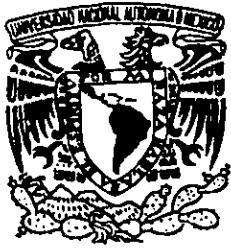


42

UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

CAMPUS ARAGÓN



FILTRO DIGITAL TIPO FIR EN UN FPGA

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
P R E S E N T A:
MONTIEL CORNEJO DANTE

ASESORES: ING. JOEL LÓPEZ CONTRERAS

SAN JUAN DE ARAGON, ESTADO DE MÉXICO

2000.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



ESTADOS UNIDOS MEXICANOS
 REPUBLICA NACIONAL
 SUPREMACIA DE
 MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
 ARAGÓN
 DIRECCIÓN

DANTE MONTIEL CORNEJO
 PRESENTE.

En contestación a la solicitud de fecha 28 de junio del año en curso, relativa a la autorización que se le debe conceder para que el señor profesor, Ing. JOEL LÓPEZ CONTRERAS pueda dirigirle el trabajo de tesis denominado, "FILTRO DIGITAL TIPO FIR EN UN FPGA", con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

Aprovecho la ocasión para reiterarle mi distinguida consideración.

Atentamente
 "POR MI RAZA HABLARÁ EL ESPÍRITU"
 San Juan de Aragón, México, 3 de julio del 2006
 EL DIRECTOR

Lic. CARLOS EDUARDO LEVY VÁZQUEZ



- C p Secretaría Académica.
- C p Jefatura de la Carrera de Ingeniería Mecánica Eléctrica.
- C p Asesor de Tesis.

CELV/AIR/VSRL:lia.

DEDICATORIAS
Y
AGRADECIMIENTOS

A Dios

Por darnos su regalo más hermoso: la vida.

A mis padres: Manuel y Guadalupe

Por su infinita paciencia y amor incondicional.

A mis hermanos: David y Orlando

Por su apoyo y sus palabras de aliento para superarme.

A mis amigos y amigas

Por su solidaridad y su entusiasmo.

Especialmente a Rubén y Juan.

A mi familia

Por todas las veces en que sentí su apoyo y cercanía.

A mis Abuelos: Esteban (qdep) y Celia (qdep)

Por la sabiduría que me transmitieron.

A la UNAM y mis profesores

Por formarme con un espíritu abierto a nuevas ideas.

Una mente crítica y culta.

*Dedico esta tesis, a todos los seres humanos sabios,
En especial a Gloria L. por otorgarme amor,
Por su valentía y generosidad.*

Labor Omnia Superat

FILTRO DIGITAL

TIPO FIR

EN UN FPGA

Objetivo:

Diseñar y Construir un Filtro Digital del tipo FIR en un dispositivo de Arreglo de Compuertas Programables en Campo (FPGA), mostrando la Teoría de Procesamiento Digital de Señales.

INTRODUCCIÓN

En el proceso de formación académica profesional se llegó a adquirir conocimientos en las áreas de electrónica, comunicaciones y sistemas digitales; siendo éstas dos últimas las más apasionante para autor en cuanto a todo lo relacionado con la tecnología digital. En el área de comunicaciones digitales, modulación, intrigaba bastante cual era el proceso detallado de un filtrado digital, primero se investigó la naturaleza del proceso. El proceso de filtrar una señal es para eliminar componentes no deseados, estos componentes no deseados pueden ser ruido, alguna frecuencia correlacionada con nuestra señal de interés. Buscando la forma en que este proceso pudiera implementarse en algún circuito digital a la vez de utilizar componentes programables de los cuales se tienen noticias en los cursos de sistemas digitales, se procede a efectuar un trabajo de investigación si es posible realizar un filtro digital en un dispositivo programable.

En el capítulo uno se introduce al lector en las definiciones básicas para ubicarlo en el contexto de la tesis; tipos de filtros, tipos de respuestas de los filtros y la definición básica del dispositivo utilizado.

En el capítulo dos se presenta la teoría matemática del procesamiento digital de señales; indicando sus ventajas de procesar la información en forma digital, también se presentan las herramientas matemáticas para el procesamiento de señales digitales, tales como: el análisis de Fourier, la transformada z y finalmente correlación y convolución.

En el capítulo tres se describe la metodología de diseño del filtro digital, las partes que lo componen y a que leyes se sujetan cada parte que lo componen.

En el capítulo cuatro se indican los antecedentes de los dispositivos programables, hasta la aparición del FPGA, su clasificación de las familias de FPGA's, y una descripción de las partes que componen al FPGA y las funciones que llevan a cabo cada parte.

En el capítulo cinco se muestra el proceso que se lleva a cabo para la programación del filtro digital, que a su vez involucran la captura del diagrama esquemático, su simulación y la configuración del FPGA, es decir, todo el proceso que se hace con ayuda de la computadora para programar al dispositivo reconfigurable.

En el capítulo seis se comentan los resultados obtenidos después de programar al FPGA, así también se hace un recuento de los recursos materiales utilizados para llevar a cabo esta investigación. Posteriormente se hace una reflexión de estos resultados lo que da como resultado un capítulo de Conclusiones y Recomendaciones.

Finalmente en la bibliografía aparece un listado de obras consultadas para desarrollar esta tesis.

ANTECEDENTES

En los 35 mil años aproximadamente de la existencia de ser humano¹, como homo sapiens sapiens (hombre Cro-Magnon, los vestigios de esta especie indican que ya poseía una industria lítica², es decir instrumentos hechos de piedra), los descendientes de este ser humano han llegado a desarrollar tecnologías para el aprovechamiento y explotación de la naturaleza. En ese desarrollo el hombre descubrió la electricidad y posteriormente la electrónica para desarrollar herramientas tecnológicas hasta nuestros días; como todo conocimiento que se precie de ser especializado, es decir, debe de ser dividido en ramas de especialización, unas de las ramas en que se divide la electrónica, es la electrónica digital, de la cual cada día se descubren más aplicaciones, extendiendo la frontera de éste vasto campo de conocimientos.

Se puede argumentar que el campo de la electrónica digital se ha visto limitada, por toda la tecnología en la que vivimos inmersos; el tema de vanguardia es la optoelectrónica, componentes electrónicos que trabajan y procesan señales de energía fotónica. Sin embargo, estos son los tiempos en los que aparentemente se han agotado las aplicaciones básicas de la electrónica digital, en particular en los sistemas digitales; la cual corresponde a las nuevas generaciones, o bien, a la gente interesada en desarrollar nuevas tecnologías, buscando nuevas aplicaciones, en nuevos campos y efectuar experimentos para validar las hipótesis de las investigaciones que hoy en día se llevan a cabo.

Desde 1800 a 1837, se inicio el desarrollo de las aplicaciones de electricidad con Volta, los estudios de Laplace, los tratados matemáticos de Fourier, así otros como Cauchy, Jacob, Maxwell, Chebyshev, Darlington y Cauer, llevando a cabo desarrollos matemáticos, que posteriormente fueron aprovechados en su época y en la actualidad; en 1845 se enuncian las leyes de Kirchoff; fue en 1915 con la aparición de la teoría moderna de filtros

¹ AYALA, Francisco J., El Prodigio de la Evolución, Especial de Muy Interesante (Mensual), Edit. Televisa, No. 19, 1998, pp. 12-18.

² IGLESIAS LEAL, Ramiro, La Ruta hacia el Hombre Cósmico, 1ª edición, Instituto Politécnico Nacional, México, 1993, pp. 14.

en que se agrupan todos esos conocimientos. K. Wagner y G. Cambell [12] dan a conocer por separado (Alemania y EU, respectivamente) el concepto de filtro. en 1923 Zobel y los laboratorios Bell publican un método para el diseño de filtros usando matemáticas simples; alrededor de 1940 las teorías de Foster fueron extendidas por Darlington y Cauer; después fueron los desarrollos de polinomios por Butterworth, Chebysev, Bessel y Gauss entre otros, hasta llegar a la década de los 60's con la aparición del transistor y posteriormente las comunicaciones digitales de alta velocidad y es en la década de los 70's cuando aparece el concepto de los filtros digitales teóricos, que eran simulaciones matemáticas de los polinomios, basándose en los estudios realizados por Butterworth, Chebysev, Bessel y Gauss; en lo que respecta a este trabajo de tesis, mencionaremos la aplicación de uno de ellos, siendo el de Butterworth el tipo de aproximación elegido por sus características de respuesta en frecuencia.

A partir de los años 80's con ayuda de la tecnología VLSI, los costos de fabricación de circuitos integrados se reduce así como algunas características de estos circuitos como la reducción del consumo de potencia y el incremento de la velocidad de tiempos de propagación; debido a su escala de integración ya se pueden construir sistemas digitales físicos funcionando como filtros digitales.

ALCANCES

El procesamiento Digital de Señales en nuestros días es importante porque estamos viviendo una época en que lo más valioso es la información que se transmite al mundo, esta información se transmite por principalmente por Internet, que tiene diversos medios de comunicación, por ejemplo, fibra óptica, redes telefónicas y satélite; la mayoría de ese flujo de información va codificada en forma digital. Hablamos de información de casi cualquier tipo: militar, negocios, entretenimiento, comunicaciones, investigación y simples saludos.

Hasta donde se ven las fronteras de aplicaciones del procesamiento digital de señales es en [4]:

- Procesamiento de Imágenes (Reconocimiento de patrones, visión robótica, realce de imagen, facsímiles, satélites de mapas climáticos, animación).
- Instrumentación y Control (análisis espectral, control de posición y velocidad, reducción de ruido, compresión de datos).
- Habla y Audio (reconocimiento de voz, síntesis de voz, texto a voz, audio digital, ecualización).
- Militar (comunicaciones codificadas, procesamiento de radar, procesamiento de sonar, guía de misiles).
- Telecomunicaciones (cancelación de eco, ecualización adaptiva, ADPCM, transcoders, espectro de difusión, videoconferencia, comunicación de datos).
- Biomédicas (monitoreo de pacientes, exploradores, mapeo de electroencefalogramas (EEG), análisis de electrocardiogramas (ECG), almacenamiento y mejora de rayos X).

El filtrado en nuestros días se aplica sobre todo al procesamiento de información de imágenes y voz en formatos digitales, ya sea para almacenamiento, limpieza, etc. Un área de investigación de vanguardia es el usar bancos de filtros digitales, se utilizan para implementar algoritmos de compresión y descompresión de información digital.

ÍNDICE

Carta de Aceptación

Objetivo

Introducción

Antecedentes

Alcances

1. Conceptualización del Sistema	1
1.1 El Filtro	2
1.2 Las señales digitales	4
1.3 Respuesta en frecuencia	6
1.4 Aproximaciones Matemáticas	10
1.5 El Arreglo de Compuertas Programables en Campo	12
2. Procesamiento Digital de Señales	14
2.1 Ventajas del procesamiento digital	15
2.2 Formas de Representación de números binarios	17
2.3 Análisis de Fourier	18
2.4 Transformada Z	32
2.5 Correlación y Convolución	38
3. Diseño del Filtro Digital	43
3.1 Diagrama a bloques del filtro	44
3.2 Arquitectura del sumador	50
3.3 Arquitectura del multiplicador	52
3.4 El retardo	55

4. Arreglo de Compuertas Programables en Campo	57
4.1 El bloque lógico configurable	65
4.2 El bloque de entradas y salidas	69
4.3 Interconexiones	71
5. Programación del Filtro Digital	74
5.1 Diagrama Esquemático	77
5.2 Simulación	78
5.3 Configuración del FPGA	79
6. Resultados y Recursos Utilizados	80
6.1 Resultados	81
6.2 Recursos Utilizados	83
Conclusiones y Recomendaciones	86
Bibliografía	88
Anexos	93

CAPÍTULO

1

La Ciencia es el alma de la prosperidad de las naciones
Y la fuente de la vida de todo progreso

Louis Pasteur

CONCEPTUALIZACIÓN DEL SISTEMA

El área de procesamiento digital de señales es dividida en dos ramas principales: la primera en el análisis espectral, y la segunda en el filtrado. En este primer capítulo se muestran algunas definiciones básicas para ubicar el contexto del trabajo de investigación que se enfoca en la rama del filtrado digital.

1.1 EL FILTRO

Definiendo en un punto de vista de matemáticas: es un proceso de cálculo o algoritmo que permite transformar una señal o una secuencia de números, llamada señal de entrada en otra secuencia, llamada señal de salida. El algoritmo se puede realizar en programas de cómputo o en circuitos digitales.¹

Definiendo en un punto de vista de ingeniería: El filtro es aquella conexión de elementos con ciertas características de sensibilidad que es capaz de discriminar bandas de frecuencia.²

Filtro Analógico.

Son aquellas conexiones que procesan señales analógicas, en las cuales se utilizan elementos analógicos como transistores, amplificadores, resistencias, capacitores, etc. Los cuales están constituidos por pasivos y activos.

¹ <http://cantera.reduaz.mx/~gmiram/intpds.htm>

² Apuntes de clase "Filtrado y Modulación", 1996

Filtros Pasivos.

Estos circuitos están contruidos exclusivamente de elementos pasivos tales como: resistencias, bobinas y capacitores, los cuales son elementos que no tienen un efecto de amplificación de la señal. Estos pueden tener varias configuraciones ya sean T, Π y L: cualquiera de ellos puede ser simple, doble o compuesto.

Se pueden diseñar de dos formas:

- a) Diseño del parámetro imagen, que se trata como una línea de transmisión, estos han evolucionado en dos tipos: el constante k y el m -derivado; y
- b) Diseño del método de redes, trata al circuito como elementos con parámetros propios, se diseña con el balanceo de fórmulas para valores propuestos de algún elemento insertado en el circuito del filtro.

Filtros Activos.

Los filtros activos analógicos [2] se configuran con la ayuda de circuitos sumadores, sustractores, multiplicadores, divisores, derivadores e integradores con amplificadores operacionales para simular y resolver ecuaciones diferenciales que describen sistemas físicos. Constan de elementos pasivos (excepto bobinas ya que su uso sería en ocasiones voluminosos y costosos) asociados a elementos activos (válvulas de vacío, transistores, amplificadores operacionales). Los filtros con válvulas tenían un consumo de potencia muy alto y baja ganancia. Los filtros con transistores disminuyeron el consumo de potencia, pero tenían la ganancia muy por abajo de lo satisfactorio. Los filtros con amplificadores operacionales tienen como característica alta resistencia de entrada, y baja resistencia de salida, buena ganancia.

Filtros Digitales.

Los filtros digitales son expresados o definidos matemáticamente mediante una ecuación diferencial lineal de coeficientes constantes, la cual es una relación entre las secuencias de señales de entrada y las secuencias de señales de salida. Expresado en ingeniería, un filtro digital consiste de un sistema para filtrar señales muestreadas representadas en forma binaria, la operación de filtrado se realiza por medio de cálculos aritméticos directos con las señales muestreadas, estas operaciones son la suma, multiplicación y retardo: son extremadamente estables, no modifican su comportamiento con el tiempo ni con la temperatura. Los filtros digitales realizan sus funciones pero con la diferencia de que el dominio del tiempo es digital, es decir, estamos hablando de muestras codificadas a partir de una señal analógica, el filtro digital funciona puramente con números. Estos suelen dividirse en: Respuesta al Impulso Infinita (IIR por sus siglas en inglés) y en Respuesta al Impulso Finita (FIR por sus siglas en inglés).

Filtro de Respuesta al Impulso Infinita.

El filtro digital de respuesta al impulso infinita (IIR), tiene como característica que sus salidas dependen de señales presentes en la entrada y señales que se encuentren retrasadas, es decir, almacenadas en alguna memoria del filtro.

Filtro de Respuesta al Impulso Finita.

El filtro digital de respuesta al impulso finita (FIR), tiene como característica que sus salidas dependen sólo de las entradas presentes en ese instante.

1.2 LAS SEÑALES DIGITALES

Una señal que varía continuamente se desea convertir a la forma digital, esto se logra al tomar muestras de la señal original a una frecuencia (velocidad de muestreo) que sea al menos dos veces la máxima frecuencia que contenga esa señal; esto se conoce como la

velocidad de muestreo de Nyquist [13]. Ésta frecuencia de muestreo nos permite obtener suficiente información acerca de la señal y se puede estar seguro de que no se ha perdido información importante que contenga la señal.

Teorema del Muestreo.

Una señal de banda limitada sin componentes espectrales por encima de una frecuencia de N Hz, se determina unívocamente por sus valores equidistantes a intervalos no mayores de $1/(2N)$ segundos. Esta es una condición suficiente para que una señal analógica pueda ser totalmente reconstruida a partir de un conjunto de muestras discretas uniformemente espaciadas.

Cuantización.

El sistema numérico normalizado utilizado para el procesamiento de señales digitales es el sistema binario, en este sistema numérico el grupo de códigos consisten en N pulsos cada uno de los cuales indica "encendido" (1 lógico) o "apagado" (0 lógico). En el caso de la cuantización, los N pulsos "encendido-apagado" pueden representar 2^n niveles de amplitud de la señal original que fue previamente muestreada [7]. Dicho de otra forma, es cuando una cantidad física se representa en forma numérica.

Codificación.

Es el proceso de asignación de una palabra o código digital a cada uno de los niveles discretos [7].

1.3 RESPUESTA EN FRECUENCIA

Las bandas de frecuencia al pasar a través de un filtro van a verse afectadas según las características del filtro, unas se verán atenuadas otras tal vez con ganancia, la frecuencia a la cual se delimita la banda de paso y de rechazo se conoce como frecuencia de corte, que se define como, cuando el nivel de ganancia máxima cae al 70 %, que es lo mismo que, la señal tiene una atenuación de -3dB con respecto al nivel máximo de ganancia.

$$70\% = \frac{A}{2}$$

La frecuencia de supresión de banda, es la frecuencia en la que se especifica la atenuación mínima que se requiere en un circuito.

Paso Bajo.

Cuando una señal con frecuencias que se encuentran por debajo de la frecuencia de corte, puede transmitirse a través del filtro paso bajos para entregarse esta señal a una carga; las demás frecuencias que están por encima de la frecuencia de corte son atenuadas: una ganancia negativa en formato de decibeles [12] (Fig. 1).



Fig. 1 Respuesta en frecuencia de un filtro paso bajas.

Paso Alto.

Cuando una señal con frecuencias que se encuentran por encima de la frecuencia de corte, puede transmitirse a través del filtro paso altos para entregarse esta señal a una carga; las demás frecuencias que están por abajo de la frecuencia de corte son atenuadas: una ganancia negativa en formato de decibeles [12] (Fig. 2).

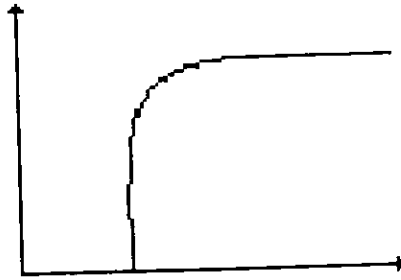


Fig. 2 Respuesta en frecuencia de un filtro paso altos.

Paso Banda.

La señal va a transmitirse a través del filtro paso banda cuando las frecuencias de la señal se encuentren entre una frecuencia de corte inferior y una frecuencia de corte superior. La banda de rechazo ocurre cuando las frecuencias están por arriba y por debajo de la banda de paso. La frecuencia central de la banda de paso es definida geométricamente [12] (Fig. 3).

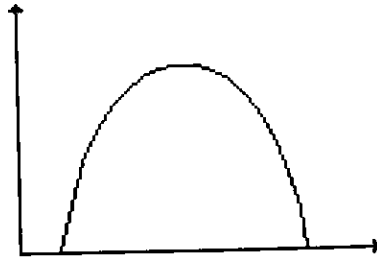


Fig. 3 Respuesta en frecuencia de un filtro pasa banda.

Supresor de Banda.

El filtro rechaza banda va a transmitir una señal en dos bandas de frecuencias. una desde una señal en corriente directa hasta la frecuencia de corte inferior y otra desde la frecuencia de corte superior hasta teóricamente la frecuencia infinita [12] (Fig. 4).

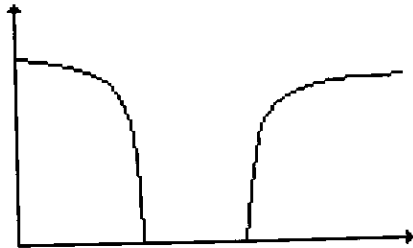


Fig. 4 Respuesta en frecuencia de un filtro supresor de banda.

1.4 APROXIMACIONES MATEMÁTICAS

Butterworth.

Estos presentan una pendiente mayor conforme su orden aumenta; el orden de un filtro en términos matemáticos es el número de polos de su función de transferencia y en términos físicos el número de redes de retardo presentes en su estructura. La curva de respuesta se denomina plana ya que no presenta rizado (Fig. 5).



Fig. 5 Aproximación Butterworth en un filtro paso bajas.

Chebyshev.

Estos presentan rizado en la banda pasante y su curva de transición tiene mayor pendiente al mismo orden respecto a los filtros anteriores (Fig. 6).



Fig. 6 Aproximación Chebyshev en un filtro paso bajas.

Cauer o Elípticos.

Estos presentan rizado en la banda de paso y en la de corte. Son los de mayor definición en la frecuencia de corte, es utilizado en equipos que requieren bastante precisión en el punto de corte y atenuación elevada en la banda de corte, por lo que su pendiente casi es vertical (Fig. 7).



Fig. 7 Aproximación Cauer en un filtro paso bajas.

Bessel.

Llamados también de fase lineal o de retardo lineal en el tiempo. Presentan una variación de atenuación en la región de transición de -6 dB/octava. La frecuencia de corte se define como la frecuencia a la cual el retardo de fase del filtro es la mitad del retraso de fase máximo. su pendiente de corte es menor (Fig 8)

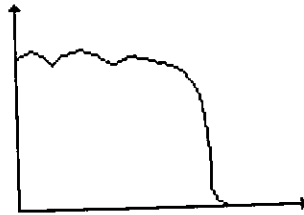


Fig. 8 Aproximación Bessel en un filtro paso bajas.

1.5 EL ARREGLO DE COMPUERTAS PROGRAMABLES EN CAMPO

Uno de los fabricantes de este tipo de dispositivos es la compañía Xilinx, que tiene la característica de desarrollar arreglos de compuertas programables en campo del tipo reconfigurables; los cuales son los más idóneos para implantar los filtros digitales porque nos ayuda a programar el circuito y reprogramarlo para mejoras subsecuentes. así como también nos permite realizar pruebas dinámicas y rápidas. Las características más sobresalientes en su arquitectura son los bloques de entradas y salidas (IOB por sus siglas en inglés) y sus bloques lógicos configurables (CLB por sus siglas en inglés).

El IOB tiene las funciones de:

- a) inversión no lógica de una señal antes de su entrada o de su salida.
- b) control de señales en tri-estado.
- c) selección de salida del tipo combinacional o con ayuda de registro.
- d) selección del tiempo de subida de una señal (slew-rate), y
- e) colocación de resistencias de polarización a nivel lógico alto (pull-up).

Un CLB está compuesto de dos *flip-flops* del tipo *D*, un bloque de lógica combinacional y diversos multiplexores para seleccionar diversos modos de funcionamiento de la celda programable. Las interconexiones entre diferentes dispositivos de CLB e IOB, se hace mediante interconexiones de propósito general, interconexiones directas y si alguna señal es crítica, es decir, que esté presente en varios bloques CLB en el mismo instante, se utilizan líneas largas. Estos dispositivos tienen una gran variedad de opciones para ser programadas.

En este capítulo repasamos las definiciones básicas que se manejan en el área del filtrado, así como también se expresó la descripción básica del dispositivo programable que en el capítulo 4 se verá con mayor detalle. En el siguiente capítulo repasaremos la teoría del procesamiento digital de señales que es el fundamento formal del diseño y análisis del diseño de filtros digitales.

CAPÍTULO

2

La matemática es el arte de hacer una pregunta
Y encontrar la manera de contestarla.

Evaristo Galois

PROCESAMIENTO DIGITAL DE SEÑALES

El procesamiento de señales, es una parte de las matemáticas utilizada por los ingenieros para hacer análisis y diseños de sistemas digitales que involucran el proceso de información en forma digital. Así en este capítulo se presenta la teoría matemática del procesamiento digital de señales que es necesario tener en cuenta para llevar a cabo un buen diseño de sistema digital que involucra el proceso de información digital como lo es un filtro digital.

2.1 VENTAJAS DEL PROCESAMIENTO DIGITAL [4]:

- Exactitud garantizada. La exactitud es determinada por el número de bits usados (rango dinámico o tamaño de palabra).
- Repetibilidad perfecta. Desempeño idéntico de un dispositivo a otro dispositivo es posible de obtenerse ya que no hay variaciones debido a tolerancias de componentes.
- No cambia su desempeño con las variaciones de temperatura o por envejecimiento de componentes.
- Utiliza las ventajas de las tecnologías de semiconductores para lograr mayor fiabilidad, tamaño pequeño (miniaturización), bajo costo, bajo consumo de potencia y alta velocidad.
- Mayor flexibilidad. Los sistemas de procesamiento digital de señales pueden ser programados y reprogramados (depuración) para desempeñar una variedad de funciones, sin modificar el hardware (simulación).
- Versatilidad. El procesamiento digital de señales permite implementar funciones que no son posibles en forma analógica (compresión de imágenes, filtros de ranura, algoritmos de filtrado adaptivo).

Como en todos los sistemas, no sólo puede haber ventajas, el procesamiento digital de señales tiene también algunas limitantes:

- Velocidad y costo. Los diseños de procesamiento digital de señales pueden ser caros especialmente cuando un ancho de banda grande es involucrado en el diseño. Los circuitos integrados manejan un ancho de banda restringido de alrededor de los 100 MHz.
- Tiempo de diseño. Se necesitan conocimientos en las técnicas de procesamiento digital de señales y en ocasiones los recursos necesarios, aún con ellos el sistema debe ser analizado y sintetizado.
- Longitud de palabra finita. En situaciones de tiempo real, las consideraciones económicas con frecuencia limita que los algoritmos de procesamiento digital de señales sean implementados usando un número limitado de bits.

2.2 FORMAS DE REPRESENTACIÓN DE NÚMEROS BINARIOS

Existen distintas formas de representar a los números binarios, como se muestra en la tabla siguiente, donde los bits de magnitud el bit más a la izquierda será el bit que indique el signo de la cantidad formada por los tres bits más a la derecha:

Magnitud	Signo	Complemento a 1	Complemento a 2
0111	7	7	7
0110	6	6	6
0101	5	5	5
0100	4	4	4
0011	3	3	3
0010	2	2	2
0001	1	1	1
0000	0	0	0
1111	-7	-0	-1
1110	-6	-1	-2
1101	-5	-2	-3
1100	-4	-3	-4
1011	-3	-4	-5
1010	-2	-5	-6
1001	-1	-6	-7
1000	-0	-7	-8

Tabla 1 Formas de representación de números binarios.

Como se observa en la tabla 1, en las columnas de signo y en complemento a 1, esta forma de acomodar y representar los números binarios nos muestra que tenemos un problema de redundancia, ya que nos muestra que tenemos dos veces representado el cero, uno con signo positivo y otro con signo negativo, lo cual no es apropiado para algunos sistemas; para ello es más conveniente representarlos en formato complemento a dos.

El formato complemento a dos, se obtiene invirtiendo el patrón de bits de 1's por 0's y los 0's por 1's, a esa cantidad binaria se le suma un 1 binario.

2.3 ANÁLISIS DE FOURIER

La *Théorie analytique de la chaleur*, de Jean-Baptiste-Joseph Fourier [3], dio a conocer unos métodos sencillos para la solución de algunos problemas de valor en la frontera, que se presentan en el tratamiento analítico de la conducción (propagación y difusión [8]) del calor (Fourier estudiaba este tema para resolver un problema que tenían los cañones del ejército napoleónico). Sin embargo, este método de análisis se ha extendido a otras aplicaciones físicas diferentes a las del calor; ahora encontramos el análisis de Fourier en disciplinas de física moderna tales como teoría de comunicaciones, sistemas lineales, etc.

Serie de Fourier.

Es la representación matemática de una señal cuya función matemática cumple con ciertas características entre ellas, que sea periódica (Fig. 9); mediante una sumatoria de sus componentes sinusoidales y de corriente directa. La razón para analizar una señal mediante la serie de Fourier es conocer el espectro en frecuencia [4], [8].

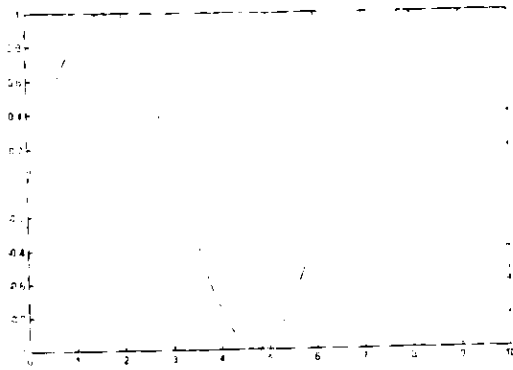


Fig. 9 Señal senoidal periódica.

$$F(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos nw_0 t + \sum_{n=1}^{\infty} B_n \operatorname{sen} nw_0 t$$

$$A_0 = \frac{1}{T} \int_0^T f(t) dt$$

$$A_n = \frac{2}{T} \int_0^T f(t) \cos nw_0 t dt$$

$$B_n = \frac{2}{T} \int_0^T f(t) \operatorname{sen} nw_0 t dt$$

Donde:

t es la variable independiente, la cual representa el tiempo.

$f(t)$ es una onda con variación de voltaje contra tiempo.

w_0 es conocida como la primera armónica o armónica fundamental, $w_0 = 2\pi/T$.

T es el periodo de repetición de la onda.

nw_0 son las n -ésimas armónicas de la frecuencia w_0 .

A_0 es la magnitud de la componente fundamental.

A_n es la magnitud de la n -ésima componente cosenoidal positiva.

B_n es la magnitud de la n -ésima componente senoidal positiva.

En particular, la representación de la serie de Fourier de una señal periódica en tiempo discreto es una serie finita [8]. Una señal en tiempo discreto $x[n]$ es periódica con periodo N si:

$$x[n] = x[n + N]$$

y su frecuencia fundamental es:

$$w_0 = \frac{2\pi}{N}$$

La expresión del par de la serie de Fourier es:

$$x[n] = \sum_k a_k e^{jkw_0 n} = \sum_k a_k e^{jk \left(\frac{2\pi}{N}\right) n}$$
$$a_k = \frac{1}{N} \sum_n x[n] e^{-jkw_0 n} = \frac{1}{N} \sum_n x[n] e^{-jk \left(\frac{2\pi}{N}\right) n}$$

a_k es el coeficiente espectral de $x[n]$. Estos coeficientes especifican una descomposición de $x[n]$ dentro de una suma de N exponenciales complejos armónicamente relacionados.

Ejemplo: Considere la señal:

$$x[n] = 1 + \operatorname{sen}\left(\frac{2\pi}{N}\right)n + 3 \cos\left(\frac{2\pi}{N}\right)n + \cos\left(\frac{4\pi}{N}n + \frac{\pi}{2}\right)$$

La señal es periódica con periodo N , y podemos expandir $x[n]$ directamente en términos de exponenciales complejos para obtener:

$$x[n] = 1 + \frac{1}{2j} \left[e^{j(2\pi/N)n} - e^{-j(2\pi/N)n} \right] + \frac{3}{2} \left[e^{j(2\pi/N)n} + e^{-j(2\pi/N)n} \right] \\ + \frac{1}{2} \left[e^{j(4\pi/N + \pi/2)} + e^{-j(4\pi/N + \pi/2)} \right]$$

Agrupando términos, encontramos que:

$$x[n] = 1 + \left(\frac{3}{2} + \frac{1}{2j} \right) e^{j(2\pi/N)n} + \left(\frac{3}{2} - \frac{1}{2j} \right) e^{-j(2\pi/N)n} + \\ \left(\frac{1}{2} e^{j\pi/2} \right) e^{j2(2\pi/N)n} + \left(\frac{1}{2} e^{-j\pi/2} \right) e^{-j2(2\pi/N)n}$$

Entonces los coeficientes de la serie de Fourier para este ejemplo son:

$$a_0 = 1$$

$$a_1 = \frac{3}{2} + \frac{1}{2j} = \frac{3}{2} - \frac{1}{2}j$$

$$a_{-1} = \frac{3}{2} - \frac{1}{2j} = \frac{3}{2} + \frac{1}{2}j$$

$$a_2 = \frac{1}{2}j$$

$$a_{-2} = -\frac{1}{2}j$$

Los resultados gráficos se muestran en la figura 10.

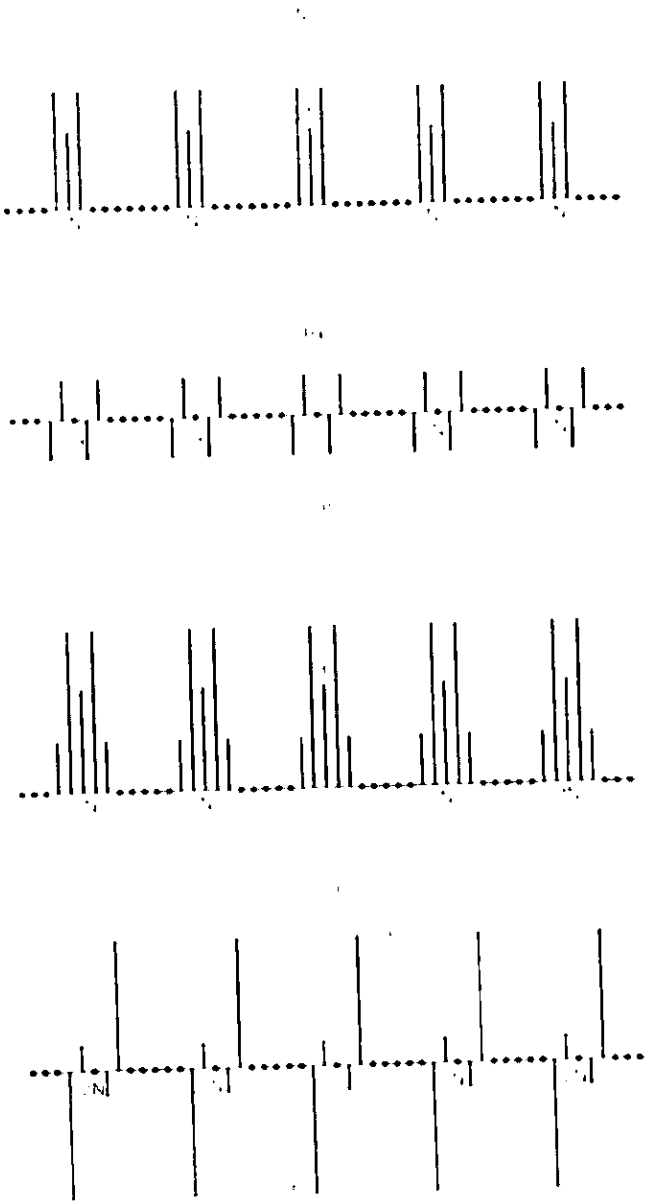


Fig. 10 (a) Parte real e imaginaria de los coeficientes del ejemplo, (b) magnitud y fase.

Transformada de Fourier.

Es una expresión matemática que se usa cuando la onda no es periódica [4] (Fig. 11), donde se ven a los coeficientes espectrales como muestras de una señal envolvente, cuya expresión es:

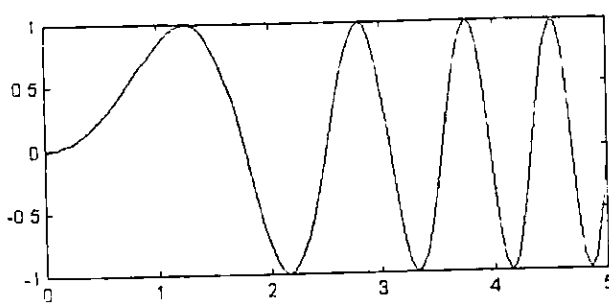


Fig. 11 Señal senoidal no periódica.

$$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt$$

Donde:

$e^{j\omega t}$ es una función exponencial.

$F(j\omega)$ es la integral de Fourier de una función continua multiplicada por una función exponencial y es compleja.

Si tenemos algunos datos digitales para ser transformados que son discretos y probablemente no periódicos, entonces no es posible aplicar la transformada de Fourier, ya que ésta es para datos continuos, entonces es necesario usar la transformada discreta de Fourier.

Transformada Discreta de Fourier.

La Transformada Discreta de Fourier (DFT por sus siglas en inglés), es la misma expresión matemática que la transformada de Fourier, sólo que ésta está restringida a muestras digitales. Se define como la secuencia de valores complejos en el dominio de la frecuencia. contiene componentes reales e imaginarios para las ésimas armónicas.

$$X(k) = F_D[x(nT)] = \sum_{n=0}^{N-1} x(nT)e^{-jk\Omega nT}$$

$k=0, 1, \dots, N-1$

Donde:

F_D denota la transformación discreta de Fourier.

k representa el número de armónica de la componente transformada.

$k\Omega$ son las armónicas de la señal discreta.

nT es el número de muestra.

Ejemplo: La secuencia {1,0,0,1} debe ser evaluada usando la ecuación de la transformada discreta de Fourier. Entonces tenemos que $N=4$, esto requiere encontrar los valores complejos $X(k)$ para $k=0, k=1, k=2, k=3$; con $k=0$, la evaluación es:

$$\begin{aligned} X(0) &= \sum_{n=0}^3 x(nT)e^{-j0} = \sum_{n=0}^3 x(nT) \\ &= x(0) + x(T) + x(2T) + x(3T) \\ &= 1 + 0 + 0 + 1 = 2 \end{aligned}$$

Con $k=1$, la evaluación es:

$$\begin{aligned}
 X(1) &= \sum_{n=0}^3 x(nT) e^{j\Omega nT} \\
 X(1) &= \sum_{n=0}^3 x(nT) e^{j(2\pi/\Lambda) nT} = \sum_{n=0}^3 x(nT) e^{j2\pi n/\Lambda} \\
 &= 1 + 0 + 0 + 1e^{-j2\pi 3/4} = 1 + e^{-j3\pi/2} \\
 &= 1 + \cos\left(\frac{3\pi}{2}\right) - j \operatorname{sen}\left(\frac{3\pi}{2}\right) = 1 + j
 \end{aligned}$$

Para $k=2$, la evaluación es.

$$\begin{aligned}
 X(2) &= \sum_{n=0}^3 x(nT) e^{-j2\Omega nT} = \sum_{n=0}^3 x(nT) e^{-j2n2\pi/\Lambda} = \sum_{n=0}^3 x(nT) e^{-j4\pi n/\Lambda} \\
 &= 1 + 0 + 0 + 1e^{-j4\pi 3/4} = 1 + 0 + 0 + e^{-j3\pi} = 1 - 1 = 0
 \end{aligned}$$

Para $k=3$, la evaluación es:

$$\begin{aligned}
 X(3) &= \sum_{n=0}^3 x(nT) e^{-j3n2\pi/\Lambda} \\
 &= 1 + 0 + 0 + e^{-j9\pi/2} = 1 - j
 \end{aligned}$$

Con los resultados llenamos una tabla (tabla 2), y tenemos las figuras (Fig. 12).

$X(k)$	Magnitud	Fase
0	2	0
1	$1+j$	45°
2	0	0
3	$1-j$	-45°

Tabla 2 Magnitud y fase del ejemplo de TDF.

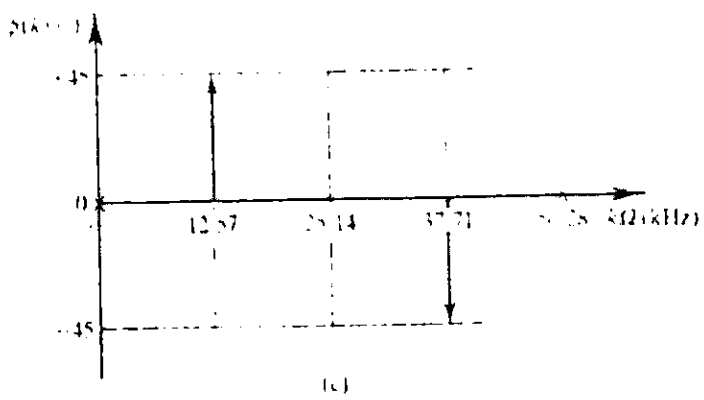
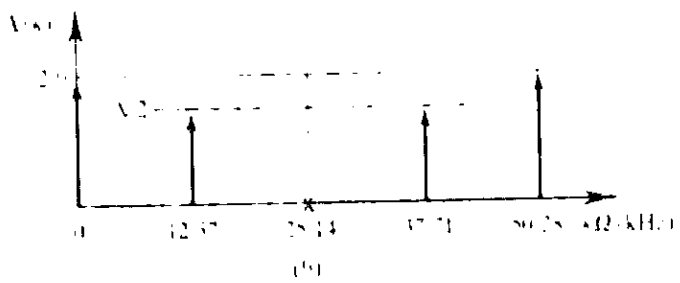
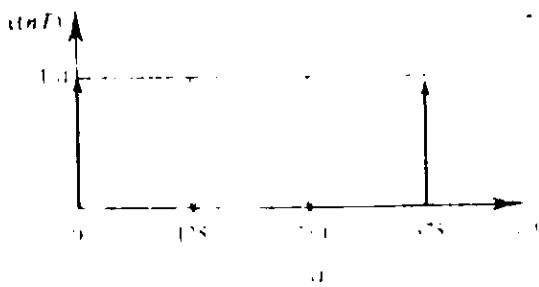


Fig. 12 (a) $x(nT)$ vs T . (b) $X(k)$ vs k (c) $\phi(k)$ vs k .

Transformada Rápida de Fourier.

En 1965 Cooley y Tukey desarrollaron una serie de algoritmos que permiten reducir el número de operaciones necesarias para calcular la transformada discreta de Fourier: conocidos como la transformada rápida de Fourier (FFT por sus siglas en inglés) [4]. Los algoritmos más rápidos los tenemos cuando $N = 2^l$. Existen algoritmos básicos que son:

- a) Decimación en Tiempo Radix-2, y
- b) Decimación en Frecuencia Radix-2.

El término *decimación* se refiere a la reducción significativa en el número de cálculos efectuados en datos en dominio del tiempo. El término *radix-2* se refiere a que la operación básica para evaluar este algoritmo (operación mariposa, *vide infra*) acepta dos valores a la entrada.

Direccionamiento Bit Invertido (*Bit-Reversi*).

Algoritmo que se consiste en una secuencia de multiplicaciones complejas, donde la serie de datos discretos, tomemos: 0, 1, 2, 3 se acomodan en el orden: 0, 2, 1, 3; esto se debe a que el orden de los bits se invierte, cuando tenemos: 00, 01, 10, 11, al direccionarlos en bit invertido tendremos: 00, 10, 01, 11 [4]. Esta secuencia se multiplicará por un factor w_N . En la tabla 3 se observa el orden en que quedan para $N=8$.

X(0)	000	→	000	x(0)
X(1)	001	→	100	x(4)
X(2)	010	→	010	x(2)
X(3)	011	→	110	x(6)
X(4)	100	→	001	x(1)
X(5)	101	→	101	x(5)
X(6)	110	→	011	x(3)
X(7)	111	→	111	x(7)

Tabla 3 Direccionamiento bit invertido.

El algoritmo de decimación en tiempo *radix-2* se basa en el empleo de la operación conocida como mariposa para dos puntos (Fig. 13).

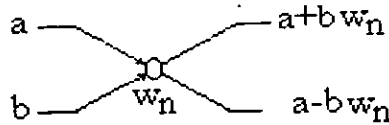


Fig. 13 Operación "mariposa" para decimación en tiempo.

El número de puntos al que se le va a aplicar el esquema de operación mariposa debe ser un número de la forma $N=2^L$.

1. El número de iteraciones es igual a L .
2. El número de mariposas por iteración es el mismo, es decir 2^{L-1} .
Si i es un contador de la iteración, $i=0,1,2,\dots,L-1$.
3. La cantidad de mariposas por grupo es 2^i .
4. La cantidad de grupos por iteración es 2^{L-1-i} .
5. La separación entre las cantidades a y b de cada mariposa es una determinada iteración es 2^i .
6. El incremento del índice del coeficiente w_v en cada grupo es 2^{L-1-i} , la primer mariposa de cada grupo tiene coeficiente w_0 .
La secuencia de entrada estará en orden bit invertido, y la salida tendrá orden creciente normal.

Los coeficientes tienen la forma: $w_n = e^{j \frac{2\pi}{N} n}$

Para la transformada rápida inversa (IFFT por sus siglas en inglés), se tienen las mismas consideraciones anteriores de la transformada rápida.

Para la evaluación de la transformada rápida inversa, se tiene:

1. $w_n = e^{j \frac{2\pi}{N} n}$
2. El resultado final se divide entre N.

Para el algoritmo de decimación en frecuencia *radix-2* para dos puntos, la operación mariposa se muestra en la figura 14.

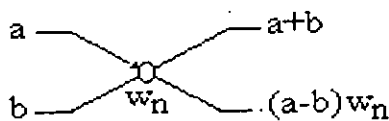


Fig. 14 Operación "mariposa" para decimación en frecuencia.

Para este algoritmo la secuencia de entrada debe de estar en orden normal, y la secuencia de salida estará en orden bit invertido. El índice i , adquirirá los valores: $i=L-1, \dots, 2, 1, 0$.

Un diagrama de flujo que indica el algoritmo para encontrar la secuencia bit invertido se presenta en la figura 15.

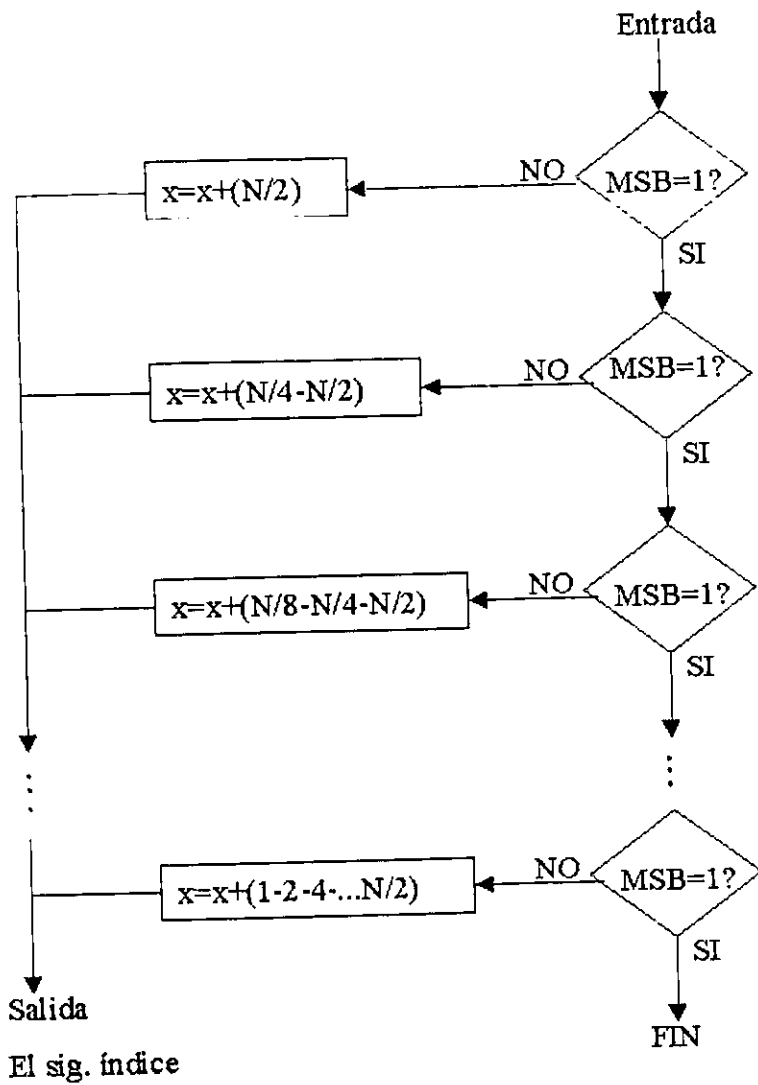


Fig. 15 Algoritmo para encontrar la secuencia bit invertido.

Ejemplo: Construir el diagrama de mariposas para N=8 y calcular los valores de los coeficientes en DIF. Primero calculamos los valores para W_N para las mariposas:

$$W_0 = e^{j \frac{2\pi}{8} (0)} = 1$$

$$W_1 = e^{j \frac{2\pi}{8} (1)} = \frac{1}{2} - \frac{j}{2}$$

$$W_2 = e^{-j \frac{2\pi}{8} (2)} = -j$$

$$W_3 = e^{-j \frac{2\pi}{8} (3)} = -\frac{1}{2} - \frac{j}{2}$$

Entonces, se tiene el esquema (Fig. 16).

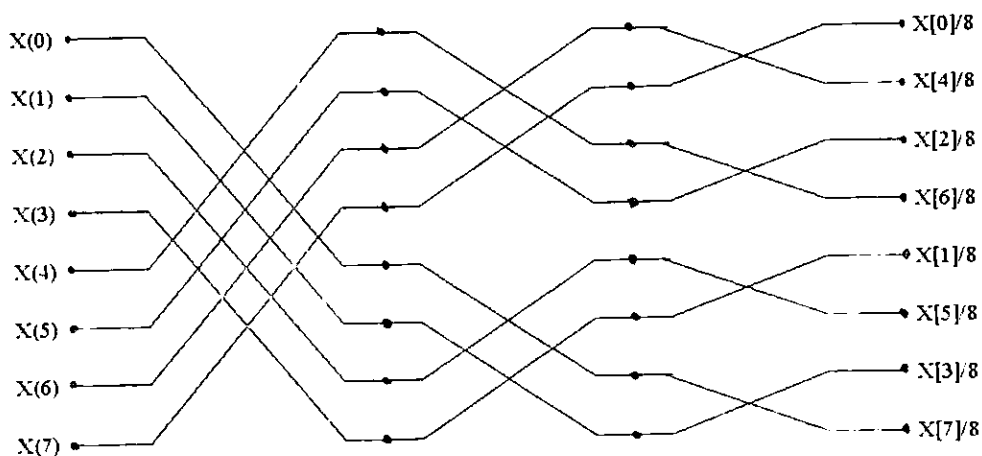


Fig. 16 Diagrama mariposa para N= 8.

Con estos resultados se construye la tabla (tabla 4).

$x[n]$	$X[n]$
0	2
1	3
2	4
3	5
4	6
5	5
6	4
7	3

Tabla 4 Valores de secuencia y su resultado por FFT.

2.4 TRANSFORMADA Z.

La Transformada z de una secuencia $x[n]$, la cual es válida para toda n [4], es definida como:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

Donde:

z es una variable compleja ($z = re^{j\omega}$) [8]. Con r como la magnitud de z y ω como el ángulo de z . En sistemas causales, $x[n]$ puede ser no cero sólo en el intervalo $0 < n < \infty$ y la ecuación anterior se reduce en su tamaño llamada transformada z de un lado:

$$X(z) = \sum_{n=0}^{\infty} x[n]z^{-n}$$

La transformada z es una serie de potencias con un número infinito de términos y que pueden no converger para todos los valores de z . La región donde la transformada z converge es conocida como la *región de convergencia* (ROC por sus siglas en inglés), y en esta región los valores de $X(z)$ son finitos. La región de convergencia es determinada por las propiedades de $x(n)$.

La transformada z se reduce a la transformada de Fourier cuando la magnitud de la variable de transformación z es unitaria, entonces la transformada z reduce a la transformada de Fourier sobre el contorno en el plano complejo z correspondiendo a un círculo con un radio unitario (Fig. 17); este círculo en el plano z es referido como el círculo unitario.

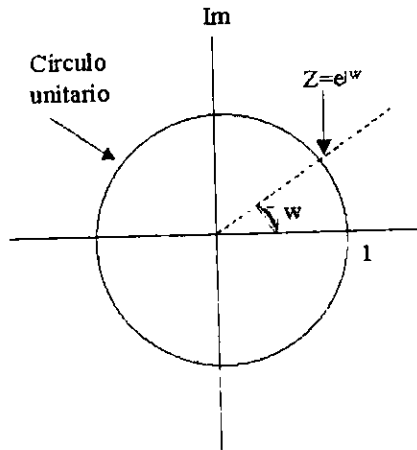


Fig. 17 Círculo unitario en el plano z .

La Transformada z Inversa.

La transformada z inversa (IZT por sus siglas en inglés), nos permite recuperar la secuencia en tiempo discreto $x(n)$, la IZT es útil en trabajos de procesamiento digital de señales, por ejemplo, en encontrar la respuesta impulso de filtros digitales. Simbólicamente, la transformada z inversa puede ser definida como:

$$x(n) = Z^{-1}[X(z)]$$

Donde:

$X(z)$ es la transformada z de $x(n)$

Z^{-1} es el símbolo para la transformada z inversa.

Tomando una secuencia causal, la transformada z $X(z)$, puede ser expandida en una serie de potencias como:

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n} = x(0) + x(1)z^{-1} + x(2)z^{-2} + \dots$$

En la ecuación anterior se observa que los valores de $x(n)$ son los coeficientes de z^{-n} ($n=0,1,\dots$) y como pueden ser obtenidos directamente por inspección. En la práctica, $X(z)$ es frecuentemente expresado como una relación de dos polinomios en z^{-1} o equivalentemente en z :

$$X(z) = \frac{a_0 + a_1z^{-1} + a_2z^{-2} + \dots + a_Nz^{-N}}{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Mz^{-M}}$$

De esta forma, la transformada z inversa, $x(n)$ puede ser obtenida usando uno de estos métodos:

A) Método de expansión de series de potencias.

Dada la transformada z , $X(z)$, de una secuencia causal como en la ecuación anterior, esta puede ser expandida en una serie infinita en z^{-1} o z por división extendida (algunas veces llamada división sintética):

$$\begin{aligned} X(z) &= \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}} \\ &= x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \dots \end{aligned}$$

En este método el numerador y denominador de $X(z)$ son primero expresados o en potencias descendientes de z o potencias ascendentes de z^{-1} y el cociente es entonces obtenido por división extensa.

B) Método de fracciones parciales.

En este método, la transformada z es primero expandida en una suma de fracciones parciales. La transformada inversa z de cada fracción parcial es entonces obtenida desde tablas, entonces se suman todos para obtener la transformada inversa. En muchos casos prácticos, la transformada z es dada como una relación de polinomios en z o z^{-1} y tiene la forma:

$$X(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}$$

Si los polos de $X(z)$ son de primer orden y $N \geq M$, entonces $X(z)$ puede ser expandido como:

$$\begin{aligned}
 X(z) &= B_0 + \frac{C_1}{1-p_1 z^{-1}} + \frac{C_2}{1-p_2 z^{-1}} + \dots + \frac{C_M}{1-p_M z^{-1}} \\
 &= B_0 + \frac{C_1 z}{z-p_1} + \frac{C_2 z}{z-p_2} + \dots + \frac{C_M z}{z-p_M} \\
 &= B_0 + \sum_{k=1}^M \frac{C_k z}{z-p_k}
 \end{aligned}$$

donde p_k son los polos de $X(z)$, C_k son los coeficientes de las fracciones parciales y $B_0 = a_N/b_N$; C_k es también conocido como el residuo de $X(z)$. Si el orden del numerador es menor que el del denominador en la primera ecuación, esto es que $N < M$, entonces B_0 debe ser cero. Si $N > M$ entonces $X(z)$ debe ser reducido primero, para hacer $N \leq M$, por división extendida con polinomios del numerador y denominador escritos en potencias descendientes de z^{-1} . El coeficiente, C_k , asociado con el polo p_k debe ser obtenido por multiplicar ambos lados de la ecuación anterior por $(z - p_k) / z$ y entonces teniendo $z = p_k$:

$$C_k = \lim_{z \rightarrow p_k} \frac{X(z)}{z} (z - p_k)$$

Si $X(z)$ contiene uno más polos múltiples entonces se necesitan términos extras para tenerlos en cuenta. Por ejemplo, si $X(z)$ contiene un polo de m -ésimo orden a $z = p_k$ la expansión de la fracción parcial debe incluir términos de la forma:

$$\sum_{i=1}^m \frac{D_i}{(z - p_k)^i}$$

Los coeficientes, D_i , pueden ser obtenidos de la relación:

$$D_i = \frac{1}{(m-i)!} \frac{d^{m-i}}{dz^{m-i}} \left[(z - p_k)^m X(z) \right]_{z=p_k}$$

C) Método del residuo.

En este método la IZT es obtenida por evaluación de la integral cerrada

$$x(n) = 1 \int_{C'} z^{n-1} X(z) dz$$

donde C' es la forma de la integración cerrada de todos los polos de $X(z)$. Para polinomios racionales, la integral cerrada es evaluada usando un resultado fundamental en teoría de variable compleja conocida como teorema del residuo de Cauchy:

$$x(n) = 1 \int_{C'} z^{n-1} X(z) dz$$

es suma de residuos de $z^{n-1}X(z)$ para todos los polos dentro de C' .

En la última sección, establecimos que los coeficientes de fracciones parciales, los C_k , son también referidos a todos los residuos de $X(z)$ y una manera de obtener esos valores. El punto clave para recordar es que cada residuo, C_k , está asociado con un polo, p_k . En el presente método, el residuo de $z^{n-1}X(z)$ al polo p_k (no el residuo de $X(z)$) está dado por:

$$\text{Res}[F(z), p_k] = \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} [(z-p_k)F(z)]_{z=p_k}$$

donde $F(z) = z^{n-1}X(z)$, m es el orden del polo a p_k y $\text{Res}[F(z), p_k]$ es el residuo de $F(z)$ para $z = p_k$. Para un simple (distinto) polo; entonces la ecuación se reduce a:

$$\begin{aligned} \text{Res}[F(z), p_k] &= (z-p_k)F(z) \\ &= (z-p_k)z^{n-1}X(z)|_{z=p_k} \end{aligned}$$

2.5 CORRELACIÓN Y CONVOLUCIÓN

Correlación.

La *correlación* es una medida de la similitud o la dependencia que existe entre dos señales o procesos [4]. Se define por:

$$\Phi_{hx} = \sum_{k=-x}^x x[k]h[k]$$

Donde:

Φ_{hx} es la correlación de dos secuencias.

$x[k]$ es una secuencia de valores discretos.

$h[k]$ es otra secuencia de valores discretos.

Si las dos formas de onda digitalizadas varían similarmente punto a punto, entonces una medida de su correlación puede ser obtenida tomando la suma de sus productos de sus correspondientes pares de puntos. La existencia de una suma finita puede indicar un grado de correlación. Una suma negativa puede indicar correlación negativa, que es un aumento en una variable es asociada con un decremento en otra variable. Cuando la correlación se aplica a 2 secuencias distintas, le llamamos correlación cruzada, que se define por:

$$\Phi_{hx} = \sum_{k=-x}^x x[k]h[n+k]$$

$$\Phi_{xh} = \sum_{k=-x}^x h[k]x[n+k]$$

La correlación no cumple con la propiedad conmutativa. Cuando la correlación se aplica sobre una misma secuencia, se llama autocorrelación, que es definida por:

$$\Phi_{hh} = \sum_{k=-r}^r h[k]h[n+k]$$

$$\Phi_{xx} = \sum_{k=-r}^r x[k]x[n+k]$$

Ejemplo: Dadas la secuencias $x=\{3,3,1\}$ y $h=\{2,2\}$, encontrar los cuatro casos de correlación: Φ_{hx} , Φ_{xh} , Φ_{xx} , Φ_{hh} .

$$\text{Para } \Phi_{hx} = \sum_{k=-r}^{\infty} x[k]h[n+k]$$

$$\Phi_{hx}[0] = 6 + 0 = 6$$

$$\Phi_{hx}[1] = 6 + 6 = 12$$

$$\Phi_{hx}[2] = 6 + 2 = 8$$

$$\Phi_{hx}[3] = 2 + 0 = 2$$

$$\Phi_{hx}[4] = 0$$

El resultado es: $\Phi_{hx} = [6 \ 12 \ 8 \ 2]$, donde $n=1$ tiene la máxima correlación.

$$\text{Para } \Phi_{vh} = \sum_{k=-r}^r h[k]x[n+k]$$

$$\Phi_{vh}[0] = 2 + 0 + 0 = 2$$

$$\Phi_{vh}[1] = 2 + 6 + 0 = 8$$

$$\Phi_{vh}[2] = 0 + 6 + 6 = 12$$

$$\Phi_{vh}[3] = 0 + 0 + 6 = 6$$

$$\Phi_{vh}[4] = 0$$

El resultado es: $\Phi_{vh} = [2 \ 8 \ 12 \ 6]$, donde $n=2$ tiene la máxima correlación.

$$\text{Para } \Phi_{xx} = \sum_{k=-r}^r x[k]x[n+k]$$

$$\Phi_{xx}[0] = 3 + 0 + 0 = 3$$

$$\Phi_{xx}[1] = 3 + 9 + 0 = 12$$

$$\Phi_{xx}[2] = 9 + 9 + 1 = 19$$

$$\Phi_{xx}[3] = 9 + 3 + 0 = 12$$

$$\Phi_{xx}[4] = 3 + 0 + 0 = 3$$

$$\Phi_{xx}[5] = 0$$

El resultado es: $\Phi_{xx} = [3 \ 12 \ 19 \ 12 \ 3]$, donde $n=2$ tiene la máxima autocorrelación.

$$\text{Para } \Phi_{hh} = \sum_{k=-r}^r h[k]h[n+k]$$

$$\Phi_{hh}[0] = 4 + 0 = 4$$

$$\Phi_{hh}[1] = 4 + 4 = 8$$

$$\Phi_{hh}[2] = 0 + 4 = 4$$

$$\Phi_{hh}[3] = 0$$

El resultado es: $\Phi_{hh} = [4 \ 8 \ 4]$, donde $n=1$ tiene la máxima autocorrelación.

Convolución.

El término *convolución* se describe como la entrada a un sistema que opera con el sistema para producir la salida. Generalmente la salida del sistema puede ser retardada y atenuada o una versión amplificada de la entrada. Esto es útil para considerar la salida de un sistema debido a una entrada impulso. Esto es porque cualquier entrada puede ser representada como una secuencia de impulsos de diferente longitud.

En un sistema LTI en tiempo discreto [8], se tienen una entrada, una señal impulso y una salida, esto se define:

$$y[n] = x[n] * h[n]$$

Ejemplo: Encontrar la convolución de las secuencias $x=[1 \ 1 \ 1]$ y $h=[0.5 \ 2]$

Para resolver este ejemplo primero invertimos la secuencia $h[n]$ y tenemos: $h=[2 \ 0.5]$, entonces se aplica la expresión:

$$\begin{aligned}y[0] &= 0.5 + 0 = 0.5 \\y[1] &= 0.5 + 2 = 2.5 \\y[2] &= 0.5 + 2 = 2.5 \\y[3] &= 0 + 2 = 2\end{aligned}$$

Esto se puede probar en MatLab® con las siguientes instrucciones:

```
>> h = [0.5 2];  
>> x = [1 1 1];  
>> y = conv(h, x);  
>> stem(y)
```

Y el resultado se puede ver en la figura, (Fig. 18).

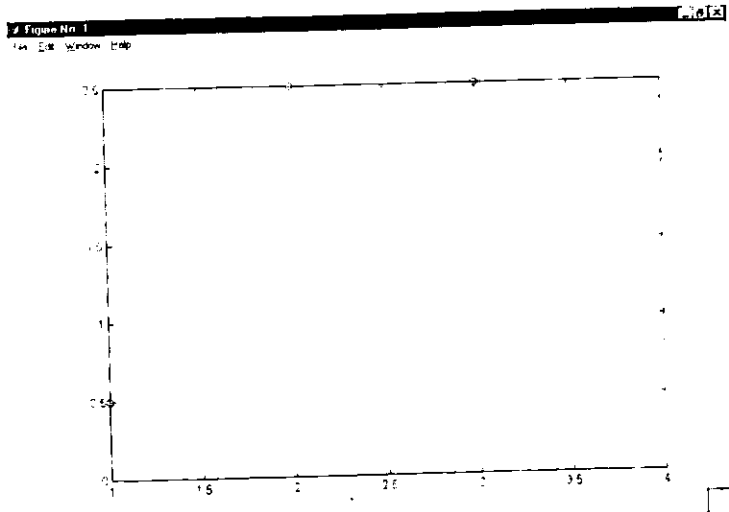


Fig. 18 Convolución de $x[n]$ y $h[n]$.

El algoritmo para efectuar la convolución rápida es:

1. Aumentar el soporte de las secuencias con ceros hasta completar un número potencia de 2.
2. Obtener la transformada de Fourier de ambas secuencias.
3. Se multiplican las dos secuencias anteriores.
4. Al resultado se le aplica la transformada de Fourier inversa.

En este capítulo se revisó las ventajas que se obtienen de procesar la información en forma digital, también se observó la forma en que los números binarios se pueden representar; y lo más importante de este capítulo fue el repaso del análisis de Fourier que es una herramienta fundamental en el diseño de sistemas de procesamiento digital de señales. En el capítulo siguiente se muestra el diseño del filtro digital.

CAPÍTULO

3

DISEÑO DEL FILTRO DIGITAL

En este capítulo se hace el proceso de diseño del filtro digital tipo FIR, también se muestran los bloques que componen al filtro y la función que desempeñan cada uno de ellos. Diseñar un filtro consiste en calcular los coeficientes para multiplicar los valores digitales de una señal y al sumarlas, entonces se tendrá un proceso de filtrado. El filtro FIR tiene una función de transferencia que es un polinomio en z^{-1} y esto es un filtro de todos ceros en el sentido de que los ceros en el plano z determinan la característica de magnitud de respuesta en frecuencia. Aunque un filtro FIR de grado N tiene un polo de orden $N-1$ al origen del plano z , un polo en el origen no afecta la magnitud de la respuesta en frecuencia del filtro [10]. Un filtro FIR puede tener una respuesta al impulso unitario que es simétrico alrededor del punto $(N-1)/2$ y puede por lo tanto tener fase lineal exacta. Al no tener recursividad, el filtro se mantendrá estable. El filtro FIR requiere de más recursos de hardware para tener una respuesta similar a los filtros IIR. Para aplicaciones donde la forma de onda es importante, el filtro FIR con sus características de fase es usualmente una buena selección.

3.1 DIAGRAMA A BLOQUES DEL FILTRO

Un filtro digital al operar sobre una secuencia de números digitales, va a ejecutar operaciones matemáticas básicas tales como: suma, multiplicación y retardos, de los cuales los sumadores son circuitos fáciles de diseñar y construir (con acarreo o sin ello); los multiplicadores se diseñan de acuerdo a sus necesidades (con/sin signo, operación en un solo pulso de reloj o en varios de forma secuencial); casi todos ellos se pueden implementar en circuitos digitales para llevar a cabo sus pruebas.

Al calcular los coeficientes con ayuda del programa de cálculos computacionales denominado MatLab®¹, versión 5.2² para el filtro pasa-bajas tipo FIR con aproximación butterworth (ver anexo A1), donde el programa de cómputo tiene la siguiente instrucción:

$$B = \text{FIR1}(W_N, N);$$

Se introducen los valores calculados previamente como:

$$W_N = 8 \quad \text{y} \quad N = 2$$

Entonces se teclea:

$$B = \text{FIR1}(8, 2)$$

Y se obtiene la salida que son los coeficientes:

$$B = 0.5 \quad 0.5$$

Estos coeficientes se deben transformar en un radix-3, con la finalidad de usar un formato de número entero positivo y éste número se pueda introducir en binario en la parte del multiplicador del diseño esquemático del circuito, así:

$$0.5 \times 2^3 = 4$$

¹ www.mathworks.com

² Este programa de cómputo es de gran ayuda para aquellos que utilicen cálculos matemáticos iterativos o de gran complejidad y por lo tanto se requiere de la ayuda de una computadora, el cálculo de coeficientes mediante transformada de fourier a mano puede llevar casi una hora y pueden existir algunos errores; mediante el algoritmo mariposa este tiempo se reduce a 30 minutos y también se reduce el porcentaje de errores; con ayuda del programa fir1 de matlab, este tiempo se reduce a menos de 5 minutos y con un bajo porcentaje de errores.

Así, se tiene conocimiento que los datos de entrada van a multiplicarse por cuatro en el bloque del multiplicador. Se tiene entonces la ecuación en diferencias que va a resolver el sistema es del tipo:

$$y[n] = x[n] + a_1x[n-1] + a_2x[n-2]$$

Donde:

$y[n]$ es la señal digital de salida

$x[n]$ es la señal digital de entrada

$x[n-1]$ es la señal digital de entrada retardada un ciclo de reloj

$x[n-2]$ es la señal digital de entrada retardada dos ciclos de reloj

a_1 y a_2 son los coeficientes calculados para el filtro.

En la figura 19 se muestra la forma en que el sistema debe de colocarse para ejecutar la operación de filtrado [11], según la ecuación anterior; el multiplicador se muestra como un símbolo de amplificación (\triangleright), el sumador se identifica con el símbolo aritmético de suma (+) y el retardo se identifica como un bloque de retención (z^{-1}).

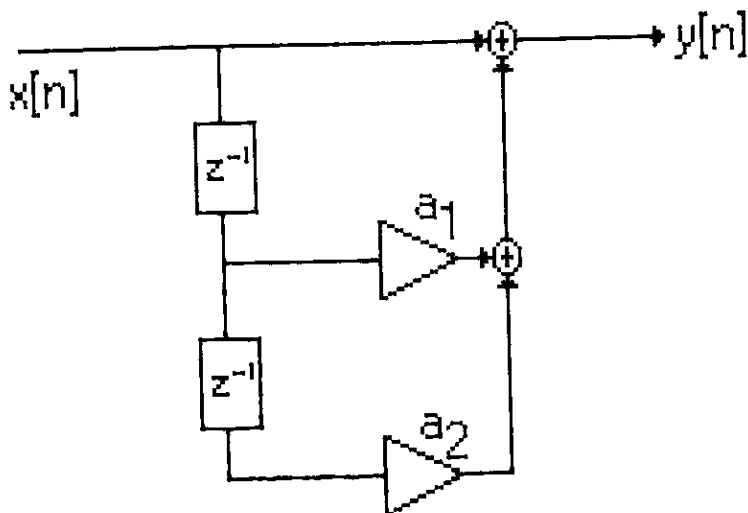


Fig. 19 Diagrama a bloques del filtro digital.

Donde en el primer bloque de retardo se retrasa la señal de entrada, entonces, se tiene: $x[n-1]$, esta señal pasa por el primer multiplicador donde se multiplica con el coeficiente a_1 y la señal ahora es: $a_1 x[n-1]$; donde se suma con la señal de los bloques inferiores que están formados por la señal que sale del retardo inferior: $x[n-2]$, que al pasar a través del multiplicador inferior se convierte en: $a_2 x[n-2]$; estas dos señales retardadas y multiplicadas se suman también con la señal presente en las entradas del filtro, lo cual forma la señal: $y[n] = x[n] + a_1 x[n-1] + a_2 x[n-2]$, que es la solución idéntica a la ecuación en diferencias vista anteriormente (*vide supra*).

A la ecuación propuesta se le aplica la transformada de fourier discreta, a fin de calcular la respuesta en frecuencia del filtro digital.

$$y[n] = x[n] + a_1x[n-1] + a_2x[n-2]$$

Aplicando la transformada de fourier:

$$Y[\Omega] = X[\Omega] + a_1e^{-j\Omega}X[\Omega] + a_2e^{-j\Omega^2}X[\Omega]$$

Reordenando:

$$Y[\Omega] = X[\Omega][1 + a_1e^{-j\Omega} + a_2e^{-j\Omega^2}]$$

Acomodando para obtener la función de transferencia:

$$H[\Omega] = \frac{Y[\Omega]}{X[\Omega]} = 1 + a_1e^{-j\Omega} + a_2e^{-j\Omega^2}$$

$$\Omega = \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi\}$$

Evaluando para cada valor de omega y con los valores de coeficientes igual a 0.5, se llena una tabla (tabla 5) y se tiene la gráfica de respuesta en frecuencia (Fig. 20).

Valor	Magnitud
0	2
$\pi/4$	1.8477
$\pi/2$	1.4142
$3\pi/4$	0.7653
π	1

Tabla 5 Valuación de π y magnitud de la función de transferencia.

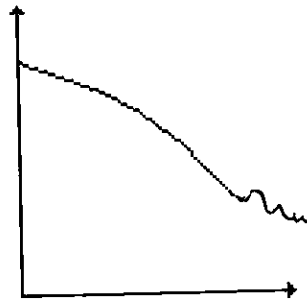


Fig. 20 Respuesta en frecuencia del filtro.

3.2 ARQUITECTURA DEL SUMADOR

En la adición binaria se deben cumplir con las reglas de suma binaria y hay que tener en cuenta que cuando exista una suma de $1+1$, el resultado será 0 con el acarreo de 1 lógico hacia el bit mas significativo.

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0$$

El sumador es un circuito combinacional que considera a tres bits de entrada, dos de sumandos (a y b) y un acarreo de entrada (c_i); asimismo tiene como salidas la suma (s) y un acarreo de salida (c_o), esto es válido para la suma de un solo bit [6] [16] y [9], a continuación se presenta el diagrama esquemático (Fig. 21), la tabla de verdad (tabla 6) y su símbolo lógico (Fig. 22).

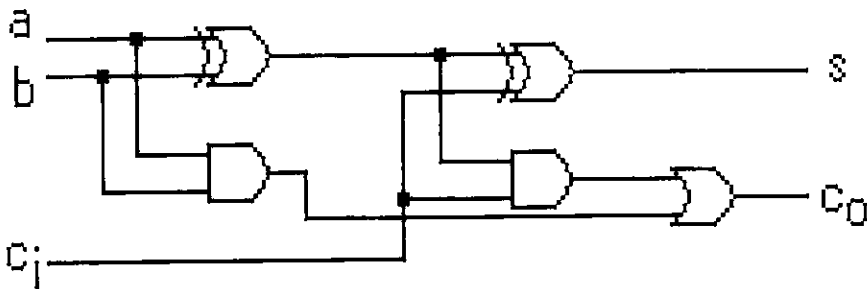


Fig. 21 Diagrama lógico de un sumador completo

a	B	c_i	s	c_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabla 6 Tabla de verdad para el sumador.

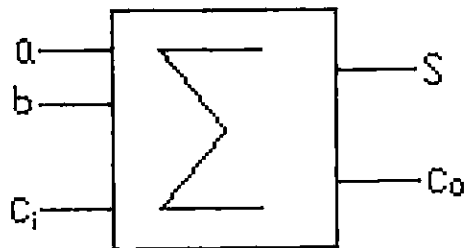


Fig. 22 Símbolo lógico del sumador.

3.3 ARQUITECTURA DEL MULTIPLICADOR

En sistemas de signo y magnitud, el bit más significativo representa el signo del número, mientras que los bits restantes representan el valor absoluto como una magnitud sin signo [5], [16] y [9]. Si el bit de signo es cero, el número es positivo; si el bit de signo es uno, el número es negativo (*vide supra* capítulo 2, inciso 2.2). Las reglas de multiplicación binaria son bastante simples, son semejantes a la multiplicación decimal, estas reglas se muestran a continuación:

$$0 * 0 = 0$$

$$0 * 1 = 0$$

$$1 * 0 = 0$$

$$1 * 1 = 1$$

Para el diseño del multiplicador se utiliza un bloque de construcción fundamental (Fig. 23 (a)), donde se observa que es un circuito sumador (*vide supra*) al cual se le ha agregado una compuerta AND, así se obtiene el producto parcial de cada par de bits a multiplicar; con este bloque se construye cada producto parcial que se va acumulando y los acarreos se van sumando y propagando; esta arquitectura (fig. 23 (b)) como se observa, está compuesta de bloques combinatoriales y hace la operación de multiplicación en un solo ciclo de reloj, esto es importante a considerar en el diseño, ya que existen otros diseños de circuitos multiplicadores pero con la desventaja de que efectúan la operación de multiplicación en cuatro o más ciclos de reloj. También se muestra su tabla de operación (Tabla 7) y su símbolo lógico (Fig. 24).

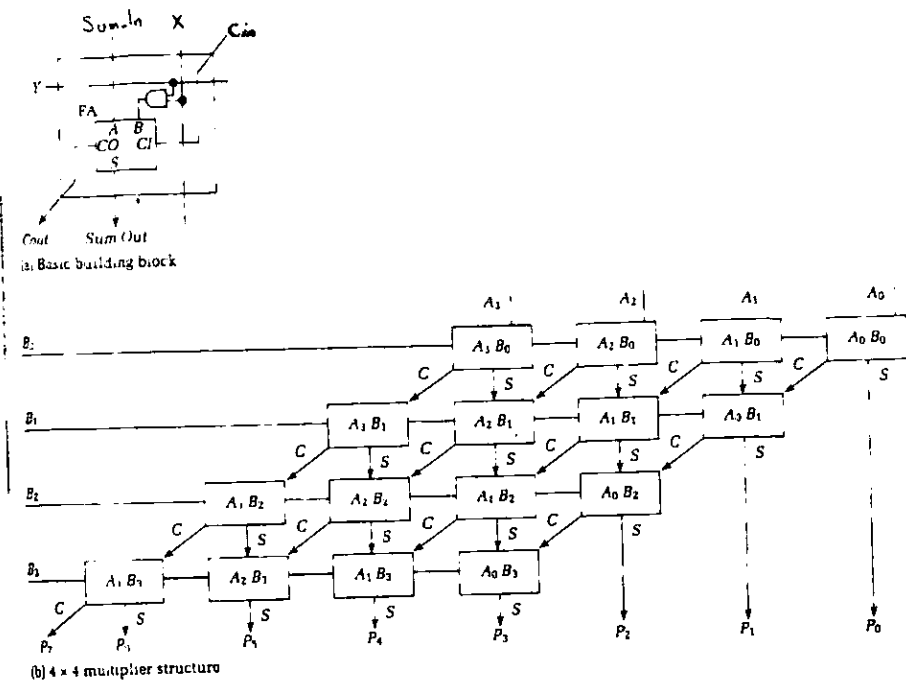


Fig 23 (a) bloque de construcción básico y (b) Arquitectura del multiplicador.

X	Y	Sum_in	Ci	Sum_out	Co
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	1

Tabla 7 Tabla de verdad para el multiplicador.

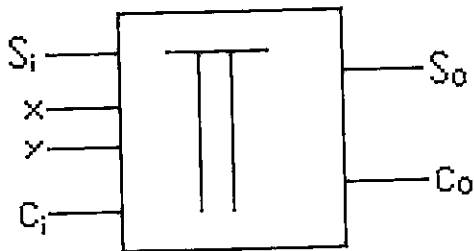


Fig 24 Símbolo lógico del multiplicador.

3.4 EL RETARDO

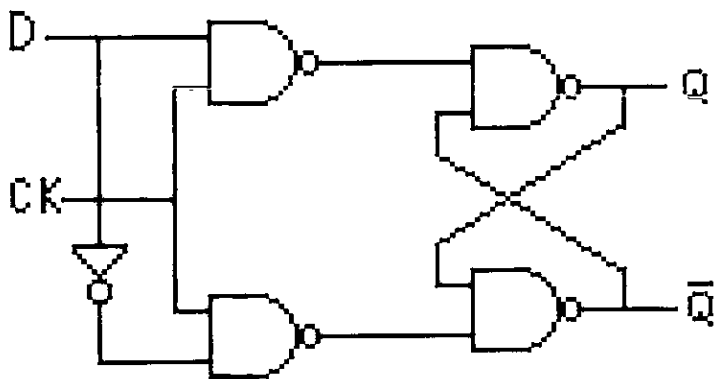


Fig. 25 Diagrama lógico del flip-flop tipo D.

El elemento de retardo [6] [16] para un filtro, es simplemente un flip-flop tipo D (Fig. 25) ya que nos permite retardar la señal presente un ciclo de reloj, según su tabla de verdad (Tabla 8); de aquí se desprende que el sistema debe de funcionar bajo los tiempos marcados por una señal maestra o mejor dicho sincronizados con un reloj general; asimismo tiene la opción de habilitar el dispositivo mediante una terminal y con otra terminal se inicializa el dispositivo (Fig. 26).

D	Q	Q+1
0	0	0
0	1	0
1	0	1
1	1	1

Tabla 8 Tabla de verdad para el flip-flop tipo D.

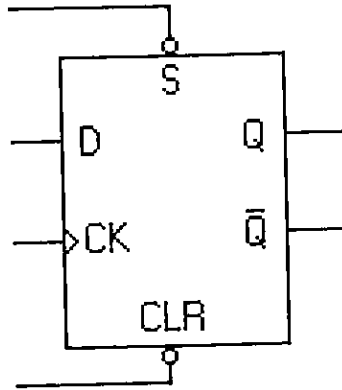


Fig. 26 Símbolo lógico para un flip-flop tipo D con entradas de disposición y limpieza.

En este capítulo se efectuó el diseño del filtro digital tipo FIR de acuerdo a la ecuación en diferencias propuesta. Así, también se presentaron los elementos que componen al filtro y como se llevó a cabo el diseño de ellos. cabe aclarar que el diseño del sumador y el flip-flop al ser ya tan conocidos se usan directamente de cómo han sido mostrados en diversos cursos de electrónica digital a lo largo de la formación profesional.

CAPÍTULO

4

ARREGLO DE COMPUERTAS PROGRAMABLES EN CAMPO

“FIELD PROGRAMMABLE GATE ARRAY”

En este capítulo revisaremos la evolución tecnológica de los dispositivos programables y en particular se muestra la estructura del arreglo de compuertas programable en campo.

Dispositivos Programables.

La tecnología de muy alta escala de integración (VLSI por sus siglas en inglés) permite fabricar componentes de los cuales hace 50 años no se tenía idea de su posibilidad de existencia; esta tecnología permitió la fabricación de microprocesadores en un circuito integrado, y también permite ahora la fabricación de circuitos de aplicación específica y otros a medida [1]. En la evolución de los circuitos programables comenzó por vez primera con las Memorias de Sólo Lectura (PROM por sus siglas en inglés), estas memorias se programaban una sola vez y podían programarse con una máscara durante el proceso de su fabricación o por el usuario. Una continuación de memorias programables más versátiles son las Memorias de Sólo Lectura Borrables¹ (EPROM por sus siglas en inglés) que se pueden programar con ayuda de equipo grabador de memorias y borrarse con exponer su ventana durante un tiempo a rayos ultravioletas. Más versátiles aún, son las Memorias de Sólo Lectura Borrables Eléctricamente (EEPROM por sus siglas en inglés) las cuales se programan y se borran por el usuario. Pero existía la inquietud de integrar en un circuito programable varios dispositivos de alta escala de integración (LSI por sus siglas en inglés) y que cumpliera ciertas funciones lógicas predeterminadas por el usuario. Aquí es donde aparecen los Dispositivos Lógicos Programables (PLD por sus siglas en inglés) de los cuales los primeros en salir al mercado fueron la Lógica de Arreglos Programable (PAL por

¹ Teóricamente la programación y borrado de estas memorias llega a un número infinito, pero en la práctica esto no se cumple, se ha comprobado que la mayoría de estas memorias llegan alrededor de 100 procesos consecutivos de programación y borrado pasando este número los datos pueden degenerarse y entregar información errónea a las salidas del dispositivo programable.

sus siglas en inglés), este consiste de un plano AND programable seguido de un plano OR fijo. las salidas de las compuertas OR pueden ser del tipo combinacional o tipo registro. la programación se hace siguiendo una fundición de fusibles. Una versión más flexible fueron los Arreglos Lógicos Programables (PLA por sus siglas en inglés). este dispositivo tiene la característica de tener un plano AND seguido de un plano OR ambos programables. estos dispositivos están disponibles comercialmente para ser programados por máscara o por el usuario. Semejantes y contemporáneas son la Lógica de Arreglos Genéricos (GAL por sus siglas en inglés), las cuales a diferencia de las PAL's pueden ser reprogramadas [15]. Posteriormente con el avance de la tecnología de fabricación de dispositivos programables aparecieron en el mercado los PEEL que contienen un plano AND programable seguido de un OR fijo más un conjunto de macrocélulas. Los CPLD son dispositivos que están compuestos de macrocélulas y pueden soportar programación con lenguajes de desarrollo tales como ABEL, PALASM por citar ejemplos; Aparecen los Arreglos de Compuertas Programables por Máscara (MPGA por sus siglas en inglés), estos dispositivos se programan mediante un proceso de fabricación en serie y resulta muy caro su fabricación a bajos volúmenes. En forma inversa se desarrollan los Arreglos de Compuertas Programables en Campo (FPGA por sus siglas en inglés *vide supra*) son dispositivos compuestos de macrocélulas (estos dispositivos pueden tener diversas arquitecturas. lo cual depende del fabricante, ver Fig. 27), que pueden ser programadas mediante un programa de cómputo el cual permite diseñar el circuito o expresiones lógicas. Entre sus principales aplicaciones se encuentran: diseños de prototipos y aplicarles diversas pruebas. ya que al ser reconfigurables, se pueden modificar el diseño de una manera rápida y de bajo costo; frecuentemente se utilizan para diseñar las nuevas arquitecturas de microprocesadores y sistemas que necesiten de modificaciones continuas y frecuentes.

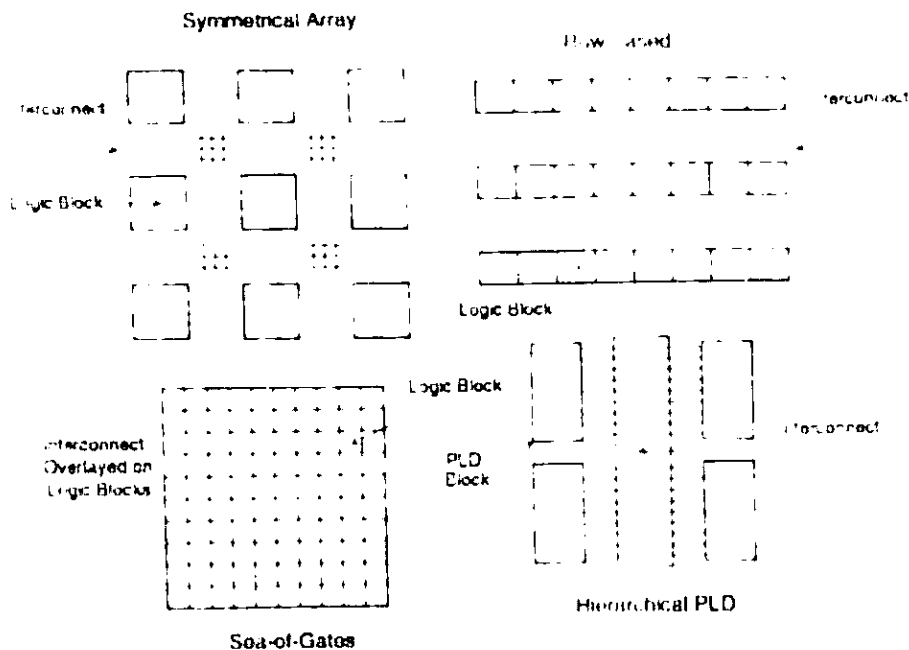


Fig. 27 Diferentes arquitecturas para FPGA's.

Codificación de los dispositivos.

Los circuitos programables al igual que varios dispositivos electrónicos disponibles comercialmente, los fabricantes los dotan de un código para ser diferenciados unos de otros y especificar en el código algunas características que contiene el circuito, a continuación se observa un ejemplo de ello [14].

XC4010-10 PG156C

Donde:

XC4010 es el tipo de dispositivo

10 es la velocidad

PG es el tipo de encapsulado

156 es el número de terminales

C es el rango de temperatura

En la tabla 9 se muestran los códigos de diversos fabricantes para los FPGA's.

Característica	Código	Descripción	Código	Descripción
Código de Fabricante	A	Actel	ATT	AT&T (Lucent)
	XC	Xilinx	Isp	Lattice Logic
	EPM	Altera MAX	M5	AMD MACH 5
	EPF	Altera FLEX	QL	QuickLogic
	CY7C	Cypress		
Tipo de Encapsulado	PL o PC	Plástico, PLCC	VQ	Muy delgado, cuad.
	PQ	Plástico plano cuadrado	TQ	Delgado, cuadrado
	CQ o CB	Cerámico plano cuad.	PP	Plástico, red de term.
	PG	Cerámico red de term.	WB, PB	Red de mini-term.
Aplicación	C	Comercial	B	Militar STD-883
	I	Industrial	E	Extendido
	M	Militar		

Tabla 9 Códigos para FPGA's.

Familias de FPGA reconfigurables de Xilinx.

En las tablas 10, 11 y 12 se muestran características tales como a que familia pertenecen, el número de compuertas programables para que ejecuten alguna función lógica cuando sean programadas y/o reconfiguradas y el número de bloques de entradas y salidas que poseen determinados dispositivos FPGA.

Referencia	Número de compuertas	Número de CLB's	Número de IOB's
XC 2064	1200	64	58
XC 2018	1800	100	74

Tabla 10 Familia 2000 de FPGA.

Referencia	Número de compuertas	Número de CLB's	Número de IOB's	Número de terminales
XC 3020	2000	64	64	74
XC 3030	3000	100	80	98
XC 3042	4200	144	96	118
XC 3064	6400	224	120	140
XC 3090	9000	320	144	166

Tabla 11 Familia 3000 de FPGA.

Referencia	XC4002	4003	4004	4005	4006	4008	4010	4013	4016	4020
Número de compuertas	2,000	3,000	4,000	5,000	6,000	8,000	10,000	13,000	16,000	20,000
Matriz de CLB	8 x 8	10x10	12x12	14x14	16x16	18x18	20x20	24x24	26x26	30x30
Número de CLB	64	100	144	196	256	324	400	576	676	900
Tamaño RAM	2,048	3,200	4,608	6,272	8,192	10,368	12,800	18,432	21,632	28,800
Número de IOB	64	80	96	112	128	144	160	192	208	240

Tabla 12 Familia 4000 de FPGA.

El FPGA reconfigurable.

Introducidas al mercado en 1984 por Xilinx[®] (Posteriormente surgieron más compañías fabricantes de FPGA's tales como Actel[®], Altera[®], Advanced Micro Devices[®] o AMD[®], etc.), con lo cual tenemos en el mercado un amplio catalogo de dispositivos programables que podemos elegir de acuerdo a nuestras necesidades y presupuesto. Al ser del tipo RAM estática, al interrumpir la fuente de alimentación, se pierde la configuración; para evitar este problema, existen tres posibles soluciones:

- Utilizar una batería de litio como una fuente de energía independiente.
- Utilizar una memoria tipo PROM previamente programada y cuando se requiera la memoria puede vaciar la configuración al FPGA cada vez que se inicialize el sistema.
- Utilizar los recursos de un microprocesador, el FPGA se puede programar mediante el modo de esclavo, es decir actúa como un periférico, el microprocesador es el control del sistema y el FPGA es sólo un bloque de aplicación específica dentro de un sistema digital.

Este dispositivo consiste de un arreglo bidimensional de bloques lógicos (Fig. 28) que pueden ser conectados por recursos de interconexión general conocidos como bloques lógicos configurables, el arreglo se encuentra rodeado de bloques de entradas y salidas, los cuales se describen a continuación.

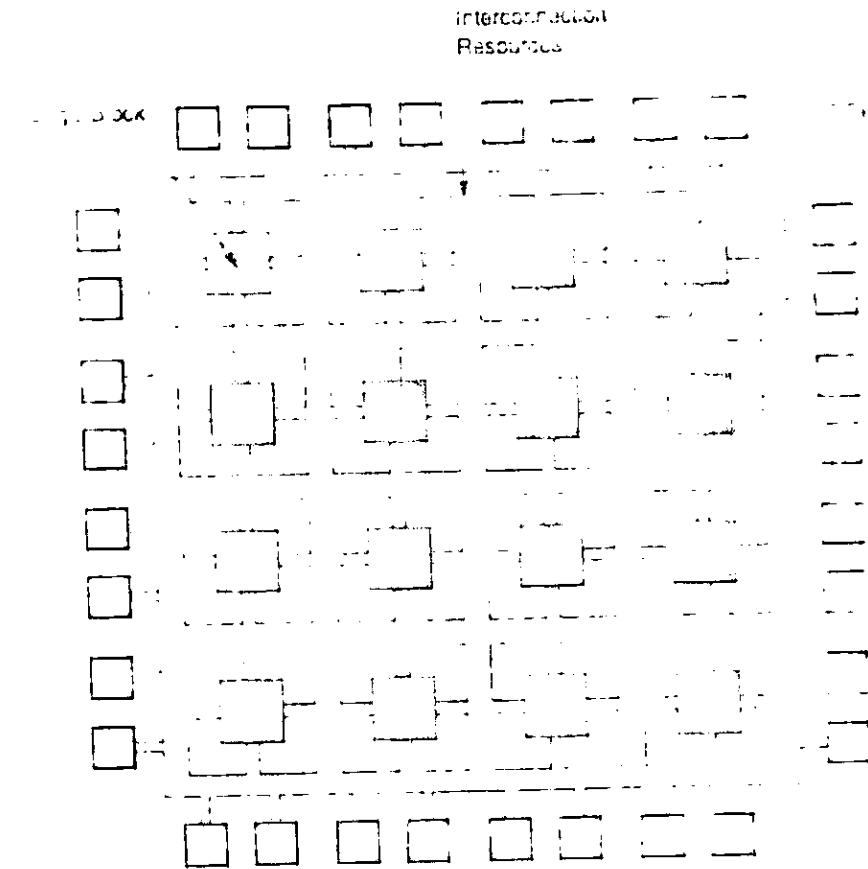


Fig. 28 Arquitectura del FPGA.

4.1 BLOQUE LÓGICO CONFIGURABLE

Un Bloque Lógico Configurable (*Configurable Logic Block* o CLB por sus siglas en inglés) puede ser tan simple como una compuerta NAND de dos entradas, otros CLB tienen estructuras más complejas, tales como multiplexores o bloques completos de lógica combinatoria programables también conocidos como tablas "look-up". En algunos FPGA's, un bloque lógico corresponde a una estructura entera semejante a la PAL. Existe una cantidad muy grande de posibilidades para definir el bloque lógico como un circuito más complejo, consistiendo de varios sub-circuitos y teniendo más de una salida. Muchos bloques lógicos también contienen algún tipo de flip-flop, para ayudar en la implementación de circuitos secuenciales. (Fig. 29) está compuesto por cierto número de subconjuntos básicos:

- Dos flip-flops del tipo D
- Un bloque de lógica combinatorial
- Diversos multiplexores utilizados para seleccionar los diversos modos de funcionamiento de la celda.

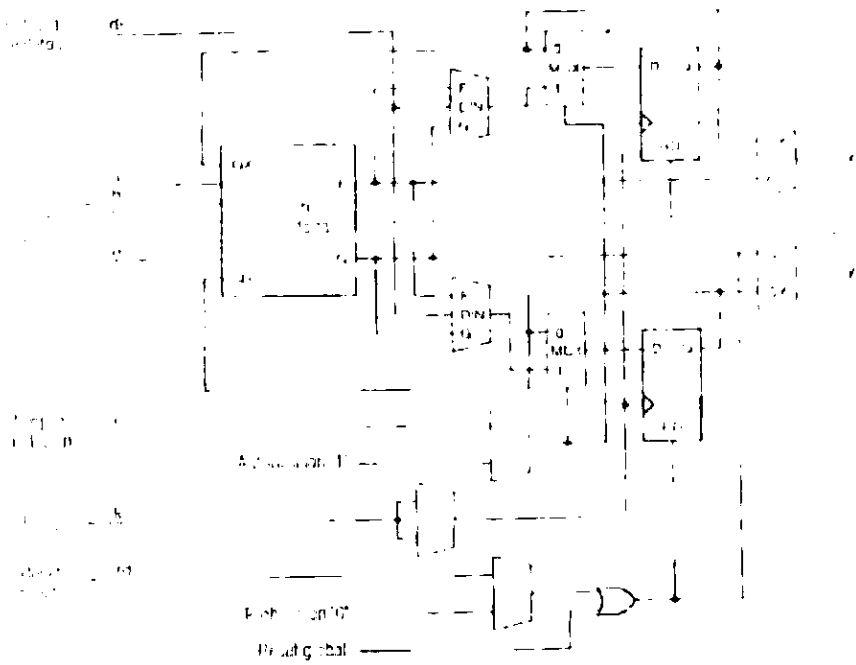


Fig. 29 Estructura de un CLB.

El bloque lógico dispone de cinco entradas lógicas directas (*a*, *b*, *c*, *d* y *e*); dos entradas que provienen de los flip-flops *QX* y *QY*, y dos salidas: *F* y *G*, y puede realizar cualquier función lógica combinatoria de estas entradas (Fig. 29). Este bloque puede trabajar en tres modos distintos: el *modo FG*, el *modo F* y el *modo FGM*. (Fig. 30).

En el modo *FG*, se pueden generar dos funciones lógicas combinatorias independientes de cuatro variables cada una. Para ello, la variable *a* debe ser común a las dos funciones. la segunda y tercera variables pueden ser libremente elegidas entre *b*, *c*, \overline{QX} y \overline{QY} y la cuarta variable puede ser *d* o *e*.

El modo *F* permite realizar cualquier función lógica de cinco variables. las necesarias son: *a*, *d*, *e* y las otras dos elegidas entre *b*, *c*, \overline{QX} y \overline{QY} .

El modo *FGM* permite que se utilice la variable *e* como selector de salida para dos funciones de cuatro variables. Estas funciones utilizan necesariamente como entradas *a* y *d*, y otras dos variables a elegir entre *b*, *c*, \overline{QX} y \overline{QY} .

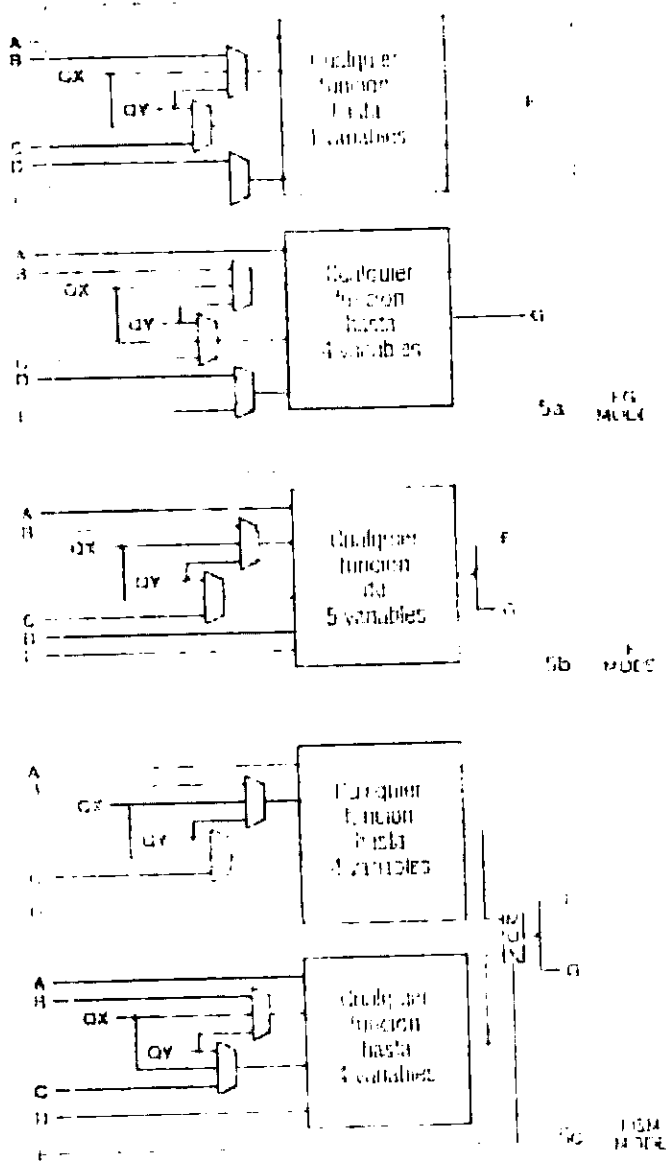


Fig. 30. Modos de funcionamiento de bloques de lógica combinatoria o tablas "look-up".

4.2 BLOQUE DE ENTRADAS/SALIDAS

Este bloque de entradas y salidas (IOB por sus siglas en inglés) permite seleccionar la forma en que se dispondrán las señales de entrada y de salida, especificando cuando y por donde, además de otras funciones, estas funciones se listan a continuación (Fig. 31).

Funciones del IOB:

- Inversión o no de la señal antes de su aplicación al IOB (lógica positiva o negativa).
- Inversión o no de la señal de control del buffer de salida que es del tipo de tres estados.
- Selección del tipo de salida con o sin intervención del registro (combinatoria o registrada).
- Selección del tiempo de subida (*slew rate*) de la señal de salida para adaptarse a la lógica conectada al FPGA.
- Colocación una resistencia de polarización a nivel alto (*pull-up*) cuando la terminal es de entrada (lo que permite dejarla sin conectar estableciendo su nivel).

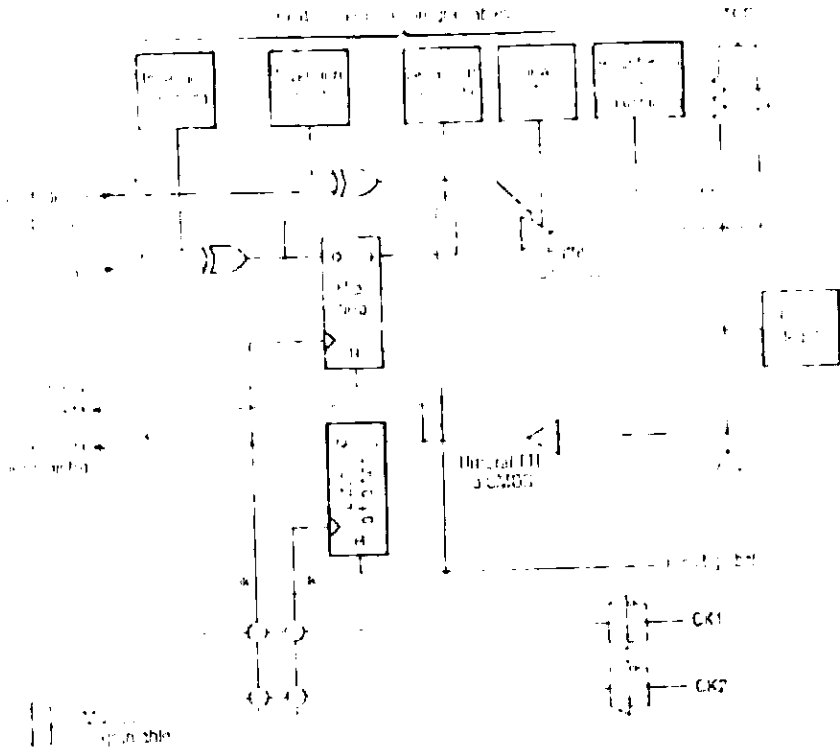


Fig. 31 Estructura de un IOB.

4.3 INTERCONEXIONES

La interconexión comprende segmentos de alambrados, donde los segmentos pueden tener varias longitudes. Presentes en la interconexión están las matrices de conmutación que sirven para conectar los bloques lógicos a los segmentos de alambrado, o uno de los segmentos de alambrado a otro.

Tipos de Interconexiones.

Todos estos caminos de interconexiones están disponibles a nivel de transistores MOS de conmutación, llamados Puntos de Interconexión Programables (*Programmable Interconnect Points*, PIP por sus siglas en inglés) que están repartidos por toda la superficie de la pastilla, (Fig. 32).

- Interconexiones de propósito general
- Interconexiones directas
- Líneas largas

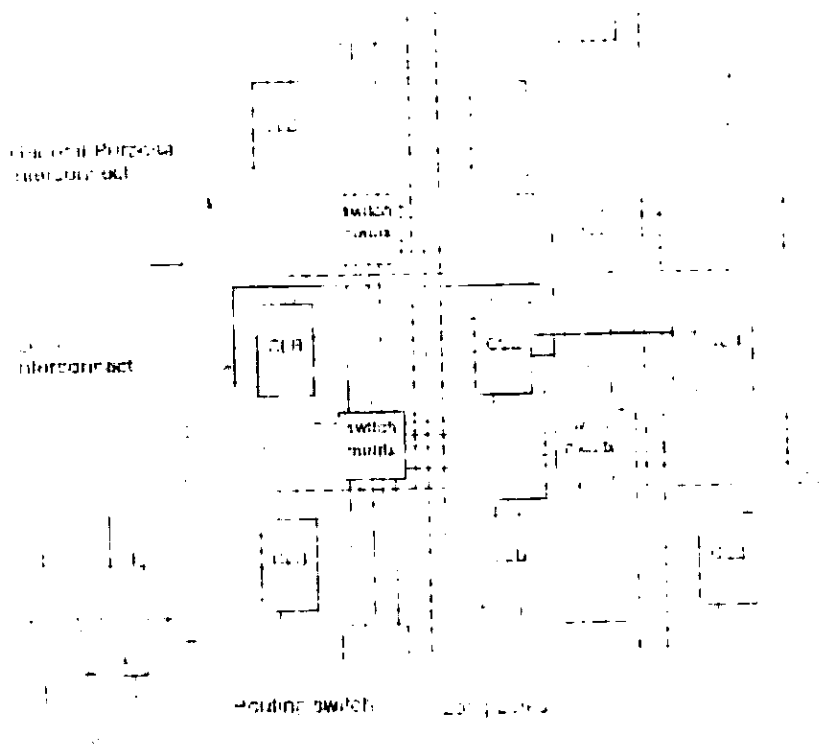


Fig. 32 Interconexiones.

Interconexiones de propósito general.

Estas consisten en una rejilla de cinco segmentos metálicos verticales y cinco segmentos metálicos horizontales colocados entre las filas y columnas del CLB y del IOB. Cada segmento tiene la altura o anchura de un bloque lógico. En cada intersección de fila y columna se encuentra colocada una matriz de conmutación que permite ajustar los segmentos entre sí según las diversas configuraciones. Aunque no están permitidas todas las configuraciones, la cantidad disponible permite una gran flexibilidad.

Con el fin de que no se atenúen demasiado las señales que recorren grandes distancias a través de estas líneas sobre la pastilla, se distribuyen regularmente unos búffers por encima y a la derecha de todas las matrices de conmutación. Es posible, por tanto, hacer pasar por ello las señales que lo necesiten.

Interconexiones directas.

Estas permiten establecer uniones entre los CLB y los IOB con un máximo de eficacia en términos de velocidad y de ocupación de la pastilla. Su realización es posible debido a que, en función de la geometría de las celdas, se pueden conectar directamente ciertas entradas de una a las salidas de otras sin necesitar otros recursos de interconexión.

Líneas largas.

Estas líneas tienen procedimientos de interconexión un poco diferente; no pasan por las matrices de conmutación, se utilizan líneas metálicas que atraviesan la pastilla de arriba a abajo y de izquierda a derecha. Tienen prioridad de uso cuando las señales deben ser transmitidas con un mínimo de retraso entre los diferentes elementos, con el fin de asegurar un sincronismo de funcionamiento lo más perfecto posible.

Oscilador de cuarzo.

Xilinx® ha previsto dotar a sus FPGA de un amplificador inversor de alta velocidad, colocado en un ángulo de la pastilla, que puede ser activado durante la fase de programación con el fin de realizar la operación de un oscilador de cuarzo.

En el presente capítulo se observó la evolución de los dispositivos programables y la estructura del FPGA, indicando cuales son las funciones que realizan sus componentes más destacados.

CAPÍTULO

5

PROGRAMACIÓN DEL FILTRO DIGITAL

En el presente capítulo se conjuga lo visto en los capítulos 3 y 4: es decir, se muestra el proceso de diseño del filtro digital se programa en un dispositivo programable, en particular el FPGA.

Programación de los FPGA.

Se disponen de tres terminales M0, M1 y M2, que sirven para definir el modo de carga (Tabla 13).

El primer modo es el más simple y el que menos espacio requiere en un circuito impreso. Es necesario recurrir a una memoria de acceso serie como las: XC1736 ó XC1765. Esta memoria tipo programable, ha sido previamente programada con el mismo sistema de desarrollo para los FPGA's reconfigurables y sólo sirve para volver a copiar su contenido en los FPGA's reconfigurables cuando estos se apagan y se vuelven a encender.

El segundo modo es análogo al anterior pero los datos se suministran en paralelo, ya que provienen de una memoria PROM o puede ser hasta una memoria EEPROM (anexo B2), dependiendo de su compatibilidad y características deseadas.

El tercer modo es el pasivo de tipo periférico; ha sido concebido para que el FPGA pueda ser configurada a partir de un microprocesador cualquiera, como si fuera un periférico: es decir, cuando es configurada desde un sistema de desarrollo que se encuentra en una computadora personal (*software*). En este trabajo se utilizó la herramienta de diseño (CAD) *Foundation Series 1.5i Software Development System* que suministra un ambiente de programación completo para dispositivos lógicos programables Xilinx® desde CPLD's a FPGA's [17]. El *Foundation* contiene:

- Captura del diseño (*Entry*).
- Síntesis (*Synthesis*).
- Simulación (*Simulation*).
- Herramientas de programación del dispositivo lógico. (*FPGA/CPLD Implementation Tools*).

M0	M1	M2	Modo	Datos
0	0	0	Maestro	Serie
0	0	1	Maestro	Paralelo a partir de 0000
0	1	0	No utilizar	
0	1	1	Maestro	Paralelo a partir de FFFF
1	0	0	No utilizar	
1	0	1	Periférico	Paralelo
1	1	0	No utilizar	
1	1	1	Esclavo	Serie

Tabla 13 Modos de programación de FPGA's.

5.1 DIAGRAMA ESQUEMÁTICO

En la figura siguiente (Fig. 33) se muestra la pantalla del diagrama esquemático del circuito que indica como se conectan los multiplicadores, sumadores y flip-flops de acuerdo a la arquitectura que se diseñó en la figura 19, capítulo 3 (*vide supra*).

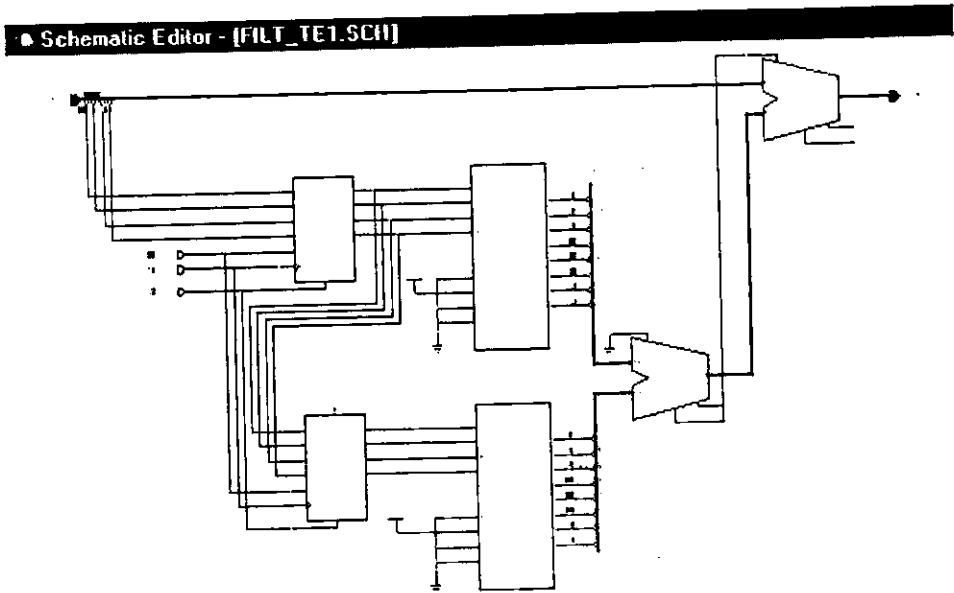


Fig. 33 Diagrama esquemático.

5.2 SIMULACIÓN

En la figura siguiente (Fig. 34) se muestra la pantalla de la simulación del circuito diseñado, es decir, la forma en que opera a ciertas señales de entrada y es aquí donde se verifica si el funcionamiento del circuito es correcto.

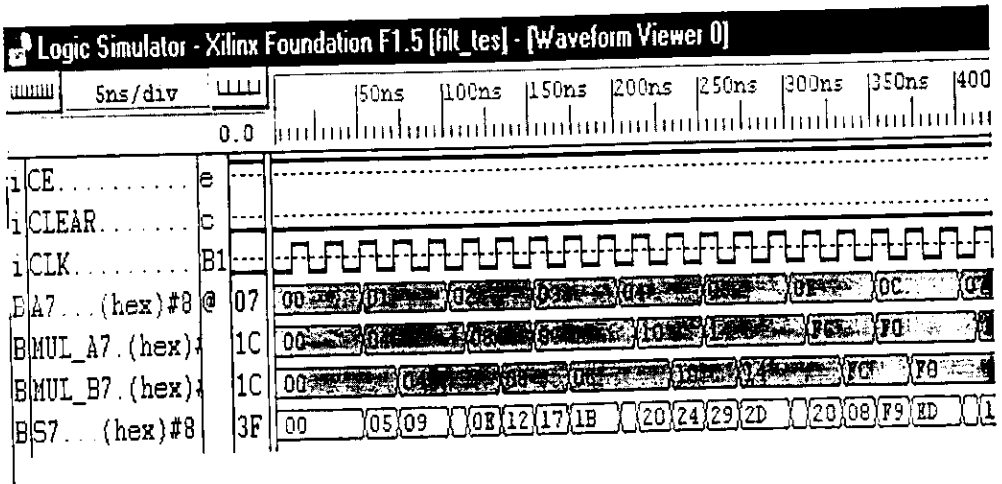


Fig. 34 Simulación del filtro.

5.3 CONFIGURACIÓN DEL FPGA

En la figura siguiente (Fig.35) se muestra la forma en que el programa de cómputo interconecta los bloques internos del FPGA para que realicen la operación determinada en el circuito esquemático y también se puede observar el porcentaje de uso de recursos del FPGA.

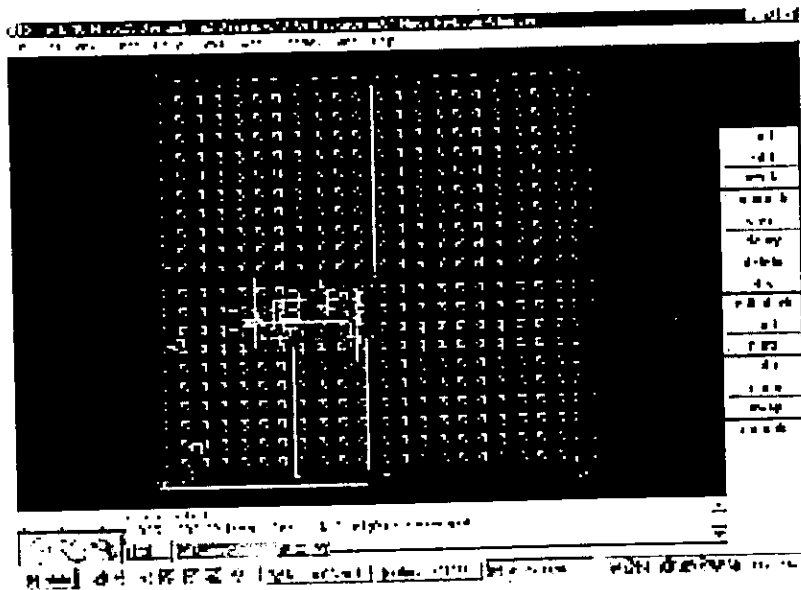


Fig. 35 Configuración del FPGA y enrutamiento.

En el presente capítulo se revisaron los modos de programación de los FPGA's en general. En particular se mostraron las pantallas que presenta el programa de cómputo en el monitor de la computadora personal al efectuar los pasos de diseño y alambrado del circuito esquemático; su posterior simulación en dominio del tiempo discreto y finalmente una pantalla donde el diseño es programado en el dispositivo lógico reconfigurable.

CAPITULO

6

RESULTADOS Y RECURSOS UTILIZADOS

En este capítulo se revisarán los resultados obtenidos de los capítulos anteriores del diseño y programación del filtro digital tipo FIR en un dispositivo reconfigurable (FPGA) a fin de observar su desempeño con señales digitales introducidas al sistema. Asimismo se listan los recursos que se utilizaron en esta prueba.

6.1 RESULTADOS

Los resultados operacionales del filtro digital de acuerdo a su diagrama a bloques son correctos, ya que al ir analizando los resultados de la simulación se observa que los datos digitales se van modificando de acuerdo al circuito por el que pasa la señal, es decir, cuando se multiplica el valor digital de la señal de entrada por el valor del coeficiente y luego se observa que cumple la suma de las señales procesadas (Fig. 34). Por esto se infiere que el funcionamiento del circuito es correcto.

Al hacer un análisis comparativo entre los resultados que se obtienen en un análisis matemático y uno operacional; se utiliza el mismo programa de cómputo para efectuar éste análisis, para ello se utiliza el programa *filter.m* de Matlab® donde se introducen las siguientes instrucciones:

```
b = [1 1 1]
```

Se introducen unos en este vector ya que el número uno es el elemento neutro de la multiplicación ya que de otra forma el programa marcará errores de sintaxis.

```
a = [1 4 4]
```


Donde cada número representa al coeficiente en su formato *radix-3*. (*vide supra*).

$$x = [1 \ 2 \ 3 \ 4 \ 5]$$

Que es el vector de entrada de una señal propuesta.

$$y = \text{filter}(a,b,x);$$

Finalmente la instrucción que hace una llamada al programa que calcula los valores a filtrar, lo cual genera el resultado siguiente:

$$y = 1 \ 5 \ 9 \ 10 \ 14$$

Al observar la figura 34 se nota que los resultados son:

$$y = 0 \ 5 \ 9 \ 0E \ 12$$

Cabe aclarar que los datos entregados por la simulación están en formato hexadecimal, por lo que en formato decimal quedan:

$$y = 0 \ 5 \ 9 \ 14 \ 18$$

Comparando los dos resultados, tanto de análisis matemático como del sistema diseñado y simulado, se observa que las salidas son cuasi-similares.

El circuito diseñado en el diagrama esquemático finalmente se puede programar en un dispositivo real (Fig. 36) que se encuentra en una tarjeta de desarrollo comercializada por el mismo fabricante del FPGA para efectuar pruebas rápidas. Esta tarjeta se conecta a la computadora personal en un puerto (serial o paralelo) mediante un cable igualmente comercializado por el mismo fabricante. Esta tarjeta se conecta con unos convertidores digital analógico y analógico digital, alambrados en una tarjeta para pruebas de prototipos (*proto-board*) para hacer pruebas con señales de instrumentos de laboratorio electrónico.

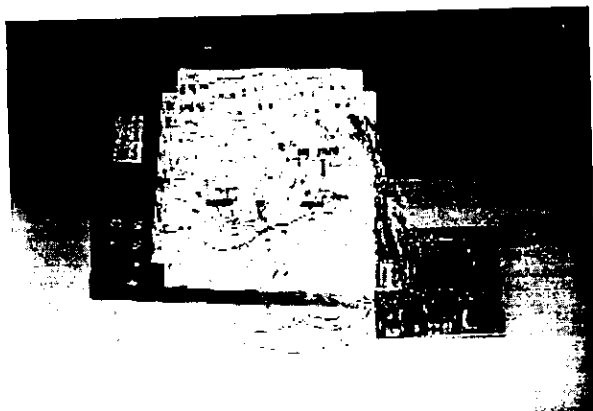


Fig. 36 Fotografía del FPGA.

6.2 RECURSOS UTILIZADOS

Para llevar a cabo este trabajo, se necesitaron los instrumentos de laboratorio que se listan en la tabla 14.

Cantidad	Equipo
1	Computadora personal
2	Generador de funciones
1	Fuente de alimentación
1	Multímetro

Tabla 14 Equipo utilizado.

En la tabla 15, se lista el material utilizado para realizar el proyecto.

Cantidad	Componente
1	Tarjeta con FPGA
1	C. I. DAC0808LCN
1	C. I. ADC0801LCN
1	C. I. LM358N
1	C. I. NE555N
1	C. I. 74LS00
1	osc. 2 MHz
1	pot. 10k
2	pot. 1k
2	cap. Elect 0.47uF
1	cap.ceram. 0.1uF
1	res. 100 ohms
1	res. 330 ohms
3	res. 1 k ohms
1	res. 10 k ohms
1	res. 12 k ohms
20	headers
1/2 mt	alambre cal. 22
1	proto-board

Tabla 15 Material utilizado.

Como se observa se necesitaron relativamente pocos recursos fisicos, casi todos los componentes se encuentran en un laboratorio de electrónica digital bien equipado: algo importante a señalar en estos tiempos es, el profesional debe ahora utilizar la computadora como una herramienta que le ayude a solucionar problemas que se encuentran en el área, ya que las nuevas tecnologías son comercializadas con la característica de ajustarse a los requerimientos mediante la ayuda de una computadora personal, profesional que no sea capaz de manejar y entender estas herramientas tiene el riesgo de quedar rebasado por profesionales de otras instituciones o incluso profesionales extranjeros.

CONCLUSIONES
ET
RECOMENDACIONES

CONCLUSIONES

El realizar este trabajo de tesis fue muy motivante ya que se utilizó tecnología relativamente nueva en México, también se observa que casi todas las tecnologías de desarrollo de proyectos como dispositivos programables hasta microprocesadores digitales de señales vienen con un conjunto de herramientas de diseño en forma de programas de cómputo. corresponde al profesional comprender y manejar adecuadamente estas herramientas ya que nos ayudan a reducir tiempos en diseños de proyectos. En cuanto al diseño, se tienen muchos datos en intervalos de tiempos muy pequeños, siendo el instrumento más idóneo para procesar estos datos digitalizados en la computadora por sus características de velocidad y repetibilidad; pero como es un proceso repetitivo, como el autómata, entonces, el autómata no necesita recursos muy grandes. Por lo tanto, se eligió un circuito programable, el FPGA, que es un dispositivo relativamente nuevo en México, por lo que aún no existen recursos humanos en grandes proporciones en el ámbito de licenciatura con los que se pueden desarrollar proyectos en esta tecnología, por lo que esta tesis sirve como una actualización tecnológica para el desarrollo de nuevos proyectos.

RECOMENDACIONES

Este trabajo puede tener otras formas de solución, tales como: desarrollar proyectos utilizando un dispositivo diferente, tal como un procesador digital de señales (DSP), diseñar filtros del tipo IIR, ampliar más etapas a este mismo filtro. También se pueden programar filtros en lenguajes de computación, tales como: VHDL, C y C++, etc.

THE UNIVERSITY OF CHICAGO PRESS

[1] BROWN, Stephen D. FRANCIS, Robert J. ROSE, Jonathan VRANESIC, Zvonko G.

Field-Programmable Gate Arrays

Kluwer Academic Publishers

E.U.A., 1992.

[2] COUGHLIN, Robert F y DRISCOLL, Frederick F.

Circuitos Integrados Lineales y Amplificadores Operacionales

2ª edición

Prentice-Hall

México, 1987.

[3] HSU, Hwei P.

Análisis de Fourier

Addison-Wesley

México, 1989.

[4] IFEACHOR, Emmanuel C y JERVIS, Barrie W.

Digital Signal Processing (A Practical Approach)

Addison-Wesley

Gran Bretaña, 1993.

[5] KATZ, Randy H.

Contemporary Logic Design

The Benjamin/Cummings Publishing Company

E. U. A., 1994.

- [6] MORRIS MANO, M.
Lógica Digital y Diseño de Computadores
Prentice-Hall
México, 1982.
- [7] OGATA, Katsuhiko
Sistemas de Control en Tiempo Discreto
2ª edición
Prentice Hall
México, 1996.
- [8] OPPENHEIM, Alan V. y WILLSKY, Alan S.
Signals & Systems
2ª edition
Prentice-Hall
E.U.A., 1996.
- [9] PARHAMI, Behrooz
Computer Arithmetic (algorithms and Hardware Designs)
Oxford University Press
E.U.A., 1999.
- [10] PARKS, T W y BURRUS, C S.
Digital Filter Design (Topics in Digital Signal Processing)
John Wiley and Sons
E.U.A., 1987.

- [11] PIRSCH, Peter
Architectures for Digital Signal Processing
John Wiley and Sons
England, 1998.
- [12] RHEA, Randall W.
HF Filter Design and Computer Simulation
McGraw-Hill
E.U.A., 1995.
- [13] SCHWARTZ, Micha
Transmisión de Información, Modulación y Ruido
3ª edición, (1ª en español)
McGraw-Hill
México, 1990.
- [14] SMITH, Michael John Sebastian
Application-Specific Integrated Circuits
Addison-Wesley
E.U.A., 1997.
- [15] TAVERNIER, Christian
Circuitos Lógicos Programables
Paraninfo
España, 1994.

[16] TOKHEIM, Roger L.

Principios Digitales

McGraw-Hill, Serie Schaum

México, 1982.

[17] VAN DE BOUT, Dave

The Practical Xilinx Designer Lab Book

Prentice-Hall

E.U.A., 1997.

ANEXOS

ANEXO A1

Programa de Cómputo: FIR1 para calcular coeficientes del filtro.

```
function [b,a] = fir1(N,Wn,varargin)
nargchk(2,5,nargin);

switch length(varargin)
case 1
    if isstr(varargin{1}) & (length(varargin{1}) > 0)
        s = upper(varargin{1});
        switch upper(s)
            case {'SCALE','NOSCALE'}
                Scale = s;
            otherwise
                Ftype = s;
        end
    else
        Wind = varargin{1};
    end
case 2
    if isstr(varargin{1})
        Ftype = varargin{1};
    else
        Wind = varargin{1};
    end
    if isstr(varargin{2})
        Scale = varargin{2};
```

```

else
    Wind = varargin{2};
end
case 3
    Ftype = varargin{1};
    Wind = varargin{2};
    Scale = varargin{3};
end

switch upper(Scale)
case 'NOSCALE'
    SCALING = 0;
case 'SCALE'
    SCALING = 1;
otherwise
    error('Scaling option must be "noscale" or "scale".')
end

if isempty(N) | ~isnumeric(N) | ~isreal(N) | N~=round(N) | N<=0
    error('N must be a real, positive integer.')
end

Ftype = upper(Ftype);
if ~strcmp(Ftype,'HIGH',1) & ~strcmp(Ftype,'STOP',1) & ...
    ~strcmp(Ftype,'DC-0',4) & ~strcmp(Ftype,'DC-1',4) & ...
    ~isempty(Ftype)
    error('Filter type must be "high", "stop", "DC-0", or "DC-1".')
end

nw = length(Wind);

```

```

nbands = length(Wn) + 1;
if (nbands > 2) & isempty(Ftype)
    Ftype = 'DC-0'; % make sure default 3 band filter is bandpass
end
First_Band = isempty(findstr('DC-0',Ftype)) & isempty(findstr('HIGH',Ftype));
mags = rem( First_Band + (0:nbands-1), 2);

L = N + 1;
odd = rem(L, 2);
if (mags(nbands) & ~odd)
    disp('For highpass and bandstop filters, order must be even.')
    disp('Order is being increased by 1.')
    N = N + 1; L = L + 1;
    odd = 1;
end
if nw ~= 0 & nw ~= L
    error('The window length must be the same as the filter length.')
end
if nw == 0 % replace the following with the default window of your choice.
    Wind = hamming(L);
end

if any( Wn<0 | Wn>1 )
    error('Frequencies must fall in range between 0 and 1.')
end
if any(diff(Wn)<0)
    error('Frequencies must be increasing')
end

Wn = Wn(:);

```

```

ff = [0,Wn(1:nbands-1); Wn(1:nbands-1),1];
mags = [mags(:)'; mags(:)'];
hh = firls(L-1,ff(:),mags(:));

b = hh.*Wind(:)';
a = 1;

if SCALING
    if First_Band
        b = b / sum(b); % unity gain at DC
    else
        if ff(4)==1
            % unity gain at Fs/2
            f0 = 1;
        else
            % unity gain at center of first passband
            f0 = mean(ff(3:4));
        end
        b = b / abs( exp(-j*2*pi*(0:L-1)*(f0/2))*(b.' ) );
    end
end
end

```


ANEXO A2

Programa de Cómputo: FILTER para calcular respuesta del filtro.

```
%FILTER One-dimensional digital filter.
% Y = FILTER(B,A,X) filters the data in vector X with the
% filter described by vectors A and B to create the filtered
% data Y. The filter is a "Direct Form II Transposed"
% implementation of the standard difference equation:
%
%  $a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + \dots + b(nb+1)*x(n-nb)$ 
%  $- a(2)*y(n-1) - \dots - a(na+1)*y(n-na)$ 
%
% If a(1) is not equal to 1, FILTER normalizes the filter
% coefficients by a(1).
%
% When X is a matrix, FILTER operates on the columns of X. When X
% is an N-D array, FILTER operates along the first non-singleton
% dimension.
%
% [Y,Zf] = FILTER(B,A,X,Zi) gives access to initial and final
% conditions, Zi and Zf, of the delays. Zi is a vector of length
% MAX(LENGTH(A),LENGTH(B))-1 or an array of such vectors, one for
% each column of X.
%
% FILTER(B,A,X,[],DIM) or FILTER(B,A,X,Zi,DIM) operates along the
% dimension DIM.
%
```

ANEXO B1

Características de FPGA's de Xilinx®.

XC4000E and XC4000X Series

Features

Note: Information in this data sheet covers the XC4000E, XC4000EX, and XC4000XL families. A separate data sheet covers the XC4000XLA and XC4000XV families. Electrical Specifications and package/pin information are covered in separate sections for each family to make the information easier to access, review, and print. For access to these sections, see the Xilinx WEBLIX web site at <http://www.xilinx.com/partinfo/databook.htm#xc4000>.

- System featured Field-Programmable Gate Arrays
- Select-RAM™ memory: on-chip ultra-fast RAM with
 - synchronous write option
 - dual-port RAM option
- Fully PCI compliant (speed grades -2 and faster)
- Abundant flip-flops
- Flexible function generators
- Dedicated high-speed carry logic
- Wide edge decoders on each edge
- Hierarchy of interconnect lines
- Internal 3-state bus capability
- Eight global low-skew clock or signal distribution networks
- System Performance beyond 80 MHz
- Flexible Array Architecture
- Low Power Segmented Routing Architecture

- **Systems-Oriented Features**
- IEEE 1149.1-compatible boundary scan logic support
- Individually programmable output slew rate
- Programmable input pull-up or pull-down resistors
- 12 mA sink current per XC4000E output
- **Configured by Loading Binary File**
- Unlimited re-programmability
- **Read Back Capability**
- Program verification
- Internal node observability
- **Backward Compatible with XC4000 Devices**
- **Development System runs on most common computer platforms**
- Interfaces to popular design environments
- Fully automatic mapping, placement and routing
- Interactive design editor for design optimization

Low-Voltage Versions Available

- Low-Voltage Devices Function at 3.0 - 3.6 Volts
- XC4000XL: High Performance Low-Voltage Versions of XC4000EX devices

Additional XC4000X Series Features

- Highest Performance — 3.3 V XC4000XL
- Highest Capacity — Over 180,000 Usable Gates
- 5 V tolerant I/Os on XC4000XL
- 0.35 mm SRAM process for XC4000XL
- Additional Routing Over XC4000E
 - almost twice the routing capacity for high-density designs
- Buffered Interconnect for Maximum Speed Blocks
- Improved VersaRing™ I/O Interconnect for Better Fixed Pinout Flexibility
- 12 mA Sink Current Per XC4000X Output
- Flexible New High-Speed Clock Network
 - Eight additional Early Buffers for shorter clock delays
 - Virtually unlimited number of clock signals
- Optional Multiplexer or 2-input Function Generator on Device Outputs
- Four Additional Address Bits in Master Parallel Configuration Mode
- XC4000XV Family offers the highest density with 0.25 mm 2.5 V technology

ANEXO B2

Memoria EEPROM para reconfigurar el FPGA.

**FPGA Configuration EEPROM Memory
64K, 128K and 256K**

AT17C65 AT17LV65 AT17C128 AT17LV128 AT17C256

Features

- EE Programmable 65,536 x 1-, 131,072 x 1-, and 262,144 x 1-bit Serial Memories
Designed to Store Configuration Programs for Field Programmable Gate Arrays (FPGAs)
- In-System Programmable Via 2-wire Bus
- Simple Interface to SRAM FPGAs
- Compatible with Atmel AT6000, AT40K FPGAs, Altera FLEX® Devices, Lucent ORCA® FPGAs, Xilinx XC3000, XC4000, XC5200, SPARTAN® FPGAs, Motorola MPA1000 FPGAs
- Cascadable Read Back to Support Additional Configurations or Future Higher-density Arrays (128K and 256K only)
- Low-power CMOS EEPROM Process
- Programmable Reset Polarity
- Available in the Space-efficient Plastic DIP or SOIC Packages; PLCC Package is Pin-compatible Across Product Family
- Emulation of Atmel's AT24CXXX Serial EEPROMs
- Available in 3.3V ± 10% LV and 5V ± 5% C Versions
- Low-power Standby Mode

Description

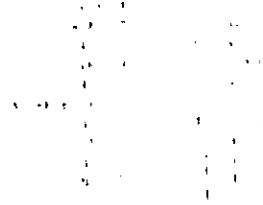
The AT17C65/128/256 and AT17LV65/128/256 (low-density AT17 Series) FPGA Configuration EEPROMs (Configurators) provide an easy-to-use, cost-effective con-figuration memory for Field Programmable Gate Arrays. The low-density AT17 Series is packaged in the 8-pin DIP and the popular 20-pin PLCC and SOIC. The AT17 Series family uses a simple serial-access procedure to configure one or more FPGA devices. The AT17 Series organization supplies enough memory to configure one or multiple smaller FPGAs. Using a feature of the AT17 Series, the user can select the polarity of the reset function by programming a special EEPROM byte. These devices also support a write-protection mechanism within its programming mode. The AT17 Series Configurators can be programmed with industry-standard program-mers, or Atmel's ATDH2200E Programming Kit.

Pin Configurations

PLCC



SOIC



DIP

