

35
20



**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN**



**DISEÑO Y CONSTRUCCION DE UN MODULO DE
OSCILOSCOPIO DE ALMACENAMIENTO DIGITAL
PARA LA IBM PC APLICADO EN UN SISTEMA DE
INSTRUMENTACION DIGITAL ASISTIDO POR
COMPUTADORA**

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
P R E S E N T A :
ARMANDO ROMERO DOMINGUEZ

ASESOR: ING. JORGE BUENDIA GOMEZ

CUAUTITLAN IZCALLI, EDO. DE MEX.

1994

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

U. N. A. M.
FACULTAD DE ESTUDIOS
SUPERIORES - CUAUTITLAN



DEPARTAMENTO DE
EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLAN
P R E S E N T E . .

AT'N: Ing. Rafael Rodríguez Ceballos
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:

"Diseño y construcción de un módulo de Osciloscopio de
Almacenamiento Digital para la IBM PC aplicado en un sistema de
Instrumentación Digital Asistido por Computadora".
que presenta el pasante: Armando Romero Domínguez
con número de cuenta: 8657229-3 para obtener el TITULO de:
Ingeniero Mecánico Electricista

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautilán Izcalli, Edo. de Méx., a 2 de marzo de 1994

PRESIDENTE	Ing. Ubaldo Ramírez Urizar		030394
VOCAL	Ing. Héctor Santoyo Morales		940302
SECRETARIO	Ing. Jorge Buendía Gómez		03/03/94
PRIMER SUPLENTE	Ing. Blanca G. de la Peña Valencia		23/03/94
SEGUNDO SUPLENTE	Ing. Nicolás Calva Tapia		

Indice

Indice	i
Contenido.....	ii
Capítulo 1 Introducción	1
PARTE 1: FUNDAMENTOS DE LA COMPUTADORA PERSONAL	9
Capítulo 2 Hardware de la IBM PC	10
Capítulo 3 Arquitecturas de bus en sistemas XT y AT	27
Capítulo 4 Interrupciones y DMA	55
PARTE 2: FUNDAMENTOS DEL OSCILOSCOPIO	75
Capítulo 5 Osciloscopios Analógicos	76
Capítulo 6 Osciloscopios Digitales	85
PARTE 3: DISEÑO DEL HARDWARE	96
Capítulo 7 Especificaciones de diseño del hardware	97
Capítulo 8 Interfase con la IBM PC	103
Capítulo 9 Conversión A/D	113
Capítulo 10 Módulo de osciloscopio digital	135
PARTE 4: DISEÑO DEL SOFTWARE	164
Capítulo 11 Especificaciones de diseño del software	165
Capítulo 12 Arquitectura del programa	184
Capítulo 13 Librerías de funciones	213
PARTE 5: GUIA DE USUARIO Y CONCLUSIONES	228
Capítulo 14 Guía de usuario	229
Capítulo 15 Conclusiones y comentarios finales	242
APENDICES Y BIBLIOGRAFIA	249
A Definición de términos para convertidores A/D y D/A	249
B Definición de términos para osciloscopios	253
C Definición de términos para la IBM PC	256
D Modos de operación de microprocesadores Intel	261
Bibliografía	264

Contenido

Índice	i
Contenido	ii

Capítulo 1 Introducción 1

1.1	Introducción	1
1.2	Definición del problema e importancia.....	2
1.3	Alcances, limitaciones y objetivos.....	3
1.4	¿Por qué IBM PC?.....	4
1.5	Organización del trabajo.....	6

Capítulo 2 Hardware de la IBM PC 10

2.1	Introducción	10
2.2	Monitor de la IBM PC	10
2.3	El teclado de la IBM PC	12
2.4	La unidad del sistema.....	12
2.4.1	La tarjeta del sistema.....	12
2.4.2	El microprocesador Intel 8088	14
2.4.3	La memoria RAM del sistema.....	15
2.4.4	La memoria ROM del sistema.....	16
2.4.5	El contador-temporizador del sistema	17
2.4.6	El controlador de DMA del sistema.....	17
2.4.7	La interfase a periféricos programable del sistema (PPI).....	19
2.4.8	El canal de entrada/salida (E/S)	20
2.4.9	La fuente de voltaje.....	21
2.5	La computadora IBM PC AT	21
2.6	Sistemas 386 de bus AT	23
2.7	Sistemas PS/2.....	23
2.8	Sistemas 486 y Pentium	25
2.9	Sumario.....	26

Capítulo 3 Arquitecturas de bus en sistemas XT y AT 27

3.1	Introducción	27
3.2	Arquitecturas de bus.....	27
3.3	Señales del bus de expansión en sistemas XT	30
3.4	Operaciones en el bus de expansión en sistemas XT	35
3.4.1	Ciclos de bus controlados por el microprocesador.....	36

3.4.1.1	Ciclo de bus de lectura de memoria	36
3.4.1.2	Ciclo de bus de escritura de memoria	38
3.4.1.3	Ciclo de bus de lectura de puerto de E/S	39
3.4.1.4	Ciclo de bus de escritura de puerto de E/S	40
3.4.2	Ciclos de bus controlados por DMA	42
3.4.2.1	Ciclo de bus de escritura de memoria por DMA	43
3.4.2.2	Ciclo de bus de lectura de memoria por DMA	44
3.5	Señales en el bus de expansión AT	46
3.6	Dimensiones de tarjetas en el bus de expansión XT y AT	52
3.7	Sumario	53

Capítulo 4 Interrupciones y DMA 55

4.1	Introducción	55
4.2	El proceso de interrupción	55
4.3	Típos de interrupción	57
4.3.1	Interrupciones de hardware	57
4.3.2	Interrupciones de software	58
4.4	Interrupciones en el 8088	59
4.5	El Controlador Programable de Interrupciones (PIC) 8259A	61
4.5.1	Palabras de comando de inicialización (ICW)	64
4.5.2	Palabras de comando de operación (OCW)	68
4.6	Interrupciones en el canal de E/S	69
4.7	Acceso Directo a Memoria (DMA)	70
4.8	Secuencia de la transferencia de datos por DMA	70
4.9	Razón de transferencia de DMA	72
4.10	Programación del 8237A-5	72
4.11	Sumario	74

Capítulo 5 Osciloscopios Analógicos 76

5.1	Introducción	76
5.2	Fundamentos del osciloscopio	76
5.3	El tubo de rayos catódicos	78
5.3.1	El cañón electrónico	79
5.3.2	Elementos de foco y aceleración	80
5.3.3	Placas de deflexión horizontal y vertical	81
5.3.4	Cubierta de vidrio al vacío	81
5.4	La base de tiempos y el disparo	82
5.5	La puntas del osciloscopio	83
5.6	Sumario	84

Capítulo 6	Osciloscopios Digitales	85
6.1	Introducción	85
6.2	Descripción funcional de un DSO	86
6.3	Características generales de un DSO	87
6.4	Ancho de banda y razón de muestreo	89
6.5	Muestreo en tiempo real	90
6.6	Muestreo en tiempo equivalente	92
6.7	Sumario	95
Capítulo 7	Especificaciones de diseño del hardware	97
7.1	Introducción	97
7.2	Descripción global del proyecto	97
7.3	Diseño del hardware	99
	7.3.1 Adquisición de datos	99
	7.3.2 Interfase de datos	101
7.4	Sumario	102
Capítulo 8	Interfase con la IBM PC	103
8.1	Introducción	103
8.2	El PPI 8255	103
	8.2.1 Pines del PPI	104
	8.2.2 Programación del PPI	106
8.3	Diseño de una interfase con la IBM PC	107
8.4	Comunicaciones con la IBM PC por DMA	110
8.5	Sumario	111
Capítulo 9	Conversión A/D	113
9.1	Introducción	113
9.2	Conversión de valores analógicos a digitales	113
9.3	Circuitos de muestreo y retención	117
9.4	Características en un convertidor A/D	118
9.5	Tipos de error en convertidores A/D	119
9.6	Técnicas y tipos de convertidores A/D	121
	9.6.1 Convertidor A/D con rampa digital	121
	9.6.2 Convertidor A/D de rampa dual	124
	9.6.3 Convertidor A/D con convertidor de voltaje a frecuencia	126
	9.6.4 Convertidor A/D de aproximaciones sucesivas	127
	9.6.5 Convertidor A/D de ráfaga	129

9.7	Parámetros de selección de un convertidor A/D.....	132
9.8	Sumario.....	134

Capítulo 10 Módulo de osciloscopio digital 135

10.1	Introducción.....	135
10.2	Descripción global del módulo.....	135
10.3	Selector de módulo.....	137
10.4	Optoacopladores.....	140
10.5	Control de E/S y buffer de registros.....	141
10.6	Indicadores luminosos.....	145
10.7	Fuente de voltaje.....	146
10.8	Acoplo de CA/CD.....	147
10.9	Proceso de la señal.....	148
10.11	Base de tiempos.....	149
10.12	Frecuencímetro.....	151
10.13	Adquisición y almacenamiento.....	156
10.14	Secuencia de operaciones.....	158
10.14.1	Selección del módulo.....	159
10.14.2	Inicialización.....	159
10.14.3	Programación.....	160
10.14.4	Adquisición de datos.....	160
10.14.5	Transmisión de los datos.....	161
10.15	Sumario.....	163

Capítulo 11 Especificaciones de diseño del software 165

11.1	Introducción.....	165
11.2	Plataforma de software.....	165
11.3	Lenguaje de programación.....	167
11.4	Compilador.....	168
11.5	Nombre oficial del programa.....	169
11.6	Hardware requerido.....	169
11.7	Elementos del programa.....	170
11.7.1	Argumentos desde la línea de comandos.....	170
11.7.2	El ratón.....	171
11.7.3	Cajas de diálogo.....	171
11.7.4	Mensajes.....	173
11.7.5	Iconos.....	174
11.7.6	Menús.....	174
11.7.7	Archivos.....	174
11.7.8	Pantallas del programa.....	176

13.3	Manejo de memoria.....	215
13.4	Manejo de textos.....	216
13.5	Manejo de gráficos.....	217
13.5.1	Manejo de pixeles.....	222
13.6	Manejo de cadenas de caracteres.....	223
13.7	Manejo del ratón.....	224
13.8	Manejo del teclado.....	226
13.9	Sumario.....	227

Capítulo 14 Guía de usuario 229

14.1	Introducción.....	229
14.1.1	¿Qué es INDICO?.....	229
14.1.2	Requerimientos de hardware y software.....	230
14.1.3	Qué se necesita saber para usar INDICO.....	231
14.1.4	Especificaciones técnicas de INDICO.....	232
14.2	Comenzando INDICO.....	232
14.2.1	Instalación.....	232
14.2.1.1	Instalación del hardware.....	233
14.2.1.2	Instalación del software.....	234
14.2.2	Iniciando el programa INDICO.....	234
14.2.3	La distribución de la pantalla.....	235
14.3	Uso de INDICO.....	238
14.3.1	Funciones de archivo.....	238
14.3.2	Captura de una señal.....	238
14.3.3	Funciones relativas a la pantalla.....	239
14.3.4	Control y estadísticas de la señal.....	239
14.3.5	Operaciones de cursor.....	240
14.3.6	Control de la posición Y de la señal.....	240
14.3.7	Configuración.....	240
14.3.8	Funciones en los iconos.....	241

Capítulo 15 Conclusiones y comentarios finales 242

A	Definición de términos para convertidores A/D y D/A.....	249
B	Definición de términos para osciloscopios.....	253
C	Definición de términos para la IBM PC.....	256
D	Modos de operación de microprocesadores Intel.....	261
	Bibliografía.....	264

Capítulo 1 Introducción

1.1 Introducción

La electrónica ha tenido un desarrollo impresionante en este siglo. Desde la invención del transistor hasta la fecha, los avances logrados han sido rápidos y sostenidos. Aparatos cuyo funcionamiento se encontraba basado en métodos neumáticos o mecánicos ahora son electrónicos, y los aparatos netamente electrónicos son ahora más sofisticados. Evidentemente, en la creación, prueba y mantenimiento de estos dispositivos es necesario contar con instrumentos que permitan realizar mediciones de diversos parámetros con objeto de evaluación. De esta forma, multímetros, osciloscopios, frecuencímetros, analizadores lógicos, analizadores de espectro, puentes de impedancias y una larga lista de aparatos de medición han sido creados para satisfacer estos requerimientos. Estos instrumentos, naturalmente, tampoco están exentos de sufrir cambios. Los primeros multímetros, por ejemplo, eran únicamente de lectura analógica. Actualmente es posible encontrar multímetros digitales controlados por microprocesador y con más funciones que los analógicos.

Estos cambios en la instrumentación han alcanzado el terreno de las computadoras personales. Dada la capacidad de almacenamiento y procesamiento de datos de una computadora, no es raro encontrar que algunos fabricantes saquen provecho de ella. Los modernos osciloscopios digitales, por ejemplo, cuentan

con una interfase serial que se apega a las especificaciones del puerto serial de comunicaciones RS-232 de una computadora personal. Tienen la capacidad de enviar una señal capturada para que sea dibujada por un plotter, o a la computadora para que sea manipulada, procesada o almacenada.

Otros fabricantes van más allá e incorporan el osciloscopio directamente en la computadora a través de los buses de expansión de ésta. La idea de usar la computadora como osciloscopio está basada en el hecho de que la computadora ya tiene un monitor de video y puede manejar gráficas. Todo lo que se requiere, por lo tanto, es una unidad que almacene la señal a ser medida en la memoria de la computadora para que ésta pueda mostrar los resultados en el monitor. Si bien esto parece ser muy simple, la tarea puede alcanzar complejos niveles de sofisticación dependiendo de las metas a alcanzar.

1.2 Definición del problema e importancia

En esta tesis se plantea el diseño y construcción de un módulo de osciloscopio digital para la IBM PC y compatibles. Aprovechando las capacidades de la computadora, se consideró como una de las características del proyecto la posibilidad de aceptar más módulos de instrumentación, de tal manera que el equipo inicial pudiera crecer añadiéndole nuevos módulos independientes. Debido a esto, se adoptó como nombre genérico para este proyecto el de INDICO, a partir de las siglas de "Instrumentación Digital asistida por Computadora".

¿Por qué hacer el diseño de algo que ya existe a nivel comercial? La importancia de este proyecto se puede observar considerando los siguientes factores:

Precio. Los modelos comerciales con características muy básicas cuestan desde 200 dólares y llegan a alcanzar precios más elevados conforme crece la sofisticación. Un diseño personal permite ajustar presupuestos y otorgarle al producto las características que se quieran.

Ajuste a las necesidades. Cuando se compra un modelo comercial el cliente debe ajustarse a lo que el producto ofrece. En el caso de un diseño personal, se tiene la oportunidad de analizar los productos disponibles y en base a esto aplicarles ingeniería inversa para mejorarlos, ofreciendo nuevas facilidades. Una de ellas se acaba de mencionar en párrafos anteriores. Si bien ya existen osciloscopios por computadora, la mayoría aíslan esta tarea y no proporcionan capacidad de expansión hacia otros módulos de instrumentación.

1.3 Alcances, limitaciones y objetivos.

Por supuesto, en éste trabajo se encuentran desventajas con respecto a modelos comerciales. Tales modelos cuentan con el soporte de una planta de producción, tecnología de punta y un grupo de investigación y desarrollo (R&D : Research and Development) dedicado al producto con altos presupuestos. Phillips, por ejemplo, destina más de 1,000 millones de dólares anuales en investigación. Desde luego, en el diseño que se presenta en esta tesis no se pretende competir al mismo nivel con tales modelos por las limitaciones mencionadas, pero esto no es motivo para presentar un mal diseño. De hecho, la única limitación mayor con la que opera el equipo presentado es el ancho de banda de adquisición de datos.

Debido a las características propias del trabajo, existe otra limitación en el producto. Ya se mencionó el ancho de banda del osciloscopio desarrollado. Otra limitación es el monitor de la PC, ya que su resolución puede llegar a limitar la presentación de la señal. Esto puede ser atenuado por el software con funciones de zoom o cambio de escala en la pantalla.

Los objetivos a cumplir en este trabajo fueron:

- 1) Diseñar un sistema modular de instrumentación digital asistido por computadora que tenga facilidad de aceptar nuevos módulos y actualizaciones de los ya existentes, compatible tanto con computadoras IBM PC básicas como con aquellas con arquitecturas más avanzadas, de tal manera que se pueda abarcar al mayor número de usuarios. Igualmente, el sistema diseñado debe poderse ocupar a nivel didáctico en los laboratorios de electrónica en la FES-Cuautitlán y presentar, por lo tanto, protección contra daños en la PC por corto-circuito o sobrecargas.
- 2) Diseñar y construir un módulo de osciloscopio de almacenamiento digital para el sistema de instrumentación digital mencionado.

1.4 ¿Por qué IBM PC?

Un aspecto importante fue la computadora a seleccionar. Entre otras, existen computadoras personales IBM, Apple Macintosh, estaciones de trabajo NeXT, SPARCstation de SUN, APOLLO, IRIS INDIGO, VAX, etc. ¿Cuál es la indicada? Indudablemente, las estaciones de trabajo como SUN y NeXT presentan capacidades de procesamiento muy atractivas en las aplicaciones de ciencia e

ingeniería, pero su precio las pone fuera del alcance de la mayoría de los usuarios. La elección en este proyecto fue seleccionar IBM PC debido a varias razones:

Costo. El precio de una computadora personal es generalmente inferior al de otros tipos de computadoras, quedando al alcance del usuario común. Por ejemplo, una Apple Macintosh Quadra 800 con el microprocesador 68040 a 33 MHz se cotiza en 8,700 dólares, una estación de trabajo SUN SPARCstation LX con el microprocesador microSPARC a 50 MHz en 9,500 y una HP Apollo 9000 modelo 715/50 con el microprocesador PA/RISC 7100 a 50 MHz en 17,200. Por otro lado, una PC compatible con IBM Compaq Deskpro 5/66M modelo 510 con el microprocesador Pentium a 66 MHz cuesta entre 7,500 y 8,000 dólares (los precios dados corresponden a junio de 1993).

Expansión. Ya sea que se tenga una computadora IBM PC XT con el microprocesador 8088 de 10 Mhz o una computadora con el microprocesador 80486DX2 de 66 MHz, la capacidad de expansión es la misma y la compatibilidad entre ellas es completa. Por otro lado, computadoras como la Apple Macintosh Classic II no incluyen la capacidad de conectar tarjetas a la computadora ya que no tienen puertos de expansión.

Proyección. Si bien el mercado de las computadoras IBM PC o compatibles es dedicado prácticamente en su totalidad a aplicaciones administrativas, éstas han alcanzado grados de sofisticación que exigen un avance tecnológico con tendencias a alcanzar el rendimiento de avanzadas estaciones de trabajo dedicadas a ciencia e ingeniería. Se proyecta, por ejemplo, que futuras computadoras IBM PC

cuenten con la capacidad de manejar dos microprocesadores Pentium simultáneamente y que alcancen velocidades de 100 Mhz o más.

Versatilidad. Un usuario de PC puede realizar casi cualquier tarea con una computadora personal, e incluso intercambiar información de un programa a otro con relativa facilidad. Esta característica ha hecho que la cantidad de usuarios de PC sea muy superior a la de otras plataformas. En base a esto, proyectar un diseño que ejecute en computadoras que no muchos usuarios manejan lo convertiría en poco práctico.

1.5 Organización del trabajo

El diseño de todo proyecto consta de las siguientes etapas básicas:

- Definición del problema.** En esta etapa se define el problema y se delimita el alcance del proyecto a elaborar.
- Búsqueda de información.** Consiste en buscar y organizar toda la información disponible sobre el tema.
- Definición de las especificaciones de diseño.** Una vez que se sabe lo que se va a realizar y se tiene la información requerida, se definen detalladamente las características deseables del producto a elaborar.
- Descripción del diseño.** Basados en las especificaciones de diseño, se determina cómo llevar a cabo el proyecto

dividiéndolo en bloques funcionales y definiendo la solución para cada uno de éstos.

A partir de la metodología anterior, esta tesis se encuentra dividida en cinco partes. La primera de ellas, "**Fundamentos de la computadora personal**", analiza el hardware de la IBM PC para poder establecer las bases sobre las cuales parte el diseño y conocer las limitaciones que tendrá. Esta parte consta de tres capítulos. El capítulo 2, "*Hardware de la IBM PC*", analiza y describe los componentes fundamentales de una computadora IBM PC. El capítulo 3, "*Arquitecturas de bus en sistemas XT y AT*" describe el bus de expansión de la computadora personal y el capítulo 4, "*Interrupciones y DMA*" presenta las capacidades del sistema de interrupciones y acceso directo a memoria.

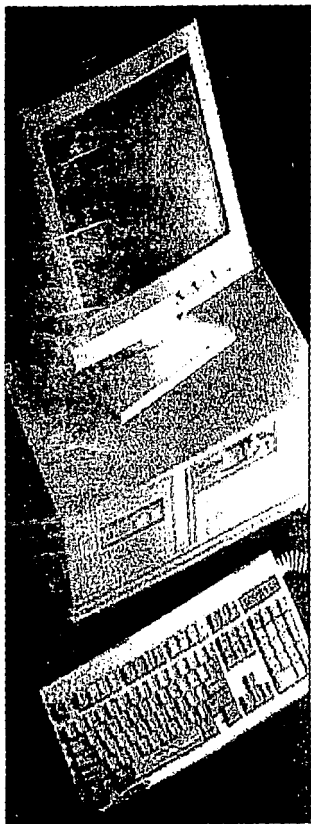
La parte 2, "**Fundamentos del osciloscopio**", analiza los componentes de un osciloscopio con objeto de establecer los elementos constitutivos del sistema a diseñar. Consta de los capítulos 5, "*Osciloscopios analógicos*" y el 6, "*Osciloscopios digitales*".

En la parte 3, "**Diseño del hardware**", se diseña el hardware de adquisición de datos y de transferencia de éstos hacia la PC tomando como base la información recopilada en las primeras dos partes del documento. Consta de cuatro capítulos. El capítulo 7, "*Especificaciones de diseño del hardware*", indica las características del proyecto a diseñar. En el capítulo 8, "*Interfase con la IBM PC*", se diseña un puerto de comunicaciones de propósito general usado para la transferencia de datos hacia la PC. El capítulo 9, "*Conversión A/D*", analiza las técnicas de conversión de datos analógicos a digitales y el capítulo 10, "*Módulo de osciloscopio digital*", se encarga de describir el diseño del hardware de adquisición de datos de éste proyecto.

La parte 4, "**Diseño del software**", se dedica a desarrollar el software requerido para operar el módulo de hardware. Consta de tres capítulos. El capítulo 11, "*Especificaciones de diseño del software*", indica los lineamientos y herramientas requeridas para el desarrollo del programa. En el capítulo 12, "*Arquitectura del programa*" se describe a nivel de bloques al programa desarrollado para operar al osciloscopio y en el capítulo 13, "*Librerías de funciones*", se describen las funciones auxiliares que se usaron en el programa.

La parte 5 y última, "**Manual de usuario y conclusiones**", presenta una guía para el uso del proyecto desarrollado y analiza el producto terminado. Consta de cuatro capítulos: el capítulo 14, "*Introducción*", el 15, "*Comenzando INDICO*", el 16, "*Uso de INDICO*" y el 17, "*Conclusiones y comentarios finales*".

Finalmente, se incluyen apéndices con glosarios de términos sobre convertidores A/D, osciloscopios y computadoras personales, además de una breve semblanza de los modos de operación de los microprocesadores Intel. Posteriormente a estos apéndices, se lista la bibliografía utilizada, clasificándola de acuerdo a su origen (libros manuales, tesis y revistas) y su tema (electrónica, software, etc.).



Parte 1

Fundamentos de la computadora personal

Objetivo: *Analizar el hardware de la IBM PC y establecer a partir de esto las bases sobre las cuales parte el diseño, así como las limitaciones del mismo.*

Capítulos

- Hardware de la IBM PC
- Arquitecturas de bus en sistemas XT y AT
- Interrupciones y DMA

Capítulo

2

Hardware de la IBM PC

2.1 Introducción

En este capítulo se introducirán las características principales de la IBM PC, preparando el camino para una discusión más detallada en capítulos posteriores para los elementos de hardware que así lo requieran. Partiremos analizando los tres componentes físicos más importantes en la IBM PC: el monitor, el teclado y la unidad del sistema. Tomando en cuenta la diferencia tan amplia entre estos elementos para los diversos modelos de computadoras personales, las especificaciones que se darán corresponden a una computadora IBM PC XT. Para finalizar el capítulo, se hablará en términos generales sobre computadoras IBM modelos AT, PS/2 y computadoras 386, 486 y Pentium, citando las mejoras que presentan con respecto a modelos que les preceden. Algunos de los conceptos mencionados, tales como ciclos de bus, DMA o interrupciones se cubren con mayor detalle en los capítulos posteriores.

2.2 Monitor de la IBM PC

IBM originalmente proporcionó dos tipos de monitor para usarse en la IBM PC: el monocromático y el de color. Cada uno de ellos se conecta a la unidad del sistema a través de una tarjeta de interfase específica que se inserta en la IBM PC.

El adaptador para monitor monocromático IBM estaba orientado al uso de caracteres, y era apropiado para mostrar información en forma de texto. La mayor limitación de éste tipo de monitor era que el conjunto de caracteres mostrados era fijo, impidiéndose el uso de gráficas sofisticadas. Sin embargo, IBM proporcionó un conjunto predefinido de caracteres gráficos, los cuales podían ser usados para crear algunos gráficos elementales, como diagramas de barras, por ejemplo.

El adaptador para monitor de colores Color Graphic Display Adapter o CGA soportaba el uso de caracteres, así como de modos de direccionamiento de puntos. Los caracteres no tenían la misma calidad que en el monitor monocromático, pero los modos de direccionamiento de puntos permitían mostrar información gráfica que podía reemplazar a la descripción textual. Además, el Color Graphics Adapter también soportaba conjuntos de caracteres definidos por el usuario.

Desafortunadamente, estos dos tipos de adaptadores no eran compatibles. No se podía usar un adaptador CGA con un monitor monocromático y viceversa. Si se deseaban las ventajas de los dos monitores se debían de comprar ambos adaptadores y tener un switch que permitiera conmutar entre el uso de ambas tarjetas. Sin embargo, algunas compañías que aprovecharon esta deficiencia presentando adaptadores con muchas ventajas. Uno de ellos, y que se convirtió en un estándar, era el fabricado por **Hercules Company**, el cual era compatible con el adaptador monocromático de IBM, y que además tenía un modo de direccionamiento de puntos que permitía el uso de gráficos. La tarjeta adaptadora Hercules tuvo tanta aceptación en el mercado, que otras compañías fabricaron adaptadores compatibles con Hercules.

2.3 El teclado de la IBM PC

El corazón del teclado de la IBM PC es el controlador de teclado 8048 de Intel. Cada vez que una tecla es presionada o liberada, el teclado interrumpe a la unidad del sistema y le envía, por medio de comunicación serial, un código de exploración (scan code). El programa manejador de teclado de la PC interpreta ese código. El 8048 realiza todas las operaciones de mantenimiento necesarias para la comunicación entre PC - teclado. Entre otras tareas, explora el estado del teclado, mantiene un buffer de hasta 16 códigos de exploración, y controla la comunicación serial con la PC.

2.4 La unidad del sistema

El hardware dentro de la unidad del sistema puede dividirse en cuatro partes (funcional y físicamente): Una tarjeta de circuito impreso llamada tarjeta del sistema o tarjeta madre (motherboard), ranuras de expansión, referidas como el canal de entrada/salida (canal de E/S), unidades de discos (al menos una, usualmente), y una fuente de alimentación para el sistema.

2.4.1 La tarjeta del sistema

La tarjeta del sistema es la que realmente define a la IBM PC. Si se reemplazara el monitor, el teclado, la unidad de discos y la fuente de alimentación, aún se tendría una IBM PC. La tarjeta del sistema contiene el microprocesador, los chips de soporte y el BIOS (Basic Input Output System - Sistema Básico de Entrada y Salida), los cuales convierten a la IBM PC en lo que es. Los modelos de PC compatibles con IBM deben de apearse mucho al

diseño de la tarjeta del sistema de la IBM PC, o simplemente no serán compatibles. Los principales componentes y características de la tarjeta del sistema son:

- El microprocesador 8088 de Intel.
- Un socket e interfase para el coprocesador matemático 8087.
- Un temporizador para el sistema, el 8253-5.
- Ocho ranuras (slots) de expansión.
- Un controlador de acceso directo a memoria (DMA : Direct Memory Access), el 8237A-5 de Intel.
- Un controlador de interrupciones, el 8259A de Intel.
- Una expansión del puerto paralelo, el 8255A de Intel.
- Sockets para 256 Kbytes de RAM dinámica.
- Hasta 48 kbytes de ROM.
- Una bocina para audio.
- Un banco de switches para establecer la configuración del sistema

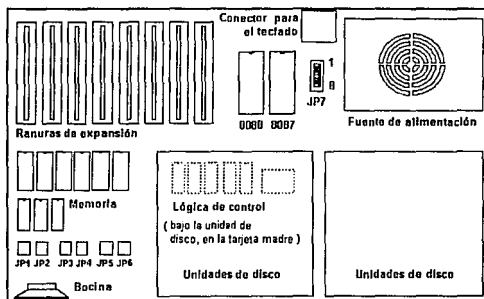


Figura 2.1 Elementos de la tarjeta madre en una computadora IBM PC XT

La figura 2.1 ilustra la distribución de los elementos principales en la motherboard. Por simplicidad en el dibujo, se omiten algunos de los elementos mencionados anteriormente.

2.4.2 El microprocesador Intel 8088

El microprocesador 8088 de Intel es comúnmente llamado un microprocesador de 16 bits debido a que puede procesar datos (sumar, restar, recorrer, etc.) 16 bits a la vez. Esto es en forma interna. En forma externa, debido a que su bus de datos es de sólo 8 bits, tiene que manejar los datos de 16 bits por partes, lo cual le resta cierta velocidad si lo comparamos con la lograda al transferir 16 bits a la vez.

Cuando el 8088 se comunica al exterior, tiene dos vías de acceso: memoria y puertos. La memoria puede ser direccionada empleando el total de sus 20 líneas de dirección, lo cual arroja una capacidad total de :

$$2^{20} = 1,048,576 \text{ bytes de información}$$

lo cual también puede expresarse como 1024 Kilobytes, donde un Kilobyte es igual a 1024 bytes, o como 1 MegaByte de información.

Por otro lado, cuando el microprocesador requiere acceso a puertos, se usan sólo 16 bits, lo cual le da una capacidad de:

$$2^{16} = 65,536 = 64 \text{ K puertos}$$

El 8088 soporta el uso de un circuito de apoyo para los cálculos matemáticos. Este circuito es el coprocesador matemático 8087.

Para instalarlo en una computadora, basta con insertarlo en la base de C.I. que le corresponde en la motherboard y ajustar la posición de uno de los switches de la misma para declararlo como instalado. El uso de este circuito incrementa grandemente la respuesta del μp cuando se hacen extensos cálculos matemáticos. Para poder hacer uso del coprocesador es necesario tener software capaz de manipularlo.

La velocidad de operación del 8088 en la IBM PC es de 4.77 Mhz, lo cual implica un período de reloj de 210 ns. El ciclo de acceso de memoria del 8088 toma cuatro pulsos de reloj, es decir, 840 ns aproximadamente. La lógica de la tarjeta del sistema inserta un estado de espera en todos los accesos a puerto de E/S, de modo que se tienen 5 ciclos de reloj, o 1050 ns como máxima velocidad operativa a la cual pueden introducirse o exportar datos de la IBM PC.

2.4.3 La memoria RAM del sistema

La tarjeta del sistema contiene hasta 256 Kbytes de memoria RAM dinámica, instalada en cuatro bancos de memoria de 64 Kbytes. Cada uno de esos bancos consta de 9 memorias. Un byte de información se reparte entre 8 de ellas, correspondiéndoles un bit a cada una. La novena memoria almacenará un bit de paridad, el cual es usado para revisar errores de memoria. Si uno de estos se presenta, la tarjeta del sistema genera una interrupción no mascarable (NMI) para el 8088, deteniendo a la computadora. La RAM del sistema se localiza en la parte baja de la memoria, en el rango de direcciones de 00000H a 3FFFFH. La memoria RAM dinámica debe ser refrescada a intervalos no mayores de 2 ms. El ciclo de refresco es realizado por el controlador de DMA, el 8237A-5 de Intel, a intervalos especificados por el circuito

temporizador 8253-5 de Intel cada 15.2 μ s. El 8088 es puesto en un estado de espera (WS - Wait State) durante el ciclo de refresco, lo cual disminuirá su velocidad efectiva. Un estado de espera es la adición de un pulso extra de reloj en un ciclo de bus. Para más información sobre ciclos de bus, remítase al capítulo 3.

2.4.4 La memoria ROM del sistema

La tarjeta del sistema contiene bases de C.I. para 48 Kbytes de ROM. La ROM está instalada en C.I.'s de 8 Kbytes y está ubicada en la parte alta de la memoria, en el rango de direcciones de F4000H a FFFFFH. La razón de esto se hace evidente si se tiene en cuenta que cuando el μ p 8088 recibe energía por primera vez, la primera localidad de memoria en donde busca instrucciones para ser ejecutadas es en la FFFF0H, prácticamente al final del mapa de memoria.

Los últimos 40 Kbytes de los 48 Kbytes totales de ROM almacenan código que implementa:

- El intérprete de BASIC.
- La auto-prueba de encendido (POST : Power On-Self Test).
- El Sistema Básico de Entrada/Salida (BIOS : Basic Input Output System).
- El cargador del sistema desde disco (diskette bootstrap loader).
- Los patrones de puntos para la generación de caracteres en modo gráfico.

2.4.5 El contador-temporizador del sistema

La tarjeta del sistema incluye un C.I. contador-temporizador 8253-5 de Intel que se emplea para añadir funciones de temporización al sistema. Además de intervenir en el refresco de la memoria dinámica, este circuito proporciona la señal de reloj para el μ p. El 8253-5 es en realidad tres temporizadores independientes en un mismo circuito. Cada uno de los canales del temporizador es usado para una tarea específica:

- ❑ **Canal 0:** reloj en tiempo real. Este canal está programado para interrumpir al 8088 cada 55 ms. La rutina de interrupción asociada proporciona el reloj en tiempo real.
- ❑ **Canal 1:** Ciclos de refresco de DMA. Este canal está programado para solicitar una transferencia de DMA del controlador de DMA, 8237A-5 cada 15.2 μ s. El controlador de DMA responde al requerimiento ejecutando un ciclo de refresco de memoria.
- ❑ **Canal 2:** interfase de cassette y bocina. Este canal es usado para poner tonos en una cinta. El canal 2 está conectado a la bocina de audio a través de una compuerta y puede ser usado para producir tonos, música o ruido.

2.4.6 El controlador de DMA del sistema

Siendo tan rápido como es, hay ocasiones en que el 8088 no alcanza la velocidad requerida para algunas de las operaciones que debe realizar el sistema. Tómese por ejemplo el acceso a disco. Un programa muy simple que haga un acceso a disco, aún sin incluir una rutina que verifique el estado de éste, dará una

velocidad de transferencia máxima de 100 Kbytes/segundo. Usando el controlador de DMA, la transferencia máxima se eleva a 422 Kbytes/segundo, lo cual incrementa 4 veces la velocidad. En la IBM PC, todas las transferencias de disco flexible y rígido, además del refresco a memoria, son realizadas por el chip de DMA.

El 8237A-5 es en realidad un chip muy flexible. Algunas de sus características importantes son:

- Cuatro canales independientes de DMA.
- Registros de dirección y conteo de 16 bits (Lo deseable, por supuesto, es de 20 bits, lo cual permitiría direccionar el total de la memoria).
- Cuatro tipos de transferencias:
 - memoria a puerto de E/S.
 - memoria a memoria.
 - puerto de E/S a puerto de E/S.
 - puerto de E/S a memoria.
- Esquema de programación para atender a prioridades.
- Sincronización programable y condiciones de terminación.

Como se puede apreciar, la flexibilidad de este circuito es muy amplia. Sin embargo, en la IBM PC se ha restringido un tanto, puesto que las transferencias de datos se han limitado a solo aquellas entre puertos de E/S y memoria. Además, el 8237A-5 está programado en el modo de transferencia simple, limitando la razón de transferencia a 422 KHz.

En la IBM PC se han hecho las siguientes asignaciones para los canales del DMA:

Canal 0:	Ciclos de refresco de memoria:
Canal 1:	No usado.

Canal 2:	Usado por el disco flexible.
Canal 3:	Usado por el disco duro.

Entre más bajo sea el número del canal, más alta es la prioridad. Así, la más alta prioridad corresponde al refresco de la memoria dinámica.

2.4.7 La interfase a periféricos programable del sistema (PPI)

A través de tres puertos programables en la tarjeta del sistema, es posible leer el código de exploración del teclado, los switches de configuración, activar la bocina del sistema y otras cosas más. El C.I. 8255A de Intel, un PPI (Programmable Peripheral Interface) cuyo uso en computadoras data de los tiempos del Z-80 de 8 bits de Zilog, es el encargado de la implementación de estos puertos en hardware. El 8255A tiene 24 pines de E/S digitales configurables en una amplia variedad de modos. En la IBM PC, el BIOS se encarga de configurar al 8255A para representar tres puertos paralelos de 8 bits como se muestra en la tabla 2.1.

El 8255A aparece como los puertos 60H, 61H y 62H. La tabla 2.1 ilustra el uso de cada puerto. Los puertos 60H y 62H son puertos de entrada, y el puerto 61H es de salida. Nótese que tanto el puerto 60H como los bits 0-3 del puerto 62H tienen una doble función. La elección particular de una u otra depende del valor del bit 7 en el puerto 61H. El bit 2 del puerto 61H controla la función de los bits 0-3 del puerto 62H.

Puerto	Bits	Descripción
60H	0-7	Código de exploración de teclado o configuración del sistema (ver bit 7, puerto 61H)
61H	0	Temporizador 2 (1= ON).
	1	Bocina (1=ON).
	2	1 = Switch del tamaño de memoria. 0 = Teclas compartidas (ver puerto 62H).
	3	Motor de la casetera (1 = OFF).
	4	0 = Habilitador de la revisión de la paridad de la memoria.
	5	0 = Habilitador de la revisión del canal de E/S.
	6	1 = Habilitador del reloj de teclado.
	7	1 = Switches de configuración. 0 = Exploración del teclado (ver puerto 60H).
62H	0-3	Switch del tamaño de memoria o teclas compartidas.
	4	Datos de entrada a cassette.
	5	Salida del temporizador 2.
	6	Revisión del canal de E/S.
	7	Revisión de la paridad de memoria.

Tabla 2.1 Configuración para el PPI 8255A en la IBM PC.

2.4.8 El canal de entrada/salida (E/S)

El canal de entrada salida es básicamente un bus provisto para expansión del sistema. El canal de E/S consiste en un bus de 62 líneas, las cuales incluyen 20 líneas de dirección, 8 líneas

bidireccionales de datos, 4 líneas de control (IOR, IOW, MEMR, MEMW), seis líneas de petición de interrupción (IRQ2-IRQ7), 3 canales de DMA, refresco de memoria e inserción de estados de espera, y todos los voltajes de la fuente de voltaje. Para proporcionar más flexibilidad a la IBM PC, se asignan 8 conectores a este canal. Cada conector acepta tarjetas de circuito impreso de doble cara, con 31 contactos por lado.

Puesto que esos ocho conectores son iguales, al insertar una tarjeta al bus de expansión se puede usar indistintamente cualquiera de ellos.

2.4.9 La fuente de voltaje

Las especificaciones para la fuente de voltaje para la IBM PC y la IBM PC XT son las siguientes:

Voltaje	Corriente (PC)	Corriente (XT)
+5 VDC	7.00 A	15.0 A
-5 VDC	0.30 A	0.30 A
+12 VDC	2.00 A	4.20 A
-12 VDC	0.25 A	0.25 A

Tabla 2.2 Especificaciones de la fuente de alimentación.

2.5 La computadora IBM PC AT

La introducción de la PC AT marcó el uso de un nuevo microprocesador de Intel, el 80286. Para acomodar el nuevo bus de 16 bits del μ p 80286, la AT extendió el bus de la PC XT a 16 bits para datos y 24 bits para el direccionamiento de datos. La

compatibilidad con productos anteriores se mantuvo esencialmente en los slots de expansión con adaptadores diseñados para la PC y XT. Las señales adicionales del bus fueron añadidas en un segundo conector del slot de expansión. La IBM AT original soportó un reloj de 6 Mhz con un ciclo de espera. Posteriormente, IBM lo incrementó a 8 Mhz. Muchos fabricantes de clones AT han incrementado la velocidad del reloj a 10, 12, 16, 20, ó 25 Mhz. Una AT con una alta velocidad en el reloj y sin ciclos de espera rivaliza con computadoras basadas en el μ p 80386 en muchas aplicaciones.

El μ p 286 soporta dos tipos de operación: modo real y modo protegido (ver apéndice D). El modo real emula el ambiente del 8088/8086 con un espacio de direccionamiento de 1 Mbyte, lo cual permite a los programas de aplicación para PC y XT ejecutar sin problemas en una AT. En modo protegido, se proporciona soporte para ambientes multitarea. El espacio total de direccionamiento de 16 MBytes del 286 y un espacio de direccionamiento virtual de 1 Gbyte por tarea mapeada en el espacio real de 16 Mbytes también están soportados en modo real. Desafortunadamente, debido al poco software en modo protegido, esta potencia en computadoras AT se desaprovechó por largo tiempo.

En adición a la expansión del bus, la AT introdujo también discos flexibles de alta densidad (1.2 Mbytes en discos de 5 1/4 pulgadas) y un disco duro de 20 Mbytes como características estándar. La AT adicionó también un reloj/calendario con batería y un nuevo adaptador de video: el EGA (Enhanced Graphics Adapter), el cual da una mejora notable en resolución y colores en comparación con el CGA.

2.6 Sistemas 386 de bus AT

El siguiente desarrollo mayor fue la introducción del μ p 80386 de 32 bits de Intel. IBM introdujo el 386 en la PS/2, una familia de PC's que usan una nueva arquitectura propietaria llamada Arquitectura de Micro Canal o **MCA** (Micro Channel Architecture). El resto de la industria se quedó con el bus AT y produjo sistemas 386 con buses AT.

Los sistemas de bus AT 386 son simplemente computadoras tipo PC AT, pero con el μ p 386 sustituyendo al 286. El μ p 386 ofrece de 12 a 50 veces la respuesta de la PC original de 4.77 Mhz y de 2 a 10 veces la respuesta de un sistema 286 de 8 Mhz. Estas mejoras se producen aumentando la frecuencia del reloj del sistema, menos pulsos de reloj por ciclo de bus, menos pulsos de reloj por instrucción, nuevas instrucciones para operaciones con 32 bits, y buses de datos de 32 bits. El 386 también soporta los modos reales del 8088, 8086, y 80286. En modo protegido, soporta el modo protegido del 286 y extiende un nuevo modo protegido para el 386.

2.7 Sistemas PS/2

En 1987, IBM presentó una nueva familia de PC's denominadas "familia PS/2". Las siguientes son las características que estos sistemas tienen en común:

- Integración del adaptador de video en la tarjeta del sistema.
- Integración del controlador del diskette en la tarjeta del sistema.

- Integración del puerto serial y puertos paralelos en la tarjeta del sistema.
- Puerto para mouse integrado.
- Uso de unidades de discos de 3 1/2 pulgadas.
- Introducción del nuevo adaptador para gráficas VGA.

Modelo PS/2	µp	Reloj	Memoria base	Expansión de memoria	Bus	NPU
Modelo 25	8086	8 MHz	512 K	640 K	XT	8087
Modelo 30	8086	8 MHz	640 K	640 K	XT	8087
Modelo 30 286	286	10 MHz	512 K	16 M	AT	287
Modelo 50	286	10 MHz	1 M	16 M	MCA	287
Modelo 50z	286	10 MHz (0 WS)	2 M	16 M	MCA	287
Modelo 55SX	386SX	16 MHz	4 M	16 M	MCA	387 SX
Modelo 60	286	10 MHz	1 M	16 M	MCA	287
Modelo 70	386	16,20, 25 MHz	2 M	16 M	MCA	287 / 387
Modelo 70P	386	20 MHz	4 M	16 M	MCA	387
Modelo 80	386	16, 20 MHz	2 M	16 M	MCA	287 / 387

Tabla 2.3 Comparación entre sistemas PS/2

Nota : La abreviación WS se refiere a Wait States (estados de espera).

Las computadoras PS/2 introdujeron para los sistemas basados en 286 y 386 la nueva Arquitectura de Micro Canal, mientras que las basadas en 8086/8088 continuaron usando el bus XT. En la tabla 2.3 se agrupan los diversos modelos de computadoras PS/2 comparándose por tipo de microprocesador, frecuencia de reloj, capacidad de memoria base, expansión de memoria, tipo de bus y tipo de coprocesador matemático (o NPU: Numerical Process Unit).

2.8 Sistemas 486 y Pentium

La sofisticación del software para aplicaciones comerciales y el éxito de las plataformas en modo gráfico, como Windows y Presentation Manager, permitieron el avance hacia sistemas más sofisticados y caros. Los sistemas 486 incorporan al microprocesador 80486 de Intel, el cual consta de alrededor de 1.3 millones de transistores e incluye al coprocesador matemático en el mismo chip, alcanzando velocidades de 33 y 50 MHz. Una versión posterior del chip, el 486DX2, trabaja internamente a una velocidad de 66 MHz y externamente a 33 MHz, lo cual le permite ejecutar 54 MIPS (Millones de Instrucciones Por Segundo). Por otro lado, los sistemas 486 incorporan también un tipo nuevo de bus para el sistema de video: el bus local de video VESA, que opera a la velocidad del microprocesador.

En marzo de 1993 Intel anunció el sucesor del 80486. Contra lo esperado, el nombre para este microprocesador no fue 80586, sino Pentium, un microprocesador de 64 bits con alrededor de 3.2 millones de transistores que alcanza velocidades de 60 y 66 MHz, con una ejecución de 112 MIPS. Una de sus características más notables es que incluye dos unidades aritmético lógicas y tecnología superescalar, lo cual le permite ejecutar dos

instrucciones en un ciclo de reloj. Los fabricantes aseguran que en un futuro se podrá contar con computadoras Pentium que soporten dos de estos microprocesadores a bordo, con objeto de aprovechar la característica de Windows NT y OS/2 de soportar sistemas multiprocesadores. A la fecha, los sistemas Pentium arrojan un rendimiento superior al 50 u 80% cuando se les compara con las computadoras 486 más veloces.

2.9 Sumario

En este capítulo se introdujeron conceptos fundamentales del hardware de la IBM PC. Los componentes básicos son el monitor (además de una tarjeta adaptadora), el teclado y la unidad del sistema. La unidad del sistema consta de la tarjeta del sistema o motherboard, las ranuras para los canales de E/S, las unidades de disco, y la fuente de voltaje. La tarjeta del sistema contiene al μ p Intel 8088, memoria y los chips de soporte requeridos para la lógica de control.

Una vez que se han establecido globalmente las bases de hardware del sistema, así como las capacidades de éste, se pueden analizar de manera individual las partes involucradas en el diseño de una interfase de comunicación de datos con la PC. A partir de lo anteriormente visto, es evidente que la parte más directamente afectada es el bus de expansión de la IBM PC, el cual se analizará en el siguiente capítulo.

Capítulo

3

Arquitecturas de bus en sistemas XT y AT

3.1 Introducción

Este capítulo examina las arquitecturas del bus de E/S existentes: ISA, EISA y MCA. De estas tres, ISA, que se remonta a los orígenes de computadora IBM PC original, es la que se analizará más profundamente, ya que es la arquitectura en la que se basa el diseño del sistema descrito en la parte tres de este documento. El capítulo comienza analizando las arquitecturas presentes y continúa con la descripción del bus de E/S en sistemas XT y AT, detallando las señales presentes en el bus y los ciclos de bus que existen.

3.2 Arquitecturas de bus

En la figura 2.1 puede observarse en la tarjeta madre de una IBM PC un bloque de 8 ranuras de expansión. Estas son ranuras que permiten conectar diferentes tipos de circuitos a la computadora. Es común que por lo menos dos de ellas estén ocupadas y el resto estén disponibles para el usuario. Una de esas ranuras la ocupa una tarjeta de circuito impreso que contiene al manejador del monitor y el puerto de la impresora, y la otra contiene al controlador de unidades de disco. Puesto que existen otros tipos de tarjetas que pueden conectarse a una computadora (puertos seriales, puertos de juegos, controladores para disco duro, tarjetas para redes, etc.), hay un estándar para las ranuras de expansión

(o *slots* de expansión). Existen tres tipos de bus para ellas: **ISA** (Industry Standard Architecture: Arquitectura Estándar de la Industria), **EISA** (Enhanced Industry Standard Architecture: Arquitectura Mejorada Estándar de la Industria) y **MCA** (Micro Channel Architecture). Este último es exclusivo de computadoras AT IBM. Dado que el diseño de éste bus es de ésta empresa, el uso del bus MCA implica pagar altas licencias por el uso de recursos patentados por IBM.

El bus PC/XT/AT o bus ISA se remonta al bus de expansión de la computadora personal original. Soporta un bus de datos de 8 o 16 bits y un bus de direcciones de 20 o 24 bits. Con el advenimiento de computadoras más rápidas, como la AT, se hizo necesario que el bus fuera capaz de manejar altas velocidades. La respuesta de IBM fue la arquitectura de microcanal o MCA, que tiene como característica principal su soporte para bus de datos y direcciones de 32 bits y su incompatibilidad con sistemas ISA. Por otro lado, un consorcio de fabricantes de equipos compatibles con IBM decidió desarrollar una versión propia que sí apoyara a los sistemas con bus ISA existentes. El producto de esto fue el bus EISA, que ofrece la potencia del MCA sin sus desventajas, y además acepta las tarjetas ISA que ya existen en el mercado. En la actualidad, MCA ha sido mejorado y es difícil definir cual de estos dos es preferible. Sin embargo, la arquitectura ISA es aún la mejor elección para sistemas de un solo usuario o de redes de computadoras de hasta una docena de terminales, en donde el trabajo del bus no es tan pesado.

La mayoría de los sistemas con arquitectura de bus AT o ISA constan en realidad de dos arquitecturas de bus. Por un lado, se pueden conectar dispositivos de E/S de 8 o 16 bits en el bus de expansión, mientras que por otro la memoria puede encontrarse en un bus local asíncrono (es decir, con diferente velocidad de reloj), junto con el microprocesador. A este bus se le suele llamar bus

local. De acuerdo con este esquema de buses separados, el bus ISA puede ser utilizado en sistemas 80386 y 80486 de 32 bits más rápidos: en un bus rápido pueden encontrarse el microprocesador, la memoria y el adaptador de vídeo y en un bus de E/S más lento se pueden encontrar dispositivos como adaptadores para LAN, disco duro, comunicaciones e impresora.

El bus ISA consta de dos bloques de contactos: uno de 62 y otro de 36, distribuidos en ambas caras del conector. Existe una separación de 0.1 pulgadas entre cada contacto y una altura de 0.31 pulgadas. La existencia de estos dos bloques obedece a la compatibilidad de buses para computadoras XT y AT. Las computadoras XT requieren buses de datos de 8 bits, incluidos en el bloque de 62 contactos, mientras que las computadoras AT con bus de datos de 16 bits usan los dos bloques.

La tabla 3.1 muestra las características de cada una de éstas tecnologías de buses y la figura 3.1 muestra la disposición de los contactos en las tarjetas ISA, EISA y MCA.

Bus	Ancho en bits	Trasferencia por DMA	Velocidad de control de bus	Velocidad de reloj
ISA	16	1.5 Mb/s	No aplica	8 Mhz
MCA	32	5 Mb/s	40 Mb/s	10 Mhz
EISA	32	33 Mb/s	32 Mb/s	8 Mhz

Tabla 3.1 Rendimiento de los buses

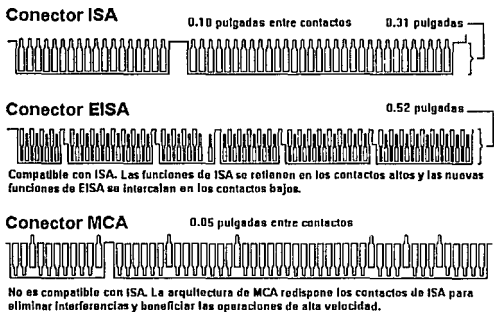


Fig 3.1 Conectores en tarjetas ISA, EISA y MCA

3.3 Señales del bus de expansión en sistemas XT

En computadoras tipo XT la ranura del bus de expansión consta de solo uno de los bloques de contactos que forman al bus ISA. Visto desde el frente de la computadora, como en la figura 2.1, para cada ranura de expansión la hilera de contactos de la derecha es denominada "A" y la de la izquierda "B". La numeración de los contactos es, desde la parte trasera de la computadora hasta el frente, de 1 a 31.

Las señales de la tarjeta madre disponibles en el bus de expansión de una IBM PC XT se indican en la figura 3.2.

Cada una de las señales, excepto en las de polarización de - 5 V \pm 12 V y tierra, tienen un nivel lógico TTL.

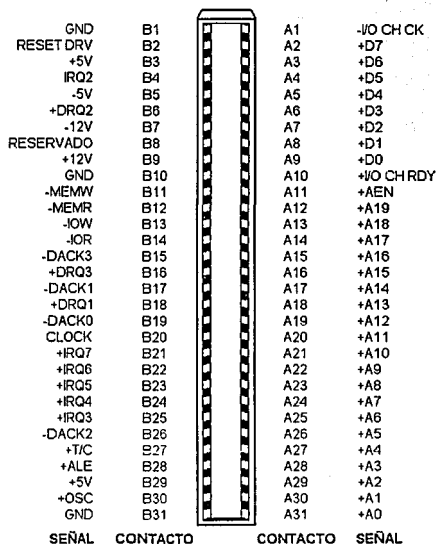


Figura 3.2 Señales en la ranura de expansión del bus XT

OSC

La señal **OSC** (oscillator - oscilador) es una señal de salida con una frecuencia de 14.31818 MHz y un período de aproximadamente 70 ns. Tiene un ciclo de trabajo cercano al 50%. La frecuencia de la señal **OSC**, la más alta en el bus, puede ser ajustada mediante un capacitor variable que se encuentra generalmente en la tarjeta madre del sistema.

CLK

La señal **CLK** (clock - reloj) es una señal de salida que se deriva de la señal **OSC** para usarla como reloj para el microprocesador 8088. La señal **CLK** se obtiene dividiendo entre 3 la señal **OSC**, lo cual arroja una velocidad de 4.77 MHz. La señal **OSC** no es simétrica, lo cual puede causar problemas si se cambia el microprocesador 8088 por uno de la serie V20 de NEC, ya que éste microprocesador requiere alimentación de reloj simétrica.

RESET DRV

RESET DRV (reset driver - restaurar manejador) es una señal de salida que se mantiene activa en estado alto durante secuencias de arranque, cuando se proporciona la alimentación de voltaje.

A0-A19

Las señales **A0** a **A19** (address - direcciones) son señales de sólo salida usadas para direccionar memoria y puertos.

D0-D7

Las señales **D0-D7** (data- datos) son bidireccionales y se usan para transmitir los datos entre el microprocesador, memoria y puertos de E/S.

ALE

La señal **ALE** (Address Latch Enable - habilitador del candado de direcciones) es una señal de salida que indica el momento en que el bus de direcciones es válido, habilitando el comienzo de un ciclo de bus.

I/O CH CK

La señal **I/O CH CK** (I/O Channel Check - verificador del canal de E/S) es una señal de entrada activa en bajo nivel. Se usa para reportar condiciones de error en tarjetas insertadas en las ranuras de expansión. Cuando esta señal es puesta en nivel bajo, genera una interrupción no mascarable (NMI) en el microprocesador 8088.

I/O CH RDY

La señal **I/O CH RDY** (I/O channel ready - canal de E/S listo) es una señal de entrada usada para extender la longitud de los ciclos de bus de tal forma que la memoria o puertos de E/S que no son lo suficientemente rápidos para responder a un ciclo de bus normal de cuatro pulsos de reloj puedan conectarse al sistema.

IRQ2-IRQ7

Las señales **IRQ2** hasta **IRQ7** (interrupt requests -peticiones de interrupciones) son seis señales de entrada; son usadas por el bus del sistema para generar peticiones de interrupción en el microprocesador 8088. Estas señales van directamente al controlador de interrupciones 8259A. BIOS inicializa al controlador de tal manera que **IRQ2** es la señal de más alta prioridad e **IRQ7** es la de más baja prioridad.

IOR

La señal **IOR** (I/O read - lectura de E/S) es una señal de salida del controlador de bus 8288 que se usa para indicar a los puertos de E/S que el ciclo de bus iniciado por el 8088 es un ciclo de lectura de puerto y que la dirección en el bus es una dirección de puerto.

IOW

La señal **IOW** (I/O write - escritura de E/S) es una señal de salida activa en bajo nivel. Indica que el bus de direcciones contiene una dirección de puerto de E/S y que el bus de datos contiene datos que se van a escribir en el puerto de E/S.

MEMW

MEMW (memory write - escritura de memoria) es una señal activa en bajo nivel. Indica que el bus de direcciones contiene una dirección de memoria en donde los datos que se encuentran en el bus de datos se van a escribir.

MEMR

La señal **MEMR** (memory read - lectura de memoria) es una señal de salida activa en bajo nivel. Indica que el bus de direcciones contiene una dirección válida para realizar lectura de memoria.

DRQ1-DRQ3

Las líneas **DRQ1** a **DRQ3** (direct memory access request 1 - 3 - petición de acceso directo a memoria 1 a 3) son señales de entrada activas en nivel alto. Son usadas por alguna interfase conectada al bus de expansión para pedir ciclos de datos (lectura y escritura) mediante acceso directo a memoria, sin intervención del microprocesador.

DACK0-DACK3

Las señales **DACK0-DACK3** (DMA acknowledge 0 - 3 - reconocimiento de DMA 0 a 3) son señales de salidas activas en bajo nivel. Son emitidas por el controlador de DMA, e indican que

el correspondiente DRQ ha sido aceptado y que el controlador de DMA tomará el manejo del bus y procederá con el ciclo de DMA pedido.

AEN

la señal **AEN** (address enable - habilitador de dirección) es una señal de salida activa en nivel alto. Es emitida por la lógica de control de DMA, e indica que un ciclo de bus de DMA está en progreso.

TC

La señal **TC** (terminal count - conteo terminal) es una señal de salida activa en nivel alto, emitida por el controlador de DMA. Indica que uno de los canales de DMA ha alcanzado su número preprogramado de ciclos de transferencia. Se usa para saber cuando ha terminado una transferencia de bloque por DMA.

3.4 Operaciones en el bus de expansión en sistemas XT

En el bus del sistema, los datos se transfieren en un ciclo de bus, el cual consta de por lo menos cuatro pulsos de reloj (llamados también ciclos T). Las dos clasificaciones generales del ciclo de bus son el ciclo de bus controlado por el 8088 y el controlado por DMA. En las siguientes secciones se describen estos ciclos de bus y cómo son usados para transferir información entre memoria, E/S, y el microprocesador. Cuando se especifiquen velocidades, es necesario tomar en cuenta que trabajar en una computadora más rápida no implica que la velocidad manejada en el bus de

expansión sea mayor, ya que el microprocesador se encuentra en el bus local.

3.4.1 Ciclos de bus controlados por el microprocesador

Cuando el microprocesador 8088 genera un ciclo de bus, entrega al bus de direcciones del sistema una localidad de memoria o puerto de E/S, controla la dirección del flujo de datos y es la fuente o el destino de los datos. Uno de cinco diferentes tipos de ciclos de bus se generan:

- Ciclo de bus de lectura de memoria.
- Ciclo de bus de escritura de memoria.
- Ciclo de bus de lectura de puerto de E/S.
- Ciclo de bus de escritura de puerto de E/S.
- Ciclo de bus de reconocimiento de interrupciones.

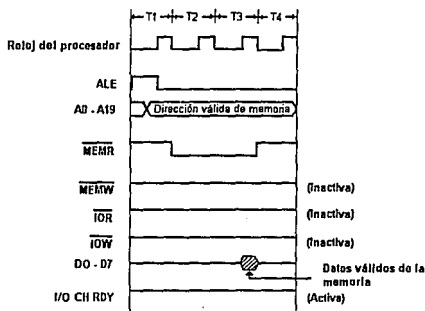
El ciclo de bus de reconocimiento de interrupciones no se cubrirá aquí porque ocurre solamente en el sistema de bus local.

3.4.1.1 Ciclo de bus de lectura de memoria

Este ciclo es usado para decodificar instrucciones y datos para el sistema de memoria. Esta memoria puede estar en la tarjeta del sistema, en tarjetas conectadas al bus del sistema, en ROM o en RAM.

El ciclo de lectura de memoria requiere al menos cuatro ciclos de reloj, de tal manera que para un sistema de 4.77 MHz la longitud mínima del ciclo de reloj es de 840 ns, mientras que para un sistema de 10 MHz el mínimo es de 400 ns. Estas longitudes

mínimas pueden extenderse cuando un dispositivo lento solicita un estado de espera, alargando al ciclo completo lo suficiente como para que pueda operar tal dispositivo.



$$T1 = T2 = T3 = T4 = 210 \text{ ns}$$

$$\text{Duración del ciclo} = 4 \times 210 \text{ ns} = 840 \text{ ns}$$

Figura 3.3 Diagrama de tiempos del ciclo de bus de lectura de memoria

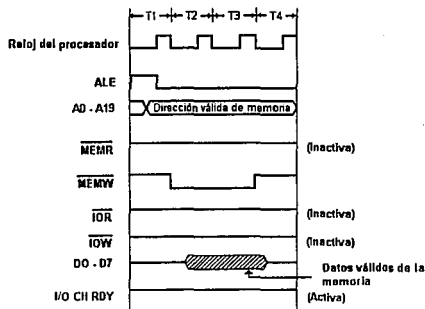
La figura 3.3 muestra el diagrama de tiempos y las señales que se presentan en este ciclo. El ciclo de lectura de memoria comienza durante el pulso T1 con la señal ALE pasando a estado activo. El estado lógico "1" de esta señal indica que el bus de direcciones contiene una dirección válida. A continuación, en el pulso de reloj T2, la señal del bus MEMR se activa. Esto indica a los adaptadores de memoria conectados al bus que el ciclo es de lectura de memoria. También indica que si el adaptador contiene memoria con una dirección correspondiente con la que se encuentra en el bus de datos, el adaptador deberá entregar al bus

de datos el contenido de la dirección apropiada. Al comienzo del pulso T4, el microprocesador 8088 captura los datos del bus. Poco tiempo después de que comienza el pulso T4, la señal MEMR del bus es desactivada. Al final del pulso T4, el ciclo de bus termina.

3.4.1.2 Ciclo de bus de escritura de memoria

El ciclo de escritura de memoria es usado en cualquier momento en que una instrucción del 8088 indica que se escriba un dato en una localidad de memoria. La figura 3.4 muestra el diagrama de tiempos del ciclo de escritura de memoria. Al igual que con el ciclo de lectura de memoria, puede insertarse un estado de espera si es necesario.

Durante el pulso de reloj T1, la señal ALE se activa, indicando que existe una dirección válida de memoria en el bus de direcciones. Después, en el ciclo T2, la señal del bus MEMW se activa, lo cual quiere decir que el ciclo de bus es de escritura de memoria. Poco después de que ésta señal se activa, el 8088 envía el dato que va a ser escrito en la memoria. En T4 la señal MEMW se desactiva. Al final de T4 el ciclo ha terminado.



$T1 = T2 = T3 = T4 = 210 \text{ ns}$
 Duración del ciclo = $4 \times 210 \text{ ns} = 840 \text{ ns}$

Figura 3.4 Diagrama de tiempos del ciclo de bus de escritura de memoria

3.4.1.3 Ciclo de bus de lectura de puerto de E/S

Este ciclo se inicia cada vez que el microprocesador ejecuta una instrucción IN. El ciclo de lectura de puerto de E/S es similar al de lectura de memoria. Su propósito es traer los datos de una de las direcciones de puerto de E/S. En el diseño de la PC, el ciclo de bus consta siempre de un mínimo de cinco pulsos de reloj, es decir, aproximadamente 1.5 microsegundos para una computadora de 4.77 MHz o 0.5 microsegundos para una computadora de 10 Mhz. Durante una lectura de puerto de E/S el microprocesador 8088 entrega una dirección de puerto de 16 bits. En este ciclo de bus los cuatro bits más significativos del bus de direcciones nunca se activan. La figura 3.5 muestra el diagrama de tiempos correspondiente.

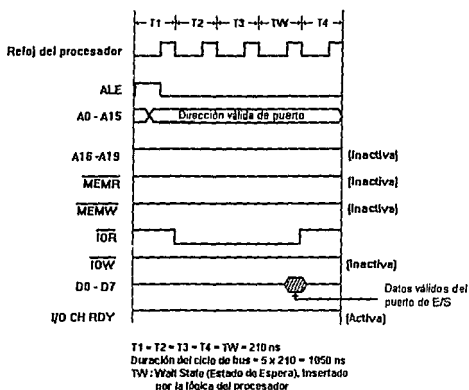


Figura 3.5 Diagrama de tiempos del ciclo de bus de lectura de puerto de E/S

3.4.1.4 Ciclo de bus de escritura de puerto de E/S

Un ciclo de escritura de puerto de E/S se inicia cada vez que el microprocesador 8088 ejecuta una instrucción OUT. La función de este ciclo es la de escribir en una dirección de puerto específica un dato proveniente del 8088. Normalmente este ciclo de bus requiere cuatro pulsos de reloj, pero el diseño de la PC automáticamente inserta un estado de espera extra llamado pulso TW. Esto implica que la longitud de este ciclo en tiempo es de 1.05 microsegundos para una computadora de 4.77 MHz. En computadoras más rápidas es común que se agreguen estados de espera adicionales para ajustar la longitud del ciclo a 1 microsegundo. esto garantiza que diseños de tarjetas adaptadoras de 4.77 MHz trabajen en computadoras más rápidas.

Solamente los bits 0-15 del bus de direcciones son usadas para las direcciones de los puertos de E/S. La figura 3.6 muestra el diagrama de tiempos de este ciclo. Como con los otros ciclos, en T1 la señal ALE se activa para indicar que hay una dirección válida en el bus conteniendo una dirección válida de puerto. Luego, en T2, la señal IOW se activa, indicando que se está realizando un ciclo de escritura de puerto de E/S. Poco después del ciclo T2, el microprocesador entrega los datos al bus para que el puerto correspondiente los reciba. Al comienzo de T4 se desactiva la señal IOW. Al final de T4, el ciclo ha terminado.

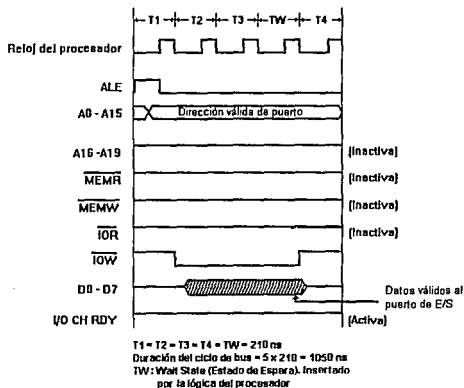


Figura 3.6 Diagrama de tiempos del ciclo de bus de escritura de puerto de E/S

3.4.2 Ciclos de bus controlados por DMA

La segunda clasificación general de los ciclos de bus es la de los dirigidos por el controlador de DMA (el circuito 8237-5). Durante las operaciones de Acceso Directo a Memoria (DMA), el microprocesador cede el control de los buses al controlador de DMA. Este se encarga de entregar una dirección de memoria al bus de direcciones y controla el flujo de datos entre una interfase conectada al bus de expansión y la memoria. Debe notarse que en los ciclos de bus de DMA, los datos no son controlados directamente por el controlador de DMA, sino que se transfieren directamente entre la interfase y la memoria.

Dos tipos de ciclos se generan en este caso. El primero de ellos es un ciclo en el que se leen los datos de una interfase conectada al sistema y se escriben en una localidad de memoria especificada por el controlador de DMA. El segundo tipo de ciclo es el caso opuesto, ya que se leen los datos de la memoria y se escriben en la interfase.

Un ciclo de bus DMA puede ser iniciado desde el bus del sistema activando cualquiera de las señales DRQ1, DRQ2 o DRQ3. Al comenzar un ciclo DMA, se activa la señal AEN seguida por una señal DACK0, DACK1, DACK2 o DACK3. Un ciclo normal de DMA se lleva 5 pulsos de reloj. En el diseño de la PC se ha añadido un pulso de reloj adicional, alargando el ciclo de DMA a 6 pulsos de reloj, o 1.26 microsegundos para una computadora de 4.77 Mhz.

Cuando el controlador de DMA tiene el control del bus del sistema, sus señales de temporización se generan de la misma señal de reloj que usa el microprocesador 8088, o de esta misma señal dividida entre dos para sistemas más rápidos. Debido a que el controlador de DMA genera temporizaciones para el bus

ligeramente diferentes, los pulsos de reloj usados para DMA son llamados ciclos S.

Durante el período de tiempo en el que el controlador de DMA no se encuentra ejecutando un ciclo de bus, se encuentra en un estado de ocio, ejecutando continuamente un estado de reloj S1 (un ciclo de ocio) mientras busca peticiones de DMA en el sistema. En el momento en que una petición se detecta, el controlador envía una señal al microprocesador pidiéndole que se retire del bus. El controlador de DMA va al estado del pulso de reloj S0, en el cual continuamente busca la respuesta del microprocesador indicando que el bus está libre y de que puede comenzar un ciclo DMA.

Al recibir el controlador la señal HLDA (hold acknowledge) por parte del microprocesador, entra al estado de reloj S1, señalizando el comienzo del ciclo de DMA. El controlador de DMA procede entonces de acuerdo a lo mostrado en las figuras 3.7 y 3.8.

3.4.2.1 Ciclo de bus de escritura de memoria por DMA

El propósito de un ciclo de bus de escritura de memoria por DMA es traer datos de un adaptador de interfase conectado a un puerto de la computadora y escribirlos en la localidad de memoria especificada por el controlador de DMA. después de que el ciclo de DMA ha iniciado, el controlador y sus circuitos de soporte entregan al bus de direcciones la localidad de memoria de destino. Después, la señal IOR se activa, indicando que el adaptador de interfase que solicitó el ciclo de DMA debe entregar los datos al bus.

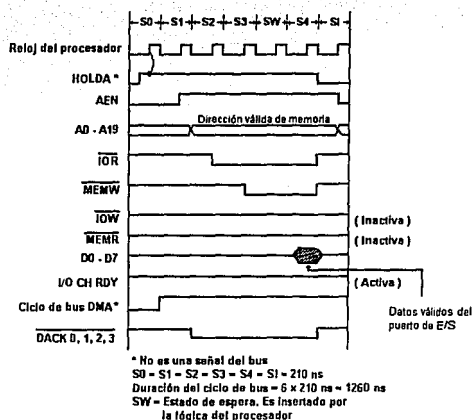


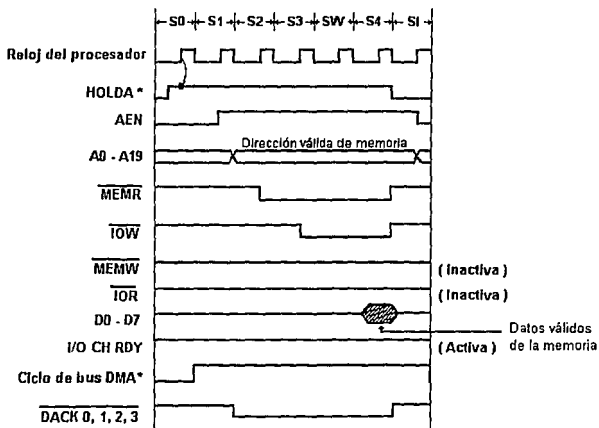
Figura 3.7 Diagrama de tiempos del ciclo de bus de escritura de memoria por DMA

A continuación, la señal MEMW se activa. Esto indica que la memoria direccionada debe de tomar el dato y almacenarlo. Es responsabilidad del adaptador mantener los datos válidos hasta que la memoria pueda realizar la operación de escritura. La figura 3.7 ilustra las temporizaciones usadas en este ciclo de bus.

3.4.2.2 Ciclo de bus de lectura de memoria por DMA

El ciclo de lectura de memoria por DMA es usado para transferir datos desde la memoria del sistema a un adaptador de interfase. Después de que el ciclo de DMA es iniciado, el controlador de

DMA y sus circuitos de soporte entregan una dirección de memoria al bus. Una vez hecho esto, el controlador activa la señal MEMR, indicando así a la memoria que debe entregar los datos de la localidad direccionada al bus de datos. Después, el controlador activa la señal IOW, indicándole al adaptador de interfase que deberá tomar los datos provenientes de la memoria. La figura 3.8 ilustra la temporización básica del ciclo de lectura de memoria por DMA.



* No es una señal del bus
 S0 - S1 - S2 - S3 - S4 - S11 = 210 ns
 Duración del ciclo de bus = $6 \times 210 \text{ ns} = 1260 \text{ ns}$
 SW = Estado de espera. Es insertado por la lógica del procesador

Figura 3.8 Diagrama de tiempos del ciclo de bus de lectura de memoria por DMA

3.5 Señales en el bus de expansión AT

La computadora IBM PC AT original soporta un total de ocho ranuras de expansión, al igual que la IBM PC XT. Dos de esas ranuras son de 8 bits, cada una con dos hileras de 62 contactos, al igual que la PC XT. Las seis ranuras restantes son de 16 bits. Cada una de ellas consta del bloque de 62 contactos que tiene el bus XT, y de un bloque de 36 contactos que contiene nuevas señales para el bus. La figura 3.9 muestra el aspecto de la tarjeta madre AT en la zona de las ranuras de expansión y la figura 3.10 muestra las señales en el bus AT.

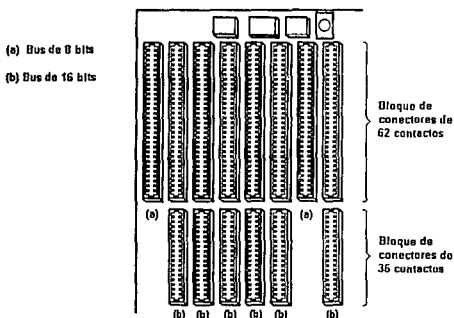


Figura 3.9 Ranuras de expansión en la IBM PC AT

El conector compatible con el bus XT tiene cuatro cambios:

- La señal no usada en B8 es llamada ahora 0WS (zero wait state - cero estados de espera). Esta es

una señal activa en bajo nivel usada para forzar al actual ciclo de memoria a no insertar estados de espera adicionales.

- La señal IRQ2 en el contacto B4 se convierte en IRQ9. El nivel 2 de interrupciones se usa ahora para aceptar interrupciones del segundo controlador de interrupciones incluido en el diseño de la PC AT.
- Las señales MEMR y MEMW, de los contactos B12 y B11 respectivamente, se convierten en SMEMR y SMEMW. Estas señales se activan solamente cuando la dirección de memoria usada esta por debajo de la dirección 1 MB (FFFF Hex).
- La señal CLK en el contacto B20 ya no es de 4.787 MHz. Este contacto refleja ahora la velocidad del reloj que alimenta al microprocesador, una fracción de éste, o un reloj completamente independiente.

El resto de los cambios para una computadora AT residen en el conector de 36 contactos. Estas nuevas señales se describen a continuación.

SBHE

La señal **SBHE** (system bus high enable - habilitador de la parte alta del bus del sistema) es una señal de salida. Indica que el dispositivo conectado al bus espera transferir datos en el byte más significativo del bus de datos, en las líneas SD8-SD15. Esta señal y el bit A0 del bus de direcciones pueden ser usados para decodificar el tipo de ciclo de bus, tal como lo muestra la tabla 3.2.

Señal	Conector	Conector	Señal
GND	B1	A1	-1/0 CH CK
RESET DRV	B2	A2	SD7
+ 5 V DC	B3	A3	SD6
RQ0	B4	A4	SD5
- 5 V DC	B5	A5	SD4
DRQ2	B6	A6	SD3
- 12 V DC	B7	A7	SD2
0 VDS	B8	A8	SD1
+ 12 V DC	B9	A9	SD0
GND	B10	A10	-1/0 CH RDY
-SMBW	B11	A11	ADR
-SMBR	B12	A12	SA19
-IOW	B13	A13	SA18
-IOR	B14	A14	SA17
-DACK3	B15	A15	SA16
DRQ3	B16	A16	SA15
-DACK1	B17	A17	SA14
DRQ1	B18	A18	SA13
-REFRESH	B19	A19	SA12
CLK	B20	A20	SA11
RQ7	B21	A21	SA10
RQ6	B22	A22	SA9
RQ5	B23	A23	SA8
RQ4	B24	A24	SA7
RQ3	B25	A25	SA6
-DACK2	B26	A26	SA5
T / C	B27	A27	SA4
BALE	B28	A28	SA3
+ 5 V DC	B29	A29	SA2
OSC	B30	A30	SA1
GND	B31	A31	SA0
<hr/>			
-MEM CS16	D1	C1	SBHE
-1/0 CS16	D2	C2	LA23
RQ10	D3	C3	LA22
RQ11	D4	C4	LA21
RQ12	D5	C5	LA20
RQ15	D6	C6	LA19
RQ14	D7	C7	LA18
-DACK0	D8	C8	LA17
DRQ0	D9	C9	-MEMR
-DACK5	D10	C10	-MEMW
DRQ5	D11	C11	SD08
-DACK6	D12	C12	SD09
DRQ6	D13	C13	SD10
-DACK7	D14	C14	SD11
DRQ7	D15	C15	SD12
+ 5 V DC	D16	C16	SD13
-MASTER	D17	C17	SD14
GND	D18	C18	SD15

Figura 3.10 Señales en la ranura de expansión del bus AT

SBHE	A0	Función
1	0	Transferencia de 16 bits
1	1	Transferencia del byte más significativo
0	0	Transferencia del byte menos significativo
0	1	Inválida

Tabla 3.2 Decodificación del bus de datos

MEMR

La señal **MEMR** (memory read - lectura de memoria) es idéntica a la señal del mismo nombre del bus XT, excepto que es activada en todos los ciclos de lectura de memoria, incluyendo aquellos arriba del rango de 1 MB. Si **SMEMR** está inactiva y **MEMR** está activa, entonces el acceso a memoria es en una localidad arriba de la dirección 1 MB (FFFF Hex).

MEMW

La señal **MEMW** (memory write - escritura a memoria) es idéntica a la señal del mismo nombre en el bus XT, excepto que es activada en todos los ciclos de escritura de memoria, incluyendo aquellos arriba del rango de 1 MB.

DRQ5-DRQ7

Las señales **DRQ5-DRQ7** (DMA request 5-7 - petición de DMA 5-7) son similares a las señales **DRQ0-DRQ4** en el bus de expansión XT, con la diferencia de que estas señales solicitan transferencias de datos de 16 bits por DMA.

DACK5-DACK7

Las señales **DACK5-DACK7** (DMA acknowledge 5-7 - reconocimiento de DMA 5-7) son equivalentes a las señales **DACK0-DACK3** presentes en el bus de expansión XT, excepto que éstas reconocen transferencias de DMA de 16 bits.

MEM CS16

La señal **MEM CS16** es una señal de entrada activa en bajo nivel. Se usa para indicar que la tarjeta en la ranura de expansión puede soportar un estado de espera en el presente ciclo de bus, con una transferencia de datos de 16 bits.

I/O CS16

La señal **I/O CS16** (input/output select 16 - selector de entrada/salida de 16 bits) es una señal de entrada activa en bajo nivel. Se usa para indicar que la tarjeta en la ranura de expansión puede soportar un estado de espera en el presente ciclo de bus de E/S 16 bits.

MASTER

La señal **MASTER** es una señal de entrada activa en bajo nivel. Se usa con las señales DRQ5-DRQ7 para permitir a un bus maestro que tome al bus del sistema. Cuando se regresa una señal DACK de una activación DRQ en los canales 5 a 7, la señal **MASTER** puede ser activada, cancelando las operaciones del canal de DMA, poniendo en tercer estado los buses de datos, direcciones y control. Esto permite a una tarjeta en la ranura de expansión tomar el control de los dispositivos conectados al bus y la memoria.

IRQ10, IRQ11, IRQ12, IRQ14, IRQ15

Estas son señales de entrada activas en nivel alto que son prácticamente idénticas a las señales IRQ2-IRQ7 en el bus XT. Son usadas para generar peticiones de interrupción al controlador de interrupciones.

LA17-LA23

Las señales **LA17-LA23** (unlatched addresses 17-23 -direcciones no aseguradas 17-23) son señales de salida usadas para proporcionar información de la dirección de memoria en el presente ciclo de bus.

SD8-SD15

Las señales **SD8-SD15** (system data bus bits 8-15 - bus de datos del sistema, bits 8-15) son ocho señales bidireccionales que soportan la transferencia de datos en el byte más significativo de las transferencias de 16 bits. Todos los dispositivos de 8 bits conectados a la ranura de expansión AT deben usar las señales SD0-SD7, ya que si el software del sistema pide una transferencia de 16 bits a un dispositivo de 8, la tarjeta AT del sistema obliga a dos ciclos de bus en los bits menos significativos SD0-SD7. Las tarjetas que se conecten al sistema deben de indicar su capacidad de soportar transferencias de 16 bits activando la señal MEM CS16 para el caso de memoria y la señal I/O CS16 para puertos.

+5VCD , GND

Las señales **+5VCD** y **GND** Son señales de 5 volts y tierra, respectivamente.

3.6 Dimensiones de tarjetas en el bus de expansión XT y AT

La figura 3.11 muestra las dimensiones máximas que pueden tener las tarjetas que se pueden conectar en la ranuras de expansión XT. La tarjeta debe tener al menos 1.52 mm (0.06 pulgadas) de ancho para asegurar buen contacto entre los conectores.

La ranura de expansión AT puede soportar una tarjeta más alta que la de un sistema XT. El área de los conectores no fue expandida para mantener la compatibilidad mecánica con sistemas XT. La figura 3.12 muestra las dimensiones máximas en una tarjeta AT.

Unidades: pulgadas

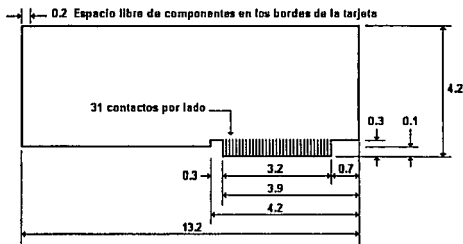


Figura 3.11 Dimensiones de la tarjeta de expansión en un bus tipo XT

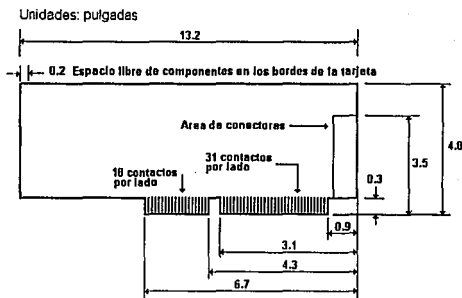


Figura 3.12 Dimensiones de la tarjeta de expansión en un bus tipo AT

3.7 Sumario

Una de las piezas clave para el diseño de una interfase con la IBM PC es el conocimiento de los ciclos de bus y las señales del canal de E/S. En este capítulo se hizo un análisis al bus de expansión en una computadora personal, comenzando por una descripción de las arquitecturas de bus disponibles: ISA, EISA y MCA. Posteriormente, se describieron las señales en el bus XT, así como los ciclos de bus en el sistema y las señales en un bus AT.

Los datos se transfieren en el sistema por medio de ciclos de bus. Existen dos tipos: los controlados por el microprocesador y los controlados por DMA. Los controlados por el microprocesador son el ciclo de reconocimiento de interrupciones y los ciclos de lectura y escritura a memoria y puertos. Los ciclos controlados por DMA son dos: lectura de memoria - escritura en puerto y lectura de puerto - escritura en memoria.

En el diseño de la interfase con la IBM PC se deben considerar los siguientes puntos: la velocidad del dispositivo conectado, el seguimiento de los ciclos de bus que maneja el sistema y no sobrecargar al bus.

Una vez que se ha examinado la estructura de los buses de expansión y los ciclos de transferencia de datos en la IBM PC, se finalizará el análisis de la computadora personal discutiendo el sistema de interrupciones y el acceso directo a memoria en el siguiente capítulo.

Capítulo 4

Interrupciones y DMA

4.1 Introducción

Este capítulo presenta las características y capacidades del sistema de interrupciones y de DMA (acceso directo a memoria) de la IBM PC. El objetivo de esto es hacer un análisis de las características de estos procesos en busca del mejor método a usar durante el diseño de la comunicación de los datos que este proyecto implica.

4.2 El proceso de interrupción

Se puede pensar en una computadora como una *máquina de estados finitos*, es decir, un sistema con varios estados únicos y definidos. La ejecución de un programa puede ser descrita como una secuencia de estados de máquina S_0, S_1, \dots, S_n . Si en algún momento del programa una acción prioritaria requiere suspender tal secuencia de estados para la realización de otra tarea, se solicita una *interrupción*. Al generarse una interrupción se suspende temporalmente la ejecución de la secuencia normal de estados mientras es ejecutada una rutina de servicio de interrupción (ISR: Interrupt Service Routine), cuya dirección se encuentra almacenada en una tabla llamada *tabla de vectores de interrupción* (en este contexto, un vector es una dirección). Para regresar al estado previo a la petición de interrupción, el estado del microprocesador debe haber sido guardado en algún lugar, así

como el valor del contador del programa (PC - program counter) Esto lo hace el microprocesador enviando estos datos a la pila (o stack) del sistema cuando la interrupción es reconocida. La ISR debe salvar todos los registros del microprocesador que serán cambiados y regresarlos a su valor original antes de terminar.

Los cuatro aspectos a considerar para evitar errores al usar interrupciones son:

- 1) Interrupción de procesos críticos. Esto implica a procesos específicos o tareas que causarían un comportamiento indeseable o impredecible si son interrumpidas. Un ejemplo de esto ocurre cuando los registros de segmento son cambiados. Una protección contra esto es deshabilitar las interrupciones durante la realización de éstas tareas.
- 2) Deshabilitación de interrupciones. Puede causar que se pierdan interrupciones durante el tiempo en que sean deshabilitadas, de modo que debe hacerse solo cuando sea necesario y durante poco tiempo.
- 3) Prioridades de interrupción. Puesto que diferentes dispositivos pueden solicitar una interrupción simultáneamente, debe contarse con un sistema con niveles de prioridad, de tal manera que se pueda decidir si una interrupción puede interrumpir a otra en ejecución.
- 4) Duración de la interrupción. En los eventos en que la temporización sea crítica, debe considerarse el tiempo en que una interrupción es reconocida y ejecutada.

4.3 Tipos de interrupción

Las peticiones de interrupción usualmente asociadas con dispositivos externos al microprocesador, tal como unidades de disco, teclado, impresoras y temporizadores son interrupciones de hardware. A esta categoría pertenecen también las generadas por eventos en la tarjeta del sistema que informan al microprocesador que una falla importante se ha generado, tal como una error de paridad de memoria o una falla inminente de la alimentación principal de voltaje.

Otro tipo de interrupción es el de las generadas dentro del procesador mismo, ya sea por un error de software, o por petición misma del programa en ejecución. Estas son llamadas interrupciones de software.

Una petición de interrupción que puede ser ignorada por el procesador es llamada interrupción mascarable. Estas interrupciones pueden ser activadas y desactivadas por instrucciones en código máquina del procesador. Por otro lado, existen líneas de petición de interrupción que siempre generan una interrupción cuando son activadas. Este tipo de interrupción es llamado interrupción no mascarable.

4.3.1 Interrupciones de hardware

Los dispositivos externos usan o un nivel de voltaje o un cambio en el nivel de voltaje para registrar una petición de interrupción. El método que el dispositivo use depende de que el microprocesador tenga una línea de interrupciones sensible al nivel de voltaje o al flanco de subida o bajada. Si el microprocesador tiene una línea de petición de interrupciones sensible al nivel de voltaje, el

dispositivo debe mantener activa la línea de petición de interrupción hasta que el microprocesador reconozca la petición poniendo a una o más líneas de estado en activo. Los microprocesadores que tienen una línea de petición de interrupción sensitiva al flanco de subida o de bajada de voltaje usan flip-flops para capturar las peticiones de interrupción.

El microprocesador debe tener alguna manera de identificar al dispositivo que pidió el servicio de interrupción de modo que la ISR apropiada sea ejecutada. Esta identificación la proporciona el dispositivo durante la secuencia de reconocimiento de interrupción, o bien es definida a partir de la línea de interrupción activada.

4.3.2 Interrupciones de software

Las interrupciones de software son activadas desde el microprocesador, ya sea por solicitud expresa mediante una instrucción en código máquina, o como resultado de una operación con un error, tal como un desborde (es decir, que el número de bytes destinados a contener un resultado sean insuficientes) o una división entre cero. Las interrupciones de software tienen dos usos principales: en la prueba de rutinas de interrupción que han sido escritas para dar servicio a un dispositivo externo y en el establecimiento de un enlace transparente con rutinas de utilerías.

La idea de un enlace transparente consiste en tener rutinas de propósito general cuyas direcciones sean desconocidas para el programador y que pueda tenerse acceso a ellas a través de la tabla de vectores de interrupción cuando se hace una llamada a interrupción. Una implementación de ésta técnica es el ROM BIOS (Basic Input / Output System - Sistema Básico de Entrada / Salida)

en la IBM PC, que contiene este tipo de rutinas. El BIOS implementa un conjunto de funciones para permitir el acceso y control de los diferentes periféricos de la computadora a través de interrupciones de software.

4.4 Interrupciones en el 8088

El microprocesador 8088 puede manejar un total de 256 interrupciones, cada una de ellas especificada por un número de interrupción de 0 a 255. El microprocesador usa una tabla con vectores de interrupción para ubicar la localización de cada ISR específica. Cada vector de interrupción es un apuntador de 4 bytes que contiene el offset y el segmento de la dirección asociada con la ISR. La figura 4.1 ilustra esta tabla. La tabla comienza en la dirección 0000.0000 y tiene 1 KB de longitud (256 x 4 bytes = 1024 bytes).

Cuando ocurre una interrupción, los registros CS, IP y el registro de banderas son copiados en ese orden a la pila (o stack). Los registros IP y CS se cargan entonces con los valores en la tabla de vectores de interrupción que correspondan a la interrupción generada y se limpian las banderas de interrupción (IF) y la de atrapamiento de pasos simples (TF). Al finalizar la ISR la instrucción de código ensamblador IRET restaura el registro de banderas, el IP y el CS desde la pila, de tal manera que el programa continúe desde la instrucción que iba a ser ejecutada antes de que la interrupción ocurriera.

Direcciones	Estructura de la tabla		Interrupción
003FC - 003FF	Offset	Segmento	Interrupción 255
003F8 - 003FB	Offset	Segmento	Interrupción 254
003F4 - 003F7	Offset	Segmento	Interrupción 253
00010 - 00013	Offset	Segmento	Interrupción 4
0000C - 0000F	Offset	Segmento	Interrupción 3
00008 - 0000B	Offset	Segmento	Interrupción 2
00004 - 00007	Offset	Segmento	Interrupción 1
00000 - 00003	Offset	Segmento	Interrupción 0

Figura 4.1 Tabla de vectores de interrupción

Interrupciones externas

El 8088 tiene dos pines que pueden ser ocupados para señalar una interrupción: NMI e INTR. La línea NMI (NonMaskable Interrupt- Interrupción No Mascarable) se dispara por flancos positivos, y el tipo de interrupción que genera no requiere de ciclos de reconocimiento de interrupción. La IBM PC usa esta línea para responder a tres diferentes fuentes de interrupción: un error de paridad en la tarjeta del sistema, la señal I/O CH CK, y una interrupción por el 8087. La línea NMI es mascarada externamente durante la secuencia de inicialización del sistema para evitar una interrupción durante la inicialización de memoria (NMI es habilitada escribiendo el dato 80H en el puerto AOH y deshabilitada limpiando el dato en ese puerto).

El otro pin para solicitar interrupciones es INTR, el cual se dispara por nivel de voltaje. Esta señal no tiene un candado (o latch) conectado en el microprocesador. El tipo de interrupción que

solicita puede ser mascarada limpiando el registro IF por medio de la instrucción de ensamblador CLI y habilitada restaurando el registro IF mediante la instrucción STI. El diseño de la IBM PC conecta INTR al controlador programable de interrupciones (PIC - Programmable Interrupt Controller) 8259A. Comprender la secuencia de interrupción sobre INTR en la IBM PC implica conocer el funcionamiento del PIC, al cual se analiza en la siguiente sección.

Interrupciones internas

Las interrupciones internas en el 8088 no son mascarables. Cuando se genera la instrucción INT, la interrupción ocurre. Las interrupciones internas tienen un nivel de prioridad más alto que las externas.

4.5 El Controlador Programable de Interrupciones (PIC) 8259A

La ventaja de usar el PIC yace en la habilidad de éste de asignar 8 niveles de prioridad a las peticiones de interrupción antes de que sean pasadas al microprocesador, ahorrándole el trabajo de determinar el origen y prioridad de la interrupción solicitada. El PIC puede conectarse al canal de E/S y ahorrar el diseño de hardware y software de interrupción. En la descripción funcional del PIC que sigue, se darán sólo los puntos relevantes en la IBM PC. Para una descripción más detallada, puede consultarse la nota de aplicación de Intel correspondiente (AP-59 *Using the 8259A Programmable Interrupt Controller*).

La programación del 8259A puede dividirse en dos partes: inicialización y operación. La programación de la inicialización

determina el modo básico de operación del 8259A y la programación de la operación permite que características específicas del 8259A sean analizadas y cambiadas durante la operación. Más adelante en este capítulo se detalla el uso de Palabras de Comando de Operación (OCW - Operational Command Words) y Palabras de Comando de Inicialización (ICW - Initialization Command Words).

La figura 4.2 muestra los cuatro bloques requeridos para explicar el procesamiento básico de interrupciones del 8259A. Nótese que uno de los registros por mencionar tiene la misma abreviatura (ISR) que la rutina de servicio de interrupción. Para diferenciarlas en éste documento se hará referencia al registro del 8259A como "registro ISR" y a la rutina de servicio de interrupción simplemente como "ISR".

IRR - Interrupt Request Register (Registro de Petición de Interrupciones). Es un registro de 8 bits donde cada bit corresponde a cada uno de los pines de interrupción IR0-IR7 del 8259A. Un bit en estado activo indica que se ha solicitado una interrupción en el correspondiente pin. Este registro puede leerse mediante una palabra de control OCW3.

IMR - Interrupt Mask Register (Registro de enmascaramiento de interrupciones). Es un registro de 8 bits donde cada bit corresponde a cada uno de los 8 niveles de interrupción. Un bit en estado activo enmascara al correspondiente nivel. Enmascarar un nivel no implica que será removida alguna interrupción pendiente. Desenmascarar una petición de interrupción pendiente permitirá darle servicio. Este registro puede leerse o escribirse mediante una palabra de control OCW1.

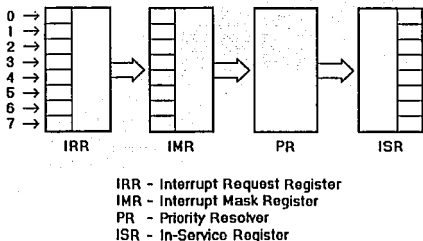


Figura 4.2 Operación del 8259A

PR - Priority Resolver (Determinante de prioridad). El PR determina si una petición de interrupción tiene la suficiente prioridad como para interrumpir a una ISR en ejecución.

ISR - In-Service Register (Registro de Interrupción en Servicio). Es un registro de 8 bits donde cada bit corresponde a un nivel de interrupción. Un bit en estado activo indica que el correspondiente nivel de interrupción está en servicio; es decir, que la correspondiente rutina de servicio de interrupción se está ejecutando. Los bits de este registro no se limpian con la instrucción IRET, sino desde la rutina de servicio de interrupción mediante una palabra de control OCW2. Para leer este registro se usa una palabra de control OCW3.

Supóngase que una petición de interrupción se ha señalado en el pin IR4 del 8259A. La secuencia de operaciones a seguir es la siguiente:

- 1) Si la petición no está enmascarada ($IMR4=0$), es pasada al PR.

- 2) El PR decide si la interrupción tiene la suficiente prioridad para que se le de servicio. Si es así, el 8259A activa la línea INTR del microprocesador 8088.
- 3) Si la bandera de interrupciones del 8088 ha sido habilitada (IF = 1), el 8088 responde con un ciclo de reconocimiento de interrupciones. Durante este ciclo, el estado del IRR es retenido y el PR determina el nivel más alto en el IRR. El número de interrupción correspondiente a IR4 es puesto en el bus de datos y el bit ISR4 del registro ISR es activado.
- 4) El 8088 obtiene del bus de datos el número de interrupción generada (4) y el vector correspondiente a la ISR de la tabla de vectores de interrupción.
- 5) Poco antes de que termine la ejecución de la ISR, el programador debe enviar un comando de fin de interrupción (EOI - End of Interrupt) por medio de una palabra de control OCW2 para limpiar el bit ISR4 del registro ISR. Hasta que esto no sea hecho, posteriores interrupciones en el nivel 4 no serán aceptadas.

4.5.1 Palabras de comando de inicialización (ICW)

El 8259A se inicializa enviando de dos a cuatro palabras de control llamadas Palabras de Comando de Inicialización (ICW - Initialization Command Words). En la IBM PC, las rutinas del Sistema Básico de Entrada y Salida (BIOS) se encargan de inicializar al sistema de interrupciones del 8259A.

La figura 4.3 ilustra la secuencia de inicialización. Una vez que el 8259A es inicializado, subsecuentes cambios a las ICW requieren

repetir la secuencia entera. ICW1 e ICW2 se requieren siempre, mientras que ICW3 e ICW4 únicamente si lo especifica ICW1.

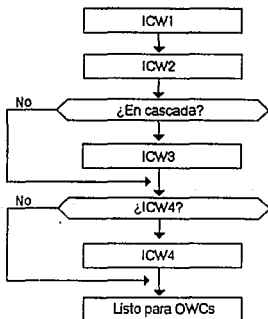


Figura 4.3 Secuencia de inicialización del 8259A

La figura 4.4 ilustra los formatos para ICW1, ICW2 e ICW4. ICW3 no se muestra porque es usado únicamente cuando el 8259A se conecta en cascada para expandirse hasta 64 niveles de interrupción.

ICW1

Las "X" en la figura 4.4 indican bits cuyo estado no importa cuando el 8259A se usa en la IBM PC. Quedan dos decisiones por hacerse en ICW1. El bit B3 determina si el 8259A se dispara por nivel de voltaje (level trigger) o por cambio de nivel de voltaje (edge trigger). en ambos casos es requerido que la fuente de la solicitud de interrupción mantenga el nivel de voltaje en el pin IRn

del 8259A hasta que el 8088 realice el ciclo de reconocimiento de interrupción.

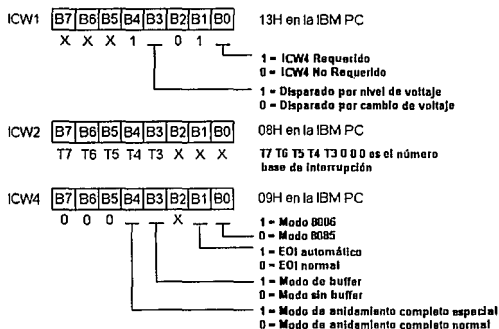


Figura 4.4 Formatos de las palabras de inicialización del 8259A

La otra opción es B4. Este bit determina si IC4 es requerido o no.

En la IBM PC ICW1 es programado con el valor 13H (00010011 binario), implicando el modo de disparo por cambio de voltaje y requiriendo ICW4.

ICW2

Los bits B7-B3 de ICW2 determinan los cinco bits más significativos T7-T3 de los ocho posibles números de interrupción proporcionados al 8088 durante el ciclo de reconocimiento de interrupción. Los bits T2-T0 del número de interrupción son determinados por el nivel de interrupción, como sigue:

IR0	T7	T6	T5	T4	T3	0	0	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR7	T7	T6	T5	T4	T3	1	1	1

En la IBM PC ICW2 esta programado con el valor 8H (00001000 binario), lo cual resulta en números de interrupción del 8 al 15.

ICW4

El bit B0 indica al 8259A si el ciclo de reconocimiento de interrupción que espera del procesador proviene de un 8085 o un 8088. El bit B1 permite elegir la característica de fin de interrupción automático (AEOL - Automatic End Of Interrupt), ahorrándole a la ISR que envíe un EOI.

En el modo de buffer (bit B3) el pin SP/EN del 8259A se irá a nivel bajo si la salida del bus de datos de éste es habilitada. Así, SP/EN puede ser usado para selectivamente habilitar y deshabilitar buffers para darle al 8259A acceso exclusivo al bus de datos durante el ciclo de reconocimiento de interrupción.

El bit B4 permite activar el modo especial de anidamiento completo o el modo normal de anidamiento completo. El uso principal del modo especial de anidamiento completo es cuando se tiene un sistema en cascada de chips 8259A, aunque puede ser útil al conectar un 8259A al canal de E/S. En el modo normal una petición entrante de interrupción puede causar una interrupción si está a un nivel de prioridad más alto que peticiones en servicio. En el modo especial, una petición entrante puede causar una

interrupción si está a un nivel igual o mayor que peticiones en servicio.

En la IBM PC, ICW4 esta programada con el valor 9H (00001001 binario), resultando en un modo 8088, un EOI normal, un modo de buffer, y un modo normal de anidamiento completo.

El 8259A del sistema se encuentra en los puertos 20H y 21H. El código para su inicialización es el siguiente:

```
MOV AL,13H
OUT 20H,AL      ;ICW1
MOV AL,08H
OUT 21H,AL      ;ICW2
MOV AL,09H
OUT 21H,AL      ;ICW4
```

4.5.2 Palabras de comando de operación (OCW)

Una vez que el 8259A ha sido inicializado, las palabras de comando de operación permiten personalizar las características prioritarias, así como preguntar por el estado de varios registros.

OCW1

OCW1 permite leer y escribir al registro de máscara. Poner un bit en el registro de máscara permite enmascarar un interrupción específica,

OCW2

Proporciona varias maneras de enviar comandos EOI (fin de interrupción) y cambiar el esquema de prioridades. En la inicialización, la estructura de prioridades es puesta de tal forma que IR0 tiene la prioridad más alta mientras que IR7 tiene la más baja.

OCW3

Permite leer a los registros IRR e ISR. OCW3 también contiene al comando "poll" (exploración). Este comando se proporciona para sistemas que no usan la secuencia de interrupción y reconocimiento de interrupciones mediante INT e INTA, como sucede si se desea agregar un 8259A en el canal de E/S.

4.6 Interrupciones en el canal de E/S

Los pines IR2-IR7 del 8259A del sistema se conectan directamente al canal de E/S, donde aparecen como IRQ2-IRQ7. IBM especifica el siguiente uso para estas líneas:

IRQ2	Reservado.
IRQ3	Comunicaciones asíncronas (secundario).
IRQ4	Comunicaciones asíncronas (primario).
IRQ5	Disco duro.
IRQ6	Disco flexible.
IRQ7	Impresora.

Aunque a primera vista parece que todos los IRQ's han sido ocupados, la mayoría de los sistemas tienen algunos libres. IRQ2

no está actualmente en uso, pero está reservado para un posible producto futuro de IBM.

Debe mencionarse que es posible conectar varias fuentes de interrupción en una línea de IRQ. El hardware que implemente esto debe proporcionar alguna manera en que la rutina de servicio de interrupción determine la fuente de la interrupción. Si no se desea un hardware muy sofisticado, se debe implementar por software la resolución de prioridades. Una solución más simple es añadir otro 8259A en la línea de IRQ.

4.7 Acceso Directo a Memoria (DMA)

El acceso directo a memoria (DMA- Direct Memory Access) es una técnica de transferencia de información entre memoria y puertos que no requiere intervención del microprocesador. En el capítulo referente a hardware de la IBM PC se consideraron los aspectos generales referentes a DMA y el 8237A-5, el controlador de DMA incluido en la tarjeta del sistema de la PC y en el capítulo dedicado a arquitecturas de bus en sistemas XT y AT se habló de los ciclos de bus de DMA. En las siguientes secciones se retomará y extenderá esta información para describir la operación del controlador de DMA.

4.8 Secuencia de la transferencia de datos por DMA

Cada canal de DMA tiene asignado un pin en el chip 8237A-5 (DREQ0-DREQ3) y un pin de reconocimiento de DMA (DACK0-DACK3). Una secuencia típica de DMA toma lugar de la siguiente manera:

- 1) Un dispositivo de E/S señala una petición de DMA en uno de los pines de petición de DMA.
- 2) El 8237a-5 asigna la prioridad correspondiente a la petición respecto a otras peticiones pendientes.
- 3) El 8237A-5 pide el bus al 8088 activando el pin Hold Request (HRQ).
- 4) La lógica de la tarjeta del sistema espera al 8088 para que entre a un período de reloj inactivo, desconecta al microprocesador de los buses, y activa el pin Hold Acknowledge (HOLDA).
- 5) El 8237A-5, una vez que tiene el control del bus, activa la línea de reconocimiento de DMA al dispositivo que requiera el servicio.
- 6) El dispositivo responde al reconocimiento de DMA desactivando la línea de petición de DMA y preparándose para la transferencia de datos.
- 7) El 8237A-5 activa las señales apropiadas de control para completar la transferencia (MEMR e IOW para una lectura y MEMW e IOR para una escritura).
- 8) Finalmente, el pin Hold Request es desactivado, permitiendo al procesador continuar.

4.9 Razón de transferencia de DMA

Si la línea de petición de DMA (DREQx) se mantiene activa o reactivada, el proceso se repetirá al final del ciclo actual del procesador. Este ciclo tiene un largo de 10 ciclos de reloj (cuatro estados normales y seis de espera). Puesto que un ciclo de DMA toma lugar concurrentemente con cada diez estados de ciclo de procesador, la máxima razón de transferencia de DMA es de 479 Kbytes/seg. La razón de transferencia real será menor debido a que el canal 0, con la más alta prioridad, realizará un ciclo de refresco de memoria cada 15.2 microsegundos. de este modo, cada siete ciclos de DMA serán seguidos de un ciclo de refresco. El número total de ciclos de reloj para completar siete ciclos de DMA son $10 \times 7 + 9 = 79$, resultando en una razón de transferencia de 422 Kbytes/seg.

4.10 Programación del 8237A-5

El controlador de DMA se programa a través de 16 puertos. Ocho de los puertos dan acceso a los registros de dirección y conteo para cada canal, y los otros ocho permiten comandos más generales.

Registros de Conteo y dirección

Hay un puerto de conteo y uno de dirección para cada canal. Cada canal cuenta con dos registros de conteo y de dirección. Un registro de conteo, CURRENT, contiene el conteo actual de la transferencia que esta teniendo lugar, mientras que el otro, denominado BASE, retiene la cuenta inicial para autoinicialización de un nuevo ciclo de transferencia. Los registros de dirección BASE y CURRENT se definen de manera similar. El número de

bytes transferidos durante una operación de DMA es siempre uno más que el valor programado en los registros de conteo.

Comandos de software

Los comandos de software permiten la inicialización y el control mediante el acceso a varios registros internos adicionales. Varios de estos comandos no son aplicables en la IBM PC, ya que el controlador de DMA ha sido limitado a transferencias entre puerto y memoria.

Entre los registros que existen están:

Registro de comando. Determina varias características que no deben ser cambiadas para el correcto funcionamiento en una IBM PC. Se programa con un byte 0.

Registro de modo. Permite especificar detalles en la transferencia por DMA, tal como el modo de transferencia por bytes simples o por bloques de bytes, el incremento o decremento de direcciones y especificar si es una operación de lectura o escritura.

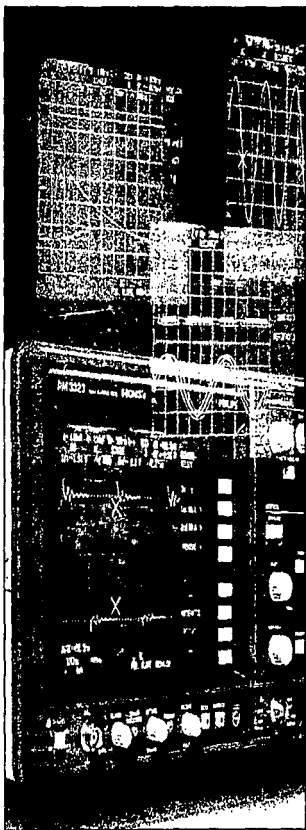
Registro de máscara. Contiene un bit de máscara para cada uno de los cuatro canales.

Registro de estado. Indica cuales canales tienen una petición de DMA pendiente y cuales han alcanzado el conteo terminal (TC).

4.11 Sumario

En éste capítulo se analizó la estructura de interrupciones y de transferencia de información por acceso directo a memoria. Ambos procesos son controlados por chips dedicados exclusivamente a tales tareas.

Con éste capítulo se da fin al análisis del hardware y operación de la IBM PC. Una vez que se conocen las capacidades y limitaciones del sistema, los siguientes capítulos se ocuparán de analizar a los osciloscopios, con el fin de determinar las características que se deben incluir en el módulo de osciloscopio digital. Una vez que se hayan sentado las bases del hardware disponible y análisis de características del proyecto a realizar, se tendrán los elementos requeridos para empezar el diseño, tanto en el hardware como en el software.



Parte 2

Fundamentos del osciloscopio

Objetivo: *Analizar los componentes de un osciloscopio y establecer los elementos constitutivos y funcionales del sistema diseñado.*

Capítulos

- Osciloscopios analógicos
- Osciloscopios digitales

Capítulo 5

Osciloscopios analógicos

5.1 Introducción

En este capítulo y en el siguiente se dará una breve exposición sobre los osciloscopios analógicos y los digitales. Se analizarán primero los osciloscopios analógicos, describiendo primero lo que es un osciloscopio, los tipos de osciloscopios que hay y el funcionamiento mediante diagramas de bloques del osciloscopio analógico. Puesto que el objetivo de analizar al osciloscopio en estos capítulos es la búsqueda de las características a incluir en el diseño de éste proyecto, no se profundizará demasiado en la operación del osciloscopio.

5.2 Fundamentos del osciloscopio

El osciloscopio es un instrumento diseñado para reproducir en forma gráfica por medio de una pantalla la amplitud de las señales eléctricas, en función de una base de tiempos usada como barrido de referencia. Existen en el mercado dos tipos de osciloscopio: los analógicos y los digitales. Estos últimos son de aparición más reciente. Ambos tipos de osciloscopio tienen algunos elementos similares y otros muy diferentes. La figura 5.1 muestra el diagrama a bloques de un osciloscopio analógico. La etapa de entrada de la señal consta de un atenuador y un preamplificador vertical en cada canal, los cuales son multiplexados a una línea de retardo y a un amplificador de salida vertical. La señal de salida es enviada al

tubo de rayos catódicos, en donde deflexiona verticalmente a un haz de electrones en una magnitud proporcional a su valor. Al mismo tiempo, una señal de disparo generada interna o externamente llega a un generador de barrido horizontal y a un amplificador de salida horizontal. La señal de salida es enviada también al tubo de rayos catódicos, con la diferencia de que en este caso se provoca una deflexión horizontal del haz de electrones. Cuando el haz de electrones choca con la superficie del tubo de rayos catódicos, se visualiza un punto en la pantalla. La combinación de las deflexiones horizontal y vertical del haz de electrones provocan que cuando éste choque con la superficie del CRT se pueda visualizar la forma de onda de la señal de entrada. Las siguientes secciones detallan más la operación completa del osciloscopio analógico.

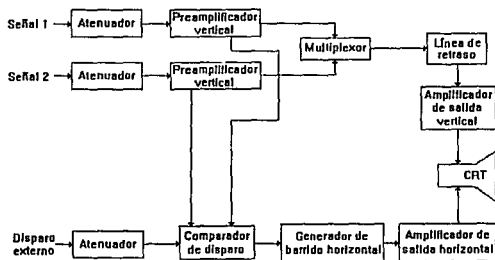


Figura 5.1 Diagrama a bloques de un osciloscopio analógico

5.3 El tubo de rayos catódicos

El principal elemento del osciloscopio es el tubo de rayos catódicos (CRT - Cathode Ray Tube), el cual consta de cuatro elementos principales:

- 1.- Un cañón electrónico para producir la corriente de electrones.
- 2.- Elementos de foco y aceleración para producir un haz de electrones bien definido.
- 3.- Placas de deflexión horizontal y vertical para controlar la ruta del haz.
- 4.- Una cubierta de vidrio al vacío con una pantalla fosforescente que se ilumina cuando es golpeada por el haz de electrones.

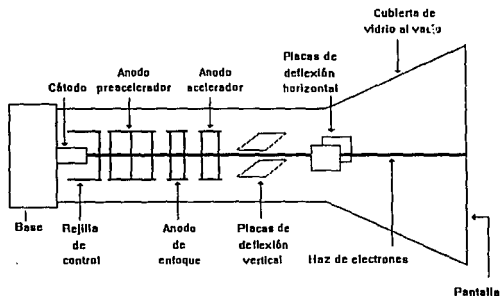


Figura 5.2 El tubo de rayos catódicos

La figura 5.2 muestra un diagrama del tubo de rayos catódicos. El cañón electrónico tiene la función de producir una corriente de electrones que sigue una trayectoria hasta la pantalla fosforescente, situada en la cara delantera del tubo. Los electrones al ser sometidos a fuerzas electrostáticas serán controlados en número, posición y velocidad. Si el haz electrónico no está sometido a otras fuerzas, formará un punto en el centro de la pantalla. Esto se logra revistiendo la superficie del tubo con un material fosforescente, que tiene la propiedad de emitir luz al recibir el impacto de los electrones.

5.3.1 El cañón electrónico

El cañón electrónico se compone de un cátodo, un filamento, una rejilla de control y un ánodo de enfoque y aceleración, los cuales se muestran en la figura 5.3.

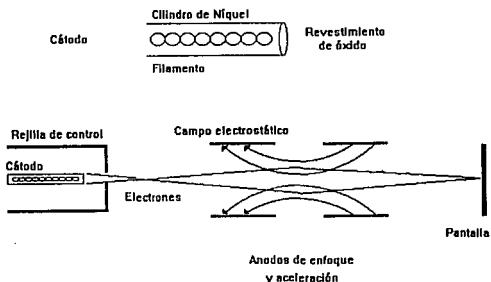


Fig 5.3 El cañón electrónico

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

El cátodo es cilíndrico, hecho de níquel y se encuentra revestido en un terminal con una película de óxidos de bario y estroncio. Este revestimiento se aplica al forro del tubo apuntando en la dirección de la pantalla. Mediante una corriente que se establece en el filamento, el cual está hecho de tungsteno o de una aleación del mismo, éste calienta indirectamente al cátodo. Los electrones serán liberados de la capa de óxido en la superficie del cátodo como efecto del calentamiento provocado por el filamento.

5.3.2 Elementos de foco y aceleración

Para proporcionar algún enfoque de los electrones se coloca una rejilla de control con una pequeña apertura en la dirección de la pantalla después del cátodo (figura 5.3). Además, se aplica un voltaje de polarización en la rejilla para controlar al flujo de electrones que pasan a través de ella. Si el voltaje aplicado es negativo con respecto al cátodo, habrá una reducción en el número de electrones que pasan por la apertura de la rejilla. De ésta manera la rejilla permite el ajuste del número de electrones que llegan finalmente a la pantalla del CRT.

Una vez que los electrones han dejado la rejilla de control se les dará un enfoque y aceleración hacia la pantalla del tubo por dos ánodos que tienen un voltaje positivo con respecto al cátodo. El segundo ánodo, además, tiene un potencial mayor que el primero. De hecho existe una diferencia de potencial entre ambos de varios miles a volts. Cuando los electrones han pasado por éstos ánodos, lo hacen en forma de un haz muy estrecho debido al campo eléctrico producido por el voltaje en los ánodos.

5.3.3 Placas de deflexión horizontal y vertical

En su trayectoria hacia la pantalla del CRT, el haz de electrones pasa por un par de placas de deflexión horizontal y un par de placas de deflexión vertical. Cada par de placas tiene una diferencia de potencial que provoca que el haz de electrones sea desviado en una dirección u otra. El movimiento producido por las placas de deflexión vertical darán una visualización en la pantalla de la magnitud de la señal medida por el osciloscopio, mientras que el movimiento producido por las placas de deflexión horizontal permitirán visualizar adecuadamente la frecuencia de la señal. Si no se aplicara señal en la placas de deflexión vertical, se visualizaría en el osciloscopio una línea horizontal en el centro de la pantalla. Si por otro lado, no se proporcionara señal a las placas de deflexión horizontal, se visualizaría únicamente una línea vertical en el centro de la pantalla. Una ausencia total de señal en las placas de deflexión provocaría que se muestre sólo un punto en el centro de la pantalla.

5.3.4 Cubierta de vidrio al vacío

La cubierta de vidrio cuenta con una pantalla al frente que se ilumina cuando recibe al haz de electrones como efecto del revestimiento que tiene. Esta cubierta se encuentra al vacío para prevenir la presencia de gases en su interior que pudieran interferir en el movimiento del haz electrónico y para proteger el filamento del tubo de la oxidación y otras reacciones químicas. El material más usado como revestimiento es la Willemita (ortosilicato de zinc), que emite una luz verde. Otros materiales son el óxido de zinc, el cual emite una luz azul y el silicato de zinc y berilio, que emite una luz amarilla. La duración de la luminosidad (llamada también *persistencia*) de la imagen varía con el material

utilizado como revestimiento y con la cantidad de energía del haz de electrones.

5.4 La base de tiempos y el disparo

El propósito del generador de la base de tiempos (o generador del barrido) es el de producir una señal diente de sierra necesaria para deflexionar al haz de electrones linealmente a lo largo del eje horizontal. El tiempo de barrido corresponde al tiempo de elevación de la rampa en el diente de sierra, determinando que tan rápido se deflexiona el haz a través del CRT. Después de que la señal diente de sierra es generada, se envía al amplificador de la base de tiempos, el cual consta de un amplificador y de un inversor de fase. A la salida de este amplificador se entregan dos dientes de sierra amplificados, simétricos con respecto al eje horizontal. La señal diente de sierra con elevación positiva es aplicada a la placa de deflexión horizontal derecha y la de elevación negativa a la placa de la izquierda. El resultado es una deflexión lineal del haz de electrones de izquierda a derecha a través de la pantalla.

Si la forma de onda observada en el osciloscopio quiere representarse como un desplegado estable en la pantalla, cada diente de sierra del barrido horizontal deberá iniciarse en el mismo punto de cada forma de onda sucesiva. Esto se logra disparando el barrido en un punto deseado usando una muestra de la señal de entrada tomada del amplificador vertical. En ese punto de la forma de onda seleccionada por los controles del circuito de disparo se genera un pulso. Este pulso es aplicado al generador de la base de tiempos, permitiendo definir la frecuencia con la que se genera la señal diente de sierra.

Una vez que la señal diente de sierra alcanza su punto más alto, el haz de electrones ha sido desplazado de izquierda a derecha. Cuando la señal diente de sierra desciende, el haz se desplazará de derecha a izquierda, a lo cual se le llama el "retrazo" (sic) de la señal. El retrazo simplemente reposiciona el haz en el extremo izquierdo para el siguiente barrido. Puesto que no es deseable que esto sea visible en la pantalla, el generador de la base de tiempos produce otra onda de nominada señal de "no borrado", la cual coincide con el instante de tiempo en que el diente de sierra se eleva. La función de esta señal es la de permitir que el haz sea visible en la pantalla. Una vez que esta señal esta ausente, el haz es cortado, impidiendo que el retrazo sea mostrado en la pantalla.

El disparo, la generación del diente de sierra y las operaciones de "no borrado" requieren alrededor de $0.15 \mu\text{s}$ antes de que el barrido sea iniciado. Este retardo entre el inicio de la forma de onda y el punto donde el disparo es efectuado, implica que cierta información de la señal puede perderse, de modo que es requerido retardar también la señal del amplificador vertical. Esto se logra introduciendo la señal de entrada del osciloscopio a una línea de retardo (ver fig. 5.1), la cual impone un retardo total de $0.25 \mu\text{s}$ a la forma de onda que se observa en el osciloscopio.

5.5 La puntas del osciloscopio

Las puntas del osciloscopio son tan importantes para la visualización de la señal como cualquiera de los componentes internos del osciloscopio. Unas buenas puntas deben entregar la mejor señal posible al osciloscopio y aplicar tan poca carga como sea posible en el circuito medido. Para lograr esto, muchas puntas pasivas atenúan la señal de entrada antes de que llegue al osciloscopio. La mayoría de ellas tienen factores de atenuación de

x1, x10 y x100. Una punta x1 no tiene atenuación. Para prevenir los efectos de carga capacitiva comúnmente encontrados en una punta x1, se usa con frecuencia una punta de atenuación x10.

5.6 Sumario

En este capítulo se analizó al osciloscopio analógico. Sus partes principales son el circuito de entrada, el disparador, el generador de la base de tiempos o barrido, el tubo de rayos catódicos en donde se observará la señal, el sistema de deflexión horizontal para amplificar el componente de la base de tiempos y el sistema de deflexión vertical, el cual amplifica la componente vertical de la señal (o amplitud).

En el siguiente capítulo se continuará el análisis del osciloscopio describiendo el funcionamiento y las principales características del osciloscopio digital.

Capítulo

6

Osciloscopios digitales

6.1 Introducción

En este capítulo se dará una descripción básica del funcionamiento y características de los osciloscopios digitales. Uno de los procesos más importantes efectuados por estos es la conversión analógico a digital de la señal de entrada. Debido a la importancia de éste tema, su análisis se reserva para un capítulo posterior. Cuando se haya finalizado el presente capítulo se tendrán los conceptos requeridos para incluir en el diseño del módulo de osciloscopio digital para la PC la funcionalidad y versatilidad de estos instrumentos.

Los osciloscopios de almacenamiento digital (DSO -Digital Storage Oscilloscopes) no son ideales para todas las aplicaciones, de modo que es importante conocer sus características para saber si deben ser usados en lugar de los analógicos o no. Por lo general, sin embargo, los DSO ofrecen características que las versiones analógicas no pueden igualar:

- ❑ Temporización digital, más exacta que la de versiones analógicas.
- ❑ Las formas de onda pueden ser manipuladas para resaltar ciertas características para un análisis especial, permitiendo al usuario hacer un acercamiento (zoom) en las áreas de interés.

- Las formas de onda pueden ser almacenadas indefinidamente y rellamadas de la memoria para ser comparadas, una característica no compartida por los osciloscopios de almacenamiento analógico, los cuales almacenan una forma de onda durante un tiempo limitado en la pantalla, pero no en memoria.

- Un DSO es esencialmente un equipo de prueba automático donde un microprocesador actúa como controlador, recibiendo los valores medidos y cambiando parámetros de medición como la sensibilidad vertical o la razón de muestreo.

6.2 Descripción funcional de un DSO

Un DSO es similar a un osciloscopio analógico en algunos aspectos, pero muy diferente en otros. La figura 6.1 muestra un diagrama a bloques de un osciloscopio digital de dos canales. La etapa de entrada consta de un atenuador, un preamplificador vertical y un disparador (trigger), cada uno de ellos con la misma función que en un osciloscopio analógico. Sin embargo, los dos canales no están multiplexados como en el caso de la versión analógica. Cada uno de los canales tiene una trayectoria separada hacia el microprocesador, permitiendo procesar los datos simultáneamente.

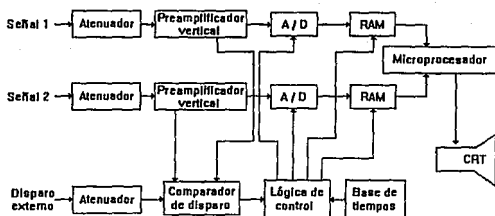


Figura 6.1 Diagrama a bloques de un osciloscopio digital

La señal de entrada pasa a través de un convertidor A/D, el cual envía la señal digitalizada a memoria de semiconductor hasta que el microprocesador tome esa información y la presente en pantalla. El hecho de que la información sea tomada de la memoria hace posible que la señal sea mostrada en el monitor con la misma nitidez e intensidad, a diferencia de los osciloscopios analógicos, donde se requieren frecuentes ajustes de éstos parámetros debido a la dependencia que tiene la visualización de la señal con la frecuencia de la señal de barrido requerida.

6.3 Características generales de un DSO

Al principio del capítulo se mencionaron algunas de las características de un DSO. Podemos mencionar también las siguientes:

- **Mediciones automáticas.** Un DSO puede medir un número de características de las señales de entrada y mostrarlas en la pantalla, incluyendo el tiempo de elevación, el tiempo de caída, retraso después del disparo, voltaje pico a pico,

offset, frecuencia, ancho del pulso y ciclo de trabajo de la señal, entre otros. También es posible realizar operaciones matemáticas como suma, multiplicación integración o incluso FFT (Fast Fourier Transform - Transformada Rápida de Fourier) de las señales de entrada. Un usuario puede también controlar la posición de los cursores en el monitor para mostrar la diferencia en tiempo y amplitud entre dos puntos en una forma de onda. Estas características aceleran el proceso de medir una señal, eliminando el error humano y proporcionando más exactitud.

- ❑ **Persistencia infinita.** Esta característica, llamada también modo de acumulación de puntos, permite observar el "jitter" total en una señal. En este modo, la señal en el monitor nunca es borrada, simplemente se acumula en la pantalla. Las partes estables de la señal se muestran como líneas sólidas, pero cualquier "jitter" se muestra difuso.
- ❑ **Modo de envolvente.** Similar al modo de persistencia infinita, en el que muestra excursiones mínimas y máximas de una forma de onda en el tiempo y por lo tanto mostrando los puntos mínimos y máximos de "jitter".
- ❑ **Impresión de la forma de onda.** Algunos DSO pueden ser conectados a una impresora gráfica o a un plotter para obtener una copia en papel de la imagen en el monitor.
- ❑ **Modo de Promedio (averaging).** Permite mostrar el promedio de una señal repetitiva, uniformizándola para una visualización más clara. Este modo permite minimizar el efecto de ruido en la señal.
- ❑ **Modo de enrollamiento (Roll mode).** Un DSO puede almacenar señales lentas usando velocidades de hasta 1

hora/div, actuando de manera análoga a un sismógrafo. A medida que la información es mostrada en la pantalla, la señal se desplaza de derecha a izquierda. Esta capacidad es llamada "roll mode".

- **Disparo previo y posterior (pre-triggering y post-triggering).** Un DSO muestrea y almacena datos en numerosas formas. Puede mostrar y almacenar datos en un flujo constante, y entonces mostrar lo que tenga en memoria cuando el disparo ocurre. Igualmente, puede muestrear señales antes o después de que el disparo ocurra.

6.4 Ancho de banda y razón de muestreo

Hasta el advenimiento de los osciloscopios digitales, la especificación más importante era la del ancho de banda. Esta se refiere a la máxima frecuencia de una señal que el osciloscopio puede mostrar con una atenuación no mayor a 3 dB. Para un osciloscopio analógico, además, esta señal tiene que ser repetitiva. Con ésta definición, los usuarios fácilmente comprendían la respuesta que un equipo podía proporcionar en la captura de una señal.

Sin embargo, el ancho de banda por sí sólo no proporciona una medición completa o adecuada de la capacidad de captura de un DSO. Para comprender esto es necesario considerar también el tipo de técnica de muestreo que se está ocupando.

Un DSO adquiere señales tomando valores (o muestras) momentáneamente de la señal analógica de entrada a intervalos de tiempo regulares. Los valores muestreados son almacenados en memoria y la forma de onda original es entonces reconstruida

mostrando los contenidos de la memoria. Generalmente, entre más alta sea la razón de muestreo (especificada en muestras/segundo), será más alta la frecuencia y mayor la exactitud de la señal a ser capturada, mostrada y analizada.

Los DSO actuales son capaces de tener razones de muestreo del orden de cientos de millones de muestras por segundo. Los fabricantes consiguen esto mediante técnicas muy sofisticadas. Una de ellas consiste en almacenar temporalmente la señal en un dispositivo de acoplo de carga (CCD - Charge Coupled Device). Este consiste en microcapacitores de semiconductor dispuestos como celdas, cada una de las cuales almacena una carga y la transfiere a la siguiente celda. El dispositivo actúa como un registro de corrimiento analógico, en bases de "primera entrada - primera salida". Cuando todas las celdas están llenas, la carga es una réplica de la señal original que puede ser leída más lentamente con el fin de convertirla a forma digital. Es posible, por ejemplo, almacenar una señal a una razón de 100 millones de muestras por segundo y después usar un convertidor A/D de precisión que tome muestras a una razón de 100 mil muestras por segundo. Para acelerar aún más este proceso, es posible también tener dos sistemas con CCD y multiplexarlos, con el fin de duplicar la razón de muestreo.

6.5 Muestreo en tiempo real

Para señales no repetitivas en el tiempo ("single shot"), o para eventos repetitivos de baja frecuencia, toda la información debe ser rápidamente obtenida en un paso, mientras que la señal o el evento esta sucediendo. El muestreo rápido en tiempo real (Fig. 6.2) logra esto tomando una serie de muestras rápidas de un solo

período de la señal medida, y usándolas para reconstruir la forma de onda original.

La razón a la cual estas muestras en tiempo real son tomadas esta directamente relacionada con la máxima velocidad de las señales no repetitivas en el tiempo que pueden ser capturadas y mostradas con exactitud. Por ejemplo, la regla aceptada a nivel industrial es que se requieren 10 muestras por período de la señal de entrada de frecuencia más alta. Así, para un DSO con una razón de muestreo en tiempo real de 250 MS/s (millones de muestras por segundo), la frecuencia más alta a medir será de 25 MHz. Para osciloscopios que solamente tienen muestreo en tiempo real, esta frecuencia es llamada el ancho de banda digital, aunque éste término tiene otras bases para osciloscopios con técnicas de muestreo adicionales.

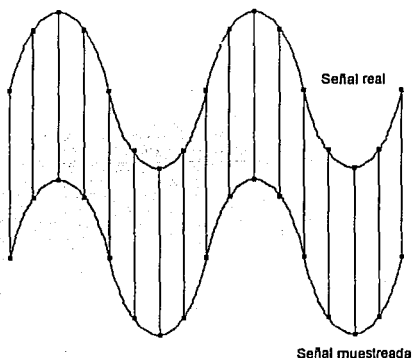


Figura 6.2 Muestreo en tiempo real

En adición a considerar esta respuesta en la captura de señales no repetitivas en términos de la frecuencia, también se le puede conceptualizar en términos del tiempo, proporcionando la "resolución horizontal" del osciloscopio. El ejemplo anterior de 250 MS/s tendría así una equivalencia de 4 ns como resolución horizontal.

Cuando se comparan razones de muestreo de diferentes DSO se tiene que considerar el número de convertidores A/D que se usan en el proceso de muestreo. Algunos DSO tienen únicamente 1 convertidor A/D compartido entre 2 o más canales de entrada. Cuando se dedica un convertidor a un sólo canal, puede muestrear la señal de entrada a velocidad completa. Cuando se usa un convertidor para dos canales, la respuesta se reduce a la mitad.

6.6 Muestreo en tiempo equivalente

Para señales repetitivas en el tiempo es posible usar muestreo en tiempo equivalente (llamado también muestreo repetitivo) para capturar señales de frecuencias más altas que en el caso de las señales muestreadas en tiempo real. El muestreo en tiempo equivalente toma una o varias muestras de cada ciclo de la señal, almacenando información del muestreo realizado con el fin de poder reconstruir posteriormente la señal original a partir de las muestras tomadas. Esto permite alcanzar resoluciones del orden de fracciones de nanosegundos sin tener que recurrir a los caros convertidores A/D de alta velocidad. Es claro que este método se fundamenta en el hecho de que la señal es repetitiva en el tiempo y no se puede aplicar a aquellas señales que no lo son. Igualmente, tampoco se aplica a señales que tienen poca repetibilidad, puesto que puede darse el caso de que la señal haya

terminado antes de que todas las muestras requeridas hayan sido tomadas, impidiendo reconstruir fielmente la señal original.

Existen dos tipos de muestreo en tiempo equivalente:

Secuencial. Como se observa en la figura 6.3, el muestreo consiste en tomar una muestra en cada ciclo de la señal, con cada muestra ocurriendo a un intervalo de tiempo que se incrementa progresivamente con respecto al punto de disparo. Puesto que el muestreo es disparado por la misma señal, no puede especificarse una razón de muestreo.

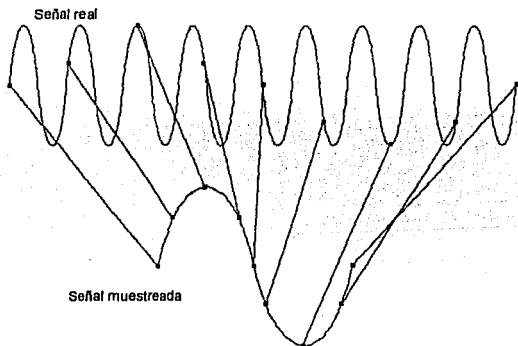


Figura 6.3 Muestreo en tiempo equivalente secuencial

Aleatorio. En este caso se usa el reloj interno del DSO para hacer un número de muestras durante cada ciclo de la señal repetitiva, tal como se muestra en la figura 6.4. Un dispositivo de

temporización mantiene un rastro de la distancia entre el punto de disparo original de la señal y el momento en que cada muestra es tomada, de tal manera que el osciloscopio pueda entonces reconstruir la señal con las muestras en la secuencia correcta.

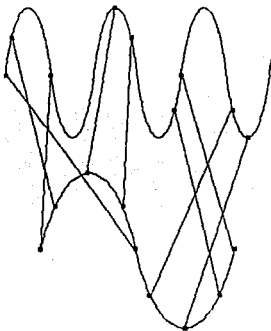


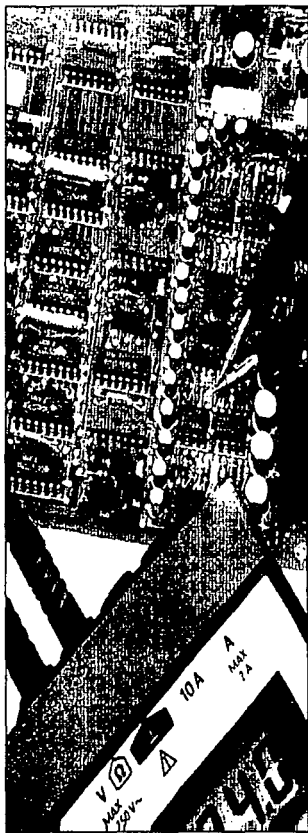
Figura 6.4 Muestreo en tiempo equivalente aleatorio

Estos dos métodos hacen posible digitalizar señales con un ancho de banda igual al de los osciloscopios analógicos. Cada una de éstas técnicas tiene sus ventajas. El método de muestreo en tiempo equivalente secuencial da una más rápida reconstrucción de señales de alta frecuencia, mientras que el método de muestreo en tiempo equivalente aleatorio ofrece información de "pretriggering" y da una visualización más rápida de señales de baja frecuencia.

6.7 Sumario

En éste capítulo se analizó brevemente el funcionamiento y descripción de las capacidades de los osciloscopios de almacenamiento digital. Se mencionaron las características por las cuales se han destacado por encima de sus contrapartes analógicas: capacidad de mediciones automáticas, impresión y almacenamiento de señales, diversos modos de operación, capacidad de ser conectados con computadoras personales, impresoras o plotters, etc. Igualmente, se analizaron los modos de muestreo en tiempo equivalente y tiempo real, los cuales determinan en buena parte la capacidad de medición de éstos instrumentos.

Con la finalización de éste capítulo termina la revisión hecha tanto a las computadoras personales como a los osciloscopios. Con la información recopilada de estos dos temas ahora es posible establecer las capacidades que pueden esperarse de un sistema como el que se pretende diseñar. En el siguiente capítulo se detallan las especificaciones de hardware y en capítulos posteriores se procede con el diseño del módulo de osciloscopio digital para la PC.



Parte 3

Diseño del hardware

Objetivo: *Diseñar el hardware de comunicación de datos para la PC por medio de un puerto paralelo, así como el sistema de adquisición de datos.*

Capítulos

- Especificaciones de diseño del hardware
- Interfase con la IBM PC
- Conversión A/D
- Módulo de osciloscopio digital

Capítulo

7

Especificaciones de diseño del hardware

7.1 Introducción

Desde este capítulo hasta los restantes se tomará como referencia el marco teórico descrito en capítulos anteriores para realizar un diseño que satisfaga los requerimientos de éste proyecto. Una vez que se ha estudiado lo que es posible obtener de una PC y lo que un osciloscopio puede hacer, este capítulo concreta dichos conceptos en especificaciones del producto a diseñar. Igualmente, en este capítulo se seleccionan los elementos de diseño y se establecen los procedimientos a seguir para llevarlo a cabo.

7.2 Descripción global del proyecto

Todo diseño en ingeniería requiere que primero se haga un análisis del problema para poder dividirlo en sus partes fundamentales, de tal forma que la solución completa conste de la suma de las soluciones dadas para cada parte. Esto representa como principal ventaja que las pruebas durante el desarrollo del proyecto serán más fáciles que si se hicieran sobre el conjunto completo. Igualmente, al darle mantenimiento al producto terminado o hacerle una mejora, será más fácil si se tiene que operar solamente sobre alguna parte en específico. Para éste proyecto en particular, la solución más general consiste en el diseño de tres bloques:

- Hardware de adquisición de datos
- Hardware de interfase de datos entre el bloque de adquisición de datos y la computadora personal.
- Software de control y manipulación de la información recibida.

Esta división por bloques tiene como principal ventaja que el primer bloque mencionado, el de adquisición de datos, puede ser sustituido por un bloque de instrumentación diferente. Más ventajas serán descritas posteriormente. Analicemos estos tres bloques como cajas negras:

Bloque de adquisición de datos. En este bloque se recibe como entrada una señal analógica, la cual es transformada y convertida a formato digital comprensible para la computadora. Esta información digital es la que se entregará a la salida de este bloque.

Bloque de interfase de datos. Este bloque toma como entrada los datos digitales que el bloque de adquisición de datos envía. Su función será hacer que estos datos sean reconocidos por la computadora personal, de modo que su salida son los mismos datos digitales, dirigidos al microprocesador a través del bus de expansión de la PC.

Software de control y manipulación. La entrada de datos proviene de dos fuentes: el usuario (comandos de operación) y el bloque de transferencia de datos (la información digital). La salida será a través de la pantalla o la impresora.

Este planteamiento del diseño puede agruparse como diseño de hardware y diseño de software. En la parte cuatro de este

documento se detalla la etapa de software, por lo que en el presente capítulo se continuará únicamente con la descripción del hardware.

7.3 Diseño del hardware

Una vez que se han definido globalmente los componentes del diseño del hardware, puede hacerse una descripción más elaborada del mismo. Las dos siguientes secciones analizan las dos etapas de hardware.

7.3.1 Adquisición de datos

En el capítulo 1 se mencionó que uno de los objetivos por alcanzar es el de elaborar un diseño con capacidad de crecimiento a más módulos de instrumentación. Para esto, es necesario que el módulo de adquisición de datos o módulo de osciloscopio digital cuente con una manera de ser habilitado mediante un comando emitido por software desde la computadora. Usando este habilitador se permitirá activar a uno sólo de los diferentes módulos de instrumentación conectados al sistema. Por otro lado, para evitar problemas de espacio y de sobrecarga en el bus de expansión de la computadora personal, es necesario que estos módulos de instrumentación se encuentren en el exterior de la computadora, con su propia fuente de alimentación.

Regresando al módulo de adquisición de datos, éste deberá contar con las siguientes características:

- Diseño externo a la computadora personal.

- Fuente de alimentación propia. Los voltajes que entregará esta fuente son de ± 5 V regulados.
- Dos canales de entrada analógica. Cada canal aceptará una entrada analógica máxima de ± 5 V.
- Ancho de banda de 1 millón de muestras por segundo.
- 4 Kb de memoria RAM por canal.
- Inicialización remota desde la PC.
- Indicadores luminosos que muestren que el módulo ha sido encendido, que ha sido seleccionado y que esta recibiendo datos por los canales de entrada analógica.
- Controles para calibrar la amplitud.
- Conversión A/D de 8 bits, no multiplexada.
- Conversión A/D usando muestreo en tiempo real.
- Operación controlada desde la computadora. Las operaciones incluyen:
 - La habilitación del módulo.
 - La adquisición de datos.
 - La habilitación de cada canal.
 - La selección del tiempo de muestreo.
 - Inicialización del direccionamiento de RAM.
 - Control de la recuperación de los datos adquiridos.
- Todas las partes usadas serán baratas y fáciles de conseguir.

Las características del diseño anteriormente presentadas cuentan con las siguientes ventajas:

Operaciones realizadas por software. Tomando en cuenta que la única función del módulo de adquisición de datos es precisamente la de recopilar datos, las funciones inherentes a un osciloscopio digital pueden realizarse totalmente por software. Esto tiene como principal ventaja que es posible hacer virtualmente cualquier cambio que se requiera en el osciloscopio sin tener que cambiar el hardware que ya se encuentra disponible.

Mantenimiento más fácil. Al ubicarse en el exterior de la computadora, es posible darle mantenimiento a este módulo sin tener que abrir la computadora y sacar tarjetas de la misma.

Independencia de la computadora. Si se requiere conectar éste dispositivo a otro tipo de computadora no compatible con IBM, lo único que se tiene que cambiar es la interfase de datos con la computadora y el software de control, ya que éste módulo no tiene dependencias con la arquitectura de la PC.

7.3.2 Interfase de datos

La función de la interfase de datos implica que ésta debe de estar en el interior de la computadora, ya que requiere insertarse en las ranuras del bus de expansión. Considerando la arquitectura de la computadora personal, es posible tener un diseño basado en pocos circuitos.

Las características que deberá tener la tarjeta diseñada son:

- 2 puertos de 8 bits de entrada o salida.
- 2 puertos de 4 bits para control.
- Capacidad de introducir datos a la computadora por medio de DMA.
- Compatibilidad con buses tipo ISA.

Las dos primeras características pueden cumplirse fácilmente con un circuito que ya se ha estudiado en el capítulo de hardware de la IBM PC: el PPI 8255. Con respecto al DMA, se asumirá que la computadora cuenta con un controlador de DMA, siendo requerido únicamente la lógica de control para manejar el acceso directo a memoria. Con lo que respecta al último requisito, se usará la

información del capítulo 3 para diseñar la conexión con el bus de expansión de la IBM PC con arquitectura ISA.

Las ventajas que presenta el módulo que se acaba de describir son las siguientes:

Interfase universal. Puesto que la única función de éste módulo es la de transferir información a la PC, puede ser usado para cualquier otro propósito, siempre y cuando se respeten los protocolos de transferencia de información requeridos por la tarjeta.

Carga para el bus de expansión. Ya que se compone de pocos circuitos, no sobrecarga al bus con requerimientos altos de voltaje o corriente.

7.4 Sumario

En este capítulo se reunieron los elementos teóricos reunidos en las primeras partes de este documento para elaborar las especificaciones del diseño global del módulo de osciloscopio digital. El diseño se dividió en tres fases: el módulo de adquisición de datos, el módulo de interfase con la IBM PC y el módulo de software de control.

Una vez que se han realizado las especificaciones del diseño del hardware, en el siguiente capítulo se diseñará el primero de los elementos de hardware: la interfase con la IBM PC. En ese capítulo se describirán los elementos requeridos para obtener un circuito simple y muy eficaz para los propósitos de éste trabajo.

Capítulo 8

Interfase con la IBM PC

8.1 Introducción

Una vez que se han planteado los lineamientos a seguir para el diseño del hardware, tocaremos en éste capítulo la primera fase de hardware del proyecto: la comunicación de datos desde la IBM PC con el mundo externo. Para realizar este diseño, nos basaremos en el PPI (Programmable Peripheral Interface - Interfase a Periféricos Programable) 8255 de Intel para establecer un sistema de comunicaciones por puertos paralelos. Con lo que respecta al diseño de las comunicaciones por DMA, se asumirá que la PC cuenta con un controlador de DMA 8259 de Intel.

8.2 El PPI 8255

El PPI 8255 es un dispositivo de E/S programable de propósito general diseñado para usarse con microprocesadores Intel. Consta de tres puertos de 8 bits de E/S, los cuales pueden ser programados individualmente en dos grupos de 12 pines y ser usado en cualquiera de los tres modos de operación disponibles. En el primer modo (MODE 0), cada grupo de 12 pines de E/S puede ser programado en conjuntos de 4 para ser usados como entrada o salida. En MODE 1, el segundo modo, cada grupo puede ser programado para tener 8 líneas de entrada o salida. de los restantes 4 pines, 3 son usados para handshaking y señales de control de interrupción. El tercer modo de operación (MODE 2),

es un modo de bus bidireccional el cual usa 8 líneas como bus bidireccional y 5 líneas, tomando una línea del otro grupo, para handshaking.

8.2.1 Pines del PPI

La figura 8.1 muestra la disposición de pines del 8255. Consta de los siguientes elementos:

Buffer de bus de datos. Es un buffer de 8 bits bidireccionales triestado usado para interfasar al 8255 con un bus de datos. Los datos son transmitidos o recibidos por el buffer bajo la ejecución de instrucciones de entrada o salida por parte del CPU. Las palabras de control son también transmitidas a través del buffer de bus de datos.

Read/Write y lógica de control. La función de éste bloque es manejar todas las transferencias internas y externas de datos y bytes de control. La siguiente es la descripción de los pines que conforman este bloque:

CS (Chip Select). Una señal baja en este pin habilita la comunicación entre el 8255 y el CPU.

RD (Read). Una señal baja en este pin habilita al 8255 para enviar los datos al CPU en el bus de datos. En esencia, permite al CPU leer el PPI.

WR (Write) Una señal baja en éste pin permite que el CPU escriba un dato o un byte de control en el PPI.

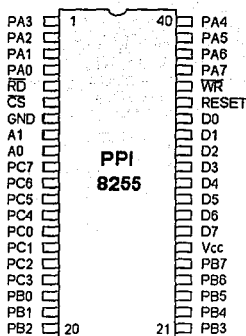


Figura 8.1 Esquema de pines del 8255

A₀ y A₁ (Port Select 1 y Port Select 2). Estas señales de entrada, en conjunción con las entradas RD y WR, controlan la selección de los tres puertos de datos del PPI o del registro de control. La tabla 8.1 muestra el acceso a los tres puertos del PPI y al registro de control a través de los valores de A₀ y A₁.

A1	A0	Puerto
0	0	A
0	1	B
1	0	C
1	1	Registro de control

Tabla 8.1 Acceso a los puertos del PPI

Reset. Una señal alta en este pin limpia el contenido del registro de control y todos los puertos son puestos como puertos de entrada.

Los tres puertos del PPI tienen las siguientes características:

- Puerto A.** Un buffer con latch de 8 bits de datos de salida y un latch de 8 bits de datos de entrada.
- Puerto B.** Un buffer con latch de 8 bits de datos de entrada y salida y un latch de 8 bits de datos de entrada.
- Puerto C.** Un buffer con latch de 8 bits de datos de salida y un buffer de 8 bits de datos de entrada (no hay latch en la entrada). Este puerto puede ser dividido en dos puertos de cuatro bits. Cada puerto puede ser usado para señales de control en conjunción con los puertos A y B.

8.2.2 Programación del PPI

Antes de poder usar el PPI, se requiere programarlo. Para realizar esto, se tiene que enviar un byte de control al registro de control del PPI, al cual se tiene acceso de acuerdo a lo mostrado en la tabla 8.1. El significado de cada uno de bits del byte de control para el ajuste de modo se puede observar en la figura 8.2.

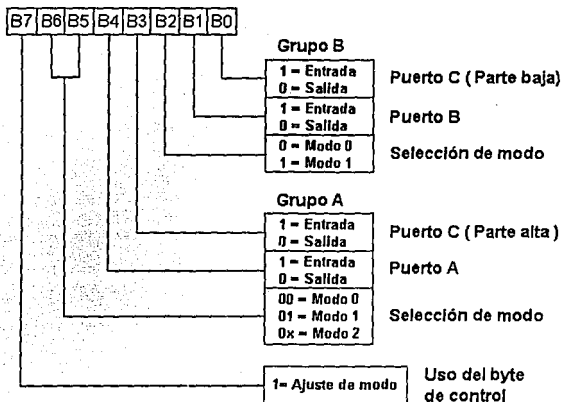


Figura 8.2 Byte de control del PPI 8255 (ajuste de modo)

8.3 Diseño de una interfase con la IBM PC

Como se indicó en los capítulos 2 y 3, en la IBM PC las comunicaciones entre el CPU y diversos dispositivos se realizan a través del canal de E/S. Se indicó también que estas comunicaciones se realizaban atendiendo a ciclos de bus, los cuales requieren que ciertas señales se encuentren en un estado en particular, atendiendo a una temporización dada. Haciendo un análisis de las señales presentes en el bus de expansión de la PC, las señales requeridas en el diseño de una interfase con la PC son AEN, las señales de voltajes de polarización, y los buses de direcciones y datos.

la señal **AEN** (address enable - habilitador de dirección) es una señal de salida activa en nivel alto. Es emitida por la lógica de control de DMA, e indica que un ciclo de bus de DMA está en progreso. Su uso es en la deshabilitación de las decodificaciones de los puertos de E/S de manera que direcciones de ciclos DMA no sean consideradas como una dirección de puerto. Esto es requerido en el puerto que se está diseñando para que cuando el sistema realice un ciclo de DMA no se tenga acceso a los puertos.

Al diseñar una interfase de este tipo se debe considerar la dirección a la cual se debe de conectar el puerto. En el diseño de la IBM PC las direcciones de puertos se encuentran repartidas entre los puertos de comunicaciones, video, y registros internos de hardware. Una de las opciones disponibles es usar la dirección asignada al puerto de juegos (200 Hex), o la dirección asignada específicamente a tarjetas de prototipo (300 Hex).

La figura 8.3 muestra el circuito requerido para conectar tres puertos paralelos en la IBM PC usando el PPI 8255 en el modo MODE 0. Como se observa, se compone de tres elementos: el PPI, el decodificador de direcciones y un banco de switches para cambiar direcciones. En este diseño los rangos de direcciones son decodificados por IC2, un demultiplexor TTL 74LS138. Cuando la línea de dirección A9 se encuentra en nivel alto y la señal AEN se encuentra en nivel bajo, el bloque de switches puede seleccionar las direcciones mostradas en la tabla 8.2.

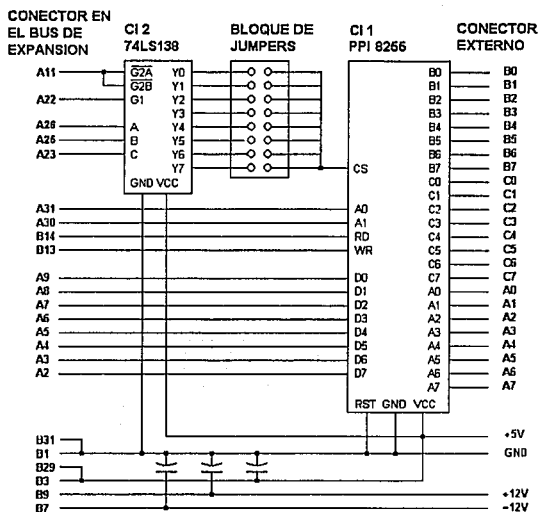


Figura 8.3 Puerto paralelo para la IBM PC

Switch en ON	Dirección	Switch en ON	Dirección
1	200 Hex	5	300 Hex
2	220 Hex	6	320 Hex
3	240 Hex	7	340 Hex
4	260 Hex	8	360 Hex

Tabla 8.2 Selección de puertos en la tarjeta de interfase con la IBM PC.

Esto permite que se use la dirección del puerto de juegos, o la de prototipos de tarjetas. Tener una dirección programable por un switch es importante, ya que permite evitar el uso de direcciones que correspondan a otros dispositivos presentes en la IBM PC.

La programación del PPI, suponiendo que el circuito se ajusta a una dirección base de 300H, se realizará escribiendo un byte en la dirección 303H. Este byte debe ajustarse a lo establecido en la figura 8.2. Asumiendo que se quiere que los puertos A y B sean de entrada, la parte alta del puerto C sea de salida y la parte baja del puerto C sea de entrada, el byte de control debe ser 93H (10010011 binario).

8.4 Comunicaciones con la IBM PC por DMA

La figura 8.4 muestra el circuito requerido para que la IBM PC acepte datos externos de una interfase de 8 bits por medio de acceso directo a memoria.

Para transferir un byte de datos (asumiendo que los dispositivos de DMA han sido inicializados), la interfase externa aplica los datos en el puerto de entrada y pone una petición de escritura por DMA, elevando la señal REQUEST. Cuando esta señal hace una transición positiva, fija los datos de entrada en el bus y solicita un ciclo de DMA. En el momento en que la señal DACK asociada se activa, los datos se transfieren a la memoria de la computadora y la petición es borrada. Esta acción es repetida cada vez que la señal REQUEST tiene una transición positiva y el conteo terminal del DMA no ha sido alcanzado.

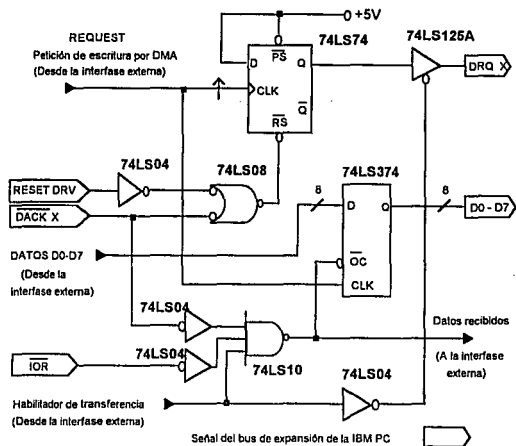


Figura 8.4 Interfase con la IBM PC por DMA

8.5 Sumario

En este capítulo se diseñó una tarjeta de comunicaciones para la IBM PC basada en un sistema de puertos paralelos y por acceso directo a memoria. La parte dedicada a puertos se fundamentó en el 8255 de Intel. La flexibilidad y potencia de este circuito permitió realizar el diseño con solamente 2 circuitos integrados y un banco de switches. Sumando esto al diseño del circuito para recibir datos por DMA (el cual requirió 6 circuitos integrados), da como resultado una tarjeta que no solamente puede ser usada para este proyecto, sino también en diversas tareas de control digital, tales

como el control de un brazo robot o de un horno, o para la adquisición de datos en tiempo real.

Una vez que se ha diseñado la primera fase de hardware de este proyecto, los próximos dos capítulos se orientan al diseño de la fase de adquisición de datos. Para esto, el siguiente capítulo tocará el tema de convertidores A/D, para dar paso posteriormente al capítulo dedicado al diseño del módulo de osciloscopio digital.

Capítulo 9 Conversión A/D

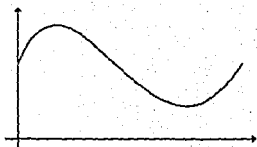
9.1 Introducción

Antes de comenzar el diseño del módulo de hardware del osciloscopio digital, éste capítulo analiza el proceso de la conversión de señales analógicas a digitales (conversión A/D). Ya que la computadora manipula internamente información digital y las señales externas a ella son analógicas (aún las llamadas señales digitales), es requerido incluir en el diseño una etapa de conversión para poder manejar la información que la computadora reciba. El presente capítulo comienza discutiendo los detalles de la conversión A/D, pasando después a una descripción de los diferentes métodos de conversión y de los factores a considerar al seleccionar un convertidor.

9.2 Conversión de valores analógicos a digitales

Un convertidor A/D toma un voltaje de entrada analógico y después de cierto tiempo produce un código de salida digital que representa a la entrada analógica. Las fases requeridas para llevar esto a cabo son:

Señal analógica



Toma de muestras de la señal en periodos de tiempo T .

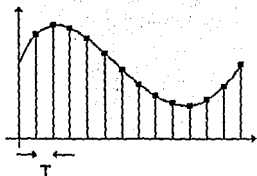


Figura 9.1 Muestreo

Muestreo. Antes de convertir la señal, se requiere tomar muestras de ella en unidades de tiempo T . Se denomina *frecuencia de muestreo* a la frecuencia con la que estas muestras son tomadas. La figura 9.1 ilustra la toma de muestras de una señal. Debe notarse que dada la naturaleza continua de una señal analógica y al tiempo finito T entre muestras, se pierde información entre cada muestra que se toma. De acuerdo con el teorema de **Shannon** o **Nyquist**, la información muestreada puede ser usada para recuperar la señal original sin pérdidas si se usa como mínima frecuencia de muestreo el doble de la máxima frecuencia que presente la señal a muestrear. Por ejemplo, si se desea muestrear sonido, y teniendo en cuenta que el rango audible va de 20 a 20,000 Hertz, la frecuencia de muestreo mínima debe ser 40 KHz.

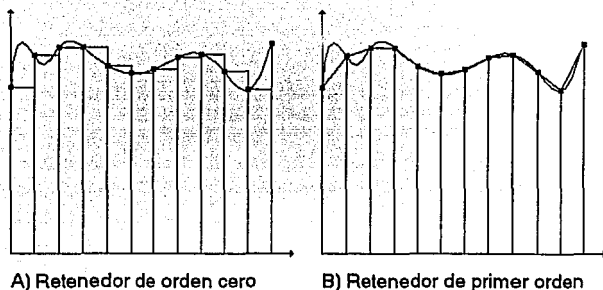


Figura 9.2 Retenedores de orden cero y primer orden

Retención. Como se observa en la figura 9.1, después de realizado el muestreo se obtiene el equivalente a un tren de pulsos multiplicado por la señal analógica. Para que esta información sea utilizable para el convertidor A/D, es necesario conservar constante durante el tiempo T entre muestras el valor que el muestreador ha tomado tal como se observa en la figura 9.2a, con el fin de que el convertidor A/D tenga una referencia constante durante el tiempo de conversión. Este requerimiento es satisfecho usando un retenedor de orden cero. Los efectos de otro tipo de retenedor, el retenedor de primer orden, se muestran en la figura 9.2b. Si bien la señal obtenida es más cercana a la original, no le proporciona al convertidor una señal constante.

Cuantización. La diferencia entre las señales analógicas y las señales digitales se basa fundamentalmente en el rango de valores que pueden alcanzar. Una señal analógica puede tener cualquier valor en el tiempo, mientras que una señal digital tiene una resolución finita, determinada por el número de bits que se

usarán para representar el dato. Si la salida del convertidor A/D consiste en n bits, el número de niveles discretos de salida está fijo a 2^n . Para una correspondencia uno a uno, el rango de entrada a escala completa (FS: full scale) debe de ser cuantizado dentro de este mismo número de niveles. Cada uno de tales niveles es el valor analógico por el cual dos códigos adyacentes difieren. A esto se le llama el tamaño del bit menos significativo (o tamaño del LSB). Así:

$$Q = \text{LSB} = \text{FS} / 2^n$$

donde Q es el tamaño del valor de cuantización, LSB se refiere al valor analógico del bit menos significativo, y FS se refiere al rango analógico de entrada en escala completa.

Código binario

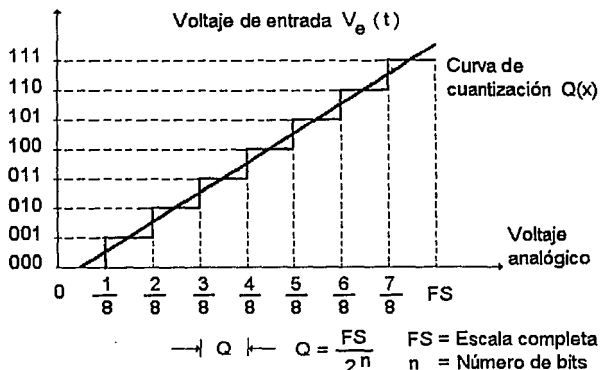
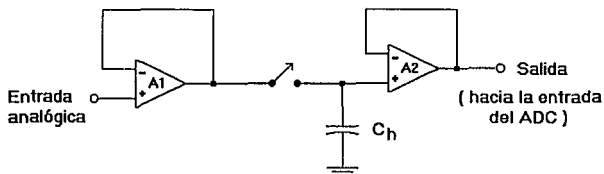


Figura 9.3 Cuantización en un convertidor A/D ideal

La figura 9.3 muestra esta relación de conversión en un convertidor ideal de 3 bits. La escalera de cuantización muestra los valores digitales de salida del convertidor y la rampa representa los valores analógicos de entrada. Nótese que el valor 111 no se encuentra asignado a FS, sino a $7 \times FS/8$, ya que el código 000 se encuentra asignado al 0 de entrada.

9.3 Circuitos de muestreo y retención

Cuando un voltaje analógico se conecta directamente a la entrada de un convertidor A/D, el proceso de conversión puede ser afectado adversamente si el voltaje analógico varía durante el momento de la conversión. La estabilidad del proceso de conversión puede depurarse utilizando un circuito de muestreo y retención para contener constante la entrada de voltaje analógico en tanto que la conversión A/D se lleva a cabo. Un diagrama simplificado del circuito de muestreo y retención (S/H : Sample and Hold) se muestra en la figura 9.4.



Interruptor cerrado: Modo de muestreo

Interruptor abierto: Modo de retención

Figura 9.4 Circuito de muestreo y retención

El circuito S/H contiene un amplificador de entrada A1 que presenta una alta impedancia a la señal analógica, y tiene una impedancia de salida baja que puede cargar rápidamente al capacitor de retención Ch. El capacitor será conectado a la salida de A1 cuando el interruptor digitalmente controlado este cerrado. Esto se llama operación de muestreo. El interruptor estará cerrado el tiempo suficiente para que Ch se cargue al valor regular de la entrada analógica. Por ejemplo, si el interruptor está cerrado al tiempo t_0 , la salida A1 rápidamente cargará Ch hasta un voltaje V_0 . Cuando el interruptor se abra, Ch contendrá este voltaje de manera que la salida de A2 aplique éste voltaje al convertidor A/D. El amplificador de salida A2 presenta una alta impedancia de entrada que no descargará el voltaje del capacitor apreciablemente durante el momento de la conversión del convertidor A/D, de modo que éste recibirá esencialmente un voltaje de entrada de dc de V_0 .

La cantidad de tiempo que el interruptor tiene que permanecer cerrado se llama tiempo de adquisición y depende del valor de Ch y de las características del circuito S/H. El LF198, por ejemplo, es un circuito integrado S/H que tiene un tiempo de adquisición común de 4 microsegundos para $Ch = 1000 \text{ pF}$ y 20 microsegundos para $Ch = 0.01 \text{ }\mu\text{F}$.

9.4 Características en un convertidor A/D

Resolución. Este término se define como el más pequeño cambio requerido en la entrada analógica de un convertidor A/D para cambiar su código de salida en un nivel digital. Puede expresarse como un porcentaje de la escala completa, en milivolts por un determinado rango de entrada, o simplemente como el número de bits en el código de cuantización.

Exactitud. La exactitud en convertidores A/D se define como la diferencia entre el voltaje de entrada verdadero y el código binario de salida equivalente. Es llamada exactitud absoluta cuando se especifica en volts reales. Es más comúnmente especificada relacionando a la señal analógica con el tamaño del LSB y es llamada exactitud relativa. En cualquier caso es la máxima suma de todos los errores de conversión, incluyendo el de cuantización (éste se define más adelante).

Tiempo de conversión. Después de que un comando START es recibido en un convertidor A/D, le lleva un tiempo finito T_c o tiempo de conversión antes de que pueda proporcionar datos válidos. La duración del tiempo de conversión depende del método de conversión usado, como se verá en posteriores secciones.

9.5 Tipos de error en convertidores A/D

Los convertidores pueden desviarse de las características ideales de la figura 9.3 en varias formas, algunas de las cuales se observan en la figura 9.5. Los errores más comunes son los siguientes:

Error de offset. Se caracteriza porque la función de transferencia está recorrida con respecto a su valor ideal. Este offset o "error de cero" está definido por el valor analógico con el cual la función de transferencia falla al pasar por el cero.

Error de ganancia. Este se presenta cuando la pendiente de la función de transferencia difiere de su valor ideal, generándose un error de pendiente o ganancia. Para la mayoría de los convertidores A/D en el mercado los errores de offset y de

ganancia son muy pequeños o pueden ser virtualmente eliminados mediante ajustes iniciales.

Error de linealidad integral. Se define como la máxima desviación de la función de transferencia de sus características ideales de linealidad cuando los errores de offset y de ganancia son cero.

Error de linealidad diferencial. Se presenta cuando existe una desviación de los códigos de cuantización con respecto a su valor ideal. Si este error es mayor a 1 LSB, la salida puede generar un código de cuantización perdido.

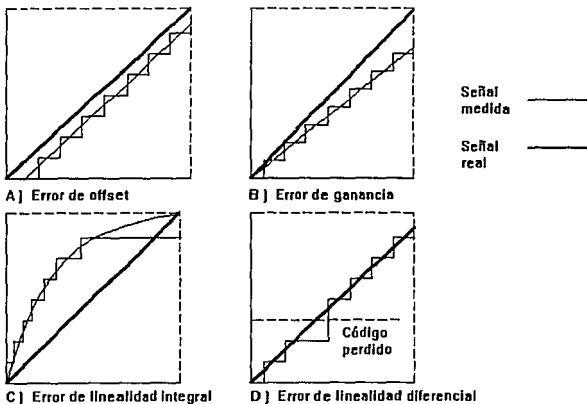


Figura 9.5 Errores en convertidores A/D

Error de cuantización. Considérese un convertidor A/D con una resolución de 8 bits, los cuales permiten usar $2^8=256$ niveles discretos diferentes. Suponiendo que esta resolución fuera usada para representar un rango de valores de 0 a 10 v, se tendría que el mínimo valor que se puede conseguir es $10/256 = 0.039$ v. Esto implica que entre cada valor binario existirá una diferencia de 39 mV. Si se intentara medir un valor entre 39 y 78 mV, se encontraría que no existe un valor binario que permitiera representarlo con exactitud. En estos casos se tiene que tomar uno de los dos valores digitales más cercanos al valor real, siendo el peor caso cuando la señal analógica está exactamente entre dos valores digitales (es decir $\pm 1/2$ LSB). A este tipo de error se le llama error de cuantización y existe en todos los convertidores A/D. Únicamente puede ser minimizado aumentando la resolución en bits del convertidor.

9.6 Técnicas y tipos de convertidores A/D

9.6.1 Convertidor A/D con rampa digital

La figura 9.6 presenta el diagrama completo de un convertidor A/D con rampa digital. Su operación es como sigue:

1. Se aplica un pulso "START" positivo, el cual devuelve el contador a cero. También inhibe a la compuerta AND de manera que ningún pulso del reloj pase hacia el contador cuando el pulso "START" es ALTO.

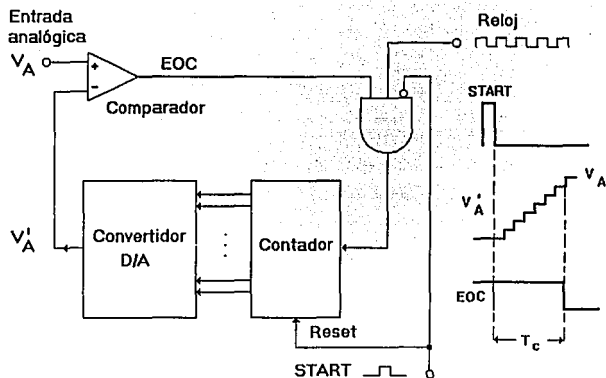


Figura 9.6 Convertidor A/D de rampa digital

2. Con el contador en cero, $V_A' = 0$, de modo que la salida del comparador es ALTA (supóngase que V_A es algún voltaje positivo).
3. Cuando el pulso "START" retorna a BAJO, la compuerta AND es activada y se permite a los pulsos entrar en el contador.
4. A medida que el contador avanza, la salida V_A' del D/A crece en etapas de voltaje iguales a su resolución.
5. Esto continúa hasta que V_A' llega a una etapa que excede a V_A en una cantidad igual a V_T o mayor. En este punto, la salida del comparador pasa a BAJA, inhibiendo los pulsos

que van al contador de manera que éste se haya detenido en un conteo que sea la representación digital deseada de VA. El proceso de conversión ahora está completo. La transición de ALTA a BAJA en la salida del comparador señala el final de la conversión. A la señal a menudo se le llama EOC (End Of Conversion - fin de conversión).

La resolución del convertidor es igual a la resolución del convertidor D/A que éste contiene. El voltaje de salida VA' del D/A es una onda en escalera que asciende en etapas discretas hasta que excede de VA. En consecuencia, VA' es un valor aproximado del valor de VA.

En el convertidor con rampa digital, el contador empieza en cero y cuenta hasta VA' > VA. El tiempo que se lleve completar la conversión dependerá del valor de VA. Un valor mayor de VA requerirá más etapas antes de que el voltaje de escalera exceda de VA. El tiempo máximo de conversión ocurre cuando VA es ligeramente menor que la escala completa, de manera que la escalera tiene que llegar a la etapa de escala completa a fin de que VA' > VA. En consecuencia, se tiene

$$T_c(\text{máx}) = 2^n - 1 \text{ ciclos de reloj}$$

Algunas veces se especifica el tiempo promedio de conversión el cual no es más que la mitad del tiempo máximo de conversión. Para el convertidor con rampa digital, esto sería:

$$T_c(\text{prom}) = T_c(\text{máx})/2 = 2^{n-1} - 1/2 \text{ ciclos de reloj}$$

La desventaja principal del método de la rampa digital es que el tiempo de conversión se duplica esencialmente por cada bit que se agrega al contador, de manera que la resolución puede ser mejorada solamente con un Tc mayor o más largo. Esto hace de

éste tipo de convertidor A/D inadecuado para aplicaciones donde tienen que efectuarse conversiones A/D repetitivas de una señal analógica que cambia rápidamente. Sin embargo, para aplicaciones de baja velocidad la simplicidad relativa del convertidor con rampa digital es una ventaja sobre los convertidores A/D de alta velocidad más complejos.

9.6.2 Convertidor A/D de rampa dual

La figura 9.7 ilustra la técnica de conversión A/D de rampa dual. El voltaje de entrada V_i es integrado por un intervalo de tiempo fijo T_1 , el cual generalmente corresponde a la máxima cuenta de un contador interno. Al final de este intervalo el contador se restaura a cero y la entrada del integrador se conmuta a la señal de referencia negativa $-V_r$. La salida del integrador se decrementa entonces linealmente durante un tiempo T_2 hasta que alcanza el valor de cero cuando el contador es detenido y el integrador es restaurado. La carga ganada por el capacitor del integrador durante el intervalo de tiempo T_1 debe ser igual a la carga perdida durante el tiempo T_2 , de tal forma que:

$$T_1 V_i(\text{prom}) = T_2 V_r$$

y por lo tanto:

$$T_2/T_1 = V_i(\text{prom})/V_r = \text{constante}$$

Debe notarse que la razón de tiempos T_2/T_1 es también el conteo binario relativo al conteo total del contador. De ahí que la cuenta al final de T_2 es la palabra de salida del convertidor A/D de rampa dual.

La técnica de rampa dual ofrece como mayor ventaja su excelente rechazo al ruido. Puesto que el voltaje de entrada es integrado durante un período de tiempo, cualquier ruido de alta frecuencia sobre la señal de entrada es cancelado. Además, el tiempo T_1 puede ser seleccionado para eliminar casi completamente el ruido a frecuencias que sean múltiplos de $1/T_1$. Comúnmente se selecciona por esto la frecuencia de la línea (60 Hz).

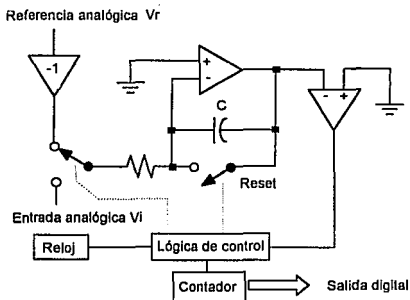


Figura 9.7 Convertidor A/D de rampa dual

La mayor limitación de este método es su baja velocidad de conversión y el hecho de que el tiempo de conversión depende del voltaje de entrada analógico, como puede observarse en la figura 9.8.

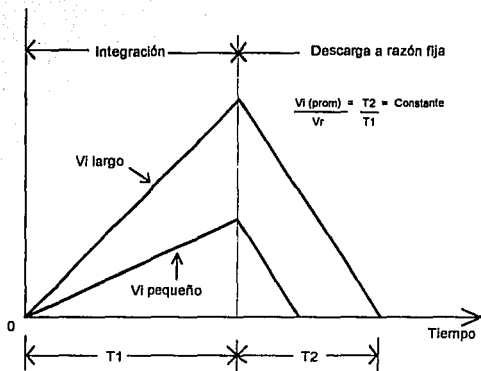


Figura 9.8 Formas de onda del circuito de conversión A/D de rampa dual

9.6.3 Convertidor A/D con convertidor de voltaje a frecuencia

El esquema de la figura 9.9 muestra la técnica de conversión A/D por conversión de voltaje a frecuencia. El voltaje analógico de entrada es convertido por un convertidor de voltaje a frecuencia (VFC) de alta precisión a un tren de pulsos con una frecuencia proporcional al voltaje de entrada. Un contador entonces genera la palabra digital de salida contando esos pulsos sobre un intervalo fijo de tiempo. Al igual que la técnica de rampa dual, se caracteriza por bajas velocidades pero buena inmunidad al ruido.

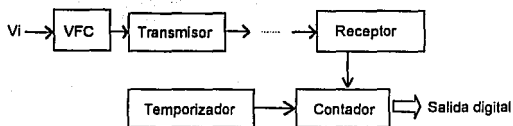


Figura 9.9 Convertidor A/D por conversión de voltaje a frecuencia

9.6.4 Convertidor A/D de aproximaciones sucesivas

Este es uno de los tipos más ampliamente utilizados de convertidores A/D. Tiene circuitos más complejos que el convertidor A/D con rampa digital pero un tiempo de conversión mucho más corto que los anteriormente descritos. Además, los convertidores de aproximaciones sucesivas (SAC) tienen un valor fijo de tiempo de conversión que no depende del valor de la entrada analógica.

La disposición básica, como se muestra en la figura 9.10, es semejante a la del convertidor con rampa digital. Sin embargo, el SAC no utiliza un contador para dar la entrada en el bloque del convertidor D/A pero en cambio usa un registro. La lógica de control modifica el contenido del registro bit por bit hasta que los datos del registro son el equivalente digital de la entrada analógica VA (dentro de la resolución del convertidor). El proceso se efectúa de la siguiente manera:

1. La lógica de control fija el MSB del registro ALTO y todos los otros bits BAJOS. Esto produce un valor de VA' en la salida del DAC igual al valor del MSB. Si VA' es ahora mayor que VA, la salida del comparador, COMP, pasa a BAJA y ocasiona que la lógica de control regrese el MSB a BAJO. En caso contrario, el MSB se conserva ALTO.

2. La lógica de control fija el siguiente bit del registro en 1. Esto produce un nuevo valor de VA'. Si este valor es mayor que VA, COMP pasa a BAJO para indicar a la lógica de control que regrese el bit a 0. En caso contrario, el bit se mantiene en 1.
3. Este proceso se continúa para cada uno de los bits del registro. Este proceso de ensayo y error requiere un ciclo del reloj por bit. Después de que todos estos bits han sido probados, el registro contiene el equivalente digital de VA.

La misma operación ocurre en SAC's más prácticos que utilizan un número mayor de bits para lograr una mejor resolución. En términos generales, un SAC de N bits requerirá N ciclos del reloj para realizar la conversión sin importar el valor de VA. Esto se debe a que los circuitos de control tienen que ensayar un 1 en cada posición del bit para ver si se necesita o no.

Ya que los SAC tienen tiempos de conversión relativamente rápidos, su uso en aplicaciones de adquisición de datos permitirá que se adquieran mas valores de datos en un intervalo de tiempo dado. Esto puede ser muy importante cuando los datos analógicos cambian a una razón de tiempo relativamente rápida.

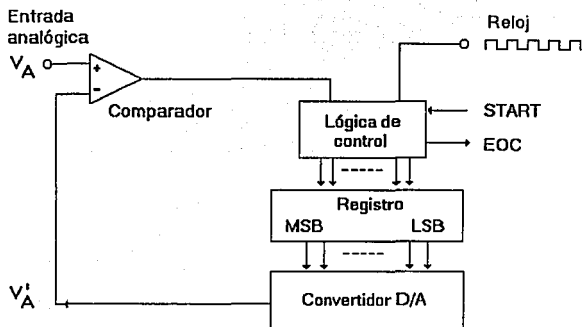


Fig 9.10 Convertidor A/D de aproximaciones sucesivas

9.6.5 Convertidor A/D de ráfaga

Este tipo es el convertidor A/D de más alta velocidad de que se dispone, pero requieren muchos más circuitos que para los otros tipos. Por ejemplo, un convertidor de ráfaga de 6 bits necesita 63 comparadores analógicos, una unidad de 8 bits requiere 255 y un convertidor de 10 bits necesita 1023. El gran número de comparadores que se necesita limita a los convertidores de ráfaga discretos al nivel de 6 bits por el precio y el espacio que ocupa. Los convertidores de ráfaga con IC se tienen a disposición actualmente en unidades de 8 bits y muchos fabricantes predicen que las unidades de 9 y 10 bits inundarán el mercado en un futuro cercano.

El convertidor de ráfaga de la figura 9.11 tiene una resolución de 3 bits y un tamaño de etapa de 1 V. El divisor de voltaje fija niveles

de referencia para cada comparador de manera que haya siete niveles correspondientes a 1 V (valor del LSB), 2 V, 3 V, ..., y 7 V (escala completa). La entrada analógica, VA, se conecta a la otra entrada de cada comparador.

Con $VA < 1 V$, todas las salidas del comparador C1-C7 serán ALTAS. Con $VA > 1 V$, una o mas salidas del comparador serán BAJAS. Las salidas del comparador se alimentan a un codificador de prioridad BAJO activo que genera una salida binaria correspondiente a la salida del comparador con el número mas grande que es BAJA. Por ejemplo, cuando VA esta entre 3 y 4 V, las salidas C1, C2 y C3 serán BAJAS y todas las otras serán ALTAS. El codificador de prioridad responderá solamente a salida BAJA en C3 y producirá una salida binaria CBA = 011, el equivalente digital de VA, dentro de la resolución de 1 V. Cuando VA es mayor que 7 V, C1-C7 serán todas BAJAS y el codificador producirá CBA = 111 como el equivalente digital de VA. La tabla 9.1 muestra las respuestas para todos los posibles valores de la entrada analógica.

VA	C1	C2	C3	C4	C5	C6	C7	C	B	A
< 1V	1	1	1	1	1	1	1	0	0	0
>1V, <2v	0	1	1	1	1	1	1	0	0	1
>2V, <3V	0	0	1	1	1	1	1	0	1	0
>3V, <4V	0	0	0	1	1	1	1	0	1	1
>4V, <5V	0	0	0	0	1	1	1	1	0	0
>5V, <6V	0	0	0	0	0	1	1	1	0	1
>6V, <7V	0	0	0	0	0	0	1	1	1	0
>7v	0	0	0	0	0	0	0	1	1	1

Tabla 9.1 Tabla de verdad para el convertidor de ráfaga de la figura 9.11

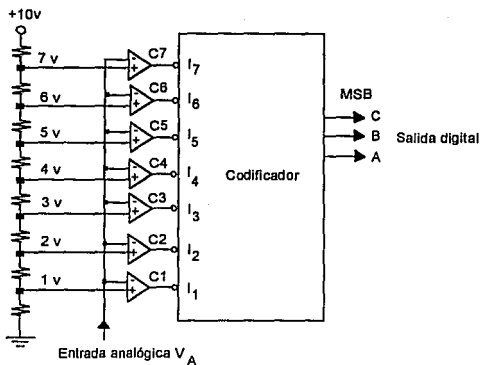


Figura 9.11 Convertidor A/D de ráfaga

El convertidor de ráfaga no utiliza señales del reloj debido a que no necesitan distribución ni secuenciación. El proceso de conversión se lleva a cabo tan pronto como se aplica V_A y el tiempo de conversión depende únicamente de las demoras en la propagación de los comparadores y el codificador. Un convertidor de ráfaga común puede tener un tiempo de conversión de 50 ns.

Ahora debe resultar sencillo observar por que los circuitos del convertidor de ráfaga aumentan dramáticamente cuando el número de bits deseados crece. El convertidor de 3 bits de la figura 9.11 requiere siete comparadores ya que hay $2^3 = 8$ posibles niveles de voltaje 0 V, 1 V, 2 V, ..., 7 V. No se requiere ningún comparador para el nivel 0 V. Para un convertidor de ráfaga de 8 bits habría $2^8 = 256$ niveles de voltaje inclusive 0 V, de manera que se requerirían 255 comparadores. Por tanto, en

general, un convertidor de rfaga de N bits requiere $2^n - 1$ comparadores.

La ventaja principal del convertidor de rfaga es su velocidad de operaci3n; sin embargo, logra esta velocidad de operaci3n a expensas de mayor complejidad de los circuitos. El alto costo de estos convertidores limita su uso a aquellas aplicaciones donde la alta velocidad es un requisito primordial.

9.7 Parmetros de selecci3n de un convertidor A/D

Para seleccionar un convertidor A/D se deben de considerar los siguientes factores:

Rango de voltajes de entrada del convertidor. La especificaci3n del rango de voltajes de entrada es importante por dos razones. En primer lugar, determina si se puede conectar directamente el convertidor al origen de la seal. En general este no es el caso, ya que la fuente puede generar un nivel de voltaje a la salida que puede ser demasiado pequeo o demasiado alto para sensarse directamente. Si el nivel de voltaje es demasiado bajo, se tiene que amplificar la seal usando un amplificador operacional. Si la seal es demasiado grande, una red de resistencias bastar para reducir la seal a un nivel aceptable. En segundo lugar, conocer el rango de voltajes de entrada del convertidor permite conocer el mnimo valor que puede ser medido dividiendo el rango entre la resoluci3n. Por ejemplo, si la entrada a medir vara de 0 a 5 volts y el dispositivo tiene 8 bits de resoluci3n, el mnimo valor que se puede medir es igual a $5/2^8 = 0.0195$ volts.

Capacidad de medir voltajes diferenciales, positivos y negativos. Esto puede ser importante, particularmente si la salida

de voltaje no está referida a una tierra conocida o si la salida de voltaje puede ser bipolar.

Voltaje de referencia de precisión en el convertidor. La mayoría de las técnicas de conversión requieren un voltaje de referencia de precisión. Si no es parte del dispositivo, deberá ser añadido como un componente separado, agregando más costos y espacio en la tarjeta.

Error de offset inducido por el convertidor. Si el nivel de voltaje que se está tratando de medir es muy bajo, el error de offset puede ser un factor importante. El error de offset debe ser mucho más pequeño que el voltaje más pequeño a ser medido.

Carga que presenta el convertidor a la fuente que está sensando. Se debe asegurar que el dispositivo no altera la fuente y modifica el verdadero valor que se está midiendo. Si este es el caso, puede agregarse un amplificador operacional de ganancia unitaria con alta impedancia de entrada como buffer a la entrada del convertidor. Nótese que esto puede causar a su vez un error adicional de offset en el sistema.

Bits de resolución. Como se mencionó anteriormente, el número de bits de resolución determina el valor mínimo que puede ser medido por el sistema. Por ejemplo, un sistema de 8 bits mide 256 niveles y un sistema de 12 bits mide 4096 niveles sobre el rango de voltajes de entrada.

Representación del valor medido. Si el convertidor proporciona un bus compatible con microprocesador, será fácil interfazar éste con una PC. Por otro lado, si es difícil obtener los datos del dispositivo, tendrá un efecto adverso en el tiempo requerido para convertir un valor y tenerlo disponible.

Tiempo de conversión. Esta especificación es importante porque determina la razón a la cual el dispositivo puede muestrear y convertir los datos analógicos a un valor digital. Típicamente, entre más rápido sea el convertidor, será más caro.

9.8 Sumario

Este capítulo analizó las fases que componen al proceso de conversión: muestreo, retención y cuantización, los métodos existentes para convertir una señal analógica a digital, sus características y los factores a considerar al seleccionar un convertidor A/D.

De acuerdo a las características de cada uno de los métodos de conversión A/D, los más adecuados para éste proyecto son los de aproximaciones sucesivas y de ráfaga. La única desventaja en el método de ráfaga es la cantidad de elementos que componen a una convertidor A/D, lo cual eleva su precio comparado con uno de aproximaciones sucesivas. Por otro lado, la ventaja con los convertidores de aproximaciones sucesivas es que tienen una buena relación precio-respuesta, aunque son más lentos que un convertidor de ráfaga.

Una vez que se analizó el proceso de conversión A/D, en el siguiente capítulo se desarrolla el módulo de osciloscopio digital que tendrá como función adquirir los datos que la computadora ha de presentar en la pantalla.

Capítulo 10

Módulo de osciloscopio digital

10.1 Introducción

En este capítulo se desarrolla la etapa de hardware del proyecto dedicada a recibir la información analógica y transformarla en un formato digital comprensible para la computadora. Para llevar a cabo el diseño se describirá primero el sistema completo mediante un diagrama de bloques, explicando la función de cada uno de ellos. Una vez hecha la división por bloques, el diseño se reducirá a describir el circuito diseñado para cada módulo, indicando el enlace entre estos.

10.2 Descripción global del módulo

La figura 10.1 muestra un diagrama a bloques que se apega a las especificaciones de diseño establecidas en el capítulo 7. El diagrama muestra las siguientes partes:

Selector de módulo. Su función es la de permitir la comunicación entre el módulo de osciloscopio digital y la PC si esta envía el comando de activación adecuado. La presencia de este bloque permite tener varios módulos de instrumentación conectados y permitir que solamente uno de ellos envíe y reciba datos a la vez. Esta etapa debe ser incluida en todos los módulos de instrumentación que se pretenda diseñar para el sistema INDICO.

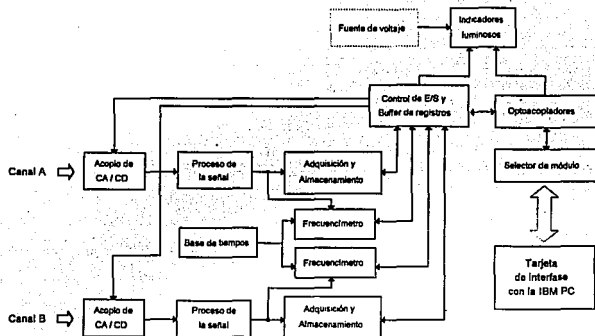


Figura 10.1 Diagrama a bloques del módulo de osciloscopio digital

Optoacopladores. La función de este bloque es aislar las señales de la PC con respecto a las del módulo de osciloscopio digital para que en caso de presentarse alguna falla en éste la PC no tenga ningún problema por corto circuito o sobrecargas.

Control de E/S (entrada/salida) y buffer de registros. En este bloque el módulo de osciloscopio digital selecciona el tipo de dato que enviará o recibirá de la PC. Los tipos de datos pueden ser: amplitud de la señal, frecuencia de la misma o registros de operación del osciloscopio.

Indicadores luminosos. Este bloque consiste en cuatro LED's que indican si el osciloscopio está encendido, si ha sido seleccionado por la PC para ser usado y el estado de la adquisición de datos en cada canal.

Fuente de voltaje. Este bloque se encarga de suministrar a todo el sistema la alimentación de voltaje requerida.

Acoplo de CA/CD.- Este bloque se encarga de seleccionar, a partir del estado de uno de los registros del osciloscopio, el tipo de acoplo de la señal de entrada.

Proceso de la señal. En este bloque se ajusta la amplitud de la señal al rango de adquisición del convertidor A/D (0-5 V).

Base de tiempos. Este bloque se encarga de generar señales de reloj para la temporización de diversas partes del sistema.

Frecuencímetro. Este bloque usa un contador para calcular la frecuencia de la señal de entrada, tomando como base una señal de reloj de frecuencia conocida generada por la base de tiempos.

Adquisición y almacenamiento. En este bloque se toma la señal de entrada y se toman las muestras usando un convertidor A/D. Cada una de las muestras tomadas es enviada a una memoria RAM para su posterior envío a la PC.

10.3 Selector de módulo

El circuito dedicado a la habilitación de las comunicaciones entre el módulo de osciloscopio digital y la IBM PC se muestra en la figura 10.2. Para describir este circuito recordemos que la interfase de comunicaciones con la IBM PC diseñada en el capítulo 8 cuenta con dos puertos de 8 bits y dos puertos de 4 bits, todos programables para entrada o salida de datos mediante un byte de control enviado al 8255 PPI contenido en la tarjeta. Tomemos como convención para referirnos a cada uno de esos

puertos la misma que toma el 8255 PPI programado en el modo 0: P_A para el primer puerto de 8 bits, P_B para el segundo puerto de 8 bits, P_{CH} para la parte alta del puerto tercer puerto (4 bits), y P_{CL} para la parte baja del tercer puerto C (4 bits).

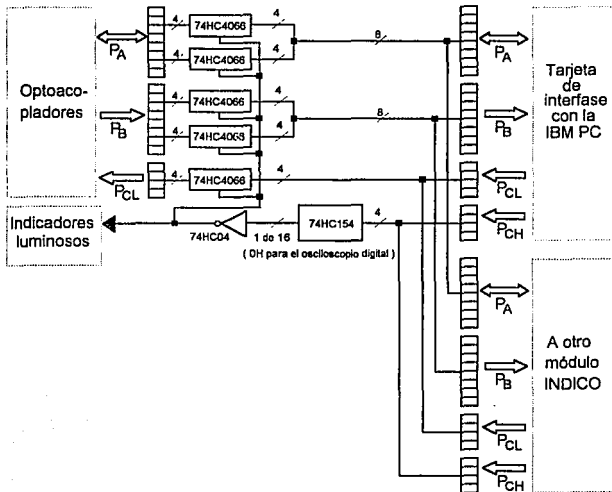


Figura 10.2 Circuito del selector de módulo

La tabla 10.1 indica el uso que el módulo de osciloscopio digital da a estos tres puertos.

Puerto	Datos transmitidos
P _A	Datos de E/S entre el osciloscopio y la PC.
P _B	Datos del osciloscopio hacia la PC.
P _{CH}	Selección del módulo de instrumentación.
P _{CL}	Selección del dato de E/S transmitido entre la PC y el módulo de instrumentación.

Tabla 10.1 Uso de los puertos en el módulo de osciloscopio digital

El circuito del selector de módulo consta de un decodificador de 4 líneas a 16 líneas 74HC154 conectado al puerto P_{CH} y de un banco de switches analógicos formado con circuitos 74HC4066. La introducción de un valor de 4 bits en el puerto P_{CH} permitirá seleccionar a 1 de 16 posibles módulos de instrumentación. Cuando uno de estos módulos ha sido seleccionado, el decodificador cierra los switches para que dejen pasar la información. Puesto que estos switches presentan la capacidad de dejar pasar la información bidireccionalmente, no es necesario lógica de control adicional. El banco de switches consta de 5 circuitos 74HC4066, cada uno de ellos con 4 switches. Estos switches se repartirán entre los 2 puertos de 8 bits P_A y P_B y el puerto de 4 bits P_{CL}.

La elección de circuitos lógicos de la familia 74HCxx obedece a que tienen un mejor aprovechamiento de la energía: las corrientes mínimas de entrada son hasta 400 veces menores que la familia LS, lo cual elimina la necesidad de considerar fanouts, ya que las entradas son prácticamente circuitos abiertos. Con lo que respecta a las corrientes de salida, son aproximadamente 10 veces mayores que en la familia LS.

Adicionalmente, la familia HC tiene una velocidad de operación tan buena como la de la familia LS. En todos los circuitos que se

muestran en este capítulo se observará que se ha favorecido su elección por encima de circuitos lógicos de otras familias.

10.4 Optoacopladores

En esta etapa se monta un circuito como el mostrado en la figura 10.3 para cada bit de los puertos P_A , P_B y P_{CL} . Considerando que el flujo de información en este circuito no es bidireccional, tendrán que usarse 2 de ellos para cada bit del puerto P_A , que como se indica en la sección 10.5, es bidireccional. Esto hace un total de 28 circuitos (16 para el puerto A, 8 para el puerto B y 4 para el puerto C).

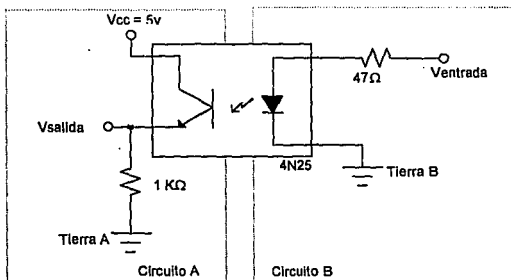


Figura 10.3 Circuito de optoacoplador.

El circuito optoacoplador usado es un 4N25, que proporciona un aislamiento de hasta 7.5 KV.

10.5 Control de E/S y buffer de registros

Esta parte se encarga de controlar la entrada y salida de información del módulo de osciloscopio digital, así como de almacenar los registros de éste. La existencia de un control de E/S se debe a que existen varios datos a transmitir y se cuenta con pocos puertos, de manera que el mismo puerto deberá de transmitir información diversa. Los tipos de datos que se pueden transmitir son tres:

- ❑ *Amplitud de la señal.* Este tipo de dato es enviado a la PC y corresponde a cada valor muestreado entregado por el convertidor A/D. Su longitud es de 8 bits.
- ❑ *Frecuencia de la señal.* Este dato también es enviado a la PC y corresponde al valor de la frecuencia de la señal medida. Su longitud es de 16 bits.
- ❑ *Registros.* Los registros son constantes de 8 bits enviadas o recibidas por la PC que definen varias características de operación del osciloscopio. Los registros definidos se dividen en dos grupos, dependiendo del sentido de flujo de la información. El grupo uno comprende a 3 registros de programación que son enviados por la PC al osciloscopio:

ESCFA - Escala de frecuencia. Indica el rango de frecuencias entre los cuales se encuentra la señal que se esta midiendo en el canal A. Los rangos son: 0-0.1 Hz, 0.1-1 Hz, 1-10 Hz, 10-100 Hz, 100-1000 Hz, 1-10 KHz, 10-100 KHz y 100 KHz - 1 MHz.

ESCFB - Escala de frecuencia. Indica el rango de frecuencias entre los cuales se encuentra la señal que se esta midiendo en el canal B.

BANDERAS 1 - Define características de operación del osciloscopio. En la tabla 10.2 se muestra el nombre y significado de cada bit dentro del registro.

Bandera	Significado	Valores
<i>HCA</i>	Habilitar canal A	1 - Activo. 0 - Inactivo.
<i>HCB</i>	Habilitar canal B	1 - Activo. 0 - Inactivo.
<i>ACOA</i>	Acoplo de señal en A	1 - C.A. 0 - C.D.
<i>ACOB</i>	Acoplo de señal en B	1 - C. A. 0 - C.D.
<i>PDF</i>	Petición del siguiente dato de frecuencia.	1 - Petición 0 - Espera
<i>ADSEL</i>	Habilitar convertidor A/D	1 - No habilitar. 0 - Habilitar.
<i>PDA</i>	Petición del siguiente dato de amplitud.	1 - Petición. 0 - Espera
<i>MSEL</i>	Habilitar RAM del DSO	1 - No habilitar. 0 - Habilitar.

Tabla 10.2 Descripción del registro de banderas 1

El grupo dos comprende un registro de señalización que el osciloscopio envía a la computadora:

BANDERAS 2 - Es un campo de 7 bits mediante el cual el osciloscopio señala a la computadora cuatro eventos. La tabla 10.3 muestra el nombre y significado de cada bit de este registro.

Bandera	Significado	Valores
<i>RECDSO</i> (4 bits)	Reconocimiento del osciloscopio.	AH - Activo
<i>FINMEMO</i>	Fin de lectura de RAM del osciloscopio.	1 - Fin 0 - Leyendo RAM
<i>FDA</i>	Fin de adquisición.	1 - Fin 0 - Adquiriendo
<i>FDCF</i>	Fin del conteo de frecuencia	1 - Fin del conteo 0 - Conteo en progreso

Tabla 10.3 Descripción del registro de banderas 2.

El circuito de la figura 10.4 muestra al bloque de control de E/S y buffer de registros. Consta de un decodificador de 4 líneas a 16 líneas 74HC154 conectado al puerto P_{CL} y a un bloque de buffers 74HC244 con salida triestado y a latches tipo D 74HC373, también con salida triestado.

Cuando el puerto P_{CL} recibe uno de los valores indicados en la tabla 10.4, el decodificador selecciona a uno de los circuitos conectados a él. Una vez que el circuito ha sido seleccionado, podrá transferir información por el puerto P_A o P_B , mientras que los demás circuitos ponen su salida en tercer estado. El sentido de flujo de la información desde el módulo de osciloscopio digital se muestra en la tercera columna de la tabla 10.4.

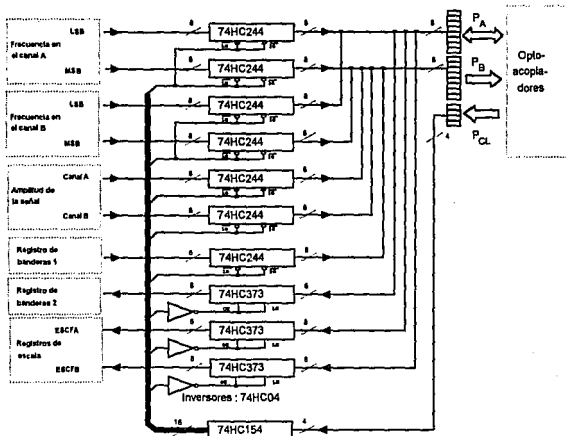


Figura 10.4 Circuito del control de E/S y buffer de registros

Valor en PCL	Tipo de dato seleccionado	E/S
0H	Frecuencia en el canal A	Salida
1H	Frecuencia en el canal B	Salida
2H	Amplitud en el canal A	Salida
3H	Amplitud en el canal B	Salida
4H	Reservado	
5H	Reservado	
6H	Registro de banderas 2	Salida
7H	Registro de banderas 1	Entrada
8H	Registro ESCFA	Entrada
9H	Registro ESCFB	Entrada

Tabla 10.4 Programación del selector de entrada/salida de datos

10.6 Indicadores luminosos

El bloque de indicadores luminosos tiene como función informarle al usuario de 4 eventos en el módulo de osciloscopio:

- ❑ *Encendido.* Este indicador se enciende cuando el módulo recibe voltaje de la fuente de alimentación.
- ❑ *Módulo seleccionado.* Este indicador se enciende cuando la PC selecciona al módulo de osciloscopio digital.
- ❑ *Adquisición de datos en progreso en el canal A.* Este indicador enciende cuando el osciloscopio se encuentra recibiendo datos en el canal A.
- ❑ *Adquisición de datos en progreso en el canal B.* Este indicador enciende cuando el osciloscopio se encuentra recibiendo datos en el canal B.

El circuito de control de los indicadores luminosos se muestra en la figura 10.5.

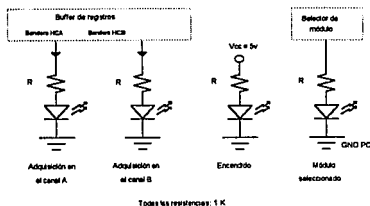


Figura 10.5 Circuito de los indicadores luminosos

Como se observa, es bastante simple. Consta de 4 diodos emisores de luz y 4 resistencias de 1 K Ω . Los primeros dos LEDs encienden o no dependiendo del estado de las banderas HCA y HCB dentro del registro de banderas 1. Los otros dos indicadores toman como señal de control al voltaje $V_{cc} = 5$ v del osciloscopio y a la señal proveniente del bloque de selección de módulos INDICO. Nótese que este último toma la tierra de la PC y no la del osciloscopio.

10.7 Fuente de voltaje

De acuerdo a la especificación, la fuente debe entregar voltajes de ± 5 V. Ya que los circuitos usados en este proyecto son en su mayoría de la familia 74HCxx, los cuales funcionan con una polarización de 2 a 6 V, no es necesaria una exactitud muy grande en el voltaje entregado por la fuente. Por otro lado, los circuitos de esta familia tampoco requieren demasiada corriente, por lo que no se requiere de una fuente de gran potencia. En vista a estos requerimientos, el circuito en la figura 10.6 representa a la fuente de voltaje a ocupar en este módulo.

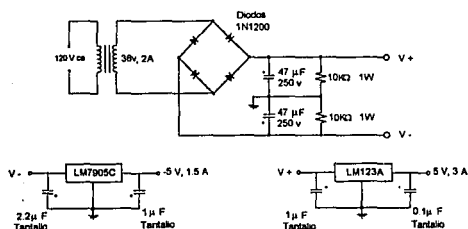


Figura 10.6 Circuito de la fuente de voltaje

La fuente de voltaje usa dos reguladores de voltaje para obtener valores de ± 5 volts. Para el caso de la salida de 5 V se usó un regulador LM123A y para la salida de -5 V un regulador LM7905C. Los diodos rectificadores que se usaron fueron los 1N1200, con capacidad de manejar voltajes de 100V y corrientes de 12 A.

La tabla 10.5 muestra los valores de salida y entrada para los reguladores de acuerdo a las especificaciones del fabricante. El asterisco en la última columna indica que éste no especificó el valor.

Circuito	V salida	Corriente	V entrada
LM123A	5v, ± 0.05 v	3 A	7.5 v a 20 v
LM7905C	-5v, ± 0.5 v	1.5 A	* a -35 v

Tabla 10.5 Especificaciones para los reguladores de voltaje usados

10.8 Acoplo de CA/CD

La figura 10.7 muestra el circuito para seleccionar el acoplo de corriente alterna o corriente directa en la señal de entrada para el canal A del osciloscopio. Para el canal B se usa otro circuito igual.

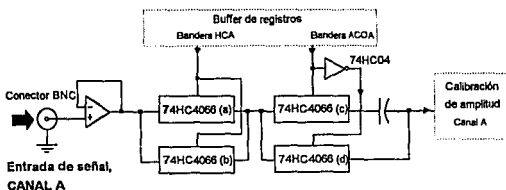


Figura 10.7 Circuito de acoplo de CA/CD en la entrada (1 por canal).

El circuito consta de un buffer conectado en la entrada de cada canal y de switches analógicos controlados por el estado del registro de banderas del osciloscopio. El circuito seleccionado para el buffer fue un LH2110, el cual contiene dos buffers de ganancia unitaria que le dan al osciloscopio una resistencia de entrada muy elevada (10^{10} ohms).

La selección del tipo de acoplo se realiza atendiendo al valor de las banderas ACOA y ACOB (ver tabla 10.2). Para lograr esto se usan switches analógicos 74HC4066. La elección de este tipo de switches obedece a que presentan una resistencia de solamente 30 ohms cuando se encuentran cerrados. Si se deseara reducir aún más la resistencia de entrada, pueden ponerse dos de estos switches en paralelo para obtener 15 ohms, tal y como lo indican los switches conectados junto al buffer en la figura 10.7.

10.9 Proceso de la señal

La figura 10.8 muestra al circuito que controla la amplitud de la señal.

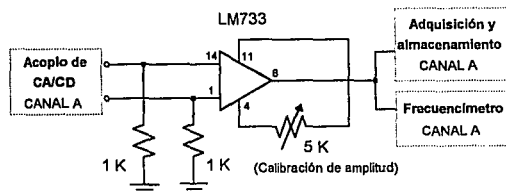


Figura 10.8 Circuito para el procesamiento de la señal (1 por canal)

Este circuito se basa en el amplificador diferencial para video LM733, el cual se programa con ganancia variable conectando los pines 4 y 11 a un potenciómetro. Esto permite realizar un ajuste de la ganancia con el propósito de calibrar al osciloscopio. El amplificador acepta entradas de ± 5 V y entrega un voltaje de 0 a 5 V a la salida.

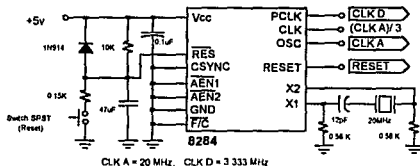
10.10 Base de tiempos

La figura 10.9 muestra a la base de tiempos, cuya función es la de generar pulsos de reloj para el frecuencímetro y para el módulo de adquisición de datos. El circuito se divide en tres partes. En la parte A se genera la señal del reloj principal y la de RESET. Esto se hace mediante el 8284, un circuito que Intel creó para proporcionar la frecuencia de reloj y la señal de RESET para el microprocesador 8086. En este caso, la señal principal de reloj que proporciona al osciloscopio esta dada por un cristal de 20 MHz. En lo que respecta a la señal de RESET, esta se activa (estado lógico "1") cuando el usuario reinicializa al osciloscopio presionando el switch tipo push-button del lado izquierdo del circuito. La resistencia de 10K Ω y el capacitor de 4.7 μ F tienen como finalidad asegurar que cuando se presione el switch, la señal de RESET se mantenga activa al menos 50 ms aproximadamente.

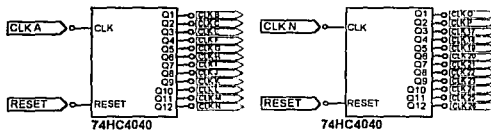
El circuito de la parte B toma la frecuencia de 20 MHz y la divide sucesivamente para generar 24 señales de reloj usando dos circuitos 74HC4040 (contadores binarios de rizo de 12 etapas).

Finalmente, en la parte C se toman las señales de frecuencia más alta y se conectan a dos circuitos 74HC244 (8 buffers triestado en cada uno), con la finalidad de filtrar el ruido presente en ellas.

A) RELOJ



B) DIVISORES DE FRECUENCIA



C) BUFFER

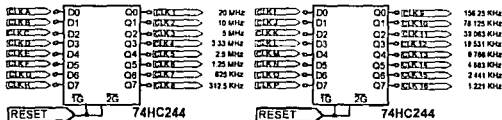


Figura 10.9 Circuito para la base de tiempos.

La tabla 10.6 muestra el nombre de la señal de reloj en el circuito de la figura 10.9 y su correspondiente frecuencia. En total, la base de tiempos entrega 26 señales de reloj al sistema y una señal de RESET.

Señal	Frecuencia	Señal	Frecuencia
CLK 1	20 MHz	CLK 14	4 882.813 Hz
CLK 2	10 MHz	CLK 15	2441.406 Hz
CLK 3	5 MHz	CLK 16	1220.703 Hz
CLK 4	3.333 MHz	CLK 17	610.3516 Hz
CLK 5	2.5 MHz	CLK 18	305.1758 Hz
CLK 6	1.25 MHz	CLK 19	152.5879 Hz
CLK 7	625 KHz	CLK 20	76.29395 Hz
CLK 8	312.5 KHz	CLK 21	38.14697 Hz
CLK 9	156.25 KHz	CLK 22	19.07349 Hz
CLK 10	78.125 KHz	CLK 23	9.536743 Hz
CLK 11	39 062.5 Hz	CLK 24	4.768372 Hz
CLK 12	19 531.25 Hz	CLK 25	2.384186 Hz
CLK 13	9 765.625 Hz	CLK 26	1.192093 Hz

Tabla 10.6 Frecuencias entregadas por la base de tiempos.

10.11 Frecuencímetro

El circuito para el frecuencímetro de 16 bits ha sido dividido en tres partes: creación de una señal de referencia, escalamiento de la señal (prescaler) y conteo de frecuencia. La primera de ellas, mostrada en la figura 10.10, corresponde a la creación de una señal digital de referencia basada en la señal de entrada en el canal A. Este circuito se basa en el amplificador diferencial para video LM733, el cual se programa con ganancia de 400 conectando los pines 4 y 11. Esto permite que señales mayores a 0.01 Vp-p sean amplificadas a un nivel tal que la señal pueda sea identificada como señal digital. A la salida del LM733 se conectan dos compuertas NAND schmitt trigger con el fin de perfeccionar los flancos de la señal y la amplitud de la misma.

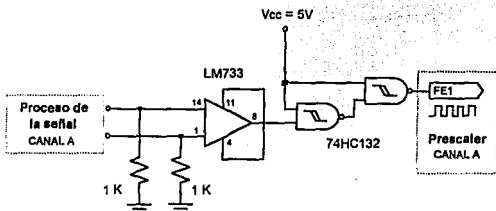


Figura 10.10 Circuito para crear una señal digital de referencia (1 por canal).

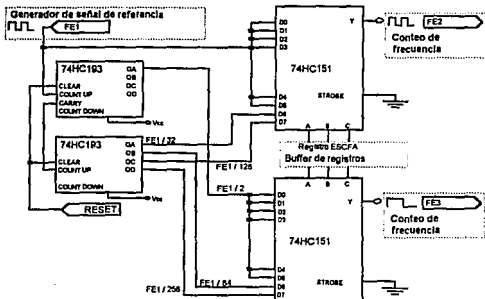


Figura 10.11 Circuito para el prescaler del frecuencímetro en el canal A (1 por canal).

La figura 10.11 muestra el circuito correspondiente al prescaler del canal A. La finalidad de esta etapa es la de dividir en frecuencia la señal de entrada cuando la frecuencia es mayor a 10,000 Hz. El circuito toma la señal digital de referencia FE1 entregada en la etapa mostrada en la figura 10.10 y la divide entre 2, 4, 8, 16, 32, 64, 128 y 256 usando dos contadores binarios síncronos 74HC193

conectados en cascada. El valor almacenado en el registro ESCFA permite a un multiplexor de 8 canales 74HC151 seleccionar una de estas 8 señales para transferirla a la etapa de conteo de frecuencia como la señal FE2. La otra señal entregada, FE3, corresponde a la mitad de la frecuencia en FE2.

En la figura 10.12 se observa la etapa de conteo de frecuencia. En esta etapa es donde se calcula el valor de la frecuencia de entrada basándose en la señal digital de referencia y en una frecuencia de reloj conocida.

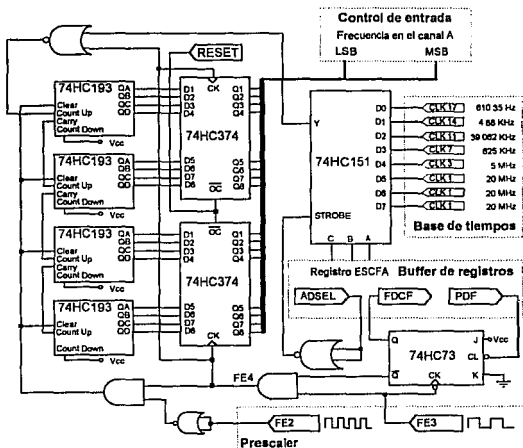


Figura 10.12 Circuito para el conteo de frecuencia en el canal A (1 por canal).

En la figura 10.13 se observa el diagrama de tiempos bajo el cual opera este circuito. La frecuencia que se desea conocer está en la señal FE2 y la mitad de esa frecuencia en FE3. La señal CK corresponde a la frecuencia entregada por la base de tiempos que el multiplexor 74HC151 ha seleccionado a partir del valor en el registro ESCFA. Las señales CK2 y CL corresponden a las señales en las terminales "COUNT UP" y "CLEAR" en el bloque de contadores binarios 74HC193 de la figura 10.12. CK2 será la señal de reloj bajo la cual operen los contadores y CL les indicará el momento en que se debe reinicializar el conteo.

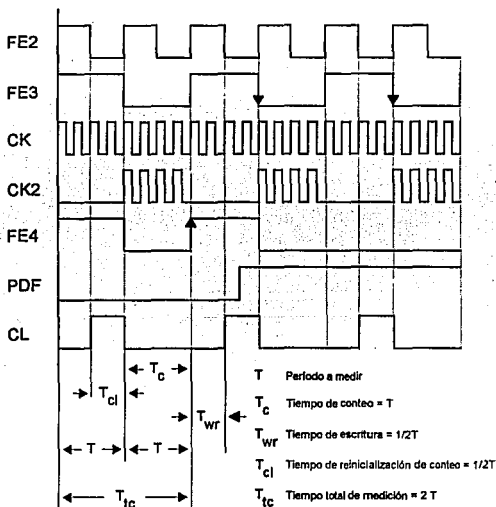


Figura 10.13 Diagrama de tiempos para el conteo de frecuencia.

La operación del circuito es como sigue: durante la parte baja de FE4 (esto es, durante un ciclo completo de FE2), el bloque de contadores realiza un conteo desde 0 hasta "N". En el momento en que la señal FE4 alcanza un valor alto, la señal de reloj que alimenta a los contadores se detiene y un bloque de 16 latches tipo D (dos circuitos 74HC374) fijan el valor "N" hasta el cual contaron los circuitos 74HC193. Una vez que los latches tienen almacenado el valor "N", los contadores son reinicializados. Este proceso se repite hasta que se recibe la bandera PDF en estado activo. Esto hace que la señal FE4 que dispara a los latches sea igual a cero, impidiendo que los latches cambien de valor durante la lectura de la frecuencia. El valor "N" que alcanzaron los contadores es el que se transmite a la computadora, donde se calcula la frecuencia de la señal de entrada considerando que se conoce:

- 1) El valor al cual fue dividida la frecuencia de entrada FE1 en el prescaler.
- 2) El número "N" de 16 bits hasta el cual contaron los circuitos 74HC193 durante la parte baja del ciclo de FE4.
- 3) La frecuencia a la cual se encontraban operando los contadores.

La tabla 10.7 muestra los valores que se deben conocer para el cálculo de la frecuencia en Hertz.

Frecuencia a medir	ESCFA	División de la frecuencia	Reloj en los contadores
0.02 Hz 0.1 Hz	0	+ 1	610.3516 Hz
0.1 Hz 1 Hz	1	+ 1	4882.813 Hz
1 Hz 10 Hz	2	+ 1	39.0625 KHz
10 Hz 100 Hz	3	+ 1	625 KHz
100 Hz 1000 Hz	4	+ 1	5 MHz
1000 Hz 10 KHz	5	+ 1	20 MHz
10 KHz 100 KHz	6	+ 32	20 MHz
100 KHz 1 MHz	7	+ 128	20 MHz

Tabla 10.7 Datos para el cálculo de la frecuencia de la señal medida.

10.12 Adquisición y almacenamiento

Las figuras 10.14 y 10.15 muestran al circuito del osciloscopio dedicado a la conversión A/D de la señal analógica y al almacenamiento de los datos muestreados.

La figura 10.14 muestra un circuito de adquisición que tiene la capacidad de tomar hasta 1.1 millones de muestras por segundo y que almacena hasta 4095 datos en una memoria RAM. Está basado en el circuito ADC0820, un convertidor A/D de 8 bits de semi-ráfaga que tiene un tiempo de conversión de 1.5 μ s. Para mejorar el tiempo de respuesta de éste circuito, se conectan los convertidores A/D usando la técnica de "ping-pong". El nombre se deriva del hecho de que los convertidores se activan alternadamente, de tal manera que mientras el primer convertidor procesa la muestra "n", el segundo convertidor se encuentra listo para procesar la muestra "n+1". Una vez que el segundo convertidor se encuentra procesando la muestra "n+1", el primer convertidor ha terminado con la muestra "n" y se encuentra listo

para la muestra "n+2". El proceso se repite hasta que se tienen todos los 4095 datos listos en la memoria. La figura 10.15 muestra la lógica de control requerida usando esta técnica.

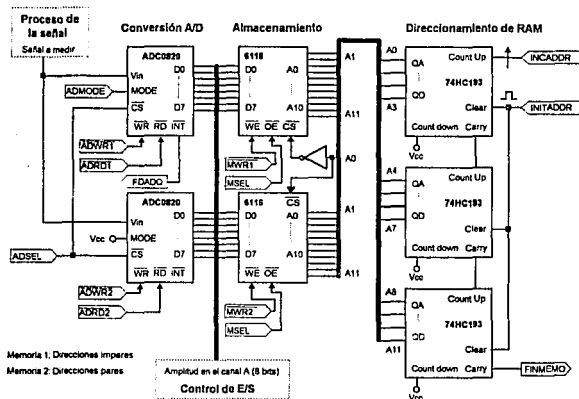


Figura 10.14 Circuito para la adquisición y almacenamiento de datos

La figura 10.15 muestra la lógica de control para operar al bloque de adquisición y almacenamiento. Los circuitos principales son el 74HC164, un registro de corrimiento de 8 bits de entrada serie salida paralelo, y el 74HC266, un bloque de 4 compuertas XNOR. Las compuertas se encargan de que la entrada de reloj controle las señales ADWR1, ADWR2, ADRD1, ADRD2, MWR1 y MWR2 de tal forma que la frecuencia de muestreo sea 1/3 de la frecuencia de reloj en el registro de corrimiento, alternando el uso de los convertidores y las memorias.

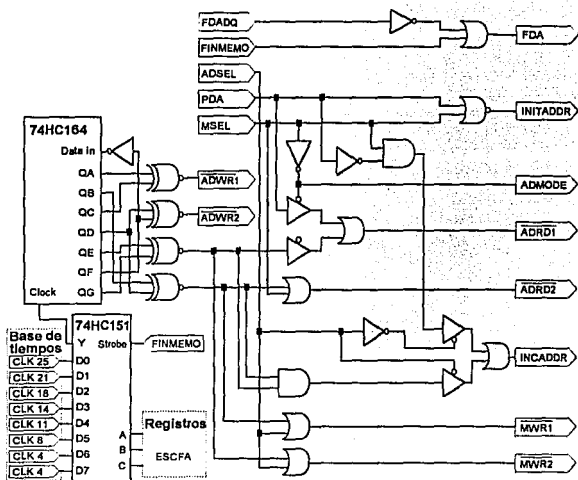


Figura 10.15 Lógica de control para el bloque de adquisición y almacenamiento

10.13 Secuencia de operaciones

El análisis de la estructura general del proyecto da una idea de la secuencia de pasos en la operación del módulo del osciloscopio digital. Los pasos generales son:

- Selección del módulo.** La computadora envía el código para seleccionar al módulo de osciloscopio digital.
- Inicialización.** El módulo de osciloscopio digital señala a la computadora que se encuentra listo para operar.

- C) **Programación.** El osciloscopio recibe datos sobre la manera de operación.
- D) **Adquisición de datos.** El osciloscopio comienza la adquisición de datos y envía los datos adquiridos a memoria RAM.
- E) **Transmisión de los datos.** El osciloscopio envía los datos en RAM a la computadora.

Una vez que se ha realizado esta secuencia de eventos, el proceso se repite completamente si se desea realizar una nueva adquisición. A continuación se da una descripción más detallada de la secuencia de operaciones para el caso de una adquisición usando solamente el canal A.

10.13.1 Selección del módulo

- 1) La PC programa al PPI de la tarjeta de interfase de datos con la IBM PC como P_A = salida, P_B = entrada, P_{CH} = salida y P_{CL} = salida.
- 2) La PC envía al puerto P_{CH} el código 0H para seleccionar al osciloscopio digital y al puerto P_{CL} el código 6H para seleccionar al registro de banderas 2, en donde se encuentra el registro RECDISO. Después de esto, se espera la respuesta del osciloscopio en el puerto P_B .

10.13.2 Inicialización

- 4) Al encender el osciloscopio, éste envía al puerto P_B el código AH para indicarle a la PC que el módulo está listo para operar.

10.13.3 Programación

- 5) Una vez que la PC ha leído en el puerto P_B que el código AH de reconocimiento está presente, envía al puerto P_{CL} el código 8H para activar la escritura en el registro ESCFA del osciloscopio. Después de esto, envía al puerto P_A el byte de programación de escala de frecuencia en el canal A.
- 6) La PC envía al puerto P_{CL} el código 9H para activar la escritura en el registro ESCFB del osciloscopio. Después de esto, envía al puerto P_A el byte de programación de escala de frecuencia en el canal B.
- 7) La PC envía al puerto P_{CL} el código 7H para activar la escritura en el registro de banderas 1 del osciloscopio. Después de esto, envía al puerto P_A el registro de banderas 1 con las banderas HCA, HCB, ACOA, ACOB, PDF, ADSEL, PDA y MSEL. Estas tres últimas deberán tener un valor de 0 para inicializar el direccionamiento de la memoria RAM en el osciloscopio (ver tabla 10.8).
- 8) La PC envía al puerto P_{CL} el código 6H para seleccionar la lectura del registro de banderas 2, en donde se encuentra la bandera FDA, la cual señalará el fin de adquisición. Una vez hecho esto, La PC espera hasta que esta bandera se active.

10.13.4 Adquisición de datos

- 9) Cuando las banderas ADSEL=0, MSEL=0 y PDA=1 (ver tabla 10.8), el osciloscopio comienza a adquirir datos. Una vez que la adquisición termina y los datos se han

almacenado en memoria, se envía a la PC la bandera FDA en estado activo para indicarle que los datos están listos.

Significado	ADSEL	PDA	MSEL
Inicializar el direccionamiento de RAM del osciloscopio	0	0	0
Iniciar adquisición sin uso de RAM	0	0	1
Iniciar adquisición usando RAM	0	1	0
Leer el dato adquirido sin uso de RAM	0	1	1
Inicializar el direccionamiento de RAM	1	0	0
Incrementar en 1 la dirección actual de RAM	1	0	1
Leer la dirección actual de RAM	1	1	0
Espera (Stand by)	1	1	1

Tabla 10.8 Combinación de banderas para el inicio de la adquisición y la recuperación de datos.

10.13.5 Transmisión de los datos

- 10) La tabla 10.8 muestra la combinación en las banderas usadas en la recuperación de los datos de amplitud de la señal. El primer paso será inicializar el direccionamiento de la memoria RAM del osciloscopio enviando las banderas ADSEL=1, PDA=0 y MSEL=0. Para esto, la PC solicita la escritura en el registro de banderas 1 enviando al puerto P_{CL} el código 7H. Después de esto, envía por el puerto P_A el registro de banderas 1.

- 11) Para leer el dato en la dirección actual de RAM del osciloscopio, se asignan valores en el registro de banderas 1 de tal forma que ADSEL=1, PDA=1, MSEL=0 y se envía por el puerto P_A. Después de esto, la PC envía al puerto P_{CL} el código 2H para solicitar la lectura de amplitud en el canal A del osciloscopio. Finalmente, se recupera por el puerto P_B el dato de amplitud solicitado.
- 12) Para incrementar la dirección de RAM del osciloscopio, la PC envía al puerto P_{CL} el código 7H para activar la escritura en el registro de banderas 1 y después envía por el puerto P_A el byte de banderas 1 con las banderas ADSEL=1, PDA=0 y MSEL=1.
- 13) Los dos pasos anteriores se repiten hasta que se haya leído toda la memoria RAM del osciloscopio.
- 14) Para recuperar el dato de frecuencia en el canal A del osciloscopio, La PC envía al puerto P_{CL} el código 7H para poder escribir al registro de banderas 1, en donde se indicará que la PC desea hacer una lectura de frecuencia (bandera PDF = 1). Después de esto, se envía al puerto P_{CL} el código 6H para solicitar lectura del registro de banderas 2, en donde se recibirá la bandera FDCF en estado activo cuando el dato de frecuencia se encuentre disponible.
- 15) Cuando la PC recibe por el puerto P_B a la bandera FDCF en estado activo, programa al PPI de la tarjeta de interfase de datos como P_A = entrada, P_B = entrada, P_{CH} = salida y P_{CL} = salida.
- 16) La PC envía al puerto P_{CL} el código 0H para solicitar la lectura de frecuencia en el canal A. Después de esto,

recupera por el puerto P_A el byte menos significativo y por el puerto P_B el byte más significativo.

10.14 Sumario

Tomando como base las especificaciones de diseño establecidas con anterioridad, en este capítulo se diseñó el módulo de osciloscopio de almacenamiento digital que se conectará a la PC como el primer módulo de instrumentación para el sistema INDICO. Como lo muestra el total de los diagramas de los circuitos presentados, el osciloscopio completo es un diseño complejo. Sin embargo, tomando como base un diagrama a bloques, el circuito se simplificó de una manera notable. Partiendo de este diagrama, en este capítulo se diseñaron circuitos para el acoplo de CA/CD, el procesamiento de la señal, la adquisición de datos y su almacenamiento, el generador de disparo, la base de tiempos, el frecuencímetro, los indicadores luminosos, el control de entrada, el buffer de registros y el selector de módulos de instrumentación.

Con este capítulo se finaliza el desarrollo del hardware de este proyecto. En el capítulo siguiente se comenzará con el desarrollo del software para controlar al módulo de osciloscopio digital diseñado. Al igual que en el caso del desarrollo de hardware, se partirá de unas especificaciones de diseño y se dividirá al sistema por bloques para facilitar la creación del programa.



Parte 4

Diseño del software

Objetivo:

Diseñar y codificar el programa de control y procesamiento de los datos recibidos en la computadora desde el sistema de adquisición de datos

Capítulos

- Especificaciones de diseño del software
- Arquitectura del programa
- Librerías de funciones

Capítulo

11

Especificaciones de diseño del software

11.1 Introducción

Una vez que se ha descrito el desarrollo de las etapas de hardware para el módulo de osciloscopio digital, los siguientes tres capítulos describirán la etapa del software requerido para poderlo usar. En este capítulo se darán las especificaciones a las cuales el programa deberá apegarse, tanto en su aspecto visual como en su aspecto operativo y de interfase usuario-computadora. Se especificarán también las herramientas a utilizar para el desarrollo del mismo, así como una justificación del uso de éstas. Posteriormente, en los siguientes dos capítulos se hará una descripción de cómo se desarrolló el software que cumple con dichas especificaciones.

11.2 Plataforma de software

El primer aspecto a considerar en el desarrollo de un programa es la plataforma en donde va a ejecutarse. Existen varias posibilidades para una PC: DOS, Windows, OS/2 y UNIX, entre otros. Cada uno de ellos tiene características que lo convierten en el apropiado para cierto tipo de aplicaciones, pero sin duda alguna, el entorno operativo más popular es DOS. Alrededor de 100,000,000 PC's en todo el mundo lo tienen instalado. DOS ha superado los retos de CP/M, UNIX y OS/2 debido a su simplicidad, su velocidad de ejecución, sus mínimos requerimientos de

hardware, y a la gran variedad de aplicaciones desarrolladas para este entorno. De hecho, es normal que cuando un usuario adquiere una computadora, recibe una copia de DOS como parte del equipo.

Sin embargo, DOS está muy lejos de ser el sistema operativo más avanzado técnicamente. Su tecnología de 16 bits con operación del microprocesador en modo real (ver el apéndice D para una discusión del modo real y el modo protegido en microprocesadores 80x86) obliga al usuario a trabajar en una línea de comandos en modo texto, su manejo de memoria se limita a 640 KB y no permite la multitarea. Computadoras con un microprocesador tan poderoso como el 486DX2 o el Pentium y plenitud de memoria instalada ejecutarán programas en DOS con las mismas restricciones que si tuvieran un procesador 8088 y 640 KB en RAM, quedando desaprovechado todo el hardware disponible.

A pesar de lo anteriormente mencionado, en el desarrollo de este proyecto la elección es usar DOS. La incapacidad de tener multitarea no es desventaja, ya que esto implica que el programa tendrá el 100% de la atención del microprocesador. La memoria tampoco es limitante, ya que el programa requiere menos de 640 KB para ejecutar. En cuanto a la forma de comunicarse con el usuario, DOS permite diseñar sofisticadas interfases gráficas con soporte para el ratón (Windows es un ejemplo de esto), mucho más atractivas que las de modo texto, el modo de ejecución normal de DOS. Por otro lado, el enfoque de este proyecto no es hacia su ejecución en máquinas muy sofisticadas, sino en cualquier PC que se tenga disponible. Esto implica, evidentemente, tener requerimientos de hardware bajos, al igual que DOS.

11.3 Lenguaje de programación

La elección del lenguaje de programación en la actualidad conduce generalmente al lenguaje C. Este lenguaje es el de más amplio uso por los desarrolladores de software, debido a la gran variedad de plataformas de hardware que le dan soporte, y a la eficiencia con la que opera. Prácticamente cualquier cosa que se quiera hacer, es posible hacerla con C. Además, es un lenguaje con una estructuración muy simple. Debido a esto, la conversión de un programa desarrollado en C hacia otro lenguaje no presenta mayores dificultades. Una ventaja más para el lenguaje C es que es un lenguaje que le deja al programador la libertad de hacer lo que quiera, puesto que asume que él sabe lo que hace. C no tiene las validaciones que lenguajes como Pascal o BASIC hacen ni restringe las operaciones que el programador desea realizar.

Una de las características de C es su capacidad de permitir la modularidad en la escritura de programas. Durante mucho tiempo la norma de los programadores era programar de manera modular, y C lo permite. Sin embargo, una de las tendencias en programación seguidas actualmente va más allá de la simple programación modular: se trata de la programación orientada a objetos (OOP : Object Oriented Programming). Esta no es una nueva técnica de programar, ya que data de hace más de 20 años. Con la programación orientada a objetos no se sigue una programación orientada a funciones, como lo hace el lenguaje C tradicional, sino a objetos que tienen atributos y que pueden ser cambiados dinámicamente en el transcurso del programa. Lenguajes como Eiffel o Small Talk son ejemplos de lenguajes puros de este tipo de programación. Existen otras implementaciones, tal como C++, que sin ser lenguajes puros de OOP, manejan su concepto.

La mejores opciones para el desarrollo de este proyecto quedan entre C y C++. Dada la simpleza de C sobre C++, se selecciona éste como herramienta de desarrollo. En C++ es común hallar términos como herencia, clases, polimorfismo y otros que denotan el gran poder de la OOP, pero que no son indispensables en éste caso.

11.4 Compilador

Un compilador de lenguaje C se encarga de traducir el código fuente en C a código máquina. Cada uno de los compiladores de C existentes se apega en menor o mayor grado a la estandarización del lenguaje C publicada en el estándar ANSI XJ11.34, en 1990. La diferencia al usar un compilador u otro se hace patente al hacer uso de las extensiones propias al estándar que cada compilador tiene. Hacer uso de tales extensiones puede restarle portabilidad al código escrito, pero puede darle mayor poder a los programas creados.

Entre las opciones a elegir se encuentran Borland C, Microsoft C, Aztec C, Lattice C, Watcom C y Zortech C, entre otros. De acuerdo a una revisión hecha en la revista "Computer language" (hoy fuera de circulación) de mayo de 1990, entre los mejores se encontraban Turbo C 2.0 de Borland y Microsoft C 6.0. Las versiones más recientes de estos compiladores son Borland C++ 4.0 y Visual C 1.5, los cuales permiten compilar código en C o C++, tanto para DOS como para Windows 3.1 o Windows NT y cuentan con extensas librerías a disposición del programador. Un análisis personal a compiladores de Borland y Microsoft favorece a Borland por su simpleza de uso, su potente ambiente integrado de desarrollo (IDE), una mejor librería de funciones y un depurador (debugger) muy eficiente y amigable. De este modo, el compilador

a usar es Borland C++ 3.1 (Esta versión ofrece más facilidades para la programación en DOS que la versión 4.0).

11.5 Nombre oficial del programa

El nombre seleccionado para el programa es INDICO, tomando las siglas de "Instrumentación Digital Asistida por Computadora".

11.6 Hardware requerido

INDICO se desarrollará para requerimientos mínimos de hardware. Este equipo mínimo incluye:

- 1) Una computadora personal IBM o compatible con las siguientes características:
 - Microprocesador 8088 o mejor.
 - 640KB de memoria.
 - 1 Puerto paralelo.
 - 1 unidad de disco flexible (disco duro deseable).
 - Tarjeta de video Hercules monocromático o compatible, o VGA de colores o monocromático.
 - 1 ranura de expansión de 8 bits ISA.

- 2) Un ratón (mouse) Microsoft con un driver versión 8.2 si se desea hacer operaciones de cursor.

- 3) Una impresora STAR NX-1000 o compatible si se desean imprimir las señales capturadas.

- 4) La tarjeta de interfase con la IBM PC y el módulo de adquisición de datos diseñados en la parte tres de éste documento.

Considerando que no todos los equipos son 100% compatibles, las computadoras en las que se asegura que este proyecto operará adecuadamente son:

- Printaform XT de 10 MHz con microprocesador 8088 de Intel y monitor Hercules monocromático.
- Gamma 88 XT de 12 Mhz con microprocesador V20 de NEC y monitor Hercules monocromático.
- Auva 390 de 40 Mhz con microprocesador 80386DX de AMD y monitor super VGA.

Como puede observarse por los requerimientos indicados, no existirá soporte para video CGA. Esto se debe a su baja resolución gráfica (320x200 pixeles con colores y 600x200 en modo monocromático), lo cual le impide tener una buena presentación en la zona de visualización de datos del osciloscopio o del menú de opciones.

11.7 Elementos del programa

11.7.1 Argumentos desde la línea de comandos

El programa contará con los siguientes argumentos opcionales, los cuales se indicarán desde la línea de comandos de DOS al ejecutar el programa:

- /S** Saltar la pantalla de la portada e ir directamente a la pantalla del osciloscopio.

/M	En una computadora VGA de colores, obliga a simular video en modo monocromático.
/SIMULA	Le indica al programa que opere en el modo de simulación de captura de señales. Este modo se usará para fines demostrativos del programa cuando no se tiene el módulo de hardware conectado.
/?	El programa muestra únicamente una pantalla donde se indican estas opciones y después termina.

11.7.2 El ratón

INDICO dará soporte al uso de un ratón Microsoft o compatible como dispositivo apuntador. Para que el programa pueda hacer uso de él, se debe tener instalado el driver del ratón, ya sea desde el archivo AUTOEXEC.BAT o CONFIG.SYS.

El programa reconocerá la entrada por ratón siempre que su cursor se encuentre presente. El cursor del ratón es presentado como una flecha cuando se espera una entrada al menú del programa o como una cruz cuando se realiza una operación de cursor o de zoom por ventana. Durante la espera de selección del menú, el botón derecho del ratón tendrá equivalencia con la tecla **ESC** y el botón izquierdo con la tecla **ENTER**. La posición del cursor en el menú al presionar el botón izquierdo determinará el elemento de menú que ha sido escogido.

11.7.3 Cajas de diálogo

Las cajas de diálogo consisten en ventanas a través de las cuales el usuario y un programa se intercambian información. El programa se encuentra en estado de espera hasta que el diálogo

haya terminado. En el presente programa, las cajas de diálogo constarán de una barra de título, una zona dedicada a presentar un mensaje, y una zona de opciones, tal como lo muestra la figura 11.1.

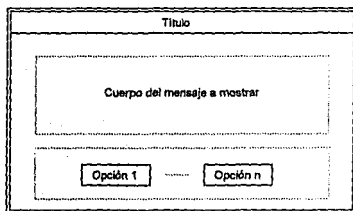


Figura 11.1 Aspecto de las cajas de diálogo

En este programa existirán los siguientes tipos de cajas de diálogo:

- 1) Aquellas que informen al usuario de un error irrecuperable (es decir, aquel error para el cual el sistema aborta la operación de la tarea ejecutada) de la terminación de un evento, o de su cancelación. La única opción que presentan estas ventanas es "Continuar".
- 2) Aquellas que informen al usuario de la realización de una operación que demora cierto tiempo. No se requiere respuesta, ya que estas ventanas desaparecen cuando la operación ha sido finalizada.
- 3) Las que requieren de un dato alfanumérico proporcionado por el usuario.

- 4) Las que informen de un error recuperable (es decir, aquel error para el cual el sistema reintentará la operación fallida) o soliciten una confirmación. Requieren del usuario que responda con una de las opciones presentadas: "Si" o "No", o bien "Cancelar" o "Reintentar".

11.7.4 Mensajes

Los mensajes de error, advertencia y confirmación que el programa podrá enviar al usuario mediante cajas de diálogo son:

"Se requiere ratón para esta operación."

"No hay señal en los canales."

"No hay archivos."

"La impresora no se encuentra lista."

"El osciloscopio no responde."

"Impresión cancelada."

"Impresión lista"

"¿Terminar y salir a DOS?"

"El puerto está adquiriendo datos. Espere un momento"

"El software está simulando una señal. Espere un momento."

"Recuperando señal de archivo. Espere un momento."

"Salvando señal en archivo. Espere un momento."

Adicionalmente, el programa mostrará dos tipos de mensajes con información:

Información general: mostrará el nombre del programa, el número de versión, autor, la memoria libre en la PC, la dirección de puerto usada por la tarjeta de comunicaciones, y la impresora instalada.

Información sobre la señal: mostrará la frecuencia en hertz, el voltaje pico en volts y el voltaje pico a pico en volts.

11.7.5 Iconos

Los iconos son imágenes en el programa que representan algún comando que puede ejecutarse. Para llamar a tal comando, podrá usarse la tecla que el icono indica, o seleccionarlo con el ratón, señalando el icono con el cursor y presionando el botón izquierdo.

11.7.6 Menús

Los menús contienen las opciones mediante los cuales el usuario podrá programar las características del osciloscopio o seleccionar las tareas a ejecutar haciendo uso del teclado o el ratón.

11.7.7 Archivos

Los archivos que el programa manejará para almacenar la información deberán ubicarse dentro del directorio de trabajo del programa y tendrán la extensión .OSC. El archivo se dividirá en tres partes: un encabezado general, destinado a todos los archivos que INDICO soporte en un futuro, un encabezado particular, que en este caso contendrá información del osciloscopio, y los datos.

Encabezado general (32 bytes):

Byte	Uso
0	Identificador (28 Hex).
1-2	Versión (10 = versión 1.0).

3	Tipo de archivo INDICO (1 = Osciloscopio).
4-5	Longitud del encabezado particular.
6-31	Reservados.

Encabezado particular al osciloscopio (105 bytes):

Byte	Uso
32	Número de señales.
33-40	Voltaje pico de la señal 1 (en volts).
41-48	Voltaje pico-pico de la señal 1 (en volts).
49-56	Frecuencia de la señal 1 (en hertz).
57-58	Número de datos de la señal 1.
59-60	Ancho en bits de los datos de la señal 1.
61-68	Posición del nivel de tierra de la señal 1.
69-76	Escala de voltaje al muestrear la señal 1.
77-84	Tiempo entre muestras de la señal 1.
85-92	Voltaje pico de la señal 2 (en volts).
93-100	Voltaje pico-pico de la señal 2 (en volts).
101-108	Frecuencia de la señal 2 (en hertz).
109-110	Número de datos de la señal 2.
111-112	Ancho en bits de los datos de la señal 2.
113-120	Posición del nivel de tierra de la señal 2.
121-128	Escala de voltaje al muestrear la señal 2.
129-136	Tiempo entre muestras de la señal 2.

Datos (n bytes):

Byte	Uso
137- x	Datos medidos.

11.7.8 Pantallas del programa

11.7.8.1 Portada

La portada del programa deberá contener la siguiente información:

Programa: "INDICO V 1.0 Instrumentación Digital asistida por computadora. Módulo de osciloscopio digital"

Autor: "Armando Romero Domínguez",

Objetivo: "Tesis para obtener el título de Ingeniero Mecánico Electricista"

Institución: "Facultad de Estudios Superiores Cuautitlán. Universidad Nacional Autónoma de México"

Fecha. Solamente el año (4 dígitos).

Esta portada deberá permanecer en la pantalla por 15 segundos o hasta que el usuario presione cualquier tecla, lo que pase primero.

11.7.8.2 Pantalla principal

La pantalla de operación del programa aparecerá después de la portada, y deberá contener las áreas que indica el esquema de la figura 11.2:

- 1.- **Título.** Contendrá la leyenda "Osciloscopio Digital".
- 2.- **Menú.** En esta zona se presentarán los menús del programa.
- 3.- **información de canales y de escalas.** La figura 11.3 muestra el detalle de esta zona. Constará de 6 recuadros en los cuales se indicarán los canales activos del osciloscopio, así como el acoplo de corriente directa o

alterna para cada canal. Esta zona será interactiva, de modo que el usuario podrá activar o desactivar cada cuadro haciendo uso del ratón. Igualmente, en esta zona se mostrarán las escalas de voltaje y tiempo seleccionadas.

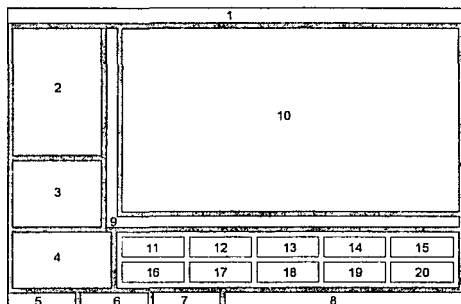


Figura 11.2 Esquema de la pantalla de operación del programa

Escala	
Volt/Div	= _____
Tiempo/Div	= _____
Canal	Acoplo
<input type="checkbox"/> A	<input type="checkbox"/> CA <input type="checkbox"/> CD
<input type="checkbox"/> B	<input type="checkbox"/> CA <input type="checkbox"/> CD

Figura 11.3 Zona de información de canales y escalas.

- 4.- **Resultados de las operaciones de cursor.** En ella se mostrarán los resultados de las tres operaciones de cursor: delta V, delta T y frecuencia.
- 5.- **Opción del menú en el nivel 1.** Una vez que el usuario seleccione una opción del menú principal, esa opción se escribirá en esta zona y un nuevo menú reemplazará al menú principal en la zona de menús.
- 6.- **Opción del menú en el nivel 2.** Cuando una opción de un menú de nivel 2 genere a su vez otro menú, en esta zona se escribirá la opción del menú de nivel dos seleccionada.
- 7.- **Opción del menú en el nivel 3.** Tiene una función análoga a las dos zonas anteriormente descritas.
- 8.- **Ayuda de operaciones de cursor.** En esta zona se mostrará una ayuda sobre el uso de los botones del ratón durante las operaciones de cursor.
- 9.- **Divisiones de escalas.** Mostrará una regla con divisiones que representará las escalas en el eje horizontal y vertical. Para ambos ejes una división se representará por 25 píxeles. El usuario podrá darse una idea de la magnitud de la señal actualmente mostrada relacionando estas divisiones con la señal y los valores de escala de volts/división y tiempo/división.
- 10.- **Monitor del osciloscopio.** Esta zona estará reservada para tres usos:
 - Presentación de las formas de onda capturadas por el módulo de hardware.
 - Operaciones de cursor y zoom de la señal.

- Presentación de cajas de diálogo.

11 al 20.- Iconos. Constará de 10 cuadros con un icono, el nombre de una tecla y el nombre del comando que se podrá ejecutar. Esta zona también deberá ser sensible al ratón.

11.8 Operaciones del programa

11.8.1 Menú de funciones de archivo

Se contemplarán las siguientes opciones:

Cargar	Solicitará mediante una caja de diálogo el nombre de un archivo que contenga una señal almacenada. Durante el proceso de carga, se le deberá informar al usuario que ésta operación ya se está realizando.
Salvar	Solicitará mediante una caja de diálogo el nombre de un archivo donde se guardará la señal presente. Si no existe señal presente, se deberá generar un mensaje de error.
Imprimir	Imprimirá la señal presente en el monitor del osciloscopio, excepto si la impresora no está lista o si no hay señal presente, en cuyo caso se presentará el respectivo mensaje de error.
Salir	Permitirá concluir la ejecución del programa, previa confirmación de ésta operación.
Información	Presentará información del nombre del programa, versión y autor, así como de la memoria disponible.

11.8.2 Menú de captura y opciones de captura

Las opciones de este menú serán:

Capturar	Realizará la captura de la señal. Si el módulo de hardware no se encuentra presente, se generará un mensaje de error a menos que el usuario haya usado la opción /SIMULA desde la línea de comandos, en cuyo caso se simulará la captura de dos señales, las cuales serán generadas por series de Fourier.
Muestreo	En esta opción se determinará la frecuencia a usar para el muestreo. Para mayor facilidad para el usuario, solo tendrá que indicar el rango en hertz de la señal a medir.

11.8.3 Menú de funciones en la pantalla

Vistas	Indicará si se debe presentar en la pantalla o no la retícula, los ejes de las escalas, el canal A y el canal B.
Limpiar	Limpiará el contenido de la pantalla.
Trazo	Determinará la forma en que se trazará la señal en el osciloscopio: solamente por puntos, o uniendo esos puntos con líneas.
Zoom	Permitirá efectuar las operaciones de zoom por ventana o zoom predefinido. En la opción de zoom por ventana el usuario marcará una ventana apuntando con el cursor del ratón a la primera esquina de una ventana, presionando el botón derecho y apuntando a la esquina opuesta de la ventana sin soltar el botón. Al

soltar el botón, la posición del cursor determinará la segunda esquina de la ventana. En el zoom predefinido se hace un acercamiento de la señal en escalas de x2, x5, x10, x20 y x50.

11.8.4 Menú de control de la señal

Escalas	Presentará cajas de diálogo para solicitar las escalas en los ejes de voltaje y tiempo.
Memoria	Indicará la memoria en KB reservada para cada canal.
Acoplo	Permitirá seleccionar el tipo de acoplo (corriente alterna o directa) para cada canal. Esta opción será válida solamente si se usa antes de capturar la señal.
Información	Indicará para cada canal las estadísticas de la señal: frecuencia, voltaje pico y voltaje pico a pico.

11.8.5 Menú de operaciones en la señal

Para esta versión, sólo se realizarán operaciones de cursor, las cuales se describen a continuación:

Delta V	Mostrará dos líneas horizontales. Cuando se presione el botón izquierdo del ratón, la línea más cercana se moverá a donde se encuentre el cursor. En la zona de resultados de las operaciones de cursor se mostrará la diferencia en volts de separación entre las dos
----------------	--

	líneas horizontales. El botón derecho del ratón dará por terminada la operación.
Delta T	Su comportamiento será idéntico al de la operación Delta V , excepto que en este caso las líneas serán verticales y los resultados se darán en segundos.
Frecuencia	Su comportamiento es idéntico al de la operación Delta T , excepto que los resultados serán en Hertz.

11.8.6 Menú de control de la posición de la señal

Este menú dirigirá la posición de la señal en cada canal en el eje vertical. Al seleccionar el canal a mover, aparecerá el nivel de tierra de cada canal. El usuario podrá mover únicamente el nivel de tierra del canal seleccionado usando el ratón. Al presionar el botón izquierdo de éste, el nivel de tierra se moverá a la posición del cursor. Finalmente, al presionar el botón derecho del ratón, la pantalla se repintará para mostrar las nuevas posiciones.

11.8.7 Menú de configuración

Permitirá configurar el puerto al cual se tiene conectado el módulo de hardware. Adicionalmente, se tiene contemplado para futuras versiones configurar la impresora. Para la presente versión, en la opción de configurar impresora se presentará un mensaje indicando que la impresora disponible es la Star NX-100.

11.8.8 Funciones en los iconos

- F1** **Ayuda.** Mostrará una caja de diálogo con ayuda del programa.
- F2** **Salvar.** Operación idéntica a la descrita en el menú.
- F3** **Cargar.** Operación idéntica a la descrita en el menú.
- F4** **Posición Y.** Moverá la señal en el canal activo hacia arriba y después hacia abajo en pasos de 20 píxeles cada vez que este icono sea invocado.
- F5** **CA-CD.** Cambiará el tipo de señal a capturar en los canales A y B. Esta opción será válida solamente si se usa antes de capturar la señal.
- F6** **Canales.** Conmutará entre la visualización del canal A o el B (sólo uno a la vez).
- F7** **Zoom.** Igual que el zoom por ventana descrito en las operaciones de menú.
- F8** **Imprimir.** Operación idéntica a la descrita en el menú.
- F9** **Escalas.** Igual a la operación en el menú, solo que este icono invocará tanto a la caja de diálogo de escala en V y escala en T.
- F10** **Salir.** Operación idéntica a la descrita en el menú.

11.9 Sumario

En este capítulo se especificaron las características que debe incluir el software de éste proyecto. Esto facilitará las cosas en el desarrollo del programa, ya que bastará con codificar las rutinas que se apegan a esto. En el capítulo siguiente se desarrollará un diagrama de flujo que describa la operación del programa.

Capítulo 12

Arquitectura del programa

12.1 Introducción

En el capítulo anterior se especificaron las características que el software de este proyecto debe tener, tanto en diseño como en aspecto. En este capítulo se desarrolla el software que se apega a tales especificaciones. Para esto, se comienza con el desarrollo de los diagramas de flujo que describen la operación completa del programa. Cada uno de estos diagramas de flujo se diseñó de una manera general, de tal manera que cada elemento pudiera tratarse como un módulo dentro del programa. Después de los diagramas de flujo, este capítulo continúa con la codificación de tales módulos presentando algunas rutinas en lenguaje C. Debido al tamaño del programa completo (más de 500 KB, lo cual representa alrededor de 15,000 líneas o 600 páginas de 25 renglones), no se presentan en este capítulo los listados de todo el programa.

12.2 Diagrama de bloques del programa

Las opciones de las cuales consta el programa son seleccionadas a través de menús. Este sistema de menús tendrá el control completo del programa, exceptuando en las fases de inicialización y de terminación. Esto nos permite desarrollar un diagrama del programa como el mostrado en la figura 12.1.

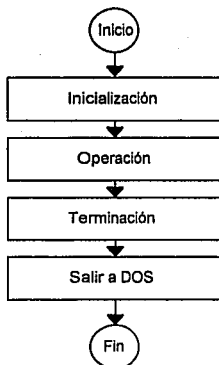


Figura 12.1 Rutina principal

Como puede apreciarse, el diagrama consta únicamente de 3 partes: el bloque de inicialización, el bloque de operación, y el bloque de terminación, para finalmente salir a DOS. El bloque de inicialización se encarga de inicializar las variables y estructuras que el programa requiere. El bloque de operación es el encargado de capturar y controlar los eventos del menú generados a partir de las acciones del usuario. Finalmente, el bloque de terminación libera los recursos de memoria solicitados por el programa durante la inicialización.

12.2.1 Bloque de inicialización

Desarrollando más a fondo el diagrama de la figura 12.1, la figura 12.2 muestra los componentes para el bloque de inicialización. La siguiente es una descripción de cada uno de éstos:

Reconocimiento del sistema. Determina el tipo de computadora que se encuentra ejecutando el programa, la unidad de disco desde la cual el programa se encuentra ejecutando, el número de unidades de disco presentes, la versión del sistema operativo, monitor monocromático o de colores, memoria disponible y otros parámetros de éste tipo. Igualmente, en este bloque se determina si los archivos de soporte para este programa se encuentran presentes o no.

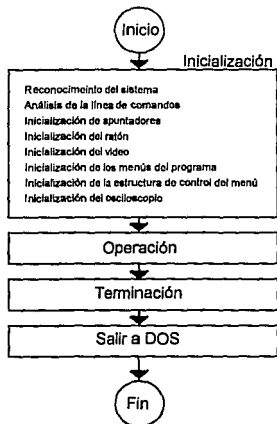


Figura 12.2 Bloque de inicialización

Análisis de la línea de comandos. Analiza los parámetros con los cuales se corrió el programa. Para ésta versión del programa, existen cuatro parámetros disponibles: /S, el cual permite eliminar la portada de presentación y saltar directamente a la pantalla de operación del osciloscopio. El parámetro /M obliga al programa a presentar el video en modo monocromático cuando se está usando un monitor VGA a colores. Esto puede ser necesario en algunos tipos de computadora cuya tarjeta de video no es totalmente compatible con el manejador de gráficos que usa el programa. El parámetro /SIMULA le indica al programa que simule la captura de señales para propósitos de demostración del programa. La opción /? solicita ayuda sobre los parámetros con los que cuenta el programa.

Inicialización de apuntadores. Durante el programa se hace la ubicación dinámica de memoria para varios buffers de datos, estructuras y variables auxiliares. Este bloque se encarga de ubicar la memoria necesaria para cada uno de estos elementos. En caso de no poder ubicar la memoria suficiente, el programa es abortado presentando un indicativo del archivo y número de línea donde se generó el error.

Inicialización del ratón. Este bloque se encarga de determinar si se encuentra disponible el ratón o no, e inicializarlo en caso de encontrarlo. Este programa da soporte a un ratón Microsoft o compatible. Para que pueda hacerse uso de él, es necesario previamente cargar el manejador MOUSE.DRV o MOUSE.SYS.

Inicialización del video. En este bloque se inicializa el video en modo gráfico y se presentan las pantallas de presentación del programa y de operación del mismo. Las tarjetas de video soportadas son la VGA monocromática y de colores y la Hercules monocromática. Cualquier otra tarjeta presente provocará que el programa aborte presentando un mensaje de error. Naturalmente,

una tarjeta súper VGA también es aceptada por el programa, ya que puede comportarse como una VGA estándar.

Inicialización de los menús del programa. Se encarga de copiar las cadenas de texto que cada menú presentará al usuario a sus respectivos apuntadores.

Inicialización de la estructura de control de menús. Copia a la estructura de control de menús los parámetros que éste requiere: colores, posiciones, el identificador del menú principal, y el primer comando de menú a ejecutar: `CMD_REPINTAR`.

Inicialización del osciloscopio. Inicializa a la estructura de control del osciloscopio con los parámetros predefinidos: canal activo, tipo de señal en cada canal, escalas, etc. Igualmente, inicializa al hardware de captura de datos.

12.2.2 Bloque de operación

El bloque de operación es el corazón del programa. Si ejecutáramos el programa hasta este punto, el usuario vería una pantalla de operación en modo gráfico con el monitor del osciloscopio vacío y la zona de menús en blanco. Internamente, ya se encuentra activado el ratón, aunque no es utilizable. Igualmente, los menús que el usuario puede llamar ya se encuentran en memoria, pero sin conexión alguna con el programa. Las tareas básicas que este bloque tiene que realizar para que el programa se encuentre operativo son:

- ❑ Reconocer la entrada de comandos del usuario por teclado o ratón.

- ❑ Interpretar dichas entradas como eventos de menú, eventos de iconos o eventos de recuadro de canal.
- ❑ Ejecutar el evento generado.

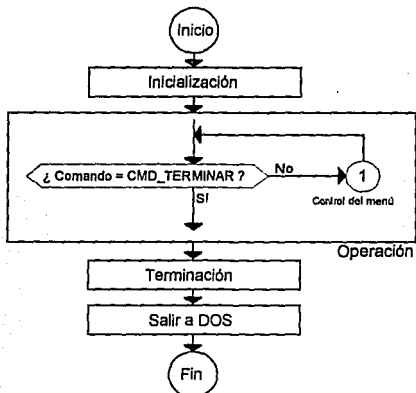


Figura 12.3 Bloque de operación

El diagrama de bloques en la figura 12.3 muestra la forma más simple del bloque de operación. En ella se pregunta por el comando de menú **CMD_TERMINAR** para poder terminar el programa. Mientras este comando de menú no se genere, el programa se encontrará ejecutando cíclicamente al bloque de control del menú. La siguientes secciones describen a éste bloque por partes.

12.2.2.1 Control del menú

La figura 12.4 muestra al bloque de control del menú, el cual recibe cuatro de los cinco comandos de menú que el programa reconoce:

CMD_SALIR Solicita la terminación del programa. Si esta es concedida, se activa el comando **CMD_TERMINAR**.

CMD_FUNCION Solicita que se ejecute alguna de las funciones del programa: captura de una señal, salvar la señal en disco, zoom, operaciones de cursor, etc.

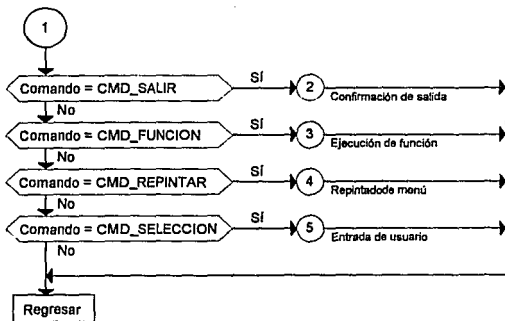


Figura 12.4 Control del menú

CMD_REPINTAR Solicita que la zona de la pantalla dedicada a la presentación del menú sea refrescada con

el menú actual. El menú actual puede cambiar cuando el usuario selecciona una opción que manda llamar a otro menú, o cuando se regresa de un menú al menú previo. Este comando es el primero en ser ejecutado por el programa.

CMD_SELECCION Solicita al usuario que seleccione una opción mediante teclado o con el ratón, para su análisis por parte del programa.

12.2.2.2 Confirmación de salida

La figura 12.5 muestra la operación efectuada cuando el comando **CMD_SALIR** se ha generado. Como puede observarse, después de su ejecución el comando debe ser cambiado, ya que de lo contrario el programa seguiría llamando a éste bloque.

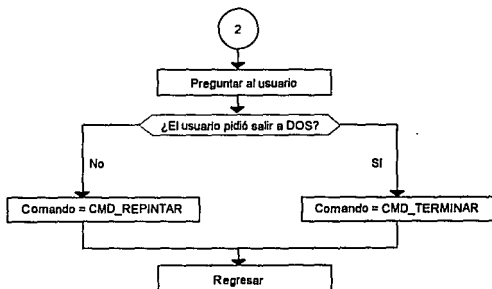


Figura 12.5 Confirmación de salida

12.2.2.3 Ejecución de función

La figura 12.6 muestra la parte más importante del programa, ya que es la que permite la ejecución de las opciones que el programa presenta. Cada uno de las funciones del osciloscopio, desde la captura de una señal hasta la presentación de la ayuda pasan por este bloque.

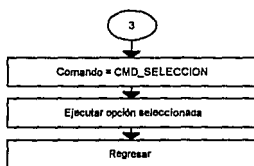


Figura 12.6 Ejecución de función

12.2.2.4 Repintado del menú

La figura 12.7 muestra la operación de repintado del menú.

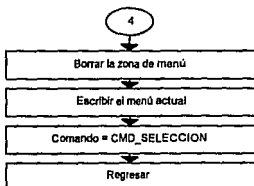


Figura 12.7 Repintado del menú

12.2.2.5 Entrada de usuario

La figura 12.8 muestra la captura de entradas del usuario por teclado o por ratón. Esta información es almacenada en una estructura, la cual es analizada para saber si el usuario solicitó un comando de navegación en el menú, o una acción directa de ejecución de función.

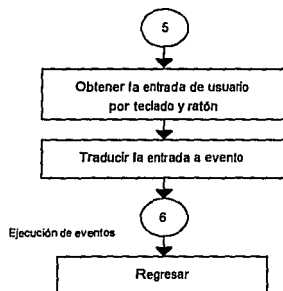


Figura 12.8 Entrada de usuario

12.2.2.6 Ejecución de eventos

La figura 12.9 muestra la operación a seguir para la ejecución de un evento. Como se mencionó anteriormente existen eventos de menú, eventos de iconos y eventos de recuadro de canal. Estos dos últimos, por tener una acción inmediata en la ejecución de una función, se clasifican como el evento de menú **ME_FUNCION**. A continuación se describen los eventos de menú generados.

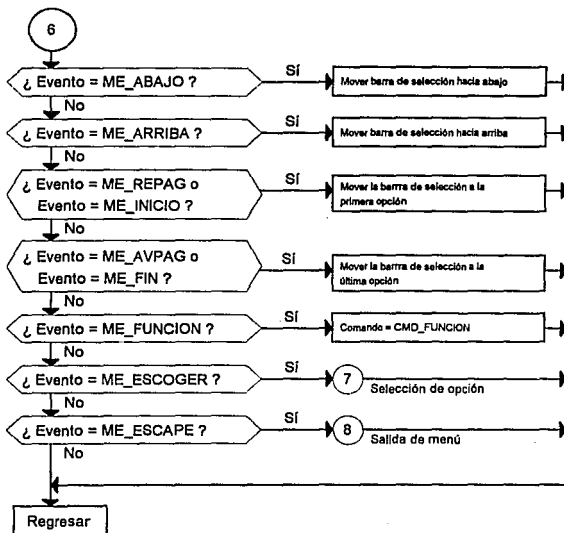


Figura 12.9 Ejecución de eventos

Cuando el usuario solicita que la barra de selección del menú se mueva, los siguientes eventos indican la dirección del movimiento:

ME_ABAJO	Hacia abajo.
ME_ARRIBA	Hacia arriba.
ME_REPAG	Al inicio.
ME_INICIO	Al inicio.
ME_AVPAJ	Al final.
ME_FIN	Al final.

Cuando el usuario solicita una ejecución o cancelación, los siguientes eventos indican la acción a realizar:

ME_FUNCION	El usuario solicita que se ejecute una función del osciloscopio.
ME_ESCOGER	El usuario selecciona la opción actual del menú.
ME_ESCAPE	El usuario pide salir del menú.

12.2.2.7 Escoger opción

La figura 12.10 muestra la operación del programa cuando el usuario ha escogido una opción del menú.

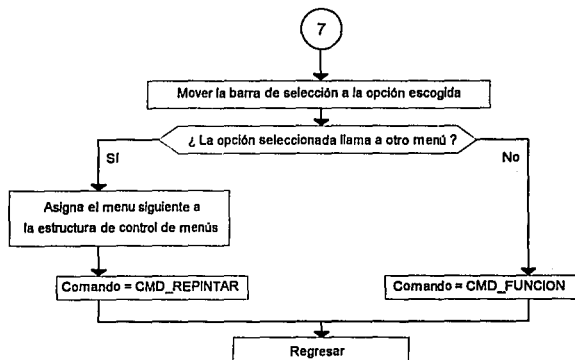


Figura 12.10 Selección de opción

12.2.2.8 Salir del menú

La figura 12.11 muestra el comportamiento del programa cuando el usuario solicita abandonar el menú actual.

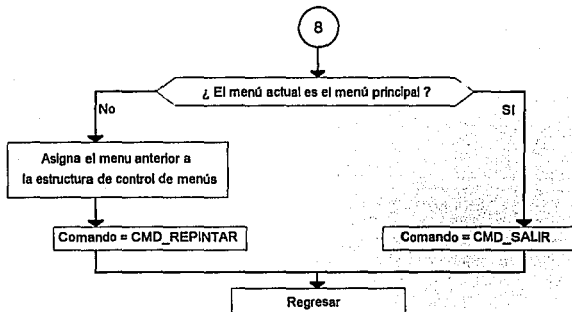


Figura 12.11 Salida del menú

12.2.3 Bloque de terminación

El bloque de terminación, mostrado en la figura 12.12, se encarga básicamente de tres cosas: liberar el ratón, liberar los recursos de memoria solicitados para los apuntadores, y regresar el video a modo texto. Una vez que ha hecho esto, se encuentra listo para terminar y salir a DOS.

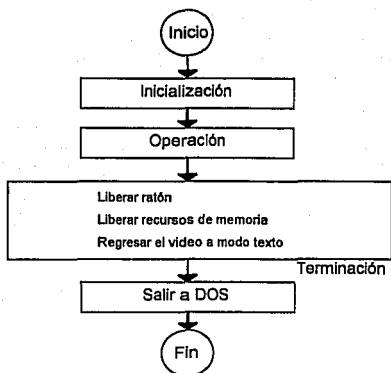


Figura 12.12 Bloque de terminación

12.3 Módulos del programa

Una vez que se ha analizado el diagrama de flujo del programa, se pueden crear las rutinas en lenguaje C necesarias. A partir del análisis de los bloques de inicialización, operación y terminación, se concluyó crear los siguientes módulos en el programa:

- Inicialización
- Control y operación
- Escritura en pantalla
- Operación del osciloscopio

Los siguientes incisos darán una descripción de los archivos del programa a partir de esta clasificación de los módulos.

12.3.1 Módulos de inicialización

Los módulos de inicialización que componen al programa son los siguientes:

INIT.C	Funciones de inicialización de apuntadores y estructuras globales.
INITDSO.C	Funciones de inicialización del osciloscopio.
INITMENU.C	Funciones de inicialización del sistema de menús.
INITMOUS.C	Funciones de reconocimiento e inicialización del ratón.
INITVID.C	Funciones de reconocimiento e inicialización del video, así como de la presentación de las pantallas de presentación y operación del programa.

12.3.2 Módulos de control y operación

Los módulos de operación y control que componen al programa son:

MAINFILE.C	Función main()
CTRLDLG.C	Funciones que controlan a las ventanas de diálogo del programa.
CTRLHW.C	Funciones que controlan al hardware externo de adquisición.
CTRLMENU.C	Funciones que controlan los eventos del menú.
CTRLMOUS.C	Funciones que controlan la operación del ratón.

CTRLPRN.C	Funciones de control de impresión de resultados.
CTRLVID.C	Funciones de control de video.
INPUT.C	Funciones de captura de información por teclado y ratón y su interpretación en eventos de menú.
EXEC.C	Funcion vEjecutor(), encargada de ejecutar todas las operaciones que el osciloscopio puede realizar.
EDITLINE.C	Editor de líneas de texto, usado para capturar nombres de archivos en donde el usuario salva o recupera información.
LISTADIR.C	Funciones de listado del directorio, usado para mostrar los archivos de señales.
OPENSVE.C	Funciones para salvar y recuperar señales desde disco.

12.3.3 Módulos de escritura en pantalla

Los módulos de escritura en pantalla son los siguientes:

DRAWDLG.C	Funciones de escritura y manipulación de ventanas de diálogo.
DRAWDSO.C	Funciones de escritura y refresco de la zona del monitor del osciloscopio.
DRAWMENU.C	Funciones de escritura y refresco de la zona de menús.

12.3.4 Módulos de operación del osciloscopio

Consta de los siguientes módulos:

CAPTURA.C	Función vCaptura(), encargada de realizar la adquisición de datos provenientes del equipo externo de hardware.
CURSOR.C	Funciones que realizan las operaciones de cursor del osciloscopio.

12.4 Muestra de funciones

Las siguientes son las funciones más representativas del programa:

- main()
- vCaptura()

En las siguientes dos secciones se darán los listados de estas dos funciones.

En la creación del programa se trató de dar la mayor legibilidad al código, para lo cual se siguieron las siguientes reglas:

- 1.- Las variables y funciones deben tener nombres significativos y deben comenzar con una abreviatura del tipo de dato. Por ejemplo, para la función vCaptura(), cuyo tipo de dato es *void*, la primera letra será "v". Una función o variable tipo *int* comenzará con "i".
- 2.- No usar más de 80 columnas por renglón.
- 3.- Documentar cada archivo indicando las funciones que contiene, así como la función global de las funciones contenidas en el mismo.

- 4.- Documentar cada función, indicando los argumentos que requiere y lo que regresa.
- 5.- Aislar las tareas específicas de éste programa con aquellas genéricas que puedan emplearse en cualquier otro programa. Esto permite la creación de librerías, las cuales pueden aplicarse en otros proyectos.

12.4.1 Función main()

El siguiente listado muestra el contenido completo del archivo MAINFILE.C, donde se encuentra la función `main()`. Como puede observarse, esta función se apega al esquema de la figura 12.1. La fase de inicialización se cubre desde la primera línea hasta el llamado a `vinitDSO()`. La fase de operación es llevada a cabo por `sfpControlDeMenu()` y las funciones restantes se encargan de la fase de terminación.

```

/*
Programa:      INDICO.EXE
Versión:      1.0
Fecha:        marzo 1994
Uso:          Módulo de Osciloscopio de Almacenamiento Digital para el
              sistema de Instrumentación Digital Asistido por Computadora.
Archivo:      Mainfile.C
Contenido:    MODULO PRINCIPAL DEL PROGRAMA
Proyecto:     INDICO.PRJ
Funciones:    1) main()
              2) iAnalizaLineaDeComandos()
              3) iValidaPrograma()
              4) vOpcionesDelPrograma()
Sist.de Comp: IBM PC o compatibles
Sist. Op.:    MS-DOS v 5.0
Lenguaje:     C
Compilador:   Borland C++ 3.1 Modelo de memoria Large
Propietario:  Armando Romero Domínguez
*/

```

```
#include <stdlib.h>
```

```
/* exit() */
```

```

#include <string.h>          /* strcmp()                */
#include <stdio.h>          /* printf()                */
#include <io.h>              /* access()                */
#include <bios.h>           /* bioskey()               */
#include <ltexto.h>         /* iClearVideo()           */
#include <misc.h>           /* Definiciones varias     */
#include <init.h>           /* Prototipos y estructuras */
#include <ctrl.h>           /* Prototipos de funciones de control */
#include <mainfile.h>      /* Definiciones y prototipos */

```

```

STRUCTMOUSE far          *sfpgRaton;
int              igParametros;
unsigned int     uigPuerto;

```

/*

FUNCION: *main()*
DESCRIPCION: *Inicializa los apuntadores globales, el ratón, el video el sistema de menús y controla al despachador de eventos de menú hasta que se recibe el mensaje de terminación del programa.*

LINEA DE COMANDOS:

```

/s      No presenta la pantalla de presentación del programa
/m      Para monitores VGA a colores, este argumento obligar a usar el video VGA en blanco y negro.
/simula Simula la captura de una señal si no hay enlace con el osciloscopio
/?      Ayuda en la línea de comandos.

```

LLAMA A:

```

iAnalizaLineaDeComandos().....MainFile.C
vInicializaApuntadores().....Init.C
sfInicializaRaton().....InitMous.C
iInItVideo().....InitVid.C
vTerminaVideo().....CtrlVid.C
vInItMenuPrograma().....InitMenu.C
sfInicializaMenuPrincipal().....InitMenu.C
vInItDSO().....InitDSO.C
sfpControlDeMenu().....CtrlMenu.C
vTerminaRaton().....CtrlMouse.C
vTerminaVideo().....CtrlVid.C
iValidaPrograma().....MainFile.C

```

REGRESA:

Nada

No. función :

- 1 -

*/

```

void main(int iNumArg , char **cppArgumentos )

```

{

```

    STRUCTMENUVERT far    *sfpMenu;
    int                  iTerminar;

```

```

    igParametros = iAnalizaLineaDeComandos(iNumArg , cppArgumentos);

```

```

vOpcionesDelPrograma(igParametros);
if (!iValidaPrograma(igParametros))
    exit(0);
vInicializaApuntadores();
sfpgRaton = sfpInicializaRaton();
iTerminar = !iInitVideo(sfpgRaton, igParametros);
if (iTerminar==0)
{
    vInitMenuPrograma();
    sfpMenu = sfpInicializaMenuPrincipal();
    vInitDSO();
    while ( sfpMenu->iTerminar == 0)
        sfpMenu = sfpControlDeMenu(sfpMenu, sfpgRaton) ;
}
vTerminaRaton(sfpgRaton);
vTerminaVideo(iTerminar);
}
}

```

/*																						
FUNCION:	<i>iAnalizaLineaDeComandos()</i>																					
PROTOTIPO EN:	<i>MainFile.H</i>																					
DESCRIPCION:	<p>Lee la línea de comandos en busca de las siguientes banderas:</p> <p><i>/S</i> No presentar la portada del programa</p> <p><i>/M</i> Usar el modo monocromático cuando se tenga una tarjeta VGA de colores.</p> <p><i>/SIMULA</i> Simular una señal por Fourier cuando se le indique al programa que se capture una señal.</p> <p><i>/?</i> Ayuda en la línea de comandos.</p> <p>Si encuentra cualquiera de las anteriores opciones, se pone a 1 el bit correspondiente del valor de regreso.</p>																					
ARGUMENTOS:	<p><i>iNumArgs</i> Número de argumentos</p> <p><i>cppCadena</i> Lista con los parámetros</p>																					
REGRESA:	<p>Un entero con la información encontrada. La información se almacena de la siguiente forma:</p> <table border="1"> <thead> <tr> <th>BIT</th> <th>VALORES</th> <th>SIGNIFICADO</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>Corre el programa en modo monocromático.</td> </tr> <tr> <td>0</td> <td>0</td> <td>El programa detecta el modo de video.</td> </tr> <tr> <td>1</td> <td>1</td> <td>No muestra la portada del programa.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Muestra la portada.</td> </tr> <tr> <td>2</td> <td>1</td> <td>Simula la captura de señales.</td> </tr> <tr> <td>2</td> <td>0</td> <td>Captura realmente una señal.</td> </tr> </tbody> </table>	BIT	VALORES	SIGNIFICADO	0	1	Corre el programa en modo monocromático.	0	0	El programa detecta el modo de video.	1	1	No muestra la portada del programa.	1	0	Muestra la portada.	2	1	Simula la captura de señales.	2	0	Captura realmente una señal.
BIT	VALORES	SIGNIFICADO																				
0	1	Corre el programa en modo monocromático.																				
0	0	El programa detecta el modo de video.																				
1	1	No muestra la portada del programa.																				
1	0	Muestra la portada.																				
2	1	Simula la captura de señales.																				
2	0	Captura realmente una señal.																				
LLAMADA POR:	<i>main()</i> <i>MainFile.C</i>																					
No. función :	<u> 2 </u>																					
*/																						

```

int iAnalizaLineaDeComandos(int iNumArgs , char **cpCadena)
{
    #define      cChar  cpCadena[iAux][iAux2]
    int         iAux;
    int         iAux2;
    int         iParametros = 0;

    for (iAux = 1 ; iAux < iNumArgs ; iAux++ )
    {
        for (iAux2=0 ; cChar != '\0' ; iAux2++)
            cChar = ( cChar >= 'a' && cChar <= 'z' )?
                cChar - ' ' : cChar;
        if ( strcmp( cpCadena[iAux], "7S" ) == 0 )
            iParametros |= SALTARPORTADA ;
        else if ( strcmp( cpCadena[iAux], "7M" ) == 0 )
            iParametros |= VIDEOVGAMONO ;
        else if ( strcmp( cpCadena[iAux], "7SIMULA" ) == 0 )
            iParametros |= SIMULACAPTURA ;
        else if ( strcmp( cpCadena[iAux], "7?" ) == 0 )
            iParametros |= OPCIONESDELPROGRAMA;
    }
    #undef cChar
    return iParametros;
}

```

FUNCION:	<i>iValidaPrograma()</i>
PROTOTIPO EN:	<i>MainFile.H</i>
DESCRIPCION:	<i>Valida la ejecución del programa, verificando que:</i> A) <i>Los archivos que se requieren se encuentren presentes.</i> B) <i>El enlace entre el osciloscopio y la computadora verifique OK.</i>
ARGUMENTOS:	<i>iParametros</i> <i>Información encontrada en la línea de comandos.</i>
REGRESA:	<i>0</i> <i>Se encontró algo mal.</i> <i>1</i> <i>Todo OK.</i>
LLAMA A:	<i>iClearVideo()</i> <i>LTEXT0.LIB</i> <i>vNoCursor()</i> <i>LTEXT0.LIB</i> <i>iComunicacionesConHw()</i> <i>CtrlHw.C</i>
LLAMADA POR:	<i>main()</i> <i>MainFile.C</i>
No. función:	<i>-3</i>

```

*/
int iValidaPrograma(int iParametros)
{
    int         iVerComunicaciones;

    iVerComunicaciones = I( (iParametros & SIMULACAPTURA) >> 2);
    ulgPuerto = 0x300;
    iClearVideo();
}

```

```
printf("\nIniciando programa...\n");
printf("Revisando archivos gráficos.... \n");
if (access("PHGCMONO.PCX", 0)!=0)
{
    printf("\PHGCMONO.PCX no encontrado\n");
    return 0;
}
else
    printf("\PHGCMONO.PCX ok\n");
if (access("OHGCMONO.PCX", 0)!=0)
{
    printf("\OHGCMONO.PCX no encontrado\n");
    return 0;
}
else
    printf("\OHGCMONO.PCX ok\n");
if (access("PVGAMONO.PCX", 0)!=0)
{
    printf("\PVGAMONO.PCX no encontrado\n");
    return 0;
}
else
    printf("\PVGAMONO.PCX ok\n");
if (access("OVGAMONO.PCX", 0)!=0)
{
    printf("\OVGAMONO.PCX no encontrado\n");
    return 0;
}
else
    printf("\OVGAMONO.PCX ok\n");

if (access("PVGACOL.PCX", 0)!=0)
{
    printf("\PVGACOL.PCX no encontrado\n");
    return 0;
}
else
    printf("\PVGACOL.PCX ok\n");
if (access("OVGACOL.PCX", 0)!=0)
{
    printf("\OVGACOL.PCX no encontrado\n");
    return 0;
}
else
    printf("\OVGACOL.PCX ok\n");
printf("Revisando comunicaciones con el osciloscopio....");
if ( !IComunicacionesConHw() && !VerComunicaciones == 1 )
{
    printf( "Error\nEl osciloscopio no responde en la dirección
default (300 H).\n");
    printf( "\nEl programa podrá manipular señales almacenadas "
```

```

        "en disco,\n");
printf( "¡pero no podrá adquirir señales. "
        "Recomendaciones: \n\n");
printf( "\1) Revise la conexión externa y vuelva a correr el "
        "programa.\n");
printf( "\2) Vuelva a correr el programa usando el "
        "parámetro /SIMULA\n");
printf( "\4 para simular el enlace con la unidad externa "
        "de hardware.\n");
printf( "\3) Seleccione una dirección de puerto "
        "diferente desde la\n");
printf( "\4 opción CONFIGURACION del programa si "
        "el osciloscopio\n");
printf( "\4 se encuentra conectado a la computadora.\n\n");
printf( "\4Presione una tecla para continuar o ESC "
        "para terminar.\n");
if ( bioskey(0) == 0x11B )
    return 0;
}
else
    printf("OK\n");
iClearVideo();
vNoCursor();
return 1;
}
*/

```

FUNCION:	<i>vOpcionesDelPrograma()</i>
PROTOTIPO EN:	<i>MainFile.H</i>
DESCRIPCION:	<i>Proporciona ayuda en la línea de comandos</i>
ARGUMENTOS:	<i>iParametros</i> <i>información encontrada en la línea de comandos.</i>
REGRESA:	<i>Nada</i>
LLAMA A:	<i>N/A</i>
LLAMADA POR:	<i>main()</i> <i>MainFile.C</i>
No. función:	<i>4</i>

```

*/
void vOpcionesDelPrograma(int iParametros)
{
    if ( (iParametros & OPCIONESDELPROGRAMA) >> 4 )
    {
        printf("¡INDICO v 1.0!\n");
        printf("Uso:\1INDICO [opciones]\n"); printf("Opciones:\n");
        printf("\1/S\1Omitir la portada del programa.\n");
        printf("\1/M\1VForzar el video a monocromático.\n");
        printf("\1/SIMULA\1Simula el enlace de hardware.\n");
        printf("\1/?\1Esta pantalla.\n");
        exit(0);
    }
}

```


12.4.2 Función vCaptura()

La función mas importante del programa es la función de captura de información. En ella se controla la transferencia de datos hacia la PC provenientes de la tarjeta de adquisición de datos diseñada en el capítulo 10. Como se recordará, esta tarjeta se comunica con la computadora usando la tarjeta de interfase de propósito general con la IBM PC diseñada en el capítulo 8, la cual consiste en dos puertos paralelos de 8 bits y 2 puertos paralelos de 4 bits, todos programables como puertos de entrada o de salida. En el capítulo 10 se especificó que la computadora usaría los dos puertos de 8 bits como puertos de transferencia de datos y los 2 puertos de 4 bits como puertos de control.

Teniendo esto como antecedente, el siguiente es el listado parcial del archivo CAPTURA.C, donde se encuentra la función vCaptura().

```

/*
Programa: INDICO.EXE
Versión: 1.0
Fecha: marzo 1994
Uso: Módulo de Osciloscopio de Almacenamiento Digital para el
sistema de Instrumentación Digital Asistido por Computadora.
Archivo: Captura.C
Contenido: MODULO DE CAPTURA Y ANALISIS DE INFORMACION
Proyecto: INDICO.PRJ
Funciones: 1) vCaptura()
           2) vSimula()
           3) vAnálisisDeInformacion()
Sist.de Comp: IBM PC o compatibles
Sist. Op.: MS-DOS v 5.0
Lenguaje: C
Compilador: Borland C++ 3.1 Modelo de memoria Large
Propietario: Armando Romero Domínguez

```

```

*/
#include <IMouse.h>           /* Requerido por Ctrl.h          */
#include <ITeclado.h>        /* InputKey()                   */
#include <dso.h>              /* Macros varias                 */

```

```
#include <struct.h>          /* Estructuras                */
#include <draw.h>            /* vReprintar()              */
#include <ctrl.h>           /* vFallaHw(), vProgramaHw(), ... */
#include <captura.h>        /* vCaptura(), vAnalisisDeinformacion() */
#include <errores.h>        /* macro ADQUISICIONCANCELADA */
```

```
/*-----
  Variables globales definidas en INIT.C
  -----*/
```

```
extern STRUCTDSO far *s1pgDSO;
extern char far *c1pgCanalA;
extern char far *c1pgCanalB;
```

```
/*
```

FUNCION:	vCaptura()
DESCRIPCION:	Programa al hardware de adquisición de datos para comenzar la adquisición y recuperar los datos en los apuntadores globales c1pDatosA y c1pDatosB.
ARGUMENTOS:	Ninguno.
REGRESA:	Nada
LLAMA A :	vProgramaHw() CtrlHw.C iComunicacionesConHw() CtrlHw.C iDatoDeAmplitudListo() CtrlHw.C vErrorDeUsuario() CtrlDlg.C vFallaHW() CtrlDlg.C vAnalisisDeinformacion() Captura.C vEscalaEjes() DrawDSO.C vReprintar() DrawDSO.C iInputKey() LTECLADO.LIB
LLAMADA POR:	vExecutor() Exec.C
REGRESA :	Nada
No. de función:	_1_

```
*/
```

```
void vCaptura(void)
```

```
{
```

```
    register int    riAux;
    int             iFDA ;
    int             iTecla;
```

```
/*-----
  Inicializa la adquisición
  -----*/
```

```
if ( iComunicacionesConHw() == 0 )
{
    vFallaHw();
    return;
}
for (riAux = 0 ; riAux < MEMOXCANALMAX ; riAux++)
{
```

```

        cfpgCanalA[riAux] = 0;
        cfpgCanalB[riAux] = 0;
    }
    sfpgDSO->iCanalAescrito = 0;
    sfpgDSO->iCanalBescrito = 0;
    vProgramaHw( PRGHW_INITHW );
    vProgramaHw( PRGHW_STARTADQ );

/*-----
  Espera a que el osciloscopio señale el fin de la adquisición
  -----*/
for(iFDA= 0 ; iFDA == 0 ; )
{
    iFDA = iDatoDeAmplitudListo();
    if (iFDA == -1)
    {
        vFallaHw();
        return;
    }
    iTecla= iInputKey(-1);
    if ( iTecla == ESCAPE )
    {
        vProgramaHw( PRGHW_STANDBY );
        vErrorDeUsuario(ADQUISICIONCANCELADA);
    }
}

/*-----
  Recupera los datos de amplitud
  -----*/
vProgramaHw( PRGHW_INITRAMRD );
for ( riAux= 0;
      riAux < sfpgDSO->iMemoXcanal && sfpgDSO->iCanalA;
      riAux++)
    cfpgCanalA[riAux] = ucLeerAmplitudA();

vProgramaHw( PRGHW_INITRAMRD );
for ( riAux= 0;
      riAux < sfpgDSO->iMemoXcanal && sfpgDSO->iCanalB;
      riAux++)
    cfpgCanalB[riAux] = ucLeerAmplitudB();

/*-----
  Recupera los datos de frecuencia
  -----*/
sfpgDSO->uiFrecuenciaA = uiLeerFrecuenciaEnA();
sfpgDSO->uiFrecuenciaB = uiLeerFrecuenciaEnB();

/*-----
  Pone al osciloscopio en espera
  -----*/

```

```
vProgramaHw( PRGHW_STANDBY );  
  
/*  
Calcula Vp, Vpp, Frecuencia y escribe la señal en la pantalla  
*/  
vAnálisisDeInformación();  
vEscalaEjes();  
vRepintar();  
}
```

Como puede observarse, esta función consta de cuatro fases: 1) Inicialización de la adquisición, 2) espera del fin de adquisición, 3) recuperación de datos y 4) análisis y visualización en pantalla de la información.

En el inicio de la adquisición, la función **vCaptura()** llama a la función **iComunicacionesConHw()** para preguntar por la bandera **RECDSO**, la cual se encuentra dentro del registro **BANDERAS1**. Si esta bandera es encontrada, significa que el osciloscopio está conectado y se puede realizar la captura. Una vez hecho esto, se limpian tanto el contenido de los bloques de memoria que van a contener los datos adquiridos así como las banderas que indican que esos bloques contienen información. Posteriormente, se escriben los registros **ESCVA**, **ESCVB**, **ESCFA**, **ESCFB**, y **BANDERAS2** con la función **vProgramaHW()**.

La fase de espera del fin de adquisición consiste en un ciclo **for** durante el cual se lee el registro **BANDERAS1**, en donde se encuentran las banderas **FDA** y **RECDSO**. El ciclo **for** termina cuando no se encuentra la bandera **RECDSO** (en cuyo caso termina la función) o hasta que la bandera **FDA** se activa. Adicionalmente, se explora el teclado para verificar si el usuario ha pulsado la tecla **<ESC>**.

En la fase de recuperación de datos la función **vCaptura()** usa dos ciclos **for** para llamar a las funciones **uLeerAmplitudA()**, y

ucLeerAmplitudB(), las cuales recuperan los códigos binarios entregados por los convertidores A/D y los almacenan en los arreglos *cfpgCanalA* y *cfpgCanalB*. Para recuperar la frecuencia de las señales medidas en los canales A y B, se usan las funciones **uiLeerFrecuenciaEnA()** y **uiLeerFrecuenciaEnB()**, las cuales entregan la frecuencia que el hardware midió. Como se vió en el capítulo dedicado al diseño del osciloscopio, se deben usar estos datos y la información en la tabla 10.7 para poder obtener la frecuencia en Hertz.

Finalmente, en la fase de análisis y visualización de información se llama a tres funciones:

vAnálisisDeInformación() - Analiza la información en los arreglos *cfpgCanalA* y *cfpgCanalB* para encontrar los valores mínimos y máximos y calcular los voltajes pico (V_p) y pico a pico (V_{pp}). Igualmente, determina la frecuencia en Hertz.

vEscalaEjes() - Se encarga de convertir los códigos binarios almacenados en *cfpgCanalA* y *cfpgCanalB* en información que pueda visualizarse en la pantalla. La información que esta función entrega se encuentra en los arreglos globales *ifpgVoltajePixelA*, *ifpgVoltajePixelB*, *ifpgTiempoPixelA* e *ifpgTiempoPixelB*.

vRepintar() - Esta función toma los arreglos globales que la función anterior actualizó y los usa para escribir la señal en la pantalla. Igualmente, escribe en la pantalla las escalas, la retícula, el voltaje pico V_p , el voltaje pico a pico V_{pp} y la frecuencia en cada canal.

12.5 Sumario

El desglose de un programa en diagramas de flujo parece ser una práctica no muy popular entre algunos programadores debido a la idea de "trabajar doble" y el retardo en la generación de resultados. Trabajar directamente en la obtención de código es la solución más fácil cuando se escribe un programa que presenta en la pantalla el mensaje "Hola, mundo", pero en un proyecto grande sólo trae como consecuencias diseños endebles que no soportan modificaciones de mantenimiento del programa o crecimiento a nuevas versiones.

Buscando el mejor diseño posible, en este capítulo se proporcionó una descripción detallada del diagrama de flujo, deteniéndolo hasta el punto en que se pudieran insertar las operaciones del osciloscopio como rutinas independientes.

En la transcripción de este diagrama de flujo a lenguaje C son requeridas varias herramientas de software, tal como el manejo de memoria, cadenas de texto, video, teclado y ratón. El siguiente capítulo trata de una manera general la biblioteca de rutinas creada para este programa.

Capítulo 13

Librerías de funciones

13.1 Introducción

La creación de un programa implica escribir funciones básicas de nivel general que ejecuten tareas tales como la ubicación dinámica de memoria, el manejo del monitor, el teclado y el ratón, el uso de cadenas de caracteres, el reconocimiento del hardware del sistema, etc. Dichas funciones son requeridas por prácticamente cualquier programa, de modo que es conveniente tener ese conjunto de funciones dispuestas en forma de librerías. Borland C++ 3.1 ofrece un conjunto bastante amplio de funciones en sus librerías (más de 450), pero no cuenta con todas las requeridas o algunas no son lo suficientemente rápidas debido a su carácter general. Otro conjunto de librerías, Turbo Vision, cuenta con un amplio soporte para el programador, pero requiere programación orientada a objetos.

En este capítulo se describen las librerías requeridas por el programa descrito en el capítulo anterior, indicando las funciones que las componen y su finalidad. Al igual que en el caso del programa, la magnitud de las librerías (alrededor de 3500 líneas de código) impide listar cada una de ellas. Sin embargo, se lista por lo menos una función de cada librería.

13.2 Creación y uso de librerías en Borland C++ 3.1

Borland C++ permite crear librerías mediante uno de los programas de soporte que componen al producto: TLIB.EXE. Este programa permite convertir un archivo objeto en una librería para su uso posterior. El proceso para crear una librería se puede describir en los siguientes pasos:

- 1) Generar un archivo de código fuente con las funciones que se desee incluir en la librería. Tal archivo no debe incluir ninguna función main().
- 2) Compilar el archivo fuente de extensión .C para generar un archivo objeto de extensión .OBJ.
- 3) Usar el programa TLIB.EXE para convertir el archivo objeto .OBJ en una librería .LIB. Por ejemplo, si se desea crear la librería nueva.lib con el archivo nueva.obj, el comando a ejecutar es el siguiente:

```
TLIB nueva.lib + nueva.obj
```

Para usar una librería generada de esta manera, puede trabajarse el programa por proyectos. Para esto, se requiere de un archivo .PRJ donde se listen los archivos .C, y .LIB que el programa requiere. Para generar el archivo .EXE, se declara el nombre del archivo .PRJ dentro del menú "Project" del ambiente integrado de Borland C++ y se ejecuta la opción "Build all" dentro del menú "Compile". El compilador automáticamente compilará todos los archivos .C declarados en el archivo de proyecto y ligará las librerías y los módulos .OBJ generados para crear el programa .EXE.

13.3 Manejo de memoria

El primer bloque de rutinas elaboradas son las de manejo de memoria. Su función es la de solicitar al sistema recursos de memoria mediante las funciones malloc() y farmalloc() de Borland. A continuación se da un listado de las rutinas y su descripción, así como el código de una de ellas.

Funciones	Descripción
cpMalloc()	Ubica memoria con malloc()
ucpMalloc()	Ubica memoria con malloc()
cfpFarMalloc()	Ubica memoria con farmalloc()
ucfpFarMalloc()	Ubica memoria con farmalloc()
vFallaDeMemoria()	Aborta el programa si no hay memoria

```
#include <alloc.h>
#include <stdio.h>
#include <stdlib.h>
/*
```

FUNCION :	<i>cfpFarMalloc()</i>
PROTOTIPO EN :	<i>LMemoria.h</i>
DESCRIPCION :	<i>Ubica bloques de memoria con farmalloc()</i>
ARGUMENTOS :	<i>uTamano Número de bytes a reservar</i>
	<i>iLine Número de línea donde se llama</i>
	<i> a la función.</i>
	<i>(usar la macro <u>__LINE__</u>).</i>
	<i>cpFileName Nombre del archivo donde se</i>
	<i> manda llamar la función</i>
	<i>(usar la macro <u>__FILE__</u>).</i>
REGRESA :	<i>Un apuntador al bloque de datos reservado</i>

```
*/
char far *cfpFarMalloc(size_t uTamano , int iLine, char *cpFileName )
{
    char far *cfpAux;

    cfpAux = (char far*)farmalloc( uTamano );
    if (cfpAux != NULL)
        return(cfpAux);
    else
        vFallaDeMemoria( iLine, cpFileName );
    return NULL;
}
```

13.4 Manejo de textos

La librería de manejo de textos incorpora funciones que permiten desde usar atributos de color para la escritura de texto y recuadros en la pantalla hasta el control del video mediante llamadas a la interrupción 10H de BIOS destinada a video.

Funciones	Descripción
iClearVideo()	Reestablecer el modo de video.
vNoCursor()	Esconde el cursor
vMueveCursor()	Posiciona el cursor.
iRevisaCoordenadas()	Revisa la validez de dos pares de coordenadas.
iEscribeTexto()	Presenta texto en la pantalla.
iCambiarAtributo()	Cambia el color de una parte de la pantalla.
vVentana()	Dibuja una ventana en la pantalla.usando una estructura definida como STRUCTWIN.
vAviso()	Escribe un mensaje en la pantalla
iMideElementos()	Longitud del elemento más largo de un arreglo. Usada por vAviso().
InitVentana()	Inicializa una estructura STRUCTWIN
vMuevePantalla()	Mueve una zona de la pantalla.
vNoBlink()	Pone a 0 el atributo de centelleo.
vReturnBlink()	Pone a 1 el atributo de centelleo.
iModoVideo()	Ajusta el modo de video.
iEscribeTextoFar()	Escribe cadenas tipo far

El siguiente es el listado para dos de esas funciones:

```
#include <dos.h>
```

```

/*
FUNCION :          IClearVideo()
PROTOTIPO EN :    L.Texto.h
DESCRIPCION :    Reestablece el modo de vídeo (limpia la pantalla
                    y retorna el modo de vídeo al default).
ARGUMENTOS :     Ninguno.
REGRESA :        Modo de vídeo.
*/

```

```

int IClearVideo(void)
{
    union REGS  rg;
    int         iModo;

    rg.h.ah=0x0F;
    int86(0x10,&rg,&rg);
    iModo=rg.h.ah;
    rg.h.ah=0x0;
    rg.h.al=iModo;
    int86(0x10,&rg,&rg);
    return(iModo);
}

```

```

/*
FUNCION :          vNoCursor()
PROTOTIPO EN :    L.Texto.h
DESCRIPCION :    Esconde el cursor de la pantalla
ARGUMENTOS :     Ninguno.
REGRESA :        Nada.
*/

```

```

void vNoCursor(void)
{
    union REGS rg;
    rg.h.ah=2;
    rg.x.dx=0x1900;
    rg.h.bh=0;
    int86(0x10,&rg,&rg);
}

```

13.5 Manejo de gráficos

La librería de gráficos es la parte más fuertemente usada en el programa INDICO. Consta de rutinas que permiten inicializar el modo gráfico, escribir caracteres usando un sistema de

coordenadas análogo al usado dentro del modo texto, mostrar en la pantalla archivos con imágenes en formato PCX, etc.

Función	Descripción
<code>iInicializaGraficas()</code>	Inicializa el modo gráfico.
<code>vOutTextGraf()</code>	Escribe una cadena de texto.
<code>vEscribeCharGraf()</code>	Escribe un caracter.
<code>vVentanaGraf()</code>	Dibuja una ventana en gráficos.
<code>vGetCoordsVentanaGraf()</code>	Traduce coordenadas de texto a gráficos.
<code>vSetPalette()</code>	Ajusta el color para un número de paleta.
<code>vSetVideoMode()</code>	Ajusta el modo de video.
<code>vDisplayEGAVGAByte()</code>	Escribe un byte en video EGAVGA.
<code>iDisplayEGAVGAPCXFile()</code>	Recupera en pantalla una imagen a colores almacenada en un archivo PCX.
<code>vOutTextGrafFar()</code>	Escribe una cadena tipo far en modo gráfico.
<code>vDisplayPCX_Mono()</code>	Recupera en pantalla una imagen monocromática almacenada en un archivo PCX
<code>ucfpExpandeRenglon()</code>	Descomprime un renglón de archivo PCX. Usada por <code>vDisplayPCX_Mono()</code> .
<code>vEscribeRenglonPCX()</code>	Escribe un renglón de un archivo PCX. Usada por <code>vDisplayPCX_Mono()</code> .

```
#include <graphics.h>
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
```

```

/*
FUNCION:          vSetPalette()
PROTOTIPO EN:    LGráfica.h
DESCRIPCION:    Ajusta el color para un número de paleta
ARGUMENTOS:    iPaleta: Número de paleta de colores
                   iColor: Color al cual se ajusta la paleta.
REGRESA:        Nada
*/

```

```
void vSetPalette(int iPaleta, int iColor)
```

```
{
    union REGS reg;

    reg.h.ah = 0x10;
    reg.h.al = 0;
    reg.h.bh = iColor;
    reg.h.bl = iPaleta;
    int86(0x10, &reg, &reg);
}

```

```

/*
FUNCION:          vDisplayEGAVGAByte()
PROTOTIPO EN:    LGráfica.h
DESCRIPCION:    Muestra un byte en la pantalla (video EGA y VGA)
ARGUMENTOS:    uliaddress  Dirección de video en donde se ubica el
                   byte a escribir.
                   iColorPlane Plano de colores.
                   ucCh        Byte a escribir.
REGRESA:        Nada
*/

```

```
void vDisplayEGAVGAByte(    unsigned long int    uliAddress,
                           int                          iColorPlane,
                           unsigned char                ucCh)
```

```
{
    char far * cfpDireccion; cfpDireccion = (char far *)0xA000000L +
                                           (unsigned long int)uliAddress;
    outp( 0x3C4 , 2 );
    outp( 0x3C5 , (0x01 << iColorPlane) );
    *cfpDireccion = ucCh;
}

```

```

/*
FUNCION:          iDisplayEGAVGAPCXFile()
PROTOTIPO EN:    LGráficas.h
DESCRIPCION:    Recupera una imagen EGA o VGA de 16 colores
                   (640x350 o 640x350) de un archivo en formato PCX.
ARGUMENTOS:    cFileName  Nombre del archivo completo,
                   incluyendo la extensión.
                   i_EGA_o_VGA Modo de video. Con -1 no ajusta el
                   modo.
REGRESA:        -1 En caso de error
*/

```

```

*/
int iDisplayEGAVGAPCXFile(char cFileName[], int i_EGA_o_VGA)
{
    register unsigned int    ruiVideoOffset,
                             ruiRenglonActual;
    FILE                    *fsave;
    unsigned char           ucCh,
                             ucCh1,
                             ucRojo,
                             ucVerde,
                             ucAzul,
                             color;
    unsigned int            uiFinLinea,
                             uiPlano,
                             uiAux,
                             uiRepeticion,
                             uiXini,
                             uiYini,
                             uiXfin,
                             uiYfin;

    if ((fsave = fopen(cFileName,"rb")) == NULL)
        return(-1);
    else
    {
        ucCh = fgetc(fsave);
        if (ucCh != 0x0A)
        {
            fclose(fsave);
            return(-1);
        }
    }
    if (i_EGA_o_VGA != -1)
    {
        vSetVideoMode(i_EGA_o_VGA);
        vSetVideoMode(i_EGA_o_VGA);
    }
    for (ruiVideoOffset=1; ruiVideoOffset<4; ruiVideoOffset++)
        ucCh = fgetc(fsave);
    uiXini = getw(fsave);
    uiYini = getw(fsave);
    uiXfin = getw(fsave);
    uiYfin = getw(fsave);
    for (ruiVideoOffset=12; ruiVideoOffset<16; ruiVideoOffset++)
        ucCh = fgetc(fsave);
    for (ruiVideoOffset=0; ruiVideoOffset<16; ruiVideoOffset++)
    {
        ucRojo = fgetc(fsave)/85;
        ucVerde = fgetc(fsave)/85;
        ucAzul = fgetc(fsave)/85;
        color = ((ucRojo & 0x01) << 5) |
                ((ucRojo & 0x02) << 1) |

```



```

    }
    fclose(fsave);
    return(uiXfin);
}

```

13.5.1 Manejo de pixeles

La librería de manejo de pixeles tiene como finalidad acelerar la graficación de pixeles en el modo gráfico. Realizando pruebas de rendimiento, la rutina putpixel() de Borland tarda 14 segundos en llenar una pantalla VGA pixel por pixel, mientras que la función vPonPixelVGA() que se muestra más adelante tarda solamente 10 segundos. La pruebas fueron realizadas en una computadora 80386DX de 40 MHz.

Funciones	Descripción
vPonPixelHGC()	Escribe un pixel en video Hercules
vPonPixelBlancoHGC()	Escribe un pixel blanco en video Hercules
vPonPixelNegroHGC()	Escribe un pixel negro en video Hercules
vPonPixelVGA()	Escribe un pixel en video EGA/VGA

```

/*
FUNCION : vPonPixelVGA()
PROTOTIPO EN : Pixel.h
DESCRIPCION : Escribe un pixel en la pantalla EGA/VGA
ARGUMENTOS : IX,IY      Coordenadas del pixel a escribir
              iColor     Color a escribir de 0 a 15
REGRESA : Nada.
*/

```

```

void vPonPixelVGA(int iX, int iY, int iColor)
{
    #include <dos.h>
    unsigned int    uiOffset;
    int             iMask;

```



```
uiOffset = (long)iy * 80L + ((long)ix / 8L);
iMask = 0x80 >> (ix % 8);
_ES = 0xA000;
_BX = uiOffset;
_CX = iColor;
_AX = iMask;
asm MOV AH,AL
asm MOV AL,08
asm MOV DX,03CEH
asm OUT DX,AX
asm MOV AX,0FF02H
asm MOV DL,0C4H
asm OUT DX,AX
asm OR ES:[BX],CH
asm MOV BYTE PTR ES:[BX],00H
asm MOV AH,CL
asm OUT DX,AX
asm MOV BYTE PTR ES:[BX],0FFH
asm MOV AH,0FFH
asm OUT DX,AX
asm MOV DL,0CEH
asm MOV AX,0003
asm OUT DX,AX
asm MOV AX,0FF08H
asm OUT DX,AX
}
```

Como puede observarse, la función `vPonPixelVGA()` logra superar a `putpixel()` usando código en ensamblador. Para lograr esto, la función usa la palabra clave de Borland `asm` para escribir directamente código en ensamblador.

13.6 Manejo de cadenas de caracteres

La librería de manejo de cadenas de caracteres realiza funciones análogas a las contenidas en la librería estándar de Borland, exceptuando algunas de ellas que no están disponibles.

Funciones	Descripción
vAllTrim()	Elimina espacios de una cadena.
vCambiaCaracteres()	Intercambia caracteres en una cadena.
vCopyStrFarStrFarConNull()	Copia cadenas
iComparaStrFarStrNorm()	Compara una cadena con otra
vCopyStrFarStrNormConNull()	Copia cadenas
iLongitFarStr()	Mide cadenas
vCopyFarStr2FarBuffer()	Copia cadenas a un buffer

```

/*
FUNCION :      vCambiaCaracteres()
PROTOTIPO EN:  cadena.h
DESCRIPCION :  Intercambia caracteres cCarac1 por caracteres cCarac2
                en una cadena cpCad
ARGUMENTOS :   cpCad      Cadena a explorar.
                iNumCar    Numero maximo de caracteres totales
                                en la cadena.
                cCarac1    Caracter a buscar.
                cCarac2    Caracter con el que se va a reemplazar.
REGRESA :      Nada.
*/

```

```

void vCambiaCaracteres(   char   *cpCad,
                          int     iNumCar,
                          char    cCarac1,
                          char    cCarac2)
{
    int    i, j;

    for (i=0; i<iNumCar && *(cpCad+i)!='\0'; i++);
    for (j=0; j<i; j++)
        if (*(cpCad+j)==cCarac1)
            *(cpCad+j) =cCarac2;
}

```

13.7 Manejo del ratón

Las funciones de manejo del ratón contenidas en esta librería inicializan y reconocen al ratón, preguntan por el estado de

botones presionados y la posición en la pantalla del cursor del ratón.

Funciones	Descripción
iMouseInit()	Inicializa al mouse.
vMousePointer()	Muestra o esconde el cursor del mouse.
vMouseStatTex()	Regresa el estado del mouse en modo texto.
vMouseClose()	Libera al mouse.
iCheckMouseDriver()	Revisa si está instalado el driver del mouse.
vMouseStat()	Regresa el estado del mouse.
vResetMouse()	Reinicializa al raton

```

/*
FUNCION : iCheckMouseDriver(void)
PROTOTIPO EN: LMouse.h
DESCRIPCION : Revisa si está instalado el driver del mouse.
ARGUMENTOS : Ninguno.
REGRESA : 0 El driver del mouse no se ha instalado.
          1 El driver del mouse ya está instalado.
*/

```

```

int iCheckMouseDriver(void)
{

```

```

#include <dos.h>

```

```

union REGS InRegs, OutRegs;
struct SREGS SegRegs;
unsigned long ulDireccion;
unsigned char uc1erByte;

```

```

InRegs.x.ax = 0x3533;
intdosx ( &InRegs, &OutRegs, &SegRegs);
ulDireccion = (((long) SegRegs.es) << 16) + (long) OutRegs.x.bx ;
uc1erByte = (unsigned char) * (long far *)ulDireccion;
if ( (ulDireccion == 0L) || (uc1erByte == 0xCF) )
    return 0;
else
    return 1;
}

```

13.8 Manejo del teclado

Las rutinas del manejo del teclado proporcionan una mejora a la tarea desempeñada por `bioskey()` de Borland, permitiendo la temporización de la recepción de teclas.

Funciones	Descripción
<code>iInputKey()</code>	Captura teclas pulsadas.
<code>iSystemKeys()</code>	Presenta el estado de las teclas <code>shiflt</code> , <code>ctrl</code> , <code>alt</code> y <code>num</code>
<code>iInputAllKeys()</code>	Captura teclas normales y del sistema.

```
#include <time.h>
#include <bios.h>
```

FUNCION :	<code>iInputKey()</code>
PROTOTIPO EN:	<code>LTeclado.h</code>
DESCRIPCION :	<i>Captura teclas pulsadas por el usuario</i>
ARGUMENTOS :	<code>IT</code> <i>Tiempo en segundos durante los cuales se espera al usuario para que presione una tecla</i>
REGRESA :	<i>El código correspondiente a la tecla presionada.</i>

```
*/
int iInputKey( int IT )
{
    int iC=0;
    time_t tTiempoInicial=0,
           tTiempoFinal;
    double dTiempoTranscurrido;

    time(&tTiempoInicial);
    time(&tTiempoFinal);
    dTiempoTranscurrido= difftime( tTiempoFinal , tTiempoInicial );
    if (tTi=0)
    {
        while( dTiempoTranscurrido <= IT )
        {
            iC=bioskey(1);
            if (iC!=0)
            {
                iC=bioskey(0);
                return (iC);
            }
        }
    }
}
```

```
    }
    time(&TIempoFinal);
    dTiempoTranscurrido= difftime(TIempoFinal,
    TIempoinicial);
}
if ( dTiempoTranscurrido > IT)
    return 0;
}
else
    while(IC==0)
    {
        IC=bioskey(1);
        if (IC!=0)
        {
            IC=bioskey(0);
            return (IC);
        }
    }
return IC;
}
```

13.9 Sumario

En este capítulo se dio una breve descripción de las librerías usadas en el desarrollo del software de este proyecto, así como los listados de unas cuantas funciones que las componen.

La ventaja de una construcción de librerías como las presentadas se hace patente al momento de desarrollar proyectos nuevos, ya que el tiempo de desarrollo se reduce por contar con un conjunto de rutinas de propósito general. Esto puede ser importante en la fase de desarrollo de otros módulos de INDICO.

Con este capítulo queda concluida la descripción del programa que maneja al osciloscopio digital. En el siguiente capítulo comienza la guía de usuario, donde se darán los detalles de la instalación y uso del osciloscopio desarrollado.



Parte 5

Manual de usuario y conclusiones

Objetivos:

*Presentar una guía de uso del sistema diseñado, incluyendo la instalación del hardware y el software.
Analizar el producto terminado indicando posibles mejoras para siguientes versiones.*

Capítulos

- Guía de usuario
- Conclusiones y comentarios finales

Capítulo 14

Guía de usuario

14.1 Introducción

En este capítulo se manejará información destinada a la creación de un manual de usuario para INDICO. Para esto, se tomarán elementos referidos anteriormente en los capítulos de especificaciones y de desarrollo. El propósito de ésta guía es presentar en forma compacta la información necesaria para que un usuario de INDICO sea capaz de instalarlo y usarlo mediante la simple lectura de éstas páginas.

14.1.1 ¿Qué es INDICO?

INDICO es un sistema de instrumentación digital asistido por computadora que trae la funcionalidad de un laboratorio de instrumentación al ambiente de una computadora personal IBM o compatible. En su fase inicial, el primer instrumento incorporado es un osciloscopio de almacenamiento digital. Otros módulos a ser incorporados incluyen un analizador de circuitos TTL, un comprobador de transistores y un multímetro digital.

El sistema incluye tres partes: el software de control, una tarjeta de interfase de puerto paralelo, y los módulos del hardware de instrumentación.

Software de control. Se encarga de interpretar los datos enviados por el módulo de instrumentación, así como de la programación remota del mismo. Se encuentra contenido en un sólo diskette.

Interfase de puerto paralelo. Es una tarjeta que se inserta en el bus de expansión de la computadora. Su función es la de servir como interfase entre los módulos de instrumentación y la computadora. El carácter genérico de ésta tarjeta permite usarla por sí misma como interfase para otro tipo de proyectos, tal como un controlador de dispositivos externos o para monitoreo de eventos, por ejemplo.

Hardware de instrumentación. Consta en su fase inicial del módulo de osciloscopio digital y cable de conexión con la tarjeta de interfase de puerto paralelo. En éste módulo es donde se realiza la adquisición de datos, los cuales son enviados a la computadora para su presentación y proceso.

14.1.2 Requerimientos de hardware y software

El equipo requerido para el uso de INDICO como osciloscopio digital es el siguiente:

- 1) Una computadora personal IBM o compatible con las siguientes características:
 - Microprocesador 8088 o mejor.
 - 640 KB de memoria.
 - 1 Puerto paralelo.
 - 1 unidad de disco flexible (disco duro deseable).

- Tarjeta de video Hercules monocromático o compatible, o VGA de colores.
 - 1 ranura de expansión de 8 bits ISA.
 - MS-DOS 3.3 o superior instalado.
- 2) Un ratón (mouse) Microsoft con un driver versión 8.2 si se desea hacer operaciones de cursor.
 - 3) Una impresora STAR NX-1000 o compatible si se desean imprimir las señales capturadas.
 - 4) La interfase de puerto paralelo y el módulo de osciloscopio digital.

Las computadoras en las que se garantiza la operación del sistema son:

- Printaform XT de 10 Mhz con microprocesador 8088 de Intel y monitor Hercules monocromático con tarjeta de video Magic Combo.
- Gamma 88 XT de 12 MHz con microprocesador V20 de NEC y monitor Hercules monocromático con tarjeta de video Magic Combo.
- Auva 390 de 40 MHz con microprocesador 80386DX de AMD y monitor súper VGA con tarjeta de video Trident.

14.1.3 Qué se necesita saber para usar INDICO

El usuario de INDICO deberá tener conocimientos generales sobre el uso de osciloscopios y computadoras personales. Referente a

ésto último, un conocimiento sobre el uso de Windows brindará ayuda para el manejo del ratón.

14.1.4 Especificaciones técnicas de INDICO

Módulo de Software:

- Espacio en disco: 250 KB
- Memoria requerida: 640 KB
- Sistema operativo: MS-DOS 3.3 o superior.
- Vídeo soportado: Hercules monocromático y VGA a colores o monocromático.

Módulo de Hardware (osciloscopio digital):

- Número de canales de entrada analógica: 2
- Rango de voltaje de entrada: ± 5 V.
- Ancho de banda: 1 millón de muestras por segundo.
- Resistencia de entrada de $2\text{ M}\Omega$.
- 4 KB de memoria por canal.
- Puertos paralelos de comunicación: 2 de 8 bits y 2 de 4 bits.
- Operación programable desde la computadora.
- Conversión A/D de 8 bits, no multiplexada.

14.2 Comenzando INDICO

14.2.1 Instalación

14.2.1.1 Instalación del hardware.

La figura 14.1 muestra la tarjeta de interfase de puerto paralelo, la cual debe insertarse en la ranura de expansión de la computadora. La tarjeta cuenta con un bloque de 8 switches que permiten seleccionar la dirección de puerto a usar. La tabla 14.1 muestra la dirección que se puede seleccionar al poner un switch en la posición ON:

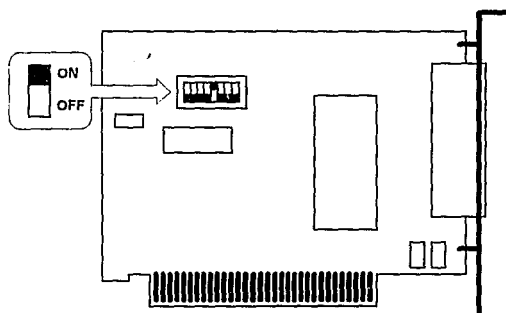


Figura 14.1 Tarjeta de interfase con la IBM PC

Switch en ON	Dirección	Switch en ON	Dirección
1	200 Hex	5	300 Hex
2	220 Hex	6	320 Hex
3	240 Hex	7	340 Hex
4	260 Hex	8	360 Hex

Tabla 14.1 Selección de puertos en la tarjeta de interfase con la IBM PC.

De estas direcciones, el default es la 300 H, que se encuentra libre en las PC. Si no se tiene puerto de juegos en la computadora, se puede ocupar la dirección 200 H.

Una vez colocada la tarjeta, se debe usar el cable de conexión proporcionado con INDICO para conectar la tarjeta instalada con el módulo de osciloscopio digital y conectar éste a la toma de corriente. Esto finaliza la instalación de hardware.

14.2.1.2 Instalación del software.

El software consta de los siguientes archivos:

INDICO.EXE
PVGACOL.PCX
OVGACOL.PCX
PVGAMONO.PCX
OVGAMONO.PCX
PHGCMONO.PCX
OHGCMONO.PCX

Copie estos archivos a un directorio en el disco duro, o use el disco original proporcionado con INDICO para usarse desde una unidad de discos flexibles.

14.2.2 Iniciando el programa INDICO

Para comenzar el programa INDICO se deberá cambiar al directorio en donde se encuentre instalado INDICO y teclear:

INDICO

Con esto, el programa empezará a cargarse. Si no se instaló el hardware, el usuario podrá ser capaz de usar todas las características de INDICO, excepto la captura de señales.

El programa puede correrse con parámetros de la siguiente manera:

INDICO [opciones]

donde las opciones pueden ser:

- | | |
|---------|--|
| /S | Saltar la pantalla de la portada e ir directamente a la pantalla del osciloscopio. |
| /M | En una computadora VGA de colores, obliga a simular video en modo monocromático. |
| /SIMULA | Le indica al programa que opere en el modo de simulación de captura de señales. Este modo se usará para fines demostrativos del programa cuando no se tiene el módulo de hardware conectado. |
| /? | El programa muestra únicamente una pantalla donde se indican estas opciones y después termina. |

14.2.3 La distribución de la pantalla

La figura 14.2 muestra la portada del programa, la cual aparece después de que el programa ha comenzado. Esta portada permanecerá durante 15 segundos en la pantalla o hasta que se presione cualquier tecla, lo que pase primero.

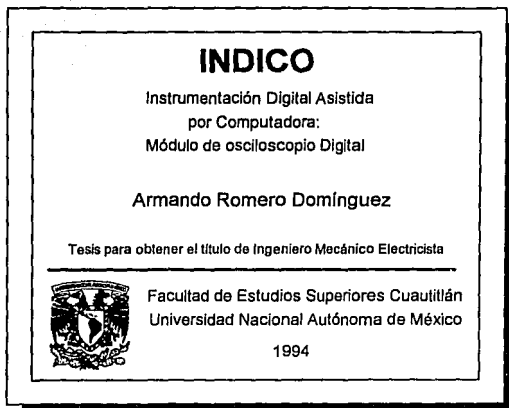


Figura 14.2 Portada de INDICO

La figura 14.3 muestra la pantalla de operación del programa, la cual aparecerá después de la portada. En esta figura pueden observarse las siguientes áreas:

- 1.- **Título.** Contiene la leyenda "Osciloscopio Digital".
- 2.- **Menú.** En esta zona se presentan los menús del programa.
- 3.- **Información de canales y escala.** consta de 6 recuadros en los cuales se indican los canales activos del osciloscopio, así como el tipo de acoplo en cada canal (corriente directa o alterna). También se muestran los valores actualmente seleccionados para la escalas de tiempo y voltaje.

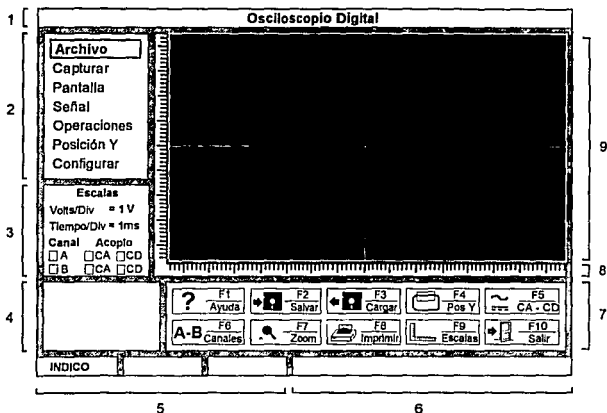


Figura 14.3 Pantalla principal de INDICO

- 4.- **Resultados de operaciones de cursor.** En ella se muestran los resultados de las operaciones de cursor.
- 5 **Barra de estado.** Indica las opciones del menú seleccionadas actualmente.
- 6.- **Ayuda de operaciones de cursor y zoom.** Proporciona al usuario ayuda sobre el funcionamiento de los botones del ratón durante las operaciones de cursor y Zoom por ventana.
- 7.- **Iconos.** Consta de 10 cuadros con un icono, el nombre de una tecla y el nombre del comando que se puede ejecutar.

- 8.- **Divisiones de escala.** Presenta una regla con divisiones que representan la escala seleccionada.
- 9.- **Monitor del osciloscopio.** En esta zona se presentan las formas de onda capturadas por el módulo de hardware, y las ventanas de diálogo que presenta el programa.

14.3 Uso de INDICO

INDICO es un programa conducido por menú. El usuario tiene acceso a todas las opciones del menú seleccionando las opciones mediante el teclado o el ratón. Al usar el ratón, el botón derecho equivale a ESCAPE y el botón izquierdo a ENTER. Las siguientes secciones describen la operación de los menús del programa.

14.3.1 Funciones de archivo

Las funciones de archivo y del sistema que incluye INDICO son:

Cargar	Carga una señal almacenada en disco.
Salvar	Salva la señal actual en disco.
Imprimir	Imprime la señal actual.
Salir	Termina el programa.
Información	Presenta información de la versión.

14.3.2 Captura de una señal

La función más importante del sistema es la de captura de señales. Las opciones son:

Capturar	Captura una señal considerando todas las opciones de captura que el usuario haya incluido. El usuario deberá tener cuidado de no sobrepasar el rango de voltaje de entrada del módulo de hardware de osciloscopio digital.
Muestreo	Solicita al usuario la frecuencia más aproximada de la señal a medir, para así determinar la frecuencia óptima de muestreo.

14.3.3 Funciones relativas a la pantalla

Vistas	Permite al usuario ajustar la visualización de la señal quitando o incluyendo una retícula, un eje de escalas o cada canal.
Limpiar	Limpia la información capturada de la pantalla.
Trazo	Indica si la señal capturada se mostrará en forma punteada o por un trazo continuo.
Zoom	Permite realizar un acercamiento a la señal marcando un área de la pantalla con el ratón o a toda la señal con acercamientos predefinidos. Al usar un mouse, la zona se marca usando el botón derecho del ratón.

14.3.4 Control y estadísticas de la señal

Escalas	Permite seleccionar al usuario las escalas en los ejes.
Memoria	Indica la cantidad de memoria seleccionada por canal.

Acoplo	Indica el acoplo de señal (c.a. o c.d.) en cada canal.
Información	Muestra el voltaje pico, el voltaje pico a pico y la frecuencia de la señal en cada canal.

14.3.5 Operaciones de cursor

Delta V	Usando el botón izquierdo del ratón, el usuario cambiará la posición entre dos líneas horizontales, las cuales representarán una diferencia de voltaje que se indicará en la pantalla. Se usa el botón derecho del ratón para terminar la operación.
Delta T	Realizan una operación similar a la anterior.
Frecuencia	La operación es análoga a las anteriores.

14.3.6 Control de la posición Y de la señal

Puede cambiarse la posición de la señal en el eje vertical usando esta opción. El usuario moverá la tierra de la señal seleccionada usando el botón izquierdo del ratón y dará por terminada la operación usando el botón derecho. Al hacer esto, la señal se redibujará en su nueva posición.

14.3.7 Configuración

Es permitido configurar el puerto al cual se conecta el módulo de osciloscopio. Las direcciones disponibles son la 200H, 220H, 240H, 260H, 300H, 320H, 340H y 360H. En lo que respecta a la impresora, sólo se encuentra disponible la STAR NX-1000.

14.3.8 Funciones en los iconos

Los comandos de los iconos pueden ejecutarse con una tecla de función, o seleccionando el icono con el botón izquierdo del ratón. Las funciones son las siguientes:

- F1** **Ayuda.** Mostrará una caja de diálogo con ayuda del programa.
- F2** **Salvar.** Operación idéntica a la descrita en el menú.
- F3** **Cargar.** Operación idéntica a la descrita en el menú.
- F4** **Posición Y.** Moverá la señal en el canal activo hacia arriba y después hacia abajo en pasos de 20 píxeles cada vez que este icono sea invocado.
- F5** **CA-CD.** Cambiará el tipo de señal a capturar en los canales A y B.
- F6** **Canales.** Conmutará entre la visualización del canal A o el B (sólo uno a la vez).
- F7** **Zoom.** Igual que el zoom por ventana descrito en las operaciones de menú.
- F8** **Imprimir.** Operación idéntica a la descrita en el menú.
- F9** **Escalas.** Igual a la operación en el menú, solo que este icono invocará tanto a la caja de diálogo de escala en V y escala en T.
- F10** **Salir.** Operación idéntica a la descrita en el menú.

Capítulo 15

Conclusiones y comentarios finales

El desarrollo de un proyecto de esta naturaleza ofrece características de diseño muy interesantes, ya que combina el desarrollo de hardware y software destinados a aplicarse en una computadora personal. Teniendo en cuenta que cada una de estas áreas ofrece en sí misma los elementos suficientes como para un proyecto de tesis, éste trabajo implicó muchos esfuerzos al reunir los suficientes conocimientos de ambas áreas como para llevar a su finalización el proyecto. Sin embargo, es un perfecto ejemplo de un trabajo de ingeniería. Lo más común en un proyecto de ingeniería es encontrar que se requieren elementos tales como la mecánica, la física, la electrónica, la informática o las matemáticas. Un proyecto de tesis que se aísle en una sola de éstas áreas puede ser más fácil de desarrollar, pero no se puede considerar tan apegado a la realidad como lo es un proyecto multidisciplinario. Incluso en el presente proyecto se podrían llevar más allá las cosas si se analizara el aspecto comercial que pudiera tener. Esto implicaría analizar productos análogos disponibles en el mercado, analizar el mercado para tales productos, el costo que lleva la producción, el volumen de producción, la mercadotecnia, la publicidad y muchos factores más. Lo anterior resalta que la ingeniería no es independiente. Requiere comunicación con las áreas administrativas y humanísticas para su desenvolvimiento. De ahí se observa la importancia de que desde su formación el alumno de ingeniería reciba elementos básicos de estas áreas.

Durante el desarrollo del proyecto fue algo muy importante analizar las fases requeridas para llevarlo a cabo. Desde la etapa de planeación hasta la construcción y pruebas de prototipo final fue importante llevar orden y estructura. A través de la experiencia ganada se pueden resumir las siguientes reglas de diseño:

- 1) Analizar las tareas a realizar y verificar si existen circuitos o librerías existentes que puedan realizar el trabajo.
- 2) Si la tarea puede realizarse por hardware o software, es mejor realizarla por software, ya que los cambios y las pruebas son más fáciles de hacer.
- 3) Llevar un cuaderno de notas con la información necesaria de circuitos, tablas, diagramas, etc. La información puede estar disponible en folletos, libros o revistas, pero a veces es difícil recordar en donde se encontró cierta información. Tener todo en un sólo lugar permite tener la información a la mano.
- 4) Buscar siempre soluciones alternas. Algunas veces funciona el apartarse de las ideas convencionales.
- 5) Nunca suponer nada. Las "condiciones ideales" no reciben su nombre por nada, además de que la teoría y la práctica muchas veces solo están teóricamente relacionadas.
- 6) Consultar todas las fuentes posibles de información. En el caso del desarrollo del software, por ejemplo, la información para recuperar imágenes de archivos en formato PCX se encontró en un libro que trataba sobre graficación de fractales, un tema de matemáticas.

- 7) La complejidad de un proyecto es directamente proporcional con la flexibilidad que tenga la operación de éste. Al establecer las especificaciones de diseño no se deben de exagerar las capacidades de un sistema, o de lo contrario el desarrollo de éste se prolongará por demasiado tiempo.
- 8) Saber lo que se quiere hacer antes de empezar nada. La solución de un problema siempre empieza con la descripción detallada del mismo.
- 9) Pensar por uno mismo. Las ideas vistas en otros diseños pueden ser buenas, pero el análisis propio de las soluciones disponibles y el problema presente es el camino más corto hacia la terminación de un diseño.
- 10) Conservar márgenes de seguridad. Esta es una práctica de ingeniería muy sana, ya que evita errores debidos a saturaciones o deficiencias en los rangos de operación tanto en hardware como en software.
- 11) Durante la fase de diseño se debe imaginar siempre lo peor. En el caso del software, por ejemplo, se debe considerar que el usuario está dispuesto a hacer todo lo posible para que el programa falle (algunos usuarios presentan una extraña satisfacción al lograr esto). Ya sea en el caso del hardware o el software, siempre es normal encontrar que el usuario olvidó alguna de las recomendaciones de uso, o que se puede alcanzar una condición que este fuera del rango de operación. Un proyecto bien diseñado debe tomar en cuenta cualquier origen posible de falla, desde las internas (diseño) hasta las externas (usuario).
- 12) La mejor herramienta es aquella que se sabe usar. Si el tiempo lo permite, siempre es bueno aprender cosas

nuevas; pero en el caso contrario, las herramientas conocidas son las que se deberán emplear.

Interfazar hardware a una computadora personal es posible debido a que ésta fue diseñada con la visión de poder extender sus capacidades normales. Esta misma filosofía de diseño fue la que se trató de dar en el desarrollo del osciloscopio digital, otorgándole la capacidad de incrementar sus capacidades de instrumentación, tal como lo propone el nombre global del proyecto (INDICO: Instrumentación Digital asistida por Computadora), de modo tal que fuera posible añadir más módulos de instrumentación al diseño inicial. Para lograr esto, al desarrollar el módulo de osciloscopio digital se analizó el hardware disponible en la computadora personal con dos motivos: conocer la manera de añadir el diseño a la PC y analizar la forma en que ésta logra sus capacidades de expansión.

El análisis de la arquitectura de la computadora personal fue una de las mejores inversiones de tiempo, ya que permitió observar cómo resolvieron otros ingenieros diversos problemas de diseño. Una de las mejores maneras de aprender, además de la práctica, es teniendo un problema real y la solución que se le dió. De esta forma, es posible aplicar las mismas técnicas de diseño en otras partes, o desarrollar técnicas propias basadas en aquellas conocidas.

Con respecto al desarrollo del software, éste no fue de ninguna manera simple, ya que antes de tener el primer modelo del programa fue necesario estudiar la manera de controlar el video, el ratón, el teclado, el timer, el canal de E/S y otras características internas de la computadora. Tener control sobre estos elementos fue un trabajo arduo que llevó varios meses para su culminación, incluso más que el desarrollo mismo del hardware. Sin embargo, el software tiene una ventaja sobre el hardware, y esta radica en su

capacidad de ser reusado. Los esfuerzos realizados en la creación de una librería de rutinas como la bosquejada en el capítulo 13 dan como fruto la capacidad de crear otros programas teniendo un conjunto de herramientas ya disponibles, de tal manera que se puede reducir el tiempo de desarrollo para futuros proyectos. Es importante hacer notar que todo el software de ésta tesis ha sido diseñado de una manera modular con el fin de lograr esto.

El diseño del programa principal fue llevado usando diagramas de flujo y reglas en la programación orientadas a facilitar la creación y mantenimiento del programa. Esto hizo una diferencia notable al hacer comparaciones contra programas hechos de manera personal antes de efectuar esta tesis y el presente trabajo.

Una de la inquietudes en la elaboración de éste proyecto fue la de presentar lo que es posible realizar si se aprovecha de la mejor manera posible los recursos que una computadora personal ofrece, aún los de una modesta computadora IBM PC XT de 10 Mhz. Una computadora personal puede ser una gran ayuda para los alumnos de ingeniería cuando la usan como máquina de escribir para elaborar sus reportes de laboratorio, pero se debe resaltar que una computadora esta muy por encima de ésto. Con el presente proyecto se intenta que la computadora también sea parte de las prácticas de laboratorio, ofreciendo las capacidades suficientes como para que el alumno pueda capturar, manipular y almacenar las señales que mida en una práctica de laboratorio. Otra facilidad que se puede encontrar es en el módulo de impresión que se ha incorporado en el programa, el cual permite al alumno incluir en un reporte de actividades la señal exacta que midió, sin tener que recurrir a dibujar manualmente lo que observó.

Con respecto a la computadora en la que se decidió realizar el proyecto, la elección de una computadora personal IBM PC asegura que se tendrá una base sólida en al cual opere el sistema,

ya que es un tipo de computadora con un éxito comercial muy grande que se ha caracterizado por crecimientos constantes en sus capacidades de procesamiento y por el soporte desde hardware y software a modelos anteriores. Esto último es muy importante, ya que desde la aparición de la IBM PC en 1981 hasta la fecha, han existido cinco generaciones de microprocesadores Intel (el 8088, el 80286, el 80386, el 80486 y el Pentium), cada uno con diferentes características de procesamiento que superaban a la generación predecesora. Estos microprocesadores se han incorporado a cada nuevo modelo de computadora personal de tal manera que para el usuario final el único cambio era que obtenía una computadora más poderosa, y el software que tenía anteriormente podía ser conservado. Gran parte del éxito de la IBM PC se debe a esto.

Los cambios en la industria de las computadoras personales han sido muy rápidos y la tendencia es que esto continúe así. Al comienzo de ésta tesis se intentó pisar terreno seguro marcando las limitaciones del proyecto en base a las limitaciones que imponía un modelo básico de PC, la computadora tipo XT con un microprocesador 8088 a 4.77 Mhz. Esto fue para asegurar que el proyecto tendría terreno de aplicación en cualquier computadora disponible. Sin embargo, en el momento de la terminación del proyecto, éste modelo de computadora ha sido descontinuado y ya no se encuentra en venta. Es claro que mantener compatibilidad con este modelo es absurdo. Incluso la computadora que le sucedió, la basada en el microprocesador 80286, se encuentra obsoleta. En el momento de escribir esto, la mejor PC es aquella basada en el microprocesador Pentium de Intel con reloj a 66 Mhz. En el futuro, se vislumbran computadoras con microprocesadores AMD (Advanced Micro Devices, uno de los competidores de Intel) 486 con reloj triplicado de 120 Mhz, computadoras con soporte de dos microprocesadores Pentium, y computadoras personales con microprocesadores RISC (como el Alpha de DEC, de 150 Mhz).

Todo lo anteriormente escrito no le quita validez a las especificaciones tomadas para éste proyecto, ya que los cambios y mejoras en computadoras van a continuar y forzosamente se debe marcar un punto de inicio con respecto al cual se va a diseñar el proyecto. En su momento, el modelo de referencia fue la IBM PC XT. Lo más importante del proyecto INDICO es que desde su concepción ya se encontraba preparado para ejecutar en computadoras más avanzadas. El desarrollo del software, incluso, fue realizado en una computadora 386DX de 40 Mhz. El soporte de video incluye al monitor Hercules de la computadora XT y al monitor VGA a colores, el estándar actual. Quizá el único problema sería la lenta velocidad de captura de datos del osciloscopio digital, pero es solamente aparente, ya que la velocidad del bus de expansión con arquitectura ISA de la computadora personal actual tiene el mismo límite que en las primeras computadoras.

Se acaban de describir los prospectos para las computadoras personales. Ahora bien, ¿que proyección se puede encontrar para el presente diseño? Podemos mencionar los siguientes aspectos para desarrollos futuros de INDICO:

- 1) En el caso del software, es posible ocupar el programa actual y mandarlo llamar desde un menú que seleccione módulos de INDICO. Otra posibilidad es reescribir el programa para otro sistema operativo, como Windows, sin que sea requerido cambiar nada en el hardware.
- 2) Para el hardware, pueden añadirse a futuro módulos de multímetro digital, analizadores de circuitos TTL y transistores, analizador de espectro, analizador de estados lógicos, etc.

Apéndice A

Definición de términos para convertidores A/D y D/A

Tiempo de conversión: El tiempo requerido para una medición completa por un convertidor analógico a digital.

Error de modo común (DC Common-mode Error): Esta especificación se aplica a convertidores A/D con entradas diferenciales. Es el cambio en el código de salida que ocurre cuando el voltaje analógico en las dos entradas es cambiado en una cantidad igual. Es expresado en LSBs.

No linealidad diferencial: Idealmente, dos códigos digitales adyacentes cualesquiera corresponden a voltajes analógicos que tienen una separación exacta de un LSB. La no linealidad diferencial es una medida del peor caso de desviación del paso de un LSB ideal. Por ejemplo, un convertidor D/A con un cambio en la salida de 1.5 LSB por un cambio en el código digital de 1 LSB, exhibe una no linealidad diferencial de 1/2 LSB. La no linealidad puede ser expresada en fracciones de bits o como un porcentaje de la escala total. Una no linealidad diferencial mayor a 1 LSB provocará una función de transferencia no monótona en un convertidor D/A y códigos perdidos en un A/D.

Error de ganancia (error de escala total): Para un convertidor A/D, es la diferencia (usualmente expresada en LSBs) entre el voltaje de entrada que idealmente debería producir un código de salida a escala total y el voltaje real que genera a éste código. Para convertidores D/A, es la diferencia entre el voltaje de salida (o corriente) con un

código de entrada a escala total y el voltaje ideal (o corriente) que debería generarse con éste código.

Coefficiente de ganancia de temperatura (Coeficiente de temperatura a escala total): El cambio en el error de ganancia dividido por el cambio en la temperatura. Usualmente expresado en partes por millón por grado Celsius (ppm/°C).

No linealidad integral (error de linealidad): El peor caso de desviación de la línea trazada entre los extremos (cero y escala total). Puede ser expresado como un porcentaje de la escala total o como fracciones de LSB.

LSB (Least Significant Bit): Bit menos significativo. En un sistema codificado en binario éste es el bit que representa el valor más pequeño o de menos peso. Su valor es el voltaje de escala total (o corriente) dividido por 2^n , donde n es la resolución del convertidor.

Códigos perdidos: Cuando un incremento positivo o negativo en el voltaje de entrada provoca que el convertidor incremente o decremente su salida numérica en más de 1 LSB se dice que el convertidor exhibe "códigos perdidos". Si hay códigos perdidos, hay un valor numérico en la salida del convertidor que no puede ser alcanzado por ningún valor de entrada.

Monotonía: Una función monótonica tiene una pendiente cuyo signo no cambia. Un convertidor D/A monótonico tiene una salida que cambia en la misma dirección (o se mantiene constante) por cada incremento en el código de entrada. Lo mismo es cierto para códigos que se decrementan.

MSB (Most Significant Bit): Bit más significativo. En un sistema codificado en binario éste es el bit que representa el valor o peso más elevado. Su valor es la mitad de la escala total.

Convertidor D/A multiplicativo: En cierta forma, todos los convertidores D/A son multiplicativos, puesto que el voltaje de salida (o corriente) es igual a voltaje de referencia veces una constante determinada por el código digital de entrada

dividido por 2^n (n es el número de bits de resolución). En un convertidor D/A multiplicativo de dos cuadrantes el voltaje de referencia o el código digital de entrada puede cambiar la polaridad del voltaje de salida. Si tanto el voltaje de referencia como el código digital cambian la polaridad del código de salida, entonces existe multiplicación de cuatro cuadrantes.

Error de offset o ajuste (error de cero): En un convertidor D/A, éste es el voltaje de salida que existe cuando el código digital de entrada es puesto para entregar una salida ideal de cero volts. En el caso de un convertidor A/D, es la diferencia entre el voltaje ideal de entrada ($1/2$ LSB) y el verdadero voltaje de entrada que es necesario para hacer la transición de cero a 1 LSB. Todos los códigos digitales en la curva de transferencia están desajustados en la misma magnitud. Muchos convertidores permiten el ajuste a cero con un potenciómetro externo. Este tipo de error se expresa usualmente en LSBs.

Rechazo al suministro de alimentación - Power supply rejection (Power supply sensitivity): La sensibilidad de un convertidor a los cambios en el suministro del voltaje de cd de la fuente de alimentación.

Error de cuantización: El error inherente en todos los convertidores A/D. Puesto que incluso un convertidor "ideal" tiene una resolución finita, cualquier voltaje analógico que se ubique entre dos códigos de salida adyacentes resultará en un código de salida que sea inexacto en hasta $1/2$ LSB.

Operación radiométrica: Muchas aplicaciones A/D requieren una estable y exacta referencia de voltaje contra la cual el voltaje de entrada es comparado. Esta aproximación resulta en una **conversión absoluta**. Algunas aplicaciones, sin embargo, usan transductores u otras fuentes de señales cuyos voltajes de salida son proporcionales a alguna

referencia externa. En estas aplicaciones **radiométricas**, la referencia para la fuente de la señal debe ser conectada a la entrada de referencia del convertidor. Así, cualquier variación en el voltaje de referencia de la fuente también cambiará el voltaje de referencia del convertidor y producirá una conversión exacta.

Resolución:

El más pequeño incremento analógico correspondiente a un cambio de 1 LSB de código del convertidor. Para convertidores, la resolución es normalmente expresada en bits, donde el número de niveles analógicos es igual a 2^n . Como un ejemplo, un convertidor de 12 bits divide la señal analógica en $2^{12}=4096$ niveles discretos de voltaje (o corriente).

Tiempo de asentamiento: El tiempo desde un cambio en el código de entrada hasta que la señal de salida en un convertidor D/A se mantiene en $\pm 1/2$ LSB (o alguna otra tolerancia especificada) del valor final.

Apéndice B

Definición de términos para osciloscopios.

Las siguientes definiciones se refieren a los términos más comunes al hablar sobre osciloscopios analógicos y digitales, así como de los botones y controles que se encuentran en ellos.

Attenuator.	Atenuador. Reduce la amplitud de una onda sin causar distorsión en la entrada.
Beam switch.	Conmutador de haz. Conecta alternativamente el haz de electrones a una u otra de las entradas en la unidad de dos canales.
Brill (intens).	Intensidad, control de brillo.
Calibrator.	Calibrador. Genera una señal cuadrada con la amplitud y frecuencia dada.
CRT.	Tubo de rayos catódicos.
Chopped alternate.	Traceador. Para la interrupción automática y a intervalos regulares del haz de electrones en un osciloscopio de dos canales.
Delay line.	Línea de retardo. Demora la señal hasta que ha arrancado el barrido.
DSO.	Digital Storage Oscilloscope. Osciloscopio de almacenamiento digital.
Ext (ernal).	Disparador externo. La señal de disparo es aplicada a la entrada de disparo del osciloscopio.
Ext trigg in.	Entrada de disparo exterior.
Focus.	Enfoque.
Glitch.	Pulsos de ruido producido por operación marginal de computas. Pueden ocurrir también por transitorios en

	la línea de corriente de alimentación. Algunas veces son inducidos por fuertes campos externos, como en instalaciones industriales.
Grat.	Retícula, iluminación de escala.
Int (ernal).	Disparador interno. La señal de disparo se obtiene del amplificador Y.
Jitter.	Vibración en la visualización de la señal. En osciloscopios analógicos puede producir que el haz que representa a la señal se vea más grueso, mientras que en osciloscopios digitales provoca que la señal aparezca con discontinuidades.
Level.	Nivel (al cual comienza el barrido).
Output amplifier.	Amplificador final. Alimenta al CRT con tensiones suficientemente altas para la deflexión del haz de electrones.
Persistencia.	Duración del brillo de la imagen en el CRT. La persistencia corta dura microsegundos, la media dura milisegundos y la larga dura segundos. En osciloscopios digitales existe también la persistencia infinita, la cual no tiene un límite de duración.
Plug in unit.	Unidad enchufable (por ej., amplificador X o base de tiempos).
Position (shift).	Posición (para calibrar la posición horizontal y vertical del rayo).
Power supply.	Dispositivo de conexión a la red. Convierte la tensión de la red en tensiones continuas que se necesitan para alimentar el amplificador y el CRT.
Power.	Interruptor de corriente.
Probe.	Sonda. Cable con deflector de tensiones, disminuye las interferencias y cargas capacitivas del objeto que será medido.
Sweep.	Barrido. Desplazamiento del haz de electrones sobre la pantalla; normalmente el desplazamiento suele ser rectilíneo y con velocidad uniforme.

TimeBase.	Base de tiempos, generador de barrido. Genera la tensión de barrido.
Trig select.	Selector de disparo. Se usa para que el barrido vaya al ritmo de la señal deseada.
Trig source.	Selector de la fuente del disparo.
X amplifier.	Amplificador para deflexión X.

Apéndice C

Definición de términos para la IBM PC

286.	Microprocesador Intel 80286.
386, 386DX.	Microprocesador Intel 80386 o 80386DX.
386SX.	Microprocesador Intel 80386SX.
486, 486DX.	Microprocesador Intel 80486 o 80486DX.
486SX.	Microprocesador Intel 80486SX.
8086.	Microprocesador Intel de 16 bits, con una velocidad de 5 Mhz. Apareció en junio de 1978.
8087.	Coprocesador matemático Intel que opera con el microprocesador 8086/8088.
8088.	Microprocesador Intel de 16 bits, con una velocidad de 5 Mhz. Apareció en junio de 1979. A diferencia del 8086, tiene un bus de datos externo de 8 bits.
8237.	Controlador de DMA usado en la computadora personal.
8253-5.	Temporizador de intervalos programable usado en la tarjeta madre de la IBM PC.
8255.	Interfaz a Periféricos Programable (PPI) usada en la IBM PC.
8259.	Controlador Programable de Interrupciones (PIC) usado en la IBM PC.
80286.	Microprocesador Intel 80286 de 16 bits, con una velocidad de 8 Mhz. Apareció en febrero de 1982.
80287.	Coprocesador matemático Intel para operar con el microprocesador Intel 80286.
80386.	Microprocesador Intel de 32 bits, llamado también 80386DX. Tiene una velocidad de 16 Mhz. Apareció en octubre de 1985.

80386SX.	Microprocesador Intel de 32 bits con una velocidad de 16 Mhz. Apareció en junio de 1988. Tiene un bus externo de datos de 16 bits.
80387.	Coprocador matemático Intel que se usa con el microprocesador 80386.
80486.	Microprocesador Intel de 32 bits, llamado también 80486DX. Tiene una velocidad de reloj de 25 MHz. Apareció en abril de 1989.
80486SX.	Microprocesador Intel de 32 bits, con una velocidad de 20 MHz. Apareció en abril de 1991.
80486DX2.	Microprocesador Intel de 32 bits, con una velocidad de 50 Mhz. Apareció en marzo de 1992.
AMD.	American Micro Devices. Compañía fabricante de microprocesadores usados en las computadoras personales IBM y compatibles.
AT, IBM PC.	Computadora personal IBM que se introdujo en el mercado en 1984. Como cambio presentaba al nuevo microprocesador 80286 y un bus de expansión de 16 bits.
BIOS.	Basic Input Output System. Sistema Básico de Entrada y Salida. Conjunto de rutinas en ROM que controlan a los dispositivos de entrada y salida de información en la computadora personal: video, teclado, discos, impresoras y puerto de comunicaciones.
Bit.	Contracción de "binary digit". Es la más pequeña unidad de información manejada en computación.
Byte.	Conjunto de 8 bits.
Buses.	Conjunto de enlaces o líneas de transmisión de información.
CGA.	Color Graphics Adapter. El primer sistema de video en computadoras personales a colores. Manejaba 4 colores de una paleta de 16 con resolución de 320 x 200 pixeles.

Ciclo de espera.	Ciclo de reloj introducido por un dispositivo lento para que el microprocesador se detenga y espere a que éste se encuentre listo.
CPU.	Central Process Unit.
Cyrix.	Compañía fabricante de microprocesadores usados en las computadoras personales IBM y compatibles.
Dinámica, RAM.	Memoria de acceso aleatorio que requiere de refresco continuo para retener la información. Es más barata que la memoria RAM estática, pero más lenta. Se usa para la memoria principal del sistema. Se le conoce también como memoria DRAM.
DMA.	Direct Memory Access. Acceso Directo a Memoria. Técnica de transferencia de información entre memoria y puertos sin la intervención del microprocesador.
EGA.	Enhanced Graphics Adapter. Tarjeta de video que puede mostrar 16 colores de una paleta de 256 con una resolución de 640 x 350 pixeles.
EISA.	Enhanced Industry Standard Architecture. Tecnología de manejo del bus de expansión.
E/S	Entrada/Salida. Ver I/O.
Estado de espera.	Ver ciclo de espera.
Estática, RAM.	Memoria de acceso aleatorio más rápida que la dinámica, pero más cara. Se usa para caches de memoria en la computadora personal.
Hercules.	Compañía fabricante de la tarjeta de video aceptada como estándar en monitores monocromáticos. Se le llama también así a la tarjeta de video que fabrican, la cual tiene una resolución de 720 x 348 pixeles.
I/O.	Input/Output. Canal de entrada y salida de datos.
IBM PC.	Computadora IBM con el microprocesador 8088. Apareció en 1981.
IBM.	International Business Machines. Compañía fabricante de la PC original en 1981.

Intel.	Compañía fabricante de los microprocesadores usados en las computadoras personales IBM y compatibles.
Interrupción.	Suspensión temporal de las actividades que actualmente se encuentre realizando el microprocesador para atender un evento determinado.
ISA.	Industry Standard Architecture. Tecnología de manejo del bus de expansión.
Memoria.	Dispositivo de almacenamiento de información. Se refiere comúnmente a los circuitos integrados de ROM o RAM.
Memoria, Acceso Directo a. Micro Canal.	Ver DMA. Tecnología propietaria de IBM para el manejo del bus de expansión.
Monitor.	Pantalla de video donde se visualizan datos que provienen del microprocesador.
Motherboard.	Tarjeta madre. Es la tarjeta donde se aloja el microprocesador, la memoria, y las ranuras de expansión.
NMI.	Interrupción no mascarable.
NPU.	Numerical Process Unit. Coprocesador matemático. Algunos lo llaman también Unidad de Punto Flotante (FPU: Float Point Unit).
Pentium.	Microprocesador Intel sucesor al 80486. Apareció en mayo de 1993. Tiene una velocidad de operación de 60 y 66 MHz, un largo de palabra de 64 bits y tecnología superescalar que le permite ejecutar más de una instrucción por ciclo de reloj.
Pixel.	Punto direccionable en el monitor de la computadora.
POST.	Power On Self Test. Auto-prueba de encendido. Rutina en ROM que se ejecuta cada vez que se enciende la computadora; tiene como función comprobar el estado de todo el hardware base de la computadora.
PPI 8255.	Programmable Peripheral Interfase. Ver 8255.

PS/2.	Familia de computadoras IBM introducidas al mercado en 1987. Como novedad presentaban la introducción del sistema de video VGA y la integración de varios controladores en la tarjeta madre en lugar de tenerlos en tarjetas de expansión.
RAM.	Random Access Memory.
ROM.	Read Only Memory.
Slots.	Ranuras en la tarjeta madre de la PC donde se conectan tarjetas de interfase.
VGA.	Video Graphics Array. Tarjeta de video que presenta una visualización de 16 colores de una paleta de 256 con una resolución de 640 x 480 pixeles, o 256 colores con una resolución de 320 x 200 pixeles. Varios modos más se encuentran disponibles, incluso el soporte a EGA, CGA o Hercules.
WS.	Wait State. Ver Estados de Espera.
XT.	Tipo de computadora IBM PC con el microprocesador 8086/8088 y bus de expansión de 8 bits.

Apéndice D

Modos de operación de microprocesadores Intel

Modo Real

Los microprocesadores 80286, 80386 y 80486 tienen un modo de operación en el cual pretenden ser un 8088. En este modo las direcciones que un programa usa para su área de código y datos tienen una correspondencia directa con las genuinas direcciones que se escriben en el bus del sistema. Debido a esto, es llamado **modo real**.

Cuando una computadora con alguno de estos microprocesadores usa **DOS** (incluso las más poderosas 80486), se encuentra operando en modo real. Eso la convierte básicamente en una computadora 8088 muy veloz. Debido a que el 8088 puede direccionar únicamente 1 MB de memoria, un 80286, 80386 u 80486 corriendo en modo real puede también direccionar 1 MB únicamente, sin importar cuanta memoria extra se le instale. De hecho, debido a que **DOS** reserva cierta memoria para video y otras funciones, un programa corriendo en **DOS** tiene únicamente posibilidad de usar 640 KB.

Modo Protegido

Los modos nativos de direccionamiento de los microprocesadores 80286, 80386 y 80486 son llamados **modos protegidos**. Aquí, los microprocesadores ya no se comportan como un 8088, y los

programas ya no usan direccionamiento real. En su lugar, los programas hacen accesos a memoria indirectamente, a través de tablas descriptoras que son mantenidas por medio de un ambiente que ejecute en modo protegido, tal como Windows u OS/2.

El sistema es "protegido" en el sentido de que el microprocesador usa las tablas de descriptores para validar los accesos de memoria de un programa. Si bien los accesos a memoria son menos directos que en el modo real, esa indirección es invisible para los programas, y se puede ubicar mucha más memoria. Las limitaciones de los 640 KB usables y 1 MB direccionable desaparecen.

Los diferentes modos protegidos son:

- ❑ **Modo protegido de 16 bits:** fue introducido en el 80286; está disponible también en el 80386 y 80486. Como en el modo real, 16 bits determinan el tamaño máximo de un segmento: $2^{16} = 64$ KB. Al contrario que el modo real, un programa de modo protegido de 16 bits puede ubicar hipotéticamente 16,384 de tales segmentos: 1024 MB.
- ❑ **Modo protegido de 32 bits:** se introdujo con el 80386; está disponible también en el 80486. Aquí, los programas usan el total de los 32 bits de los registros del microprocesador. En este modo el número de bits también determina el tamaño máximo del segmento, el cual se incrementa ahora a $2^{32} = 4$ GB. Puesto que este es el tamaño potencial de un segmento, los programas sólo necesitan de un segmento. Esto es llamado a veces como "modelo plano".
- ❑ **Modo virtual 86 (V86):** se introdujo con el 80386; también está disponible con el 80486. El modo V86 es esencialmente un modo protegido de 1 MB. Un programa

corriendo en modo **V86** creará que está corriendo en modo real. De hecho, estará corriendo bajo el control de un programa de control **V86**.

Bibliografía

La siguiente es una lista de la referencia bibliográfica en la cual se basó la presente tesis. Se agregan, además, aquellas fuentes donde es posible encontrar información adicional relacionada con el tema.

La lista se presenta clasificada de acuerdo al tema y a su origen.

I) Libros sobre hardware

Diseño

- 1.1.- Electrónica. Teoría de circuitos.
Boylestad, Robert; Nashelsky, Louis.
Prentice Hall. 1a Edición.
México. 1987.

- 1.2.- Diseño de sistemas digitales y microprocesadores.
Hayes, John P.
Mc Graw-Hill. 1a Edición.
México. 1990.

- 1.3.- Lógica digital y diseño de computadores.
Morris Mano, M.
Prentice Hall. 1a Edición.
México. 1991

- 1.4.- Diseño con circuitos integrados TTL.
Texas Instruments Inc.
CECSA. 1a Edición.
México. 1991.

- 1.5.- Sistemas Digitales. Principios y aplicaciones.
Tocci, Ronald J.
Prentice Hall. 3a Edición.
México. 1990.

Osciloscopios e instrumentación

- 1.6.- Instrumentación digital.
AMICEE.
Limusa. 1a Edición.
Mexico. 1989
- 1.7.- Frecuencímetros y contadores.
Biblioteca de diseño electrónico.
Ediciones Universidad y Cultura. 1a Edición.
Madrid, España. 1990.
- 1.8.- Instrumentación electrónica.
Diefenderfer, A. James.
Interamericana. 2a Edición.
México. 1990.
- 1.9.- Troubleshooting with the oscilloscope.
Middleton, Robert G.
Howard W. Sams & Company. 5a Edición.
IN, USA. 1989.
- 1.10.- Manual del osciloscopio.
Ruiz Vassallo, Francisco.
CEAC. 3a Edición.
Barcelona, España. 1983.

- 1.11.- Instrumentos electrónicos de medida.

Ruiz Vassallo, Francisco.

CEAC. 3a Edición.

Barcelona, España. 1984.

- 1.12.- El osciloscopio y sus aplicaciones.

Zapata Ferrer, Angel R.

Limusa. 1a Edición.

México. 1991.

Interfase con la IBM PC

- 1.13.- Interfasing to the IBM Personal Computer.

Eggebretch, Lewis C.

Howard W. Sams & Company. 2a Edición.

IN, USA. 1991.

- 1.14.- The 8088 project book.

Grossblatt, Robert.

TAB Books. 1a Edición.

PA, USA. 1989.

- 1.15.- Handbook of software and hardware interfacing for the IBM PC.

Royer, Jeffrey P.

Prentice Hall. 1a Edición.

NJ, USA. 1987.

- 1.16.- Interfasing sensors to the IBM PC.

Tompkins, Willis; Webster, John, Editores.

Prentice Hall. 1a Edición.

NJ, USA. 1988.

II) Libros sobre software

Lenguaje C

- 2.1.- Turbo C Bible.
Barkakati, Nabajyoti.
Howard W. Sams & Company. 1a Edición.
IN, USA. 1990.
- 2.2.- Supercharging C with assembly language.
Chesley, Harry R.; Waite Mitchell.
Addison Wesley. 1a Edición.
USA. 1988.
- 2.3.- Using Turbo C.
Schildt, Herbert.
Mc Graw-Hill. 2a Edición.
CA, USA. 1989.
- 2.4.- Advanced Turbo C
Schildt, Herbert.
Mc Graw-Hill. 2a Edición
CA, USA. 1989.

Ensamblador

- 2.5.- Lenguaje ensamblador para microcomputadoras IBM.
Godfrey, J. Terry.
Prentice-Hall. 1a Edición.
México. 1991.

- 2.6.- Advanced assembly language on the IBM PC.
Holzner, Steven.
Brady. 1a Edición.
NY. USA. 1987.
- 2.7.- 8086/8088/80286 assembly language.
Scanlon, Leo J.
Brady. 2a Edición.
NY. USA. 1988.

Video y manejo de gráficas

- 2.8.- Programmer's guide to the EGA and VGA cards.
Ferraro, Richard F.
Addison-Wesley. 2a Edición.
USA. 1991.
- 2.9.- Practical image processing in C.
Lindley, Craig A.
Jonh Wiley & Sons.
NY, USA. 1991.
- 2.10.- Fractal programming in C.
Stevens, Roger T.
M&T Books. 1a edición.
CA, USA. 1989.
- 2.11.- PC & PS/2 video systems.
Wilton, Richard.
Microsoft Press. 1a Edición.
WA, USA. 1989.

Varios

- 2.12.- Writing MS-DOS device drivers.
Lai, Robert S.
Addison Wesley. 1a Edición.
USA. 1990.
- 2.13.- Microsoft Mouse programmers reference.
Microsoft Press. 2a Edición.
WA, USA. 1991.
- 2.14.- System BIOS for IBM PC/XT/AT computers.
Phoenix Technologies Ltd.
Addison-Wesley. 1a Edición.
USA. 1990.

III) Manuales

- 3.1.- Test & Measurement Catalog 1990.
Fluke - Phillips.
USA. 1990.
- 3.2.- Peripherals.
Intel.
IL, USA. 1990.
- 3.3.- Memory.
Intel.
IL, USA. 1990.

- 3.4.- Optoelectronics device data.
Motorola.
USA. 1989.
- 3.5.- CMOS Logic Databook.
National Semiconductor.
CA, USA. 1988.
- 3.6.- Data acquisition Linear Devices Databook.
National Semiconductor.
CA, USA. 1989.
- 3.7.- Linear applications handbook.
National Semiconductor.
CA, USA. 1991.
- 3.8.- General purpose linear devices.
National Semiconductor.
CA, USA. 1989.
- 3.9.- Random Access Memory Databook.
National Semiconductor.
CA, USA. 1987.
- 3.10.- Special purpose linear devices.
National Semiconductor.
CA, USA. 1989.
- 3.11.- Basic Principles of DSOs.
Using the Phillips PM 3350.
Phillips.
USA. 1988.

IV) Tesis

- 4.1.- Diseño y construcción de una interfaz de memoria para un osciloscopio, empleando el microprocesador Z-80.
Alemón Reyes, Oscar; et. al.
Facultad de Estudios Superiores Cuautitlán.
Universidad Nacional Autónoma de México, 1987.

V) Artículos

- 5.1.- "All about oscilloscopes".
Bigelow, Stephen J.
Popular electronics. Julio 1991. Vol. 8 No. 7.
USA.
- 5.2.- "Putting a new scope to work".
Ramírez, Robert W.
Radio Electronics. Enero 1991. Vol. 62 No. 1.
USA.
- 5.3.- "Inside digital oscilloscopes".
Stover, Allan C.
Radio Electronics. Mayo 1990. Vol. 61 No. 5.
USA.
- 5.4.- "Pentium power".
Feibus, Michael; Slater, Michael.
PC Magazine. Abril 27, 1993. Vol 12 No. 8.
USA.