



G-601964

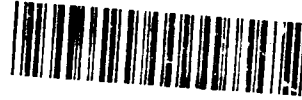
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
FACULTAD DE INGENIERIA



CIFI

Análisis y Procesamiento

601964



CAJA 158

TEMAS DE  
MICROCOMPUTACION

JUAN B. MARTINEZ GARCIA

DIVISION DE INGENIERIA MECANICA Y ELECTRICA  
DEPARTAMENTO DE COMPUTACION

FI/DIME/85-006

## Introducción

|            |   |     |
|------------|---|-----|
| Temas:     | 1. 'Diseño de un sistema básico Z-80'                                     | 3   |
|            | 2. 'Periféricos para microprocesadores'                                   | 4   |
|            | 3. 'Indicadores visuales fosforescentes'                                  | 12  |
|            | 4. 'Control de video'   | 43  |
|            | 5. 'Líneas telefónicas y módems'  | 48  |
|            | 6. 'M200, módem asíncrono a 200 bauds'                                    | 70  |
|            | 7. 'Características específicas de microprocesadores de 16 bits'          | 79  |
|            | 8. 'Medición de temperatura con microcomputadoras'                        | 86  |
|            | 9. 'Sistema digital de información para la planta de bombeo de Xotepingo' | 97  |
|            | 10. 'Red Telemétrica Digital'   | 117 |
|            | 11. 'Sistema automático de supervisión y control de pozos'                | 124 |
|            | 12. 'Ejercicios 1, 2, 3 y 4'  | 145 |
|            |   | 160 |
| Apéndices: | a) Espec. Z-80. CPU   | 181 |
|            | b) Espec. 2716 MOS ROM  | 213 |
|            | c) Espec. 6116 Ram Estática   | 220 |
|            | d) Espec. MOC 3011 Controlador de Triac                                   | 225 |
|            | e) Espec. ADC 0801 Conversor A/D  | 231 |
|            | f) Espec. MTX 1632 Controlador de video                                   | 237 |
|            | g) Espec. 80C48 Microcomputadora en un solo chip                          | 241 |

### Introducción

En los temas presentados en este escrito, se pretende dar una visión general de los microprocesadores, el diseño de sistemas básicos y su acoplamiento con los periféricos más empleados, como son los teclados, los despliegues luminosos, los receptores de TV comerciales, etc.

Se cubren temas relacionados a la telecomunicación entre microcomputadoras, módems y líneas telefónicas, y medición de temperatura.

También se incluye un tema en relación con los nuevos microprocesadores de 16 bits que da una visión general de este nuevo campo.

Por último, se exponen tres aplicaciones reales: un sistema de adquisición y despliegue de datos, una red de telemetría con un nivel de concentración, empleando los nuevos dispositivos 'microcomputadoras en un solo circuito integrado', y un sistema programable para control de procesos.

Los ejercicios propuestos en el último tema cubren gran parte del material y permiten su mejor aprovechamiento.

En los apéndices se dan especificaciones de varios circuitos LSI empleados en diversos acoplamientos utilizados a lo largo de los temas.

DISEÑO DE UN SISTEMA BASICO

Z80

## 1. RESUMEN

Se describen y analizan los circuitos para construir un sistema básico con el  $\mu p$  Z80. El sistema cuenta con 2K de ROM, 1 puerto de entrada y uno de salida con posibilidad de expansión hasta 4 puertos de entrada y 4 de salida.

## 2. DESCRIPCIÓN GENERAL

El campo de aplicación del sistema está reducido a pequeños sistemas de control, temporizadores, terminales remotas, etc. Dispositivos como un teclado, despliegues luminosos de 7 segmentos, LEDs, relevadores, micro-interruptores se pueden conectar y controlar en forma inmediata.

Como el sistema no cuenta con RAM, no pueden hacerse ninguna de las siguientes operaciones:

- Llamados a subrutina, puesto que la dirección de retorno se guarda precisamente en RAM. Sin embargo, con el uso de la instrucción JP(HL) puede resolverse el problema.
- Almacenamiento de resultados parciales. Los 14 registros internos son suficientes en muchas aplicaciones.

Para simplificación del circuito, se utiliza el mapeo de puertos de E/S en memoria, lo que permite, además, utilizar todas las instrucciones de memoria sobre los puertos, en particular SET, RES y BIT.

Se dejan disponibles 3 líneas de control para manejo de 3 puertos de entrada y otras tantas para puertos de salida de tal forma de facilitar su probable expansión.

El circuito se muestra en la fig. 1.

En seguida se hace un análisis en condiciones estáticas y dinámicas de los acoplamientos entre circuitos.

### 3. ANALISIS ESTÁTICO

En el análisis estático debe verificarse que los niveles de voltaje y corriente existentes entre las componentes no rebasen las especificaciones dadas por el fabricante.

-Bus de datos: durante los ciclos de lectura, solo 1 dispositivo tendrá acceso al bus. Como todos las componentes poseen salidas compatibles con TTL y las entradas del Z80 también son compatibles TTL, no existe ningún problema.

En los ciclos de escritura sin embargo, habrá, en el peor de los casos, 4 dispositivos (los 4 puertos de salida) cargando simultáneamente cada línea del bus de datos. Para continuar, es necesario separar el análisis para niveles altos y bajos.

A nivel alto el Z80 es capaz de suministrar 250  $\mu$ A y a nivel bajo es capaz de 'absorber' hasta 1.8 mA. (ver especificaciones Z80).

Las entradas del puerto de salida 74LS364 toman 20  $\mu$ A en estado alto y suministran 0.4 mA en estado bajo. (ver fig. 2).

Lo anterior quiere decir que cada línea del bus de datos del Z80 puede manejar hasta 4 dispositivos 74LS364 lo que significa un 'fan out' de 4.

-Bus de direcciones y control: Las líneas  $A_0$ ,  $A_1$ ,  $A_{11}$  y  $\overline{RD}$  son las únicas que podrían dar problemas debido a que manejan simultáneamente varias cargas.

Sin embargo la memoria 2516 no carga las líneas de direcciones porque sus entradas son MOS (pero también son compatibles TTL).

$A_{11}$ , que es la que más cargas tiene, está conectada a 3 entradas 'LS' simultáneamente. Pero como ya se vió el 'fan out' de cada salida Z80 es de 4 'LS'.

#### 4. ACOPLAMIENTO DINAMICO Z80-2516.

Los diagramas de tiempos de ambos dispositivos se muestran en la figura 3. Con el Z80 trabajando a 2MHz, se tienen disponibles casi 1,250 ns a partir del momento que el bus de direcciones tiene información válida hasta que se hace el muestreo del bus de datos. Como el tiempo de acceso de la 2516 es de 450 nseg. como máximo, no existe ninguna restricción en el ciclo de lectura de memoria y el acoplamiento se hace sin necesidad de añadir ciclos de reloj mediante la señal WAIT.

#### 5. ACOPLAMIENTO DINAMICO Z80 - PUERTOS DE E/S.

Se eligieron los circuitos 74LS244 como puerto de entrada y 74LS364 como puerto de salida.

Es obvio que no habrá problemas dinámicos debido a que ambos dispositivos responden muy rápidamente. Hay que cuidar, sin embargo, que la señal de lectura del 74LS244 sea activo baja, y que la señal de carga del 74LS364

terga su flanco ascendente después que la información en el bus de datos sea válida. Ambas condiciones se cumplen con las señales  $\overline{RD}$  y  $\overline{WR}$  del  $\mu p$  Z80 (ver diagramas de tiempos del Z80, 74LS244 y 74LS364).

#### 6. DECODIFICACION DE DIRECCIONES

Con los puertos de E/S mapeados en memoria, el circuito de decodificación utiliza  $A_{11}$  para diferenciar entre memoria y puertos. Se usa un circuito integrado (74LS155) con 2 decodificadores 2 a 4. Las líneas de dirección  $A_0$  y  $A_1$  se conectan comunes a ambos. Con  $\overline{A_{11}}$  y  $\overline{RD}$  se habilita alguno de los puertos de entrada y con  $A_{11}$  y  $\overline{WR}$  se habilita alguno de los puertos de salida. (ver fig. 4).

Las direcciones así asignadas son:

0 0 0 0 H a 0 7 F F H Memoria PROM

0 8 0 0 H  
0 8 0 1 H  
0 8 0 2 H  
0 8 0 3 H

Puertos de entrada y salida

Pueden emplearse idénticas direcciones para puertos de entrada y salida por que las señales  $\overline{RD}$  ó  $\overline{WR}$  garantizan el acceso solo a uno de ellos.

#### 7. PRUEBAS

Para probar el correcto funcionamiento del circuito, se conecta J2 al módulo de E/S que se muestra en la fig. 5.

De esta forma cada bit de salida tiene conectado un LED y cada bit de entrada un miniinterruptor.



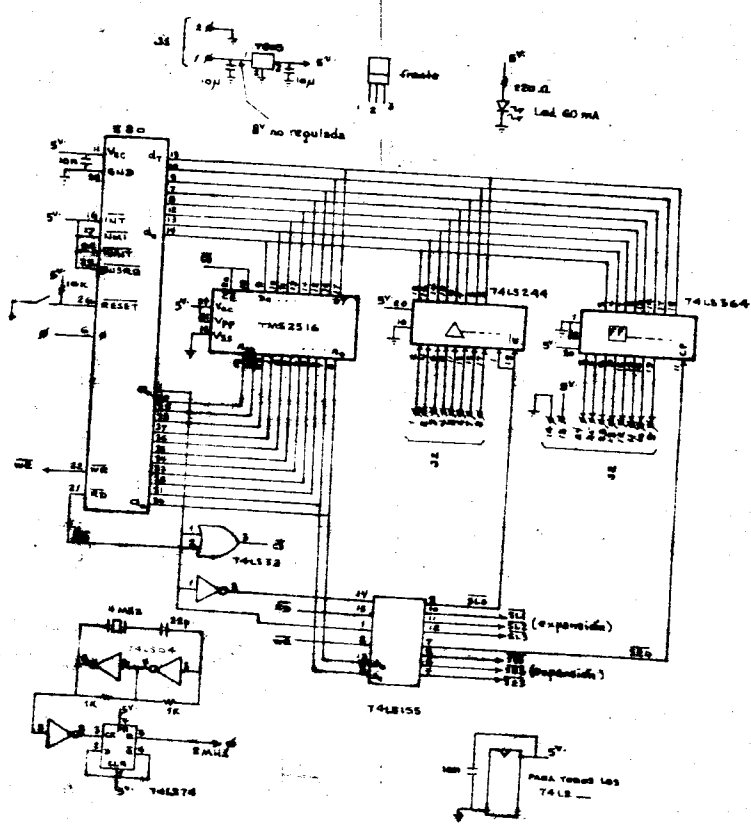


FIG 1. SISTEMA BASICO Z-80  
 DIC. 1982

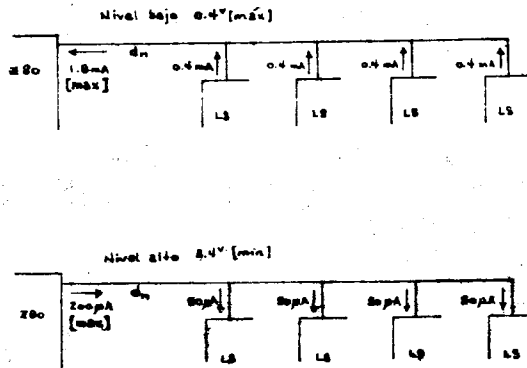


FIG 2. VOLTAJES Y CORRIENTES PARA UNA LÍNEA DEL Z80, A CARGA MÁXIMA (4 CIRCUITOS 'LS').

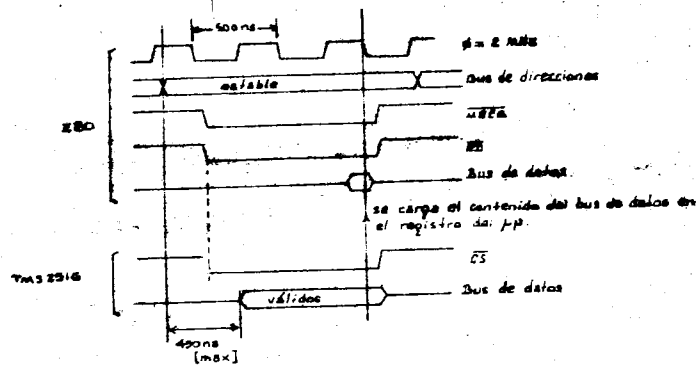


FIG 3. OPERATIVIDAD DURANTE EL CICLO DE LECTURA DEL pp Z80 Y LA PROM. TMS2216.

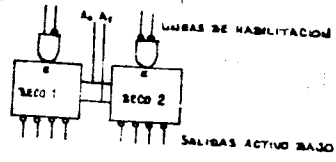


FIG 4. DIAGRAMA DE BLOQUES DEL CIRCUITO 74LS155

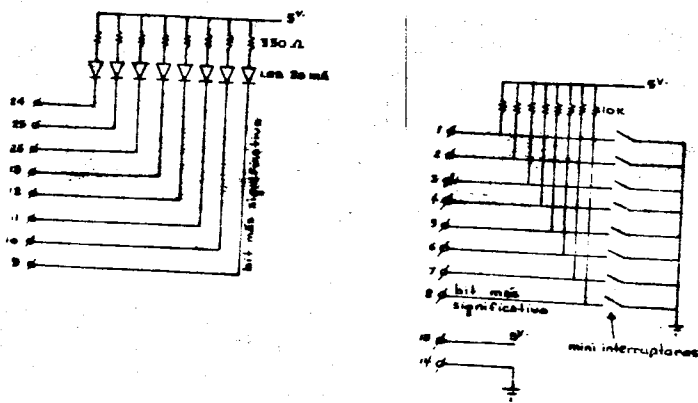


FIG 5. MODULO DE E/S PARA SISTEMA BASICO 880

**PERIFERICOS  
PARA  
MICROPROCESADORES**

- Introducción
- 1. Teletipo y CRT
- 2. Teclado
- 3. Indicador Visual
- 4. ADC-DAC
- 5. Standards de Comunicación
- 6. Bibliografía

### 1. Teletipo:

El teletipo es el dispositivo de entrada-salida más empleado para intercomunicación con microcomputadoras. Opera a velocidades que van de 110 a 300 bauds dependiendo del modelo.

La transmisión se realiza de manera asncrona y en serie, empleándose palabras de 11 bits distribuidos de la siguiente manera (fig. 1.1):

- 1 bit, de inicio (start bit)
- 7 bits, el código ASCII del caracter correspondiente
- 1 bit de paridad
- 2 bits de fin de palabra (stop bits)

La fig. 1.2 muestra la manera como el distribuidor de un teletipo electromecánico genera un caracter.

Existen 2 maneras de realizar la transmisión:

- a) por voltaje (+12<sup>v</sup> y -12<sup>v</sup>)
- b) Lazo de corriente (20 ó 60 ma)

Por otro lado, la comunicación puede ser:

- a) Half duplex: Una sola línea, intercomunicación no simultánea.
- b) Full duplex: 2 líneas, intercomunicación simultánea.

La fig. 1.3 muestra una interface H.D. para malla de corriente, 20 ma. La fig. 1.4 muestra transmisión por lazo de 60 ma. - para líneas de hasta 3,000 m.

La interface del TTY con el microprocesador se realiza, por lo general, con ayuda de un USART (Universal Synchronous Asynchronous Receiver Transceiver) que realiza la conversión paralelo serie y viceversa.

Sin embargo, frecuentemente se realiza por software y en serie, eliminando el empleo del USART. fig. (1.5).

La fig. 1.6 muestra el diagrama de flujo de rutinas para leer y escribir 1 caracter del teletipo.

El CRT\* puede operar a velocidades muy superiores al teletipo (1,200 bauds), debido a que la escritura la realiza un haz de electrones por deflexión electrostática.

La escritura de caracteres en la pantalla requiere de una memoria interna para el continuo "refresco" de información.

Por lo general los CRT realizan su interface a la microcomputadora con el standard EIA RS 232-C (ver sección 5).

Sus aplicaciones son, sin embargo, mucho más numerosas que las de un teletipo (Gráficas, objetos tridimensionales, etc).

\*Cathode Ray Tube

## 2. Teclado:

La interface de la microcomputadora al teclado puede realizarse de 2 formas: por lógica alambrada (hardware) o por programa (software). En el primer caso el módulo del teclado contiene circuitos que identifican automáticamente la tecla oprimida y la  $\mu$ c. solo necesita leer el carácter en ASCII. En el segundo, la  $\mu$ c. realiza la identificación mediante programa.

### Rebote:

El rebote (Bouncing) es un problema que se presenta al hacer contacto 2 partes metálicas (opresión de una tecla), consecuencia de un rebote entre las mismas antes de tener contacto fijo. (fig. 2.1).

Es obvio que realizar lecturas durante el lapso de rebote derivaría en errores. El problema se evita de 2 maneras: filtrando el rebote en hardware (fig. 2.2) o por medio de un retraso por software que evite lecturas en dicho lapso.

### Identificación por Software:

El siguiente ejemplo ilustra la identificación de una tecla oprimida en un teclado de 16 en arreglo matricial. (fig. 2.3).

El  $\mu$ p emplea un puerto (4 bits) de salida y otro (4 bits) de entrada.

Supóngase que existe una tecla oprimida en el punto indicado y es tarea del  $\mu$ p identificarla.

El  $\mu$ p escribirá secuencialmente unos en c-b-c-d, leyendo cada vez e-f-g-h. De esta manera, la tecla oprimida queda indentificada por el cruce de las líneas c-f.



Identificación por hardware:

17

Supóngase un teclado de 64 teclas en arreglo matricial 8 x 8 (fig.2.4).

Mientras no exista tecla oprimida, el contador barre con unos las columnas mientras los renglones son leídos a través del multiplexor.

Tan pronto como se oprime una tecla, la salida del multiplexor generará la señal STROBE. Dicha señal comunica a circuitos externos que una tecla fué oprimida e identificada, al tiempo que inhibe la cuenta del contador.

La tecla es identificada por el código presente en las líneas NO...N5 del contador.

Una memoria ROM conectada a dichas líneas generará finalmente el código ASCII de la tecla correspondiente.

Identificación por inversión de líneas:

Los PPI permiten la programación de cada uno de los puertos.

Este hecho puede ser aprovechado de la siguiente manera, para el caso del teclado de 16 teclas (fig.2.5):

- (a) Se escriben ceros en el puerto de salida,
- (b) Se genera una interrupción al oprimirse una tecla,
- (c) Se lee el puerto de entrada,
- (d) Se invierte el sentido de flujo de información y se vuelve a escribir el código leído en (c) y
- (e) Se lee el nuevo puerto de entrada

La tecla es identificada por ambas lecturas:  $\begin{matrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{matrix}$

El procedimiento requiere de una programación muy sencilla.

### 3. Indicador Visual:

Existen varios tipos de indicadores visuales. Los más empleados son los de diodos emisores de luz (LED); entre éstos se tienen 2 opciones:

- (a) De 7 segmentos
- (b) De matriz de puntos.

#### DISPLAY DE 7 SEGMENTOS:

La interface puede realizarse directamente conectando un decodificador BCD-7 segmentos a cada dígito y controlando los mismos a través de puertos de salida.

Sin embargo, con el fin de ahorrar decodificadores y puertos, se emplea frecuentemente la técnica de multiplexaje (fig. 3-1).

Cada uno de los dígitos es habilitado en forma consecutiva por medio de sus drivers durante un cierto lapso, al tiempo que el código BCD correspondiente es escrito sobre el decodificador el cual habilita los correspondientes drivers de segmento.

Si la operación se realiza a una frecuencia suficientemente alta, el multiplexaje es invisible al observador que verá cada dígito encendido de manera continua.

Si se tienen N dígitos multiplexados, la corriente inyectada por los drivers de segmento deberá ser N veces mayor que en el caso no multiplexado.

REFRESCO: Como en el CRT, el circuito requiere de una memoria de refresco.

La memoria es barrida continuamente por un contador, el cual también activará los drivers de cada dígito.

Si se desea escribir un nuevo dígito en cualquier posición, la  $\mu$ c cargará el contador con la dirección, al tiempo que escribe el nuevo dato por las líneas de entrada (fig. 3.2).

#### MATRIZ DE PUNTOS:

Cada módulo contiene 35 leds en arreglo matricial de 7 x 5 (fig. 3.3).

#### GENERADOR DE CARACTERES (fig. 3.4)

El ROM G.D.C contiene el código para cada columna, correspondiente a cada carácter ASCII. Las líneas A8, A9, A10 seleccionan la columna.

Para el multiplexaje de varios módulos, el barrido se realiza como lo indica el diagrama de flujo de la fig. 3.5 para un indicador de 16 caracteres.

El control completo del barrido, - efectuado por 2 contadores - se muestra en la fig. 3.6. Los bits menos significativos efectúan el barrido por columna y los más significativos contienen la dirección del módulo habilitado.

#### 4. DAC-ADC

Los conversores (DAC ó ADC) se emplean frecuentemente como periféricos de microcomputadoras, siempre que es necesario el monitoreo y/o control con señales analógicas.

##### Convertor Digital Analógico (DAC):

El convertor más simple puede construirse como se muestra en la fig. (4.1). Cada bit duplica la ganancia de su inmediato anterior y la suma total de corrientes a través de R produce el voltaje correspondiente.

Es más usual, sin embargo, adquirirlo en forma integrada (fig. - 4.2). El circuito consta de un circuito de referencia que define  $I_{REF}$ , una red resistiva en escalera que divide por 2 la corriente en cada rama con respecto a la inmediata anterior ( $I_{REF}$ ,  $I_{REF}/2$ ,  $I_{REF}/4...$ ), y un sumador de salida.

Los switches - en realidad transistores controlados por las entradas digitales - conducen la corriente por cada rama de la red ya sea a tierra o al sumador (tierra virtual). La ganancia queda determinada por  $I_{REF}$ .

Entre los DAC más conocidos en el mercado se tiene:

| Fabricante | Número   | Bits    | Tiempo de conversión |
|------------|----------|---------|----------------------|
| PMI        | DAC -08  | 8 bits  | 100 ns               |
| MOTOROLA   | MC 1408  | 8 bits  | 300 ns               |
| DATEL      | DAC4212D | 12 bits | 1 $\mu$ s            |

### Conversor Analógico Digital (ADC):

Existen varios tipos:

- (a) De aproximaciones sucesivas
- (b) De rastreo (tracking)
- (c) De comparación

#### Aproximaciones Sucesivas:

El método de aproximaciones sucesivas es el más extensamente usado con microcomputadoras, debido a que el propio microprocesador puede efectuar por programa las aproximaciones.

Esta posibilidad reduce grandemente el costo total del conversor, cuyos circuitos se simplifican a un DAC y un comparador (fig. 4.3).

La fig. 4.4 muestra el diagrama de flujo para la rutina de aproximaciones sucesivas, para un conversor de 4 bits.

Para un conversor de 8 bits, el tiempo total de conversión sería de  $8\tau$  donde  $\tau$  es el período transcurrido en la ejecución del programa entre los puntos a y b del diagrama.

Los sistemas de adquisición de datos que no requieren altas frecuencias de muestreo emplean usualmente esta técnica.

#### Sample & Hold (S&H)

Los ADC requieren absoluta estabilidad de la señal durante el período de conversión. Para señales de cierta frecuencia se hace necesario el empleo de circuitos que retengan el valor de la señal durante dicho período.

Los circuitos S&H guardan en un capacitor el valor del voltaje muestreado durante el tiempo necesario (fig. 4.5).

#### Convertor de Rastreo:

Como su nombre lo indica, el convertor sigue continuamente a la señal como se muestra en la fig. 4.6.

La estructura del circuito (fig. 4.7) es idéntica a la de un modulador delta.

El tiempo de conversión se encuentra limitado únicamente por la rapidez de los circuitos internos, pudiendo leerse muestras a la misma frecuencia del reloj.

#### Comparación Directa:

Es la técnica más rápida de conversión análogo-digital, pero también la más costosa. La fig. 4.8 muestra el diagrama de un convertor de 3 bits.

El codificador de prioridad dará en su salida el código correspondiente al comparador con mayor prioridad y cuyo  $V_{REF}$  sea mayor que  $V_x$ .

Un convertor de 8 bits requeriría 256 comparadores para su funcionamiento.

El tiempo total de conversión es de aproximadamente 50 ns.

## 5. Estándares de comunicación.

Existen algunos estándares de comunicación serie y paralelo, usados por la mayoría de los fabricantes.

Entre los más empleados se encuentran:

### EIA RS 232C:

La norma define un número de características eléctricas y mecánicas del canal de comunicación entre Equipo Terminal de datos y Equipo de comunicación de datos, empleando información binaria en serie.

Algunas de las características son las siguientes:

- aplicable para velocidades de hasta 20,000 bauds.
- aplicable para comunicación síncrona o asíncrona.
- aplicable para canales FULL DUPLEX Y HALF DUPLEX.

La fig. 5.1 muestra la asignación de señales para un conector de 25 pines; la norma define señales para 2 canales de comunicación, uno principal y otro secundario.

Generalmente, el estándar es empleado para comunicación entre un modem y la computadora o terminal.

### BUS 100

El Bus 100 es un standard empleado para transferencia de información entre módulos.

El bus provee 100 líneas distribuidas de la siguiente manera:

16 ---- datos (E/S)  
16 ---- direcciones  
39 ---- control  
8 ---- interrupciones  
21 ---- no empleados

Muchas de estas líneas no son usualmente empleadas v.gr frecuentemente se tiene 1 solo bus de datos bidireccional (8 líneas).

El estándar es empleado por muchos fabricantes como opción para expansión de sistemas.

#### IEEE 488:

El IEEE 488 fué diseñado para comunicación entre sistemas - más que módulos (computadoras, volímetros, generadores, etc).

La configuración general de comunicación se muestra en la - fig. 5-4.

El bus consiste de:

- 8 líneas bidireccionales de datos.
- 3 líneas para control de transferencia de datos.
- 5 líneas generales de control.

El bus de datos transfiere también la dirección y comandos al dispositivo seleccionado.



Los sistemas intercomunicados deben tener alguna de las siguientes funciones:

- Controlador
- Transmisor (Talker)
- Receptor (Listener)

6. Bibliografia.

1. **Microprocessor Interfacing Techniques.**

AUSTIN LESEA

RODNAY ZAKS

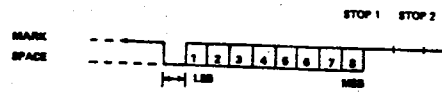


FIG.1.1 TRANSMISION SERIE DEL TELETIPO

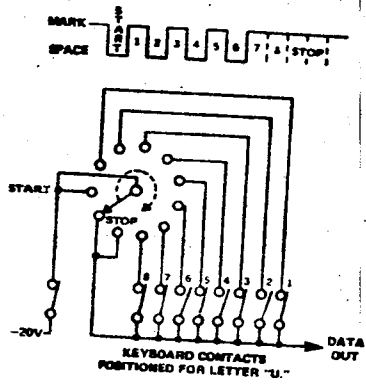


FIG.1.2 FUNCIONAMIENTO DEL DISTRIBUIDOR

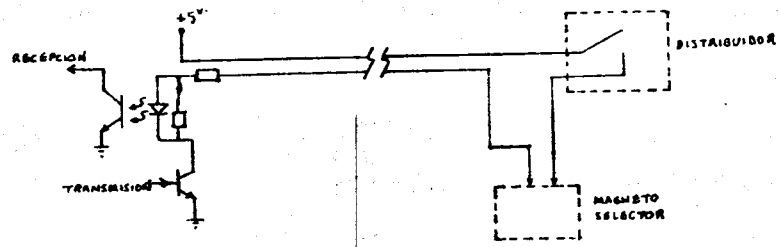


FIG.1.3 INTERFAZ HALF DUPLEX 20 MA

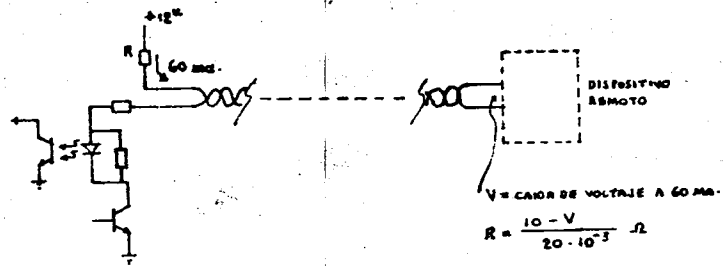


FIG.1.4 INTERFAZ HALF DUPLEX 60 MA

```

TELETYPE OUTPUT SUBROUTINE (ASSUME TTY CONNECTED TO PORT 2 BIT 0)
: THIS SUBROUTINE ENTERED WITH CHARACTER TO BE OUTPUT IN THE C REGISTER
TYOUT: MVI B,11 : SET COUNTER FOR 11 BITS
MOV A,C : CHARACTER TO ACCUMULATOR
ORA A : CLEAR CARRY-FOR START BIT
MORC: OUT 2 : SEND CARRY TO TTY
CALL DELAY : KILL TIME
RAR : POSITION NEXT BIT
STC : SET CARRY-FOR STOP BITS
DCR B : DECREMENT BIT COUNTER
JNZ MORE : DONE?
RET : YES

: 9 MSEC DELAY (ASSUME HC WAIT STATES)
DELAY: MVI D,6
DLO: MVI E,2000
DHI: DCR E : 1.5 MSEC
JNZ DHI : INNER LOOP
DCR D
JNZ DLO
RST

```

FIG.1.5 RUTINA DE SALIDA. TRANSMISION SERIE

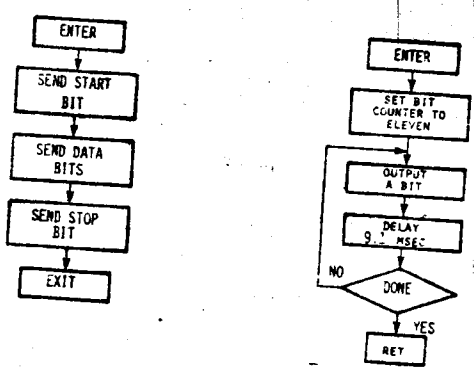
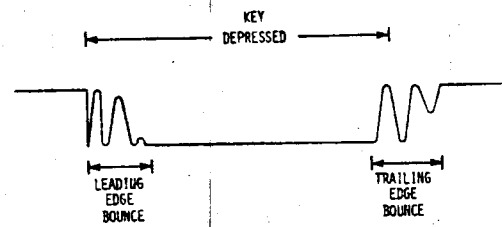


FIG.1.6 DIAGRAMAS DE FLUJO RUTINAS DE ENTRADA Y SALIDA.



- BOUNCE IS 10-20 MSEC
- HARDWARE SOLUTION: R-C FILTER
- SOFTWARE SOLUTION: VERIFY KEY STATUS FOR 20 MS

FIG.2.1 REBOTE

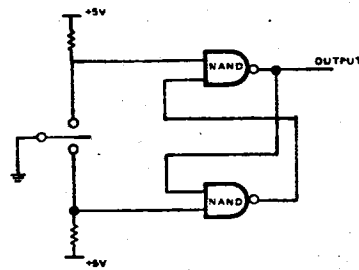


FIG.2.2 ELIMINACION DEL REBOTE POR HARDWARE

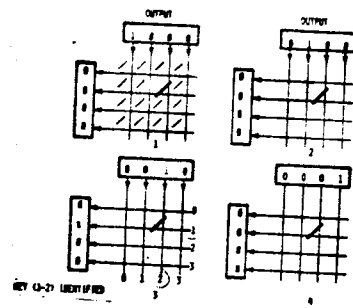


FIG. 2.3 IDENTIFICACION POR PROGRAMA

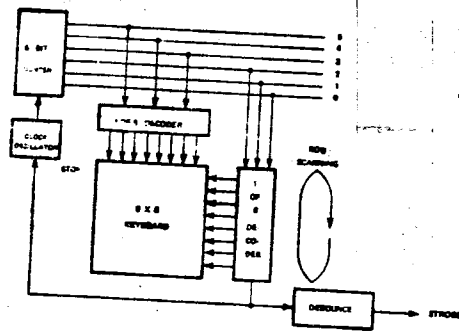
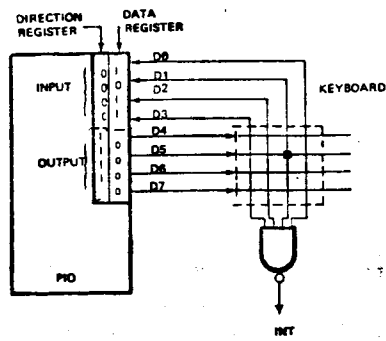
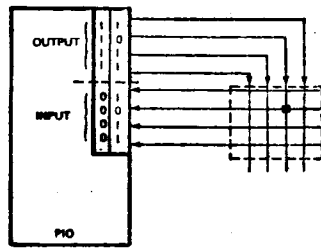


FIG. 2.4 IDENTIFICACION CON LOGICA ALAMBRADA



(1)



(2)

FIG. 2.5 IDENTIFICACION POR INVERSION DE LINEAS



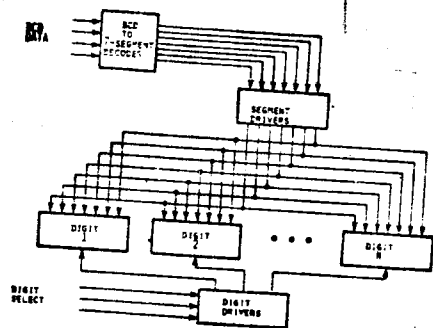


FIG. 3.1 MULTIPLEXAJE PARA DISPLAY 7 SEGMENTOS

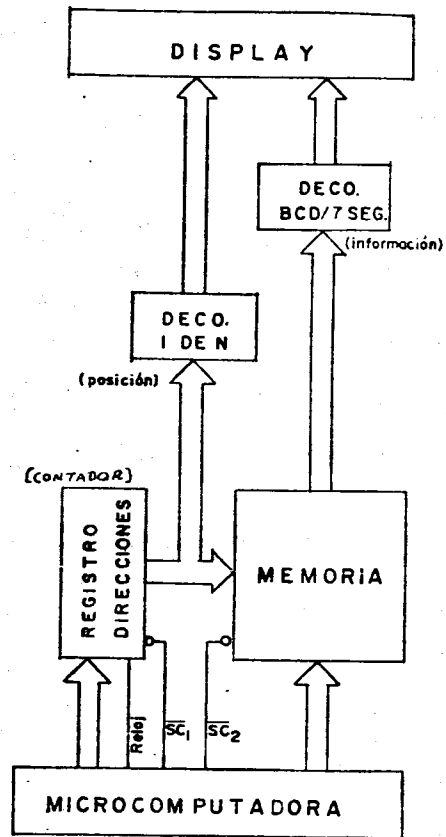


FIG. 3.2 ESCRITURA DE NUEVOS DATOS

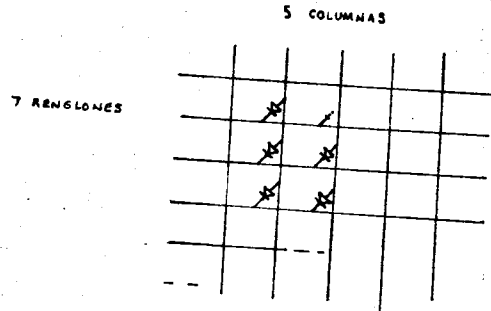


FIG. 3.3 MODULO DE MATRIZ DE PUNTOS LED 7 x 5

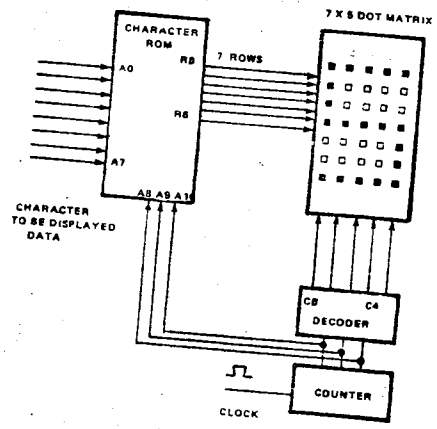


FIG. 3.4 GENERADOR DE CARACTERES

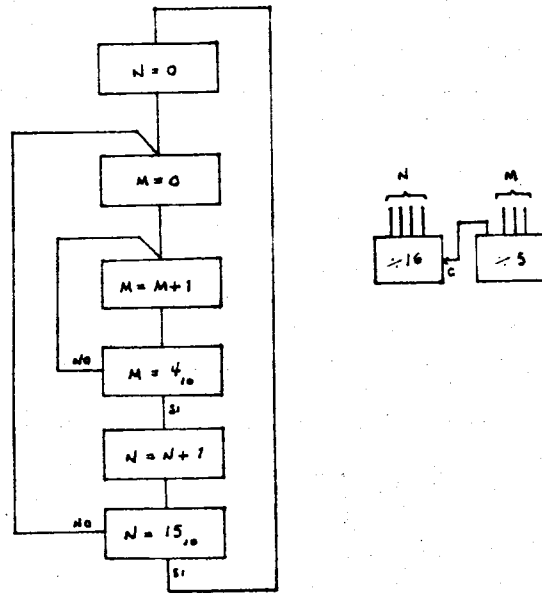


FIG. 3.5 DIAGRAMA DE FLUJO DE BARRIDO DE VARIOS MODULOS MULTIPLEXADOS.

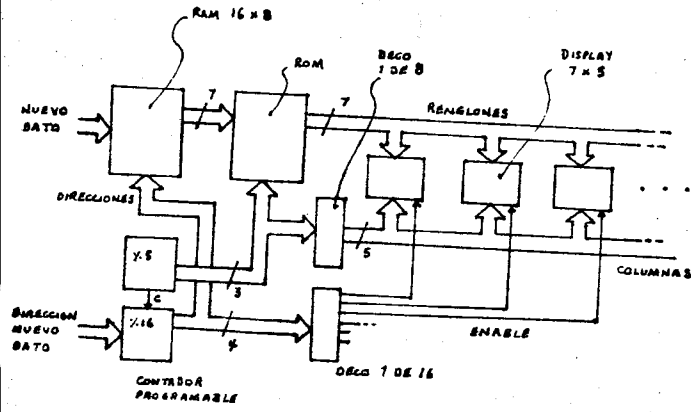


FIG. 3.6 MULTIPLEXAJE PARA DISPLAY MATRIZ DE PUNTOS

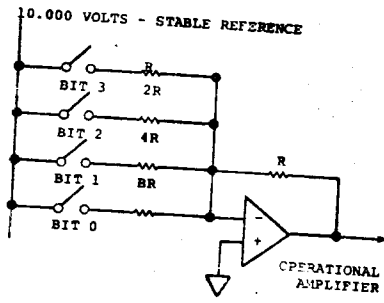


FIG. 4.1 CONVERSION D-A SIMPLE

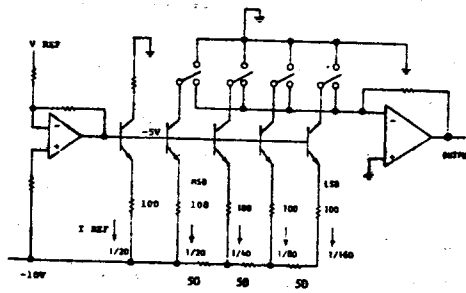


FIG. 4.2 DIAGRAMA DE UN DAC INTEGRADO

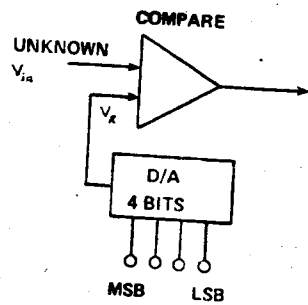


FIG. 4.3 BLOQUES BASICOS DEL ADC

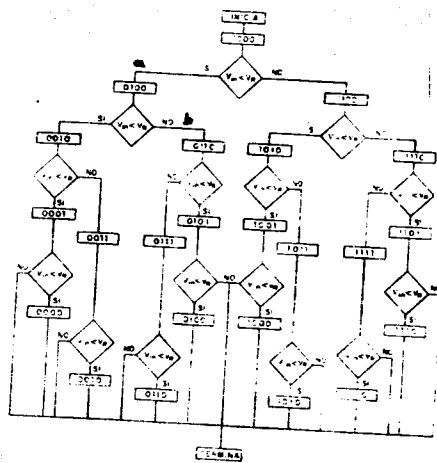


FIG. 4.4 DIAGRAMA RUTINA DE APROXIMACIONES SUCCESIVAS

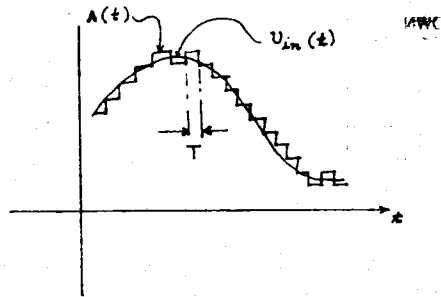


FIG. 4.5 VOLTAJES EN EL CONVERTOR SEGUIDOR

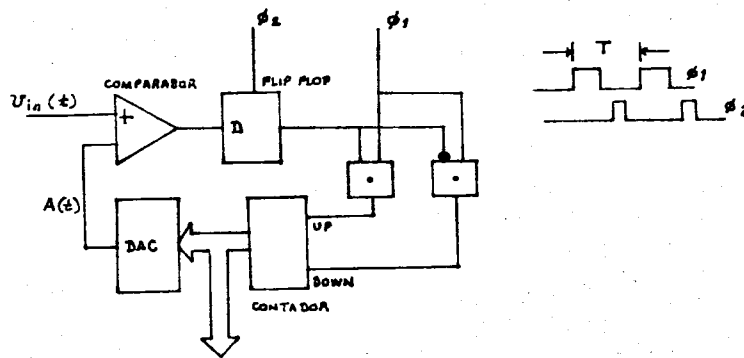


FIG. 4.6 CONVERTOR SEGUIDOR (TRACKING)



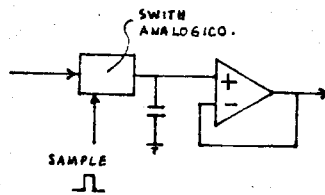


FIG. 4.7 SAMPLE &amp; HOLD

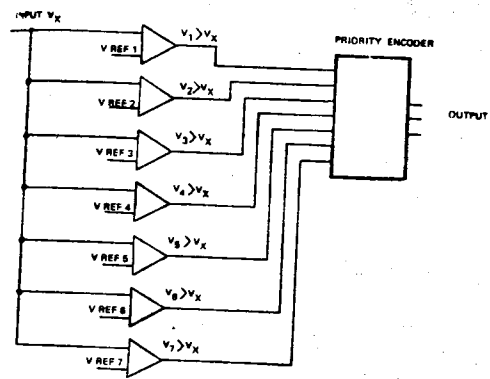


FIG. 4.8 COMPARACION DIRECTA

- SPOUND
- WRT DATA (TO COP EQUIPMENT)
- REC DATA (FROM COP)
- REQUEST TO SEND (TO COP)
- CLEAR TO SEND (FROM COP)
- DATA SET READY (FROM COP)
- DATA TERMINAL READY (TO COP)
- RING INDICATOR (FROM COP)
- RECEIVED LINE SIGNAL DETECTOR (FROM COP)
- SIGNAL QUALITY DETECTOR (FROM COP)
- DATA RATE SELECTOR (TO COP)
- TRANSMITTER TIMING (TO COP)
- RECEIVER TIMING (FROM COP)
- SECONDARY DATA REQ. REQUESTS (FROM COP)

FIG. 5.1 SEÑALES DEL ESTANDAR RS 232-C

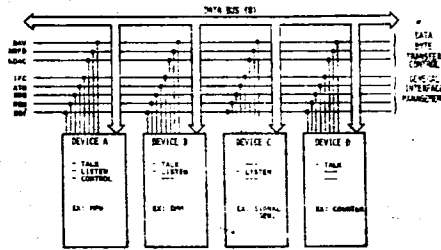


FIG. 5.2 ESTRUCTURA DEL ESTANDAR IEEE-488

**INDICADORES VISUALES FOSFORESCENTES**

### INTRODUCCION

Los despliegues fosforescentes son arreglos de caracteres alfanuméricos contenidos en una cápsula al vacío.

Estos dispositivos consisten básicamente de 3 electrodos: el cátodo, que es un filamento de tungsteno de diámetro reducido; la rejilla que es una malla de metal, y el ánodo.

El ánodo es un arreglo de 7 (ó más) segmentos eléctricamente independientes y cubiertos por fósforo fluorescente.

Todos los electrodos están aislados entre sí y se encuentran alojados en una ampolla en la cual se hace vacío hasta que las moléculas remanentes de aire no tengan ya ningún efecto (químico ó eléctrico) sobre la operación del dispositivo.

En la actualidad, el diseño de éstos indicadores se hace por lo regular con varios dígitos, conectando internamente todos los segmentos comunes. Las rejillas correspondientes a cada dígito se conectan independientemente. De esta forma se tiene un control multiplexado de todo el indicador.

### PRINCIPIO DE FUNCIONAMIENTO

El principio de funcionamiento de estos indicadores es similar a aquel de las ampolletas de emisión termoiónica (bulbos).

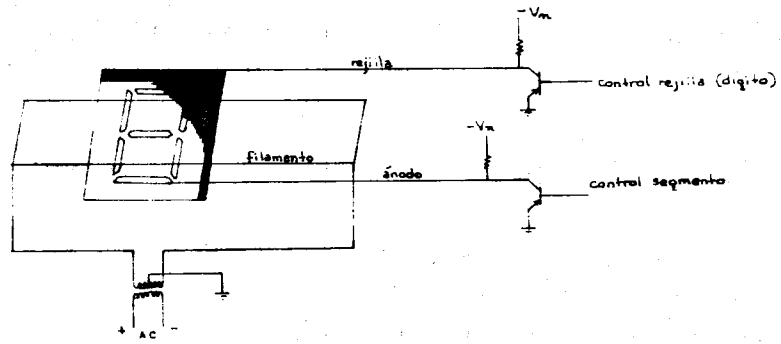
El filamento se calienta a una temperatura de aprox. 700°C. A esta temperatura emite suficientes electrones para proporcionar una luminosidad conveniente.

Cuando los electrones (termiones) que emite el cátodo se aceleran bajo la influencia de un campo eléctrico y chocan contra el fósforo en el ánodo, éste emite luz (típicamente azul ó verde).

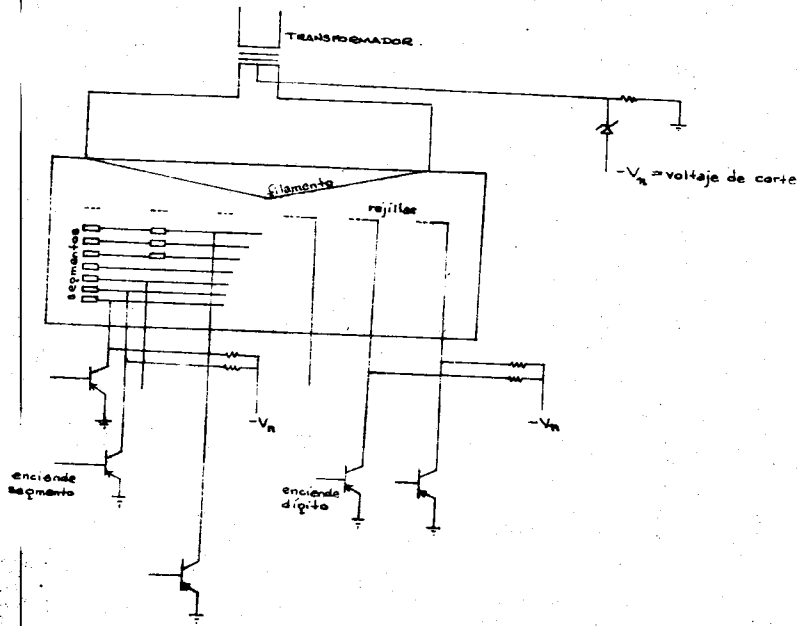
Cuando no existe diferencia de potencial en la rejilla y ánodo con respecto al cátodo, algunos electrones tienen, sin embargo, energía suficiente para alcanzar el ánodo y hacen brillar el indicador levemente.

Para evitar el efecto anterior, se polariza en forma inversa a la rejilla con respecto al filamento con algunos volts de voltaje negativo (voltaje de corte).

Cuando la rejilla y el ánodo se polarizan con voltaje positivo con respecto al filamento, el campo eléctrico resultante acelera electrones hacia la rejilla. La mayoría de éstos pasan a través de ella y solo algunos son atrapados en la malla. Los electrones que pasan se aceleran hacia el ánodo con el cual chocan. Esta transferencia de energía excita al fósforo y emite luz.



CONTROL DE UN DÍGITO



CONTROL MULTIPLEXADO DE VARIOS DÍGITOS

CONTROLADOR DE VIDEO

JUAN B. MARTINEZ

Dpto. de Computación  
Facultad de Ingeniería  
1984 U.N.A.M.



## CONTROLADOR DE VIDEO

## 1. RESUMEN:

La mayoría de los fabricantes de sistemas de microcomputación proporcionan una televisión comercial como dispositivo de salida (despliegue de texto o gráficas) del sistema (ver fig 1).

Lo anterior permite a los fabricantes ofrecer sistemas muy baratos, dado que prácticamente cualquier usuario tiene en casa una TV que puede conectar a su microcomputadora.

En este trabajo se realiza, primeramente, el análisis por bloques de una TV comercial; más tarde se describe el circuito controlador de video para escritura de textos en una televisión a través de la microcomputadora, dando criterios de diseño para definir formatos de despliegue, resolución, etc.

Finalmente, se proporciona información sobre técnicas de acceso directo de memoria (DMA) para la actualización de datos en el despliegue, así como sobre controladores LSI, que integran prácticamente en un solo circuito integrado, todas las funciones de un controlador de CRT.

Los términos 'controlador de CRT' y 'controlador de video', se usarán indistintamente a través de los siguientes párrafos, queriendo siempre significar el dispositivo de acoplamiento entre una microcomputadora y una TV comercial.

## 2. FUNCIONAMIENTO BASICO DE UNA TV COMERCIAL MONOCROMATICA:

En la fig 2, se muestra el diagrama de bloques de una TV comercial blanco y negro. Toda la sección de radio-frecuencia (RF), amplificador de FI y detector de video es muy similar a la de un radioreceptor de AM convencional. A la salida del detector se obtiene la señal de video com puesto que contiene la información que habrá de generar la imagen en la pantalla. La señal de audio viene modu lada en FM y en una banda que no interfiere con la señal de video (en el punto 4 se verán con más detalle estos conceptos).

El cinescopio consiste de un tubo de rayos catódicos (CRT) con deflexión magnética. Su funcionamiento básico es similar al de los despliegues luminosos fosforescentes (ver tema 'Indicadores visuales fosforescentes') con las siguientes observaciones (ver fig 3):

- El cátodo tiene calentamiento indirecto, es decir, el filamento calienta a un cilindro hueco metálico (cátodo) que es el que finalmente realiza la emisión de electrones.
- La tensión ánodo - cátodo es del orden de 18 KV. Esta tensión la desarrolla el transformador conocido comercialmente como 'fly back'.
- Las rejillas de enfoque proyectan el haz de electrones en un solo punto sobre la pantalla.
- El control de intensidad del punto, se hace variando el voltaje en la rejilla de control.

- Por medio de corrientes con forma diente de sierra, aplicados a las bobinas de deflexión, se realiza un barrido cíclico del haz de electrones sobre toda la superficie de la pantalla.
- Los osciladores vertical (60 HZ) y horizontal (15,750 HZ), que son los que manejan a las bobinas de deflexión, funcionan aún en ausencia de señales de sincronía (SYNC H y SYNC V). Lo anterior evita que el punto quede inmóvil-en ausencia de señal de video-y quemé la pantalla.
- La deflexión es magnética (no electrostática como en los osciloscopios). Esto permite obtener deflexiones muy pronunciadas a bajos voltajes en las bobinas de deflexión.

### 3. FORMACION DE LA IMAGEN:

Toda la información necesaria para la formación de la imagen viene contenida en la señal de video compuesto. El bloque separador de sincronía obtiene de esta señal, otras 3: video, SYNC H y SYNC V. Estas 2 últimas se alimentan (y sincronizan) a los osciladores vertical y horizontal que finalmente manejarán a las bobinas de deflexión en sincronía con la señal de video.

En las transmisiones comerciales, cada imagen está formada por 2 cuadros consecutivos entrelazados. Cada cuadro se forma con el barrido completo de la pantalla a una frecuencia de 60 HZ.

Las imágenes (2 cuadros), por tanto, se reproducen a una frecuencia de 30 HZ.

El barrido de un cuadro, se forma con 262 y media líneas consecutivas, mientras que cada imagen se forma con 525 líneas. La frecuencia de línea es entonces de 15,750 HZ.

En la fig 4, se muestra la manera como se forma la imagen entrelazada, suponiendo solo 10 líneas horizontales de barrido para facilitar la comprensión. En una pantalla comercial, la frecuencia para la bobina de deflexión horizontal debe ser 262 y media veces mayor que la correspondiente a la bobina de deflexión vertical.

El regreso del haz de electrones (que está sincronizada con las señales de SYNC H y SYNC V) del punto final de la línea o cuadro, al punto inicial, se le conoce como 'fly back'.

Este efecto no se ve en la pantalla porque durante ese período la señal debe 'blanquearse', es decir, debe estar a nivel de negro.

#### 4. LA SEÑAL DE VIDEO COMPUESTO:

Como ya se mencionó, la señal de video compuesto contiene la siguiente información:

- a) Cuándo regresa el punto a la posición inicial en la pantalla (SYNC V)
- b) Cuándo regresa el punto a la posición inicial en la línea (SYNC H)
- c) Qué intensidad tiene el punto a cada instante (video).

En la fig 5, se muestra la forma de esta señal y la correspondiente imagen, generando sólo niveles de blanco o negro durante ciertos lapsos. La señal de SYNC H, se detecta porque su nivel es mayor al resto de la señal. La señal de SYNC V es, por otro lado, de mayor duración que la correspondiente de SYNC H.

El ancho de banda estándar de una señal de video compuesto es de 4 MHz. Sin embargo, el ancho de banda asignado comercialmente a un canal de TV es de 6 MHz, conteniendo información de audio y video.

En la fig 6, se muestra el espectro de frecuencias de la señal para el canal 3. La señal de audio se modula en FM a 65.75 MHz y ocupa la parte superior del espectro. La señal de video compuesto se modula en AM (banda lateral única) a 61.25 MHz. El ancho de banda del canal queda, por tanto, de 6 MHz.

#### 5. ESCRITURA DE TEXTO EN LA TV

La escritura de texto, se obtiene al generar una señal de video compuesto que reproduzca puntos blancos sobre la pantalla.

Cualquier letra o número se forma partiendo de una matriz de puntos (ver fig 7).

Es posible - por medio de un controlador secuencial - generar una señal de video compuesto que defina en qué lugar se iluminan los puntos en cada línea de la pantalla. Estos puntos formarán los caracteres al barrerse todas las líneas sucesivamente.

Es necesario, sin embargo, el 'refrescamiento' continuo de la misma información. Si se escribiese un texto sobre la pantalla una sola vez, el observador no vería absolutamente nada puesto que cada cuadro se completa en un lapso de solo 16.6 mseg.

Cabe aclarar que el método de 'entrelazado' empleado en las transmisiones comerciales no se emplea para la escritura de texto, puesto que los puntos siempre aparecerán en las líneas impares, mientras las líneas pares quedarán siempre en negro. De esta forma se 'refrescan' a una frecuencia de 60 HZ, solo las líneas impares obteniéndose una imagen clara y sin parpadeo.

#### 6. FORMATO DE LA IMAGEN:

La velocidad del haz de electrones no es constante durante todo el barrido, sino que presenta aceleración sobre todo al inicio de cada línea. Esto hace que el texto se distorsione en los extremos de la pantalla, apareciendo caracteres más alargados que los del centro.

Para evitar esta distorsión, se asigna una zona de escritura que ocupa usualmente  $9/16$  del área total de la pantalla:  $3/4$  de la longitud horizontal y  $3/4$  de la magnitud vertical.

Para el diseño del controlador, debe partirse de un formato para la zona de escritura, y de una matriz de puntos específica para formar cada carácter.

Un formato muy generalizado es de 384 caracteres para la zona de escritura: 32 columnas y 24 renglones para caracteres formados en matriz de puntos de 7 x 5 (ver fig 7).

Una vez seleccionado el formato y el área de la zona de escritura, se realiza el cálculo del ancho de banda de la señal de video:

Frecuencia de línea : 15,750 HZ  
 Período de línea : 62.5  $\mu$ s  
 Período de línea en la zona de escritura: 62.5  $\mu$ s x (3/4) = 46.8  $\mu$ s

Durante estos 46.8  $\mu$ s podrán aparecer en cada línea:

$$N_p = 32 \text{ caracteres} \times (6 \text{ puntos/caracter}) = 192 \text{ puntos}$$

Y el período transcurrido entre un punto y el siguiente:

$$\tau_p = 46.8 \mu\text{s} / 192 = 243 \text{ mseg.}$$

Lo que resulta en una frecuencia máxima de:

$$f_p = (1/243 \times 10^{-9}) \text{ HZ} = 4.10 \text{ MHZ.}$$

Que corresponde al ancho de banda asignado a la señal de video compuesto.

En el sentido vertical se dispone de:

$$262 \times (3/4) = 196 \text{ líneas}$$

Y se necesitan, para el formato elegido:

$$24 \text{ caracteres} \times (8 \text{ líneas/caracter}) = 192 \text{ líneas}$$

Por lo que la zona de escritura seleccionada cumple con los requerimientos de ancho de banda y formato especificados.

7. **DIAGRAMA DE BLOQUES DEL CONTROLADOR DE VIDEO:**

El controlador consta de los siguientes bloques (ver fig 8):

- a) Memoria RAM: la memoria es necesaria para el refrescamiento continuo de la información. Para el formato especificado se necesitan:  $32 \times 24 = 768$  bytes de memoria RAM. En cada localidad se almacena el código ASCII del carácter deseado. Dicho carácter debe aparecer en la pantalla en la misma posición que en memoria (considerando la organización matricial prevista).
- b) Generador de caracteres: al igual que en los sistemas de encendido multiplexado de varios módulos de matriz LED de puntos (ver tema 'Periféricos para microprocesadores'), el generador de caracteres - que es una memoria PROM - proporciona el código de puntos correspondiente al carácter ASCII y el renglón seleccionado (ver fig 9).



c) Registro de corrimiento: el registro de corrimiento proporciona la salida en serie del código de puntos proveniente del generador de caracteres. La señal obtenida, junto con las señales de SYNC V y SYNC H habrán de formar la señal de video compuesto.

d) Control secuencial: este módulo controla las direcciones de la RAM, la selección del renglón del generador de caracteres y la carga y corrimiento del registro de corrimiento.

Por otro lado, el control debe generar también las señales de SYNC H y SYNC V, así como la señal de 'blanqueo' del video (BLANK). En el siguiente punto (8), se analiza con detalle el funcionamiento de este circuito.

e) Circuito de acoplamiento: este bloque tiene como funciones la formación de la señal de video compuesto, sumando la señal de video, SYNC H y SYNC V, y el ajuste de niveles de voltaje e impedancia de salida para hacerlos compatibles con la TV comercial.

### 3. EL CONTROL SECUENCIAL:

El control secuencial consta de una serie de contadores y compuertas lógicas acoplados para generar las señales correspondientes.

La fig 10 muestra varios bloques en cascada que, partiendo de la frecuencia base de 4.10 MHZ, generan las señales requeridas en el diagrama de la fig 8.

En el circuito de la fig 10 se distinguen 2 tipos de módulos: los bloques contadores y los bloques de control. Los bloques de contadores mostrados cuentan, primero, columnas punto, columnas caracter, renglones punto y, finalmente, renglones caracter, (ver fig 7).

Cabe recordar que la zona de escritura se limita a 9/16 del área total de barrido. Los bloques de control deben inhibir o habilitar estos contadores, según convenga de tal forma de hacer coincidir sus salidas con la posición del punto en la zona de escritura. También deben generar periodos de 'blanqueo' de la señal de video en las zonas externas a la zona de escritura.

Para información completa sobre el diagrama electrónico con todos sus detalles, se recomienda la ref 3.

#### 9. ACCESO DIRECTO DE MEMORIA:

Cuando 2 o más dispositivos procesadores comparten la información de una misma memoria (en lectura y/o escritura) se establece un sistema de 'Acceso directo de memoria' o DMA.

Lo anterior significa que cualquiera de los procesadores puede leer y/o escribir en ella sin interferencia de los demás.

En el problema particular de este trabajo, se tiene una memoria RAM de refresco compartida por el controlador de video y por el microprocesador (ver fig 8).

El problema encuentra varias soluciones; el empleo de un controlador LSI para DMA no parece ser una solución sencilla ni barata para el caso que nos ocupa.

El uso de una memoria RAM con buses de entrada y salida de datos separados (por ejemplo, la 2102, de 1024 x 1 bytes) permite la escritura de datos por parte del microprocesador (para actualizar la información), y la lectura de datos por parte del controlador (para el refresco de la pantalla), sin interferencia de ambos en cuanto a líneas de datos se refiere.

La compartición del bus de direcciones debe resolverse con compuertas de tercer estado que otorguen el control al microprocesador, cuando éste lo requiera. La alta impedancia hacia el controlador de video (> 50 K) hará que su bus de direcciones permanezca virtualmente desconectado durante este lapso.

De preferencia debe aprovecharse el período de retorno vertical (aprox 500  $\mu$ s), para escribir nuevos datos en la memoria. De esta forma se elimina el problema de ruido generado en la pantalla al interrumpirse el funcionamiento del controlador de video durante el barrido sobre la zona de escritura.

De esta manera, el controlador de video es visto por el microprocesador como parte de su memoria RAM, siendo el refrescamiento totalmente transparente a aquél.

#### 16. CONTROLADORES LSI:

Existen en el mercado todo tipo de controladores LSI con diferentes posibilidades y funciones. La mayoría son programables y, aparte de realizar las tareas básicas ya expuestas, algunos permiten también funciones especiales como:

- Formato programable
- Control de cursor
- 'Scrolling' (todos los renglones se desplazan hacia arriba dejando en blanco el último renglón)
- Manejo de tonos de gris
- Inversión de video
- Manejo de gráficas de alta resolución
- Parpadeo de cualquier caracter

En la fig 11, se muestra el controlador 8257 que forma, junto con otros circuitos adicionales, un módulo controlador de CRT completo.

El circuito MTX 16-32 es un módulo controlador de CRT para escritura de texto, que se acopla directamente a la microcomputadora por un lado, y a la TV por el otro (ver especificaciones técnicas del MTX 16-32), sin necesidad de ningún circuito adicional.

El microprocesador 've' a este módulo como parte de su memoria, y, al escribir en ella los códigos ASCII deseados, se generan en la pantalla automáticamente los caracteres correspondientes.

El circuito cuenta solo con la función especial de 'parpadeo' programable para cada carácter. Sin embargo, funciones como el control del cursor y el 'scrolling', pueden realizarse el propio microprocesador, manipulado convenientemente los datos en la memoria de refresco.

#### 11. ACOPLAMIENTO A LA TV:

El acoplamiento del controlador descrito a una TV comercial se puede hacer de 2 maneras:

a) Entrada por antena: en este caso, la señal de video compuesto debe modularse en RF, sintonizada a un canal prefijado, antes de acoplarla a la entrada de antena. Esta solución tiene la ventaja que no es necesario abrir la TV ni tener ningún conocimiento de electrónica para realizar el acoplamiento, por lo que es preferida por los fabricantes de microcomputadoras.

b) Entrada por video compuesto: el acoplamiento debe hacerse directamente sobre la salida del detector de video. Usualmente no es necesario inhibir la señal de video proveniente de la antena porque es demasiado pequeña en comparación con la señal del controlador. Sin embargo, ha de tenerse cuidado al hacer este acoplamiento en televisores con chasis 'caliente' (sin transformador de entrada), por el peligro que representa tener la tierra del circuito conectada a una de las terminales de la red de 127 AC.

En la ref 3 se dan detalles en cuanto al diseño de ambos acoplamientos.

## 12. CONCLUSIONES:

El campo de controladores de video es muy extenso y con una amplia gama de aplicaciones. Lo que aquí se describió constituye solo los principios del funcionamiento del controlador de CRT para escritura de texto. Estos principios, sin embargo, son igualmente aplicables a controladores orientados al despliegue de gráficas de alta resolución, siendo la diferencia la forma en que se interpreta la información contenida en la RAM de refresco.

Todos los efectos y 'trucos' visuales como son los objetos tridimensionales, objetos en movimiento, etc., se generan mediante la manipulación adecuada de los datos almacenados en la RAM de refresco del controlador, por parte del microprocesador.

Gracias a la relativa lentitud del ojo humano en percibir y seguir un objeto en movimiento (en relación con la alta velocidad del microprocesador para procesar y cambiar los datos en la RAM), pueden programarse complicados juegos de video (que han dado origen a una gran industria en los últimos años), en donde el usuario toma parte a través de teclados, sensores analógicos, etc.

## 13. BIBLIOGRAFIA:

Ref 1: 'Basic Television', Bernard Grob.  
Mc. Graw-Hill, Kogakusha

Ref 2: 'Microprocessor Interfacing Techniques'  
Austin Lesea - Rodney Laks,  
SYBEX

Ref 3: 'TV Typewriter Cookbook'  
Don Lancaster,  
HOWARD W. SAMS

Ref 4: 'Mate microprocessor with CRT displays'  
Electronic Design, Sept. 1977

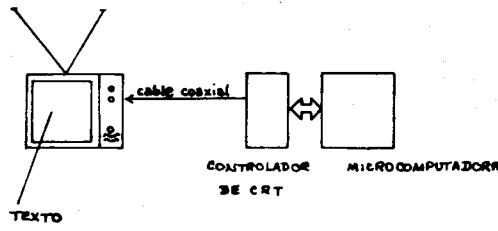


FIG 1 : ACOPLAMIENTO TV - MICROCOMPUTADORA

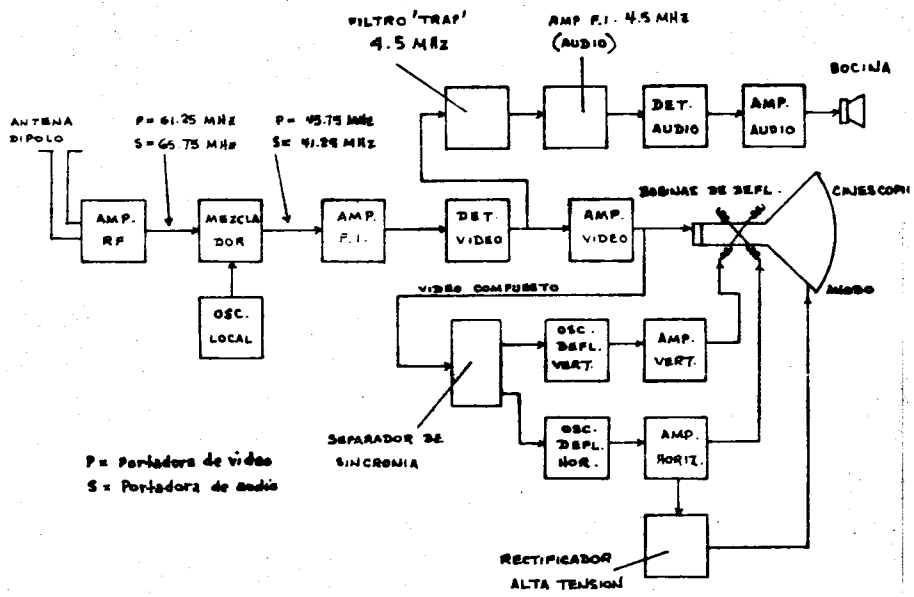


FIG 2 : DIAGRAMA DE BLOQUES DE UNA TELEVISION COMERCIAL.



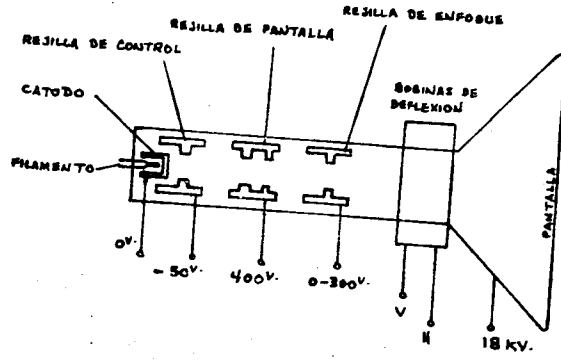


FIG. 3 : CINESCOPIO DE UNA TV. COMERCIAL

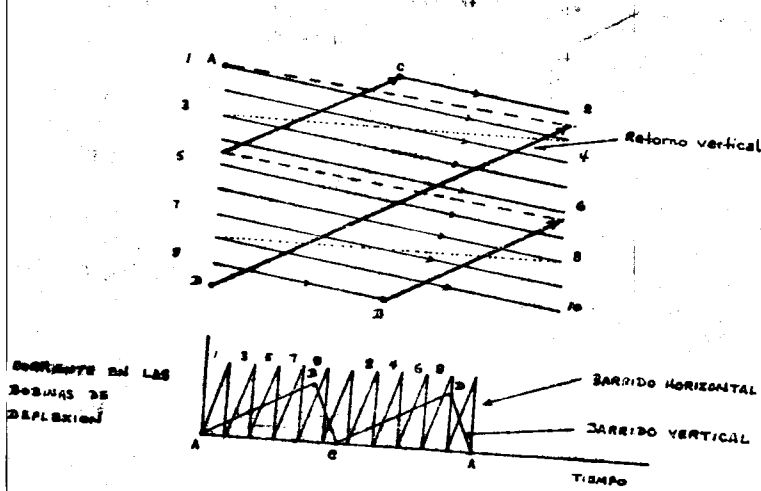


FIG. 4 : FORMACION DE UNA IMAGEN ENTRELAZADA.

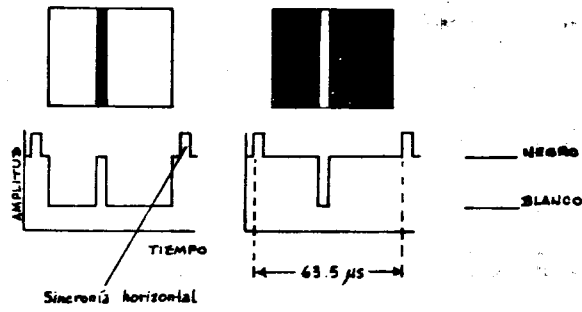


FIG 5 : SEÑAL DE VIDEO COMPUESTO Y SU CORRESPONDIENTE IMAGEN

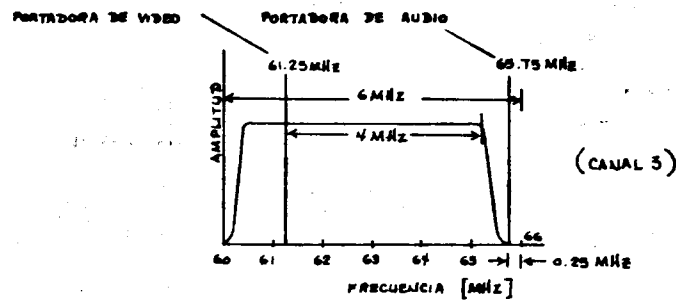


FIG 6 : CANAL ESTANDAR PARA TRANSMISION POR TELEVISION.

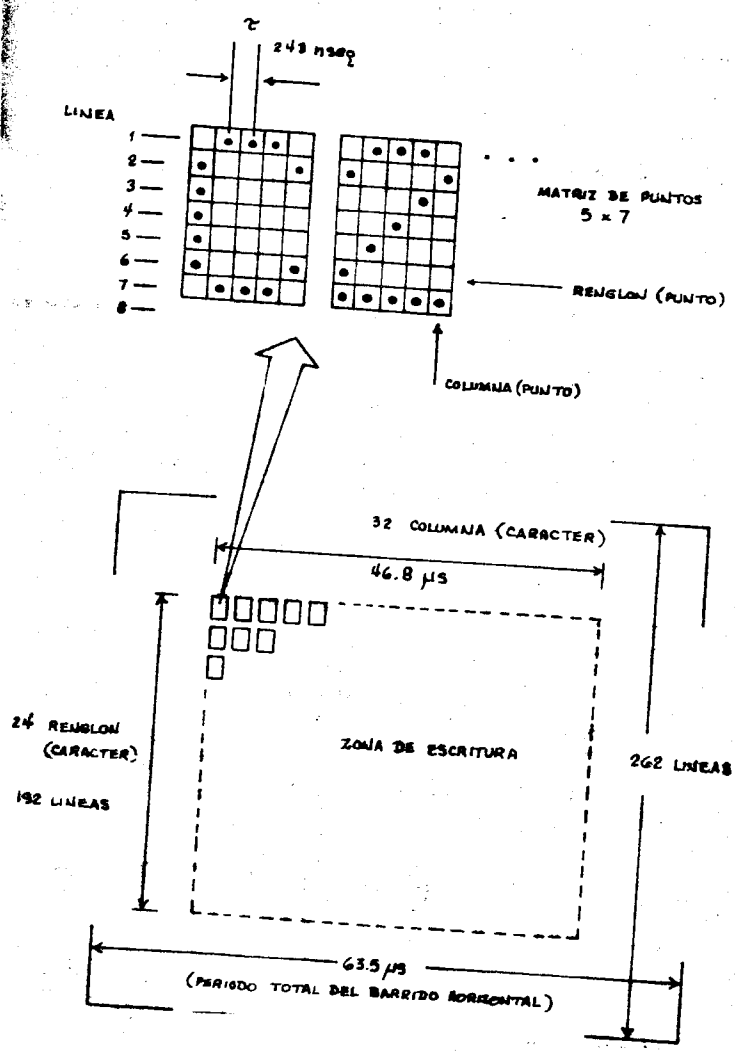


FIG 7 : FORMATO DE LA IMAGEN PARA ESCRITURA DE TEXTO.

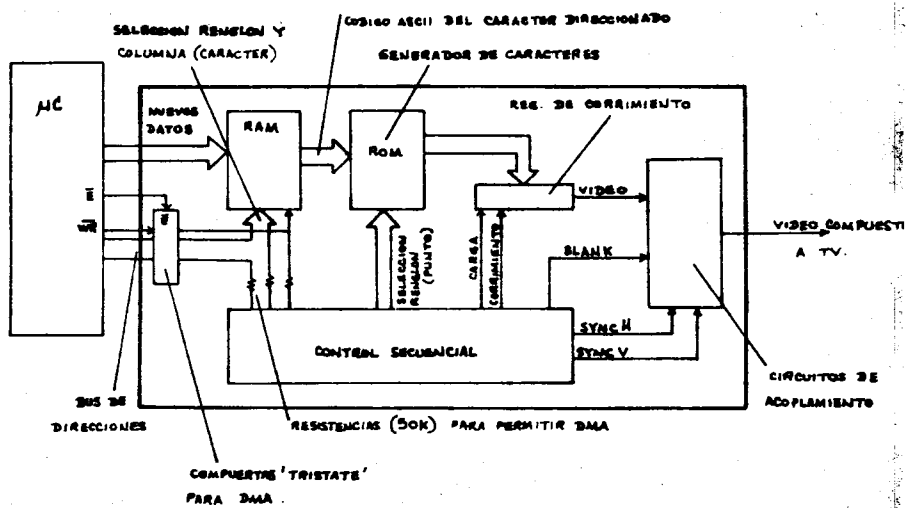


FIG 8 : DIAGRAMA DE BLOQUES DEL CONTROLADOR DE VIDEO.

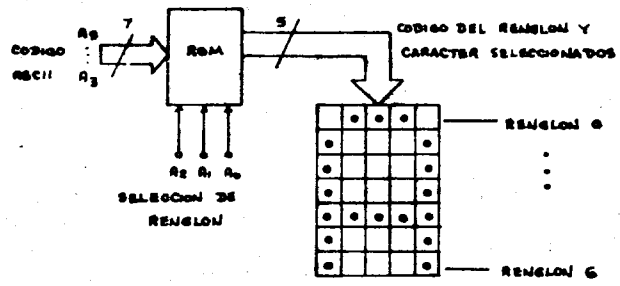


FIG 9 : GENERADOR DE CARACTERES

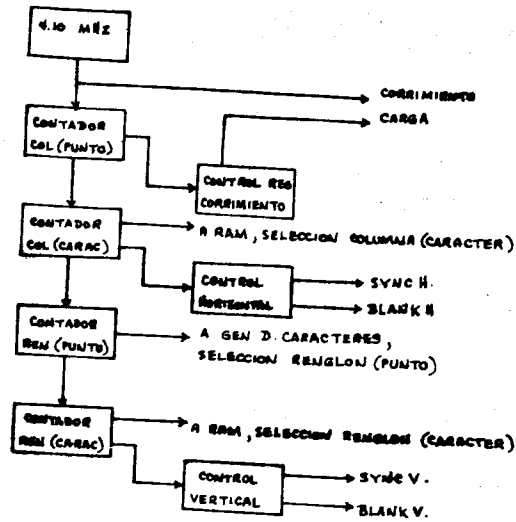


FIG 10: CONTROL SECUENCIAL PARA EL CONTROLADOR DE VIDEO

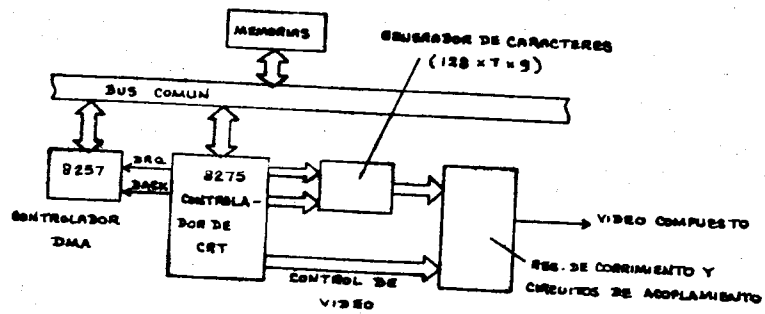


FIG 11: INTEGRACION DE UN CONTROLADOR DE CRT EMPLEANDO EL CIRCUITO 8275 DE INTEL.

## 1. INTRODUCCION

Con el desarrollo de las computadoras y equipos digitales, surgió la necesidad de su intercomunicación a grandes distancias.

Y la forma más sencilla (transmisión en banda base) era también la más costosa por requerirse instalación de líneas especiales de punto a punto.

Por el contrario, la infraestructura telefónica ya existente hacía factible la intercomunicación entre prácticamente cualesquiera puntos.

Las líneas telefónicas, sin embargo, no fueron originalmente diseñadas para transmisión digital: su ancho de banda - si bien satisface los requerimientos de comunicación por voz - está limitado entre las frecuencias de 300 y 3,000 Hz, además de ser susceptibles de interferencia.

Estos problemas condujeron al diseño de acopladores entre los equipos digitales y las líneas telefónicas - módems - con velocidades de transmisión que la actual tecnología ha llevado hasta 9,600 bits por segundo.

## 2. LINEAS TELEFONICAS

Una línea telefónica puede considerarse como un filtro paso banda. Cada línea tiene sus propias características de amplitud (ver fig. 1) y retraso, variables dentro de ciertos límites.

Las líneas introducen principalmente 2 tipos de distorsión en la señal modulada:

- Distorsión de amplitud: se explica por la característica irregular del espectro de amplitud que hace que ciertas armónicas se atenúen más que otras.

- Distorsión de fase: se explica por la condición no lineal de la curva de fase que introduce retrasos desiguales en las diferentes armónicas.

Los efectos anteriores no son significativos a bajas velocidades de transmisión (hasta 600 bauds). Pero a velocidades mayores es indispensable el uso de igualadores de amplitud y fase en los módems para contrarrestar dichos efectos.

Existen 2 tipos de líneas telefónicas:

- Privadas: simplemente son un par (ó 2 pares) de hilos que comunican 2 puntos. Estas líneas se rentan a la compañía de teléfonos.
- Conmutadas: estas líneas comunican 2 puntos a través de la central telefónica. El enlace se hace de módem a módem como si se tratara de una llamada de comunicación por voz.

Por lo tanto pueden comunicarse entre sí equipos ubicados en cualquier punto cubierto por la red telefónica.

### 3. MODEMS

Una señal digital transmitida directamente sobre una línea telefónica se distorsiona principalmente por el filtrado de bajas frecuencias.

El módem tiene varias funciones: primeramente, al modular la señal se traslada su espectro a una banda poco susceptible a la distorsión. Por otro lado, las modulaciones FSK y PSK (que son las más usadas) proporcionan cierta protección al ruido aditivo.

Además, la modulación misma permite la transmisión 'FULL DUPLEX' por 2 hilos, porque los espectros de las señales transmitida y recibida no se traslapan. (ver fig. 3).

Las frecuencias estándar de transmisión y recepción usadas en los módems comerciales varían de acuerdo a la norma vigente en cada país: en los Estados Unidos se usa la norma Bell mientras México y algunos países europeos emplean la recomendación CCITT.

En la fig. 3 se muestra la asignación de las bandas de frecuencia en el estándar CCITT V.21, para módems de baja velocidad.

Existen comercialmente 2 tipos de módems:

- Módems para línea privada:

Simplemente se conectan a ambos extremos de la línea privada. Los hay 'FULL DUPLEX' ó 'HALF DUPLEX' a diferentes velocidades.

- Módems para línea conmutada:

La comunicación puede establecerse de diferentes formas; hay módems que requieren de un teléfono estándar, en donde se marca el número deseado ó los hay con conexión automática.

Hay módems con respuesta manual ó respuesta automática.

Por otra parte en cuanto al modo de transmisión, pueden ser asíncronos y síncronos.

Los asíncronos modulan y demodulan mensajes con formato asíncrono. Por su simplicidad son los más usados a bajas velocidades.



Los módems síncronos, que son más eficientes, requieren electrónica más complicada (sobre todo en los demoduladores) por la necesidad de derivar la señal de sincronía de la señal modulada. Por lo general emplean modulación de fase, llegando a manejar velocidades de 9,600 bits por segundo.

En la fig. 4 se muestra una tabla que especifica el tipo de modulación y la velocidad alcanzada por diferentes módems síncronos.

#### 4. EL ESTANDAR RS232-C

Prácticamente todos los módems - en Europa y América - utilizan el estándar RS232-C (cuyo equivalente es el CCITT V.24) para su acoplamiento con los equipos digitales.

El estándar RS232-C define características eléctricas, mecánicas y funcionales de 25 señales.

Sin embargo el número de señales utilizadas varía de acuerdo al tipo de módem y su aplicación.

En módems asíncronos, para líneas privadas ó conmutadas se asignan las siguientes señales:



| Pin. | Designación | Función                  | Abreviatura | Sentido<br>E.a. ↔ módem |
|------|-------------|--------------------------|-------------|-------------------------|
| 1    | AA          | Tierra de protección     | - -         | NO                      |
| 7    | AB          | Tierra de señal          | GND         | NO                      |
| 2    | BA          | Datos transmitidos       | XMIT        | +                       |
| 3    | BB          | Datos recibidos          | RCV         | +                       |
| 4    | CA          | Solicitud de transmisión | RTS         | +                       |
| 5    | CB          | Listo para transmitir    | CTS         | +                       |
| 8    | CF          | Detector de portadora    | CD          | +                       |
| 20   | CD          | Terminal de datos listo  | DTR         | +                       |
| 22   | CE          | Indicador de llamada     | RI          | +                       |

Para la aplicación más simple, puede establecerse comunicación 'FULL DUPLEX' sobre una línea privada usando solo GND, XMIT y RCV.

RTS y CTS se utilizan en 'HALF DUPLEX'. La primera habilita el transmisor del módem y la segunda indica al equipo digital el momento en que puede iniciar el mensaje.

DTR se usa en líneas conmutadas e indica al módem que el equipo digital está listo para originar o recibir llamadas.

La señal RI se activa cuando el módem recibe una llamada. RI es generada por un circuito detector integrado al módem.

En la fig.2 se muestran las señales usadas para un acoplamiento con línea conmutada.

## 5. BIBLIOGRAFIA

'Technical Aspects of Data Communication', John E. Mc Namara.

'Telecommunication System Engineering', Roger L. Freeman.

'Digital Telephony', John C. Bellamy.

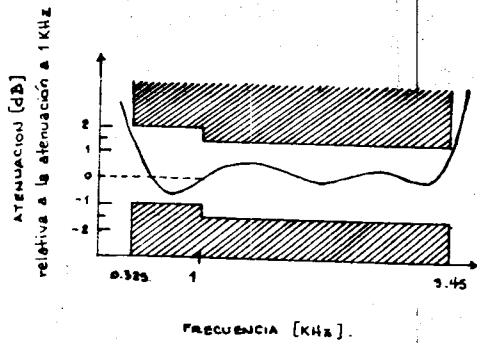


FIG. 1 : RESPUESTA EN FRECUENCIA TIPICA DE UNA LINEA TELEFONICA. LA ZONA ACHURADA SEÑALA LOS LIMITES OBSERVADOS ESTADISTICAMENTE.

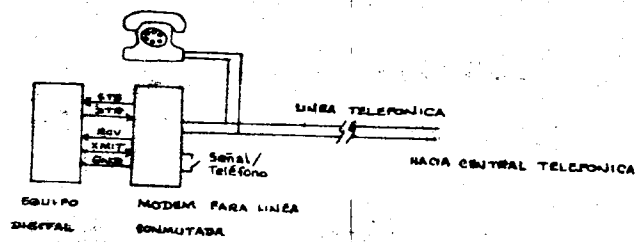


FIG 2 : ACOPLAMIENTO MODEM - EQUIPO DIGITAL EN TRANSMISION POR LINEA CONMUTADA.

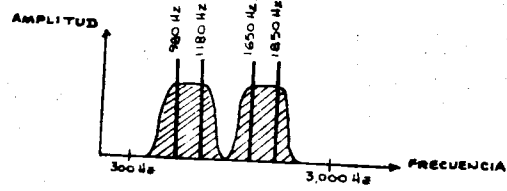


FIG 3 : ASIGNACION DE FRECUENCIAS EN UN MODEM DE 300 BAUDS , FULL DUPLEX , DE ACUERDO AL ESTANDAR CCITT V.21

| VELOCIDAD<br>BITS POR<br>SEGUNDO | VEL. DE<br>MODULACION<br>[BAUDS] | MODULACION                          | BITS<br>POR<br>SIMBOLO | ANCHO DE<br>BANDA [HERTZ] |
|----------------------------------|----------------------------------|-------------------------------------|------------------------|---------------------------|
| 2400                             | 1200                             | DIFERENCIAL, 4 FASES                | 2                      | 1,200                     |
| 2400                             | 800                              | DIFERENCIAL, 8 FASES                | 3                      | 900                       |
| 4800                             | 1600                             | DIFERENCIAL, 4 FASES                | 3                      | 1,600                     |
| 9600                             | 4800                             | DIFERENCIAL, 2 FASES,<br>2 NIVELES. | 2                      | 2,400 *                   |
| * USA IGUALADOR ADAPTIVO         |                                  |                                     |                        |                           |

FIG 4 : CARACTERISTICAS DE MODEMS SINCRONOS DE ALTA VELOCIDAD .

## M-200 MODEM ASINCRONO A 200 BAUDS

### 1. RESUMEN

Construcción de un modem de 200 bauds para línea telefónica privada, 2 hilos full duplex, asíncrono para aplicaciones de tiempo real.

El diseño minimiza la complejidad del circuito con el consiguiente ahorro en espacio y costo.

El modem cubre en gran parte las recomendaciones CCITT V.21 y V.24.

### 2. DESCRIPCION GENERAL

El diagrama de bloques se muestra en la fig. 1. Cada bloque tiene sus circuitos correspondientes en el diagrama electrónico de la fig. 2:

#### Circuito 1

(MC14412) concentra las funciones de modulación y demodulación digitales con las siguientes características:

- El demodulador requiere en su entrada una señal digital con niveles de 0 a 5V, a las frecuencias especificadas en la norma CCITT V.21 (Canal A ó Canal B).

La señal se sintetiza digitalmente con un rango dinámico de 18 db.

#### Circuito 2

Filtro pasa bajas de segundo orden, con frecuencia de corte  $f_c \approx 5\text{KHz}$ . Su función es filtrar la señal sintetizada digitalmente y darle amplitud adecuada.

### Circuito 3

Hace las funciones de circuito híbrido i.e. presenta una impedancia de 500  $\Omega$  hacia la línea telefónica y suma las señales de transmisión y recepción en el punto A.

### Circuitos 4, 5 y 6

Forman las 3 etapas del filtro de 6°orden pasabandas Chebyshev y cuya frecuencia central se ajusta a la señal de recepción (Canal A ó Canal B).

### Circuito 7

Es un comparador cuya salida se satura positiva o negativamente en sincronía con los cruces por cero de la señal de entrada.

De esta forma se cumplen los requerimientos del demodulador del circuito 1.

### Circuitos 8 y 9

Acoplan la entrada y salida digitales del circuito 1 con el equipo terminal de datos.

## 3. CONEXION A LA LINEA TELEFONICA

La línea telefónica se conecta directamente a los puntos indicados en el diagrama electrónico (Fig. 2).

Los modems deben complementarse a ambos extremos de la línea, es decir, un modem canal A transmite a otro canal B y viceversa.

La fig. 3 muestra las frecuencias de operación de transmisores y receptores en un enlace "full duplex" y con 2 hilos.

#### 4. PRUEBAS

Las pruebas al M-200 se realizaron en una línea privada con extremos en el propio Instituto de Ingeniería y una estación ubicada a una distancia de 10 kms.

La fig. 4 muestra a bloques el equipo instalado en cada extremo.

La función de la repetidora es regenerar la señal y retransmitirla.

La prueba consiste en la transmisión de mensajes con códigos consecutivos a una velocidad de 150 bauds y utilizando un formato asíncrono con 8 bits de información y 2 "stop bits".

Cada palabra recibida en el puesto central se compara con la antes transmitida. Cualquier disparidad se detecta y despliega en la terminal.

El resultado de las pruebas fue satisfactorio, obteniéndose una probabilidad de error menor a  $10^{-6}$ .

#### APENDICE 1. Especificaciones técnicas del M-200

Modulación: por corrimiento de frecuencias (FSK)

Velocidad de transmisión: 0 a 200 bauds.

Modo de transmisión: Full Duplex, 2 hilos

Formato: Asíncrono.

Frecuencias: Canal A frecuencia media: 1080 Hz

Canal B frecuencia media: 1750 Hz

El corrimiento es  $\pm 100$  Hz. En cada canal la frecuencia más alta corresponde al símbolo 0.



La máxima tolerancia para todos los casos es de  $\pm 6$  Hz.

Nivel de portadora: 0dbm.  $\pm 20\%$

Acoplamiento digital: cuenta con las siguientes señales con niveles de  $\pm 12V$ .

Pin 3: Recepción de datos

Pin 2: Trasmisión de datos

Pin 7: Retorno

Acoplamiento a la línea telefónica: sólo para líneas privadas, M200 tiene impedancia de salida de 600  $\Omega$ .

Probabilidad de error: menor de  $10^{-6}$ .

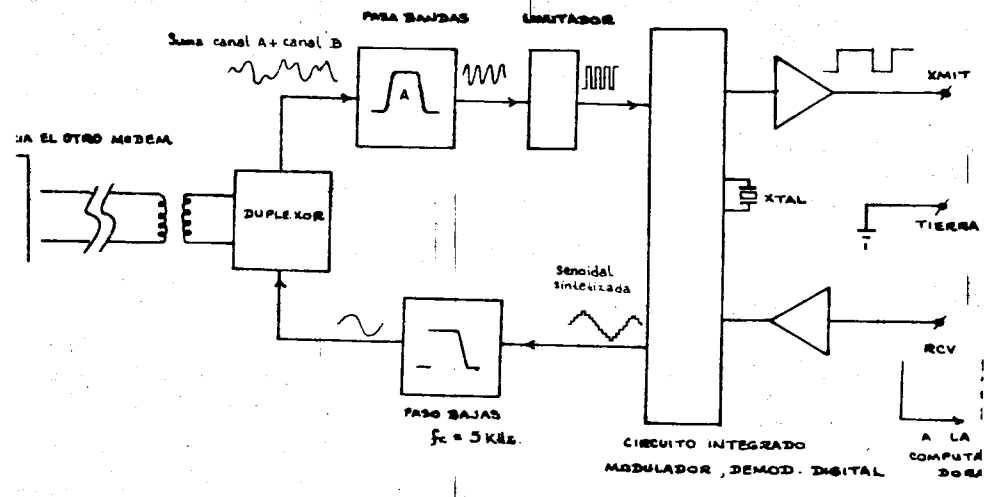


FIG 1: DIAGRAMA DE BLOQUES

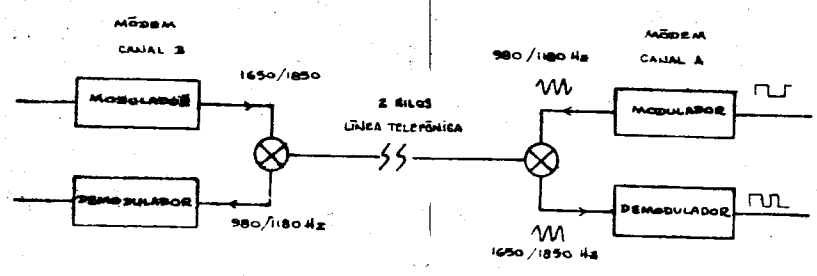


FIG 3: ASIGNACION DE FRECUENCIAS



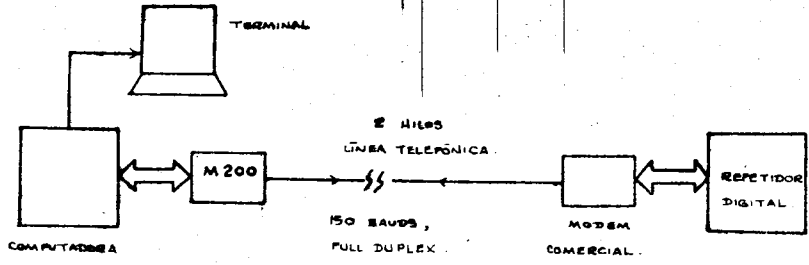


FIG 4. PRUEBAS AL M-200

"CARACTERISTICAS ESPECIFICAS DE MICROPROCESADORES  
DE 16 BITS"

|    |                                      |   |
|----|--------------------------------------|---|
| 1. | RESUMEN                              | 1 |
| 2. | INTRODUCCION                         | 1 |
| 3. | EL MICROPROCESADOR 8086 DE INTEL     | 2 |
| 4. | LA SERIE Z8000 DE ZILOG              | 3 |
| 5. | EL MICROPROCESADOR 68000 DE MOTOROLA | 5 |
| 6. | BIBLIOGRAFIA                         | 6 |
|    | FIGURAS                              | 7 |

1. RESUMEN

Se hace un análisis de las características e innovaciones sobresalientes de los microprocesadores de 16 bits 8086, de Intel, Z8000 de Zilog y 68000 de Motorola.

2. INTRODUCCION

La primera generación de microprocesadores, con un bus de datos típicamente de 4 bits, podía integrar algunos sistemas antes diseñados con lógica discreta. Su campo de acción, sin embargo, estaba restringido a pequeños sistemas de control y calculadoras manuales.

Con la segunda generación (por ejemplo, el 8080) con buses de dirección y datos (8 bits) separados, se incrementó considerablemente la velocidad de procesamiento. Se empezaron a implementar con grandes ventajas sistemas orientados a aplicaciones que antes eran dominio absoluto de las minicomputadoras, sobre todo en el campo de las computadoras para oficina.

En 1978 se fabrica el primer producto de la llamada tercera generación de microprocesadores. El microprocesador de 16 bits aparece en el mercado con varios criterios de diseño novedosos.

Su gran capacidad de direccionamiento de memoria y su alta velocidad de procesamiento, con la incorporación de técnicas co-

mo el manejo de datos en 'Pipeline', lo colocan como un serio competidor de la minicomputadora convencional.

En seguida se analizan en sus características específicas, 3 de los microprocesadores de 16 bits más conocidos en el mercado: El 8086 de Intel, la serie Z8000 de Zilog y el 68000 de Motorola.

### 3. EL MICROPROCESADOR 8086 DE INTEL.

Entre sus características más importantes se cuentan:

- a) La unidad central de procesamiento (CPU) se divide en otras 2 unidades que operan asincrónamente:
  - Unidad de acoplamiento del bus (BIU)
  - Unidad de ejecución (EU).

La BIU manipula todas las transferencias a través del bus externo. Cada vez que la EU necesita direccionar la memoria o puertos de E/S, hace una solicitud de acceso al BUS a la BIU. Si ésta última no se encuentra ocupada responde con una señal de reconocimiento y cede el control.

Si la BIU no tiene ninguna solicitud de acceso de la EU realiza ciclos **FETCH**<sup>1</sup>, llenando parcial o totalmente

---

<sup>1</sup> Transferencia de memoria al  $\mu$ p, del código de operación de la siguiente instrucción.



una pila (STACK) de 6 bytes, con los nuevos códigos de operación (ver fig. 1).

Lo anterior permite realizar procesamiento en paralelo (PIPELINE), con lo cual se reduce grandemente el tiempo de ejecución.

- b) Su arquitectura permite diseñar desde sencillas configuraciones con un solo CPU, hasta sistemas de multiprocesamiento. Para satisfacer este último punto, el 8086 cuenta con lógica para manipular el BUS por prioridades y señales de control.
- c) En configuración de multiprocesamiento, pueden existir bloques de memoria compartidos por varios procesadores.
- d) Direccionamiento relativo de memoria en todos los casos, lo que genera programas relocizables.
- e)  $10^6$  bytes de memoria externa.

#### 4. LA SERIE Z8000 DE ZILOG.

Zilog proporciona 2 versiones del microprocesado. De 16 bits: el Z8001 y el Z8002. El primero se emplea en configuraciones sencillas, mientras que el último tiene capacidad para el manejo de memoria por segmentos relocizables en forma programada.

Esta última versión puede direccionar hasta 8 megabytes de memoria.

La serie Z8000 ofrece 16 registros de propósito general. Cualquiera puede utilizarse como acumulador o registro índice.

La unidad aritmética/lógica maneja 7 tipos diferentes de información: bits, bytes, dígitos BCD, palabras de 16 bits, palabras de 32 bits y cadenas de bytes y palabras (16 bits).

Como casi todos los procesadores de su tipo, la serie Z8000 cuenta en su arquitectura con facilidades para el diagnóstico de fallas y protección del sistema, mediante el uso de 'traps' que son interrupciones al sistema generadas externa ó internamente, por condiciones específicas; ejecución de instrucciones ilegales ó acceso a segmentos de memoria protegidos son algunos ejemplos.

Por otra parte, el empleo de 2 modos de operación (modo usuario y modo sistema), con la existencia de instrucciones 'clave' no ejecutables en modo usuario y de 2 pilas (stack) independiente para cada modo, añaden claridad y protección a la programación del procesador.

Finalmente, el uso de la unidad externa para el manejo de memoria MMU, de fácil acoplamiento a la serie Z8000 proporciona al sistema 3 funciones adicionales: (a) relocalización dinámica de segmentos de memoria, (b) protección a zonas específicas

y (c) manejo del espacio de memoria por segmentos (ver fig. 2).

#### 5. EL MICROPROCESADOR 68000 DE MOTOROLA.

El 68000 es el único de los 3 procesadores en cuestión con capacidad para direccionar, directamente y no a través de multiplexaje, 16 megabytes de memoria.

Entre sus características sobresalientes se encuentran:

- (a) sistema de E/S mapeado en memoria
- (b) operación con 5 tipos diferentes de datos
- (c) 14 tipos de direccionamiento
- (d) 17 registros de 32 bits.

El 68000 contiene lógicas para compartición de memoria con otros procesadores.

Los sistemas operativos que se diseñen para control de los periféricos, cuentan con lógica de soporte residente en el 68000 como es la detección de errores de programación a través de 'traps' bajo las siguientes condiciones:

- (a) Ejecución de instrucciones ilegales
- (b) Error en la transferencia de información de ó hacia los periféricos.
- (c) División entre cero
- (d) Sobreflujo en un registro

(e) Contenido de un registro, fuera de límites.

Adicionalmente, el procesador permite no solo correr programas paso a paso, sino la generación automática de interrupciones que permiten analizar el estado interno del 68000 después de cada paso.

El 68000, al igual que el Z8000 puede operar en los modos 'sistema' y 'usuario', reservando ciertas instrucciones 'privilegiadas' para utilizarse únicamente en el primer caso.

El diseño del 68000 permite su acoplamiento directo con todos los controladores de la línea del 6800, pudiendo integrarse de esta forma configuraciones más simples.

En la fig. 3 se muestra la distribución y función de las señales del procesador.

#### 6. BIBLIOGRAFIA

"16 bit Microprocessor Handbook" Adam Osborne, Jerry Kane.

"Z8000 Technical Manual" Advanced Micro Devices.

"MC68000 16 bit Microprocessor User's Manual" Motorola Inc.

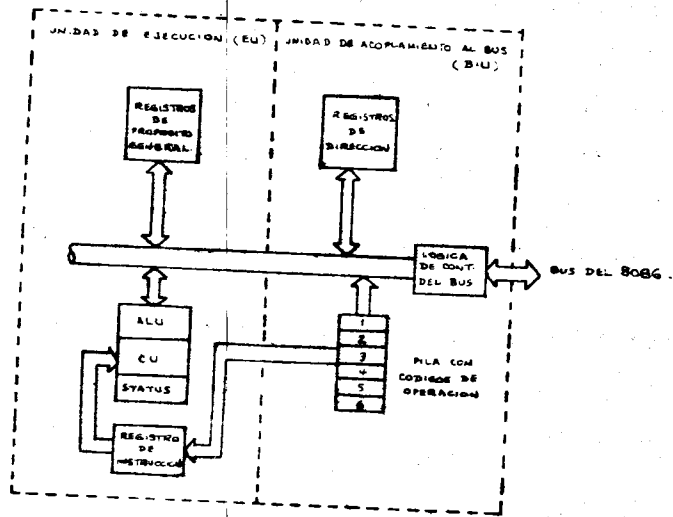


FIG 1. SISTEMA DE 'PIPELINE' EN EL BOGG .

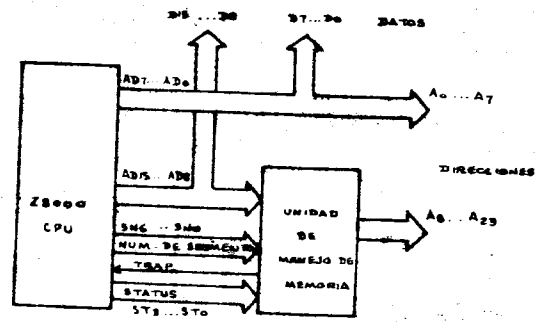


FIG 2 : ACOPLAMIENTO DEL Z8000 Y LA UNIDAD DE MANEJO DE MEMORIA.

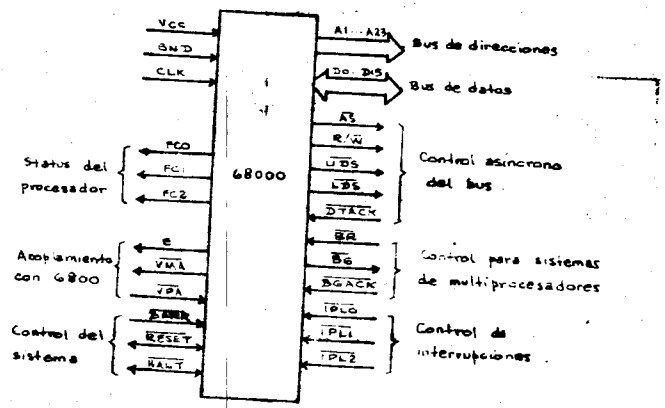


FIG 3. SEÑALES PRESENTES EN EL MICROPROCESADOR 68000

MEDICION DE TEMPERATURA  
CON MICROCOMPUTADORAS

1. Introducción
2. Termopares
3. El RTD
4. El termistor
5. Sensores de circuito integrado
6. Diseño de un sistema de adquisición de datos

JUAN B. MARTINEZ  
Septiembre, 1984



## MEDICION DE TEMPERATURA CON MICROCOMPUTADORAS

### 1. INTRODUCCION:

En medios industriales, es frecuente el uso de sensores de temperatura para efectos de supervisión y control. El empleo de estos sensores se hace en una amplia gama de procesos industriales: en plantas químicas, plantas térmicas, altos hornos, industria del plástico, tratamientos térmicos, industria del vidrio, etc.

Los sensores más usualmente empleados son de 4 tipos:

- a) Termopares
- b) Dispositivos de resistencia de platino (RTD)
- c) Termistores
- d) Sensores de circuito integrado

En la fig 1 se muestran las funciones de transferencia aproximadas de cada uno de los 4 tipos mencionados.

En seguida se listan sus características más importantes.

| Características           | Termopar     | RTD         | Termistor   | Sensor de C.I. |
|---------------------------|--------------|-------------|-------------|----------------|
| Linealidad                | media        | buna        | no lineal   | muy buena      |
| Rango de temperatura      | hasta 2000°C | hasta 600°C | hasta 150°C | hasta 100°C    |
| Costo                     | bajo         | alto        | bajo        | bajo           |
| Sensitividad              | baja         | baja        | alta        | alta           |
| Estabilidad               | media        | alta        | media       | media          |
| Requieren fuente de poder | no           | sí          | sí          | sí             |

Todos estos sensores funcionan bajo principios distintos, y sus diferentes características los hacen utilizables en las aplicaciones más diversas. El termopar es, sin duda, el dispositivo de uso más generalizado para medición de temperatura, sobre todo en medios industriales.

En seguida se analiza el funcionamiento de cada uno de los 4 tipos y su acoplamiento a instrumentos de medición y microcomputadoras. El termopar, por las razones arriba expuestas, recibirá mayor atención que el resto de los dispositivos. En el capítulo 5 se describe el diseño de un sistema de adquisición de datos para 8 canales, aplicable a los sensores aquí descritos, pero también de cualquier tipo (presión, gasto, nivel, etc.)

## 2. TERMOPARES:

Los termopares funcionan en base al llamado efecto termoeléctrico, que se produce cuando 2 conductores formados por metales distintos se conectan en ambos extremos y uno de los extremos se calienta; en la malla aparece entonces una circulación de corriente (ver fig 2).

Si se abre el circuito, el voltaje que aparece en los extremos es una función de la temperatura de la juntura caliente y la composición de ambos metales.

En algunas regiones de operación, el voltaje obtenido es aproximadamente proporcional a la temperatura, es decir:

$$V_0 = \alpha T$$

Dependiendo de los metales empleados, se tienen termopares que desarrollan diversos valores de voltaje entre sus terminales, y a diferentes temperaturas de operación.

En la siguiente tabla se da la lista de los tipos de termopares más empleados industrialmente, junto con sus principales características:

| Tipo | Metales                         | Temp. de operación (°C) | Voltaje (mV)  |
|------|---------------------------------|-------------------------|---------------|
| J    | Acero - cuproníquel             | -200° a 850°            | -7.52 a 50.05 |
| K    | Níquel-cromo<br>Níquel-aluminio | -200° a 1250°           | -5.51 a 51.05 |
| E    | Níquel-cromo<br>cuproníquel     | 0° a 1000°              | 0 a 75.12     |
| T    | Cobre/cuproníquel               | -200° a 400°            | -5.28 a 20.80 |

(juntura de ref a 0°C)

En la fig 3 se muestran las curvas de respuesta para los termopares tipos E, J y K.

En muchas aplicaciones, es suficiente tomar la aproximación con una recta sin incurrir en errores mayores a °C. En instrumental de precisión, sin embargo, se aproximan usualmente las curvas mostradas con varios segmentos lineales.

El mismo efecto termoeléctrico que permite a los termopares generan un voltaje al calentar la juntura de medición, se presenta en la denominada 'juntura fría', que es el punto de unión entre ambos extremos del termopar, y las líneas de cobre del instrumento de medición o controlador (ver fig 4). Este voltaje desarrollado en la juntura fría, aunque bastante menor que el correspondiente a la juntura de medición, da origen a errores considerables en la medición. Si la juntura fría se encuentra a temperatura ambiente, la compensación de este error se hace relativamente

te complicada, debido a las variaciones de temperatura ambiente, que generan a su vez diferentes voltajes.

La compensación del error generado en la junta fría puede hacerse de varias formas; las dos más comúnmente empleadas son:

1. Uso de un baño de hielo en la junta fría; de esta forma, se mantiene a temperatura constante (  $0^{\circ}\text{C}$  ), siendo la lectura en la junta de medición muy confiable. El método presenta problemas por el hecho de tener que conservar el hielo. Los fabricantes venden, sin embargo, cámaras de refrigeración que conservan permanentemente la temperatura en el punto de congelación. (ver fig 5).
2. Red electrónica de compensación; en este caso, se conecta un circuito-sensor a la junta fría que, para mayor precisión, puede estar en un bloque isoterma. De esta forma, conociendo la temperatura del bloque, pueden calcularse las caídas de voltaje  $V_{e1}$  y  $V_{e2}$  y compensar el error. El sensor puede ser un RTD, un termistor o un sensor de circuito integrado.

La compensación del error puede hacerse conectando un circuito puente de compensación que genere automáticamente un voltaje inverso al voltaje de error. En la fig 5, se propone un circuito cuyo sensor puede ser un termistor o un RTD. Este tipo de circuitos se emplean sobre todo si el instrumento de medición es un voltmetro de precisión o algún otro dispositivo que no disponga de posibilidades de programación.

Si, por otro lado, los datos de temperatura provenientes del termopar van a ser procesados por una microcomputadora, entonces es más sencillo hacer una compensación dentro del programa de aplicación. De esta forma, con un circuito como el de la fig 4, la microcomputadora lee, a través de un conversor A/D, un valor proporcional a la temperatura de la junta fría. Conocido este parámetro, el procesador realiza sencillas operaciones que compensan el error de lectura proveniente del termopar.

El método anterior es especialmente eficiente cuando se tienen varios termopares conectados a una microcomputadora, dado que solo se requiere de un solo sensor, mientras que la solución analógica requeriría de un circuito de compensación por cada termopar.

Para encontrar el valor de compensación (en forma aproximada) pueden seguirse los siguientes pasos:

1. Hacer la medición con la juntura caliente a una temperatura conocida, por ejemplo el punto de ebullición, y la juntura fría en un baño de hielo.
2. Repetir la operación, esta vez con la juntura fría a la temperatura ambiente, medida con algún termómetro calibrado.
3. La diferencia entre las lecturas en los puntos (1) y (2) da el valor de error. Este valor dividido entre la temperatura ambiente (en °C), resulta en el error/°C, con lo que se puede compensar el error al conocerse la temperatura de la juntura fría.

Dada la característica semilineal del termopar, el método anterior permite obtener una precisión razonable y es suficiente en muchas aplicaciones.

Además del error en que se incurre al tomar como una recta la característica del termopar, se añade a este el error de medición del voltmetro o del conversor A/D según el caso.

En la siguiente tabla se listan los termopares tipos J, K, E y T junto con su sensibilidad.

| Tipo | Sensitividad ( $\mu\text{V}/^\circ\text{C}$ ) |
|------|---|
| J    | 51  |
| K    | 40  |
| E    | 62  |
| T    | 40  |

Suponiendo un circuito similar al de la fig 4, que cuente con un conversor de 8 bits, con rango de 0-5V; se obtiene un valor de aproximadamente 20 mV por cada incremento a la lectura de los 8 bits del conversor (es decir, 1 LSB de incremento).

Supongamos que, con un termopar tipo K, deseamos hacer lecturas de 0 a 1000°C. La lectura obtenida directamente con un voltmetro a 1000°C sería aproximadamente de 40 mV. Si el amplificador de instrumentación tiene una ganancia de 125, obtendríamos un voltaje de 5V a su salida (a 1000°C).

Con las condiciones anteriores, suponiendo un conversor A/D de 8 bits con un error máximo de  $\pm 1$  LSB, se tiene un error máximo en la lectura, en °C de:  $Q = \pm (1000^\circ\text{C}/256 \text{ pasos}) = \pm 4^\circ\text{C}$ .

La precisión conseguida en este ejemplo es razonable en algunas aplicaciones, siendo enteramente impráctica en muchas otras.

Dependiendo de la precisión deseada, pueden emplearse conversores de 10 ó hasta 12 bits.

### 3. EL RTD:

Los RTDs(1), son transductores de temperatura. Operan sobre el principio del cambio de la resistencia eléctrica en los metales, en función de la temperatura.

(1) Del inglés 'Resistance temperature device'



El platino es el metal que mejor desempeño tiene para este tipo de dispositivos, dada su alta estabilidad eléctrica, su comportamiento lineal y su facilidad de calibración durante su fabricación.

Los termómetros de resistencia de platino se usan para medición de temperatura, en los rangos de  $-220^{\circ}\text{C}$  a  $+600^{\circ}\text{C}$ . El coeficiente de cambio alcanza los  $0.4 \Omega/^{\circ}\text{C}$  (el termopar tiene, en promedio,  $20 \mu\text{V}/^{\circ}\text{C}$ ).

Debido a su alto costo (alrededor de 50 U.S., precios de 1982) no han desplazado, sin embargo, a los termopares en el mercado de sensores de temperatura.

La manera más sencilla de realizar la medición es a través de un puente calibrado para salida de  $0 \text{ mV}$  a  $0^{\circ}\text{C}$  (ver fig 7.a).

El valor más común de los RTD, a  $0^{\circ}\text{C}$ , es de  $100 \Omega$ . Si el sensor se conecta a varios metros de distancia del instrumento de medición, la resistencia de los cables de interconexión puede introducir errores en la medición, como se muestra en la fig 7.b.

Con el objeto de que los cables de interconexión no afecten la calibración, se prefiere usualmente la conexión de la fig 7.c, en la que se cancela los efectos de la resistencia del cable, al encontrarse en ramas opuestas del puente.

Por otro lado, el cable de interconexión también experimenta cambios de resistencia con la temperatura, que se cancelan igualmente con la configuración anterior.

La corriente que circula por el RTD conectado como rama de un puente genera un calentamiento del dispositivo, que a su vez dará origen a un error en la medición. Para minimizar este error, debe hallarse un compromiso entre la corriente que circula por el RTD (la mínima posible) y el voltaje de salida del puente (que debe ser el máximo posible). Lo anterior se decide sobre todo en base a la sensibilidad del instrumento de medición disponible.

#### 4. EL TERMISTOR:

Los termistores se fabrican a partir de óxidos de níquel, magnesio, cobre, magnesio y otros metales.

Al igual que el RTD, el termistor es un sensor que varía su resistencia en función de la temperatura.

De todos los tipos de sensores, el termistor es el que exhibe el mayor coeficiente de cambio de resistencia con respecto a la temperatura (un promedio de  $-500 \Omega/^\circ\text{C}$  en la región alrededor de  $0^\circ\text{C}$ ).

Por otro lado, la respuesta del termistor no es lineal, como se muestra en la fig 1.

El termistor tiene un coeficiente de cambio negativo, con variaciones muy rápidas (del orden de  $-100\text{KR}/^\circ\text{C}$ ) en la región de bajas temperaturas ( $-50^\circ\text{C}$ ). En la fig 8, se muestra la respuesta de un termistor típico en la zona de  $0^\circ$  a  $50^\circ\text{C}$ .

La curva del termistor se puede aproximar con bastante precisión a través de la siguiente ecuación:

$$\frac{1}{T} = A + B(\text{Ln } R) + C (\text{Ln } R)^3$$

T = grados Kelvin

R = resistencia del termistor

A, B, C = constantes

A, B y C se encuentran experimentalmente escribiendo 3 ecuaciones a partir de 3 puntos de operación conocidos y resolviendo para A, B y C. El error producido por esta ecuación es menor a  $\pm 0.02^\circ\text{C}$ , comparándola con la curva real.

Los termistores se conectan generalmente en una de las ramas de un circuito puente. La compensación de la respuesta no lineal puede hacerse a través de programa, si se cuenta con una computadora, ya sea resolviendo la ecuación arriba propuesta, o a través de una tabla conteniendo información solo de algunos puntos de operación, y tomando secciones lineales entre ellos.

Existen también en el mercado los llamados termistores lineales, formados en realidad por un circuito con 2 termistores, calibrados de tal forma que uno compensa la respuesta no lineal del otro. En la fig 8 se muestra el circuito electrónico típico de estos dispositivos.

#### 5. SENSORES DE TEMPERATURA EN CIRCUITO INTEGRADO:

A pesar de su reducida gama de temperatura de operación ( $-10^{\circ}\text{C}$  a  $100^{\circ}\text{C}$ ), los sensores de temperatura en circuito integrado (por ejemplo el LM335 de National) constituyen una buena opción dado su bajo costo y su alta linealidad.

El LM335 es un diodo zener, de voltaje de ruptura variable con la temperatura; con una corriente de 1 mA, su coeficiente de variación es de  $10\text{ mV}/^{\circ}\text{C}$ .

En la fig 9 se muestra un circuito que genera una salida de 0v a  $0^{\circ}\text{C}$ , con una variación de  $+10\text{ mV}/^{\circ}\text{C}$ . El circuito LM 329C es una referencia a 6.9v.

Este tipo de sensores encuentran su aplicación, ya sea en controles de temperatura o dentro de los circuitos de compensación de la juntura fría en los termopares.

## 6. DISEÑO DE UN SISTEMA DE ADQUISICIÓN DE DATOS:

En la fig 6 se muestra el diagrama electrónico de un sistema de adquisición de datos típico, aplicable a termopares o algún otro sensor con salida de voltaje de nivel bajo o medio. Los sensores con niveles bajos ( $V_o \leq 2 \text{ mV}$ ) se hacen pasar por una etapa de preamplificación. Dado los bajos niveles manejados a la entrada de esta etapa, debe utilizarse un amplificador de instrumentación de alta precisión (por ejemplo LF0038) que exhiba alta ganancia, muy bajo voltaje de offset y alto rechazo a las señales de modo común. En el caso del LF0038 se tiene:

Ganancia: 100 a 2,000  
 CMRR: 94 db mín  
 $Z_{in}$ : 5 M $\Omega$   
 $V_{OOS}^{(1)}$ : 10 mV

El multiplexor analógico selecciona el sensor que se desea medir; un canal se puede reservar para la medición de la temperatura ambiente, con el objeto de efectuar -por programa- la compensación correspondiente al error generado en la juntura fría, para el caso del termopar. La selección del canal se hace a través de 3 bits de un puerto de salida, conectados también a las líneas de dirección de una memoria RAM, que a su vez selecciona, a través de sus 2 bits de salida, la ganancia del amplificador de instrumentación programable (por ejemplo LF0084). De esta forma pueden manejarse sensores con una amplia gama de niveles de voltaje.

(1) Voltaje offset de salida, con los entradas conectadas a tierra.

A la salida del multiplexor analógico se conecta el amplificador de instrumentación que es, como ya se dijo, de ganancia programable. Sus parámetros son:

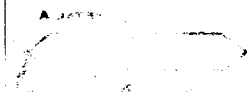
Ganancia: 10, 20, 50 ó 100

CMRR: 80 db mín.

Zin:  $10^{11} \Omega$

$V_{00S}^{(1)}$ : 10 mV

Finalmente, la señal generada por el amplificador de instrumentación se hace pasar por un convertor A/D de 10 ó 12 bits, dependiendo de la precisión deseada. Las subsecuentes correcciones de medición (compensación en la juntura fría ó voltaje de offset) pueden hacerse por programa.



(1) Voltaje offset de salida, con las entradas a tierra.

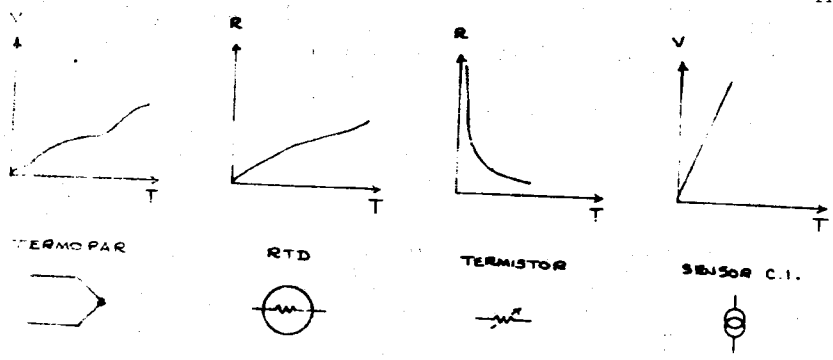


FIG. 1 : CURVAS DE TRANSFERENCIA PARA  
SENSORES DE TEMPERATURA.

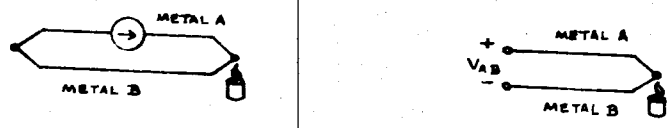


FIG 2 : PRINCIPIO DE FUNCIONAMIENTO DEL  
TERMOPAR (EL EFECTO SEEBECK)

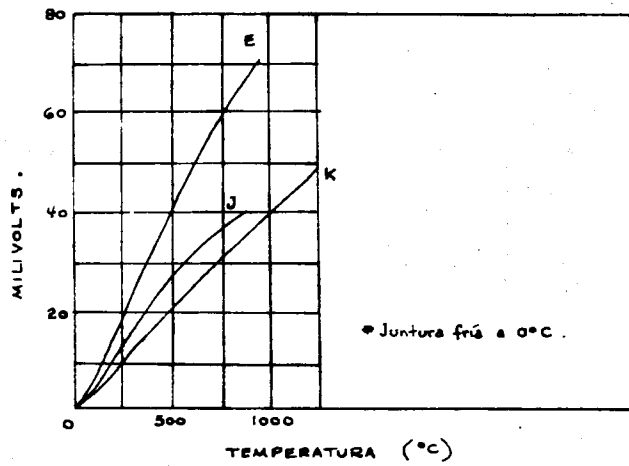


FIG 3. CURVAS DE TRANSFERENCIA DE TERMOPARES TIPOS E, J, K.

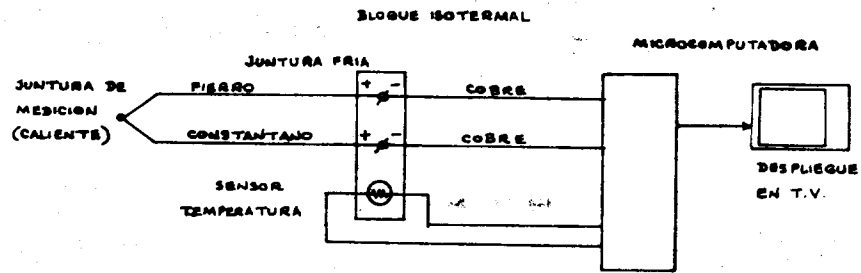


FIG 4. MEDICION DE TEMPERATURA CON COMPENSACION DEL VOLTAGE DE ERROR EN LA JUNTA FRIA.



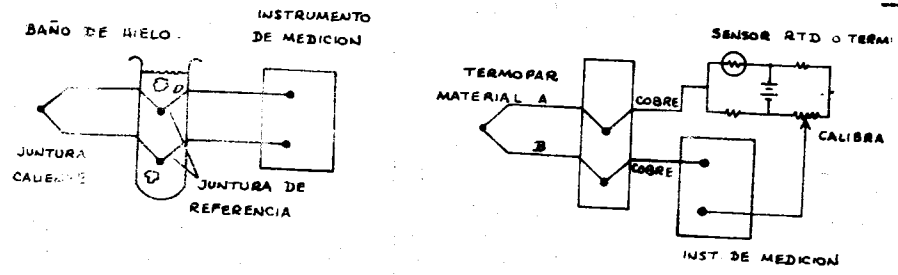


FIG 5. METODOS DE COMPENSACION DEL ERROR GENERADO EN LA JUNTURA FRIA .

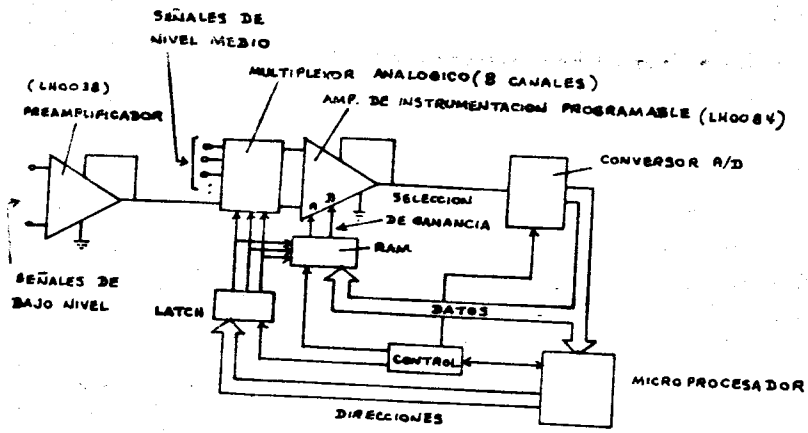


FIG 6. SISTEMA DE ADQUISICION DE DATOS

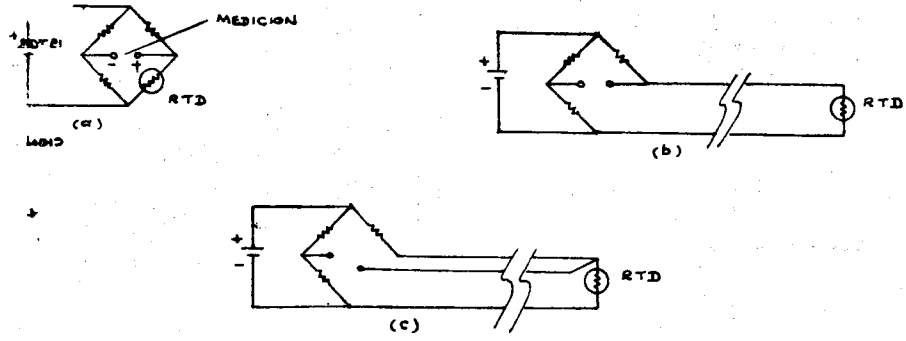


FIG 7. CONEXIONES DE SENSORES RTD

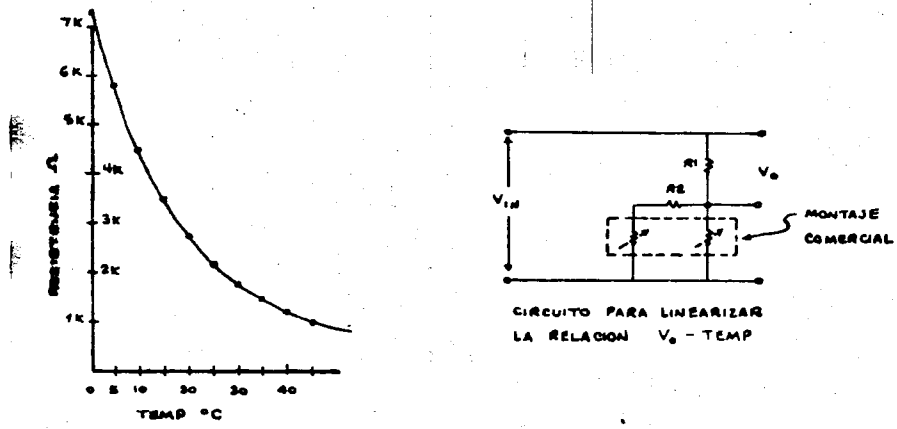


FIG 8. RESPUESTA TIPICA DEL TERMISTOR Y CIRCUITO PARA SALIDA LINEAL.

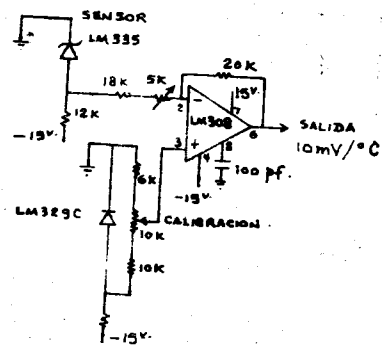
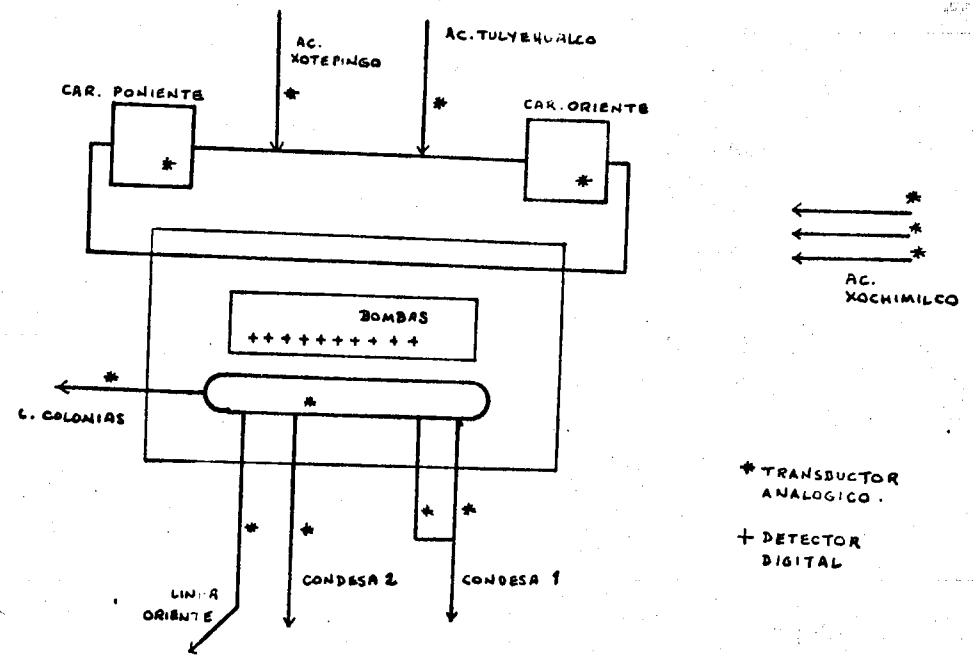
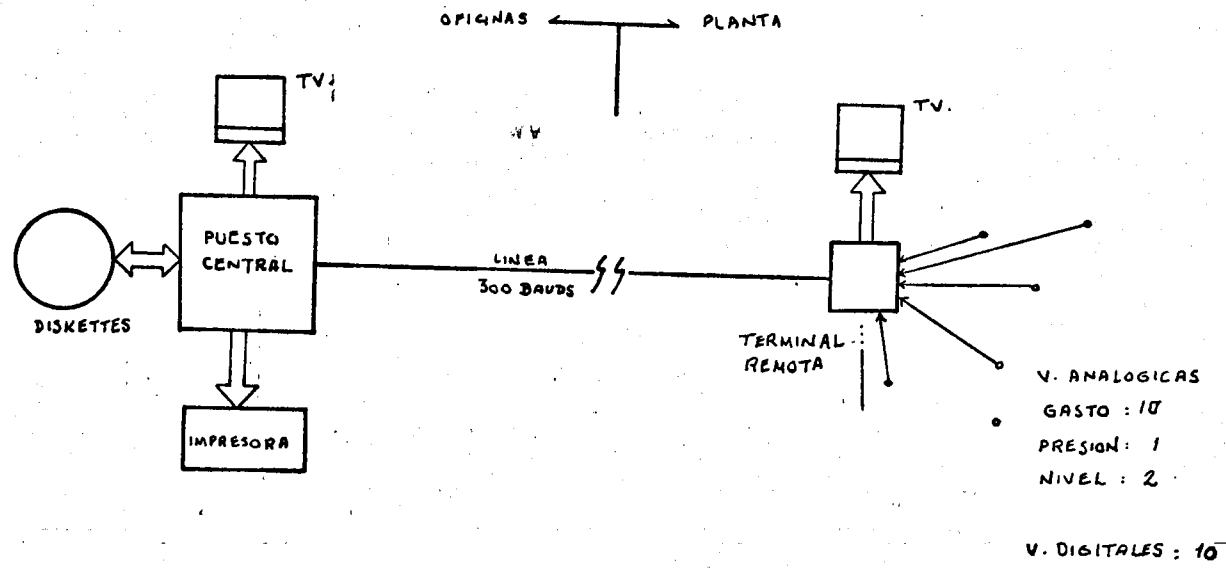


FIG. 9. CIRCUITO PARA SENSOR DE TEMPERATURA, CON SALIDA REFERIDA A TIERRA

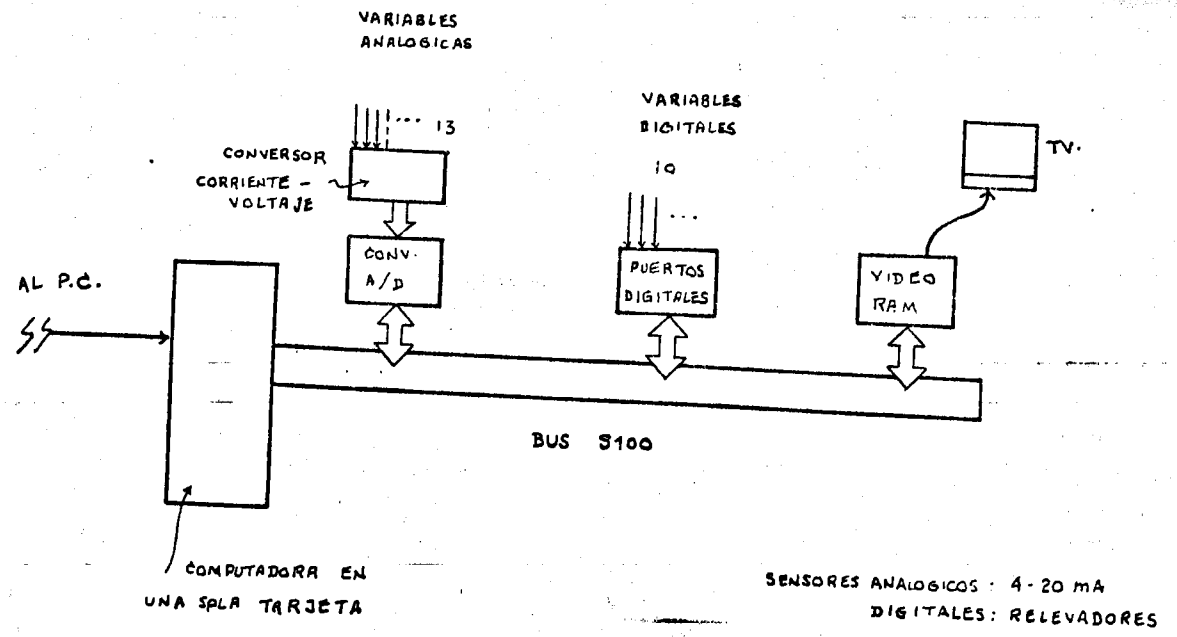
**SISTEMA DIGITAL DE INFORMACION PARA LA PLANTA  
DE BOMBEO DE XOTEPINGO**



1 ● PLANTA DE BOMBEO DE XOTEPINGO



2 ● CONFIGURACION GENERAL -



3 • DIAGRAMA TERMINAL REMOTA

32 x 16

|         |                                   |         |             |
|---------|-----------------------------------|---------|-------------|
| ENT :   | AC . X O C H I M I L C O          | 760     | L T S / S   |
|         | AC . T U L Y E H U A L C O        | 1 , 850 | "           |
|         | A U X . X O T E P I N G O         | 1 , 000 | "           |
|         | T O T A L                         | 3 , 610 | "           |
| S A L : | C O N D E S A U N O               | 540     | "           |
|         | C O N D E S A D O S               | 710     | "           |
|         | L I N E A O R I E N T E           | 1 , 100 | "           |
|         | L I N E A C O L O N I A           | 920     | "           |
|         | T O T A L                         | 3 , 290 |             |
|         | N I V E L C A R . O R I E N T E   | 320     | C M S       |
|         | N I V E L C A R . P O N I E N T E | 660     | C M S       |
|         | P R E S I O N A L A D E S C .     | 11 . 6  | K G / C M 2 |
|         | B O M B A S E N G P 1 2           | 4 5 6   | 9           |

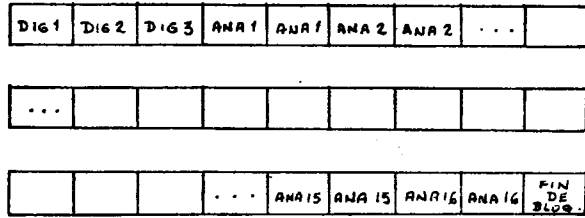
4 ● FORMATO DE DESPLIEGUE



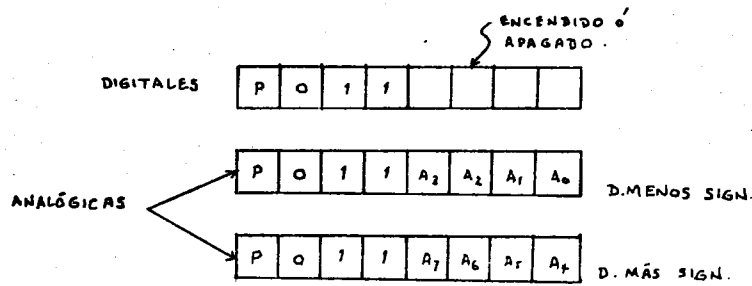


TRASMISION : • FULL DUPLEX  
 • 300 BAUDS  
 • ASINCRONA

BLOQUE DE DATOS :



TOTAL 36 BYTES



6 • FORMATO DE TRASMISIÓN

# RED TELEMETRICA DIGITAL

INSTITUTO DE INGENIERIA  
PATROCINIO : D.G.C.O.H. D.D.F.  
1983

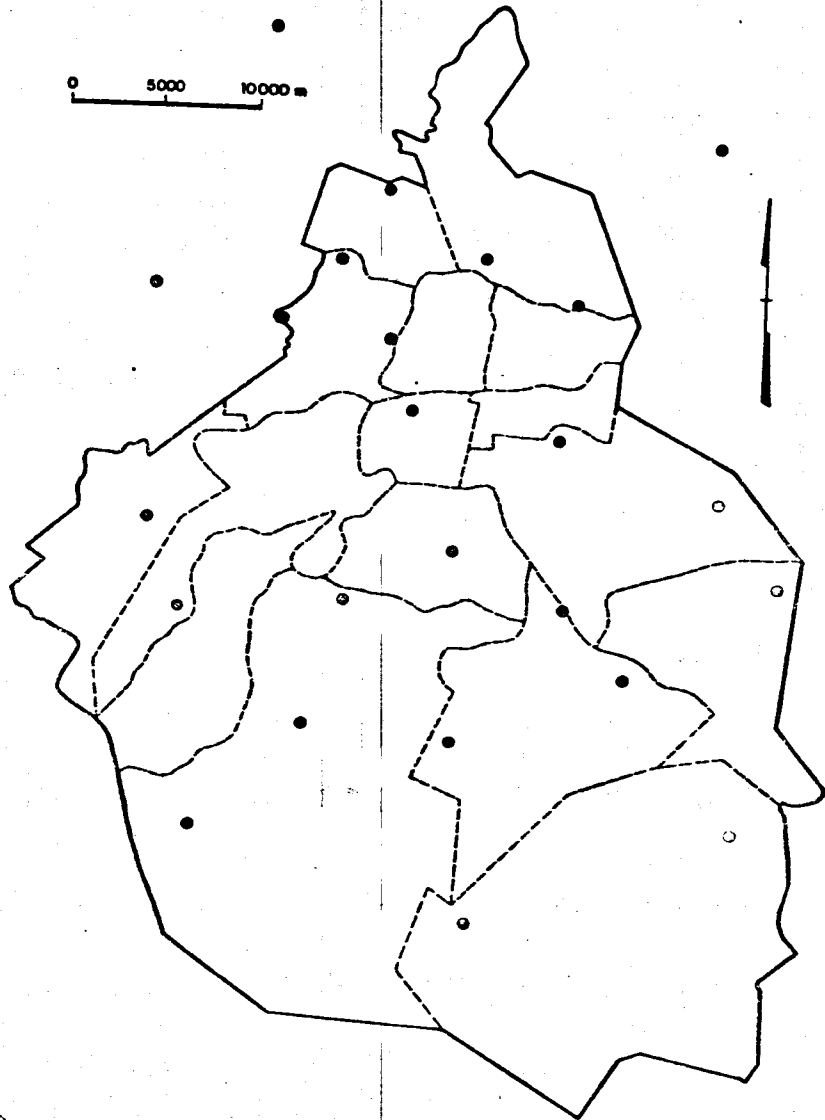
REV. 3 AM

INSTITUTO DE INGENIERIA

**OBJETIVOS :**

- INSTALACION DE UNA RED PARA MEDICION DE PRECIPITACION PLUVIAL EN 64 DIFERENTES PUNTOS EN EL AREA DEL D.F. Y ALGUNAS ZONAS ALEDAÑAS .

- TODA LA INFORMACION DEBE CONCENTRARSE AUTOMATICAMENTE EN UN PUESTO CENTRAL DE REGISTRO .



1

Localización de pluviómetros

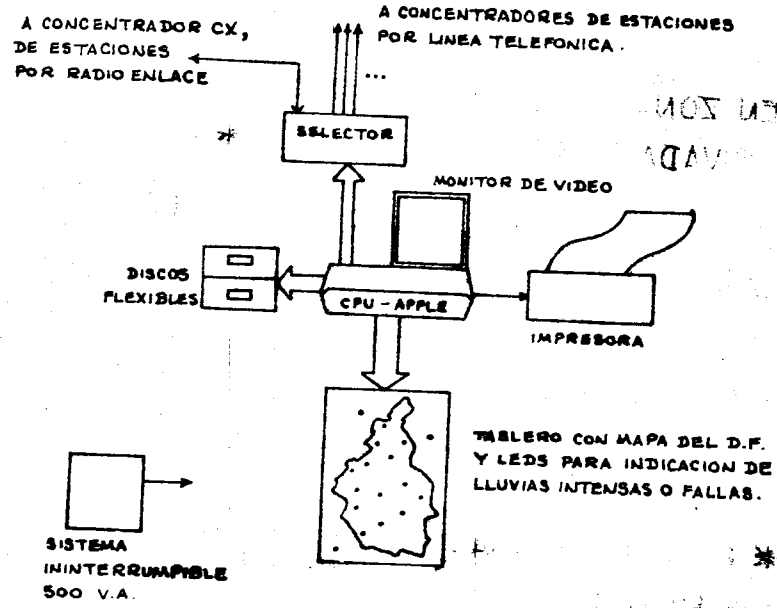
CANAL PARA LA TELECOMUNICACION :

- EN ZONAS URBANAS, LINEA TELEFONICA PRIVADA (APROX : 40 PUNTOS) \*
- EN ZONAS SEMIURBANAS, RADIOENLACE (APROX : 24 PUNTOS) \*\*

\* RED CONCLUIDA .

\*\* RED ACTUALMENTE EN DESARROLLO.

- REPORTE INSTANTANEO DE LLUVIAS
- REPORTE DE FALLAS EN LOS ENLACES
- REPORTES IMPRESOS
- FORMACION DE ARCHIVOS EN DISCOS FLEXIBLES



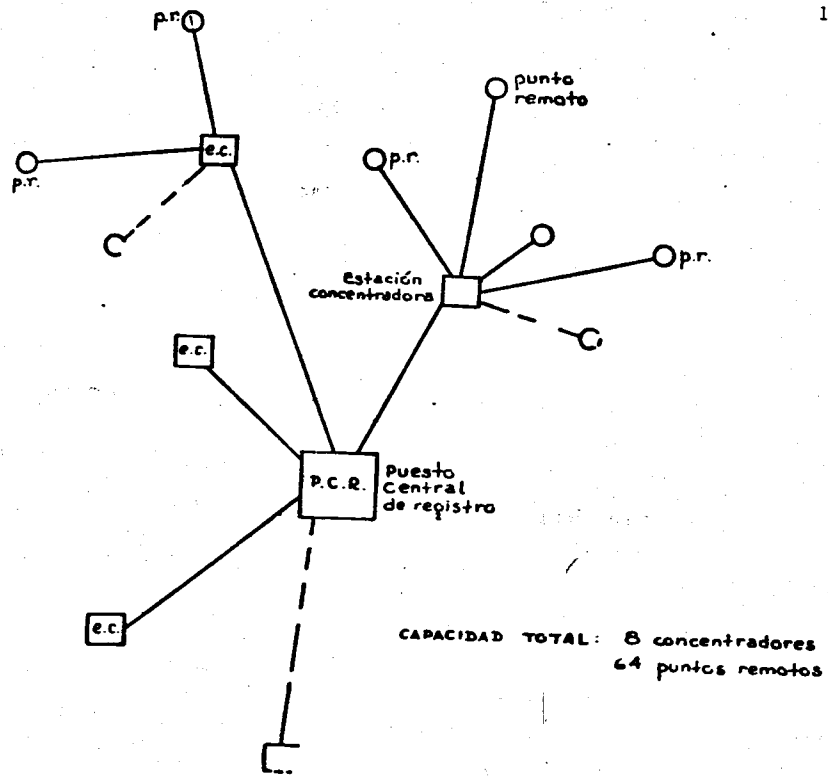


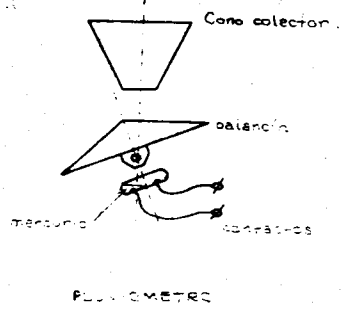
FIG 1 CONFIGURACION GENERAL DE LA RED. (VIA TELEFONICA)

2



Ventajas de la configuración propuesta:

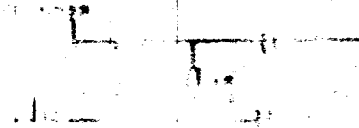
- se libera al P.C.R. de tiempo de procesamiento
- se reduce considerablemente la longitud total de las líneas de transmisión.
- facilidad en la renta de líneas privadas (a un punto se conectan un máx. de 8 líneas)

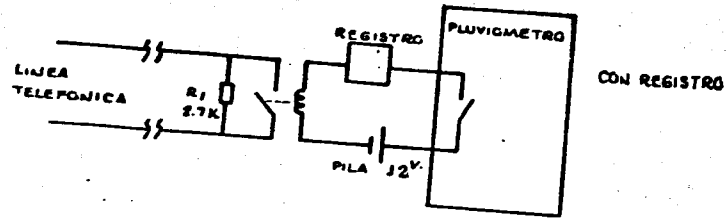
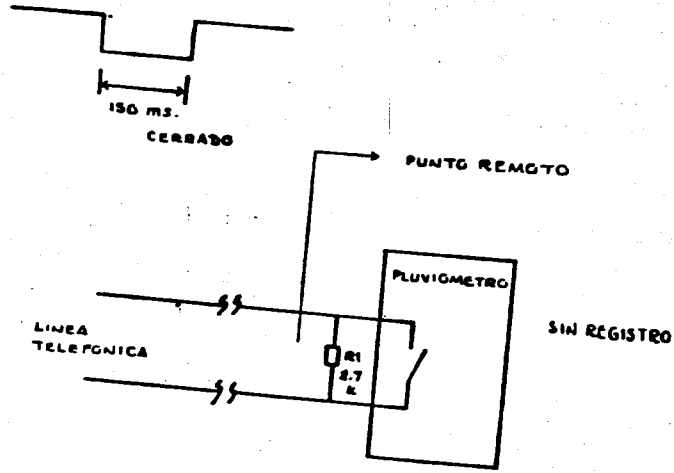


1 pulg = 0.254 mm de lluvia

los puntos remotos son pasivos

- no requieren mantenimiento eléctrico
- el funcionamiento es independiente de fallas en el suministro de energía.





5

FIG 2: CONEXION DE LGS PLUVIOMETROS

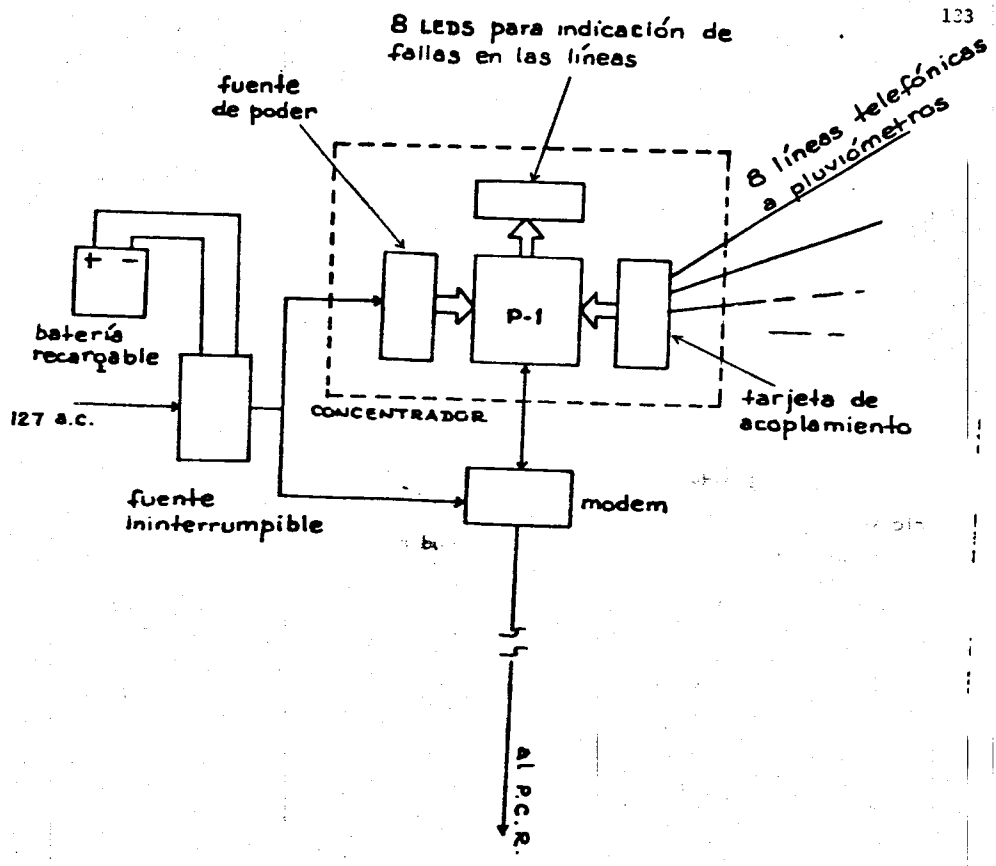
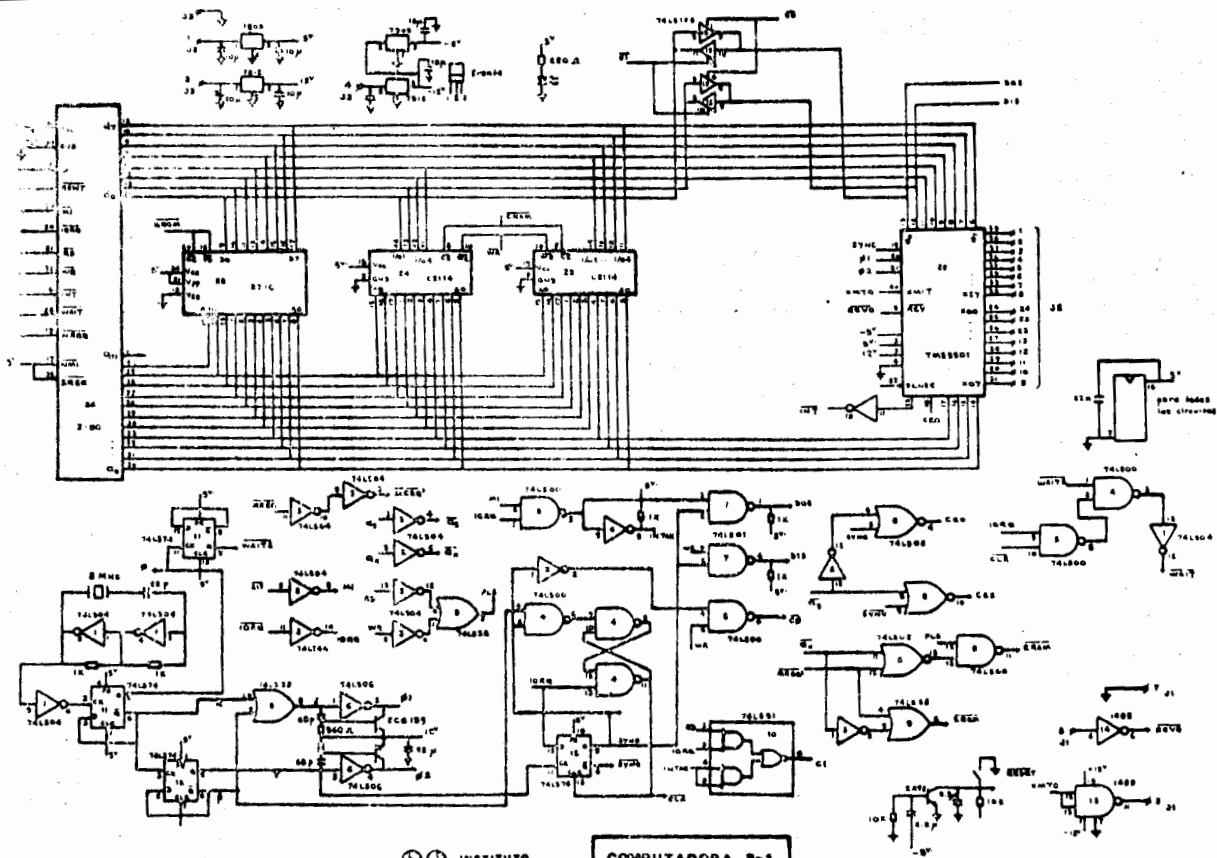


Diagrama de bloques de una estación concentradora.

6

Computadora en una tarjeta P-1

|   |     |   |
|---|-----|---|
|   | Z80 | 4 MHz   |
|   | 1K  | RAM   |
|   | 2K  | ROM   |
| controlador de<br>funciones múltiples<br>TMS 5501 | 5   | temporizadores  |
|   | 1   | puerto entrada  |
|   | 1   | puerto salida   |
|   | 1   | UART (baud rate programable)<br>control de interrupciones |



8

INSTITUTO DE INGENIERIA

COMPUTADORA P-1  
 diagrama electrónico  
 autor: j. martínez  
 feb. 1961

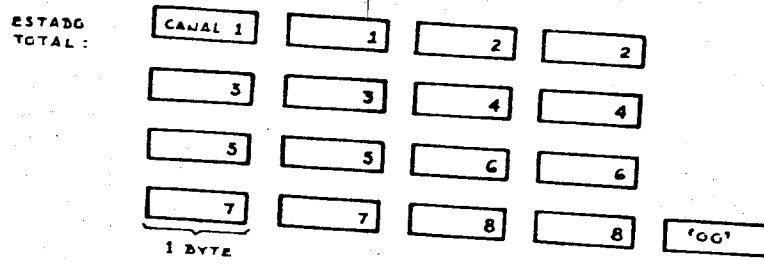
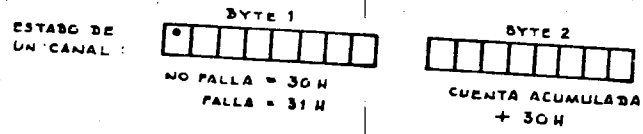
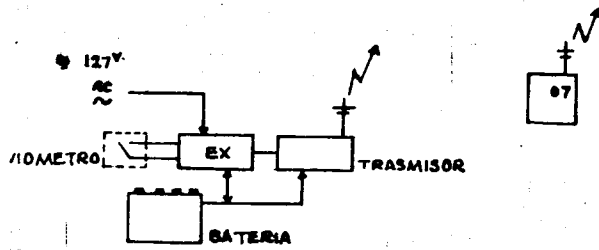
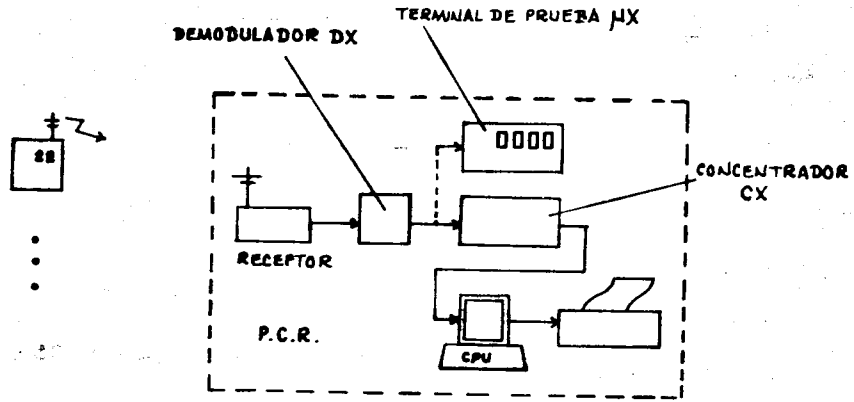
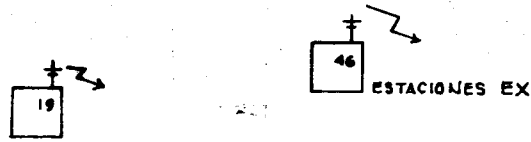
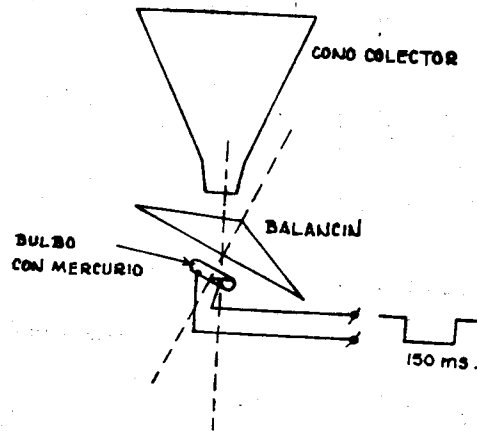


FIG 11 : ESTADO DE LA ESTACION CONCENTRADORA.



CONFIGURACION GENERAL  
(RED DE RADIOENLACE)



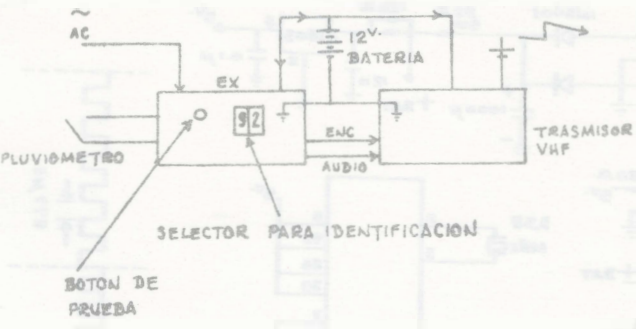


- CADA PULSO, EQUIVALENTE A 0.254 mm. DE LLUVIA.

- LA INFORMACION ES, DESDE SU ORIGEN, DIGITAL.

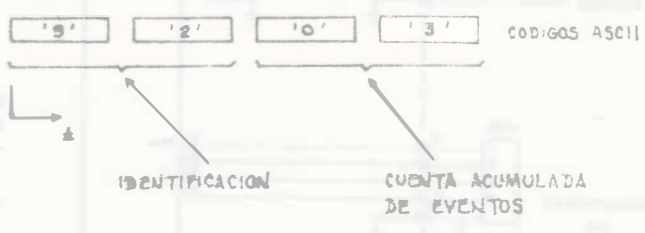
SMAN

# PLUVIOMETRO



➔ CADA VEZ QUE EN EL PLUVIOMETRO SE CIERRAN LOS CONTACTOS :

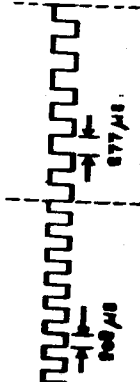
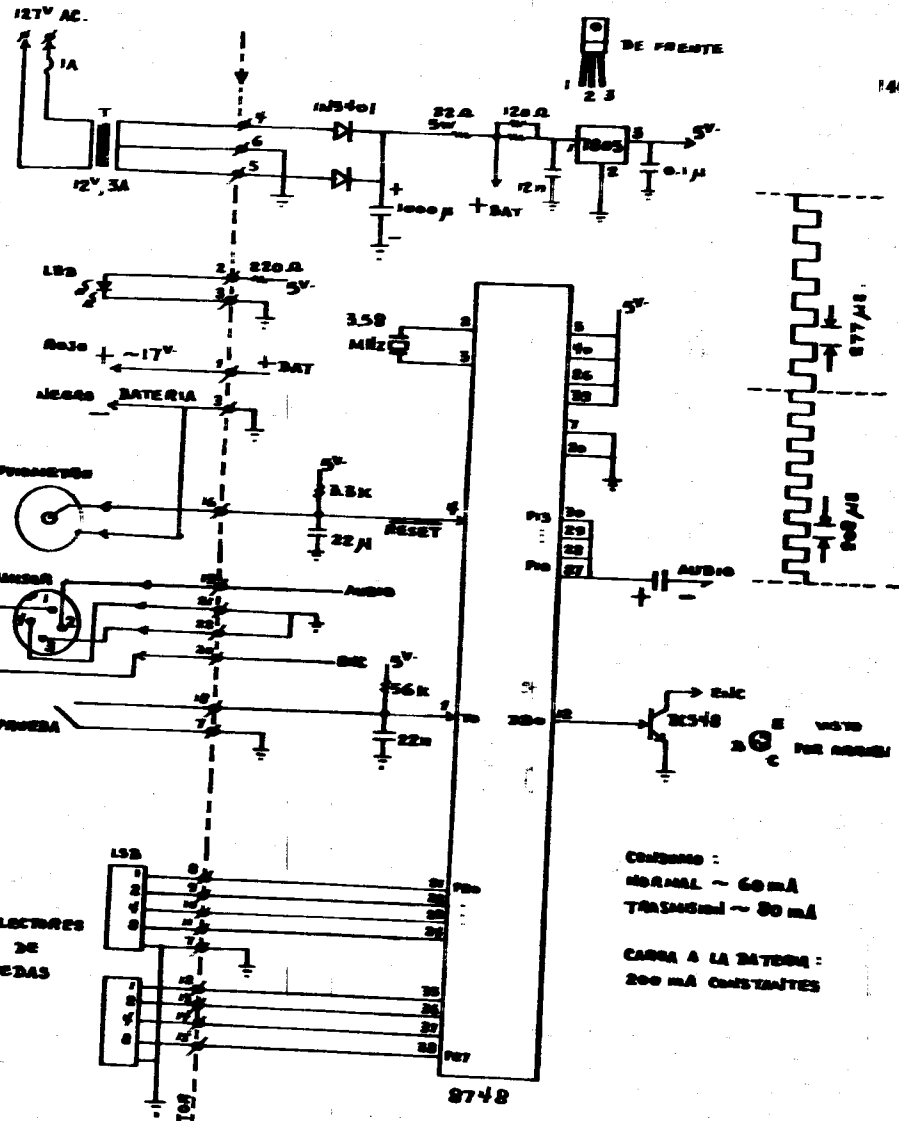
1. SE INCREMENTA UN ACUMULADOR
2. SE TRASMITE UN MENSAJE CODIFICADO



➔ CADA 30 MINUTOS, LA ESTACION SE REPORTA TRANSMITIENDO EL MENSAJE ESPECIFICADO.

ESTACION EX

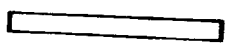
X3 UNIMET

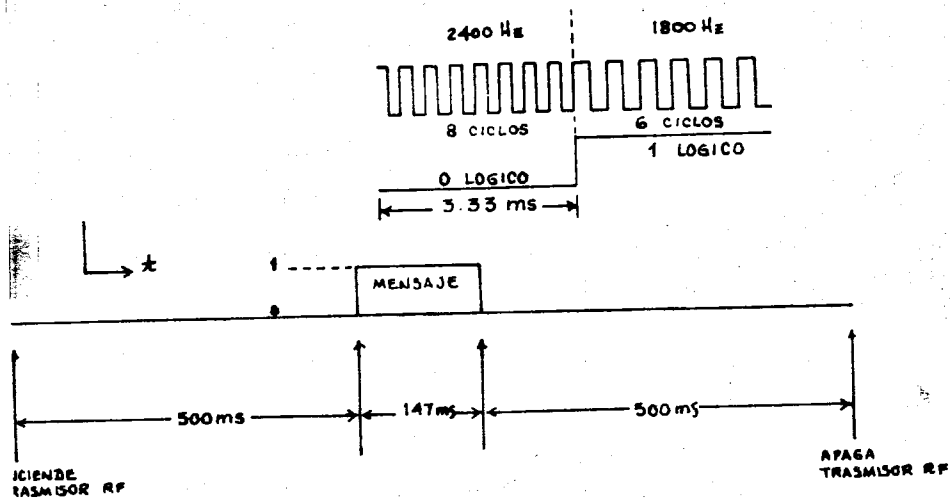


CONDICIONES:  
 NORMAL ~ 60 mA  
 TRANSICION ~ 80 mA  
 CARGA A LA ENTRADA:  
 200 mA CONSTANTES

ESTACION EX

COMPONENTES EXTERNAS ↔ TARJETA CI.

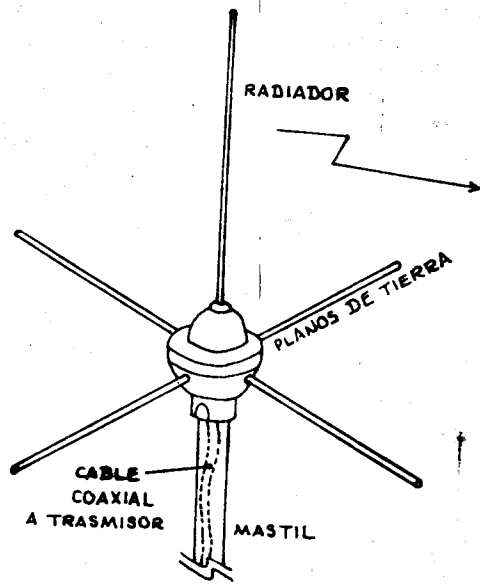




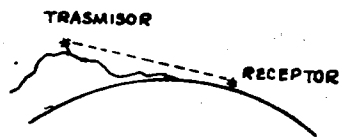
- MENSAJE :**
- MODULACION FSK [1800 - 2400 Hz]
  - 300 BITS POR SEGUNDO
  - FORMATO ASINCRONO
  - CODIGOS ASCII.

- TRASMISOR VHF :**
- POTENCIA DE SALIDA : 6 WATTS
  - MODULACION FM [155.950 MHz]

### ESPECIFICACION DEL MENSAJE



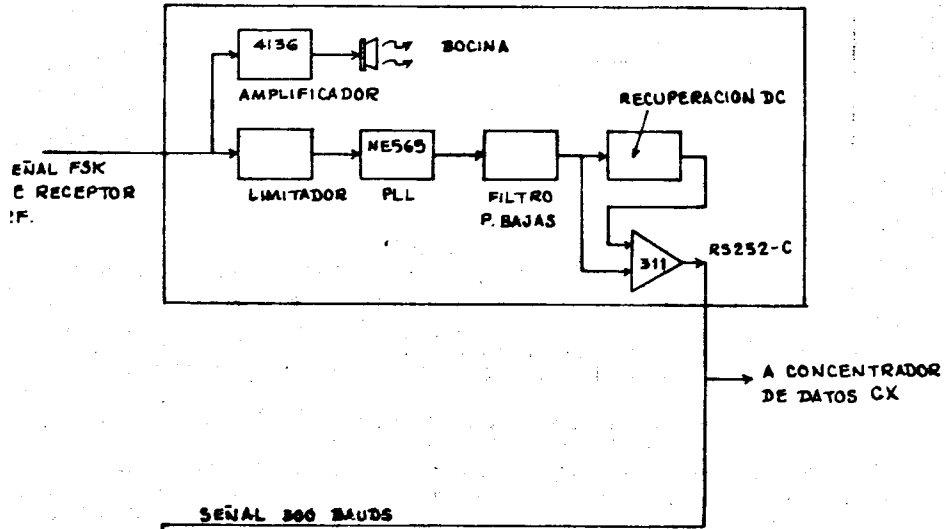
- OMNIDIRECCIONAL
- IMPEDANCIA  $50 \Omega$
- GANANCIA UNITARIA
- PATRON DE RAD.:  $78^\circ$



- LINEA DE VISTA  
INDISPENSABLE

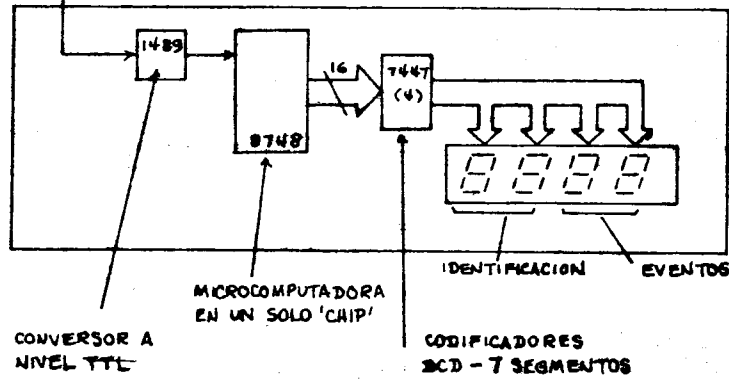
ENLACE RF

### DEMODULADOR DX



SEÑAL 300 BAUDS

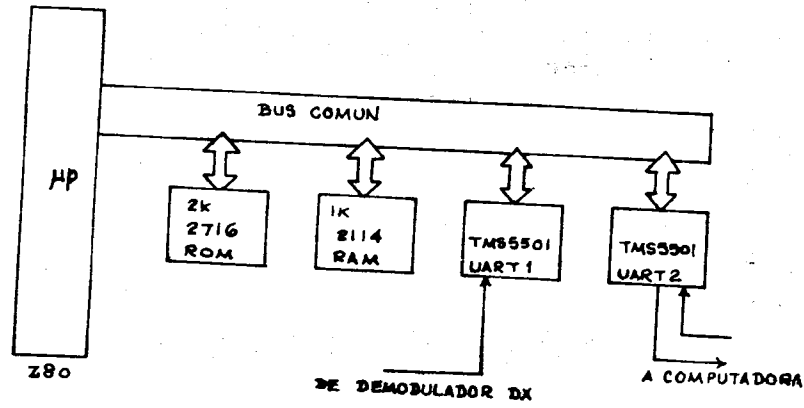
### TERMINAL DE PRUEBA JIX



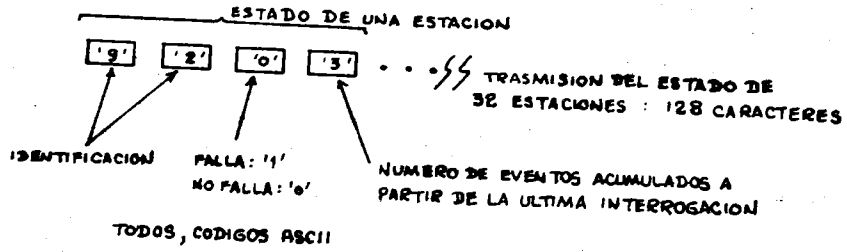
CONVERSION A NIVEL TTL

MICROCOMPUTADORA EN UN SOLO 'CHIP'

CODIFICADORES BCD - 7 SEGMENTOS



- FUNCIONES**
- RECEPCION DE MENSAJES DE UN MAX. DE 32 ESTACIONES.
  - CALCULO DE INCREMENTOS EN EL N° DE EVENTOS, POR ESTACION.
  - ASIGNACION DE FALLA A ESTACIONES QUE NO SE HAYAN REPORTADO EN LOS ULTIMOS 30 MINUTOS.
  - TRASMISION, SOBRE INTERROGACION, DEL ESTADO DE LA RED:



CONCENTRADOR CX

" SISTEMA AUTOMÁTICO DE  
SUPERVISIÓN Y CONTROL  
DE POZOS (SASCP) "

COORDINACIÓN : AUTOMATIZACIÓN

PATROCINIO : D.G.C.O.H.D.D.F.

AVANCE : 95 %



ANTECEDENTES :

146

- \* EL 90% DEL AGUA POTABLE CONSUMIDA EN LA CD. DE MÉXICO PROVIENE DE MANTOS SUBTERRÁNEOS .
  
- \* EN EL D.F. EXISTEN APROXIMADAMENTE 600 POZOS MUNICIPALES DE DONDE SE EXTRAE AGUA CONTINUAMENTE .
  
- \* EN LOS POZOS NO SE CUENTA CON REGISTROS DE LA PRESIÓN EN LA RED Y FLUJO EN LA SALIDA DE LAS BOMBAS .
  
- \* EL CONTROL DE LAS BOMBAS LO REALIZA UN OPERADOR EN FORMA MANUAL .

**OBJETIVOS :**

\* INSTALACIÓN , EN UN POZO - PILOTO (ALAMEDA TACUBAYA), DE UN SISTEMA ELECTRÓNICO CON CAPACIDAD PARA :

- 1 - MEDICIÓN DE PRESIÓN Y FLUJO EN LA RED
- 2 - CONTROL AUTOMÁTICO DEL ENCENDIDO Y APAGADO DE LA BOMBA , DE ACUERDO A LA PRESIÓN DE LA RED .
- 3 - SUPERVISIÓN DE FALLAS E INTERRUPCIONES EN LA INSTALACION .
- 4 - SUPERVISIÓN Y CONTROL REMOTOS DESDE UN PUESTO CENTRAL , A TRAVÉS DE LÍNEA TELEFONICA PRIVADA .

A ESTE SISTEMA SE LE DENOMINÓ SASCP

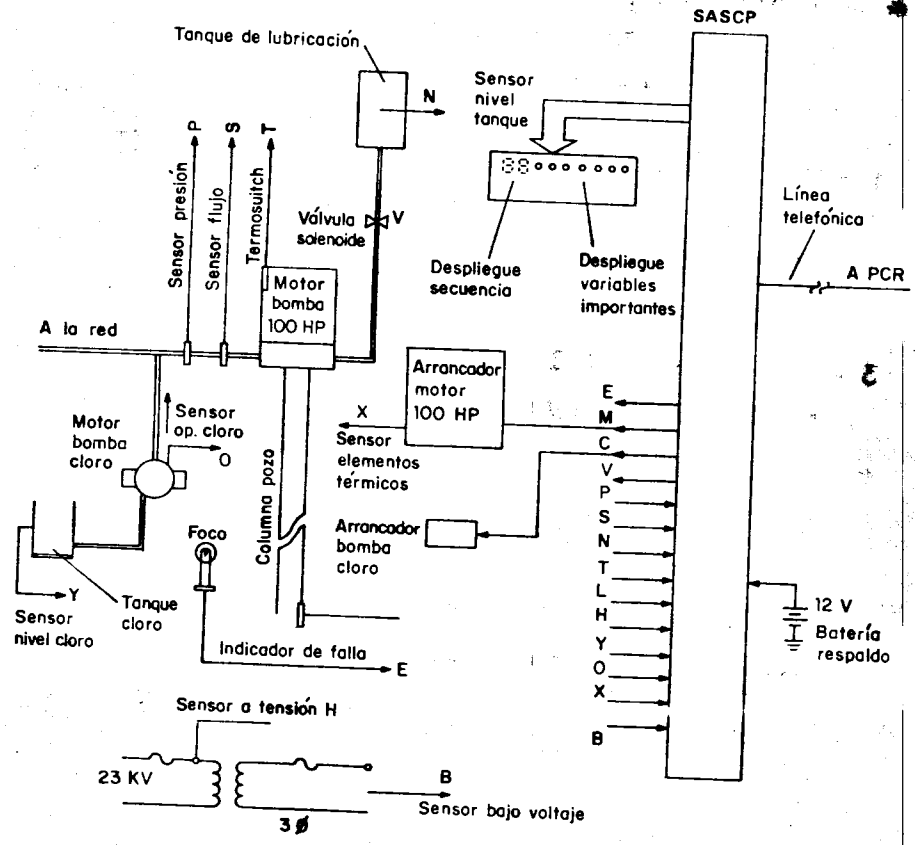


Fig 1 Diagrama instalación pozo municipal

| VARIABLE:       | DISPOSITIVOS A OPERAR:                              | ACTUADOR: <sup>149</sup>         |
|-----------------|---|----------------------------------|
| M               | Motor bomba de agua (100 HP)                        | Arrancador 100 HP.               |
| C               | Motor bomba cloro (3/4 HP)                          | Arrancador 3/4 H.P.              |
| V               | Válvula lubricación de la bomba de agua             | Válvula Solenoide                |
| E               | Foco rojo externo para indicación de falla.         | Relevador 200 W.                 |
| INTERRUPCIONES: |   | SENSOR:                          |
| X               | Sobrecorriente en los embobinados del motor 100 HP. | Elementos Térmicos               |
| B               | Bajo voltaje en la línea 127 VAC                    | Sensor electrónico de comparador |
| P               | Presión alta en la red                              | S. de presión 4-20 mA            |
| FALLAS:         |   | SENSOR:                          |
| T               | Sobrecalentamiento del motor 100 HP.                | Termoswitch                      |
| N               | Nivel bajo en el tanque de lubricación              | Flotador                         |
| Y               | Nivel bajo en el tanque de cloro                    | Flotador                         |
| O               | La bomba del cloro no opera                         | Transformador de corriente.      |

La aparición de alguna interrupción o falla deriva en la suspensión de la acción de control del sistema sobre el pozo.

Si la causa de la interrupción desaparece, el sistema se autorrestablece

La falla debe ser corregida por personal que a su vez restablece el sistema.

EL SISTEMA SASCP OPERA EN 2 MODOS :

- 1. MODO LOCAL (CONTROL Y SUPERVISIÓN LOCALES)
- 2. MODO REMOTO (CONTROL Y SUPERVISIÓN DESDE EL PUESTO CENTRAL, POR LÍNEA TELEFÓNICA).

DEFINICIONES :

- si  $F = 0$  , no existe falla .
- si  $F = 1$  , existe alguna falla .
- si  $I = 0$  , no existe interrupción .
- si  $I = 1$  , existe alguna interrupción .

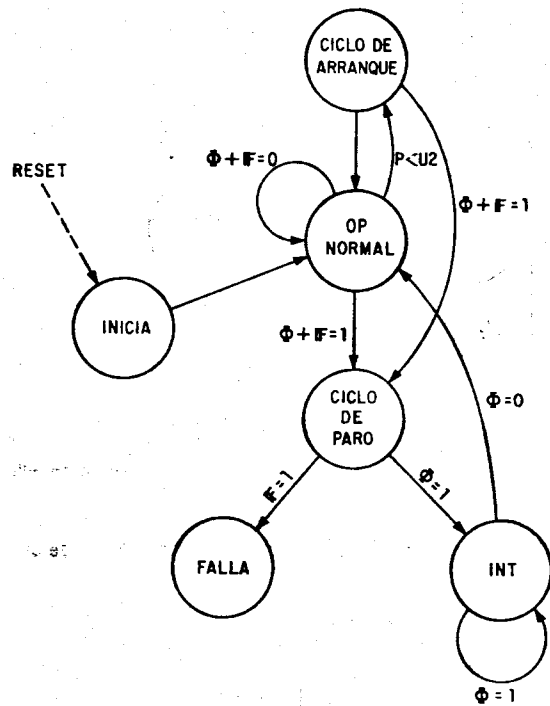


Fig 3 Diagrama de estados del controlador SASCP (modo local)

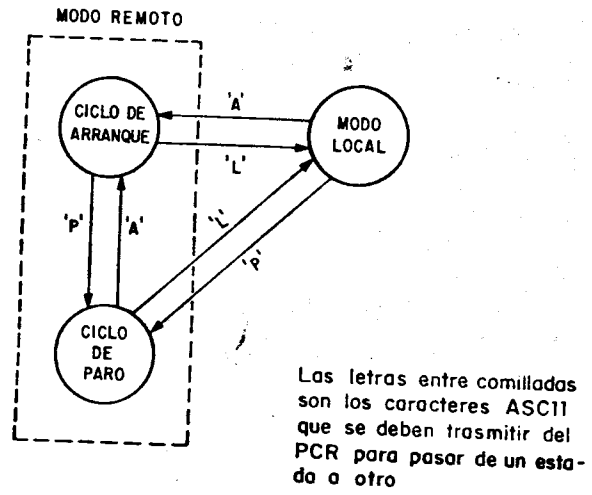


Fig 4 Diagrama de estados para el modo remoto de SASCP

COMUNICACIONES PCR - SASCP

- L.T. PRIVADA
- MÓDEM ASÍNCRONO, 300 BAUDS.

El PCR Transmite  
(códigos ASCII):

'CNTR G'

'A'

'P'

'L'

El SASCP responde:

Transmite todas las variables  
al PCR. (ver ejemplo).

Efectúa el ciclo de arranque  
de la bomba de agua. (Modo remoto).

Efectúa el ciclo de paro de  
la bomba de agua. (Modo remoto).

Regresa al control local.  
(Modo local).

Ejemplo de la transmisión del estado del pozo al PCR.  
(Todos los caracteres en código ASCII).

FALLA 5

|           |          |    |
|-----------|----------|----|
| MOTOR 0   | PRESION  | 19 |
| CLORO 0   | FLUJO    | 30 |
| VALVULA 0 | U. FLUJO | 50 |

MODER

90242 1er. Ejemplo de transmisión del SASCP



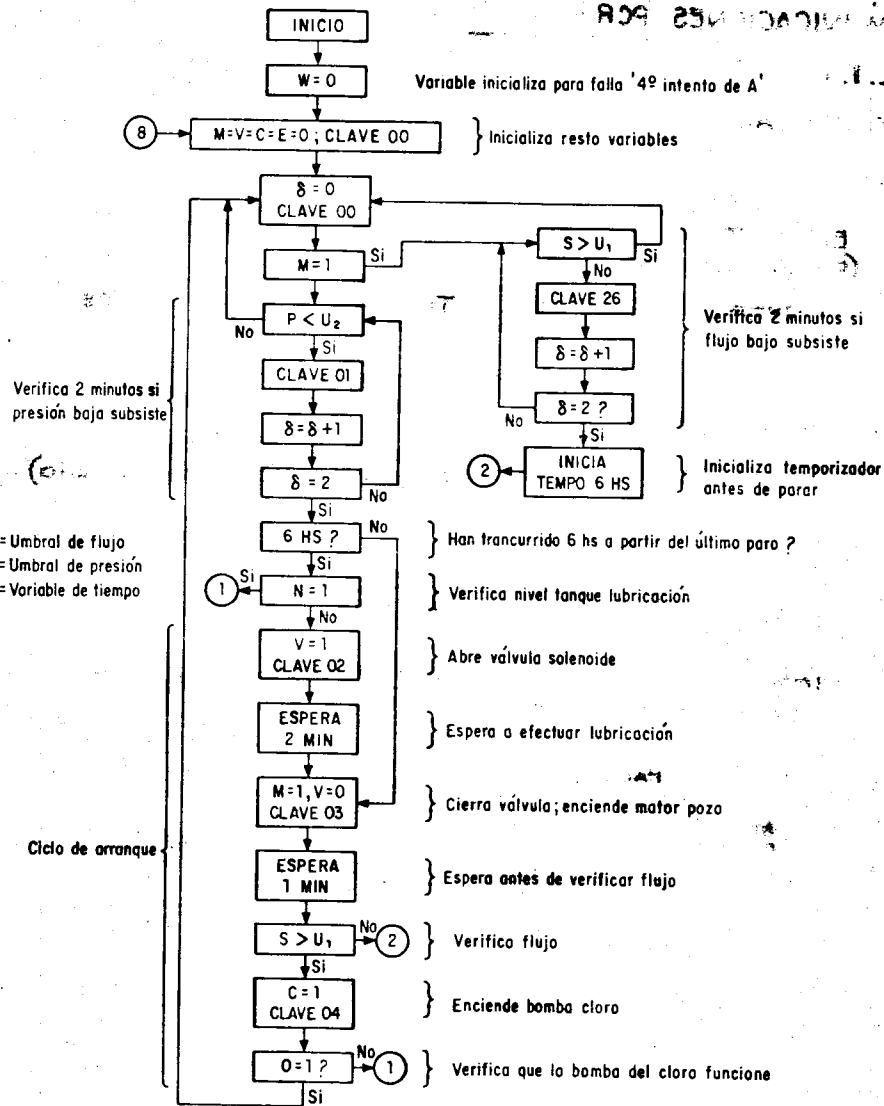


Fig 5 Diagrama de flujo, programa principal del SASCP

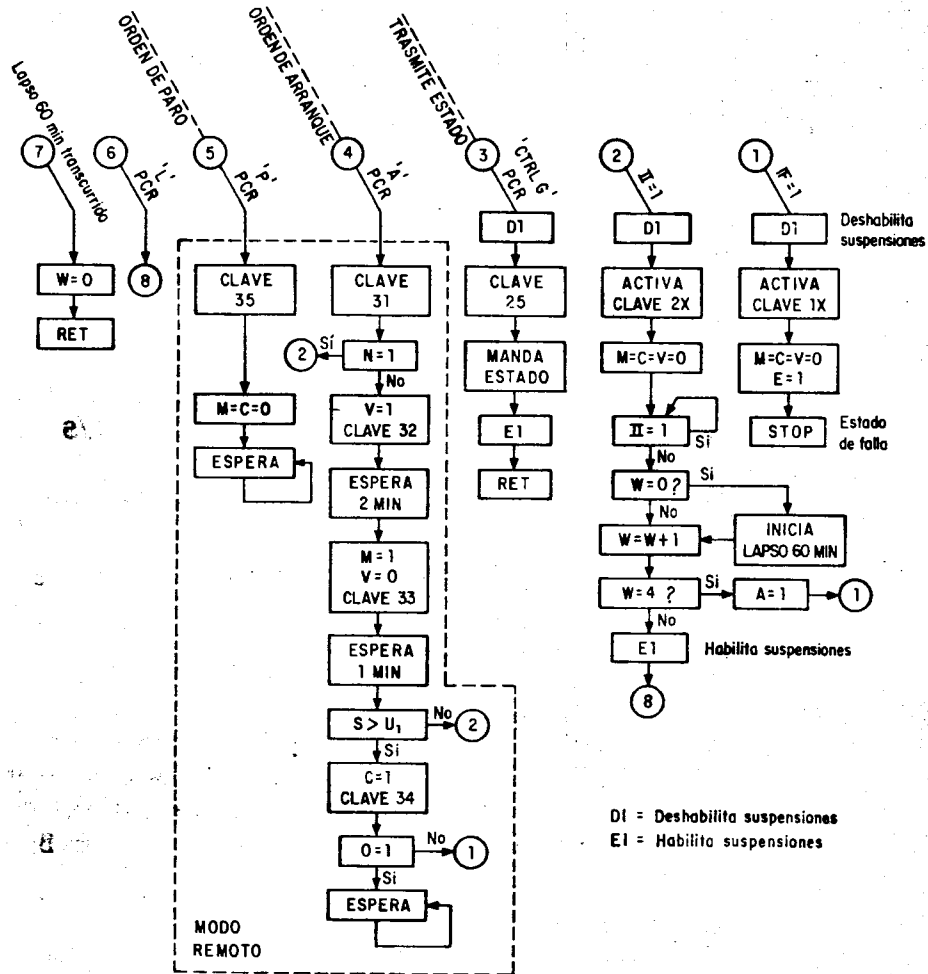


Fig 6 Diagrama de flujo, suspensiones en el programa del SASCP

DISEÑO ELECTRÓNICO DEL SISTEMA SASCP :

EN BASE A UN CONTROLADOR MODULAR  
PROGRAMABLE (DISEÑO DEL I.I.)

MÓDULOS EMPLEADOS :

- TARJETA CPU PAT 85 :
- Z80 CPU
  - PROM 8K
  - RAM 4K
  - PUERTO SERIE
  - 24 LINEAS DE E/S
- TARJETA MO - 1 :
- MÓDEM ASÍNCRONO  
300 BAUDS, HALF DUPLEX
  - CONVERSOR A/D DE  
8 CANALES
- BUS COMÚN :
- ÓMNIBUS
- SOPORTE DE PROGRAMACIÓN:-
- INTÉRPRETE BASIC  
RESIDENTE (4K PROM)
- FUENTE DE PODER FO-1 -
- CON SOPORTE DE BA-  
TERIA 12V, CON CAP.  
PARA 24 HS.

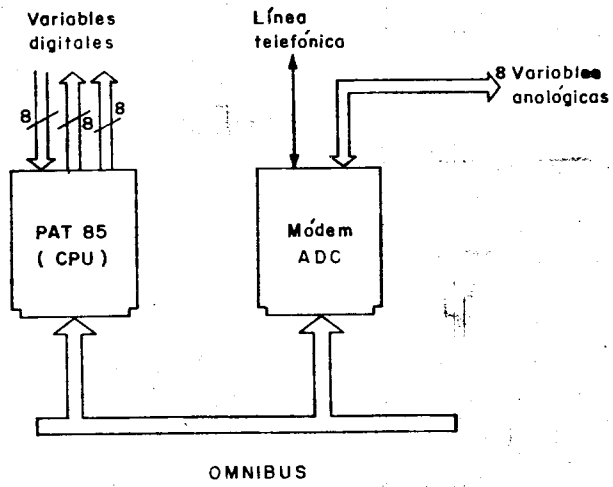


Fig. 2 Configuración del sistema "OMNIBUS" en el controlador SASCP

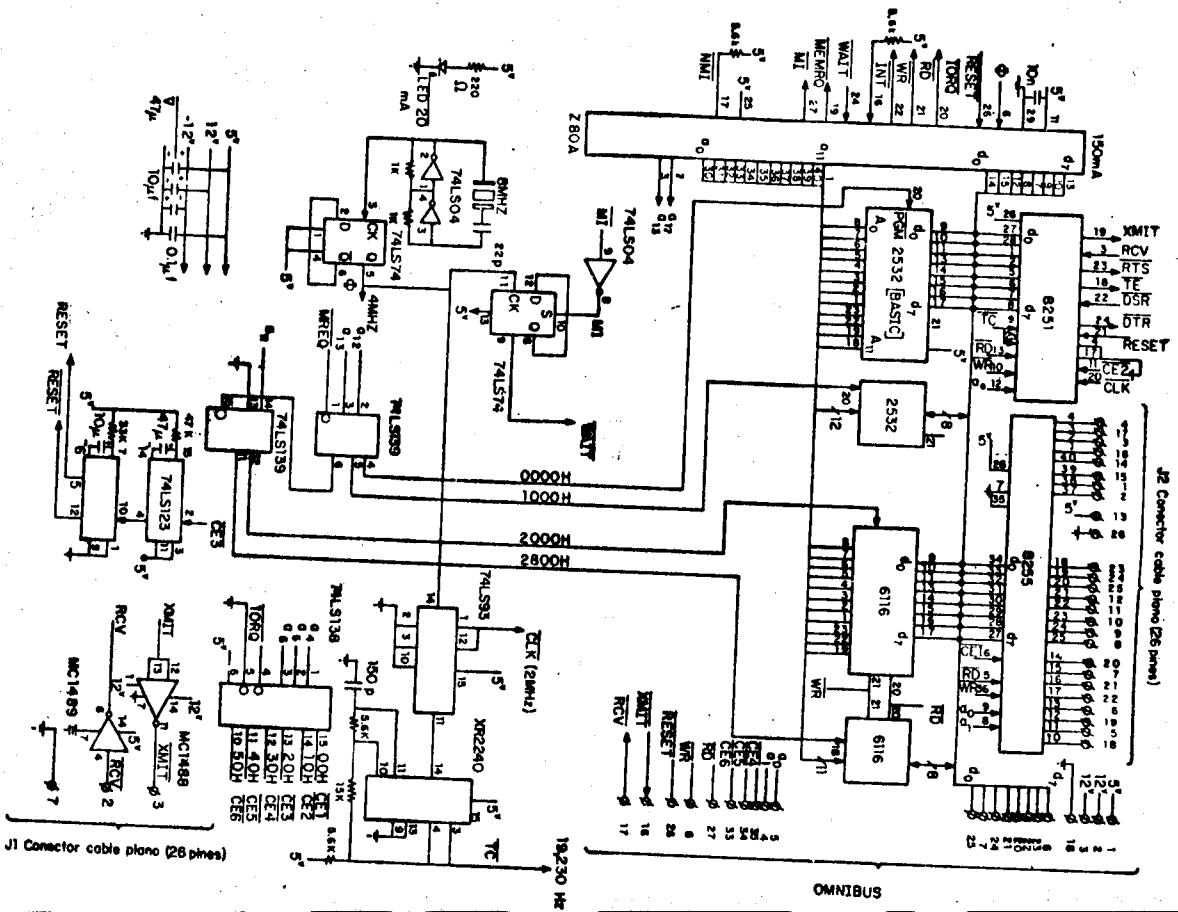
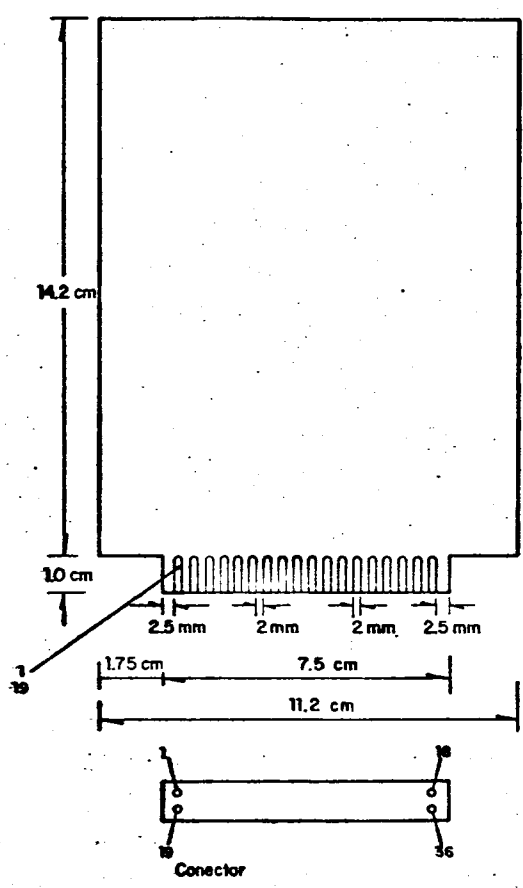


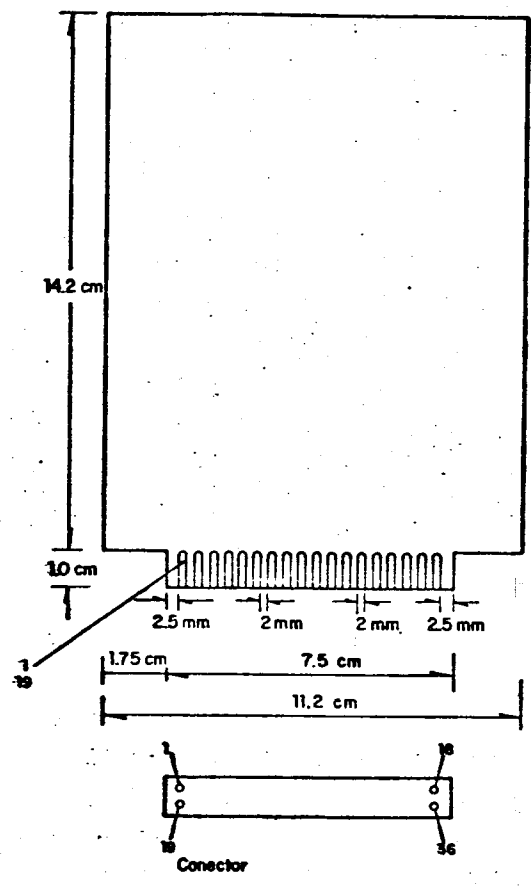
Fig A2 Sistema PAT 85



|    |                  |    |                |
|----|------------------|----|----------------|
| 1  | 5 <sup>v</sup>   | 19 | 5 <sup>v</sup> |
| 2  | 12 <sup>v</sup>  | 20 | d 4            |
| 3  | -12 <sup>v</sup> | 21 | d 3            |
| 4  | d 1              | 22 | d 5            |
| 5  | d 0              | 23 | d 6            |
| 6  | d 7              | 24 | d 2            |
| 7  | d 1              | 25 | d 0            |
| 8  | WR               | 26 | RESET          |
| 9  |                  | 27 | RD             |
| 10 | } Disponible     | 28 | } Disponible   |
| 11 |                  | 29 |                |
| 12 |                  | 30 |                |
| 13 |                  | 31 |                |
| 14 |                  | 32 |                |
| 15 |                  | 33 | CE 6           |
| 16 | XMIT             | 34 | CE 5           |
| 17 | RCV              | 35 | CE 4           |
| 18 | GND              | 36 | GND            |

las señales son  
provenientes de un Z80  
Los diagramas de tiempos  
consultarlos en los manuales

Fig A1 Módulo estándar OMNIBUS



|    |                  |    |                |
|----|------------------|----|----------------|
| 1  | 5 <sup>v</sup>   | 19 | 5 <sup>v</sup> |
| 2  | 12 <sup>v</sup>  | 20 | d 4            |
| 3  | -12 <sup>v</sup> | 21 | d 3            |
| 4  | d 1              | 22 | d 5            |
| 5  | d 0              | 23 | d 6            |
| 6  | d 7              | 24 | d 2            |
| 7  | d 1              | 25 | d 0            |
| 8  | WR               | 26 | RESET          |
| 9  |                  | 27 | RD             |
| 10 | } Disponible     | 28 | } Disponible   |
| 11 |                  | 29 |                |
| 12 |                  | 30 |                |
| 13 |                  | 31 |                |
| 14 |                  | 32 |                |
| 15 |                  | 33 | CE 6           |
| 16 | XMIT             | 34 | CE 5           |
| 17 | RCV              | 35 | CE 4           |
| 18 | GND              | 36 | GND            |

las señales son  
provenientes de un Z80  
Los diagramas de tiempos  
consultarlos en los manuales

Fig A1 Módulo estándar OMNIBUS

## EJERCICIO 1

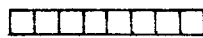
1. Haga las siguientes operaciones en binario, utilizando representación complemento a 2 en 8 bits. Indique si existe overflow. Todos los números son decimales.

$$\begin{array}{r} -58 \\ -110 \end{array}$$

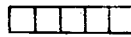
$$\begin{array}{r} 20 \\ -89 \end{array}$$

$$\begin{array}{r} 127 \\ -128 \end{array}$$

2. Con el formato indicado, halle la representación en binario de los números decimales dados



entero

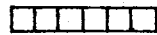


fracción

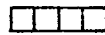
26.27

1.8

102.268

signo  
exp.

mantisa



exp

0.612

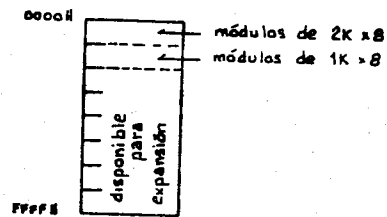
11,200.8

3. Haga un diagrama de flujo que detecta sobreflujo en una suma de números en representación complemento a 2. Asigne las variables A como bit del signo del primer sumando, B del segundo y C del resultado.
4. Escriba y codifique un programa para el Z80 que cuente del 0 al 99 en BCD, guardando el resultado de cada incremento en el registro C.

Calcule el tiempo de ejecución del programa si el  $\mu p$  se encuentra trabajando a 4 MHz.



5. Escriba y codifique, empleando solo direccionamiento relativo, un programa que, dado un número  $N$  del 0 al 7 en el registro  $C$ , se tenga como salida el bit  $d_N$  del acumulador encendido.
6. Diseñe un circuito decodificador para el siguiente mapa de memoria utilizando decodificadores 2-4 y un mínimo de compuertas adicionales.



7. Liste las características más importantes de la memoria TMS2532. ¿Cuál es el 'fan-out' para circuitos TTL y cual para circuitos MOS?

## EJERCICIO 2

Escriba en lenguaje ensamblador Z80 (no es necesario codificar), las siguientes subrutinas; especifique en todos los casos, la función, entradas, salidas, que registros se destruyen y cual es la dirección de inicio.

1. Función: Suma en BCD los números A y B. Doble precisión.  
Entradas: A y B en 800 - 803.  
Salidas: resultado en 804, 805  
Destruye:  
Dir. de inicio:
2. Obtiene la raíz cuadrada de un número A entero, resultado entero.
3. Convierte 2 caracteres ASCII a hexadecimal en un byte.
4. Realiza un retraso variable de  $x/256$  DEC. donde  $0 \leq x \leq 255$ .
5. Mueve un bloque de memoria de  $ZZ_{16}$  número de bytes, de la dirección  $XXXX_{16}$  a la  $YYYY_{16}$ , pudiendo existir un área de traslape entre ambos bloques.
6. Dado un BCD  $XX_{16}$ , encontrarlo (dar su dirección) en una tabla de 256 caracteres.
7. Codifique las siguientes instrucciones.

## Dirección

|     |      |             |
|-----|------|-------------|
| E1D | UNO: | JR Z, DOS   |
| :   | :    | :           |
| :   | :    | :           |
| E5A | DOS: | JR NZ, UNO. |

8. Se tiene una microcomputadora como la mostrada en la figura. El sistema cuenta con un pequeño sistema operativo residente en PROM y se comunica con el usuario a través de una terminal de video.

Además el sistema cuenta con un circuito temporizador cuyo contador se carga en la dirección 0AH. de E/S.

Cada vez que se carga el contador ocurre la siguiente secuencia de eventos:

- (a) El temporizador decrementa el contador una unidad cada 64  $\mu$ s.
- (b) Cuando el contenido del contador es cero, el temporizador activa la línea  $\overline{\text{INT}}$  del Z80.
- (c) Al recibir el reconocimiento de interrupción, el temporizador pone en el bus de datos el código CFH y regresa la línea de  $\overline{\text{INT}}$  a su estado inactivo.

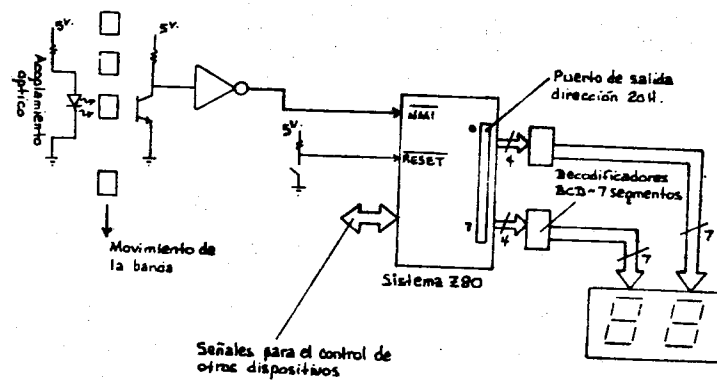
Escriba una subrutina de interrupción que, de manera completamente transparente al usuario, actualice en la terminal, en tiempo real, la hora.

No es necesario escribir la subrutina que actualice horas, minutos y segundos sino solamente la que lleva la base de tiempo.



10. En una fábrica de tornillos, se tiene una microcomputadora realizando diversas tareas de control en tiempo real. Una de estas tareas, que se efectúa por interrupción, concierne la cuenta de cajas que pasan por la línea de producción: un detector óptico genera un pulso cada vez que una caja que pasa, arrastrada por una banda, obstruye el haz luminoso de un acoplamiento óptico.

Escriba una subrutina de interrupción para el sistema especificado, que cuente los pulsos y escriba en unos módulos de 7 segmentos, el valor acumulado del  $00_{10}$  al  $99_{10}$ . Después de RESET, la cuenta debe ser  $00_{10}$ . La interrupción se genera con el flanco descendente del pulso.

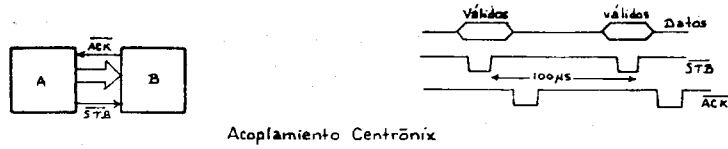


EJERCICIO 3

- Se requiere transferir 8K bytes de información almacenados en memoria en la computadora A, a la computadora B, que se encuentra a 5mts. de distancia.

Haga un análisis de un acoplamiento en serie y otro en paralelo especificando los siguientes puntos:

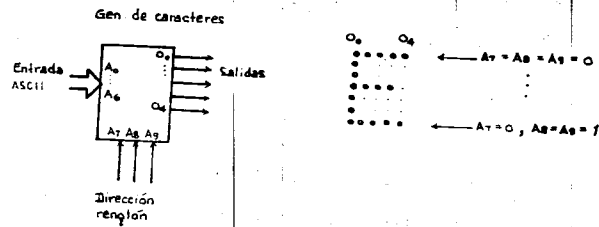
- Tiempo total de transmisión empleando un acoplamiento 'CENTRONIX' (vel. máx).
- Velocidad de transmisión serie (asíncrono) para igualar el tiempo de transmisión en (a).
- Circuitos y metros de cable empleados en cada caso.



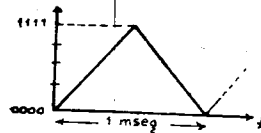
- La figura muestra un generador de caracteres para matriz de puntos de 5 x 7.

Para el código de la E (45H) llene la siguiente tabla.

| REGLON | DIRECCION<br>[en HEXADECIMAL] | CODIGO (0...0 <sub>4</sub> ) |
|--------|-------------------------------|------------------------------|
| 1      |                               |                              |
| 2      |                               |                              |
| 3      |                               |                              |
| 4      |                               |                              |
| 5      |                               |                              |
| 6      |                               |                              |
| 7      |                               |                              |



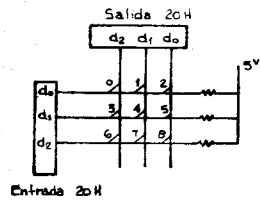
3. Un conversor de 'rastreo' de 4 bits debe seguir a una onda triangular de frecuencia fija. ¿Cuál es la frecuencia mínima de las fases  $\theta_1$  y  $\theta_2$ ?



4. Explique claramente el funcionamiento del circuito de la fig. 3.6 (PERIF P/MICROPROC), indicando la función de cada módulo.
5. Basándose en la información contenida en el reporte 'Control de CRT' diga cual es el valor de N de cada contador divisor/N mostrado en la fig. 3.
6. Escriba un programa para el  $\mu p$  Z80 que siga el algoritmo de aproximaciones sucesivas de la fig. 4.4 (PERIF P/ $\mu p$ ).

Suponga el circuito de la fig. 4.3 conectado a los puertos 10H de salida (bits  $d_0 - d_3$ ) y 20H de entrada (bit  $d_0$ ).

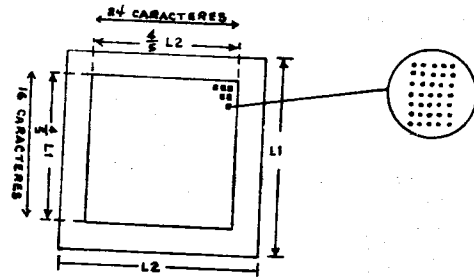
7. Explique de manera resumida, el principio de funcionamiento de los des\_ pliegues fosforescentes ITRON.
8. Escriba un programa que identifique que tecla se oprimió en un tecla- do de 9 teclas. La salida de la rutina debe ser el código ASCII de la tecla almacenado en el registro C. Empleé una tabla para hallar el cò- digo ASCII correspondiente.



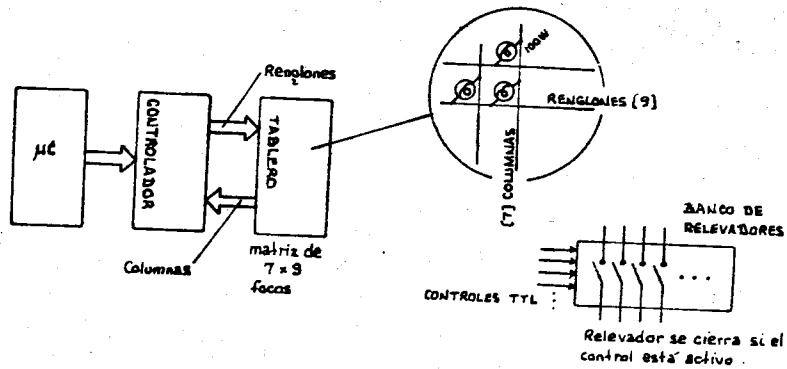


9. Se tiene una TV casera acoplada a un sistema de computación. Si se emplea un formato como el que se indica para la zona de escritura,

- a) ¿Cuál es el ancho de banda de la señal de video?
- b) Con referencia en la fig 8 del escrito 'Controlador de video' asigne, a las flechas anchas, el número de líneas, y a la RAM y ROM, su capacidad.

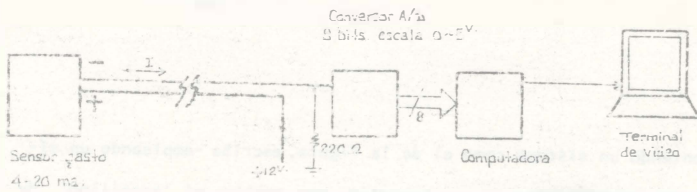


10. Diseñe el controlador de un tablero alfanumérico formado por una matriz de  $9 \times 7$  focos incandescentes de 100 watts c/u. Suponga disponibles bancos de  $n$  relevadores de estado sólido disparados por entradas TTL y capaces de manejar hasta 5KW cada uno. Especifique solo el diagrama de bloques.



## EJERCICIO 4

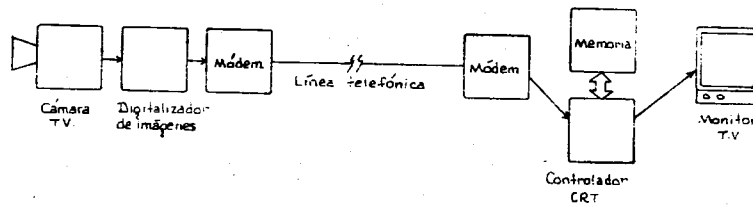
1. Dibuje el diagrama de bloques de un MODEM de baja velocidad, explique el funcionamiento de cada uno de los bloques y dé criterios básicos de diseño de cada uno de ellos.
2. Calcule el tiempo de transmisión del mensaje en la fig. "FORMATO DE TRANSMISION" del reporte "SIST. DIGITAL DE INFORMACION DE LA PLANTA DE BOMBEO DE XOTEPINGO".
3. El sistema de la figura mide constantemente el gasto de una tubería. El sensor proporciona una salida de corriente de 4 a 20 mA. El fabricante proporciona la fórmula indicada para el cálculo del gasto a partir de la corriente.  
Derive una fórmula que la computadora pueda procesar a partir de la lectura digital obtenida del conversor.



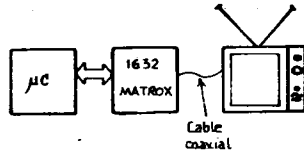
$$Q [M^3/S66] = 0.57 \sqrt{1.25 (I - 4)^3}$$

I en [mA]

4. Se tiene un sistema de transmisión de imágenes monocromáticas de TV. Cada imagen está formada por una matriz de  $256 \times 256$  puntos. Cada punto tiene una tonalidad de gris que se transmite codificada del 0 (negro) al 15 (blanco). La transmisión es digital, vía modem, y a 1,200 bauds, asíncrona. Cada palabra transmitida proporciona 8 bits de información. (Sin paridad).
- ¿Cuál es el período de actualización de las imágenes si se transmite a velocidad máxima?
  - ¿Qué capacidad debe tener la memoria de refresco en el receptor? (debe almacenar una imagen completa).



5. Suponiendo un sistema como el de la figura, escriba -empleando un mínimo de instrucciones - una subrutina que realice el 'scrolling' de la pantalla (todos los renglones se desplazan hacia arriba dejando en blanco el último renglón).



Problema 5

6. En una planta de productos químicos, se tiene un proceso de fabricación en donde se requiere de un sistema remoto de supervisión y control (SRSC) con las siguientes funciones:

- Adquisición de los valores de 8 variables analógicas.
- Despliegue de los valores reales de las variables en un monitor de TV.
- Encendido y apagado - a control remoto - de un motor síncrono.
- Comunicaciones por vía telefónica con una computadora central.

El sistema debe actualizar en el monitor de TV el valor de las variables por lo menos cada 20 segundos. El cálculo de los valores reales se hace con ayuda de las fórmulas proporcionadas por los fabricantes de los transductores.

Sobre petición de la computadora central, el SRSC tendrá capacidad de:

- Transmitir a la computadora central los valores de las ocho variables en forma codificada.
- Encender ó apagar el motor síncrono.

Considere que cualquier petición de la computadora central deberá ser atendida de inmediato.

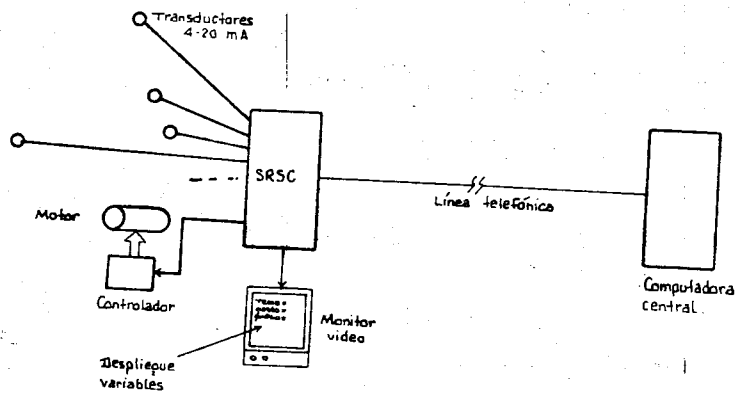
Considere ya disponibles:

- Los transductores de las 8 variables, los cuales generan una señal de 4 a 20 miliamperes.

-El controlador del motor síncrono, cuya única entrada es una señal TTL que automáticamente enciende o apaga el motor.

Haga un análisis completo del sistema incluyendo los siguientes puntos:

1. Diagrama general de bloques.
2. Diagrama detallado de bloques, incluyendo el acoplador a la línea telefónica. Especifique completamente todos los circuitos LSI que incluya en su diseño.
3. Diagrama de flujo del programa interno del sistema.
4. Tipo de línea telefónica, número de hilos, modo de transmisión y velocidad de transmisión.
5. Especificación del protocolo y formato de transmisión de mensajes y datos para la comunicación con la computadora central.
6. Evaluación aproximada del costo del sistema, incluyendo costos de diseño y construcción.



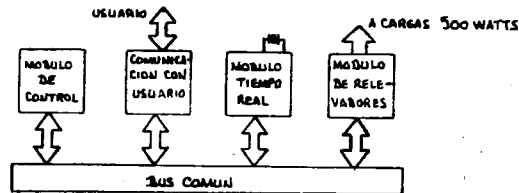


7. En una fábrica de productos plásticos se requiere de un temporizador industrial. Este equipo debe tener capacidad para encender y/o apagar cualesquiera de hasta 8 cargas de 500 watts (120 v. A.C.) cada una como máximo, en instantes previamente programados por el usuario y repetidos diariamente, con precisión de horas y minutos.

Las cargas pueden ser solenoides (porteros electrónicos, válvulas), lámparas de iluminación, motores síncronos (bombas, generadores), resistencias, etc.

Con base en el diagrama de bloques mostrado, diseñe un sistema basado en un microprocesador que cumpla con los requisitos anteriores detallando para cada bloque:

1. Diagrama electrónico, mostrando todos los circuitos empleados en cada bloque. Pueden usarse flechas anchas, sin necesidad de detallar cada conexión.
2. Diagrama de flujo del programa interno del sistema en el módulo de control. Recuerde que el sistema debe funcionar en 2 modos: modo 'programa' y modo 'control'. En el primero se comunica con el usuario y en el segundo ejecuta las instrucciones en las horas especificadas.
3. Costo de cada módulo y costo total del sistema, suponiendo la fabricación de 1 solo prototipo. No olvide incluir gastos de diseño, fabricación, gabinete, pruebas, indirectos, etc. No se asigne utilidades.



8. Un hospital con especialidad en atención a pacientes enfermos del corazón requiere de un equipo con las siguientes posibilidades:
1. Tomar muestras de la señal cardíaca del paciente (electrocardiograma) en su casa.
  2. Almacenar en algún medio los valores de la señal correspondientes a un minuto de muestreo.
  3. Transmitir esta señal por vía telefónica, para su recuperación y graficación en el hospital.

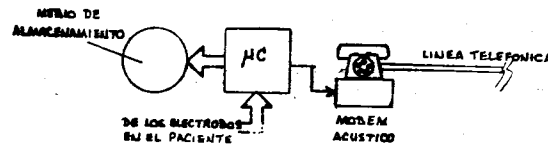
Posteriormente, ya en el mismo hospital, se haría la lectura directa del medio de almacenamiento, por una computadora que se encargaría de procesar la señal y dar eventualmente algún diagnóstico.

Suponga ya disponibles:

1. Los electrodos para la detección de las señales cardíacas
2. Los módems acústicos (half duplex, 300 bauds)

Diseñe el equipo que cumpla las especificaciones anteriores documentando

1. Diagrama general de bloques
2. Diagrama electrónico, detallando los circuitos importantes y utilizar flechas anchas para señalar su interconexión
3. Diagrama de flujo del programa interno del sistema
4. Análisis de la frecuencia de muestreo, capacidad de memoria y tiempo de transmisión para la transferencia del electrocardiograma vía telefónica
5. Costo total de un prototipo, incluyendo costos de diseño, material, pruebas, indirectos, etc. No se asigne utilidades
6. Análisis de el porqué se eligió el medio de almacenamiento y su comparación (costo, espacio, eficiencia) con otras alternativas.



En una fábrica de cuadernos se requiere automatizar el sistema que fabrica las espirales de plástico, que más tarde habrán de montarse en los cuadernos.

En la fig 1 se muestra el diagrama general del sistema. El sistema funciona de la siguiente forma: la espiga de plástico se extrae del extrusor con temperatura controlada. Esta espiga se enrolla sobre un eje que gira a velocidad constante, formando la espiral. A través de un actuador de solenoide se cortan finalmente las espiras a la longitud deseada.

Los dispositivos que debe controlar la microcomputadora son:

- a) El extrusor: el extrusor tiene una resistencia de 5,000 watts que es la que genera el calor para fundir el plástico. La temperatura se controla a través de un termopar, mediante un control ON-OFF.
- b) El solenoide de corte: este solenoide se energiza con un voltaje de 120 VAC, siendo su consumo de 200 watts. Al activarlo, su extremo afilado corta la espira.
- c) El motor de pesos: el motor es manejado por un controlador que requiere de pulsos con nivel TTL a su entrada. Cada pulso hace avanzar el motor 3.6°, de tal forma que con 100 pulsos se tiene una revolución completa. Las velocidades de giro varían de 0 a 100 vueltas por minuto.



d) El despliegue y teclado para programación: a través de este módulo, el usuario puede programar:

- la temperatura de operación del extrusor (20° a 200°C)
- la velocidad del motor (0 a 100 rev/min)
- la frecuencia de corte de las espiras. (0 a 10 seg)

Después de RESET, la microcomputadora debe realizar las siguientes funciones:

- a) Poner a funcionar el motor a la velocidad constante previamente programada.
- b) Controlar la temperatura en el punto de operación programado.
- c) Cerrar el solenoide con la frecuencia programada.
- d) Supervisar el módulo de despliegue y teclado de tal forma que el usuario pueda, en cualquier momento, introducir nuevos datos.

Para la introducción de nuevos datos el usuario debe oprimir una tecla 'P' de programación, de tal forma que la microcomputadora suspenda sus tareas de control y 'observe' únicamente el módulo de despliegue y teclado.

Para el reinicio del control de la operación del sistema debe oprimirse la tecla '0' de operación.

Diseñe la microcomputadora que habrá de cumplir con las especificaciones anteriores. Documente:

1. Diagrama de bloques del sistema de microcomputación. En este diagrama se deben especificar completamente todos los circuitos LSI y MSI. Utilice flechas anchas para indicar su interconexión.
2. Diagrama de flujo del programa del  $\mu p$ , incluyendo todas las funciones descritas en la hoja anterior.
3. Especificación completa del termopar y los relevadores que manejan al extrusor y al solenoide de corte, así como los acoplamientos de los mismos a la microcomputadora.
4. Diseño completo del módulo de despliegue y teclado, con diagrama electrónico y especificación de los módulos de 7 segmentos y teclado.
5. Análisis de costos incluyendo:
  - (a) costo directo
  - (b) costo de desarrollo e implantación
  - (c) costo indirecto (equipo, renta, local, etc.)

No se asigne utilidades.

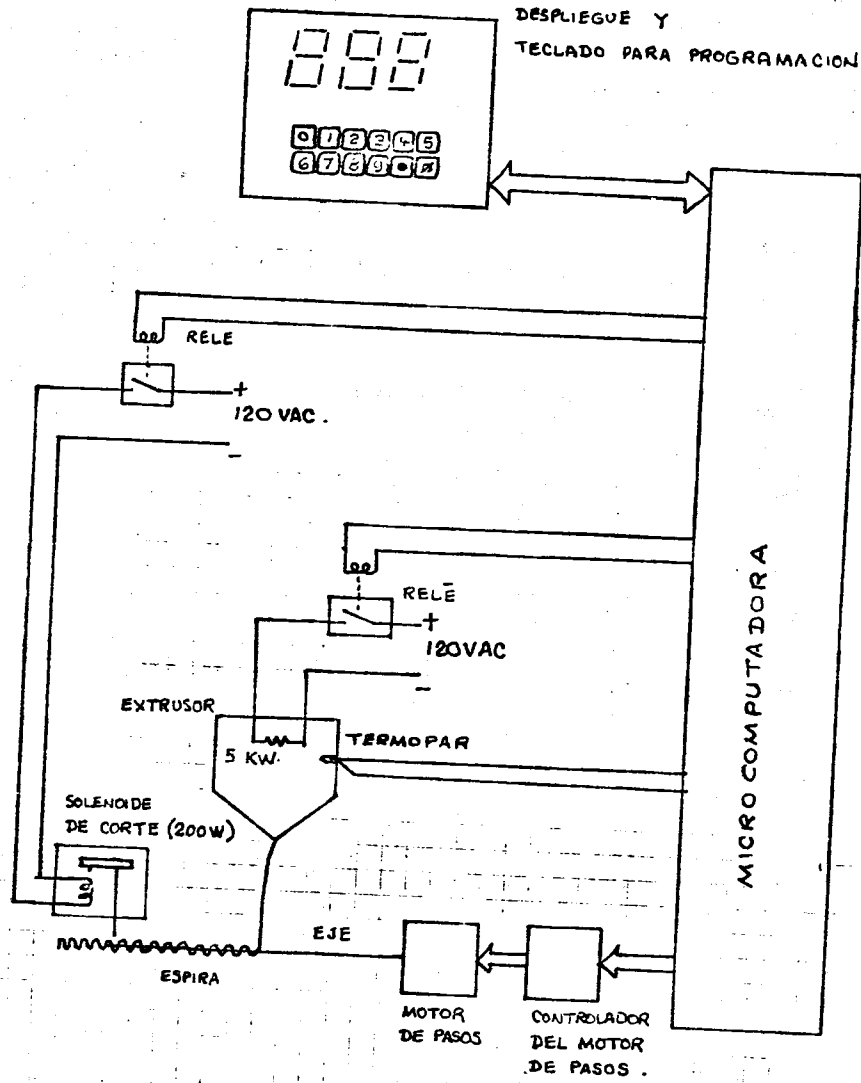


FIG 1 SISTEMA DE FABRICACION DE ESPIRALES DE PLASTICO

**MOSTEK**

Z80 MICROCOMPUTER DEVICES

**Technical Manual**

---

**MK 3880  
CENTRAL  
PROCESSING  
UNIT**

---

A block diagram of the internal architecture of the Z80-CPU is shown in Figure 2.0-1. The diagram shows all of the major elements in the CPU and it should be referred to throughout the following description.

### Z80-CPU BLOCK DIAGRAM

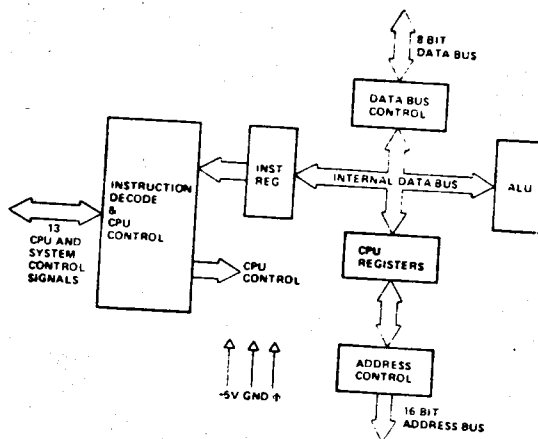


FIGURE 2.0-1

### 2.1 CPU REGISTERS

The Z80-CPU contains 208 bits of R/W memory that are accessible to the programmer. Figure 2.0-2 illustrates how this memory is configured into eighteen 8-bit registers and four 16-bit registers. All Z80 registers are implemented using static RAM. The registers include two sets of six general purpose registers that may be used individually as 8-bit registers or in pairs as 16-bit registers. There are also two sets of accumulator and flag registers.

#### Special Purpose Registers

1. **Program Counter (PC).** The program counter holds the 16-bit address of the current instruction being fetched from memory. The PC is automatically incremented after its contents have been transferred to the address lines. When a program jump occurs the new value is automatically placed in the PC, overriding the incrementer.
2. **Stack Pointer (SP).** The stack pointer holds the 16-bit address of the current top of a stack located anywhere in external system RAM memory. The external stack memory is organized as a last-in first-out (LIFO) file. Data can be pushed onto the stack from specific CPU registers or popped off of the stack into specific CPU registers through the execution of PUSH and POP instructions. The data popped from the stack is always the last data pushed onto it. The stack allows simple implementation of multiple level interrupts, unlimited subroutine nesting and simplification of many types of data manipulation.

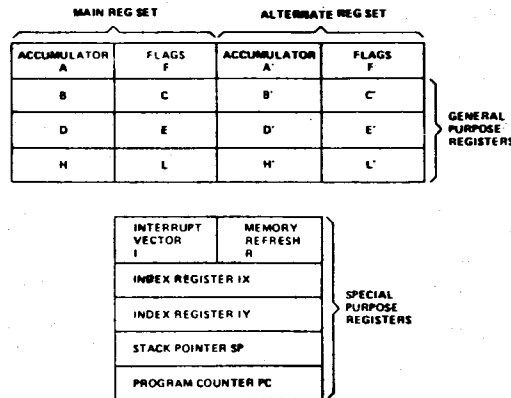


FIGURE 2.0-2

- Two Index Registers (IX & IY).** The two independent index registers hold a 16-bit base address that is used in indexed addressing modes. In this mode, an index register is used as a base to point to a region in memory from which data is to be stored or retrieved. An additional byte is included in indexed instructions to specify a displacement from this base. This displacement is specified as a two's complement signed integer. This mode of addressing greatly simplifies many types of programs, especially where tables of data are used.
- Interrupt Page Address Register (I).** The Z80-CPU can be operated in a mode where an indirect call to any memory location can be achieved in response to an interrupt. The I Register is used for this purpose to store the high order 8-bits of the indirect address while the interrupting device provides the lower 8-bits of the address. This feature allows interrupt routines to be dynamically located anywhere in memory with absolute minimal access time to the routine.
- Memory Refresh Register (R).** The Z80-CPU contains a memory refresh counter to enable dynamic memories to be used with the same ease as static memories. This 7-bit register is automatically incremented after each instruction fetch. The data in the refresh counter is sent out on the lower portion of the address bus along with a refresh control signal while the CPU is decoding and executing the fetched instruction. This mode of refresh is totally transparent to the programmer and does not slow down the CPU operation. The programmer can load the R register for testing purposes, but this register is normally not used by the programmer.

#### Accumulator and Flag Registers

The CPU includes two independent 8-bit accumulators and associated 8-bit flag registers. The accumulator holds the results of 8-bit arithmetic or logical operations while the flag register indicates specific conditions for 8 or 16-bit operations, such as indicating whether or not the result of an operation is equal to zero. The programmer selects the accumulator and flag pair that he wishes to work with with a single exchange instruction so that he may easily work with either pair.

## General Purpose Registers

184

There are two matched sets of general purpose registers, each set containing six 8-bit registers that may be used individually as 8-bit registers or as 16-bit register pairs by the programmer. One set is called BC, DE, and HL while the complementary set is called BD, DE' and HL'. At any one time the programmer can select either set of registers to work with through a single exchange command for the entire set. In systems where fast interrupt response is required, one set of general purpose registers and an accumulator/flag register may be reserved for handling this very fast routine. Only a simple exchange command need be executed to go between the routines. This greatly reduces interrupt service time by eliminating the requirement for saving and retrieving register contents in the external stack during interrupt or subroutine processing. These general purpose registers are used for a wide range of applications by the programmer. They also simplify programming, especially in ROM based systems where little external read/write memory is available.

## 2.2 ARITHMETIC & LOGIC UNIT (ALU)

The 8-bit arithmetic and logical instructions of the CPU are executed in the ALU. Internally the ALU communicates with the registers and the external data bus on the internal data bus. The type of functions performed by the ALU include:

|                      |  |
|----------------------|--|
| Add                  | Left or right shifts or rotates (arithmetic and logical) |
| Subtract             | Increment  |
| Logical AND          | Decrement  |
| Logical OR           | Set bit  |
| Logical Exclusive OR | Reset bit  |
| Compare              | Test bit   |

## 2.3 INSTRUCTION REGISTER AND CPU CONTROL

As each instruction is fetched from memory, it is placed in the instruction register and decoded. The control section performs this function and then generates and supplies all of the control signals necessary to read or write data from or to the registers, controls the ALU and provides all required external control signals.

The Z80-CPU is packaged in an industry standard 40 pin Dual In-Line Package. The I/O pins are shown in Figure 3.0-1 and the function of each is described below.

80 PIN CONFIGURATION

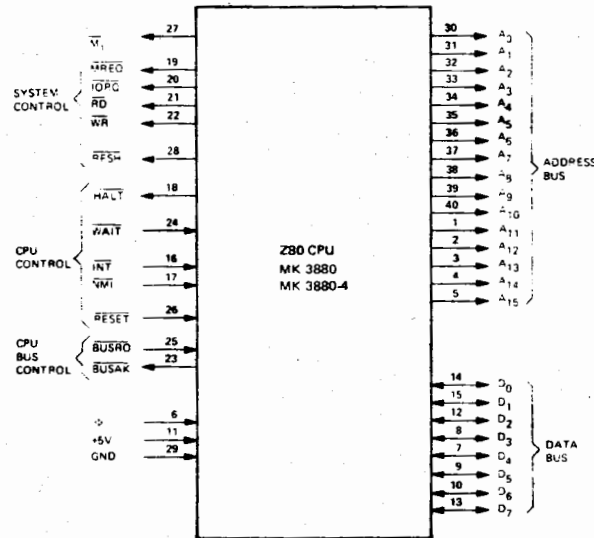


FIGURE 3.0-1

**A<sub>0</sub>-A<sub>15</sub>**  
(Address Bus)

Tri-state output, active high. A<sub>0</sub>-A<sub>15</sub> constitute a 16-bit address bus. The address bus provides the address for memory (up to 64K bytes) data exchanges and for I/O device data exchanges. I/O addressing uses the 8 lower address bits to allow the user to directly select up to 256 input or 256 output ports. A<sub>0</sub> is the least significant address bit. During refresh time, the lower 7 bits contain a valid refresh address.

**D<sub>0</sub>-D<sub>7</sub>**  
(Data Bus)

Tri-state input/output, active high. D<sub>0</sub>-D<sub>7</sub> constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.

**M<sub>1</sub>**  
(Machine Cycle one)

Output, active low. M<sub>1</sub> indicates that the current machine cycle is the OP code fetch cycle of an instruction execution. Note that during execution of 2-byte op-codes, M<sub>1</sub> is generated as each op code byte is fetched. These two byte op-codes always begin with CBH, DDH, EDH, or FDH. M<sub>1</sub> also occurs with IORQ to indicate an interrupt acknowledge cycle.

**MREQ**  
(Memory Request)

Tri-state output, active low. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.



- $\overline{IORQ}$**   
(Input/Output Request) Tri-state output, active low. The  $\overline{IORQ}$  signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An  $\overline{IORQ}$  signal is also generated with an  $\overline{M1}$  signal when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus. Interrupt Acknowledge operations occur during  $M_1$  time while I/O operations never occur during  $M_1$  time.
- $\overline{RD}$**   
(Memory Read) Tri-state output, active low.  $\overline{RD}$  indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.
- $\overline{WR}$**   
(Memory Write) Tri-state output, active low.  $\overline{WR}$  indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.
- $\overline{RFSH}$**   
(Refresh) Output, active low.  $\overline{RFSH}$  indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and current  $\overline{MREQ}$  signal should be used to do a refresh read to all dynamic memories.  $A_7$  is a logic zero and the upper 8 bits of the Address Bus contains the I Register.
- $\overline{HALT}$**   
(Halt state) Output, active low.  $\overline{HALT}$  indicates that the CPU has executed a HALT software instruction and is awaiting either a non maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOP's to maintain memory refresh activity.
- $\overline{WAIT}^*$**   
(Wait) Input, active low.  $\overline{WAIT}$  indicates to the Z80-CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active. This signal allows memory or I/O devices of any speed to be synchronized to the CPU.
- $\overline{INT}$**   
(Interrupt Request) Input, active low. The Interrupt Request signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled and if the  $\overline{BUSRQ}$  signal is not active. When the CPU accepts the interrupt, an acknowledge signal ( $\overline{IORQ}$  during  $M_1$  time) is sent out at the beginning of the next instruction cycle. The CPU can respond to an interrupt in three different modes that are described in detail in section 8.
- $\overline{NMI}$**  Input, negative edge triggered. The non maskable interrupt request line has a higher priority than  $\overline{INT}$  and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop.  $\overline{NMI}$  automatically forces the Z80-CPU to restart to location 0066H. The program counter is automatically saved in the external stack so that the user can return to the program that was interrupted. Note that continuous  $\overline{WAIT}$  cycles can prevent the current instruction from ending, and that a  $\overline{BUSRQ}$  will override a  $\overline{NMI}$ .

**RESET**

Input, active low. **RESET** forces the program counter to zero and initializes the CPU. The CPU initialization includes:

127

- 1) Disable the interrupt enable flip-flop
- 2) Set Register I = 00H
- 3) Set Register R = 00H
- 4) Set Interrupt Mode 0

During reset time, the address bus and data bus go to a high impedance state and all control output signals go to the inactive state. No refresh occurs.

**BUSRQ**

(Bus Request)

Input, active low. The bus request signal is used to request the CPU address bus, data bus and tri-state output control signals to go to a high impedance state so that other devices can control these buses. When **BUSRQ** is activated, the CPU will set these buses to a high impedance state as soon as the current CPU machine cycle is terminated.

**BUSAK\***

(Bus Acknowledge)

Output, active low. Bus acknowledge is used to indicate to the requesting device that the CPU address bus, data bus and tri-state control bus signals have been set to their high impedance state and the external device can now control these signals.

$\Phi$

Single phase system clock.

\*While the Z80-CPU is in either a **WAIT** state or a Bus Acknowledge condition, Dynamic Memory Refresh will not occur.

The Z80-CPU executes instructions by stepping through a very precise set of a few basic operations. These include:

- Memory read or write
- I/O device read or write
- Interrupt acknowledge

All instructions are merely a series of these basic operations. Each of these basic operations can take from three to six clock periods to complete or they can be lengthened to synchronize the CPU to the speed of external devices. The basic clock periods are referred to as T states and the basic operations are referred to as M (for machine) cycles. Figure 4.0.0 illustrates how a typical instruction will be merely a series of specific M and T cycles. Notice that this instruction consists of three machine cycles (M1, M2 and M3). The first machine cycle of any instruction is a fetch cycle which is four, five or six T states long (unless lengthened by the wait signal which will be fully described in the next section). The fetch cycle (M1) is used to fetch the OP code of the next instruction to be executed. Subsequent machine cycles move data between the CPU and memory or I/O devices and they may have anywhere from three to five T cycles (again they may be lengthened by wait states to synchronize the external devices to the CPU). The following paragraphs describe the timing which occurs within any of the basic machine cycles. In section 7, the exact timing for each instruction is specified.

#### BASIC CPU TIMING EXAMPLE

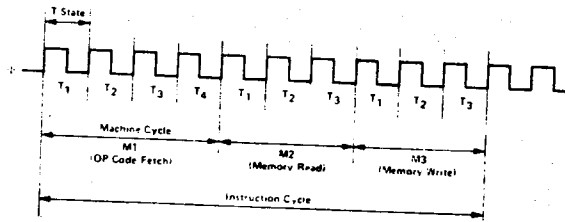


FIGURE 4.0.0

All CPU timing can be broken down into a few very simple timing diagrams as shown in Figure 4.0.1 through 4.0.7. These diagrams show the following basic operations with and without wait states (wait states are added to synchronize the CPU to slow memory or I/O devices).

- 4.0-1. Instruction OP code fetch (M1 cycle)
- 4.0-2. Memory data read or write cycles
- 4.0-3. I/O read or write cycles
- 4.0-4. Bus Request/Acknowledge Cycle
- 4.0-5. Interrupt Request/Acknowledge Cycle
- 4.0-6. Non maskable Interrupt Request/Acknowledge Cycle
- 4.0-7. Exit from a HALT instruction

Figure 4.0-1 shows the timing during an M1 cycle (OP code fetch). Notice that the PC is placed on the address bus at the beginning of the M1 cycle. One half clock time later the  $\overline{MREQ}$  signal goes active. At this time the address to the memory has had time to stabilize so that the falling edge of  $\overline{MREQ}$  can be used directly as a chip enable clock to dynamic memories. The  $\overline{RD}$  line also goes active to indicate that the memory read data should be enabled onto the CPU data bus. The CPU samples the data from the memory on the data bus with the rising edge of the clock of state T3 and this same edge is used by the CPU to turn off the  $\overline{RD}$  and  $\overline{MREQ}$  signals. Thus the data has already been sampled by the CPU before the  $\overline{RD}$  signal becomes inactive. Clock state T3 and T4 of a fetch cycle are used to refresh dynamic memories. (The CPU uses this time to decode and execute the fetched instruction so that no other operation could be performed at this time). During T3 and T4 the lower 7 bits of the address bus contain a memory refresh address and the  $\overline{RFSH}$  signal becomes active to indicate that a refresh read of all dynamic memories should be accomplished. Notice that a  $\overline{RD}$  signal is not generated during refresh time to prevent data from different memory segments from being gated onto the data bus. The  $\overline{MREQ}$  signal during refresh time should be used to perform a refresh read of all memory elements. The refresh signal can not be used by itself since the refresh address is only guaranteed to be stable during  $\overline{MREQ}$  time.

## INSTRUCTION OP CODE FETCH

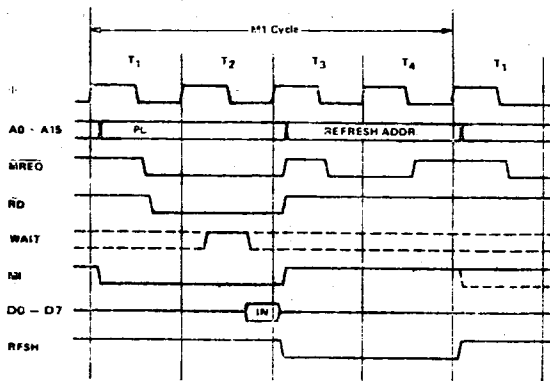


FIGURE 4.0-1

Figure 4.0-1A illustrates how the fetch cycle is delayed if the memory activates the  $\overline{WAIT}$  line. During T2 and every subsequent Tw, the CPU samples the  $\overline{WAIT}$  line with the falling edge of  $\Phi$ . If the  $\overline{WAIT}$  line is active at this time, another wait state will be entered during the following cycle. Using this technique the read cycle can be lengthened to match the access time of any type of memory device.

**INSTRUCTION OP CODE FETCH WITH WAIT STATES**

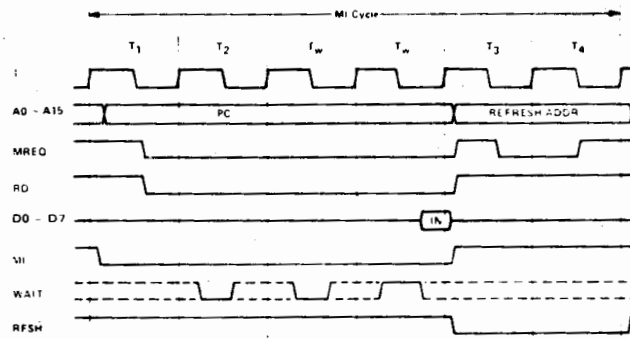


FIGURE 4.0-1A

**MEMORY READ OR WRITE**

Figure 4.0-2 illustrates the timing of memory read or write cycles other than an OP code fetch (M1 cycle). These cycles are generally three clock periods long unless wait states are requested by the memory via the WAIT signal. The MREQ signal and the RD signal are used the same as in the fetch cycle. In the case of a memory write cycle, the MREQ also becomes active when the address bus is stable so that it can be used directly as a chip enable for dynamic memories. The WR line is active when data on the data bus is stable so that it can be used directly as a R/W pulse to virtually any type of semiconductor memory. Furthermore the WR signal goes inactive one half T state before the address and data bus contents are changed so that the overlap requirements for virtually any type of semiconductor memory type will be met.

**MEMORY READ OR WRITE CYCLES**

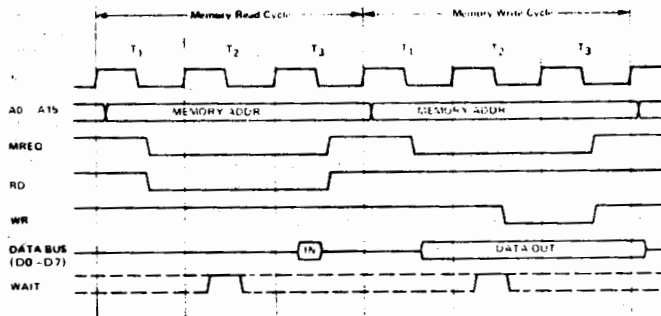


FIGURE 4.0-2

Figure 4.0-2A illustrates how a  $\overline{\text{WAIT}}$  request signal will lengthen any memory read or write operation. This operation is identical to that previously described for a fetch cycle. Notice in this figure that a separate read and a separate write cycle are shown in the same figure although read and write cycles can never occur simultaneously.

#### MEMORY READ OR WRITE CYCLES WITH WAIT STATES

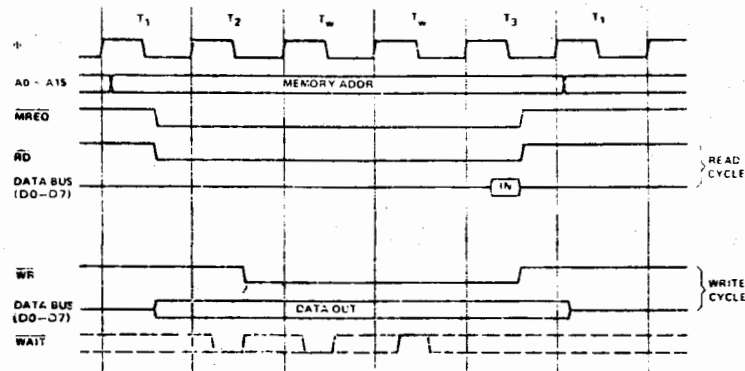


FIGURE 4.0-2A

#### INPUT OR OUTPUT CYCLES

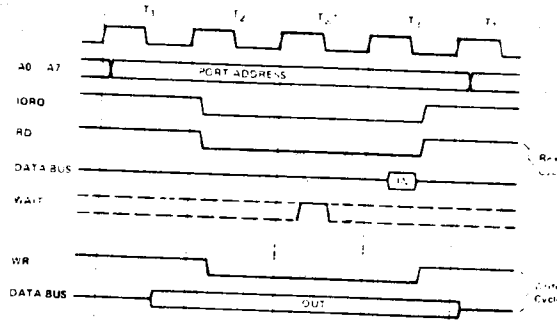
Figure 4.0-3 illustrates an I/O read or I/O write operation. Notice that during I/O operations a single wait state is automatically inserted. The reason for this is that during I/O operations, the time from when the  $\overline{\text{IORQ}}$  signal goes active until the CPU must sample the  $\overline{\text{WAIT}}$  line is very short and without this extra state sufficient time does not exist for an I/O port to decode its address and activate the  $\overline{\text{WAIT}}$  line if a wait is required. Also, without this wait state it is difficult to design MOS I/O devices that can operate at full CPU speed. During this wait state time the  $\overline{\text{WAIT}}$  request signal is sampled. During a read I/O operation, the  $\overline{\text{RD}}$  line is used to enable the addressed port onto the data bus just as in the case of a memory read. For I/O write operations, the  $\overline{\text{WR}}$  line is used as a clock to the I/O port, again with sufficient overlap timing automatically provided so that the rising edge may be used as a data clock.

Figure 4.0-3A illustrates how additional wait states may be added with the  $\overline{\text{WAIT}}$  line. The operation is identical to that previously described.

#### BUS REQUEST/ACKNOWLEDGE CYCLE

Figure 4.0-4 illustrates the timing for a Bus Request/Acknowledge cycle. The  $\overline{\text{BUSRQ}}$  signal is sampled by the CPU with the rising edge of the last clock period of any machine cycle. If the  $\overline{\text{BUSRQ}}$  signal is active, the CPU will set its address, data and tri-state control signals to the high impedance state with the rising edge of the next clock pulse. At that time any external device can control the buses to transfer data between memory and I/O devices. (This is generally known as Direct Memory Access (DMA) using cycle stealing). The maximum time for the CPU to respond to a bus request is the length of a machine cycle and the external controller can maintain control of the bus for as many clock cycles as is desired. Note, however, that if very long DMA cycles are used, and dynamic memories are being used, the external controller must also perform the refresh function. This situation only occurs if very large blocks of data are transferred under DMA control. Also note that during a bus request cycle, the CPU cannot be interrupted by either a  $\overline{\text{NMI}}$  or an  $\overline{\text{INT}}$  signal.

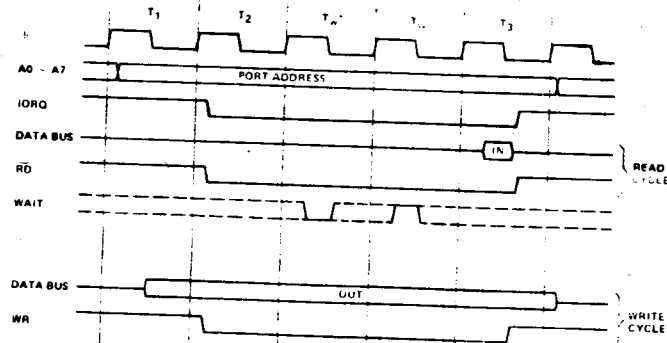
### INPUT OR OUTPUT CYCLES



\*Inserted by Z80 CPU

FIGURE 4.0-3

### INPUT OR OUTPUT CYCLES WITH WAIT STATES



\*Inserted by Z80 CPU

FIGURE 4.0-3A

BUS REQUEST/ACKNOWLEDGE CYCLE

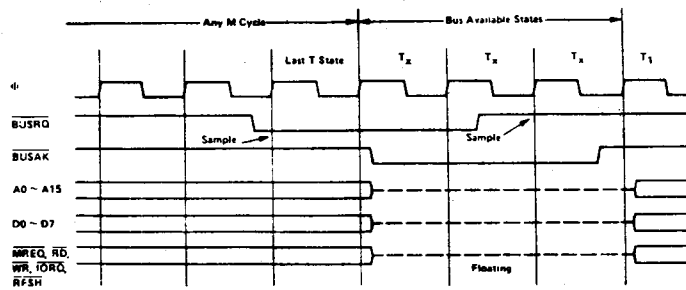


FIGURE 4.0-4

INTERRUPT REQUEST/ ACKNOWLEDGE CYCLE

Figure 4.0-5 illustrates the timing associated with an interrupt cycle. The interrupt signal ( $\overline{INT}$ ) is sampled by the CPU with the rising edge of the last clock at the end of any instruction. The signal will not be accepted if the internal CPU software controlled interrupt enable flip-flop is not set or if the  $\overline{BUSRQ}$  signal is active. When the signal is accepted a special M1 cycle is generated. During this special M1 cycle the  $\overline{IORQ}$  signal becomes active (instead of the normal  $\overline{MREQ}$ ) to indicate that the interrupting device can place an 8-bit vector on the data bus. Notice that two wait states are automatically added to this cycle. These states are added so that a ripple priority interrupt scheme can be easily implemented. The two wait states allow sufficient time for the ripple signals to stabilize and identify which I/O device must insert the response vector. Refer to section 8.0 for details on how the interrupt response vector is utilized by the CPU.

INTERRUPT REQUEST/ACKNOWLEDGE CYCLE

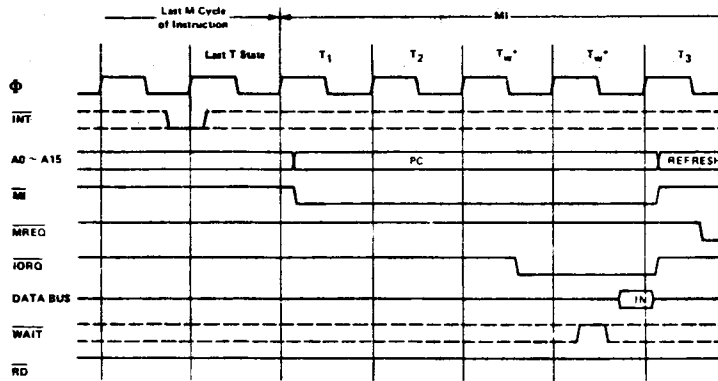


FIGURE 4.0-5



## 6.0 FLAGS

194

Each of the two Z80-CPU Flag registers contains six bits of information which are set or reset by various CPU operations. Four of these bits are testable; that is, they are used as conditions for jump, call or return instructions. For example a jump may be desired only if a specific bit in the flag register is set. The four testable flag bits are:

- 1) Carry Flag (C) – This flag is the carry from the highest order bit of the accumulator. For example, the carry flag will be set during an add instruction where a carry from the highest bit of the accumulator is generated. This flag is also set if a borrow is generated during a subtraction instruction. The shift and rotate instructions also affect this bit.
- 2) Zero Flag (Z) – This flag is set if the result of the operation loaded a zero into the accumulator. Otherwise it is reset.
- 3) Sign Flag(S) – This flag is intended to be used with signed numbers and it is set if the result of the operation was negative. Since bit 7 (MSB) represents the sign of the number (A negative number has a 1 in bit 7), this flag stores the state of bit 7 in the accumulator.
- 4) Parity/Overflow Flag(P/V) – This dual purpose flag indicates the parity of the result in the accumulator when logical operations are performed (such as AND A, B) and it represents overflow when signed two's complement arithmetic operations are performed. The Z80 overflow flag indicates that the two's complement number in the accumulator is in error since it has exceeded the maximum possible (+127) or is less than the minimum possible (-128) number that can be represented two's complement notation. For example consider adding:

$$\begin{array}{r} +120 = 0111\ 1000 \\ +105 = 0110\ 1001 \\ \hline C = 0\ 1110\ 0001 = -95 \text{ (wrong) Overflow has occurred;} \end{array}$$

Here the result is incorrect. Overflow has occurred and yet there is no carry to indicate an error. For this case the overflow flag would be set. Also consider the addition of two negative numbers:

$$\begin{array}{r} -5 = 1111\ 1011 \\ -16 = 1111\ 0000 \\ \hline C = 1\ 1110\ 1011 = -21 \text{ correct} \end{array}$$

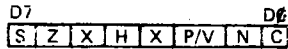
Notice that the answer is correct but the carry is set so that this flag can not be used as an overflow indicator. In this case the overflow would not be set.

For logical operations (AND, OR, XOR) this flag is set if the parity of the result is even and it is reset if it is odd.

There are also two non-testable bits in the flag register. Both of these are used for BCD arithmetic. They are:

- 1) Half carry(H) – This is the BCD carry or borrow result from the least significant four bits of operation. When using the DAA (Decimal Adjust Instruction) this flag is used to correct the result of a previous packed decimal add or subtract.
- 2) Add/Subtract Flag (N) – Since the algorithm for correcting BCD operations is different for addition or subtraction, this flag is used to specify what type of instruction was executed last so that the DAA operation will be correct for either addition or subtraction.

The Flag register can be accessed by the programmer and its format is as follows:



X means flag is indeterminate.

Table 6.0-1 lists how each flag bit is affected by various CPU instructions. In this table a '0' indicates that the instruction does not change the flag, an 'X' means that the flag goes to an indeterminate state, an 'O' means that it is reset, a '1' means that it is set and the symbol  $\downarrow$  indicates that it is set or reset according to the previous discussion. Note that any instruction not appearing in this table does not affect any of the flags.

Table 6.0-1 includes a few special cases that must be described for clarity. Notice that the block search instruction sets the Z flag if the last compare operation indicated a match between the source and the accumulator data. Also, the parity flag is set if the byte counter (register pair BC) is not equal to zero. This same use of the parity flag is made with the block move instructions. Another special case is during block input or output instructions, here the Z flag is used to indicate the state of register B which is used as a byte counter. Notice that when the I/O block transfer is complete, the zero flag will be reset to a zero (i.e. B=0) while in the case of a block move command the parity flag is reset when the operation is complete. A final case is when the refresh or I register is loaded into the accumulator, the interrupt enable flip flop is loaded into the parity flag so that the complete state of the CPU can be saved at any time.

8-BIT LOAD GROUP

| Mnemonic     | Symbolic Operation | Flags |   |   |     | Op-Code |     |    |     | No. of Bytes | No. of M Cycles | No. of T States | Comments |     |     |           |  |
|--------------|--------------------|-------|---|---|-----|---------|-----|----|-----|--------------|-----------------|-----------------|----------|-----|-----|-----------|--|
|              |                    | S     | Z | H | P/V | N       | C   | 76 | 543 |              |                 |                 |          | 210 | Hex |           |  |
| LD r, s      | r ← s              | .     | . | X | .   | X       | .   | .  | .   | 01           | r               | s               | 1        | 1   | 4   | r, s Reg. |  |
| LD r, n      | r ← n              | .     | . | X | .   | X       | .   | .  | .   | 00           | r               | n               | 2        | 2   | 7   | 000 B     |  |
| LD r, (HL)   | r ← (HL)           | .     | . | X | .   | X       | .   | .  | .   | 01           | r               | (HL)            | 1        | 2   | 7   | 010 C     |  |
| LD r, (IX+d) | r ← (IX+d)         | .     | . | X | .   | X       | .   | .  | .   | 11           | 011             | 101             | 3        | 5   | 19  | 011 D     |  |
|              |                    |       |   |   |     |         |     |    |     | 01           | r               | (IX+d)          |          |     |     | 100 E     |  |
|              |                    |       |   |   |     |         |     |    |     | -            | d               | -               |          |     |     | 101 H     |  |
| LD r, (IY+d) | r ← (IY+d)         | .     | . | X | .   | X       | .   | .  | .   | 11           | 111             | 101             | 3        | 5   | 19  | 101 L     |  |
|              |                    |       |   |   |     |         |     |    |     | 01           | r               | (IY+d)          |          |     |     | 111 A     |  |
|              |                    |       |   |   |     |         |     |    |     | -            | d               | -               |          |     |     |           |  |
| LD (HL), r   | (HL) ← r           | .     | . | X | .   | X       | .   | .  | .   | 01           | 110             | r               | 1        | 2   | 7   |           |  |
| LD (IX+d), r | (IX+d) ← r         | .     | . | X | .   | X       | .   | .  | .   | 11           | 011             | 101             | 3        | 5   | 19  |           |  |
|              |                    |       |   |   |     |         |     |    |     | 01           | 110             | r               |          |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | -            | d               | -               |          |     |     |           |  |
| LD (IY+d), r | (IY+d) ← r         | .     | . | X | .   | X       | .   | .  | .   | 11           | 111             | 101             | 3        | 5   | 19  |           |  |
|              |                    |       |   |   |     |         |     |    |     | 01           | 110             | r               |          |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | -            | d               | -               |          |     |     |           |  |
| LD (HL), n   | (HL) ← n           | .     | . | X | .   | X       | .   | .  | .   | 00           | 110             | 110             | 36       | 2   | 3   | 10        |  |
| LD (IX+d), n | (IX+d) ← n         | .     | . | X | .   | X       | .   | .  | .   | 11           | 011             | 101             | 4        | 5   | 19  |           |  |
|              |                    |       |   |   |     |         |     |    |     | 00           | 110             | 110             |          |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | -            | d               | -               |          |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | -            | n               | -               |          |     |     |           |  |
| LD (IY+d), n | (IY+d) ← n         | .     | . | X | .   | X       | .   | .  | .   | 11           | 111             | 101             | 4        | 5   | 19  |           |  |
|              |                    |       |   |   |     |         |     |    |     | 00           | 110             | 110             |          |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | -            | d               | -               |          |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | -            | n               | -               |          |     |     |           |  |
| LD A, (BC)   | A ← (BC)           | .     | . | X | .   | X       | .   | .  | .   | 00           | 001             | 010             | 0A       | 1   | 2   | 7         |  |
| LD A, (DE)   | A ← (DE)           | .     | . | X | .   | X       | .   | .  | .   | 00           | 011             | 010             | 1A       | 1   | 2   | 7         |  |
| LD A, (nn)   | A ← (nn)           | .     | . | X | .   | X       | .   | .  | .   | 00           | 111             | 010             | 3A       | 3   | 4   | 13        |  |
|              |                    |       |   |   |     |         |     |    |     | -            | n               | -               |          |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | -            | n               | -               |          |     |     |           |  |
| LD (BC), A   | (BC) ← A           | .     | . | X | .   | X       | .   | .  | .   | 00           | 000             | 010             | 02       | 1   | 2   | 7         |  |
| LD (DE), A   | (DE) ← A           | .     | . | X | .   | X       | .   | .  | .   | 00           | 010             | 010             | 12       | 1   | 2   | 7         |  |
| LD (nn), A   | (nn) ← A           | .     | . | X | .   | X       | .   | .  | .   | 00           | 110             | 010             | 32       | 3   | 4   | 13        |  |
|              |                    |       |   |   |     |         |     |    |     | -            | n               | -               |          |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | -            | n               | -               |          |     |     |           |  |
| LD A, I      | A ← I              | I     | I | X | 0   | X       | IFF | 0  | .   | 11           | 101             | 101             | ED       | 2   | 2   | 9         |  |
| LD A, R      | A ← R              | I     | I | X | 0   | X       | IFF | 0  | .   | 01           | 010             | 111             | 57       |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | 11           | 101             | 101             | ED       | 2   | 2   | 9         |  |
| LD I, A      | I ← A              | .     | . | X | .   | X       | .   | .  | .   | 01           | 011             | 111             | 5F       |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | 11           | 101             | 101             | ED       | 2   | 2   | 9         |  |
| LD R, A      | R ← A              | .     | . | X | .   | X       | .   | .  | .   | 01           | 000             | 111             | 47       |     |     |           |  |
|              |                    |       |   |   |     |         |     |    |     | 11           | 101             | 101             | ED       | 2   | 2   | 9         |  |
|              |                    |       |   |   |     |         |     |    |     | 01           | 001             | 111             | 4F       |     |     |           |  |

Notes: r, s means any of the registers A, B, C, D, E, H, L  
 IFF the content of the interrupt enable flip-flop (IFF) is copied into the P/V flag  
 Flag Notation: . = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,  
 I = flag is affected according to the result of the operation.

Table 7.0-1

| Mnemonic    | Symbolic Operation                                   | Flags |   |   |   |     |   |   | Op-Code |     |     | No. of Bytes | No. of M Cycles | No. of T States | Comments |     |   |                                    |
|-------------|--|-------|---|---|---|-----|---|---|---------|-----|-----|--------------|-----------------|-----------------|----------|-----|---|------------------------------------|
|             |  | S     | Z | X | H | P/V | M | C | 76      | 543 | 210 |              |                 |                 |          | Hex |   |                                    |
| LD dd, nn   | dd ← nn  | •     | • | X | • | X   | • | • | •       | •   | 00  | dd0          | 001             | 3               | 3        | 10  | dd Pair<br>00 BC<br>01 DE<br>10 HL<br>11 SP |                                    |
| LD IX, nn   | IX ← nn  | •     | • | X | • | X   | • | • | •       | •   | 11  | 011          | 101             | DD              | 4        | 4   | 14  |                                    |
| LD IY, nn   | IY ← nn  | •     | • | X | • | X   | • | • | •       | •   | 11  | 111          | 101             | FD              | 4        | 4   | 14  |                                    |
| LD HL, (nn) | H ← (nn+1)<br>L ← (nn)                               | •     | • | X | • | X   | • | • | •       | •   | 00  | 101          | 010             | 2A              | 3        | 5   | 16  |                                    |
| LD dd, (nn) | dd <sub>H</sub> ← (nn+1)<br>dd <sub>L</sub> ← (nn)   | •     | • | X | • | X   | • | • | •       | •   | 11  | 101          | 101             | ED              | 4        | 6   | 20  |                                    |
| LD IX, (nn) | IX <sub>H</sub> ← (nn+1)<br>IX <sub>L</sub> ← (nn)   | •     | • | X | • | X   | • | • | •       | •   | 11  | 011          | 101             | DD              | 4        | 6   | 20  |                                    |
| LD IY, (nn) | IY <sub>H</sub> ← (nn+1)<br>IY <sub>L</sub> ← (nn)   | •     | • | X | • | X   | • | • | •       | •   | 11  | 111          | 101             | FD              | 4        | 6   | 20  |                                    |
| LD (nn), HL | (nn+1) ← H<br>(nn) ← L                               | •     | • | X | • | X   | • | • | •       | •   | 00  | 100          | 010             | 22              | 3        | 5   | 16  |                                    |
| LD (nn), dd | (nn+1) ← dd <sub>H</sub><br>(nn) ← dd <sub>L</sub>   | •     | • | X | • | X   | • | • | •       | •   | 11  | 101          | 101             | ED              | 4        | 6   | 20  |                                    |
| LD (nn), IX | (nn+1) ← IX <sub>H</sub><br>(nn) ← IX <sub>L</sub>   | •     | • | X | • | X   | • | • | •       | •   | 11  | 011          | 101             | DD              | 4        | 6   | 20  |                                    |
| LD (nn), IY | (nn+1) ← IY <sub>H</sub><br>(nn) ← IY <sub>L</sub>   | •     | • | X | • | X   | • | • | •       | •   | 11  | 111          | 101             | FD              | 4        | 6   | 20  |                                    |
| LD SP, HL   | SP ← HL  | •     | • | X | • | X   | • | • | •       | •   | 11  | 111          | 001             | F9              | 1        | 1   | 6   |                                    |
| LD SP, IX   | SP ← IX  | •     | • | X | • | X   | • | • | •       | •   | 11  | 011          | 101             | DD              | 2        | 2   | 10  |                                    |
| LD SP, IY   | SP ← IY  | •     | • | X | • | X   | • | • | •       | •   | 11  | 111          | 001             | F9              | 2        | 2   | 10  |                                    |
| PUSH qq     | (SP-2) ← qq <sub>L</sub><br>(SP-1) ← qq <sub>H</sub> | •     | • | X | • | X   | • | • | •       | •   | 11  | qq0          | 101             | F9              | 1        | 3   | 11  | qq Pair<br>00 BC<br>01 DE<br>11 AF |
| PUSH IX     | (SP-2) ← IX <sub>L</sub><br>(SP-1) ← IX <sub>H</sub> | •     | • | X | • | X   | • | • | •       | •   | 11  | 011          | 101             | DD              | 2        | 4   | 15  |                                    |
| PUSH IY     | (SP-2) ← IY <sub>L</sub><br>(SP-1) ← IY <sub>H</sub> | •     | • | X | • | X   | • | • | •       | •   | 11  | 111          | 101             | FD              | 2        | 4   | 15  |                                    |
| POP qq      | qq <sub>H</sub> ← (SP+1)<br>qq <sub>L</sub> ← (SP)   | •     | • | X | • | X   | • | • | •       | •   | 11  | qq0          | 001             | E5              | 1        | 3   | 10  |                                    |
| POP IX      | IX <sub>H</sub> ← (SP+1)<br>IX <sub>L</sub> ← (SP)   | •     | • | X | • | X   | • | • | •       | •   | 11  | 011          | 101             | DD              | 2        | 4   | 14  |                                    |
| POP IY      | IY <sub>H</sub> ← (SP+1)<br>IY <sub>L</sub> ← (SP)   | •     | • | X | • | X   | • | • | •       | •   | 11  | 111          | 101             | FD              | 2        | 4   | 14  |                                    |

Notes: dd is any of the register pairs BC, DE, HL, SP  
 qq is any of the register pairs AF, BC, DE, HL  
 (PAIR)<sub>H</sub>, (PAIR)<sub>L</sub> refer to high order and low order eight bits of the register pair respectively.  
 e.g. BC<sub>L</sub> = C, AF<sub>H</sub> = A

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,  
 † flag is affected according to the result of the operation.

Table 7.0-2

EXCHANGE GROUP AND BLOCK TRANSFER AND SEARCH GROUP

| Mnemonic    | Symbolic Operation                                 | Flags |   |   |     |   |   |    | Op-Code |            |     | No. of Bytes | No. of Cycles | No. of T States | Comments  |
|-------------|--|-------|---|---|-----|---|---|----|---------|------------|-----|--------------|---------------|-----------------|---|
|             |  | S     | Z | H | P/V | N | C | 76 | 543     | 210        | Hex |              |               |                 |   |
| EX DE, HL   | DE ← HL  | •     | • | X | •   | X | • | •  | •       | 11 101 011 | E8  | 1            | 1             | 4               |   |
| EX AF, AF   | AF ← AF  | •     | • | X | •   | X | • | •  | •       | 00 001 000 | 08  | 1            | 1             | 4               |   |
| EXX         | (BC ← BC)<br>(DE ← DE)<br>(HL ← HL)                | •     | • | X | •   | X | • | •  | •       | 11 011 001 | 09  | 1            | 1             | 4               | Register bank and auxiliary register bank exchange                              |
| EX (SP), HL | H ← (SP+1)<br>L ← (SP)                             | •     | • | X | •   | X | • | •  | •       | 11 100 011 | E3  | 1            | 5             | 19              |   |
| EX (SP), IX | (X <sub>H</sub> ← (SP+1)<br>(X <sub>L</sub> ← (SP) | •     | • | X | •   | X | • | •  | •       | 11 011 101 | 0D  | 2            | 6             | 23              |   |
| EX (SP), IY | (Y <sub>H</sub> ← (SP+1)<br>(Y <sub>L</sub> ← (SP) | •     | • | X | •   | X | • | •  | •       | 11 111 101 | FD  | 2            | 6             | 23              |   |
| LDI         | (DE) ← (HL)  | •     | • | X | 0   | X | ↑ | 0  | •       | 11 101 101 | ED  | 2            | 4             | 16              | Load (HL) into (DE), increment the pointers and decrement the byte counter (BC) |
|             | DE ← DE+1  |       |   |   |     |   |   |    |         | 10 100 000 | A0  |              |               |                 |   |
|             | HL ← HL+1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
| LDIR        | (DE) ← (HL)  | •     | • | X | 0   | X | 0 | 0  | •       | 11 101 101 | ED  | 2            | 5             | 21              | Repeat until BC = 0   |
|             | DE ← DE+1  |       |   |   |     |   |   |    |         | 10 110 000 | 8D  | 2            | 4             | 16              |   |
|             | HL ← HL+1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
| LDO         | (DE) ← (HL)  | •     | • | X | 0   | X | ↑ | 0  | •       | 11 101 101 | ED  | 2            | 4             | 16              | Repeat until BC = 0   |
|             | DE ← DE-1  |       |   |   |     |   |   |    |         | 10 101 000 | A8  |              |               |                 |   |
|             | HL ← HL-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
| LDDR        | (DE) ← (HL)  | •     | • | X | 0   | X | 0 | 0  | •       | 11 101 101 | ED  | 2            | 5             | 21              | Repeat until BC = 0   |
|             | DE ← DE-1  |       |   |   |     |   |   |    |         | 10 111 000 | B8  | 2            | 4             | 16              |   |
|             | HL ← HL-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
| CPI         | A ← (HL)   | ↑     | ↑ | X | ↑   | X | ↑ | 1  | •       | 11 101 101 | ED  | 2            | 4             | 16              | Repeat until A = (HL) or BC = 0   |
|             | HL ← HL+1  |       |   |   |     |   |   |    |         | 10 100 001 | A1  |              |               |                 |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
| CPIR        | A ← (HL)   | ↑     | ↑ | X | ↑   | X | ↑ | 1  | •       | 11 101 101 | ED  | 2            | 5             | 21              | Repeat until A = (HL) or BC = 0   |
|             | HL ← HL+1  |       |   |   |     |   |   |    |         | 10 110 001 | B1  | 2            | 4             | 16              |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
| CPD         | A ← (HL)   | ↑     | ↑ | X | ↑   | X | ↑ | 1  | •       | 11 101 101 | ED  | 2            | 4             | 16              | Repeat until A = (HL) or BC = 0   |
|             | HL ← HL-1  |       |   |   |     |   |   |    |         | 10 101 001 | A9  |              |               |                 |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
| CPDR        | A ← (HL)   | ↑     | ↑ | X | ↑   | X | ↑ | 1  | •       | 11 101 101 | ED  | 2            | 5             | 21              | Repeat until A = (HL) or BC = 0   |
|             | HL ← HL-1  |       |   |   |     |   |   |    |         | 10 111 001 | B9  | 2            | 4             | 16              |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |
|             | BC ← BC-1  |       |   |   |     |   |   |    |         |            |     |              |               |                 |   |

Notes: ① P/V flag is 0 if the result of BC-1 = 0, otherwise P/V = 1  
 ② Z flag is 1 if A = (HL), otherwise Z = 0.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,  
 ↑ = flag is affected according to the result of the operation.

Table 7.D-3

8-BIT ARITHMETIC AND LOGICAL GROUP

| Mnemonic      | Synthetic Operation | Flags |   |   |     |   |   | Op-Code |     |     |     | No. of Bytes | Size of Operands | Format |    |      |  |
|---------------|---------------------|-------|---|---|-----|---|---|---------|-----|-----|-----|--------------|------------------|--------|----|------|--|
|               |                     | S     | Z | H | P/V | N | C | 76      | 543 | 210 | Hex |              |                  |        |    |      |  |
| ADD A, r      | A ← A+r             | †     | † | X | †   | X | V | 0       | †   | 10  | 000 | r            | 1                | 1      | 4  | 1    | Reg.   |
| ADD A, n      | A ← A+n             | †     | † | X | †   | X | V | 0       | †   | 11  | 000 | 110          | 2                | 2      | 7  | 000  | C  |
|               |                     |       |   |   |     |   |   |         |     |     |     | - n          |                  |        |    | 001  | D  |
|               |                     |       |   |   |     |   |   |         |     |     |     |              |                  |        |    | 010  | E  |
| ADD A, (HL)   | A ← A+(HL)          | †     | † | X | †   | X | V | 0       | †   | 10  | 000 | 110          | 1                | 2      | 7  | (11) | F  |
| ADD A, (IX+d) | A ← A+(IX+d)        | †     | † | X | †   | X | V | 0       | †   | 11  | 011 | 101          | DD               | 3      | 5  | 100  | H  |
|               |                     |       |   |   |     |   |   |         |     |     |     |              |                  |        |    | 101  | I  |
|               |                     |       |   |   |     |   |   |         |     |     |     |              |                  |        |    | 111  | J  |
| ADD A, (IY+d) | A ← A+(IY+d)        | †     | † | X | †   | X | V | 0       | †   | 11  | 111 | 101          | FD               | 3      | 5  | 100  | K  |
|               |                     |       |   |   |     |   |   |         |     |     |     |              |                  |        |    | 100  | L  |
|               |                     |       |   |   |     |   |   |         |     |     |     |              |                  |        |    | 101  | M  |
| ADCF A, s     | A ← A+CY            | †     | † | X | †   | X | V | 0       | †   |     | 001 |              |                  |        |    |      | is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instructions. The underlined bits replace the 001 in the ADD instructions. |
| SUB s         | A ← A-s             | †     | † | X | †   | X | V | 1       | †   |     | 010 |              |                  |        |    |      |  |
| SBC A, r      | A ← A-s-CY          | †     | † | X | †   | X | V | 1       | †   |     | 011 |              |                  |        |    |      |  |
| AND s         | A ← A & s           | †     | † | X | †   | X | F | 0       | †   |     | 100 |              |                  |        |    |      |  |
| OR s          | A ← A   s           | †     | † | X | †   | X | F | 0       | †   |     | 110 |              |                  |        |    |      |  |
| XOR s         | A ← A ⊕ s           | †     | † | X | †   | X | F | 0       | †   |     | 101 |              |                  |        |    |      |  |
| CP s          | A ← s               | †     | † | X | †   | X | V | 1       | †   |     | 111 |              |                  |        |    |      |  |
| INC r         | r ← r+1             | †     | † | X | †   | X | V | 0       | 0   | 00  | r   | 100          | 1                | 1      | 4  |      |  |
| INC (HL)      | (HL) ← (HL)+1       | †     | † | X | †   | X | V | 0       | 0   | 00  | 110 | 100          | 1                | 1      | 11 |      |  |
| INC (IX+d)    | (IX+d) ← (IX+d)+1   | †     | † | X | †   | X | V | 0       | 0   | 11  | 011 | 101          | DD               | 3      | 6  | 20   |  |
|               |                     |       |   |   |     |   |   |         |     |     |     |              |                  |        |    |      |  |
|               |                     |       |   |   |     |   |   |         |     |     |     |              |                  |        |    |      |  |
| INC (IY+d)    | (IY+d) ← (IY+d)+1   | †     | † | X | †   | X | V | 0       | 0   | 11  | 111 | 101          | FD               | 3      | 6  | 20   |  |
|               |                     |       |   |   |     |   |   |         |     |     |     |              |                  |        |    |      |  |
|               |                     |       |   |   |     |   |   |         |     |     |     |              |                  |        |    |      |  |
| DEC s         | s ← s-1             | †     | † | X | †   | X | V | 1       | †   |     | 101 |              |                  |        |    |      | is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and status as INC. Replace 100 with 101 in Op-Code.          |

Notes: The V symbol in the P/V flag column indicates that the P/V flag contains the overflow of the result of the operation. Similarly the P symbol indicates parity. V = 1 means overflow, V = 0 means not overflow, P = 1 means parity of the result is even, P = 0 means parity of the result is odd.

Flag Notation: 0 = flag not affected, 1 = flag reset, X = flag is unknown, † = flag is affected according to the result of the operation.

Table 7.0-4

GENERAL PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

| Mnemonic | Symbolic Operation   | Flags |   |   |   |     |   |   |    | Op-Code |     |     |     | No. of Bytes | No. of M Cycles | No. of T States | Comments |   |   |
|----------|--|-------|---|---|---|-----|---|---|----|---------|-----|-----|-----|--------------|-----------------|-----------------|----------|---|---|
|          |  | S     | Z | X | H | P/V | N | C | 76 | 543     | 210 | Hex |     |              |                 |                 |          |   |   |
| DAA      | Converts acc. content into packed BCD following add or subtract with packed BCD operands | ↓     | ↓ | X | ↓ | X   | P | • | ↓  | ↓       | ↓   | 00  | 100 | 111          | 27              | 1               | 1        | 4 | Decimal adjust accumulator                |
| CPL      | $A - \bar{A}$  | •     | • | X | ↓ | X   | • | ↓ | •  | ↓       | •   | 100 | 101 | 111          | 2F              | 1               | 1        | 4 | Complement accumulator                    |
| NEG      | $A - \bar{A} + 1$  | ↓     | ↓ | X | ↓ | X   | V | ↓ | ↓  | ↓       | •   | 11  | 101 | 101          | ED              | 2               | 2        | 8 | Complement accumulator (One's complement) |
| CCF      | $CY - \bar{CY}$  | •     | • | X | X | X   | • | ↓ | •  | ↓       | •   | 01  | 000 | 100          | 44              | 1               | 1        | 4 | Negate acc. (two's complement)            |
| SCF      | $CY - 1$   | •     | • | X | 0 | X   | • | ↓ | •  | ↓       | •   | 100 | 110 | 111          | 37              | 1               | 1        | 4 | Complement carry flag                     |
| NOP      | No operation   | •     | • | X | • | X   | • | • | •  | •       | •   | 00  | 000 | 000          | 00              | 1               | 1        | 4 | Set carry flag                            |
| HALT     | CPU halted   | •     | • | X | • | X   | • | • | •  | •       | •   | 01  | 110 | 110          | 76              | 1               | 1        | 4 |   |
| DI*      | IFF = 0  | •     | • | X | • | X   | • | • | •  | •       | •   | 11  | 110 | 011          | F3              | 1               | 1        | 4 |   |
| EI*      | IFF = 1  | •     | • | X | • | X   | • | • | •  | •       | •   | 11  | 111 | 011          | F8              | 1               | 1        | 4 |   |
| IM 0     | Set interrupt mode 0   | •     | • | X | • | X   | • | • | •  | •       | •   | 11  | 101 | 101          | ED              | 2               | 2        | 8 |   |
| IM 1     | Set interrupt mode 1   | •     | • | X | • | X   | • | • | •  | •       | •   | 01  | 000 | 110          | 46              | 2               | 2        | 8 |   |
| IM 2     | Set interrupt mode 2   | •     | • | X | • | X   | • | • | •  | •       | •   | 01  | 010 | 110          | 56              | 2               | 2        | 8 |   |
|          |  |       |   |   |   |     |   |   |    |         |     | 01  | 011 | 110          | 5E              |                 |          |   |   |

Notes: IFF indicates the interrupt enable flip-flop  
CY indicates the carry flip-flop.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,  
↓ = flag is affected according to the result of the operation.

\*Interrupts are not sampled at the end of EI or DI

Table 7.0-5

16-B. ARITHMETIC GROUP

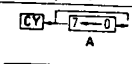
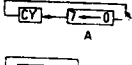
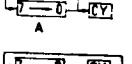
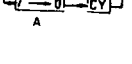
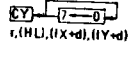
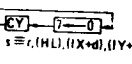
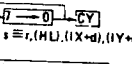
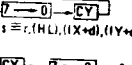
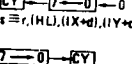
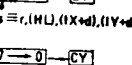
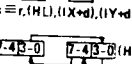
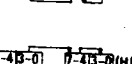
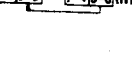
| Mnemonic   | Symbolic Operation | Flags |   |   |    |   |   |   |   | Op-Code                  | Hex | No. of Bytes | No. of Cycles | No. of States | Comments                                       |
|------------|--------------------|-------|---|---|----|---|---|---|---|--------------------------|-----|--------------|---------------|---------------|--|
|            |                    | S     | Z | M | OV | V | M | C |   |                          |     |              |               |               |  |
| ADD HL, ss | HL - HL+ss         | •     | • | X | X  | X | • | 0 | • | 00 ss1 001               |     | 1            | 3             | 11            | ss<br>Reg.                                     |
| ADC HL, ss | HL - HL+ss+CY      | •     | • | X | X  | X | V | 0 | • | 11 101 101<br>01 ss1 010 | ED  | 2            | 4             | 15            | 00 BC<br>01 DE<br>10 HL<br>11 SP               |
| SBC HL, ss | HL - HL-ss-CY      | •     | • | X | X  | X | V | 1 | • | 11 101 101<br>01 ss0 010 | ED  | 2            | 4             | 15            |  |
| ADD IX, pp | IX - IX+pp         | •     | • | X | X  | X | • | 0 | • | 11 011 101<br>00 pp1 001 | DD  | 2            | 4             | 15            | pp<br>Reg.<br>00 BC<br>01 DE<br>10 IX<br>11 SP |
| ADD IY, rr | IY - IY+rr         | •     | • | X | X  | X | • | 0 | • | 11 111 101<br>00 rr1 001 | FD  | 2            | 4             | 15            | rr<br>Reg.<br>00 BC<br>01 DE<br>10 IY<br>11 SP |
| INC ss     | ss - ss + 1        | •     | • | X | •  | X | • | • | • | 00 ss0 011               |     | 1            | 1             | 6             |  |
| INC IX     | IX - IX + 1        | •     | • | X | •  | X | • | • | • | 11 011 101<br>00 100 011 | DD  | 2            | 2             | 10            | 23   |
| INC IY     | IY - IY + 1        | •     | • | X | •  | X | • | • | • | 11 111 101<br>00 100 011 | FD  | 2            | 2             | 10            | 23   |
| DEC ss     | ss - ss - 1        | •     | • | X | •  | X | • | • | • | 00 ss1 011               |     | 1            | 1             | 6             |  |
| DEC IX     | IX - IX - 1        | •     | • | X | •  | X | • | • | • | 11 011 101<br>00 101 011 | DD  | 2            | 2             | 10            | 28   |
| DEC IY     | IY - IY - 1        | •     | • | X | •  | X | • | • | • | 11 111 101<br>00 101 011 | FD  | 2            | 2             | 10            | 28   |

Notes: ss is any of the register pairs BC, DE, HL, SP  
pp is any of the register pairs BC, DE, IX, SP  
rr is any of the register pairs BC, DE, IY, SP.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.  
‡ = flag is affected according to the result of the operation.

Table 7.0-6



| Mnemonic   | Symbolic Operation  | Flags |   |   |   |   |   |    |     | Op-Code |     |       | No. of No. of |        |   | Comments |  |
|------------|---|-------|---|---|---|---|---|----|-----|---------|-----|-------|---------------|--------|---|----------|--|
|            |   | S     | Z | N | V | N | C | 76 | 543 | 210     | Hex | Bytes | Cycles        | States | M |          | T  |
| RLCA       |    | .     | . | X | 0 | X | . | 0  | 1   | 00      | 000 | 111   | 07            | 1      | 1 | 4        | Rotate left circular accumulator                                       |
| RLA        |    | .     | . | X | 0 | X | . | 0  | 1   | 00      | 010 | 111   | 17            | 1      | 1 | 4        | Rotate left accumulator  |
| RRCA       |    | .     | . | X | 0 | X | . | 0  | 1   | 00      | 001 | 111   | 0F            | 1      | 1 | 4        | Rotate right circular accumulator                                      |
| RRA        |    | .     | . | X | 0 | X | . | 0  | 1   | 00      | 011 | 111   | 1F            | 1      | 1 | 4        | Rotate right accumulator   |
| RLC r      |   | .     | . | X | 0 | X | P | 0  | 1   | 11      | 001 | 011   | CB            | 2      | 2 | 8        | Rotate left circular register r  |
| RLC (HL)   |   | .     | . | X | 0 | X | P | 0  | 1   | 11      | 001 | 011   | CB            | 2      | 4 | 15       | Reg.   |
| RLC (IX+d) |    | .     | . | X | 0 | X | P | 0  | 1   | 11      | 011 | 101   | DD            | 4      | 6 | 23       | 000 B  |
| RLC (IY+d) |   | .     | . | X | 0 | X | P | 0  | 1   | 11      | 001 | 011   | CB            |        |   |          | 001 C  |
|            |   | .     | . | X | 0 | X | P | 0  | 1   | 11      | 111 | 101   | FD            | 4      | 6 | 23       | 010 D  |
|            |   | .     | . | X | 0 | X | P | 0  | 1   | 11      | 001 | 011   | CB            |        |   |          | 011 E  |
|            |   | .     | . | X | 0 | X | P | 0  | 1   | 00      | 000 | 110   |               |        |   |          | 100 H  |
|            |   | .     | . | X | 0 | X | P | 0  | 1   | 00      | 000 | 110   |               |        |   |          | 101 L  |
|            |   | .     | . | X | 0 | X | P | 0  | 1   | 11      | 111 | 101   | FD            | 4      | 6 | 23       | 111 A  |
| RLs        |    | .     | . | X | 0 | X | P | 0  | 1   | 00      | 000 | 110   |               |        |   |          |  |
| RRCs       |    | .     | . | X | 0 | X | P | 0  | 1   | 00      | 001 |       |               |        |   |          |  |
| RRs        |    | .     | . | X | 0 | X | P | 0  | 1   | 01      | 1   |       |               |        |   |          |  |
| SLAs       |    | .     | . | X | 0 | X | P | 0  | 1   | 10      | 0   |       |               |        |   |          |  |
| SRAs       |   | .     | . | X | 0 | X | P | 0  | 1   | 10      | 1   |       |               |        |   |          |  |
| SRLs       |  | .     | . | X | 0 | X | P | 0  | 1   | 11      | 1   |       |               |        |   |          |  |
| RLD        |  | .     | . | X | 0 | X | P | 0  | 1   | 11      | 101 | 101   | ED            | 2      | 5 | 18       | Rotate digit left and right between the accumulator and location (HL). |
| RRD        |  | .     | . | X | 0 | X | P | 0  | 1   | 01      | 101 | 111   | 6F            |        |   |          | The content of the upper half of the accumulator is unaffected         |

Flag Notation: . = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↑ = flag is affected according to the result of the operation.

Table 7.0-7

| Mnemonic                   | Symbolic Operation                                      | Flags |   |   |     |   |   |     | Op-Code |            | No. of Bytes | No. of M Cycles | No. of T States | Comments | Reg. |              |
|----------------------------|---|-------|---|---|-----|---|---|-----|---------|------------|--------------|-----------------|-----------------|----------|------|--------------|
|                            |   | S     | Z | H | P/V | N | C | 176 | 543     | 210        |              |                 |                 |          |      | Hex          |
| BIT b, r                   | Z - r <sub>b</sub>                                      | X     | 1 | X | 1   | X | X | 0   | •       | 11 001 011 | CB           | 2               | 2               | 8        |      |              |
|                            |   |       |   |   |     |   |   |     | •       | 01 b r     |              |                 |                 |          |      | 000 B        |
| BIT b, (HL)                | Z - (HL) <sub>b</sub>                                   | X     | 1 | X | 1   | X | X | 0   | •       | 11 001 011 | CB           | 2               | 3               | 12       |      | 001 C        |
|                            |   |       |   |   |     |   |   |     | •       | 01 b 110   |              |                 |                 |          |      | 010 D        |
| BIT b, (IX+d) <sub>b</sub> | Z - (IX+d) <sub>b</sub>                                 | X     | 1 | X | 1   | X | X | 0   | •       | 11 011 101 | DD           | 4               | 5               | 20       |      | 011 E        |
|                            |   |       |   |   |     |   |   |     | •       | 11 001 011 | CB           |                 |                 |          |      | 100 H        |
|                            |   |       |   |   |     |   |   |     | •       | - d -      |              |                 |                 |          |      | 101 L        |
|                            |   |       |   |   |     |   |   |     | •       | 01 b 110   |              |                 |                 |          |      | 111 A        |
| BIT b, (IY+d) <sub>b</sub> | Z - (IY+d) <sub>b</sub>                                 | X     | 1 | X | 1   | X | X | 0   | •       | 11 111 101 | FD           | 4               | 5               | 20       |      | b Bit Tested |
|                            |   |       |   |   |     |   |   |     | •       | 11 001 011 | CB           |                 |                 |          |      | 000 0        |
|                            |   |       |   |   |     |   |   |     | •       | - d -      |              |                 |                 |          |      | 001 1        |
|                            |   |       |   |   |     |   |   |     | •       | 01 b 110   |              |                 |                 |          |      | 010 2        |
|                            |   |       |   |   |     |   |   |     | •       |            |              |                 |                 |          |      | 011 3        |
|                            |   |       |   |   |     |   |   |     | •       |            |              |                 |                 |          |      | 100 4        |
|                            |   |       |   |   |     |   |   |     | •       |            |              |                 |                 |          |      | 101 5        |
|                            |   |       |   |   |     |   |   |     | •       |            |              |                 |                 |          |      | 110 6        |
|                            |   |       |   |   |     |   |   |     | •       |            |              |                 |                 |          |      | 111 7        |
| SET b, r                   | r <sub>b</sub> - 1                                      | •     | • | X | •   | X | • | •   | •       | 11 001 011 | CB           | 2               | 2               | 8        |      |              |
|                            |   |       |   |   |     |   |   |     | •       | 11 b r     |              |                 |                 |          |      |              |
| SET b, (HL)                | (HL) <sub>b</sub> - 1                                   | •     | • | X | •   | X | • | •   | •       | 11 001 011 | CB           | 2               | 4               | 15       |      |              |
|                            |   |       |   |   |     |   |   |     | •       | 11 b 110   |              |                 |                 |          |      |              |
| SET b, (IX+d)              | (IX+d) <sub>b</sub> - 1                                 | •     | • | X | •   | X | • | •   | •       | 11 011 101 | DD           | 4               | 6               | 23       |      |              |
|                            |   |       |   |   |     |   |   |     | •       | 11 001 011 | CB           |                 |                 |          |      |              |
|                            |   |       |   |   |     |   |   |     | •       | - d -      |              |                 |                 |          |      |              |
|                            |   |       |   |   |     |   |   |     | •       | 11 b 110   |              |                 |                 |          |      |              |
| SET b, (IY+d)              | (IY+d) <sub>b</sub> - 1                                 | •     | • | X | •   | X | • | •   | •       | 11 111 101 | FD           | 4               | 6               | 23       |      |              |
|                            |   |       |   |   |     |   |   |     | •       | 11 001 011 | CB           |                 |                 |          |      |              |
|                            |   |       |   |   |     |   |   |     | •       | - d -      |              |                 |                 |          |      |              |
|                            |   |       |   |   |     |   |   |     | •       | 11 b 110   |              |                 |                 |          |      |              |
| RES b, s                   | s <sub>b</sub> - 0<br>s ≡ r, (HL),<br>(IX+d),<br>(IY+d) | •     | • | X | •   | X | • | •   | •       | 10         |              |                 |                 |          |      |              |

To form new Op-Code replace 11 of SET b, s with 10. Flags and time states for SET instruction

Notes: The notation s<sub>b</sub> indicates bit b (0 to 7) or location s.  
 Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, 10 = flag is affected according to the result of the operation.

Table 7.0-8

JUMP GROUP

| Mnemonic    | Symbolic Operation                                  | Flags |   |   |   |     |   |   |    | Op-Code |     |            | No. of Bytes | No. of Cycles | No. of States | Comments |   |                      |
|-------------|---|-------|---|---|---|-----|---|---|----|---------|-----|------------|--------------|---------------|---------------|----------|---|----------------------|
|             |   | S     | Z | H | X | P/V | N | C | 76 | 543     | 210 | Hex        |              |               |               |          |   |                      |
| JP nn       | PC - nn   | .     | . | . | . | .   | . | . | .  | .       | .   | 11 000 011 | C3           | 3             | 3             | 10       |   |                      |
| → JP cc, nn | If condition cc is true PC - nn, otherwise continue | .     | . | X | . | X   | . | . | .  | .       | .   | 11 cc 010  |              | 3             | 3             | 10       | cc Condition<br>000 NZ non zero<br>001 Z zero<br>010 NC non carry<br>011 C carry<br>100 PO parity odd<br>101 PE parity even<br>110 P sign positive<br>111 M sign negative |                      |
| JR e        | PC - PC + e   | .     | . | X | . | X   | . | . | .  | .       | .   | 00 011 000 |              | 18            | 2             | 3        | 12  |                      |
| JR C, e     | If C = 0, continue<br>If C = 1, PC - PC + e         | .     | . | X | . | X   | . | . | .  | .       | .   | 00 111 000 |              | 38            | 2             | 2        | 7   | If condition not met |
| JR NC, e    | If C = 1, continue<br>If C = 0, PC - PC + e         | .     | . | X | . | X   | . | . | .  | .       | .   | 00 110 000 |              |               | 2             | 3        | 12  | If condition is met  |
| → JR Z, e   | If Z = 0, continue<br>If Z = 1, PC - PC + e         | .     | . | X | . | X   | . | . | .  | .       | .   | 00 101 000 |              | 28            | 2             | 2        | 7   | If condition not met |
| JR NZ, e    | If Z = 1, continue<br>If Z = 0, PC - PC + e         | .     | . | X | . | X   | . | . | .  | .       | .   | 00 100 000 |              |               | 2             | 3        | 12  | If condition is met  |
| JP (HL)     | PC - HL   | .     | . | X | . | X   | . | . | .  | .       | .   | 11 101 001 | E9           | 1             | 1             | 4        |   |                      |
| JP (IX)     | PC - IX   | .     | . | X | . | X   | . | . | .  | .       | .   | 11 011 101 | DD           | 2             | 2             | 8        |   |                      |
| JP (IY)     | PC - IY   | .     | . | X | . | X   | . | . | .  | .       | .   | 11 111 101 | FD           | 2             | 2             | 8        |   |                      |
| DJNZ, e     | B - B - 1<br>If B = 0, continue                     | .     | . | X | . | X   | . | . | .  | .       | .   | 00 010 000 |              | 10            | 2             | 2        | 8   | If B = 0             |
|             | If B ≠ 0, PC - PC + e                               | .     | . | . | . | .   | . | . | .  | .       | .   | - e-2 -    |              |               | 2             | 3        | 13  | If B ≠ 0             |

Notes: e represents the extension in the relative addressing mode.  
 e is a signed two's complement number in the range <126, 129>  
 e-2 in the op-code provides an effective address of pc+e as PC is incremented by 2 prior to the addition of e.

Flag Notation: . = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ! = flag is affected according to the result of the operation.

Table 7.0-9

CALL AND RETURN GROUP

| Mnemonic          | Symbolic Operation   | Flags |   |   |   |     |   |   |    | Op-Code |     |     |     | No. of Bytes | No. of Cycles | No. of States | No. of T <sup>1</sup> | Comments |   |
|-------------------|--|-------|---|---|---|-----|---|---|----|---------|-----|-----|-----|--------------|---------------|---------------|-----------------------|----------|---|
|                   |  | S     | Z | X | H | P/V | N | C | 78 | 543     | 210 | Hex |     |              |               |               |                       |          |   |
| CALL nn           | (SP-1) - PC <sub>H</sub><br>(SP-2) - PC <sub>L</sub><br>PC - nn                                    | .     | . | X | . | X   | . | . | .  | .       | .   | 11  | 001 | 101          | CD            | 3             | 5                     | 17       |   |
| CALL cc, nn       | If condition cc is false, continue, otherwise same as CALL nn                                      | .     | . | X | . | X   | . | . | .  | .       | .   | 11  | cc  | 100          |               | 3             | 3                     | 10       | If cc is false  |
|                   |  |       |   |   |   |     |   |   |    |         |     | -   | n   | -            |               | 3             | 5                     | 17       | If cc is true   |
| RET               | PC <sub>L</sub> - (SP)<br>PC <sub>H</sub> - (SP+1)   | .     | . | X | . | X   | . | . | .  | .       | .   | 11  | 001 | 001          | CS            | 1             | 3                     | 10       |   |
| RET cc            | If condition cc is false, continue, otherwise same as RET  | .     | . | X | . | X   | . | . | .  | .       | .   | 11  | cc  | 000          |               | 1             | 1                     | 5        | If cc is false  |
|                   |  |       |   |   |   |     |   |   |    |         |     |     |     |              |               | 1             | 3                     | 11       | If cc is true   |
| RETI              | Return from interrupt  | .     | . | X | . | X   | . | . | .  | .       | .   | 11  | 101 | 101          | ED            | 2             | 4                     | 14       | 000 NZ non zero<br>001 Z zero<br>010 NC non carry<br>011 C carry                      |
| RETN <sup>1</sup> | Return from non maskable interrupt   | .     | . | X | . | X   | . | . | .  | .       | .   | 11  | 101 | 101          | ED            | 2             | 4                     | 14       | 100 PO parity odd<br>101 PE parity even<br>110 P sign positive<br>111 M sign negative |
|                   |  |       |   |   |   |     |   |   |    |         |     |     |     |              |               |               |                       |          |   |
| RST p             | (SP-1) - PC <sub>H</sub><br>(SP-2) - PC <sub>L</sub><br>PC <sub>H</sub> - 0<br>PC <sub>L</sub> - p | .     | . | X | . | X   | . | . | .  | .       | .   | 11  | r   | 111          |               | 1             | 3                     | 11       |   |

<sup>1</sup>RETN loads IFF<sub>2</sub> - IFF<sub>1</sub>

Flag Notation: . = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, † = flag is affected according to the result of the operation.

Table 7.0-10

INPUT AND OUTPUT GROUP

| Mnemonic   | Symbolic Operation   | Flags |   |   |     |   |   |    | Op-Code |     |     |     | No. of Bytes | No. of M Cycles | No. of T States | Comments |                                    |   |   |
|------------|--|-------|---|---|-----|---|---|----|---------|-----|-----|-----|--------------|-----------------|-----------------|----------|------------------------------------|---|---|
|            |  | S     | Z | H | P/V | N | C | 76 | 543     | 210 | Hex |     |              |                 |                 |          |                                    |   |   |
| INA (n)    | A ← (n)  | .     | . | X | .   | X | . | .  | .       | .   | 11  | 011 | 011          | DB              | 2               | 3        | 11                                 | n to A <sub>0</sub> ~ A <sub>7</sub><br>Acc to A <sub>8</sub> ~ A <sub>15</sub> |   |
| IN r, (C)  | r ← (C)<br>if r = 110 only the flags will be affected        | .     | ↓ | X | .   | X | . | P  | .       | 0   | .   | 11  | 101          | 101             | ED              | 2        | 3                                  | 12  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |
| INI        | (HL) ← (C)<br>B ← B - 1<br>HL ← HL + 1                       | X     | . | X | .   | X | X | X  | .       | 1   | .   | 11  | 101          | 101             | ED              | 2        | 4                                  | 16  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |
| INIR       | (HL) ← (C)<br>B ← B - 1<br>HL ← HL + 1<br>Repeat until B = 0 | X     | 1 | X | .   | X | X | X  | .       | 1   | .   | 11  | 101          | 101             | ED              | 2        | 5<br>(if B ≠ 0)<br>4<br>(if B = 0) | 21<br>16  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |
| IND        | (HL) ← (C)<br>B ← B - 1<br>HL ← HL - 1                       | X     | . | X | .   | X | X | X  | .       | 1   | .   | 11  | 101          | 101             | ED              | 2        | 4                                  | 16  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |
| INDR       | (HL) ← (C)<br>B ← B - 1<br>HL ← HL - 1<br>Repeat until B = 0 | X     | 1 | X | .   | X | X | X  | .       | 1   | .   | 11  | 101          | 101             | ED              | 2        | 5<br>(if B ≠ 0)<br>4<br>(if B = 0) | 21<br>16  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |
| OUT (n), A | (n) ← A  | .     | . | X | .   | X | . | .  | .       | .   | .   | 11  | 010          | 011             | D3              | 2        | 3                                  | 11  | n to A <sub>0</sub> ~ A <sub>7</sub><br>Acc to A <sub>8</sub> ~ A <sub>15</sub> |
| OUT (C), r | (C) ← r  | .     | . | X | .   | X | . | .  | .       | .   | .   | 11  | 101          | 101             | ED              | 2        | 3                                  | 12  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |
| OUTI       | (C) ← (HL)<br>B ← B - 1<br>HL ← HL + 1                       | X     | 1 | X | .   | X | X | X  | .       | 1   | .   | 11  | 101          | 101             | ED              | 2        | 4                                  | 16  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |
| OTIR       | (C) ← (HL)<br>B ← B - 1<br>HL ← HL + 1<br>Repeat until B = 0 | X     | 1 | X | .   | X | X | X  | .       | 1   | .   | 11  | 101          | 101             | ED              | 2        | 5<br>(if B ≠ 0)<br>4<br>(if B = 0) | 21<br>16  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |
| OUTD       | (C) ← (HL)<br>B ← B - 1<br>HL ← HL - 1                       | X     | . | X | .   | X | X | X  | .       | 1   | .   | 11  | 101          | 101             | ED              | 2        | 4                                  | 16  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |
| OTDR       | (C) ← (HL)<br>B ← B - 1<br>HL ← HL - 1<br>Repeat until B = 0 | X     | 1 | X | .   | X | X | X  | .       | 1   | .   | 11  | 101          | 101             | ED              | 2        | 5<br>(if B ≠ 0)<br>4<br>(if B = 0) | 21<br>16  | C to A <sub>0</sub> ~ A <sub>7</sub><br>B to A <sub>8</sub> ~ A <sub>15</sub>   |

Notes: ① If the result of B - 1 is zero the Z flag is set, otherwise it is reset.

Flag Notation: . = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↓ = flag is affected according to the result of the operation.

Table 7.0-11

The purpose of an interrupt is to allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start a peripheral service routine. Usually this service routine is involved with the exchange of data, or status and control information, between the CPU and the peripheral. Once the service routine is completed, the CPU returns to the operation from which it was interrupted.

#### INTERRUPT ENABLE - DISABLE

The Z80-CPU has two interrupt inputs, a software maskable interrupt and a non-maskable interrupt. The non-maskable interrupt (NMI) can not be disabled by the programmer and it will be accepted whenever a peripheral device requests it. This interrupt is generally reserved for very important functions that must be serviced whenever they occur, such as an impending power failure. The maskable interrupt (INT) can be selectively enabled or disabled by the programmer. This allows the programmer to disable the interrupt during periods where his program has timing constraints that do not allow it to be interrupted. In the Z80-CPU there is an enable flip flop (called IFF) that is set or reset by the programmer using the Enable Interrupt (EI) and Disable Interrupt (DI) instructions. When the IFF is reset, an interrupt can not be accepted by the CPU.

Actually, for purposes that will be subsequently explained, there are two enable flip flops, called IFF<sub>1</sub> and IFF<sub>2</sub>.

IFF<sub>1</sub>

Actually disables interrupts  
from being accepted.

IFF<sub>2</sub>

Temporary storage location  
for IFF<sub>1</sub>.

The state of IFF<sub>1</sub> is used to actually inhibit interrupts while IFF<sub>2</sub> is used as a temporary storage location for IFF<sub>1</sub>. The purpose of storing the IFF<sub>1</sub> will be subsequently explained.

A reset to the CPU will force both IFF<sub>1</sub> and IFF<sub>2</sub> to the reset state so that interrupts are disabled. They can then be enabled by an EI instruction at any time by the programmer. When an EI instruction is executed, any pending interrupt request will not be accepted until after the instruction following EI has been executed. This single instruction delay is necessary for cases when the following instruction is a return instruction and interrupts must not be allowed until the return has been completed. The EI instructions sets both IFF<sub>1</sub> and IFF<sub>2</sub> to the enable state. When an interrupt is accepted by the CPU, both IFF<sub>1</sub> and IFF<sub>2</sub> are automatically reset, inhibiting further interrupts until the programmer wishes to issue a new EI instruction. Note that for all of the previous cases, IFF<sub>1</sub> and IFF<sub>2</sub> are always equal.

The purpose of IFF<sub>2</sub> is to save the status of IFF<sub>1</sub> when a non-maskable interrupt occurs. When a non-maskable interrupt is accepted, IFF<sub>1</sub> is reset to prevent further interrupts until reenabled by the programmer. Thus, after a non-maskable interrupt has been accepted maskable interrupts are disabled but the previous state of IFF<sub>1</sub> has been saved so that the complete state of the CPU just prior to the non-maskable interrupt can be restored at any time. When a Load Register A with Register I (LD A, I) instruction or a Load Register A with Register R (LD A, R) instruction is executed, the state of IFF<sub>2</sub> is copied into the parity flag where it can be tested or stored.

A second method of restoring the status of IFF<sub>1</sub> is thru the execution of a Return From Non-Maskable Interrupt (RETN) instruction. Since this instruction indicates that the non maskable interrupt service routine is complete, the contents of IFF<sub>2</sub> are now copied back into IFF<sub>1</sub>, so that the status of IFF<sub>1</sub> just prior to the acceptance of the non-maskable interrupt will be restored automatically.

**INTERRUPT ENABLE/DISABLE FLIP FLOPS**

| Action     | IFF <sub>1</sub> | IFF <sub>2</sub> |                                     |
|------------|------------------|------------------|-------------------------------------|
| CPU Reset  | 0                | 0                |                                     |
| DI         | 0                | 0                |                                     |
| EI         | 1                | 1                |                                     |
| LD A, I    | •                | •                | IFF <sub>2</sub> → Parity flag      |
| LD A, R    | •                | •                | IFF <sub>2</sub> → Parity flag      |
| Accept NMI | 0                | •                |                                     |
| RETN       | IFF <sub>2</sub> | •                | IFF <sub>2</sub> → IFF <sub>1</sub> |
| Accept INT | 0                | 0                |                                     |
| RETI       | •                | •                |                                     |

"•" indicates no change

FIGURE 8.0-1

**CPU RESPONSE**

**Non-Maskable**

A non-maskable interrupt will be accepted at all times by the CPU. When this occurs, the CPU ignores the next instruction that it fetches and instead does a restart to location 0066H. Thus, it behaves exactly as if it had received a restart instruction but, it is to a location that is not one of the 8 software restart locations. A restart is merely a call to a specific address in page 0 memory.

**Maskable**

The CPU can be programmed to respond to the maskable interrupt in any one of three possible modes.

**Mode 0**

This mode is identical to the 8080A interrupt response mode. With this mode, the interrupting device can place any instruction on the data bus and the CPU will execute it. Thus, the interrupting device provides the next instruction to be executed instead of the memory. Often this will be a restart instruction since the interrupting device only need supply a single byte instruction. Alternatively, any other instruction such as a 3 byte call to any location in memory could be executed.

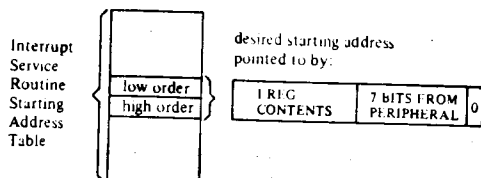
The number of clock cycles necessary to execute this instruction is 2 more than the normal number for the instruction. This occurs since the CPU automatically adds 2 wait states to an interrupt response cycle to allow sufficient time to implement an external daisy chain for priority control. Section 4.0 illustrates the detailed timing for an interrupt response. After the application of RESET the CPU will automatically enter interrupt Mode 0.

**Mode 1**

When this mode has been selected by the programmer, the CPU will respond to an interrupt by executing a restart to location 0038H. Thus the response is identical to that for a non maskable interrupt except that the call location is 0038H instead of 0066H. Another difference is that the number of cycles required to complete the restart instruction is 2 more than normal due to the two added wait states.

This mode is the most powerful interrupt response mode. With a single 8-bit byte from the user an indirect call can be made to any memory location.

With this mode the programmer maintains a table of 16 bit starting addresses for every interrupt service routine. This table may be located anywhere in memory. When an interrupt is accepted, a 16 bit pointer must be formed to obtain the desired interrupt service routine starting address from the table. The upper 8 bits of this pointer is formed from the contents of the I register. The I register must have been previously loaded with the desired value by the programmer, i.e. LD I, A. Note that a CPU reset clears the I register so that it is initialized to zero. The lower eight bits of the pointer must be supplied by the interrupting device. Actually, only 7 bits are required from the interrupting device as the least bit must be a zero. This is required since the pointer is used to get two adjacent bytes to form a complete 16 bit service routine starting address and the addresses must always start in even locations.



The first byte in the table is the least significant (low order) portion of the address. The programmer must obviously fill this table in with the desired addresses before any interrupts are to be accepted.

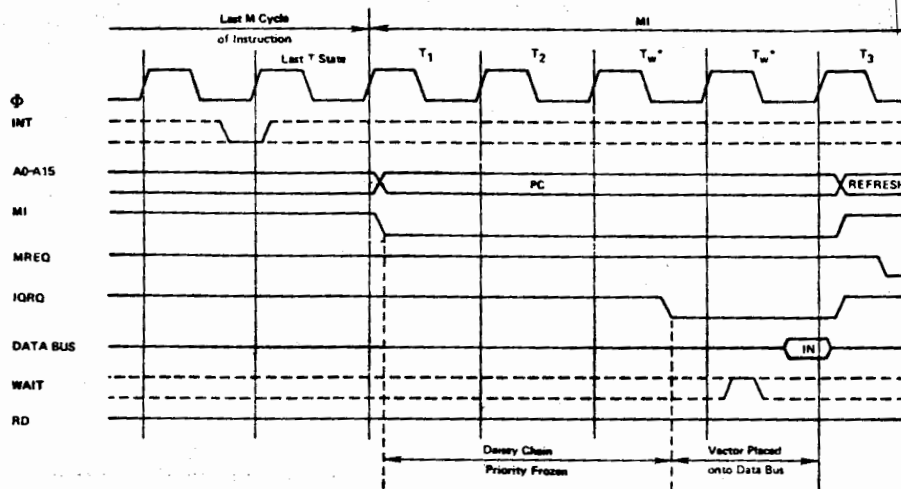
Note that this table can be changed at any time by the programmer (if it is stored in Read/Write Memory) to allow different peripherals to be serviced by different service routines.

Once the interrupting device supplies the lower portion of the pointer, the CPU automatically pushes the program counter onto the stack, obtains the starting address from the table and does a jump to this address. This mode of response requires 19 clock periods to complete (7 to fetch the lower 8 bits from the interrupting device, 6 to save the program counter, and 6 to obtain the jump address.)

Note that the Z80 peripheral devices all include a daisy chain priority interrupt structure that automatically supplies the programmed vector to the CPU during interrupt acknowledge. Refer to the Z80-PIO, Z80-SIO and Z80-CTC manuals for details.



## INTERRUPT REQUEST/ACKNOWLEDGE CYCLE



## Z80 INTERRUPT ACKNOWLEDGE SUMMARY

- 1) **PERIPHERAL DEVICE REQUESTS INTERRUPT.** Any device requesting and interrupt can pull the wired-or line  $\overline{INT}$  low.
- 2) **CPU ACKNOWLEDGES INTERRUPT.** Priority status is frozen when  $\overline{MI}$  goes low during the Interrupt Acknowledge sequence. Propagation delays down the IEI/IEO daisy chain must be settled out when  $\overline{IORQ}$  goes low. If IEI is HIGH, an active Peripheral Device will place its Interrupt Vector on the Data Bus when  $\overline{IORQ}$  goes low. That Peripheral then releases its hold on  $\overline{INT}$  allowing interrupts from a higher priority device. Lower priority devices are inhibited from placing their Vector on the Data Bus or Interrupting because IEO is low on the active device.
- 3) **INTERRUPT IS CLEARED.** An active Peripheral device (IEI=1, IEO=0) monitors OP Code fetches for an RETI (ED 4D) instruction which tells the peripheral that its Interrupt Service Routine is over. The peripheral device then re-activates its internal Interrupt structure as well as raising its IEO line to enable lower priority devices.

### 11.0 ELECTRICAL SPECIFICATIONS ABSOLUTE MAXIMUM RATINGS\*

|   |                           |
|---|---------------------------|
| Temperature Under Bias .....                    | Specified Operating Range |
| Storage Temperature .....                       | -65°C to +150°C           |
| Voltage on Any Pin with Respect to Ground ..... | -0.3V to +7V              |
| Power Dissipation .....                         | 1.5W                      |

### D.C. CHARACTERISTICS

$T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5\text{V} \pm 5\%$  unless otherwise specified

| SYMBOL    | PARAMETER                                 | MIN.         | TYP. | MAX.         | UNIT          | TEST CONDITION               |
|-----------|---|--------------|------|--------------|---------------|------------------------------|
| $V_{ILC}$ | Clock Input Low Voltage                   | -0.3         |      | 0.8          | V             |                              |
| $V_{IHC}$ | Clock Input High Voltage                  | $V_{CC}-0.6$ |      | $V_{CC}+0.3$ | V             |                              |
| $V_{IL}$  | Input Low Voltage                         | -0.3         |      | 0.8          | V             |                              |
| $V_{IH}$  | Input High Voltage                        | 2.0          |      | $V_{CC}$     | V             |                              |
| $V_{OL}$  | Output Low Voltage                        |              |      | 0.4          | V             | $I_{OL} = 1.8\text{mA}$      |
| $V_{OH}$  | Output High Voltage                       | 2.4          |      |              | V             | $I_{OH} = -250\ \mu\text{A}$ |
| $I_{CC}$  | Power Supply Current                      |              |      | 150*         | mA            |                              |
| $I_{LI}$  | Input Leakage Current                     |              |      | 10           | $\mu\text{A}$ | $V_{IN} = 0$ to $V_{CC}$     |
| $I_{LOH}$ | Tri-State Output Leakage Current in Float |              |      | 10           | $\mu\text{A}$ | $V_{OUT} = 2.4$ to $V_{CC}$  |
| $I_{LOL}$ | Tri-State Output Leakage Current in Float |              |      | -10          | $\mu\text{A}$ | $V_{OUT} = 0.4\text{V}$      |
| $I_{LD}$  | Data Bus Leakage Current in Input Mode    |              |      | $\pm 10$     | $\mu\text{A}$ | $0 \leq V_{IN} \leq V_{CC}$  |

\*200mA for -4, -10 or -20 devices

### CAPACITANCE

$T_A = 25^\circ\text{C}$ ,  $f = 1\text{MHz}$  unmeasured pins returned to ground

| SYMBOL    | PARAMETER          | MAX. | UNIT |
|-----------|--------------------|------|------|
| $C_\phi$  | Clock Capacitance  | 35   | pF   |
| $C_{IN}$  | Input Capacitance  | 5    | pF   |
| $C_{OUT}$ | Output Capacitance | 10   | pF   |

#### \*Comment

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

MK 3880, MK 3880-10, MK 3880-20 Z80-CPU

A C CHARACTERISTICS

T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = +5V ± 5%, Unless Otherwise Noted

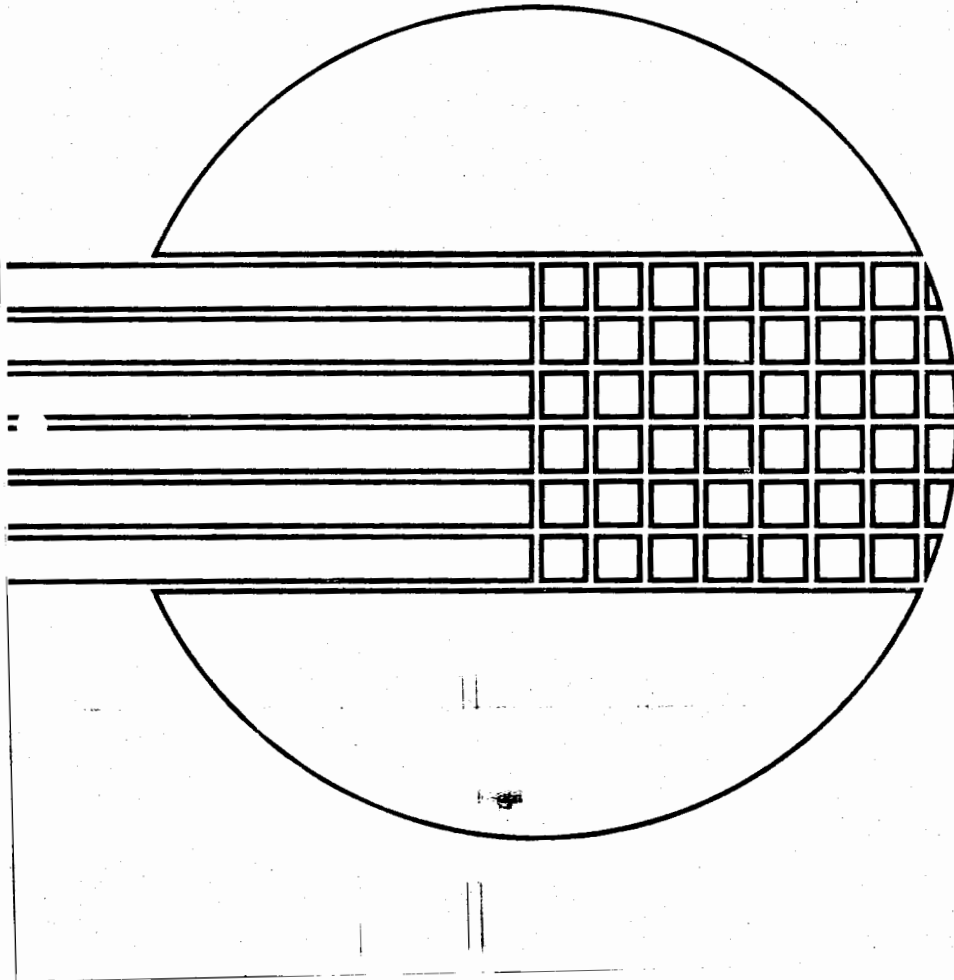
| SIGNAL            | SYMBOL                | PARAMETER  | MIN. | MAX. | UNIT | TEST CONDITION        |
|-------------------|-----------------------|--|------|------|------|-----------------------|
| Φ                 | t <sub>c</sub>        | Clock Period   | 4    | [12] | μsec |                       |
|                   | t <sub>w</sub> (ΦH)   | Clock Pulse Width, Clock High  | 180  | (D)  | nsec |                       |
|                   | t <sub>w</sub> (ΦL)   | Clock Pulse Width, Clock Low   | 180  | 2000 | nsec |                       |
|                   | t <sub>r,f</sub>      | Clock Rise and Fall Time   |      | 30   | nsec |                       |
| A <sub>0-15</sub> | t <sub>D</sub> (AD)   | Address Output Delay   |      | 145  | nsec | C <sub>L</sub> = 50pF |
|                   | t <sub>F</sub> (AD)   | Delay to Float   | [1]  | 110  | nsec |                       |
|                   | t <sub>acm</sub>      | Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)                                       | [2]  |      | nsec |                       |
|                   | t <sub>aci</sub>      | Address Stable Prior to $\overline{IORQ}$ , $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)     | [3]  |      | nsec |                       |
| D <sub>0-7</sub>  | t <sub>ca</sub>       | Address Stable From $\overline{RD}$ , $\overline{WR}$ , $\overline{IORQ}$ or $\overline{MREQ}$ | [4]  |      | nsec | C <sub>L</sub> = 50pF |
|                   | t <sub>caf</sub>      | Address Stable From $\overline{RD}$ or $\overline{WR}$ During Float                            |      |      | nsec |                       |
|                   | t <sub>D</sub> (D)    | Data Output Delay  |      | 230  | nsec |                       |
|                   | t <sub>F</sub> (D)    | Delay to Float During Write Cycle  | 50   | 90   | nsec |                       |
|                   | t <sub>SΦ</sub> (D)   | Data Setup Time to Rising Edge of Clock During M1 Cycle  | 60   |      | nsec |                       |
|                   | t <sub>SΦ</sub> (D)   | Data Setup Time to Falling Edge at Clock During M2 to M5                                       | [5]  |      | nsec |                       |
| MREQ              | t <sub>dcn</sub>      | Data Stable Prior to $\overline{WR}$ (Memory Cycle)  | [6]  |      | nsec | C <sub>L</sub> = 50pF |
|                   | t <sub>dci</sub>      | Data Stable Prior to $\overline{WR}$ (I/O Cycle)   | [7]  |      | nsec |                       |
|                   | t <sub>cdf</sub>      | Data Stable From $\overline{WR}$   | [8]  |      | nsec |                       |
|                   | t <sub>H</sub>        | Input Hold Time  | 0    |      | nsec |                       |
|                   | t <sub>DLΦ</sub> (MR) | $\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ Low                      |      | 100  | nsec |                       |
| IORQ              | t <sub>DHΦ</sub> (MR) | $\overline{MREQ}$ Delay From Rising Edge of Clock, $\overline{MREQ}$ High                      |      | 100  | nsec | C <sub>L</sub> = 50pF |
|                   | t <sub>DHΦ</sub> (MR) | $\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ High                     |      | 100  | nsec |                       |
|                   | t <sub>w</sub> (MRL)  | Pulse Width, $\overline{MREQ}$ Low   | [8]  |      | nsec |                       |
|                   | t <sub>w</sub> (MRR)  | Pulse Width, $\overline{MREQ}$ High  | [9]  |      | nsec |                       |
| IORQ              | t <sub>DLΦ</sub> (IR) | $\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ Low                       |      | 90   | nsec | C <sub>L</sub> = 50pF |
|                   | t <sub>DLΦ</sub> (IR) | $\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ Low                      |      | 110  | nsec |                       |
|                   | t <sub>DHΦ</sub> (IR) | $\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ High                      |      | 100  | nsec |                       |
|                   | t <sub>DHΦ</sub> (IR) | $\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ High                     |      | 110  | nsec |                       |
| RD                | t <sub>DLΦ</sub> (RD) | $\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ Low                           |      | 100  | nsec | C <sub>L</sub> = 50pF |
|                   | t <sub>DLΦ</sub> (RD) | $\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ Low                          |      | 130  | nsec |                       |
|                   | t <sub>DHΦ</sub> (RD) | $\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ High                          |      | 100  | nsec |                       |
|                   | t <sub>DHΦ</sub> (RD) | $\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ High                         |      | 110  | nsec |                       |
| WR                | t <sub>DLΦ</sub> (WR) | $\overline{WR}$ Delay From Rising Edge of Clock, $\overline{WR}$ Low                           |      | 80   | nsec | C <sub>L</sub> = 50pF |
|                   | t <sub>DLΦ</sub> (WR) | $\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ Low                          |      | 90   | nsec |                       |
|                   | t <sub>DHΦ</sub> (WR) | $\overline{WR}$ Delay From Rising Edge of Clock, $\overline{WR}$ High                          |      | 100  | nsec |                       |
|                   | t <sub>w</sub> (WRL)  | Pulse Width, $\overline{WR}$ Low   | [10] |      | nsec |                       |

NOTES:

A Data should be enabled onto the CPU data bus when RD is active. During interrupt acknowledge data should be enabled when  $\overline{M1}$  and  $\overline{IORQ}$  are both active.

B The  $\overline{RESET}$  signal must be active for a minimum of 3 clock cycles.  
cont'd on page 79

**SIGNETICS  
16384-BIT  
ERASABLE AND  
REPROGRAMMABLE  
MOS ROM (2048X8)  
2716**



**16,384-BIT ERASABLE AND REPROGRAMMABLE MOS ROM (2048X8)**

PRELIMINARY SPECIFICATION

2716-I

**DESCRIPTION**

The Signetics 2716 is a 16,384 bit erasable programmable read only memory (EPROM). The 2716 is organized as 2048 words of 8 bits each and features fast single address location programming. Erasure is accomplished by exposure to ultraviolet light and programming is performed electrically. Once a program is finalized the 2716 can convert to Signetics pin-for-pin compatible 16K ROM, the 2616.

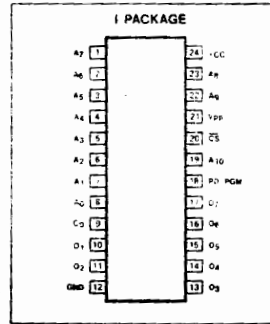
The 2716 operates from a single 5 volt power supply which makes it ideal for use with the newer high performance 5 volt microprocessors. A power down mode reduces power during standby to 25% that of operating power.

Single pulse TTL level programming makes the 2716 simple and fast to program. All control signals are TTL level allowing on board programming. Words can be selected individually, sequentially or randomly. Total programming time for all 16,384 bits is 100 seconds.

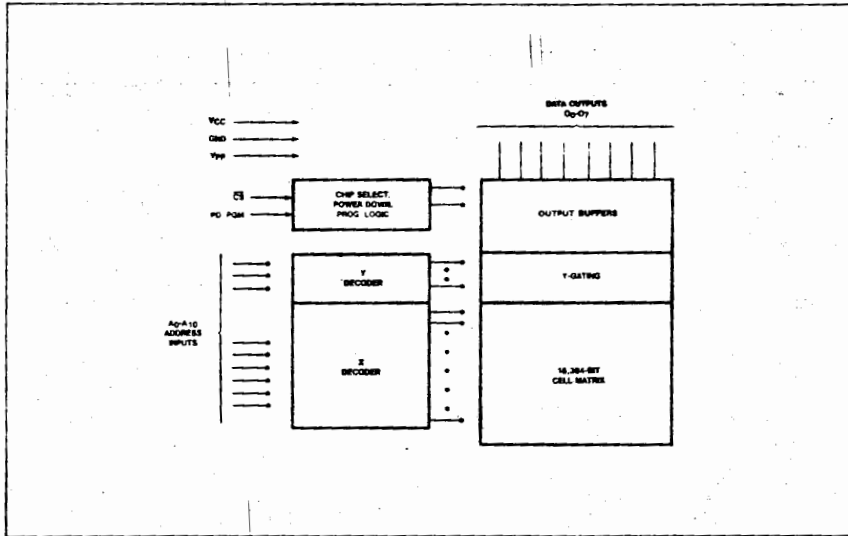
**FEATURES**

- Single +5V power supply
- Simple programming:
  - Single location programming
  - Single 50ms pulse
  - TTL level signals
- Fast programming—100sec
- Fast access time—450ns max
- Low Power dissipation:
  - 525mW max, active power
  - 132mW max, standby power
- Pin compatible to Signetics 2616 ROM

**PIN CONFIGURATION**



**BLOCK DIAGRAM**



## 16,384-BIT ERASABLE AND REPROGRAMMABLE MOS ROM (2048X8)

2716

PRELIMINARY SPECIFICATION

2716-1

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

| PARAMETER | RATING  | UNIT       |
|-----------|---|------------|
| $T_A$     | Temperature range                                   | °C         |
|           | Operating   |            |
| $T_{STG}$ | Storage   | -10 to 80  |
|           | All input or output voltages with respect to ground | -85 to 125 |
|           | $V_{pp}$ supply voltage with respect to ground      | -0.3 to 6  |
|           |   | -0.3 to 28 |

## NOTE

<sup>1</sup> Stresses above those listed as "Absolute Maximum Ratings" may damage the device. These ratings are meant for short term stress only; prolonged exposure at these ratings may affect device reliability.

## PIN DESIGNATION

| PIN NO.        | SYMBOL           | FUNCTION               |
|----------------|------------------|------------------------|
| 1-8, 22-23, 19 | $A_0$ - $A_{10}$ | Address inputs         |
| 18             | PD/PGM           | Power down/Program     |
| 20             | CS               | Chip select            |
| 9-11, 13-17    | $O_0$ - $O_7$    | Outputs                |
| 24             | $V_{CC}$         | Power (+5V)            |
| 21             | $V_{pp}$         | Program voltage (+25V) |
| 12             | GND              | Ground                 |

DC ELECTRICAL CHARACTERISTICS  $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC}^2 = +5\text{V}$ ,  $\pm 5\%$ ,  $V_{pp}^3 = V_{CC} \pm 0.8\text{V}^4$ , unless otherwise specified.

| PARAMETER   | TEST CONDITIONS            | LIMITS |                  |              | UNIT          |
|-------------|----------------------------|--------|------------------|--------------|---------------|
|             |                            | Min    | Typ <sup>5</sup> | Max          |               |
| $V_{IL}$    | Input voltage              |        |                  |              | V             |
| $V_{IH}$    | High                       | -0.1   |                  | 0.8          |               |
| $V_{OL}$    | Output voltage             |        |                  |              | V             |
| $V_{OH}$    | High                       |        |                  | $V_{CC} + 1$ |               |
| $I_{LI}$    | Input load current         |        |                  | 10           | $\mu\text{A}$ |
| $I_{LO}$    | Output leakage current     |        |                  | 10           | $\mu\text{A}$ |
| $I_{pp}^3$  | $V_{pp}$ current           |        |                  | 5            | mA            |
| $I_{CC1}^3$ | $V_{CC}$ current (standby) |        | 10               | 25           |               |
| $I_{CC}^3$  | $V_{CC}$ current (active)  |        | 57               | 100          |               |
| $C_{IN}$    | Capacitance <sup>6</sup>   |        |                  |              | pF            |
| $C_{OUT}$   | Output                     |        |                  |              |               |

## NOTES

- $V_{CC}$  must be applied simultaneously or before  $V_{pp}$  and removed simultaneously or after  $V_{pp}$ .
- $V_{pp}$  may be connected directly to  $V_{CC}$  except during programming. The  $V_{CC}$  supply current would then be the sum of  $I_{CC}$  and  $I_{pp}$ .
- The tolerance of 0.8V allows the use of a driver circuit for switching the  $V_{pp}$  supply pin from  $V_{CC}$  as used to 25V for programming.
- Typical values are for  $T_A = 25^\circ\text{C}$  and nominal supply voltages.
- This parameter is only sampled and is not 100% tested.
- $I_{ACC}^3$  is referenced to PD/PGM or the address, whichever occurs last.

**16,384-BIT ERASABLE AND REPROGRAMMABLE MOS ROM (2048X8)**

**2716**

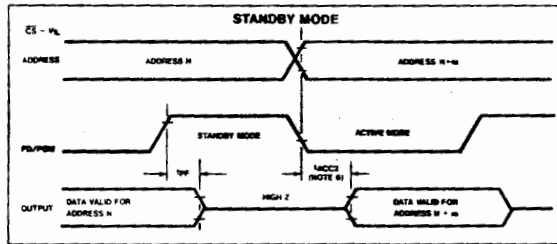
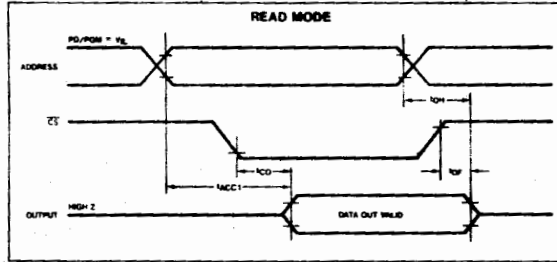
PRELIMINARY SPECIFICATION

2716-1

AC ELECTRICAL CHARACTERISTICS  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = +5V, \pm 5\%$ ,  $V_{pp} = V_{CC} \pm 0.6V$ , unless otherwise specified.

| PARAMETER  | TO                         | FROM                              | TEST CONDITIONS  | LIMITS |                  |                   | UNIT |
|--|----------------------------|-----------------------------------|--|--------|------------------|-------------------|------|
|  |                            |                                   |  | Min    | Typ <sup>5</sup> | Max               |      |
| Delay time<br>$t_{ACC1}$<br>$t_{ACC2}$<br>$t_{CO}$ | Output<br>Output<br>Output | Address<br>PD, PGM<br>Chip select | PD/PGM = $\overline{CS} = V_{IL}$<br><br>PD/PGM = $V_{IL}$ |        | 250<br>280       | 450<br>450<br>150 | ns   |
| Float time<br>$t_{PF}$<br>$t_{DF}$                 | Output<br>Output           | PD, PGM<br>Chip deselect          | $\overline{CS} = V_{IL}$<br>PD/PGM = $V_{IL}$              | 0<br>0 |                  | 100<br>100        | ns   |
| $t_{OH}$   | Output                     | Address                           | PD/PGM = $\overline{CS} = V_{IL}$                          | 0      |                  |                   | ns   |

**VOLTAGE WAVEFORMS**



**Signetics**

## 16,384-BIT ERASABLE AND REPROGRAMMABLE MOS ROM (2048X8)

2716

PRELIMINARY SPECIFICATION

2716-1

DC PROGRAMMING CHARACTERISTICS<sup>8</sup>  $T_A = 25^\circ\text{C} \pm 5^\circ\text{C}$ ,  $V_{CC}^9 = 5\text{V} \pm 5\%$ ,  $V_{pp}^{9,10} = 25\text{V} \pm 1\text{V}$ , unless otherwise specified.

| PARAMETER | TEST CONDITIONS                                  | LIMITS                       |     |              | UNIT          |
|-----------|--|------------------------------|-----|--------------|---------------|
|           |  | Min                          | Typ | Max          |               |
| $I_{LI}$  | Input current (for any input)                    | $V_{IN} = 5.25\text{V}/0.45$ |     |              | $\mu\text{A}$ |
| $I_{PP1}$ | $V_{pp}$ supply current                          | $PD/PGM = V_{IL}$            |     |              | $\text{mA}$   |
| $I_{PP2}$ | $V_{pp}$ supply current during programming pulse | $PD/PGM = V_{IH}$            |     |              | $\text{mA}$   |
| $I_{CC}$  | $V_{CC}$ supply current                          |                              |     |              | $\text{mA}$   |
| $V_{IL}$  | Input low level                                  | -0.1                         |     | 0.8          | V             |
| $V_{IH}$  | Input high level                                 | 2.2                          |     | $V_{CC} + 1$ | V             |

AC PROGRAMMING CHARACTERISTICS<sup>8</sup>  $T_A = 25^\circ\text{C} \pm 5^\circ\text{C}$ ,  $V_{CC}^9 = 5\text{V} \pm 5\%$ ,  $V_{pp}^{9,10} = 25\text{V} \pm 1\text{V}$ , unless otherwise specified.

| PARAMETER | TEST CONDITIONS                     | LIMITS |     |     | UNIT          |
|-----------|-------------------------------------|--------|-----|-----|---------------|
|           |                                     | Min    | Typ | Max |               |
| $t_{AS}$  | Address setup time                  | 2      |     |     | $\mu\text{s}$ |
| $t_{CSS}$ | CS setup time                       | 2      |     |     | $\mu\text{s}$ |
| $t_{DS}$  | Data setup time                     | 2      |     |     | $\mu\text{s}$ |
| $t_{AH}$  | Address hold time                   | 2      |     |     | $\mu\text{s}$ |
| $t_{CSH}$ | CS hold time                        | 2      |     |     | $\mu\text{s}$ |
| $t_{DH}$  | Data hold time                      | 2      |     |     | $\mu\text{s}$ |
| $t_{DF}$  | Chip deselect to output float delay | 0      |     | 120 | ns            |
| $t_{CO}$  | Chip select to output delay         |        |     | 150 | ns            |
| $t_{PW}$  | Program pulse width                 | 45     | 50  | 55  | ns            |
| $t_{PRT}$ | Program pulse rise time             | 5      |     |     | ns            |
| $t_{PFT}$ | Program pulse fall time             | 5      |     |     | ns            |

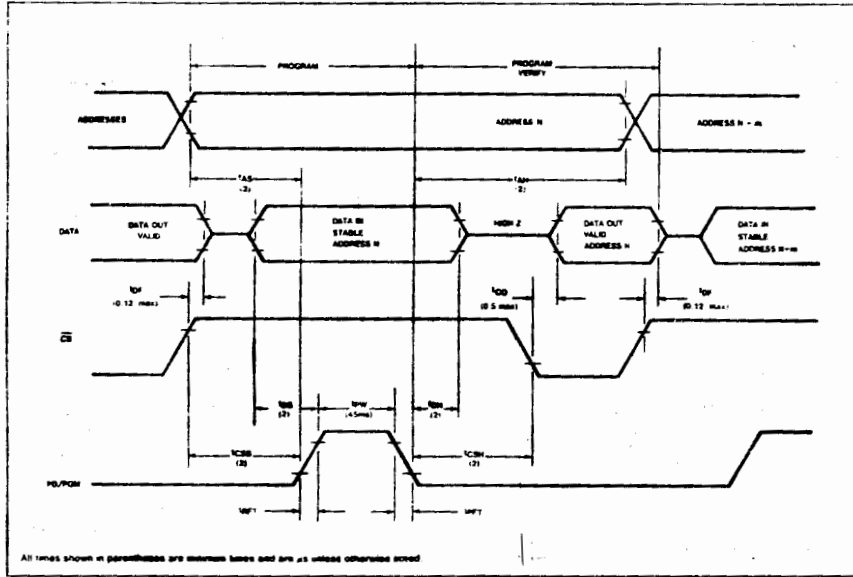
## NOTES

8. Signetics standard product warranty applies only to devices programmed to specifications described herein.
9.  $V_{CC}$  must be applied simultaneously or before  $V_{pp}$  and removed simultaneously or after  $V_{pp}$ . The 2716 must not be inserted into or removed from a board with  $V_{pp}$  at  $\pm 1\text{V}$  to prevent damage to the device.
10. The maximum allowable voltage which may be applied to the  $V_{pp}$  pin during programming is +26V. Care must be taken when switching the  $V_{pp}$  supply to prevent overshoot exceeding this 26V maximum specification.

1.2



PROGRAMMING WAVEFORM



All times shown in parentheses are minimum values and are  $\mu$ s unless otherwise noted

ERASE

The 2716 can be erased by using the following procedure:

- Use an ultraviolet lamp with a wave length of 2537 angstroms (Å)
- Place lamp tube within 1 inch of device
- Remove any filter from lamp tube
- Expose device for a total dose (intensity x exposure time) of 15W-sec/cm<sup>2</sup>

This procedure places all 16,384 bits in the HIGH or logic "1" state ready for selective electrical programming. Light sources with wave lengths shorter than 4000Å can cause unintentional erasure. Direct sunlight could erase a typical 2716 in 1 week while room level fluorescent could do the same in approximately 3 years. Signetics has opaque labels available if the 2716 is to be exposed to high ambient light levels for extended periods. These labels could also be used during erasure of parts on the board. The labels would be placed on devices with good programs.

The following list of ultraviolet lamps are available from Ultra violet Products Inc.

5114 Walnut Grove Ave., San Gabriel, CA:

| MODEL  | POWER RATING                  | TIME FOR 15W-sec/cm <sup>2</sup> |
|--------|-------------------------------|----------------------------------|
| S-68   | 12000 $\mu$ W/cm <sup>2</sup> | 21 minutes                       |
| S-52   | 12000 $\mu$ W/cm <sup>2</sup> | 21 minutes                       |
| UVS-54 | 5700 $\mu$ W/cm <sup>2</sup>  | 44 minutes                       |
| R-52   | 13000 $\mu$ W/cm <sup>2</sup> | 20 minutes                       |
| UVS-11 | 5500 $\mu$ W/cm <sup>2</sup>  | 46 minutes                       |

It should be noted that ultraviolet lamps have a tendency to degrade with constant On-Off operation.

PROGRAM

After erasure all bits are in the "1" state. Programming to a "0" is accomplished using the following procedure:

- PD/PGM initially low ( $V_{IL}$ )
- VCC to +5V, Vpp to +25V
- CS HIGH ( $V_{IH}$ )
- Select desired word by applying addresses
- Pull outputs for "0" bits low ( $V_{IL}$ )
- Apply single active HIGH TTL program pulse to PD/PGM

• Repeat for all words

Multiple 2716's can be loaded with the same data by paralleling all pins and following the program procedure.

Multiple 2716's can be loaded with different data by paralleling all pins except PD/PGM and following the program procedure. The program pulse should be applied to the selected chip while all deselected chips have their PD/PGM inputs held LOW.

The program can be verified without reducing Vpp by holding both the PD/PGM and CS pins LOW.

READ

The Vpp pin should be at +5 volts for all except program operations. A LOW on both the PD/PGM and CS pins presents the data of the selected word to the pins. A HIGH on either PD/PGM or CS deselects the chip for reading of parallel devices. A HIGH on PD/PGM reduces the active power dissipation by 75%.



# HM6116P-2, HM6116P-3, HM6116P-4

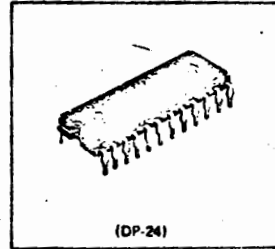
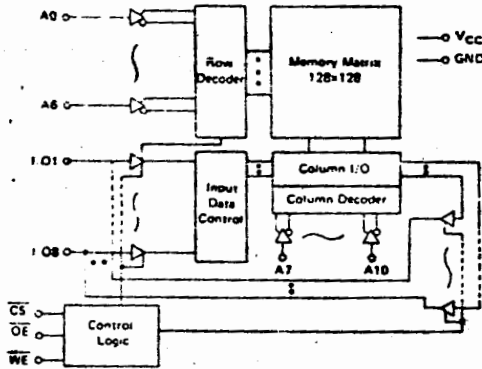


## 2048-word X 6-bit High Speed Static CMOS RAM

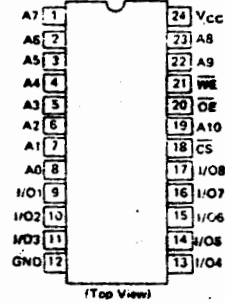
### FEATURES

- Single 5V Supply and High Density 24 pin Package
- High Speed: Fast Access Time 120ns/150ns/200ns (max.)
- Low Power Standby and Low Power Operation; Standby: 100μW (typ.)  
Operation: 180mW (typ.)
- Completely Static RAM: No clock or Timing Strobe Required
- Directly TTL Compatible: All Input and Output
- Pin Out Compatible with Standard 16K EPROM/MASK ROM
- Equal Access and Cycle Time

### FUNCTIONAL BLOCK DIAGRAM



### PIN ARRANGEMENT



### ABSOLUTE MAXIMUM RATINGS

| Item                               | Symbol     | Rating       | Unit |
|------------------------------------|------------|--------------|------|
| Voltage on Any Pin Relative to GND | $V_{IN}$   | -0.5 to +7.0 | V    |
| Operating Temperature              | $T_{OP}$   | 0 to +70     | °C   |
| Storage Temperature                | $T_{ST}$   | -55 to +125  | °C   |
| Temperature Under Bias             | $T_{bias}$ | -10 to +85   | °C   |
| Power Dissipation                  | $P_T$      | 1.0          | W    |

### TRUTH TABLE

| CS | OE | WE | Mode         | $V_{CC}$ Current   | I/O Pin | Ref. Cycle           |
|----|----|----|--------------|--------------------|---------|----------------------|
| H  | X  | X  | Not Selected | $I_{SB} / I_{SB1}$ | High Z  |                      |
| L  | L  | H  | Read         | $I_{CC}$           | Dout    | Read Cycle (1) ~ (3) |
| L  | H  | L  | Write        | $I_{CC}$           | Di      | Write Cycle (1)      |
| L  | L  | L  | Write        | $I_{CC}$           | Di      | Write Cycle (2)      |

■ RECOMMENDED DC OPERATING CONDITIONS ( $T_a = 0$  to  $+70^\circ\text{C}$ )

| Item           | Symbol   | min   | typ. | max. | Unit |
|----------------|----------|-------|------|------|------|
| Supply Voltage | $V_{CC}$ | 4.5   | 5.0  | 5.5  | V    |
|                | GND      | 0     | 0    | 0    | V    |
| Input Voltage  | $V_{IH}$ | 2.2   | 3.5  | 6.0  | V    |
|                | $V_{IL}$ | -1.0* | -    | 0.8  | V    |

\* Pulse Width: 50 ns, DC  $V_{IL}$  min. = -0.3V.

■ DC AND OPERATING CHARACTERISTICS ( $V_{CC} = 5\text{V} \pm 10\%$ , GND = 0V,  $T_a = 0$  to  $+70^\circ\text{C}$ )

| Item                           | Symbol       | Test Conditions   | HM6116P-2 |      |      | HM6116P-3/4 |      |      | Unit          |
|--------------------------------|--------------|---|-----------|------|------|-------------|------|------|---------------|
|                                |              |   | min       | typ* | max. | min         | typ* | max. |               |
| Input Leakage Current          | $I_{IL1}$    | $V_{CC} = 5.5\text{V}$ , $V_{in} = \text{GND to } V_{CC}$   | -         | -    | 10   | -           | -    | 10   | $\mu\text{A}$ |
| Output Leakage Current         | $I_{LO1}$    | $V_{CC} = V_{IH}$ or $\text{DE} = V_{IH}$ ,<br>$V_{IO} = \text{GND to } V_{CC}$                         | -         | -    | 10   | -           | -    | 10   | $\mu\text{A}$ |
| Operating Power Supply Current | $I_{CC}$     | $\text{CS} = V_{IL}$ , $I_{IO} = 0\text{mA}$  | -         | 40   | 80   | -           | 35   | 70   | mA            |
|                                | $I_{CC1}$ ** | $V_{IH} = 3.5\text{V}$ , $V_{IL} = 0.6\text{V}$ ,<br>$I_{IO} = 0\text{mA}$                              | -         | 35   | -    | -           | 30   | -    | mA            |
| Average Operating Current      | $I_{CC2}$    | Min. cycle duty = 100%  | -         | 40   | 80   | -           | 35   | 70   | mA            |
| Standby Power Supply Current   | $I_{SB}$     | $\text{CS} = V_{IH}$  | -         | 5    | 15   | -           | 5    | 15   | mA            |
|                                | $I_{SB1}$    | $\text{CS} \geq V_{CC} - 0.2\text{V}$ , $V_{in} \geq V_{CC} - 0.2\text{V}$ or $V_{in} \leq 0.2\text{V}$ | -         | 0.02 | 2    | -           | 0.02 | 2    | mA            |
| Output Voltage                 | $V_{OL}$     | $I_{OL} = 4\text{mA}$   | -         | -    | 0.4  | -           | -    | -    | V             |
|                                |              | $I_{OL} = 2.1\text{mA}$   | -         | -    | -    | -           | -    | 0.4  | V             |
|                                | $V_{OH}$     | $I_{OH} = -1.0\text{mA}$  | 2.4       | -    | -    | 2.4         | -    | -    | V             |

\*  $V_{CC} = 5\text{V}$ ,  $T_a = 25^\circ\text{C}$

\*\* Reference Only

■ AC CHARACTERISTICS ( $V_{CC} = 5\text{V} \pm 10\%$ ,  $T_a = 0$  to  $+70^\circ\text{C}$ )

● AC TEST CONDITIONS

Input Pulse Levels: 0.8 to 2.4V  
Input Rise and Fall Times: 10 ns

Input and Output Timing Reference Levels: 1.5V  
Output Load: 1TTL Gate and  $C_L = 100\text{pF}$   
(including scope and Jig)

● READ CYCLE

| Item                                | Symbol    | HM6116P-2 |     | HM6116P-3 |     | HM6116P-4 |     | Unit |
|-------------------------------------|-----------|-----------|-----|-----------|-----|-----------|-----|------|
|                                     |           | min       | max | min       | max | min       | max |      |
| Read Cycle Time                     | $t_{RC}$  | 120       | -   | 150       | -   | 200       | -   | ns   |
| Address Access Time                 | $t_{AA}$  | -         | 120 | -         | 150 | -         | 200 | ns   |
| Chip Select Access Time             | $t_{ACS}$ | -         | 120 | -         | 150 | -         | 200 | ns   |
| Chip Selection to Output in Low Z   | $t_{CLZ}$ | 10        | -   | 15        | -   | 15        | -   | ns   |
| Output Enable to Output Valid       | $t_{OE}$  | -         | 80  | -         | 100 | -         | 120 | ns   |
| Output Enable to Output in Low Z    | $t_{OLZ}$ | 10        | -   | 15        | -   | 15        | -   | ns   |
| Chip deslection to Output in High Z | $t_{CHZ}$ | 0         | 40  | 0         | 50  | 0         | 60  | ns   |
| Chip Disable to Output in High Z    | $t_{OHZ}$ | 0         | 40  | 0         | 50  | 0         | 60  | ns   |
| Output Hold from Address Change     | $t_{OH}$  | 10        | -   | 15        | -   | 15        | -   | ns   |

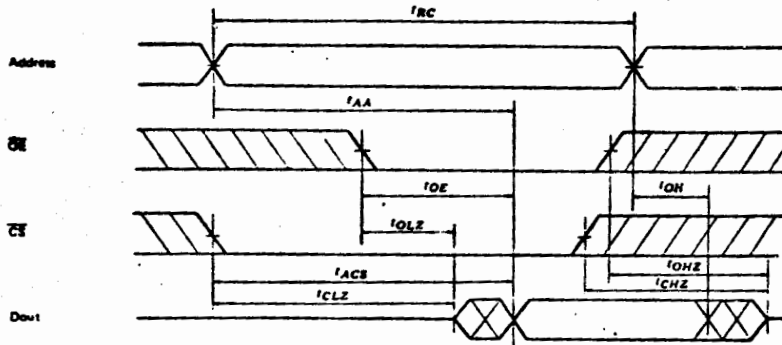
● WRITE CYCLE

| Item                               | Symbol    | HM6116P-2 |      | HM6116P-3 |      | HM6116P-4 |      | Unit |
|------------------------------------|-----------|-----------|------|-----------|------|-----------|------|------|
|                                    |           | min       | typ. | min       | max. | min       | max. |      |
| Write Cycle Time                   | $t_{WC}$  | 120       | -    | 150       | -    | 200       | -    | ns   |
| Chip Selection to End of Write     | $t_{CW}$  | 70        | -    | 90        | -    | 120       | -    | ns   |
| Address Valid to End of Write      | $t_{AW}$  | 105       | -    | 120       | -    | 140       | -    | ns   |
| Address Set Up Time                | $t_{AS}$  | 20        | -    | 20        | -    | 20        | -    | ns   |
| Write Pulse Width                  | $t_{WP}$  | 70        | -    | 90        | -    | 120       | -    | ns   |
| Write Recovery Time                | $t_{WR}$  | 5         | -    | 10        | -    | 10        | -    | ns   |
| Output Disable to Output in High Z | $t_{OHZ}$ | 0         | 40   | 0         | 50   | 0         | 60   | ns   |
| Write to Output in High Z          | $t_{WHZ}$ | 0         | 50   | 0         | 60   | 0         | 60   | ns   |
| Data to Write Time Overlap         | $t_{DW}$  | 35        | -    | 40        | -    | 60        | -    | ns   |
| Data Hold from Write Time          | $t_{DH}$  | 5         | -    | 10        | -    | 10        | -    | ns   |
| Output Active from End of Write    | $t_{OW}$  | 5         | -    | 10        | -    | 10        | -    | ns   |

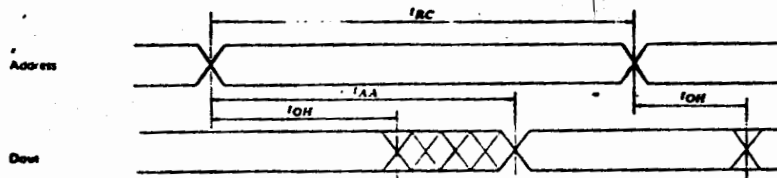
■ CAPACITANCE ( $V = 1\text{MHz}$ ,  $T_a = 25^\circ\text{C}$ )

| Item                     | Symbol    | Test Conditions | typ | max | Unit |
|--------------------------|-----------|-----------------|-----|-----|------|
| Input Capacitance        | $C_{in}$  | $V_{in} = 0V$   | 3   | 5   | pF   |
| Input/Output Capacitance | $C_{i/o}$ | $V_{i/o} = 0V$  | 5   | 7   | pF   |

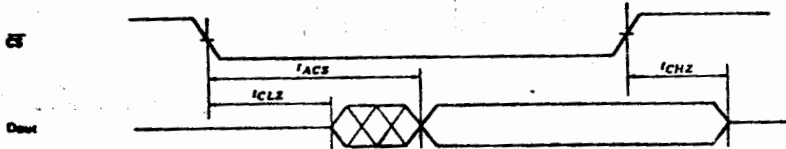
● Read Cycle (1) Notes 1, 5



● Read Cycle (2) Notes 1, 2, 4, 5



● Read Cycle (3) Notes 1, 3, 4, 5

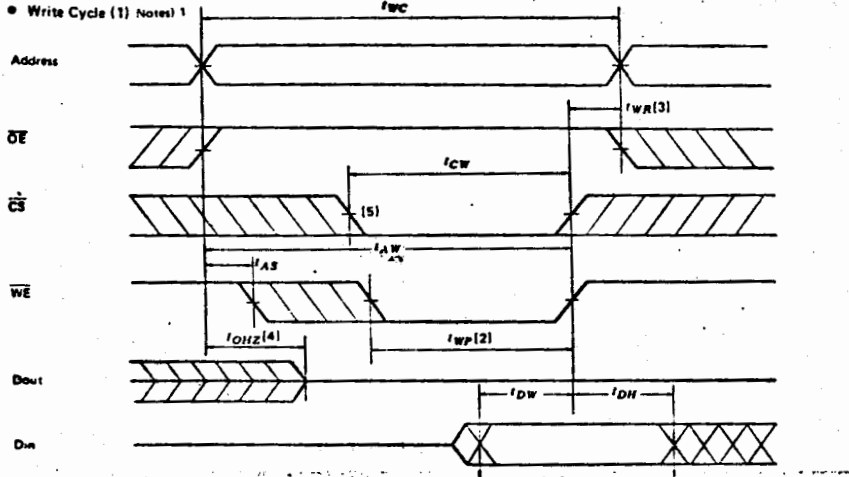


- NOTES
- 1  $\overline{WE}$  is High for Read Cycle.
  - 2 Device is continuously selected,  $\overline{CS} = V_{IL}$ .
  - 3 Address Valid prior to or coincident with  $\overline{CS}$  transition Low.
  - 4  $\overline{OE} = V_{IL}$ .
  - 5 When  $\overline{CS}$  is Low, the address input must not be in the high impedance state.

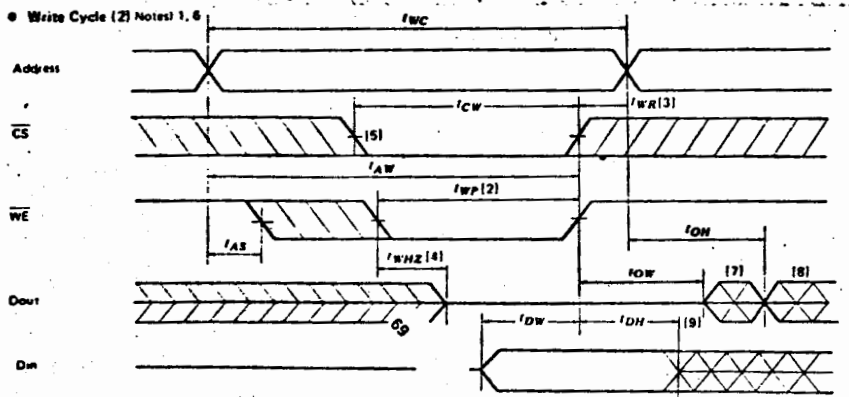
HM6116P-2, HM6116P-3, HM6116P-4

■ TIMING WAVEFORM

● Write Cycle (1) Notes 1

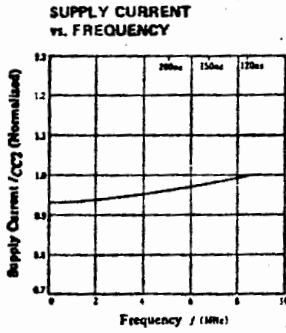
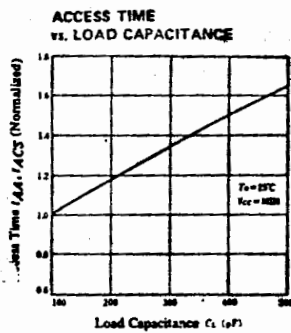
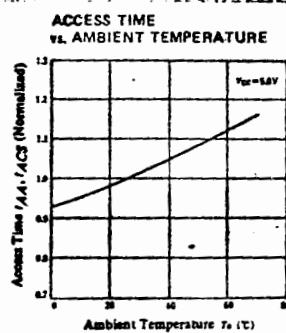
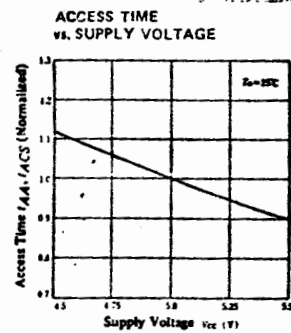
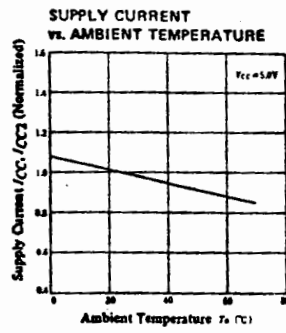
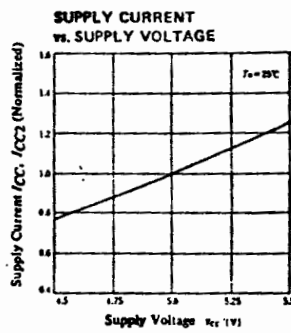


● Write Cycle (2) Notes 1, 6



- NOTES: 1.  $\overline{WE}$  must be high during all address transitions.  
 2. A write occurs during the overlap ( $t_{WP}$ ) of a low  $\overline{CS}$  and a low  $\overline{WE}$ .  
 3.  $t_{WR}$  is measured from the earlier of  $\overline{CS}$  or  $\overline{WE}$  going high to the end of write cycle.  
 4. During this period, I/O pins are in the output state so that the input signals of opposite phase to the outputs must not be applied.  
 5. If the  $\overline{CS}$  low transition occurs simultaneously with the  $\overline{WE}$

- low transitions or after the  $\overline{WE}$  transition, output remain in a high impedance state.  
 6.  $\overline{OE}$  is continuously low. ( $\overline{OE} = V_{IL}$ )  
 7.  $D_{out}$  is the same phase of write data of this write cycle.  
 8.  $D_{out}$  is the read data of next address.  
 9. If  $\overline{CS}$  is Low during this period, I/O pins are in the output state. Then the data input signals of opposite phase to the outputs must not be applied to them.



## APPLICATIONS OF THE MOC3011 TRIAC DRIVER

Prepared by:  
 Pat O'Neil

This note describes methods of applying the MOC3011 optically coupled triac driver to provide simple and effective interfaces from logic or microprocessor systems to AC power systems.

### DESCRIPTIONS OF THE MOC3011

#### Construction

The MOC3011 consists of a gallium arsenide infrared LED optically exciting a silicon detector chip, which is especially designed to drive triacs controlling loads on the 115 Vac power line. The detector chip is a complex device which functions in much the same manner as a small triac, generating the signals necessary to drive the gate of a larger triac. The MOC3011 allows a low power exciting signal to drive a high power load with a very small number of components, and at the same time provides practically complete isolation of the driving circuitry from the power line.

The construction of the MOC3011 follows the same highly successful coupler technology used in Motorola's broad line of plastic couplers (Figure 1). The dual lead

frame with an epoxy undermold provides a stable dielectric capable of sustaining 7.5 kV between the input and output sides of the device. The detector chip is passivated with silicon nitride and uses Motorola's annular ring to maintain stable breakdown parameters.

#### Basic Electrical Description

The GaAs LED has nominal 1.3 V forward drop at 10 mA and a reverse breakdown voltage greater than 3 V. The maximum current to be passed through the LED is 50 mA.

The detector has a minimum blocking voltage of 250 Vdc in either direction in the off state. In the on state, the detector will pass 100 mA in either direction with less than 3 V drop across the device. Once triggered into the on (conducting) state, the detector will remain there until the current drops below the holding current (typically 100  $\mu$ A) at which time the detector reverts to the off (non-conducting) state. The detector may be triggered into the on state by exceeding the forward blocking voltage, by voltage ramps across the detector at rates exceeding the static dv/dt rating, or by photons from the LED. The LED is guaranteed by the specifications to trigger the detector into the on state when 10 mA or more is passed through the LED. A similar device, the MOC3010, has exactly the same characteristics except it requires 15 mA to trigger.

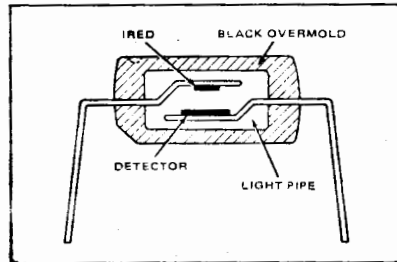


FIGURE 1 - Motorola Double-Molded Coupler Package



Since the MOC3011 looks essentially like a small optically triggered triac, we have chosen to represent it as shown on Figure 2.

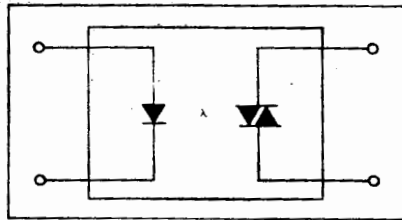


FIGURE 2 - Schematic Representation of MOC3011 and MOC3010

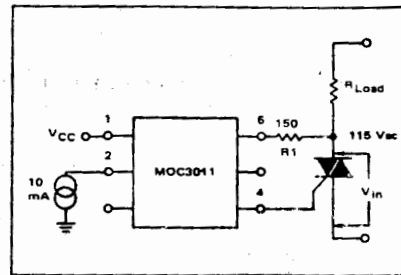


FIGURE 3 - Simple Triac Gating Circuit

**USING THE MOC3011 AS A TRIAC DRIVER**

**Triac Driving Requirements**

Figure 3 shows a simple triac driving circuit using the MOC3011. The maximum surge current rating of the MOC3011 sets the minimum value of R1 through the equation:

$$R1(\min) = V_{in(pk)} / 1.2 A$$

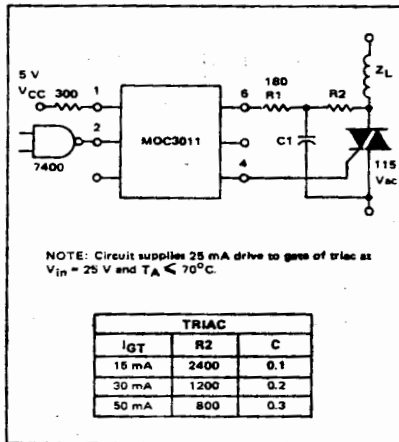
If we are operating on the 115 Vac nominal line voltage,  $V_{in(pk)} = 180 V$ , then

$$R1(\min) = V_{in(pk)} / 1.2 A = 150 \text{ ohms.}$$

In practice, this would be a 150 or 180 ohm resistor. If the triac has  $I_{GT} = 100 \text{ mA}$  and  $V_{GT} = 2 V$ , then the voltage  $V_{in}$  necessary to trigger the triac will be given by  $V_{inT} = R1 \cdot I_{GT} + V_{GT} + V_{TM} = 20 V$ .

**Resistive Loads**

When driving resistive loads, the circuit of Figure 3 may be used. Incandescent lamps and resistive heating elements are the two main classes of resistive loads for which 115 Vac is utilized. The main restriction is that the triac must be properly chosen to sustain the proper inrush loads. Incandescent lamps can sometimes draw a peak current known as "flashover" which can be extremely high, and the triac should be protected by a fuse or rated high enough to sustain this current.



NOTE: Circuit supplies 25 mA drive to gate of triac at  $V_{in} = 25 V$  and  $T_A \leq 70^\circ C$ .

| TRIAC    |      |     |
|----------|------|-----|
| $I_{GT}$ | R2   | C   |
| 15 mA    | 2400 | 0.1 |
| 30 mA    | 1200 | 0.2 |
| 50 mA    | 800  | 0.3 |

FIGURE 4 - Logic to Inductive Load Interface

Circuit diagrams utilizing Motorola products are included as a means of illustrating typical semiconductor applications; consequently, complete information sufficient for construction purposes is not necessarily given. The information has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any license under the patent rights of Motorola Inc. or others.

**Line Transients—Static dv/dt**

Occasionally transient voltage disturbance on the ac line will exceed the static dv/dt rating of the MOC3011. In this case, it is possible that the MOC3011 and the associated triac will be triggered on. This is usually not a problem, except in unusually noisy environments, because the MOC3011 and its triac will commute off at the next zero crossing of the line voltage, and most loads are not noticeably affected by an occasional single half-cycle of applied power. See Figure 5 for typical dv/dt versus temperature curves.

**Inductive Loads—Commutating dv/dt**

Inductive loads (motors, solenoids, magnets, etc.) present a problem both for triacs and for the MOC3011 because the voltage and current are not in phase with each other. Since the triac turns off at zero current, it may be trying to turn off when the applied current is zero but the applied voltage is high. This appears to the triac like a sudden rise in applied voltage, which turns on the triac if the rate of rise exceeds the commutating dv/dt of the triac or the static dv/dt of the MOC3011.

**Snubber Networks**

The solution to this problem is provided by the use of "snubber" networks to reduce the rate of voltage rise seen by the device. In some cases, this may require two snubbers—one for the triac and one for the MOC3011. The triac snubber is dependent upon the triac and load used and will not be discussed here. In many applications

the snubber used for the MOC3011 will also adequately protect the triac.

In order to design a snubber properly, one should really know the power factor of the reactive load, which is defined as the cosine of the phase shift caused by the load. Unfortunately, this is not always known, and this makes snubbing network design somewhat empirical. However a method of designing a snubber network may be defined, based upon a typical power factor. This can be used as a "first cut" and later modified based upon experiment.

Assuming an inductive load with a power factor of PF = 0.1 is to be driven. The triac might be trying to turn off when the applied voltage is given by

$$V_{to} = V_{pk} \sin \phi \approx V_{pk} \approx 180 \text{ V}$$

First, one must choose R1 (Figure 4) to limit the peak capacitor discharge current through the MOC3011. This resistor is given by

$$R1 = V_{pk} / I_{max} = 180 / 1.2 \text{ A} = 150 \Omega$$

A standard value, 160 ohm resistor can be used in practice for R1.

It is necessary to set the time constant for  $\tau = R_2 C$ . Assuming that the triac turns off very quickly, we have a peak rate of rise at the MOC3011 given by

$$dv/dt = V_{to} / \tau = V_{to} / R_2 C$$

Setting this equal to the worst case dv/dt (static) for the MOC3011 which we can obtain from Figure 5 and solving for R2C:

$$dv/dt(T_J = 70^\circ\text{C}) = 0.8 \text{ V}/\mu\text{s} = 8 \times 10^5$$

$$R_2 C = V_{to} / (dv/dt) = 180 / (8 \times 10^5) \approx 230 \times 10^{-6}$$

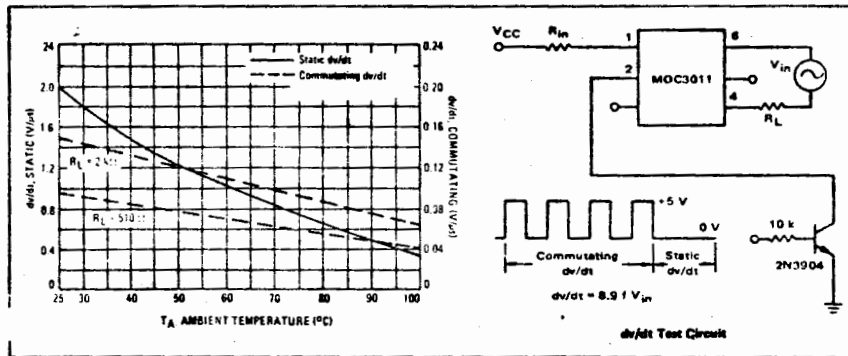


FIGURE 5 — dv/dt versus Temperature

**Input Protection Circuits**

In some applications, such as solid state relays, in which the input voltage varies widely the designer may want to limit the current applied to the LED of the MOC3011. The circuit shown in Figure 7 allows a non-critical range of input voltages to properly drive the MOC3011 and at the same time protects the input LED from inadvertent application of reverse polarity.

**LED Lifetime**

All light emitting diodes slowly decrease in brightness during their useful life, an effect accelerated by high temperatures and high LED currents. To allow a safety margin and insure long service life, the MOC3011 is actually tested to trigger at a value lower than the specified 10 mA input threshold current. The designer can therefore design the input circuitry to supply 10 mA to the LED and still be sure of satisfactory operation over

a long operating lifetime. On the other hand, care should be taken to insure that the maximum LED input current (50 mA) is not exceeded or the lifetime of the MOC3011 may be shortened.

**APPLICATIONS EXAMPLES**

**Using the MOC3011 on 240 Vac Lines**

The rated voltage of a MOC3011 is not sufficiently high for it to be used directly on 240 Vac line, however, the designer may stack two of them in series. When used this way, two resistors are required to equalize the voltage dropped across them as shown in Figure 8.

**Remote Control of ac Voltage**

Local budding codes frequently require all 115 Vac light switch wiring to be enclosed in conduit. By using a MOC3011, a triac, and a low voltage source, it is

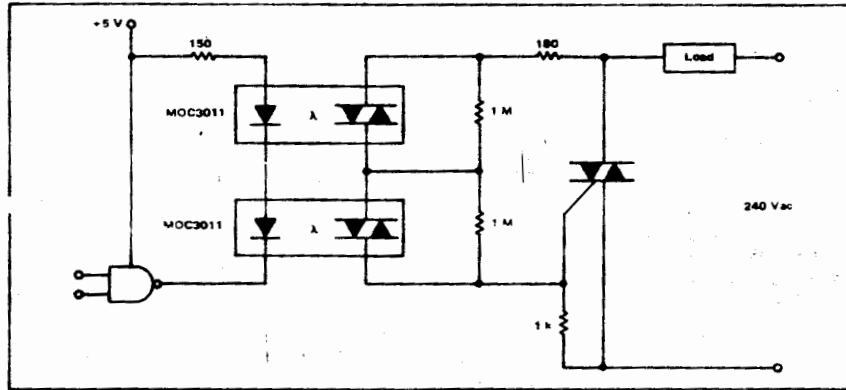


FIGURE 8 - 2 MOC3011 Triac Drivers in Series to Drive 240 V Triac

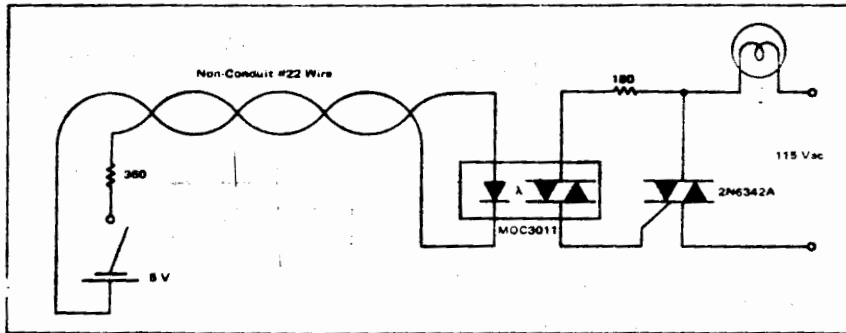


FIGURE 9 - Remote Control of ac Loads Through Low Voltage Non-Conduit Cable

The largest value of R2 available is found, taking into consideration the triac gate requirements. If a sensitive gate triac is used, such as a 2N6071B, I<sub>GT</sub> = 15 mA @ -40°C. If the triac is to be triggered when V<sub>in</sub> < 40 V

$$(R1 + R2) \approx V_{in}/I_{GT} \approx 40/0.015 \approx 2.3 \text{ k}$$

If we let R2 = 2400 ohms and C = 0.1 μF, the snubbing requirements are met. Triacs having less sensitive gates will require that R2 be lower and C be correspondingly higher as shown in Figure 4.

**INPUT CIRCUITRY**

**Resistor Input**

When the input conditions are well controlled, as for example when driving the MOC3011 from a TTL, DTL, or HTL gate, only a single resistor is necessary to interface the gate to the input LED of the MOC3011. This resistor should be chosen to set the current into the LED to be a minimum of 10 mA but no more than 50 mA. 15 mA is a suitable value, which allows for considerable degradation of the LED over time, and assures a long operating life for the coupler. Currents higher than 15 mA do not improve performance and may hasten the aging process inherent LED's. Assuming the forward drop to be 1.5 V at

15 mA allows a simple formula to calculate the input resistor.

$$R_i = (V_{CC} - 1.5)/0.015$$

Examples of resistive input circuits are seen in Figures 2 and 6.

**Increasing Input Sensitivity**

In some cases, the logic gate may not be able to source or sink 15 mA directly. CMOS, for example, is specified to have only 0.5 mA output, which must then be increased to drive the MOC3011. There are numerous ways to increase this current to a level compatible with the MOC3011 input requirements; an efficient way is to use the MC14049B shown in Figure 6. Since there are six such buffers in a single package, the user can have a small package count when using several MOC3011's, in one system.

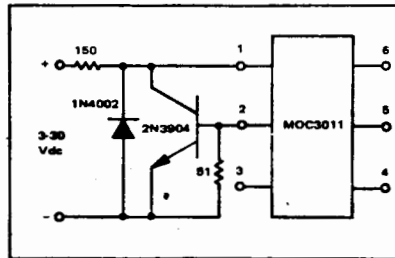


FIGURE 7 - MOC3011 Input Protection Circuit

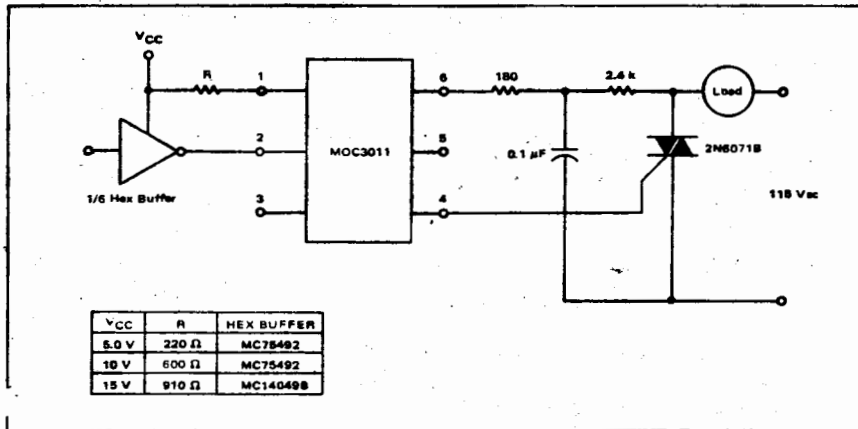


FIGURE 6 - MOS to ac Load Interface

possible to control a large lighting load from a long distance through low voltage signal wiring which is completely isolated from the ac line. Such wiring usually is not required to be put in conduit, so the cost savings in installing a lighting system in commercial or residential buildings can be considerable. An example is shown in Figure 9. Naturally, the load could also be a motor, fan, pool pump, etc.

**Solid State Relay**

Figure 10 shows a complete general purpose, solid state relay snubbed for inductive loads with input protection. When the designer has more control of the input and output conditions, he can eliminate those components which are not needed for his particular application to make the circuit more cost effective.

(I/O) port is a TTL-compatible terminal capable of driving one or two TTL loads. This is not quite enough to drive the MOC3011, nor can it be connected directly to an SCR or triac, because computer common is not normally referenced to one side of the ac supply. Standard 7400 series gates can provide an input compatible with the output of an MC6820, MC6821, MC6846 or similar peripheral interface adaptor and can directly drive the MOC3011. If the second input of a 2 input gate is tied to a simple timing circuit, it will also provide energization of the triac only at the zero crossing of the ac line voltage as shown in Figure 11. This technique extends the life of incandescent lamps, reduces the surge current strains on the triac, and reduces EMI generated by load switching. Of course, zero crossing can be generated within the microcomputer itself, but this requires considerable software overhead and usually just as much hardware to generate the zero-crossing timing signals.

**Interfacing Microprocessors to 115 Vac Peripherals**

The output of a typical microcomputer input-output

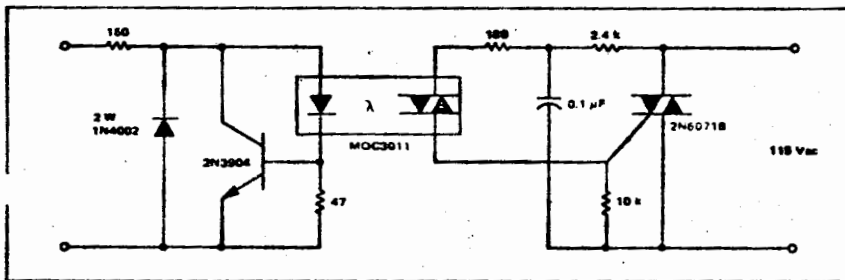


FIGURE 10 - Solid-State Relay

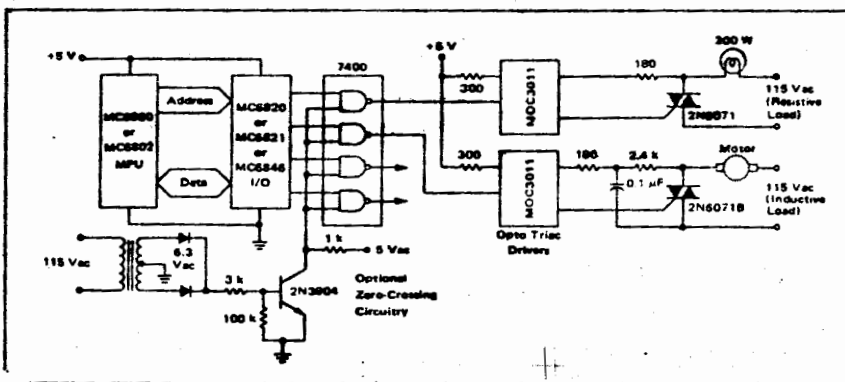


FIGURE 11 - Interfacing an MCB00 Microcomputer System to 115 Vac Loads

**ADC0801, ADC0802, ADC0803, ADC0804**  
8-Bit  $\mu$ P Compatible A/D Converters

**General Description**

The ADC0801, ADC0802, ADC0803, ADC0804 are CMOS 8-bit successive approximation A/D converters which use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the 8080A control bus, and TRI-STATE<sup>®</sup> output latches directly drive the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

A new differential analog voltage input allows increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

**Features**

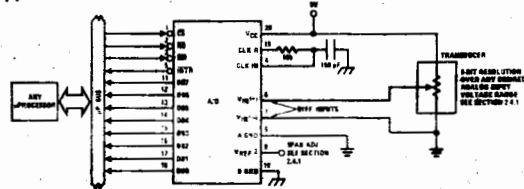
- Compatible with 8080  $\mu$ P derivatives—no interfacing logic needed
- Easy interface to all microprocessors, or operates "stand alone"

- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package

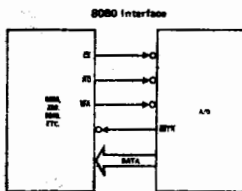
**Key Specifications**

- Resolution: 8 bits
- Total error:  $\pm 1/4$  LSB,  $\pm 1/2$  LSB and  $\pm 1$  LSB
- Conversion time: 100  $\mu$ s
- Access time: 135 ns
- Single supply: 5 VDC
- Operates ratiometrically or with 5 VDC, 2.5 VDC, or analog span adjusted voltage reference

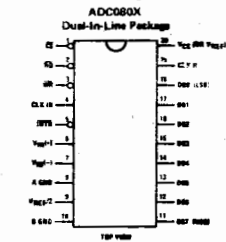
**Typical Applications**



**Connection Diagram**



**8080 Interface**



**ADC0801X**

**Dual-In-Line Package**

TRI-STATE<sup>®</sup> is a registered trademark of National Semiconductor Corp.

© 1981 National Semiconductor Corp.

NO FL20811 Printed in U.S.A.

ADC0801, ADC0802, ADC0803, ADC0804 8-Bit  $\mu$ P Compatible A/D Converters

**Absolute Maximum Ratings** (Notes 1 and 2)

|  |                                   |
|--|-----------------------------------|
| Supply Voltage (V <sub>CC</sub> ) (Note 3)   | 6.5V                              |
| Voltage Logic Control Inputs                 | -0.3V to +18V                     |
| At Other Inputs and Outputs                  | -0.3V to (V <sub>CC</sub> + 0.3V) |
| Storage Temperature Range                    | -65°C to +150°C                   |
| Package Dissipation at T <sub>A</sub> = 25°C | 875 mW                            |
| Lead Temperature (Soldering, 10 seconds)     | 300°C                             |

**Operating Ratings** (Notes 1 and 2)

|                                   |  |
|-----------------------------------|--|
| Temperature Range (Note 1)        | T <sub>MIN</sub> ≤ T <sub>A</sub> ≤ T <sub>MAX</sub> |
| ADC0B01/02/03 LQ                  | -65°C ≤ T <sub>A</sub> ≤ +125°C                      |
| ADC0B01/02/03/04 LCD              | -40°C ≤ T <sub>A</sub> ≤ +85°C                       |
| ADC0B01/02/03/04 LCN              | 0°C ≤ T <sub>A</sub> ≤ 70°C                          |
| Range of V <sub>CC</sub> (Note 1) | 4.5 V <sub>DC</sub> to 6.3 V <sub>DC</sub>           |

**Electrical Characteristics**

The following specifications apply for V<sub>CC</sub> = 5 V<sub>DC</sub>, T<sub>MIN</sub> ≤ T<sub>A</sub> ≤ T<sub>MAX</sub> and f<sub>CLK</sub> = 640 kHz unless otherwise specified.

| PARAMETER                                     | CONDITIONS   | MIN        | TYP        | MAX                   | UNITS           |
|---|--|------------|------------|-----------------------|-----------------|
| ADC0B01<br>Total Adjusted Error<br>(Note 6)   | With Full-Scale Adj<br>(See Section 2.5.2)   |            |            | ±1A                   | LSB             |
| ADC0B02<br>Total Unadjusted Error<br>(Note 6) | V <sub>REF/2</sub> = 2.500 V <sub>DC</sub><br>(I <sub>POUR</sub> ≤ 380μA)  |            |            | ±1/2                  | LSB             |
| ADC0B03<br>Total Adjusted Error<br>(Note 6)   | With Full-Scale Adj<br>(See Section 2.5.2)   |            |            | ±1/2                  | LSB             |
| ADC0B04<br>Total Unadjusted Error<br>(Note 6) | V <sub>REF/2</sub> = 2.500 V <sub>DC</sub><br>(I <sub>POUR</sub> ≤ 380μA)  |            |            | ±1                    | LSB             |
| V <sub>REF/2</sub> Input Resistance (Pin 6)   | ADC0B01/02/03<br>ADC0B04 (Note 6)  | 2.5<br>1.9 | 8.0<br>1.3 |                       | kΩ<br>kΩ        |
| Analog Input Voltage Range                    | (Note 4) V <sub>H</sub> or V <sub>L</sub><br>Gnd-0.05  |            |            | V <sub>CC</sub> +0.05 | V <sub>DC</sub> |
| DC Conversion Mode Rejection                  | Over Analog Input Voltage Range  |            | ±1/16      | ±1/8                  | LSB             |
| Power Supply Sensitivity                      | V <sub>CC</sub> = 5 V <sub>DC</sub> ±10% Over<br>Allowed V <sub>REF</sub> (+) and V <sub>REF</sub> (-)<br>Voltage Range (Note 4) |            | ±1/16      | ±1/8                  | LSB             |

**AC Electrical Characteristics**

The following specifications apply for V<sub>CC</sub> = 5 V<sub>DC</sub> and T<sub>A</sub> = 25°C unless otherwise specified.

| PARAMETER           | CONDITIONS   | MIN        | TYP        | MAX         | UNITS              |
|---------------------|--|------------|------------|-------------|--------------------|
| f <sub>CLK</sub>    | Clock Frequency<br>V <sub>CC</sub> = 5V, (Note 5)<br>V <sub>CC</sub> = 6V  | 100<br>100 | 640<br>640 | 1280<br>880 | kHz                |
| t <sub>c</sub>      | Conversion Time<br>(Note 6)  | 66         |            | 73          | 1/f <sub>CLK</sub> |
| CR                  | Conversion Rate to First Running State<br>DTR tied to WR with<br>CS = 0 V <sub>DC</sub> , f <sub>CLK</sub> = 640 kHz |            |            | 8770        | conv/s             |
| t <sub>SPUR</sub>   | Width of WR Input (Short Pulse Width)<br>CS = 0 V <sub>DC</sub> (Note 7)   | 100        |            |             | ns                 |
| t <sub>ACC</sub>    | Access Time (Delay from Falling Edge of RD to Output Data Valid)   |            | 135        | 200         | ns                 |
| t <sub>H, TR</sub>  | TRI-STATE Control (Delay from Rising Edge of RD to M-2 State)  |            | 125        | 250         | ns                 |
| t <sub>TR, TR</sub> | Delay from Falling Edge of WR or RD to Return of DTR   |            | 300        | 450         | ns                 |
| C <sub>IN</sub>     | Input Capacitance of Logic Control Inputs  |            | 5          | 7.5         | pF                 |
| C <sub>OUT</sub>    | TRI-STATE Output Capacitance (Data Buffer)   |            | 5          | 7.5         | pF                 |

### Electrical Characteristics

The following specifications apply for  $V_{CC} = 5 \text{ VDC}$  and  $T_{MIN} \leq T_A \leq T_{MAX}$ , unless otherwise specified.

| PARAMETER  | CONDITIONS  | MIN   | TYP   | MAX        | UNITS           |                                    |
|--|---|---|-------|------------|-----------------|------------------------------------|
| <b>CONTROL INPUTS</b> (Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately) |   |   |       |            |                 |                                    |
| $V_{IN(1)}$  | Logical '1' Input Voltage<br>(Except Pin 4 CLK IN)      | $V_{CC} = 5.25 \text{ VDC}$   | 2.0   | 15         | VDC             |                                    |
| $V_{IN(0)}$  | Logical '0' Input Voltage<br>(Except Pin 4 CLK IN)      | $V_{CC} = 4.75 \text{ VDC}$   |       | 0.8        | VDC             |                                    |
| $I_{IN(1)}$  | Logical '1' Input Current<br>(All Inputs)               | $V_{IN} = 5 \text{ VDC}$  | 0.005 | 1          | $\mu\text{ADC}$ |                                    |
| $I_{IN(0)}$  | Logical '0' Input Current<br>(All Inputs)               | $V_{IN} = 0 \text{ VDC}$  | -1    | -0.005     | $\mu\text{ADC}$ |                                    |
| <b>CLOCK IN AND CLOCK R</b>  |   |   |       |            |                 |                                    |
| $V_{T+}$   | CLK IN (Pin 4) Positive Going<br>Threshold Voltage      |   | 2.7   | 3.1        | 3.5             | VDC                                |
| $V_{T-}$   | CLK IN (Pin 4) Negative<br>Going Threshold Voltage      |   | 1.5   | 1.8        | 2.1             | VDC                                |
| $V_H$  | CLK IN (Pin 4) Hysteresis<br>( $V_{T+} - V_{T-}$ )      |   | 0.8   | 1.3        | 2.0             | VDC                                |
| $V_{OUT(0)}$   | Logical '0' CLK R Output<br>Voltage                     | $I_O = 380 \mu\text{A}$<br>$V_{CC} = 4.75 \text{ VDC}$  |       |            | 0.4             | VDC                                |
| $V_{OUT(1)}$   | Logical '1' CLK R Output<br>Voltage                     | $I_O = -350 \mu\text{A}$<br>$V_{CC} = 4.75 \text{ VDC}$   | 2.4   |            |                 | VDC                                |
| <b>DATA OUTPUTS AND ITRR</b>   |   |   |       |            |                 |                                    |
| $V_{OUT(0)}$   | Logical '0' Output Voltage<br>Data Outputs              | $I_{OUT} = 1.5 \text{ mA}, V_{CC} = 4.75 \text{ VDC}$   |       |            | 0.4             | VDC                                |
|  | ITRR Output   | $I_{OUT} = 1.0 \text{ mA}, V_{CC} = 4.75 \text{ VDC}$   |       |            | 0.4             | VDC                                |
| $V_{OUT(1)}$   | Logical '1' Output Voltage                              | $I_O = -380 \mu\text{A}, V_{CC} = 4.75 \text{ VDC}$   | 2.4   |            |                 | VDC                                |
| $V_{OUT(1)}$   | Logical '1' Output Voltage                              | $I_O = -10 \mu\text{A}, V_{CC} = 4.75 \text{ VDC}$  | 4.5   |            |                 | VDC                                |
| $V_{OUT}$  | TRI-STATE Disabled Output<br>Leakage (All Data Buffers) | $V_{OUT} = 0 \text{ VDC}$<br>$V_{OUT} = 5 \text{ VDC}$  | -3    |            | 3               | $\mu\text{ADC}$<br>$\mu\text{ADC}$ |
| $I_{SOURCE}$   |   | $V_{OUT}$ Short to Gnd, $T_A = 25^\circ\text{C}$  | 4.5   | 6          |                 | $\text{mAADC}$                     |
| $I_{SINK}$   |   | $V_{OUT}$ Short to $V_{CC}$ , $T_A = 25^\circ\text{C}$  | 0.0   | 16         |                 | $\text{mAADC}$                     |
| <b>POWER SUPPLY</b>  |   |   |       |            |                 |                                    |
| $I_{CC}$   | Supply Current (Includes<br>Leakage Current)            | $f_{CLK} = 640 \text{ kHz}$ ,<br>$V_{REF/2} = NC$ , $T_A = 25^\circ\text{C}$<br>and $CS = '1'$<br>ADC0801/02/03<br>ADC0804 (Note 9) |       | 1.1<br>1.9 | 1.8<br>2.5      | $\text{mA}$<br>$\text{mA}$         |

Note 1: Absolute maximum ratings are those values beyond which the life of the device may be impaired.

Note 2: All voltages are measured with respect to Gnd, unless otherwise specified. The assembly A Gnd point should always be wired to the D Gnd.

Note 3: A series diode exists, internally, from  $V_{CC}$  to Gnd and has a typical breakdown voltage of 7 VDC.

Note 4: For  $V_{IN(1)} > V_{IH(1)}$  the digital output code will be 0000 0000. Two on-chip diodes are used in each analog input (see back diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the  $V_{CC}$  supply. Be careful, during testing at low  $V_{CC}$  levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures—and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog  $V_{IN}$  does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 VDC to 5 VDC input voltage range will therefore require a minimum supply voltage of 4.950 VDC over temperature variations, initial tolerance and loading.

Note 5: With  $V_{CC} = 5\text{V}$ , the digital logic interfaces are no longer TTL compatible.

Note 6: With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process. The start request is internally latched, see Figure 2 and section 2.0.

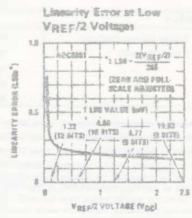
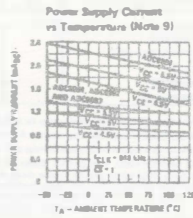
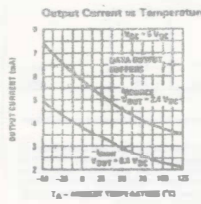
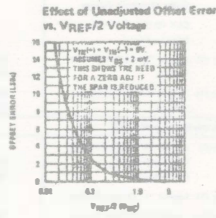
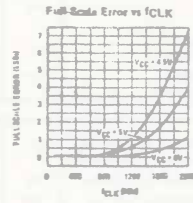
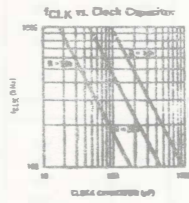
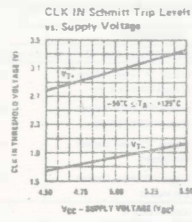
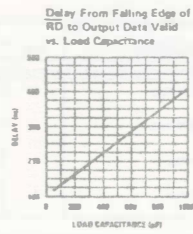
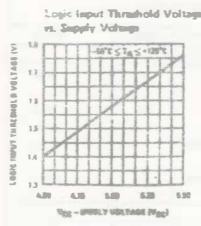
Note 7: The  $CS$  input is assumed to bracket the WR strobe input and therefore timing is dependent on the WR pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the WR pulse (see timing diagrams).

Note 8: None of these A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and Figure 5.

Note 9: For ADC0804LCD typical value of  $V_{REF/2}$  input resistance is 5 k $\Omega$  and of  $I_{CC} < 1.1 \text{ mA}$ .

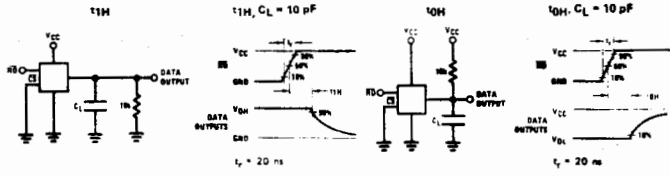


Typical Performance Characteristics

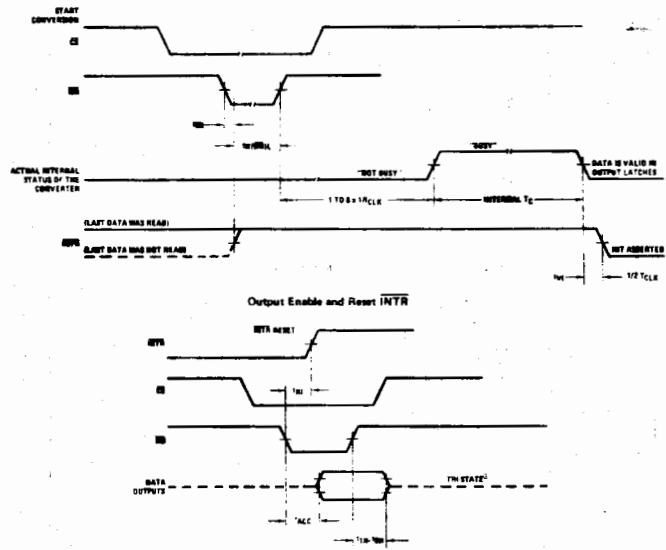


TRI-STATE<sup>®</sup> Test Circuits and Waveforms

235

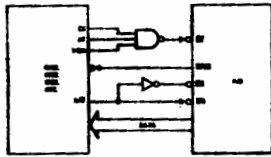


Timing Diagrams

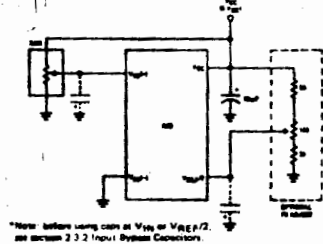


Note: All timing is measured from the 50% voltage points.

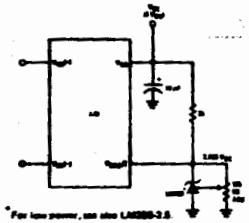
6800 Interface



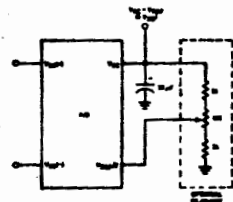
Resistive with Full-Scale Adjust



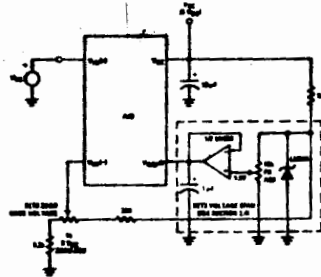
Absolute with a 2.500V Reference



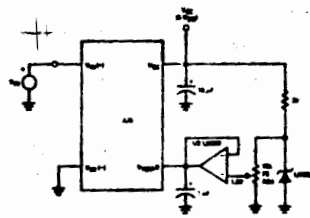
Absolute with a 5V Reference



Zero-Shift and Span Adjust:  $2V \leq V_{ref} \leq 5V$



Zero-Shift and Span Adjust:  $0V \leq V_{ref} \leq 2V$





**matrox**  
 electronic systems Ltd.  
 5800 ANDOVER AVE., T.M.R., QUE., H4T 1H4, CANADA  
 TEL: 514-735-1182 TELEX: 05-925651

# MTX-1632

## MTX TV CRT CONTROLLER FAMILY

### 16x32 VIDEO RANDOM ACCESS MEMORY

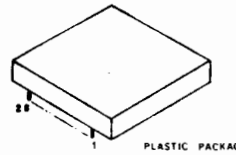
The MTX-1632 is a TVCRT controller designed for use in systems that require display of alphanumeric data. On the input side the device is directly connected to bus organized systems and looks like a 512x8 RAM. The output is a video signal which directly drives a TV monitor to provide a 16x32 field of 512 ASCII characters.

- organized as 512x8 RAM
- 16x32 display field
- no external refresh
- bidirectional data bus
- TTL compatible
- access time < 550 ns
- flicker free display
- single +5V power supply
- low power
- ASCII font standard (7x9)
- standard video output
- drives up to 25 TV monitors
- character blinking
- electronic intensity control

#### ABSOLUTE MAXIMUM RATINGS

|                   |                  |                         |                 |
|-------------------|------------------|-------------------------|-----------------|
| Supply Voltage    | V <sub>CC</sub>  | -0.3 to 6               | V <sub>dc</sub> |
| Input Voltage     | V <sub>I</sub>   | -0.3 to V <sub>CC</sub> | V <sub>dc</sub> |
| Operating Temp.   | T <sub>A</sub>   | 0 to 70                 | °C              |
| Storage Temp.     | T <sub>stg</sub> | -55 to 125              | °C              |
| Lead Solder Temp. | T <sub>sd</sub>  | 400                     | C               |

HYBRID  
 (MOS, TTL)  
 16x32 ASCII Character  
 VIDEO RANDOM ACCESS MEMORY



PLASTIC PACKAGE

#### PIN ASSIGNMENT

|     |      |     |      |
|-----|------|-----|------|
| 1   | GND  | 20  | CS   |
| 2   | D0   | 21  | MD   |
| 3   | D1   | 22  | A1   |
| 4   | D2   | 23  | A2   |
| 5   | D3   | 24  | A3   |
| 6   | D4   | 25  | A4   |
| 7   | D5   | 26  | A5   |
| 8   | D6   | 27  | A6   |
| 9   | D7   | 28  | A7   |
| 10  | D8   | 29  | A8   |
| 11  | D9   | 30  | A9   |
| 12  | D10  | 31  | A10  |
| 13  | D11  | 32  | A11  |
| 14  | D12  | 33  | A12  |
| 15  | D13  | 34  | A13  |
| 16  | D14  | 35  | A14  |
| 17  | D15  | 36  | A15  |
| 18  | D16  | 37  | A16  |
| 19  | D17  | 38  | A17  |
| 20  | D18  | 39  | A18  |
| 21  | D19  | 40  | A19  |
| 22  | D20  | 41  | A20  |
| 23  | D21  | 42  | A21  |
| 24  | D22  | 43  | A22  |
| 25  | D23  | 44  | A23  |
| 26  | D24  | 45  | A24  |
| 27  | D25  | 46  | A25  |
| 28  | D26  | 47  | A26  |
| 29  | D27  | 48  | A27  |
| 30  | D28  | 49  | A28  |
| 31  | D29  | 50  | A29  |
| 32  | D30  | 51  | A30  |
| 33  | D31  | 52  | A31  |
| 34  | D32  | 53  | A32  |
| 35  | D33  | 54  | A33  |
| 36  | D34  | 55  | A34  |
| 37  | D35  | 56  | A35  |
| 38  | D36  | 57  | A36  |
| 39  | D37  | 58  | A37  |
| 40  | D38  | 59  | A38  |
| 41  | D39  | 60  | A39  |
| 42  | D40  | 61  | A40  |
| 43  | D41  | 62  | A41  |
| 44  | D42  | 63  | A42  |
| 45  | D43  | 64  | A43  |
| 46  | D44  | 65  | A44  |
| 47  | D45  | 66  | A45  |
| 48  | D46  | 67  | A46  |
| 49  | D47  | 68  | A47  |
| 50  | D48  | 69  | A48  |
| 51  | D49  | 70  | A49  |
| 52  | D50  | 71  | A50  |
| 53  | D51  | 72  | A51  |
| 54  | D52  | 73  | A52  |
| 55  | D53  | 74  | A53  |
| 56  | D54  | 75  | A54  |
| 57  | D55  | 76  | A55  |
| 58  | D56  | 77  | A56  |
| 59  | D57  | 78  | A57  |
| 60  | D58  | 79  | A58  |
| 61  | D59  | 80  | A59  |
| 62  | D60  | 81  | A60  |
| 63  | D61  | 82  | A61  |
| 64  | D62  | 83  | A62  |
| 65  | D63  | 84  | A63  |
| 66  | D64  | 85  | A64  |
| 67  | D65  | 86  | A65  |
| 68  | D66  | 87  | A66  |
| 69  | D67  | 88  | A67  |
| 70  | D68  | 89  | A68  |
| 71  | D69  | 90  | A69  |
| 72  | D70  | 91  | A70  |
| 73  | D71  | 92  | A71  |
| 74  | D72  | 93  | A72  |
| 75  | D73  | 94  | A73  |
| 76  | D74  | 95  | A74  |
| 77  | D75  | 96  | A75  |
| 78  | D76  | 97  | A76  |
| 79  | D77  | 98  | A77  |
| 80  | D78  | 99  | A78  |
| 81  | D79  | 100 | A79  |
| 82  | D80  | 101 | A80  |
| 83  | D81  | 102 | A81  |
| 84  | D82  | 103 | A82  |
| 85  | D83  | 104 | A83  |
| 86  | D84  | 105 | A84  |
| 87  | D85  | 106 | A85  |
| 88  | D86  | 107 | A86  |
| 89  | D87  | 108 | A87  |
| 90  | D88  | 109 | A88  |
| 91  | D89  | 110 | A89  |
| 92  | D90  | 111 | A90  |
| 93  | D91  | 112 | A91  |
| 94  | D92  | 113 | A92  |
| 95  | D93  | 114 | A93  |
| 96  | D94  | 115 | A94  |
| 97  | D95  | 116 | A95  |
| 98  | D96  | 117 | A96  |
| 99  | D97  | 118 | A97  |
| 100 | D98  | 119 | A98  |
| 101 | D99  | 120 | A99  |
| 102 | D100 | 121 | A100 |
| 103 | D101 | 122 | A101 |
| 104 | D102 | 123 | A102 |
| 105 | D103 | 124 | A103 |
| 106 | D104 | 125 | A104 |
| 107 | D105 | 126 | A105 |
| 108 | D106 | 127 | A106 |
| 109 | D107 | 128 | A107 |
| 110 | D108 | 129 | A108 |
| 111 | D109 | 130 | A109 |
| 112 | D110 | 131 | A110 |
| 113 | D111 | 132 | A111 |
| 114 | D112 | 133 | A112 |
| 115 | D113 | 134 | A113 |
| 116 | D114 | 135 | A114 |
| 117 | D115 | 136 | A115 |
| 118 | D116 | 137 | A116 |
| 119 | D117 | 138 | A117 |
| 120 | D118 | 139 | A118 |
| 121 | D119 | 140 | A119 |
| 122 | D120 | 141 | A120 |
| 123 | D121 | 142 | A121 |
| 124 | D122 | 143 | A122 |
| 125 | D123 | 144 | A123 |
| 126 | D124 | 145 | A124 |
| 127 | D125 | 146 | A125 |
| 128 | D126 | 147 | A126 |
| 129 | D127 | 148 | A127 |
| 130 | D128 | 149 | A128 |
| 131 | D129 | 150 | A129 |
| 132 | D130 | 151 | A130 |
| 133 | D131 | 152 | A131 |
| 134 | D132 | 153 | A132 |
| 135 | D133 | 154 | A133 |
| 136 | D134 | 155 | A134 |
| 137 | D135 | 156 | A135 |
| 138 | D136 | 157 | A136 |
| 139 | D137 | 158 | A137 |
| 140 | D138 | 159 | A138 |
| 141 | D139 | 160 | A139 |
| 142 | D140 | 161 | A140 |
| 143 | D141 | 162 | A141 |
| 144 | D142 | 163 | A142 |
| 145 | D143 | 164 | A143 |
| 146 | D144 | 165 | A144 |
| 147 | D145 | 166 | A145 |
| 148 | D146 | 167 | A146 |
| 149 | D147 | 168 | A147 |
| 150 | D148 | 169 | A148 |
| 151 | D149 | 170 | A149 |
| 152 | D150 | 171 | A150 |
| 153 | D151 | 172 | A151 |
| 154 | D152 | 173 | A152 |
| 155 | D153 | 174 | A153 |
| 156 | D154 | 175 | A154 |
| 157 | D155 | 176 | A155 |
| 158 | D156 | 177 | A156 |
| 159 | D157 | 178 | A157 |
| 160 | D158 | 179 | A158 |
| 161 | D159 | 180 | A159 |
| 162 | D160 | 181 | A160 |
| 163 | D161 | 182 | A161 |
| 164 | D162 | 183 | A162 |
| 165 | D163 | 184 | A163 |
| 166 | D164 | 185 | A164 |
| 167 | D165 | 186 | A165 |
| 168 | D166 | 187 | A166 |
| 169 | D167 | 188 | A167 |
| 170 | D168 | 189 | A168 |
| 171 | D169 | 190 | A169 |
| 172 | D170 | 191 | A170 |
| 173 | D171 | 192 | A171 |
| 174 | D172 | 193 | A172 |
| 175 | D173 | 194 | A173 |
| 176 | D174 | 195 | A174 |
| 177 | D175 | 196 | A175 |
| 178 | D176 | 197 | A176 |
| 179 | D177 | 198 | A177 |
| 180 | D178 | 199 | A178 |
| 181 | D179 | 200 | A179 |
| 182 | D180 | 201 | A180 |
| 183 | D181 | 202 | A181 |
| 184 | D182 | 203 | A182 |
| 185 | D183 | 204 | A183 |
| 186 | D184 | 205 | A184 |
| 187 | D185 | 206 | A185 |
| 188 | D186 | 207 | A186 |
| 189 | D187 | 208 | A187 |
| 190 | D188 | 209 | A188 |
| 191 | D189 | 210 | A189 |
| 192 | D190 | 211 | A190 |
| 193 | D191 | 212 | A191 |
| 194 | D192 | 213 | A192 |
| 195 | D193 | 214 | A193 |
| 196 | D194 | 215 | A194 |
| 197 | D195 | 216 | A195 |
| 198 | D196 | 217 | A196 |
| 199 | D197 | 218 | A197 |
| 200 | D198 | 219 | A198 |
| 201 | D199 | 220 | A199 |
| 202 | D200 | 221 | A200 |
| 203 | D201 | 222 | A201 |
| 204 | D202 | 223 | A202 |
| 205 | D203 | 224 | A203 |
| 206 | D204 | 225 | A204 |
| 207 | D205 | 226 | A205 |
| 208 | D206 | 227 | A206 |
| 209 | D207 | 228 | A207 |
| 210 | D208 | 229 | A208 |
| 211 | D209 | 230 | A209 |
| 212 | D210 | 231 | A210 |
| 213 | D211 | 232 | A211 |
| 214 | D212 | 233 | A212 |
| 215 | D213 | 234 | A213 |
| 216 | D214 | 235 | A214 |
| 217 | D215 | 236 | A215 |
| 218 | D216 | 237 | A216 |
| 219 | D217 | 238 | A217 |
| 220 | D218 | 239 | A218 |
| 221 | D219 | 240 | A219 |
| 222 | D220 | 241 | A220 |
| 223 | D221 | 242 | A221 |
| 224 | D222 | 243 | A222 |
| 225 | D223 | 244 | A223 |
| 226 | D224 | 245 | A224 |
| 227 | D225 | 246 | A225 |
| 228 | D226 | 247 | A226 |
| 229 | D227 | 248 | A227 |
| 230 | D228 | 249 | A228 |
| 231 | D229 | 250 | A229 |
| 232 | D230 | 251 | A230 |
| 233 | D231 | 252 | A231 |
| 234 | D232 | 253 | A232 |
| 235 | D233 | 254 | A233 |
| 236 | D234 | 255 | A234 |
| 237 | D235 | 256 | A235 |
| 238 | D236 | 257 | A236 |
| 239 | D237 | 258 | A237 |
| 240 | D238 | 259 | A238 |
| 241 | D239 | 260 | A239 |
| 242 | D240 | 261 | A240 |
| 243 | D241 | 262 | A241 |
| 244 | D242 | 263 | A242 |
| 245 | D243 | 264 | A243 |
| 246 | D244 | 265 | A244 |
| 247 | D245 | 266 | A245 |
| 248 | D246 | 267 | A246 |
| 249 | D247 | 268 | A247 |
| 250 | D248 | 269 | A248 |
| 251 | D249 | 270 | A249 |
| 252 | D250 | 271 | A250 |
| 253 | D251 | 272 | A251 |
| 254 | D252 | 273 | A252 |
| 255 | D253 | 274 | A253 |
| 256 | D254 | 275 | A254 |
| 257 | D255 | 276 | A255 |
| 258 | D256 | 277 | A256 |
| 259 | D257 | 278 | A257 |
| 260 | D258 | 279 | A258 |
| 261 | D259 | 280 | A259 |
| 262 | D260 | 281 | A260 |
| 263 | D261 | 282 | A261 |
| 264 | D262 | 283 | A262 |
| 265 | D263 | 284 | A263 |
| 266 | D264 | 285 | A264 |
| 267 | D265 | 286 | A265 |
| 268 | D266 | 287 | A266 |
| 269 | D267 | 288 | A267 |
| 270 | D268 | 289 | A268 |
| 271 | D269 | 290 | A269 |
| 272 | D270 | 291 | A270 |
| 273 | D271 | 292 | A271 |
| 274 | D272 | 293 | A272 |
| 275 | D273 | 294 | A273 |
| 276 | D274 | 295 | A274 |
| 277 | D275 | 296 | A275 |
| 278 | D276 | 297 | A276 |
| 279 | D277 | 298 | A277 |
| 280 | D278 | 299 | A278 |
| 281 | D279 | 300 | A279 |
| 282 | D280 | 301 | A280 |
| 283 | D281 | 302 | A281 |
| 284 | D282 | 303 | A282 |
| 285 | D283 | 304 | A283 |
| 286 | D284 | 305 | A284 |
| 287 | D285 | 306 | A285 |
| 288 | D286 | 307 | A286 |
| 289 | D287 | 308 | A287 |
| 290 | D288 | 309 | A288 |
| 291 | D289 | 310 | A289 |
| 292 | D290 | 311 | A290 |
| 293 | D291 | 312 | A291 |
| 294 | D292 | 313 | A292 |
| 295 | D293 | 314 | A293 |
| 296 | D294 | 315 | A294 |
| 297 | D295 | 316 | A295 |
| 298 | D296 | 317 | A296 |
| 299 | D297 | 318 | A297 |
| 300 | D298 | 319 | A298 |
| 301 | D299 | 320 | A299 |
| 302 | D300 | 321 | A300 |
| 303 | D301 | 322 | A301 |
| 304 | D302 | 323 | A302 |

## DC OPERATING CONDITIONS AND CHARACTERISTICS

(Full operating voltage and temperature range unless otherwise noted)

238

### RECOMMENDED DC OPERATING CONDITIONS

| PARAMETER          | SYMBOL   | MIN. | TYP. | MAX.     | UNIT |
|--------------------|----------|------|------|----------|------|
| Supply Voltage     | $V_{CC}$ | 4.75 | 5.0  | 5.25     | Vdc  |
| Input High Voltage | $V_{IH}$ | 2.0  |      | $V_{CC}$ | Vdc  |
| Input Low Voltage  | $V_{IL}$ | 0    |      | 0.7      | Vdc  |

SH (pin 11) is horizontal sync.  
Low = 5.8 us  
High = 58.3 us  
SV (pin 14) is vertical sync.  
Low = .256 ms  
High = 16.41 ms

### DC CHARACTERISTICS

|                            |            |     |     |      |           |
|----------------------------|------------|-----|-----|------|-----------|
| Input Current (High)       | $I_{IH}$   |     |     | 20   | $\mu$ A   |
| Input Current (Low)        | $I_{IL}$   | -   | -   | -0.4 | mA        |
| Output Current (High)      | $I_{OH}$   |     |     | -400 | $\mu$ A   |
| Output Current (Low)       | $I_{OL}$   |     |     | 8    | mA        |
| Output High Voltage        | $V_{OH}$   | 2.4 | 3.4 | -    | Vdc       |
| Output Low Voltage         | $V_{OL}$   | -   | 0.3 | 0.5  | Vdc       |
| Output Leakage (D0-D7)     | $I_{OL,L}$ | -   | -   | 20   | $\mu$ Adc |
| Output Current (High)D0-D7 | $I_{OH}$   | -   | -   | -2.6 | mA        |
| Output Current (Low)D0-D7  | $I_{OL}$   | -   | -   | 24   | mA        |
| Supply Current (-5V)       | $I_{CC}$   | -   | 400 | 600  | mA        |

### CAPACITANCE

| CHARACTERISTIC     | SYMBOL    | MAX. | UNIT |
|--------------------|-----------|------|------|
| Input Capacitance  | $C_{IN}$  | 15   | pF   |
| Output Capacitance | $C_{OUT}$ | 15   | pF   |

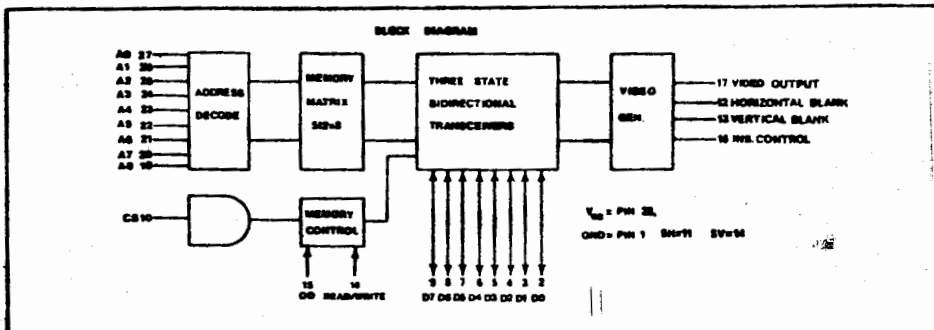
Note 2: All MTX-1632 inputs are TTL equivalent (74LSxxx family or equivalent).

### FUNCTIONAL DESCRIPTION

The MTX-1632 is a TV CRT controller module which generates a video signal that can drive directly any standard TV monitor. The display is 512 ASCII alphanumeric characters arranged in a 16x32 matrix. Characters are displayed white on a black background. Intensity control pin allows electronic control of the brightness of the whole field. It can also be used for blanking of the screen during a read/write operation or for flashing.

On the input side, the MTX-1632 looks like an ordinary 512x8 RAM and can be directly connected to the address and data bus of any bus organized system.

The MTX-1632 is particularly suitable for use in microcomputer systems due to low cost, small size, high speed, low power requirement, and no additional interface circuitry.



AC OPERATING CONDITIONS AND CHARACTERISTICS  
(Full operating voltage and temperature unless otherwise noted)

RECOMMENDED AC OPERATING CONDITIONS

| PARAMETER          | SYMBOL   | MIN. | UNIT |
|--------------------|----------|------|------|
| Address Setup Time | $t_{AS}$ | 0    | ns   |
| Address Hold Time  | $t_{AH}$ | 0    | ns   |

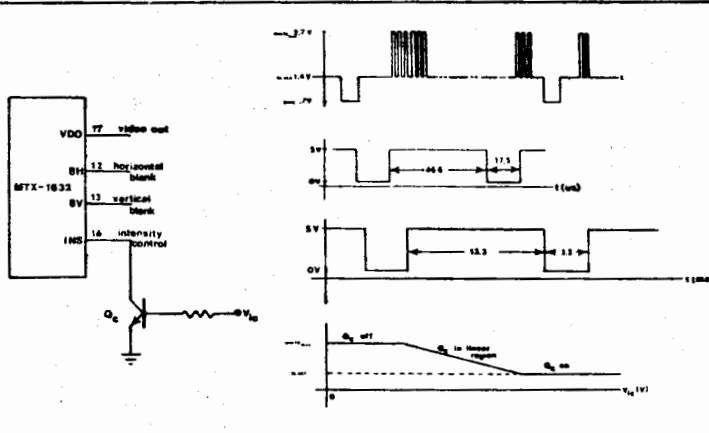
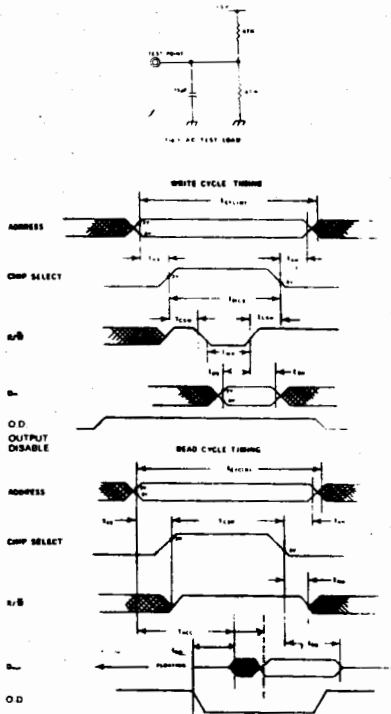
WRITE CYCLE (All timing with  $t_r = t_f = 20ns$ )  
(Load of Fig. 1)

| CHARACTERISTIC            | SYMBOL       | MIN. | MAX. | UNIT |
|---------------------------|--------------|------|------|------|
| Write Cycle Time          | $t_{CYC(W)}$ | 565  | —    | ns   |
| Write Chip Select Width   | $t_{WCS}$    | 565  | —    | ns   |
| Write to Write Setup Time | $t_{CSW}$    | 240  | —    | ns   |
| Write Pulse Width         | $t_{WP}$     | 250  | —    | ns   |
| Write Select Hold Time    | $t_{CSH}$    | 75   | —    | ns   |
| Write Setup Time          | $t_{DS}$     | 250  | —    | ns   |
| Write Hold Time           | $t_{DH}$     | 40   | —    | ns   |

READ CYCLE (all timing with  $t_r = t_f = 20ns$ )  
(Load of Fig. 1)

| CHARACTERISTIC         | SYMBOL       | MIN. | MAX. | UNIT |
|------------------------|--------------|------|------|------|
| Read Cycle Time        | $t_{CYC(R)}$ | 550  | —    | ns   |
| Read Chip Select Width | $t_{CSR}$    | 550  | —    | ns   |
| Read Access Time       | $t_{ACC}$    | —    | 550  | ns   |
| Output Enable Delay    | $t_{ED}$     | —    | 50   | ns   |
| Output Disable Delay   | $t_{DD}$     | —    | 50   | ns   |
| Read Hold Time         | $t_{RH}$     | 0    | —    | ns   |

WRITE PULSE SWITCHES TRANCEIVERS FROM JP TO VRAM (WRITE)  
READ PULSE SWITCHES TRANCEIVERS TO JP FROM VRAM (READ)



Sync frequency  
Horizontal: 15.625 kHz  
Vertical: 60 Hz  
Crystal controlled  
Output impedance 75 ohms  
Short circuit protection built in

BH or BV signal can be used as a flag to access VRAM during horizontal or vertical retrace only (No flicker during access)

Pin 16 open - white max.  
Pin 16 ground - no video (black)  
INS input allows electronic control of the brightness of the whole field

ASCII CHARACTER FONT

MTX-1632 A



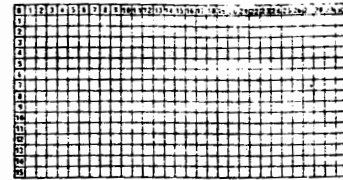
MTX-1632 B



Additional character fonts are available from stock. Custom made font (128 characters in a 7x9 matrix) available at extra cost.

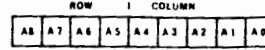
SCREEN DISPLAY ORGANISATION

ROW

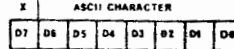


COLUMN

CHARACTER POSITION ADDRESS



DATA



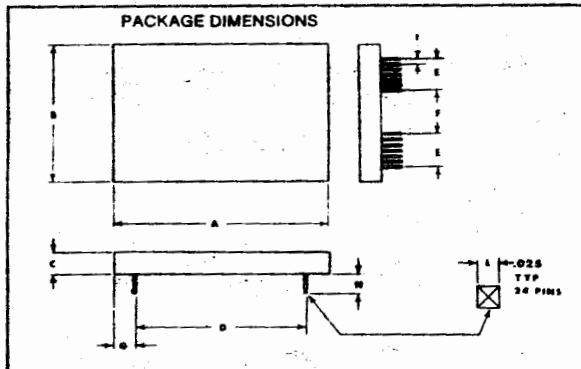
D0-D6 ...character

D7 ...blink bit  
0...normal  
1...character flashing at 1 Hz

NOTE 3:

MODULE CASE IS ISOLATED ALLOWING ETCH RUNS UNDERNEATH.

PACKAGE DIMENSIONS



| DIM | MM      |      | INCHES |      |
|-----|---------|------|--------|------|
|     | NOM.    | TOL. | NOM.   | TOL. |
| A   | 114.300 | .254 | 4.500  | .01  |
| B   | 101.600 | .254 | 4.000  | .01  |
| C   | 12.700  | .254 | .50    | .01  |
| D   | 99.568  | .254 | 3.920  | .01  |
| E   | 15.240  | .254 | .600   | .01  |
| F   | 20.320  | .254 | .800   | .01  |
| G   | 7.366   | .254 | .29    | .01  |
| H   | 4.318   | .254 | .170   | .01  |
| I   | 2.540   | .076 | .100   | .003 |
| L   | .635    | .025 | .025   | .001 |

Matrox Electronic Systems Ltd reserves the right to make changes in specifications at any time and without notice. The information furnished by Matrox Electronic Systems Ltd in this publication is believed to be accurate and reliable. However, no responsibility is assumed by Matrox Electronic Systems Ltd for its use, nor for any infringements of patents or other rights of third parties resulting from its use. No license is granted under any patents or patent rights of Matrox Electronic Systems Ltd.



December 1981

### NS80C48/80C35 P<sup>2</sup>CMOS™ Microcomputer/Microprocessor Family



#### General Description

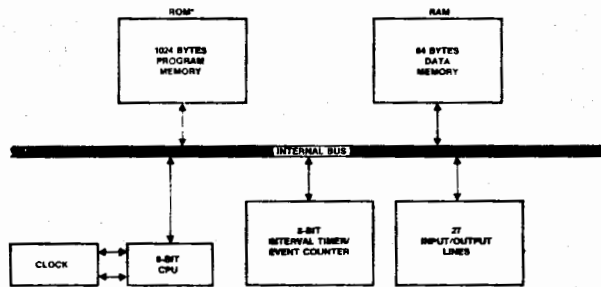
The NS80C48 is a parallel 8-bit microcomputer contained in a standard 40-pin, dual-in-line package. The device is fabricated using P<sup>2</sup>CMOS silicon gate technology. This technology provides the system designer with devices that equal the speed performance levels of comparable NMOS products, combined with low-power advantages of CMOS.

The NS80C48 is a stand-alone microcomputer\* designed for efficient controller applications. It executes powerful bit manipulative instructions and BCD as well as binary arithmetic. The NS80C48 contains on-chip oscillator and clock circuits, 1K x 8 ROM program memory, 64 x 8 RAM data memory, 27 I/O lines, and an 8-bit Timer/Counter. Also, it is pin and instruction compatible with the XMOS™ INS8048.

#### Features

- 8-Bit CPU, RAM, ROM, I/O in a Single Package
- 2.5 μsec Cycle Time, 6MHz Oscillator
- Low Power
- Very Low Stand-by Power
- Expandable Memory and I/O
- Single-Level Interrupt
- Efficient Instructions
- Instruction Compatible to INS8048
- Pin Compatible to INS8048

#### NS80C48/NS80C35 Block Diagram



\*Not Applicable to INS80C35.

TM-STATE is a registered trademark. XMOS and P<sup>2</sup>CMOS are trademarks of National Semiconductor Corporation.

NS80C48/80C35 P<sup>2</sup>CMOS™ Microcomputer/Microprocessor Family



### Absolute Maximum Ratings

|   |  |
|---|--|
| Voltage at any pin                              | V <sub>DD</sub> - 2.0V to V <sub>SS</sub> - 0.3V |
| Operating Temperature                           | See Operating Range below                        |
| Storage Temperature                             | -65°C to +150°C                                  |
| Operating V <sub>DD</sub> V <sub>CC</sub> Range | 4.5 to 5.5V                                      |
| Lead Temperature (Soldering, 10 seconds)        | 300°C  |

NOTE: Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.

### Operating Range

| Component Type | Ambient Temperature | V <sub>CC</sub>  | Part Number                       |
|----------------|---------------------|------------------|-----------------------------------|
| Industrial     | -40°C to +85°C      | + 4.5V to - 5.5V | NS80C48XXX:NI NS80C35NI (ROMless) |
| Commercial     | 0°C to +70°C        | + 4.5V to + 5.5V | NS80C48XXX:N NS80C35N (ROMless)   |

### DC Electrical Characteristics — NS80C48 and NS80C35

V<sub>CC</sub> = V<sub>DD</sub> = +5V ± 10%, V<sub>SS</sub> = 0V, unless otherwise specified. The following values are valid over the 0°C to 70°C temperature range or -40°C to +85°C for the industrial part.

| Symbol           | Parameter   | Min                         | Typ    | Max      | Units | Test Conditions  |
|------------------|---|-----------------------------|--------|----------|-------|--|
| V <sub>IL</sub>  | Input Low Voltage   |                             |        | 1.5      | V     |  |
| V <sub>IH</sub>  | Input High Voltage  | 3.5                         |        |          | V     |  |
| V <sub>OL</sub>  | Output Low Voltage  |                             |        | 45       | V     | I <sub>SENK</sub> ≤ 2 mA   |
| V <sub>OH</sub>  | Output High Voltage   | 2.4<br>V <sub>CC</sub> -0.5 |        |          | V     | I <sub>SOURCE</sub> ≤ 100 μA<br>I <sub>SOURCE</sub> ≤ 10 μA  |
| I <sub>L</sub>   | Input Leakage Current<br>(T1, EA, BT)<br>All Others                 |                             |        | 10<br>10 | μA    | V <sub>IH</sub> = V <sub>CC</sub> = V <sub>DD</sub><br>V <sub>DD</sub> ≥ V <sub>IH</sub> ≥ V <sub>SS</sub> |
| I <sub>OL</sub>  | Output Leakage Current<br>(High Impedance State)<br>Bus Outputs, T0 |                             |        | 10       | μA    | V <sub>DD</sub> ≥ V <sub>IH</sub> ≥ V <sub>SS</sub>  |
| I <sub>DD</sub>  | Operating Supply Current  |                             | 5<br>1 | 10<br>5  | mA    | f <sub>C</sub> = 6 MHz<br>f <sub>C</sub> = 1 MHz<br>(All Outputs Open)                                     |
| I <sub>DDO</sub> | Quiescent Current<br>(Stand-by)                                     |                             | 20     |          | μA    | XTAL1 = RESET = V <sub>SS</sub><br>V <sub>CC</sub> = 0V, Outputs open                                      |

**AC Electrical Characteristics — NS80C48 and NS80C35**

V<sub>CC</sub> = V<sub>CC</sub> = -5V ±10%, V<sub>SS</sub> = 0V, unless otherwise specified

| Symbol           | Parameter                           | Min | Typ | Max | Units | Test Conditions         |
|------------------|-------------------------------------|-----|-----|-----|-------|-------------------------|
| t <sub>AA</sub>  | ALE Pulse Width                     | 400 |     |     | ns    |                         |
| t <sub>AS</sub>  | Address Setup to ALE                | 150 |     |     | ns    |                         |
| t <sub>AH</sub>  | Address Hold from ALE               | 80  |     |     | ns    |                         |
| t <sub>CC</sub>  | Control Pulse Width<br>PSEN, RD, WR | 700 |     |     | ns    |                         |
| t <sub>DS</sub>  | Data Set-Up Before WR               | 500 |     |     | ns    |                         |
| t <sub>DS</sub>  | Data Hold After WR                  | 80  |     |     | ns    | C <sub>L</sub> = 20pF   |
| t <sub>CT</sub>  | Cycle Time                          | 2.5 |     |     | μs    | f <sub>C</sub> = 6 MHz  |
|                  |                                     | 15  |     |     | μs    | f <sub>C</sub> = 1MHz   |
|                  |                                     | 30  |     |     | μs    | f <sub>C</sub> = 500KHz |
| t <sub>DR</sub>  | Data Hold                           | 0   |     | 200 | ns    |                         |
| t <sub>DP</sub>  | PSEN RD to Data In                  |     |     | 500 | ns    |                         |
| t <sub>AW</sub>  | Address Setup to WR                 | 230 |     |     | ns    |                         |
| t <sub>AP</sub>  | Address Setup to Data In            |     |     | 950 | ns    |                         |
| t <sub>ARC</sub> | Address Float to RD, PSEN           | 0   |     |     | ns    |                         |
| t <sub>CA</sub>  | Control Pulse to ALE                | 10  |     |     | ns    |                         |

**Port 2 Timing**

| Symbol          | Parameter   | Min  | Typ | Max | Units | Test Conditions |
|-----------------|---|------|-----|-----|-------|-----------------|
| t <sub>CP</sub> | Port Control Setup before<br>Falling Edge of PROG | 110  |     |     | ns    |                 |
| t <sub>CH</sub> | Port Control Hold after<br>Falling Edge of PROG   | 140  |     |     | ns    |                 |
| t <sub>PI</sub> | PROG to P2 Input valid                            | 220  |     | 310 | ns    |                 |
| t <sub>PO</sub> | Output Data Setup Time                            | 250  |     |     | ns    |                 |
| t <sub>PH</sub> | Output Data Hold Time                             | 65   |     |     | ns    |                 |
| t <sub>SI</sub> | Input Data Hold Time                              | 0    |     | 150 | ns    |                 |
| t <sub>PP</sub> | PROG Pulse Width                                  | 1510 |     |     | ns    |                 |
| t <sub>PL</sub> | Port 2 I/O Data Setup                             | 400  |     |     | ns    |                 |
| t <sub>PH</sub> | Port 2 I/O Data Hold                              | 150  |     |     | ns    |                 |

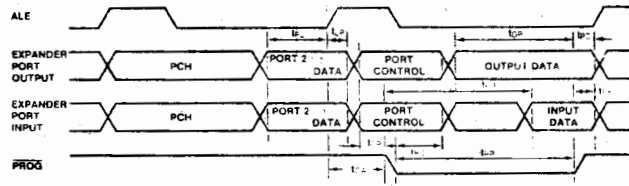
**Capacitance**

T<sub>A</sub> = 25° C, V<sub>CC</sub> = V<sub>SS</sub> = 0V

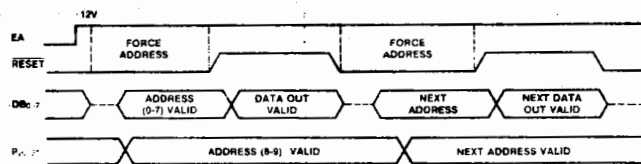
| Symbol           | Parameter                               | Min | Typ | Max | Units | Test Conditions                                |
|------------------|---|-----|-----|-----|-------|--|
| C <sub>IN</sub>  | Input Capacitance<br>(All Except RESET) |     | 6   |     | pF    | f <sub>C</sub> = 1 MHz                         |
| C <sub>OUT</sub> | OUTPUT and RESET<br>Capacitance         |     | 10  |     | pF    | unmeasured pins<br>returned to V <sub>SS</sub> |



## Timing Waveforms - Continued



Port 2 Timing



Verity Mode Timing

## Functional Pin Description

## INPUT SIGNALS

**Reset (RESET):** An active low (0) input that initializes the processor and is used to verify program memory.

**Single Step (SS):** Active low (0) input which, in conjunction with ALE, can single step the processor through each instruction.

**External Access (EA):** An active high (1) input that forces all program memory fetches to reference external program memory.

**Testable Input 0 (T<sub>0</sub>):** Testable input pin using conditional branch functions JT<sub>0</sub> (T<sub>0</sub> = 1) or JNT<sub>0</sub> (T<sub>0</sub> = 0). T<sub>0</sub> can be designated as the clock output using instruction ENTJ CLK.

**Testable Input 1 (T<sub>1</sub>):** Testable input pin using conditional branch functions JT<sub>1</sub> (T<sub>1</sub> = 1) or JNT<sub>1</sub> (T<sub>1</sub> = 0). T<sub>1</sub> can be designated as the Timer Counter input from an external source using instruction STRT CNT.

**Interrupt (INT):** An active low (0) input that initiates an interrupt when interrupt is enabled. Interrupt is disabled after a reset. Also can be tested with instruction JNI (INT = 0).

## OUTPUT SIGNALS

**Read Strobe (RD):** An active low (0) output strobe activated during a Bus read. Can be used to enable data from an external device onto the bus. Used as a Read Strobe to External Data Memory.

**Write Strobe (WR):** An active low (0) output strobe activated during a Bus write. Used as a Write Strobe to External Data Memory.

**Program Store Enable (PSEN):** An active low (0) output that occurs only during an external program memory fetch.

**Address Latch Enable (ALE):** An active high (1) output that occurs once during each cycle and is used as a data output. The negative going edge of ALE strobes the address into external data or program memory.

**Program (PROG):** This active low (0) output provides the strobe for INS8243 I/O Expander.

**INPUT OUTPUT SIGNALS**

**Crystal Input (XTAL1, XTAL2):** These two pins connect the crystal for internal oscillator operation. XTAL1 is the timing input for external source.

**Port 1 (P10-P17):** 8-bit quasi-bidirectional port.

**Port 2 (P20-P27):** 8-bit quasi-bidirectional port. During an external program memory fetch, the four high-order program counter bits occur at P20-P23. They also serve as 4-bit I/O expander bus when the INS8243 I/O Expander is used.

**BUS (DB0-DB7):** True bidirectional port, either statically latched or synchronous. Can be used to write to an external device using WR Strobe, or Read from using RD Strobe. During an external program memory fetch, the 8 lower order program counter bits are present at this port. The addressed instruction appears on this bus when PSEN is low. During an external RAM data transaction this port presents address and data under control of ALE, RD, and WR.

**Vss:** Processor ground potential.

**Vdd:** Primary Processor Power Supply. During standby operation Vdd may vary from 4.5V to 5.5V.

**Vcc/DILE:** A logic input. When the input is a logic 0 (low) the device is in the standby mode, when the input is a logic 1 (high) the device is in the run mode.

**Functional Description**

The following paragraphs contain the functional description of the major elements of the 48-Series microcomputer/microprocessor. Figure 1 is a block diagram of the 48-Series devices. The data paths are illustrated in simplified form to show how the various logic elements communicate with each other to implement the instruction set common to 8048 family devices.

**Program Memory**

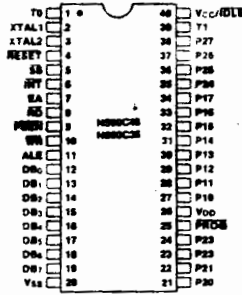
The Program Memory (ROM) contained on the NS80C48 device is comprised of 1024, 8-bit bytes.

As is seen by examining the 48-Series instruction set, these bytes may be program instructions, program data or ROM addressing data. The ROM for the above devices must be mask programmed at the National Semiconductor factory. The ROMless microcomputer NS80C35 uses external program memory. This makes program development straightforward using standard UV erasable ROMs to emulate a possible future single chip (using the on-board ROM) system. ROM addressing up to a maximum of 4K is accomplished by a 12-bit Program Counter (PC). The NS80C48 will automatically address external memory when its internal 1K byte memory is exceeded. The binary value of the address selects one of the 8-bit bytes contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential binary count value.

With reference to the Program Memory Map (see Figure 2) there are three ROM addresses which provide for the control of the microcomputer.

1. Memory Location 0000 - Asserting the Reset (negative true) input to the microcomputer forces the first instruction to be fetched from address 0000.
2. Memory Location 0003 - Asserting the Interrupt (negative true) input to the microcomputer (when interrupt is enabled) forces a jump to subroutine.
3. Memory Location 0007 - A timer/counter interrupt that results from timer/counter overflow (when enabled) forces a jump to subroutine.

**Pin Configuration**



00-1711



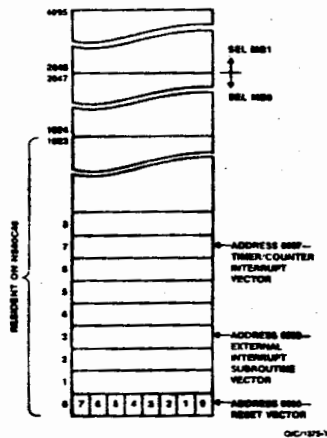


FIGURE 2. NS80C48 Resident ROM Program Memory Map

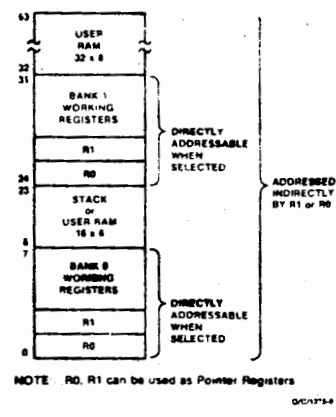


FIGURE 3. NS80C48 Resident RAM Data Memory Map

**Data Memory (RAM)**

The resident RAM data memory is arranged as 64 bytes. RAM addressing is implemented indirectly via either of two 8-bit RAM pointer registers R0 and R1. These pointer registers are essentially the first two locations in the RAM (see Figure 3), addresses 00 and 01. RAM addressing may also be performed directly by 11 direct register instructions. The register area of the RAM array is made up of eight working registers that occupy either the first bank (0), locations 0-7, or the second bank (1), locations 24-31. The second bank of working registers is selected by using the Register Bank Switch instruction (SEL RB). If this bank is not used for working registers, it can be used as user RAM.

There is an 8-level stack after Bank 0 that occupies address locations 8 to 23. These RAM locations are addressed indirectly through R0, R1 or the 3-bit Stack Pointer (SP). The stack pointer keeps track of the return address and pushes each return address down into the stack. There are 8 levels of subroutine nesting possible in the stack because each address occupies 12 bits of two consecutive bytes in RAM. When the level of subroutine nesting is less than 8, the stack locations not used may be utilized as user RAM locations.

**Input/Output**

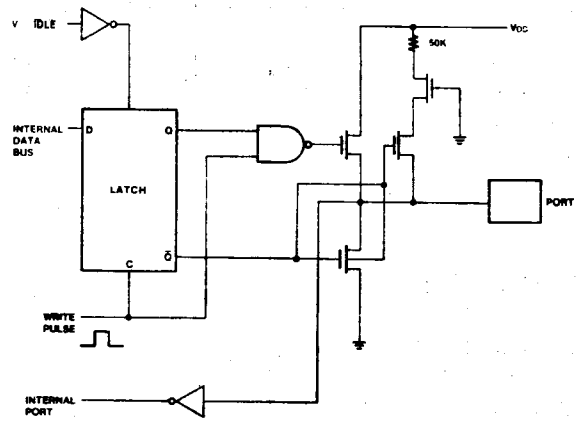
The 48-Series devices have 27 lines of input/output organized as three, 8-bit ports plus three test inputs. The three ports may be used as inputs, outputs or bidirectional ports. Ports 1 and 2 differ from port 0 (Bus Port) in that they are quasi-bidirectional ports. Ports 1 and 2 can be used as input and output while being statically latched. If more I/O lines are required, Port 2 can also serve as a 4-bit I/O bus expander when used in conjunction with the INS8243 I/O Expander. See Figure 4.

The bus port is a true bidirectional port and is either statically latched or synchronous. It can be written to using WR strobe or read from using RD strobe. During an external program memory fetch, the 8 lower order program counter bits are present at this port. The addressed instruction appears on this bus when PSEN is low. During an external RAM data transaction, this port presents address and data under control of ALE, RD, and WR. See Figure 5.

**INSTRUCTION SET**

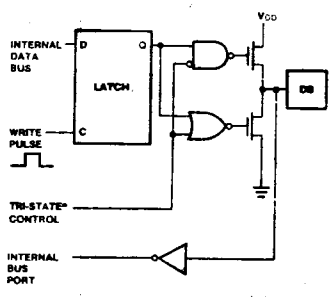
Both the NS80C48 and the NS80C35 have the same instruction set as the X808™-NS8048.

Table 1 provides the mnemonic, function, description, instruction code, and number of cycles. And where applicable, the flag settings.



O.C. 1275-11

FIGURE 4. Equivalent I/O Configuration for Port 1 and 2



O.C. 1275-12

FIGURE 5. Equivalent Bus Port Configuration

TRI-STATE is a registered trademark of National Semiconductor Corporation



## Instruction Set

Table 1 details the 56 instructions common to the entire 48 Series of processors. The table provides the mnemonic, function and description, instruction code, number of cycles and, where applicable, flag settings.

Table 1 Instruction Set

| MNEMONIC       | FUNCTION  | DESCRIPTION   | CYCLES | BYTES | FLAGS |    |    |    |    |
|----------------|---|---|--------|-------|-------|----|----|----|----|
|                |   |   |        |       | C     | AC | FO | FI | PS |
| CONTROL        |   |   |        |       |       |    |    |    |    |
| EN I           |   | Enable the External Interrupt Input   | 1      | 1     |       |    |    |    |    |
| DIS I          |   | Disable the External Interrupt Input  | 1      | 1     |       |    |    |    |    |
| EN TC CLK      |   | Enable TD as the Clock Output   | 1      | 1     |       |    |    |    |    |
| SEL MB0        | (DBF) - 0   | Select Bank 0 (locations 0 - 2047) of Program Memory  | 1      | 1     |       |    |    |    |    |
| SEL MB1        | (DBF) - 1   | Select Bank 1 (locations 2048 - 4095) of Program Memory   | 1      | 1     |       |    |    |    |    |
| SEL RB0        | (BS) - 0  | Select Bank 0 (locations 0 - 7) of Data Memory  | 1      | 1     |       |    |    |    |    |
| SEL RB1        | (BS) - 1  | Select Bank 1 (locations 24 - 31) of Data Memory  | 1      | 1     |       |    |    |    |    |
| DATA MOVES     |   |   |        |       |       |    |    |    |    |
| MOV A, #data   | (A) - data  | Move immediate the specified data into the Accumulator  | 2      | 2     |       |    |    |    |    |
| MOV A, Rr      | (A) - (Rr); r = 0 - 7                               | Move the contents of the designated registers into the Accumulator  | 1      | 1     |       |    |    |    |    |
| MOV A, @Rr     | (A) - @Rr; r = 0 - 1                                | Move indirect the contents of data memory location into the Accumulator   | 1      | 1     |       |    |    |    |    |
| MOV A, PSW     | (A) - (PSW)   | Move contents of the Program Status Word into the Accumulator   | 1      | 1     |       |    |    |    |    |
| MOV Rr, #data  | (Rr) - data; r = 0 - 7                              | Move immediate the specified data into the designated register  | 2      | 2     |       |    |    |    |    |
| MOV Rr, A      | (Rr) - (A); r = 0 - 7                               | Move Accumulator contents into the designated register  | 1      | 1     |       |    |    |    |    |
| MOV @Rr, A     | @Rr - (A); r = 0 - 1                                | Move indirect Accumulator contents into data memory location  | 1      | 1     |       |    |    |    |    |
| MOV @Rr, #data | @Rr - data; r = 0 - 1                               | Move immediate the specified data into data memory  | 2      | 2     |       |    |    |    |    |
| MOV PSW, A     | (PSW) - (A)   | Move contents of Accumulator into the Program Status Word   | 1      | 1     | *     | *  | *  | *  | *  |
| MOV P, A @ A   | (PC 0 - 7) - (A);<br>(A) - @PC                      | Move the content of program memory location in the current page addressed by the contents of Accumulator into the Accumulator | 2      | 1     |       |    |    |    |    |
| MOV P3, A @ A  | (PC 0 - 7) - (A);<br>(PC 8 - 15) - 011<br>(A) - @PC | Move the content of program memory location in page 3 address by the content of Accumulator into the Accumulator              | 2      | 1     |       |    |    |    |    |
| MOVX A, @R     | (A) - @R; r = 0 - 1                                 | Move indirect the contents of external data memory into the Accumulator   | 2      | 1     |       |    |    |    |    |
| MOVX @R, A     | (R) - (A); r = 0 - 1                                | Move indirect the contents of the Accumulator into external data memory   | 2      | 1     |       |    |    |    |    |
| XCH A, Rr      | (A) - (Rr); r = 0 - 7                               | Exchange the Accumulator and designated register's contents   | 1      | 1     |       |    |    |    |    |
| XCH A, @Rr     | (A) - @Rr; r = 0 - 1                                | Exchange indirect contents of Accumulator and location in data memory   | 1      | 1     |       |    |    |    |    |
| XCHD A, @Rr    | (A0 - A3) - @Rr; R = 0; r = 0 - 1                   | Exchange indirect 4-bit contents of Accumulator and data memory   | 1      | 1     |       |    |    |    |    |

Table 1 Instruction Set - Continued

| MNEMONIC                  | FUNCTION   | DESCRIPTION  | CYCLES | BYTES | FLAGS |    |    |    |  |
|---------------------------|--|--|--------|-------|-------|----|----|----|--|
|                           |  |  |        |       | C     | AC | PS | FT |  |
| <b>TIMER COUNTER</b>      |  |  |        |       |       |    |    |    |  |
| EN TCNT                   |  | Enable Internal Interrupt Flag for Timer Counter output.                         | 1      | 1     |       |    |    |    |  |
| DIS TCNT                  |  | Disable Internal Interrupt Flag for Timer Counter output.                        | 1      | 1     |       |    |    |    |  |
| MOV A, T                  | (A) ← (T)  | Move contents of Timer Counter into Accumulator.                                 | 1      | 1     |       |    |    |    |  |
| MOV T, A                  | (T) ← (A)  | Move contents of Accumulator into Timer Counter.                                 | 1      | 1     |       |    |    |    |  |
| STOP TCNT                 |  | Stop Count for Event Counter.  | 1      | 1     |       |    |    |    |  |
| START CNT                 |  | Start Count for Event Counter.   | 1      | 1     |       |    |    |    |  |
| START T                   |  | Start Count for Timer.   | 1      | 1     |       |    |    |    |  |
| <b>ACCUMULATOR</b>        |  |  |        |       |       |    |    |    |  |
| ADD A, #data              | (A) ← (A) + data                                       | Add immediate specified data to the Accumulator.                                 | 2      | 2     | *     | *  |    |    |  |
| ADD A, R <sub>i</sub>     | (A) ← (A) + (R <sub>i</sub> )<br>for i = 0 - 7         | Add contents of designated register to the Accumulator.                          | 1      | 1     | *     | *  |    |    |  |
| ADD A, [R <sub>i</sub> ]  | (A) ← (A) + (R <sub>i</sub> )<br>for i = 0 - 1         | Add indirect the contents of data memory location of the Accumulator.            | 1      | 1     | *     | *  |    |    |  |
| ADDC A, #data             | (A) ← (A) + (C) + data                                 | ADD immediate with carry the specified data to the Accumulator.                  | 2      | 2     | *     | *  |    |    |  |
| ADDC A, R <sub>i</sub>    | (A) ← (A) + (C) + (R <sub>i</sub> )<br>for i = 0 - 7   | ADD with carry the contents of the designated register to the Accumulator.       | 1      | 1     | *     | *  |    |    |  |
| ADDC A, [R <sub>i</sub> ] | (A) ← (A) + (C) + (R <sub>i</sub> )<br>for i = 0 - 1   | ADD indirect with carry the contents of data memory location to the Accumulator. | 1      | 1     | *     | *  |    |    |  |
| AND A, #data              | (A) ← (A) AND data                                     | Logical AND specified immediate data with Accumulator.                           | 2      | 2     |       |    |    |    |  |
| AND A, R <sub>i</sub>     | (A) ← (A) AND (R <sub>i</sub> )<br>for i = 0 - 7       | Logical AND contents of designated register with Accumulator.                    | 1      | 1     |       |    |    |    |  |
| AND A, [R <sub>i</sub> ]  | (A) ← (A) AND (R <sub>i</sub> )<br>for i = 0 - 1       | Logical AND indirect the contents of data memory with Accumulator.               | 1      | 1     |       |    |    |    |  |
| CPL A                     | (A) ← NOT (A)  | Complement the contents of the Accumulator.                                      | 1      | 1     |       |    |    |    |  |
| CLR A                     | (A) ← 0  | CLEAR the contents of the Accumulator.   | 1      | 1     |       |    |    |    |  |
| DA A                      |  | DECIMAL ADJUST the contents of the Accumulator.                                  | 1      | 1     | *     |    |    |    |  |
| DEC A                     | (A) ← (A) - 1  | DECREMENT by 1 the Accumulator's contents.                                       | 1      | 1     |       |    |    |    |  |
| INC A                     | (A) ← (A) + 1  | INCREMENT by 1 the Accumulator's contents.                                       | 1      | 1     |       |    |    |    |  |
| ORL A, #data              | (A) ← (A) OR data                                      | Logical OR specified immediate data with Accumulator.                            | 2      | 2     |       |    |    |    |  |
| ORL A, R <sub>i</sub>     | (A) ← (A) OR (R <sub>i</sub> )<br>for i = 0 - 7        | Logical OR contents of designated register with Accumulator.                     | 1      | 1     |       |    |    |    |  |
| ORL A, [R <sub>i</sub> ]  | (A) ← (A) OR (R <sub>i</sub> )<br>for i = 0 - 1        | Logical OR indirect the contents of data memory with Accumulator.                | 1      | 1     |       |    |    |    |  |
| RLA                       | (An + 1) ← (An)<br>for n = 0 - 6<br>(A0) ← (A7)        | Rotate Accumulator left by 1-bit without carry.                                  | 1      | 1     |       |    |    |    |  |
| RLC A                     | (An + 1) ← (An), n = 0 - 6<br>(A0) ← (C)<br>(C) ← (A7) | Rotate Accumulator left by 1-bit through carry.                                  | 1      | 1     | *     |    |    |    |  |
| RRA                       | (An) ← (An + 1), n = 0 - 6<br>(A7) ← (A0)              | Rotate Accumulator right by 1-bit without carry.                                 | 1      | 1     |       |    |    |    |  |

Table 1 Instruction Set - Continued

| MNEMONIC                       | FUNCTION   | DESCRIPTION  | CYCLES | BYTES | FLAGS |    |    |    |  |  |  |
|--------------------------------|--|--|--------|-------|-------|----|----|----|--|--|--|
|                                |  |  |        |       | C     | AC | FD | F1 |  |  |  |
| <b>ACCUMULATOR (Continued)</b> |  |  |        |       |       |    |    |    |  |  |  |
| RRC A                          | $(A)_n \rightarrow (A)_{n+1}$ ; $n = 0 - 6$<br>(A)7 ← (C)<br>(C) ← (A0)              | Rotate Accumulator right by 1 bit through carry                                  | 1      | 1     |       |    |    |    |  |  |  |
| SLAP A                         | $(A4 - A7) \rightarrow (A0 - A3)$  | Swap the 2 4-bit nibbles in the Accumulator                                      | 1      | 1     |       |    |    |    |  |  |  |
| XRL A, #data                   | (A) ← (A) XOR data   | Logical XOR immediate specified data with Accumulator                            | 2      | 2     |       |    |    |    |  |  |  |
| XRL A, Rr                      | (A) ← (A) XOR (Rr)<br>for $r = 0 - 7$  | Logical XOR contents of designated register with Accumulator                     | 1      | 1     |       |    |    |    |  |  |  |
| XRL A, @Rr                     | (A) ← (A) XOR (Rr)<br>for $r = 0 - 7$  | Logical XOR indirect the contents of data memory location with Accumulator       | 1      | 1     |       |    |    |    |  |  |  |
| <b>BRANCH</b>                  |  |  |        |       |       |    |    |    |  |  |  |
| DJNZ Rr, addr                  | $(Rr) \rightarrow (Rr) - 1$ ; $r = 0 - 7$<br>if (Rr) = 0,<br>(PC) ← (PC) + 2 if addr | Decrement the specified register and test contents.                              | 2      | 2     |       |    |    |    |  |  |  |
| J0 addr                        | (PC) ← (PC) + 2 if S0 = 1<br>(PC) ← (PC) + 2 if S5 = 0                               | Jump to specified address if Accumulator bit is set                              | 2      | 2     |       |    |    |    |  |  |  |
| JC addr                        | (PC) ← (PC) + 2 if C = 1<br>(PC) ← (PC) + 2 if C = 0                                 | Jump to specified address if carry flag is set                                   | 2      | 2     |       |    |    |    |  |  |  |
| JFD addr                       | (PC) ← (PC) + 2 if FD = 1<br>(PC) ← (PC) + 2 if FD = 0                               | Jump to specified address if Flag FD is set                                      | 2      | 2     |       |    |    |    |  |  |  |
| JF1 addr                       | (PC) ← (PC) + 2 if F1 = 1<br>(PC) ← (PC) + 2 if F1 = 0                               | Jump to specified address if Flag F1 is set                                      | 2      | 2     |       |    |    |    |  |  |  |
| JMP addr                       | (PC) ← (PC) + 2 if D = 1<br>(PC) ← (PC) + 2 if D = 0<br>(PC) ← (PC) + 2 if D = 0     | Direct jump to specified address with the 2K address block                       | 2      | 2     |       |    |    |    |  |  |  |
| JMPP @A                        | (PC) ← (A)   | Jump indirect to specified address pointed to by the Accumulator in current page | 2      | 1     |       |    |    |    |  |  |  |
| JNC addr                       | (PC) ← (PC) + 2 if C = 0<br>(PC) ← (PC) + 2 if C = 1                                 | Jump to specified address if carry flag is low                                   | 2      | 2     |       |    |    |    |  |  |  |
| JNI addr                       | (PC) ← (PC) + 2 if I = 0<br>(PC) ← (PC) + 2 if I = 1                                 | Jump to specified address if interrupt is low                                    | 2      | 2     |       |    |    |    |  |  |  |
| JNT0 addr                      | (PC) ← (PC) + 2 if T0 = 0<br>(PC) ← (PC) + 2 if T0 = 1                               | Jump to specified address if Test 0 is low                                       | 2      | 2     |       |    |    |    |  |  |  |
| JNT1 addr                      | (PC) ← (PC) + 2 if T1 = 0<br>(PC) ← (PC) + 2 if T1 = 1                               | Jump to specified address if Test 1 is low                                       | 2      | 2     |       |    |    |    |  |  |  |
| JNZ addr                       | (PC) ← (PC) + 2 if A = 0<br>(PC) ← (PC) + 2 if A = 1                                 | Jump to specified address if Accumulator is non-zero                             | 2      | 2     |       |    |    |    |  |  |  |
| JTF addr                       | (PC) ← (PC) + 2 if TF = 1<br>(PC) ← (PC) + 2 if TF = 0                               | Jump to specified address if Timer Flag is set to 1                              | 2      | 2     |       |    |    |    |  |  |  |
| JT0 addr                       | (PC) ← (PC) + 2 if T0 = 1<br>(PC) ← (PC) + 2 if T0 = 0                               | Jump to specified address if Test 0 is a 1                                       | 2      | 2     |       |    |    |    |  |  |  |
| JT1 addr                       | (PC) ← (PC) + 2 if T1 = 1<br>(PC) ← (PC) + 2 if T1 = 0                               | Jump to specified address if Test 1 is a 1                                       | 2      | 2     |       |    |    |    |  |  |  |
| JZ addr                        | (PC) ← (PC) + 2 if A = 0<br>(PC) ← (PC) + 2 if A = 1                                 | Jump to specified address if Accumulator is 0                                    | 2      | 2     |       |    |    |    |  |  |  |
| <b>INPUT/OUTPUT</b>            |  |  |        |       |       |    |    |    |  |  |  |
| ANL BUS, #data                 | (BUS) ← (BUS) AND data   | Logical AND immediate specified data with contents of BUS                        | 2      | 2     |       |    |    |    |  |  |  |
| ANL Pp, #data                  | (Pp) ← (Pp) AND data<br>p = 1 - 2  | Logical AND immediate specified data with designated port (1 or 2)               | 2      | 2     |       |    |    |    |  |  |  |

Table 1 Instruction Set - Continued

| MNEMONIC                        | FUNCTION   | DESCRIPTION   | CYCLES | BYTES | FLAGS |   |    |    |
|---------------------------------|--|---|--------|-------|-------|---|----|----|
|                                 |  |   |        |       | C     | A | F0 | F1 |
| <b>INPUT OUTPUT (Continued)</b> |  |   |        |       |       |   |    |    |
| AND Pd A                        | Pd ← Pd AND Ad<br>Ad ← Ad, 0 ≤ Ad ≤ 7  | Logical AND contents of Accumulator with designated Port A.       | 2      | 1     |       |   |    |    |
| In Pd A                         | A ← Pd, 0 ≤ Pd ≤ 7   | Input data from designated port Pd to Accumulator.                | 2      | 1     |       |   |    |    |
| NSA BUS                         | A ← (BUS)  | Input strobed BUS data into Accumulator.                          | 2      | 1     |       |   |    |    |
| MOVD Pd A                       | AD ← Ad, Pd<br>A ← Pd, 0 ≤ Ad ≤ 7  | Move contents of designated port Ad to Accumulator.               | 2      | 1     |       |   |    |    |
| MOVD Pd A                       | Pd ← Ad, Ad<br>A ← Pd, 0 ≤ Ad ≤ 7  | Move contents of Accumulator to designated port Ad.               | 2      | 1     |       |   |    |    |
| ORL BUS #data                   | BUS ← BUS OR data  | Logical OR immediate specified data with contents of bus.         | 2      | 2     |       |   |    |    |
| ORL Pd A                        | (Pd) ← (Pd) OR Ad<br>Ad ← Ad, 0 ≤ Ad ≤ 7   | Logical OR contents of Accumulator with designated Port A.        | 2      | 1     |       |   |    |    |
| ORL Pd #data                    | (Pd) ← (Pd) OR data<br>data ← 1 - 2  | Logical OR immediate specified data with designated port (1 - 2). | 2      | 2     |       |   |    |    |
| OUTL BUS A                      | (BUS) ← A  | Output contents of Accumulator onto BUS.                          | 2      | 1     |       |   |    |    |
| OUTL Pd A                       | Pd ← A, 0 ≤ Pd ≤ 7   | Output contents of Accumulator to designated port (0 - 7).        | 2      | 1     |       |   |    |    |
| <b>REGISTERS</b>                |  |   |        |       |       |   |    |    |
| DEC Rr                          | (Rr) ← (Rr) - 1, 0 ≤ Rr ≤ 7  | Decrement by 1 contents of designated register.                   | 1      | 1     |       |   |    |    |
| INC Rr                          | (Rr) ← (Rr) + 1, 0 ≤ Rr ≤ 7  | Increment by 1 contents of designated register.                   | 1      | 1     |       |   |    |    |
| INC @ Rr                        | @(Rr) ← @(Rr) + 1<br>Rr ← 0 - 7  | Increment indirect by 1 the contents of data memory location.     | 1      | 1     |       |   |    |    |
| <b>SUBROUTINE</b>               |  |   |        |       |       |   |    |    |
| CALL #addr                      | (SP) ← (PC)<br>(SP) ← (PSW 4 - 7)<br>(SP) ← (SP) - 1<br>PC ← 10 + addr 8 - 10<br>(PC) ← 7 + addr 9 - 7<br>(PC) ← DBF | Call designated Subroutine.                                       | 2      | 2     |       |   |    |    |
| RET                             | (SP) ← (SP) + 1<br>(PC) ← (SP)   | Return from Subroutine without restoring Program Status Word.     | 2      | 1     |       |   |    |    |
| RETR                            | (SP) ← (SP) + 1<br>(PC) ← (SP)<br>(PSW 4 - 7) ← (SP)   | Return from Subroutine restoring Program Status Word.             | 2      | 1     |       |   |    |    |
| <b>FLAGS</b>                    |  |   |        |       |       |   |    |    |
| CPL C                           | (C) ← NOT (C)  | Complement Content of carry bit.                                  | 1      | 1     | *     |   |    |    |
| CPL F0                          | (F0) ← NOT (F0)  | Complement Content of Flag F0.                                    | 1      | 1     |       | * |    |    |
| CPL F1                          | (F1) ← NOT (F1)  | Complement Content of Flag F1.                                    | 1      | 1     |       |   | *  |    |
| CLR C                           | (C) ← 0  | Clear content of carry bit to 0.                                  | 1      | 1     | *     |   |    |    |
| CLR F0                          | (F0) ← 0   | Clear content of Flag 0 to 0.                                     | 1      | 1     |       | * |    |    |
| CLR F1                          | (F1) ← 0   | Clear content of Flag 1 to 0.                                     | 1      | 1     |       |   | *  |    |
| <b>MISCELLANEOUS</b>            |  |   |        |       |       |   |    |    |
| NOP                             |  | No operation.   | 1      | 1     |       |   |    |    |

Symbol Definitions:

| SYMBOL                          | DESCRIPTION                      |
|---------------------------------|----------------------------------|
| A                               | The Accumulator                  |
| AC                              | The Auxiliary Carry Flag         |
| ador                            | Program Memory Address (12 bits) |
| b                               | Bit Designator (b = 0 - 7)       |
| BS                              | The Bank Switch                  |
| BUS                             | The BUS Port                     |
| C                               | Carry Flag                       |
| CLK                             | Clock Signal                     |
| CNT                             | Event Counter                    |
| D                               | Nibble Designator (4 bits)       |
| data                            | Number of Expression (8 bits)    |
| DBF                             | Memory Bank Flip-Flop            |
| F <sub>0</sub> - F <sub>1</sub> | Flags 0, 1                       |
| I                               | Interrupt                        |
| P                               | "In-Page" Operation-Designator   |

| SYMBOL                          | DESCRIPTION   |
|---------------------------------|---|
| p                               | Port Designator (p = 1, 2 or 4 - 7)                               |
| PSW                             | Program Status Word   |
| r                               | Register Designator (r = 0, 1 or 0 - 7)                           |
| SP                              | Stack Pointer   |
| T                               | Timer   |
| TF                              | Timer Flag  |
| T <sub>0</sub> , T <sub>1</sub> | Testable Flags 0, 1   |
| X                               | External RAM  |
| #                               | Prefix for Immediate Data   |
| @                               | Prefix for Indirect Address                                       |
| \$                              | Program Counter's Current Value                                   |
| (x)                             | Contents of Register  |
| ((xx))                          | Contents of Memory Location Addressed by the Contents of Register |
| —                               | Replaced By   |

TYPICAL APPLICATIONS

Figure 6 shows a typical way to use the NS80C48 Microcomputers in a stand-alone system

- Crystal used is
  - Parallel resonant
  - AT cut
  - 1 to 6 MHz
- All outputs are standard TTL compatible @ 5V
- P1 and P2 outputs drive 5V CMOS directly (others require 10-15K $\Omega$  pullup for CMOS compatibility)

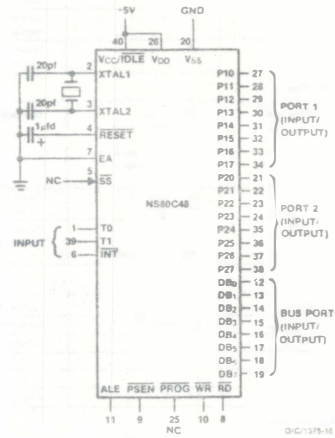


FIGURE 6. Stand-Alone NS80C48

TYPICAL APPLICATIONS - Continued

Figure 7 shows a typical way to add more Program Memory (CMOS ROM), Data Memory (CMOS RAM) and I/O to the NS80C48.

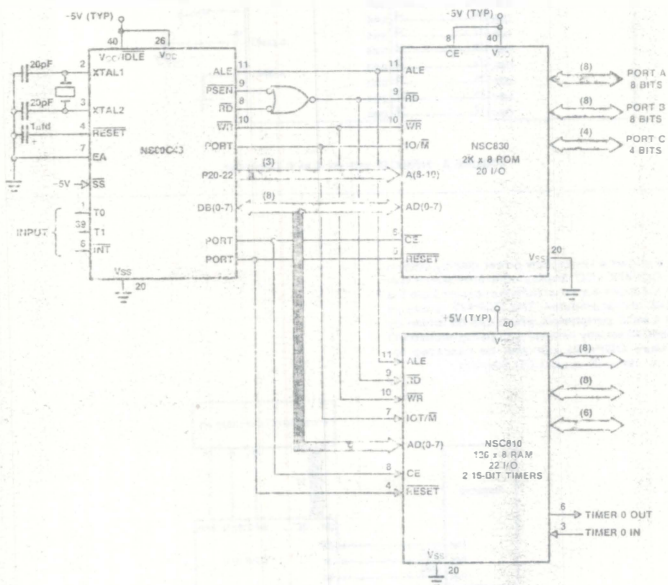


FIGURE 7. Expanded Low Power System

TYPICAL APPLICATION - Continued

Figure 8 shows a typical way to add an A to D converter to the NS80C48 microcomputer.

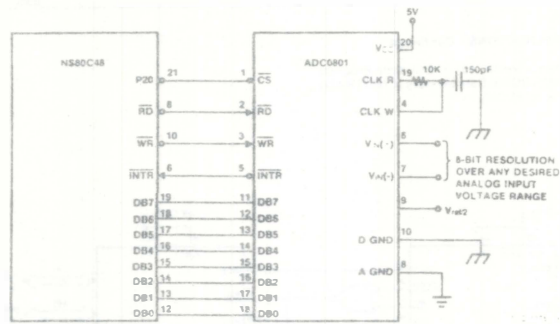


FIGURE 8. NS80C48 with an A to D Converter

Figure 9 shows a typical low power display system with a COP472 LCD display controller which can drive 4 1/2 digits via serial communication from the NS80C48 microcomputer. The COP472 is one of several CMOS peripherals which can be driven by the NS80C48 serially without occupying needed I/O ports. More COP472s can also be cascaded together for use with larger LCD displays.

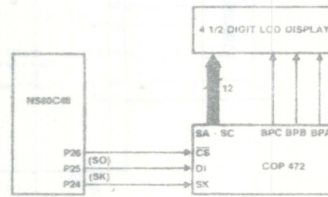


FIGURE 9. LCD Display Application

D.C. 1375-1





