

63
24



Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

**PROCESAMIENTO DE SEÑALES
POR MEDIO DE UNA
COMPUTADORA PERSONAL**

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
P R E S E N T A N
HECTOR GARCIA HERNANDEZ
NELSON MARTINEZ CEDEÑO



Ciudad Universitaria

1991

FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

TESIS CON FALLA DE ORIGEN

INDICE

Capítulo I INTRODUCCION.

- I.1 Generalidades.
- I.2 Objetivo.
- I.3 Síntesis de desarrollo.

Capítulo II EL OSCILOSCOPIO Y EL ANALIZADOR DE ESPECTROS.

II.1 OSCILOSCOPIO.

II.1.1 Generalidades

II.1.2.1 Operación básica.

II.1.2.2 Tubo de rayos catódicos.

II.1.2.3 Varios: Atenuador y amplificador vertical.

Amplificador horizontal. Circuito de barrido.
Pendiente y nivel. Entrada de disparo externo.
Intensidad y foco.

II.1.3 Características adicionales.

II.1.4 Osciloscopio digital.

II.1.4.1 Generalidades.

II.1.4.2 Principales parámetros de consideración de un osciloscopio digital.

II.2 ANALIZADOR DE ESPECTROS.

II.2.1 Trazos espectrales.

II.2.1.1 Señales de onda continua.

II.2.1.2 Modulación en amplitud.

II.2.1.3 Modulación en frecuencia.

II.2.1.4 Modulación por pulsos.

II.2.2 El analizador de espectros.

II.2.2.1 Aplicaciones.

II.2.2.2 Ejemplo descriptivo.

II.2.3 Características básicas de un analizador de espectros.

Capítulo III PC

III.1 ARQUITECTURA GENERAL DE LA PC.

III.1.1 Unidad del sistema

III.2 EL MICROPROCESADOR 8086 Y EL MICROPROCESADOR 8088.

III.2.1 Arquitectura del 8086

III.2.1.1 La unidad de interfaz para Bus (BIU).

III.2.1.2 La unidad de ejecución (EU).

- III.2.2 Acceso de datos de memoria (direccionamiento).
- III.2.3 Hardware del 8086.
- III.2.4 Señales de flujo básicas en los buses.
- III.2.5 Direccionamiento y decodificación de direcciones.
- III.2.6 Acceso a memoria y a puertos.

III.3 ORGANIZACION DE LA MEMORIA.

- III.3.1 Categorías de RAM.
- III.3.2 Distribución de memoria.
- III.3.3 Mapa de memoria.
- III.3.4 Mapa de puertos.
- III.3.5 Segmentación de memoria.

III.4 INTERRUPCIONES DE 8086.

- III.4.1 Interrupción no mascarable.
- III.4.2 Interrupciones por software - Tipos 0 a 255.
- III.4.3 Interrupciones INTR - Tipos 0 a 255.

III.4.3.1 Controlador de prioridad de interrupciones.

III.4.4 Prioridades de las interrupciones del 8088.

III.5 OPERACIONES DE LA UNIDAD DE BUS DEL SISTEMA.

- III.5.1 Ciclos del Bus.
- III.5.2 Tiempos del Bus del sistema.
- III.5.3 Señales del Bus.
- III.5.4 Capacidades de Bus del sistema para manejo de carga y datos.

III.5.4.1 Capacidad de manejo de señales del Bus del sistema.

III.5.4.2 Capacidad de manejo de corriente del Bus del sistema.

III.5.4.3 Carga del Bus del sistema presentada en los slots.

III.5.4.4 Carga capacitiva del Bus.

III.5.5 Características mecánicas y de potencia del Bus del sistema.

III.5.5.1 Slot para tarjetas en el Bus del sistema.

III.5.5.2 Tamaño de la tarjeta.

III.5.5.3 Potencia de la unidad del sistema.

III.5.5.4 Desacoplamiento de potencia.

Capítulo IV SISTEMAS DE ADQUISICION DE DATOS.

IV.1 SAD'S BASADOS EN UN SISTEMA DIGITAL DE COMPUTO TIPO IBM PC.

- IV.1.1 Consideraciones de diseño.
- IV.1.2 Muestreo de datos.
- IV.1.3 Procesamiento digital de señales.

IV.2 CIRCUITOS PARA EL PROCESO DIGITAL DE SEÑALES (DSP's)

IV.3 FAMILIA TMS32010

IV.3.1 Introducción.

IV.3.2 TMS32010

IV.3.2.1 Arquitectura.

IV.3.2.2 Elementos aritméticos.

IV.3.2.3 Memoria de datos.

IV.3.3 Modos de direccionamiento.

IV.3.4 Registros.

IV.3.5 Memoria externa para programa.

IV.3.5.1 Inicio de ejecución.

IV.3.6 Contador de programa (PC) y stack.

IV.3.7 Registro de estado.

IV.3.8 Funciones de entrada salida.

IV.3.9 Decodificación del Bus de direcciones.

IV.3.5.10 La terminal BIO.

IV.3.5.11 Interrupciones.

IV.3.5.12 Reset.

IV.3.5.13 Reloj/oscilador.

IV.3.5.14 Descripciones de las terminales.

Capítulo V ALGORITMOS PARA PROCESAMIENTO DE SEÑALES.

V.1.- Serie Discreta de Fourier (DFS) y Transformada Discreta de Fourier (DFT).

V.1.1 Introducción.

V.1.2 Series Discretas de Fourier (DFS).

V.1.3 Transformada Discreta de Fourier (DFT).

V.1.4 Transformada Rápida de Fourier (FFT).

V.1.4.1 Descomposición en tiempo.

Capítulo VI DESCRIPCION DEL SISTEMA.

VI.1 Generalidades y requerimientos de equipo.

VI.2 Hardware del sistema.

VI.2.1 Funciones del sistema.

VI.2.2 Arquitectura del sistema.

VI.2.2.1 Descripción de los subsistemas.

VI.2.3 Requerimientos de potencia.

VI.3 Software.

VI.3.1 Arbol de llamada de programas

VI.3.2 Programa de interacción con el usuario (GRAPHICS).

VI.3.2.1 Invocando al Programa

VI.3.2.2 Procesos.

VI.3.2.2.1 Archivo.

VI.3.2.2.1.1 Leer.

VI.3.2.2.1.2 Escribir.

VI.3.2.2.1.3 Terminar.

VI.3.2.2.2 Muestreo.

VI.3.2.2.2.1 Tiempo por división

VI.3.2.2.2.2 Iniciar

- VI.3.2.2.3 Pantalla.
 - VI.3.2.2.3.1 Tiempo.
 - VI.3.2.2.3.2 Frecuencia.
 - VI.3.2.2.3.3 Imprimir.
- VI.3.3 Programas de control de la tarjeta.
- VI.3.4 Algoritmos procesamiento para el TMS32010.
- VI.3.5 Programas de apoyo para el desarrollo
- VI.4 Implementación y desarrollo.

Capítulo VII PRUEBAS DE CAMPO.

Capítulo VIII CONCLUSIONES.

Apéndice A INSTRUCCIONES DEL TMS32010.

Apéndice B MANUAL DE USUARIO.

Apéndice C LISTADO DE PROGRAMAS.

Apéndice D FUNCIONES DE BIOS

Apéndice E DIAGRAMA DE TARJETA. HOJAS DE DATOS.

Apéndice F COMPONENTES Y COSTOS.

BIBLIOGRAFIA

I. INTRODUCCION

I.1 Generalidades

Durante su transcurso en la carrera de Ingeniería Mecánica Eléctrica, el estudiante puede darse cuenta de que, aún cuando existe la mejor disposición del profesorado para impartir sus cátedras, hay ocasiones en las que los equipos utilizados para realizar prácticas son insuficientes. Esta deficiencia se debe básicamente a que el costo de los equipos es muy elevado y no siempre se dispone de fondos suficientes para cubrirla. Un caso en particular corresponde a los equipos del laboratorio de procesamiento de señales.

Consideramos que como alumnos de la Facultad de Ingeniería tenemos la responsabilidad de retribuir en alguna forma todo lo que esta institución nos ha entregado y pensamos que una opción es la de desarrollar un equipo que permita realizar diversos tipos de análisis de señales eléctricas sin tener que disponer necesariamente de un osciloscopio o de un analizador de espectros.

Dadas las condiciones anteriores y pensando en la gran cantidad de equipos de cómputo tipo PC de que se dispone actualmente decidimos que sería una buena idea implementar un dispositivo que permitiera emplear estos equipos para el análisis de señales evitando así la necesidad de adquirir equipos que funcionan únicamente para este fin y aprovechando las ventajas que nos ofrece el poder utilizar la computadora para otros usos. Asimismo, se plantea una ventaja adicional que es la de permitir al alumno guardar los datos obtenidos en el laboratorio en un medio transportable tal como es un diskette a fin de poder estudiar a detalle los fenómenos observados en el laboratorio, en forma más personal con ayuda de una PC.

I.2 Objetivo

Tomando en cuenta las ideas expuestas anteriormente diremos que el objetivo a cubrir es : Desarrollar una tarjeta basada en circuitos electrónicos digitales que permita discretizar una señal proveniente de un sistema que genere una señal de tipo eléctrico para su análisis en los dominios tanto del tiempo como de la frecuencia y que permita guardar dicha señal para su análisis fuera del laboratorio

Esta tarjeta deberá cumplir con las siguientes características :

- Constará en su etapa de entrada de un filtro digital que permitirá evitar problemas de ruido así como mantener un ancho de banda de 100 kHz. Tendrá además un convertidor A/D de tipo rápido (*flash*)
- En la etapa de proceso manejará un microprocesador de uso específico para procesamiento de señales TMS32010, el cual manejará su propio banco de memoria de acceso rápido.
- Debe ser susceptible de ser instalada en el *slot* de expansión de cualquier PC XT compatible.
- Se podrá comunicar con el procesador 8088 disponible en la PC mediante programas en ensamblador.

- El análisis en tiempo permitirá determinar los voltajes máximo y mínimo de la señal y mostrará la gráfica de la señal en el monitor de video indicando los tiempos de inicio y fin del muestreo.
- El análisis en frecuencia se realizará mediante el método de la Transformada Rápida de Fourier (FFT) mediante el cual se podrá determinar la potencia de la señal así como su componente de Cd y cuyo espectro se mostrará en el monitor de video indicando las frecuencias inicial y final.
- La información obtenida podrá ser grabada en un disco flexible a fin de poder ser estudiada en cualquier momento.
- Las gráficas mostradas en la pantalla podrán ser impresas en papel.
- Las acciones a realizar por el sistema se manejarán mediante un programa a base de menús que permitirá realizar todas las acciones indicadas anteriormente.
- Se manejará un rango de voltaje de - 12 V a + 12 V.
- Se manejará un rango de frecuencias de 0 a 100 kHz

I.3 Síntesis del desarrollo

El hecho de utilizar una computadora PC nos indicó que teníamos dos posibilidades : a) Desarrollar un dispositivo externo que se conectara a la PC por su puerto serie o paralelo ; o b) Construir una tarjeta que se conectara a un slot interno de la PC. Esta segunda idea fue la que consideramos más viable dado que en esta forma se tiene un control muy amplio sobre las señales de control del computador.

Ahora bien, se tenían varias opciones para la construcción de la tarjeta, siendo las dos más viables las siguientes : a) Emplear el 8086 de la PC para realizar los cálculos que fuesen necesarios, utilizando también los circuitos de memoria de la PC teniendo en la tarjeta solamente los circuitos de adaptación, transformación y control de la señal a analizar ; o bien b) Realizar la implementación en base a un procesador dedicado exclusivamente al procesamiento de señales el cual requiere, para funcionar adecuadamente, de su propio banco de memoria de alta velocidad. Esta última opción fue la que nos pareció más interesante pues nos presentaba todo un campo nuevo por explorar, en los microprocesadores de uso específico, nunca habíamos manejado ninguno.

Antes de comenzar a construir la tarjeta desarrollamos un amplio trabajo de investigación que nos permitió conocer a fondo los requerimientos que deberíamos de satisfacer así como las características del equipo que habíamos decidido emplear. El resultado de este trabajo se muestra en los capítulos II, III, IV y V

Posteriormente pasamos a la práctica implementando los circuitos en la tarjeta y realizando una innumerable lista de pruebas conforme se desarrollaba cada sección. Paralelamente se trabajó en el desarrollo de *software* que permitiera al usuario observar las señales que son muestreadas. Cuando ambas partes llegaron a un nivel adecuado de desarrollo se comenzaron a realizar pruebas de campo que permitieron detectar errores y afinar detalles. Lo anterior se detalla en los capítulos VI y VII.

Esperamos que este trabajo resulte útil, como lo ha sido para nosotros, para otros estudiantes que requieran información acerca de los tópicos aquí mencionados, pues consideramos que el presente es un trabajo que puede emplearse en las áreas de Electrónica, Computación y Comunicaciones.

II.1.-OSCILOSCOPIO.

II.1.1- Generalidades.

Actualmente el uso del osciloscopio se ha generalizado para la medición de señales eléctricas por la facilidad que representa la visualización de tales señales en la determinación de sus parámetros o para esbozar el comportamiento de un sistema específico.

Parte importante del uso del osciloscopio estriba en el hecho de que este instrumento permite hacer la medición de cantidades de tipo diferente a las eléctricas (v.g. mecánicas, luminosas, etc.) a través de transductores adecuados.

El conocimiento de las características y limitaciones del osciloscopio, permite aprovechar mejor sus potencialidades y lograr así mayor calidad es las mediciones que se efectuen con este instrumento.

II.1.2.1.- Operación básica.

El diagrama de bloques básico de un osciloscopio es:

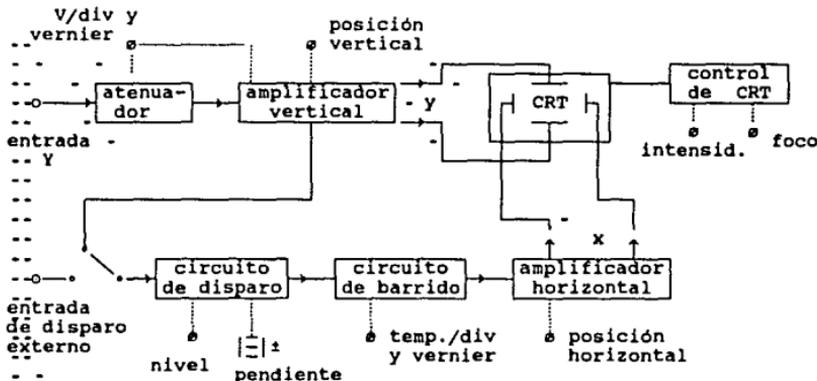


Fig. II.1 Diagrama de bloques básicos.

El funcionamiento de cada uno de los bloques se explica a continuación:

II.1.2.2.- Tubo de rayos catódicos.

CRT.- Tubo de rayos catódicos (Cathode-ray tube).

Este elemento se puede considerar la parte principal del osciloscopio, pues en él se transforma una señal eléctrica a una imagen visible.

Consiste básicamente en un tubo de vacío que contiene un cañón de electrones, que son concentrados en un haz y enfocados hacia una pantalla de fósforo en el extremo opuesto del cañón.

El haz sufre una deflexión debida a un par de placas verticales y horizontales antes de incidir sobre el fósforo de la pantalla.

Con mas detalle el diagrama del CTR es el siguiente:

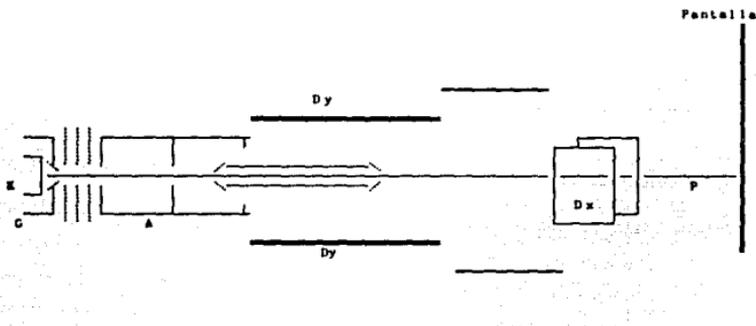


Fig II-2 Detalle del osciloscopio.

En él se muestran:

- 1.-Cátodo (K).- Emite los electrones.
- 2.-Cilindro de Wehnelt (G).- Controla la intensidad del haz luminoso dependiendo de su diferencia de potencial con respecto al cátodo.
- 3.-Serie de ánodos (A).- Enfocan el haz hacia la pantalla.
- 4.-Placas de deflexión verticales (Dy) y horizontales (Dx).- Desvían el haz de electrones según la magnitud del voltaje aplicado a estas placas.
- 5.-Post acelerador (P).- Hace que la incidencia del haz sobre la pantalla se lleve a cabo a gran velocidad y se produzca luminiscencia.
- 6.-Pantalla.- Consiste en una superficie de vidrio sobre la que se aplica una capa de fósforo. Sobre el fósforo se vaporiza una delgada capa de metal que sirve como disipador del calor generado por la el haz incidente; si esta capa metálica no existiera el fósforo sería dañado por el haz de electrones incidente.

Con el fin de realizar las mediciones deseadas se coloca un cuadrícula que puede ser interna o externa dependiendo del grado de precisión que se requiera.

II.1.2.3.- Varios.

Atenuador y amplificador vertical.

Permite que señales desde pocos milivolts hasta cientos de volts sean desplegadas en la pantalla y produzcan deflexión vertical del haz por medio de las placas destinadas para este fin.

Amplificador horizontal.

Amplifica el voltaje de barrido a un valor que produce deflexión en el haz de electrones que se envía por el cañón del CRT en forma horizontal.

Circuito de disparo.

Este circuito produce un pulso cada vez que la señal de disparo alcanza cierto nivel; tiene como finalidad iniciar el barrido horizontal en el osciloscopio. (Generalmente se usa la señal de entrada como señal de disparo).

Circuito de barrido.

Se encarga de producir una rampa para controlar la placas de deflexión horizontal. La pendiente de esta rampa depende de la selección de tiempo/div. que se haga.

Pendiente y nivel.

Controla y selecciona el punto sobre el cual se inicia el despliegue de la señal.

Entrada de disparo externo.

Permite utilizar una señal diferente de la señal de entrada como señal de disparo.

Intensidad y foco.

Permite dar el brillo suficiente y enfocar el haz de electrones en forma adecuada para su observación en la pantalla.

II.1.3.-Características adicionales.

La estructura básica del osciloscopio puede modificarse aumentando ciertas funciones que permitan tener una mayor precisión y exactitud y además faciliten la realización de las mediciones. Algunas de estas funciones son:

Acoplamiento AC y DC

AC.- Elimina la presencia de las componentes de DC con respecto a tierra, con la desventaja de que las componentes de baja frecuencia de una señal no son reproducidas en forma fiel y además, se requiere de un intervalo de tiempo adicional para que el despliegue en pantalla alcance una posición vertical estable cuando se cambia el switch de DC a AC.

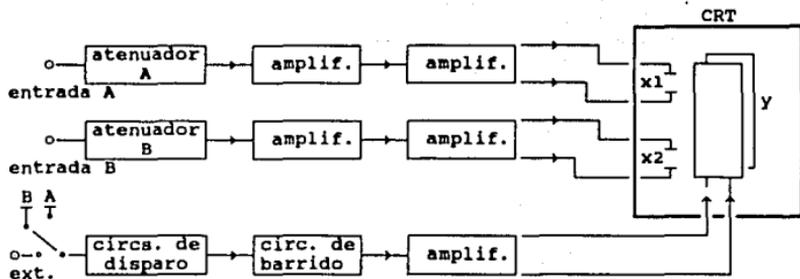
DC.- Permite el paso de las componentes de corriente directa al osciloscopio.

Doble haz y doble trazo

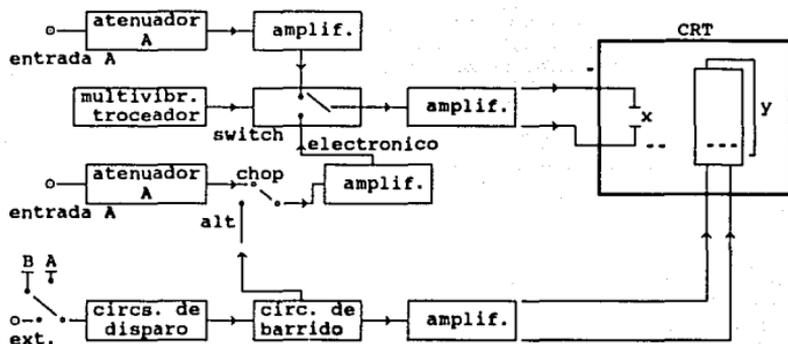
Para lograr el despliegue de dos señales diferentes en el osciloscopio (v.g. 2 canales) se usan básicamente dos técnicas:

Doble haz.- El CRT es equipado con dos cañones y dos pares de placas de deflexión vertical. Cada sistema de deflexión tiene su amplificador, y no existe multiplexaje de la señal.

Doble trazo.- Se logra multiplexando las señales al amplificador de control de deflexión vertical del haz. Este método involucra solo un cañón y un par de placas de deflexión, pero tiene como inconveniente que para coeficientes de tiempo menores de 0.1 ms/div el corte de los trazos comienza a ser visible.



a)



b)

Fig. II-3 a) Doble haz. b) Doble trazo.

Modo alternado (ALT) o modo trozado (CHOP)

Los osciloscopios de un cañón permiten estas dos modalidades para evitar los desperfectos (v.g. trozado) en la observación de las señales cuando se usan coeficientes de tiempo menores de 1 ms/div. Así pues la elección de cualquiera de estos modos depende del coeficiente de tiempo que se use :

Alternado.- Se hace el trazo de una de las señales a lo largo de toda la pantalla durante todo un barrido, en el siguiente barrido de pantalla se presenta la segunda señal, así alternativamente se muestran cada una de las señales.

Trozado.- En este modo se despliega un trozo de cada una de las señales durante un solo barrido alternativamente: una sección de la primera, una sección de la segunda, etc.

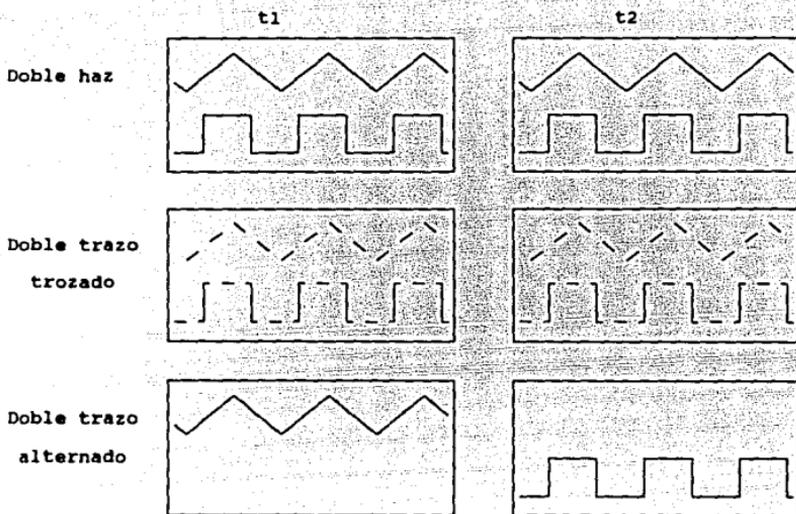


Fig. II-4 Efectos de doble haz y doble trazo.

Para coeficientes de tiempo mayores que 10 ms/div. el modo trozado visualiza la señal en forma aparentemente continua, mientras que en el modo alternado se observa el despliegue de una de las señales a la vez.

Para rangos de 10 a 0.1 ms/div se puede visualizar la señal correctamente con ambos modos, prefiriéndose el trozado pues el alternado produce un pequeño destello que fatiga la vista.

Para tiempos mayores de 0.1 ms/div se usa el modo alternado, por que el efecto de trozado se observa en pantalla.

Modo diferencial (A-B)

Este modo permite eliminar el voltaje común entre dos señales con respecto a tierra. Midiendo las señales en modo diferencial el voltaje común es rechazado y permanece solo la señal de interés.

La característica que refleja capacidad de un osciloscopio de realizar la diferencia de dos señales y rechazar el voltaje común se denomina CMRR (common mode rejection ratio), que se define en dB como:

$$CMRR = 20 \log_{10} (V_{com} / V_{dif})$$

donde

V com .- voltaje modo común.
V dif .- voltaje modo diferencial.

Línea de retardo

Es una línea de transmisión larga (o emulada por medio de circuitos) que se colocan en el recorrido de la señal de entrada hacia el osciloscopio, de manera que esta señal de entrada sufra un retardo.

El retardo de la señal de entrada tiene como objeto que la deflexión vertical del haz se inicie al mismo tiempo que la deflexión producida en las placas horizontales.

La razón de la diferencia de tiempo entre estas señales se debe a que el circuito de disparo es excitado cuando la señal de entrada alcanza cierto nivel, por lo tanto, si no existiera la línea de retardo, el intervalo de 0 V a voltaje de disparo en la señal de entrada no se visualizaría.

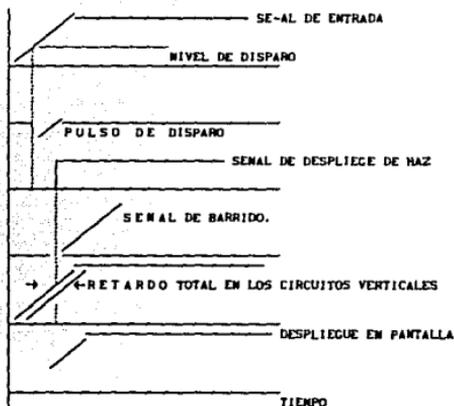


Fig. II-5 Efecto de la línea de retardo

Fuente de disparo

Puede adoptar las siguientes modalidades:

Interna: la señal de entrada realiza el disparo.

Externa: el disparo se realiza por una señal externa elegida por el usuario para este propósito. Se conecta a la terminal de TRIG EXT.

Línea : el disparo se realiza por las fluctuaciones de la línea de suministro de energía. Este modo se usa para determinar si algún tipo de amplitud modulada presente en la señal de entrada proviene de la línea de suministro.

Circuito de base de tiempo.

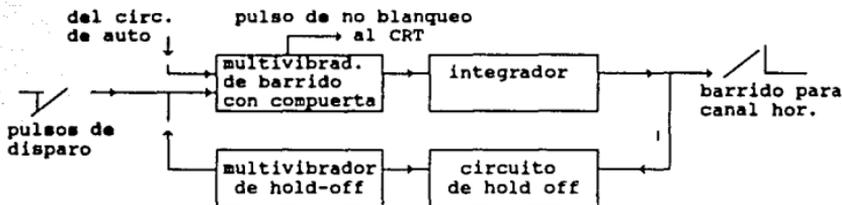


Fig. II-6 Diagrama de bloque del circuito de base de tiempo.

El multivibrador de barrido de compuerta es un disparador Schmit que genera pulsos cuadrados a partir de pulsos de disparo.

El integrador genera una rampa al integrar los pulsos, esta rampa se usa para realizar el barrido horizontal.

El circuito de "hold-off" (inhibidor de haz) sirve para suspender el haz de manera que al invertir la polaridad de las placas de deflexión horizontal no se observe su regreso en la pantalla (blanqueo).

El multivibrador de "hold-off" suspende el efecto de los pulsos de disparo durante el pulso de blanqueo de la pantalla.

Modos de disparo: STARTS y TRIG.

TRIG : El disparo se inicia después de cierto retardo de tiempo a partir del primer pulso, cuando se presenta un nuevo pulso (u otra señal) que pueda realizar el disparo. (Fig. II-7 a).

STARTS: El disparo de inicio de deflexión en las placas horizontales se origina en forma periódica con cierto retardo, a partir del primer pulso. (Fig II-7 b).

La modalidad STARTS producirá una visualización falsa de la señal si ésta no es perfectamente periódica como se indica en la siguiente figura:

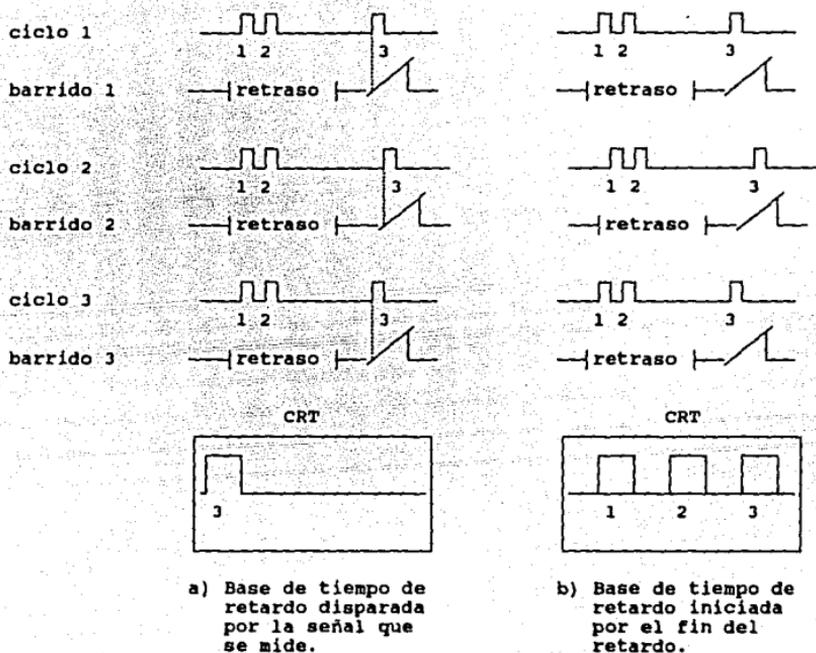


Fig. II-7

II.1.4.- Osciloscopio digital.

II.1.4.1.- Generalidades.

El osciloscopio digital se basa en el principio de muestreo que consiste en tomar muestras de una señal.

Debido a que es difícil desplegar directamente en la pantalla del osciloscopio señales de muy alta frecuencia (vhf; señales por arriba de 1000 MHz), se realiza un muestreo de la señal en diversos puntos, y se despliega el valor de tal señal en un instante de tiempo determinado. La forma de la señal de entrada será apreciada con mayor fidelidad dependiendo del número de muestras que se tomen; a mayor número de muestras mayor fidelidad.

Para este tipo de osciloscopio la señal debe ser repetitiva, así, una señal de disparo simple (que no se repite) no se puede desplegar.

Generalmente la tasa de muestreo del osciloscopio es menor que la tasa de repetición de la señal, de manera que una muestra se toma solo de vez en cuando (v.g. de la señal 1, 101, 201, 301, etc.). De hecho solo los circuitos de entrada deben manejar altas frecuencias, una vez que las muestras son tomadas y almacenadas, el resto de los circuitos pueden trabajar a bajas frecuencias.

El muestreo que se realiza puede ser de dos tipos:

- a) Aleatorio: no existe relación entre la rampa de voltaje de tiempo del osciloscopio y el instante de muestreo.

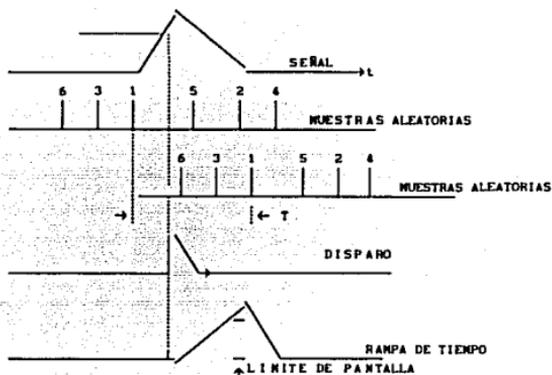


Fig. II-8a Muestreo aleatorio.

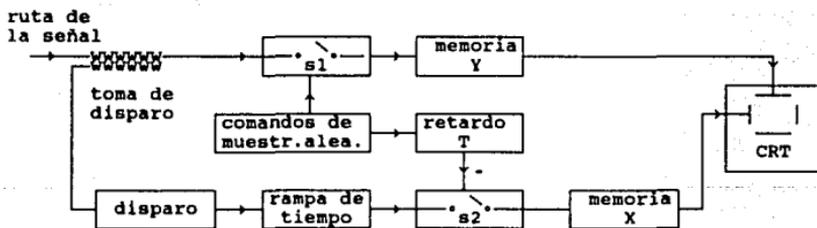


Fig. II-8b Diagrama de bloques de un osc. de muestreo aleatorio.

Durante cada muestra, Si se cierra y la amplitud de la señal se almacena en Y. El pulso de muestreo se retrasa un intervalo de tiempo T. Este pulso de muestreo retrasado se usa para muestrear la rampa de tiempo que se inicia por la señal de disparo.

b) Secuencial: las muestras se toman una después de otra en repeticiones sucesivas de la señal.

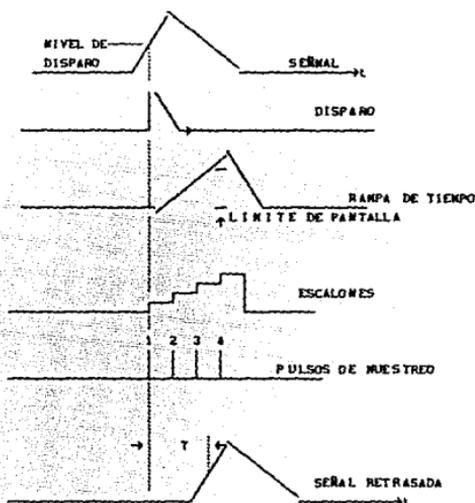


Fig. II-9a Muestreo secuencial.

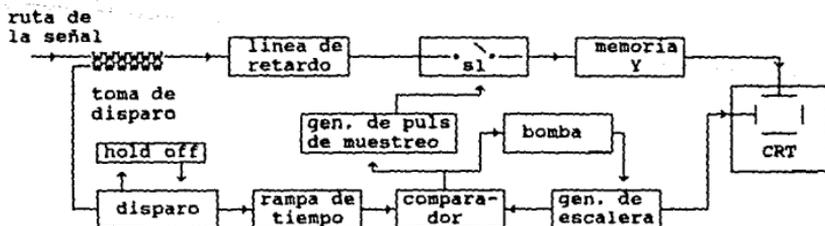


Fig. II-9b Digrama de bloque de un osc. de muestreo secuencial.

Se inicia la rampa de tiempo por un pulso. El comparador compara la rampa de voltaje con la salida del generador de escalera; cuando la rampa de voltaje iguala el voltaje del escalón, el comparador genera un pulso que va al generador de pulsos de muestreo y al excitador (bomba).

El pulso de muestreo abre S1 y se toma una muestra del voltaje de entrada y se almacena en la memoria Y de deflexión vertical. El excitador produce un salto del generador de escalera al siguiente escalón.

Este método tiene el inconveniente de requerir una línea de retardo, lo que puede producir distorsiones en la señal de entrada al ser reproducida a la salida.

El diagrama completo de un osciloscopio digital se muestra a continuación:

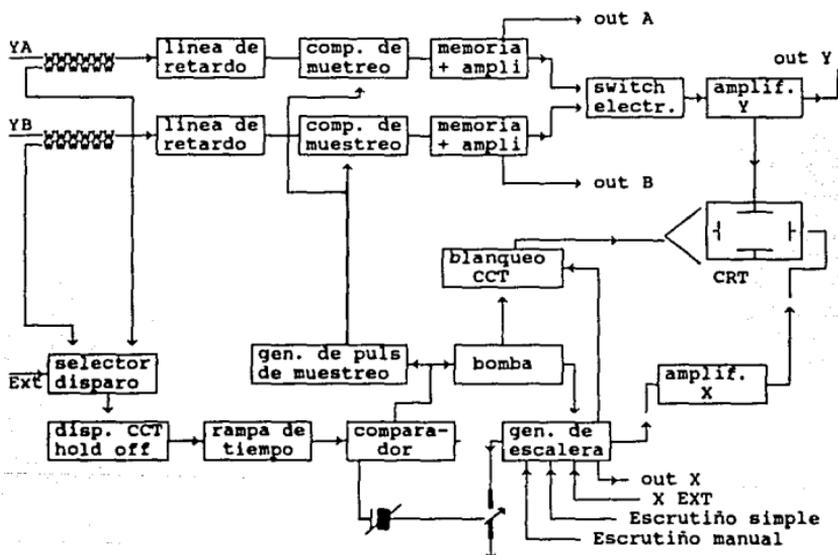


Fig. II-10 Diagrama de bloque de un osciloscopio digital.

Un aspecto importante de los osciloscopios digitales es la forma en como se hace la digitalización de la señal de entrada, básicamente existen 4 técnicas para hacer la digitalización:

- a) **Digitalizadores en tiempo real**, basados en convertidores analógico-digitales tipo flash, que codifican casi instantáneamente la salida de 64 comparadores o más en paralelo. Algunos pueden proveer largos registros, tanto como

32,000 niveles. La frecuencia de muestreo con estos convertidores puede llegar hasta a 1 Gigahertz, lo que permite rangos de frecuencia de disparo sencillo de hasta 250 Megahertz.

- b) Digitalizadores de tiempo equivalente de conversión, que construyen una señal a partir de muchas ocurrencias del evento de disparo, tomando pocos puntos digitalizados en un intervalo de tiempo en cierta ocurrencia, y combinándolos con los de la adquisición previa. Este tipo de digitalización puede dar un ancho de banda de hasta 1 Gigahertz, pero la señal de entrada debe ser repetitiva.
- c) Digitalizadores por convertidores de escudriñamiento, que digitalizan la señal de una manera indirecta. La señal se transmite a un tubo de rayos catódicos y se escudriña sobre el fósforo con un elemento de imagen como un CCD (Charge Coupled Devices). La salida es entonces procesada para obtener la versión digitalizada de la onda. Esta técnica permite tasas de muestreo de hasta 100 Gigamuestras por segundo, pero se generan registros de solo unos cientos de niveles y su costo es elevado.
- d) Digitalizadores fast-in, slow-out (FISO), estos usan CCD que funcionan como registros de corrimiento analógicos para muestrear las señales rápidamente y almacenarlas temporalmente. A la señal almacenada se le hace un corrimiento mas lento y se digitaliza a velocidad moderada. Estos digitalizadores alcanzan tasas de 40 a 400 Megamuestras por segundo.

II.1.4.2.- Principales parámetros de un osciloscopio digital.

- 1.- Procesamiento digital de señales en tiempo "real" o "no real". Se refiere a la posibilidad de hacer un muestreo y un procesamiento de forma que los cambios que se verifican en la señal de entrada se verifiquen simultáneamente en la pantalla de la PC (tiempo real), o de hacer el muestreo y almacenar la información para posterior procesamiento y despliegue (tiempo no real).
- 2.- Tipo de muestreo: aleatorio (probabilístico): Se hace mas frecuentemente en puntos en donde la probabilidad de que la pendiente de la señal aumente o secuencial que se hace uniformemente sin importar la magnitud de la pendiente de la señal.
- 3.- Nivel de disparo de generador de rampa de las placas horizontales.
- 4.- Rango de frecuencia. Considerado para el límite de la componente de frecuencia para la cual la respuesta del osciloscopio en amplitud a la salida es -3 dB la amplitud de la misma componente de frecuencia de entrada. Actualmente los osciloscopios digitales comerciales alcanzan rangos de hasta 1000 MHz con señales repetitivas.

- 5.- Tasa de muestreo. Actualmente se alcanzan tasas en osciloscopios comerciales de 1000 Mega muestras por segundo (MSS).
- 6.- Velocidad del registro de información.
- 7.- Sensitividad vertical. Es el mínimo voltaje en la entrada vertical que produce un deflexión del haz de una división.
En el caso de un osciloscopio digital está determinado por el número de bits del convertidor A/D que se usa.
- 8.- Acoplamiento AC o DC.
- 9.- Filtros de baja frecuencia, frecuencia media, o alta frecuencia.
- 10.- Posibilidad de disparo para inicio de barrido horizontal por medio de una señal externa.
- 11.- Implementación digital de línea de retardo (anticipo en el almacenamiento de información).
- 12.- Magnificadores de escala.
- 13.- Diversas escalas para tiempo y voltaje. Posibles modalidades: automática o manual.
- 14.- Número de canales. Los osciloscopios comerciales actualmente poseen hasta 8 canales. Convencionalmente tienen 2 canales.
- 15.- Funciones adicionales: multiplicación de dos señales, modo diferencial, suma de dos o mas señales, etc.
- 16.- Posibilidad de realizar el barrido con señales externas.
- 17.- Rango dinámico: diferencia entre la señal mínima y máxima que puede detectar el osciloscopio. Actualmente en osciloscopios comerciales la menor señal medible es de 0.01 mV, en general se obtienen mediciones mínimas de 5mV.
- 18.- Despliegue de señales en diferentes colores.
- 19.- Despliegue de menus para definir la modalidad en que se opera.
- 20.- Tiempo de subida del osciloscopio. Idealmente está entre 1/10 y 1/3 de la transición mas rápida. El producto ancho de banda por tiempo de subida debe ser aproximadamente 0.35 (v.g. 100 MHz, 3.5 nanoseg).
- 21.- Ancho de banda de disparo único (*single shot band width*). Es la componente de máxima frecuencia que se puede medir en la ocurrencia única de una señal (señales no repetitivas o de disparo simple "*single shot*"). En osciloscopios digitales comerciales se alcanzan rangos de hasta 250 MHz. Los valores típicos se encuentran entre 25 y 50 MHz.
- 22.- Tiempo equivalente de muestreo. Es el grado de muestreo que se logra haciendo muestreo sobre una señal repetitiva por medio de una tasa menor de muestreo (v.g. lograr tasas de 100 GHz por

medio de muestreo a 100 KHz).

- 23.- Resolución. Cambio mas pequeño en el valor medido para el cual el instrumento responderá.

En el mejor de los casos se tiene para el tiempo décimas de picrosegundo y para voltaje hasta 0.002% .

- 24.- Precisión (medida del grado con el cual mediciones sucesivas difieren una de otra). En osciloscopios comerciales se tiene hasta +- 1%.

- 25.- Tipo de digitalización. Puede ser: conversión en tiempo real, conversión en tiempo equivalente, conversión de escudriñamiento ("scan conversion") y entrada rápida salida lenta ("fast-in, slow-out").

- 26.- Impedancia de entrada. Típico en osciloscopios comerciales de 1 Mohm en paralelo con 25 pF.

- 27.- Diferencia de fase entre fase entre canal X-Y. Típico de 3 grados.

- 28.- Capacidad de memoria.

II.2 ANALIZADOR DE ESPECTROS

II.2.1 Trazos Espectrales

Un Osciloscopio de Rayos Catódicos (ORC) se emplea generalmente para trazar señales eléctricas con respecto al tiempo (en el "dominio del tiempo"). En el analizador de espectros las señales se descomponen en sus componentes de frecuencia las cuales se despliegan a lo largo del eje horizontal de un TRC, el cual está equilibrado en frecuencia, para presentar en el eje vertical la amplitud de la señal. Se dice que estas medidas están en el dominio de la frecuencia. Es importante hacer notar que se pueden analizar señales periódicas y no periódicas, sin embargo, en este último caso se obtienen lecturas que son sólo promedios de la señal que llega en el momento de hacer la lectura.

En la figura II-11a se muestra una representación tridimensional de una frecuencia fundamental (f_1) y su segunda armónica ($2f_1$) ilustrando las características en el dominio del tiempo y de la frecuencia. Se muestran, además, en la figura II-11b las dos señales trazadas por medio de líneas discontinuas, siendo la línea sólida la suma algebraica del valor instantáneo de las dos señales. Por último, la figura II-11c muestra las dos señales en el plano amplitud-frecuencia como se verían en el ORC y dos componentes de señales compuestas a medida que el analizador espectral barre el rango de frecuencia de la señal.

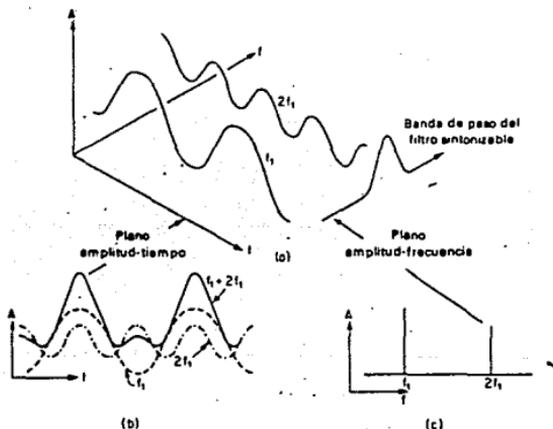


Figura II-11 Representación tridimensional de amplitud, frecuencia y tiempo. (a) La suma de la fundamental y su segundo armónico. (b) Vista en el plano t-A. En un osciloscopio únicamente la componente $f_1 + 2f_1$ se verá. (c) Vista en el plano f-A. Se ven claramente las componentes de la señal compuesta.

Es conveniente considerar el espectro de algunas señales comunes y trazos en el TRC que resultan cuando dos señales del tipo f_1 y $2f_1$ se aplican al analizador de espectros.

II.2.1.1 Señales de onda continua (CW).

Si el oscilador local del analizador barre a través de una señal de entrada CW lentamente, la respuesta resultante sobre la pantalla es simplemente un trazo de la banda de paso del amplificador de frecuencia intermedia (F.I.) Una señal CW pura, por definición, tiene energía únicamente a una frecuencia y por consiguiente debe aparecer como un simple impulso sobre la pantalla del TRC. Esto ocurrirá si el ancho de barrido de RF, o ancho del espectro, es comparable con el ancho de banda de F.I. en el analizador.

II.2.1.2 Modulación en amplitud.

Cuando una señal CW de frecuencia f_c se modula en amplitud por medio de una señal moduladora de frecuencia f_a , se generan bandas laterales a $f_c + f_a$ y $f_c - f_a$. El analizador desplegará la frecuencia portadora f_c , flanqueada por las dos frecuencias de las bandas laterales cuyas amplitudes relativas con respecto a la portadora dependen del porcentaje de modulación (ver figura II-12).

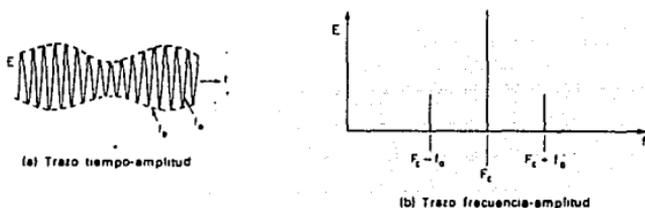


Figura II-12. Modulación en amplitud monótonica. (a) Trazo tiempo - amplitud. (b) Trazo frecuencia - amplitud.

Si la frecuencia, el ancho del espectro y la respuesta vertical del analizador están calibradas, el despliegue del TRC suministrará la siguiente información :

- frecuencia portadora
- frecuencia de modulación
- porcentaje de modulación
- modulación no lineal
- modulación de frecuencia incidental (se evidencia por las ondulaciones de las líneas espectrales)

II.2.1.3 Modulación en frecuencia.

Si una señal CW de frecuencia f_c es modulada en frecuencia a una razón f_r , producirá un número infinito de bandas laterales. Estas bandas están localizadas en los intervalos de $f_c + n f_r$, donde $n = 1, 2, 3, \dots$. En la práctica, únicamente se toman en cuenta las bandas laterales que contienen potencia significativa (Fig. II-13).

II.2.1.4 Modulación por pulsos.

Consideremos una forma de onda rectangular ideal, con tiempo de subida cero y sin sobresaltos ni otras alteraciones (ver fig.

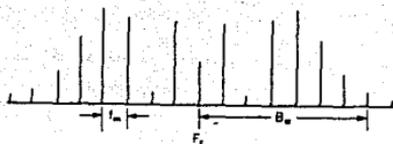
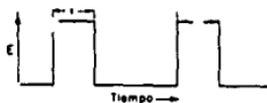
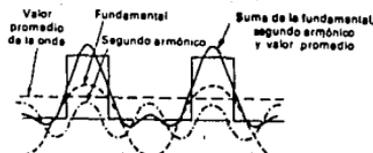


Figura II-13 Espectro de amplitud de una modulación de frecuencia monotónica.

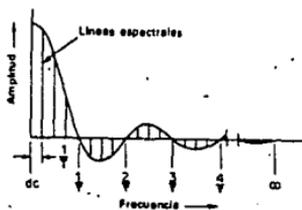
II-14a). Para analizar un pulso de este tipo en el dominio de la frecuencia, se debe descomponer (por medio de los algoritmos de la transformada de Fourier) en sus componentes individuales (Fig. II-14b). La suma de un voltaje constante, una frecuencia fundamental y sus armónicas formarán una onda que eventualmente llegará a ser una onda cuadrada a medida que las armónicas se sumen en fase con la fundamental. En el trazo espectral (mostrado en la figura II-14c) se tendrán entonces las amplitudes y fases de un número infinito de armónicas, dando como resultado una envolvente suave.



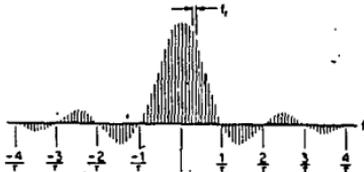
(a) Tren de pulsos rectangulares periódicos



(b) Suma de una onda seno (fundamental y sus armónicas) para formar pulsos rectangulares



(c) Espectro de un pulso rectangular perfecto. Las amplitudes y fases de un número infinito de armónicas se trazan resultando en una envolvente suave



(d) Espectro resultante de una portadora modulada en amplitud con un pulso rectangular

Figura II-14 Modulación con pulso.

Cuando este pulso se emplea para modular en amplitud una portadora, se efectúan las sumas y las diferencias de la portadora y todos los componentes armónicos contenidos en el pulso. Las armónicas producen bandas laterales múltiples de la misma manera que la señal modulante lo hace en la modulación de amplitud. Estas bandas laterales múltiples son llamadas líneas espectrales en el trazo del analizador. Habrá dos veces tantas bandas laterales o líneas laterales como frecuencias armónicas contenidas en el pulso modulante. Las líneas de trazo de frecuencia aparecen en el TRC espaciadas una cantidad igual a la

razón de repetición del pulso original. El lóbulo principal en el centro y los lóbulos laterales forman grupos de líneas espectrales que se extienden por encima y por debajo de la línea de base, sin embargo, por limitaciones físicas, aparecerán en el TRC sólo por encima de la línea de base. Para un pulso rectangular perfecto, el número de lóbulos laterales es infinito. El lóbulo principal contiene la frecuencia portadora representada por la línea más larga del centro (ver fig. II-14d).

II.2.2 El analizador de espectros

El analizador de espectros es un aparato electrónico que tiene como objetivo permitir al usuario realizar un análisis en frecuencia de una señal en base a una representación gráfica (llamada espectrograma) mostrado en la pantalla de un tubo de rayos catódicos, semejante al empleado por el osciloscopio.

Existen básicamente dos tipos de analizadores de espectros : los analógicos y los digitales. En este análisis nos ocuparemos de los primeros.

Los analizadores de espectros consisten básicamente de un receptor superheterodino de banda angosta sintonizado electrónicamente variando la frecuencia de un oscilador local y de un osciloscopio de rayos catódicos.

La figura II-15 muestra el diagrama de bloques básico de un analizador de espectros.

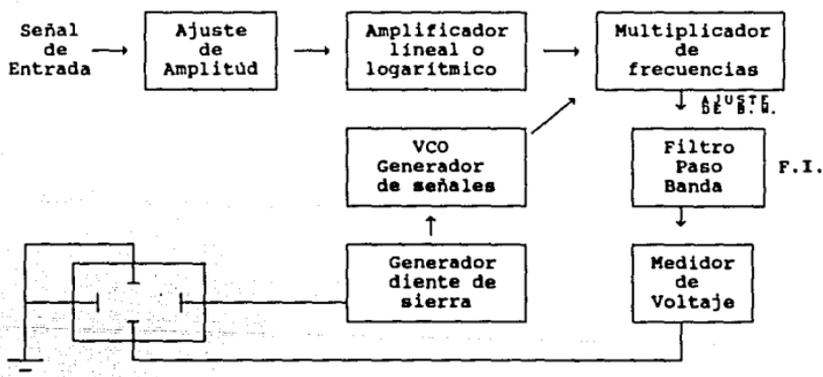


Figura II - 15

En su fase de entrada un analizador de espectros puede recibir señales con una amplitud que puede ir desde décimas de volt hasta varios volts (V_{rms}) y con una frecuencia que puede variar entre 1 Hz y una frecuencia máxima f_{max} kHz.

Una vez que se ha permitido el paso de la señal, esto es

cuando se ha verificado que no está por debajo de los niveles mínimos o más allá de los niveles máximos de frecuencia y voltaje, existe la posibilidad de ampliar su magnitud haciéndola pasar por un amplificador de tipo logaritmico que maneja una amplitud máxima M_x en dB (desde $-a$ dBV hasta a dBV).

La señal, amplificada o no, pasa por un multiplicador de frecuencias controlado por un generador de señales controlado por voltaje (VCO) que realiza un barrido sobre la señal a fin de poder filtrar, posteriormente, las secciones de la señal que se vayan analizando.

El VCO es la parte más importante del analizador de espectros pues con este podemos controlar el rango, en frecuencia, de barrido de la señal. Esto se refleja en el dominio del tiempo, cuando se aprecia la posibilidad de hacer el barrido desde un tiempo que puede ir desde decimas de segundo hasta cientos segundos dependiendo, entre otras cosas, de la precisión que se deseé. La señal producida por el VCO es la que se manda a las placas deflectoras horizontales del tubo de rayos catódicos, teniendo dicha señal la forma de un diente de sierra.

Conforme las diferentes secciones de la señal a ser analizada van siendo multiplicadas en frecuencia produciendo la frecuencia intermedia (F.I.) requerida, van pasando por un filtro paso-banda que tiene una frecuencia fija de F kHz con un ajuste de ancho de banda (BW) que puede ir de 1 Hz (alta precisión) a cientos de Hz (baja precisión). De esta forma se puede decir que una componente F.I. se produce únicamente cuando la componente correspondiente está presente en la señal de entrada. Las señales F.I. resultantes son amplificadas y detectadas y se aplican a las placas de deflexión vertical del TRC, produciendo un trazo de amplitud contra frecuencia

Notas técnicas adicionales :

- Se tiene un ajuste manual para indicar la frecuencia inicial de barrido.
- La escala vertical puede estar en dBV o en Volts.
- La pantalla muestra 10 divisiones horizontales por 10 divisiones verticales.
- La escala horizontal puede ir de 0 a f_{max} kHz, siendo la primera escala visible $f_{max}/1000$ Hz por cuadro.

II.2.2.1 Aplicaciones

El analizador de espectros permite observar directamente la amplitud de la señal, así como su distribución espectral; sus usos principales son :

+ Análisis espectral de varias formas de onda :

1. Análisis espectral de 2_2 pulsos.

2. Análisis de alta distorsión armónica de una señal de onda continua.
3. Análisis de distorsión por intermodulación debidas a componentes no lineales tales como un multiplicador de frecuencias y un modulador de amplitud.
4. Medición del ruido lateral en las cercanías de la señal.
5. Detección de oscilaciones parásitas de un transmisor.

+ Análisis espectral de señales moduladas.

1. Medición de la frecuencia de modulación, así como del factor de modulación de señales de A.M.
2. Medición de la frecuencia de modulación en una señal de P.M. y calibración del índice de modulación.
3. Medición de la frecuencia de repetición y la amplitud del pulso en una señal de modulación de pulsos.

+ Medición de fuerza de campo.

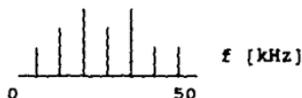
1. En conjunción con una antena de altura conocida, el analizador de espectros es capaz de medir la magnitud del campo.

+ Recepción panorámica.

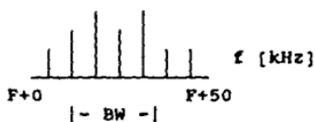
1. En unión con una antena, el analizador de espectros permite monitorear señales transmitidas así como de interferencia.

II.2.2.2 Ejemplo descriptivo

Para ejemplificar el proceso descrito anteriormente digamos que tenemos una señal de entrada que originalmente va desde 0 hasta 50 kHz.

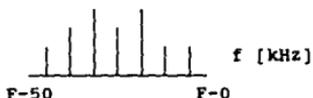


Conforme nuestra señal va llegando al multiplicador de frecuencias, las diferentes frecuencias que la componen sufren un corrimiento de F kHz.



Ahora bien, dependiendo del ancho de banda elegido se hará el filtrado de la señal. Si el ancho de banda (BW) es muy grande es posible que veamos una componente y a sus armónicas como un solo lóbulo de ancho considerable y no como diferentes espigas separadas. De esto se desprende que entre mayor precisión deseemos menor tendrá que ser el ancho de banda elegido.

Conforme se va realizando el filtrado la señal se va recorriendo en frecuencia, pero a diferencia de la primera vez que sufrió un traslado (hacia una frecuencia mayor) ahora lo va haciendo hacia una frecuencia menor de tal forma que el límite de frecuencia al que llegará la señal serán los F kHz.



Las secciones de la señal que van pasando por el filtro son medidas (su amplitud se mide en volts) y la señal resultante de cada medición es la que va a las placas deflectoras verticales del tubo de rayos catódicos.

Adicional al proceso descrito se puede tener un tratamiento digital de la señal (conversión analógica-digital, almacenamiento en memoria RAM y conversión digital-analógica) que le permite al usuario ver la señal tantas veces como sea necesario.

II.2.3 Características básicas de un analizador de espectros

+ Rango de frecuencia.

Entre mayor sea el rango de frecuencias que puede manejar el analizador, más conveniente será su uso. El tipo universal tiene un rango de frecuencia superior a 10^4 en banda específica. El rango de frecuencias que pueden ser medidas está restringido por el rango de operación del multiplicador de frecuencias, así como por el rango variable del oscilador.

+ Ancho de la frecuencia de rastreo.

Este término se refiere al ancho de banda que se puede monitorear mientras se hace un barrido en la pantalla del TRC. El término "espacio de frecuencia" se aplica, en este contexto, de diversas formas, una de ellas es "espacios/división" o sea el ancho del barrido por división de la pantalla del TRC, y la otra "espacio", es decir, el valor total mostrado en la pantalla. El término "barrido total" se usa para expresar el espacio de frecuencias que se forma con el rango completo de frecuencias antes mencionado, mientras que el "barrido de banda" se refiere a las bandas en que el rango de frecuencias está dividido. En ocasiones se expresa el espacio de frecuencias en términos de dos frecuencias - frecuencia de inicio y frecuencia final. Es necesario que el espacio de frecuencia pueda estar lo más cercano posible a los límites establecidos por el análisis espectral. Tanto el barrido total como el barrido de banda son útiles cuando se está buscando una señal de frecuencia desconocida.

+ Resolución de frecuencia

Cuando dos señales adyacentes se presentan en el TRC, la capacidad para analizar cada una de dichas señales es llamada resolución. En un caso como el expuesto anteriormente, si ambas señales son del mismo nivel, la resolución se representa por 3dB en el ancho de banda de la FI, ²⁵ y si la diferencia es grande, se

representa por el límite característico de la FI. Además, la resolución se ve influenciada por la FM residual del VCO, la pureza del espectro (ruido de banda lateral) y por la velocidad de muestreo. Estos efectos se muestran en las figuras II-17 y II-18. En la figura II-17a, si 3dB y 60 dB en la banda de F.I. son representados mediante Δf 3dB y Δf 60 dB, entonces el comportamiento característico está representado por la razón de cambio $(\Delta f \text{ 60 dB})/(\Delta f \text{ 3 dB})$. La figura II-17b ilustra el caso en el que las dos señales tienen el mismo nivel, y el "hueco" del centro es causado por la cancelación de las fases de las dos señales. La figura II-17c muestra un ejemplo en el que las dos señales tienen un nivel de diferencia de 60 dB.

El Filtro Paso Banda (BPF) para FI usado comunmente tiene una característica Gaussiana como la mostrada en la figura II-17a.

La figura II-17d muestra la influencia de la Frecuencia Modulada (F.M.) residual del analizador de espectros, deteriorando la resolución a partir de Δf_{ms} . La figura II-17e muestra el efecto del ruido de banda lateral, esto es, una aparente expansión del comportamiento característico de la F.I.

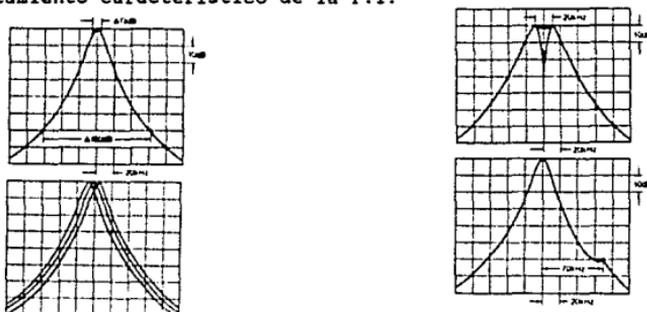


Figura II-17 Resolución de frecuencia. (a) $\Delta f_{3dB} = 10$ kHz. (b) Despliegue de dos señales de amplitud igual. (c) Dos señales de 60 dB con diferencia en amplitud. (d) Un ejemplo de una FM residual grande.

El espectro que sería desplegado en el TRC se muestra en la figura II-18a dependiendo de la velocidad de muestreo. En el caso de una velocidad de muestreo apropiada, aparecerá la forma de onda A, incrementando la velocidad de muestreo, las formas de onda resultantes serán B y C. Si la velocidad de muestreo es muy rápida, la amplitud decrecerá, el ancho de banda aparente se incrementará y la frecuencia se desviará. En la figura II-18b se muestra la relación de la velocidad de muestreo tanto contra la pérdida de amplitud como contra el ancho de banda aparente.

+ Nivel promedio de ruido y sensibilidad.

El nivel de referencia promedio hace referencia al valor medio del nivel interno de ruido debido al ruido térmico del analizador de espectros. Esto es proporcional al ancho de banda de la F.I., de tal forma que el nivel mínimo apreciable (la sensibilidad máxima) está determinado por el ancho de banda de la F.I. más cercana. La figura II-19 muestra la relación entre el ancho de banda de la F.I. y el nivel de ruido térmico.

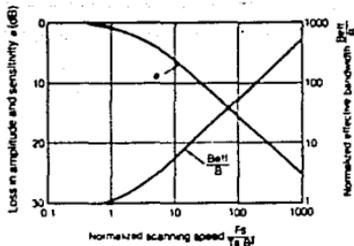
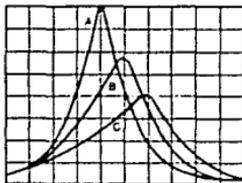


Figura 11-18 Velocidad de Muestreo y despliegue del espectro. (a) Velocidad de muestreo y resolución. (b) Pérdida de sensibilidad y ancho de banda efectivo normalizado contra velocidad normalizada de muestreo. F_s = Tasa de barrido, T_s = Tiempo de barrido, $B = 3$ dB del ancho de banda de IF. B_{eff} = Ancho de banda efectivo.

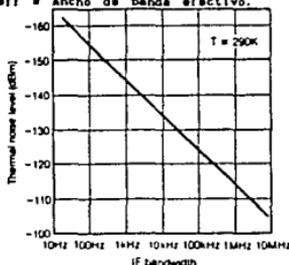


Figura 11-19. Ruido térmico contra Ancho de banda de IF

+ Rango dinámico

El rango dinámico de un analizador de espectros se define por el rango que es capaz de medir (la diferencia mayor de valor) para dos señales de diferentes niveles que se aplican simultáneamente a la entrada cuando se está midiendo con una precisión conocida. Esto difiere tanto de el "rango dinámico desplegado del TRC" el cual se puede encontrar utilizando la escala del eje de amplitud, como de el "mayor rango dinámico utilizable" (nivel de rango de medición) el cual se expresa por la diferencia entre la señal mayor de entrada y el nivel medio de ruido. El factor dinámico se determina por factores tales como el nivel promedio de ruido, la distorsión de la 2a. y 3a. armónicas y las respuestas residual y la proveniente de fuentes diferentes a la original. La primera se genera por la interferencia entre las salidas de dos osciladores locales, aún cuando no exista señal de entrada. La segunda se genera en los multiplicadores y amplificadores del analizador de espectros cuando se tiene una señal almacenada.

+ Rango de nivel de medición.

El nivel inferior del rango de nivel de medición está determinado por el nivel promedio de ruido, y el nivel superior está determinado por el nivel en el que se asegura una ganancia lineal (el nivel de entrada al cual la ganancia es de 1 dB) o por el funcionamiento del atenuador de RF.

+ Razón de tiempo real.

Esta expresa la habilidad del analizador para registrar variaciones en la señal de entrada. Si el analizador puede procesar los datos tan rápido como estos llegan, se dice que está operando en "tiempo real". En forma alternativa, la tasa de tiempo real puede ser llamada ancho de banda de tiempo real, refiriéndose a la frecuencia más alta a la cual la razón de proceso de datos iguala a la tasa de captura de datos. La razón de tiempo real es el producto de la velocidad a la que se actualizan los datos en el analizador, multiplicada por el número de elementos (líneas) en las que el analizador divide el espectro de frecuencias desplegado. Por ejemplo, un instrumento de 400 líneas, actualizándose 25 veces por segundo, tendrá una razón de tiempo real de 10 kHz. Las aplicaciones que son relativamente invariantes con el tiempo no requieren de altas tasas de tiempo real.

+ Almacenamiento en memoria.

Quando el trazo derivado de los datos se puede almacenar en memoria y se puede llamar en cualquier momento, el usuario dispone de medios convenientes para comparar la señal almacenada con otra en estudio. Tales comparaciones son útiles para poder determinar, e incluso anticipar, cambios en el desempeño de las funciones del equipo o bien para comparar dicho desempeño contra normas establecidas.

+ Zoom de frecuencia.

Aplicable para una resolución fina sobre bandas muy cercanas.

+ Despliegue de frecuencia con intervalos de 3/8 o de 1/8.

Diferentes modos de despliegue para mediciones de tipo acústico.

+ Procesamiento de datos con o sin promedios.

Se puede utilizar tanto en el dominio del tiempo como en el de la frecuencia para señales estables, dinámicas o transitorias.

+ Función de transferencia.

Este es un cálculo complejo (conteniendo partes reales e imaginarias) que normaliza el espectro de salida de acuerdo al espectro de entrada caracterizando al sistema en estudio de tal forma independiente al nivel de la señal de entrada.

+ Coherencia en la información.

Nos dice en que medida un espectro de salida se debe a un espectro de entrada dado. Esto es útil cuando se intenta aislar la fuente de una señal dada de entre varias en situaciones en las que ninguna de las fuentes adicionales puede ser cesada. Los valores de coherencia van desde cero (una falta completa de relación causal) hasta uno (respuesta debida totalmente al impulso aplicado).

III.1 ARQUITECTURA GENERAL DE LA PC

En el presente capítulo se dará una breve descripción de la arquitectura general de la PC, así como una explicación del sistema.

Los componentes básicos de una computadora personal consisten en: la Unidad del Sistema, el teclado, el despliegue monocromático, y el adaptador para impresora.

III.1.1 Unidad del Sistema:

La importancia primordial de entender esta unidad estriba en que la mayoría de las aplicaciones en interfases requieren utilizar las ventajas y capacidades presentes en esta tarjeta.

El concepto Unidad del Sistema se refiere a la caja que contiene casi en su totalidad a la computadora, es decir, consta de un microprocesador de 16 bits, las memorias RAM y ROM, un manejador de discos flexibles, una bocina con su respectivo manejador para aplicaciones de audio, de forma estándar contiene también una serie de ranuras o canales de conexión para puertos que permiten la expansión del sistema y por último contiene un manejador para almacenamiento de datos en fuentes distintas al disco flexible. Está polarizado por una línea convencional de 110 volts, que ya regulada, entrega 65 W de potencia. En la figura III-1 se muestra un esquema de componentes de la computadora IBM-PC modelo XT.

La parte primordial de esta unidad es la tarjeta de sistema o tarjeta madre, la cual contiene la gran mayoría de las partes electrónicas de la PC, entre ellas los elementos antes mencionados a excepción de la fuente de poder, la cual es independiente.

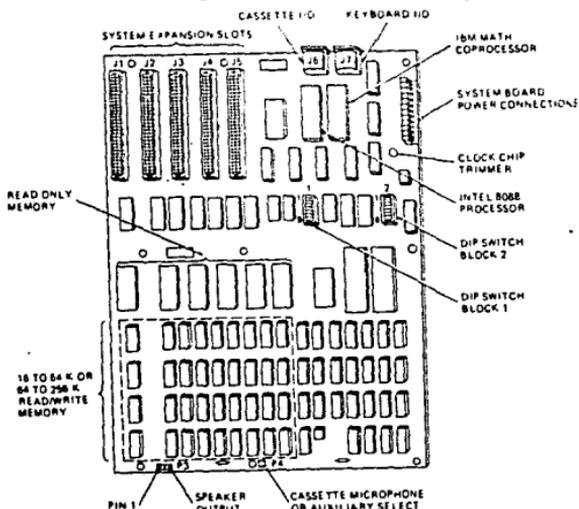


Figura III-1. Diagrama de distribución de los componentes en la tarjeta madre de la PC.

En el sistema IBM PC y compatibles el microprocesador empleado es un 8088 de 16 bits, la RAM es de hasta 1 MByte, la ROM de 40 KBytes y los adaptadores I/O son los que controlan los periféricos.

Los canales de conexión o *slots* son la clave del éxito del diseño de la PC, ya que la convierte en una computadora abierta y versátil a la que se le puede conectar fácilmente sistemas electrónicos según las necesidades específicas del usuario.

Si la tarjeta del sistema es el corazón de la máquina, el microprocesador 8088 es el centro de la tarjeta. Este microprocesador surgió como consecuencia de la necesidad de tener un dispositivo que pudiera unificar los sistemas digitales de proceso de datos existentes en el mercado creando un nuevo diseño de computadora. El primero de esta serie fue el 8086, utilizando ciclos de *pre-fetch* con cola de 6 bytes de longitud y canal de datos de 16 bits. Esto ponía en problemas a los diseñadores de sistemas del momento, ya que el canal de datos estandar era de 8 bits, por lo que se tenía que crear un microprocesador que internamente manejara 16 bits, pero acoplándose a los requerimientos del mercado; de allí surgió el 8088.

A continuación se enumeran las características principales del 8088 :

- + Arquitectura interna de 16 bits.
- + Capacidad de acceso de 1 MegaByte de memoria.
- + Aritmética de 8 y 16 bits signada y no signada en decimal y binario, incluyendo multiplicación y división.
- + 14 registros de 16 bits.
- + Capacidades de interrupción mascarables y no mascarables.
- + Modos de direccionamiento con 25 modalidades.
- + Capacidad de acceso directo a memoria.
- + Puede soportar coprocesadores en el canal local.
- + Soporta tanto mapeo I/O de memoria como mapeo de puertos de I/O
- + Operaciones con cadenas de caracteres (*Strings*).
- + Es del tipo *pipeline*.
- + Maneja procesamiento en paralelo mediante sus unidades de Ejecución y de Interfaz para bus.

Puede acceder hasta un megabyte de memoria debido a que utiliza 20 bits para direccionamiento, conteniendo datos o programas, y soporta las funciones de interrupción y acceso directo a memoria.

Refiriendonos a los parámetros de tiempo, diremos que como un ciclo de reloj es de alrededor de 210 ns cuando el CPU opera con un reloj de 4.77 MHz, y ya que el mínimo ciclo de lectura/escritura del canal consiste de cuatro períodos de reloj del CPU, un ciclo de canal o bus consiste de 840 ns. Se entiende como "ciclo de bus" al tiempo que transcurre para que se haga una lectura de datos de una unidad específica de memoria y se manden estos datos al adaptador de interfaz o viceversa.

En la figura III-2 se presenta la tarjeta principal de la microcomputadora o "tarjeta madre". Se aprecian las áreas destinadas a memoria RAM, ROM y el microprocesador, también se aprecian las ranuras de expansión del sistema. Estos *slots* permiten conectar tarjetas de uso específico adicionales a la tarjeta básica del CPU. Por ejemplo, se pueden agregar tarjetas controladoras de disco, tarjetas para puerto serie, tarjetas para gráficas de alta

resolución, tarjetas para memoria adicional y, lo que nos concierne directamente, tarjetas de aplicación diseñadas por el usuario, por ejemplo una tarjeta que permita hacer análisis de señales haciendo conversiones A/D y D/A. Este sistema "abierto" permite adaptar el sistema a las necesidades del usuario.

El diagrama de bloques de la tarjeta madre de la IBM PC, aplicable también a las computadoras compatibles con ella, (Fig. III-2) nos muestra en primera instancia el CPU 8088, el conector para un procesador auxiliar (por lo general se emplea el coprocesador numérico 8087) y el controlador de prioridad de interrupciones 8259A. A continuación aparecen los buffers del bus de direcciones, los buffers del bus de datos y el chip controlador de puertos 8288. Este último es necesario debido a que el μ P 8088 está operando en modo máximo (concepto que se explicará más adelante). Los buses se conectan a los conectores de las tarjetas de 62 terminales ("pines") para periféricos. El CPU puede entonces usar estos buses para comunicarse directamente con las tarjetas que están conectadas a los slots de expansión. Se muestra también el área de ROM que consiste de 6 bloques de memoria de 8 K por 8 y en donde está residente el BIOS, la RAM dinámica que consta de 4 bloques que pueden ser de 16 K por 9 o bien de 64 K por 9, la lógica de control de Lectura/Escritura a memoria, los multiplexores 74LS158 empleados para manejar las direcciones de memoria durante las operaciones de Lectura/Escritura, la lógica del teclado, la lógica de la bocina, la lógica del manejador de cassette y las líneas de alimentación y tierra. Por otra parte se pueden apreciar un circuito 8253-5 que es un timer programable, un 8255A-5 que es un circuito manejador de puertos programable. Ahora sólo quedan por mencionar los tres dispositivos que tienen acceso directo a memoria (DMA), el 8237A-5 es el controlador de DMA, el latch octal 74LS373 se usa para tomar los 8 bits más significativos de la dirección de DMA enviados al bus de datos por el 8237A-5 durante una transferencia. El registro de corrimiento 74LS670 se usa para sacar los bits A16-A19 de la dirección de transferencia de DMA.

A continuación se explican en forma más explícita algunos de los componentes antes mencionados :

El Coprocesador 8087

Este aumenta el juego de instrucciones del 8086/8088, mejorando su capacidad de tratamiento de números. Se utiliza como procesador paralelo aunque dependiente (esclavo) del 8088. Utiliza su propia cola de instrucciones para controlar el flujo de instrucciones del 8088, ejecutando sólo aquellas que le corresponden, e ignorando las destinadas al CPU maestro. Requiere de la misma estructura de canales, la misma alimentación y el mismo tipo de sincronización. Sus instrucciones incluyen un juego completo de funciones aritméticas así como un potente núcleo de funciones exponenciales, logarítmicas y trigonométricas. Utiliza un formato interno de números en punto flotante de 80 bits con el cual puede trabajar con siete formatos (tipos) de datos distintos; enteros de tres longitudes distintas, un tipo con signo, y tres tipos de representación con punto flotante.

Las rutinas de tratamiento de números en punto flotante se pueden realizar a velocidades muy altas si se utiliza el 8087 como ampliación

del 8086/8088.

Si bien este coprocesador acelera las operaciones matemáticas, en el caso del procesamiento de señales se necesita manejar una gran cantidad de datos y una gran cantidad de operaciones. Además el 8087 se ve limitado por la capacidad del sistema IBM PC para realizar accesos de I/O y a la cantidad de memoria del sistema. Es por ello que se considera que un Sistema de Adquisición de Datos como el que se trata en este trabajo no podría ser desarrollado adecuadamente en base a este coprocesador.

Dispositivos de Soporte

Existen una serie de chips que proveen al 8088 circuitería para tres tipos de funciones : lógica para generación de pulsos de reloj, lógica para la interfase de canal y controladores. Estos dispositivos realizan funciones vitales para el sistema, como la sincronización, la conexión de los procesadores con el resto del sistema y la conexión de la computadora con el mundo exterior.

Lógica de generador de pulsos de reloj

Cualquier sistema basado en 8088 requiere una lógica adicional encargada de generar las señales de sincronización para todo el sistema. Esto se logra en el sistema IBM PC mediante el generador de pulsos de reloj 8284 de Intel, que trabaja con un cristal oscilador externo.

El sistema puede operar a una velocidad de reloj de 5 MHz. La velocidad de la IBM PC es de 4.77 MHz. Esta frecuencia es obtenida del cristal oscilador de 14.31818 MHz. El cual es dividido por tres por el generador de reloj 8284A. La frecuencia del cristal es dividida por cuatro para obtener la señal de color de 3.587 MHz para televisiones a color.

El 8284 es un chip con 18 terminales que se emplea para generar los pulsos de reloj. Estos determinan la velocidad de funcionamiento del sistema. La máxima velocidad estandar para estos procesadores es, como antes se mencionó, de 5 MHz, aunque algunos casos se manejan velocidades de 8, 10,16 e incluso de 25 MHz. A lo largo de este estudio nos referiremos a sistemas con velocidades de reloj de 5 MHz.

Las señales que normalmente salen del generador de reloj hacia el procesador son CLK, RESET y READY.

La función READY es importante en nuestra aplicación de interfases debido a que ayuda a sincronizar al procesador con los dispositivos externos más lentos. Va desde el dispositivo externo al procesador, pasando a través del generador de pulsos de reloj. Cuando el procesador quiere acceder a un dispositivo que no está preparado para la transferencia, este envía un 0 lógico por línea, de esta manera el procesador esperará hasta que aparezca un 1 lógico para continuar.

Lógica de Interfaz del Canal

Esta lógica es necesaria por dos razones: La primera es que las señales de los procesadores pueden no ser lo suficientemente potentes para controlar al resto del sistema y la segunda es que las señales producidas por los procesadores puede que no correspondan directamente a las señales que requiere el resto del sistema.

El controlador de Canal 8288, el Transceptor de Datos Octal 8286 y el Latch Octal 8282 se utilizan para solventar estos problemas de interfaz en un sistema 8086/8088 en modo máximo. Además, el selector de canal 8289 se utiliza como interfaz de los grupos de procesadores con el canal principal del sistema.

Estos tres componentes resuelven el problema de exceso de señales e incluyen toda la lógica de control necesaria. Un canal del sistema tiene cuatro subcanales: de alimentación, control, dirección y datos. Descartando el primero, cada uno de los componentes citados realizan la labor de interfaz de los elementos del sistema con cada uno de los subcanales.

El controlador de Canal 8288

Es un componente de 20 terminales encargado de realizar la interfaz entre el procesador y el canal de control. Decodifica las señales S0, S1 y S2 del 8086 en modo máximo y genera un conjunto completo de señales de control, tales como la de lectura en memoria (MRDC), control de lectura de I/O (IORC), control de escritura en memoria (MWTC), control de escritura de I/O (IOWC), latch de direcciones disponible (ALE) y datos disponibles (DEN). Algunas de estas señales de control, como las de lectura y escritura, tienen como destino el canal del sistema, mientras que otras, tales como las direcciones o datos disponibles, son señales destinadas a los otros componentes de interfaz del procesador con los otros subcanales.

Controladores de Dispositivos y del Sistema

Los controladores del sistema se encargan de controlar automáticamente funciones especiales de éste como la transferencia rápida de bloques de datos ó la coordinación de varias interrupciones. El controlador programable de acceso directo a memoria (DMA) realiza la función de transferencia rápida, y el controlador programable de interrupciones 8259 que, como su nombre lo indica, controla las interrupciones.

Los controladores de dispositivos funcionan como una interfaz inteligente con los dispositivos externos como unidades de disco, teclados e impresoras. Se pueden dividir en dos grupos: los controladores de dispositivos de propósito general y los de propósito especial. Dentro de los primeros se contemplan: el 8251 (controlador programable serie de interfaz), el 8255 (controlador programable paralelo de interfaz). Dentro de los controladores de propósito especial están el controlador programable de CRT (8275) y el controlador de discos flexibles de densidad simple/doble (8272).

Controlador Programable de DMA 8237

Es un dispositivo capaz de controlar las transferencias directas de información de una parte a otra del sistema. Dichas transferencias son importantes por cuanto muchas veces es necesario mover bloques de datos muy rápidamente, algunas veces a velocidades mayores que las que se podrían conseguir moviendo los datos byte a byte a través del CPU.

En una PC, la lógica del DMA incluye el controlador 8237-A, el muelle de dirección 74LS373, el registro de página 74LS670 y el reloj.

El 8237-A tiene cuatro canales independientes, permitiendo transferencias de datos entre la memoria y dispositivos externos. Los

canales del 1 al 3 son reservados como canales de I/O, mientras que el canal 0 es utilizado para refrescar la memoria. El canal 2 es utilizado para transferir datos del controlador de disco flexible a la memoria. Tiene 16 líneas de dirección, de los cuales A8-A15 son multiplexadas con las señales D0-D7.

Ya que el controlador DMA solo tiene 16 líneas de dirección, el 74ls670 es utilizado como registro de página, el cual proporciona los cuatro bits más significativos, de los 20 bits de dirección para poder acceder un Megabyte de memoria.

El reloj requerido por el 8237-A deberá ser como mínimo de 80 ns, ya que el reloj del CPU es de 70 ns.

Controlador Programable de Interrupciones (PIC) 8259-A

Este se utiliza en la gestión directa de las interrupciones de hasta ocho dispositivos distintos, y de hasta 64 dispositivos si se conectan ocho PICs 8259 juntos. En una modalidad, está preparado para trabajar con el 8085, y en la otra puede hacerlo con el 8088.

Actúa como recepcionista o árbitro en la comunicación de diversos dispositivos con el CPU. Su función es la de hacer pasar una interrupción de dichos dispositivos a la vez, contemplando ciertas prioridades, dadas anteriormente por un byte enviado por el CPU y llamado máscara de interrupciones, a través de un puerto situado en el espacio de I/O; dicho puerto recibe el nombre de puerto de control. Los ocho bits de la máscara corresponden a los ocho dispositivos.

Existe una línea de interrupción que va del PIC a la línea INTR (interrupt request) del CPU. Cuando éste recibe una petición por esta línea, contesta con un INTA (interrupt acknowledge). Posteriormente hablaremos más a fondo respecto a este particular en la sección III-4.

Controlador Programable Paralelo de Interfaz (PPIC) 8255

El PPIC 8255 sirve de ayuda en la conexión a la computadora de dispositivos que envían bytes completos cada vez. La transmisión paralela es útil en todas aquellas aplicaciones que requieran transmisiones a gran velocidad y utilicen dispositivos no demasiado alejados de la computadora. No existe ninguna sincronización especial en las transmisiones paralelas. Los bytes se envían tan rápidamente, o tan lentamente, como lo permita el software. La velocidad máxima de transmisión se ve afectada por la rapidez con que el sistema pueda sacar los datos. La mayor velocidad se obtiene utilizando el DMA.

Lógica del Teclado

El dato formado sobre el teclado es sacado de un registro de corrimiento el cual convierte la entrada serie de datos en una salida en paralelo. El formato de transmisión de datos serial utiliza dos bits de inicio y ocho bits de información, con uno de alto. Su velocidad de transmisión es de 9600 bauds.

El teclado proporciona una señal de reloj al registro de corrimiento. Está definido de tal forma que tiene la máxima flexibilidad para definir la operación del teclado, como por ejemplo en estados shift de las teclas y la operación type-matic. El teclado genera los códigos de barrido, en lugar de los ASCII. Todas las teclas excepto NUMLOCK y CAPLOCK son type-matic y generan un código break y

uno make. El último aparece cuando se oprime una tecla y el primero cuando ésta deja de oprimirse.

Interfaz de disco flexible

La interfaz del drive, está centrada en el controlador UPD765, utilizando un reloj cuya frecuencia es de 8 MHz, la cual entra a un contador de 4 bits para ser dividida en 8, 4, 2 y 1 MHz. El controlador DMA 8237-A forma toda la transferencia de datos, entre el controlador de disco flexible y la memoria del sistema.

III.2 EL MICROPROCESADOR 8086 Y EL 8088

El microprocesador (μP) 8086 es un procesador de propósito general de 16 bits, este término quiere decir que la Unidad Aritmético-Lógica (ALU), sus registros internos, y la mayoría de sus instrucciones están diseñadas para trabajar con palabras binarias de 16 bits. El 8086 tiene un canal (*bus*) de datos de 16 bits, de tal manera que puede leer o escribir datos a memoria o puertos de 16 bits por medio de un solo acceso al canal, o por grupos de 8 bits. Este microprocesador tiene un bus de direcciones de 20 bits, de tal forma que puede direccionar hasta 2^{20} (1,048,576) localidades de memoria, cada una con una magnitud de un byte (8 bits). Las palabras se guardan en dos localidades de memoria consecutivas. Si el primer byte de una palabra tiene una dirección non, el 8086 puede leer toda la palabra en una sola operación. En caso contrario, leerá el primer byte en una operación y el segundo en otra.

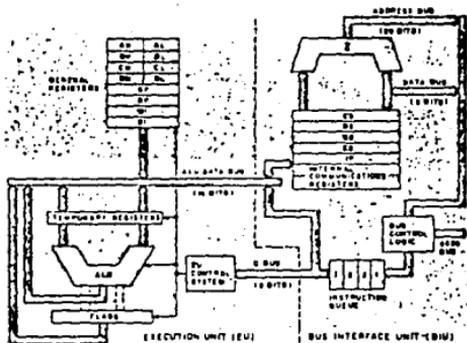


Figura III-3. Diagrama de bloques de las dos unidades de proceso del 8086.

El microprocesador 8088 de Intel tiene la misma ALU, los mismos registros y el mismo conjunto de instrucciones que el 8086. Así mismo, también tiene un bus de direcciones de 20 bits pero su bus de datos es de solo 8 bits, de tal forma que puede leer o escribir datos a memoria o puertos en grupos de 8 bits. Para leer una palabra de 16 bits de dos localidades contiguas de memoria, el 8088 siempre tiene que realizar dos operaciones de lectura. Debido a que el 8086 y el 8088 son casi idénticos, casi cualquier referencia que se haga al 8086 puede ser extensiva al 8088, el cual, por cierto, es el que se emplea como el CPU maestro de la computadora personal IBM PC y compatibles.

III.2.1 Arquitectura del 8086

Como se muestra en el diagrama de bloques de la figura III-3, este CPU está dividido para dos partes funcionales independientes, la Unidad de Interfase para Bus (BIU), y la Unidad de Ejecución (EU). Al dividirse el trabajo entre estas dos secciones, se aumenta la velocidad del procesador.

Intel diseñó estos microprocesadores para realizar al mismo tiempo las principales funciones internas de transferencia de datos y búsqueda de intrucciones. Para conseguirlo, constan de dos procesadores interconectados en el mismo componente. Una unidad está encargada de buscar instrucciones y la otra de ejecutarlas. Además, la unidad encargada de buscar instrucciones utiliza un método de estructura tubular o por cola para almacenar nuevas instrucciones hasta que se requieran (*prefetch*).

Al procesador principal se le llama unidad de ejecución (EU). Está encargado de codificar y ejecutar todas las instrucciones. La EU de ambos micros es idéntica. Al otro procesador se le llama la Unidad de Interfaz de Canal (BIU: *Bus Interface Unit*). Esta se encarga de localizar las instrucciones y de transferir todos los datos entre los registros y el mundo exterior. La del 8088 es más compleja que la del 8086, ya que debe transferir datos entre el canal de datos interno de 16 bits y el canal externo de 8 bits.

III.2.1.1 La Unidad de Interfaz para Bus (BIU)

Esta unidad es capaz de coordinar múltiples EUs tales como un procesador numérico de datos (NDP) 8087, o un procesador de I/O 8089. La única diferencia entre el 8086 y el 8088 es su BIU.

La BIU manda direcciones, lee instrucciones de memoria, lee datos de los puertos y de memoria, y escribe datos a los puertos y a memoria. En otras palabras la BIU maneja todas las transferencias de datos y direcciones en los buses para la unidad de ejecución. Una EU provee una dirección lógica hacia la BIU la cual la convierte en una dirección física. A continuación se describen las partes que conforman a la BIU.

La Cola (*Queue*)

Para acelerar la ejecución del programa, la BIU lee hasta seis bytes de instrucciones de memoria antes de mandarlas ejecutar. Los bytes de instrucciones previamente leídos son almacenados para la EU en un grupo de registros organizado en la forma FIFO (primero en entrar - primero en salir), denominado como Cola. La BIU puede continuar leyendo instrucciones mientras que la EU está decodificando una instrucción o ejecutándola, lo cual no requiere del empleo de los buses. Cuando la EU está lista para la siguiente instrucción, simplemente la lee de la cola que hay en la BIU. Al proceso mediante el cual se lee la siguiente instrucción mientras se ejecuta la actual se le denomina "*pipelining*" por instrucción.

Registros de Segmentos

La BIU contiene cuatro registros de segmentos de 16 bits que son: el registro de *segmento de código* (CS), el registro de *segmento de stack* (SS), el registro de *segmento extra* (ES) y el registro de *segmento de datos* (DS). Estos registros de segmentos se emplean para guardar los 16 bits altos de las direcciones de inicio de los cuatro segmentos de memoria con los que esté trabajando el procesador en un momento dado, agregando la BIU ceros a los cuatro bits restantes. Esto se debe a que el 8086 sólo trabaja con cuatro segmentos (secciones relocizables de memoria) de 64 KBytes a la vez, dentro de su rango de 1 MByte. Los cuatro segmentos pueden estar separados, o, para programas pequeños que no requieren de los 64 KBytes para cada

segmento, se pueden traslapar. Los segmentos se localizan mediante límites de 16 bytes llamados párrafos. Los programas en lenguaje ensamblador se escriben en segmentos lógicos. La residencia de estos segmentos en memoria son una función del enlazador (linker) y el DOS.

El Apuntador de Instrucciones (IP).

Dado que el registro del segmento de código tiene los 16 bits más altos de la dirección de inicio de un segmento del cual la BIU está leyendo los bytes con los códigos de instrucciones, el registro apuntador de instrucciones tiene la dirección de 16 bits del siguiente byte de código dentro del segmento de código. El valor contenido en el IP se le conoce a menudo como desplazamiento (*offset*), dado que este valor debe ser se le debe sumar (desplazar) a la dirección base del segmento en el CS para producir la dirección física requerida de 20 bytes.

III.2.1.2 La Unidad de Ejecución (EU)

Esta unidad le indica a la BIU de dónde leer instrucciones o datos, decodifica instrucciones y las ejecuta. A continuación se describen las partes que la componen.

Circuitería de control, Decodificador de Instrucciones y ALU.

La circuitería de control dirige operaciones internas. Un decodificador se encarga de traducir las instrucciones leídas de memoria en series de acciones que la EU desempeña. Esta circuitería actúa de la misma forma que un secuenciador. Por último tenemos una Unidad Aritmético Lógica (ALU) de 16 bits que puede sumar, restar, incrementar, decrementar, complementar, hacer corrimientos y realizar operaciones lógicas AND, OR, XOR con números binarios.

Registro de banderas

Una bandera es un flip-flop que indica alguna condición producida por la ejecución de una instrucción, en base a lo cual puede controlar ciertas operaciones de la EU. El registro de banderas del 8086 es un registro de 16 bits que contiene 9 banderas activas. Seis de las nueve se emplean para indicar alguna condición producida por alguna instrucción. Estas banderas son : la bandera de *carry* o acarreo (CF), la bandera de *paridad* (PF), la bandera de *acarreo auxiliar* (AF), la bandera de *cero* (ZF), la bandera de *signo* (SF) y la bandera de *overflow* o sobrepaso (OF).

Las tres banderas restantes se emplean para controlar ciertas operaciones del procesador y, a diferencia de las anteriores, pueden ser activadas o desactivadas a voluntad por el usuario. Estas banderas son : la bandera *trap* (TF), que se emplea para la ejecución paso a paso de un programa, la bandera de *interrupciones* (IF), la cual permite o no la interrupción de un programa, y la bandera de *dirección* (DF), que se usa con instrucciones con cadenas de caracteres (*strings*).

Registros de propósito general

La EU tiene 8 registros de propósito general llamados AH, BH, BL, CH, CL, DH y DL. Estos registros pueden ser empleados individualmente para almacenamiento temporal de datos de 8 bits. El registro AL se le

llama también *acumulador*. Tiene algunas atribuciones que no tienen los otros.

Ciertos pares de estos registros pueden emplearse para almacenar palabras de 16 bits. Estos pares de registros son : AH y AL, BH y BL, CH y CL, Y DH y DL, a estos pares se les denomina AX, BX, CX y DX respectivamente. Para operaciones con 16 bits, el registro AX es el acumulador.

Registro de apuntador de pila (stack).

Recordemos que una pila (stack) es una sección de memoria que guarda los valores de las direcciones y datos mientras se ejecuta un subprograma. El 8086 permite utilizar un segmento de hasta 64 Kbytes como stack. Los 16 bits más significativos de la dirección de inicio de este segmento se guardan en el registro del segmento de stack. El registro apuntador al stack (SP) contiene el offset de 16 bits a partir de la dirección de inicio del segmento hasta la localidad de memoria donde está la palabra más recientemente almacenada en el stack. A esta localidad se le llama tope del stack.

Otros registros apuntadores y de índices.

La EU contiene un registro *apuntador base* (BP), un registro de índice fuente o *source index* (SI) y un registro de índice destino (DI), los tres de 16 bits cada uno. Estos pueden ser empleados para almacenamiento temporal de datos, al igual que los registros de propósito general. Sin embargo, su uso principal es el de almacenar offsets de 16 bits en un segmento determinado como podría ser el segmento de datos.

III.2.2 Acceso de datos en memoria (Direccionamiento).

El 8088 tiene 25 modalidades de direccionamiento; es un conjunto de reglas que especifican la localización de un dato utilizado durante la ejecución de una instrucción. La modalidad más sencilla es cuando un dato se localiza en un registro determinado; la más compleja, cuando se suma el contenido de dos registros en una cantidad de 8 ó 16 bits (se tiene un *offset*), que se encuentra en el programa. El resultado de la suma nos indica la dirección del dato.

Un *offset* también se conoce como una dirección efectiva. La EU genera una dirección efectiva empleando una serie de métodos llamados modos de direccionamiento. Una dirección efectiva puede hacerse con uno o más de los siguientes modos : de base, de índice y de desplazamiento.

Un modo de base puede ser hecho con un registro base o con un apuntador base; un modo de direccionamiento indexado con un índice fuente o con un índice destino; y el de desplazamiento es con un número de 16 bits fijo.

Si no se especifican registros de segmento, el registro de segmento de datos de usará en este caso. Si el registro apuntador de base se especifica como la base, el registro de segmento de pila se usará también. De otra manera si se reemplaza un registro de segmento que no es el dado por omisión se le llamará salto de segmento prefijo. El segmento no puede ser saltado por un IP, SP o registros de índice-destino a una operación de anillos o series.

La figura III-4 muestra cómo los diferentes modos de direccionamiento en la EU y la BIU se combinan para formar la

dirección física.

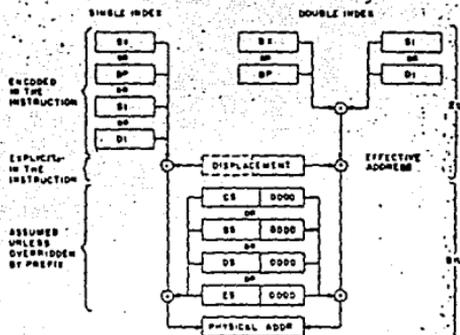


Figura III-4. Nodos de direccionamiento en la unidad de ejecución y la unidad de interfaz de bus se combinan para formar la dirección física.

Existen 3 tipos de direcciones: corto, próximo (NEAR) y distante (FAR). El direccionamiento corto se usa para enlazamiento, saltos condicionales y para algunos saltos incondicionales. Los direccionamientos NEAR y FAR se usan en llamadas y saltos incondicionales que no llenan los requisitos para un direccionamiento corto. Estos sólo afectan registros apuntadores de instrucción y son siempre relativos.

El direccionamiento FAR afecta a los registros de CS e IP y son absolutos. Cuando el valor del registro del CS no cambia, se tendrá direccionamiento de intrasegmento. Cuando el valor del registro del CS cambia, se tendrá direccionamiento de intersegmento. Una llamada NEAR pone solamente el registro del IP para que regrese sobre el stack de datos. Por otra parte, una llamada FAR pone a los registros del CS y del IP para que regresen sobre el stack de datos. Consecuentemente existen diferentes retornos FAR y NEAR, es por ello que un procedimiento que llame de NEAR a FAR, o viceversa, puede traer complicaciones. El esquema de direccionamiento segmento del 8088 difiere del esquema tradicional de direccionamiento lineal en varias formas, pues se hace cortando la longitud de la instrucción y el tamaño del número de datos requeridos para el direccionamiento.

Esto reserva memoria y aumenta la velocidad de ejecución ya que se requieren pocos ciclos de búsqueda (*fetch*). Esto también nos permite relocalizar fácilmente segmentos lógicos. El direccionamiento lineal es mejor para procesamiento de datos que son mayores de 64 KBytes.

III.2.3 Hardware del 8086

Del diagrama III-5 se aprecia en primera instancia que Vcc está en la terminal 40 y tierra en las terminales 1 y 20. En la terminal 19 está la entrada para la señal de reloj.

A continuación se pueden observar las líneas de los buses de datos y de direcciones. Si bien sabemos que se tienen 20 bits para direcciones y 16 para datos, no se aprecian directamente 36 líneas. La razón es que los diseñadores multiplexaron las 16 líneas inferiores de

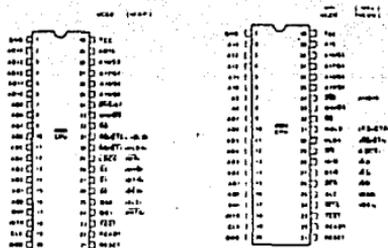


Figura III-5. Funciones de los pines de (a) 8086. (b) 8088

direcciones en el bus de datos para minimizar el número de terminales requeridas a 40. En otras palabras, las líneas del bus de datos denominadas ADO a AD15 se emplean al inicio del ciclo de máquina para mandar direcciones, y después en el ciclo de máquina se emplean para enviar o recibir datos. El 8086 expide una señal llamada habilitamiento del latch de direcciones (*address latch enable*) o ALE, mediante la terminal 25 para dar a conocer a la circuitería externa que la dirección está en el bus de datos. Los 4 bits más altos de la dirección son enviados mediante las líneas A16/S3 a A19/S6. El hecho de que estas terminales tengan 2 mnenónicos indica que durante el inicio del ciclo de máquina se emplean para enviar la dirección, y que durante el ciclo mandan la información de status, la cual identifica el tipo de operación a ser realizada durante ese ciclo.

Observemos ahora las líneas de control. La terminal -RD pasará a un estado bajo cuando el procesador esté leyendo datos de la memoria o de un puerto, esta señal está en la terminal 32. La terminal 29 tiene 2 funciones: -WR y -LOCK, la función tanto de esta terminal como la de todas aquellas en el rango de la 24 a la 31 dependerán del modo en el que esté operando el 8086.

El modo de operación del 8086 se determina por el nivel lógico aplicado a la terminal 33 (MN/-MX).

Si la terminal 33, rotulada MN/-MX recibe una señal alta (5V), el microprocesador trabaja en modo mínimo y las terminales 24 a 31 generan las señales que se muestran entre paréntesis en el diagrama. En este modo el 8086 genera directamente las señales de control de bus -RD, -WR y M/-IO, esta última indica si se está operando para lectura o escritura con memoria o con los puertos.

Por otra parte, si la señal aplicada a la terminal 33 es baja (0V), el 8086 trabaja en modo máximo y las terminales 24 a 31 generan las señales mostradas junto a las terminales. En este modo son enviadas en forma codificada por las líneas de estatus, -S0, -S1 y -S2. Para manejar estas señales se utiliza un dispositivo externo como puede ser el Controlador de Bus 8288 de Intel.

Las terminales "Request/Grant" (Solicitar/Otorgar), -RQ/-GT0 y -RQ/-GT1, son empleadas por otros dispositivos para indicarle al 8086 que ellos quieren los buses de direcciones, datos y control. Estas terminales son bidireccionales. Operan en forma similar a las señales de las terminales HOLD y HLDA cuando algún otro dispositivo quiere pedir los buses cuando se está operando en modo mínimo el 8086. El microprocesador puede enviar una señal a través de la terminal -LOCK mediante un programa de control para prevenir a algún otro dispositivo de tomar control del bus durante la ejecución de una instrucción crítica. Las señales de estado de cola (queue), QS1 y QS0, indican la operación más recientemente ejecutada en la

unidad de interfase de bus (BIU), en un byte de instrucción de cola. Estas señales permiten que un dispositivo externo pueda monitorear la cola del 8086 y que pueda leer los mismos bytes de instrucciones.

Por otra parte tenemos la terminal 21 o RESET, si ésta recibe un nivel lógico alto, no importa lo que esté haciendo, reestablecerá en ceros los valores de los registros de DS, SS, ES, IP y banderas. El registro CS obtendrá el valor FFFFH. Cuando el nivel lógico aplicado pase a ser bajo, el 8086 buscará la siguiente instrucción a ejecutar después de una reinicialización en la dirección física FFFFH.

Por último sólo resta mencionar a las terminales de interrupción: la NMI (interrupción no mascarable) en la terminal 17 y la INTR (interrupción) en la terminal 18, las cuales discutiremos más adelante.

III.2.4. Señales de flujo básicas en los buses

A partir del diagrama de tiempos del 8086 se pueden apreciar las actividades de los buses durante un ciclo de lectura o escritura simples. Este punto en particular se trata con más detalle en la sección III.5.

III.2.5 Direccionamiento y Decodificación de direcciones

Cuando el usuario escribe una palabra a memoria con una instrucción tal como por ejemplo MOV DS: WORD PTR[437AH], BX, la palabra es escrita en dos direcciones consecutivas de memoria. Asumiendo que DS contiene 0000, el byte bajo de la palabra se escribe en la dirección especificada de memoria, 4037AH, y el byte alto de la palabra se escribe en la siguiente localidad: 0437BH. Para hacer posible la lectura o la escritura en un ciclo de máquina, la memoria se arregla en dos secciones de 524,288 bytes cada uno.

Uno de los bancos contiene todas las direcciones pares y el otro todas las direcciones impares. Las líneas de datos conectadas al primer banco están conectadas a las terminales D0 - D7 del 8086, mientras que las líneas de datos del segundo banco están conectadas a las terminales D8 - D15. La línea de dirección A0 se emplea para habilitar dispositivos de memoria en el banco par, para lo cual la línea A0 debe tener un nivel lógico bajo. Las líneas de direcciones A1 a A19 se combinarán entonces con la línea A0 para indicar el dispositivo de memoria a seleccionar. En forma similar, se emplean las líneas A1 a A19 para el banco impar cuando la línea A0 está en estado alto. Para este banco existe una forma adicional para habilitar dispositivos de memoria, esto se hace mediante la señal *Bus High Enable* o *Habilitación Alta de Canal (-BHE)*, el 8086 envía esta señal al mismo tiempo que envía una dirección. Una memoria (*latch*) externa manejada por el μP toma la señal *-BHE* y la mantiene estable durante el resto del ciclo de máquina y permanecerá así si un byte está siendo accedido a una dirección impar o si una palabra está siendo accedida a una dirección par.

La razón principal para que A0 y *-BHE* trabajen en esta forma es la de prevenir la escritura de un byte no deseado en una localidad adyacente a la que corresponde durante un ciclo de escritura.

III.2.6 Acceso a memoria y a puertos

Existen dos grandes diferencias entre el 8086 y el 8088. En primer lugar, la cola de bytes de instrucciones es de sólo 4 bytes en vez de 6. En segundo lugar, y más importante, la memoria del 8088 no está dividida en dos bancos como la del 8086. El 8088 utiliza un bus de datos de 8 bits. Todos los dispositivos y puertos de un sistema basado en el 8088 están conectados a estas 8 líneas. Así, la memoria del 8088 funciona como un solo banco de 1 MByte. Esto significa, como ya hemos mencionado, que el 8088 requiere de dos ciclos de máquina para leer o escribir palabras de 16 bits. Las direcciones A0 a A19 se emplean con ciertos decodificadores para seleccionar el byte de memoria deseado. El 8088 no produce la señal -BHE puesto que no la necesita.

Dado que el canal de datos del 8086 es del doble de ancho que el del 8088, se podría esperar que el primero fuera el doble de rápido. Esto no es verdad. Una razón es que muchas aplicaciones requieren transferir datos en 8 bits. Otra es que el procesador tiene que hacer bastantes más cosas que transferir datos. La razón primordial tiene que ver con algunas características del diseño, a saber, el procesador interno dual y la cola de instrucciones de estructura tubular (pipeline).

III.3 ORGANIZACION DE LA MEMORIA

Una computadora del tipo PC o compatible tiene la capacidad para direccionar hasta 1 Mbyte de memoria (1,048,575 bytes) para datos y programas. La computadora trae en forma intrínseca, esto es programas y datos en memoria ROM tales como el ROM-BIOS (Basic Input Output System), el BASIC-ROM (esto no en todos los modelos de IBM PC compatibles) y, si se tiene disco duro, Hard Disk ROM BIOS, cada uno de los cuales ocupan una parte de la memoria principal. Estos programas necesitan áreas de memoria RAM ya que no se pueden modificar estando en ROM, de tal forma que se reserva otra parte para este fin. Las funciones de mapeo para el video requieren otra sección de almacenamiento, siendo necesario reservar un área para futuras aplicaciones de video.

Hay áreas en RAM reservadas a futuro para guardar la información que pudieran traer nuevas ROM. Y no olvidemos la sección de memoria reservada para guardar el DOS (Disk Operating System). Eso deja a la PC con un máximo de aproximadamente 600 Kbytes para el usuario, con 20 Kbytes más o menos dependiendo de la versión de sistema operativo que estemos utilizando. Esto es una cantidad "saludable" para programas. Además, siempre se pueden usar superposiciones de los diferentes módulos de un programa en un área de memoria conforme se van necesitando (overlays) y archivos en disco para reducir la cantidad de memoria requerida.

III.3.1 Categorías de RAM.

Memoria convencional. También llamada Memoria del Usuario ya que está disponible de forma inmediata para las aplicaciones del mismo, ésta es el área de 640 KB a partir de la dirección absoluta cero. Pede ir desde 256 KB en una PC aumentando en bloques de 128 KB con circuitos integrados o tarjetas de expansión.

Memoria base. Es la cantidad de memoria instalada actualmente en el área de memoria convencional.

Memoria reservada. Es el área que va desde la localidad correspondiente a 640 KB hasta la correspondiente a 1024 KB. Con excepción del segmento ocupado con la memoria para gráficas y ROM. Está reservada para futuras aplicaciones de IBM. Un bloque de 640 KB de direcciones dentro de esta área, situada entre la memoria de video y ROM, puede ser utilizada para un dispositivo de expansión de memoria.

Memoria de video. Conocida también como memoria de gráficas, es el área de memoria de 128KB que está inmediatamente después de la memoria convencional, está ocupada por RAM en varios adaptadores de despliegue.

Memoria extendida. Disponible sólo en el sistema AT, es RAM instalada para una o más tarjetas de expansión. La dirección inicial de la primera de estas tarjetas deberá estar en 1024KB.

Memoria expandida. Esto es hasta 8MB de RAM que se puede acceder a través de una porción de memoria de 64KB localizada dentro de la región de memoria reservada.

Memoria de expansión. Es un término aplicado a cualquier RAM empleada para aumentar la capacidad de una PC. Debido a la semejanza entre los términos "expandida" y "expansión", es importante recordar que este último se refiere a cualquier RAM o dispositivo adicional que pueda ser instalado en el área de memoria convencional o en la de memoria extendida.

III.3.2 Distribución de memoria

A continuación se muestra la distribución de memoria de una PC con capacidad de 1 Mbyte en bloques de 64 Kbytes.

Hex	Dec	***** Mapa de memoria en bloques de 64 K *****
00000	0K	----- Vectores, Datos, DOS, Disk/Advanced BASIC
10000	64K	----- Programas del usuario *
20000	128K	----- Programas del usuario *
30000	192K	----- Programas del usuario *
40000	256K	----- Programas del usuario *
50000	320K	----- Programas del usuario *
60000	384K	----- Programas del usuario *
70000	448K	----- Programas del usuario *
80000	512K	----- Programas del usuario *
90000	576K	----- Programas del usuario *
A0000	640K	----- Reservado para futuro video
B0000	704K	----- Video de color o monocromático
C0000	768K	----- ROM Futuro/ROM de disco duro
D0000	832K	----- ROM Futuro
E0000	896K	----- ROM Futuro
F0000	960K	----- Pruebas, ROM BASIC, ROM BIOS
FFFFF	1024K	-----
:	:	:
:	:	:

Hasta 15 Mb de memoria extendida

* Los programas en BASIC están limitados a un espacio de 64K

Es conveniente hacer notar que se manejan bloques de 64 Kbytes debido a que se maneja una arquitectura de 16 bits.

III.3.3 Mapa de Memoria

Para el siguiente mapa de memoria se considera : uso de el sistema operativo DOS 2.10, no existencia de los programas CONFIG.SYS ni AUTOEXEC.BAT y 384 KBytes de memoria.

Hex	Dec	
000	OK	----- Vectores del 8088 INT 0-7 Vectores del 8259 INT 8-F Vectores del BIOS INT 10-1F Vectores del DOS INT 20-2F INT Assignables 40-FF
400	K	----- Area de comunicaciones del BIOS en ROM
500		----- Areas de datos de DOS
700		----- IBMBIO
E30		----- IBMDOS
4DB9		----- Manejadores de Dispositivos Extensiones de usuario para el IBMBIOS tales como el ANSI.SYS y parametros del CONFIG.SYS
53F0		----- COMMAND.COM residente
5FD0		----- ENVIRONMENT (ambiente) para COMMAND.COM
0680		----- ENVIRONMENT para el siguiente programa
60B0		----- Extensiones para BASIC : Disco = 12K, Avanzado = 22K Inicio del espacio de trabajo (64K) de BASIC 4K de área de trabajo del interprete Buffers de comunicaciones (/C) Tamaño por omisión = 180h Rutinas de RS-232 Tamaño por omisión = 5E0h Bloques de control de archivo (/F) Tamaño por omisión = 234h Bloques de archivos aleatorios (/S) Tamaño por omisión = 80h Texto del programa en BASIC DS:30-31h Escalares hasta FFFFh DS:358-9h Arreglos hasta FFFFh DS:35A-Bh

```

.....
Espacio libre DS:35C-Dh
.....
Cadenas de caracteres, hasta 0000h DS:32F-0h
DS:30A-Bh
.....
Stack 200h bytes DS:2C-Dh
-----
10000 Sin usar y disponible
-----
1C000 Buffer de video en la PC jr.
-----
fin-3410 COMMAND transitorio
fin-F8h Mensajes de error
fin-B10h Tabla de comandos internos
fin-9F5h Texto del ultimo comando
fin-9F6h Longitud del ultimo comando
fin-8AEh Especificación de archivo formateado
-----
:::FIN DE LA EXPANSION DE LA MEMORIA RAM:::
-----
A0000
:
:
.....

```

III.3.4 Mapa de Puertos

Usando las mismas líneas de datos y direcciones que la memoria principal, el espacio de direcciones del puerto de I/O se distingue de la memoria principal sólo por la presencia de una señal en una línea de control.

La arquitectura del 8088 permite un espacio direccionable para el puerto de I/O de 1024 bytes (1K) debido a que solo 10 bits son usados para derivar la dirección del puerto. Los puertos en este espacio de direcciones son accedidos mediante el uso de instrucciones especiales de lenguaje ensamblador tales como IN y OUT.

La implementación de los puertos en la PC los separa en varios grupos de uso: para la tarjeta del sistema solamente, para el uso de la tarjeta del sistema y el canal de I/O (solo para salida), y solo para el uso del canal de I/O.

Estos puertos del canal de I/O no pueden ser usados uniformemente para propósitos tanto de entrada como de salida. Algunos puertos del dispositivo son usados para diferentes tipos de datos mediante una forma de flip-flop. Otros puertos son utilizados para diferentes propósitos dependiendo del contenido actual de un segundo puerto. Los contenidos esperados de un puerto así como las direcciones de I/O manejadas son determinadas solamente por el dispositivo conectado al puerto.

Puertos de I/O desde 000h hasta 0FFh - Uso : Tarjeta del sistema.

	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh	Eh	Fh
000h	DMA 8237-A															
010h	MFG															
020h	INT 8259															
030h																
040h	TIMER 8253-5															
050h																
060h	PPI 8255A-5															
070h																
080h	Regs. Pag. DMA															
090h																
0A0h	Registro de mascarar de N M I															
0B0h																
	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh	Eh	Fh
0C0h	(1)															
0D0h																
0E0h																
0F0h																

(1) Selección de conexión de canal 3 de DMA

Puertos de I/O desde 100h hasta 1FFh - Tarjeta del sistema y uso del canal de I/O.

**** Uso restringido para salida, No utilizado en la PC ****

Puertos de I/O desde 200h hasta 3FFh - Uso del canal de I/O

	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh	Eh	Fh
200h	Control para juegos															
210h	Unidad de expansion															
220h																
230h	RESERVADO															
240h																

	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh	Eh	Fh
250h																
260h																
270h										3a. impresora en paralelo						
280h																
290h																
2A0h																
2B0h																
2C0h																
2D0h								3 7 2 0 P C								
2E0h																
2F0h			Reservado							Comunicaciones Asincronas 2						
300h										Tarjeta Prototipo						
310h										Tarjeta Prototipo						
320h										Disco Duro						
330h		XT/370														
340h																
350h																
360h																
370h										2a. impresora en paralelo						
380h										SDLC o 2a. Comunicacion Bisincrona						
390h																
3A0h										1a. comunicacion bisincrona						
3B0h										Monitor monocromático y 1a. impresora en paralelo						
3C0h										Reservado						
3D0h										Monitor de Color/Graficas						
3E0h										Reservado						
3F0h										Disco						Comunicaciones Asincronas 1

III.3.5 Segmentación de Memoria

Conociendo el hecho de que el microprocesador 8088 de la PC puede direccionar hasta 1 MByte (1024 K), esto es, que una dirección de memoria puede llegar hasta FFFFh (hexadecimal), requiriendo de 20 bits para expresar esta cantidad, nos damos cuenta de que se necesita medio byte adicional al que pueden manejar los dos registros de 16 bits del 8088.

La figura III-6 ilustra el proceso que el un 8088 utiliza para desarrollar una dirección que abarca el rango de memoria que va desde 0h hasta FFFFh. Se agrega una compensación (desplazamiento) de dos bytes a un segmento (dirección de inicio de 16 bits (10h) o múltiplo que es escrita sin el dígito menos significativo) con un valor ajustado de dos bytes para manejar la dirección de memoria. El 8088 realiza esta operación en una forma rápida, y para asegurar que el registro del segmento es cargado con el número de segmento correcto, se coloca el desplazamiento el número de bytes que excedan al inicio del segmento.

El registro del segmento se ajusta al agregar un cero de orden menor, el cual cambia el número del segmento a la dirección de la localidad anterior de memoria más cercana que termine en 0h. En otras palabras, el número de segmento B800h se cambia a la dirección B8000h. No tiene sentido "arrastrar" el cero remanente en el número de segmento, puesto que se necesita el espacio en el registro para especificar todo el rango de posibles segmentos: desde 0 hasta FFFFh. El 8088 ajusta el número de segmento, sumándole lógicamente un cero de orden menor, le agrega un valor de desplazamiento, y saca de una pila (stack) una dirección de memoria.

De esta forma, todas las direcciones que terminen en 0h son potencialmente un número de segmento, el cual no incluye el 0h de orden menor. Para direccionar FFFF2h, el registro del segmento podría contener FFFFh y el desplazamiento sería 2h. Algunas veces ayuda especificar una dirección indicando el segmento y el desplazamiento separados por dos puntos. Por ejemplo, la dirección FFFF2h se podría escribir como FFFF:2.

Debido a que un offset puede variar desde 0 hasta FFFFh, es de imaginarse que sería posible hacer referencia a una dirección de memoria utilizando distintas combinaciones segmento : desplazamiento. Un desplazamiento puede direccionar cualquier byte a lo largo de 64 K de memoria, no habiendo razón para que el segmento no pueda comenzar en cualquier dirección que termine con cero (llamada límite de párrafo), mientras que la relación cumpla con la dirección deseada.

La convención que se tiene para evitar confusiones es la de sólo iniciar segmentos que terminen con cero (como podría ser el segmento número 40) y que se refieran al principio de un bloque mayor de datos o instrucciones (por ejemplo el segmento B800h).

En resumen : cualquier dirección tiene dos partes, cada una de las cuales es una cantidad de 16 bits. Una es el desplazamiento y la otra es la dirección de segmento. La primera se compone de un número fijo, una base y un índice. La dirección de segmento se almacena en uno de los cuatro registros de segmento (CS, DS, ES, OS). El procesador utiliza estas dos cantidades de 16 bits para calcular la dirección real de 20 bits, según la siguiente fórmula:

Dir. real= 16* (dir. del segmento) + Desplazamiento

III.4 INTERRUPCIONES DEL 8086

Una interrupción al 8086 puede venir de una de tres fuentes. Una de ellas es una señal externa aplicada a la terminal de Interrupciones No Mascarables (NMI) o bien a la terminal de interrupción (INTR). Ambas formas de interrumpir al procesador son conocidas como Interrupciones por Hardware.

Una segunda fuente de interrupción es la ejecución de la instrucción INT. Esta se conoce como una interrupción por software.

La tercera fuente de interrupción es debida a alguna condición anormal en el 8086 producida por la ejecución de alguna instrucción; un ejemplo de esto es la interrupción producida por una división entre cero. Este tipo de interrupciones condicionales también se conocen como de tipo software.

Al final de cada ciclo de instrucción, el 8086 verifica si se ha solicitado alguna interrupción, en caso afirmativo el 8086 responde tomando las siguientes acciones secuenciales :

1. Decrementa el contador de la pila (*stack*) en dos e introduce (hace un "Push") del registro de banderas en el Stack.
2. Deshabilita la entrada de interrupciones INTR limpiando la bandera de interrupciones en el registro de banderas.
3. Reestablece la bandera de ejecución por pasos (*trap flag*) del registro de banderas.
4. Decrementa el apuntador de pila (*Stack Pointer*) en dos y hace un "Push" del contenido del registro de Segmento de Código (*Code Segment*) actual en el Stack.
5. Decrementa el Stack Pointer una vez más en dos y hace un "Push" del contenido del apuntador de instrucción (*Instruction Pointer*) actual en el Stack.
6. Transfiere en forma indirecta el contador del programa (*Program Counter*) al inicio de la rutina de interrupción desarrollada por el usuario.

Cuando el 8086 hace una llamada a un procedimiento pone un nuevo valor en el registro del Segmento de Código (CS) así como en el Apuntador de instrucción (IP).

Para una llamada indirecta el 8086 obtiene de ciertas localidades de memoria los valores del CS así como del IP para ubicar el inicio de la rutina de interrupción. En un sistema basado en el 8086 el primer KByte de memoria principal, desde 00000H hasta 003FFH se utiliza como una tabla donde se guardan las direcciones iniciales de los procedimientos de interrupción. Debido a que se requieren cuatro bytes para guardar los valores de CS y de IP para cada procedimiento de interrupción, la tabla puede tener las direcciones de inicio de hasta 256 procedimientos de interrupción. La dirección de inicio de una rutina de interrupción almacenada en esta tabla es comunmente conocida como vector de interrupciones o apuntador de interrupciones, siendo conocida la tabla como tabla de vector de interrupciones o tabla de apuntador de interrupciones.

224 apuntadores disponibles	3FFH	Apuntador tipo 255	Dirección base del CS y Offset del IP
	3FCH	(disponible)	
	084H	Apuntador tipo 33 (Disponible)	
	080H	Apuntador tipo 32 (Disponible)	
27 apuntadores reservados	07FH	Apuntador tipo 31 (Reservado)	
	014H	Apuntador tipo 5 (Reservado)	
	010H	Apuntador tipo 4 (Overflow)	
	00CH	Apuntador tipo 3 Instrucción de interrupciones de 1 byte	
	008H	Apuntador tipo 2 No mascarable	
5 apuntadores dedicados	004H	Apuntador tipo 1 Paso por paso	
	000H	Apuntador tipo 0 Error de división	

+----- 16 bits -----+

La figura anterior muestra como están arreglados los apuntadores de interrupción en la tabla de memoria, cada uno se identifica por un número que va del 0 al 255. Intel llama a este número el tipo de la interrupción. Los cinco tipos de menor nominación están dedicados a interrupciones de tipo específico, los 27 tipos siguientes (del 5 al 31) son reservados por Intel para uso en futuros microprocesadores. Los 224 últimos son disponibles para el usuario, tanto para interrupciones de hardware como de software.

Quando el 8086 responde a una interrupción, va automáticamente a la localidad especificada en la tabla de apuntadores de interrupciones para obtener la dirección inicial de la rutina de interrupciones. Sin embargo, el 8086 no carga automáticamente esta dirección, esto debe ser hecho por el usuario. Nótese que el nuevo valor del *Instruction Pointer* está en

la parte baja de la palabra del apuntador IP, mientras que el nuevo valor del Code Segment se pone en la parte baja de la palabra alta del apuntador.

A continuación se discuten algunas formas específicas en las que se puede interrumpir el 8086, así como la forma en la que este microprocesador responde a las diferentes interrupciones.

III.4.1 Interrupción No Mascarable

Quando el 8086 detecte una transición de bajo a alto en su terminal de entrada NMI ejecutará una respuesta a interrupción de tipo 2. Cuando esto suceda, como ya se explicó, el 8086 hace un "Push" de las banderas en el Stack, reestablece en cero las banderas TF (ejecución por pasos) e IF (bandera de interrupciones), y hace un Push de los valores de CS e IP en la siguiente instrucción del Stack. Toma el valor de CS para la rutina de interrupción de la dirección 0000AH, y el valor para IP de la dirección 00008H.

En una interrupción de tipo 2 la respuesta no puede ser deshabilitada por ninguna instrucción de ningún programa, por lo tanto a esta clase de interrupciones les podemos asociar el término de "No mascarables".

Comunmente se emplea este tipo de interrupción para salvar datos en caso de una falla en la alimentación de energía al sistema. Existe una circuitería externa que detecta cuando la energía de CA falla, en este momento manda una señal de interrupción a la terminal de NMI del 8088. Debido a los filtros capacitivos en las fuentes de potencia, esta prevalecerá en el sistema por lo menos durante los 50 ms siguientes a la falla. Este tiempo es más que suficiente para permitirle a una rutina de interrupción de tipo 2 que guarde los datos del programa en una memoria RAM con fuente de alimentación de respaldo.

III.4.2 Interrupciones por Software - Tipos 0 a 255

La instrucción INT del 8088 puede ser utilizada para que el microprocesador pueda ejecutar alguno de sus 256 tipos de interrupciones. El tipo de interrupción deseado se especifica como parte de la instrucción. Cuando esta instrucción se ejecute, el μP hará un push del registro de banderas en el stack, pone en ceros las banderas TF e IF, y hace un push de los valores de CS e IP de la siguiente instrucción en el stack. En ese momento traerá los valores de CS e IP para poder comenzar la rutina de interrupción a partir del apuntador de interrupciones en la tabla de memoria. El valor de IP para cualquier tipo de interrupción siempre se encuentra en una dirección que es 4 veces el tipo de interrupción, y el valor del CS se encuentra en una localidad dos direcciones mayor.

Las interrupciones producidas por la instrucción INT tienen múltiples usos, como puede ser el de invocar a diferentes procedimientos desde programas distintos en un sistema como lo es el BIOS en la IBM PC, la cual tiene en ROM una serie de procedimientos, cada uno de los cuales ejecuta una función específica (como veremos más adelante), tal como leer un caracter del teclado. Los procedimientos del BIOS son llamados mediante la instrucción INT

III.4.3 Interrupciones INTR - Tipos 0 a 255

La terminal de entrada INTR del 8086 permite que una señal externa interrumpa la ejecución de un programa. A diferencia de la interrupción NMI, esta puede ser "mascarada" a fin de que no se cause la interrupción. Si la bandera de interrupciones (IF) se pone en cero, INTR es deshabilitada. La IF se puede poner en cero en cualquier momento mediante la instrucción CLI (Clear Interrupt). Si IF adquiere un valor 1, la entrada INTR se habilitará, esto se hace mediante la instrucción STI (Set Interrupt).

Cuando se le aplica una señal de reinicialización (Reset) al 8086, la IF se pone en cero en forma automática, de tal manera que si se quiere utilizar una señal INTR deberá ejecutarse previamente una instrucción STI. El microprocesador fué diseñado de esta manera a fin de que los puertos, los temporizadores (timers), los registros, etc. puedan ser inicializados antes de habilitar la terminal INTR, esto es, se permite que el 8086 esté listo para manejar las interrupciones de éste nivel antes de que ellas tengan lugar.

Así mismo, la bandera de interrupciones se pone en cero en forma automática inmediatamente que el 8086 responde a una interrupción. Esto es así por dos razones. Una de ellas es prevenir que una señal en INTR haga que se interrumpa una rutina de interrupción de mayor prioridad que se esté ejecutando; otra razón es la de asegurarse que la señal ya recibida por el 8086 no lo interrumpa una y otra vez.

Cuando la entrada de INTR esté alta e INTR esté habilitado, se interrumpirá el 8086.

La instrucción IRET (Regreso de una Interrupción) al final de un procedimiento de interrupción reestablece las banderas a la condición que tenían antes de ejecutarse la rutina extrayendo el valor del registro de banderas del Stack. Esto reestablece la entrada INTR. Si la señal de nivel alto prevalece en dicha terminal, el microprocesador será interrumpido una vez más. Si no se quiere que esto suceda, se deben emplear circuitos externos para asegurarse de que la señal pase a un estado bajo antes de reestablecer a INTR.

La respuesta a las interrupciones INTR diferente al resto de las interrupciones (NMI e INT), siendo la diferencia principal que el tipo de interrupción se manda al 8086 por medio de un dispositivo externo tal como es el Controlador de Prioridad de Interrupciones 8259A.

Desde el punto de vista del hardware del 8086 se tiene que al recibir una interrupción de este tipo, se realizan dos ciclos de máquina que tienen como propósito obtener el tipo de interrupción del dispositivo externo. Al principio del primer ciclo de máquina de reconocimiento de la interrupción el 8086 "flota" (cede) las líneas del bus de datos, AD0-AD15. Entonces manda un pulso de reconocimiento de interrupción en su terminal de salida -INTA. Este pulso sirve para que el dispositivo externo esté listo para mandar el tipo de interrupción. Durante el segundo ciclo de máquina de reconocimiento de interrupción el 8086 manda

otro pulso en -INTA, al cual el dispositivo externo debe responder con el tipo (número) de interrupción. en las 8 líneas más bajas (0-7) del canal de datos, de donde son leídas por el 8086. Una vez que ha sucedido esto el microprocesador realiza el proceso ya descrito para salvar los valores de TF,IF,CS e IP. El 8086 utiliza entonces el valor del tipo de interrupción que leyó del dispositivo externo para saber de dónde va a tomar los valores para CS e IP correspondientes al procedimiento de interrupción, a partir de la tabla de apuntadores de interrupción que existe en memoria. El valor de IP para el procedimiento será colocado en una dirección tal que sea el valor del tipo de interrupción por 4, mientras que el valor de CS será puesto en una dirección igual a 4 veces el valor de la interrupción más 2.

La desventaja de que un dispositivo externo indique el tipo de interrupción es que este debe comportarse como un embudo de las diferentes señales provenientes de diferentes fuentes que buscan interrumpir al 8086. Cuando el 8086 responde con pulsos en -INTA, el dispositivo externo debe enviar al 8086 el tipo de interrupción que corresponde a la fuente de la señal de interrupción. Es por ello que el dispositivo externo debe avisar si existen dos fuentes que deseen interrumpir al microprocesador al mismo tiempo. Sin embargo, esto libera al μP de la tarea de tomar la decisión referente a la asignación de recursos para la atención de una interrupción, pues es evidente que no todas se pueden atender al mismo tiempo.

III.4.3.1 Controlador de Prioridad de Interrupciones (PIC) 8259A

Para apreciar como el 8259A se acopla a la operación INTR se debe hacer notar que el bus de datos de 8 bits le permite al 8086 enviar palabras de control al 8259A, así como leer de este una palabra de estatus. Las entradas -RD y -WR controlan estas transferencias cuando el dispositivo es seleccionado poniendo en estado bajo la entrada a la terminal -CS (Chip Select). El bus de datos también le permite al 8259A enviar los tipos de interrupción al 8086. Por otra parte se tienen las 8 entradas de interrupciones (IRO-IR7), una señal de interrupción aplicada a cualquiera de ellas causará que la terminal de salida INT cambie a un estado alto, si está conectada a la terminal INTR del 8086 y si la bandera de interrupciones del microprocesador está activada, entonces dicha señal causará la respuesta a la interrupción previamente expuesta.

La entrada -INTA del 8259A está conectada a la salida -INTA del 8086. El PIC usa el primer pulso de -INTA del microprocesador para realizar ciertas actividades que dependen del modo en el que está programado y, como ya se ha mencionado, cuando recibe el segundo pulso de -INTA manda el tipo de interrupción al bus de datos. El tipo de interrupción depende de la señal recibida en las terminales IR y de un número que se le da al 8259A cuando se le inicializa.

El PIC 8259A se opera comunmente en un modo llamado "modo de prioridad fija", en dicho modo la señal que llega a IRO tiene la mayor prioridad, mientras que IR7 tiene la menor. Esto significa que si dos señales de interrupción llegan al mismo tiempo se atenderá primero a la de mayor prioridad.

El 8259A cuenta con 4 registros : el Registro de Solicitud de Interrupciones (IRR), el registro de máscara de interrupciones (IMR), el registro de servicio a interrupciones (ISR) y el determinado

de prioridades.

El IRR lleva registro de cuales terminales IR solicitan servicio preñdiendo el bit correspondiente a la terminal.

El IMR se emplea para deshabilitar (con un 0) o habilitar (con un 1) entradas de interrupciones en forma individual. Cada bit en este registro corresponde a la entrada con el mismo número.

El ISR guarda la información referente a cuales interrupciones están siendo atendidas en el momento actual, para cada una se prenderá el bit correspondiente en el registro ISR.

A fin de que las interrupciones sean atendidas adecuadamente, debe existir lo que se conoce como Vector de interrupciones, el cual es un arreglo de localidades de memoria que contiene las direcciones de las rutinas de atención a interrupciones. A continuación se describe la inicialización en este aspecto del 8259A dentro del marco de referencia del sistema de la IBM PC.

El controlador puede ser direccionado a través de las direcciones del mapa de puertos 0020H y 0021H (H significa hexadecimal). Sin embargo, existen más de dos registros a inicializar. Para resolver este problema el 8259A utilizando primeramente las direcciones del mapa de puertos en un modo de inicialización y después reutilizándolas en un modo de comandos de operación. El modo de inicialización puede darse en cualquier momento mediante la escritura a la dirección 0020H del mapa de puertos el nibble (4 bits) 1111. Este conjunto de datos contiene información que será definida más adelante y que se conoce como palabra comando de inicialización uno (ICW1). A esta palabra le siguen otras tres palabras de inicialización antes de entrar al modo de comandos de operación. Las 3 ICW's adicionales se escriben a la dirección 21H y son introducidas en un registro de pila interno en el PIC 8259A. Las palabras deben entrar secuencialmente. Cuando se ha introducido la última, el PIC entra el modo de control de operación. Al escribir en la dirección 21H del mapa de puertos se coloca la palabra de control de operación uno (OCW1), la OCW2 se direcciona escribiendo en la dirección 20H un nibble con los bits 3 y 4 puestos en cero. La OCW3 se direcciona escribiendo un nibble en la dirección 20H con el bit 3 puesto en 1 y el 4 en cero.

Lo anterior se puede resumir en la siguiente tabla :

Dirección del Puerto	Registro
HEX 0020	ICW1
HEX 0021	ICW2
HEX 0021	ICW3
HEX 0021	ICW4
HEX 0021	OCW1
HEX 0020	OCW2
HEX 0020	OCW3

Tabla III.1

III.4.4. Prioridades de las interrupciones del 8088

Las prioridades de las interrupciones del 8088 son las siguientes :

<u>Interrupción</u>	<u>Prioridad</u>
Error de división, INT n, INTO	Mayor
NMI	
INTR	
Ejecución por pasos	Menor

III.5 OPERACIONES DE LA UNIDAD DE BUS DEL SISTEMA

La mayoría de los componentes funcionales de la tarjeta del sistema están unidos al microprocesador (μP) 8088 mediante el bus del sistema. Este bus consta de varios tipos de líneas de señales: el bus de datos, el bus de direcciones, el bus de control, señales de tiempos, de solicitud de interrupciones y de control de acceso directo a memoria. Existen dos clasificaciones para los buses que maneja la IBM PC. La primera se refiere al "bus local", el cual es una interfaz altamente codificada y multiplexada. Asociados a esta están: (1) una conexión especial capaz de alojar al coprocesador numérico 8087 de Intel, (2) el controlador de interrupciones 8259A, (3) el controlador de bus 8288 y, (4) los circuitos de energización y demultiplexación. Las salidas del controlador de bus así como los circuitos de mencionados en cuarto lugar forman las señales básicas que conforman al bus del sistema, asociadas al cual están: (1) los circuitos soporte del procesador (incluyendo los canales de DMA y los contadores temporizadores), (2) la lógica de decodificación de direcciones de memoria y puertos de I/O, (3) ROM, (4) RAM, (5) interruptores con funciones de reconocimiento, (6) adaptadores integrados de I/O, y (7) los cinco slots de expansión de la tarjeta del sistema.

La mayoría de las aplicaciones de conexión de la PC con el mundo exterior se realizan a través de uno de los cinco slots de expansión del bus del sistema. En este bus, los datos se transfieren durante lo que se llama un "ciclo de bus". En esta sección se describirán los tipos de ciclos de bus que existen y como se emplean para transmitir información entre la memoria, el I/O, y el μP 8088.

III.5.1 Ciclos de Bus.

Se manejan dos categorías generales para los ciclos de bus: los utilizados por el 8088 y los empleados por DMA. Cuando el 8088 genera un ciclo de bus, maneja el sistema de bus con una dirección de memoria o con la de un puerto de I/O, controla la dirección del flujo de datos y se comporta ya sea como emisor o receptor de datos. Cuando el 8088 maneja el bus, se pueden generar cinco clases distintas de ciclos de bus. El primero es un ciclo de bus de lectura de memoria. El segundo es un ciclo de bus de escritura a memoria. El tercer tipo es un ciclo de bus de lectura a un puerto de I/O. El cuarto es de escritura a un puerto de I/O. El quinto es un ciclo de bus de reconocimiento de interrupciones, este último sólo se da en el bus local y no en el del sistema en su totalidad.

La segunda clasificación general se refiere, como ya dijimos, a los ciclos de bus manejados por el circuito controlador del DMA 8237-5. Cuando se presenta un Acceso Directo a Memoria el 8237-5 toma el control del sistema, haciendo a un lado al 8088. El controlador de DMA maneja las direcciones de memoria en el bus y controla el flujo de datos entre el adaptador de interfaz y la memoria. En este caso el controlador no actúa como emisor o receptor de datos, sino sólo como un intermediario entre los elementos mencionados. Cuando se da una situación de este tipo, se pueden generar dos tipos de ciclos de bus. El primero lee de el adaptador de interfaz y escribe datos en la localidad de memoria especificada por el controlador de DMA. El

segundo tipo de ciclo de bus lee datos de una localidad específica de memoria y manda esos datos al adaptador de interfaz.

En resumen, tendremos la siguiente clasificación :

- Ciclos : a) 8088 →
1. Lectura de Memoria
 2. Escritura a Memoria
 3. Lectura de un Puerto de I/O
 4. Escritura a un Puerto de I/O
 5. Reconocimiento de Interrupción
- b) DMA →
1. Lectura de Interfaz - Escritura a Memoria
 2. Lectura de Memoria - Escritura a Interfaz

1. Ciclo de Bus de Lectura de Memoria

Este ciclo se emplea para manejar datos e instrucciones alojados en la memoria del sistema. Esta memoria puede estar en la tarjeta madre o en alguna tarjeta alojada en uno de los slots de expansión y puede ser de tipo RAM o ROM. Todos los ciclos de bus constan de un mínimo de 4 ciclos de reloj, cada uno de los cuales dura aproximadamente 210 ns, lo cual nos daría una duración aproximada de 840 ns para el ciclo de bus de lectura de memoria. Esta duración se puede incrementar mediante un dispositivo de memoria colocado en la tarjeta alojada en el slot de expansión al colocar en estado bajo la línea de interfaz denominada READY. Aquellas señales que están activas son manejadas por el 8088 y sus circuitos de almacenamiento de señales (buffers). Se tiene una excepción que es el bus de datos, el cual es manejado con los datos a partir de las direcciones de las localidades de memoria. Se ilustra gráficamente el ciclo.

El ciclo de lectura de memoria comienza durante el pulso de reloj con la señal ALE activa. El borde anterior de esta señal indica que bus de direcciones contiene una dirección válida de memoria. continuación, la señal de bus -MEMR se activa aproximadamente en T2. Es le indica a los dispositivos asociados al bus que el ciclo es de lectura de memoria. También indica si el dispositivo contiene memoria con u dirección que corresponda a la que está presente en el bus direcciones, debe manejar el bus de datos con su contenido. Todos l dispositivos de memoria deben decodificar las direcciones presentes en bus y, así, determinar si es el dispositivo que debe responder. El 8088 captura los datos del bus de datos al inicio del ciclo T4. Po después del inicio de este ciclo, la señal del bus -MEMR se desactiva el ciclo de bus termina al final del ciclo T4 (fig. III-6a).

2. Ciclo de Bus de Escritura a Memoria

Este se emplea cuando se le indica al 8088 escribir datos en una localidad de memoria. Así como en el ciclo de lectura, el 8088 y sus buffers de bus manejan una dirección en el bus del sistema, indicando la localidad de memoria que debe aceptar el dato que le manda el 8088. Además de manejar el bus de direcciones y las señales de control, el 8088 también maneja el bus de datos con el dato a ser escrito. Se ilustran los tiempos básicos del ciclo.

La señal de bus ALE se activa durante el ciclo T1, indicando que el bus de direcciones contiene una dirección válida. A continuación, la señal de bus -MEMW se activa indicando que el ciclo de bus es un ciclo de lectura a memoria, esta señal se activa aproximadamente en el ciclo T2. Poco después, el 8088 maneja el bus de datos del sistema con los datos a ser escritos. En el ciclo T4, la señal de bus -MEMW se desactiva, terminando el ciclo de bus al finalizar T4 (fig. III-6b).



Figura III-6. Tiempos para el ciclo de bus de (a) lectura de memoria. (b) Escritura en memoria.

3. Ciclo de Bus de Lectura de Puerto de I/O

Este ciclo se inicia cada vez que se ejecuta una instrucción IN. Este ciclo de bus es similar al de lectura de memoria, su función es buscar datos a partir de alguna de las direcciones de los puertos de I/O en el espacio de direcciones reservado para este propósito. En el diseño de la PC, este ciclo ocupa un mínimo de 5 ciclos de máquina, o sea, aproximadamente 1.05 ms. Un dispositivo específico de I/O puede incrementar esta duración desactivando la señal READY del bus. Durante este ciclo, el 8088 maneja un puerto de direcciones de 16 bits en el bus de direcciones del sistema. Nótese que durante este ciclo de bus, los 4 bits más altos del bus de direcciones nunca son activados. Este ciclo se ilustra gráficamente en la figura III.7.



Figura III-7. Tiempos de ciclo de bus para lectura del puerto de I/O.

Durante el ciclo de relojes T1, la señal de bus ALE se activa indicando que los bits 0 a 15 del bus de direcciones contienen una dirección válida del puerto de I/O. Al llegar el ciclo T2, se activa la señal -IOR indicando que se trata de un ciclo de lectura a un puerto de I/O y que el puerto direccionado debe responder por el

manejo del contenido del bus de datos. Al inicio del ciclo T4, el procesador muestrea los datos en el bus de datos y la señal -IOR se desactiva. El ciclo termina al final de T4. Debe notarse que se necesitan solo 4 ciclos de reloj, el ciclo extra, llamado ciclo TW, se inserta automáticamente en cada ciclo de bus de este tipo.

4. Ciclo de Bus de Escritura a un Puerto de I/O

Este ciclo dá comienzo cada vez que se ejecuta una instrucción OUT, permitiéndole al 8088 escribir datos en un puerto determinado de I/O, conociendo su dirección específica. Al igual que el anterior, este ciclo ocupa normalmente cuatro ciclos de reloj, pero el diseño de la PC inserta un ciclo de reloj extra TW. De esta forma se tiene un mínimo de cinco ciclos de reloj para este ciclo de bus, el cual se ilustra en forma gráfica en la figura III.8.

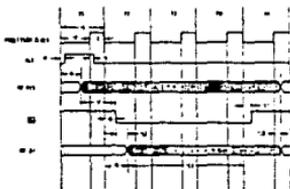


Figura III-8. Tiempos de ciclo de bus para escritura en el puerto de I/O

En forma semejante a los ciclos anteriores, la señal de bus ALE se activa durante T1, indicando que el bus de direcciones contiene una dirección de puerto válida. A continuación, la señal de control de bus -IOW se activa al llegar el ciclo de reloj T2, indicando que se trate de un ciclo de escritura a un puerto de I/O y que la dirección del puerto seleccionado debe tomar los datos del bus de datos. Poco después de terminar T2, el μP maneja el bus de datos con los datos para la dirección del puerto. Al inicio del ciclo T4, la señal de control -IOW se desactiva, terminando el ciclo de bus al finalizar T4.

III.5.2 Tiempos del Bus del sistema

La clave para diseñar una interfaz adaptada al bus del sistema es el comprender la compatibilidad de tiempos con dicho bus. Por ejemplo, es necesario saber que tan rápido deben presentarse los datos después de que la memoria o un registro han sido accesador por el 8088.

Para tener un margen de seguridad, se presentan los tiempos del máximo peor caso y el mínimo peor caso. Esto significa que la información de tiempos es válida para todas las condiciones del bus en las que se manejen niveles de energía que estén dentro de los límites especificados.

Los tiempos mostrados en la tabla III-1 son :

- a) Ciclo de bus para inicio de lectura de memoria por el 8088.
- b) Ciclo de bus para inicio de escritura a memoria por el 8088.

- c) Ciclo de bus para inicio de lectura de puerto de I/O por el 8088.
- e) Ciclo de bus para inicio de escritura a puerto de I/O por el 8088.
- f) Ciclo de bus para inicio de lectura de memoria y escritura a un puerto de I/O por DMA.
- g) Ciclo de bus para inicio de lectura de un puerto de I/O y escritura a memoria por DMA.

Símbolo	Max	Min
t1	-	209.5
t2	-	124.5
t3	-	71.8
t4	15	-
t5	15	-
t6	128	16.0
t7	-	91.5
t8	35	10.0
t9	-	42.0
t10	-	10.0
t11	35	10.0
t12	-	342.0
t13	-	458.5

Tiempos para el ciclo de bus de
Lectura de Memoria

Símbolo	Max	Min
t1	-	209.5
t2	-	124.5
t3	-	71.8
t4	15	-
t5	15	-
t6	128	16.0
t7	-	91.5
t8	35	10.0
t9	122	14.0
t10	-	10.0
t11	35	10.0
t12	112	-
t13	-	297.0

Tiempos para el ciclo de bus de
Escritura a Memoria

Símbolo	Max	Min
t1	-	209.5
t2	-	124.5
t3	-	71.8
t4	15	-
t5	15	-
t6	128	16.0
t7	-	91.5
t8	35	10.0
t9	-	42.0
t10	-	10.0
t11	35	10.0
t12	-	551.5
t13	-	668.0

Tiempos para el ciclo de bus de
Lectura de un puerto de I/O

Símbolo	Max	Min
t1	-	209.5
t2	-	124.5
t3	-	71.8
t4	15	-
t5	15	-
t6	128	16.0
t7	-	91.5
t8	35	10.0
t9	122	14.0
t10	-	10.0
t11	35	10.0
t12	112	-
t13	-	506.5

Tiempos para el ciclo de bus de
Escritura a un puerto de I/O

Símbolo	Max	Min
t1	-	209.5
t2	-	119.0
t3	-	79.0
t4	183	132.0
t5	183	130.0
t6	170	-
t7	45	-
t8	-	11.0
t9	202	-
t10	142	-
t11	333	-
t12	-	4.0

Tiempos para el ciclo de bus de
DMA

Símbolo	Max	Min
t1	-	209.5
t2	-	119.0
t3	-	79.0
t4	183	132.0
t5	183	130.0
t6	170	-
t7	45	-
t8	-	11.0
t9	202	-
t10	142	-
t11	30	-
t12	-	-
t13	240	-

Más tiempos para ciclos de bus

Tabla III-1

Nota : Todos los tiempos están en nanosegundos.

III.5.3 Señales del Bus

Las señales mostradas en la figura III-9 son dirigidas a los seis conectores periféricos :

a) A0-A19 Estas líneas sirven para direccionar tanto memoria como dispositivos de Entrada/Salida (I/O) dentro del

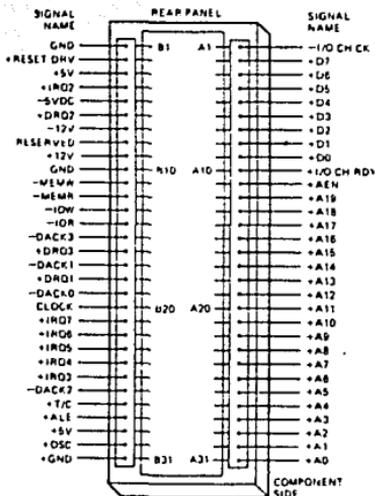


Figura III-9. Nombres y números de las terminales del slot periférico conectados a la tarjeta madre de la PC.

sistema. A0 es el bit menos significativo (LSB) mientras que A19 es el más significativo (MSB). Las señales que van a estas líneas pueden ser generadas tanto por el microprocesador como por el control de DMA.

b) AEN (*Address Enable*) Esta línea se usa para desconectar al microprocesador y a otros dispositivos del canal de I/O a fin de permitir que existan transferencias de tipo DMA.

c) ALE (*Address Latch Enable*) Esta señal es generada por el controlador de bus 8288 para guardar direcciones válidas provenientes del microprocesador. Está disponible para el canal de I/O como un indicador de direcciones válidas (cuando se usa con AEN).

d) -CARD SLCTD (*-Card Selected*) Esta línea se activa sólo si existe una tarjeta de expansión en el slot J8. Le indica a la tarjeta del sistema que la tarjeta ha sido seleccionada y que los manejadores (*drivers*) del sistema se deberán ser direccionados ya sea para leer o para escribir al slot J8. Los conectores desde J1 hasta J8 poseen esta señal, sin embargo el sistema sólo utiliza la del slot J8.

e) CLK (*Clock*) El reloj del sistema es una señal de 1/3 de la del oscilador con un período de 210 ns (4.77 MHz para la IBM PC). El reloj tiene un ciclo de trabajo de 33%.

f) D0 - D7 Son los bits de datos del 0 al 7.

g) -DACK0 a -DACK3 (-DMA Acknowledge 0 to 3) Se usan estas líneas para reconocer las solicitudes de DMA (DRQ1 a DRQ3) y de refresco de la memoria dinámica del sistema (-DACK0).

h) DRQ1 - DRQ3 (DMA Request 1 to 3) Estas líneas son solicitudes sobre el canal asincrónico usadas por los dispositivos periféricos para ganar el servicio del DMA.

i) -I/O CH CK (-I/O Channel Check) La utilidad de esta línea es permitir detectar errores de paridad en la información de la memoria o de los dispositivos periféricos, presente en el canal de I/O.

j) I/O CH RDY (I/O Channel Ready) Normalmente indica que el canal de I/O está listo para transmitir datos. Permite que dispositivos de baja velocidad utilicen el canal de I/O sin mayor dificultad. La señal nunca debe estar desactivada por más de 10 ciclos de reloj.

k) -IOR (I/O Read) Se emplea como un comando que le indica a un dispositivo de I/O que dirija su información hacia el bus de datos. También pueden mandar datos tanto el μ P como el controlador de DMA.

l) IRQ2-IRQ7 (Interrupt Request 2 to 7) Son las líneas de solicitud de interrupción que van al 8259A para que los dispositivos de las tarjetas puedan interrumpir al 8088 si es necesario.

m) -MEMR (Memory Read) Es la línea que le indica a la memoria, al μ P o al controlador de DMA que manden información al bus de datos.

n) -MEMW (-Memory Write) Esta línea le indica a la memoria que almacene los datos presentes actualmente en el bus de datos. Puede ser manejada por el μ P o por el controlador de DMA.

o) OSC (Oscillator) Es un reloj de alta velocidad (14.3181 MHz), con un ciclo de trabajo del 50%.

p) RESET DRV (Reset Drive) Se utiliza para reinicializar la lógica del sistema en caso de una falla de energía.

q) T/C (Terminal Count) Genera un pulso cuando se llega a la cuenta terminal para cualquier canal de DMA.

III.5.4 Capacidades del Bus del sistema para manejo de carga y datos

Una de las consideraciones más importantes cuando se une un diseño al bus del sistema es la capacidad de manejo de señales, y carga del bus. Para señales de salida del bus, se deberá determinar si hay suficiente capacidad de manejo de corriente para que el diseño funcione adecuadamente. En caso contrario, será necesario incrementar la potencia de la señal. En el caso de las señales de entrada al bus, es necesario determinar si el diseño cuenta con la capacidad de manejo suficiente para llevar la señal en forma correcta

III.5.4.1 Capacidad de manejo de señales del Bus del sistema

Hay dos especificaciones en las señales de salida del bus que son interesantes, IOL e IOH. La primera se refiere a la máxima corriente que puede ser recibida en el manejador (*driver*) de salida cuando existe un nivel lógico bajo, mientras que IOH se aplica cuando el nivel lógico es alto. Se muestra una tabla con las capacidades de los

drivers de salida existentes para los 5 slots de I/O. La capacidad de manejo remanente para un diseño específico en un slot corresponde a la diferencia entre los valores dados en la tabla general menos la suma de los valores tomados por las tarjetas de los slots ocupados.

III.5.4.3 Capacidad de manejo de la corriente del Bus del sistema

Existen dos especificaciones para las señales de salida del bus, IOL e IOH, donde IOL se refiere a la máxima corriente que puede pasar por el manejador de salida a un nivel bajo, mientras que IOH es la máxima corriente que puede llevar el manejador a un nivel lógico alto. La tabla III-3 lista la capacidad de los manejadores de salida tal y como existen en los cinco slots de E/S. Nótese que algunas señales del bus son utilizadas en la tarjeta del sistema antes de llegar a los slots.

Señal de salida del Bus	IOL (mA)	IOH (mA)
D0 - D7	23.6	-14.96
A19 - A16	7.2	- 2.46
A14 - A15	21.2	- 2.51
A13	23.2	- 2.56
A0 - A12	23.4	- 2.56
IOR, IOW, MEMR, MEMW	23.8	- 4.98
CLK	23.2	-14.96
AEN	24.0	-15.00
DACK0	24.0	-15.00
DACK1	3.2	- 0.20
DACK2, DACK3	2.8	- 0.18
ALE	14.8	- 0.94
RESET DRV	8.0	- 0.40
T C	8.0	- 0.40
OSC	5.0	- 1.00

Tabla III-3

III.5.4.3 Carga del Bus del sistema presentada en los slots

La carga mostrada en la tabla III-4 representa el menor nivel de corriente requerido por el driver para establecer una señal lógica de nivel bajo aceptable. A esta carga se la llama IIL, o corriente de entrada de nivel bajo. El circuito de manejo debe, también, proveer de corriente suficiente en un nivel alto para el circuito de recepción del bus. A esta se le llama corriente de entrada de nivel alto, o IIH.

Señal del Bus	IIL	IIH
D0 - D7	-0.4	0.04
I/O CH CK	-0.4	0.02
I/O CH RDY	-0.4	0.02
IRQ2 - IRQ7	-0.01	0.01
DRQ1 - DRQ3	-0.01	0.01

Tabla III-4

III.5.4.4 Carga capacitiva del bus

Algo que se debe tomar en cuenta, cuando se diseña en base al bus del sistema, es la carga capacitiva en las señales de salida del bus. Conforme se acumula carga, se incrementa la capacitancia que ve el manejador (*driver*). Conforme esto ocurre, la señal comienza a distorsionarse y a retardarse. Cada carga agrega de 10 a 20 pF de capacitancia en el bus. Aquellas señales con valores superiores a 200 pF afectarán, en general, a las señales del bus.

III.5.5 Características mecánicas y de potencia del bus del sistema

A continuación se mencionan algunas de las características que debe cumplir una tarjeta que se inserte en alguno de los slots de la IBM PC.

III.5.5.1 Slots para tarjetas en el Bus del sistema.

En la tarjeta madre del sistema existe espacio y conectores para manejar hasta cinco tarjetas. Los conectores pueden manejar hasta 62 señales asociadas a la tarjeta, 31 de cada lado de la tarjeta. Las terminales de conexión tienen una separación de 1/10 de pulgada. Cada slot es capaz de aceptar una tarjeta, con componentes y conexiones, de aproximadamente 10.42 cm por 33.02 cm. El slot 2 es capaz de aceptar una tarjeta un poco mayor, esto es posible debido a que la tarjeta madre de la Unidad del Sistema no tiene componentes bajo esta tarjeta, lo cual le permite extenderse hacia abajo. Los slots tienen una separación entre sí de una pulgada.

III.5.5.2 Dimensiones de la tarjeta

Se muestra un dibujo (fig. III-10) con las dimensiones máximas que se pueden utilizar para una tarjeta, esta debe estar construida de material con un ancho mínimo de 0.06 pulgadas para asegurar buen contacto en los conectores. Además, las terminales deben ser terminados con o para asegurar un contacto estable y perdurable.

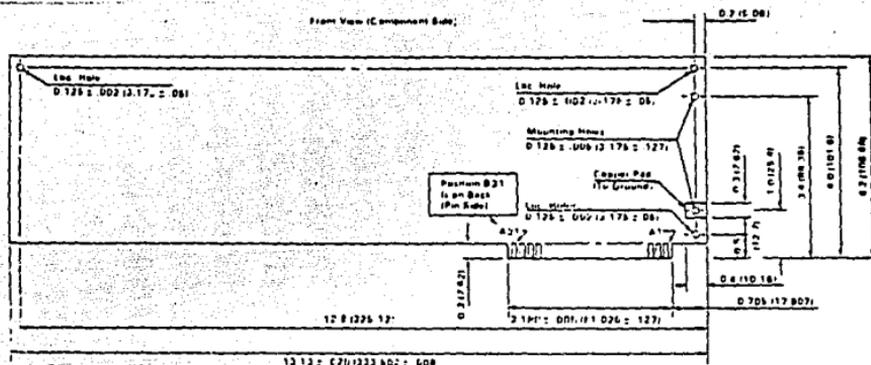


Figura III-10. Dimensiones de la tarjeta.

III.5.5.3 Potencia de la unidad del sistema

La tabla nos muestra los cuatro niveles disponibles en las terminales de señal en cada slot, la potencia total dada por la fuente de poder, sus tolerancias y el voltaje total.

Potencia de DC	Max (Vcd)	Min (Vcd)	Corriente (amps)	Potencia (Watts)	Corriente Típica en Slot (A)
+ 5 Vcd	5.25	4.80	7.0	35.0	0.70
- 5 Vcd	5.50	4.60	0.3	1.5	0.03
+12 Vcd	12.6	11.52	2.0	24.0	0.10
-12 Vcd	13.2	10.92	0.25	3.0	0.05

Tabla III- 5

III.5.5.4 Desacoplamiento de potencia

Un problema común en el diseño de circuitos es el de tener una distribución y desacoplamiento de potencia incorrectos. Debido a que la mayoría de los elementos electrónicos tienen requerimientos de potencia muy variables, dependientes de la operación que se esté ejecutando en un momento dado, requieren de un dispositivo de desacople. El propósito de éste es el de satisfacer los requerimientos de potencia del dispositivo conforme se van presentando, a fin de que esta no tenga que llegar directamente de la fuente de poder. Debido a que el sistema de potencia con sus conexiones y cables agrega inductancia a la fuente de poder, el sistema no puede responder

rápidamente a requerimientos transitorios de un orden alto. Para resolver este problema se emplean capacitores de desacople en puntos clave a fin de que la potencia transitoria pueda ser obtenida de ellos. Para fluctuaciones de potencia grandes , pero de baja frecuencia, se emplean grandes capacitores. En cambio, para requerimientos de transición de potencia menores en intensidad pero de alta frecuencia, los capacitores empleados son más pequeños.

IV. SISTEMAS DE ADQUISICION DE DATOS (SADs).

El objetivo de un Sistema de Adquisición de Datos (SAD) es el de recopilar un conjunto de señales provenientes del mundo real (analógicas), convertirlas a digitales (maneables por un sistema de procesamiento) para analizarlas y, con los resultados del análisis, ejercer funciones de control, ya sea sobre la o las señales que alimentan al SAD, o bien sobre el sistema que las produce. La referencia que se hace a la palabra "datos" y no a "señales" se debe a que se considera que dichas señales llevan información útil que se puede emplear, principalmente, en diversos sistemas de tipo digital, los cuales manejan información en forma de dígitos binarios (bits).

Los sistemas de adquisición de datos se usan para medir y registrar señales en dos formas básicas :

- a) Señales originadas de la medición directa de cantidades eléctricas; estas pueden incluir voltajes de cd y ca, frecuencia e impedancia y que se encuentran fácilmente en áreas tales como las de pruebas de componentes electrónicos, estudios ambientales o análisis de calidad.
- b) Las señales originadas en los transductores eléctricos, que convierten señales físicas en eléctricas.

Los sistemas de instrumentación se pueden clasificar, de acuerdo al manejo que hacen de las señales eléctricas en dos categorías principales : sistemas analógicos y sistemas digitales. Los primeros tienen que ver con la información de las mediciones de señales continuas tanto en tiempo como en amplitud. Los sistemas digitales manejan información en forma discreta en tiempo y en amplitud, esto es una función que sólo tiene valores en determinados instantes de tiempo, consiste en una cantidad de pulsos de duración finita cuya relación de tiempo contiene información acerca de la naturaleza de la cantidad.

Existen otros criterios para clasificar a los SADs, los cuales se muestran en el siguiente cuadro sinóptico :

- | | |
|---|--|
| a) Por el manejo de las señales eléctricas | { Analógicos
Digitales
Híbridos |
| b) Por su distancia a la fuente de la señal | { Remotos
Locales |
| c) Por el número de señales que manejan | { Un canal
Multicanal |
| d) Por la forma en la que se controla al sistema. | { Control centralizado
Control lineal
Control autónomo |
| e) Por su dependencia de otro sistema | { Integrados a un sistema
Aislados |

La operación de un SAD de tipo digital consta de tres etapas principales : a) Conversión analógico-digital, b) Análisis, c) Conversión digital- analógica y d) Almacenamiento en algún dispositivo de memoria secundaria. A su vez, la etapa de análisis se puede dividir en :

1. Medición de voltajes, corrientes o frecuencias.
2. Comparación de los valores medidos contra valores deseables o contra otros valores medidos anteriormente.
3. Transformación de las señales, ya sea del dominio del tiempo al dominio de la frecuencia o viceversa.
4. Filtrado de la señal para eliminar el ruido.
5. Visualización, ya sea de la señal en su forma original o modificada.

Un sistema de adquisición de datos digital puede incluir algunos o todos los elementos mostrados en la figura IV.1.

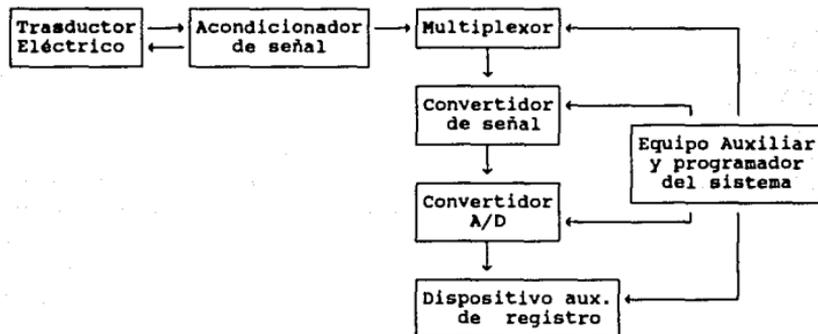


Fig. IV.1 Diagrama de bloques de un SAD

La operación funcional esencial dentro de un sistema digital incluye la manipulación de señales analógicas, la medición, conversión y manipulación de los datos digitales así como el control y programación internos de los dispositivos de control del sistema. La función de cada uno de los elementos del sistema de la figura XX.1 se enuncia a continuación :

1. **Trasductores eléctricos:** Transforman parámetros físicos a señales eléctricas aceptables por el sistema de adquisición. Algunos parámetros típicos incluyen temperatura, presión, velocidad, etc. Por otra parte tenemos que cantidades eléctricas, tales como voltaje o frecuencia se pueden medir directamente, esto es, sin necesidad de transductores.

2. **Acondicionadores de señal :** Generalmente incluye el circuito de soporte para el transductor. Este circuito puede suministrar potencia de excitación y también puede amplificar o atenuar la señal a fin de acoplar las impedancias.

3. **Multiplexor** : Acepta señales de entrada múltiples y secuencialmente las conecta a un instrumento de medición.

4. **Convertidor de señal** : Transforma la señal analógica a una forma aceptable para el convertidor analógico-digital. Un ejemplo de esto es un amplificador de voltajes de bajo nivel.

5. **Convertidor analógico - digital (A/D)** : Este es un dispositivo electrónico que realiza una transformación de cualquier señal eléctrica analógica a un patrón binario susceptible de ser manejado por un sistema digital de cómputo. Para lograr la conversión de una señal continua en una señal digital, primeramente se divide el intervalo de la señal continua, en el dominio del tiempo, entre varios valores discretos, a este proceso se le llama "muestreo".

Las muestras que se toman de una señal analógica son por sí mismas señales analógicas que varían en forma continua con el tiempo, sin embargo, en una representación digital esta variación no es continua por lo que la representación digital de cada muestra difiere en el bit menos significativo de los empleados en dicha representación. De lo anterior se deriva que las muestras analógicas en señales digitales son tan solo una buena aproximación. A este proceso de digitalización se le conoce como cuantización. Cuantización es el nombre del proceso que asigna o fija un número discreto de valores para una señal continua. Codificación es el nombre del proceso que asigna un patrón binario a cada uno de los valores discretos, completando así el proceso de conversión de una señal analógica en una señal digital. El código binario puro o sistema binario de numeración es el código más comunmente empleado en el proceso de codificación, sin embargo, existen aplicaciones particulares como lo es la transmisión de datos, en la que se requieren códigos como el ASCII, el Gray o el BCD.

El tiempo de respuesta de un convertidor A/D es crítico para algunas aplicaciones. Por ejemplo, para muestrear señales que cambian rápidamente con el tiempo. El nivel de dicha señal puede cambiar en forma significativa a partir del inicio al final del proceso de conversión. El cambio puede causar grandes errores de medición. Para solucionar este tipo de situaciones existe cierto tipo de convertidores muy rápidos llamados *flash converters*.

6. **Equipo auxiliar** : Esta sección contiene los instrumentos para la programación del sistema y el procesamiento de los datos digitales. Algunas funciones auxiliares típicas incluyen la linealización y comparación de la señal contra ciertos límites máximos, o bien, contra valores medidos con anterioridad. Estas funciones se pueden realizar por medio de instrumentos individuales o mediante un sistema digital de cómputo, lo cual quiere decir que al mencionar el equipo auxiliar se habla tanto de circuitos digitales de uso específico como de programas para el manejo de información. Estos últimos incluyen los algoritmos para el procesamiento digital de señales (DSP) que permiten la transformación de señales en el dominio del tiempo al dominio de la frecuencia y viceversa, o bien para el filtrado de señales.

7. **Dispositivos auxiliares de registro** : Registran la información digital en el monitor de video, en la unidad de disco, en la impresora o en una combinación de estos sistemas. Se diferencian de los mencionados en el punto 6 dado que la palabra registro sólo hace mención a la forma de mostrar o almacenar la información y no a su

procesamiento.

Los sistemas de adquisición de datos se usan en un número, cada vez más creciente, de aplicaciones en una variedad de áreas científicas e industriales, tales como la biomedicina, la telefonía o la telemetría entre otras. Antiguamente los sistemas analógicos se empleaban para procesos físicos con una variación muy lenta (ancho de banda angosto), sin embargo, actualmente las tasas de adquisición y muestreo de datos a altas velocidades le permiten a los sistemas digitales competir con los analógicos. Los sistemas digitales van en complejidad desde la medición de un voltaje de cd en un solo canal hasta un sistema multicanal sofisticado que mide un gran número de parámetros de entrada, los compara contra límites preseleccionados y realiza cálculos y decisiones sobre la señal de entrada. Los SADS digitales son en general más complejos que los sistemas analógicos, ambos en términos de la instrumentación involucrada, el volumen y complejidad de los datos de entrada que ellos pueden manejar.

Por otra parte, así como la información analógica se puede convertir a digital, la información en forma digital se puede convertir a una señal analógica, tal como un voltaje o una corriente, la cual se puede emplear como una realimentación para el control de un proceso.

A continuación se ennumeran las características deseables que debe tener un SAD :

1. Capacidad de manejo de un amplio rango de voltajes, corrientes y/o frecuencias, tanto en la etapa de entrada como en la de salida.
2. Baja distorsión (resultante de un amplio ancho de banda).
3. Rapidez de proceso.
4. Alta eficiencia en todos sus algoritmos.
5. Bajo consumo de potencia.
6. Alta capacidad de almacenamiento.
7. Costo accesible.
8. Portabilidad.
9. Interfaz sencilla con el usuario.
10. Compatibilidad con computadores comunes en el mercado como lo podría ser la IBM PC, ya sea para compartir información o para el caso en el que el SAD sea huésped de la computadora.
11. Alta eficiencia en el manejo de recursos, sobre todo en el caso de ser un sistema huésped de otro más general.
12. Alto rechazo a ruido.

IV.1 SADS basados en un Sistema Digital de Cómputo tipo IBM PC.

Un sistema de adquisición de datos que cumpla con las características previamente mencionadas puede ser desarrollado en base a un sistema digital de cómputo de amplia utilización tal como lo es la microcomputadora IBM PC. La utilización de una PC en una aplicación de este tipo involucra el uso de ciertas características de la computadora. En primer lugar, la PC puede estar ejecutando el análisis y enviando resultados a la pantalla o a la impresora. Se puede mostrar información relativa a un ciclo, tal como los valores medidos, los valores de estabilidad, etc. Por otra parte se pueden mostrar gráficas de los valores medidos contra el tiempo o contra la frecuencia. De

esta forma la PC se transforma en un auxiliar en el control de funciones dado que provee la información necesaria para el análisis de la señal que es muestreada.

Para poder emplear adecuadamente una herramienta tan poderosa como es la PC es importante considerar tanto los elementos de software como de hardware. La elección de software afecta la implementación del prototipo a diseñar. El sistema operativo elegido debe soportar ejecución de programas en tiempo real, debe ser eficiente en operaciones de E/S y debe ser ampliamente conocido. Por otra parte, cuando se va a elegir el lenguaje de programación, lo más importante es verificar que las entradas y las salidas sean direccionadas adecuada y fácilmente.

Por lo que se refiere al hardware, la arquitectura del sistema define las características de acuerdo a las cuales se procesará la información, cómo llegará al sistema y cómo saldrá de él. Las dos formas usuales para comunicar una computadora con el mundo real mediante un subsistema de interfaz es mediante interfaces serie o paralelo.

Todo μP tiene una estructura de bus de datos paralelo. Esta estructura provee medios bien definidos para comunicaciones locales entre el procesador, la memoria y los periféricos. Por ejemplo, como se mencionó en el capítulo III, dentro de la IBM PC, el 8088 tiene un bus interno y uno externo, y se tiene además el bus IBM PC. Una vez que se ha diseñado y conectado un dispositivo compatible con el bus seleccionado, la comunicación con éste es controlada por el software de la PC.

Una forma de realizar la comunicación entre el dispositivo de proceso de datos y la PC es mediante una interfaz de mapa de memoria. Recordemos que el mapa de memoria es una tabulación de todas las localidades de memoria que un procesador puede direccionar así como de los dispositivos que están presentes en estas localidades. Para comunicarse con dispositivos "mapeados" la computadora necesita conocer la dirección de inicio (base), el número de localidades que ocupa, la función de cada localidad, el formato de los datos comunicados y los tiempos de comunicación requeridos para una operación adecuada.

Alternativamente, para realizar esta comunicación, se pueden emplear las señales de DMA que se localizan en los conectores de expansión de la IBM PC. Las terminales de solicitud de DMA permiten a la tarjeta periférica solicitar el uso de los buses.

IV.1.1 Consideraciones para el diseño

El costo de implementar una aplicación basada en una tarjeta de interfaz con el mundo real con una PC incluye el costo de su hardware y de su software, el costo inherente a configurar, "alambrar" o conectar los dispositivos y el costo de realizar los programas que coordinen al sistema.

IV.1.2 Muestreo de datos

Un sistema de medición maneja por lo general una señal que es el voltaje analógico de alguna variable física. Para registrar perfectamente dicha señal, se debería conocer su valor exacto en todo momento, lo cual no es práctico dado que bastará muestras de la señal con una frecuencia y exactitud suficiente como para poder tener una

imagen cercana a la realidad. Sin embargo ¿Cuál frecuencia de muestreo es suficiente? Existen varios criterios que intentan responder a esta pregunta.

Aún con una alta tasa de muestreo no se garantiza una reproducción exacta de la señal (recordemos que una señal analógica tiene un número infinito de valores). La exactitud depende también de la resolución de los ADCs. La elección de la tasa de muestreo es una parte realmente crítica en la planeación de un sistema computarizado de proceso de señales dado que esta elección marca los límites por lo que respecta a cuánto proceso puede hacer el sistema. El hecho de aumentar al doble una tasa de muestreo dada lleva consigo dos situaciones adversas: en primer lugar el tiempo de atención al ADC se multiplica por 2, así como el tiempo de almacenamiento de datos, etc., en segundo lugar el tiempo de proceso se puede elevar a más del doble. Así, se deduce que se debe elegir la tasa de muestreo menor que permita una resolución aceptable dadas las características de la señal.

Los errores comienzan a aparecer cuando el sistema monitorea señales que tienen componentes de frecuencia mayores a dos veces la frecuencia de muestreo. Se presenta un fenómeno de sobreposición cuando la tasa de muestreo es ligeramente distinta a algún múltiplo de una componente de alta frecuencia de la señal. Cuando eso sucede, aparece esporádicamente una señal de baja frecuencia. Aún si no se dá la sobreposición, frecuencias por encima de la de muestreo aparecen como ruido en los datos muestreados. La solución a estos problemas no es aumentar la tasa de muestreo sino decrementar el ancho de banda de la señal antes de muestrearla.

IV.1.3 Procesamiento Digital de Señales.

Dos grandes ventajas del procesamiento digital sobre el analógico son exactitud y estabilidad. Una vez que el valor de un dato está presente en un sistema digital, puede ser mantenido o manipulado sin pérdida de exactitud. Es extremadamente difícil diseñar circuitos analógicos de gran exactitud; las propiedades de rechazo a ruido, linealidad o corrientes parásitas son un reto constante.

Así mismo los cálculos que requieren acumular datos durante largos períodos de tiempo son especialmente difíciles de implementar con circuitos analógicos. Dada esta característica, las tareas de procesamiento de señales hacen posible recuperar datos obtenidos previamente y efectuar cálculos con respecto a un evento basándose en los valores previamente muestreados.

Este procesamiento se puede realizar en una computadora digital de propósito general, sin embargo existen también microprocesadores diseñados específicamente para procesamiento de señales los cuales tienen, entre otras características, multiplicadores implementados en hardware y tiempos de instrucción muy cortos.

IV.2 CIRCUITOS PARA EL PROCESO DIGITAL DE SEÑALES (DSPs)

El hecho de filtrar una señal ha sido hasta ahora una tarea que puede llegar a ser muy complicada, sobre todo cuando se desea realizar una implementación física de filtros de un orden muy alto o que funcionen en base a algoritmos con fundamentos teóricos muy complejos. Hoy, sin embargo, gracias a microprocesadores que se dedican al proceso digital de señales, estos filtros pueden desarrollarse en un solo circuito.

Básicamente, un circuito DSP es un μP cuya arquitectura ha sido optimizada para procesar datos muestreados a alta velocidad. Este tipo de circuitos ejecuta operaciones aritméticas a una velocidad mucho mayor a la de un μP normal. Esta arquitectura ha sido diseñada para explotar al máximo las tasas de recepción de datos en el proceso mediante el traslape (*pipelining*) del flujo de datos.

Ahora bien, para elegir un circuito DSP que satisfaga los objetivos de este proyecto se deberán tomar en cuenta varias características que iremos describiendo a continuación. Sin embargo, es importante hacer notar que un factor de interés radica en el costo de la implementación del prototipo.

En algunas aplicaciones, la velocidad con la que se procesan las señales es menos importante que la precisión al representar los datos y los coeficientes que caracterizan a los filtros digitales. Con filtros recursivos o de respuesta a impulso infinito (IIR) en los cuales la salida retroalimenta al filtro para el cálculo siguiente, los errores de truncamiento o de redondeo pueden decrementar la eficiencia del sistema que emplea estos filtros.

Un elemento clave en un circuito de DSP típico (ver fig. IV.2) es un acumulador y multiplicador de arreglos rápido que permite que una operación de multiplicación y acumulación sea ejecutada en un solo ciclo de reloj que, comparado con los 25 ciclos de reloj que este proceso llevaría para un μP normal, nos da idea de lo poderoso que es un circuito DSP. Un acumulador y multiplicador de punto fijo típico ejecuta una multiplicación de 16×16 bits y suma el resultado de 32 bits con el contenido del registro acumulador de 32 bits en un solo ciclo de instrucción. Además, este tipo de circuitos puede contener memorias de tipo *cache* y normal para poder guardar datos e instrucciones traslapados.

El hecho de tener múltiples canales (*buses*) y memorias también afecta la eficiencia de un sistema basado en un DSP. Un DSP típico tiene dos memorias de datos y dos buses de datos. Con ellos puede entregar los dos operandos requeridos para la función ya descrita de acumulación y multiplicación. Al contrario de los microprocesadores normales, los cuales guardan tanto instrucciones como datos en la misma memoria, la mayoría de los DSPs emplean una arquitectura tipo Harvard, con memorias de programa y de datos separadas. De esta forma, los datos estáticos como los coeficientes de los filtros pueden guardarse en el tipo de memoria de programa que es más lenta y barata, y entonces ser transferida cuando se requiera a la memoria de datos que, aunque es más pequeña, es más rápida.

1. Esta es una memoria de poca capacidad pero muy rápida que se coloca entre una memoria primaria grande y el microprocesador; de esta forma se puede tener un control sobre bloques de datos adyacentes a datos o instrucciones accedidos recientemente por la memoria más grande, lo cual agiliza los tiempos de acceso.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

Existen diferentes caminos para incrementar la eficiencia del sistema mediante el aumento de la tasa de acceso de datos (ancho de banda de memoria). Esto se logra usualmente mediante múltiples memorias y sus correspondientes canales. Varios circuitos de DSP tienen memorias internas duales, así como acceso a una memoria de datos externa. Normalmente las memorias internas contienen de 128 a 512 palabras de 16 bits, adecuadas para ejecutar DSPs. Sin embargo, existe una desventaja a la que no se enfrentan los usuarios de μ Ps de propósito general, pues debe tenerse especial cuidado en el tamaño y la localización de los arreglos de datos, así como de los ciclos iterativos en los programas.

Otra forma de aumentar la eficiencia es emplear memorias externas que sean lo suficientemente rápidas para poder leer de ellas instrucciones y operandos en un solo ciclo de reloj. Esto requiere de RAMs estáticas con una capacidad mínima que puede ir de 16 Kbits a 64 Kbits y un tiempo de acceso mínimo de 25 a 70 ns. El problema aquí reside en el costo de estas memorias que pueden llegar a costar de 6 a 9 veces más que las RAMs dinámicas empleadas por los μ Ps de propósito general.

La mayoría de los circuitos de DSP tienen una señal de reconocimiento que se activa después de un direccionamiento. El procesador espera esta señal antes de continuar un proceso. Se puede tomar ventaja de esto y emplear memorias menos rápidas pero más baratas, claro que el costo de esto es un decremento en la velocidad de operación que podría llegar a tener el sistema.

La arquitectura de un circuito de DSP también se tipifica por tener dos unidades aritmética y lógica, una para datos y otra específica para direcciones. Dado que varias de las instrucciones ejecutadas por los algoritmos de DSP emplean la unidad aritmética lógica especificada para datos, la eficiencia se incrementa substancialmente cuando se agrega una segunda unidad para el cálculo de direcciones.

El desarrollo de un algoritmo de DSP comienza generalmente con una simulación de punto flotante que arroje resultados satisfactorios. Entonces se emplea una implementación de punto fijo tal que sea eficiente y que no dañe la ejecución del algoritmo. Este paso es por lo común lento y difícil. Es por lo anterior que se han desarrollado circuitos de DSP que manejan operaciones con punto flotante.

Existen variaciones en la arquitectura de los circuitos de DSP. Una de ellas es la inclusión de memoria ROM interna para programa y/o datos que puede reemplazar a la RAM externa. Esto trae como ventaja un número menor de terminales y un costo menor. En general, los circuitos de DSP reducen el costo de los sistemas al tener altos niveles de integración. Además de esto, incluyen generalmente interfaces para dispositivos seriales, así como unidades con funciones especiales como son los timers.

Los circuitos de DSP se pueden agrupar por su función en circuitos de propósito general y de aplicación específica. Estos últimos son diseñados para ejecutar una función dada con mayor precisión y más rápido que un circuito de propósito general. Ejemplos típicos de esto son los circuitos para filtros digitales o para transformada rápida de Fourier, algoritmos que si bien se pueden realizar en μ Ps de propósito general, debido a las características del hardware de los DSPs se ejecutan mucho más rápido en estos que en los procesadores comunes.

También se puede categorizar a estos circuitos por su precisión y por sus tipos de aritmética. Existen circuitos de punto fijo con 16, 24 y 32 bits de precisión. Se han introducido recientemente circuitos con capacidad de manejo de aritmética de punto flotante. El diagrama de bloques de un circuito DSP se muestra en la figura IV.2.

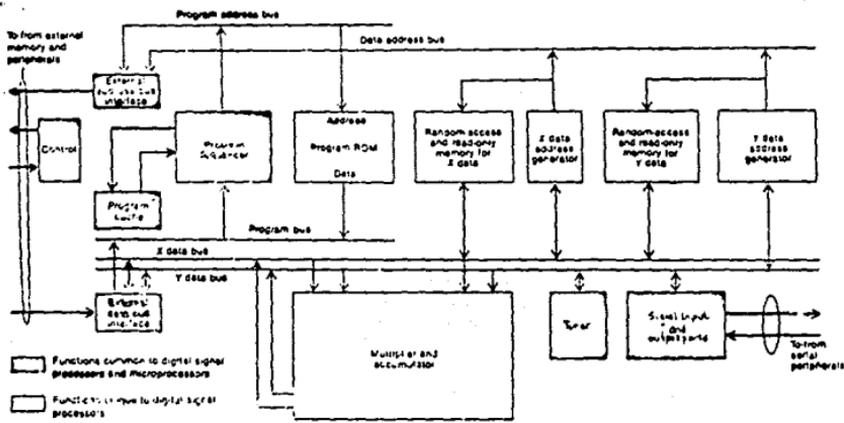


Figura IV-2 Las capacidades más representativas en este diagrama genérico de un DSP comparado con un μP de uso general son su acumulador multiplicador y sus memorias de bases múltiples.

Un circuito de DSP se puede integrar a un sistema en tres diferentes formas : a) como μP maestro, b) como μP esclavo y c) en multiproceso. La configuración más económica es la de μP maestro, empleada comúnmente en aplicaciones de alto volumen de información. En este caso se requiere de una circuitería de programación bastante sofisticada.

En la configuración de esclavo, el DSP actúa como periférico de un μP de propósito general. En este caso se le dejan al DSP las tareas que consumen más tiempo y los procesos más repetitivos, mientras que el control y las tareas de comunicación son ejecutadas por el procesador de propósito general. Esta arquitectura es muy empleada para DSPs de propósito específico.

La actuación más eficiente de un circuito DSP es cuando se emplea en un ambiente de multiprocesamiento. En un arreglo de este tipo se tienen por lo menos dos procesadores que envían y reciben señales de control de información mediante líneas de comunicación. Los procesadores también comparten datos de una misma memoria mediante una lógica de arbitraje de acceso.

En el siguiente apartado se describe el DSP utilizado en esta tesis, el TMS32010 de Texas Instruments.

IV.3 FAMILIA TMS320

IV.3.1.- Introducción.

Dentro de la amplia gama de circuitos para procesamiento de señales existentes en el mercado, uno de los conjuntos más empleadas en nuestro medio es el desarrollado por la compañía Texas Instruments y cuya denominación comienza con la clave TMS320. Esta "familia" está constituida por procesadores diseñados para realizar una amplia gama de aplicaciones a alta velocidad de procesamiento. La familia 320 puede realizar hasta 5 millones de instrucciones por segundo (5 MIPS), debido a su conjunto de instrucciones y a su arquitectura enfocada al uso del pipeline

La familia 320 usa la arquitectura Harvard (programas y datos separados en espacios de memoria diferentes), para hacerlo mas eficiente, aunque permite la transferencia de información entre espacios de datos y de programa.

IV.3.2.- TMS32010

IV.3.2.1.- Arquitectura.

Este microprocesador constituye la primera generación de procesadores digitales de señales de la familia 320.

El TMS32010 implementa en *hardware* funciones que típicamente se realizaban con *software*, como son las multiplicaciones, que este μP realiza utilizando complemento a 2. Usa un registro de corrimiento para introducir datos en la ALU. Tiene registros auxiliares configurables en modo de auto-incremento/decremento para la manipulación de tablas de datos.

El TMS32010 puede ser usado en dos modos, elegidos a través de la terminal del μP designada como MC/MP estos modos son:

- a) Modo microcomputadora (MC): en este modo el μP está equipado con 1536 palabras de 16 bits de ROM interna (solo TMS320M10), preprogramadas en la fábrica con programas enviados por el usuario. Se cuenta adicionalmente con 2560 palabras de ROM externa.
- b) Modo microprocesador (MP): el DSP usa 4096 palabras de ROM externa para almacenar instrucciones.

Las características mas sobresalientes del TMS32010 son:

- Memoria volátil de 144x16 de escritura y lectura de datos.
- Memoria no volátil de 144x16 de programa .
- Acumulador y ALU de 32 bits.
- Multiplicador rápido de 200 ns.
- Registro "tonel" para hacer corrimiento de bits en las palabras de datos que se envían a la ALU.
- Reg. de corrimiento para hacer corrimientos al acumulador cuando se almacena en RAM.
- Bus de datos de 16 bits para buscar instrucciones en memoria externa (16 bits) a alta velocidad.

- Stack de 4x12 que permite cambio de contexto (branch).
- Regs. de autoincremento/decremento para direccionamiento indirecto de datos.
- Interrupciones de vector simple.
- Oscilador integrado al chip.
- Ciclo de instrucción de 200 ns
- Tecnología NMOS

La arquitectura del TMS32010 se muestra a continuación :

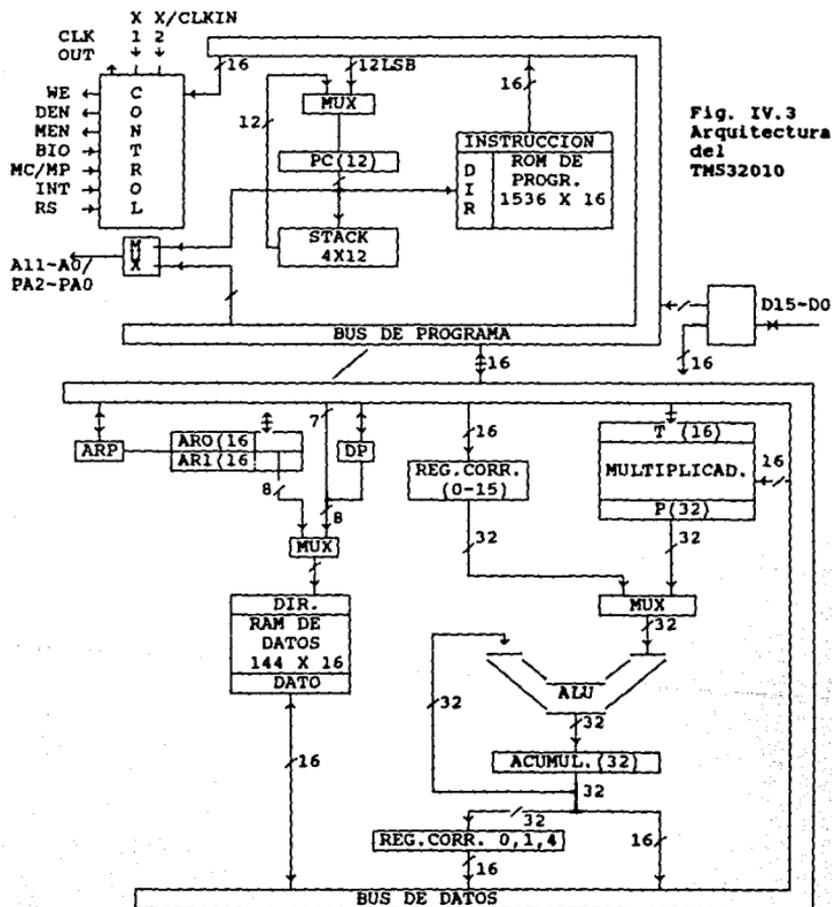


Fig. IV.3
Arquitectura
del
TMS32010

Se observa que el TMS32010 tiene una memoria interna RAM de 144X16 bits. Los operandos de las instrucciones se buscan en esta RAM, en este caso no son permitidas las búsquedas de datos fuera de esta memoria, pero si es posible efectuar transferencias de datos entre esta memoria y una externa (ver instrucciones IN, OUT, TBLR, TBLW en la sección IV.6.15.).

IV.3.2.2.- Elementos aritméticos.

Los elementos aritméticos básicos son:

- a) ALU
- b) Acumulador
- c) Multiplicador
- d) Registros de corrimiento

Las operaciones se realizan sobre un acumulador, al cual se le puede sumar o restar una cantidad. El resultado del acumulador se debe almacenar en la RAM de datos en forma separada, pues el acumulador es de 32 bits y las localidades de memoria son de 16 bits.

A continuación se describen los elementos aritméticos del μP :

- a) ALU.- opera con palabras de 32 bits. Puede realizar operaciones de suma, resta u operaciones lógicas. La siguiente tabla resume las operaciones lógicas.

Función	Resultado del acumulador	
	Acum. bits 31 a 16	Acum. bits 15 a 0
XOR	0000H \oplus ACC bits 31-16	dato mem. \oplus ACC bits 15-0
AND	0000H \otimes ACC bits 31-16	dato mem. \otimes ACC bits 15-0
OR	0000H $+$ ACC bits 31-16	dato mem. \oplus ACC bits 15-0

Tabla IV.1

Existen dos modos en los que puede trabajar la ALU:

- 1) Modo sobreflujo (*overflow*) encendido.- se habilita con la instrucción SOVM (véase sección 6.15 de este capítulo). En este modo, si existe sobreflujo, se carga el valor mas positivo o negativo proveniente de la ALU en el acumulador.
- 2) Modo sobreflujo apagado.- se habilita con la instrucción ROVM (véase la sección referida al software), en este caso si existe sobreflujo, el acumulador permanece sin modificar.

b) Acumulador.-

Este registro, que es de 32 bits almacena la salida del ALU y se divide en 2 partes: palabra baja (bits 0 a 15) y palabra alta (bits 16 a 31), mismas que pueden ser accedidas independientemente.

Es posible conocer el estado de sobreflujo del acumulador leyendo la bandera de sobreflujo del registro de estado (OV), la cual solo puede ser reestablecida por la instrucción BV (branch on overflow) o mediante la modificación directa del registro de estado.

A fin de tener un control sobre el estado de un programa en ejecución se pueden verificar otras condiciones del acumulador como son:

Instr.	Condic. de ACC verificada
BLZ	<0
BLEZ	<=0
BGZ	>0
BGEZ	>=0
BNZ	<>0
BZ	=0

Tabla IV.2

c) Multiplicador.- multiplica palabras de 16X16 bits y está compuesto de 3 unidades:

- 1) Registro T: registro de 16 bits que almacena el multiplicando.
- 2) Registro P: registro de 32 bits que almacena el producto.
- 3) Arreglo multiplicador.

Para usar el multiplicador se carga el multiplicando proveniente de la RAM de datos en el registro T, usando alguna de las instrucciones LT, LTA o LTD. A continuación se usa la instrucción MPY (multiply) o MPYK (multiply immediate).

Si se usa MPY el multiplicador es un número de 16 bits proveniente de la RAM de datos, si se usa MPYK se da implícitamente una constante de 13 bits, ésta es justificada a la derecha y tiene formato signado. Después de la instrucción MPY o MPYK el producto se almacena en el registro P y puede sumársele o restársele otra cantidad o cargarse al acumulador por medio de las instrucciones: APAC, SPAC, LTA, LTD o PAC.

Las instrucciones MPY y MPYK deben ser seguidas de instrucciones que combinan el registro P con el acumulador.

d) Registros de corrimiento.-

Existen dos tipos de registros: registros de contención (barrel shifter), y registros de corrimiento paralelos.

- 1) Registro de contención: Permite corrimientos a la derecha de 0 a 15 lugares en las palabras de 16 bits provenientes de la RAM de datos, que serán cargadas, sumadas o restadas al acumulador (instr. LAC, SUB, ADD). Este registro amplía la palabra de 16 a 32 bits, haciendo el corrimiento según los bits que se indican en la instrucción y conservando la representación positiva o negativa en complemento a 2 de la palabra de 16 bits original a la de 32 bits. Por ejemplo:

7EBCH con corrimiento de 4bits

(notese que es positiva en complemento a 2)

quedaría:

0007EBC0H

(notese que es positiva en complemento a 2)

8EBCH con corrimiento de 4bits

(notese que es negativa en complemento a 2)

quedaría:

FF8EBBC0H

(notese que es negativa en complemento a 2)

Es posible hacer el manejo de una palabra de 16 bits como si fuera de 32 (ADDS, SUBS, etc) en el momento de realizar las operaciones con este registro.

- 2) Registro paralelo: este registro se activa con la instrucción SACH (Store high order ACcUmulator with Shift). Esto hace un corrimiento a la izquierda de los 32 bits del acumulador, para posteriormente almacenar los 16 bits de mas alto orden en la RAM de datos. Este registro solo puede realizar corrimientos de 0, 1 o 4 lugares, lo cual se especifica en la instrucción misma.

Por ejemplo:

Suponiendo que: ACUMULADOR t1 A34B,78CDH.
 INSTR SACH 40,4

se realiza corrimiento de cuatro bits y almacena en la localidad 40H de la RAM de datos 34B7H.

IV.3.2.3. Memoria de datos.

Consiste en 144 palabras de 16 bits cada una, las cuales que se encuentran en la RAM del microprocesador. Todos los operadores no inmediatos se encuentran en esta RAM.

Las instrucciones TBLR y TBLW (TaBLE Read y TaBLE Write) permiten realizar intercambio de información que se localice en diferentes zonas de memoria; de memoria interna a memoria externa y viceversa. Estas instrucciones se ejecutan en 3 ciclos de reloj.

Otro medio de hacer transferencia de información entre el sistema y el mundo exterior es utilizar las instrucciones IN o OUT. IN lee datos de un periférico y los transfiere a la RAM de datos interna. Esta instrucción puede usarse junto con OUT para hacer la interfaz por I/O o por DMA con la PC (ver sección III.6.2 y III.6.3). Esta instrucción se realiza en 3 ciclos de reloj.

IV.3.3. Modos de direccionamiento.

El TMS32010 maneja tres formas de direccionamiento:

- a) Directo. En este modo, 7 bits de la palabra de la instrucción se concatenan con el apuntador de la página de datos a partir de la dirección de memoria de datos. En esta forma se implementa un esquema de paginación en el que la 1a. página de la memoria de datos contiene 125 palabras y la 2a. página contiene 16 palabras. En una aplicación típica las variables accedidas con poca frecuencia, como las empleadas en una subrutina de servicio de interrupción, son almacenadas en la segunda página.
- b) Indirecto. Este modo forma la dirección de memoria de datos a partir de los 8 bits menos significativos de uno de 2 registros auxiliares, ARO y AR1. La selección del registro auxiliar se hace mediante el apuntador de registro auxiliar (ARP). Estos registros se pueden incrementar ó decrementar automáticamente en forma paralela mediante la ejecución de cualquier instrucción indirecta, a fin de permitir la manipulación de tablas de datos en un solo ciclo.
- c) Inmediato. Las instrucciones que emplean este modo derivan datos a partir de parte de la palabra de la instrucción y no con base en los datos en RAM. La constante en todas las instrucciones inmediatas se puede referir a valores obtenidos a partir de un símbolo de referencia externo.

IV.3.4 Registros.

- a) Registros Auxiliares (AR).- Estos son usados para almacenamiento temporal, así como para direccionamiento indirecto (ver sección anterior) o control de ciclos iterativos.

El uso de estos registros para el control de ciclos se hace por medio de las instrucciones BANZ, (Branch if Auxiliary register Not Zero). En esta instrucción se usan 9 bits del bit 0 al bit 8 del registro como contador de autoincremento/decremento para direccionamiento indirecto, solo los 9 bits menos significativos son afectados. El contador es circular (i.e. incremento a 11111111 produce 00000000).

Estos registros pueden ser salvados y cargados integros (16 bits) en la RAM de datos con las instrucciones SAR (Store Auxiliary Register) y LAR (Load Auxiliary Register).

- b) **Registro Apuntador Auxiliar (ARP)** .- Este registro de un bit, es parte del registro de estado, e indica cual de los registros auxiliares está habilitado, según la siguiente tabla:

ARP	Registro habilitado
0	ARO
1	ARI

IV.3.5 Memoria externa para programa.

La memoria consiste en 4K palabras de 16 bits. El TMS320M10 tiene 1536 palabras de ROM interna, mientras que el TMS32010 no tiene ROM.

La terminal designada como MC $\overline{\text{MP}}$ permite trabajar con el TMS32M10 en dos modos: microcomputadora, que cuenta con 1523 localidades de ROM interna disponibles y microprocesador donde las 4K palabras posibles de direccionar están en memoria externa.

A partir de este punto se realizan todas las explicaciones del considerando que el TMS32010 está en modo microprocesador.

Para realizar el direccionamiento de memoria externa el DSP cuenta con doce terminales (A0 - A11). Cuando se busca una instrucción la terminal $\overline{\text{MEN}}$ (Memory ENable) generará una señal para habilitar la memoria externa y direccionarla. Esta señal nunca se da al mismo tiempo que las señales $\overline{\text{WE}}$ y $\overline{\text{DEN}}$.

IV.3.5.1 Inicio de ejecución.

Existen localidades de memoria predefinidas en las que se inicia la ejecución después cierta acción:

- Después de un **RESET** el DSP inicia la ejecución en la localidad 0 de memoria.
- Después de una interrupción se inicia la ejecución en la localidad 2 de memoria.

IV.3.6 Contador de programa (PC) y stack.

El PC y el stack permiten llamadas a subrutinas, interrupciones, y ejecutar las instrucciones TBLR y TBLW.

- a) **PC** .- Es un registro de 12 bits que contiene la dirección de la siguiente instrucción a ejecutarse. El PC se inicializa con cero en el momento de llevar la terminal del μP de restablecer (\overline{RS}) a un estado lógico cero.
- b) **Stack** .- Consiste en 4 registros de 12 bits apilados. La instrucción **PUSH** almacena los 12 bits menos significativos del Acumulador en la parte alta del STACK. **POP** devuelve el registro más alto del STACK al acumulador. Se pueden usar hasta 4 niveles de anidamiento de subrutinas si no se usan las instrucciones **TBLR** y **TBLW** y tres niveles si se usan estas instrucciones, sin que se pierda información del STACK. Además si se almacena un nivel más de información, el nivel más profundo del STACK se pierde. Si se ejecuta la instrucción **POP** con más niveles de los existentes, el valor del fondo del STACK se copia en los niveles más altos hasta llenar el STACK.

IV.3.7 Registro de estado.

Compuesto por 5 bits con la siguiente definición:

OV	OVM	INTM	ARP	DP
----	-----	------	-----	----

- OV** .- Sobreflujo (*overflow*) del acumulador. Si es 0 no hay sobreflujo en el acumulador. Si es 1 hay sobreflujo en el acumulador.
La instrucción **BV** (*Branch on overflow*) pone el byte **OV** en 0 y salta a una rutina de atención especificada.
- OVM** .- Bit de modo sobreflujo. Toma el valor de 0 lógico si el modo está deshabilitado, en caso contrario es un 1 lógico si está habilitado (ver sección IV.6.2.2 ALU). La instrucciones **SOVM** carga 1 en **OVM** y **ROVM** carga 0 en **OVM**.
- INTM** .- Bit de máscara de interrupción.- 0 interrupción habilitada. 1 interrupción deshabilitada.
Las instrucciones **EINT** carga **INTM** con 0 y **DINT** carga **INTM** con 1.
Solo estas instrucciones modifican **INTM** (no se puede cargar un valor de una palabra de memoria para modificarlo). Al ejecutar una interrupción **INTM** toma el valor de 1 antes de que la rutina de servicio se inicie.
- ARP** .- Registro apuntador auxiliar.- 0 selecciona **ARO**, 1 selecciona **ARI**.
Este bit se puede modificar con las instrucciones **MAR** y **LARP** o por instrucciones que hagan direccionamiento indirecto.
- DP** .- Apuntador de página de memoria.- 0 selecciona las primeras 128 palabras, 1 selecciona las últimas 16 palabras de la RAM interna de datos. Se puede emplear **DP** con las instrucciones **LDP** o **LDPK**.

El contenido del registro de estado puede ser almacenado en memoria de datos por medio de la instrucción SST (Store Status) en la página 1 de la RAM de datos interna, y puede ser restablecido en el registro de estado por la instrucción LST, esta no asume que se toma de la página 1, por lo que DP debe ser puesto en 1

IV.3.8 Funciones de entrada salida.

Las instrucciones IN y OUT, TBLW y TBLR permiten enviar o recibir 16 bits por el bus de datos.

El bus de datos permanece en alta impedancia excepto cuando \overline{WE} baja durante el primer ciclo de la instrucción OUT y en el segundo ciclo de la instrucción TBLW para indicar que se escriben datos.

Para la instrucción IN se usa la señal \overline{DEN} baja para indicar que se realiza una lectura. Esta es la única instrucción para la cual se habilita \overline{DEN} .

Los diagramas de tiempo de las instrucciones se muestran en las figuras IV.4 y IV.5

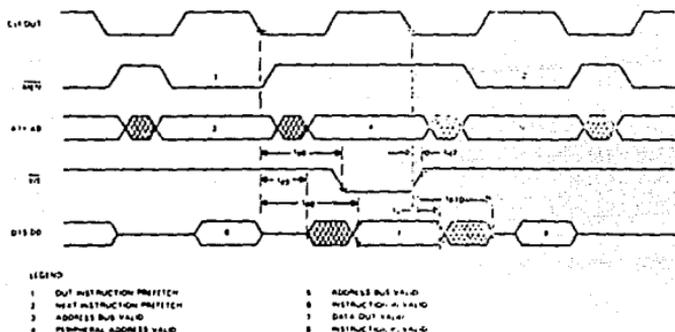


Fig. IV.4 Diagrama de tiempo para OUT

Para direccionar el puerto al que se escribe se usan los bits de dirección multiplexados PA2, PA1 y PA0, el resto de los bits (A11-A3) tiene un cero lógico.

Las instrucciones TBLR y TBLW permiten la transferencia de información entre la RAM interna de datos y la memoria externa. TBLR genera un \overline{MEN} para leer las palabras de memoria externa mientras que TBLW genera un \overline{WE} .

Los diagramas de tiempo se muestran en las figuras IV.6 y IV.7

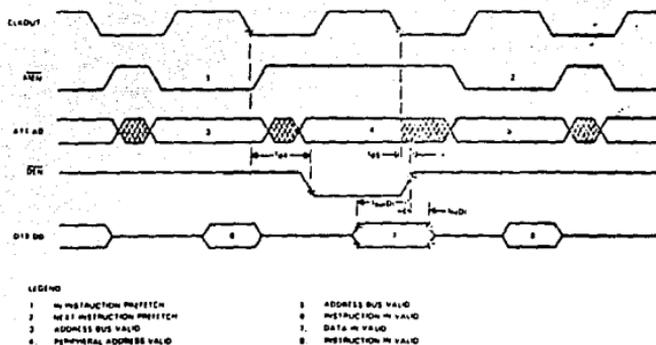


Fig. IV.5 Diagrama de tiempo para IN

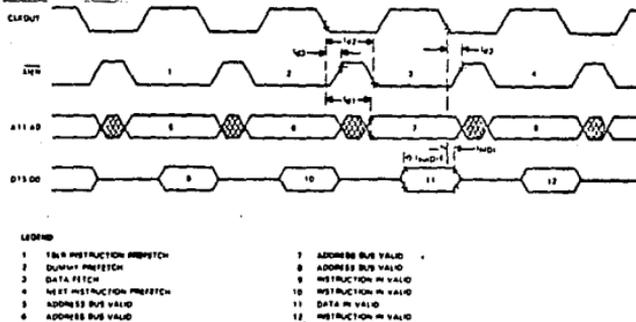


Fig. IV.6 Diagrama de tiempo para TBLR

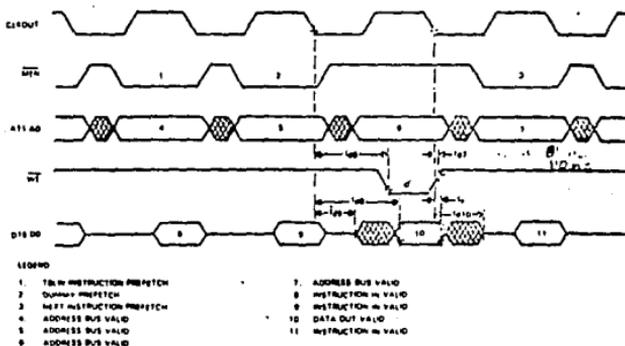


Fig. IV.7 Diagrama de tiempo para TBLW

IV.3.9 Decodificación del Bus de Direcciones.

Dado que \overline{MEN} , \overline{WE} y \overline{DEN} son mutuamente excluyentes, se deben tomar ciertas consideraciones para el diseño. Como las instrucciones OUT y TBLW usan solo \overline{WE} para indicar datos válidos, estas no pueden distinguirse con base en la habilitación de la señal. A menos que el bus de direcciones sea decodificado, la ejecución de TBLW escribirá datos a los periféricos y la ejecución de OUT reescribirá la memoria de programa en las localidades de 0 a 7. No importa la lógica de decodificación empleada, no será posible emplear TBLW para escribir únicamente a las localidades de memoria de programa de 0 a 7, esto se debe a que el bus de direcciones será idéntico para OUT y TBLW, y no habrá manera de distinguir entre ambas instrucciones.

IV.3.10 La terminal \overline{BIO}

Esta terminal maneja la prueba de bits y operaciones de salto. Cuando se activa la terminal, la ejecución de la instrucción BIOZ causará un salto. El estado de la terminal se registra cada ciclo de reloj. Esta terminal es útil para monitorear el estado de un dispositivo periférico o también para manejar interrupciones cuando se debe evitar interrumpir ciertos ciclos con tiempos críticos.

IV.3.11 Interrupciones.

Estas se activan al registrarse un nivel bajo en la terminal \overline{INT} . Si el registro de modo de interrupción (INTM) está activado, la señal de interrupción se valida mediante un "Procesador Interno de Interrupciones" (IIP) del DSP. El servicio a una interrupción será retardado en los siguientes casos :

- 1) Hasta el fin de la ejecución de una instrucción multiciclo.
- 2) Hasta que se completa la instrucción siguiente a MPY, MPYK ó EINT (cuando las interrupciones han sido previamente deshabilitadas).

Cuando la rutina de atención a interrupción comienza, el IIP manda una señal de reconocimiento, esto modifica el registro de interrupción, deshabilitando las interrupciones y poniendo en cero la bandera de interrupciones.

La figura IV.8 muestra la secuencia de instrucciones que ocurre una vez que se activa la interrupción.

IV.3.12 Reset

Esta función se activa cuando llega una señal baja a la terminal \overline{RS} durante por lo menos 5 ciclos de reloj. Las líneas de control \overline{DEN} , \overline{WE} y \overline{MEN} pasan a un estado lógico alto, y el bus de datos pasa a 3er. estado (alta impedancia). El PC y el bus de direcciones se ponen en 0 lógico. Las interrupciones se deshabilitan y el registro de bandera de interrupciones se pone en cero.

IV.3.13 Reloj/Oscilador

El uso del oscilador interno se lleva a cabo al conectar un cristal entre las terminales X1 y X2/CLKIN. La frecuencia de CLKOUT y el tiempo de ciclo del TMS32010 es de la frecuencia fundamental del cristal. El ciclo de trabajo es del 50 %.

INTERRUPT (E/T) THARG

Timing requirements over recommended operating conditions

PARAMETER		MIN	TYP	MAX	UNIT
t_{setup}	Set time E/T			10	ns
t_{hold}	Hold time E/T			10	ns
t_{setup}	Set time E/T, before CLKOUT			50	ns

NOTE: E/T is an asynchronous input and set hold times during a clock cycle. The set hold times are measured to the beginning of the clock cycle.

Interrupt timing



NOTE: Timing requirements are not indicated to set up to a low voltage of 0.8 volts and a high voltage of 2.0 volts unless otherwise noted.

IO (E/T) TIMING

Timing requirements over recommended operating conditions

PARAMETER		MIN	TYP	MAX	UNIT
t_{setup}	Set time E/T			10	ns
t_{hold}	Hold time E/T			10	ns
t_{setup}	Set time E/T, before CLKOUT			50	ns

NOTE: E/T is an asynchronous input and set hold times during a clock cycle. The set hold times are measured to the beginning of the clock cycle as shown on the timing diagram and similar conditions.

IO timing



Fig. IV.6 Diagrama de tiempo para interrupciones

IV.3.14 Descripciones de las terminales.

Señal	No.	E/S	Descripción
Vcc	30		Fuentes de Poder
Vss	10		Voltaje de alimentación (+ 5V) Referencia de tierra
X2/CLKIN	8	E	Relojes
X1	7	S	Entrada de cristal/Oscilador interno ó externo
CLKOUT	6	S	Entrada para oscilador interno Señal de salida de reloj
WE	31	S	Control
			Habilita escritura. Indica que existen dato válidos en el bus de datos.
DEN	32	S	Habilita datos. Indica que se acepten datos del bus de datos.
MEN	33	S	Habilita fetches de instrucciones en la memoria de programa. Habilita la memoria.
RS	4	E	Interrupciones
			Reset. Reestablece el DSP a su estado inicial.

<u>INT</u>	5	E	Interrupción (ver sección IV.6.11).
<u>BIO</u>	9	E	Control de salto para E/S (v. secc. IV.6.10).
Modos de memoria de programa			
MC/ <u>MP</u>	3	E	Modo μ C/ μ P (ver sección IV.6.2).
Bus de datos bidireccional			
D15	18	E/S	D15 (MSB) a D0 (LSB). Siempre está en alta impedancia, excepto cuando se activa \overline{WE} .
D14	17	E/S	
D13	16	E/S	
D12	15	E/S	
D11	14	E/S	
D10	13	E/S	
D9	12	E/S	
D8	11	E/S	
D7	19	E/S	
D6	20	E/S	
D5	21	E/S	
D4	22	E/S	
D3	23	E/S	
D2	24	E/S	
D1	25	E/S	
D0	26	E/S	
Bus de direcciones de memoria de programa y bus de direcciones de puertos.			
A11	27	O	Memoria de programa A11 (MSB) a A0 (LSB) y direcciones de puertos PA2 (MSB) a PA0 (LSB). Las direcciones A11 a A0 siempre están activas y nunca pasan a alta impedancia. PA2 a PA0 se se activan durante la ejecución de las instrucciones IN y OUT.
A10	28	O	
A9	29	O	
A8	34	O	
A7	35	O	
A6	36	O	
A5	37	O	
A4	38	O	
A3	39	O	
A2/PA2	40	O	
A1/PA1	1	O	
A0/PA0	2	O	

IV.4 REALIZACION DE LA INTERFAZ DEL SISTEMA DE ADQUISICION DE DATOS (SAD) Y LA PC.

Las formas en que se puede realizar la comunicación entre el sistema de adquisición de datos y la computadora PC son básicamente 3:

- 1.- Por una interrupción del Sistema de Adquisición de Datos (SAD) al 8088 de la PC.
- 2.- A través de DMA realizado por el SAD.
- 3.- Por registro de entrada/salida, por lectura o escritura a un puerto del 8088 al que esté conectado el SAD.

Para efectos de esta tesis sólo se usaron los métodos 1 y 3, los cuales se describen a continuación.

IV.4.1 Esquema general de intercambio de información del SAD y la PC por interrupciones.

Este esquema utiliza interrupciones dirigidas a la terminal INTR del μP 8088 de alguna de las terminales del IRQ de la ranura de expansión (SLOT) de expansión la PC a través del controlador de interrupciones 8289A-5, la descripción de este tipo de interrupción se efectuó en la sección III.4.3.

El sistema BIOS se encarga de inicializar el 8259, programándolo por medio del envío bytes de control cuando la PC es encendida. Esta programación dispone que la solicitud de interrupciones por medio de las señales IRQ que entran al 8259 tengan la siguiente prioridad y uso:

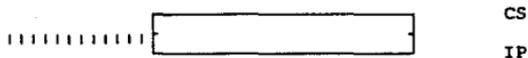
PRIORIDAD	NOMBRE	USO
Mayor	IRQ0	Salida del canal 0 del reloj 8253-5.
	IRQ1	Intrr. de escrutinio de código del teclado.
	IRQ2	No usada.
	IRQ3	No usada.
	IRQ4	Puerto serie RS-232C.
	IRQ5	No usada.
Menor	IRQ6	Estado del diskette DRV.
	IRQ7	Puerto paralelo PRT (no usado en BIOS).

Debe hacerse notar que sólo las interrupciones IRQ2 a IRQ7 están disponibles en la ranura expansión (SLOT).

Las direcciones de los vectores que serán cargados en los registros IP y CS del μP para llamar a la rutina de atención a la interrupción son las siguientes :

BYTE		TIPO DE INTERRUPCION
		IRQ7 15
		IRQ6 14
		IRQ5 13
		IRQ4 12
		IRQ3 11
		IRQ2 10
		IRQ1 9
		IRQ0 8
		RESERVADA 7
		RESERVADA 6
		IMPRIMIR PANTALLA 5
		INTO 4
		INT 3
0000CH		NMI 2
00008H		PASO SENCILLO 1
00004H		ERROR DE DIV. 0
00000H		

Donde los vectores son los siguientes:



Para usar las interrupciones IRQ2, IRQ3 y IRQ5, se debe colocar el vector de la rutina de atención en la localidad de memoria correspondiente. Esto se puede hacer desde lenguaje ensamblador por medio de la función de 25H MSDOS (ver apéndice de funciones de MSDOS).

Una vez que se inicializó el 8259, se colocó el vector que apunta a la rutina de atención a la interrupción y se cargó en memoria la rutina de atención misma, el SAD (Sistema de Adquisición de Datos) puede comenzar a enviar y recibir información a la PC por medio del

SLOT.

Las acciones básicas que se deben realizar son:

a) SAD envía una señal de interrupción al 8088. Esta señal llega a alguna de las terminales de IRQ del SLOT. En el SLOT sólo se encuentran disponibles las señales IRQ5, IRQ3 y IRQ2 para propósitos del usuario pues las demás IRQ's son requeridas por el sistema para realizar otras funciones. Como ya se dijo la prioridad que el sistema BIOS asigna a estas señales es descendente de IRQ2 a IRQ5. La señal de interrupción enviada por el SAD se debe mantener en estado lógico alto hasta recibir una señal de reconocimiento de parte del 8088.

b) El 8088 carga IP y CS con la dirección de la rutina de atención a la interrupción (los procesos inherentes a la comunicación entre el microprocesador 8088 y el controlador de bus 8259 son transparentes al usuario).

c) La rutina de atención envía un reconocimiento al SAD por un puerto para pasar a un estado lógico bajo la señal de INT del SAD. Para realizar la escritura en el puerto se utilizan las señales del SLOT de expansión IOW, A0-A7 y se pueden utilizar también las terminales Do-D7. Estas señales entran a una lógica de decodificación y pasan al ACK del SAD.

d) El 8088 debe resguardar los registros que requiera utilizar nuevamente al regresar de la rutina de atención de interrupción a la tarea que se encontraba realizando anteriormente.

e) Se utiliza la instrucción IN para leer la información de un puerto, el cual está conectado al bus de datos del SAD. La señal IOR del SLOT se conecta a una señal de WAIT del SAD para lograr una relación de tiempos adecuada. (Suponemos que la velocidad a la que trabaja el Procesador Digital de Señales (DSP Digital Signal Processor) que es el componente fundamental de nuestro SAD es mayor que la del 8088, por que los DSPs comerciales funcionan a aprox. 20 MHz mientras que en una PC el 8088 lo hace convencionalmente a 4.77 MHz).

La interconexión entre ambos sistemas se puede esbozar como sigue:

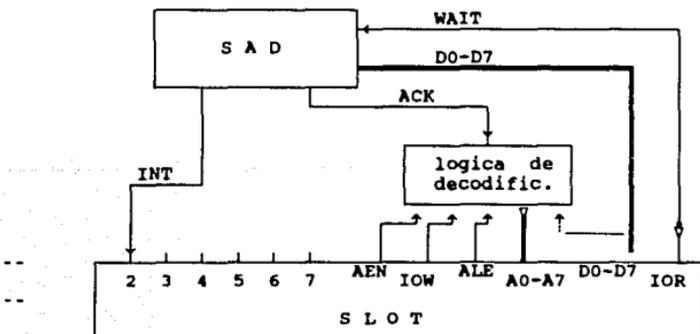


Fig. IV.4 Interfaz por interrupción.

El esquema de interconexión por medio de interrupciones presenta algunas desventajas como son:

- a.- El tiempo que toma al 8088 recibir la interrupción, tomar el nivel de voltaje y apuntador del 8259, respaldar los registros en el STACK¹ y saltar a la rutina de atención es de 61 ciclos de reloj (con reloj de 4.77 MHz esto es 12.81 microsegundos).
- b.- Las interrupciones se atienden hasta que el 8088 termina de realizar la instrucción que se encuentra² ejecutando, esto tarda normalmente entre 1 y 5 microsegundos². Sin embargo hay operaciones que pueden tomar mas tiempo como multiplicaciones y divisiones. Las instrucciones del tipo repite, asegura y especificación prefija de segmento (*repeat*, *lock* y *segment prefix override*), son consideradas como parte de la instrucción a que se anteponen y no permiten interrupción entre ellas y su instrucción, pues para concluir el procedimiento completo de la primera instrucción se debe terminar con la segunda.
- c.- Si la interrupción que se usa para la interfaz es de menor prioridad que otra del sistema, se debe esperar a que concluya la de mayor jerarquía, lo que aumenta el tiempo de retardo de atención a la interrupción. Una solución a este problema es enmascarar los otros niveles de interrupción (se debe tener cuidado de no mascarar una función que el BIOS usa para operar normalmente).

La ventaja de este esquema es que no se pierde tiempo en hacer verificaciones del estado de cada uno de los dispositivos (poleo), además permite el uso de rutinas de control para dar una atención mas adecuada al SAD.

IV.4.2 Interfaz por registro de entrada salida.

Este método consiste en realizar lecturas y escrituras sucesivas de la PC a registros existentes en la tarjeta de desarrollo para realizar la interfaz. Estos registros son conocidos con el nombre de registros de propósito general de entrada (DI) y de registros de propósito general de salida (DO).

Los métodos básicos de uso de estos registros son como:

- a) Dispositivos de mapeo de memoria.
- b) Dispositivos de mapeo de entrada/salida (I/O).

1

El problema de reducir el tiempo de respuesta a una interrupción cuando se deben respaldar una gran cantidad de registros, se puede resolver en ocasiones salvando los registros mas críticos al principio de la rutina de interrupción y los demás en momentos sucesivos de la rutina.

2

2 Todos los datos de tiempo se dan para una PC con reloj de 4.77 Hz.

El uso de estos dos métodos es muy similar, el primero consiste en considerar los registros como una localidad de memoria y accederlos con instrucciones del microprocesador destinadas al manejo de memoria mientras que en el mapeo de entrada salida considera los registros DI o DO como parte del mapa de puertos y son accedidos por medio de las instrucciones IN o OUT.

Algunas ventajas del mapeo de I/O son:

- a) No requiere registro de segmento para direccionar.
- b) Se requieren menos bits para de codificar una dirección.
- c) Las operaciones de DMA entre puertos y memoria se realizan en un solo ciclo de bus.

Sus desventajas:

- a) Sólo se pueden usar las instrucciones IN y OUT para leer y escribir.

Algunas ventajas del mapeo de memoria son:

- a) Gran espacio de direccionamiento.
- b) Se puede usar cualquier instrucción que haga referencia a memoria.

Las desventajas:

- a) Necesidad del uso de segmentos.
- b) Necesidad de decodificar mayor número de bits.

En nuestro proyecto el interés se centra en el uso de mapeo de I/O, debido a las ventajas que este presenta con respecto al mapeo de memoria.

Para mapeo de I/O se usan 9 bits, para decodificar los puertos I/O; esto da capacidad para direccionar 512 puertos, para lo que se usan los bits de dirección A0 a A8. También se usa A9 como décimo bit para indicar si el bus recibe o es fuente para las instrucciones INT o OUT. Para hacer la decodificación y habilitar una tarjeta, se debe utilizar también la señal AEN (Address ENable) proveniente de controlador de DMA, esto permite que el DMA no escriba o lea de los puertos.

Para realizar la interfaz de la PC con otra tarjeta se requiere circuitería adicional que se encargue de habilitar los chips de tal tarjeta cuando se escribe a cierta dirección del mapa de puertos asignada para la interfaz tarjeta-PC.

La lógica adicional para este propósito puede estar constituida por circuitos lógico combinacionales, decodificadores y/o comparadores.

Una configuración propuesta para este fin es la que sigue:

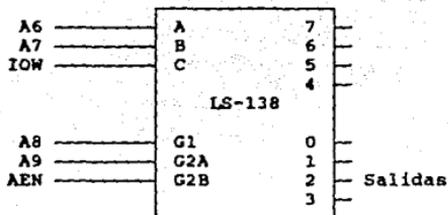


Fig. IV.5 Diseño de lógica de escritura de puertos I/O

V.- ALGORITMOS PARA PROCESAMIENTO DE SEÑALES.

V.1.- Serie Discreta de Fourier (DFS) y Transformada Discreta de Fourier (DFT).

V.1.1.- Introducción.

La transformada discreta de Fourier (DFT) es un procedimiento por el cual se obtiene la transformada de Fourier de una señal que ha sido discretizada.

La definición de la DFT proviene de la representación de una función periódica discreta en series de Fourier. El desarrollo de las series discretas de Fourier (DFS) y de la transformada discreta de Fourier (DFT) se expone a continuación.

V.1.2.- Series Discretas de Fourier (DFS).

Sea $x[n]$ una función periódica con período N de manera que $x[n]=x[n + kN]$ para valores enteros de k . Es posible representar esta función como suma de senos y cosenos o exponenciales cuyas frecuencias sean múltiplos de la frecuencia fundamental $2\pi/N$.

En contraste con las funciones continuas hay sólo N distintas exponenciales complejas cuyo período es un entero submúltiplo del período fundamental N . Esto es consecuencia de que la exponencial:

$$e_{x[n]} = e^{j(2\pi/N)nk}$$

es periódica en k , con un período de N . Por que $e_{n}[n] = e_1[n]$, $e_{n+1}[n]= e_1[n]$, etc para $k=0,1,2,\dots,N-1$. Por lo tanto la serie se puede representar con N exponenciales complejos de la forma:

$$x[n] = 1/N \sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)nk}$$

Para obtener x se usa el hecho de que:

$$1/N \sum e^{j(2\pi/N)nr} = \begin{cases} 1 & \text{para } r=mN; \text{ Si } m \text{ es entero} \\ 0 & \text{para otra } r \end{cases}$$

con lo que se puede llegar a que:¹

1

Los detalles matemáticos se pueden consultar en la bibliografía.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)nk}$$

por comodidad en la notación hacemos:

$$w_N = e^{-j(2\pi/N)}$$

entonces las fórmulas anteriores se pueden expresar como:

$$X[k] = \sum_{n=0}^{N-1} x[n] w_N^{kn}$$

$$x[n] = 1/N \sum_{k=0}^{N-1} X[k] w_N^{-kn}$$

en que $X[k]$ y $x[n]$ son periódicas.

V.1.3.- Transformada Discreta de Fourier (DFT).

La DFT nos permite la representación de secuencias de duración finita en el tiempo por medio de las Series Discretas de Fourier.

Considerando una función $x(n)$ de longitud N tal que $x(n)=0$ para $0 > n > N-1$, tenemos que su función periódica correspondiente es:

$$x[n] = \sum_{r=-\infty}^{\infty} x[n + rN]$$

lo cual se puede expresar como:

$$x[n] = x\{[n]\}_N$$

Para mantener la dualidad entre los dominios del tiempo y la frecuencia, se asocian a la secuencia de duración finita los coeficientes de Fourier que corresponden a un periodo $X[k]$.

Por lo tanto se cumplen las relaciones:

$$X[k] = X\{[k]\}_N$$

$$x[n] = x\{[k]\}_N R_N[k]$$

donde:

$$R_N[k] = \begin{cases} 1 & 0 < n \leq N-1 \\ 0 & \text{en otro caso} \end{cases}$$

y la DFT se define como:

$$X[k] = \begin{cases} \sum_{n=0}^{N-1} x[n] w_N^{kn} & 0 \leq k \leq N-1 \\ 0 & \text{en otro caso} \end{cases}$$

y se cumple que:

$$x[n] = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} X[k] w_N^{-kn} & 0 \leq k \leq N-1 \\ 0 & \text{en otro caso} \end{cases}$$

V.1.4 Transformada Rápida de Fourier (FFT).

La Transformada Rápida de Fourier (FFT) es un procedimiento por el cual se calcula la DFT. La FFT usa una serie de simplificaciones para el cálculo de la sumatoria:

$$X[k] = \sum_{n=0}^{N-1} x[n] w_N^{kn}$$

En donde los términos $x[n]$ son números complejos al igual que w_N .

Existen dos métodos básicos de reducción de operaciones (y en consecuencia de reducción de tiempo de proceso) para el cálculo de la DFT a partir de la sumatoria anterior, estos métodos se conocen como descomposición en tiempo (*decimation in time*) y descomposición en frecuencia (*decimation in frequency*).

Estos procedimientos arrojan iguales resultados en la reducción de cálculos, y son, en esencia, muy similares, por lo que se explicará solo la descomposición en tiempo. (Los detalles de descomposición en frecuencia pueden ser consultados en la bibliografía).

V.1.4.1.- Descomposición en tiempo.

Este método usa la descomposición de la secuencia $x[n]$ en secuencias menores, que se utilizan para el cálculo de la DFT. Esta descomposición en subsecuencias se basa en las propiedades de simetría y periodicidad de la exponencial:

$$w_N^{kn} = e^{-j(2\pi/N)kn}$$

En aplicaciones prácticas (y en particular para nuestra aplicación) y con el fin de ilustrar mejor los algoritmos de descomposición en tiempo se usan un número de muestras N , tomadas a una señal de manera que N sea potencia de 2, es decir se verifica que $N=2^m$ donde m es entero.

Con el fin de lograr una mejor comprensión del proceso empleado para el cálculo de la DFT, se ejemplifica en lo sucesivo para $N=8$.

Hechas estas consideraciones se parte, para el cálculo de la DFT, de la fórmula :

$$X[k] = \sum_{n=0}^{N-1} x[n] w_N^{nk}$$

y se separa en términos pares e impares de la siguiente manera:

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] w_N^{2rk} + \sum_{r=0}^{(N/2)-1} x[2r+1] w_N^{(2r+1)k}$$

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] \{w_N^2\}^{rk} + w_N^k \sum_{r=0}^{(N/2)-1} x[2r+1] \{w_N^2\}^{rk}$$

usando el hecho de que:

$$w_N^2 = w_{N/2}$$

se llega a que:

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] w_{N/2}^{rk} + w_N^k \sum_{r=0}^{(N/2)-1} x[2r+1] w_{N/2}^{rk}$$

$$X[k] = G[k] + w_N^k H[k]$$

donde:

$$G[k] = \sum_{r=0}^{(N/2)-1} x[2r] w_{N/2}^{rk}$$

$$H[k] = \sum_{r=0}^{(N/2)-1} x[2r+1] w_{N/2}^{rk}$$

Esto propone ya el cálculo de la DFT original a partir de subsecuencias con términos pares e impares como cálculo de DFTs de $N/2$ muestras. Lo anterior se ilustra en la figura V.1, conocido como diagrama de mariposa:

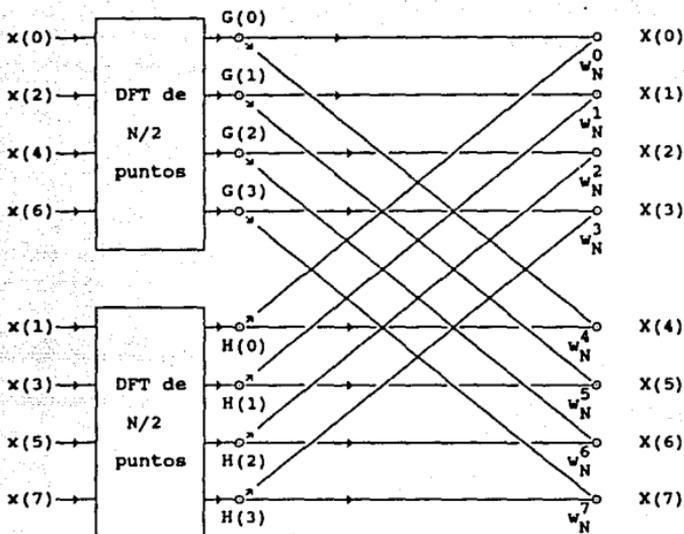


Figura V.1

En esta figura se indican los términos $G(k)$ y $H(k)$, anteriormente descritos. Los círculos representan lugares de suma de las cantidades indicadas por las rutas y las constantes w indican una multiplicación previa que se debe realizar a las cantidades antes de ser sumadas.

El siguiente paso consiste en la descomposición de las cantidades $G[k]$ y $H[k]$ de la siguiente forma:

$$G[k] = \sum_{r=0}^{(N/2)-1} g[r] w_N^{rk}$$

$$G[k] = \sum_{l=0}^{(N/4)-1} g[2l] w_N^{2lk} + \sum_{l=0}^{(N/4)-1} g[2l+1] w_N^{(2l+1)k}$$

$$G[k] = \sum_{l=0}^{(N/4)-1} g[2l] w_N^{lk} + w_N^k \sum_{l=0}^{(N/4)-1} g[2l+1] w_N^{lk}$$

similarmente:

$$H[k] = \sum_{l=0}^{(N/4)-1} h[2l] w_N^{lk} + w_N^k \sum_{l=0}^{(N/4)-1} h[2l+1] w_N^{lk}$$

Esta vez el cálculo de la DFT se descompone en subsecuencias de cálculo de DFT de $N/4$.

v.2. Para 8 puntos el diagrama de cómputo se muestra en la figura

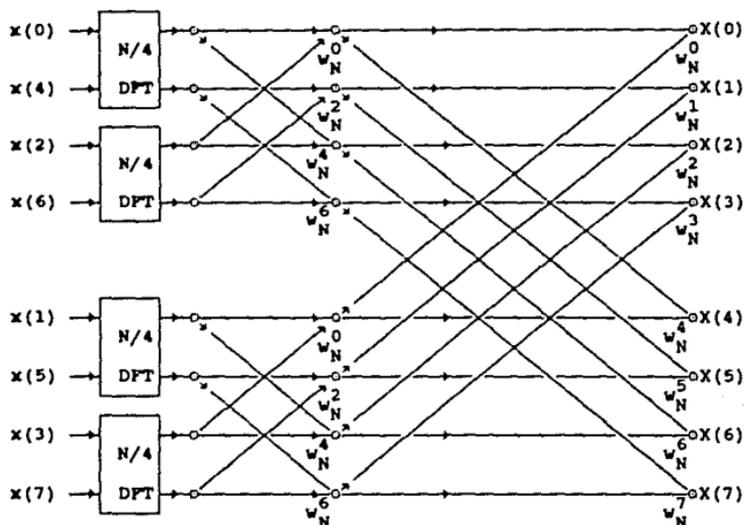


Figura V.2

Para casos más generales se continua la descomposición en transformadas realizadas sobre N/8, N/16, ...muestras del total N, hasta lograr que el proceso de transformación se realice con únicamente con dos muestras. Se puede realizar el cálculo de estas transformadas usando las expresiones:

$$X_{m+1}[p] = X_m[p] + w_N^r X_m[q]$$

$$X_{m+1}[q] = X_m[p] + w_N^{r+N/2} X_m[q]$$

Donde dichas expresiones se representan comunmente por medio de diagramas como el de la figura V.3.

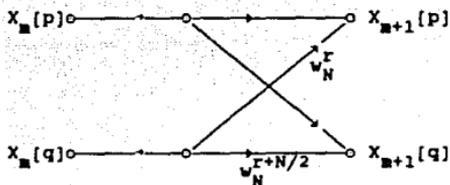


Figura V.3

Utilizando este diagrama, la computación de la FFT con $N=8$ se verifica de la siguiente forma (figura V.4):

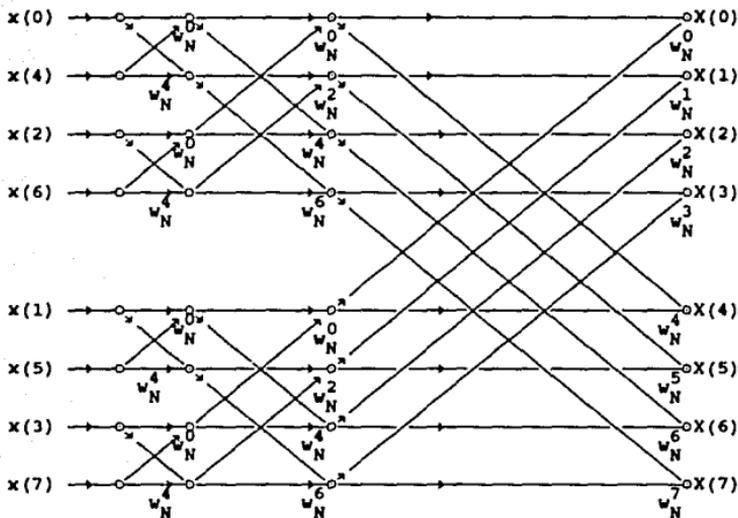


Figura V.4

Aquí son posibles otras simplificaciones aprovechando el hecho de que:

$$w_N^{N/2} = e^{-j\pi} = -1$$

entonces:

$$X_{m+1}(p) = X_m(p) + w_N^r X_m(q)$$

$$X_{m+1}(q) = X_m(p) - w_N^r X_m(q)$$

El diagrama de mariposa para dos puntos queda (figura V.5):

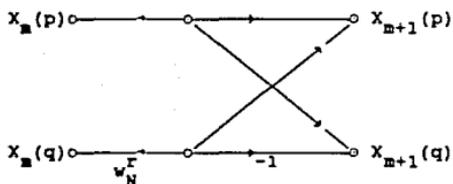


Figura V.5

y finalmente la FFT para $N=8$ queda (figura V.6):

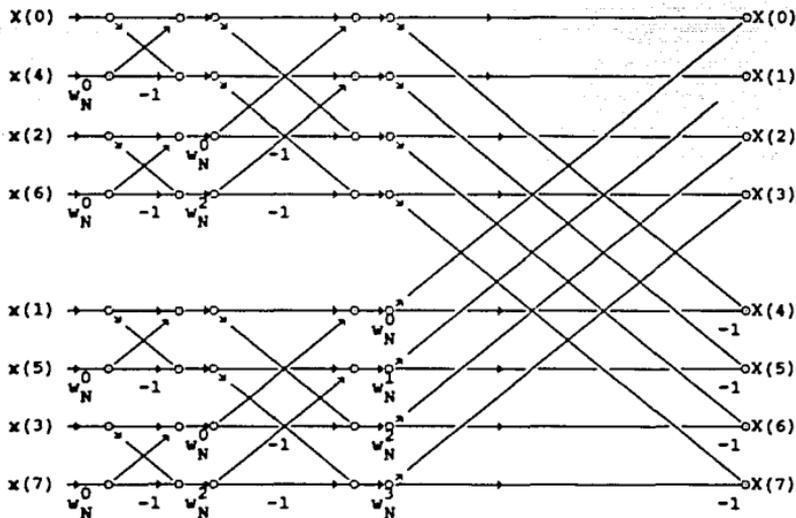


Figura V.6

CAPITULO VI DESCRIPCION DEL SISTEMA.

VI.1 Generalidades y requerimientos de equipo.

El sistema emulador de osciloscopio y analizador de espectro (OSANEM) permite desplegar la forma de onda de una señal eléctrica y su espectro de frecuencia en la pantalla de una computadora PC.

El hardware del sistema se construyó sobre una tarjeta insertable en los SLOTS de una computadora PC XT. Los circuitos que lo componen se colocaron sobre bases y se interconectaron con wire wrap.

Para hacer funcionar el sistema se requiere, además de la tarjeta del sistema y del software correspondiente, el siguiente equipo:

- Computadora IBM-PC, IBM-PC XT o compatible con:
 - Sistema operativo MS-DOS versión 2.1 o mayor.
 - Monitor tipo Color Graphics Adapter (CGA) o equivalente, o en su defecto utilizar un programa emulador (ej. SIMCGA.COM).
 - Impresora que soporte el modo gráfico proporcionado por el programa GRAPHICS.COM del sistema operativo MS-DOS.

VI.2. Hardware del sistema.

VI.2.1 Funciones.

El sistema de adquisición de datos realiza las siguientes funciones:

- a) Adquisición y digitalización de señales eléctricas.
- b) Almacenamiento de la información en memoria.
- c) Cálculo de la transformada rápida de Fourier (FFT) de 256 puntos de la señal.
- d) Despliegue en pantalla de la señal eléctrica y de su espectro de frecuencia.
- e) Almacenamiento de la información en medios magnéticos.
- f) Impresión de gráfica de la señal y espectro de frecuencia.

Este está diseñado para muestrear señales con las siguientes especificaciones:

Vmax = 0 V
Vmin = +5 V
fmax de la señal muestreada = 7 kHz

VI.2.2 Arquitectura.

La arquitectura del sistema se presenta en la figura VI.1

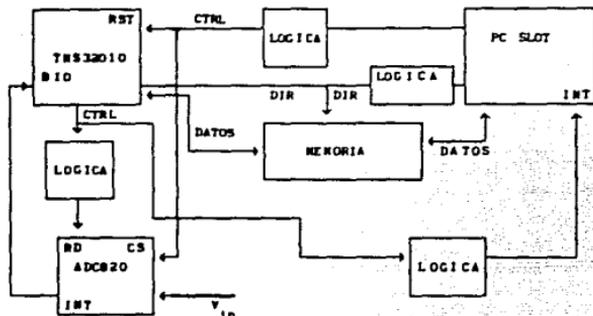


Fig. VI.1

En el diagrama anterior se observan los siguientes subsistemas:

- TMS32010. Procesador digital de señales (DSP).
- Memoria del sistema.
- PC (se comunica por medio del SLOT).
- Convertidor analógico digital (ADC820).
- Lógica de decodificación.

VI.2.2.1 Descripción de los subsistemas.

a) TMS32010.

Procesador digital de señales (DSP) que realiza la adquisición de datos a través del convertidor analógico digital, hace la FFT, de manera que la PC reciba la información ya procesada.

b) Memorias del sistema.

Las memorias empleadas son las 2147 de INTEL, éstas son de 4096 X 1 bit, estáticas y de alta velocidad, debido a que el DSP requiere tiempos de acceso menores a 70 ns para guardar y acceder la información.

Debido a que el bus de datos del DSP es de 16 bits, se requieren 16 circuitos de este tipo para componer el banco de memoria.

El bus de datos del DSP se conecta al banco de memoria a través de circuitos 74LS245, transreceptores que se ponen en alta impedancia cuando las memorias se accesan por la PC.

La PC accesa las memorias de 8 en 8 bits (en contraste con el DSP que lo hace de 16 en 16 bits). Para lograr esto, se incluyeron las

memorias 2147 dentro del mapa de memoria de la PC, las localidades usadas para este propósito fueron de EC000 a EFFFF que representan un espacio de 16 K de memoria. Dentro de este espacio se usan sólo 8 K (se ignore el bit 12 en la lógica de decodificación), 4 K para los 8 bits bajos de la palabra de 16 bits y 4 k para los bits altos. La forma como se mapea la memoria en el DSP y en la PC se ilustra a continuación:

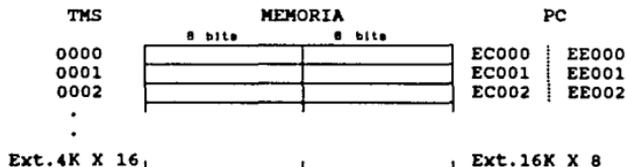


Fig. VI.2

* Las direcciones de la PC se establecen por segmento:dirección (i.e. EC00:0000)

c) PC Slot.

En él se encuentran las señales ya descritas en la sección III.5, usadas para la interfaz con el resto del sistema.

d) Convertidor analógico digital (ADC820).

Es un convertidor analógico digital de 8 bits. Sus especificaciones más importantes son:

- P max: 75 mW
- Resolución: 8 bits
- Tiempo de conversión máximo: 2.5 μ s (en modo RD)
- Voltaje de alimentación: 5 V
- Salida Tri-state
- Rango de voltaje de entrada: 0-5 V (con alimentación de 5

Este convertidor puede trabajar en 2 modalidades, RD y WR-RD. En el sistema se usa el modo RD, por las ventajas que presenta para la interfaz. Los detalles de tiempos y especificaciones del ADC820 se presentan en las hojas de datos.

La forma en como se realiza la interfaz TMS32010-ADC820 es la siguiente:

- 1) La PC activa el TMS32010 y el ADC por medio de un voltaje bajo en las terminales RST y CS respectivamente.
- 2) El TMS32010 activa las señal \bar{RD} para que se inicie la conversión.

3) El TMS32010 espera a que la señal \overline{INT} se haga baja por medio de la verificación en software de la terminal \overline{BIO} .

4) Al hacerse bajo el voltaje \overline{INT} , el TMS32010 realiza la lectura del voltaje codificado y desactiva la señal \overline{RD} del ADC820.

Sólo los 8 bits bajos del bus de datos del TMS32010 se encuentran conectados al ADC820, los 8 bits altos leen información irrelevante.

e) Lógica de decodificación.

Esta lógica se encarga de adecuar las señales del slot o del TMS32010, de manera que se logre un adecuado acoplamiento entre ellos y con los otros subsistemas. Está constituida a su vez por varios subsistemas.

- Lógica para control del la PC sobre el DSP.
- Lógica para control del DSP sobre el ADC.
- Lógica de interrupción del DSP a la PC.
- Lógica de decodificación de la PC a la memoria.

VI.2.3 Requerimientos de potencia.

A continuación se hace una descripción de la potencia máxima requerida por los elementos del sistema.

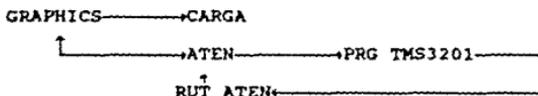
Circuito empleado	Cantidad	Icc(mA)	Pot(W) total
74LS04	1	4.2	.021
74LS08	2	4.0	.040
74LS11	1	3.0	.015
74LS32	3	4.9	.074
74LS74	2	4.0	.040
74LS138	1	10.0	.050
74LS155	1	9.0	.045
74LS244	4	27.0	.540
74LS245	4	58.0	1.160
TMS32010	1	275.0	1.375
2147	16	140.0	.700
ADC820	1		.075
Total	37		4.135

* Falta agregar los circuitos de la parte de entrada (de interfase).

VI.3.- Software.

El software del sistema se puede dividir en 3 partes: programas de interacción con el usuario, programas de control de la tarjeta, y algoritmos de procesamiento del TMS32010. Adicionalmente se usaron algunos programas de apoyo para el desarrollo.

VI.3.1 Arbol de llamada de programas



VI.3.2 Programa de interacción con el usuario (GRAPHICS).

El sistema GRAPHICS, realiza la parte de interacción con el usuario y la graficación de las señales eléctricas. Fué realizado mediante el compilador QUICK BASIC y ligado con rutinas de ensamblador 8086 :

GRAPHICS.BAS + CARGA.ASM + IR2PROC.ASM = GRAPHICS.EXE

En las siguientes secciones se describe a nivel técnico las llamadas que el sistema realiza.

Para obtener una referencia del funcionamiento a nivel usuario se puede acudir al Apéndice B de este documento.

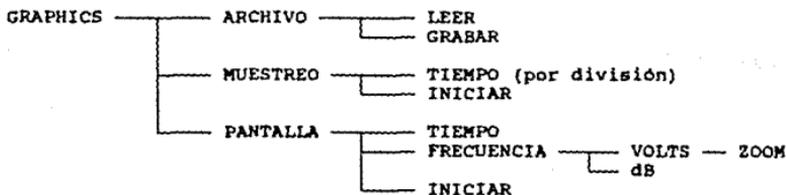
VI.3.2.1. Invocando al Programa

El sistema GRAPHICS permite al usuario utilizar la información muestreada a partir de la Tarjeta de Adquisición de Datos. Para correr el programa basta con escribir, estando a nivel de sistema operativo :

A> GRAPHICS

o sea que sólo se necesita de el archivo GRAPHICS.EXE para poder trabajar.

VI.3.2.2. Procesos.



VI.3.2.2.1 Archivo.

La función de este módulo es la de permitir la manipulación de datos mediante archivos de la forma :

abscisa	ordenada	
xt0	yt0	Tiempo
xt1	yt1	
xt255	yt255	Frecuencia
xf0	yf0	
xf1	yf1	
xf255	yf255	

Estos archivos están escritos en ASCII por lo que pueden ser generados o editados por el usuario.

VI.3.2.2.1.1 Leer.

Como su nombre lo indica, esta rutina permite leer un archivo de disco, existe una rutina auxiliar para manejo de errores que considera los siguientes casos :

- a) Nombre incorrecto del archivo - Cuando no se cumple con las restricciones impuestas por MS-DOS.
- b) Archivo no encontrado
- c) El disco está lleno
- d) La unidad de disco no está lista
- e) Existe error en el disco
- f) Otro error = Código

VI.3.2.2.1.2 Escribir.

Una vez que se ha muestreado o leído información de un archivo, esta puede ser grabada en disco. Para el nombre del archivo no hay una extensión por omisión, por lo que el usuario deberá escribirla si es que la desea.

VI.3.2.2.1.3 Terminar.

Esta opción nos permite abandonar el programa y regresar a nivel de sistema operativo.

VI.3.2.2.2. Muestreo.

Este módulo permite establecer los parámetros bajo los cuales se

hará el muestreo.

VI.3.2.2.2.1 Tiempo por división.

Esta rutina pide al usuario elegir entre 8 escalas posibles de tiempo. Este dato sirve como parámetro para que el TMS 32010 conozca la frecuencia de muestreo :

$$f = 1/t$$

VI.3.2.2.2.2 Iniciar

Cuando se llega a este punto el sistema pregunta por un Voltaje de Disparo, que es el voltaje que al ser alcanzado permite que se inicie el muestreo de la señal. En caso de que este voltaje no se registre después de un cierto número de lecturas, el sistema lo indicará solicitando una vez más el parámetro.

El muestreo se hace a través de una rutina en ensamblador que transfiere el control al TMS 32010, el cual controla al convertidor A/D. Una vez terminado el muestreo el control regresa a la PC y el sistema de graficación puede leer la información generada, la cual ha sido grabada en los circuitos de memoria de la tarjeta.

En primer lugar se leen los datos correspondientes a la señal en el tiempo. A continuación se leen los correspondientes a la frecuencia. En este caso se leen las partes real e imaginaria de la componente, considerando que la magnitud está dada por :

$$\text{magnitud} = \sqrt{(\text{parte imaginaria})^2 + (\text{parte real})^2}$$

Estos datos, después de ser leídos, son escalados considerando las siguientes condiciones :

Voltaje : $V_{in \text{ min}} = 0$ [V] ■ 0 H
 $V_{in \text{ max}} = + 5$ [V] ■ 255 H

Frecuencia :

Los valores se guardan en vectores : XT, YT, XF y YF para cada uno de los cuales se determinan los valores máximo, mínimo e incremento.

VI.3.2.2.3 Pantalla.

Este módulo tiene como objetivo presentar en la pantalla o en la impresora la gráfica correspondiente a la información muestreada o la leída de un archivo.

Cuando se dibuja en la pantalla el procedimiento común tanto para tiempo como para frecuencia es :

- a) Trazo de Ejes (X y Y)
- b) Calcula Escalamiento

- c) Calcula Desplazamiento
- d) Grafica

VI.3.2.2.3.1 Tiempo.

Se dibuja la función en el dominio del tiempo.

VI.3.2.2.3.2 Frecuencia.

Se dibuja la función en el dominio de la frecuencia preguntando al usuario si desea esta gráfica en decibeles (dB) o en Volts, existiendo la opción para este último caso de presentar una ampliación de un determinado rango de frecuencias (zoom).

VI.3.2.2.3.3 Imprimir.

En este caso se manda a imprimir una imagen de la gráfica presentada en la pantalla (Impresión de pantalla), por lo que el procedimiento es común a las gráficas mencionadas en los puntos 2.3.1 y 2.3.2.

VI.3.3 Programas de control de la tarjeta.

El programa GRAPHICS llama a dos programas en ensamblador:

- a) CARGA. Realiza la carga del programa del TMS32010 en las memorias 2147 del sistema. (Ver apéndice C. Programas).
- b) ATIEN. Carga la rutina RUT_ATEN para atender la interrupción 2 del slot (Ver sección III.4.3 y Apéndice D: vectores de interrupción). Transfiere el control al TMS32010 y espera a que se realice la int.2, desactiva el TMS.

VI.3.4 Algoritmos procesamiento para el TMS32010.

El programa de procesamiento es leído y ejecutado por el TMS32010. Las funciones que realiza se explican a través de los módulos que lo componen:

- a) Lectura de datos a través del ADC820. Se inicia el almacenamiento de los datos válidos cuando se alcanza el nivel de voltaje especificado en la localidad 21H de la memoria del TMS32010; envía un mensaje de error si este voltaje nunca se alcanza.
- b) Copia de los datos anteriormente adquiridos a otra sección de memoria, para calcular la transformada rápida de Fourier (FFT).
- c) Cálculo de la transformada rápida de Fourier (FFT, ver sección V.1.4).
- d) Sección de datos de SENOS y COSENOS para la FFT.

Este programa deja la información del muestreo de voltajes y la FFT en las memorias para ser leídas por el programa de graficación.

VI.3.5 Programas de apoyo para el desarrollo

Durante el desarrollo del sistema se usaron varios programas de apoyo:

- a) Programas en ensamblador. Para visualizar el contenido de las memorias 2147 del sistema y para cargarlas con datos secuenciales
- b) Turbopascal. Desarrollo de programas para pasar la información binaria de las memorias a ASCII. Colocar el tiempo y nivel de disparo en la localidad correspondiente.
- c) Lotus. Para graficar las formas de onda y las espigas de la FFT a fin de comprobar los resultados obtenidos a través de GRAPHICS.
- d) Programas de prueba del TMS32010 y del ADC820. Programas para verificar la lectura de niveles de voltaje.

VI.4 .- Implementación y desarrollo.

Los pasos de desarrollo del sistema se pueden resumir en las siguientes etapas:

- a) Instalación de memorias 2147.

Esta etapa se llevó a cabo alambando con wire rap las memorias y su lógica de decodificación en la tarjeta que se conecta al SLOT. A continuación se desarrollaron programas en ensamblador para cargar y visualizar la información y verificar que este módulo funcionaba adecuadamente.

- b) Instalación del TMS32010.

Se colocó el TMS en la tarjeta. Se cargaron programas simples de prueba para verificar que el TMS accedía a las memorias y realizaba operaciones sobre sus localidades.

- c) Instalación del ADC820.

Finalmente se alambó el convertidor analógico digital y la etapa aisladora de entrada al ADC. Se elaboró un programa para probar el ADC a través del control del TMS, haciendo lecturas de voltajes en forma digital.

El Apéndice E presenta un diagrama del sistema implementado y las hojas de datos de los principales componentes.

El Apéndice F muestra una lista de los componentes y los costos.

- d) Desarrollo de software

En forma paralela se desarrolló el programa para el TMS de lectura de voltajes y cálculo de la FFT. Este se probó en el simulador del TMS32010 de Texas Ins. para PC. Una vez correcto se cargó en las memorias y se ejecutó.

Se desarrolló el programa de graficación de las formas de onda y del espectro de frecuencia en Quick basic.

e) Interfaz del software.

Se ligaron los programas en ensamblador con el programa de graficación con el fin de integrar el sistema en un sólo programa.

VII.- PRUEBAS DE CAMPO.

VII.1.- Introducción.

Este capítulo es el resultado de una serie de pruebas y ajustes que se hicieron al sistema con objeto de verificar sus características.

Las pruebas que se efectuaron son de dos tipos:

- a) en el dominio del tiempo.
- b) en el dominio de la frecuencia.

Comenzaremos con las pruebas en el dominio del tiempo, describiendo el proceso que se usó para lograr el ajuste de la escala y los errores típicos que se encontraron al muestrear las señales.

Posteriormente se analizan los errores en el espectro de frecuencia a través de la reconstrucción de la señal en tiempo.

Para finalizar se detallan las características del instrumento y se dan ejemplos de muestreos de señales típicas realizados con el sistema.

VII.2.- Dominio del tiempo.

En este dominio los problemas de lograr precisión en el muestreo de señales se presentan tanto en el eje horizontal (tiempo), como en el eje vertical (voltaje).

VII.2.1 Eje X: tiempo.

En el eje horizontal, el problema fundamental consistió en lograr que el convertidor analógico digital (ADC), controlado por el microprocesador (DSP) realice el muestreo de la señal cada determinado período de tiempo. Para lograrlo el microprocesador hace un conteo antes de habilitar el ADC para que tome una muestra, de esta manera por ejemplo, al usar la escala de 250 microsegundos/división, el programa GRAPHICS pasa a la rutina ATEN un parámetro de tiempo de espera entre muestreos menor que si se realiza con una escala de 10 milisegundos/división.

La implementación del contador se realiza con un loop dentro del programa de control del DSP. El número de ciclos que se debe realizar este loop depende directamente de la escala que se use. Si se grafican en total 256 pixels en el tiempo cada división tiene 25.6 pixels, por lo que el tiempo entre muestras se determina como $(\text{tiempo/división})/25.6$.

Los tiempos entre muestras se componen de dos partes:

- a) Tiempo de muestreo de ADC: Este es un tiempo fijo en el cual se activa, se muestrea y se desactiva el ADC, siempre se presenta y es debido en parte al tiempo de conversión del ADC.

Para determinar este tiempo se muestrea una señal con cierto periodo conocido y se observa cuántas muestras se tomaron de la misma.

Para el caso particular del sistema diseñado, se muestreó una señal triangular con periodo de 100 ms y se observó que se lograban entre 17 y 18 muestras cada vez.

Al realizar una serie de muestreos se llegó a que el tiempo de ADC que usaba el sistema era de 5.66 μ s con una desviación estándar de 0.15 μ s.

- b) Tiempo debido al loop de espera: Este tiempo está compuesto de dos partes: 10 ciclos de reloj (fijos) que el μ P usa para saltar a la rutina de espera y regresar de esta; y el tiempo de espera dentro de la rutina, consistente en 4 ciclos de reloj cada vez que se ejecuta el loop.

Los tiempos de espera (número de ciclos de loop) se determinaron con la tabla VII.1

Esc	t/div [μ s]	t/pixel [μ s]	t de loop [μ s]	ciclos de 200 ns	ci.var. de loop	Num.de ciclos teórico	Num.de ciclos real	% de dif.
2	250	9.77	4.11	20.55	10.55	2.64	4	51.69%
3	500	19.53	13.88	69.38	59.38	14.84	15	1.05%
4	1000	39.06	33.41	167.03	157.03	39.26	39	-0.66%
5	2500	97.66	92.00	460.00	450.00	112.50	112	-0.44%
6	5000	195.31	189.66	948.28	938.28	234.57	229	-2.37%
7	10000	390.63	384.97	1924.84	1914.84	478.71	474	-0.98%
8	50000	1953.13	1947.47	9737.34	9727.34	2431.84	2414	-0.73%

Tabla VII.1

Los valores de esta tabla se obtuvieron calculando los valores teóricos de ciclos de espera entre muestras de la señal para una escala determinada (Número de ciclos teóricos). Dichos valores se ajustaron experimentalmente al muestrear señales con un periodo conocido, visualizadas en un osciloscopio analógico (Número de ciclos real) para lograr la mayor fidelidad posible en sucesivas mediciones. El procedimiento anterior es análogo al de calibrar un osciloscopio.

La explicación de cada columna es la siguiente:

ESC es el número de escala con que el programa identifica el tiempo por división.

T/DIV es la escala de tiempo por división en μ s/div.

T/PIX es la escala de (t/div)/(25.6 pixels/div) en μ s/pixel.

T DE es el t/pix menos el tiempo que usa el ADC para muestrear LOOP (como ya se dijo este es de 5.66 μ s)

CICLOS es el número de ciclos de reloj de microprocesador que se debe esperar para cubrir el tiempo de loop (el tiempo de reloj del microprocesador con un cristal de 20 MHz es de 200 ns).

CI. ciclos variables de loop, resulta de restar a los ciclos **VAR.** de reloj de microprocesador, 10 ciclos de reloj fijos (ver **DE** inciso b) pág. anterior).
LOOP

NUM. es el número de ciclos variables de loop dividido entre 4 **DE** que son los ciclos de reloj del microprocesador que se **CICLOS** gastan por cada loop que se ejecuta.
TEORICO

NUM. este es el parámetro real que el programa **GRAPHICS** para el **DE** microprocesador como tiempo de espera, para lograr el **CICLOS** nor error posible en la escala de tiempo.
REAL

% DE es el % de desviación entre el número de ciclos real y el **DESV.** número de ciclos teórico.

Los errores máximos que se obtuvieron después de una serie de muestreos se observan en la tabla VII.2.

esc	t/div µs	% de error
1	150	5.3%
2	250	1.5%
3	500	2.4%
4	1000	1.1%
5	2500	.7%
6	5000	1.9%
7	10000	1.1%
8	50000	1.1%

Tabla VII.2

Cabe aclarar que los errores derivados de fijar el periodo de tiempo de la señal a muestrear a través del osciloscopio analógico se consideran 0, dado que la base de medición de periodos fué el mismo osciloscopio.

VII.2.2 Eje Y: voltaje.

Este eje se ajusta automáticamente a los valores máximo y mínimo de la señal muestreada.

El rango de voltajes que se puede medir para el sistema va de 0 a 4.5 V. Este puede ser ampliado por medio de un atenuador de voltaje a la entrada (p.e. de 0 a 45 V si se usa un atenuador de /10) y

modificando los factores por los que se multiplican los niveles de voltaje conseguidos cuando el ADC hace el muestreo. Incluso se pueden medir voltajes negativos y positivos, si se coloca una interfaz de entrada conveniente y se modifica el software (p.e. -12 a +12 V).

El problema fundamental para lograr medidas correctas de voltaje se remite al voltaje de referencia del ADC. El ADC posee dos terminales de referencia (Vref(+) y Vref(-)), éstas son usadas por el convertidor para comparar el voltaje de entrada referenciado a tierra (Vin) y asignarle el nivel discreto (0-255) correspondiente. Si los niveles de referencia no permanecen constantes se tienen errores en la conversión (ver hoja de datos de ADC820)

La fuente de poder de la PC provee voltaje de +- 5 V, pero éste presenta un rizo considerable, por lo que no se debe conectar el Vref(+) directamente a la fuente. En el sistema se conecta a un arreglo con un diodo zener y una resistencia, lográndose mantener fijo Vref en 4.5V. El arreglo se muestra en la figura VII.1.

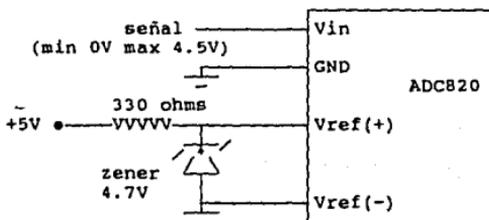


Figura VII.1

Una vez resuelto el problema de mantener el voltaje de referencia fijo, se ajustó al voltaje máximo que se grafica, lo que se logra dividiendo $4.5 \text{ V} / 255$ niveles del ADC. Así se logró que los voltajes se graficaran correctamente. Se considera que hay un error de 3% en las mediciones del sistema, éste es debido a que el osciloscopio con el que se calibró también presenta este error.

El error de cuantización máximo proveniente de la digitalización es tan sólo de $.0088 \text{ V}$ ($4.5\text{V}/512$) lo que significa el 0.2% de la escala máxima.

VII.3.- Dominio de la frecuencia.

Para medir los errores provocados por redondeos en el cálculo de la Transformada Rápida de Fourier (FFT) y por los factores de escala en el programa GRAPHICS se hizo una reconstrucción de la señal muestreada a partir de sus componentes de frecuencia.

Sabemos que una señal puede ser reconstruida de sus componentes de frecuencia a través de la sumatoria:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) w_N^{-kn} \quad n = 0, 1, 2, \dots, N-1$$

donde $x(n)$ es la magnitud del n -ésimo punto de la señal y $X(k)$ es la k -ésima espiga de frecuencia de la señal.

Se hizo un programa en TURBOPASCAL, el cual recibe los componentes real e imaginario de las espigas y genera la magnitud del la señal en tiempo. Este programa se muestra en el apéndice C.

Las señales e reconstruyeron y se pasaron a LOTUS en donde se hizo una comparación del muestreo de la señal original y de la señal reconstruida. Los resultados de esta comparación se muestran en la tabla VII.3 .

Tipo de señal	Señal original contra señal reconstruida		
	Error promedio	Desv. estandar	Error máximo
SENOIDAL (*)	0.04%	0.7%	5.39%
TRIANGULAR (*)	0.054%	0.62%	8.44%

Tabla VII.3

Los errores máximos ocurrieron en aquellos lugares donde la amplitud de la señal era más cercana a cero, esto debido a que los errores de cuantización representan un porcentaje mayor del voltaje medido.

Como se ve los valores derivados de la reconstrucción son muy aproximados a los de la señal original, por lo que podemos consider que el espectro de frecuencia representa fielmente a la señal muestreada.

VII.4 Muestreo de señales.

Se hicieron pruebas muestreando diferentes tipos de señales:

a) Señal senoidal.

Se muestreó una señal con periodo de aproximadamente 240 μ s, ésta se observa en la figura VII.3a. Existe congruencia entre la señal medida en el osciloscopio analógico y la señal desplegada en la pantalla.

En la figura VII.3b se observa su espectro de frecuencia en dB, en ella se muestran dos espigas máximas en alrededor de 5 kHz, lo cual concuerda con la frecuencia de la senoidal: $f = 1/T = 1/240E-6 \text{ s} = 4,166 \text{ Hz}$. Las espigas menores que se observan son debidas a ruido introducido a la señal.

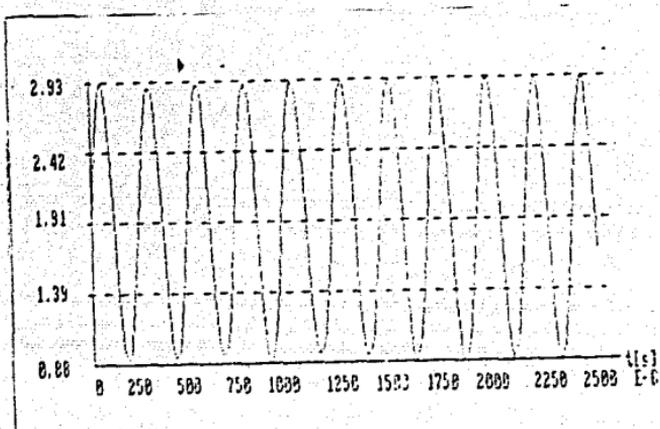


Fig. VII.3a Señal Senoidal

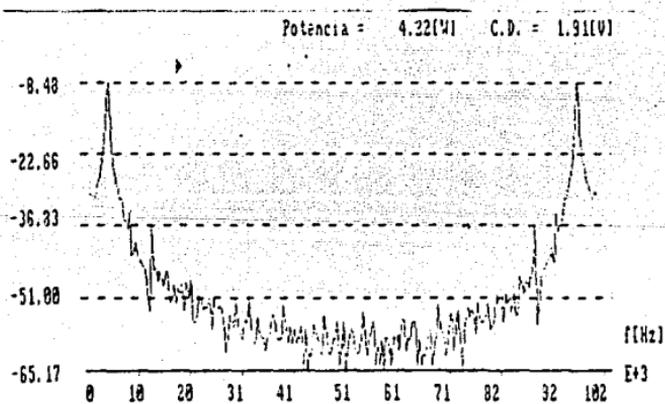


Fig. VII.3b Espectro en Frecuencia de la señal senoidal

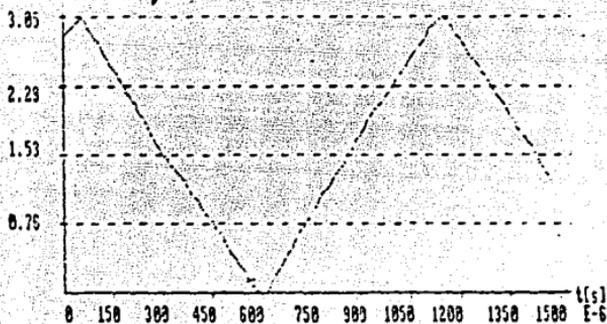


Fig. VII.4a) Señal triangular muestreada con una escala de 150 $\mu s/div$

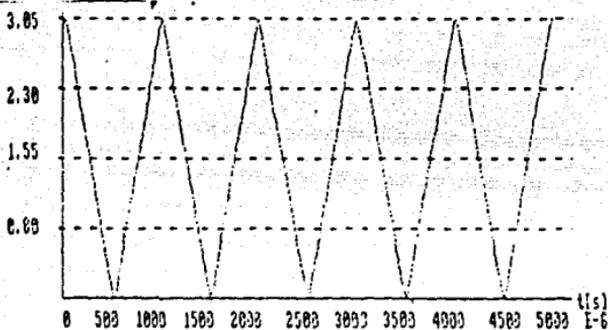


Fig. VII.4b) Señal triangular muestreada con una escala de 500 $\mu s/div$

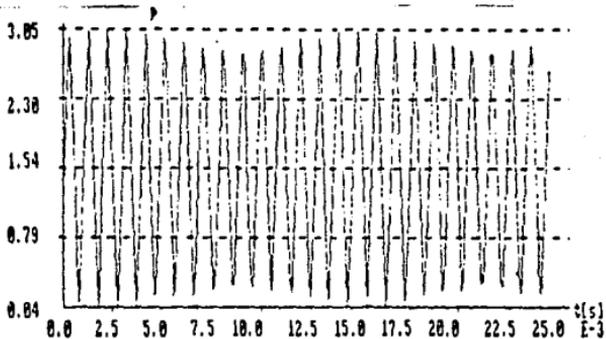


Fig. VII.4c) Señal triangular muestreada con una escala de 2.5 ns/div

POTENCIA = 3.19[W] C.D. = 1.54[V]

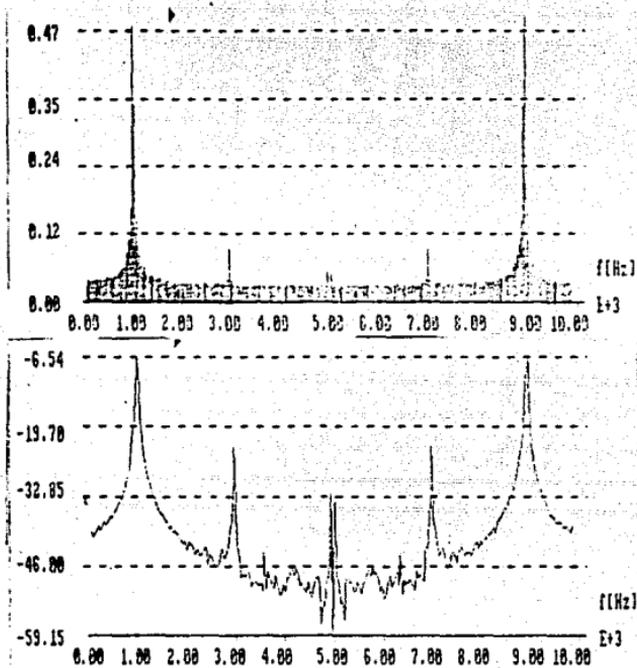


Fig. VII.4c2 y 4c3 Espectros de la señal de la fig. VII.4b1

Potencia = 3.28[W] C.D. = 1.55[V]

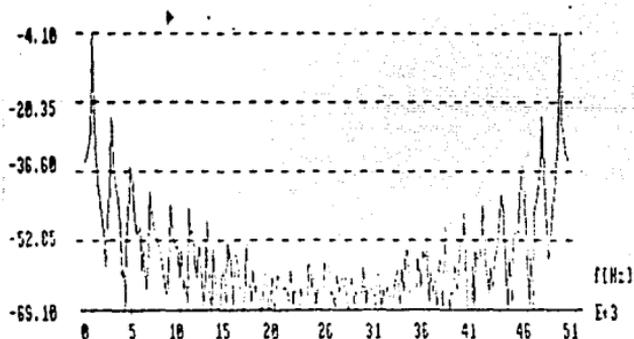


Fig. VII.4b2 Espectro de frec. para la señal de la fig. VII.4c1

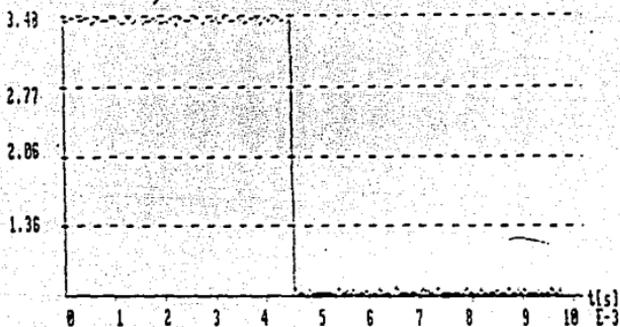


Fig VII.5a Señal rectangular

Potencia = 5.62[W] C.D. = 2.05[V]

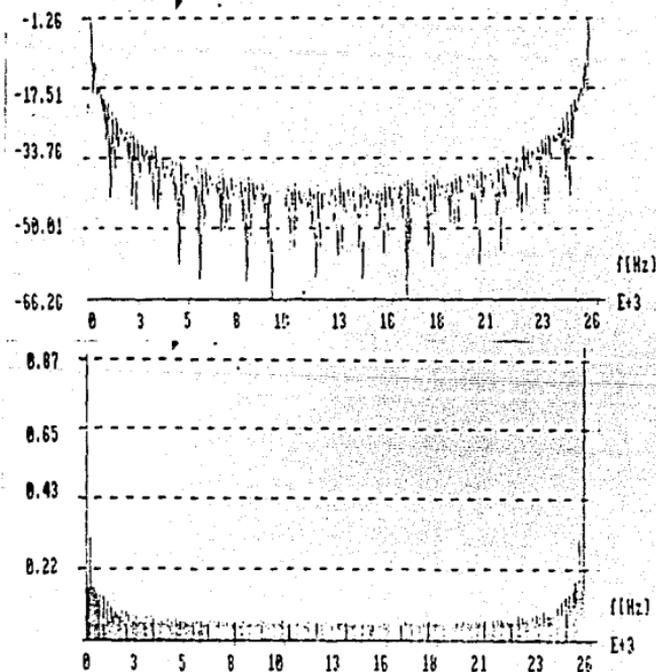


Fig. VII.5b y VII.6 Espectros de la señal de Fig. VII.5a

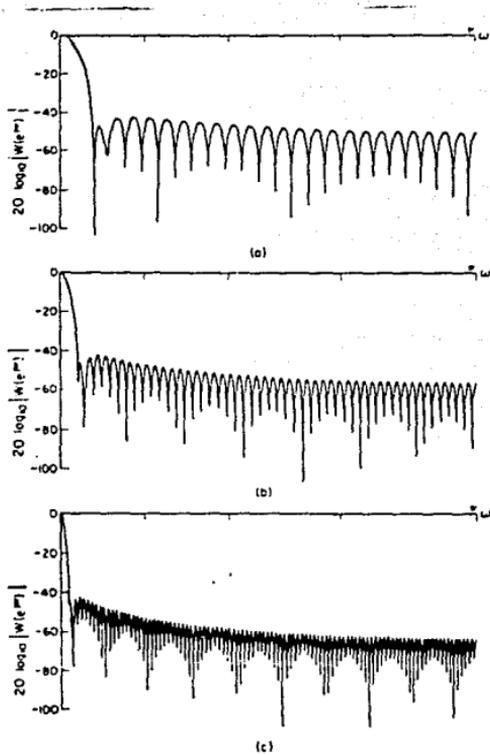


Fig. VII.66.,b,c Espectro teórico para una señal rectangular

b) Señal triangular.

Esta se muestra en diferentes escalas: 1 (150 μ s/div) 3 (500 μ s/div) y 5 (2.5 ms/div) en las figuras VII.4 al,b1 y c1.

También se muestran los espectros de frecuencia al hacer los muestreos con las escalas 3 y 5.

Para la escala 5 se observa el espectro hasta la frecuencia de 10 kHz ($f_{max} = 1/T \text{ muestreo} = 1/98 \mu$ s), tanto en la escala en dB (fig. VII.4c2) como en la escala lineal en Volts (fig. VII.4c3).

La comparación entre la magnitud en dB de las espigas de la gráfica y las magnitudes en dB de los componentes de la serie de Fourier de una señal triangular, muestra que en ambos casos las espigas se presentan en los múltiplos impares de la frecuencia fundamental (para este caso 1000, 3000, 5000, ... Hz). Asimismo observamos que en la serie de Fourier la magnitud de las espigas cae 12 dB/octava, lo que para el cambio de 1000 a 3000 significaría una caída de 19 dB lo cual se verifica aproximadamente en la gráfica VII.4c2.

Para la escala 3, el espectro de frecuencia se muestra en la fig. VII.4b2

c) Señal rectangular.

Se muestreó una señal rectangular en la escala de 1 ms/div. El resultado se aprecia en la figura VII.5a. Esta señal se muestreó como una señal rectangular que se inicia en 0 y termina en 4.5 ms. El resultado de la FFT se equipara con el de la transformada de una ventana de muestreo rectangular:

$$W(e^{j\omega}) = e^{-j\omega((N-1)/2)} \frac{\sin(\omega N/2)}{\sin(\omega/2)}$$

esta fórmula muestra que la transformada consiste en una serie de lóbulos como los que se muestran en la figura VII.5b.

En la figura VII.6 se muestra el efecto en la forma de la FFT al incrementar el número muestras sobre de una ventana semejante a la rectangular (Fig. VII.6a, b, c)¹ de 51 a 101 muestras y de 101 a 201 muestras. Nótese la semejanza con la FFT de la ventana rectangular de 256 muestras.

¹ Ventana Hamming del Oppenheim Alan Y. Digital Signal Processing pg. 548

VII.5 Especificaciones del sistema.

SISTEMA:

(Referido a la sección II.1.4.2)

1.- Tipo de procesamiento:	Tiempo no real.
2.- Tipo de muestreo:	Lineal.
3.- Rango de frecuencia:	2 Hz - 7 KHz (ampliable a 84.66 kHz con S H).
4.- Tasa de muestreo máximo:	169 kmuestras/seg
5.- Acoplamiento:	DC
6.- Canales:	1
7.- Modos:	Señal simple
8.- Número de muestras:	256

OSCILOSCPIO:

Eje vertical.

1.- Sensitividad vertical:	17.65 mV
2.- Escala de voltaje:	0 - 4.5 V
3.- Error máximo:	8.82 mV
4.- Voltaje de disparo:	0 - 4.5 V

Eje horizontal.

1.- Escalas:	150 μ s
	250 μ s
	500 μ s
	1 ms
	2.5 ms
	5 ms
	10 ms
	50 ms

2.- Errores típicos:	Ver tabla VII.2
----------------------	-----------------

ANALIZADOR DE ESPECTROS.

Eje vertical:

1.- Escalas	dB y Volts
2.- Piso de ruido	65 dB
3.- Rango confiable de frecuencia:	2 Hz - 7 KHz (ampliable con S H)
4.- Ventaneo:	Rectangular
5.- Resolución:	254 líneas

CARACTERISTICAS ADICIONALES.

- 1.- Almacenamiento de señal en archivo de datos.
- 2.- Impresión de señales.

VIII. CONCLUSIONES

Existen varios resultados positivos derivados de el desarrollo de este trabajo los cuales mencionaremos a continuación :

1. Podemos afirmar que logramos desarrollar nuestro proyecto en su totalidad. Nuestra tarjeta permite al usuario analizar señales eléctricas sin tener que disponer de un osciloscopio ni de un analizador de espectros, dado que ambos aparatos son emulados fielmente, cumpliéndose las siguientes características :
 - La tarjeta se conecta al slot de expansión de cualquier microcomputadora IBM PC XT compatible.
 - Se maneja un rango de voltajes de 0 a + 5 Volts.
 - Es posible elegir entre 8 diferentes tiempos de muestreo.
 - Se puede elegir el voltaje de disparo para iniciar el muestreo.
 - La gráfica de la señal en el tiempo se muestra en el monitor de video de la PC mostrando sus características principales.
 - Se maneja un ancho de banda de 7 KHz. Este podría ser ampliado instalando un filtro de capacitor conmutado.
 - El espectro de la señal aparece en la pantalla mostrando el valor de la potencia, así como el del voltaje de CD.
 - Toda la información mencionada en estos puntos puede ser almacenada en disco flexible para su posterior estudio.
 - El usuario puede manejar el sistema a través de un programa que le facilita manipular la información como más le convenga.
2. Consideramos que también cumplimos con otro objetivo, que era el de realizar un proyecto de tesis a la altura de nuestra preparación tecnológica. Esto lo afirmamos tomando en cuenta que aplicamos una gran cantidad de los conocimientos aprendidos durante la carrera, pero no sólo eso, también tuvimos que absorber una gran cantidad de conocimientos nuevos, que tal vez no hubiésemos logrado comprender de no ser por las sólidas bases que se nos formaron mientras eramos estudiantes.
3. En el punto anterior hemos hecho referencia a conocimientos nuevos, pero ¿ cuáles ? En primer lugar debemos mencionar que ahora no solo conceptualizamos a una PC como una herramienta de desarrollo de software, sino también para el desarrollo de hardware, dado que conocemos sus características de funcionamiento en una forma bastante detallada. En segundo lugar haremos mención del procesador de señales TMS32010, el cual es uno de los muchos procesadores de uso específico utilizados en la actualidad. En tercer lugar podemos mencionar el conocimiento de algoritmos para el procesamiento de datos tal como lo es la Transformada Rápida de Fourier.
4. Sin embargo no todo se desarrolló en una forma sencilla y rápida. Realmente padecimos bastante por la falta de componentes en el mercado mexicano. Algunos circuitos fueron

conseguidos solicitándolos al extranjero y otros por trueque con personas que de alguna forma podían disponer de ellos. Este no es un problema solamente nuestro, sino que también atañe a otra serie de personas que desean elaborar tesis para áreas afines a la nuestra o que trabajan en proyectos de investigación para esta casa de estudios. Lo anterior nos hace reflexionar en la importancia que debería tener para la U.N.A.M. el establecer algún tipo de convenio para facilitar al estudiantado y a los investigadores el desempeño de su trabajo.

Cabe aclarar que esta fue una de las razones por las cuales no fue posible implementar un filtro digital de entrada que nos permitiera reducir el nivel de ruido de la señal recibida, lo cual nos llevó a manejar un ancho de banda de 7 kHz y no de 100 kHz como se había propuesto originalmente.

5. Por lo que se refiere al ambiente de desarrollo, éste fue bastante singular ya que al no poder asistir a un laboratorio para realizar pruebas tuvimos que echar mano de recursos, algunos propios y otros prestados, tales como osciloscopio, generador de señales, microcomputadora tipo PC y otros implementos diversos teniendo como recinto una casa habitación.
6. El tiempo empleado para el desarrollo fué mucho (más de dos años) dado que se combinaron varios factores, como los mencionados en los puntos (4) y (5) aparte del hecho de que los dos participantes tenemos comprometido nuestro tiempo y bajo estas condiciones no es fácil dedicarse por entero a un proyecto que hubiera tomado, según nuestras consideraciones, aproximadamente de cuatro a seis meses en terminarse se hubiera trabajado en él tiempo completo. Quisieramos remarcar que con esto no estamos emitiendo algún tipo de queja, sino simplemente expresando nuestro sentir respecto a una posible causa del bajo índice de titulación tanto en nuestra carrera como en otras muchas impartidas en la U.N.A.M.
7. Por lo que se refiere a la tarjeta en sí, pensamos que es factible realizar un proceso para su construcción en serie lo cual abatiría en gran medida los costos, aunque debemos reconocer que no se ha realizado ningún estudio de factibilidad en este aspecto.
8. Consideramos que este proyecto puede ser atractivo para diversos tipos de usuarios como pueden ser todos aquellos que se relacionen con el campo de las comunicaciones ya sean estudiantes, investigadores o técnicos.
9. Sin embargo nos damos cuenta de que este proyecto es susceptible de varias mejoras como pueden ser :
 - Filtrar adecuadamente la señal de entrada con un circuito adecuado o bien mediante software.
 - Manipulación de la señal en tiempo real.
 - Manipulación de las escalas (zooms).
 - Obtención de la función de transferencia.
 - Muestreo de varias señales a la vez (manejar más de un

- canal).
- Manejar un convertidor A/D de mayor resolución y velocidad.
 - Emplear un convertidor D/A que permita reconstruir la señal a fin de poder alimentar a algún otro sistema con esta nueva señal de salida.
 - Trasladar el programa realizado en Basic Compilador a otro lenguaje que maneje los recursos en forma más eficiente como puede hacerlo Pascal o C.
 - Aumentar funciones del programa como podrían ser las referentes al filtrado digital, permitiendo elegir el tipo de filtro a utilizar.
 - Despliegue de varias gráficas por medio de ventanas.
 - Análisis estadístico de la señal.

APENDICE A
INSTRUCCIONES DEL TMS32010

1. Introducción.

El conjunto de instrucciones de este μP maneja operaciones tanto para manipulación de datos numéricos para el procesamiento de señales, como operaciones de propósito general. Este conjunto consta básicamente de operaciones que se ejecutan en un ciclo de reloj y que tienen longitud de una palabra, sin embargo, sólo las instrucciones de transferencia de control del programa y las de E/S se ejecutan en más de un ciclo.

2. Formato del direccionamiento de las instrucciones.

A continuación se describen los formatos del código de operación para los diferentes modos de direccionamiento descritos.

a) Formato para direccionamiento directo.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 bits

OPCODE	0	lmd
--------	---	-----

OPCODE = Código de Operación
lmd = Localidad de memoria de datos

Si el bit 7 tiene un valor de 0 lógico se indica un modo de direccionamiento directo. El código de operación está contenido en los bits del 8 al 15 y los bits del 0 al 6 indican la localidad de memoria de datos (lmd).

b) Formato para direccionamiento indirecto.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 bits

OPCODE	1	0	INC	DEC	ARP	0	0	ARP
--------	---	---	-----	-----	-----	---	---	-----

El bit 7 (con valor 1 lógico) indica el modo de direccionamiento indirecto. El código de operación está en los bits del 8 al 15 y los bits del 0 al 6 son para el control del direccionamiento.

Los bits 3 y 0 controlan el apuntador ARP. Si el bit 3 es cero, el contenido del registro indicado por el bit 0 se carga al ARP después de la ejecución, en caso contrario su contenido queda intacto. Si el bit 0 toma el valor de cero, se emplea el AR0, si es uno se emplea el AR1.

Los bits 4 y 5 controlan los registros auxiliares. Si el bit 5 toma el valor de uno, entonces el ARP indica cuál registro auxiliar será incrementado en 1 después de la ejecución, en caso contrario su contenido no se modifica. Por otra parte, si el bit 4 vale 1, el ARP indicará cual registro se decrementará en 1 después de la ejecución, lo cual no sucederá si el bit 4 vale cero.

Los bits 6, 2 y 1 deben ser siempre cero.

c) Formato de direccionamiento inmediato.

Se incluye en las operaciones LDPK, LARK, MPYK, LACK y LARP (vease la sección siguiente) en las que la letra K hace referencia a una constante contenida dentro de la palabra de la instrucción.

3. Conjunto de Instrucciones

a) Instrucciones referidas al acumulador (Ac.)

ABS Valor absoluto del acumulador.
ADD Sumar un dato, con un corrimiento a la izquierda, al Ac.
ADDH Suma un dato con los bits altos (16 al 31) del Ac.
ADDS Suma un dato, ignorando su signo, con el Ac.
AND Realiza la operación lógica Y entre un dato y los bits bajos (0 a 15) del Ac.
LAC El contenido de una dirección de memoria de datos sufre un corrimiento a la izquierda y se carga en el Ac.
LACK Carga el Ac. con una constante de 8 bits.
OR Realiza la operación lógica O entre un dato y los bits bajos del Ac.
SACH Guarda la mitad alta del Ac., con un corrimiento a la izquierda, en la memoria de datos.
SACL Guarda la mitad baja del Ac. en la memoria de datos.
SUB Un dato de memoria sufre un corrimiento a la izquierda y se le resta al acumulador.
SUBC Al Ac. se le resta un dato de memoria. Si los bits altos de la operación dan un resultado mayor a cero, éste se guarda en el Ac., en caso contrario éste permanece sin cambio.
SUBH Resta de la mitad alta del Ac. un dato de memoria.
SUBS Resta del Ac. un dato de memoria, ignorando el signo del dato.
XOR Realiza la operación lógica O EXCLUSIVO entre un dato y los bits bajos del Ac.
ZAC El Ac. se pone en ceros.
ZALH Pone el Ac. en ceros y carga un dato de memoria en los bits altos del Ac.
ZALS Pone el Ac. en ceros y carga un dato de memoria en los bits bajos del Ac. ignorando el signo del dato.

b) Instrucciones de apuntador de página de datos y de registros auxiliares.

LAR Se carga un dato de memoria en el registro auxiliar (Reg. Aux.) actual.
LARK Carga al Reg. Aux. con un dato de 8 bits.
LARP Carga una constante de 1 bit al apuntador ARP para indicar el registro auxiliar a emplear.
LDP El LSB de una localidad de memoria dada se carga en el DP indicando el número de página.
LDPK Carga una constante de 1 bit en el DP indicando el número de página.
MAR El Reg. Aux. actual se incrementa, decrementa o permanece

constante, dependiendo del argumento. Asimismo el ARP se puede modificar.

SAR El contenido del Reg. Aux. actual se guarda en una localidad de memoria.

c) Instrucciones de transferencia de control (salto).

B Salta a la localidad indicada por dirección de memoria de programa alterando el contador del programa (PC).

BANZ El Reg. Aux. se decrementa. Si los bits 0 al 8 de este registro son diferentes de cero la dirección contenida en la siguiente palabra de memoria se carga en el PC. En caso contrario se incrementa el PC.

BGEZ Si el contenido del Ac. es ≥ 0 se hace un salto a una cierta localidad.

BGZ Si el contenido del Ac. es > 0 se hace un salto a una cierta localidad.

BIOZ Si la terminal BIO está activada baja, hace un salto a una cierta localidad.

BLEZ Si el contenido del Ac. es ≤ 0 se hace un salto a una cierta localidad.

BLZ Si el contenido del Ac. es < 0 se hace un salto a una cierta localidad.

BNZ Si el contenido del Ac es $\neq 0$ se hace un salto a una cierta localidad.

BV Si se prende la bandera de overflow (sobreflujo) se hace un salto a una cierta localidad y la bandera se restablece a cero.

BZ Si el contenido del Ac. es $= 0$ se hace un salto a una cierta localidad.

CALA Llamado a una subrutina. El PC se incrementa y se guarda en el stack. Entonces los 12 bits menos significativos (LSBs) del Ac. se cargan en el PC.

CALL Llamado a una subrutina. El PC se incrementa y se guarda en el stack. Entonces se carga una dirección de memoria de programa en el PC.

RET Se saca del stack la dirección de regreso de una subrutina a donde fué llamada y se carga en el PC.

d) Instrucciones para los registros T, P y multiplicación.

APAC Suma el contenido del registro P con el Ac.

LT Carga un dato de memoria al registro T.

LTA Suma el contenido del registro P con el Ac y carga un dato de memoria al registro T.

LTD Suma el contenido del registro P con el Ac, carga un dato de memoria al registro T y transfiere el contenido de una localidad especificada en la siguiente.

MPY Multiplica el contenido del registro T por un dato de memoria y guarda el resultado en el registro P.

MPYK Multiplica el contenido del registro T por una constante y guarda el resultado en el registro P.

PAC Carga el acumulador en el registro P.

SPAC Resta el registro P del acumulador.

e) Instrucciones de control.

DINT Deshabilita la bandera de interrupciones mascarables (INTM).
EINT Habilita la bandera INTM
LST Recupera el contenido del registro de estatus, previamente salvado en una palabra de memoria de datos.
NOP No se ejecuta operación alguna.
POP Extrae el último dato introducido al stack y lo guarda en el Ac.
PUSH Guarda el contenido del Ac. en el stack.
ROVM Pone en ceros el registro de modo de overflow.
SOVM Activa el registro de modo de overflow.
SST Almacena el contenido del registro de estatus en una cierta localidad de memoria.

f) Operaciones de memoria de datos y de E/S.

DMOV Copia el contenido de una cierta localidad de memoria en la siguiente localidad.
IN Accesa un dato de un puerto y lo guarda en una localidad de memoria de datos.
OUT Transfiere un dato de memoria a un periférico externo.
TBLR Transfiere una palabra de una localidad en la memoria de programa a una localidad en la memoria de datos.
TBLW Transfiere una palabra de una localidad en la memoria de datos a una localidad de una RAM de programa externa.

APENDICE B

MANUAL DEL USUARIO

1. Invocando al Programa

El sistema GRAPHICS permite al usuario utilizar la información muestreada a partir de la Tarjeta de Adquisición de Datos. Para correr el programa basta con escribir, estando a nivel de sistema operativo :

A> GRAPHICS

Hecho lo anterior, se presentará la siguiente pantalla :

ARCHIVO	MUESTREO	PANTALLA

Donde Archivo, Muestreo y Pantalla son las opciones que el usuario puede elegir para manejar la información.

La forma de elegir una opción es presionando la primera letra de ésta. El mismo criterio se aplica para las sub-opciones que aparezcan a continuación.

A continuación se describe cada una de las opciones de acuerdo al orden lógico con que deberían ser accedidas.

2. MUESTREO

Este módulo permite establecer los parámetros bajo los cuales se hará el muestreo. En este caso la pantalla mostrada será :

ARCHIVO	MUESTREO	PANTALLA
	T/DIV INICIAR	

2.1 T/DIV (Tiempo por división)

En este caso podemos alterar la frecuencia de muestreo ($f = 1/t$), la pantalla se nos muestra en este caso de la siguiente manera :

ARCHIVO	MUESTREO	PANTALLA
	T/DIV	[1] 125 μ s [2] 250 μ s [3] 500 μ s [4] 1 ms [5] 2.5 ms [6] 5 ms [7] 10 ms [8] 50 ms ?

En este momento el usuario deberá elegir una opción que estará en el rango entre 1 y 8. Hecho lo anterior se regresará a la pantalla original.

2.2 INICIAR

Al elegir esta opción indicaremos al sistema el momento en que debe iniciar el muestreo de datos respondiendo a la siguiente pregunta

ARCHIVO	MUESTREO	PANTALLA
	INICIAR	V. DE DISPARO ($0 \leq V \leq 5$) ?

Donde el voltaje de disparo es aquel que al ser alcanzado por la señal para que inicie el muestreo. En caso de no cumplirse esta condición después de un cierto tiempo, aparecerá el letrero :

NO SE ALCANZO EL VOLTAJE DE DISPARO

y se repetirá la pregunta.

Por otra parte, una vez cumplida la condición, el sistema realizará el muestreo. Cuando este proceso termine se escuchará una campana.

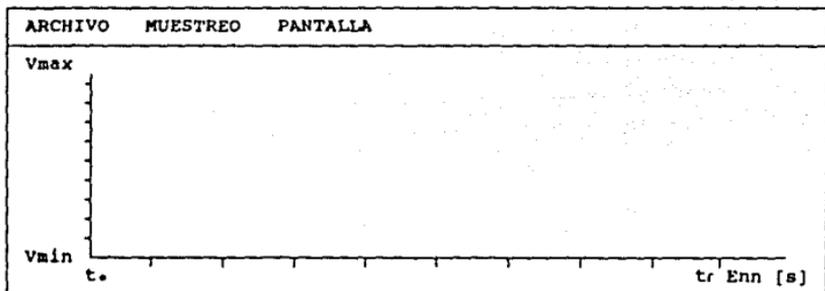
3. PANTALLA

Este módulo tiene como objetivo presentar en la pantalla o en la impresora la gráfica correspondiente a la información muestreada o la leída de un archivo. Cuando se elija esta opción se presentarán las siguientes opciones :

ARCHIVO	MUESTREO	PANTALLA
		TIEMPO FRECUENCIA IMPRIMIR

3.1 TIEMPO

Al elegir esta pantalla se nos mostrará la gráfica de la función en el dominio del tiempo :



En este caso V_{max} corresponderá al Voltaje máximo alcanzado por la señal y V_{min} al voltaje mínimo, t_0 al tiempo inicial y t_r al tiempo final. En el caso del eje de tiempo este se mostrará en notación exponencial, siendo nn el exponente.

Cabe aclarar que la gráfica siempre estará sujeta a un escalamiento tanto vertical como horizontal automático, de tal suerte que siempre ocupará el mismo espacio en la pantalla aún cuando varíen la magnitud y el tiempo de muestreo. El mismo criterio se aplica para la siguiente gráfica.

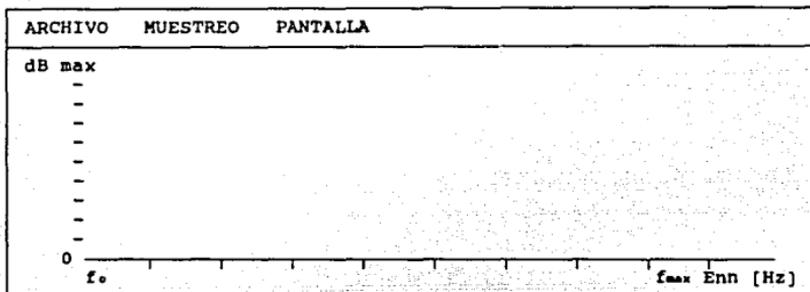
3.2 FRECUENCIA

En este caso se nos preguntará la presentación que se desea para la gráfica de la función en el dominio de la frecuencia :

ARCHIVO	MUESTREO	PANTALLA	1. dB	2. Volts ?
		FRECUENCIA		

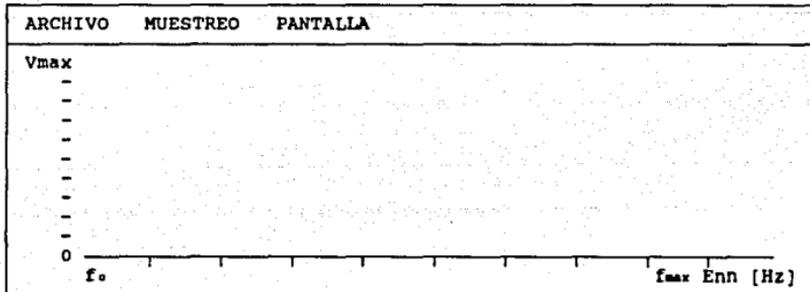
3.2.1 Decibeles (dB).

La forma de una gráfica de frecuencia en decibeles es del siguiente tipo :



3.2.2. Volts

Cuando el usuario eligimos esta opción se nos presenta una gráfica del siguiente tipo :



Donde Vmax representa el Voltaje alcanzado por la espiga de mayor magnitud (componente de CD), f_0 la frecuencia inicial y f_{max} la frecuencia máxima de la señal presentadas con notación exponencial.

A continuación se nos pregunta :

ZOOM (S/N) ?

En caso de responder S (Si) se nos preguntará el rango de frecuencias que queremos expandir :

Frecuencia Inicial (Hz) : _____

Frecuencia Final (Hz) : _____

Una vez proporcionados estos datos al sistema, este nos presentará una gráfica similar a la anterior pero que muestre solamente el rango expandido que tenga como límites las frecuencias indicadas.

3.3 IMPRIMIR

Esta tercera opción de pantalla realiza un "hard copy" de la pantalla en el papel. Esto quiere decir que si se desea imprimir la gráfica de la función en el tiempo, habrá que tenerla presente en la pantalla antes de llamar a esta opción. El mismo criterio se aplica para la gráfica en frecuencia. Asimismo es necesario que la impresora esté conectada, encendida y en línea antes de ejecutar la opción, pues de otra manera no se asegura la integridad de la información.

4. ARCHIVO

La función de este módulo es la de permitir al usuario guardar la información que aparece en la pantalla en un archivo para su posterior análisis. Estos archivos están en formato ASCII por lo que pueden ser generados o editados por el usuario. Para este caso las sub-opciones son :

ARCHIVO	MUESTREO	PANTALLA
RECUPERAR		
GRABAR		
FIN		

4.1 RECUPERAR

Como su nombre lo indica, esta rutina permite leer un archivo de disco. En este caso el sistema preguntará :

ARCHIVO	MUESTREO	PANTALLA	ARCHIVO : ?
RECUPERAR			

a lo cual se deberá responder con un nombre válido de archivo :

XXXXXXXX.YXX = 8 caracteres (máximo), punto, extensión (3 caracteres).

Existe un manejo de errores que considera los siguientes casos :

- a) Nombre incorrecto del archivo - Cuando no se cumple con las restricciones expuestas anteriormente.
- b) Archivo no encontrado
- c) El disco está lleno
- d) La unidad de disco no está lista
- e) Existe error en el disco
- f) Otro error - Código

Este control actúa también para la siguiente opción.

4.2 GRABAR

Una vez que se ha muestreado o leído información de un archivo, esta puede ser grabada en disco. Para el nombre del archivo no hay una extensión por omisión, por lo que el usuario deberá escribirla si es que la desea.

ARCHIVO	MUESTREO	PANTALLA	ARCHIVO : ?
GRABAR			

Se aplica el mismo criterio que en el punto anterior.

4.3 TERMINAR

Esta opción nos permite abandonar el programa y regresar a nivel de sistema operativo.

ARCHIVO	MUESTREO	PANTALLA	Presione <ENTER> para terminar
FIN			

Si el usuario presiona la tecla <ENTER> se terminará la ejecución del programa, en caso contrario se podrá ejecutar otra opción cualquiera.

3. Características del sistema :

Para poder utilizar el sistema se requiere de una computadora IBM-PC, IBM-PC XT o compatible empleando el sistema operativo MS-DOS versión 2.1 o mayor.

Se requiere de manejar el paquete con un monitor tipo Color Graphics Adapter (CGA) o equivalente, o en su defecto utilizar un programa emulador cuando se tenga otro tipo de monitor.

Es necesario utilizar una impresora que soporte el modo gráfico proporcionado por el programa GRAPHICS.COM del sistema operativo MS-DOS.

APENDICE C
LISTADOS DE LOS PROGRAMAS

title Programa de carga de programas de TMS
page 60,132

```
; 20-jul-89
; Lee el archivo prog.dat y lo carga en la memoria del tms
; de 944k a 944k+3FFFh EC00:0000 EE00:0000
; DO-D7 D8-D15
;
;
; La información del programa se carga del archivo prog.dat
; el formato de este archivo es nnnn<LF><CR> por instrucción
; donde nnnn es un dígito hexadecimal. Por lo tanto debe hacerse
; preprocesamiento del archivo .LST de XASM3 del TMS para dejar el
; archivo de carga en el formato adecuado. El programa que hace
; la transformación es LST2EXE.ASM
```

```
;stack segment stack
; dw 512 dup(?)
;stack ends
```

```
DATA SEGMENT WORD PUBLIC 'DATA'
nomfile db 'prog.dat',00h
inst db 2 dup(?)
DATA ENDS
```

DGROUP GROUP DATA

```
code segment WORD PUBLIC 'CODE'
assume cs:code
assume DS:DGROUP,ES:DGROUP
```

PUBLIC CARGA

carga proc far

```
;
; push ds
; mov ax,0
; push ax

PUSH BP ; establece el marco para que se
MOV BP,SP ; puedan acceder los parametros.
PUSH DI ; guarda los registros que se van
PUSH SI ; a alterar a través del programa..
PUSH ES
PUSH DS
PUSH SS
PUSH AX
PUSH BX
PUSH CX
PUSH DX

; mov ax,data
; mov ds,ax
; mov es,ax ; regreso
```

;codigo principal

```
lea dx, noafile ; abre el archivo
mov al, 0
mov ah, 3Dh
int 21h
push ax
```

```
mov cx, 1000h ; contador de bytes
mov bx, 0000h ; dirección de inicio de carga
```

sal1:

```
mov ax, data
mov ds, ax
pop ax
push ax
push bx
mov bx, ax
lea dx, inst
push cx
mov cx, 2
mov ah, 3Fh ; lee el siguiente byte bajo del archivo
int 21h
cmp ax, 0000h
jz sal2
call hex_bin ; regresa en ah valor del byte bajo en binario de
buffer
```

```
pop cx
pop bx
```

```
push ax
mov ax, 0EE00h
mov ds, ax
pop ax
mov [bx], al
```

```
mov ax, data
mov ds, ax
pop ax
push ax
push bx
mov bx, ax
lea dx, inst
push cx
mov cx, 2
mov ah, 3Fh ; lee el siguiente byte alto del archi
int 21h
cmp ax, 0000h
jz sal2
call hex_bin ; regresa en ah valor del byte bajo en binario de
buffer
```

```
pop cx
pop bx
```

```
push ax
mov ax, 0EC00h
mov ds, ax
pop ax
```

```

    mov [bx],al
; quita los bytes de LF CR

    mov ax,data
    mov ds,ax
    pop ax
    push ax
    push bx
    mov bx,ax
    lea dx,inst
    push cx
    mov cx,2
    mov ah,3Fh           ;lee el siguiente byte alto del archi
    int 21h
    cmp ax,0000h
    jz sal2
    pop cx
    pop bx

    inc bx

loop sal1

sal2:
    pop cx           ; pops antes de salto
    pop bx
    pop bx           ; cierra el archivo pop de handle
    mov ah,3Eh
    int 21h

POP DX ; reestablece todo para QBASIC.
POP CX
POP BX
POP AX
POP SS
POP DS
POP ES
POP SI
POP DI
POP BP

ret           ; proceso externo

```

```
carga endp
```

```

PUBLIC HEX_BIN
hex_bin proc

```

```
; Transforma un byte en hexadec a binario
```

```
push bx
```

```
lea bx,inst
mov ah,[bx]
```

```
;carga al con el valor hex
;del byte alto del buffer inst
```

```
cmp ah,41h
```

```
; checa si es numero 0-9 o letra A-F
```

```

jnl letra                ; y coloca los 4 bits segun el código hex
  sub ah,30h
  jmp comple

letra:                   sub ah,37h

comple:
  mov al,ah
  shl ax,1
  shl ax,1
  shl ax,1
  shl ax,1

  mov ah,[bx+1]          ;carga al con el valor hex
                        ;del byte bajo del buffer inst

  cmp ah,41h            ; checa si es numero 0-9 o letra A-F
  jnl letra2           ; y coloca los 4 bits segun el código hex
  sub ah,30h
  jmp comple2

letra2:                  sub ah,37h

comple2:
  or al,ah
  pop bx
  ret

hex_bin endp

code ends
end

```

title ON OFF TMS32010
page 60,80

```
; Pasa de uso de 8088 a tms escribiendo en el puerto
; 01FFh      9 8 7 6 | 5 4 3 2 1 0      64 localidades cada grupo ptos.
;           0 1 0 0 | 0 0 0 0 0 0

; Pasa de uso de tms a 8088 escribiendo en el puerto
; 01FFh      9 8 7 6 | 5 4 3 2 1 0      64 localidades cada grupo ptos.
;           0 1 0 1 | 0 0 0 0 0 0
```

code segment WORD PUBLIC 'CODE'
assume cs:code

PUBLIC ENTRA

entra proc far
;

```
PUSH DI
PUSH SI
PUSH ES
PUSH DS
PUSH AX
PUSH DX
```

```
; push ds
; mov ax,0
; push ax
; mov ax,data
; mov ds,ax
; mov es,ax
```

;codigo principal

```
mov dx,0140h          ;Usa TMS
mov ax,0FFFFh
out dx,ax
```

mov cx,115CH

dummi:

loop dummi

```
mov dx,01C0h          ;; out para desactivar IR2 alta
mov ax,0FFFFh          ;; de 74LS74 de interrupciones
out dx,ax
```

```
mov dx,0100h          ;Usa PC
mov ax,0FFFFh
out dx,ax
```

```
POP DX
POP AX
POP DS
POP ES
POP SI
POP DI
```

```
ret                ;; proceso externo
entra endp
code ends
end entra
```

title Programa de impant
page 60,132

; Programa de impresion de pantalla (printscreen)
; 23 jul 90

code segment WORD PUBLIC 'CODE'
assume cs:code

PUBLIC IMPANT

impant proc far
;

PUSH DI
PUSH SI
PUSH ES
PUSH DS
PUSH AX
PUSH DX

;codigo principal

int 05h

POP DX
POP AX
POP DS
POP ES
POP SI
POP DI

ret

impant endp

code ends

end impant

title Programa de atencion a IR2
page 60,132

; Programa de rutina de atencion para IR2
; 8 jul 90

code segment WORD PUBLIC 'CODE'
assume cs:code

;,ds:data,es:data

PUBLIC ATEN

aten proc far
;

PUSH DI
PUSH SI
PUSH ES
PUSH DS
PUSH AX
PUSH DX

;codigo principal

mov ax,code
mov ds,ax
lea ax,rut_atencion
mov dx,ax

mov al,0Ah ;; coloca el vector de interrupcion en
mov ah,25h ;; la int 0Ah
int 21h

mov dx,0021h ;; OCW1 habilita inter. 2
mov al,00000000b
out dx,al

mov dx,0140h ;; out para activar TMS
mov ax,0FFFFh
out dx,ax

dumm: cmp al,00h ;; espera a que se active IR2
jne dumm ;; la entrada es con al = FFh
;; la rutina de atencion hace al al=00h

POP DX
POP AX
POP DS
POP ES
POP SI
POP DI

ret

aten endp

PUBLIC RUT_ATENCION

rut_atencion proc far

mov dx,01C0h ;; out para desactivar IR2 alta

mov ax,0FFFFh ;; de 74LS74 de interrupciones
out dx,ax

mov dx,0020h ;; envia comando EOI
mov al,00100000b
out dx,al

mov dx,0100h ;; Desactiva TMS
mov ax,0FFFFh
out dx,ax

mov ax,0000h

; sti ;; rehabilita interrupciones
iret ;; proceso externo

rut_atencion endp

code ends

end aten

```
PROGRAM REGEN (INPUT,OUTPUT,ENTRA,SALE);
```

Regenera una señal de la cual se conocen sus componentes de frecuencia por medio de la formulas $x(n) = 1/N \sum(X(k) w^{nk})$

N

```
VAR
```

```
ARCHOUT,ARCHIN : STRING;  
XRE,XIM : ARRAY[0..255] OF DOUBLE;   VECTOR DE SENAL EN FRECUENCIA  
YRE,YIM : ARRAY[0..255] OF DOUBLE;   VECTOR DE SENAL EN TIEMPO  
ENTE : ARRAY[0..255] OF LONGINT;  
WN : DOUBLE; 2*PI/N  
PARRE,PARIM : DOUBLE;  
COSITO,SINITO : ARRAY[0..255] OF DOUBLE;  
I,J,K : LONGINT;  
ENTRA,SALE : TEXT;
```

```
BEGIN
```

```
WRITELN('ARCH. DE ENTRADA?');  
READLN(ARCHIN);  
WRITELN('ARCH. DE SALIDA?');  
READLN(ARCHOUT);  
ASSIGN(ENTRA,ARCHIN);  
ASSIGN(SALE,ARCHOUT);  
RESET(ENTRA);  
REWRITE(SALE);
```

```
WN := 2*PI/256;
```

```
FOR I:=0 TO 255 DO
```

```
  BEGIN
```

```
    COSITO[I] := COS(WN*I);
```

```
    SINITO[I] := SIN(WN*I);
```

```
  END;
```

```
FOR I:=0 TO 255 DO
```

```
  READLN(ENTRA,ENTE[I],XRE[I],XIM[I]);
```

```
FOR I := 0 TO 255 DO
```

```
  BEGIN
```

```
    PARRE := 0;
```

```
    PARIM := 0;
```

```
    WRITELN(I);
```

```
    FOR J:= 0 TO 255 DO
```

```
      BEGIN
```

```
        K := (I*J) MOD 256;
```

```
        PARRE := PARRE + XRE[J]*COSITO[K] - XIM[J]*SINITO[K];
```

```
        PARIM := PARIM + XRE[J]*SINITO[K] + XIM[J]*COSITO[K];
```

```
      END;
```

```
    YRE[I]:= PARRE;
```

```
    YIM[I]:= PARIM;
```

```
  END;
```

```
FOR I:= 0 TO 255 DO
```

```
  WRITELN(SALE,I:4,',',YRE[I]:8:2,',',YIM[I]:8:2);
```

```
CLOSE(ENTRA);
```

```
CLOSE(SALE)
```

```
END.
```

IDT 'TOTAL'

* Código del TMS 32010 para realizar la lectura de datos y la FFT para
* la tarjeta emuladora de osciloscopio y analizador de espectros
*
*

* Programa de Cooley-Tukey de Base 4, para FFT

* Mariposa de Base simple 4
* Utiliza una tabla de búsqueda con Factores de Giro (Twiddle)
* Los datos en la RAM externa se asumen como complejos con sus
* correspondientes partes real e imaginaria en localidades
* consecutivas.

* Modificado por Nelson Martinez 23-9-89

* N es el tamaño de la transformada ($N = 4**M$).

*
N EQU 256
M EQU 4

* Utilización de la memoria de datos.

XI EQU 0 * Valores para la mariposa
YI EQU 1
XI1 EQU 2
YI1 EQU 3
XI2 EQU 4
YI2 EQU 5
XI3 EQU 6
YI3 EQU 7
I EQU 8 * Indices de datos
I1 EQU 9
I2 EQU 10
I3 EQU 11
CO1 EQU 12 * Coeficientes para los factores de giro
CO2 EQU 13
CO3 EQU 14
SI1 EQU 15
SI2 EQU 16
SI3 EQU 17
N1 EQU 18 * Incremento para I
N2 EQU 19 * Separación de índices
IA EQU 20 * Índice para los factores de giro
IE EQU 21 * Incremento para IA
J EQU 22 * Contadores
K EQU 23
R1 EQU 24 * Variables temporales
R2 EQU 25
S1 EQU 26
S2 EQU 27
R3 EQU 28

R4 EQU 29
S3 EQU 30
S4 EQU 31
YJ EQU 32
YJ EQU 33
TEMP EQU 34
HOLDN EQU 35
QUARTN EQU 36
ONE EQU 37
TABLE EQU 38

- * Contiene el valor de N
- * Contiene el valor N/4
- * Contiene el valor 1
- * Localización de la tabla de coeficientes

* Variables no usadas en FFT

DATADC EQU 39
ZERO EQU 40
TDIV EQU 41
AUX EQU 42

*
*

AORG 0
B DEBUT

* Localidad 2 del TMS

NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP

* Localidad 10d

NOP
NOP

* Localidad 20d

NOP
NOP

- * Localidad 20h parametro de tiempo
- * Voltaje de disparo (poner en formato 00VV)
- * No disparo: error (00FF) no error (0000)

NOP
NOP

```

*
* carga Memoria de Datos 1 con 1 para incrementar acumulador
*
DEBUT   LACK 1
        SACL ONE
        ZAC
        SACL ZERO
*
        LACK 32           * Escribe el parametro de tiempo por
        TBLR TDIV        * division en la localidad TDIV (RAM)
*
        LACK 0           * Escribe el codigo 0000h en 22h
        SACL DATADC      * indica que no hay error
        LACK 34
        TBLW DATADC
*
        LDPK 0
        LARP 1

* realiza lecturas dummy al ADC antes de poder comenzar las lecturas
* correctas
        LARK AR1,8
INICO   IN 3,5           * Activa RD del ADC
WARTO   BIOZ ATTO       * Espera a que los datos esten disponibles en el A
        B WARTO
*
ATTO    IN DATADC,7
        IN 3,6
        BANZ INICO
*
* carga ar0 con 255 para contar numero de datos insertados
        LARK ARO,255
*
* los datos muestreados se colocan en localidades contiguas (solo
* se ocupan 256 localidades de TMS el byte alto es basura y el bajo
* contiene los 8 bytes generados por el ADC,)
*
* Carga TEMP con 7 para despues generar la direccion de inicio de
* traspaso de los datos del osciloscopio (700h)
        LACK 7
        SACL TEMP

* lee el voltaje de disparo que esta en ROMI 21h y lo coloca en RAMI R
* (el nivel de disparo debe estar como FF<nivel>)
        LAC ONE,5
        ADD ONE
        TBLR R1

* lee voltajes y compara contra voltaje de disparo para el uso del ADC
        LARK AR1,255
COMP1   CALL WASTE
        IN 3,5           * Activa RD del ADC y deja en stand alone
WART1   BIOZ ATT1
        B WART1
ATT1    IN DATADC,7
        IN 3,6
        BANZ COMP2
        LACK 255
        SACL DATADC
        * Desactiva RD
        * Verifica que no se hayan hecho 255 lectura
        * antes de encontrar el voltaje de disparo
        * escribe error 00FFh en 22h

```

```

LACK 34
TBLW DATADC
IN 2,4          * Activa IR2
B STOP         * Fin de programa
*
COMP2  LACK 255      * Compara contra voltaje de disparo y toma
AND DATADC
SUB R1         * otra muestra si este no se ha alcanzado
BLZ COMP1
*
LARP 0
* carga acumulador con la direccion de traspaso 0111 0000 0000 b
LAC TEMP,8
*
* inicia la carga de datos
INICIA IN 3,5      * Activa RD
*
* espera a que el ADC realice la conversion
ESPERA BIOZ ATIEND
B ESPERA
*
* lee dato de dmd, dasactiva RD de ADC, pone el dato en la localidad de
* memoria de programa correspondiente a partir de 800h, regresa al loop
ATIEND IN DATADC,7 * lee dato de ADC en localidad DATADC(39) de RA
IN 3,6          * Desactiva RD
TBLW DATADC
CALL WASTE
ADD ONE        * Incrementa dir de localidad a escribir en
(dir = ACC)
BANZ INICIA
*****
* Copia los datos de la localidad 700h (formato real con solo 256
* localidades a la localidad 800h (formato complejo con 512
* localidades) y en los bits indicados 0000 00bb bbbb bb00.
*****
LARK ARD,255   * Numero de localidades a barrer
LAC TEMP,8
SACL R1        * Guarda la localidad fuente de inicio (700h)
LAC ONE,11
SACL R2        * Guarda la localidad destino de inicio (800h)
*
* Genera numero 03FCh para realizar el AND de eliminacion de basura
LAC ONE,8
SUB ONE
SACL R3
LAC R3,2
SACL R3
*
* Inicia el intercambio de localidades
SWAP  LAC R1
TBLR DATADC   * Lectura del dato fuente
ADD ONE      * Incrementa la direccion para leer sig. dato
SACL R1
LAC DATADC,2

```

*
 * Realiza AND con 03FCh para quitar los bits de basura del formato del
 AND R3
 SACL DATADC
 LAC R2
 TBLW DATADC * Escribe el dato real en localidad par
 ADD ONE
 TBLW ZERO * Escribe dato imag. en localidad impar (siem
 ADD ONE
 SACL R2
 BANZ SWAP

 * Comienzo de la sección de FFT

LDPK 0
 LAC ONE
 SACL IE * Inicializa IE = 1
 LT ONE
 MPYK SINE
 PAC
 SACL TABLE * Guarda la dirección de la tabla de coeficientes
 MPYK N
 PAC
 SACL HOLDN
 SACL N2 * Initialize N2 = N
 LAC HOLDN,14
 SACH QUARTN * QUARTN = N/4
 LARK ARO,M-1 * ARO contiene el valor de K

KLOOP LARP 1
 LAC N2
 SACL N1 * N1 = N2
 LAC N2,14
 SACH N2 * N2 = N2/4
 ZAC
 SACL J
 SACL IA
 LAR AR1,N2 * AR1 contiene el valor de J
 MAR *- * Comienza en N2-1
 JLOOP LAC TABLE
 ADD IA
 TBLR SI1 * Obtiene el factor de Giro 1
 ADD QUARTN
 TBLR CO1
 LAC TABLE
 ADD IA,1
 TBLR SI2 * Obtiene el factor de Giro 2
 ADD QUARTN
 TBLR CO2
 LAC TABLE
 ADD IA,1
 ADD IA
 TBLR SI3 * Obtiene el factor de Giro 3
 ADD QUARTN
 TBLR CO3
 LAC IA

```

ADD IE
SACL IA * IA = IA + IE
LAC J,1
SACL I * I = J (Datos organizados por un valor real
* seguido por un imaginario. Así la dirección
* de I es 2 veces J).
*
*
ILOOP LAC I
ADD ONE,11 * Añade el offset 800h de la zona de datos (en
ADD N2,1
SACL I1 * I1 = I + N2
ADD N2,1
SACL I2 * I2 = I1 + N2
ADD N2,1
SACL I3 * I3 = I2 + N2
*
*
LAC I * Carga la dirección de XI
ADD ONE,11 * Añade el offset 800h de la zona de datos
* (en 2K)
*
*
TBLR XI * Lee la parte real e imaginaria
ADD ONE
TBLR YI
LAC I1 * Carga la dirección de XI1
TBLR XI1 * Lee la parte real e imaginaria
ADD ONE
TBLR YI1
LAC I2 * Carga la dirección de XI2
TBLR XI2 * Lee la parte real e imaginaria
ADD ONE
TBLR YI2
LAC I3 * Carga la dirección de XI3
TBLR XI3 * Lee la parte real e imaginaria
ADD ONE
TBLR YI3
*
*
* Calcula la mariposa.
*
LAC XI
ADD XI2
SACL R1 * R1 = X(I) + X(I2)
SUB XI2,1
SACL R3 * R3 = X(I) - X(I2)
LAC YI
ADD YI2
SACL S1 * S1 = Y(I) + Y(I2)
SUB YI2,1
SACL S3 * S3 = Y(I) - Y(I2)
LAC XI1
ADD XI3
SACL R2 * R2 = X(I1) + X(I3)
SUB XI3,1
SACL R4 * R4 = X(I1) - X(I3)
LAC YI1
ADD YI3
SACL S2 * S2 = Y(I1) + Y(I3)
SUB YI3,1
SACL S4 * S4 = Y(I1) - Y(I3)
*
*
LAC R1
ADD R2

```

SACL XI * X(I) = X0 + X1 + X2 + X3
 SUB R2,1
 SACL R2 * R2 = X0 + X2 - X1 - X3
 LAC R3
 ADD S4
 SACL R1 * R1 = X0 - X2 - Y1 + Y3
 SUB S4,1
 SACL R3 * R3 = X0 - X2 + Y1 - Y3

LAC S1
 ADD S2
 SACL YI * Y(I) = Y0 + Y1 + Y2 + Y3
 SUB S2,1
 SACL S2 * S2 = Y0 + Y2 - Y1 - Y3
 LAC S3
 SUB R4
 SACL S1 * S1 = Y0 - Y2 - X1 + X3
 ADD R4,1
 SACL S3 * S3 = Y0 - Y2 + X1 - X3

LT CO1
 MPY S1
 PAC
 LT SI1
 MPY R1
 SPAC
 SACH YI1,1 * Y(I1) = CO1*S1 - SI1*R1
 MPY S1
 PAC
 LT CO1
 MPY R1
 APAC
 LT CO2
 SACH XI1,1 * X(I1) = CO1*R1 + SI1*S1
 MPY S2
 PAC
 LT SI2
 MPY R2
 SPAC
 SACH YI2,1 * Y(I2) = CO2*S2 - SI2*R2
 MPY S2
 PAC
 LT CO2
 MPY R2
 APAC
 LT CO3
 SACH XI2,1 * X(I2) = CO2*R2 + SI2*S2
 MPY S3
 PAC
 LT SI3
 MPY R3
 SPAC
 SACH YI3,1 * Y(I3) = CO3*S3 - SI3*R3
 MPY S3
 PAC
 LT CO3
 MPY R3
 APAC
 SACH XI3,1 * X(I3) = CO3*R3 + SI3*S3

* Salida resultante de la mariposa

ZAC
LAC XI,15
SACH XI * Escalamiento /2
ZAC
LAC YI,15
SACH YI * Escalamiento /2
ZAC
LAC XI1,15
SACH XI1 * Escalamiento /2
ZAC
LAC YI1,15
SACH YI1 * Escalamiento /2
ZAC
LAC XI2,15
SACH XI2 * Escalamiento /2
ZAC
LAC YI2,15
SACH YI2 * Escalamiento /2
ZAC
LAC XI3,15
SACH XI3 * Escalamiento /2
ZAC
LAC YI3,15
SACH YI3 * Escalamiento /2

LAC I * Carga la dirección de XI
ADD ONE,11 * Añade el offset 800h de la zona de datos
(en 2K)

TBLW XI * Escribe la prate real e imaginaria
ADD ONE
TBLW YI
LAC I1 * Carga la dirección de XI1
TBLW XI1 * Escribe la prate real e imaginaria
ADD ONE
TBLW YI1 * Carga la dirección de XI2
LAC I2 * Escribe la prate real e imaginaria
TBLW XI2 * Escribe la prate real e imaginaria
ADD ONE
TBLW YI2 * Carga la dirección de XI3
LAC I3 * Escribe la prate real e imaginaria
TBLW XI3 * Escribe la prate real e imaginaria
ADD ONE
TBLW YI3

* Incrementos para el siguiente ciclo.

LAC I
ADD N1,1
SACL I * $I = I + N1$
SUB HOLDN,1
BLZ ILOOP

LAC J
ADD ONE
SACL J * $J = J + 1$
BANZ JLOOP

```

LAC IE,2
SACL IE      * IE = IE*4
LARP 0
BANZ KLOOP
*
* Cálculo del contador regresivo para la FFT base 4.
* Hace el reverso de los bits de 256 localidades (8bits)
* 76543210 ----> 10325476
*
DRC4      ZAC
          SACL J
          SACL I
          LARP 0
          LARK ARO,255
DRLOOP   LAC I,10      * Acum 32 bits
          SACL TEMP
          SACH J
          LACK 3      * 00000011b
          AND J      * Elimina bits indeseables.
          SACL J
          LAC TEMP,2
          SACL TEMP
          SACH R1
          LACK 3      * 00000011b
          AND R1
          SACL R1
          LAC R1,2
          ADD J
          SACL J
          LAC TEMP,2
          SACL TEMP
          SACH R1
          LACK 3      * 00000011b
          AND R1
          SACL R1
          LAC R1,4
          ADD J
          SACL J
          LAC TEMP,8
          SACH TEMP
          LACK 192    * 11000000 b
          AND TEMP
          ADD J
          SACL J
*
* Swapea las localidades I y J.
*
          LAC I,1      * Calcula las direcciones de XI
          ADD ONE,11   * Añade el offset 800h de la zona de
*                                     datos (en 2K)
          TBLR XI      * Lee la prate real e imaginaria
          ADD ONE
          TBLR YI
          LAC J,1
          ADD ONE,11
          ADD ONE,9    * Añade el offset A00h de la zona de
*                                     datos (en 2K)
          TBLW XI      * Escribe la prate real e imaginaria

```


DATA 18204
DATA 18867
DATA 19519
DATA 20159
DATA 20787
DATA 21402
DATA 22004
DATA 22594
DATA 23169
DATA 23731
DATA 24278
DATA 24811
DATA 25329
DATA 25831
DATA 26318
DATA 26789
DATA 27244
DATA 27683
DATA 28105
DATA 28510
DATA 28897
DATA 29268
DATA 29621
DATA 29955
DATA 30272
DATA 30571
DATA 30851
DATA 31113
DATA 31356
DATA 31580
DATA 31785
DATA 31970
DATA 32137
DATA 32284
DATA 32412
DATA 32520
DATA 32609
DATA 32678
DATA 32727
DATA 32757

COSINE EQU \$

DATA 32767
DATA 32757
DATA 32727
DATA 32678
DATA 32609
DATA 32520
DATA 32412
DATA 32284
DATA 32137
DATA 31970
DATA 31785
DATA 31580
DATA 31356
DATA 31113
DATA 30851
DATA 30571
DATA 30272
DATA 29955

DATA 29621
DATA 29268
DATA 28897
DATA 28510
DATA 28105
DATA 27683
DATA 27244
DATA 26789
DATA 26318
DATA 25831
DATA 25329
DATA 24811
DATA 24278
DATA 23731
DATA 23169
DATA 22594
DATA 22004
DATA 21402
DATA 20787
DATA 20159
DATA 19519
DATA 18867
DATA 18204
DATA 17530
DATA 16845
DATA 16150
DATA 15446
DATA 14732
DATA 14009
DATA 13278
DATA 12539
DATA 11792
DATA 11038
DATA 10278
DATA 09511
DATA 08739
DATA 07961
DATA 07179
DATA 06392
DATA 05601
DATA 04807
DATA 04011
DATA 03211
DATA 02410
DATA 01607
DATA 804
DATA 0
DATA -804
DATA -1607
DATA -2410
DATA -3211
DATA -4011
DATA -4807
DATA -5601
DATA -6392
DATA -7179
DATA -7961
DATA -8739
DATA -9511

DATA -10278
DATA -11038
DATA -11792
DATA -12539
DATA -13278
DATA -14009
DATA -14732
DATA -15446
DATA -16150
DATA -16845
DATA -17530
DATA -18204
DATA -18867
DATA -19519
DATA -20159
DATA -20787
DATA -21402
DATA -22004
DATA -22594
DATA -23169
DATA -23731
DATA -24278
DATA -24811
DATA -25329
DATA -25831
DATA -26318
DATA -26789
DATA -27244
DATA -27683
DATA -28105
DATA -28510
DATA -28897
DATA -29268
DATA -29621
DATA -29955
DATA -30272
DATA -30571
DATA -30851
DATA -31113
DATA -31356
DATA -31580
DATA -31785
DATA -31970
DATA -32137
DATA -32284
DATA -32412
DATA -32520
DATA -32609
DATA -32678
DATA -32727
DATA -32757
DATA -32767
DATA -32757
DATA -32727
DATA -32678
DATA -32609
DATA -32520
DATA -32412
DATA -32284

DATA -32137
DATA -31970
DATA -31785
DATA -31580
DATA -31356
DATA -31113
DATA -30851
DATA -30571
DATA -30272
DATA -29955
DATA -29621
DATA -29268
DATA -28897
DATA -28510
DATA -28105
DATA -27683
DATA -27244
DATA -26789
DATA -26318
DATA -25831
DATA -25329
DATA -24811
DATA -24278
DATA -23731
DATA -23169
DATA -22594
DATA -22004
DATA -21402
DATA -20787
DATA -20159
DATA -19519
DATA -18867
DATA -18204
DATA -17530
DATA -16845
DATA -16150
DATA -15446
DATA -14732
DATA -14009
DATA -13278
DATA -12539
DATA -11792
DATA -11038
DATA -10278
DATA -9511
DATA -8739
DATA -7961
DATA -7179
DATA -6392
DATA -5601
DATA -4807
DATA -4011
DATA -3211
DATA -2410
DATA -1607
DATA -804
END

```
' PROGRAMA GRAPHICS.BAS
```

```
' Objetivo : Este programa es la interfase entre la tarjeta prototipo  
' para adquisición de datos y el usuario.
```

```
' Lenguaje : QuickBasic
```

```
COMMON ErrCtrl
```

```
ErrCtrl = 0
```

```
'TRON 'Activa el Debugger
```

```
' XT(259), YT(259) 'Tiempo  
' XF(259), YF(259) 'Frecuencia (dB)  
' RE(256), IM(256) 'Parte imaginaria (FFT)  
' YFL(259) 'Frecuencia (V)
```

```
DIM XT(259),YT(259),XF(259),YF(259),RE(256),IM(256),YFL(259)
```

```
OP = 0
```

```
POT = 0
```

```
ARCH$ = ""
```

```
ON ERROR GOTO ERRATA
```

```
call carga ' Carga programa TMS
```

```
call entra ' Activa TMS
```

```
call carga
```

```
call entra
```

```
SCREEN 2
```

```
LINE (1, 5) - (639, 18) ,,b
```

```
LINE (1,20) - (639,199) ,,b
```

```
LOCATE 2,4
```

```
PRINT "Archivo Muestreo Pantalla"
```

```
WHILE OP <> 3
```

```
OK = 0
```

```
CALL ESPERA(T$)
```

```
CALL VENTANA(T$,OP,OK)
```

```
SELECT CASE OP
```

```
CASE 1
```

```
LOCATE 2,40
```

```
INPUT "ARCHIVO : ",ARCH$
```

```
LOCATE 4,40
```

```
IF ARCH$ <> "" THEN
```

```
PRINT "Espere un momento ..."
```

```
CALL LEEARCH(ARCH$,XT(),YT(),XF(),YF(),YFL(),OK,TIEMPOS$,POT,TIME$)
```

```
IF OK THEN
```

```
LOCATE 2,40
```

```
PRINT SPACES$(25)
```

```
LOCATE 4,40
```

```
PRINT SPACES$(25)
```

```
LOCATE 4,40
```

```
END IF
```

```
ELSE
```

```
OK = 0
```

```
PRINT "Nombre incorrecto del archivo"
```

```
END IF
```

```

IF OK THEN
CALL MIMAEINC(XT(),"T")
CALL MIMAEINC(YT(),"T")
CALL MIMAEINC(XF(),"F")
CALL MIMAEINC(YF(),"F")
CALL MIMAEINC(YFL(),"F")
CD = (YT(257) + YT(258)) / 2
PRINT "ARCHIVO RECUPERADO"
ELSE
LOCATE 2,40
PRINT SPACES(30)
CALL LIMPIA(OP)
END IF
CALL ESPERA(T$)
LOCATE 4,40
PRINT SPACES(30)
CASE 2
LOCATE 2,40
PRINT SPACES(30)
IF XT(259) <> 0 AND YT(259) <> 0 THEN
LOCATE 2,40
INPUT "ARCHIVO : ",ARCH$
LOCATE 4,40
IF ARCH$ <> "" THEN
PRINT "Espere un momento ..."
CALL ESCARCH(ARCH$,XT(),YT(),XF(),YF(),YFL(),OK,TIEMPOS,POT,TIME
IF OK THEN
LOCATE 2,40
PRINT SPACES(25)
LOCATE 4,40
PRINT SPACES(25)
LOCATE 4,40
END IF
ELSE
OK = 0
PRINT "Nombre incorrecto del archivo"
END IF
IF OK THEN
PRINT "ARCHIVO GRABADO"
OK = -1
ELSE
LOCATE 2,40
PRINT SPACES(20)
CALL LIMPIA(OP)
END IF
ELSE
LOCATE 4,40
PRINT "NO SE HA RECUPERADO INFORMACION"
LOCATE 2,40
PRINT SPACES(20)
END IF
CALL ESPERA(T$)
LOCATE 4,40
PRINT SPACES(35)
CASE 3
LOCATE 4,40
PRINT "Presione <ENTER> para terminar"
CALL ESPERA(T$)
IF T$ = CHR$(13) THEN

```

```

SCREEN 0,0,0
LOCATE 12,15
PRINT "FIN DEL PROCESO"
ELSE
  OP = 9
END IF
CASE 4
  TIME% = 0
  LOCATE 5,13
  PRINT SPACE$(8)
  LOCATE 4,22
  PRINT "[1] 150 us [2] 250 us [3] 500 us [4] 1 ms "
  WHILE TIME% < 1 OR TIME% > 8
    LOCATE 5,22
    INPUT "[5] 2.5 ms [6] 5 ms [7] 10 ms [8] 50 ms " ; TIME%
  WEND
  LOCATE 4, 22
  PRINT SPACE$(48)
  LOCATE 5, 22
  PRINT SPACE$(48)

CALL TXD(TIME%,ESCT,ESCF,TIEMPOS,TPORD%,FPORD%)

LOCATE 4,13
PRINT SPACE$(8)
LOCATE 5,13
PRINT SPACE$(8)
LOCATE 4,40
PRINT SPACE$(25)
OK = -1
CASE 5
  IF TIME% > 0 THEN
    VERR = 255
    WHILE VERR = 255
      VD = -6
      WHILE VD < 0 OR VD > 5
        LOCATE 5, 22
        INPUT "V. DE DISPARO ( 0 <= V <= 5 ) : " ; VD
      WEND
      LOCATE 5, 22
      PRINT SPACE$(42)
      VD = INT((255/4.5)*VD) 'Escalamiento
      IF VD = 256 THEN
        VD = 255
      END IF
      DEF SEG = &HEC00
      POKE &H21, VD
      DEF SEG = &HEE00
      POKE &H21, 0
      DEF SEG
      call aten 'Muestrea y graba datos
      DEF SEG = &HEC00
      VERR = PEEK(&H22)
      IF VERR = 255 THEN
        LOCATE 2, 40
        PRINT "NO SE ALCANZO EL VOLTAJE DE DIPARO"
      END IF
      DEF SEG
    WEND
  
```

```
LOCATE 2,40
PRINT SPACE$(35)+CHR$(7)
```

Lectura de datos de memoria para OSCILOSCOPIO Y FOURIER

```
POT = 0
imag% = 0
real% = 0
LO10 = LOG(10)
```

```
LOCATE 2,40
PRINT "Muestreando ..."
```

```
FOR I = 0 TO 255
  OSCILOSCOPIO
  DEF SEG = &HEC00 'Segmento parte baja, parte alta siempre 47H
  J = 1792+I 'A partir de la localidad 700H
  YT(I+1) = PEEK(J)

  YT(I+1) = (YT(I+1)/56.6666) 'Escalamiento 0 a 4.5/255
  XT(I+1) = I*ESCT

  FOURIER
  DEF SEG = &HECA0 'Segmento parte baja
  LR = PEEK(2*I) 'a partir de la localidad A00H
  LI = PEEK(2*I+1)

  DEF SEG = &HEEA0 'Segmento parte alta
  HR = PEEK(2*I)
  HI = PEEK(2*I+1)

  DEF SEG
  addr = varptr(real%) 'Se pegan las dos partes
  poke addr,lr 'que forman cada componente
  addr = addr + 1 'para tener un solo numero real
  poke addr,hr 'y un solo imaginario
  addr = varptr(imag%)
  poke addr,li
  addr = addr + 1
  poke addr,hi
  RE(I+1) = real%
  IM(I+1) = imag%

  YF(I+1) = SQR(RE(I+1)*RE(I+1)+IM(I+1)*IM(I+1))
  YF(I+1) = YF(I+1)/3626.6666 'Escalamiento: se obtuvo empirico
  POT = POT + YF(I+1)*YF(I+1) 'Teorema de Parseval
  IF YF(I+1) = 0 THEN
    YF(I+1) = 1.0 E-6
  END IF
  YFL(I+1) = YF(I+1)
  YF(I+1) = 20 * LOG(ABS(YF(I+1)))
  YF(I+1) = YF(I+1)/ LO10 'dB
  XF(I+1) = I*ESCT
```

```
NEXT I
```

```
LOCATE 2,40
PRINT SPACE$(15)+chr$(7)
```

```

call carga
CALL TXD(TIME%,ESCT,ESCF,TIEMPO$,TPORD%,FPORD%)

CALL MIMAEINC(XT(),"T")
CALL MIMAEINC(YT(),"T")
CALL MIMAEINC(XF(),"F")
CALL MIMAEINC(YF(),"F")
CALL MIMAEINC(YPL(),"F")
CD = (YT(257) + YT(258)) / 2

LOCATE 4,13
PRINT SPACE$(8)
LOCATE 5,13
PRINT SPACE$(8)
LOCATE 4,40
PRINT SPACE$(25)
OK = -1
ELSE
LOCATE 5,22
PRINT "NO SE HA ELEGIDO ESCALA DE TIEMPO"
TECLA$ = ""
CALL ESPERA(TECLA$)
LOCATE 5,22
PRINT SPACE$(35)
END IF
CASE 6
IF XT(259)<>0 AND YT(259)<>0 THEN
LINE (1,20) - (639,199) ,B,bf
LINE (1,20) - (639,199) ,,b
TEXT$ = "T"
CALL TRAZAEJES(XT(),YT(),TEXT$,TIEMPO$,TPORD%,FPORD%,0,TIME%)
CALL FUNCTIME(XT(),YT())
ELSE
LOCATE 4,40
PRINT "NO SE HA RECUPERADO INFORMACION"
CALL ESPERA(T$)
LOCATE 4,23
PRINT SPACE$(10)
LOCATE 5,23
PRINT SPACE$(10)
LOCATE 6,23
PRINT SPACE$(10)
LOCATE 4,40
PRINT SPACE$(35)
END IF
CASE 7
IF XT(259)<>0 AND YT(259)<>0 THEN
OPF = 0
WHILE OPF <1 OR OPF >2
LOCATE 5,34
INPUT "[1] dB, [2] V ";OPF
WEND
LINE (1,20) - (639,199) ,B,bf
LINE (1,20) - (639,199) ,,b
TEXT$ = "F"
IF OPF = 1 THEN
CALL TRAZAEJES(XF(),YF(),TEXT$,TIEMPO$,TPORD%,FPORD%,OPF,TIME%)
CALL FUNCFREC(XF(),YF(),POT,OPF,CD)

```

ELSE

CALL TRAZAEJES(XF(),YFL(),TEXT\$,TIEMPO\$,TPORD\$,FPORD\$,OPF,TIME\$)

CALL FUNCFREC(XF(),YFL(),POT,OFF,CD)

END IF

ELSE

LOCATE 4,40

PRINT "NO SE HA RECUPERADO INFORMACION"

CALL ESPERA(T\$)

LOCATE 4,23

PRINT SPACES(10)

LOCATE 5,23

PRINT SPACES(10)

LOCATE 6,23

PRINT SPACES(10)

LOCATE 4,40

PRINT SPACES(35)

OK = -1

END IF

CASE 8

LOCATE 4,23

PRINT SPACES(10)

LOCATE 5,23

PRINT SPACES(10)

LOCATE 6,23

PRINT SPACES(10)

call impant

OK = -1

'Impresion de pantalla

CASE ELSE

LOCATE 4,40

PRINT SPACES(40)

END SELECT

D

ATA:

CODIGO = ERR

IF CODIGO <> 0 THEN

CALL HAYERR(CODIGO)

SELECT CASE CODIGO

CASE 52

CODIGO = 0

RESUME NEXT

CASE 53

ErrCtrl = 1

CODIGO = 0

RESUME NEXT

CASE 61

ErrCtrl = 1

CODIGO = 0

RESUME NEXT

CASE 62

ErrCtrl = 1

CODIGO = 0

RESUME NEXT

CASE 64

ErrCtrl = 1

CODIGO = 0

RESUME NEXT

CASE ELSE

CODIGO = 0

```
RESUME
END SELECT
END IF
```

```
SCREEN 0,0,0
LOCATE 12,15
PRINT "FIN DEL PROCESO"
```

```
END
```

```
SUB LEEARCH(ARCH$,XT(1),YT(1),XF(1),YF(1),YFL(1),OK,TIEMPOS,POT,TIME$) STAT
```

```
SHARED ErrCtrl
```

```
OPEN ARCH$ FOR INPUT AS #1
IF ErrCtrl = 0 THEN
  INPUT #1, TIEMPOS$, POT, TIME$
  FOR I = 1 TO 256
    INPUT #1, XT(I), YT(I)
  NEXT I
  FOR I = 1 TO 256
    INPUT #1, XF(I), YF(I), YFL(I)
  NEXT I
  CLOSE #1
  OK = -1
ELSE
  ErrCtrl = 0
  OK = 0
END IF
```

```
END SUB
```

```
SUB ESCARCH(ARCH$,XT(1),YT(1),XF(1),YF(1),YFL(1),OK,TIEMPOS,POT,TIME$) STAT
```

```
OPEN ARCH$ FOR OUTPUT AS #1
CODE = ERR
IF CODE = 0 THEN
  WRITE #1, TIEMPOS$, POT, TIME$
  FOR I = 1 TO 256
    WRITE #1, XT(I), YT(I)
  NEXT I
  FOR I = 1 TO 256
    WRITE #1, XF(I), YF(I), YFL(I)
  NEXT I
  CLOSE #1
  OK = -1
ELSE
  OK = 0
END IF
```

```
END SUB
```

```
SUB MIMAEINC(V(1),ENCAB$) STATIC
```

```
Vmax = -65535
Vmin = 65535
```

```
IF ENCAB$ = "T" THEN
  LIM = 1
```

```

ELSE
  LIM = 2
END IF

FOR I = LIM TO 256
  IF V(I) > Vmax THEN
    Vmax = V(I)
  END IF
  IF V(I) < Vmin THEN
    Vmin = V(I)
  END IF
NEXT I
Inc = (Vmax - Vmin)/256
V(257) = Vmin
V(258) = Vmax
V(259) = Inc

```

D SUB

B TRAZAEJES(X(1),Y(1),ENCAB\$,TIEMPOS\$,TPORD\$,FPORD\$,OPF,TIME\$) STATIC

```

IF ENCAB$ = "T" THEN
  INCX = TPORD$/10
  XMAX = TPORD$
  INCY = (Y(258)-Y(257))/4
ELSE
  INCX = FPORD$/10
  XMAX = FPORD$
  IF OPF = 1 THEN
    'Nivel piso de ruido = 65 dB = 20 log Vmax/Vruido
    NPR = Y(258) - 65
    IF Y(257) > NPR THEN
      NPR = Y(257)
    END IF
  ELSE
    NPR = Y(257)
  END IF
  INCY = (Y(258)-NPR)/4
END IF

```

DIV = 1

```

TRAZO EJE Y
IF ENCAB$ = "T" THEN
  LINE (80,40)-(80,170)
  LOCATE 6,11
  PRINT "v"
  LIM1 = Y(257)
ELSE
  LOCATE 6,9
  IF OPF = 1 THEN
    PRINT "dB"
    LIM1 = NPR
  ELSE
    PRINT "v"
    LIM1 = Y(257)
  END IF
END IF

```

END IF

```
FOR I = 1 TO 4
  LINE (76,30*I+20)-(588,30*I+20),,,&HFF00
NEXT I
```

```
REN = 7
J = 1
FOR I = Y(258) TO LIM1 STEP INCY*-1
  LOCATE REN, 2
  PRINT USING "###.##" ; I
  IF J = 2 THEN
    REN = REN + 3
  ELSE
    REN = REN + 4
  END IF
  J = J + 1
NEXT I
```

TRAZO EJE X

```
LINE (80,170)-(588,170) 'Eje y Marcas
IF ENCAB$ = "T" THEN
  dx = 79
ELSE
  dx = 81
END IF
FOR I = 1 TO 10
  LINE (50*I+dx,170)-(50*I+dx,172)
NEXT I
```

```
COL = 8 'Valores Marcas
J = 1
FOR I = 0 TO COL
  LOCATE 23, COL
  IF ENCAB$ = "T" THEN
    IF TIME$ = 5 THEN
      PRINT USING "###.##" ; INCX*I
    ELSE
      PRINT USING "#####" ; INCX*I
    END IF
  ELSE
    IF TIME$ > 4 THEN
      PRINT USING "###.##" ; INCX*I
    ELSE
      PRINT USING "#####" ; INCX*I
    END IF
  END IF
  IF J = 5 OR J = 9 THEN
    COL = COL + 1
  END IF
  COL = COL + 6
  J = J + 1
NEXT I
```

```
IF ENCAB$ = "T" THEN
  LOCATE 22,COL-1
  PRINT "t[s]"
ELSE
  LOCATE 20,COL
  PRINT "f[Hz]"
```

```

END IF

IF ENCAB$ = "F" THEN
  LOCATE 22,COL
  PRINT "E+3"
ELSE
  LOCATE 23,COL
  IF TIEMPO$ = "3" THEN
    PRINT "E-3"
  ELSE
    PRINT "E-6"
  END IF
END IF

END SUB

END SUB

SUB ESPERA(T$) STATIC
  T$ = ""
  WHILE T$ = ""
    T$ = INKEY$
  WEND

  IF ASC(T$) > 96 AND ASC(T$) < 123 THEN
    T$ = CHR$(ASC(T$) - 32)
  END IF
END SUB

SUB FUNCTIME(XT(1),YT(1)) STATIC
  DIM ABC(256), ORD(256)

  ' ESCALAMIENTOS Y DESPLAZAMIENTOS
  ESCX = 490/(XT(258) - XT(257))
  ESCY = 120/(YT(258) - YT(257))
  IF XT(257) < 0 THEN
    DESPX = -XT(257)*ESCX+80
  ELSE
    DESPX = 80
  END IF
  PROM = (YT(258) + YT(257))/2
  DESPY = 110 + PROM*ESCY

  FOR I = 1 TO 256
    ABC(I) = XT(I)*ESCX + DESPX
    'ORD(I) = YT(I)*ESCY + DESPY
    ORD(I) = (-YT(I)-PROM) * ESCY + DESPY
    IF I > 1 AND I < 256 THEN
      LINE (ABC(I),ORD(I))-ABC(I-1),ORD(I-1))
    END IF
  NEXT I

END SUB

END SUB

SUB FUNCFREC(XF(1),YF(1),P,OPF,CD) STATIC
  ' ESCALAMIENTOS Y DESPLAZAMIENTOS
  ESCX = 490/(XF(258) - XF(257))

  IF OPF = 1 THEN

```

```

'Nivel piso de ruido = 65 dB = 20 log Vmax/Vruido
NPR = YF(258) - 65
IF YF(257) > NPR THEN
  NPR = YF(257)
END IF
DESPY = 170
ELSE
  NPR = YF(257)
  DESPY = 110
END IF
ESCY = 120/(YF(258) - NPR)
DESPX = 82

IF OFF = 1 THEN
  ABC = XF(2)*ESCX + DESPX
  IF YF(2) < NPR THEN
    ORD = 170
  ELSE
    ORD = -(YF(2)-NPR)*ESCY + DESPY
  END IF
  PSET (ABC,ORD)

  FOR I = 2 TO 256 STEP 1
    ABC = XF(I)*ESCX + DESPX
    IF YF(I) < NPR THEN
      ORD = 170
    ELSE
      ORD = -(YF(I)-NPR)*ESCY + DESPY
    END IF
    LINE -(ABC,ORD)
  NEXT I
ELSE
  FOR I = 2 TO 256 STEP 1
    ABC = XF(I)*ESCX + DESPX
    ORD = 274-{YF(I)*ESCY + DESPY}      'Escala lineal
    LINE (ABC,ORD)-(ABC,170)
  NEXT I
END IF

IF P > 0 THEN
  LOCATE 4, 35
  PRINT "Potencia = ";
  PRINT USING "####.##";P;
  PRINT "[W]"
  LOCATE 4, 60
  PRINT "C.D. = ";
  PRINT USING "##.##";CD;
  PRINT "[V]"
END IF

END SUB

SUB TXD(TIME%,ESCT,ESCF,TIEMPO$,TPORD$,FPORD$) STATIC

SELECT CASE TIME%
CASE 1
CASE ELSE
DEF SEG = &HEC00
POKE &H67 , &H00

```

```

POKE &H68 , &H89
POKE &H46 , &H00
POKE &H47 , &H89
DEF SEG = &HEE00
POKE &H67 , &HF8
POKE &H68 , &H01
POKE &H46 , &HF8
POKE &H47 , &H01
DEF SEG = &HEC00
END SELECT
DEF SEG
SELECT CASE TIME#
CASE 1
DEF SEG = &HEC00
POKE &H67 , &H80
POKE &H68 , &H80
POKE &H46 , &H80
POKE &H47 , &H80
DEF SEG = &HEE00
POKE &H67 , &H7F
POKE &H68 , &H7F
POKE &H46 , &H7F
POKE &H47 , &H7F
ESCT = 6.1475E-6 '(150 microseg/div) / (25.4 pix/div)
ESCF = 666.6666
TIEMPO$ = "6"
TPORD# = 1500 'us
FPORD# = 204.8 'kHz
CASE 2
DEF SEG = &HEC00
POKE &H20 , &H04
DEF SEG = &HEE00
POKE &H20 , &H00
ESCT = 9.8425E-6
ESCF = 400
TIEMPO$ = "6"
TPORD# = 2500
FPORD# = 102.4
CASE 3
DEF SEG = &HEC00
POKE &H20 , &H0F
DEF SEG = &HEE00
POKE &H20 , &H00
ESCT = 19.685E-6
ESCF = 200
TIEMPO$ = "6"
TPORD# = 5000
FPORD# = 51.2
CASE 4
DEF SEG = &HEC00
POKE &H20 , &H27
DEF SEG = &HEE00
POKE &H20 , &H00
ESCT = 39.37E-6
ESCF = 100
TIEMPO$ = "3"
TPORD# = 10 'ms
FPORD# = 25.6

```

```

CASE 5
  DEF SEG = &HEC00
  POKE &H20 , &H70
  DEF SEG = &HEE00
  POKE &H20 , &H00
  ESCT = 98.425E-6
  ESCF = 40
  TIEMPO$ = "3"
  TPORD% = 25
  FPORD% = 10.24
CASE 6
  DEF SEG = &HEC00
  POKE &H20 , &HE5
  DEF SEG = &HEE00
  POKE &H20 , &H00
  ESCT = 196.85E-6
  ESCF = 20
  TIEMPO$ = "3"
  TPORD% = 50
  FPORD% = 5.12
CASE 7
  DEF SEG = &HEC00
  POKE &H20 , &HDA
  DEF SEG = &HEE00
  POKE &H20 , &H01
  ESCT = 393.7E-6
  ESCF = 10
  TIEMPO$ = "3"
  TPORD% = 100
  FPORD% = 2.56
CASE 8
  DEF SEG = &HEC00
  POKE &H20 , &H6E
  DEF SEG = &HEE00
  POKE &H20 , &H09
  ESCT = 196.85E-5
  ESCF = 5
  TIEMPO$ = "3"
  TPORD% = 500
  FPORD% = 1.28
CASE ELSE
END SELECT
DEF SEG

```

END SUB

SUB VENTANA(LETRA\$,OP,OK) STATIC

```

CALL LIMPIA(OP)
OP1 = 0
OP$ = LETRAS
AR$ = CHR$(16)
WHILE OP1 = 0
  SELECT CASE LETRAS$
  CASE "A"
    LOCATE 4,3
    PRINT " RECUPERAR"+SPACES$(40)
    LOCATE 5,3
    PRINT " GRABAR " +SPACES$(40)

```

```
LOCATE 6,3
PRINT " FIN "
LOCATE 6,13
PRINT SPACES(20)
CALL ESPERA(OP$)
```

```
SELECT CASE OP$
CASE "R"
  OP1 = 1
  LOCATE 4,3
  PRINT AR$
CASE "G"
  LOCATE 5,3
  OP1 = 2
  PRINT AR$
CASE "F"
  LOCATE 6,3
  OP1 = 3
  PRINT AR$
CASE ELSE
  CALL LIMPIA(OP1)
  LOCATE 4,3
  LETRA$ = OP$
  OP$ = ""
END SELECT
```

```
CASE "M"
  LOCATE 4,3
  PRINT SPACES(10)+"T/DIV "+SPACES(15)
  LOCATE 5,3
  PRINT SPACES(10)+"INICIAR"+SPACES(15)
  LOCATE 6,3
  PRINT SPACES(30)
  CALL ESPERA(OP$)
  SELECT CASE OP$
  CASE "T"
    LOCATE 4,12
    PRINT AR$
    OP1 = 4
  CASE "I"
    LOCATE 5,12
    PRINT AR$
    OP1 = 5
  CASE ELSE
    CALL LIMPIA(OP1)
    LETRA$ = OP$
    OP$ = ""
  END SELECT
```

```
CASE "P"
  LOCATE 4,3
  PRINT SPACES(20)+"TIEMPO"
  LOCATE 5,3
  PRINT SPACES(20)+"FRECUENCIA"+SPACES(6)
  LOCATE 6,3
  PRINT SPACES(20)+"IMPRIMIR "+SPACES(6)
  CALL ESPERA(OP$)
```

```
SELECT CASE OP$
CASE "T"
  LOCATE 4,22
  PRINT AR$
  OP1 = 6
CASE "P"
  LOCATE 5,22
  PRINT AR$
  OP1 = 7
CASE "I"
  LOCATE 6,22
  PRINT AR$
  OP1 = 8
CASE ELSE
  CALL LIMPIA(OP1)
  LETRA$ = OP$
  OP$ = ""
END SELECT

CASE ELSE
  CALL LIMPIA(OP1)
  CALL ESPERA(LETRA$)
END SELECT
WEND
OP = OP1
```

END SUB

SUB LIMPIA(OP) STATIC

```
SELECT CASE OP
CASE 1
  LOCATE 4,3
  PRINT SPACE$(10)
  LOCATE 5,3
  PRINT SPACE$(10)
  LOCATE 6,3
  PRINT SPACE$(10)
CASE 2
  LOCATE 4,3
  PRINT SPACE$(10)
  LOCATE 5,3
  PRINT SPACE$(10)
  LOCATE 6,3
  PRINT SPACE$(10)
CASE 3
  LOCATE 4,3
  PRINT SPACE$(10)
  LOCATE 5,3
  PRINT SPACE$(10)
  LOCATE 6,3
  PRINT SPACE$(10)
CASE 4
  LOCATE 4,11
  PRINT SPACE$(12)
  LOCATE 5,11
  PRINT SPACE$(12)
  LOCATE 6,11
  PRINT SPACE$(12)
```

```
CASE 5
  LOCATE 4,11
  PRINT SPACE$(12)
  LOCATE 5,11
  PRINT SPACE$(12)
  LOCATE 6,11
  PRINT SPACE$(12)
```

```
CASE 6
  LOCATE 4,21
  PRINT SPACE$(12)
  LOCATE 5,21
  PRINT SPACE$(12)
  LOCATE 6,21
  PRINT SPACE$(12)
```

```
CASE 7
  LOCATE 4,21
  PRINT SPACE$(12)
  LOCATE 5,21
  PRINT SPACE$(12)
  LOCATE 6,21
  PRINT SPACE$(12)
```

```
CASE 8
  LOCATE 4,21
  PRINT SPACE$(12)
  LOCATE 5,21
  PRINT SPACE$(12)
  LOCATE 6,21
  PRINT SPACE$(12)
```

```
CASE ELSE
END SELECT
```

```
END SUB
```

```
SUB HAYERR(CODE) STATIC
```

```
A$ = ""
LOCATE 4,40
SELECT CASE CODE
  CASE 52
    PRINT "Nombre incorrecto del archivo"
  CASE 53
    PRINT "Archivo no encontrado"
  CASE 61
    PRINT "El disco está lleno"
  CASE 64
    PRINT "Nombre incorrecto del archivo"
  CASE 71
    PRINT "La unidad de disco no está lista"
  CASE 72
    PRINT "Existe error en el disco"
  CASE ELSE
    PRINT "Error : ";CODE
END SELECT
CALL ESPERA(A$)
LOCATE 4,40
PRINT SPACE$(30)
```

```
END SUB
```

APENDICE D

DESCRIPCION DE ALGUNAS FUNCIONES DE MS-DOS UTILES PARA LA ADQUISICION DE DATOS, ALMACENAMIENTO, PROCESAMIENTO Y DESPLIEGUE.

El sistema operativo MSDOS permite el uso de procedimientos específicos para realizar diversas acciones de control de los componentes de la PC, el acceso a estos procedimientos de MSDOS se hace, al igual que al acceso a las funciones del BIOS por la instrucción del μP INT, el sistema operativo MSDOS se reserva las interrupciones 20h a 3Fh para realizar sus procedimientos, los vectores de estas interrupciones se localizan en las localidades 80h a FFh de la memoria de la PC.

INT 21H.- LLLAMADO A FUNCIONES.

La solicitud del uso de una función se hace por medio de colocar el número de la función requerida en el registro AH del microprocesador, y la información adicional requerida en los diferentes registros, dependiendo de la función de que se trate, a continuación se realiza una interrupción (INT 21).

Esta interrupción requiere un stack suficientemente grande para almacenar el sistema de interrupciones. Se recomienda que sea al menos de 80H bytes mas lo necesario para el usuario.

Cada función puede regresar alguno de los siguientes tipos de error (esto se indica en la descripciones que se dan a continuación):

- 1 Número inválido de función.
- 2 Archivo no encontrado.
- 3 Ruta no encontrada.
- 4 Muchos archivos abiertos (no hay manejadores).
- 5 Acceso no permitido.
- 6 Manejador inválido.
- 7 Bloques de control de memoria destruidos.
- 8 Memoria insuficiente.
- 9 Dirección de bloque de memoria inválida.
- 10 Ambiente inválido.
- 11 Formato inválido.
- 12 Código de acceso inválido.
- 13 Dato inválido.
- 15 Especificación de drive inválida.
- 16 Intento de remover el directorio actual.
- 17 No es el mismo dispositivo.
- 18 No hay mas archivos.

Las funciones que presentan interés para el desarrollo de este proyecto son:

A) Salida al video (*VIDEO OUTPUT*) 02h

El caracter en el registro DL es puesto en video. Si el caracter es un <Ctrl C> se realiza una interrupción 23H.

B) Poner vector (*SET VECTOR*) 25h

Esta función se utiliza para colocar un vector de interrupción en la tabla de vectores de interrupción.

DS:DX contiene el vector de dirección de una subrutina de interrupción. (Este vector se coloca en la tabla de vectores de interrupción).

AL contiene el número de interrupción. (Llamada por la instrucción INT).

C) Abrir un archivo (*OPEN A FILE*) 3Dh

DS:DX contiene la dirección de un "string" ASCII con el drive y la ruta y nombre del archivo. AL contiene el código de acceso. En AX se regresa un código de error o un manejador de archivo (handle). AX puede tomar los valores: 0 abierto para leer, 1 = abierto para escribir, 2 = abierto para leer o escribir. Los errores posibles regresados en AX son: 2, 4, 5 y 12.

D) Cerrar un archivo (*CLOSE A FILE HANDLE*) 3Eh

BX contiene el manejador que regresó la apertura del archivo. El error que se puede presentar en AX es: 6.

E) Leer de un archivo o dispositivo (*READ FROM A FILE OR DEVICE*) 3Fh

Tiene como fin la lectura de los bytes de un archivo o de un dispositivo.

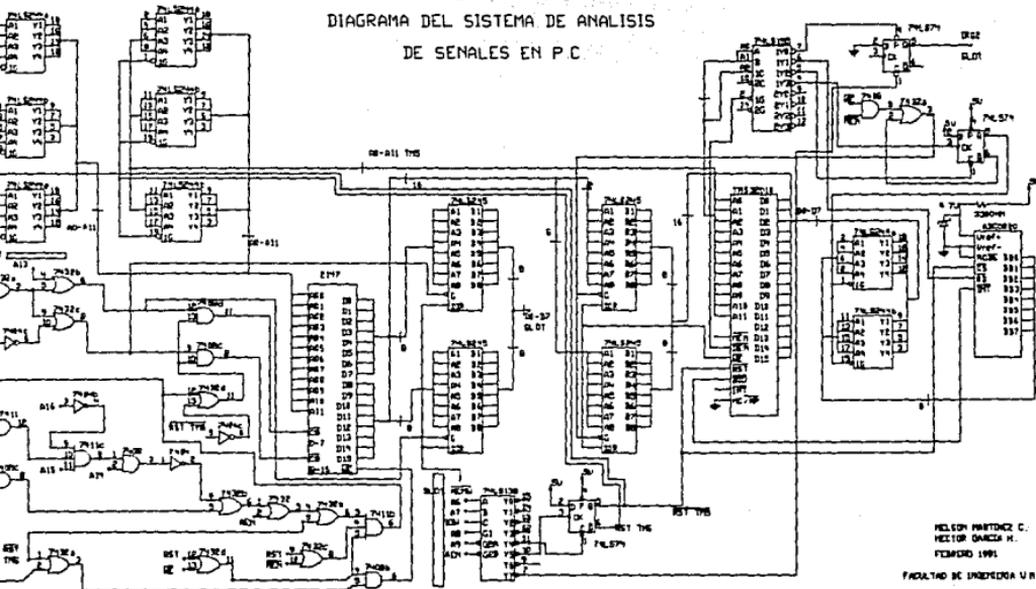
Inicialmente BX contiene el manejador (handle) de un archivo. CX contiene el número de bytes que se leerán. DS:DX contiene la dirección del buffer

AX regresa el número de bytes leídos. Si AX = 0 se intentó leer del fin de archivo.

Los posibles errores que se regresan en AH son: 5 y 6.

APENDICE E
DIAGRAMA DE LA TARJETA
HOJAS DE DATOS

DIAGRAMA DEL SISTEMA DE ANALISIS
DE SEÑALES EN P.C.



HELSON MARTINEZ C.
HECTOR GARCIA H.
FEBRERO 1991
FACULTAD DE INGENIERIA U

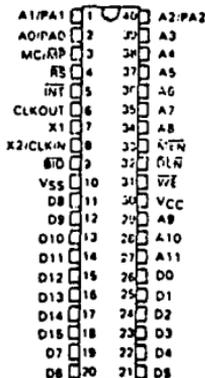
- 200-ns Instruction Cycle
- 288-Byte On-Chip Data RAM
- ROMless Version - TMS32010
- 3K-Byte On-Chip Program ROM - TMS32010
- External Memory Expansion to A Total of 8K Bytes at Full Speed
- 16-Bit Instruction/Data Word
- 32-Bit ALU/Accumulator
- 16 x 16-Bit Multiply in 200 ns
- 0 to 15-Bit Barrel Shifter
- Eight Input and Eight Output Channels
- 16-bit Bidirectional Data Bus with 40-Megabits-per-Second Transfer Rate
- Interrupt with Full Context Save
- Signed Two's Complement Fixed-Point Arithmetic
- NMOS Technology
- Single 5-V Supply

description

The TMS32010 is the first member of the new TMS320 digital signal processing family, designed to support a wide range of high-speed or numeric-intensive applications. This 16/32-bit single-chip microcomputer combines the flexibility of a high-speed controller with the numerical capability of an array processor, thereby offering an inexpensive alternative to multichip bit-slice processors. The TMS320 family contains the first MOS microcomputers capable of executing five million instructions per second. This high throughput is the result of the comprehensive, efficient, and easily programmed instruction set and of the highly pipelined architecture. Special instructions have been incorporated to speed the execution of digital signal processing (DSP) algorithms.

The TMS320 family's unique versatility and power give the design engineer a new approach to a variety of complicated applications. In addition, these microcomputers are capable of providing the multiple functions often required for a single application. For example, the TMS320 family can enable an industrial robot to synthesize and recognize speech, sense objects with radar or optical intelligence, and perform mechanical operations through digital servo loop computations.

TMS32010 ... JDL PACKAGE
(TOP VIEW)



PIN NOMENCLATURE

SIGNATURE	I/O	DEFINITION
A11-AD; PA2-PA0	OUT	External address bus. I/O port address multiplexed over PA2-PA0.
DT0	IN	External polling input for bit test and pump operations.
CLKOUT	OUT	System clock output. If crystal/CLKIN frequency.
D15-D0	I/O	16 bit data bus.
DTN	OUT	Data enable indicates the processor accepting input data on D15-D0.
INT	IN	Interrupt.
MC/MP	IN	Memory mode select pin. High selects microcomputer mode. Low selects microprocessor mode.
MEN	OUT	Memory enable indicates that D15-D0 will accept external memory instruction.
RS	IN	Reset used to initialize the device.
VCC	IN	Power.
VSS	IN	Ground.
WE	OUT	Write enable indicates valid data on D15-D0.
X1	IN	Crystal input.
X2/CLKIN	IN	Crystal input or external clock input.

ADVANCE INFORMATION

This document contains information on a new product. Specifications are subject to change without notice.

TEXAS
INSTRUMENTS

Copyright © 1983 by Texas Instruments Incorporated

SPAS007

ADC0820 8-Bit High Speed μ P Compatible A/D Converter with Track/Hold Function

General Description

By using a half-flash conversion technique, the 8-bit ADC0820 CMOS A/D offers a 1.5 μ s conversion time and dissipates only 75 mW of power. The half-flash technique consists of 32 comparators, a most significant 4-bit ADC and a least significant 4-bit ADC.

The input to the ADC0820 is tracked and held by the input sampling circuitry eliminating the need for an external sample-and-hold for signals moving at less than 100 mV/ μ s.

For ease of interface to microprocessors, the ADC0820 has been designed to appear as a memory location or I/O port without the need for external interfacing logic.

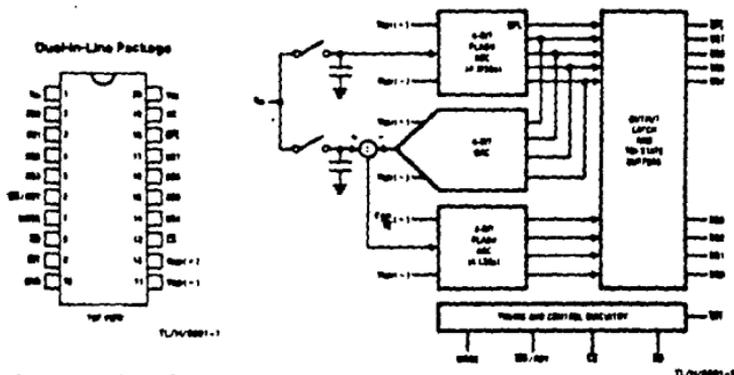
Key Specifications

- 8 Resolution 8 Bits
- Conversion Time \rightarrow \geq 2.5 μ s Max (RD Mode)
1.5 μ s Max (WR-RD Mode)
- Input signals with slew rate of 100 mV/ μ s converted without external sample-and-hold to 8 bits
- Low Power 75 mW Max
- Total Unquoted Error \pm 1/2 LSB and \pm 1 LSB

Features

- Built-in track-and-hold function
- No missing codes
- No external clocking
- Single supply—5 V_{CC}
- Easy interface to all microprocessors, or operate stand-alone
- Latched TRI-STATE® output
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Operates ratiometrically or with any reference value equal to or less than V_{CC}
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Overflow output available for cascading
- 0.3" standard width 20-pin DIP

Connection and Functional Diagrams



Absolute Maximum Ratings (Notes 1 & 2)

Supply Voltage (V_{CC})	10V
Logic Control Inputs	-0.2V to $V_{CC} + 0.2V$
Voltage at Other Inputs and Outputs	-0.2V to $V_{CC} + 0.2V$
Storage Temperature Range	-65°C to +150°C
Package Dissipation at $T_A = 25^\circ\text{C}$	875 mW
Lead Temp. (Soldering, 10 seconds)	300°C

Operating Conditions (Notes 1 & 2)

Temperature Range	$T_{MIN} \leq T_A \leq T_{MAX}$
ADC08208D, ADC0820CD	-65°C $\leq T_A \leq$ +125°C
ADC08208CQ, ADC0820CCD	-40°C $\leq T_A \leq$ +85°C
ADC08208CH, ADC0820CCH	0°C $\leq T_A \leq$ 70°C
V_{CC} Range	4.5V to 5V

Converter Characteristics

The following specifications apply for RD mode (pin 7 = 0), $V_{CC} = 5V$, $V_{REF}(+) = 5V$, and $V_{REF}(-) = GND$ unless otherwise specified. Boldface limits apply from T_{MIN} to T_{MAX} ; all other limits $T_A = T_I = 25^\circ\text{C}$.

Parameter	Conditions	ADC08208D, ADC0820CD ADC08208CQ, ADC0820CCD			ADC08208CH, ADC0820CCH		
		Typ (Note 8)	Tested Limit (Note 7)	Design Limit (Note 8)	Typ (Note 8)	Tested Limit (Note 7)	Design Limit (Note 8)
Resolution			8		8	8	
Total Unadjusted Error (Note 3)	ADC08208D, BCD ADC08208CH ADC0820CD, CQD ADC0820CCH		$\pm 1/2$ ± 1		$\pm 1/2$ ± 1	$\pm 1/2$ ± 1	
Minimum Reference Resistance		2.3	1.25		2.3	1.4	
Maximum Reference Resistance		2.3	8		2.3	8	
Maximum $V_{REF}(+)$ Input Voltage			V_{CC}		V_{CC}	V_{CC}	
Minimum $V_{REF}(-)$ Input Voltage			GND		GND	GND	
Minimum $V_{REF}(+)$ Input Voltage			$V_{REF}(-)$		$V_{REF}(-)$	$V_{REF}(-)$	
Maximum $V_{REF}(-)$ Input Voltage			$V_{REF}(+)$		$V_{REF}(+)$	$V_{REF}(+)$	
Maximum V_{IN} Input Voltage			$V_{CC} + 0.1$		$V_{CC} + 0.1$	$V_{CC} + 0.1$	
Minimum V_{IN} Input Voltage			GND - 0.1		GND - 0.1	GND - 0.1	
Maximum Analog Input Leakage Current	$CS = V_{CC}$ $V_{IN} = V_{CC}$ $V_{IN} = GND$		8 -3		0.3 -0.3	8 -3	
Power Supply Sensitivity	$V_{CC} = 5V \pm 5\%$	$\pm 1/2$	$\pm 1/2$		$\pm 1/2$	$\pm 1/2$	

DC Electrical Characteristics The following specifications apply for $V_{CC} = 5V$, unless otherwise specified. Testload limits apply from T_{MIN} to T_{MAX} ; all other limits $T_A = T_J = 25^\circ C$.

Parameter	Conditions	ADC0820B0, ADC0820CD ADC0820BCD, ADC0820CCD			ADC0820B0C, ADC0820CCN			Limit Units	
		Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)	Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)		
$V_{IN(1)}$, Logical "1" Input Voltage	$V_{CC} = 5.25V$ CS, WR, RD		2.8			2.0	2.8	V	
		Mode		3.5			3.5	3.5	V
$V_{IN(0)}$, Logical "0" Input Voltage	$V_{CC} = 4.75V$ CS, WR, RD		0.8			0.8	0.8	V	
		Mode		1.8			1.5	1.8	V
$I_{IN(1)}$, Logical "1" Input Current	$V_{IN(1)} = 5V$, CS, RD $V_{IN(1)} = 5V$, WR $V_{IN(1)} = 5V$, Mode		0.006	1		0.006	1	μA	
			0.1	3		0.1	3	μA	
			50	200		50	200	μA	
$I_{IN(0)}$, Logical "0" Input Current	$V_{IN(0)} = 0V$, CS, RD, WR, Mode	-0.006	-1		-0.006		-1	μA	
$V_{OUT(1)}$, Logical "1" Output Voltage	$V_{CC} = 4.75V$, $I_{OUT} = -380 \mu A$; DB0-DB7, OFL, INT $V_{CC} = 4.75V$, $I_{OUT} = -10 \mu A$; DB0-DB7, OFL, INT		2.4			2.8	2.4	V	
				4.8			4.8	4.8	V
$V_{OUT(0)}$, Logical "0" Output Voltage	$V_{CC} = 4.75V$, $I_{OUT} = 1.8 mA$; DB0-DB7, OFL, INT, RDY		0.4			0.34	0.4	V	
I_{OUT} , TRI-STATE Output Current	$V_{OUT} = 5V$; DB0-DB7, RDY $V_{OUT} = 0V$; DB0-DB7, RDY		0.1	3		0.1	3	μA	
			-0.1	-3		-0.1	-3	-3	μA
I_{SOURCE} , Output Source Current	$V_{OUT} = 0V$; DB0-DB7, OFL INT		-12	-6		-12	-7.2	-6	mA
			-8	-4.8		-8	-5.3	-4.8	mA
I_{SINK} , Output Sink Current	$V_{OUT} = 5V$; DB0-DB7, OFL INT, RDY	14	7		14	8.4	7	mA	
I_{CC} , Supply Current	CS = WR = RD = 0	7.8	18		7.5	13	18	mA	

AC Electrical Characteristics The following specifications apply for $V_{CC} = 5V$, $t_r = t_f = 20 ns$, $V_{IN(1)}(+) = 5V$, $V_{IN(1)}(-) = 0V$ and $T_A = 25^\circ C$ unless otherwise specified.

Parameter	Conditions	Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)	Units
t_{CONV} , Conversion Time for RD Mode	Pin 7 = 0, (Figure 2)	1.8		2.8	μs
t_{ACCESS} , Access Time (Delay from Falling Edge of RD to Output Valid)	Pin 7 = 0, (Figure 2)	$t_{CONV} + 20$		$t_{CONV} + 50$	ns
$t_{CONV, RD}$, Conversion Time for WR-RD Mode	Pin 7 = V_{CC} ; $t_{WR} = 800 ns$, $t_{RD} = 800 ns$; (Figures 3a and 3b)			1.52	μs
t_{WRITE} , Write Time	Min	Pin 7 = V_{CC} ; (Figures 3a and 3b)		600	ns
	Max	(Note 4) See Graph	50		μs
t_{READ} , Read Time	Min	Pin 7 = V_{CC} ; (Figures 3a and 3b) (Note 4) See Graph		600	ns
t_{ACC1} , Access Time (Delay from Falling Edge of RD to Output Valid)	Pin 7 = V_{CC} ; $t_{RD} < t_r$; (Figure 3a) $C_L = 15 pF$	180		280	ns
	$C_L = 100 pF$	210		320	ns
t_{ACC2} , Access Time (Delay from Falling Edge of RD to Output Valid)	Pin 7 = V_{CC} ; $t_{RD} > t_r$; (Figure 3b)	70		120	ns
	$C_L = 100 pF$	90		150	ns

AC Electrical Characteristics (Continued)

The following specifications apply for $V_{CC} = 5V$, $t_c = t_r = 20 ns$, $V_{REF(+)} = 5V$, $V_{REF(-)} = 0V$ and $T_A = 25^\circ C$ unless otherwise specified.

Parameter	Conditions	Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)	Units
t_c : Internal Conversion Time	Pin 7 = V_{CC} ; (Figures 3b and 4) $C_L = 50 pF$	800		1300	ns
t_{1M} : I_{CS} TRI-STATE Control (Delay from Rising Edge of RD to M=2 State)	$R_L = 1k$, $C_L = 10 pF$	100		200	ns
t_{2M} : Delay from Rising Edge of WR to Falling Edge of INT	Pin 7 = V_{CC} , $C_L = 50 pF$ $t_{RD} > t_c$; (Figure 3b) $t_{RD} < t_c$; (Figure 3a)	$t_{RD} + 200$		t_c $t_{RD} + 280$	ns ns
t_{2M} : Delay from Rising Edge of RD to Rising Edge of INT	(Figures 2, 3a and 3b) $C_L = 50 pF$	128		228	ns
t_{2M} : Delay from Rising Edge of WR to Rising Edge of INT	(Figure 4), $C_L = 50 pF$	178		270	ns
t_{RD} : Delay from CS to RDY	(Figure 2), $C_L = 50 pF$, Pin 7 = 0	50		100	ns
t_{D} : Delay from INT to Output Valid	(Figure 4)	20		80	ns
t_{M} : Delay from RD to INT	Pin 7 = V_{CC} , $t_{RD} < t_c$ (Figure 3a)	200		280	ns
t_{M} : Delay from End of Conversion to Next Conversion	(Figures 2, 3a, 3b and 4) (Note 4) See Graph			800	ns
Slew Rate, Tracking		0.1			V/ μs
C_{IN} : Analog Input Capacitance		45			pF
C_{OUT} : Logic Output Capacitance		5			pF
C_{IN} : Logic Input Capacitance		5			pF

Note 1: Absolute Maximum Ratings are those values beyond which the life of the device may be impaired.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: Total unqualified error includes offset, full scale, and linearity errors.

Note 4: Accuracy may degrade if t_{M} or t_{M} is shorter than the minimum value specified. See Accuracy vs t_{M} and Accuracy vs t_{M} graphs.

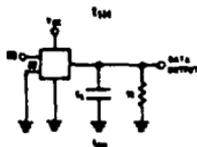
Note 5: The voltage at these pins should never go higher than V_{CC} nor lower than GND.

Note 6: Typicals are at $25^\circ C$ and represent most likely parametric norm.

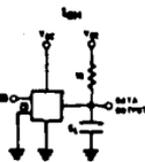
Note 7: Guaranteed and 100% production tested.

Note 8: Guaranteed, but not 100% production tested. These limits are not used to calculate outgoing quality levels.

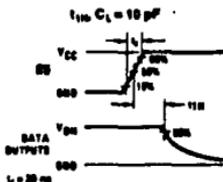
TRI-STATE Test Circuits and Waveforms



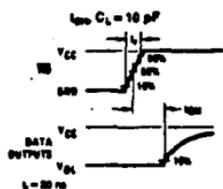
TLV5801-3



TLV5801-2



TLV5801-3



TLV5801-2

Timing Diagrams

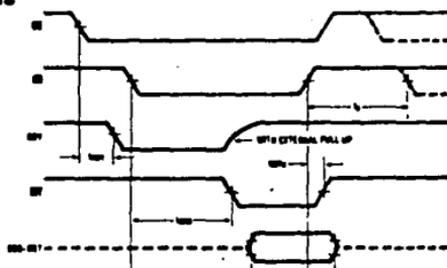
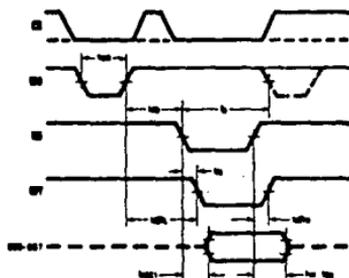
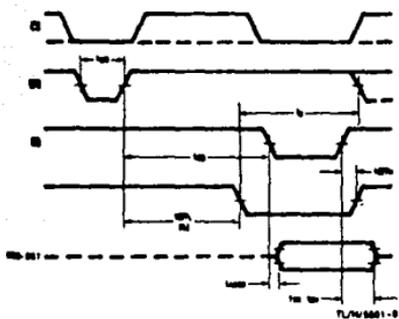


FIGURE 2. RD Mode (Pin 7 is Low)

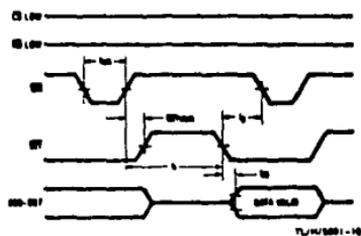
TLV5501-7

FIGURE 3a. WR-RD Mode (Pin 7 is High and $t_{p0} < t_0$)

TLV5501-8

FIGURE 3b. WR-RD Mode (Pin 7 is High and $t_{p0} > t_0$)

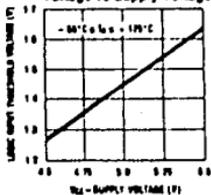
TLV5501-9

FIGURE 4. WR-RD Mode (Pin 7 is High)
Stand-Alone Operation

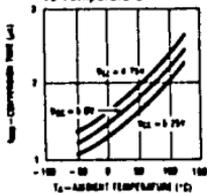
TLV5501-10

Typical Performance Characteristics

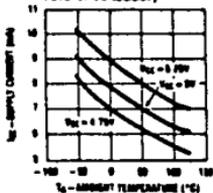
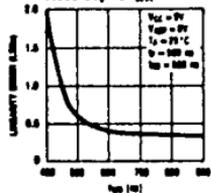
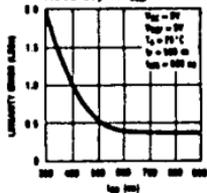
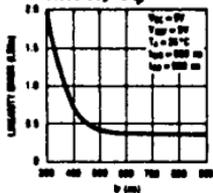
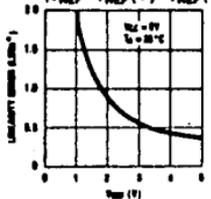
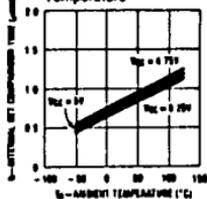
Logic Input Threshold Voltage vs Supply Voltage



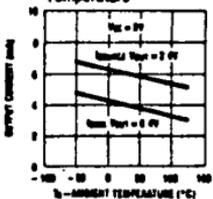
Conversion Time (RD Mode) vs Temperature



Power Supply Current vs Temperature (not including reference ladder)

Accuracy vs t_{CON} Accuracy vs t_{AP} Accuracy vs t_p Accuracy vs V_{REF} ($V_{REF} = V_{REF} (+) - V_{REF} (-)$) t_{D} Internal Time Delay vs Temperature

Output Current vs Temperature

*1 LSB = $\frac{V_{REF}}{255}$

TL900001-11

Description of Pin Functions

Pin Name	Function	Pin Name	Function
1 V_{ref}	Analog input range - $GND \leq V_{ref} \leq V_{CC}$	9 INT	WR-RD Mode INT going low indicates that the conversion is completed and the data result is in the output latch. INT will go low, ~ 800 ns (the preset internal time out, t_i) after the rising edge of WR (see Figure 3b); or INT will go low after the falling edge of RD, if RD goes low prior to the 800 ns time out (see Figure 3a). INT is reset by the rising edge of RD or CS (see Figure 3a and 3b).
2 DB0	TRI-STATE data output—bit 0 (LSB)	10 GND	Ground
3 DB1	TRI-STATE data output—bit 1	11 $V_{REF(-)}$	The bottom of resistor ladder, voltage range: $GND \leq V_{REF(-)} \leq V_{REF(+)}$ (Note 5)
4 DB2	TRI-STATE data output—bit 2	12 $V_{REF(+)}$	The top of resistor ladder, voltage range: $V_{REF(-)} \leq V_{REF(+)} \leq V_{CC}$ (Note 5)
5 DB3	TRI-STATE data output—bit 3	13 CS	CS must be low in order for the RD or WR to be recognized by the converter.
6 WR/RDY	WR-RD Mode WR: With CS low, the conversion is started on the falling edge of WR. Approximately 800 ns (the preset internal time out, t_i) after the WR rising edge, the result of the conversion will be strobed into the output latch, provided that RD does not occur prior to this time out (see Figures 3a and 3b). RD Mode RDY: This is an open drain output (no internal pull-up device). RDY will go low after the falling edge of CS; RDY will go TRI-STATE when the result of the conversion is strobed into the output latch. It is used to simplify the interface to a microprocessor system (see Figure 2). Mode: Mode selection input—it is internally tied to GND through a 50 μA current source. RD Mode: When mode is low WR-RD Mode: When mode is high	14 DB4	TRI-STATE data output—bit 4
OUT		15 DB5	TRI-STATE data output—bit 5
7 Mode		16 DB6	TRI-STATE data output—bit 6
8 RD	WR-RD Mode With CS low, the TRI-STATE data outputs (DB0-DB7) will be activated when RD goes low (see Figure 4). RD can also be used to increase the speed of the converter by reading data prior to the preset internal time out (t_i , ~ 800 ns). If this is done, the data result transferred to output latch is latched after the falling edge of the RD (see Figures 3a and 3b). RD Mode With CS low, the conversion will start with RD going low, also RD will enable the TRI-STATE data outputs at the completion of the conversion. RDY going TRI-STATE and INT going low indicates the completion of the conversion (see Figure 4).	17 DB7	TRI-STATE data output—bit 7 (MSB)
IN		18 OPL	Overflow output—if the analog input is higher than the $V_{REF(+)}$, OPL will be low at the end of conversion. It can be used to cascade 2 or more devices to have more resolution (8, 10-bit).
		19 NC	No connection
		20 VCC	Power supply voltage

1.0 Functional Description

1.1 GENERAL OPERATION

The ADC0820 uses two 4-bit flash A/D converters to make an 8-bit measurement (Figure 1). Each flash ADC is made up of 15 comparators which compare the unknown input to a reference ladder to get a 4-bit result. To take a full 8-bit reading, one flash conversion is done to provide the 4 most significant data bits (via the MS flash ADC). Driven by the 4 MSBs, an internal DAC recreates an analog approximation of the input voltage. This analog signal is then subtracted from the input, and the difference voltage is converted by a second 4-bit flash ADC (the LS ADC), providing the 4 least significant bits of the output data word.

The internal DAC is actually a subsection of the MS flash converter. This is accomplished by using the same resistor ladder for the A/D as well as for generating the DAC signal. The DAC output is actually the tap on the resistor ladder which most closely approximates the analog input. In addition, the "sampled-data" comparators used in the ADC0820 provide the ability to compare the magnitudes of several analog signals simultaneously, without using input summing amplifiers. This is especially useful in the LS flash ADC, where the signal to be converted is an analog difference.

2147H HIGH SPEED 4096 x 1 BIT STATIC RAM

	2147H-1	2147H-2	2147H-3	2147HL-3	2147H	2147HL
Max. Access Time (ns)	35	45	55	55	70	70
Max. Active Current (mA)	180	180	180	125	180	140
Max. Standby Current (mA)	30	30	30	15	20	10

- Pinout, Function, and Power Compatible to Industry Standard 2147
- HMOS II Technology
- Completely Static Memory—No Clock or Timing Strobe Required
- Equal Access and Cycle Times
- Single +5V Supply
- 0.8-2.0V Output Timing Reference Levels
- Direct Performance Upgrade for 2147
- Automatic Power-Down
- High Density 18-Pin Package
- Directly TTL Compatible—All Inputs and Output
- Separate Data Input and Output
- Three-State Output

The Intel® 2147H is a 4096-bit static Random Access Memory organized as 4096 words by 1-bit using HMOS-II, Intel's next generation high-performance MOS technology. It uses a uniquely innovative design approach which provides the ease-of-use features associated with non-clocked static memories and the reduced standby power dissipation associated with clocked static memories. To the user this means low standby power dissipation without the need for clocks, address setup and hold times, nor reduced data rates due to cycle times that are longer than access times.

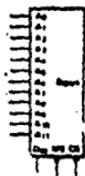
CS controls the power-down feature. In less than a cycle time after **CS** goes high—deselecting the 2147H—the part automatically reduces its power requirements and remains in this low power standby mode as long as **CS** remains high. This device feature results in system power savings as great as 85% in larger systems, where the majority of devices are deselected.

The 2147H is placed in an 18-pin package configured with the industry standard 2147 pinout. It is directly TTL compatible in all respects: inputs, output, and a single +5V supply. The data is read out nondestructively and has the same polarity as the input data. A data input and a separate three-state output are used.

PIN CONFIGURATION



LOGIC SYMBOL



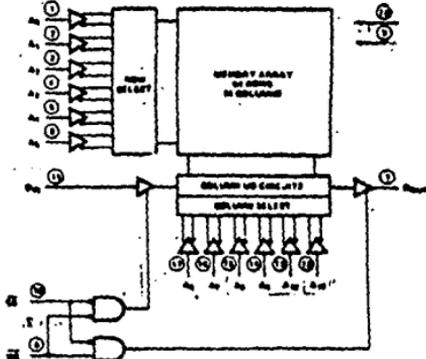
PIN NAMES

Pin	Function	Power / Input
A0-A17	ADDRESS INPUTS	5V
WE	WRITE ENABLE	2147: GROUND
CS	Chip SELECT	
D0-D17	DATA INPUT	
D0-D17	DATA OUTPUT	

TRUTH TABLE

CS	WE	MODE	OUTPUT	ADDRESS
H	X	NOT SELECTED	HIGH Z	STANDBY
L	L	WRITE	HIGH Z	ACTIVE
L	H	READ	DATA	ACTIVE

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias	-10°C to 60°C
Storage Temperature	-55°C to +150°C
Voltage on Any Pin	
With Respect to Ground	-3.5V to +7V
Power Dissipation	1.2W
D.C. Output Current	20 mA

COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

($T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5\text{V} \pm 10\%$, unless otherwise noted.)

Symbol	Parameter	2147H-1, 2, 3 Min. Typ. Max.	2147H-2 Typ. Min. Max.	2147H-3 Typ. Min. Max.	2147H-4 Typ. Min. Max.	Unit	Test Conditions
I_{IH}	Input Load Current (AR Input Pin)	0.01 10	0.01 10	0.01 10	0.01 10	μA	$V_{OH} = \text{Min.}$ $V_{OL} = \text{GND to } V_{CC}$
I_{OH}	Output Leakage Current	0.1 50	0.1 50	0.1 50	0.1 50	μA	$V_{OH} = V_{CC}$, $V_{OL} = \text{Min.}$ $V_{OUT} = \text{GND to } \pm 2.5\text{V}$
I_{OZ}	Operating Current	100 170	110	100 190	100 130	mA	$T_A = 25^\circ\text{C}$, $V_{OH} = \text{Min.}$ $T_A = 0^\circ\text{C}$, $V_{OH} = \text{Max.}$ Outputs Open
I_{DD}	Standby Current	110 130	110	110 130	7 10	mA	$V_{OH} = \text{Min. to Max.}$ $V_{OL} = V_{CC}$
I_{DDP}	Peak Power-On Current	25 70	25 50	25 50	15 30	mA	$V_{OH} = \text{GND to } V_{CC} \text{ Min.}$ $V_{OL} = \text{Linear of } V_{CC} \text{ to } V_{OH} \text{ Min.}$
V_{IL}	Input Low Voltage	-3.0 0.0	-3.0 0.0	-3.0 0.0	-3.0 0.0	V	
V_{IH}	Input High Voltage	2.0 5.0	2.0 5.0	2.0 5.0	2.0 5.0	V	
V_{OL}	Output Low Voltage	0.1	0.1	0.1	0.1	V	$I_{OH} = 0 \text{ mA}$
V_{OH}	Output High Voltage	2.4	2.4	2.4	2.4	V	$I_{OH} = -4.0 \text{ mA}$

807200

- The operating ambient temperature range is consistent with the power dissipation limits of the device.
- Typical limits are at $V_{CC} = 5\text{V}$, $T_A = 25^\circ\text{C}$, and specified loading.
- A pull-up resistor to V_{CC} on the OE input is required to keep the device functional otherwise, power current approaches I_{DD} .

A.C. TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	5 ns
Input Timing Reference Levels	1.5V
Output Timing Reference Level (2147H-1)	1.5V
Output Timing Reference Levels (2147H, H-2, H-3, HL, HL3)	0.5-2.0V
Output Load	See Figure 1

CAPACITANCE⁶ ($T_A = 25^\circ\text{C}$, $f = 1.0 \text{ MHz}$)

Symbol	Parameter	Max.	Unit	Conditions
C_{IN}	Input Capacitance	5	pF	$V_{OH} = 0\text{V}$
C_{OUT}	Output Capacitance	5	pF	$V_{OUT} = 0\text{V}$

NOTE:

- This parameter is sampled and not 100% tested.

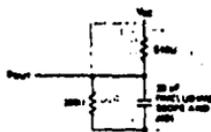


Figure 1. Output Load

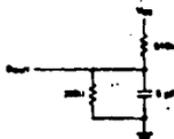


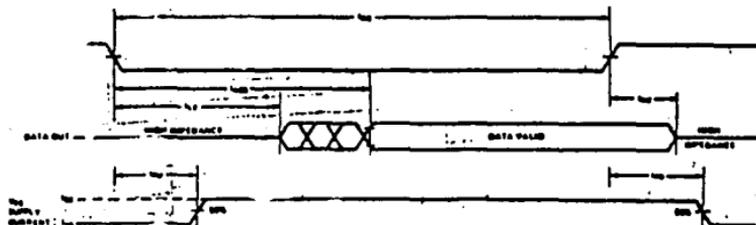
Figure 2. Output Load for 1 kHz, 10 kHz, low

A.C. CHARACTERISTICS ($T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = +5V \pm 10\%$, unless otherwise noted)

Read Cycle

Symbol	Parameter	2147H-1		2147H-2		2147H-3, HL-3		2147H, 2147HL		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
$t_{RD}^{(1)}$	Read Cycle Time	35		45		55		70		ns
$t_{AA}^{(2)}$	Address Access Time		35		45		55		70	ns
$t_{CS}^{(3)}$	Chip Select Access Time		35		45		55		70	ns
$t_{ACE}^{(4)}$	Chip Select Access Time		35		45		65		80	ns
$t_{OH}^{(5)}$	Output Hold from Address Change	5		5		5		5		ns
$t_{LZ}^{(6,7)}$	Chip Selection to Output in Low Z	5		5		10		10		ns
$t_{HZ}^{(6,7)}$	Chip Deselection to Output in High Z	0	30	0	30	0	30	0	40	ns
$t_{PU}^{(8)}$	Chip Selection to Power Up Time	0		0		0		0		ns
$t_{PD}^{(8)}$	Chip Deselection to Power Down Time	20		20		20		30		ns

WAVEFORMS

Read Cycle No. 1⁽¹⁾Read Cycle No. 2⁽¹⁾

NOTES:

- All Read Cycle timings are referenced from the last valid address to the first transitioning address.
- At any given temperature and voltage condition, t_{OH} max. is less than t_{LZ} min. both for a given device and from device to device.
- Transition is measured at 500 mV from steady state voltage with specified loading in Figure 2.
- \overline{CS} is high for Read Cycles.
- Device is continuously selected, $\overline{CS} = V_L$.
- Addresses valid prior to or coincident with \overline{CS} transition low.
- This parameter is sampled and not 100% tested.
- Chip deselected for greater than 55 ns prior to selection.
- Chip deselected for a finite time that is less than 55 ns prior to selection. If the deselect time is 0 ns, the chip is by definition selected and access occurs according to Read Cycle No. 1. Applies to 2147H, 2147HL, 2147H-3, and 2147HL-3.

A.C. CHARACTERISTICS (Continued)

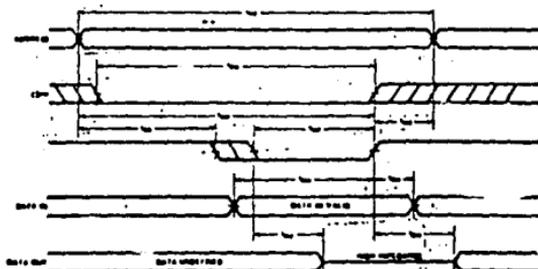
Write Cycle

Symbol	Parameter	2147H-1		2147H-2		2147H-3, HL-3		2147H, 2147HL		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
$t_{WC}^{(1)}$	Write Cycle Time	35	45	45	55	55	70	ns		
$t_{CS}^{(2)}$	Chip Selection to End of Write	35	45	45	55	55	70	ns		
$t_{AV}^{(3)}$	Address Valid to End of Write	35	45	45	55	55	70	ns		
$t_{AS}^{(3)}$	Address Setup Time	0	0	0	0	0	0	ns		
$t_{WP}^{(4)}$	Write Pulse Width	20	25	25	25	25	40	ns		
$t_{WR}^{(4)}$	Write Recovery Time	0	0	0	10	10	15	ns		
$t_{DVE}^{(4)}$	Data Valid to End of Write	20	25	25	25	25	30	ns		
$t_{DHS}^{(4)}$	Data Hold Time	10	10	10	10	10	10	ns		
$t_{WE}^{(5)}$	Write Enabled to Output in High Z	0	20	0	25	0	25	0	35	ns
$t_{OZ}^{(6)}$	Output Active from End of Write	0	0	0	0	0	0	ns		

WAVEFORMS

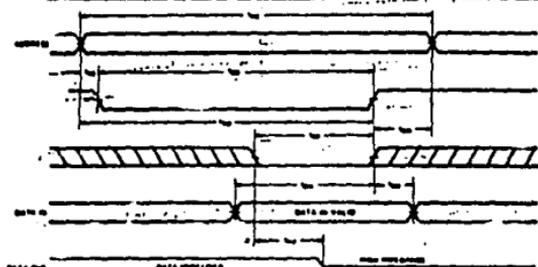
Write Cycle No. 1

(WE CONTROLLED)



Write Cycle No. 2

(CS CONTROLLED)



NOTES:

1. If \overline{CS} goes high simultaneously with \overline{WE} high, the output remains in a high impedance state.
2. All Write Cycle timings are referenced from the last valid address to the last transitioning address.
3. Transition is measured a 500 mV from steady state voltage with specified loading in Figure 2.
4. \overline{CS} or \overline{WE} must be high during address transitions.

APENDICE F

COSTOS DEL SISTEMA.

Componentes:

Circuito	Cantidad	Precio (USD.)	Total
74LS04	1	0.43	0.43
74LS08	2	0.43	0.86
74LS11	1	0.43	0.43
74LS32	3	0.52	1.56
74LS74	2	0.52	1.04
74LS138	1	0.53	0.53
74LS155	1	0.61	0.61
74LS244	4	0.84	3.36
74LS245	4	0.84	3.36
TMS32010	1	32.00	32.00
2147	16	1.50	24.00
ADC820	1	14.64	14.64
CRISTAL	1	2.60	2.60
TARJETA	1	42.00	42.00
BASES	38	1.00	38.00
WIRE WRAP	6	3.08	18.48
MISCELANEOS		4.00	4.00
		Total	187.90

Tiempo de ingeniería.

1170 Hrs X 3.90 USD = 4563.00 USD

Equipo.

Los costos se encuentran distribuidos principalmente en:

Osciloscopio: 1,000.00

PC XT 800.00

Generador de funciones 160.00

Total: 1,960.00 USD

BIBLIOGRAFIA

1. Cooper, David W. - Instrumentación Electrónica y Mediciones
Prentice Hall Internacional
Colombia, 1982
2. Klein, P. F. - El Osciloscopio
Editorial Marcombo S.A.
México
3. Sutaner, H. - El Osciloscopio y su Uso
Editorial Marcombo S.A.
México
3. Hall, Douglas V. - Microprocessors and Interfacing,
Programming and Hardware
Mc Graw Hill
E.E.U.U., 1986
4. Aliphas, Amnon
Fieldman, Joel - The Versatility of DSP Chips
IEEE Spectrum, June 1986
5. Meehon, Pat
Reidy, John - Spectrum Analyzer Using FFT Techniques
Electronics & Wireless World, December 1988
6. Nolen, Julian - Working with DSP
Electronics & Wireless World, January 1989
7. Data Aquisition using the IBM PC
Electronics & Wireless World, March 1989
8. Stone, Harold S. - Micro Computer Interfacing
Addison Wesley Publishing Co.
E.E.U.U. 1983
9. IBM PC Technical Reference Manual
IMB Corp.
E.E.U.U. 1980
10. Rauch, Ken - Math Chips : How They Work
IEEE Spectrum, July 1987
11. Eggebrecht, L. - Interfacing the IBM PC
Howard W. Sams & Co.
E.E.U.U. 1987
12. TMS 32010 User's Guide / DSP Products
Texas Instruments
E.E.U.U. 1983
13. Papoulis, A. - Sistemas Digitales y Analógicos,
Transformada de Fourier,
Estimación Espectral
Editorial Marcombo S.A.
México

14. DSP Applications with the TMS 320 Family
Texas Instruments
E.E.U.U. 1986
15. Tzafestas, S. - Digital Techniques in Simulation,
Communications and Control
North Holland Editors
E.E.U.U. 1985
16. Cardzow, James - Discrete Time Systems
Prentice Hall
E.E.U.U.
17. Daebelin - Diseño y Aplicación de Sistemas de Medición
18. Cage, Oliver - Electronic Measurements & Instrumentation
Mc Graw Hill
E.E.U.U.
19. Interconexión de Periféricos a
Microprocesadores
Editorial Marcombo S.A.
México
20. Worth, Steve - Introduction to Real-Time Software Design
Srpingeo - Verlag Editors
E.E.U.U. 1981
21. Carr, Joseph - Microprocessor Based Instrumentation
22. Oppenheim, Alan - Digital Signal Processing
Prentice Hall International
E.E.U.U. 1975
23. Papoulis, A. - Signal Analysis
Mc Graw Hill
E.E.U.U. 1984
24. Schwatz, Micha - Transmisión de Información, Modulación y
Ruido
Mc Graw Hill
México
25. Muñoz, Caupolican- Adquisición y Procesamiento de Señales
Curso de la D.E.C.F.I. - U.N.A.M.
Junio 1988