

20  
Lej



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERIA

DISEÑO Y CONSTRUCCION DE UN SISTEMA  
DE ADQUISICION DE DATOS BASADO EN  
EL MICROPROCESADOR 8088

T E S I S

QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N :

EDUARDO CASTILLO FUENTES  
JUAN ALBERTO IBARRA MANRIQUEZ

ASESOR: ING. RODOLFO PETERS LAMMEL



MEXICO, D. F.

FALLA DE ORIGEN

1991



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

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

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

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

## INDICE

<b>INTRODUCCION</b>	<b>1</b>
<b>CAPITULO I DESCRIPCION DEL PROYECTO</b>	<b>3</b>
1.1. Discusión del Problema	3
1.2. Características del Sistema de Adquisición de Datos	7
<b>CAPITULO II CARACTERISTICAS DEL SISTEMA DE ADQUISICION DE DATOS</b>	<b>10</b>
2.1. Amplificador Diferencial	10
2.2. Amplificador de Instrumentación	16
<b>CAPITULO III DISEÑO DEL SISTEMA DE ADQUISICION DE DATOS</b>	<b>21</b>
3.1. Descripción General	21
3.2. Descripción del Microprocesador 8088	23
3.3. Puerto Paralelo Programable 8255	44
3.4. Interrupciones	52
3.5. Temporizador Programable 8253	56
3.6. Convertidor Analógico/Digital	59
3.7. Modulación por Desplazamiento de Frecuencia	66
3.8. Interfaz en Serie USART 8251	71
3.9. Interfaz Serie RS-232C	77
3.10. Circuito Lineal 1488	80
3.11. Circuito Lineal 1489	81
3.12. Generador de Frecuencia MC14411	82

3.13. Reloj de Tiempo Real	83
3.14. Decodificación de Direcciones de Memoria y Dispositivos de Entrada/Salida	85
<b>CAPITULO IV DISPOSITIVOS AUXILIARES</b>	<b>90</b>
4.1. Teclado	90
4.2. Display de Cristal Liquido	94
4.3. Unidad de Cinta	97
<b>CAPITULO V PROGRAMAS DEL SISTEMA DE ADQUISICION DE DATOS</b>	<b>102</b>
5.1. Programa Monitor	102
5.2. Rutina de Adquisición de Datos	103
5.3. Rutina de Almacenamiento de Datos	107
5.4. Rutina de Actualización del Reloj de Tiempo Real	108
<b>CAPITULO VI RESULTADOS Y CONCLUSIONES</b>	<b>110</b>
6.1. Resultados	110
6.2. Conclusiones y Recomendaciones	117
<b>APENDICE A DIAGRAMAS DE FLUJO Y LISTADO DE PROGRAMAS</b>	<b>119</b>
<b>APENDICE B HOJAS DE DATOS</b>	<b>120</b>
<b>BIBLIOGRAFIA</b>	<b>121</b>

## INTRODUCCION

## INTRODUCCION

En el presente trabajo se describe el diseño de un Sistema de Adquisición de Datos construido de tal forma que sea compatible con una computadora personal o un sistema digital similar. El trabajo se llevó a cabo por medio del microprocesador 8088 de Intel y sus dispositivos periféricos principales, característica que le hace tener cualidades de enlace con la mayoría de los sistemas con arquitectura de microprocesadores anteriores a éste, actuales y futuros.

El propósito de este sistema es, como su nombre lo indica, obtener y procesar datos provenientes de fenómenos físicos sensados por medio de transductores analógicos, datos que son registrados en forma paralela en el momento en que ocurren por medio de un reloj de tiempo real, y almacenados dentro del mismo sistema o en un dispositivo de memoria externo, dependiendo de la cantidad de datos adquiridos.

El sistema cuenta con 24 canales analógicos de adquisición por medio de los cuales se puede manejar gran cantidad de variables de distintos tipos, en diferentes tiempos y sistemas. El trabajo que se presenta fue desarrollado en la Coordinación de Instrumentación del Instituto de Ingeniería como una alternativa a los diferentes problemas que se presentan en la adquisición de datos cuando se requiere el procesamiento de los mismos en forma automática y digital, de manera muestreada, además de un almacenamiento confiable.

El orden en que se presentan los diferentes temas es el siguiente: se describe en primer lugar un Sistema de Adquisición de este tipo. A continuación se hace una descripción del Amplificador de Instrumentación, su configuración y características de acondicionamiento de señales. Posteriormente se describe, en forma global, el Sistema Adquisidor incluyendo la operación de los circuitos e interfaces que lo conforman, así como de los dispositivos auxiliares del sistema. En la parte final se incluyen los resultados obtenidos así como los diagramas de flujo y programas que son necesarios para el funcionamiento del sistema.

## CAPITULO I

### DESCRIPCION DEL PROYECTO



## CAPITULO I

### DESCRIPCION DEL PROYECTO

#### 1.1. DISCUSION DEL PROBLEMA

La Adquisición de Datos es un factor importante dentro de las distintas ramas de la Ingeniería y la Ciencia. Debido a las características propias del fenómeno que se esté estudiando se requerirá sensar las variables involucradas para conocer su comportamiento a través del tiempo, como podría ser el caso de un proceso químico en el cual se desee estudiar el comportamiento de una sustancia cuando se encuentre bajo condiciones ambientales controladas. Por un lado estarán los instrumentos para poder medir variables físicas como temperatura, alcalinidad o Ph, concentración de componentes, volumen, etc., y por el otro cada uno de los canales del sistema conectados a los distintos aparatos. Es de notarse que estas variables serán captadas por el Sistema Adquisidor obteniendo las muestras de cada una, mientras que si se hiciera con base en un sistema manual de adquisición, habría más dificultad para captar cada muestra a través del tiempo, dado que los datos serian tomados en forma manual, imprecisa, discontinuos en el tiempo y, por la misma complejidad del proceso, con omisiones por la rapidez de los cambios en los mismos. De entre la gran cantidad de procesos que presentan características similares, se pueden mencionar los siguientes:

En un proceso de Manufactura Industrial en el cual se esté

controlando la calidad del producto tomando durante determinado tiempo un número limitado de muestras para conocer su composición durante una de las fases de fabricación del mismo.

En la rama Geofísica, para conocer las alteraciones sufridas en una región con ayuda de un acelerógrafo, con el cual los datos son registrados en forma continua durante, digamos, 6 meses.

En Ingeniería Civil, cuando se requiera conocer el asentamiento de la cortina de una presa respecto a la superficie inferior o del suelo obteniendo el desplazamiento por medio de un extensómetro.

Se podrían seguir mencionando procesos en los cuales es importante la adquisición de datos, sin embargo, se observa que hay varias características comunes a todos. En cada uno de ellos es necesario que los datos sean tomados en forma automática, confiable, registrada en tiempo real, visualizada y, en caso necesario, almacenada en un dispositivo externo. Se ve entonces que no únicamente es necesario registrar los datos en forma simple y llana sino que se deben tener las características mencionadas anteriormente para no caer en errores que distorsionen la información.

Con base en lo anterior puede establecerse que un Sistema de Adquisición de Datos, para ser confiable, debe contar con las siguientes características:

- Realizar el muestreo de una forma sencilla y rápida, sin

ocasionar alteraciones en la recolección de la información.

- Poder visualizar, con ayuda de algun dispositivo, la información que se está obteniendo, a la par del tiempo en el cual sucede.

- Manejar la información obtenida de manera segura y confiable, además de tener acceso a ella de forma clara y útil.

- Obtener muestras precisas y confiables.

Otro punto importante que se debe mencionar es el referente a la ubicación del sistema a estudiar. Si bien una computadora puede ser una alternativa para esta tarea por contar con interfaces normalizadas (RS-232) para conexión de eventos externos, no cuenta con un acondicionador de señales que permita la adecuación de los datos de entrada, ni con un convertidor Analógico-Digital para conversión de los mismos. Quizá no cuente ni con los canales que son necesarios para este tipo de señales, y lo más importante de todo que, para energizar la computadora, se necesita una fuente de voltaje portátil. No siempre el proceso a estudiar se va a ubicar en donde haya alimentación de voltaje doméstica, por lo general siempre se encontrará en lugares donde se carezca del mismo. Además, desde un punto de vista práctico y económico, sería un desperdicio que una computadora se utilizara solo para este fin, sin contar con los accesorios extra que son necesarios para este tipo de situaciones como, por ejemplo, un acumulador y la adición de una interfaz apropiada.

Por lo anteriormente citado, se cuenta con los elementos que debe tener un Sistema de Adquisición de Datos para realizar las funciones propuestas. En cuanto a las ventajas que presenta hacer la toma de datos de forma automática, se tienen las siguientes:

- Límite automático de toma de muestras en tiempo y cantidad, o simplemente dando el número de muestras requerido mediante un comando.

- No causar alteración al sistema estudiado y tampoco a las variables involucradas.

- La información obtenida por el Sistema, dependiendo del número de datos y su ubicación, puede ser almacenada en disco flexible o duro, de una computadora (programa de interfaz con RS-232) o en cinta (cassette).

- Una vez almacenada, la información podrá listarse para su estudio más preciso y detallado, teniendo la posibilidad de que los datos puedan graficarse para análisis rápido y comparación de los cambios presentados desde la última toma de datos.

El sistema propuesto presenta en especial la ventaja de poder utilizarse para diferentes funciones cuando no se encuentre operando como adquisidor de datos (interfaz entre sistemas, indicador de tiempo real, etc.).

## 1.2. CARACTERISTICAS DEL SISTEMA DE ADQUISICION DE DATOS

Ya se analizó el porqué de utilizar un sistema automático de Adquisición de Datos cuando se desee estudiar un fenómeno que, por su complejidad, no puede efectuarse en forma manual.

Para poder cubrir todos los requisitos necesarios para estudiar este tipo de sistemas, es necesario contar con las siguientes características:

- Una interfaz de entrada que pueda acondicionar las señales provenientes del sistema a estudiar (señales analógicas) de forma que se puedan introducir, para su lectura y procesamiento, en el Sistema Digital.

- Un teclado, por medio del cual se pueda interactuar con el sistema e indicarle, dentro de las distintas funciones con las cuales cuenta, el comando que se desee ejecutar.

- Un controlador, basado en un microprocesador, que tenga a su cargo la comunicación con los distintos periféricos del sistema, así como el manejo del procesamiento, la toma y almacenamiento de datos. Este controlador deberá generar las señales correspondientes para los circuitos que se enlazan tanto con el dispositivo de registro, como con los que se encargan de la comunicación con la interfaz de almacenamiento de información.

- Un bloque de memoria en el Sistema Adquisidor para las muestras,

con el fin de no utilizar el almacenamiento externo de información, a menos que la cantidad de datos lo amerite. De esta forma se le da un grado de autonomía al Sistema Adquisidor en la operación de muestreo.

Dentro de la arquitectura se cuenta con circuitos integrados que realizan actividades específicas como son la transmisión y recepción en serie de las señales necesarias para el funcionamiento de la interfaz, la comunicación con ésta y con el generador de tiempos para el microprocesador, que es el que permite dar tiempos de espera y generar una inicialización al sistema sin necesidad de apagarlo, o sea, un "Reset".

Además de la memoria de almacenamiento momentáneo que se debe tener para el almacenamiento de los datos, el sistema debe contar con una memoria de tipo permanente, EPROM, que contenga el programa que el microprocesador debe ejecutar.

Aunado a esto, se debe disponer de un programa en el sistema para proporcionar al usuario la información acerca del número de canales usados, número de muestras a tomar, tiempo de muestreo, el tiempo en el cual serán tomados, si se almacenará en cinta o se manda la información a la computadora, así como si los datos obtenidos del Sistema de Adquisición serán listados o graficados.

Una vez que el programa del sistema cuente con los parámetros necesarios para el inicio de la toma de muestras, los datos son enviados primero al Sistema de Adquisición y luego al dispositivo de

almacenamiento en forma serial utilizando la norma de comunicación RS-232.

El identificador que contenga los datos de la información obtenida, estará formado por el número del canal al cual correspondan las muestras obtenidas y por los datos del tiempo, es decir, tendrá además los siguientes parámetros: mes, día, hora y minutos del registro de información.

Para lograr que el sistema realice todas funciones, se tiene la necesidad de incluir un reloj de tiempo real que sea apto para realizar este tipo de funciones.

Este mismo reloj servirá para llevar los tiempos de toma de muestras y podrá ser accedido directamente por el programa que se encarga de interactuar con el usuario y con el Sistema Adquisidor.

## CAPITULO II

### CARACTERISTICAS DEL SISTEMA DE ADQUISICION DE DATOS



## CAPITULO II

### CARACTERISTICAS DEL SISTEMA DE ADQUISICION DE DATOS

A continuación se describe la forma por medio de la cual se adquiere y acondiciona la información proveniente de la variable que se esté sensando.

#### 2.1. AMPLIFICADOR DIFERENCIAL

El amplificador más útil para medición, instrumentación o control es el Amplificador de Instrumentación. Este amplificador se diseña con varios amplificadores operacionales y resistencias de precisión, lo que permite una gran estabilidad. Una primera aproximación a este tipo de amplificador es el Amplificador Diferencial, el cual, con algunas adiciones, conduce al denominado Amplificador de Instrumentación. Por ello se tratará primero el Amplificador Diferencial, para después entender, de una forma más lógica y clara, el de Instrumentación.

##### 2.1.1. Amplificador Diferencial Básico

Este tipo de amplificador puede amplificar señales débiles superpuestas a señales mucho más intensas. Está constituido por cuatro resistencias de precisión (1%) y un amplificador operacional, como se muestra en la FIG. 2.1.. Tiene 2 terminales de entrada, etiquetadas como

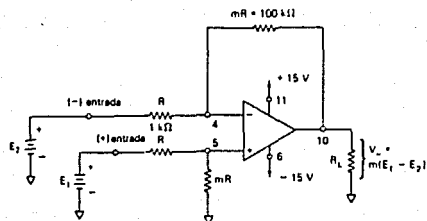


FIG. 2.1. Amplificador Diferencial Básico

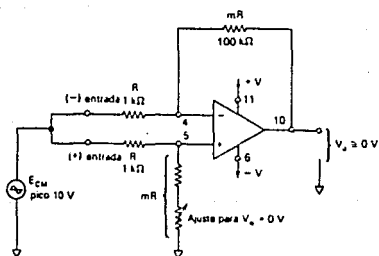


FIG. 2.2. Amplificador Diferencial con Entrada en Modo Común

entrada (-) y entrada (+). Si se reemplaza  $E_1$  por un corto circuito,  $E_2$  ve un amplificador inversor con una ganancia de  $-m$  y el voltaje de salida es  $-mE_2$ . Ahora, si se cortocircuita  $E_2$ ,  $E_1$  se divide entre  $R$  y  $mR$  para aplicar un voltaje de  $E_1m/(1 + m)$  a la entrada (+) del amplificador operacional. Este divisor de voltaje ve un amplificador no inversor con una ganancia de  $(m + 1)$ . El voltaje de salida debido a  $E_1$  es el voltaje  $E_1m$  dividido  $(m + 1)$  veces por la ganancia del amplificador no inversor, lo cual finalmente da  $mE_1$ . Cuando tanto  $E_1$  como  $E_2$  están presentes en las entradas (+) y (-), respectivamente,  $V_o$  es  $mE_1 - mE_2$ , o

$$V_o = mE_1 - mE_2 = m(E_1 - E_2) \quad (2.1)$$

En la ecuación (2.1) se muestra que el voltaje de salida del amplificador diferencial,  $V_o$ , es proporcional a la diferencia de los voltajes aplicados a las entradas (+) y (-). El multiplicador  $m$  se denomina "ganancia diferencial" y se establece por medio de las razones de las resistencias.

La salida del amplificador diferencial debe ser cero (0) cuando  $E_1 = E_2$ . El modo más simple de aplicar voltajes iguales es alambrear ambas entradas juntas y conectarlas al voltaje fuente, FIG. 2.2.. Para dicha conexión, el voltaje de entrada se denomina "voltaje de entrada en modo común", Ecm. Ahora  $V_o$  será 0 si las razones entre las resistencias son iguales ( $mR$  a  $R$  para la ganancia del amplificador inversor es igual a  $mR$  a  $R$  del circuito divisor de voltaje). Prácticamente, las razones de resistencias se igualan por la conexión de un potenciómetro en serie con una resistencia, como se muestra en la FIG. 2.2.. El potenciómetro se

ajusta hasta que  $V_o$  se reduce a un valor despreciable. Esto hace que la ganancia de voltaje en modo común,  $V_o/E_{cm}$ , se aproxime a 0. Esta es la característica de un Amplificador Diferencial, que permite que una señal débil se capte, extrayéndola de una señal más intensa.

### 2.1.2. Mejoras al Amplificador Diferencial Básico

Hay dos desventajas del amplificador diferencial básico: tiene baja impedancia de entrada y el cambio de ganancia es difícil, debido a que las razones de las resistencias deben igualarse estrechamente. La primera desventaja se elimina al aislar las entradas con seguidores de voltaje. Esto se lleva a cabo con dos amplificadores operacionales conectados como seguidores de voltaje en la FIG. 2.3. El voltaje diferencial de salida  $V_o$  se desarrolla a través de la resistencia de carga  $R_L$ .

Se observa que la salida del amplificador en la FIG. 2.1. es una salida referida a tierra, y  $V_o$  se mide desde la terminal de salida del amplificador operacional a tierra.

El amplificador diferencial reforzado en la FIG. 2.3. tiene una salida diferencial. Es decir, ningún extremo de  $R_L$  está conectado a tierra, y  $V_o$  se mide sólo a través de  $R_L$ .

La segunda desventaja del amplificador diferencial básico es la falta de ganancia ajustable. Este problema se elimina agregando tres resistencias más al amplificador reforzado. La entrada diferencial reforzada agregada al amplificador diferencial de salida con ganancia

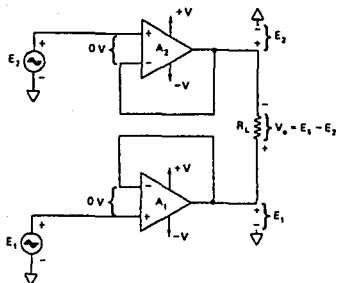


FIG. 2.3. Amplificador Diferencial Reforzado

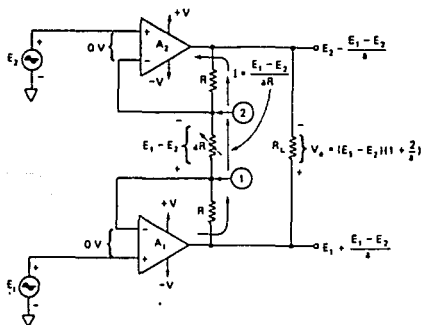


FIG. 2.4. Amplificador Diferencial Reforzado con Ganancia Ajustable

ajustable se muestra en la FIG. 2.4..

Ya que el voltaje diferencial de entrada de cada amplificador operacional es 0 V, los voltajes en los puntos 1 y 2 (con respecto a tierra) son iguales a  $E_1$  y  $E_2$  respectivamente. Por tanto, el voltaje a través de la resistencia  $aR$  es  $E_1 - E_2$ . La resistencia  $aR$  es un potenciómetro que se utiliza para ajustar la ganancia. La corriente a través de  $aR$  es:

$$I = (E_1 - E_2)/aR \quad (2.2)$$

cuando  $E_1$  es más grande que  $E_2$ , la dirección de la corriente,  $I$ , es como se muestra en la FIG. 2.4.  $I$  fluye a través de ambas resistencias designadas por  $R$ , y el voltaje a través de las tres resistencias establece el valor de  $V_o$ . En forma de ecuación :

$$V_o = (E_1 - E_2) (1 + 2/a) \quad (2.3)$$

donde:

$$a = aR/R$$

Concluyendo, se puede decir que para cambiar la ganancia del amplificador, tiene que ajustarse ahora una sola resistencia,  $aR$ . Sin embargo, el amplificador diferencial reforzado tiene una desventaja: puede manejar sólo cargas flotantes. Las cargas flotantes son cargas que no tienen ninguna terminal conectada a tierra. Para manejar cargas a tierra debe agregarse un circuito que convierta un voltaje diferencial de entrada en un voltaje de salida de extremo único. Dicho circuito es el

amplificador diferencial básico. La configuración resultante de circuito se denomina Amplificador de Instrumentación.

## 2.2. AMPLIFICADOR DE INSTRUMENTACION

El Amplificador de Instrumentación es uno de los amplificadores más útiles, preciso y versátil disponible en la actualidad. Para simplificar el análisis del circuito, se observa que el amplificador de instrumentación en realidad se hace conectando un amplificador reforzado (FIG. 2.4.) a un amplificador diferencial básico (FIG. 2.1.). El amplificador operacional A3 y sus cuatro resistencias iguales, R, forman un amplificador diferencial con una ganancia de 1. Sólo las resistencias de A3 tienen que igualarse. La resistencia marcada con prima, R', puede hacerse variable para balancear, eliminando cualquier voltaje en modo común, como se muestra en la FIG. 2.2.. Sólo una resistencia, aR, se usa para establecer la ganancia de acuerdo con la ecuación (2.3) :

$$V_o / ( E_1 - E_2 ) = 1 + 2/a \quad (2.4)$$

donde:

$$a = aR/R$$

El se aplica a la entrada (+) y E2 a la entrada (-). Vo es proporcional a la diferencia entre los voltajes de entrada. Las características del Amplificador de Instrumentación se puede resumir como sigue :

1. La ganancia de voltaje, desde la entrada diferencial ( $E_1 - E_2$ ) a la salida de extremo único, se establece por un resistor.

2. La resistencia de entrada de ambas entradas es muy alta y no cambia conforme se varia la ganancia.

3.  $V_o$  no depende del voltaje común a ambos  $E_1$  y  $E_2$  (voltaje modo común), sólo de su diferencia.

### 2.2.1. Voltaje de Salida Referenciado

En algunas aplicaciones es deseable desviar el voltaje de salida a un nivel de referencia diferente de 0. Esto puede realizarse con bastante facilidad mediante la adición de un voltaje de referencia en serie con una resistencia al amplificador básico. Supóngase que  $E_1$  y  $E_2$  valen 0 V en la FIG. 2.5.. Las salidas de  $A_1$  y  $A_2$  serán iguales a 0 V. Así, puede mostrarse en la entrada del amplificador diferencial como 0 V en la FIG. 2.6..

Un voltaje de referencia  $V_{REF}$  se conecta en serie con la resistencia  $R$  a la terminal + de  $A_3$  y por la presencia de  $R$  de la terminal de entrada  $V_{REF}$  se divide entre 2. El amplificador no inversor da una ganancia de 2, de modo que  $V_o$  iguala a  $V_{REF}$ . Ahora  $V_o$  puede establecerse en cualquier valor de referencia deseado mediante el ajuste de  $V_{REF}$ . En la práctica  $V_{REF}$  es la salida de un circuito seguidor de voltaje.



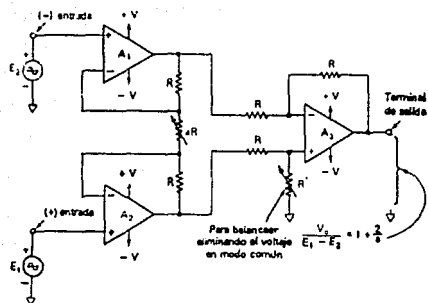


FIG. 2.5. Amplificador de Instrumentación

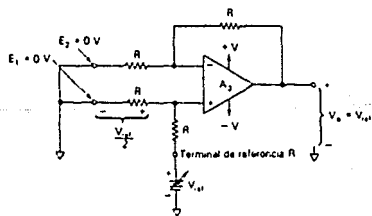


FIG. 2.6. Amplificador Diferencial con Salida Referenciada

## 2.2.2. Detección y Medición con el Amplificador de Instrumentación

La versatilidad y rendimiento de este amplificador puede mejorarse abriendo el circuito de retroalimentación negativa alrededor del amplificador operacional  $A_1$  y sacando 3 terminales de él como se muestra en la FIG. 2.7.. Estas terminales son la terminal de salida O, la terminal sensora S, y la terminal de referencia R. Al usar cables largos o un transistor reforzador de corriente entre el amplificador y la carga, se presentarán caídas de voltaje a lo largo de los cables de conexión.

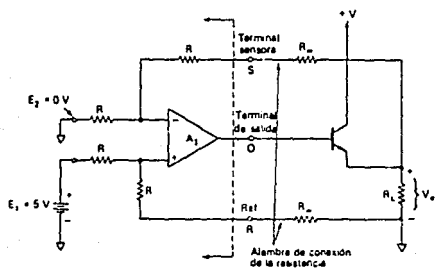


FIG. 2.7. Configuración Práctica para Detección y Medición  
Usando el Amplificador de Instrumentación

Para eliminar estas caídas de voltaje, la terminal sensora y la terminal de referencia se alambran directamente a la carga. Esto permite que la resistencia del cable se agregue por igual a las resistencias en serie con las terminales sensora y de referencia para minimizar cualquier desigualdad. Aún más importante, al detectar el voltaje en las terminales de carga y no en la terminal de salida del amplificador, la retroalimentación actúa para mantener constante el voltaje de carga. Si sólo se utiliza el amplificador diferencial básico, el voltaje de salida se encuentra mediante la ecuación (2.1) con  $m = 1$ . Si se usa el amplificador de instrumentación, el voltaje de salida se determina por medio de la ecuación (2.3).

## CAPITULO III

### DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS

El sistema se diseñó con base en componentes electrónicos integrados. Entre los circuitos utilizados hay algunos más complejos que otros, lo cual no significa que sean más importantes pues cada componente tiene una función específica dentro del sistema.

#### 3.1. DESCRIPCIÓN GENERAL

En la FIG 3.1. se muestra un Diagrama de Bloques del Sistema. Como se puede observar en el diagrama de bloques, se incluye una microcomputadora como una opción para el almacenamiento de la información, pues debido a la variedad de situaciones en las cuales este sistema puede ser utilizado, es probable que la capacidad disponible de almacenamiento en memoria no sea suficiente para la aplicación que se le esté dando.

En el capítulo 5, en la sección en la cual se incluye la descripción de los programas, se presenta el programa que permite realizar el enlace entre la computadora y el Sistema de Adquisición.

## CAPITULO III

### DISEÑO DEL SISTEMA DE ADQUISICIÓN DE DATOS

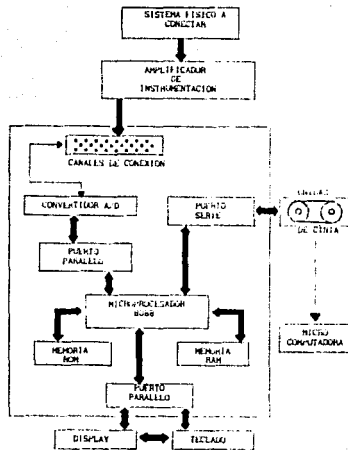


FIG. 3.1. Diagrama de Bloques del Sistema.

### 3.2. ARQUITECTURA DEL MICROPROCESADOR 8088

A continuación se verá como se encuentra constituido este microprocesador de 16 bits, que es uno de los más importantes en la historia de los sistemas digitales, especialmente en el de las computadoras personales. En la FIG. 3.2. se muestra la arquitectura del microprocesador 8088 de INTEL.

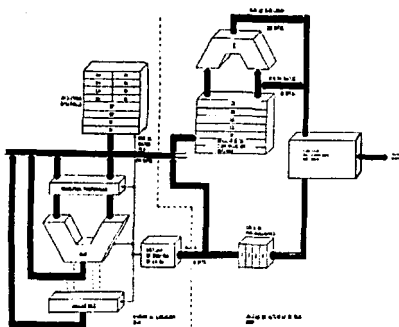


FIG. 3.2. Arquitectura del Microprocesador 8088.

### 3.2.1. Capacidad de direccionamiento

El 8088 tiene un bus de direcciones de 20 bits lo que le da capacidad para direccionar 1 Mbyte de memoria. En términos matemáticos se puede expresar como :

$$2^{**} 20 = 1\ 048\ 576 = 1\ \text{Mbyte} \quad (3.1)$$

Sin embargo, el registro de direccionamiento del 8088 tiene únicamente 16 bits, o sea que puede direccionar 64 kbytes, entonces el procesador usa un método llamado Segmentación de Memoria para permitir el direccionamiento a todo el megabyte de memoria.

El 8088 tiene otra memoria separada, llamada espacio de E/S, que puede considerarse como una memoria extra para direccionar, en la cual se encuentran los dispositivos de E/S. En el 8088, los dispositivos de E/S pueden conectarse a la memoria principal o al espacio de E/S. Así mismo, la memoria puede conectarse a la memoria principal o al espacio de E/S. El espacio de E/S del 8088 usa un direccionamiento de 16 bits (permite direccionar 64 kbytes).

### 3.2.2. Señales de reloj

El 8088 requiere una señal única de reloj. Estos procesadores no generan su propia señal de reloj, sino que dependen del generador de reloj 8284 que usa un cristal oscilador para determinar la frecuencia de la señal. Intercambiando este cristal, se pueden seleccionar diferentes



velocidades de operación. Intel cuenta con 2 versiones para el 8088 : de 5 MHz y 8 Mhz. Estas velocidades representan las velocidades más altas, recomendadas para estos circuitos. No se recomienda bajar de 2 MHz. Una velocidad más baja o más alta puede ser correcta o puede causar error. Para un rendimiento óptimo, el 8088 requiere una señal de reloj que se mantenga en un nivel alto una tercera parte del tiempo total del ciclo. Esto significa que el reloj está activo una tercera parte del tiempo y desactivado las dos terceras partes del tiempo.

### 3.2.3. Requisitos de Potencia

El 8088 requiere una alimentación de 5 V. Cualquier dispositivo de Intel de la serie 8000 relacionado con este sistema tiene este requerimiento.

### 3.2.4. Modos de Direccionamiento

Un modo de direccionamiento es un conjunto de reglas que especifican la localización (posición) de un dato usado durante la ejecución de una instrucción. En el modo más sencillo, un dato se localiza en un registro determinado mientras que en el modo más complejo, se suma el contenido de los 2 registros en una cantidad de 8 ó 16 bits, que se encuentra en el programa. El resultado indica la dirección del dato.

El 8088 tiene 25 modos diferentes de direccionamiento o reglas para localizar un operando de una instrucción. En la FIG 3.3. se ilustra la forma en que se realiza el direccionamiento con el 8088.

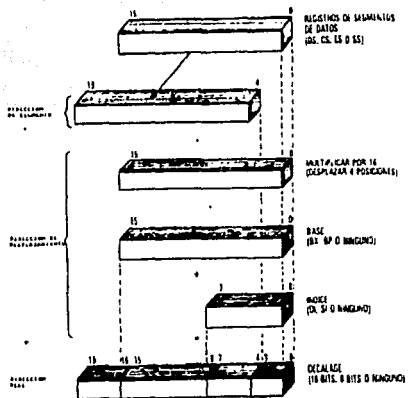


FIG. 3.3. Modos de Direccionamiento en el 8088.

Se tienen varios casos especiales de los casos tipo, por ejemplo, referencia a registros y referencia a memoria. En el primer caso, el operando está localizado en un registro específico. Sin embargo, deben sumarse 4 cantidades para obtener la dirección de un operando en memoria. Dichas cantidades son: 1) Dirección de segmento, 2) Dirección base, 3) Una cantidad índice y 4) Un desplazamiento.

La dirección del segmento se almacena en el registro de segmentación (DS, ES, SS, CS). La base se almacena en el registro base (BX o BP). El índice se almacena en el registro índice (SI o DI). Cualquiera de estas cantidades, las dos, o ninguna, pueden utilizarse para calcular la dirección real. El programador puede usar tanto la base como el índice de diferentes formas para gestionar ciertas cosas, tales como matrices de dos dimensiones o estructuras internas a otra estructura, esquemas que se utilizan en las prácticas modernas de programación.

La base y el índice son variables o dinámicas, ya que están almacenadas en registros de la CPU de propósito general. Además del segmento, base e índice, se usan unos desplazamientos de 16 bits, 8 bits o 0 bits (Inexistente). Este desplazamiento es una cantidad estática. Se fija durante el tiempo de ensamblaje (paso de código fuente a código máquina) y no puede cambiarse durante la ejecución del programa. El desplazamiento se utiliza para cosas como compilar datos, organizar la memoria y reubicarla más rápida y fácilmente.

A continuación se muestra la tabla con los modos de direccionamiento con que cuenta el 8088.

mm	aaa	Parte de desplazamiento de la dirección
00	000	(BX) + (SI)
00	001	(BX) + (DI)
00	010	(BP) + (SI)
00	011	(BP) + (DI)
00	100	(SI)
00	101	(DI)
00	110	Dirección directa
00	111	(BX)
01	000	(BX) + (SI) + número 8 bits
01	001	(BX) + (DI) + número 8 bits
01	010	(BP) + (SI) + número 8 bits
01	011	(BP) + (DI) + número 8 bits
01	100	(SI) + número 8 bits
01	101	(DI) + número 8 bits
01	110	(BP) + número 8 bits
01	111	(BX) + número 8 bits
10	000	(BX) + (SI) + número 16 bits
10	001	(BX) + (DI) + número 16 bits
10	010	(BP) + (SI) + número 16 bits
10	011	(BP) + (DI) + número 16 bits
10	100	(SI) + número 16 bits
10	101	(DI) + número 16 bits
10	110	(BP) + número 16 bits
10	111	(BX) + número 16 bits
11	000	registro AX (palabra) o AL (byte)
11	001	registro CX (palabra) o CL (byte)
11	010	registro DX (palabra) o DL (byte)
11	011	registro BX (palabra) o BL (byte)
11	100	registro SP (palabra) o AH (byte)
11	101	registro BP (palabra) o CH (byte)
11	110	registro SI (palabra) o DH (byte)
11	111	registro DI (palabra) o BH (byte)

Donde mm significa los 2 primeros bits del byte de direccionamiento y aaa los 3 últimos bits de dicho byte.

### 3.2.5. Encapsulado

El 8088 viene en un encapsulado de 40 terminales. Aunque este no es impedimento para generar más señales que algunos procesadores de más terminales, que cuentan con 64 terminales por ejemplo.

### 3.2.6. Multiprocesamiento y Procesamiento Paralelo

El 8088 cuenta con características especiales que permiten coordinar sus actividades con otros procesadores en contextos de multiproceso y procesamiento en paralelo. El procesamiento en paralelo es un sistema en el cual dos o más procesadores trabajan en tandem en la misma porción de un programa. Multiproceso es un sistema en el cual dos o más procesadores trabajan en diferentes programas, pero comparten la misma memoria.

### 3.2.7. Arquitectura Pipeline

Intel diseñó el 8088 para realizar al mismo tiempo las principales funciones internas de transferencia de datos y búsqueda de instrucciones. Para conseguirlo el 8088 cuenta de 2 procesadores interconectados en la misma pastilla de silicio. Una unidad está encargada de buscar instrucciones y la otra de ejecutarlas. La unidad encargada de buscar instrucciones utiliza un método llamado pipeline que es una estructura tubular o por cola que permite almacenar nuevas instrucciones hasta el momento en que se necesiten.

Al procesador principal se le llama unidad de ejecución (EU:

Execution Unit). Está encargado de codificar y ejecutar todas las instrucciones. Al otro procesador se le llama la Unidad de Interfaz de Bus (BIU: Bus Interfase Unit). Cuando la BIU localiza en memoria un byte de código máquina, lo coloca en una línea de espera especial llamada cola de instrucciones. En el 8088 la cola de instrucciones tiene sólo 4 bytes de longitud y el código máquina se guarda de byte en byte. En la FIG 3.2. se muestra este tipo de arquitectura.

### 3.2.8. Conjunto de Registros

El 8088 contiene 14 registros de 16 bits. Algunos pertenecen a la EU y otros a la BIU. En la FIG 3.4. pueden verse estos registros.

Se tienen los siguientes registros:

- Cuatro registros generales de 16 bits (AX, BX, CX, DX), que pueden subdividirse, y direccionarse separadamente, en 8 registros de 8 bits (AH, AL, BH, BL, CH, CL, DH y DL). En este caso la X representa extendido (16 Bits), la H alto y la L bajo. A significa acumulador, B base, C contador y D datos.

- Cuatro registros apuntadores y de índice (SP, BP, SI y DI), los cuales no pueden subdividirse. SP es el apuntador de pilas, BP es el apuntador base y SI y DI indican a los registros de índice fuente y de datos respectivamente.

- Un registro de banderas de 16 bits, que contiene varios bits de

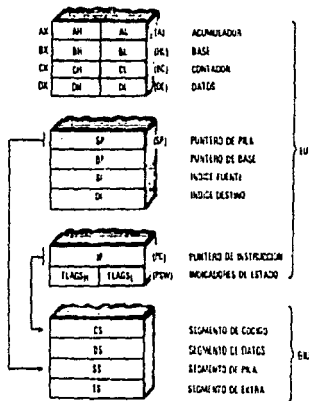


FIG. 3.4. Conjunto de Registros del 8088.

estado para el procesador. Estos incluyen: bandera de cero (ZF), de signo (SF), de paridad (PF), de acarreo (CR), auxiliar (AF), de dirección (DF), de interrupción (IF), de sobreflujo (OF) y de desvío (TF).

La BIU tiene además los siguientes registros:

- Cuatro registros segmento (CS, DS, SS y ES). Sus códigos representan a los registros segmento de código, datos, pila y extra respectivamente.

- Un apuntador de instrucciones.

Cada registro tiene su nombre propio. Por ejemplo, para sumar datos inmediatos a la mayoría de registros se requieren 2 bytes para instrucciones más 1 ó 2 bytes para almacenar datos. Sin embargo, sumar un dato inmediato al acumulador (AX) requiere únicamente 1 byte, más los necesarios para almacenar datos. Por ello es más eficiente usar AX para cálculos en los que intervengan constantes. Las instrucciones para multiplicar y dividir usan el registro acumulador (AX) y el de datos (DX). El registro AX es el acumulador de 16 bits. Usándolo, a veces se provoca que el ensamblador produzca un lenguaje de máquina codificado en muy pocos bytes. El registro CX se usa a menudo para almacenar datos, iteración de movimientos de cadenas, desplazamientos y rotaciones. El registro DX se utiliza para almacenar datos de 16 bits. Puede tomarse como una extensión del registro AX para multiplicaciones y divisiones de datos 16 bits. Los registros SI y DI del 8088 son los encargados de realizar el direccionamiento indirecto. El registro BX (base de propósito



general) se utiliza como registro base para los direccionamientos.

Los registros de propósito específico como los de índice fuente(SI), índice de datos (DI) y el apuntador base (BP) se utilizan como parte de los modos de direccionamiento.

Los registros apuntador de instrucciones (IP) y apuntador de pila (SP) se encargan del control del flujo del programa.

### 3.2.9. Señales y terminales

El 8088 puede conectarse en un circuito de dos formas distintas: en modo máximo y en modo mínimo. Debe estar en modo máximo si quiere trabajar en conjunto con el Procesador de Datos Numéricos 8087 y el Procesador de E/S 8089. En el modo máximo el 8088 se debe conectar a un dispositivo como el Controlador de Bus 8288 para generar el conjunto completo de señales de control de bus. El modo mínimo permite trabajar de una forma más autónoma pues no requiere de las señales necesarias para el arbitraje del bus. La FIG. 3.5. muestra un esquema de ambos modos en los cuales las señales se pueden agrupar de la siguiente manera:

Alimentación  
Reloj  
Control y Estado  
Direcciones  
Datos

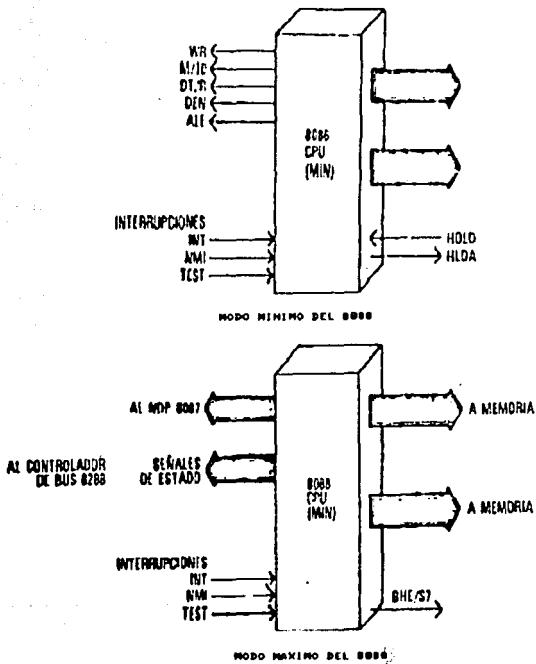


FIG. 3.5. Modos Mínimo y Máximo del 8088.

Hay 3 terminales para la alimentación: tierra (GND) en las terminales 1 y 20, y una tensión de entrada de 5 V. en la terminal 40.

Hay una señal de reloj en la terminal 19.

En el 8088 hay 20 bits de dirección. Los 4 bits más significativos de la dirección comparten terminales con algunas de las señales de estado. Sólo los 8 bits menos significativos comparten terminales con los datos. En ciertos instantes, tales terminales conducen parte de una dirección, y en otros llevan información sobre el estado y los datos. El Latch (memoria) 8282 está diseñado para seleccionar la información sobre la dirección de dichas terminales en el instante preciso, e ignorar el estado y los datos.

Hay varios grupos de control y señales de estado.

La terminal MN/MX controla si el procesador está en modo mínimo o máximo, conectándolo a tierra o a una tensión de 5 V.

Del S0 al S7 son señales de estado en las terminales 26, 27, 28, 38, 37, 36, 35 y 34 respectivamente.

En ciertos momentos aparecen otras señales distintas en las mismas terminales. Teniendo en cuenta el estado pueden decirse cosas tales como el tipo de acceso al bus (lectura o escritura, memoria o E/S), el registro de segmento en uso y el estado del sistema de interrupciones. S0, S1 y S2 son sólo accesibles en modo máximo, en cuyo caso se

introducen los integrados controladores de bus 8288. Este controlador de bus genera, a partir de éstas, otras señales de control importantes. En el modo mínimo el 8088 genera alguna de las señales del 8288 directamente.

RD es una señal de estado generada por el procesador sobre la terminal 32. Indica cuando el procesador lee datos, sea de memoria o de E/S.

La señal READY (terminal 22) es una entrada para los dispositivos externos (memoria o controladores E/S). Esta señal pasa a través del generador de pulsos 8284 para sincronizarse con la señal de reloj.

La señal de RESET (terminal 21) también pasa por el generador de pulsos 8284 para sincronizarse con la señal de reloj. RESET se utiliza para inicializar el procesador, lo mismo que si se hubiese desconectado y vuelto a encender. Resulta bastante más sencillo reinicializar el sistema con la señal RESET que desactivar y volver a activar cada componente del circuito. Al dar RESET, se borra la cola de instrucciones y ciertos registros (los indicadores: apuntador de instrucción, segmento de datos, de pila y registros de segmentación adicionales). Tras un RESET, el segmento de código se pone al valor FFFFH.

Los terminales NMI (Non-Maskable Interrupt: Interrupción No Enmascarada - terminal 17) e INTR (INTeRrupt Request : Petición de Interrupción - terminal 18) son parte del sistema de interrupciones del 8088. Un pulso en la línea NMI provoca una interrupción especial llamada

Interrupción tipo 2. Una señal en la línea INTR causará una respuesta de interrupción de tipo general.

En el modo mínimo se tienen las siguientes señales de salida de control y de estado: SSO (EstadoSO - terminal 34), HLDA (terminal 30), WR (Control de escritura- terminal 29), M/IO (Control de Memoria o de E/S- terminal 28), DT/R (Recepción/Transmisión de datos - terminal 27), DEN (Datos accesibles - terminal 26), ALE (Latch de Direcciones Activo - terminal 25) e INTA (Reconocimiento de Interrupciones - terminal 24). Una de las señales es de entrada, HOLD (terminal 31). La señal M/IO del 8088 se invierte para hacerla compatible con las señales del 8085. La señal BHE (terminal 34) se usa en el direccionamiento de 16 bits.

En el modo máximo, las siguientes señales son de salida : S2, S1, S0 (Estado - terminales 28-26); Q51, Q50 (Estado de la Cola - terminales 25 y 24); RQ/GT1, RQ/GT0 (Petición/Concesión - terminales 31 y 30), y LOCK (terminal 29). Realmente, las líneas de petición/concesión (RQ/GT) también pueden aceptar líneas de entrada.

La señal TEST (terminal 23) se utiliza para enlazar el 8088 con un procesador, tal como el procesador numérico 8087, sincronizando el procesador principal con el o los otros procesadores.

Una estructura de interrupción es una forma de que el procesador provea un servicio rápido y uniforme para la E/S, correcciones y ciertos tipos de error.

Varios dispositivos pueden generar interrupciones por hardware, por ejemplo, el NDP 8087, el IOP 8089, el 8259, así como el propio 8088. La interrupción (INT) del 8088 puede generar interrupciones software. La estructura de interrupciones del 8088 utiliza una tabla de 256 posiciones de 4 bytes cada una, la cual está en el inicio absoluto del mapa de memoria. Cada una de estas posiciones de la tabla de interrupciones puede cargarse con un apuntador a diferentes rutinas de la memoria principal. Estos apuntadores contienen el nuevo contenido del segmento de código (2 bytes) para la rutina que puede estar localizada en cualquier parte de la memoria. A cada uno de estos apuntadores de 4 bytes se le asigna un número del 0 al 255, según su posición en la memoria. A este número se le llama tipo, al tipo de interrupción 0 se le asigna la posición de memoria 0, al tipo 1 se le asigna la posición 4, y así hasta la posición 1020.

Cada tipo de interrupción puede ser llamado tanto por hardware como por software. Esto hace posible probar vía software las interrupciones hardware.

### 3.2.10. Estructura de Segmentación de Memoria

El 8088 usa un esquema ingenioso llamado segmentación para acceder correctamente un Mbyte completo de memoria, con referencias de direcciones de sólo 16 bits.

Cualquier dirección en el 8088 tiene 2 partes, cada una de las cuales es una cantidad de 16 bits. Una parte es el desplazamiento y la otra la dirección de segmento. El desplazamiento de 16 bits se compone de

varias partes: un escalamiento (un número fijo), una base (almacenada en el registro base) y un índice (almacenado en el registro índice). La dirección del segmento se almacena en uno de los cuatro registros de segmento (CS, DS, ES, y SS). El procesador usa estas dos cantidades de 16 bits para calcular la dirección real (FIG. 3.6.) de 20 bits, según la siguiente fórmula:

$$\text{Dirección real} = 16 * (\text{dirección del segmento}) + \text{desplazamiento}$$

Dado que 16 en decimal es 10 en hexadecimal, multiplicar por 16 decimal es lo mismo que correr una posición a la izquierda un número hexadecimal.

Los registros de segmentación reciben los nombres siguientes: de código, datos, pila y extra (CS, DS, SS, y ES, respectivamente). Las instrucciones se gestionan usando el segmento código, las operaciones con pilas usan el segmento de pila, y para los datos se utilizan el segmento de datos y el extra.

Por ejemplo, supongamos que el procesador ejecuta un programa y el apuntador de instrucciones (IP) contiene 234h y el registro de código contiene 800h. La siguiente fórmula indica que el siguiente byte de instrucción se encuentra en la dirección de memoria

$$8234 = 800 * 10 + 234 \text{ (hexadecimal)}$$

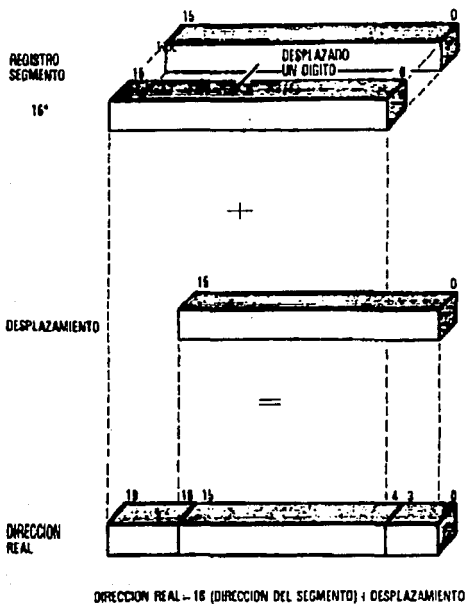


FIG. 3.6. Cálculo de la Dirección en el 8088.



Esta dirección de segmento únicamente indica dónde se inicia un segmento. Pero, lo que no hace de ninguna forma es dividir el segmento en párrafos.

Hay un byte especial de prefijo que permite ignorar algunas de estas asignaciones, pero no todas.

La combinación del índice destino (DI) y el segmento extra (ES) se usa siempre para calcular la dirección del destino en cualquier operación con cadenas. El byte prefijo puede usarse, sin embargo, para obligar a utilizar alguno de los 4 registros de segmento en el cálculo de la dirección puente de una operación de cadenas. El valor por defecto (o sea, el valor sin byte prefijo) es el registro de datos.

Los 24 modos de referenciar la dirección (usados para el acceso de datos) pueden aceptar un byte prefijado e ignorar sus asignaciones por defecto al segmento, usando cualquier registro segmento. La asignación por defecto es o el segmento de datos (DS) o el de pila (SS) y, de hecho se usa siempre el segmento de datos a menos que el modo de direccionamiento use el apuntador de base (BP), en cuyo caso se usa el registro de pila. Intel aconseja utilizar el apuntador base para acceder datos en la pila del sistema y normalmente con llamadas a subrutina.

El uso de estos diferentes segmentos significa que hay áreas de trabajo separadas para el programa, la pila y los datos. Cada área de trabajo tiene un tamaño máximo de 64 kbytes y mínimo de 0. Dado que hay 4 registros de segmento, uno de programa (CS), uno de pila (SS) y dos de

datos - segmento de datos (DS) y segmento extra (ES)- el área de trabajo puede llegar hasta  $4 * 64 \text{ k} = 256 \text{ k}$  en un momento dado. Se supone que las distintas áreas no se superponen. Sin embargo, aunque no es necesario, es posible y aconsejable colocar los 4 segmentos en la misma área. En este caso programa, pila y datos residirían en la misma área de trabajo de 64k.

El programador puede determinar la posición de estos segmentos, cargando el registro de segmento de 16 bits apropiado con la dirección de segmento apropiada. Cambiando el desplazamiento, el programador puede acceder a cualquier punto del segmento.

### 3.2.12. Conjunto de Instrucciones

En general, las instrucciones del procesador se pueden clasificar en los siguientes grupos:

- 1) Transferencia de datos.
- 2) Aritmética entera binaria.
- 3) Operaciones lógicas.
- 4) Desplazamientos y rotaciones.
- 5) Gestión de bits.
- 6) Aritmética codificada en binario.
- 7) Gestión de cadenas.
- 8) Control del programa.
- 9) Control del Sistema.

Para las instrucciones del 8088 se usarán los mnemotécnicos de Microsoft dado que son, probablemente, los más populares. Con estos mnemotécnicos, muchas instrucciones indican (como parte de sus códigos de operación) si los operandos son en bytes o palabras o si la fuente son datos inmediatos. Una B extra en el mnemotécnico de la operación indica modo byte (datos de 8 bits), mientras que su ausencia indica modo de palabra (datos de 16 bits). Una I extra en el mnemotécnico de la operación indica datos inmediatos en la fuente. Cuando la fuente es un dato inmediato de 8 bits, entonces aparecerán la B y la I (en este orden) en el mnemotécnico de la operación. El uso de la B y la I como parte del código de operación no debe confundirse con los modos de direccionamiento y la forma en que éstos afectan al operando.

A continuación se muestran las tablas de instrucciones con las que cuenta el 8088.

### 3.3. PUERTO PARALELO PROGRAMABLE 8255

Cuando es necesario transmitir varios datos a un mismo tiempo y a un mismo sitio, sin importar cual fuere la causa de esto, lo hacemos por medio de un puerto paralelo. Como su nombre lo indica, los datos se mandan o se reciben en forma paralela, es decir, en el mismo instante y agrupados en un número que depende del circuito a utilizar. La ventaja que se tiene respecto a la comunicación serial es que se cuenta con más líneas de transmisión de datos. La desventaja es, por supuesto, el costo extra de las líneas, estos costos se incrementan con la distancia. Este tipo de comunicación se utiliza para grandes distancias si los datos lo requieren.

A diferencia de la comunicación serial, la comunicación en paralelo no ha sido normalizada. La comunicación en paralelo puede producirse colocando los datos dentro del buffer del registro de la interfaz o tomando los datos de este mismo registro para mandarlos hacia afuera, o simplemente conduciéndolos bajo el control de señales de tiempo o protocolo (Handshaking). Normalmente, un caracter ( u otra pieza de la información) es transferida a la vez y no hay problema en determinar el final de un caracter o encontrar el principio de la transmisión. Si solamente se emplean señales de tiempo, la transmisión se puede hacer de forma asíncrona.

Esta interfaz de Entrada/Salida es un dispositivo de propósito general programable que puede utilizarse en la mayoría de los microprocesadores que existen en la actualidad, gracias a que puede

servir de interfaz entre los dispositivos periféricos y el bus del sistema del microprocesador que se este utilizando.

Entre las características con las cuales cuenta esta interfaz se encuentra la capacidad de activar y desactivar bits (Set/Reset) y la de alimentar 1 mA de corriente a 1.5 V. Esto permite, entre otras cosas, el manejo directo de transistores Darlington en aplicaciones tales como impresoras y displays de alto voltaje.

La configuración funcional del 8255 se realiza por la programación del Sistema. Cuenta con 24 terminales de E/S que se pueden programar individualmente en dos grupos de 12 y usarse en tres modos principales de operación (0, 1 y 2). La FIG. 3.7. muestra el diagrama de bloques de esta interfaz, indicando el nombre de cada una de las terminales así como el de las funciones que realizan. A continuación se describen las características de cada uno de los modos de operación.

### 3.3.1. Modos de Operación

#### Modo de Operación 0

Este modo de operación proporciona operaciones simples de Entrada/Salida para cada uno de los 3 puertos. No se requiere de un protocolo previo, los datos simplemente se leen o se escriben en un puerto específico. El 8255 tiene las siguientes características en este modo:

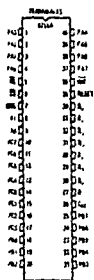


TABLE 3.1: MAPPING OF THE TERMINALS

P0	CS
P1	RD
P2	RD
P3	RD
P4	RD
P5	RD
P6	RD
P7	RD
P8	RD
P9	RD
P10	RD
P11	RD
P12	RD
P13	RD
P14	RD
P15	RD
P16	RD
P17	RD
P18	RD
P19	RD
P20	RD
P21	RD
P22	RD
P23	RD

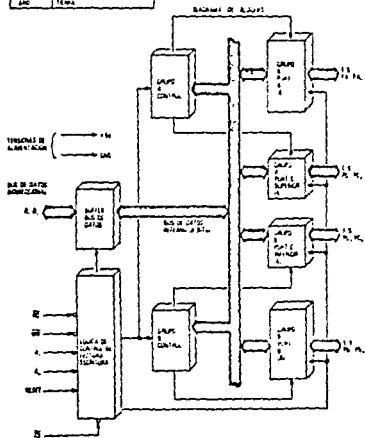


FIG. 3.7. Puerto Paralelo Programable 8255

- Tres puertos de 8 bits
- Cualquier puerto puede ser de entrada o de salida
- Las salidas se almacenan en latches
- Se pueden realizar 16 configuraciones diferentes de E/S
- Las entradas no se almacenan en latches

#### Modo de Operacion 1

Esta configuración funcional proporciona una forma de transferencia de datos de E/S con un puerto utilizando señales de muestreo (Strobe) o de protocolo. En este modo los puertos A y B se definen como puertos de entrada o de salida y el puerto C se utiliza para proporcionar las señales de protocolo. El 8255 programado de este modo cuenta con las siguientes características:

- Dos grupos (A y B)
- Cada grupo contiene un puerto de datos de 8 bits y un puerto de control (Puerto C) de 4 bits.
- Los puertos pueden ser tanto de entrada como de salida y cada una de éstas se almacena en latches.
- El puerto de 4 bits de cada grupo se usa para las señales de control y de estados del puerto de 8 bits

#### Modo de Operacion 2

Este modo permite al puerto A actuar como puerto bidireccional de datos. Se proporcionan señales de protocolo a través del puerto C (5

señales) para mantener el control del flujo de datos del periférico. Estas señales de control son una combinación de las señales de control de entrada y de salida del Modo 1. La generación de solicitudes de interrupción y las funciones Set/Reset de los Flip-Flops INTE también se encuentran disponibles. El 8255 programado en este modo cuenta con las siguientes características:

- El puerto A como puerto bidireccional de datos de 8 bits y el puerto C como control con 5 bits
- Las salidas y entradas se almacenan en latches

### 3.3.2. Conexión en el sistema

A continuación se describirá la forma de conexión del puerto paralelo, así como las señales que habilitan el sistema. En el sistema se cuenta con 3 circuitos 8255, realizando cada uno de ellos una función determinada.

El 8255-3 como se encuentra en el diagrama de circuitos correspondiente al Sistema, funciona como interfaz entre el multiplexor de canales y el convertidor Analógico/Digital, teniendo los puertos la siguiente configuración :

#### Puerto A del 8255-3

Este puerto tiene como función asignar la entrada de estado, por medio de la cual indica el convertidor Analógico/Digital 14433 que ha



finalizado la conversión correspondiente por medio de la señal de fin de conversión EOC (End of Conversion) . Unicamente se conecta una terminal del puerto:

PA0 - EOC

**Puerto B del 8255-3**

La función que realiza este puerto es de las más importantes del sistema. Seis líneas se encargan de configurar la decodificación para los 24 canales de adquisición de datos conectadas a las señales del multiplexor de canales 14051. Tres líneas se encargan de seleccionar cuál de los 3 circuitos 14051 localizados en el sistema es seleccionado. Las siguientes 3 líneas producen la multiplexión de cada uno de los 8 Canales con los cuales cuenta cada 14051. Otra línea se encarga de la habilitación del relevador, el cual conecta a los demas circuitos que tienen la función de la conversión A/D.

En total se utilizan 7 líneas, de las 8 con que cuenta el puerto:

PB0 - SCA1

PB1 - SCA2

PB2 - SCA3

PB3 - HCA1

PB4 - HCA2

PB5 - HCA3

PB6 - FWC

### Puerto C del 8255-3

Junto con el B, este puerto realiza otra de las funciones importantes dentro de la conversión A/D. Sus 8 líneas se encuentran conectadas a las terminales del convertidor 14433.

Las primeras 4 líneas reciben las señales de salida codificadas en BDC referentes al valor analógico convertido a valor digital.

Mientras que las restantes 4 líneas se encargan de seleccionar el dígito a ocupar por parte del convertidor A/D.

Cada uno de los grupos anteriores se configura desde el MSD (Digito Más Significativo) hasta el LSB (Digito Menos Significativo).

PC0 - DQ3

PC1 - DQ2

PC2 - DQ1

PC3 - DQ0

PC4 - DS1

PC5 - DS2

PC6 - DS3

PC7 - DS4

Por su parte, el 8255-1 tiene como función enlazar las señales que comunican el teclado con el sistema de adquisición y el display. Se utilizan únicamente 2 puertos, el puerto A y el puerto B. La

configuración de éstos se muestra a continuación :

**Puerto A del 8255-1**

Los datos provenientes del teclado de configuración matricial son leídos por medio de este puerto. Se utilizan 4 líneas, las cuales se conectan a las teclas realizando la función de renglones. El nombre de las señales es el siguiente:

PAD - RL1

PA1 - RL2

PA2 - RL3

PA3 - RL4

**Puerto B del 8255-1**

Este puerto se encarga de realizar la función de columnas cuando son introducidos los datos para la decodificación de las teclas. Junto con el puerto A, conforman la estructura matricial y se conectan, por medio de Software, a una tabla de identificación, para posteriormente ser enviadas al Display. El nombre de las señales es el siguiente :

PBD - CL1

PB1 - CL2

PB2 - CL3

PB3 - CL4

PB4 - CL5

PB5 - CL6

PB6 - CL7

PB7 - CL8

Por lo que respecta al tercer circuito de puerto paralelo, en el diagrama localizado con 8255-2, se instaló para futuros usos y expansiones del sistema.

#### 3.4. INTERRUPCIONES

Para estar al tanto de los eventos que suceden en el exterior, un microprocesador debe de ser capaz de detectar peticiones de interrupción para su atención. Por ejemplo, una computadora cómo sabe cuándo debe leer el puerto del teclado?. En este caso el teclado envía una señal al microprocesador cuando una tecla es oprimida y éste recibe y decodifica la señal.

La señal que envía el teclado es una Interrupción y el sistema de manejo de interrupciones del microprocesador es el responsable de darle la atención requerida. Una interrupción puede presentarse en cualquier momento, y no hay manera de predecir lo que hará el microprocesador cuando ocurra. Por esto es necesario que el mecanismo de interrupción incluya dos requisitos importantes. Primero, debe ser posible deshabilitar el reconocimiento de la señal de interrupción en el microprocesador. Segundo, cuando la señal de interrupción es recibida, el microprocesador debe ser capaz de salvar la información de lo que está

procesando en ese momento. Cuando se presenta una interrupción, pasa algo muy similar que cuando se llama una subrutina, pues primero se almacena el registro de banderas en el stack y la interrupción y sus banderas, IF y TF, se limpian (esto se hace con el fin de prevenir que una segunda interrupción interrumpa al procesador mientras la primera está siendo procesada). Entonces, el registro CS (definiendo el código de segmento común) es también almacenado en el stack. Finalmente, la dirección de desplazamiento de la siguiente instrucción se ejecuta (se toma del registro IP) y se guarda en el stack. Una vez que el estado del microprocesador ha sido salvado, el mecanismo de interrupción transfiere el control a una rutina especial en una dirección específica. Esta rutina se llama Rutina de Servicio de Interrupción y es la responsable de manejar el evento externo que ha sido llamado. Cuando esta tarea ha sido realizada, el control retorna a la línea que se estaba ejecutando cuando la interrupción fue recibida. Ahora cómo determina el microprocesador la dirección donde se localiza el servicio de interrupción?. El 8088 resuelve este problema permitiendo identificar cada posible señal de interrupción con un tipo de código. El tipo de código es identificado por un byte, por tanto, podemos tener 256 señales de interrupción únicas.

Cuando el microprocesador recibe una señal de interrupción, también espera un tipo de código de un byte que el dispositivo de interrupción envía en forma de código para identificarlo. El microprocesador usa el tipo de código para determinar la dirección apropiada de la rutina de servicio de interrupción. Los primeros 1024 bytes (1k) de memoria (dirección física 00000H-003FFH) se reservan para una tabla de direcciones de rutinas de servicio de interrupción.

Dadas las características de nuestro sistema es necesario añadir, aparte del espacio de memoria RAM mencionado, un Controlador de Interrupciones que ayude al microprocesador en esta tarea. El dispositivo que se eligió para este fin es el 8259A de INTEL cuyas características se describen a continuación.

#### 3.4.1. Controlador de Interrupciones Programable 8259A

El 8259A es un circuito de soporte para el 8088. Este Controlador Programable de Interrupciones cuenta con varias líneas de interrupción, mientras el 8088 cuenta únicamente con una. El 8259A acepta 8 líneas de señales de interrupción independientes, numeradas de 0 a 7. Para cada interrupción que recibe, presenta una señal de interrupción al microprocesador. Estas 8 señales se alambran sobre el bus de control para que cualquier dispositivo conectado a éste pueda acceder este mecanismo de interrupción. Las señales son identificadas de IRQ0 a IRQ7 (FIG. 3.8).

Cada señal es independiente, por esta razón debe preverse la posibilidad de que 2 o más señales ocurran al mismo tiempo. El 8259 maneja este evento reteniendo la segunda interrupción, mientras el microprocesador le da servicio a la primera. Cuando acaba el servicio, envía la segunda al microprocesador. Para eventos que ocurren en el mismo momento, el 8259 tiene una prioridad de orden, donde la señal de interrupción 0 tiene la más alta prioridad y la señal de interrupción 7 tiene la más baja. El microprocesador le indica al controlador cuando se ha completado el servicio de interrupción de cada una. Por esta razón se configuran los tipos de código de la dirección 08H-0FH, en

correspondencia con las líneas de interrupción 0-7. Desde este punto de vista, la programación del controlador consiste de 2 acciones básicas.

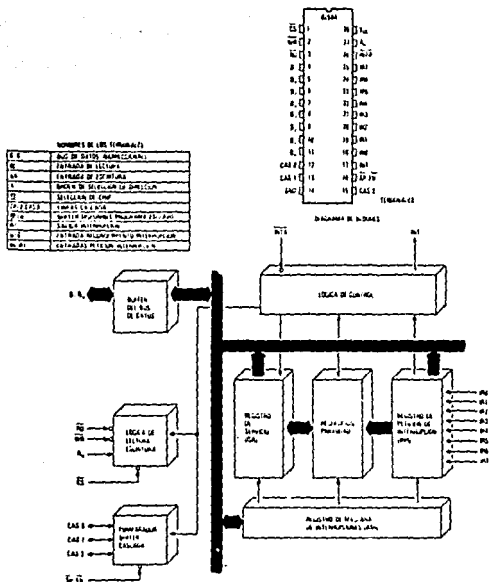


FIG. 3.8. Controlador de Interrupciones Programable 8259.

La primera es que podemos habilitar o deshabilitar cada interrupción independientemente escribiendo un valor dentro del registro de interrupción mascarable, IMR. El IMR es un registro de un byte incluido en el 8259. Cada bit del IMR corresponde a un tipo de interrupción. Si el bit es 0, entonces la interrupción correspondiente es habilitada y si el bit es 1, entonces la interrupción es desmascarada, deshabilitando la interrupción. La segunda acción de programación, tiene que ver con la señal de final de la rutina de servicio de interrupción y se lleva a cabo enviando el "final de interrupción" por medio del commando EO1. En la figura se puede ver la configuración del 8259A, junto con la del Timer 8253. Esta parte del sistema se describe posteriormente.

### 3.5. TEMPORIZADOR PROGRAMABLE 8253

Este circuito puede producir varias señales y/o funciones de conteo diferentes. En su interior hay 3 contadores independientes, numerados como 0, 1 y 2. Cada uno de estos "canales temporizables" puede programarse de 6 modos diferentes, referidos como Modo 0 hasta Modo 5. Una vez que han sido programados, todos los canales pueden producir sus funciones de tiempo simultáneamente. El hardware relacionado con cada canal es idéntico. Cada canal contiene un registro latch de 16 bits y un registro contador de 16 bits. Cada canal tiene también 2 señales de entrada dedicadas, llamadas CLOCK y GATE, así como una señal de salida. En general, se programa un valor contable dentro del registro latch y cada vez que aparece un pulso en la entrada del reloj, el valor en el registro contador es decrementado por uno. Cuando el contador alcanza el cero, se



genera una señal en la línea de salida. El modo en el cual se programa el canal temporizable determinará exactamente cómo se lleva a cabo cada una de estas operaciones.

El 8253 se programa escribiendo dentro del registro de comandos de 16 bits. Inclusive, cada canal tiene un puerto dedicado que se usa para leer/escribir el contenido de su correspondiente registro latch.

La programación de un canal temporizable siempre se inicia escribiendo una instrucción en el registro de puerto. Los 2 bits más significativos se usan para seleccionar qué canal se programará. El resto de los bits en el comando define cómo operará ese canal y cómo se comunicará con su registro latch de 16 bits. Pero sólo se pueden acceder 8 bits, por esta causa podría ser necesario leer ó escribir 2 veces el registro latch. Para acceder el registro latch entero, se colocan los bits 4 y 5 en "1" en el registro de comandos. Si se necesitara acceder el LSB o el MSB de la mitad del registro, se puede usar una simple instrucción IN ó OUT después de seleccionar la mitad deseada vía bits 4 y 5 del registro de comando. El modo en el cual el canal operará se establece por medio de los bits 1,2 y 3 del registro de comando. Finalmente el bit 0 determinará la forma en la que el registro contador decrementará el valor, ya sea en BCD ó binario.

Cada uno de los modos se describe en la FIG. 3.9.. Para cada uno de estos modos, la señal en la entrada de reloj determinará el formato en el cual el contador se decrementará. La señal de entrada GATE puede habilitar o deshabilitar el reloj dentro del contador, o puede usarse

para señalar el inicio de la operación de conteo, dependiendo del modo. El resultado de cualquier operación de conteo es la aparición de una señal en la línea de salida que puede ser un simple pulso, una salida constante o una señal repetida periódicamente, dependiendo del modo elegido.

En cada uno de estos modos, el registro latch actúa como un lugar donde retener el valor de cuenta destinado para el registro contador.

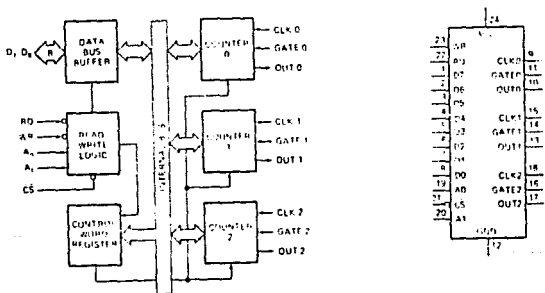


FIG. 3.9. Circuito Temporizador y Contador 8253.

### 3.6. CONVERTIDOR ANALOGICO-DIGITAL

Un convertidor A/D, como su nombre indica, convierte tensiones analógicas en una representación digital compatible con la entrada del sistema digital. Si se pretende leer una señal de 10 V con un convertidor de 8 bits, la resolución será  $1/256$  de 10 V (o 40 mV) y la precisión será de  $+1/2$  LSB.

Cuanto mayor sea la resolución, tantos más bits se necesitan. El número de bits no establece la escala de trabajo de un convertidor, sólo determina lo bien que se representa el valor. Un convertidor de 8 bits (A/D o D/A) puede prepararse con la misma facilidad para cubrir un intervalo de 0 a 1 V como para otro de 0 a 1000 V. Se entiende, por supuesto, que con una escala de 1000 V y un convertidor de 8 bits la resolución es de 4 V. Tal unidad sería inútil en señales de 0 a 10 V. El problema se puede solucionar de varias formas, pero la solución más fácil es utilizar un convertidor con más bits. Un convertidor de 16 bits tiene 65536 ( $2^{16}$ ) pasos, en vez de 256 ( $2^8$ ) y cubriría el mismo intervalo de 1000 V en incrementos de 15 mV.

La conversión A/D es considerablemente más cara que la D/A, pues el precio está directamente relacionado con la resolución y con la exactitud. Un convertidor A/D que explora sondas de termistor y proporciona datos para controlar la temperatura en un laboratorio grande puede costar 5 dólares, pero no puede codificar información de video procedente de un lector óptico. La mayoría de las técnicas de conversión A/D que se conocen hasta ahora son por lo regular de 8 bits. Esta

característica los hace de una resolución limitada y además son dispositivos de un solo canal. Son adecuados para mediciones de temperatura en un Sistema de Calefacción Solar, pero es dudoso que tengan la resolución adecuada para controlar el gradiente de temperatura a lo largo de un conducto de calefacción. Los sensores utilizados para medir tales parámetros necesitarían tener una resolución más alta que los sensores de temperatura del ambiente. Para una escala entre  $-20^{\circ}$  y  $108^{\circ}$  C, un A/D de 8 bits podría dar una resolución de  $0.5^{\circ}$ .

En una aplicación de calefacción solar, considerando las variaciones de movimiento del aire, cobertura de nubes y modelos generales de clima, ésta es toda la resolución que se necesitaría. Dentro del sistema, sin embargo, hay zonas que exigirían mediciones más estrictas.

Un sistema solar es un ejemplo típico. Después de la instalación, el siguiente paso suele ser investigar cómo aumenta su rendimiento funcional. Nueve veces de cada diez, esto requiere reducir las pérdidas de calor en los tubos y conductos. Una forma de determinar tales pérdidas es poner sensores de temperatura a lo largo del recorrido de distribución del calor y buscar las zonas frías. Las diferencias medidas entre sensores pueden ser muy pequeñas, apenas unas décimas de grado o de un orden de magnitud similar, pero las pérdidas totales podrían ser significativas. Medir temperaturas con una precisión décimas o de centésimas de grado y mantener el mismo rango dinámico requiere una resolución de más de 8 bits. Se necesitan de 10 a 12 bits.

La situación se complica más por el gran número de puntos que se

puede necesitar controlar dentro de un sistema. Es raro encontrar solamente un indicador de temperatura en un sistema.

Muy pocos sistemas de adquisición de datos utilizan un solo canal. Suelen suministrarse con 8 ó 16 canales multiplexados. La entrada de un convertidor A/D se conmuta (normalmente sobre una base de demanda) entre los canales y los resultados se compilan y promedian mediante el Sistema Digital. Esta información puede registrarse en una cinta de grabación, transmitirse en serie a otro sistema o utilizarse para la ejecución de una presentación visual en tiempo real.

Cuando se ha concluido la conversión, se dispondrá de 12 bits de datos en paralelo. La mayoría de estos convertidores son costosos porque están concebidos para dar la apariencia de convertidores en paralelo. Al habilitar la línea de habilitación de conservación y su código zonal, hay 12 bits de respuesta. Cuando el sistema requiere estos datos, explora, manipula y los almacena en una tabla para su empleo en otros programas.

El Sistema Digital que aquí se describe fue diseñado con base en las premisas de la discusión anterior. La solución para la cuestión de alta resolución frente a los gastos viene dada en la forma de un CI de conversión A/D multiplexado de 3 1/2 dígitos. El circuito CMOS MC14433 fue concebido principalmente para su empleo en voltímetros digitales, pero, debido a su versatilidad, puede utilizarse en otras aplicaciones. Se trata de un convertidor de 11 bits de un solo canal, pero se denomina de 3 1/2 dígitos. La salida es BCD (decimal codificado en binario) y cubre específicamente un rango de -1999 a + 1999 en su conteo.

### 3.6.1. Convertidor A/D de 3 1/2 dígitos MC14433

Las especificaciones básicas del circuito son las siguientes:

Exactitud: + 0.05 % de lectura + 1 cuenta

Dos escalas de voltaje de 1.999 V y 199.9 mV

25 conversiones por segundo

Impedancia de entrada de 1000 Mohms

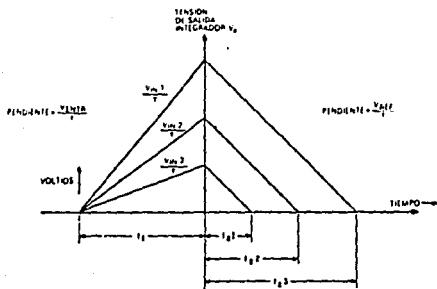
Ajuste automático del cero

Ajuste automático de la polaridad

Disponibilidad de señales de rango por exceso, por defecto y automático

El MC14433 es un convertidor A/D integrador de doble rampa modificado como se muestra en la FIG. 3.10.. La secuencia de conversión está dividida en dos períodos de integración: desconocido y de referencia. Durante la secuencia de integración de  $V_{in}$  (entrada desconocida), la tensión desconocida se aplica a un integrador con una constante de tiempo de integración definida para un límite de tiempo predeterminado. La salida de voltaje del integrador se hace, pues, una función de la entrada desconocida. Cuanto más positiva es la entrada, tanto más alta es la salida del integrador.

Durante el segundo ciclo de la secuencia de integración, una señal de referencia de 2 Volts. está conectada a  $V_{in}$ . Ello hace que el integrador se desplace hacia cero, mientras que la circuitería digital del chip lleva la cuenta del tiempo que toma llegar a cero. La diferencia



$T_1$  - CONSTANTE DE TIEMPO DE INTEGRACION  
 $T_2$  - PERIODO DE INTEGRACION DE TENSION DE SECONDA EDONSTANTE  
 $T_3$  - PERIODO DE INTEGRACION DE TENSION DE REFERENCIA (VARIABLE)

$$V_o = \frac{V_{ref}}{T_1} T_1 + \frac{V_{ref}}{T_2} T_2$$

ES DECIR

$$\frac{V_{o1}}{V_{ref}} = \frac{T_1}{T_2}$$

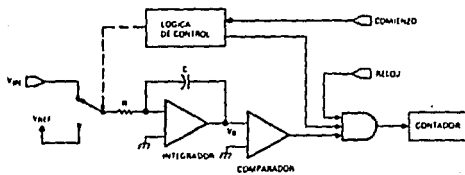


FIG. 3.10. Convertidor Analógico/Digital de Doble Rampa.

de tiempo entre las dos secuencias de integración es entonces una función de su diferencia de voltaje.

Si 2 V fuera la  $V_{in}$  aplicada, entonces  $t_2$  sería igual a  $t_1$ . La tensión desconocida es equivalente a la relación de los periodos multiplicada por el voltaje de referencia ( $V_{ref}$ ). La escala completa del convertidor viene determinada por  $V_{ref}$ . El cambio de  $V_{ref}$  a 0.200 V hará que la salida de conteo de 1999 represente 199.9 mV en lugar de 1.999 V de escala completa.

La salida del chip es una combinación de datos en serie y en paralelo. Hay cuatro salidas de selección de dígitos y cuatro líneas de datos en BCD:

#### Salidas de datos en BCD

Terminal 23	Q3 (MSB)
Terminal 22	Q2
Terminal 21	Q1
Terminal 20	Q0

#### Salidas de selección de dígitos

Terminal 19	DS1 (MSD)
Terminal 18	DS2
Terminal 17	DS1
Terminal 16	DS0



Con respecto a lo que el sistema ve a través de las memorias intermedias de salida del 74LS04, la salida de selección de dígitos está a nivel bajo cuando se selecciona el dígito respectivo. El dígito más significativo (1/2 DS1) pasa a nivel bajo inmediatamente después de un pulso de EOC (final de conversión) y va seguido por los dígitos restantes en una secuencia desde el MSD. La frecuencia del reloj de multiplexado es la del reloj del sistema dividida por 80. Se intercalan 2 periodos de reloj entre las salidas de los dígitos.

Durante DS1, la polaridad y algunos bits de estado están disponibles. La polaridad está en Q2 y el valor 1/2 dígitos está en Q3. Si Q2 es un <1> , entonces el 1/2 dígito es un 0.

El MC14433 está ajustado para 25 conversiones en un segundo. El LM325 es el componente que suministra la señal VREF. Dado su comportamiento con la temperatura de trabajo, el valor nominal que proporciona es de 2 V. Una vez concluida la conversión, una señal activa el pin PC0 del 8255. Esta señal se conecta a un relevador que se encarga de abrir o cerrar en cuanto inicie o termine la conversión. Para lograr la operación multicanal, el sistema cuenta con tres circuitos CD4051 conectados al puerto paralelo 8255. Así, para configurar la operación de 24 canales, las 8 señales de entrada con que cuenta el CD4051 son habilitadas por el puerto B (PB0-PB7). Cada uno de los canales es llamado por el Programa Monitor para completar la operación. Cuando la señal de PC0 se encuentra en bajo, se aplica 2 V a la entrada de VREF y se ajusta la constante de tiempo de integración con una resistencia de 82 kohms. Con BS = 0, VREF = 0.2 V, y la resistencia de integración es de 10 kohms.

### 3.7. MODULACION POR DESPLAZAMIENTO DE FRECUENCIA

El sistema desarrollado tiene un almacenamiento eficiente de datos, pero, a menos que sean transferidos a una memoria permanente, se perderán cuando se desconecta la alimentación. Por supuesto que el sistema puede dejarse energizado constantemente, pero qué pasa si se quiere almacenar un segundo grupo de datos de magnitud similar al primero?

La solución ideal es poseer algún medio que almacene grandes bloques de memoria temporalmente.

En sistemas de computadoras grandes esto se consigue mediante uno o más discos duros y sistemas de cintas magnéticas de 9 pistas. Estos medios de almacenamiento de alto volumen y alta velocidad son de gran ayuda, pero el costo de los mismos es, la mayoría de las veces, demasiado alto. Una alternativa de bajo costo y menos capacidad es un sistema de memoria de cassette de audio.

En general, una interfaz para almacenamiento en cassette consiste en 3 subsistemas principales:

1. Un Transmisor/Receptor en serie.
2. Un montaje de hardware que convierte los datos en serie en niveles TTL, de forma que sea posible su grabación en cassette de audio.
3. Un programa de aplicación que esté al corriente de lo que está

saliendo a la cinta y que puede cargar estos datos en el lugar adecuado.

La configuración básica se ilustra en el Diagrama de Bloques de la FIG. 3.11..

La sección de transmisor/receptor en serie no es más que una interfaz en serie USART 8251, que se describirá más adelante. Se comunica mediante cable RS-232C, con los convertidores MC-1488 y MC-1489 conectados a sus líneas en serie. Sin embargo, si se conecta una interfaz de cassette a estas líneas, se obtiene un mayor beneficio de este arreglo de circuitos como medio de almacenaje. Un beneficio adicional es que los datos en serie generados por el USART ofrecerán cierta compatibilidad con sistemas de computación personal. Se obtienen velocidades de transmisión de datos normales y un protocolo de comunicación en serie normal.

La salida del USART es TTL. Incluso con el controlador RS-232, la salida lógica sigue siendo un nivel de c.c. Como las grabadoras de audio no pueden registrar dc, la salida del USART debe ser convertida de alguna manera a ac. La solución es Modulación por Desplazamiento de Frecuencia, FSK. La salida TTL procedente del USART se convierte en tonos de audio.

Una frecuencia representa un nivel lógico "0" y la otra representa un nivel lógico "1". La FIG. 3.12. muestra un circuito que producirá tonos de audio modulados por desplazamiento de frecuencia. Se obtiene una frecuencia de referencia de 4800 Hz a partir de un generador de velocidad de transferencia de datos MC14411.

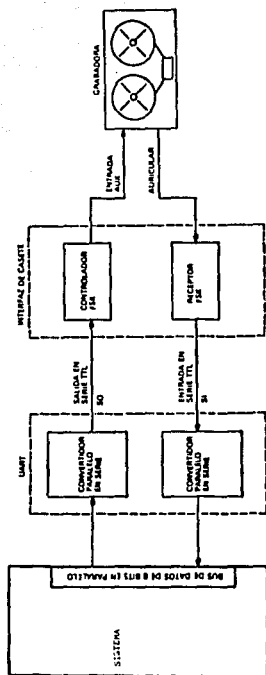


FIG. 3.11. Diagrama de Bloques del Sistema de Almacenamiento de Datos.

Los circuitos integrados CD4027 funcionan como una cadena de divisor programable. Un nivel lógico 1 en el circuito 2 de entrada divide los 4800 Hz por 2, resultando una salida de 2400 Hz. Cuando el nivel de entrada se cambia a un nivel lógico 0, lo divide por 4, produciendo una salida de 1200 Hz. Las frecuencias de FSK son generadas a una velocidad de salida en serie de 300 bps y están conectadas directamente a la grabadora a través de la entrada auxiliar o de micrófono. (Estas frecuencias y la velocidad de transferencia de datos son, a menudo, referidas como KANSAS CITY STANDARD).

Para obtener sonidos grabados a partir de la cinta de audio se requiere el circuito que se muestra en la FIG. 3.13.. En general, consiste en un par de filtros pasabanda y un comparador de tensiones. La grabadora se ajusta a un nivel de salida de 1 V pico a pico. Este nivel no es crítico porque es amplificado y limitado cuando pasa a través del CI1, CI2 y CI3 que son filtros pasabanda con frecuencias centrales de 2400 y de 1200 Hz, respectivamente. La salida del CI1 es alimentada a ambos, pero debe pasar por uno sólo de ellos. CI4 compara la salida de los dos filtros y genera un nivel lógico 1 cuando un tono de 2400 Hz es recibido, lo mismo que cuando se recibe uno de 1200 Hz.

La elección de las frecuencias FSK y de la velocidad de transferencia de datos no se realiza al azar. Estas son una función de la velocidad de respuesta del receptor y del ancho de banda de la grabadora. La mayoría de las grabadoras de cassette tienen una respuesta de frecuencia de unos 8 kHz. Unidades menos costosas pueden ser de un nivel de 5 o 6 kHz. No es prudente tratar de grabar sonidos que pasen de este



limite superior. El centro del intervalo de frecuencias ofrece mayor confiabilidad, por lo que el tono del nivel lógico 1 del FSK debe ser ajustado a menos de 3 kHz (2400 Hz). En nuestro caso, además, lleva cierto tiempo al receptor reconocer una frecuencia particular. El circuito de la FIG. 3.12. tarda 2 o 3 ciclos en responder. Ello quiere decir que, a la baja frecuencia de 1200 Hz, cada bit de nivel lógico 0 necesitará 3 ciclos a 1200 Hz para ser identificado.

Si consideramos una condición del caso más desfavorable de emisión de solo "ceros", la velocidad de transmisión podría tener que ser más lenta que 400 bps para ser recibida exactamente. La velocidad de transferencia de datos normal más próxima a este valor es de 300 bps. La elevación al tono de 1200 Hz para incrementar la velocidad de transmisión solamente complica el diseño del filtro cuanto más próximo está a 2400 Hz. Este diseño ha sido probado a 4800 bps, pero requiere de un ajuste especial para alcanzar velocidades más elevadas. Las frecuencias bajas y las velocidades de transferencia moderadas son escogidas específicamente para incrementar la probabilidad de construcciones satisfactorias en vez de competir con sistemas de almacenaje de datos de alta velocidad.

### 3.8. INTERFAZ EN SERIE USART 8251

Algunos dispositivos de E/S tales como el CRT y los cassettes transmiten y reciben datos en forma serie. Si una computadora va a transmitir o a recibir datos en serie deberá tener una interfaz con lógica capaz de convertir los datos serie a datos paralelos y viceversa.

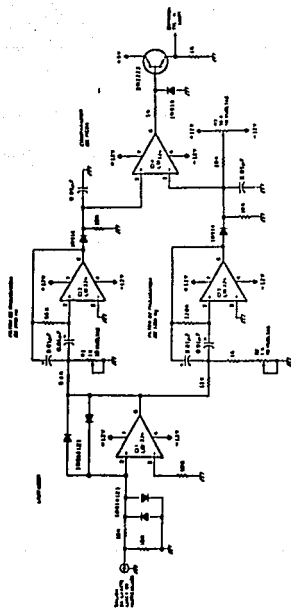


FIG. 3.13. Demodulador de FSK.



La comunicación es, simplemente, la transferencia de información desde un dispositivo a otro. En el caso de una unidad de visualización de CRT, la computadora envía información de caracteres para visualización en pantalla, mientras que el teclado transmite la entrada de datos del usuario a la computadora. Cada extremo de la línea de comunicación de full dúplex debe tener un receptor y un transmisor. En ambos casos, la información que se transfiere son datos ASCII normalmente constituidos por un código de 7 bits y, en algunos casos, un bit de paridad adicional para comprobación de errores. Los datos de 7 bits (ignorando el bit de paridad) aparecerán en las líneas de un puerto paralelo. Estas 7 líneas más una referencia de tierra y una señal de habilitación "strobe" pueden llevarse a la entrada del CRT. Si además deseamos tener una línea similar entre la salida del teclado y un puerto paralelo de 8 bits en la computadora, ello exige 9 líneas adicionales. Para complicar todavía más el problema, separemos la terminal y la computadora una distancia de 100 a 120 m., el resultado es que 120 m. de cable de 18 hilos (17 si se combinan referencias de tierra) costarán más que el terminal. También ha de considerarse que la salida en paralelo TTL no debe emplearse para excitar líneas más largas que 6 metros, sin buffers/excitadores especiales. De no ser así, podrían producirse errores.

La solución a este problema de cableado costoso es utilizar la comunicación en serie en lugar de paralelo. Los datos en paralelo se convierten en serie y se envían, un bit cada vez, a un hilo de par trenzado único. Si se necesitan buffers/excitadores para largas distancias, se requieren menos con el procedimiento en serie. Los bits de inicio y de parada ("start" y "stop") especialmente codificados, e

incluidos en la transmisión en serie, indican al receptor que se están enviando datos válidos. Para el ejemplo anterior, solamente dos pares de hilos se necesitan para realizar una interacción full dúplex. En el modo de "semidúplex" puede reducirse a un par trenzado único, pero la sincronización de la línea de comunicación compartida es más complicada. Todas las referencias de transmisión en serie que haremos se limitarán al funcionamiento en full dúplex completo.

Ahora en que hemos visto por qué es conveniente la comunicación en serie, describiremos el 8251, dispositivo por medio del cual se lleva a cabo la conversión paralelo a serie.

La FIG. 3.14. muestra el diagrama de bloques y terminales del USART 8251A. Este circuito puede ser habilitado de varias formas de acuerdo con cada uno de sus bloques de funcionamiento. Cuenta con una sección de control de modem para transmisión o recepción de datos. La gran versatilidad de este dispositivo reside en que puede funcionar ya sea de manera sincrónica o asincrónica. De manera sincrónica los datos son transmitidos por grupos, de acuerdo a un carácter de sincronía y una señal de reloj, mientras que de manera asincrónica funcionan independientemente de la señal de reloj, mandando bits de inicio y de fin.

La manera en que se controla es por medio de la palabra de Modo y de Comando. En la palabra de Modo se especifica el factor Bauds por Segundo, Longitud del Carácter de Transmisión (5,6,7 ó 8 bits), Paridad Habilitada, Generación de Paridad y Número de Bits de Parada.

En la palabra de Comando se determina el Bit de Listo para Transmitir, Terminal de Datos Lista, Habilitar Recepción, Enviar Caracteres de Paro, Error de Inicio, Solicitud para Transmitir, Reseteo Interno y Modo de Espera de Datos (Hunt).

Otra característica importante del 8251 es que cuenta con varias velocidades de transmisión de datos, dependiendo la señal de reloj que reciba del Generador de Pulsos 14411. Las más comunes son del orden de 300 bps, 600 bps, 1200 bps, 2400 bps, 4800 bps, 9600 bps y, en sistemas de alta velocidad, 19200 bps. Las especificaciones de funcionamiento del 8251A para este sistema son:

Modo de transmisión asíncrono

Generación y verificación de paridad impar

Longitud de carácter igual a 7 bits

Habilitación de la transmisión y recepción de datos

Velocidad de transmisión y recepción igual a 4800 bits/s.

Hay dos consideraciones más acerca del hardware: La Velocidad de Transmisión de Datos y el Nivel de las Señales de Transmisión. Ha de tenerse presente que a una velocidad de datos más baja, solamente 8 de 11 bits de cada palabra transmitida son datos con un bit de comienzo y 2 de parada. Sin embargo, la velocidad de transmisión es parte solamente de los prerequisites de la intercomunicación.



### 3.9. INTERFAZ SERIE RS-232C

La interfaz eléctrica RS-232C es un estándar para la conexión de componentes del sistema como modems, impresoras y computadoras. El estándar fue establecido por Electronic Industries Association (EIA), una organización de comercio industrial. La RS-232 define un camino de 25 conductores que conforma 18 circuitos con retorno a través de tierra. El estándar también define los voltajes, los niveles del 0 y 1 lógicos, usados en todos los circuitos. Una ventaja adicional importante es que no se produce ningún daño físico como resultado de un corto circuito dentro de los cables o conectores.

Por qué necesita el RS-232C 25 conductores?. El estándar fue definido por un comité y la interfaz resultante tenía que satisfacer las necesidades de una industria entera. La mayoría de las PCs existentes en el mercado, utilizan como mucho 9 conductores, pero a menudo se las arreglan sólo con 3.

Las señales lógicas dentro de un Sistema Digital, por ejemplo, una PC, se conforman con lo que se llama "niveles TTL", un estándar para la interconexión de circuitos integrados. Un voltaje entre 2 y 5 Volts es un "1" lógico, y un voltaje entre 0 y 0.8 Volts es un "0" lógico. Estos niveles no se usan fuera de la PC a causa de su insuficiente inmunidad al ruido eléctrico. En su lugar se utiliza el esquema que se muestra en la FIG. 3.15.. Podemos observar que los umbrales en el extremo receptor son distintos a los del extremo emisor debido a la degradación de la señal en el cable. Recuérdese que es el voltaje el que define el estado lógico de

un circuito, no la presencia o ausencia de corriente.

Entonces, las salidas de voltaje para todas las señales RS-232C son como sigue. Un "1" lógico, o marca, es un voltaje entre  $-3\text{ V}$  y  $-15\text{ V}$  bajo carga (  $-25\text{ V}$  sin carga ). Un "0" lógico, o espacio, es un voltaje entre  $+3\text{ V}$  y  $+15\text{ V}$  bajo carga (  $+25\text{ V}$  sin carga ). Voltajes tales como  $\pm 12\text{ V}$  se usan comúnmente.

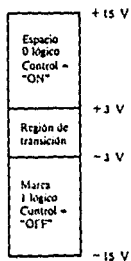


FIG. 3.15. Niveles Lógicos Definidos para RS-232

Una PC y la mayoría del equipo de comunicaciones producen señales de salida de + y -12 Volts. Sin embargo, una entrada de + o -3 Volts es suficiente para definir un estado lógico. La FIG. 3.16a. muestra una señal limpia entrando por un cable de 100 m de longitud. la señal ruidosa que emerge por el otro extremo aparece en la FIG. 3.16b. y la FIG. 3.16c. muestra cómo aparece la señal dentro de la PC receptora. Recuérdese que la señal de la RS-232C tiene la polaridad cambiada respecto a la señal TTL de dentro de la PC. Un "1" lógico TTL de 3 Volts equivale a "1" lógico o marca de menos 12 Volts.

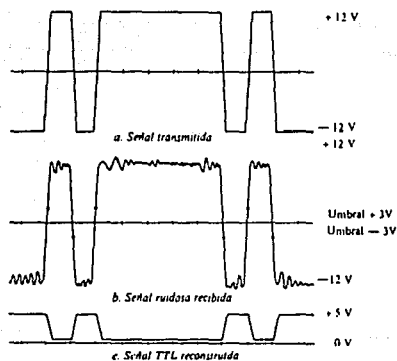


FIG. 3.16. Transmisión y Recepción de una Señal por Medio de RS-232.

Los cables terminan en conectores, pero la constitución física del conector no está definida por el estándar. Los conectores tienen terminales, y los terminales están numerados del 1 al 25.

La mayoría de las tareas de comunicación de datos no utilizan todas las terminales de la interfaz. Las funciones más importantes son Transmitir Datos y Recibir Datos, terminales 2 y 3.

Estos son los dos conductores por los que los datos serie se envían y reciben simultáneamente. El resto de las terminales, con excepción de la tierra (pin 7), son Terminales de Control.

Las señales de control RTS y CTS no se usan tanto como las otras, habiendo sido diseñadas para controlar la comunicación Half-Duplex.

Algunos programas para PC utilizan la línea RTS de forma impredecible. Las impresoras serie utilizan a menudo los circuitos de control para indicar que la impresora está ocupada o desconectada.

Ahora se describirá cómo se pueden sincronizar estas señales con los circuitos exteriores.

### **3.10. CIRCUITO LINEAL 1488**

Este circuito es un emisor lineal que sirve como interfaz entre el Equipo Terminal de Datos y el Equipo de Comunicación de Datos cuando



se desean enviar caracteres de acuerdo con la especificación RS-232C.

Algunas se las especificaciones de este circuito son:

Corriente limitada de salida +/- 10 mA.

Control simple de SR con un capacitor externo.

Compatible con las familias Motorola MDTL y Lógica MTL.

Por medio de este circuito se logran enviar los datos de la Tarjeta de Adquisición a la cinta de cassette.

### 3.11. CIRCUITO LINEAL 1489

Al igual que el 1488, este circuito es un receptor lineal entre el Equipo Emisor de Datos y el Equipo Terminal de Datos. Algunas de las especificaciones de este circuito son :

Resistencia de Entrada de 3.0 a 7.0 kOhms.

Intervalo de Señal de Entrada de +/- 30 Volts.

Entrada de histerésis en "tres estados".

Por medio de este circuito se logra recibir los datos que manda la cinta de cassette a la Tarjeta de Adquisición.

### 3.12. GENERADOR DE FRECUENCIA MC14411

Para el funcionamiento adecuado de los circuitos que necesitan cierta señal de reloj para trabajar correctamente, es necesario contar con un circuito que genere la señal de la frecuencia necesaria.

Ejemplos de este tipo de circuitos son el 8251A que necesita una frecuencia de 4800 Hz, el 8253 según la frecuencia de salida deseada, en este caso también de 4800 Hz., que también se utilizan en este sistema. El circuito que resuelve la necesidad de contar con varias frecuencias a la vez es el MC14411 de Motorola.

Su principio de funcionamiento reside en que contiene una red divisora de frecuencia, que le permite dar una amplio intervalo de salidas de frecuencia. La fuente de reloj para la red es un oscilador de cristal. La salida de frecuencia deseada se selecciona por medio de una dirección de 2 bits. A continuación se enumeran las características de este dispositivo.

- Alimentación de 5 V dc (+/- 5%)
- Oscilador de cristal controlado interno para estabilidad (1.8432 MHz)
- 16 Salidas de Frecuencia diferentes
- Base de tiempo programable para una de 4 salidas múltiples
- Salidas por medio de buffers compatibles con TTL
- Inmunidad al ruido, 45% de  $V_{DD}$  típico
- Protección por medio de diodos en todas las entradas

### 3.13. RELOJ DE TIEMPO REAL

Una vez que el sistema ha llevado a cabo el registro de los datos, es necesario que almacene las lecturas en un medio de almacenamiento permanente como puede ser un cassette. Esto puede realizarse en forma continua o a intervalos regulares, lo cual hace deseable que el sistema incluya un Reloj de Tiempo Real, de tal forma que además de registrarse las lecturas puedan registrarse los tiempos en que fueron efectuadas.

Un reloj de tiempo real permite tener siempre dentro del sistema un registro de tiempo que incluya tanto la fecha como la hora en todo instante. Esto se puede realizar utilizando aplicaciones de software o hardware. Una técnica sencilla es utilizar un circuito de reloj para proporcionar las marcas de tiempo a la línea de interrupción no mascarable del microprocesador cada segundo. Cuando el sistema indica que recibió la interrupción, primero, salva todos los registros del programa que estaba ejecutando y luego atiende a la interrupción de tiempo real. Frecuentemente, la primera acción de la rutina de atención de esta interrupción es incrementar un contador interno que mantiene el registro del tiempo transcurrido y suele ser un valor equivalente al número total de marcas de reloj, ya sea en segundos o en milisegundos. Una vez que ha sido establecido este intervalo periódico, es más sencillo para el sistema realizar funciones de tiempo real.

Las resoluciones de reloj hasta milisegundos parecen aconsejables pues hacen que la temporización de los intervalos tenga gran precisión. Sin embargo, este sistema no es muy conveniente dada la cantidad de

software que implica y el tiempo que consume para ejecutarlo, contrario a la depuración que se pretende. Se prefiere un circuito que sea más fácil de implementar y de utilización más eficiente.

Esencialmente, el tipo de Sistema de Tiempo Real más atractivo tiene una resolución de quizá 1 minuto mucho mejor que 1 ms. También es mejor si puede leerse directamente en horas y minutos en lugar de leerlo como si fuera un conteo de reloj total.

La forma por la cual se optó para proporcionar una entrada hora a hora y minuto a minuto al sistema, fue interconectándole un circuito de Reloj de Tiempo Real 58274. Hay dos métodos para diseñar una interfaz de reloj, uno es dejar funcionar el circuito de reloj independientemente del sistema, conectado de tal forma que el sistema pueda controlar las líneas de salida y leer un valor de tiempo cuando así se desee. El otro método, que es más eficiente porque implica menos software, es dar al sistema el control completo sobre el flujo de información del reloj de una manera sincrónica.

Este reloj de tiempo real es accesado en las localidades de decodificación de los puertos que ofrece el mapa de puertos de E/S del sistema. Estas localidades son de la 14h a la 17h.

El diagrama de conexión del reloj de tiempo real se muestra en la FIG. I donde se pueden observar todas las señales necesarias para su funcionamiento.

El 58274 sólo maneja cuatro bits por dato, ya que los registros que contiene son de dicho tamaño, manejando así unidades y decenas de año, mes, horas, minutos y segundos y día de la semana.

Para llevar a cabo la decodificación se utiliza un 74LS138. Este entrega la señal que habilita al circuito de reloj para poder realizar la lectura del tiempo, así como también para actualizar la fecha y la hora.

El 58274 necesita, por especificaciones, un cristal de 32 kHz que junto con un grupo de capacitores, permiten ajustar la base de tiempo para realizar sus funciones de reloj.

### 3.14. DECODIFICACION DE DIRECCIONES DE MEMORIA Y DISPOSITIVOS DE ENTRADA/SALIDA

Para comprender con más claridad el proceso de decodificación de los dispositivos de E/S apuntaremos lo siguiente: la dirección a la que el microprocesador apunta al polarizarlo ó después de un Reset es la FFF0H, por tanto, la memoria EPROM que contiene el programa principal que se encarga de monitorear y controlar el sistema, deberá tener como una de sus direcciones, precisamente la FFF0H.

La memoria EPROM que utiliza este sistema es la 2716-1 de 2k x 8, con un tiempo de respuesta de 350 ns. La memoria RAM es la 6264 de 8k x 8, con un tiempo de respuesta de 150 ns.

FIG. 3.17. Decodificación de Memoria del Sistema.

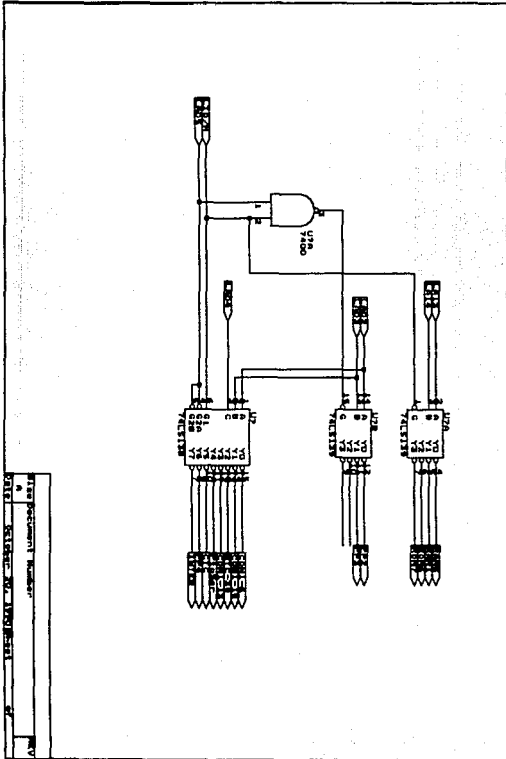


Diagrama de conexión de la memoria del sistema.  
A  
D0 D1 D2 D3 D4 D5 D6 D7

La Memoria EPROM es direccionada de:

FFFFH a F800H      2kbytes

La Memoria RAM es direccionada de:

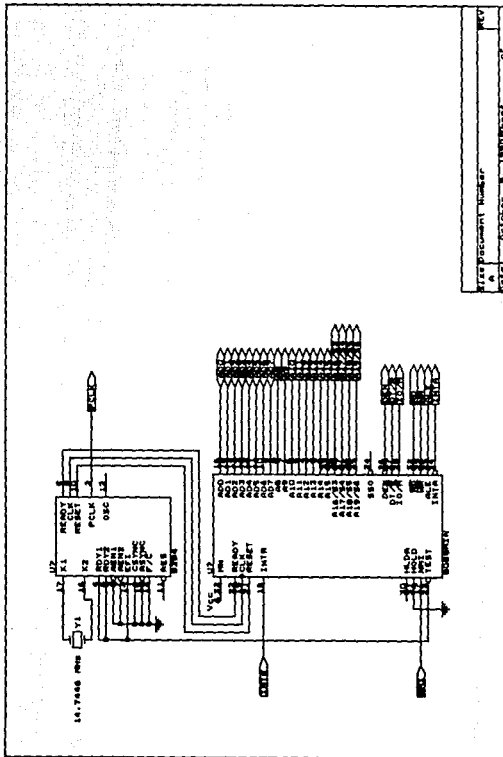
DFFFH a 8000H      8kbytes

En la Fig. 3.17. se muestra el diagrama de la conexión de los decodificadores para las memorias RAM y EPROM. Se utiliza el circuito 74LS139. Con otro 74LS139 y un 74LS138 se completa la decodificación del resto de dispositivos dentro del sistema.

Las memorias serán accedadas mediante la instrucción MOV, habilitando a los decodificadores de memorias mediante la señal (M), que proporciona el microprocesador y que se verifica en bajo.

Al direccionar la memoria EPROM, el decodificador correspondiente genera la señal OE<sub>EPROM</sub>, la cual va conectada al habilitador de salida de la memoria (OE). De la misma forma, el decodificador para la memoria RAM genera la señal CS<sub>RAM</sub>, que se conectará al habilitador de salida de la memoria (OE).

En la FIG. 3.18. , se muestra el diagrama de conexión de las memorias así como el diagrama de conexión para cada uno de los dispositivos periféricos del sistema.



A15 Document Number: \_\_\_\_\_  
 A \_\_\_\_\_  
 A152 - E1082 - 9 - 1978/REV. 3

FIG. 3.18. Diagrama de Conexiones del Sistema.



El diagrama de la FIG. 3.19. ilustra las direcciones de memoria correspondientes a cada dispositivo.

DISPOSITIVO	LOCALIDADES DE MEMORIA																																																																																																																																																																																																																																																																																																																																																																																				
	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0																																																																																																																																																																																																																																																																																																																																																																					
7211-M1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	7211-M2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	58274	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	BUFFER	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	8251	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	7211-M2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	58274	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	BUFFER	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	8251	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																	
7211-M2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	58274	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	BUFFER	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	8251	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																		
	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	58274	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	BUFFER	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	8251	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																			
58274	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	BUFFER	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	8251	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																				
	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	BUFFER	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	8251	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																					
BUFFER	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	8251	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																						
	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	8251	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																							
8251	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																								
	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																									
8253	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																										
	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																											
8255-3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																												
	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																																													
8259	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																																																														
	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																																																																															
8255-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																																																																																																
	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																																																																																																																	
RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																																																																																																																																		
	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																																																																																																																																																			
ROM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																																																																																																																																																																				
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																																																																																																																																																																																																																																																																																																																																					

FIG. 3.19. Mapa de Memoria del Sistema.

## CAPITULO IV

### DISPOSITIVOS AUXILIARES

## CAPITULO IV

### DISPOSITIVOS AUXILIARES

Además de los componentes básicos que conforman la arquitectura de un sistema digital, es necesario proveer los medios para su comunicación con el mundo exterior. Los dispositivos que realizan esta función son los dispositivos auxiliares del sistema y en este caso se tienen básicamente 3:

El Teclado.

El Display de Datos.

La Unidad de Almacenamiento en Cinta.

A continuación se describen los dispositivos auxiliares del Sistema de Adquisición de Datos.

#### 4.1. TECLADO

Conforme a los requerimientos de diseño del sistema en cuanto a la forma de manejo de la entrada y salida de la información, se incluyó un teclado hexadecimal que es suficiente para llevar a cabo las funciones programadas.

Un teclado hexadecimal permite la entrada de datos e instrucciones en forma de números hexadecimales de 2 dígitos. Además de las 16 teclas numéricas, hay tres teclas de órdenes denominadas <<EXEC>>, <<NEXT>> y <<SHIFT>>.

<<NEXT>> se utiliza para indicar la ejecución del comando que se acaba de teclear.

<<SHIFT>> es similar a un teclado regular y se utiliza para doblar el número de código de teclas permitiendo tener más funciones en el mismo teclado. El teclado requerido cuenta con 20 teclas. Su configuración se realizó en forma matricial. La FIG. 4.1. muestra la forma por medio de la cual se conecta el teclado. Los renglones de la matriz se conectan a 4 líneas del puerto de salida. Las líneas de columnas de la matriz se conectan a 4 líneas pertenecientes al puerto de entrada. Cuando no se tiene presionada alguna tecla, las líneas de la columna retienen por medio de las resistencias de pull-up un voltaje de +5 V.

El principio por medio del cual funciona este teclado, consiste en que cuando se oprime una tecla se conecta un renglón y una columna. Si una señal se encuentra en bajo en la salida del renglón y una tecla en ese renglón es presionada, entonces la señal en bajo aparecerá sobre la columna que contiene la tecla, por lo cual se podrá detectar en el puerto de entrada. Si se conoce el renglón y la columna de la tecla oprimida, entonces es posible conocer cuál tecla fue presionada y puede idearse cualquier código que se quiera para representar esa tecla. La manera por medio de la cual funciona el algoritmo para detectar, decodificar y

producir el código hexadecimal para cada tecla es el siguiente:

El primer paso es mandar 0's a todos los renglones por medio del puerto paralelo. A continuación las columnas se leen y verifican una y

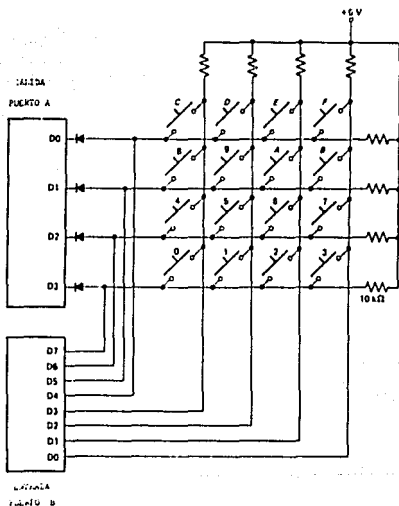


FIG. 4.1. Forma de Conectar un Teclado Matricial a Puertos.

otra vez hasta que se encuentren todas en alto. Esto se hace para asegurarse de que no se ha presionado alguna tecla antes de buscar la siguiente. Una vez que todas las columnas se encuentran en alto el programa entra en un lazo, en el cual espera hasta que aparece una señal en bajo sobre una de las columnas, indicando que una tecla ha sido presionada. Un simple retardo de 20 ms identifica la coordenada de la tecla presionada.

Después de este lazo, se hace otra verificación para ver si la tecla permanece oprimida. Si las columnas se encuentran todas en alto, entonces ninguna tecla ha sido presionada y la detección inicial fue solo un pulso de ruido o un rozón de alguna tecla. Si cualquiera de las columnas está en bajo, entonces se supone que una tecla ha sido oprimida.

La tarea final es determinar el renglón y la columna de la tecla presionada, y convertir esta información de renglón y columna en un código hexadecimal de identificación para cada una de las teclas. Para obtener la información del renglón y la columna, se debe leer una señal en bajo en un renglón mientras las columnas se leen y se pregunta si alguna de ellas se encuentra en nivel bajo. Si ninguna de las columnas se encuentra en bajo, la tecla presionada no está en ese renglón, por tanto la señal en bajo se rota al renglón próximo y las columnas se leen nuevamente. El proceso se repite hasta que una señal en bajo aparece en un renglón, produciendo una señal en bajo en alguna de las columnas. La tecla presionada se encuentra en el renglón con señal en bajo en ese momento.

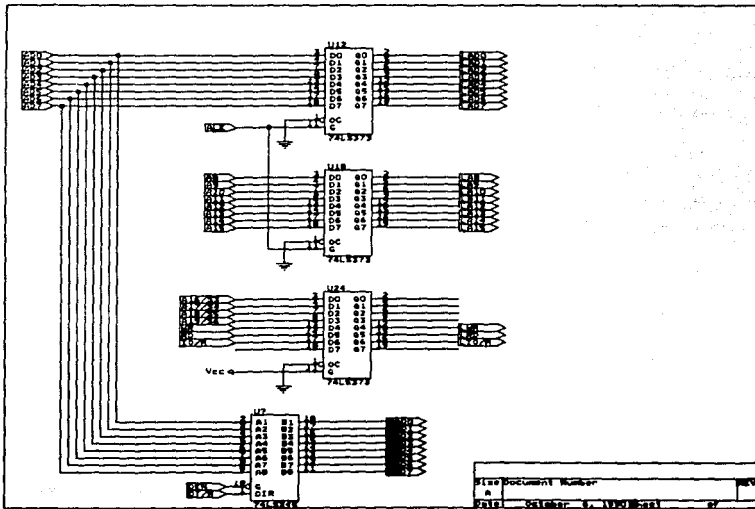
El byte leído desde el puerto de entrada contiene el código de 4 bits que representa el renglón de la tecla oprimida y el código de 4 bits que representa la columna de la misma tecla. Este código de renglón y columna se convierte a código hexadecimal usando una tabla de asignación.

Una vez que ya tenemos el dispositivo por donde introducir nuestros datos para comunicarnos con el sistema, es necesario que los veamos por medio de alguna pantalla o medio similar para darnos cuenta tanto de los caracteres que estamos introduciendo, como de lo que está enviando el sistema cuando se encuentra en funcionamiento y también para saber los datos que nos está mandando el reloj de tiempo real, como son el año, mes, día, hora, minutos y segundos. Esto lo logramos por medio de un display que en este caso es de cristal líquido.

#### **4.2. DISPLAY DE CRISTAL LIQUIDO**

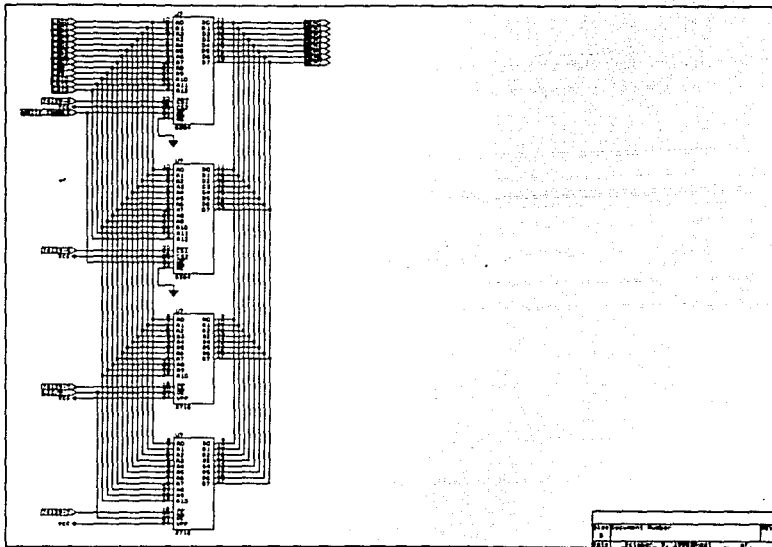
El tipo de display que se adoptó para este sistema fue precisamente el fabricado de cristal líquido. Dado el tamaño del sistema, tenía que ser uno que además de adaptarse a las características de éste en cuanto al despliegue de la información, fuera funcional y nítido en donde quiera que el sistema fuese instalado. De entre los tipos existentes en el mercado, se escogió el que pudiera satisfacer las necesidades de desplegar a un mismo tiempo los parámetros de tiempo y los valores de las muestras de conversión.

Este display cuenta con un despliegue de 8 dígitos en código de 7

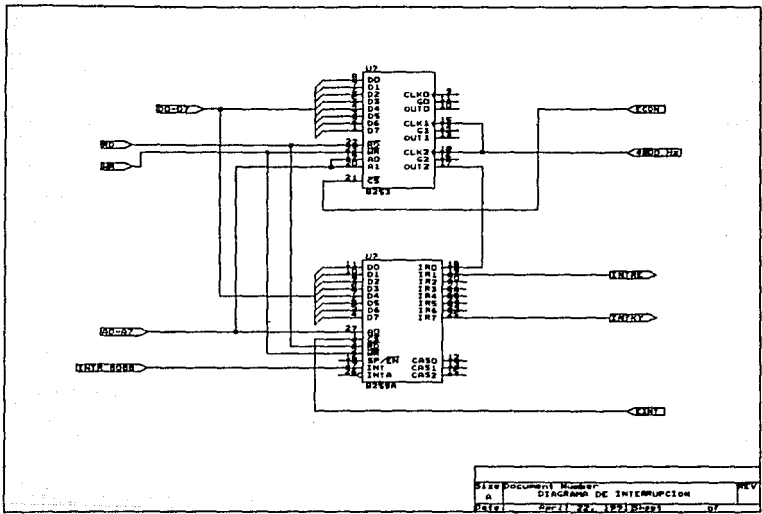


File Document Number	REV
A	
Date	October 8, 1978
By	ST

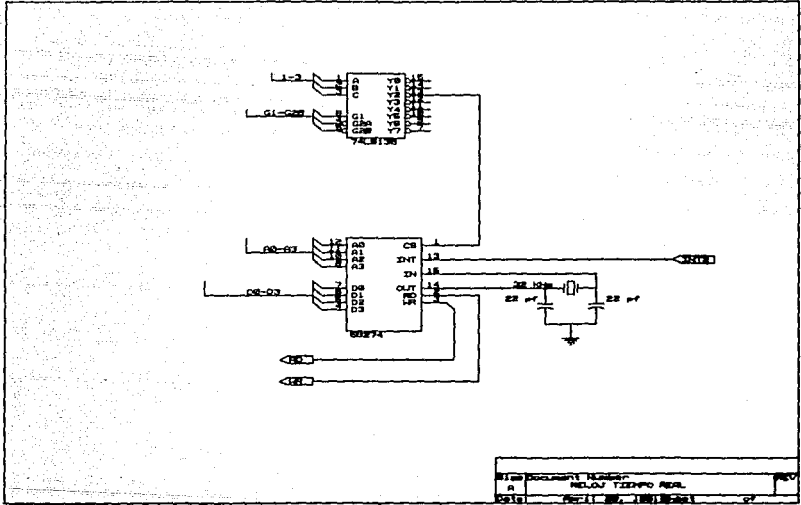


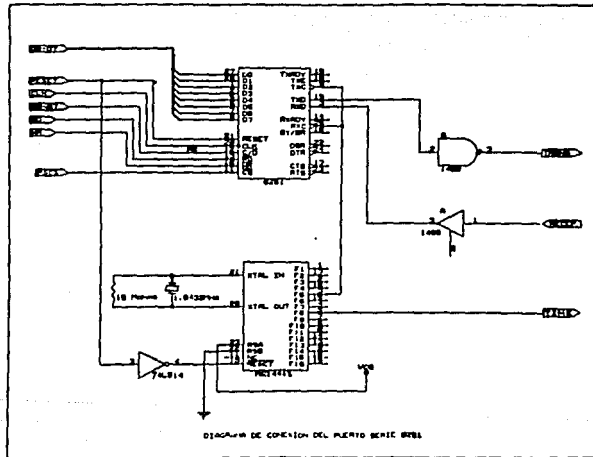






SIN	DOCUMENT NUMBER	REV
A	DIAGRAMA DE INTERRUPCION	
FECH:	Abril 22, 1991	07









segmentos, además contiene el signo (:). La configuración de cada caracter se hace generalmente por software. Es de hacer notar que para poder manejar este tipo de display es necesario el uso de circuitos especiales. Tanto para la función de desplazar dígitos a través del display, como el control de caracteres. Para poder satisfacer estas necesidades, se utilizó el circuito manejador de display ICM7211 de Intersil que a continuación se describirá.

#### 4.2.1. Manejador de Display ICM7211

Es un circuito integrado construido con técnica CMOS y pertenece a la familia de Decodificadores/Manejadores de display no multiplexados de 4 dígitos y 7 segmentos. Estos dispositivos están construidos para manejar displays convencionales LCD (Liquid Cristal Display, Display de Cristal Líquido) cuenta con un oscilador completo RC, un divisor de cadena, un decodificador de señales y 28 segmentos de salida.

Otra característica de este dispositivo es que puede manejar tanto entradas multiplexadas como entradas de cualquier configuración de microprocesador. La versión para microprocesador contiene entrada de datos "latcheada" y latches de dirección digital bajo el control de entradas selectoras de circuito de alta velocidad.

El dispositivo simplifica la tarea de implementar a un costo efectivo, un display alfanumérico de 7 segmentos, sin requerir ROM o tiempo de CPU para decodificar y desplegar caracteres. El 7211 se encuentra disponible en 2 versiones, Tipo "A" y Tipo "B". Para nuestros



fines, se escogio el Tipo "A" por contener todos los caracteres utilizados en el código hexadecimal. En la FIG. 4.2. se muestra el diagrama de bloques del ICM7211MIPL.

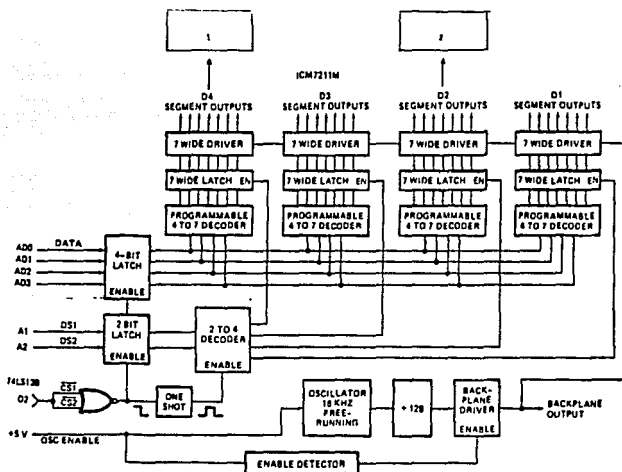


FIG. 4.2. Diagrama de Bloques del Manejador de Display ICM7211.

### 4.3. UNIDAD DE CINTA

Como se habla mencionado anteriormente, la forma en la cual se almacenan los datos provenientes del Sistema Adquisidor de Datos, es por medio de un cassette, el cual contiene en su interior, una cinta en la cual son grabados los caracteres. A continuación se explica el proceso de grabación de datos.

#### 4.3.1. Cinta Magnética

La cinta magnética es una tira o cinta de material plástico recubierto de una película de material ferromagnético (óxido de hierro), lo que permite que sea grabada en forma continua sobre toda una superficie. Para indicar la longitud utilizable de la cinta se le colocan al inicio y al final marcas reflectantes que pueden ser detectadas por la unidad de cinta.

La cinta adecuada para la grabación de datos del sistema adquisidor, es como las que se utilizan normalmente para grabación de audio con una grabadora normal. Este tipo de cinta cuenta con una longitud de 90 ó 135 metros, dependiendo de la duración deseada, y con un ancho aproximado de 0.5 cm. Es importante diferenciar el medio de almacenamiento (cinta magnética), del dispositivo que permite operarlas (lectora-grabadora de cinta magnética), ya que de ésta dependen las distintas modalidades de su utilización.

Específicamente, podemos hablar de 4 características en la unidad de

cinta :

- Densidad de Grabación.- Se refiere a la cantidad de información que puede ser almacenada en la cinta por unidad de longitud y que va de 800 BPI (bytes por pulgada) a 6250 BPI. Suponiendo que las características intrínsecas del material ferromagnético lo permitan, la densidad de grabación dependerá del dispositivo utilizado.

- Número de Pistas.- Se refiere al número de bits que pueden grabarse en forma perpendicular a la longitud de la cinta, pudiendo ser de 7 ó 9 pistas en cintas profesionales, en nuestro caso se tiene 1 pista.

- Velocidades de Lectura o Escritura.- Se refiere a la velocidad con la que pueden leerse o escribirse los datos en una cinta y se da en pulgadas por segundo (ln/seg).

- Velocidad de transferencia de datos.- Esta es una característica importante, ya que permite conocer la velocidad máxima a la que puede moverse la información en el dispositivo. Está relacionada con la velocidad de lectura o escritura y con la densidad de grabación usada. Por ejemplo, si la velocidad de lectura o escritura es de 45 ln/seg y la densidad de grabación de 1600 BPI, se tendrá una velocidad de transferencia de datos de 72000 Bytes/segundo.

La lectura o escritura se lleva a cabo al pasar la cinta bajo una serie de bobinas (colocadas en forma perpendicular a la cinta) que detectan y magnetizan la película de óxido de hierro al deslizarse. Estas

bobinas se agrupan en lo que se llama una cabeza de lectura o escritura, y precisamente del número de bobinas dependerá el número de pistas de grabación. Es muy frecuente que se utilice una de las pistas para grabar un bit de paridad. Este bit de paridad que forma parte de cada byte almacenado podrá tomar los valores 0 ó 1, de tal forma que el número total de bits prendidos en el carácter sea par (paridad par) o impar (paridad impar).

Existen diferentes formas de controlar el movimiento de la cinta, pero en la mayoría de los casos es controlada por servomecanismos de columnas de vacío pudiéndose conocer dónde principia la cinta y dónde termina por las marcas reflectantes de comienzo de cinta y fin de cinta. El control de la unidad de cinta, generalmente se lleva a cabo en forma manual.

Debemos considerar que, para leer o escribir en la cinta, esta debe iniciar el movimiento y alcanzar su velocidad nominal, lo que requiere de un cierto tiempo. De la misma forma, cuando se deja de leer o escribir en la cinta, se requiere un cierto tiempo para que ésta se detenga. Debido a ello, entre grupos de caracteres existe un espacio que no puede ser utilizado, llamado espacio entre registros (Inter Record Gap: IRG), el cual mide de 0.5 cm a 1 cm aproximadamente.

Al grupo de caracteres entre dos IRG, se le llama registro físico o bloque y su tamaño es un factor muy importante, ya que el número de caracteres almacenados en la cinta disminuye al aumentar el espacio utilizado por los IRG. En la FIG. 4.3. se ilustra este fenómeno.



longitud de 90 m, el número de caracteres provenientes del Sistema de Adquisición de Datos que pueden ser almacenados es de 248,047, equivalente a aproximadamente 243 kbytes.

Esta cantidad permitida de caracteres es suficiente para el muestreo de datos en la mayoría de los sistemas físicos que lo requieren, no olvidando que el Sistema de Adquisición de Datos cuenta con 16 kbytes de memoria RAM en su tarjeta, si únicamente se desea muestrear una cantidad de datos no muy grande y no haya necesidad de utilizar la unidad de cinta.

## CAPITULO V

### PROGRAMAS DEL SISTEMA DE ADQUISICION DE DATOS

## CAPITULO V

### PROGRAMAS DEL SISTEMA DE ADQUISICION DE DATOS

En este capítulo se explican los programas que forman parte del sistema. En los diagramas de flujo se encuentran separadas en forma modular las etapas que conforman el Programa de Funcionamiento General del Sistema de Adquisición de Datos.

Este programa se divide en 4 partes:

1. Programa Monitor del Sistema
2. Programa de Adquisición de Datos
3. Programa de Almacenamiento de Datos
4. Programa de Reloj de Tiempo Real

Los algoritmos se codifican en lenguaje ensamblador del microprocesador 8088. Posteriormente se ensambla el programa y se obtiene el código objeto que se almacena en una memoria EPROM 2716, por medio de la cual le son asignadas a cada dispositivo las funciones a desempeñar dentro del sistema.

#### 5.1. PROGRAMA MONITOR

Este programa es el encargado de controlar el funcionamiento total



del sistema, es algo así como su sistema operativo por tanto se encarga de supervisar a los demás programas, así como de coordinar el trabajo resultante de las otras partes. Es desde aquí de donde se puede abordar un módulo en especial del sistema.

Una vez que se enciende el sistema, se configuran los dispositivos que intervienen en cada una de las etapas de este programa. Se configura el puerto serie 8251 y los puertos paralelos 8255-1 y 8255-2. Ya realizado este paso inicial, se pone en funcionamiento la subrutina de Eco de Teclado, por medio de la cual se pregunta mediante 3 opciones qué función se desea llevar a cabo. Las diferentes opciones son:

1. Adquisición de Datos
2. Actualizar Reloj de Tiempo Real
3. Salir del Sistema

## 5.2. RUTINA DE ADQUISICION DE DATOS

Una vez escogida la opción, el programa va a la subrutina deseada. Si se elige la primera opción, el programa se traslada a un menú en donde se preguntan los siguientes parámetros:

- a) Número de muestras a tomar
- b) Número de canales a utilizar
- c) Tiempo de muestreo

Dependiendo del evento a medir, se pueden introducir los parámetros deseados. Si se quieren dejar los ya establecidos simplemente se oprime la Tecla <<Enter>> del teclado. Los valores se dan en forma decimal. Una vez establecidos estos parámetros, empieza a correr la Rutina de Adquisición de Datos.

El primer paso de esta rutina es configurar el 8253 que, por hardware, se configura con una señal de entrada de 4800 Hz. para dar, por medio de software, una salida de 1000 Hz.. Esto significa que el tiempo de muestreo por software se encuentra en un intervalo que va de 1000 Hz. a 1 Hz. Si se quiere expandir este intervalo a tiempos mucho más pequeños de muestreo, entonces se debe cambiar la señal de entrada del temporizador programable por una de 9600 Hz. si se quiere del orden de microsegundos. Esta salida de frecuencia puede ir cambiando, según las necesidades del programa.

A continuación se configura el Controlador de Interrupciones 8259. Las interrupciones que han sido establecidas en el programa son 3, las cuales van a cada una de las respectivas terminales de este circuito:

IR0: Temporizador Programable 8253

IR1: Reloj de Tiempo Real

IR7: Teclado

El orden en el cual se han establecido las interrupciones, ha sido conforme a su importancia. Esto quiere decir que mientras se encuentre funcionando el temporizador, no puede ser interrumpido en su operación,

ni por el reloj de tiempo real ni por el teclado. Lo mismo ocurre con el reloj de tiempo real, su funcionamiento no puede ser interrumpido por el teclado, pero sí puede hacerlo el reloj de tiempo real 8253. Este orden fue establecido con base en el trabajo que realiza cada dispositivo, por ejemplo, mientras se encuentre trabajando la etapa de la conversión A/D, no puede efectuarse otra función dentro del sistema porque crearía problemas tanto de continuidad como de funcionamiento de dicha etapa. El tiempo en el cual se puede interrumpir al reloj de tiempo real es de 0.1 seg. Posteriormente se explicará el porqué de esto.

Es importante mencionar que al inicio de la conversión el programa inicia un contador de 5 seg. para que se reciban datos. Si durante este lapso llega un dato, el programa efectúa la conversión, de lo contrario, aparece en el display el letrero " BAC ", indicando que no hay datos. Respecto a cómo se puede encontrar los datos provenientes de un canal en particular en la memoria de almacenamiento, se tiene la siguiente expresión:

$$\text{Posición} = L + 4 ( N - 1 ) \quad (5.1)$$

donde:

L = localidad de memoria inicial, y

N = número de canal

Entonces, la interrupción de mayor importancia es la del Temporizador 8253 porque allí reside el segmento base del sistema que es la conversión A/D. La segunda interrupción en importancia es la del Reloj de Tiempo Real, la cual estará enviando al display los parámetros durante

cada intervalo de tiempo.

La tercera interrupción será la de teclado. Esto se debe a que si en un determinado momento se desea interrumpir alguna de las etapas del programa, deben llevarse a cabo las rutinas antes mencionadas. Cuando concluyen estas rutinas, entonces entra la interrupción de teclado, la cual comprende las funciones ejecutables del sistema.

El proceso es el siguiente: cuando ocurre la primera interrupción, se activa el PIC 8259 provocando una pausa de 100 mseg. Durante esta pausa se lleva a cabo la conversión A/D. El Puerto paralelo 8255-3 pregunta si hay algún dato presente, si no lo hay, espera hasta que se encuentre alguno. Si lo hay, manda una señal por medio de una terminal del puerto B, indicando que se cierre el relevador, iniciando así la rutina de conversión. En forma conjunta a esta acción se selecciona el primer canal por medio del mismo puerto B, activando las terminales correspondientes para configurar los multiplexores de canal 14051. El 14433 realiza la conversión por medio de las terminales DS1 a DS4 y selecciona el dígito a mandar al display por medio de las terminales de salida Q0 a Q3. Una vez concluida la conversión A/D, el 14433 manda la señal de EOC (fin de conversión) a una de las terminales del 8255-3, indicando que se desactive la señal del relevador.

El 8088 almacena la conversión de 3 1/2 dígitos en un buffer, reservando 4 bytes de memoria para tales valores. Esto quiere decir que cada dígito resultante de la conversión se guarda en un espacio de 8 bits en forma BCD.

A continuación se envía al display lo que haya en el buffer por medio de software, utilizando un contador de rotación de números.

### 5.3. RUTINA DE ALMACENAMIENTO DE DATOS

Una vez exhibidos estos datos en display, el programa pasa a otra rutina. Los valores localizados en el buffer son procesados a través de una rutina de conversión de BCD a binario con el fin de almacenarlos en memoria RAM. El programa verifica, por medio de indicadores de tabla, si hay espacio de 4 bytes en memoria. Si hay la cantidad requerida, entonces procede a almacenar los datos. Si no hay memoria suficiente para estos valores, entonces el programa recurre a la rutina del Puerto Serie 8251 enviando los datos a cinta utilizando la norma RS-232C.

Si se quiere almacenar estos datos en un drive de una computadora, entonces se conectan las señales de la RS-232C a los de la computadora. Esto es algo opcional ya que queda fuera del alcance de este trabajo el proveer tal interfaz, pues puede ser necesario dejar el sistema diseñado en el sitio de interés durante mucho tiempo. Se incluye dicha opción para hacer el sistema más versátil y darle un mayor espectro de aplicaciones, encontrándose a tono con los dispositivos actuales de procesamiento de datos. En la sección de algoritmos y de listado de programas se encuentra la rutina por medio de la cual el sistema se conecta con una PC, indicándose el formato de la transmisión.

Cuando el programa termina de almacenar los datos, regresa al punto

en donde se define el número de canales y pregunta el número de canales a utilizar por medio de la variable CHAN. Si queda algún canal pendiente, se repite toda la rutina anterior de conversión para ese canal en particular. Si todos los canales han sido procesados entonces el programa envía la palabra de control EOI al PIC 8259, indicando que ha finalizado la interrupción. Concluida ésta, el programa regresa al menú inicial.

#### 5.4. RUTINA DE ACTUALIZACION DEL RELOJ DE TIEMPO REAL

En caso de haber elegido la opción de actualizar el reloj de tiempo real, el programa entra en la rutina de tiempo real. Desplegará en el display el año, mes y día, la hora, minutos y segundos utilizando para ello la tecla <<ENTER>>. Si se está de acuerdo con los parámetros exhibidos se debe oprimirla tecla <<NEXT>> regresando al menú principal. Si se quieren cambiar los parámetros del reloj de tiempo real, el programa corre una rutina en la cual pide uno a uno los datos de tiempo. En primer lugar pide el año, dando únicamente las dos últimas cifras, por ejemplo para 1990, únicamente escribir 90. Para indicar que el año es bisiesto, se carga el registro correspondiente al año, con la palabra de control que configura el dato. Para indicar el mes, el valor va desde 01 a 12. Para introducir el día, se teclea el dato de 01 a 07, teniendo la opción de numerar los días conforme al orden deseado. Por ejemplo, si se quiere que el miércoles sea el día 01 y no el lunes, se puede indicar así. Los valores elegidos aparecerán en el display mientras se encuentre funcionando el sistema.

Una vez definidos los parámetros de tiempo, el programa regresa al menú principal en donde se escogerá la función a utilizar. Si se desea salir del sistema, simplemente se oprime la Tecla <<3>>.

Cada vez que se enciende el sistema es necesario que se programe el reloj. La causa reside en que se desconecta la alimentación de la tarjeta, perdiéndose la programación del mismo. Este problema fue contemplado dentro del diseño del sistema adquisidor, y hay algunas alternativas para resolverlo, dependiendo de las características del sitio en donde se instale.

Una alternativa reside en que el sistema sea alimentado por una batería de automóvil o acumulador. Esta es una de las formas más eficientes de resolver este problema, tanto desde el punto de vista práctico como económico y funcional dadas las características de tamaño y manejo del acumulador. Un ejemplo de esto puede ser el lugar en el cual son colocados los acelerógrafos, que usualmente son las zonas más altas de los cerros y montes.

## CAPITULO VI

### RESULTADOS Y CONCLUSIONES



## CAPITULO VI

### RESULTADOS Y CONCLUSIONES

#### 6.1 RESULTADOS

A continuación se muestran los resultados obtenidos al realizar pruebas con el Sistema de Adquisición de Datos. Las pruebas consistieron en la adquisición de datos provenientes de de 3 acelerómetros. Es decir, la variable física a muestrear es la aceleración. Se escogió el uso de acelerómetros por la importancia que representa para toda la población el que se tenga un conocimiento del comportamiento de estructuras ante el movimiento provocado por un fenómeno sísmico.

Una vez obtenidos los datos, se utilizó un programa para procesar las señales y presentarlas de manera conveniente. Este programa es capaz de muestrear 3 diferentes canales, mostrando cada una de estas señales de manera gráfica. Por medio del mismo se puede conocer el valor de la señal en cada muestreo, para cada uno de los canales, el tiempo en el cual se obtuvieron los valores, unidades de acotación tanto en el eje X como en el eje Y, el Factor de expansión, el Factor de Diezmación, el segmento con el comportamiento total del Acelerómetro en el intervalo de tiempo deseado, el número de muestra y el valor de ganancia escogido.

La manera en la cual se acopló el Sistema de Adquisición a la Computadora fue por medio del Puerto Serie a través de la Norma RS-232,

modificando el programa original en su asignación de periféricos.

Cada acelerómetro se conectó al Amplificador de Instrumentación para, después de acondicionarse la señal, pasar a la tarjeta de Adquisición. El número de muestras programadas fue de 400 por canal, listando 200 para visualizar los valores convertidos de forma analógica a digital en el formato de cuentas, que es el formato por medio del cual se relacionan los voltajes obtenidos con respecto a los valores dados por el acelerómetro. A continuación se dan los parámetros de funcionamiento usados para este muestreo de datos.

Número de Canales:	3
Tiempo de Muestreo:	100 mseg
Número de muestras:	200 por canal
Ganancia:	50
Escala eje X:	3.92 gals/div
Escala eje Y:	2.5 seg/div
Factor de Diezmación:	10
Factor de Expansión:	1
Tiempo de Origen Universal	

El número de canales es el utilizado para procesos sísmicos. Canal 1 para componente horizontal, Canal 2 para componente vertical y Canal 3 para componente transversal.

La unidad de medición usada en este tipo de procesos sísmicos es el gal. Su equivalencia es:

$$1 \text{ gal} = 1 \text{ cm/seg}^2,$$

Es necesario conocer la equivalencia de voltaje con respecto al valor en BCD para un acelerómetro para poder interpretar los resultados. Los acelerómetros disponibles manejan un margen de voltaje de -15 V a +15 V. La señal se acondiciona para que a la salida se tengan valores que no excedan el límite máximo de conversión que maneja el conversor A/D que en este caso es de 1.999 Volts., de lo contrario se tendrá un error de sobrerango en el conversor y las mediciones no serán confiables.

Los sensores que se utilizaron en las pruebas fueron acelerómetros lineales Systron Donner Modelo 4310A-1-A que tienen un intervalo de trabajo de  $\pm 1 \text{ G}$ .

En código BCD se tiene la siguiente asignación para el caso de las mediciones efectuadas:

Valor para 2 G (2000 gals)	4096
Valor para 0.0 G (0 gals)	2048
Valor para -2 G (-2000 gals)	0000

De acuerdo con estos datos, se listan 200 muestras de cada canal correspondientes a cada uno de los 3 acelerómetros. Luego se incluyen las gráficas de los resultados para cada acelerómetro (en las gráficas se incluyen más puntos que en listado de datos). Dada la gran cantidad de muestras que pueden ser adquiridas fue necesario escoger segmentos representativos para ilustrar el comportamiento del sistema.

RESULTADOS DE PRUEBAS CON 3 ACELEROMETROS SYSTRON DONNER MOD. 4310A-1-A.

MUESTRA No.	CANAL 1 VALOR BCD	CANAL 1 [gals]	CANAL 2 VALOR BCD	CANAL 2 [gals]	CANAL 3 VALOR BCD	CANAL 3 [gals]
1	2103	1.08	1999	-0.96	2047	-0.02
2	2098	0.98	2001	-0.92	2045	-0.06
3	2075	0.52	2003	-0.88	2043	-0.10
4	2066	0.36	2012	-0.70	2039	-0.18
5	2059	0.21	2022	-0.51	2033	-0.30
6	2053	0.10	2028	-0.39	2023	-0.49
7	2049	0.02	2033	-0.30	2014	-0.67
8	2048	0.00	2035	-0.26	2007	-0.81
9	2046	-0.04	2037	-0.22	1998	-0.97
10	2042	-0.12	2042	-0.12	1995	-1.03
11	2037	-0.22	2046	-0.03	1989	-1.16
12	2031	-0.33	2048	0.00	1983	-1.27
13	2023	-0.49	2050	0.04	1976	-1.41
14	2017	-0.60	2053	0.09	1968	-1.57
15	2010	-0.74	2054	0.11	1960	-1.72
16	2002	-0.89	2058	0.19	1953	-1.86
17	1998	-0.97	2059	0.21	1946	-1.99
18	1995	-1.04	2061	0.26	1944	-2.04
19	1986	-1.21	2064	0.31	1938	-2.15
20	1977	-1.39	2068	0.39	1942	-2.07
21	1972	-1.49	2074	0.51	1945	-2.01
22	1966	-1.60	2077	0.57	1950	-1.91
23	1957	-1.77	2084	0.71	1954	-1.83
24	1965	-1.63	2088	0.79	1958	-1.76
25	1971	-1.51	2091	0.85	1971	-1.51
26	1978	-1.37	2099	0.99	1981	-1.30
27	1987	-1.20	2103	1.07	1990	-1.14
28	1993	-1.07	2108	1.17	1996	-1.01
29	1998	-0.97	2114	1.30	2003	-0.88
30	2007	-0.81	2116	1.33	2013	-0.69
31	2013	-0.69	2120	1.41	2021	-0.53
32	2021	-0.53	2124	1.49	2027	-0.42
33	2027	-0.41	2128	1.57	2035	-0.25
34	2033	-0.29	2134	1.69	2041	-0.13
35	2039	-0.18	2139	1.77	2045	-0.05
36	2044	-0.08	2141	1.82	2048	0.00
37	2048	0.00	2145	1.90	2052	0.08
38	2053	0.09	2150	1.99	2058	0.20
39	2058	0.19	2151	2.02	2065	0.34
40	2061	0.25	2153	2.06	2078	0.59
41	2064	0.31	2158	2.16	2086	0.74
42	2072	0.47	2162	2.23	2061	0.25
43	2079	0.61	2166	2.31	2099	1.00
44	2087	0.77	2168	2.35	2106	1.13
45	2094	0.89	2176	2.50	2113	1.28
46	2099	0.99	2180	2.59	2123	1.47
47	2103	1.07	2183	2.63	2130	1.61
48	2109	1.19	2190	2.77	2140	1.79
49	2113	1.27	2197	2.91	2148	1.96
50	2117	1.35	2200	2.97	2155	2.09

RESULTADOS DE PRUEBAS CON 3 ACELEROMETROS SYSTRON DONNER MOD.4310A-1-A.

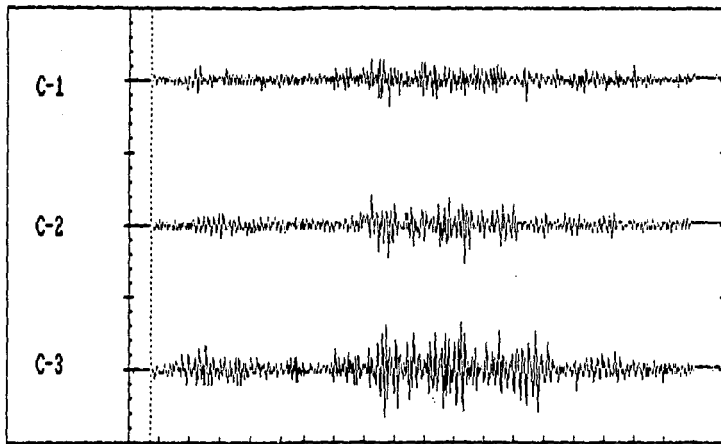
MUESTRA No.	CANAL 1 VALOR BCD	CANAL 1 [gals]	CANAL 2 VALOR BCD	CANAL 2 [gals]	CANAL 3 VALOR BCD	CANAL 3 [gals]
51	2123	1.47	2203	3.03	2150	2.00
52	2129	1.59	2210	3.16	2141	1.81
53	2136	1.72	2218	3.33	2133	1.67
54	2145	1.89	2213	3.22	2121	1.42
55	2150	1.99	2209	3.14	2111	1.23
56	2154	2.07	2206	3.08	2104	1.09
57	2160	2.19	2203	3.03	2099	0.99
58	2165	2.28	2196	2.90	2089	0.80
59	2169	2.37	2192	2.81	2081	0.64
60	2175	2.49	2188	2.74	2075	0.53
61	2177	2.53	2183	2.64	2069	0.42
62	2185	2.67	2180	2.58	2063	0.29
63	2191	2.79	2176	2.51	2057	0.18
64	2196	2.89	2172	2.43	2050	0.04
65	2189	2.75	2169	2.37	2048	0.00
66	2182	2.63	2166	2.31	2052	0.08
67	2177	2.52	2160	2.19	2057	0.18
68	2171	2.41	2158	2.16	2063	0.29
69	2165	2.29	2151	2.01	2064	0.31
70	2156	2.11	2146	1.91	2059	0.22
71	2151	2.02	2140	1.80	2055	0.14
72	2145	1.89	2131	1.63	2048	0.01
73	2135	1.71	2124	1.49	2048	0.00
74	2130	1.61	2115	1.31	2046	-0.04
75	2126	1.52	2107	1.16	2037	-0.22
76	2119	1.39	2101	1.04	2036	-0.23
77	2110	1.22	2094	0.90	2022	-0.51
78	2106	1.13	2082	0.67	2013	-0.69
79	2103	1.07	2076	0.55	2007	-0.81
80	2097	0.96	2073	0.48	1999	-0.96
81	2089	0.81	2072	0.46	1993	-1.08
82	2082	0.67	2069	0.42	1986	-1.22
83	2071	0.46	2067	0.37	1976	-1.41
84	2059	0.21	2064	0.32	1982	-1.29
85	2052	0.08	2062	0.27	1991	-1.11
86	2047	-0.02	2059	0.21	1997	-0.99
87	2042	-0.11	2056	0.16	2007	-0.80
88	2036	-0.23	2052	0.08	2014	-0.67
89	2032	-0.31	2050	0.03	2021	-0.53
90	2023	-0.48	2048	0.00	2023	-0.48
91	2017	-0.61	2047	-0.02	2029	-0.37
92	2011	-0.73	2045	-0.06	2038	-0.20
93	2007	-0.81	2042	-0.11	2045	-0.06
94	1998	-0.97	2039	-0.17	2048	0.00
95	1993	-1.07	2035	-0.26	2053	0.10
96	1989	-1.16	2029	-0.37	2059	0.21
97	1981	-1.30	2022	-0.51	2068	0.39
98	1973	-1.47	2011	-0.72	2076	0.55
99	1980	-1.33	2004	-0.86	2087	0.76
100	1985	-1.24	2000	-0.93	2090	0.83

RESULTADOS DE PRUEBAS CON 3 ACELEROMETROS SYSTRON DONNER MOD.4310A-1-A.

MUESTRA No.	CANAL 1 VALOR BCD	CANAL 1 [gals]	CANAL 2 VALOR BCD	CANAL 2 [gals]	CANAL 3 VALOR BCD	CANAL 3 [gals]
101	1990	-1.14	1995	-1.04	2095	0.91
102	1995	-1.04	1989	-1.15	2099	0.99
103	1998	-0.97	1983	-1.27	2104	1.09
104	2004	-0.86	1977	-1.39	2110	1.21
105	2009	-0.77	1970	-1.52	2119	1.39
106	2014	-0.66	1961	-1.69	2125	1.51
107	2021	-0.53	1954	-1.83	2139	1.77
108	2027	-0.41	1946	-1.99	2146	1.91
109	2032	-0.32	1943	-2.06	2151	2.02
110	2037	-0.21	1946	-2.00	2158	2.16
111	2041	-0.13	1949	-1.93	2153	2.06
112	2046	-0.03	1956	-1.80	2149	1.97
113	2052	0.07	1965	-1.63	2142	1.83
114	2057	0.18	1972	-1.49	2136	1.72
115	2063	0.29	1980	-1.33	2129	1.59
116	2068	0.39	1986	-1.21	2123	1.47
117	2074	0.51	1991	-1.12	2115	1.31
118	2080	0.63	1996	-1.01	2109	1.20
119	2086	0.75	1998	-0.98	2103	1.08
120	2091	0.85	1999	-0.96	2099	0.99
121	2098	0.97	2002	-0.89	2091	0.84
122	2103	1.08	2009	-0.77	2085	0.72
123	2109	1.19	2013	-0.68	2080	0.62
124	2113	1.27	2019	-0.56	2074	0.51
125	2118	1.36	2022	-0.50	2067	0.37
126	2121	1.43	2029	-0.38	2061	0.26
127	2127	1.55	2036	-0.23	2055	0.14
128	2133	1.67	2040	-0.15	2053	0.09
129	2139	1.78	2044	-0.07	2048	0.00
130	2146	1.91	2048	0.00	2045	-0.06
131	2152	2.03	2051	0.05	2039	-0.17
132	2146	1.91	2055	0.13	2033	-0.29
133	2141	1.82	2057	0.17	2027	-0.42
134	2137	1.74	2061	0.26	2017	-0.60
135	2131	1.63	2064	0.31	2011	-0.73
136	2126	1.52	2062	0.27	2003	-0.88
137	2120	1.41	2058	0.20	1997	-0.99
138	2116	1.32	2056	0.16	1993	-1.08
139	2109	1.20	2054	0.12	1988	-1.17
140	2104	1.09	2053	0.09	1982	-1.29
141	2099	0.99	2050	0.03	1976	-1.41
142	2092	0.87	2048	0.00	1966	-1.60
143	2085	0.73	2048	-0.01	1956	-1.79
144	2080	0.62	2045	-0.06	1953	-1.86
145	2074	0.51	2042	-0.12	1946	-2.00
146	2068	0.39	2038	-0.20	1942	-2.08
147	2061	0.26	2035	-0.26	1937	-2.17
148	2057	0.17	2031	-0.34	1931	-2.29
149	2054	0.11	2022	-0.51	1925	-2.41
150	2048	0.00	2008	-0.79	1912	-2.66

RESULTADOS DE PRUEBAS CON 3 ACELEROMETROS SYSTRON DONNER MOD.4310A-1-A.

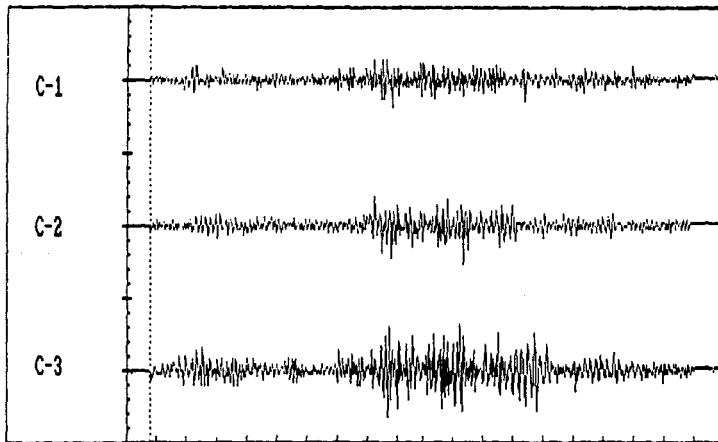
MUESTRA	CANAL 1	CANAL 1	CANAL 2	CANAL 2	CANAL 3	CANAL 3
No.	VALOR BCD	[gals]	VALOR BCD	[gals]	VALOR BCD	[gals]
151	2044	-0.07	2003	-0.87	1904	-2.81
152	2039	-0.18	1999	-0.96	1896	-2.97
153	2034	-0.28	1996	-1.02	1890	-3.09
154	2029	-0.37	1997	-1.00	1883	-3.23
155	2023	-0.48	1999	-0.95	1873	-3.41
156	2018	-0.59	2002	-0.90	1865	-3.57
157	2012	-0.71	2008	-0.78	1859	-3.69
158	2006	-0.83	2017	-0.61	1854	-3.79
159	1999	-0.95	2025	-0.44	1860	-3.67
160	2004	-0.86	2033	-0.30	1868	-3.52
161	2010	-0.74	2038	-0.19	1873	-3.41
162	2016	-0.63	2042	-0.12	1881	-3.27
163	2021	-0.52	2046	-0.04	1887	-3.15
164	2027	-0.41	2048	0.00	1893	-3.03
165	2032	-0.32	2049	0.02	1899	-2.92
166	2037	-0.21	2051	0.06	1904	-2.81
167	2042	-0.12	2055	0.13	1906	-2.77
168	2046	-0.03	2062	0.27	1912	-2.65
169	2051	0.06	2068	0.39	1918	-2.54
170	2056	0.15	2077	0.56	1924	-2.43
171	2060	0.24	2083	0.69	1930	-2.31
172	2065	0.33	2087	0.77	1936	-2.18
173	2069	0.42	2092	0.86	1942	-2.07
174	2075	0.53	2099	0.99	1948	-1.95
175	2081	0.64	2104	1.09	1955	-1.81
176	2086	0.75	2111	1.23	1961	-1.69
177	2091	0.84	2121	1.42	1970	-1.52
178	2098	0.97	2130	1.60	1978	-1.37
179	2103	1.08	2134	1.68	1985	-1.24
180	2109	1.20	2142	1.84	1990	-1.14
181	2115	1.31	2150	1.99	1995	-1.03
182	2121	1.43	2155	2.10	2002	-0.90
183	2127	1.54	2163	2.25	2009	-0.76
184	2132	1.65	2175	2.48	2016	-0.62
185	2138	1.76	2181	2.61	2023	-0.48
186	2143	1.86	2190	2.77	2032	-0.32
187	2148	1.95	2197	2.92	2037	-0.21
188	2152	2.04	2203	3.04	2042	-0.11
189	2157	2.14	2206	3.09	2048	0.00
190	2162	2.22	2212	3.21	2052	0.08
191	2166	2.31	2207	3.10	2059	0.21
192	2172	2.42	2201	3.00	2067	0.37
193	2177	2.53	2195	2.88	2074	0.51
194	2183	2.64	2191	2.80	2083	0.69
195	2176	2.51	2183	2.64	2089	0.80
196	2170	2.39	2177	2.52	2095	0.92
197	2165	2.28	2170	2.39	2101	1.03
198	2159	2.17	2167	2.32	2107	1.15
199	2154	2.08	2161	2.20	2112	1.26
200	2150	1.99	2154	2.08	2117	1.35



M1=	+0.00 gals	Tc: %1665-2:65:11	0.00( 1.00 ) 50.00
M2=	+0.00 gals	Y: 3.92 gals/div	F.D.= 10    SEG#: 1.0    GAN: 50
M3=	+0.00 gals	X: 2.5 seg/div	F.E.= 1    Muestra No.: 0

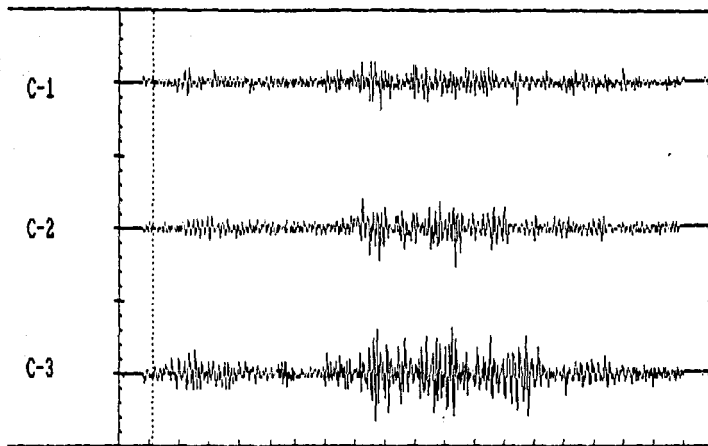
CURSOR: [+ → y Home, PgUp]





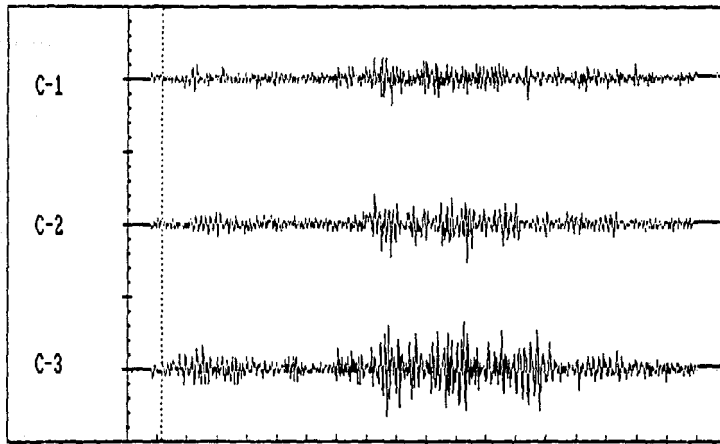
M1=	+1.91 gals	Tc:	14-21:11:42	0.00<	2.00 >	50.00
M2=	-0.96 gals	Y:	3.92 gals/div	F.D.=	10	SEGM: 1.0 GAN: 50
M3=	+0.00 gals	X:	2.5 seg/div	F.E.=	1	Muestra No.: 200

CURSOR: [+ → y Home, PgUp]



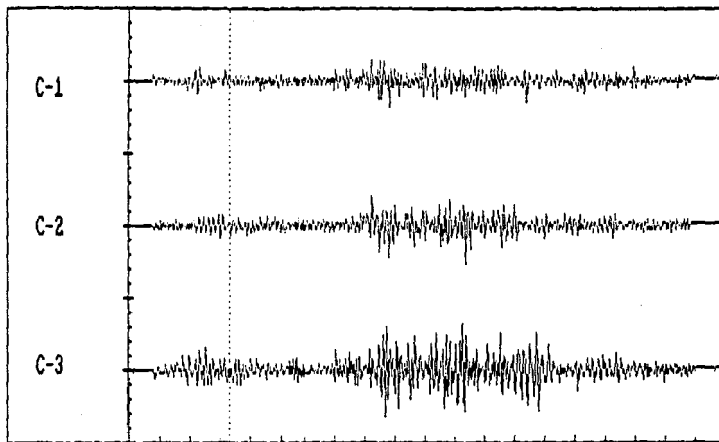
M1=	+0.96 gals	Tc:	14-21:11:43.58	0.00<	2.80	>	50.00
M2=	+0.48 gals	Y:	3.92 gals/div	F.D.=	10	SEGM:	1.0 GAN: 50
M3=	-0.96 gals	X:	2.5 seg/div	F.E.=	1	Muestra No.:	280

URSOR: [← → y Home, PgUp]



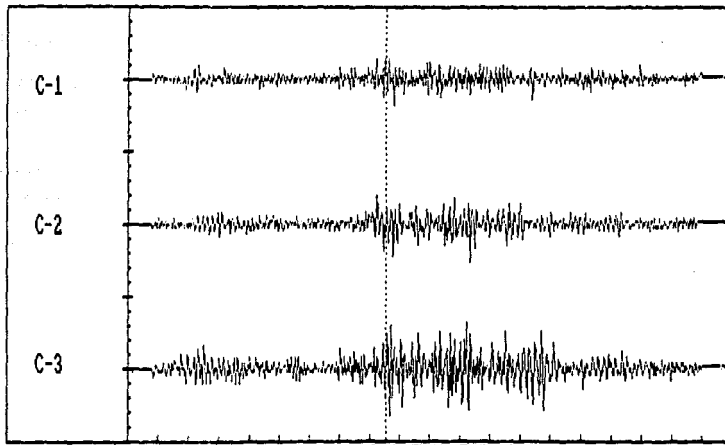
M1=	-0.48 gals	Tc:	14-21:11:43.68	0.00<	2.90 >	50.00
M2=	+0.00 gals	Y:	3.92 gals/div	F.D.=	10	SEGM: 1.0 GAN: 50
M3=	-0.48 gals	X:	2.5 seg/div	F.E.=	1	Muestra No.: 290

CURSOR: [+ → y Home, PgUp]



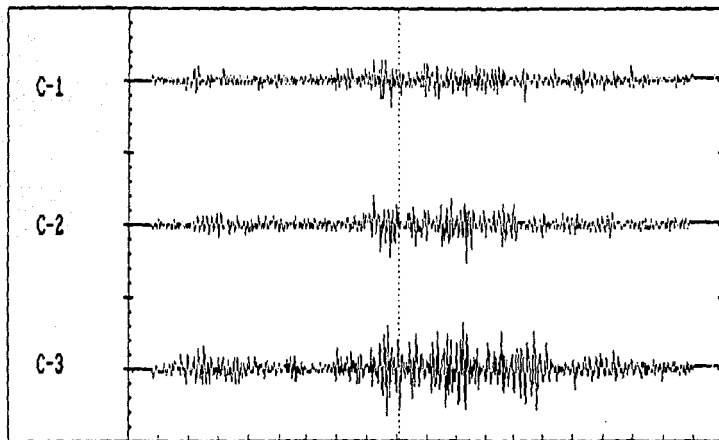
M1=	-0.96 gals	Tc:	14-21:11:49.00	0.00( 8.30 ) 50.00
M2=	+1.91 gals	Y:	3.92 gals/div	F.D.= 10
M3=	+2.39 gals	X:	2.5 seg/div	SEGM: 1.0 GAN: 50
			F.E.= 1	Muestra No.: 830

CURSOR: [+ → y Home, PgUp]



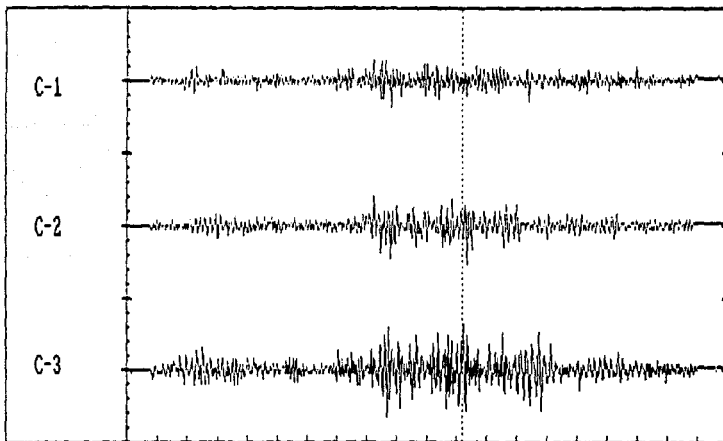
M1=	-4.79 gals	Tc:	14-21:12: 2.18	0.00	( 21.40 )	50.00
M2=	-0.48 gals	Y:	3.92 gals/div	F.D.=	10	SEGM: 1.0 GAN: 50
M3=	-7.18 gals	X:	2.5 seg/div	F.E.=	1	Muestra No.: 2140

CURSOR: [← → y Home, PgUp]



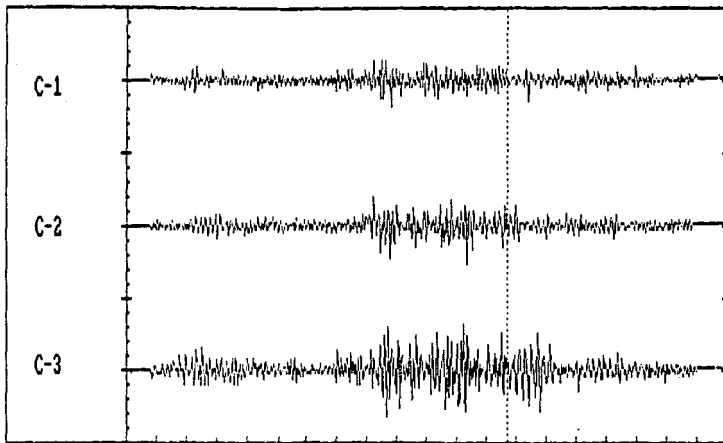
M1=	-0.48 gals	Tc:	14-21:12: 3.48	0.00<	22.70	>	50.00
M2=	-2.39 gals	Y:	3.92 gals/div	F.D.=	10	SEGM:	1.0 GAN: 50
M3=	+5.26 gals	X:	2.5 seg/div	F.E.=	1	Muestra No.:	2270

CURSOR: [← → y Home, PgUp]



M1=	+2.87 gals	Tc:	14-21:12: 8.78	0.00< 28.00 > 50.00
M2=	-2.39 gals	Y:	3.92 gals/div	F.D.= 10
M3=	-1.44 gals	X:	2.5 seg/div	SEGN: 1.0 CAN: 50
			F.E.= 1	Muestra No.: 2800

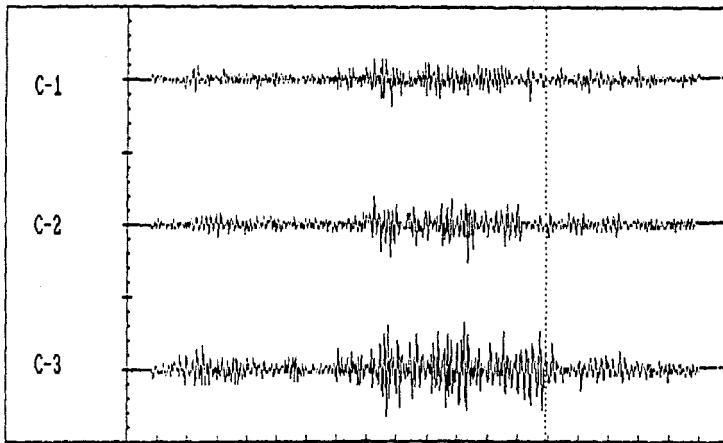
CURSOR: [+ → y Home, PgUp]



M1=	+0.96 gals	Tc:	14-21:12:12.58	0.00<	31.00	>	50.00
M2=	-3.83 gals	Y:	3.92 gals/div	F.D.=	10	SEGM:	1.0 GAN: 50
M3=	-2.39 gals	X:	2.5 seg/div	F.E.=	1	Muestra No.:	3180

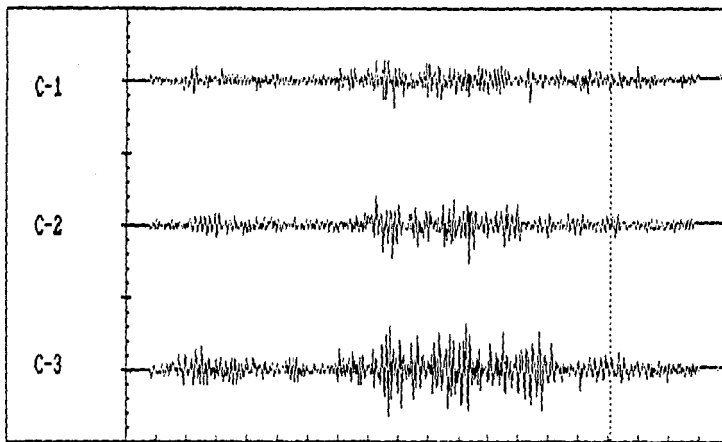
CURSOR: [← → y Home, PgUp]





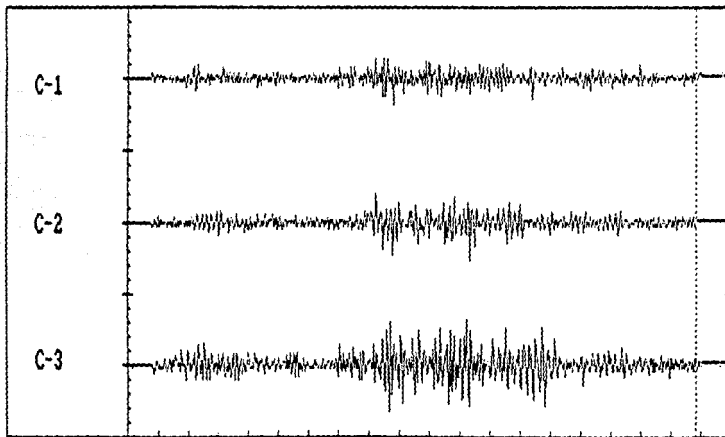
M1=	+1.44 gals	Tc:	14-21:12:15.58	0.00<	34.80	>	50.00
M2=	+1.91 gals	Y:	3.92 gals/div	F.D.=	10	SEGM:	1.0 GAN: 50
M3=	+3.83 gals	X:	2.5 seg/div	F.E.=	1	Muestra No.:	3480

CURSOR: [← → y Home, PgUp]



M1=	+0.00 gals	Tc:	14-21:12:20.98		0.00 < 40.20 > 50.00
M2=	-1.91 gals	Y:	3.92 gals/div	F.D.= 10	SEGM: 1.0 CAN: 50
M3=	-1.44 gals	X:	2.5 seg/div	F.E.= 1	Muestra No.: 4020

CURSOR: [+ → y Home, PgUp]



M1=	+0.00 gals	Tc: %1665-%151:25:22.88	0.00( 47.10 > 50.00
M2=	+0.00 gals	Y: 3.92 gals/div	F.D.= 10
M3=	+0.00 gals	X: 2.5 seg/div	F.E.= 1
			SEGM: 1.0 GAN: 50
			Muestra No.: 4710

CURSOR: [← → y Home, PgUp]

## 6.2. CONCLUSIONES Y RECOMENDACIONES

El presente trabajo tiene como objetivo el Diseño de un Sistema de Adquisición de Datos utilizando el Microprocesador 8088, por lo tanto, los datos provenientes del sistema físico a estudiar deben reunir algunos requisitos previos para poder ser analizados correctamente.

1.- Los valores provenientes del sensor utilizado ya sea termopar, acelerómetro, celda de presión, etc. deben de ser acondicionados por medio del Amplificador de Instrumentación.

2.- Los datos estarán disponibles en un formato de 16 bits para su procesamiento por medio de un sistema que puede ser una PC.

En cuanto al desarrollo de los programas, se enfocó de tal manera que pudiera acoplarse tanto al mayor número de procesos físicos como a las circunstancias en que se presenten éstos. Para ello se cuenta con 24 canales de adquisición, cantidad que puede ser suficiente para la mayoría de estos sistemas, y un tiempo de muestreo variable, característica que le da un grado mayor de versatilidad.

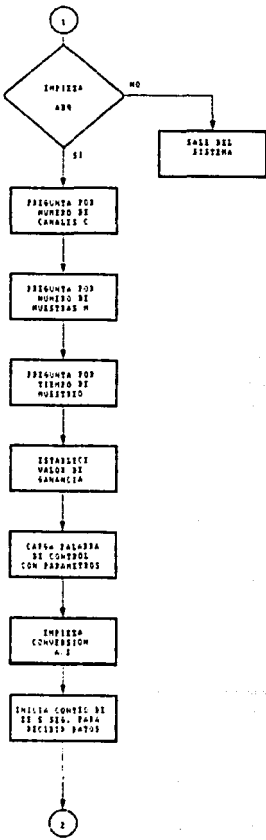
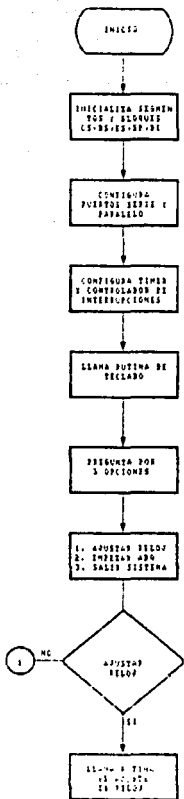
El sistema se diseñó pensando en su posible comercialización y por esta razón se pueden notar algunas características que pueden parecer excesivas como el hecho de tener 3 circuitos de puerto paralelo 8255. Se pensó que en una etapa posterior se podría cambiar la arquitectura de microprocesador a una arquitectura con microcontrolador, lo cual es recomendable para ahorrar tanto tiempo como espacio por el tamaño físico

del sistema pues con muy pocos componentes se puede tener un sistema equivalente al aquí presentado. Actualmente hay disponibles microcontroladores incluso con varios canales de conversores A/D integrados como 68HC11 de Motorola y el 8751 de Intel por ejemplo, que permitirían el desarrollo de un sistema similar con un costo bastante inferior.

Entonces, el Sistema de Adquisición de Datos presentado aquí cumple con los objetivos propuestos, logrando el muestreo de datos en forma automática y confiable, dejando, gracias a las características de expansión de la tarjeta, la posibilidad de realizar modificaciones o mejoras a este mismo sistema, o si se desea, tomarlo como base para la realización de un sistema similar utilizando otro procesador o bien utilizando un microcontrolador.

## APENDICE A

DIAGRAMA DE FLUJO DEL SISTEMA



CONTINUACION DE DIAGRAMA DE FLUJO DEL SISTEMA

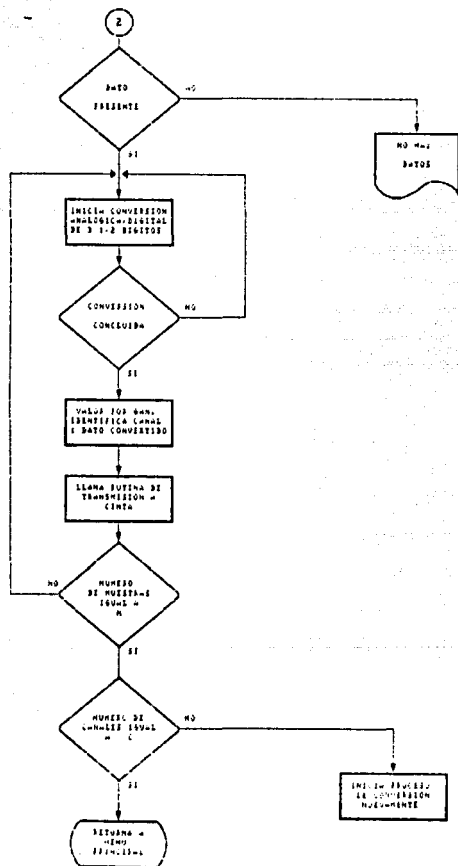




DIAGRAMA DE FLECHAS DE MENÚ DE TIPO 3110

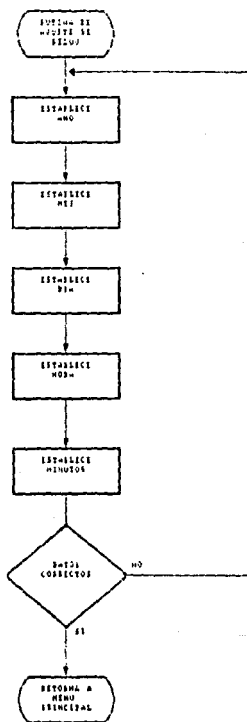
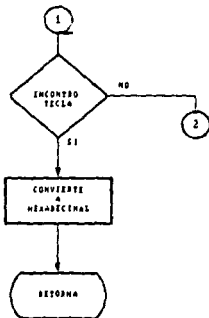
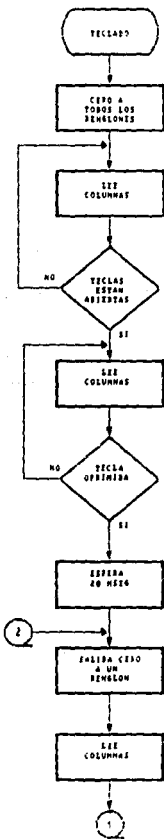


DIAGRAMA DE FLUJO DE TECLADO



## PROGRAMA TERMINAL DEL SISTEMA DE ADQUISICION DE DATOS

REALIZADO EN EL LENGUAJE ENSAMBLADOR DEL 8088

```

MINT_TABLE SEGMENT WORD PUBLIC
TYPE_64 DW 2 DUP(0)      * Reserva interrupcion para TIMER
TYPE_65 DW 2 DUP(0)      * Reserva interrupcion para RELOJ
TYPE_71 DW 2 DUP(0)      * Reserva interrupcion para TECLADO
MINT_TABLE ENDS

```

```

DATA_HERE SEGMENT WORD PUBLIC
DATO_BUFFER DW 4 DUP(0)  * Variable de espacio de Conversion
DATO_TOTAL DB 4 DUP(0)  * Buffer de Conversion Total
DB 14H                  * Caracter de valor cero
VALOR DB 00H            * Valor de conversion inicial
NUM DW 0000H            * Numero de octetos
DATO DB 100 DUP(0)      * Datos para Reloj
DATO_WEL OR 2 DUP(0)    * Valor de mes
DATE DB 6 DUP(0)       * Dato de dia
HOUR DR 6 DUP(0)       * Reserva para dato de hora
COUNT DB 00H          * Dato inicial para Timer
EMP EQU 00FFH           * Direccion inicial de Memoria
IER EQU 00FFH           * Direccion final de Memoria
MIL EQU 3EBH            * Equivalente Hex. de mil
GAN DB 00H              * Dato inicial de ganancia

```

\*Codigo para teclado

```

TABLE DB 77H, 78H, 79H, 7EH, 0B7H, 0B8H, 0BDH, 0BEH
;      0 1 2 3 4 5 6 7
;
; DB 0B7H, 0B8H, 0BDH, 0BEH, 0E7H, 0E8H, 0EDH, 0EEH
;      B 9 A H C D E F

```

\*Codigo para Display

```

DISP DB 80H, 90H, A0H, B0H, C0H, D0H, E0H, F0H
;      0 1 2 3 4 5 6 7
;
; DB 80H, 82H, A7H, B4H, B5H, 86H, 87H, 88H
;      8 9 A B C D E F

```

```

STACK_HERE SEGMENT
DW 100 DUP(0)          * Sin directiva de Stack porque
TOP_STACK LABEL WORD  * se usa EXE2BIN
STACK_HERE ENDS

```

```

CODE_HERE SEGMENT WORD PUBLIC
USEBND CODE_HERE, DS:MINT_TABLE, SS:STACK_HERE

```

\*Inicializa registro de Segmento de Stack, Segmento de datos

```

MOV AX,Stack_HERE
MOV SS,AX
MOV SP,100 SET TOP_STACK
MOV AX,MINT_TABLE
MOV DS,AX

```

\*Define direcciones de procedimientos de interrupciones

```
MOV TYPE_04+2, SEG_TIMER      *Primera interrupcion para TIMER
MOV TYPE_04, OFFSET_TIMER
MOV TYPE_05+2, SEG_RELOJ      *Segunda interrupcion para RELOJ
MOV TYPE_05, OFFSET_RELOJ
MOV TYPE_71+2, SEG_KEY        *Tercera interrupcion para TECLADO
MOV TYPE_71, OFFSET_KEY
```

\*Inicializa Segmento de Datos

```
ASSUME DS: DATA_HERE
MOV AX, DATA_HERE
MOV DS, AX
```

\*Inicializa Puerto Serie

```
MOV DX, 11H                    *Puntero en direccion Registro de Comando
MOV AL, 00H                    *envia zeros para garantizar que el
OUT DX, AL                     *dispositivo esta en formato instruccion
MOV CX, 02H                    *y de comando antes de Reset
DO: LOOP DO                    *retardo despues de enviar cada
OUT DX, AL                     *instruccion de comando
MOV CX, 02H
D1: LOOP D1
OUT DX, AL
MOV CX, 02H
D2: LOOP D2
MOV AL, 40H
OUT DX, AL
MOV CX, 02H
D3: LOOP D3                    *Palabra modo asincrono
MOV AL, 59H                    *4800 bps, x 1, 8 bits, paridad impar
OUT DX, AL                    *1 bit de parada
MOV DX, 11H                    *Direccion de registro de comando
MOV AL, 05H                    *Habilita transmision, DTR=1, Habilita recepcion
OUT DX, AL                    *Operacion Normal, Listo para enviar
```

\* Inicializa PIC 8259

```
MOV DX, 70H                    *Direccion de registro de Comando
MOV AL, 13H                    *Modo simple, ICW4
OUT DX, AL                    *envia ICW1
MOV AL, 01000000B              *Tipo 04, primera interrupcion
MOV DX, 72H                    *puntero en direccion de ICW2
OUT DX, AL                    *envia ICW2
MOV AL, 00000001B              *ICW4, modo 8088
OUT DX, AL                    *envia ICW4
MOV AL, 01111100B              *ICW1 desmascara IR0, IR1, IR7
OUT DX, AL                    *envia ICW1
```

\* Inicializa TIMER 8253

```
MOV DX, 50H                    *Direccion de registro de control
MOV AL, 10010100B              *Palabra para contador 0, modo cuadrada, 32
OUT DX, AL                    *bits, palabra de comando
MOV AL, 05H                    *Valor para 100 Hz
MOV DX, 51H                    *Direccion de contador
OUT DX, AL                    *envia contador
```

Inicializa entrada de programas

```
ARG: CALL TECLADO              *entra de teclado
```

```

CMP AL,01H          *elige opcion 1
JE AD              *va a Adquisicion
CMP AL,02H        *elige opcion 2
JE RE              *salta a Reloj
CMP AL,03H        *sale de Sistema
JE FIN             *termina
JMP ERROR         *otro numero, error
REI CALL CLOCX
JMP AA

```

\* Inicializa variables

```

MOV MOV CANAL,00H  *inicializa canal
MOV TIME,00H      *inicializa tiempo
MOV VEZ,00H       *variable para canal
MOV STO,00H       *conversion total

```

\* Inicializa Adquisicion de Datos

```

CALL TECLADO       *Pregunta por numero de canales
CMP AL,1BH        *de 1 a 24
JA ERR             *mayor numero, error
MOV CANAL,AL      *carga numero de canales
CALL TECLADO      *Pregunta por tiempo entre muestras
MOV COUNT,AL      *Carga constante de tiempo
MOV TIME,COUNT    *salva valor
CALL TECLADO      *Pregunta por ganancia
CMP AL,3BH        *1, x10, x50, x100, x1000
JA ERR             *otro numero, error
MOV BAN,AL        *salva constante
JMP Z              *salta a interrupcion
ERR: MOV AL,87H   *menda mensaje de error
CALL DISPLAY      *a desplegar
MOV AL,00H        *limpia display
CALL TECLADO

```

\* Habilita entrada de interrupcion

```

S11
HERE: JMP HERE    *espera por interrupcion
NOP
RMP

```

\* Procedimiento de Adquisicion de Datos

```

TIMER PROC FAR
FISH AA          *Salva registros
FISH BB
FISH DA
FISH DS
S11              *Salva DS de programa de interrupcion
MOV AA,BTA_HERE *Habilita interrupcion mas alta
MOV DS,00       *Carga datos
MOV COUNT       *Llena DS necesario
INC COUNT       *Incrementa contador
JNZ EXIT        *Si no es cero, espera
MOV COUNT,10H  *si cero, restaura contador
MOV DI,CANAL
MOV CH,41H     *Configuracion de canales
S10: MOV CL,04H *Coloca bit para activar switch
ROR DI
JNZ S10
END TIMER PROC

```

Empieza de Conversion

MOV	BX,OFFSET DATO_BUFFER	'Principia con MSB
MOV	DX,62H	'Direccion de registro de 6255
MOV	AL,89H	'Modo 0, Puerto A y B de salida
OUT	DX,AL	'envia palabra de control
DI:	MOV DX,00H	'posiciona en puerto A
MOV	AL,01H	'habilita switch de conversion
OUT	DX,AL	
TEST	AL,00H	'esta listo?
JZ	OT	'si no, regresa a cargo
MOV	DX,61H	'palabra para puerto B
MOV	AL,00H	
OR	AL,CH	'operacion para canal
OUT	DX,AL	'envia numero de canal
MOV	DX,62H	'direccion para puerto C
D1:	IN AL,DX	'lee byte de A/D
CALL	DELAY	'retardo de 1 mseg
AND	AL,10H	'checa para MSB
JZ	D1	
IN	AL,DX	'lee MSB de A/D
AND	AL,0FH	'enmascara bits mas altos
CMP	AL,04H	've si MSB es 1
JE	L1	'si, carga 1 en buffer
L1:	MOV BYTE PTR[BX],01H	
NC:	DEC BX	'puntero decrementa buffer
D2:	IN AL,DX	'poleo en digito 2
AND	AL,20H	
JZ	D2	
IN	AL,DX	'lee digito 2 de MSB
AND	AL,0FH	'enmascara digitos mas altos
MOV	[BX],AL	'digito 2 BCD a Buffer
DEC	BX	
D3:	IN AL,DX	'poleo para digito 3
AND	AL,40H	
JZ	D3	
IN	AL,DX	'Lee digito 3 de A/D
AND	AL,0FH	'enmascara bits
MOV	[BX],AL	'digito 3 a buffer
DEC	BX	'decrementa buffer
D4:	IN AL,DX	'Poleo para digito 4 (LSB)
AND	AL,80H	
JZ	D4	
IN	AL,DX	'lee digito 4 de A/D
AND	AL,0FH	
MOV	[BX],AL	'digito 4 BCD a buffer
MOV	STO,[BX]	'guarda valor de conversion
MOV	AX,STO	
MUL	GAN	'lo multiplica por ganancia
MOV	STO,AX	'salva valor convertido
CALL	DISFLAY	'manda datos a display
CALL	CVIH	'convierte a hexadecimal
CALL	STORE	'manda almacena dato
INC	VEZ	
CMP	VEZ,DX	
JA	EXIT	
INC	CH	
JMP	STA	

'Deshabilita interrupcion de 8088

EXIT: MOV AL,00100000H

'ENVIA DATOS A LA ESPECIFICA ECU

MOV DX,70H	'direccion de OCW2
OUT DX,AL	'envia OCW2 para fin de interrupcion
POP AX	
POP BX	
POP DX	
POP DS	
IRET	
TIME ENDP	
RELOJ PROC NEAR	
PUSH AX	'Salva registros
PUSH BX	
PUSH DS	'Salva DS de programa de interrupcion
STI	'Habilita interrupcion intermedia
MOV AX,DATA_HERE	'Carga datos
MOV DS,AX	'Carga DS necesario
MOV BX,3E8H	'Constante para un segundo
DEC BX	
JNZ EXIT	'no cero, espera
MOV BX,3E8H	'cero, restaura contador
CALL RECL	'manda parametros a Display
EXIT: MOV AL,00100000B	'OCW2 para no especifica EOI
MOV DX,70H	'direccion de OCW2
OUT DX,AL	'envia OCW2 para fin de interrupcion
POP AX	
POP BX	
POP DS	'Restaura registros
IRET	
RELOJ ENDP	
KEY PROC NEAR	
PUSH AX	'Salva registros
PUSH BX	
PUSH CX	
PUSH DS	'salva DS programa de interrupcion
STI	'habilita tercera interrupcion
MOV AX,DATA_HERE	'Carga datos
MOV DS,AX	
MOV BX,64H	'constante para 100 mseg
DEC BX	'decrementa contador
JNZ EXIT	'no cero, espera
MOV BX,64H	'si cero, restaura contador
CALL TECLADO	
EXIT: MOV AL,00100000	'OCW2
MOV DX,70H	'direccion de OCW2
OUT DX,AL	'envia OCW2 para fin de interrupcion
POP AX	
POP BX	
POP CX	
POP DS	'restaura registros
IRET	
KEY ENDP	
DELAY PROC NEAR	
PUSHF	'salva registros
PUSH BX	
PUSH CX	
MOV CX,0FFH	'retardo de 1 mseg
HERE: LOOP HERE	
DELAY ENDP	

\* Procedimiento para almacenar datos

```
STORE PROC NEAR
    PUSHF                                'salva registros
    PUSH BX
    PUSH DX
    PUSH CX
    PUSH DI
    MOV CX,12H
    MOV SI,OFFSET EMP                    'direccion de RAM
    MOV AL,STO                            'manda valor convertido
    MOV AH,CH                             'con numero de canal
    MOV [SI],AX                           'localidad de RAM
    INC SI
    INC SI
    MOV BX,OFFSET TER                    'ultima localidad de RAM
    MOV AX,[SI]
    CMP BX,AX
    JE YA
    FI: MOV DX,41H                       'registro de control 825I
    YI: IN AL,DX
    AND AL,10000001B                     'checa estado de banderas

    CMP AL,10000001B
    JNE FI
    MOV DX,40H                            'registro de lectura 825I
    MOV AX,STO
    OUT DX,AL                             'manda dato a cinta
    POP DI                                 'restaura registros
    POP CX
    POP DX
    POP BX
    POPF
    YA: RET
STORE ENDP
```

\* Procedimiento para Conversion de BCD a Hexadecimal

```
CVTH PROC NEAR
    PUSHF                                'salva registros
    PUSH BX
    PUSH DX
    PUSH CX
    PUSH DI
    MOV AX,STO                            'copia numero en AX
    MOV BX,AX
    MOV BH,AH                             'lugar para los 2 digitos mas altos
    MOV BL,BL                             'lugar para los 2 digitos mas bajos

    ' Divide digitos, digito en cada registro
    MOV CL,0AH                            ' cuenta de nibble para rotar
    ROR BH,CL                             ' digito 1 en lugar correcto
    ROR BL,CL                             ' digito 3 en lugar correcto
    AND AX,0F0FH                          ' enmascara nibbles mas altos de cada digito

```

\* Copia AX en CX, se puede usar AX para multiplicacion

```
MOV CX,AX
```



```

MOV AX,0000H

* Ahora multiplica cada numero por su posicion

MOV AL,CH          *multiplica byte en AL # word
MOV DI,MIL         *multiplicacion no inmediata
MUL DI             *digito 1 # 1000

* Resultado en DX y AX porque digito BCD no es mas grande que 9
* El resultado estara en AX
* cero DX y suma BL porque ese digito no necesita multiplicacion
* para ocupar posicion, entonces suma el resultado en AX para
* digito 4

MOV AX,0064H      *byte # byte, resultado en AX
MUL CL            *digito 2 # 1000
ADD DX,CX         *suma digito 3
MOV AX,000AH      *byte # byte, resultado en AX
MUL BH
ADD DX,AX         *suma digito 2
MOV AX,DX         *coloca resultado en lugar correcto
POP DI
POP CX
POP DX            *restaura registros
POP BX
POPF
RET
CVI1 ENDP

```

\* Procedimiento para Reloj de Tiempo Real

```

CLOCK PROC NEAR
PUSH DS          *salva registros
PUSHF
PUSH AX
PUSH BX
PUSH CX
MOV AX,00H
C1: CALL TECLADO *pregunta por parametros
CMP AL,01H      *elige opcion 1
JE DEB         *salta a lectura
CMP AL,02H      *elige opcion 2
JE DED         *salta a escritura
CMP AL,03H      *elige opcion 3
JE FIN         *sale del sistema
JMP ERROR      *otro numero, error

```

\* Captura fecha

```

DEB: JMP DEL2   *introduce año, mes, dia
DED: CALL DISP1 *dia de la semana
DEF: MOV CL,02H
CML: RUI1
CMP BH,55H
JNE DEL1
PLL1: JMP DISP
LEL1: CMP BH,0FFH
JE DELC
DEC CH
JNZ DELF
CALL DEL2
CMP BL,FFH

```

```
JE      DELC
CALL   RUT4
CMP    AL,0DH
JNE    DELC
```

\* Captura la hora

```
DELH:  CALL DISP2
DELI:  MOV CL,02H
CALL   RUT1
CMP    BH,55H
JE     DELI1
CMP    BH,FFH
JE     DELH
DEC    CH
JZ     DELM
JMP    DELI
DELM:  CALL TECLADO
CMP    AL,0DH
JNE    DELH
```

\* Se mandan datos capturados de fecha y hora al reloj de tiempo real

```
MOV    DX,20H
MOV    AL,0FH
OUT    DX,AL *selecciona interrupciones para reloj
MOV    AL,0DH
MOV    DX,2FH
OUT    DX,AL *no interrupciones
MOV    AL,05H
MOV    DX,20H
OUT    DX,AL *selecciona programacion de reloj
MOV    AL,01H
MOV    DX,2FH
OUT    DX,AL *modo de 24 horas
MOV    DI,OFFSET DATO
MOV    CL,12
MOV    DX,2DH
DELI:  MOV AL,(DI)
OUT    DX,AL
DEC    DX
INC    DI
DEC    CL
JNZ    DELI
MOV    SI,OFFSET DATOWEK
MOV    AL,(SI)
MOV    DX,2EH
OUT    DX,AL
```

\* Escritura del año bisiesto

```
MOV    AL,01H
MOV    DX,2FH
OUT    DX,AL *año bisiesto
MOV    AL,00H
MOV    DX,20H
OUT    DX,AL *arranca reloj
```

\* Se lleva a cabo la lectura del Reloj

```
DEL20: CALL RUT2
CALL   RUT3
```

```

CALL  TECLADO
JZ    DEL20
CALL  TECLADO
CMP   AL,1BH
JNE   DEL20
JMP   DISP
RET
CLOCK: ENDF

```

'Subrutinas del Reloj de Tiempo Real

```

D11 PROC NEAR
CALL  TECLADO
MOV   DI,OFFSET DATO           'coloca año, mes, dia,
MOV   CH,03H                   'dia de la semana (1-7)
RET
D11 ENDF

```

```

DISP2 PROC NEAR
CALL  TECLADO
MOV   DI,OFFSET DATO           'coloca horas, minutos y segundos
MOV   CL,06H
DES:  INC  DI
DEC   CL
JNZ   DES
MOV   CH,03H
RET
DISP2 ENDF

```

```

RECL PROC NEAR
RL:   IN  AL,DX
AND   AL,0FH
OR    AL,30H
MOV   [SI+BX],AL
DEC   DX
INC   BX
DEC   CL
JNZ   RL
RET
RECL ENDF

```

```

DAY PROC NEAR
CALL  RUT4
MOV   BL,AL
AND   AL,FOH
CMP   AL,30H
JNE   DEL10
MOV   AL,BL
AND   AL,0FH
CMP   AL,01H
JF    D10
CMP   AL,07H
JA    DEL10
MOV   SI,OFFSET DATOWEEK
MOV   [SI],AL
JMP   D12
D10:  MOV  BL,OFFH
D12:  RET
DAY ENDF

```

```

RUT4 PROC
BL:   CALL  TECLADO

```

```
JZ DL
CALL RUT4
CMP AL,1BH
JE D5
MOV BL,AL
AND AL,0FH
CMP AL,30H
JE DJ
MOV BH,0FFH
JMP FIN
DS: MOV BH,55H
JMP DE
DJ: MOV AL,BL
AND AL,0FH
MOV [DI],AL
INC DI
DEC CL
JNZ DL
MOV BH,00H
DE: RET
RUT1 ENDP
```

```
RUT2 PROC NEAR
MOV BL,26
MOV CH,03H
MOV DX,2DH
DLY: MOV CL,02H
DLX: IN AL,DX
AND AL,0FH
OR AL,30H
CALL DISPLAY
DEC CL
JNZ DLX
CALL DISPLAY
INC BL
DEC CH
JNZ DLY
MOV DX,2EH
IN AL,DX
AND AL,0FH
CMP AL,01H
JNE A
```

```
RUT3 PROC NEAR
MOV BL,26
MOV CH,03H
MOV DX,27H
DP: MOV CL,02H
DR: IN AL,DX
AND AL,0FH
OR AL,30H
CALL DISPLAY
INC BL
DEC DX
DEC CL
JNZ DR
DEC CH
JZ DP
JMP DP
DD: RET
```

```
RUT4 PROC NEAR
```



```

MOV AH,01
JMP SAL
HD: MOV AL,HL 'codigo Hexadecimal para tecla en AL
MOV AH,00 'codigo valido para tecla en AH
SAL: NOP
POP DX 'restaura registros
POP CX
POP BX
POPF
RET
TECLADO ENDP

```

' Procedimiento para despliegue de datos

```

DISPLAY PROC NEAR
PUSHF 'guarda registros
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH SI
MOV DI,00H 'puntero en la direccion de buffer
CMP AL,00H 'pregunta si hay dato
JZ DD 'si, carga palabra de control para dato
MOV AL,02H 'no, carga palabra de control de registro
DD: MOV AL,90H 'carga palabra de control para datos
VI: OUT DX,AL 'envia palabra para manejador de display
MOV CL,04H 'contador para numero de caracteres
MOV SI,BX 'libro B para usar en tabla
CMP AL,DISP[BX] 'busca digito en tabla
MOV DX,00H 'puntero a manejador de Display
AV: MOV AL,[SI] 'obtiene caracter para desplegar
CMP CL,02H 'verifica digito con punto decimal
JNE CR 'no, envia caracter
UR AL,B0H 'si, UR en punto decimal
CR: OUT DX,AL 'envia caracter a display
INC SI 'puntero en proximo caracter
LOOP AV 'hasta que se envíen los 4 caracteres
POP SI
POP DX
POP CX
POP BX
POP AX 'restaura banderas
POPF
RET
DISPLAY ENDP
CODE_HERE ENDS

```

```

CALL    CLEAN
LOCATE  12,17
DISPLAY IDE_ADD
LOCATE  14,17
DISPLAY IDE_ADD
LOCATE  16,17
DISPLAY IDE_ADD1
LOCATE  22,27
DISPLAY STO_SIST

```

'Pregunta si se detiene la Adquisicion, almacenando la informacion  
'de las muestras obtenidas para cada canal hasta este instante  
'si no, continua

```

MOV     AH,01
INT     16H
JZ      RDT
MOV     AH,00H
INT     16H
CMP     AL,'P'
JE      SI_AL
CMP     AL,18H
JNE     LEE_KEY
JMP     FIN
SI_AL:
MOV     [STOP],AL
JMP     ROT
SI_TR:
CALL    CLEAN
LOCATE  20,9
DISPLAY IDE_ADD1
CALL    DELAY
CALL    DELAY
JMP     INI

```

\*\*\*\*\*SUBROUTINAS\*\*\*\*\*

' Procedimiento que efectua la Conversion A/D

```

CAD      PROC
                PUSH CX                'guarda registros
                PUSH DX
                PUSH BX
                MOV CH,41H              'configuracion de canales
                YS: MOV CL,04H           'bit para activar switch
                MOV BX,OFFSET DATO_BUFFER 'desplaza buffer
                YX: MOV BYTE PTR[BX],00H 'inicializa puntero
                INC BX
                LOOP YX                 'poleo para recibir dato

'Empiezo de conversion
                MOV BX,OFFSET DATO_BUFFER+3 'principia con MSB
                MOV DX,18H                'direccion de registro 025
                MOV AL,89H                'modo 0, Puertos A,B salida
                OUT DX,AL                 'envia palabra de control
                OT: MOV DX,18H
                MOV AL,01H                'habilita switch de conver
                OUT DX,AL
                TEST AL,00H               'esta listo el dato?
                JZ OT                    'si no, regresa a inicio

```

```

MOV DX,19H      'direccion de Fuerto A
MOV AL,06H     'carga palabra de control
OR AL,CH       'operacion para canal
OUT DX,AL      'envia numero de canal
MOV DX,1AH     'direccion de canal B
DS1: IN AL,DX  'lee byte de CAD
CALL DELAY
AND AL,10H     'checa para MSD
JZ DS1         'lazo para encontrar digit
IN AL,DX       'lee MSD de A/D
AND AL,0FH     'enmascara bits mas altos
CMP AL,04H     've si MSD es 1
JE L1         'si no, carga 1 e buffer
MOV BYTE PTR[BX],01H
JMP NCAR      'salta al siguiente caract
L1: MOV BYTE PTR[BX],01H
NCAR: DEC BX   'puntero a siguiente local
DS2: IN AL,DX  'poleo para digito 2 BCD
AND AL,20H
JZ DS2
IN AL,DX      'lee digito 2 A/D
AND AL,0FH    'enmascara digitos mas alt
MOV [BX],AL   'digito BCD a buffer
DEC BX        'puntero a siguiente local
DS3: IN AL,DX  'poleo para digito 3 BCD
AND AL,40H
JZ DS3
IN AL,DX      'lee digito 3 A/D
AND AL,0FH    'digito 3 a buffer
MOV [BX],AL   'puntero a siguiente local
DEC BX
DS4: IN AL,DX  'poleo para digito 4 BCD
AND AL,80H
JZ DS4
IN AL,DX      'lee digito 4 A/D
AND AL,0FH    'digito 4 BCD a buffer
MOV [BX],AL   'valor convertido
MUL GAN       'multiplica por ganancia
MOV [BX],AL   'salva valor final
POP BX
POP CX
POP DX        '*Salva registros
CAD          ENDP

```

\* Procedimiento para limpieza de Pantalla

```

CLEAN      PROC
           CLS
           RET
CLEAN      ENDP

```

\* Procedimiento para verificar captura de datos por el teclado

```

KEY        PROC
           Y1: MOV AH,01H
           INT 16H
           JZ Y1
           MOV AH,00H
           INT 16H
KEY        ENDP

```



## PROGRAMA DE COMUNICACION CON MICROCOMPUTADORA

```

*****
CODE SEGMENT
    ASSUME CS:CODE, DS:CODE, ES:CODE
    ORG 100H
MIC PROC FAR
    JMP INI                                ' Desplaza a codigo de inicio

    ' DECLARACION DE VARIABLES

    FRO_PATHNAME DB 'C:\FRO_ADD.EXE', 0
    PAR_BLOCK LABEL WORD
    COM_POT_OFFSET DW 7EH
    COM_POT_SEGMENT DW ?
    FRI_AD_OFFSET DW 5CH
    FRI_AD_SEGMENT DW ?
    SEG_AD_OFFSET DW 6CH
    SEG_AD_SEGMENT DW ?
    INF_TO_ARCHIVO DB 1000 DUP (5')
    NOM_ARCHIVO DB 'C:\FILES\00000000.000', 0
    OT_CANAL DB 'MUESTREAR OTRO CANAL (S/N): ', '$'
    IDE_ADD DB 'EL SISTEMA CUENTA CON 24 CANALES', '$'
    IDE_ADD1 DB 'PARA MUESTREAR DE INFORMACION', '$'
    STD_SIST DB 'PARA PARAR LA ADQUISICION TECLEA <P>', '$'
    INFOR DB 'SE ENVIA LA INFORMACION A CADA CANAL', '$'
    ALMACEN DB 'C:\MUESTADD.RAS', 0
    GAN DB 00H
    INF_CANALES DB 50 DUP (0)
    CONT DB 2 DUP (0)
    ALTO DB 2 DUP (0)
    CANAL DB 2 DUP (0)
    DELAY DB 2 DUP (0)
    HORA DB 2 DUP (0)
    MINUTOS DB 2 DUP (0)
    MEN1 DB 'FECHA: ', '$'
    MEN2 DB 'HORA: ', '$'
    MEN3 DB 'Para salir teclee <ESC>', '$'
    MEN4 DB 'Lectura <L>', '$'
    MEN5 DB 'Escritura <E>', '$'
    MEN6 DB 'Reloj de Tiempo Real', '$'
    MEN7 DB 'Introducir Ano, Mes, Dia, Dia de sem (1-7)', '$'
    MEN8 DB 'Introducir Horas: Minutos: Segundos', '$'
    DATO DB 100 DUP (0)
    DATOHEX DB 2 DUP (0)
    LUN DB 'Lunes', '$'
    MAR DB 'Martes', '$'
    MIE DB 'Miercoles', '$'
    JUE DB 'Jueves', '$'
    VIE DB 'Viernes', '$'
    SAB DB 'Sabado', '$'
    DOM DB 'Domingo', '$

    ' Inicializar Segmentos y Parametros

    MOV COM_POT_SEGMENT, DS
    MOV FRI_AD_SEGMENT, DS                ' Inicializa los Segmentos de
    MOV SEG_AD_SEGMENT, DS                ' Todos los parametros

```

\*Aumenta parametro "\C" y ajustar el contador\*

```
MOV BX,80H          'Direccion del contador original
MOV AL,[BX]        'Carga contador original
ADD AL,2
MOV DI,7EH         'Direccion del nuevo contador
CLD                'Para incrementar a DI
STOSB              'guarda nuevo contador
MOV AL,'\'         'carga slash
STOSB              'lo guarda
MOV AL,'C'         'carga parametro
STOSB              'lo guarda
```

\*Ejecuta el Programa

```
LEA DX,PRO_PATHNAME 'Direccion de nombre del programa
LEA BX,PAR_BLOCK    'Direccion control de block
MOV AL,00H          'subfuncion deseada
MOV AL,40H
INT 21H             'ejecuta el otro programa
```

\*Guarda las muestras obtenidas provenientes del 8251 en el archivo  
\*muest.adq estableciendo el numero de Canal seleccionado

```
CALL CAD
LEA DX,MUESTADO
MOV AL,00H          'abre archivo
INT 21H
MOV BX,AX
LEA DX,INF_CANAL
MOV AH,3FH
MOV CX,51
INT 21H
MOV AH,3EH
INT 21H             'cierra archivo
MOV BX,51H
MOV AL,[DX+BX]
CMP AL,'R'
JE REL
```

\*Verifica salida del Sistema

```
MOV DI,OFFSET INF_CANALES
MOV BX,40H
MOV AX,[DI+BX]
CMP AL,1BH
JNE VA
JMP FIN
REL: CALL CLOCK
JMP INT
```

\*Programacion del Puerto Serie 8250 de la Microcomputadora para  
\*comunicacion con el Sistema estableciendo el envio de muestras  
\*provenientes de cada canal

```
CURSOROFF          'inicio de pantalla
MOV SI,OFFSET PARO
MOV CS17,AL
LOCATE 17,9
DISPLAY INFOR
CALL PRO_SERIE     'programa puerto serie
CALL COM_CANAL     'numero de canal
CALL DELAY
```

\* Procedimiento para desplegar caracteres en pantalla

```
DISP PROC
    PUSH BX
    MOV BH,00H
    MOV AH,14
    INT 10H
    POP BX
    RET
DISP ENDP
```

\* Procedimiento para programacion del Puerto Serie B250  
\* para comunicacion con el Sistema de Adquisicion

```
PRO_SERIE PROC
    MOV DX,3FBH 'Direccion de registro
    MOV AL,80H
    OUT DX,AL
    MOV DX,3FBH 'Direccion de lectura
    MOV AL,18H 'velocidad de 4800 BITS/SEG
    OUT DX,AL
    MOV DX,3F9H
    MOV AL,00H
    OUT DX,AL
```

\* Proceso para programar registro de control

```
    MOV DX,3FBH 'registro para palabra de control
    MOV AL,0AH '8 bits de transmision, paridad par
    OUT DX,AL
```

\* Proceso de programacion del Modem

```
    MOV DX,3FCH 'direccion de registro
    MOV AL,03H 'palabra de control
    OUT DX,AL 'envia palabra
```

\* Deshabilitacion de interrupciones

```
    MOV DX,3F9H 'direccion de registro
    MOV AL,05H 'palabra de control
    OUT DX,AL 'envia palabra
    RET
```

PRO\_SERIE ENDP

\* Procedimiento para capturar datos a cada canal

```
PRO_ADD PROC
    MOV CX,00H
    OUT DX,AL
    MOV DX,3FDH 'direccion de registro
    S1: IN AL,DX 'envia dato
    TEST AL,01H
    JZ S1
    MOV DX,3FBH
    IN AL,DX 'pregunta si se desea seguir proceso
    CNP AL,'N' 'si no, salta a almacenamiento
    JE S5
    CNP AL,'F'
    MOV SI,OFFSE1 INF_CANAL
    S2: OUT DX,AL
    MOV DX,3FDH 'direccion de registro
```

```

S6: IN AL,DX
TEST AL,01H
JZ S6
MOV DX,3FBH
IN AL,DX
MOV IH,AL
CMP AL,'F'
JE S9
OUT DX,AL
MOV DX,3FDH
S10: IN AL,DX
TEST AL,01H
JZ S10
MOV DX,3FBH
IN AL,DX
MOV BH,CL
MOV CL,04H
SHL AL,CL
OR AL,BL
MOV [SI],AL
MOV CL,BH
INC SI
INC CX
JMP S2
S9: CALL OPEN_FIL
MOV AL,OFFH
S5: RET

```

PRO\_ADD ENDP

'Procedimiento para guardar en archivo informacion proveniente  
'de los diferentes canales de adquisicion

OPEN\_FIL PROC

```

MOV [NUM],CL
MOV [NUM+1],CH
CALL UP_FIL
LEA DX,NA_FI
MOV AH,3CH
MOV CX,00H
INT 21H
MOV BX,AX
LEA DX,INF_CANAL
MOV AH,40H
MOV CL,[NUM]
MOV CH,[NUM+1]
INT 21H
MOV AH,3EH
INT 21H
RET

```

OPEN\_Fil. ENDP

'Procedimiento que actualiza el nombre del archivo

UP\_FIL PROC

```

MOV SI,OFFSET NA_FI
MOV AL,[CANAL]
MOV BX,20
MOV [SI+BX],AL
INC BX

```

\*Llama procedimiento de lectura de reloj para actualizacion

```
MOV DX,200H
MOV CL,02H
CALL RE_CLOCK
IN AL,DX
AND AL,0FH
MOV CX,BX
MOV BL,41H
CMP AL,00H
JNE Z1
MOV BL,30H
Z1: DEC DX
IN AL,DX
AND AL,0FH
OR AL,BL
MOV BX,CX
MOV [SI+BX],AL
DEC DX
INC BX
MOV CL,04H
CALL RE_CLOCK
INC BX
MOV CL,02H
CALL RE_CLOCK
MOV AL,(STOP)
CMP AL,'P'
JNE W1
MOV AL,'I'
MOV [SI+BX],AL
JMP T1
W1: MOV AL,'C'
MOV [SI+BX],AL
T1: RET
```

UP\_FIL ENDP

\* Procedimiento de lectura de Reloj de tiempo real

RE\_CLOCK PROC

```
TR: IN AL,DX
AND AL,0FH
OR AL,30H
MOV [SI+BX],AL
DEC DX
INC BX
DEC CL
JNZ TR
RET
```

RE\_CLOCK ENDP

\* Procedimiento para retardo en despliegue de mensajes

DELAY PROC

```
PUSH DX
PUSH CX
MOV CX,12H
D1: MOV DX,0FFFFH
D2: DEC DX
JNZ D2
DFC CX
JNZ D1
POP CX
POP DX
```

DELAY ENDP

RET

\* Se pregunta si se desea lectura o modificacion de Reloj

UP\_CLOCK PROC

```
CURSOROFF
DP: CALL DISP5
CA: MOV AH,01H
INT 16H
JZ CA
MOV AH,00H
INT 16H
CMP AL,4CH
JE CB
CMP AL,'L'
JE CB
CMP AL,45H
JE CC
CMP AL,'e'
JE CC
CMP AL,1BH
JNE DP
JMP CFIN
```

\* Se captura la fecha

```
CB: JMP C2
CC: CALL DIS1
CF: MOV CL,02H
CALL DIS1
CMP BH,55H
JNZ DM
CL: JMP DIS
CM: CMP BH,OFFH
JE CC
MOV AL,'/'
MOV AH,14H
INT 10H
DEC CH
JNZ CF
CALL DAY
CMP BL,OFFH
JE CC
CALL DISP4
CMP AL,0DH
JNE CC
```

\* Captura de la hora

```
CJ: CALL DISP2
CI: MOV CL,02H
CALL DISP1
CMP BH,55H
JE CL
CMP BH,OFFH
JE CJ
DEC CH
JNE CM
MOV AL,':'
MOV AH,14
```

```

INT 10H
JMP CI
CM: MOV AH,00H
INT 16H
CMP AL,0DH
JNE CJ

```

\* Se mandan datos de Fecha y Reloj establecidos al Reloj

```

CALL START
MOV DI,OFFSET DATO
MOV CL,12
MOV DX,20DH
C1: MOV AL,(DI)
OUT DX,AL
DEC DX
INC DI
DEC CL
JNE C1
MOV SI,OFFSET DATOWEK
MOV AL,(SI)
MOV DX,20EH
OUT DX,AL

```

\* Escritura del año bisiesto

```

MOV AL,01H
MOV DX,20FH
OUT DX,AL
MOV AL,00H
MOV DX,20EH
OUT DX,AL

```

\* La lectura del reloj se efectua

```

C2: CLS
LOCATE 25,51
DISPLAY MEN3
LOCATE 12,19
DISPLAY MEN1
LOCATE 17,19
DISPLAY MEN2
C20: CALL PRO2
CALL PRO3
MOV AH,01H
INT 16H
JZ C20
MOV AH,00H
INT 16H
CMP AL,1BH
JNE C20
JMP DISP

```

```

CFIN: CURSORON
CLS
RET

```

UP\_CLOCK: ENDP

\* Procedimiento para iniciar Reloj de Tiempo Real

```

START PROC
MOV DX,200H
MOV AL,0FH
OUT DX,AL

```

```

MOV AL,00H
MOV DX,20FH
OUT DX,AL
MOV AL,05H
MOV DX,200H
OUT DX,AL
MOV AL,01H
MOV DX,20FH
OUT DX,AL
RET
* modo de 24 Horas

START ENDP

* Procedimiento para desplegar mensajes de Reloj
DISP1 PROC
CLS
LOCATE 25,51
DISPLAY MEN3
LOCATE 9,19
DISPLAY MEN7
LOCATE 12,19
DISPLAY MEN1
MOV DI,OFFSET DATO
MOV CH,03H
RET
DISP1 ENDP

* Segundo procedimiento
DISP2 PROC
CLS
LOCATE 25,51
DISPLAY MEN3
LOCATE 9,19
DISPLAY MEN8
LOCATE 12,19
DISPLAY MEN2
MOV DI,OFFSET DATO
MOV CL,06
C15: INC DI
DEC CL
JNZ C15
MOV CH,03H
RET
DISP2 ENDP

* Procedimientos para captura, actualizacion, almacenamiento,
* programacion y despliegue de la informacion del Reloj
DAY PROC
CALL PRD4
MOV AL,BL
AND AL,0F0H
CMP AL,50H
JNE C10
MOV AL,BL
AND AL,0FH
CMP AL,01H
JB C10
CMP AL,07H
JA C10
MOV SI,OFFSET DATOWE1:
MOV [SI],AL
JMP C10

```



```
C10: MOV BL,0FFH
C12: RET
DAY ENDP
```

PROC4 PROC

```
CD: MOV AH,01H
INT 16H
JZ CD
CALL PROC4
CMP AL,1BH
JE CS
MOV BL,AL
AND AL,0F0H
CMP AL,30H
JE CJ
MOV BH,0FFH
JMP NO
CS: MOV BH,55H
JMP CE
CJ: MOV AL,BL
AND AL,0FH
MOV [DI],AL
INC DI
DEC CL
JNZ CD
MOV BH,00H
NO: JMP START
CE: RET
```

PROC1 ENDP

\* Segundo procedimiento

PROC2 PROC

```
MOV BL,26
MOV CH,03
MOV DX,20DH
XY: MOV CL,02H
XX: LOCATE 13,BL
IN AL,DX
AND AL,0FH
OR AL,30H
MOV AH,14
INT 10H
INC BL
DEC DX
DEC CL
JNZ XX
LOCATE 12,BL
MOV AL,'/'
MOV AH,14
INT 10H
INC BL
DEC CH
JNZ XY
MOV AX,20E1H
IN AL,DX
AND AL,0FH
CMP AL,01H
JNE X1
DISPLAY LUN
X1: CMP AL,02H
JNE X2
```

```

DISPLAY MAR
X2:  CMP AL,03H
JNE  X3
DISPLAY HIE
X3:  CMP AL,04H
JNE  X4
DISPLAY JUE
X4:  CMP AL,05H
JNE  X5
DISPLAY VIE
X5:  CMP AL,06H
JNE  X6
DISPLAY SAB
X6:  CMP AL,07H
JNE  X7
DISPLAY DOM
X7:  RET

```

```

PRO2  ENDP

```

```

* Tercer procedimiento

```

```

PRO3  PROC

```

```

MOV  BL,26H
MOV  CH,03H
MOV  DX,207H
XP:  MOV CL,02H
XR:  LOCATE 17,BL
IN   AL,DX
AND  AL,0FH
OR   AL,3CH
MOV  AH,14
INT  10H
INC  BL
DEC  DX
DEC  CL
JNZ  XR
DEC  CH
JZ   XQ
LOCATE 15,BL
MOV  AL,'!'
MOV  AH,14
INT  10H
INC  BL
JMP  XP
XQ:  RET

```

```

* Cuarto procedimiento

```

```

PRO4  PROC

```

```

MOV  AH,00H
INT  16H
MOV  AH,14
INT  10H
RET

```

```

PRO4  ENDP

```

```

* Quinto procedimiento

```

```

PRO5  PROC

```

```

CLS
LOCATE 7,24
DISPLAY MEN6

```

PROG    ENDP

LOCATE 12,19  
DISPLAY MEN4  
LOCATE 17,19  
DISPLAY MEN5  
LOCATE 25,51  
DISPLAY MEN3  
RET

## BIBLIOGRAFIA

## BIBLIOGRAFIA

HALL, DOUGLAS V.

Microprocessors and Interfacing.

McGraw-Hill N.Y., 1986

LIU, YU-CHENG & GIBSON, GLENN A.

Microcomputer Systems:

The 8086/8088 Family

Architecture, Programming, and Design.

Prentice Hall. N.J., 1986

NORTON, PETER.

Assembly Language Book for the IBM PC

Braddy. U.S.A., 1986

Mc NAMARA, JOHN E.

Technical Aspects of Data Communications

D. E. C. U.S.A., 1982

MORGAN, CHRISTOPHER & WHITE, MITCHELL

Introducción al Microprocesador 8086/8088

McGraw-Hill. México, 1985

HNATEK, EUGENE

Applications of Linear Integrated Circuits

John Wiley & Sons. U.S.A., 1975

WIATROWSKY, CLAUDE

Circuitos Lógicos y Sistemas de Microcomputadoras

Ed. Limusa. México, 1987

KRUGLINSKY, DAVID

Guía a las Comunicaciones del IBM PC

McGraw-Hill. España, 1985

COUGHLIN, ROBERT & DRISCOLL, FREDERICK

C. I. Lineales y Amplificadores Operacionales

Prentice-Hall. México, 1987

SERIE: MUNDO ELECTRONICO

Interconexión de Periféricos a Microprocesadores

Boixareu Editores. Barcelona, 1980

INTEL

Microprocessor and Peripheral Handbook Vol. 1

U.S.A. 1988

INTEL

Memory Component Handbook

U.S.A. 1988

MOTOROLA

CMOS

U.S.A. 1978

NATIONAL SEMICONDUCTOR

Linear Databook

U.S.A. 1980

NATIONAL SEMICONDUCTOR

TTL Logic Databook

U.S.A. 1982