



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA MEQ-800 DIGITAL

FALLA DE ORIGEN

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACION

P R E S E N T A N

ROSA MARIA GUADALUPE BARBA VARELA

VERONICA DIAZ LOPEZ

Director de Tesis:

M. I. Pablo Roberto Pérez Alcázar



MEXICO, D. F.

1995

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

**A MI PADRE DIOS:**

Porque desde el instante en que medio la vida,  
guía mis pasos, mis pensamientos y mi corazón.

**A MI MAMA ROSA MA VARELA:**

Porque cuando DIOS te creo, te dió  
bondad, fortaleza y espíritu de lucha; pero  
sobre todo, porque lo que soy te lo debo a ti.

**A MIS HERMANOS MARGARITA Y SALVADOR:**

Porque sin su cariño y apoyo, el camino  
recorrido hubiera sido muy difícil.

**A GERMAN MALANCHE:**

Solo una palabra... GRACIAS.

**A MAMA CARO Y MAMA TOTOY:**

Porque guiaron mis primeros pasos y desde donde  
se encuentran son mis eternas compañeras.

**A LAS FUTURAS ESPERANZAS:**

Deseando de todo corazón, que lo que  
anhelen logren verlo realizado.

**A MIS AMIGOS:**

Por ser, estar y compartir conmigo.

**A MI ALMA MATER:**

La Universidad Nacional Autónoma de México, en  
especial a la Facultad de Ingeniería, por  
forjar al ser humano completo que hoy soy.

**A MI DIRECTOR DE TESIS:**

Porque cualquiera que fuera la circunstancia en  
la que nos encontrábamos, siempre recibí de él,  
la sonrisa del amigo y el consejo del maestro.

**AL LIC. MARIO CALDERON MORENO:**

Como un homenaje a su calidad humana y en  
testimonio de gratitud y respetuosa admiración.

**ROSA MARIA BARBA VARELA**

**DEDICO ESTA TESIS**

**A MIS PADRES**

Por el cariño y el amor que me han brindado a lo largo de mi vida y porque sin ellos no hubiera llegado a este momento.

**A MIS HERMANOS**

Porque siempre me han demostrado su cariño.

**A GUILLERMO**

Porque su gran amor me alienta a emprender nuevas cosas en mi vida.

**A MIS AMIGOS**

Por todos los momentos felices compartidos.

**A MI QUERIDA UNIVERSIDAD.**

Porque dentro de sus aulas obtuve los conocimientos necesarios que me permitieron llegar a este momento.

**VERONICA DIAZ LOPEZ**



## I N D I C E

### PROLOGO

I	INTRODUCCION	.. 1
II	ANTECEDENTES	.. 7
III	DESCRIPCION DEL SISMOGRAFO MEQ-800	.. 11
IV	ANÁLISIS DEL SISTEMA DIGITAL(ADQUISIDOR)	
	IV.1 Definición del problema	.. 23
	IV.1.1 Objetivos	.. 26
	IV.2 Determinación de los requerimientos	.. 27
	IV.3 Estudio de factibilidad	.. 29
	IV.4 Diseño conceptual	.. 30
V	DISEÑO Y DESARROLLO DEL HARDWARE DEL ADQUISIDOR	
	V.1 Base de tiempo	.. 34
	V.2 Acondicionador de la señal analógica a digital	.. 35
	V.3 Detector de evento	.. 38
	V.4 Controlador basado en el microprocesador 80C88	.. 39
	V.4.1 Microprocesador	.. 39
	V.4.2 Circuito de arranque	.. 40
	V.4.3 Mapa de memoria y almacenamiento de datos(FLASH)	
	V.4.4 Puertos	.. 43
	V.4.4.1 Teclado	.. 44
	V.4.4.2 Display de cristal líquido	.. 44
	V.4.4.3 Conversor A/D	.. 45
	V.4.4.4 Puerto serie de comunicación	.. 46
	V.4.4.5 Lectura y escritura en FLASH	.. 46
	V.4.4.6 Controlador de interrupciones NMI	.. 47
	V.5 Fuente de alimentación	.. 48
VI	ANÁLISIS DEL SOFTWARE DEL SISTEMA	
	VI.1 Definición del problema	.. 53
	VI.2 Análisis del software del adquisidor	
	VI.2.1 Objetivo	.. 55
	VI.2.2 Determinación de los requerimientos	.. 55
	VI.2.2.1 Lenguaje ensamblador,herramientas y equipo elegido para la programación	.. 57

VI.2.3	Diseño conceptual	.. 62
VI.2.3.1	Manejo de errores	.. 63
VI.2.3.2	Interacción con el usuario	.. 63
VI.3	Análisis del software para la computadora	.. 63
VI.3.1	Objetivo	.. 63
VI.3.2	Determinación de los requerimientos	.. 64
VI.3.2.1	Hardware	.. 65
VI.3.2.2	Lenguaje C y compilador de programación	.. 67
VI.3.3	Diseño conceptual	.. 68
VII	DISEÑO Y DESARROLLO DEL SOFTWARE	.. 69
VII.1	Programación del adquisidor	.. 69
VII.1.1	Programa principal	.. 69
VII.1.1.1	Establecimiento de condiciones de operación	.. 70
VII.1.1.2	Manejo de activación y desactivación de las interrupciones	.. 71
VII.1.1.3	Manejo de la base de tiempo	.. 72
VII.1.1.4	Comunicación con el usuario mediante teclado y display de cristal líquido	.. 73
VII.1.1.5	Selección y transmisión de un evento hacia PC	.. 78
VII.1.1.6	Manejo de la tabla resumen	.. 79
VII.1.1.7	Manejo de preevento y postevento	.. 80
VII.1.1.8	Borrado de memoria FLASH	.. 81
VII.1.1.9	Comunicación con la computadora	
VII.1.1.9.1	Control y programación del puerto serie	.. 82
VII.1.1.9.2	Protocolo de comunicación con la computadora	.. 83
VII.1.1.9.3	Condiciones de error	.. 86
VII.1.2	Subrutinas de atención a interrupciones NMI	
VII.1.2.1	Adquisición de datos del convertor A/D.	.. 86
VII.1.2.2	Adquisición de los dígitos del reloj	.. 88
VII.2	Programación de la computadora para procesar los datos adquiridos	
VII.2.1	Archivo de cabecera diseñado y desarrollado especialmente para el programa	.. 98

VII.2.2	Rutina principal	.. 98
VII.2.3	Comunicación con el adquisidor	.. 106
VII.2.3.1	Protocolo de comunicación con el adquisidor	.. 109
VII.2.4	Manejo de la memoria RAM	.. 114
VII.2.5	Recepción de evento y grabación en memoria del mismo	.. 117
VII.2.6	Detección y corrección de errores de la información guardada en memoria	.. 118
VII.2.7	Creación del archivo de evento en formato ASCII	.. 120
VII.2.8	Graficación de evento	.. 121

## VIII PRUEBAS, IMPLEMENTACIÓN Y EVALUACIÓN DEL SISTEMA

VIII.1	Pruebas de integración	.. 125
IX	CONCLUSIONES	.. 130
X	RECONOCIMIENTOS	.. 133
XI	BIBLIOGRAFIA	.. 134

### APENDICE A

Tabla 1 y 2 de comparación del sismógrafo analógico MEQ-800 con respecto a otros sismógrafos.

### APENDICE B

Especificaciones de las memorias FLASH

---

# PROLOGO

---

El presente trabajo de tesis que lleva por nombre **SISTEMA MEQ-800 DIGITAL**, tiene como objetivo describir como se llevó a cabo el análisis, diseño y desarrollo de un sistema de adquisición, procesamiento y observación de señales sísmicas adaptado a un sismógrafo con registro analógico en papel ahumado. El sistema permite tomar la señal analógica del sismógrafo, y en paralelo, convertirla en información digital, procesándola y obteniendo un registro que se guarda en memorias de estado sólido (EEPROM tipo FLASH). Posteriormente, los datos registrados son enviados a una computadora personal, donde son graficados y elegidos para ser guardados en archivos ASCII, con la finalidad de ser procesados más finamente por programas de uso común en sismología.

El sistema implementado se divide en tres partes principales:

- 1) Sistema digital de adquisición alrededor del MEQ-800.
- 2) Software que controla al sistema digital para que adquiera datos, los almacene y además pueda ser interrogado por una computadora personal, vía puerto serial.
- 3) Software para computadora que logra leer los datos almacenados en la memoria del sistema digital, en base a un algoritmo de comunicación bidireccional que controla al puerto serie; además, realiza sobre la información el procesamiento adecuado para su graficación y almacenamiento.

Cabe hacer mención que para desarrollar el sistema propuesto, fue necesario tomar como base los recursos proporcionados por la Coordinación de Ingeniería Sísmológica, del Instituto de Ingeniería de la UNAM.

A continuación se presenta una breve sinopsis, capítulo por capítulo del sistema desarrollado para un sismógrafo MEQ-800.

**Capítulo I**

**Introducción.** Se da una breve explicación del origen del problema y se plantea la alternativa de solución, que da origen al sistema desarrollado. De la misma forma, el usuario comprenderá el alcance del sistema y la potencialidad del mismo.

**Capítulo II**

**Antecedentes.** Se presentan los diferentes trabajos que antecedieron a éste y que influyeron para plantear un sistema como el que se desarrolló.

**Capítulo III**

**Descripción del sismógrafo analógico MEQ-800.** Se describe la forma de operación y los módulos más importantes que componen al sismógrafo MEQ-800.

**Capítulo IV**

**Análisis del Sistema Digital (Adquisidor).** Se realiza un estudio del sismógrafo MEQ-800 de tal forma que se puedan establecer los puntos adecuados para ligarse al sistema digital a desarrollar. Se evalúan los requerimientos, las características, posibilidades y limitaciones que tendría el sistema digital para su realización. Obteniéndose de esta forma un panorama de factibilidad técnica, económica y operacional.

**Capítulo V**

**Diseño y desarrollo del hardware del Adquisidor.** Se explica el diseño y desarrollo de los circuitos electrónicos de las secciones de control, base de tiempo, adaptador de la señal analógica a digital, detector de evento, comunicación con el usuario y comunicación con una computadora personal.

**Capítulo VI**

**Análisis del software del sistema.** Aquí se analizan 2 puntos importantes: 1) La evaluación de los requerimientos, características, posibilidades y limitaciones del software que controlará al sistema digital. 2) Al igual que en el primer punto, se realiza una evaluación parecida del sistema de software para computadora.

**Capítulo VII**

**Diseño y desarrollo del software.** En primer lugar, se explica el diseño y desarrollo del programa elaborado en lenguaje ensamblador que permite al sistema digital establecer las condiciones necesarias para capturar y almacenar datos, obtener y guardar en una tabla la información que caracteriza a los eventos y enviar datos hacia la computadora de acuerdo al protocolo establecido de comunicación bidireccional ó por medio de la operación manual del teclado del sismógrafo digital.

En segundo lugar, se explica el diseño y desarrollo del programa realizado en lenguaje C , el cual permite al usuario interrogar al sismógrafo digital en base al protocolo establecido de comunicación bidireccional, con lo que se logra extraer la tabla de eventos y los datos del evento seleccionado.

De la misma forma, se habla del funcionamiento del programa para recibir eventos, cuando el sismógrafo digital es operado manualmente y no por medio del protocolo.

Además, se describe como el programa captura los datos y los graba en memoria, detecta y corrige los errores de la información almacenada, gráfica el evento y guarda los datos en un archivo ASCII.

#### **Capítulo VIII**

**Pruebas, implementación y evaluación del sistema.** Se describen las pruebas realizadas para activar todas y cada una de las partes que integran al sistema, las cuales permitieron verificar su correcto funcionamiento.

#### **Capítulo IX**

**Conclusiones.** En este capítulo se presentan los resultados obtenidos, una vez que se finalizaron las etapas de implementación y pruebas del sistema desarrollado.



## CAPITULO I

---

### *INTRODUCCION*

## INTRODUCCIÓN

Los sismos son perturbaciones súbitas en el interior de la tierra que dan origen a vibraciones o movimientos del suelo; la causa principal es la ruptura o fractura de las rocas en las capas exteriores de la tierra, como resultado de un proceso gradual de acumulación de energía.

Estudios de Sismología, denotan que una zona de la República Mexicana, que abarca desde las costas de Jalisco hasta la frontera con Guatemala pertenece a un segmento del llamado "Cinturón de Fuego", en donde ocurren la mayor parte de los fenómenos sísmicos de nuestro país.

Aunque la mayor parte de los sismos que ocurren en México se relacionan directamente con el movimiento de las placas tectónicas, hay temblores menos frecuentes que ocurren en los continentes hacia el interior de las placas, a los cuales se les llama sismos locales. A diferencia de los terremotos que ocurren lejos de la Ciudad de México y que son sentidos en la ciudad como mecidas oscilatorias de período largo, los sismos locales se presentan como una fuerte sacudida vertical casi instantánea, seguida por vibraciones rápidas de muy corta duración. Generalmente, estos sismos son pequeños, la mayoría menor de 4 grados en la escala Richter.

Por lo anterior, estudiar la actividad sísmica en México es necesario y de gran importancia para conocer sobre las propiedades que caracterizan a los movimientos en los diferentes tipos de suelo, y así, a partir de estos estudios decidir que acciones se deben llevar a cabo para reducir los efectos ó el impacto sobre la población en general.

Parte de los estudios que se realizan corresponden a sismicidad local y son realizadas por algunas instituciones en el país; como ejemplo, podemos mencionar que el Departamento de Sismotectónica de Comisión Federal de Electricidad (CFE), lleva a cabo estudios geofísicos con el propósito de conocer los procesos geodinámicos que tienen lugar en el entorno de futuros sistemas hidroeléctricos. Así pues, se estudia la sismicidad histórica y reciente a través de la instalación de sismógrafos. Los sismógrafos son instrumentos sumamente sensibles a los movimientos de la superficie de la tierra, que proporcionan representaciones gráficas de los mismos (sismogramas). De acuerdo a la información que proporcionan los sismogramas, se obtiene la sismicidad local, lo cual permite determinar el potencial sísmico del área para decidir si es factible construir una presa.

Con la finalidad de realizar estudios de microtemblores y sismicidad local, la Coordinación de Ingeniería Sismológica del Instituto de Ingeniería adquirió, a mediados de los años 70's, sismógrafos analógicos portátiles, del tipo MEQ-800, que permiten tener una historia continua de la actividad sísmica sobre papel ahumado. En la actualidad, estos registros se complementan con los proporcionados por equipos digitales, los cuales tienen la ventaja de proporcionar datos que son fáciles de analizar. Sin embargo, los equipos digitales que se tienen en la Coordinación tienden a estar descontinuados en breve tiempo, debido a problemas de soporte por parte de las compañías fabricantes, componentes electrónicos obsoletos y falta del mantenimiento adecuado; lo que no sucede con los equipos analógicos, como el MEQ-800, que en la actualidad se siguen utilizando debido a que se caracterizan por un bajo consumo de energía, resistencia

a la intemperie, banda ancha, sencillez en su construcción y una gran calidad de su registro analógico continuo, el cual puede variar entre 12 horas y un máximo de 16 días.

Se debe destacar, no obstante, que la principal desventaja del MEQ-800, es que el análisis del registro se realiza sobre el mismo papel ahumado, utilizando reglas para la medición, lo cual sólo permite la obtención de un número limitado de características de cada evento; además, el proceso es lento y poco preciso.

De la necesidad de evitar esta desventaja y contar con equipo sísmico digital confiable y económico, nace la idea de realizar algunas modificaciones en ellos, de tal forma que se pueda agilizar el procesamiento de sus datos, así como mejorar la precisión de los mismos. Las modificaciones planteadas involucraron el diseño y construcción de un sistema de adquisición digital, que adaptado al MEQ-800, registre simultáneamente y en paralelo con su registrador analógico, los eventos de mayor interés existentes durante el tiempo de operación.

Para dirigir las actividades del sistema digital, entre las cuales se tiene la adquisición, control, procesamiento, almacenamiento y salida de la información, fue necesario diseñar e implementar un programa en ensamblador.

El sistema desarrollado, adquiere continuamente la información analógica que se obtiene del único canal del sismógrafo y la transforma a información digital en un formato

## INTRODUCCIÓN

de palabras de 12 bits con datos adicionales de tiempo; mediante un circuito de disparo, se analiza la amplitud de la señal analógica y se decide si los datos corresponden o no a un evento. La información adquirida segundos antes del disparo se considera parte del registro digital, y se le denomina preevento. Además, una vez que se verificó que el disparo ya no está activado, se procede a grabar segundos de información posteriores al evento; esta información se denomina postevento. El registro digital se almacena en memorias EEPROM tipo FLASH; de donde se puede enviar hacia una PC por medio del puerto serie.

Desde el inicio del proyecto se decidió buscar la forma adecuada de recuperar un evento guardado en la memoria FLASH del adquisidor para su posterior procesamiento.

Una vez analizado el problema, se empezó a proponer una serie de alternativas para darle solución. Una de las alternativas planteadas fue la de investigar si existía en el mercado algún sistema para PC que pudiera, a través del puerto serie, comunicarse con el adquisidor digital para leer los eventos almacenados en su memoria y así, tener la posibilidad de guardarlos en archivos que pudieran ser procesados después.

De ahí que, se analizaron los programas que son utilizados por el personal de la Coordinación de Ingeniería Sismológica para leer datos de los sismógrafos digitales tipo DR-100 y EDA. Estos programas leen datos vía puerto serial, pero están diseñados para capturar y decodificar 3 canales de información, en los formatos que envían estos equipos, por lo que, no se adaptaron a nuestras necesidades; sin embargo, se encontró que en el mercado hay

programas, como DEGTRA5, PITSA y PLOTXY (versiones para PC y estación de trabajo), que leen datos guardados en archivos ASCII, los presentan en forma gráfica y cuentan con varias funciones para procesar finamente los datos.

Con la información anterior, surgió la idea de diseñar un programa para PC que pudiera comunicarse, por medio del puerto serie, con el adquisidor digital y que capturara la información del único canal, en el formato en que es enviada, para almacenarla en archivos ASCII. Sin embargo, el lograr sólo lo anterior significaba que para saber la importancia de la información obtenida del adquisidor digital, las personas que manipularán al equipo tendrían que manejar al mismo tiempo los programas de procesamiento de datos mencionados en el párrafo anterior (en sus versiones para PC) ó en su caso esperarse hasta llegar al laboratorio.

Así pues, tomando en cuenta que es muy importante evitar una pérdida inútil de tiempo, se decidió agregar al programa una rutina de graficación que permitiera observar el evento en el lugar dónde se esté utilizando el equipo.

El programa elaborado en lenguaje C, le da al usuario la posibilidad de que con su PC portátil, en el mismo lugar donde se está monitoreando el terreno, reciba vía puerto serie los eventos que tiene el adquisidor ó si se decide, interrogar al sismógrafo mediante el protocolo de comunicación implementado, con lo cual se pueden recibir eventos sin manipular manualmente al equipo. Los datos pertenecientes a un evento se capturan y guardan en memoria RAM; después un algoritmo detecta si existe

error en la información y si es posible los corrige; finalmente, se puede observar la gráfica correspondiente. Esto último se hace con el objeto de que sólo la información de interés se almacene en archivos ASCII. Además, el sistema fue elaborado de tal forma que es amigable y sencillo de usar, con la característica de que una persona con los conocimientos básicos de cómputo lo puede operar.

Como se puede observar, el **SISTEMA MEQ-800 DIGITAL** involucra al adquirente digital, su software de control y el software de procesamiento de datos para PC.

Dentro de las ventajas que nos proporciona este sistema se mencionan las siguientes:

- 1) Como este sistema de adquisición utiliza para el almacenamiento de datos, memorias semiconductoras EEPROM tipo FLASH, el usuario tendrá la confianza de que no se perderá la información aún cuando el equipo no reciba soporte de batería. Además, la información queda lista para ser enviada a una computadora personal.
- 2) El rápido procesamiento de los datos por medio de una computadora, puede acelerar todo un programa experimental, de tal manera que no se desperdicien recursos humanos, dinero ni equipo, por el hecho de esperar excesivos periodos de tiempo entre la recopilación de los datos, el procesamiento y el análisis.
- 3) Se tendrá una dependencia menor en instrumentación sísmica pues el sistema MEQ-800 es confiable y económico.

## CAPITULO II

---

### *ANTECEDENTES*



## ANTECEDENTES

Debido a la localización geográfica de nuestro país, es indispensable establecer una sólida infraestructura instrumental para el registro de temblores, con el objeto de analizar la respuesta de los diferentes tipos de suelos y estructuras a un movimiento sísmico.

Es por ello que, desde la década de los 70's, se ha presentado la necesidad de adquirir equipos que proporcionen datos confiables a un costo bajo.

En un principio se adquirieron sismógrafos analógicos del tipo MEQ-800, posteriormente se compraron sismógrafos digitales como son: el DCA-302, el EDA y el DR-100.

Desafortunadamente, al paso del tiempo, las personas que trabajan en la sección de Ingeniería Sismológica han tenido problemas para mantener en condiciones de operación los equipos digitales, ya que no existe el soporte por parte de las compañías fabricantes, algunos componentes están obsoletos y, por lo tanto, el mantenimiento no es el adecuado. Además, se vuelve cada vez más difícil adquirir nuevo equipo, debido al alto costo que presentan. Para eliminar estos problemas, surge la idea de ir actualizando los diversos equipos con los que se cuenta.

En esta dirección, en la Coordinación de Ingeniería Sismológica del Instituto de Ingeniería se han realizado varios trabajos de actualización previa y que han servido de antecedente al descrito en este trabajo.

#### ANTECEDENTES

El **primer prototipo** desarrollado fue un Acelerógrafo Digital Portátil, con las siguientes funciones:

- Supervisión y/o evaluación del funcionamiento global del registrador mediante la adquisición y presentación de la información sin necesidad de interrumpir el funcionamiento del registrador.

- Asignación de valores iniciales a las variables que el microprocesador del registrador manejaba.

Este prototipo, trabaja en base a un microprocesador 6502, un conversor A/D; Memorias RAM y EPROM y dispositivos periféricos tales como Teclado y Display con el fin de tener un manejo y visualización adecuado de la información. Así mismo, también cuenta con una Base de Tiempo, tal y como lo tiene el Sistema MEQ-800 y un Módulo de Comunicación con el usuario.

El prototipo mencionado, a diferencia del Sistema desarrollado que se describe en este trabajo, cuenta con 3 canales de información con 100 muestras por cada canal, y el almacenamiento de la información se realiza en casete.

El **segundo prototipo** desarrollado es un Sistema de Almacenamiento de Datos Sísmicos en Memorias de Semiconductor, construido con memorias FLASH del tipo 48F512 (64 Kbytes de capacidad de almacenamiento). Este prototipo es un Acelerógrafo Digital que tiene 3 canales de información con 100 muestras por segundo para cada uno de ellos.

En este prototipo, todo el desarrollo gira en torno a un microprocesador 8086, el cual tiene dentro de sus funciones el control del bloque de decodificación de información, un

#### ANTECEDENTES

bloque que controla las interrupciones y los voltajes de programación, manejo de memorias como son: EPROM, RAM y FLASH y un bloque de comunicación con el usuario via teclado, display y un último bloque que corresponde a la comunicación con la PC.

El objetivo de este prototipo, era el de sustituir el Sistema de Almacenamiento en casete del Acelerógrafo Digital, por un sistema paralelo a él, que desarrollara las mismas funciones en cuanto a tiempo y capacidad de grabación del casete, con la ventaja de que los cambios climáticos a los que está expuesto el sistema no afectan de manera determinante al prototipo desarrollado.

El **tercer prototipo** realizado es el antecedente más cercano al sistema diseñado para el sismógrafo MEQ-800, y corresponde a un Sistema de Almacenamiento e Interfaz entre el sismógrafo DR-100 y una computadora. Este equipo cuenta con 3 canales de información con 100 muestras por segundo para cada uno de ellos. El objetivo de este prototipo era el de actualizar al sismógrafo DR-100, proporcionándole un mayor campo de acción.

Este prototipo sienta las bases para el desarrollo y la actualización del Sismógrafo MEQ-800, ya que ambos cuentan con los bloques de: Almacenamiento de datos, comunicación con el operador, comunicación con la computadora y control y procesamiento de datos.

La diferencia fundamental entre el sistema digital del DR-100 y el sistema digital del MEQ-800, es que el primero adquiere 300 muestras por segundo provientes de 3 canales de

#### ANTECEDENTES

información; y el segundo únicamente cuenta con un canal que, de acuerdo a la frecuencia de muestreo seleccionada en el sistema, puede adquirir 60 ó 120 muestras por segundo.

Finalmente, diremos que, con el objeto de reducir la dependencia tecnológica en instrumentación sísmica y contar con equipo sísmico digital confiable y económico, se llevó a cabo la idea de actualizar los sismógrafos analógicos de registro en papel ahumado utilizando para ello algunas ideas de los prototipos que lo antecedieron.

## **CAPITULO III**

---

### ***DESCRIPCION DEL SISMOGRAFO ANALOGICO MEQ-800***

#### DESCRIPCION DEL SISMOGRAFO ANALOGICO MEQ-800

El sismógrafo MEQ-800, de la compañía SPRENGNETHET INSTRUMENT CO. INC, es un equipo portátil con un sistema de grabación sísmica de rango ancho. El sismógrafo fue diseñado, básicamente, para sitios en los que se localicen temblores pequeños o con moderados altos niveles de actividad. Sin embargo, puede ser usado para monitorear fuertes disturbios, empleando un atenuador externo con el objeto de evitar la saturación.

El sismógrafo MEQ-800 se caracteriza por su bajo consumo, .4 watts, por su resistencia a la intemperie y la gran calidad de su registro continuo, el cual puede variar entre 12 hrs y un máximo de 15 días(dependiendo de la rotación del motor elegida).

#### **Componentes.**

El sistema MEQ-800 está constituido por los siguientes componentes:

- Batería GC1215(paquete de 4 baterías).
- Amplificador AS1110 y un panel de control principal.
- Sistema de tiempo digital de precisión.
- Registrador R-6040.
- Sismómetro modelo L-4C o modelo S-7000.

En la figura 3.1, se presenta la forma como se estructuran las partes mencionadas anteriormente; se puede observar, que los dispositivos externos que se conectan al sismógrafo, están dibujados como bloques fuera del recuadro.

**Funcionamiento.**

Si se quiere observar el movimiento del suelo, que se produce al ocurrir un temblor, se debe estar en un punto fijo fuera de la tierra para no sufrir ese movimiento y poder detectarlo; esto obviamente es imposible. Sin embargo, un sismómetro puede medir este movimiento relativo y es por lo tanto el componente básico para la operación del sismógrafo.

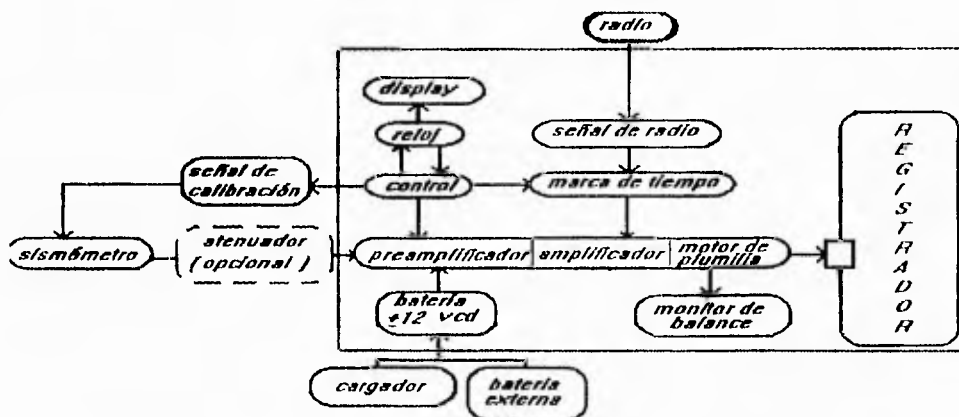


FIGURA 3.1. Diagrama de bloques del sismógrafo MEQ-800.

DESCRIPCION DEL SISMOGRAFO ANALOGICO MEQ-800

El mecanismo interno del sismómetro consiste de una masa suspendida de un resorte que está atado a un soporte acoplado al suelo (figura 2). Cuando el suelo se mueve por el paso de las ondas sísmicas, también se mueve el soporte. Sin embargo, la inercia hace que ésta permanezca un instante en el mismo sitio. Posteriormente, cuando la masa sale del reposo oscila. Este movimiento de la masa no refleja el movimiento del suelo, por lo tanto, el sismómetro debe contar con un mecanismo de amortiguamiento para volver a la masa a su sitio original.

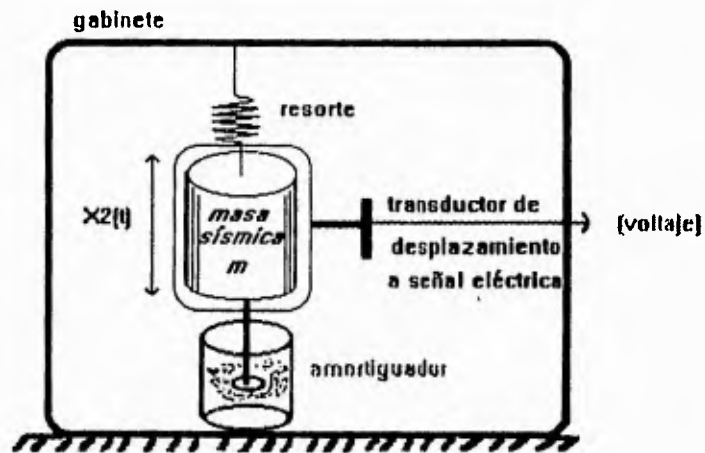


FIGURA 3.2. Diagrama esquemático de un sismómetro típico.



#### DESCRIPCION DEL SISMOGRAFO ANALOGICO MEQ-800

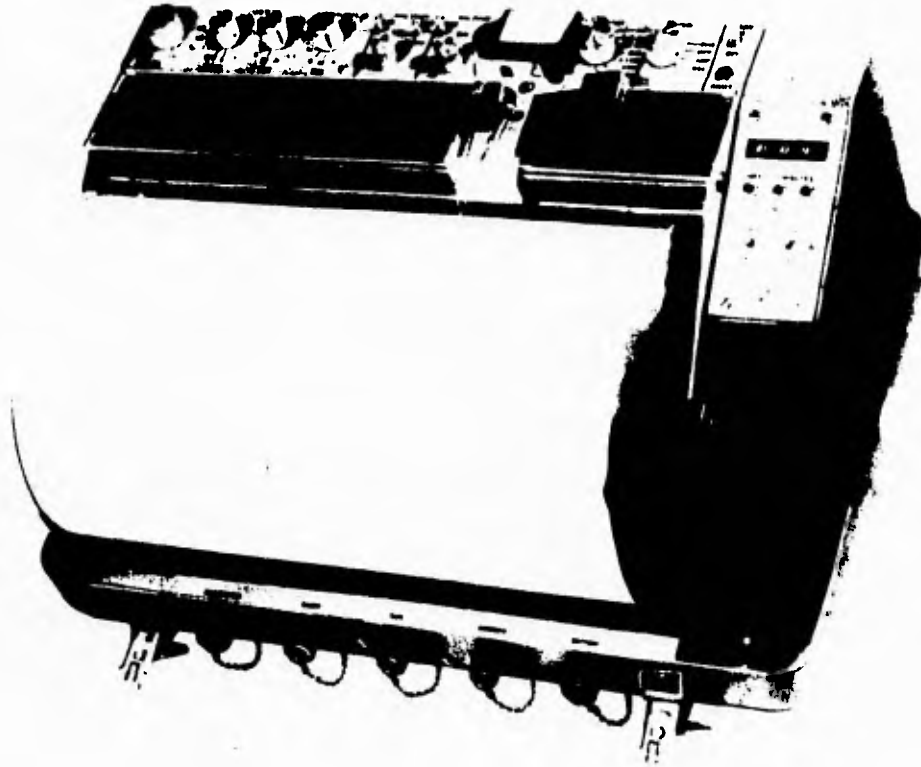
En la figura 3.2 se representa el amortiguamiento como una lámina sumergida en un líquido (comúnmente aceite).

El mecanismo hasta aquí descrito, detecta el desplazamiento vertical del movimiento del suelo. Este desplazamiento, a su vez, se convierte en señal eléctrica (voltaje) por medio de un transductor (figura 3.2).

Normalmente, el desplazamiento relativo de la masa con respecto al suelo es tan pequeño que es necesario amplificarlo para poder medirlo. En la figura 1, se puede ver que, la señal de salida del sismómetro es aplicada a la **sección del preamplificador**, el cual sirve de medio de acoplamiento con el **amplificador principal AS110**. El amplificador AS110 tiene ganancias desde 60 a 120 decibeles y posee filtros paso altas y paso bajas ajustables que pueden dar un ancho de banda de .2 a 70 Hz.

Por último, la señal de la etapa amplificadora va hacia el **registrador** (figura 3.3), el cual consiste básicamente de un tambor y una plumilla graficadora. En base a dispositivos electromecánicos se logra que el tambor rote y la plumilla se traslade en forma helicoidal alrededor del tambor.

El gráfico (**sismograma**) se puede trazar entintando sobre papel blanco o también sobre papel blanco, pero recubierto de una tenue capa de carbón, con la finalidad de que la plumilla al pasar deje un surco blanco sobre fondo negro.



**FIGURA 3.3. Registrador del MEQ-800.**

Por otra parte, el sismógrafo cuenta con un control de tiempo preciso que se inscribe sobre los sismogramas; las **marcas de tiempo** provienen de un cronómetro de precisión digital, con una estabilidad  $\pm 3 \times 10^{-7}$ /día. El tiempo es muy importante para identificar exactamente el tiempo de llegada de las diferentes ondas sísmicas a una estación sísmológica.

DESCRIPCION DEL SISMOGRAFO ANALOGICO MEQ-800

En la figura 3.4 se observa un sismograma en papel ahumado que representa el desplazamiento vertical proporcional al movimiento. Finalmente, diremos que, para obtener algunas características de los eventos es necesario medir con reglas sobre la superficie del sismograma.

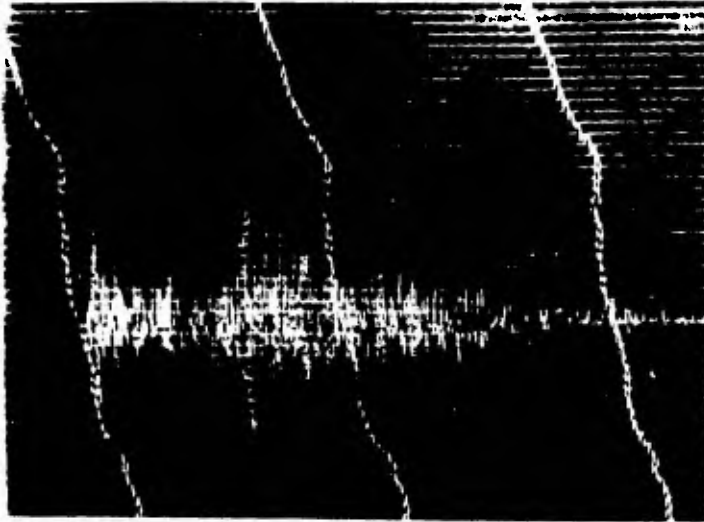


FIGURA 3.4. Sismograma en papel ahumado.

**Especificaciones principales del sismógrafo MEQ-800**

GENERALES	
Tamaño	46 cm x 32 cm x 24 cm
Peso	14.5 kg. (excluyendo al sismometro)

## DESCRIPCION DEL SISMOGRAFO ANALOGICO MEQ-800

Voltaje máximo de sensibilidad	$3 \times 10^5$ cm/volt
Voltaje mínimo de sensibilidad	3 cm/volt con un atenuador de entrada, $3 \times 10^2$ cm/volt sin atenuador de entrada
Requerimientos de potencia	baterías internas: +12VDC y -12VDC
Velocidad	$5.6 \times 10^5$ cm/cm/seg, máximo
Desplazamiento	$35.0 \times 10^6$ cm/cm máximo a 10 Hz
Rango de temperatura	0° a 50° C (32° a 122° F)
<b>AMPLIFICADOR</b>	
Ganancia de voltaje	$1 \times 10^3$ mínimo
Control de ganancia	60 DB a 120 DB en incrementos de 6 DB
Respuesta a la frecuencia sin filtrado	.3 Hz, 5Hz, ó 10 Hz (en bajas) y 5 Hz, 10 Hz, 30 Hz ó 70 Hz (en altas)
Impedancia de entrada	100 K
<b>TAMBOR REGISTRADOR</b>	

## DESCRIPCION DEL SISMOGRAFO ANALOGICO MEQ-800

Tamaño	343 mm x 600 mm
Velocidad de grabación	60 mm por minuto ó 120 mm por minuto
Espacio entre líneas	1/2 mm, 1 mm ó 2 mm
Tipo de motor	inductivo sincronizado a 60 Hz
Precisión del motor	igual a la frecuencia base del oscilador del sistema de tiempo
Velocidad del motor que rota al tambor	1 RPM
Velocidad del motor que realiza la traslación de la plumilla	1/6 RPM
Largo del estilógrafo de la plumilla	12.5 cm
Motor de la pluma, responde a una frecuencia en DC de:	75 Hz
<b>SISTEMA DE TIEMPO</b>	
Display	en él se puede visualizar: segundos, minutos y horas

## DESCRIPCION DEL SISMOGRAFO ANALOGICO MEQ-800

Marcas de tiempo	0.02 de segundo por cada segundo, 1 pulso de segundo por cada minuto, 2 pulsos de segundo por cada hora, 10 pulsos de segundo por cada 12 horas
Para sincronizar el tiempo	se puede avanzar o retrasar 16 ms por cada segundo (por medio de un "switch")
Estabilidad de reloj	$\pm 3 \times 10^{-7}$ / día

**Controles y partes de identificación. (observar figura 3.5A)**

1. Panel principal de control.
2. Ganancia en decibeles.
3. Filtro paso bajas(Hz).
4. Filtro paso altas(Hz).
5. Deflexión máxima.
6. Controles de marca de tiempo.
7. Control del pulso de calibración.
8. Carátula del medidor de voltaje y corriente.
9. Perilla para seleccionar una función en el medidor.
10. Perilla que selecciona 4 funciones principales del sistema: encendido, cargador, amplificador y comienzo del movimiento del tambor para registrar.
11. Encendido y apagado del reloj.
12. Batería.
13. Cargador.
14. Señal analógica de salida.

DESCRIPCION DEL SISNOGRAFO ANALOGICO MEQ-800

15. Conector para la entrada de radio.
16. Conector para el sismómetro.
17. Placa de los conectores.
18. Brazo que controla y sostiene el tambor registrador.
19. Seguro transversal trasero.
20. Motor que rota al tambor.
21. Dispositivo de fricción.
22. Brazo del estilógrafo.
23. Botón que libera el brazo del estilógrafo.
24. Eje del motor de la pluma.
25. Display del cronómetro.
26. Control para sincronizar el reloj.  
(adelanta o retrasa 16ms).
27. Control para visualizar los dígitos de tiempo  
en el display.
29. Motor de traslación.
30. Poleas para la traslación.
31. Salida de la señal de reloj.

**Para los siguientes puntos observar la figura 3.5B.**

34. Fusibles de la batería interna.
35. Bloque de fusibles.
36. Conector del reloj.
37. Conector para el motor de traslación.
38. Conector para el motor de rotación.
39. Conector para el motor de la pluma.
40. Baterías "A".
41. Baterías "B"

DESCRIPCION DEL SISMOGRATO ANALOGICO MEQ-800

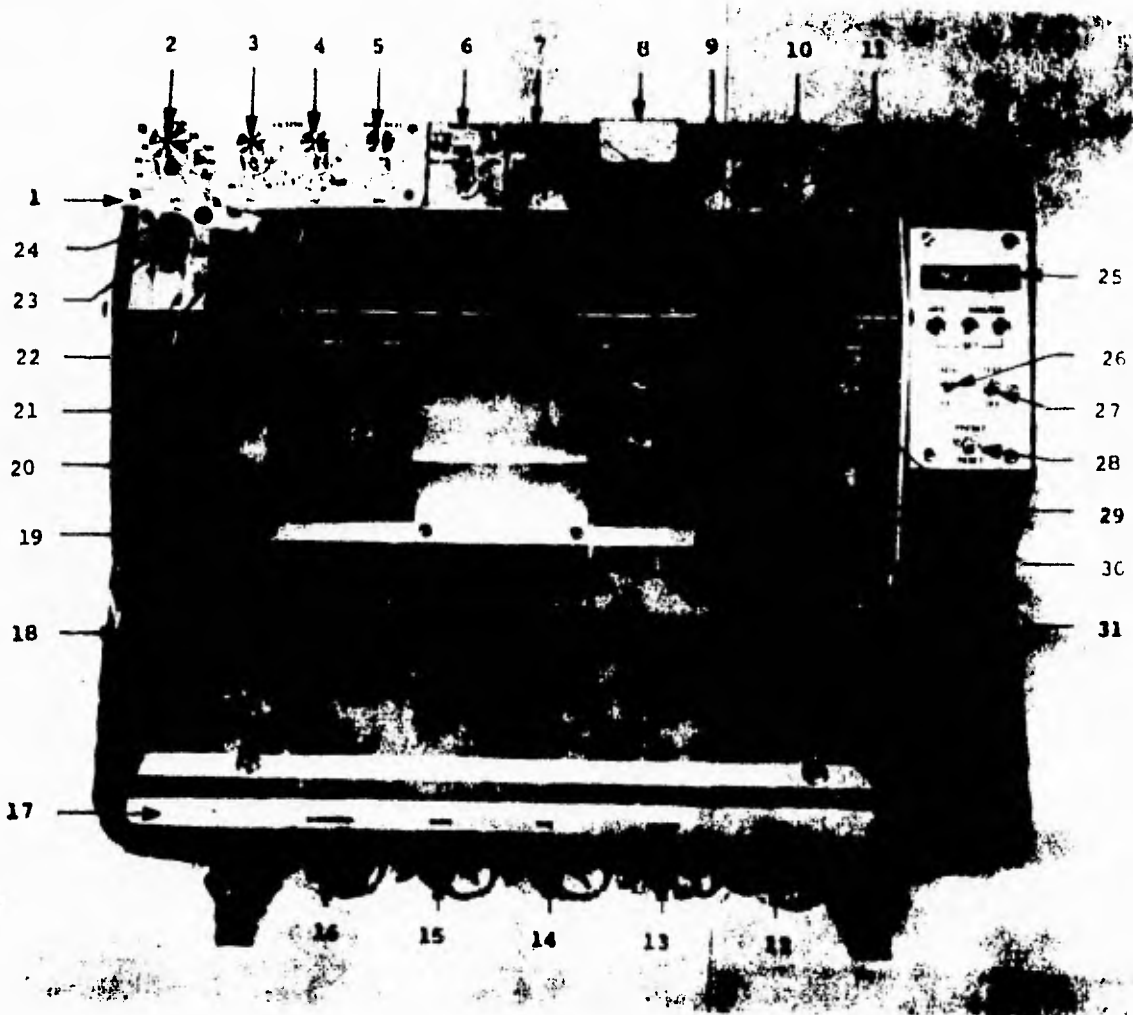


FIGURA 3.5A. Visualización de controles y partes del MEQ-800.



DESCRIPCION DEL SISMOGRATO ANALOGICO MEQ-800

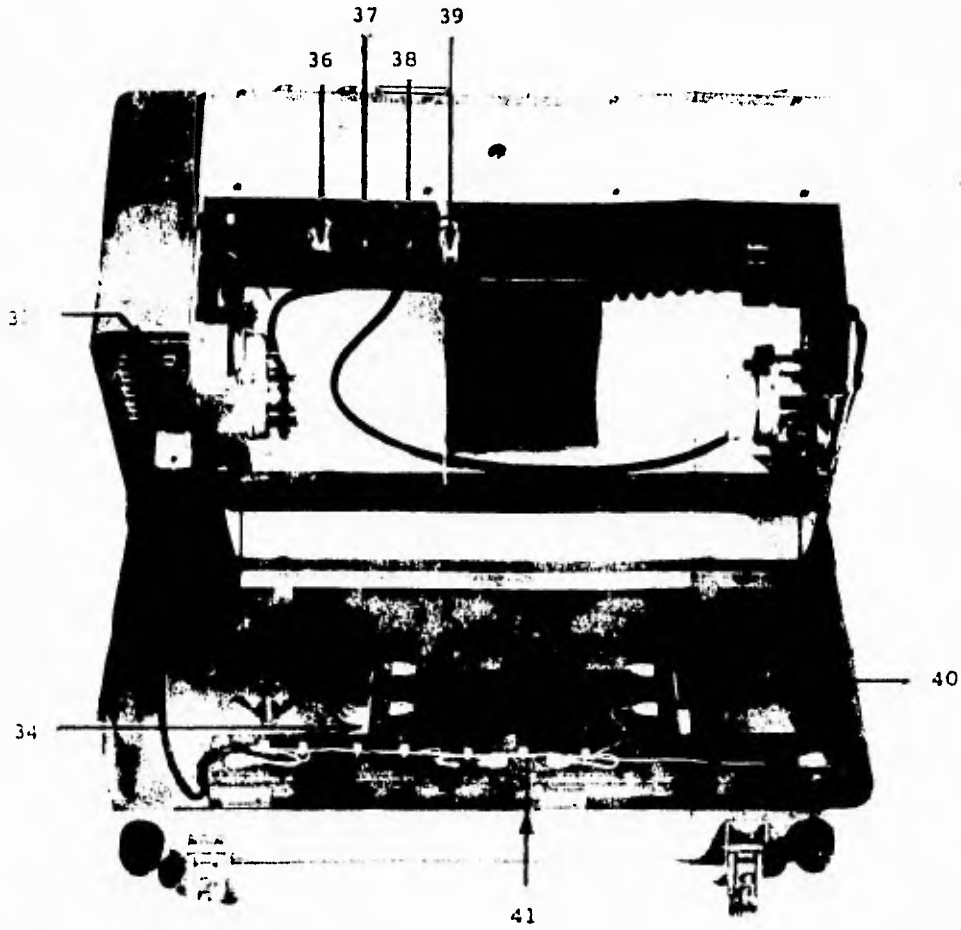


FIGURA 3.5B. Visualización de controles y partes del MEQ-800.

## ***CAPITULO IV***

---

### ***ANALISIS DEL SISTEMA DIGITAL (ADQUISIDOR)***

El ciclo de vida de un sistema empieza en un grupo de planeación, al principio como una idea conceptual muy general, que posteriormente toma forma en un grupo de diseño. Si se considera factible el proyecto, desde distintos puntos de vista se desarrolla e implementa.

El costo económico y la complejidad tecnológica de los sistemas que actualmente se desarrollan es tan grande, que es necesario planear con todo cuidado estos proyectos, analizar diversas alternativas, tomando en cuenta todos los aspectos de un proyecto.

Se requiere, entonces, una secuencia lógica de pasos detallados de análisis, para resolver estos problemas.

En este capítulo se presentan las etapas que se siguieron en la fase de análisis; estas etapas se mencionan a continuación:

- Definición del problema.
- Determinación de los requerimientos.
- Estudio de factibilidad.
- Diseño conceptual.

#### **IV.1 DEFINICIÓN DEL PROBLEMA**

Para estudiar la sismicidad local de una zona es necesario instalar sismógrafos que proporcionen observaciones instrumentales, con las cuales se va conformando un catálogo de sismos. El catálogo de una región muestra algunas veces un

patrón recurrente de eventos importantes, que permite estimar el potencial sísmico de la zona.

Como se había mencionado, en la década de los 70's la Coordinación de Ingeniería Sismológica adquirió sismógrafos analógicos tipo MEQ-800 de registro en papel ahumado, con la finalidad de realizar estudios de sismicidad local.

Se decidió comprar este tipo de sismógrafos porque están diseñados básicamente para el estudio de sismos locales, pero también, debido a que en comparación con los sismógrafos analógicos de ese tiempo resultaba en muchos aspectos con una calidad superior. De hecho, hasta la fecha, la compañía Sprengnether sigue produciendo estos equipos sin cambios. En el apéndice A se presentan tablas en las cuales se comparan las características del MEQ-800 con respecto a otros equipos.

Conforme avanzó la tecnología, se adquirieron sismógrafos digitales tipo DR-100, DCA-302 y EDA. Pero como estos equipos están contruidos en base a circuitos integrados que en la actualidad son obsoletos y ya ni la compañía proporciona, el mantenimiento resulta complicado.

Sin embargo, los MEQ-800 se siguen utilizando debido a la confiabilidad en los registros que proporcionan y a que la sencillez en su construcción permite darles mantenimiento fácilmente. La desventaja principal es que a diferencia de los equipos digitales, se desperdicia mucho tiempo y recursos humanos, en el proceso que va de la recopilación de eventos hasta el momento en el cual se analizan los datos; además,

pocos parámetros característicos de las señales registradas pueden ser obtenidos.

Para dar una idea, de todo el trabajo que se realiza para obtener un evento con el MEQ-800, se hará una breve descripción del proceso desde el momento en que se coloca la hoja para sismograma en el tambor registrador, hasta el punto en el cual se obtiene la localización del evento.

Como se sabe, el sismógrafo MEQ-800 tiene un tambor registrador el cual se cubre con hojas blancas para sismogramas. Una vez colocada la hoja, se procede a ahumarla con la ayuda de un ahumador de petróleo, con el fin de que tengan un recubrimiento similar al del papel carbón, para que con ayuda de la plumilla con la que cuenta el equipo, se deje un surco sobre el papel, y de esta forma se obtengan los registros de los eventos. Realizado lo anterior, se procede a instalar el equipo en el campo para que registren los eventos sísmicos que se presentan durante el período de tiempo en el cual permanece instalado el equipo (que puede variar entre 12 horas y 15 días). Al término de este período se procede a retirar el tambor y dejar instalado uno nuevo.

Una vez que se retira el tambor del equipo, hay que tener cuidado de que no se altere o borre la información de las hojas, ya que, como es un registro que viene en papel ahumado, puede sufrir accidentes al rozar el tambor con cualquier superficie, eliminándose los eventos que contiene.

Al llegar los registros analógicos al laboratorio, se

procede a retirar la hoja del registro del tambor, teniendo cuidado de no alterar la información que contiene; posteriormente, esta hoja se pasa por un líquido fijador y se deja secando, con el propósito de que al cristalizar el sismograma se pueda manipular correctamente.

La medición de la amplitud de un evento se realiza mediante una regla puesta directamente sobre el sismograma; asimismo, la duración del evento se obtiene directamente de la observación de las marcas de tiempo inscritas sobre el sismograma.

Es evidente que para obtener y analizar un evento el procedimiento es largo y poco preciso. Además, se puede observar que hasta el momento en el que se fija el sismograma, se corre el riesgo de que si no se trata adecuadamente la hoja, es posible alterar o borrar la información que contiene.

Por ello, surge la necesidad de realizar un sistema paralelo al anterior que obtenga la misma información, pero de forma digital, con lo cual se logrará un rápido procesamiento de la misma.

#### **IV.1.1 Objetivo**

Desarrollar un sistema digital de adquisición, para ser adaptado a un sismógrafo MEQ-800 con registro en papel ahumado. El sistema permitirá tomar la señal analógica del sismógrafo y en paralelo, convertirla en información digital, procesándola y obteniendo un registro que se guardará en

memorias EEPROM tipo FLASH.

Además, dentro del objetivo está incluir puertos que sirvan para la comunicación directa con el usuario y con una computadora portátil.

#### **IV.2. DETERMINACIÓN DE LOS REQUERIMIENTOS.**

La determinación de los requerimientos es el estudio de un sistema para conocer como debe funcionar.

Para crear un sistema se debe trabajar conjuntamente con los usuarios para identificar los requerimientos que deben satisfacerse. En este caso, las personas de la Coordinación de Ingeniería Sismológica que han trabajado con el sismógrafo MEQ-800, nos proporcionaron conocimientos, hechos y detalles que nos ayudaron a comprender la situación.

A continuación se mencionan los requerimientos que el adquisidor digital debe satisfacer:

- a) El sistema debe tener la capacidad de adquirir datos a 2 diferentes frecuencias de muestreo: 60 muestras por segundo ó 120 muestras por segundo.
- b) Capacidad de grabación de un evento de manera continua cuya única limitante sea la capacidad de almacenamiento del medio a utilizar y no un tiempo fijo de grabación para cada evento.
- c) Tener un medio de almacenamiento seguro que pueda ser reutilizable un gran número de veces y no presente problemas de borrado.

ANALISIS DEL SISTEMA DIGITAL (ADQUISIDOR)

- d) Se requiere que la disipación de potencia del sistema sea baja.
- e) Utilizar componentes electrónicos de fácil obtención en el mercado.

En base a los requerimientos anteriores se plantearon como secciones constitutivas del adquisidor digital las etapas siguientes:

- 1) Una etapa de entrada, la cual consista de un medio de adaptación de la señal analógica que proporciona el sismómetro.
- 2) Una etapa de conversión de señal, la cual tome la señal analógica del sismógrafo y la manipule para presentarla en forma digital.
- 3) Una etapa de control y procesamiento, en base a un microprocesador 80C88, que se encargue de seleccionar los recursos del sistema.
- 4) Una etapa de almacenamiento temporal y permanente de datos. El almacenamiento permanente de la información se hará en memorias semiconductoras no volátiles EEPROM tipo FLASH, que han respondido adecuadamente en anteriores proyectos.
- 5) Una etapa de comunicación con el usuario, que consista en interrelacionar un teclado, un display y el microprocesador.
- 6) Una etapa que logre establecer la compatibilidad con una computadora portátil.



**IV.3. ESTUDIO DE FACTIBILIDAD.**

Cuando se propone un nuevo proyecto normalmente se lleva a cabo un estudio de factibilidad. Este estudio da como resultado el reconocimiento tanto de los beneficios como de los riesgos inherentes al desarrollo y a la implementación del sistema de aplicación.

Para comenzar este estudio nos hicimos la siguiente pregunta: ¿Se puede implementar la aplicación propuesta con la tecnología actual?. La respuesta es afirmativa, ya que en el mercado existen circuitos integrados que pueden ser útiles para el desarrollo del sistema. Sin embargo, nos encontramos con la restricción de que nuestro diseño, tenía que partir de los recursos que el Instituto de Ingeniería nos proporciona.

Una vez resuelta la pregunta anterior, la interrogante a resolver fue: ¿Dará el sistema beneficios mayores que los costos?. Como respuesta a ello se comenta lo siguiente: el sistema digital en circuito impreso con el que se implementará al MEQ-800 tendrá un costo aproximado de \$830 dólares. Si tomamos en cuenta que los sismógrafos digitales existentes en el mercado tipo EDA ó tipo Kinometrics tienen un costo que fluctúa entre \$7000 y \$8000 dólares, entonces podemos tener un punto de comparación para justificar la inversión realizada, ya que el costo de actualizar 12 sismógrafos MEQ-800 de la Coordinación equivale a invertir \$10000 dólares, lo que resulta más económico que comprar dos sismógrafos digitales del mismo tipo de los mencionados anteriormente.

## IV.4. DISEÑO CONCEPTUAL.

El diseño conceptual de un sistema es un bosquejo que señala sus características, identifica la función para la que sirve e indica como éste interactúa con otros elementos.

En la figura 4.1 se muestran los bloques esenciales del sistema y la forma como están relacionados.

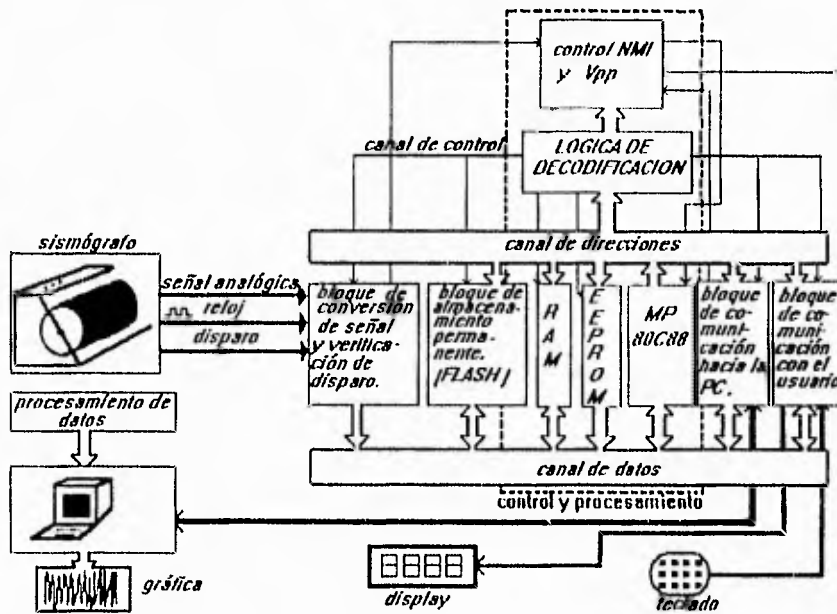


FIGURA 4.1. Diagrama de bloques del sistema digital (adquisidor).

#### ANALISIS DEL SISTEMA DIGITAL (ADQUISIDOR)

En el diagrama de bloques (figura 4.1) se pueden apreciar básicamente, seis secciones: base de tiempo, adaptador de la señal analógica a digital y detector de evento, almacenamiento de datos, control y procesamiento, comunicación con el operador y comunicación con la computadora. A continuación se explica lo que realizará cada sección.

La **base de tiempo** se encargará de generar las señales necesarias para que opere el microprocesador, dar la velocidad de conversión y hacer funcionar al USART. Además, de la base de tiempo propia del MEQ se tomará una señal de 60 Hz para generar el muestreo y dar la señal de inicio de conversión. Asimismo, de las líneas de datos que van a los circuitos que manejan el display del MEQ, se tomarán los dígitos correspondientes al tiempo.

El bloque de **conversión de la señal analógica a digital** se desarrollará alrededor de un conversor A/D de National para generar palabras de 12 bits. Dentro de este bloque se **detectará un evento** cuando la señal sobrepase un nivel especificado de umbral.

El **bloque de control y procesamiento** se encargará de coordinar la operación de las demás secciones del sistema. Puesto que este bloque se construirá alrededor de un microprocesador CMOS de 16 bits de INTEL, se podrá considerar al sistema como una microcomputadora dedicada, la cual contendrá cierta cantidad de memoria RAM y EPROM y ciertos dispositivos periféricos.

El **bloque de almacenamiento permanente** guardará en memorias no volátiles EEPROM tipo FLASH, la información enviada por el conversor A/D multiplexada con el tiempo y además grabará una tabla que contendrá información de los eventos. Este bloque elimina la necesidad de utilizar soporte de baterías, el cual sería necesario si se utilizaran memorias RAM.

El diálogo entre el sistema y el operador se realizará mediante el **bloque de comunicación con el usuario**, el cual constará de los circuitos necesarios para permitir la interrelación entre el teclado, el display y el microprocesador.

Finalmente, en la figura 4.1 se observa un **bloque de comunicación con la computadora**, el cual permitirá dar el formato adecuado a los datos para transmitirlos asincrónica y bidireccionalmente hacia la PC. El circuito principal de este bloque será el USART, de la familia de dispositivos del 80C88, el cual es un receptor-transmisor síncrono y asíncrono universal, que permite su programación desde el microprocesador.

## CAPITULO V

---

### *DISEÑO Y DESARROLLO DEL HARDWARE DEL ADQUISIDOR*

Al empezar con el diseño del sistema, se observó que para acoplar los circuitos que se tendrían que desarrollar alrededor del MEQ-800, era necesario modificar parte del alambrado del equipo, con el fin de obtener las señales necesarias para tomar el código BCD correspondiente al tiempo interno.

En el espacio que ocupan las baterías internas con las que vienen equipados los sismógrafos, se decidió colocar los circuitos desarrollados. Esta selección se hizo tomando en cuenta que en el campo estas baterías no se utilizan y el soporte de alimentación se realiza generalmente, con acumuladores externos que proporcionan  $\pm 12$  volts.

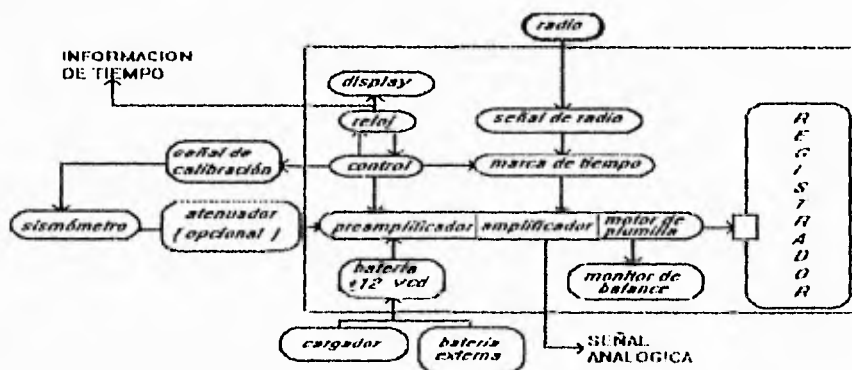


FIGURA 5.1 Diagrama que especifica las 2 señales del sismógrafo analógico que están ligadas al adquisidor digital.

Después de estudiar la forma en que estaban conectadas las baterías de alimentación a los circuitos analógicos y digitales que forman parte del equipo y de considerar que se necesitaba desarrollar circuitos que van ligados a ambas partes, se vio que era indispensable modificar parte del alambrado de la alimentación sin afectar las características de operación del sismógrafo MEQ-800.

Se decidió que la información sobre el tiempo propio del MEQ-800 se tomara de las líneas de datos que van a los circuitos de control de display del equipo (figura 5.1).

En la figura 5.1 también se puede observar, que de la sección de amplificación y filtrado, se toma la señal analógica a ser digitizada.

#### **V.1 BASE DE TIEMPO.**

Como el MEQ-800 tiene una base de tiempo propia que le permite llevar su reloj interno, se consideró conveniente tomar de dicho módulo una señal de 60 Hz proveniente del cristal, para generar la señal de inicio de conversión y una señal que va al generador de muestreo (el cual proporciona 60 ó 120 pulsos por segundo).

Se desarrolló un circuito basado en un cristal de 1.8432 MHz para la operación del microprocesador (figura 5.4), junto con contadores del tipo 4526 y 40163, el cual sirve para hacer funcionar al USART y dar la velocidad de conversión(84 KHz), (figura 5.5).

La información sobre el tiempo propio del MEQ-800 se toma de las líneas de datos que van a los circuitos de control del display. En este punto, los datos van en formato BCD, multiplexados en el tiempo y en el orden siguiente: dígito de decenas de hora, dígito de unidades de hora, dígito de decenas de minuto, dígito de unidades de minuto, dígito de decenas de segundo y dígito de unidades de segundo. Además, para identificar cada dígito, se utilizan otras dos líneas: la que marca la presencia de un dígito y la que indica el dígito de las decenas de hora. Esta información es la que permite identificar el evento registrado. (Figura 5.4).

La señal de un pulso por segundo tomada del MEQ-800, permite sincronizar el flujo de datos e indicar el momento de adquirir la información sobre el tiempo.

## V.2 ACONDICIONADOR DE LA SEÑAL ANALÓGICA A DIGITAL.

En el MEQ-800, la señal analógica del sismómetro es amplificada a 60 dB mediante un preamplificador de ganancia fija, y después filtrada y amplificada con un circuito de parámetros variables. Este circuito posee capacidad de amplificar la señal otros 60 dB en pasos de 6 dB en 6 dB y filtrar la señal en una banda de paso que tiene frecuencias de corte inferior de 3.5, 5 y 10 Hz y superior de 5, 10, 30 y 70 Hz.

La señal no se puede tomar directamente de la salida que va a la plumilla, para llevarla al conversor A/D, porque lleva mezcladas las marcas de tiempo que alterarían la señal analógica real proveniente del sismómetro. Debido a lo anterior,



la señal fue tomada en un punto en el cual no se suma con la marca del tiempo, lo anterior se realiza mediante un seguidor de voltaje de tal forma que no se cargue el circuito en ese punto. La salida del seguidor es amplificada aproximadamente 7 veces mediante un amplificador LF 358 (ver figura 5.5), el cual permite que la señal varíe en el rango de  $\pm 5$  volts. La salida del L 358 se aplica al circuito de muestreo y retención de señal, que fue realizado a partir de un LF 198. La muestra de la señal se convierte a palabras de 12 bits mediante un circuito ADC-1210, el cual para su operación en el rango de  $\pm 5$  volts requiere un voltaje de referencia positivo de 5 volts y uno negativo de -7.5 volts. Estos voltajes se obtienen a partir de la fuente de  $\pm 12$  volts con la que se polariza el MEQ-800 y mediante un regulador del tipo 7805 y un L 358 tal y como se presenta en la figura 5.5 en la parte superior izquierda. El proceso de conversión se inicia con cada pulso presente en el pin 9 del monoestable del circuito 4538 y prosigue a una velocidad dada por los pulsos de la señal de 100 Hz. generada en la base de tiempo. doce pulsos de la señal de 100 KHz después de presentarse el pulso de inicio de conversión, se genera la señal de fin de conversión, la cual indica que a la salida del conversor se encuentra la representación en binario del voltaje correspondiente a la muestra de la señal analógica tomada. los 12 bits de representación binaria de la señal son enviados al bus del microprocesador a través de un par de circuitos 74C244, los cuales son controlados por el microprocesador mediante instrucciones del tipo OUT.

De la sección de amplificación y filtrado se toma la señal analógica a ser digitizada. Esta señal proviene de un sensor de velocidad en contacto con el terreno, por tanto tiene un contenido de frecuencias máximo de 25 a 30 Hz; en algunos

casos las frecuencias de interés son menores a estos valores.

La sección de adaptación de la señal analógica a digital está construida a partir de un LM358, un LF398 y el conversor análogo a digital ADC-1210, el cual para su operación en el rango de  $\pm 5$  volts requiere de un voltaje de referencia positivo de 5 volts y uno negativo de - 7.5 volts. Estos voltajes se obtienen a partir de la fuente de  $\pm 12$  volts con la que se polariza el MEQ-800 y mediante un regulador del tipo 7805 y un LM358. Mediante el LM358, se construye un desacoplador de impedancia y una etapa de amplificación de tal forma que la señal adquiriera una amplitud similar a la que llega a la plumilla de graficación del MEQ-800. La etapa de filtrado no fue desarrollada debido a que se aprovecha la existente en el sismógrafo.

El LF398 permite tomar una muestra de la señal analógica, con cada flanco de subida de la señal de 60 ó 120 pulsos por segundo, y mantener el valor adquirido fijo a la entrada del conversor análogo-digital, durante el tiempo necesario para que el conversor realice la obtención del dato digital correspondiente. El ADC-1210 es un convertidor del tipo de aproximaciones sucesivas, de baja potencia y velocidad media, que proporciona una resolución y precisión de 12 bits. En el circuito construido, el conversor se configuró para operar con una entrada analógica bipolar de valor máximo de  $\pm 5$  volts.

El que se tenga configurado de esta manera el conversor, permite presentarle al microprocesador una palabra de 16 bits, la cual incluye los 12 bits del conversor A/D y 4 bits auxiliares; estos bits generalmente tienen un valor de cero, a

excepción de cuando se presenta el pulso por segundo, en donde adquieren un valor de 1100.

### V.3 DETECTOR DE EVENTO.

La mayoría de los equipos digitales cuentan con disparadores que ejecutan un algoritmo de análisis de ondas sísmicas. Variando los parámetros del algoritmo se puede escoger, a criterio del operador la clase de evento a registrar, ya sean microtemblores, temblores lejanos, etc. Cuando no se tiene conocimiento del terreno en que se está registrando no siempre es exitoso y se pueden registrar eventos no deseados, que pueden llegar a saturar la memoria.

Por lo anterior, se decidió utilizar en este sistema el disparo por umbral, y construir, el detector de evento en base a un detector de pico y un comparador (figura 5.3). En ésta sección se compara la amplitud de la señal proveniente de la entrada de la plumilla con un voltaje que se puede ajustar al valor al cual se requiere disparar, tomando en consideración que 1 volt es aproximadamente 1mm de desplazamiento de la plumilla.

La operación del disparador es la siguiente: La señal analógica del sismómetro es tomada de la salida TAPE del MEQ, que se encuentra en el amplificador AS110, mediante un seguidor de voltaje. A continuación se le da una ganancia que permite establecer una relación con la amplitud que se tiene en el desplazamiento de la plumilla del registrador y se pasa a un detector de pico; posteriormente, esta señal se compara con el voltaje preestablecido del disparo.

voltaje preestablecido del disparo.

#### V.4 CONTROLADOR BASADO EN EL MICROPROCESADOR 80C88.

El sistema se basó en el microprocesador 80C88, debido a que en la Coordinación de Ingeniería Sismológica se tiene todo el soporte necesario para trabajar con él.

##### V.4.1 Microprocesador.

El elemento principal del sistema digital es el microprocesador 80C88, debido a que él coordina la actividad de todas las etapas del sistema y realiza las operaciones aritméticas y lógicas que el proceso de los datos requiere.

El 80C88 de INTEL es un microprocesador de "16 bits", que puede ser usado como el CPU en una microcomputadora. El término "16 bits" significa que tiene una Unidad Aritmética Lógica (ALU), Registros Internos y un Set de Instrucciones diseñado para trabajar con palabras binarias de 16 bits.

El 80C88 tiene un bus de direcciones de 20 bits, que le permite direccionar hasta 1,048,576 bytes de memoria. Este microprocesador cuenta con un bus de datos de 8 bits, lo que significa que solo puede leer ó escribir datos a memoria y hacia los puertos de 8 bits al mismo tiempo. El microprocesador se configuró en modo mínimo de operación.

Como el microprocesador maneja un único canal para

direcciones y datos, lo cual imposibilita la conexión de éste con dispositivos que no cuentan con retenedores internos en sus entradas, fue necesario establecer mediante circuitos 74C73 un medio que permitiera mantener las direcciones enviadas al canal del microprocesador.

La lógica alrededor de éste integrado se puede observar en la figura 5.4.

#### **V.4.2 Circuitos de Arranque.**

La figura 5.5 muestra el circuito de RESET, que es indispensable para que el microprocesador funcione adecuadamente.

La entrada de RESET del microprocesador, puede llevarse a un nivel alto cuando se presiona el interruptor del tipo normalmente abierto o bien cuando se enciende el sistema (RESET de encendido). Este RESET de encendido se obtiene mediante un circuito multivibrador, el cual queda polarizado en cuanto se energiza el sistema, y permite tener en su salida un nivel alto que hace conducir al transistor Q1. Cabe mencionar que el microprocesador reconoce la señal de "reset" cuando ésta permanece en un nivel alto durante un tiempo mínimo de cuatro ciclos de reloj.

#### **V.4.3 Mapa de Memoria y Almacenamiento de Datos (FLASH).**

Para poder realizar la actualización del Sismógrafo MEQ-800 y a partir de las necesidades del microprocesador 80C88, se planteó el MAPA DE MEMORIA, presentado en la figura 5.2. En dicha figura se muestra el uso de 3 tipos de memorias diferentes: RAM,

## FLASH y EPROM.

La memoria RAM abarca de la dirección 0000H hasta la 1FFFFH. El primer Kbyte esta asignado a los 256 vectores de Interrupción INTR que puede manejar el microprocesador, de los cuales los 5 primeros son reservados por el fabricante para operaciones especificas tales como: división entre cero, ejecución paso a paso, interrupciones no mascarables, punto de prueba y sobreflujo. Los 27 siguientes son reservados por INTEL y los 224 restantes son para uso general del usuario.

De la localidad 0400H hasta la 0700H se pueden guardar datos, registros y/o direcciones que se originan durante la ejecución del programa asi como la sección correspondiente a la PILA.

A partir de la localidad 0700H se almacenan los digitos provenientes de la base de tiempo que proporciona el Sismógrafo.

De la localidad 0800H hasta la localidad 01AC1H, se tiene la sección reservada para guardar la información del preevento.

Los siguientes 12 segmentos de 128 Kbytes cada uno, están ocupados por las memorias FLASH utilizadas para guardar los datos que provienen del Sismógrafo. El rango de memoria para almacenamiento está comprendido entre las direcciones 10000H y CFFFFH. El primer Kbyte de esta sección de memoria se decidió reservarlo para la tabla de datos. El espacio que se ocupa para identificar cada evento en la tabla es de 11 bytes, guardándose como a continuación se muestra:

1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	----	----

**BYTES**

- 1            Segmento inicial del evento.
- 2,3        Dirección del inicio del evento.  
(Formado por el segmento y desplazamiento correspondiente)
- 4            Segmento final de evento.
- 5,6        Dirección final formada por el segmento y desplazamiento.
- 7,8,9      Reloj inicial del evento, con el siguiente formato: HH:MM:SS
- 10,11     Máximo absoluto del evento.

Lo que se desea con esta tabla de datos es tener un medio que permita reiniciar la operación del sistema a pesar de las fallas en la alimentación; es decir, que si el sistema se apaga y se enciende nuevamente, el programa principal en el segmento puede recuperar la información que indica la dirección y el segmento, con lo cual se continuará la grabación de los datos en memorias FLASH. Al obtener las direcciones iniciales y finales de cada evento, es posible seleccionar cualquiera de ellos para la transmisión de datos hacia la PC y su correspondiente manipulación.

Finalmente, las memorias EPROM ocupan de la localidad E0000H hasta la FFFFFH y en ellas se encuentran almacenados los diversos programas de control del sistema: el programa principal a partir de la localidad 0100H, la subrutina de servicio a interrupciones NMI para capturar los datos del conversor se encuentra en las localidades 1000H y 1450H de EPROM, la subrutina

para capturar los dígitos del reloj se encuentra en la localidad 1300H y el vector de reset está en la localidad 01FF0H.

#### V.4.4 PUERTOS.

La lógica de decodificación del sistema se encarga de seleccionar el dispositivo que utilizará el canal en un cierto momento. Entre los dispositivos que tienen acceso al canal de datos se encuentran: las memorias RAM, FLASH y EPROM, un puerto serie programable, un display de 7 segmentos de cristal líquido y un teclado. Ver las figuras 5.4 y 5.5

Las líneas del canal de direcciones empleadas para seleccionar los diferentes puertos fueron las cuatro menos significativas. (A0 a la A3).

Las señales utilizadas para habilitar el decodificador son: IO y DEN del microprocesador. La primera se verifica en bajo cuando se presentan las instrucciones OUT ó IN en el programa principal del Sismógrafo y la segunda se verifica en bajo cuando existe un dato válido en el canal de datos.

##### V.4.4.1 Teclado.

Como medio para que el usuario pueda dar comandos al sistema, se ha empleado un teclado, el cual está compuesto por un arreglo de 20 interruptores de un polo un tiro distribuidos en 5 renglones y 4 columnas. Par utilizar dicho teclado se ha utilizado un circuito codificador 74C923, el cual permite al microprocesador una fácil determinación de la tecla pulsada por el usuario.



De acuerdo al mapa de puertos, el Teclado puede ser leído por el microprocesador mediante la instrucción IN AL,04H.

#### V.4.4.2 Display de cristal liquido.

Para que el operador pueda tener conocimiento del correcto funcionamiento del sistema se dispuso de un medio de salida de información numérica basado en un display de cristal liquido(LCD), con cuatro dígitos de 7 segmentos. Cada dígito está perfectamente identificado y puede ser accesado mediante el valor binario de 2 líneas del bus de datos, tal y como se muestra a continuación:

Ejemplo: Si se desea enviar a la posición 2 el dígito 8 y visualizarlo en el display, se ejecutan las siguientes instrucciones:

```
MOV AL,08H      ;Valor del dígito.
OR AL,20H      ;Indica la posición de salida.
OUT 05H,AL     ;Enviamos al display la
                posición del dígito.
```

Este puerto, puede mostrar a la vez cuatro dígitos, cada uno de ellos tiene una posición única en él. Las posiciones en el display se muestran a continuación:

posición 3	posición 2	posición 1	posición 0
------------	------------	------------	------------

El display se puede observar en la figura 5. 5.

#### V.4.4.3 Conversor A/D.

Para acceder este puerto es necesario utilizar 2 instrucciones de entrada, debido a que se necesita adquirir la

parte baja y la parte alta de cada una de las palabras de información que provienen del conversor. Las instrucciones que los representan son:

IN AL,00H      Para la parte BAJA de la palabra    e  
 IN AL,01H      Para la parte ALTA.

Correspondiendo la dirección 00H para el byte bajo y 01H para el byte alto. Este sección se encuentra en la figura 5.4.

#### V.4.4.4 Puerto serie de comunicación.

El circuito elegido para realizar la comunicación entre el sistema y la computadora fue el 82C51. La operación básica de este circuito programable es la de recepción y transmisión universal para comunicación sincrónica y asíncrona de datos (USART). El USART acepta datos con formato paralelo provenientes del microprocesador 80C88 y los convierte a un formato serie, adecuándolos para su transmisión. Asimismo, puede recibir datos serie enviados a él por algún dispositivo externo y convertirlos en paralelo para presentarlos al microprocesador.

A continuación se muestran las instrucciones que permiten manejar este puerto.

DIRECCIÓN	NOMBRE	INSTRUCCIÓN.
03H	DAT USART	IN AL,03H (LEER DATO) OUT 03H,AL (ESC.DATO)

```

13H           REG USART           IN AL,13H (LEER DATO)
                                OUT 13H,AL (ESC.DATO)

```

Esta sección se puede visualizar en la figura 5.5.

#### V.4.4.5 Lectura y escritura en FLASH.

Las memorias FLASH 28F020, elegidas para este sistema, poseen tiempos de escritura del orden de 16  $\mu$ s por byte y además no necesitan soporte de batería para mantener su información, lo cual las hizo adecuadas para la aplicación de almacenamiento permanente.

La operación de estas memorias es controlada mediante voltajes e instrucciones cargadas en el registro de comandos. Si el voltaje de programación tiene un valor de 0 a 5 Volts, sólo es posible la lectura de la memoria; pero si el voltaje es de 12 volts, entonces es posible programarla o borrarla.

El valor del voltaje de programación es controlado mediante dos líneas del decodificador de puertos las cuales manejan cada una de las operaciones de manera independiente. A continuación se muestra una tabla que da: la dirección del puerto correspondiente, el nombre y la instrucción para cada una de ellas:

DIRECCIÓN	NOMBRE	INSTRUCCIÓN.
08H	VppF=12	OUT 08H,AL
09H	VppF=0	OUT 09H,AL

Para este punto observar la figura 5.4.

**V.4.4.6 Controlador de interrupciones.**

Para el manejo de esta sección, contamos con 3 opciones:

- 1). Activación de Interrupciones.
- 2). Restablecimiento de Interrupciones hacia el microprocesador.
- 3). Desactivación de Interrupciones.

Estas 3 opciones están directamente relacionadas con el microprocesador. Para cada una de ellas, es necesario tener una de las líneas del decodificador de puertos.

En la siguiente tabla se muestra la dirección, el nombre y la instrucción de cada una de estas opciones.

DIRECCIÓN	NOMBRE	INSTRUCCIÓN
07H	DESACTIVA NMI	OUT 07H,AL
0AH	ACTIVA NMI	OUT 0AH,AL
0EH	RESTABLECE INT HACIA EL MICROPROCESADOR	OUT 0EH,AL

**V.5 FUENTE DE ALIMENTACIÓN.**

El sismógrafo MEQ-800 viene equipado con cuatro acumuladores internos de dióxido de plomo que proveen potencia al MEQ-800; sin embargo, en el campo la fuente de alimentación se realiza con

acumuladores externos conectados al equipo.

Para que el circuito opere se necesita alimentar al equipo con dos fuentes de poder las cuales proporcionen  $\pm 12$  VDC respectivamente. Cada fuente contiene 2 unidades de 1.5 amperes por hora, para combinar un total de capacidad de 3 amperes por hora.

El sistema de voltaje está controlado por el interruptor S-9, (ver capítulo correspondiente a la descripción del MEQ-800). Este interruptor en su posición de AMP da el voltaje necesario para todos los circuitos (excepto el reloj). En posición OFF se elimina el voltaje de todos los circuitos excepto los correspondientes al cronómetro, ya que el voltaje es mantenido a la salida del integrado del reloj.

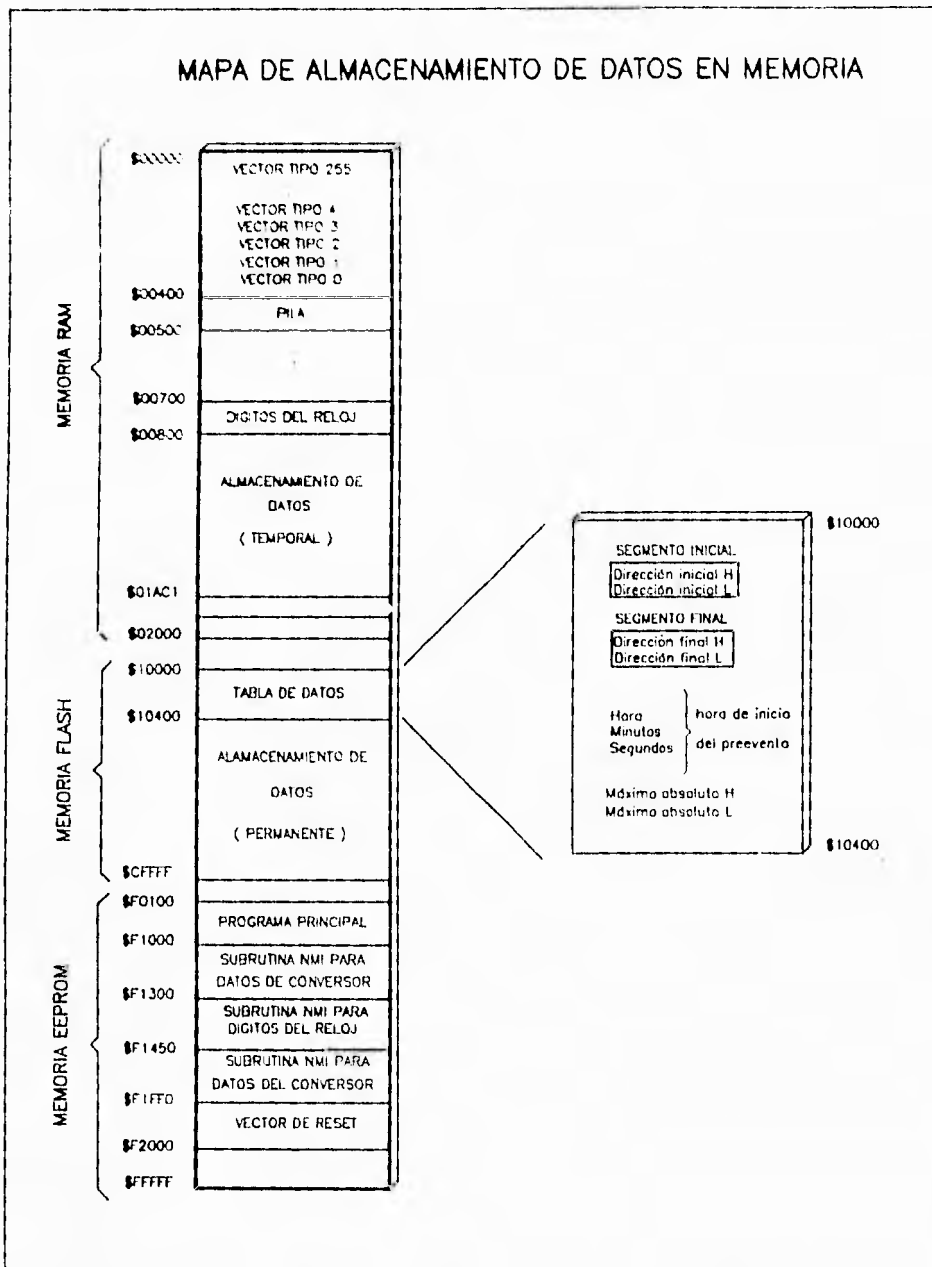
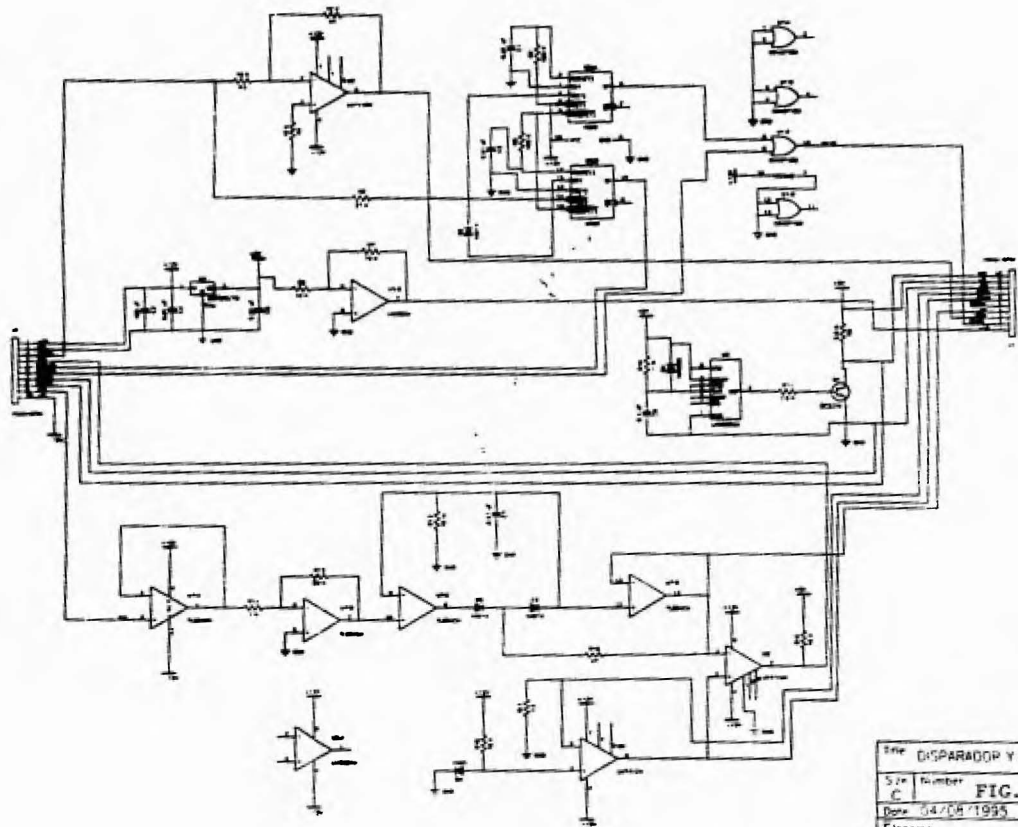


FIG. 5.2 MAPA DE MEMORIA Y FORMATO DE ALMACENAMIENTO DEL MEQ-800



Title: DISPARADOR Y GEN. MUESTREO			
Size	Number	FIG. 5.3	Rev.
C			1
Date:	04/08/1995	Drawn by:	
Filename:		Sheet	1 of 3

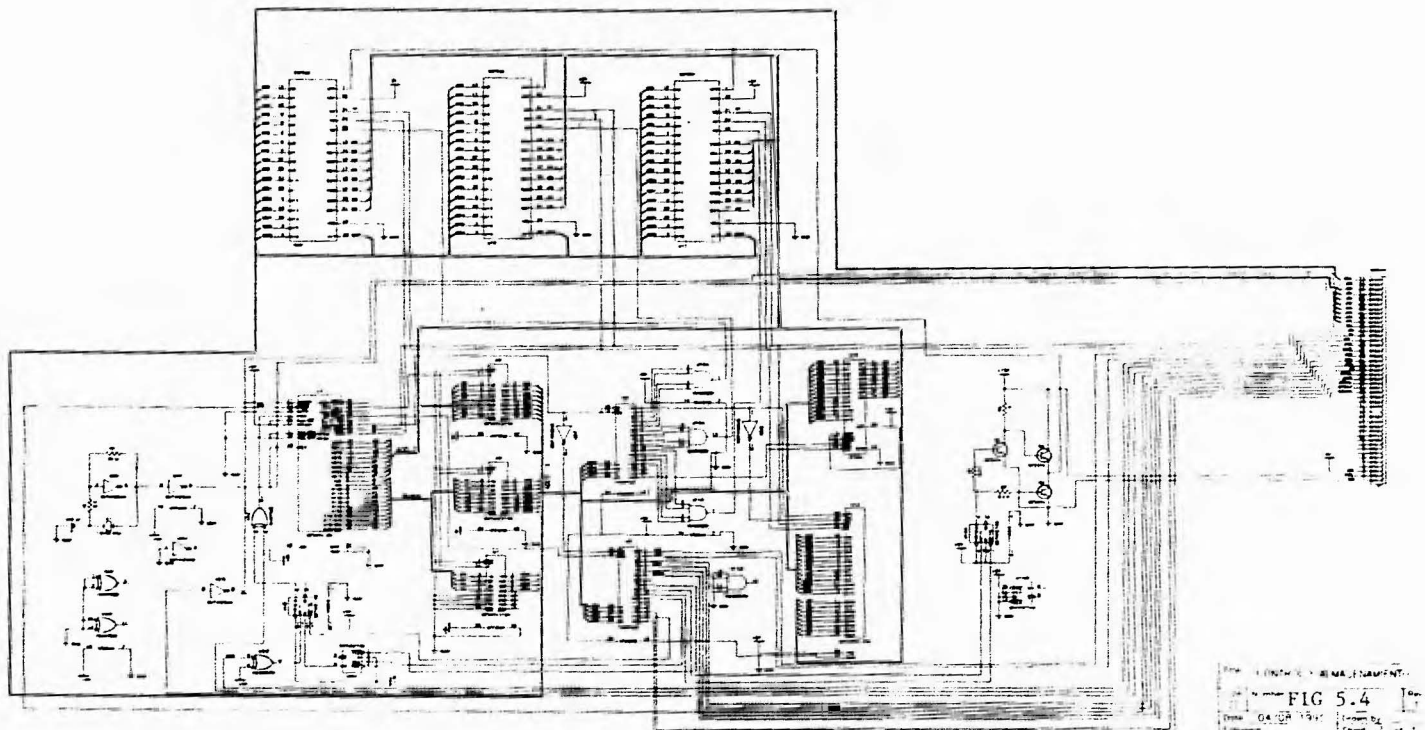


FIG 5.4



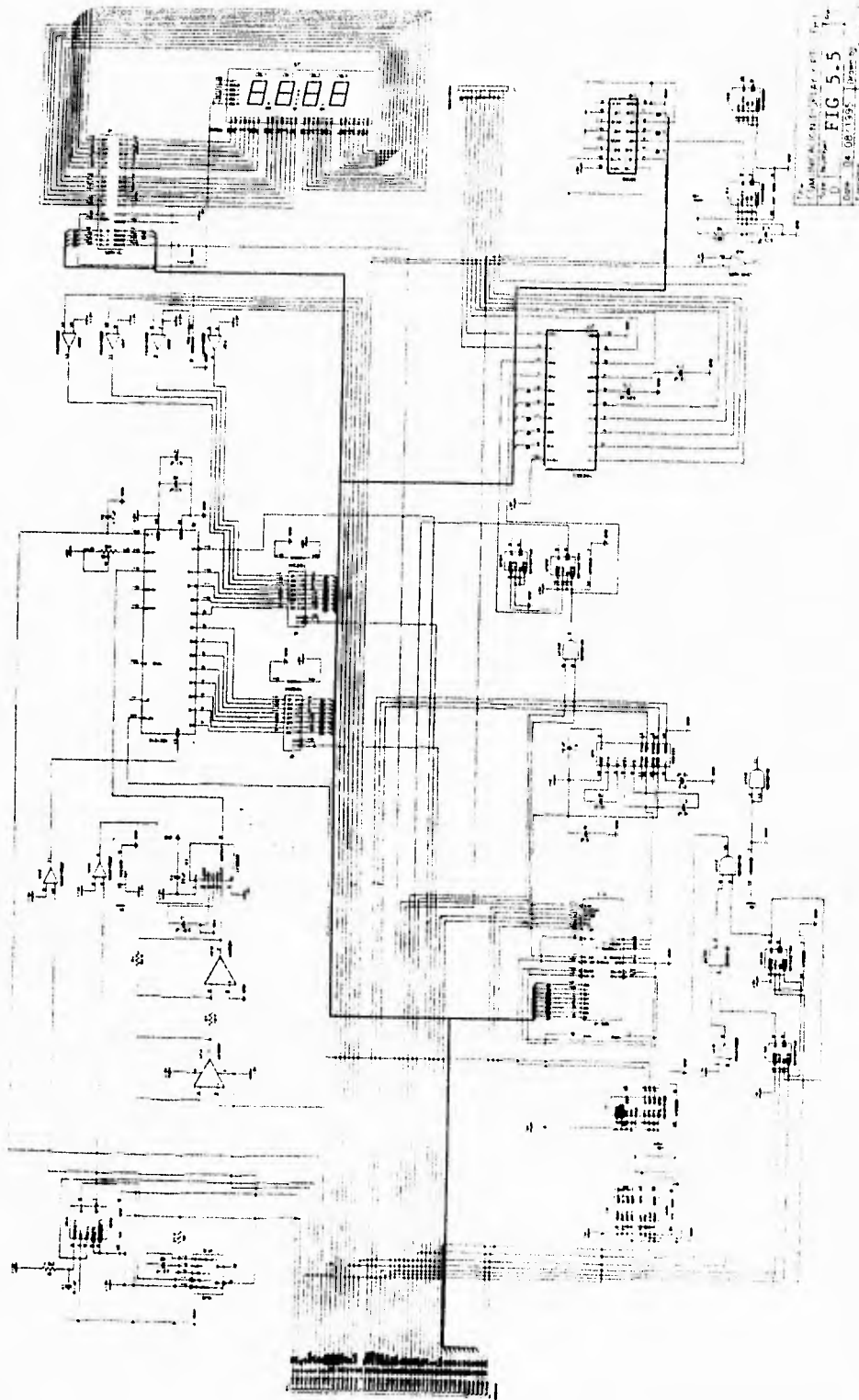


FIG 5-5  
Date: 08/1988

## **CAPITULO VI**

---

### ***ANALISIS DEL SOFTWARE DEL SISTEMA***

## VI.1 DEFINICIÓN DEL PROBLEMA.

Un sistema completo de adquisición y procesamiento de datos sísmicos involucra un medio de adquisición y un medio de procesamiento; en algunos casos, el medio de adquisición permite desarrollar también labores de procesamiento y en otros, el medio de adquisición está incluido en el sistema de procesamiento, tal es el caso de una PC con tarjetas de digitización.

El sistema pensado alrededor del MEQ-800 contempla un adquisidor digital y una PC para el procesamiento; por esto, es necesario contar con programas de control tanto para el adquisidor como para la PC.

A continuación se hace una descripción de los problemas y las alternativas de solución que se plantearon. Para, finalmente mencionar las elegidas para el mejor funcionamiento del sistema.

En primer lugar se investigó si el programa en ensamblador desarrollado para el sismógrafo DR-100 podía ser útil para controlar al adquisidor digital adaptado al MEQ-800. En un principio, se tomaron algunas piezas de este programa para valorar algunas secciones que conforman nuestro sistema de adquisición. Sin embargo, al analizarlo, se observó que presentaba problemas para cumplir con los requerimientos que se pedían, ya que éste permitía la captura de un evento en el momento en que el circuito de disparo así lo indicaba, pero no se incluía información adicional referente al tiempo, el preevento, postevento, el número de eventos registrados y no podía transmitir bidireccionalmente a PC. Además, el programa

#### ANALISIS DEL SOFTWARE DEL SISTEMA

antes mencionado sólo permitía registrar un evento con una duración de un minuto y medio como máximo. Y para registrar otro evento era necesario poner nuevamente al sistema en la condición de captación, lo cual no permitía utilizarlo para operación continua en el campo.

Todas estas carencias, llevaron a plantear el desarrollo de un programa en ensamblador para el sistema digital más completo, el cual realizará las labores de adquisición, control, procesamiento, almacenamiento y salida de la información; con la característica de que permitiera adquirir automáticamente eventos consecutivos, lo que significa que grabará de manera permanente hasta que se agote la capacidad de almacenamiento de las memorias de estado sólido (FLASH).

Por otra parte, los programas de captura y almacenamiento de datos en archivos para PC que se tienen en la Coordinación de Ingeniería Sismológica, presentaron problemas para adaptarse a nuestras necesidades, ya que leen datos vía puerto serial, pero están diseñados para capturar y decodificar tres canales de información en un formato específico que no coincide con el formato de información diseñado para el MEQ.

Además, como con los programas antes mencionados no se tiene la posibilidad de observar las gráficas de los eventos, los usuarios se enfrentan al problema de que tienen que manejar al mismo tiempo programas como DEGTRA5 ó PITSA, para hacer dicha observación , lo cual no es algo muy eficiente.

Así pues, tomando en cuenta todos estos problemas fue

necesario diseñar un programa para PC en lenguaje C para que desempeñe las siguientes funciones: comunicación bidireccional con el adquisidor via puerto serie, almacenamiento temporal de los datos de un evento en memoria RAM, decodificación y detección de errores, graficación de evento y almacenamiento en un archivo tipo ASCII. Además, el programa contará con una rutina de graficación, la cual permitirá observar el evento adquirido en el mismo lugar en donde se encuentra instalado el equipo, ahorrando al usuario tiempo y ampliando la gama de posibilidades, ya que en ese mismo instante, el operador no solamente podrá obtener información importante de cada evento, sino que además observará la gráfica correspondiente y decidirá si desea guardar la información en un archivo tipo ASCII, para posteriormente realizar un procesamiento más fino sobre los datos obtenidos.

## **VI.2 ANÁLISIS DEL SOFTWARE DEL ADQUISIDOR.**

### **VI.2.1 Objetivo.**

Realizar un programa para el microprocesador 80C88 de INTEL, que tenga control absoluto sobre todos los recursos con los que cuenta el sistema y aprovecharlos en forma eficiente para que cumpla con los requerimientos solicitados por el usuario.

### **VI.2.2 Determinación de los requerimientos.**

Dentro de las funciones que debe cumplir el programa de adquisición se encuentran las siguientes:

- Manipular y procesar la información que proporciona el conversor A/D.

#### ANALISIS DEL SOFTWARE DEL SISTEMA

- Permitir autorrestablecer las condiciones de trabajo con las cuales estaba operando, en caso de que por alguna circunstancia no tuviera alimentación durante un determinado lapso de tiempo.

- Controlar los recursos con los que cuenta el sistema, manejando para ello los segmentos de datos en las memorias de almacenamiento, los dígitos del reloj los cuales proporciona la base de tiempo desarrollada, la activación y desactivación de interrupciones y el establecimiento adecuado de la comunicación bidireccional con una computadora personal.

- Elaborar una tabla de eventos, que contenga, por cada evento, la dirección inicial y la final del mismo, el tiempo de inicio del evento y el valor máximo absoluto de cada evento.

- Permitir la comunicación bidireccional con la PC, con lo cual se logrará: transmitir un evento, cambiar la velocidad de transmisión entre el adquisidor y la PC, borrar las memorias de almacenamiento, proporcionar el número de eventos almacenados y enviar la tabla de eventos.

- Grabar un evento de manera continua, teniendo como única limitante la capacidad de almacenamiento de las memorias.

- Manejar un preevento con duración de 20 segundos, con el propósito de tener completa la información correspondiente a las ondas de arribo P y S.

- Manejar un postevento cuya duración sea de 30 segundos, con el objeto de tener la información final de cada uno de ellos.

- Revisar continuamente la señal del disparo durante el postevento, con el propósito de seguir almacenando la información correspondiente al mismo y evitar perder el arribo de un nuevo evento.

- Fácil manipulación del equipo ya sea de manera manual al utilizar el puerto del teclado o utilizando el protocolo de comunicación entre el equipo y una computadora portátil.

- Mantener actualizada la información del número de eventos almacenados en memoria.

- Borrar las memorias de almacenamiento (FLASH) de datos una vez que ya se obtuvo la información de interés, sin tener que retirarlas del equipo.

Como se puede observar, lo que se busca es un sistema que sea completamente autónomo en su operación, además de que sea lo suficientemente robusto para proporcionar y almacenar la información completa de un evento.

#### **VI.2.2.1 Lenguaje ensamblador, herramientas y equipo elegidos para la programación.**

La parte digital construida como complemento del MEQ-800 de registro analógico, está basada en el microprocesador 80C88 de Intel; por lo tanto, toda la programación del adquisidor debe ser escrita en base al ensamblador de dicho microprocesador, que es el mismo ensamblador utilizado para PC's.

El lenguaje ensamblador usa una representación simbólica del

código binario real que el microprocesador ejecuta directamente. Cada operación en lenguaje ensamblador corresponde a una tarea básica que lleva a cabo el microprocesador. El "set de instrucciones" del lenguaje ensamblador 8088 cuenta con 92 instrucciones diferentes.

Para programar en lenguaje ensamblador en la Coordinación de Ingeniería Sismológica se cuenta con los siguientes programas:

PROGRAMA	DESCRIPCIÓN
EDITOR DE TEXTO	El editor de texto es un programa que permite crear código fuente. En el mercado existen varios. Se empleará el editor de Turbo Pascal.
MASM (ASM)	MASM es el programa Macroensamblador. Para cargarlo en memoria se requieren 96K de RAM. MASM se emplea para ensamblar código fuente y generar código objeto. Durante su operación, MASM pide al usuario los nombres del archivo fuente (extensión ASM) y genera un archivo objeto(extensión OBJ).



<b>LINK</b>	Este programa se emplea para encadenar diversos módulos objeto generados ya sea por MASM o por otros compiladores. El programa se encarga de asignar las localidades de memoria absolutas para relocalizar el código objeto. El encadenador permite el desarrollo de código modular ya que con él es posible combinar módulos individuales y producir un programa completo.
<b>DEBUG</b>	El programa DEBUG es útil durante la fase de desarrollo de programas. Tiene características que permiten ejecutar al usuario: un programa paso a paso y examinar dinámicamente como cambian tanto la memoria como las banderas.

Aunque el lenguaje ensamblador da a los programadores la potencia de realizar las tareas con la mayor eficiencia, es notoriamente difícil trabajar con él en el desarrollo y depuración de un programa; por lo mismo, se eligió al diagrama de flujo como la herramienta adecuada para programar, ya que

será de gran ayuda para diseñar cada módulo, que posteriormente será traducido a lenguaje ensamblador. De esta forma, se podrán realizar pruebas de manera independiente para cada una de las rutinas, con el fin de modificar o perfeccionar el adecuado funcionamiento de las mismas; para posteriormente integrar todas las rutinas en un programa principal.

Una computadora con un microprocesador 80C88 en adelante es necesaria para el correcto funcionamiento de los diferentes programas de la tabla de la página anterior. En la Coordinación se cuenta con una computadora que tiene un microprocesador 80286 y es suficiente para llevar a cabo esta aplicación. Además, se cuenta con los dispositivos para grabar programas en memorias semiconductoras, así como con el borrador de memorias de rayos ultravioleta; como se puede observar se cuenta con el equipo necesario para realizar la programación en ensamblador que controlará al adquisidor digital.

### **VI.2.3 Diseño Conceptual.**

El diseño conceptual establece un enfoque más completo orientado hacia el usuario para la adecuada aplicación del sistema. Esta aplicación se basa en la forma en la cual el usuario ve la operación y funcionamiento del sistema. Establece las funciones de entrada y salida del sistema.

El programa de control del adquisidor estará constituido básicamente de 3 partes:

- Programa Principal.
- Subrutina NMI para adquisición de datos provenientes del Conversor A/D.

- Subrutina NMI para la captura de la información del reloj del MEQ-800.

A continuación se muestran los diagramas de bloques diseñados para cada parte mencionada anteriormente.

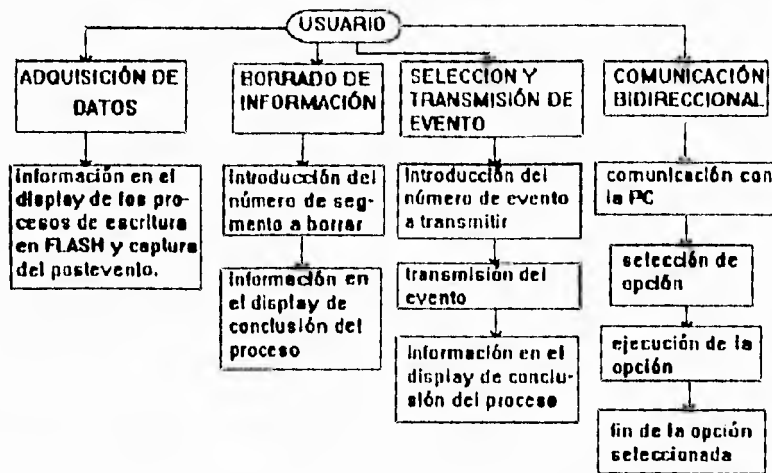


FIGURA 6.1. Diagrama de Bloques del programa principal.

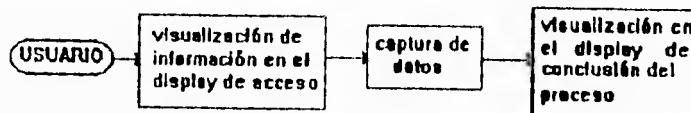
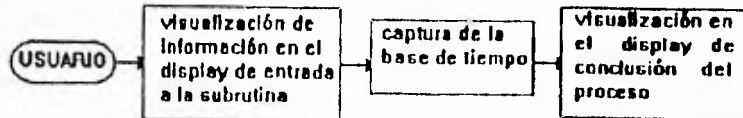


FIGURA 6.2. Diagrama de bloques de la subrutina de interrupción NMI para adquirir los datos del conversor A/D.



**FIGURA 6.3.** Diagrama de bloques para la segunda subrutina NMI la cual adquirirá los dígitos provenientes de la base de tiempo.

#### VI.2.3.1 Manejo de Errores.

El adquisidor digital sólo contará con el display para proporcionarle al usuario la información de los procesos que realizará el sistema.

Por esto es necesario que el programa en ensamblador contemple la detección de error en la manipulación del equipo y lo indique al usuario, para así tener la seguridad de que el programa hará operaciones sólo cuando se dé la información correcta.

Así pues, a través de la visualización de dígitos y letras en el display, el usuario podrá conocer si dió las instrucciones

correctas para que el sistema opere como él desea.

#### **VI.2.3.2 Interacción con el Usuario.**

El usuario podrá mantener comunicación con el adquisidor mediante 2 formas:

- Usando el teclado y el display. Con el teclado podrá indicarle al equipo la operación que desea y con el display podrá observar dígitos y/o letras que le indicarán el proceso que se está ejecutando.

- Usando un programa en PC, diseñado para este sistema, el adquisidor responderá al ser interrogado y podrá enviar la tabla de datos que contiene información importante de los eventos. Además , el usuario podrá elegir la operación que quiere que realice el adquisidor, sin necesidad de operarlo manualmente con el teclado. En este caso en la pantalla de la computadora se observará si se realizó o no la operación correctamente; al mismo tiempo el display mostrará los dígitos y/o letras correspondientes al procedimiento que está realizando.

### **VI.3 ANÁLISIS DEL SOFTWARE PARA LA COMPUTADORA**

#### **VI.3.1 Objetivo.**

Realizar un programa para computadora personal que se comunique bidireccionalmente con el sistema digital vía puerto serie; capture los eventos enviados por él y los almacene temporalmente en memoria; decodifique y detecte errores en la

información; grafique los eventos y permita la creación de archivos en formato ASCII. Este último punto se estableció con la finalidad de tener compatibilidad con los programas de procesamiento de uso común en la Coordinación de Ingeniería Sismológica.

### VI.3.2 Determinación de los requerimientos.

Para realizar el programa que tendrá que interactuar con el adquisidor digital fue necesario identificar los requerimientos del usuario.

Para llevar a cabo lo anterior, se realizaron una serie de entrevistas con los usuarios y de ellas, se estableció que el programa debe contar con las siguientes funciones:

- Manejo del puerto serie para establecer la comunicación bidireccional con el sistema digital, con la finalidad de interrogar al equipo y ponerlo a operar en una función específica.
- Captura de datos de un evento, vía puerto serie, y almacenamiento de los mismos en memorias RAM de la PC.
- Dar a conocer al usuario cuantos segundos de información puede almacenar en la memoria libre de su PC.
- Decodificación de la información guardada en RAM, lo que significa que las muestras marcadas cada segundo, con dígitos de reloj deben ser separados de ellos.
- Análisis de las muestras, sin información multiplexada de reloj, para detección de aquellas que no cumplan con el formato de palabra definido.
- Corrección de las muestras erróneas de un evento

adquirido con la finalidad de tener señales de evento muy aproximadas a la real.

- Almacenamiento de la información de un evento, un archivo ASCII, teniendo 2 líneas de encabezado con información importante para el usuario y una columna de números reales (4 bytes). Donde cada número real representará a una muestra del evento.

- Graficación de los datos de un evento, ya sea que estén almacenados en memoria (corregidos) o en un archivo con formato ASCII. La graficación deberá permitir observar con detalle alguna sección del evento que resulte de interés. Este proceso equivale lo que en algunos programas comerciales llaman "ZOOM".

Además, el sistema se debe caracterizar por:

- Hacer una estricta validación de la entrada de datos, lo que significa que si el usuario da una instrucción errónea, el programa indicará al usuario su error.

- Ser amigable y fácil de usar.

- Ser portable.

- Ser modular, para que dé opción a seguir creciendo si se considera necesario agregar más procesos.

#### **VI.3.2.1 Hardware**

El hardware es el equipo necesario para que el sistema funcione óptimamente; para evaluarlo, es necesario tener una concepción clara de lo que el usuario espera del sistema.

En el caso específico de este sistema de software para PC,

**ANALISIS DEL SOFTWARE DEL SISTEMA**

no se contempla la evaluación y compra de equipo, debido a que la Coordinación de Ingeniería Sísmológica donde será instalado, cuenta con el equipo que a continuación se describe:

<b>EQUIPO</b>	<b>CARACTERÍSTICAS</b>	<b>CANTIDAD</b>
PC HEWLETT PACKARD PORTÁTIL	Procesador 286 a 16 MHz, con 640 Kbytes de memoria base y 40 Mbytes de disco duro.	2
PC ACERMATE	Procesador 486 a 25 MHz, 640 Kbytes de memoria base, 3 Mbytes de memoria extendida y 250 megabytes de disco duro.	2
PC PORTÁTIL 486	Procesador 486 a 33 MHz con 640 Mbytes de memoria base, 3 Mbytes de memoria extendida y 120 Mbytes de disco duro.	1

El sistema de software para PC se diseñará para que tenga un buen desempeño en cualquiera de los equipos antes mencionados.



### VI.3.2.2 Lenguaje C y compilador para la programación.

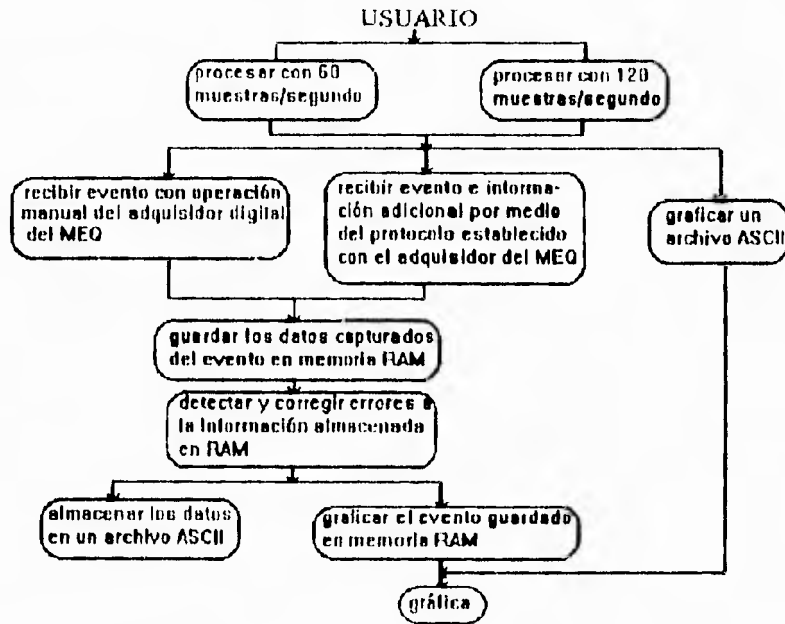
El lenguaje C es un lenguaje de nivel medio, porque combina elementos de lenguajes de alto nivel con la funcionalidad del lenguaje ensamblador. Como lenguaje de nivel medio, C permite la manipulación de bits, bytes y direcciones (los elementos básicos con los que funciona una computadora). El código de C es muy portable, la portabilidad significa que es posible adaptar el software de una computadora en otra. Esta portabilidad ahorra tanto tiempo como dinero. C da al programador lo que el programador quiere: pocas restricciones, estructuras de bloques, funciones independientes y un compacto conjunto de palabras clave. De acuerdo a las ventajas anteriores, se decidió desarrollar el programa para PC, que tiene que interactuar con el sistema digital, en lenguaje C.

Para este proyecto se cuenta con el compilador para lenguaje C de Borland, versión 2.0. Este compilador viene con una biblioteca estándar que proporciona funciones para llevar a cabo las tareas necesarias más comunes.

Cuando la longitud de un programa es muy grande se incrementa el tiempo de compilación y en cualquier proyecto el ahorro de tiempo es sustancial. Por ello, este compilador permite partir un programa en muchos archivos para que cada uno sea compilado por separado. Una vez que han sido compilados los programas se enlazan entre sí, junto con las rutinas de la biblioteca, para formar el código objeto completo. La ventaja de la compilación separada es que un cambio en el código de uno de los archivos no requiere de la recompilación del programa entero.

#### VI.3.4 Diseño conceptual

En el diseño conceptual se establece la forma general como el sistema cumplirá con los requerimientos identificados. En la figura 6.4 se muestran los módulos con los que contará el programa, así como la interacción con el usuario.



**FIGURA 6.4.** Diagrama que presenta los módulos que tendrá el programa para PC, el cual va a interactuar con el adquirente del MEQ.

Cabe mencionar que, cada uno de los módulos contará con rutinas que le indicarán al usuario cuando se ha producido un error en cualquiera de los procesos del sistema.

## **CAPITULO VII**

---

### ***DISEÑO Y DESARROLLO DEL SOFTWARE***

### VII.1 PROGRAMACION DEL ADQUISIDOR.

El Sistema Digital del Sismógrafo MEQ-800 está centrado en el uso del microprocesador 80C88 de Intel.

La programación del microprocesador mencionado anteriormente, esta elaborada en lenguaje ensamblador, por lo tanto, el software de soporte para el adquisidor no necesita de ningún otro lenguaje de programación (el cual sirva de soporte al primero) que realice procesos paralelos al lenguaje usado.

La programación del MEQ-800 digital, se puede dividir en 3 partes:

- Programa Principal.
- Subrutina de atención a interrupciones NMI, que adquiere los datos que provienen del conversor A/D.
- Subrutina que se encarga de capturar la base de tiempo proporcionada cada segundo. Esta subrutina es atendida por interrupciones NMI.

#### VII.1.1 PROGRAMA PRINCIPAL.

El Programa Principal es el encargado de establecer las condiciones iniciales del sistema; activar y desactivar interrupciones, insertar en las muestras correspondientes los dígitos de la base de tiempo adquiridos en cada segundo, mantener comunicación con el usuario mediante un teclado y un display de cristal líquido; seleccionar y transmitir un evento hacia PC; manejar una tabla de eventos en donde se registra la dirección inicial y final de cada evento, el reloj inicial y el valor

máximo absoluto correspondiente a cada uno de ellos; borrar las memorias FLASH; manejar el preevento y el postevento de cada evento y autorestablecer las condiciones de operación en caso de existir una falla eléctrica.

Este programa le da al usuario la alternativa de poder ejecutar cualquier opción predefinida siempre y cuando no ese realizando la captura de algún evento.

Al inicio de la ejecución del programa el usuario dispone de 50 seg para seleccionar alguna opción predefinida. Si durante ese tiempo, el usuario presiona alguna tecla que determina una función específica ó bien el programa detecta que se le esta enviando un carácter que establece la comunicación bidireccional, el programa va al módulo que atiende dicha función. Por otra parte, si al programa no se le indica que realice ninguna operación, durante el tiempo establecido; entonces, comienza la captura de los datos.

#### **VII.1.1.1 ESTABLECIMIENTO DE CONDICIONES DE OPERACION.**

Cuando se empieza a ejecutar el programa, en el display se le muestra al usuario el letrero A C C A para indicarle que el sistema esta en operación.

El establecimiento de las condiciones de operación, esta sujeto al hecho de verificar si existen eventos grabados en las memorias FLASH. Por lo que, al inicio del programa se va a la localidad en donde empieza la tabla de eventos y se verifica si la primera localidad tiene un valor de FF , con el propósito de establecer si se tiene que inicializar el sistema ó bien que nuevos valores se tienen que asignar a los contadores, registros y banderas.

De tal forma que si la primera localidad de la tabla tiene

FF, implica que no existen eventos grabados y el programa tendrá que dar las condiciones iniciales de operación; por el contrario, si la primera localidad tiene un valor diferente de FF, significa que ya existen datos en la tabla correspondientes al almacenamiento de un evento. Si esto ocurre, el programa detecta el número de eventos que existen almacenados, la localidad donde tendrá que empezar la grabación de un nuevo evento, así como también el segmento de grabación dentro de las memorias FLASH en donde se tiene que posicionar en el momento en el cual necesite empezar a hacer la operación de escritura.

Una vez que asignó los valores al contador de eventos y a la localidad de memoria donde se guarda la dirección de escritura en FLASH y que determino el segmento de datos para la grabación; el programa, inicializa las banderas, los registros, las localidades de memoria; los vectores de interrupción que atienden a cada subrutina y al USART.

Realizado lo anterior, el programa contabiliza 50 segundos para que el usuario realice alguna de las funciones establecidas; si una vez transcurrido el tiempo, el usuario no seleccionó ninguna opción, el programa empieza la captura de datos.

#### **VII.1.1.2 MANEJO DE ACTIVACION Y DESACTIVACION DE LAS INTERRUPTIONES.**

El manejo de la activación y desactivación de las interrupciones, se realiza a través de 2 localidades de memoria, las cuales actúan como banderas. El que se active la bandera contenida en la localidad que atiende la captura de los datos provenientes del conversor A/D ó la bandera que atiende la captura de los digitos del reloj, depende de la subrutina que se está ejecutando en ese momento.

El propósito de utilizar banderas para activar o desactivar interrupciones es el de evitar que se pueda duplicar información como es el caso de los dígitos del reloj o bien que se pueda llegar a perder algún dato que proporcione el conversor A/D.

Antes de que se manejen las señales de activación para cada interrupción, las banderas correspondientes se encuentran con un valor inicial de cero, indicando que aún no se ejecutan las subrutinas que atienden a cada interrupción. Una vez que se envía la señal de activación de las interrupciones y estas empiezan a presentarse, las banderas correspondientes están con un valor diferente al que se les había proporcionado inicialmente, permitiendo de esta manera la ejecución de cada subrutina e impidiendo que se presente otra interrupción cuando aún no ha terminado el proceso que realiza cada una de ellas. Una vez que termina la ejecución de la subrutina, la bandera correspondiente vuelve a estar en condición adecuada para seguir atendiendo la interrupción a la cual esta asignada.

Para que el usuario se percate de que están siendo activadas las interrupciones, en el display se muestra el siguiente letrero: 8 2 1 0.

#### **VII.1.1.3 MANEJO DE LA BASE DE TIEMPO.**

El sistema MEQ-800 digital, proporciona cada segundo un pequeño pulso, lo denominamos Pulso Por Segundo (PPS), que al detectarse cuando se realiza la lectura del puerto que se tiene asignado para este fin; permite la adquisición de los dígitos del reloj, los cuales se presentan de la siguiente forma: 2 dígitos corresponden a la hora, 2 a los minutos y 2 más a los segundos; como el tiempo en el cual permanecen los datos en el puerto es

muy breve, la captura de los dígitos debe ser muy rápida. Los datos del reloj se guardan en 6 localidades de la memoria RAM.

Una vez que se adquirió el reloj, se procede a intercalar los dígitos en la parte alta de la palabra más significativa del conversor, que como se sabe, tienen un valor de cero. Esto se realiza en las primeras 6 muestras posteriores a la muestra que marca la señal del pulso por segundo (PPS).

Durante este proceso, el usuario solo observa en el display el siguiente mensaje: 8 2 1 0, ya que el tiempo en el cual se adquieren los datos, se formatean y se intercalan en las muestras respectivas es muy rápido y no se alcanza a percibir cambio alguno en el display.

#### **VII.1.1.4 COMUNICACION CON EL USUARIO MEDIANTE TECLADO Y DISPLAY DE CRISTAL LIQUIDO.**

Durante todo el proceso, el sistema mantiene contacto con el usuario utilizando el display y el teclado. Con el primero, el programa le indica al usuario las partes que se van ejecutando del programa principal; por ejemplo, la atención de las subrutinas de interrupción, el momento en el cual se presenta el disparo, la captura del postevento, el proceso de borrado de algún segmento de las memorias de almacenamiento, la transmisión del evento, así como también el inicio de la comunicación bidireccional con la PC.

En el display se presentan dígitos y/o letras, las cuales corresponden a cada uno de las instrucciones que ejecuta el programa principal. Es importante mencionar, que en el display



solamente contamos con 4 posiciones para desplegar algún tipo de información.

A continuación se muestra una lista del dígito que aparece para cada proceso, la posición que ocupa en el display y la instrucción que realiza conforme se va ejecutando el programa principal; cabe hacer mención que, el listado involucra al programa principal y a las subrutinas de interrupciones.

DIGITO	POSICION	FUNCION
A	3	Indica al usuario que
A	0	entro el vector de reset.
C	1	Indica al usuario que
C	2	cuenta con 50 seg. para ejecutar alguna operación definida en el programa principal ó bien para que empiece el programa la captura de datos.
3	3	Inicialización de condiciones de operación.
8	3	Activación de
1	1	interrupciones NMI.
5	0	Desactivación de interrupciones para la captura del reloj.
7	2	Indica que existe un

DISEÑO Y DESARROLLO DEL SOFTWARE

		evento y esta grabando información en la memoria FLASH.
9	0	Verifica la señal de disparo para continuar con el proceso de almacenamiento.
F	0	Manda en las cuatro
F	1	posiciones el dígito,
F	2	indicando fin de
F	3	captura de evento.
7	2	Indica que entro a capturar el postevento.
5	0	Muestra la terminación
5	1	de la captura del
5	2	postevento.
5	3	
1	0	
1	1	Indica que entra a la
1	2	opción de transmisión
1	3	del evento.
E	0	
E	1	Manda mensaje de que
E	2	esta transmitiendo el
E	3	evento seleccionado.
F	0	
F	1	Indica fin de
F	2	transmisión.
F	3	
B	0	Entra a la opción de

DISEÑO Y DESARROLLO DEL SOFTWARE

B	2	borrado.
B	1	Después de que se le
B	3	proporciona el segmento a borrar.
F	0	Al poner todas las
F	2	localidades de memoria en cero.
F	1	Después de que finaliza
F	3	la operación de borrado.
E	0	Entra a escribir los datos en FLASH.
A	0	Mensaje de saturación
A	1	de las memorias FLASH
A	2	en la escritura de
A	3	datos.
F	1	Escribe información en la tabla de datos en la memoria FLASH.
2	2	Entra a la subrutina de atención a las interrupciones NMI. (Adquisición de datos del convertor A/D).
E	0	Indica que llegó al tope máximo del segmento de datos en RAM y va a irse al inicio de este segmento.
D	1	Muestra que acaba de

hacer cambio de  
segmento en la  
transmisión de datos de  
memoria hacia PC.

C	1
D	2

Indica que se cometió  
un error al introducir  
el número de evento que  
se desea transmitir y  
que es necesario volver  
a capturar el número de  
evento seleccionado  
previamente.

El teclado es la vía mediante la cual el usuario puede especificarle al programa de control la función que desea se ejecute.

Así tenemos que: si se presiona la tecla 1 y después la tecla D, el programa interpreta que tiene que mandar al display el número de eventos que tiene almacenados; si se presiona la tecla B, el programa interpreta que se desea realizar el borrado de un segmento de la memoria, por lo que espera recibir el número del segmento a borrar; si se presiona la tecla C se establece de forma manual la primera fase de la comunicación bidireccional, por lo que, el programa espera el carácter que empiece la comunicación entre la PC y el adquisidor.

#### **VII.1.1.5 SELECCION Y TRANSMISION DE UN EVENTO HACIA PC.**

Una de las tareas que puede realizar el programa principal es la de Seleccionar y Transmitir un evento.

Al pulsar las teclas 1 y D, el programa principal muestra al usuario, mediante el display, el número de eventos que tiene almacenados; posteriormente, si el usuario desea transmitir algún evento hacia PC, bastará con proporcionarle al programa principal el número de evento que se desea transmitir para que el programa lo seleccione y lo transmita. Cabe hacer mención que, para seleccionar el evento a través del teclado será necesario introducir 2 dígitos y presionar la tecla de MENU. Por ejemplo si se desea el evento 1, se le proporcionan las teclas 0,1 y posteriormente la tecla MENU. En caso de que el número del evento seleccionado no se encuentre en la tabla de datos, el programa manda al display en sus 4 posiciones el dígito 1, para indicarle al usuario que tiene que introducir otro número de evento, ya que el que proporcionado anteriormente no se encuentra registrado en la tabla de datos. Por otra parte si se comete un error al introducir el número del evento que se desea transmitir, el programa envía al display las letras D y C en las posiciones 2 y 1 respectivamente, para indicarle al usuario que se cometió un error en la captura de los dígitos del evento y que nuevamente tendrá que introducir el número del evento que desea transmitir.

Si no se desea transmitir ningún evento, es posible salir de esta opción y seguir con la ejecución del programa principal.

#### **VII.1.1.6 MANEJO DE LA TABLA RESUMEN.**

Para el mejor manejo de la información almacenada en memorias FLASH, se cuenta con una Tabla de Eventos, a partir de

**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**

**DISEÑO Y DESARROLLO DEL SOFTWARE**

la localidad 10000H la cual proporciona información de cada evento.

Los datos que se almacenan en la Tabla Resumen y que son para cada evento se muestran a continuación:

- Dirección inicial del evento.
- Dirección final del evento.
- Reloj inicial del evento.
- Máxima amplitud alcanzada por la señal correspondiente al evento.

De esta manera, el usuario con ir a la Tabla, puede obtener información específica de cada evento. Además, en base a esta tabla podemos obtener información adicional, como:

- Número de eventos almacenados.
- Condiciones de operación, en el caso de que por un período de tiempo, no se haya tenido energía eléctrica.

La escritura de información en esta tabla se inicia con la verificación de que esta activa la señal de disparo, y se finaliza una vez que termina de capturarse el postevento; el usuario no se percata de lo que esta ocurriendo debido a que el proceso de captura del evento, absorbe el proceso de escritura en la tabla.

Es importante mencionar, la actualización de la tabla de datos se realiza cada vez que se captura un evento.

**VII.1.1.7 MANEJO DE PREEVENTO Y POSTEVENTO.**

Para el manejo del preevento se realiza un procedimiento de almacenamiento temporal en memoria RAM cuando no ha sido detectada la señal de disparo.

Todos los datos provenientes del sismógrafo, antes de

ser grabados en las memorias FLASH, se encuentran en una sección reservada en RAM; esta sección tiene reservados 4 Kbytes. El manejo de esta sección en RAM se realiza mediante un contador, el cual permite verificar si no se ha rebasado la capacidad de almacenamiento asignada. Si detecta que la última localidad de almacenamiento que se ha reservado esta siendo ocupada, entonces, tiene que apuntar al inicio de la sección reservada para datos y continuar con este proceso. De tal forma que cuando el algoritmo detecta que la señal de disparo se encuentra activada, baja la información contenida en RAM a FLASH, almacenando el nuevo dato proveniente del conversor A/D en la localidad que acaba de ser dejada libre en RAM.

Este proceso se realiza durante el preevento, en la captura del evento y 30 segundos posteriores al fin del mismo. Una vez, que se terminó de almacenar la información del evento en la memoria FLASH y que se verificó que el disparo ya no se encuentra activado, los datos que proporciona el conversor se continúan almacenando en memoria RAM hasta que de nuevo se produzca otro evento.

En cuanto se detecta la señal de disparo, se empieza a grabar en la memoria FLASH, la información correspondiente a los 20 segundos previos al instante en que se detecto el evento, la cual se le denomina PREEVENTO. Cuando se deja de detectar la señal de disparo, mediante un contador de segundos, se adquieren 30 segundos adicionales de información proporcionados por el conversor A/D; a este conjunto de datos se le denomina POSTEVENTO; por lo tanto un sismo almacenado queda compuesto por:

20 seg. de PREEVENTO / EVENTO / 30 seg. de POSTEVENTO

De tal forma, que el usuario podrá percatarse de la ejecución de este proceso, ya que en el display en la posición 0 aparece la letra E, cuando el programa vuelve al inicio del segmento de datos.

#### **VII.1.1.8 BORRADO DE MEMORIAS FLASH.**

De las teclas utilizadas como medio de selección de operación del sistema se encuentra la que esta relacionada con el borrado de las memorias FLASH.

Las memorias FLASH utilizadas en este sistema son del tipo 28F020, las cuales se caracterizan por tener una densidad de 256 Kbytes.

Para borrar la FLASH, es necesario controlar tanto los voltajes de programación como el registro de comando del modo de operación que la memoria posee.

Es importante mencionar que, para realizar este procedimiento, es necesario seguir paso a paso el algoritmo que proporciona el fabricante con el fin de evitar el funcionamiento erróneo de las memorias FLASH. (El algoritmo de borrado que proporciona el fabricante, se encuentra en el Apéndice B).

El procedimiento de borrado se inicia con la colocación de una B en los dígitos 3 y 1. A partir de ese momento se espera que se presione la tecla 0, para proceder a colocar una B en los dígitos 2 y 0 y esperar por la selección de una tecla diferente de 0, que indique el segmento que se desea borrar. Al terminar el proceso de borrado el microprocesador pone en las 4 posiciones del display, una F y espera por la selección de una nueva tarea a ejecutar. Una vez que al sistema se le proporciona el número



de segmento que se desea borrar ya no se puede parar este proceso a menos que se presione el RESET, lo que equivale a reinicializar al adquisidor.

#### **VII.1.1.9 COMUNICACION CON LA COMPUTADORA.**

##### **VII.1.1.9.1 CONTROL Y PROGRAMACION DEL PUERTO SERIE.**

Para poder establecer la comunicación bidireccional entre el Sismógrafo MEQ-800 y la PC es necesario establecer un procedimiento de inicialización del USART, el cual se menciona a continuación:

1). **Inicializar USART.** Para lograr ésto, mediante el programa se limpia el registro de control del USART tres veces, utilizando palabras de modo de instrucción y comando de instrucción. En este caso se carga una secuencia de ceros, enviándolos al puerto correspondiente.

2). **Borrar registro de datos del USART.** Realizado lo anterior, es necesario asignar el valor 40H al comando de borrado interno, con el fin de que no deje ningún valor en el registro de datos.

3). **Indicar el modo de instrucción en que va a operar el USART.** El modo de instrucción especifica si el USART va a trabajar en forma síncrona ó asíncrona y si se va a dividir la frecuencia del reloj entre 1, 16 ó 64, de tal forma que se pueda operar a velocidad de 600,2400 y 38400 bauds. La instrucción de modo también indica la longitud del carácter, el número de bits de parada y el tipo de paridad, si es que se va a utilizar. En este caso se cargó un 4EH, de tal forma que se tuviera: 1 bit de inicio, un carácter de 8 bits de longitud, no paridad, 1 bit de parada y una velocidad de 2400 bauds.

4). Indicar el Comando de Instrucción. Esta instrucción define la forma en la cual va a trabajar el USART, ya sea de modo simplex ó duplex.

Una vez, que ya se ejecutaron los pasos anteriores, el USART esta listo para trabajar en la forma en la cual el programador desee.

#### VII.1.1.9.2 PROTOCOLO DE COMUNICACION CON LA COMPUTADORA.

Para lograr la comunicación entre la PC y el Sismógrafo MEQ-800, fue necesario plantear un protocolo que se apegara a las necesidades de ambas partes.

Es por ello que se planteó el uso de caracteres estándar, como es el %, y de caracteres que definieran cada una de las opciones que tiene implementado el sistema desarrollado para el MEQ-800.

Para iniciar la comunicación entre la PC y el Sismógrafo es necesario indicarle a ambas partes que se desea establecer comunicación entre si y que además necesitan inicializar los registros necesarios para la comunicación. Es por ello que en el protocolo planteado la PC envía %I, como forma de indicar al equipo, que se quiere establecer comunicación con él. Si el sismógrafo capturo bien estos caracteres, retorna éstos a la PC para indicarle que está listo para empezar la comunicación con ella.

Cuando ya no se desea mantener la comunicación bidireccional, o ha pasado cierto tiempo sin que el adquisidor reciba alguna de las opciones establecidas en el protocolo, él envía un %F indicando que finalizo la comunicación.

Las opciones dentro del protocolo de comunicación se definieron de la siguiente manera:

**%I** Para iniciar el protocolo de comunicación.

**%F** Indica el fin del protocolo de comunicación.

**%B** Cambio de velocidad de transmisión. Al seleccionar el usuario esta opción, el equipo recibe y envía un **%B** para indicar que esta lista para hacer el cambio de velocidad. Posteriormente, el equipo espera un número que le indique el valor de la velocidad de transmisión que desea, una vez hecho esto, y ya que cambio la velocidad, el sismógrafo envía a PC un **%** indicando que realizo el cambio de velocidad y que está en espera de un nuevo comando.

**%M** Proporciona el número de eventos que tiene almacenados la tabla de datos. Una vez que el sistema recibe estos caracteres, envía a la PC el número de eventos que tiene almacenados en FLASH. Para finalizar esta opción, el sistema envía a la PC un **%N**, indicando que ya terminó de hacer la operación indicada.

**%T** Visualizar la tabla de datos. Estos caracteres le indican al sistema que tiene que enviar los datos correspondientes a la hora, minuto, segundo y máximo absoluto de cada evento. Cuando termina de enviar todos los datos, el adquisidor manda los caracteres FFFF, con el objeto de evitar una posible combinación de datos, que cortara la comunicación.

**%L** Lectura de un evento específico. Cuando el sistema detecta que se le enviaron estos caracteres, sabe que tiene que acudir a la rutina de envío de datos. El siguiente paso consiste en proporcionar el número de evento que se desea transmitir, para que el programa localice en la tabla de datos la dirección de inicio y la dirección final del evento y así lo

pueda transmitir.

**%D** Borrado de un Segmento de la memoria FLASH. Estos caracteres indican al sistema que se desea borrar un segmento de la memoria. Para lograr lo anterior, basta con proporcionar el número de segmento a borrar. Una vez que ya finalizó el proceso, el sistema le enviara de regreso a la PC el **%D** que había mandado al inicio de esta opción.

En algunos casos el Sismógrafo envía los mismos caracteres que le envió la PC, esto con el fin de indicarle a la PC que ya realizó la opción que el usuario había pedido y que ya envió la información deseada.

#### VII.1.1.9.3 CONDICIONES DE ERROR.

Como el sistema está esperando un carácter predefinido para cada una de las opciones anteriores, constantemente se está revisando que alguna de ellas se cumpla. Si no se cumplió ninguna de ellas, entonces el programa da por entendido que no se desea establecer la comunicación con el Sismógrafo y envía un **%F** para indicar que finalizó la comunicación con la PC y que va a seguir adquiriendo datos del conversor A/D para almacenarlos en la sección de RAM reservada para ello.

El diagrama de flujo correspondiente a las diversas tareas que realiza el programa principal se muestran en las figuras 7.1, 7.1.A y 7.1.B.

**VII.1.2 SUBROUTINA DE ATENCION A INTERRUPCIONES NMI.****VII.1.2.1 ADQUISICION DE DATOS DEL CONVERSOR A/D.**

La tarea principal que realiza la subrutina es la de adquirir los datos que provienen del conversor A/D. Esta subrutina se encuentra entre las localidades 1000H y 1450H especificadas para la memoria EPROM.

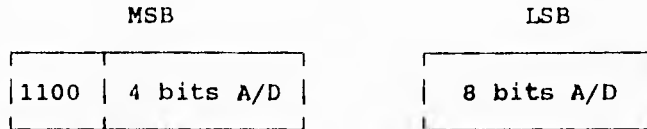
Para que se pueda ejecutar esta subrutina, el microprocesador debe cargar en las direcciones 0008H y 000AH, del segmento cero, la dirección donde se encuentran las subrutinas.

La subrutina de adquisición de datos del conversor A/D se empieza a ejecutar cuando se presenta un pulso en la entrada NMI, proveniente de la salida de indicación de conversión completa del conversor.

Lo primero que realiza la subrutina es deshabilitar la llegada de otra interrupción NMI para poder adquirir la información que proviene del conversor A/D.

En el transcurso de la ejecución del programa, se revisa la bandera correspondiente al pulso por segundo; si esta bandera no está activada se sigue el proceso de adquisición de datos, en el momento en que se detecta que la bandera del pulso por segundo está activada, entonces se reemplazan los 4 bits más significativos del dato de 2 bytes provenientes del conversor por los 4 bits correspondientes a un dígito del reloj del MEQ-800, de tal forma, que dentro de los datos del conversor, se encuentre la información del reloj.

Cuando existe una interrupción en la cual se detecto el pulso por segundo se marca la muestra donde se hizo la detección, tal y como se muestra a continuación:



En la figura 7.2 se encuentra el diagrama de flujo correspondiente a esta subrutina y en la figura 7.3 aparece una sección de la información leída de la memoria FLASH en donde aparecen los datos del conversor, la muestra en donde esta marcado el pulso por segundo y las muestras donde se localizan los dígitos del reloj que proporciona el MEQ-800. La marca del PPS y los dígitos capturados se encuentra subrayada. Este archivo es para cuando se tienen 120 muestras por segundo.

#### VII.1.2.2 ADQUISICION DE LOS DIGITOS DEL RELOJ.

En el momento en que en la subrutina de captura de los datos que provienen del conversor A/D detecta la llegada del pulso por segundo, modifica el valor de la bandera asignada para el pulso por segundo, de tal forma que cuando se regrese de la interrupción se pueda preguntar a continuación por la bandera del reloj. Si no se ha activado esta bandera, la captura de datos sigue de manera normal, en caso de que el valor de esta bandera se haya modificado, se realiza el cambio del vector de interrupciones, de tal forma que pueda realizarse la captura de los dígitos del reloj. Posteriormente, se activan las interrupciones y se apunta a las localidades de RAM donde se guardan temporalmente los dígitos del reloj. La subrutina lo único que hace es leer el puerto por donde arriban los dígitos

y almacenarlos en las localidades definidas previamente (0700H a la 0705H de RAM localizada en el segmento cero).

Como el tiempo que permanecen los digitos en el puerto es muy breve, la subrutina los adquiere tal y como se presentan en el puerto, para darles posteriormente el formato correspondiente.

La figura 7.4 presenta el diagrama relacionado con esta sección.

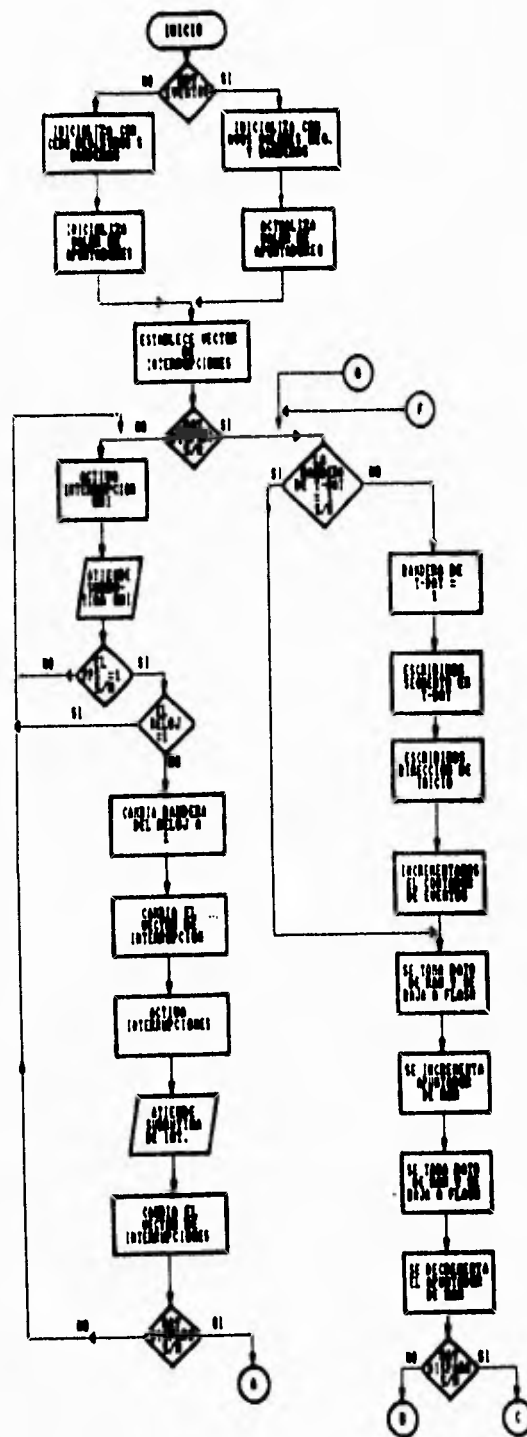


FIGURA 7.1 PROGRAMA PRINCIPAL DE LA SECCION DEL REGISTRADOR DEL MEQ-800.



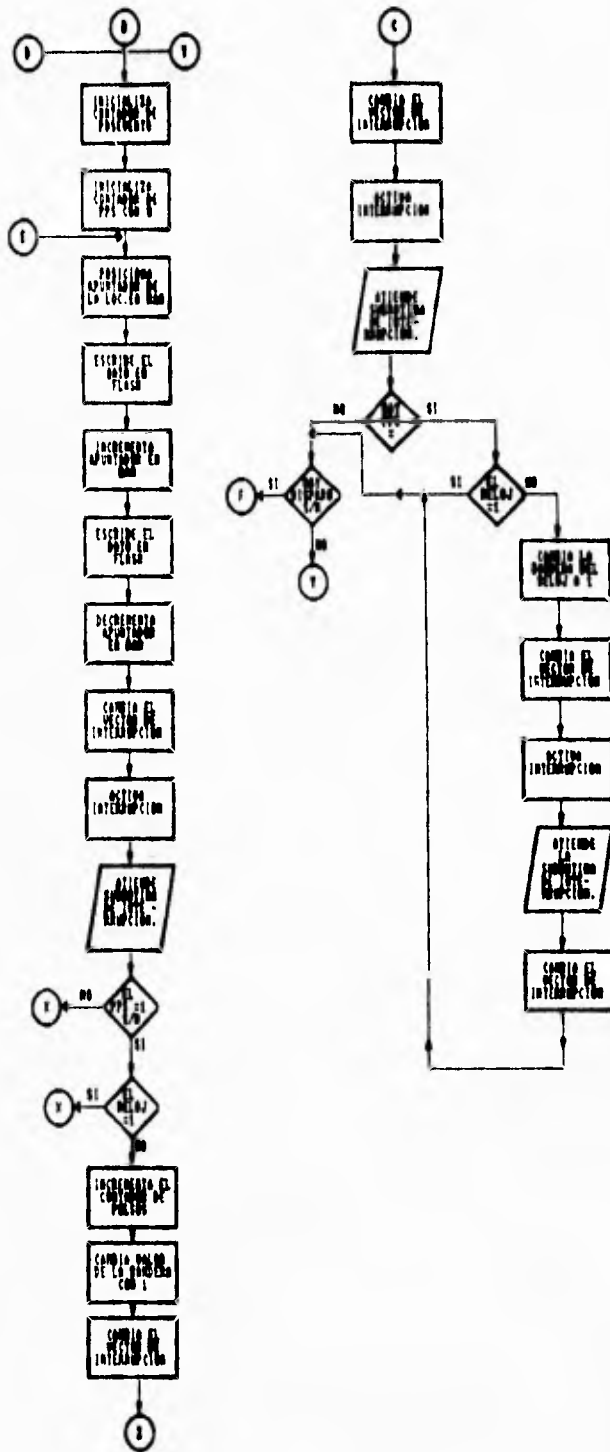


FIGURA 7.1.A CONTINUACION DEL PROGRAMA PRINCIPAL.

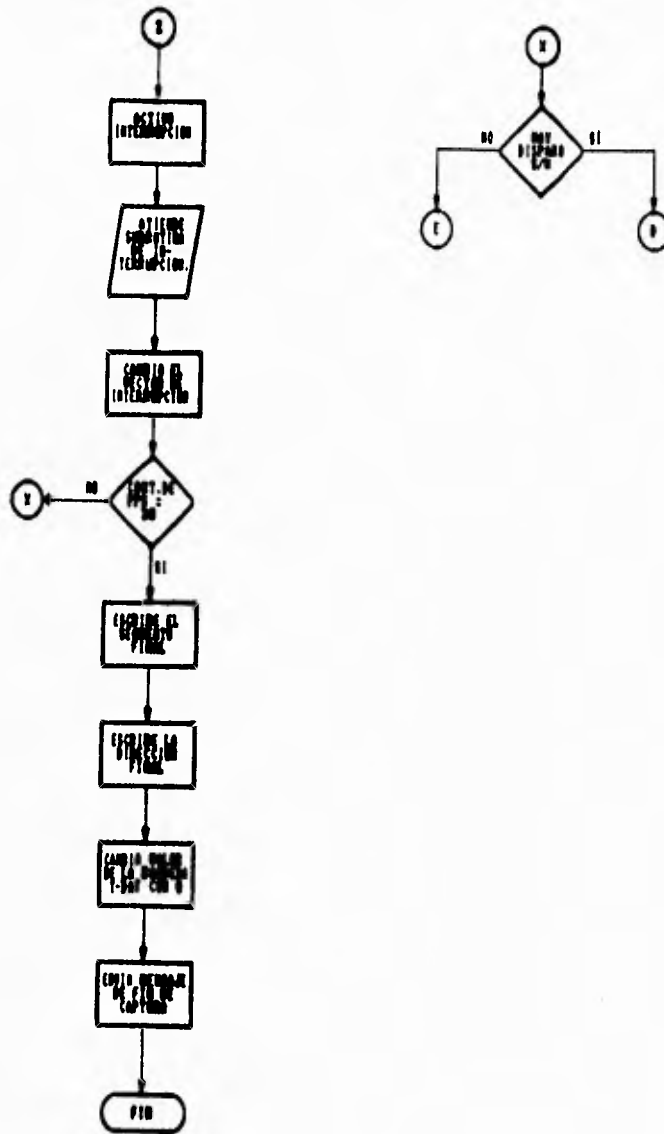
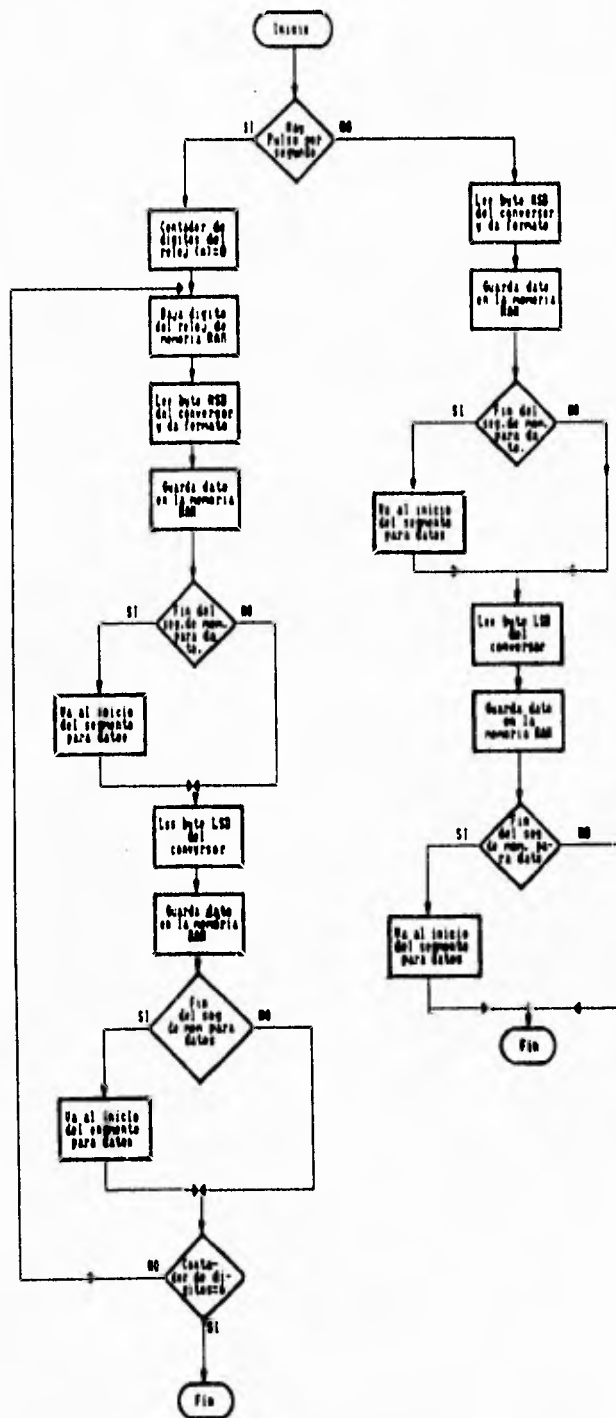


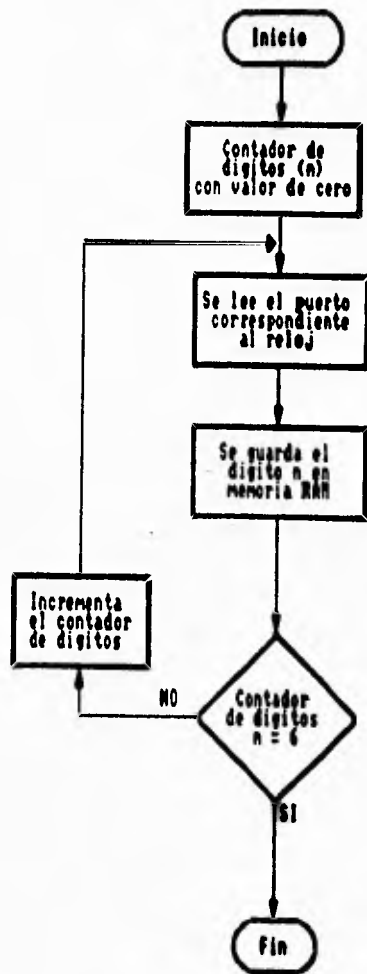
FIGURA 7.1.B CONTINUACION DEL PROGRAMA PRINCIPAL.



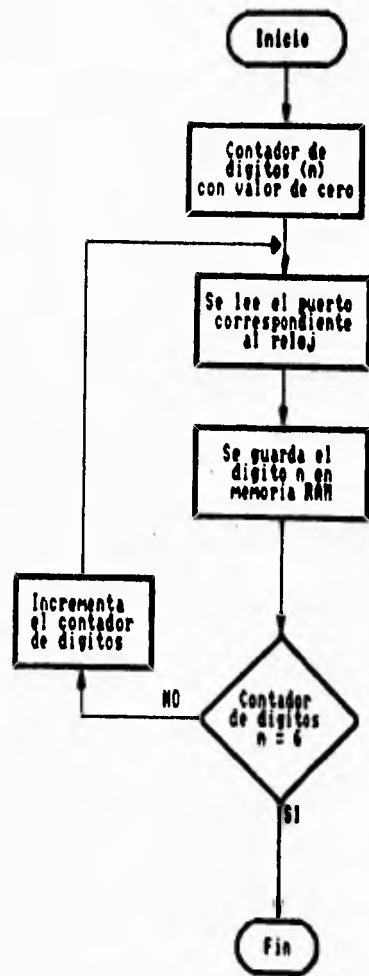
**FIGURA 7.2 SUBROUTINA NMI DE ADQUISICION DE DATOS DEL CONVERTOR A/D.**

000000	07	AA	07	AB	07	AD	07	AF	07	B1	07	B4	07	B4	07	B5
000010	07	AD	07	A9	07	AB	07	A4	07	A4	07	A5	07	A9	07	AC
000020	07	AB	07	AD	07	AC	07	AA	07	A7	07	A7	07	A7	07	AB
000030	<u>C7</u>	<u>A7</u>	<u>27</u>	<u>AA</u>	<u>37</u>	<u>A9</u>	<u>17</u>	<u>A7</u>	<u>57</u>	<u>AB</u>	<u>57</u>	<u>AC</u>	<u>17</u>	<u>AB</u>	07	AF
000040	07	AF	07	B3	07	B3	07	B1	07	B1	07	AD	07	AF	07	B1
000050	07	B0	07	B1	07	AF	07	AD	07	AD	07	AC	07	AB	07	A5
000060	07	A4	07	A3	07	A3	07	A6	07	AB	07	A9	07	A7	07	A9
000070	07	AB	07	AD	07	B1	07	B1	07	B2	07	B0	07	AE	07	AE
000080	07	AD	07	AA	07	AB	07	AD	07	AF	07	B3	07	B0	07	AF
000090	07	AC	07	A6	07	A7	07	AE	07	AF	07	B1	07	AE	07	AB
0000A0	07	A9	07	AB	07	A9	07	AC	07	AD	07	AD	07	AE	07	B1
0000B0	07	AE	07	AB	07	AB	07	A7	07	A6	07	A7	07	AB	07	A7
0000C0	07	AA	07	A9	07	AD	07	AB	07	AB	07	A7	07	AB	07	AA
0000D0	07	AC	07	AB	07	AD	07	AE	07	AF	07	B1	07	B3	07	B1
0000E0	07	B3	07	B3	07	B2	07	B1	07	B0	07	B0	07	B0	07	AC
0000F0	07	<u>AA</u>	<u>C7</u>	<u>AA</u>	<u>27</u>	<u>AA</u>	<u>37</u>	<u>AD</u>	<u>17</u>	<u>AF</u>	<u>57</u>	<u>AD</u>	<u>57</u>	<u>AC</u>	<u>27</u>	<u>A7</u>
000100	07	AB	07	A9	07	A4	07	A3	07	A5	07	9F	07	A6	07	A4
000110	07	A6	07	A7	07	A7	07	A9	07	A9	07	AC	07	B3	07	B3
000120	07	B4	07	B2	07	B1	07	B0	07	B2	07	AF	07	B0	07	B0

FIGURA 7.3 FORMA DE ALMACENAR LOS DATOS DEL SENSOR Y DEL TIEMPO (SUBRAYADO) EN MEMORIAS FLASH.



**FIGURA 7.4 SUBROUTINA DE ADQUISICION DE LOS DIGITOS DEL RELOJ.**



**FIGURA 7.4 SUBROUTINA DE ADQUISICION DE LOS DIGITOS DEL RELOJ.**

## VII.2 PROGRAMACIÓN DE LA COMPUTADORA PARA PROCESAR LOS DATOS ADQUIRIDOS

El programa diseñado para interactuar con el adquisidor digital adaptado al MEQ-800, fue codificado en lenguaje C, y su preparación para ejecución se realizó mediante el compilador de Borland, versión 2.0.

El código del programa resultó ser demasiado largo, para entrar en un sólo archivo. La solución a este problema fue romper el programa en partes más pequeñas y compilar cada parte y enlazarlas juntas. Este proceso se conoce como **compilación separada y enlazamiento**, y es la columna vertebral del programa desarrollado.

En el entorno integrado de desarrollo de Borland 2.0, un programa con archivos múltiples es llamado un **archivo de proyecto**. Para crear un programa proyecto, primero se asigna un nombre con extensión .PRJ. Una vez que se ha dado el nombre, se le indica al compilador los nombres de los archivos que forman el proyecto. Cuando hay un archivo con extensión .PRJ especificado en la opción **project** del compilador, se usa como guía para que Borlandc 2.0 compile el programa. Se lee el contenido del archivo con extensión .PRJ y cada archivo que lo integra se compila a uno con extensión .OBJ. A continuación, se enlazan estos archivos, se genera un archivo con extensión .EXE y se ejecuta el programa.

Una de las ventajas de esta forma de programación es que sólo se compilan los archivos que realmente lo necesitan y nos ahorramos tiempo de compilación.

El archivo proyecto que se desarrolló con el método descrito anteriormente lleva por nombre **TRANSER.PRJ** y lo integran 4 archivos. A continuación se describen las principales funciones

que tiene cada archivo.

El **archivo 1** llamado **"PRINCIPA"** contiene:

- Los archivos de cabecera necesarios para que la compilación de este archivo sea correcta. Posteriormente se definen las constantes, las variables y los arreglos dinámicos globales que serán utilizados por los demás archivos.
  - La función principal(main).
  - Funciones que inicializan el puerto serie de comunicación.
  - La función para capturar información del puerto serie.
  - La función que envía información por el puerto serie.
  - La función que guarda en memoria RAM la información capturada en el puerto serie.
  - La función de detección y corrección de errores de la información almacenada en RAM.
  - La función que salva la información corregida en formato ASCII.
- 
- Diversas funciones que generan pantallas para interactuar con el usuario e indicarle cuando se generan condiciones de error.

Además, en este archivo se incluyen funciones para manipular y guardar en formato binario la información proveniente del sismógrafo DR-100; hay que aclarar que no se comentará nada de estas funciones, pues no corresponden a los objetivos de este trabajo.

El **archivo 2** llamado **"PROTOCOL"** incluye:

- Las funciones de cabecera necesarias para la compilación correcta de este archivo.
- Funciones para tener un protocolo de comunicación bidireccional con el adquisidor del sismógrafo MEQ-800.
- Funciones que presentan pantallas al usuario para el manejo de las opciones del protocolo.



- Funciones que indican al usuario cuando se produjo un error al operar alguna función del protocolo.

El archivo 3 llamado "DECIBEL" contiene:

- Los archivos de cabecera necesarios para que compile correctamente este archivo.
- Las rutinas que presentan 2 menús gráficos:
  - 1) menú de selección de decibeles y 2) menú para escoger la constante electromotriz que corresponde al tipo de sismómetro utilizado, para la operación del sismógrafo MEQ-800.

El archivo 4 llamado "GRAFICA" contiene:

- Los archivos de cabecera, pero además, se definen al inicio del archivo las constantes externas que están definidas en el archivo PRINCIPA.
- La función que inicializa en modo gráfico el video.
- La función que lee los datos del evento y los gráfica. Los datos se pueden leer del archivo temporal (contiene datos corregidos de la memoria) ó de un archivo ASCII.
- Las funciones que acotan la gráfica en tiempo y amplitud.
- Las funciones que dibujan ventanas en el borde inferior de la gráfica, con la finalidad de presentar comandos e información del evento al usuario.

Una parte muy importante del trabajo desarrollada en el programa, fue el diseño e implementación de pantallas, ya que es dónde se establece el nivel de comunicación entre el usuario y el sistema. Las pantallas de cada módulo se pensaron para que el usuario pueda manejarlos fácilmente y además para presentar de forma agradable los resultados que produzca el sistema. En este capítulo se podrá observar las pantallas principales que se

desarrollaron para este sistema.

### **VII.2.1 Archivo de cabecera diseñado y desarrollado especialmente para el programa.**

Muchas de las rutinas que se implementan en un programa trabajan con sus propios tipos de datos y variables a las que el programa debe acceder. Estas variables y tipos están definidos en los **archivos de cabecera**. Los archivos de cabecera se proporcionan junto con el compilador y deben incluirse (utilizando #include) cuando el programa utilice tipos de datos definidos en ellos. El compilador identifica a los archivos de cabecera porque tienen extensión .h.

Para el programa **TRANSER** se tuvo que crear un archivo de cabecera, llamado **SERIAL.H**. En el archivo SERIAL.H se definieron constantes que son indispensables para manipular cada uno de los 10 registros del 8250 (USART). Además, se describe cómo funciona cada registro. También se definen los procedimientos que se utilizan en la ejecución del programa.

### **VII.2.2 Rutina principal**

En la rutina principal se implementó la secuencia lógica que debe llevarse a cabo para llamar a los procedimientos, con el fin de que cumpla con el diseño conceptual definido en el capítulo 6 ( ver la sección VI...).

Esta rutina se puede considerar un procedimiento más en el programa, sin embargo, el compilador identifica que es el procedimiento principal porque tiene el nombre **main**.

La rutina principal del programa **TRANSER** da primero los valores a los parámetros para inicializar al USART. Estos parámetros fueron definidos como variables. A continuación se

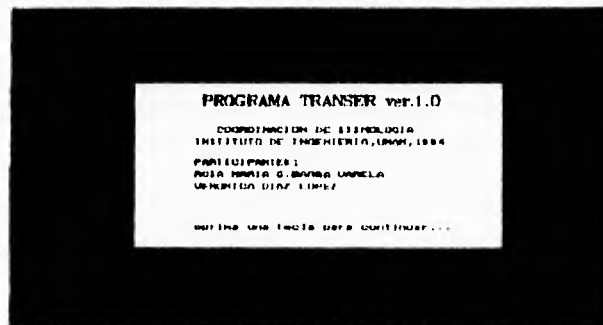
muestra el nombre y el valor que adquieren esas variables:

**variables de tipo entero:** ptocon=0, paridad=0, bitsdatos=8, stopbits=1.

**variable de tipo entero largo:** baud=2400 (es la velocidad de transmisión)

Asimismo, se inicializan las banderas de tipo entero que se utilizarán durante la ejecución del programa.

El primer procedimiento ejecutado es **princi**, el cual se encarga de mostrar al usuario la siguiente pantalla:



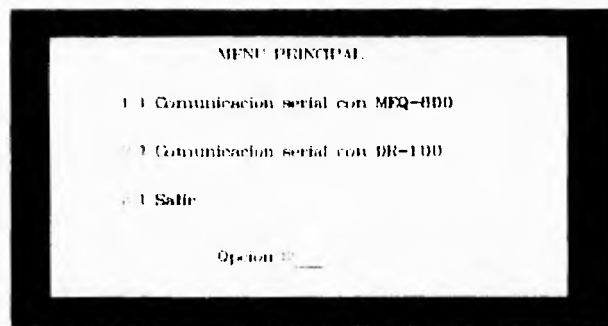
**FIGURA 7.5** Pantalla de presentación del programa TRANSER.

Cuando el usuario presiona una tecla, se muestra entonces

la pantalla que cumple la función de menú principal del programa. En la pantalla de la figura 7.6 se puede observar que se solicita al usuario elija el aparato, con el cual desea comunicarse para obtener eventos. El procedimiento **princi** espera la entrada de información con la función **getch()** y la guarda en una variable de tipo **char** llamada **var** que se pasa a la siguiente estructura:

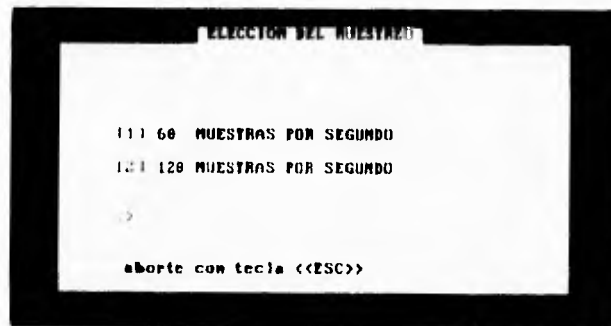
```
switch(var){  
case '0': /* realiza llamadas a procedimientos para interactuar  
con el sismógrafo MEQ-800.*/  
case '1': /* llama a los procedimientos para interactuar con el  
sismógrafo DR-100.*/  
case '2': /* sale del programa y retorna el control al sistema  
operativo con la función exit(0)*/  
}
```

Es conveniente aclarar, que la opción a describir en el desarrollo de este trabajo, es la correspondiente al sismógrafo MEQ-800; ya que, como se había mencionado anteriormente, el sismógrafo DR-100 no es nuestro objetivo.



**FIGURA 7.6 Menú principal del programa**

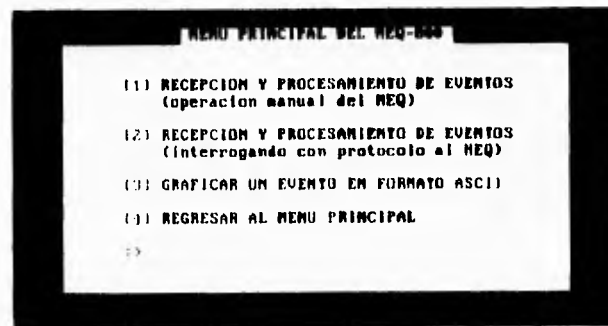
presenta la pantalla que sirve para que el usuario elija el muestreo con el que se procesará la información de los eventos, lo que significa que el usuario tiene que saber en que condiciones estaba operando el MEQ-800. Existen dos posibilidades: 1) en opción de captura y grabación de evento a 60 muestras/seg ó 2) en opción de captura y grabación a 120 muestras/seg. El valor de la selección se guarda en una variable global, la cual se utiliza para realizar cálculos en otras rutinas. Sin embargo, si el usuario oprima la tecla **ESC** la variable tomará el valor de -1, por lo que se regresará al **MENÚ PRINCIPAL**(figura 7.6). En la figura 7.7 se presenta la pantalla para la elección del muestreo.



**FIGURA 7.7** Menú para elección del muestreo.

Si no se oprimió **ESC** en el anterior menú, entonces se

presenta la pantalla de la figura 7.7, en la cual se muestra un menú, con dos opciones diferentes para interactuar con el MEQ-800 y con una opción que permite graficar un archivo ASCII generado en este sistema ó en otro.



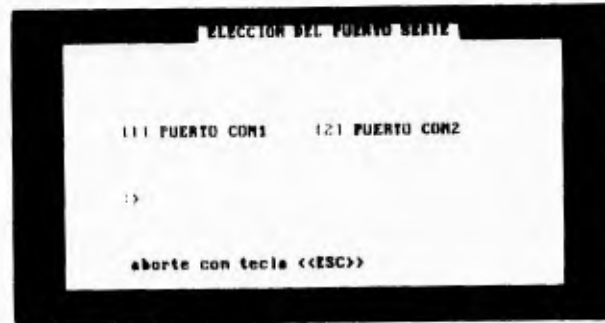
**FIGURA 7.8 .MENÚ PRINCIPAL DEL MEQ-800**

**Selección de la opción [1] del menú principal del MEQ-800 (figura 7.8)**

La opción [1] del **MENÚ PRINCIPAL DEL MEQ-800** significa que el programa manipulará el puerto serial sólo para recepción de datos.

Con ésta opción, se presenta otra pantalla, en la cual el usuario seleccionará el puerto serie por el cual desea recibir datos. Hay que hacer notar que si se da un puerto serie que no esté incluido en la PC, entonces la rutina presenta al usuario un mensaje indicándole su error. Por otro lado, si el usuario da

**ESC** regresará al menú principal del MEQ-800(figura 7.8). A continuación se muestra la pantalla correspondiente para esta selección.



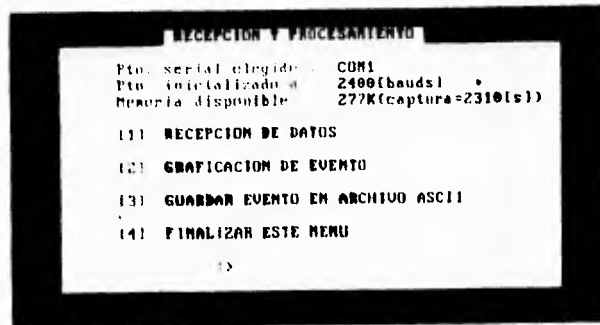
**FIGURA 7.9 Elección del puerto serie.**

En la variable **ptocom** se guarda el valor de la selección del menú anterior, ya que es uno de los parámetros importantes para inicializar el USART.

Inmediatamente después, se manda llamar a la rutina que da valores iniciales al USART; ésta rutina necesita se le pasen las variables de inicialización en la siguiente secuencia: **puerto com,baud(velocidad de transmisión), paridad, longitud del carácter y bits de stop**. La rutina se manda llamar de la siguiente forma:

**SetSerial(ptocom,baud,paridad,bitsdatos,stopbits).**

Después de esto se llama a la función **SUBPRIN()**, la cual se encarga de investigar la cantidad de memoria libre disponible en la PC para el almacenamiento de datos. **SUPPRIN** también llama al proceso que presenta la pantalla de **RECEPCION Y PROCESAMIENTO DE EVENTOS**. La llamada a **SUBPRIN** produce la siguiente pantalla:



**FIGURA 7.10 RECEPCION Y PROCESAMIENTO DE EVENTOS.**

En el menú de la figura 7.10 se puede observar que se informa al usuario cual es el puerto serial por donde se realizará la recepción, a que velocidad llegarán los caracteres y cuanta memoria disponible se tiene en la PC. Se presenta, además, el número de segundos de información que el sismógrafo MEQ-800 puede transmitir a la PC.

Con la selección de la opción [1] del menú de la figura 7.10, se reciben los caracteres transmitidos desde el MEQ y se almacenan en memoria. Cuando finaliza la transmisión, el usuario puede elegir la opción [2] con la cual podrá graficar el evento recién recibido ó un archivo ASCII ya existente. Si se elige la opción [3], entonces se guardará el evento recibido en un archivo ASCII. Si la bandera que sirve para indicar la recepción o no de un evento, está en cero, significa que no hay evento que guardar en un archivo y se le informará al usuario. Si se elige la opción [4] se sale de éste menú y se retorna al menú principal del MEQ-800(figura 7.6). Las pantallas y la explicación detallada de cómo se implemetaron los procesos que se presentan a selección en la



figura 7.10, se puede ver en las secciones siguientes de este capítulo.

**Selección de la opción [2] de el menú principal del MEQ-800 (figura 8).**

Cuando se elije la opción [2] del menú principal del MEQ-800, entonces se puede interrogar al equipo con el protocolo de comunicación implementado. Con ésta opción el puerto de comunicación se manipula para comunicación duplex y asincrona, con lo cual se podrá enviar y recibir caracteres por el puerto serie. Las pantallas y forma de operar del protocolo se presentan en el punto VII.2.3.2 de éste capítulo.

**Selección de la opción [3] de el menú principal del MEQ-800.**

Si el usuario tiene datos en un archivo ASCII y desea obtener la gráfica, debe elegir la opción [3] del menú principal del MEQ-800. En éste caso se llama a la función **nombre**, en dónde se le pide al usuario la ruta y el nombre de archivo (ver figura 7.11). Si no hubo error al leer el archivo, entonces se manda llamar a la subrutina de graficación. Con ella se puede observar una pantalla como la que se muestra en la figura 7.12. Los comandos que se observan debajo de la gráfica se explican con detalle en el sección VII.2.8 de éste capítulo.

**Selección de la opción [4] de el menú principal del MEQ-800.**

Cuando se elije esta opción se retorna al menú dónde se selecciona el sismógrafo (ver figura 7.6). Si se desea salir definitivamente del programa se tiene que dar la opción [3] en el menú de la figura 7.6.

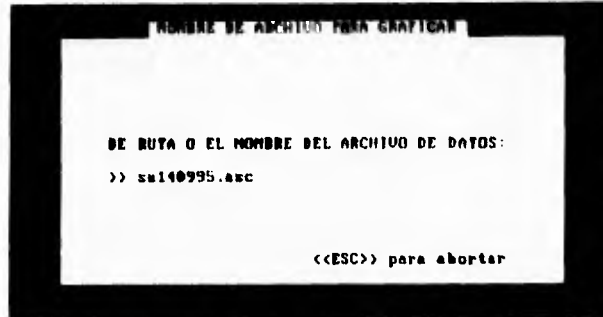


FIGURA 7.11 Pantalla para que el usuario de el nombre ó la ruta de archivo a ser graficado.

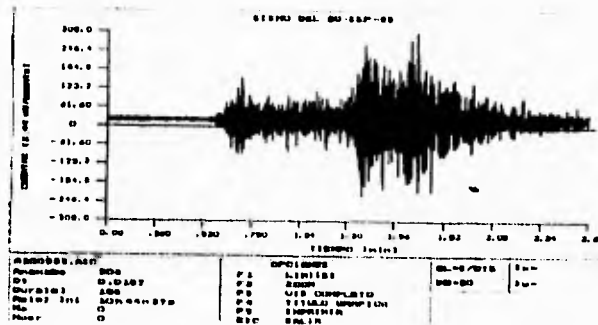


FIGURA 7.12 Pantalla que presenta la gráfica de un evento.

### VII.2.3. Comunicación con el adquisidor

El circuito integrado que en la computadora lleva a cabo la función de comunicación es el 8250(USART). La operación básica de éste circuito programable es la recepción y transmisión

universal para comunicación síncrona y asíncrona de datos.

El USART necesita ser inicializado para que acondicione los caracteres tanto para la transmisión como para la recepción.

El tipo de comunicación que se requiere establecer con el adquisidor es del tipo duplex y asíncrona. Al usar éste tipo de transmisión se pueden enviar y recibir caracteres en forma individual en intervalos de tiempo aleatorio. Los parámetros fijos que se tienen que dar al USART son: bits de control (1 bit de inicio, bit que indica que "no existe paridad" y 1 bit de paro) y 8 bits correspondientes al carácter. El parámetro para el puerto COM es variable pues el usuario puede elegir el puerto. El parámetro que da la velocidad de transmisión es fijo cuando no se utiliza el protocolo para interrogar al equipo y toma desde el inicio del programa el valor de 2400 bauds. Sin embargo, si se realiza el protocolo el usuario puede seleccionar 600, 2400 ó 38400 bauds.

En el programa se realizaron rutinas que se encargan de escribir las palabras para establecer la comunicación con el 8250 (USART), tal y como se haría utilizando el lenguaje ensamblador. Para tener acceso a cada registro, se determina la dirección relativa del mismo, tomando como base la dirección del puerto COM.

A continuación se da una breve descripción de las actividades que controlan las subrutinas del programa que permiten la comunicación:

- **SetSerial**: Es la primera subrutina a llamar y permite establecer la variable global "ptobase" y verificar si existe algún error en los parámetros que se pasaron desde el programa principal. Ésta se encarga de llamar a tres funciones adicionales que ejecutan realmente la tarea. Cada una de las funciones regresa un valor de -1 en caso de error.

- La función **SetPort** pasa el valor que eligió el usuario y establece la variable global ptobase.

- La función **SetBaud** permite establecer la velocidad de transmisión, en el rango de 1 a 115200. Esto lo logra mediante la determinación de un valor válido de velocidad y después dividiendo esta velocidad entre 115200L para crear un divisor que será utilizado por el 8250 para establecer dicha velocidad. En esta misma función se lee el valor que se tiene en el registro de control de línea (LCR) y se hace una OR bit a bit de dicho valor con 0x80 para establecer el bit que permite direccionar el registro de datos (DLAB). Este es el modo a través del cual se puede establecer la velocidad de transmisión. Se debe señalar aquí que el 8250 tiene 7 direcciones de puerto para acceder 10 registros, por tanto, para acceder los registros 8 (DLL) y 9 (DLH) se debe poner primero en uno el DLAB en el registro LCR y después acceder cada registro mediante un desplazamiento cero o uno con respecto a la dirección base del puerto (ptobase). El registro 8 captura el byte bajo y el registro 9 el byte alto del divisor. Después de realizado lo anterior, se necesita apagar DLAB escribiendo el valor que estaba previamente almacenado en Current\_value en el registro de control de línea.

- La última función, **Otros**, permite indicar las características de la comunicación, como son: bit de paridad, longitud de

carácter y bits de parada. Como las otras funciones, las primeras líneas de código determinan si se ha dado algún parámetro inválido. El resto del código permite a partir de los parámetros anteriores construir el byte para ser escrito en el registro de control de línea.

Para poder capturar o enviar caracteres por el puerto serie se diseñaron dos subrutinas: **SerialIn** y **SerialOut**.

En la función **SerialIn**, el puerto se maneja a través de lo que se conoce como poleo y no mediante interrupciones. La razón por la cual no se utilizaron interrupciones es porque las funciones del lenguaje C que las manejan ocupan espacio en memoria, lo cual no es conveniente, debido a que se necesita tener una cantidad importante de memoria para almacenar los datos de un evento.

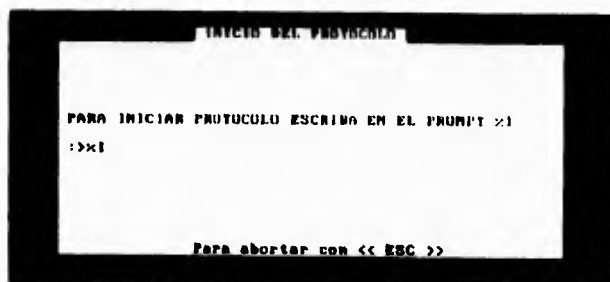
Para la función **SerialIn** se emplea la función clock para señalar que ha transcurrido cierto intervalo de tiempo, después de dicho intervalo se ve si ya se tiene un dato, revisando el bit Data Ready (DR) o dato listo del registro de estado de línea (LSR). Si DR es igual a uno, entonces se lee el registro de recepción y se guarda el carácter leído en una variable llamada Char\_Value. Si en ésta función transcurre un segundo sin que llegue el carácter entonces se regresa un -1, así el proceso que controla el poleo sabrá que no ha llegado dato en el puerto.

Al llamar a la función **SerialOut(char)** hay que pasarle el carácter que se colocará en el registro de transmisión del USART. De ésta forma, se logra que un carácter salga por el puerto en forma serie.

#### **VII.2.3.1 Protocolo de comunicación con el adquisidor.**

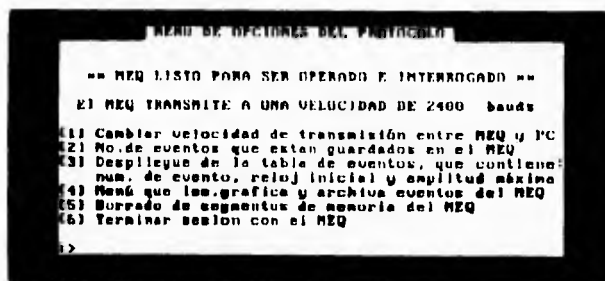
la elección del protocolo de comunicación se realiza en el

**MENÚ PRINCIPAL DEL MEQ\_800.** Cuando se hace ésta elección, se llama a la pantalla de elección del puerto serie (figura 7.9). Después aparece una pantalla en la cual se tiene que dar %I para iniciar la comunicación bidireccional con el adquisidor vía puerto serial. Los caracteres de % e I son transmitidos al adquisidor por medio de la función `SerialOut`. A continuación se muestra la pantalla con la cual se da inicio al protocolo.



**FIGURA 7.13** Pantalla que presenta el inicio del protocolo.

Si el equipo detectó que son correctos estos bytes, entonces manda los mismos caracteres para indicarle a la PC que está listo para ser interrogado. El usuario sabrá que el equipo está listo, porque aparecerá el menú de opciones de protocolo que a continuación se muestra.



**FIGURA 7.14** Menú de opciones del protocolo.

Como se puede observar en el menú anterior se tienen varias opciones que se explican a continuación:

#### **Selección de cambio de velocidad**

Al elegir ésta opción se envía al equipo los caracteres % y B. Si en el puerto serie se detecta que llegan los mismos caracteres, significa que el adquisidor se encuentra listo para hacer el cambio de velocidad.

Posteriormente se presenta un menú al usuario para que elija a que velocidad de transmisión desea que operen tanto la PC y el adquisidor. Si se elige la opción [1]600 bauds se manda al adquisidor el hexadecimal 01, si la opción es [2]2400 bauds, se manda al adquisidor el hexadecimal 02 y finalmente si se elige la opción [3]38400, se manda el hexadecimal 03 hacia el equipo. Posteriormente con la función SerialIn se patea el puerto para obtener el carácter de respuesta. Si llega un % significa que tanto la PC como el adquisidor ya operan a la velocidad seleccionada.

#### **Lectura del número de eventos que tiene almacenados el adquisidor**

Si se envía con la función SerialOut los 2 caracteres consecutivos % y N, el adquisidor interpreta que se solicita el número de eventos que tiene almacenados. En el puerto se verifica que llegue un carácter, correspondiente al número de evento.

#### **Lectura de la tabla de eventos**

Cuando se elige esta opción del menú del protocolo, se manda al adquisidor, por medio de SerialOut, los caracteres % y T.

Inmediatamente se polea el puerto de la PC para capturar los datos que arriben y se guardan en un arreglo estático como el definido anteriormente. Después se observa en la PC una pantalla como la siguiente:

TABLA DE EVENTOS			
NO. DE EVENTO	HR	MIN	SEG
1	12	34	43
2	15	52	31
3	17	15	6

oprima cualquier tecla para regresar al menú..

**FIGURA 7.15** Tabla de eventos transmitidos por el adquisidor.

#### **Elección del menú de RECEPCION Y PROCESAMIENTO.**

Cuando se selecciona esta opción, se pregunta al usuario cual evento de la tabla de datos desea mande el adquisidor; si dio un valor válido de evento, entonces se manda al adquisidor un % y enseguida el carácter que corresponde al número del evento elegido. Antes de enviar los caracteres se pone la pantalla de captura.

El adquisidor al recibir un % y el número de evento, envía los datos del evento. Inmediatamente después se manda llamar al procedimiento que realiza la captura de datos por el puerto



serie.

Para evitar que lleguen datos de evento antes de que el programa se ponga en condiciones de recepción, se acordó que el adquisidor realizara un ciclo de espera antes del envío.

#### **Elección del borrado de un segmento**

Para borrar un segmento se envían los caracteres % y D, para indicarle al equipo que se desea realizar un borrado en algún segmento de la memoria FLASH. El usuario visualiza en la PC una pantalla donde se le informa cuantos segmentos tiene la memoria y se le pregunta cual desea borrar. Antes de mandar el carácter que corresponde al segmento elegido por el usuario, a éste se le cuestiona si está seguro de querer llevar a cabo la operación. Si es afirmativa la respuesta, entonces se envía el carácter que corresponde al segmento y el adquisidor procede a borrarlo.

#### **Elección de finalización de protocolo.**

Cuando se elige esta opción se manda un %F al adquisidor utilizando la función SerialOut. %F indica al adquisidor que se desea terminar la interrogación. Al dar ésta opción se regresa al menú principal del MEQ-800 (figura 7.8).

#### **Condiciones de error en el protocolo.**

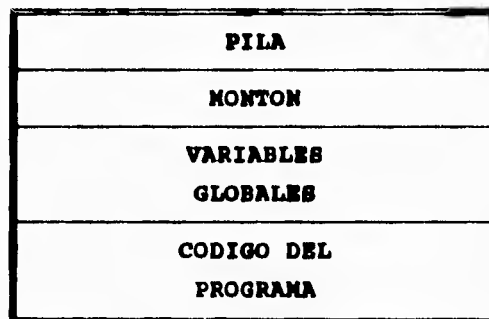
Si el equipo no está conectado correctamente para realizar el protocolo, cuando se quiera iniciar la interrogación éste no mandará el carácter que indica "listo" y por lo tanto el usuario podrá observar una pantalla donde se le indicará que la PC no pudo establecer la comunicación con el adquisidor.

Otro error puede ocurrir cuando el adquisidor conteste incorrectamente a una función elegida del protocolo. Cuando pase

esto se avisará al usuario que el MEQ-800 mandó dato equivocado y que sería recomendable verificar la conexión.

#### VII.2.4 Manejo de la memoria RAM de la computadora para almacenamiento temporal de los datos recibidos.

El programa fue compilado con la opción de **modelo pequeño de memoria** con lo cual se crean y utilizan 4 regiones lógicas diferentes de memoria durante su ejecución, las cuales sirven para realizar ciertas funciones. Estas regiones se presentan en la figura siguiente:



**FIGURA 7.16** Diagrama simplificado de la forma como organiza el compilador la memoria.

La primera y segunda región contienen el código ejecutable del programa y las variables globales, respectivamente. Las dos

regiones restantes son la pila y el montón. La pila se utiliza para diversas cosas durante la ejecución del programa: mantiene las direcciones de regreso en el caso de las llamadas a funciones, mantiene las variables locales y el estado del CPU. El montón es la región de memoria libre que puede utilizar el programa mediante las funciones de asignación dinámica.

Este sistema de asignación de memoria fue el utilizado en el programa. Con este método, el espacio de almacenamiento se asigna según se vaya necesitando área de memoria libre y se devuelve a dicha área cuando se ha terminado de utilizar. La región denominada montón es la que se utiliza para satisfacer las peticiones de memoria dinámica. Una ventaja de utilizar la asignación dinámica de memoria para almacenar los datos se debe a que la misma zona de la memoria se puede emplear para contener cosas diferentes a lo largo de la ejecución del programa. El programa permite utilizar toda la memoria RAM del sistema y almacenar temporalmente los datos recibidos en bloques de memoria mayores de 64 Kbytes.

Para llevar a cabo las labores de asignación de memoria del montón lejano, que se encuentra fuera del segmento de datos, se utilizó la siguiente función:

-**farcoreleft**: Se encarga de indicar el número de bytes libres de memoria que quedan en el montón lejano. Se debe hacer notar que esta función necesita el archivo de cabecera alloc.h para poder operar y no se encuentra definida como una función del estándar ANSI. Con la ayuda de esta función se puede determinar cuantos segundos de información se pueden guardar en memoria; el cálculo se realiza de la siguiente manera:

```
farcoreleft()  
lib_mmin=-----  
          (mps*2)
```

La variable mps indica la velocidad de muestreo. Esta variable sólo puede tomar dos valores:

60, cuando el registrador MEQ-800 capturó a 60 muestras/segundo ó 120 cuando se capturó a 120 muestras/segundo. Por tanto, (mps\*2) proporciona el número de bytes que están siendo generados por el registrador cada segundo. Si se divide el número de bytes libres en la RAM entre (mps\*2), entonces se obtiene el número de segundos de datos que se pueden grabar. Este número de segundos se da a conocer al usuario en el menú de **RECEPCION Y PROCESAMIENTO**, para que tenga conocimiento sobre el tamaño máximo del evento que puede descargar de el registrador a la PC.

Para guardar la información se decidió dividir la memoria en bloques capaces de almacenar un segundo de datos generados por los registradores. Para asignar la memoria en bloques se utilizó la función farcalloc.

- farcalloc: Esta función necesita del archivo de cabecera stdlib.h para poder ejecutarse. Se encarga de regresar un puntero a la memoria que acaba de asignar e inicializa con ceros todos los bloques asignados. El número de bloques reservados deben ocupar toda la memoria RAM libre. Los bloques de memoria se van llenando con los datos a medida que estos se reciben desde los registradores a través del puerto serie. El código de esta

sección está diseñado para que, si se requiere una nueva transmisión y ya se grabaron los datos previos en un archivo, no se libere la memoria, sino que se regresen los apuntadores a la posición inicial de memoria y se vuelva a poner en ceros las localidades para preparar para la captura de otro evento. Si el usuario decide no hacer más lecturas de eventos, entonces se libera la memoria mediante la función **farfree**.

#### **VII.2.5 Recepción de Evento y grabación en memoria del mismo.**

Para realizar este proceso se creó una rutina que básicamente realiza lo siguiente: Se llama iterativamente a la función **SerialIn** para polear el puerto serie y así lograr capturar byte. Si la función **SerialIn** regresa un -1, significa que no existe dato en el puerto y por lo tanto se debe estar iterando hasta que la función **SerialIn** regrese un valor entero distinto de -1. Cuando el valor es diferente de -1 entonces el dato capturado es guardado en una variable llamada **Char\_Value** e inmediatamente asignada a una localidad de un bloque de memoria. El usuario sabe que se acabó de transmitir, cuando la rutina considera que ha pasado un tiempo considerable sin que vuelva a llegar un byte al puerto.

Además, si el usuario decide truncar la transmisión lo puede hacer oprimiendo la tecla **ESC** y de ésta forma se detiene la captura y almacenamiento y se retorna al menú de **RECEPCIÓN Y PROCESAMIENTO**.

Por otra parte, si el adquisidor transmite un evento que llena todos los bloques de memoria reservados, entonces el programa, envía un mensaje al usuario indicándole que se truncó la información por falta de espacio en memoria.

A continuación se presenta la pantalla con la cual el usuario visualiza la captura:



**FIGURA 7.17** Pantalla en donde se visualiza la llegada de datos al puerto serie.

En la pantalla anterior cuando el adquisidor comienza a transmitir se va llenando de asteriscos(\*) la pantalla. Donde cada asterisco representa la llegada de un carácter en el puerto.

#### **VII.2.6 Detección y Corrección de error de la información guardada en memoria.**

El adquisidor envía bytes al puerto serie tal y como se encuentran almacenados en la memoria FLASH del adquisidor. Como cada muestra se forma por 2 bytes, el algoritmo para esta parte debe verificar que los 4 primeros bits de el primer byte sea siempre cero a excepción de que se detecte el hexadecimal 'C' que indica que en las 6 muestras consecutivas viene la información del reloj.

Básicamente éste algoritmo realiza lo siguiente:

-lee cada muestra(2 bytes) que se encuentra en la memoria y se le aplica el algoritmo de decodificación y detección de errores; si la muestra no presentó ningún problema se almacena

en un archivo temporal, creado con la finalidad de ir almacenando la información codificada y corregida.

-Si la muestra que entra al algoritmo se detecta que en los 4 bits del primer byte no tiene ceros, entonces significa que está errónea y por lo tanto se sustituye con la muestra válida anterior y en caso de que no exista una muestra anterior, no se guarda en el archivo temporal.

-Si se detecta que la parte alta está marcada con una C hexadecimal entonces guarda en 3 variables el reloj inicial que viene en las muestras consecutivas. En otras 3 variables se va guardando el reloj que se vaya capturando de cada segundo de información. Al final estas 3 variables se quedarán con el reloj final que el adquisidor grabó.

-Además, el algoritmo verifica que entre cada reloj existan exactamente el número de muestras con las que capturó el adquisidor la información. Esto quiere decir que si se eligió que el equipo capturara a 60 muestras por segundo, entonces debe haber exactamente éste número de muestras por cada segundo. Por otra parte, si hacen falta muestras entonces se escriben en el archivo temporal las muestras faltantes. Las muestras faltantes son sustituidas por la muestra válida anterior.

Al final de la detección y corrección de errores queda grabado en un archivo binario temporal la información decodificada y corregida del evento. Este archivo se manipula para graficarse y se pasa la información a un archivo ASCII cuando el usuario decide elegir la opción [3] del menú de **RECEPCION Y PROCESAMIENTO**. Si por el contrario, no desea guardar la información recién corregida el archivo se elimina.

## VII.2.7 Creación del Archivo de evento en formato ASCII.

La información almacenada en RAM puede ser enviada a un archivo en disco, si el usuario lo decide. El archivo generado es de tipo ASCII. El archivo contiene en sus dos primeras líneas un encabezado. El encabezado tiene información del evento, tal como: reloj inicial, reloj final, incremento de tiempo, muestras totales, muestras anteriores al reloj inicial, número de serie del sismógrafo transmisor, constante del sensor a utilizar y los decibelios con los que se operó el sismógrafo. El archivo guarda las muestras del evento como números reales en una sola columna.

A continuación, se muestra una fracción de archivo que ejemplifica lo dicho en el párrafo anterior:

Hi	Ni	Si	Hf	Mf	Sf	Dt	N_ant	N	Nosr	GL(u/m/s)	DB
14	03	30	14	19	15	0.0167	00	056760	0000	S/CTE	60
3.300000E+01											
4.200000E+01											
3.800000E+01											
3.300000E+01											
2.900000E+01											
2.700000E+01											
2.400000E+01											
1.900000E+01											
1.700000E+01											
1.600000E+01											
1.100000E+01											
9.000000E+00											

FIGURA 7.18 Se muestra el formato como queda guardada la información de un evento en un archivo ASCII.



### VII.2.8 Graficación de Evento.

Para tener un sistema completo de captación y observación de datos provenientes del MEQ-800, se decidió desarrollar esta subrutina de graficación. Dicha subrutina utiliza para trabajar en pantalla funciones del modo gráfico.

La forma como se presenta la gráfica al usuario queda ilustrada en las figura 7.19 y 7.20. La primera figura muestra una gráfica completa y la segunda figura muestra sólo una ventana de muestras, que se logra con la ejecución de funciones de ésta subrutina, las cuales se explicarán posteriormente. Se observa que la graficación de los datos está acotada en X y Y.

El cálculo del tiempo se realiza contabilizando las muestras del evento y sirve para acotar el eje X. El número resultante se divide entre el muestreo (60 mps ó 120 mps) y así se obtiene el tiempo total del evento.

El eje Y se acota con respecto a la muestra que presente la mayor amplitud. La amplitud está dada en cuentas, en donde cada cuenta es igual a 2.44 milivolts.

Para graficar las muestras en pantalla, el programa calcula la relación que existe entre cada muestra del evento, y el pixel que le corresponde en pantalla. Al terminar el cálculo, se procede, a encender en la pantalla el pixel correspondiente a la posición (X,Y).

En el borde inferior de la gráfica se visualizan ventanas, que tienen las siguientes funciones:

- **La primera ventana** da al usuario información general del evento como: nombre del archivo ASCII leído (aparecerá **MEMORIA**

si los datos son leídos de archivo temporal de datos corregidos), amplitud máxima, incremento de tiempo en cada muestra, duración en segundos, reloj inicial (hora, minuto, segundo), muestras anteriores capturadas antes del reloj inicial y el número de serie del equipo transmisor.

- En la segunda ventana podemos observar una serie de operaciones que se pueden realizar sobre la gráfica:

**F1-LIMITES.** Al oprimir la tecla F1 aparecerán en pantalla dos líneas verticales que se pueden desplazar para delimitar la ventana de datos que se desea apreciar de cerca.

**F2-ZOOM.** Si se ha delimitado la ventana como se indicó anteriormente, al presionar esta tecla, la zona delimitada se agranda para observarse con más detalle.

**F3-VISUALIZAR COMPLETO.** Al presionar la tecla F3, se gráfica de nuevo el evento completo.

**F4-TITULO DE GRAFICA.** Con esta opción se puede colocar un título en la parte superior de la gráfica.

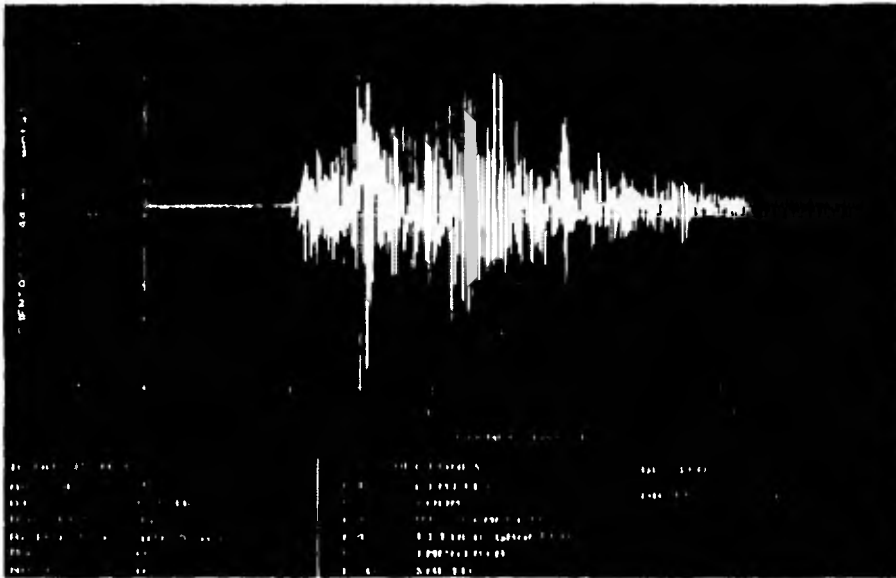
**F5-IMPRESION.** Con la tecla F5 se manda la gráfica hacia el puerto de la impresora, para obtenerla en papel. En esta opción sólo se tiene definida para una impresora de matriz de puntos.

**ESC-SALIR.** Al presionar la tecla ESC finalizamos el proceso de graficación.

- La tercera ventana nos informa a cuantos decibeles trabajó el sismógrafo y la constante del sensor utilizado.

- La cuarta ventana presenta valores cuando se oprime la función F1-LIMITES, ya que al desplazarse los límites se puede observar la posición (X,Y) en ésta ventana. Donde X es tiempo y Y es amplitud de evento.

- La quinta ventana tiene la función de presentar letreros de aviso y ayuda al usuario.



**FIGURA 7.19** Pantalla que presenta la gráfica completa de un evento adquirido el 20 de septiembre de 1995.

## **CAPITULO VIII**

---

### ***PRUEBAS, IMPLEMENTACION Y EVALUACION DEL SISTEMA***

La calidad de un sistema depende de su diseño, desarrollo, pruebas e implantación. Un aspecto de la calidad del sistema es su confiabilidad. Un sistema es confiable si al usarse de manera razonable, no produce fallas peligrosas o costosas.

El proceso para asegurar la calidad de un sistema incluye pruebas para garantizar que el funcionamiento del mismo se realiza de forma adecuada y cumple con los requerimientos establecidos. El propósito de la prueba es hallar errores, no el demostrar lo correcto de un sistema.

Después de implantar el sistema debe hacerse una evaluación para determinar si el sistema cumple con las expectativas y dónde serán necesarias las mejoras.

#### **VIII.1 PRUEBAS DE INTEGRACIÓN, IMPLEMENTACION Y RESULTADOS DEL SISTEMA.**

El sistema completo de adquisición y procesamiento, desarrollado en este trabajo, involucra el hardware del adquisidor digital, el programa en ensamblador que controla, registra y se comunica bidireccionalmente con la PC y el programa para PC en lenguaje C, llamado TRANSER, que captura, almacena, gráfica y genera archivos de eventos que son recibidos desde el adquisidor.

En primer lugar se realizaron pruebas con el hardware del adquisidor digital. Para poder asegurar que todos los módulos que iban formando parte del sistema funcionaban adecuadamente, fue

necesario ir implementando y realizando pruebas conforme

se iba avanzando en el desarrollo del diseño. En muchos casos fue necesario desarrollar programas en lenguaje ensamblador que tuvieran asignadas pequeñas funciones enfocadas a comprobar la adecuada operación del circuito. Si se tenía algún problema en la forma de operación, se observaban las señales provenientes del circuito bajo prueba mediante un osciloscopio y se realizaban ajustes, ya sea al circuito o al programa en ensamblador hasta que se obtenía la operación adecuada del circuito.

Bajo esta perspectiva, el programa en ensamblador se desarrolló modularmente. Cada módulo permitió verificar el correcto funcionamiento de las interrupciones, el display, el teclado, el circuito de disparo, la captura del reloj, el almacenamiento en las memorias de estado sólido EEPROM, tipo FLASH, el muestreo de 60 ó 120 pulsos por segundo, los selectores de memoria y puertos, la adquisición de la información proveniente del Conversor A/D y la comunicación bidireccional entre el adquisidor y la PC.

Una vez que se comprobaba, con los módulos desarrollados en ensamblador que los circuitos operaban adecuadamente se iban realizando funciones específicas encaminadas a cumplir con los requerimientos establecidos; de esta forma se logró obtener, un programa que es lo bastante robusto para dar soporte al hardware diseñado y que conjuntamente y en paralelo al sismógrafo analógico puede realizar la adquisición, almacenamiento y comunicación bidireccional con una PC.

Respecto al programa **TRANSER** para PC, se aplicó la filosofía de integración ascendente, que consistió en desarrollar y probar las rutinas individualmente antes de su

integración. Es por esto, que las pruebas se realizaron a la par del desarrollo del programa. Cabe mencionar la importancia que tuvo la participación de las personas de la Coordinación de Ingeniería Sismológica en muchas de las pruebas, ya que con su experiencia se evaluaron muchos casos no contemplados en el procesamiento de la información capturada, los cuales fueron incluidos y también probados.

El procedimiento para evaluar el sistema integrado de adquisición y procesamiento de datos completo fue el siguiente: En el laboratorio se conectaba un sensor al MEQ-800 tipo L4 y se ponían en operación continua tanto el adquisidor digital como el registrador analógico del MEQ-800. El adquisidor ya en operación detectaba eventos y los guardaba en memorias FLASH. Posteriormente, los eventos eran transmitidos a PC y capturados mediante el programa **TRANSER**. La rutina de graficación del programa **TRANSER** permitió observar las gráficas de los eventos. Cuando un evento era considerado de importancia se guardaba en un archivo ASCII y además se mandaba a impresión en papel, todo esto con funciones del programa **TRANSER**.

El registro en papel del adquisidor digital se comparaba con el registro analógico proporcionado por el sismógrafo MEQ-800, con el fin de tener una visión de que lo que se capturaba y procesaba correspondía a la información analógica.

#### PRUEBAS, IMPLEMENTACION Y EVALUACION DEL SISTEMA

Cabe hacer mención, que se realizaron pruebas de forma exhaustiva llegando a hacerse inclusive diariamente.

De las pruebas antes mencionadas, se llegaron a obtener registros de bastante calidad, de eventos de gran importancia, tales como los ocurridos los días 14 y 20 de Septiembre de 1995.

Es importante que los datos se guarden en archivos con formato ASCII, ya que entonces es posible procesarlos con programas de uso común en la Coordinación de Ingeniería Sismológica, como son DEGTRA5 y PITSA.

Los datos que se presentan en las gráficas mostradas a continuación fueron obtenidos los días 14 y 20 de Septiembre de 1995. Los registros analógicos son presentados en las figuras 8.1 y 8.4.

Las gráficas que aparecen en las figuras 8.2 y 8.5 se obtuvieron de procesar el archivo que se captura en el adquisidor digital desarrollado. El proceso que da lugar a la gráfica, consiste de: grabación del evento en las memorias FLASH, transmisión de él a PC, detección y corrección de errores y graficación; además de la creación del archivo ASCII correspondiente.

En las gráficas generadas por el programa TRANSER se puede observar información de relevancia al usuario, como: la duración del evento, la amplitud máxima, el reloj inicial, los decibeles con que operó el MEQ, la constante del sensor y, finalmente el nombre del archivo ASCII, donde se encuentra esta información.



Las figuras 8.3 y 8.6 presentan los resultados obtenidos de procesar, con el paquete llamado PITSA, los archivos ASCII correspondientes a los sismos de los días 14 y 20 de Septiembre.

En estas figuras(8.3 y 8.6) se pueden observar 4 ventanas:

- (1) En la primera ventana se observa la gráfica de 4096 muestras del archivo ASCII(generado por el programa TRANSER). Esta gráfica representa la velocidad del evento.
- (2) En la segunda ventana se observa la corrección del desplazamiento("offset") que tiene el evento de la primera ventana.
- (3) En la tercera ventana se observa el espectro de frecuencias obtenido a partir de los datos de la ventana 2.
- (4) En la cuarta ventana se observa el resultado de derivar los datos de velocidad de la ventana 2. Por lo tanto, lo que se visualiza en la ventana cuatro es la aceleración.

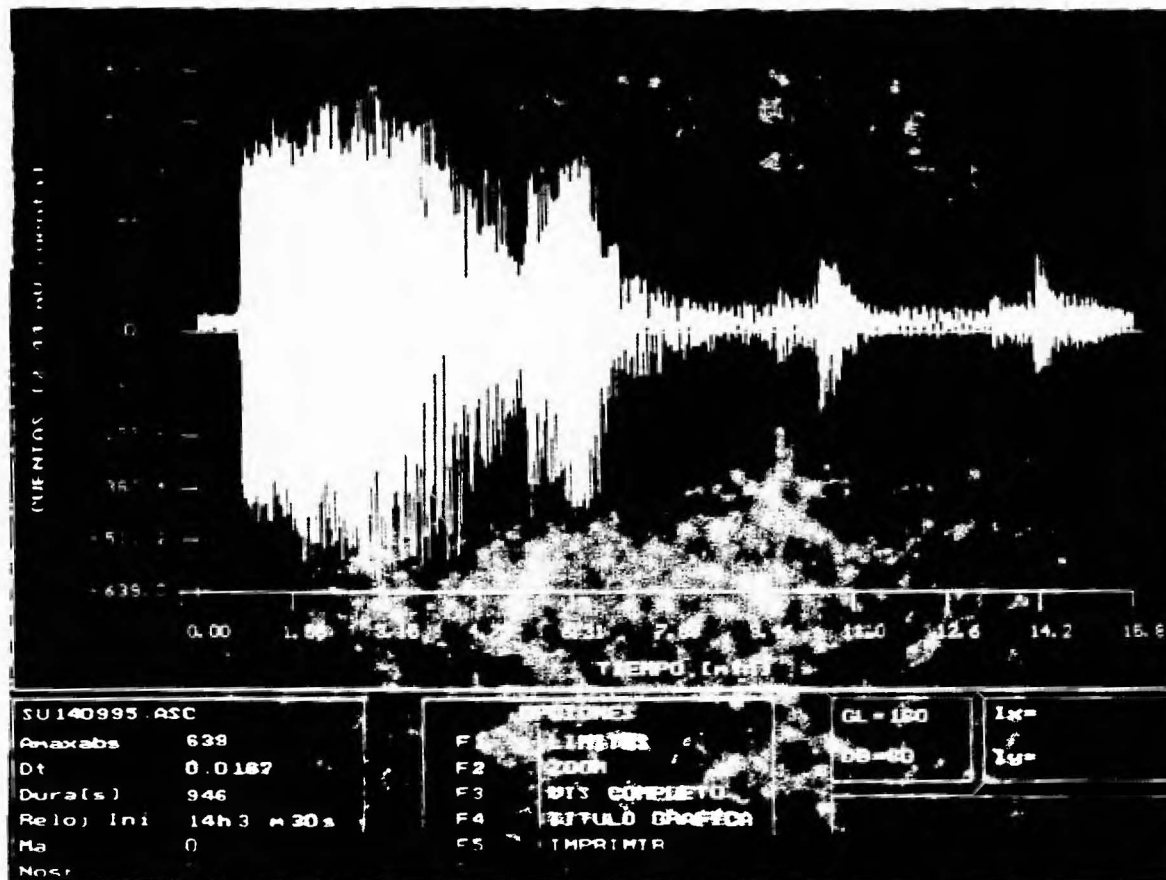


FIGURA 8.2 REGISTRO DIGITAL OBTENIDO POR EL SISTEMA DESARROLLADO CORRESPONDIENTE AL SISMO OCURRIDO EL DIA 14 DE SEPTIEMBRE DE 1995.

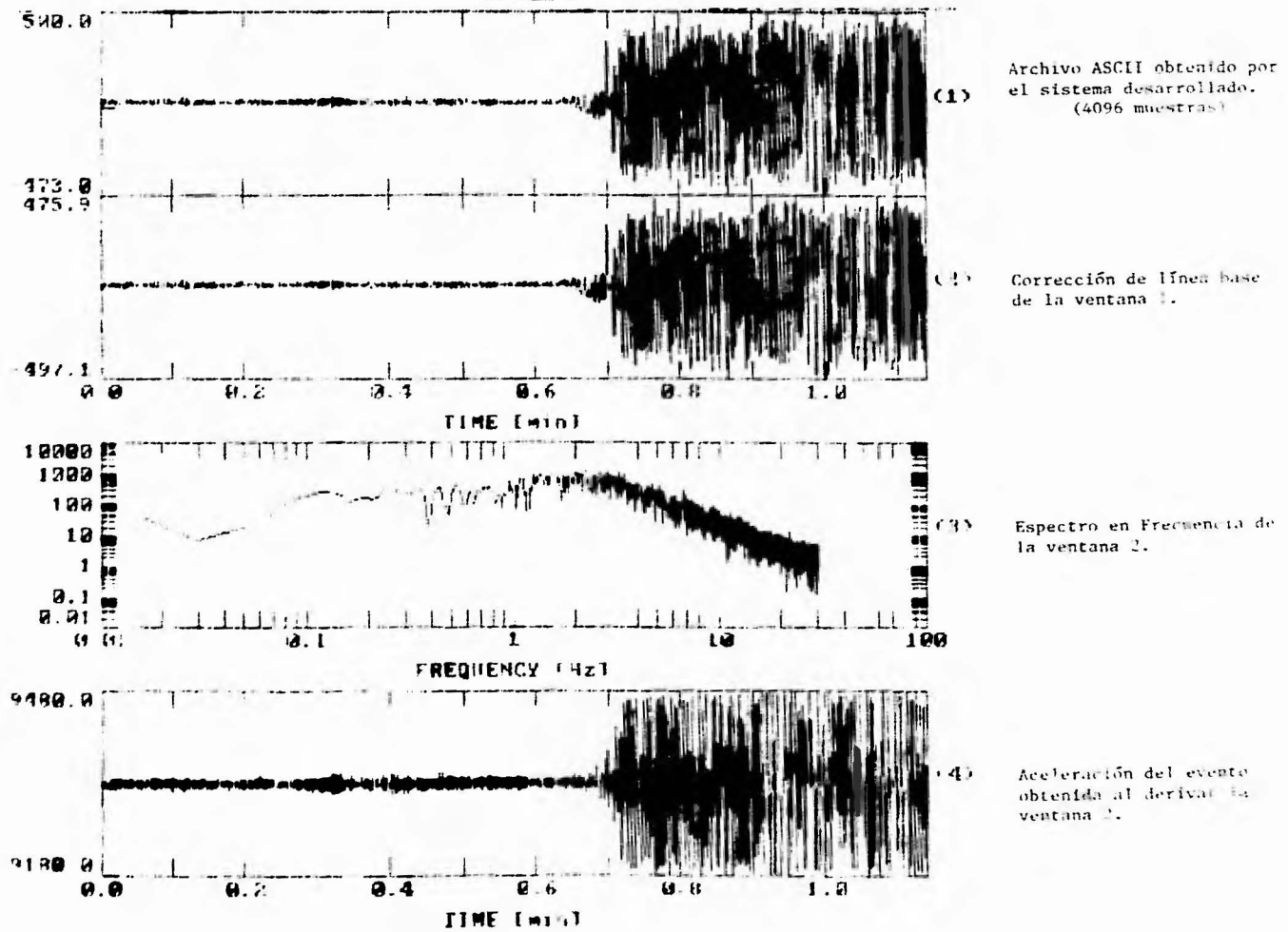


FIGURA 8.3 ARCHIVO ASCII OBTENIDA POR EL SISTEMA DESARROLLADO, CORRESPONDIENTE AL SISMO OCURRIDO EL DIA 14 DE SEPTIEMBRE DE 1995 PROCESADO CON EL PAQUETE PITSA

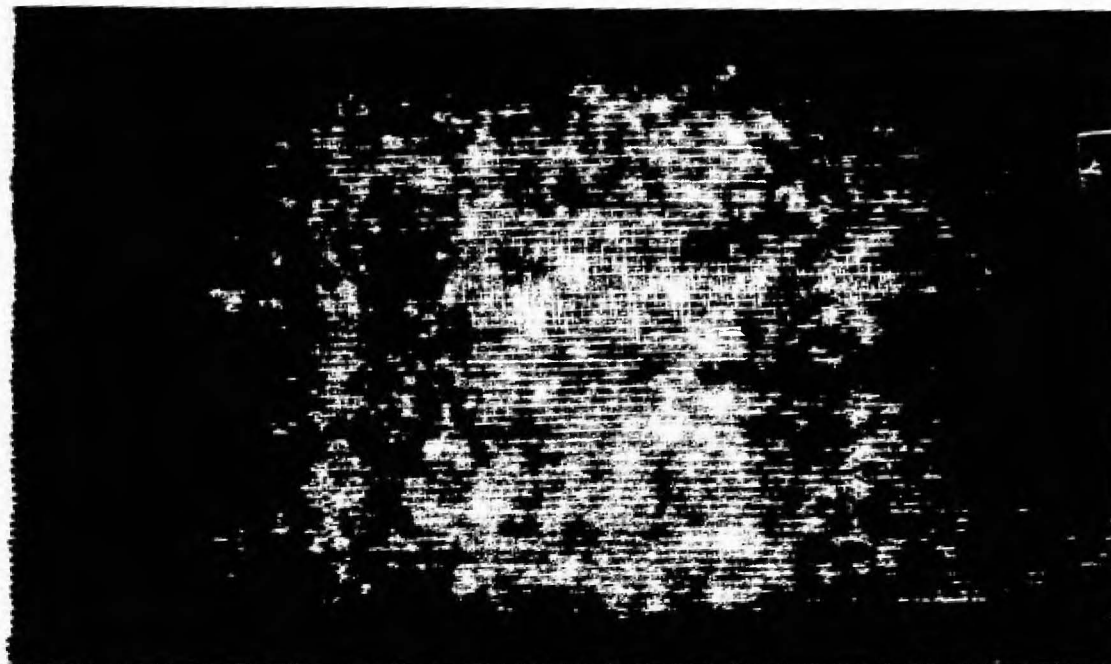


FIGURA 8.4 REGISTRO ANALOGICO OBTENIDO DEL SISMOGRAFO MEQ-800 CORRESPONDIENTE AL SISMO OCURRIDO EL DIA 20 DE SEPTIEMBRE DE 1995.

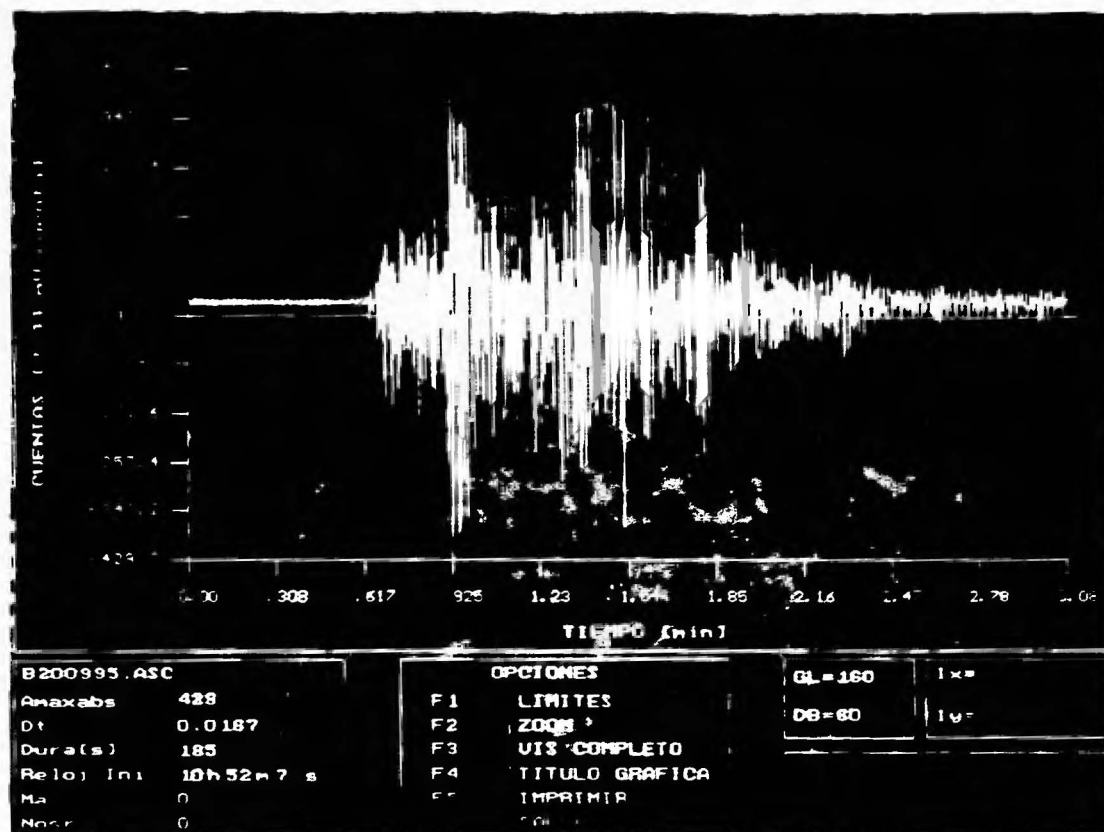


FIGURA 8.5 REGISTRO DIGITAL OBTENIDO POR EL SISTEMA DESARROLLADO CORRESPONDIENTE AL SISMO OCURRIDO EL DIA 20 DE SEPTIEMBRE DE 1995.

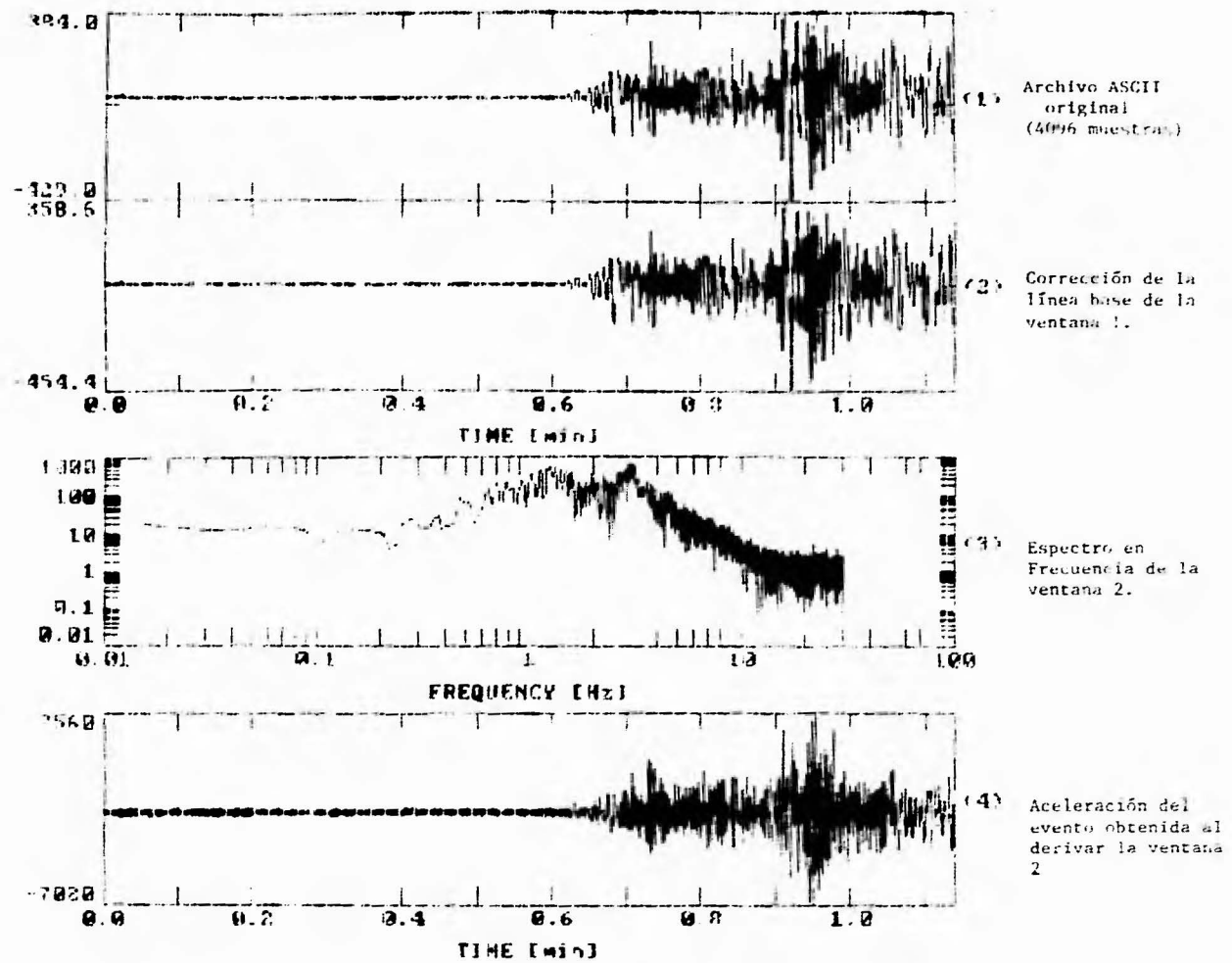


FIGURA 8.6 ARCHIVO ASCII OBTENIDO POR EL SISTEMA DESARROLLADO, CORRESPONDIENTE AL SISMO OCURRIDO EL DIA 20 DE SEPTIEMBRE DE 1995 PROCESADO CON EL PAQUETE PIISA

## **CAPITULO IX**



## ***CONCLUSIONES***

## CONCLUSIONES

Con el desarrollo de este trabajo, tal como lo muestran los resultados indicados en el capítulo anterior, se ha mejorado en gran medida la calidad de los datos obtenidos y procesados con el sismógrafo MEQ-800. Con este sistema, se han podido registrar eventos importantes como los ocurridos los días 14 y 20 de Septiembre del 1995.

Al realizar una comparación entre la información digital que proviene del sistema desarrollado y la información analógica proporcionada por el adquisidor, se puede observar que la información digital es confiable, ya que demuestra semejanzas muy particulares con la información analógica.

Los programas diseñados tanto para el adquisidor como para la PC, están organizados de manera tal que permiten agregar nuevas funciones sin tener que realizar grandes cambios a lo ya establecido. De ésta forma, se amplía el campo de acción de los programas, ya que no sólo se limitan a cumplir funciones específicas, definidas con anterioridad, si no que, además, se pueden ir adaptando a las nuevas necesidades que vaya teniendo el usuario.

Al implementar los programas que realizan la comunicación bidireccional entre la PC y el adquisidor, se crea una nueva alternativa para el usuario, ya que de una manera fácil y amigable, obtiene la información que proporciona el adquisidor desde una PC portátil, sin tener que manipular el equipo de manera manual.

Una innovación importante en este sistema, es el de utilizar memorias con capacidad de almacenamiento de 256 Kbytes, lo que



#### CONCLUSIONES

significa que el tiempo de grabación de la información se amplía de manera importante con respecto al prototipo anterior, además, este tipo de memorias al no necesitar de un soporte externo de baterías para retener la información almacenada, le da al usuario confiabilidad en el manejo de la información obtenida.

El sistema realizado deja abierta la puerta al desarrollo de nuevas alternativas. A continuación se describen algunas de ellas:

1). Lograr la comunicación entre el adquisidor y la PC mediante el uso de un MODEM con el propósito de permitir la interrogación lejana de los sismógrafos.

2). Elaboración y aplicación de los programas que permitan implementar un algoritmo de compactación de información para aumentar la eficiencia en el uso de la memoria.

3). Implementación en el programa del adquisidor, de una función que proporcione la cantidad de memoria ocupada y la cantidad de memoria disponible.

4). También es importante mencionar que de acuerdo a la nueva tecnología que está apareciendo en el mercado, es posible reducir el tamaño del hardware diseñado para la detección, captura, almacenamiento y manipulación de la información que proviene del adquisidor, esto es posible, al utilizar un microcontrolador, el cual tenga en un solo integrado: memorias RAM y EPROM, puertos de salida y entrada, el Conversor A/D, entre otras cosas; simplificando de esta manera el hardware del adquisidor, además de que solucionamos el problema de espacio,

#### CONCLUSIONES

debido al uso de varios componentes electrónicos.

Con el presente trabajo, se propone una nueva alternativa para lograr disminuir la dependencia tecnológica en el área de instrumentación sísmica. La importancia de lo anterior, es que gracias a los nuevos desarrollos, nuestro país va en dirección de ser autónomo por lo menos en el área sísmológica.

## **CAPITULO X**

---

# ***RECONOCIMIENTOS***

**Al M.I. Pablo Roberto Pérez Alcázar** por compartir sus valiosos conocimientos, su dirección, ayuda y disposición permanente.

**Al Ing. Horacio Mijares Arellano** por sus atinados comentarios a lo largo de este trabajo.

**Al Ing. Emilio Nava Alatorre** por su ayuda y disposición.

**A Ernesto Briones Ruiz** por permitirnos trabajar con sus equipos y por su ayuda incondicional.

**A la Coordinación de Ingeniería Sismológica del Instituto de Ingeniería de la UNAM** por habernos brindado su apoyo económico y tecnológico.

**A todas las personas** que nos apoyaron desde el principio.

## **CAPITULO XI**



## ***BIBLIOGRAFIA***

- Herbert Schildt  
TURBO C/C ++  
Manual de Referencia  
Osborne McGraw - Hill, 1992.
  
- Joe Campbell  
C PROGRAMMER'S GUIDE TO SERIAL COMMUNICATIONS  
McGraw - Hill.
  
- Tom Swan  
MASTERING BORLAND  
COVERS BORLAND C++ 3.1  
McGraw - Hill.
  
- Douglas V. Hall  
MICROPROCESSORS AND INTERFACING  
PROGRAMMING AND HARDWARE  
McGraw - Hill.
  
- Christopher L.Morgan - Mitchell Wait  
INTRODUCCION AL MICROPROCESADOR 8086/8088 (16 bits)  
McGraw - Hill.
  
- Yu-Cheng Lin - Glenn A. Gibson  
MICROCOMPUTER SYSTEMS: THE 8086/8088 FAMILY  
ARCHITECTURE, PROGRAMMING AND DESIGN  
Prentice Hall Inc.
  
- J. Terry Godfrey  
LENGUAJE ENSAMBLADOR PARA MICROCOMPUTADORAS IBM  
PARA PRINCIPIANTES Y AVANZADOS.  
Prentice Hall inc.
  
- G. Suárez R. - Z. Jiménez J.  
CUADERNOS DEL INSTITUTO DE GEOFISICA 1 Y 2  
Universidad Nacional Autónoma de México.

**A  
P  
E  
N  
D  
I  
C  
E  
  
A**

TABLE 1

Comparison of Portable Drum Recorders

SPECIFICATION	MEQ 80G-B SPRENGNETHER	RV-320B (PORTACORDER) <sup>1</sup> TELEDYNE GEOTECH	PS-2 KINEMATICS
DRUM ROTATION RATES	60 and 120 mm/min (Motor & Pulley Change Required)	30, 60, 120, 240 mm/min (Switchable)	15, 30, 60, 120, 240 mm/min (Switchable)
PEN TRANSLATION RATES	0.5, 1, 2 mm/Rev. @ 60 mm/min 1, 2, 4 mm/Rev. @ 120 mm/min	1, 2, 4, 8 mm/Rev. (Switchable)	0.5, 1, 2, 4, 8 mm/Rev. (Switchable)
HIGH PASS FILTER	0.15, 5, 10 Hertz	0.2, 5, 10 Hz	DC, 0.1, 5 Hz
LOW PASS FILTER	5, 10, 30 Hertz, Out	5, 12.5, 25, 50 Hertz	2.5, 5, 12.5, 25, 50 Hertz Also Switch Selectable Wood-Anderson Filter
INPUT IMPEDENCE	100K Single-Ended	10K	100K Differential 50K Single Ended
TYPE OF DRUM DRIVE	Friction	Gear	Gear
ELECTRONIC PACKAGING	3 Printed Wiring Boards	1 Printed Wiring Board	5 Plug-in Modules
PROVISION FOR EASY CONNECTION TO TELEMETRY	No	No	Yes, Simple Substitution for Plug-in Amplifier Module with Plug-in Discriminator
DIGITAL CLOCK DISPLAY	LED	Liquid Crystal	Liquid Crystal
RECORDING TIME BETWEEN BATTERY CHARGES	5 Days	Not Specified	8 Days
ABILITY TO CHARGE BATTERIES WHILE OPERATING	No	Yes (External Charger)	Yes (Internal Float Charger)
LIGHTNING PROTECTION	No	No	Yes (Standard)
ABILITY TO CONNECT FOR COMMON TIMING	No	No	Yes, not only for common time marks but also for common drum and pen drive
EXTERNAL BATTERY INPUT	Yes, Two 12V Batteries Req'd	Yes, Single 12V Battery Req'd	Yes, Single 12V Battery Req'd
AUTOMATIC STYLUS LIFT AT EDGE OF RECORD	No	No	Yes



BENCHMARK STUDY  
PORTABLE SEISMOGRAPHS

TABLE 2

<u>Specifications:</u>	SPRENGNETHA MEQ-800	GEOTECH Paralaseefer	KINEMETRICS PS-B
Frequency Response (Amplifier) (Pen)	100 Hz 80 Hz	100 Hz 40 Hz	100 Hz Not Specified
Sensitivity	3.0 mm/ $\mu$ V	0.19 mm/ $\mu$ V	1.0 mm/ $\mu$ V
Amp. Noise	0.3 $\mu$ V P-P	1.2 $\mu$ V P-P	0.7 $\mu$ V P-P
Gain: (dB) (voltage)	120 dB 1,000,000	96 dB 65.536	72 dB 4,000
Clock Accuracy: (0-50°C)	$1 \times 10^{-9}$ sec	$3 \times 10^{-9}$	$1 \times 10^{-8}$
Stability	$\pm 3 \times 10^{-7}$	$\pm 1 \times 10^{-6}$	$\pm 3 \times 10^{-7}$
Drift	10-15 mSec/day	25 mSec/day	100-150 mSec/day
Pen Deflection	5, 10, 35 mm	7, 14, 28 mm	30 mm
Operating Power	35 MA	30 MA	60 MA
Operating Time	120 hrs	400 hrs	190 hrs
External Supply	12 VDC	12 VDC	12 VDC
<u>Features:</u>			
Drum:			
Standard	Smoke (ink optional)	Ink (smoke optional)	Smoke (ink optional)
Spare Drum	Included (no extra charge)	at extra cost	at extra cost
Rotation Rate	7.5 - 600 mm/min	30, 60, 120, 240 mm/min	15, 30, 60, 120, 240 mm/min
End of record shutdown:	Mech & elec (except clock)	Mech.	Mech. & elec (except clock)
Record size	340 x 600 mm	400 x 600 mm	340 x 600 mm
Motor:			
Type	Sync.	Stepper	Stepper
Speed (hours/record)	12.5 hr to 50 hrs	2 hr to 120 hrs	.5 hr to 192 hrs
Record format	Curvilinear	Rectilinear	Curvilinear
Ability to connect for common timing	Yes	No	Yes
Easy connection to Telemetry	Yes	No	Yes
Automatic stylus lift at end of record	Not Required	No	Yes
Lightning protection	Yes	No	Yes
Digital Clock Display	L.E.D. (operates at low temperature)	LCD (limited Temp range)	LCD (limited Temp range)
Ability to charge batteries while operating	Yes (external charger)	Yes (external charger)	Yes (internal charger)
<u>Physical:</u>			
Dimensions	18 x 12 x 19	23 x 13 x 9.19	23 x 13.5 x 9
Weight	32 lbs	35.2 lbs	37 lbs
Drum Drive	Friction	Gear	Gear
Operating Temperature Range	-30°C to 55°C	Not Given	0°C to 50°C

**A  
P  
P  
E  
N  
D  
I  
C  
E  
  
B**

FALLA DE ORIGEN

intel.

28F020

## 2048K (256K x 8) CMOS FLASH MEMORY

- Flash Electrical Chip-Erase
  - 2 Second Typical Chip-Erase
- Quick-Pulse Programming™ Algorithm
  - 10  $\mu$ s Typical Byte-Program
  - 4 Second Chip-Program
- 10,000 Erase/Program Cycles Minimum
- 12.0V  $\pm$  5% Vpp
- High-Performance Read
  - 150 ns Maximum Access Time
- CMOS Low Power Consumption
  - 10 mA Typical Active Current
  - 50  $\mu$ A Typical Standby Current
  - 0 Watts Data Retention Power
- Integrated Program/Erase Stop Timer
- Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface
- Noise Immunity Features
  - $\pm$  10% Vcc Tolerance
  - Maximum Latch-Up Immunity through EPI Processing
- ETOX™ II Nonvolatile Flash Technology
  - EPROM-Compatible Process Base
  - High-Volume Manufacturing Experience
- JEDEC-Standard Pinouts
  - 32-Pin Plastic Dip
  - 32-Lead PLCC
  - 32-Lead TSOP

(See Packaging Spec, Order #28F020)

Intel's 28F020 CMOS flash memory offers the most cost-effective and reliable alternative for read/write random access nonvolatile memory. The 28F020 adds electrical chip-erase and reprogramming to familiar EPROM technology. Memory contents can be rewritten: in a test socket; in a PROM-programmer socket; on-board during subassembly test; in-system during final test; and in-system after-sale. The 28F020 increases memory flexibility, while contributing to time and cost savings.

The 28F020 is a 2048-kibibit nonvolatile memory organized as 262,144 bytes of 8 bits. Intel's 28F020 is offered in 32-pin plastic DIP, 32-lead PLCC, and 32-lead TSOP packages. Pin assignments conform to JEDEC standards for byte-wide EPROMs.

Extended erase and program cycling capability is designed into Intel's ETOX™ II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional EEPROMs. With the 12.0V Vpp supply, the 28F020 performs a minimum of 10,000 erase and program cycles well within the time limits of the Quick-Pulse Programming and Quick-Erase algorithms.

Intel's 28F020 employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 150 nanosecond access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 100  $\mu$ A translates into power savings when the device is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from  $-1V$  to  $V_{CC} + 1V$ .

With Intel's ETOX II process base, the 28F020 leverages years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

3

# FALLA DE ORIGEN

3-83

MAY-17-93 MON 12:58 DICOPEL PIMENTEL

Int'l.

28F020

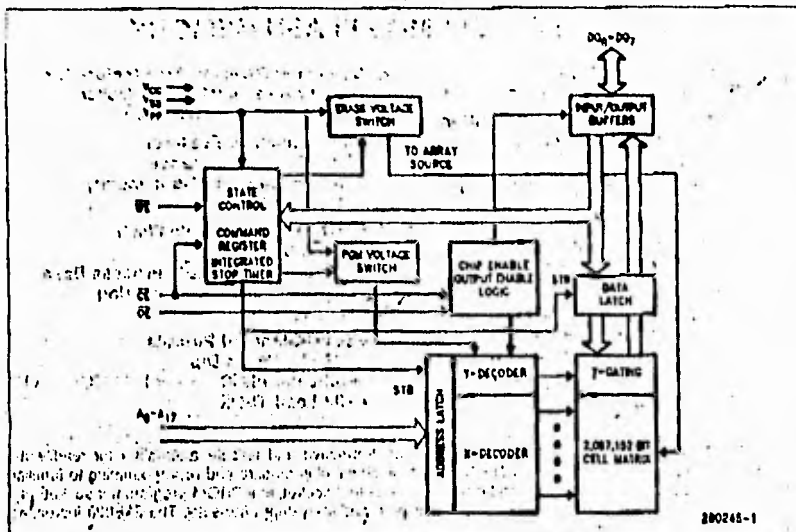


Figure 1. 28F020 Block Diagram

Table 1. Pin Description

Symbol	Type	Name and Function
A <sub>0</sub> -A <sub>17</sub>	INPUT	ADDRESS INPUTS for memory addresses. Addresses are internally latched during a write cycle.
DQ <sub>0</sub> -DQ <sub>7</sub>	INPUT/OUTPUT	DATA INPUT/OUTPUT: Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
CE	INPUT	CHIP ENABLE: Activates the device's control logic, input buffers, decoders and sense amplifiers. CE is active low; CE high deselects the memory device and reduces power consumption to standby levels.
OE	INPUT	OUTPUT ENABLE: Gates the device's output through the data buffers during a read cycle. OE is active low.
WE	INPUT	WRITE ENABLE: Controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the WE pulse. Note: With V <sub>pp</sub> ≤ 6.5V, memory contents cannot be altered.
V <sub>pp</sub>		ERASE/PROGRAM POWER SUPPLY for writing the command register, erasing the entire array, or programming bytes in the array.
V <sub>CC</sub>		DEVICE POWER SUPPLY (5V ± 10%)
V <sub>SS</sub>		GROUND

3-82

MAY-17-93 MON 12:57 DICOPEL PIMENTEL

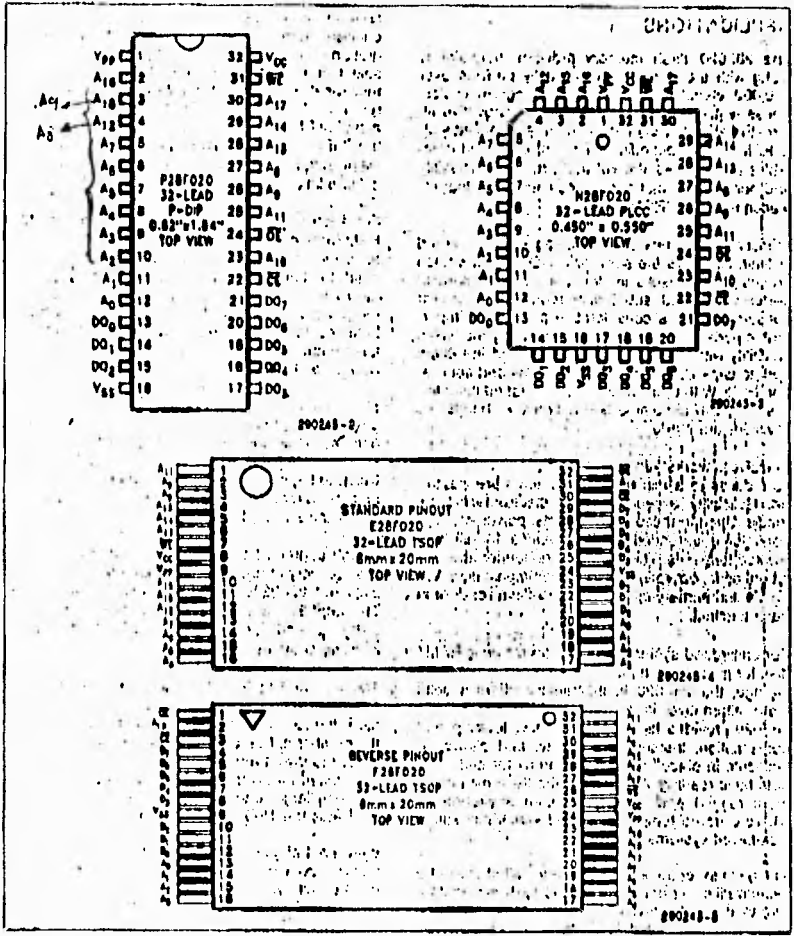


Figure 2. 28F020 Pin Configurations

FALLA DE ORIGEN

intel

PRELIMINARY

intel

28F010

PRELIMINARY

### 28F010 1024K (128K x 8) CMOS FLASH MEMORY

- Flash Electrical Chip-Erase
    - 1 Second Typical Chip-Erase
  - Quick-Pulse Programming™ Algorithm
    - 10  $\mu$ s Typical Byte-Program
    - 2 Second Chip-Program
  - 10,000 Erase/Program Cycles Minimum
  - 12.0V  $\pm$  5% Vpp
  - High-Performance Read
    - 135 ns Maximum Access Time
  - CMOS Low Power Consumption
    - 30 mA Maximum Active Current
    - 100  $\mu$ A Maximum Standby Current
  - Command Register Architecture for Microprocessor/Microcontroller Compatible Write Interface
  - Noise Immunity Features
    - $\pm$  10% VCC Tolerance
    - Maximum Latch-Up Immunity through EPI Processing
  - ETOX™-II Flash-Memory Technology
    - EPROM-Compatible Process Base
    - High-Volume Manufacturing Experience
  - Compatible with JEDEC-Standard Byte-Wide EPROM Pinouts
    - 32-Pin Cerdip
    - 32-Lead PLCC
- (See Packaging Spec., Order # 231361)

Intel's 28F010 CMOS flash memory offers the most cost-effective and reliable alternative for read/write random access nonvolatile memory. The 28F010 adds electrical chip-erase and reprogramming to familiar EPROM technology. Memory contents can be rewritten, in a test socket, in a PROM-programmer socket, on-board during subassembly test, in-system during final test, and in-system after-sale. The 28F010 increases memory flexibility, while contributing to time- and cost-savings.

The 28F010 is a 1024-kilobit nonvolatile memory organized as 131,072 bytes of 8 bits. Intel's 28F010 is offered in 32-pin cerdip or 32-lead PLCC packages. Pin assignments conform to JEDEC standards for byte-wide EPROMs.

Extended erase and program cycling capability is designed into Intel's ETOX-II (EPROM Tunnel Oxide) process technology. Advanced oxide processing, an optimized tunneling structure, and lower electric field combine to extend reliable cycling beyond that of traditional EEPROMs. With the 12.0V Vpp supply, the 28F010 performs a minimum of 10,000 erase and program cycles well within the time limits of the Quick-Pulse Programming™ and Quick-Erase™ algorithms.

Intel's 28F010 employs advanced CMOS circuitry for systems requiring high-performance access speeds, low power consumption, and immunity to noise. Its 135 nanosecond access time provides no-WAIT-state performance for a wide range of microprocessors and microcontrollers. Maximum standby current of 100  $\mu$ A translates into power savings when the device is deselected. Finally, the highest degree of latch-up protection is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins, from -1V to VCC + 1V.

With Intel's ETOX-II process base, the 28F010 leverages years of EPROM experience to yield the highest levels of quality, reliability, and cost-effectiveness.

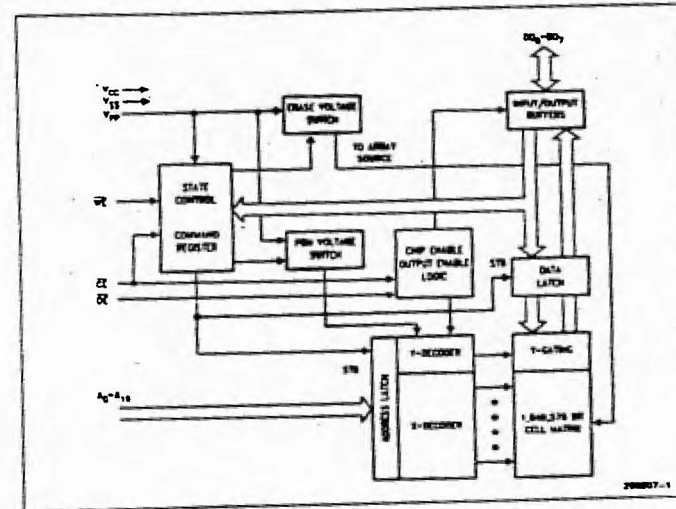


Figure 1. 28F010 Block Diagram

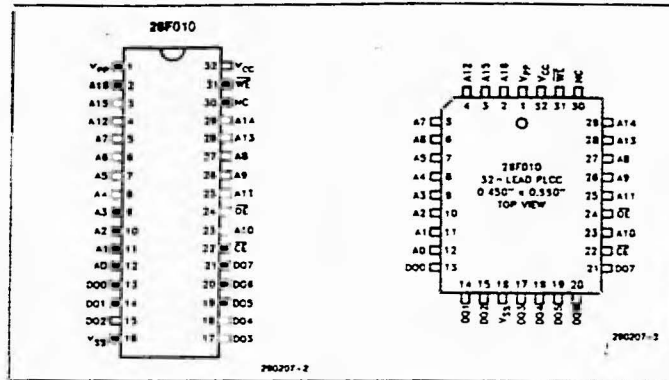


Figure 2. 28F010 Pin Configurations

Table 1. Pin Description

Symbol	Type	Name and Function
A <sub>0</sub> -A <sub>16</sub>	INPUT	ADDRESS INPUTS for memory addresses. Addresses are internally latched during a write cycle.
DO <sub>0</sub> -DO <sub>7</sub>	INPUT/OUTPUT	DATA INPUT/OUTPUT: Inputs data during memory write cycles; outputs data during memory read cycles. The data pins are active high and float to tri-state OFF when the chip is deselected or the outputs are disabled. Data is internally latched during a write cycle.
CE	INPUT	CHIP ENABLE: Activates the device's control logic, input buffers, decoders and sense amplifiers. CE is active low. CE high deselects the memory device and reduces power consumption to standby levels.
OE	INPUT	OUTPUT ENABLE: Gates the device's output through the data buffers during a read cycle. OE is active low.
WE	INPUT	WRITE ENABLE: Controls writes to the control register and the array. Write enable is active low. Addresses are latched on the falling edge and data is latched on the rising edge of the WE pulse. Note: With V <sub>pp</sub> ≤ V <sub>CC</sub> = 2V, memory contents cannot be altered.
V <sub>pp</sub>		ERASE/PROGRAM POWER SUPPLY for writing the command register, erasing the entire array, or programming bytes in the array.
V <sub>CC</sub>		DEVICE POWER SUPPLY (5V ± 10%)
V <sub>SS</sub>		GROUND
NC		NO INTERNAL CONNECTION to device. Pin may be driven or left floating.

## APPLICATIONS

The 28F010 flash-memory adds electrical chip-erase and reprogrammability to EPROM non-volatility and ease of use. The 28F010 is ideal for storing code or data-tables in applications where periodic updates are required. With a minimum of 10,000 erase/program cycles, the 28F010 also serves as a dense, nonvolatile data acquisition and storage medium.

The need for code updates pervades all phases of a system's life—from prototyping to system manufacture to after-sale service. In the factory, during prototyping, revisions to control code necessitate ultraviolet erasure and reprogramming of EPROM-based prototype codes. The 28F010 replaces the 15- to 20-minute ultraviolet erasure with one-second electrical erasure. Electrical chip-erase and reprogramming occur in the same workstation or PROM-programmer socket.

Diagnostics, performed at subassembly or final assembly stages, often require the socketing of EPROMs. Socketed test codes are ultimately replaced with EPROMs containing the final program. With electrical chip-erase and reprogramming, the 28F010 is soldered to the circuit board. Test codes are programmed into the 28F010 as it resides on the circuit board. Ultimately, the final code can be downloaded to the device. The 28F010's in-circuit alterability eliminates unnecessary handling and less-reliable socketed connections, while adding greater test flexibility.

Material and labor costs associated with code changes increase at higher levels of system integration—the most costly being code updates after sale Code "bugs", or the desire to augment system functionality, prompt after-sale code updates. Field revisions to EPROM-based code require the removal of EPROM components or entire boards. The service technician performs the twenty-minute ultraviolet erasure and reprogramming on-site, or returns boards to the factory for rework. An alternate ap-

proach is to use one-time programmable EPROMs. The service technician removes the "old" devices and replaces them with updated versions. The used components are discarded.

Designing with the in-circuit alterable 28F010 eliminates socketed memories, reduces overall material costs, and drastically cuts the labor costs associated with code updates. With the 28F010, code updates are implemented locally via an edge-connector, or remotely over a serial communication link.

The 28F010's electrical chip-erase, byte reprogrammability, and complete nonvolatility fit well with data accumulation needs. Electrical chip-erase gives the designer a "blank slate" in which to log data. Data can be periodically off-loaded for analysis—erasing the state and repeating the cycle. Or, multiple devices can maintain a "rolling window" of accumulated data.

With high density, nonvolatility, and extended cycling capability, the 28F010 offers an innovative alternative for mass storage. Integrating main memory and backup storage functions into directly executable flash memory boosts system performance, shrinks system size, and cuts power consumption. Reliability exceeds that of electromechanical media, with greater durability in extreme environmental conditions.

A high degree of on-chip feature integration simplifies memory-to-processor interfacing. Figure 3 depicts two 28F010s tied to the 80C186 system bus. The 28F010's architecture minimizes interface circuitry needed for complete in-circuit updates of memory contents.

With cost-effective in-system reprogramming and extended cycling capability, the 28F010 fills the functionality gap between traditional EPROMs and EPROMs. EPROM-compatible specifications, straightforward interfacing, and in-circuit alterability allows designers to easily augment memory flexibility and satisfy the need for updatable nonvolatile storage in today's designs.

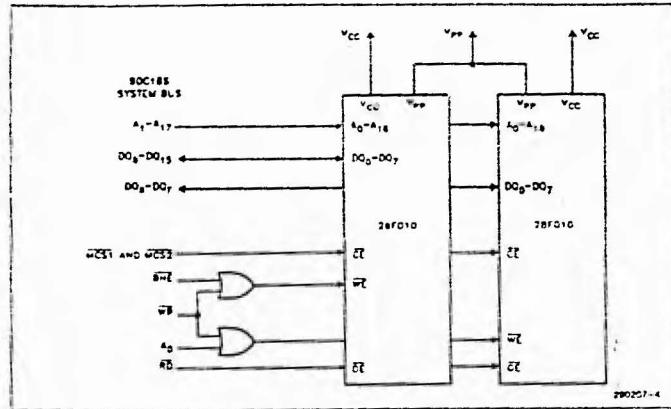


Figure 3. 28F010 in a 80C186 System

**PRINCIPLES OF OPERATION**

Flash-memory augments EPROM functionality with in-circuit electrical erasure and reprogramming. The 28F010 introduces a command register to manage the new functionality. The command register allows for: 100% TTL-level control inputs; fixed power supplies during erasure and programming; and maximum EPROM compatibility.

In the absence of high voltage on the Vpp pin, the 28F010 is a read-only memory. Manipulation of the external memory-control pins yields the standard EPROM read, standby, output disable, and intelligent identifier™ operations.

The same EPROM read, standby, and output disable operations are available when high voltage is applied to the Vpp pin. In addition, high voltage on Vpp enables erasure and programming of the device. All

functions associated with altering memory contents—intelligent identifier, erase, erase verify, program, and program verify—are accessed via the command register.

Commands are written to the register using standard microprocessor write timings. Register contents serve as input to an internal state-machine which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for programming or erase operations. With the appropriate command written to the register, standard microprocessor read timings output array data, access the intelligent identifier codes, or output data for erase and program verification.

The command register is only alterable when Vpp is at high voltage. Depending upon the application, the system designer may choose to make the Vpp power supply switchable—available only when memory updates are desired. When high voltage is removed,

Table 2. 28F010 Bus Operations

		Pins		Vpp <sup>(1)</sup>	A <sub>0</sub>	A <sub>1</sub>	CE	OE	WE	DO <sub>0</sub> -DO <sub>7</sub>
READ-ONLY	Read	VppL	A <sub>0</sub>	A <sub>1</sub>	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>PH</sub>	Data Out		
	Output Disable	VppH	X	X	V <sub>IL</sub>	V <sub>PH</sub>	V <sub>PH</sub>	Tri-State		
	Standby	VppL	X	X	V <sub>PH</sub>	X	X	Tri-State		
	Intelligent Identifier™ (Manufacturer)	VppH	V <sub>IL</sub>	V <sub>PH</sub> <sup>(2)</sup>	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>PH</sub>	Data = 89H		
	Intelligent Identifier™ (Device)	VppH	V <sub>PH</sub>	V <sub>PH</sub> <sup>(2)</sup>	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>PH</sub>	Data = B4H		
READ/WRITE	Read	VppH	A <sub>0</sub>	A <sub>1</sub>	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>PH</sub>	Data Out <sup>(4)</sup>		
	Output Disable	VppH	X	X	V <sub>IL</sub>	V <sub>PH</sub>	V <sub>PH</sub>	Tri-State		
	Standby <sup>(5)</sup>	VppH	X	X	V <sub>PH</sub>	X	X	Tri-State		
	Write	VppH	A <sub>0</sub>	A <sub>1</sub>	V <sub>IL</sub>	V <sub>PH</sub>	V <sub>IL</sub>	Data In <sup>(6)</sup>		

**NOTES:**

- VppL may be ground, a no-connect with a resistor tied to ground, or  $V_{CC} + 2.0V$ . VppH is the programming voltage specified for the device. Refer to DC Characteristics. When Vpp = VppH, memory contents can be read but not written or erased.
- Manufacturer and device codes may also be accessed via a command register write sequence. Refer to Table 3. All other addresses low.
- V<sub>IO</sub> is the intelligent identifier high voltage. Refer to DC Characteristics.
- Read operations with Vpp = VppH may access array data or the intelligent identifier™ codes.
- With Vpp at high voltage, the standby current equals Icc = Icc (standby).
- Refer to Table 3 for valid Data In during a write operation.
- X can be V<sub>IL</sub> or V<sub>PH</sub>.

the contents of the register default to the read command, making the 28F010 a read-only memory. Memory contents cannot be altered.

Or, the system designer may choose to "hardwire" Vpp, making the high voltage supply constantly available. In this instance, all operations are performed in conjunction with the command register. The 28F010 is designed to accommodate either design practice, and to encourage optimization of the processor-memory interface.

**BUS OPERATIONS**

**Read**

The 28F010 has two control functions, both of which must be logically active, to obtain data at the outputs. Chip-Enable (CE) is the power control and should be used for device selection. Output-Enable (OE) is the output control and should be used to gate data from the output pins, independent of device selection. Figure 6 illustrates read timing waveforms.

When Vpp is high (VppH), the read operation can be used to access array data, to output the intelligent identifier™ codes, and to access data for program/erase verification. When Vpp is low (VppL), the read operation can only access the array data.

**Output Disable**

With Output-Enable at a logic-high level (V<sub>PH</sub>), output from the device is disabled. Output pins are placed in a high-impedance state.

**Standby**

With Chip-Enable at a logic-high level, the standby operation disables most of the 28F010's circuitry and substantially reduces device power consumption. The outputs are placed in a high-impedance state, independent of the Output-Enable signal. If the 28F010 is deselected during erasure, programming, or program/erase verification, the device draws active current until the operation is terminated.

**Intelligent Identifier™ Operation**

The intelligent identifier operation outputs the manufacturer code (89H) and device code (B4H). Programming equipment automatically matches the device with its proper erase and programming algorithms.

FALLA DE ORIGEN



With Chip-Enable and Output-Enable at a logic low level, raising A9 to high voltage V<sub>DD</sub> (see DC Characteristics) activates the operation. Data read from locations 0000H and 0001H represent the manufacturer's code and the device code, respectively.

The manufacturer- and device-codes can also be read via the command register, for instances where the 28F010 is erased and reprogrammed in the target system. Following a write of 90H to the command register, a read from address location 0000H outputs the manufacturer code (89H). A read from address 0001H outputs the device code (84H).

**Write**

Device erasure and programming are accomplished via the command register, when high voltage is applied to the V<sub>pp</sub> pin. The contents of the register serve as input to the internal state-machine. The state-machine outputs dictate the function of the device.

The command register itself does not occupy an addressable memory location. The register is a latch

used to store the command, along with address and data information needed to execute the command.

The command register is written by bringing Write-Enable to a logic-low level (V<sub>IL</sub>), while Chip-Enable is low. Addresses are latched on the falling edge of Write-Enable, while data is latched on the rising edge of the Write-Enable pulse. Standard microprocessor write timings are used.

Refer to A.C. Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

**COMMAND DEFINITIONS**

When low voltage is applied to the V<sub>pp</sub> pin, the contents of the command register default to 00H, enabling read-only operations.

Placing high voltage on the V<sub>pp</sub> pin enables read/write operations. Device operations are selected by writing specific data patterns into the command register. Table 3 defines these 28F010 register commands.

Table 3. Command Definitions

Command	Bus Cycles	First Bus Cycle		Second Bus Cycle	
		Req'd Operation <sup>(1)</sup>	Address <sup>(2)</sup> Data <sup>(3)</sup>	Operation <sup>(4)</sup>	Address <sup>(2)</sup> Data <sup>(3)</sup>
Read Memory	1	Write	X 00H		
Read Intelligent Identifier™ Codes <sup>(4)</sup>	2	Write	X 90H	Read	IA ID
Set-up Erase/Erase <sup>(5)</sup>	2	Write	X 20H	Write	X 26H
Erase Verify <sup>(5)</sup>	2	Write	EA 40H	Read	X EVD
Set-up Program/Program <sup>(6)</sup>	2	Write	X 40H	Write	PA PD
Program Verify <sup>(6)</sup>	2	Write	X C0H	Read	X PVD
Reset <sup>(7)</sup>	2	Write	X FFH	Write	X FFH

**NOTES:**

1. Bus operations are defined in Table 2.
2. IA = Identifier address: 00H for manufacturer code, 01H for device code  
EA = Address of memory location to be read during erase verify  
PA = Address of memory location to be programmed  
Addresses are latched on the falling edge of the Write-Enable pulse
3. ID = Data read from location IA during device identification (Mfr = 89H, Device = 84H)  
EVD = Data read from location EA during erase verify  
PVD = Data read from location PA during program verify. PA is latched on the Program command
4. Following the Read Intelligent ID command, two read operations access manufacturer and device codes
5. Figure 4 illustrates the Quick-Erase™ Algorithm.
6. Figure 5 illustrates the Quick-Program™ Algorithm.
7. The second bus cycle must be followed by the desired command register write

**Read Command**

While V<sub>pp</sub> is high, for erasure and programming, memory contents can be accessed via the read command. The read operation is initiated by writing 00H into the command register. Microprocessor READ cycles retrieve array data. The device remains enabled for reads until the command register contents are altered.

The default contents of the register upon V<sub>pp</sub> power-up is 00H. This default value ensures that no spurious alteration of memory contents occurs during the V<sub>pp</sub> power transition. Where the V<sub>pp</sub> supply is hard-wired to the 28F010, the device powers-up and remains enabled for reads until the command register contents are changed. Refer to the A.C. Read Characteristics and Waveforms for specific timing parameters.

**Intelligent Identifier™ Command**

Flash-memories are intended for use in applications where the local CPU alters memory contents. As such, manufacturer- and device-codes must be accessible while the device resides in the target system. PROM programmers typically access signature codes by raising A9 to a high voltage. However, multiplexing high voltage onto address lines is not a desired system-design practice.

The 28F010 contains an Intelligent Identifier operation to supplement traditional PROM-programming methodology. The operation is initiated by writing 90H into the command register. Following the command write, a read cycle from address 0000H retrieves the manufacturer code of 89H. A read cycle from address 0001H returns the device code of 84H. To terminate the operation, it is necessary to write another valid command into the register.

**Set-up Erase/Erase Commands**

Set-up Erase is a command-only operation that stages the device for electrical erasure of all bytes in the array. The set-up erase operation is performed by writing 20H to the command register.

To commence chip-erasure, the erase command (20H) must again be written to the register. The erase operation begins with the rising edge of the Write-Enable pulse and terminates with the rising edge of the next Write-Enable pulse (i.e., Erase-Verify Command).

This two-staged sequence of set-up followed by execution ensures that memory contents are not accidentally erased. Also, chip-erasure can only occur when

high voltage is applied to the V<sub>pp</sub> pin. In the absence of this high voltage, memory contents are protected against erasure. Refer to A.C. Erase Characteristics and Waveforms for specific timing parameters.

**Erase-Verify Command**

The erase command erases all bytes of the array in parallel. After each erase operation, all bytes must be verified. The erase-verify operation is initiated by writing 40H into the command register. The address for the byte to be verified must be supplied as it is latched on the falling edge of the Write-Enable pulse. The register write terminates the erase operation with the rising edge of its Write-Enable pulse.

The 28F010 applies an internally-generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are erased.

The erase-verify command must be written to the command register prior to each byte verification to latch its address. The process continues for each byte in the array until a byte does not return FFH data or the last address is accessed.

In the case where the data read is not FFH, another erase operation is performed. (Refer to Set-up Erase/Erase). Verification then resumes from the address of the last-verified byte. Once all bytes in the array have been verified, the erase step is complete. The device can be programmed. At this point, the verify operation is terminated by writing a valid command (e.g. Program Set-up) to the command register. Figure 5, the Quick-Erase™ algorithm, illustrates how commands and bus operations are combined to perform electrical erasure of the 28F010. Refer to A.C. Erase Characteristics and Waveforms for specific timing parameters.

**Set-up Program/Program Commands**

Set-up program is a command-only operation that stages the device for byte programming. Writing 40H into the command register performs the set-up operation.

Once the program set-up operation is performed, the next Write-Enable pulse causes a transition to an active programming operation. Addresses are internally latched on the falling edge of the Write-Enable pulse. Data is internally latched on the rising edge of the Write-Enable pulse. The rising edge of Write-Enable also begins the programming operation. The programming operation terminates with the next rising edge of Write-Enable, used to write the program-verify command. Refer to A.C. Program-

ing Characteristics and Waveforms for specific timing parameters.

**Program-Verify Command**

The 28F010 is programmed on a byte-by-byte basis. Byte programming may occur sequentially or at random. Following each programming operation, the byte just programmed must be verified.

The program-verify operation is initiated by writing 00H into the command register. The register write terminates the programming operation with the rising edge of its Write-Enable pulse. The program-verify operation stages the device for verification of the byte last programmed. No new address information is latched.

The 28F010 applies an internally-generated margin voltage to the byte. A microprocessor read cycle outputs the data. A successful comparison between the programmed byte and true data means that the byte is successfully programmed. Programming then proceeds to the next desired byte location. Figure 4, the 28F010 Quick-Pulse Programming™ algorithm, illustrates how commands are combined with bus operations to perform byte programming. Refer to A.C. Programming Characteristics and Waveforms for specific timing parameters.

**Reset Command**

A reset command is provided as a means to safely abort the erase- or program-command sequences. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

**EXTENDED ERASE/PROGRAM CYCLING**

EEPROM cycling failures have always concerned users. The high electrical field required by thin oxide EEPROMs for tunneling can literally tear apart the oxide at defect regions. To combat this, some suppliers have implemented redundancy schemes, reducing cycling failures to insignificant levels. However, redundancy requires that cell size be doubled—an expensive solution.

Intel has designed extended cycling capability into its ETOX-II flash memory technology. Resulting improvements in cycling reliability come without increasing memory cell size or complexity. First, an advanced tunnel oxide increases the charge carrying ability ten-fold. Second, the oxide area per cell subjected to the tunneling electric field is one-tenth that of common EEPROMs, minimizing the probability

of oxide defects in the region. Finally, the peak electric field during erasure is approximately 2 MV/cm lower than EEPROM. The lower electric field greatly reduces oxide stress and the probability of failure—increasing time to wearout by a factor of 100,000,000.

The 28F010 is specified for a minimum of 10,000 program/erase cycles. The device is programmed and erased using Intel's Quick-Pulse Programming™ and Quick-Erase™ algorithms. Intel's algorithmic approach uses a series of operations (pulses), along with byte verification, to completely and reliably erase and program the device.

**QUICK-PULSE PROGRAMMING™ ALGORITHM**

The Quick-Pulse Programming algorithm uses programming operations of 10 μs duration. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The algorithm allows for up to 25 programming operations per byte, although most bytes verify on the first or second operation. The entire sequence of programming and byte verification is performed with V<sub>pp</sub> at high voltage. Figure 4 illustrates the Quick-Pulse Programming algorithm.

**QUICK-ERASE™ ALGORITHM**

Intel's Quick-Erase algorithm yields fast and reliable electrical erasure of memory contents. The algorithm employs a closed-loop flow, similar to the Quick-Pulse Programming™ algorithm, to simultaneously remove charge from all bits in the array.

Erasure begins with a read of memory contents. The 28F010 is erased when shipped from the factory. Reading FFH data from the device would immediately be followed by device programming.

For devices being erased and reprogrammed, uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished, using the Quick-Pulse Programming algorithm, in approximately two seconds.

Erasure execution then continues with an initial erase operation. Erase verification (data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. With each erase operation, an increasing number of bytes verify to the erased state. Erase efficiency may be improved by storing the address of the last byte verified in a register. Following the next erase operation, verification starts at that stored address location. Erasure typically occurs in one second. Figure 5 illustrates the Quick-Erase algorithm.

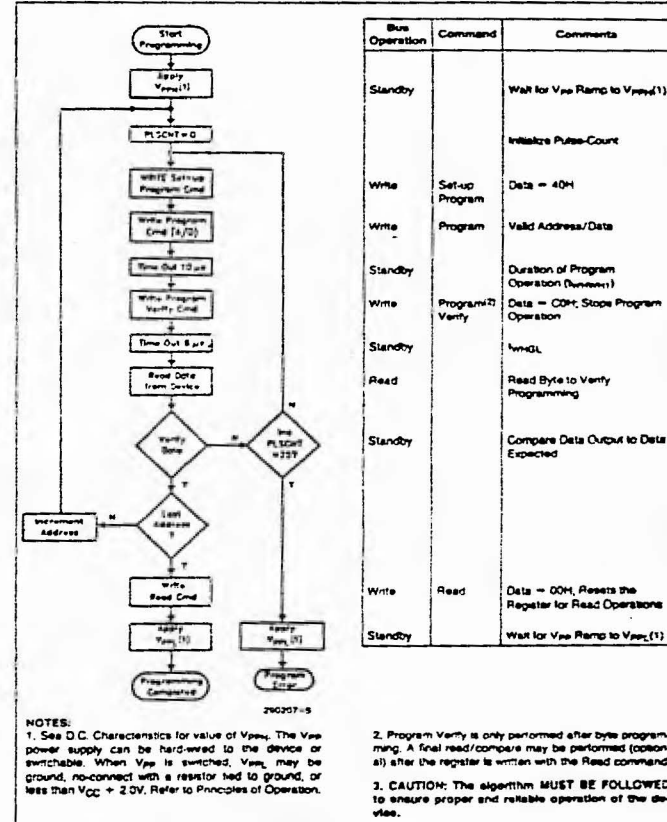


Figure 4. 28F010 Quick-Pulse Programming™ Algorithm

FALLA DE ORIGEN

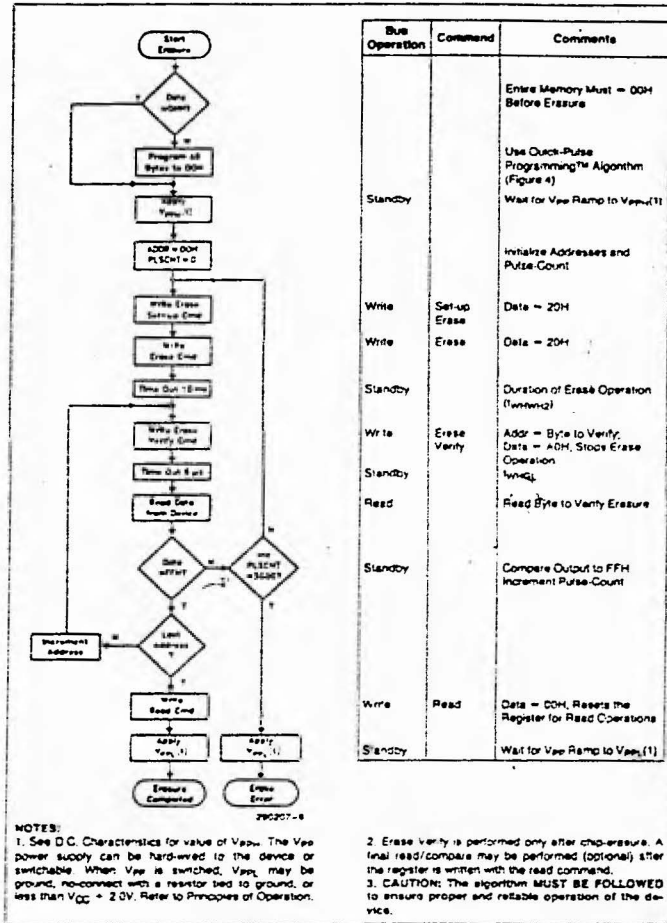


Figure 5. 28F010 Quick-Erase™ Algorithm  
5-57

DESIGN CONSIDERATIONS

Two-Line Output Control

Flash-memories are often used in larger memory arrays. Intel provides two read-control inputs to accommodate multiple memory connections. Two-line control provides for:

- a. the lowest possible memory power dissipation and,
- b. complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address-decoder output should drive chip-enable, while the system's read signal controls all flash-memories and other parallel memories. This assures that only enabled memory devices have active outputs, while deselected devices maintain the low power standby condition.

Power Supply Decoupling

Flash-memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current (Icc) issues—standby, active, and transient current peaks produced by falling and rising edges of chip-enable. The capacitive and inductive loads on the device outputs determine the magnitudes of these peaks.

Two-line control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 µF ceramic capacitor connected between VCC and VSS, and between Vpp and VSS.

Place the high-frequency, low-inductance capacitors as close as possible to the devices. Also, for every eight devices, a 4.7 µF electrolytic capacitor should be placed at the array's power supply connection, between VCC and VSS. The bulk capacitor will overcome voltage sumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

Vpp Trace on Printed Circuit Boards

Programming flash-memories, while they reside in the target system, requires that the printed circuit board designer pay attention to the Vpp power supply trace. The Vpp pin supplies the memory cell current for programming. Use similar trace widths and layout considerations given the VCC power bus. Adequate Vpp supply traces and decoupling will decrease Vpp voltage spikes and overshoots.

Power Up/Down Sequencing

The 28F010 is designed to offer protection against accidental erasure or programming, caused by spurious system-level signals that may exist during power transitions. The 28F010 powers-up in its read-only state. Also, with its control register architecture, alteration of memory contents only occurs after successful completion of the two-step command sequence. While these precautions are sufficient for most applications, it is recommended that VCC reach its steady-state value before raising Vpp above VCC + 2.0V. In addition, upon powering-down, Vpp should be below VCC + 2.0V, before lowering VCC.

FALLA DE ORIGEN

**ABSOLUTE MAXIMUM RATINGS\***

Operating Temperature	
During Read	0°C to +70°C <sup>(1)</sup>
During Erase/Program	0°C to +70°C
Temperature Under Bias	-10°C to +80°C
Storage Temperature	-65°C to +125°C
Voltage on Any Pin with Respect to Ground	-2.0V to +7.0V <sup>(2)</sup>
Voltage on Pin A <sub>9</sub> with Respect to Ground	-2.0V to +13.5V <sup>(2,3)</sup>
V <sub>pp</sub> Supply Voltage with Respect to Ground	
During Erase/Program	-2.0V to +14.0V <sup>(2,3)</sup>
V <sub>cc</sub> Supply Voltage with Respect to Ground	-2.0V to +7.0V <sup>(2)</sup>
Output Short Circuit Current	100 mA <sup>(4)</sup>

\*Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

NOTICE: Specifications contained within the following tables are subject to change.

**NOTES:**

- Operating temperature is for commercial product defined by this specification.
- Minimum D.C. input voltage is -0.5V. During transitions, inputs may undershoot to -2.0V for periods less than 20 ns. Maximum D.C. voltage on output pins is V<sub>CC</sub> + 0.5V, which may overshoot to V<sub>CC</sub> + 2.0V for periods less than 20 ns.
- Maximum D.C. voltage on A<sub>9</sub> or V<sub>pp</sub> may overshoot to +14.0V for periods less than 20 ns.
- Output shorted for no more than one second. No more than one output shorted at a time.

**OPERATING CONDITIONS**

Symbol	Parameter	Limits		Unit	Comments
		Min	Max		
T <sub>A</sub>	Operating Temperature	0	70	°C	For Read-Only and Read/Write Operations
V <sub>CC</sub>	V <sub>CC</sub> Supply Voltage	4.50	5.50	V	

**D.C. CHARACTERISTICS—TTL/NMOS COMPATIBLE**

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
I <sub>LI</sub>	Input Leakage Current		±1.0	µA	V <sub>CC</sub> = V <sub>CC</sub> Max V <sub>IN</sub> = V <sub>CC</sub> or V <sub>SS</sub>
I <sub>LO</sub>	Output Leakage Current		±10	µA	V <sub>CC</sub> = V <sub>CC</sub> Max V <sub>OUT</sub> = V <sub>CC</sub> or V <sub>SS</sub>
I <sub>CCS</sub>	V <sub>CC</sub> Standby Current		1.0	mA	V <sub>CC</sub> = V <sub>CC</sub> Max CE = V <sub>IN</sub>
I <sub>CC1</sub> <sup>(1)</sup>	V <sub>CC</sub> Active Read Current		30	mA	V <sub>CC</sub> = V <sub>CC</sub> Max, CE = V <sub>IL</sub> f = 5 MHz, I <sub>OUT</sub> = 0 mA
I <sub>CC2</sub> <sup>(1)</sup>	V <sub>CC</sub> Programming Current		30	mA	Programming in Progress
I <sub>CC3</sub> <sup>(1)</sup>	V <sub>CC</sub> Erase Current		30	mA	Erase in Progress
I <sub>ppS</sub>	V <sub>pp</sub> Leakage Current		±10	µA	V <sub>pp</sub> = V <sub>ppL</sub>

**D.C. CHARACTERISTICS—TTL/NMOS COMPATIBLE (Continued)**

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
I <sub>pp1</sub>	V <sub>pp</sub> Read Current		200	µA	V <sub>pp</sub> = V <sub>ppH</sub>
I <sub>pp2</sub> <sup>(2)</sup>	V <sub>pp</sub> Programming Current		±10	mA	V <sub>pp</sub> = V <sub>ppL</sub>
I <sub>pp3</sub> <sup>(2)</sup>	V <sub>pp</sub> Erase Current		30	mA	V <sub>pp</sub> = V <sub>ppH</sub> Programming in Progress
V <sub>IL</sub>	Input Low Voltage	-0.5	0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0	V <sub>CC</sub> + 0.5	V	
V <sub>OL</sub>	Output Low Voltage		0.45	V	I <sub>OL</sub> = 5.8 mA V <sub>CC</sub> = V <sub>CC</sub> Min
V <sub>OHI</sub>	Output High Voltage	2.4		V	I <sub>OHI</sub> = -2.5 mA V <sub>CC</sub> = V <sub>CC</sub> Min
V <sub>ID</sub>	A <sub>9</sub> Intelligent Identifier™ Voltage	11.50	13.00	V	
I <sub>ID</sub>	A <sub>9</sub> Intelligent Identifier™ Current		500	µA	A <sub>9</sub> = V <sub>ID</sub>
V <sub>ppL</sub>	V <sub>pp</sub> during Read-Only Operations	0.00	V <sub>CC</sub> + 2.0V	V	NOTE: Erase/Program are inhibited when V <sub>pp</sub> = V <sub>ppL</sub>
V <sub>ppH</sub>	V <sub>pp</sub> during Read/Write Operations	11.40	12.60	V	

**D.C. CHARACTERISTICS—CMOS COMPATIBLE**

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
I <sub>LI</sub>	Input Leakage Current		±1.0	µA	V <sub>CC</sub> = V <sub>CC</sub> Max V <sub>IN</sub> = V <sub>CC</sub> or V <sub>SS</sub>
I <sub>LO</sub>	Output Leakage Current		±10	µA	V <sub>CC</sub> = V <sub>CC</sub> Max V <sub>OUT</sub> = V <sub>CC</sub> or V <sub>SS</sub>
I <sub>CCS</sub>	V <sub>CC</sub> Standby Current		100	µA	V <sub>CC</sub> = V <sub>CC</sub> Max CE = V <sub>CC</sub> ± 0.2V
I <sub>CC1</sub> <sup>(1)</sup>	V <sub>CC</sub> Active Read Current		30	mA	V <sub>CC</sub> = V <sub>CC</sub> Max, CE = V <sub>IL</sub> f = 5 MHz, I <sub>OUT</sub> = 0 mA
I <sub>CC2</sub> <sup>(1)</sup>	V <sub>CC</sub> Programming Current		30	mA	Programming in Progress
I <sub>CC3</sub> <sup>(1)</sup>	V <sub>CC</sub> Erase Current		30	mA	Erase in Progress
I <sub>ppS</sub>	V <sub>pp</sub> Leakage Current		±10	µA	V <sub>pp</sub> = V <sub>ppL</sub>

D.C. CHARACTERISTICS—CMOS COMPATIBLE (Continued)

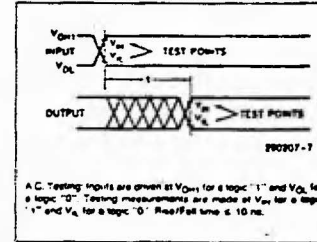
Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
$I_{PP1}$	$V_{PP}$ Read Current		200 $\pm 10$	$\mu A$	$V_{PP} = V_{PPH}$ $V_{PP} = V_{PPL}$
$I_{PP2(2)}$	$V_{PP}$ Programming Current		30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
$I_{PP3(2)}$	$V_{PP}$ Erase Current		30	mA	$V_{PP} = V_{PPH}$ Erase in Progress
$V_L$	Input Low Voltage	-0.5	0.8	V	
$V_{IH}$	Input High Voltage	$0.7 V_{CC}$	$V_{CC} + 0.5$	V	
$V_{OL}$	Output Low Voltage		0.5	V	$I_{OL} = 5.8 \text{ mA}$ $V_{CC} = V_{CC \text{ Min}}$
$V_{OH1}$	Output High Voltage	$0.65 V_{CC}$		V	$I_{OH} = -2.5 \text{ mA}, V_{CC} = V_{CC \text{ Min}}$
$V_{OH2}$		$V_{CC} - 0.4$		V	$I_{OH} = -100 \mu A, V_{CC} = V_{CC \text{ Min}}$
$V_{ID}$	$A_9$ Intelligent Identifier™ Voltage	11.50	13.00	V	
$I_{ID}$	$A_9$ Intelligent Identifier™ Current		500	$\mu A$	$A_9 = V_{ID}$
$V_{PPL}$	$V_{PP}$ during Read-Only Operations	0.00	$V_{CC} + 2.0V$	V	NOTE: Erase/Programs are inhibited when $V_{PP} = V_{PPL}$
$V_{PPH}$	$V_{PP}$ during Read/Write Operations	11.40	12.60	V	

CAPACITANCE(2)  $T_A = 25^\circ C, f = 1.0 \text{ MHz}$

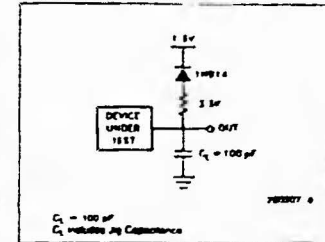
Symbol	Parameter	Limits		Unit	Conditions
		Min	Max		
$C_{IN}$	Address/Control Capacitance		6	pF	$V_{IN} = 0V$
$C_{OUT}$	Output Capacitance		12	pF	$V_{OUT} = 0V$

- NOTES:  
 1. Active  $I_{CC}$  current of a typical device is 12 mA with nominal  $V_{CC}$  at room temperature.  
 2. Active  $I_{PP}$  current of a typical device is 10 mA with nominal  $V_{PP}$  at room temperature.  
 3. Sampled, not 100% tested.

A.C. TESTING INPUT/OUTPUT WAVEFORM



A.C. TESTING LOAD CIRCUIT



A.C. TEST CONDITIONS

- Input Rise and Fall Times (10% to 90%) ..... 10 ns  
 Input Pulse Levels .....  $V_{OL}$  and  $V_{OH1}$   
 Input Timing Reference Level .....  $V_{IL}$  and  $V_{IH}$   
 Output Timing Reference Level .....  $V_{OL}$  and  $V_{OH}$

A.C. CHARACTERISTICS—Read-Only Operations

Version		28F010-135P1C4		28F010-150P1C4		28F010-200P1C4		Unit
Symbol	Characteristic	Min	Max	Min	Max	Min	Max	
$t_{AVAV}/t_{AC}$	Read Cycle Time	135		150		200		ns
$t_{ELOW}/t_{CE}$	Chip Enable Access Time		135		150		200	ns
$t_{AVOV}/t_{ACC}$	Address Access Time		135		150		200	ns
$t_{GLOW}/t_{OE}$	Output Enable Access Time		50		55		60	ns
$t_{ELOW}/t_{LZ}$	Chip Enable to Output in Low Z	0		0		0		ns
$t_{GLOW}/t_{OLZ}$	Output Enable to Output in Low Z	0		0		0		ns
$t_{GHOZ}/t_{OH}$	Output Disable to Output in High Z		30		35		40	ns
$t_{OH}$	Output Hold from Address, CE, or OE Change(1)	0		0		0		ns
$t_{WHGL}$	Write Recovery Time before Read	6		6		6		$\mu s$

- NOTES:  
 1. Whichever occurs first.  
 2. Rise/Fall Time  $\leq 10 \text{ ns}$ .

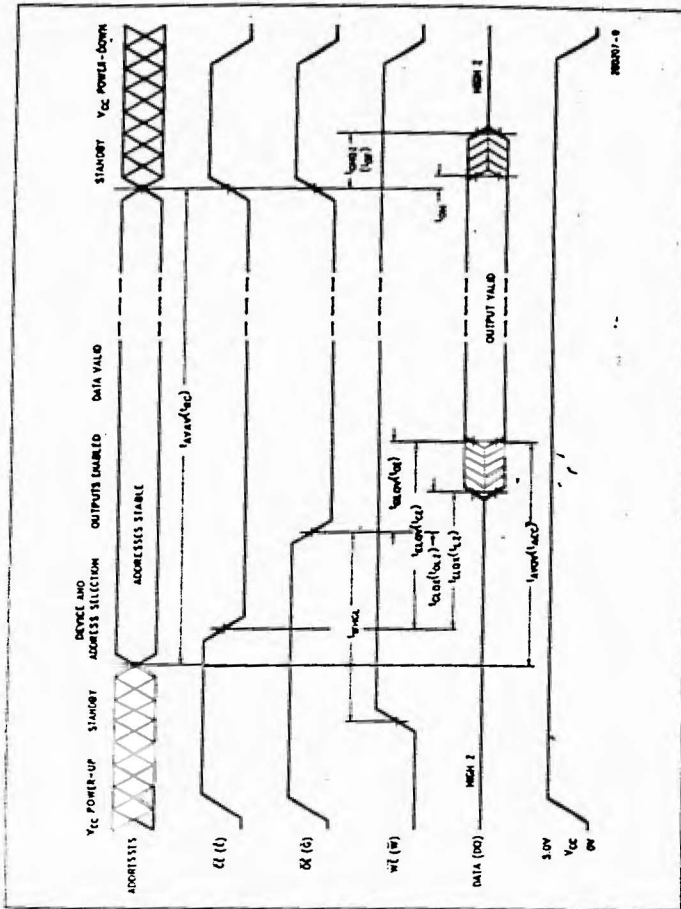


Figure 6. A.C. Waveforms for Read Operations

5-68

A.C. CHARACTERISTICS—Write/Erase/Program Operations<sup>(1)</sup>

Version		28F010-128P1C4		28F010-160P1C4		28F010-256P1C4		Unit
Symbol	Characteristic	Min	Max	Min	Max	Min	Max	
t <sub>AVW</sub> /t <sub>WC</sub>	Write Cycle Time	135		150		200		ns
t <sub>AVW</sub> /t <sub>AS</sub>	Address Set-Up Time	0		0		0		ns
t <sub>WLAX</sub> /t <sub>AH</sub>	Address Hold Time	60		60		75		ns
t <sub>DVW</sub> /t <sub>DS</sub>	Data Set-Up Time	50		50		50		ns
t <sub>DVW</sub> /t <sub>DH</sub>	Data Hold Time	10		10		10		ns
t <sub>WHGL</sub>	Write Recovery Time before Read	6		6		6		µs
t <sub>RHWL</sub>	Read Recovery Time before Write	0		0		0		µs
t <sub>CEW</sub> /t <sub>CS</sub>	Chip Enable Set-Up Time before Write	20		20		20		ns
t <sub>WEH</sub> /t <sub>CH</sub>	Chip Enable Hold Time	0		0		0		ns
t <sub>WLW</sub> /t <sub>WP</sub>	Write Pulse Width <sup>(2)</sup>	50		50		60		ns
t <sub>ELW</sub>	Alternative Write <sup>(2)</sup> Pulse Width	70		70		80		ns
t <sub>WWW</sub> /t <sub>WPH</sub>	Write Pulse Width High	20		20		20		ns
t <sub>WWW1</sub>	Duration of Programming Operation	10	25	10	25	10	25	µs
t <sub>WWW2</sub>	Duration of Erase Operation	9.5	10.5	9.5	10.5	9.5	10.5	ms
t <sub>VPEL</sub>	V <sub>pp</sub> Set-Up Time to Chip Enable Low	100		100		100		ns

NOTES:

- Read timing characteristics during read/write operations are the same as during read-only operations. Refer to A.C. Characteristics for Read-Only Operations.
- Chip-Enable Controlled Writes. Write operations are driven by the valid combination of Chip-Enable and Write-Enable in systems where Chip-Enable defines the write pulse width (within a longer write Enable timing waveform) at set-up, hold, and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.
- Read/Fall time ≤ 10 ns.

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Limits			Unit	Comments
	Min	Typ	Max		
Chip Erase Time		0.5 <sup>(1)</sup>	30	Sec	Excludes OOH Programming Prior to Erasure
Chip Program Time		2 <sup>(1)</sup>	24 <sup>(2)</sup>	Sec	Excludes System-Level Overhead
Erase/Program Cycles <sup>(3)</sup>	10,000	100,000		Cycles	

NOTES:

- 25°C, 12 DV V<sub>pp</sub>, 10,000 Cycles
- Maximum byte programming time excluding system overhead is 16 µsec (10 µsec program + 6 µsec write recovery); write maximum is 400 µsec/byte (16 µsec + 25 loops allowed by algorithm). Max Chip programming time is specified lower than the worst case allowed by the programming algorithm since most bytes program significantly faster than the worst case byte.
- Refer to RR-80 "ETOX" Flash Memory Reliability Data Summary for typical cycling data and failure rate calculations.

5-64

FALLA DE ORIGEN

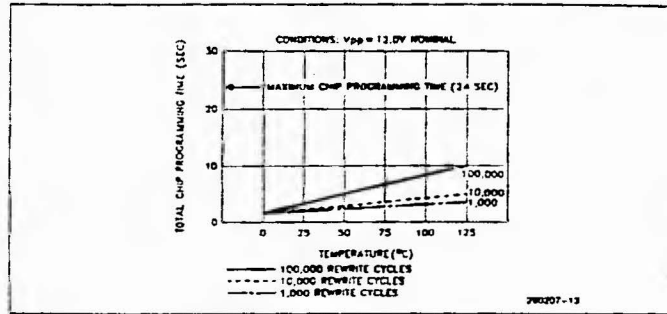


Figure 7. 28F010 Typical Programming Time vs. Temperature

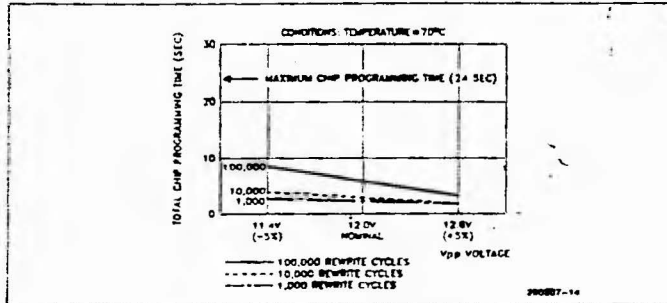


Figure 8. 28F010 Typical Programming Time vs.  $V_{pp}$  Voltage

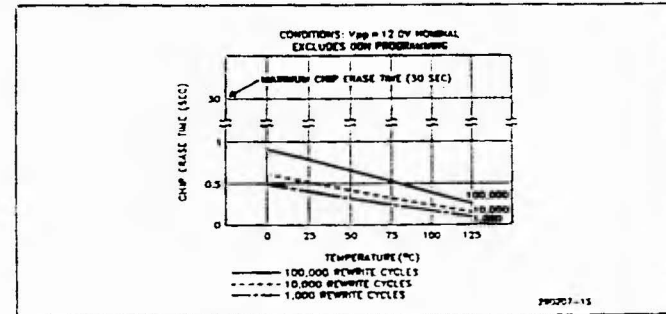


Figure 9. 28F010 Typical Erase Time vs. Temperature

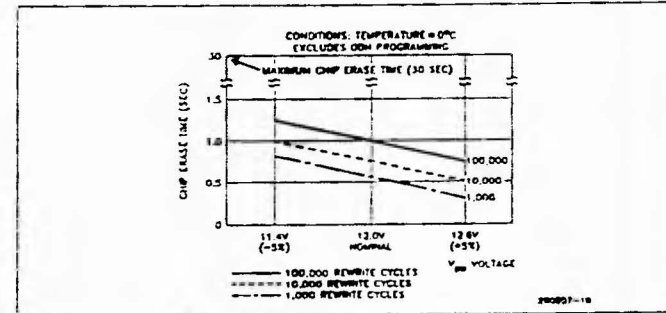


Figure 10. 28F010 Typical Erase Time vs.  $V_{pp}$  Voltage

FALLA DE UNIDAD

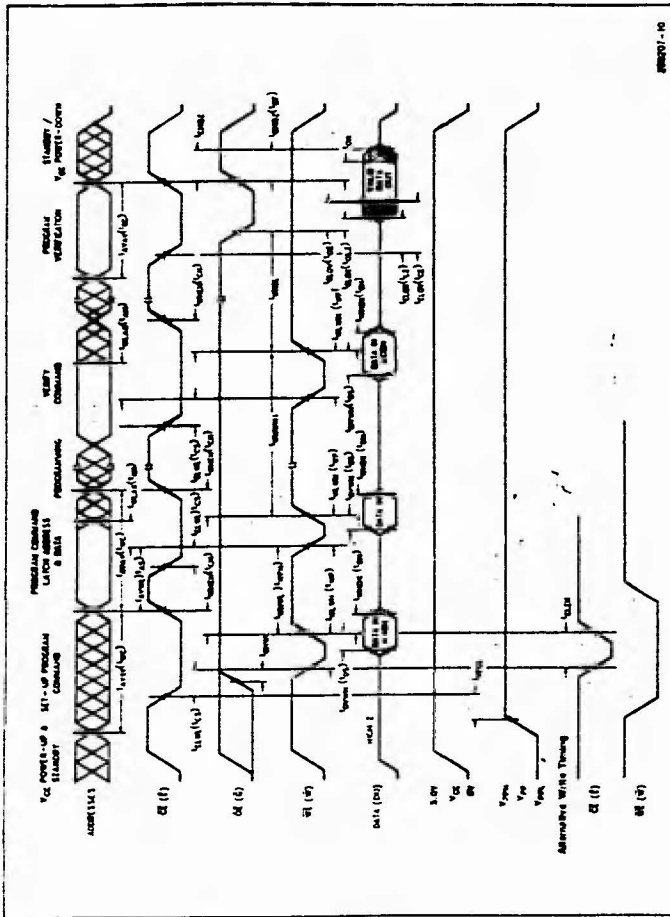


Figure 11. A.C. Waveforms for Programming Operations

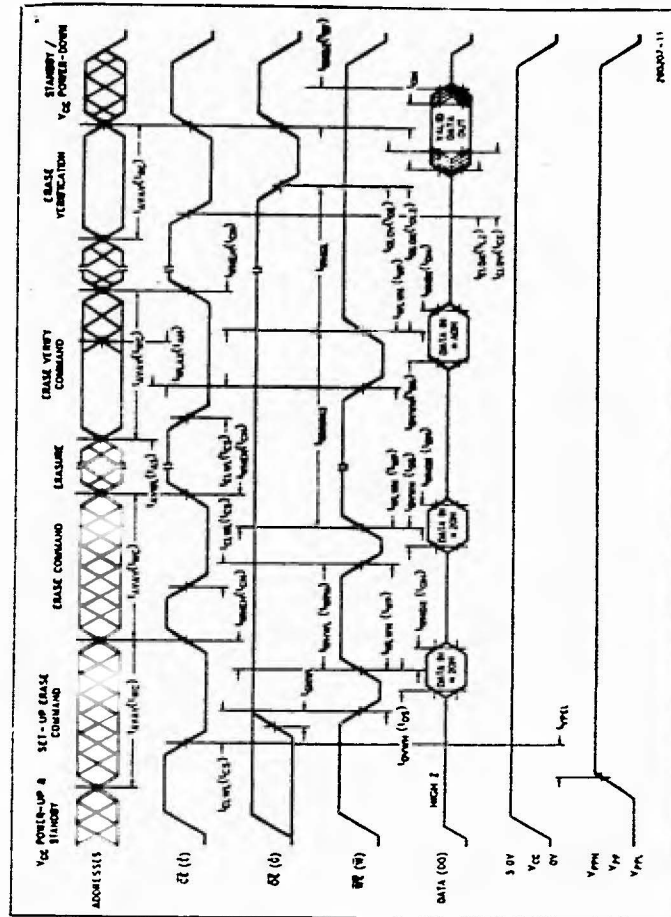
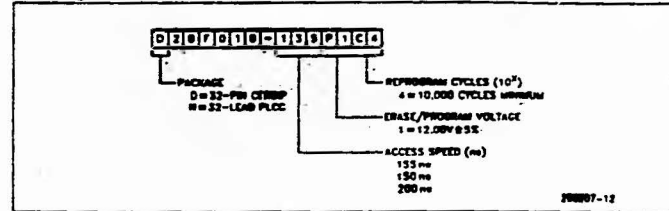


Figure 12. A.C. Waveforms for Erase Operations



Ordering Information



Valid Combinations:

D28F010-135P1C4	N28F010-135P1C4
D28F010-150P1C4	N28F010-150P1C4
D28F010-200P1C4	N28F010-200P1C4

ADDITIONAL INFORMATION

	Order Number
ER-20, "ETOX™ Flash Memory Technology"	294005
ER-24, "The Intel 28F010 Flash Memory"	294008
RR-60, "ETOX™ Flash Memory Reliability Data Summary"	293002
AP-316, "Using Flash Memory for In-System Reprogrammable Nonvolatile Storage"	292048