

42
2 es.



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

CONTROL ACTIVO DE RUIDO ACÚSTICO



T E S I S

QUE PARA OBTENER EL TÍTULO DE
INGENIERO MECÁNICO ELECTRICISTA
ÁREA ELÉCTRICA ELECTRÓNICA
P R E S E N T A N
CONCHA DIMAS, AHMED
DE LA TORRE ALATRISTE ERYX D.

MÉXICO D.F. 1998

DIRECTOR DE TESIS: BOHUMIL PSENICKA

TESIS CON
FALLA DE ORIGEN

175821



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mis padres, por su invaluable apoyo, pero sobre todo, por su amor.

Agradecimientos.

Mein lieber Freund, diese ist für dich!!!
Ich danke dir deine Freundschaft!!!

A mis hermanas Aline y Lizbeth
por soportarme todos estos años.

A Isabel, Vero, Oscar, Alvaro y Horacio.

A Catalina Vargas Ramos por ser
tan linda y apoyarme siempre!

Quiero expresar mi más sincero agradecimiento a mis padres *Jaime De La Torre e Hilda Alatríste*, quienes en todo momento me han brindado su amor, atención y apoyo para realizar todas las metas que me propongo, y ésta no fue la excepción.
Gracias.

A mi familia, por haberme impulsado y guiado a través de sus enseñanzas para hacer de mí un hombre responsable y con objetivos claros.

A mis Estimais *Gabi, el buen Dolph, Vladi, Horacio, Miguel Arturo y Chris* con quienes inicié el andar de mis pasos por esta vida.

Al profesor **Bohumil Psenicka** por su
confianza y amabilidad hacia nosotros.

A nuestros amigos de la Facultad:

Carlos, Yaaxni, Rafael, Jaime Cerón,
Alejandro Mancilla, Kikito, Lenz,
Ricardo, Jaime Lozano, Cisneros,
Laura, Zamantha, Mariza, Gaby,
Esther, Alicia, Carmen y Nidia.



Eryx D. De la Torre A.



Ahmed Concha Dimas

Contenido

1	Antecedentes	1
1.1	Filtrado Digital.	1
1.2	Señales discretas.	2
1.3	Sistemas lineales e invariantes en el tiempo.	3
1.4	Convolución.	4
1.5	Respuesta en frecuencia.	5
1.6	Filtros con la respuesta finita al impulso.	7
1.7	Valor esperado (E).	8
2	Filtros adaptativos	9
2.1	Introducción	9
2.2	Estructuras	9
2.2.1	Estructura general.	9
2.2.2	Identificación del sistema	10
2.2.3	Cancelación de ruido a la salida	11
2.2.4	Cancelación de ruido aditivo en la entrada	12
2.3	Combinador lineal adaptativo.	12
2.3.1	Múltiples entradas.	12
2.3.2	Una sola entrada.	14
2.3.3	Función de desempeño	14
2.3.4	Buscando el mínimo.	16
3	Control Activo de Ruido	20
3.1	Generalidades	20
3.2	Aplicaciones generales	24
3.2.1	ANC Aire-Acústico	25
3.2.2	ANC Hidroacústico	27
3.2.3	ANC Vibratorio	27
3.3	Sistemas ANC	27
3.3.1	Sistemas ANC feedforward de banda ancha	28
3.3.2	Sistemas ANC de banda estrecha feedforward	30
3.3.3	Sistemas ANC Feedback	31

3.4	Sistemas ANC Multicanal	31
3.5	ANC Feedforward de banda ancha	34
3.6	Efectos de la trayectoria secundaria.	36
3.7	El algoritmo FXLMS	39
3.7.1	Obtención del algoritmo FXLMS	40
3.7.2	Técnicas de Modelado Off-Line	41
3.8	Sistemas ANC feedback monocanal	44
3.8.1	Algoritmo y estructura.	45
3.9	Evaluación del desempeño	47
4	El TMS320C50	49
4.1	Surgimiento del TMS320	49
4.2	Quinta Generación, el DSP TMS320C5x	51
4.3	Arquitectura	54
4.3.1	Estructura de los Buses	54
4.3.2	Memoria	57
4.3.3	Unidad central de procesos (CPU)	58
4.3.4	CALU	58
4.3.5	PLU	62
4.3.6	Registros mapeados en memoria	65
4.3.7	Controlador de Programa	67
4.3.8	Modos de direccionamiento	78
5	Implementación de un sistema ANC monocanal mediante el C50 DSK.	102
5.1	Descripción del Controlador	102
5.1.1	Número de Derivaciones para \hat{H} y W	105
5.2	Consideraciones para el Hardware	105
5.2.1	Distancia entre la bocina y el micrófono	105
5.2.2	Etapas de Entrada	105
5.2.3	Niveles de la Señal	106
5.2.4	Etapas de Salida	106
5.3	Implementación en el DSK	107
5.3.1	Inicialización del C50	107
5.3.2	Inicialización del AIC	108
5.3.3	Habilitación de Interrupciones	109
5.3.4	Espera de interrupciones	109
5.3.5	Cálculo de la compensación de la entrada del AIC	109
5.3.6	Identificación de H	109
5.3.7	Control	112
6	Conclusiones.	114
A	Código en Ensamblador del Sistema ANC.	118

Lista de Figuras

1.1	Ejemplo de una señal discreta en el tiempo	3
1.2	El impulso unitario y el impulso unitario trasladado	3
1.3	Estructura de un filtro FIR y sus retrasos.	7
2.1	Estructura adaptativa en general.	10
2.2	Identificación de un sistema.	11
2.3	Cancelación de ruido a la salida.	11
2.4	Cancelación de ruido aditivo en la entrada	12
2.5	Combinador lineal con múltiples entradas.	13
2.6	Combinador lineal adaptativo.	13
2.7	Curva de desempeño de un solo peso.	16
2.8	Busqueda del mínimo para un solo peso.	17
3.1	Concepto físico del ANC.	22
3.2	Sistema ANC feedforward de banda ancha de un solo canal en un ducto.	29
3.3	Sistema ANC feedforward de banda estrecha de un solo canal en un ducto.	30
3.4	Sistema ANC feedback de un solo canal en un ducto.	31
3.5	Sistema ANC acústico multicanal de 3-D.	32
3.6	Sistema ANC feedforward monocanal en un ducto.	34
3.7	Identificación de un sistema desde el punto de vista ANC.	35
3.8	Diagrama a bloques de un sistema ANC.	37
3.9	Diagrama a bloques simplificado de un sistema ANC.	39
3.10	Diagrama a bloques de un sistema ANC utilizando el algoritmo FXLMS.	41
3.11	Arreglo experimental para modelar la trayectoria secundaria fuera de línea.	42
3.12	Sistema ANC mono canal retroalimentado.	44
3.13	Diagrama de bloques de un sistema ANC monocanal retroalimentado.	44
3.14	Uso de la señal de referencia sintetizada a partir de las señales disponibles $y(n)$ y $e(n)$	45

3.15	Sistema ANC retroalimentado haciendo uso del algoritmo FXLMS.	46
4.1	Generaciones de DSPs.	50
4.2	Tipos de C5x.	52
4.3	Arquitectura del C50.	55
4.4	Buses del C50.	56
4.5	Memoria del C50.	58
4.6	CPU del C50.	59
4.7	CALU del C50.	61
4.8	PLU del C50.	63
4.9	Diagrama funcional a bloques de la ARAU.	64
4.10	Diagrama funcional a bloques del PC.	69
4.11	El CBCR.	70
4.12	El PMST.	70
4.13	El registro STO	72
4.14	El registro ST1	72
4.15	Operación del Pipeline	77
4.16	Diagrama del direccionamiento directo y un ejemplo del mismo.	80
4.17	Ejemplo del direccionamiento indirecto.	83
4.18	Direccionamiento indirecto indexado.	85
4.19	Direccionamiento inmediato corto.	86
4.20	Direccionamiento inmediato largo con 2 operandos.	87
4.21	Direccionamiento de registro dedicado usando el BMAR y el DBMR.	88
4.22	Direccionamiento de Registros mapeados en memoria, y en el modo Directo de direccionamiento.	90
4.23	Registro de banderas para las interrupciones	92
4.24	Registro de enmascarado de interrupciones.	93
4.25	El registro PDWSR.	96
4.26	El registro IOWSR.	97
4.27	CWSR	98
4.28	Diagrama del temporizador.	99
4.29	El registro TCR	99
5.1	Cancelador Activo Adaptivo de ruido de un solo canal.	103
5.2	Controlador Retroalimentado utilizando una Aproximación Feed-forward	104
5.3	Proceso de identificación de la Planta	104
5.4	Etapas de Entrada.	106
5.5	Sección principal	107
5.6	Sección de interrupción	108
5.7	Señal regresada por el AIC con la entrada conectada a tierra.	110
5.8	Pasos de la fase de identificación. Ejecutada en ITER iteraciones	111

5.9 Pasos de la Fase de Control, Efectuada después de la identificación y hasta la reinicialización del DSP.	113
B.1 Atenuación de un tono de 100 Hz.	129
B.2 Atenuación de un tono de 150 Hz.	130
B.3 Atenuación de un tono de 200 Hz.	131
B.4 Atenuación de un tono de 250 Hz.	132
B.5 Atenuación de un tono de 300 Hz.	133
B.6 Diámetro de la zona de silencio.	134
B.7 El sistema funcionando en el dominio eléctrico.	135
B.8 Sistema ANC monocanal.	136
B.9 Trabajo en el laboratorio.	137
C.1 Diagrama de Bode.	138
C.2 Diagrama Polar.	139

Lista de Tablas

3.1	Jerarquía de Análisis para el desempeño ANC.	48
4.1	Mapa de direcciones de la página 0. Registros del CPU	66
4.2	El registro CBCR.	70
4.3	El registro PMST.	71
4.4	Continuación del registro PMST.	72
4.5	El registro ST0.	73
4.6	Registro ST1.	74
4.7	Registro ST1. Continuación.	75
4.8	Modos de Direccionamiento.	79
4.9	Más opciones del direccionamiento indirecto.	84
4.10	Instrucciones que soportan direccionamiento inmediato	85
4.11	Localidades y prioridades del vector de interrupciones.	91
4.12	Estado de los registros del CPU al ejecutar un reset.	93
4.13	Estado de los registros de los circuitos periféricos después de un RESET.	94
4.14	Continuación de la tabla anterior.	95
4.15	Registro de control del temporizador	100

Síntesis

El control activo de ruido o ANC (Active Noise control) es una técnica de control basada en la interferencia destructiva. Es decir, se genera una onda de la misma magnitud que el ruido a cancelar pero defasada 180 grados. De manera que la superposición de las dos ondas en el campo en el que se propagan da lugar a la cancelación del ruido indeseable.

El **objetivo** de este trabajo de tesis es la implementación de un cancelador de ruido acústico monocanal haciendo uso del DSP TMS320C50 y de una configuración de control retroalimentado.

El cancelador deberá generar una zona de silencio alrededor del micrófono sensor y deberá ser capaz de manejar tonos que van desde los 100Hz hasta los 500Hz.

Capítulo 1

Antecedentes

1.1 Filtrado Digital.

El filtrado digital es una de las operaciones más útiles en el procesamiento de señales. Como el filtrado analógico, el filtro digital tiene características selectivas de frecuencia.

Mientras los filtros analógicos operan en señales continuas y son típicamente construidos con resistencias, capacitores, e inductores, los filtros digitales operan en muestras o secuencias discretas y son implementados con lógica digital o microprocesadores enfocados al procesamiento digital de señales (DSPs) tales como el TMS320C50.

Las ventajas de los filtros digitales sobre los filtros analógicos incluyen alta confiabilidad, exactitud, y flexibilidad, con mínima sensibilidad a la temperatura y al paso del tiempo.

Los filtros digitales pueden ajustarse a características severas de fase y magnitud. Usando una implementación con un DSP, los parámetros del filtro son fácilmente modificables para cambiar características tales como el ancho de banda, frecuencia de resonancia, o el tipo de filtro.

El filtrado digital con un DSP involucra el uso de un convertidor A/D para muestrear la señal analógica y obtener una secuencia de muestras de entrada, que son procesadas y después convertidas a una señal analógica de salida usando un convertidor D/A.

El diseño de tales filtros digitales consiste en:

1. Aproximación. Obtener una función de transferencia o su correspondiente función en el tiempo para satisfacer las especificaciones deseadas.
2. Realización. Conversión de la función de transferencia en estructuras de filtrado tales como una tabla de coeficientes y variables de retraso.

3. Consideración de errores aritméticos. La consideración de los efectos de la aproximación aritmética cuando se usa un hardware tal como el TMS320C50. Es decir la exactitud con la que los coeficientes del filtro pueden ser representados está limitada por la longitud finita del procesador.
4. Implementación. Construir el filtro usando software y hardware. Con la tecnología VLSI, los DSPs están ahora disponibles para implementar filtros en tiempo real, permitiendo que el filtro opere con señales en tiempo real.

En un curso básico de circuitos, hemos visto como una ecuación diferencial puede representar un filtro, y como la transformada de Laplace nos permite resolver ecuaciones diferenciales asociadas con sistemas continuos en el tiempo.

Similarmente, la transformada Z nos permite resolver ecuaciones en diferencias, asociadas con sistemas discretos en el tiempo, representando filtros digitales.

Los sistemas que construiremos son discretos, lineales e invariantes en el tiempo. Consecuentemente, existe una relación con la convolución que es usada en la realización de filtros de respuesta finita al impulso (FIR).

Dado que la convolución está estrechamente relacionada al filtro FIR, será examinada junto con el diseño de filtros FIR.

Existen dos clases de filtros digitales: (1) Filtros con respuesta finita al impulso (FIR), y (2) Filtros con respuesta infinita al impulso (IIR). En esta introducción sólo consideraremos a los filtros FIR.

1.2 Señales discretas.

Una señal discreta es una secuencia de valores que pueden ser representados como

$$\{x(n)\} = \{x(1), x(2), x(3), \dots\} \quad (1.1)$$

donde n es el índice en el tiempo, y cada muestra es tomada por separado en un intervalo fijo de tiempo.

La secuencia $\{x(n)\}$ se muestra gráficamente en la Figura 1.1.

Una secuencia importante es la unidad de muestreo o impulso unitario, que se define como:

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad (1.2)$$

Esta función es un impulso localizado en el origen. Un impulso trasladado en el tiempo se escribe como:

$$\delta(n - m) = \begin{cases} 1 & n = m \\ 0 & n \neq m \end{cases} \quad (1.3)$$

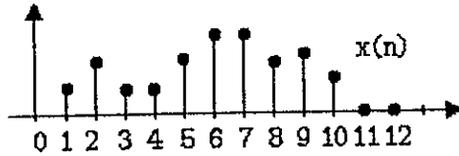


Figura 1.1: Ejemplo de una señal discreta en el tiempo

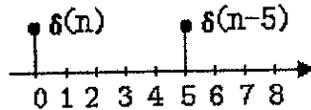


Figura 1.2: El impulso unitario y el impulso unitario trasladado

Esta muestra está trasladada m unidades a la derecha, u ocurre m muestras después del impulso unitario en el origen. Ejemplos de ambos se muestran en la Figura 1.2. Una secuencia $x(n)$ puede ser escrita en términos del impulso unitario:

$$x(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n - m) \quad (1.4)$$

1.3 Sistemas lineales e invariantes en el tiempo.

Los sistemas digitales con los que trataremos son lineales e invariantes en el tiempo. Dos importantes propiedades asociadas con estos sistemas, son la superposición y la invariación en el traslado en el tiempo, que a continuación se describen.

Suponga que una entrada $x(n)$ nos lleva a una respuesta $y(n)$, escrita como:

$$x(n) \rightarrow y(n) \quad (1.5)$$

Si multiplicamos la entrada por una constante, la salida es multiplicada por la misma constante como se muestra:

$$ax(n) \rightarrow ay(n) \quad (1.6)$$

Aún más, si:

$$ax_1(n) \rightarrow ay_1(n) \quad (1.7)$$

y

$$bx_2(n) \rightarrow by_2(n)$$

entonces

$$ax_1(n) + bx_2(n) \rightarrow ay_1(n) + by_2(n) \quad (1.8)$$

Esta última describe la propiedad de superposición, que establece que la respuesta total a un sistema es igual a la suma de las respuestas individuales para cada entrada.

La segunda operación, que es la *invariancia en el traslado*, se describe como sigue:

Si

$$x(n) \rightarrow y(n)$$

entonces

$$x(n - m) \rightarrow y(n - m) \quad (1.9)$$

Retrasando la entrada m muestras, se retrasa la salida m muestras.

1.4 Convolución.

Usando las dos propiedades arriba descritas, superposición e invariancia en el traslado, la suma de convolución puede ser fácilmente obtenida. Suponga que introducimos un impulso unitario en la entrada del sistema, la respuesta $h(n)$, que es llamada la respuesta al impulso unitario, resulta:

$$\delta(n) \rightarrow h(n) \quad (1.10)$$

Si esta entrada está retardada, la salida será retardada similarmente:

$$\delta(n - m) \rightarrow h(n - m) \quad (1.11)$$

Si el impulso unitario está multiplicado por una constante, la respuesta será multiplicada por una constante, para dar como resultado:

$$x(m)\delta(n - m) \rightarrow x(m)h(n - m) \quad (1.12)$$

En general una señal de entrada puede ser escrita como una suma de respuestas al impulso unitario ponderadas como se muestra en la ecuación 1.4 y repetida aquí por conveniencia:

$$x(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n - m) \quad (1.13)$$

Por superposición podemos agregar la respuesta individual de cada impulso unitario para obtener la respuesta colectiva:

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) \quad (1.14)$$

Si el sistema es causal, la respuesta $h(n)$ no puede comenzar antes de que el impulso unitario sea aplicado. De esta forma, m no puede exceder a n en la ecuación (1.14). La suma de convolución puede ser ajustada para reflejar esto:

$$y(n) = \sum_{m=-\infty}^n x(m)h(n-m) \quad (1.15)$$

Con un cambio de variable $k = n - m$, la sumatoria se convierte en:

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \quad (1.16)$$

En el diseño de los filtros FIR usamos solamente un número finito de términos para representar $h(n)$, y de esta forma la convolución puede ser escrita como

$$y(n) = \sum_{k=0}^N h(k)x(n-k) = \sum_{m=0}^N h(m)x(n-m) \quad (1.17)$$

Si $x(n) = \delta(n)$, el impulso unitario, en la ecuación (1.17), la respuesta es:

$$y(n) = \sum_{m=0}^N h(m)\delta(n-m) \Big|_{m=n} = h(n) \quad (1.18)$$

La respuesta al impulso unitario es la respuesta al impulso, $h(n)$, que es lo que esperabamos.

1.5 Respuesta en frecuencia.

Dado que la respuesta de los filtros generalmente se especifica en términos de la respuesta en frecuencia, encontraremos una expresión para la respuesta en frecuencia de un sistema.

$$X(N) = X \exp(j\omega) \quad (1.19)$$

La respuesta a esta entrada es

$$y(n) = \sum_{m=0}^{\infty} h(m)X \exp[j\omega(n-m)] \quad (1.20)$$

Dado que la suma es respecto a m , $\exp(j\omega n)$ puede sacarse de la sumatoria, dando:

$$y(n) = \left[\sum_{m=0}^{\infty} h(m) \exp(-j\omega m) \right] X \exp(j\omega n) \quad (1.21)$$

Así la respuesta a una senoide es una senoide con amplitud modificada por el término dentro de los paréntesis. Este término es llamado la función de respuesta en frecuencia.

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = \left[\sum_{m=0}^{\infty} h(m) \exp(-j\omega m) \right] \quad (1.22)$$

La ecuación es una interesante relación, las muestras en el tiempo de la función de respuesta al impulso son la magnitud de las componentes discretas en frecuencia de $H(j\omega)$. Esta relación será usada más tarde para el diseño de filtros FIR.

Similarmente, si $x(n) = X z^n$, usando (1.17),

$$y(n) = \sum_{m=0}^{\infty} h(m) X z^{(n-m)} \quad (1.23)$$

y

$$y(n) = \left[\sum_{m=0}^{\infty} h(m) z^{-m} \right] X z^n \quad (1.24)$$

Como en la ecuación (1.21), la respuesta es de la misma forma que la entrada pero modificada por el término dentro del paréntesis rectangular. Este término es la transformada Z de $h(n)$, y para un número finito de términos está dada por:

$$H(z) = \sum_{m=0}^N h(m) z^{-m} = h(0) + h(1)z^{-1} + \dots + h(N)z^{-N} \quad (1.25)$$

1.6 Filtros con la respuesta finita al impulso.

La ecuación (1.25) implica que

$$Y(z) = H(z)X(z) = h(0)X(z) + h(1)z^{-1}X(z) + \dots + h(N)z^{-N}X(z) \quad (1.26)$$

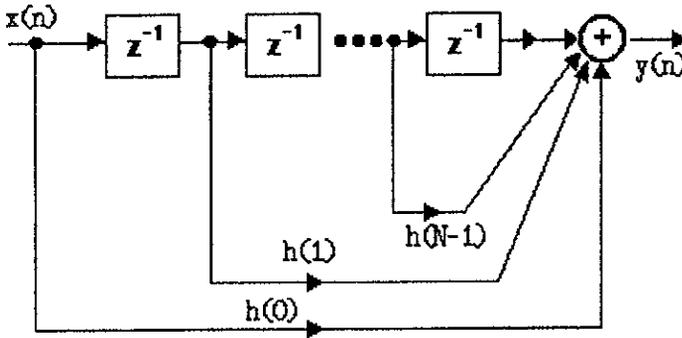


Figura 1.3: Estructura de un filtro FIR y sus retrasos.

Las ecuaciones (1.25) y (1.26) pueden ser representadas por la figura 1.3, donde z^{-1} representa los retrasos. La función de transferencia $H(z)$ puede ser representada como

$$H(z) = \frac{h(0)z^N + h(1)z^{N-1} + h(2)z^{N-2} + \dots + h(N)}{z^N}$$

con polos solamente en el origen. Como consecuencia, los filtros FIR son inherentemente estables. Tales filtros pueden ser implementados sin retroalimentación o el conocimiento de salidas anteriores. Una segunda y muy importante característica de los filtros FIR es que pueden ser diseñados para garantizar linealidad en la fase. Con la linealidad de fase garantizada, los filtros FIR son muy útiles en aplicaciones donde la distorsión de fase es crítica. Esta característica ocasiona que todas las componentes sinusoidales de entrada sean retrasadas en la misma cantidad.

Considere una ecuación en diferencias representando idealmente el retraso de una muestra $y(n) = x(n - k)$. Tomando la transformada de Fourier estos nos lleva a

$$Y(j\omega) = e^{-j\omega kT} X(j\omega)$$

La función de transferencia

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = e^{-j\omega kT}$$

tiene una magnitud constante de 1 y una fase lineal $\theta = -\omega kT$. La función de retraso de grupo, definida como la derivada de la fase con respecto a ω , es $-kT$, que es una constante. Los filtros FIR también se conocen como no recursivos, de movimiento promedio, transversales y filtros de retraso.

1.7 Valor esperado (E).

Si una variable continua aleatoria x tiene una función de frecuencia $f(x)$, su *valor esperado* $E[x]$ se define como

$$E[x] = \int_{-\infty}^{\infty} x f(x) dx$$

Así el valor esperado de una variable aleatoria es su valor medio. De forma más general, si $h(x)$ es una función de x , el valor esperado de $h(x)$ se define como

$$E[h(x)] = \int_{-\infty}^{\infty} h(x) f(x) dx$$

Propiedades de E.

Si c es cualquier constante,

$$E[cg] = cE[g]$$

Si g_1 y g_2 son dos funciones de un conjunto de variables aleatorias.

$$E[g_1 + g_2] = E[g_1] + E[g_2]$$

Si g_1 y g_2 son estadísticamente independientes

$$E[g_1 g_2] = E[g_1] E[g_2]$$

Capítulo 2

Filtros adaptativos

2.1 Introducción

Un sistema adaptativo es un sistema cuyos coeficientes pueden automáticamente ajustarse a un ambiente o señal de entrada que cambia continuamente. La transición de implementar filtros digitales con microprocesadores DSP a implementar filtros digitales adaptativos es una extensión natural. Con toda la capacidad de toma de decisiones que tiene el microprocesador, sería una lástima construir un sistema rígido que no pueda ajustarse a las condiciones cambiantes. Los filtros adaptativos son usados al máximo cuando hay incertidumbre acerca de las características de una señal o cuando estas cambian durante la operación del filtro. Un sistema adaptativo puede "aprender" las características de la señal y rastrear pequeños cambios. El análisis de sistemas adaptativos es mucho más difícil que el análisis de sistemas no adaptativos, es por esto que en este capítulo nos apoyaremos en ideas intuitivas para entender este tipo de sistemas.

2.2 Estructuras

Aunque existen muchas configuraciones para un sistema adaptativo, nosotros examinaremos las cuatro básicas que han demostrado un buen funcionamiento en un gran número de aplicaciones en el pasado.

2.2.1 Estructura general.

Conceptualmente el esquema adaptativo es muy simple. Casi todos los esquemas adaptativos pueden ser descritos por la estructura mostrada en la Figura 2.1. En esta figura la salida y es comparada con una salida deseada d para producir una señal de error e . La señal de error es la entrada a el algoritmo adaptativo que ajusta el filtro variable para satisfacer algún criterio predeterminado.

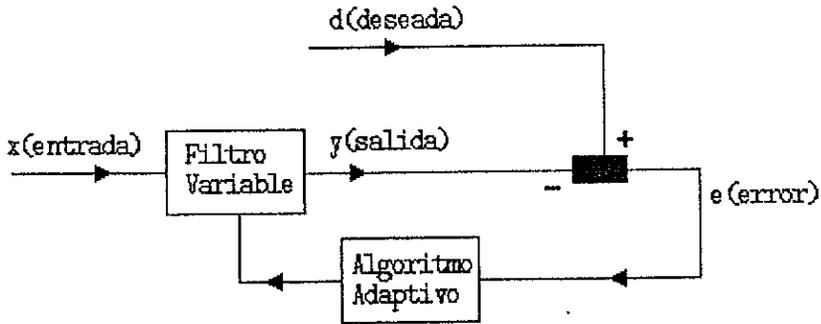


Figura 2.1: Estructura adaptativa en general.

La señal deseada es usualmente la más difícil de obtener. Una de las primeras preguntas que vienen a la mente es: Por qué tratamos de generar la señal deseada en y si ya la conocemos?. Sorprendentemente, en muchas aplicaciones la señal deseada existe en algún lugar en el sistema o la conocemos *a priori*. El reto en aplicar técnicas adaptativas es imaginarnos donde obtener la señal deseada d , cuál debe ser la salida y , y cuál debe ser el error e . Los siguientes párrafos nos mostrarán como y , d , y e son escogidos para algunas situaciones donde se aplican técnicas adaptativas.

2.2.2 Identificación del sistema

Las técnicas adaptativas son muy útiles para el modelado o identificación de sistemas, y la estructura usada se muestra en la Figura 2.2. La respuesta de un sistema desconocido a una entrada x es la señal deseada en este esquema. La respuesta del filtro adaptativo y , a la misma entrada x , es comparada con la respuesta del sistema desconocido d , para dar lugar a la señal de error e , que es usada para ajustar el bloque adaptativo de forma que se acerque al sistema desconocido. El filtro será ajustado hasta que la señal de error sea aproximadamente cero, y cuando esto sucede la adaptación se ha completado. Si el sistema desconocido es lineal e invariante en el tiempo, después de que la adaptación se ha completado, las características del filtro no cambiarán. El comportamiento de entrada/salida es igual que el sistema desconocido, de esta forma, obtenemos el modelo. Cuando el sistema se ha adaptado, y es igual a d y e es aproximadamente cero. La entrada debe ser ruido blanco limitado en banda con un espectro lo suficientemente ancho para excitar los polos del sistema desconocido.

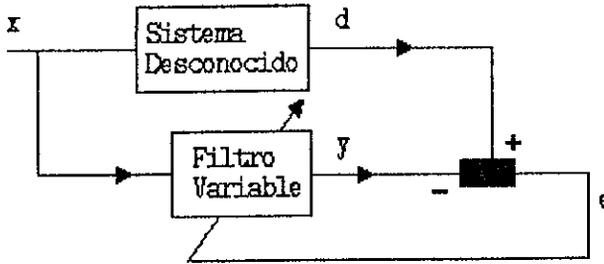


Figura 2.2: Identificación de un sistema.

2.2.3 Cancelación de ruido a la salida

Una segunda aplicación de un sistema adaptativo es la eliminación de distorsión o de ruido a la salida de un sistema cuya salida deba seguir a la entrada. Ejemplos de tales sistemas son los amplificadores o un servomecanismo. El diagrama para esta situación se muestra en la Figura 2.3. El ruido a la salida es modelado como ruido aditivo insertado en la salida del sistema. La señal contaminada proporciona la entrada al filtro adaptativo. Esta entrada es filtrada para producir una salida limpia en y . La salida en y es comparada con la señal deseada, que en este caso es la señal de entrada pero retrasada. El retardo es introducido para compensar los retrasos del sistema y los del propio filtro adaptativo. Nuevamente, después que el sistema se ha adaptado, y iguala a d y el error e tiende a cero.

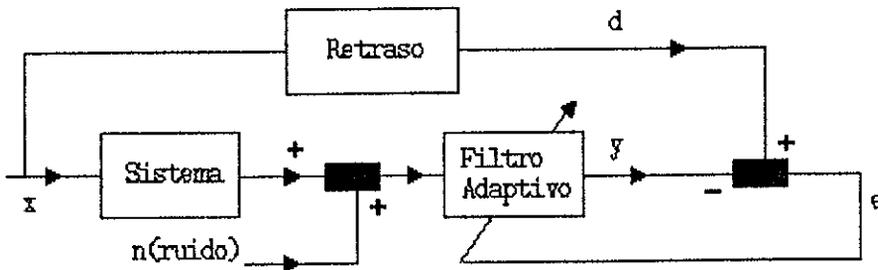


Figura 2.3: Cancelación de ruido a la salida.

2.2.4 Cancelación de ruido aditivo en la entrada

Un tercer ejemplo es el cancelador de ruido mostrado en la Figura 2.4. Aquí la señal x está contaminada con ruido aditivo n . La entrada del filtro en este caso es n' , que está correlacionada con el ruido n en la señal. Esto usualmente significa que el ruido viene de la misma fuente pero ha sido modificado por el ambiente de alguna manera. Cuando el sistema se ha adaptado, y se acerca al ruido aditivo n , y la señal de error se acerca a la señal de entrada x . Si x no está correlacionado con n , la estrategia es minimizar $E(e^2)$, donde $E()$ es el valor esperado. El valor esperado es generalmente desconocido, por eso, usualmente se aproxima con un valor promedio o con la función instantánea del mismo. La componente de la señal, $E(x^2)$, no será afectada y solamente la componente del ruido $E[(n - y)^2]$ será minimizada, dándonos una señal e cuya componente de ruido es minimizada.

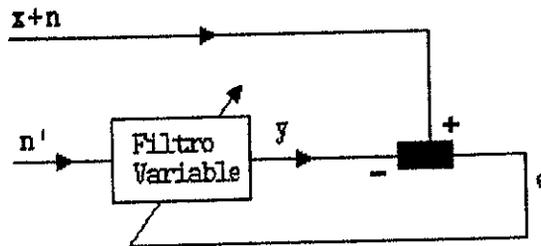


Figura 2.4: Cancelación de ruido aditivo en la entrada

2.3 Combinador lineal adaptativo.

Ahora consideraremos una de las más útiles estructuras para filtros adaptativos, el combinador lineal adaptativo. Existen dos tipos cuando se usa un combinador lineal: (1) con múltiples entradas y (2) con una sola entrada.

2.3.1 Múltiples entradas.

El caso de múltiples entradas se describe en la figura 2.5. La configuración consiste de K entradas independientes cada una de ellas ponderada por $w(k)$ y combinadas para formar la salida,

$$y(n) = \sum_{k=0}^K w(k, n)x(k, n) \quad (2.1)$$

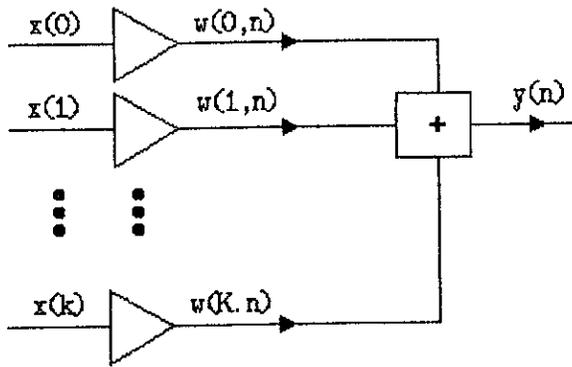


Figura 2.5: Combinador lineal con múltiples entradas.

La entrada puede ser representada como un vector dimensional $(K + 1)$,

$$\mathbf{X}(n) = [x(0, n) \quad x(1, n) \quad \dots \quad x(K, n)]^T \quad (2.2)$$

donde n es el índice de tiempo y la transpuesta T se usa con objeto de que podamos escribir el vector en una sola línea.

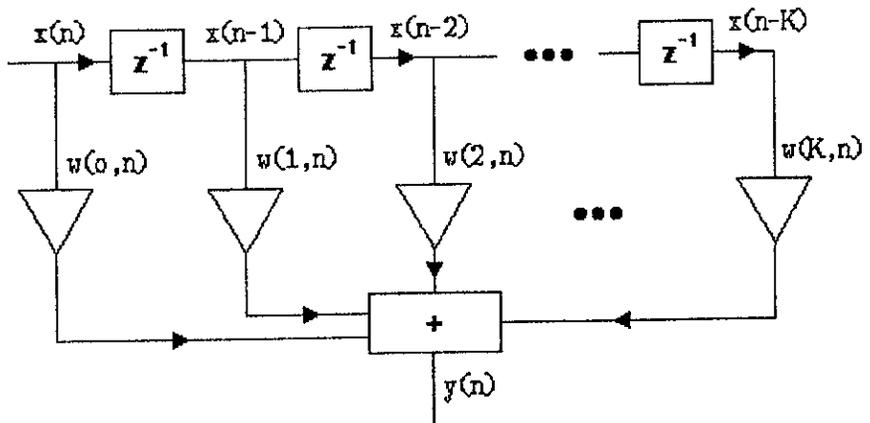


Figura 2.6: Combinador lineal adaptativo.

2.3.2 Una sola entrada.

En el caso de una sola entrada, la estructura se reduce a un filtro FIR con derivaciones y coeficientes ajustables, como se muestra en la figura 2.6. Cada entrada retrasada es ponderada y sumada para producir la salida.

$$y(n) = \sum_{k=0}^K w(k, n)x(n - k) \quad (2.3)$$

La única entrada y los pesos pueden ser escritos como vectores,

$$\mathbf{X}(n) = [x(n) \quad x(n - 1) \quad \dots \quad x(n - K)]^T \quad (2.4)$$

$$\mathbf{W}(n) = [w(0, n) \quad w(1, n) \quad w(2, n) \quad \dots \quad w(K, n)]^T \quad (2.5)$$

donde n es el índice de tiempo, que no escribiremos en la notación, ya sea para \mathbf{W} y x .

Usando notación vectorial, (2.3) se convierte en

$$y(n) = \mathbf{X}^T(n)\mathbf{W}(n) = \mathbf{W}^T(n)\mathbf{X}(n) \quad (2.6)$$

Las ecuaciones (2.1),(2.3), y (2.6) así como las Figuras 2.5 y 2.6 contienen exactamente la misma información.

2.3.3 Función de desempeño

En la sección anterior proporcionamos la estructura de un filtro cuyas características pueden cambiar por medio del ajuste del vector de pesos. De cualquier forma, todavía necesitamos una forma de valorar que tan bien está funcionando el filtro, es decir, una medición de su desempeño es necesaria. Esta función de desempeño, sera basada en el error, que se obtiene del bloque en la Figura 2.1, con el índice de tiempo incorporado:

$$e(n) = d(n) - y(n) \quad (2.7)$$

El cuadrado de esta función es

$$e^2(n) = d^2(n) - 2d(n)y(n) + y^2(n) \quad (2.8)$$

que es la función instantánea de error cuadrático. En términos de los pesos, se convierte

$$e^2(n) = d^2(n) - 2d(n)\mathbf{X}^T(n)\mathbf{W} + \mathbf{W}^T\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W} \quad (2.9)$$

donde el índice de tiempo en \mathbf{W} no lo escribimos. La ecuación (2.9) representa una superficie cuadrática en \mathbf{W} , lo que significa que la más alta potencia de los

pesos está al cuadrado. La estrategia consiste en ajustar los pesos de forma que la función de error al cuadrado sea un mínimo.

Para entender a la ecuación de la superficie de desempeño (2.9), consideremos el caso de un solo peso. La superficie de error será entonces

$$e^2(n) = d^2(n) - 2d(n)x(n)w(0) + x^2(n)w^2(0) \quad (2.10)$$

que es una función de segundo orden en $w(0)$. Para encontrar el mínimo, derivamos (2.10) con respecto a $w(0)$ e igualamos a cero, o

$$\frac{de^2(n)}{dw(0)} = -2d(n)x(n) + 2x^2(n)w(0) = 0 \quad (2.11)$$

dando como resultado

$$w(0) = \frac{d(n)}{x(n)} \quad (2.12)$$

que es el valor de $w(0)$ que nos proporciona el mínimo deseado.

Dado que las señales d y x son funciones en el tiempo, el mínimo y la superficie de desempeño también varían con las señales. Esto último NO es deseable, lo mejor sería una función de desempeño rígida. Para eliminar este problema podemos tomar el valor esperado de la función de error al cuadrado, que para un solo peso es

$$E[e^2(n)] = E[d^2(n)] - 2E[d(n)x(n)]w(0) + E[x^2(n)]w^2(0) \quad (2.13)$$

Esta función de desempeño se le conoce como el error cuadrático medio.

Observe que el valor esperado de cualquier suma es la suma de los valores esperados. Y que el valor esperado de un producto es el producto de los valores esperados sí y sólo sí son independientes estadísticamente. Las señales $d(n)$ y $x(n)$ generalmente NO son estadísticamente independientes. Si las señales d y x son estadísticamente invariables en el tiempo, los valores esperados de los productos de la señal d y x son constantes, y la ecuación (2.13) se puede reescribir como

$$E[e^2(n)] = A - 2Bw(0) + Cw^2(0) \quad (2.14)$$

donde A, B, y C son constantes.

Usando la ecuación (2.13) como la función de desempeño para un solo peso, obtenemos un punto mínimo fijo en una función de desempeño rígida,

$$w(0) = \frac{B}{C} \quad (2.15)$$

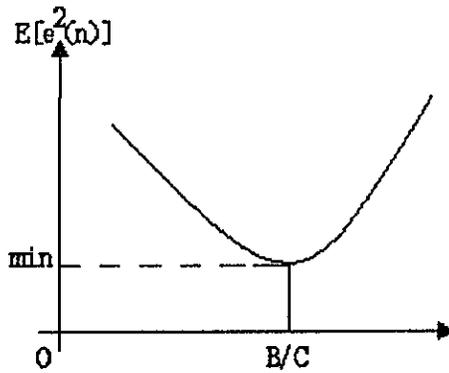


Figura 2.7: Curva de desempeño de un solo peso.

En la figura 2.7 podemos observar la gráfica de la función de error unidimensional con respecto a $w(0)$. Esta es una simple curva de segundo orden en dos dimensiones $(E[e^2], w(0))$ con un solo mínimo en $w(0) = B/C$. Si examinamos el caso de dos pesos, obtendremos una superficie tridimensional de segundo orden parecida a la mitad de un balón de fútbol americano. Con más pesos, obtendremos una superficie dimensionalmente más alta de segundo orden, y que no puede ser visualizada por los humanos. En la práctica, los pesos (el peso en este caso) comenzaran con un valor inicial w_i y serán ajustados hacia el mínimo valor de la función de desempeño. El procedimiento para ajustar los pesos será el objetivo de la próxima sección.

Tomando la media de la función general de error al cuadrado, la ecuación 2.9 da lugar a una función de desempeño general:

$$E[e^2(n)] = E[d^2(n)] - 2E[d(n)\mathbf{X}^T(n)]\mathbf{W} + \mathbf{W}^T E[\mathbf{X}(n)\mathbf{X}^T(n)]\mathbf{W} \quad (2.16)$$

Observe nuevamente que el valor medio de cualquier suma es la suma de los valores medios. El producto de los valores de d y \mathbf{X} con \mathbf{X}^T ya no pueden ser reducidos dado que el valor medio de un producto es el producto de los valores medios solamente cuando las dos variables son estadísticamente independientes; d y \mathbf{X} generalmente no lo son. Esta es aún una superficie de desempeño de segundo orden como la anterior, pero ahora está fluctuando con d y \mathbf{X} pero es rígida. De cualquier forma, si d y \mathbf{X} son estadísticamente variables en el tiempo la superficie de error también variara conforme lo hace d y \mathbf{X} .

2.3.4 Buscando el mínimo.

En esta sección trataremos como los pesos deben de ser ajustados para encontrar el mínimo de una forma razonablemente eficiente. Por supuesto, los pesos podrían ser ajustados aleatoriamente, pero dado que manejaremos eventos en

tiempo real, los cambios se deben seguir muy de cerca y es por eso que necesitamos una manera rápida de encontrar el mínimo.

Consideremos el sistema con un solo peso para tener una idea de como esta búsqueda puede llevarse a cabo. Inicialmente el peso será igual a un valor arbitrario $w(0, n)$, y será ajustado aproximación por aproximación hasta que el mínimo sea alcanzado (Figura 2.8).

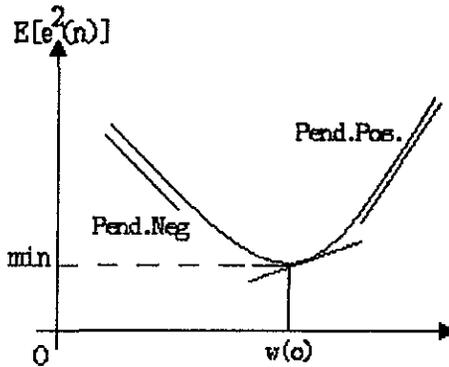


Figura 2.8: Búsqueda del mínimo para un solo peso.

El tamaño y dirección de cada aproximación son dos cosas que deben escogerse cuando se hace cada aproximación. Cada aproximación consistirá en sumar un incremento a $w(0, n)$. Observe que si el valor actual de $w(0, n)$ está a la derecha del mínimo, la aproximación debe ser negativa (pero la derivada de la curva es positiva); similarmente, si el valor actual está a la izquierda del mínimo, el incremento debe ser positivo (pero la derivada es negativa). Esta observación nos conduce a la conclusión que el valor negativo de la derivada nos indica la dirección apropiada del incremento. Dado que la derivada desaparece en el mínimo, también puede ser usada para ajustar la aproximación. Con estas observaciones concluimos que el tamaño y dirección de la aproximación pueden hacerse proporcionalmente al valor negativo de la derivada y cada iteración de los pesos entonces puede ser expresada como

$$w(0, n + 1) = w(0, n) - \beta \frac{dE[e^2]}{dw(0)} \quad (2.17)$$

donde β es una constante arbitraria y positiva, usualmente entre 0.1 y 1. Como se muestra en la Figura 2.8, la aplicación repetida de la ecuación (2.17) ocasionará que $w(0)$ se mueva por aproximaciones de su valor inicial hasta que alcanza el mínimo.

La derivada de la función usada en la búsqueda unidimensional puede ser

extendida a una superficie de N dimensiones reemplazándola con el gradiente de la función. El gradiente es un vector formado por las primeras derivadas con respecto a cada uno de los pesos:

$$\mathbf{grad}\{E[e^2]\} = \mathbf{grad}\{P\} = \left[\frac{\partial P}{\partial w(0)} \quad \frac{\partial P}{\partial w(1)} \quad \cdots \quad \frac{\partial P}{\partial w(K)} \right]^T \quad (2.18)$$

El gradiente apunta en la dirección en que la función, en este caso P , se incrementa más rápidamente. De esta forma, el tamaño y dirección de cada aproximación pueden hacerse proporcionalmente a el gradiente de la función de desempeño.

Similarmente, el mínimo de la curva de desempeño N dimensional ocurre cuando el gradiente desaparece,

$$\mathbf{grad}\{P\} = 0 \quad (2.19)$$

o cuando la derivada parcial con respecto a cada peso se hace cero.

$$\frac{\partial P}{\partial w(0)} = 0, \quad \frac{\partial P}{\partial w(1)} = 0, \quad \cdots \quad \frac{\partial P}{\partial w(K)} = 0 \quad (2.20)$$

Remplazando un solo peso por un vector de pesos y la derivada con la ecuación (2.17) obtenemos la regla de iteración para múltiples pesos.

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \beta \mathbf{grad}\{\mathbf{P}\} \quad (2.21)$$

El único problema por resolver ahora es como encontramos el $\mathbf{grad}\{P\}$. Para obtenerlo de una forma simple y práctica, usaremos una estimación en lugar del gradiente exacto. En vez de usar el gradiente del error cuadrático esperado, lo aproximaremos con el $\mathbf{grad}\{e^2\}$:

$$\mathbf{grad}\{P\} \simeq \mathbf{grad}\{e^2\} \quad (2.22)$$

Para obtener una expresión manejable, realicemos la operación del gradiente en la función de error al cuadrado,

$$\mathbf{grad}\{e^2\} = 2e \cdot \mathbf{grad}\{e\} \quad (2.23)$$

donde

$$e(n) = [d(n) - \mathbf{X}^T(n)\mathbf{W}(n)] \quad (2.24)$$

Sustituyendo nos lleva a

$$\mathbf{grad}\{e^2\} = 2e \cdot \mathbf{grad}\{d(n) - \mathbf{X}^T(n)\mathbf{W}(n)\} \quad (2.25)$$

Expandiendo el término gradiente tenemos que

$$\text{grad}\{e^2\} = -2e \begin{bmatrix} \frac{\partial e}{\partial w(0)} \\ \frac{\partial e}{\partial w(1)} \\ \dots \\ \frac{\partial e}{\partial w(K)} \end{bmatrix} = \begin{bmatrix} x(0) \\ x(1) \\ \dots \\ x(K) \end{bmatrix} \quad (2.26)$$

y

$$\text{grad}\{e^2(n)\} = -2e(n)\mathbf{X}(n) \quad (2.27)$$

Sustituyendo este resultado $\text{grad}\{P\}$ en la ecuación 2.21 tenemos que

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\beta e(n)\mathbf{X}(n) \quad (2.28)$$

El índice n ha sido incluido en las dos últimas ecuaciones, implicando que e será renovado cada vez que se muestrea. Observe que si e es cero, entonces $\mathbf{W}(n+1) = \mathbf{W}(n)$ y los pesos permanecen constantes.

La ecuación 2.28 forma el resultado más importante de este capítulo, y es la base para el algoritmo LMS (least mean square). Esta fórmula permite actualizar los pesos sin elevar al cuadrado, promediar, o diferenciar, y aún así es poderosa y eficiente.

Capítulo 3

Control Activo de Ruido

3.1 Generalidades

Los problemas de ruido acústico se han vuelto más evidentes debido al incremento en el uso de equipo industrial de grandes dimensiones, tales como máquinas, compresores, ventiladores, transformadores y extractores. Además, el crecimiento en la densidad de población ha incrementado la exposición de la población al ruido de una gran variedad de fuentes.

Aun más, la reducción de costos ha dado lugar a la tendencia de usar materiales más ligeros para construir edificios y equipo de transporte, que también ocasiona un aumento en el ruido en el ambiente. El ruido acústico no está necesariamente limitado al aire, el ruido hidroacústico es un problema bastante fuerte en algunas instalaciones marinas y también podría darse en instrumentos de laboratorio y aplicaciones médicas de scaneo por ultrasonido. La vibración mecánica es otro tipo de ruido muy relacionado que crea problemas en todas las áreas de transporte y manufactura, así como también en muchas aplicaciones para el hogar.

La forma tradicional para controlar el ruido acústico usa técnicas pasivas tales como aislamientos, barreras y silenciadores para atenuar el ruido no deseado. Los silenciadores pasivos usan ya sea el concepto de cambio de impedancia causado por la combinación de deflectores y tubos para eliminar el ruido indeseable (silenciadores reactivos) o el concepto de pérdida de energía causada por la propagación del sonido en un ducto con material especial que absorbe el sonido (silenciadores resistivos).

Los silenciadores reactivos son comúnmente usados como mofle en máquinas de combustión interna, mientras los silenciadores resistivos son usados casi exclusivamente para la reducción de ruido en sistemas con ductos de ventilación.

Los silenciadores pasivos son valiosos por su alta atenuación en un rango amplio de frecuencias; aunque son relativamente grandes, costosos e inefectivos a bajas frecuencias. La atenuación de los silenciadores pasivos es baja cuando la longitud de onda acústica es más grande comparada con las dimensiones del silenciador. Y lo que es más, estos silenciadores frecuentemente crean una contra-presión si hay un flujo de aire en el ducto.

Es conocido que el ruido de baja frecuencia es irritante si se está expuesto a largos períodos, y también que produce fatiga y pérdida de concentración, resultando en un decremento en la productividad. La mayoría de lo descrito hasta ahora, también es aplicable al ruido hidroacústico. Por las mismas razones, la atenuación del ruido hidroacústico no es efectiva o resulta impráctica a bajas frecuencias. Los problemas de vibración es otra área donde las técnicas pasivas tienen limitaciones. El control pasivo de vibraciones se lleva a cabo desacoplando con un amortiguador masa resorte, que es la analogía mecánica de un filtro paso bajas. La frecuencia de resonancia debajo de la cual el aislamiento vibratorio es inefectiva, es inversamente proporcional a la raíz cuadrada del resorte y de la masa. Así, conforme la frecuencia de interés decrece, la masa o el resorte deben incrementarse, hasta llegar a un punto en el que o no es práctico, o es demasiado caro. De esta forma, como en el ruido acústico, la efectividad de las técnicas de control de ruido pasivas se limitan a bajas frecuencias.

En un esfuerzo por resolver estos problemas, el control activo de ruido¹, en el que se usan fuentes adicionales secundarias para cancelar la fuente primaria de ruido, ha despertado un considerable interés y ha mostrado soluciones promisorias. El control activo de ruido involucra un sistema electroacústico o electromecánico que cancela el ruido primario (no deseado) usando el principio de superposición; específicamente un "antiruido" de igual amplitud y fase opuesta es generado y combinado con la fuente primaria de ruido, resultado en la cancelación de ambos ruidos.

La Figura 3.1 muestra las formas de onda del ruido primario no deseado, el antiruido secundario, y el ruido residual que resulta de la superposición de ambos. La cancelación del ruido primario depende de la exactitud de la amplitud y fase del antiruido generado. El sistema ANC atenúa eficientemente el ruido de baja frecuencia donde los métodos pasivos pueden ser inefectivos o tender a ser extremadamente caros y/o demasiado grandes. Así la aplicación de la técnica ANC es un moderno suplemento a los sistemas convencionales pasivos.

El ANC acústico se está desarrollando rápidamente porque permite mejoras en el control de ruido, generalmente con beneficios tales como tamaño, peso, volumen y costo. Y lo que es más, el ruido puede ser reducido en la posición del escucha sin modificaciones físicas o cambios de lugar de las fuentes de ruido existentes.

¹Active Noise Control,ANC

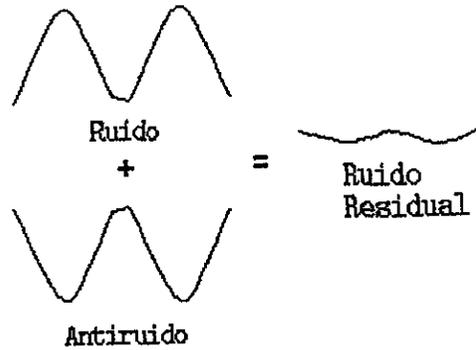


Figura 3.1: Concepto físico del ANC.

Beneficios semejantes de la aplicación del ANC pueden obtenerse en problemas de reducción de vibraciones. También estos sistemas pueden ser fácilmente modificados dentro de los confines de un equipo ya existente.

El diseño de un sistema ANC acústico utilizando un micrófono y una bocina controlada electrónicamente para generar un sonido cancelador, fué propuesto por primera vez en 1936 en una patente por Lueg. Mientras la patente delineaba la idea básica de un sistema ANC, no tuvo aplicación en el mundo real en ese entonces. Dado que las características de la fuente de ruido acústico y el ambiente son variantes en el tiempo, el contenido espectral, la amplitud, fase y velocidad del sonido del ruido no deseado son no estacionarias, de tal forma que un sistema ANC debe ser adaptable para poder manejar estas variaciones.

Técnicas similares de diseño pueden aplicarse a ANC vibratorio. En este caso, las fuentes secundarias son realizadas como actuadores hidráulicos o electromagnéticos que tienen un efecto similar que el de la bocina en un sistema acústico. Las soluciones adaptables también son deseables para esta clase de problemas dado que las propiedades de las vibraciones estructurales depende de las variables que existen en el medio tales como temperatura y carga estática.

El problema fundamental que se debe considerar en los sistemas ANC es el requerimiento de un control muy preciso, estabilidad temporal y confiabilidad. Con el objetivo de producir un alto grado de atenuación, la amplitud y la fase de ambos, la fuente primaria de ruido y las fuentes secundarias deben igualarse con una precisión muy cercana. Es por esto que es deseable que el cancelador de ruido sea digital, donde las señales de los transductores electroacústicos o electromecánicos sean muestreadas y procesadas usando un sistema de procesamiento digital de señales ² con la suficiente velocidad y precisión para ejecutar sofisticadas funciones matemáticas en tiempo real.

²DSP, Digital Signal Processing

El control activo no adaptivo de sistemas dinámicos ha sido desarrollado desde un acercamiento basado en formar un modelo del sistema y calcular el controlador óptimo de la respuesta deseada. Dada la complejidad de los sistemas acústicos y mecánicos reales donde se desea un control activo, se presentan dificultades cuando se toma en cuenta la impedancia de acoplamiento entre las fuentes por medios analíticos. Aun más, en muchas aplicaciones ANC, las propiedades del campo de ruido son no estacionarias. Por ejemplo, la velocidad y la carga de una máquina o de un extractor están cambiando, lo que conlleva a una frecuencia y amplitud variable del ruido no deseado. Otro ejemplo sería la temperatura y los cambios en el flujo en un ducto, que da lugar a cambios de velocidad en el sonido. De la misma forma, las propiedades de las vibraciones mecánicas pueden cambiar con la temperatura. Con el objeto de seguir y responder a tales cambios en tiempo real, los sistemas ANC tienen que ser adaptables.

En el campo del procesamiento digital de señales existe un tipo de sistema adaptable en el que las propiedades de un filtro digital son ajustadas para minimizar una función específica dada.

Sólo con el reciente advenimiento del procesamiento adaptable de señales y la introducción de poderosos y no muy costosos procesadores DSP, estos procesos han llegado a ser prácticos. En los ochentas, la investigación de ANC fué dramáticamente afectada por el desarrollo de poderosos procesadores para el procesamiento digital de señales y el desarrollo de algoritmos adaptables. Estos DSPs fueron diseñados para un procesamiento numérico de muy alta velocidad de señales digitalizadas en tiempo real. Además de permitir la implementación de poderosos algoritmos adaptables han ocasionado que una nueva tecnología emerga.

Los filtros adaptables han recibido una considerable atención de los investigadores en los últimos 20 años, algunos autores que pueden mencionarse son Goodwin y Sin, Widrow y Stearns, Cowan y Grant, Honig y Messerschmitt, entre muchos otros. Los filtros adaptivos ajustan sus coeficientes para minimizar una señal de error y pueden ser realizados como ya vimos con una respuesta finita (FIR o transversal) o una respuesta infinita (IIR) al impulso. La forma más común de filtro utilizado es el filtro transversal que utiliza el algoritmo LMS.

Actualmente muchos sistemas ANC dependen en gran parte de un procesamiento con cada vez menos elementos electromecánicos o electroacústicos, y con más esquemas muy elaborados de filtrado adaptivo. De cualquier forma, si el arreglo físico de las fuentes no es optimizado, el sistema tendrá problemas para compensar las deficiencias y puede fallar. Es necesario entender y diseñar la acústica o la mecánica de las instalaciones con el objeto de que el controlador de ruido haga su trabajo.

El rango de problemas de control de ruido que puede ser resuelto por medio de un sistema activo aún es limitado, solamente el ruido a frecuencias relativamente bajas puede ser exitosamente atenuado por tales métodos. La com-

binación de la absorción pasiva con la cancelación activa puede dar un mejor resultado que el uso de una sola técnica. En muchos aspectos, ANC es una tecnología complementaria a los silenciadores pasivos.

3.2 Aplicaciones generales

Las aplicaciones ANC serán determinadas basándose en la efectividad de este método en comparación con las técnicas pasivas. ANC es una forma atractiva de eliminar gran cantidad de ruido a una mínima cantidad, especialmente a bajas frecuencias. Su desempeño en altas frecuencias está limitado por varios factores, incluyendo la necesidad de una tasa de muestreo mucho más grande y la existencia de modos de un orden más alto que dan como resultado un campo de ruido mucho más complicado. Afortunadamente, las técnicas pasivas tienden a ser mucho más adecuadas a más altas frecuencias y ofrecen una buena solución para casi todos los problemas de ruido. En bajas frecuencias, donde las tasas de muestreo bajas son adecuadas y las longitudes de onda son más grandes, la atenuación activa ofrece ventajas muy significativas.

Aplicaciones específicas para ANC que están ahora en desarrollo, incluyen la atenuación de fuentes inevitables de ruido en el siguiente equipo terminal:

Automotriz

- Sistemas de un solo canal (unidimensionales): Mofles electrónicos para sistemas de extracción, de inducción, etcétera.
- Sistemas multicanal (tridimensionales): Atenuación de ruido dentro del compartimento de pasajeros y cabinas de operación de equipo pesado, teléfono celular a manos libres, etcétera.

Doméstico

- Sistemas de un solo canal: Ductos de aire acondicionado, refrigeradores, lavadoras, deshumificadores, calentadores, etcétera.
- Sistemas multicanal: podadoras de pasto, limpiadores de vacío, aislamiento de espacios cerrados (zonas de silencio locales), etcétera.

Industrial

- Ventiladores, ductos de aire, chimeneas, transformadores, compresores, bombas, motosierras, tuneles de viento, protección auditiva, audífonos, etcétera.

Transporte

- Aviones, barcos, helicópteros, locomotoras de Diesel, etcétera.

3.2.1 ANC Aire-Acústico

Desde un punto de vista geométrico, las aplicaciones ANC acústicas pueden ser clasificadas en las siguientes cuatro categorías:

1. Ruido en los ductos: un ducto unidimensional usado en la calefacción, aire acondicionado, extracción, etcétera.
2. Ruido interior: ruido confinado en un espacio interior.
3. Ruido en el espacio libre: ruido radiado en el espacio abierto.
4. Protección auditiva personal: un caso en pequeña escala del ruido interior.

Ruido en los ductos. Una gran parte del trabajo realizado en ANC ha sido dirigido hacia el control de ruido acústico en ductos de calefacción, ventilación, aire acondicionado (HVAC) o ductos de extracción, debido a que tiene un enorme potencial en aplicaciones industriales y la facilidad de construir sistemas experimentales. Para los sistemas de aire acondicionado, la reducción de ruido ha sido generalmente en los ductos, que son los medios de transmisión del ruido acústico, no en las unidades de aire acondicionado en sí. Esto es muy útil no sólo en hospitales y en salas de conciertos, que intrínsecamente necesitan un nivel de ruido muy bajo, sino que también esto es deseable en edificios con oficinas, salas de conferencias, salones de clases y por qué no? la sala de nuestros hogares.

La turbulencia causada por el rápido movimiento del aire en las esquinas puede generar un sonido aleatorio muy ruidoso. ANC puede ser aplicado efectivamente a este problema en los sistemas HVAC. También se han hecho experimentos para examinar la capacidad de los sistemas ANC para adaptarse a cambios de temperatura, aún en un rango de 30° por medio del control adaptivo. El silenciador activo de un sistema HVAC no sólo proporciona una mejor atenuación de baja frecuencia sino que también reduce la carga del ventilador arriba de un 40%. Un amplio campo de desarrollo en ANC ha sido aplicado a la cancelación del ruido provocado por el ventilador en los ductos de ventilación, en los ductos de entrada y de descarga de ventiladores industriales, y en manejadores comerciales de aire.

La ventaja más sobresaliente de ANC sobre los silenciadores convencionales es su superior atenuación a bajas frecuencias con una restricción de flujo casi despreciable, lo que implica un ahorro de energía. Otros beneficios adicionales de ANC en esta aplicación son la ligereza y la capacidad de colocar el sistema en arreglos de ductos ya existentes.

Ruido interior. El ruido generado dentro de un espacio confinado puede ser atenuado por varios métodos. El método depende de donde proviene el

ruido, si de una fuente interior o del espacio exterior. Cuando el ruido está dentro, la fuente de cancelación puede ser colocada cerca de la fuente, de cualquier forma se necesita tener cuidado para reducir la retroalimentación de la fuente canceladora con el micrófono de referencia. Por ejemplo en un avión de propulsión, la fuente principal de ruido dentro de la cabina de pasajeros es el sonido generado por los propulsores que están afuera. Esto produce ruido en múltiplos de la frecuencia de las aspas. Aunque el campo de sonido es producido en sólo unas cuantas frecuencias, la distribución espacial en el ruido de la cabina puede ser algo complicado. Sistemas similares se han desarrollado también para el interior de un vehículo de pasajeros, y en algunos casos estos son integrados con el sistema de radio y bocinas.

Ruido en el espacio libre. Los sistemas ANC pueden también ser usados para reducir la exposición de los trabajadores al ruido propio del lugar de trabajo y reducir las molestias a las comunidades cercanas. Si se requiere cancelar el ruido en un área específica de un campo de sonido muy amplio, el control local es realizable colocando bocinas y micrófonos de error cerca del punto de control. Este concepto es realizado como una "barrera acústica", o como un "asiento silencioso" donde las bocinas canceladoras son colocadas cerca de la cabeza de un asiento con el fin de cancelar el ruido de baja frecuencia en los oídos de la persona sentada.

Protección auditiva personal. La situación acústica más sencilla para un sistema ANC está definida por un espacio cuyas dimensiones son pequeñas comparadas con la longitud de onda del rango de frecuencias de interés. Tal espacio es definido por el pequeño volumen entre el audífono y la membrana del tímpano. El sistema activo cancela los ruidos de baja frecuencia, el caracol del oído (sistema pasivo) atenúa los ruidos de alta frecuencia. Los protectores activos auditivos proporcionan la suficiente protección para la mayoría de aplicaciones industriales y militares, al mismo tiempo que facilitan la comunicación. Estos protectores pueden ser usados en el ejército, minas, plantas manufactureras y en scanners de imagen de resonancia magnética ³.

Los sistemas ANC también pueden ser usados en combinación con sistemas de comunicación para mejorar la inteligibilidad dentro de un ambiente ruidoso. En este caso, ANC es combinado con la cancelación de eco acústico y con la cancelación adaptable de ruido aplicada a una señal eléctrica. La voz que llega es transmitida por una bocina que también produce un ruido secundario para reducir el ruido acústico del ambiente percibido por el escucha. La voz de respuesta es sensada por un micrófono, que también puede ser usado como el micrófono de error en el sistema ANC, y transmite una señal hacia el otro extremo. Un cancelador activo de ruido es usado

³Magnetic Resonance Imaging. MRI

para reducir el ruido indeseado tomado por el micrófono. Finalmente, un cancelador de eco acústico puede ser usado para reducir el eco acústico que va de la bocina a el micrófono.

3.2.2 ANC Hidroacústico

Numerosas fuentes subacuáticas de ruido interfieren con los sistemas militares así como también de sistemas comerciales para monitoreo ambiental, rastreo de posición y búsqueda de peces. De esta forma, las técnicas ANC podrían jugar un papel muy importante en este campo. También podrían concebirse aplicaciones en otro tipo de sistemas hidroacústicos donde el ruido se propaga en fluidos, tales como instrumentación de laboratorio que involucra dispositivos hidráulicos y equipo médico ultrasónico de exploración. Aunque se han hecho algunos trabajos en esta área, existe aún mucho territorio sin explorar. A diferencia del ANC aire-acústico donde ha habido una extensa investigación y desarrollo, muchas aplicaciones para ANC hidroacústico están esperando ser descubiertas.

3.2.3 ANC Vibratorio

Aún no hay textos definitivos acerca de ANC vibratorio, de cualquier forma existen muchas referencias en la literatura contemporánea.

3.3 Sistemas ANC

Existen dos tipos de ruido en el ambiente, el de banda estrecha y el de banda ancha. El de banda ancha es causado, por ejemplo por turbulencia, que es totalmente aleatorio, y por ende distribuye su energía más o menos de manera uniforme en la banda de frecuencias. Ejemplos de ruido turbulento de banda ancha son los sonidos de baja frecuencia de los aviones o el ruido de impulsión de una explosión. En contraste, el ruido de banda estrecha concentra la mayoría de su energía en frecuencias específicas. Este ruido está relacionado a máquinas rotatorias o repetitivas, de tal forma que son periódicas o cuasiperiódicas. Ejemplos de fuentes de ruido de banda estrecha son los motores de combustión interna usados en el transporte o en aplicaciones estacionarias, turbinas como fuentes auxiliares de energía, los compresores en sistemas de refrigeración y bombas de vacío usadas para transportar materiales en muchas industrias.

Para cancelar ruido de banda ancha, la señal de entrada de referencia debe darnos una indicación lo suficientemente avanzada del ruido que se aproxima. El ruido primario que se correlaciona con la señal de entrada de referencia será cancelado en la dirección de la fuente secundaria. De cualquier forma, la retroalimentación de la fuente secundaria y el sensor de referencia puede causar problemas. Por ejemplo, en aplicaciones aire-acústicas, tal retroalimentación ocurre de la bocina canceladora y el micrófono de referencia.

Para ruido periódico ocasionado por maquinaria rotante, se han desarrollado técnicas de banda estrecha que son muy efectivas en reducir ruidos repetitivos sin apoyarse en la causalidad. En lugar de usar un sensor de referencia, los sistemas ANC periódicos pueden emplear una señal proveniente de un tacómetro para proporcionar la información acerca del ruido. Dado que todo el ruido repetitivo se encuentra en las armónicas de la frecuencia a la cual rota la máquina, el sistema cancelará estas frecuencias ruidosas ya conocidas. Este tipo de control es necesario, por ejemplo en la cabina de algún vehículo, dado que no afectara las señales preventivas, el desempeño del radio o de voz, que están raramente sincronizados con la maquinaria rotatoria.

ANC se basa en dos técnicas de control: feedforward, donde una señal de entrada, coherente y ruidosa, es tomada como referencia y sensada antes de que se propague hacia la fuente secundaria, o feedback, donde el controlador activo de ruido intenta cancelar el ruido sin el beneficio de una entrada "rio arriba" de referencia.

Las estructuras para el ANC feedforward están clasificadas en (1) control adaptivo feedforward de banda ancha con un sensor de referencia para control de campo (por ejemplo el acústico); y (2) control adaptivo feedforward de banda estrecha con un sensor de referencia que no está influenciado por el campo de control (por ejemplo un tacómetro). Se han hecho grandes progresos sobretodo en el campo del control feedforward para sistemas con ductos. La técnica feedforward es generalmente más robusta que la feedback, particularmente cuando el sistema feedforward tiene una entrada de referencia aislada de la fuente antiruido secundaria. Los sistemas ANC feedback están más dentro de la teoría de control clásica pero pueden ser interpretados dentro del contexto del ANC tipo feedforward como el usar un predictor lineal para proporcionar la señal de referencia.

Las estructuras y algoritmos desarrollados para ANC acústico pueden ser igualmente aplicados a ANC vibratorio, que se usa para aislar las vibraciones de una variedad de máquinas, así como para estabilizar varias plataformas en la presencia de vibraciones perturbadoras. Conforme el desempeño y confiabilidad de los sistemas activos continúe mejorando y los costos iniciales continúen reduciéndose, pueden llegar a convertirse en la solución preferible a una gran variedad de problemas de control a vibraciones.

3.3.1 Sistemas ANC feedforward de banda ancha

Este tipo de sistema lo podemos ejemplificar en el control de ruido acústico en ductos largos y estrechos, tales como conductos de drenado y sistemas de ventilación, como se ilustra en la Figura 3.2.

En este diagrama, una señal indeseada $x(n)$ es sensada antes de que pase por la bocina que se encuentra "rio abajo" en el ducto usando un micrófono de referencia ("rio arriba") cerca de la fuente de ruido. El espaciado entre el micrófono y la bocina (fuente secundaria) permite la causalidad y la alta coheren-

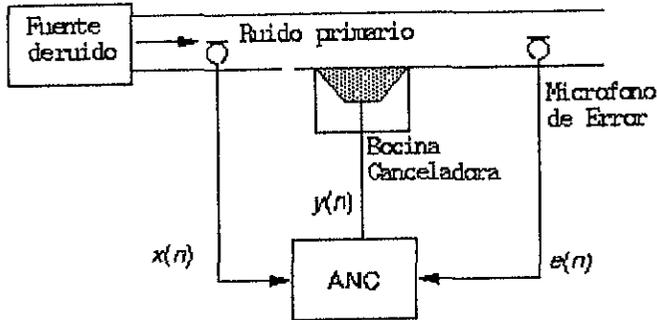


Figura 3.2: Sistema ANC feedforward de banda ancha de un solo canal en un ducto.

cia entre el sensor de ruido "rio arriba" y el sonido a ser cancelado. El sistema ANC usa una señal de referencia para generar una señal $y(n)$ que controla a la bocina que produce el ruido secundario en el ducto acústico que está defasada 180 grados con respecto a la perturbación, cancelando de esta forma el ruido no deseado. Siempre y cuando el rango de frecuencias del ruido esté por debajo de la frecuencia natural del primer modo en el ducto, los sistemas activos son capaces de reducir substancialmente los niveles de ruido. La frecuencia límite inferior del sistema activo es determinada por las fluctuaciones de presión turbulenta y el límite superior por la frecuencia de corte para el modo más alto de propagación del sonido en el ducto o por la velocidad del procesador de señales.

Escoger el lugar correcto en donde colocar la bocina cerca de la fuente primaria resultará en una reducción total de la radiación acústica de energía debido al acoplamiento mutuo; las fuentes primaria y secundarias actúan la una sobre la otra para afectar la impedancia de radiación, reduciendo de esta forma la potencia del sonido radiada.

El micrófono de error mide el error residual $e(n)$, que es minimizado adaptando los coeficientes de un filtro adaptable. Una característica básica del sistema ANC feedforward de banda ancha es que el tiempo de retraso de la propagación entre el sensor de ruido "rio arriba" y la fuente de control activa permite que el antiruido sea reintroducido en una posición en el campo donde causará cancelación. El uso de una señal de error "rio abajo" para ajustar los coeficientes del filtro adaptable no constituyen una retroalimentación porque la señal de error no regresa a la fuente secundaria. A muy bajas frecuencias, el

sonido se propaga en el ducto como una onda plana, que es efectivamente cancelada por el controlador de una sola referencia y una sola salida mostrado en la Figura 3.2. Aparte del modo de onda plana, el sonido se puede propagar en modos más altos arriba de una frecuencia crítica que depende de la geometría del ducto. La solución general es aplicar un controlador de múltiples referencias y múltiples salidas para cancelar todos los modos que se presenten.

Como se muestra en la Figura 3.2, el uso de un micrófono de referencia introduce la posibilidad de retroalimentación indeseada que va de la fuente secundaria a la entrada del micrófono de referencia. De esta forma el sistema completo forma un lazo cerrado, complicando el problema de adaptación, o convirtiendo al sistema en inestable. Este problema puede resolverse si el micrófono de referencia es direccional "rio arriba" y la bocina es direccional "rio abajo". De cualquier forma, este problema se resuelve típicamente, modelando la función de transferencia y neutralizándolo por medios electrónicos.

3.3.2 Sistemas ANC de banda estrecha feedforward

Para muchas aplicaciones, el ruido primario es producido por máquinas recíprocas o rotatorias y por lo tanto es periódico o cuasiperiódico. En este caso, el sensor de referencia puede ser remplazado por sensores no acústicos o sensores no vibratorios, tales como un tacómetro o un sensor óptico. Un diagrama a bloques para un sistema ANC acústico de este tipo en un ducto se muestra en la Figura 3.3.

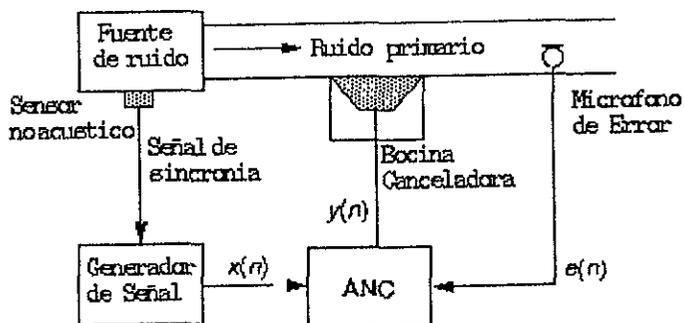


Figura 3.3: Sistema ANC feedforward de banda estrecha de un solo canal en un ducto.

La gran ventaja de los sensores de referencia no acústicos es que no son sensibles al sonido cancelador, eliminando el problema antes mencionado de la retroalimentación; una ventaja similar ocurren en los sistemas vibratorios.

3.3.3 Sistemas ANC Feedback

Los sistemas ANC tipo feedback fueron propuestos por Olson y May en 1953. Este concepto se ilustra en la Figura 3.4 para el problema de ruido en un ducto.

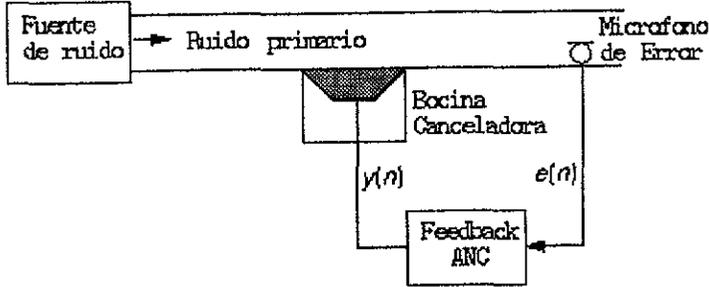


Figura 3.4: Sistema ANC feedback de un solo canal en un ducto.

La señal de error del sensor es regresada a través de un amplificador con respuesta en magnitud y fase diseñada para producir cancelación en el sensor. El sonido indeseado puede ser reducido por medio del uso de una ganancia de lazo cerrado muy alta. Este tipo de sistema retroalimentado generalmente es muy problemático. Por ejemplo, si existe algún error en la respuesta de fase del controlador generalmente dará lugar a oscilaciones, ya que este error puede cambiar la retroalimentación negativa (que es la deseada) a positiva, y volverse inestable.

Un caso especial de ruido de banda estrecha hace al ANC feedback más manejable. En este caso, el ANC feedback puede ser interpretado como el uso de la salida del sensor de error para crear una entrada de referencia, dada la naturaleza predictiva de las señales de banda estrecha.

3.4 Sistemas ANC Multicanal

Muchas aplicaciones involucran un comportamiento modal complejo, tales como las aplicaciones ANC en grandes ductos o lugares cerrados y ANC vibratorio en cuerpos rígidos o estructuras con múltiples grados de libertad. Cuando la geometría del campo del ruido primario es más complicada, una sola fuente secundaria se vuelve insuficiente para cancelar el ruido primario usando un solo sensor de error. El control de campos de ruido más complicados requiere conjuntamente la exploración y el desarrollo de estrategias óptimas y la construcción de un adecuado controlador multi-canales. Este problema requiere el uso de un

algoritmo adaptativo multi-referencias y multi-salidas. Un sistema ANC multi-canal en general, involucra un arreglo de sensores y de fuentes secundarias. Un diagrama a bloques de un sistema multi-canal ANC acústico se muestra en la Figura 3.5

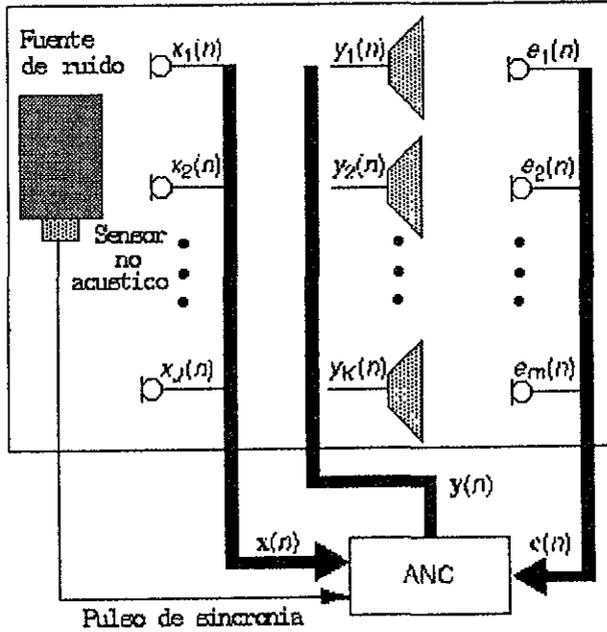


Figura 3.5: Sistema ANC acústico multicanal de 3-D.

Si una fuente primaria de tonos puros controla un ambiente de tres dimensiones como el de un compartimento de un vehículo, muchos modos acústicos de tal ambiente serán excitados. En un principio el número de fuentes necesarias para obtener una cancelación perfecta del sonido en un ambiente es el mismo que el número de modos acústicos que están siendo excitados. En la práctica, es muy difícil lograr un control independiente sobre un modo acústico en particular sin afectar a los otros, algunos de los cuales quizá ni siquiera hayan sido significativamente excitados por la fuente primaria. Dado que la densidad modal acústica en ambientes se incrementa rápidamente con la frecuencia, el control activo de campos acústicos en frecuencias más altas se vuelve muy complicado. De cualquier forma, los métodos pasivos tienden a trabajar mejor a altas frecuencias, y de esta forma complementan el ANC, que es mucho más efectivo a bajas frecuencias.

A el sonido de baja frecuencia se le reconoce cada vez más como un peligro para la salud y causa un malestar y fatiga en ambientes tales como las cabinas de vehículos pesados de construcción. Varios sonidos son responsables por el ruido

en las cabinas, el más predominante es la unidad de alimentación(un motor, una línea de extracción, un ventilador de enfriamiento, etcétera). Las componentes de baja frecuencia del ruido están directamente relacionadas con la velocidad de la máquina. Aunque las fuentes de ruido en la cabina son repetitivas, sus longitudes de onda son pequeñas comparadas con las dimensiones del ambiente y su compleja geometría frecuentemente es tal, que es imposible dividir las fuentes separadas y eliminar cada una individualmente. Existen dos posibles soluciones a este problema: (1) colocar la fuente canceladora cerca o dentro de la cavidad auditiva, como en los audífonos; o (2) crear una zona espacial de cancelación de ruido usando bocinas.

Los micrófonos de error, mientras ayudan a crear la zona de silencio en el ambiente, deben ser posicionados cuidadosamente con el objeto de adquirir una estimación aceptable de la función de costo de la zona de silencio a ser minimizada. Las fuentes secundarias deben ser localizadas para facilitar el acoplamiento a cada modo acústico.

3.5 ANC Feedforward de banda ancha

Principios básicos

Un sistema ANC de banda ancha usado dentro del ducto mostrado en la Figura 3.6 puede ser descrito dentro de las estructuras para identificar sistemas. El ruido no deseado de la fuente primaria es medido por un micrófono de entrada de referencia, procesado a través de un filtro adaptativo, y usado para controlar el sonido de la fuente secundaria, como es el caso de la bocina que cancelará el ruido en el ducto.

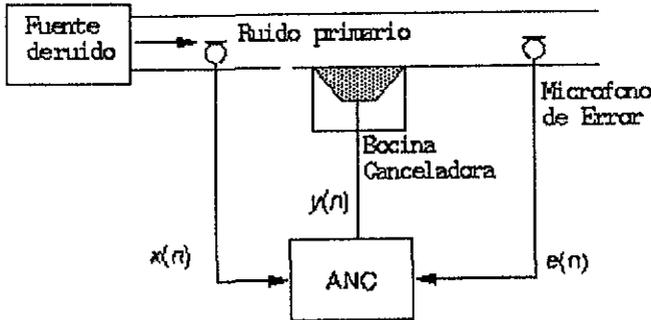


Figura 3.6: Sistema ANC feedforward monocanal en un ducto.

La señal de entrada del sensor de referencia debe estar correlacionada con el ruido de la fuente primaria. En sistemas que controlan ruido de banda ancha (aleatorio), la señal de referencia proporciona información por adelantado del ruido primario antes de que llegue a la bocina canceladora, esto es un requerimiento necesario para un controlador causal. El ruido residual es detectado por un micrófono de error y la información que proporciona sirve para cambiar los coeficientes del filtro adaptativo de forma que minimize el ruido residual. Para simplificar el problema asumimos que no existe retroalimentación entre la bocina canceladora y el micrófono de referencia.

Como se muestra en la Figura 3.7 una forma simplificada de este tipo de sistemas ANC es considerada meramente como un problema de identificación de un sistema en el que un filtro $W(z)$ es usado para estimar una planta desconocida $P(z)$. Asumimos que la planta y el filtro adaptativo son excitados por la misma entrada $x(n)$. La salida de la planta proporciona la respuesta deseada para el filtro adaptativo. Si la planta es dinámica, el modelo será variante en el tiempo. El algoritmo adaptativo tiene la tarea de mantener el error del modelo lo más pequeño posible, esto lo logra siguiendo las variaciones en el tiempo de la dinámica del sistema. La trayectoria primaria $P(z)$ consiste de la respuesta acústica que va desde el sensor de referencia al sensor de error donde la ate-

nuación de ruido va a realizarse. En las aplicaciones reales se deben incluir otras funciones de transferencia que se explicarán más tarde. En este texto asumimos que todos los elementos acústicos, electrónicos y electroacústicos son lineales.

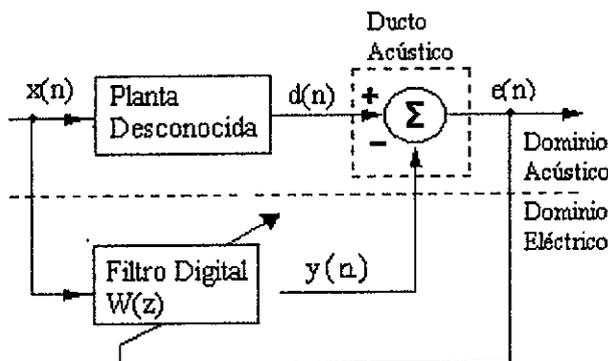


Figura 3.7: Identificación de un sistema desde el punto de vista ANC.

Existen 3 elementos a considerar cuando se identifica un sistema de forma adaptativa: la señal de excitación, la estructura del filtro y el mecanismo adaptativo. Si la señal de excitación $x(n)$ es rica en su contenido espectral (para excitar todos los modos del sistema) y el ruido interno de la planta es pequeño, el filtro adaptativo convergerá a una buena aproximación del sistema desconocido. El filtro puede tener únicamente ceros, polos o una combinación de ambos. Un modelo con únicamente ceros está representado por un filtro FIR. Un modelo con únicamente polos o una combinación de polos y ceros está representado por un filtro IIR. El algoritmo más común es el algoritmo LMS del cual ya se habló, pero también existen otros más complejos que hacen uso de la estructura de la matriz de autocorrelación de $x(n)$.

El objetivo del filtro adaptativo $W(z)$ es minimizar la señal de error residual $e(n)$. La diferencia más importante entre la Figura 3.7 y el esquema tradicional para identificar un sistema es que el medio es usado como una unión sumadora en lugar de restar eléctricamente las señales. De cualquier forma para ser consistentes continuaremos representando el punto de suma como una resta, aunque en realidad esto es arbitrario ya que puede ser implementado con un cambio de signo en la señal secundaria.

De la Figura 3.7, la transformada Z de $e(n)$ denotada por $E(z)$ y expresada como

$$\begin{aligned} E(z) &= D(z) - Y(z) \\ &= P(z)X(z) - W(z)X(z) \end{aligned} \quad (3.1)$$

Idealmente, $E(z) = 0$ después de que el filtro $W(z)$ convergió. De la ecuación

3.1 tenemos que

$$W(z) = P(z) \quad (3.2)$$

para $X(z) \neq 0$ esto implica que

$$y(n) = d(n) \quad (3.3)$$

La ecuación 3.3 significa que la salida del filtro adaptativo $y(n)$ es idéntica a la perturbación primaria $d(n)$. De esta forma, cuando $d(n)$ y $y(n)$ son combinadas acústicamente, el ruido residual es

$$e(n) = d(n) - y(n) = 0 \quad (3.4)$$

que da lugar a una cancelación perfecta de ambos sonidos basados en el principio de superposición.

Así, si el modelo de $W(z)$ es una buena representación de la planta $P(z)$, la aproximación feedforward es capaz de cancelar ruido de banda ancha de una manera muy efectiva. Esto, por supuesto, asume que hay suficiente tiempo para que el sistema complete los cálculos del modelo, determine la señal inversa, y genere el sonido secundario requerido antes de que el ruido indeseado pase a través de la planta.

3.6 Efectos de la trayectoria secundaria.

La sección anterior describe el principio simplificado del ANC en un ambiente ideal, pero el uso de un filtro adaptativo como el mostrado en la Figura 3.7 para el sistema ANC de la Figura 3.6 se ve complicado por el hecho de que la señal de referencia eléctrica debe ser obtenida a través de la presión acústica usando un micrófono. También, una señal eléctrica de error debe ser obtenida del ruido acústico residual usando un micrófono de error. Finalmente el sonido cancelador debe ser producido de la señal eléctrica de salida a través de una bocina. De esta manera, un número de otras funciones de transferencia deben ser incluidas. El punto suma en la Figura 3.7 representa la superposición acústica que va de la bocina canceladora a el micrófono de error, donde el ruido primario se combina con la salida del filtro adaptativo. Por eso es necesario compensar la función de transferencia de la trayectoria secundaria que va de $y(n)$ a $e(n)$, que incluye el convertidor digital-analógico, el filtro reconstructivo, el amplificador, la bocina, y el camino acústico desde la bocina a el micrófono de error, el error propio del micrófono, preamplificador, filtro anti-aliasing y el convertidor digital analógico.

Como se ilustra en la Figura 3.8 la función de transferencia de la trayectoria secundaria $S(z)$, puede separarse en dos funciones de transferencia en cascada:

$$S(z) = R(z)S'(z), \quad (3.5)$$

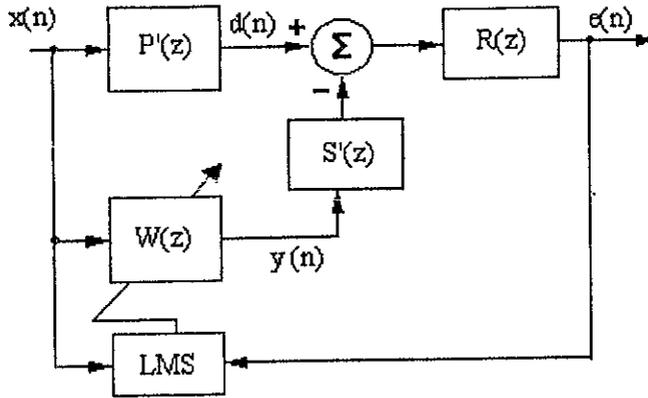


Figura 3.8: Diagrama a bloques de un sistema ANC.

donde $S'(z)$ representa la función de transferencia secundaria que va de la salida del filtro adaptativo a el punto suma y $R(z)$ representa la función de transferencia residual del punto suma a la señal de error. Para el ejemplo del ducto acústico, $S'(z)$ representa el sistema de la bocina, incluyendo al convertidor digital analógico, el filtro de reconstrucción, el amplificador, la bocina y la función de transferencia que va de la bocina a el punto suma, y $R(z)$ es la función de transferencia del sistema conformado por el micrófono, el preamplificador, el filtro antialiasing, el convertidor analógico digital. Similarmente, la función de transferencia primaria $P(z)$ puede descomponerse en

$$P(z) = R(z)P'(z), \quad (3.6)$$

donde $P'(z)$ es la función de transferencia de la planta acústica desconocida y que va del micrófono de referencia al punto suma. Los coeficientes del filtro adaptativo deben de converger a los valores que minimizan la señal de error a pesar de la presencia de las funciones de transferencia asociadas a los transductores y filtros.

De la Figura 3.8, la transformada z de la señal de error es

$$E(z) = R(z)[P'(z) - S'(z)W(z)]X(z) \quad (3.7)$$

Asumiendo que $W(z)$ tiene el orden suficiente. Entonces, después de que el filtro adaptativo ha convergido, el error residual se vuelve cero, esto es, $E(z) = 0$, lo que implica que $W(z)$ cumpla con la función de transferencia óptima

$$W^0(z) = \frac{P'(z)}{S'(z)} \quad (3.8)$$

en estado permanente. De esta forma, el filtro adaptativo tiene que invertir la función de transferencia de la trayectoria secundaria $S'(z)$. De cualquier

forma, es imposible compensar el retraso inherente ocasionado por $S'(z)$ si la trayectoria primaria $P'(z)$ no contiene un retraso de al menos igual tamaño. Esta es en general la restricción causal que limita a los sistemas feedforward de banda ancha. Y lo que es más, de la Ecuación 3.8, el sistema de control se vuelve inestable si existe una frecuencia ω tal que $S'(\omega) = 0$. Por ejemplo, en el problema acústico del ducto, si existe un cero en la función de transferencia del sistema de la bocina $S'(z)$, esto implica que existe una frecuencia que no se puede controlar.

De la Ecuación 3.7, el sistema de control no es efectivo si existe una frecuencia donde $R(\omega) = 0$; en otras palabras, un cero en la función de transferencia del sensor de error provoca que haya una frecuencia no observable. Aún si un filtro adaptativo estable es realizable bajo estas circunstancias, no es posible obtener una cancelación del ruido significativa a esta frecuencia: $P'(\omega) - S'(\omega)W(\omega)$ puede tomar cualquier valor, pero el controlador nunca sabrá si $R(\omega) = 0$. Obviamente, el ruido primario solo es controlable si el sistema es observable por medio del sensor de error.

De la Ecuación 3.8, el filtro adaptativo $W(z)$ realiza un modelado directo de la función primaria acústica $P'(z)$ para generar el ruido cancelador, y un modelado inverso de $S'(z)$ para compensar el efecto de la trayectoria secundaria. En términos del procesamiento de señales en el dominio del tiempo, la introducción de la función de transferencia de la trayectoria secundaria en un controlador que usa el algoritmo LMS ocasionará inestabilidad. Esto es debido a que la señal de error no está correctamente "alineada" en el tiempo con la señal de referencia, debido a la presencia de $S(z)$.

Dado que, tanto la función de transferencia primaria como la secundaria tienen a $R(z)$ en común, es conveniente absorber este término dentro del modelo, resultando en la figura equivalente mostrada en la Figura 3.9.

Observe que matemáticamente, la Figura 3.9, es exactamente equivalente a la Figura 3.8, la única diferencia es que se redefinió el punto suma de forma que ahora está en el dominio eléctrico en lugar del acústico.

De las Ecuaciones 3.5 - 3.7, la función de transferencia en estado permanente del filtro adaptativo puede ser expresada de una manera más sencilla. La transformada z de la señal de error es

$$E(z) = [P(z) - S(z)W(z)]X(z). \quad (3.9)$$

Después de que el filtro adaptativo ha convergido, el error residual es cero, lo que implica que $W(z)$ debe realizar la función óptima de transferencia

$$W^0(z) = \frac{P(z)}{S(z)}, \quad (3.10)$$

en otras palabras, el filtro adaptativo $W(z)$ tiene que modelar a $P(z)$ y modelar inversamente a $S(z)$. Conociendo las funciones de respuesta primaria y secundaria podemos calcular los pesos óptimos para minimizar el error cuadrático

medio en la superficie de desempeño. La respuesta del modelo actual es una función compleja que depende del contenido espectral de la fuente y de la planta acústica del sistema. Debido a la incertidumbre en las características de la planta acústica, no es práctico determinar un modelo exacto antes de que el sistema empiece a operar. El filtro $W(z)$ proporciona un medio completamente adaptativo para que simultáneamente modele la planta primaria $P(z)$ y la trayectoria secundaria $S(z)$ con una fuente dada para minimizar el ruido residual. Una ventaja clave en este tipo de acercamiento al problema es que con un modelo propio de la planta, el sistema puede responder instantáneamente a cambios en la señal de entrada ocasionados por cambios en las fuentes.

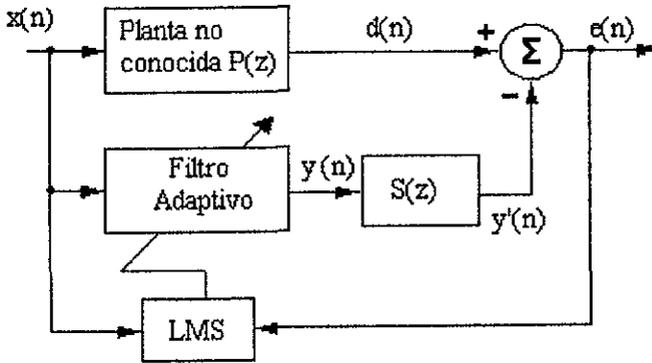


Figura 3.9: Diagrama a bloques simplificado de un sistema ANC.

3.7 El algoritmo FXLMS

Como vimos anteriormente, el algoritmo LMS convencional debe modificarse para asegurar su convergencia. Existen dos posibilidades para compensar el efecto de $S(z)$. La primera solución consiste en colocar un filtro inverso, $1/S(z)$, a continuación de $S(z)$ lo cual anulará su efecto, lamentablemente no siempre existe el inverso de $S(z)$ y por lo tanto se utiliza la segunda solución que consiste en colocar un filtro idéntico a $S(z)$ en la trayectoria de la señal de referencia hacia la actualización de los pesos del algoritmo LMS. Debido a que la señal de referencia es filtrada por $S(z)$, a este algoritmo se le conoce como algoritmo FXLMS.

Este algoritmo fué desarrollado por Widrow en 1981 dentro del contexto del control adaptativo y paralelamente Burgess lo desarrolló para aplicaciones de ANC.

3.7.1 Obtención del algoritmo FXLMS

De la Figura(3.9) se observa que $S(z)$ se localiza a continuación del filtro adaptativo $W(z)$ que está controlado por el algoritmo LMS, quedando expresada la señal de residuo por:

$$\begin{aligned} e(n) &= d(n) - y'(n) \\ &= d(n) - s(n) * y(n) \\ &= d(n) - s(n) * [w^T(n)x(n)] \end{aligned} \quad (3.11)$$

donde $s(n)$ es la respuesta al impulso de la trayectoria secundaria $S(z)$ en el tiempo n , $*$ denota convolución lineal.

$$w(n) = [w_0(n) \quad w_1(n) \quad \dots \quad w_{L-1}(n)]^T \quad (3.12)$$

es el vector de coeficientes de $W(z)$ en el tiempo n ,

$$x(n) = [x(n) \quad x(n-1) \quad \dots \quad x(n-L+1)]^T \quad (3.13)$$

es el vector de la señal de entrada en el tiempo n , y L es el orden del filtro $W(z)$. Para señales de entrada de banda ancha, $w(n)$ debe representar la respuesta al impulso de la función de transferencia de la ecuación 3.11, mientras que para señales de entrada de banda angosta, $w(n)$ debe cubrir una fracción substancial del periodo de la señal de entrada. Por lo que el número de coeficientes L del filtro debe ser elegido cuidadosamente para obtener una resolución suficiente en el tiempo para poder modelar la respuesta requerida.

El objetivo del filtro adaptativo es minimizar el error cuadrático instantáneo, $\xi(n) = e^2(n)$, para lo cual se utiliza el algoritmo LMS, el cual actualiza el vector de coeficientes en la dirección negativa del gradiente con un factor de convergencia μ .

$$w(n+1) = w(n) - \frac{\mu}{2} \nabla \hat{\xi}(n) \quad (3.14)$$

donde $\nabla \hat{\xi}(n)$ es una estimación instantánea del gradiente del error cuadrático medio (MSE) en el tiempo n , y que puede expresarse como

$$\nabla \hat{\xi}(n) = \nabla e^2(n) = 2[\nabla e(n)]e(n) \quad (3.15)$$

De la ecuación 3.11, tenemos que

$$\nabla e(n) = -s(n) * x(n) = -x'(n) \quad (3.16)$$

donde

$$x'(n) = [x'(n) \quad x'(n-1) \quad \dots \quad x'(n-L+1)]^T \quad (3.17)$$

y

$$x'(n) = s(n) * x(n). \quad (3.18)$$

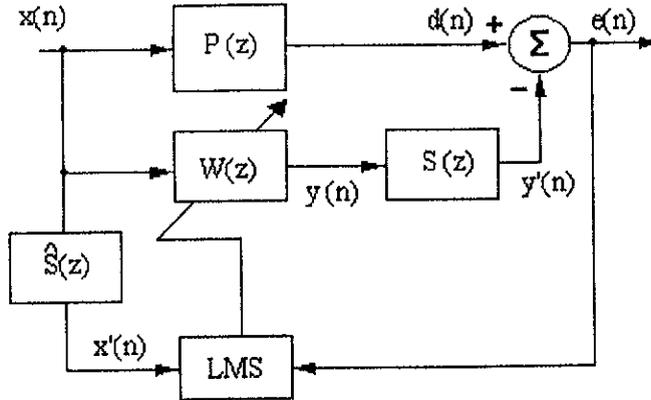


Figura 3.10: Diagrama a bloques de un sistema ANC utilizando el algoritmo FXLMS.

De esta forma el gradiente se convierte en

$$\nabla \hat{\xi}(n) = -2\mathbf{x}'(n)e(n) \quad (3.19)$$

Substituyendo la ecuación 3.19 en la ecuación 3.14, obtenemos el algoritmo FXLMS

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}'(n)e(n) \quad (3.20)$$

donde

$$\mathbf{x}' = \hat{s} * \mathbf{x}(n) \quad (3.21)$$

El resultado muestra que cuando una función de transferencia $S(z)$ sigue a un filtro adaptativo, esta misma función de transferencia debe ser colocada en la trayectoria de la actualización de los pesos, con lo cual el vector de entrada $\mathbf{x}(n)$ es filtrado por $S(z)$. Vea la Figura 3.10.

Al implementarse este algoritmo, se ha logrado que la convergencia del filtro sea mucho más rápida de lo que la teoría indica, además se ha demostrado que el algoritmo es más tolerante con el error producido en la estimación de la función de transferencia $S(z)$ mediante el filtro $\hat{S}(z)$.

3.7.2 Técnicas de Modelado Off-Line

El algoritmo FXLMS requiere del conocimiento de la función de transferencia $S(z)$ que es variante con el tiempo y desconocida debido a efectos tales como el deterioro de la bocina con el tiempo, cambios en la temperatura, etc.. Asumiendo que las características de $S(z)$ son invariantes con el tiempo pero desconocidas, pueden utilizarse las técnicas Off-Óline para estimar $S(z)$ durante

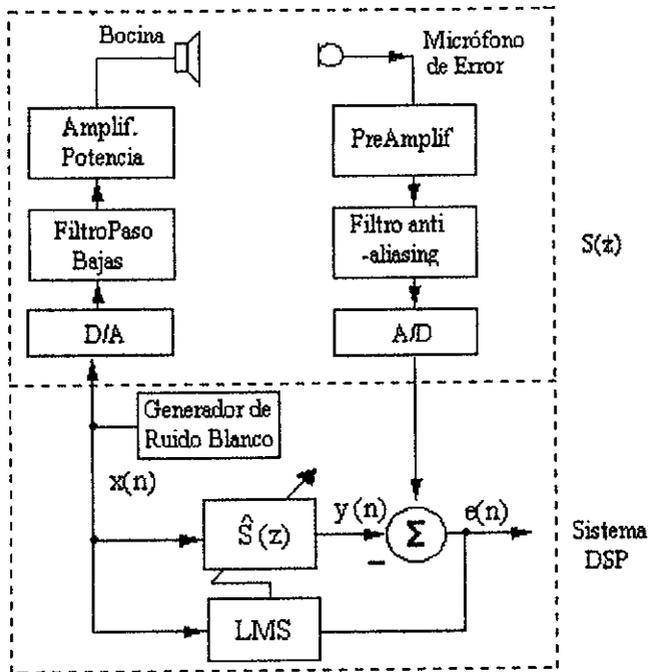


Figura 3.11: Arreglo experimental para modelar la trayectoria secundaria fuera de línea.

un período inicial de entrenamiento. El ruido blanco es una buena señal de banda ancha para dicho entrenamiento.

La configuración para implementar la técnica de modelado Off-Line se muestra en la Figura 3.11, donde un ruido blanco no correlacionado es generado internamente por el DSP, esta señal sirve de entrada a una fuente secundaria, en este caso una bocina, y al mismo tiempo alimenta un filtro adaptativo $\hat{S}(z)$ que es utilizado para modelar la trayectoria secundaria. El proceso es el siguiente:

1. Generar la señal muestreada de ruido blanco $x(n)$.
2. Obtener la señal deseada $d(n)$ del sensor de error.
3. Aplicar el algoritmo adaptativo de la siguiente forma:
 - a) Obtener la salida del filtro adaptativo

$$y(n) = \sum_{l=0}^{L-1} \hat{s}_l(n)x(n-l)$$

donde $\hat{s}_l(n)$ es el l avo coeficiente del filtro $\hat{S}_l(z)$ que estima la trayectoria secundaria en un tiempo n .

b) Calcular la señal de error

$$e(n) = d(n) - y(n)$$

c) Actualizar los coeficientes usando el algoritmo LMS tradicional

$$\hat{s}_l(n+1) = \hat{s}_l(n) + \mu x(n-l)e(n)$$

donde $l = 0, 1, \dots, L-1$

4. Ir al paso número uno para realizar la siguiente iteración hasta que el filtro adaptativo $\hat{S}(z)$ converja hacia la solución óptima, es decir, hasta que la potencia del error sea minimizada.

Después de que el algoritmo converja, se detiene la adaptación y los coeficientes del vector $\hat{s}(n)$ son utilizados en el sistema ANC.

3.8 Sistemas ANC feedback monocanal

Un sistema ANC clásico retroalimentado se muestra en la siguiente figura:

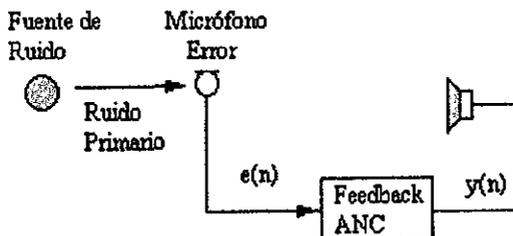


Figura 3.12: Sistema ANC mono canal retroalimentado.

Este tipo de sistema crea una zona de silencio usando el control retroalimentado de una fuente secundaria localizada en la vecindad de un sensor de error. La salida del sensor es procesada por un amplificador que tiene una ganancia total mayor que la unidad y un cambiador de fase a 180 grados. El sistema requiere solamente un sensor, evitando de esta forma el problema de la retroalimentación de la fuente secundaria al sensor de referencia que existe en los sistemas feedforward.

Esta técnica puede ser vista como un sistema adaptativo feedforward que, en efecto, sintetiza, o regenera, su propia señal de referencia, basado solamente en la salida del filtro adaptativo y la señal de error. Bajo ciertas condiciones, los sistemas ANC adaptativos retroalimentados pueden interpretarse como un predictor adaptativo.

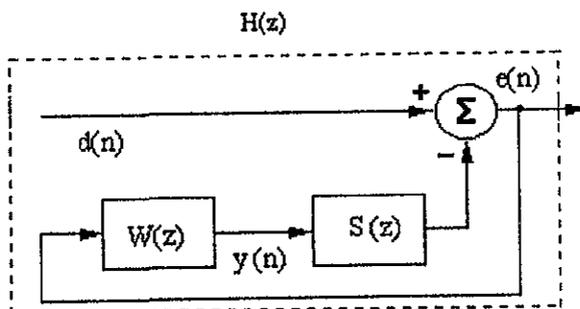


Figura 3.13: Diagrama de bloques de un sistema ANC monocanal retroalimentado.

3.8.1 Algoritmo y estructura.

En la aplicación de la Figura 3.13, el ruido primario $d(n)$ no está disponible durante la operación del sistema ANC porque se supone que está siendo cancelado por la fuente secundaria. De esta forma, la idea básica de un sistema ANC adaptativo retroalimentado es estimar el ruido primario y usarlo como una señal de referencia $x(n)$ para el filtro ANC, $W(z)$.

En la Figura 3.13, el ruido puede ser expresado en el dominio de z como

$$D(z) = E(z) + S(z)Y(z), \quad (3.22)$$

donde $E(z)$ es la señal obtenida del sensor de error y $Y(z)$ es la señal secundaria generada por el filtro adaptativo. Así, tanto $E(z)$ como $Y(z)$ están disponibles para el sistema ANC retroalimentado. Si la función de transferencia de la trayectoria secundaria $S(z)$ es también medible y aproximada por $\hat{S}(z)$, específicamente, $\hat{S}(z) = S(z)$, podemos estimar el ruido primario $d(n)$ y usar éste como una señal de referencia sintetizada $x(n)$:

$$X(z) \equiv \hat{D}(z) = E(z) + \hat{S}(z)Y(z), \quad (3.23)$$

Esta técnica de síntesis (o regeneración) se ilustra en la Figura 3.14, donde la señal secundaria $y(n)$ es filtrada por el estimado de la trayectoria secundaria $\hat{S}(z)$ y sumado con $e(n)$ para regenerar el ruido primario.

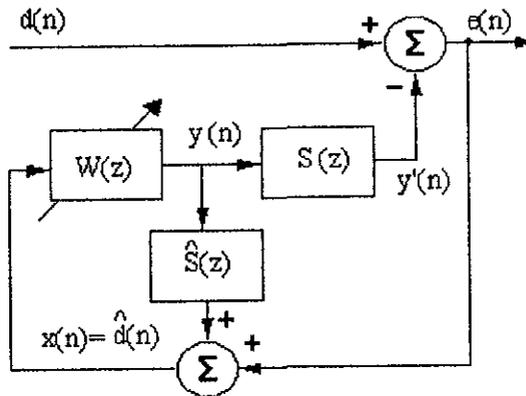


Figura 3.14: Uso de la señal de referencia sintetizada a partir de las señales disponibles $y(n)$ y $e(n)$.

El sistema ANC retroalimentado completo que hace uso del algoritmo FXLMS se ilustra en la Figura 3.15, donde $\hat{S}(z)$ es también requerida para compensar la trayectoria secundaria. La señal de referencia $x(n)$ es sintetizada como un

estimado de $d(n)$, que se expresa como

$$x(n) \equiv \hat{d}(n) = e(n) + \sum_{m=0}^{M-1} \hat{s}_m y(n-m) \quad (3.24)$$

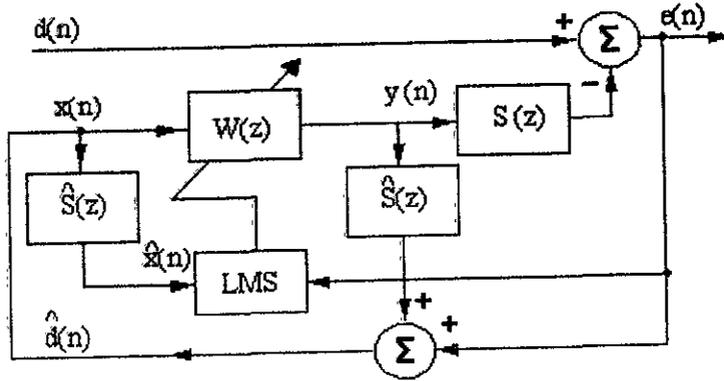


Figura 3.15: Sistema ANC retroalimentado haciendo uso del algoritmo FXLMS.

donde $\hat{s}_m, m = 0, 1, \dots, M-1$ son los coeficientes del filtro $\hat{S}(z)$ FIR de orden M utilizado para estimar la trayectoria secundaria. La estimación de $S(z)$ puede realizarse ya sea en línea o fuera de ella. La señal secundaria es generada como

$$y(n) = \sum_{l=0}^{L-1} w_l(n)x(n-l) \quad (3.25)$$

donde $w_l(n), l = 0, 1, \dots, L-1$ son los coeficientes del filtro $W(z)$ en el tiempo n , y L es el orden del filtro FIR $W(z)$. Estos coeficientes son actualizados por el algoritmo FXLMS de la siguiente manera

$$w_l(n+1) = w_l(n) + \mu x'(n-l)e(n) \quad l = 0, 1, \dots, L-1 \quad (3.26)$$

donde μ es el factor de convergencia, y

$$x'(n) = \sum_{m=0}^{M-1} \hat{s}_m x(n-m) \quad (3.27)$$

es la señal filtrada de referencia.

3.9 Evaluación del desempeño

Los sistemas ANC aire-acústicos son más útiles para sonidos de baja frecuencia donde las técnicas pasivas son menos efectivas, generalmente debajo de los 500Hz. Los sonidos tonales o de banda estrecha pueden ser típicamente reducidos 30dB o más, mientras que los de banda ancha o las perturbaciones aleatorias pueden ser reducidos entre 15 y 20 dB. Los sonidos tonales o de banda estrecha generalmente son reducidos a el nivel del ruido de fondo, resultando en un espectro menos molesto como el que se determina usando criterios de ruido recomendados. Comentarios similares se aplican al ANC vibratorio e hidroacústico, que también tienden a ser más efectivos a bajas frecuencias donde la densidad modal es baja.

Una de las formas en que se valua el desempeño de los sistemas ANC adaptivos involucra una jerarquía de técnicas, que comienza con el problema simplificado en forma ideal y agrega progresivamente restricciones y otras complicaciones. El análisis del desempeño involucra los siguientes puntos: (1) las limitaciones fundamentales del desempeño, (2) las restricciones prácticas que limitan el desempeño, (3) como el desempeño es balanceado con la complejidad, y (4) como determinar una arquitectura de diseño práctica. Para llevar a cabo estos puntos, se han definido 4 niveles de desempeño:

Nivel I—deriva los límites fundamentales de desempeño por medio de mediciones continuas sobre toda la función de desempeño

Nivel II—trae consigo las restricciones prácticas de un número fijo de sensores en locaciones discretas

Nivel III—incorpora el conocimiento de la estructura de la función de transferencia

Nivel IV—agrega los efectos prácticos restantes y restricciones de diseño requeridos para los cálculos detallados del desempeño

En cada paso, un grado de confianza es alcanzado y un estándar de medición es establecido para poder comparar y revisarlo con el siguiente nivel de complejidad. La jerarquía analítica va de lo simple a lo complicado como se muestra en la Tabla 3.9.

El principio de ANC es simple. De cualquier forma, cuando es llevado a aplicaciones reales, surgen muchos problemas prácticos. Por ejemplo, en ANC aire-acústico, algunas de las preguntas que necesitan ser contestadas dentro del procedimiento jerárquico son

- Dónde se deben colocar las bocinas y los micrófonos? (Niveles I y II.)
- Cómo se puede reducir el ruido del flujo de aire en los micrófonos? (Niveles II y IV.)

- Qué algoritmo debe adoptarse? (Niveles II y IV.)
- Cómo puede incrementarse la potencia de las bocinas? (Nivel IV.)
- Cómo se puede incrementar el tiempo de vida útil de los micrófonos y las bocinas? (nivel IV)
- Cómo puede reducirse el costo del controlador? (Nivel IV.)

Tabla 3.1: Jerarquía de Análisis para el desempeño ANC.

Simple	Complicado
Banda estrecha	Banda ancha
Fuente primaria estacionaria	Fuente primaria dinámica
Una sola fuente secundaria	Fuentes secundarias múltiples
Mediciones sin ruido	Mediciones con ruido
Sensor de campo continuo	Sensor de campo discreto

La meta de ANC puede ser reducir el sonido a un mínimo valor y/o alcanzar ciertas características espectrales. Por lo tanto en la aplicación final, se vuelve necesario inculcar aspectos de psicología acústica, que están relacionados con el carácter del espectro residual. La forma espectral de este ruido residual puede posiblemente ser optimizada para igualar algunas curvas de ponderación de ruido, con esto podemos observar que tan subjetivo de objetar es el ruido residual.

Con el objeto de ser adecuado para aplicaciones industriales, un sistema ANC debe tener ciertas propiedades:

1. Máximo de eficiencia sobre la banda de frecuencia más grande posible para cancelar un gran rango de ruido.
2. Autonomía con respecto a la instalación, de forma que el sistema pueda ser construido y calibrado en el área de manufactura y después insertado en el sitio deseado.
3. Auto-adaptatividad con el objeto de poder manejar cualquier variación en los parámetros físicos (temperatura, velocidad del flujo de aire, etcétera.)
4. Robustez y confiabilidad de los diferentes elementos del sistema y una electrónica de control simplificada

El continuo progreso de los sistemas ANC involucra el desarrollo de mejores algoritmos adaptivos, transductores y hardware para el procesamiento digital de señales.

Capítulo 4

El TMS320C50

4.1 Surgimiento del TMS320

El proceso de diseño de un filtro digital o de cualquier sistema de procesamiento digital de señales involucra varias fases, desde la concepción inicial del sistema hasta su realización final. Anteriormente, el proceso de diseño de los filtros digitales terminaba con la obtención de la función de transferencia $H(z)$ o la respuesta al impulso. El resto del problema que consistía en la realización física del sistema se dejaba a gente más especializada o al futuro, que a su debido tiempo dotaría de nuevas tecnologías para la implementación en Hardware.

Durante el rápido crecimiento de las teorías de diseño de filtros digitales también avanzó la tecnología de los circuitos integrados, lográndose pasar de la integración a grande escala (LSI) a la integración a muy grande escala (VLSI), y con esto al desarrollo de los microprocesadores. Estos ofrecieron las primeras alternativas reales a bajo costo para completar el proceso de diseño de filtros digitales (la implementación).

El microprocesador al ser combinado con un sistema operativo funcional permitía llegar al resultado deseado, sin embargo debido a que el procesamiento digital de señales inherentemente maneja grandes cantidades de operaciones aritméticas así como numerosos datos de entrada y salida, los microprocesadores de propósito general no tenían la velocidad suficiente para cumplir con estos requisitos, por lo tanto se desarrollaron los microprocesadores especiales para el procesamiento digital de señales, los DSPs.

La familia TMS320 de Texas Instruments está integrada por microprocesadores implementados en tecnología CMOS. Dicha familia consta de dos tipos principales de DSPs: los de punto fijo de 16 bits y los de punto flotante de 32 bits. Estos DSPs combinan la flexibilidad de operación de los controladores de alta velocidad con la capacidad numérica de los arreglos de procesadores resultando en una gran versatilidad para innumerables aplicaciones en distintos campos,

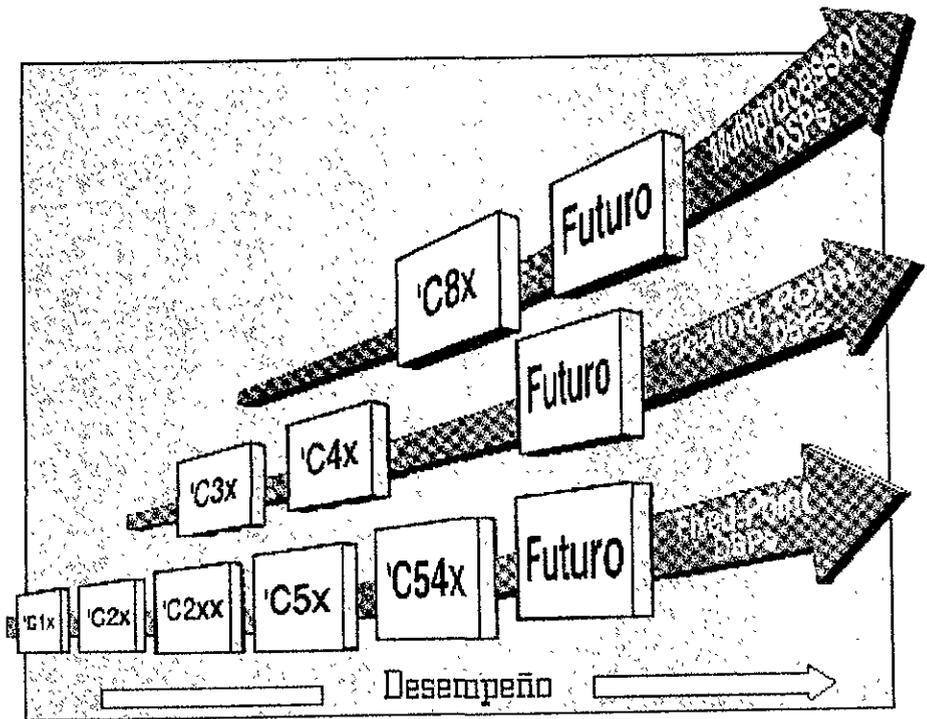


Figura 4.1: Generaciones de DSPs.

tales como la medicina, la instrumentación, el control, las telecomunicaciones, la industria, la computación, etc.

Entre las características más representativas que hacen de los DSPs una opción viable para el procesamiento digital de señales se tienen:

- Un set de instrucciones muy completo y flexible
- Flexibilidad de operación
- Gran desempeño a una gran velocidad
- Relación costo-efectividad muy favorable

Hasta el día de hoy, la familia TMS320 abarca ocho generaciones de DSPs: C1x, C2x, C2xx, C5x, C54x (DSPs de punto fijo). C3x y C4x (DSPs de punto flotante) y el C8x (multiprocesador).

Como mencionamos anteriormente, el procesamiento digital de señales maneja señales en tiempo real, y es necesario que el DSP realice operaciones aritméticas a gran velocidad, por esto, la familia TMS320 está basada en la

arquitectura Harvard, la cual se caracteriza por tener buses separados para los datos y el programa, lo que permite el direccionamiento simultáneo de instrucciones de programa y de datos.

4.2 Quinta Generación, el DSP TMS320C5x

Para lograr un máximo desempeño el TMS320C5x utiliza una arquitectura Harvard modificada. Estas modificaciones consisten en mejoras a la arquitectura Harvard añadiéndole algunas características propias de la arquitectura Von Neumann, tales como la habilidad de inicializar la memoria de datos desde la memoria de programa, y la habilidad de transferir memoria dato a memoria programa. Estas propiedades permiten al C5x multiplexar en tiempo su memoria de modo que además de la realización de tareas pueda inicializar su memoria dato (por ejemplo leer coeficientes almacenados en su memoria ROM). Lo anterior permite minimizar el costo del sistema, ya que se elimina el uso de memorias ROM externas y se maximiza la utilización de la memoria dato al redefinir dinámicamente la función de ésta. Aunado a este tipo de arquitectura, la generación C5x incluye un CPU con aplicaciones específicas de hardware, periféricos y memoria a bordo y un set de instrucciones altamente especializado. El resultado de la interacción de todos estos elementos es la capacidad del C5x de ejecutar hasta 40 millones de operaciones por segundo (40 MIPS).

El Hardware interno del C5x ejecuta funciones que otros procesadores implementan en software o microcódigo, por ejemplo, el multiplicador por hardware del C5x ejecuta multiplicaciones de 16 x 16 bits en un solo ciclo de máquina, corrimientos y direccionamientos.

Los dispositivos de la familia C5x difieren unos de otros por la cantidad y tipo de memoria que poseen. Otras diferencias son el tipo de empaquetamiento y el voltaje de operación

El TMS320C50 es un procesador de punto fijo de 16 bits que forma parte de la quinta generación de la familia TMS320 y tiene las siguientes características:

Compatibilidad: Su código fuente es compatible con el de las generaciones C1x, C2x y C2xx.

Velocidad: El tiempo de ejecución de un ciclo de máquina es de 25/35/50 ns. Lo que significa tener 40/28.6/20 MIPS.

Alimentación: 5 Volts con dos modos de bajo consumo (power down).

Encapsulado: Tipo bumpered quad flat-pack (BQFP) de 132 pines.

Memoria: El C50 tiene un rango total de direccionamiento de 224K x 16 bits. El espacio de memoria se divide en 4 regiones:

- 64K de memoria programa

TMS320 Device	ID	On-Chip Memory (16-bit words)			I/O Ports		Power Supply (V)	Cycle Time (ns)	Package Type
		DARAM†	SARAM‡	ROM	Serial	Parallel ∅			
‘C50	PQ	1056	9K	2K\$	2¶	64K	5	50/35/25	132 pin BQFP™
‘LC50	PQ	1056	9K	2K\$	2¶	64K	3.3	50/40/25	132 pin BQFP™
‘C51	PQ	1056	1K	8K\$	2¶	64K	5	50/35/25/20	132 pin BQFP™
‘C51	PZ	1056	1K	8K\$	2¶	64K	5	50/35/25/20	100 pin TQFP™
‘LC51	PQ	1056	1K	8K\$	2¶	64K	3.3	50/40/25	132 pin BQFP™
‘LC51	PZ	1056	1K	8K\$	2¶	64K	3.3	50/40/25	100 pin TQFP™
‘C52	PJ	1056	—	4K\$	1	64K	5	50/35/25/20	100 pin QFP™
‘C52	PZ	1056	—	4K\$	1	64K	5	50/35/25/20	100 pin TQFP™
‘LC52	PJ	1056	—	4K\$	1	64K	3.3	50/40/25	100 pin QFP™
‘LC52	PZ	1056	—	4K\$	1	64K	3.3	50/40/25	100 pin TQFP™
‘C53	PQ	1056	3K	16K\$	2¶	64K	5	50/35/25	132 pin BQFP™
‘C53S	PZ	1056	3K	16K\$	2	64K	5	50/35/25	100 pin TQFP™
‘LC53	PQ	1056	3K	16K\$	2¶	64K	3.3	50/40/25	132 pin BQFP™
‘LC53S	PZ	1056	3K	16K\$	2	64K	3.3	50/40/25	100 pin TQFP™
‘LC56	PZ	1056	6K	32K	2#	64K	3.3	50/35/25	100 pin TQFP™
‘C57S	PGE	1056	6K	2K\$	2#	64K	5	50/35/25	144 pin TQFPΔ
‘LC57	PBK	1056	6K	32K	2#	64K	3.3	50/35/25	128 pin TQFP™
‘LC57S	PGE	1056	6K	2K\$	2#	64K	3.3	50/35	144 pin TQFPΔ

† Dual-access RAM (DARAM)

‡ Single-access RAM (SARAM)

\$ ROM bootloader available

¶ Includes time-division multiplexed (TDM) serial port

Includes buffered serial port (BSP)

|| Includes host port interface (HPI)

™ 20 × 20 × 3.8 mm bumped quad flat-pack (BQFP) package

™ 14 × 14 × 1.4 mm thin quad flat-pack (TQFP) package

™ 14 × 20 × 2.7 mm quad flat-pack (QFP) package

Δ 20 × 20 × 1.4 mm thin quad flat-pack (TQFP) package

∅ Sixteen of the 64K parallel I/O ports are memory mapped.

Figura 4.2: Tipos de C5x.

- 64K de memoria dato
- 64K de memoria de puertos
- 32K de memoria global

Unidad Central de procesos (CPU)

Unidad aritmética lógica central (CALU)

Unidad lógica paralela (PLU)

Unidad aritmética para los registros auxiliares (ARAU).

Control de Programa El cual contiene:

- Un stack o pila por hardware con 8 niveles.
- Una operación de pipeline de 4 niveles
- Once registros "sombra" para almacenar los registros de control estratégicos del CPU durante una rutina de interrupción
- Operación de espera para la atención de accesos directos a memoria (DMA).
- Dos buffers de direccionamiento indirecto para direccionamiento circular

Set de instrucciones .

- Instrucciones de multiplica/acumula en un solo ciclo de máquina.
- Instrucciones simples para realizar operaciones de repetición y repetición de bloques.
- Instrucciones para el movimiento de bloques entre la memoria de datos y la memoria programa.
- Instrucciones para almacenar y cargar registros mapeados.
- Modo de direccionamiento indexado.
- Modo de direccionamiento indexado con bit reverso para efectuar FFTs de radical 2.

Periféricos a bordo .

- 64K de puertos paralelos de entrada/salida, con 16 puertos mapeados en memoria.
- 16 estados de espera programables mediante software para los espacios de memoria programa,dato o de entrada/salida.
- Temporizador

- Generador interno de reloj que también puede funcionar como reloj externo.
- Puerto serial síncrono Full-duplex para comunicarse directamente con otros C5x y/u otros dispositivos seriales.
- Puerto serial de múltiple acceso por división de tiempo (TDM).
- Interrupciones mascarables para el usuario.

4.3 Arquitectura

La arquitectura interna del C50 tiene como objetivo poder procesar grandes volúmenes de operaciones aritméticas y lógicas en el menor tiempo posible, es por ello que debe tener un alto grado de paralelismo y poseer un gran rango de memoria interna. El TMS320C50 está formado por 4 bloques principales:

- Los Buses
- La memoria interna
- La unidad central de procesos, y
- Los periféricos a bordo.

4.3.1 Estructura de los Buses

Buena parte del gran desempeño del C50 es debido a la configuración interna de sus buses y en consecuencia al alto grado de paralelismo resultante. Los 2 buses principales son:

- Bus de programa y
- Bus de datos

El bus de programa se encarga de direccionar el código y de leer los operandos, y el bus de datos se encarga de leer operandos y escribir los resultados. El paralelismo permite que se realicen operaciones separadas simultáneamente:

- Realizar operaciones matemáticas en la CALU ó
- Realizar operaciones lógicas en la PLU +
- Realizar operaciones aritméticas en el ARAU +
- Direccionar una nueva instrucción en la memoria programa

Por ejemplo, mientras un dato es multiplicado, el producto previo puede ser cargado, sumado o restado del acumulador y, al mismo tiempo, puede generarse una nueva dirección.

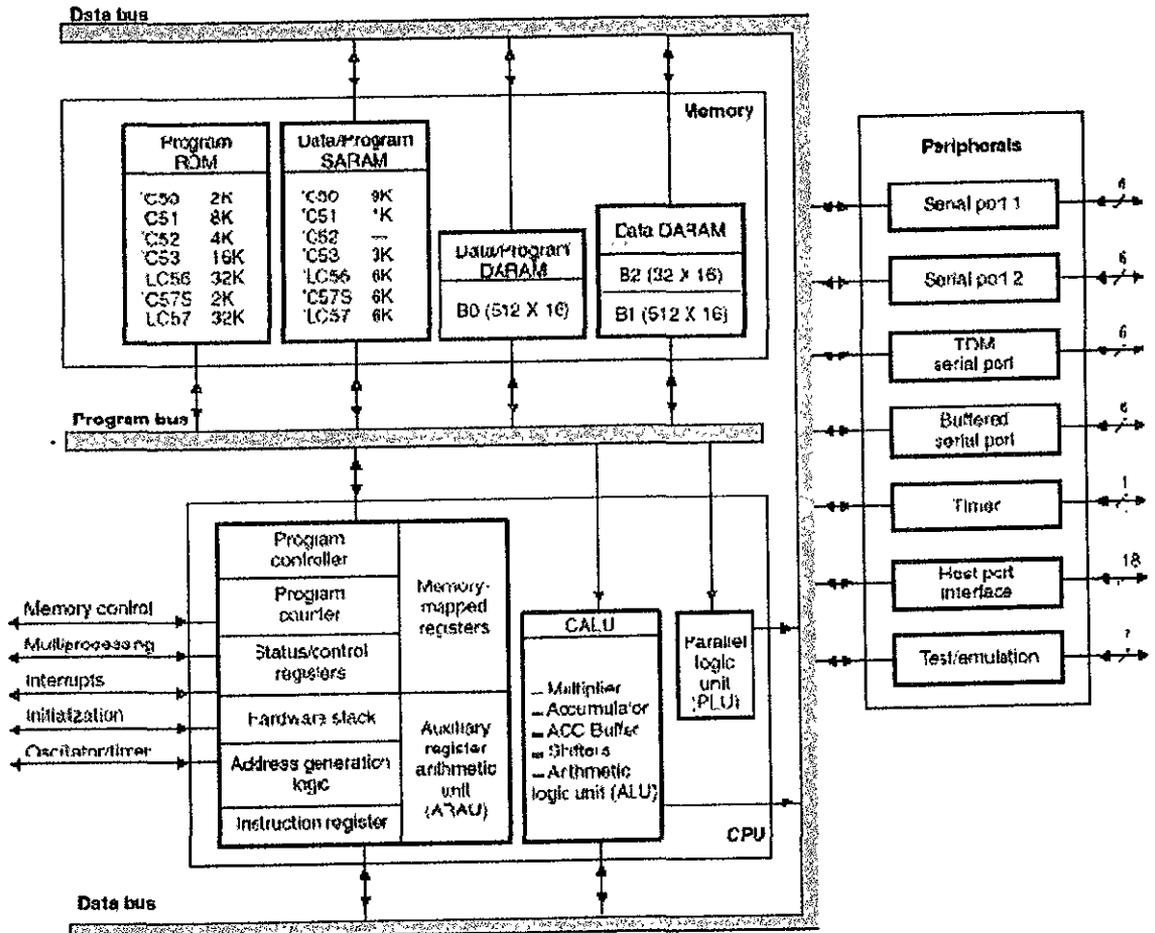


Figura 4.3: Arquitectura del C50.

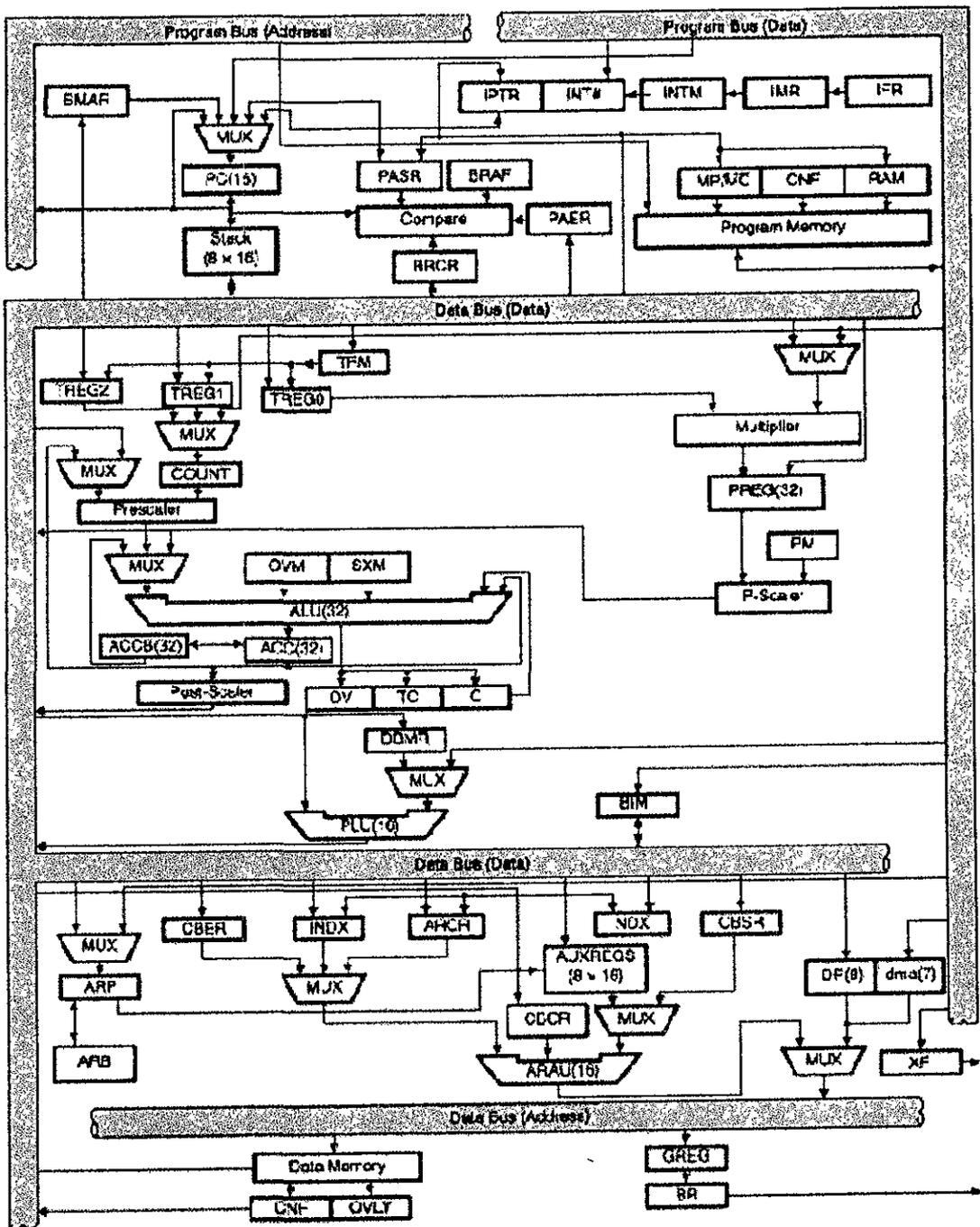


Figura 4.4: Buses del C50.

4.3.2 Memoria

El C50 cuenta con una considerable cantidad de memoria interna para un buen desempeño dentro de los sistemas a los que se vaya a integrar:

- Memoria programa de solo acceso (ROM).
- Memoria dato/programa de doble acceso (DARAM).
- Memoria dato/programa de acceso simple (SARAM).

El C50 tiene 2K x 16 bits de memoria mascarable y programable ROM localizada en el espacio de memoria programa en el rango de dirección de 0000-07FFh. Esta memoria se utiliza para cargar el programa de inicialización desde memorias lentas externas tales como ROMs o EPROMs hacia espacios de memoria más rápidos como memoria interna o externa RAM. Una vez cargado el programa en RAM, el espacio de la memoria ROM puede ser removido del espacio de memoria programa colocando el bit MP/MC en el registro de estado del modo del procesador (PMST).

El C50 tiene 1056 palabras x 16 bits de memoria DARAM. Se dice que la DARAM es de doble acceso ya que si se tiene una secuencia de instrucciones en donde la primera instrucción sea una escritura y la segunda una lectura de datos entonces el CPU será capaz de leer desde y escribir hacia la DARAM en el mismo ciclo de máquina.

La DARAM está dividida en tres bloques individuales:

- Bloque B0, 512 palabras localizables en memoria dato o programa. Su configuración es por software alterando el bit 12 del registro de estado ST1.
- Bloque B1, 512 palabras en memoria dato.
- Bloque B2, 32 palabras en memoria dato.

En el C5x la SARAM ocupa 9K x 16 bits. En esta memoria puede cargarse el código de las ROMs externas para ejecutarlo posteriormente a velocidad máxima. Después de una reinicialización, esta memoria debe de configurarse mediante software para que se localice dentro del espacio de dato o programa. La SARAM puede configurarse de 3 maneras:

- Toda la SARAM en memoria dato.
- Toda la SARAM en memoria programa.
- La SARAM presente en ambas memorias simultáneamente.

Al situarse la SARAM, esta se divide en bloques contiguos de palabras de 1K o 2K. Estos bloques de SARAM soportan el acceso dual o paralelo, con la condición de que cada acceso sea hacia un bloque distinto.

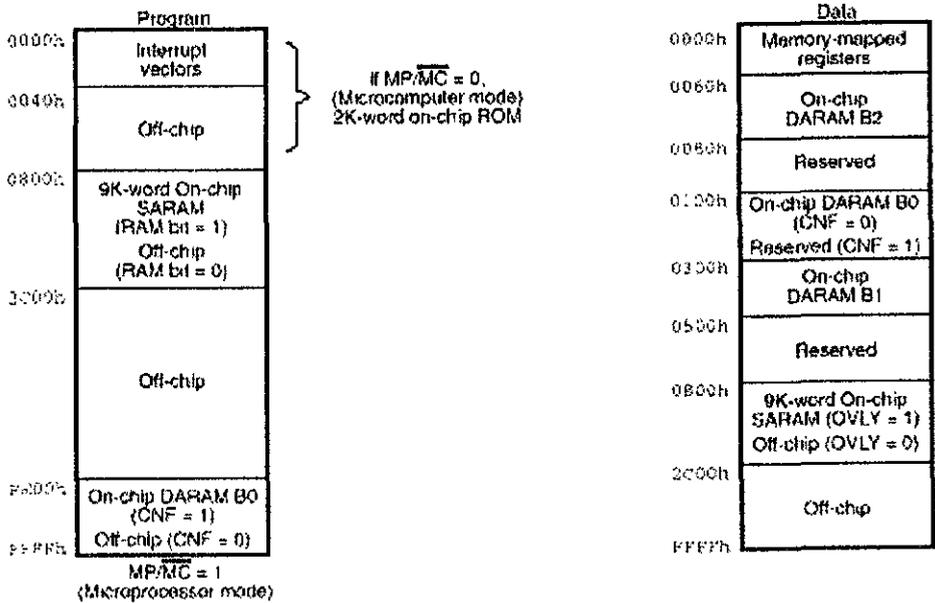


Figura 4.5: Memoria del C50.

4.3.3 Unidad central de procesos (CPU)

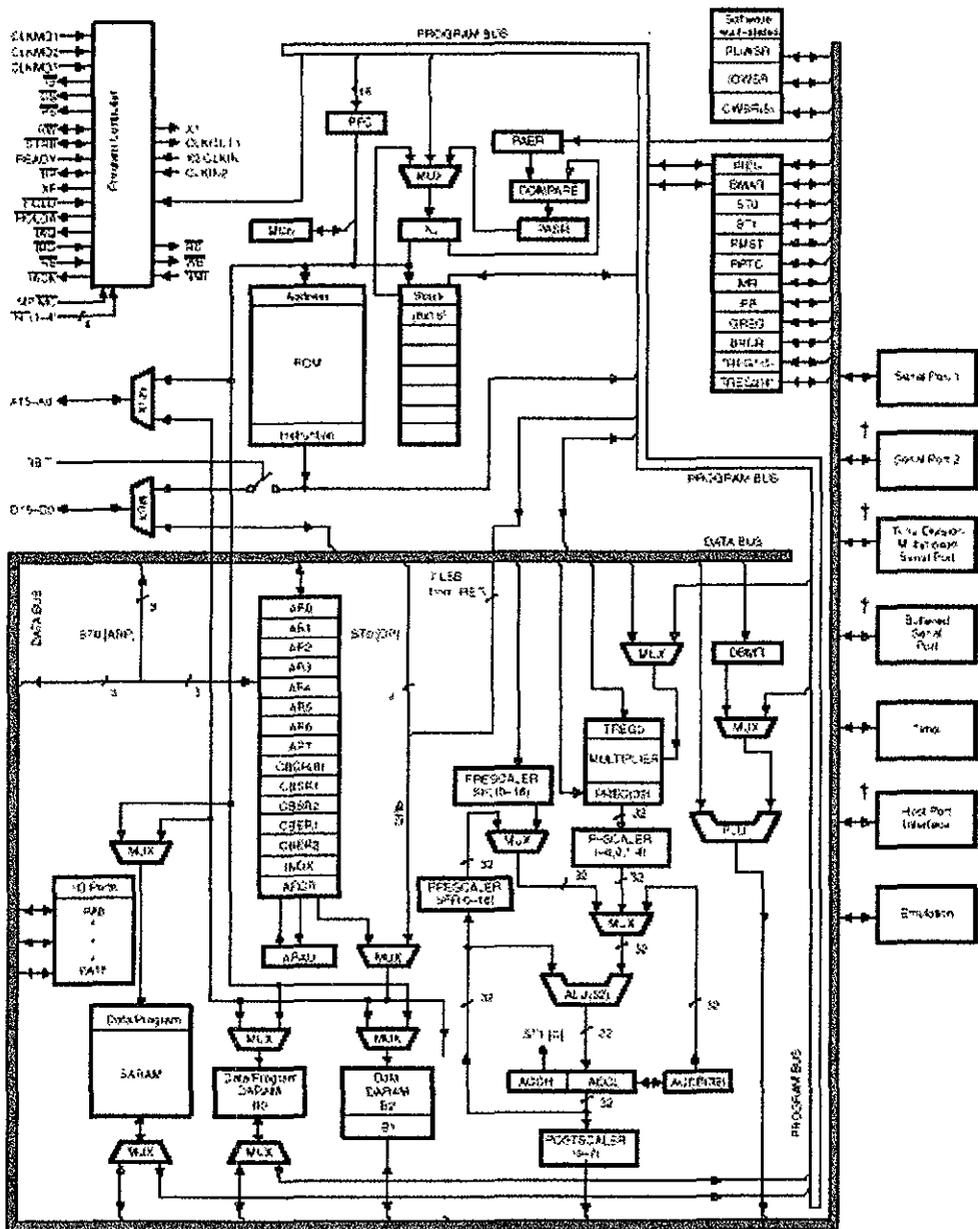
La unidad central de procesos del C50 está constituida por la unidad aritmética lógica central (CALU), la unidad lógica paralela de 16 bits (PLU), los registros mapeados en memoria y el controlador de programa.

Algunas funciones de control del CPU incluyen la capacidad de realizar escalamientos, y la administración de la memoria de datos a través del uso de instrucciones que permiten mover bloques de datos e instrucciones que permiten manejar los registros mapeados en memoria.

4.3.4 CALU

El CPU utiliza a la CALU para realizar aritmética de complemento a 2. Dentro de la CALU se localizan:

- Unidad aritmética lógica de 32 bits (ALU)
- Un acumulador de 32 bits (ACC) y un buffer para el acumulador de 32 bits (ACCB).
- Un multiplicador en paralelo de 16 x 16 bits que tiene un registro para almacenar el producto con capacidad de 32 bits.
- 4 registros de corrimiento:



Notes: All registers and data lines are 16-bits wide unless otherwise specified.
 † Not available on all devices

Figura 4.6: CPU del C50.

- Un registro de corrimiento para desplazar 0-,1- o 4- bits hacia la izquierda o 6 bits a la derecha.
- Un registro de corrimiento para desplazar de 0- a 16- bits hacia la izquierda.
- Un registro de corrimiento para desplazar de 0- a 16- bits hacia la derecha.
- Un registro de corrimiento para desplazar de 0- a 7- bits hacia la izquierda.

La ALU y el acumulador realizan una gran cantidad de funciones aritméticas y lógicas, la mayoría de éstas en un solo ciclo de máquina. Una vez que la operación ha sido realizada en la ALU, el resultado es transferido al acumulador, donde pueden ocurrir operaciones adicionales tales como corrimientos.

El acumulador (ACC) y el buffer del acumulador (ACCB) junto con un nuevo conjunto de instrucciones (LACB,SACB,ADDB,SSB,EXAR,CRGT,CRLT) permiten mover desde o hacia el acumulador 32 bits en un solo ciclo de máquina.

Una instrucción de la ALU se realiza de la siguiente manera:

1. Se lee el dato desde la RAM en el bus de datos.
2. El dato pasa a través del registro de corrimiento y de la ALU, donde se realiza la operación.
3. El resultado se almacena en el acumulador

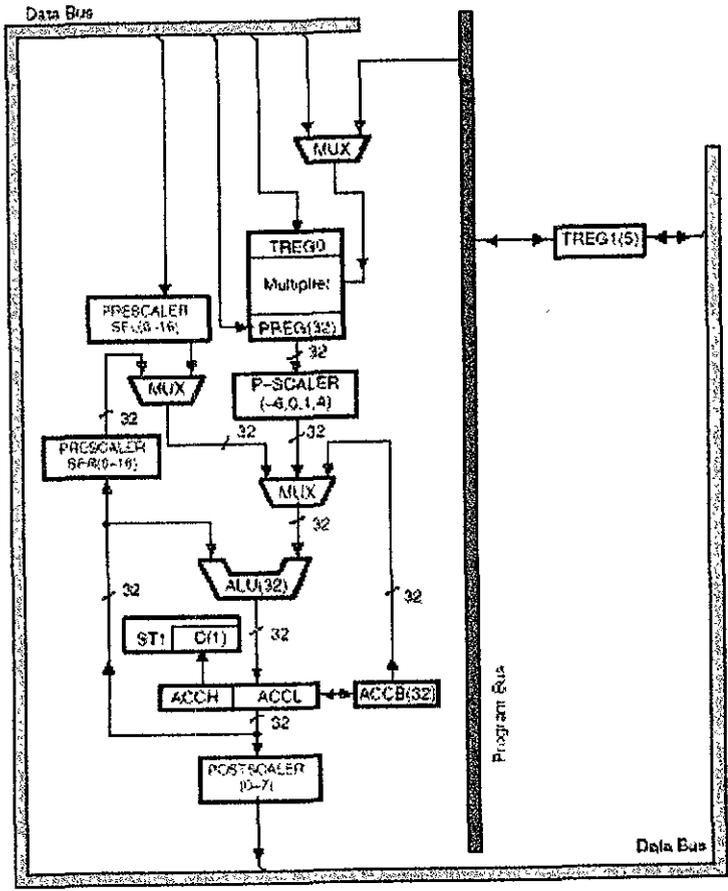
A la salida del acumulador se encuentra el registro de corrimiento post-escalador el cual permite realizar corrimientos de 0 a 7 bits hacia la izquierda. Este corrimiento se ejecuta mientras la información está siendo transferida al bus de datos o almacenada, por lo que el contenido del acumulador permanece como estaba.

Una de las entradas de la ALU siempre proviene del acumulador. La otra puede ser transferida desde el registro producto (P), del ACCB o de la salida del registro de corrimiento pre-escalador(que ha sido cargado desde la memoria de datos o el acumulador).

Multiplicador

El Hardware del multiplicador está diseñado para realizar multiplicaciones signadas con dos valores de 16 bits en un solo ciclo de máquina, es decir, realiza aritmética de complemento a 2. A excepción de la instrucción de multiplicación no signada (MPYU) la cual realiza una operación no signada en el multiplicador.

Una entrada del multiplicador proviene del registro temporal (T) y la otra del bus de datos o del bus de programa. El resultado de 32 bits es almacenado en el registro producto (P) el cual entonces se encuentra disponible para ser manipulado por la ALU. El resultado de 32 bits de la ALU es almacenado en el acumulador y debido a que las localidades de memoria son solamente de 16 bits,



Notes: All registers and data lines are 16-bits wide unless otherwise specified.

Figura 4.7: CALU del C50.

se cuenta con las instrucciones SACL y SACH para guardar la parte baja del acumulador (bits 0-15) y la parte alta del acumulador (bits 16-31) en memoria.

Los registros de corrimiento (escalador del registro (P), registro pre-escalador y registro post-escalador) hacen posible que la CALU pueda realizar escalamientos, extracción de bits, aritmética con precisión extendida y prevención de sobreflujo. Estos registros de corrimiento están conectados a la salida del registro (P) y el acumulador.

El C5x tiene instrucciones multiplica/acumula las cuales se ejecutan en un solo ciclo de máquina cuando se utilizan dentro de las instrucciones de repetición.

Los cuatro modos de corrimiento (PM) configurables para afectar la salida del registro (P) se utilizan precisamente en las operaciones de multiplica/acumula, y en la aritmética fraccional. El campo PM se localiza dentro del registro de estado ST1 y de acuerdo a su contenido será el modo de corrimiento:

- Si $PM=00$, la salida de 32 bits del registro (P) no sufre ningún corrimiento al ser transferida a la ALU o almacenada en memoria.
- Si $PM=01$, la salida del registro (P) sufre un corrimiento hacia la izquierda de un bit al ser transferida a la ALU o almacenada en memoria, y el bit menos significativo (LSB) es llenado con un cero. Este modo de corrimiento es utilizado para compensar el bit de signo extra que se produce al multiplicar dos números de 16 bits con aritmética de complemento a 2.
- Si $PM=10$, la salida del registro (P) sufre un corrimiento hacia la izquierda de 4 bits cuando se transfiere a la ALU o es almacenada en memoria, y los 4 bits menos significativos son llenados con cero. Este modo de corrimiento es utilizado junto con la instrucción MPY con un valor corto inmediato (13 bits o menos) para eliminar los 4 bits de signo extras que se producen al multiplicar un número de 16 bits con un número de 13 bits.
- Si $PM=11$, la salida del registro (P) sufre un corrimiento hacia la derecha de 6 bits cuando se transfiere a la ALU o es almacenada en memoria, se pierden los 6 bits menos significativos. Este modo de corrimiento permite la ejecución de hasta 128 operaciones consecutivas de multiplica/acumula sin que exista sobreflujo.

4.3.5 PLU

La unidad lógica paralela es una unidad independiente de la ALU pero opera en paralelo con ésta. Esta unidad constituye un segundo camino para la operación de funciones Booleanas o la manipulación de bits, ya que permite colocar, borrar, y probar bits en un registro de control/estado o en cualquier localidad de memoria dato. Dentro de la PLU se tienen:

- La unidad aritmética de los registros auxiliares (ARAU)

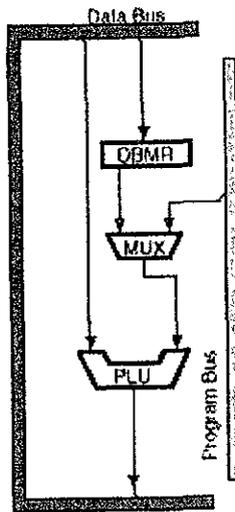


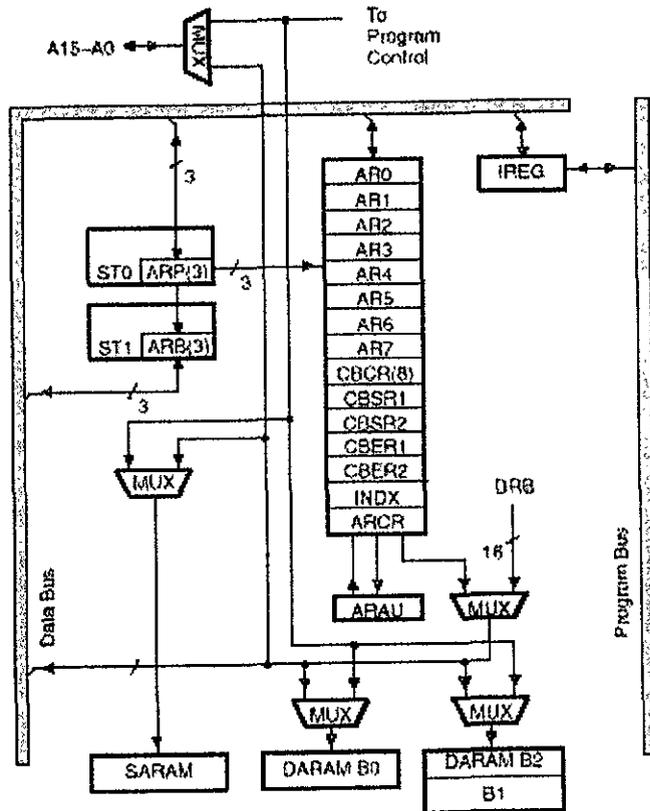
Figura 4.8: PLU del C50.

- Ocho registros auxiliares.

La PLU realiza operaciones con los contenidos de las localidades de memoria dato sin afectar el contenido del acumulador o el del registro producto. Cuando se hace un llamado a la PLU, ésta lee un valor dato desde la localidad de memoria dato especificada, este valor es manipulado junto con otro el cual es una constante contenida dentro de la misma instrucción o puede ser un valor variable contenido en el registro dinámico de manipulación de bit (DBMR). El resultado de la PLU se escribe de vuelta a la localidad de memoria dato original.

ARAU

La ARAU es una unidad lógica que realiza aritmética no signada de 16 bits. Contiene ocho registros auxiliares mapeados en memoria (AR0-AR7), los cuales son utilizados en el direccionamiento indirecto o para almacenamiento temporal de datos. Los registros auxiliares son seleccionados mediante un registro apuntador de 3 bits (para poder seleccionar los 8 ARs) conocido como registro apuntador de los registros auxiliares (ARP). Tanto los ARs como el ARP pueden ser cargados desde la memoria de datos, el acumulador o el registro producto o por un operando inmediato definido en la instrucción. El contenido de los registros auxiliares puede ser almacenado en memoria dato o puede ser utilizado como una entrada para la CALU. Los ARs están conectados a la ARAU.



Notes: All registers and data lines are 16-bits wide unless otherwise specified.

Figura 4.9: Diagrama funcional a bloques de la ARAU.

La ARAU es la encargada de calcular los direccionamientos indirectos utilizando como entradas los registros auxiliares (ARs), el registro índice (INDX) y el registro auxiliar de comparación (ARCR). La ARAU puede autoindexar el registro auxiliar actual (el valor indexado puede ser +1, -1, o el valor contenido en el registro índice) mientras que la dirección de memoria está siendo direccionada. El acceso a los datos no requiere del uso de la CALU, por lo tanto, está se encuentra libre para realizar otras operaciones en paralelo.

4.3.6 Registros mapeados en memoria

El C50 tiene 96 registros mapeados dentro de la página 0 del espacio de memoria dato, 28 de los cuales son registros del CPU y 16 son registros de puertos de entrada/salida. Los registros mapeados son utilizados para almacenar información temporal. También son utilizados como registros de control y estado del CPU y realizan aritmética entera a través de la ARAU.

Dec	Hex	Nombre	Descripción
0-3	0-3	---	Reservado
4	4	IMR	Registro para enmascarar interrupciones
5	5	GREG	Registro para localización global de memoria
6	6	IFR	Registro de banderas de interrupción
7	7	PMST	Registro de modo del estado del procesador
8	8	RPTC	Registro contador de repeticiones
9	9	BRCR	Registro contador de repetición de bloque
10	A	PASR	Dirección inicial de programa para la repetición de bloque
11	B	PAER	Dirección final de programa para la repetición de bloque.
12	C	TREG0	Registro temporal para guardar el multiplicando.
13	D	TREG1	Registro temporal para guardar el contador de corrimiento dinámico
14	E	TREG2	Registro temporal usado como apuntador de bit en las pruebas dinámicas de bit
15	F	DBMR	Manipulación dinámica de bit
16	10	AR0	Registro auxiliar 0
17	11	AR1	Registro auxiliar 1
18	12	AR2	Registro auxiliar 2
19	13	AR3	Registro auxiliar 3
20	14	AR4	Registro auxiliar 4
21	15	AR5	Registro auxiliar 5
22	16	AR6	Registro auxiliar 6
23	17	AR7	Registro auxiliar 7
24	18	INDX	Registro índice
25	19	ARCR	Registro auxiliar de comparación
26	1A	CBSR1	Dirección inicial del buffer circular 1
27	1B	CBER1	Dirección final del buffer circular 1
28	1C	CBSR2	Dirección inicial del buffer circular 2
29	1D	CBER2	Dirección final del buffer circular 2
30	1E	CBCR	Registro de control del buffer circular
31	1F	BMAR	Registro de dirección para el movimiento de bloque
32-35	20-23	---	Registros del puerto serial mapeados en memoria
36-42	24-2A	---	Periféricos mapeados en memoria
43-47	2B-2F	---	Reservado para pruebas y simulación
48-55	30-37	---	Registros del puerto serial mapeados en memoria
56-79	38-4F	---	Reservado
80-95	50-5F	---	Puertos de entrada/salida mapeados en memoria
96-127	60-7F	---	Bloque B2 de DARAM Content-Transfer -Encoding quoted-printable

Tabla 4.1: Mapa de direcciones de la página 0. Registros del CPU

4.3.7 Controlador de Programa

El controlador de programa contiene circuitería lógica que decodifica las instrucciones, maneja el pipeline, almacena el estado de las operaciones realizadas por el CPU y decodifica las operaciones condicionadas. El controlador de programa tiene los siguientes elementos:

- Contador del programa (PC)
- Registro de instrucción
- Stack o pila por Hardware
- Registros de estado y control

El contador del programa.

El PC del C50 de 16 bits contiene la dirección de la instrucción residente en memoria programa interna o externa que es buscada en la etapa Fetch. El PC direcciona la memoria de programa mediante el bus de dirección de programa (PAB). A través del PAB, la instrucción buscada es cargada en el registro de instrucción (IREG). Cuando el IREG es cargado, el PC está listo para iniciar un nuevo ciclo de búsqueda.

El PC se carga de distintas maneras dependiendo del código de operación que se vaya a realizar:

Código secuencial. El PC se carga con $PC+1$.

Ramificación (B). El PC se carga con el valor (largo) inmediato que sigue a la instrucción Branch.

Llamado a subrutina. El valor de $PC+2$ es guardado en el stack, después el PC se carga con el valor (largo) inmediato que sigue a la instrucción Call. La instrucción de retorno carga al PC con el valor guardado en el stack para continuar con la secuencia del programa.

Interrupción. El PC se carga con la dirección del vector de interrupciones apropiado.

GOTO. El PC se carga con el contenido de la parte baja del acumulador (ACCL). Mediante las instrucciones BACC (salta a la dirección especificada por el acumulador) o CALA (llamado a la subrutina localizada en la dirección que indica el acumulador) pueden realizarse las operaciones GOTO.

Instrucción de retorno. El PC se carga con el contenido del nivel superior del stack.

Dirección cargada en el PC

El PC puede cargarse con coeficientes residentes en memoria programa para la ejecución de instrucciones que utilizan la repetición. Durante una operación de repetición, una vez que la instrucción es repetida, ésta no vuelve a ser buscada en la etapa Fetch, por lo que el PC puede ser utilizado para direccionar la memoria de programa secuencialmente. Las instrucciones que utilizan esta facilidad son: MAC, MACD, MADD, MADS, BLDD, BLPD, BLDP, TBLR Y TBLW.

Stack o pila por Hardware.

El stack es de 16 bits y contiene 8 niveles. Las instrucciones PUSH y POP permiten salvar y recuperar la información contenida en el stack. Cada vez que el contenido del PC es guardado en el stack, éste se coloca en la localidad superior del stack empujando los contenidos previos un nivel hacia abajo, por lo que si se guardan ms de ocho datos en el stack se empezará a perder información.

El stack es utilizado durante las interrupciones y los llamados a subrutina para salvar y restaurar el contenido del PC. Cuando surge un llamado de subrutina (CALA, CALAD, CALL, CALLD, CC o CCD) o surge una interrupción (INT1-INT4, NMI, INTR o TRAP), la dirección de retorno es salvada automáticamente en el stack (se realiza una operación PUSH).

Cuando ocurre un regreso de subrutina (RET, RETC, RETCD, RETD, RETE o RETI) la dirección de retorno se carga al PC proveniente del stack (operación POP).

Registros de Control y Estado

El C50 tiene cuatro registros de control y estado:

El registro de control del buffer circular (CBCR) y el registro de estado del modo del procesador (PMST) contienen información sobre el control y estado del procesador. Ya que estos registros están mapeados en memoria pueden ser almacenados en o cargados desde la memoria de datos; por lo tanto el estado del CPU puede ser salvado y restaurado durante la ejecución de subrutinas y rutinas de interrupción.

Los registros de estado ST0 y ST1 contienen el estado de los distintos modos en que puede operar el C50. El CBCR y el PMST residen en el espacio de los registros mapeados de la página 0 de la memoria de datos. Ambos registros pueden ser manipulados directamente por la CALU y la PLU.

El PMST tiene asociado un registro sombra para que su contenido sea salvado durante la atención a una interrupción. El PMST es restaurado automáticamente cuando ocurren las instrucciones de regreso de interrupción (RETI) o regreso de interrupción habilitando interrupciones (RETE).

Circular Buffer Control Register (CBCR)

15-8	7	6-4	3	2-0
Reserved	CENB2	CAR2	CENB1	CAR1

Figura 4.11: El CBCR.

Bit	Nombre	*	Función
15-8	Reservado	—	Estos bits estan reservados.
7	CENB2	0	Bit para habilitar o deshabilitar el buffer circular 2: CENB2= 0, Buffer circular 2 deshabilitado. CENB2= 1, Buffer circular 2 habilitado.
6-4	CAR2	—	Registro auxiliar para el buffer circular 2. Estos bits seleccionan un registro auxiliar (AR0-AR7) para ser asignado al buffer circular 2.
3	CENB1	0	Bit para habilitar o deshabilitar el buffer circular 1: CENB1= 0, Buffer circular 1 deshabilitado. CENB1= 1, Buffer circular 1 habilitado.
2-0	CAR1	—	Registro auxiliar para el buffer circular 1. Estos bits seleccionan un registro auxiliar (AR0-AR7) para ser asignado al buffer circular 1 (No se puede asignar el mismo AR para los buffers circulares).

Tabla 4.2: El registro CBCR.

Los registros de estado ST0 y ST1 no residen en el mapa de memoria por lo que no pueden ser afectados por la PLU y sus instrucciones. Sin embargo estos registros pueden ser almacenados en y cargados desde la memoria de datos permitiendo salvar y restaurar el estado del C50 durante la ejecución de subrutinas. Los registros ST0 y ST1 también tienen asociados un registro sombra para salvar su contenido cuando se presenta una interrupción. Dichos registros son restaurado automáticamente cuando ocurren las instrucciones de regreso de interrupción (RETI) o regreso de interrupción habilitando interrupciones (RETE).

La instrucción LST escribe al ST0 o al ST1, y la instrucción SST lee el contenido de ellos. (la instrucción LST #0 no afecta el bit INTM) Los bits INTM y OVM en ST0 y los bits C, CNF, HM, SXM, TC y XF en ST1 pueden habilitarse o deshabilitarse individualmente mediante las instrucciones SETC

Processor Mode Status Register (PMST)

15-11	10	9	8	7	6	5	4	3	2	1	0
IPTR	0	0	0	AVIS	0	OVLY	FAM	MP/MC	NDX	TRM	BRAE

Figura 4.12: El PMST.

Bit	Nombre	*	Función
15-11	IPTR	00000	Bits del apuntador de los vectores de interrupciones. Estos bits seleccionan alguna de las 32 páginas divididas en bloques de 2K donde residen los vectores de interrupción. Estos vectores se ubican al inicio de la memoria de programa pero pueden ser relocalizados en memoria RAM al cargar los bits IPTR. Estos bits se leen como ceros.
10-08	—	000	
7	AVIS	0	Bit para la visualización de las direcciones. Habilita o deshabilita la visualización de las direcciones internas de programa en los pines de direcciones. AVIS= 0, la dirección de programa interna es conducida a los pines. AVIS= 1, las líneas de dirección no cambian con la dirección interna de programa. Este bit se lee como cero.
6	—	0	
5	OVLY	0	Bit para sobreponer la SARAM en memoria dato. OVLY= 0, la SARAM no es direccionable en el espacio de memoria dato. OVLY= 1, la SARAM esta localizada en el espacio de memoria dato.
4	RAM	0	Bit para habilitar la SARAM en memoria programa. RAM= 0, la SARAM no es direccionable en el espacio de memoria programa. RAM= 1, la SARAM está localizada en el espacio de memoria programa.
3	MP/MC	1	Bit para habilitar el modo de microprocesador o microcomputador. MP/MC= 0, memoria ROM interna localizada en el espacio de memoria programa MP/MC= 1, memoria ROM no es direccionable en el espacio de memoria programa.
2	NDX	0	Bit para habilitar el registro índice manteniendo la compatibilidad con la generación C2x. NDX= 0, Modo de compatibilidad con el C2x. Si ocurre una instrucción compatible del C2x que modifique o escriba al AR0, también se modifica el INDX y el ARCR, ya que el C2x utiliza al AR0 para realizar las operaciones de indexación y comparación. NDX= 1, Modo estricto para el C5x. Cualquier instrucción compatible del C2x no afecta el INDX ni el ARCR, dicha instrucción solo afecta el AR0 del C5x.
1	TRM	0	Bit para habilitar múltiples registros T para mantener la compatibilidad con la generación C2x. TRM= 0, Modo de compatibilidad con el C2x. Si ocurre una instrucción compatible del C2x que cargue el registro T0, también cargará los registros T1 y T2, ya que el C2x utiliza al registro T como cuenta de corrimiento para el registro de corrimiento pre-escalador.

Tabla 4.3: El registro PMST.

Bit	Nombre	*	Función
			TRM= 1, Modo estricto para el C5x. Cualquier instrucción compatible del C2x no afecta los registros T0 ni T1, dicha instrucción solo afecta el registro T0 del C5x.
0	BRAF	0	Bit de la bandera de repetición de bloque. Este bit indica que una repetición de bloque está activa. BRAF= 0, No esta activa la repetición de bloque. El bit BRAF cambia a cero cuando el BRCCR se decrementa a cero. BRAF= 1, La repetición de bloque está activa. El bit BRAF cambia a uno automáticamente cuando la instrucción RPTB es ejecutada.

Tabla 4.4: Continuación del registro PMST.

(pone el bit en uno) y CLRC (pone el bit en cero). Los bits DP en ST0 pueden ser cargados utilizando la instrucción LDP. y los bits PM en ST1 pueden ser cargados mediante la instrucción SPM.

Status Register 0 (ST0)

15-13	12	11	10	9	8-0
ARP	OV	OVM	1	INTM	DP

Figura 4.13: El registro ST0

Status Register 1 (ST1)

15-13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARB	CNF	TC	SXM	C	1	1	HM	1	XF	1	1	1	PM

Figura 4.14: El registro ST1

Bit	Nombre	*	Función
15-13	ARP	0	Apuntador del registro auxiliar.
12	OV	0	Bit de la bandera de sobreflujo. Indica que una operación aritmética produjo sobreflujo en la ALU. El bit OV puede ser modificado con la instrucción LST #0. OV= 0, No ocurrió sobreflujo en la ALU. El bit OV se pone en cero después de un reset o una instrucción de ramificación condicionada (BCND). OV= 1, Ocurrió sobreflujo en la ALU. La señal de sobreflujo se mantiene como en un latch.
11	OVM	0	Bit de modo de sobreflujo. Habilita o deshabilita el modo de saturación del acumulador al ocurrir un sobreflujo en la ALU. El bit OVM puede ser modificado con la instrucción LST #0. OVM= 0, Deshabilitado. El resultado al ocurrir sobreflujo es cargado en el acumulador sin modificaciones. El bit OVM puede ser borrado con la instrucción CLRC OVM. OVM= 1, Habilitado. Al ocurrir sobreflujo, el acumulador se carga ya sea con el valor más positivo (7FFF FFFFh) o con el valor más negativo (8000 0000h). El bit OVM puede ser puesto en 1 con la instrucción SET OVM. Este bit se lee como uno.
10	—	1	
9	INTM	1	Bit de modo de interrupción. Este bit actúa globalmente enmascarando o habilitando las interrupciones mascarables (RS y NM son interrupciones no mascarables). El bit INTM no es salvado ni restaurado al ocurrir una interrupción y su respectivo regreso. INTM= 0, Todas las interrupciones mascarables están habilitadas. El bit INTM puede ser borrado con las instrucciones INTM y RETE. INTM= 1, Todas las interrupciones mascarables están deshabilitadas. El bit INTM puede ser colocado en 1 con las instrucciones SETC INTM e INTR, después de un reset o cuando se toma la interrupción mascarable.
8-0	DP	0	Bits del apuntador de la página de memoria dato. Estos bits especifican la dirección de la página de memoria dato actual. Pueden ser modificados con las instrucciones LST #0 o LDP.

Tabla 4.5: El registro ST0.

Bit	Nombre	*	Función
15-13	ARB	0	Bits del buffer del apuntador del registro auxiliar. En este campo se guarda el valor previo contenido en el ARP, excepto cuando el ARP ha sido cargado con la instrucción LST #0. Cuando se carga el ARB con la instrucción LST #1, este valor se copia en el ARP.
12	CNF	0	Bit que controla la configuración del bloque DARAM B0. CNF= 0, El bloque de DARAM B0 esta localizado en el espacio de memoria dato. El bit CNF puede ser borrado mediante la instrucción CLRC CNF o despues del reset. CNF= 1, El bloque de DARAQM B0 esta localizado en el espacio de memoria programa. El bit CNF puede ser colocado en 1 con la instrucción SETC CNF.
11	TC	0	Bit de la bandera de prueba/control. Este bit guarda el resultado de las pruebas de bit realizadas por la ALU o la PLU. El estado del bit TC determina si las instrucciones de ramificación condicionada, llamada y regreso son ejecutadas. El bit TC puede ser modificado con la instrucción LST #1. TC= 0, El bit TC puede ser borrado por la instrucción CLRC TC o por cualquiera de los siguientes eventos: - El resultado de la operación lógica es 1 al ejecutar las instrucciones OPL, APL O XPL. - El bit probado por las instrucciones BIT o BITT es igual a 0. - Una condición de comparación es falsa cuando se prueba la instrucción CMPR o CPL - El resultado de la operación OR-exclusiva es falso cuando se realiza la instrucción NORM TC= 1, El bit TC se pone en uno con la instrucción SETC TC o por cualquiera de los siguientes eventos: - El resultado de la operación lógica es 0 al ejecutar las instrucciones OPL, APL O XPL. - El bit probado por las instrucciones BIT o BITT es igual a 1. - Una condición de comparación es verdadera cuando se prueba la instrucción CMPR o CPL - El resultado de la operación OR-exclusiva es verdadero cuando se realiza la instrucción NORM.
10	SXM	1	Bit del modo de signo extendido. Este bit habilita o deshabilita el modo de signo extendido de una operación aritmética. Las instrucciones ADCC, ADDS, SUBB o SUBS suprimen la extensión de signo no importando que el bit SXM este habilitado. El bit SXM puede ser modificado por la instrucción LST #1. SXM= 0, Modo de signo extendido deshabilitado. El bit SXM puede ser borrado con la instrucción CLRC SXM. SXM= 1, Habilitado. El modo de signo extendido se produce en el dato cuando éste se transfiere al acumulador a través del registro de corrimiento escalador.

Tabla 4.6: Registro ST1.

Bit	Nombre	*	Función
9	C	1	<p>Bit de acarreo. Este bit indica que la operación aritmética originó un acarreo o un préstamo en la ALU. El bit C puede ser modificado con la instrucción LST #1.</p> <p>C= 0, El resultado de una resta generó un préstamo o el resultado de una suma (excepto la instrucción ADD con un corrimiento de 16 bits) no produjo acarreo. El bit C puede ser borrado por la instrucción CLRC C.</p> <p>C= 1, El resultado de una suma generó acarreo o el resultado de una resta (excepto la instrucción SUB con un corrimiento de 16 bits) no generó acarreo. El bit C puede ser colocado en 1 con la instrucción SETC C.</p> <p>Estos bits se leen como uno.</p>
8-7	—	11	
6	HM	1	<p>Bit de modo de espera. Este bit determina si el CPU para o continúa con la ejecución cuando se presenta la señal de HOLD. El bit HM puede ser modificado con la instrucción LST #1.</p> <p>HM= 0, El CPU continúa con la ejecución desde la memoria programa interna pero coloca en estado de alta impedancia a sus interfaces externas. El bit HM puede ser borrado por la instrucción CLRC HM.</p> <p>HM= 1, El CPU para su ejecución interna. El bit HM puede ser colocado en 1 con la instrucción SETC HM o después de un reset.</p> <p>Este bit se lee como uno.</p>
5	—	1	
4	XF	1	<p>Bit de estado del pin XF. Este bit determina el estado en que se encuentra la bandera del pin de salida XF. El bit XF no es salvado ni restaurado durante una interrupción y su respectivo regreso. El bit HM puede ser modificado con la instrucción LST #1.</p> <p>XF= 0, El pin de salida XF está en un nivel lógico bajo. El bit XF puede ser borrado por la instrucción CLRC XF.</p> <p>XF= 1, El pin de salida XF tiene un nivel lógico alto. El bit XF puede ser colocado en 1 con la instrucción SETC XF.</p> <p>Estos bits se leen como uno.</p>
3-2	—	11	
1-0	PM	0	<p>Bits de modo del registro de corrimiento producto. Estos bits especifican el modo en el que trabajará el registro de corrimiento del producto.</p>

Tabla 4.7: Registro ST1. Continuación.

Operación Pipeline

Los microprocesadores C50, al acceder una instrucción de memoria y proceder a ejecutarla, separan dicha tarea en cuatro operaciones independientes, lo cual resulta en una operación y ejecución de programa más eficiente y se conoce como operación Pipeline.

Las fases de la operación Pipeline son las siguientes:

- **Búsqueda (Fetch) (F):** Esta fase obtiene las palabras de instrucción de la memoria y actualiza el contador de programa (PC).
- **Decodificación (D):** Esta fase realiza la decodificación de la instrucción y genera una dirección, Por otra parte la ARAU se encarga de realizar las modificaciones hechas a los registros auxiliares.
- **Lectura (R):** En caso de ser necesario, en esta fase se realiza la lectura de operandos.
- **Ejecución (E):** En esta fase se realiza la operación especificada y de ser necesario escribe el resultado de una operación anterior a memoria.

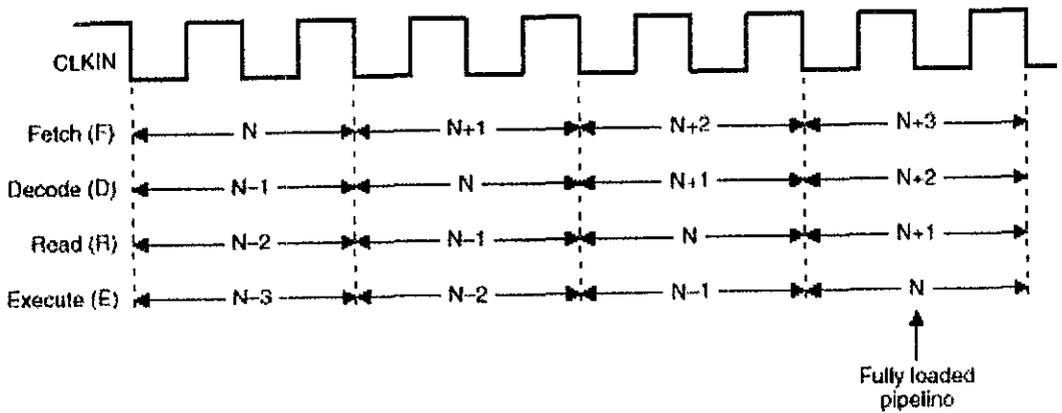
Cuando estas cuatro operaciones se superponen perfectamente en el mismo ciclo de Pipeline, es decir, trabajan en paralelo hasta 4 diferentes instrucciones pueden estar activas, cada una de las cuales estará en una etapa de ejecución diferente. Para el programador, la ejecución de las instrucciones en el Pipeline es transparente, excepto en algunos casos como la actualización de los registros auxiliares, acceso a los registros mapeados del CPU y los comandos de configuración de memoria. Un problema que se presenta con el Pipeline es cuando se presentan instrucciones de ramificaciones, llamados a subrutinas o regreso de subrutinas ya que el contador de programa debe de continuar en otra dirección de programa, perdiéndose entonces la secuencia del Pipeline.

Temporizador

Este elemento de Hardware realiza conteos de 16 bits, conforme indiquen los registros de control y de período. Las funciones de inicio, paro, reinicio, inicialización y deshabilitación son controlados mediante el registro de control del temporizador.

Generador de Reloj

Está constituido por un oscilador interno y un circuito de bucle de fase cerrada (PLL). El generador de reloj puede ser controlado internamente por un circuito resonante o externamente mediante una fuente de reloj.



Ejemplo de Operación Pipeline para instrucciones de 1 palabra:

```

ADD    *+
SAMM  TREG0
MPY    *+
SQRA   *+, AR2

```

Cycle	PC	Pipeline operation				ARP	AR6	TREG0	PREG	ACC
		F	D	R	E					
1	[SAMM]	ADD				6	60h	XX	XX	20h
2	[MPY]	SAMM	ADD			6	61h	XX	XX	20h
3	[SQRA]	MPY	SAMM	ADD		6	61h	XX	XX	20h
4		SQRA	MPY	SAMM	ADD	6	62h	XX	XX	30h
5			SQRA	MPY	SAMM	2	63h	30h	XX	30h
6				SQRA	MPY	X	XX	30h	90h	30h
7					SQRA	X	XX	06h	24h	C0h

Figura 4.15: Operación del Pipeline

Generación por software de estados de espera

Este circuito permite que se puedan realizar interfaces entre elementos de memoria externa lentos y los puertos de entrada/salida del C5x. Este periférico elimina la necesidad de utilizar circuitería lógica externa para crear estados de espera vía el pin READY.

Puertos paralelos de entrada/salida

El C5x tiene un total de 64K direcciones de entrada salida, 16 de los cuales están mapeados en memoria. Estos puertos son multiplexados con las líneas de datos y son identificados mediante la señal de espacio seleccionado de entrada/salida.

Interrupciones mascarables para el usuario.

El TMS320C50 tiene cuatro interrupciones externas (INT1-INT4), 5 interrupciones generadas por los periféricos a bordo, una interrupción para el temporizador (TINT) y las otras cuatro para el puerto serial (RINT, XINT, TRNT, TXNT), 3 interrupciones por software (INTR, NMI y TRAP) y 2 interrupciones no mascarables (RS y NMI). Cuando se ejecuta una rutina de interrupción (ISR), el contenido del contador de programa (PC) es guardado en la pila o stack por hardware de 8 niveles, y los contenidos de 11 registros del CPU son salvados automáticamente en los registros sombra. Cuando se ejecuta la instrucción de regreso de interrupción se restauran los valores de los registros del CPU.

4.3.8 Modos de direccionamiento

El C50 puede direccionar un total de 64K palabras de la memoria de programa y 64K palabras de la memoria dato. Ahora bien, existen varios métodos por los cuales una instrucción puede calcular la dirección de los datos (localidades de memoria donde residen los operandos) que va a utilizar. A estos métodos se les conoce como modos de direccionamiento, y son:

Modo de direccionamiento directo

El direccionamiento directo se utiliza para realizar operaciones con valores localizados dentro de la memoria de datos. Debido a que el C50 tiene un tamaño de instrucción de palabra igual a 16 bits y a que el tamaño de las localidades de memoria es también de 16 bits, sería obvio pensar, que para poder utilizar directamente un valor de la memoria dato se necesitarían dos ciclos de máquina; uno para especificar la operación que se quiere realizar y el segundo para especificar la dirección de memoria dato a ser accesada. Esto resultaría en un acceso muy lento para el direccionamiento directo y por lo tanto indeseable, por lo que el C50 permite que el usuario especifique un área reducida de memoria

Tabla 4.8: Modos de Direccionamiento.

MODO	PROPOSITO
Direccionamiento Directo	Accesar datos localizados dentro de una página de memoria dato.
Direccionamiento Indirecto	Accesar datos desde arreglos localizados en cualquier parte de la memoria dato.
Direccionamiento Inmediato	Inicializar registros, operar con constantes.
Direccionamiento de registros "dedicados"	Utiliza los registros "dedicados" BMAR y DBMR.
Direccionamiento Circular	Implementar el deslizamiento de una ventana, la cual contiene los datos más recientes a ser procesados.
Direccionamiento de registros mapeados	Permite tener acceso rápido a registros con funciones especiales localizadas en el mapa de memoria dato.

dentro de la cual operarán las instrucciones del direccionamiento directo bajo un compromiso entre la velocidad y el rango de memoria adecuados.

El C50 utiliza un registro de 9 bits llamado apuntador de página (DP) para especificar el área activa de memoria, entonces el DP que es de 9 bits permite apuntar a una de las 512 páginas compuestas por 128 palabras cada una, en las que se divide la memoria de datos. Las 128 localidades de cada página serán seleccionadas mediante los 7 bits menos significativos de la palabra de instrucción (los 8 bits más significativos se utilizan para especificar la operación), entonces al concatenar los 9 bits del DP con los 7 bits de dirección se obtiene la dirección de memoria de datos (dma) de 16 bits lo que permitirá realizar el direccionamiento directo en un solo ciclo. Esta DMA es colocada en el bus de direcciones de la memoria dato (DAB). Note que el bit 7, con el valor de 0, define al modo de direccionamiento directo, cuando este bit está en 1 define un modo de direccionamiento indirecto.

Entonces, para realizar direccionamientos directos, primero se inicializa el DP mediante la instrucción LDP y posteriormente se realizan las operaciones deseadas, ejemplo:

- LDP #3 Se activa la página 3 de la memoria de datos, localidades absolutas 180h-1FFh.
- LACC 6 Carga el acumulador con el contenido de la localidad 6 de la página 3, localidad absoluta 186h.
- ADD 13h Suma al acumulador el contenido de la localidad 13 de la página 3, localidad absoluta 193h.

Para acceder valores localizados en diferentes páginas, se requiere utilizar un ciclo extra para recargar el DP.

- LDP #1 se activa la página 1 de la memoria de datos, localidades 0 080h-00FFh.
- SUB 2 resta el contenido de la localidad 2 de la página 1 al acumulador, localidad absoluta 82h.
- SACL 5 guarda el contenido del acumulador en la localidad 5 de la página 1, localidad absoluta 85h.

Modo de direccionamiento indirecto

El direccionamiento indirecto es una forma muy eficiente y versátil para poder acceder datos almacenados en arreglos, listas, tablas u otros grupos ordenados dentro de la memoria dato. En este modo, la dirección no está expresada en la instrucción como en el direccionamiento directo, sino dentro de uno de los 8 registros auxiliares (AR0-AR7), los cuáles deben de ser previamente activados e inicializados. Los registros auxiliares se activan mediante la instrucción MAR (modifica registro auxiliar) y se inicializan con la instrucción LAR (carga registro auxiliar).

Los ARs ofrecen grandes ventajas:

- Por ser registros de 16 bits, estos pueden direccionar todo el espacio de memoria dato (64K sin la ayuda del registro DP).
- Facilitan los procesos iterativos, ya que después de utilizar la dirección contenida dentro del registro auxiliar (operando) en la instrucción actual, puede incrementar o decrementar dicho operando, quedando un nuevo dato apuntado que será utilizado en una operación futura

Operandos del modo de direccionamiento indirecto

Veamos el siguiente ejemplo:

```
MAR  *,AR3
LAR  AR3,13h
ADD  *, 1, AR0
```

Siendo la secuencia del direccionamiento indirecto:

1. * El apuntador de registro auxiliar ARP selecciona el AR3 para contener al operando.
2. 1 El operando sufre un corrimiento de 1 bit hacia la izquierda (OPCIONAL)
3. ADD La instrucción opera con el operando desplazado
4. + Modifica el valor del AR (OPCIONAL)
5. AR0 Nuevo valor para el ARP (OPCIONAL)

Como mencionamos anteriormente, el primer paso es activar e inicializar el registro auxiliar. Esto se realiza con las primeras dos instrucciones, en este

caso se activa el registro auxiliar AR3 y se le inicializa con la dirección de memoria dato 13h. En la tercera instrucción el primer operando (*+) sirve para activar el registro auxiliar AR3 en la instrucción (suma al acumulador el contenido, afectado por el corrimiento, de la dirección apuntada por AR3 y después incrementa el valor de AR3), el siguiente operando (1) se debe a que las instrucciones que trabajan con el acumulador, como lo es ADD, ofrecen una opción de corrimiento en el espacio del segundo operando. Si no se escribe algún valor, se toma el valor 0 por default. El espacio del tercer operando se denomina "campo del AR nuevo", con el cual se activará un AR nuevo después de haber utilizado el anterior en la instrucción.

La ARAU es la encargada de realizar las operaciones aritméticas para auto-indexar los registros auxiliares, mientras se realiza el direccionamiento del operando. Por esto, la CALU no es utilizada para la manipulación de las direcciones cuando se accesa a información contenida en tablas, sino que se utiliza para realizar operaciones paralelas. Existen diferentes tipos de auto-incremento y auto-decremento, teniéndose las opciones siguientes:

Opciones del direccionamiento indirecto

OPCION	FUNCION
*	No modifica el AR actual.
+	Incrementa el AR actual.
-	Decrementa el AR actual.

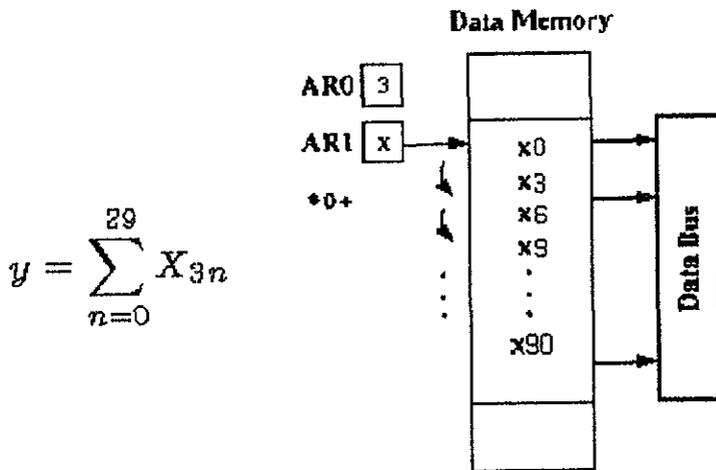


Figura 4.17: Ejemplo del direccionamiento indirecto.

Ejemplo:

```

LAR  AR1, #x
MAR  *,AR1
RPTZ #99
ADD  *+
SACL y

```

Direccionamiento indirecto con indexación

Hay ocasiones en las que se desea incrementar o decrementar por un valor diferente de uno, y es en estos casos que se utiliza el registro índice (INDX) para especificar el tamaño del incremento/decremento que se aplicará. Para denotar que se está haciendo uso de este tipo de incremento/decremento se añade un 0 antes del modificador como en *0+ y *0-.

Note que el C2x utiliza al AR0 como registro índice para guardar el valor de indexación que se aplicará en los demás ARs. Sin embargo, el C5x puede separar en 3 registros independientes al registro AR0:

- Un registro auxiliar (AR0).
- Un registro índice (INDX).
- Un registro auxiliar de comparación (ARCR).

Tabla 4.9: Más opciones del direccionamiento indirecto.

OPCION	FUNCION
*0+	Suma el contenido del registro INDX al AR actual.
0-	Resta el contenido del registro INDX al AR actual.
BR0+	Suma el contenido del registro INDX al AR actual con propagación de carry en reversa.
BR0-	Resta el contenido del registro INDX al AR actual con propagación de carry en reversa.
if (AR actual) = (ARCR), TC= 1	Comparación entre el registro AR actual y el registro ARCR, si la condición es verdadera
if (AR actual) = (ARCR), TC= 1	el bit TC del ST1 toma el valor de 1,
if (AR actual) = (ARCR), TC= 1	si es falsa toma el valor de 0.

Esto se realiza al colocar en 1 el bit NDX en el registro de estado del modo del procesador (PMST). El ARCR se utiliza para la comparación de bloques de datos. Mediante la instrucción CMPR se realizan comparaciones lógicas entre el AR actual y el ARCR, el resultado de la comparación se guarda en el bit TC del registro de estado 1 (ST1).

Modo de direccionamiento inmediato.

En este modo de direccionamiento, el operando inmediato es parte de la palabra de instrucción y se identifica mediante el símbolo de la libra (#). El C5x tiene instrucciones de una palabra para operar constantes de 8, 9 y 13 bits (instrucciones inmediatas cortas), e instrucciones de 2 palabras para constantes de 16 bits (instrucciones inmediatas largas). Ejemplo de direccionamiento inmediato:

```
ADD    #16h   Se suma la constante 16h al acumulador en un ciclo
ADD    #3754h Se suma la constante 3754h al acumulador en dos ciclos
```

Es muy importante hacer notar que el espacio de memoria dato no es utilizado por las instrucciones anteriores. Una máquina con arquitectura Harvard obtiene sus operandos solamente del espacio de datos. La característica de obtener los operandos del espacio de programa es una de las razones por las cuales se dice que el TMS320 tiene una arquitectura Harvard modificada. El direccionamiento inmediato largo permite especificar un segundo operando que denota un valor de corrimiento, el cual puede ser utilizado para colocar una constante de 16 bits dentro de un registro de 32 bits como lo ejecutan las instrucciones AND y LACC. Ejemplo:

```
LACC  #731h,2  carga al acumulador con el número 731 desplazado 2
                bits hacia la izquierda.(ACC= 0E62h).
```

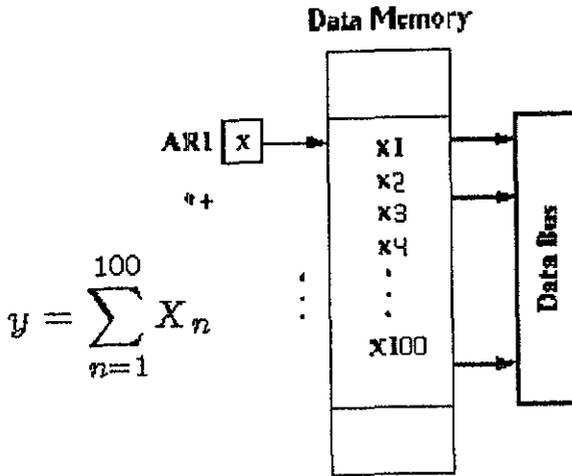


Figura 4.18: Direccinamiento indirecto indexado.

Corta inmediata (1-palabra)		Larga inmediata (2 palabras)	
Constante de 8 Bits	Constante de 9 bits	Constante de 13 bits	Constante de 16 bits
ADD	LDP	MPY	ADD
ADRK			OR
LACL			RPT
LAR			RPTZ
RPT			CPL
SBRK			SPLK
SUB			LACC
			SUB
			LAR
			XOR
			MPY
			XPL
			OPL

Tabla 4.10: Instrucciones que soportan direccinamiento inmediato

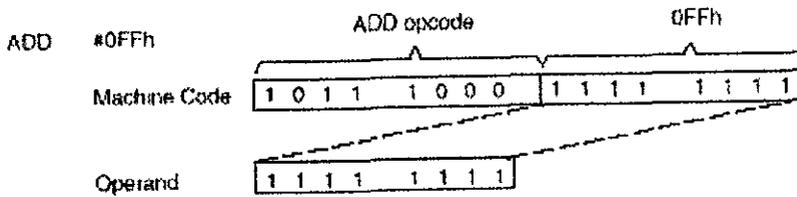


Figura 4.19: Direccionamiento inmediato corto.

El direccionamiento largo también puede darse con un acceso a memoria dato, es decir, tener una localidad de memoria dato como segundo operando. El contador de pre-búsqueda (PFC) es salvado en un stack de microlamadas (MCS), y el valor largo inmediato es cargado en el PFC. En seguida el bus de direcciones de programa (PAB) o el bus de datos se utilizan para buscar al operando o para leerlo, según sea el caso. Completada la instrucción, el MCS restaura el valor del PFC, el contador del programa (PC) es incrementado por dos y la ejecución continúa. El PFC se utiliza cuando se repite una instrucción, ya que la dirección generada puede ser autoincrementada.

Direccionamiento de Registros "dedicados"

Este modo de direccionamiento opera como el modo de direccionamiento inmediato largo, a excepción de que la dirección proviene de uno de los dos registros de propósito especial mapeados en memoria: El registro de dirección para el movimiento de bloque (BMAR) y el registro de manipulación dinámica de bit (DBMR). La ventaja de este tipo de direccionamiento es que la dirección del bloque de memoria a ser manipulado puede ser modificada durante la ejecución del programa. Las instrucciones BLDD, BLDP y BLPD utilizan al registro BMAR para apuntar la dirección de origen o destino a partir de la cual se hará el movimiento de datos. Las instrucciones MADD y MADS también utilizan al registro BMAR para direccionar un operando localizado en la memoria programa para realizar la operación multiplica/acumula.

Las instrucciones APL, CPL, OPL y XPL utilizan a la PLU y al contenido del registro DBMR cuando no se especifica un valor inmediato como uno de los operandos.

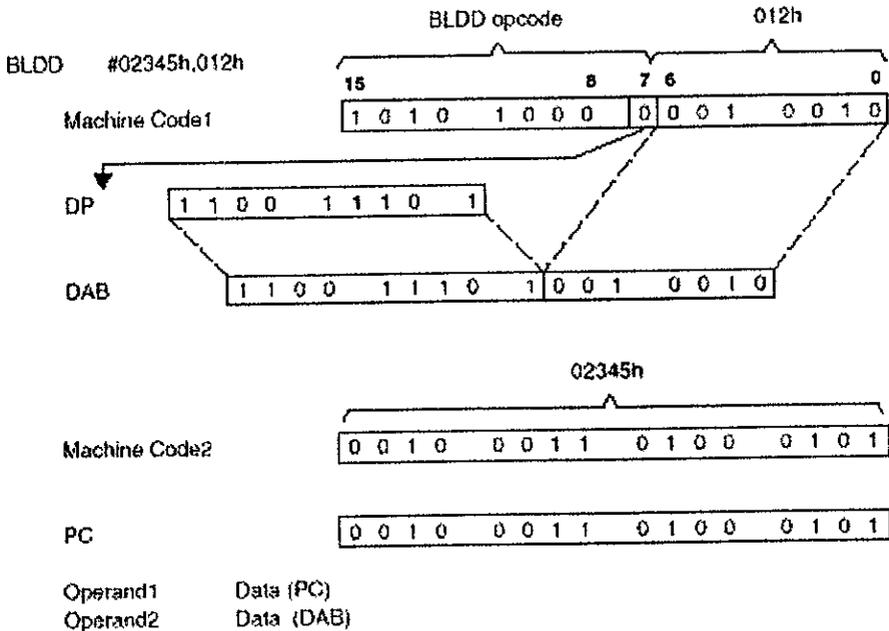


Figura 4.20: Direccionamiento inmediato largo con 2 operandos.

Ejemplos:

BLDD BMAR, 100h Teniendo en cuenta que: DP=0, BMAR = 200h

El contenido de la localidad de memoria 200h, es copiado a la localidad de memoria dato 100h de la página actual.

OPL 10h Teniendo en cuenta que: DP= 6, DBMR = FFF0h y que la dirección 030Ah tiene el valor de 01h.

Se realiza la operación lógica OR entre el contenido de la localidad de memoria dato 030Ah y el contenido del DBMR (FFF0h). El resultado FFF1h es guardado de nuevo en la localidad de memoria 030Ah.

Direccionamiento de Registros Mapeados.

Este modo de direccionamiento trabaja como el direccionamiento directo, con la diferencia de que los 9 bits más significativos de la dirección son llenados con ceros en lugar de ser cargados con el contenido del DP, lo cual permite modificar los registros mapeados en memoria, el bloque de memoria DARAM B2

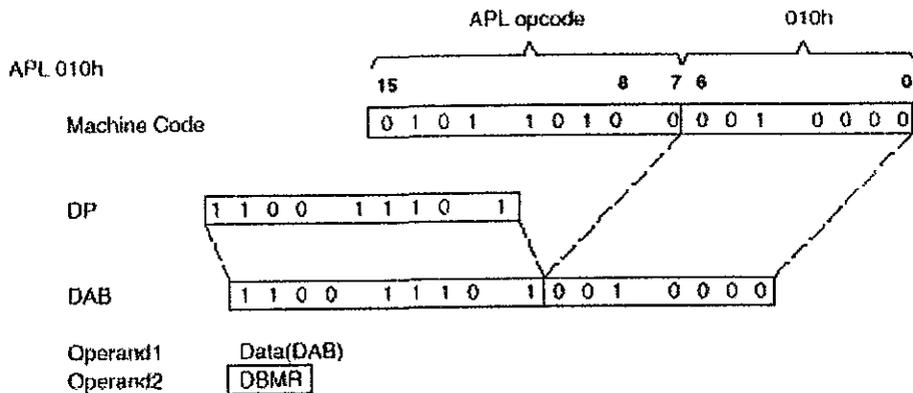
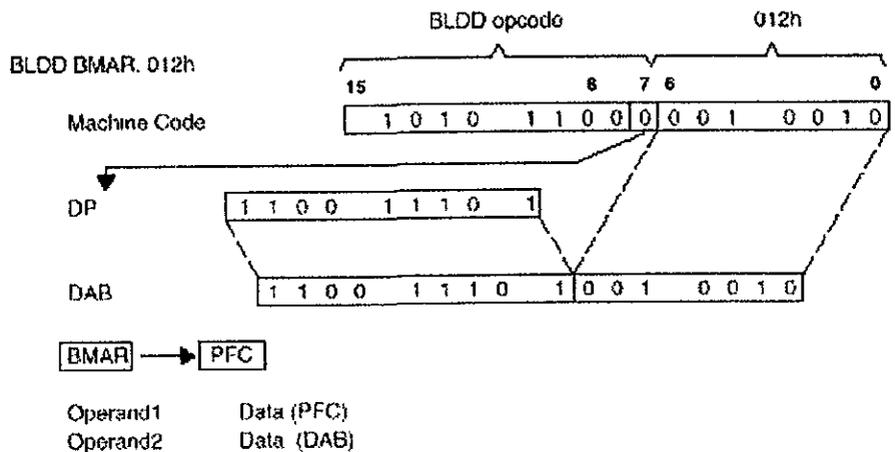


Figura 4.21: Direcccionamiento de registro dedicado usando el BMAR y el DBMR.

y todas las localizadas ubicadas en la página 0 sin importar en que página se este trabajando. Las siguientes instrucciones operan en el modo de direccionamiento de registros mapeados:

- LAMM Carga el acumulador con el contenido del registro mapeado.
- LMMR Carga el registro mapeado
- SAMM Guarda el contenido del acumulador en el registro mapeado.
- SMMR Guarda el contenido del registro mapeado.

Direccionamiento circular

Muchos algoritmos como la convolución, correlación, y los filtros con respuesta finita pueden utilizar el direccionamiento circular para implementar el deslizamiento de una ventada, la cual contiene los datos más recientes a ser procesados. El C50 tiene dos buffers circulares, cada uno de los cuales utiliza un registro auxiliar diferente. Los registros que controlan el direccionamiento circular son:

- CBSR1 Contiene la dirección inicial del buffer circular 1.
- CBER1 Contiene la dirección final del buffer circular 1.
- CBSR2 Contiene la dirección inicial del buffer circular 2.
- CBER2 Contiene la dirección final del buffer circular 2
- CBCR Registro de control del buffer circular.

Estos registros pueden cargarse con la instrucción SPLK, la cual implica que la PLU cargue una constante inmediata larga. Ejemplo:

```
SPLK #75h,CBSR1
SPLK #7Ch,CBSR2
```

Para utilizar el direccionamiento circular, primero tienen que definirse las direcciones de inicio y final correspondientes a cada registro de buffer, después cargar un valor intermedio (entre el valor inicial y final) en el registro auxiliar asignado. Cargar el CBCR con dicho registro auxiliar y habilitar el correspondiente buffer circular. El algoritmo para el direccionamiento circular es:

Si	(ARn = CBER) y se indica alguna modificación para el AR
entonces	ARn = CBSR
de otra forma	ARn = ARn + el paso indicado

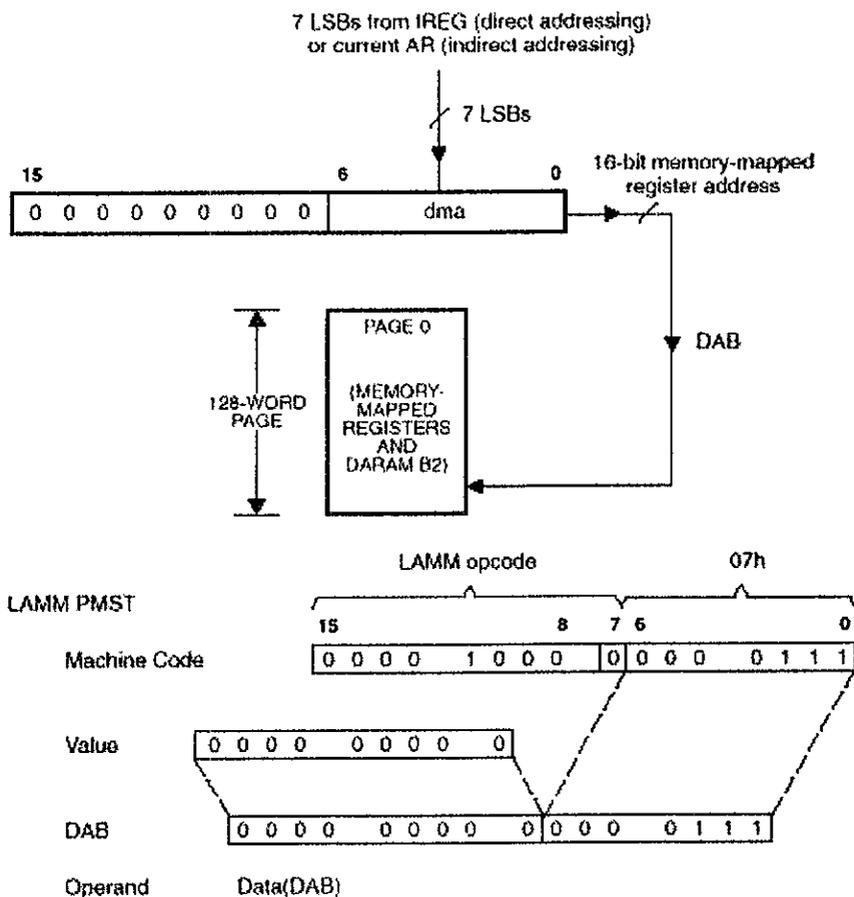


Figura 4.22: Direccionamiento de Registros mapeados en memoria, y en el modo Directo de direccionamiento.

Nombre	Dec	Hex	Prioridad	Función
/RS	0	0	1	Seal externa no mascarable de reset
/INT1	2	2	3	Interrupción externa del usuario #1
/INT2	4	4	4	Int. externa del usuario #2
/INT3	6	6	5	Int. externa del usuario #3
TINT	8	8	6	Int. interna del temporizador
RINT	10	A	7	Int. de recepción del puerto serial
XINT	12	C	8	Int. de transmisión del puerto serial
TRNT	14	E	9	Int. de recepción del puerto TDM
TXNT	16	10	10	Int. de transmisión de puerto TDM
/INT4	18	12	11	Int. externa del usuario #4
—				
TRAP	34	22	N/A	Vector de interrupción TRAP
/NMI	36	24	2	Int. externa no-mascarable
—				
—	40-63	28-3F	N/A	Ints. por software

Tabla 4.11: Localidades y prioridades del vector de interrupciones.

Si $AR_n = CBER$ pero ninguna modificación al AR ocurre, el AR actual no sufre modificaciones y mantiene el valor del $CBER$, en caso de darse la modificación al AR éste tomará el valor del $CBSR$.

Interrupciones

Las interrupciones son un medio por el cual se fuerza a que el C50 suspenda la ejecución de su programa principal con el propósito de responder a un evento originado vía hardware. Esto elimina la necesidad de manejar los eventos externos a través de software, lo que implica obtener tiempos de respuesta muy cortos. Las interrupciones son generadas principalmente por dispositivos que necesitan dar o recibir información del C50, tales como convertidores A/D y D/A. Cuando ocurre una interrupción, el C50 suspende la ejecución del programa principal e inicia la ejecución del código especificado en el vector de interrupción de la interrupción originada. El programador puede seleccionar dinámicamente cuando y cuáles interrupciones serán reconocidas.

La tabla siguiente muestra la localización y la prioridad de los vectores de interrupción de las interrupciones internas y externas. Las direcciones de las interrupciones están espaciadas dos localidades para poder acomodar las instrucciones de salto (Branch) a las subrutinas. Los vectores de interrupción pueden ser relocalizados al principio de cualquiera de las páginas de 2K en las que se divide el espacio de memoria programa. La dirección del vector de interrupciones se genera al concatenar los bits $IPTR$ situados en el $PMST$ y el número del vector de interrupción.

Después de la señal de reset, los bits $IPTR$ se colocan en cero, localizando

Interrupt Flag Register (IFR)

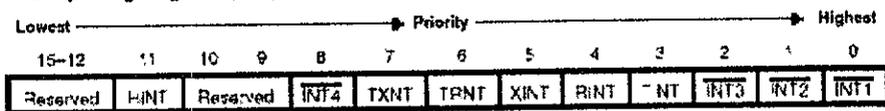


Figura 4.23: Registro de banderas para las interrupciones .

los vectores en la página 0 de la memoria de programa.

El reset (/RS) tiene la mayor prioridad de las interrupciones y la /INT4 la menor. Las interrupciones /INT1-/INT4 y la /NMI son válidas si la señal externa está en alto por lo menos dos ciclos y baja por un mínimo de 3 ciclos. Esto permite que el C50 evite las interrupciones falsas ocasionadas por el ruido o que tome varias interrupciones con una sola señal de interrupción larga.

RESET

El reset (/RS) es una interrupción externa no-mascarable la cual tiene la mayor prioridad las interrupciones, se aplica en el encendido para que el procesamiento se inicie desde un estado conocido del procesador. El reset puede efectuarse en cualquier momento del procesamiento para terminar la ejecución presente y reestablecer sus condiciones iniciales. Para tener una correcta operación del reset, el pin /RS del C50 debe permanecer en estado lógico bajo por un mínimo de 6 ciclos de reloj. Cuando ocurre la señal de reset ocurren una serie de acciones: Se termina la ejecución del programa actual, se inicializan los registros de estado del CPU, el contador de programa se sitúa en la localidad cero de la memoria de programa y por último se inicializan los registros de los circuitos periféricos.

Control de las interrupciones

En el C50 las interrupciones son controladas mediante 2 registros mapeados en memoria (localizados en la página 0 de la memoria de datos): el registro de bandera de interrupción (IFR), el registro de enmascaramiento de interrupciones (IMR) y el bit de modo de interrupción INTM (bit 9 del ST0).

Generadores de estados de espera por software

Los generadores de estados de espera (programables mediante software) son utilizados para extender los ciclos externos de bus hasta por 7 ciclos de máquina. Esta operación le permite al C50 interactuar con dispositivos externos lentos que

Registro	Edo. de los bits	Resultado
ST0	INTM = 1 OV = 0	Deshabilita las interrupciones mascarables. Se borra el bit de sobreflujo
ST1	C = 1 CNF = 0 HM = 1 PM = 0 SXM = 1 XF = 1	Se habilita el bit de acarreo. El bloque DARAM B0 es situado en memoria dato. El procesador para la ejecución durante /HOLD. La salida del PREG no sufre ningún corrimiento. Se habilita el modo de extensión de signo. Se pone en alto el pin de bandera externa.
PMST	AVIS = 0 BRAFF = 0 IPTR = 0 MP/MC=(edo. pin) NDX = 0 OVLY = 0 RAM = 0 TRM = 0	La dirección interna de programa se refleja en los pines de dirección (A0-A15). Se deshabilita la repetición de bloque. Se sitúa en la dirección cero al vector de interrupciones. Se verifica el valor de este pin para seleccionar el uso de la ROM interna. Modo C2x compatible habilitado Bloque SARAM no está habilitado en memoria dato. Bloque SARAM no está habilitado en memoria programa. Modo C2x compatible habilitado.
IFR	Tlb = 0	No hay interrupciones pendientes.
CBCR	CENB1 = 0 CENB2 = 0	Buffer circular 1 deshabilitado. Buffer circular 2 deshabilitado.
GREG	Tlb = 0	Se configura todo el espacio de memoria como local.
RPTC	Tlb = 0	El contador de repeticiones se sitúa en 0.

Tabla 4.12: Estado de los registros del CPU al ejecutar un reset.

Interrupt Flag Register (IFF)

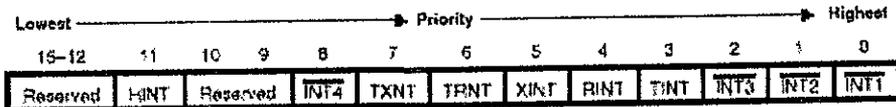


Figura 4.24: Registro de enmascaramiento de interrupciones.

Registro	Edo. de los bits	Resultado
PDWSR	T1b = 1	Todos los registros del espacio de programa y dato generados de estados de espera se cargan con 7.
IOWSR	T1b = 1	Todos los registros de entrada/salida generadores de estados de espera se cargan con 7.
CWSR	BIG = 0	El espacio de entrada/salida se divide en 8 bloques de 8K.
	D = 1	Se habilitan los edos. de espera para el espacio de memoria dato.
	I/O HIGH = 1	Se habilitan los edos. de espera para la mitad superior del espacio de entrada/salida.
	I/O LOW = 1	Se habilitan los edos. de espera para la mitad superior del espacio de entrada/salida.
	P = 1	Se habilitan los edos. de espera para el espacio de memoria programa.
DRR	T1b = 0	Se borra el registro de recepción de datos.
DXR	T1b = 0	Se borra el registro de transmisión de datos
SPC/BSPC	DLB = 0	Se deshabilita el bucle digital.
TSPC	FO = 0	Los datos son transmitidos/recibidos como palabras de 16 bits.
	Free = 0	El reloj serial se detiene después de completar una palabra completa.
	FSM = 0	El puerto serial opera en modo continuo.
	INO = (edo. pin)	Refleja el nivel actual del pin CLKR.
	INI = (edo. pin)	Refleja el nivel actual del pin CLKX.
	MCM = 0	El pin CLKX se configura como pin de entrada.
	RRDY = 0	La seal RRDY se encuentra lista.
	/RRST = 0	La señal de recepción del puerto serial está lista.
	RSRFULL = 0	Se inicializa el registro de bandera de recepción .
	Soft = 0	Se habilita el paro inmediato del reloj serial.
	TDM = 0	El puerto TDM se configura como puerto serial standard.
	TXM = 0	El pin FSX se configura como pin de entrada.
	XRDY = 1	La señal de transmisión está lista.
	/XRST = 0	La señal de transmisión del puerto serial está lista.
	/XSREMPY = 0	Se inicializa el registro de bandera de transmisión.
SPCE	BRE = 0	Se deshabilita el buffer en la recepción.
	BXE = 0	Se deshabilita el buffer en la transmisión.
	CLKDV = 00011	El factor de división para el reloj de transmisión interno se coloca en 3.
	CLKP = 0	El dato es muestreado por el receptor durante el flanco de bajada de CLKR y enviada por el transmisor durante el flanco de subida de CLKX.
	FE = 0	Los datos son transmitidos/recibidos como palabras de 16 bits.
	FIG = 0	La serie de pulsos que preseden a la primera serie reinician la interface del puerto serial.
	FSP = 0	La sincronía de pulsos está activa.
	HALTR = 0	Se inicializa el modo de paro de la retransmisión en el receptor.

Tabla 4.13: Estado de los registros de los circuitos periféricos después de un RESET.

Registro	Edo. de los bits	Resultado
	HALTX = 0	Se inicializa el modo de para de la retransmisión en el transmisor.
	PCM = 0	La modulación por codificación de pulsos está deshabilitada.
	RH = 0	Se inicializa el estado de paro del buffer de recepción.
	XH = 0	Se inicializa el estado de paro del buffer de transmisión.
TIM	Tlb = 1	El temporizador realiza cuenta reversible a partir de FFFFh.
PRD	Tlb = 1	Se deshabilita el temporizador.
TCR	TDDR = 0	Cada ciclo decremента al temporizador en 1.
	TSS = 0	El temporizador se encuentra en modo activo.
HPIC	SMOD 0 - 1	Se encuentra en 0 durante el reset y se pone en 1 cuando la señal de reset se pone en alto.
	HINT = 0	No hay interrupciones (el pin /HINT está en alto).

Tabla 4.14: Continuación de la tabla anterior.

no satisfacen el requerimiento de tiempo de acceso "a toda velocidad" del C50, es decir que el DSP puede generar estados de espera para extender los ciclos de lectura/escritura mediante estos generadores de estados de espera programables. Los dispositivos externos que requieren más de 7 estados de espera utilizan una interfaz con el pin READY para generar los estados de espera necesarios. Los generadores de estados de espera son controlados mediante 2 registros de 16 bits y un registro de control de 5 bits. Cada uno de los tres espacios de memoria (programa, dato, y espacio de entrada/salida) tiene asignado un campo en un registro de estado de espera.

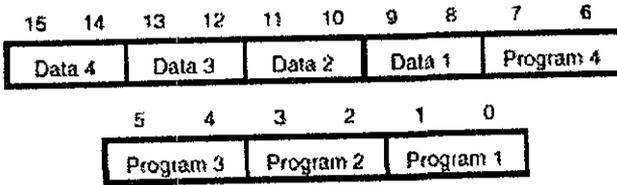
Registro de Estados de Espera de Memoria Programa y Memoria Dato (PDWSR)

Cada uno de los espacios de memoria dato y memoria programa contienen 64K de localidades de memoria las cuales pueden visualizarse como 4 bloques de 16 K. Cada bloque de 16K se asocia con un campo de estados de espera de 2 bits en el PDWSR. El valor dentro de este campo de dos bits ser el número de estados de espera que serán insertados para cada acceso en el rango de direcciones especificado. Después de una reinicialización el PDWSR tiene el valor de FFFFh.

Registros de estados de espera del espacio de entrada/salida (IOWSR)

El IOWSR también contiene 64K direcciones y puede ser configurado de dos maneras, según lo especifique el bit BIG localizado en el registro de control de estados de espera (CWSR). El valor dentro del campo de dos bits del IOWSR especifica el número de estados de espera que serán insertados en cada acceso en

Program/Data Wait-State Register (PDWSR)

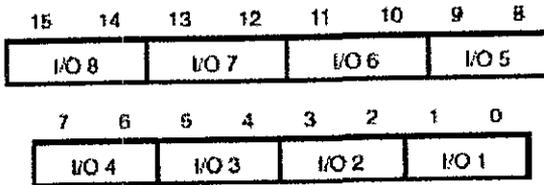


Rangos de Direcciones para el PDWSR

PDWSR Bits	Memory Space	Hex Address Range
15-14	Data 4	C000-FFFF
13-12	Data 3	8000-BFFF
11-10	Data 2	4000-7FFF
9-8	Data 1	0000-3FFF
7-6	Program 4	C000-FFFF
5-4	Program 3	8000-BFFF
3-2	Program 2	4000-7FFF
1-0	Program 1	0000-3FFF

Figura 4.25: El registro PDWSR.

I/O Port Wait-State Register (IOWSR)



Rango de Direcciones para el IOWSR

Wait-State Field Bits	I/O Space	Ports/Hex Address Range	
		BIG = 0	BIG = 1
0-1	I/O 1	Port 0/1, port 10/11, etc.	0000-1FFF
2-3	I/O 2	Port 2/3, port 12/13, etc.	2000-3FFF
4-5	I/O 3	Port 4/5, port 14/15, etc.	4000-5FFF
6-7	I/O 4	Port 6/7, port 16/17, etc.	6000-7FFF
8-9	I/O 5	Port 8/9, port 18/19, etc.	8000-9FFF
10-11	I/O 6	Port 0A/0B, port 1A/1B, etc.	A000-BFFF
12-13	I/O 7	Port 0C/0D, Port 1C/1D, etc.	C000-0FFF
14-15	I/O 8	Port 0E/0F, Port 1E/1F, etc.	E000-FFFF

Figura 4.26: El registro IOWSR.

el rango de direcciones especificada. Después de una reinicialización el IOWSR tiene el valor de FFFFh.

Si el bit BIG está en cero, cada uno de los ocho pares de puertos de entrada/salida mapeados en memoria ser asociado con un campo de dos bits en el IOWSR. Todo el espacio de entrada/salida es configurado con estados de espera por parejas (esto es, puerto 0/1, puerto 10/11 y puerto 20/21 todos tienen el mismo número de estados de espera). Esta configuración nos proporciona una gran flexibilidad cuando se accesan dispositivos periféricos tales como el convertidor A/D a través del espacio de entrada/salida.

Si el bit BIG está en uno, el espacio de 64K se encontrará dividido en ocho bloques de 8K. Cada bloque de 8K en el espacio de entrada/salida estará asociada con un campo de estado de espera de dos bits en el IOWSR. El valor dentro de este campo de dos bits será el número de estados de espera que serán insertados para cada acceso en el rango de direcciones especificado.

Wait-State Control Register (CWSR)

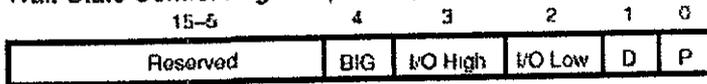


Figura 4.27: CWSR

Registro de control de estado de espera (CWSR)

El CWSR permite seleccionar alguno de los dos tipos de configuración del IOWSR y el número de estados de espera para el espacio correspondiente según el PDWSR y el IOWSR. Si algún bit del CWSR es cero, el número de estados de espera para los accesos externos en ese espacio será el que está indicado por el registro de estado de espera correspondiente. Si un bit del CWSR es un uno, el número de estados de espera para los accesos externos estará dado por: Siempre hay que programar el CWSR antes de configurar el PDWSR y el IOWSR.

Temporizador

El temporizador realiza cuentas regresivas que pueden utilizarse para generar interrupciones periódicas en el CPU. El temporizador es controlado por un pre-escalador el cual se decrementa en 1 por cada ciclo CLKOUT1. La interrupción del temporizador (TINT) se genera cada vez que el contador se decrementa hasta 0. De esta forma, el temporizador ofrece una alternativa conveniente para realizar funciones periódicas de entrada/salida.

Registros del Temporizador

El temporizador es controlado mediante 3 registros:

- Registro de control del temporizador (TCR)
- Registro del contador del temporizador (TIM)
- Registro del período del temporizador (PRD)

Operación del Temporizador

Cuando el PSC se decrementa hasta cero o el temporizador es reinicializado al poner en uno el bit TRB, el contenido del TDDR se carga en el PSC y se decrementa el TIM. Cuando el TIM se decrementa hasta 0 o el temporizador es reinicializado al poner un uno el bit TRB, el TIM se carga con el contenido del PRD. (El bit TRB siempre se lee como 0, aún cuando se le escriba un uno para reinicializar el temporizador).

Diagrama a bloques del Temporizador

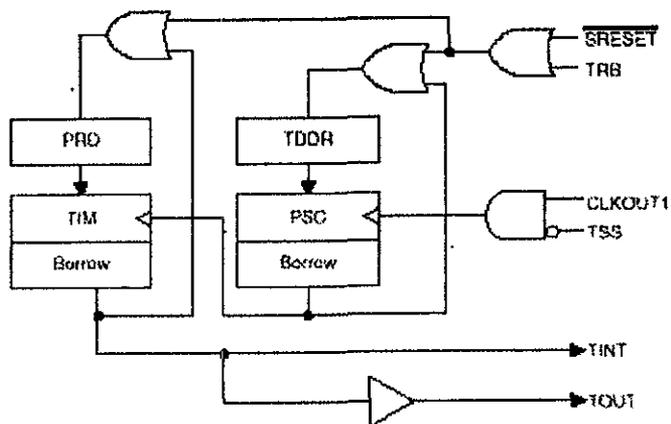


Figura 4.28: Diagrama del temporizador.

Timer Control Register (TCR)

15-12	11	10	9-6	5	4	3-0
Reserved	Soft	Free	PSC	TAB	TSS	YDDR

Figura 4.29: El registro TCR

Tabla 4.15: Registro de control del temporizador

BIT	NOMBRE	*	FUNCION
15-12	Reservado	—	Estos bits estan reservados y se leen como ceros.
11	Soft	0	Este bit se utiliza en conjunción con el bit Free para determinar el estado del temporizador cuando se encuentra un truncamiento (Halt). Cuando el bit Free está en cero, el bit Soft selecciona el modo de emulación. Soft =0 El temp. para inmediatamente, abortando cualquier transmisión. Soft =1 El temp. para después de completar la transmisión.
10	Free	0	Funciona como se explicó anteriormente. Free =0 El bit Soft selecciona el modo del temp. Free =1 El temp. corre libremente, no importando el estado del bit Soft.
9-6	PSC	—	Bits del contador pre-escalador. Estos bits especifican la cuenta del temp. . Cuando el PSC se decrementa hasta 0 ó el temp. es reinicializado, el PSC se carga con el contenido del TDDR, y el TIM se decrementa.
5	TRB	—	Bit para recargar el temp. Este bit reinicializa el temp. Cuando el TRB =1, el TIM se carga con el valor del PRD y el PSC se carga con el valor del TDDR. El TRB siempre se lee como 0.
4	TSS	0	Bit de paro del temp. Este bit para o empieza el temp. TSS =0 El temp. comienza a contar. TSS =1 El temp. para su cuenta.
3-0	TDDR	0000	Estos bits especifican el período del temporizador. Cuando los bits PSC se decremantan hasta 0, el PSC se carga con el contenido del TDDR.

El rango de la interrupción del temporizador (TINT) está dada por:

$$\text{RangoTINT} = \frac{1}{t_c(c) * U * V} = \frac{1}{t_c(c) * (TDDR + 1) * (PRD + 1)}$$

donde $t_c(C)$ es el período de CLKOUT1, U es la suma del contenido de TDDR + 1, y V es la suma del contenido de PRD + 1.

El rango de TINT está dado por dos etapas. Los campos PSC y el TDDR dentro del TCR son utilizados en la primera etapa y el TIM y el PRD son utilizados en la segunda etapa. Cada vez que un contador regresivo (PSC ó TIM) se decrementa hasta 0, se genera un préstamo en el siguiente ciclo CLKOUT1, y el contador regresivo se recarga con el contenido de su correspondiente registro de período (TDDR ó PRD). La salida de la segunda etapa es la seal TINT enviada al CPU y al pin de salida del temporizador (TOUT). El ancho del pulso de préstamo que aparece en la salida de la segunda etapa es igual a $t_c(c)$.

Puerto serial

El C50 ofrece dos tipos de interface de puerto serial, la interface básica standard de puerto serial (SP) y la interfaz de puerto serial TDM. En todos los puertos seriales del C50 se realiza una transmisión full duplex continua de 8 o 16 bits. El modo continuo provee una operacin que, una vez inicializada no requiere de más cadenas de pulsos de sincronizacin (FSR y FSX) cuando se transmite a la máxima frecuencia. La máxima frecuencia de operacin del puerto serial standard es de un cuarto de la frecuencia CLKOUT1 (5M bps a 50 ns, 7.14M bps a 35 ns) al utilizar los relojes internos del puerto.

Capítulo 5

Implementación de un sistema ANC monocanal mediante el C50 DSK.

5.1 Descripción del Controlador

El cancelador de ruido utiliza una topología de control retroalimentada y está diseñado para cancelar ruido periódico. La elección de un control retroalimentado conlleva a un sistema sencillo compuesto solamente por algunos componentes simples (ver Figura 5.1).

El objetivo del sistema es crear una zona de silencio (libre de ruido) alrededor del micrófono. La dimensión de la zona de silencio depende de la frecuencia del ruido a ser atenuado, mientras más baja sea la frecuencia del ruido más grande será esta zona. La dimensión aproximada de la zona de silencio es un décimo de la longitud de onda del ruido.

El sistema puede alcanzar un control global si se logra que la fuente de control (fuente secundaria) genere un campo sonoro que coincida con la distribución espacial del campo sonoro creado por la fuente de ruido (fuente primaria). Es decir, si la fuente de ruido es una bocina que genera un tono puro de 100 Hz, entonces el control global podrá obtenerse si se coloca otra bocina como fuente de control lo suficientemente cerca (comparado con la longitud de onda) de la fuente de ruido. La igualación de la distribución espacial de campo es posible puesto que ambas fuentes son (o casi son) fuentes puntuales.

El problema a ser resuelto es la generación de interferencia acústica para reducir el ruido que capta el micrófono. En la Figura 5.2 se muestra un controlador retroalimentado que utiliza una aproximación feedforward donde H representa la función de transferencia de la planta, o la respuesta al impulso entre la señal

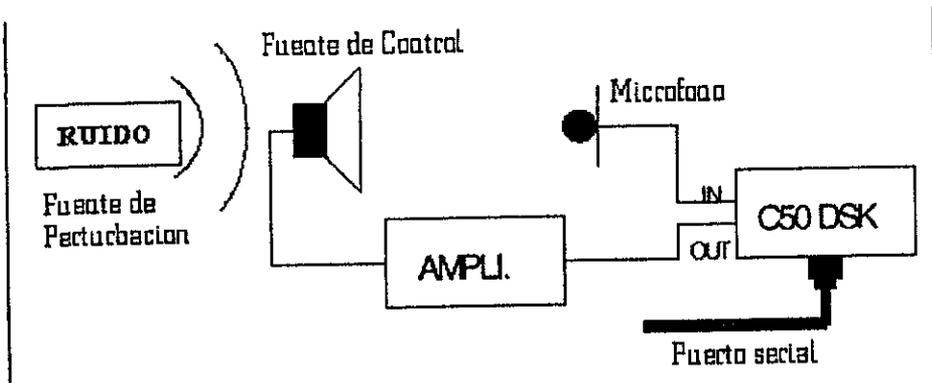


Figura 5.1: Cancelador Activo Adaptivo de ruido de un solo canal.

de salida enviada a la bocina $y(n)$ y la señal de error captada por el micrófono $e(n)$. Un filtro FIR de 256 coeficientes que modela a H (representado por \hat{H}) estima la contribución de la señal de control en el micrófono. Esta estimación es digitalmente restada de la señal del micrófono, resultando la señal $\hat{d}(n)$, la cual representa únicamente a la señal no deseada (ya que la contribución de la señal de control ha sido restada). La salida de la fuente secundaria (la onda de interferencia) es una combinación lineal de la señal $\hat{d}(n)$, el estimado del ruido. El filtro adaptivo W es el filtro de control el cual tiene una estructura transversal. Sus 16 valores son adaptados mediante el algoritmo de filtrado X-LMS. Este filtro tiene dos propósitos:

- Predecir el valor de la señal perturbadora en algún tiempo futuro a partir de los valores de las muestras pasadas y la presente. El tiempo de la predicción no corresponde totalmente al retardo de propagación entre la bocina y el micrófono.
- Generar a partir del valor predecido la muestra de control a ser enviada a la bocina de control. Esta generación toma en cuenta la función de transferencia entre la bocina y el micrófono.

De hecho, ambos filtros están combinados en un filtro de control simple W . Esta descripción nos muestra una limitación de los controladores retroalimentados: para lograr un control total, la señal de disturbio debe ser predecible.

Por lo tanto, los controladores retroalimentados trabajan bien con señales de banda angosta y periódicas, en cambio el desempeño que tienen con una señal de ruido blanco aleatoria es muy pobre.

La Figura 5.3 muestra el proceso de identificación que debe realizarse antes del control mismo para estimar la función de transferencia de la planta \hat{H} .

Durante este proceso de identificación, el C50 genera una señal de ruido blanco la cual es reproducida por la bocina durante unos segundos. La señal

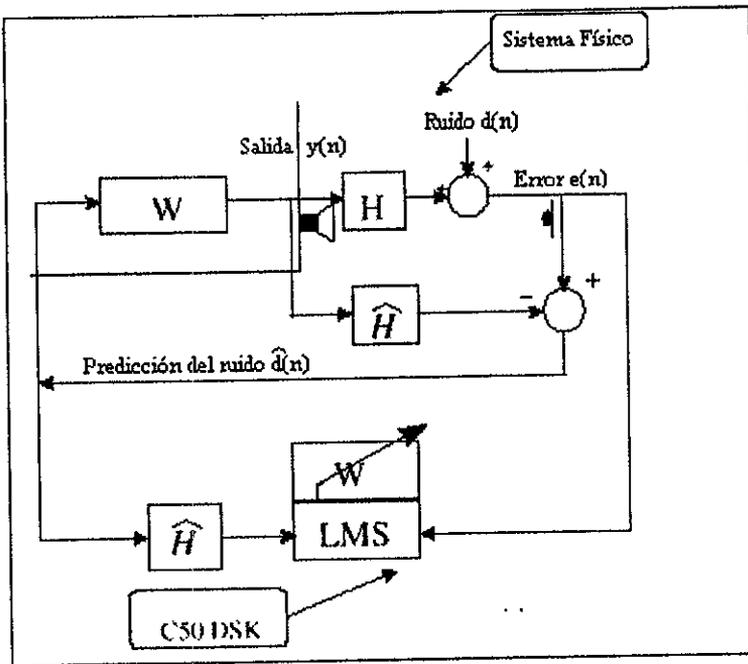


Figura 5.2: Controlador Retroalimentado utilizando una Aproximación Feed-forward

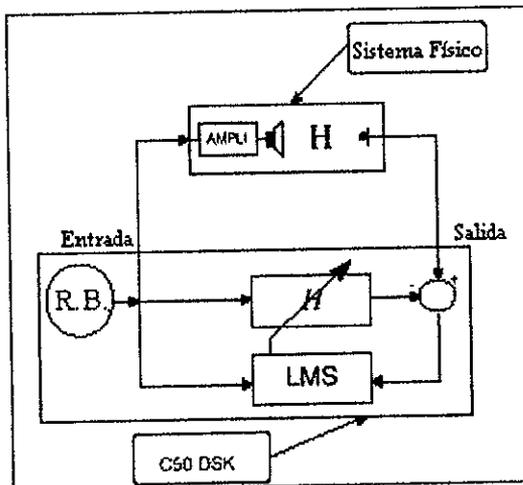


Figura 5.3: Proceso de identificación de la Planta

del micrófono es monitoreada al transcurrir esta excitación, y el algoritmo LMS identifica la función de transferencia H entre la señal de entrada y la de salida. Posteriormente, se realiza el algoritmo de control de la Figura 5.22 donde el algoritmo de filtrado X-LMS adapta W para minimizar la energía de la señal de error en el micrófono. El filtrado X-LMS continúa la adaptación por lo que el controlador puede detectar las variaciones en las frecuencias de las señales que están siendo canceladas, o las variaciones en el contenido espectral (amplitudes de los tonos).

5.1.1 Número de Derivaciones para \hat{H} y W

El número de derivaciones necesario para identificar a la planta depende de la frecuencia de muestreo y de la duración de la respuesta al impulso de la planta. La respuesta al impulso de la planta es la respuesta al impulso entre la bocina y el micrófono. Esta se ve afectada por la reverberación acústica del lugar; por lo tanto, entre más reverberación produzca el lugar más larga será la respuesta al impulso. Si el cancelador es utilizado en una habitación reverberante y además se utiliza una frecuencia de muestreo mayor que la recomendada (600 Hz), se necesitarán más derivaciones para \hat{H} . Sin embargo, para la frecuencia de muestreo de 600 Hz, las 256 derivaciones serán suficientes casi para cualquier tipo de habitación.

Para el control del filtro W , el número de derivaciones será de 16 lo cual es suficiente para controlar hasta 8 tonos. Se necesita un mínimo de 2 coeficientes por tono para que el control de tonos sea adecuado.

5.2 Consideraciones para el Hardware

5.2.1 Distancia entre la bocina y el micrófono

Para que el controlador trabaje bien, la distancia entre el micrófono y la bocina no debe ser cambiada una vez que el proceso de identificación se ha realizado. La distancia utilizada en nuestros experimentos fué de 60cm. Pudimos observar que conforme aumenta la distancia aumenta la inestabilidad del sistema, así que una distancia pequeña es lo más adecuado.

5.2.2 Etapa de Entrada

La etapa de entrada está compuesta por un micrófono dinámico, un filtro pasa bajas y un preamplificador como se muestra en la Figura 5.4.

Una consideración importante en la etapa de entrada es la respuesta en frecuencia del micrófono (**Apéndice C**). Esto es, debido a que si se intenta controlar tonos menores a la frecuencia de corte paso altas del micrófono el desempeño será muy pobre. Para nuestro caso, esa frecuencia está localizada en los 90Hz.

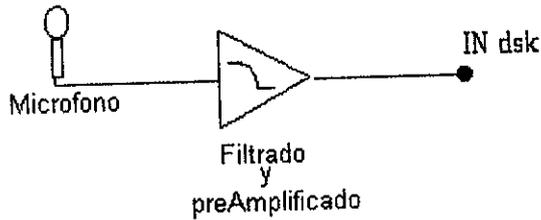


Figura 5.4: Etapa de Entrada.

Para filtrar y darle ganancia a la señal de entrada se utilizó una mezcladora de audio donde podíamos ajustar la ganancia adecuada para obtener los niveles de señal requeridos.

5.2.3 Niveles de la Señal

Para que el cancelador de ruido funcione adecuadamente deben especificarse algunos niveles de señal, como por ejemplo:

- La señal de ruido, antes de aplicar el control, deberá alcanzar de 50 a 200 mV pico al ser entregada por el micrófono.
- Para una entrada al amplificador de 3 V pico (la máxima salida del AIC¹), la ganancia del amplificador debe ser ajustada para obtener por lo menos 200 mV pico entregados por el micrófono. Es decir, que el amplificador debe ser capaz de manipular la bocina para que genere la misma presión de sonido en el micrófono como lo hace el ruido antes de aplicar el control.

5.2.4 Etapa de Salida

Es importante contar con un par de bocinas que tenga una respuesta plana a baja frecuencia, ya que de otra forma la onda generada para cancelar el ruido sera ineficiente por más ganancia que se le de al amplificador.

¹Analog Interface Circuit, convertidor A/D y D/A integrado al C50 DSK

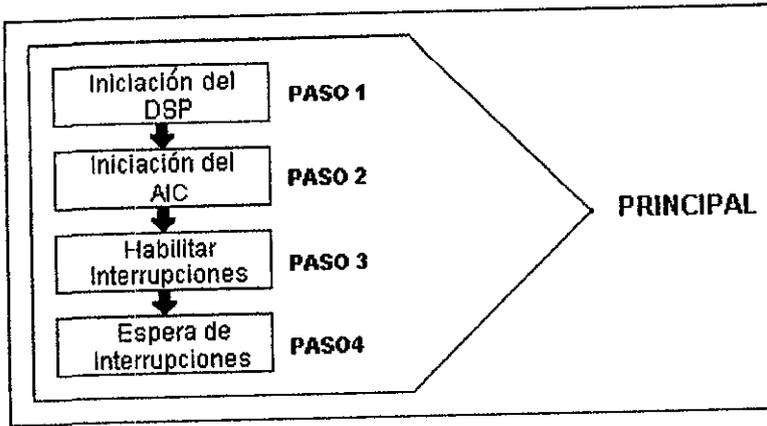


Figura 5.5: Sección principal

5.3 Implementación en el DSK

El código del controlador de ruido se encuentra dentro del procesador C50. Una vez que el código ha sido ensamblado, éste se cargará mediante el depurador (Debugger). El programa del DSP puede dividirse en 2 secciones principales:

- Sección principal. Ver Figura 5.5.
- Interrupción. Ver Figura 5.6

La sección principal ejecuta las tareas de inicialización del DSP, la inicialización del AIC, habilita las interrupciones y la espera las interrupciones.

La sección de interrupción maneja todo el procesamiento en tiempo real, el cual incluye la identificación, el control y la optimización de los procesos. Manipula muestra por muestra de acuerdo a las interrupciones originadas por el AIC. Todos estos procesos se ejecutan en 3 fases, las cuales se muestran a continuación:

Rutinas principales.

5.3.1 Inicialización del C50

El primer paso es inicializar el DSP, es decir, debemos configurar al procesador para que trabaje en el modo que convenga a nuestra aplicación. Dicha configuración incluye especificar la memoria con que se desea trabajar (si utilizaremos memoria SARAM, DARAM interna o externa), modificar los registros de control para trabajar en modo de saturación, modificar registros de estado para realizar corrimientos con el ACC o con el PREG, etc.

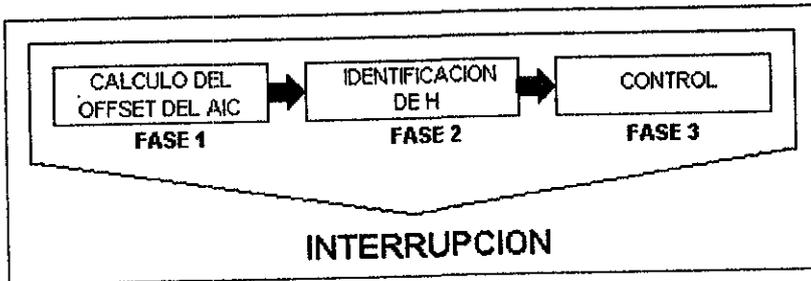


Figura 5.6: Sección de interrupción

5.3.2 Inicialización del AIC

Ya que el convertidor se encuentra dentro de la tarjeta del DSK, éste debe ser inicializado para proporcionar una frecuencia de muestreo deseada, aquí se explicará como modificar los registros que controlan el AIC. Primero, como después del reset el AIC se encuentra en modo síncrono, solamente las variables globales TA, TB y VPRD son utilizadas para configurar el período de muestreo y los filtros anti-aliasing. TA y TB son registros del AIC, los cuales ajustan la frecuencia de muestreo y la frecuencia de corte de sus filtros. El VPRD ajusta la frecuencia del temporizador del C50, cuya salida controla el reloj maestro (Mc) del AIC.

El valor del registro TB ajusta la relación entre la frecuencia de muestreo y la frecuencia de corte del filtro paso bajas. Al ajustar TB a 32, decidimos que la frecuencia de corte (-70 dB) es aproximadamente la mitad de la frecuencia de muestreo. Este valor no debe ser cambiado a menos que se desee modificar la relación entre la frecuencia de muestreo y la frecuencia de corte. Hay que tomar en cuenta que al incrementar la frecuencia de corte pueden ocasionarse problemas de "aliasing". Por lo que solamente TA y VPRD deben ser ajustados para modificar la frecuencia de muestreo. La frecuencia de muestreo (F_s) esta dada por:

$$TA = \frac{M_c}{2 \cdot SCF} = \frac{10^6}{2 \cdot TB \cdot F_s} = \frac{10^6}{2 \cdot 32 \cdot F_s}$$

Si el resultado de TA es menor que 31 (el máximo valor permitido para TA), el registro PRD del C50 puede permanecer en 1 (el valor por default del VPRD que proporciona una frecuencia de reloj maestro de 10 MHz). De otra manera, TA puede modificarse a su valor máximo 31 y se calculará un nuevo valor de

VPRD:

$$VPRD = INT\left(\frac{10^6}{TA \times TB \times F_s} - 1\right) = INT\left(\frac{10^6}{31 \times 32 \times F_s} - 1\right)$$

Los valores recomendados para el VPRD (16), TA (31) y TB(32) proporcionan una frecuencia de muestreo de 600 Hz, una frecuencia de corte paso bajas (-70dB) de 300 Hz y una frecuencia de corte paso altas de 20 Hz. Con estos valores, el cancelador deberá trabajar adecuadamente con señales periódicas con frecuencias entre 100 y 250 Hz.

5.3.3 Habilitación de Interrupciones

Esta etapa habilita la interrupción recibida por el puerto serial del C50. Cuando esta interrupción está activa, y ya que el AIC opera en modo síncrono, una muestra de 16 bits acaba de ser transmitida hacia el AIC y otra ha sido recibida desde el AIC.

5.3.4 Espera de interrupciones

Empieza un bucle infinito de NOP (no operation). El C50 espera a que ocurran interrupciones del puerto serial.

Rutina de interrupción

La rutina de interrupción se divide en 3 fases las cuales se ejecutan secuencialmente, inmediatamente después que el programa principal ha habilitado la interrupción. La primera fase calcula la compensación (offset) de la entrada del AIC. La segunda fase identifica la función de transferencia entre la fuente de control y el micrófono. La tercera fase es precisamente el control.

5.3.5 Cálculo de la compensación de la entrada del AIC

Al observar la señal regresada por el AIC al DSP cuando su entrada esta conectada a tierra, podrá notarse que el promedio de la señal no es cero sino un cierto offset (ver Figura 5.7). Este offset deberá restarsele a cualquier entrada proveniente del AIC. Por esta razón, el primer paso después de habilitar la interrupción del AIC es promediar este offset en 64 muestras de entrada. Ya que el AIC regresa una serie de valores irrelevantes inmediatamente después de su configuración, es necesario dejar transcurrir unas 200 muestras antes de hacer el promedio con las siguientes 64 muestras.

5.3.6 Identificación de H

Una vez que el DSP y el AIC han sido inicializados y que el offset del AIC ha sido calculado, el proceso de identificación de H comienza. Este proceso consiste en

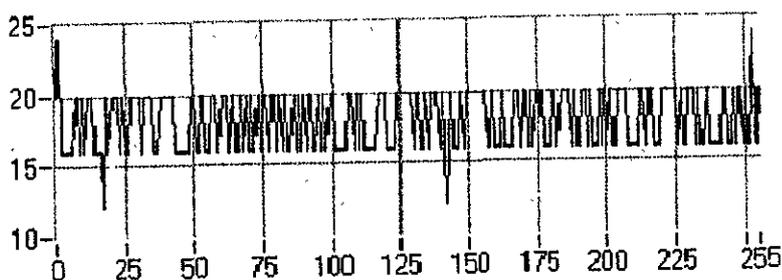


Figura 5.7: Señal regresada por el AIC con la entrada conectada a tierra.

ajustar los 256 coeficientes del filtro FIR (\hat{H}) para minimizar la señal de error (ERR). El algoritmo adaptivo responsable de ajustar estos coeficientes es un simple pero efectivo algoritmo LMS. Para un buen desempeño de identificación, el ruido indeseado no debe estar presente en este proceso. La convergencia del proceso de identificación puede ser medida mediante la relación, en dB, de la energía de la señal de entrada (IN) y la energía del error (ERR).

$$20 \log \frac{E(IN)^2}{E(ERR)^2}$$

Para caracterizar completamente a la planta, se utiliza un ruido blanco como entrada del sistema. La adaptación de los coeficientes se lleva a cabo en ITER iteraciones, lo que significa que las adaptaciones continúan hasta que el contador counter llega al valor ITER. Esta variable se modifica de acuerdo a la precisión que se necesite en la etapa de identificación.

Un número de iteraciones igual a N veces el orden del filtro mejora el desempeño de la identificación por un factor de N (disminuye la amplitud de la señal de error ERR por un factor de N). Por ejemplo, 2560 iteraciones (10 veces el orden del filtro) conlleva a una convergencia de 20 dB mientras que 25600 iteraciones (100 veces el orden del filtro) conlleva a una convergencia de 40 dB. Esto es cierto, sólo si la constante de identificación de ganancia m (variable MUident) es ajustada. El valor por default de MUident es de 15000 para obtener una buena identificación, el nivel regresado por el micrófono durante la fase de identificación debe estar alrededor de 100 mV pico-pico. Para una señal regresada por el micrófono N veces más pequeña/grande que 100 mV, el valor MUident debe ser multiplicado/dividido por N^2 .

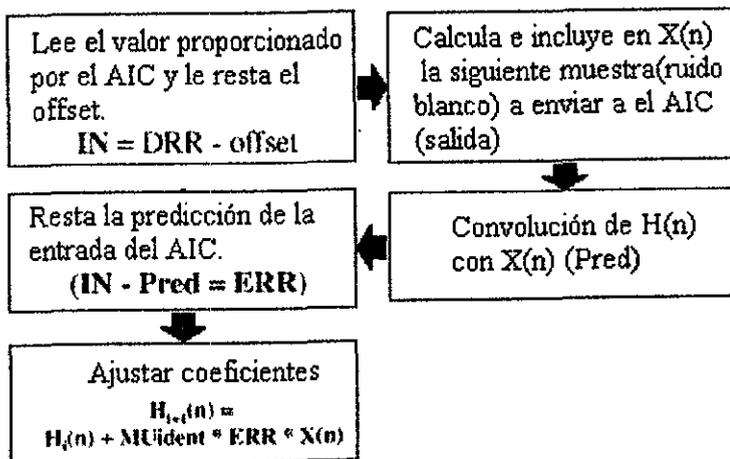
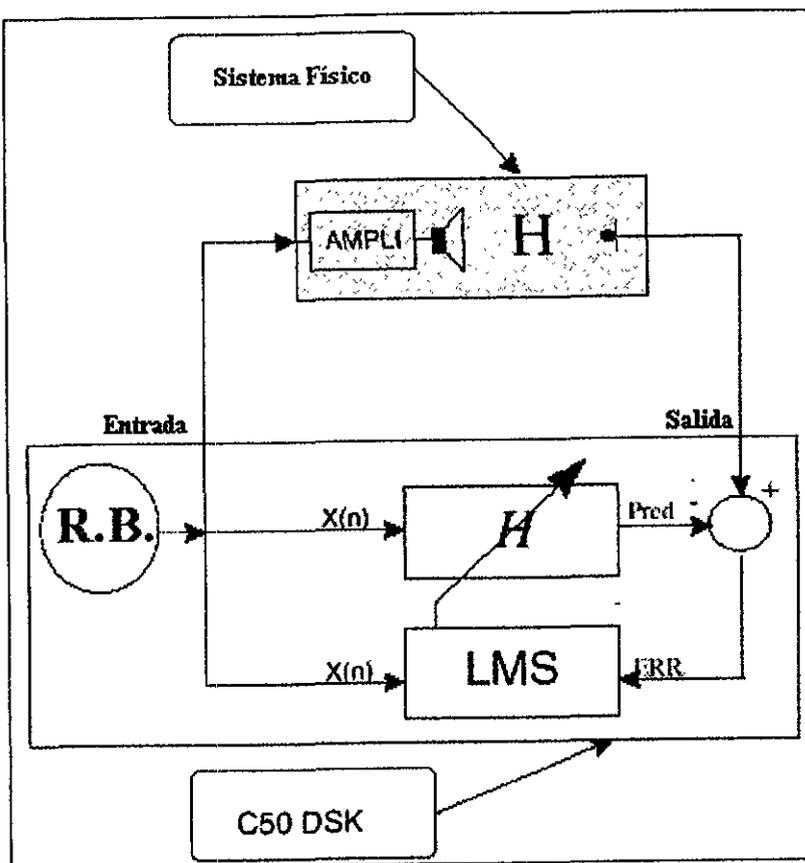


Figura 5.8: Pasos de la fase de identificación. Ejecutada en ITER iteraciones

5.3.7 Control

Después de completar las ITER iteraciones (interrupciones) en la fase de identificación, la fase de control inicia. Esta fase es la última del proceso y se ejecuta hasta que el DSP es reinicializado. El propósito de esta fase es cancelar el ruido en el micrófono adaptando los coeficientes del filtro de control W . El proceso de adaptación es continuo para poder captar los cambios en las características de la señal de control.

Para un control óptimo, la constante de control de ganancia (variable MU-control) necesita ser ajustada. Su valor por default es 6000 el cual lleva a un buen desempeño si el nivel de la señal de disturbio regresada por el micrófono es de alrededor de 100mVpico-pico, sin control. Si dicha señal es mayor se puede disminuir la constante de ganancia.

El parámetro MU control ajusta el tamaño del escalón del algoritmo de filtrado X-LMS. Dándo un valor más grande resultará en una convergencia más rápida, hasta un punto en el cuál el algoritmo divergerá. Una manera de obtener el mejor valor para MU control es encontrar mediante prueba y error el valor más pequeño que hace que el algoritmo diverja. Al dividir este valor entre 2 se obtendrá el valor que hace converger más rápidamente al algoritmo. Dividiendo por 2 nuevamente obtendremos un valor que hará que el algoritmo converja a la mitad de la velocidad pero a 3 dB menos del ruido residual. En el filtro de control W se utilizan dos arreglos. Uno es de 16 bits (cada derivación del filtro es un bit) y el otro es un arreglo de 32 bits. Ambos contienen las 16 derivaciones del filtro, teniendo el último más resolución. El arreglo de 32 bits es utilizado para actualizar los coeficientes, el arreglo de 16 bits es utilizado para realizar convoluciones. Esto permite tener una mejor resolución de los coeficientes para el proceso de adaptación. El arreglo de 16 bits es actualizado guardando solamente los 16 MSB de cada derivación del arreglo de 32 bits.

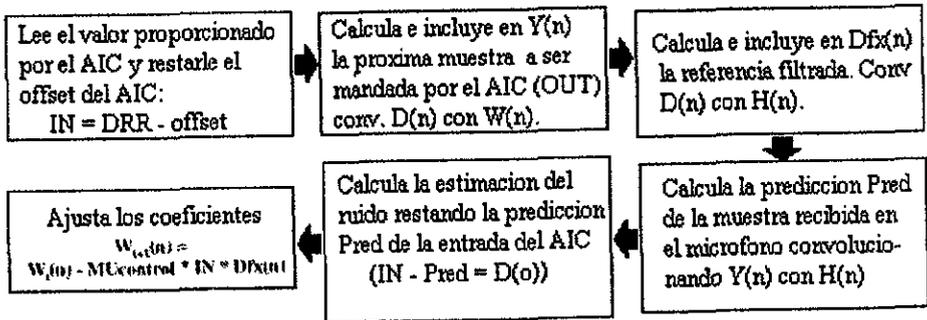
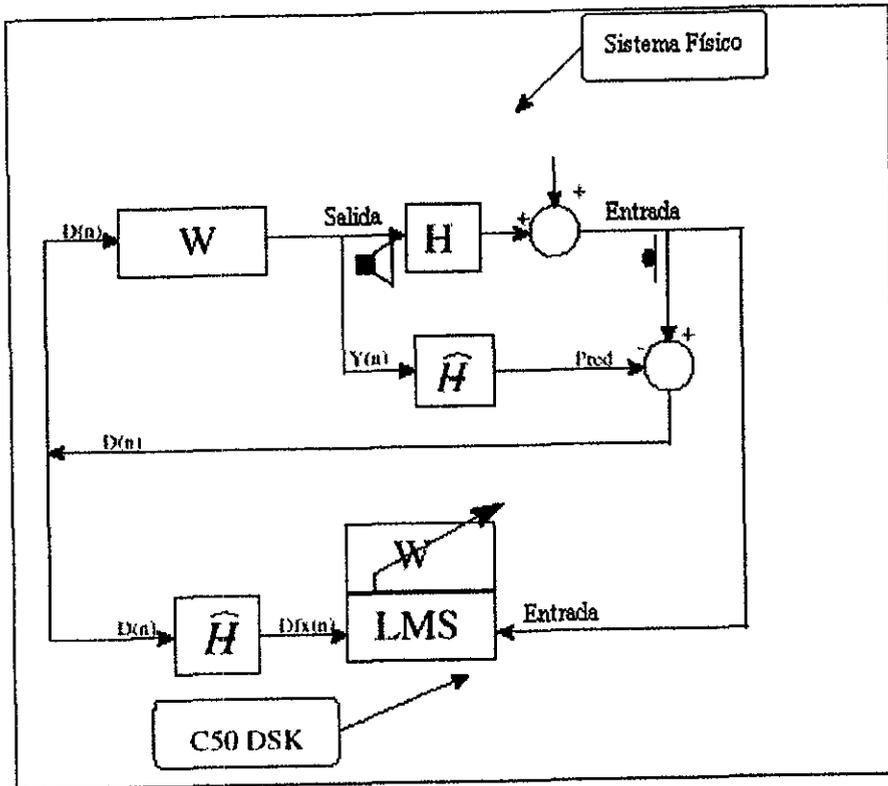


Figura 5.9: Pasos de la Fase de Control, Efectuada después de la identificación y hasta la reinicialización del DSP.

Capítulo 6

Conclusiones.

El control del campo acústico es un problema muy complejo, es por eso que la mayoría de las aplicaciones para ANC se dan en ductos, donde el sonido se comporta como una onda plana. Nuestro proyecto está enfocado a crear una zona de silencio alrededor del micrófono en un espacio cerrado y NO a una reducción global del ruido en dicho espacio. Esto es debido a que un solo actuador es incapaz de cancelar todos los modos de vibración que puede generar un tono puro. Estos modos dependen del lugar donde está localizado el cancelador de ruido.

Mediante niveles adecuados de señal, un par de buenas bocinas, un buen micrófono y un tono puro como señal de disturbio, se deben poder alcanzar 40 dB de reducción en el ruido captado por el micrófono. Esto puede comprobarse en las Figuras B.1 hasta B.6 en el Apéndice B, donde se muestra el espectrograma del ruido residual que capta el micrófono en varias pruebas realizadas. Además comprobamos que la zona de silencio es aproximadamente un decimo de la longitud de onda de la señal de ruido. Esto lo realizamos monitoreando la señal de ruido residual con otro micrófono, el cual acercábamos y alejábamos del micrófono principal. Ver Figura B.6 en el Apéndice B.

Para implementar nuestro sistema en un problema real es necesario hacer un estudio previo sobre la acústica del lugar. Esto es con objeto de encontrar el mejor sitio para colocar la(s) bocina(s) de control. Además, hay que modificar algunos parámetros en el programa, tales como el factor de convergencia (μ), el número de pesos del filtro digital (W) y la frecuencia de muestreo (f_s).

La frecuencia de muestreo puede incrementarse hasta 19200 Hz (máximo valor del AIC), aunque esta frecuencia no es útil para cancelación de ruido acústico debido a que el tamaño de la zona de silencio será muy pequeña. Nuestro sistema permitirá un control efectivo para tonos puros de hasta 7kHz.

Por último queremos resaltar la importancia que tiene el estudio de la física de la Acústica dentro del Control Activo de Ruido, ya que ésta constituye un factor relevante en el desempeño de los sistemas. Durante el desarrollo de nuestro proyecto pudimos darnos cuenta que electrónicamente, el sistema cumple con su cometido (Ver Figura B.7), pero al momento de la interacción con el dominio acústico los efectos sobre el campo no son totales sino parciales, de ahí la importancia de conocer su naturaleza.

Bibliografía

- [1] Craig Marven, Gillian Ewers : *A simple approach to Digital Signal Processing*. Texas Instruments. Great Britain, 1994.
- [2] C.F.N Cowan, P.M. Grant : *Adaptive Filters*. Prentice Hall Signal Processing Series. New Jersey USA, 1985.
- [3] Peter M. Clarkson: *Optimal and Adaptive Signal Processing*. CRC Press. Florida USA, 1993.
- [4] Texas Instruments: *TMS320C5x User's Guide*. Texas Instruments. Missouri USA, 1997.
- [5] Texas Instruments: *TMS320C5x DSP Starter Kit User's Guide*. Texas Instruments. USA, 1994.
- [6] Texas Instruments Application Report: *Design of Active Noise Control Systems with the TMS320 Family*. USA 1996.
- [7] Rulph Chassaing, Darrell W. Horning: *Digital signal Processing with the TMS320C25*. John Wiley & Sons. USA 1990.
- [8] John R. Treichler, C. Richard Johnson Jr., Michael G. Larimore : *Theory and Design of Adaptive Filters*, John Wiley & Sons. USA 1987.
- [9] Nelson and Elliot: *Active Control of Sound*. San Diego: Academic Press, 1992.
- [10] Sen M. Kuo, Dennis R. Morgan: *Active Noise Control Systems. Algorithms and DSP Implementations*. John Wiley & Sons. Series in Telecommunications and Signal Processing. USA 1996
- [11] Leslie Paige Fowler: *Application of the Filtered-X LMS Algorithm for Disturbance Rejection in Time-Periodic Systems*. Master Thesis of Science in Mechanical Engineering. Virginia Polythechnic Institute and State University. Blacksburg, Virginia USA. 1996

- [12] Yifeng Tu: *Multiple Reference Active Noise Control*. Master Thesis of Science in Mechanical Engineering. Virginia Polytechnic Institute and State University. Blacksburg, Virginia USA. 1997
- [13] Mattias Bornhager, Carl Magnus Nilsson: *Active Noise and Vibration Control*. Master Thesis in Computer System Engineering. Center for Computer Architecture, Halmstadt University. Sweden.
- [14] Wolfgang Svensson, Björn Amstrand: *Active Noise Control in Enclosures*. Master Thesis in Computer System Engineering. Center for Computer Architecture, Halmstadt University. Sweden.

Apéndice A

Código en Ensamblador del Sistema ANC.

```
*****
*****
.mmregs
.data ;2000h,

*****
*** Las siguientes variables globales se utilizan en la rutina
*** de interrupcion del puerto serial. Todas ellas deben estar
*** en la misma pagina de datos para evitar recargar constan-
*** temente el "dp" (data page pointer).
*****

ITER          .word 10240          ; Numero de iteraciones para identificar H.
Muident       .word 15000         ; Valor de la ganancia para identificar H.
Mucontrol     .word 6000          ; Valor de la ganancia para la optimizacion
                                           ; del control.
noislvl      .word 2000           ; Nivel del ruido aleatorio bipolar en la
                                           ; identificacion de la fase H.

IN            .word 0
total        .word 0
counter      .word 0
offset       .word 0
PHASE        .word 1              ; Fase de inicio.
OUT          .word 0
ONE          .word 1
ERR          .word 0
Pred         .word 0
ErrxMu      .word 0
InxMu       .word 0

*****
*****
*** Variables Globales para controlar el "AIC" ***
```

TA .word 31 ; Registros del AIC
RA .word 31
TB .word 32
RB .word 32
AIC_CTR .word 29h
VPRD .word 16 ; Registro PRD del C50

*** Variables para la generacion del ruido aleatorio ***

SEED .word 07e6dh ; Alimentacion para el generador
TEMP .word 0 ; de ruido aleatorio

*** Arreglos o Vectores ***

Wprog .set 1c00h ; Direccion en la memoria programa de W(15)
; (necesario para la instruccion MACD).
Wdata .set 1c00h ; Direccion en memoria dato de W(15).
Wdata32 .set 1c10h ; W(15) direccion para los coefs. de 32 bits.
Hprog .set 1c30h ; Direccion en memoria programa de H(255)
; (necesario para la instruccion MACD).
Hdata .set 1c30h ; H(255) direccion en memoria dato.
Y0 .set 1d30h ; Direccion de Y(0).
Y255 .set 1e2fh ; Direccion de Y(255).
Dfx .set 1e30h ; Direccion de Dfx(-1).
Dfx0 .set 1e31h ; Direccion de Dfx(0).
Dfx15 .set 1e40h ; Direccion de Dfx(15).

X0 .set 300h ; Direccion de X(0).
X255 .set 3ffh ; Direccion de X(255).
X256 .set 400h ; Direccion de X(256).
; El vector D(n) utiliza la misma memoria que
; el vector X(n).
D0 .set 300h ; Direccion de D(0).
D16 .set 30fh ; Direccion de D(16).
D255 .set 3ffh ; Direccion de D(255).
.sect "COEFF" ; Reset de los vectores W(n), H(n),
; Y(n) y Dfx(n).
.bes 9232 ; (1e41h-1c00h)palabras * 16bits = 9232 bits
.sect "REF" ; Reset de X(n) y D(n)
.bes 4096

*** Inicializacion de los vectores de la ISR
*** (rutina de servicio de interrupcion)

.sect "VECTORS"

```

.space 2 * 16
int2:      B COMM
.space 4 * 16
rint:      B RINT      ; OA Puerto serial recibe
                    ; interrupcion ve a RINT.
xint:      B TRANSMIT   ; OC Puerto serial transmite
                    ; interrupcion ve a TRANSMIT

```

```

*****
*****
***

```

```

*** PROGRAMA PRINCIPAL
***
*** 1- Inicializacion del DSP.
*** 2- Inicializacion del AIC.
*** 3- Habilitacion de Interrupciones.
*** 4- Espera de Interrupciones.
***

```

```

*****
*****

```

```

.text
SETC INTM
call Init_DSP      ; Inicializacion del DSP.
CALL AICINIT       ; Inicializacion del AIC.
call Enab_INT      ; Habilitacion de Interrupciones.
WAIT nop           ; Espera de Interrupciones.
B WAIT
;

```

```

***** Fin de Programa Principal *****
;

```

```

*****
*** INICIALIZACION DEL TMS320C50 ***
*****

```

```

Init_DSP          LDP #0
OPL #0834h,PMST
LACC #0
SAMB CWSR
SAMB PDWSR
setc OVM           ; OVM = 1
SPM 0             ; PM = 0
setc SXM
setc CNF           ; CNF = 1
RET
;

```

```

***** Fin de la Inicializacion del DSP *****
*****
;
;

```

```

*** INICIALIZACION DEL AIC ***
*****

```

```

AICINIT:SPLK #022h,IMR ; Habilita la interrupcion XINT (syn TX & RX)
SPLK #20h,TCR ; Para generar una seal de 10 MHz a partir de Tout.
lmmr PRD,VPRD
MAR *,ARO
LACC #0008h ; Modo no continuo
SACL SPC ; FSX como entrada
LACC #00c8h ; Palabras de 16 bits
SACL SPC
LACC #080h ; Pulso de reset del AIC, nivel logico bajo.
SACH DXR
SACL GREG
LAR ARO,#OFFFh
RPT #10000 ; se toma un nivel logico alto despues de 10000 ciclos.
LACC *,0,ARO ; (.5ms a 50ns)
SACH GREG ; Restaura el valor del registro GREG a 0000
lacc #2448h ; En caso de que el AIC, por error, este esperando una
sac1 DXR ; comunicacion secundaria, mandamos los valores por
splk #20h,IFR ; default de TA y RA (18h) por lo que el AIC al
; responder vuelve al estado de comunicacion primaria.
;-----

LDP #TA
SETC SXM
LACC TA,9 ; Registros TA y RA inicializados.
ADD RA,2
CALL AIC_2ND
;-----

LDP #TB
LACC TB,9 ; Registros TB y RB inicializados.
ADD RB,2
ADD #02h
CALL AIC_2ND
;-----

LDP #AIC_CTR
LACC AIC_CTR,2 ; Registros de control inicializados.
ADD #03h
CALL AIC_2ND
RET
AIC_2ND: LDP #0
sach DXR
intaic0 bit IFR,10
bcnd intaic0,NTC
splk #20h,IFR
ADD #6h,15 ; 0000 0000 0000 0011 XXXX XXXX XXXX XXXX b
sach DXR
intaic1 bit IFR,10
bcnd intaic1,NTC
splk #20h,IFR
SACL DXR
intaic2 bit IFR,10
bcnd intaic2,NTC
splk #20h,IFR
lacl #0
SACL DXR
intaic3 bit IFR,10
bcnd intaic3,NTC

```

```

splk    #20h,IFR
RET
;
**** Fin de la Inicializacion del AIC ****
*****
;
;
*** HABILITACION DE INTERRUPTONES ***
*****

Enab_INT      SPLK #012h,IMR ; habilita RINT y INT2
CLRC INTM     ; habilita int.
RET
;
**** Fin de Habilitacion de Interrupciones ****

*****
*****
***
*** RUTINAS DE INTERRUPTCION
***
*****
*****
*****
*** Rutina de interrupcion "recepcion del puerto serie" (RINT)
*****

RINT:      ldp #OUT      ; Envia el valor que esta en "OUT" al
lacl OUT   ;registro de transmision de datos (DXR).
samm DXR
cpl #01,PHASE
bcnd Fase1,TC      ; Calculo del Offset del AIC.
cpl #02,PHASE
bcnd Fase2,TC      ; IDENTIFICACION DE H
cpl #03,PHASE
bcnd Fase3,TC      ; CONTROL

*****
*** CALCULO DEL OFFSET DEL AIC ---> FASE 1
*****

Fase1      lacl #0      ; Manda un 0 al registro de transmision del
sacl OUT   ;puerto serie (DXR) mientras calcula el offset.
lacl #1
add total
sacl total
lamm DRR
sacl IN
lacl #200      ; Deja pasar las primeras 200 muestras
NEG          ; para que elAIC se estabilice.
add total
bcnd END_IN,LT
lacl IN

```

```

add total ; Suma 64 muestras en la variable"total".
sac1 total
cpl #263,total
bcnd END_IN,NTC
lacc total
bsar 6 ; Divide total entre 64 para calcular el
; promedio del offset.

sac1 offset
lacl #0
sac1 total
sac1 total
lacl #2 ; Conmuta a la fase de identificacion (fase #2)
sac1 PHASE
B END_IN
;
*** Fin de la Fase "Calculo del Offset del AIC" ***
*****
;
;
*** IDENTIFICACION DE LA FASE H ---> FASE 2
***
*** Variables Globales a configurar:
*** ITER: Numero de iteraciones para efectuar
*** la Identificacion de la fase H.
*** Muident: Valor de la ganancia para identificar H.
*** noislvl: Nivel del ruido aleatorio bipolar.
***
*****

Fase2 ; En esta fase se lee el valor que regresa el
; AIC y se le resta el Offset de entrada del
; AIC: IN = DRR -;offset.
lamm DRR ; Lee el valor que regresa el AIC
sub offset ; Resta el offset de entrada del AIC.
sac1 IN

; Calcula e incluye en X(n) la muestra siguiente
; (Ruido blanco) para que sea enviada al AIC(OUT)
; Generador de Ruido Aleatorio.
LACC SEED,1
XOR SEED
SACL TEMP,2
XOR TEMP
AND #8000h
ADD SEED,16
SACH SEED,1 ; Reduce la salida en un factor de al menos 1/8
LACC SEED,11 ; para prevenir un overflow (sobreflujo).
ROL ; Dependiente del signo de la salida del ruido
LACC noislvl ;aleatorio se manda "noislvl" + o - al AIC
; (ruido ;aleatorio bipolar).
AND #0fffh ; Forza a los 2 LSBs a cero para que el AIC vea
bcnd POS,NC ; estos valores como datos y no como comandos.
neg

POS sac1 OUT ; Almacena en OUT la siguiente interrupcion del DXR

```

```

; (el valor siguiente que se mandara al registro de
; transmision del puerto serie).
; Incluye en X(n) la muestra siguiente que sera enviada.
lar ar3,#X0
mar *,ar3
sac1 *
; Convolucion de H(n) con X(n) (Pred)

zpr
lacc #1,15
mar *,ar3
lar ar3,#X255
FIR rpt #255
macd Hprog,*-apac
sach Pred
; Resta la prediccion de la entrada del AIC
;(IN - Pred ; = ERR)

neg
lar ar3,#IN
add *,16
sach ERR
; Ajusta los coeficientes:
; Hi+1(n) = Hi(n) + Muident * ERR * X(n)
; ERR * Muident

lt ERR
mpy Muident
pac
add ONE,13 ; Redondeo
sach ErrxMu,2
lacc #254 ; Actualiza los 256 Coeficientes.
samm BRCR

lar ar2,#Hdata
lar ar3,#X256
lt ErrxMu
mpy *-,ar2
rptb LOOP-1
ADAPT zalr *,ar3
mpya *-,ar2
sach ** ;Almacena los nuevos coeficientes de H.
LOOP zalr *
apac
sach ** ;Almacena el ultimo coeficiente de H..
lacl #1
add total
sac1 total

INCR lacl ITER
samm DBMR
cpl total
bcnd END_IN,NTC
lacl #0
sac1 OUT
sac1 total
mar *,ar3
lar ar3,#X0
rpt #256
sac1 **,+0

```

```

lacl #3
sac1 PHASE
B END_IN
;
**** Fin de la Identificacion de la Fase H ****
*****
;
;
*** FASE DE CONTROL ---> FASE 3 ***
*** Variables Globales a configurar: ***
*** Mucontrol: Valor de la ganancia para el Control. ***
*** *****

Fase3
; Esta fase lee el valor que regresa el AIC y le resta
;el offset de entrada del AIC: IN = DRR - offset

lamm DRR          ; Lee el valor que regresa el AIC
sub offset        ; Resta el offset de entrada del AIC.
sac1 IN

; Calcula e incluye en Y(n) la muestra siguiente a ser
;enviada al AIC (OUT)obtenida de la convolucion de D(n)
;con W(n).

debut zpr
lacc #1,12        ; Redondeo
mar *,ar3
lar ar3,#D16
FIRC rpt #15      ; Convolucion
mac Wprog,*-apac
AND #Offch,13     ; Forza a que los dos LSBs de la siguiente muestra
; sean cero

sach OUT,3
lar ar3,#Y0       ; Incluye la siguiente muestra a enviar en Y(n)
sach *,3

; Calcula e incluye en Dfx(n) la referencia filtrada
; por la convolucion de D(n) con H(n).

zpr
lacc #1,11        ; Redondeo
lar ar3,#D255
FIRC1 rpt #255    ; Convolucion
macd Hprog,*-apac
lar ar3,#Dfx      ; Incluye la referencia filtrada en Dfx(n)
sach *,4
lar ar3,#Dfx15    ; Recorre el vector de referencia filtrado.
mar *-,ar3        ; Debido a que este vector no se utiliza en alguna
; convolucion, este tiene que ser recorrido manualmente

rpt #14           ; (ultimo en entrar, ultimo en salir)
dmov *-
dmov *

; Calcula la prediccion "Pred" de la muestra recibida
;en el microfono mediante la convolucion de Y(n)
;con H(n).

zpr
lacc #1,15        ; Redondeo

```

```

lar ar3,#Y255          ; La normalizacion que se efectue debe ser la misma
FIRC2 rpt #255        ; que se realizo en la convolucion (FIR) en la
                      ; IDENTIFICACION DE H.
                      ; Convolucion.

macd Hprog,*-
apac
sach Pred

                      ; Calcula el estimado del ruido al restar la prediccion
                      ; "Pred" de la entrada del AIC: D(0) = IN - Pred

neg
lar ar3,#IN
add *,16
lar ar3,#D0
sach *

                      ; Ajusta los coeficientes:
                      ;  $W_{i+1}(n) = W_i(n) - \text{Mucontrol} ; * \text{IN} * \text{Dfx}(n)$ 
                      ;  $\text{IN} * \text{Mucontrol}$ 

lt IN
mpy Mucontrol
pac
add ONE,8             ; Redondeo
sach InxMu,7
lacl #14
samm BRCR
lar ar2,#Wdata32     ; Coeficientes de W de 32 bits.
lar ar3,#Dfx15       ; Referencia filtrada
lar ar4,#Wdata       ; Coeficientes de W de 16 bits.
lt InxMu
mpy *-,ar2
spm 3                ; Origina que las transferencias del registro producto
                    ; (p) hacia la ALU sufran un corrimiento de 6 bits a
                    ; la derecha (sean divididos entre 64). Debido a que el
                    ; resultado de "Mucontrol * IN * Dfx(n)" es dividido
                    ; entre 64, "IN * Mucontrol" puede tomar valores 64
                    ; veces mas grande. Esto permite obtener resultados
                    ; diferentes de cero para los valores de IN que sean
                    ; 64 veces mas pequenos.

rptb LOO-1
ADAPTC lacc **,+16   ; Carga el acumulador con un coeficiente de W de 32
                    ; bits.

or *-,ar3
mpys *-,ar4
sach **,+ar2         ; Almacena un nuevo coeficiente W en el vector W
                    ; de 16 bits.

sach **              ; Almacena un nuevo coeficiente W en el vector W
                    ; de 32 bits.

sacl ** ;
LOO lacc **,+16
or *-,ar4
spac
spm 0
sach **,+ar2        ; Almacena el ultimo (reciente) coeficiente
                    ; W en el vector W de 16 bits.

sach **             ; Almacena el ultimo (reciente) coeficiente W
                    ; W en el vector W de 32 bits.

sacl ** ;
B END_IN

```

***** Fin de la Fase de Control *****

END_IN RETE

;

***** Fin de la rutina RINT *****

;

*** Otras Rutinas de Interrupcion. ***

COMM: RETE

TRANSMIT: RETE

.end

/* Archivo "linker command" para el codigo dsk */
-o anc.out /* Especifica el archivo de salida */
-m anc.map /* Genera el archivo mapa */
anc.obj /* Archivo objeto para realizar el link */
MEMORY

{

PAGE 0 : VECT: origin = 00802h, length = 003dh

PAGE 0 : PROG: origin = 00a00h, length = 1200h

PAGE 0 : UPG : origin = 01c00h, length = 0300h

PAGE 1 : DAT : origin = 02000h, length = 0c00h

PAGE 1 : B1 : origin = 00300h, length = 0100h

}

SECTIONS

{

.text : {} > PROG PAGE 0

VECTORS: {} > VECT PAGE 0

COEFF : {} > UPG PAGE 0

.data : {} > DAT PAGE 1

REF : {} > B1 PAGE 1

Apéndice B

Gráficas de los resultados.

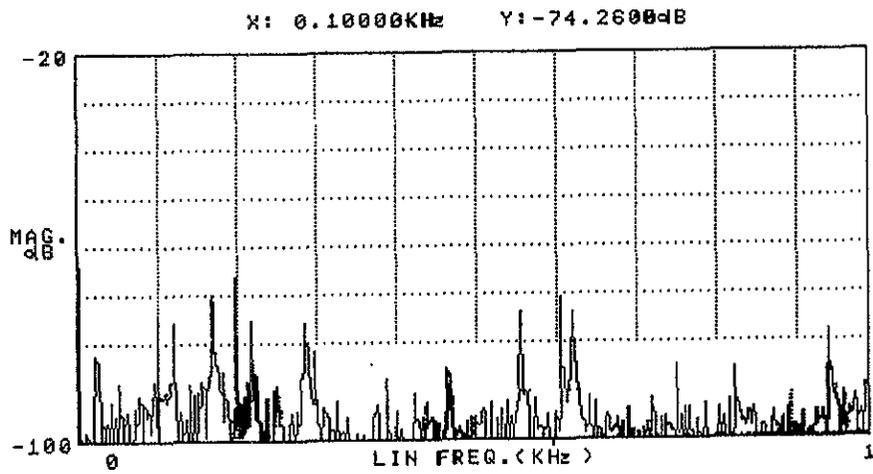
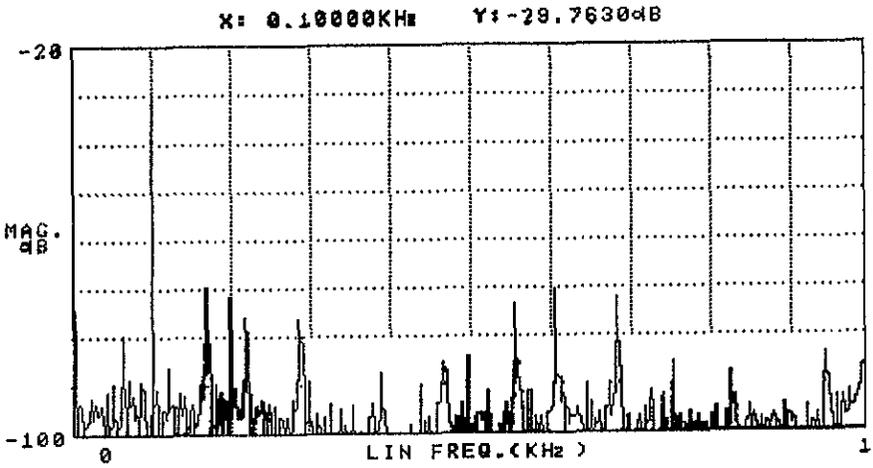


Figura B.1: Atenuación de un tono de 100 Hz.

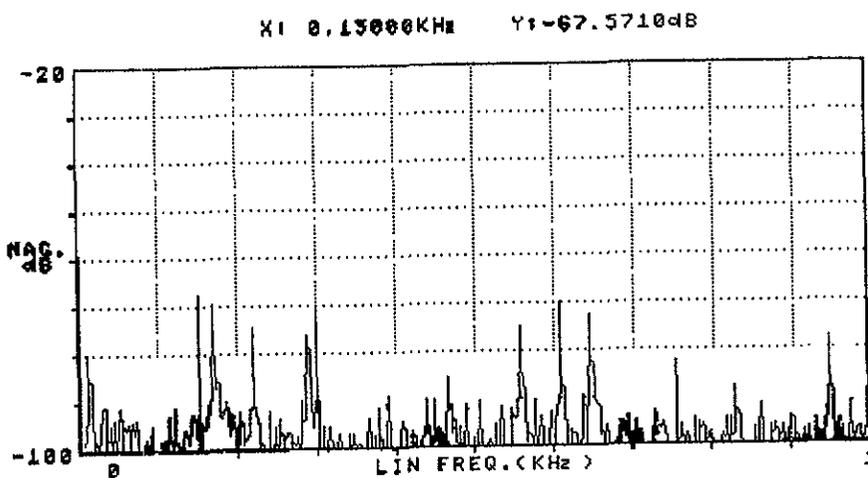
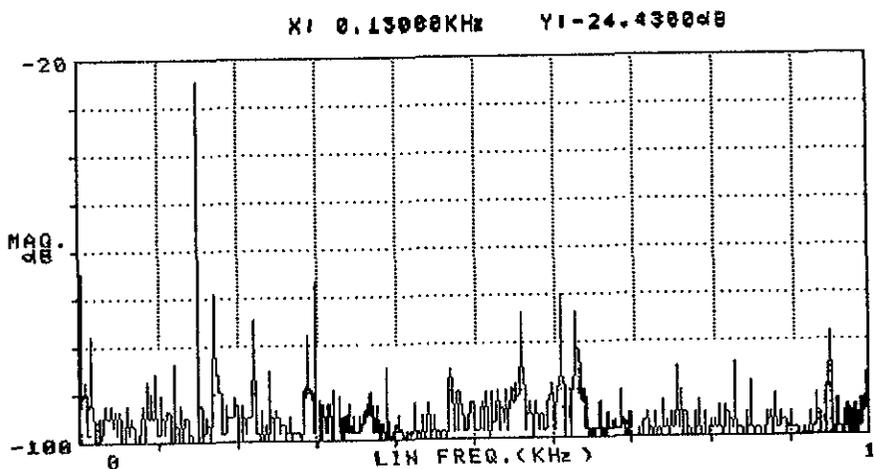


Figura B.2: Atenuación de un tono de 150 Hz.

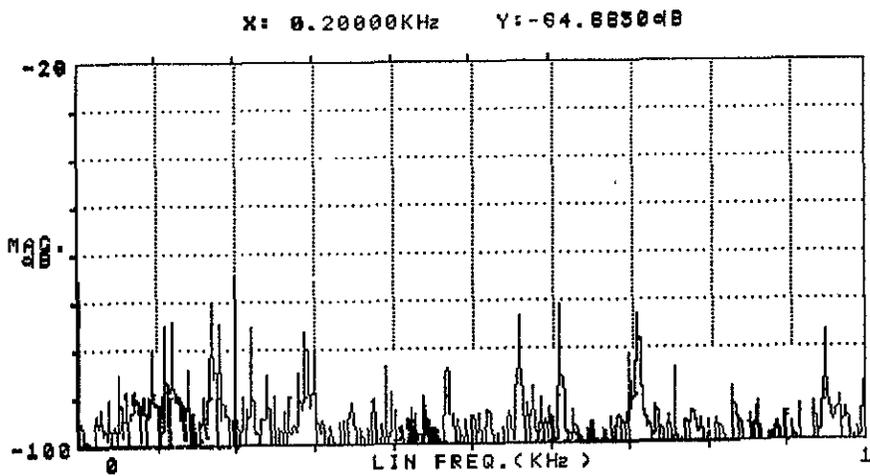
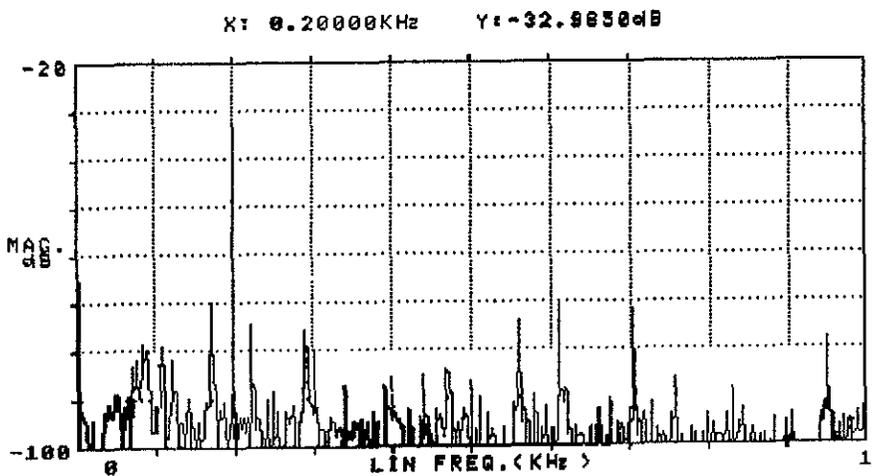


Figura B.3: Atenuación de un tono de 200 Hz.

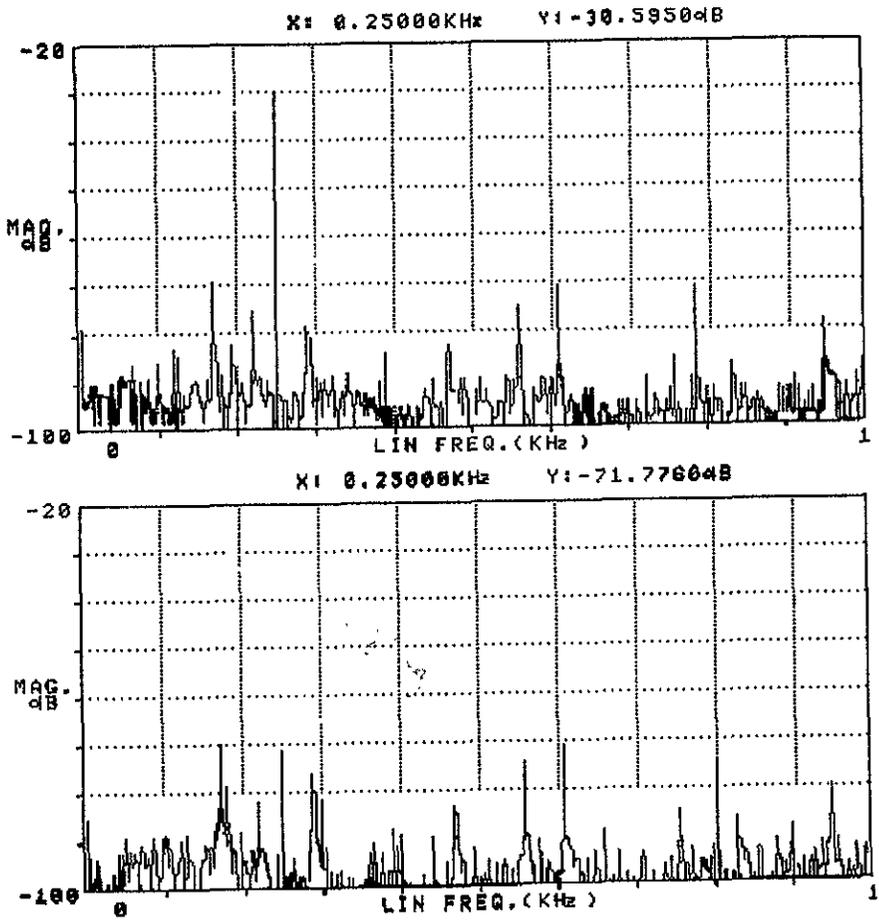


Figura B.4: Atenuación de un tono de 250 Hz.

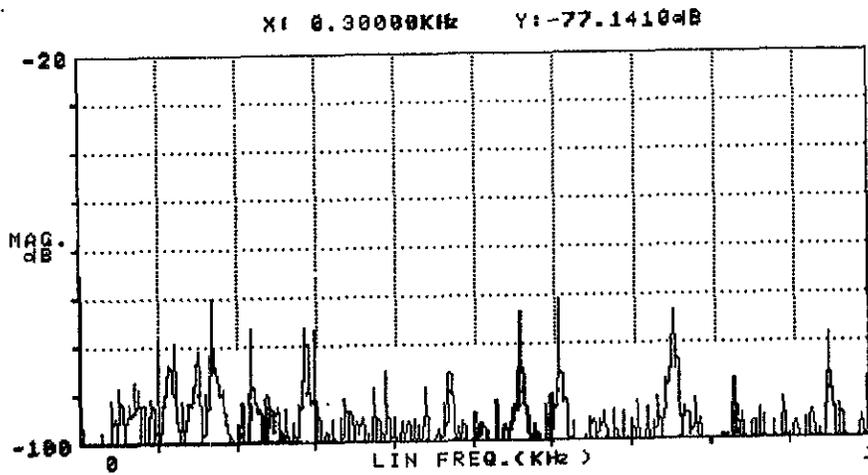
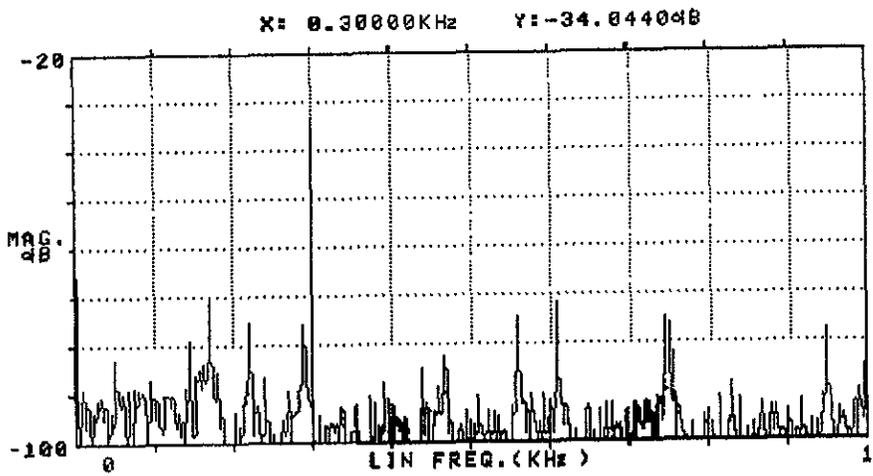


Figura B.5: Atenuación de un tono de 300 Hz.

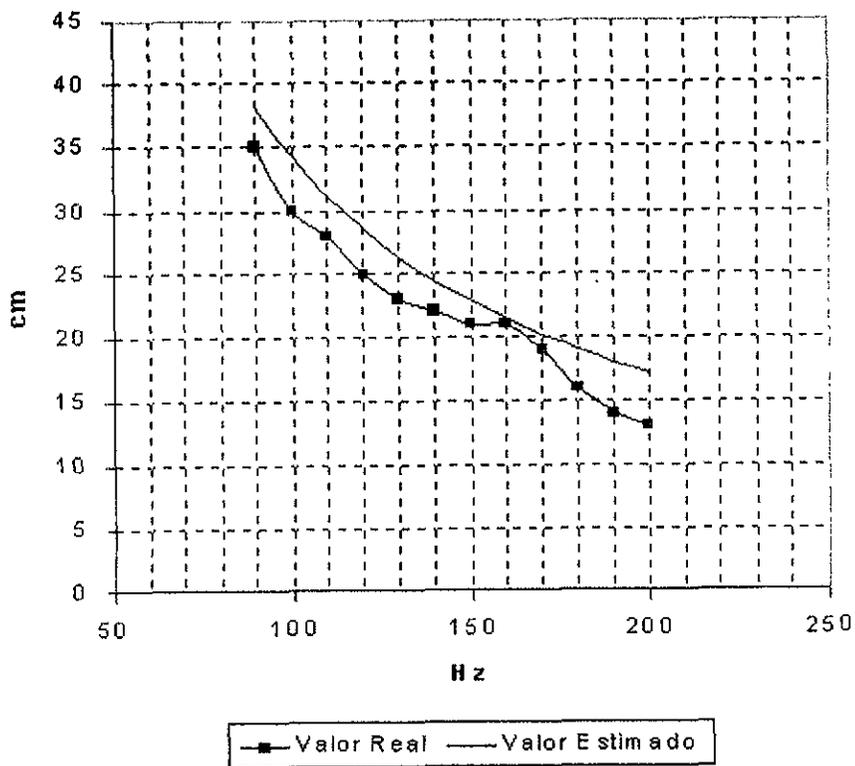


Figura B.6: Diámetro de la zona de silencio.

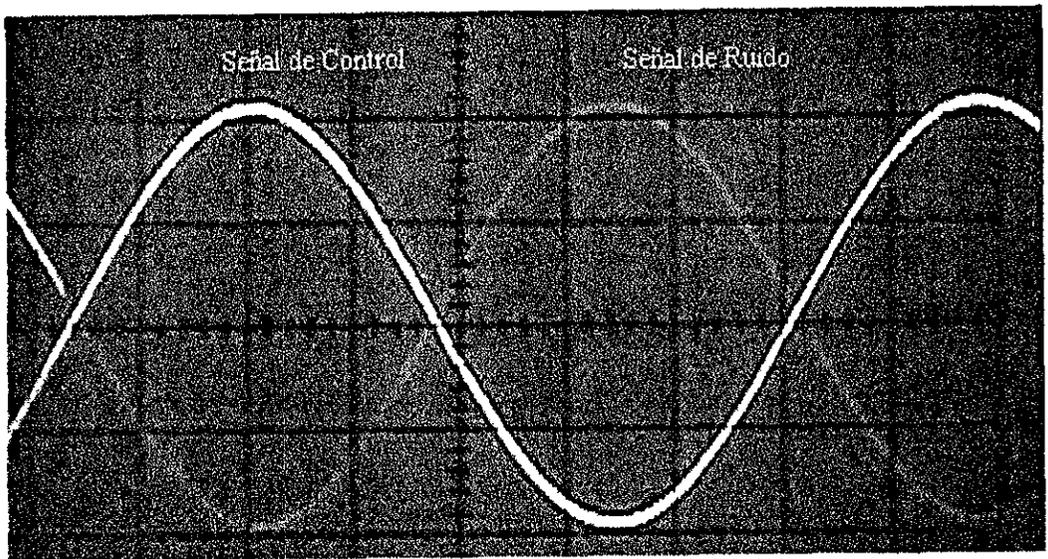


Figura B.7: El sistema funcionando en el dominio eléctrico.

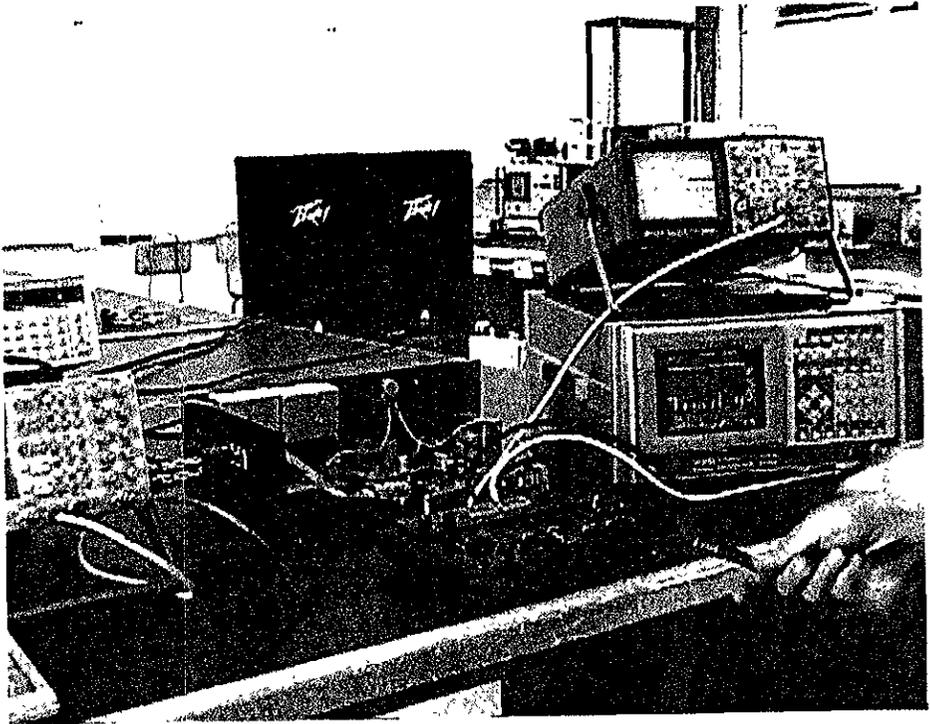


Figura B 8: Sistema ANC monocanal.

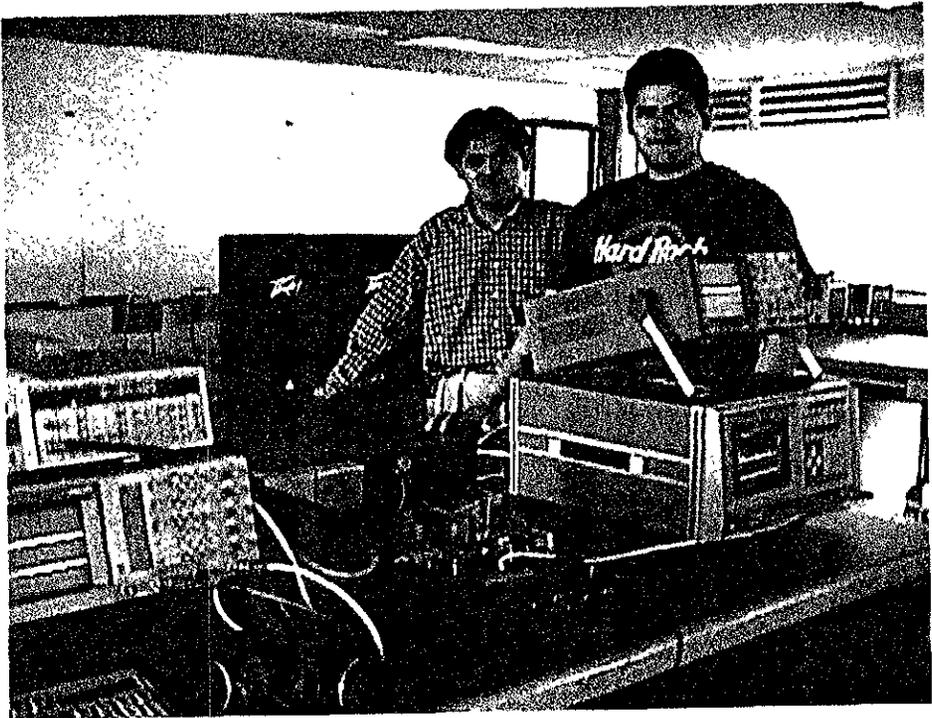
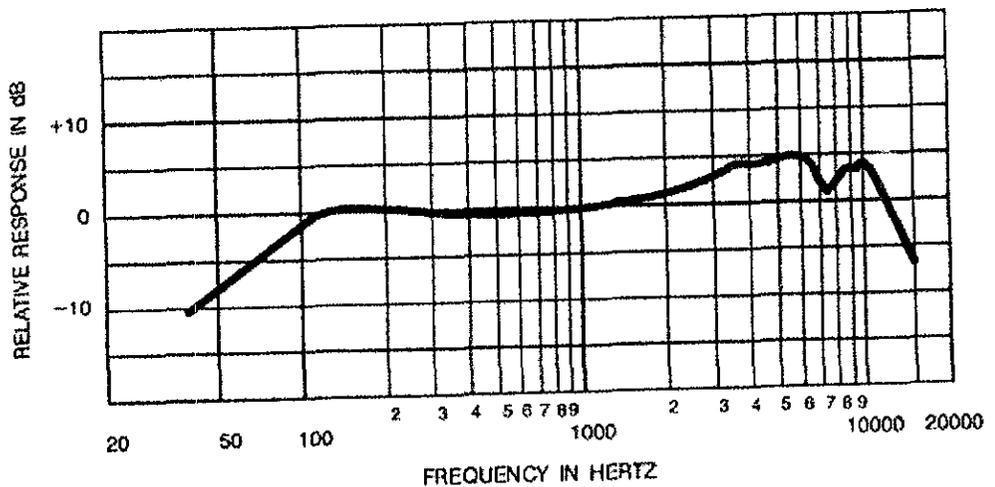


Figura B.9: Trabajo en el laboratorio.

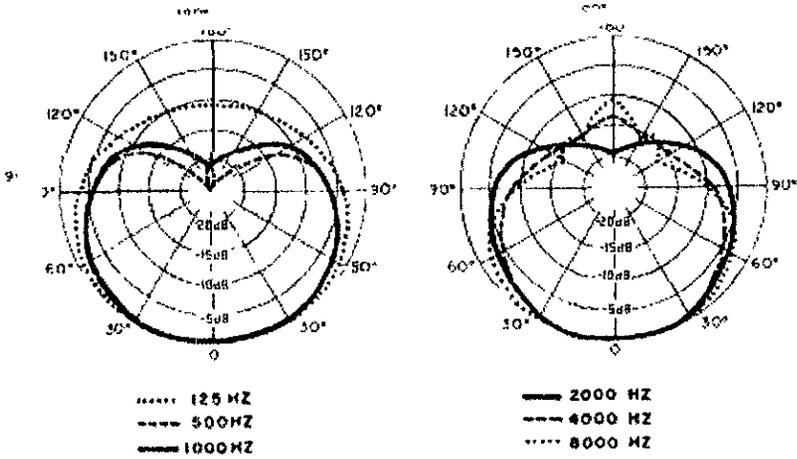
Apéndice C

Especificaciones Técnicas del micrófono



TYPICAL FREQUENCY RESPONSE

Figura C.1: Diagrama de Bode.



TYPICAL POLAR PATTERNS

Figura C.2: Diagrama Polar.

Apéndice D

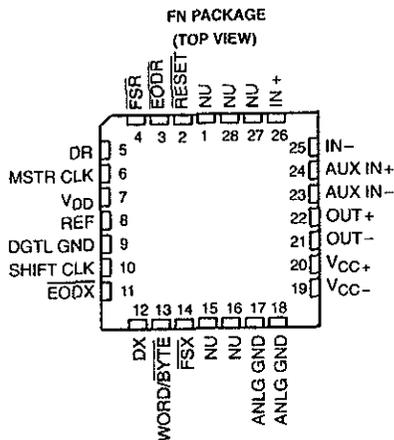
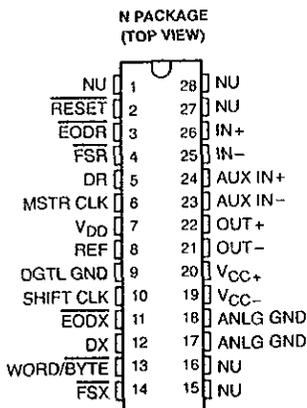
Hoja de especificaciones del TLC32040 AIC

- 14-Bit Dynamic Range ADC and DAC
- Variable ADC and DAC Sampling Rate Up to 19,200 Samples per Second
- Switched-Capacitor Antialiasing Input Filter and Output-Reconstruction Filter
- Serial Port for Direct Interface to TMS32011, TMS320C17, TMS32020, and TMS320C25 Digital Signal Process
- Synchronous or Asynchronous ADC and DAC Conversion Rate With Programmable Incremental ADC and DAC Conversion Timing Adjustments
- Serial Port Interface to SN74299 Serial-to-Parallel Shift Register for Parallel Interface to TMS32010, TMS320C15, or Other Digital Processors
- 600-Mil Wide N Package (C_L to C_L)
- 2s Complement Format
- CMOS Technology

PART NUMBER	DESCRIPTION
TLC32040	Analog interface circuit with internal reference. Also a plug-in replacement for TLC32041.
TLC32041	Analog interface circuit without internal reference

description

The TLC32040 and TLC32041 are complete analog-to-digital and digital-to-analog input/output systems, each on a single monolithic CMOS chip. This device integrates a bandpass switched-capacitor antialiasing input filter, a 14-bit-resolution A/D converter, four microprocessor-compatible serial port modes, a 14-bit-resolution D/A converter, and a low-pass switched-capacitor output-reconstruction filter.



NU - Nonusable; no external connection should be made to these terminals.

AVAILABLE OPTIONS

T _A	PACKAGE	
	PLASTIC CHIP CARRIER (FN)	PLASTIC DIP (N)
0°C to 70°C	TLC32040CFN TLC32041CFN	TLC32040CN TLC32041CN
-40°C to 85°C		TLC32040IN TLC32041IN



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA Information is 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.



TLC32040, TLC32041, TLC32041C, TLC32041I ANALOG INTERFACE CIRCUITS

SLAS014E - SEPTEMBER 1987 - REVISED MAY 1995

description (continued)

The device offers numerous combinations of master clock input frequencies and conversion/sampling rates, which can be changed via digital processor control.

Typical applications for this integrated circuit include modems (7.2-, 8-, 9.6-, 14.4-, and 19.2-kHz sampling rate), analog interface for digital signal processors (DSPs), speech recognition/storage systems, industrial process control, biomedical instrumentation, acoustical signal processing, spectral analysis, data acquisition, and instrumentation recorders. Four serial modes, which allow direct interface to the TMS32011, TMS320C17, TMS32020, and TMS320C25 digital signal processors, are provided. Also, when the transmit and receive sections of the analog interface circuit (AIC) are operating synchronously, it can interface to two SN74299 serial-to-parallel shift registers. These serial-to-parallel shift registers can then interface in parallel to the TMS32010, TMS320C15, other digital signal processors, or external FIFO circuitry. Output data pulses are emitted to inform the processor that data transmission is complete or to allow the DSP to differentiate between two transmitted bytes. A flexible control scheme is provided so that the functions of this integrated circuit can be selected and adjusted coincidentally with signal processing via software control.

The antialiasing input filter comprises seventh-order and fourth-order CC-type (Chebyshev/elliptic transitional) low-pass and high-pass filters, respectively and a fourth-order equalizer. The input filter is implemented in switched-capacitor technology and is preceded by a continuous time filter to eliminate any possibility of aliasing caused by sampled data filtering. When no filtering is desired, the entire composite filter can be switched out of the signal path. A selectable, auxiliary, differential analog input is provided for applications where more than one analog input is required.

The A/D and D/A converters each have 14 bits of resolution. The A/D and D/A architectures ensure no missing codes and monotonic operation. An internal voltage reference is provided on the TLC32040 to ease the design task and to provide complete control over the performance of this integrated circuit. The internal voltage reference is brought out to a terminal and is available to the designer. Separate analog and digital voltage supplies and grounds are provided to minimize noise and ensure a wide dynamic range. Also, the analog circuit path contains only differential circuitry to keep noise to an absolute minimum. The only exception is the DAC sample and hold, which utilizes pseudo-differential circuitry.

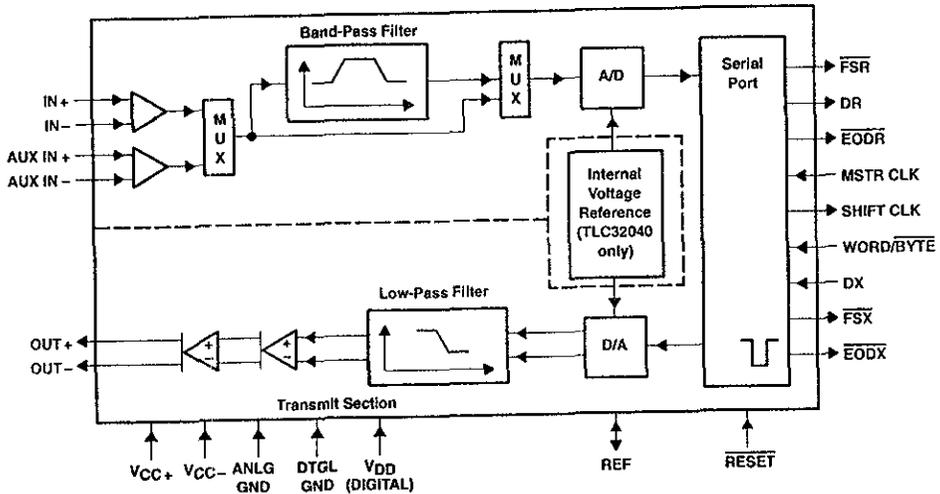
The output-reconstruction filter is a seventh-order CC-type (Chebyshev/elliptic transitional low-pass filter followed by a fourth-order equalizer) and is implemented in switched-capacitor technology. This filter is followed by a continuous-time filter to eliminate images of the digitally encoded signal.

The TLC32040C and TLC32041C are characterized for operation from 0°C to 70°C, and the TLC32040I and TLC32041I are characterized for operation from -40°C to 85°C.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

functional block diagram



Terminal Functions

TERMINAL NAME	NO.	I/O	DESCRIPTION
ANLG GND	17,18		Analog ground return for all internal analog circuits. Not internally connected to DTGL GND.
AUX IN+	24	I	Noninverting auxiliary analog input state. This input can be switched into the bandpass filter and A/D converter path via software control. If the appropriate bit in the control register is a 1, the auxiliary inputs replace the IN+ and IN- inputs. If the bit is a 0, the IN+ and IN- inputs are used (see the AIC DX data word format section).
AUX IN-	23	I	Inverting auxiliary analog input (see the above AUX IN+ description)
DTGL GND	9		Digital ground for all internal logic circuits. Not internally connected to ANLG GND.
DR	5	O	DR is used to transmit the ADC output bits from the AIC to the TMS320 serial port. This transmission of bits from the AIC to the TMS320 serial port is synchronized with the SHIFT CLK signal.
DX	12	I	DX is used to receive the DAC input bits and timing and control information from the TMS320. This serial transmission from the TMS320 serial port to the AIC is synchronized with the SHIFT CLK signal.
EODR	3	O	End of data receive. See the WORD/BYTE description and the Serial Port Timing diagrams. During the word-mode timing, EODR is a low-going pulse that occurs immediately after the 16 bits of A/D information have been transmitted from the AIC to the TMS320 serial port. EODR can be used to interrupt a microprocessor upon completion of serial communications. Also, EODR can be used to strobe and enable external serial-to-parallel shift registers, latches, or external FIFO RAM, and to facilitate parallel data bus communications between the AIC and the serial-to-parallel shift registers. During the byte-mode timing, EODR goes low after the first byte has been transmitted from the AIC to the TMS320 serial port and is kept low until the second byte has been transmitted. The TMS32011 or TMS320C17 can use this low-going signal to differentiate between the two bytes as to which is first and which is second. EODR does not occur after secondary communication

TL0320700, TL0320701, TL0320710, TL0320711
ANALOG INTERFACE CIRCUITS

SLAS014E - SEPTEMBER 1987 - REVISED MAY 1995

Terminal Functions (continued)

TERMINAL NAME	NO.	VO	DESCRIPTION
EODX	11	O	End of data transmit. See the WORD/BYTE description and the Serial Port Timing diagram. During the word-mode timing, EODX is a low-going pulse that occurs immediately after the 16 bits of D/A converter and control or register information have been transmitted from the TMS320 serial port to the AIC. EODX can be used to interrupt a microprocessor upon the completion of serial communications. Also, EODX can be used to strobe and enable external serial-to-parallel shift registers, latches, or an external FIFO RAM, and to facilitate parallel data-bus communications between the AIC and the serial-to-parallel shift registers. During the byte-mode timing, EODX goes low after the first byte has been transmitted from the TMS320 serial port to the AIC and is kept low until the second byte has been transmitted. The TMS32011 or TMS320C17 can use this low-going signal to differentiate between the two bytes as to which is first and which is second.
FSR	4	O	Frame sync receive. In the serial transmission modes, which are described in the WORD/BYTE description, FSR is held low during bit transmission. When FSR goes low, the TMS320 serial port begins receiving bits from the AIC via DR of the AIC. The most significant DR bit is present on DR before FSR goes low. (See Serial Port Timing and Internal Timing Configuration diagrams.) FSR does not occur after secondary communication.
FSX	14	O	Frame sync transmit. When FSX goes low, the TMS320 serial port begins transmitting bits to the AIC via DX of the AIC. In all serial transmission modes, which are described in the WORD/BYTE description, FSX is held low during bit transmission (see the Serial Port Timing and Internal Timing Configuration diagrams).
IN+	26	I	Noninverting input to analog input amplifier stage
IN-	25	I	Inverting input to analog input amplifier stage
MSTR CLK	6	I	Master clock. MSTR CLK is used to derive all the key logic signals of the AIC, such as the shift clock, the switched-capacitor filter clocks, and the A/D and D/A timing signals. The Internal Timing Configuration diagram shows how these key signals are derived. The frequencies of these key signals are synchronous submultiples of the master clock frequency to eliminate unwanted aliasing when the sampled analog signals are transferred between the switched-capacitor filters and the A/D and D/A converters (see the Internal Timing Configuration).
OUT+	22	O	Noninverting output of analog output power amplifier. OUT+ can drive transformer hybrids or high-impedance loads directly in either a differential or a single-ended configuration.
OUT-	21	O	Inverting output of analog output power amplifier. OUT- is functionally identical with and complementary to OUT+.
REF	8	VO	Internal voltage reference for the TLC32040. For the TLC32040 and TLC32041 an external voltage reference can be applied to this terminal.
RESET	2	I	Reset. A reset function is provided to initialize the TA, TA', TB, RA, RA', RB, and control registers. This reset function initiates serial communications between the AIC and DSP. The reset function initializes all AIC registers including the control register. After a negative-going pulse on RESET, the AIC registers are initialized to provide an 8-kHz data conversion rate for a 5.184-MHz master clock input signal. The conversion rate adjust registers, TA' and RA', are reset to 1. The control register bits are reset as follows (see AIC DX data word format section): d7 = 1, d6 = 1, d5 = 1, d4 = 0, d3 = 0, d2 = 1 This initialization allows normal serial-port communication to occur between AIC and DSP.
SHIFT CLK	10	O	Shift clock. SHIFT CLK is obtained by dividing the master clock signal frequency by four. SHIFT CLK is used to clock the serial data transfers of the AIC, described in the WORD/BYTE description below (see the Serial Port Timing and Internal Timing Configuration diagrams).
VDD	7		Digital supply voltage, 5 V ±5%
VCC+	20		Positive analog supply voltage, 5 V ±5%
VCC-	19		Negative analog supply voltage, -5 V ±5%



POST OFFICE BOX 655303 • DALLAS TEXAS 75265

Terminal Functions (continued)

TERMINAL NAME NO.	I/O	DESCRIPTION
WORD/BYTE 13	I	<p>WORD/BYTE, in conjunction with a bit in the control register, is used to establish one of four serial modes. These four serial modes are described below.</p> <p><i>AIC transmit and receive sections are operated asynchronously</i></p> <p>The following description applies when the AIC is configured to have asynchronous transmit and receive sections. If the appropriate data bit in the control register is a 0 (see the AIC DX data word format section), the transmit and receive sections are asynchronous</p> <ul style="list-style-type: none"> L Serial port directly interfaces with the serial port of the TMS32011 or TMS320C17 and communicates in two 8-bit bytes. The operation sequence is as follows (see Serial Port Timing diagrams): <ol style="list-style-type: none"> 1. \overline{FSX} or \overline{FSR} is brought low. 2. One 8-bit byte is transmitted or one 8-bit byte is received 3. \overline{EODX} or \overline{EODR} is brought low 4. \overline{FSX} or \overline{FSR} emits a positive frame-sync pulse that is four shift clock cycles wide 5. One 8-bit byte is transmitted or one 8-bit byte is received 6. \overline{EODX} or \overline{EODR} is brought high. 7. \overline{FSX} or \overline{FSR} is brought high H Serial port directly interfaces with the serial port of the TMS32020, TMS320C25, or TMS320C30 and communicates in one 16-bit word. The operation sequence is as follows (see Serial Port Timing diagrams): <ol style="list-style-type: none"> 1. \overline{FSX} or \overline{FSR} is brought low. 2. One 16-bit word is transmitted or one 16-bit word is received. 3. \overline{FSX} or \overline{FSR} is brought high. 4. \overline{EODX} or \overline{EODR} emits a low-going pulse. <p><i>AIC transmit and receive sections are operated synchronously</i></p> <p>If the appropriate data bit in the control register is a 1, the transmit and receive sections are configured to be synchronous. In this case, the bandpass switched-capacitor filter and the A/D conversion timing are derived from the TX counter A, TX counter B, and TA, TA', and TB registers, rather than the RX counter A, RX counter B, and RA, RA', and RB registers. In this case, the AIC \overline{FSX} and \overline{FSR} timing are identical during primary data communication; however, \overline{FSR} is not asserted during secondary data communication since there is no new A/D conversion result. The synchronous operation sequences are as follows (see Serial Port Timing diagrams).</p> <ul style="list-style-type: none"> L Serial port directly interfaces with the serial port of the TMS32011 or TMS320C17 and communicates in two 8-bit bytes. The operation sequence is as follows (see Serial Port Timing diagrams): <ol style="list-style-type: none"> 1. \overline{FSX} and \overline{FSR} are brought low. 2. One 8-bit byte is transmitted and one 8-bit byte is received. 3. \overline{EODX} and \overline{EODR} are brought low 4. \overline{FSX} and \overline{FSR} emit positive frame-sync pulses that are four shift clock cycles wide 5. One 8-bit byte is transmitted and one 8-bit byte is received 6. \overline{EODX} and \overline{EODR} are brought high 7. \overline{FSX} and \overline{FSR} are brought high. H Serial port directly interfaces with the serial port of the TMS32020, TMS320C25, or TMS320C30 and communicates in one 16-bit word. The operation sequence is as follows (see Serial Port Timing diagrams): <ol style="list-style-type: none"> 1. \overline{FSX} and \overline{FSR} are brought low. 2. One 16-bit word is transmitted and one 16-bit word is received. 3. \overline{FSX} and \overline{FSR} are brought high. 4. \overline{EODX} or \overline{EODR} emit low-going pulses. <p>Since the transmit and receive sections of the AIC are now synchronous, the AIC serial port with additional NOR and AND gates will interface to two SN74299 serial-to-parallel shift registers. Interfacing the AIC to the SN74299 shift register allows the AIC to interface to an external FIFO RAM and facilitates parallel data bus communications between the AIC and the digital signal processor. The operation sequence is the same as the above sequence (see Serial Port Timing diagrams).</p>



FALTAN PAGINAS

De la: **6**

A la: **18**

absolute maximum ratings over operating free-air temperature (unless otherwise noted)†

Supply voltage range, V_{CC+} (see Note 1)	–0.3 V to 15 V
Supply voltage range, V_{DD}	–0.3 V to 15 V
Output voltage range, V_O	–0.3 V to 15 V
Input voltage range, V_I	–0.3 V to 15 V
Digital ground voltage range	–0.3 V to 15 V
Operating free-air temperature range, T_A : TLC32040C, TLC32041C	0°C to 70°C
TLC32040I, TLC32041I	–40°C to 85°C
Storage temperature range, T_{stg}	–40°C to 125°C
Case temperature for 10 seconds: FN package	260°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: N package	260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: Voltage values for maximum ratings are with respect to V_{CC-} .

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC+} (see Note 2)	4.75	5	5.25	V
Supply voltage, V_{CC-} (see Note 2)	–4.75	–5	–5.25	V
Digital supply voltage, V_{DD} (see Note 2)	4.75	5	5.25	V
Digital ground voltage with respect to ANALG GND, DGTL GND	0			V
Reference input voltage, $V_{ref(ext)}$ (see Note 2)	2	4		V
High-level input voltage, V_{IH}	2	$V_{DD} + 0.3$		V
Low-level input voltage, V_{IL} (see Note 3)	–0.3	0.8		V
Load resistance at OUT+ and/or OUT–, R_L	300			Ω
Load capacitance at OUT+ and/or OUT–, C_L	100			pF
MSTR CLK frequency (see Note 4)	0.075	5	10.368	MHz
Analog input amplifier common mode input voltage (see Note 5)	± 1.5			V
A/D or D/A conversion rate	20			kHz
Operating free-air temperature, T_A	TLC32040C, TLC32041C		0	70
	TLC32040I, TLC32041I		–40	85

NOTES: 2 Voltages at analog inputs and outputs, REF, V_{CC+} , and V_{CC-} , are with respect to ANALG GND. Voltages at digital inputs and outputs and V_{DD} are with respect to DGTL GND.

- The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels and temperature only.
- The bandpass low-pass switched-capacitor filter response specifications apply only when the switched-capacitor clock frequency is 288 kHz. For switched-capacitor filter clocks at frequencies other than 288 kHz, the filter response is shifted by the ratio of switched-capacitor filter clock frequency to 288 kHz.
- This range applies when (IN+ – IN–) or (AUX IN+ – AUX IN–) equals ± 6 V.

TLC32040, TLC32041, TLC32041C, TLC32041H
ANALOG INTERFACE CIRCUITS

SLAS014E - SEPTEMBER 1987 - REVISED MAY 1995

electrical characteristics over recommended operating free-air temperature range, $V_{CC+} = 5\text{ V}$, $V_{CC-} = -5\text{ V}$, $V_{DD} = 5\text{ V}$ (unless otherwise noted)

total device, MSTR CLK frequency = 5.184 MHz, outputs not loaded

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	$V_{DD} = 4.75\text{ V}$, $I_{OH} = -300\ \mu\text{A}$	2.4			V
V_{OL}	Low-level output voltage	$V_{DD} = 4.75\text{ V}$, $I_{OL} = 2\text{ mA}$			0.4	V
I_{CC+}	Supply current from V_{CC+}	TLC3204_C			35	mA
		TLC3204_I			40	
I_{CC-}	Supply current from V_{CC-}	TLC3204_C			-35	mA
		TLC3204_I			-40	
I_{DD}	Supply current from V_{DD}	(MSTR CLK = 5.184 MHz)			7	mA
V_{ref}	Internal reference output voltage		3		3.3	V
αV_{ref}	Temperature coefficient of internal reference voltage			200		ppm/°C
r_o	Output resistance at REF			100		k Ω

receive amplifier input

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
	A/D converter offset error (filters bypassed)			25	65	mV
	A/D converter offset error (filters in)			25	65	mV
CMRR	Common-mode rejection ratio at IN+, IN-, or AUX IN+, AUX IN-	See Note 6		55		dB
r_i	Input resistance at IN+, IN-, or AUX IN+, AUX IN-, REF			100		k Ω

transmit filter output

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OO}	Output offset voltage at OUT+, OUT-, (single-ended relative to ANALG GND)			15	75	mV
V_{OM}	Maximum peak output voltage swing across R_L at OUT+ or OUT-, (single ended)	$R_L \geq 300\ \Omega$, Offset voltage = 0	± 3			V
V_{OM}	Maximum peak output voltage swing between R_L at OUT+ and OUT-, (differential output)	$R_L \geq 600\ \Omega$	± 6			V

system distortion specifications, SCF clock frequency = 288 kHz

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Attenuation of second harmonic of A/D input signal	Single ended	$V_1 = -0.5\text{ dB}$ to -24 dB referred to V_{ref} . See Note 7		70		dB
	Differential		62	70		
Attenuation of third and higher harmonics of A/D input signal	Single ended	$V_1 = -0.5\text{ dB}$ to -24 dB referred to V_{ref} . See Note 7		65		dB
	Differential		57	65		
Attenuation of second harmonic of D/A input signal	Single ended	$V_1 = -0\text{ dB}$ to -24 dB referred to V_{ref} . See Note 7		70		dB
	Differential		62	70		
Attenuation of third and higher harmonics of D/A input signal	Single ended	$V_1 = -0\text{ dB}$ to -24 dB referred to V_{ref} . See Note 7		65		dB
	Differential		57	65		

† All typical values are at $T_A = 25^\circ\text{C}$.

NOTES: 6. The test condition is a 0-dBm, 1-kHz input signal with an 8-kHz conversion rate.

7. The test condition V_1 is a 1-kHz input signal with an 8-kHz conversion rate (0 dB relative to V_{ref}). The load impedance for the DAC is 600 Ω .



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

A/D channel signal-to-distortion ratio

PARAMETER	TEST CONDITIONS (see Note 7)	$A_V = 1^\dagger$		$A_V = 2^\dagger$		$A_V = 4^\dagger$		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
A/D channel signal-to-distortion ratio	$V_I = -6 \text{ dB to } -0.1 \text{ dB}$	58		>58§		>58§		dB
	$V_I = -12 \text{ dB to } -6 \text{ dB}$	58		58		>58§		
	$V_I = -18 \text{ dB to } -12 \text{ dB}$	56		58		58		
	$V_I = -24 \text{ dB to } -18 \text{ dB}$	50		56		58		
	$V_I = -30 \text{ dB to } -24 \text{ dB}$	44		50		56		
	$V_I = -36 \text{ dB to } -30 \text{ dB}$	38		44		50		
	$V_I = -42 \text{ dB to } -36 \text{ dB}$	32		38		44		
	$V_I = -48 \text{ dB to } -42 \text{ dB}$	26		32		38		
$V_I = -54 \text{ dB to } -48 \text{ dB}$	20		26		32			

D/A channel signal-to-distortion ratio

PARAMETER	TEST CONDITIONS (see Note 7)	MIN	MAX	UNIT
D/A channel signal-to-distortion ratio	$V_I = -6 \text{ dB to } 0 \text{ dB}$		58	dB
	$V_I = -12 \text{ dB to } -6 \text{ dB}$		58	
	$V_I = -18 \text{ dB to } -12 \text{ dB}$		56	
	$V_I = -24 \text{ dB to } -18 \text{ dB}$		50	
	$V_I = -30 \text{ dB to } -24 \text{ dB}$		44	
	$V_I = -36 \text{ dB to } -30 \text{ dB}$		38	
	$V_I = -42 \text{ dB to } -36 \text{ dB}$		32	
	$V_I = -48 \text{ dB to } -42 \text{ dB}$		26	
$V_I = -54 \text{ dB to } -48 \text{ dB}$		20		

gain and dynamic range

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Absolute transmit gain tracking error while transmitting into 600 Ω	-48-dB to 0-dB signal range, See Note 8		± 0.05	± 0.15	dB
Absolute receive gain tracking error	-48-dB to 0-dB signal range, See Note 8		± 0.05	± 0.15	dB
Absolute gain of the A/D channel	Signal input is a -0.5-dB, 1-kHz sine wave		0.2		dB
Absolute gain of the D/A channel	Signal input is a 0-dB, 1-kHz sine wave		-0.3		dB

power supply rejection and crosstalk attenuation

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{CC+} or V_{CC-} supply voltage rejection ratio, receive channel	$f = 0 \text{ to } 30 \text{ kHz}$		30		dB
	$f = 30 \text{ kHz to } 50 \text{ kHz}$		45		
V_{CC+} or V_{CC-} supply voltage rejection ratio, transmit channel (single ended)	$f = 0 \text{ to } 30 \text{ kHz}$		30		dB
	$f = 30 \text{ kHz to } 50 \text{ kHz}$		45		
Crosswalk attenuation, transmit-to-receive (single ended)			80		dB

† A_V is the programmable gain of the input amplifier.

‡ All typical values are at $T_A = 25^\circ\text{C}$.

§ A value > 58 is overrange and signal clipping occurs.

NOTES: 7. The test condition V_{IN} is a 1-kHz input signal with an 8-kHz conversion rate (0 dB relative to V_{REF}). The load impedance for the DAC is 600 Ω .

8. Gain tracking is relative to the absolute gain at 1 kHz and 0 dB (0 dB relative to V_{REF}).



**TEXAS
 INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265