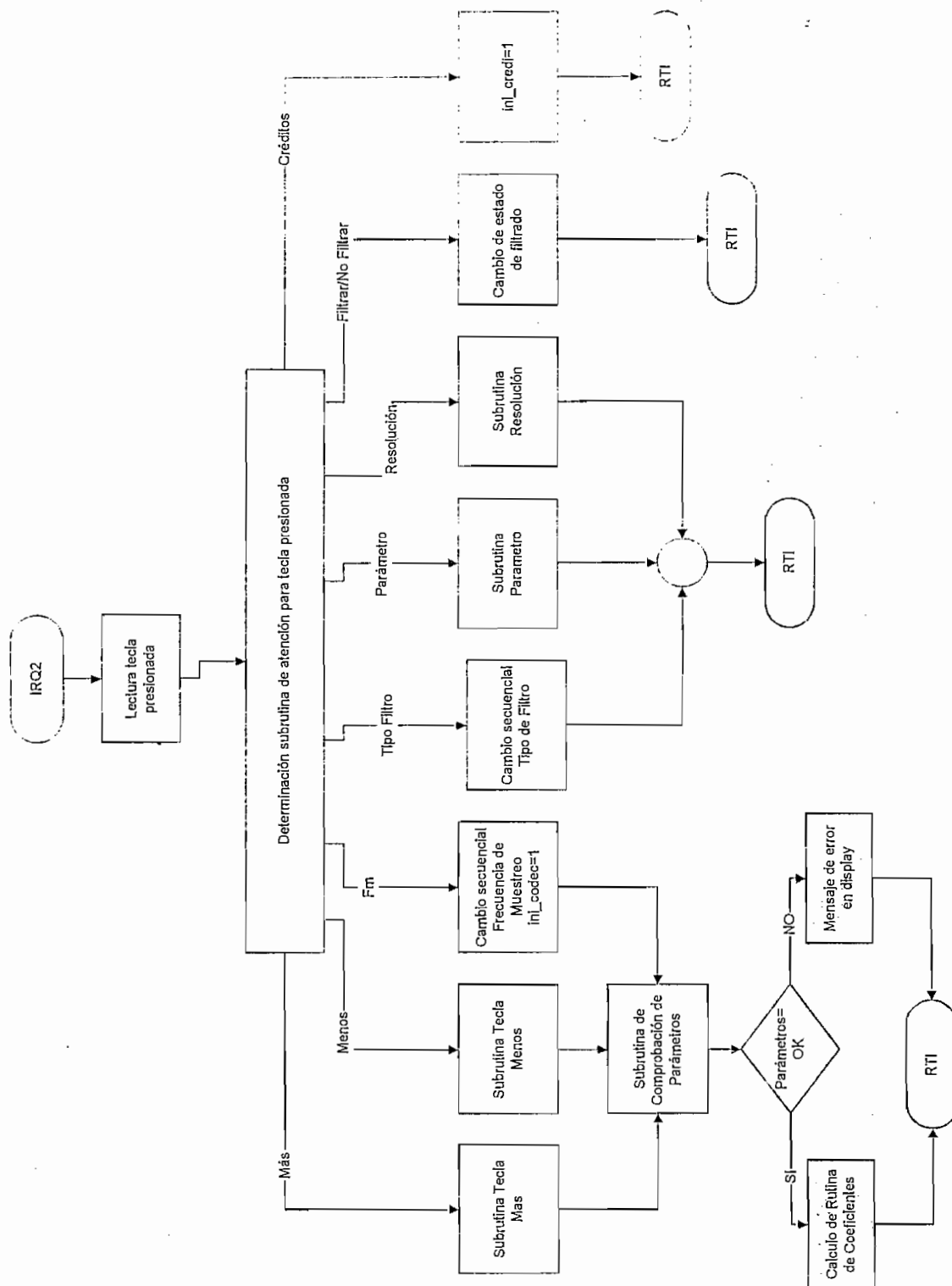


Figura 3.25 (b). Diagrama de Flujo. Programa de atención al teclado

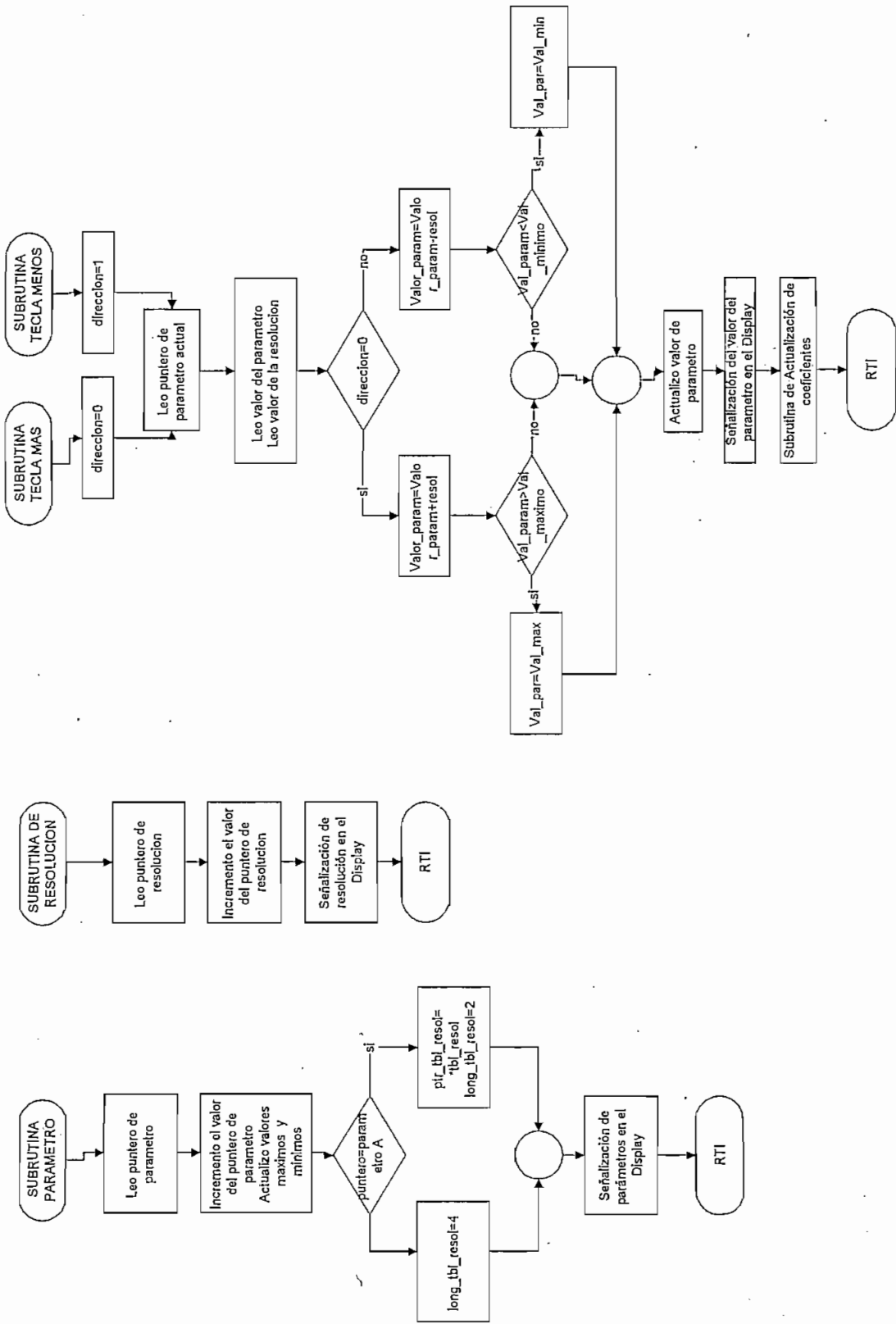


Figura 3.25 (c) Diagrama de Flujo. Subrutinas del teclado

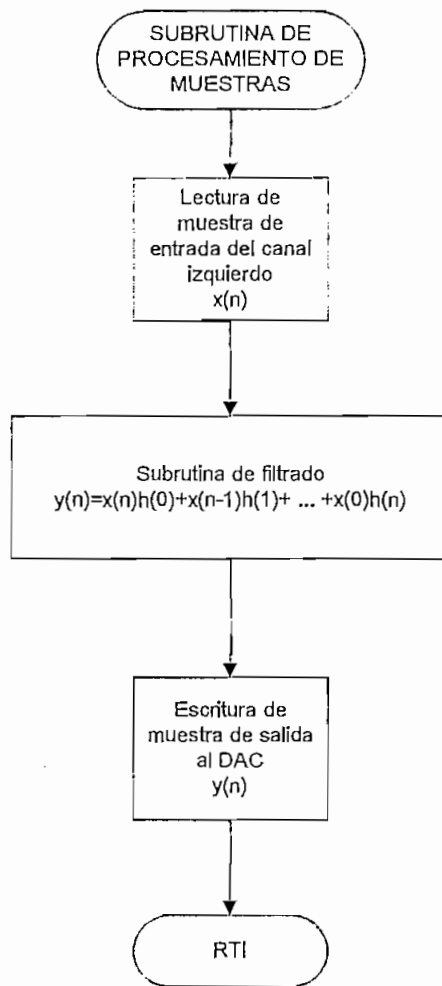


Figura 3.25(d) Diagrama de Flujo. Procesamiento de muestras

## CAPITULO 4

### RESULTADOS Y APLICACIONES

---

Con el conocimiento adquirido en los capítulos anteriores y luego de haber efectuado una serie de pruebas experimentales, a continuación se procederá a exponer los resultados obtenidos de cada una de las experiencias obtenidas en la implementación de los diferentes tipos de filtros no recursivos.

Se presentarán sistemas en los cuales existe la factibilidad de utilizar los filtros diseñados, debido a las ventajas que poseen éstos frente a los filtros analógicos expuestas en el Capítulo 1.

#### 4.1. EXPOSICIÓN DE RESULTADOS

En general, los filtros digitales implementados sobre Procesadores Digitales de Señales tienen muy alto rendimiento (debido a su velocidad de procesamiento, costo, etc). Sin embargo una de las mayores desventajas que presenta específicamente el ADSP-2181 en el diseño de filtros es precisamente sus limitaciones al trabajar con presentaciones de 16 bits en sus registros.

Este proyecto está orientado a realizar el filtrado de señales; en donde, los coeficientes del filtro fueron diseñados usando teoría de Procesamiento Digital de Señales (mediante la técnica de *Kaiser* aplicada en *MATHCAD 4.0*) y además paquetes de *software* adicional (mediante fórmulas preestablecidas en *MATLAB 5.3*). Estos valores teóricos permiten tener una idea de los datos que se deben obtener en la programación de la DSP.

Luego de realizar varios ejemplos experimentales de cada uno de los filtros digitales se los ha clasificado en ejemplos críticos y no críticos, esta clasificación es debido a anomalías que se obtiene cuando la atenuación de la banda de

supresión excede los 60 dB (caso crítico). Experimentalmente la atenuación recomendada para los diferentes tipos de filtros es menor o igual a 60 dB (caso no crítico).

#### 4.1.1 Resultados del Filtro Pasa Bajo

##### a) Caso no crítico

Diseñar un filtro pasa bajo que cumpla con los siguientes parámetros:

$f_p = 4000$ Hz	Frecuencia de paso
$f_s = 6000$ Hz	Frecuencia de supresión
$f_m = 16000$ Hz	Frecuencia de muestreo
$A = 40$ dB	Atenuación en la banda de supresión.

Coefficientes Sin Cuantificar (Obtenidos de *Mathcad* y *Matlab*) en formato decimal.

Coefficientes Cuantificados (Obtenidos del DSP) en formato 1.15

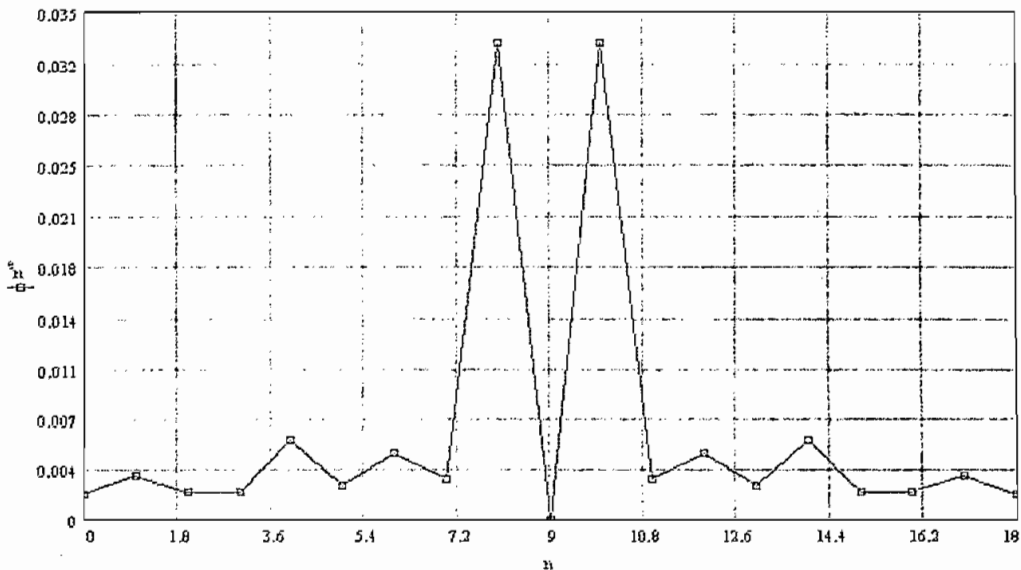
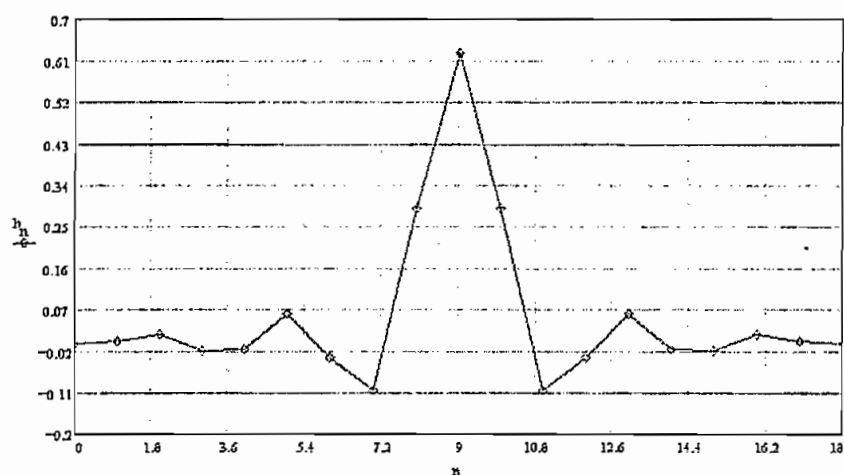


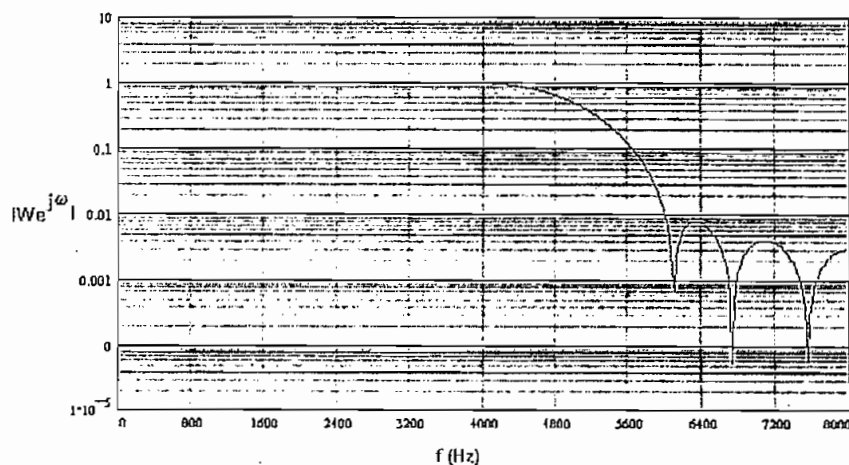
Figura 4.1 Error porcentual de los coeficientes cuantificados y sin cuantificar para el ejemplo (Filtro Pasa Bajo) no crítico.

## Resultados Teóricos

En la figura 4.1 se observa el error en porcentaje, de los datos teóricos (coeficientes sin cuantificar) y experimentales (coeficientes cuantificados), ésta gráfica nos permite apreciar la precisión de la DSP, ya que el error es menor que 0.035%. Los coeficientes sin cuantificar son graficados en la figura 4.2 (a), en donde se encuentra la respuesta impulsiva del filtro de orden 19. Nótese la simetría que existe en la respuesta impulsiva para un valor de  $n = 9$ .



(a)



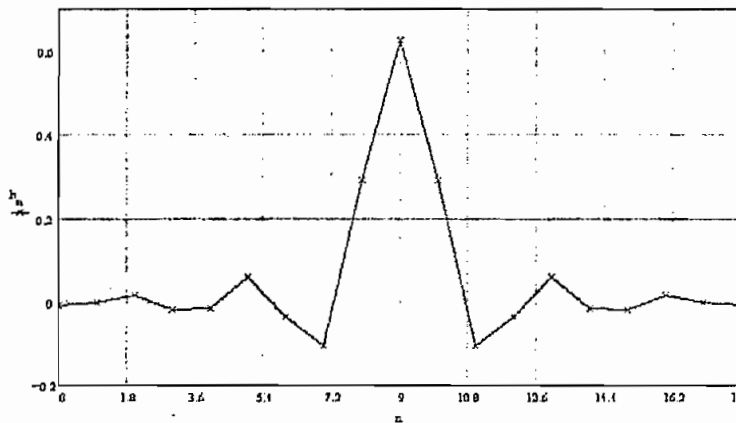
(b)

FIGURA 4.2. Características teóricas de un filtro pasa bajo de orden  $M = 19$  (caso no crítico). (a) Respuesta impulsiva teórica (sin cuantificar) diseñada en *Mathcad*. (b) Respuesta de frecuencia (logaritmo del módulo)

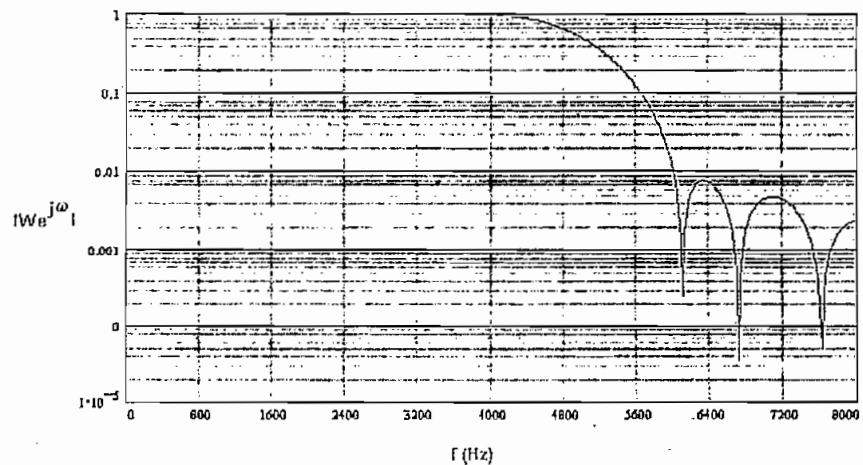
En la figura 4.2 (b) se ilustra la gráfica de  $20 \log_{10} |w(e^{j\omega})|$  (transformada de *Fourier*) vs frecuencia [Hz]. Nótese la atenuación de la banda de supresión por debajo de los 40 dB; es decir, con niveles de rizado de 0.01.

### Resultados Experimentales

Los coeficientes cuantificados son ilustrados en la figura 4.3 (a), en donde se puede observar la respuesta impulsiva del filtro de orden 19. Éstos coeficientes del filtro son almacenados en un *buffer* en la memoria de programas del procesador ADSP-2181.



(a)



(b)

**FIGURA 4.3.** Características experimentales de un filtro pasa bajo de orden  $M = 19$  (caso no crítico). (a) Respuesta impulsiva experimental (cuantificado) diseñada en ADSP-2181. (b) Respuesta en frecuencia.



En la figura 4.3 (b) se ilustra la respuesta de frecuencia experimental del filtro pasa bajo, se puede observar la frecuencia de paso en 4000 Hz, mientras que la frecuencia de supresión alrededor de los 6000 Hz. Además, la atenuación en la banda de supresión es de 40 dB.

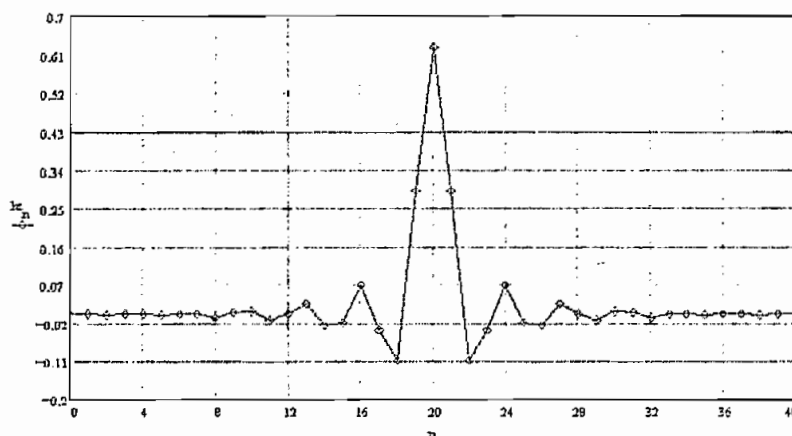
### b) Caso crítico

Diseñar un filtro pasa bajo que cumpla con los siguientes parámetros:

$f_p = 4000$ Hz	Frecuencia de paso
$f_s = 6000$ Hz	Frecuencia de supresión
$f_m = 16000$ Hz	Frecuencia de muestreo
$A = 80$ dB	Atenuación en la banda de supresión.

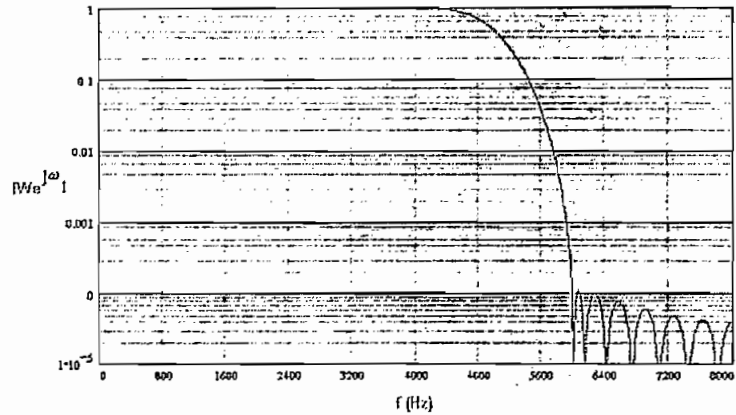
Si se observa los gráficos tanto experimental como teórico del caso crítico en el cual los parámetros del filtro son similares al ejemplo anterior, con excepción de la atenuación que se incrementa a 80 dB, se aprecia que el orden del filtro aumenta de 19 a 41, localizando el eje de simetría en  $n = 20$ .

### Resultados Teóricos



(a)

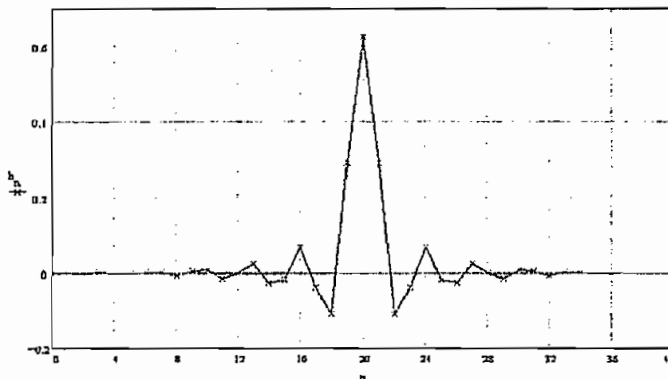
Al comparar los gráficos de la respuesta de frecuencia del filtro pasa bajo teórico (figura 4.4 b) con el experimental (figura 4.5 b) se observa que la banda de supresión llega a un valor máximo de 60 dB, es decir no cumple los 80 dB propuestos, debido a la cuantificación de truncamiento y redondeo.



(b)

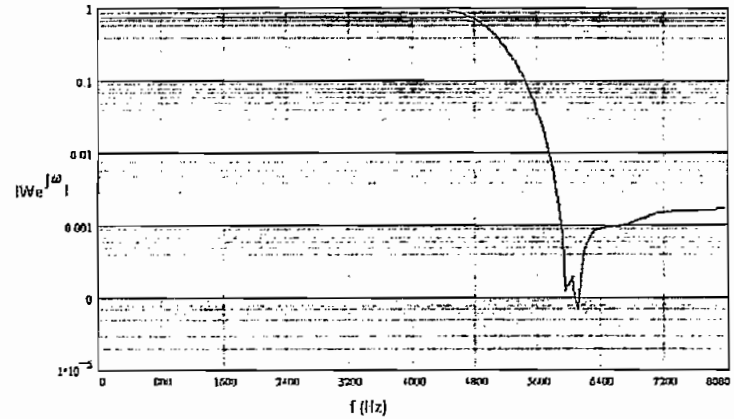
FIGURA 4.4 Características teóricas de un filtro pasa bajo de orden  $M = 41$  y atenuación de 80 dB (caso crítico). (a) Respuesta impulsiva teórica (sin cuantificar) diseñada en *Mathcad*. (b) Respuesta de frecuencia.

### Resultados Experimentales



(a)

Para los demás filtros fundamentales (pasa alto, pasa banda y rechaza banda) se analizará únicamente ejemplos no críticos debido a que las características y errores que se obtienen para el caso crítico son similares a los expuestos en el filtro pasa bajo.



(b)

FIGURA 4.5. Características experimentales de un filtro pasa bajo de orden  $M = 41$  y atenuación 80dB (caso crítico). (a) Respuesta impulsiva experimental (cuantificado) diseñada en ADSP-2181. (b) Respuesta de frecuencia.

#### 4.1.2 Resultados del Filtro Pasa Alto

Diseñar un filtro pasa alto que cumpla con los siguientes parámetros:

$f_s = 2000$ Hz	Frecuencia de supresión
$f_p = 2300$ Hz	Frecuencia de paso
$f_m = 8000$ Hz	Frecuencia de muestreo
$A = 50$ dB	Atenuación en la banda de supresión.

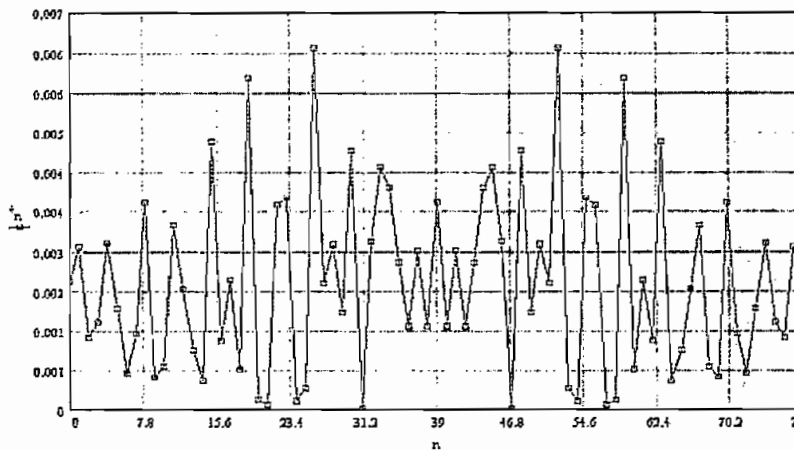
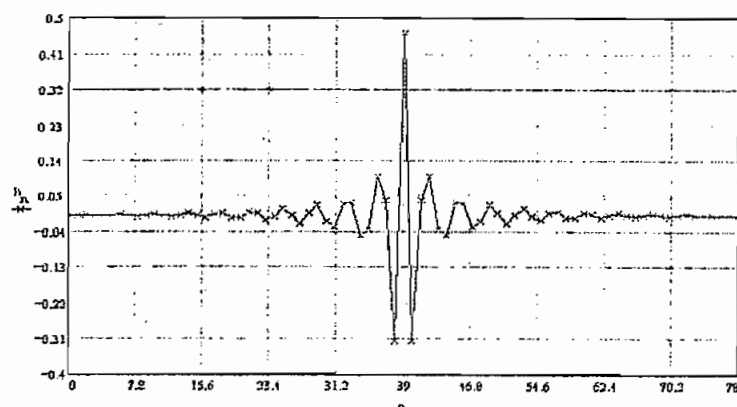


Figura 4.6 Error porcentual de los coeficientes cuantificados y sin cuantificar para el ejemplo (Filtro Pasa Alto)

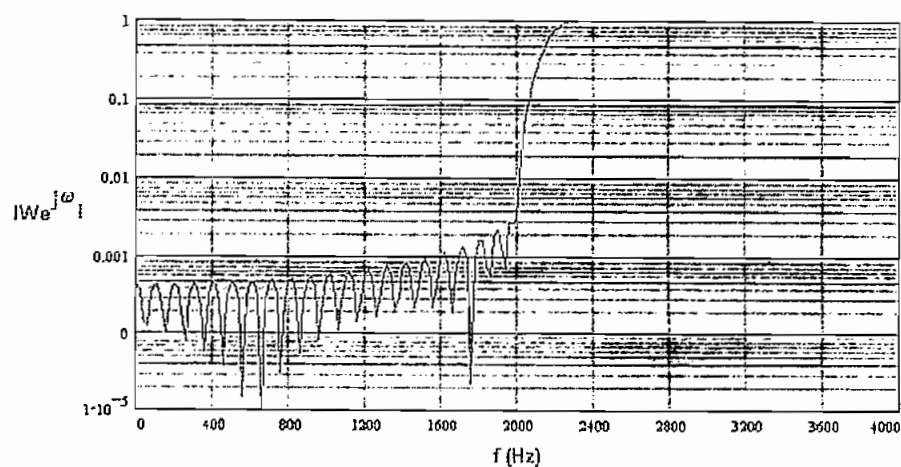
## Resultados Teóricos

El error porcentual de la figura 4.6 indica la diferencia entre los coeficientes cuantificados y sin cuantificar, con un error menor que el 0.007%. En la figura 4.7 (a), se encuentra la respuesta impulsiva del filtro de orden 79. Nótese la simetría que existe en la respuesta impulsiva, para un valor de  $n = 39$ , éste eje permitirá comparar los lóbulos laterales izquierdo y derecho.

En la figura 4.7 (b) se ilustra la gráfica de  $20 \log_{10} |w(e^{j\omega})|$  vs frecuencia [Hz]. Nótese la atenuación de la banda de supresión por debajo de los 50 dB.



(a)

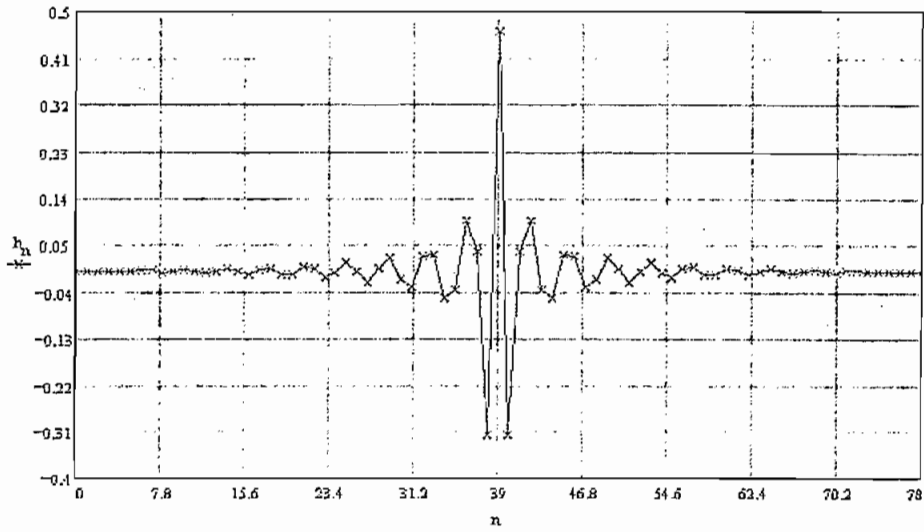


(b)

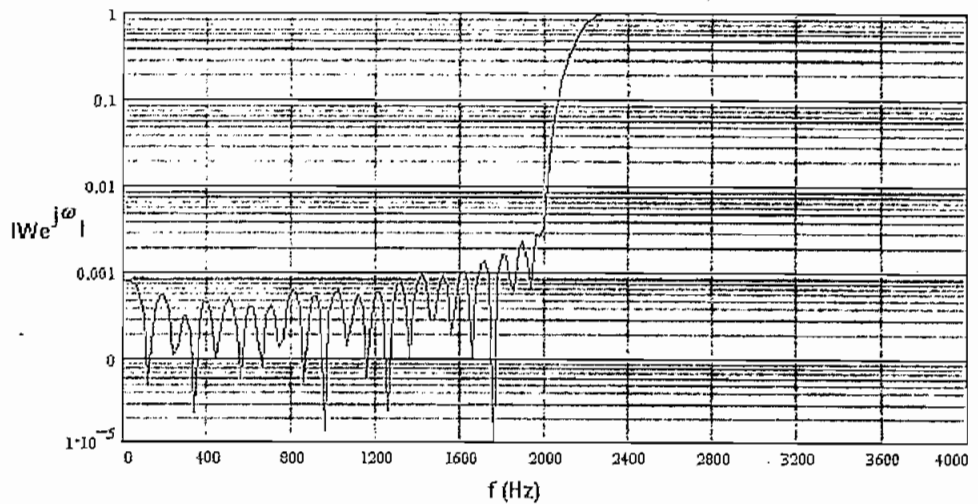
FIGURA 4.7. Características teóricas de un filtro pasa alto de orden  $M = 79$ .

## Resultados Experimentales

Los datos experimentales son ilustrados en la figura 4.8 (a), en donde se puede observar la respuesta impulsiva del filtro de orden 79. En la figura 4.8 (b) se ilustra la respuesta de frecuencia del filtro pasa alto.



(a)



(b)

FIGURA 4.8 Características experimentales de un filtro pasa alto de orden  $M = 79$ . (a) Respuesta impulsiva experimental (cuantificado) diseñada en ADSP-2181. (b) Respuesta en frecuencia.

### 4.1.3 Resultados del Filtro Pasa Banda

Diseñar un filtro pasa banda que cumpla con los siguientes parámetros:

$f_{s1} = 1000$ Hz	Frecuencia de supresión 1
$f_{p1} = 1500$ Hz	Frecuencia de paso 1
$f_{p2} = 2500$ Hz	Frecuencia de paso 2
$f_{s2} = 3000$ Hz	Frecuencia de supresión 2
$f_m = 8000$ Hz	Frecuencia de muestreo
$A = 40$ dB	Atenuación en las de supresión

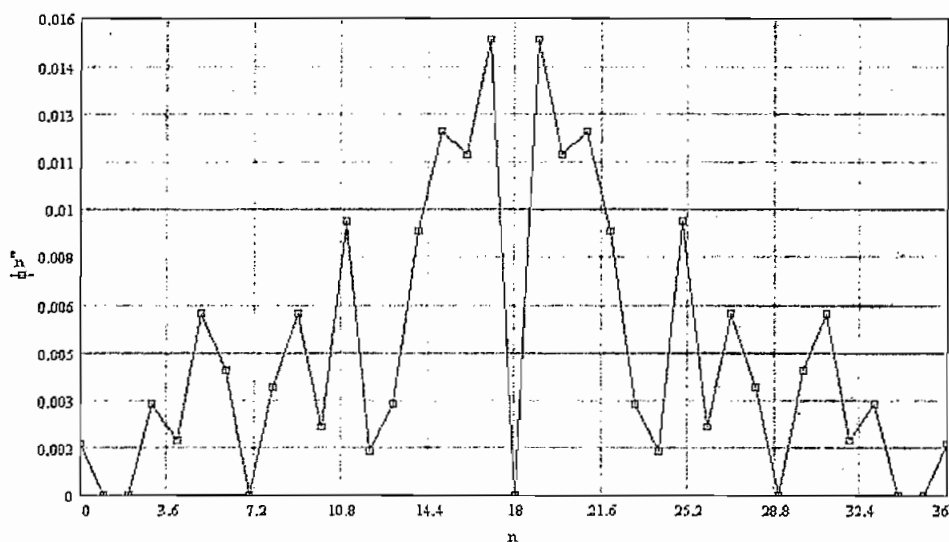
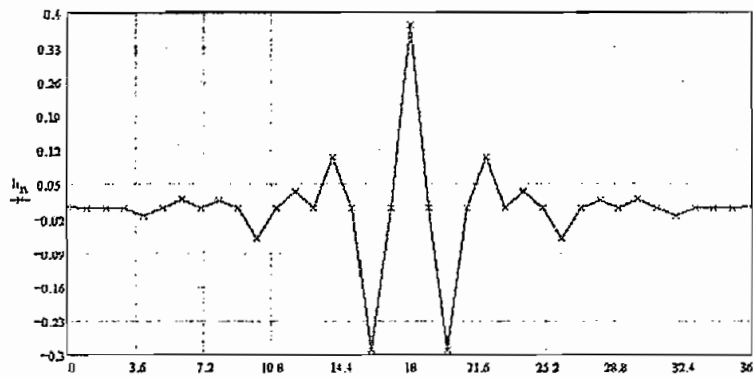


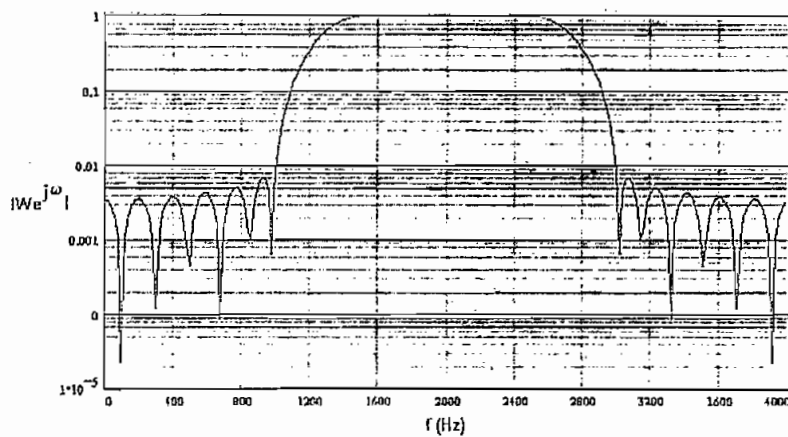
Figura 4.9 Error porcentual de los coeficientes cuantificados y sin cuantificar para el ejemplo (Filtro Pasa Banda).

### Resultados Teóricos

La figura 4.9 representa el error porcentual de los coeficientes cuantificados y sin cuantificar. Los coeficientes teóricos son graficados en la figura 4.10 (a), en donde se encuentra la respuesta impulsiva del filtro de orden 37. En la figura 4.10 (b) se ilustra la gráfica de  $20 \log_{10} |w(e^{j\omega})|$  vs frecuencia [Hz]. Nótese la atenuación de la banda de supresión por debajo de los 40 dB.



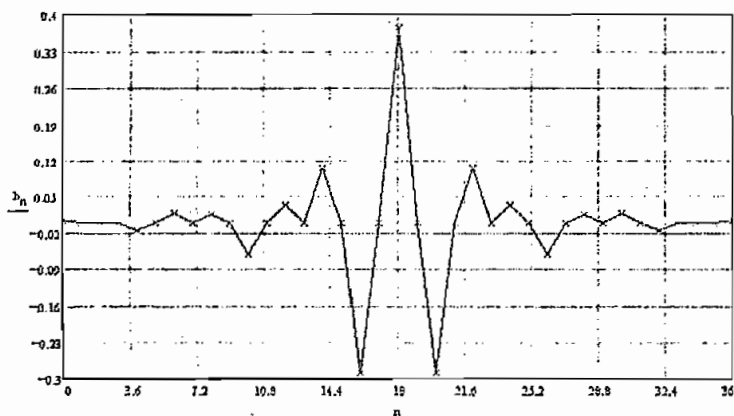
(a)



(b)

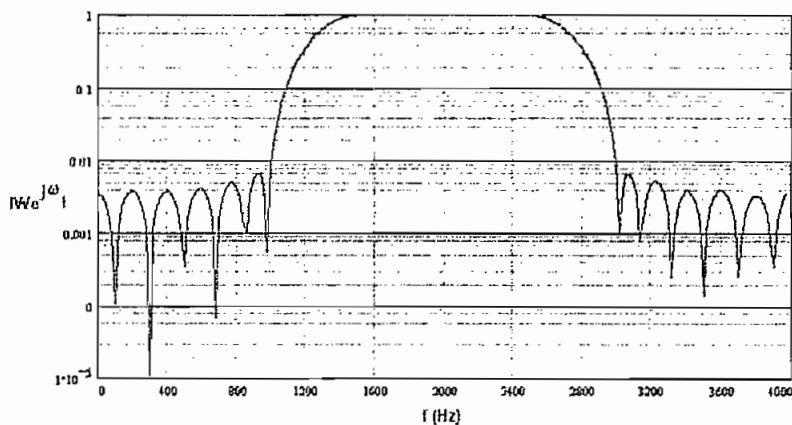
FIGURA 4.10 Características teóricas de un filtro pasa banda de orden  $M = 37$ . (a) Respuesta impulsiva teórica (sin cuantificar) diseñada en *Mathcad*. (b) Respuesta en frecuencia.

### Resultados Experimentales



(a)

Los datos experimentales son ilustrados en la figura 4.11 (a), en donde se puede observar la respuesta impulsiva del filtro de orden 37. En la figura 4.11 (b) se ilustra la respuesta de frecuencia del filtro pasa banda.



(b)

FIGURA 4.11 Características experimentales de un filtro pasa banda de orden  $M = 37$ . (a) Respuesta impulsiva experimental (cuantificado) diseñada en ADSP-2181. (b) Respuesta en frecuencia.

#### 4.1.4 Resultados del Filtro Rechaza Banda

Diseñar un filtro rechaza banda que cumpla con los siguientes parámetros:

$f_{p1} = 1000$ Hz	Frecuencia de supresión 1
$f_{s1} = 1200$ Hz	Frecuencia de paso 1
$f_{s2} = 2000$ Hz	Frecuencia de paso 2
$f_{p2} = 2200$ Hz	Frecuencia de supresión 2
$f_m = 8000$ Hz	Frecuencia de muestreo
$A = 30$ dB	Atenuación en la banda de supresión.

En la figura 4.12 se ilustra el error (porcentual) que existe entre los coeficientes cuantificados y sin cuantificar, se puede observar que el error es menor que el 0.005%.



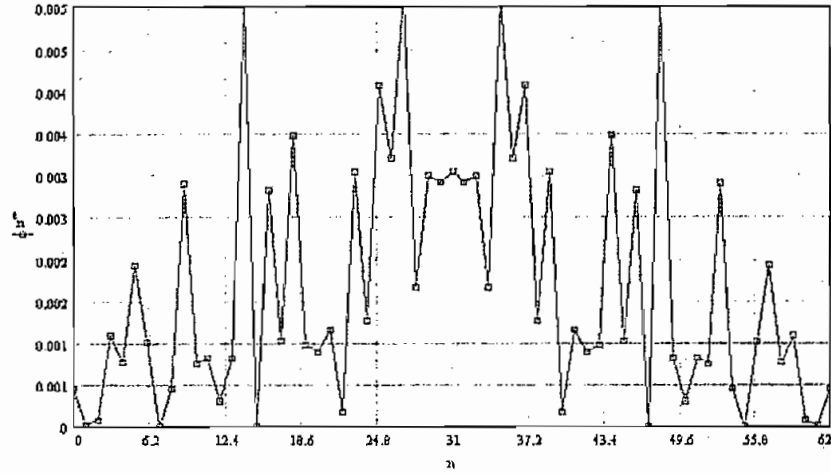
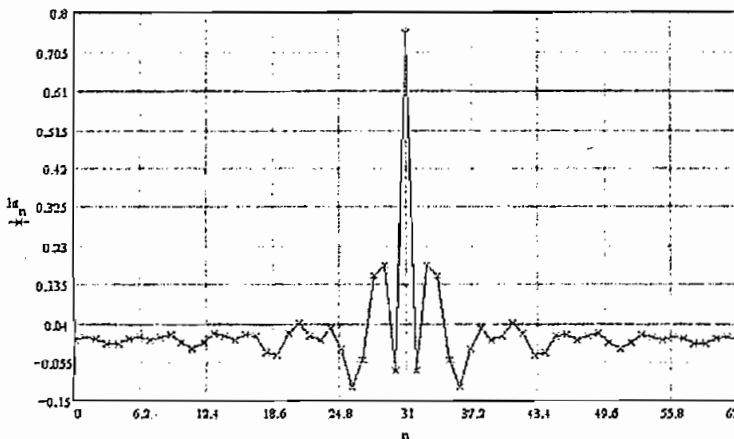


Figura 4.12 Error porcentual de los coeficientes cuantificados y sin cuantificar para el ejemplo (Filtro Rechaza Banda).

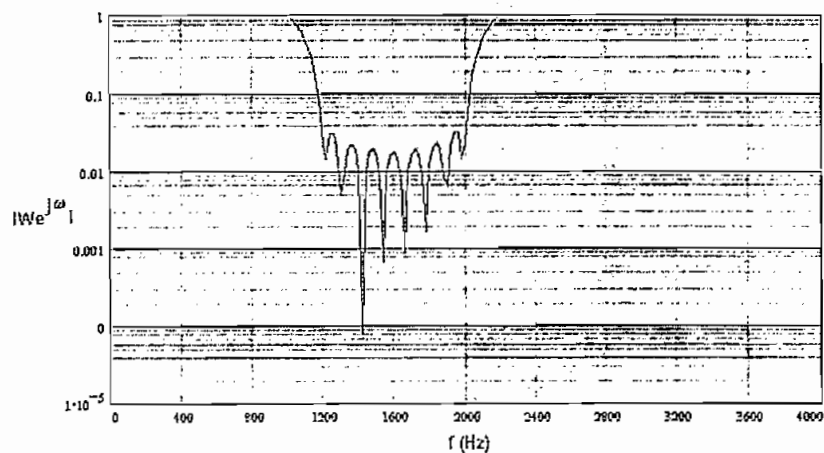
### Resultados Teóricos

Los coeficientes sin cuantificar son graficados en la figura 4.13 (a), en donde se encuentra la respuesta impulsiva del filtro de orden 63. Nótese la simetría que existe en la respuesta impulsiva para  $n = 31$ , éste eje permitirá comparar los lóbulos laterales izquierdo y derecho.

En la figura 4.13 (b) se ilustra la gráfica de  $20 \log_{10} |w(e^{j\omega})|$  vs frecuencia [Hz].



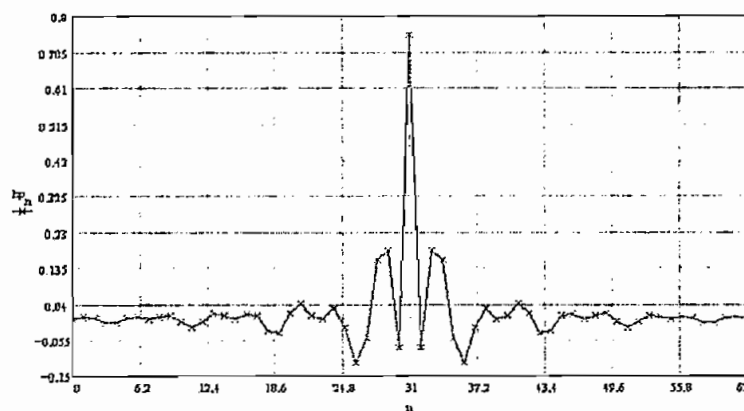
(a)



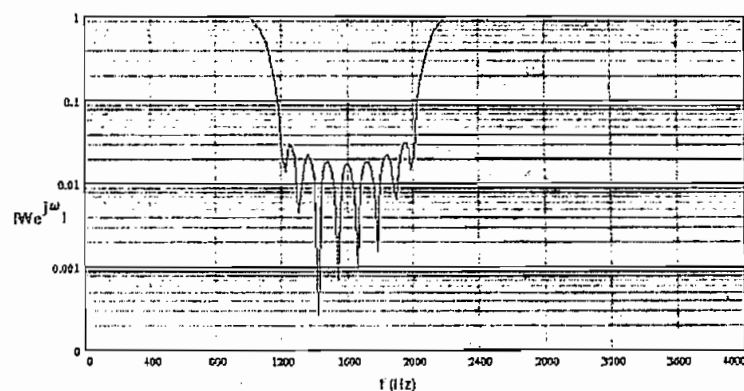
(b)

FIGURA 4.13 Características teóricas de un filtro rechaza banda de orden  $M = 63$ . (a) Respuesta impulsiva teórica (sin cuantificar) diseñada en *Mathcad*. (b) Respuesta en frecuencia.

### Resultados Experimentales



(a)



(b)

FIGURA 4.14 Características experimentales de un filtro rechaza banda de orden  $M = 63$ . (a) Respuesta impulsiva experimental (cuantificado) diseñada en ADSP-2181. (b) Respuesta en frecuencia.

Los coeficientes cuantificados son ilustrados en la figura 4.14 (a), en donde se puede observar la respuesta impulsiva del filtro de orden 63.

## 4.2. ANÁLISIS DE LOS RESULTADOS

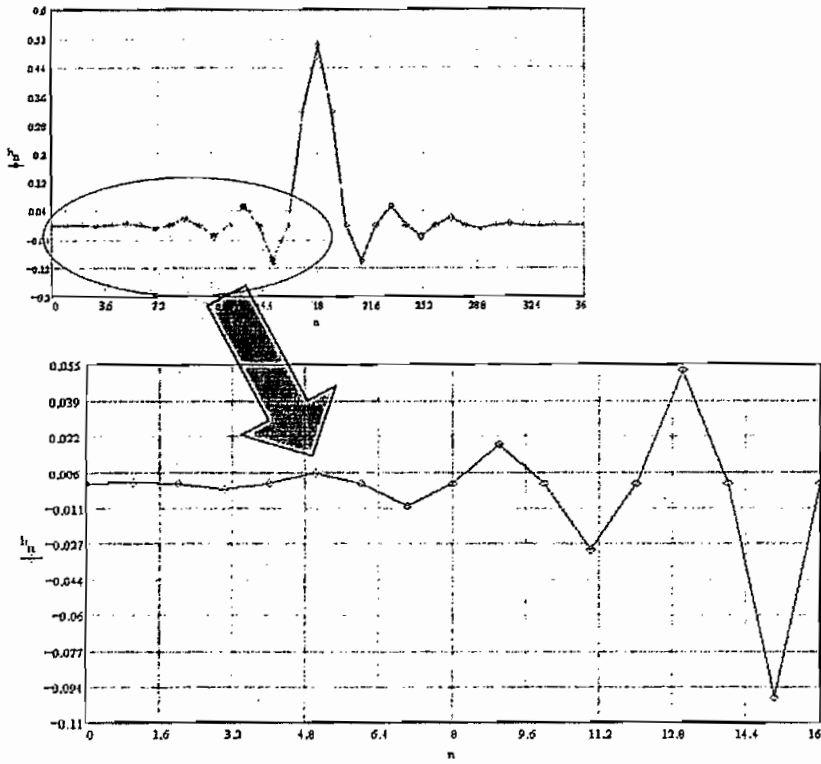
El ADSP-2181 es un microprocesador optimizado para Procesamiento Digital de Señales, sin embargo una de sus fuertes desventajas es precisamente la longitud de sus registros de almacenamiento (el ADSP-2181 trabaja con registros de 16 bits de punto fijo), provocando necesariamente errores de cuantificación.

A simple vista, se puede apreciar que existe una gran similitud entre las respuestas impulsivas teóricas y experimentales. Pero si se analiza nuevamente las gráficas de las figuras 4.2a y 4.3a se observa que los lóbulos laterales correspondientes a los niveles de rizado tanto en la banda de paso como de supresión, varían considerablemente.

Si bien estos valores son reducidos (en el orden de las milésimas de la unidad), una pequeña variación entre un valor teórico y experimental resulta significativa, ya que representa alteraciones en las atenuaciones de la banda de paso y supresión.

Precisamente, la causa de las variaciones entre estos valores se debe a los errores de truncamiento y redondeo introducidos por las limitaciones en los registros de almacenamiento de la ADSP-2181, así como por el redondeo que se origina en las multiplicaciones. Ya que se trabaja con 16 bits; para representar los valores de los coeficientes, el mejor formato que se ajusta es 1.15. Pero precisamente este formato no logra abarcar en sus registros con gran precisión, produciéndose errores significativos de cuantización. En la figura 4.15 se observa mas detalladamente las repuestas impulsivas.

a) TEÓRICO



b) EXPERIMENTAL (DSP)

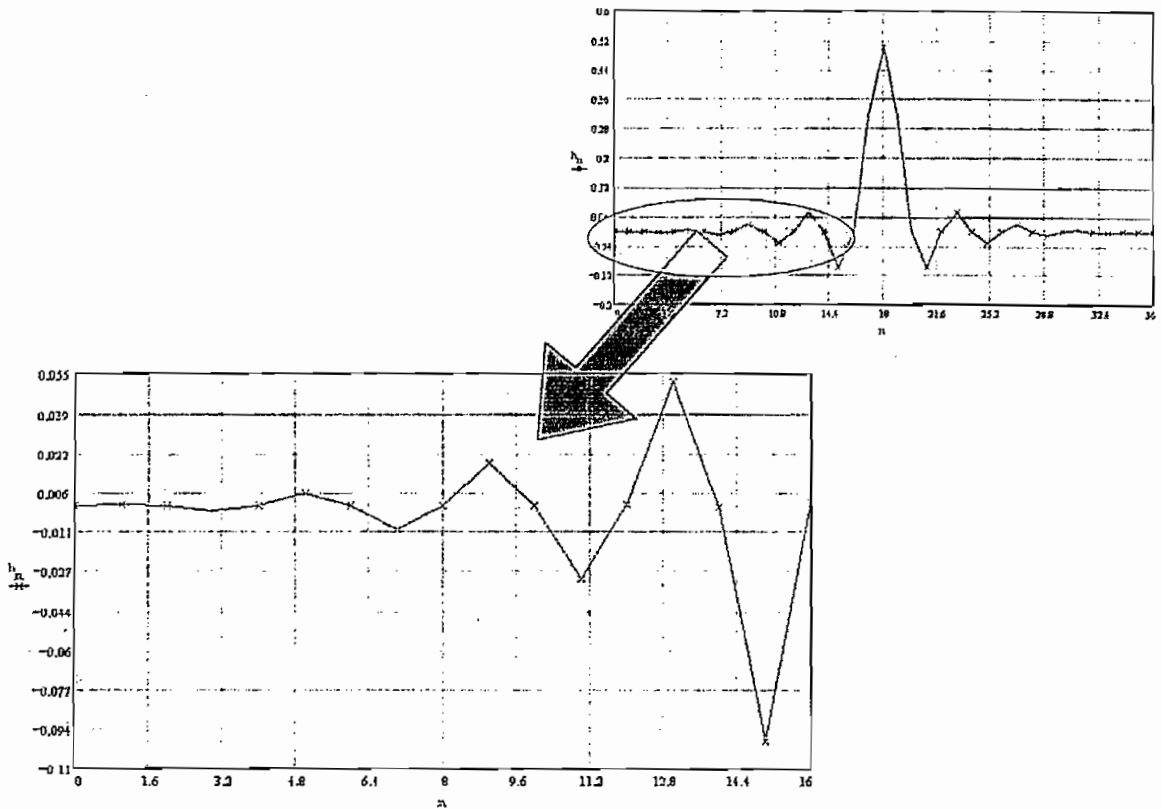
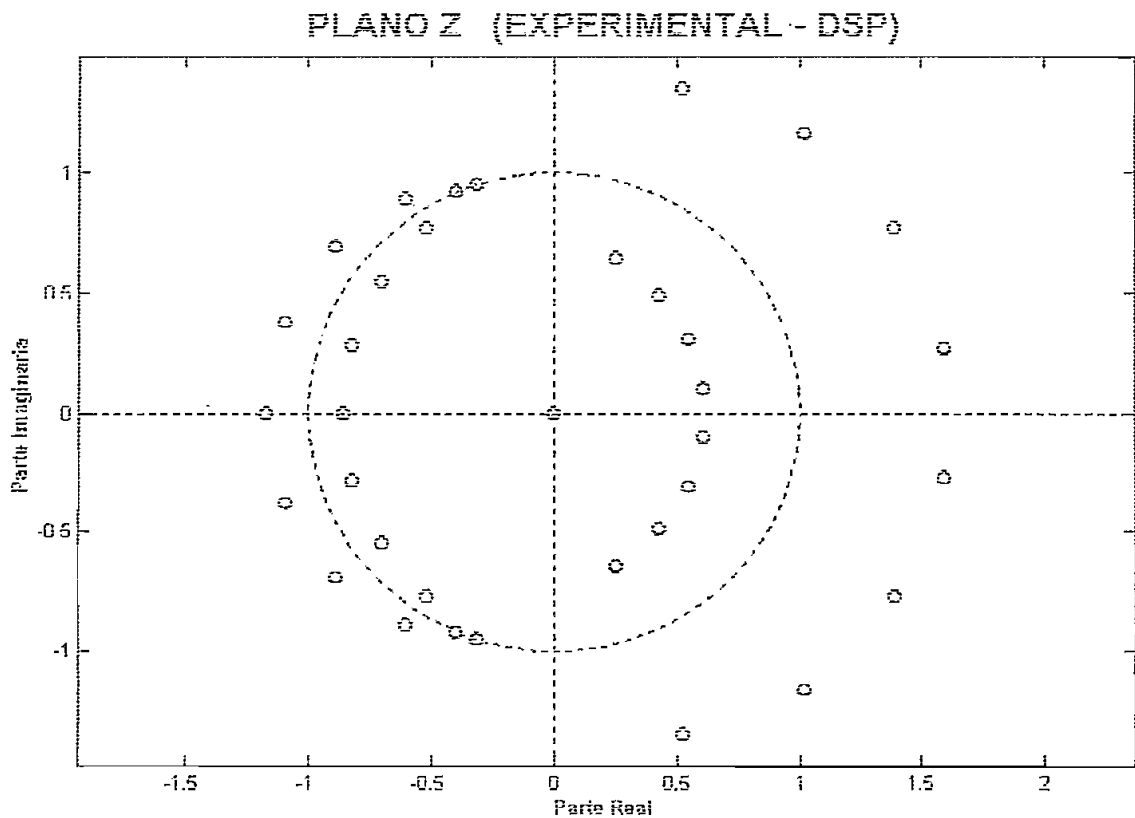


FIGURA 4.15 Ampliación de los lóbulos secundarios de la repuesta impulsiva a) teórica y b) experimental.

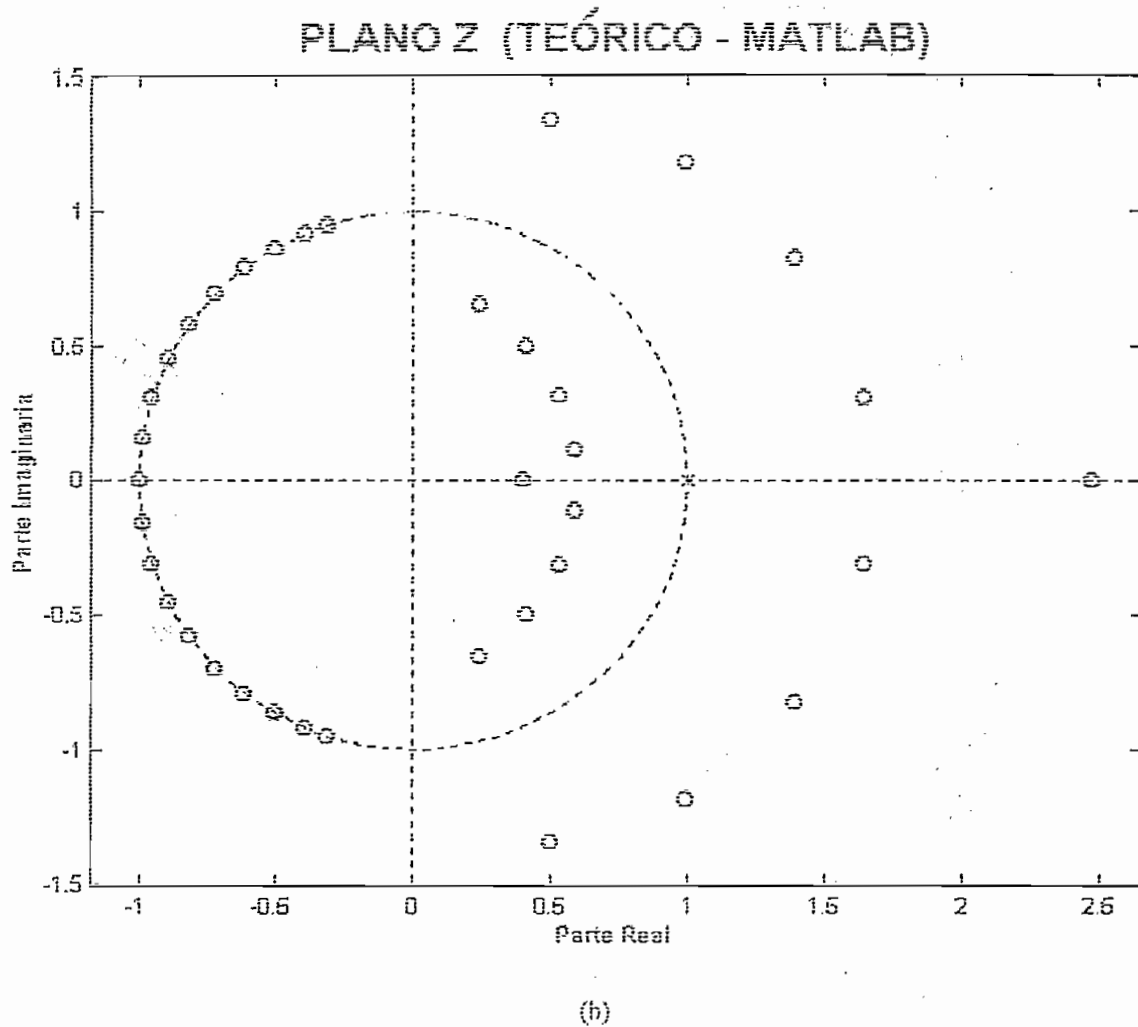
Ahora se va a realizar un análisis comparativo entre las respuestas en frecuencia teórica (FIGURA 4.2b) y experimental (FIGURA 4.3b) del filtro pasa bajo expuesto en la sección 4.1.1. En las respuestas experimentales, se pueden observar errores de precisión en determinados rangos (especialmente en la atenuación de la banda de supresión), esto es debido a los errores de cuantificación y redondeo.

Una forma de analizar la sensibilidad de una estructura FIR, es examinando la sensibilidad de los ceros a los errores de cuantificación de los coeficientes de la respuesta al impulso en el plano  $z$ , que son los coeficientes del polinomio  $H(z)$ .

La figura 4.16 muestra el efecto de cuantificación de los coeficientes en las posiciones de los ceros del filtro. Nótese, que en el caso de no usar cuantificación (datos experimentales), que se muestra en la figura 4.16 (a), los ceros están dispersos por el plano  $z$ , aunque existe un pequeño en el círculo unitario.



(a)



**FIGURA 4.16** Efecto de cuantificación de la respuesta al impulso en los ceros de  $H(z)$ .  
 (a) Sin cuantificación. (b) Cuantificado con 16 bits

Los ceros en el círculo unitario son los principales responsables de la atenuación en la banda de supresión, mientras que los situados en las posiciones inversas conjugadas fuera del círculo unitario son los principales responsables de formar la banda de paso.

Nótese, que al realizar una cuantificación con 16 bits (figura 4.16b), los ceros en la circunferencia unidad se han desplazado de forma significativa, emparejándose entre si y moviéndose fuera de dicha circunferencia, para ocupar posiciones inversas conjugadas.

Este comportamiento de los ceros es completamente consistente con el comportamiento de la respuesta en frecuencia.

### 4.3. APLICACIONES

El filtrado digital es una parte muy importante en el procesamiento digital de señales debido a las numerosas aplicaciones en las que los filtros digitales son empleados. Estas aplicaciones que pueden ser biomédicas, acústicas, sísmicas, de instrumentación y comunicaciones de audio y datos, se resumen en dos usos generales: la separación de señales que han sido mezcladas y la restauración de señales que han sido distorsionadas.

Para describir aplicaciones en las cuales interviene el sistema que se desarrolló en este proyecto es necesario recordar las características que presenta el sistema diseñado.

Teniendo en cuenta cada una de las características, existen aplicaciones en las cuales es imperativo que el sistema cumpla con el filtraje en tiempo real, mientras que en otras, esta característica básicamente no es muy importante y solo se necesita que el sistema realice un correcto filtrado de señales.

Tomando en cuenta todos los precedentes antes mencionados, a continuación se enumerará algunas de las muchas aplicaciones en las cuales se puede implementar el sistema desarrollado, analizando de una manera breve cada aplicación citada.

#### a. Ecuador Digital

Para este tipo de aplicación se puede encontrar dos métodos mediante los cuales se logra sintetizar un filtro digital cuya respuesta en frecuencia se puede controlar fácilmente; esto se realiza del mismo modo como se ajusta la ganancia en los

distintos canales de un ecualizador que se puede encontrar en cualquier equipo de sonido doméstico.

- En el primer método se diseña un filtro FIR de 128 *taps* por muestreo de frecuencia y se utiliza la ventana de Kaiser para mejorar su respuesta en frecuencia.
- En el segundo método se utiliza un filtro IIR de segundo orden por cada canal de ecualización.

De este tipo de aplicación se concluye que un filtro ecualizador digital puede diseñarse de forma sencilla y que la elección de cualquiera de los dos métodos propuestos dependerá de la aplicación que se le da a este filtro digital.

El método de muestreo en frecuencia para el diseño de filtros FIR es bastante sencillo; consiste en tener primeramente una función que represente la respuesta en frecuencia que se desea que tenga el filtro. Luego se discretiza esta función, muestreándola a intervalos regulares. El número de puntos que se tomen determinará cuan exacta será la respuesta del filtro respecto de la respuesta en frecuencia que se desea tener.

Hay que tener presente que en el campo de la frecuencia las funciones son pares respecto del origen, por lo que esto se deberá tener en cuenta cuando se diseñen filtros por este método.

La respuesta impulsiva del filtro se modifica con la ventana de *Kaiser*<sup>1</sup> con  $\beta$  igual a 3.5; la elección de esta ventana y de su parámetro (valor de  $\beta$ ) se la realiza de manera experimental. Otra ventana que mejora la respuesta en frecuencia del filtro de manera, que está se acerque a la que intuitivamente se espera es tanto la ventana de *Hanning* como la de *Hamming*.

---

<sup>1</sup> Datos tomados de estudios realizados en la Universidad Nacional de Ingeniería, Lima-Perú.



Por último cabe mencionar que si se desea aumentar el número de canales de ecualización mediante este método lo único que habría que variar es la función que se muestrea para obtener la respuesta impulso del filtro, lo cual significa un mínimo de esfuerzo.

#### b. Sistemas de altavoces digitales.

Todos los altoparlantes o altavoces convencionales contienen una red *crossover* analógica que divide la señal de audio analógica de entrada en componentes de baja y alta frecuencia que conducen al *woofer*<sup>1</sup> y *tweeter*<sup>2</sup> de los altavoces.

La mayoría de los altoparlantes avanzados pueden contener una tercera componente para rangos de media frecuencia, por lo que este tipo de sistemas no se limita a la utilización de una red *crossover* de dos vías. Las vías a las cuales son conducidas son: *woofer*, *mid-range*<sup>3</sup> y *tweeter*.

Los sistemas de altavoces digitales operan sobre señales de entrada de audio digitalizada y usan filtros digitales FIR o IIR para dividir en bandas de frecuencia apropiadas, las cuales son entonces convertidas a formato analógico, amplificadas y dirigidas a las partes correspondientes de los altoparlantes. Cada altoparlante "digital" se encontrará disponible para estudios digitales profesionales y llegando a ser comercialmente disponible para el uso doméstico (donde típicamente la salida digital de un *CD player* es conectado a la entrada digital de los altoparlantes).

Como un ejemplo ilustrativo se tomará un Filtro *Crossover* de dos vías, el cual posee una frecuencia de corte del filtro pasa bajo y pasa alto conocida como la frecuencia *crossover*.

---

<sup>1</sup> *Woofer* Parte de un altoparlante por el cual la señal de salida posee únicamente componentes de baja frecuencia.

<sup>2</sup> *Tweeter* Parte de un altoparlante por el cual la señal de salida posee únicamente componentes de alta frecuencia.

<sup>3</sup> *Mid-range* Parte de un altoparlante por el cual la señal de salida posee únicamente componentes de frecuencia intermedia.

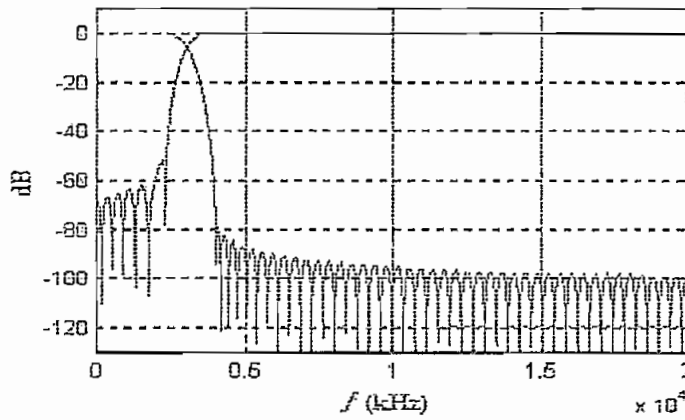
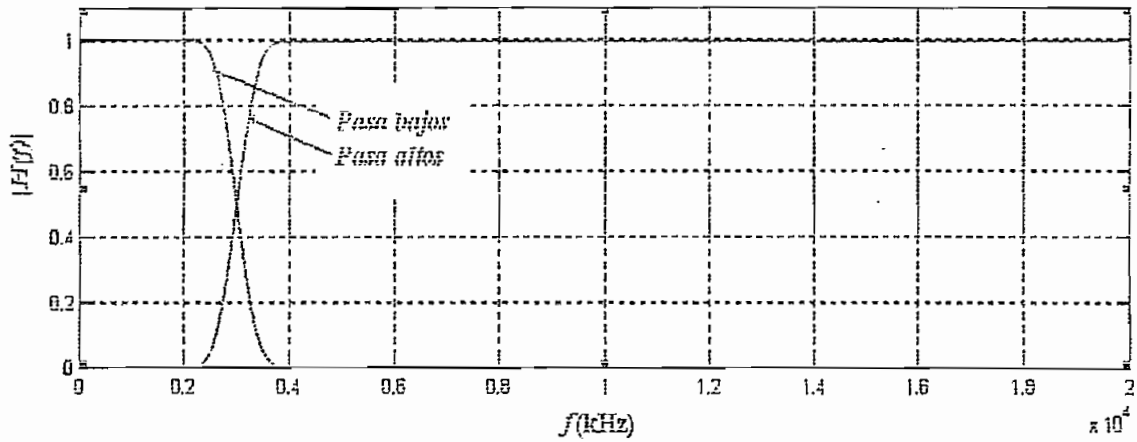


FIGURA 4.17 Magnitud de las respuestas de baja y alta frecuencia en filtros *Crossover* de dos vías.

El diseño de filtros que conducen alta frecuencia y baja frecuencia son entonces:  
para  $n = 0, 1, \dots, M-1$

$$h_{LP}(n) = w(n)d_{LP}(n - \alpha) = w(n) \left[ \frac{\text{sen}(\omega_c(n - \alpha))}{\pi(n - \alpha)} \right]$$

$$h_{HP}(n) = w(n)d_{HP}(n - \alpha) = w(n) \left[ \delta(n - \alpha) - \frac{\text{sen}(\omega_c(n - \alpha))}{\pi(n - \alpha)} \right]$$

$$h_{HP}(n) = \delta(n - \alpha) - h_{LP}(n)$$

Donde  $w(n)$  es la ventana de *Kaiser* obtenida en el capítulo 2. La magnitud de la respuesta del filtro diseñado es mostrada en la figura 4.17 graficadas ambas en escalas absolutas,  $|H(\omega)|$ , y en decibelios,  $20 \log_{10}|H(\omega)|$ .

Para la elaboración de la figura 4.18, en lugar de la realización de los filtros pasa bajo y pasa alto separadamente, éste requiere solamente el filtro pasa bajo y un retraso múltiple que es originado por la expresión  $\delta(n-\alpha)$ .

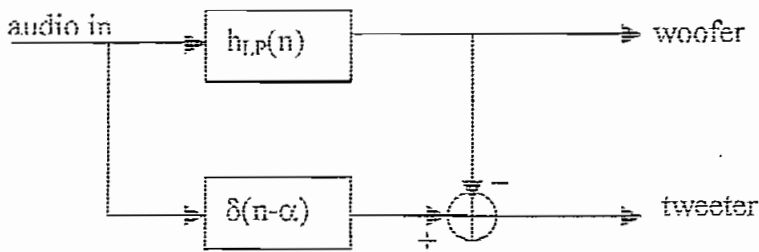


FIGURA 4.18. Implementación complementaria de filtros *crossover* de dos vías.

### c. Cancelación de Dereverberación y *Cross-Talk* en sonido.

En sistemas de reproducción de sonido las señales son distorsionadas por reverberación, así como efectos de *cross-talk* producidos en sistemas *stereo*. Las diferentes técnicas que se utilizan para evitar estos problemas son basadas en bancos de filtros ubicados antes de la fuente.

En sistemas de reproducción de sonido, la música y señales de voz son distorsionadas por la presencia de paredes reflectantes cerca, produciendo reverberación. Además, en sistemas *stereo* los efectos de *cross-talk* son producidos por la emisión de señales desde el altavoz izquierdo que llegan al oído derecho y viceversa.

Un filtro ecualizador ubicado entre la fuente de sonido y el amplificador de poder es generalmente utilizado a fin de eliminar los varios problemas acústicos. El

método de ecualización más común usa un ecualizador gráfico, el cual es compuesto de múltiples filtros pasa banda.

La figura 4.19 presenta un esquema básico que debe cumplir el sistema a fin de solucionar considerablemente el inconveniente que presenta la dereverberación y *cross-talk*.

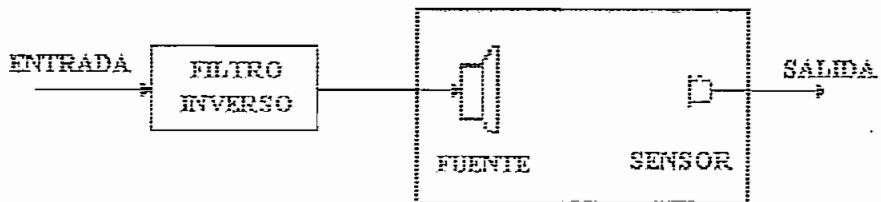


FIGURA 4.19 Sistema acústico consistente de un altoparlante y un micrófono.

## CAPITULO 5

### CONCLUSIONES Y RECOMENDACIONES

---

#### 5.1 CONCLUSIONES

- Los códigos que implementan los algoritmos presentados en este proyecto, están basados en dos características del ADSP-2181: el uso de **los registros circulares**, que son necesarios en este tipo de aplicaciones iterativas en tiempo real; y la **Multiplicación-Acumulación** agregado al **redondeo** y **movimientos paralelos** en una sola instrucción, que ayuda a ahorrar líneas de código, haciendo a los filtros más rápidos.
- Para poder optimizar al máximo el tiempo de procesamiento, tanto los estados como los valores característicos de los filtros son almacenados en buffers circulares.
- Existen muchas aplicaciones del procesamiento de señales en las cuales las relaciones de **fase** son importantes y no deben ser alteradas por el filtrado, debido a esto se deben diseñar filtros que posean fase lineal, como por ejemplo los filtros FIR. A estos filtros se les llaman de fase lineal o no dispersivos.
- Este trabajo está orientado a realizar aplicaciones generales de filtrado de señales. Un filtro digital cualquiera es definido por los coeficientes de la respuesta al impulso del filtro (también llamado el "kernel" del filtro). Estos coeficientes deben diseñarse partiendo de las características en frecuencia del filtro deseado, ya sea usando teoría de Procesamiento Digital de Señales o algún software (en el presente proyecto se usa MATLAB 5.3 y MATHCAD 4.0). Luego de introducidos los coeficientes de la respuesta al impulso del filtro a un registro circular, son convolucionados con la señal digitalizada de entrada.

- En general, los Filtros Digitales implementados sobre Procesadores Digitales de Señales tienen un muy alto rendimiento, esta es una de las razones por las que los DSP se han vuelto tan populares. Sin embargo, se puede apreciar que existen errores entre las respuestas en frecuencia teóricas y experimentales, esto es en gran parte debido a las limitaciones que presenta el ADSP-2181 al trabajar con representaciones de 16 bits en sus registros.
- Básicamente hay dos métodos para el diseño de filtros no recursivos. El primero trata de definir la respuesta en frecuencia del filtro para luego determinar los coeficientes del filtro mediante la transformada inversa de Fourier; mientras que la segunda estrategia utiliza métodos de optimización los cuales pueden ir modificando los coeficientes del filtro para aproximarlos a la respuesta de frecuencia deseada.
- A medida que el orden del filtro crece, el ancho del lóbulo principal decrece. Esta característica se relaciona con la región de transición entre la banda de paso y la de supresión, esto es, cuando se aumenta el orden del filtro, el corte entre estas bandas será más abrupto.
- Los filtros no recursivos tienen ventajas muy interesantes que les hacen ser ampliamente utilizados en múltiples aplicaciones. La característica más destacable es su facilidad de diseño para conseguir una respuesta en frecuencias de fase lineal. Los FIR son por su propia constitución estables, no habiendo problemas en su diseño o en su fase de implementación.
- En la práctica, los filtros FIR se emplean en problemas de filtrado donde hay un requisito de fase lineal dentro de la banda de paso del filtro. Si no existe este requisito se pueden emplear tanto filtros FIR como IIR. Sin embargo, como regla general, un filtro IIR tiene menos rizado y el corte es más abrupto que un FIR con el mismo grado de polinomio.
- Si se puede tolerar alguna distorsión de fase o ésta no es importante, se prefiere un filtro IIR, principalmente porque su implementación involucra

menos parámetros, requiere menos memoria y tiene menor complejidad computacional.

- Los métodos de diseño de filtros FIR utilizan métodos de prueba y error hasta encontrar el filtro que satisfaga las características deseadas. Un método interesante ha sido desarrollado por Kaiser y permite diseñar filtros FIR con características predefinidas.
- Se ha observado que la causalidad del filtro requiere del truncamiento de la respuesta impulsiva, y la característica más destacable de los filtros FIR (fase lineal), hizo introducir la condición de simetría o antisimetría en los coeficientes del FIR. Sin embargo, aunque el orden del filtro sea elevado y se impongan condiciones de simetría, los rizados tanto en la banda de paso como en la de supresión se mantienen.
- Para el caso de ventanas rectangulares los lóbulos laterales son altos y de hecho, a medida que  $m$  crece las amplitudes del pico del lóbulo principal y de los lóbulos secundarios crecen de forma que el área encerrada en cada lóbulo permanece constante mientras que su anchura decrece con  $m$ . Como el área del lóbulo permanece constante al crecer  $m$ , las oscilaciones ocurren más rápidamente, pero no disminuye su amplitud al aumentar  $m$ .
- Los efectos de cuantización se producen al tener obligatoriamente que truncar (o cuantizar) los coeficientes del filtro y las señales de entrada y salida. Esta cuantización puede dar lugar a que las características del filtro realizado difieran de las especificaciones del filtro diseñado.
- Los efectos de cuantización deben ser tenidos muy en cuenta cuando el diseño se realiza en microprocesadores con aritmética de punto fijo (por ejemplo, ADSP-2181). En caso de utilizar micros de 32 bits con aritmética en punto flotante, los efectos de cuantización pueden ser despreciados.

## 5.2 RECOMENDACIONES

- Uno de los objetivos de este trabajo es que el procedimiento de implementación de filtros digitales presentado pueda ser usado como parte aplicativa de los cursos relacionados con el procesamiento digital de señales.
- El presente trabajo brinda una herramienta; a más de la implementación en el ensamblador del ADSP-2181, de algoritmos de filtrado conocidos en el procesamiento digital de señales. El código utilizado en el ADSP-2181 es fácilmente transportable a otros procesadores de la familia ADSP -21xx de *Analog Devices*, por lo que se recomienda realizar un estudio minucioso en el lenguaje ensamblador del procesador digital.
- El método de las ventanas es una técnica relativamente simple para diseñar filtros FIR de fase lineal. Sin embargo, poseen algunas pequeñas desventajas que pueden hacerlos indeseables para algunas aplicaciones, una de estas anomalías es el rizado no constante en las frecuencias críticas por lo que es recomendable que el presente proyecto se lo complemente con algoritmos que logren como resultado un rizado constante.



**1. REPRESENTACIONES Y FORMATOS NUMÉRICOS**

**2. REGISTROS DE ESTADO E INTERRUPCIONES DEL  
ADSP-2181**

**3. PROGRAMAS DESARROLLADOS PARA LA  
IMPLEMENTACIÓN DE FILTROS DIGITALES**

**4. IMPLEMENTACIÓN EN HARDWARE**

**5. FOTOS Y VISUALIZACIÓN EN EL OSCILOSCOPIO**

# ANEXO 1

## FORMATOS Y REPRESENTACIONES NUMÉRICAS

---

La familia de procesadores ADSP-2100 trabaja con datos de 16-bits (punto fijo) en hardware. Características especiales en las unidades computacionales permiten que puedan soportar otros formatos en software.

### REPRESENTACIÓN CON SIGNO O SIN SIGNO: FORMATOS EN COMPLEMENTO DE DOS

Los números binarios sin signo son considerados como positivos, los cuales tienen aproximadamente dos veces la magnitud de un número con signo de la misma resolución binaria. La parte menos significativa de los números de precisión doble (números representados con 32 bits) son tratados como números sin signo. En todas las referencias que se hacen de la aritmética del ADSP-2181, los números con signo se refieren a números representados complemento de dos.

### REPRESENTACIÓN ENTERA Y FRACCIONARIA

El ADSP-2181 admite formatos de datos enteros o fraccionarios. En la figura A1.1 se muestran las características del formato de números enteros. En el formato de números enteros, el punto binario se encuentra localizado a la derecha del bit menos significativo, por lo cual el peso de cada uno de los bits es mayor o igual a 1. Se debe notar que para los números negativos (complemento de dos) el bit de signo tiene un peso negativo.

En el formato de números fraccionarios, el punto binario se encuentra dentro del número, por lo cual alguno o todos los bits tienen un peso menor que uno. En el formato mostrado en la figura A1.2, el punto binario se localiza a la izquierda de los tres bits menos significativos.

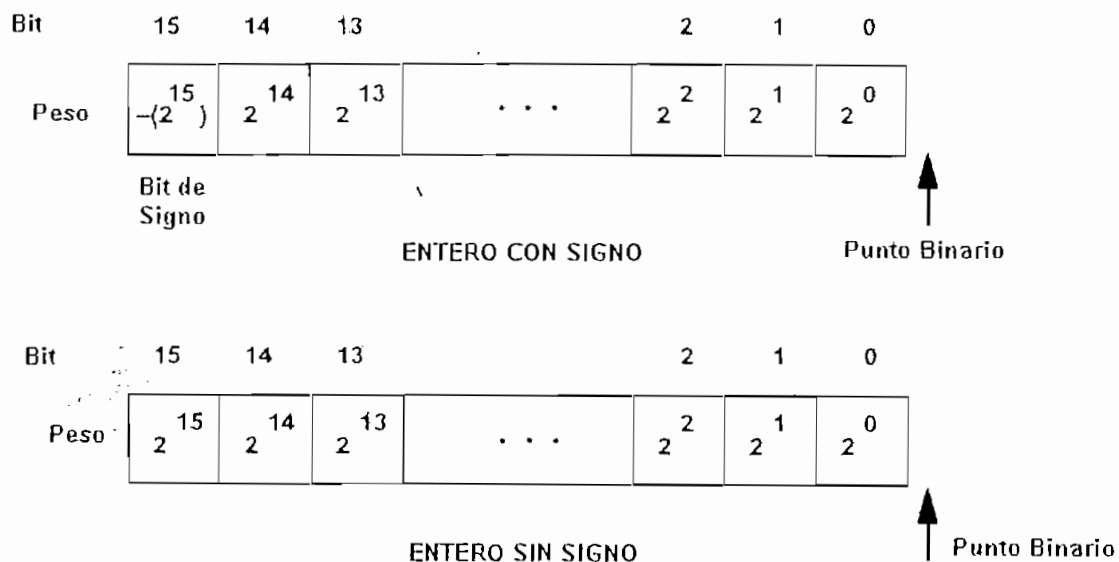


Figura A1.1. Representación en formato entero

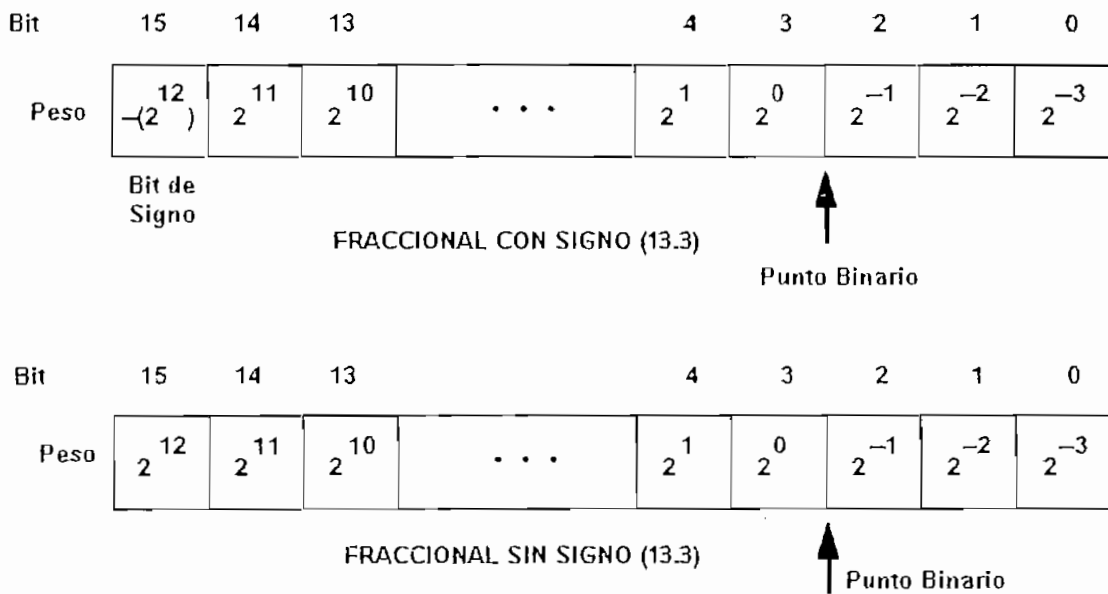


Figura A1.2 Representación en formato fraccionario

La notación utilizada para escribir un formato consiste de dos números separados por un punto (.), el primero indica el número de bits a la izquierda del punto binario, y el segundo los bits a la derecha de este.

La tabla A1.1 indica el rango de los números representados en el formato fraccionario que son posibles de realizar con 16 bits.

Formato	N* de bits enteros	N* de bits binarios	Máximo valor positivo en decimal (0x7FFF)	Máximo valor negativo en decimal (0x8000)	Valor de 1 LSB (bit menos significativo) en decimal (0x0001)
1.15	1	15	0.999969482421875	-1.0	0.000030517578125
2.14	2	14	1.999938964843750	-2.0	0.000061035156250
3.13	3	13	3.999877929687500	-4.0	0.000122070312500
4.12	4	12	7.999755859375000	-8.0	0.000244140625000
5.11	5	11	15.999511718750000	-16.0	0.000488281250000
6.10	6	10	31.999023437500000	-32.0	0.000976562500000
7.9	7	9	63.998046875000000	-64.0	0.001953125000000
8.8	8	8	127.996093750000000	-128.0	0.003906250000000
9.7	9	7	255.992187500000000	-256.0	0.007812500000000
10.6	10	6	511.984375000000000	-512.0	0.015625000000000
11.5	11	5	1023.968750000000000	-1024.0	0.031250000000000
12.4	12	4	2047.937500000000000	-2048.0	0.062500000000000
13.3	13	3	4095.875000000000000	-4096.0	0.125000000000000
14.2	14	2	8191.750000000000000	-8192.0	0.250000000000000
15.1	15	1	16383.500000000000000	-16384.0	0.500000000000000
16.0	16	0	32767.000000000000000	-32768.0	1.000000000000000

Tabla A1.1 Formatos fraccionarios y sus rangos.

## MULTIPLICACIÓN BINARIA

En las sumas y restas, ambos formatos pueden estar en el mismo formato (con o sin signo, y el punto binario en la misma localización); y el formato resultante está en el mismo formato de las entradas. Las sumas y las restas son ejecutadas de la misma manera cuando las entradas son con signo o sin signo.

En la multiplicación, en cambio, las entradas pueden ser de diferente formato, y el resultado dependerá de esos formatos, es decir, la localización del punto binario en el resultado puede ser derivado desde su localización en cada una de las entradas. El producto de dos números de 16 bits, es otro de 32 bits. Si los formatos de las entradas son M.N y P.Q, el producto tiene el formato

$(M+N).(N+Q)$ . Por ejemplo el producto de dos números de 13.3, es un número de 26.6. El producto de dos números de 1.15 es uno de 2.30.

Regla General	Ejemplo: 4 bits	Ejemplo: 16 bits
$\begin{array}{r} M.N \\ \times P.Q \\ \hline \end{array}$	$\begin{array}{r} 1.111 \quad 1.3 \text{ formato} \\ \times 11.11 \quad 2.2 \text{ formato} \\ \hline \end{array}$	$\begin{array}{r} 5.3 \quad 1.15 \\ \times 5.3 \quad \times 1.15 \\ \hline \end{array}$
$(M+P) . (N+Q)$	$\begin{array}{r} 1111 \\ 1111 \\ 1111 \\ 1111 \\ \hline 111.00001 \end{array}$	$\begin{array}{r} 10.6 \quad 2.30 \\ \hline \end{array}$
	$3.5 \text{ formato} = (1+2) . (2+3)$	

## FORMATOS EN PUNTO FIJO Y EN PUNTO FLOTANTE

Un número en punto flotante tiene un exponente que indica la posición del punto binario en el valor actual. En formato de punto flotante, un set de valores de datos muestran un exponente común. Para convertir un bloque de valores en formato de punto fijo a un formato en punto flotante, se puede desplazar cada valor izquierdo por igual cantidad y guardar el valor desplazado como un bloque exponente.

Típicamente, un bloque en formato de punto flotante permite desplazar la parte mas significativa MSB, e incrementar la precisión de cada valor. Se puede usar un formato de punto flotante para eliminar la posibilidad de datos cuyo valor se desborda.

## ANEXO 2

### REGISTROS DE ESTADO E INTERRUPCIONES DEL ADSP-2181

Cuando se produce una interrupción, el controlador de interrupciones del secuenciador de programa responde a éstas, transfiriendo el control del programa a la instrucción localizada en el vector de direcciones de interrupción apropiado.

#### INTERRUPCIONES

Las localidades del vector de interrupciones están situadas cada cuatro localidades de memoria de programa entre sí. Esto permite que cortas rutinas de interrupción sean codificadas en este lugar, sin necesidad de saltos para la atención de la rutina requerida. Sin embargo, para rutinas de atención de interrupciones con más de cuatro instrucciones, el control de programa debe ser transferido a la subrutina requerida por medio de una instrucción de salto localizada en la localidad del vector de interrupciones apropiado.

Después de que se ha atendido una interrupción, una instrucción RTI (retorno de interrupción) vuelve el control del programa a la rutina principal. La dirección en la parte superior de la Pila del Contador de Programa es leída, y se utiliza como la dirección de retorno al programa principal.

Cuando ocurre una petición de interrupción, esta es mantenida mientras el procesador termina la ejecución de la instrucción actual. El controlador de interrupciones compara la petición de interrupción con el registro IMASK (Registro de Enmascaramiento de Interrupciones). Si la instrucción no está enmascarada, el secuenciador del programa almacena el valor presente en el contador del programa en la pila del PC. Esto permite que la ejecución continúe, después que se atiende la interrupción, con la siguiente instrucción del programa principal.

La tabla A2.1 muestra las interrupciones y las direcciones de los vectores asociados para el ADSP-2181.

FUENTE DE LA INTERUPCIÓN	DIRECCIÓN DEL VECTOR DE INTERRUPTIONES
RESET startup	0x0000 (Prioridad más alta)
Powerdown	0x002C
IRQ2	0x0004
IRQL1 (Activado por nivel)	0x0008
IRQL0 (Activado por nivel)	0x000C
SPORT0 (Transmisión)	0x0010
SPORT0 (Recepción)	0x0014
IRQE (Activado por flanco)	0x0018
Byte DMA (BDMA) Interrupción	0x001C
SPORT1 Transmisión o IRQ1	0x0020
SPORT1 Recepción o IRQ0	0x0024
Timer	0x0028 (Prioridad más baja)

**TABLA A2.1** Interrupciones y direcciones del vector de interrupciones del ADSP-2181

## REGISTROS DE ESTADO Y PILA DE ESTADO

Los bits de estado y de modo de procesador son almacenados en registros internos los cuales pueden ser leídos y escritos independientemente por medio del bus DMD. Estos registros son:

ASTAT	Registro de estado aritmético
SSTAT	Registro de estado de Pila (solamente de lectura)
MSTAT	Registro de estado de Modo
ICNTL	Registro de control de Interrupciones
IMASK	Registro de Enmascaramiento de Interrupciones
IFC	Registro de Forzamiento / Eliminación de interrupciones

Los contenidos de los registros ASTAT, MSTAT, e IMASK son almacenados en la Pila de Estado cuando el procesador responde a una interrupción. Estos valores son leídos de la Pila de Estado luego de que se retorna de la subrutina de interrupción con la instrucción RTI.

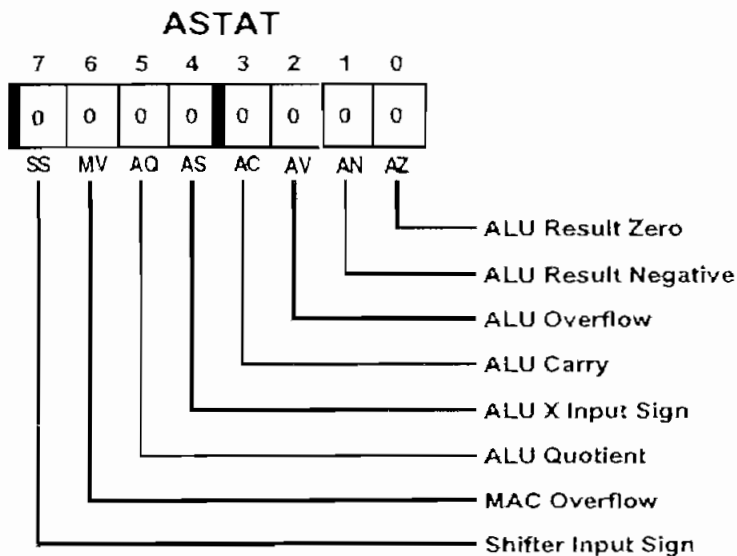
Los registros ICNTL, IMASK e IFC son utilizados para configurar las interrupciones. El registro ICNTL, determina si las interrupciones pueden ser anidadas y configura las interrupciones IRQ2, IRQ1, IRQ0 como sensitivas al nivel, o sensitivas por flanco. IMASK habilita o deshabilita (enmascara) individualmente cada interrupción. El registro IFC fuerza una interrupción (bajo control de software) o elimina una interrupción pendiente (si es activada por flanco).

El **registro de estado aritmético (ASTAT)** es de 8 bits y mantiene la información de estado generada por los bloques computacionales del procesador.

El **registro (SSTAT)** es de 8 bits y mantiene la información de las cuatro pilas (stacks) del procesador. El bits es verdadero si el igual a 1.

El **registro (MSTAT)** determina el modo de operación del procesador.

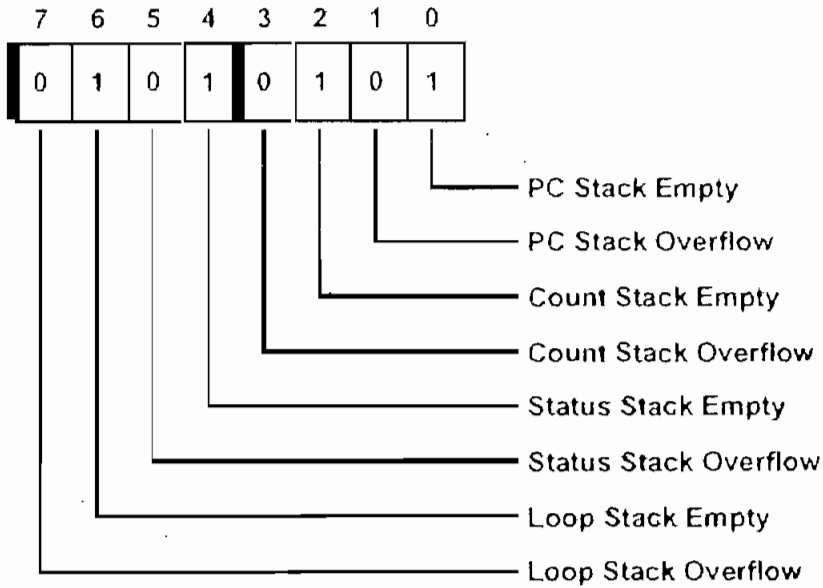
## REGISTRO ASTAT



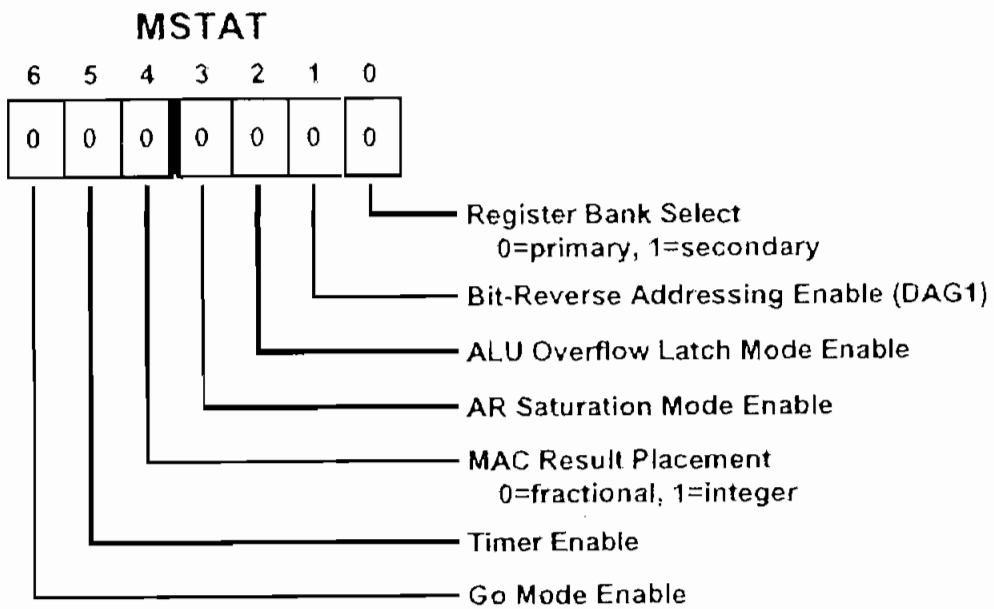
## REGISTRO SSTAT



### SSTAT (read-only)



### REGISTRO MSTAT



## IMPLEMENTACIÓN EN HARDWARE

---

- Data Sheet      SN74LS04    HEX INVERSOR
  
- Data Sheet      LM7512      REGULADOR DE VOLTAJE
  
- Data Sheet      MM74C373   LATCH TIPO D 3-estados
  
- Data Sheet      MM74C922   DECODIFICADOR DE TECLADO 16  
teclas
  
- Data Sheet      Matriz LCD Display
  
- Circuito de conexión

**SN5404, SN54LS04, SN54S04,  
SN7404, SN74LS04, SN74S04  
HEX INVERTERS**

DECEMBER 1983—REVISED MARCH 1988

- Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs
- Dependable Texas Instruments Quality and Reliability

**description**

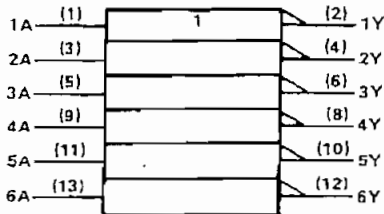
These devices contain six independent inverters.

The SN5404, SN54LS04, and SN54S04 are characterized for operation over the full military temperature range of -55°C to 125°C. The SN7404, SN74LS04, and SN74S04 are characterized for operation from 0°C to 70°C.

FUNCTION TABLE (each inverter)

INPUTS		OUTPUT	
A		Y	
H		L	
L		H	

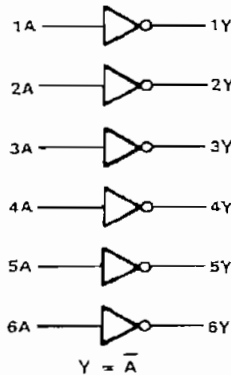
**logic symbol†**



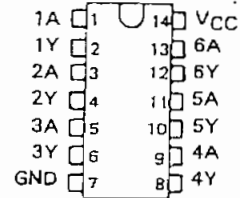
† This symbol is in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.

Pin numbers shown are for D, J, and N packages.

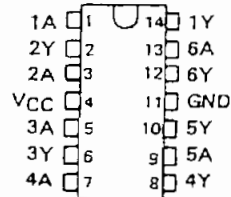
**logic diagram (positive logic)**



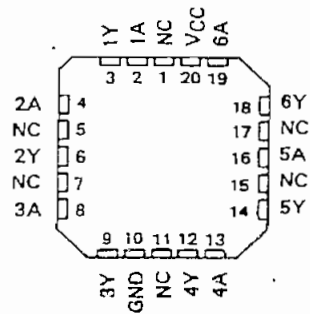
SN5404 . . . J PACKAGE  
SN54LS04, SN54S04 . . . J OR W PACKAGE  
SN7404 . . . N PACKAGE  
SN74LS04, SN74S04 . . . D OR N PACKAGE  
(TOP VIEW)



SN5404 . . . W PACKAGE  
(TOP VIEW)



SN54LS04, SN54S04 . . . FK PACKAGE  
(TOP VIEW)



NC - No internal connection

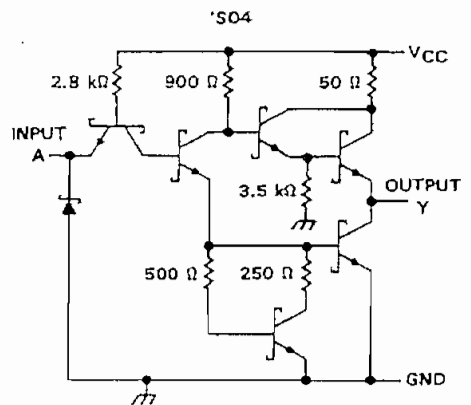
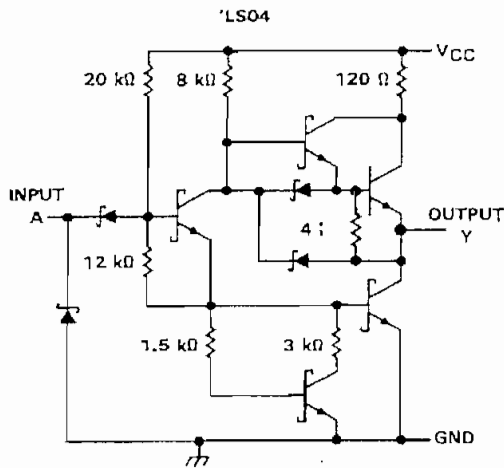
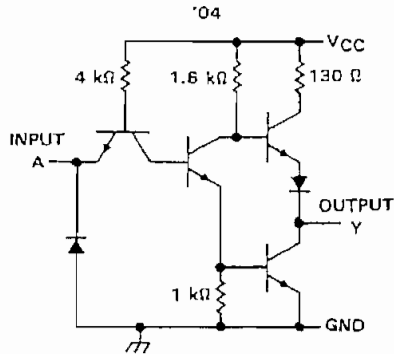
PRODUCTION DATA does not contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



POST OFFICE BOX 955012 • DALLAS, TEXAS 75265

**SN5404, SN54LS04, SN54S04,  
SN7404, SN74LS04, SN74S04  
HEX INVERTERS**

schematics (each gate)



Resistor values shown are nominal.

**absolute maximum ratings over operating free-air temperature range (unless otherwise noted)**

Supply voltage, $V_{CC}$ (see Note 1) .....	7 V
Input voltage: '04, 'S04 .....	5.5 V
'LS04 .....	7 V
Operating free-air temperature range: SN54' .....	-55°C to 125°C
SN74' .....	0°C to 70°C
Storage temperature range .....	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.



POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

# SN5404, SN7404 HEX INVERTERS

## recommended operating conditions

	SN5404			SN7404			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$V_{CC}$ Supply voltage	4.5	5	5.5	4.75	5	5.25	V
$V_{IH}$ High-level input voltage	2			2			V
$V_{IL}$ Low-level input voltage	0.8			0.8			V
$I_{OH}$ High-level output current	-0.4			-0.4			mA
$I_{OL}$ Low-level output current	16			16			mA
$T_A$ Operating free-air temperature	-55			125			$^{\circ}$ C

## electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS †	SN5404		SN7404		UNIT
		MIN	TYP ‡	MAX	MIN	
$V_{IK}$	$V_{CC} = \text{MIN}$ , $I_I = -12 \text{ mA}$	-1.5		-1.5		V
$V_{OH}$	$V_{CC} = \text{MIN}$ , $V_{IL} = 0.8 \text{ V}$ , $I_{OH} = -0.4 \text{ mA}$	2.4	3.4	2.4	3.4	V
$V_{OL}$	$V_{CC} = \text{MIN}$ , $V_{IH} = 2 \text{ V}$ , $I_{OL} = 16 \text{ mA}$	0.2	0.4	0.2	0.4	V
$I_I$	$V_{CC} = \text{MAX}$ , $V_I = 5.5 \text{ V}$	1		1		mA
$I_{IH}$	$V_{CC} = \text{MAX}$ , $V_I = 2.4 \text{ V}$	40		40		$\mu$ A
$I_{IL}$	$V_{CC} = \text{MAX}$ , $V_I = 0.4 \text{ V}$	-1.6		-1.6		mA
$I_{OS} §$	$V_{CC} = \text{MAX}$	-20	-55	-18	-55	mA
$I_{CCH}$	$V_{CC} = \text{MAX}$ , $V_I = 0 \text{ V}$	6	12	6	12	mA
$I_{CCL}$	$V_{CC} = \text{MAX}$ , $V_I = 4.5 \text{ V}$	18	33	18	33	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at  $V_{CC} = 5 \text{ V}$ ,  $T_A = 25^{\circ}\text{C}$ .

§ Not more than one output should be shorted at a time.

## switching characteristics, $V_{CC} = 5 \text{ V}$ , $T_A = 25^{\circ}\text{C}$ (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{PLH}$	A	Y	$R_L = 400 \Omega$ , $C_L = 15 \text{ pF}$		12	22	ns
$t_{PHL}$				8	15	ns	

NOTE 2: Load circuits and voltage waveforms are shown in Section 1.

TEXAS  
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

# SN54LS04, SN74LS04 HEX INVERTERS

recommended operating conditions

	SN54LS04			SN74LS04			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$V_{CC}$ Supply voltage	4.5	5	5.5	4.75	5	5.25	V
$V_{IH}$ High-level input voltage	2			2			V
$V_{IL}$ Low-level input voltage	0.7			0.8			V
$I_{OH}$ High-level output current	-0.4			-0.4			mA
$I_{OL}$ Low-level output current	4			8			mA
$T_A$ Operating free-air temperature	-55			125			$^{\circ}$ C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS †	SN54LS04			SN74LS04			UNIT
		MIN	TYP ‡	MAX	MIN	TYP ‡	MAX	
$V_{IK}$	$V_{CC} = \text{MIN}$ , $I_I = -18 \text{ mA}$	-1.5			-1.5			V
$V_{OH}$	$V_{CC} = \text{MIN}$ , $V_{IL} = \text{MAX}$ , $I_{OH} = -0.4 \text{ mA}$	2.5	3.4		2.7	3.4		V
$V_{OL}$	$V_{CC} = \text{MIN}$ , $V_{IH} = 2 \text{ V}$ , $I_{OL} = 4 \text{ mA}$	0.25		0.4			0.4	V
	$V_{CC} = \text{MIN}$ , $V_{IH} = 2 \text{ V}$ , $I_{OL} = 8 \text{ mA}$				0.25	0.5		
$I_I$	$V_{CC} = \text{MAX}$ , $V_I = 7 \text{ V}$	0.1			0.1			mA
$I_{IH}$	$V_{CC} = \text{MAX}$ , $V_I = 2.7 \text{ V}$	20			20			$\mu$ A
$I_{IL}$	$V_{CC} = \text{MAX}$ , $V_I = 0.4 \text{ V}$	-0.4			-0.4			mA
$I_{OS} §$	$V_{CC} = \text{MAX}$	-20		-100	-20		-100	mA
$I_{CCH}$	$V_{CC} = \text{MAX}$ , $V_I = 0 \text{ V}$	1.2		2.4	1.2		2.4	mA
$I_{CCL}$	$V_{CC} = \text{MAX}$ , $V_I = 4.5 \text{ V}$	3.6		6.6	3.6		6.6	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at  $V_{CC} = 5 \text{ V}$ ,  $T_A = 25^{\circ}\text{C}$ .

§ Not more than one output should be shorted at a time, and the duration of the short-circuit should not exceed one second.

switching characteristics,  $V_{CC} = 5 \text{ V}$ ,  $T_A = 25^{\circ}\text{C}$  (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS		MIN	TYP	MAX	UNIT
$t_{PLH}$	A	Y	$R_L = 2 \text{ k}\Omega$ ,	$C_L = 15 \text{ pF}$	9		15	ns
$t_{PHL}$					10		15	ns

NOTE 2: Load circuits and voltage waveforms are shown in Section 1.

TEXAS  
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

**SN54S04, SN74S04  
HEX INVERTERS**

**recommended operating conditions**

	SN54S04			SN74S04			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$V_{CC}$ Supply voltage	4.5	5	5.5	4.75	5	5.25	V
$V_{IH}$ High-level input voltage	2			2			V
$V_{IL}$ Low-level input voltage	0.8			0.8			V
$I_{OH}$ High-level output current	-1			-1			mA
$I_{OL}$ Low-level output current	20			20			mA
$T_A$ Operating free-air temperature	-55			0			70 °C

**electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)**

PARAMETER	TEST CONDITIONS †	SN54S04			SN74S04			UNIT
		MIN	TYP ‡	MAX	MIN	TYP ‡	MAX	
$V_{IK}$	$V_{CC} = \text{MIN}$ , $I_I = -18 \text{ mA}$	-1.2			-1.2			V
$V_{OH}$	$V_{CC} = \text{MIN}$ , $V_{IL} = 0.8 \text{ V}$ , $I_{OH} = -1 \text{ mA}$	2.5	3.4		2.7	3.4		V
$V_{OL}$	$V_{CC} = \text{MIN}$ , $V_{IH} = 2 \text{ V}$ , $I_{OL} = 20 \text{ mA}$	0.5			0.5			V
$I_I$	$V_{CC} = \text{MAX}$ , $V_I = 5.5 \text{ V}$	1			1			mA
$I_{IH}$	$V_{CC} = \text{MAX}$ , $V_I = 2.7 \text{ V}$	50			50			µA
$I_{IL}$	$V_{CC} = \text{MAX}$ , $V_I = 0.5 \text{ V}$	-2			-2			mA
$I_{OS} §$	$V_{CC} = \text{MAX}$	-40		-100	-40		-100	mA
$I_{CCH}$	$V_{CC} = \text{MAX}$ , $V_I = 0 \text{ V}$	15			15			24 mA
$I_{CCL}$	$V_{CC} = \text{MAX}$ , $V_I = 4.5 \text{ V}$	30			30			54 mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at  $V_{CC} = 5 \text{ V}$ ,  $T_A = 25^\circ\text{C}$ .

§ Not more than one output should be shorted at a time, and the duration of the short circuit should not exceed one second.

**switching characteristics,  $V_{CC} = 5 \text{ V}$ ,  $T_A = 25^\circ\text{C}$  (see note 2)**

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{PLH}$	A	Y	$R_L = 280 \Omega$ , $C_L = 15 \text{ pF}$		3	4.5	ns
$t_{PHL}$					3	5	ns
$t_{PLH}$			$R_L = 280 \Omega$ , $C_L = 50 \text{ pF}$		4.5		ns
$t_{PHL}$					5		ns

NOTE 2: Load circuits and voltage waveforms are shown in Section 1.



## MM74C373 • MM74C374

### 3-STATE Octal D-Type Latch • 3-STATE Octal D-Type Flip-Flop

#### General Description

The MM74C373 and MM74C374 are integrated, complementary MOS (CMOS), 8-bit storage elements with 3-STATE outputs. These outputs have been specially designed to drive high capacitive loads, such as one might find when driving a bus, and to have a fan out of 1 when driving standard TTL. When a high logic level is applied to the OUTPUT DISABLE input, all outputs go to a high impedance state, regardless of what signals are present at the other inputs and the state of the storage elements.

The MM74C373 is an 8-bit latch. When LATCH ENABLE is high, the Q outputs will follow the D inputs. When LATCH ENABLE goes low, data at the D inputs, which meets the set-up and hold time requirements, will be retained at the outputs until LATCH ENABLE returns high again.

The MM74C374 is an 8-bit, D-type, positive-edge triggered flip-flop. Data at the D inputs, meeting the set-up and hold time requirements, is transferred to the Q outputs on positive-going transitions of the CLOCK input.

Both the MM74C373 and the MM74C374 are being assembled in 20-pin dual-in-line packages with 0.300" pin centers.

#### Features

- Wide supply voltage range: 3V to 15V
- High noise immunity: 0.45 V<sub>CC</sub> (typ.)
- Low power consumption
- TTL compatibility:
  - Fan out of 1 driving standard TTL
- Bus driving capability
- 3-STATE outputs
- Eight storage elements in one package
- Single CLOCK/LATCH ENABLE and OUTPUT DISABLE control inputs
- 20-pin dual-in-line package with 0.300" centers takes half the board space of a 24-pin package

#### Ordering Code:

Order Number	Package Number	Package Description
MM74C373M	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C373N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
MM74C374M	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C374N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide

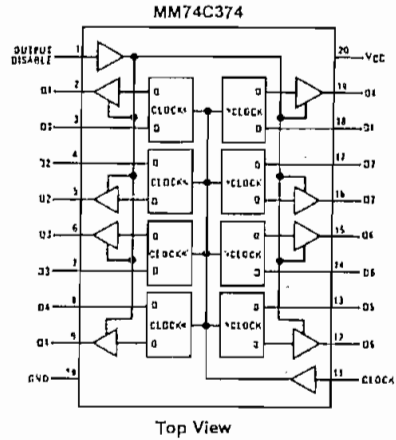
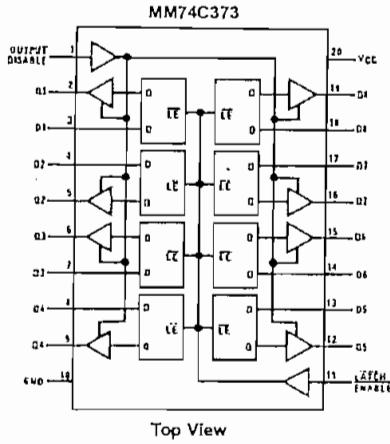
Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

MM74C373 • MM74C374 3-STATE Octal D-Type Latch • 3-STATE Octal D-Type Flip-Flop



### Connection Diagrams

Pin Assignments for DIP and SOIC



### Truth Tables

MM74C373

Output Disable	LATCH ENABLE	D	Q
L	H	H	H
L	H	L	L
L	L	X	Q
H	X	X	Hi-Z

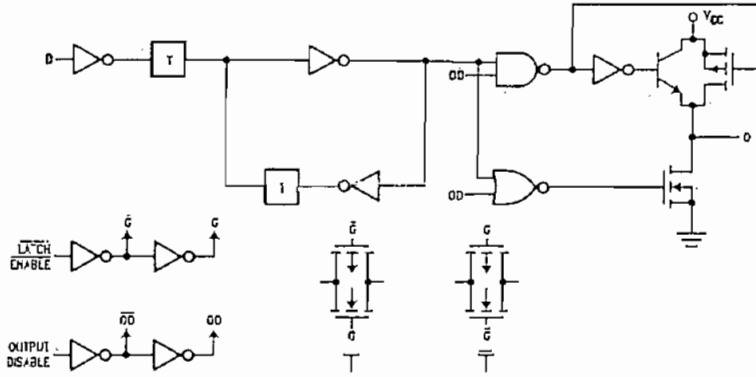
MM74C374

Output Disable	Clock	D	Q
L	↗	H	H
L	↘	L	L
L	L	X	Q
L	H	X	Q
H	X	X	Hi-Z

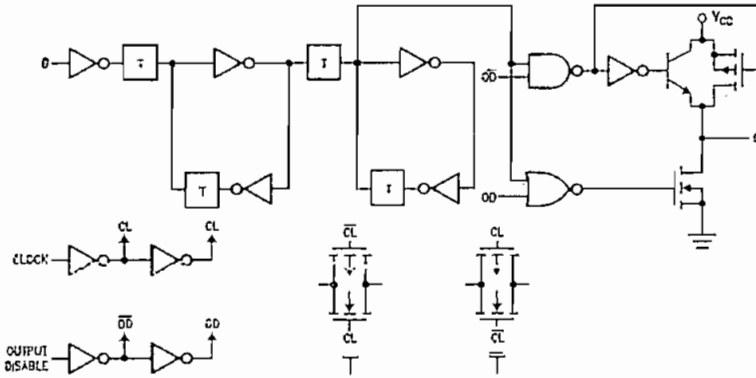
L = LOW logic level  
 H = HIGH logic level  
 X = Irrelevant  
 ↗ = LOW-to-HIGH logic level transition  
 Q = Pre-existing output level  
 Hi-Z = High impedance output state

### Block Diagrams

MM74C373 (1 of 8 Latches)



MM74C374 (1 of 8 Flip-Flops)



Absolute Maximum Ratings (Note 1)		Operating $V_{CC}$ Range	3V to 15V
Voltage at Any Pin	-0.3V to $V_{CC} + 0.3V$	Absolute Maximum $V_{CC}$	18V
Operating Temperature Range ( $T_A$ )		Lead Temperature ( $T_L$ )	260°C
MM74C373	-40°C to +85°C	(Soldering, 10 seconds)	
Storage Temperature Range ( $T_S$ )	-65°C to +150°C		
Power Dissipation			
Dual-In-Line	700 mW		
Small Outline	500 mW		

Note 1: 'Absolute Maximum Ratings' are those values beyond which the safety of the device cannot be guaranteed. Except for 'Operating Temperature Range' they are not meant to imply that the devices should be operated at these limits. The table of 'Electrical Characteristics' provides conditions for actual device operation.

## DC Electrical Characteristics

Min/Max limits apply across temperature range unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
<b>CMOS TO CMOS</b>						
$V_{IN(1)}$	Logical "1" Input Voltage	$V_{CC} = 5V$ $V_{CC} = 10V$	3.5 8.0			V V
$V_{IN(0)}$	Logical "0" Input Voltage	$V_{CC} = 5V$ $V_{CC} = 10V$			1.5 2.0	V V
$V_{OUT(1)}$	Logical "1" Output Voltage	$V_{CC} = 5V, I_O = -10 \mu A$ $V_{CC} = 10V, I_O = -10 \mu A$	4.5 9.0			V V
$V_{OUT(0)}$	Logical "0" Output Voltage	$V_{CC} = 5V, I_O = 10 \mu A$ $V_{CC} = 10V, I_O = 10 \mu A$			0.5 1.0	V V
$I_{IN(1)}$	Logical "1" Input Current	$V_{CC} = 15V, V_{IN} = 15V$		0.005	1.0	$\mu A$
$I_{IN(0)}$	Logical "0" Input Current	$V_{CC} = 15V, V_{IN} = 0V$	-1.0	0.005		$\mu A$
$I_{OZ}$	3-STATE Leakage Current	$V_{CC} = 15V, V_O = 15V$ $V_{CC} = 15V, V_O = 0V$	-1.0	0.005	1.0	$\mu A$ $\mu A$
$I_{CC}$	Supply Current	$V_{CC} = 15V$		0.05	300	$\mu A$
<b>CMOS/LPTTL INTERFACE</b>						
$V_{IN(1)}$	Logical "1" Input Voltage	$V_{CC} = 4.75V$	$V_{CC} - 1.5$			V
$V_{IN(0)}$	Logical "0" Input Voltage	$V_{CC} = 4.75V$			0.8	V
$V_{OUT(1)}$	Logical "1" Output Voltage	$V_{CC} = 4.75V, I_O = -360 \mu A$ $V_{CC} = 4.75V, I_O = 1.6 mA$	$V_{CC} - 0.4$ 2.4			V V
$V_{OUT(0)}$	Logical "0" Output Voltage	$V_{CC} = 4.75V, I_O = 1.6 mA$			0.4	V
<b>OUTPUT DRIVE (Short Circuit Current)</b>						
$I_{SOURCE}$	Output Source Current	$V_{CC} = 5V, V_{OUT} = 0V$ $T_A = 25^\circ C$ (Note 2)	12	24		mA
$I_{SOURCE}$	Output Source Current	$V_{CC} = 10V, V_{OUT} = 0V$ $T_A = 25^\circ C$ (Note 2)	24	-18		mA
$I_{SINK}$	Output Sink Current (N-Channel)	$V_{CC} = 5V, V_{OUT} = V_{CC}$ $T_A = 25^\circ C$ (Note 2)	6	12		mA
$I_{SINK}$	Output Sink Current (N-Channel)	$V_{CC} = 10V, V_{OUT} = V_{CC}$ $T_A = 25^\circ C$ (Note 2)	24	-18		mA

Note 2: These are peak output current capabilities. Continuous output current is rated at 12 mA max.

AC Electrical Characteristics (Note 3)						
MM74C373, T <sub>A</sub> = 25°C, C <sub>L</sub> = 50 pF, t <sub>r</sub> = t <sub>f</sub> = 20 ns, unless otherwise noted						
Symbol	Parameter	Conditions	Min	Typ	Max	Units
t <sub>POD</sub> , t <sub>pd1</sub>	Propagation Delay LATCH ENABLE to Output	V <sub>CC</sub> = 5V, C <sub>L</sub> = 50 pF		165	330	ns
		V <sub>CC</sub> = 10V, C <sub>L</sub> = 50 pF		70	140	ns
		V <sub>CC</sub> = 5V, C <sub>L</sub> = 150 pF		195	390	ns
		V <sub>CC</sub> = 10V, C <sub>L</sub> = 150 pF		85	170	ns
t <sub>POD</sub> , t <sub>pd1</sub>	Propagation Delay Data In to Output	LATCH ENABLE - V <sub>CC</sub>				
		V <sub>CC</sub> = 5V, C <sub>L</sub> = 50 pF		155	310	ns
		V <sub>CC</sub> = 10V, C <sub>L</sub> = 50 pF		70	140	ns
		V <sub>CC</sub> = 5V, C <sub>L</sub> = 150 pF		185	370	ns
	V <sub>CC</sub> = 10V, C <sub>L</sub> = 150 pF		85	170	ns	
t <sub>SET-UP</sub>	Minimum Set-Up Time Data In to CLOCK/LATCH ENABLE	t <sub>HOLD</sub> = 0 ns				
		V <sub>CC</sub> = 5V		70	140	ns
		V <sub>CC</sub> = 10V		35	70	ns
f <sub>MAX</sub>	Maximum LATCH ENABLE Frequency	V <sub>CC</sub> = 5V	3.5	6.7		MHz
		V <sub>CC</sub> = 10V	4.5	9.0		MHz
t <sub>PWH</sub>	Minimum LATCH ENABLE Pulse Width	V <sub>CC</sub> = 5V		75	150	ns
		V <sub>CC</sub> = 10V		55	110	ns
t <sub>r</sub> , t <sub>f</sub>	Maximum LATCH ENABLE Rise and Fall Time	V <sub>CC</sub> = 5V		NA		μs
		V <sub>CC</sub> = 10V		NA		μs
t <sub>TH</sub> , t <sub>OH</sub>	Propagation Delay OUTPUT DISABLE to High Impedance State (from a Logic Level)	R <sub>L</sub> = 10k, C <sub>L</sub> = 5 pF				
		V <sub>CC</sub> = 5V		105	210	ns
	V <sub>CC</sub> = 10V		60	120	ns	
t <sub>HT</sub> , t <sub>HO</sub>	Propagation Delay OUTPUT DISABLE to Logic Level (from High Impedance State)	R <sub>L</sub> = 10k, C <sub>L</sub> = 50 pF				
		V <sub>CC</sub> = 5V		105	210	ns
		V <sub>CC</sub> = 10V		45	90	ns
t <sub>THL</sub> , t <sub>TLH</sub>	Transition Time	V <sub>CC</sub> = 5V, C <sub>L</sub> = 50 pF		65	130	ns
		V <sub>CC</sub> = 10V, C <sub>L</sub> = 50 pF		35	70	ns
		V <sub>CC</sub> = 5V, C <sub>L</sub> = 150 pF		110	220	ns
		V <sub>CC</sub> = 10V, C <sub>L</sub> = 150 pF		70	140	ns
C <sub>IE</sub>	Input Capacitance	LE Input (Note 4)		7.5	10	pF
C <sub>CO</sub>	Input Capacitance	OUTPUT DISABLE input (Note 4)		7.5	10	pF
C <sub>IN</sub>	Input Capacitance	Any Other Input (Note 4)		5	7.5	pF
C <sub>OUT</sub>	Output Capacitance	High impedance State (Note 4)		10	15	pF
C <sub>PD</sub>	Power Dissipation Capacitance	Per Package (Note 5)		200		pF

Note 3: AC Parameters are guaranteed by DC correlated testing.  
 Note 4: Capacitance is guaranteed by periodic testing.  
 Note 5: C<sub>PD</sub> determines the no load AC power consumption of any CMOS device. For complete explanation see Family Characteristics Application Note AN-90.

## AC Electrical Characteristics (Note 6)

MM74C374  $T_A = 25^\circ\text{C}$ ,  $C_L = 50\text{ pF}$ ,  $t_r = t_f = 20\text{ ns}$ , unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$t_{PD}$ , $t_{pd1}$	Propagation Delay, CLOCK to Output	$V_{CC} = 5\text{V}$ , $C_L = 50\text{ pF}$		150	300	ns
		$V_{CC} = 10\text{V}$ , $C_L = 50\text{ pF}$		65	130	ns
		$V_{CC} = 5\text{V}$ , $C_L = 150\text{ pF}$		180	360	ns
		$V_{CC} = 10\text{V}$ , $C_L = 150\text{ pF}$		80	160	ns
$t_{SETUP}$	Minimum Set-Up Time Data In to CLOCK/LATCH ENABLE	$t_{HOLD} = 0\text{ ns}$				
		$V_{CC} = 5\text{V}$		70	140	ns
		$V_{CC} = 10\text{V}$		35	70	ns
$t_{PWH}$ , $t_{PWL}$	Minimum CLOCK Pulse Width	$V_{CC} = 5\text{V}$		70	140	ns
		$V_{CC} = 10\text{V}$		50	100	ns
$f_{MAX}$	Maximum CLOCK Frequency	$V_{CC} = 5\text{V}$	3.5	7.0		MHz
		$V_{CC} = 10\text{V}$	5	10		MHz
$t_{1H}$ , $t_{0H}$	Propagation Delay OUTPUT DISABLE to High Impedance State (from a Logic Level)	$R_L = 10\text{k}$ , $C_L = 50\text{ pF}$				
		$V_{CC} = 5\text{V}$		105	210	ns
		$V_{CC} = 10\text{V}$		60	120	ns
$t_{1L}$ , $t_{0L}$	Propagation Delay OUTPUT DISABLE to Logic Level (from High Impedance State)	$R_L = 10\text{k}$ , $C_L = 50\text{ pF}$				
		$V_{CC} = 5\text{V}$		105	210	ns
		$V_{CC} = 10\text{V}$		45	90	ns
$t_{THL}$ , $t_{TLH}$	Transition Time	$V_{CC} = 5\text{V}$ , $C_L = 50\text{ pF}$		65	130	ns
		$V_{CC} = 10\text{V}$ , $C_L = 50\text{ pF}$		35	70	ns
		$V_{CC} = 5\text{V}$ , $C_L = 150\text{ pF}$		110	220	ns
		$V_{CC} = 10\text{V}$ , $C_L = 150\text{ pF}$		70	140	ns
$t_r$ , $t_f$	Maximum CLOCK Rise and Fall Time	$V_{CC} = 5\text{V}$	15	>2000		$\mu\text{s}$
		$V_{CC} = 10\text{V}$	5	>2000		$\mu\text{s}$
$C_{CLK}$	Input Capacitance	CLOCK Input (Note 7)		7.5	10	pF
$C_{OD}$	Input Capacitance	OUTPUT DISABLE Input (Note 7)		7.5	10	pF
$C_{IN}$	Input Capacitance	Any Other Input (Note 7)		5	7.5	pF
$C_{OUT}$	Output Capacitance	High Impedance State (Note 7)		10	15	pF
$C_{PD}$	Power Dissipation Capacitance	Per Package (Note 8)		250		pF

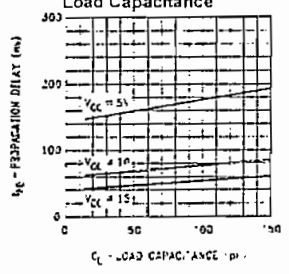
Note 6: AC Parameters are guaranteed by DC correlated testing.

Note 7: Capacitance is guaranteed by periodic testing.

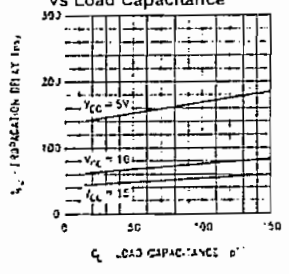
Note 8:  $C_{PD}$  determines the no load AC power consumption of any CMOS device. For complete explanation see Family Characteristics Application Note AN-90.

Typical Performance Characteristics

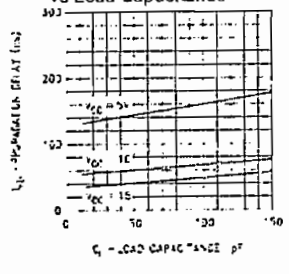
MM74C373  
Propagation Delay, LATCH ENABLE to Output vs Load Capacitance



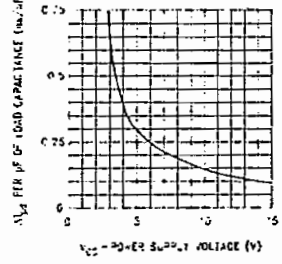
MM74C373  
Propagation Delay, Data In to Output vs Load Capacitance



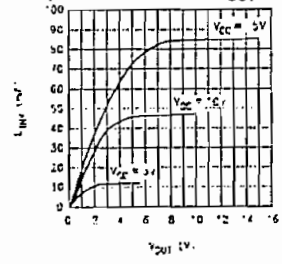
MM74C373  
Propagation Delay, CLOCK to Output vs Load Capacitance



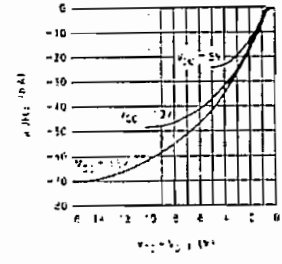
MM74C373, MM74C374  
Change in Propagation Delay per pF of Load Capacitance ( $\Delta t_{pd}/pF$ ) vs Power Supply Voltage



MM74C373, MM74C374  
Output Sink Current vs VOUT

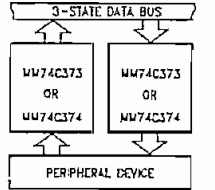


MM74C373, MM74C374  
Source Current vs VCC - VOUT

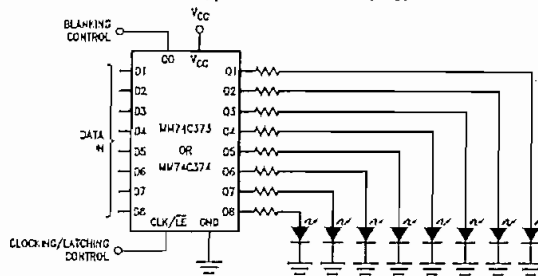


### Typical Applications

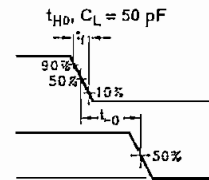
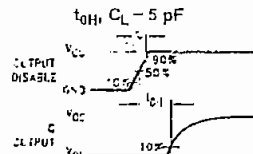
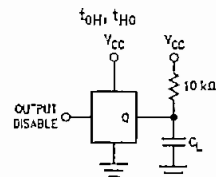
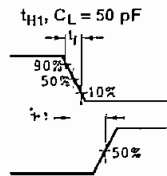
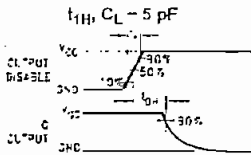
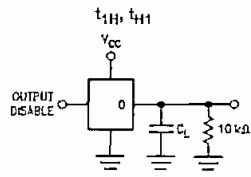
Data Bus Interfacing Element



Simple, Latching, Octal, LED Indicator Driver with Blanking for Use as Data Display, Bus Monitor,  $\mu$ P Front Panel Display, Etc.

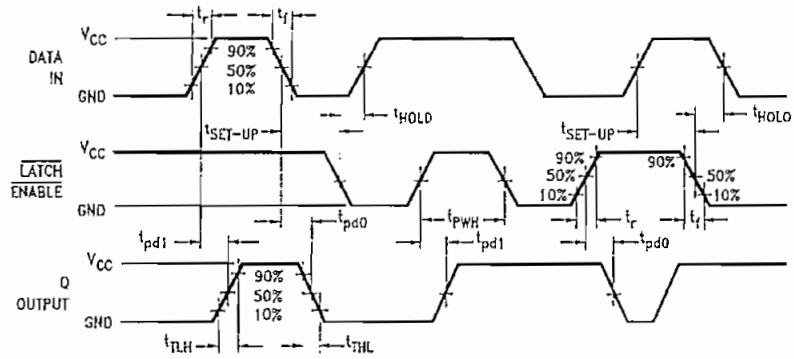


### 3-STATE Test Circuits and Switching Time Waveforms



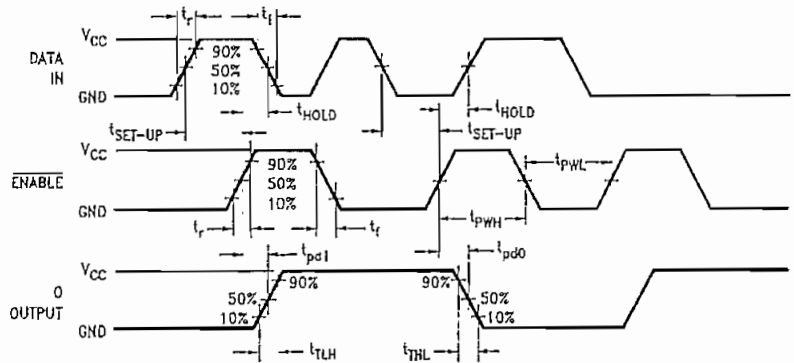
# Switching Time Waveforms

MM74C373



Output Disable = GND

MM74C374



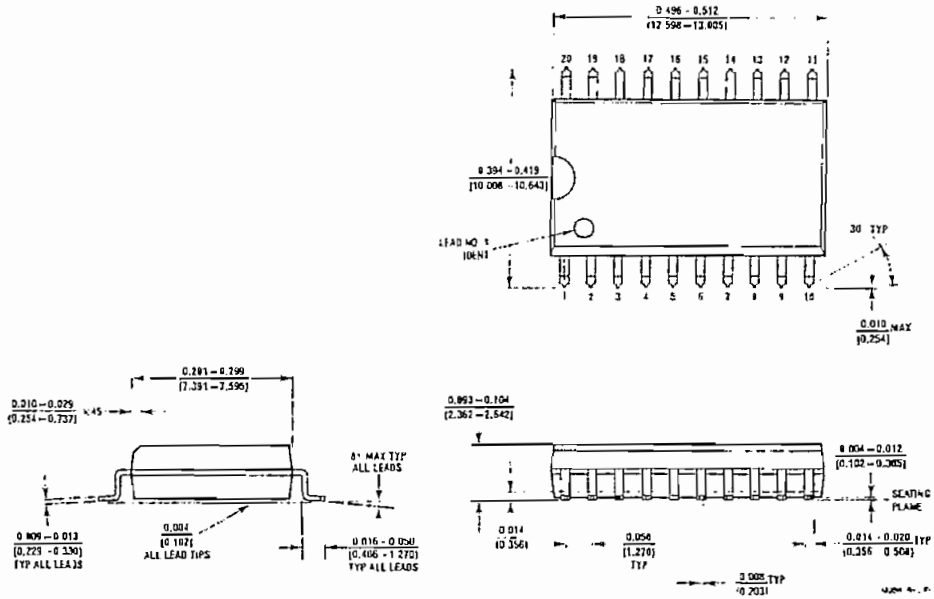
Output Disable = GND

MM74C373 • MM74C374



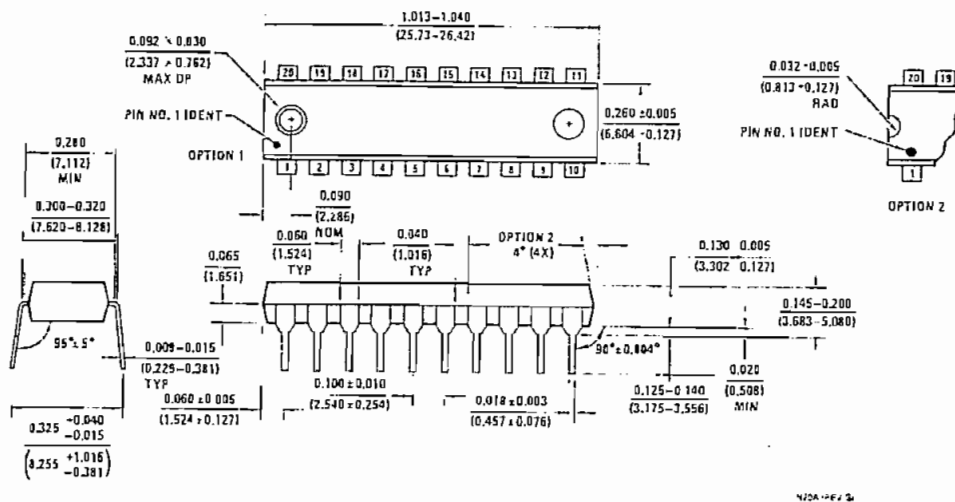
MM74C373 • MM74C374

Physical Dimensions Inches (millimeters) unless otherwise noted



20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide  
Package Number M20B

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide Package Number N20A

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

www.fairchildsemi.com

Fairchild does not assume any responsibility for use of any product described, nor does it warrant the accuracy of the information provided.

## MM74C922 • MM74C923 16-Key Encoder • 20-Key Encoder

### General Description

The MM74C922 and MM74C923 CMOS key encoders provide all the necessary logic to fully encode an array of SPST switches. The keyboard scan can be implemented by either an external clock or external capacitor. These encoders also have on-chip pull-up devices which permit switches with up to 50 kΩ on resistance to be used. No diodes in the switch array are needed to eliminate ghost switches. The internal debounce circuit needs only a single external capacitor and can be defeated by omitting the capacitor. A Data Available output goes to a high level when a valid keyboard entry has been made. The Data Available output returns to a low level when the entered key is released, even if another key is depressed. The Data Available will return high to indicate acceptance of the new key after a normal debounce period; this two-key roll-over is provided between any two switches.

An internal register remembers the last key pressed even after the key is released. The 3-STATE outputs provide for easy expansion and bus operation and are LPTTL compatible.

### Features

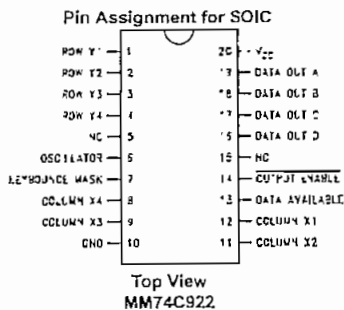
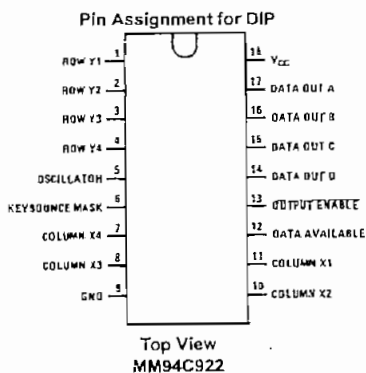
- 50 kΩ maximum switch on resistance
- On or off chip clock
- On-chip row pull-up devices
- 2 key roll-over
- Keybounce elimination with single capacitor
- Last key register at outputs
- 3-STATE output LPTTL compatible
- Wide supply range: 3V to 15V
- Low power consumption

### Ordering Code:

Order Number	Package Number	Package Description
MM74C922N	N18A	18-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
MM74C922WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C923WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C923N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide

Device also available in Tape and Reel. Specify by appending suffix letter "X" to the ordering code.

### Connection Diagrams



Absolute Maximum Ratings (Note 2)		Operating $V_{CC}$ Range	3V to 15V
Voltage at Any Pin	$V_{CC} - 0.3V$ to $V_{CC} + 0.3V$	$V_{CC}$	18V
Operating Temperature Range		Lead Temperature	
MM74C922, MM74C923	-40°C to +85°C	(Soldering, 10 seconds)	260°C
Storage Temperature Range	-65°C to +150°C		
Power Dissipation ( $P_D$ )			
Dual-In-Line	700 mW		
Small Outline	500 mW		

Note 2: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

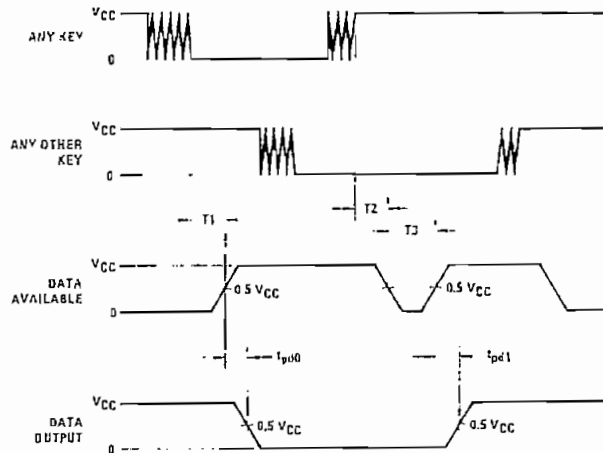
## DC Electrical Characteristics

Min/Max limits apply across temperature range unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
<b>CMOS TO CMOS</b>						
$V_{I+}$	Positive-Going Threshold Voltage at Osc and KBM Inputs	$V_{CC} = 5V, I_{IH} \geq 0.7 \text{ mA}$ $V_{CC} = 10V, I_{IH} \geq 1.4 \text{ mA}$ $V_{CC} = 15V, I_{IH} \geq 2.1 \text{ mA}$	3.0 6.0 9.0	3.6 6.8 10	4.3 8.6 12.9	V
$V_{I-}$	Negative-Going Threshold Voltage at Osc and KBM Inputs	$V_{CC} = 5V, I_{IN} \geq 0.7 \text{ mA}$ $V_{CC} = 10V, I_{IN} \geq 1.4 \text{ mA}$ $V_{CC} = 15V, I_{IN} \geq 2.1 \text{ mA}$	0.7 1.4 2.1	1.4 3.2 5	2.0 4.0 6.0	V
$V_{IN(1)}$	Logical "1" Input Voltage, Except Osc and KBM Inputs	$V_{CC} = 5V$ $V_{CC} = 10V$ $V_{CC} = 15V$	3.5 8.0 12.5	4.5 9 13.5		V
$V_{IN(0)}$	Logical "0" Input Voltage, Except Osc and KBM Inputs	$V_{CC} = 5V$ $V_{CC} = 10V$ $V_{CC} = 15V$		0.5 1 1.5	1.5 2 2.5	V
$I_{p}$	Row Pull-Up Current at Y1, Y2, Y3, Y4 and Y5 Inputs	$V_{CC} = 5V, V_{IN} = 0.1 V_{CC}$ $V_{CC} = 10V$ $V_{CC} = 15V$		2 -10 22	5 20 -45	$\mu\text{A}$
$V_{OUT(1)}$	Logical "1" Output Voltage	$V_{CC} = 5V, I_O = -10 \mu\text{A}$ $V_{CC} = 10V, I_O = -10 \mu\text{A}$ $V_{CC} = 15V, I_O = -10 \mu\text{A}$	4.5 9 13.5			V
$V_{OUT(0)}$	Logical "0" Output Voltage	$V_{CC} = 5V, I_O = 10 \mu\text{A}$ $V_{CC} = 10V, I_O = 10 \mu\text{A}$ $V_{CC} = 15V, I_O = 10 \mu\text{A}$			0.5 1 1.5	V
$R_{ON}$	Column "ON" Resistance at X1, X2, X3 and X4 Outputs	$V_{CC} = 5V, V_O = 0.5V$ $V_{CC} = 10V, V_O = 1V$ $V_{CC} = 15V, V_O = 1.5V$		500 300 200	1400 750 500	$\Omega$
$I_{CC}$	Supply Current, Osc at 0V, (one Y low)	$V_{CC} = 5V$ $V_{CC} = 10V$ $V_{CC} = 15V$		0.55 1.1 1.7	1.1 1.9 2.6	mA
$I_{IN(1)}$	Logical "1" Input Current at Output Enable	$V_{CC} = 15V, V_{IN} = 15V$		0.005	1.0	$\mu\text{A}$
$I_{IN(0)}$	Logical "0" Input Current at Output Enable	$V_{CC} = 15V, V_{IN} = 0V$	1.0	0.005		$\mu\text{A}$
<b>CMOS/PTTL INTERFACE</b>						
$V_{IN(1)}$	Except Osc and KBM Inputs	$V_{CC} = 4.75V$	$V_{CC}$	1.5		V
$V_{IN(0)}$	Except Osc and KBM Inputs	$V_{CC} = 4.75V$			0.8	V
$V_{OUT(1)}$	Logical "1" Output Voltage	$I_O = 360 \mu\text{A}$ $V_{CC} = 4.75V$ $I_O = -360 \mu\text{A}$	2.4			V
$V_{OUT(0)}$	Logical "0" Output Voltage	$I_O = 360 \mu\text{A}$ $V_{CC} = 4.75V$ $I_O = -360 \mu\text{A}$			0.4	V

DC Electrical Characteristics (Continued)						
Symbol	Parameter	Conditions	Min	Typ	Max	Units
OUTPUT DRIVE (See Family Characteristics Data Sheet) (Short Circuit Current)						
I <sub>SOURCE</sub>	Output Source Current (P-Channel)	V <sub>CC</sub> = 5V, V <sub>OUT</sub> = 0V. T <sub>A</sub> = 25°C	1.75	-3.3		mA
I <sub>SOURCE</sub>	Output Source Current (P-Channel)	V <sub>CC</sub> = 10V, V <sub>OUT</sub> = 0V. T <sub>A</sub> = 25°C	8	15		mA
I <sub>SINK</sub>	Output Sink Current (N-Channel)	V <sub>CC</sub> = 5V, V <sub>OUT</sub> = V <sub>CC</sub> . T <sub>A</sub> = 25°C	1.75	3.6		mA
I <sub>SINK</sub>	Output Sink Current (N-Channel)	V <sub>CC</sub> = 10V, V <sub>OUT</sub> = V <sub>CC</sub> . T <sub>A</sub> = 25°C	8	16		mA
AC Electrical Characteristics (Note 3)						
T <sub>A</sub> = 25°C, C <sub>L</sub> = 50 pF, unless otherwise noted						
Symbol	Parameter	Conditions	Min	Typ	Max	Units
t <sub>pd0</sub> , t <sub>pd1</sub>	Propagation Delay Time to Logical "0" or Logical "1" from D.A.	C <sub>L</sub> = 50 pF (Figure 1) V <sub>CC</sub> = 5V V <sub>CC</sub> = 10V V <sub>CC</sub> = 15V		80 35 25	150 80 60	ns ns ns
t <sub>0H</sub> , t <sub>1H</sub>	Propagation Delay Time from Logical "0" or Logical "1" into High Impedance State	R <sub>L</sub> = 10k, C <sub>L</sub> = 10 pF (Figure 2) V <sub>CC</sub> = 5V, R <sub>L</sub> = 10k V <sub>CC</sub> = 10V, C <sub>L</sub> = 10 pF V <sub>CC</sub> = 15V		80 65 50	200 150 110	ns ns ns
t <sub>H0</sub> , t <sub>H1</sub>	Propagation Delay Time from High Impedance State to a Logical "0" or Logical "1"	R <sub>L</sub> = 10k, C <sub>L</sub> = 50 pF (Figure 2) V <sub>CC</sub> = 5V, R <sub>L</sub> = 10k V <sub>CC</sub> = 10V, C <sub>L</sub> = 50 pF V <sub>CC</sub> = 15V		100 55 40	250 125 90	ns ns ns
C <sub>IN</sub>	Input Capacitance	Any Input (Note 4)		5	7.5	pF
C <sub>OUT</sub>	3-STATE Output Capacitance	Any Output (Note 4)		10		pF
Note 3: AC Parameters are guaranteed by DC correlated testing.						
Note 4: Capacitance is guaranteed by periodic testing.						

Switching Time Waveforms



$T_1 = T_2 = RC$ ,  $T_3 = 0.7 RC$ , where  $R = 10k$  and  $C$  is external capacitor at KBM input.

FIGURE 1.

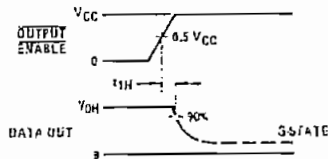
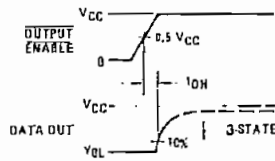
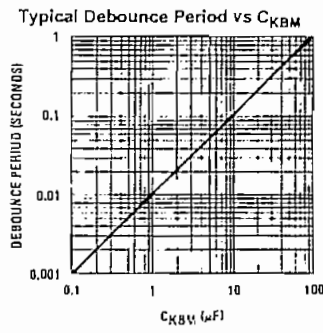
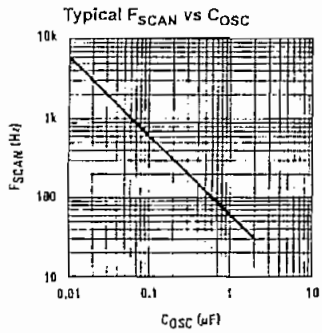
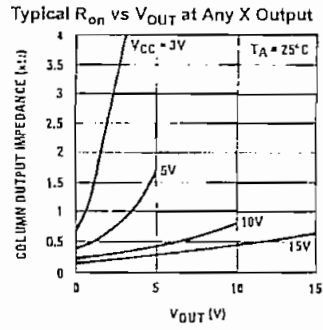
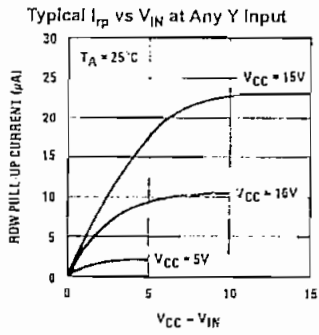


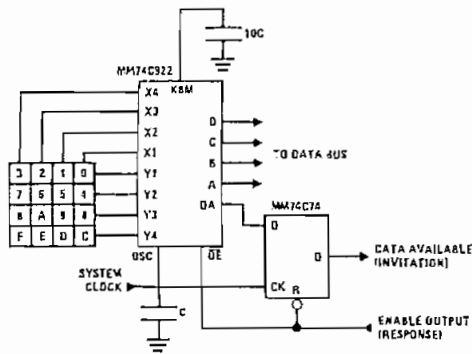
FIGURE 2.

## Typical Performance Characteristics



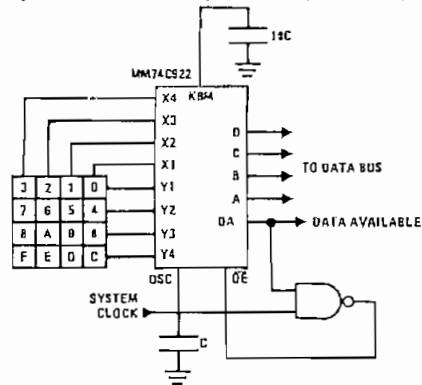
## Typical Applications

Synchronous Handshake (MM74C922)



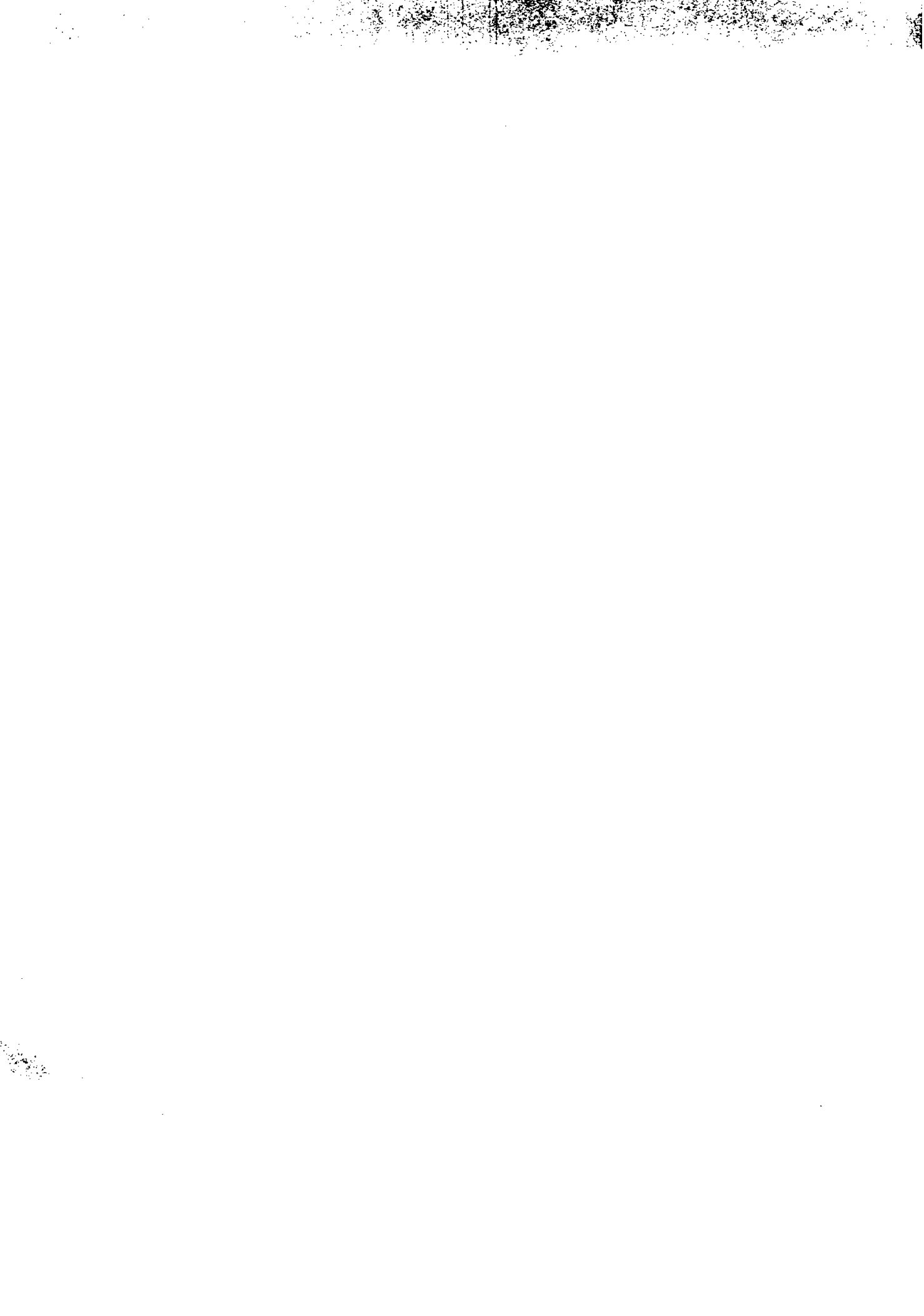
The keyboard may be synchronously scanned by omitting the capacitor at osc. and driving osc. directly if the system clock rate is lower than 10 kHz.

Synchronous Data Entry Onto Bus (MM74C922)



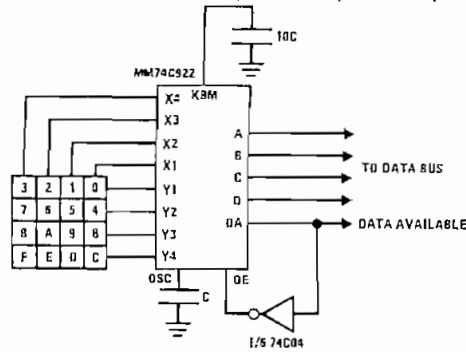
Outputs are enabled when valid entry is made and go into 3-STATE when key is released.

The keyboard may be synchronously scanned by omitting the capacitor at osc. and driving osc. directly if the system clock rate is lower than 10 kHz.



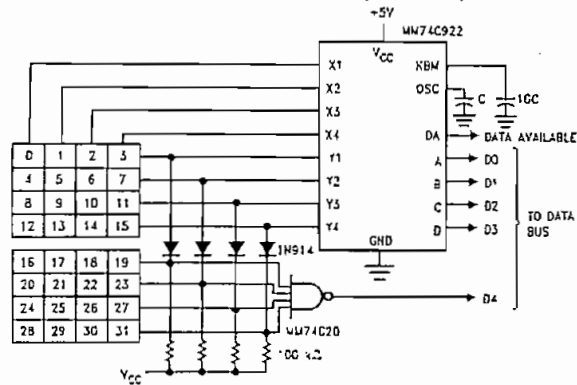


Asynchronous Data Entry Onto Bus (MM74C922)



Outputs are in 3-STATE until keys pressed, then data is placed on bus. When key is released, outputs return to 3-STATE.

Expansion to 32 Key Encoder (MM74C922)



## Theory of Operation

The MM74C922/MM74C923 Keyboard Encoders implement all the logic necessary to interface a 16 or 20 SPST key switch matrix to a digital system. The encoder will convert a key switch closure to a 4-bit (MM74C922) or 5-bit (MM74C923) bit nibble. The designer can control both the keyboard scan rate and the key debounce period by altering the oscillator capacitor,  $C_{OSC}$ , and the key bounce mask capacitor,  $C_{MSK}$ . Thus, the MM74C922/MM74C923's performance can be optimized for many keyboards.

The keyboard encoders connect to a switch matrix that is 4 rows by 4 columns (MM74C922) or 5 rows by 4 columns (MM74C923). When no keys are depressed, the row inputs are pulled high by internal pull-ups and the column outputs sequentially output a logic "0". These outputs are open drain and are therefore low for 25% of the time and otherwise off. The column scan rate is controlled by the oscillator input, which consists of a Schmitt trigger oscillator, a 2-bit counter, and a 2-4-bit decoder.

When a key is depressed, key 0, for example, nothing will happen when the X1 input is off, since Y1 will remain high. When the X1 column is scanned, X1 goes low and Y1 will go low. This disables the counter and keeps X1 low. Y1

going low also initiates the key bounce circuit limiting and locks out the other Y inputs. The key code to be output is a combination of the frozen counter value and the decoded Y inputs. Once the key bounce circuit times out, the data is latched, and the Data Available (DAV) output goes high.

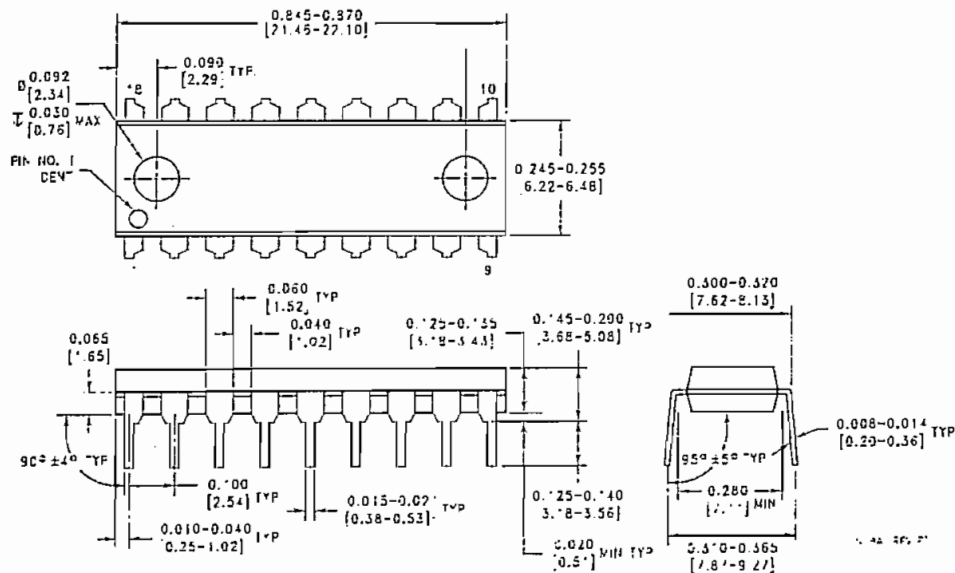
If, during the key closure the switch bounces, Y1 input will go high again, restarting the scan and resetting the key bounce circuitry. The key may bounce several times, but as soon as the switch stays low for a debounce period, the closure is assumed valid and the data is latched.

A key may also bounce when it is released. To ensure that the encoder does not recognize this bounce as another key closure, the debounce circuit must time out before another closure is recognized.

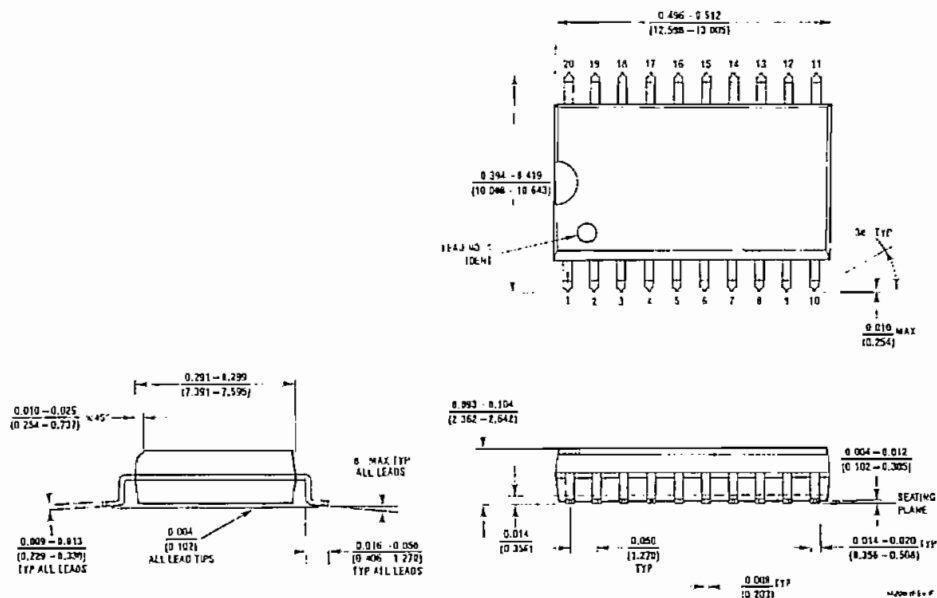
The two-key roll-over feature can be illustrated by assuming a key is depressed, and then a second key is depressed. Since all scanning has stopped, and all other Y inputs are disabled, the second key is not recognized until the first key is lifted and the key bounce circuitry has reset.

The output latches feed 3-STATE, which is enabled when the Output Enable ( $\overline{OE}$ ) input is taken low.

Physical Dimensions inches (millimeters) unless otherwise noted

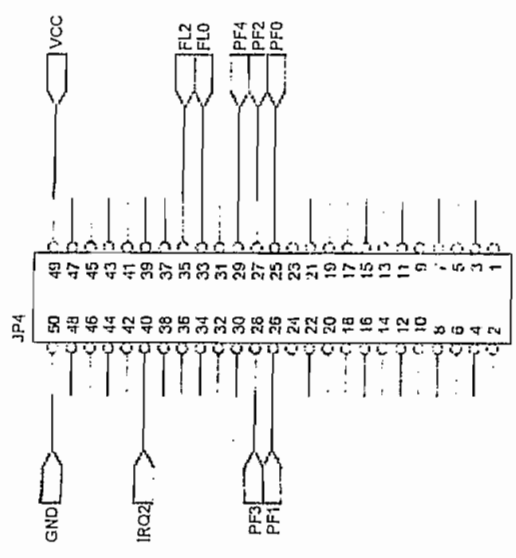
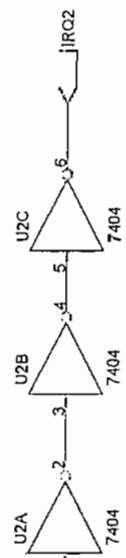
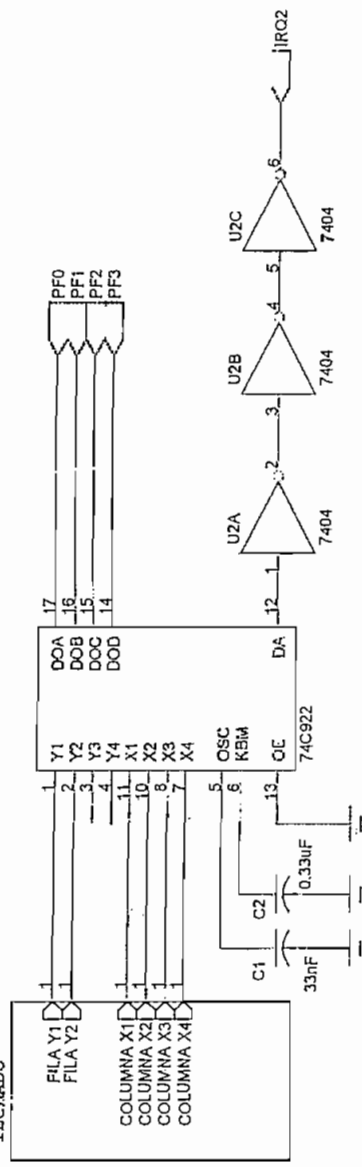


18-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide  
Package Number N18A

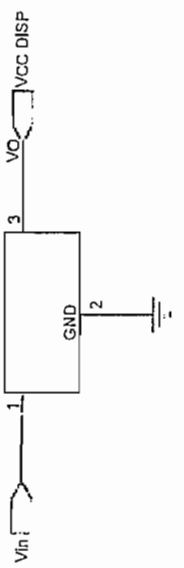
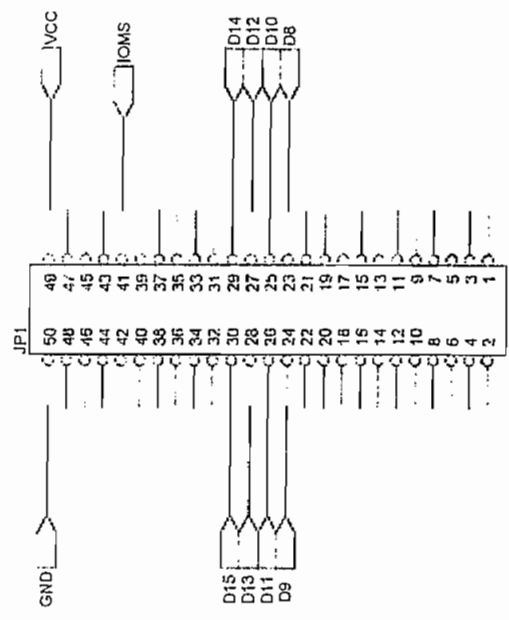
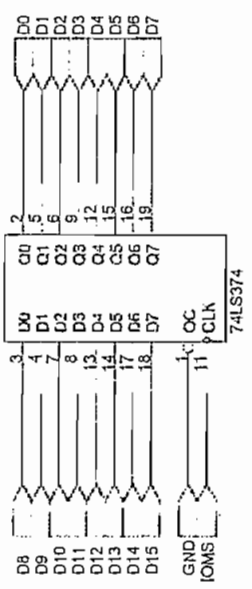


20-Lead Plastic Small Outline I.C. Package (M)  
Package Number M20B

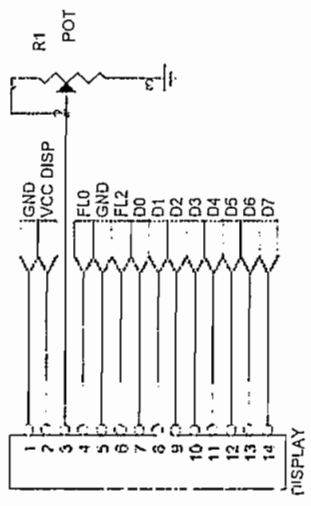
MATRIZ DE  
RECIBIDO



P3 EN EZ-LAB



VI



FILTROS FIR EN TIEMPO REAL

CIRCUITO DE CONTROL PARA PEDALERA

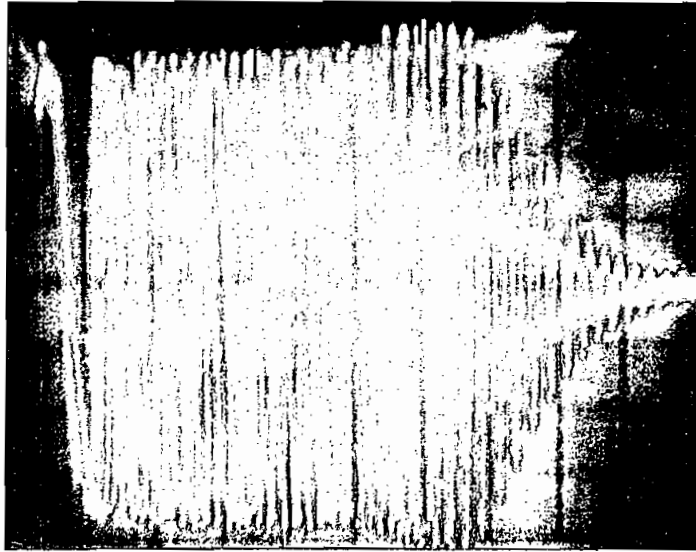
Size A4  
Document Number

Rev 1

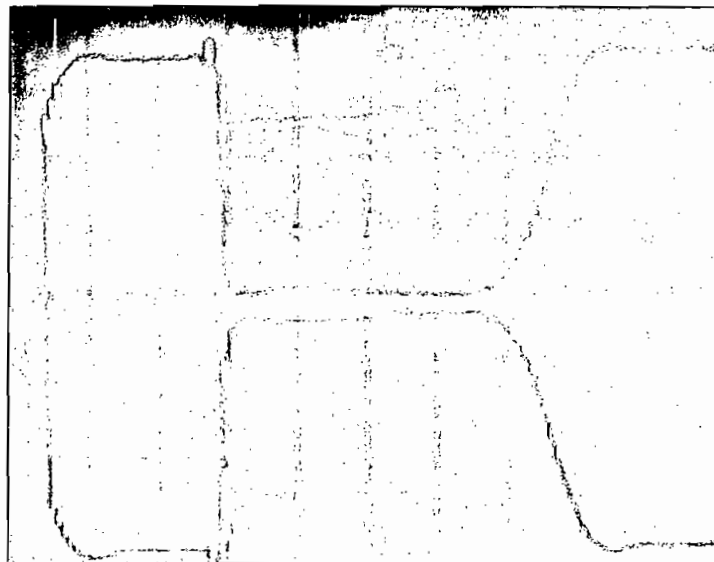
## FOTOS Y VISUALIZACIÓN EN EL OSCILOSCOPIO

---

Las figuras a continuación, indican la respuesta de un filtro a un barrido total de frecuencia hasta 4 KHz.



a) Filtro Pasa Bajo



b) Filtro Pasa alto