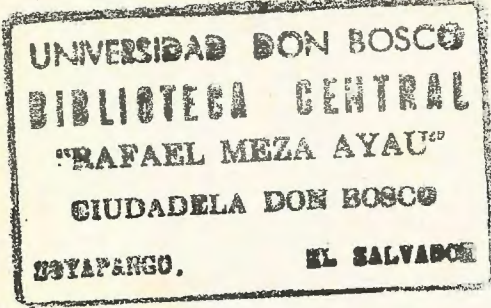


cin 9140 \$50-

UNIVERSIDAD DON BOSCO



Diseño y Construcción de una Tarjeta Electrónica compatible con PC'S AT para la demostración física de la transformada de Fourier.

Trabajo de Graduación preparado para la FACULTAD DE INGENIERIA

Para optar al grado de INGENIERO EN ELECTRONICA

Por

Victor Manuel Callejas de Paz
Luis Armando Antonio Bonifacio

Noviembre 1995

Soyapango.

El Salvador,

Centroamerica

UNIVERSIDAD
DON BOSCO



DISEÑO Y CONSTRUCCION DE UNA TARJETA ELECTRONICA COMPATIBLE CON PC'S AT
PARA LA DEMOSTRACION FISICA DE LA TRANSFORMADA DE FOURIER.

TRABAJO DE GRADUACION

PREPARADO PARA LA

FACULTAD DE INGENIERIA

PARA OPTAR AL GRADO DE

INGENIERO EN ELECTRONICA

POR

VICTOR MANUEL CALLEJAS DE PAZ

LUIS ARMANDO ANTONIO BONIFACIO

NOVIEMBRE - 1995

UNIVERSIDAD
DON BOSCO

RECTOR

ING. FEDERICO MIGUEL HUGUET RIVERA

SECRETARIO GENERAL

LIC. PIERRE MUYSHONDT S.D.B.

DECANO FACULTAD DE INGENIERIA

ING. MIGUEL HERNANDEZ

ASESOR DE TRABAJO DE GRADUACION

ING. JAIME MORALES LOPEZ

JURADO EXAMINADOR

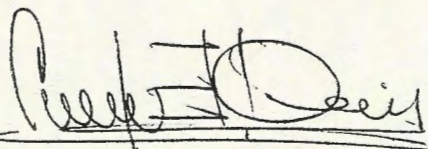
ING. JULIO HERNANDEZ ROSALES

ING. CARLOS OSORIO MENDOZA

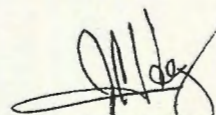
UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
DEPARTAMENTO DE INGENIERIA ELECTRONICA

JURADO EVALUADOR DEL TRABAJO DE GRADUACION

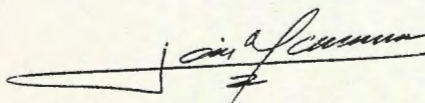
"DISEÑO Y CONSTRUCCION DE UNA TARJETA ELECTRONICA COMPATIBLE CON PC'S AT
PARA LA DEMOSTRACION FISICA DE LA TRANSFORMADA DE FOURIER"



ING. CARLOS OSORIO MENDOZA
JURADO



ING. JULIO HERNANDEZ ROSALES
JURADO



ING. JAIME MORALES LOPEZ
ASESOR

DEDICATORIA

DEDICAMOS ESTE TRABAJO DE GRADUACION A DIOS TODOPODEROSO Y A MARIA AUXILIADORA, QUIENES HAN OBRADO EN NOSOTROS MOSTRANDONOS LA LUZ AL FINAL DEL TUNEL, EN NUESTROS MOMENTOS MAS DIFICILES NUNCA NOS ABANDONARON... GRACIAS SENOR POR PRESTARNO LA VIDA NECESARIA PARA ALCANZAR ESTE LOGRO PROFESIONAL Y ESPERAMOS SERVIRTE CADA DIA CON HUMILDAD Y ETERNO AGRADECIMIENTO...

*UNA NOCHE UN HOMERE SONO QUE PASEABA
POR LA PLAYA CON NUESTRO SENOR. EN EL CIELO
APARECIERON ESCENAS DE SU VIDA CON DOS SERIES DE
HUELLAS EN LA ARENA; UNA DE EL Y LA OTRA DEL SENOR.*

*AL APARECER LA ULTIMA ESCENA DELANTE DE EL,
TORNO SU MIRADA HACIA ATRAS Y NOTO
QUE MUCHAS VECES EN EL CAMINO DE SU VIDA
SOLO HABIA UNA SERIE DE HUELLAS EN SUS
MOMENTOS MAS TRISTES Y DEPRIMIDOS.*

*ESTO LE MOLESTABA TANTO QUE PREGUNTO:
"SEÑOR, ME DIJISTE QUE UNA VEZ QUE DECIDIERA
SEGUIRTE CAMINARIAS CONMIGO TODA LA
DISTANCIA, PERO HE NOTADO QUE EN LOS
MOMENTOS MAS DIFICILES DE MI VIDA SOLO HAY
UNA SERIE DE HUELLAS. NO COMPRENDI POR QUE
CUANDO MAS TE NECESITABA TU ME DEJABAS SOLO."*

*EL SENOR LE CONTESTO: "HIJO MIO, TE AMO Y
NUNCA TE HE ABANDONADO. EN TUS MOMENTOS
DE PRUEBA Y SUFRIMIENTO, CUANDO TU SOLO
VEZ UNA SERIE DE HUELLAS,
ERA ENTONCES QUE YO TE CARGABA."*

AUTOR DESCONOCIDO

QUEREMOS AGRADECER ADEMAS A LAS SIGUIENTES PERSONAS QUIENES DESINTERESADAMENTE CONTRIBUYERON DE ALGUNA MANERA A LA CULMINACION DE ESTE PROYECTO:

ING. JULIO HERNANDEZ, ING. CARLOS OSORIO, ING. JAIME MORALES, ING. FEDERICO HUGUET, LIC. PIERRE MUYSHONDT S.D.B., PPRO. CESAR MONTENEGRO S.D.B.; AL PERSONAL DEL TALLER DE ELECTRONICA POR SU SOLIDARIDAD Y COMPRENSION: OSCAR VILLALTA, OMAR PINEDA "EL MORENO CHIQUITO", WILLBERT MARTINEZ, "ING. PABLITO", MILTON LANDOS, FRANCISCO "SINCLAIR", AL CAIDO EN ACCION ROBERTO CONTRERAS "CAMANEY"; AL PERSONAL DEL TALLER DE MECANICA POR SU INFINITA PACIENCIA: NESTOR "EL SECO" MENDEZ, WILLIAM RODRIGUEZ "EL CHINO", LEONEL GUZMAN "EL CHELE", ISMAEL GOMEZ, MARCO ANTONIO VILLALTA; A LOS BECADOS QUE ANDAN VAGANDO POR EUROPA: ING. OSCAR DURAN VIZCARRA E ING. CESAR BARAHONA; A LA NIÑA DELMY Y A LA NIÑA IMELDA POR OFRECER TAN GENTILMENTE SUS FOTOCOPIADORAS.

VICTOR Y LUIS.

DEDICATORIA PERSONAL

DEDICO ESTE TRABAJO DE GRADUACION A DIOS NUESTRO SEÑOR Y A LA VIRGEN MARIA POR EL MILAGRO QUE OBRARON JUSTO EN EL MOMENTO EN QUE MAS LO NECESITABAMOS. A MIS PADRES, EL ING. VICTOR MANUEL CALLEJAS SANTOS Y LA LIC. AURISTELA DE PAZ DE CALLEJAS, A QUIENES AGRADEZCO LA DISCIPLINA QUE FORMARON EN MI CUANDO NIÑO Y ESPERO QUE AHORA SE SIENTAN MUY ORGULLOSOS DE SU HIJO, A TI, MI AMADA Y FUTURA ESPOSA XOCHILT, A DAVID, MI UNICO HERMANO, GRACIAS POR SER COMO TU ERES Y TE ASEGURO QUE TU TAMBIEN ALCANZARAS EL TRIUNFO, A DON HUGO Y DOÑA DORIS CALDERON, POR TRATARME Y APOYARME COMO A UN HIJO MAS, A MIS FUTUROS CUÑADOS VLADIMIR Y CAROLA CALDERON, A MI CASI HERMANO, INGENIERO NESTOR ROMAN LOZANO, QUIEN POR TANTOS AÑOS HA SIDO MI APOYO MORAL Y UN VERDADERO AMIGO, A MI ABUELITA MARIA CALLEJAS (Q.D.D.G.) Y A MI ABUELO "PAPA" FELIPE DE PAZ H. (Q.D.D.G.) QUIENES SIEMPRE SONARON CON VERME CONVERTIDO EN UN PROFESIONAL, A MIS TIOS, MARIO, MARIA IGNACIA, FELIPE, ARMANDO, RIGO, ROBERTO Y ROLANDO (Q.D.D.G.) Y A TODA MI FAMILIA EN GENERAL.

ADEMAS DESEO AGRADECER A LAS SIGUIENTES PERSONAS, SIN LAS CUALES JAMAS HUBIERA PODIDO LLEGAR HASTA AQUI:

A MIS MAESTROS MAS DESTACADOS: PROF. BALTAZAR MOLINA, PROF. FRANCISCO AGUILA, PBRO. GIUSSEPPE CORO S.D.B., PBRO. MANUEL MEJIA S.D.B.; A MIS AMIGOS DE INFANCIA: "NECO", GERARDO, "EL MAESTRO GULP", TITO "TIBURON" ZELAYA, MARIO "MEMF" MARTINEZ, Y TODOS AQUELLOS DE LOS QUE NO RECUERDO EL NOMBRE; A MIS EX-COMPAÑEROS DE TRABAJO: NERIS, REYNALDO, RAUL, ING. SANDOVAL, MARIO ORELLANA, NELSON NOLASCO, ING. DANIEL ALARCON, ALFONSO ANAYA, EVERALDO, LIC. MELVIN MENDOZA, LUISA, ANGEL, GERSON, JUAN, ETC. Y MUY ESPECIALMENTE A UN SURFISTA DE LAS PLAYAS DE TAQUILLO DE QUIEN DESCONOZCO SU NOMBRE PERO QUE SIN SU AYUDA NO ESTARIA AQUI PARA CONTARLO.

A TODOS UDS. INFINITAS GRACIAS POR AYUDARME A SER LO QUE AHORA SOY...

VICTOR CALLEJAS

DEDICATORIA

ESTE TRABAJO SE LO DEDICO A MI MADRE ANA ESTHER DE ANTONIO
(Q.D.D.G.) CON TODO MI AMOR. TAMBIEN QUIERO AGRADECER A:

DIOS TODOPODEROSO POR PERMITIR QUE TERMINARA MIS ESTUDIOS SUPERIORES, A LA VIRGEN MARIA Y A DON BOSCO POR ESTAR SIEMPRE CONMIGO.

- A MI PAPA: LUIS ARMANDO ANTONIO GARCIA POR HABERME DADO LA OPORTUNIDAD DE FORJAR UNA CARRERA PROFESIONAL Y TENER SIEMPRE SU APOYO.

- MI PRIMO: JOSE ADALBERTO MARTINEZ POR LA AYUDA QUE ME BRINDO Y ME SIGUE BRINDANDO.

- ING. NESTOR LOZANO: POR LA COLABORACION Y CONSEJOS BRINDADA DURANTE LA ELABORACION DE MI TRABAJO DE GRADUACION.

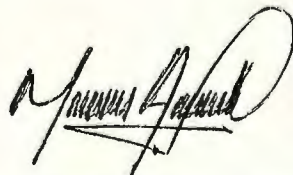
- MIS HERMANOS: AMILCAR, JAIME Y GRISELDA.

- MI MADRINA: ESMERALDA ANTONIO.

- VICKY DE GOMEZ Y A ROSY: POR EL APOYO MORAL QUE ME BRINDARON DURANTE TODA MI PREPARACION UNIVERSITARIA.

- MIS AMIGOS QUE SIEMPRE CONFIARON EN MI: EDUARDO CARRANZA, MARIO DÚBON, ALFREDO AQUINO, ALFREDO ESCOBAR.

PARA ELLOS Y TODOS LOS QUE NO HE PODIDO MENCIONAR, MI MAS GRANDE AGRADECIMIENTO POR SIEMPRE.



LUIS ARMANDO ANTONIO B.

SAN SALVADOR, 14 DE NOVIEMBRE DE 1995

INTRODUCCION

Una señal eléctrica puede ser descrita como una función del tiempo, tal como $\cos(\omega_0 t + \phi)$ o bien como e^{-at} . Esto es una descripción en el dominio del tiempo de la señal. Sin embargo, cualquier señal que pueda generarse en un laboratorio puede expresarse como una suma (que puede ser discreta o integral) de senoides de diferentes frecuencias. Esta representación por amplitudes (y fases) de las componentes de frecuencia de la señal es lo que se conoce como su espectro en frecuencia. [1]

Este concepto es fundamental en la formación de los estudiantes de las carreras de ingeniería como herramienta de análisis para sistemas que se relacionan con señales periódicas variables en el tiempo.

Lamentablemente, a nivel nacional existen pocas instituciones con los medios y la posibilidad de conseguir equipos experimentales que sirvan como apoyo práctico de la teoría impartida al respecto.

La transformada de Fourier es clave para el análisis de sistemas en el dominio de la frecuencia, por lo que es necesario cimentar bien las bases matemáticas desde los primeros años de estudios superiores. Una observación física de la demostración de la transformada puede ser de gran ayuda en la interpretación y asimilación de tales conocimientos.

El Capítulo I "ASPECTOS GENERALES", ubica al lector en el contexto del problema de adquirir un equipo de laboratorio para la demostración física de la transformada de Fourier, en forma accesible económicamente, así como dejar planteadas algunas de las alternativas de solución factibles de implementar en nuestro medio para solventar dicho problema.

El Capítulo II "LA TRANSFORMADA DE FOURIER" contiene todas las demostraciones matemáticas que soportan la transformada de Fourier.

En el Capítulo III "HERRAMIENTAS MATEMATICAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO" se da una pequeña introducción al procesamiento de señales así como se exponen los fundamentos y las demostraciones que sostienen los procedimientos de la transformada discreta de Fourier (TDF) y la transformada rápida de Fourier (TRF) empleados para el desarrollo del proyecto.

El Capítulo IV, "DISEÑO DE LA TARJETA ADQUISITORA DE DATOS" describe la estructura general del circuito diseñado y elaborado para el proyecto, explicándolo primeramente de una forma global y luego dividiéndolo por etapas.

Finalmente, el Capítulo V "ELABORACION DEL SOFTWARE DE CONTROL" brinda información básica acerca de los lenguajes de programación empleados en el proyecto, la cual puede resultar de mucha utilidad a la hora de implementar un proyecto semejante. Además se explica el programa principal de control y de procesamiento de la señal.

INDICE

INTRODUCCION	I
CAPITULO I. ASPECTOS GENERALES	
1.1. Objetivos.	1
1.1.1 Objetivo general.	1
1.1.2 Objetivos específicos.	2
1.2. Justificación del tema	3
1.3. Alcances y limitaciones.	4
1.4. Marco teórico de referencia.	6
1.4.1 Conceptos utilizados en el proyecto.	6
1.4.2 Situación actual.	11
CAPITULO II. LA TRANSFORMADA DE FOURIER	
2.1. Series de Fourier.	13
2.1.1 Funciones Periódicas	13
2.1.2 Series de Fourier	13
2.1.3 Propiedades de las Funciones Seno y Coseno: Funciones Ortogonales	14
2.1.4 Forma compleja de las series de fourier.	15
2.2. Transformada de Fourier.	22
2.2.1 Interpretacion de las transformadas de Fourier.	23
2.2.2. Propiedades de las transformadas de Fourier.	25

CAPITULO III. HERRAMIENTAS MATEMATICAS UTILIZADAS EN EL
DESARROLLO DEL PROYECTO

3.1. Fundamentos del procesamiento de señales digitales.	27
3.1.1 Digitalización de señales.	30
3.1.2 Teorema del muestreo.	38
3.2. Transformada discreta de Fourier (TDF).	41
3.2.1 Definición.	42
3.2.2. Cálculo de la TDF.	44
3.3. La transformada rápida de Fourier (TRF).	47
3.3.1 Definición.	48
3.3.2 Derivación analítica del algoritmo de la TRF por decimación en tiempo.	53

CAPITULO IV. DISEÑO DE LA TARJETA ADQUISITORA DE DATOS

4.1. Estructura general.	59
4.2. Etapa de Filtros.	60
4.3. Etapa de conversión A/D	63
4.4. Etapa de memoria auxiliar	67
4.5. Etapa de entrada/salida.	71
4.5.1. Descripción de los canales de E/S de expansión de las IBM PC/AT	72
4.5.2. Funcionamiento de la etapa de interfase	76
4.6. El PPI 8255 (Interfase Programable de Periféricos)	79
4.6.1 Modos de Operación del 8255	80
4.6.2 Modo 0: Entrada/Salida elemental	82
4.6.3 Modo 1: Entrada/Salida con protocolo	84
4.6.4 Modo 2: Puerto bidireccional con protocolo	86

CAPITULO V. ELABORACION DEL SOFTWARE DE CONTROL

- 5.1. Lenguaje TurboPASCAL 88
 - 5.1.1 Reseña histórica del lenguaje PASCAL. 88
 - 5.1.2 El TurboPASCAL y otras modalidades del lenguaje PASCAL. 90
 - 5.1.3 Escritura de un programa en PASCAL. 92
 - 5.1.4 La escritura del lenguaje PASCAL. 92
 - 5.1.5 Formatos de un programa en PASCAL. 93
 - 5.1.6. Declaraciones. 95
 - 5.1.7 Resumen de la organización de un programa. . . 95
 - 5.1.8 Organización formal de un programa. 96
- 5.2. Lenguaje ensamblador. 97
 - 5.2.1 El Lenguaje Ensamblador. 97
 - 5.2.2 Ventajas del Lenguaje Ensamblador. 97
 - 5.2.3 Formato del Programa Fuente. 99
 - 5.2.4 El Formato Interno. 100
 - 5.2.5 Formato Externo. 100
 - 5.2.6 El proceso formal de ensamblaje. 101
- 5.3. Programa de control de la tarjeta. 103

CONCLUSIONES Y RECOMENDACIONES

ANEXOS

- ANEXO A. PROPIEDADES DE LA TRANSFORMADA DE FOURIER A1
- ANEXO B. PROPIEDADES DE LA TRANSFORMADA DISCRETA DE FOURIER B1
- ANEXO C. DIAGRAMA EN BLOQUES DEL CIRCUITO C1
- ANEXO D. DIAGRAMAS PICTORICOS DEL CIRCUITO D1
- ANEXO E. FLUJOGRAMA DEL PROGRAMA PRINCIPAL E1
- ANEXO F. GUIA DEL USUARIO F1
- ANEXO G. GUIAS DE LABORATORIO PARA EL ESTUDIANTE G1

REFERENCIAS BIBLIOGRAFICAS

(11)

CAPITULO I

ASPECTOS GENERALES

En el presente capítulo se da una breve introducción sobre el tema del proyecto desarrollado. Se hace referencia a los objetivos planteados al inicio del proyecto. A continuación se expondrán las razones que apoyan y justifican el desarrollo de dicho proyecto así como los alcances y las limitaciones del mismo. Finalmente, se provee al lector de un marco teórico de referencia para definir algunos conceptos complejos que se manejan en el resto del documento.

1.1. Objetivos.

Con la realización de este proyecto de graduación se alcanzan los siguientes objetivos:

1.1.1. Objetivo general.

Crear un equipo de laboratorio para la demostración física de la transformada de Fourier accesible, con fines didácticos en forma de una tarjeta electrónica que funcione con una computadora AT o cualquiera que sea compatible.

1.1.2. Objetivos específicos.

i) Investigar aspectos referentes a las diferentes técnicas de procesamiento de señales a fin de seleccionar aquellas que más se adapten a nuestras necesidades.

ii) Diseñar y construir la tarjeta de interfaz adquisidora de datos para realizar los muestreos de la señal a analizar, digitalizar las muestras e introducirlas como datos de entrada a la computadora.

iii) Diseñar e implementar un programa de software que realice las operaciones de cálculo y eliminación de error necesarias con los datos digitalizados de las muestras obtenidas y presente los resultados en forma gráfica.

iv) Elaborar 2 guías de prácticas para el usuario del equipo.

v) Crear un folleto de instrucciones sobre el manejo adecuado del equipo.

vi) Plantear los lineamientos suficientes y necesarios para la implementación de este sistema.

1.2. Justificación del tema.

- Con éste trabajo se contribuye al desarrollo de dicho proceso mediante el diseño de un equipo de laboratorio accesible donde se desarrollen prácticas que demuestren físicamente la transformada de Fourier.
- Se estimula la asimilación de los conocimientos por parte de los estudiantes mediante la observación e interpretación física de la transformada de Fourier.
- Se sientan las bases respecto a la elaboración de equipos de laboratorio empleando tecnología y recursos disponibles en el país.
- Se crean los lineamientos suficientes y necesarios para un posterior rediseño o la creación de equipos capaces de realizar un análisis espectral completo.

1.3. Alcances y limitaciones.

Alcances:

1. Se crea un equipo de laboratorio de carácter didáctico para la demostración física de la transformada de Fourier mediante el empleo de una computadora aplicando técnicas de procesamiento de señales por software.
2. La implementación del equipo se hace con tecnología y recursos disponibles en el país.
3. Los resultados del análisis de las señales se presentan en pantalla con opción de ver el espectro resultante en forma discreta o continua.
4. El equipo es compatible con pc's AT que utilicen bus tipo ISA.
5. La velocidad de muestreo es independiente a la velocidad del procesador de la PC.

Limitaciones:

1. El rango de amplitudes de operación es de $\pm 10\text{VDC}$.
2. El equipo es capaz de analizar señales de hasta un máximo de 10KHz .
3. El equipo es capaz de presentar en pantalla el espectro correspondiente a cualquier señal de entrada; sin embargo, la exactitud del espectro se garantiza únicamente para señales periódicas en el tiempo.
4. La velocidad de procesamiento de la señal queda supeditada a la velocidad del procesador de la PC.
5. El equipo está diseñado para trabajar en conjunto con un generador de señales comercial, cuyo diseño y construcción están fuera de los alcances del trabajo.
6. La presentación de los resultados no es en tiempo real.

1.4. Marco teórico de referencia.

1.4.1. Conceptos utilizados en el proyecto.

En conceptos de electrónica y de telecomunicaciones, una señal es una corriente eléctrica modulada, portadora de informaciones aptas para ser traducidas en signos escritos, sonidos o imágenes.

Todas las señales eléctricas pueden ser descritas ya sea como una función del tiempo o de la frecuencia. Para observar una señal eléctrica como función del tiempo se utiliza el osciloscopio. Para observar una señal eléctrica como una función de la frecuencia se emplea el analizador espectral.

El analizador espectral es un equipo de laboratorio que se utiliza para visualizar y medir el espectro de frecuencia de las señales eléctricas.

El espectro de frecuencia es una representación de la distribución de potencia de una señal en un rango de frecuencias.

El análisis de las señales en el dominio de la frecuencia involucra lo que es el análisis de Fourier.

El barón Joseph de Fourier fué un físico y matemático francés nacido en Auxerre (1768-1830). En 1822, Fourier demostró que una señal periódica en el tiempo puede ser representada matemáticamente por una suma infinita de funciones seno y coseno cada una con su propia amplitud, frecuencia y fase.^[2]

A éstas funciones seno y coseno que componen a la señal se les denomina armónicos. Estos armónicos tienen la característica que su frecuencia es $n(\omega_0)$, siendo ω_0 la frecuencia fundamental y

$n=1,2,3,\dots$. La frecuencia del primer armónico es lo que de ahora en adelante entenderemos como frecuencia fundamental. Al trazar una curva continua que pase por las crestas de los armónicos de un espectro se obtiene su envolvente. La naturaleza de la señal dependerá de la amplitud y la fase de cada una de las componentes armónicas.

A pesar de que la demostración de Fourier era de carácter puramente matemático, su teoría es aplicable a las señales eléctricas: Cualquier tipo de señal periódica puede ser considerada como la composición de un número de ondas senoidales. Las ondas senoidales pueden considerarse entonces como el elemento básico de cualquier señal.

Las señales eléctricas no son usualmente generadas mediante la suma de señales senoidales. Sin embargo, el comportamiento de los circuitos eléctricos puede analizarse como si éstas realmente lo fueran, y es muy útil saber sus frecuencias y niveles relativos.

Un analizador espectral utiliza un proceso electrónico para descomponer una señal en sus diferentes ondas senoidales o componentes armónicas. La frecuencia y el nivel de potencia relativo de cada componente es desplegado gráficamente en el espectro (Figura 1.1).

El equipo de laboratorio que se ha desarrollado tiende a disminuir al máximo la parte de procesamiento de señal en base a circuitería sustituyéndolo por un procesamiento en base a procedimientos matemáticos mediante el uso de una computadora. (Ver Figura 1.2).

Los filtros digitales tienen aplicaciones en procesamiento de señales, simulación de sistemas y análisis espectral. En concreto, un

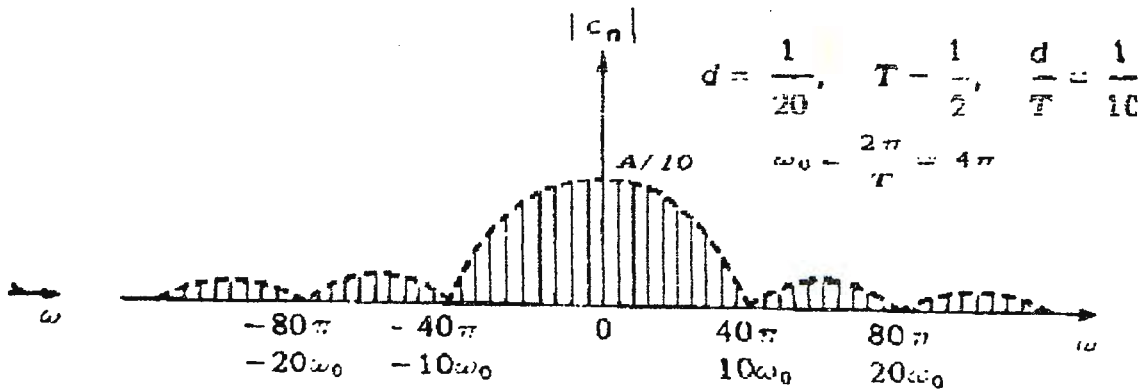


Figura 1.1. Espectro de frecuencia de un tren de pulsos periódico.[4]

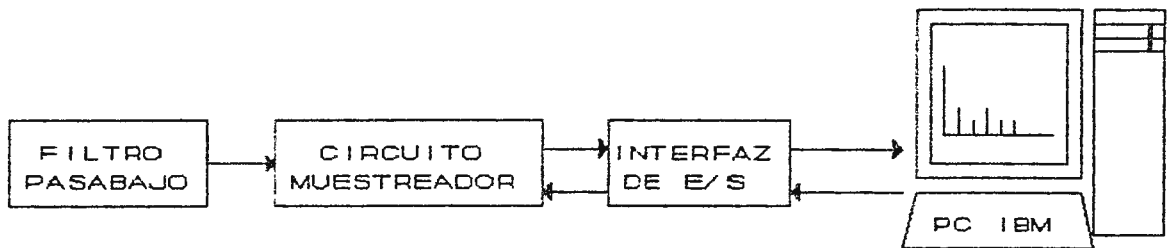


Figura 1.2. Principio de funcionamiento del equipo de laboratorio para la demostración de Fourier.

filtro digital es un procedimiento numérico o algoritmo que transforma una secuencia dada de números en una segunda secuencia que tiene muchas propiedades deseables tales como un menor ruido y distorsión.

Para poder obtener una gráfica del espectro de frecuencia de una señal a través de métodos matemáticos, existen dos opciones para implementar el análisis de las muestras: a través de filtros digitales y por procesamiento de señales.

Un filtro digital consiste en la interconexión de tres elementos simples: el *sumador*, *multiplicadores* y *retrasos*. Estos son presentados en la Figura 1.3. Los sumadores y los multiplicadores son componentes simplemente conceptuales que ya están implementados en la unidad lógica aritmética de la PC.[3]

El análisis a través de filtros digitales puede llevarse a cabo

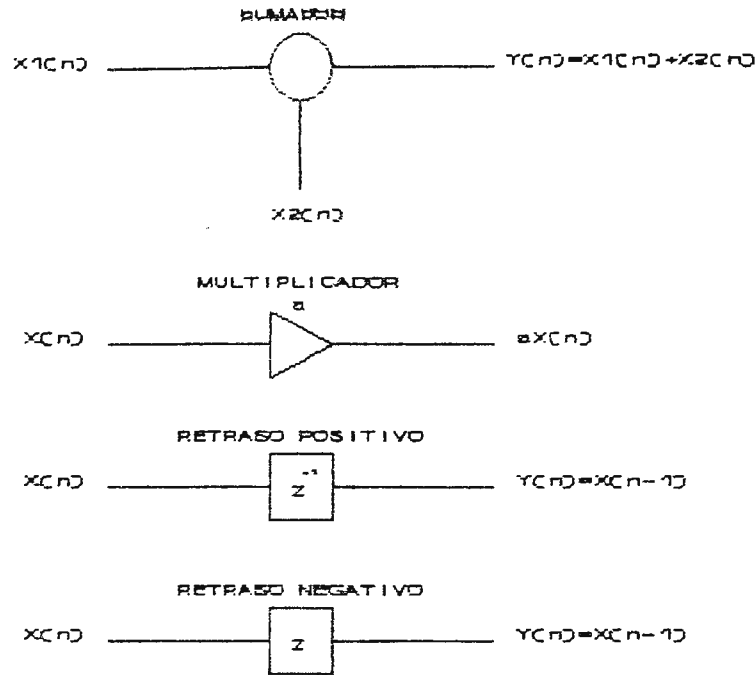


Figura 1.3. Elementos básicos de un filtro digital.

mediante la programación de un filtro digital del tipo de ranura con una banda angosta y capaz de desplazar su frecuencia central a través del rango de frecuencias en que se desea realizar el análisis espectral a fin de detectar el nivel de potencia de las componentes espectrales presentes en cada región del espectro en que se sitúe. Esto requiere de cierto nivel de conocimiento acerca de la programación de los filtros, así como del cálculo y ajuste de algunos coeficientes empleados en ellos.

La otra opción -para el caso la opción seleccionada por el grupo- consiste en obtener una serie de muestreos de la señal de la cual se desea conocer el espectro. El muestreo debe ser realizado a una frecuencia mucho mayor que la frecuencia de la señal a analizar. Además, debe asegurarse que el tiempo de muestreo sea el suficiente como para poder abarcar al menos un período completo de la señal; de

ésta manera, podemos estar seguros de obtener la suficiente información para describir el comportamiento de la señal. Por tanto, esto nos obliga a restringir el rango de frecuencia de operación del aparato de acuerdo con la máxima frecuencia de muestreo que la computadora o los componentes del circuito sean capaces de manejar. El procedimiento básico utilizado para el cálculo del espectro en frecuencia es la transformada discreta de Fourier. Por brevedad, de ahora en adelante nos referiremos a ella por sus iniciales (TDF). Otro procedimiento fundamental que se utiliza es el de la transformada rápida de Fourier. Al igual que el anterior, se hará referencia a ella por sus iniciales (TRF). La TRF no es más que un procedimiento de optimización de la TDF para obtener un resultado similar con un número significativamente menor de operaciones de cálculo.^[3]

Ambos procedimientos requieren de una serie de datos de entrada que reflejen el comportamiento de la señal de entrada en el tiempo. Para obtener la secuencia de entrada se utiliza una tarjeta de interface que realiza un muestreo de la señal a intervalos iguales de tiempo y luego traduce éstas muestras en datos manejables por la computadora. En otras palabras, se digitaliza la señal de entrada.

El concepto de interface que se maneja es el de los circuitos utilizados para la interconexión o unión de los subsistemas dentro de un sistema. En nuestro caso, la interface consiste en una tarjeta electrónica que convierte una señal eléctrica de carácter analógico de entrada proveniente de un generador de señales comercial cualquiera, en una serie de muestreos digitalizados para luego introducirlos a la computadora.

1.4.2. Situación actual.

La Universidad Don Bosco cuenta entre sus equipos de laboratorio con dos analizadores espectrales didácticos marca LabVolt que utiliza para sus prácticas de laboratorio en telecomunicaciones y que, eventualmente, pueden ser empleados para la demostración física de la transformada de Fourier. Sin embargo, no todas las instituciones pueden tener acceso a tales equipos para realizar prácticas de laboratorio debido a la alta inversión que ello representa en un equipo que estaría sub-utilizado.

Actualmente en el país, los analizadores espectrales son utilizados principalmente en el campo de las telecomunicaciones por todas aquellas empresas, estatales o privadas, que trabajan de alguna manera con transmisión y recepción de señales; Tal es el caso de ANTEL, SITA, INSATELSA, TELECOM, ALCATEL, los diferentes canales de TV. , etc.

Por tanto, los analizadores espectrales con que cuentan la mayoría de éstas entidades están ya diseñados para proveer al usuario de diferentes funciones especiales que complican la arquitectura y funcionamiento del aparato. Principalmente, existen los que funcionan bajo el principio del tratamiento de señal por hardware. Los analizadores que emplean el tratamiento de señales por software a nivel nacional existen, aunque en cantidad reducida. Sin embargo, el principio de operación de estos equipos no varía en mucho en cuanto a que ambos conllevan a obtener el mismo espectro en frecuencia de una señal eléctrica.

Tratando de resolver la falta a nivel nacional de equipos de

laboratorio para realizar dichas prácticas, proponemos solucionar este problema creando una tarjeta electrónica que funcione acoplada a los puertos de expansión de una computadora IBM y que sea manejada por un software que a la vez procese matemáticamente la información obtenida por ésta.

CAPITULO II
LA TRANSFORMADA DE FOURIER.

2.1. Series de Fourier.

2.1.1. Funciones Periódicas

Una función periódica se puede definir como una función para la cual

$$f(t) = f(t + T) \quad (2.1)$$

para todo valor de t . La constante mínima T que satisface la relación (2.1) se llama el *período* de la función. Mediante repetición de (2.1), se obtiene,

$$f(t) = f(t + nT), \quad n = 0, \pm 1, \pm 2, \dots \quad (2.2)$$

2.1.2. Series de Fourier

Sea la función $f(t)$ una función periódica de período T , la cual se puede representar por la serie trigonométrica:

$$\begin{aligned} f(t) &= \frac{1}{2}a_0 + a_1 \cos \omega_0 t + a_2 \cos 2\omega_0 t + \dots + b_1 \sen \omega_0 t + b_2 \sen 2\omega_0 t + \dots \\ &= \frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \sen n\omega_0 t) \end{aligned} \quad (2.3)$$

donde $\omega_0 = 2\pi/T$.

Una serie como la representada por (2.3) se le llama *serie trigonométrica de Fourier*.

Para evaluar un par de coeficientes trigonométricos, a_n y b_n , se utilizan las siguientes expresiones:

$$a_n = \frac{2}{T} \int_0^T f(t) \cos n\omega_0 t \, dt \quad (2.4)$$

y

$$b_n = \frac{2}{T} \int_0^T f(t) \sen n\omega_0 t \, dt \quad (2.5)$$

Esta serie también se puede representar así:

$$f(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(n\omega_0 t - \theta_n) \quad (2.6)$$

Según (2.6), es obvio que la representación en series de Fourier de una función periódica, representa la función periódica como la suma de componentes senoidales que tienen diferentes frecuencias. La componente senoidal de frecuencia $\omega_n = n\omega_0$ se denomina la *enésima armónica* de la función periódica. La primera armónica comunmente se conoce como la *componente fundamental* porque tiene el mismo período de la función y $\omega_0 = 2\pi f_0 = 2\pi/T$ se conoce como la *frecuencia angular fundamental*. Los coeficientes C_n y los ángulos θ_n se conocen como *amplitudes armónicas y ángulos de fase*, respectivamente.

2.1.3. Propiedades de las Funciones Seno y Coseno: Funciones Ortogonales:

Un conjunto de funciones $\phi_k(t)$ es ortogonal en un intervalo $a < t < b$ si para dos funciones cualesquiera $\phi_m(t)$ y $\phi_n(t)$ pertenecientes al conjunto $\phi_k(t)$, cumple:

$$\int_a^b \phi_m(t) \phi_n(t) dt = 0 \quad \text{para } m \neq n \text{ y } r_n \quad \text{para } m=n \quad (2.7)$$

Considérese, por ejemplo, un conjunto de funciones senoidales; mediante el cálculo elemental se puede demostrar que

$$\int_{-T/2}^{T/2} \cos(m\omega_0 t) dt = 0 \quad \text{para } m \neq 0 \quad (2.8a)$$

donde $\omega_0 = 2\pi/T$.

$$\int_{-T/2}^{T/2} \text{sen}(m\omega_0 t) dt = 0 \quad \text{para todo valor de } m \quad (2.8b)$$

$$\int_{-T/2}^{T/2} \cos(n\omega_0 t) \cos(m\omega_0 t) dt = 0 \quad \text{para } m \neq n \quad (2.9a)$$

$$\int_{-T/2}^{T/2} \cos(n\omega_0 t) \cos(n\omega_0 t) dt = T/2 \quad \text{para } m=n \neq 0 \quad (2.9b)$$

$$\int_{-T/2}^{T/2} \text{sen}(m\omega_0 t) \text{sen}(n\omega_0 t) dt = 0 \quad m \neq n \quad (2.10a)$$

$$\int_{-T/2}^{T/2} \text{sen}(m\omega_0 t) \text{sen}(n\omega_0 t) dt = T/2 \quad m=n \neq 0 \quad (2.10b)$$

$$\int_{-T/2}^{T/2} \text{sen}(m\omega_0 t) \cos(n\omega_0 t) dt = 0 \quad \text{para todo valor de } n \text{ y } m \quad (2.11)$$

Estas relaciones demuestran que las funciones:

$$1, \cos\omega_0 t, \cos 2\omega_0 t, \dots, \cos n\omega_0 t, \dots, \text{sen}\omega_0 t, \text{sen} 2\omega_0 t, \dots, \text{sen } n\omega_0 t, \dots$$

forman un conjunto de funciones ortogonales en el intervalo $-T/2 < t < T/2$. [4]

2.1.4. Forma compleja de las series de Fourier.

Se ha visto que la amplitud de cada componente de frecuencia depende tanto de a_n como de b_n ; es decir, tanto el término seno como el término coseno, ambos contribuyen a la amplitud. La expresión exacta para esta amplitud es $\sqrt{a_n^2 + b_n^2}$. También es posible obtener directamente la amplitud usando una forma de la serie de Fourier en la cual cada término es función coseno con un ángulo de fase; la amplitud y el ángulo de fase son funciones de $f(t)$ y n . Una forma todavía más conveniente y concisa de las series de Fourier se obtiene si los senos y cosenos se expresan como funciones exponenciales con constantes multiplicativas complejas.

Tómese primero la forma trigonométrica de las series de Fourier:

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \text{sen } n\omega_0 t)$$

y luego sustitúyanse las formas exponenciales para el seno y el

coseno. Después de reordenar,

$$f(t) = a_0 + \sum_{n=1}^{\infty} \left(e^{jn\omega_0 t} \frac{a_n - jb_n}{2} + e^{-jn\omega_0 t} \frac{a_n + jb_n}{2} \right)$$

Ahora se define una constante compleja:

$$c_n = \frac{1}{2} (a_n - jb_n) \quad (n = 1, 2, 3, \dots) \quad (2.12)$$

Los coeficientes a_n , b_n y c_n dependen todos de n y de $f(t)$. Supóngase ahora que n se sustituye por $(-n)$; ¿Cómo cambian ahora los valores de las constantes? Los coeficientes a_n y b_n están definidos por (2.4) y (2.5), y es evidente que

$$a_{-n} = a_n$$

pero

$$b_{-n} = -b_n$$

de la ecuación (2.12), entonces,

$$c_{-n} = \frac{1}{2} (a_n + jb_n) \quad (n = 1, 2, 3, \dots) \quad (2.13)$$

Así,

$$c_n = c_{-n}^*$$

También se hace

$$c_0 = a_0$$

Por tanto $f(t)$ puede expresarse como

$$f(t) = c_0 + \sum_{n=1}^{\infty} c_n e^{jn\omega_0 t} + \sum_{n=1}^{\infty} c_{-n} e^{-jn\omega_0 t}$$

o

$$f(t) = \sum_{n=0}^{\infty} c_n e^{jn\omega_0 t} + \sum_{n=1}^{\infty} c_{-n} e^{-jn\omega_0 t}$$

Finalmente, en lugar de sumar la segunda serie los enteros positivos de 1 a ∞ , se sumará sobre los enteros negativos, de -1 a $-\infty$:

$$f(t) = \sum_{n=0}^{\infty} c_n e^{jn\omega_0 t} + \sum_{N=-1}^{-\infty} c_N e^{jN\omega_0 t}$$

o

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t} \quad (2.14)$$

Por convención, se sobreentiende que una suma de $-\infty$ a ∞ incluye un término para $n = 0$.

La ecuación (2.14) es de la forma compleja de la serie de Fourier para $f(t)$; su forma concisa es una de las razones más importantes por las cuales se usa. Para obtener la expresión con la cual pueda evaluarse un coeficiente complejo en particular c_n , se sustituyen las ecuaciones (2.4) y (2.5) en la (2.12):

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \cos n\omega_0 t dt - j \frac{1}{T} \int_{-T/2}^{T/2} f(t) \operatorname{sen} n\omega_0 t dt$$

Se utilizan los equivalentes exponenciales del seno y del coseno, y se simplifica:

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jn\omega_0 t} dt \quad (2.15)$$

Así, una sola ecuación compacta sustituye a las dos ecuaciones requeridas para la forma trigonométrica de las series de Fourier. En lugar de evaluar dos integrales para encontrar los coeficientes de Fourier, se necesita una sola integración; además, casi siempre se trata de una integral más simple. Debe observarse que la integral de la ecuación (2.15) contiene el factor $1/T$, mientras que las integrales para a_n y b_n contienen el factor $2/T$.

Reuniendo las dos relaciones básicas para la forma exponencial de las series de Fourier, se tiene

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t} \quad (2.16)$$

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jn\omega_0 t} dt \quad (2.17)$$

donde, como de costumbre, $\omega_0 = 2\pi/T$.

La amplitud de la componente de la serie exponencial de Fourier a $\omega = n\omega_0$, donde $n = 0, \pm 1, \pm 2, \dots$, es $|c_n|$. Puede graficarse un espectro de frecuencia discreto dando $|c_n|$ contra $n\omega_0$ o $n\omega_0$ con una abscisa

que muestre los valores positivos y negativos, y cuando esto se haga, la gráfica será simétrica con respecto al origen, ya que las ecuaciones (2.14) y (2.15) muestran que $|c_n| = |c_{-n}|$.

También se observa en las ecuaciones (2.16) y (2.17) que la amplitud de la componente senoidal a $\omega = n\omega_0$, donde $n = 1, 2, 3, \dots$, es

$$\sqrt{a_n^2 + b_n^2} = 2|c_n| = 2|c_{-n}| = |c_n| + |c_{-n}|.$$

Para la componente de cd, $a_0 = a_0$.

Se considerará un ejemplo numérico para ilustrar el uso de la forma compleja de las series de Fourier. Un tren de pulsos rectangulares de amplitud V_0 y duración τ , repitiéndose periódicamente cada T segundos, se muestra en la Figura 2.1:

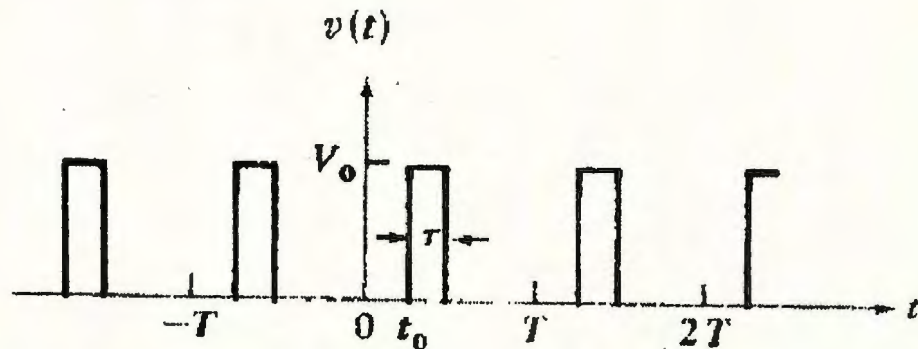


Figura 2.1. Sucesión periódica de pulsos rectangulares.[4]

Por lo tanto la frecuencia fundamental es

$$f_0 = \frac{1}{T}$$

El valor de un coeficiente complejo general se encuentra primero a partir de la ecuación (2.15):

$$\begin{aligned} c_n &= \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jn\omega_0 t} dt \\ &= \frac{V_0}{T} \int_{t_0}^{t_0+\tau} e^{-jn\omega_0 t} dt \end{aligned}$$

$$\begin{aligned}
&= \frac{V_0}{-jn\omega_0 T} (e^{-jn\omega_0(t_0+\tau)} - e^{-jn\omega_0 t_0}) \\
&= \frac{2V_0}{n\omega_0 T} e^{-jn\omega_0(t_0+\tau/2)} \operatorname{sen}\left(\frac{1}{2}n\omega_0\tau\right) \\
&= \frac{V_0\tau}{T} \frac{\operatorname{sen}\left(\frac{1}{2}n\omega_0\tau\right)}{\frac{1}{2}n\omega_0\tau} e^{-jn\omega_0(t_0+\tau/2)}
\end{aligned}$$

Entonces la magnitud de c_n es

$$|c_n| = \frac{V_0\tau}{T} \left| \frac{\operatorname{sen}\left(\frac{1}{2}n\omega_0\tau\right)}{\frac{1}{2}n\omega_0\tau} \right| \quad (2.18)$$

y el ángulo de c_n es

$$\operatorname{ang} c_n = -n\omega_0\left(t_0 + \frac{\tau}{2}\right) \quad (\text{posiblemente más } 180^\circ) \quad (2.19)$$

El factor trigonométrico en la ec. (2.18) aparece frecuentemente en la teoría moderna de comunicaciones, y recibe el nombre de *función de muestreo*. El "muestreo" se refiere a la función del tiempo de la Figura 2.2 de la cual se deriva la función de muestreo. El producto de esta sucesión de pulsos y cualquier otra función $f(t)$ representa *muestras* de $f(t)$ cada T segundos si τ es pequeño y $V_0 = 1$. Se define

$$Sa(x) = \frac{\operatorname{sen} x}{x}$$

A causa de la forma en que ayuda a determinar la amplitud de las varias componentes de frecuencia en $v(t)$, vale la pena investigar las características importantes de esta función. Primero, se observa que $Sa(x)$ vale cero siempre que x sea múltiplo entero de π ; es decir,

$$Sa(n\pi) = 0 \quad n = 1, 2, 3, \dots$$

Cuando x vale cero, la función es indeterminada, pero es fácil mostrar que su valor es uno:

$$Sa(0) = 1$$

Por tanto la magnitud de $Sa(x)$ disminuye desde uno en $x = 0$ hasta cero en $x = \pi$. Conforme x aumenta desde π hasta 2π , $|Sa(x)|$ aumenta desde cero hasta un máximo menor que uno, y luego cae de nuevo a cero. Conforme x continúa aumentando, los máximos sucesivos se van haciendo más pequeños continuamente ya que el numerador de $Sa(x)$ no puede exceder a uno y el denominador aumenta continuamente. Además, $Sa(x)$ tiene simetría par.

Ahora se obtendrá el espectro de líneas. Se considera primero $|c_n|$, escribiendo (2.18) en términos de la frecuencia cíclica fundamental f_0 :

$$|c_n| = \frac{V_0 \tau}{T} \left| \frac{\text{sen}(n\pi f_0 \tau)}{n\pi f_0 \tau} \right| \quad (2.20)$$

La amplitud de cualquier c_n se obtiene de (2.20) usando los valores conocidos τ y $T = 1/f$, y eligiendo el valor deseado de n , $n = 0, \pm 1, \pm 2, \dots$. En vez de evaluar (2.20) a estas frecuencias discretas, se trazará la *envolvente* de $|c_n|$ considerando que la frecuencia $n f_0$ es una variable continua. Es decir, f , que es $n f_0$, puede tomar realmente sólo valores discretos de las frecuencias armónicas $0, \pm f_0, \pm 2f_0, \pm 3f_0$, etc., pero por el momento puede pensarse n es una variable continua. Cuando f vale cero, $|c_n|$ evidentemente vale $V_0 \tau / T$, y cuando f ha aumentado a $1/\tau$, $|c_n|$ vale cero. La envolvente resultante se ha graficado como la línea a trazos de la Figura 2.2. A continuación se obtiene el espectro de líneas levantando simplemente rayas verticales a cada frecuencia armónica, como se ve en la gráfica, las amplitudes mostradas son las de c_n . El caso particular graficado se aplica al caso en el que $\tau/T = 1/(1.5\pi) = 0.212$. En este ejemplo, sucede que no hay armónico

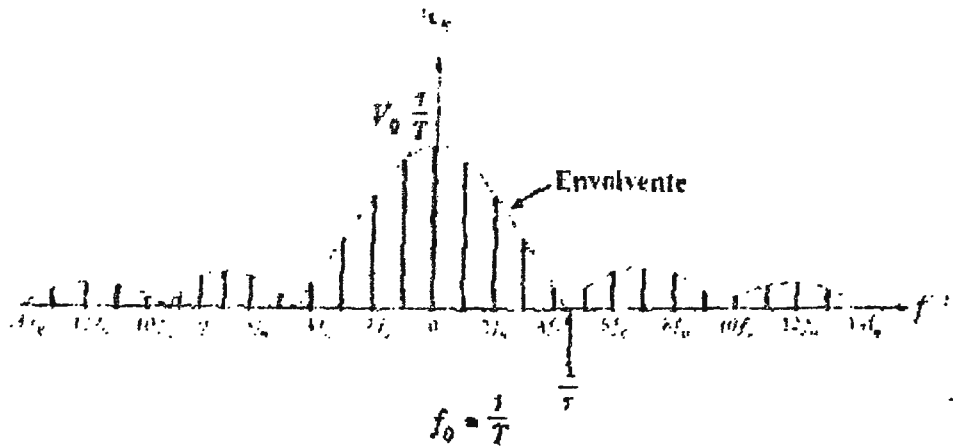


Figura 2.2. El espectro de líneas discreto correspondiente para $|c_n|$, $f=nf_0$, y $n=0, \pm 1, \pm 2, \dots$ [4]

exáctamente en la frecuencia a la cual la envolvente vale cero; sin embargo, esto podría lograrse mediante una elección diferente de τ o de T .

En la Figura 2.3 la amplitud de la componente senoidal se ha graficado en función de la frecuencia. Obsérvese de nuevo que $a_0 = c_0$

$$\text{y } \sqrt{a_n^2 + b_n^2} = |c_n| + |c_{-n}|. [5]$$

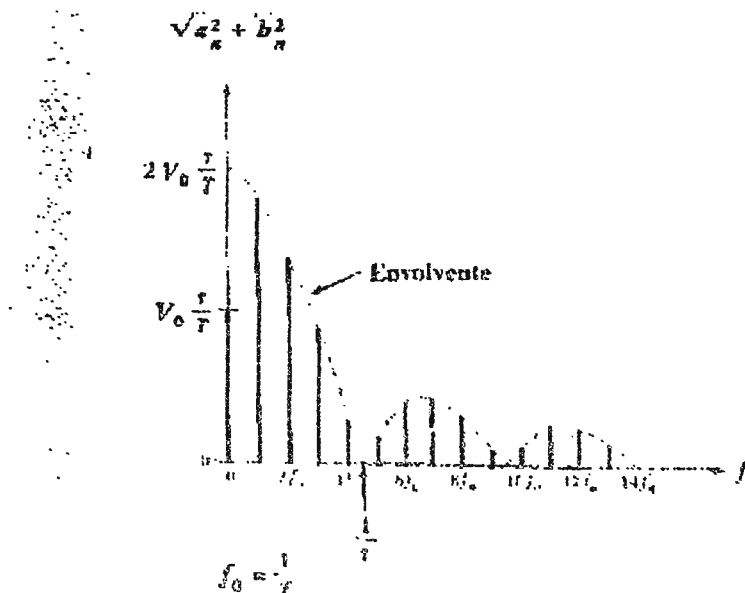


Figura 2.3. El espectro de líneas discreto correspondiente para $(a_n^2 + b_n^2)^{1/2}$, donde $f=nf_0$, para $n=0, 1, 2, \dots$ [4]

2.2. Transformada de Fourier.

La función $F(\omega)$ definida por $F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$ se conoce como la *integral de Fourier o transformada de Fourier* de $f(t)$, y la operación de integración se simboliza frecuentemente por \mathcal{F} ; esto es,

$$F(\omega) = \mathcal{F}[f(t)] = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (2.21)$$

Análogamente \mathcal{F}^{-1} es el simbolo que se utiliza para indicar la operación inversa o sea obtener $f(t)$ cuando $F(\omega)$ está dado; esto es,

$$f(t) = \mathcal{F}^{-1}[F(\omega)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (2.22)$$

y $f(t)$ se denomina *transformada inversa de Fourier* de $F(\omega)$. Las ecuaciones (2.21) y (2.22) se conocen a menudo como *par de transformadas de Fourier*.

La condición para que exista $F(\omega)$ generalmente está dada por:

$$\int_{-\infty}^{\infty} |f(t)| dt < \infty \quad (2.23)$$

En otros términos, la integral del valor absoluto de $f(t)$ debe ser infinita.

Se debe observar que (2.23) es una condición suficiente pero no necesaria para la existencia de $\mathcal{F}[f(t)]$; las funciones que no satisfacen (2.23) pueden tener transformada de Fourier; estas funciones se verán más adelante.

La función $F(\omega) = \mathcal{F}[f(t)]$ es, en general, compleja y, se tiene

$$F(\omega) = R(\omega) + j X(\omega) = |F(\omega)| e^{j\phi(\omega)} \quad (2.24)$$

donde $|F(\omega)|$ se denomina *espectro de magnitud* de $f(t)$, y $\phi(\omega)$ *espectro de fase* de $f(t)$. [4]

2.2.1. Interpretación de las transformadas de Fourier.

Si se supone que $f(t)$ es periódica con período T , entonces $f(t)$ se puede expresar como

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t}, \quad \omega_0 = \frac{2\pi}{T} \quad (2.25)$$

donde

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jn\omega_0 t} dt \quad (2.26)$$

Si ahora se considera que a medida que $T \rightarrow \infty$, $\omega_0 \rightarrow \Delta\omega = 2\pi\Delta f$, $\Delta f = 1/T$, entonces (2.25) y (2.26) se convierten, respectivamente, en

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j(n\Delta\omega)t}, \quad (2.27)$$

$$c_n = \Delta f \int_{-T/2}^{T/2} f(t) e^{-j(n\Delta\omega)t} dt \quad (2.28)$$

Si $\Delta\omega \rightarrow 0$, $n \rightarrow \infty$ tal que $n\Delta\omega \rightarrow \omega$. En otros términos, en el límite, en vez de tener armónicos discretos correspondientes a $n\omega_0$, todo valor de ω es permitido. De esta manera, en vez de C_n se tiene $C(\omega)$, y por (2.28), se tiene que

$$\lim_{\Delta f} \frac{c(\omega)}{\Delta f} = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt = F(\omega) \quad (2.29)$$

según (2.29) se observa que

$$F(\omega) d\omega = c(\omega) \quad (2.30)$$

o, puesto que $\omega = 2\pi f$, se tiene

$$\frac{1}{2\pi} F(\omega) d\omega = c(\omega) \quad (2.31)$$

Entonces (2.27) se convierte en

$$f(t) = \int_{-\infty}^{\infty} \frac{1}{2\pi} F(\omega) d\omega e^{j\omega t}$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (2.32)$$

Esta ecuación muestra que $\frac{1}{2} \pi |F(\omega)| d\omega$ representa la magnitud infinitesimal de un armónico a la frecuencia angular ω . Estos armónicos tienen frecuencia fundamental cero ($\omega_0 \rightarrow d\omega$) y están separados por infinitesimos. Aunque $|F(\omega)| d\omega$ es infinitesimal, $F(\omega)$ es finito; por esta razón a la gráfica $|F(\omega)|$ vs ω se le denomina *espectro continuo* y a $|F(\omega)|$ se le denomina generalmente, *espectro de magnitud* de $f(t)$.

La representación anterior de una función no periódica como suma de exponenciales con la frecuencia fundamental tendiendo a cero, no es un concepto fácil de aceptar.

A veces la interpretación que sigue del par de transformadas de Fourier

(2.21) y (2.22) es más directa y de mayor significado:

$$F(\omega) = \mathcal{F}[f(t)] = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (2.21)$$

$$f(t) = \mathcal{F}^{-1}[F(\omega)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} dt \quad (2.22)$$

Es decir, se supone que cualquier función dada tiene dos modos equivalentes de representación: uno en el dominio del tiempo, $f(t)$, y el otro en el dominio de la frecuencia, $F(\omega)$. La ecuación (2.21) transforma la función $f(t)$, en el dominio, a su función equivalente $F(\omega)$, en el dominio de la frecuencia, y la ecuación (2.22) invierte el proceso. La ecuación (2.21) analiza la función del tiempo en un espectro de frecuencia y la ecuación (2.22) sintetiza el espectro de frecuencia para obtener nuevamente la función en términos del

tiempo. [4]

2.2.2. Propiedades de las transformadas de Fourier.

Si $F_1(\omega) = \mathcal{F}[f_1(t)]$ y $F_2(\omega) = \mathcal{F}[f_2(t)]$, y a_1 y a_2 son dos constantes arbitrarias, entonces

$$\mathcal{F}[a_1 f_1(t) + a_2 f_2(t)] = a_1 F_1(\omega) + a_2 F_2(\omega) \quad (2.33)$$

La ecuación (2.33) es la *propiedad de linealidad* de la transformada de Fourier.

Si a es una constante real y $F(\omega) = \mathcal{F}[f(t)]$ entonces:

$$\mathcal{F}[f(at)] = \frac{1}{|a|} F\left(\frac{\omega}{a}\right) \quad (2.34)$$

Donde la ecuación (2.34) es la *propiedad de escalonamiento* de la transformada de Fourier.

La función $f(at)$ representa la función $f(t)$ contraída en la escala de tiempo por un factor a . Análogamente la función $F\left(\frac{\omega}{a}\right)$ representa la

función $F(\omega)$ expandida en la escala de frecuencia por el mismo factor a . La propiedad de escalonamiento, por consiguiente, afirma que la contracción en el tiempo es equivalente a la expansión en el dominio de la frecuencia y viceversa.

$$\mathcal{F}[f(t-t_0)] = F(\omega) e^{-j\omega t_0} \quad (2.35)$$

La ecuación (2.35) es la *propiedad de desplazamiento en el tiempo* de la transformada de Fourier.

Si ω_0 es una constante real y $F(\omega) = \mathcal{F}[f(t)]$, entonces:

$$\mathcal{F}[f(t) e^{j\omega_0 t}] = F(\omega - \omega_0) \quad (2.36)$$

La ecuación (2.36) es la *propiedad de desplazamiento en la frecuencia*

de la transformada de Fourier.

Si $F(\omega) = \mathcal{F}[f(t)]$, entonces:

$$\mathcal{F}[f(t)] = 2\pi f(-\omega) \quad (2.37)$$

La ecuación (2.37) es la *propiedad de simetría* de la transformada de Fourier. [4]

CAPITULO III
HERRAMIENTAS MATEMATICAS UTILIZADAS EN EL
DESARROLLO DEL PROYECTO.

3.1. Fundamentos del procesamiento de señales digitales.

Las señales que se encuentran en la práctica pueden o no coincidir en su forma con las funciones matemáticas de prueba. Tomemos por ejemplo el caso de un control de temperatura para un cuarto. La señal empleada para fijar la temperatura deseada en el cuarto es una función escalón. La forma de onda de la energía eléctrica presente en los tomacorrientes eléctricos de la casa es de forma senoidal con una frecuencia de 60 Hz. Estas señales se aproximan a la forma de las funciones matemáticas escalón y seno. Por el contrario, si vemos el caso de las señales mostradas en un electrocardiograma (EKG) (Ver Figura 3.1(a)), éstas no poseen una forma de onda que se ajuste a alguno de los patrones generados matemáticamente; sin embargo, muestra cierta periodicidad. Las señales mostradas en la Figura 3.1(b) muestran los patrones de señales típicos de un encefalograma (EEG) desde un estado de excitación hasta un estado de coma. Tales señales no muestran periodicidad alguna y sus formas de onda son extremadamente complicadas.

El procesamiento de señales tiene muchas aplicaciones. A partir de las irregularidades de las señales de un EKG, es posible detectar defectos en el corazón del paciente. La localización de submarinos bajo el océano es posible gracias a que cada submarino genera una onda acústica distinta debido a la rotación de sus propulsores, la

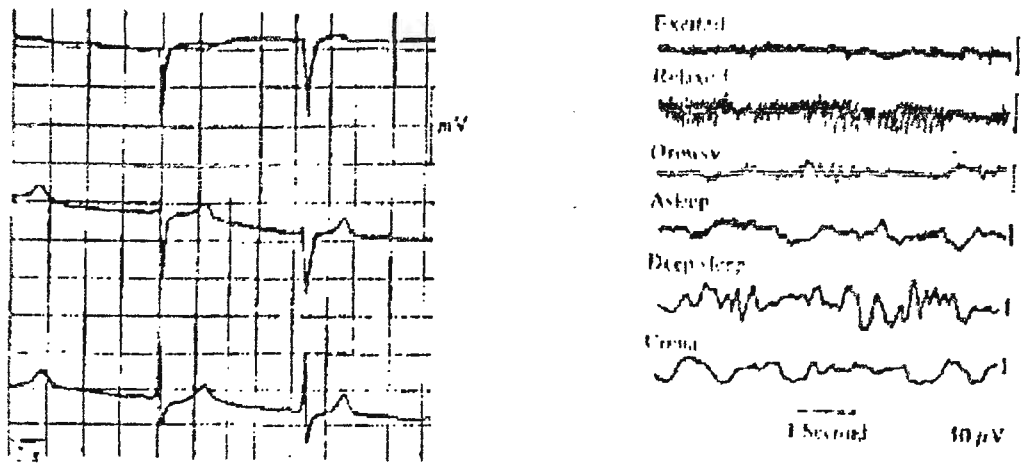


Figura 3.1. Ejemplo de señales de tiempo continuo.[1]

vibración del motor y la vibración del casco. Las abejas africanizadas o abejas "asesinas" y las abejas domésticas son casi idénticas en su tamaño y aspecto físico y la única manera de diferenciarlas es realizando un cuidadoso análisis con microscopio. Recientemente se ha descubierto que existe una diferencia en la frecuencia con que las abejas baten sus alas, con lo que consecuentemente generan minúsculas señales acústicas distintas las cuales pueden emplearse en la identificación de un enjambre. Una reciente e importante aplicación del análisis de señales es la eliminación de ruidos indeseables. Por ejemplo, en una planta industrial existen muchos ruidos indeseables que afectan la comunicación. Tales ruidos son el rugir de los motores, el zumbido de los transformadores de potencia y de los ventiladores industriales y la vibración de las maquinarias pesadas. Este tipo de ruidos son periódicos y pueden descomponerse en sus componentes básicas (Figura 3.2(a)). Por medio de una computadora, éste ruido puede ser analizado, y descompuesto en sus componentes básicas, y generar entonces a través de un altoparlante una imagen de espejo del ruido conocida como antiruido, tal como lo muestra la Figura 3.2(b). Esto cancelará

el ruido sin afectar la conversación humana. Esta técnica antiruido puede emplearse en los sistemas de ventilación inyectando señales antiruido en los ductos de ventilación en lugar de utilizar pesados aislamientos. Sin embargo, ésta técnica no es aplicable si el ruido es aleatorio o carece de patrones repetitivos.

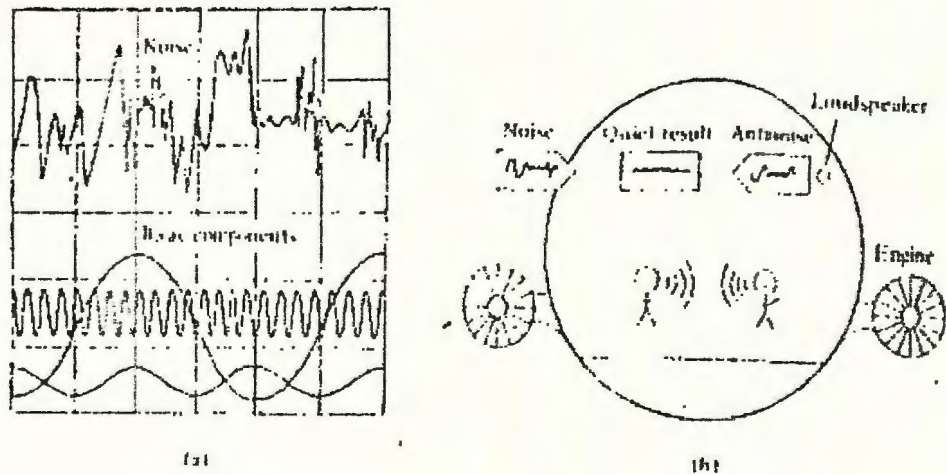


Figura 3.2. Descomposición y eliminación del ruido repetitivo.[1]

Una señal es llamada determinística si puede ser definida con toda certeza. Este tipo de señales puede ser reproducida de forma exacta y repetitiva. Una señal es llamada aleatoria si no puede ser descrita con toda precisión antes que ocurra.

Este tipo de señales generalmente no puede ser reproducida ni repetida con exactitud. El estudio de éstas señales aleatorias requiere de conocimientos sobre procesos estocásticos y de teoría de probabilidades, las cuales están fuera del análisis de éste texto. Se estudiarán entonces únicamente señales determinísticas.

En general, el análisis de señales está ampliamente relacionado con la descomposición de las mismas en términos de funciones de tiempo elementales, llamadas funciones bases. Un caso particular es el

análisis de fourier, el cual está basado en la descomposición de señales en términos de funciones sencidales (Figura 3.3). Otra forma de representar las señales es a través de funciones exponenciales complejas $e^{jk\omega_0 t}$. El análisis a desarrollar para determinar el espectro de frecuencia de una señal está basado en dicho tipo de funciones.[1]

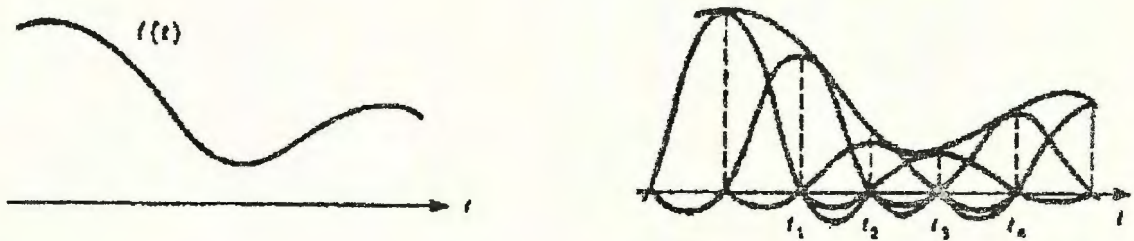


Figura 3.3. Funciones bases.[4]

3.1.1. Digitalización de señales.

El muestreo de señales continuas a intervalos periódicos se ha convertido en una importante operación, tanto práctica como matemáticamente. Algo importante a considerar en el muestreo de señales es el hecho que la mayoría de los sistemas poseen limitantes en su respuesta en frecuencia; esto es, solamente pueden responder dentro de un límite de frecuencia. Como resultado, la señal de salida de éstos sistemas está limitada en su ancho de banda; por ejemplo, los aparatos telefónicos hogareños ordinarios poseen un límite de frecuencia superior de cerca de 4000 Hz y las señales de televisión uno de 4 MHz. Una consecuencia importante de una señal con un ancho

de banda finito es que puede ser fielmente representada por una secuencia de muestras de corta duración tomadas en instantes discretos y periódicos. Esta secuencia de muestras es utilizada para digitalizar dichas señales.

La digitalización es un proceso para convertir las señales analógicas a un formato digital. La forma más común, consiste en asignar un código binario que representa un valor discreto de la amplitud de la muestra, donde una muestra es el valor instantáneo de la señal en un tiempo discreto. Esta operación es llamada "codificación" de las muestras. Todas las amplitudes comprendidas entre dos valores discretos codificables sucesivos se representan por la misma palabra binaria. Esta otra operación es conocida como cuantificación.

Procederemos primeramente a establecer la relación entre el espectro de tiempo discreto y el de tiempo continuo y luego nos presentaremos las respectivas demostraciones matemáticas involucradas. Definiremos la señal de tiempo continuo, por $C_A(t)$. Su espectro estará dado entonces por $C_A(j\Omega)$, donde Ω representa la frecuencia de tiempo continuo con unidades en radianes por segundo. Para señales de tiempo continuo, la transformada de Fourier está definida por

$$C_A(j\Omega) = \int_{-\infty}^{\infty} C_A(t) e^{-j\Omega t} dt \quad (3.1)$$

definición la cuál es bastante similar a la transformada de Fourier de señales de tiempo discreto a excepción que en lugar de una sumatoria se emplea una integral. Cuando $C_A(t)$ es muestreada cada T_s segundos, obtenemos una secuencia de tiempo discreto $C_A(nT_s)$, cuyos valores son iguales a

$$C_A(nT_s) = C_A(t) \Big|_{t=nT_s} \quad (3.2)$$

T_s es comunmente llamado *período de muestreo*, el cual se expresa en unidades de segundos por intervalo de muestreo. La transformada de Fourier para tales secuencias de datos está definida por

$$C(e^{j\omega}) = \sum_{n=-\infty}^{\infty} C_A(nT_s) e^{-jn\omega} \quad (3.3)$$

donde ω tiene como unidades radianes por intervalo de muestreo. Este espectro de tiempo discreto está relacionado al espectro de tiempo continuo $C_A(j\Omega)$ por

$$C(e^{j\omega}) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} C_A(j\omega/T_s + j2\pi k/T_s) \quad (3.4)$$

Descrito en palabras, el espectro de una secuencia de tiempo discreto, derivado del muestreo de una función de tiempo continuo, es igual a la suma de un conjunto infinito de espectros de tiempo continuo desplazados. Una interpretación gráfica de ésta entidad se muestra en la Figura 3.4.

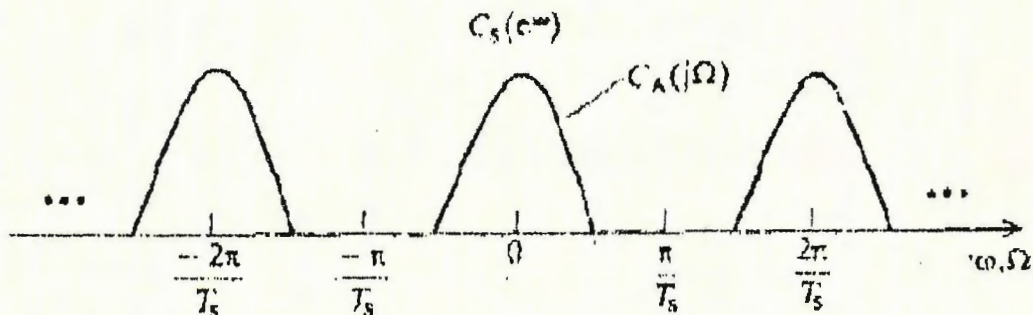


Figura 3.4. Espectro de una secuencia de tiempo discreto obtenido a partir del muestreo de una señal de tiempo continuo.[3]

Demostración:

Para mostrar esta relación, nos valdremos de la siguiente aproximación. Primero, trataremos el muestreo como una operación multiplicativa que involucra la señal de tiempo continuo $C_A(t)$ y un tren de funciones delta que ocurren cada T_s segundos. Esta multiplicación produce la función intermedia de tiempo continuo $C_g(t)$. El espectro continuo de $C_g(t)$ denotado por $C_g(j\Omega)$ posee la misma forma analítica de $C_A(e^{j\omega})$. Finalmente, determinamos $C_g(j\Omega)$ en términos del espectro analógico original $C_A(j\Omega)$.

Primeramente definiremos la función delta de Dirac y presentaremos sus propiedades que son importantes para ésta aplicación. Por definición, la función delta de Dirac, denotada por $\delta(t)$, tiene una altura infinita y un ancho igual a cero, pero su área, o peso es igual a la unidad. Para ilustrar como podemos aproximar una función delta, considere la función rectangular $r_\Sigma(t)$, cuyo ancho es Σ y su altura $1/\Sigma$, tal como se muestra en la Figura 3.5. Podemos entonces definir $\delta(t)$ como

$$\delta(t) = \lim_{\Sigma \rightarrow 0} r_\Sigma(t) \quad (3.5)$$

Las siguientes tres propiedades de la función delta son utilizables

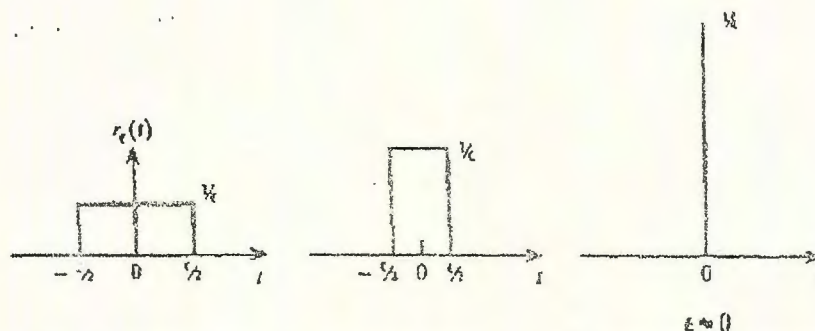


Figura 3.5. Aproximación gradual hasta obtener una función delta de Dirac.[4]

para nuestro caso.

Propiedad 1. Producto de una función y una función delta. En la multiplicación de cualquier función continua en el tiempo $f(t)$, el peso de la función delta se vuelve igual al valor de la función al instante de ocurrir la función delta, o

$$f(t)\delta(t-t_0) = f(t_0)\delta(t-t_0) \quad (3.6)$$

Propiedad 2. Integral de un producto (propiedad de examinación). La integral del producto de cualquier función $f(t)$ y una función delta es una constante, igual al valor de la función al instante de ocurrir la función delta. Para mostrar esto, integraremos el producto

$$\int_{-\infty}^{\infty} f(t)\delta(t-t_0) dt = \int_{-\infty}^{\infty} f(t_0)\delta(t-t_0) dt \quad (3.7)$$

Por la propiedad 1, y ya que $f(t_0)$ no es una función de t , podemos tomarlo fuera de la integral para dar como resultado

$$f(t_0) \int_{-\infty}^{\infty} \delta(t-t_0) dt = f(t_0) \quad (3.8)$$

Esta última igualdad obedece a que el área de $\delta(t)$ es unitaria. La función delta es utilizada para seleccionar, o examinar, el valor de la función en $t=t_0$.

Propiedad 3. Circunvolución de una función con una función delta (propiedad del desplazamiento). Cuando la función $f(t)$ es circunvolucionada con $\delta(t-t_0)$, el origen de $f(t)$ es desplazado a $t=t_0$, o

$$\delta(t-t_0) * f(t) = f(t-t_0) \quad (3.9)$$

Para mostrar esto, escribiremos la ecuación de la circunvolución de tiempo continuo,

$$\int_{-\infty}^{\infty} \delta(\lambda - t_0) f(t - \lambda) d\lambda \quad (3.10)$$

y definiremos la variable $\alpha = \lambda - t$. Entonces tendremos

$$\int_{-\infty}^{\infty} f(-\alpha) \delta(\alpha + t - t_0) d\alpha = f(t - t_0) \quad (3.11)$$

Este último resultado obedece a la propiedad 2.

Un tren infinito de funciones delta ocurriendo cada T_s segundos está denotado por $\delta_{T_s}(t)$ y puede expresarse como

$$\delta_{T_s}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad (3.12)$$

Ahora podemos utilizar estas propiedades de la función delta para determinar la relación entre los espectros. El proceso de muestreo es modelado por la operación de multiplicación dada por

$$C_s(t) = C_A(t) * \delta_{T_s}(t) \quad \text{para todo } t$$

$$C_s(t) = C_A(t) * \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad \text{para todo } t$$

como lo muestra la Figura 3.6. Ya que $C_s(t)$ es una función de tiempo continuo, su espectro $C_s(j\Omega)$ está dado por

$$C_s(j\Omega) = \int_{-\infty}^{\infty} C_s(t) e^{-j\Omega t} dt \quad (3.13)$$

$$C_s(j\Omega) = \int_{-\infty}^{\infty} [C_A(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s)] e^{-j\Omega t} dt \quad (3.14)$$

Cambiando el orden de integración y sumatoria, y aplicando la propiedad de examinación de la función delta, tenemos

$$\begin{aligned} C_s(j\Omega) &= \sum_{n=-\infty}^{\infty} \left[\int_{-\infty}^{\infty} \delta(t - nT_s) C_A(t) e^{-j\Omega t} dt \right] \\ &= \sum_{n=-\infty}^{\infty} C_A(nT_s) e^{-jn\Omega T_s} \end{aligned} \quad (3.15)$$

Hemos alcanzado entonces nuestro primer resultado intermedio.

Comparando la ecuación (3.15) con (3.3), observamos

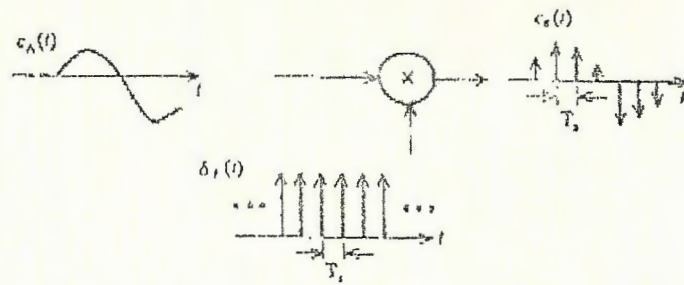


Figura 3.6. El muestreo de una señal de tiempo continuo puede ser tratado como un proceso multiplicativo con un tren de impulsos de Dirac.[3]

$$C(e^{j\omega}) = C_S(j\Omega) \Big|_{\Omega=\omega/T_s} \quad (3.16)$$

Esto es, que la forma de $C(e^{j\omega})$ puede obtenerse determinando $C_S(j\Omega)$ y reemplazando Ω por ω/T_s . La división de ω por T_s escala los valores de ω de manera que sean iguales que Ω . Para aplicaciones de datos muestreados, es convención expresar ω en el rango de $[-\pi/T_s, \pi/T_s)$, para tener el mismo valor numérico que Ω . Las unidades de π/T_s son radianes por segundo, al igual que para la frecuencia de tiempo continuo.

Para completar la comprobación, debemos determinar $C_S(j\Omega)$ en términos del espectro de tiempo continuo $C_A(j\Omega)$. Ya que $\delta_{T_s}(t)$ es una función periódica, puede ser expresada en la siguiente expansión de la serie de Fourier dada por

$$\delta_{T_s}(t) = \sum_{n=-\infty}^{\infty} a_n e^{jn\Omega_n t} \quad (3.17)$$

donde $\Omega_n = 2\pi n/T_s$, y

$$a_n = \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} \delta_{T_s}(t) e^{jn\Omega_n t} dt \quad (3.18)$$

Sobre el intervalo $-T_s/2 \leq t < T_s/2$, el tren de impulsos delta $T_s(t)$ es representado únicamente por el impulso sencillo situado en $t=0$, o $\delta(t)$. Realizando esta sustitución y aplicando la propiedad de examinación, obtenemos

$$a_n = \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} \delta(t) e^{j\Omega_n t} dt = \frac{1}{T_s} e^{j\Omega_n 0} = \frac{1}{T_s} \quad (3.19)$$

para todo n .

Substituyendo esta expansión de la serie de Fourier en el proceso de muestreo, tendremos

$$\begin{aligned} C_s(t) &= C_A(t) * \delta_{T_s}(t) \\ &= C_A(t) * \frac{1}{T_s} \sum_{n=-\infty}^{\infty} e^{-j\Omega_n t} \\ &= \frac{1}{T_s} \sum_{n=-\infty}^{\infty} C_A(t) e^{-j\Omega_n t} \end{aligned} \quad (3.20)$$

El espectro de tiempo continuo de $C_s(t)$ será entonces igual a

$$\begin{aligned} C_s(j\Omega) &= \int_{-\infty}^{\infty} C_s(t) e^{-j\Omega t} dt \\ &= \frac{1}{T_s} \int_{-\infty}^{\infty} \left[\sum_{n=-\infty}^{\infty} C_A(t) e^{-j\Omega_n t} \right] e^{-j\Omega t} dt \end{aligned} \quad (3.21)$$

Intercambiando el orden de sumatoria e integración, obtenemos

$$C_s(j\Omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \left[\int_{-\infty}^{\infty} C_A(t) e^{-j(\Omega + \Omega_n) t} dt \right] \quad (3.22)$$

Examinando el k -ésimo término de la sumatoria, denotado por $C_{s,k}(j\Omega)$, encontramos que

$$\begin{aligned} C_{s,k}(j\Omega) &= \int_{-\infty}^{\infty} C_A(t) e^{-j(\Omega + \Omega_k) t} dt \\ &= C_A(j\Omega + j\Omega_k) \end{aligned} \quad (3.23)$$

Este resultado es simplemente el espectro analógico original que ha sido desplazado de $\Omega=0$ a $\Omega=-\Omega_k$, tal como lo muestra la Figura 3.7.

El espectro total es entonces la sumatoria de estas versiones desplazadas y está dado por

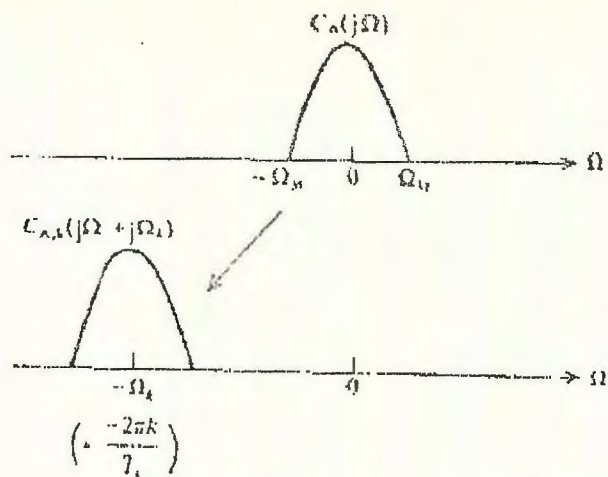


Figura 3.7. Versión desplazada del espectro analógico.[3]

$$C_S(j\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} C_{S,k}(j\Omega)$$

$$C_S(j\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} C_A(j\Omega + j2\pi k/T_s) \quad (3.24)$$

$C_S(j\Omega)$ es una función periódica de Ω , con período $2\pi/T_s$. De forma más concisa, $C_S(j\Omega)$ es llamada la *extensión periódica* de $C_A(j\Omega)$.

Esta segunda identidad es el segundo resultado intermedio deseado. Combinando éste con el resultado dado por (3.16), arribamos a la identidad final deseada dada en (3.4). [3]

3.1.2. Teorema del muestreo.

Cuando se muestrea una señal de tiempo continuo $C_A(t)$ para producir la secuencia $C_A(nT_s)$, deseamos asegurarnos que toda la información de la señal original sea retenida en las muestras. No existirán pérdidas de información si podemos recuperar exactamente la señal de tiempo continuo a partir de las muestras. Para determinar la condición bajo la cual no existirá pérdida de información, vamos a considerar que $C_A(t)$ posee un espectro limitado en banda, o uno para el cual

$$C_A(j\Omega) = 0 \quad \text{para } |\Omega| > \Omega_M \quad (3.25)$$

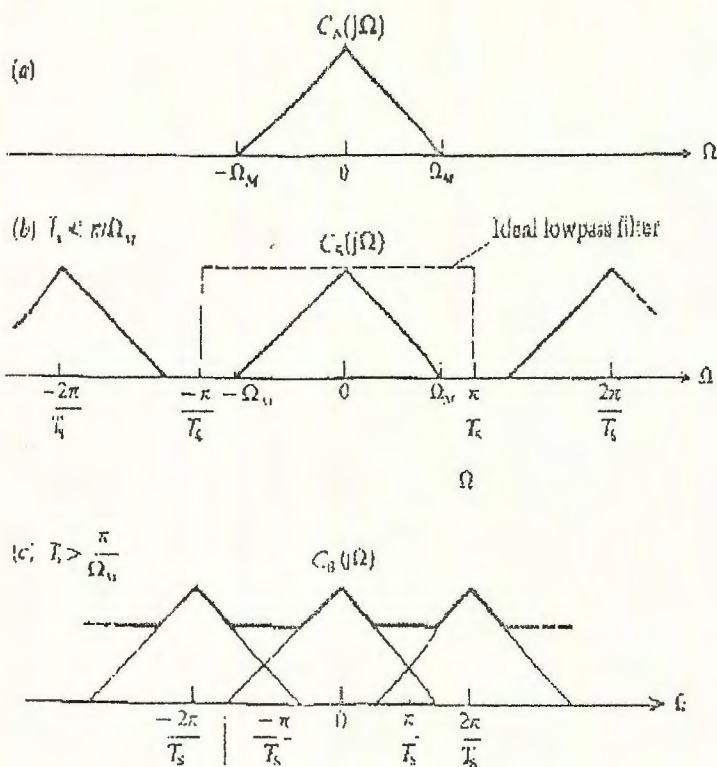


Figura 3.8. (a) Espectro de la señal original; (b) Espectro de la secuencia muestreada cuando $\Omega_M < \pi/T_s$ y (c) cuando $\Omega_M > \pi/T_s$. [3]

tal como se muestra en la Figura 3.8(a). Como se demostró anteriormente, cuando $C_A(t)$ es muestreada con un período de muestreo T_s , entonces el espectro de la señal muestreada $C_S(j\Omega)$ es la extensión periódica de $C_A(j\Omega)$ con período $2\pi/T_s$, como lo muestra la Figura 3.8(b). La forma de $C_S(j\Omega)$ en el rango de frecuencia $[-\pi/T_s, \pi/T_s]$ es idéntico a $C_A(j\Omega)$ si

$$\pi/T_s > \Omega_M \quad \text{o} \quad T_s < \pi/\Omega_M \quad (3.26)$$

En este caso, no hay traslape entre las componentes espectrales adyacentes de tal forma que podemos extraer $C_A(t)$ de forma exacta de $C_S(j\Omega)$ empleando un filtro pasabajo ideal como el mostrado en la Figura 3.8(b).

Si se escoge un T_s mayor que π/Ω_M , ocurrirá un traslape espectral en

la extensión periódica, dada por la ecuación (3.2.24), y la forma de $C_S(j\Omega)$ en el rango $[-\pi/T_s \leq \Omega < \pi/T_s]$ deja de ser similar a $C_A(j\Omega)$ como se muestra en la Figura 3.8(c). Este traslape, causado por muestrear a una razón demasiado baja, produce un error irrecuperable en los valores espectrales, llamado *aliasing*. La verdadera forma espectral es irrecuperable ya que muchas funciones $C_A(j\Omega)$ diferentes pueden producir la misma función $C_S(j\Omega)$. Dos posibilidades se muestran en la Figura 3.9.

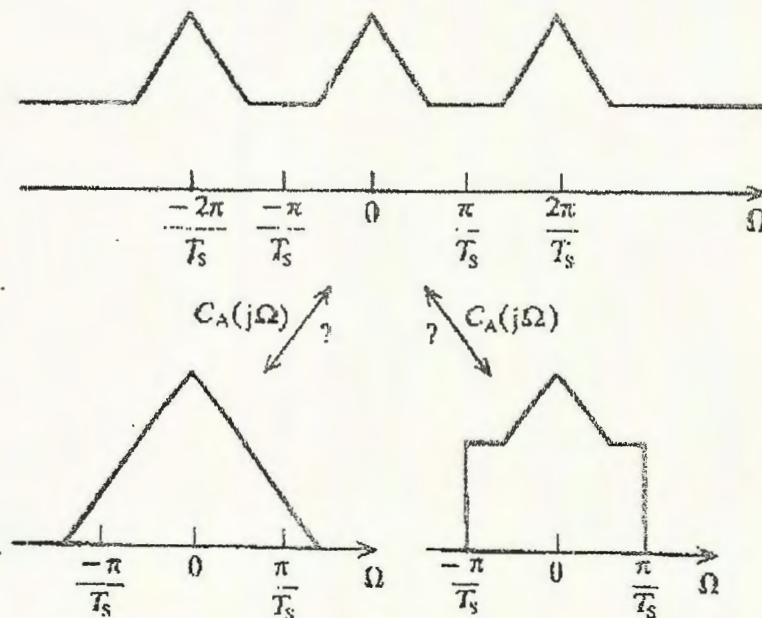


Figura 3.9. Dos posibilidades para el espectro de tiempo continuo cuando ocurre un aliasing.[3]

Una medida muy útil del muestreo es la *razón de muestreo*, $f_s = 1/T_s$, recalcando que $\Omega = 2\pi f$, y definiendo f_M como la componente de frecuencia más alta en la señal, entonces no ocurrirá un traslape espectral si

$$f_s > 2f_M \quad (3.27)$$

Para prevenir errores de aliasing, se requieren al menos dos muestras por período de la componente senoidal de frecuencia más alta presente

en la señal. La razón de muestreo más pequeña antes que ocurra un aliasing para una señal de tiempo continuo particular es llamada *razón de Nyquist*.

3.2. Transformada discreta de Fourier (TDF).

Para secuencias de duración finita, es posible representar su espectro en frecuencia mediante la secuencia de la transformada discreta de Fourier (TDF). "La TDF es en sí misma una secuencia más bien que una función de una variable continua, y que corresponde a muestras, igualmente espaciadas en frecuencia, de la transformada de Fourier de la señal." [6] La TDF es importante por dos razones. Primero, nos permite determinar el contenido de frecuencia de una señal, en otras palabras, realizar un *análisis espectral*. La segunda aplicación es la de realizar un *filtrado en el dominio de la frecuencia*. La TDF es directamente aplicable a secuencias de duración finita. Sin embargo, presenta una serie de inconvenientes al aplicarla en secuencias no finitas.

El principio de funcionamiento del equipo se fundamenta principalmente con este procedimiento, ya que nos permite analizar una secuencia de muestras obtenidas de una señal analógica de entrada -siempre que la señal de entrada cumpla con los requisitos de los rangos de medición- y obtener directamente un muestreo de su espectro de frecuencia.

El análisis de la derivación e interpretación de la TDF puede realizarse desde varios puntos de vista; Para la presentación de la TDF, se prefirió basar las definiciones a partir de la relación entre

las secuencias periódicas y las secuencias de duración finita.

3.2.1. Definición.

La TDF de una función es una secuencia de frecuencias discretas de duración finita que corresponden a un muestreo discreto y finito en frecuencia de la transformada de Fourier de la función. Este muestreo es convencionalmente realizado a N puntos equidistantes sobre la extensión del período en $0 \leq \omega \leq 2\pi$, o sea

$$\omega_k = 2\pi k/N \quad \text{Para } 0 \leq k \leq N-1 \quad (3.28)$$

Si $h(n)$ es una secuencia de tiempo discreto con la transformada de Fourier $H(e^{j\omega})$, entonces la TDF, denotada por $H(k)$, es definida como

$$H(k) = H(e^{j\omega}) \Big|_{\omega=\omega_k=2\pi k/N} \quad (3.29)$$

La secuencia TDF comienza en $k=0$, correspondiendo al valor de $\omega=0$, pero no incluye $k=N$, correspondiente a $\omega=2\pi$.

Como se muestra en la Figura 3.10, cuando $H(e^{j\omega})$ es muestreada con período $\omega_s=2\pi/N$, la secuencia de tiempo discreto correspondiente, denotada por $h(n)$, se vuelve periódica en el tiempo con período $2\pi/\omega_s=N$. Esta secuencia de tiempo discreto periódica puede ser expresada en términos de $h(n)$ como:

$$h'(n) = \sum_{m=-\infty}^{\infty} h(n + mN) \quad (3.30)$$

La secuencia $h'(n)$ es llamada la *extensión periódica* de $h(n)$. El número de los puntos de muestreo en un período del espectro N es

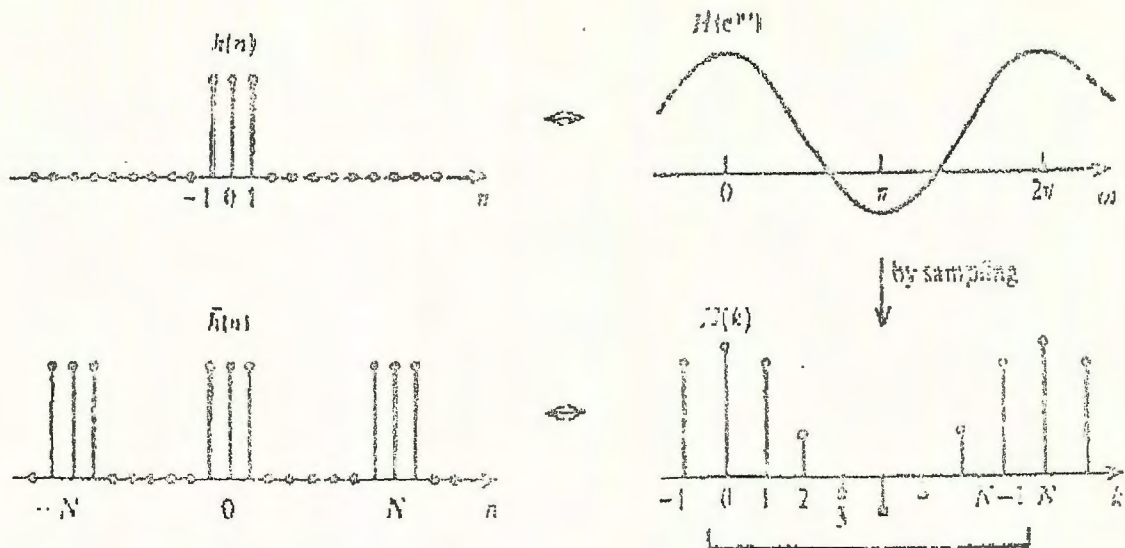


Figura 3.10. Caso para $N=8$ muestras de un espectro para $0 \leq \omega \leq 2\pi$. [3]

también igual al período de $h'(n)$.

El valor de N puede escogerse libremente, no obstante hay que tener presente que N determinará también el período de $h'(n)$ y por lo tanto hay que tener cuidado en no escoger un N demasiado pequeño. Las consideraciones involucradas en la escogitación del valor de N se ilustran a continuación. Sea $h(n)$ una secuencia de duración finita cualquiera conteniendo M muestras. Como lo muestra la Figura 3.11, si $M \leq N$, entonces

$$h'(n) = h(n) \quad \text{para } 0 \leq n \leq M-1 \quad (3.31a)$$

$$h'(n) = 0 \quad \text{para } M \leq n \leq N-1 \quad (3.31b)$$

Dos puntos resultan notorios de éste resultado cuando $M \leq N$. El primero es que la secuencia de duración finita $h(n)$, para $0 \leq n \leq M-1$, puede recuperarse únicamente del primer período de $h'(n)$.

El segundo punto es que el exceso del número de puntos en un período de $h'(n)$, o aquellos para $kM \leq n \leq k(N-1)$ para cualquier entero k , son iguales a cero.

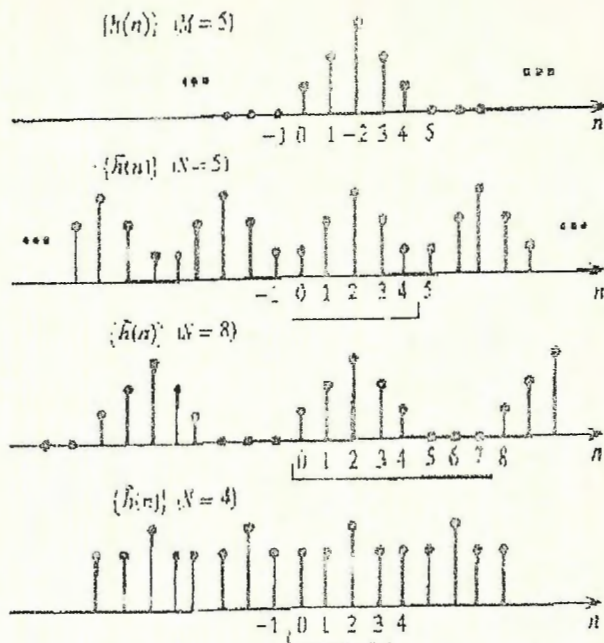


Figura 3.11. Relación de la duración de la secuencia M y el número de puntos de muestreo del espectro N . Cuando $N < M$ ocurre un efecto de aliasing.[3]

Si $M > N$, no obstante, la duración de $h(n)$ es mayor que el período de $h^*(n)$. Un traslape ocurre entonces al formarse la extensión periódica, como se aprecia en la Figura 3.3.2. Este traslape ocasiona entonces un error al tratar de recuperar $h(n)$ del primer período de $h^*(n)$.

Para prevenir el efecto del aliasing, se debe muestrear $H(e^{j\omega})$ para $0 \leq \omega \leq 2\pi$ al menos tantas veces como existan elementos en $h(n)$, o $N \geq M$ veces.

3.2.2. Cálculo de la TDF.

Ahora derivaremos la relación de la TDF con la secuencia de tiempo discreto. Primero, ya que $h^*(n)$ es una secuencia periódica, puede ser expresada como la siguiente expansión de la serie de Fourier

$$h^*(n) = \frac{1}{N} \sum_{k=-\infty}^{\infty} H'(k) e^{j2\pi kn/N} \quad (3.32)$$

donde $H'(k)$ es un conjunto de coeficientes complejos a ser determinados, y el factor $1/N$ ha sido aceptado como la convención. Debido a que la secuencia exponencial compleja de tiempo discreto $e^{j2\pi kn/N}$ es en sí periódica en k , con período N , existe únicamente un número finito de secuencias distintas, igual a $e^{j2\pi kn/N}$, para $0 \leq k \leq N-1$. Para mostrar ésto, hagamos $0 \leq k \leq N-1$ y consideremos una frecuencia fuera de éste rango, $e^{j2\pi(k+N)n/N}$ por ejemplo. Entonces

$$e^{j2\pi(k+N)n/N} = e^{j2\pi kn/N} e^{j2\pi n} = e^{j2\pi kn/N} \quad (3.33)$$

la cual es idéntica a la secuencia $e^{j2\pi kn/N}$. Así, se requiere incluir en la suma de arriba únicamente un número finito de frecuencias, y $h'(n)$ puede expresarse en términos de las siguientes N componentes de frecuencia

$$h'(n) = \frac{1}{N} \sum_{k=0}^{N-1} H'(k) e^{j2\pi kn/N} \quad (3.34)$$

Para determinar los valores de $H'(k)$, para $0 \leq k \leq N-1$, multiplicamos ambos lados de la ec. (3.34) por $e^{-j2\pi knr/N}$ y sumamos para $0 \leq n \leq N-1$, obteniendo

$$\sum_{n=0}^{N-1} h'(n) e^{-j2\pi knr/N} = \frac{1}{N} \sum_{n=0}^{N-1} \left[\sum_{k=0}^{N-1} H'(k) e^{j2\pi kn/N} \right] e^{-j2\pi knr/N} \quad (3.35)$$

Cambiando el orden de las sumatorias y notando que $H'(k)$ no es una función de n , obtenemos

$$\sum_{n=0}^{N-1} h'(n) e^{-j2\pi knr/N} = \frac{1}{N} \sum_{k=0}^{N-1} H'(k) \left[\sum_{n=0}^{N-1} e^{j2\pi kn(k-r)/N} \right] \quad (3.36)$$

La sumatoria dentro de corchetes puede evaluarse aplicando la fórmula de sumatoria geométrica finita, para obtener

$$\sum_{n=0}^{N-1} e^{j2\pi kn(k-r)/N} = \frac{1 - e^{j2\pi N(k-r)/N}}{1 - e^{j2\pi(k-r)/N}} \quad (3.37)$$

Ya que $(k-r)$ solo toma valores enteros, el término del numerador es cero para todos los valores de k y r . No obstante, para $k=r+mN$, donde

m es un entero, el denominador también es cero. En este caso, cada término en la suma es igual a 1, y la suma es N, o

$$\begin{aligned} \sum_{n=0}^{N-1} e^{j2\pi n(k-r)/N} &= N \text{ para } k=r+mN \\ \sum_{n=0}^{N-1} e^{j2\pi n(k-r)/N} &= 0 \text{ otros valores} \end{aligned} \quad (3.38)$$

Ya que k está dentro del rango de cero a N-1 en la ec. (3.36), podemos aplicar este resultado para k=r, o para m=0. Esto resulta en la solución deseada para la $H^*(r)$ dada por

$$H^*(r) = \sum_{n=0}^{N-1} h^*(n) e^{-j2\pi nr/N} \text{ para } 0 \leq r \leq N-1 \quad (3.39)$$

Sustituyendo r con k da la solución final deseada para $H^*(k)$.

Ahora relacionamos este resultado con la transformada de Fourier de la secuencia original $h(n)$. Sea $h(n)$ una secuencia de duración finita conteniendo M muestras. Entonces su transformada de Fourier vendrá dada por

$$H(e^{j\omega}) = \sum_{n=0}^{M-1} h(n) e^{-j\omega n} \quad (3.40)$$

La duración de $h(n)$ puede incrementarse de M a N muestras agregando el número apropiado de ceros. Ya que los elementos cero no contribuyen a la sumatoria, la transformada de Fourier es igual a

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} \quad (3.41)$$

El muestreo de $H(e^{j\omega})$ en N puntos equidistantes en $0 \leq \omega \leq 2\pi$ produce la secuencia

$$H(k) = H(e^{j\omega}) \Big|_{\omega=2\pi k/N} = \sum_{n=0}^{N-1} h(n) e^{-j2\pi kn/N} \quad 0 \leq k \leq N-1 \quad (3.42)$$

Ya que el número de puntos muestreados N es igual al número de elementos en la secuencia de tiempo discreto original, las primeras N muestras de $h^*(n)$ son iguales a $h(n)$, o $h(n)=h^*(n)$, para $0 \leq n \leq N-1$. Solamente los primeros N valores de $h^*(n)$ son necesarios en la sumatoria de la ec. (3.39). Comparando (3.42) con (3.39), notamos que $H(k)=H^*(k)$, para $0 \leq k \leq N-1$. Aplicando este resultado a (3.34),

obtenemos

$$h'(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k) e^{j2\pi nk/N} \quad \text{para } 0 \leq n \leq N-1 \quad (3.43)$$

Resumiendo estos resultados, arribamos a las relaciones deseadas de la TDF:

$$H(k) = \sum_{n=0}^{N-1} h(n) e^{-j2\pi nk/N} \quad \text{para } 0 \leq k \leq N-1 \quad (3.44)$$

la cual es llamada una TDF de N puntos.

3.3. La transformada rápida de Fourier (TRF).

Para facilitar la representación y el cálculo matemático involucrado en el procesamiento de señales, muy a menudo se emplea aritmética compleja para representar señales. Los números complejos toman la forma $A + jB$, donde A es llamada la componente "real", B es la componente "imaginaria", y j es igual a la raíz cuadrada de -1 . La forma compleja nos permite el manejo de intercambios entre las representaciones senoidal y exponencial debido a que

$$e^{-jt} = \cos(t) - j\text{sen}(t) \quad (3.45)$$

Cuando un circuito es estimulado por una señal senoidal, los números complejos facilitan la representación y el cálculo. Esto es porque un número complejo puede representar no solo la magnitud de una onda senoidal, sino además su fase - siendo ambos parámetros de gran importancia para caracterizar la onda. La transformada discreta de Fourier (TDF) convierte las muestras de una señal continua en el tiempo a muestras de las diferentes frecuencias y amplitudes de los armónicos que componen el espectro de frecuencia de dicha señal.

Para realizar la conversión, el cálculo de la TDF requiere n^2 multiplicaciones y sumas complejas; Donde n es el número de muestras

y usualmente es una potencia de 2. La *transformada rápida de Fourier (TRF)* es una implementación elegante de la TDF, la cuál reduce el número de multiplicaciones y sumas complejas requeridas para la conversión a $n\text{Log}_2(n)$.

Eso implica un tamaño 100 veces inferior para una TRF de 1024 puntos. El algoritmo de la TRF tiene además la ventaja de no requerir almacenamiento de datos más allá de las muestras de datos originales.

3.3.1 Definición.

El algoritmo de la TRF explota las propiedades de simetría de la exponencial compleja de tiempo discreto para reducir significativamente el número de multiplicaciones para el cálculo de la TDF. Para evaluar una TDF de N puntos, el algoritmo de la TRF utilizado acá alcanza su eficiencia máxima cuando n es una potencia de 2, o $N=2^{N\text{Log}_2}$, donde $N\text{Log}_2$ es siempre entero. Esta restricción no presenta problemas prácticos, ya que la longitud de la secuencia de muestras originales $h(n)$ puede siempre incrementarse hasta la próxima potencia de 2 rellenandola con un número apropiado de ceros.

Para lograr un mejor entendimiento de los pasos del algoritmo de la TRF, vamos a considerar la secuencia de entrada $h(n)$, para $0 \leq n \leq N-1$, como una secuencia de valores reales. Las mismas ideas básicas son aplicables para el caso de una secuencia de valores complejos.

Para secuencias de tiempo discreto de valores reales, la definición de la TDF se reduce a

$$H(k) = \sum_{n=0}^{N-1} h(n) \cos(2\pi nk/N) - j \sum_{n=0}^{N-1} h(n) \text{sen}(2\pi nk/N) \text{ para } 0 \leq k \leq N-1 \quad (3.46)$$

Para calcular ésta secuencia de N puntos la TDF requiere de $2N^2$

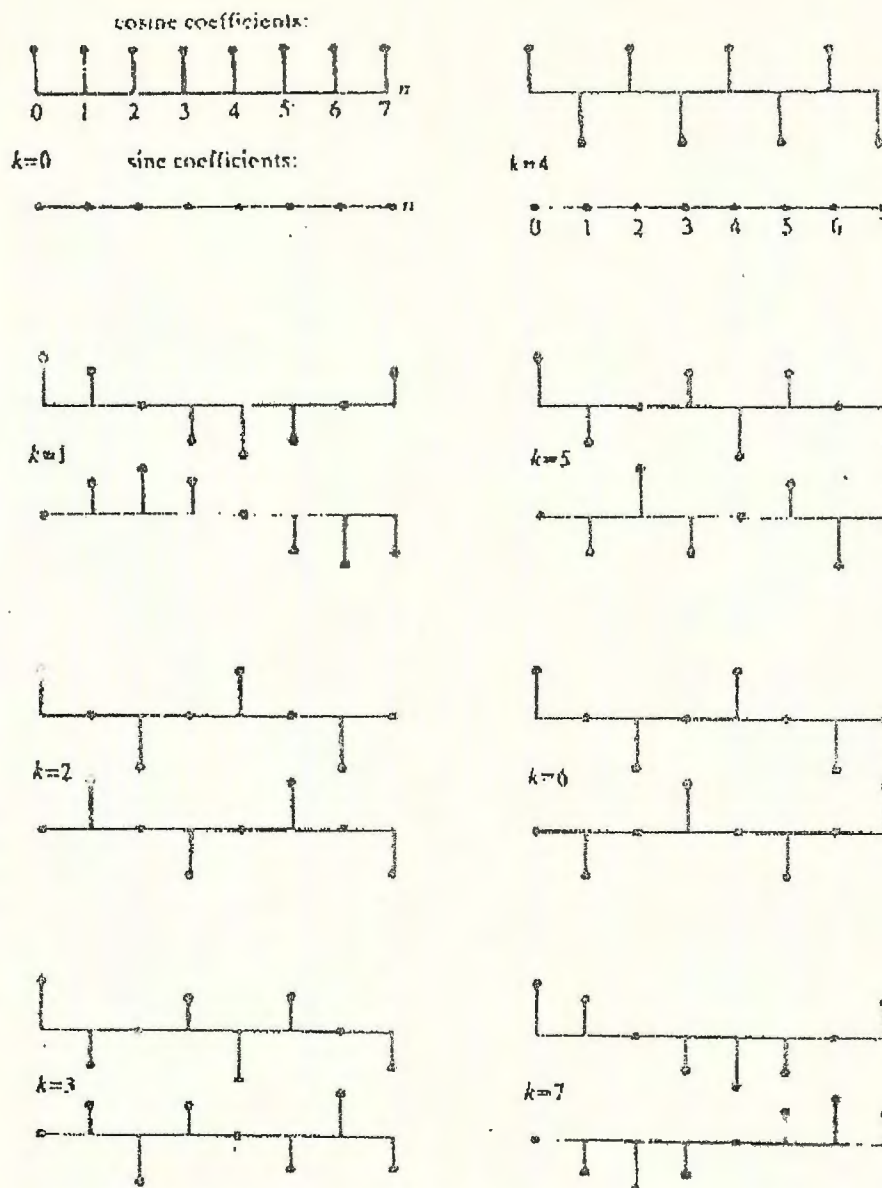


Figura 3.12. Juego de valores de coeficientes seno y coseno necesarios para evaluar el k -ésimo elemento de una secuencia de 8 puntos.

multiplicaciones. El juego de valores para los coeficientes seno y coseno requeridos para evaluar cada término $H(k)$ se muestran en la Figura 3.12 para el caso en que $N=8$. Los coeficientes para calcular los términos k y $(N/2)+k$, para $0 \leq k \leq (N/2)-1$, exhiben una curiosa similitud. Los coeficientes para los n pares son los mismos, mientras aquellos con índices impares tienen la misma magnitud, pero son de signo opuesto. Para explotar ésta similitud, descomponemos la

secuencia original $h(n)$ en dos subsecuencias: $h_E(n)$ la cual contendrá aquellos elementos cuyos índices son pares y $h_O(n)$ que contendrá a los de índices impares. Estos juegos de coeficientes se muestran en la Figura 3.13.

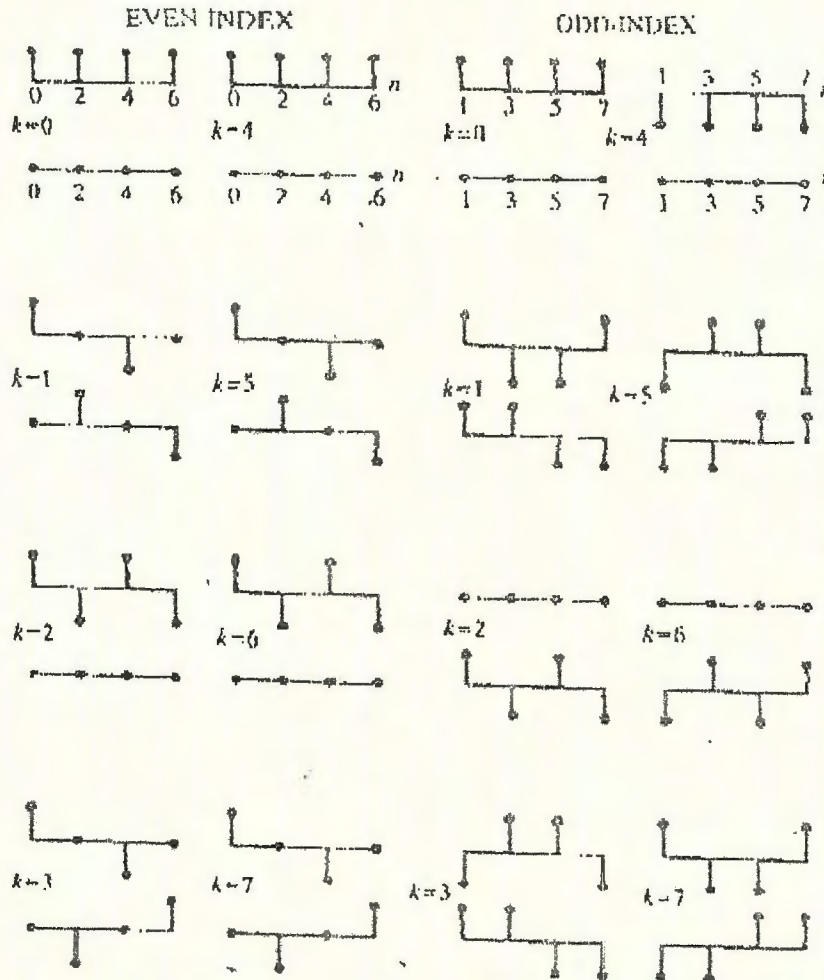


Figura 3.13. Secuencia original de 8 puntos decimada en subsecuencias de 4 puntos indexadas en valores pares o impares de k .

Dos características de los coeficientes pares son notorias. Primero, el juego de coeficientes para los puntos de frecuencia k y $(N/2)+k$, para $0 \leq k \leq (N/2)-1$ son idénticos. Ya que su contribución a los términos $H(k)$ y $H(N/2+k)$ será la misma, únicamente necesitamos calcular uno de ambos términos obteniendo el segundo al mismo tiempo. La segunda característica es que éste juego de caracteres es exactamente el

requerido para calcular la TDF de $h_E(n)$ con $N/2$ puntos. Para calcular ésta TDF de $N/2$ puntos es necesario realizar $2(N/2)^2 = N^2/2$ multiplicaciones reales. Un resultado similar exceptuando por una inversión de signos en los coeficientes ocurre para la secuencia de índices impares. Para calcular las transformadas de $N/2$ puntos de ambas secuencias (pares e impares) requeriríamos N^2 multiplicaciones, lo cual es mucho menor que las $2N^2$ multiplicaciones originales. De esta forma, dividiendo, o *decimando*, la secuencia original en subsecuencias pares e impares nos lleva a una reducción en el número de cálculos a evaluar para obtener la TDF.

Para reducir aún más el número de multiplicaciones, consideramos ahora a $h_E(n)$ como la secuencia original a ser transformada en una TDF de $N/2$ puntos. Decimamos nuevamente $h_E(n)$ en sub-secuencias pares e impares. El juego de coeficientes correspondientes se muestra en la Fig 3.14. Nuevamente pueden notarse las mismas características en los coeficientes de las nuevas subsecuencias: el conjunto de caracteres para los puntos de frecuencia k y $(N/2)+k$, para $0 \leq k \leq (N/4)-1$ son idénticos y representan los valores de coeficientes necesarios para calcular una DFT de $N/4$ puntos. En ésta etapa de la decimación, se obtienen 4 subsecuencias de $N/4$ puntos cada una requiriendo $2(N/4)^2 = N^2/8$ multiplicaciones para su cálculo, inferior al $N^2/2$ de requerido para $h_E(n)$ y mucho menor que el $2N^2$ de la secuencia original. Por lo tanto se logra una nueva reducción del número de multiplicaciones necesarias para la obtención de la TDF.

La decimación puede continuar hasta que únicamente queden parejas de la secuencia original. Cada una de éstas requerirá el cálculo de una TDF de 2 puntos. A éste nivel, sin embargo, todos los coeficientes son ya sea 0 o ± 1 , y no requieren una multiplicación.

Esto puede observarse de la ecuación (3.46) haciendo $N=2$ y fijando $k=0$ o 1. Entonces, simplemente sumando y restando elementos al nivel más bajo posible de decimación, es posible calcular TDF's de 2

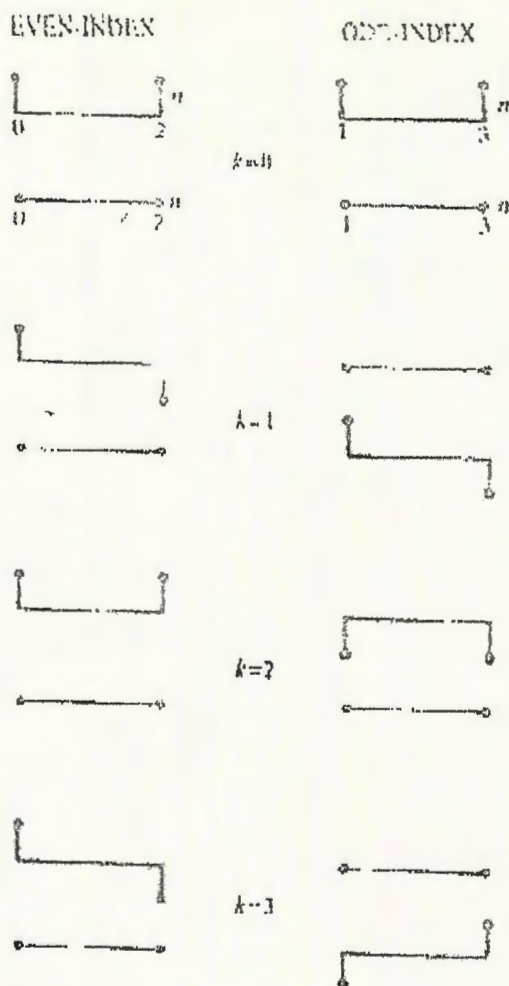


Figura 3.14. Subsecuencia par de 4 puntos decimada en subsecuencias de 2 puntos indexadas en valores pares e impares de k . [3]

puntos. Este procedimiento de decimación que se ha seguido es llamado *algoritmo de TRF por decimación en tiempo (DET)*. Existe también un *algoritmo de TRF por decimación en frecuencia (DEF)*. Ambos algoritmos difieren en el orden en que se realizan los agrupamientos de coeficientes pero ambos llegan a la misma respuesta.

La DET resulta de una división recursiva de las muestras en el dominio del tiempo en grupos pares e impares hasta llegar al dominio del tiempo (Figura 3.15a) mientras que la DEF viene de iniciar la división en grupos pares e impares desde los puntos en el dominio de la frecuencia a la salida de la TDF hasta los puntos en el dominio del tiempo (Figura 3.15b).

Habiendo descrito la operación de decimación e indicado el por qué

existe una reducción neta del número de cálculos, describiremos a continuación el procedimiento de combinar las TDF de $N/2$ puntos para formar una TDF de N puntos y determinar el número de operaciones.

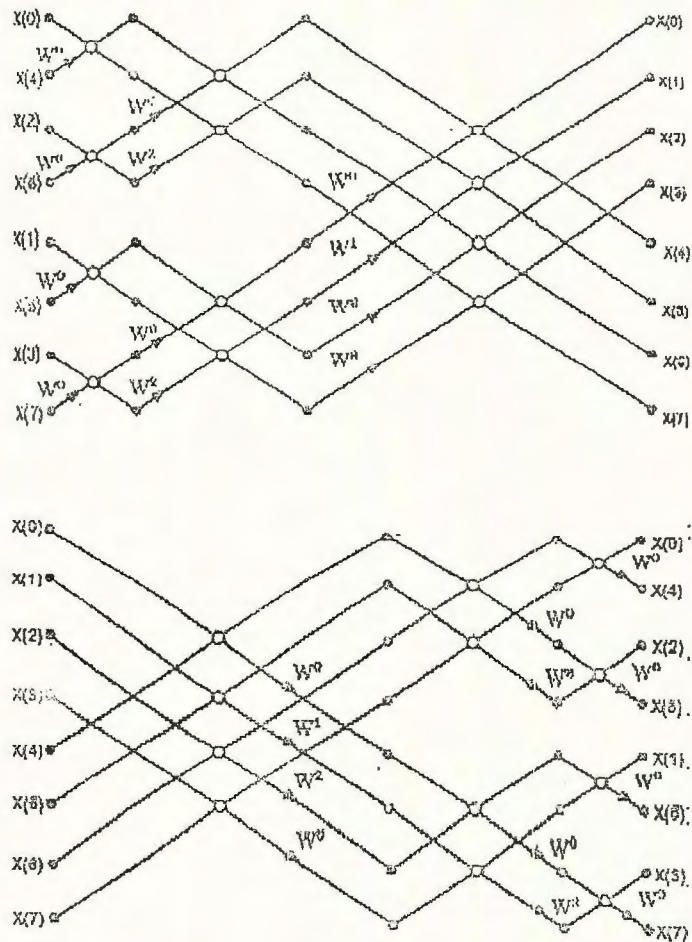


Figura 3.15. Algoritmos de decimación aplicados en la TRF; a) Decimación En Tiempo (DET) y b) Decimación En Frecuencia (DEF).[7]

3.3.2 Derivación analítica del algoritmo de la TRF por decimación en tiempo.

En esta sección se provee el análisis que respalda las intuiciones presentadas en la sección previa. Sea $h(n)$, para $0 \leq n \leq N-1$, donde $N=2^{N \log_2 2}$, donde $N \log_2 2$ es siempre entero y la secuencia compuesta de

valores complejos. Nosotros dividimos la TDF definiendo ecuaciones en dos sumatorias consistentes en las secuencias pares e impares dadas por

$$H(k) = \sum_{n=0}^{N-1} h(n) e^{-j2\pi nk/N} \quad \text{para } 0 \leq k \leq N-1 \quad (3.47)$$

$$= \sum_{n \text{ impar}} h(n) e^{-j2\pi nk/N} + \sum_{n \text{ par}} h(n) e^{-j2\pi nk/N} \quad (3.48)$$

Haciendo $n=2r$ en la primer sumatoria, y $n=2r+1$ en la segunda, obtendremos

$$H(k) = \sum_{r=0}^{N/2-1} h(2r) e^{-j4\pi rk/N} + \sum_{r=0}^{N/2-1} h(2r+1) e^{-j4\pi(2r+1)k/N} \quad (3.49)$$

$$= \sum_{r=0}^{N/2-1} h(2r) e^{-j2\pi rk/(N/2)} + e^{-j2\pi k/N} \sum_{r=0}^{N/2-1} h(2r+1) e^{-j2\pi rk/(N/2)} \quad (3.50)$$

Ambas sumatorias aparecen en el formato de una TDF de $N/2$ puntos. Definiremos $h_E(n)$ como la secuencia de valores de índice par y $h_o(n)$ como la secuencia de valores de índice impar y $H_E(k)$ y $H_o(k)$ serán, respectivamente, las TDF de $h_E(n)$ y de $h_o(n)$. Ya que $H_E(k)$ y $H_o(k)$ son TDF's de secuencias de $N/2$ puntos, ambas son periódicas en k con un período de $N/2$. Se combinarán para formar la TDF de N puntos $H(k)$ por

$$H(k) = H_E(k) + e^{-j2\pi k/N} H_o(k) \quad \text{para } 0 \leq k \leq (N/2)-1$$

$$= H_E(k-N/2) + e^{j2\pi k/N} H_o(k-N/2) \quad \text{par} \quad \text{a } N/2 \leq k \leq N-1 \quad (3.51)$$

Ya que para calcular la TDF de una secuencia compleja de N puntos se requiere realizar $4N^2$ multiplicaciones, para el cálculo de $H_E(k)$ o $H_o(k)$ se requerirán $4(N/2)^2 = N^2$ multiplicaciones, o $2N^2$ para calcular ambas. Para combinar $H_E(k)$ y $H_o(k)$ y formar $H(k)$ se requiere de cuatro multiplicaciones reales por cada punto para $0 \leq k \leq (N/2)-1$, o $2N$ para el procedimiento de combinación. Los puntos para $N/2 \leq k \leq N-1$

contienen los mismos productos pero con diferentes signos. Entonces para calcular una TDF de N puntos decimando una vez la secuencia requiere de $2(N^2+N)$ multiplicaciones, lo cual es menor que $4N^2$ para $N > 2$.

Si al decimar la secuencia una vez se logra una reducción del número de multiplicaciones, decimaciones adicionales conllevarán a reducirlo aún más. Para mostrar esto, las secuencias pares e impares obtenidas pueden considerarse como si fueran las secuencias originales y ser decimadas en sus respectivas subsecuencias pares e impares. La TDF de los elementos numerados pares de la secuencia original pueden ser expresados como

$$\begin{aligned}
 H_E(k) &= H_{E,E}(k) + e^{-j2\pi k/(N/2)} H_{E,O}(k) \quad \text{para } 0 \leq k \leq (N/4) - 1 \\
 &= H_{E,E}(k - N/4) + e^{-j2\pi k/(N/2)} H_{E,O}(k - N/4) \quad \text{para } N/4 \leq k \leq N/2 - 1
 \end{aligned} \tag{3.52}$$

donde
 $H_{E,E}($

$k)$ es la TDF de la subsecuencia par de $h_E(n)$ y $H_{E,O}(k)$ es la correspondiente a la subsecuencia impar de $h_E(n)$. Una ecuación similar puede ser escrita para $H_O(k)$.

Por el hecho que $N = 2^{N \log_2 2}$, la secuencia original puede decimarse en secuencias pares e impares $N \log_2 - 1$ veces, hasta llegar a obtener $2^{N \log_2 - 1}$ subsecuencias de 2 elementos. La primer secuencia de dos puntos de índice par está compuesta por $h(0)$ y $h(N/2)$. Estas secuencias de 2 puntos tienen la TDF de dos puntos dada por

$$H_{E,E}(k) = h(0) + h(N/2) e^{-j\pi k} \quad \text{para } k=0,1 \tag{3.53}$$

Evaluando ésta expresión, encontramos

$$H_{E,E}(0) = h(0) + h(N/2) \quad \text{y} \quad H_{E,E}(1) = h(0) - h(N/2)$$

Esta última operación se muestra gráficamente en la Figura 3.16, y es llamada *patrón de mariposa*. Los pasos para combinar las TDF's de las subsecuencias decimadas son mostrados como una secuencia de mariposas en la Figura 3.17 para un $N=8$.

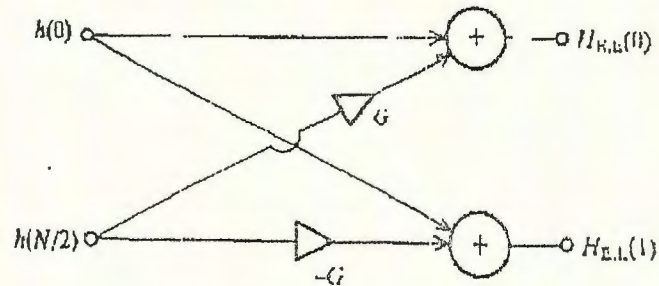


Figura 3.16. Patrón de mariposa ($G=1$).[7]

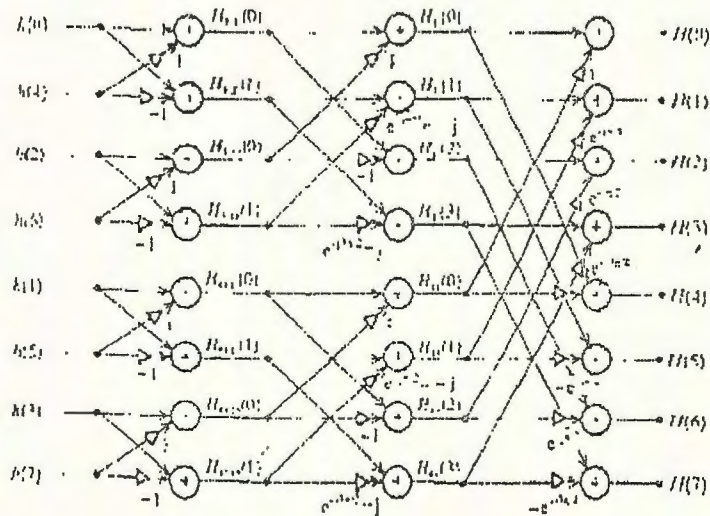


Figura 3.17. Descomposición de una TDF de n puntos en patrones de mariposa. Caso para $N=8$ puntos.[3]

Los cálculos son realizados columna por columna, operando todas las parejas primero y luego pasando a la siguiente columna de parejas y trabajar con los resultados de las parejas de la columna anterior. Debe notarse el hecho que una vez realizado un cálculo, los valores anteriores no son necesarios para futuras operaciones. Por lo tanto, las localidades de memoria que fueron utilizadas para almacenar los resultados intermedios pueden ser liberadas y reutilizadas para

almacenar los nuevos valores y, en última instancia, los valores finales de la TDF. Esta forma de calcular los valores resulta bastante eficiente en cuanto al manejo de la memoria y presenta una segunda ventaja para implementar TRFs con maquinas de poca memoria RAM. Para obtener un número aproximado de las operaciones necesarias, notese que existen N multiplicaciones complejas, o $4N$ multiplicaciones reales, que se requieren para luego de cada decimación y que existen Log_2N-1 divisiones. Por lo tanto, se requieren $4N(\text{Log}_2N-1)$ multiplicaciones reales. Lo importante en el número de multiplicaciones no es en sí el número exacto de ellas, sino su relación con el número de puntos N . La evaluación normal de la TDF requiere de N^2 multiplicaciones complejas, mientras que al evaluar la TDF por medio del algoritmo de la TRF se requieren únicamente $N\text{Log}_2N$. Ambos números se comparan en la Tabla 3.4.1. Nótese que pueden alcanzarse reducciones desde aproximadamente 6 veces (caso de $N=32$) hasta cerca de 100 veces (caso de $N=1024$) en el número de operaciones requeridas. Ya que ésta reducción está relacionada directamente al tiempo de cálculo, la abrumadora popularidad de la TRF es comprensible.

$N\log_2$	N	TDF($4N^2$)	TRF($4N\log_2N$)
5	32	4,096	640
6	64	16,384	1,536
7	128	65,536	3,584
8	256	262,144	8,192
9	512	1,048,576	18,432
10	1024	4,194,304	40,960

Tabla 3.4.1. Comparación del número de multiplicaciones reales requeridas para una TDF de N puntos ($N=2^k$)

CAPITULO IV

DISEÑO DE LA TARJETA ADQUISITORA DE DATOS.

4.1. Estructura general.

Para llevar a cabo la realización de este proyecto, se requirió del diseño de una tarjeta adquisitora de datos mediante la cual poder alimentar a la computadora con los datos necesarios sobre la señal a analizar para su posterior procesamiento.

Por tanto, la tarjeta adquisitora diseñada tiene la capacidad de controlar los siguientes procesos:

- Muestreo de la señal.
- Almacenamiento temporal de la información.
- Transferencia de la información a la computadora.

En base a las actividades anteriormente mencionadas se estructuró la arquitectura de la tarjeta dividiéndose en varias etapas funcionalmente independientes para luego integrarlas y obtener un sistema de adquisición de datos completo.

Así, se decidió estructurar la tarjeta de la siguiente manera:

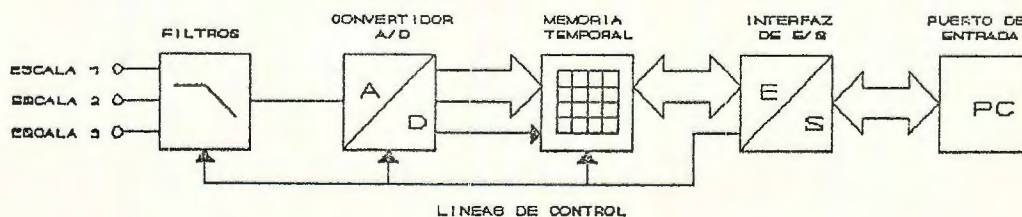


Figura 4.1. Diagrama esquemático de la tarjeta.

Prácticamente, el control de las operaciones de la tarjeta es manejada por la PC a través de la interfaz de entrada/salida mediante comandos.

La lógica de operación es la siguiente:

Primeramente, la tarjeta debe permanecer inhabilitada para evitar cualquier incompatibilidad con la máquina. Una vez que se arranque el programa, se habilita la tarjeta para dar inicio al proceso. Este inicia con el envío de una instrucción a través de la interfaz de entrada/salida hacia la etapa de filtros para seleccionar la escala (el filtro de entrada) con que se va a trabajar. Luego, mediante otra instrucción, se controla la etapa del convertidor y de la memoria temporal para que el primero inicie la conversión mientras el segundo comienza a llenarse con los datos provistos por aquél. Una vez que se han recabado suficientes datos, se desactiva la etapa del convertidor y se ordena a la etapa de memoria temporal que inicie la transferencia de la información almacenada hacia la PC siempre por medio de la interfaz de salida. Una vez introducidos los datos a la PC, ésta comienza a procesarlos y a presentar los resultados en pantalla.

A continuación se detallan cada una de éstas etapas así como su funcionamiento.

4.2. Etapa de Filtros.

La etapa de filtros juega un papel muy importante en la adquisición de los datos debido a que con ella restringimos el espectro de frecuencias de entrada al sistema para dar cumplimiento con los requerimientos del teorema de Nyquist.

El diagrama de esta etapa se muestra en la Figura 4.2.

Debido a las limitaciones en velocidad de los componentes a que se

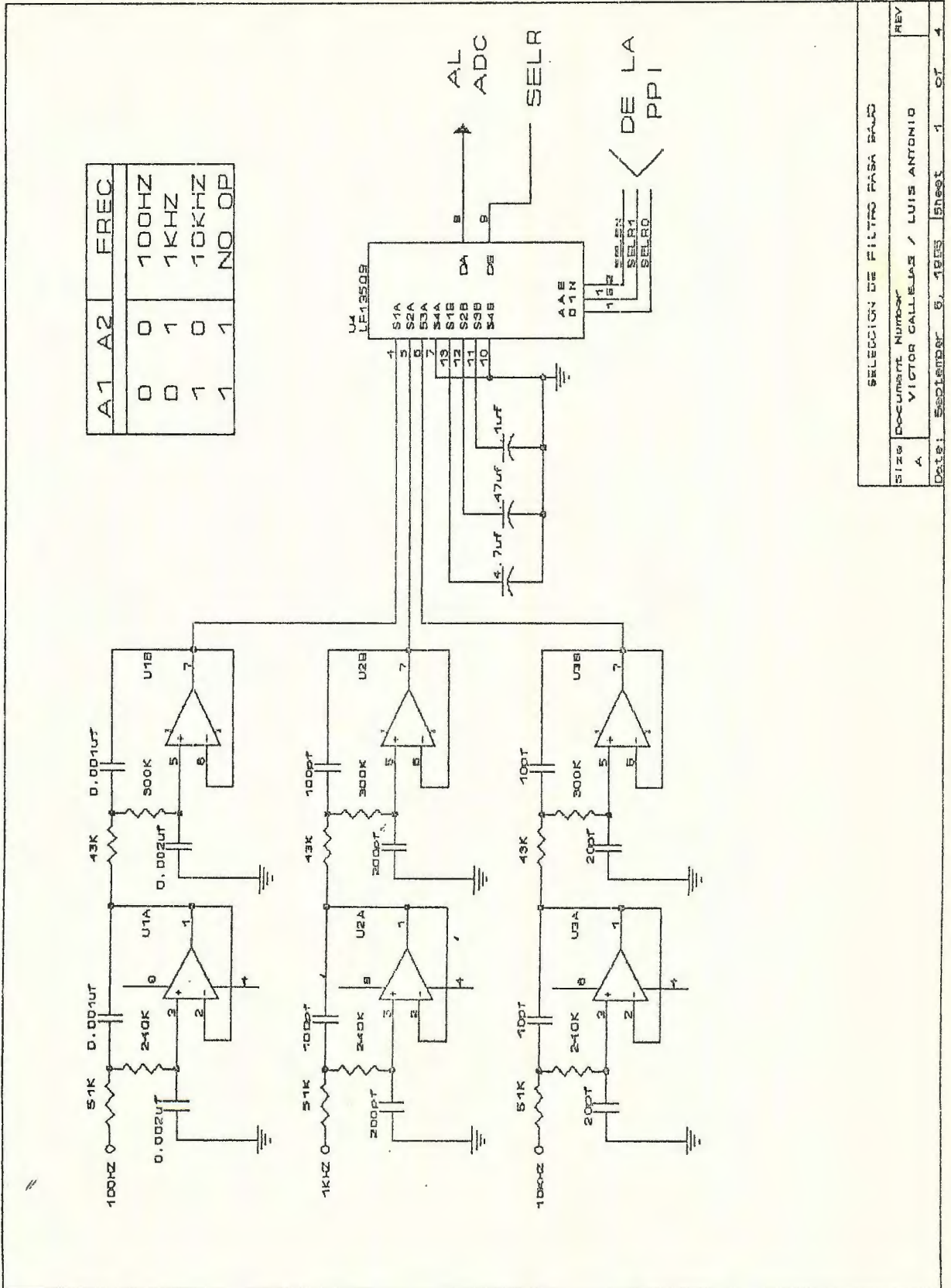


Figura 4.2. Diagrama esquemático de la etapa de filtros.

tuvo acceso (principalmente el ADC), la máxima velocidad de muestreo que se pudo lograr fué de $20\mu\text{S}$ ($f=50\text{KHz}$) por lo que se optó en fijar el tope de la escala de más alta frecuencia en 10KHz . Sin embargo, y como se verá posteriormente, ésta escala no es suficiente para cubrir con precisión todo el espectro de frecuencia desde 10KHz hacia abajo. Es por ello, que se crearon tres escalas más para poder así visualizar con mayor exactitud las componentes de frecuencia localizadas en la parte baja del espectro. Las otras dos escalas se fijaron a 1KHz y a 100Hz con velocidades de muestreo de $100\mu\text{S}$ y 1mS respectivamente. Esta división en escalas para la medición del espectro de frecuencia garantiza una razón de 10 muestras por ciclo de la componente en frecuencia más alta de la señal permisible dentro del rango estipulado por la escala. Esto obedece también al hecho que los algoritmos de la TDF y la TRF empleados en el programa principal se implementaron para 128 muestras. Por lo tanto, se pueden tener 128 puntos para la representación del espectro. Así por ejemplo, en el caso de la escala de 100Hz , puede analizarse una señal cuadrada de 100Hz muestreada con una velocidad de 1mS obteniendo un espectro hasta aproximadamente el 5° armónico. El límite inferior de cada escala está limitado por el tiempo de duración del muestreo; para nuestro caso el tiempo de duración del muestreo es de $128*1\text{mS} = 128\text{mS}$ por lo que la frecuencia mínima que puede analizarse garantizando que se ha muestreado al menos un ciclo completo de la señal es de $1/128\text{mS} = 7.81\text{Hz}$, lograndose obtener hasta el 64° armónico de la señal. Por razones de velocidad de muestreo del convertidor, la escala de 10KHz no conserva el criterio de 10 muestras por ciclo, limitándose a 5 muestras por ciclo ($f_s=50\text{KHz}$).

La tabla 4.1 muestra los valores de los filtros para cada escala así como sus correspondientes rangos de frecuencias y frecuencias de muestreo.

ESCALA	FRECUENCIA DE CORTE	FRECUENCIA MINIMA	FRECUENCIA MAXIMA	FRECUENCIA DE MUESTREO
1	10KHz	781Hz	10KHz	50KHz
2	1KHz	78.1Hz	1KHz	10KHz
3	100Hz	7.81Hz	100Hz	1KHz

Tabla 4.1. Especificaciones de operación de cada escala.

La decisión de cual filtro utilizar corresponde al usuario, quien tiene acceso por software a un menú de configuración del sistema donde puede seleccionar el filtro que se desee utilizar. Luego, el programa mediante la etapa de entrada/salida comanda un selector analógico el cual conecta la etapa del filtro deseado a la entrada de la etapa del convertidor A/D así como la base de tiempo correspondiente para el reloj del muestreo.

Las gráficas de respuesta en frecuencia de cada filtro se muestran en las Figuras 4.2a, 4.2b y 4.2c en forma de diagramas de Bode.

4.3. Etapa de conversión A/D.

La etapa de conversión A/D está formada por un convertidor análogo digital (ADC) los cuales se utilizan para obtener la secuencia de datos digitales a partir de un proceso de muestreo de la señal analógica de entrada. [8]

Para fines de obtener una alta velocidad de muestreo y de esta forma abarcar una buena parte del espectro, originalmente se había pensado en utilizar un ADC de ráfaga que trabajara en el orden de los MHz; no obstante, luego de enfrentar el problema de la obtención de un circuito de este tipo en el país, se optó por utilizar un convertidor de aproximaciones sucesivas.

El integrado seleccionado para tal fin fué el MP7574 de la Motorola. Entre, otras características, las que determinaron su selección fueron:

a) Velocidad de conversión menor de $20\mu\text{S}$.

Esta velocidad de conversión implica que el muestreo máximo que puede alcanzarse con dicho integrado anda por los 50KHz.

b) Capacidad de manejo de 8 bits.

El ADC1080 tiene una resolución de 11 bits; sin embargo, es posible configurarlo para que trabaje con una resolución de 8 bits uniendo la salida x con y.

c) Rango máximo de entrada de voltaje $\pm 10\text{VDC}$.

Esto nos permite cubrir los alcances planteados en el capítulo I respecto a los rangos de medición. El diagrama esquemático de esta etapa se muestra en la Figura 4.3.

4.3.1. Convertidores de aproximación sucesiva.

El convertidor de aproximaciones sucesivas se basa en un DAC utilizado dentro de un sistema lógico automático que actúa sobre él hasta lograr que su salida corresponda a la entrada. La entrada lógica del DAC es entonces el valor digital de salida buscado. El

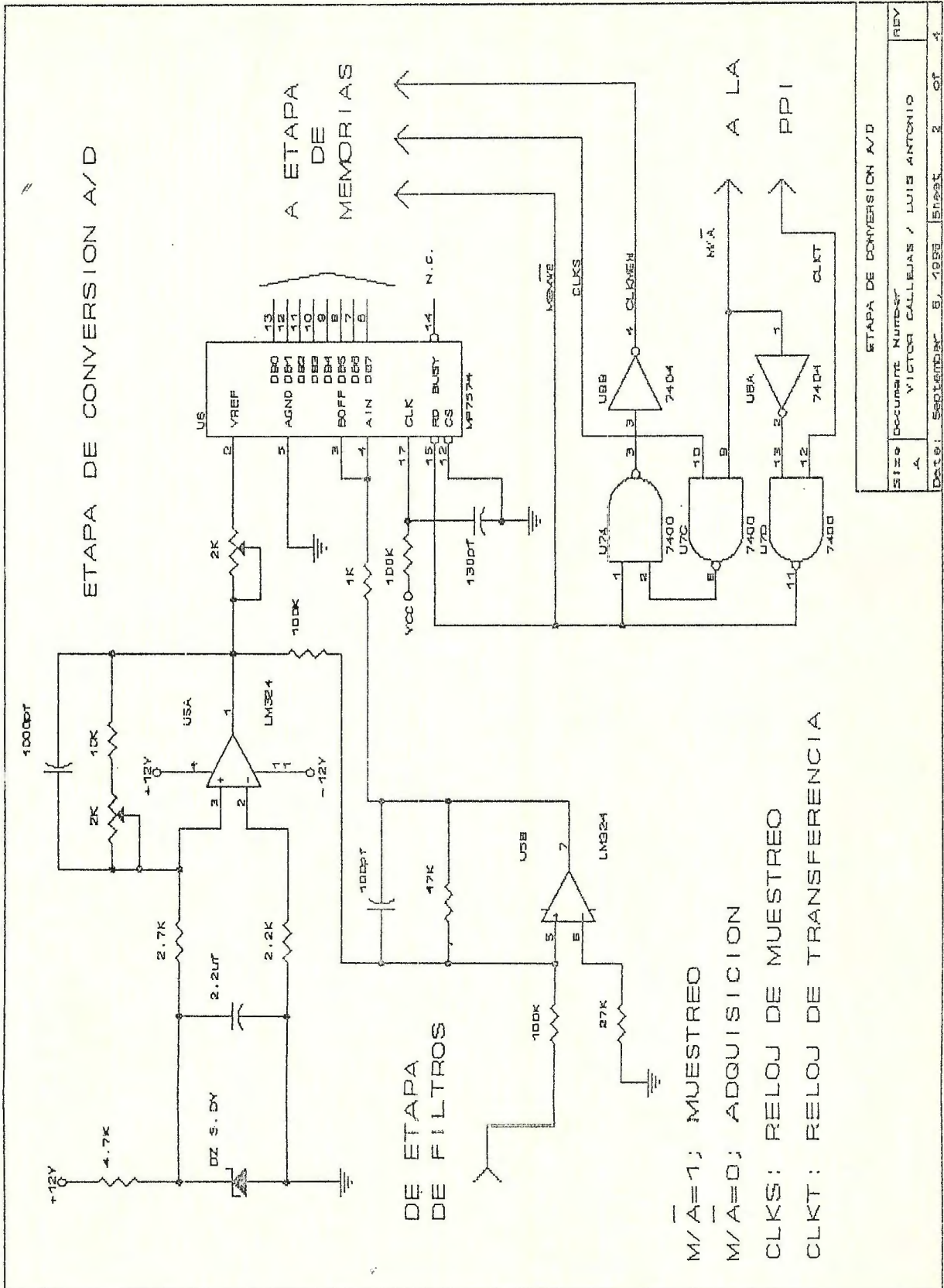


Figura 4.3. Diagrama esquemático de la etapa de conversión A/D.

diagrama elemental de bloques del sistema podría ser el indicado en la Figura 4.4, donde el "registro de aproximaciones sucesivas" sería la red lógica que realiza el algoritmo requerido.

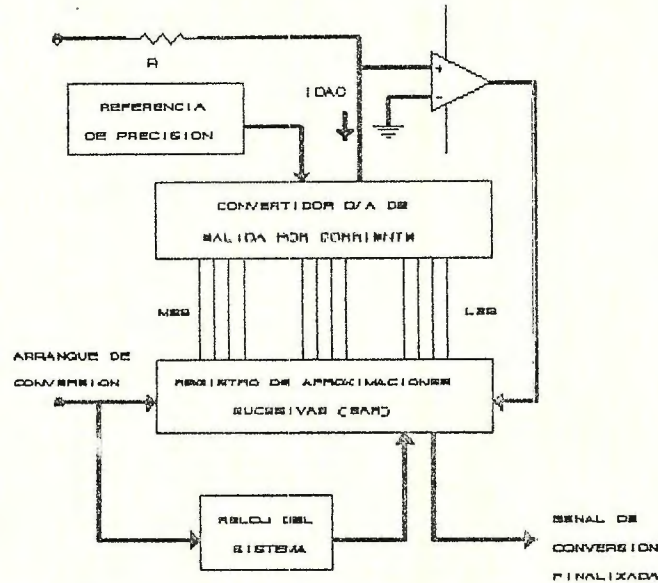


Figura 4.4. Diagrama de bloques de un convertidor de aproximaciones sucesivas.[8]

El comparador compara -valga la redundancia- la señal de entrada con la salida del DAC y devuelve el resultado al registro cuyo contenido al término del ciclo de conversión será el resultado buscado.

En la Figura 4.5 puede verse un diagrama de tiempos que ilustra el funcionamiento del convertidor. Como puede apreciarse, el registro de aproximaciones sucesivas (SAR) empieza con todos sus bits en cero excepto el más significativo (MSB). El valor analógico correspondiente es el de media escala, y un período de reloj después el comparador le indica al registro SAR si la entrada está por encima o por debajo de dicho valor. En el primer caso, el SAR mantiene el bit MSB en uno, mientras que en el segundo lo pone en cero. Luego, el SAR pone en uno el segundo bit más significativo y el proceso

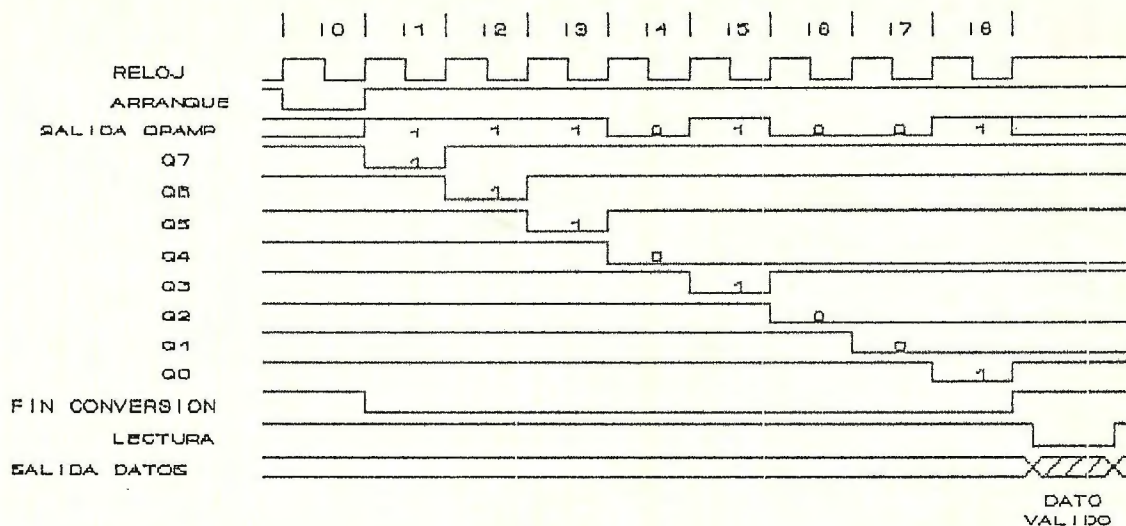


Figura 4.5. Diagrama de tiempos del funcionamiento del ADC.[11]

continúa hasta que se haya establecido y probado el bit LSB (menos significativo). La Figura 4.6 ilustra el resultado del proceso desde el punto de vista de la señal analógica, mostrando como el valor de prueba converge hacia el valor de entrada.

4.4. Etapa de memoria auxiliar.

La memoria auxiliar es una etapa que sirve como almacenamiento temporal de los datos obtenidos durante el proceso muestral de la señal de entrada para su posterior introducción a la computadora. Existen dos razones de peso por las cuales esta etapa resulta necesaria en nuestro proyecto.

El diagrama esquemático de esta etapa se muestra en la Figura 4.7.

La etapa de memoria auxiliar es una etapa necesaria para la independencia de la velocidad de muestreo con el tipo de procesador que posea el CPU a utilizar. Si el reloj de muestreo es generado a

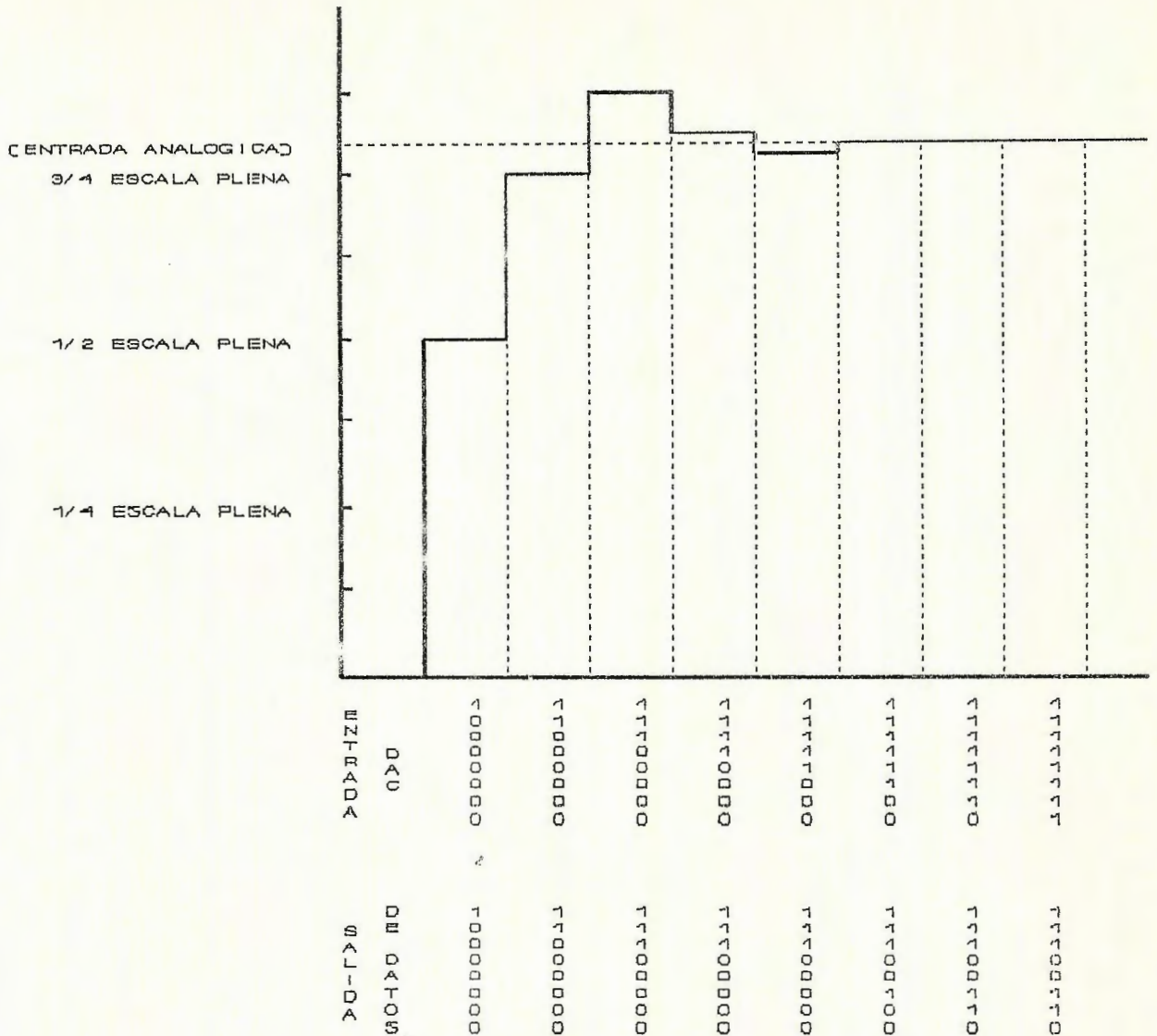
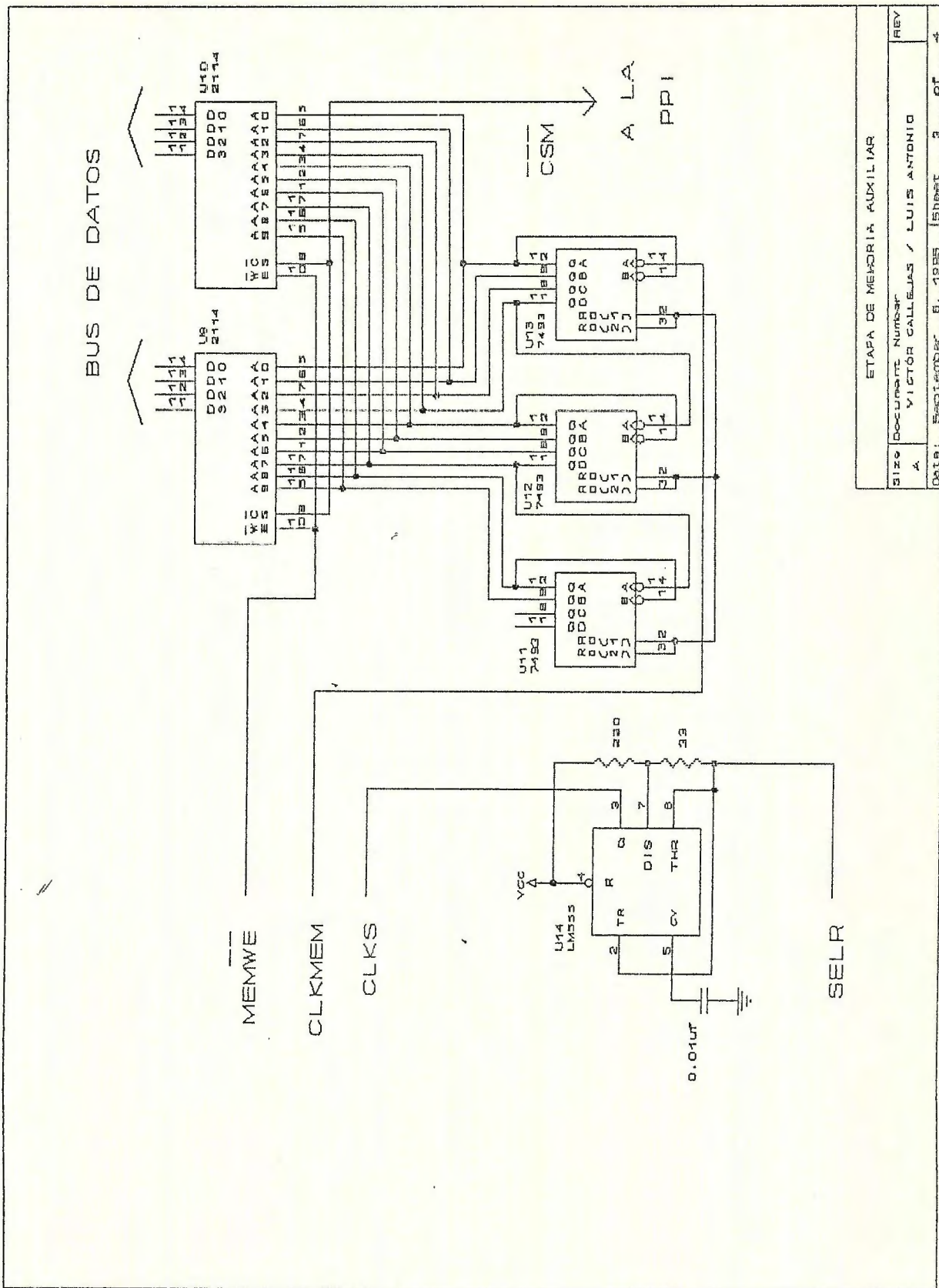


Figura 4.6. Aproximaciones sucesivas.[10]

base de señales de la PC entonces su frecuencia estaría determinada por la velocidad de procesamiento del CPU. Así por ejemplo, si se hubiera diseñado el adquirente de datos bajo ese principio tomando como base para el diseño una máquina 386, cuando el programa se quisiera emplear en una máquina 286 el muestreo realizado sería de más baja frecuencia por lo que las regiones altas de las escalas de medición podrían no cumplir con el teorema del muestreo. Por el contrario, si se utilizara en una máquina 486, la velocidad de



ETAPA DE MEMORIA AUXILIAR			
SIZE	DOCUMENT NUMBER	REV	
A	VICTOR CALLEJAS / LUIS ANTONIO		
DATE: SEPTEMBER 8, 1985	SHEET	3	OF 4

Figura 4.7. Diagrama esquemático de la etapa de memoria auxiliar.

muestreo sería superior, pero la duración del tiempo de adquisición sería menor, por lo que las regiones bajas de las escalas de medición no alcanzarían el muestreo de un ciclo completo de la señal.

La segunda razón por la que esta etapa resulta necesaria es debido a que la velocidad máxima de lectura de datos que puede alcanzarse con la PC, simplemente leyendo datos en forma directa a la memoria sin realizar ningún otro tipo de procesamiento, resulta (en el caso de una PC 386) de aproximadamente 800KHz. Por tanto obligaría una limitante en cuanto a la máxima frecuencia que podría muestrearse e introducirse al sistema. En nuestro caso, la mayor velocidad de muestreo a manejar es de 50KHz; no obstante, la arquitectura de este diseño permite cambiar la etapa de conversión A/D por otra con elementos que trabajen a velocidades superiores y con ello lograr un mayor barrido del espectro.

En resumen, el diseño de esta etapa contribuye a darle mayor flexibilidad al sistema en cuanto a lograr una compatibilidad con varios tipos de CPU que trabajen a velocidades diferentes y obtener velocidades de adquisición superiores mediante el uso de dispositivos externos al CPU capaces de manejar la velocidad requerida.

La etapa de memoria auxiliar está compuesta por dos memorias RAM estáticas de 4x256 bits conectadas en paralelo. Ya que los datos a almacenar y recuperar tienen un carácter secuencial, el direccionamiento de estas memorias es manejado por un contador binario de 10 bits formado con tres contadores binarios de 4 bits, cuyo control de avance es seleccionado de forma alterna por el reloj de muestreo y el reloj de lectura de la PC. Cada vez que se realiza un muestreo, el contador trabaja con el reloj de muestreo.

incrementando el conteo cada vez que se toma una muestra, con lo que el dato se va almacenando en la siguiente localidad de memoria. La duración del muestreo tiene un tiempo determinado que garantiza que la memoria se ha llenado completamente (no importando la dirección de inicio ni la dirección final del muestreo). Una vez finalizado el tiempo de muestreo, el control del contador pasa a manos del reloj de lectura de la PC, la cual comienza a explorar los datos capturados en la "ventana de tiempo" del muestreo en busca de algún cruce por cero (cambio de signo) para utilizarlo como sincronismo. Ya que se cuenta con 256 datos muestreados y sólo se requieren 128 para el análisis, se garantiza que al menos dos ciclos de la señal (en el caso de la frecuencia más baja aceptable por una escala) se encuentran dentro del muestreo y por lo tanto puede recuperarse un ciclo completo de ésta. El hecho de no interesar la dirección de inicio ni la dirección final del muestreo ahorra el uso de rutinas de software para su inicialización y sincronismo que de otra forma significarían mayor tiempo de procesamiento de la señal.

Finalmente y al igual que la etapa anterior, ésta etapa es controlada por la etapa de entrada/salida la cual, mediante líneas de control, determina el tipo de operación a realizar así como la velocidad de muestreo a utilizar.

4.5. Etapa de entrada/salida.

La etapa de entrada salida es la que controla el funcionamiento del resto de etapas. Es además la que se encarga de introducir los datos correspondientes a las señales cuantificadas de entrada que se

almacenaron en la memoria auxiliar dentro de la PC para así poder dar inicio al proceso de obtención del espectro discreto de frecuencia por software. Para los propósitos de éste proyecto, se optó por trabajar con el puerto paralelo de expansión de la PC/AT como medio de interconexión de la tarjeta a la computadora. Para ello, se seleccionó el chip 8255 PPI (Programmable Peripheral Interface) como base para el diseño de dicha interface. El diseño de la etapa se basó en circuitos sugeridos para ser utilizados con microcomputadoras IBM o compatibles.

El diagrama esquemático se muestra en la Figura 4.8.

Los canales E/S de expansión (slots) se encuentran disponibles en todas las PC/AT por lo que con ello se garantiza la compatibilidad de la tarjeta con aquellas PCs que utilicen el mismo tipo de canales de expansión. La descripción de los canales E/S de expansión se muestra en la Figura 4.9. [11]

4.5.1 Descripción de los canales de E/S de expansión de las computadoras IBM PC/AT.

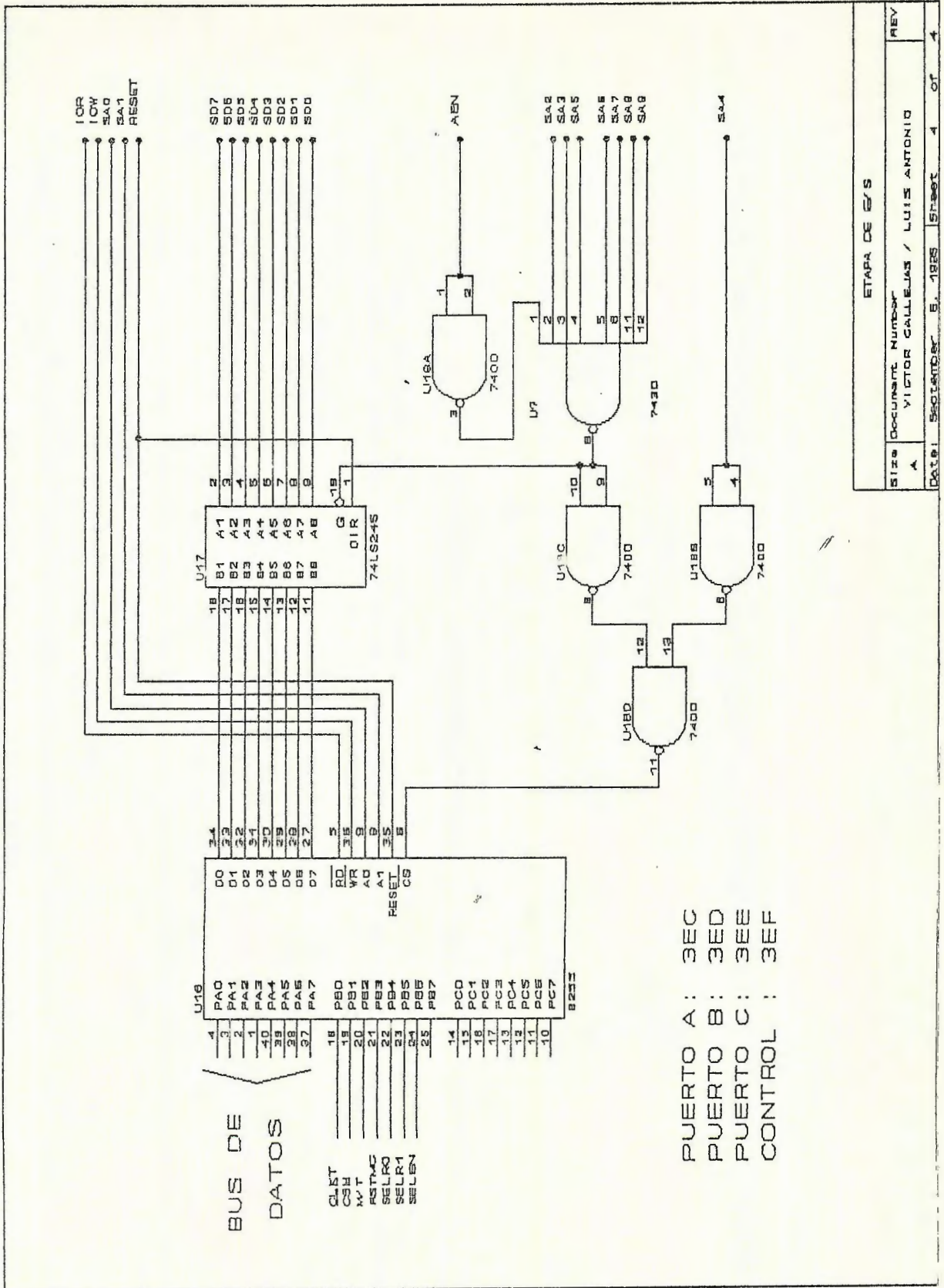
SA0 - SA19 (E/S):

Representa el bus de direcciones del PC y son usados para direccionar memoria y dispositivos de E/S dentro del sistema. Son 20 líneas de direccionamiento que permiten acceder hasta un Megabyte de memoria.

CLK (S):

Es la señal de reloj del sistema. Debe usarse solo para efectos de sincronización.

RESET DRV (S):



ETAPA DE E/S			
Size	Document Number	Author	REV
A	VICTOR CALLEJAS / LUIS ANTONIO		
DATE:	5/20/85	E. 1825	Sheet 4 of 4

Figura 4.8. Diagrama esquemático de la etapa de entrada/salida.

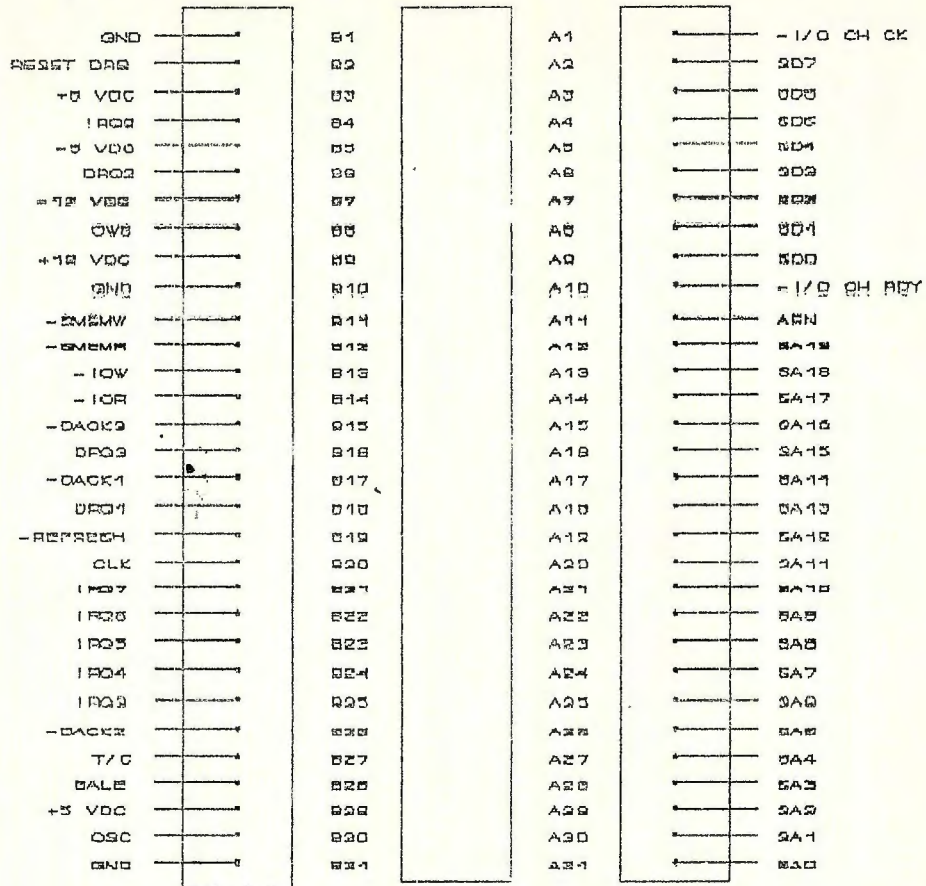


Figura 4.9. Descripción de los canales de E/S de una PC/AT.[11]

Inicializa la lógica del sistema al encender la máquina o durante un bajo voltaje.

SD0 - SD7 (E/S):

Son los 8 bits menos significativos del bus de datos del sistema. En transferencias de 8 bits son los únicos utilizados.

BALE (S):

Abreviatura de "Buffered Address Latch Enable". Esta señal proviene del controlador del bus (82288) y es utilizada para enclavar direcciones válidas del microprocesador. Se utiliza como indicador de direcciones válidas del DMA o del CPU.

I/O CH CK (E):

Esta señal proporciona al sistema información de error de paridad en memoria o dispositivos en los canales de E/S. Es activa en bajo.

I/O CH RDY (E):

Cuando esta señal está en bajo, indica que el dispositivo periférico no está listo, permitiendo prolongar los ciclos de E/S. Es utilizada para manejar dispositivos periféricos lentos.

IRQ3 - IRQ7 (E):

Son señales de solicitud de interrupción, IRQ7 es la de mayor prioridad e IRQ3 la de menor.

IOR (E/S):

Esta señal le indica a un periférico que maneje sus datos a través del bus de datos. La señal es controlada por el microprocesador o el controlador del DMA o similares, presentes en el canal de E/S. Es activa en bajo.

IOW (E/S):

Esta señal le indica a un periférico que lea los datos presentes en el bus de datos. Es controlada en igual forma que el IOR. Es activa en bajo.

SMEMR (S):

Esta señal indica a dispositivos de memoria manejar datos a través del bus de datos. Es controlada por el CPU o el controlador del DMA. Es activa solo para rangos menores de 1Mb de memoria.

SMEMW (S):

Esta señal indica a los dispositivos de memoria almacenar los datos presentes en el bus de datos. Controlada por el CPU o por el controlador del DMA. Es activa solo para rangos menores de 1Mb de memoria.

DRQ1 - DRQ3 (E):

Son las señales de solicitud de transferencia por canales DMA. DRQ1 es la de mayor prioridad y DRQ3 es la de menor prioridad. Se mantiene en alto hasta que el correspondiente DACK se activa.

DACK1 - DACK3 (S):

Se utiliza para atender solicitudes de DMA. Son activas en bajo.

AEN (S):

Cuando esta señal se activa (alto), el CPU cede el control sobre el bus de direcciones y bus de datos al controlador del DMA, permitiendo la transferencia de DMA.

REFRESH (E/S):

Esta señal indica un ciclo de refrescamiento. Es activa en bajo y puede controlarla el microprocesador en el canal de E/S.

T/C (S):

Proporciona un pulso alto cuando es alcanzada la cuenta final por cualquier canal DMA.

OSC (S):

Señal de reloj de 14.31818 MHz no sincronizada.

4.5.2. Funcionamiento de la etapa de interfase.

La etapa de interfaz utiliza las siguientes líneas:

De la A2 hasta la A9, conforman el bus de datos de entrada/salida de 8 bits para el intercambio de datos entre la tarjeta y la PC. El bit menos significativo (LSB) es A2 (SD7) y el más significativo (MSB) es A9 (SD0). Estas líneas son manejadas mediante el 74HC245, que es un transmisor/receptor de datos.

La dirección del flujo de datos, se controla mediante la línea B14 (IOR), la cual se conecta al pin 1 del 74HC245.

La habilitación del 74HC245 (pin 19) es manejada por la salida del decodificador de direcciones; Esta debe ser baja para que el 74HC245 permita el paso de datos.

El decodificador de direcciones no es más que una compuerta NAND de 8 entradas (74LS30), a cuyas entradas se conectan los canales A22-A29 correspondientes a las líneas de direcciones SA9-SA2. Además, se conecta la línea de AEN (A11) a la entrada del 74LS30 para evitar que la interfaz sea habilitada accidentalmente durante la transferencia del DMA. La salida del 74LS30 controla la habilitación del 74HC245 y además, junto con la línea A27 (SA4) controlan a través de una compuerta OR el "chip select" del 8255.

Las líneas de dirección SA1 y SA0, direccionan los puertos del 8255 como lo muestra la Tabla 4.2.

De esta manera, la etapa de entrada/salida responde a 4 direcciones (que son direcciones disponibles en el rango E/S del mapa de memoria de la AT). Tales direcciones son:

3EC: PUERTO A

3ED: PUERTO B

3EE: PUERTO C

3EF: BYTE DE CONTROL

Para el caso, la interfaz se utiliza configurando el puerto A como entrada de datos y el puerto B como una salida. El puerto A se conecta al bus de datos de la tarjeta para poder acceder los datos

provenientes de la memoria auxiliar, una vez ésta se halla llenado.

SA1	SA0	Acción realizada.
0	0	Direcciona el puerto A.
0	1	Direcciona el puerto B.
1	0	Direcciona el puerto C.
1	1	Direcciona el byte de control del 8255.

Tabla 4.2. Combinaciones de entrada para direccionamiento de puertos.

El puerto B se utiliza como puerto de control asignando a cada línea del puerto una señal de comando para algunos de los dispositivos al interior de la tarjeta. A continuación se describe cada una de estas líneas así como su función:

CLKM (PB0): Esta línea está conectada mediante un arreglo de compuertas NAND al reloj de entrada del contador que maneja las posiciones de memoria. De esta forma, el CPU puede acceder secuencialmente todas las posiciones de memoria simplemente con generar un reloj a través de éste pin.

CSM (PB1): Esta otra línea controla la selección de chip de las memorias de la etapa de memoria auxiliar. Las memorias son activas en BAJO e inactivas en ALTO.

M/T (PB2): Esta línea es la que controla el arreglo de compuertas NAND antes mencionado y sirve para configurar el modo de operación de la tarjeta. Un ALTO en esta línea configura la tarjeta para la operación de muestreo habilitando el paso del reloj de muestreo al contador de memoria. Un BAJO selecciona la operación de transferencia de datos inhabilitando el reloj de muestreo y habilitando el reloj de transferencia.

RSTMC (PB3): Permite reinicializar el contador de memoria poniendo todos sus registros en cero. Esta señal es activa en ALTO.

SELRO y SELR1 (PB4 y PB5): Estas líneas se utilizan para seleccionar el rango de operación de la tarjeta en su modo de muestreo. Están conectadas a las líneas de selección de un multiplexor analógico dual el cual selecciona una entrada de las provenientes de los filtros al mismo tiempo que selecciona el reloj de muestreo correspondiente para dicha escala.

SELEN (PB6): Esta es la señal de selección del multiplexor analógico. Es activa en ALTO.

De esta manera, el control de las operaciones que la tarjeta realiza son controladas por el programa principal mediante el envío de caracteres de control al puerto B de la PPI en la etapa de E/S.

4.6. El PPI 8255 (Interfase Programable de Periféricos).

Para comprender más claramente el funcionamiento de la interfase es necesario describir las operaciones que puede realizar el 8255, el cual es un periférico programable de entrada/salida paralelo para propósitos generales y es la parte central del circuito de la interfase, el diagrama del bloque del 8255A es mostrado en la Figura 1.4. Es de la familia INTEL de circuitos de soporte para microprocesadores, capaz de proporcionar 24 pines programables de entrada/salida, divididos en tres puertos paralelos: puerto A, puerto B y puerto C; los cuales pueden ser configurados de manera especial como se describe más adelante. La configuración se hace por medio de software y no requiere circuitos lógicos externos para ello.

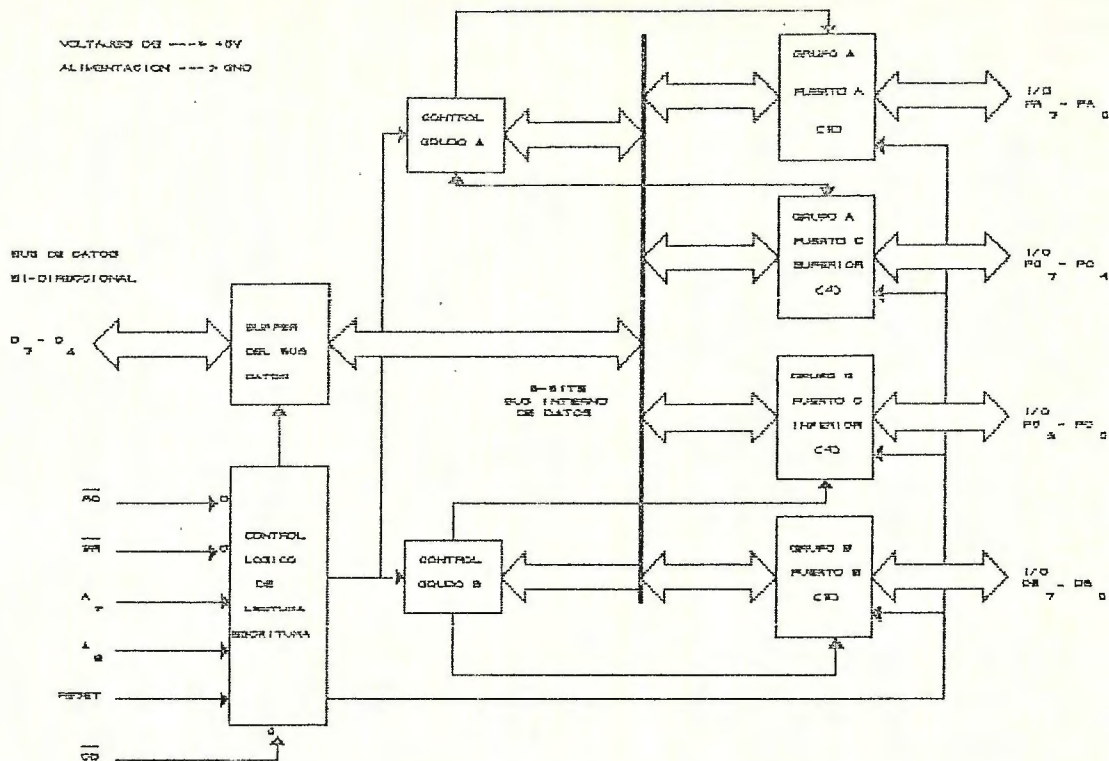


Figura 4.10. Diagrama en bloques de la PPI 8255.[11]

Antes de seguir describiendo las características de operación del 8255, cabe mencionar que todo lo que hace, podría lograrse con simples compuertas y dispositivos similares, sin embargo esto involucra mayor costo por bit de entrada/salida y mucho más espacio físico, complicando el diseño.

4.6.1. Modos de Operación del 8255.

El 8255 posee tres modos de operación conocidos como modo 0, modo 1 y modo 2. En el modo 0 los 8 pines de cada uno de los tres puertos son utilizados como entrada/salida. Los modos 1 y 2 utilizan las líneas del puerto C como entradas de solicitud de interrupción al CPU, estas interrupciones pueden ser enmascaradas o no mediante el formato bit set/reset.

El 8255 presenta tres modos básicos de operación:

- Modo 0: Entrada/Salida elemental.
- Modo 1: Entrada/Salida con protocolo (Handshaking).
- Modo 2: Bus bidireccional con protocolo (Handshaking).

El 8255 esta conformado por dos grupos de buses. El grupo A que es el puerto A más los 4 bits más significativos del puerto C y el grupo B esta formado por el puerto B más los cuatro bits menos significativos del puerto C.

Cuando se inicializa el 8255 (al encender la máquina), sus tres puertos quedan configurados como puertos de entrada. Los puertos A,B y C pueden ser configurados en modos de operación diferentes según sea necesario, escribiendo la palabra de control en el registro de control.

La palabra de control consta de 8 bits y es la que determina en que modos pueden trabajar los tres puertos; el registro de control es la dirección a la que responde la palabra de control.

Las direcciones a las que responde la interfase son las siguientes (tres de estas direcciones corresponden a los puertos A,B y C y el cuarto es un registro de control de "solo escritura").

SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0		
1	1	1	1	1	0	1	1	0	0	3E0H	PUERTO A
1	1	1	1	1	0	1	1	0	1	3E2H	PUERTO B
1	1	1	1	1	0	1	1	1	0	3E4H	PUERTO C
1	1	1	1	1	0	1	1	1	1	3E6H	Palabra de control

El formato del byte de control se muestra a continuación:

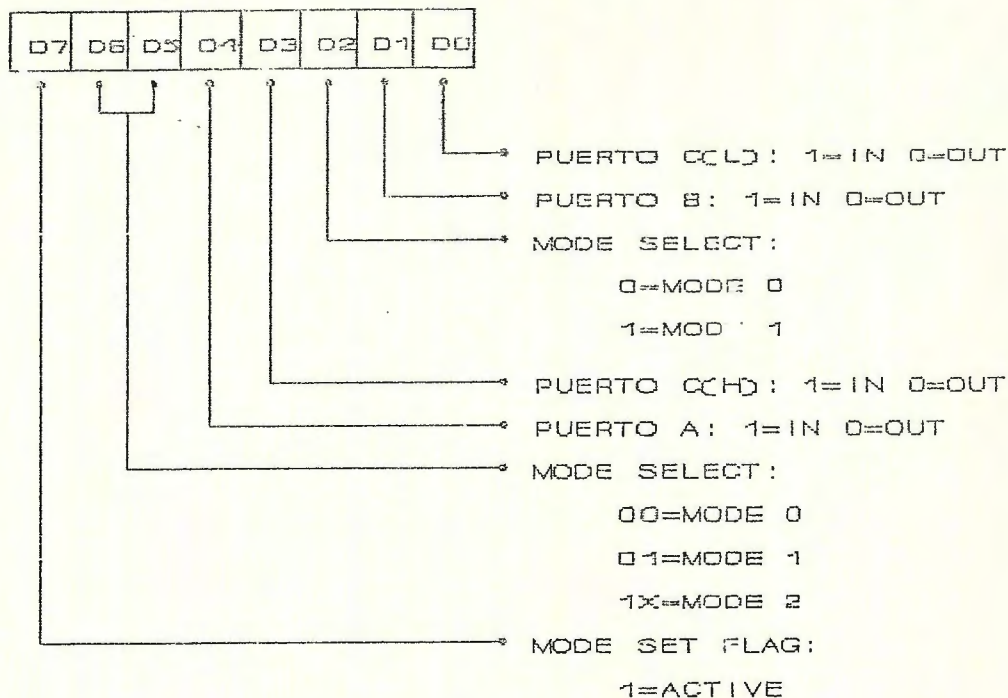


Figura 4.11. Formato del Byte de control.

4.6.2. Modo 0: Entrada/Salida Elemental.

Este modo configura a los tres puertos solamente como entrada/salida. Este modo de operación presenta las siguientes características:

- Los puertos A y B de 8 bits cada uno y el puerto C dividido en dos grupos de 4 bits cada uno, los cuales pueden ser entrada o salida.
- Cualquier puerto puede ser de entrada o salida.
- Las salidas son enclavadas.
- Las entradas no son enclavadas.
- 16 formas de configurar en este modo.

A continuación en la Tabla 4.3 se muestran las 16 posibilidades de

configuración en modo 0.

Por razones de facilitar las conexiones en el diseño de la interface, ya que esta debe implementarse en circuito impreso, el orden del bus de datos esta invertido; teniendo D7 como el LSB y el D0 como el MSB, así también las salidas de los puertos estan de igual manera. Unicamente es transposición de líneas por lo tanto cambio de palabra de control, el cual es mostrado en la Tabla 4.3.

BYTE DE CONTROL	PUERTO A	4MSB PUERTO C	PUERTO B	4LSB PUERTO C
01 HEX	salida	salida	salida	salida
81 "	salida	salida	salida	entrada
41 "	salida	salida	entrada	salida
C1 "	salida	salida	entrada	entrada
11 "	salida	entrada	salida	salida
91 "	salida	entrada	salida	entrada
51 "	salida	entrada	entrada	salida
D1 "	salida	entrada	entrada	entrada
09 "	entrada	salida	salida	salida
89 "	entrada	salida	salida	entrada
49 "	entrada	salida	entrada	salida
C9 "	entrada	salida	entrada	entrada
19 "	entrada	entrada	salida	salida
99 "	entrada	entrada	salida	entrada
59 "	entrada	entrada	entrada	salida
D9 "	entrada	entrada	entrada	entrada

Tabla 4.3. Configuración de puertos de la PPI en el modo 0.

Para una mejor comprensión la selección de la palabra de control se hace según la necesidad que se va a tener, por ejemplo: si se quiere tener como puertos de salida el A y el B y entrada el puerto C con las instrucciones de Pascal que permite tener acceso directo a los puertos, se escribe de la siguiente manera `Port[$3EF]:=$91`, y el 8255 queda así configurado. Para los propósitos de control en el Módulo de prueba la elección del modo 0 para la interfase es el más ideóneo y por esta razón se ha desarrollado con mayor profundidad, dejando los otros dos modos como información general.

4.6.3. Modo 1: Entrada/Salida con protocolo (Handshaking)

Este modo permite comunicarse con el computador con el uso de señales de protocolo o handshaking. Las características principales de este modo son:

- Presenta dos grupos (A y B) de líneas de comunicación.
- Cada grupo esta formado por un puerto de 8 bit para datos y un puerto de 4 bit para control (Handshaking).
- El puerto de 8 bit puede ser entrada o de salida. Las entradas y salidas son enclavadas.
- El puerto de 4 bit es usado para controlar el estado del puerto de 8 bits.

En este modo de operación hay dos posibilidades para cada bloque: entrada con protocolo o salida con protocolo. En la configuración como puerto de entrada con protocolo, las señales de Handshaking son las siguientes:

1. -STB: Un cero en esta señal de entrada, carga el dato en el latch de entrada.
2. IBF: Un uno en esta señal de salida, indica que el dato ha sido cargado en el latch de entrada.
3. INTR: Un uno en esta señal de salida puede ser utilizada para interrumpir el CPU cuando un dispositivo de entrada solicita su servicio. Se activa cuando -STB y IBF son unos y hay habilitación de interrupciones. Se activa y desactiva cuando ocurre una operación de lectura.

La interrupción puede ser "enmascarada" o no mediante el bit set/reset del PC4 para el grupo A y el set/reset del PC2 del grupo B.

GRUPO A. BYTE DE CONTROL

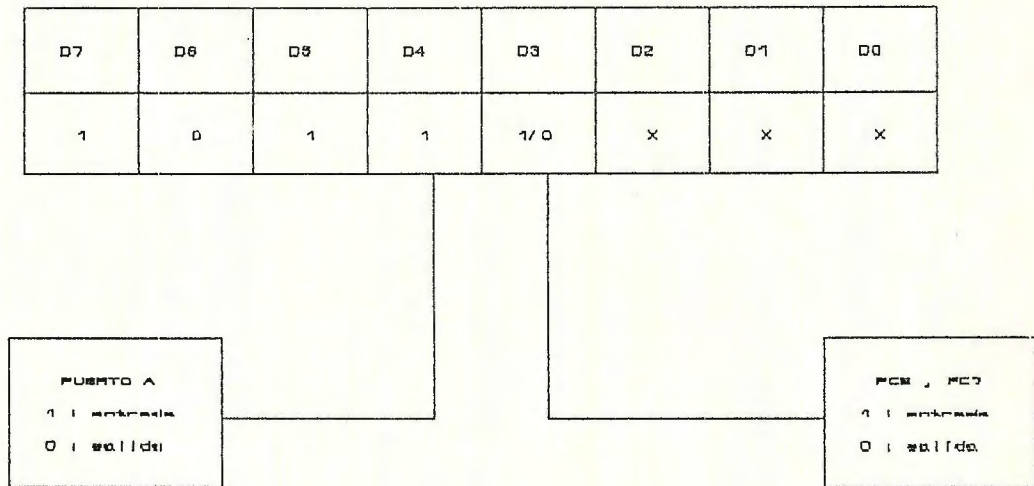


Figura 4.12. Byte de control del grupo A.

En la configuración de puerto de salida con protocolo, las señales de Handshaking son las siguientes:

- 1.-OBF: Un cero en esta señal de salida, indica que el CPU ha escrito un dato al puerto de salida.
- 2.-ACK: Un cero en esta señal de entrada, le indica al 8255A que el dato que se envió ha sido aceptado.
- 3.INTR: Un uno en esta señal de salida, puede ser utilizada para interrumpir al CPU cuando un dispositivo de salida ha aceptado datos enviados por el CPU. Se activa cuando -OBF y -ACK son unos y hay habilitación de interrupción. Se activa cuando se hace una operación de escritura.

Las interrupciones pueden "enmascarse" o no mediante el bit set/reset del PC6 para el grupo A y el bit set/reset para del PC2 para el grupo B.

GRUPO B. BYTE DE CONTROL

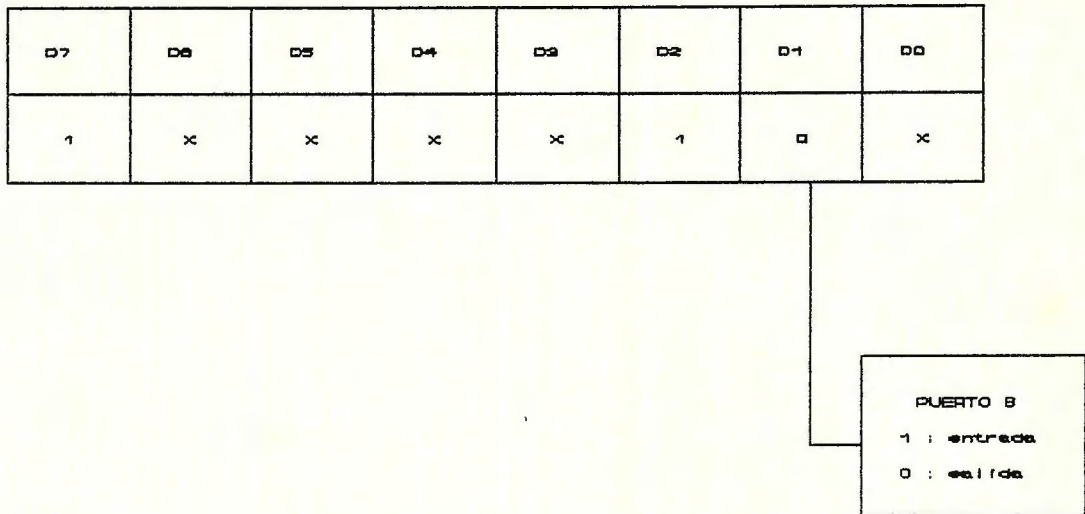


Figura 4.13. Byte de control del grupo B.

4.6.3. Modo 2: Puerto Bidireccional con protocolo.

Este modo opera con un solo puerto que sirve para enviar y recibir datos utilizando señales de protocolo. Sus características principales son:

- Usando en el grupo A únicamente.
- Un puerto de 8 bits bidireccional y un puerto de control de 5 bit.
- Entradas y salidas enclavadas.

Las señales INTR, -OBF, IBF, -ACK y -STB, tienen el mismo significado en este modo de operación que el modo 1. El modo 2 es exclusivo del puerto A y lo convierte en un puerto bidireccional con señales de protocolo. El puerto B no puede operar en esta configuración.

Existen muchas maneras de combinar los modos de operación en el

8255A. El puerto A puede operar de tres modos diferentes, el puerto B de dos modos diferentes. La forma de configurar los puertos depende de tipo de aplicación que se desee.

CAPITULO V

ELABORACION DEL SOFTWARE DE CONTROL.

El software para manejar la tarjeta adquisitora de datos, así como el que sirve para el procesamiento y presentación de la información se hizo principalmente en lenguaje TurboPascal. No obstante, ciertas rutinas tales como la de obtención del número invertido o la de lectura de puertos se realizaron en lenguaje Ensamblador para aprovechar la alta velocidad de procesamiento de éste lenguaje. A continuación se presenta una breve introducción a estos lenguajes de programación y luego los algoritmos del programa de control.

5.1. Lenguaje TurboPASCAL

5.1.1 Reseña histórica del lenguaje PASCAL.

El lenguaje Pascal utilizado en el trabajo de graduación, se eligió por las siguientes razones:

1. El acceso a puertos de entrada/salida.
2. Por la compatibilidad de Turbo Pascal con rutinas en lenguaje ensamblador.
3. Porque era el lenguaje de programación que ya se conocía, por lo tanto, una herramienta de la cual ya se disponía.

Este lenguaje nació de la necesidad de hacer una comunicación completa, inequívoca y fácil de aprender e implementar en una computadora.

Analizando la historia de los diferentes lenguajes de programación

puede observarse que uno de los primeros en aparecer fue el FORTRAN (FORmula TRANslator). El FORTRAN fué uno de los lenguajes de uso más extendido en la rama de los cálculos científicos. Debido a su temprana aparición, actualmente se ha convertido en una colección de funciones que resultan útiles, pero incómodas para aprender y manejar en una computadora.

Se hicieron intentos para definir un lenguaje sencillo directamente inspirado en el FORTRAN y que fuera fácil de aprender y pudiera usarse en forma conversacional. El resultado de estos intentos fue el BASIC (Beginner's All-purpose Symbolic Instruction Code). El lenguaje BASIC es fácil de implementar en cualquier computadora y requiere solamente un pequeño esfuerzo de memoria. Estas dos cualidades (fácil implementación y sencillez) han propiciado que el BASIC fuera el lenguaje de programación más usado en microcomputadoras, aunque, por otro lado, presenta grandes limitaciones debido a sus reglas de uso ("sintaxis") que, a menudo, le hacen inadecuado para la realización de programas complejos. [12]

Otro lenguaje, ALGOL (ALGOritHms Language), surgió como resultado del intento de crear otro lenguaje de programación distinto del FORTRAN que fuera consistente y apropiado para programar complicados algoritmos. El ALGOL alcanzó gran popularidad en ambientes educativos, aunque nunca llegó a ser ampliamente utilizado. Aunque este lenguaje es una herramienta eficaz para la programación de complicados algoritmos, es difícil de aprender y de implementar.

El lenguaje Pascal se inspiró en los lenguajes ALGOL y PL/I, y representa un intento de crear un medio fácil de aprender, apropiado para la realización de algoritmos complejos y la definición de

estructuras de datos. Fue creado por Niklaus Wirth, perteneciente al ETH, Instituto Técnico de Zurich, en los años 1970-1971, después de su regreso de la Universidad de Stanford. Rápidamente ganó aceptación en los círculos educativos como una buena herramienta para aprender a programar. Se presenta, además, como un lenguaje relativamente simple y muy coherente, por lo que el compilador necesario para su implementación requiere una pequeña porción de memoria. El nombre de este lenguaje rinde tributo al matemático francés Blaise Pascal, quien, en 1690, a la edad de dieciocho años, inventó la primera máquina de calcular.

5.1.2. El TurboPASCAL y otras modalidades del lenguaje PASCAL.

El lenguaje Pascal fue definido originalmente por Niklaus Wirth y Kathleen Jensen en su libro *Pascal User Manual and Report*, publicado por primera vez en 1974. Más tarde, esta definición fue formalizada por International Standards Organization (ISO) como Nivel 0 ISO Standards Pascal (ISO dp 7185) en 1980. La forma de Pascal adoptada por ISO fue aceptada (con muy pequeñas variaciones) en Estados Unidos por American National Standards Institute (ANSI) y el Institute of Electrical and Electronics Engineers (IEEE) como el lenguaje Pascal patrón en 1983 (ANSI/IEEE770X3.97-1983). La definición del lenguaje Pascal hecha por ISO es prácticamente idéntica a la elaborada por Wirth originalmente, pero mucho más precisa.

ISO Standard Pascal se llamará Pascal estándar por simplicidad. Actualmente hay, sin embargo, gran número de modificaciones y ampliaciones hechas sobre el Pascal estándar.

A medida que se extendía el lenguaje Pascal se fue modificando la definición actual, se añadieron algunos elementos nuevos y otros fueron interpretados de distintas formas cuando aparecieron ambigüedades.

Como ocurre con cualquier lenguaje de programación, el Pascal depende, en última instancia, de su implementación concreta. En teoría, aprender Pascal supone no sólo aprender Pascal estándar, sino, también, aprender los elementos y las diferencias propias de la versión que se utilice. Afortunadamente, en la práctica, todas las versiones responden al lenguaje Pascal estándar con algunos pocos elementos nuevos y cambios poco significativos.

El lenguaje Pascal fue diseñado, originalmente, para computadoras que trabajan por "lotes", donde el programa se introdujera en un conjunto de tarjetas, o de una cinta magnética; pero, a medida que el uso de este lenguaje se hizo más popular, comenzó a instalarse en sistemas de tiempo compartido y microcomputadoras, en los que el usuario tiene acceso directo a una terminal.

Debido precisamente a la continua interacción entre la terminal y el usuario, posible en este tipo de computadoras, fue necesaria la implementación de nuevos elementos y surgió la versión UCSD Pascal. Esta versión fue desarrollada en la Universidad de California de San Diego (UCSD), y fue instalada en muchas computadoras pequeñas, incluido el Apple II, por el año 1978. Esta nueva versión contenía varias innovaciones respecto del Pascal estándar, incluida una técnica que permitía fácilmente la realización de gráficos en la pantalla y que recibió el nombre de Turbo Graphics.

En 1983, Borland International ultimó el "Turbo Pascal", un

compilador muy sofisticado para las computadoras CP/M, IBM PC-DOS y MS-DOS. La versión Turbo Pascal es un compilador potente, barato y rápido que incluye, además, detección de errores y procesador de textos en el programa editor. Todos estos elementos configuran en la programación en Turbo Pascal.

5.1.3. Escritura de un programa en PASCAL.

Una vez que la solución de un problema cualquiera ha sido expresada en forma de algoritmo, éste debe ser convertido en un programa en Pascal. Una vez que se ha escrito el programa, el compilador debe traducirlo y proceder a su ejecución. Algunos otros programas como el editor y el sistema de archivos están en continua interacción con la computadora para facilitar estos procesos.

5.1.4. La escritura del lenguaje PASCAL.

El Pascal es un lenguaje de alto nivel y permite al programador escribir sus instrucciones en un idioma similar al inglés pero muy restringido. Para evitar cualquier ambigüedad en la traducción que del programa hace el compilador a código binario, es necesario que la sintaxis imponga unas reglas muy estrictas.

La programación requiere ingenio e inteligencia, pero también necesita de una férrea disciplina. Cualquier intrucción u orden escrita en Pascal debe seguir estrictamente las reglas de sintaxis, porque cualquier instrucción que viole alguna de estas reglas establecidas inutiliza el programa. No hay ninguna excepción. Es, por

tanto, esencial entender y respetar la sintaxis para llevar a buen fin la escritura de cualquier programa. La omisión de una simple coma hace que no funcione un programa. Por más que se diga al respecto nunca se resaltaré lo suficiente la necesidad de seguir una estricta disciplina a la hora de programar.

Las reglas del lenguaje Pascal pueden describirse de muchas formas: usando palabras, la notación BNF (Backus-Naur Form), o diagramas sintácticos.

5.1.5. Formatos de un programa en PASCAL.

El lenguaje Pascal fue creado para motivar la programación por bloques, es decir: cada paso o grupo de pasos que guarden una fuerte relación lógica dentro de un algoritmo pueden agruparse en un bloque o módulo. Cada uno de estos módulos recibe el nombre de *bloque*, *función* o *procedimiento*, dependiendo de la forma en que se usen.

La sintaxis del lenguaje Pascal requiere que todas las definiciones y declaraciones aparezcan al principio del programa, por lo que la organización general puede representarse como en la Figura 5.1.

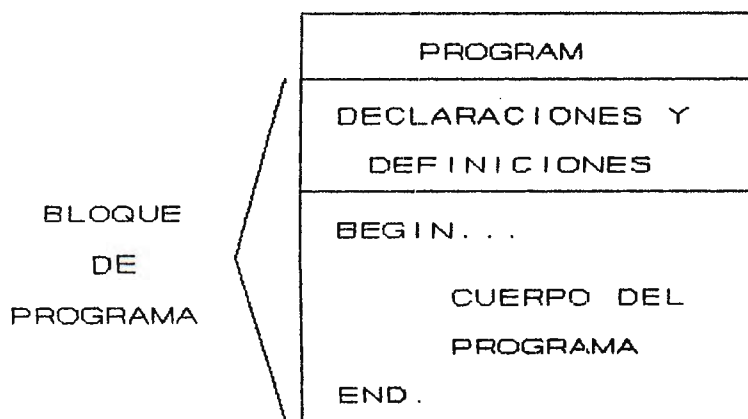


Figura 5.1. Organización general de un programa en PASCAL.

Como puede apreciarse en la Figura 5.1, todas las definiciones y declaraciones aparecen al principio del programa, seguidas por el bloque principal, que está determinado claramente por las palabras BEGIN y END.

Ahora se va a considerar la organización de cada uno de los módulos reflejados en la Figura 5.1. Un programa en Pascal puede verse en detalle en la Figura 5.2.

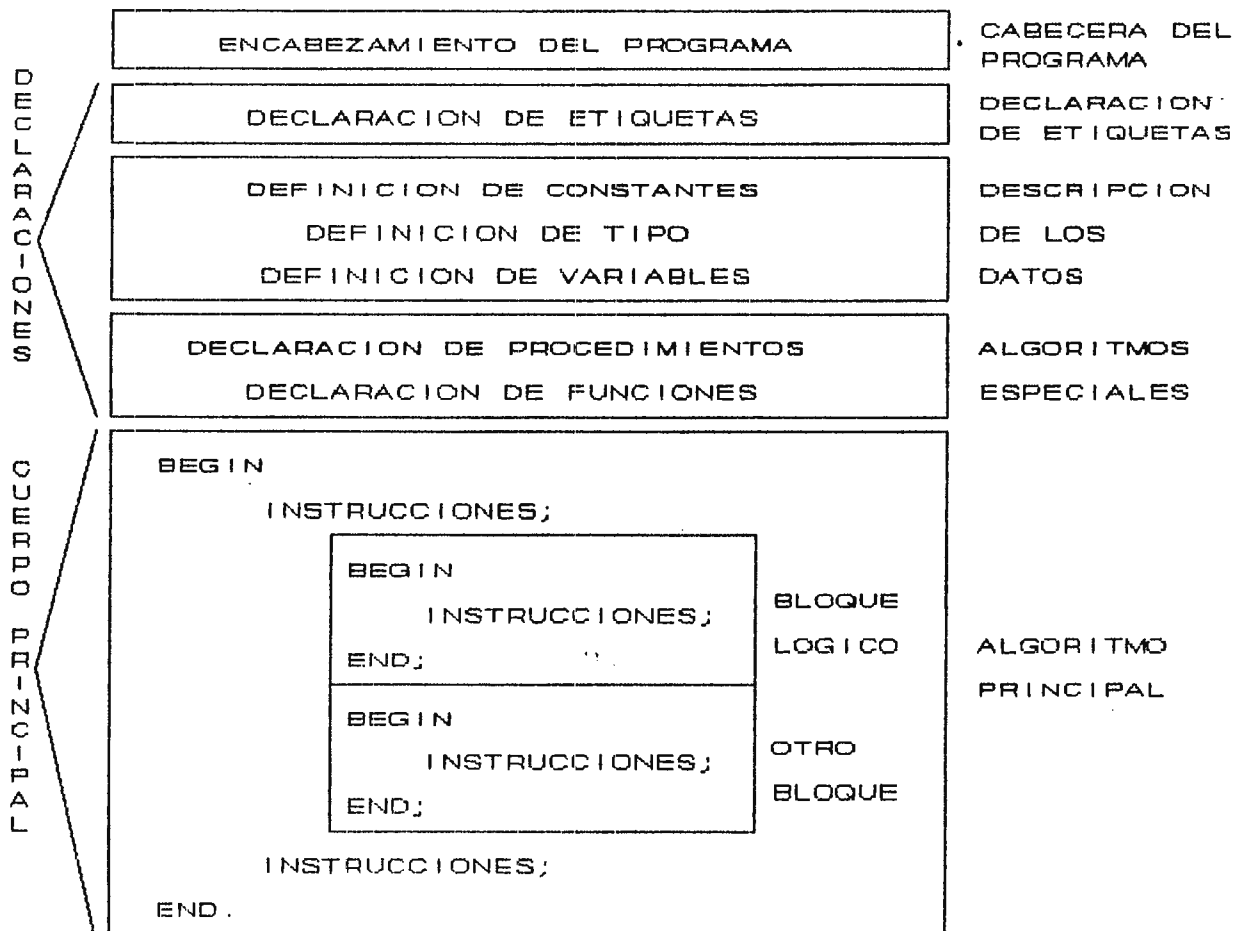


Figura 5.2. Organización detallada de un programa en PASCAL.

5.1.6. Declaraciones.

Los diferentes tipos de declaraciones que existen en Pascal deben aparecer, exactamente, en el orden en que se han colocado en la Figura 5.2: Primero las etiquetas, después las constantes, etc. Sin embargo, ninguna de ellas es obligatoria. Por ejemplo, en el siguiente programa:

```
PROGRAM SALUDO(OUTPUT);
  { ESTE ES UN PROGRAMA EN PASCAL }
  BEGIN
    WRITELN('HOLA')
  END.
```

no existen declaraciones. Únicamente se incluyen la cabecera y el cuerpo principal de la parte ejecutable. Hay que recordar siempre que los comentarios son ignorados por el compilador.

Normalmente, en cuanto el programa tenga más que unas pocas líneas, es necesario usar variables y, por tanto, es obligada su declaración explícita.

Refiriéndose al esquema de la Figura 5.2, hay que decir que las etiquetas se usan algunas veces y que las funciones y los procedimientos se usan cuando el programa tiene una longitud grande. Las únicas declaraciones necesarias en programas cortos son: CONST, TYPE y VAR.

5.1.7. Resumen de la organización de un programa.

En resumen, cada programa en Pascal debe tener, al menos, un encabezamiento o cabecera y una instrucción. Además, el programa puede contener varias declaraciones o definiciones siguiendo a la cabecera (en el orden adecuado), y un número cualquiera de

instrucciones. Los comentarios, los espacios en blanco y los sangrados que se hagan al escribir el programa pueden colocarse en cualquier lugar para facilitar su lectura.

5.1.8. Organización formal de un programa.

En relación con esto, la organización del programa será representado en la Figura 5.3.

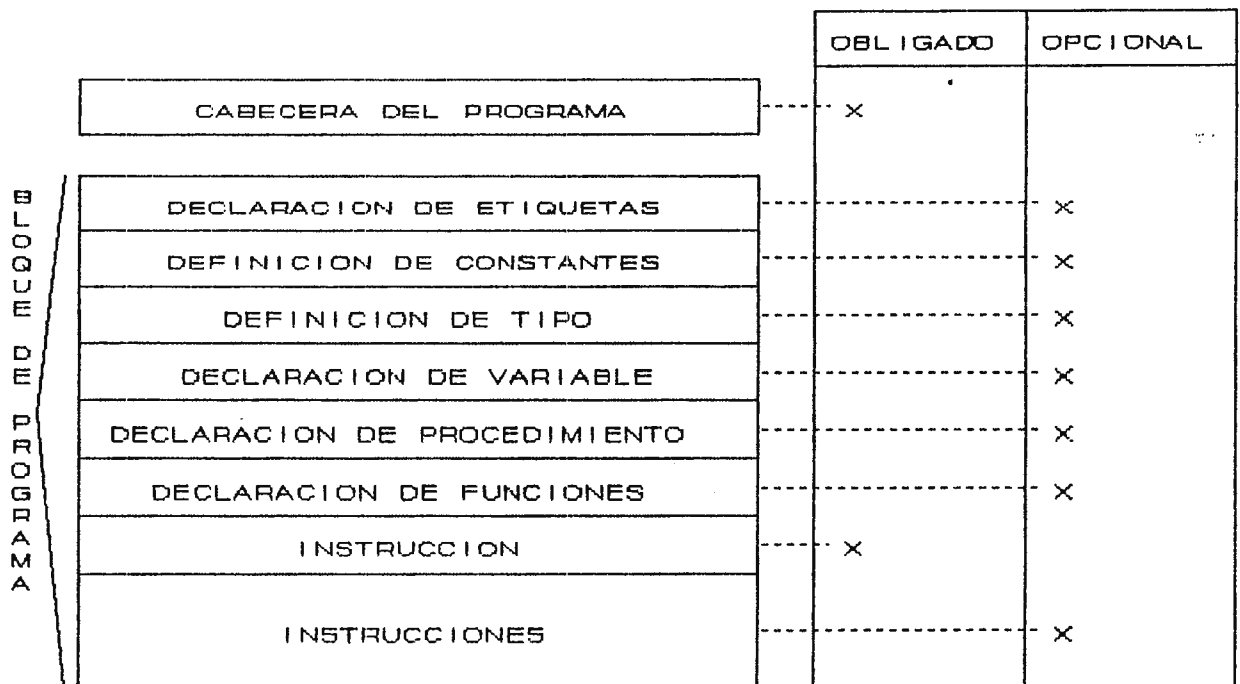


Figura 5.3. Organización formal de un programa en PASCAL.

Nótese que en la definición formal de un programa en Pascal el *bloque del programa* se refiere a todo lo que sigue en la cabecera, incluido el conjunto de declaraciones.

5.2. LENGUAJE ENSAMBLADOR.

5.2.1. El Lenguaje Ensamblador.

El ensamblador es un variante -legible para el ser humano- del lenguaje de máquina que usan las computadoras para ejecutar programas. Al mismo tiempo es la mejor manera que tiene usted de comunicarse con la PC (también conocida como microcomputadora) y con el lenguaje de programación que utilice. Este lenguaje da la facilidad y las herramientas necesarias para tomar el control de todo lo que la PC pueda realizar físicamente.

El ensamblador resulta indispensable cuando se desea escribir programas que controlen la entrada/salida de la PC, agregar nuevas interfases de entrada/salida, escribir rutinas optimizadas de un procedimiento en especial, escribir rutinas que aprovechen y maximicen el uso del hardware, y en general realizar cualquier tarea que no puedan llevar a cabo los demás lenguajes de programación.

5.2.2. Ventajas del Lenguaje Ensamblador.

Aunque cualquier lenguaje de programación ofrece facilidades para realizar determinadas tareas, ninguno proporciona un control absoluto sobre la PC, ni información fundamental de lo que sucede mientras el programa se ejecuta.

Si nunca se ha trabajado con ensamblador, se debe preguntar por qué no usar Basic, Pascal, Clipper, Modula o C. Existen varias razones que justifican el uso de ensamblador en lugar de otros lenguajes.

- La primera es que el ensamblador le brinda la oportunidad de conocer más a fondo la operación de la PC. Esto permite implementar software o hardware de una manera más consciente (comprendiendo cómo y por qué lo hace), en vez de seguir una receta fija.
- La segunda razón es que se conserva el control de lo que deberá hacer la PC, siempre y cuando ésta sea físicamente capaz de hacerlo.
- La tercera es que los programas en ensamblador son más rápidos, más compactos y tienen mayor capacidad que los creados en otros lenguajes.
- La cuarta razón, quizá la más importante, es que se puede optimizar al máximo sus programas (tanto en tamaño como en velocidad de ejecución). En el caso de los otros lenguajes ya existe un código que se genera de manera predefinida, haciendo muy difícil su optimización. A continuación se presenta una tabla que indica los tamaños de los archivos (fuente, objeto y finalmente el ejecutable) creados por diferentes compiladores para un mismo programa.

LENGUAJE	FUENTE	OBJETO	EJECUTABLE
Basic	27 bytes	651 bytes	12,814 bytes
Clipper	22 bytes	572 bytes	159,178 bytes
Ensamblador	162 bytes	169 bytes	543 bytes

No se están comparando las velocidades de ejecución, aunque si tomamos en cuenta el tamaño del programa ejecutable (cuya extensión es .EXE), resulta obvio que mientras más grande sea éste, su ejecución será más tardada debido al tiempo necesario para cargarlo.

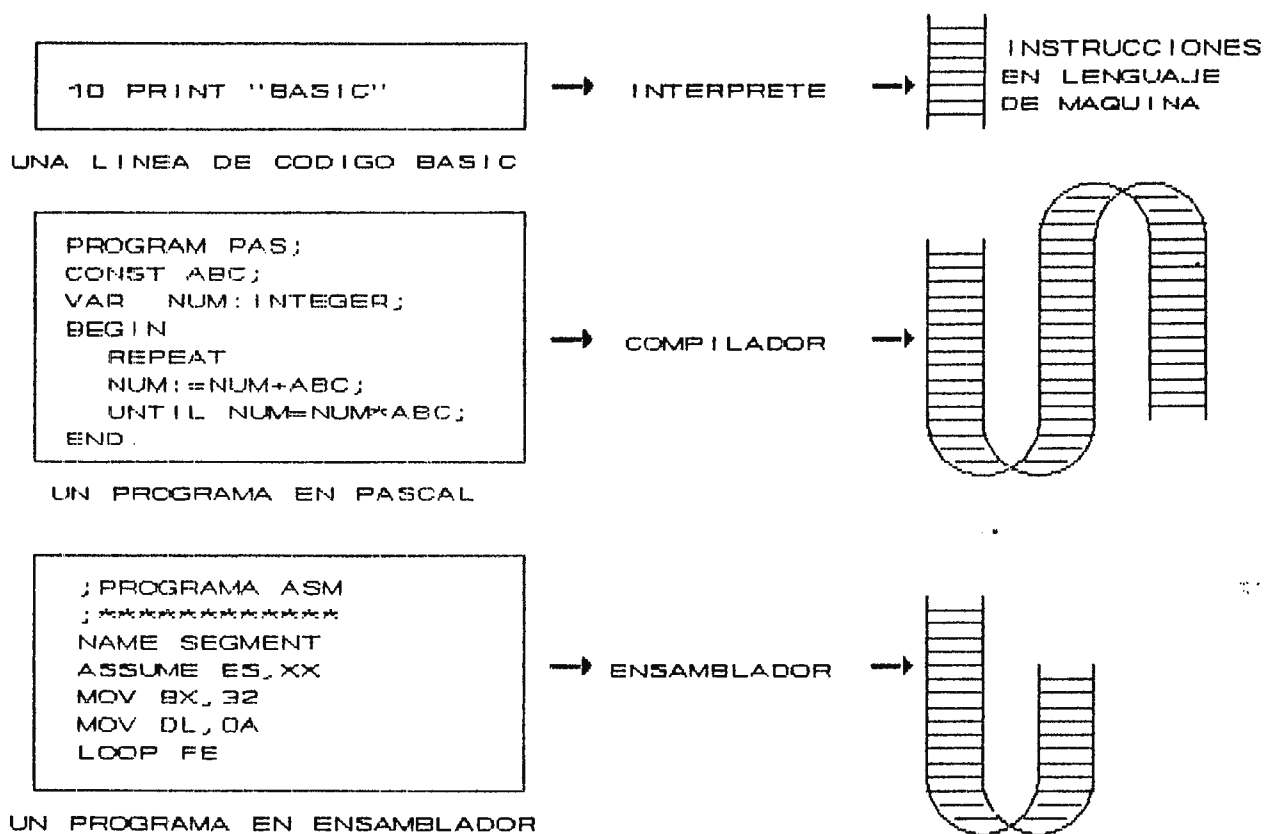


Figura 5.4. Comparación entre los códigos generados por BASIC, PASCAL y ENSAMBLADOR.

5.2.3. Formato del Programa Fuente.

Existen ciertas normas para crear un programa en lenguaje ensamblador. La primera se refiere al formato interno de cada instrucción del lenguaje. Toda línea del código debe digitarse bajo ciertas reglas, las cuales pueden considerarse como la sintaxis propia del lenguaje. Sin embargo, en la formación de instrucciones también entra el juego de la semántica. La segunda norma es el formato externo del programa, que puede equipararse con una capa que rodeará al programa para ayudar a definir su entorno.

5.2.4. El Formato Interno.

El formato interno de una línea de código se puede resumir en cuatro partes. La *etiqueta/variable/constante* (puede definirse o no, y si se define debe estar seguida por separadores, ya sea uno o más espacios o tabuladores). El *nombre mnemónico/directiva*, que puede equipararse con el *verbo* o la *acción* por realizar, seguido de uno o más separadores (espacios o tabuladores). El *operando*, que generalmente se divide en dos (aunque existen instrucciones que sólo operan con uno) el destino (que será el depósito de algún resultado) y la fuente de la información (el originador de "acción"). El destino y la fuente (si existe ésta) deben estar separados por una coma. Finalmente tenemos el *comentario*, que siempre va precedido por el símbolo ";" (debe estar separado del operando por uno o más espacios o tabuladores, y puede existir o no).

5.2.5. Formato Externo.

El formato externo está formado por varios comandos clave que permite establecer el entorno operativo del programa. Entre ellos tenemos uno que define el modelo de memoria (básicamente establece cuáles y cuántos segmentos se van a usar), los diferentes segmentos que contendrá el código, los datos y la pila, así como el comienzo y el final del programa. Ahora bien, existen dos maneras de definir el programa: en la primera se usan *directivas simplificadas* y en la segunda no se usan. Las directivas simplificadas ayudan a esclarecer ciertos comandos críticos del programa, permitiendo que sean más

legibles, y se incluyen en las versiones del *MASM 5.0* y posteriores.

5.2.6. El proceso formal de ensamblaje.

Antes de analizar más a fondo el papel que desempeñan el ensamblador y el enlazador en la creación de programas ejecutables, es importante aclarar algunos conceptos sobre los segmentos y la formación de direcciones.

El uso de segmentos está obligado debido a la arquitectura del 8086. El proceso de formar una dirección involucra un registro de segmento (CS, DS, ES o SS) y un operando que describe el desplazamiento dentro del segmento. Cuando la CPU ejecuta un programa, espera que tanto las instrucciones por ejecutar como los datos se encuentren en sus segmentos respectivos: el código (las instrucciones) en el segmento CS y los datos en el segmento DS. El MASM requiere que toda instrucción o dato se agrupe en bloques o segmentos, con este fin se usan las instrucciones simplificadas `.CODE` y `.DATA` o las no simplificadas `SEGMENT` y `ASSUME`.

Lo que se necesita finalmente es que exista una definición de los segmentos, en la cual se indique el comienzo y el final de cada uno de ellos. Al iniciarse el proceso de ensamblaje, el MASM necesita saber que registro de segmento se usará para procesar determinada instrucción o dato. Cuando se encuentra con la declaración de un segmento, inicializa un contador interno en cero y lo incrementa de acuerdo con la longitud en bytes de cada instrucción o dato. Esto le permite calcular la siguiente dirección o desplazamiento dentro del segmento por acceder. Considere las dos siguientes instrucciones:

```
MOV    AX,2
```

```
ADD    AX,3
```

las cuales se ensamblarían con el siguiente resultado:

```
250F:0100 B80200    MOV    AX,0002
```

```
250F:0103 050300    ADD    AX,0003
```

Recordando que todas las instrucciones se accesan a través del registro CS (registro de segmento de código). La primera instrucción tiene una longitud de tres bytes (B8, 02 y 00), al igual que la segunda (05, 03 y 00). Por lo tanto, al ensamblarse la primera instrucción, el contador interno del segmento de código se incrementa a tres. La segunda instrucción empieza en el desplazamiento de 3 dentro del segmento, y una vez ensamblada el contador interno se incrementa a 6 (la segunda instrucción tiene una longitud de tres bytes). Lo mismo sucede al ensamblar datos; éstos se definen a través de la directiva Db (define byte, que es un solo byte) o Dw (define word, que es una palabra o dos bytes), y se encuentran disponibles a través del registro DS (registro del segmento de datos). Al considerar las dos instrucciones siguientes:

```
Valor_1    DB0
```

```
Valor_2    DW0
```

Al ensamblarse el primer dato (Valor_1), éste tiene una longitud de un byte (DB siempre define un byte). Por lo tanto, el contador

interno del segmento de datos se incrementará a uno. El segundo dato (Valor_2) empezará en el desplazamiento uno dentro del segmento, y al ser ensamblado incrementará el contador interno a tres (DW siempre define una palabra o dos bytes). De hecho, cuando el MASM encuentra referencias a estos símbolos en su programa, los usa como operandos dentro de una instrucción y prepara una tabla de símbolos.

Todo lo anterior suena un poco complejo, pero en realidad no lo es tanto. Simplemente se trata de un proceso secuencial y lógico. No obstante, es esencial para que el usuario comprenda el proceso formal de ensamble.

3. PROGRAMA DE CONTROL DE LA TARJETA.

La elaboración del programa de control de la tarjeta conllevó un proceso de investigación sobre el funcionamiento del lenguaje PASCAL y de la forma en que este puede ser interfazado con pequeñas rutinas escritas en lenguaje ensamblador.

Los procedimientos y funciones escritos en lenguaje ensamblador se enlazan mediante las directivas de compilación \$L. Para ello se requiere que el archivo fuente de ensamblador sea ensamblado en un archivo objeto (.OBJ) utilizando un Macroensamblador. Los programas escritos en forma de archivos de comando (.COM) no son compatibles para tales fines.

Los procedimientos y funciones escritos en lenguaje ensamblador se declaran como *external* en los programas de PASCAL, por ejemplo:

```
function Inverse(num,bits:integer): integer;      external;
```

En el correspondiente archivo fuente de ensamblador, todos los procedimientos y funciones deben aparecer en el segmento de códigos CODE, y los nombres de los procedimientos y funciones deben aparecer en directivas PUBLIC.

Además debe asegurarse que el procedimiento o función escrito en lenguaje ensamblador concuerde con su definición en PASCAL respecto a su modelo de llamado (NEAR o FAR), número de parámetros, tipos de los parámetros, y tipo de los resultados.

Los datos que van a ser intercambiados se definen en el segmento de datos como variables externas con las directivas EXTRN.

En resumen todo esto es posible si se cumplen las siguientes condiciones:

- Todos los procedimientos y funciones deben ser colocados en un segmento denominado CODE y todas las variables privadas deben colocarse en un segmento llamado DATA. Todos los otros segmentos son ignorados así como las directivas GROUP.

- Cuando se declaran variables en el segmento DATA siempre debe usarse un signo de interrogación para especificar el valor:

Count	DW	?		
Buffer	DB	128	DUP	(?)

TurboPascal ignora cualquier requisición de crear variables inicializadas en el segmento DATA.

- Cuando se referencian procedimientos o funciones externas, no se debe especificar un offset. Por ejemplo, la siguiente construcción no

es permitida:

```

        EXTRN    MyProc    : NEAR
        EXTRN    MyProc + 8

```

Hay que notar que la restricción no se aplica a variables EXTRN.

El programa de control utiliza un procedimiento externo y una función externa. El procedimiento externo es llamado FDA. Se encarga de configurar y manipular la tarjeta adquisitora para tomar un muestreo de una señal analógica seleccionando el reloj de muestreo a utilizar de acuerdo con la escala elegida.

Dentro de dicho procedimiento se pueden identificar tres partes esenciales:

- Reconocimiento de la escala.
- Muestreo de la señal.
- Transferencia de resultados a la PC.

El reconocimiento de la escala es una etapa de identificación de cual de las escalas ha sido seleccionada mediante la lectura de los bits C1, C2 y C3 de la PPI, los cuales se hallan conectados a un selector el cual pone un bajo en el pin correspondiente con la escala en cuestión.

La parte de muestreo de señal se encarga de habilitar el proceso de adquisición de datos dentro de la etapa de memoria auxiliar de la tarjeta. Este proceso se habilita durante un período de tiempo suficiente como para llenar la memoria auxiliar.

Una vez que el tiempo de adquisición ha expirado, FDA retoma el control inhabilitando el proceso de muestreo y comenzando a leer los datos almacenados en la memoria de la tarjeta.

Los datos leídos son almacenados en una tabla de variables de tipo

entero que son manejadas mediante un vector de posición. Una vez finalizada la transferencia de información, FDA entrega el control al programa principal.

El flujograma de dicha rutina se muestra en la Figura 5.5.

La función INV no es más que una rutina para invertir el orden de los bits de un byte suministrado. La aplicación de esta rutina se da en la realización del algoritmo de la TRF para poder llevar un control de los vectores de datos a la hora de implementar las mariposas.

La función toma dos parámetros: N que es un byte de datos del tipo entero y B que es el número de bits del byte N, también del tipo entero. Luego mediante un juego de instrucciones de rotación hacia la izquierda y derecha, y empleando el bit de acarreo como memoria intermedia, se revierte el orden de los bits obteniendo así el vector correspondiente al dato a utilizar en la mariposa.

El flujograma de la función INV se muestra en la Figura 5.6.

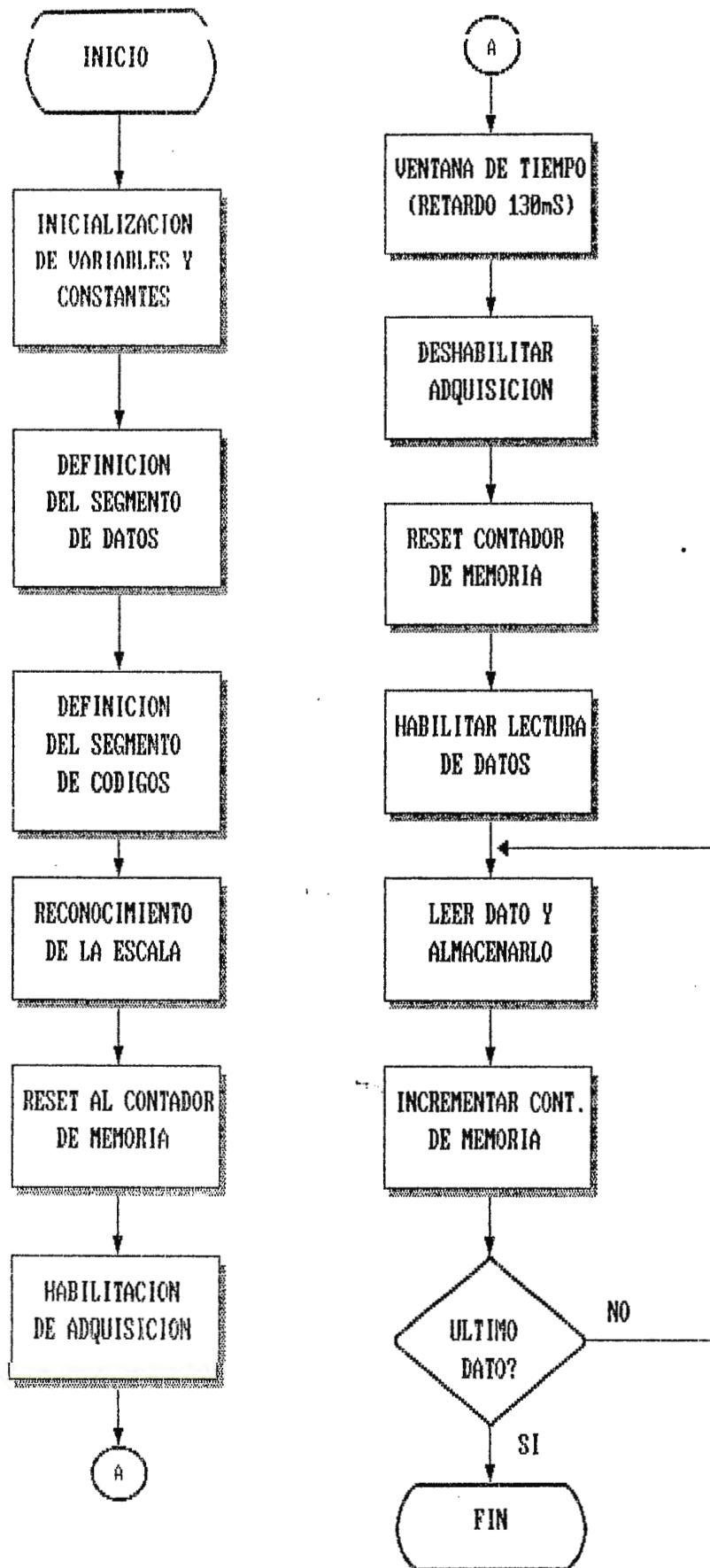


Figura 5.5. Flujoograma de la rutina FDA.ASM

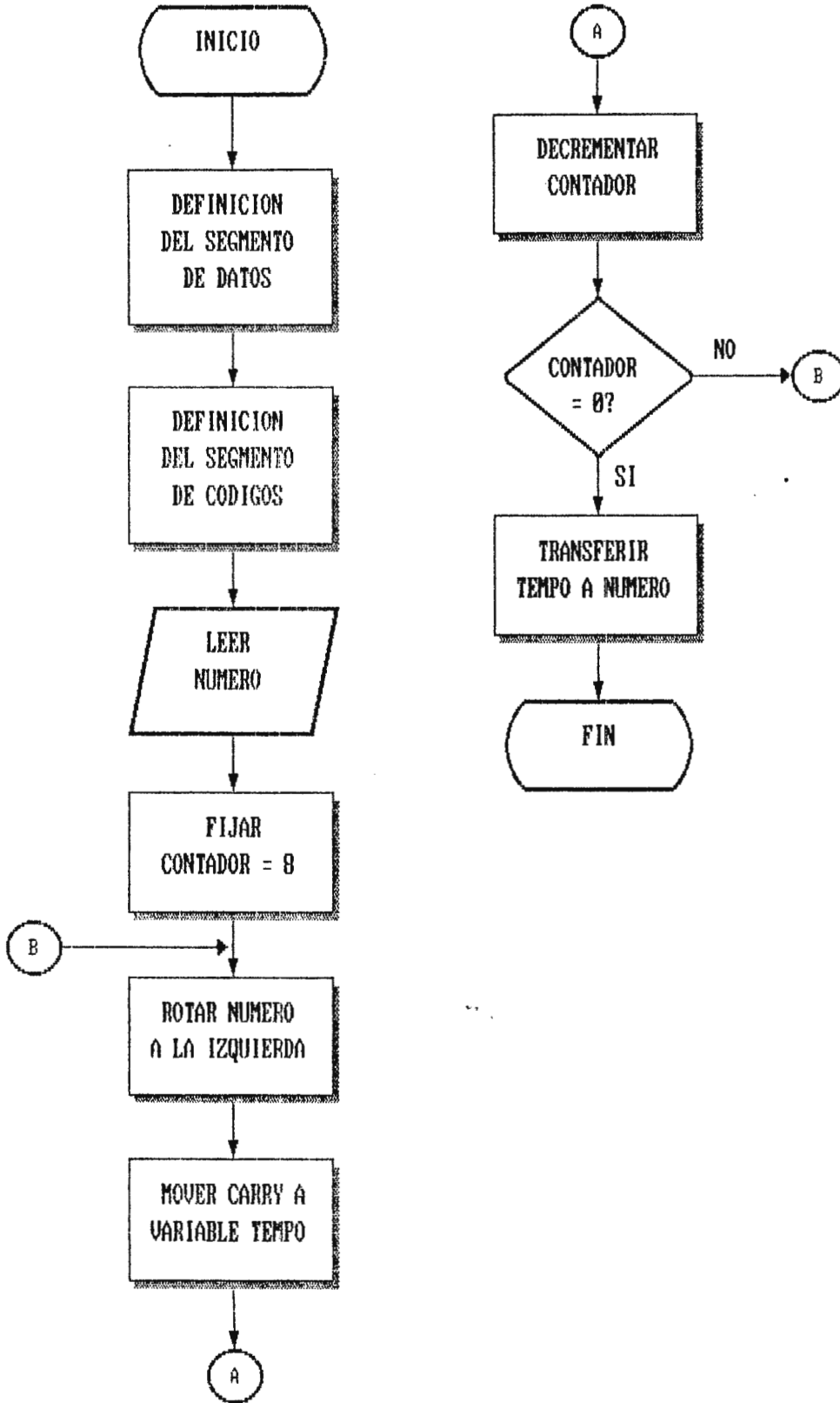


Figura 5.6. Flujoograma de función INV.ASH

El funcionamiento del programa principal está basado en la detección de teclas clave y la interpretación de su significado de acuerdo con el menú en que el usuario se encontrara.

El programa inicia con una pantalla de presentación (Figura 5.7), la cual hace alusión al nombre de la tesis así como los nombres de los integrantes del grupo. Para su realización se emplearon rutinas contenidas dentro de las unidades GRAPH y CRT de TURBOPASCAL únicamente adaptándolas a las necesidades de la pantalla.

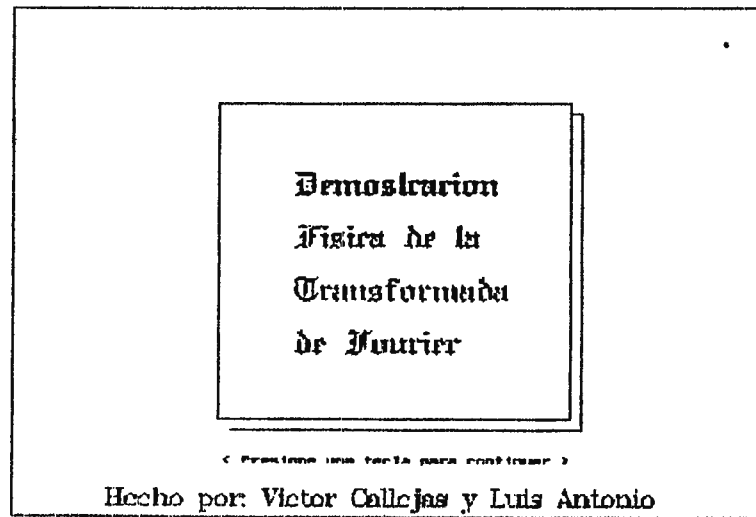


Figura 5.7. Pantalla de presentación.

Luego de un período de 3 segundos el programa pide presionar una tecla para continuar. Al presionarse la tecla se cae en el menú principal(Figura 5.8), el cual es en forma de una barra de comandos con los nombres de todos los comandos permisibles en la pantalla. La barra aparece en la parte superior de dicha pantalla, cuyos comandos se accesan tecleando la letra correspondiente que aparece en color rojo, diferenciandose de las demás que se encuentran en color gris.

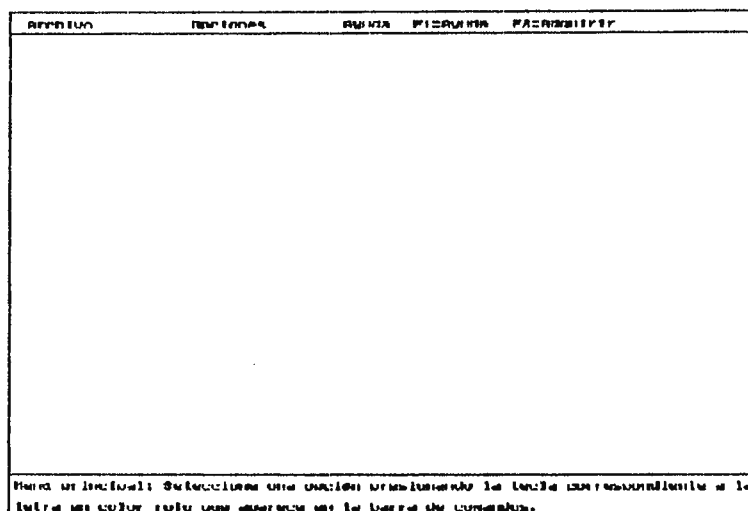


Figura 5.8. Pantalla del menú principal.

Existen 5 posibilidades dentro del menú principal:

- Accesar el menú de archivos.
- Accesar el menú de opciones.
- Accesar el menú de ayuda.
- Accesar la ayuda rápida.
- Iniciar el proceso con los valores por defecto.

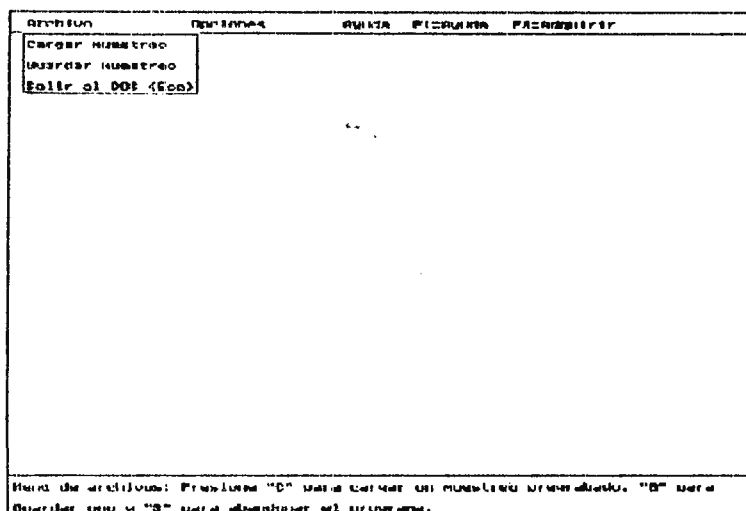


Figura 5.9. Pantalla del menú de archivos.

El menú de Archivos (Figura 5.9) incluye los comandos relacionados con la carga y el almacenaje de muestreos adquiridos con el programa, en y desde, archivos de datos de extensión .DAT. Además en este menú está la opción de abandonar el programa y retornar al DOS.

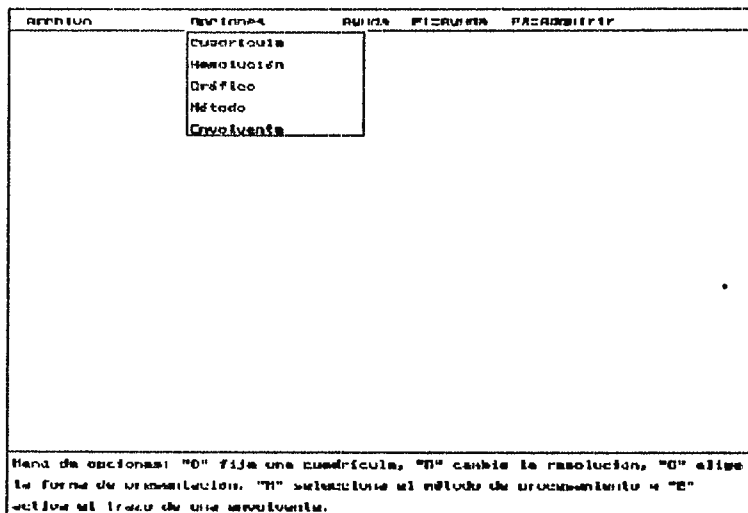


Figura 5.10. Pantalla del menú de opciones.

El menú de opciones contiene comandos que modifican los parámetros de funcionamiento del programa. Los parámetros accesibles a modificaciones son:

- Cuadrícula: Provee al usuario de una ayuda visual para realizar mediciones al dibujar una cuadrícula de 4x8 líneas en las gráficas en tiempo y frecuencia de la señal.

- Acercamiento: Sirve para seleccionar el número de puntos que se desea aparezcan graficados en las ventanas del tiempo y de la frecuencia. Esto no implica que se esté variando el número de muestras a tomar, sino simplemente el número de ellas que va a graficarse en pantalla. Los números de datos a graficar ya están establecidos y son 32, 64 y 128.

- Visualización: Elige el modo de visualización de las señales. Hay tres opciones disponibles; Discreto implica que se va a trazar una línea vertical desde la base del eje X hasta el valor de cada muestra. Continuo permite trazar una línea desde una muestra a la siguiente simulando la envolvente del gráfico. Ambas sirven entonces para poder emplear ambas técnicas a la vez.

- Método: Elige el algoritmo por defecto a utilizar en el proceso de obtención del espectro. Las opciones son aplicar la TRF o la TDF.

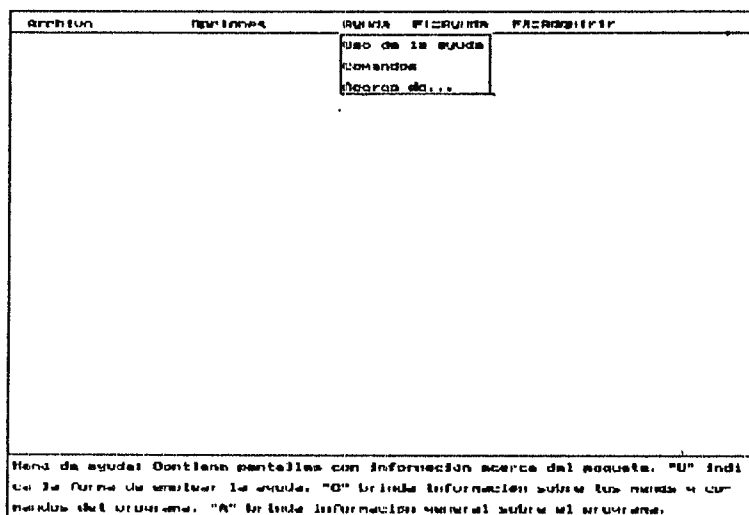


Figura 5.11. Pantalla del menú de ayuda.

El menú de Ayuda contiene los comandos que evocan cierta información de ayuda al usuario. Posee tres comandos:

- Uso de la ayuda. Donde se explica la forma en que se debe utilizar los menús de ayuda.

- Comandos. Despliega información acerca de los diferentes comandos del programa. Tiene la misma estructura que el programa principal pero con la diferencia que la información se presenta en un recuadro que aparece al centro de la pantalla. Para avanzar en los tópicos de la ayuda basta con presionar las letras de color verde

correspondientes que aparecen al interior del recuadro.

- Acerca de... Es simplemente un recuadro donde se resumen algunos datos de tipo general sobre el paquete así como sus requerimientos y limitaciones, tanto en software como en hardware.

El menú de ayuda rápida no es más que entrar de forma directa dentro de la ayuda de comandos simplemente con presionar la tecla <F1>.

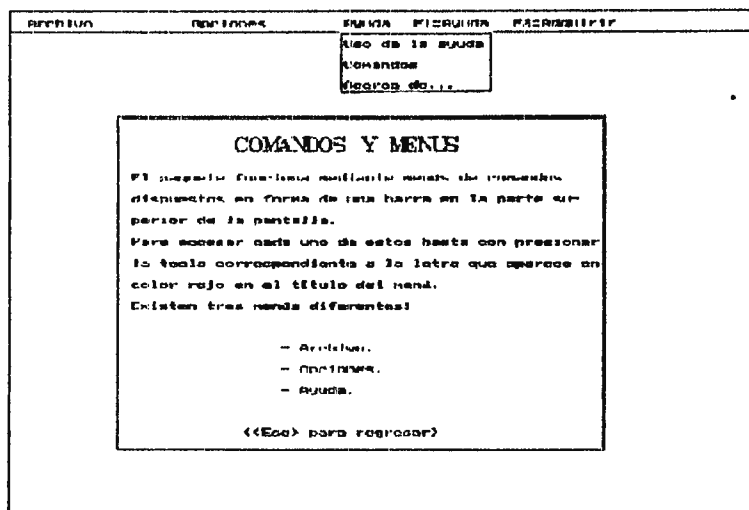


Figura 5.12. Pantalla de ayuda sobre los comandos.

La tecla <F2> inicia el proceso de adquisición y procesamiento de datos limpiando la pantalla y dibujando en la parte superior de ésta el gráfico correspondiente a los datos en el dominio del tiempo. Una vez que los datos han sido procesados, el programa presenta el gráfico en el dominio del tiempo en la parte inferior de la pantalla. Las teclas de cursor <+> y <-> sirven para desplazar el gráfico correspondiente al dominio de la frecuencia, lo cual le permite al usuario poder movilizarse a través del espectro.

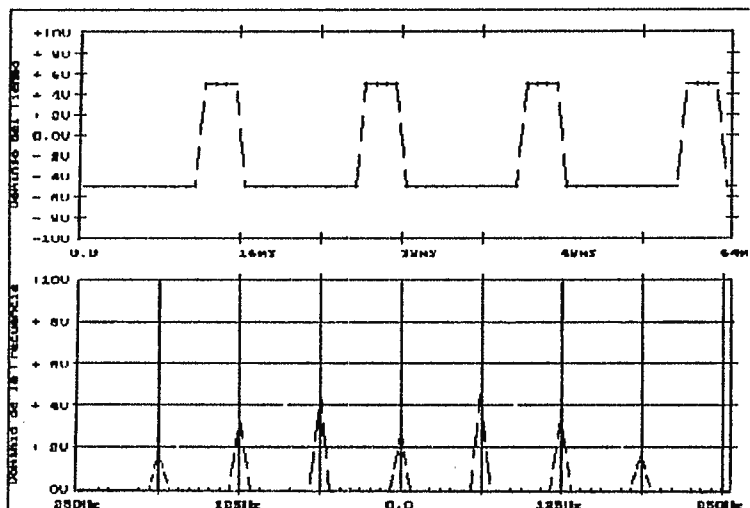


Figura 5.13. Pantalla de presentación de gráficos en el dominio del tiempo y la frecuencia.

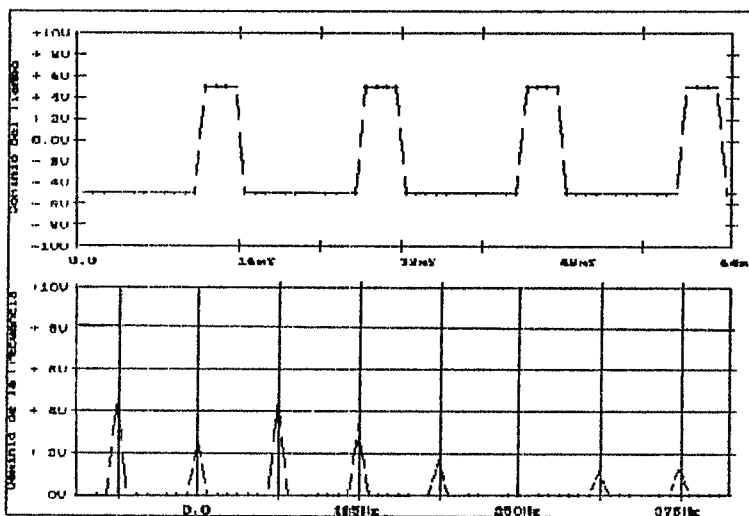


Figura 5.14. Espectro de frecuencia desplazado mediante los cursores.

Con todo esto, el programa de control de la tarjeta se queda corto en cuanto a presentación del mismo (los menús no son accesibles mediante la tecla <ALT>, ni puede avanzarse de menú en menú con los cursores, etc.); sin embargo, estos aspectos quedan relegados a un segundo plano si se ve la funcionalidad de la tarjeta en cuanto a tener la capacidad de servir no solo como un equipo para la demostración física de Fourier sino que ofrece la oportunidad de experimentar con algoritmos de procesamiento de señales tales como la TDF y la TRF.

CONCLUSIONES Y RECOMENDACIONES

Luego de llevar a término el trabajo de graduación se presentan los resultados obtenidos por el grupo, los cuales se ponen de manifiesto en las siguientes conclusiones:

- El procesamiento de señales resulta actualmente una herramienta muy útil para el Ingeniero, ya que la tecnología nos lleva cada vez más a la interacción del hombre con la computadora.
- La implementación de trabajos que involucren un procesamiento de señales no es privativa de grandes consorcios ni corporaciones, sino que por el contrario está accesible para aquellos investigadores que vean en el procesamiento de datos una solución práctica para el estudio y la solución de problemas.
- El algoritmo de la TDF implica un proceso de convolución entre una secuencia de muestras obtenidas a partir de una función continua en el tiempo (señal de entrada) y un impulso unitario de Dirac, lo cual no es más que la operación de cada elemento de la secuencia de entrada contra el impulso unitario de Dirac desplazado en fase a lo largo de todo un período de 0 a 2π .
- La TRF es un algoritmo que, basado en la redundancia de operaciones de cálculo producida por la TDF, permite obtener un aproximado al espectro de frecuencia discreto de la señal de entrada, con un menor número de operaciones, y por consiguiente, en un tiempo menor.

- El diseño de una tarjeta adquirentora de datos resulta de mucha utilidad para que el estudiante de último año pueda ampliar su panorama en cuanto a aplicaciones prácticas que pueden implementarse nacionalmente, sin tener obligadamente que depender o recurrir a productos y Cías. extranjeras.
- El uso de una memoria auxiliar o intermedia para almacenar los datos e independizar con ello el reloj de adquisición de la velocidad de procesamiento de la máquina, permite hacer compatible la tarjeta con diferentes tipos de PC, no importando la velocidad de procesamiento de éstas.
- No obstante las ventajas de la TRF en cuanto a menor tiempo de procesamiento y número de operaciones de cálculo, cabe aclarar que para obtener resultados satisfactorios es necesario cumplir con que el número de ciclos muestreados de la señal analógica debe ser igual a una potencia de 2.
- El problema entonces de llevar un control del número de períodos muestreados puede solucionarse rellenando el excedente de períodos muestreados con valores 0. Esto permite obtener una aproximación aceptable del espectro de frecuencia discreto correspondiente a la señal de entrada, aunque con un leve incremento en el grado de error.
- El máximo número de componentes espectrales que puede obtenerse corresponde al número de muestras con que se haya realizado el algoritmo (indiferente si se utiliza la TDF o la TRF).

- Las muestras obtenidas deben ser ajustadas mediante la adición de un nivel "offset" para eliminar las cantidades negativas, ya que el algoritmo de la TRF no funciona como lo esperado al manejar datos y cifras negativas.

- El grado de confiabilidad de los resultados de la TRF comparados con los de la TDF es óptimo en el caso de señales del tipo cuadradas, perdiéndose precisión con las señales de tipo senoidal y triangular.

- La resolución del espectro depende en gran medida del número de muestras que se hayan analizado.

Luego de exponer nuestras conclusiones, presentamos a continuación algunas recomendaciones para la implementación de sistemas similares en un futuro.

- Cuando se va a experimentar por primera vez con cualquier procedimiento o teoría relativamente nueva y desconocida, es recomendable avanzar en pasos cortos, plantear pequeñas metas sobre las cuales poder luego basarse para la implementación total del sistema. El lema "divide y vencerás" se aplica en este caso al lograr solventar las situaciones y problemas que se dan en el desarrollo de un proyecto de manera individual y no en conjunto. Así, es recomendable realizar experimentos con un prediseño (modelo a escala) del sistema a implementar.

- Cuando se trabaja con los puertos de entrada/salida, es necesario verificar en todo momento el fiel cumplimiento de los estándares de los conectores, ya que un ligero desvío en las medidas puede conducir a graves consecuencias.

- Las variaciones de frecuencia en el muestreo causan errores en la distribución de los armónicos en el espectro discreto, ya sea adelantándolos o atrasándolos una posición relativa al lugar donde le hubiese correspondido estar.

- Este diseño tiene poca resolución, ya que se implementó con un convertidor de 8 bits y con una memoria auxiliar de 256 bytes. Si se desea hacer un análisis espectral más extenso y riguroso, debería incrementarse la resolución de la etapa de conversión, así como incrementar el número de muestras a tomar, por consiguiente, elevar la capacidad de la memoria auxiliar.

- Una mejoría que puede hacerse de éste proyecto consiste en agregar un control (por software) de la cantidad de ciclos muestreados para así ajustar el número de períodos muestreados a una potencia exacta de 2 mediante la adición de ceros, y así explotar la velocidad de procesamiento de la TRF.

- Si se logra el punto anterior, podría pensarse en realizar un análisis espectral con datos en línea de manera alternada: mientras se está procesando un bloque de datos, otro se está adquiriendo en esos momentos.

- Otra mejora que puede hacerse es incluir dentro del módulo externo o de la tarjeta misma un generador de señales, para tener un equipo didáctico más completo.
- La tarjeta puede utilizarse también para el análisis de señales de baja frecuencia mediante la adición de una nueva escala y un reloj de muestreo más lento.
- Otras aplicaciones prácticas a las que podría extenderse este equipo, además de ser un equipo de carácter didáctico, podrían ser: un análisis espectral de señales en telecomunicaciones; un equipo de monitoreo de señales biomédicas; Implementación de un filtro selectivo por computadora.
- La otra opción que se había planteado al inicio del proyecto de realizar un análisis espectral mediante la implementación de filtros pasabanda digitales capaces de barrer un rango del espectro en busca de niveles de potencia queda abierta para un posible nuevo proyecto de graduación.
- Al momento de realizar nuestro trabajo de graduación, el tema del procesamiento digital de señales era desconocido para nosotros; no obstante, consideramos que por las tendencias que lleva el avance tecnológico de nuestra sociedad resulta de gran utilidad del Ingeniero. Por tanto, debería incluirse como una materia electiva técnica que trate acerca del procesamiento digital de señales en el

plan de estudios de las carreras de Ingeniería (para el caso de la UDB se recomendaría en Electrónica y Biomédica).

- Además debería de fomentarse más el aprendizaje de al menos un lenguaje de programación de bajo nivel y otro de alto nivel entre los estudiantes de las carreras antes mencionadas, pero desde una perspectiva orientada hacia el manejo de interfaces y puertos de E/S.

REFERENCIAS BIBLIOGRAFICAS

- [1] Chen, Chi-Tsong. SYSTEM AND SIGNAL ANALYSIS. New York :
Sonders Publishing, 1989.
- [2] "Fourier, Barón de". (En: DICCIONARIO ENCICLOPEDICO
U.T.E.H.A., T. V., 1953), PP. 216.
- [3] Kuc, Roman. INTRODUCTION TO DIGITAL SIGNAL PROCESSING. United
States of America: McGraw-Hill, 1988.
- [4] Hyatt, William. C. Análisis de Circuitos en Ingeniería. Buenos
Aires : McGraw-Hill, 1978.
- [5] P. Hsu, Hwei. ANALISIS DE FOURIER. Bogotá : ADDISON-WESLEY
IBEROAMERICANA, 1987.
- [6] Poularikas, Alexander y Seely, Samuel. SIGNALS AND SYSTEMS
Boston: PWS Publishers, 1985.
- [7] Margulis, Neal. i860 Microprocessor Architecture. United
States of America: Osborne McGraw-Hill, 1990.
- [8] Williams, Arthur B., CIRCUITOS LOGICOS Y DE CONVERSION DE A/D
Y D/A., tr. por Alexis Méndez Chamorro: McGrawHill,
México, 1989.
- [9] National Semiconductors. Linear Databook.: National
Semiconductors, Edition 1988.

- [10] By the staff of Lab-Volt (Québec) Ltd., INTERFAZ DE ENTRADA/SALIDA DE MICROPROCESADORES. CONCEPTOS Y APLICACIONES. Québec : Lab-Volt, 1988.
- [11] Intel Corporation, 386sx Microprocessor. PROGRAMMER'S REFERENCE MANUAL. Osborne McGrawHill, 1989.
- [12] Zaks, Rodney. PROGRAMACION EN TURBO PASCAL, Ediciones ANAYA Multimedia S.A., Madrid, 1986.
- [13] Rojas P., Alberto, ENSAMBLADOR BASICO Computec Editores S.A. de C.V., Mexico D.F. 1993.
- [14] Espinoza Q., Damaris, Cómo hacer referencias bibliográficas y notas de pie de página. San José, C. R. : Instituto Centroamericano de Administración Pública - ICAP, 1989.

ANEXO A. PROPIEDADES DE LA TRANSFORMADA DE FOURIER

Tabla 2.1. Pares de transformadas de Fourier.

$g(t)$	$G(\omega)$
1. $e^{-at}u(t)$	$\frac{1}{a + j\omega}$
2. $te^{-at}u(t)$	$\frac{1}{(a + j\omega)^2}$
3. $ t $	$\frac{-2}{\omega^2}$
4. $\delta(t)$	1
5. 1	$2\pi \delta(\omega)$
6. $u(t)$	$\pi \delta(\omega) + \frac{1}{j\omega}$
7. $\cos \omega_0 t$	$\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$
8. $\text{sen } \omega_0 t$	$j\pi[\delta(\omega + \omega_0) - \delta(\omega - \omega_0)]$
9. $\cos \omega_0 t u(t)$	$\frac{\pi}{2}[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)] + \frac{j\omega}{\omega_0^2 - \omega^2}$
10. $\text{sen } \omega_0 t u(t)$	$\frac{\pi}{2j}[\delta(\omega - \omega_0) - \delta(\omega + \omega_0)] + \frac{\omega_0}{\omega_0^2 - \omega^2}$
11. $e^{-at} \text{sen } \omega_0 t u(t)$	$\frac{\omega_0}{(a + j\omega)^2 + \omega_0^2}$
12. $2B \text{ sinc}(2Bt)$	$\Pi\left(\frac{\omega}{4\pi B}\right)$
13. $\Pi\left(\frac{t}{\tau}\right)$	$\tau \text{ sinc}\left(\frac{\omega\tau}{2\pi}\right)$
14. $\begin{matrix} 1 - \frac{ t }{\tau} \dots t < \tau \\ 0 \dots t > \tau \end{matrix}$	$\tau \text{ sinc}^2\left(\frac{\omega\tau}{2\pi}\right)$
15. $e^{-a t }$	$\frac{2a}{a^2 + \omega^2}$
16. $\sum_{n=-\infty}^{\infty} \delta(t - kT)$	$\omega_0 \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0) \quad \omega_0 = \frac{2\pi}{T}$
17. $e^{-t^2/2\sigma^2}$	$\sigma\sqrt{2\pi} e^{-\sigma^2\omega^2/2}$

ANEXO B. PROPIEDADES DE LA TRANSFORMADA DISCRETA DE FOURIER

TABLE 4.1
Properties of the discrete Fourier transform

1. Definition: For a finite-duration sequence $h(n)$, for $0 \leq n \leq N-1$,

$$H(k) = \sum_{n=0}^{N-1} h(n) e^{-j2\pi kn/N} \quad \text{for } 0 \leq k \leq N-1 \quad (\text{DFT})$$

and

$$h(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k) e^{j2\pi kn/N} \quad \text{for } 0 \leq n \leq N-1 \quad (\text{IDFT})$$

2. Periodic extension:

$$\tilde{h}(n) = \sum_{m=-\infty}^{\infty} h(n + mN)$$

3. Linearity: If $\{x(n)\} = \{h_1(n)\} + \{h_2(n)\}$, then $X(k) = H_1(k) + H_2(k)$.

4. Periodicity: $H(k) = H(k + N)$.

5. Magnitude and phase functions: If $H(k) = H_R(k) + jH_I(k)$, then

$$\begin{aligned} |H(k)|^2 &= H_R^2(k) + H_I^2(k) \\ \text{Arg}[H(k)] &= \arctan[H_I(k)/H_R(k)] \end{aligned}$$

6. Fourier transform of delayed sequence:

$$\text{if } \{y(n)\} = \{x(n - n_0)\} \quad \text{then } Y(k) = X(k) e^{-j2\pi kn_0/N}$$

7. Fourier transform of the convolution of two sequences:

$$\text{if } \{y(n)\} = \{h(n)\} * \{x(n)\} \quad \text{then } Y(k) = H(k)X(k)$$

A linear convolution of $\{h(n)\}$ and $\{x(n)\}$ results when N_y -point DFT sequences are computed for $\{h(n)\}$ and $\{x(n)\}$, where N_y is the duration of $\{y(n)\}$. Otherwise, a circular convolution of $\{h(n)\}$ and $\{x(n)\}$ results.

8. Fourier transform of the product of two sequences:

$$\text{if } \{y(n)\} = \{h(n)x(n)\} \quad \text{then } Y(k) = H(k) \odot X(k) \quad (\text{circular convolution})$$

9. For the real-valued sequence $h(n)$, for $0 \leq n \leq N-1$:

$$H_R(k) = \sum_{n=0}^{N-1} h(n) \cos(2\pi kn/N), \quad \text{for } 0 \leq k \leq N-1$$

$$H_I(k) = -\sum_{n=0}^{N-1} h(n) \sin(2\pi kn/N) \quad \text{for } 0 \leq k \leq N-1$$

- (a) Complex-conjugate symmetry: $H(k) = H^*(N-k)$
 (b) Real component is even function: $H_R(k) = H_R(N-k)$
 (c) Imaginary component odd function: $H_I(k) = -H_I(N-k)$
 (d) Magnitude function is even function: $|H(k)| = |H(N-k)|$
 (e) Phase function is odd function: $\text{Arg}[H(k)] = -\text{Arg}[H(N-k)]$
 (f) If $\{h(n)\} = \{h(-n)\}$ (even sequence), then $H(k)$ is purely real.
 (g) If $\{h(n)\} = \{-h(-n)\}$ (odd sequence), then $H(k)$ is purely imaginary.

DIAGRAMA MODULO EXTERNO

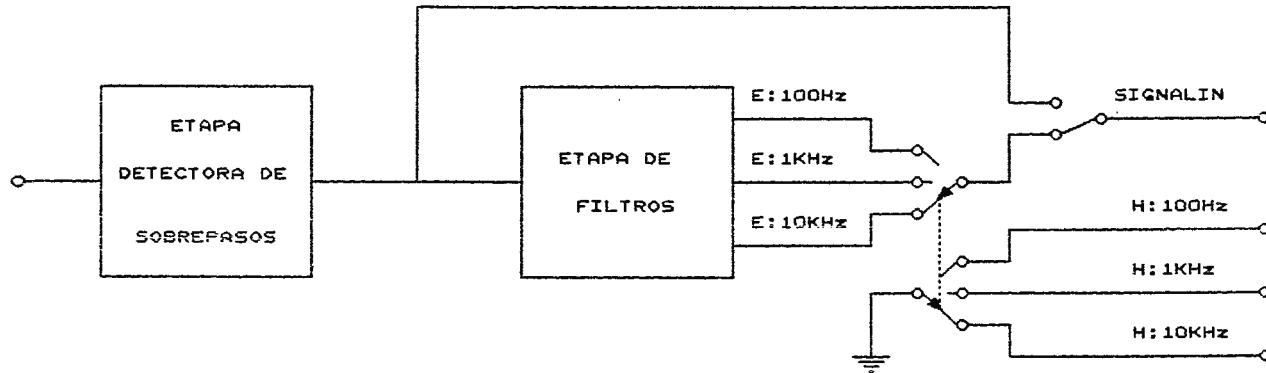
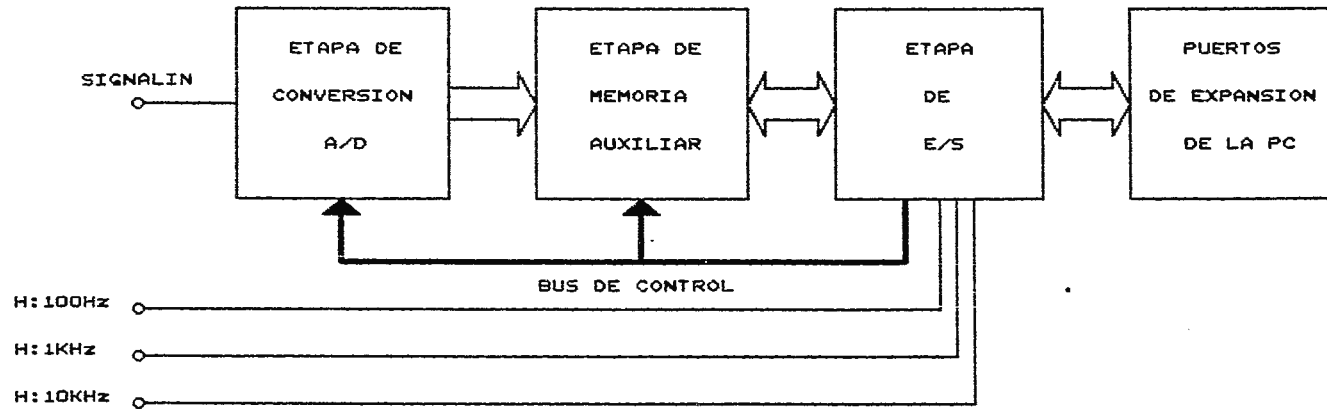
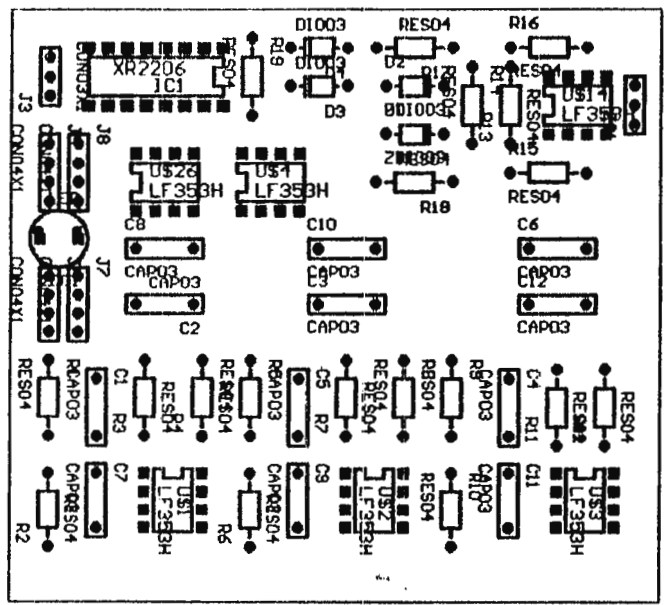


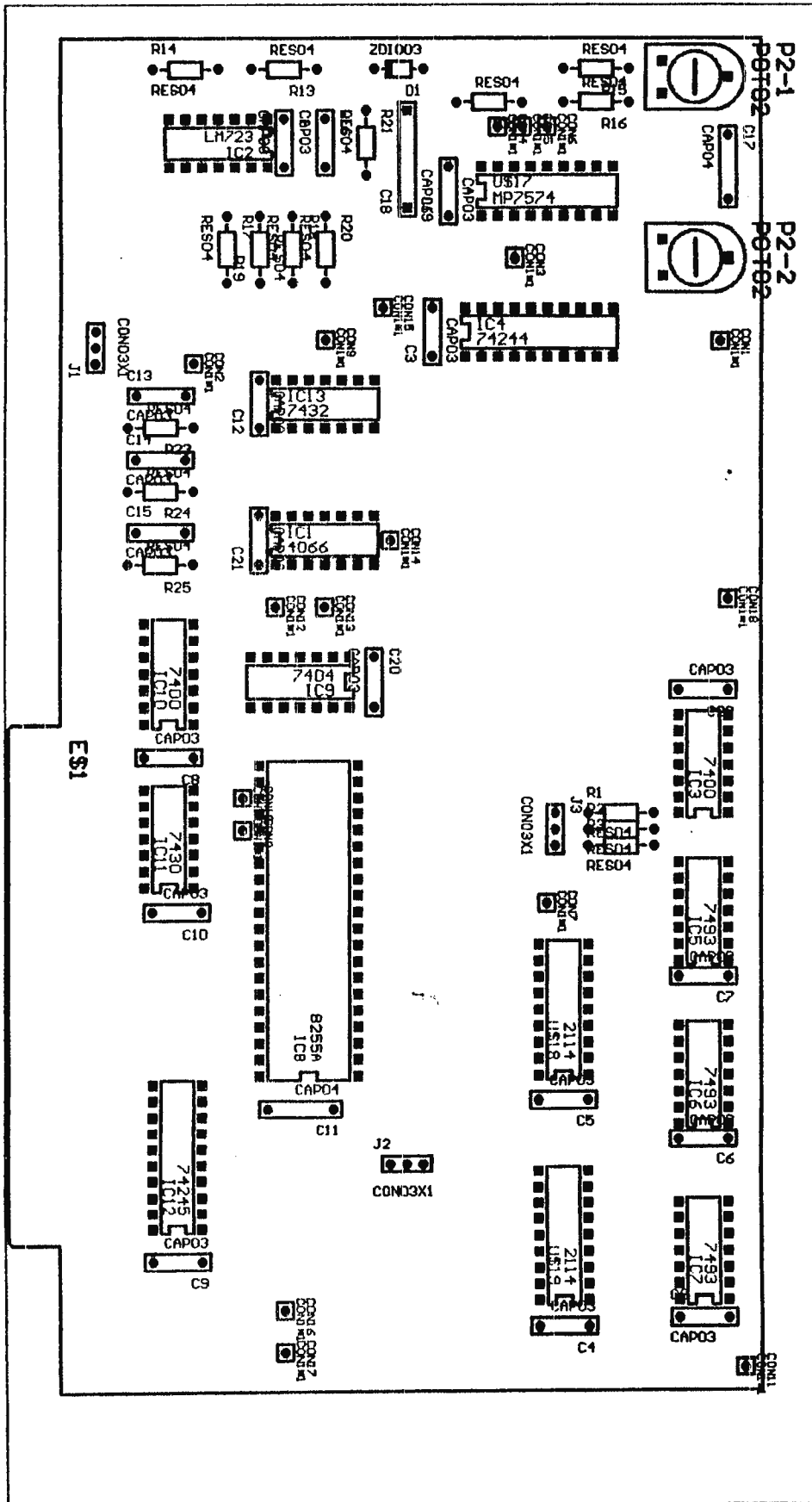
DIAGRAMA TARJETA ADQUISITORA



ANEXO D. DIAGRAMAS PICTORICOS DEL CIRCUITO

TARJETA DEL MODULO EXTERNO

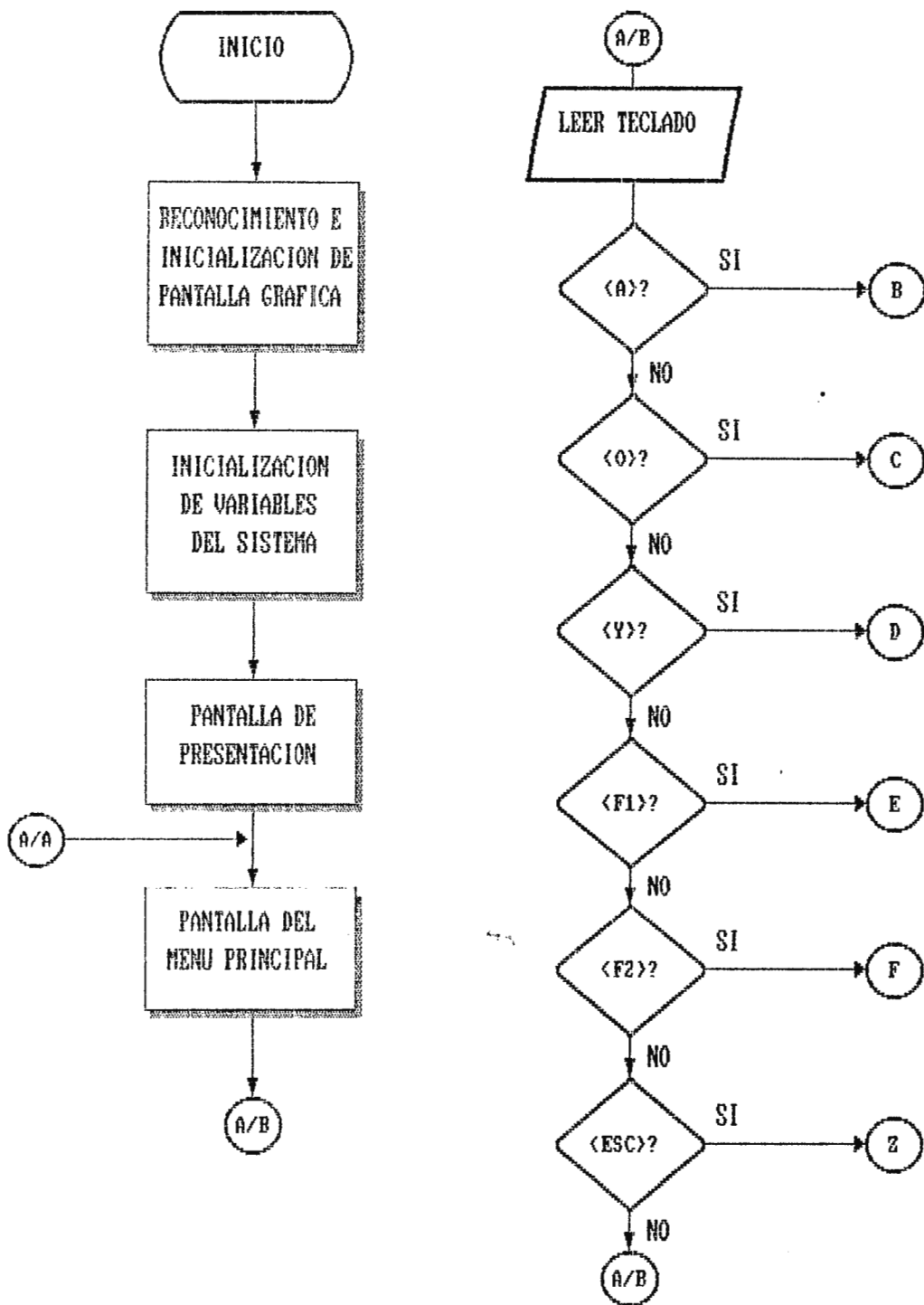




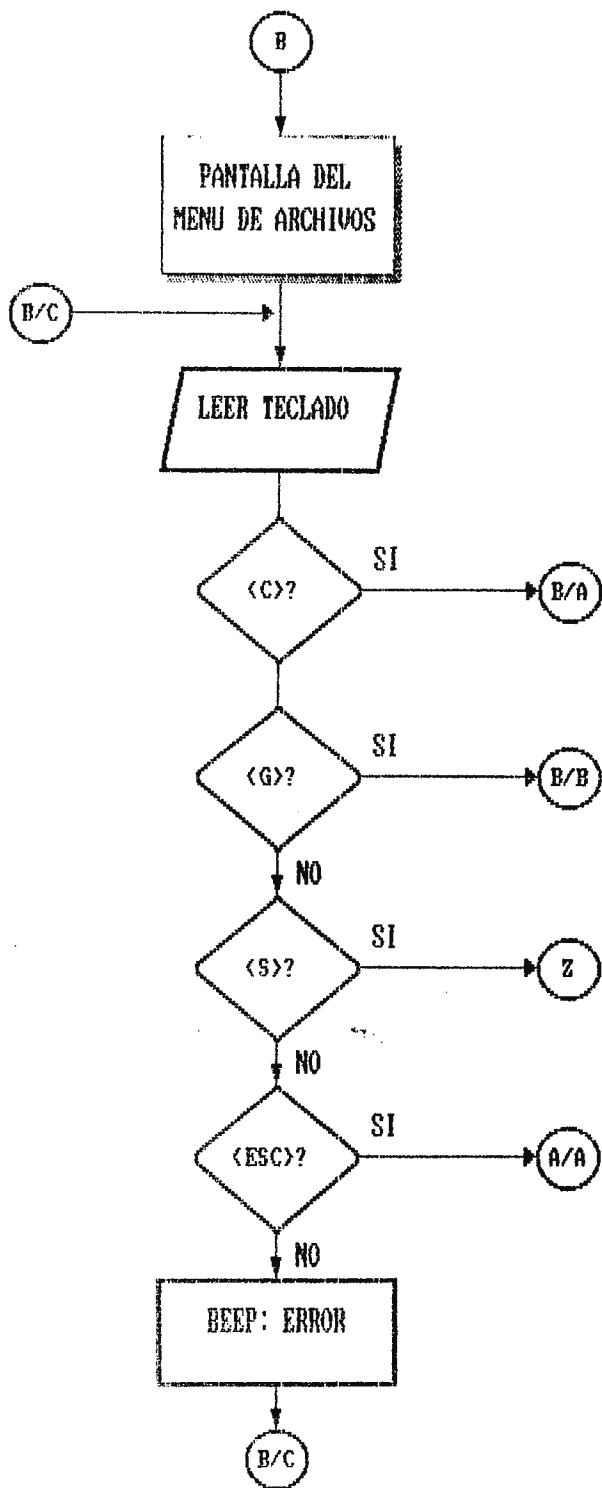
FLUJOGRAMA DEL PROGRAMA PRINCIPAL

E.1

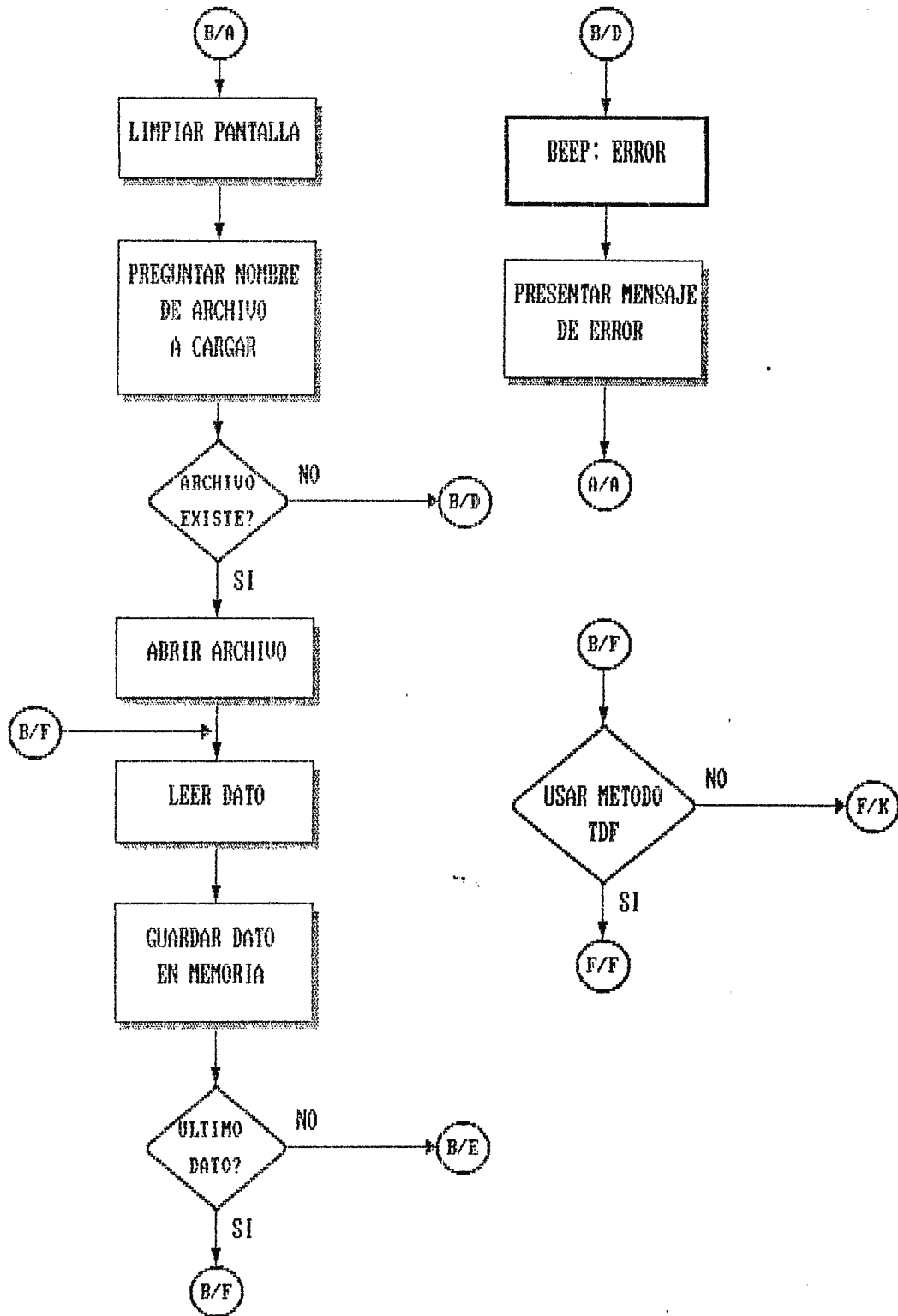
ANEXO E. FLUJOGRAMA DEL PROGRAMA PRINCIPAL



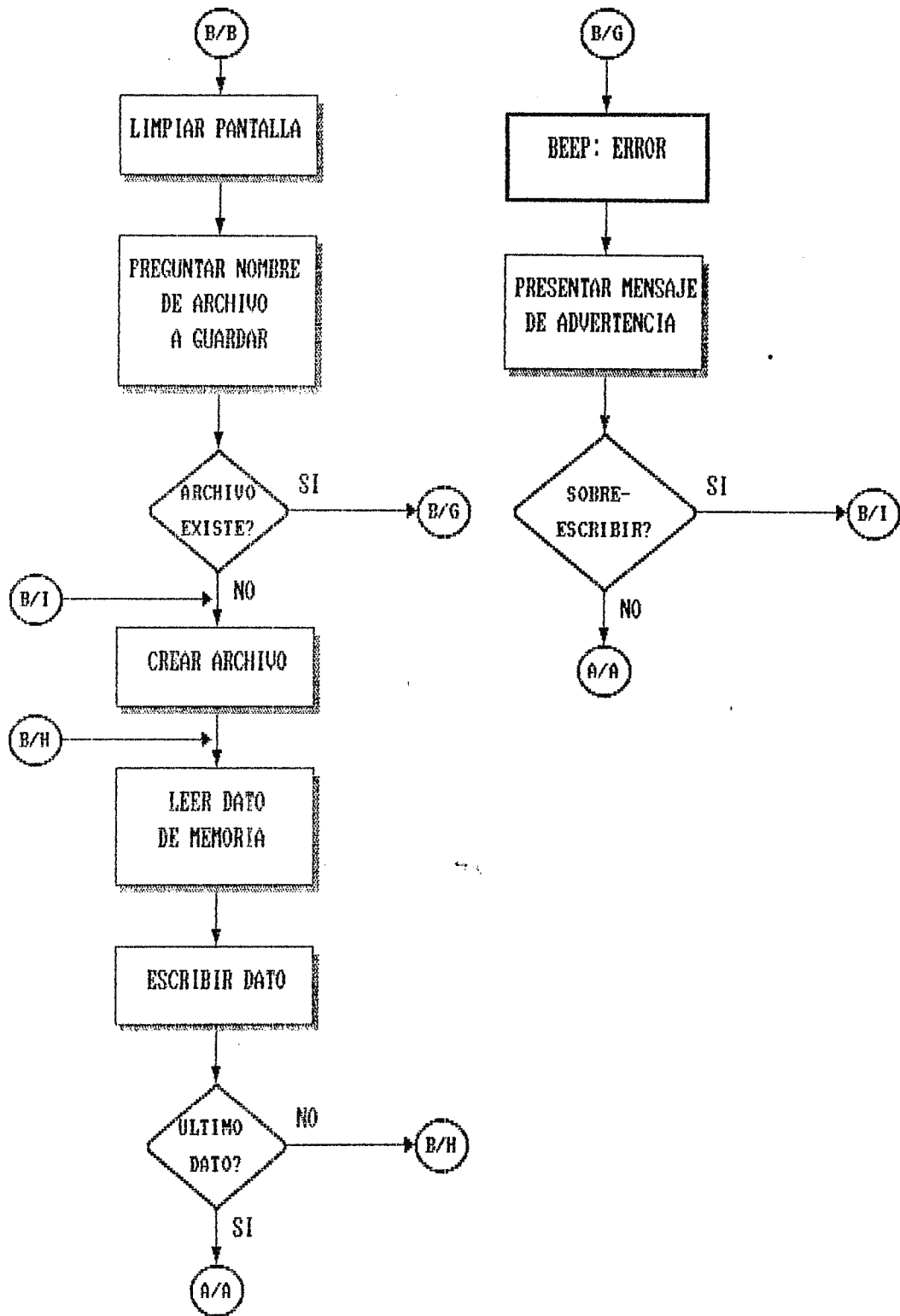
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)

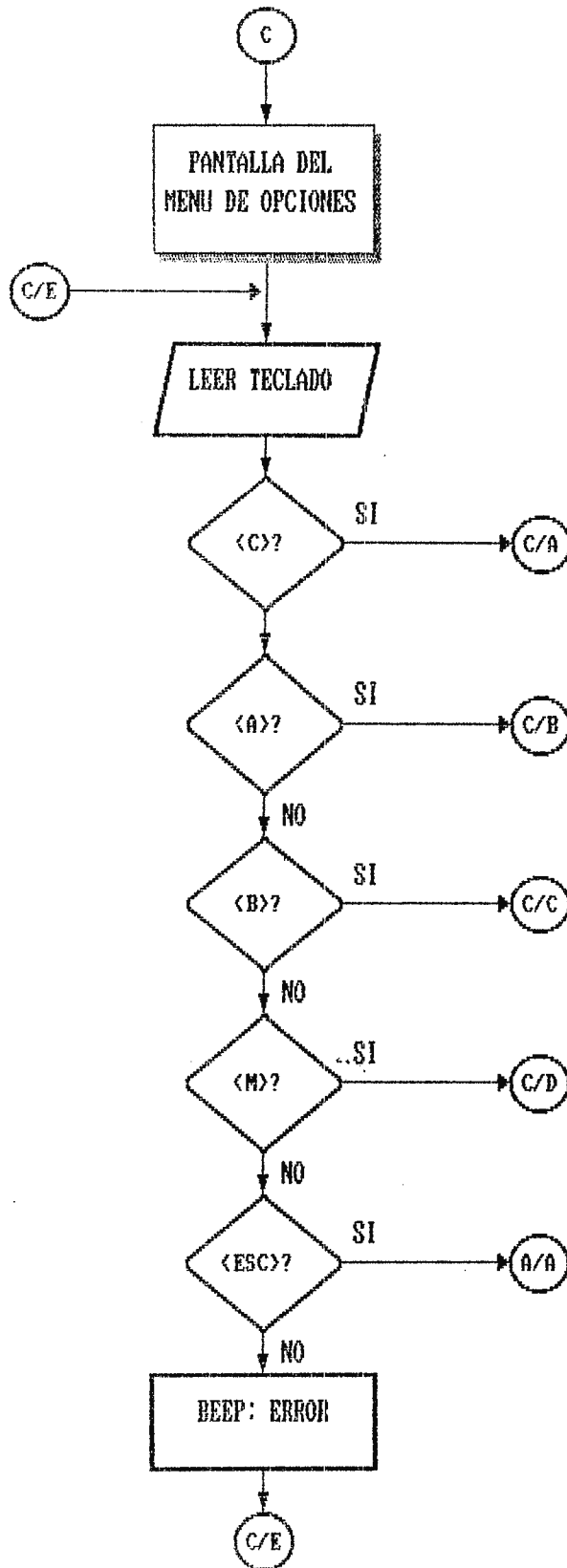


FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)

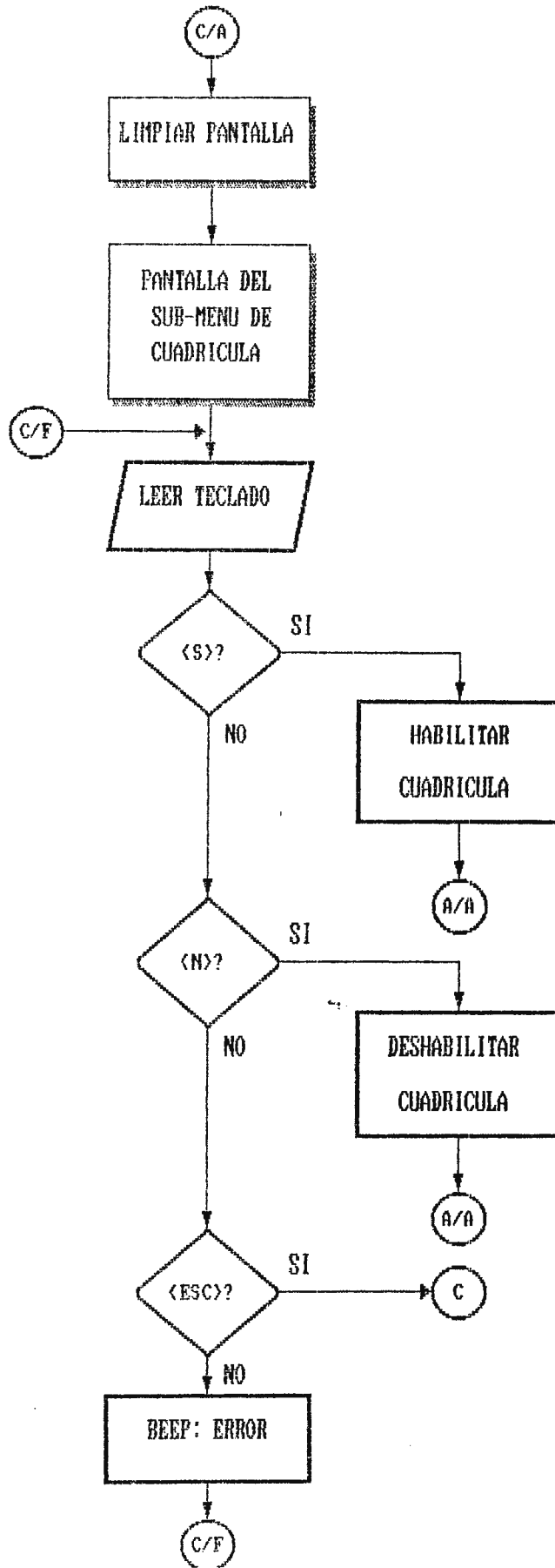


FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)

E.5

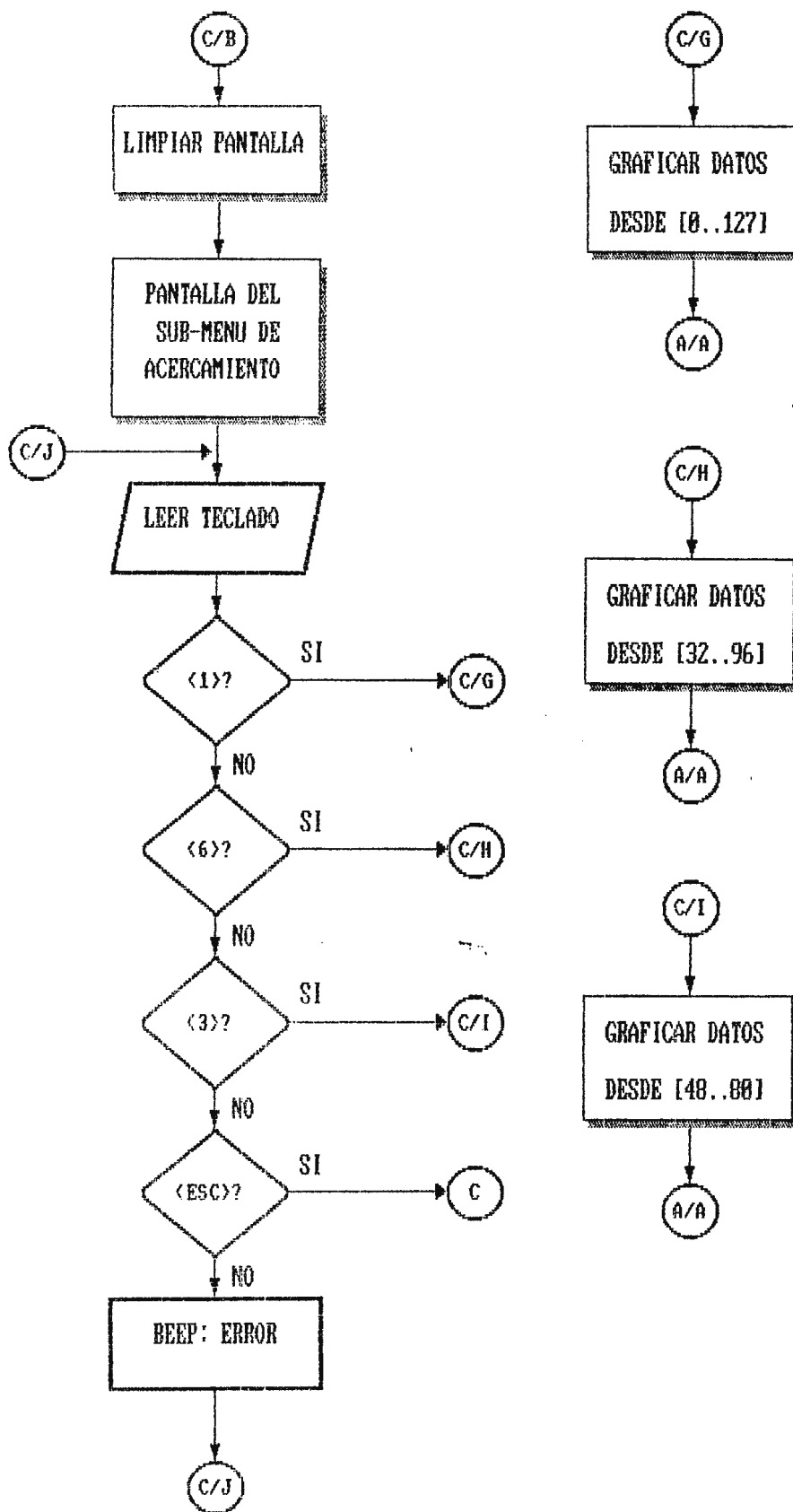


FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)

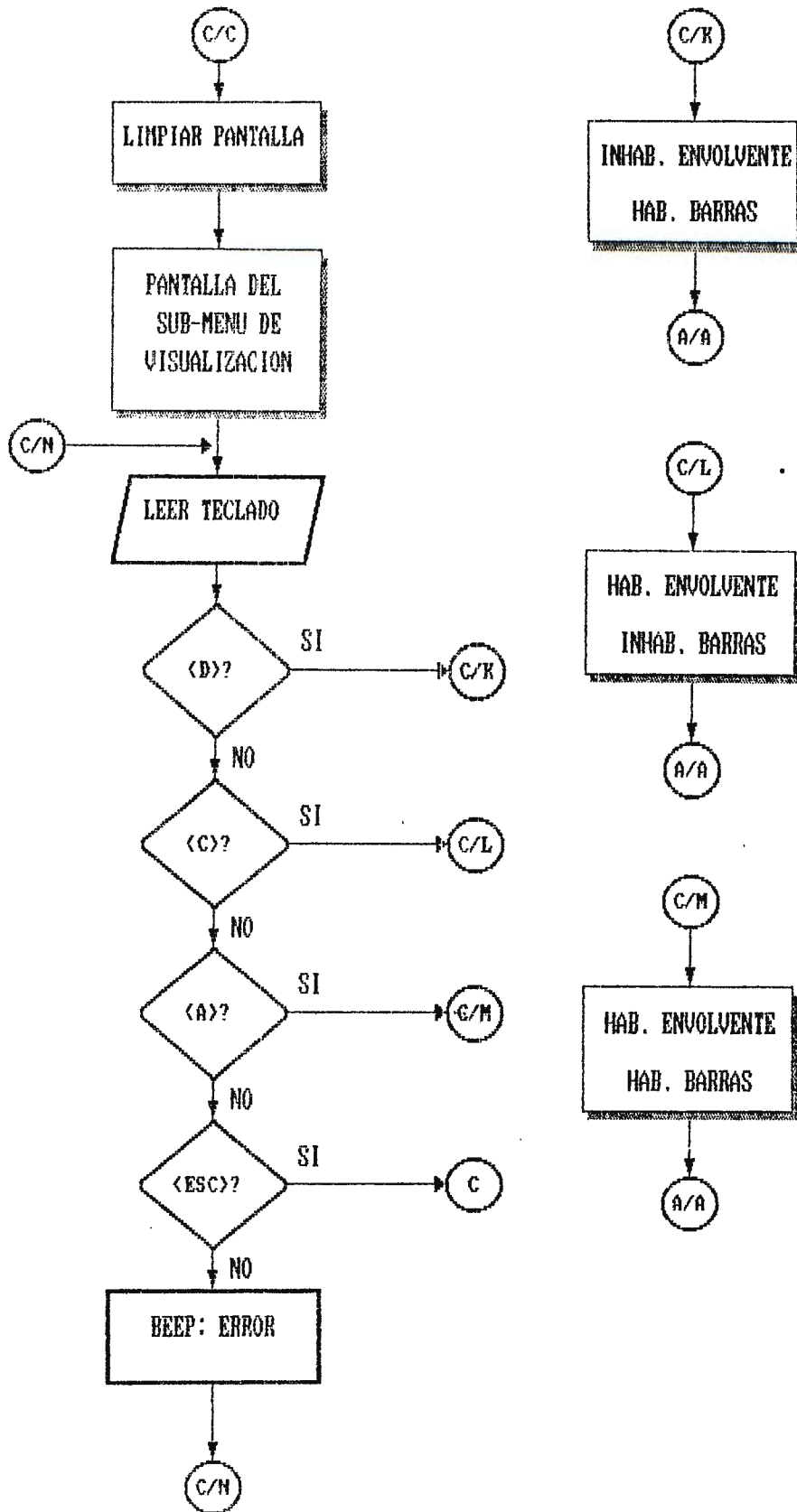


FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)

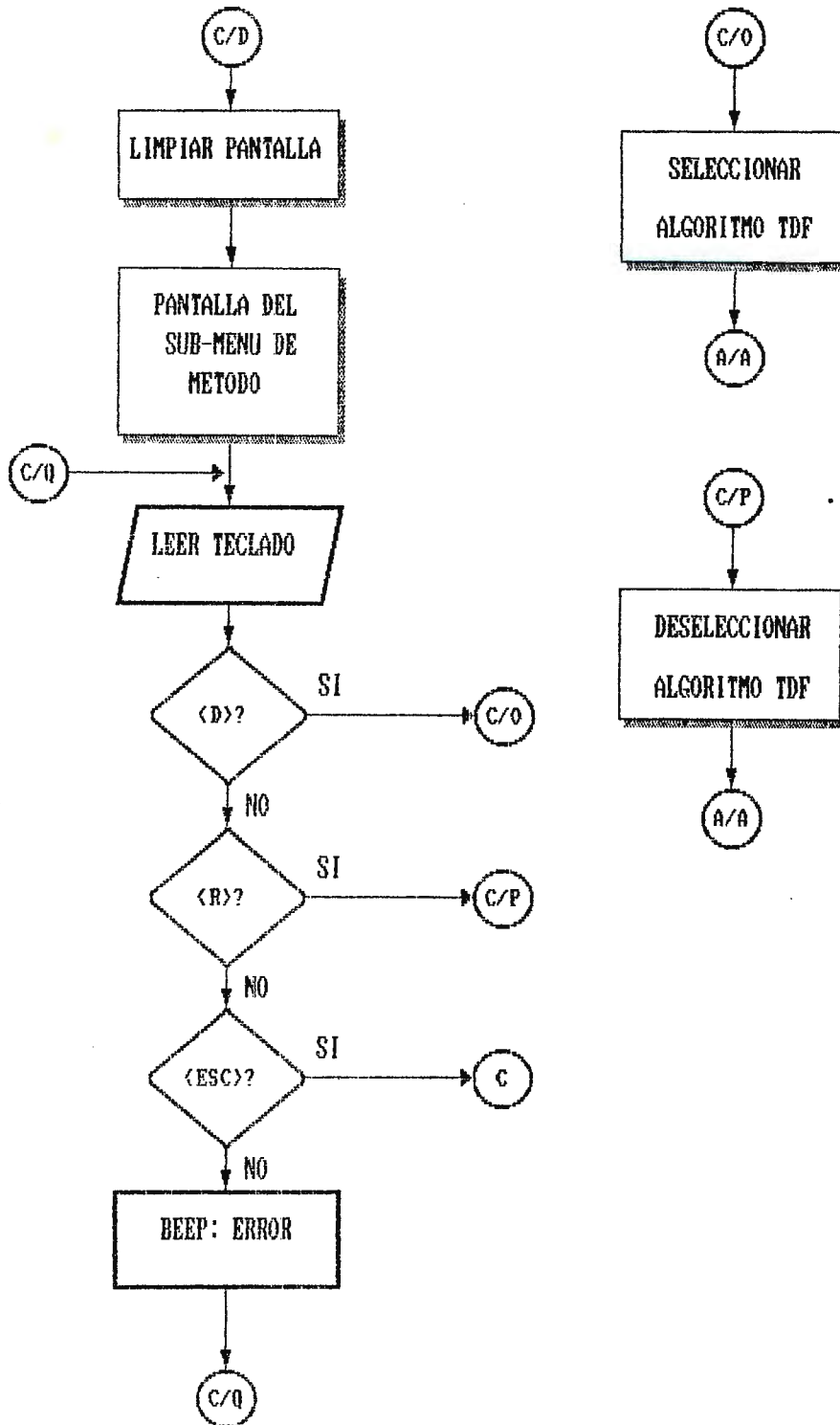
E.7



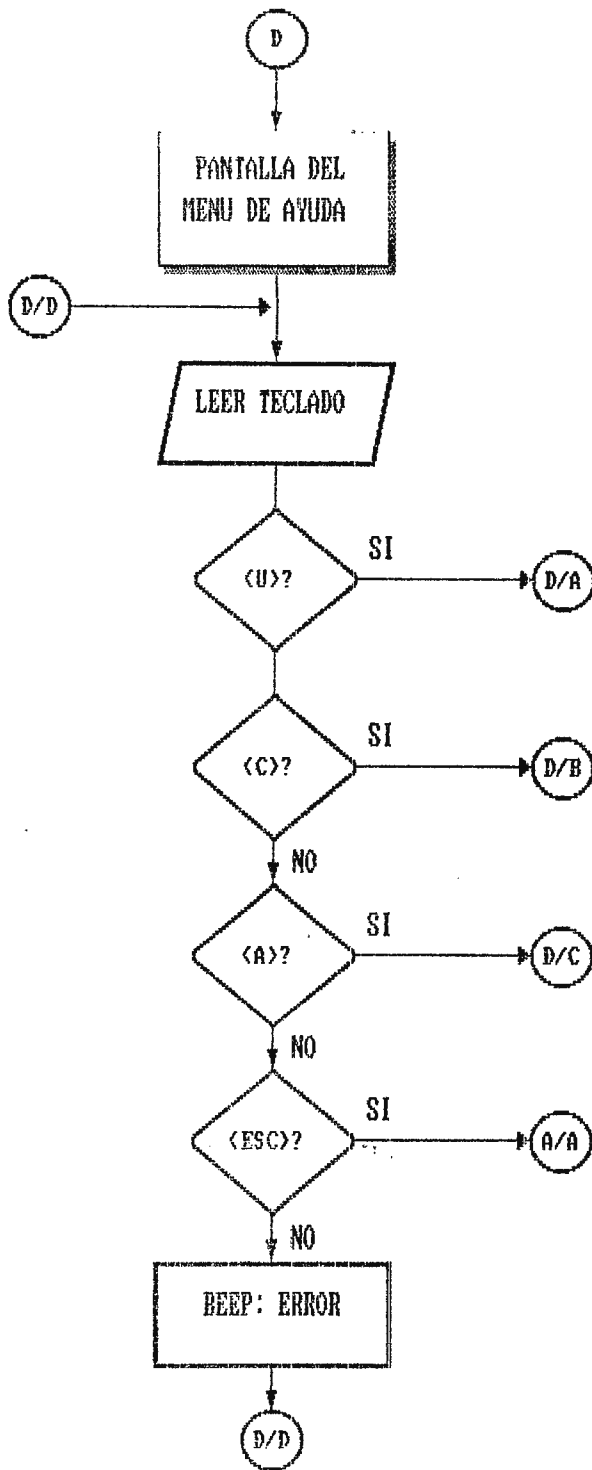
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



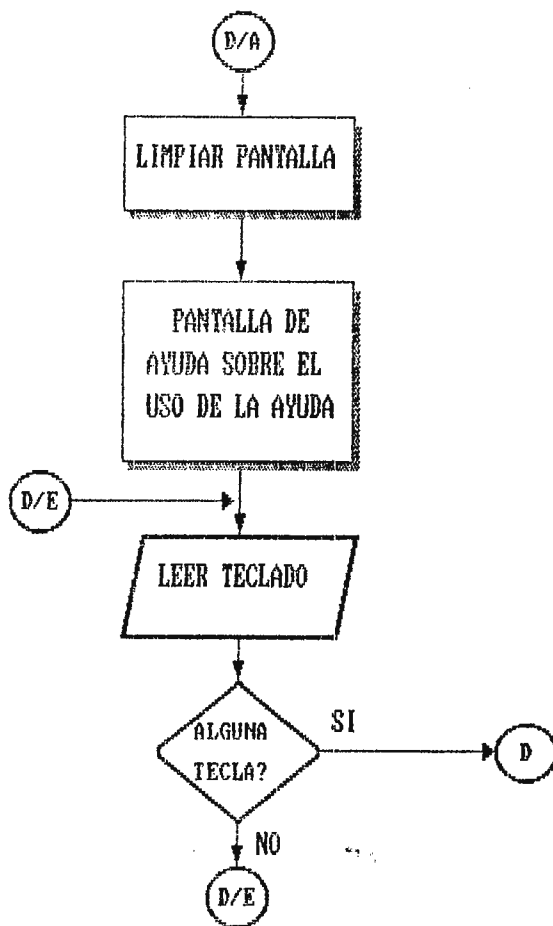
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



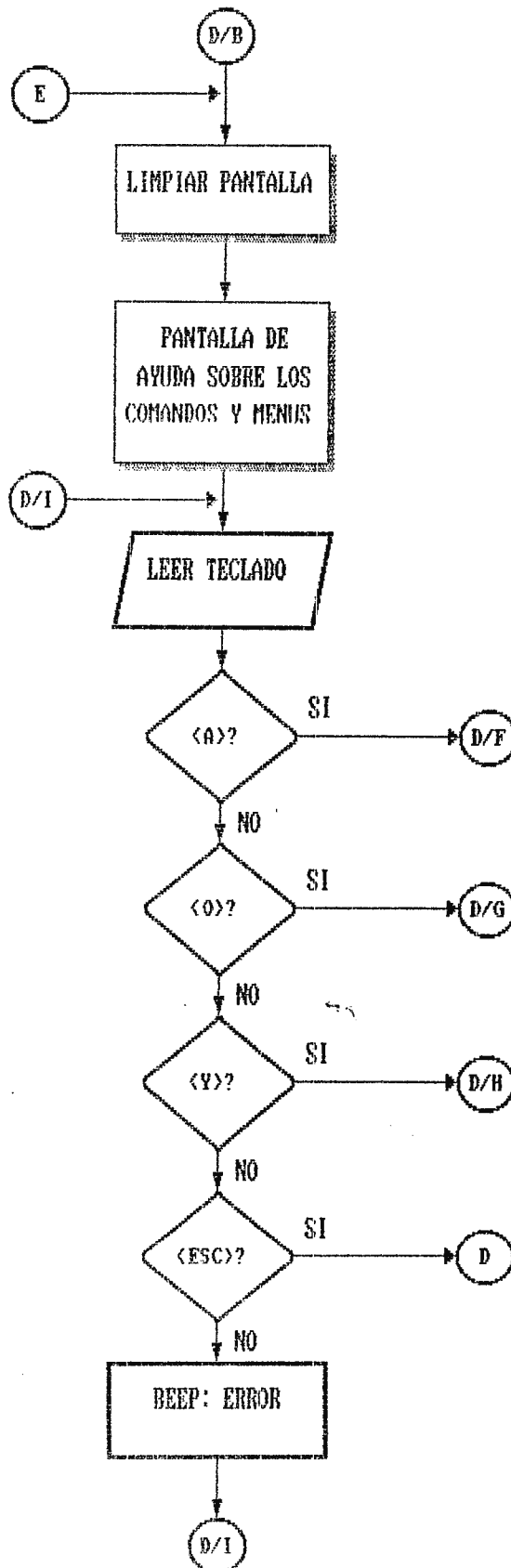
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



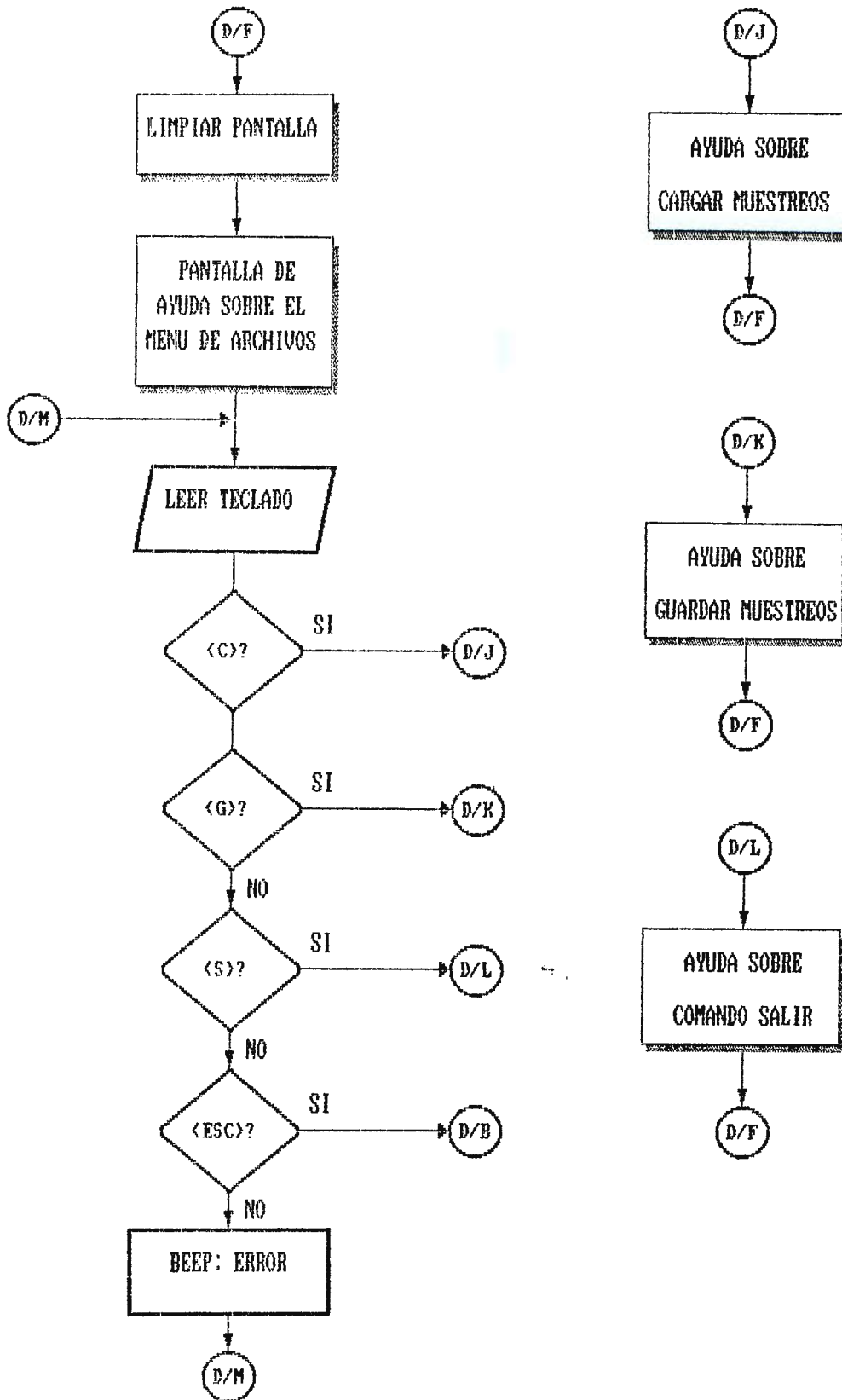
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



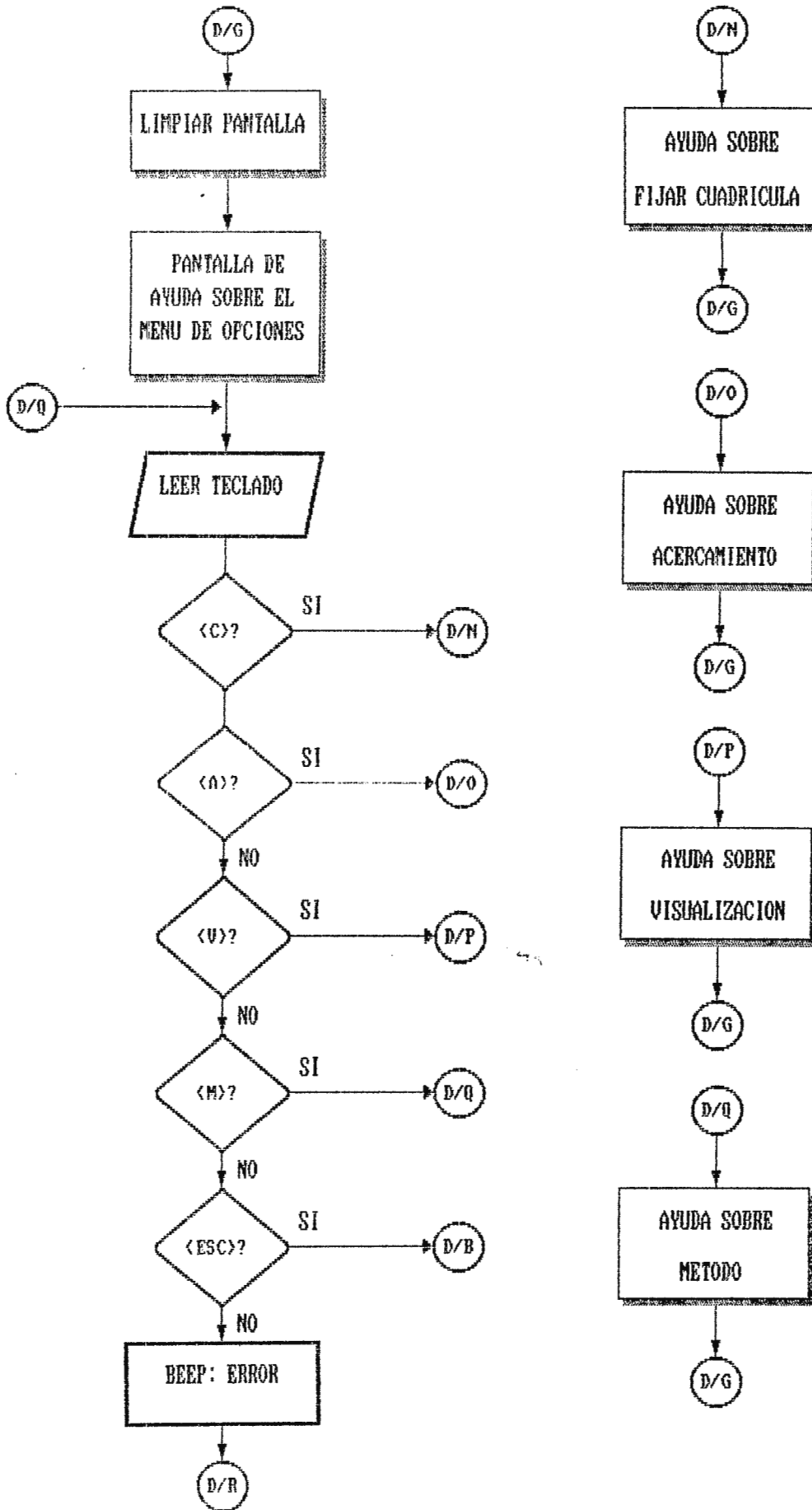
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



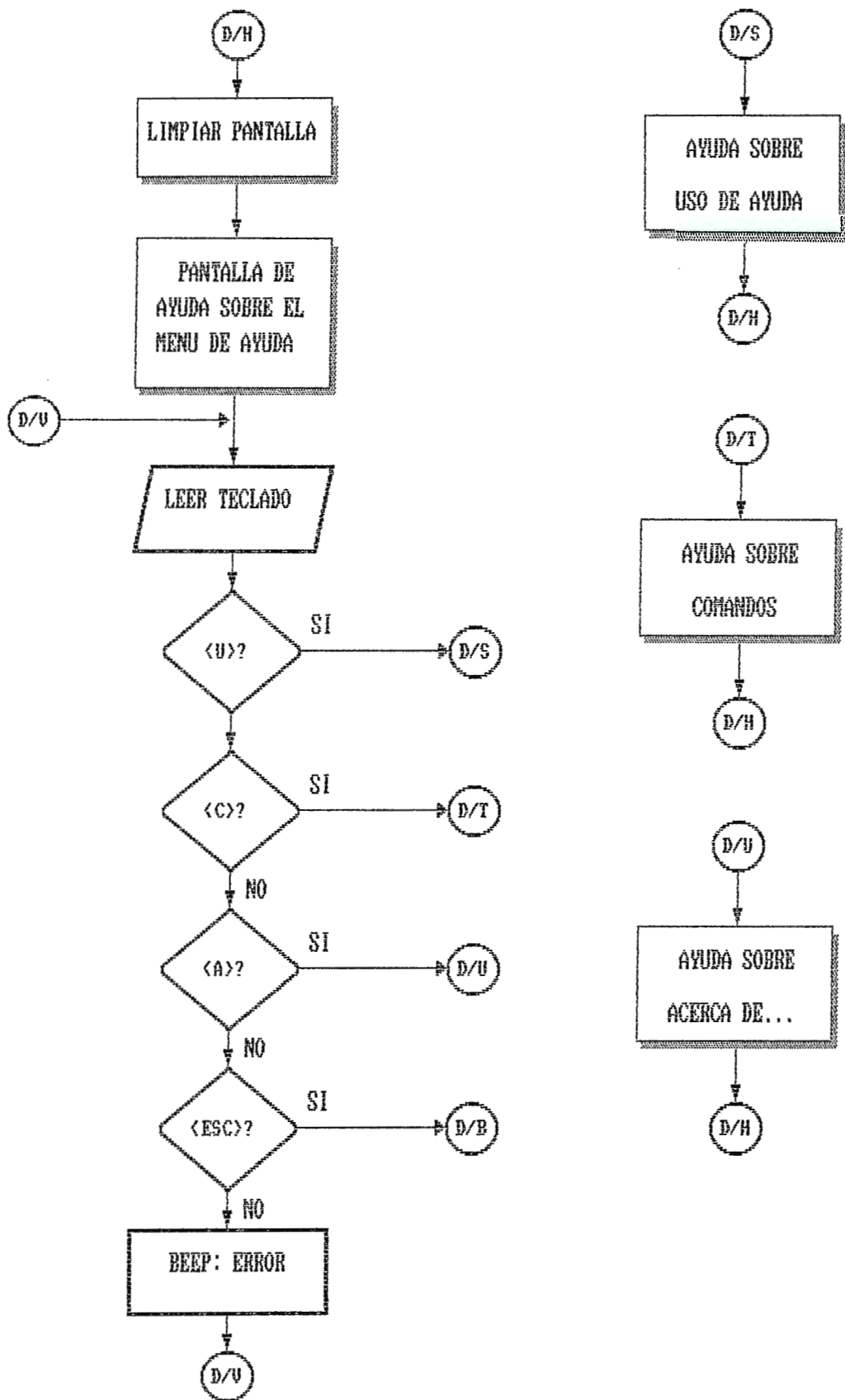
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



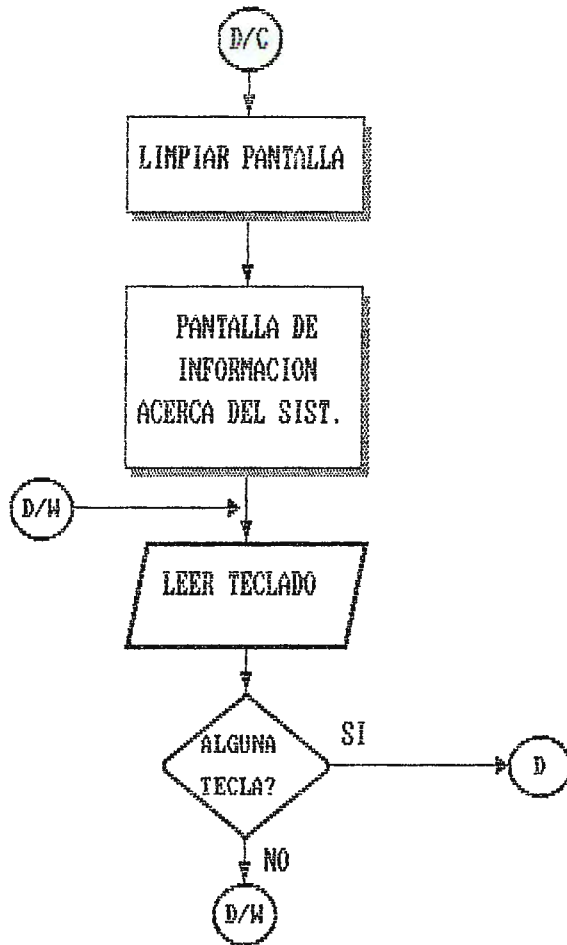
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



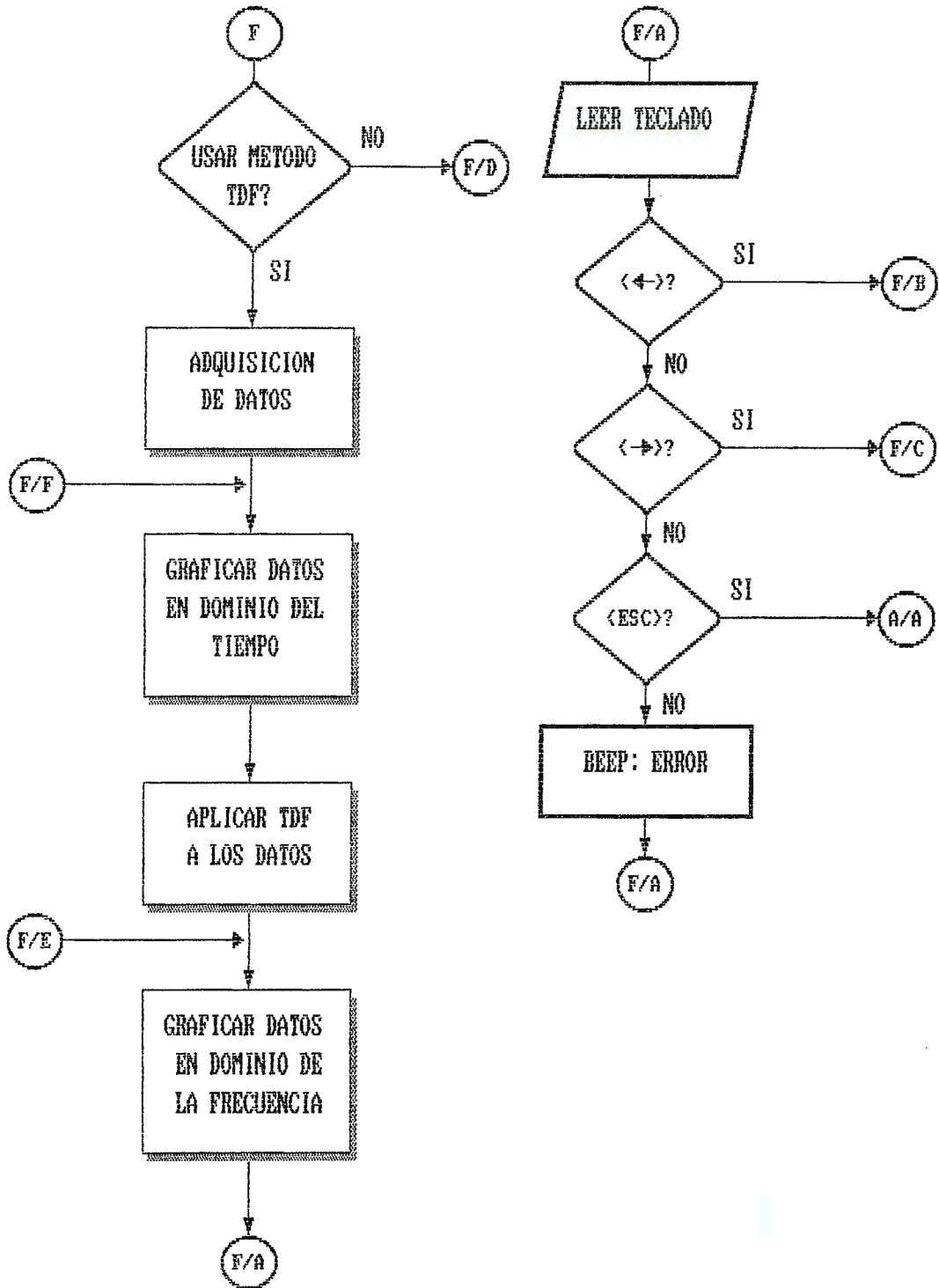
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



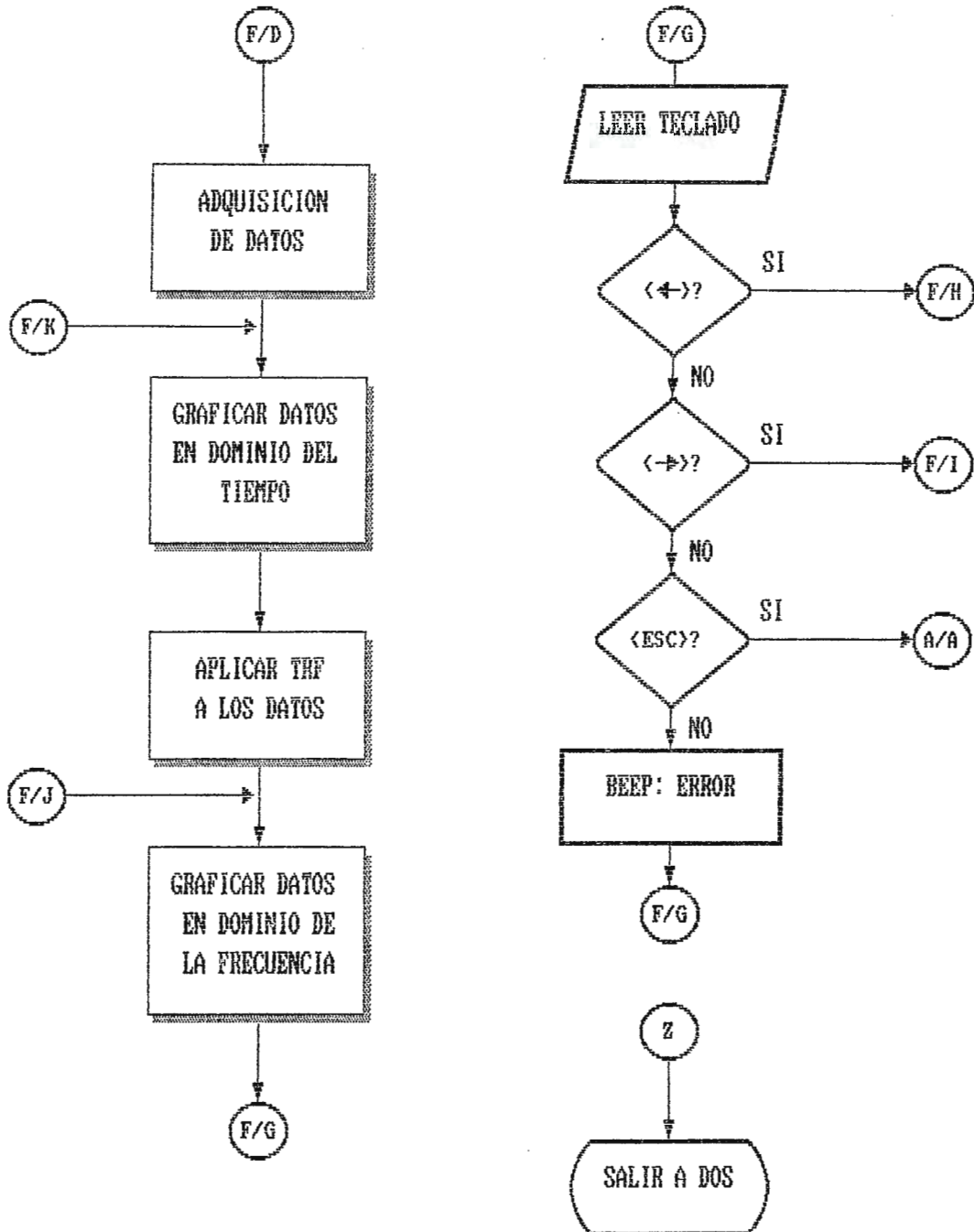
FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



FLUJOGRAMA DEL PROGRAMA PRINCIPAL (CONT)



ANEXO F. GUIA DEL USUARIO



DEMOSTRACION
FISICA DE LA
TRANSFORMADA
DE FOURIER

GUIA DEL USUARIO

1. INTRODUCCION

Este programa fue creado por Victor Manuel Callejas De Paz y Luis Armando Antonio Bonifacio, para controlar una tarjeta adquisitora de datos y realizar un procesamiento de los mismos, como requisito para optar al grado de Ingenieros en Electrónica.

La tarjeta adquisitora tiene como función obtener una secuencia de muestras, a partir de una señal analógica de entrada e introducir esta información dentro de la PC. Además, el programa se encarga de procesar dicha información para obtener a partir de ella, un espectro discreto, correspondiente al espectro de frecuencia de la señal de entrada.

El presente manual sirve para dar una rápida orientación, al usuario del equipo, sobre el uso de éste y el manejo del programa.

2. DESCRIPCIÓN DEL EQUIPO

El equipo consta de dos módulos:

Un módulo interno, que consiste en una tarjeta para computadora PC/AT de 8 bits. Esta se conecta en los puertos de expansión de la computadora. Posee un conector de entrada/salida de 9 pines, cuyas líneas se hallan distribuidas como se muestra en la figura 1.

Un módulo externo, que contiene una etapa de filtros de entrada, así como algunos selectores y el conector de entrada al sistema.

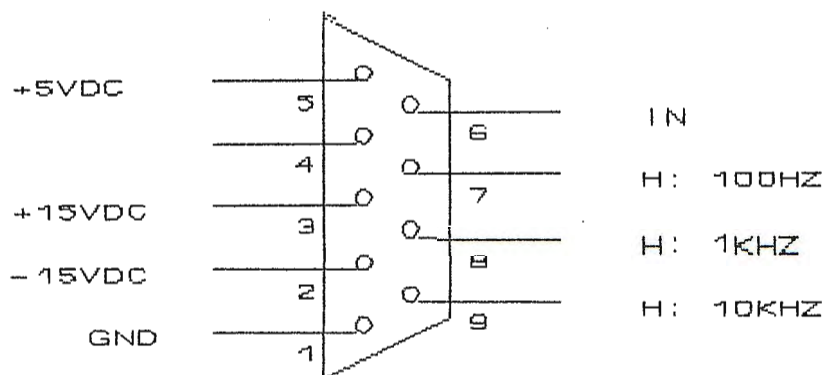


Figura 1. Distribución de líneas en conector externo.

La carátula frontal de este módulo se muestra en la figura 2.

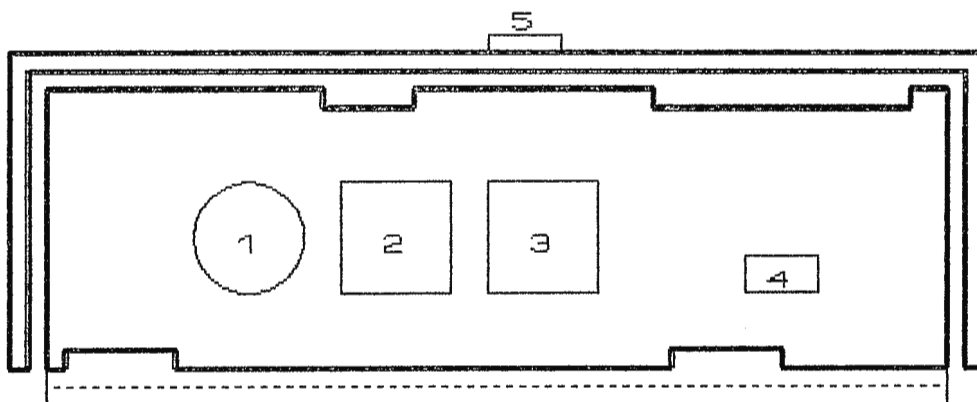


Figura 2. Carátula del módulo externo.

1. Conector de entrada, tipo BNC.
2. Interruptor de encendido/apagado.
3. "Bypass" de filtros de entrada.
4. Indicador de sobretensión.
5. Selector de escalas.

3. DESCRIPCION DEL PROGRAMA

El formato del programa consta básicamente de 2 tipos de pantalla las cuales son:

3.1. La Pantalla de Operación:

Es la que permite al usuario, controlar el programa. La pantalla muestra al inicio 2 barras; en una de ellas, se dá una breve explicación de las opciones que el usuario puede tomar (barra de información al usuario), mientras en la otra se encuentran los menús donde están agrupados los comandos del programa. Cada vez que el usuario seleccione una opción, se desplegará un menú bajo la barra de comandos, el cual será explicado brevemente en la barra de información.

Barra de comandos.

Menú desplegable.

Barra de información.

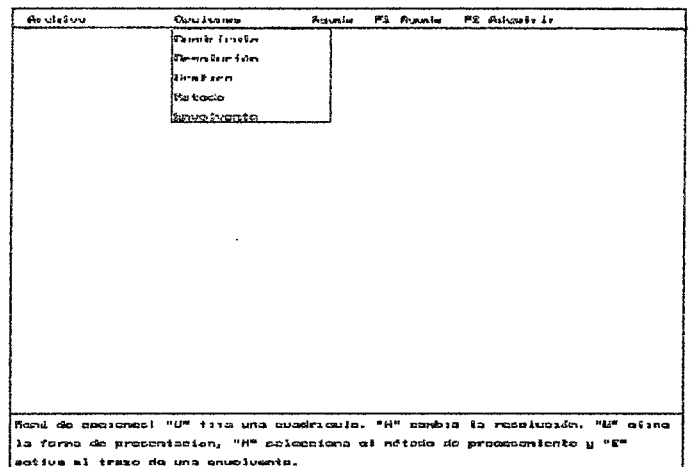


Figura 3. Pantalla de operación.

3.2. La Pantalla de Visualización:

Esta pantalla, aparecerá luego de que el usuario halla seleccionado la forma en que desea ver el espectro; la pantalla muestra dos

ventanas, las cuales poseen diferentes escalas, estas ventanas son:
 La Ventana de Tiempo: que es la que presenta las muestras obtenidas de la señal a analizar, la cual posee una escala de tiempo en el eje de las x expresado en milisegundos y una escala de amplitud en el eje de las y expresado en voltios.

La Ventana de Frecuencia: Posee una escala en frecuencia, expresada en Hertz y otra en amplitud expresada en voltios, en el eje x y y respectivamente, que muestra el espectro de la señal.

Ventana de tiempo.

Ventana de frecuencia.

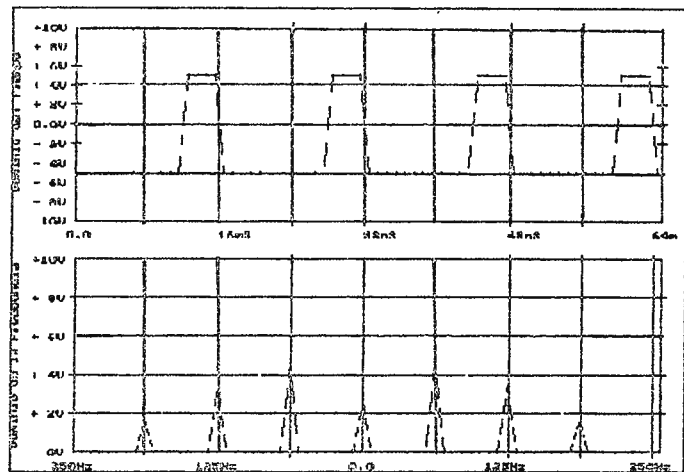


Figura 4. Pantalla de visualización.

4. USO DEL PROGRAMA

Para entrar al programa, es necesario ejecutar el archivo DFF.exe. Luego de algunos segundos, bastará presionar una tecla para llegar al menú principal.

Estando en el menú principal, el usuario puede presionar cualquier tecla según la letra respectiva que aparezca de color rojo en la

barra de comando, para poder así seleccionar cualquier menú del programa, los cuales son:

- Menú de Archivo.
- Menú de Opciones.
- Menú de Ayuda.

Menú de Archivo:

Al seleccionar este menú, se desplegará en pantalla 3 opciones, que el usuario podrá ejecutar, las cuales son:

- Cargar Muestreo:

Este comando, permite que el usuario pueda ver un muestreo, realizado anteriormente. Al elegir esta opción, el programa le preguntará el nombre del archivo, que usted desea ver; el nombre del archivo, deberá tener la dirección completa, de donde se desea leer el archivo; en caso contrario, el programa asumirá que el archivo se encuentra en el directorio corriente.

- Guardar Muestreo:

Este comando, le da la opción al usuario, para almacenar un muestreo adquirido con la tarjeta, que posteriormente pueda ser de utilidad al usuario.

El programa le pedirá al usuario, el nombre con que éste, desea

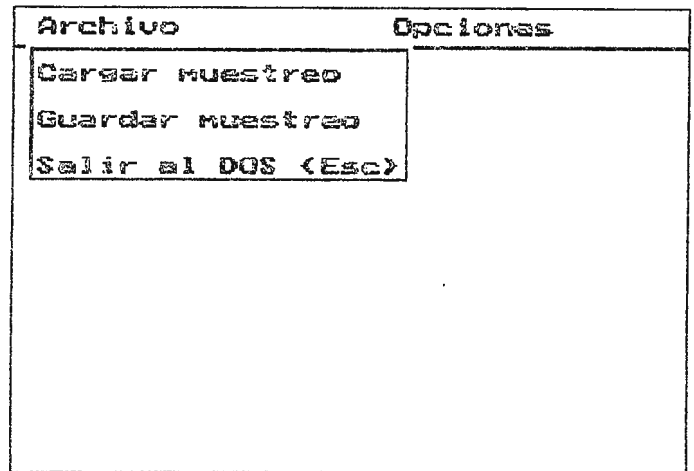


Figura 5. Menú de Archivos.

llamar al muestreo; El nombre podrá tener como máximo 8 caracteres, con la extensión que el usuario escoja.

El formato con que se guardará el archivo es binario, con signo invertido (\$00=10V, \$80=0V, \$ff=-10V).

- Salir al DOS:

Como su nombre lo indica, este comando sirve para salir del programa. Lo cual puede hacerse directamente, presionando la tecla <Esc>, cuando el usuario se encuentre en el menú principal.

Menú de Opciones:

Este menú al ser desplegado, presentará todas las opciones que el usuario pueda hacer con el programa. Estas son:

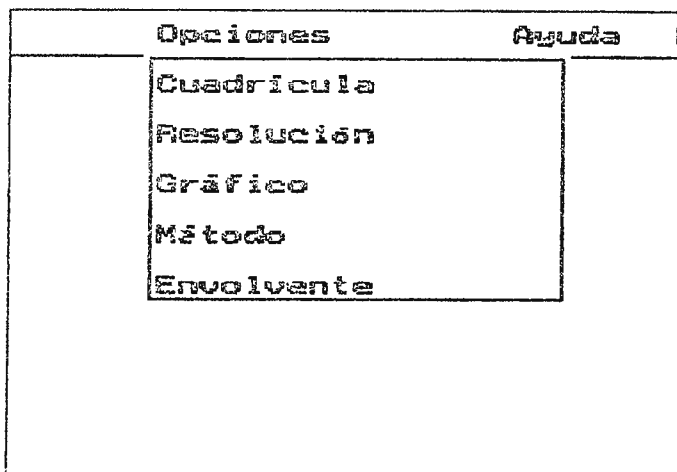


Figura 6. Menú de Opciones.

- Cuadrícula: Al seleccionar esta opción, el programa mostrará en pantalla las palabras "sí" y "no". Al elegir la primera palabra se estará diciendo al programa que las ventanas de tiempo y frecuencia de la pantalla de visualización, quieren ser observadas con cuadrícula. Si por el contrario se elige la palabra "no", el programa quita la cuadrícula de las ventanas.

- Resolución: El usuario tiene la opción de poder ver el espectro (en la ventana de frecuencia) con 128, 64 y 32 puntos. Donde 128 puntos mostrará todo el espectro y 32 ó 64 puntos solo mostrará una parte del espectro, del cual la parte faltante se podrá observar, presionando las teclas <↔> ó <→>.

- Gráfico: El comando Gráfico, presenta 3 opciones: Discreto, Continuo y Ambos. Donde el usuario podrá ver ya sea el muestreo de la señal y el análisis de ésta (espectro) en forma de barras (de forma discreta) o por medio de una línea continua. La combinación de la línea continua y las barras en ambas ventanas es la opción de Ambos.

- Método: Este comando es el que escoge con cual algoritmo se va a analizar la señal; los algoritmos son: el de la Transformada Rápida de Fourier (TRF) y la Transformada Discreta de Fourier (TDF). Cada cual tiene su característica representativa que el usuario deberá tomar en cuenta a la hora de utilizar el programa.

La TDF es la más exacta, pero con la desventaja de ser lenta y la TRF es muy rápida, pero menos exacta que la anterior.

- Envolvente: Como su nombre lo indica, este comando sirve para elegir, si la señal analizada tendrá o no la envolvente al momento de visualizarla en la ventana de frecuencia.

Menú de Ayuda:

Al seleccionar el menú de ayuda se desplegarán en la pantalla de comandos 3 opciones: Uso de la Ayuda, Comandos y Acerca de...

- **Uso de Ayuda:** Este es una breve exposición de lo que se pretende hacer con este menú. La información será desplegada en la pantalla de comandos.

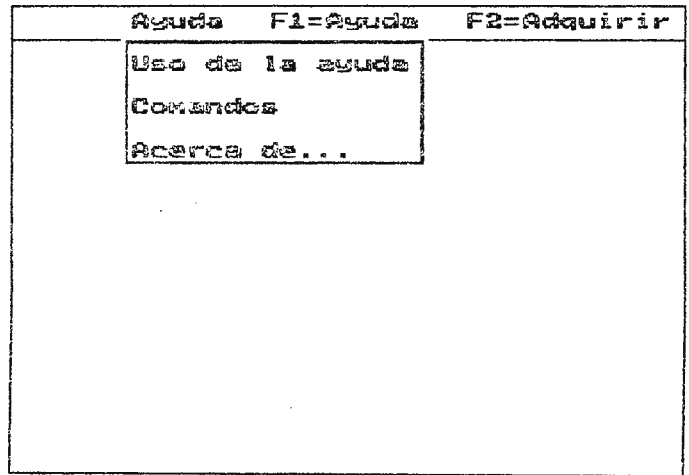


Figura 7. Menú de ayuda.

- **Comandos:** Esta opción es el resumen de este manual presentado en pantalla, para

que el usuario pueda aclarar cualquier duda que se le presente, acerca del funcionamiento del programa, sin tener que ir a ver el manual del usuario.

- **Acerca de ... :** Aquí se presenta un resumen de información general acerca del programa.

La tecla F1:

Esta tecla sirve para entrar directamente a la opción comandos del menú de ayuda. Puede ser llamada siempre que el usuario se encuentre en el menú principal; no así cuando se encuentre dentro de un sub-menú.

La tecla F2:

Esta tecla inicia el proceso de la adquisición de datos. Se le ordena a la tarjeta llenar su "buffer" de memoria auxiliar con los resultados de un muestreo para luego transferirlos hacia la memoria RAM de la PC.

La tecla F3:

Vuelve a gráficar el último muestreo presentado en pantalla, sin necesidad de cargar el archivo en el que se encuentra (en el caso de que se haya guardado el muestreo).

5. ESPECIFICACIONES TECNICAS

Alimentación externa:	No requerida.
Alimentación interna:	+5 VDC. +12 VDC. -12 VDC.
Voltaje máximo de entrada:	± 10 Vpp.
Frecuencia máxima de entrada:	10 KHz.
Impedancia de entrada:	1 M Ω .
Resolución:	8 bits.
Muestras por lectura:	128.
Memoria auxiliar:	256 Bytes.
Velocidad máxima de conversión:	20 μ S.
Tamaño del paquete:	180 KB.
Memoria requerida:	640 KB.
Video:	VGA 16 colores.
Procesador:	286, 386 o 486.

GUIA No. 1 ANALISIS DE FOURIER

I. OBJETIVOS

- Que el estudiante compruebe la teoría de la transformada de Fourier al obtener los espectros de varias señales de laboratorio.
- Que el estudiante observe el distanciamiento entre los armónicos al aumentar o disminuir la frecuencia de la señal de entrada.
- Que el estudiante compare la respuesta obtenida mediante el equipo y la compare contra la respuesta teórica esperada.

II. INTRODUCCION TEORICA

El barón Joseph de Fourier fué un físico y matemático francés nacido en Auxerre (1768-1830). En 1822, Fourier demostró que una señal periódica en el tiempo puede ser representada matemáticamente por una suma infinita de funciones seno y coseno cada una con su propia amplitud, frecuencia y fase. [1]

A éstas funciones seno y coseno que componen a la señal se les denomina armónicos. Estos armónicos tienen la característica que su frecuencia es

$n(\omega_0)$, siendo ω_0 la frecuencia fundamental y $n=1,2,3,\dots$. La frecuencia

del primer armónico es lo que de ahora en adelante entenderemos como frecuencia fundamental. Al trazar una curva continua que pase por las crestas de los armónicos de un espectro se obtiene su envolvente. La naturaleza de la señal dependerá de la amplitud y la fase de cada una de las componentes armónicas.

A pesar de que la demostración de Fourier era de carácter puramente matemático, su teoría es aplicable a las señales eléctricas: Cualquier tipo de señal periódica puede ser considerada como la composición de un número de ondas senoidales. Las ondas senoidales pueden considerarse entonces como el elemento básico de cualquier señal.

Consideremos entonces una señal $g(t)$ formada por la suma de senoidales de frecuencias $0, f_0, 2f_0, \dots, kf_0$.

$$g(t) = a_0 + a_1 \cos 2\pi f_0 t + a_2 \cos 2(2\pi f_0) t + \dots + a_k \cos 2\pi k f_0 t$$

$$+ b_1 \text{ sen } 2\pi f_0 t + b_2 \text{ sen } 2(2\pi f_0) t + \dots + b_k \text{ sen } 2\pi k f_0 t$$

$$= a_0 + \sum_{n=1}^k a_n \cos 2\pi n f_0 t + b_n \text{ sen } 2\pi n f_0 t$$

Aplicando estas ecuaciones a un tren de pulsos rectangulares con un ciclo de trabajo del 50% obtenemos el siguiente espectro:

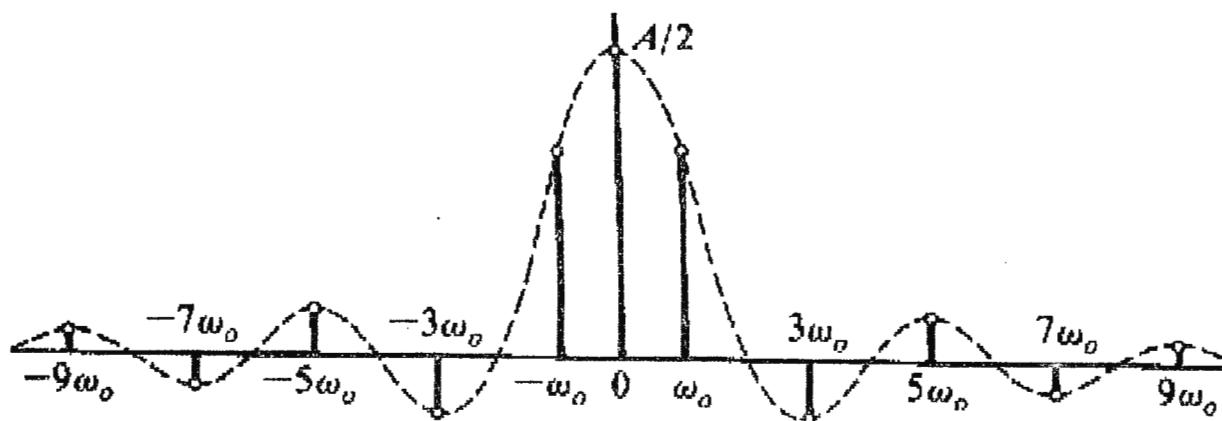


Figura 1. Espectro teórico de una señal de pulsos rectangulares con D.C. 50%.

donde la frecuencia ω_0 depende del inverso del período del tren de pulsos, la amplitud de los armónicos del valor de las constantes a_n y b_n , y la amplitud central del valor promedio de la función.

Resultado similar se obtendría al aplicar dichas ecuaciones a una función cuadrada con simetría al eje x , a excepción que el la constante a_0 del espectro en frecuencia desaparece.

Al deducir la transformada de una señal senoidal pura simplemente se obtiene una barra vertical justo en la frecuencia correspondiente a la de la función seno (Figura 2).

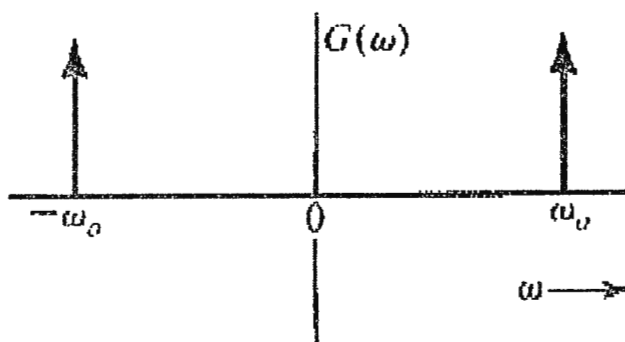


Figura 2. Espectro teórico de una señal senoidal.

A continuación se procederá a comprobar los resultados obtenidos de las transformadas de Fourier de cada función, sometiendo señales reales de características similares a las funciones estudiadas, generadas experimentalmente.

III. PROCEDIMIENTO

INSTRUCCIONES	COMENTARIOS
<p>PARTE I. INGRESO Y CONFIGURACION DE LA TARJETA ADQUISITORA</p> <p>1) Encienda su máquina e ingrese al subdirectorío FOURIER.</p> <p>2) Ejecute el programa DFF.EXE.</p> <p>3) Presione cualquier tecla para avanzar.</p> <p>4) Ingrese al menú de Opciones.</p> <p>5) Seleccione las siguientes condiciones:</p> <ul style="list-style-type: none"> - Cuadrícula: <u>Si</u>. - Resolución: <u>128</u>. - Gráfico: <u>Ambos</u>. - Método: <u>TDF</u>. - Envolvente: <u>No</u>. <p>6) Una vez configurada la tarjeta, seleccione en el módulo conector la escala de 1KHz.</p>	
<p>PARTE II. ANALISIS DE SEÑALES CUADRADAS</p> <p>7) Empleando un generador de señales, introduzca una señal de tipo cuadrada de 600 Hz con un nivel offset equivalente a la mitad de su amplitud.</p> <p>8) Presione <F2> y espere un momento a que el programa analice y presente los resultados en pantalla.</p> <p>9) Una vez obtenido el espectro, mida la amplitud de cada componente armónico y compárela con los datos teóricos deducidos al inicio de la guía.</p> <p>10) Agregue un nivel DC a la señal de salida del generador.</p> <p>11) Repita los pasos 8) y 9).</p> <p>12) Incremente la frecuencia a 800Hz. Analice nuevamente la señal. ¿Qué diferencia encuentra entre ambos espectros?</p>	<p>1) Esto equivaldría a la señal de la Figura 1a.</p> <p>2) Repita las adquisiciones hasta lograr una señal lo más fielmente parecida a la función teórica estudiada.</p>

INSTRUCCIONES	COMENTARIOS
13) Cambie el tipo de señal a analizar por una de tipo senoidal.	
14) Repita los pasos 8) y 9).	
15) Disminuya la frecuencia a 3KHz. ¿Qué ocurre ahora con el espectro?	
16) ¿Cuál ha sido la amplitud de la barra central en ambos casos? ¿Por qué?	
17) Baje todos los controles de ganancia del generador al mínimo, apague el equipo y desconecte los cables.	
18) Salga del programa presionando <Esc> repetidas veces hasta llegar al DOS.	
19) Apague la máquina y ordene el material utilizado sobre la mesa.	

IV. DISCUSION DE RESULTADOS

- ¿Qué relación encuentro entre el espectro en frecuencia de una señal y la frecuencia de la misma?
- Calcule el porcentaje de error de las magnitudes del espectro obtenido experimentalmente respecto al espectro teórico.
- Deduzca el espectro para una función triángulo y compruébelo posteriormente con el equipo de análisis de Fourier.

V. BIBLIOGRAFIA

- Sistemas de Comunicación Lathi, B.P., INTERAMERICANA S.A. DE C.V., MEHICO, D.F., 1986.
- Comunicaciones analógicas, Vol. I. Por el personal de Buck Engineering Co. Inc., LabVolt, U.S.A., 1983.

GUIA No. 2
TRANSFORMADA DISCRETA DE FOURIER VRS. TRANSFORMADA
RAPIDA DE FOURIER

I. OBJETIVOS

- Que el estudiante obtenga los resultados de un análisis espectral realizado con la TDF.
- Que el estudiante obtenga los resultados de un análisis espectral realizado con la TRF.
- Que el estudiante establezca una relación entre las velocidades de procesamiento de la TDF y la TRF.
- Que el estudiante establezca una relación entre el grado de exactitud de cada método.

II. INTRODUCCION TEORICA

Para secuencias de duración finita, es posible representar su espectro en frecuencia mediante la secuencia de la transformada discreta de Fourier (TDF). "La TDF es en sí misma una secuencia más bien que una función de una variable continua, y que corresponde a muestras, igualmente espaciadas en frecuencia, de la transformada de Fourier de la señal". La TDF es importante por dos razones. Primero, nos permite determinar el contenido de frecuencia de una señal, en otras palabras, realizar un *análisis espectral*. La segunda aplicación es la de realizar un *filtrado en el dominio de la frecuencia*. La TDF es directamente aplicable a secuencias de duración finita. Sin embargo, presenta una serie de inconvenientes al aplicarla en secuencias no finitas.

La TDF de una función es una secuencia de frecuencias discretas de duración finita que corresponden a un muestreo discreto y finito en frecuencia de la transformada de Fourier de la función. Este muestreo es convencionalmente realizado a N puntos equidistantes sobre la extensión del período en $0 \leq \omega \leq 2\pi$, o sea

$$\omega_k = 2\pi k/N \quad \text{Para } 0 \leq k \leq N-1 \quad (28)$$

Si $h(n)$ es una secuencia de tiempo discreto con la transformada de Fourier $H(e^{j\omega})$, entonces la TDF, denotada por $H(k)$, es definida como

$$H(k) = H(e^{j\omega}) \Big|_{\omega=\omega_p=2\pi k/N} \quad (29)$$

El algoritmo de la TRF explota las propiedades de simetría de la exponencial compleja de tiempo discreto para reducir significativamente el número de multiplicaciones para el cálculo de la TDF. Para evaluar una TDF de N puntos, el algoritmo de la TRF utilizado acá alcanza su eficiencia máxima cuando n es una potencia de 2, o $N=2^{N\log 2}$, donde $N\log 2$ es siempre entero. Esta restricción no presenta problemas prácticos, ya que la longitud de la secuencia de muestras originales $h(n)$ puede siempre incrementarse hasta la próxima potencia de 2 rellenando la secuencia con un número apropiado de ceros.

La realización de la TRF involucra el cálculo de una matriz formada por diferentes operaciones llamadas patrón de mariposa (Ver Figura 1), las cuales toman dos datos de la secuencia de entrada y de acuerdo con un factor los operan para obtener dos nuevos valores que, a su vez, se recombinarán con otros hasta llegar a formar el espectro de frecuencia.

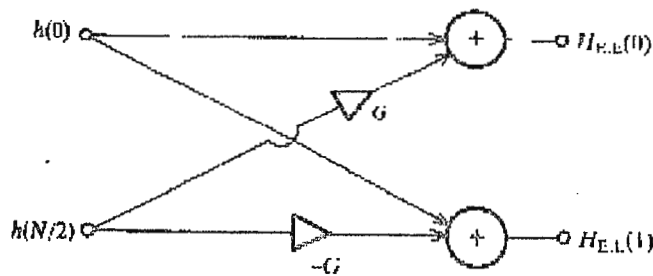


Figura 3.16. Patrón de mariposa ($G=1$).[7]

III. PROCEDIMIENTO

INSTRUCCIONES	COMENTARIOS
<p>PARTE I. INGRESO Y CONFIGURACION DE LA TARJETA ADQUISITORA</p> <ol style="list-style-type: none"> 1) Encienda su máquina e ingrese al subdirectorio FOURIER. 2) Ejecute el programa DFF.EXE. 3) Presione cualquier tecla para avanzar. 4) Ingrese al menú de Opciones. 	

INSTRUCCIONES	COMENTARIOS
<p>5) Seleccione las siguientes condiciones:</p> <ul style="list-style-type: none"> - <u>Cuadrícula</u>: <u>Si</u>. - <u>Resolución</u>: <u>128</u>. - <u>Gráfico</u>: <u>Ambos</u>. - <u>Método</u>: <u>TDF</u>. - <u>Envolvente</u>: <u>No</u>. <p>6) Una vez configurada la tarjeta, seleccione en el módulo conector la escala de 1KHz.</p>	
PARTE II. TRANSFORMADA DISCRETA DE FOURIER	
<p>7) Utilizando un generador de señales, introduzca una señal cuadrada de 15Vpp a 312Hz, sin nivel DC.</p>	
<p>8) Utilice <F2> para analizar la señal y obtener su espectro.</p>	
<p>9) Ingrese al menú de Archivos y guarde el muestreo bajo el nombre de SQ312H.DAT</p>	
<p>10) Cambie la frecuencia del generador a 234Hz y obtenga su espectro.</p>	
<p>11) Guarde este muestreo con el nombre SQ234H.DAT.</p>	
<p>12) Cambie ahora la frecuencia del generador a 468Hz.</p>	
<p>13) Igual que en los casos anteriores, obtenga su espectro y guardelo en SQ486H.dat</p>	
<p>14) Cambie ahora la señal de entrada por una senoidal de 15Vpp a 312Hz.</p>	
<p>15) Guarde su muestreo con el nombre de SEN312H.DAT</p>	
<p>16) Cambie luego la frecuencia de la señal a 1.25KHz. Grábelo como SEN1250H.DAT.</p>	
PARTE II. TRANSFORMADA RAPIDA DE FOURIER	
<p>17) Ahora comience a llamar uno a uno los diferentes archivos que se crearon alternando el método de procesamiento de TDF a TRF. Compare la rapidez con que se presentan los resultados en cada método.</p>	
<p>18) Compare cada espectro resultante contra su espectro teórico.</p>	

IV. DISCUSION DE RESULTADOS

- ¿Qué relación encuentro entre las velocidades de procesamiento de la TDF y la TRF?
- Calcule el porcentaje de error de las magnitudes del espectro obtenido experimentalmente respecto al espectro teórico para cada método.
- ¿Bajo qué condiciones se obtiene los mejores resultados con la TRF?

V. BIBLIOGRAFIA

- Lathi, B.P. Sistemas de Comunicación, INTERAMERICANA S.A. DE C.V., MEHICO, D.F., 1986.
- Por el personal de Buck Engineering Co. Inc. Comunicaciones analógicas, Vol. I., LabVolt, U.S.A., 1983.
- Kuc, Roman. INTRODUCTION TO DIGITAL SIGNAL PROCESSING, McGraw-Hill, U.S.A., 1988.