UNIVERSIDAD MICHOACANA DE SAN NICOLAS DE HIDALGO

FACULTAD DE INGENIERÍA ELÉCTRICA

"APLICACIÓN DE UN FILTRO PASABANDA BASADO EN UN PROCESADOR DIGITAL DE SEÑALES"

TESIS

QUE PARA OBTENER EL TITULO DE:

INGENIERO ELECTRICISTA

PRESENTA:

GUILLERMO TAPIA TINOCO

ASESOR:

DR. NORBERTO GARCÍA BARRIGA

MORELIA, MICHOACAN. **Noviembre de 2005**

Esta tesis se la dedico a las personas que más quiero y admiro. Mis padres Guillermo Tapia y Teresa Tinoco y mis hermanos Librado, Juan y Jesús.

A DIOS por darme todo lo tengo.

A mi novia Paty por su cariño, compresión y apoyo durante nuestro noviazgo y en especial en esta etapa trascendental en mi vida.

A mi abuelita Margarita y a mis tíos Moy, Maria y Tita, así como a Arnoldo por su cariño y apoyo durante mis estudios de licenciatura.

A mis abuelitos Jesús e Imelda y a mi madrina Maria que con su cariño y apoyo me han alentado a seguir siempre adelante

A mi asesor el Dr. Norberto García Barriga por su gran paciencia y las bastantes horas invertidas en el desarrollo de esta tesis.

Al M.I. Isidro Lázaro Castillo por las facilidades otorgadas para la realización de este trabajo.

A mis sinodales los doctores Edmundo Barrera, Félix Calderón, Aurelio Medina y Jesús Rico por dedicar parte de su tiempo en el mejoramiento de esta tesis.

A mis amigos Gera, Pancho, Pedro, Rafa, Parra y Gary con los cuales conviví todos mis estudios de licenciatura.

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ACRONIMOS Y SIMBOLOS

ACCU	Acumulador del DSP
ADC	Convertidor Analógico a Digital
AIC	Circuito de Interfaz Analógica
ALU	Unidad Aritmética y Lógica
ARAU	Unidad Aritmética de Registro Auxiliar
CALU	Unidad Aritmética Lógica Central
CLKR	Reloj de recepción de datos
CLKX	Reloj de transmisión de datos
CPU	Unidad Central de Procesamiento
DAC	Convertidor Digital a Analógico
dB	Decibeles
dma	Localidad de memoria de datos
DRR	Registro de recepción de datos del puerto serie
DSK	*DSP Starter Kit*
DSP	*Digital Signal Processor*
DXR	Registro de transmisión de datos del puerto serie
F_H	Frecuencia de corte del filtro pasabajas correspondiente al filtro pasabanda
FIR	Respuesta Finita al Impulso
F_L	Frecuencia de corte del filtro pasaaltas correspondiente al filtro pasabanda
F_m	Frecuencia de muestreo en Hz
f_p	Frecuencia final de la banda de paso del filtro elíptico en (Hz)
f_s	Frecuencia de inicio de la banda de paro del filtro elíptico en (Hz)
FSR	Señal de inicio de recepción de datos
FSX	Señal de inicio de transmisión de datos
Hz	Hertz
IFR	Registro de banderas de interrupciones
IIR	Respuesta Infinita al Impulso
j	Operador imaginario de los números complejos

K_p	Magnitud máxima del rizo en la banda de paso del filtro elíptico en (dB)
K_S	Magnitud mínima del rizo en la banda de paro del filtro elíptico en (dB)
LSBs	Bits menos significativos
LSI	l*arge-scale integration*
MIPS	Millones de Instrucciones por Segundo
MSBs	Bits mas significativos
MSI	*médium-scale integration*
MSTR CLK	Frecuencia del reloj maestro
PLU	Unidad Lógica Paralela
PMST	Registro de status del DSP
PRD	Registro de periodo de la segunda etapa del *timer*
PSC	Registro de conteo regresivo de la primera etapa del *timer*
RINT	Interrupción de la recepción del puerto serie
s	Operador en el dominio de la frecuencia en tiempo continuo
SPC	Registro de control del puerto serie
ST0	Registro de status del DSP
ST1	Registro de status del DSP
TA	Registro de la unidad AIC
TB	Registro de la unidad AIC
TCR	Registro de control del *timer*
TDDR	Registro de periodo de la primera etapa del *timer*
TIM	Registro de conteo regresivo de la segunda etapa del *timer*
TINT	Interrupción del *timer*
$T_N(x)$	Polinomio de Chebyshev de orden N
TRNT	Interrupción de la recepción del puerto TDM
TXNT	Interrupción de la transmisión del puerto TDM
VLSI	*very-large-scale integration*
XINT	Interrupción de la transmisión del puerto serie
ω_c	Componente de frecuencia mas alta de una señal f(t)
ω_p	Frecuencia final de la banda de paso del filtro elíptico en (rad/seg)
ω_s	Frecuencia de inicio de la banda de paro del filtro elíptico en (rad/seg)

RESUMEN

En este trabajo se presenta la aplicación de los filtros pasabanda y pasabajas analógicos que se encuentra en el circuito de interfaz analógica (AIC) del *DSP Starter Kit* (DSK). El DSK es una herramienta que permite realizar aplicaciones en tiempo real. Esta herramienta cuenta con una tarjeta la cual tiene como componentes principales al DSP TMS320C50 de Texas Instrument y a la unidad AIC. Además se presenta la descripción del depurador y del ensamblador que conforman al DSK y se muestran algunas de las opciones más importantes en la utilización del depurador.

La configuración de la unidad AIC se realiza a través del DSP por lo que es necesario implementar un programa el cual permita realizar esta configuración y lleva por nombre CAP3. Esta configuración permite tener control completo sobre las características de la unidad AIC y proporciona flexibilidad para configurar esta unidad según lo requiere nuestra aplicación. En este trabajo se presentan cuatro casos de estudio los cuales tienen como propósito explotar las características de la unidad AIC, en especial la facilidad de configuración de la frecuencia de muestreo y las frecuencias de corte de los filtros analógicos. Esta característica de poder configurar fácilmente la frecuencia de muestreo y las frecuencias de corte es de gran importancia ya que se pueden adaptar estos parámetros a los requerimientos de cada aplicación sin la necesidad de realizar cambios físicos sino únicamente con la reconfiguración de la unidad AIC. Asimismo, se describe el proceso de muestreo de una señal, el filtrado de una señal y el fenómeno de traslape.

CAPITULO 1

INTRODUCCION

1.1 ANTECEDENTES

El procesamiento digital de señales es una área de la ciencia y la ingeniería que se ha desarrollado rápidamente durante los últimos 30 años. Este rápido desarrollo es el resultado de los avances tecnológicos tanto en las computadoras digitales como en la fabricación de circuitos integrados. Las computadoras digitales y el *hardware* asociado hace tres décadas eran relativamente grandes y caros y, como consecuencia, su uso se limitaba a aplicaciones de propósito en tiempo no real, tanto científicas como comerciales. El rápido desarrollo de la tecnología de circuitos integrados, empezando con la integración a media escala (MSI, *medium-scale integration*), la integración a gran escala (LSI, *large-scale integration*), y ahora, la integración al más alto grado (VLSI, *very-large-scale integration*) de circuitos electrónicos integrados ha estimulado el desarrollo de computadoras digitales y de *hardware* digital de propósito general más potente, pequeño, rápido y barato. Estos circuitos digitales baratos y relativamente rápidos han hecho posible construir sistemas digitales altamente sofisticados, capaces de realizar funciones y tareas del procesado de señales analógicas. De aquí que muchas de las tareas del procesado de señal que convencionalmente se realizaba analógicamente se realicen hoy mediante *hardware* digital, más barato y a menudo más fiable [Proakis y Manolakis 1998].

El procesamiento de señales es una tecnología que engloba un amplio conjunto de disciplinas entre las que se encuentran las telecomunicaciones, el control, la exploración del espacio y la medicina, por nombrar solo unas pocas. Hoy en día, esta afirmación es incluso más cierta con la televisión digital, los sistemas de información y el entretenimiento multimedia. Además, a medida que los sistemas de comunicación se van convirtiendo cada vez más en sistemas inalámbricos, móviles y multifunción, la importancia de un procesamiento de señales sofisticado en dichos equipos se hace cada vez más relevante [Internet 2005].

El procesamiento de señales estudia la representación, transformación y manipulación de señales y de la importancia que contienen. El termino de procesamiento digital de señales, se refiere a la representación mediante secuencias de números de precisión finita y se realiza utilizando una computadora digital.

A menudo es deseable que estos sistemas funcionen en tiempo real, lo que significa que el sistema en tiempo discreto se implementa de forma que las muestras de salida se calculan a la misma velocidad a la que se muestrea la señal en tiempo continuo.

Un desarrollo importante en la historia del procesamiento de señales ocurrió en el terreno de la microelectrónica. Aunque los primeros microprocesadores eran demasiado lentos para implementar en tiempo real la mayoría de los sistemas en tiempo discreto, a mediados de los ochenta la tecnología de los circuitos integrados había avanzado hasta el nivel de permitir la realización de microcomputadoras en punto fijo y punto flotante con arquitecturas especialmente diseñadas para realizar algoritmos de procesamiento de señales en tiempo discreto. A estos procesadores se les conoce por el acrónimo de DSP (*Digital Signal Processor).* Con esta tecnología llegó, por primera vez, la posibilidad de una amplia aplicación de técnicas de tratamiento de señales en tiempo discreto.

Como ha ocurrido con la mayoría de los progresos tecnológicos, la tecnología DSP no corresponde a un descubrimiento singular o un desarrollo aislado en su creación, sino que ha sido el resultado de avances continuos en la teoría del procesamiento de señales y en la tecnología que permite su aplicación práctica. El factor económico ha sido definitivo y esto se refleja en que muchas compañías han hecho contribuciones fundamentales a la teoría y la practica del DSP, como respuesta a sus problemas de ingeniería y sus intereses comerciales [Internet 2005].

Existe una discusión referente a la producción del primer circuito integrado DSP *single-chip* y en ella se destacan las compañías que aportaron soluciones cada vez más completas que finalmente llevaron a su realización. AMI produjo en 1978 el circuito integrado S2811, al que denominó *"Signal Processing Peripheral*", el cual era esencialmente un

procesador de uso general de operaciones matemáticamente complejas, cumpliendo una función de coprocesador matemático. Un año después Intel lanza el *"Analog Signal Processo*r 2920", que incluía sistemas de conversión digital-análogo y análogo-digital, pero carecía de multiplicador. Ya para 1980, NEC presenta el procesador DSP NECμPD7720, en la conferencia de Circuitos de Estado Sólido de IEEE en febrero de 1980, con todas las características propias de un DSP actual. En ese mismo evento Bell Labs presentó el DSP1, que a diferencia del anterior, sí ofrecía herramientas de desarrollo, documentación completa y ejemplos de aplicacione*s*. El epílogo de esta competencia por establecer un sistema DSP completo contenido en un circuito integrado, favoreció a Texas Instruments, quien fue el primer fabricante en diseñar y vender el primer circuito integrado DSP comercialmente exitoso, el TMS320C1x, consistente en un procesador de 16 bits, con unidad aritmética de punto fijo. En la actualidad los fabricantes más conocidos y que posiblemente ofrecen más soporte y documentación son Texas Instrument, Analog Devices, Motorola y Lucent Technologies.

Los DSPs hoy en día son usados para un amplio rango de aplicaciones de comunicación, control y procesamiento de imágenes. Las aplicaciones realizadas con DSPs se encuentran en productos comerciales como son teléfonos celulares, fax, módems, radios, impresoras, aparatos auditivos, reproductores de MP3, televisores de alta definición, cámaras digitales y muchos más [Chassaing 2005]. Además de realizarse aplicaciones comerciales también se pueden encontrar en la literatura aplicaciones a diferentes áreas, por ejemplo los sistemas de potencia, tal es el caso de [Miller y Dewe 1993], el cual describe la implementación de un filtro FIR en conjunción con una transformada rápida de Fourier (FFT) para el análisis armónico de un sistema de potencia. Otra aplicación basada en DSPs es la presentada en [Miller y Dewe 1992], la cual describe el uso de varios DSPs para la realización de un analizador multicanal continuo de armónicos en tiempo real. Debido a esto, es claro observar que los DSPs son el futuro en el desarrollo de nuevos dispositivos tanto en el terreno comercial como en las diferentes áreas de la ciencia ya que son ideales para el procesamiento en tiempo real y ofrecen una gran flexibilidad debido a su fácil reconfiguración.

1.2 OBJETIVOS

Los principales objetivos a cumplir en este trabajo son los siguientes:

- Configurar la unidad AIC (TLC32040) a través del DSP TMS320C50 de tal manera que se pueda controlar su funcionamiento de una manera adecuada.

- Observar el fenómeno de traslape en el DSP debido a la mala elección de la frecuencia de muestreo y la frecuencias de corte del filtro pasabanda analógico de la unidad AIC.

- Controlar las frecuencias de corte del filtro pasabanda analógico, para posteriormente realizar el filtrado de señales con una componente armónica.

- Realizar el muestreo de señales para posteriormente almacenar los valores resultantes del muestreo en la memoria de datos del DSP.

- Explotar a fondo las características de la unidad AIC de tal manera que permita sentar las bases para la realización de trabajos futuros basados en el DSP.

1.3 JUSTIFICACION

Existen muchas razones por las que el procesamiento digital de una señal analógica puede ser preferible al procesamiento de la señal directamente en el dominio analógico. Primero, un sistema digital programable permite flexibilidad a la hora de reconfigurar las operaciones de procesamiento digital de señales sin más modificaciones que cambiar el programa. La reconfiguración de un sistema analógico implica habitualmente el rediseño del *hardware*, seguido de la comprobación y verificación de su correcto funcionamiento una vez que este implementado [Proakis y Manolakis 1998].

Las tolerancias en los componentes de los circuitos analógicos hacen que para el diseñador del sistema sea extremadamente difícil controlar la precisión de un sistema de procesamiento analógico de señales. En cambio, un sistema digital permite un mejor control

en los requisitos de precisión. Tales requisitos, a su vez, resultan en la especificación de requisitos de precisión del convertidor A/D y del procesador digital de señales, en términos de longitud de palabra, aritmética de punto fijo frente a aritmética de punto flotante y factores similares.

Las señales digitales se almacenan fácilmente en soporte magnético (cinta o discos) sin deterioro o pérdida en la fidelidad de la señal, aparte de la introducida en la conversión A/D. Como consecuencia, las señales se hacen fácilmente transportables y pueden procesarse en tiempo no real en un laboratorio remoto.

Sin embargo, la implementación digital tiene sus limitaciones. Una limitación práctica es la velocidad de operación de los convertidores A/D y de los procesadores digitales de señales. En el caso que las señales contengan anchos de banda extremadamente grandes necesitan convertidores A/D con una velocidad de muestreo alta y procesadores digitales de señales rápidos. De esta manera, habrá casos en el que la solución mediante el procesado digital de señales se encuentra más allá de las características del *hardware* digital.

Una consideración importante en el procesamiento digital de señales es el muestreo correcto de la señal. Cuando se muestrea correctamente una señal ésta se puede recuperar correctamente; en cambio cuando no se realiza el muestreo correctamente se presenta el fenómeno de traslape y la señal obtenida no se reproducirá correctamente. Por lo tanto, en este trabajo se presenta un caso de estudio destinado a reproducir este fenómeno.

Como se dijo anteriormente el procesamiento digital de señales ofrece la ventaja de almacenamiento en forma eficiente y sin perder exactitud para posteriormente procesarla en tiempo no real. Por lo tanto, en este trabajo se presenta un caso de estudio en el cual se realiza el muestreo y almacenamiento de una señal.

En el caso de la tarjeta *DSP Starter Kit* la unidad AIC juega un papel importantísimo en cualquier aplicación basada en DSP's ya que es la encargada de comunicar el mundo real con el dispositivo digital, y es de suma importancia el poder controlar su funcionamiento. La

programación correcta de esta unidad permite configurarla de la manera más adecuada a fin de obtener los mejores resultados y satisfacer cada aplicación en específico. Por otro parte, este trabajo sentará las bases para llevar a cabo más trabajos de investigación que requieren la implementación en el laboratorio de herramientas de procesamiento digital de señales.

1.4 METODOLOGIA

En la Figura 1.1 se muestra la configuración del sistema implementado para llevar a cabo los experimentos en el laboratorio relacionados con este trabajo. Los cuatro casos de estudio implementados en el laboratorio tienen como objetivo explotar la capacidad de los filtros analógicos de la unidad AIC para ser utilizados en diferentes aplicaciones.

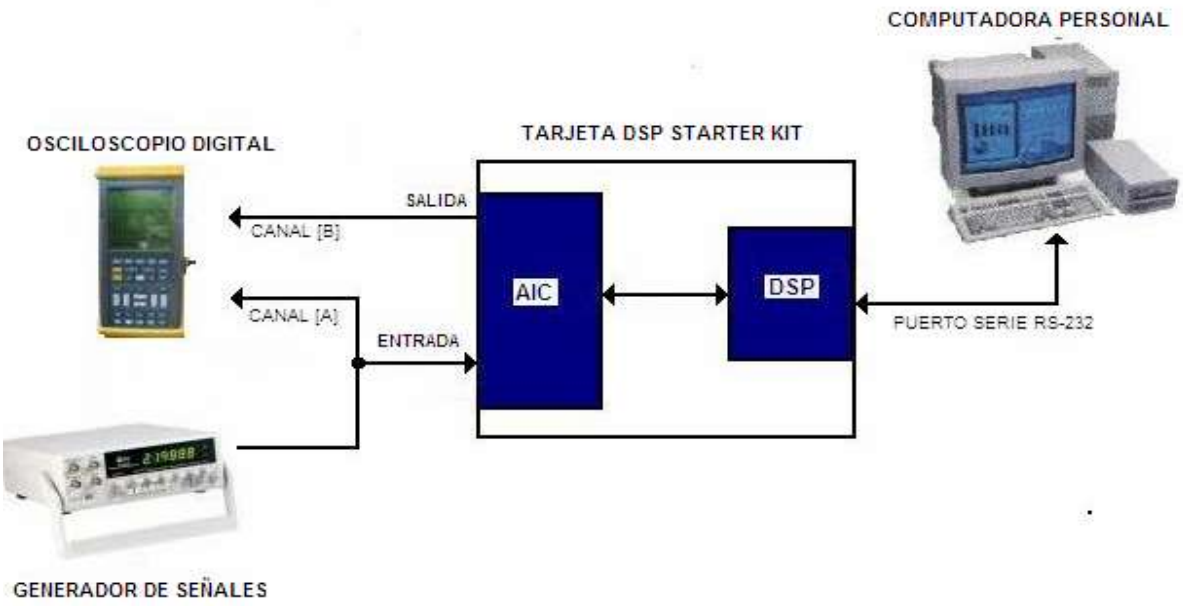


Figura 1.1Elementos utilizados en los experimentos de laboratorio

Para poder llevar a cabo cualquiera de los cuatro casos de estudio es necesario contar con tres elementos principales: El primero de ellos consiste en la tarjeta *DSP Starter Kit* y una computadora personal en la cual se instala el *software* con el que viene acompañada la tarjeta. El segundo elemento consiste del programa realizado en código fuente, el cual es el encargado de llevar a cabo la aplicación específica. El tercer elemento consiste en los dispositivos

externos encargados de suministrar y medir las señales que serán usadas en el procesamiento y, que en el caso de este trabajo se utilizaron generadores de señales y un osciloscopio digital. El osciloscopio digital utilizado en este trabajo es el Fluke 123, el cual permite medir dos señales simultáneamente proporcionando información sobre la magnitud, frecuencia, forma de onda de las señales, etc.

En este trabajo se llevaron a cabo cuatro casos de estudio. En los primeros tres casos se utilizó el mismo programa CAP3, el cual tiene como objetivo configurar tanto la unidad AIC como el DSP. Además de llevar a cabo la configuración, el programa se encarga de muestrear una señal de entrada la cual es suministrada a la unidad AIC. Una vez que se obtiene cada muestra, ésta sufre un proceso de conversión analógico a digital y es enviada al DSP a través del puerto serie. Al llegar el dato al DSP éste se puede procesar y en su caso se envía de regreso a la unidad AIC. El dato que llegó a la unidad AIC sufre una conversión digital a analógico y es llevado a la salida de la unidad AIC.

Para llevar a cabo el caso de estudio cuatro, fue necesario realizar un segundo programa CAP4. Este programa tiene un funcionamiento similar al programa anterior con la diferencia que ahora el dato que llega al DSP no es enviado de regreso a la unidad AIC. En este caso el dato se almacena dentro de la memoria interna del DSP y una vez que se han tomado N muestras se deja de muestrear la señal de entrada. Cabe mencionar que fue necesario la realización de dos programas en Matlab MAT1 y MAT2 los cuales se utilizaron para calcular los valores de configuración de la unidad AIC.

1.5 DESCRIPCION DE CAPITULOS.

El presente trabajo se estructura en seis capítulos los cuales se describen a continuación.

En el Capítulo 1 se presenta una breve descripción de la evolución que ha sufrido el procesamiento digital de señales y se presentan los objetivos marcados en este trabajo así como la metodología utilizada para llevarlos a cabo. Además, se describen de manera general los componentes necesarios para llevar a cabo una aplicación con la tarjeta *DSP Starter Kit.*

En el Capítulo 2 se presenta una clasificación de los filtros analógicos más conocidos y su respuesta a la frecuencia (Butterwort, Chebyshev y elíptico). También se presentan una descripción de los filtros digitales, el proceso de muestreo de una señal y la conversión digital-analógico.

En el Capítulo 3 se presenta la descripción detallada del DSP TMS320C50, la cual abarca su arquitectura, organización de la memoria, los dispositivos periféricos y las instrucciones utilizadas en este trabajo. También se presenta la descripción del *DSP Starter Kit* y se presenta el *software* y h*ardware* con el que viene acompañado para poder realizar aplicaciones en tiempo real.

En el Capítulo 4 se presenta la descripción de la unidad AIC, la cual abarca desde aspectos físicos hasta su funcionamiento interno, registros de control y la manera de realizar la configuración. Además, se proporcionan las ecuaciones necesarias para calcular la frecuencia de muestreo y las frecuencias de corte de los filtros analógicos de la unidad AIC a partir de los parámetros de configuración. Por último, se muestran los pasos necesarios para configurar la tarjeta *DSP Starter Kit* y realizar una aplicación determinada.

En el Capítulo 5 se presenta el resultado de las aplicaciones realizadas con el *DSP Starter Kit.* Estas aplicaciones son: *i*).- La obtención de la respuesta a la frecuencia del filtro pasabajas de la unidad AIC, *ii*).- El fenómeno de traslape debido a la mala elección de la frecuencia de muestreo y las frecuencias de corte del filtro pasabanda de la unidad AIC, *iii*).- El filtrado de señales con una componente armónica y, por último, *iv*).- El muestreo y almacenamiento en la memoria de datos de una señal entrada.

En el Capítulo 6 se presentan las conclusiones generales y se comentan los resultados obtenidos en este trabajo.

CAPITULO 2

FILTROS DIGITALES

En este capítulo se presentan las características más importantes de los filtros analógicos Butterworth, Chebyshev y elípticos siendo estos últimos la clase de filtros con los que cuenta la unidad AIC que serán utilizados a lo largo de esta tesis. También se muestra una descripción breve de los filtros digitales y su relación con los filtros analógicos. Además se presentan algunos conceptos básicos relacionados con el muestreo de una señal.

2.1 INTRODUCCION.

Un filtro es una red utilizada para separar señales en base a su frecuencia. Un filtro es útil cuando la señal de interés tiene un contenido de frecuencias que es diferente al contenido de frecuencias de las señales no deseadas, ruido por ejemplo. Existen diferentes maneras de clasificar a los filtros, una de ellas es por los dispositivos utilizados para su implementación y se clasifican en pasivos, activos y digitales. Los filtros pasivos son circuitos implementados en base a resistencias, condensadores e inductancias. Son especialmente adecuados para altas frecuencias. Para operación en bajas frecuencias resultan caros y físicamente grandes y voluminosos. Los filtros activos son circuitos que incluyen resistencias, condensadores y amplificadores operacionales. Se pueden lograr características de alta impedancia de entrada y baja impedancia de salida, por lo cual se pueden conectar fácilmente en cascada. Su respuesta a la frecuencia esta limitada a bajas y medianas frecuencias. Además requieren una fuente de poder para operar. Los filtros digitales se implementan en base a convertidores A/D, circuitos digitales y convertidores D/A. Estos filtros son adecuados para la implementación de filtros de alto orden. La sintonización se puede realizar mediante la reprogramación de los coeficientes del algoritmo matemático. La respuesta a la frecuencia de estos filtros está determina por la velocidad de procesamiento del sistema digital utilizado [Chen 1987].

Respecto a los filtros analógicos, los más comúnmente usados son el Butterworth, Chebyshev y Elíptico. El filtro Butterworth se caracteriza por tener una respuesta plana tanto en la banda de paso como en la banda de paro mientras que el filtro Chebyshev tiene rizado en

la banda de paso y una respuesta plana en la banda de paro o viceversa. El filtro elíptico tiene rizado en ambas bandas. De estos tres tipos de filtros, el filtro Butterworth tiene un diseño más simple y es el más utilizado.

Por su parte los filtros digitales comúnmente usados son los FIR e IIR, el filtro FIR tiene una respuesta al impulso finita o no recursiva, su función de transferencia se caracteriza por la ausencia de polos. El filtro IIR se caracteriza por tener una respuesta al impulso infinita o recursiva, su función de transferencia cuenta con la presencia de polos y ceros. Para el diseño de filtros IIR existen diferentes métodos pero el más utilizado es el de la transformación bilineal, este método consiste en trasladar las características de un filtro digital al dominio analógico y una vez que se obtiene la función de transferencia del filtro analógico se realiza una transformación al dominio digital y así obtener la función de transferencia del filtro digital [Proakis y Manolakis 1998].

2.2 CARACTERISTICAS DE LOS FILTROS ANALOGICOS UTILIZADOS HABITUALMENTE.

El diseño de filtros analógicos es un campo bien desarrollado y se han escrito muchos libros sobre la materia. En esta sección se describirá brevemente las características más importantes de los filtros analógicos pasabajas mencionados anteriormente y se introducirán los parámetros más relevantes del filtro [Proakis y Manolakis 1998].

2.2.1 FILTROS BUTTERWORTH

Los filtros Butterworth pasabajas son filtros que contienen únicamente polos caracterizados por la magnitud al cuadrado de la respuesta a la frecuencia, la cual es

$$\left|H(\Omega)\right|^2 = \frac{1}{1+\left(\Omega/\Omega_c\right)^{2N}} = \frac{1}{1+\varepsilon^2\left(\Omega/\Omega_p\right)^{2N}} \tag{2.1}$$

donde N es el orden del filtro, Ω_c es su frecuencia a -3 dB (normalmente denominada frecuencia de corte), Ω_p es la frecuencia de corte de la banda de paso y $1/(1+\varepsilon^2)$ es el valor en el limite de la banda de $|H(\Omega)|^2$. La característica de respuesta a la frecuencia de los filtros Butterworth se muestran en la Figura 2.1 para varios valores de N. Se puede observar en la figura que $|H(\Omega)|^2$ es monótona tanto en la banda de paso como en la banda de paro.

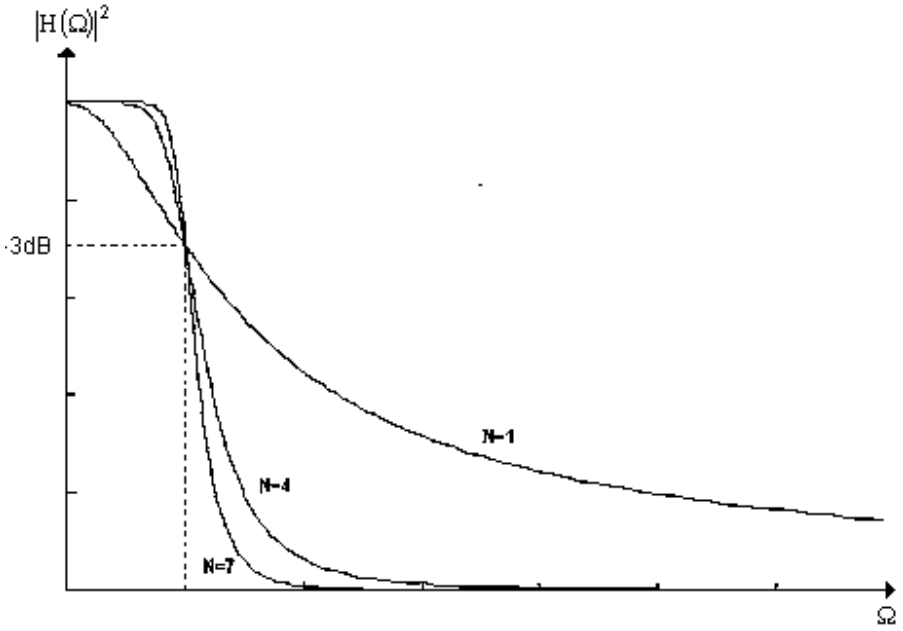


Figura 2.1 Magnitud de la respuesta a la frecuencia de los filtros Butterworth.

2.2.2 FILTROS CHEBYSHEV

Existen dos tipos de filtros Chebyshev. Los filtros de Chebyshev de tipo I son filtros que contienen únicamente polos que exhiben un comportamiento de rizado constante en la banda de paso y una característica monótona en la banda de paro. Por otro lado, la familia de filtros Chebyshev de tipo II contiene polos y ceros y exhiben un comportamiento monótono en la banda de paso y un comportamiento de rizado constante en la banda de paro. Los ceros en esta clase de filtro yacen en el eje imaginario del plano **s.** El cuadrado de la magnitud de la característica de respuesta a la frecuencia de un filtro Chebyshev de tipo I viene dado por.

$$|H(\Omega)|^2 = \frac{1}{1+\varepsilon^2 T_N^2(\Omega/\Omega_p)} \tag{2.2}$$

donde ε es un parámetro del filtro relacionado con el rizado en la banda de paso y $T_N(x)$ es el polinomio de Chebyshev de N-esimo orden, definido como

$$T_N(x) = \begin{cases} \cos(N \cos^{-1} x) & |x| \le 1 \\ \cosh(N \cosh^{-1} x) & |x| > 1 \end{cases} \qquad (2.3)$$

Los polinomios de Chebyshev se pueden generar mediante la ecuación recursiva

$$T_{N+1}(x) = 2x T_N(x) - T_{N-1}(x) \qquad N = 1,2\ldots\ldots \qquad (2.4)$$

donde $T_0(x) = 1$ y $T_1(x) = x$. De la ecuación 2.4 obtenemos $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x$ y así sucesivamente. Algunas de las propiedades de estos polinomios son:

1. $|T_N(x)| \le 1$ para todo $|x| \le 1$.
2. $T_N(1) = 1$ para todo N.
3. Todas las raíces del polinomio $T_N(x)$ ocurren en el intervalo $-1 \le x \le 1$.

El parámetro del filtro ε se relaciona con el rizado en la banda de paso, como se ilustra en la Figura 2.2, para N impar y para N par. Para N impar, $T_N(0) = 0$ y por lo tanto, $|H(0)|^2 = 1$. Por otro lado, para N par, $T_N(0) = 1$ y, por lo tanto, $|H(0)|^2 = 1 / (1+\varepsilon^2)$.

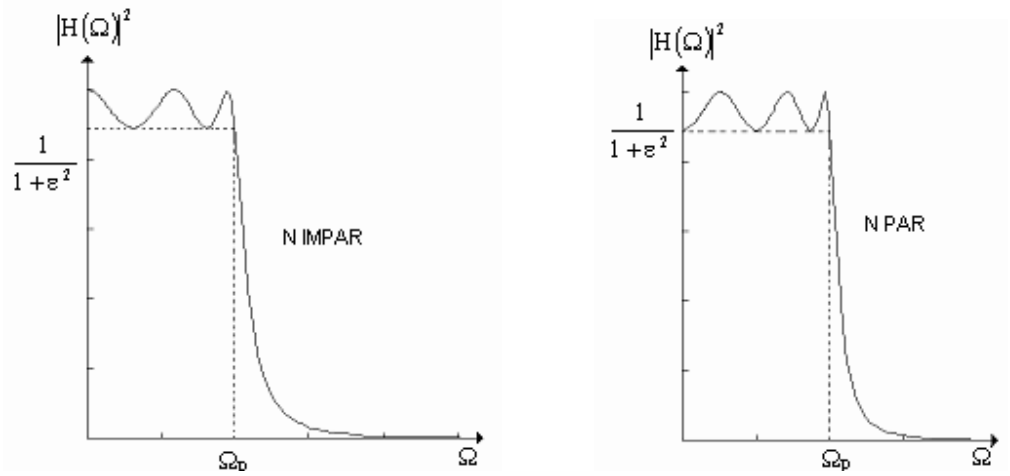


Figura 2.2 Magnitud de la respuesta a la frecuencia de los filtros Chebyshev tipo I.

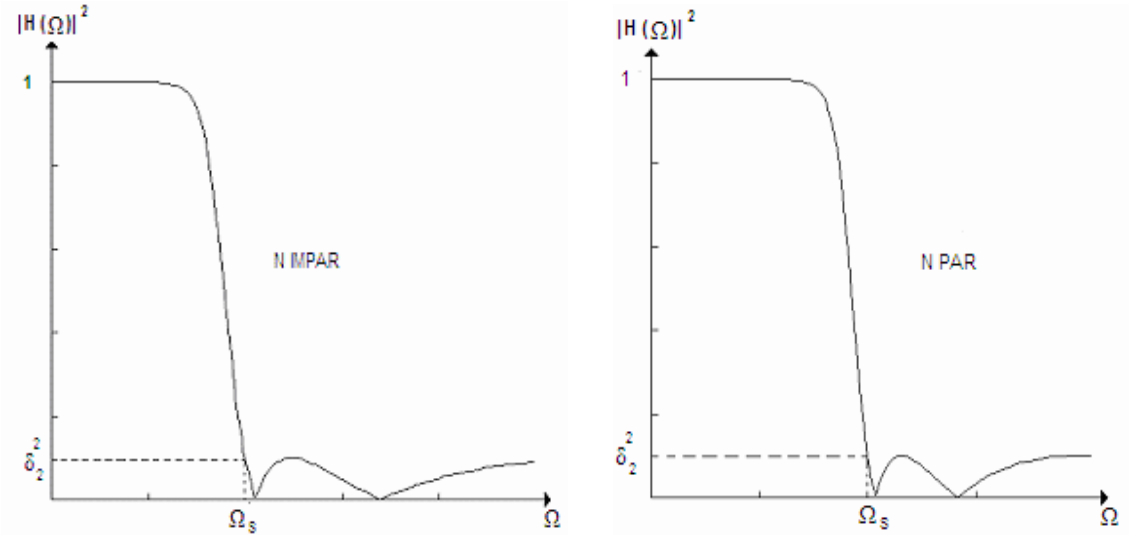


Figura 2.3 Magnitud de la respuesta a la frecuencia de los filtros Chebyshev tipo II.

Un filtro Chebyshev tipo II contiene tanto ceros como polos. El cuadrado de la magnitud de su respuesta a la frecuencia esta dado por

$$\left|H(\Omega)\right|^2 = \frac{1}{1+\varepsilon^2\left[T_N^2(\Omega_s/\Omega_p)/T_N^2(\Omega_s/\Omega)\right]} \tag{2.5}$$

donde $T_N(x)$ es el polinomio de Chebyshev de N-esimo orden y Ω_s es la frecuencia de la banda de paro, como se ilustra en la Figura 2.3.

2.2.3 FILTROS ELIPTICOS

Los filtros elípticos o Cauer presentan un comportamiento de rizado constante tanto en la banda de paso como en la banda de paro, como se ilustra en la Figura 2.4 para N par y N impar. Esta clase de filtros contienen polos y ceros y se caracterizan por la magnitud al cuadrado de la respuesta a la frecuencia dada por,

$$\left|H(\Omega)\right|^2 = \frac{1}{1+\varepsilon^2 U_N(\Omega/\Omega_p)} \tag{2.6}$$

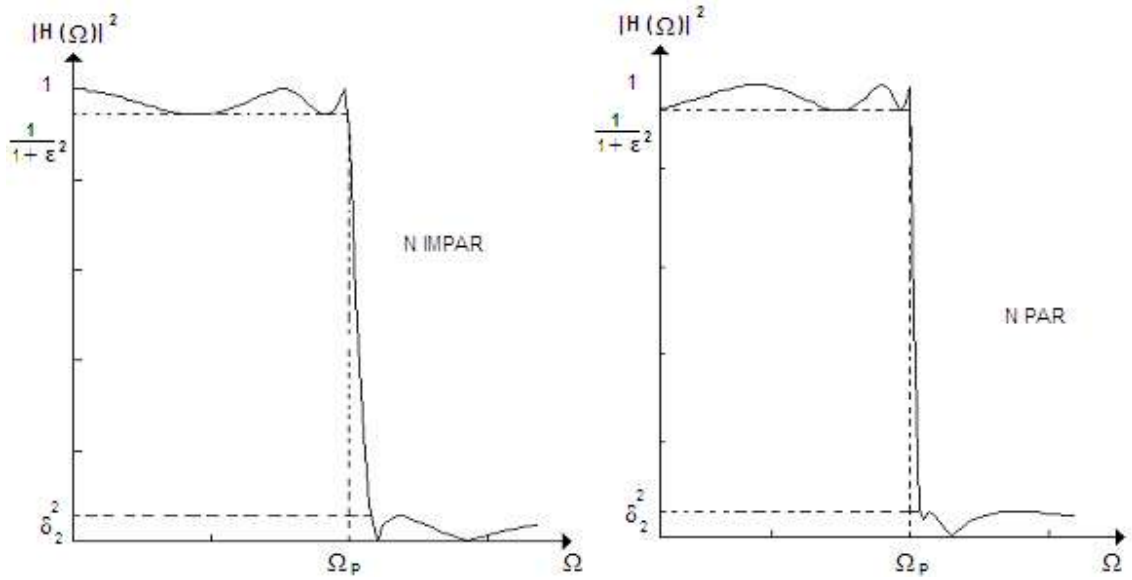


Figura 2.4 Magnitud de la respuesta a la frecuencia de los filtros elípticos.

donde $U_N(x)$ es la función elíptica jacobina de orden N y ε es un parámetro relacionado con el rizado en la banda de paso. Para este filtro los ceros yacen en el eje jΩ.

No se pretende describir en esta sección las funciones elípticas en detalle por que tal discusión va más allá del objetivo de esta tesis. Es suficiente decir que existen programas de computadora como Matlab para diseñar filtros Butterworth, Chebyshev o Elípticos a partir de sus especificaciones en frecuencia.

2.3 MUESTREO DE UNA SEÑAL Y FENOMENO DE TRASLAPE

El proceso de muestreo de una señal requiere el uso de la frecuencia adecuada, de lo contrario se podría presentar el fenómeno de confusión de frecuencia ó traslape (*frequency aliasing*). Por ejemplo, considérese una señal senoidal de frecuencia f, la cual se muestrea seis veces por ciclo ($f_m = 6\ f$), como lo muestra la Figura 2.5.

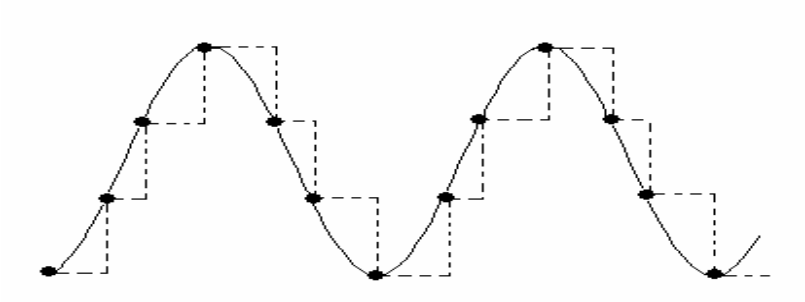


Figura 2.5 Señal senoidal muestreada seis veces por ciclo.

En este caso se tiene que la Frecuencia observada = Frecuencia verdadera y por lo tanto $f_o = f$. Ahora bien, si la señal se muestrea tres veces por ciclo entonces $f_m = 3\ f$ y se tiene nuevamente el caso en que $f_o = f$. Este caso se puede ver en la Figura 2.6

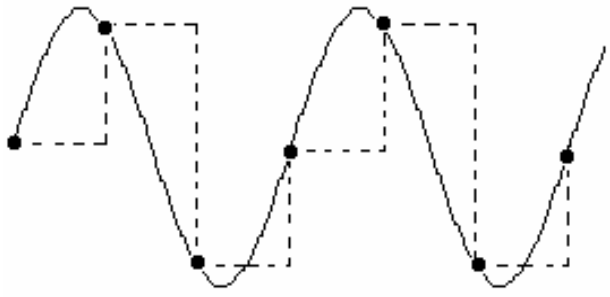


Figura 2.6 Señal senoidal muestreada tres veces por ciclo.

Por otra parte, si la señal se muestrea seis veces en cinco ciclos, entonces se puede ver que en este caso $f_m = \frac{6f}{5}$. En este caso, al utilizar esta frecuencia de muestreo se detecta una frecuencia aparente $f_o < f$ como se muestra en la Figura 2.7.

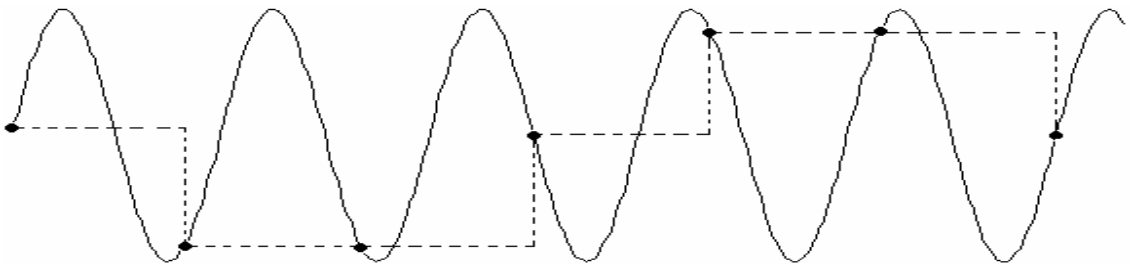


Figura 2.7 Señal senoidal muestreada 1.2 veces por ciclo.

La frecuencia de muestreo f_m necesaria para muestrear una señal, sin que ocurra el fenómeno de confusión de frecuencias se puede determinar mediante el teorema de muestreo de Nyquist también conocido como el criterio de Shannon, el cual se presenta a continuación:

Teorema de muestreo: Si ω_m, definida como $2\pi/T$, donde T es el periodo de muestreo, es mayor que $2\omega_c$, o bien

$$\omega_m \geq 2\omega_c \tag{2.7}$$

en donde ω_c es la componente de más alta frecuencia presente en la señal en tiempo continuo, entonces la señal original se puede construir completamente a partir de la señal muestreada [Ogata 1996].

Consideremos una señal f(t) con un espectro de frecuencia F(ω) como el de la Figura 2.8, cuya componente de frecuencia más alta esta dada por ω_c. Si esta señal se muestrea con una frecuencia de muestreo ω_m, el proceso de muestreo produce un espectro de frecuencias centrada alrededor de los múltiplos de la frecuencia de muestreo, como se ilustra en la Figura 2.9.

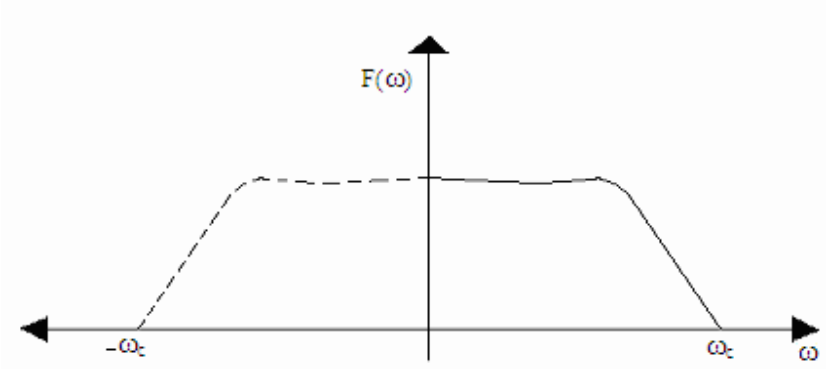


Figura 2.8 Espectro de frecuencias de una señal f(t) cuya componente de frecuencia más alta es ω_c.

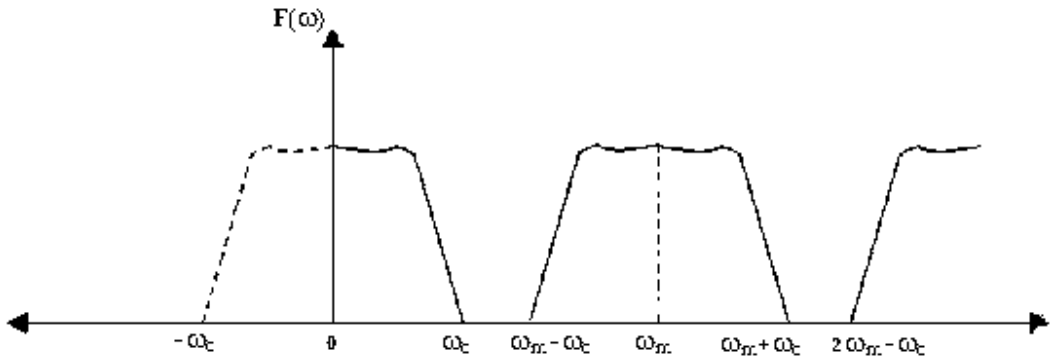


Figura 2.9 Espectro de frecuencias producido por el muestreo de una señal f(t).

Se puede observar que para la situación mostrada en la figura anterior, es decir, cuando $\omega_m \geq 2\omega_c$ no ocurre traslape en el espectro de frecuencias resultante del muestreo. En este caso, la señal original f(t) se puede recobrar utilizando un filtro pasabajas con una frecuencia de corte ω_n tal que:

$$\omega_c < \omega_n < (\omega_m - \omega_c) \tag{2.8}$$

Si $\omega_m = 2\omega_c$ se tendrá la situación mostrada en la Figura 2.10.

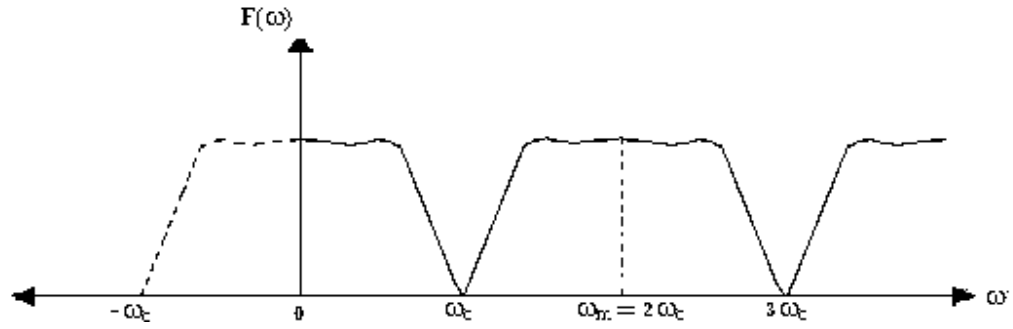


Figura 2.10 Espectro de frecuencias producido al tener una frecuencia de muestreo $\omega_m = 2\omega_c$

Sin embargo, si $\omega_m < 2\omega_c$ ocurre un traslape en el espectro de frecuencia resultante del muestreo, lo cual causa errores de confusión de frecuencias por lo cual la señal original no se puede recuperar. Esta situación se ilustra en la Figura 2.11.

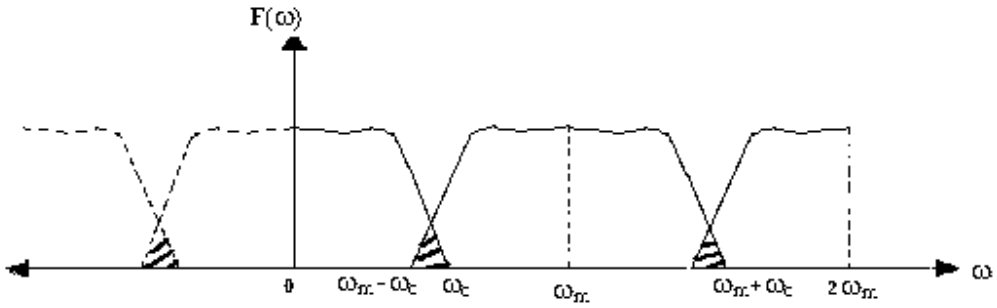


Figura 2.11 Espectro de frecuencias producido al tener una frecuencia de muestreo $\omega_m < 2\omega_c$.

Se puede observar que el teorema de muestreo de Nyquist define la velocidad de muestreo mínima para evitar traslapes en el espectro de frecuencias. En la práctica se escoge una frecuencia de muestreo mayor que $2\omega_c$, ya que en caso contrario se requerirá un filtro de muy alto orden.

Para garantizar que no ocurran errores de traslape de frecuencias se debe incluir un filtro pasabajas que remueva las frecuencias arriba de $\frac{f_m}{2}$ ó $\frac{\omega_m}{2}$ dependiendo de las unidades en que se decida realizar los cálculos. Dicha etapa de filtrado analógico se implementa antes de muestrear la señal, como se ilustra en la Figura 2.12.

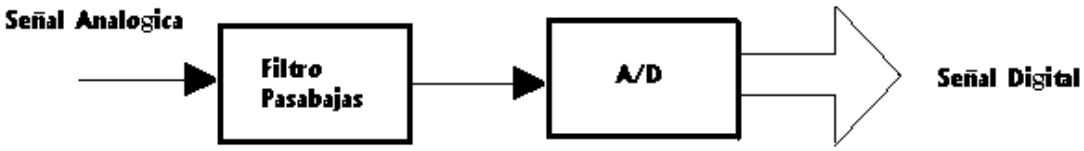


Figura 2.12 Diagrama de bloques que garantiza el muestreo correcto de una señal f(t).

En el caso de la unidad AIC del DSK no se cuenta con un filtro pasabajas pero en cambio se tiene un filtro pasabanda el cual puede realizar la misma función. Este filtro eliminará las frecuencias menores a F_L y mayores a F_H por lo tanto todas las frecuencias dentro del rango comprendido entre F_H y F_L no serán atenuadas. Lo anterior será cierto siempre y cuando se cumpla con el criterio de Shannon ya que de no cumplirse ocurrirá el fenómeno traslape.

2.4 CONVERSION DIGITAL A ANALOGICA

Para convertir una señal digital en analógica se usa un convertidor digital a analógico (D/A). El cometido de un convertidor digital a analógico es interpolar entre las muestras. El teorema del muestreo especifica la interpolación óptima para una señal de banda limitada. Sin embargo, este tipo de interpolación es demasiado complicado y, por ello, impráctico. Desde un punto de vista pragmático, el convertidor D/A más simple es el retenedor de orden cero que se muestra en la Figura 2.13, y que simplemente mantiene constante el valor de una muestra hasta que se recibe la siguiente. Se pueden tener mejoras adicionales utilizando interpolación lineal, tal como se muestra en la Figura 2.14, para conectar las muestras sucesivas con segmentos de línea recta. Utilizando técnicas más sofisticadas de interpolación de órdenes superiores se puede alcanzar una mejor interpolación [Proakis y Manolakis 1998].

En general, al utilizar técnicas de interpolación subóptimas aparecen componentes de frecuencia indeseables que deben ser eliminadas pasando la salida del interpolador a través de un filtro analógico adecuado, que recibe el nombre de postfiltro o filtro suavizante. De este modo, en la conversión D/A normalmente interviene un interpolador subóptimo seguido de un postfiltro.

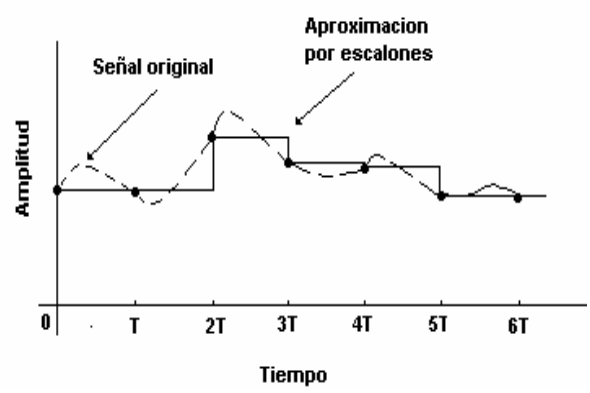


Figura 2.13 Conversión analógica a digital con retenedor de orden cero.

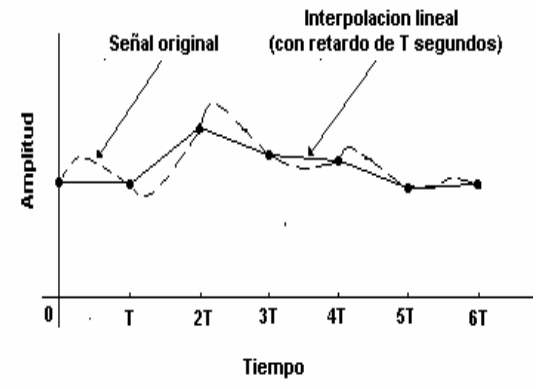


Figura 2.14 Conector lineal de puntos (con retardo de T segundos).

CAPITULO 3

PROCESADOR DIGITAL DE SEÑALES

En este capítulo se presenta la descripción del DSP TMS320C50 el cual es el encargado de llevar a cabo la programación, configuración y control de la unidad AIC. Por lo tanto es de suma importancia describir a detalle las partes del DSP que se utilizarán en el desarrollo de este trabajo entre las que se encuentran la organización de la memoria, los modos de direccionar la memoria, los dispositivos periféricos y las instrucciones utilizadas en este trabajo. En este mismo capítulo se presenta la descripción de la tarjeta DSK y se muestran los componentes que la conforman entre los cuales se encuentra el DSP. Además se muestra el *software* con el que viene acompañada dicha tarjeta y la forma de utilizarlo. Cabe mencionar que este *software* se encuentra disponible en un *diskette* y es proporcionado por el fabricante.

3.1 INTRODUCCION

La familia de procesadores digitales de señales TMS320 de Texas Instrument está diseñada para soportar un amplio rango de aplicaciones de procesamiento digital de señales. Hoy en día la familia TMS320 consiste de 5 generaciones, la familia C1x, C2x, C3x, C4x y C5x. Es importante hacer notar que las generaciones C1x, C2x, C5x son de punto fijo y la C3x y C4x son de punto flotante. El código fuente es compatible entre las generaciones de punto fijo y lo mismo sucede entre las dos generaciones de punto flotante. Esto es una ventaja económica, pues la inversión hecha en *software* se preserva [Texas Instrumen 1993].

Cada generación de dispositivos TMS320 tiene diferentes configuraciones de memoria y periféricos, lo que los hace ideales para satisfacer las distintas necesidades del mercado. Cuando la memoria y los periféricos se incluyen en el mismo circuito integrado se reduce el costo, así como el tamaño de los circuitos. El TMS320C50 está fabricado con la tecnología de circuito CMOS. La arquitectura en la cual se basa el C50 es la arquitectura Harvard (buses separados para memoria de datos y memoria de programa) con lo que se logra la mayor velocidad. El TMS320C50 está diseñado para ejecutar más de 28 Millones de instrucciones por segundo (MIPS).

3.2 CARACTERISTICAS DEL DSP TMS320C50.

Las características principales del DSP TMS320C50 se listan a continuación:

- Ciclo de instrucción 35/50 ns.
- Operación de memoria basada en RAM.
- 9K x 16 bits de SARAM (*single access* RAM).
- 2K x 16 bits de ROM
- 1056 x 16 bits de DRAM (*dual access* RAM)
- 224K x 16 bits de espacio de memoria direccionable.
- Unidad Aritmética lógica (ALU) de 32 bits.
- Acumulador (ACC) de 32 bits
- Acumulador de buffer (ACCB) de 32 bits.
- Unidad lógica paralela (PLU) de 32 bits.
- Multiplicador paralelo de 16 x 16 bits con capacidad de producto de 32 bits.
- Instrucción de multiplicación y acumulación en un solo ciclo.
- Ocho registros auxiliares con una unidad aritmética dedicada para direccionamiento indirecto.
- Pila de ocho niveles.
- Puerto serial *full-duplex* sincrono para comunicación directa entre el C50 y otro dispositivo serial.
- Puerto serial de acceso múltiple por división de tiempo (TDM).
- Puertos paralelos de E/S de 64K, de los cuales 16 se encuentran en los registros internos.
- Generador de reloj interno.
- Tecnología CMOS con dos modos de apagado.

3.3 ARQUITECTURA DEL DSP

La arquitectura del TMS320C50 consiste de tres elementos básicos:

- Unidad Central de Procesamiento (CPU)
- Memoria.
- Circuitos periféricos.

Estos elementos se detallan a continuación.

3.3.1 UNIDAD CENTRAL DE PROCESAMIENTO.

El C50 trabaja con una aritmética de complemento a dos usando la unidad ALU de 32 bits y el acumulador. La unidad ALU es una unidad aritmética de propósito general que usa palabras de 16 bits tomadas de memoria de datos, derivadas de instrucciones inmediatas o que son el resultado del multiplicador, además, es capaz de realizar operaciones boleanas. El acumulador almacena la salida de la ALU y también es la segunda entrada a la ALU. El acumulador tiene una longitud de 32 bits y se divide en dos partes, los bits MSBs (del bit 31 al 16) y los bits LSBs (del 15 al 0). Además de la ALU, el procesador cuenta con la Unidad PLU que efectúa operaciones lógicas de datos sin afectar el contenido del acumulador.

El multiplicador realiza la multiplicación en complemento a dos de 16 x 16 bits con un resultado de 32 bits en un ciclo de instrucción. Dicho multiplicador está compuesto por lo siguientes elementos: Arreglo multiplicador, registro de producto (PREG) y registro temporal (TREG0).

Los valores del multiplicador son tomados de la memoria de datos o de programa dependiendo de la instrucción. El registro de corrimientos tiene una entrada de 16 bits conectada al bus de datos y una salida de 32 bits conectada a la ALU. Con este registro se producen corrimientos hacia la izquierda de 0 a 16 bits sobre el dato de entrada según se defina en la instrucción o en TREG1 (registro de 5 bits).

El procesador cuenta con el bus de datos y con el bus de programa. El primero interconecta varios elementos de la Unidad aritmética lógica Central (CALU) y los registros auxiliares a la memoria de datos. El bus de programa lleva el código de instrucciones y operandos inmediatos de la memoria de programa. Los buses de datos y programa pueden llevar información de memoria de datos y memoria interna o externa al multiplicador en un ciclo para operaciones de multiplicación/acumulación. El procesador TMS320C50 tiene un alto grado de paralelismo, esto es, que mientras los datos son operados por la CALU, las operaciones aritméticas también son ejecutadas en la Unidad Aritmética de Registro Auxiliar (ARAU) con lo que se logra un conjunto de instrucciones aritméticas, lógicas y de manipulación de bit que pueden ser efectuadas en un ciclo de máquina.

3.3.2 ORGANIZACIÓN DE LA MEMORIA.

El rango total de direcciones de memoria del TMS320C50 es 224K-palabras donde una palabra son 16 bits. El espacio de memoria es dividido en 4 segmentos específicos: 64K-palabras de memoria de programa, 64K-palabras de memoria local, 32K-palabras de memoria global y 64K-palabras de puertos E/S. La arquitectura del dispositivo permite realizar tres operaciones de memoria en un ciclo de maquina: tomar una instrucción, leer un operando y escribir un operando.

El TMS320C50 tiene una considerable cantidad de memoria dentro del circuito integrado: 2K-palabras de memoria ROM, 9K-palabras de memoria SARAM, y 1056-palabras de memoria DARAM.

La memoria ROM se localiza en la memoria de programa (ver Figura 3.1), mientras que la memoria SARAM puede ser memoria de datos y/o memoria de programa. Por otra parte, la memoria DARAM se encuentra configurada en tres bloques: bloque0 (B0) son 512-palabras localizadas en la memoria de datos o en la memoria de programa, bloque1 (B1) son 512-palabras localizadas en la memoria de datos y bloque2 (B2) son 32-palabras localizadas en la memoria de datos.

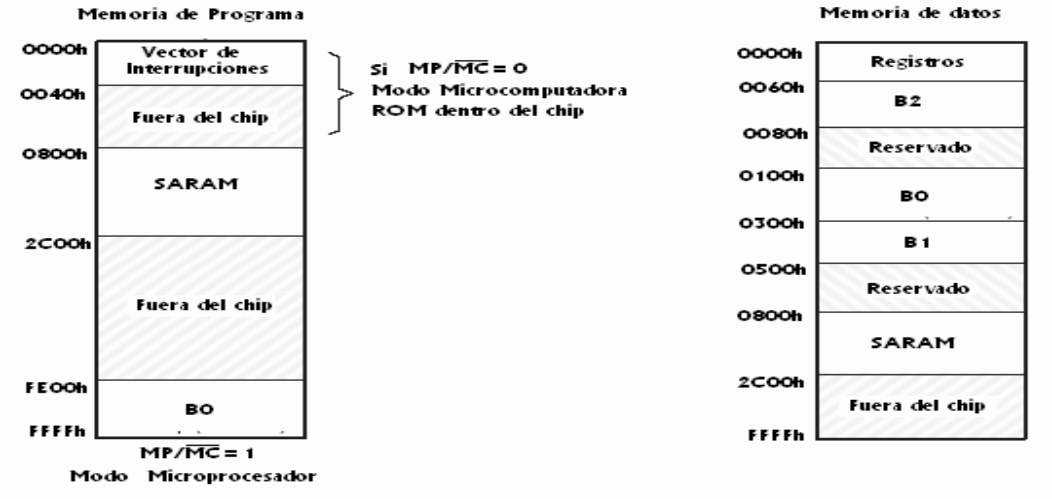


Figura 3.1 Espacio de memoria de datos y de programa.

3.3.2.1 MEMORIA DE PROGRAMA

El TMS320C50 tiene memoria ROM, SARAM y DARAM, las cuales se pueden configurar vía *software* para determinar si se encuentran dentro o fuera del espacio de programa. Cuando se encuentran dentro del espacio de programa, el dispositivo automáticamente accesa a ellas siempre y cuando la dirección a acceder se encuentre dentro de sus límites. Cuando la CALU genera un acceso fuera de estos límites automáticamente realiza un acceso externo.

La memoria de programa puede estar dentro o fuera del circuito integrado. Después del *reset*, la configuración es puesta por el nivel de la terminal MP/$\overline{MC}$. Si esta terminal esta en alto, el dispositivo esta configurado como un microprocesador, y la memoria ROM se encuentra fuera del espacio de programa. Si esta terminal se encuentra en bajo, el dispositivo es configurado como una microcomputadora, y la memoria ROM se encuentra dentro del espacio de programa.

Al ocurrir un *reset*, la memoria SARAM y el bloque B0 no se encuentran dentro del espacio de programa. Para hacer que la memoria SARAM se encuentre en el espacio de programa se debe poner en 1 el bit RAM que se encuentra dentro del registro PMST. En el caso de B0 al poner el bit CNF en 1 que se encuentra dentro del registro ST1 aparecerá dentro del espacio de programa. Esta configuración de la memoria de programa se resume en la Tabla 3.1.

Tabla 3.1 Configuración de la memoria de programa.

CNF	RAM	MP/ $\overline{MC}$	ROM	SARAM	BO	FUERA DEL CIRCUITO INTEGRADO
0	0	0	0000-07FF			0800-FFFF
0	0	1				0000-FFFF
0	1	0	0000-07FF	0800-2BFF		2C00-FFFF
0	1	1		0800-2BFF		0000-07FF
						2C00-FFFF
1	0	0	0000-07FF		FE00-FFFF	0800-FDFF
1	0	1			FE00-FFFF	0000-FDFF
1	1	0	0000-07FF	0800-2BFF	FE00-FFFF	2C00-FDFF
1	1	1		0800-2BFF	FE00-FFFF	0000-07FF
						2C00-FDFF

3.3.2.2 MEMORIA DE DATOS LOCAL

El TMS320C50 tiene memoria SARAM y DARAM, las cuales se pueden configurar vía *software* para determinar si se encuentran dentro o fuera del espacio de datos local. El bloque B0 puede ser configurado dentro del espacio de datos local si se pone el bit CNF en 0 como se muestra en la Tabla 3.2, en cambio si el bit CNF se pone en 1 el bloque B0 será configurado dentro de la memoria de programa como se muestra en la Tabla 3.1, el bit CNF se encuentra dentro del registro ST1. Si se quiere configurar la memoria SARAM dentro del área de datos local basta con poner el bit OVLY en 1 el cual se encuentra dentro del registro PMST (Tabla 3.2).

Tabla 3.2 Configuración de la memoria de datos local.

CNF	OVLY	B0	B1	B2	SARAM	FUERA DEL CIRCUITO INTEGRADO
0	0	100h-2FFh	300h-4FFh	60h-7Fh		0800h-FFFFh
0	1	100h-2FFh	300h-4FFh	60h-7Fh	800h-2BFFh	2C00h-FFFFh
1	0		300h-4FFh	60h-7Fh		0800h-FFFFh
1	1		300h-4FFh	60h-7Fh	800h-2BFFh	2C00h-FFFFh

3.3.2.3 MODOS DE DIRECCIONAMIENTO.

El TMS320C50 provee 6 modos básicos de direccionar la memoria de datos, los cuales son:

- Modo de direccionamiento directo: El direccionamiento directo concatena los 7 bits de la instrucción con los 9 bits del registro DP, formando la dirección de memoria de 16 bits.

- Modo de direccionamiento indirecto: El direccionamiento indirecto accesa a memoria de datos a través de uno de los ocho registros auxiliares (AR0-AR7).

- Modo de direccionamiento inmediato: En el direccionamiento inmediato el dato se encuentra en una porción de la instrucción de 16 bits. Hay dos tipos de direccionamiento inmediato válidos, el direccionamiento corto inmediato es un operando de (8/9/13) bits incluido en la instrucción y el direccionamiento largo inmediato es un operando de 16 bits el cual sigue a la instrucción.
- Modo de direccionamiento a registros dedicados: El direccionamiento a registros dedicados tiene como fin direccionar registros especiales tales como BMAR y DBMR.

- Modo de direccionamiento a registros internos del DSP: El direccionamiento a registros internos del DSP es usado para cargar o almacenar en estos registros sin necesidad de ir a la pagina donde se encuentran (DP=0).

- Modo de direccionamiento circular: El direccionamiento circular es un modo adicional de direccionamiento indirecto que automáticamente va al inicio del bloque de datos cuando se llega al final de este.

En el modo de direccionamiento directo (ver Figura 3.2), la instrucción contiene en los siete bits menos significativos la dirección de memoria (dma). Este campo es concatenado con los 9 bits de DP el cual se encuentra dentro del registro ST0, formando los 16 bits de la dirección de memoria. Así el registro DP apunta a una de las posibles 512 páginas de 128-palabras y los 7 bits especifican la dirección dentro de la página. El registro DP es cargado usando la instrucción LDP o la instrucción LST.

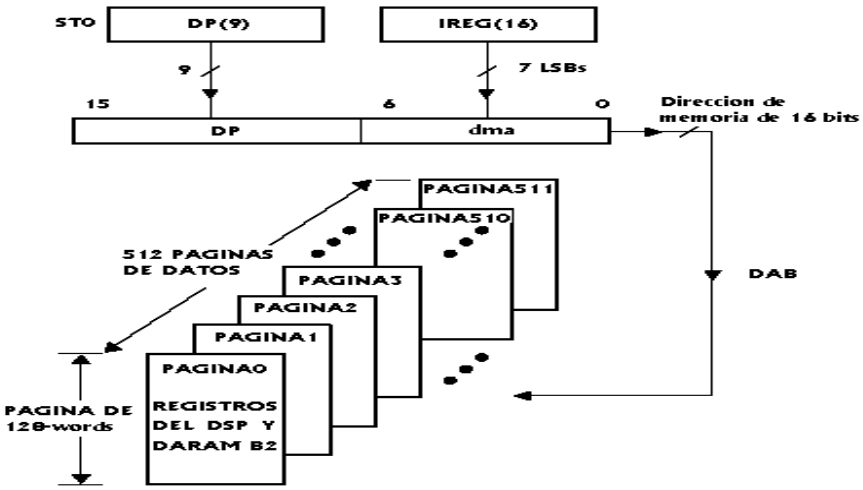


Figura 3.2 Direccionamiento directo.

Cuando se usa el direccionamiento indirecto se puede direccionar cualquiera de las 64K-palabras direcciones de memoria de datos sin importar el valor que contenga DP. Los registros encargados de llevar a cabo el direccionamiento indirecto son los registros auxiliares de 16 bits (AR7-AR0), y los tres bits del registro ST0 correspondientes a ARP.

Como se puede ver en la Figura 3.3 ARP se encarga de apuntar a uno de los ocho registros auxiliares, en este caso AR3, el cuál contiene el valor correspondiente a la dirección de memoria de datos. Como los registros auxiliares son de 16 bits se puede direccionar cualquiera de las 64K-palabras direcciones en la memoria de datos.

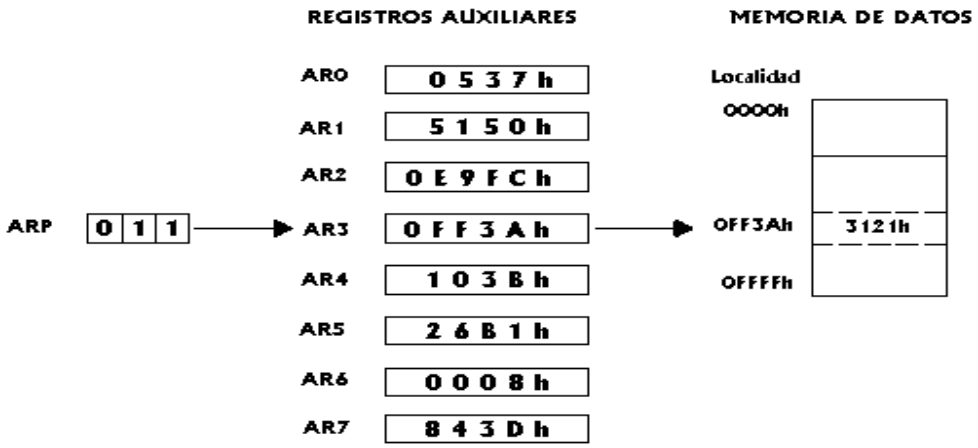


Figura 3.3 Direccionamiento indirecto.

3.3.3 PERIFÉRICOS

En esta sección se presenta la descripción básica de los periféricos, estos periféricos son las interrupciones, el *timer* y el puerto serie cada uno con una función especifica en el desarrollo de este trabajo las cuales se describen a detalle a continuación.

3.3.3.1 INTERRUPCIONES

El DSP tiene 4 interrupciones externas enmascarables ($\overline{INT1}$ - $\overline{INT4}$), las cuales pueden ser utilizadas para interrumpir al DSP. Además, una interrupción externa no enmascarable ($\overline{NMI}$). Las interrupciones internas son generadas por el puerto serie (RINT y XINT), el *timer* (TINT), el puerto TDM (TRNT y TXNT) e interrupciones hechas con *software* (TRAP, NMI y INTR).

La Tabla 3.3 describe la organización y administración de las interrupciones, la localización relativa en el vector de interrupciones y la prioridad para cada una de ellas, ya sean internas o externas.

Tabla 3.3 Organización y administración de las interrupciones.

NOMBRE	LOCALIDAD		PRIORIDAD	FUNCION
	DEC	HEX		
$\overline{RS}$	0	0	1	Señal de reset externo
$\overline{NMI}$	36	24	2	Interrupción no enmascarable
$\overline{INT1}$	2	2	3	Interrupción externa #1
$\overline{INT2}$	4	4	4	Interrupción externa #2
$\overline{INT3}$	6	6	5	Interrupción externa #3
TINT	8	8	6	Interrupción del timer
RINT	10	A	7	Interrupción de la recepción del puerto serie
XINT	12	C	8	Interrupción de la transmisión del puerto serie
TRNT	14	E	9	Interrupción de la recepción de puerto serie TDM
TXNT	16	10	10	Interrupción de la transmisión del puerto serie TDM
$\overline{INT4}$	18	12	11	Interrupción externa #4
--------	20-33	14-21	N/A	Reservado
TRAP	34	22	N/A	Instrucción Trap
--------	38-39	26-27	N/A	Reservado
--------	40-63	28-3F	N/A	Interrupciones hechas por software

La interrupción *reset* ($\overline{RS}$) tiene la mas alta prioridad mientras que la interrupción externa ($\overline{INT4}$) tiene la menor prioridad. La instrucción TRAP no tiene prioridad pero es incluida dentro de la tabla porque tiene una posición en el vector de interrupciones.

El vector de interrupciones puede ser colocado al principio de cualquiera de las 32 páginas de 2K- palabras en la memoria de programa, en donde el parámetro que determina la página en la que se encontrará el vector de interrupciones son los 5 bits del campo IPTR del registro PMST. Por ejemplo, si IPTR = 1 entonces el vector de interrupciones comenzará al principio de la segunda página de 2K-palabras, es decir, en la dirección 0800h.

Cuando una interrupción ocurre esta es almacenada en el registro de banderas (IFR). Esto ocurrirá solamente si la interrupción esta adecuadamente habilitada. Cada interrupción pone una bandera en IFR y puede ser limpiada en cualquiera de las tres siguientes formas:

- El dispositivo está en *reset,*
- El programa toma la interrupción,
- El programa escribe a uno de los bits apropiados en el registro IFR.

El registro IFR está localizado en la dirección 6 en el espacio de la memoria de datos y puede ser leído para identificar interrupciones activas y escribir para limpiar interrupciones.

Cuando una interrupción ocurre los registros ACCU, ACCB, PREG, INDX, ARCR, ST0, ST1 (excepto el bit XF), PMST y los tres registros temporales TREG0, TREG1, TREG2, son almacenados en su correspondiente *stack* con lo que se dice que se guardó el contexto. Al completarse la rutina de interrupción con las instrucciones RETI o RETE los registros son regresados con los valores que quedaron almacenados en los *stacks* y por lo tanto es devuelto el contexto (el estado de todos los registros y acumuladores antes de la interrupción).

3.3.3.2 EL *TIMER*

La Figura 3.4 muestra el diagrama de bloques del *timer.*

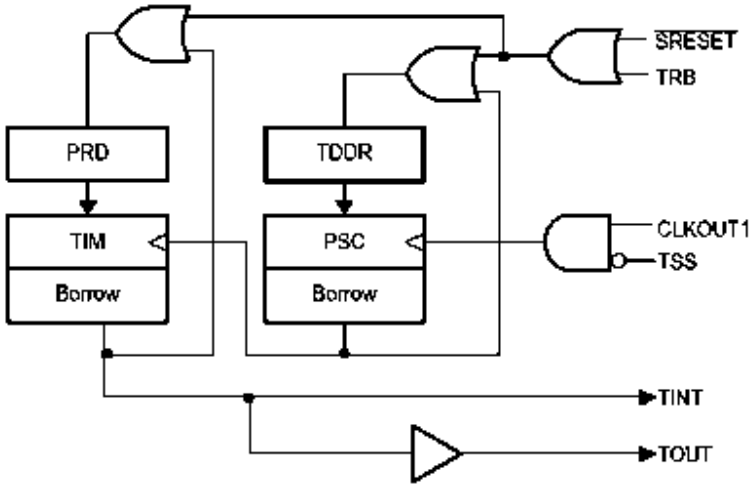


Figura 3.4 Diagrama de bloques del *timer*.

El *timer* es un dispositivo de conteo regresivo que puede ser usado para generar interrupciones periódicas a la CPU (TINT) o bien como señal de reloj (TOUT). Este cuenta con 2 etapas de conteo y cada una de ellas está integrada por un registro de periodo y un registro de conteo regresivo. Además cuenta con entradas que le permiten parar el conteo, iniciar el conteo o bien restablecer el *timer*.

La frecuencia de la señal de reloj y la frecuencia de la interrupción esta dada por:

$$\text{TOUT} = TINT = \frac{1}{t_C(c) \times u \times v} = \frac{1}{t_C(c) \times (TDDR+1) \times (PRD+1)} \qquad (3.1)$$

donde:

$t_C(c)$ es el periodo de CLKOUT1, el cual según el fabricante es de 50×10^{-9} segundos

u es la suma del contenido en TDDR más 1

v es la suma del contenido en PRD más 1

Por lo tanto la frecuencia de interrupción del *timer* es igual a la frecuencia de CLKOUT1 dividida por dos factores independientes.

El registro de periodo y de conteo regresivo para la primer etapa son TDDR y PSC, ambos son campos del registro TCR y cada uno de ellos es de 4 bits. El registro de periodo y de conteo regresivo para la segunda etapa son los registros PRD y TIM, ambos de 16 bits.

La operación del *timer* es controlada vía su registro de control (TCR) el cual se muestra en la Figura 3.5.

- Los bits 0-3 constituyen el campo de TDDR en el registro TCR. Si ocurre un *reset* TDDR es puesto en cero.
- El bit 4 corresponde a TSS, con este bit se puede detener y reestablecer el conteo del *timer*. Si TSS=1 el conteo es detenido y los relojes internos apagan el *timer* permitiendo operar en bajo consumo de potencia. Si TSS=0 el conteo es reestablecido. Al ocurrir un *reset* el bit TSS es puesto en cero por lo que el *timer* inmediatamente inicia la cuenta.
- El bit 5 corresponde a TRB y cuando TRB=1 el periodo del *timer* es recargado. Este bit siempre se lee como cero.
- Los bit 6-9 constituyen el campo del PSC en el registro TCR.
- Los bits 10 y 11 son bits de emulación, los cuales determinan el estado del puerto serial cuando un *breakpoint* es encontrado en un depurador de lenguaje de alto nivel.
- Los bits 12-15 se encuentran reservados para futuras aplicaciones.

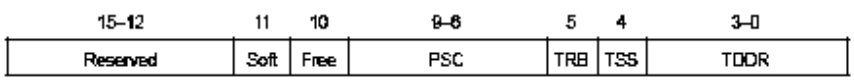


Figura 3.5 Registro de control del *timer*.

3.3.3.3 EL PUERTO SERIE

El puerto serie provee doble comunicación con dispositivos seriales con funcionamiento semejante. Este puerto puede ser utilizado para comunicación con otros DSPs, aunque para ese caso el TMS320C50 cuenta con el puerto serie TDM. Ambos puertos realizan la operación de recibir y transmitir permitiendo un flujo de comunicación continua ya sea en 8 o 16 bits.

El puerto serie opera con 3 registros (SPC, DXR, DRR) y otros dos registros (XSR y RSR) que no son accesibles pero son los encargados de permitir la comunicación bidireccional.

La Figura 3.6 muestra las terminales y registros que se utilizan en el funcionamiento del puerto serie, así como también la doble comunicación que es implementada.

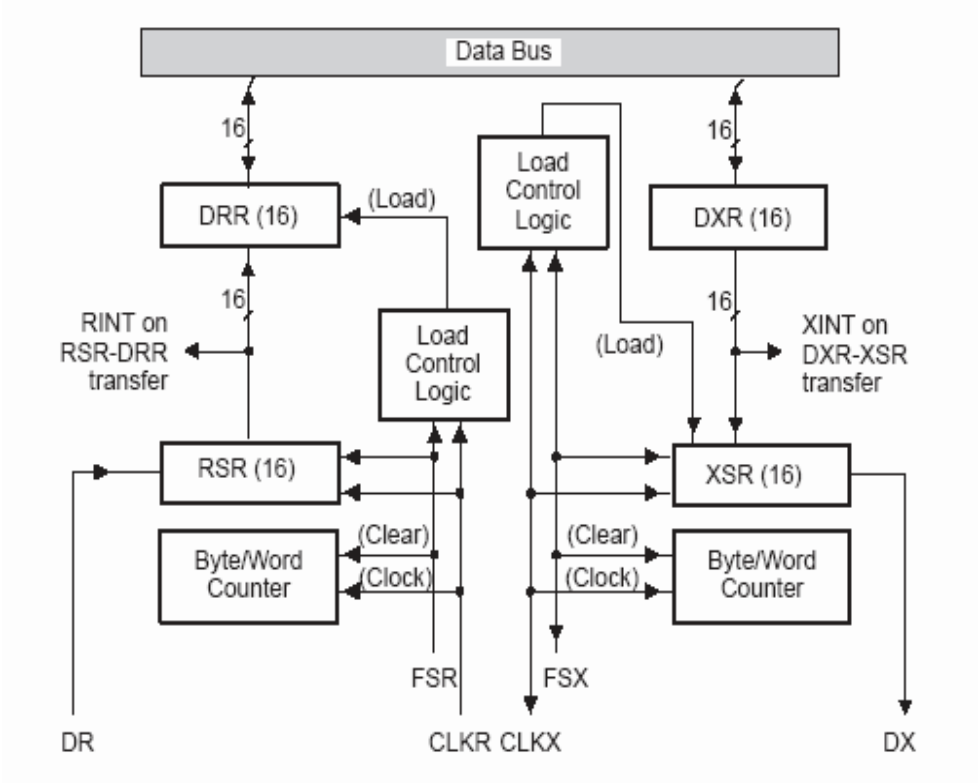
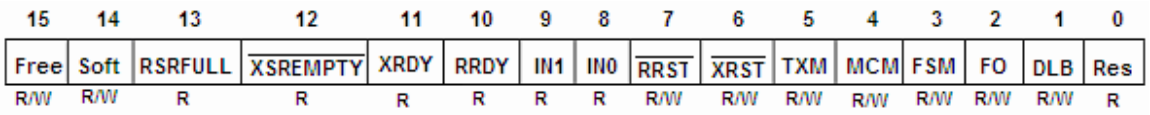


Figura 3.6 Diagrama de bloques del puerto serie.

Para transmitir un dato es necesario escribir el dato en el registro DXR, mientras que para recibir un dato es necesario leer el registro DRR. Una transmisión es ejecutada escribiendo un dato en DXR, el cual es copiado a XSR cuando se encuentra vacío. El registro XSR se encarga de enviar los bits a la terminal DX, por lo que el registro DXR se encuentra listo para recibir un nuevo dato.

Cuando se completa la copia del dato de DXR a XSR, ocurre una transición de 0 a 1 en el bit XRDY en el registro SPC y genera una interrupción XINT, esta señal indica que DXR se encuentra listo para recibir un nuevo dato. El proceso es similar en el caso de la recepción. El dato que llega a través de la terminal DR es guardado en RSR y este se copia en el registro DRR, del cual se puede realizar la lectura. Cuando se completa la copia de RSR a DRR ocurre una transición de 0-1 en el bit RRDY del registro SPC y genera una interrupción RINT. Por lo tanto, el puerto serie tiene comunicación bidireccional ya que un dato puede ser transferido de DXR o DRR mientras otra transmisión o recepción puede ser realizada.

El registro SPC se conoce como el registro de control del puerto serie, es un registro de 16 bits y algunos de sus bits son solamente de lectura mientras que otros son de lectura y escritura. Este registro se detalla en la Figura 3.7.



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Free	Soft	RSRFULL	$\overline{\text{XSREMPTY}}$	XRDY	RRDY	IN1	IN0	$\overline{\text{RRST}}$	$\overline{\text{XRST}}$	TXM	MCM	FSM	FO	DLB	Res
R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Nota R = Lectura , w = Escritura

Figura 3.7 Registro de control del puerto serie.

En la operación del puerto serie en el modo *burst*, hay periodos que el puerto serie se encuentra inactivo entre cada paquete transmitido (paquete se conoce a los 8 o 16 bits transmitidos). La transmisión inicia con una escritura a DXR, el valor en DXR es copiado en XSR, cuando ocurre un pulso de inicio de transmisión FSX (ya sea internamente o externamente generado dependiendo de TXM), el valor en XSR es llevado a la terminal DX. Si DXR es recargado antes que el contenido viejo en DXR haya sido transferido a XSR, el

contenido actual es perdido y se toma el nuevo. DXR es copiado en XSR solo si XSR se encuentra vacío y el DXR ha sido cargado desde la pasada transferencia entre DXR y XSR. Una forma segura de escribir a DXR es esperar a que XINT ocurra ya que es una garantía que la transferencia entre DXR y XSR ya fue realizada.

En la Figura 3.8 se muestra el diagrama de tiempos del puerto serie cuando opera en modo Burst (FSM =1), FSX generado internamente (TXM = 1), CLKX generado internamente (MCM = 1) y transmite 8 bits (FO = 1).

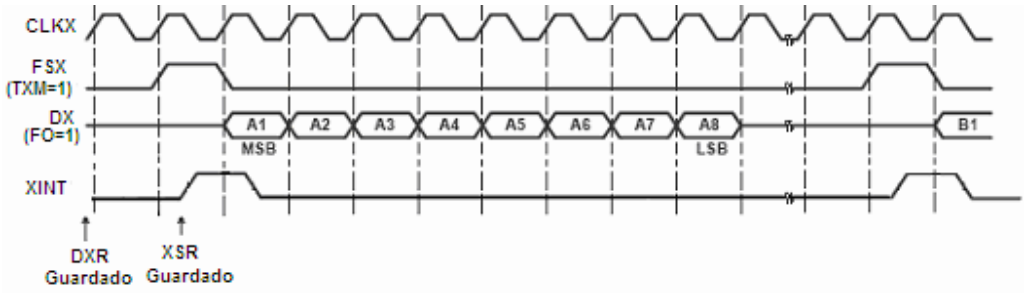


Figura 3.8 Transmisión del puerto serie en modo *Burst* con pulsos generados internamente.

En la Figura 3.9 se muestra el diagrama de tiempos del puerto serie cuando opera en modo Burst (FSM =1), FSX externamente generado (TXM = 0), CLKX externamente generado (MCM = 0) y transmite 8 bits (FO = 1).

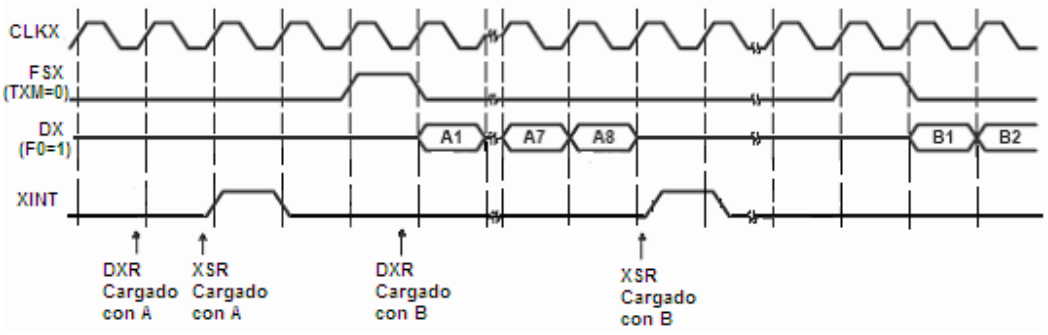


Figura 3.9 Transmisión del puerto serie en modo *Burst* con pulsos generados externamente.

El funcionamiento en los dos casos es el mismo, la única diferencia radica en que en el segundo caso el puerto serie debe de esperar para iniciar la transmisión hasta que un pulso externo sea aplicado a la terminal FSX. En el caso que se haga una escritura a DXR antes que el dato haya sido enviado, este dato se guarda y es enviado a XSR posteriormente cuando se encuentra vacío.

La recepción comienza con un pulso de inicio de recepción en FSR, después de que todos los bits han sido recibidos, el contenido de RSR es transferido a DRR, generando una interrupción RINT (ver Figura 3.10). Si DRR no ha sido leído, XSR esta lleno y además ocurre un nuevo pulso en FSR, la bandera RSRFULL se activa, ocasionando que la recepción se detenga y espera a que DRR sea leído, el dato en RSR es mantenido pero ningún dato nuevo es recibido en DR mientras la recepción este detenida. Únicamente de tres formas se puede limpiar la bandera RSRFULL leyendo DRR, restableciendo el dispositivo o el puerto serie, respectivamente.

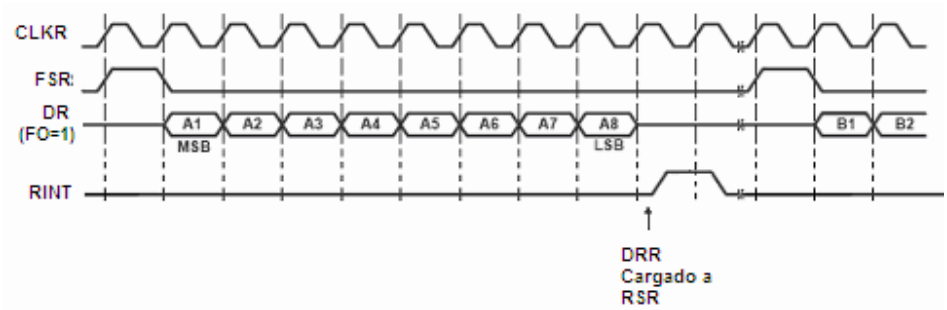


Figura 3.10 Recepción del puerto serie en el modo Burst.

3.4 EL *DSP STARTER KIT* (DSK)

Los elementos principales del DSK utilizados en este trabajo se presentan a continuación.

3.4.1 *HARDWARE* DEL DSK

El *DSP Starter Kit* es una de las herramientas más completas que permiten tener acceso a un circuito integrado DSP, para llevar a cabo la generación, verificación y ejecución del código

para diferentes aplicaciones. Normalmente está diseñado para funcionar desde una computadora, siendo la interfaz de usuario el puerto de comunicaciones seriales (RS-232).

Características del *DSP Starter Kit* [Texas Instrument 1994]:

- Procesador DSP TMS320C50 de Texas Instrument a 40 Mhz.
- Circuito de interfaz analógica (AIC).
- Conector RS-232 para comunicación con la PC.
- Reloj, oscilador de 40 Mhz.
- PROM de 32 Kbytes (Para almacenar el programa kernel)
- Conectores de RCA *Standard* para entrada y salida de audio.
- Alimentación de la tarjeta con un *jack* de 2.1mm.
- Facilidad de expansión con puertos seriales y acceso a terminales del DSP, incluyendo a señales de prueba mediante terminales de extensión.

La Figura 3.11 muestra la vista superior de la tarjeta DSK. En esta tarjeta se encuentran la unidad AIC y el DSP TMS320C50 conectados por medio del puerto serie, la comunicación de datos se hace en 16 bits. El DSP se encarga de suministrar la señal del *reset* y del MSTR CLK a la unidad AIC, mientras que la unidad AIC es la encargada de enviar las señales de FSX, FSR, CLKX y CLKR. La Figura 3.12 muestra como se encuentran conectados estos dos dispositivos en la tarjeta DSK.

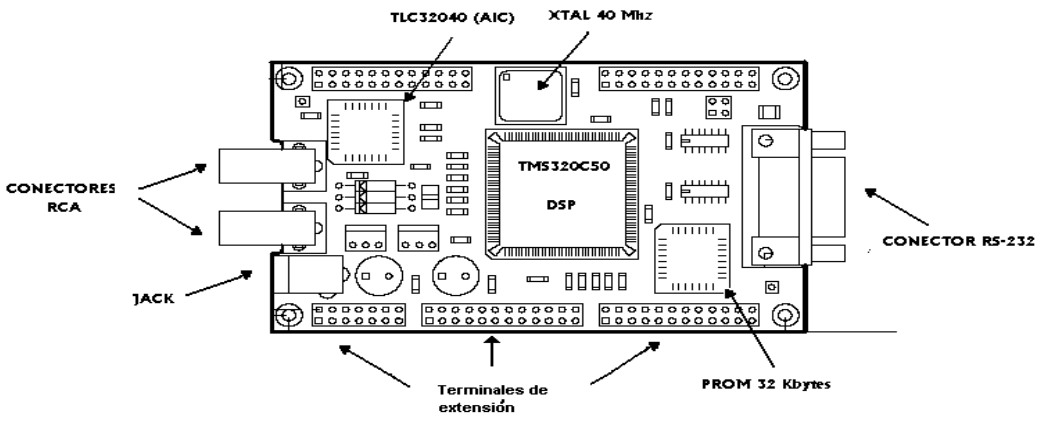


Figura 3.11 Vista superior de la tarjeta DSK.

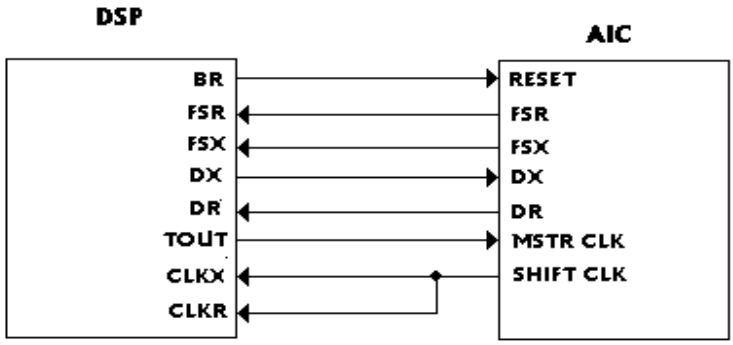


Figura 3.12 Conexión de la AIC y el DSP en la tarjeta DSK.

3.4.2 *SOFTWARE* DEL DSK

El *DSP Starter Kit* incluye un *diskette* que contiene las herramientas de *software* necesarias para producir aplicaciones con el DSP. Tales herramientas son: el ensamblador DSK5A.EXE y el depurador DSK5D.EXE.

3.4.2.1 EL ENSAMBLADOR DEL DSK

El DSK5A.EXE permite convertir un archivo fuente a un archivo ejecutable, el cual se puede invocar desde la línea de comandos DOS de la siguiente forma:

DSK5A < archivo [.asm] > [KL] [asm "código"]

donde:

< archivo [.asm] > es el nombre del archivo que se quiere ensamblar. La extensión [.asm] es opcional.

[L]: Genera un archivo de listado archivo.lst, útil para analizar el código fuente.

[K]: Ignora los errores e intenta generar un ejecutable.

[asm "código"] Pre_ensambla "código" de forma similar a un .include. Esto permite modularidad y reutilizar diferentes rutinas en diferentes aplicaciones.

Los archivos fuente para el ensamblador DSK5A se pueden crear con casi cualquier editor de texto ASCII. Entre los caracteres permitidos se incluyen los alfanuméricos, espacios y tabulaciones. Si se genera un código fuente con un procesador de texto que use caracteres especiales de control para formato, el ejecutable generado por el DSK5A contendrá errores.

Un archivo fuente puede estar ordenado en cuatro campos con la siguiente sintaxis general:

[etiqueta]: nemónico [lista de operandos] [;comentarios]

- La etiqueta es opcional y en el caso que se utilice debe de ir en la columna 1.
- Uno o más espacios deben separar cada campo.
- El campo nemónico y [lista de operandos] forman el OPCODE de la instrucción.
- Los comentarios son opcionales y si van al inicio de alguna línea deberán comenzar con un (* o ;), en el caso que el comentario vaya al final de una línea, un (;) deberá ir al inicio del comentario.
- Se pueden escribir hasta 80 caracteres por línea.

Un aspecto importante al realizar un archivo fuente, es el resolver todos los campos correspondientes a una instrucción determinada (OPCODE). Según la sintaxis de cada instrucción, se tiene un conjunto de operándos asociados (requeridos u opcionales) para implementar la instrucción de forma adecuada, asignándose un valor nulo a un operando no escrito. Al no utilizar un determinado campo de la lista de operandos, se está utilizando el siguiente campo, como el argumento del campo no definido. Por ejemplo, supóngase que se va a usar la instrucción para cargar el acumulador con corrimiento, LACC, con direccionamiento indirecto cuya sintaxis es la siguiente.

LACC (ind) [,shift [,ARn]]

Ahora supóngase que se quiere asignar un corrimiento shift=0 y el registro auxiliar escogido es AR7. Si la instrucción se escribiera:

LACC *, AR7

El ensamblador lo interpretaría como:

LACC *, 7

Asignando un valor shift=7. Así, la forma correcta de escribir la instrucción es:

LACC *, 0, AR7

Con el ejemplo anterior se puede ver la importancia de escribir adecuadamente la instrucción, por lo que si se quiere saber la sintaxis de alguna instrucción es importante revisar el manual del DSP TMS320C5X. Por otra parte, el proceso de ensamblaje del código se puede realizar sin que necesariamente esté conectada la tarjeta a la PC.

3.4.2.2 EL DEPURADOR DEL DSK

EL DSK5D.EXE es una interfaz del usuario y un depurador. Está configurado como una pantalla con diferentes áreas, donde es posible visualizar memoria de datos, memoria de programa, registros internos y estado del procesador. Permite además la carga y ejecución de código generado por el usuario, con posibilidades de ejecución paso a paso y fijación de puntos de detención del programa (*breakpoints*) entre otras. Mediante el DSK5D es posible cargar archivos generados por el ensamblador.

El depurador se invoca desde la línea de comandos DOS, de la siguiente forma:

DSK5D

Como resultado de esta instrucción se abre la ventana del depurador y se tiene acceso a todos sus menús y comandos. En la Figura 3.13 se muestra las áreas que conforman el depurador.

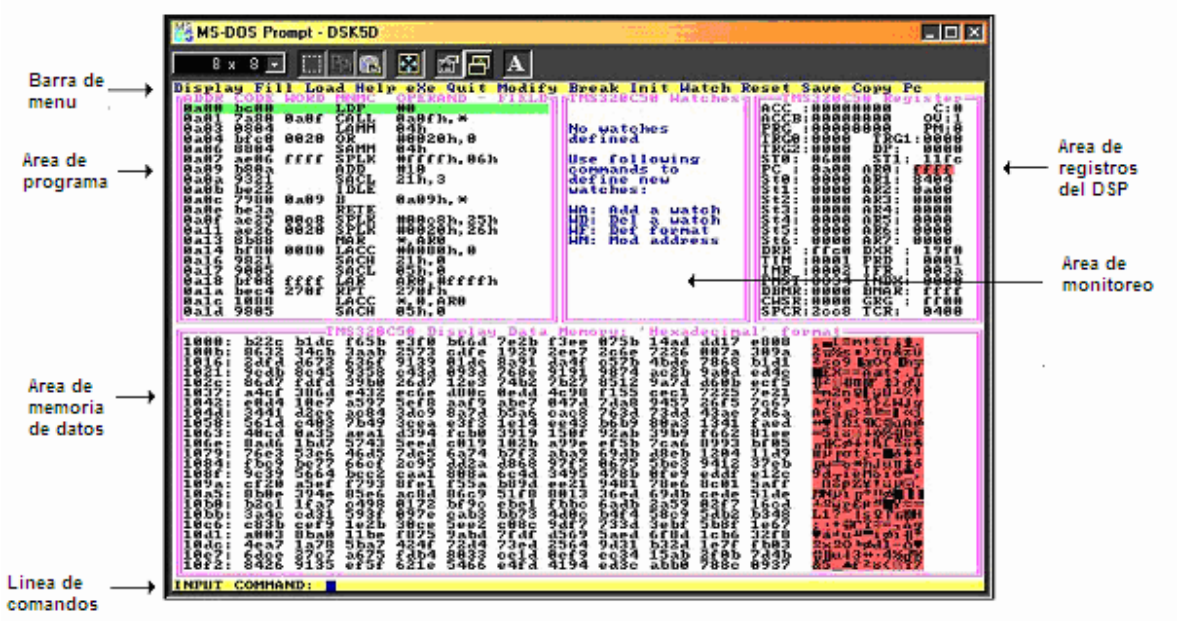


Figura 3.13 El depurador del DSK.

Características del depurador:

- El depurador separa en diferentes áreas el código, datos, comandos y los registros internos del DSP
- No es necesario memorizar los comandos ya que se tiene un menú principal y submenús los cuales contienen los comandos a ejecutar.
- Hay dos maneras de ejecutar los comandos, utilizando la línea de comandos o utilizando los menús.
- Si se teclean mayúsculas o minúsculas tiene el mismo efecto.

Descripción de las áreas del depurador.

Barra de menú

En esta barra se elige cualquiera de las opciones del menú principal simplemente con teclear la letra que se encuentra resaltada. Al hacer esto en esa misma barra aparecerá un submenú con un conjunto de opciones a elegir.

Area de programa

Esta área se encuentra organizada en 5 columnas llamadas ADDR, CODE, WORD, MNMC, OPERAND, las cuales se pueden observar en la Figura 3.14.

Figura 3.14 Area de programa del depurador.

ADDR: En esta columna se especifica la localidad de memoria de programa en la cual se encuentra almacenado el OPCODE de la instrucción.

CODE: En esta columna se encuentra el OPCODE de la instrucción.

WORD: En esta columna se encuentra el operando de los nemónicos, el cual no puede ser parte del OPCODE, sino que necesitan de otros 16 bits.

MNMC: En esta columna aparece el nemónico de la instrucción tal y como se encuentra en el archivo fuente.

OPERAND: En esta columna se encuentra el operando con el cual se va ejecutar el nemónico.

Area de memoria de datos

En esta área se despliega el contenido de las localidades de memoria a partir de una dirección inicial. Esta dirección se puede cambiar así como el formato en el que se despliegan los datos. En la Figura 3.15 se despliega la información de la localidad 1000h hasta la localidad 10f2h usando un formato hexadecimal.

Figura 3.15 Area de memoria de datos del depurador.

Línea de comandos

En esta área se pueden teclear los comandos, o bien si se trabaja con los menús, al teclear alguna letra ésta aparecerá aquí mismo.

Area de monitoreo

Esta área sirve para checar el contenido de alguna localidad de memoria en especifico, también es conocida como *watch*.

Area de registros internos del DSP

En esta área se encuentran los registros del DSP, por lo que se puede ver su contenido sin necesidad de ir a las localidades de memoria asignadas para ellos.

Descripción de las opciones del menú principal

Opción *Display*:

La opción *display* despliega el estado en el que se encuentra la memoria de datos, la memoria de programa, la actual versión del depurador, los bits de *status* del DSP y los registros internos del DSP. Además, se muestra la lista de los puntos de detención del programa *breakpoints*, así como el formato en el que se desea tener desplegado el estado de la memoria.

Opción *Fill*

La opción *fill* sirve para guardar un dato en un número de localidades de memoria que se desee, ya sea en la memoria de programa o en la memoria de datos.

Opción *Load*

La opción *load* sirve para cargar el tipo de archivo que se ejecutará. En nuestro caso se utilizará la opción DSK ya que el ensamblador regresa un archivo de este tipo.

Opción *Help*

La opción *help* se utiliza en caso que se requiera ayuda en la manipulación del monitor.

Opción *eXec*

La opción *exec* se utiliza para ejecutar un programa, con las opciones de ejecutar en tiempo real, hasta donde se encuentre un *breakpoint*, instrucción por instrucción, un número de instrucciones ó al comienzo de alguna función.

Opción *Quit*

La opción *quit* sirve para salir del depurador

Opción *Modify*.

La opción *modify* sirve para modificar los registros internos del DSP, parte del programa que se está ejecutando, alguna localidad en la memoria de datos ó un puerto de entrada/salida.

Opción *Break*

La opción *break* sirve para colocar algún punto de detención del programa en una dirección específica.

Opción *Init*

La opción *init* inicializa los registros internos de la CPU y apunta al inicio del programa que se va a ejecutar.

Opción *Watch*

La opción *watch* sirve para observar el comportamiento de alguna localidad de memoria, a la cual se le puede cambiar el formato en el que se quiera desplegar su contenido, así como modificar el valor del contenido.

Opción *Reset*

Esta opción sirve para restablecer el contenido de memoria.

Opción *Save*

La opción *save* sirve para almacenar en un archivo los datos contenidos de un rango de localidades de memoria de datos o de programa.

Opción *Copy*

La opción *copy* sirve para copiar el dato almacenado en una localidad de memoria de datos a otra, de una localidad de memoria de datos a una de programa, de una localidad de programa a otra localidad de programa y por último, de una localidad de programa a una de datos.

Opción *Pc*

La opción *Pc* sirve para modificar los registros internos del DSP

3.4.3 USO DE LAS OPCIONES MAS IMPORTANTES DEL DEPURADOR DSK

El depurador tiene una gran cantidad de comandos, de los cuales se muestran solo algunos de los más importantes en esta sección.

Los pasos principales para cargar un programa en el DSP son los siguientes:

1. Teclear la letra L correspondiente a la opción Load.
2. Teclear la letra D correspondiente a la opción DSK.
3. Después de realizar el paso 1 y 2 aparecerá una ventana como la de la Figura 3.16, en la cual se escribirá en el campo de texto el nombre del archivo a cargar (se puede omitir la extensión).
4. Por último, se oprime la tecla enter y en el área de programa del depurador aparecerá el código fuente. En este caso el nombre del archivo es MTRY2.

Figura 3.16 Ventana destinada al ingreso del nombre del archivo a cargar en el DSP.

La ejecución del programa en tiempo real requiere los siguientes pasos:

1. Teclear la letra X correspondiente a la opción eXe
2. Teclear la letra R correspondiente a la opción Run

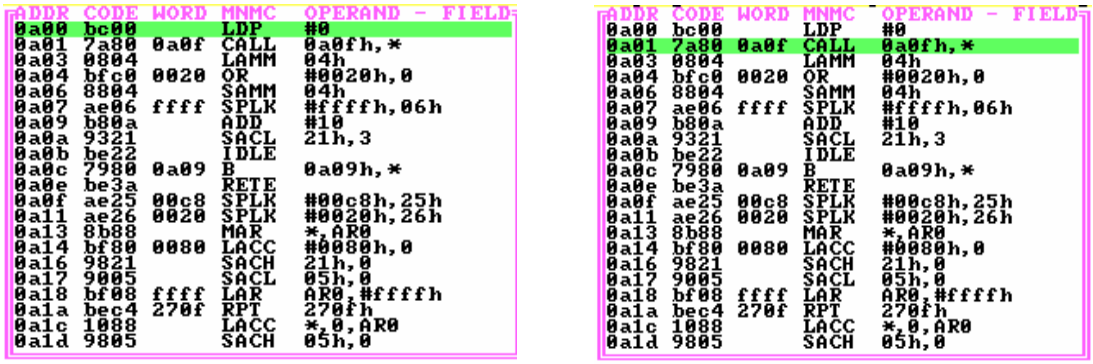
Al realizar los pasos anteriores el programa ya estará ejecutándose en tiempo real y en la línea de comandos aparecerá un mensaje como el de la Figura 3.17. Si se quiere detener la ejecución basta con presionar la tecla Esc.

Figura 3.17 Mensaje desplegado en el área de comandos cuando el DSP está ejecutándose en tiempo real.

La ejecución del programa paso a paso se lleva a cabo de la siguiente manera:

1. Teclear la letra X correspondiente a la opción eXe.
2. Teclear la letra S correspondiente a la opción Single/Step.

Al realizar los pasos anteriores el programa mostrara las siguientes pantallas, ver Figura 3.18 (a)-(b).



```
ADDR CODE WORD MNMC   OPERAND - FIELD
0a00 bc00      LDP    #0
0a01 7a80 0a0f CALL   0a0fh,*
0a03 0804      LAMM   04h
0a04 bfc0 0020 OR     #0020h,0
0a06 8804      SAMM   04h
0a07 ae06 ffff SPLK   #ffffh,06h
0a09 b80a      ADD    #10
0a0a 9321      SACL   21h,3
0a0b be22      IDLE
0a0c 7980 0a09 B      0a09h,*
0a0e be3a      RETE
0a0f ae25 00c8 SPLK   #00c8h,25h
0a11 ae26 0020 SPLK   #0020h,26h
0a13 8b88      MAR    *,AR0
0a14 bf80 0080 LACC   #0080h,0
0a16 9821      SACH   21h,0
0a17 9005      SACL   05h,0
0a18 bf08 ffff LAR    AR0,#ffffh
0a1a bec4 270f RPT    270fh
0a1c 1088      LACC   *,0,AR0
0a1d 9805      SACH   05h,0
```

```
ADDR CODE WORD MNMC   OPERAND - FIELD
0a00 bc00      LDP    #0
0a01 7a80 0a0f CALL   0a0fh,*
0a03 0804      LAMM   04h
0a04 bfc0 0020 OR     #0020h,0
0a06 8804      SAMM   04h
0a07 ae06 ffff SPLK   #ffffh,06h
0a09 b80a      ADD    #10
0a0a 9321      SACL   21h,3
0a0b be22      IDLE
0a0c 7980 0a09 B      0a09h,*
0a0e be3a      RETE
0a0f ae25 00c8 SPLK   #00c8h,25h
0a11 ae26 0020 SPLK   #0020h,26h
0a13 8b88      MAR    *,AR0
0a14 bf80 0080 LACC   #0080h,0
0a16 9821      SACH   21h,0
0a17 9005      SACL   05h,0
0a18 bf08 ffff LAR    AR0,#ffffh
0a1a bec4 270f RPT    270fh
0a1c 1088      LACC   *,0,AR0
0a1d 9805      SACH   05h,0
```

(a) (b)

Figura 3.18 Estado del programa, a).- antes del comando y b).- después del comando.

Para monitorear una localidad de memoria se siguen los siguientes pasos:

1. Teclear la letra W correspondiente a la opción Watch
2. Teclear la letra A correspondiente a la opción Add.
3. Después de haber realizado los dos pasos anteriores aparecerá en la línea de comandos un mensaje como el de la Figura 3.19. En el campo de texto se escribirá la localidad de memoria que se desea monitorear.
4. Posteriormente aparecerá en el área de monitoreo la localidad de memoria elegida con su valor actual, ver Figura 3.20. En este caso se eligió la localidad 0980h.

Figura 3.19 Ventana designada para ingresar la localidad de memoria a monitorear.

Figura 3.20 Area de monitoreo.

La modificación de un registro interno del DSP requiere los siguientes pasos.

1. Teclear la letra M correspondiente a la opción Modify.
2. Teclear la letra R correspondiente a la opción Register.
3. Después de realizar las instrucciones anteriores aparecerá una ventana como la de la Figura 3.21. En esta ventana se escribirá en el campo de texto el registro y el valor que se le asignará.

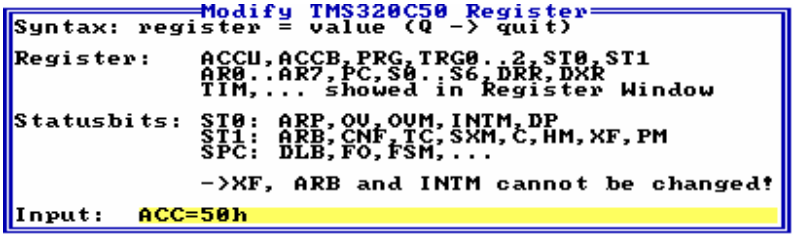
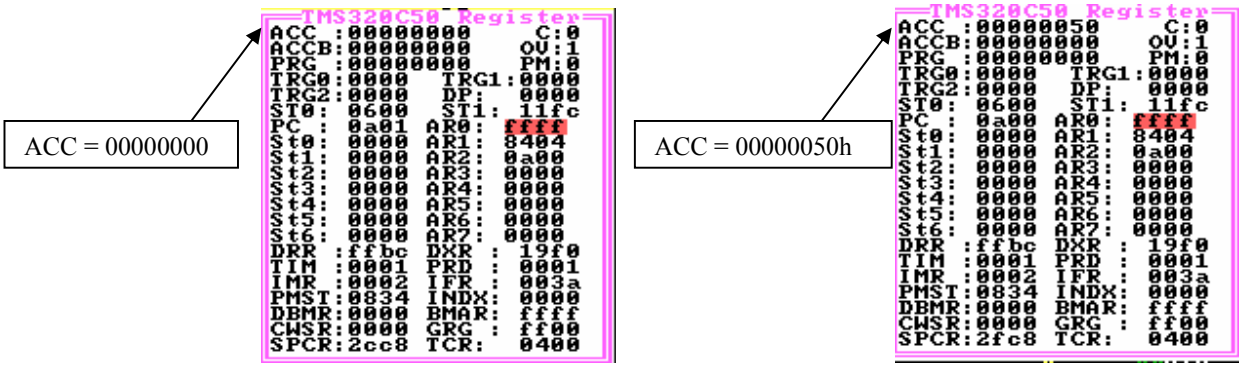


Figura 3.21 Ventana de modificación de los registros internos del DSP.

En este caso se eligió el registro ACCU con un valor de 50h, después de teclear el registro y su valor automáticamente cambiara el valor del registro en el área de registros del DSP, ver Figura 3.22 (a)-(b).

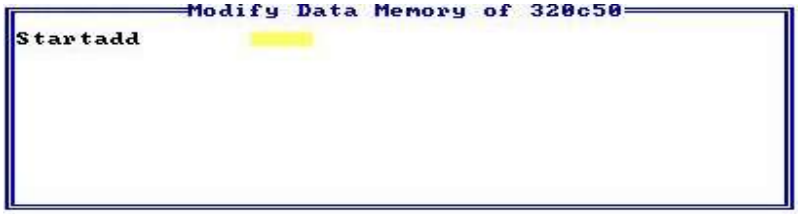


(a) (b)

Figura 3.22 Area de registros internos del DSP, a).- Antes de la instrucción, b).- Después de la instrucción.

Para modificar una localidad de memoria de datos se realizan los siguientes pasos:

1. Teclear la letra M correspondiente a la opción Modify.
2. Teclear la letra D correspondiente a la opción Data.
3. Después de realizar las instrucciones anteriores aparecerá una ventana como la de la Figura 3.23 (a). En el campo de texto se escribirá la localidad de memoria de datos que se va a modificar.
4. Cuando se ingrese la localidad, en esa misma ventana aparecerá dicha localidad con el valor actual, y un campo de texto para ingresar el nuevo valor, ver Figura 3.23 (b). En este caso se eligió la localidad 0980h.
5. Por ultimo se presiona la tecla Esc.



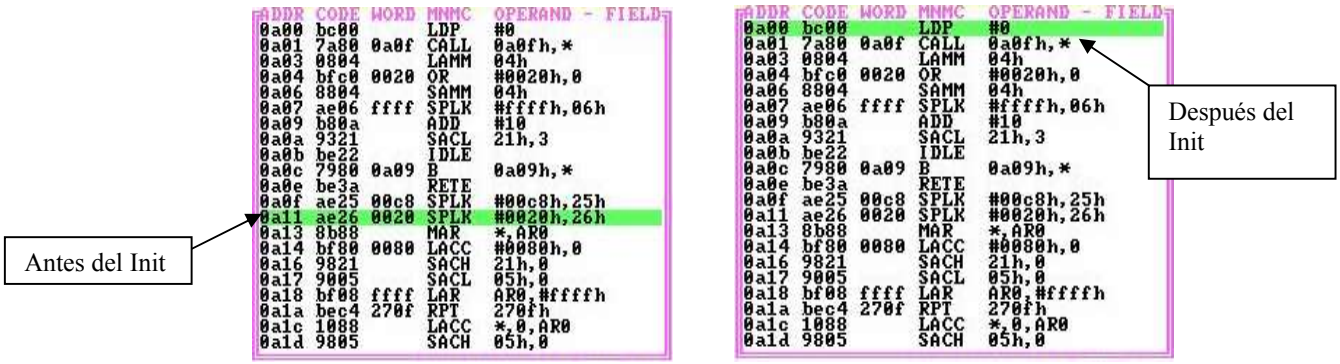
(a)

(b)

Figura 3.23 Ventana de modificación de una localidad en la memoria de datos, a).- Antes de ingresar la localidad, b).- Después de ingresar la localidad.

Para ejecutar el programa nuevamente se sigue la siguiente secuencia:

1. Una vez que ya se corrió el programa para ejecutarlo nuevamente hay que ir al inicio de éste por lo que se debe presionar la tecla I correspondiente al menú INIT. Este cambio se observa en las Figuras 3.24 (a)-(b).



(a) (b)

Figura 3.24 Uso de INIT para ejecutar un programa nuevamente, a).- Antes del INIT, b).- Después del INIT.

Para utilizar un *Breakpoint* en el programa se realizan los siguientes pasos:

1. Teclear la letra B correspondiente a la opción Break.
2. Teclear la letra A correspondiente a la opción Add.
3. Después de realizar los pasos anteriores aparecerá en la línea de comandos un mensaje como el de la Figura 3.25 y una ventana como la de la Figura 3.26 (a).
4. En el campo de texto de la Figura 3.25 se ingresa la dirección en la cual se quiere el *breakpoint.*
5. Automáticamente se habilita el *breakpoint,* ver Figura 3.26 (b).
6. En el área de programa se resaltará la dirección de memoria en la cual se puso el breakpoint (ver Figura 3.27). En este caso se eligió la localidad de programa 0a0ah.

Figura 3.25 Campo de texto destinado para ingresar la localidad del *breakpoint*.

(a)

(b)

Figura 3.26 Ventana de *breakpoints,* a).- Antes de ingresar en el campo de texto la localidad, b).- Después de ingresar la localidad.

Figura 3.27 Localidad elegida dentro del área de programa.

3.5 TIPOS DE INSTRUCCIONES

El DSP TMS320C50 cuenta con un total de 124 instrucciones, organizadas en los siguientes grupos:

- Instrucciones que afectan a la memoria y al acumulador.
- Instrucciones que afectan los registros auxiliares y al registro DP
- Instrucciones que afectan la unidad PLU
- Instrucciones que afectan al registro PREG, TREG0 y realizan multiplicaciones.
- Instrucciones *Branch.*
- Instrucciones que afectan la operación de la memoria y los puertos paralelos.
- Instrucciones de control

En el Apéndice A se incluye la Tabla a.1 la cual resume las instrucciones del DSP TMS320C50 utilizadas en el desarrollo de este trabajo, específicamente en los programas CAP3 y CAP4. Si se desea ver el resto de las instrucciones se puede consultar la referencia [Texas Instrumen 1993].

CAPITULO 4

FILTRO PASABANDA BASADO EN LA UNIDAD AIC

En este capítulo se presenta la descripción detallada de la unidad AIC, la cual abarca desde aspectos físicos hasta su funcionamiento interno. Además se presentan las ecuaciones necesarias para la obtención de las frecuencias de corte de los filtros pasabanda y pasabajas analógicos contenidos en la unidad AIC a partir de los valores que tomen los registros internos de la unidad AIC (TA, TB). Cabe mencionar que estos filtros son de gran importancia en la realización de este trabajo ya que en el capítulo siguiente son parte fundamental en la realización de los cuatro casos de estudio. También se presentan algunos segmentos del código del programa CAP3 el cual tiene como el objetivo configurar el DSP y sus periféricos, esto con el propósito de que se realice la comunicación entre el DSP y la unidad AIC. Por otro lado el *timer* se encarga de suministrar la señal MSTR CLK a la unidad AIC, el puerto serie es el encargado de realizar la comunicación entre el DSP y la unidad AIC, siendo este mismo el medio por el cual el DSP realizará la configuración de los registros internos de la unidad AIC.

4.1 INTRODUCCION

El circuito integrado TLC32040 también conocido como la unidad AIC es un convertidor A/D y D/A que está conectado al DSP a través del puerto serie. Este dispositivo es el encargado de conectar al procesador con el exterior ya que todas la señales analógicas que se desean ingresar al DSP tienen que sufrir una conversión A/D para poderlas procesar. Así mismo, toda señal que tenga que ser enviada hacia el exterior tiene que sufrir un proceso de conversión D/A [Texas Instrument 1994].

4.1.1 CARACTERÍSTICAS DE LA UNIDAD AIC

La unidad AIC es un sistema completo analógico-digital y digital -analógico de entrada y salida, el cual se encuentra contenido en un circuito integrado. Este dispositivo contiene un filtro pasabanda analógico *antialiasing* de capacitor conmutado, con resolución de 14 bits en

la conversión de A/D compatible con 4 modos de puerto serie de microprocesadores, 14 bits en la resolución de D/A y un filtro pasabajas analógico de capacitor conmutado para la reconstrucción de la salida. Además, el dispositivo ofrece numerosas combinaciones de frecuencia de entrada del reloj maestro (MSTR CLK) las cuales pueden ser cambiadas vía el DSP.

El filtro *antialiasing* de entrada está compuesto por un filtro pasabajas de séptimo orden y un filtro pasaaltas de cuarto orden, ambos son del tipo elíptico. El filtro de entrada es precedido por un filtro de tiempo continuo para eliminar cualquier posibilidad de *aliasing* causada por el muestreo del dato. Cuando no se desea utilizar el filtrado, el filtro completo puede ser eliminado de la ruta de la señal. Se pueden seleccionar algunas de las entradas auxiliares disponibles, las cuales se proporcionan para aplicaciones en donde se requiere más de una entrada analógica.

Un voltaje interno de referencia es proporcionado para el TLC32040, lo cuál facilita el desarrollo de aplicaciones y provee control completo sobre la interpretación de este circuito integrado. Mientras el voltaje interno de referencia está disponible para el diseñador en una terminal, las fuentes y tierras analógicas y digitales se encuentran por separado para minimizar el ruido.

El filtro de reconstrucción de salida es de séptimo orden tipo elíptico. Este filtro es seguido por un filtro de tiempo continuo. La Figura 4.1 muestra el diagrama de bloque de la unidad AIC.

Algunas aplicaciones típicas para este circuito integrado incluyen módems, interconexión con DSPs, sistemas de identificación, sistemas de almacenamiento, procesos de control industrial, instrumentación biomédica, procesamiento de señales acústicas, análisis espectral, adquisición de datos e indicadores de instrumentación.

4.1.2 DIAGRAMA DE BLOQUES DE LA UNIDAD AIC

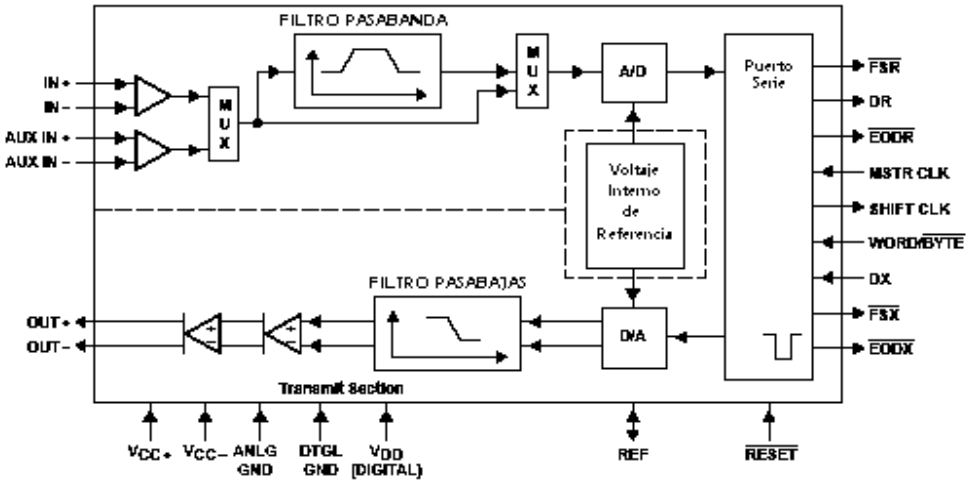


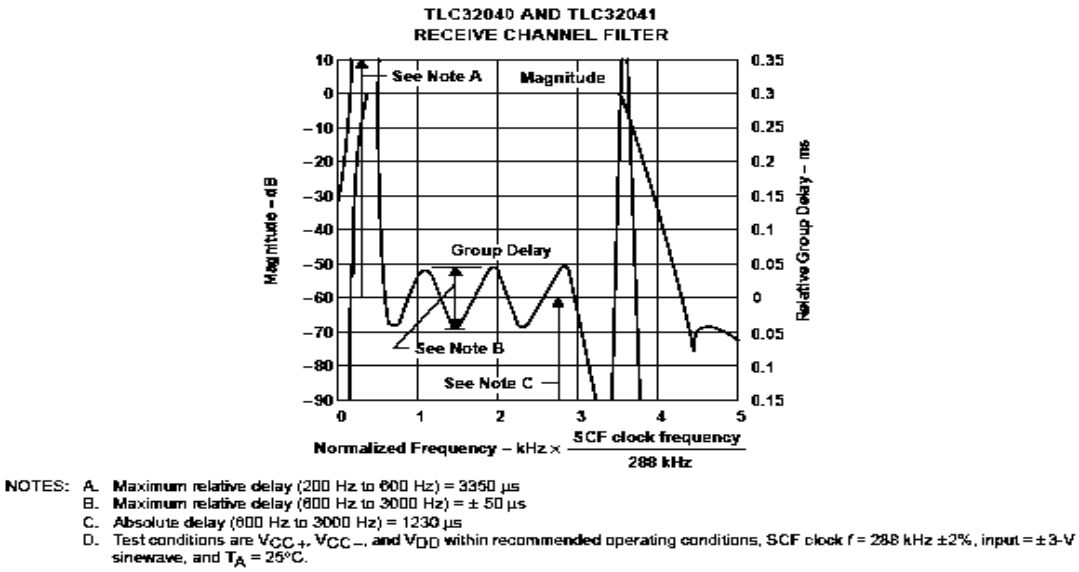
Figura 4.1 Diagrama de bloques de la unidad AIC.

Entradas analógicas.

La unidad AIC tiene dos pares de entradas analógicas. Normalmente se utilizan, el par de entradas IN+ IN-. Sin embargo, las entradas auxiliares AUX IN+ AUX IN- pueden ser usadas si una segunda entrada es requerida. Cada par de entradas puede ser operada en cualquiera de los modos, diferencial o con referencia a tierra. La ganancia para IN+, IN-, AUX IN+, AUX IN- puede ser programada con un valor de 1,2 o 4 para cualquiera de ellas.

Filtro pasabanda A/D, reloj del filtro pasabanda A/D, frecuencia de conversión A/D

El filtro pasabanda A/D puede ser seleccionado a través del registro de control. La respuesta a la frecuencia de este filtro proporcionada por el fabricante es presentada en la Figura 4.2. Esta respuesta resulta cuando la frecuencia del reloj del filtro es 288 Khz. Cuando la frecuencia del reloj del filtro no es 288 Khz la respuesta a la frecuencia del filtro es dimensionada en proporción de la actual frecuencia de reloj.



NOTES: A. Maximum relative delay (200 Hz to 600 Hz) = 3350 µs
B. Maximum relative delay (600 Hz to 3000 Hz) = ± 50 µs
C. Absolute delay (600 Hz to 3000 Hz) = 1230 µs
D. Test conditions are V_{CC+}, V_{CC-}, and V_{DD} within recommended operating conditions, SCF clock f = 288 kHz ±2%, input = ±3-V sinewave, and T_A = 25°C.

Figura 4.2 Respuesta a la frecuencia del filtro pasabanda proporcionada por el fabricante.

La configuración de las frecuencias internas y el protocolo de comunicación secundaria determinan las diferentes opciones para obtener una frecuencia de reloj del filtro pasabanda igual a 288 Khz. Mediante estas dos secciones puede ser programado RX contador A para obtener una frecuencia de 288 Khz para diferentes frecuencias de entrada de reloj maestro.

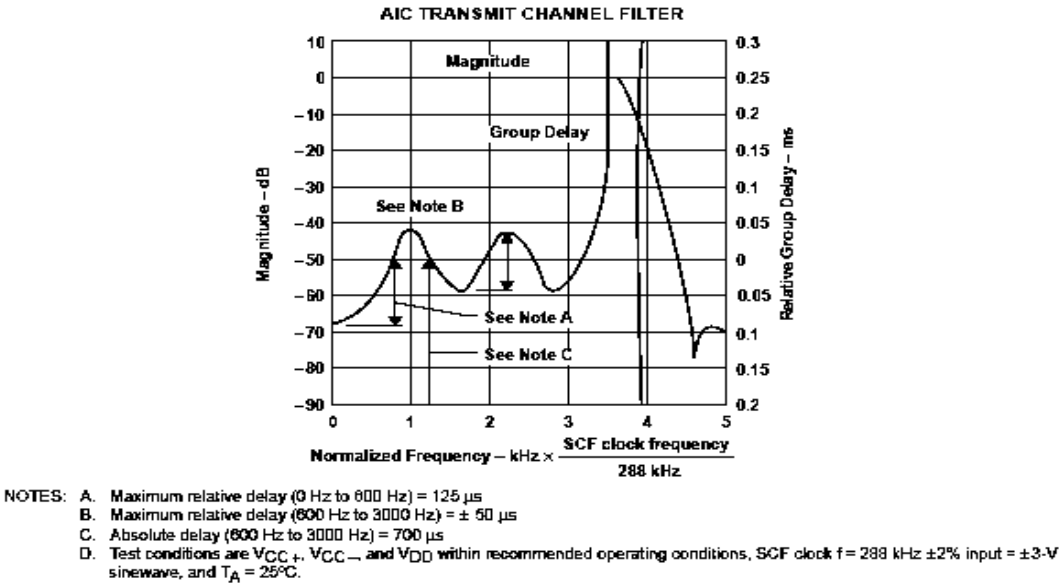
La frecuencia de conversión de analógico a digital es obtenida dividiendo la frecuencia del reloj del filtro pasabanda entre el contenido de RX contador B. De esta manera es prevenido el efecto de traslape porque la frecuencia de conversión A/D es un submúltiplo del reloj del filtro pasabanda y las dos son completamente sincronas.

Salida analógica

El circuito analógico de salida es un amplificador analógico. Ambas salidas, no invertida e invertida son llevadas fuera de este circuito integrado. Este amplificador puede ser conectado a cargas de baja impedancia directamente en cualquiera de las configuraciones ya sea diferencial o con referencia a tierra.

Filtro pasabajas D/A, reloj del filtro pasabajas D/A, frecuencia de conversión D/A

La respuesta a la frecuencia de este filtro es presentada en la Figura 4.3. Esta respuesta resulta cuando el reloj del filtro pasabajas es 288 Khz. Al igual que el filtro pasabanda la respuesta a la frecuencia de este filtro es dimensionada en proporción de la frecuencia del reloj del filtro pasabajas cuando no es 288 Khz. La frecuencia de conversión de digital a analógico es obtenida dividiendo la frecuencia del reloj del filtro pasabajas entre el contenido de TX contador B.



NOTES: A. Maximum relative delay (0 Hz to 600 Hz) = 125 µs
B. Maximum relative delay (600 Hz to 3000 Hz) = ± 50 µs
C. Absolute delay (600 Hz to 3000 Hz) = 700 µs
D. Test conditions are V_{CC+}, V_{CC-}, and V_{DD} within recommended operating conditions, SCF clock f = 288 kHz ±2% input = ±3-V sinewave, and T_A = 25°C.

Figura 4.3 Respuesta a la frecuencia del filtro pasabajas proporcionada por el fabricante.

4.1.3 PRINCIPIO DE OPERACIÓN

Operación asíncrona Vs Operación sincrona

Si la sección de transmisión de la AIC (filtro pasabajas y convertidor digital-analógico "DAC") y la sección de recepción (filtro pasabanda y convertidor analógico-digital "ADC"), son operados asincronamente entonces el reloj del filtro pasabajas y pasabanda son independientemente generados de la señal del reloj maestro. De igual manera, la frecuencia de conversión de A/D y D/A son generadas independientemente.

Si las secciones de transmisión y recepción son operadas de manera sincrona el reloj del filtro pasabajas controlará a ambos, tanto al pasabajas como al pasabanda. En la operación sincrona la frecuencia de conversión A/D es derivada de la frecuencia de conversión D/A y es la misma.

Puerto Serie

El puerto serie tiene 4 posibles modos que son descritos brevemente a continuación.

1. Las secciones de transmisión y recepción son operadas en forma asíncrona y el puerto serie se comunica en dos bytes.

2. Las secciones de transmisión y recepción son operadas en forma asíncrona y el puerto serie se comunica en 16 bits.

3. Las secciones de transmisión y recepción son operadas en forma sincrona y el puerto serie se comunica en dos bytes.

4. Las secciones de transmisión y recepción son operadas en forma sincrona y el puerto serie se comunica en 16 bits.

Operación del TLC32040 con voltaje interno de referencia.

El voltaje interno de referencia del TLC32040 elimina la necesidad de un voltaje externo de referencia y provee una gran reducción en el costo. Por lo tanto, el voltaje de referencia interno facilita el diseño del trabajo y proporciona control completo sobre la realización de este circuito integrado. El voltaje interno es llevado a una terminal y esta disponible para el diseñador.

Operación del TLC32040 con voltaje externo de referencia

La terminal REF puede ser dirigida por un circuito externo de referencia si así se decidiera. Este circuito externo debe de ser capaz de suministrar una corriente de 250 μA y debe de ser protegido tanto del ruido como de la señal analógica.

RESET

Un *reset* es proporcionado para iniciar comunicación serie entre la AIC y el DSP, el *reset* inicializará todos los registros en la AIC incluyendo los registros de control.

Retroalimentación (*LOOPBACK*)

Esta característica permitirá al usuario probar el circuito en un lugar remoto. En retroalimentación, las terminales OUT+ y OUT- son internamente conectados a las terminales IN+ e IN- . Por lo tanto los bits DAC (d15 a d12), los cuales son trasmitidos a la terminal DX pueden ser comparados con los bits ADC (d15 a d12), los cuales son recibidos de la terminal DR.

Una comparación ideal seria que los bits en la terminal DR fueran iguales a los bits en la terminal DX. Sin embargo, en la práctica habrá una diferencia entre estos bits debido a los *offsets* de salida de ADC y DAC.

En retroalimentación, si las terminales IN+ y IN- son habilitadas, las señales externas en las terminales IN+ y IN- son ignoradas. Si las terminales AUX IN+ y AUX IN- son habilitadas, las señales externas a estas terminales son sumadas a las señales OUT+ y OUT- esto en operación de retroalimentación.

4.1.4 DESCRIPCIÓN DE LAS FRECUENCIAS INTERNAS DE LA UNIDAD AIC.

La Figura 4.4 muestra las frecuencias internas de la unidad AIC. Todas las frecuencias internas de la unidad AIC son derivadas de la frecuencia del reloj que llega a la terminal de entrada MSTR CLK. La señal SHIFT CLK es derivada dividiendo la frecuencia de entrada MSTR CLK por 4.

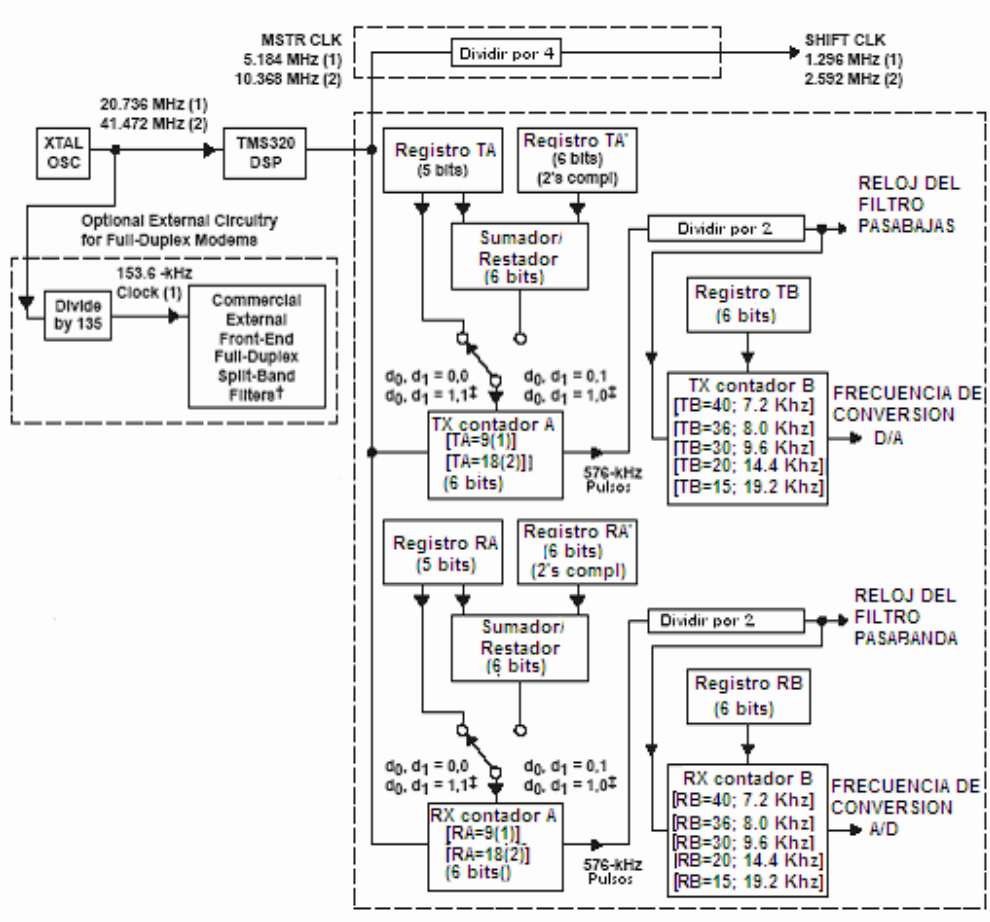


Figura 4.4 Diagrama de bloques correspondiente a las frecuencias internas de la AIC.

Con la Ecuación 4.1 se calcula el valor de la frecuencia interna del reloj del filtro pasabajas.

$$SCF = \frac{MSTR \cdot CLK}{2 \times TA} \tag{4.1}$$

Con la Ecuación 4.2 se calcula la frecuencia de conversión digital a analógico.

$$Frecuencia \cdot de \cdot conversion = \frac{SCF}{TB} \tag{4.2}$$

Con la Ecuación 4.3 se calcula la frecuencia SHIFT CLK

$$SHIFT \cdot CLK = \frac{MSTR \cdot CLK}{4} \tag{4.3}$$

El TX contador A y TX contador B, determinan la frecuencia de conversión D/A, mientras que RX contador A y RX contador B determinan la frecuencia de conversión A/D. Para el filtro pasabajas y pasabanda es necesario ver su respuesta a la frecuencia la cual corresponde cuando el reloj del filtro es 288 Khz. Si la frecuencia de entrada del reloj de los filtros no es 288 Khz, la respuesta a la frecuencia del filtro es dimensionada en proporción al actual valor de frecuencia del reloj. La respuesta a la frecuencia de los filtros cuando su reloj es 288 Khz se puede obtener de diferentes formas combinando el valor del MSTR CLK y el valor de TX contador A y RX contador A. Una vez que se obtiene la frecuencia del reloj de los filtros con el valor de TX contador B y RX contador B se obtiene la frecuencia de conversión de D/A y A/D.

El TX contador A y el TX contador B se habilitan con un valor en cada periodo de conversión de D/A, mientras que RX contador A y RX contador B son cargados cada periodo de conversión de A/D. El TX contador B y RX contador B son cargados con los valores en los registros TB y RB, respectivamente. El TX contador A puede ser cargado con el contenido en el registro TA, TA-TA' o TA+TA' vía *software* control. Si se selecciona la opción TA-TA' la siguiente conversión ocurrirá antes por una cantidad de tiempo que es igual a TA' veces el periodo del MSTR CLK. Si seleccionamos la opción TA+TA' la siguiente conversión ocurrirá después por una cantidad de tiempo que es igual a TA' veces el periodo de MSTR CLK.

Por lo tanto, la frecuencia de conversión D/A puede ser aumentada o decrementada. Una habilidad idéntica es proporcionada para alterar la frecuencia de conversión de A/D. En este caso, el RX contador A puede ser programado vía *software* con el registro RA, RA-RA' o RA+RA'. La habilidad de aumentar o disminuir la frecuencia de conversión es usada particularmente para la aplicación de módems. Esta característica permite controlar cambios en la frecuencia de conversión de A/D y D/A.

Si la transmisión y recepción son configuradas de forma sincrona, entonces el reloj del filtro pasabajas y pasabanda son derivados del TX contador A. También la frecuencia de conversión de A/D y D/A son derivados de TX contador A y TX contador B. Cuando la transmisión y recepción son configuradas de forma sincrona el RX contador A, RX contador B y los registros RA, RA', RB no son usados.

4.1.5 REGISTROS DE CONFIGURACIÓN DE LA UNIDAD AIC

La Figura 4.5 muestra el registro de configuración de la unidad AIC.

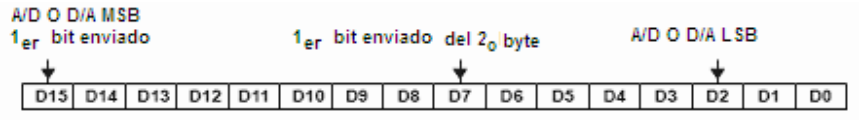


Figura 4.5 Registro de configuración de la unidad AIC.

La configuración de la unidad AIC se lleva a cabo a través del DSP. Para esto es necesario que ocurra una comunicación primaria y una comunicación secundaria. La comunicación primaria ocurre en cada proceso de conversión D/A, es decir, en cada ocasión que el DSP envía un dato por el puerto serie a la unidad AIC, mientras que la comunicación secundaria ocurre únicamente cuando se realiza la configuración de la unidad AIC y para que esto ocurra es necesario que los dos bits LSBs de la comunicación primaria tomen el valor de 1, esto se puede ver en la Tabla 4.1. Si los dos bits LSBs de la comunicación primaria toman el valor de 1 automáticamente se efectuará una pausa de 4 SHIFT CLK ciclos y entonces la siguiente conversión D/A será reconocida como el dato de configuración de la unidad AIC. El significado de este dato de configuración depende nuevamente de los dos bits LSBs y las opciones existentes de configuración se muestran en la Tabla 4.2.

Tabla 4.1 Protocolo de comunicación primaria.

d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0	**COMENTARIOS**
Protocolo de comunicación primaria DX																
←	d15 a d2 van al registro de conversión D/A												→|	0	0	Los contadores TX y RX son cargados con los valores en los registros TA y RA. Los contadores B TX y RX son cargados con los valores en los registros TB y RB.

←	d15 a d2 van al registro de conversión D/A	→|	0	1	Los contadores A TX y RX son cargados con los valores TA+TA'y RA+RA'. Los contadores B son cargados con los valores en los registros TB y RB. Nota: d1=0, d0=1 causara que el próxima periodo conversión de D/A y A/D sea cambiado por la suma TA' y RA' MSTR CLK ciclos, en el cual TA' y RA' pueden ser positivos negativos o cero.
←	d15 a d2 van al registro de conversión D/A	→|	1	0	Los contadores A TX y RX son cargados con los valores TA-TA' y RA-RA' Los contadores B son cargados con los valores en los registros TB y RB Nota: d1=1, d0=0 causara que el próximo periodo conversión de D/A y A/D sea cambiado por la resta TA' y RA' MSTR CLK ciclos, en el cual TA' y RA' pueden ser positivos negativos o cero.
←	d15 a d2 van al registro de conversión D/A	→|	1	1	Los contadores A TX y RX son cargados con los valores TA y RA. Los contadores B TX y RX son cargados con los valores TB y RB. Después de una pausa de 4 SHIFT CLK ciclos, una transmisión secundaria seguirá inmediatamente para programar la AIC.

En la Tabla 4.2 se muestran las opciones en que se puede configurar la unidad AIC y es conocida como protocolo de comunicación secundaria.

Tabla 4.2 Protocolo de comunicación secundaria.

X X |← al registro TA →| X X |← al registro RA →| 0 0	d13 y d6 son los MSB_s (son binarios sin signo)
X |← al registro TA' →| X |← al registro RA' →| 0 1	d14 y d7 son valores con complemento a 2 y se utiliza el bit MSB como bit signo
X |← al registro TB →| X |← al registro RB →| 1 0	d14 y d7 son los MSB_s (son binarios sin signo)
X X X X X X X X d7 d6 d5 d4 d3 d2 1 1	• d2 = 0/1 habilita/deshabilita el filtro pasabanda. • d3= 0/1 inhabilita/habilita la función de retroalimentación. • d4= 0/1 habilita las terminales de entrada AUX IN+ y AUX IN- • d5 = 0/1 asíncronas/ sección de transmisión y recepción sincronas. • d6= 0/1 bits de control de ganancia. • d7 =0/1 bits de control de ganancia.

Tabla 4.3 Control de ganancias y señal de entrada requerida para utilizar la escala completa del convertidor A/D.

CONFIGURACION DE ENTRADA	BITS DEL REGISTRO DE CONTROL		ENTRADA ANALOGICA	ESCALA UTILIZADA
	d6	d7		
Configuración diferencial	1	1	± 6V	Escala completa
	0	0		
Entrada analógica = IN+ - IN-	1	0	± 3V	Escala completa
=AUX IN+ - AUX IN-	0	1	± 1.5 V	Escala completa
Configuración con referencia a tierra	1	1	± 3V	Media escala
	0	0		
Entrada analógica = IN+ - ANLG GND	1	0	± 3V	Escala completa
=AUX IN+ - ANLG GND	0	1	± 1.5V	Escala completa

Tabla 4.4 Ganancias de la unidad AIC.

BITS DEL REGISTRO DE CONTROL		VALOR DE LA GANANCIA
d6	d7	
1	1	1
0	0	1
1	0	2
0	1	4

El voltaje máximo que se le puede aplicar a las entrada de la unidad AIC en configuración diferencial es ±6V, si se utiliza ese voltaje no es necesario configurar la unidad AIC con algún valor de ganancia ya que con esa magnitud se utiliza la escala completa de convertidor A/D. Pero si en cambio se utiliza la configuración con referencia a tierra el voltaje máximo aplicable es ± 3V por lo que es necesario configurar la unidad AIC con una ganancia de 2 para poder utilizar la escala completa de convertidor A/D. Lo anterior se resume en la Tabla 4.3 en término de los bits de control d6 y d7. La Tabla 4.4 muestra los valores de ganancia de la unidad AIC que se pueden especificar dependiendo del estado de los bits 6 y 7.

RESET

Una vez que se realiza un *reset* se inicia comunicación serie entre la AIC y el DSP. El *reset* inicializará todos los registros de la AIC, incluyendo los registros de control. Después de que se ha encendido la AIC, un pulso negativo en la terminal $\overline{RESET}$ inicializará los registros de la AIC y proporcionará una frecuencia de conversión de 8Khz, tanto de A/D como de D/A para una señal de entrada de MSTR CLK de 5.184 MhZ. La AIC quedará inicializada de la siguiente forma:

Tabla 4.5 Inicialización de registros después de un *RESET*.

REGISTRO	VALOR DEL REGISTRO INICIALIZADO (HEX)
TA	9
TA’	1
TB	24
RA	9
RA’	1
RB	24

Los registros de control tomaran los valores siguientes después del *reset*.

d7 = 1, d6 = 1, d5 = 1, d4 = 0, d3 = 0, d2 = 1

Esta inicialización permite comunicación normal por el puerto serie entre la AIC y el DSP. Si las secciones de transmisión y recepción son configuradas para operar sincronas y el usuario desea programar diferentes frecuencias de conversión, solo es necesario programar el registro TA, TA’ y TB, debido a que tanto las frecuencias de conversión de la sección de transmisión como de recepción son derivados de estos registros.

4.1.6 RESPUESTA DE LA AIC A CONDICIONES DE CONFIGURACIÓN ERRÓNEAS

La AIC esta provista para responder a condiciones de configuración erróneas. Estas condiciones y la respuesta de la AIC se muestran en la Tabla 4.6

Tabla 4.6. Respuesta de la AIC a condiciones de configuración erróneas.

CONDICIONES	RESPUESTA DE LA AIC
Registro TA + Registro TA' = 0 o 1 Registro TA – Registro TA' = 0 o 1	Reprograma TX contador A con el valor en el registro TA
Registro TA + Registro TA' < 0	Es usado el modulo aritmético 64 para garantizar que un valor positivo es cargado en TX contador A, i.e Registro TA + Registro TA' + 40 HEX es cargado en TX contador A
Registro RA + Registro RA' = 0 o 1 Registro RA – Registro RA' = 0 o 1	Reprograma RX contador A con el valor en el registro RA
Registro RA + Registro RA' < 0	Es usado el modulo aritmético 64 para garantizar que un valor positivo es cargado en RX contador A, i.e Registro RA + Registro RA' + 40 HEX es cargado en RX contador A
Registro TA = 0 o 1 Registro RA= 0 o 1	La AIC se apaga
Registro TA < 4 en modo WORD Registro TA < 5 en modo BYTE Registro RA < 4 en modo WORD Registro RA < 5 en modo BYTE	El puerto serie de la AIC no puede operar
Registro TB = 0 o 1	Reprograma el registro TB con un 24 HEX
Registro RB = 0 o 1	Reprograma el registro RB con un 24 HEX
La AIC y el DSP no se pueden comunicar	Sostiene la ultima salida DAC

Si la diferencia entre dos pulsos sucesivos de conversión D/A es menor que 1/19.2 Khz, la AIC opera inadecuadamente. En esta situación el segundo pulso de conversión de D/A ocurre rápidamente y no es suficiente tiempo para que la actual conversión sea completada. Esta situación puede ocurrir si los registros TA y TB son programados inadecuadamente o si el resultado de los registros TA+TA' o TA-TA' es pequeño. Cuando se esta ajustando un

aumento en el periodo de conversión vía la opción TA+TA', el diseñador deberá tener mucho cuidado con no violar estos requerimientos (ver Figura 4.6).

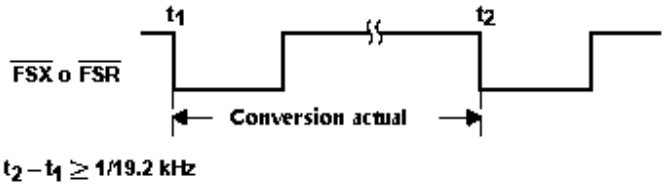


Figura 4.6 Tiempo de conversión mínimo permisible por la unidad AIC.

4.2 CONFIGURACION DE LA UNIDAD AIC

El propósito de configurar la unidad AIC es controlar el funcionamiento de la misma y así poder adaptar sus características de la manera que lo requiera nuestra aplicación. Esta configuración se basa en los siguientes parámetros.

- El registro de control (ganancia, retroalimentación, modo sincrono/asincrono etc.)
- El registro TA (y RA si el modo sincrono se encuentra deshabilitado)
- El registro TB (y RB si el modo sincrono se encuentra deshabilitado)

Al elegir los parámetros de configuración de la unidad AIC de tal manera que opere de forma sincronía, podemos calcular la frecuencia de muestreo con que trabajará la unidad AIC y las frecuencias de corte de los filtros analógicos de la unidad AIC por medio de las ecuaciones 4.4, 4.5 y 4.6 las cuales se muestran a continuación [Delauriere y Grislain].

La frecuencia de corte del filtro pasabajas de salida y del filtro pasabajas correspondiente al filtro pasabanda de la unidad AIC se calculan aproximadamente con la ecuación,

$$F_H = \frac{MSTR \cdot CLK \times 3600}{288000 \times TA} \tag{4.4}$$

En caso que se requiera la frecuencia de corte del filtro pasaaltas correspondiente al filtro pasabanda de la unidad AIC se calcula con la siguiente ecuación,

$$F_L = \frac{MSTR \cdot CLK \times 300}{288000 \times TA} \tag{4.5}$$

Y para calcular la frecuencia de muestreo de la unidad AIC se utiliza la ecuacion siguiente,

$$F_m = \frac{MSTR \cdot CLK}{2 \times TA \times TB} \tag{4.6}$$

donde:

MSTR CLK es suministrado por la terminal TOUT del DSP.

TA es un entero en el rango de 4 a 31.

TB es un entero en el rango de 2 a 63.

4.3 APLICACIÓN DEL FILTRO PASABANDA ANALOGICO DE LA UNIDAD AIC

Para poder llevar a cabo la aplicación del filtro pasabanda analógico que se encuentra en el *DSP Starter Kit* no solo es necesario configurar la unidad AIC sino que también se necesita configurar el DSP, el puerto serie y el *timer* a fin de que funcione correctamente todo el sistema. La inicialización del sistema puede ser dividida en tres distintas partes:

- Inicialización del DSP (Estados de espera (*Wait States*), modo de extensión de signo, registro de interrupciones globales, etc).
- Inicialización del *timer* y del puerto serie ya que el timer es el encargado de suministrar la frecuencia al MSTR CLK de la unidad AIC, y en el caso del puerto serie es el encargado de la comunicación con la AIC.
- Inicialización de la AIC (frecuencia de muestreo, frecuencias de corte y otros parámetros).

A continuación se muestran los segmentos de código del programa CAP3 que se encuentra en el Apéndice B. Este programa además de configurar la unidad AIC, el DSP, el *timer* y el puerto serie también configura el puerto serie para que trabaje por interrupciones. Una vez que se realizó la configuración mencionada anteriormente el programa se encarga de muestrear una señal y cada muestra que se obtiene es enviada al DSP a través del puerto serie, al llegar al DSP la muestra es enviada de regreso a través de puerto serie a la unidad AIC. El funcionamiento del programa se explica más a detalle en el capítulo 5.

La Figura 4.7 muestra el código necesario para la inicialización del DSP TMS320C50. La instrucción SETC INTM deshabilita las interrupciones globales y las vuelve a habilitar con CLRC INTM hasta que la parte de inicialización haya terminado. La instrucción OPL #0834h, PMST es la encargada de poner los bits OVLY, RAM e IPTR en 1 y al hacer esto la memoria SARAM se encuentra tanto en la memoria de datos como en la memoria de programa. Los bits de IPTR son los encargados de configurar el vector de interrupciones, al tomar el valor de 1 el vector de interrupciones se encuentra en la dirección 0800h de la memoria de programa. Con las instrucciones LACC #0, SAMM CWSR y SAMM PDWSR, se deshabilitan los *Wait States.*

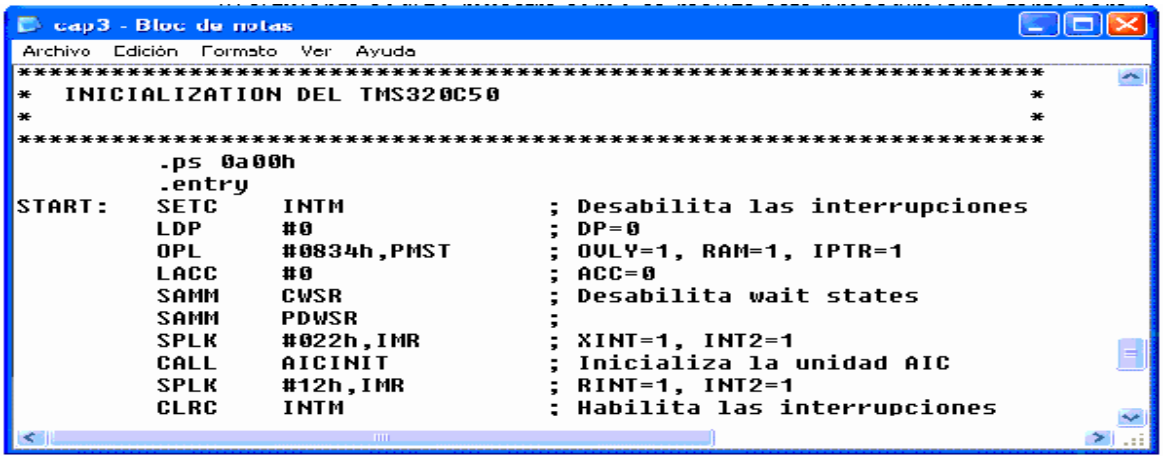


Figura 4.7 Código de inicialización del DSP TMS320C50.

La Figura 4.8 muestra el código necesario para la inicialización del puerto serie y el *timer*. Este código contiene las siguientes instrucciones:

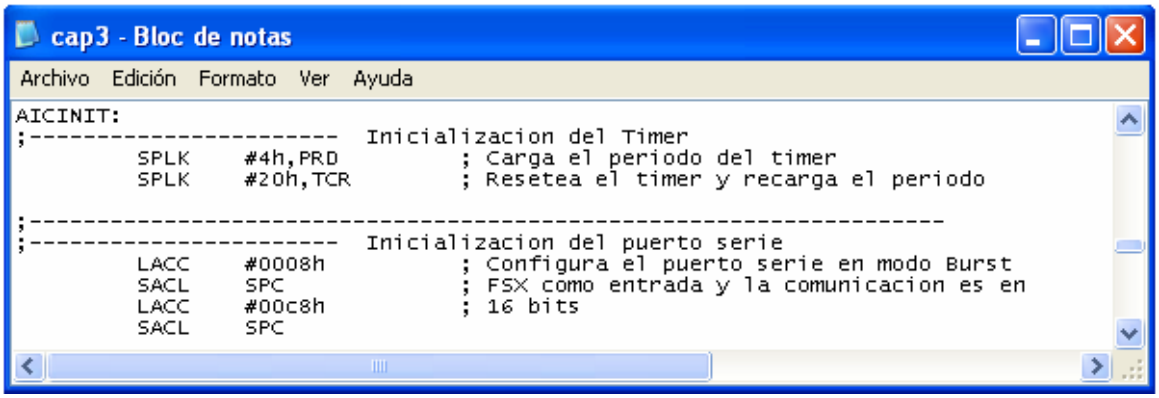


Figura 4.8 Código de inicialización del puerto serie y del *timer*.

- Con la instrucción SPLK #4, PRD se carga el periodo del timer en el registro PRD.
- Con la instrucción SPLK #20h, TCR se restablece el *timer* y recarga el periodo.
- Con las instrucción LACC #0008h y LACL SPC se restablece el puerto serie y los bits de configuración toman los valores MCM=0, FSM=1, FO=0 y DLB=0. De esta manera el puerto serie estará configurado para operar en modo *Burst*, la terminal FSX se configura como entrada, se comunica en 16 bits, la señal de reloj CLKX es externamente generada y la retroalimentación digital se encuentra deshabilitada.
- Finalmente con las instrucciones LACC # 00C8h se da de alta el puerto serie y está listo para trabajar.

Para configurar la unidad AIC es necesario aplicar primero un *reset*, lo cual se realizara por medio del código que se encuentra en la Figura 4.9.

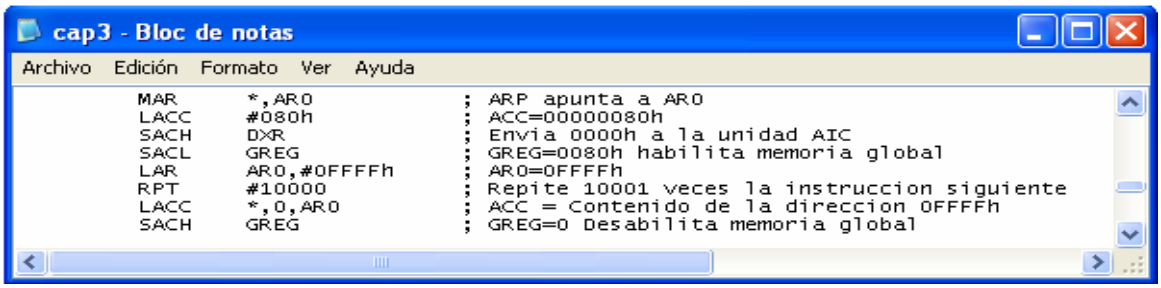


Figura 4.9 Código que suministra un *reset* a la unidad AIC.

La unidad AIC se restablece por medio de la terminal BR del DSP. Esta terminal se activa cada vez que se tiene acceso a la memoria global, por eso esta parte del código tiene como función habilitar la memoria global con la instrucción SACL GREG. Posteriormente carga a AR0 con el valor 0FFFFh y repite 10001 veces la instrucción LACC *, 0, AR0. Esta instrucción lo que hace es cargar el acumulador con el contenido de la dirección a la que apunta AR0 como esta localidad es memoria global, mantendrá el *reset* en la AIC y posteriormente con la instrucción SACH GREG se deshabilita la memoria global.

Después de ejecutar el *reset* se puede configurar la unidad AIC pero antes hay que realizar una comunicación primaria para posteriormente poder realizar la comunicación secundaria. Una vez que se inicio la comunicación secundaria se debe tener cuidado en acomodar el dato de tal manera que cumpla con el formato con el que se debe de mandar y para ello se utiliza el acumulador ACC. Por ejemplo, cuando se manda TA y RA debe observar el formato siguiente (ver el protocolo de comunicación secundaria).

X X (Los 5 bits de TA) X X (Los 5 bits de RA) 0 0

El código de la Figura 4.10 muestra como se realiza este procedimiento tanto para TA, RA, TB, RB y la modificación del registro de control donde la variable AIC_CTR contiene el valor con el que se configurará el registro de control de la unidad AIC.

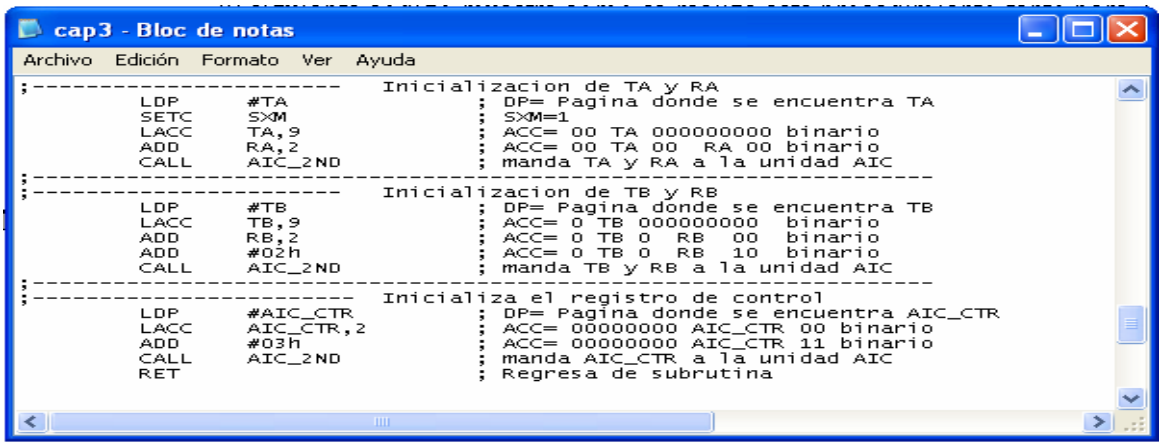


Figura 4.10 Código de programa que adapta la información al formato de configuración secundaria.

Cada vez que un dato de configuración (TA y RA, TB y RB, TA' y RA' o AIC_CTR) es arreglado de acuerdo al formato especificado por la Tabla 4.2 entonces se llama a la subrutina AIC_CTR2. Esta rutina es la encargada de configurar la unidad AIC, esto es, es la encargada de llevar a cabo el protocolo primario y el protocolo secundario. La Figura 4.11 muestra el código de la subrutina AIC_CTR2, la cual se resume en los siguientes aspectos:

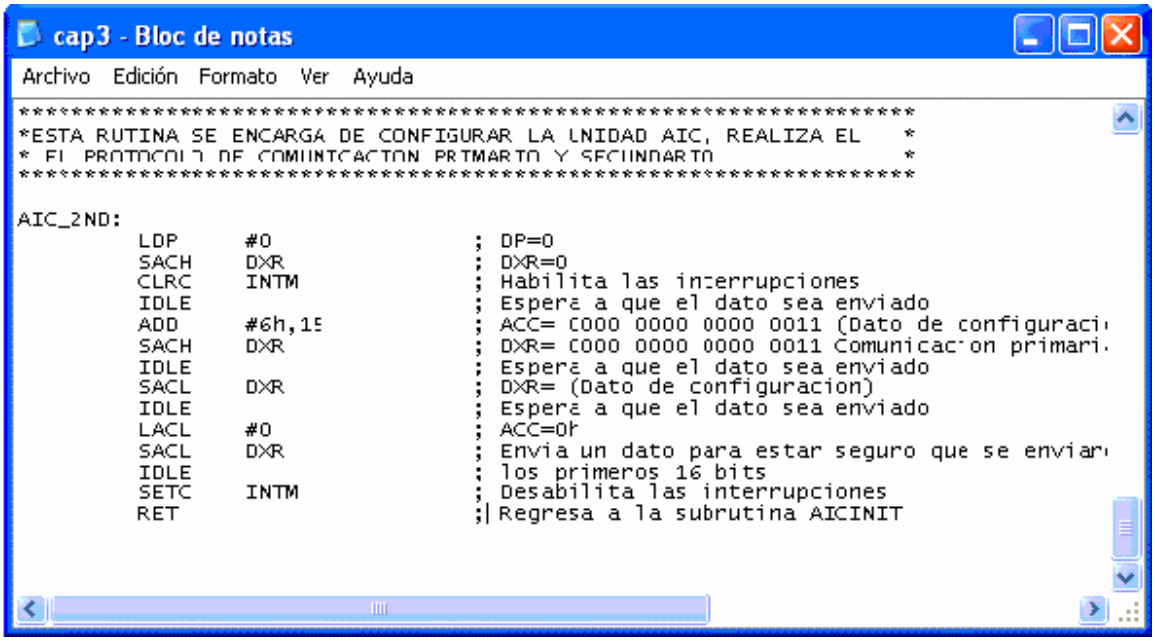


Figura 4.11 Código de configuración de la unidad AIC.

- Lo primero que realiza es regresar a la pagina 0 con la instrucción LDP #0, en la cual se encuentran los registros internos de DSP.
- Posteriormente envía los 16 MSBs de ACC al registro DXR con la instrucción SACH DXR.
- Con la instrucción CLRC INTM habilita las interrupciones y espera a que ocurra la interrupción con la instrucción IDLE, esto con el propósito de estar seguro que el dato fue enviado.
- Con la instrucción ADD #6, 15 lo que hace es guardar en los bits 17 y 18 un 1 mientras que en los 16 bits LSBs del acumulador ACC guarda el dato de configuración.

ACC = 0000 0000 0000 0011 (El dato de configuración)

- El dato de configuración puede ser TA y RA , TB y RB, TA’ y RA’ o el valor correspondiente al registro de control el cual es guardado en la variable AIC_CTR.
- Con las instrucciones SACH DXR se envían los 16 MSBs bits de ACC a la unidad AIC. Debido a que los 2 bits menos significativo del dato enviado son 1 entonces se puede iniciar la configuración secundaria, (ver protocolo de comunicación primaria).
- Por último, con la instrucción SACL DXR se envía el dato de configuración de la unidad AIC.

CAPITULO 5

CASOS DE ESTUDIO

En este capítulo se presentan cuatro casos de estudio los cuales son: la obtención de la respuesta a la frecuencia del filtro pasabajas de la unidad AIC, el fenómeno de traslape, el filtrado de una señal con armónicos y, el muestreo y almacenamiento de una señal en el DSP. Estos cuatro casos de estudio tienen como objetivo estudiar el desempeño de la unidad AIC, cuyo funcionamiento se basa en la configuración de la misma.

La configuración de la unidad AIC en los cuatro casos de estudio tendrá algunas similitudes la cuales son:

- La unidad AIC siempre trabajará en modo sincrono esto con el fin de controlar las frecuencias internas de la unidad AIC únicamente con los registros TA y TB.
- El modo de retroalimentación en los cuatro casos de estudio permanecerá desactivado, de igual forma las entradas auxiliares permaneceran deshabilitadas ya que las entradas IN+ e IN- son las que se utilizan en los cuatro casos de estudio.
- La ganancia tendrá un valor de dos debido a que el máximo voltaje de entrada es 3 volts por lo tanto se requiere dicha ganancia para lograr utilizar la escala completa del convertidor A/D.
- El filtro pasabanda será habilitado o deshabilitado dependiendo del caso de estudio.
- Por otra parte el *timer* proporcionará diferentes frecuencias a la terminal MSTR CLK y será controlado únicamente con el registro de periodo PRD debido a que el valor de frecuencias de MSTR CLK que se utilizarán en lo cuatro casos de estudio se pueden obtener únicamente con el registro PRD sin la necesidad de cargar un valor en TDDR. Debido a esto TDDR siempre tomará un valor de cero.
- Los valores que tomarán los registros de configuración TA, TB y la frecuencia de MSTR CLK dependerán de los valores de las frecuencias de corte de los filtros de la unidad AIC y la frecuencia de muestreo con la que se configure. Estos valores se calculan con ayuda de las ecuaciones (4.4), (4.5) y (4.6).

5.1 CASO DE ESTUDIO 1: ESTUDIO DEL FILTRO PASABAJAS DE LA UNIDAD AIC

Este caso de estudio tiene como propósito el comprobar matemáticamente y experimentalmente que efectivamente el filtro pasabajas de salida de la unidad AIC es un filtro elíptico de orden 7. Debido a esto se presentan las ecuaciones que describen este filtro, las cuales se utilizan para calcular el orden de un filtro elíptico a partir de sus características básicas. Además, se realiza la comparación de la respuesta a la frecuencia del filtro pasabajas elíptico de la unidad AIC programado con una frecuencia de corte de 3477.22 Hz obtenida por medio de mediciones con la respuesta a la frecuencia de un filtro pasabajas elíptico obtenida en Matlab.

5.1.1 CALCULO DEL ORDEN DE UN FILTRO PASABAJAS ELIPTICO

Para el diseño de un filtro elíptico es necesario especificar algunas características básicas, las cuales son:

1. La banda de paso esta definida para $0 \leq \omega \leq \omega_{p,}$ en donde la magnitud del máximo rizo en la banda de paso esta definido por K_p en dB.
2. La banda de paro esta definida para $\omega \geq \omega_s$ con la magnitud del mínimo rizo definido por K_s en dB.
3. La banda de transición esta definida para $\omega_p \leq \omega \leq \omega_{s.}$

La Figura 5.1 muestra la respuesta característica de un filtro elíptico. Además se muestran las características mencionadas anteriormente.

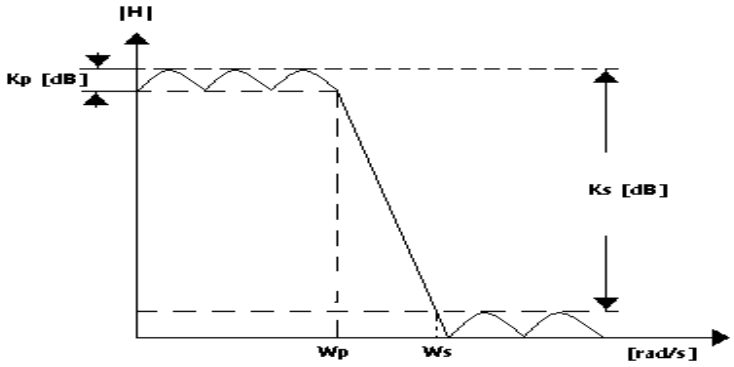


Figura 5.1 Respuesta característica de un filtro elíptico pasabajas.

Si se conocen las características de un filtro elíptico entonces se puede calcular el orden a partir de las siguientes ecuaciones [Internet 2005],

$$\Omega = \frac{\omega_S}{\omega_p} = \frac{f_s}{f_p} \tag{5.1}$$

$$\psi = \sqrt{\frac{10^{0.1Ks-1}-1}{10^{0.1Kp-1}-1}} \tag{5.2}$$

$$C = \frac{1}{16\psi^2}\left(1+\frac{1}{2\psi^2}\right) \tag{5.3}$$

$$D = \frac{\sqrt{\Omega}-1}{2\left(\sqrt{\Omega}+1\right)} \tag{5.4}$$

$$F_E(x) = \frac{1}{\pi}\ln\left(x + 2x^5 + 15x^9\right) \tag{5.5}$$

Y el orden del filtro se calcula con:

$$n_E \geq F_E(C)F_E(D) \tag{5.6}$$

donde:

ω_s frecuencia de inicio de la banda de paro en (rad/seg)

ω_p frecuencia final de la banda de paso en (rad/seg)

f_s frecuencia de inicio de la banda de paro en (Hz)

f_p frecuencia final de la banda de paso en (Hz)

K_s magnitud mínima del rizo en la banda de paro en (dB)

K_p magnitud máxima del rizo en la banda de paso en (dB)

n_E orden del filtro elíptico

En el manual del fabricante del *DSP Starter Kit* se presentan dos tablas las cuales hacen referencia al comportamiento de los filtros para diferentes valores de frecuencia, los valores contenidos en las tablas se obtuvieron con una frecuencia de reloj de los filtros CLK=288 Khz y una señal senoidal de 3V aplicada a las terminales (IN+ - IN-).

De estas tablas se puede obtener las características de los filtros elípticos de la unidad AIC, para posteriormente calcular el orden con ayuda de las ecuaciones anteriores y comprobar de esta manera el orden del filtro.

Tabla 5.1 Respuesta a la frecuencia del filtro pasabanda proporcionada por el fabricante.

FRECUENCIA (Hz)	MAGNITUD (dB)
f =100	-27
f =170	-5
3000 ≤ f ≤ 3400	-0.5
4000	-16
f ≥ 4600	-58

Tabla 5.2 Respuesta a la frecuencia del filtro pasabajas proporcionada por el fabricante.

FRECUENCIA (Hz)	MAGNITUD (dB)
f ≤ 3400	-0.5
f = 3600	-4
f = 4000	-30
f ≥ 4400	-58

De acuerdo a los datos proporcionados en la Tabla 5.2 podemos determinar las características del filtro pasabajas las cuales son:

$$K_P = 0.5 \text{ dB},\ K_s = 58 \text{ dB},\ f_p = 3400 \text{ Hz},\ f_s = 4400 \text{ Hz}.$$

Sustituyendo f_p y f_s en la Ecuación (5.1) se obtiene

$$\Omega = \frac{4400}{3400} = 1.294$$

Sustituyendo $K_{p,}$ K_S en la Ecuación (5.2) resultan en

$$\psi = \sqrt{\frac{10^{0.1*58}-1}{10^{0.1*0.5}-1}} = 2274$$

Sustituyendo ψ en la Ecuación (5.3) se obtiene el valor de C

$$C = \frac{1}{16\left(2274\right)^2}\left(1+\frac{1}{2\left(2274\right)^2}\right) = 1.2087*10^{-8}$$

Sustituyendo Ω en la Ecuación (5.4) se obtiene el valor de D

$$D = \frac{\sqrt{1.294}-1}{2\left(\sqrt{1.294}+1\right)} = 0.0322$$

Sustituyendo C y D en la Ecuación (5.5) se tiene,

$$F_E(C) = \frac{1}{\pi}\ln\left[1.2087*10^{-8}+2\left(1.2087*10^{-8}\right)^5+15\left(1.2087*10^{-8}\right)^9\right] = -5.8032$$

$$F_E(D) = \frac{1}{\pi}\ln\left[0.0322+2\left(0.0322\right)^5+15\left(0.0322\right)^9\right] = -1.0936$$

Finalmente sustituyendo $F_E(C)$ y $F_E(D)$ en la Ecuación (5.6) se obtiene el orden

$$n_E = (-5.8032)*(-1.0936) = 6.34$$

De acuerdo al resultado obtenido el orden del filtro es 7 ya que el resultado del calculo realizado fue 6.34 pero no es posible tener un filtro de orden fraccionario, debido a esto se debe de tomar el entero superior cercano. Además se comprueba que los datos proporcionados por el fabricante son correctos.

5.1.2 CALCULO DEL ORDEN DE UN FILTRO ELÍPTICO UTILIZANDO MATLAB

Existe una función en Matlab con la cual se puede determinar el orden de un filtro elíptico, la función que realiza esta operación es la función "**ellipord**" la cual se describe a continuación:

Sintaxis: $[\mathbf{n}, \boldsymbol{\omega}_\mathbf{n}] = \mathbf{ellipord}\ (\boldsymbol{\omega}_\mathbf{p}, \boldsymbol{\omega}_\mathbf{s}, \mathbf{K}_\mathbf{p}, \mathbf{K}_\mathbf{s}, \text{'}\mathbf{s}\text{'})$

donde:

n Variable en la cual se almacena el resultado correspondiente al orden del filtro.

$\boldsymbol{\omega}_\mathbf{n}$ Variable en la cual se almacena el resultado correspondiente a la frecuencia de corte del filtro elíptico, siempre toma el valor de ($\boldsymbol{\omega}_\mathbf{p}$).

$\boldsymbol{\omega}_\mathbf{s}$ Valor de la frecuencia de inicio de la banda de paro en (rad/seg).

$\boldsymbol{\omega}_\mathbf{p}$ Valor de la frecuencia final de la banda de paso en (rad/seg).

$\mathbf{K}_\mathbf{s}$ Magnitud mínima del rizo en la banda de paro en (dB).

$\mathbf{K}_\mathbf{p}$ Magnitud máxima del rizo en la banda de paso en (dB).

Como se puede observar los parámetros que necesita esta función son los mismos que se necesitan para calcular el orden del filtro con las ecuaciones anteriores, siendo la única diferencia que la frecuencia se debe expresar en (rad/seg).

Además de la función "**ellipord**" se cuenta en Matlab con otra función la cual permite calcular la función de transferencia del filtro.

Sintaxis: $[\mathbf{num},\mathbf{den}]=\mathbf{ellip}(\mathbf{n}, \mathbf{K}_\mathbf{p}, \mathbf{K}_\mathbf{S}, \boldsymbol{\omega}_\mathbf{n}, \text{'}\mathbf{s}\text{'})$

donde:

num Variable en la cual se almacenarán los coeficientes correspondientes al numerador de la función de transferencia del filtro elíptico.

den Variable en la cual se almacenarán los coeficientes correspondientes al denominador de la función de transferencia del filtro elíptico.

n Orden del filtro elíptico.

$\mathbf{K}_\mathbf{s}$ Magnitud mínima del rizo en la banda de paro en dB..

K_p Magnitud máxima del rizo en la banda de paso en dB.

ω_n Es la frecuencia de corte del filtro a una atenuación de K_p dB.

Con las dos funciones anteriores a partir de las características de un filtro elíptico se podría calcular la función de transferencia, Además, con la ayuda de la función "bode" y de la función "plot" se podría obtener la respuesta a la frecuencia de un filtro elíptico a partir de sus características.

Sintaxis: **[mag,fase,w] = bode(num,den)**

donde:

mag Es la variable en al cual quedará almacenada en forma de vector la magnitud de la respuesta a la frecuencia del filtro para diferentes valores de frecuencia

fase Es la variable en la cual quedará almacenada en forma de vector la fase para diferentes valores de frecuencia

w Es el vector donde quedan almacenados los valores de frecuencia en los cuales se evaluó la función de transferencia del filtro elíptico

A continuación se utilizarán las funciones descritas anteriormente y se mostrarán los resultados que entrega cada una de ellas. Los datos que se utilizarán serán los de la Tabla 5.2 con el fin de comparar el orden que se obtenga en Matlab con el calculado utilizando las ecuaciones de la sección anterior, así como también obtener la respuesta a la frecuencia de un filtro elíptico con las características siguientes:

$$\mathbf{K_P = 0.5\ dB\ ,\ K_S = 58\ dB,\ f_P = 3400\ Hz,\ f_s = 4400\ Hz}$$

Lo primero que se realizará será pasar la frecuencia de Hz a (rad/seg)

$$\omega_p = 2\pi(\ 3400\ \text{Hz}) = 21363\ \text{(rad/seg)}$$

$$\omega_s = 2\pi(\ 4400\ \text{Hz}) = 27646\ \text{(rad/seg)}$$

Una vez hecha la conversión de unidades se utiliza la función **"ellipord"**

$$[n, \omega_n] = \text{ellipord}\ (21363, 27646, 0.5, 58, 's')$$

El resultado que se obtiene al utilizar esta función es:

$$n=7$$

$$\omega_n = 21363$$

Como se mencionó anteriormente el valor de ω_n es el mismo que ω_p. Con lo que respecta al orden se obtuvo un valor de 7, el cual es del mismo orden que se obtuvo anteriormente mediante cálculos.

Una vez que se obtiene el valor de n y ω_n podemos utilizar la función **"ellip"** con el propósito de encontrar la función de transferencia.

$$[num,den]=ellip(7, 0.5, 58, 21363, 's')$$

Al ingresar los valores anteriores a la función **"ellip"** los coeficientes del numerador y el denominador tienen los siguientes valores.

$$num = [0\ 0.0000\ -0.0000\ 0.0000\ -0.0000\ 0.0000\ -0.0000\ 2.2966] * 1.0\ e{+}29$$

$$den = [0.0000\ 0.0000\ 0.0000\ 0.0000\ 0.0000\ 0.0000\ 0.0005\ 2.2966] * 1.0\ e{+}29$$

Una vez obtenida la función de transferencia se utiliza la función bode para obtener la respuesta a la frecuencia de un filtro elíptico obtenida con la ayuda de Matlab. La Figura 5.2 muestra la respuesta a la frecuencia de un filtro elíptico con rizado máximo en la banda de paso de K_P = 0.5 dB, rizado mínimo en la banda de paro de Ks = 58 dB, frecuencia de corte f_n = 3400 Hz y orden n=7. Este tipo de respuesta corresponde a un filtro con las características mencionadas anteriormente el cual según el fabricante es el comportamiento que tiene el filtro que se encuentra en la unidad AIC.

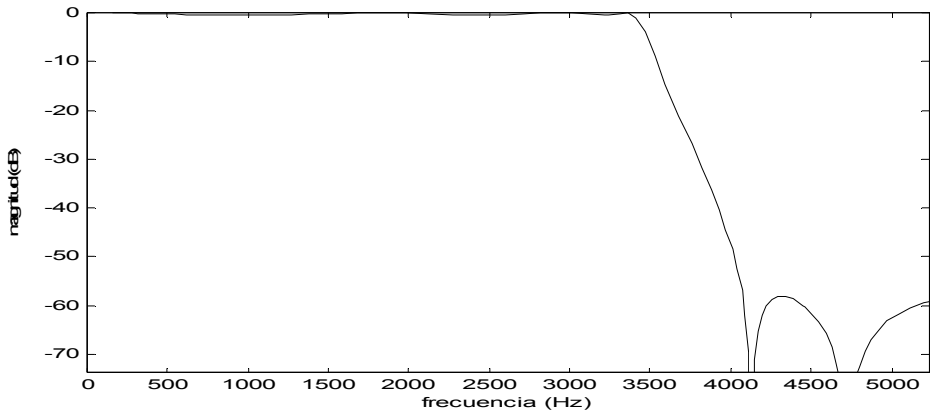


Figura 5.2 Respuesta a la frecuencia del filtro pasabajas elíptico utilizando Matlab.

5.1.3 RESPUESTA A LA FRECUENCIA DEL FILTRO PASABAJAS DE LA UNIDAD AIC MEDIANTE MEDICIONES

Para poder medir la respuesta a la frecuencia del filtro pasabajas de la unidad AIC se debe de contar con los componentes mostrados en el diagrama de bloques de la Figura 5.3. El generador de señales es el encargado de suministrar la señal de entrada a la unidad AIC. El osciloscopio se encarga de medir las señales de entrada y salida de la unidad AIC. A través de la PC se carga el programa que deberá ejecutar el DSP, siendo la tarjeta *DSP Starter Kit* el núcleo de esta aplicación. Cabe mencionar que todas las mediciones que se presenten a continuación estarán multiplicadas por un valor de dos ya que las puntas de medición del osciloscopio tienen este multiplicador.

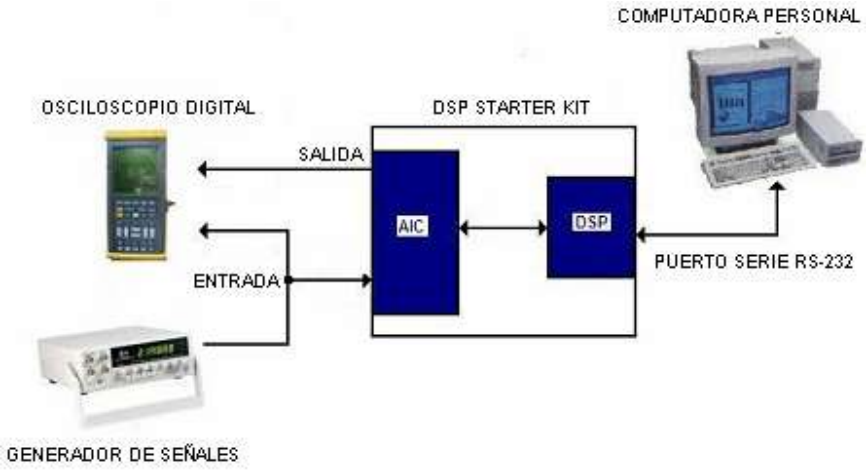


Figura 5.3 Elementos necesario para la obtención de la respuesta a la frecuencia en el caso de estudio 1.

Además de estos componentes es necesario el programa CAP3, que se detalla en el Apéndice B. El diagrama de bloques de la Figura 5.4 muestra la estructura de este programa la cual se puede resumir en un programa principal y dos rutinas de interrupción.

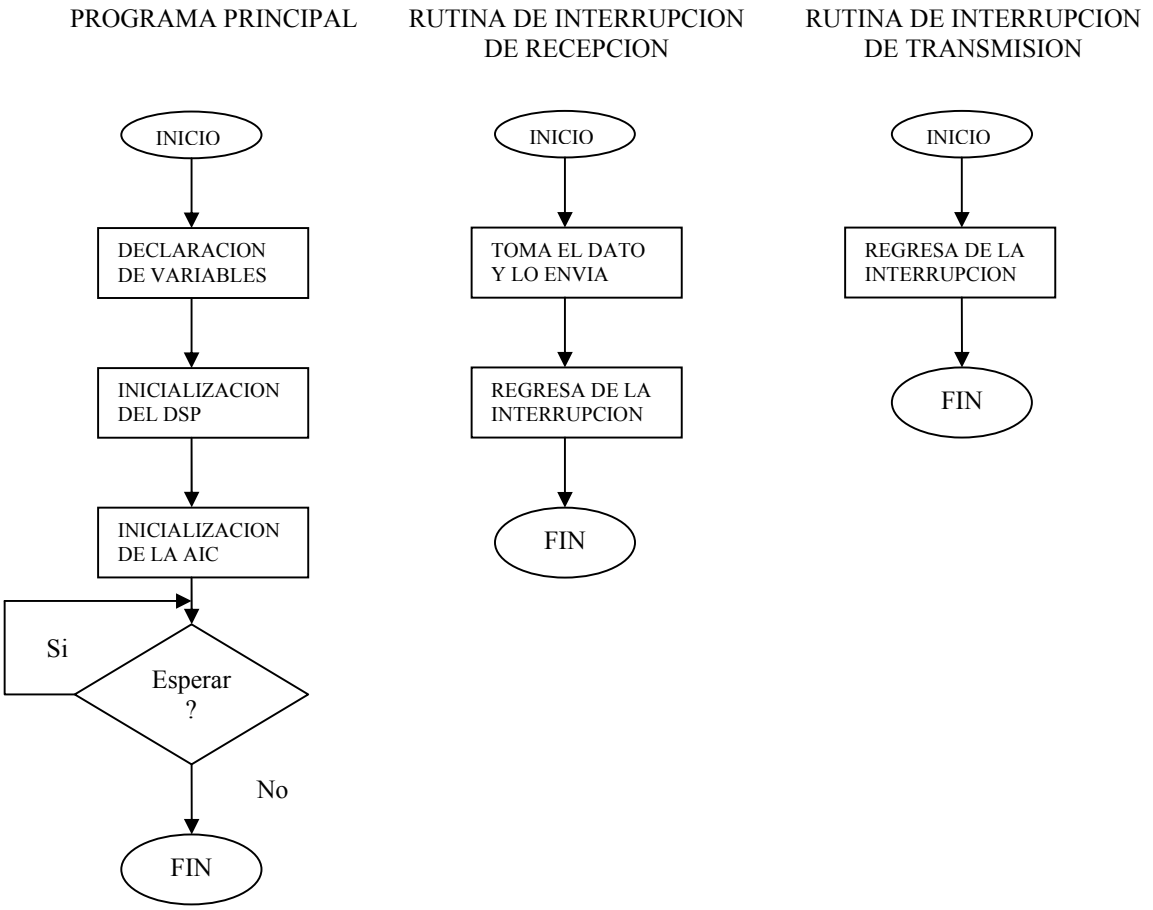


Figura 5.4 Diagrama de bloques del programa CAP3.

Para obtener la respuesta a la frecuencia del filtro pasabajas de la unidad AIC será necesario que el generador de señales suministre una señal senoidal de magnitud constante y con diferentes frecuencias (F_{in}). Con la ayuda de un osciloscopio digital se medirá simultáneamente la entrada (V_{in}) y la salida de la unidad AIC (V_{out}) y se tomará lectura de la magnitud obtenida a la salida de la unidad AIC para cada valor de frecuencia de la señal de entrada. En lo que respecta a la configuración de la unidad AIC con la ayuda del programa CAP3 se deshabilitara el filtro pasabanda esto con el fin de que no tenga ningún efecto sobre la señal de entrada, además se utilizarán los valores de configuración TA=18, TB=15 y

PRD=1, por lo tanto se obtendrá una frecuencia de corte F_H=3477.22 Hz y una frecuencia de muestreo F_m=18519 Hz. La Tabla 5.3 muestra los resultados obtenidos de estas mediciones.

Tabla 5.3 Resultados obtenidos durante las mediciones de la respuesta a la frecuencia del filtro pasabajas elíptico de la unidad AIC.

F_{in} (Hz)	V_{in} (Volts)	V_{out} (Volts)	V_{in}/V_{out} dB		F_{in} (Hz)	V_{in} (Volts)	V_{out} (Volts)	V_{in}/V_{out} dB
48.9	6.02	6	-0.0289		2526	6.07	5.65	-0.6228
101.6	6.02	6	-0.0289		2626	6.07	5.7	-0.5463
200.5	6.04	5.98	-0.0867		2733	6.06	5.79	-0.3959
300	6.03	5.96	-0.1014		2829	6.05	5.8	-0.3665
400	6.01	5.91	-0.1457		2914	6.05	5.77	-0.4116
504.2	6.01	5.88	-0.1899		3012	6.05	5.64	-0.6095
603.2	6.01	5.83	-0.2641		3141	6.05	5.49	-0.8437
705	6.01	5.75	-0.3841		3237	6.05	5.64	-0.6095
806.7	6.02	5.73	-0.4288		3334	6.05	5.47	-0.8754
905.9	6.03	5.73	-0.4433		3398	6.06	4.24	-3.1021
1006	6.04	5.74	-0.4425		3442	6.07	3.07	-5.921
1110	6.02	5.76	-0.3835		**3471**	**6.06**	**2.45**	**-7.8661**
1206	6.05	5.79	-0.3815		3482	6.05	2.19	-8.8262
1316	6.05	5.8	-0.3665		3558	6.06	1.17	-14.286
1420	6.06	5.82	-0.351		3594	6.05	0.874	-16.805
1523	6.06	5.9	-0.2324		3637	6.05	0.64	-19.512
1642	6.07	5.91	-0.232		3687	6.05	0.439	-22.786
1714	6.07	5.92	-0.2173		3730	6.05	0.319	-25.559
1829	6.06	5.88	-0.2619		3771	6.05	0.235	-28.214
1916	6.05	5.86	-0.2772		3813	6.04	0.18	-30.515
2015	6.06	5.81	-0.3659		3940	6.05	0.077	-37.905
2116	6.05	5.73	-0.472		4002	6.06	0.05	-41.67
2270	6.06	5.69	-0.5472		4114	6.06	0.022	-48.801
2357	6.06	5.63	-0.6393		4238	6.05	0.019	-50.06
2437	6.07	5.62	-0.669		4590	6.04	0.009	-56.536

En la Tabla 5.3 se reporta el resultado de un voltaje de salida $V_{out} = 2.45 = -7.8661$ dB a una frecuencia $F_{in} = 3471$ Hz. Este comportamiento del filtro en la frecuencia de corte es muy importante ya que se reflejará en todos los casos en que el filtro opere a esta frecuencia.

Al graficar las mediciones obtenidas se obtienen los resultados mostrados en la Figura 5.5. Si hacemos un acercamiento se puede determinar la frecuencia de corte del filtro elíptico de una manera aproximada (f_n) para posteriormente determinar ω_n (ver Figura 5.6).

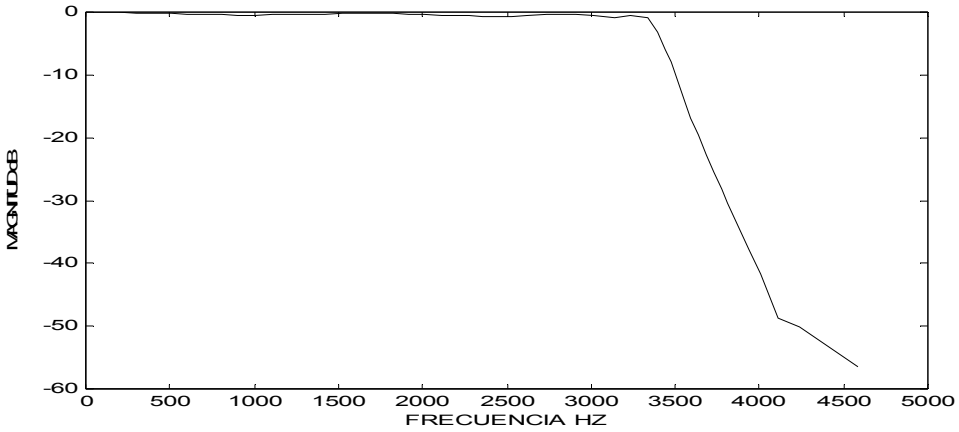


Figura 5.5 Respuesta a la frecuencia del filtro pasabajas elíptico obtenida de las mediciones.

Con el valor de $f_n = 3330$ Hz aproximadamente, $K_P = 0.5$ dB, $K_s = 58$ dB, n=7 y usando Matlab para obtener la respuesta a la frecuencia de un filtro con las características anteriores se obtiene el resultado de la Figura 5.7, en la cual se hace la comparación entre estos resultados y las mediciones

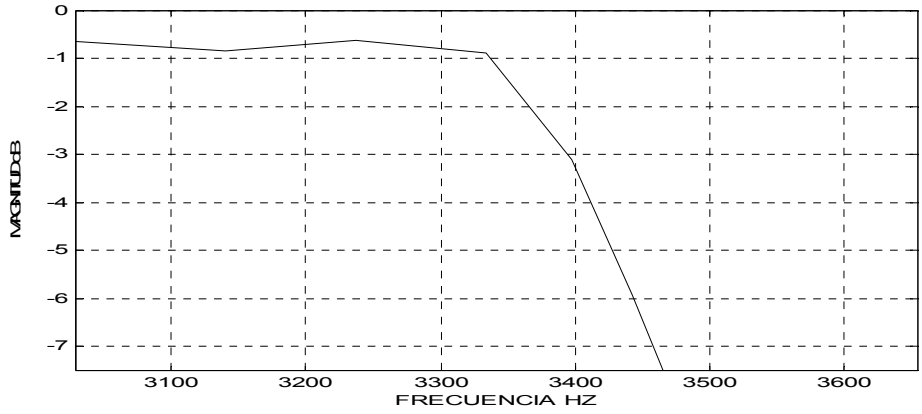


Figura 5.6 Acercamiento realizado a la Figura 5.5 para encontrar la frecuencia de corte.

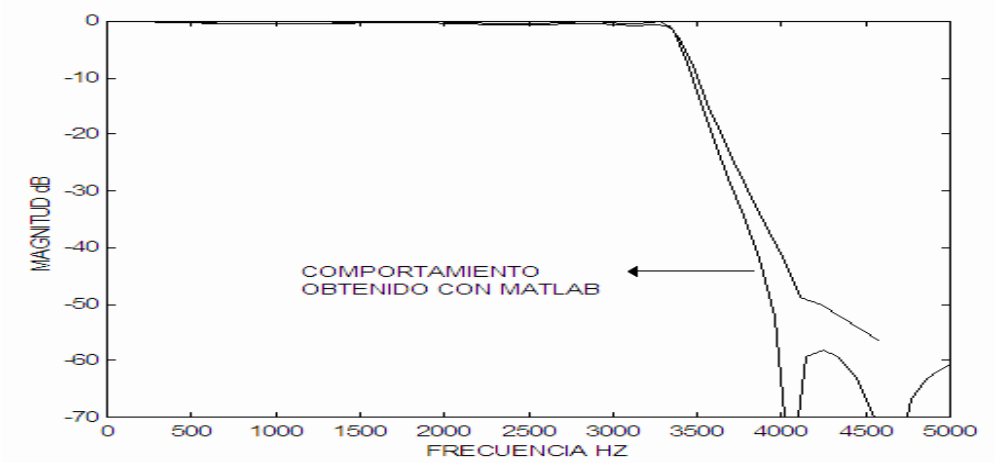


Figura 5.7 Comparación entre la respuesta obtenida en Matlab y las mediciones realizadas.

Si hacemos un acercamiento a la banda de paso podemos observar que existe un comportamiento muy parecido entre la respuesta obtenida con Matlab y las mediciones. Esto se puede observar en el acercamiento que se muestra en la Figura 5.8.

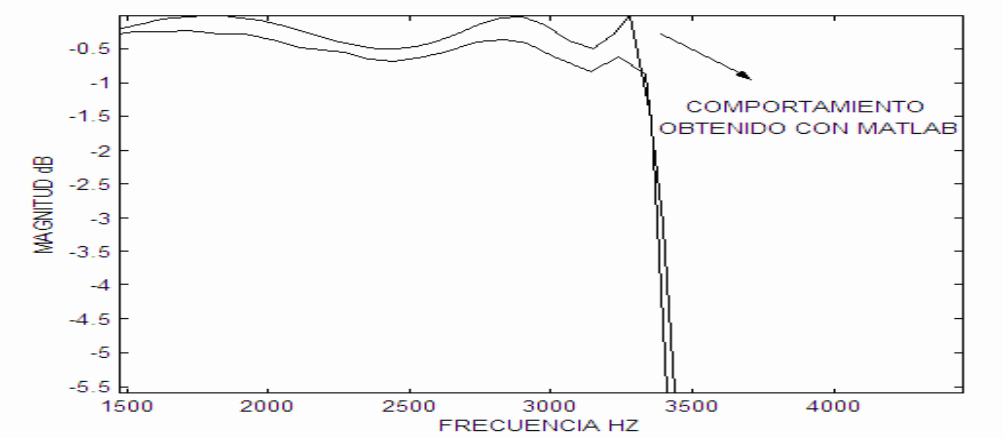


Figura 5.8 Comparación en la banda de paso entre la respuesta obtenida en Matlab y las mediciones.

Después de analizar los resultados de este caso de estudio se puede decir que los resultados obtenidos por simulación se asemejan con las mediciones ya que el comportamiento de rizado en la banda de paso es el característico al de un filtro elíptico de orden 7, además en la banda de transición el comportamiento por simulación es muy parecido al obtenido por mediciones. Por lo tanto, el filtro pasabajas efectivamente se comporta como un filtro elíptico de orden 7, rizado en la banda de paso de 0.5 dB y rizado en la banda de paro de 58 dB.

5.2 CASO DE ESTUDIO 2: EL FENÓMENO DE TRASLAPE (*ALIASING*)

Este caso de estudio tiene como objetivo principal determinar la presencia de traslape debido al no cumplimiento del criterio de Shannon. En este caso de estudio los parámetros más importantes son la frecuencia de muestreo (F_m) y las frecuencias de corte del filtro pasabanda de la unidad AIC (F_H y F_L), en especial la frecuencia de corte F_H.

5.2.1 ESTUDIO DEL FENÓMENO DE TRASLAPE EN EL DSP

Debido a la facilidad con la que se calculan las frecuencias de corte del filtro pasabanda de la unidad AIC y la frecuencia de muestreo, es posible analizar el criterio de Shannon con ayuda del DSK. Para esta aplicación también se utilizarán los mismos componentes que los usados en el caso de estudio 1 (Figura 5.3). El programa que se utilizará es el mismo CAP3. En este caso de estudio el filtro pasabanda estará habilitado.

La forma de determinar la presencia del traslape en este caso de estudio será realizando dos ejemplos. En el primero se configurará la unidad AIC de tal manera que se cumpla con el criterio de Shannon mientras que en el segundo ejemplo se configurará la unidad AIC de tal manera que no se cumpla con el criterio de Shannon. Una vez que se realizó la configuración de la unidad AIC se aplicarán señales senoidales de diferentes frecuencias a la entrada de la unidad AIC con la ayuda de un generador de señales, al mismo tiempo con la ayuda de un osciloscopio digital obtendremos diferentes mediciones en las cuales se reflejará el efecto que sufre la salida con respecto a la señal de entrada en su paso por el DSP y la unidad AIC.

Ejemplo1: El filtro pasabanda se configurará con las frecuencias de corte F_H = 2500 Hz, F_L = 208.3 Hz y una frecuencia de muestreo F_m = 8 KhZ. Por lo tanto los valores que tomaran los registros de configuración son TA=25, TB=25 y PRD=1. Estos valores de configuración aseguran que no debe de presentarse el fenómeno de traslape debido a que se cumple con el criterio de Shannon.

La primera medición que se realizó fue a un valor de 200Hz la cual corresponde aproximadamente a la frecuencia de corte calculada F_L. Esta medición se muestra en la

Figura 5.9 en donde el canal A del osciloscopio muestra la entrada y el canal B muestra la salida del DSK. Además en esta misma figura se señalan las partes correspondientes al canal A y al canal B respectivamente.

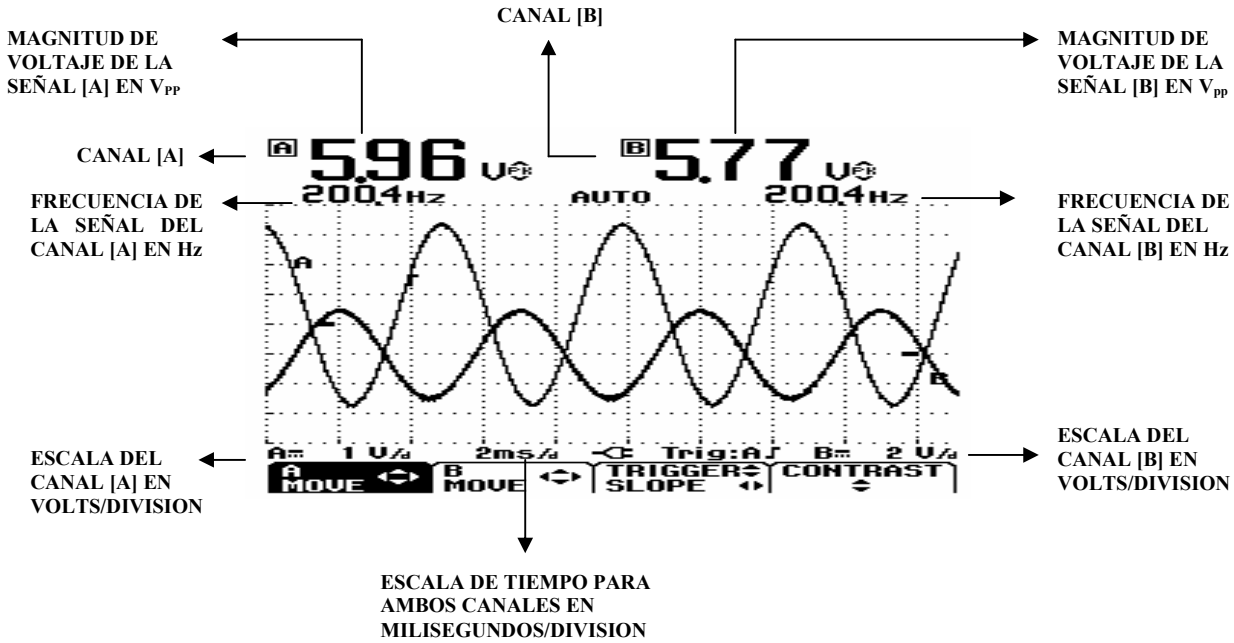


Figura 5.9 Medición realizada a la frecuencia de corte F_L = 200 Hz en el ejemplo 1.

La segunda medición realizada es a 1500Hz, este valor se encuentra comprendido dentro de la banda de paso del filtro. La Figura 5.10 muestra estos resultados en donde se puede observar que el filtro efectivamente permite el paso de la señal.

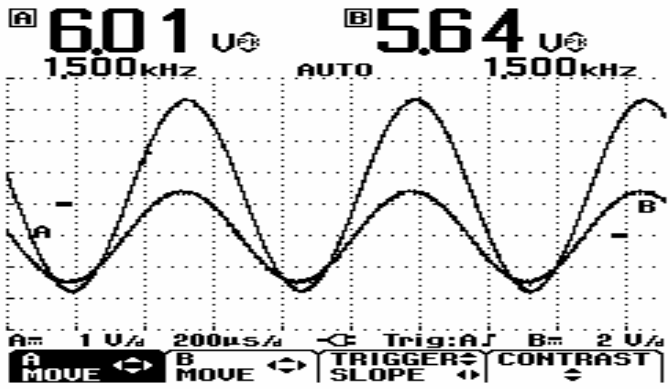


Figura 5.10 Medición realizada a 1500Hz en el ejemplo 1.

La tercera medición realizada fue a un valor de 2500 Hz la cual corresponde a la frecuencia de corte calculada F_H. La Figura 5.11 describe el comportamiento de la señal de entrada A y la señal de salida B en la cual se puede ver que las dos señales tienen la misma frecuencia pero en cambio la magnitud de la señal de salida se encuentra atenuada debido a que la frecuencia de la señal de entrada es atenuada por el filtro pasabanda de la unidad AIC.

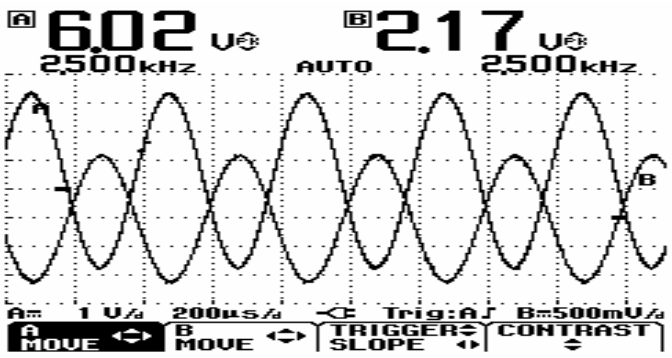


Figura 5.11 Medición realizada a la frecuencia de corte F_H = 2500 Hz en el ejemplo 1.

Se puede observar en las tres mediciones anteriores que no existe la presencia del traslape ya que la frecuencia de la señal de entrada es la misma que la señal de salida. Al seguir aumentando la frecuencia de la señal de entrada se espera que la salida se haga prácticamente cero, lo cual se aprecia en la Figura 5.12 en donde la frecuencia de la señal de entrada es 2.772 Khz, con V_{PP} = 6V y la salida tiene una frecuencia de 2.725 Khz y 12.8 mV de pico a pico. Las pequeñas discrepancias entre la frecuencia de entrada y salida no se debe al traslape sino a la medición que realiza el osciloscopio de un valor de voltaje muy pequeño en la señal de salida.

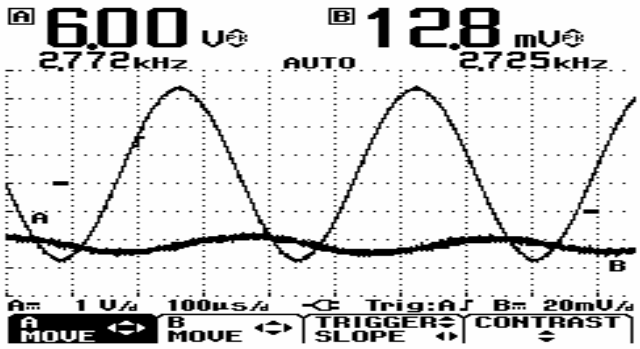


Figura 5.12 Medición realizada a 2725 Hz en el ejemplo1.

Por lo tanto, con los valores de frecuencia de corte y frecuencia de muestreo calculados se cumple con el criterio de Shannon. Para este caso el filtro pasabanda deja pasar las frecuencias dentro del ancho de banda delimitado por F_H y F_L.

Ejemplo2: Para este ejemplo los valores de configuración elegidos para este son TA=25, TB=50 y PRD = 1 lo que da por resultado F_H = 2500Hz, F_L = 208.3 Hz y F_m = 4Khz. Se puede ver que con estos valores calculados ya no se cumple con el criterio de Shannon y se espera la presencia del traslape en el rango [F_m – F_H, F_H] lo cual se verificará realizando mediciones a diferentes frecuencias de la señal de entrada.

La primera medición realizada es a 200 Hz, cuyo valor corresponde a la frecuencia de corte calculada F_L aproximadamente. El resultado de esta medición se muestra en la Figura 5.13 en la cual se aprecia que efectivamente se esta realizando un muestreo correcto.

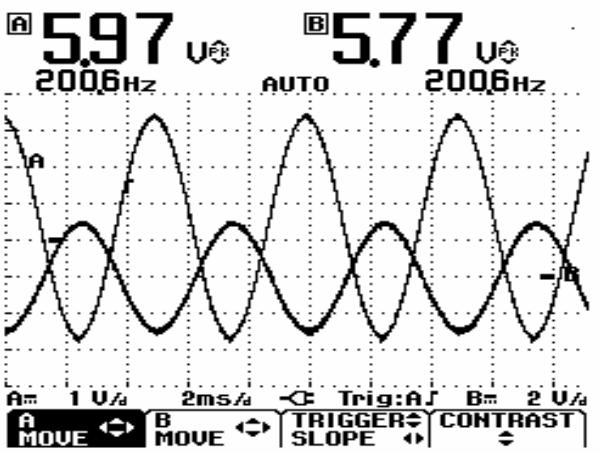


Figura 5.13 Medición realizada a la frecuencia de corte F_L = 200 Hz en el ejemplo 2.

La segunda medición realizada es en el limite en el que iniciará el traslape con una señal de entrada con frecuencia 1500Hz y V_{pp} = 6.01 V, ver Figura 5.14. En este valor de frecuencia se espera el traslape debido a que la frecuencia de muestreo es 4 Khz y la frecuencia de corte es 2500 Hz, como el criterio de Shannon no se cumple entonces el traslape se presentara en el rango [4Khz - 2.5 Khz, 2.5 Khz].

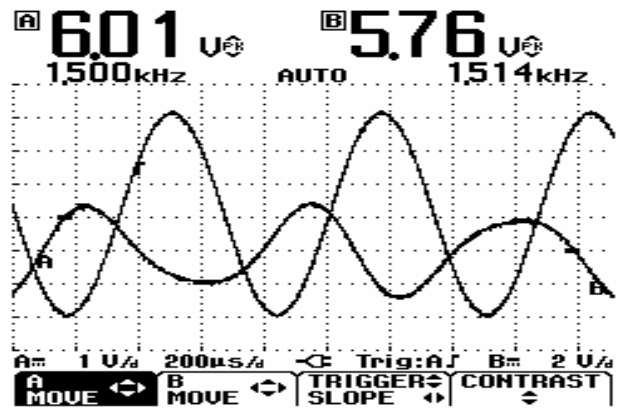


Figura 5.14 Medición realizada a 1500Hz en el ejemplo 2.

Se puede observar que a 1500Hz ya existe la presencia del traslape debido a que la frecuencia de la señal de entrada no es la misma que la señal de salida. Además, cabe hacer mención que la señal de salida no permanece estable en el osciloscopio en estas condiciones. Debido a lo anterior, se realizó una tercera medición que corresponde al valor de frecuencia límite a la cual todavía no existe el traslape, la cual se muestra en la Figura 5.15.

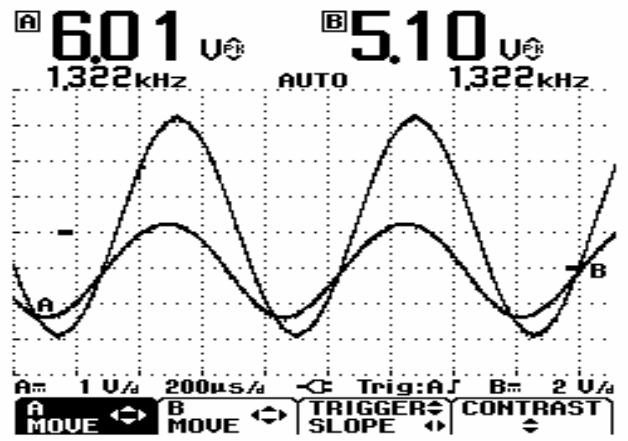


Figura 5.15 Frecuencia limite a la cual no existe traslape en el ejemplo2.

La Figura 5.16 muestra la cuarta medición realizada a una frecuencia de 2500 Hz, la cual corresponde precisamente a la frecuencia de corte calculada $F_{H.}$ En esta figura es claramente apreciable que la frecuencia de la señal de salida no corresponde con la señal de entrada por lo tanto existe la presencia del traslape.

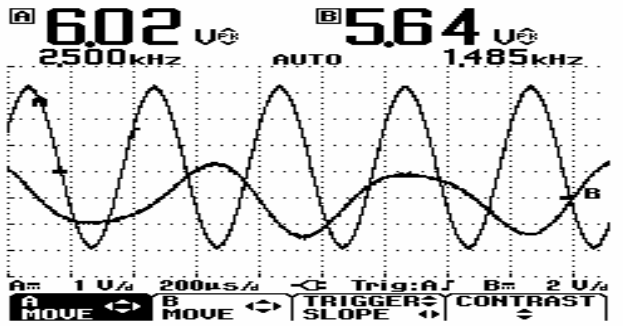


Figura 5.16 Medición realizada a la frecuencia de corte F_H = 2500 Hz en el ejemplo 2.

Como se puede observar de las mediciones anteriores no se cumplió con el criterio de Shannon y por lo tanto apareció el traslape. Además, se detectó la presencia del fenómeno del traslape antes del valor calculado ya que como se vio en el ejemplo anterior, existen pequeñas diferencias entre los valores medidos y calculados de la frecuencia de corte F_H y, por lo tanto, cambia el rango $[\mathbf{F_m - F_H},\ \mathbf{F_H}]$ ya que depende de los valores de F_m y F_H.

Adicionalmente, se realizaron varias mediciones para verificar si la frecuencia de corte (F_H) y la frecuencia de muestreo (F_m) calculadas son iguales a F_H y F_m medidos. A continuación se muestra únicamente el resultado obtenido con TA=20, TB=30 y PRD=4, ya que el resultado obtenido en los demás casos es similar. Con los valores de configuración anteriores se obtiene una frecuencia de corte calculada F_H = 1250 Hz y una frecuencia de muestreo F_m = 3333 Hz. Al realizar la medición a la frecuencia de corte F_H se obtuvo el resultado de la Figura 5.17, mientras que la frecuencia de muestreo medida se muestra en la Figura 5.18.

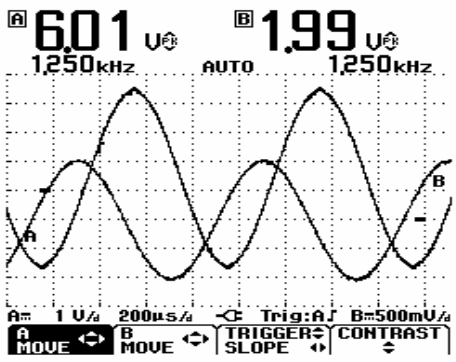


Figura 5.17 Medición obtenida a la frecuencia de corte F_H = 1250 Hz.

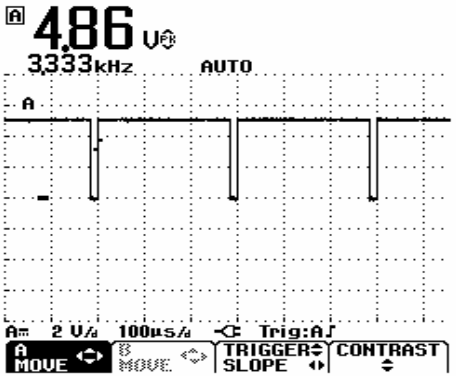


Figura 5.18 Medición correspondiente a la frecuencia de muestreo F_m = 3333 Hz.

De acuerdo a los resultados de las mediciones podemos ver que la frecuencia de corte calculada en ninguno de los casos atenuó la señal a un valor cercano a cero. En cambio el valor de voltaje al que atenúa la señal esta en un rango entre 2.51 volts a 1.99 volts. Estos valores de atenuación a la frecuencia de corte son los esperados debido al tipo y orden del filtro que esta implementado en la unidad AIC, cuyas características ya se describieron a detalle en las Tablas 5.3 y Figura 5.5. En lo que respecta a la frecuencia de muestreo el valor calculado es igual al valor medido en varios de los casos mientras que en algunos la diferencia es de un valor despreciable.

5.3 CASO DE ESTUDIO 3: FILTRADO DE UNA SEÑAL CON UNA COMPONENTE ARMONICA

La finalidad de este caso de estudio consiste en configurar la unidad AIC de tal manera que el filtro pasabanda de entrada funcione como un filtro que deje pasar una señal de 60 Hz y elimine todas aquellas frecuencias mayores. Uno de los objetivos principales de este caso de estudio es explorar los límites de la unidad AIC que permitan implementar una aplicación muy específica como la que se plantea para este caso.

Hasta ahora se han realizado mediciones con señales que contienen únicamente una frecuencia pero a continuación se muestra un ejemplo en el cual se suma una señal de 60Hz con una señal de 300Hz a través de amplificadores operacionales. Para este caso de estudio se utiliza un generador adicional y un circuito sumador (ver Figura 5.19). El programa utilizado en este caso de estudio y la configuración del registro de control de la unidad AIC son los mismos que en el caso de estudio 2.

Los valores de configuración son TA = 31, TB = 14 y PRD = 57, con lo que se obtienen un valor de frecuencias de corte calculados F_H = 70 Hz, F_L = 5.83 Hz, una frecuencia de muestreo F_m = 397.26 Hz y un MSTR CLK = 344827 Hz.

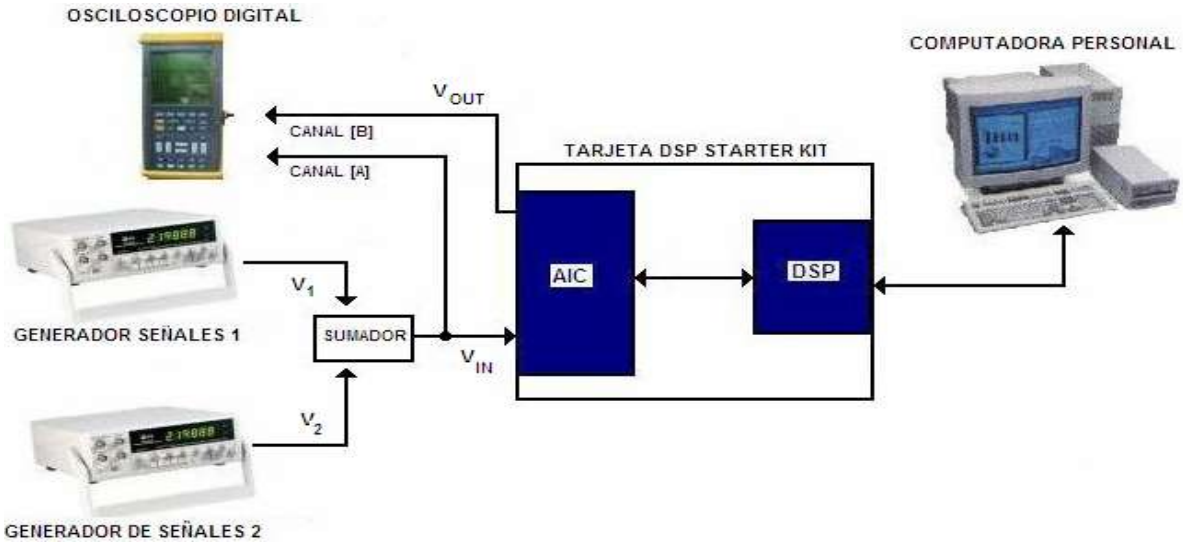


Figura 5.19 Elementos necesario para llevar a cabo el caso de estudio 3.

Con los valores calculados de frecuencia de corte y frecuencia de muestreo se elimina el problema de traslape ya que se esta cumpliendo con el criterio de Shannon. Por otra parte, al calcular la frecuencia de corte en 70Hz, la frecuencia de la señal de 60Hz se encuentra dentro de la banda de paso del filtro pasabanda y la señal de 300 Hz se elimina obteniéndose a la salida únicamente la señal de 60Hz.

Las Figuras 5.20 y 5.21 muestran las señales de entrada medidas con el osciloscopio, mientras que la señal que se obtiene al sumar estas dos señales se muestra en la Figura 5.22.

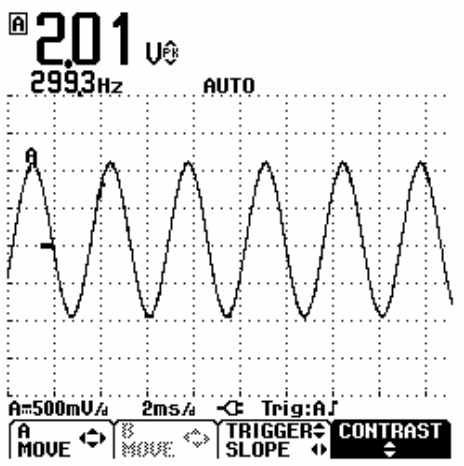


Figura 5.20 Señal de entrada V_1.

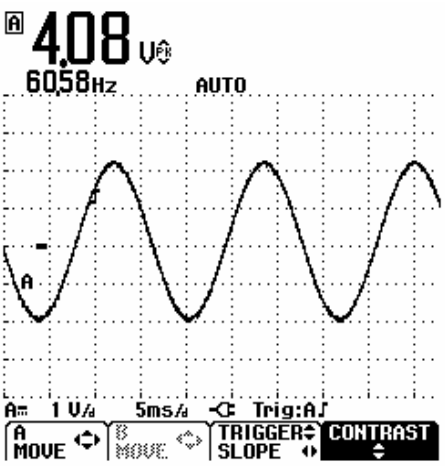


Figura 5.21 Señal de entrada V_2.

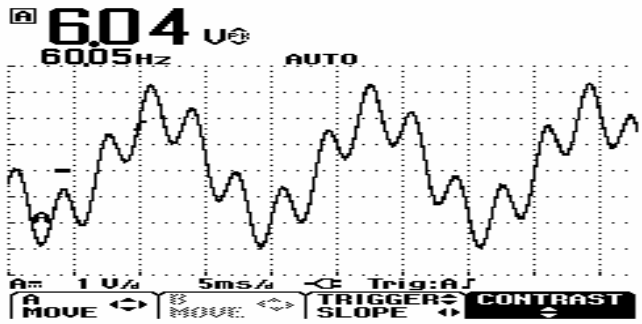


Figura 5.22 Señal sumada $V_{IN.}$

Esta señal obtenida a la salida del sumador se aplica a la unidad AIC, la cual ya esta configurada con los valores anteriormente mencionados y se espera que elimine la señal V_1 permitiendo obtener nuevamente la señal V_2. El resultado obtenido a la salida de la unidad AIC se muestra en la Figura 5.23.

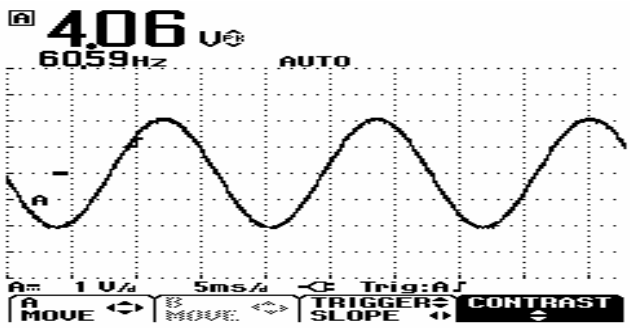


Figura 5.23 Señal obtenida a la salida de la unidad AIC $V_{OUT.}$

Si comparamos la Figura 5.23 con la figura 5.21 podemos determinar que la señal V_1 fue eliminada de manera adecuada, además que se pudo recuperar la señal V_2 de manera correcta. Se puede observar que tanto la magnitud como la frecuencia de la señal obtenida a la salida de la unidad AIC son prácticamente iguales a la señal de entrada V_2.

5.4 CASO DE ESTUDIO 4: MUESTREO Y ALMACENAMIENTO DE UNA SEÑAL DE ENTRADA

Hasta ahora lo que se ha hecho en los ejemplos anteriores es muestrear la señal de entrada y cada punto que se convierte de A/D es nuevamente enviado a la salida del DSP. Una

aplicación diferente que se le puede dar al DSP es tomar N muestras de dicha señal muestreada y almacenarlas en localidades de memoria del mismo DSP para su posterior análisis o tratamiento.

5.4.1 INTRODUCCION

El programa implementado para llevar a cabo este caso de estudio se describe por medio del diagrama de bloques de la Figura 5.24, el cual corresponde al programa CAP4 que se encuentra en el Apéndice C.

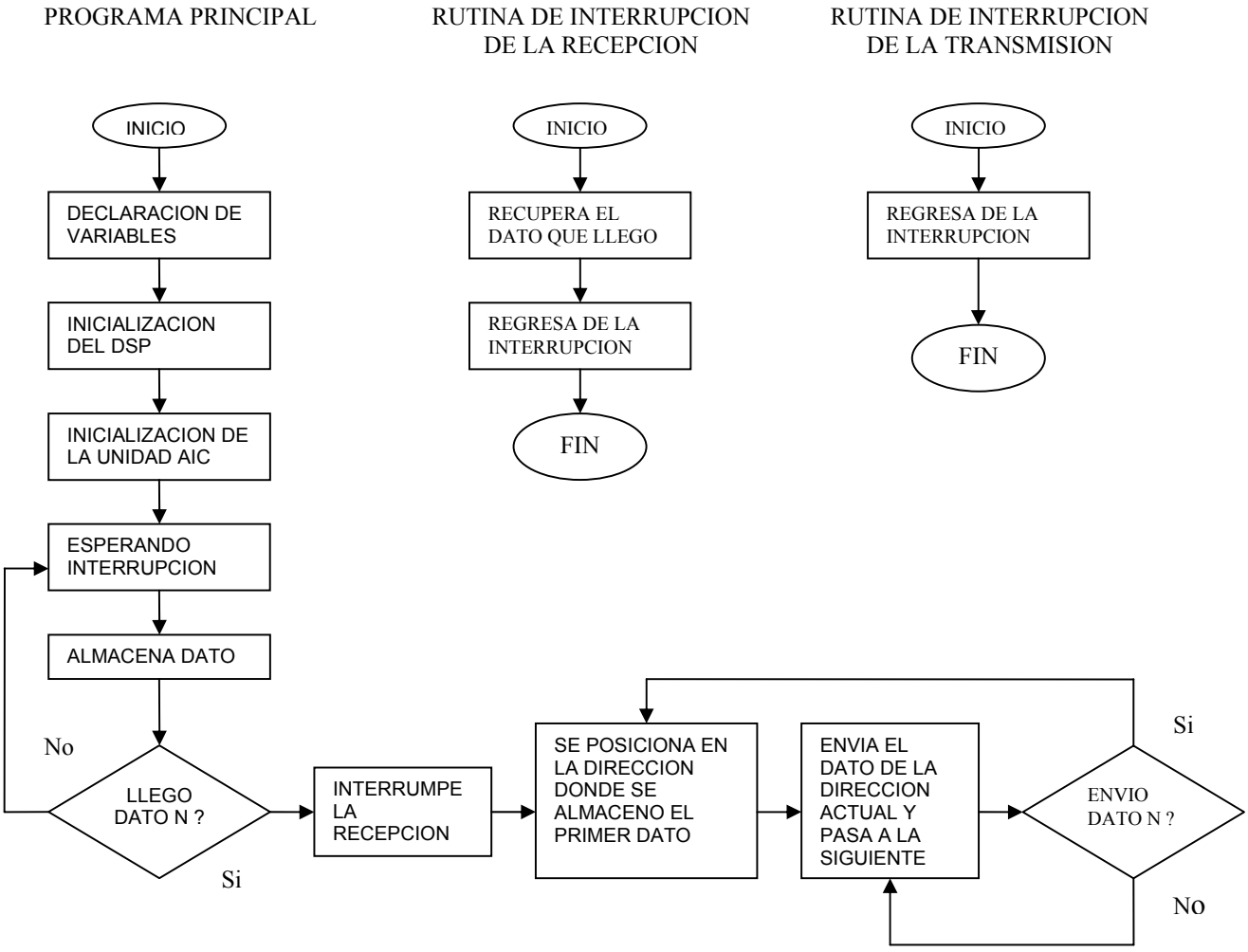


Figura 5.24 Diagrama de bloques del programa CAP4.

De acuerdo al diagrama de bloques el funcionamiento del programa CAP4 es el siguiente:

1. Declaración de variables utilizadas en el programa.

2. Inicialización del DSP (Puerto serie, Timer, Interrupciones, etc).

3. Inicialización de la unidad AIC (Los registros TA, TB y el registro de control de la unidad AIC).

4. El programa espera a que llegue un dato a través del puerto serie, posteriormente se genera una interrupción la cual lleva al programa a la rutina de interrupción correspondiente a la recepción.

5. Una vez que se encuentra en la rutina de interrupción se obtiene el dato que llegó y se regresa al programa principal donde se almacena en una localidad de memoria.

6. Después de almacenar el dato se verifica si el dato que llegó corresponde al dato N. Si esto se cumple se interrumpe la recepción, de no ser así el programa espera una nueva interrupción.

7. Una vez interrumpida la recepción se envían los datos almacenados en la memoria en el mismo orden como fueron llegando. Al enviar el último dato almacenado se vuelve a repetir el mismo proceso y de esa manera permanece el programa permanentemente.

De acuerdo a CAP4 la velocidad con la que se toman los datos dependen de la frecuencia de muestreo seleccionada. Así, si se quiere tomar un ciclo completo de una señal con un numero N de muestras se debe de tener cuidado con la elección de la frecuencia de muestreo.

Ejemplo: Supóngase que una señal de entrada con frecuencia igual a 100 Hz, la cual se suministra al DSP y la frecuencia de muestreo con la que se configuró la unidad AIC es 10 Khz. ¿Cuántas muestras deben tomarse para poder reproducir un ciclo completo de la señal de entrada? Para determinar el número de muestras se utiliza la siguiente expresión:

$$N = \frac{F_m}{F_{in}} \tag{5.7}$$

donde:

F_m Frecuencia de muestreo de la unidad AIC en (Hz)

F_{in} Frecuencia de la señal de entrada a la unidad AIC en (Hz)

N Numero de muestras.

Sustituyendo datos en la ecuación (5.7)

$$N = \frac{10000Hz}{100Hz} = 100$$

Entonces de acuerdo al resultado obtenido si se configura la unidad AIC de tal manera que se obtenga una frecuencia de muestreo de 10 Khz y se toman 100 muestras y la señal de entrada es de 100Hz, al utilizar el programa CAP4 la señal de salida del DSP será una señal periódica con la misma frecuencia que la señal que se suministró a la entrada del DSP (ver Figura 5.25).

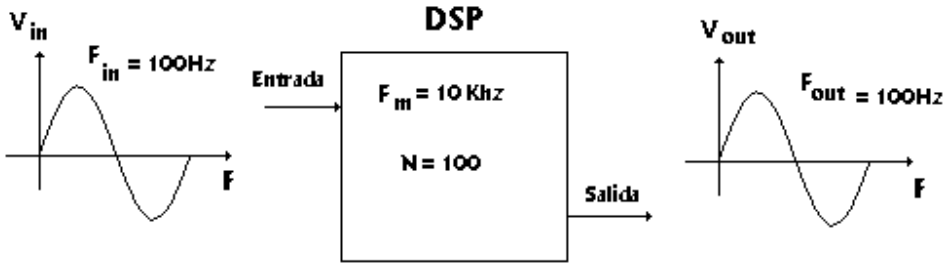


Figura 5.25 Muestreo de un ciclo completo de una señal senoidal.

Pero si ahora se cambia el numero de muestras a N = 50, mientras que la frecuencia de muestreo se mantiene igual, la señal que se obtiene a la salida no será la misma que la señal de entrada ya que no alcanzará a muestrear el periodo de una señal de 100 Hz. En este caso solamente tomará la mitad de un periodo, como se puede observar en la Figura 5.26.

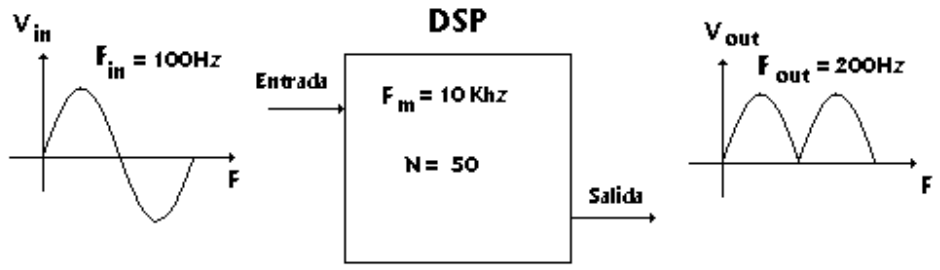


Figura 5.26 Muestreo de medio ciclo de una señal senoidal.

Como se puede ver en los dos ejemplos anteriores si conocemos la frecuencia de la señal de entrada podemos calcular una frecuencia de muestreo que satisfaga la condición de muestrear completamente un ciclo de la señal con N puntos.

5.4.2 METODO PARA CALCULAR TA, TB Y PRD A PARTIR DE LA FRECUENCIA DE MUESTREO

Las ecuaciones con las cuales se calcula el valor de la frecuencia de muestreo son las ecuaciones (3.1) y (4.6), las cuales ya se mostraron anteriormente en el capítulo 3 y 4 respectivamente. En el caso de la Ecuación 3.1 se cambia TOUT por MSTR CLK debido a que el DSP suministra la señal del MSTR CLK a través de la terminal TOUT como se muestra en la Figura 3.12, además se suprimió el término $(TDDR+1)$ debido a que el mínimo MSTR CLK permitido por la unidad AIC es de 75 Khz el cual se puede lograr únicamente con el registro PRD. Además el valor de $t_c(c)$ obtenido en el manual del fabricante es 50×10^{-9}, por lo que la ecuación se puede reescribir de la forma siguiente,

$$MSTR \cdot CLK = \frac{1}{50\times10^{-9}(PRD+1)} \tag{5.8}$$

Despejando el término $TA \times TB$ de la ecuación (4.6) se tiene,

$$TA \times TB = \frac{MSTR \cdot CLK}{2 F_m} \tag{5.9}$$

Para calcular la frecuencia de muestreo es necesario conocer los valores de TA, TB y PRD con los cuales se obtiene la frecuencia de muestreo deseada, pero eso implica realizar cálculos de manera repetitiva a fin de ver cuales son los valores de configuración con los cuales se obtiene la frecuencia de muestreo más exacta. Por lo tanto, se propone el siguiente método, que se resume en los siguientes puntos:

1. Se utilizan valores de PRD desde 1 hasta el valor que queramos calcular, la limitante es 256 ya que con ese valor se obtiene el MSTR CLK más pequeño permitido.

2. Por cada valor de MSTR CLK que se evalúa en la Ecuación 5.9 se obtiene el valor del producto $TA \times TB$. Si este producto se encuentra dentro del rango $8 \leq TA \times TB \leq 1953$ se almacena el valor de PRD así como el valor obtenido de $TA \times TB$, de no ser así se toma un nuevo valor de PRD y por lo tanto se obtendrá un nuevo MSTR CLK y se vuelve a evaluar en la Ecuación 5.9. Cabe mencionar que el rango $8 \leq TA \times TB \leq 1953$ se debe a los valores que puede tomar TA y TB, los cuales son $4 \leq \text{TA} \leq 31$ y $2 \leq \text{TB} \leq 63$.

3. Es posible que durante el proceso de cálculo se obtengan valores de $TA \times TB$ los cuales no son enteros, pero para configurar los registros TA y TB es necesario configurarlos con un valor entero. En este caso se deben obtener los valores que más se acerquen al entero superior o inferior más cercano (con una diferencia mínima previamente especificada).

Ejemplo: Supóngase que se desea tener una frecuencia de muestreo de F_m = 7500 Hz, y se requiere saber cuales son los valores de PRD y $TA \times TB$ para obtener un valor de 7500 Hz o

cercano. También considérese que la diferencia entre el valor calculado y el entero superior o inferior más cercano debe de ser menor a 0.2.

En este ejemplo solo se utilizarán valores de PRD en el rango de 1 a 10 para simplificar el número de operaciones.

Los valores obtenidos se resumen en la Tabla 5.4, en donde se puede observar que el término $TA \times TB$ en todos los casos se encuentra dentro del rango. Además, en todos los valores calculados el resultado es un número con decimales y como ya se mencionó anteriormente los valores que pueden tomar TA y TB son enteros. Por lo tanto se tendrá que determinar cuales de los datos anteriores se encuentra a una diferencia menor de 0.2, ya sea a su entero superior o inferior más cercano.

Tabla 5.4 Resultados obtenidos a diferentes valores de MSTR CLK.

PRD	1	2	3	4	5	6	7	8	9	10
MCLK (Mhz)	10	6.66	5	4	3.33	2.85	2.5	2.22	2	1.81
$TA \times TB$	666.67	444.44	333.33	266.67	222.22	190.48	166.67	148.15	133.33	121.21

También se puede observar en la Tabla 5.4 que únicamente un valor de los diez que se tiene en la tabla cumplió con la condición. PRD = 8 y $TA \times TB = 148$.

Hasta ahora se ha determinado el valor de PRD pero falta encontrar el valor de TA y TB por separado. Ahora bien, es necesario tomar en cuenta todas las combinaciones entre TA y TB, lo cual puede ser un poco tardado ya que existen un gran número de combinaciones.

Para ejemplificar la forma en que debe realizarse la búsqueda supondremos que TA toma valores del número 4 al 10 y TB toma valores del número 2 al 10. Todas las posibles multiplicaciones entre TA y TB se muestran en la Tabla 5.5.

Tabla 5.5 Posibles combinaciones entre TA y TB.

TB \ TA	4	5	6	7	8	9	10
2	8	10	12	14	16	18	20
3	12	15	18	21	24	27	30
4	16	20	24	28	32	36	40
5	20	25	30	35	40	45	50
6	24	30	36	42	48	54	60
7	28	35	42	49	56	63	70
8	32	40	48	56	64	72	80
9	36	45	54	63	72	81	90
10	40	50	60	70	80	90	100

Se puede apreciar en esta tabla que el área sombreada forma una matriz simétrica, es decir, es lo mismo $TA \times TB$ que $TB \times TA$ por lo que se podría eliminar ya sea la parte superior o la parte inferior de la diagonal principal de esta área, quedando como lo muestra la Tabla 5.6.

Tabla 5.6 Eliminación de elementos en el arreglo de posibles combinaciones de TA y TB.

TB \ TA	4	5	6	7	8	9	10
2	8	10	12	14	16	18	20
3	12	15	18	21	24	27	30
4	16	20	24	28	32	36	40
5		25	30	35	40	45	50
6			36	42	48	54	60
7				49	56	63	70
8					64	72	80
9						81	90
10							100

De la Tabla 5.6 se desprenden las siguientes consideraciones de búsqueda:

1. El primer elemento de cada renglón corresponde al valor más pequeño contenido en ese renglón, por lo que si se quiere buscar un valor X tal que al compararlo con el primer elemento de cada renglón resulta que el numero X es menor no tiene caso seguir la búsqueda en los demás renglones ya que los demás valores que contiene la tabla son mayores.

2. Si se esta buscando en un renglón un número X el cual es mayor que el primer elemento del renglón y el elemento en la columna siguiente es mayor al numero X no tiene caso seguir la búsqueda en las siguientes columnas ya que los siguientes elementos serán mayores por lo que se tendrá que pasar al siguiente renglón.

3. La búsqueda se realizará desde la primera columna en los tres primeros renglones mientras que a partir del cuarto renglón se comenzará a buscar a partir de los elementos que forman la diagonal principal.

4. Se pueden obtener diferentes valores de TA y TB con los cuales se cumpla el producto $TA \times TB$, por lo que la búsqueda no debe parar hasta que se cumpla lo explicado en el punto1.

5. Este mismo procedimiento se utiliza para los rangos originales de TA (4 a 31) y TB (2 a 63) ya que se tiene el mismo comportamiento.

Ejemplo: Suponiendo que $TA \times TB$ =30, ¿como debe realizarse la búsqueda de acuerdo a los puntos explicados anteriormente y cuales son las soluciones encontradas?

Tabla 5.7 Resultado de la búsqueda de las posibles soluciones de TA y TB.

TB \ TA	4	5	6	7	8	9	10
2	8	10	12	14	16	18	20
3	12	15	18	21	24	27	30
4	16	20	24	28	32	36	40
5		25	30	35	40	45	50
6			36	42	48	54	60
7				49	56	63	70
8					64	72	80
9						81	90
10							100

La búsqueda se realizará en el área sombreada, y las soluciones encontradas son dos, las cuales se encuentran resaltadas. De esta manera se puede determinar que para encontrar un dato no es necesario revisar toda la tabla, basta con seguir las condiciones dadas en los puntos anteriores. Pero aun así esta búsqueda es demasiado tediosa ya que los rangos de TA y TB son más grandes a los utilizados para ejemplificar la forma de la búsqueda por lo que lo mejor es realizar un programa el cual realice esta búsqueda.

Para facilitar los cálculos se implementaron dos programas en Matlab para determinar: *i*).- Los valores de PRD y el producto $TA \times TB$ que más se acerquen a un valor entero a partir de la frecuencia de muestreo, su código se encuentra en el Apéndice D y, *ii*).- A partir de $TA \times TB$ encontrar los valores de TA y TB correspondientes y su código se encuentra en el apéndice E.

A continuación se presentan 2 ejemplos en los cuales se utilizaron los programas MAT1 el cual corresponde al programa del Apéndice D y MAT2 el cual corresponde al programa del Apéndice E. Con la ayuda de estos programas se encontrarán los valores de configuración.

Ejemplo1: Se desean tomar 65 muestras de una señal de 60 Hz, determinar cual debe de ser la frecuencia de muestreo necesaria y una vez obtenida cuales son los valores de configuración. Utilizando la Ecuación 5.7 y despejando F_m, se tiene.

$$F_m = 65 \times 60Hz = 3900Hz$$

Al ingresar este valor de frecuencia de muestreo en el programa MAT1 se obtuvieron los siguientes resultados utilizando a PRD en un rango de 1 a 20 y el valor de tolerancia (tol = 0.1)

Tabla 5.8 Resultados obtenidos de MAT1 con N=65.

PRD	1	3	8	14	18
$TA \times TB$	1282	641	285	171	135

Se elige $TA \times TB$ =285 y se utiliza MAT2 para calcular lo valores de TA y TB. El resultado obtenido es TA = 15 y TB =19 o TB = 15 y TA = 19

Ejemplo 2: Se desean tomar 33 muestras de una señal de 60 Hz. ¿Cual debe de ser la frecuencia de muestreo necesaria? y una vez obtenida ¿cuales son los valores de configuración?

$$F_m = 33 \times 60Hz = 1980Hz$$

Al ingresar este valor de frecuencia de muestreo en MAT1 se obtienen los siguientes resultados utilizando a PRD en un rango de 1 a 20 y el valor de tolerancia (tol = 0.1).

Tabla 5.9 Resultados obtenidos de MAT1 con N=33.

PRD	9	16	24	26	30	32	37
$TA \times TB$	505	297	202	187	163	153	133

Se elige $TA \times TB$ =133 y se utiliza MAT2 para calcular lo valores de TA y TB. El resultado obtenido es TA = 19 y TB =7 o TB = 19 y TA = 7.

Una vez obtenida la configuración de la AIC para cada uno de los ejemplos, el siguiente paso es probarlo en el laboratorio. Para implementarlo se utilizan los mismos componentes que en el caso de estudio 3 (Figura 5.19), mientras que el programa que correrá el DSP es el programa CAP4. En la configuración del registro de control de la unidad AIC se deshabilita el filtro pasabanda.

La señal V_1 es una señal de 60Hz mientras que la señal V_2 es una señal de 300 Hz, la suma de estas dos señales es la que se suministra a la entrada de la AIC. A continuación las Figuras 5.27 y 5.28 muestran las señales V_1 y V_2 medidas con el osciloscopio, mientras que la suma de estas dos señales se muestra en la Figura 5.29

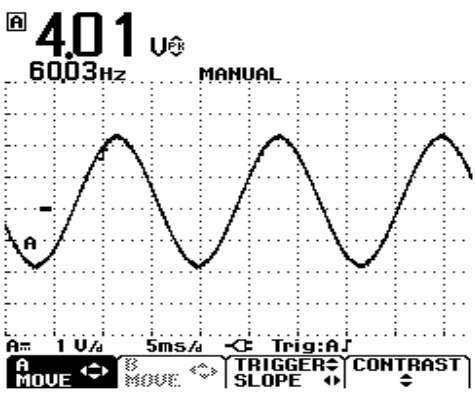


Figura 5.27 Señal V_1 en el caso de estudio 4.

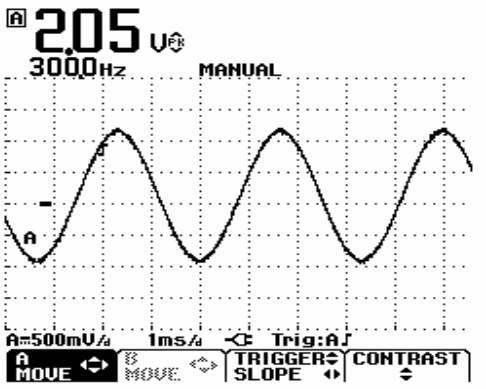


Figura 5.28 Señal V_2 en el caso de estudio 4.

La Figura 5.29 muestra la suma de la señales medida con el osciloscopio.

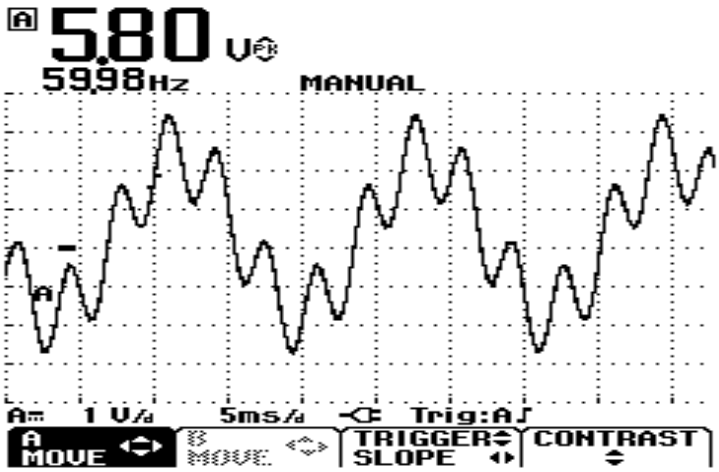


Figura 5.29 Señal V_{IN} obtenida de la suma de V_1 y V_2 en el caso de estudio 4.

Primeramente, con los valores de configuración TA =15, TB =19, PRD =8 y un numero de muestras N = 65 se obtuvo la medición de la Figura 5.30, la cual corresponde a la señal muestreada y almacenada en la memoria del DSP. Se puede observar que la señal almacenada corresponde bien con la señal de entrada. Cabe mencionar que la Figura 5.30 muestra una señal periódica debido a que las 65 muestras de la señal de entrada se están enviando permanentemente a la salida del DSK.

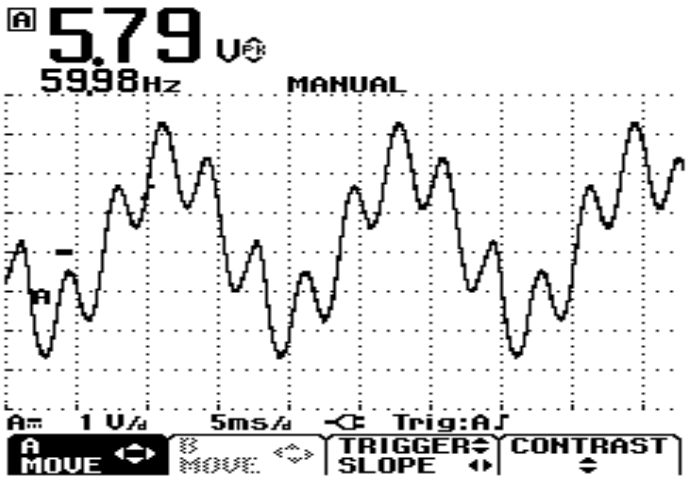


Figura 5.30 Señal V_{OUT} muestreada y almacenada en la memoria del DSP con N=65.

Con los valores de configuración TA = 7, TB =19, PRD =37 y un número de muestras N = 33 se obtuvo la medición de la Figura 5.31.

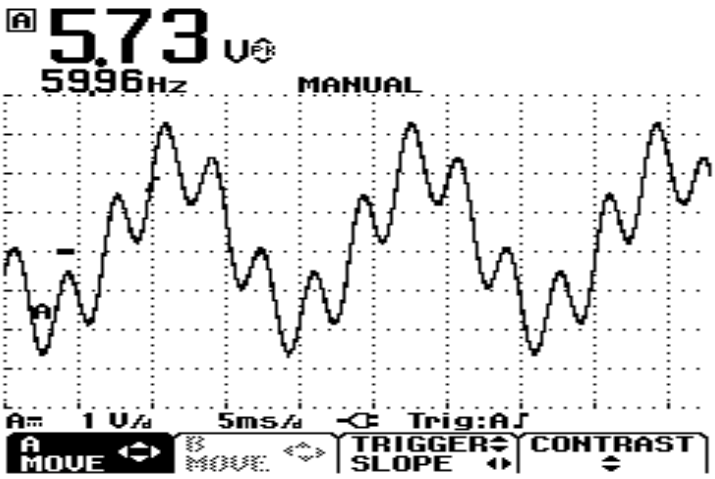


Figura 5.31 Señal V_{OUT} muestreada y almacenada en la memoria del DSP con N=33.

En los dos casos anteriores se observa que la señal ha sido muestreada y almacenada en la memoria de manera correcta ya que la frecuencia y la magnitud de cada una de ellas es similar a la de la señal de entrada.

También se realizaron mediciones para observar que tan cerca se encuentran las frecuencias de muestreo calculadas (F_m = 3900 Hz y F_m = 1980 Hz) a las frecuencias de muestreo medidas. A continuación se muestran las Figuras 5.32 y 5.33 las cuales corresponden a la frecuencia de muestreo medida par N=65 y N=33 respectivamente.

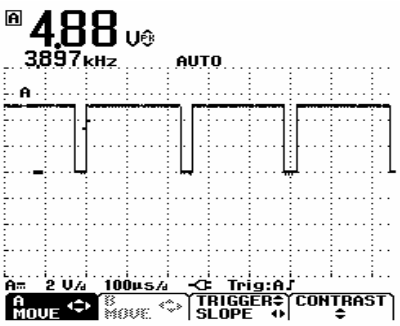


Figura 5.32 Frecuencia de muestreo medida en N = 65.

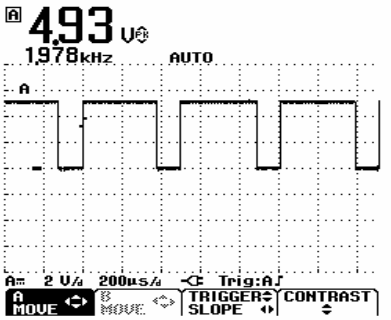


Figura 5.33 Frecuencia de muestreo medida en N = 33.

5.5 CONCLUSIONES.

A continuación se presentan las conclusiones de este capítulo, las cuales se basan en los resultados obtenidos en los casos de estudio anteriores.

- Se presentan cuatro aplicaciones que utilizan los filtros pasabajas y pasabanda analógicos de la unidad AIC.

- Se demostró que el filtro pasabajas analógico que constituye un elemento primordial en la unidad AIC del DSK es un filtro elíptico de séptimo orden con un rizado en la banda de paso de 0.5 dB y un rizado en la banda de paro de 58 dB.

- Se realizó un caso de estudio en el cual se estudia el fenómeno de traslape. Este fenómeno ocurre cuando se elige de manera inadecuada tanto la frecuencia de corte como la frecuencia de muestreo, lo que da por resultado el no poder recuperar de manera adecuada la señal original. Debido a esto se debe de tener mucho cuidado al momento de elegir la frecuencia de muestreo y la frecuencia de corte de filtro pasabanda analógico y se recomienda de preferencia elegir estos parámetros lo más lejano posible del limite del criterio de Shannon.

- También se demostró que el filtro pasabanda analógico de la unidad AIC funciona de manera adecuada para separar dos señales que han sido sumadas previamente. Este filtro es un filtro analógico, pero su configuración y control es digital ya que sus frecuencias de corte son controladas de manera digital a través de un programa en el DSK. Esto nos permite una gran flexibilidad, ya que al tener la ventaja de poder configurar al filtro pasabanda analógico de manera digital es útil para diferentes tipos de aplicaciones.

- Se mostró un caso de estudio en el cual se realizaba el muestreo y almacenamiento de una señal en la memoria interna del DSP. Este proceso de almacenamiento es dependiente de la señal de entrada, ya que si no se conoce no seria posible determinar la frecuencia de muestreo necesaria para poder tomar N muestras. Aunque en este caso se tiene esta limitante, de cualquier forma se demuestra que el DSP trabaja de manera adecuada para este tipo de aplicaciones. Como era de esperarse los mejores resultados se obtuvieron cuando se tomó el mayor número de muestras.

CAPITULO 6

CONCLUSIONES

En este capítulo se presentan las conclusiones generales, las cuales se obtiene a partir de los resultados obtenidos en este trabajo.

6.1 CONCLUSIONES GENERALES

El procesamiento digital de señales es hoy en día un campo que ha tenido un gran avance debido a las aportaciones realizadas por un sinnúmero de personas. Pero fue apenas unos años atrás cuando esta disciplina tomo auge con la llegada de dispositivos que permitían la realización de aplicaciones en tiempo real, los cuales conocemos como DSPs.

Este trabajo esta basado en la tarjeta DSK, la cual cuenta tanto con el *hardware* como con el *software* necesario para la realización de aplicaciones en tiempo real. El *hardware* cuenta con dos elementos principales que son el DSP y la unidad AIC. Estos elementos juegan un papel elemental en cada aplicación por lo que es importante el conocer su funcionamiento de una manera adecuada. La unidad AIC es la encargada de comunicar al DSP con el mundo real mientras que el DSP es considerado el centro de la aplicación ya que es el encargado del control y manipulación de la información de entrada y salida. En lo que corresponde al *software* se puede mencionar que es fácil de utilizar y no se requiere invertir demasiado tiempo en su aprendizaje.

En este trabajo se presentan cuatro casos de estudio los cuales están enfocados a la configuración de la unidad AIC a través del DSP por lo cual fue necesario realizar un programa que nos permitiera realizar dicha configuración. Los resultados obtenidos en los cuatro casos de estudio basados en los filtros analógicos ya existentes en la unidad AIC fueron muy ilustrativos ya que permitieron analizar el desempeño del dispositivo bajo diferentes consideraciones de operación. En el primer caso de estudio se obtuvo la respuesta a la

frecuencia del filtro pasabajas analógico de salida y se comparó con la respuesta ideal, la cual se obtuvo con la ayuda de Matlab. De esta manera se pudo corroborar la información entregada por el fabricante acerca del desempeño del producto. También se pudo observar el fenómeno del traslape de una manera muy simple mediante la manipulación de los parámetros de configuración de la unidad AIC. En el caso de estudio tres se demostró la flexibilidad y eficiencia del filtro pasabanda analógico, ya que se configuró para separar la suma de dos señales de entrada de una manera correcta. En lo que respecta al caso de estudio cuatro, no solo se realizó la configuración de la unidad AIC y del DSP si no que también se implementó una rutina para el almacenamiento de la señal de entrada.

Respecto al filtro *antialiasing* de entrada el cual es un filtro analógico, su configuración la realizamos de manera digital a través del DSP y un programa de cómputo. Gracias a esto tenemos la flexibilidad de reconfigurar al filtro de la manera más conveniente que lo requiera la aplicación que se desee implementar.

6.2 TRABAJOS FUTUROS

Después de realizar los cuatro casos de estudios anteriores queda demostrado que el *DSP Starter Kit* ofrece una valiosa herramienta para el desarrollo de aplicaciones en tiempo real y deja un amplio campo para posibles trabajos futuros. A continuación se mencionan algunos de estos posibles proyectos:

- Realizar la implementación de un filtro digital del tipo IIR o FIR.
- Realizar la implementación de un controlador en tiempo real por ejemplo, el desarrollo de un controlador PID.
- Es posible realizar el monitoreo y almacenar una señal proveniente de un dispositivo en la memoria del DSP con la finalidad de llevar a cabo posteriormente el procesamiento de dicha señal.
- Realizar el muestreo y almacenamiento de una señal de entrada para posteriormente mediante una transformada rápida de Fourier determinar las componentes de frecuencia de la señal original.

REFERENCIAS:

[Chassaing 2005] Rulph Chassaing, *Digital Signal Processing and Applications with the C6713 and C6416 DSK*, Ed.Wiley Interscience, 2005, EUA.

[Chen 1987] Carson Chen, *Active Filter Design*, Ed. Hayden Books, 1987, EUA.

[Delauriere y Grislain] Agnes Delauriere y Oliver Grislain, "Initializing the TLC32040 AIC on the TMS320C5x DSK", Texas Instrument Inc.

[Internet 2005] "Filtros Elípticos", www.virtual.unal.edu.co/cursos/sedes/manizales/4040016/docs-curso/cap-4/cap4-15.html.

[Internet 2005] "Procesamiento Digital de Señales", www.caveo.com.ar/

[Internet 2005] "Procesamiento Digital de Señales", www.maxwell.elai.upm.es/

[Miller y Dewe 1992] A J. V. Miller y Michael B. Dewe, "Multichanel Continuous Harmonic Analysis in Real Time", IEEE Transactions on Power Delivery, Vol. 7, No. 4, págs. 1813-1819, Octubre 1992.

[Miller y Dewe 1993] A. J. V. Miller y M. B. Dewe, "The application of Multi-rate Digital Signal Processing Techniques to the Measurement of Power System Harmonic Levels", IEEE Transactions on Power Delivery, Vol. 8, No. 2, págs 531-539, Abril 1993.

[Ogata 1996] Katsuhiko Ogata, *Sistemas de Control en Tiempo Discreto*, Ed. Prentice Hall, 1996, Mexico.

[Proakis y Manolakis 1998] John G. Proakis, Dimitris G. Manolakis, *Tratamiento Digital de Señales*, Ed. Prentice Hall, 1998, EUA.

[Texas Instrumen 1993] Texas Instrument Inc., *TMS320C5X User Guide (SPRUO56B)*, Texas Instrument, 1993.

[Texas Instrument 1994] Texas Instrument Inc., *TMS320C5X DSP Starter Kit (SPRU101)*, Texas Instrument, 1994.

APÉNDICE A:

En este apéndice se presentan las instrucciones que se utilizaron en los programas CAP3 y CAP4.

Tabla a.1 Instrucciones utilizadas en los programas CAP3 y CAP4

NEMONICO	DESCRIPCION
ADD	El contenido de una dirección de memoria o una constante es sumada al ACC.
AND	ACC realiza una operación AND con el contenido de una dirección de memoria o una constante.
B[D]	Carga en el Program Counter (PC) la dirección en la que se seguirá ejecutando el programa.
CALL[D]	El PC es incrementado y cargado en el stack TOS, mientras que el PC es cargado con la dirección en que se seguirá ejecutando el programa.
CC[D]	Si se cumple la condición el PC es cargado con la dirección en que se seguirá ejecutando el programa, si no entonces el programa se ejecutara en la instrucción siguiente a CC.
CLRC	Pone en 0 un bit en los registro ST0 y ST1.
CMPR	Realiza la comparación entre ARCR y uno de los 8 registros auxiliares.
IDLE	Espera hasta que ocurra una interrupción.
LACC	El contenido de una dirección de memoria o una constante son cargados en ACC con corrimientos.
LACL	El contenido de una dirección de memoria o una constante es cargada en lo 16 LSBs de ACC.
LAMM	Los 16 LSBs de ACC son cargados con un registro Interno del DSP.
LAR	El contenido de una dirección de memoria o constante es cargado en uno de los 8 registros auxiliares.
LDP	El contenido de una dirección de memoria o una constante es cargada en DP.
MAR	Carga un valor en ARP.
NOP	No operación.
OPL	Realiza una operación OR entre DBMR o una constante con el contenido de una dirección de memoria.
RET[D]	Es usado por las instrucciones CALA, CALL y CC para regresar el valor de PC.
RETE	Regresa el valor a PC, además devuelve el contexto y limpia el bit INTM.
RPT	Repite la siguiente instrucción.
SACH	Almacena los 16 MSBs de ACC con corrimiento.
SACL	Almacena los 16 LSBs de ACC con corrimiento.
SAMM	Almacena los 16 LSBs de ACC en un registro interno.
SETC	Pone en 1 un bit en los registros ST0 y ST1
SPLK	Almacena una constante de 16 bits en una dirección de memoria.

APÉNDICE B

En este apéndice se encuentra el código fuente del programa CAP3.

```
*************************************************************
*                                                           *
*************************************************************
*                                                           *
*                                                           *
*  DESCRIPCION: Esta rutina inicializa al DSP TMS320C50 y al *
*               TLC32040 del DSP Starter Kit. Además muestrea *
*               una señal de entrada y envía el dato        *
*               nuevamente a la salida de la unidad AIC.    *
*                                                           *
*************************************************************
        .mmregs

        .ds   0f00h
TA      .word  30       ; Valores de configuration de la AIC
TB      .word  15
RA      .word  20
RB      .word  20
AIC_CTR .word  19h


*************************************************************
*  VECTOR DE INTERRUPCIONES                                 *
*************************************************************
        .ps   080ah
rint:   B     RECEIVE           ;0A; Interrupción de la recepción del puerto serie RINT.
xint:   B     TRANSMIT ;0C;     Interrupción de la transmisión del puerto serie XINT.


*************************************************************
*  INICIALIZATION DEL TMS320C50                             *
*                                                           *
*************************************************************
        .ps 0a00h
        .entry
START:  SETC   INTM             ; Desabilita las interrupciones
        LDP    #0               ; DP=0
        OPL    #0834h,PMST      ; OVLY=1, RAM=1, IPTR=1
        LACC   #0               ; ACC=0
        SAMM   CWSR             ; Desabilita wait states
        SAMM   PDWSR            ;

        SPLK   #022h,IMR        ; XINT=1, INT2=1
        CALL   AICINIT          ; Inicializa la unidad AIC


        SPLK   #12h,IMR         ; RINT=1, INT2=1
        CLRC   INTM             ; Habilita las interrupciones

WAIT    NOP             ; Espera a que un dato sea recibido
        NOP
```

```
            NOP
            B       WAIT

;------- Fin del programa principal ----------;

*********************************************************************
* RUTINA DE INTERUPCION DE LA RECEPCION                             *
*********************************************************************

RECEIVE:  LAMM   DRR        ; ACC= Dato que llego
          AND    #0FFFCh    ; Los 2 LSBs de ACC son cero
          SAMM   DXR        ; Se envía el dato que llego
          RETE              ; Regresa al programa principal


*********************************************************************
* RUTINA DE INTERRUPCION DE LA TRANSMISION                          *
*********************************************************************

TRANSMIT:
          RETE              ;Regresa al programa principal


*********************************************************************
* ESTA RUTINA INICIALIZA LA UNIDAD AIC, EL PUERTO SERIE Y EL TIMER  *
*********************************************************************


AICINIT:
;----------------------  Inicialización del Timer
      SPLK   #4h,PRD        ; Carga el periodo del timer
      SPLK   #20h,TCR       ; Restablece el timer y recarga el periodo

;---------------------------------------------------------------------
;----------------------  Inicialización del puerto serie
      LACC   #0008h         ; Configura el puerto serie en modo Burst
      SACL   SPC            ; FSX como entrada y la comunicación es en
      LACC   #00c8h         ; 16 bits
      SACL   SPC
;---------------------------------------------------------------------
;----------------------  Restablece a la unidad AIC
      MAR    *,AR0          ; ARP apunta a AR0
      LACC   #080h          ; ACC=00000080h
      SACH   DXR            ; Envía 0000h a la unidad AIC
      SACL   GREG           ; GREG=0080h habilita memoria global
      LAR    AR0,#0FFFFh    ; AR0=0FFFFh
      RPT    #10000         ; Repite 10001 veces la instrucción siguiente
      LACC   *,0,AR0        ; ACC = Contenido de la dirección 0FFFFh
      SACH   GREG           ; GREG=0 Deshabilita memoria global
;---------------------------------------------------------------------
;----------------------  Inicialización de TA y RA
      LDP    #TA            ; DP= Pagina donde se encuentra TA
      SETC   SXM            ; SXM=1
      LACC   TA,9           ; ACC= 00 TA 000000000 binario
      ADD    RA,2           ; ACC= 00 TA 00  RA 00 binario
```

```
      CALL    AIC_2ND        ; manda TA y RA a la unidad AIC
;---------------------------------------------------------------------
;----------------------  Inicialización de TB y RB
      LDP    #TB             ; DP= Pagina donde se encuentra TB
      LACC    TB,9           ; ACC= 0 TB 000000000  binario
      ADD     RB,2           ; ACC= 0 TB 0  RB  00  binario
      ADD     #02h           ; ACC= 0 TB 0  RB  10  binario
      CALL    AIC_2ND        ; manda TB y RB a la unidad AIC
;---------------------------------------------------------------------
;----------------------  Inicializa el registro de control
      LDP    #AIC_CTR         ; DP= Pagina donde se encuentra AIC_CTR
      LACC    AIC_CTR,2       ; ACC= 00000000 AIC_CTR 00 binario
      ADD     #03h            ; ACC= 00000000 AIC_CTR 11 binario
      CALL    AIC_2ND         ; manda AIC_CTR a la unidad AIC
      RET                     ; Regresa de subrutina

**************************************************************************
*ESTA RUTINA SE ENCARGA DE CONFIGURAR LA UNIDAD AIC, REALIZA EL   *
* EL PROTOCOLO DE COMUNICACION PRIMARIO Y SECUNDARIO              *
**************************************************************************

AIC_2ND:
      LDP    #0              ; DP=0
      SACH    DXR            ; DXR=0
      CLRC    INTM           ; Habilita las interrupciones
      IDLE                   ; Espera a que el dato sea enviado
      ADD     #6h,15         ; ACC= 0000 0000 0000 0011 (Dato de configuración)
      SACH    DXR            ; DXR= 0000 0000 0000 0011 Comunicacion primaria
      IDLE                   ; Espera a que el dato sea enviado
      SACL    DXR            ; DXR= (Dato de configuración)
      IDLE                   ; Espera a que el dato sea enviado
      LACL    #0             ; ACC=0h
      SACL    DXR            ; Envía un dato para estar seguro que se enviaron
      IDLE                   ; los primeros 16 bits
      SETC    INTM           ; Deshabilita las interrupciones
      RET                    ; Regresa a la subrutina AICINIT
      .end                   ; Fin del programa
```

APÉNDICE C

En este apéndice se encuentra el código fuente del programa CAP4.

```
*************************************************************
*    DESCRIPCION:  Esta rutina inicializa la unidad AIC y al          *
*                  DSP TMS30C50, además toma N datos de una         *
*                  señal y los almacena en la memoria para          *
*                  posteriormente enviarlos a la salida de la       *
*                  AIC de manera repetitiva.                        *
*************************************************************

            .mmregs
            .ds     0f00h
TA          .word   17      ; Valores de configuración de la AIC
TB          .word   19
RA          .word   18
RB          .word   15
AIC_CTR     .word   18h

            .ds     0300h
apun1       .word   0304h     ; Dirección de memoria a partir de la cual
                              ; se almacenaran los datos
apun2       .word   0h
n           .word   129       ; Numero de muestras

*************************************************************************
* VECTOR DE INTERRUPCIONES                                              *
*************************************************************************
      .ps     080ah
      B       RINT            ; Rutina de interrupción de recepción del puerto serie
      B       XINT            ; Rutina de interrupción de transmisión del puerto serie

*************************************************************************
*  INICIALIZACION DEL TMS320C50                                         *
*************************************************************************
            .ps     00a00h
            .entry
 START:   SETC      INTM            ; Deshabilita las interrupciones
          LDP       #0              ; DP=0
          OPL       #0834h,PMST     ; OVLY=1, RAM=1, IPTR=1
          LACC      #0              ; ACC=0
          SAMM      CWSR            ; Desabilita wait states
          SAMM      PDWSR
          SPLK    #022h,IMR         ; XINT=1, INT2=1
          CALL    AICINIT           ; Inicializa las interrupciones
          SPLK    #032h,IMR         ; RINT=1, XINT=1, TINT=1
          LDP     #6                ; DP=6
          LACC    0h                ; ACC= Contenido de la loc. 0300h
          ADD     2h                ; ACC= A la suma del contenido de las loc. 0300h y 0302h
          SACL    1h                ; loc. 0301h=ACC
          MAR     *,AR1             ; ARP apunta a AR1
          SAMM    ARCR              ; ARCR=ACC
          LAR     AR1,0h            ; AR1= Contenido de la loc. 0300h
          CLRC    INTM              ; Habilita las interrupciones
```

```
NEXT:   IDLE                ; Espera a que ocurra una interrupción.
        B     NEW           ; El programa es llevado a la etiqueta NEW.

NEW:    LACC   3h           ; ACC= Contenido de la loc. 0303h
        SACL   *+,0,AR1     ; Guarda ACC en la loc. a la que apunta AR1
                            ; además incrementa el contenido en la loc. AR1

        CMPR   0            ; Si ARCR=AR1 entonces TC=1 si no TC=0
        CC     RESET,TC     ; Si TC=1 ve a la etiqueta RESET
                            ; en caso contrario pasa a la sig. instrucción
        B      NEXT         ; Ve a la etiqueta NEXT


RESET:  LDP    #0           ; DP=0
        SPLK   #0048h,SPC   ; Inhabilita la recepción del puerto serie
TRANS:  LDP    #6           ; DP=6

        LAR    AR1,0h       ; AR1= Contenido en la loc. 0300h

NEW1:   LACL   *+,AR1       ; ACC= Contenido de la loc. a la que apunta AR1
                            ; además incrementa AR1
        SAMM   DXR,0        ; Envía el dato
        IDLE                ; Espera a que ocurra la interrupción
        CMPR   0            ; Si ARCR=AR1 entonces TC=1 si no TC=0
        CC     TRANS,TC     ; Si TC=1 ve a la etiqueta TRANS
                            ; en caso contrario pasa a la sig. instrucción

        B      NEW1         ; Ve a la etiqueta NEW1

;------------ Fin del programa principal----------

*************************************************************************
* RUTINA DE INTERRUPCION DE LA RECEPCION                                *
*************************************************************************

RINT:   LAMM   DRR          ; ACC= Al dato que llego
        SACL   3h           ; loc. 0303=ACC
        RETE                ; Regresa de la interrupción

*************************************************************************
* RUTINA DE INTERRUPCION DE LA TRANSMISION                              *
*************************************************************************

XINT:   RETE          ; Regresa de la interrupción

*************************************************************************
* ESTA RUTINA INICIALIZA LA UNIDAD AIC, EL TIMER Y EL PUERTO SERIE     *
*************************************************************************
AICINIT:
;-------------------------------- Inicialización del Timer
      SPLK   #3h,PRD        ; Carga el periodo del timer
      SPLK   #20h,TCR       ; Restablece el timer y recarga el periodo
      MAR    *,AR0          ; ARP apunta a AR0

;-----------------------------------------------------------------------
```

```
;-------------------------------- Inicialización del puerto serie
      LACC   #0008h          ; Configura el puerto serie en el modo Burst
      SACL   SPC             ; FSX como entrada y se comunica en
      LACC   #00c8h          ; 16 bits
      SACL   SPC
;-------------------------------------------------------------------------
;-------------------------------- Se aplica un RESET a la unidad AIC
      LACC   #080h            ; ACC=00000080h
      SACH   DXR              ; Envía 0000h a la unidad AIC
      SACL   GREG             ; GREG=0080h habilita memoria global
      LAR    AR0,#0FFFFh      ; AR0=0FFFFh
      RPT    #10000           ; Repite 10001 veces la sig. instrucción
      LACC   *,0,AR0          ; ACC= Contenido de la loc. 0FFFFh
      SACH   GREG             ; Deshabilita memoria global
;-------------------------------------------------------------------------
;--------------------------------  Inicialización de TA y RA
      LDP    #TA              ; DP= Pagina donde se encuentra TA
      SETC   SXM              ; SXM=1
      LACC   TA,9             ; ACC= 00 TA 000000000 binario
      ADD    RA,2             ; ACC= 00 TA 00  RA 00 binario
      CALL   AIC_2ND          ; Manda TA y RA a la unidad AIC
;-------------------------------------------------------------------------
;--------------------------------  ; Inicialización de TB y RB
      LDP    #TB              ; DP= Pagina donde se encuentra TB
      LACC   TB,9             ; ACC= 0 TB 000000000 binario
      ADD    RB,2             ; ACC= 0 TB 0  RB  00 binario
      ADD    #02h             ; ACC= 0 TB 0  RB  10 binario
      CALL   AIC_2ND          ; Manda TA y RA a la unidad AIC
;-------------------------------------------------------------------------
;--------------------------------  ; Inicialización del registro de control
      LDP    #AIC_CTR         ; DP= Pagina donde se encuentra AIC_CTR
      LACC   AIC_CTR,2        ; ACC= 00000000 AIC_CTR 00 binario
      ADD    #03h             ; ACC= 00000000 AIC_CTR 11 binario
      CALL   AIC_2ND          ; Manda AIC_CTR a la unidad AIC
      RET                     ; Regresa de subrutina
*************************************************************************
* ESTA RUTINA SE ENCARGA DE CONFIGURAR LA UNIDAD AIC, REALIZA EL  *
* EL PROTOCOLO DE COMUNICACION PRIMARIO Y SECUNDARIO              *
*************************************************************************
AIC_2ND:
      LDP    #0               ; DP=0
      SACH   DXR              ; DXR=0
      CLRC   INTM             ; Habilita las interrupciones
      IDLE                    ; Espera a que ocurra la interrupción
      ADD    #6h,15           ; ACC= 0000 0000 0000 0011 (Dato de configuración) binario
      SACH   DXR              ; DXR= 0000 0000 0000 0011
      IDLE                    ; Espera a que ocurra la interrupción
      SACL   DXR              ; DXR= (Dato de configuración)
      IDLE                    ; Espera a que ocurra la interrupción
      LACL   #0               ; ACC=0
      SACL   DXR              ; Envia un dato para estar seguro que se enviaron
                              ; los primeros 16 bits
      IDLE                    ; Espera a que ocurra la interrupción
      SETC   INTM             ; Deshabilita las interrupciones
      RET
      .end
```

APÉNDICE D

En este apéndice se encuentra el código del programa en Matlab MAT1.

```
tcf=0                  % variables que se utilizaran durante el programa para almacenar datos
apun=0
result=0
result1=0
tol=0.1                % especifica la diferencia mínima que debe de haber entre el valor calculado
                       % y su entero superior o inferior mas cercano para poderlos almacenar.
tc=0                   % tc=TA*TB
tcl=8                  % tcl es el limite inferior del producto TA*TB
tch=1953               % tch es el limite superior del producto TA*TB
fm=7500                % frecuencia de muestreo

if fm<=19200           % 19200 es la máxima frecuencia de muestreo a la que opera la unidad AIC
  for PRD=1:25                 %ciclo que controla el numero de valores que puede tomar PRD
    fmaster=1/(50e-9*(1+PRD))
    tc=fmaster/(2*fm)

    if tc>=tcl & tc <= tch     % si el valor calculado de tc se encuentra dentro del rango permitido
      tcf=abs(round(tc)-tc)    % lo resta con su entero superior o inferior mas cercano
      if tcf < tol             % si el resultado es menor a tol almacena el valor de configuración
        apun=apun+1            % PRD en la variable result1 y el valor del producto
        result(apun)=round(tc)     % TA*TB en la variable result
        result1(apun)=PRD
      end
    end
  end
end
```

APÉNDICE E

En este apéndice se encuentra el código del programa MAT2.

```
aux=0
aux1=0
taf=0                  % Variables que se utilizaran durante el programa
tbf=0
apun=0
tc=148                 % tc=TA*TB

for tb=2:4             % Esta parte del programa lo que realiza es la búsqueda en los
   aux=tb*4            % cuatro primeros renglones y si encuentra algún resultado
  if aux > tc          % de ta y tb lo guarda en las variables taf y tbf
    break
  end
  for ta=4:31
    aux1=tb*ta
    if aux1==tc
      apun=apun+1
      taf(apun)=ta
      tbf(apun)=tb
    elseif aux1>tc
      break
    end
  end
end

for tb=5:63            % Esta parte del código realiza la búsqueda en los demás
  aux=tb*tb            % renglones a partir de la diagonal principal y guarda el
  if aux > tc          % resultado de ta y tb en las variables taf y tbf.
    break
  end
  for ta=tb:31
    aux1=ta*tb
    if aux1==tc
      apun=apun+1
      taf(apun)=ta
      tbf(apun)=tb
    elseif aux1>tc
      break
    end
  end
```