



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
FACULTAD DE INGENIERIA**

**G-602006**

602006



CAJA 162

**FAC. DE INGENIERIA  
DOCUMENTACION**

**APUNTES DE  
MICROPROCESADORES Y  
SISTEMAS DIGITALES**

**JUAN B. MARTINEZ**

**DIVISION DE INGENIERIA MECANICA Y ELECTRICA  
DEPARTAMENTO DE COMPUTACION**

FI/DIME/86-030

MICROPROCESADORES Y SISTEMAS DIGITALES  
.....

	pág
<b>INTRODUCCION</b>	2
<b>PARTE I: DISEÑO</b>	
1. Organización y funcionamiento básico de una microcomputadora -----	3
2. Criterios para el diseño de sistemas basados en microprocesadores -----	17
3. Diseño de un sistema básico Z80 -----	27
<b>PARTE II: PERIFERICOS</b>	
5. Periféricos para microprocesadores -----	35
6. Indicadores visuales fosforescentes -----	66
7. Control de video -----	71
<b>PARTE III: APLICACIONES</b>	
8. Líneas telefónicas y módems -----	93
9. Algunas características de procesadores de 16 bits -----	110
10. Medición de temperatura con microcomputadoras -----	120
11. Sistema de adquisición y despliegue de datos	140
12. Red telemétrica digital -----	150
13. Sistema automático de supervisión y control	166
<b>PARTE IV: EJERCICIOS</b>	
<b>APENDICES</b>	
a) Espec. Z80 CPU -----	203
b) Espec. 2716 MOS ROM -----	236
c) Espec. 6116 RAM -----	243
d) Espec. MOC 3011 CONTROLADOR DE TRIAC---	248
e) Espec. ADC0801 CONVERTOR A/D -----	254
f) Espec. MTX1632 CONTROLADOR DE VIDEO ---	260
i) Espec. MM58174 RELOJ EN TIEMPO REAL ---	264

FAC. DE INGENIERIA  
DOCUMENTACION

## INTRODUCCION:

A lo largo de los capítulos de estos apuntes, se pretende dar una visión general de los microprocesadores y sus diversas aplicaciones en sistemas digitales.

El contenido se ha dividido en 4 partes: la parte I se enfoca en los fundamentos del funcionamiento de las microcomputadoras, así como al empleo de diversas técnicas para el diseño de sistemas basados en microprocesadores.

En la parte II se revisan los periféricos más comúnmente empleados con microcomputadoras: teclado, despliegues luminosos, receptores comerciales de TV, etc. En esta parte se incluye también un capítulo relacionado a la medición de temperatura con microcomputadoras.

La parte III está dedicada a la descripción de aplicaciones con énfasis en 3 ramas: redes de telemetría, sistemas de adquisición de datos y sistemas para el control de procesos.

Finalmente en la parte IV se proponen ejercicios que cubren gran parte del material y permiten su mejor aprovechamiento.

En los apéndices se dan las especificaciones técnicas de varios circuitos empleados en diversos acoplamientos descritos en varios de los capítulos de los apuntes.

## ORGANIZACION Y FUNCIONAMIENTO BASICO DE LAS MICROCOMPUTADORAS

.....

Componentes Básicos. Funcionamiento y Operación

### COMPONENTES BASICOS

Las computadoras en general están formadas funcionalmente por 4 unidades básicas:

- La unidad aritmética lógica
- La unidad de control
- La memoria
- La unidad de entrada-salida

A partir del año 1970, sin embargo, la avanzada tecnología en semiconductores permitió la fabricación de una componente que en un solo circuito integrado agrupaba a 2 de las unidades básicas: la unidad aritmética lógica y la unidad de control. A este dispositivo se le conoce con el nombre de *microprocesador*.

Juntos, el microprocesador, la memoria y los acoplamientos de entrada/salida forman el dispositivo que conocemos como microcomputadora (ver fig 1).

Las microcomputadoras pueden manejar, a su vez, otros dispositivos que se conocen como *periféricos* y que le permiten comunicarse con el mundo exterior. Ejemplos: terminales de video, teclados, unidades de disco flexible, impresoras, etc.

Como previamente se mencionó, la microcomputadora consta de los bloques mostrados en la fig 1 y de los que enseguida se hace una descripción:

### EL MICROPROCESADOR

Los microprocesadores son dispositivos electrónicos complejos. Internamente contienen miles de transistores alojados en una superficie pequeñísima (alrededor de 16 mm<sup>2</sup>), en un cristal de silicio.

Este cristal se aloja en un encapsulado que típicamente (en microprocesadores de 8 bits) tiene 40 conexiones al exterior, formando un circuito integrado compacto y con una superficie muchísimo mayor a la del cristal de silicio: 7.5 cm<sup>2</sup>.

Los transistores están conectados de tal manera de formar dos tipos de circuitos: *compuertas lógicas* y *flip flops*. Estos circuitos se acoplan para formar otros con funciones más elaboradas:

- *registros*
- *contadores*
- *codificadores*
- *sumadores*

Finalmente, los circuitos previamente señalados se combinan para desarrollar, la relativamente complicada función que realiza un microprocesador.

Para facilitar la comprensión de la forma en que los circuitos del microprocesador interaccionan, se supondrá la existencia de un microprocesador básico y formado sólo por sus componentes más elementales. Cabe aclarar que los microprocesadores vendidos comercialmente contienen mucho más componentes que las que aquí se plantearán. El funcionamiento básico de ambas versiones es, sin embargo, idéntico.

En la fig 2 se muestra el microprocesador básico conectado a su memoria y sus acoplamientos de entrada/salida.

El circuito microprocesador contiene dos unidades básicas: la unidad aritmética lógica (ALU) y la unidad de control. Además, internamente el microprocesador hace uso de varios registros que emplea para almacenamiento y transferencia de datos entre los propios circuitos del microprocesador, o entre el microprocesador y la memoria o la unidad de entrada/salida.

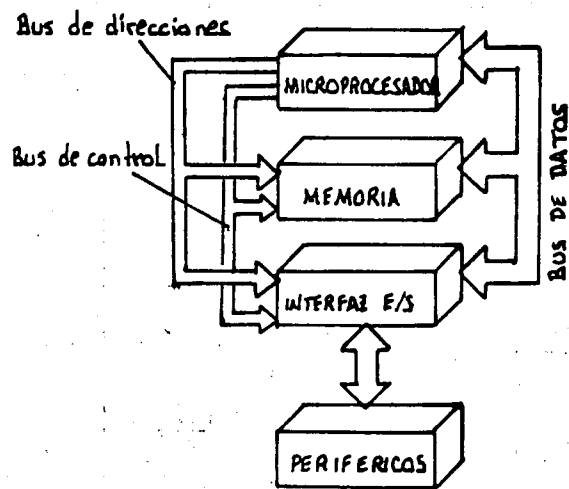


FIG 1. DIAGRAMA A BLOQUES DE LA MICROCOMPUTADORA.

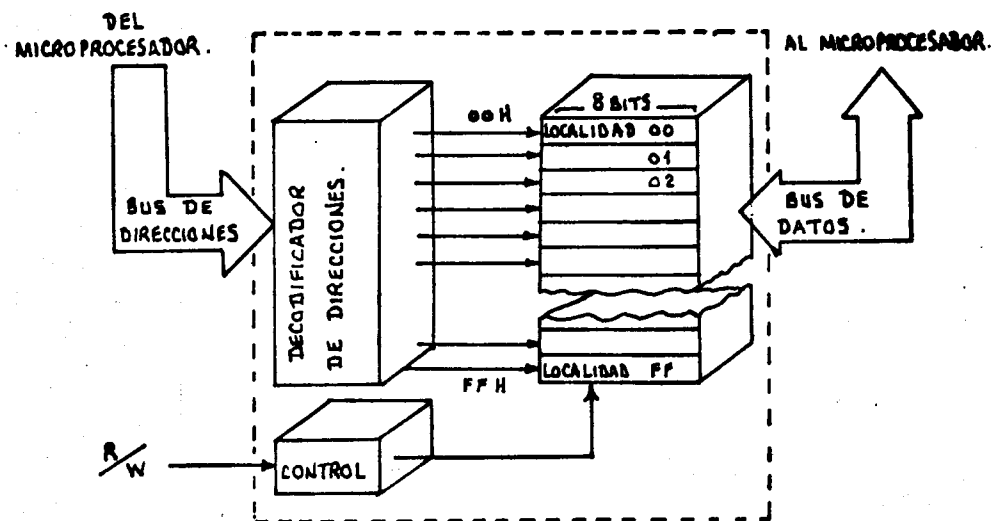


FIG 3. LA MEMORIA

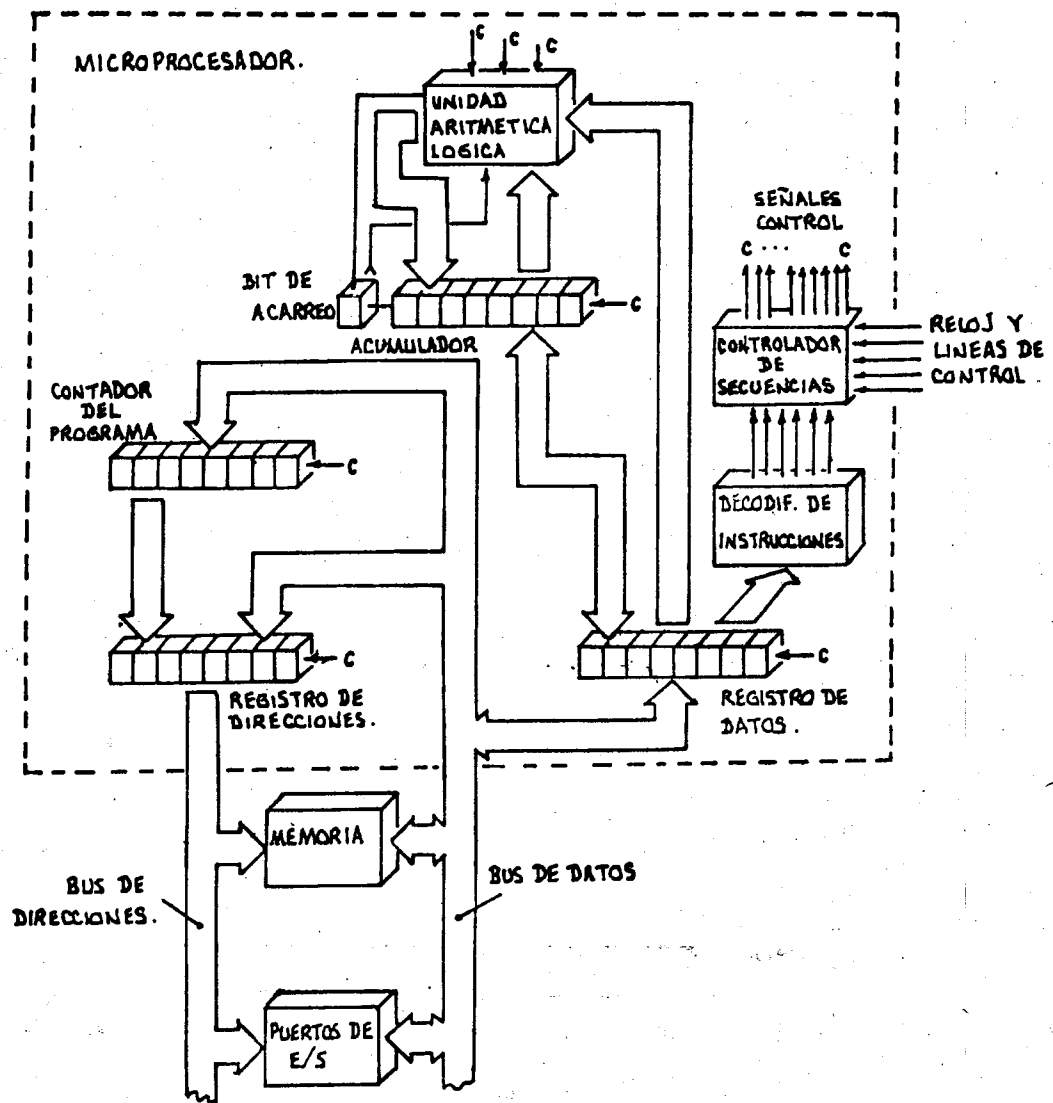


FIG 2 : DIAGRAMA DE LAS COMPONENTES DEL MICROPROCESADOR.

### UNIDAD ARITMETICA LOGICA

Como su nombre lo indica, la unidad aritmética-lógica permite realizar operaciones binarias aritméticas y lógicas entre dos operandos. El ALU básico del microprocesador propuesto está formado por un sumador y circuitos combinatoriales que efectúan las operaciones lógicas o aritméticas. La unidad del control indica al ALU qué operación específica se debe llevar a cabo, y generalmente la realiza con el valor de los operandos de dos registros:

- el registro de datos, y
- el acumulador

El acumulador se utiliza a su vez para almacenar el resultado de la operación, por lo que el valor previo de uno de los operandos se pierde.

Las operaciones que un ALU básico puede hacer son:

- las operaciones aritméticas  
Suma y Resta
- Las operaciones lógicas  
AND, OR y NOT

### UNIDAD DE CONTROL

La unidad de control coordina la función de las otras tres unidades. La unidad de control permite por ejemplo almacenar un dato en memoria, realizar una suma o transferir un byte a la unidad de E/S. La unidad de control es alimentada con las instrucciones del programa en curso y su trabajo es la ejecución, una a una, de estas instrucciones.

La unidad de control está formada por el decodificador de instrucciones y el controlador de secuencias.

### EL DECODIFICADOR DE INSTRUCCIONES

Después que una instrucción se lee de memoria y se guarda en el registro de datos, la instrucción se decodifica en este circuito. El decodificador examina el código (8 bits en este caso) y decide la operación que ha de realizarse.

### EL CONTROLADOR DE SECUENCIAS

El controlador de secuencias recibe la información del decodificador de instrucciones y genera las señales de control necesarias



para ejecutar la instrucción solicitada. Estas señales de control son niveles de voltaje o pulsos dirigidos a las componentes adecuadas y que pueden ser cualquiera de las descritas anteriormente.

## REGISTROS INTERNOS

Como se había mencionado anteriormente, los registros internos del microprocesador permiten por un lado el almacenamiento de datos que pueden ser los resultados de alguna operación, y por el otro la transferencia de información entre el microprocesador y la memoria o la unidad de *E/S*. Estos registros son:

- El acumulador
- El registro de direcciones
- EL registro de datos
- El contador del programa

### EL ACUMULADOR

El acumulador es el registro más útil. En interacción con el ALU, tiene una función doble: antes de realizar la operación contiene al operando, después de la operación contiene el resultado de la misma.

Además tiene una función importante como registro de almacenamiento temporal de datos leídos de memoria, o antes de escribirlos en la misma.

### EL REGISTRO DE DATOS

Es un registro para almacenamiento temporal de datos de o hacia la memoria. En el caso de instrucciones, el registro de datos almacena la instrucción mientras el decodificador de instrucciones la decodifica. En interacción con el ALU, este registro guarda uno de los dos operandos que el ALU requiere para efectuar las operaciones aritméticas o lógicas.

### EL REGISTRO DE DIRECCIONES

En este registro se almacena temporalmente la dirección de memoria o el dispositivo de *E/S* de o hacia donde se pretenda transferir la información.

## EL CONTADOR DEL PROGRAMA

El contador del programa contiene la dirección de la localidad de memoria de donde se leerá la siguiente instrucción. Regularmente, sigue una secuencia consecutiva, a menos que su contenido se vea modificado por una instrucción de salto (**JUMP**)

Todos los circuitos descritos realizan, dentro del microprocesador una función relativamente compleja. Sin embargo, el microprocesador sólo es incapaz de realizar alguna función. Necesita de componentes adicionales que le permiten desarrollar su potencial funcional. Estas componentes son:

- La memoria
- Los acoplamientos de entrada/salida

## LA MEMORIA

A pesar de que actualmente existen memorias comerciales que manejan hasta 64 K bytes de información, se explicará el funcionamiento de una memoria de sólo 258 bytes para simplificar el análisis. En notación hexadecimal, sus direcciones van de la **00H** a la **FFH**.

En la fig 3 se muestra el diagrama de esta memoria. La memoria se comunica con el microprocesador a través de dos buses o ductos: el bus de direcciones que contiene la dirección de la localidad de la que se desea leer o escribir, y el bus de datos, que transfiere de o hacia el microprocesador el dato leído o escrito en memoria. El bus de datos es, por tanto, bidireccional, mientras el de direcciones es unidireccional.

El decodificador de direcciones selecciona la línea correspondiente a la localidad elegida por el microprocesador. El microprocesador debe también, a través de la línea **READ/WRITE**, especificar su deseo, leer o escribir en la memoria. En el caso de **LECTURA**, el bus de datos llevará los datos de la localidad seleccionada hacia el microprocesador. En el caso de **ESCRITURA**, se almacenará en memoria; en la localidad seleccionada, el dato proveniente del microprocesador, también a través del bus de datos.

## LOS ACOPLAMIENTOS DE E/S

Como anteriormente se mencionó, la microcomputadora se comunica con el mundo exterior-o sea los periféricos- a través de los acoplamientos de entrada/salida.

Los periféricos son los dispositivos que permiten a la computadora interactuar, ya sea con variables físicas externas o con un operador. Dispositivos como los teclados, los despliegues luminosos y una televisión son periféricos que permiten al operador programar la computadora y obtener resultados.

Por otro lado, dispositivos como los relevadores o los conversores analógico - digitales, son periféricos que permiten que la microcomputadora *lea* y actúe sobre variables físicas como podrían ser el nivel en un tanque, la presión en una tubería, la temperatura de un horno, etc.

Los acoplamientos de entrada/salida están constituidos por *puertos* de entrada o salida conectados con circuitos que son los que manejan directamente a los periféricos. De esta forma, encontramos circuitos electrónicos que manejan terminales, discos floppys, impresoras, etc.

Sin embargo, lo único que la microcomputadora *ve* hacia el mundo exterior son los puertos de entrada o de salida; los acoplamientos o circuitos electrónicos asociados a éstos no interactúan directamente con la microcomputadora sino con los periféricos.

Un puerto de salida no es otra cosa que un registro (8 flip-flops) en donde la microcomputadora puede almacenar 8 bits.

Un puerto de entrada es simplemente un conjunto de 8 compuertas lógicas de donde la microcomputadora puede leer el estado de 8 bits.

De esta manera, leyendo o escribiendo en los puertos, con el programa adecuado, la microcomputadora puede manejar fácilmente periféricos tan complejos como podría ser una TV casera, debido, como ya se dijo, a que son los circuitos o acoplamientos electrónicos conectados a esos puertos los que se encargan del trabajo especializado, como lo sería escribir texto en la TV comercial.

La transferencia de información entre el microprocesador, la memoria y los acoplamientos de E/S se realiza a través de los llamados **BUS DE DATOS** y **BUS DE DIRECCIONES** que no son sino un conjunto de líneas cuya función específica se verá posteriormente.

## FUNCIONAMIENTO Y OPERACION DE LAS MICROCOMPUTADORAS

Una vez descritas las partes principales de la microcomputadora se procederá a analizar la manera como se ejecutan los programas almacenados en memoria, así como el manejo de datos en la misma.

## LAS INSTRUCCIONES

Las instrucciones almacenadas en memoria constan de dos partes:

- El código de operación
- El operando

El código de operación -que se puede almacenar en un byte de memoria-, especifica qué instrucción debe ejecutarse.

El operando, que es el byte siguiente, es el dato sobre el cual se realizará la operación.

Por ejemplo, supóngase un microprocesador donde la instrucción *LDA 07H* se codifica como:

```

86          en hexadecimal  ó
07
10000110   en binario
00000111
  
```

El primer byte es el código de operación; el microprocesador lo lee y lo identifica como la instrucción **CARGA ACUMULADOR CON OPERANDO**.

El siguiente byte es el operando y en este caso es el número 7. De esta manera, el microprocesador ejecuta la instrucción leyendo el operando de memoria y almacenándolo en el acumulador.

La longitud de palabra de las instrucciones y operandos dependen directamente del microprocesador con el cual se esté trabajando. Los microprocesadores con una longitud de palabra de 8 bits, tienen 8 líneas en su bus de datos. Eso no significa necesariamente que el código de operación y los operandos tengan también una longitud fija de 8 bits, ya que microprocesadores como el **Z80** tienen códigos de operación de 1 ó 2 bytes y operandos también de 1 ó 2 bytes dependiendo de la instrucción.

En el microprocesador básico que se ha propuesto, para facilitar el análisis se supondrán los códigos de operación y los operandos con una longitud fija de 8 bits.

## LA SECUENCIA FETCH - EXECUTE

Cuando la microcomputadora ejecuta un programa almacenado en memoria, lo hace a través de un ciclo básico que se repite constantemente.

El programa consta de instrucciones almacenadas en memoria, en cierto orden, que el microprocesador lee y ejecuta, una a una. A este ciclo repetido de lectura -ejecución de la instrucción, se le conoce en general por el término en inglés *fetch - execute*.

Cuando la microcomputadora inicia operaciones -mediante la señal **RESET** que se conecta al microprocesador-, entra en la fase *fetch*. La primera instrucción se lee y se decodifica en el microprocesador. Realizado esto, el microprocesador procede a la fase *execute* en la cual debe llevar a cabo las operaciones involucradas en la instrucción en cuestión.

Este ciclo se repite indefinidamente, de acuerdo con el programa almacenado en memoria.

### EJECUCION DEL PROGRAMA

Para ampliar la descripción del funcionamiento de la microcomputadora, nos valdremos de un sencillo programa almacenado en memoria. El programa suma los números 7 y 10 (decimales) y almacena el resultado en el acumulador.

El programa es el siguiente:

INSTRUCCION	SIMBOLOS	CODIGO DE OP	OPERANDO	DESCRIPCION
LOAD ACCUMULATOR	LDA, 07H	1 0 0 0	0 1 1 0 0 0 0 0 0 1 1 1	Carga el operando en el acumulador
ADD	ADD, 0AH	1 0 0 0	1 0 1 1 0 0 0 0 1 0 1 0	Suma el contenido del acumulador al operando y pone el resultado en el acumulador
HALT	HLT	0 0 1 1	1 1 1 0 ---	Detiene la operación del microprocesador

El programa estaría almacenado en memoria, a partir de la dirección 0 0 0 0 H, de la siguiente manera:

DIRECCION DE MEMORIA		CONTENIDO BINARIO		SIMBOLOS
0 0 0 0	0 0 0 0	1 0 0 0	0 1 1 0	LDA
0 0 0 0	0 0 0 1	0 0 0 0	0 1 1 1	7 <sub>10</sub>
0 0 0 0	0 0 1 0	1 0 0 0	1 0 1 1	ADD
0 0 0 0	0 0 1 1	0 0 0 0	1 0 1 0	10 <sub>10</sub>
0 0 0 0	0 1 0 0	0 0 1 1	1 1 1 0	HALT

es decir, los operandos de las instrucciones *LDA* y *ADD* se encuentran almacenados en la localidad siguiente a su código de operación correspondiente. La instrucción *HLT* no lleva operando.

Enseguida se seguirán, paso a paso, los procedimientos del microprocesador para ejecutar este programa. Para iniciar, debe generarse la señal *RESET* al microprocesador. El contenido del contador del programa se pone automáticamente en ceros, apuntando a la primera localidad de memoria.

Como se explicó anteriormente, el proceso se inicia con la fase *fetch* (ver fig 4), de la primera instrucción. La secuencia de eventos, generada por el controlador de secuencias es la siguiente:

- El contenido del contador del programa se transfiere al registro de direcciones
- El contenido del contador del programa se incrementa en una unidad, sin cambiarse el contenido del registro de direcciones
- El registro de direcciones, direcciona la memoria, y el dato de la localidad seleccionada se almacena en el registro de datos. En este momento, el contenido de este registro es el código de operación de la instrucción *LDA*
- El código de operación se decodifica en el decodificador de instrucciones. Este circuito reconoce que se trata de la instrucción *LDA* y a través de una señal se lo comunica al controlador de secuencias.

Una vez concluida la fase *fetch*, se procede a la fase *execute*, (ver fig 4). El controlador de secuencias conoce ahora la instrucción y debe generar las señales de control para ejecutarlas, leyendo el operando de la siguiente localidad de memoria y almacenándolo en el acumulador. La secuencia de eventos de esta fase es la siguiente:

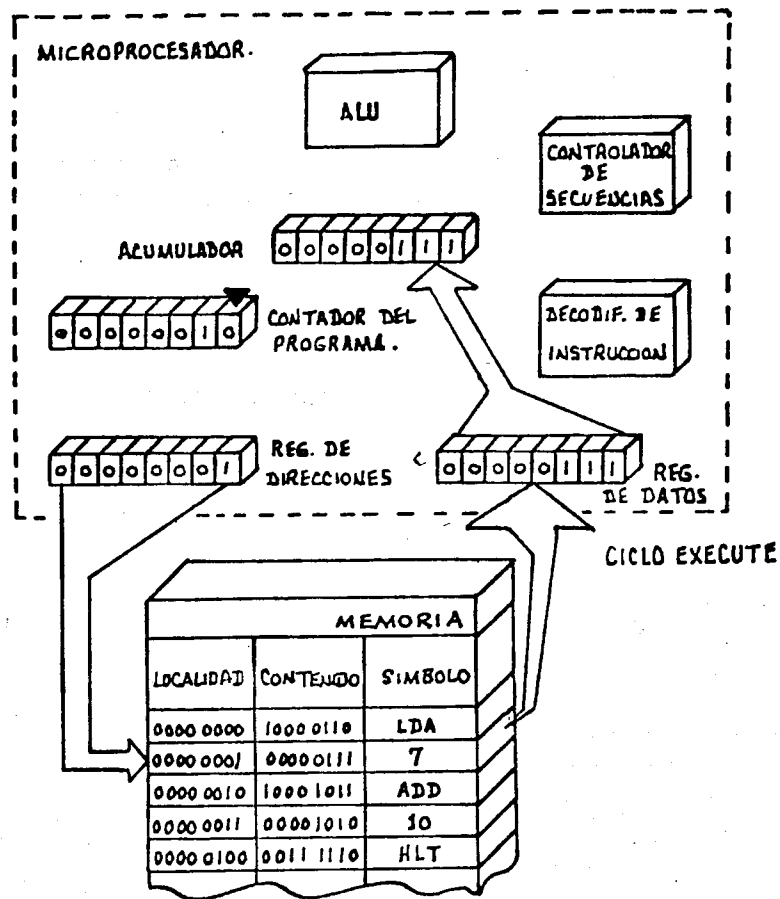
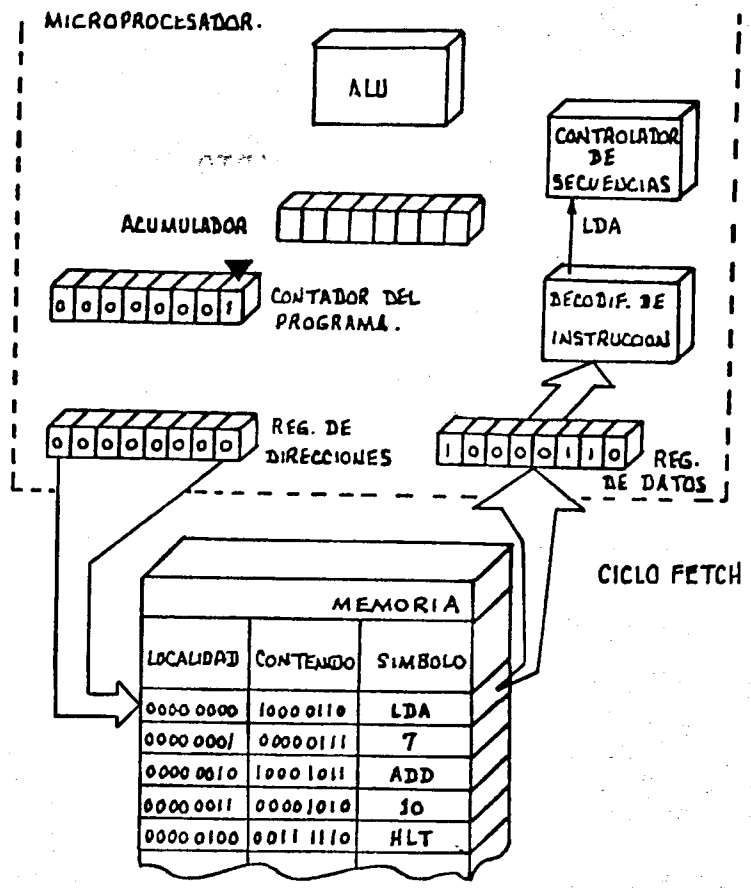


FIG 4: LECTURA Y EJECUCION DE LA INSTRUCCION LDA.

- a) El contador del programa, ahora con el contenido 01H, se carga en el registro de direcciones
- b) El contenido del contador del programa se incrementa en una unidad
- c) El registro de direcciones selecciona a través del bus de direcciones la localidad 02H de la memoria y el contenido del registro de datos será, por tanto, 07H en este momento
- d) El operador se transfiere del registro de datos al acumulador terminando con la fase de ejecución de la primera instrucción

Idéntico procedimiento sigue el microprocesador para realizar las fases *fetch* y *execute* de la instrucción *ADD*. En la fig 5 se describe la secuencia de eventos. En la fase de ejecución de la instrucción, el ALU efectúa la suma del acumulador con el contenido del registro de datos, almacenando el resultado (17<sub>10</sub>) de nuevo en el acumulador.

Finalmente, el procesador efectúa el 'fetch' de la instrucción *HALT* la cual es identificada por el decodificador de instrucciones. El controlador de secuencias detiene en este momento la operación del procesador.

A pesar de la simpleza del microprocesador propuesto y del programa analizado, su principio de funcionamiento es idéntico al de los microprocesadores comerciales, cuya arquitectura interna es más compleja que la que aquí se planteó. Estos microprocesadores cuentan generalmente con varios registros que funcionan como acumuladores, y un amplio repertorio de instrucciones. El microprocesador **Z80**, por ejemplo, cuenta con 14 registros de propósito general y un conjunto de 158 instrucciones.

FAC. DE INGENIERIA  
DOCUMENTACION



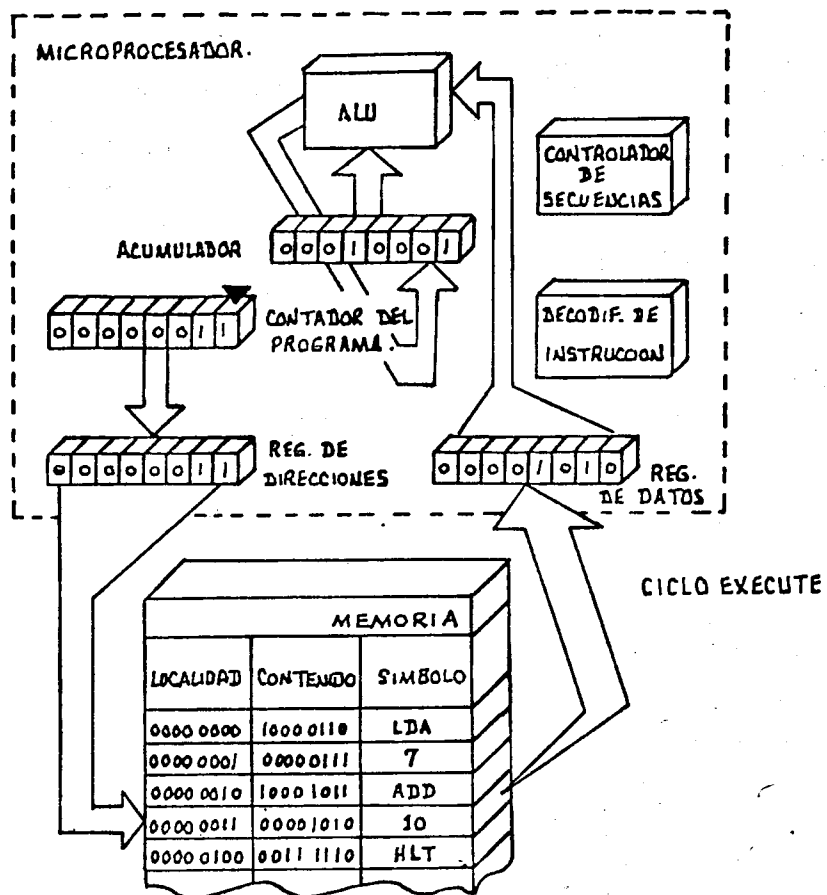
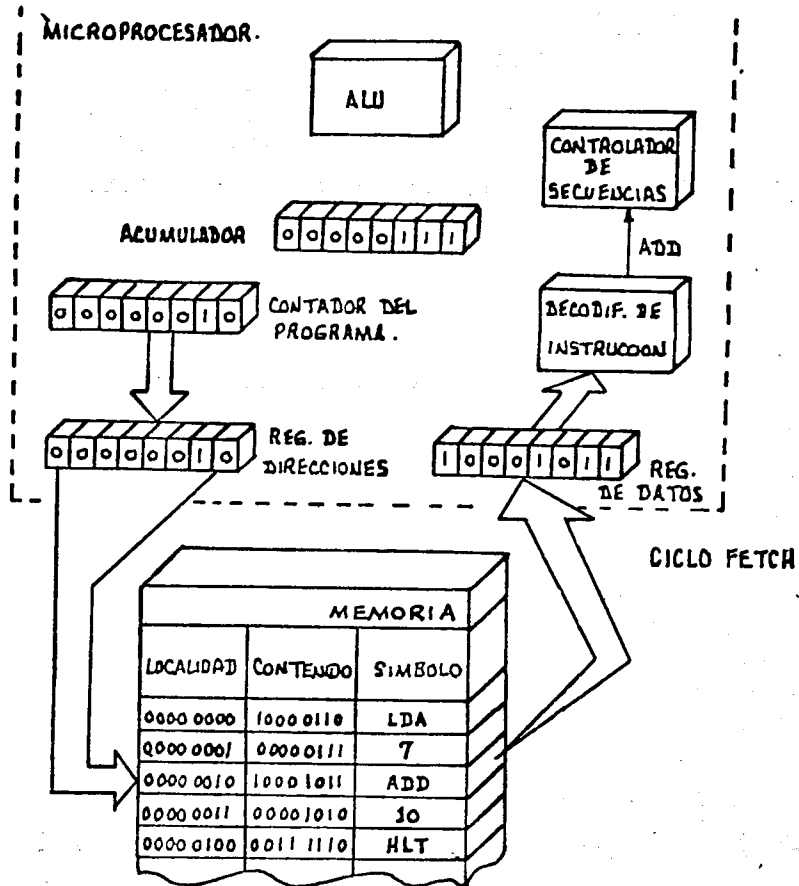


FIG. 5: LECTURA Y EJECUCION DE LA INSTRUCCION 'ADD.'

## CRITERIOS PARA EL DISEÑO DE SISTEMAS BASADOS EN MICROPROCESADORES

.....  
Introducción. El bus. Compuertas tristate.  
Circuitos TTL. Circuitos MOS. Análisis de  
los circuitos digitales. Análisis estático  
y análisis dinámico

### INTRODUCCION

Dentro de la rama electrónica, los circuitos digitales están cada día ganando terreno a los circuitos analógicos, desde el punto de vista de su utilización en sistemas electrónicos.

El empleo masivo que se hace actualmente de los microprocesadores en todo tipo de aplicaciones, ha fomentado que, en gran cantidad de casos, los sistemas digitales reemplacen ventajosamente a sus similares analógicos.

Tal es el caso, por ejemplo, de los sistemas de control de procesos en plantas industriales.

El control de procesos había sido, hasta 1970, campo exclusivo de los sistemas analógicos que, si bien resuelven los problemas con sencillez y confiabilidad, carecen de la versatilidad y capacidad de los sistemas digitales. En la actualidad, un solo sistema computarizado puede atender simultáneamente decenas de mallas de control<sup>(1)</sup>, y puede ser fácilmente reprogramado para cambiar algún parámetro de cualesquiera de los procesos.

Para comprender el funcionamiento de los sistemas digitales basados en microprocesadores, es indispensable revisar 2 conceptos básicos: el bus, (también llamado ducto), y las salidas 'tristate'

#### **EL BUS:**

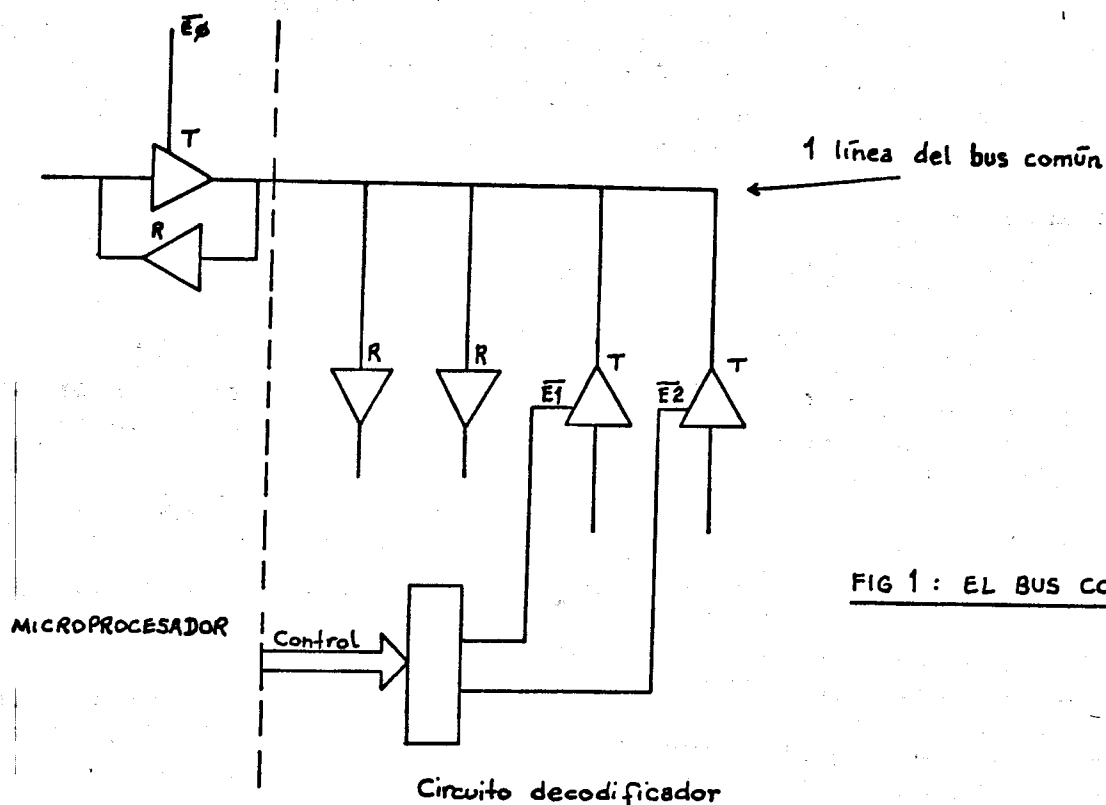
Se conoce como bus a 1 ó varios puntos comunes de interconexión de varios circuitos, con el objeto de transferir información entre múltiples dispositivos transmisores y receptores.

(1) Se entiende por malla de control al procedimiento (con técnicas digitales ó analógicas) mediante el cual es posible, por medio de realimentación, conservar el valor de ciertas variables muy cercanas a un valor prefijado y conocido como 'set point'. Tal es el caso del nivel en un tanque de almacenamiento ó la temperatura de algún proceso en la industria química por ejemplo.

Como ejemplo tómesese la fig 1, en donde se tienen 3 transmisores y 3 receptores conectados al mismo punto. En un instante dado, *sólo un transmisor* puede emitir su información a los receptores conectados al mismo punto, porque en caso contrario habría interferencia entre ellos. Es necesario entonces, algún control que silencie o deshabilite a los transmisores que no estén activos. A este esquema se le conoce como control de 'tristate' o control de 'tercer estado'. En la fig 1, las líneas que lo manejan se muestran con la variable  $\bar{E}$  (enable).

Se puede así tener por ejemplo 8 puntos o líneas comunes para transferir información entre varios dispositivos de un sistemas. Entonces se habla del "**BUS DE DATOS**" de una memoria o un microprocesador, que es generalmente bidireccional, es decir, puede transmitirse o recibirse información a través de la misma línea.

Similarmente se habla también del "**BUS DE DIRECCIONES**" y del "**BUS DE CONTROL**" para distinguir al conjunto de líneas que se emplea para transmitir información de direccionamiento o control respectivamente; estas líneas son siempre **UNIDIRECCIONALES**.

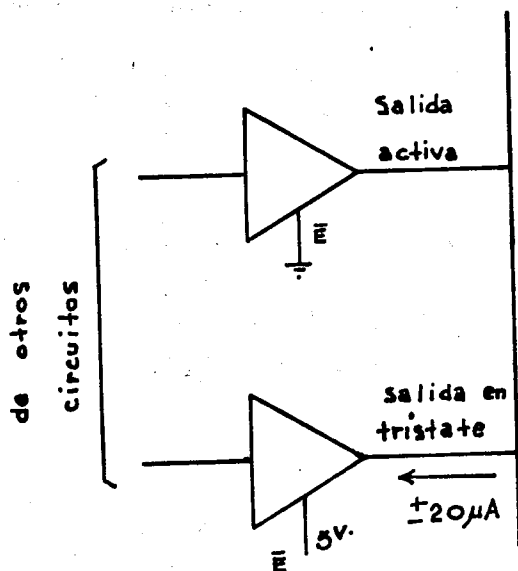


**FIG 1 : EL BUS COMUN**

### COMPUERTAS TRISTATE:

Las salidas de las compuertas lógicas manejan 2 niveles en sus salidas: el nivel alto y el nivel bajo. Sin embargo, si éstas compuertas transmiten su información a un bus común, es necesario, por los motivos previamente expuestos, que en ciertos momentos sus salidas queden virtualmente desconectadas del bus. Esto se consigue según se mencionó, con una línea de control adicional que se llama 'control de tristate' (ver fig 2).

Electrónicamente, sin embargo el circuito no está totalmente desconectado. Existe una pequeña corriente de fuga, conocida generalmente en inglés como "Tristate Leakage Current". Su valor es variable dependiendo del circuito, pero la mayoría de los fabricantes lo han estandarizado  $+20 \mu A$ , para el bus en estado alto (5V) y  $-20 \mu A$  si el bus está en estado bajo (0V), tomando como positiva a la corriente que entra a la compuerta.



$EI = 5V$  salida desconectada

$EI = 0V$  salida activa.

FIG 2 : EL CONTROL TRISTATE

## CIRCUITOS TTL. CIRCUITOS MOS

Dentro de los sistemas digitales, se manejan principalmente 2 tipos de circuitos. Los *MOS* (de Metal Oxide Silicon) y los *TTL* (de transistor - transistor logic).

Los circuitos *MOS* pueden a su vez ser de 3 tipos: *NMOS*, *PMOS* ó *CMOS* dependiendo de la tecnología empleada. Al lector interesado en conocer detalles de fabricantes de estas familias lógicas se les recomienda la ref 1. Entre sus ventajas, los circuitos *MOS* presentan bajo consumo y muy alta impedancia de entrada. Un gran porcentaje de los circuitos *LST* (de: Large Scale Integration) como son los microprocesadores y las memorias, se fabrican con esta tecnología.

Los circuitos *TTL* son ampliamente usados, sobre todo en circuitos *SST* y *MSI* (de: Small Scale Integration y Medium Scale Integration) como son las compuertas lógicas, contadores, decodificadores, etc. Entre sus ventajas se encuentran, un alto grado de estandarización comercial, relativamente bajo consumo y alta velocidad. Como en el caso de la tecnología *MOS*, la *TTL* tiene a su vez múltiples subdivisiones: *TTL*; *HTTL* (de High Power); *LTTL* (Low Power); *STTL* (Schottky) y *LSTTL* (Low Power Schottky). La subdivisión *LSTTL* es, con mucho, la más usada en la actualidad. El nivel de estandarización de los circuitos *TTL* es tan grande que virtualmente todos los fabricantes de circuitos *MOS-LST* los hacen "compatibles *TTL*", lo cual significa que manejan parámetros (voltajes, corrientes, margen de ruido) similares a los circuitos *TTL*.

Dado que el presente capítulo pretende mostrar el procedimiento de diseño de sistemas digitales basados en microprocesadores, habremos de enfocar la atención a 2 tipos de circuitos.

1. Circuitos *MOS-LST* "compatibles con *TTL*"
2. Circuitos *LSTTL*

El término "compatible con *TTL*" implica que los **VOLTAJES** máximos y mínimos que se manejan en sus entradas y salidas, son los especificados para la familia *TTL* (que incluye a todas sus subdivisiones). En la tabla 1 se muestran los valores de voltajes y corrientes manejados por los circuitos *LSTTL*, tomados de su manual de especificaciones. La fig 3 muestra la asignación de las variables para un inversor 74LS04.

## ANALISIS DE CIRCUITOS DIGITALES

El análisis electrónico de los sistemas digitales puede hacerse desde los puntos de vista estático y dinámico.

TABLA 1 = Valores estándar para circuitos *LSTTL*

Símbolo	Descripción	Valor en circuitos estándar <i>LSTTL</i>
$V_{OH}$	voltaje de salida en estado alto	2.4V. [mínimo]
$V_{OL}$	voltaje de salida en estado bajo	0.4V. [máximo]
$V_{IH}$	voltaje de entrada en estado alto	2.0V [mínimo]
$V_{IL}$	voltaje de entrada en estado bajo	0.8V [máximo]
$I_{OH}$	corriente de carga para una salida en estado alto	- 0.4 mA [máximo]
$I_{OL}$	corriente de carga para una salida en estado bajo	8 mA [máximo]
$I_{IH}$	corriente de entrada en estado alto	20 $\mu$ A [máximo]
$I_{IL}$	corriente de entrada en estado bajo	- 0.4 mA [máximo]
F.O.	Fan Out: parámetro definido como el máximo número de compuertas similares que pueden conectarse simultáneamente a la salida de otra.	20
N.M.	margen de ruido	400 mV.

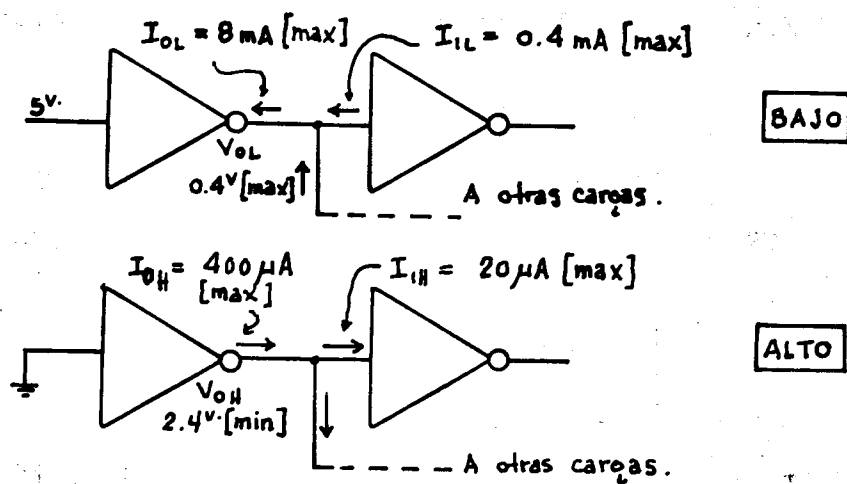


FIG 3 : VOLTAJES Y CORRIENTES EN EL CIRCUITO 74LS04

El análisis estático se enfoca sólo sobre la compatibilidad de los niveles lógicos. El análisis dinámico ve en cambio la compatibilidad de los circuitos trabajando a una cierta frecuencia de conmutación.

### **ANALISIS ESTATICO**

Nuestro objetivo ahora, es conocer si es o no posible conectar 2 o más circuitos, de los 2 tipos mencionados, sin que se presenten fallas de tipo estático en su funcionamiento.

Podemos asegurar que un circuito digital no tendrá problemas de diseño, desde el punto de vista estático, si

1. Los niveles de voltaje de salidas y entradas son compatibles
2. Las corrientes manejadas por cada salida no exceden los valores máximos especificados de fábrica

El primer punto se cumple al manejarse solo circuitos *LSTTL* y *MOS* "compatibles con *TTL*". Esto nos deja únicamente con un punto a analizar: que la corriente o carga máxima que maneja cada salida no exceda los valores máximos especificados por el fabricante. Las cargas que manejan las salidas de los circuitos pueden ser de 3 tipos.

1. La entrada de un circuito *LSTTL*
2. La entrada de un circuito *MOS* "compatible con *TTL*"
3. La salida -en tristate- de un circuito *LSTTL* o *MOS* "compatible", que también constituye una carga en esas condiciones

#### **ENTRADA LSTTL:**

Consultando la tabla 1 se puede ver, para el estado alto, una corriente de entrada de  $+20 \mu A$  como máximo. En estado bajo se tiene  $-0.4 mA$  como máximo.

#### **ENTRADA MOS "compatible TTL"**

Los circuitos *MOS* tienen una impedancia de entrada muy grande. La corriente de entrada de estos circuitos, que los fabricantes generalmente conocen en inglés como "Input Leakage Current", se ha estandarizado a  $+10 \mu A$  [max], tanto para el nivel alto como el bajo.



### **SALIDA EN 'TRISTATE':**

Como se explicó previamente, las salidas en 'tristate', aunque virtualmente desconectadas, sí representan una pequeña carga, que, dependiendo del nivel de voltaje en el bus, será de  $+20 \mu A$  para el estado alto y  $-20 \mu A$  para el bajo. Estos valores se observan generalmente tanto en circuitos *MOS* como en *LSTTL*.

Aquí es indispensable aclarar 2 puntos:

- Los valores de voltajes y corrientes hasta ahora manejados, son los que el fabricante garantiza para "el peor caso", es decir, con el circuito funcionando en las condiciones más adversas y probando un gran número de muestras. Estos valores son "máximos" ó "mínimos" dependiendo del parámetro en cuestión.
- No todos los fabricantes estandarizan sus productos a los valores aquí señalados. Para mayor seguridad, es necesario referirse a las especificaciones técnicas de cada circuito en particular.

Por otro lado, las salidas que debemos analizar pueden ser de 2 tipos:

1. La salida de un circuito integrado *LSTTL*
2. La salida de un circuito integrado *MOS* "compatible con *TTL*"

En el primer caso, la carga máxima que puede sostener la salida de un circuito *LSTTL* es de  $-400 \mu A$  en estado alto y  $8 mA$  en estado bajo, como se indica en la tabla 1.

En el segundo caso, no existen valores estandarizados para las salidas de circuitos *MOS* "compatibles con *TTL*" y cada fabricante proporciona valores diferentes: el microprocesador Z80 (Zilog), por ejemplo, especifica  $-250 \mu A$  de carga máxima en estado alto, y  $1.8 mA$  en estado bajo; la memoria 2516 de Texas Instruments especifica  $-400 \mu A$  en estado alto y  $2.1 mA$  en estado bajo.

Es indispensable entonces referirse siempre a los manuales técnicos de cada circuito para conocer sus especificaciones.

Conociendo los valores de las corrientes de carga y el valor máximo de corriente que puede manejar la salida bajo análisis, el problema se reduce ahora a calcular -por la ley de Kirchoff- la carga máxima en la salida de la compuerta que se esté examinando y compararla con el valor máximo especificado. Si el valor calculado es mayor al valor máximo, entonces es probable que el circuito falle. El análisis debe hacerse tanto para el nivel alto como para el nivel bajo, y repetirse para todas las salidas

existentes en el circuito.

En la fig 4 se muestra un ejemplo para la salida de un microprocesador Z80 conectada a 4 cargas diferentes; para los estados alto y bajo las corrientes se muestran con y sin paréntesis, respectivamente. Para la salida del microprocesador Z80, se tendría la siguiente tabla como resultado del análisis:

	Carga calculada	Carga máxima especificada
Estado alto	- 70 $\mu\text{A}$	- 250 $\mu\text{A}$
Estado bajo	430 $\mu\text{A}$	1800 $\mu\text{A}$

Y en ambos casos las cargas son menores a las máximas especificadas, y por lo tanto puede concluirse que esta salida trabajará correctamente desde el punto de vista estático.

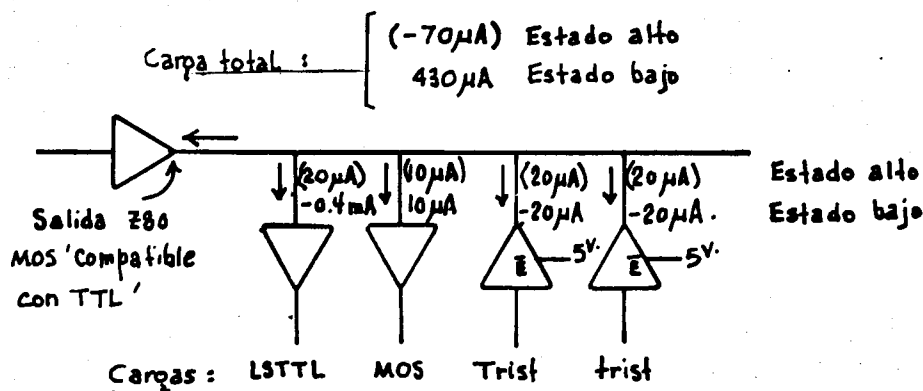


FIG 4 : EJEMPLO DE ANALISIS DE CARGAS PARA UNA SALIDA DEL MICROPROCESADOR Z80 .

I.A.C. DE INGENIERIA  
DOCUMENTACION

### *ANALISIS DINAMICO*

A diferencia del análisis estático, en el cual se tiene una estandarización considerable, en el caso dinámico es imposible fijar reglas sistemáticas de análisis.

No es lo mismo, por ejemplo, analizar el ciclo de lectura de un microprocesador a una memoria RAM, que analizar el ciclo de reconocimiento de interrupción del mismo microprocesador a un controlador determinado. Cada caso merece un estudio separado, con sus diagramas de tiempos propios.

En el capítulo 'Diseño de un sistema básico' se enfoca este tipo de análisis al procesador Z80 y la memoria 2716.

Con la práctica, este procedimiento puede hacerse extensivo a otros dispositivos.

**DISEÑO DE UN SISTEMA BASICO Z80: Descripción general.**  
 Diagrama electrónico. Análisis estático. Análisis  
 dinámico. Decodificación de direcciones. Pruebas

**DESCRIPCION GENERAL**

En el capítulo anterior se mostró el funcionamiento de una microcomputadora con sus componentes básicos. La arquitectura interna de los microprocesadores comerciales, si bien mucho más compleja que la del microprocesador básico propuesto, conserva el mismo principio de funcionamiento de éste último.

En este capítulo se presentará el desarrollo de una microcomputadora, basada en el microprocesador **Z80**, en su configuración mínima. Con el objeto de mostrar los procedimientos de diseño de éste sistema, se hará un análisis exhaustivos de su funcionamiento desde los puntos de vista estático y dinámico. Estos procedimientos de diseño pueden hacerse, sin embargo, extensivo al diseño de otros sistemas más complejos basados en un microprocesador.

El sistema básico **Z80** propuesto, cuenta con **2K EPROM**, 1 puerta de entrada y uno de salida, con posibilidad de expansión hasta 4 puertas de entrada y 4 de salida. El campo de aplicación de este sistema está reducido a pequeños sistemas de control, temporizadores, terminales remotas, etc. Dispositivos como los teclados, despliegues luminosos de 7 segmentos, LEDS, relevadores y microinterruptores se pueden conectar al sistema en forma directa.

El circuito, por su simplicidad, no cuenta con memoria **RAM**. Esto impide la utilización de las siguientes funciones:

- a) Llamados a subrutina, puesto que la dirección de retorno se guarda precisamente en **RAM**. El procesador **Z80** permite su uso equivalente mediante la instrucción **JP(HL)**, donde la dirección de retorno debe almacenarse temporalmente en el registro **HL**, antes de saltar a la subrutina, ejecutándose la instrucción indicada para regresar al programa principal.
- b) Almacenamiento de resultados parciales: si bien no es posible guardarlos en memoria **RAM**, pueden emplearse los 14 registros internos del **Z80**, los cuales son suficientes en muchas aplicaciones.

Se utiliza, también a efecto de simplificar el circuito, el mapeo de puertos de entrada/salida en memoria. Esto significa que la línea de control del **Z80** **TORQ** queda sin utilizarse, y las señales de control de los puertos de **E/S** se derivan únicamente de las di-

recciones y las señales  $\overline{RD}$  y  $\overline{WR}$ . Los puertos se utilizan entonces como si fuesen localidades de memoria, pudiendo emplearse las instrucciones  $LD A, (800H)$  ó  $LD (800H), A$  para manejarlos. En particular, las instrucciones  $SET, RES$  ó  $BIT$  del Z80 facilitan el manejo independiente de cada bit en las localidades de memoria, ó puertos.

## DIAGRAMA ELECTRONICO

El diagrama electrónico del circuito se muestra en la fig 1. Como puede verse, del procesador Z80 solo se utilizan las líneas de control  $\overline{RD}$  y  $\overline{WR}$ .  $\overline{MREQ}$  y  $\overline{IORQ}$  no se emplean por estar los puertos mapeados en memoria. Las líneas  $\overline{INT}$ ,  $\overline{WAIT}$ ,  $\overline{NMI}$  y  $\overline{BUSRQ}$  se encuentran conectadas a 5V. El inicio al Z80 se da por medio de un switch de un polo un tiro, que conecta la línea  $\overline{RESET}$  a tierra.

El circuito oscilador se forma con 2 inversores 74LS04 realimentados y conectados con un cristal de 4MHZ. A la señal de salida del oscilador, que es aproximadamente senoidal, se le da forma con otro inversor y luego se hace pasar por un FLIP FLOP 74LS74, en configuración 'TOGGLE', para darle un ciclo de trabajo del 50%. La salida de este FLIP FLOP es entonces una señal de 2MHZ que es la que alimenta el reloj del Z80.

La memoria ROM es la 2516 de Texas Instruments (equivalente a la 2716 de Intel), con tiempo máximo de acceso de 450ns. Como puerto de entrada se utiliza el driver tristate 74LS244, y como puerto de salida el registro 74LS364. El primero se habilita con la señal  $\overline{E}$ , que es activo baja. El 74LS364 almacena los datos en sus flip-flops con el flanco ascendente de la señal CP.

Las líneas de direcciones  $a_0 \dots a_{10}$  se conectan a la 2516 para dar acceso a sus 2K localidades.  $a_{11}$  se usa para diferenciar las direcciones de memoria de las direcciones de E/S, como se verá más adelante.

Para verificar el funcionamiento de cualquier circuito digital, éste debe analizarse bajo condiciones de funcionamiento estáticas y dinámicas.

En el análisis estático deben examinarse, en condiciones de nivel bajo y nivel alto, los valores de corriente de todas las salidas del circuito, de tal forma de constatar que estén dentro de las normas especificadas por el fabricante. El procedimiento no es muy complicado, si se considera que se conoce la capacidad máxima de corriente de las salidas y se conoce también el número de entradas a las que éstas están conectadas.

Por otro lado, en el análisis dinámico se considera el circuito funcionando a cierta velocidad (en este caso 2MHZ) y se examina que la transferencia de información entre microprocesador, memoria

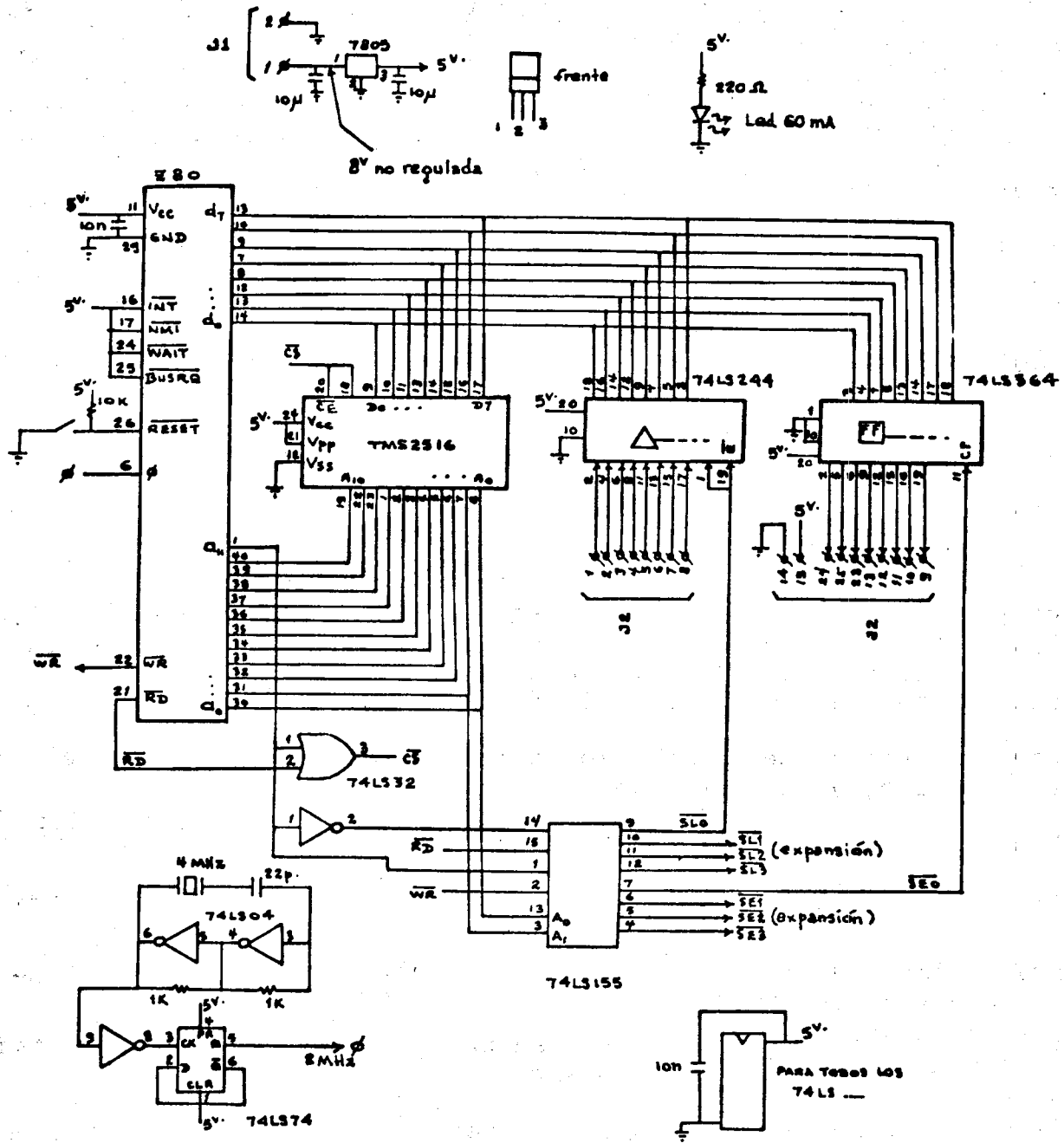


FIG 1. SISTEMA BASICO Z-80  
DIC. 1982

y puertos de entrada/salida se efectúe en los períodos especificados por el fabricante. Para esto deben compararse los diagramas de tiempos de los dispositivos para los ciclos, tanto de lectura como de escritura.

En los siguientes párrafos se explica al detalle el análisis del circuito en cuestión, tanto en el caso estático como el caso dinámico.

### Análisis estático:

En el procedimiento de análisis estático, deben examinarse las condiciones de carga de todas las salidas en cada circuito integrado, tanto en los ciclos de lectura como en los de escritura. Además, debe hacerse el análisis de cargas, para nivel alto y bajo separadamente.

El trabajo anterior puede simplificarse probando solo las salidas más susceptibles de presentar problemas, esto es, todas las salidas de los circuitos de gran escala de integración (LSI) que regularmente pueden tolerar solo hasta 4 cargas LS.

Por convención, se toman como positivas las corrientes que entran al circuito, y negativas las que salen.

Las cargas pueden ser de 3 diferentes tipos:

- (a) Entradas MOS: según especificaciones, representan una carga de + 10  $\mu A$
- (b) Salidas en tercer estado (tristate): representan una carga de + 20  $\mu A$  en estado alto, y - 20  $\mu A$  en estado bajo
- (c) Entradas TTL LS: representan una carga de - 0.4 mA en estado bajo, y 20  $\mu A$  en estado alto

El valor de los parámetros arriba descritos puede verificarse en los manuales de operación de los circuitos correspondientes. Conocidas las cargas que intervienen, debe hacerse, aplicando las leyes de Kirchoff, una suma de corrientes en cada salida examinada. Por ejemplo, tomando una salida del bus de datos del Z80 durante el ciclo de escritura, se tiene, para los niveles alto y bajo, los diagramas de la fig 2. En estos diagramas se incluyen las cargas para el peor de los casos, es decir, incluyendo los puertos previstos para expansión. Recuérdese que durante los ciclos de escritura, los puertos de entrada permanecen en estado de 'tristate'.

Además, en el Z80 y la TMS 2516 no debe olvidarse que el bus de datos es bidireccional, por lo que deben considerarse, en su caso, la carga que representa su entrada ó su salida en 'TRISTATE'. En el diagrama de la fig 2 se incluyen las cargas de  $10 \mu A$  de las entradas del bus de datos del Z80 y la TMS2516, así como la carga que representa la salida 'tristate' de la memoria TMS2516.

Realizando la suma de corrientes en nivel bajo y alto resultan, a la salida del Z80,  $1.68 mA$  y  $- 200 \mu A$ , respectivamente. Estos valores son inferiores a los máximos especificados por el fabricante, que son  $1.8 mA$  en estado bajo y  $- 250 \mu A$  en estado alto, por lo que el acoplamiento está dentro de normas.

El procedimiento puede repetirse para el resto de las salidas, como serían las salidas del bus de datos de la memoria 2516-durante el ciclo de lectura -, y las salidas de los circuitos 74LS244, cuando son habilitadas al hacer lectura de algún puerto de entrada, así como las líneas de dirección y control del Z80. Debe procurarse ver directamente los manuales de especificaciones técnicas de los circuitos en cuestión para obtener la capacidad máxima de sus salidas, ya que varían de circuito en circuito. Por ejemplo la memoria TMS2516 tolera en sus salidas unas cargas máximas de  $2.1 mA$  a nivel bajo, y  $- 400 \mu A$  a nivel alto, las cuales son claramente mayores que las mencionadas para el Z80. ( $1.8 mA$  y  $- 250 \mu A$ )

### ACOPLAMIENTO DINAMICO

El análisis dinámico debe hacerse a partir de los diagramas de tiempos de los circuitos. Como la 2516 es una EPROM, debe hacerse la verificación solo para el ciclo de lectura.

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 tendrá los datos disponibles mucho antes que el Z80 haga el muestreo del bus de datos, por tanto 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.

### 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.



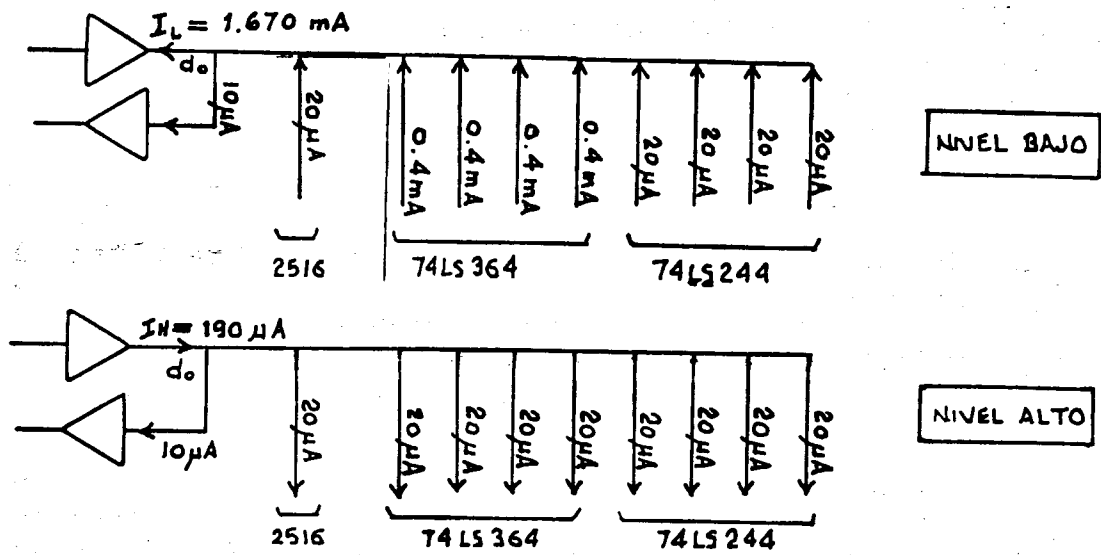


FIG 2 . CALCULO DE CARGAS DE UNA LINEA DEL BUS DE DATOS DEL Z80

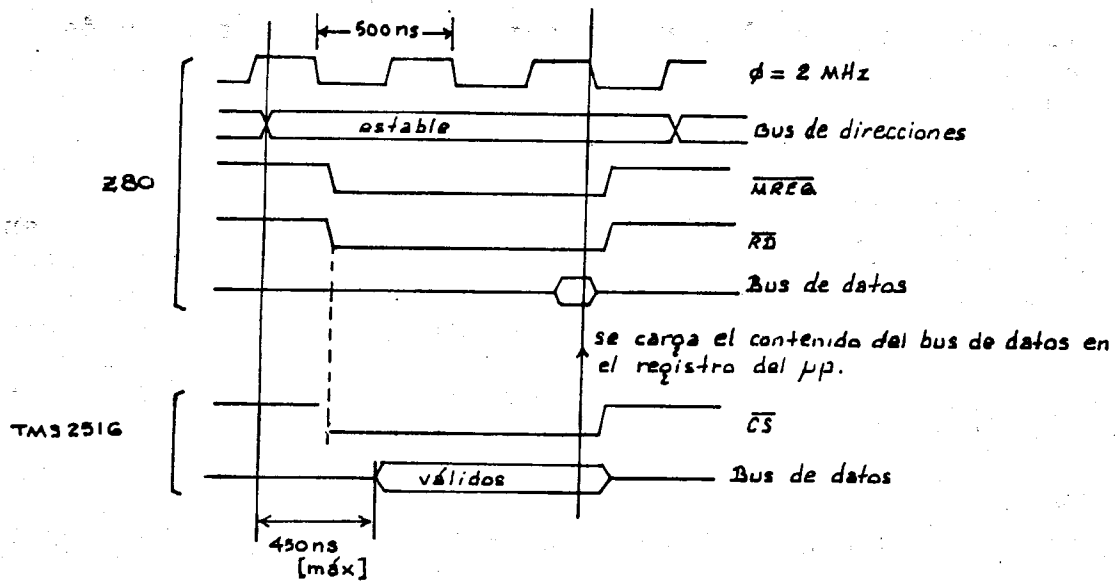


FIG 3 COMPATIBILIDAD DURANTE EL CICLO DE LECTURA DEL  $\mu\text{p}$  Z80 Y LA PROM TMS2516

que la señal de lectura del 74LS244 sea activo baja, y que la señal de carga del 74LS364 tenga 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).

## 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  $A_{11}$  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			Puertos de entrada y salida
0 8 0 3 H			

pueden emplearse idénticas direcciones para puertos de entrada y salida porque las señales  $\overline{RD}$  ó  $\overline{WR}$  garantizan el acceso sólo a uno de ellos.

## 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 mini-interruptor. El circuito, aunque sencillo, permite simular pequeños sistemas de control de procesos en donde se operen válvulas, alarmas y otros dispositivos que puedan manejarse digitalmente.

FAC. DE INGENIERIA  
DOCUMENTACION

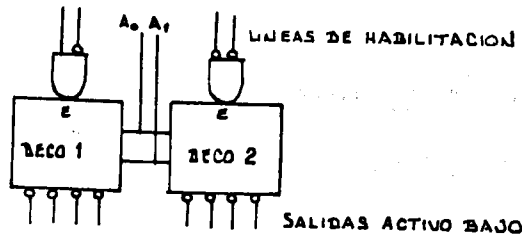


FIG 4. DIAGRAMA DE BLOQUES DEL CIRCUITO 74LS155

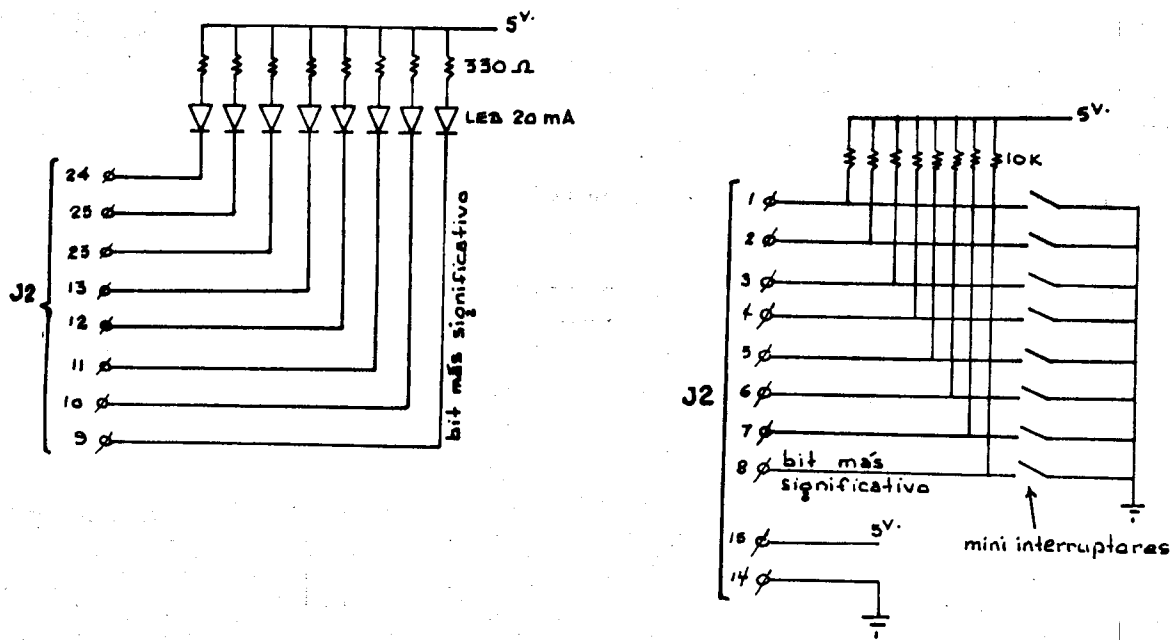


FIG 5. MODULO DE E/S PARA SISTEMA BASICO Z80

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 asíncrona 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

## 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 entre 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 curce de las líneas c-f.



ntificación por hardware:

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 multi lexor 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 caracter 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.

##### Conversor Digital Analógico (DAC):

El conversor 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 ésta 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).

#### Conversor de Rastreo:

Como su nombre lo indica, el conversor 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 conversor 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 conversor 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 estandar 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 éstas 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, vóltmetros, 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

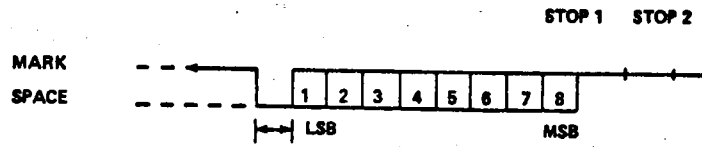


FIG.1.1 TRANSMISION SERII. DEL TELETIPO

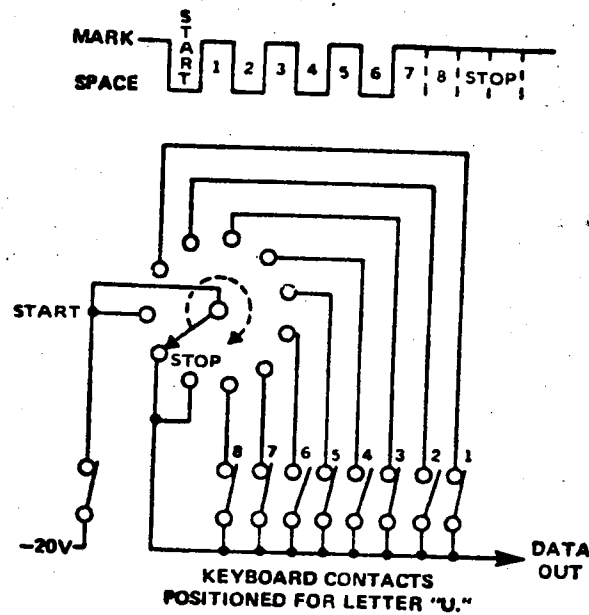


FIG.1.2 FUNCIONAMIENTO DEL DISTRIBUIDOR

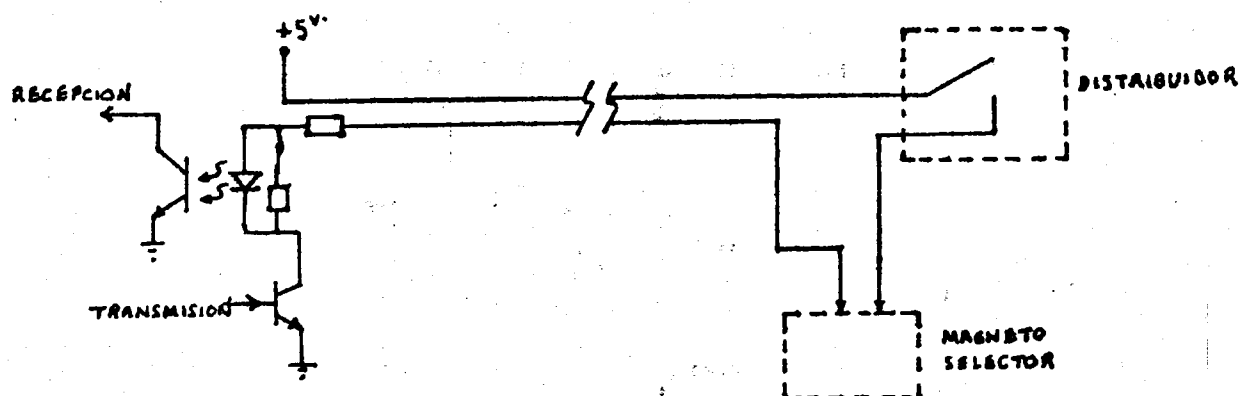


FIG.1.3 INTERFAZ HALF DUPLEX 20 MA

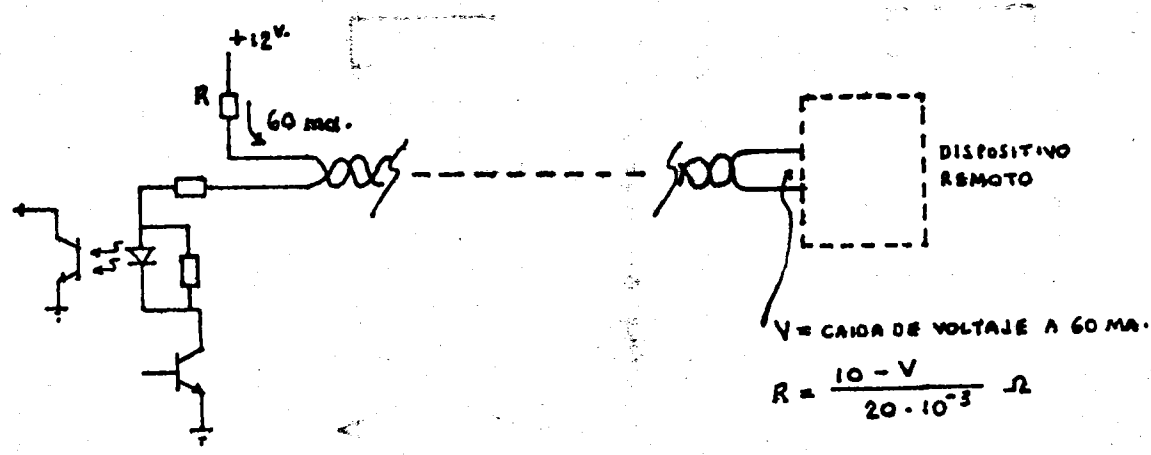


FIG.1.4 INTERFAZ HALF DUPLEX 60 MA

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
        RAL ; MOVE CARRY TO A(0)
MORE: OUT 2 ; SEND 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 NO WAIT STATES)
DELAY: MVI D,6
DLO: MVI E,2000
DL1: DCR E ; 1.5 MSEC
        JNZ DL1 ; INNER LOOP
        DCR D
        JNZ DL1
        RET
    
```

FIG.1.5 RUTINA DE SALIDA. TRANSMISION SERIE

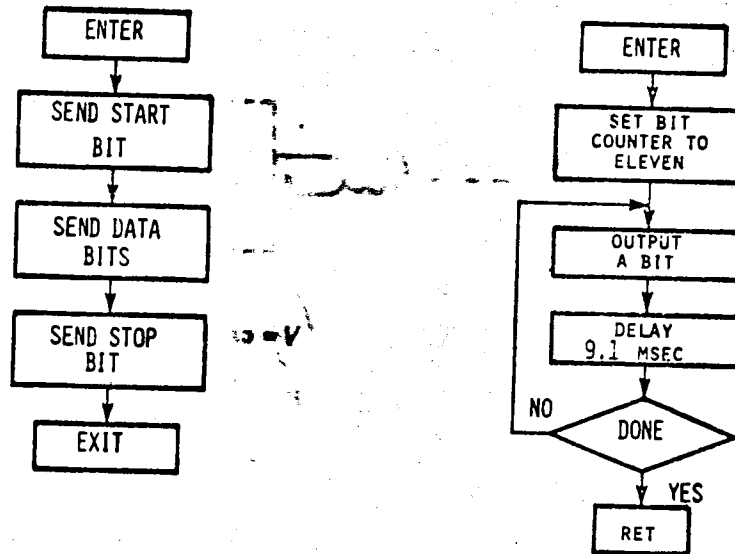
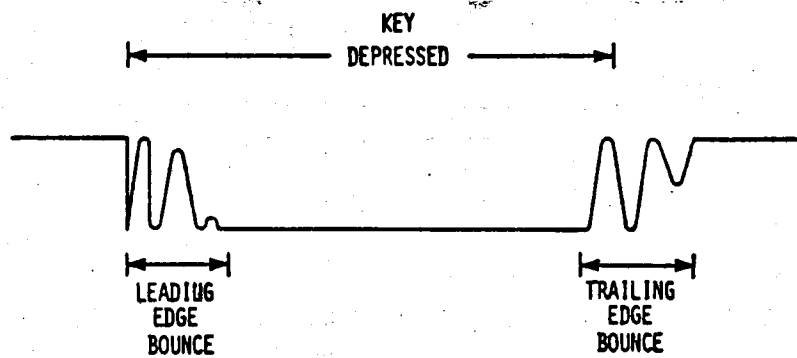


FIG.1.6 DIAGRAMAS DE FLUJO RUTINAS DE ENTRADA Y SALIDA.



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

FIG.2.1 REBOTE

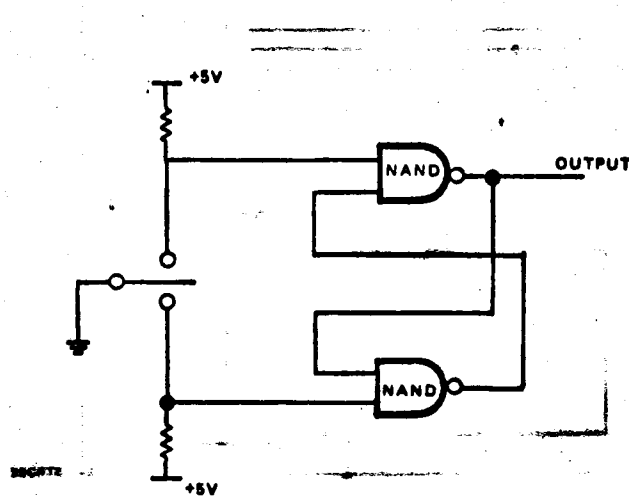


FIG.2.2 ELIMINACION DEL REBOTE POR HARDWARE

SAC DE INGENIERIA  
DOCUMENTACION

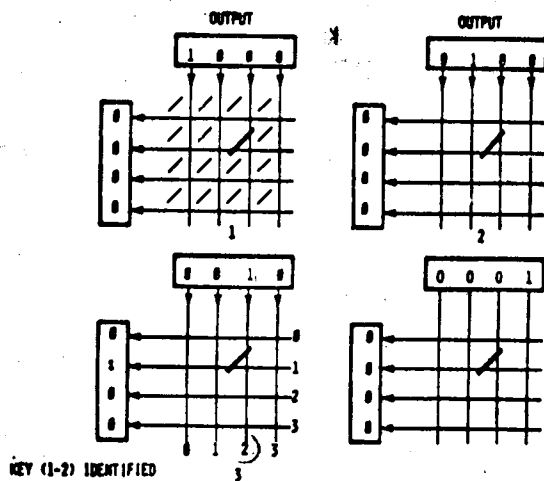


FIG.2.3 IDENTIFICACION POR PROGRAMA

ETC...

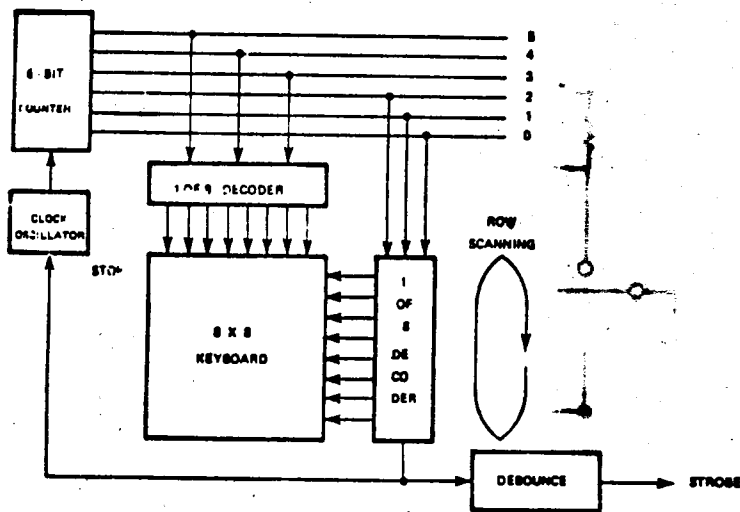
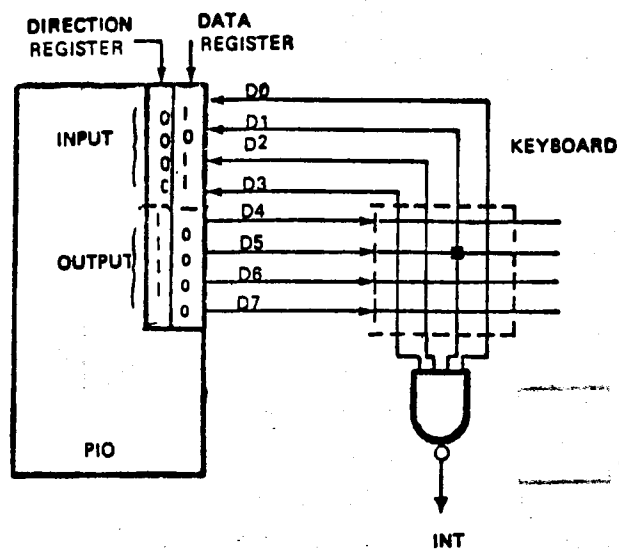
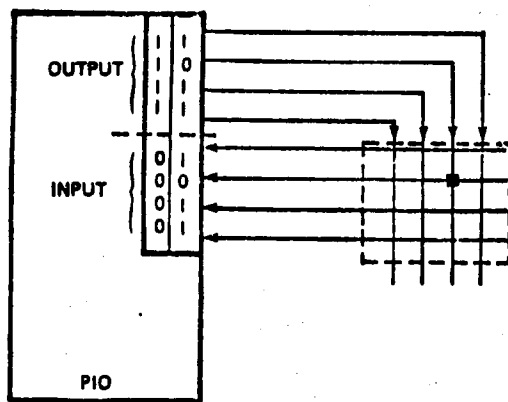


FIG. 2.4 IDENTIFICACION CON LOGICA ALAMBRADA



(1)



(2)

FIG. 2.5 IDENTIFICACION POR INVERSION DE LINEAS



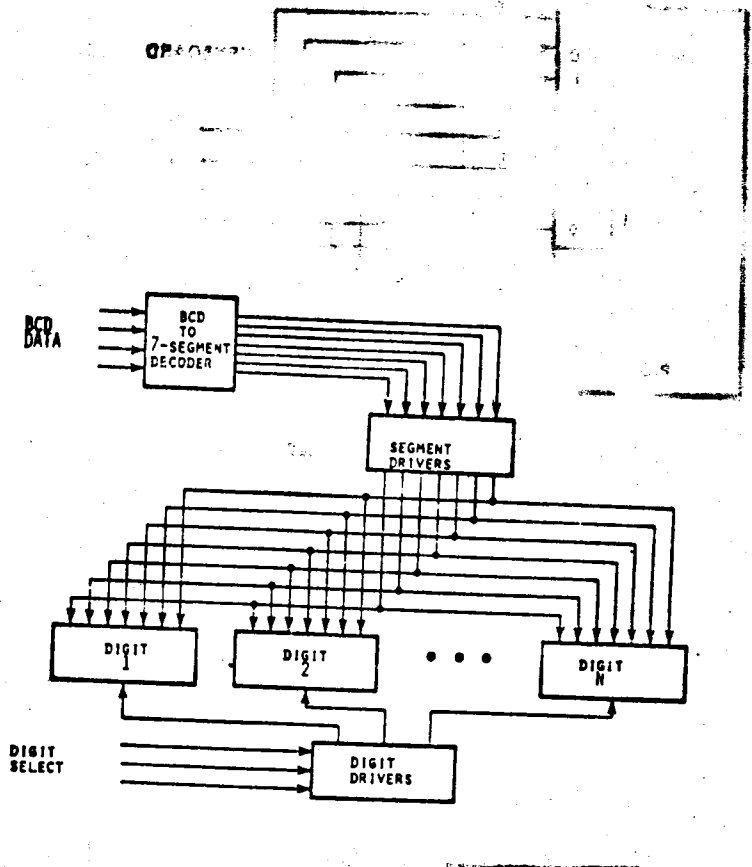


FIG. 3.1 MULTIPLEXAJE PARA DISPLAY 7 SEGMENTOS

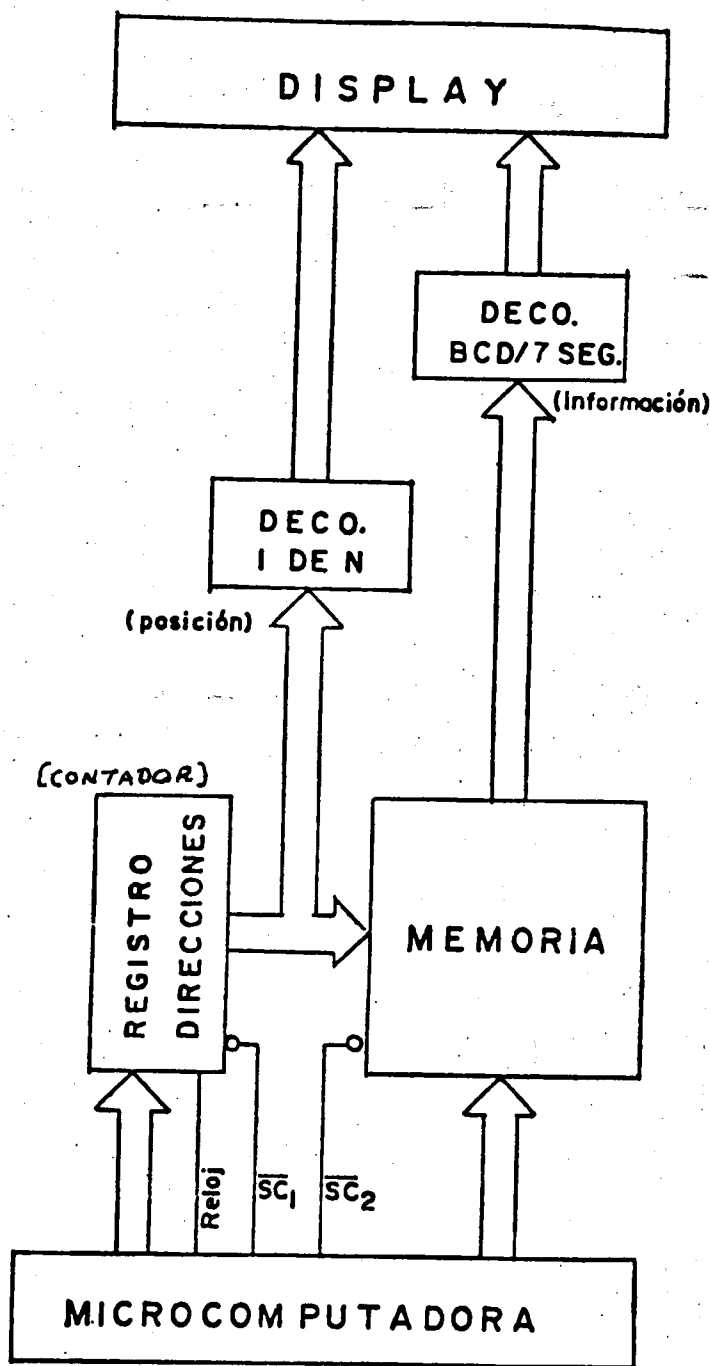


FIG. 3.2 ESCRITURA DE NUEVOS DATOS

5 COLUMNAS

7 RENGLONES

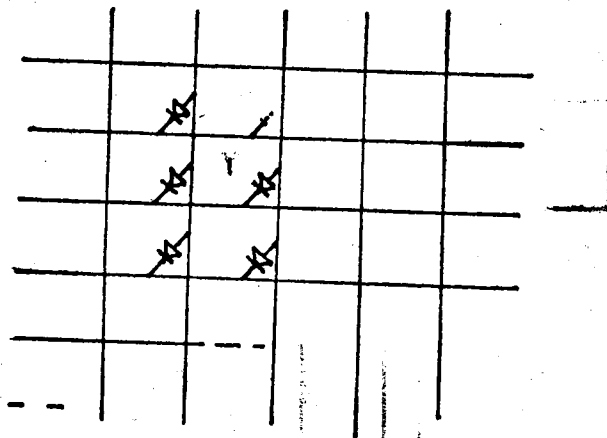


FIG. 3.3 MODULO DE MATRIZ DE PUNTOS LED 7 x 5

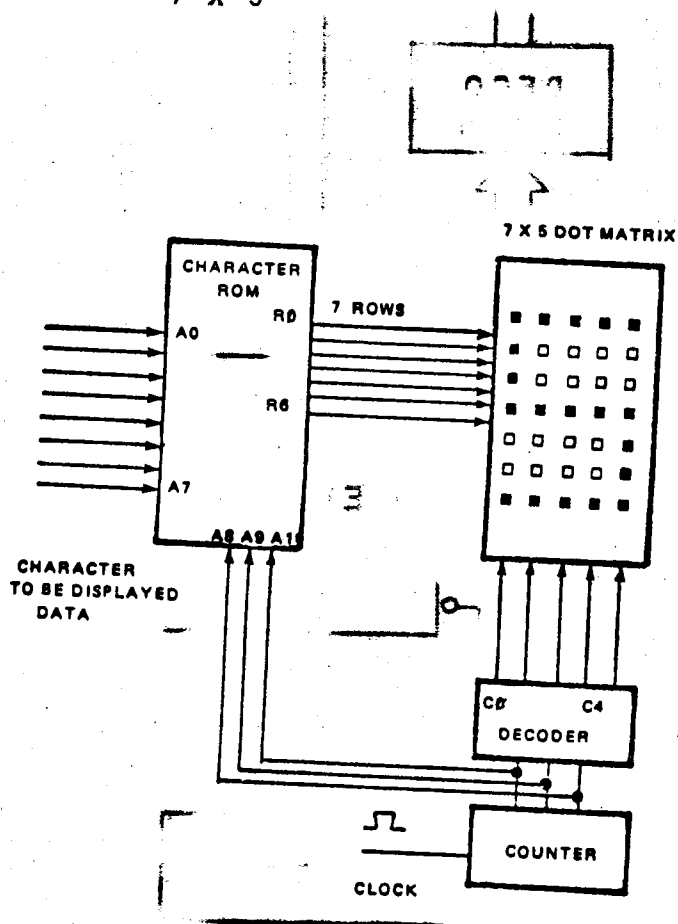


FIG. 3.4 GENERADOR DE CARACTERES

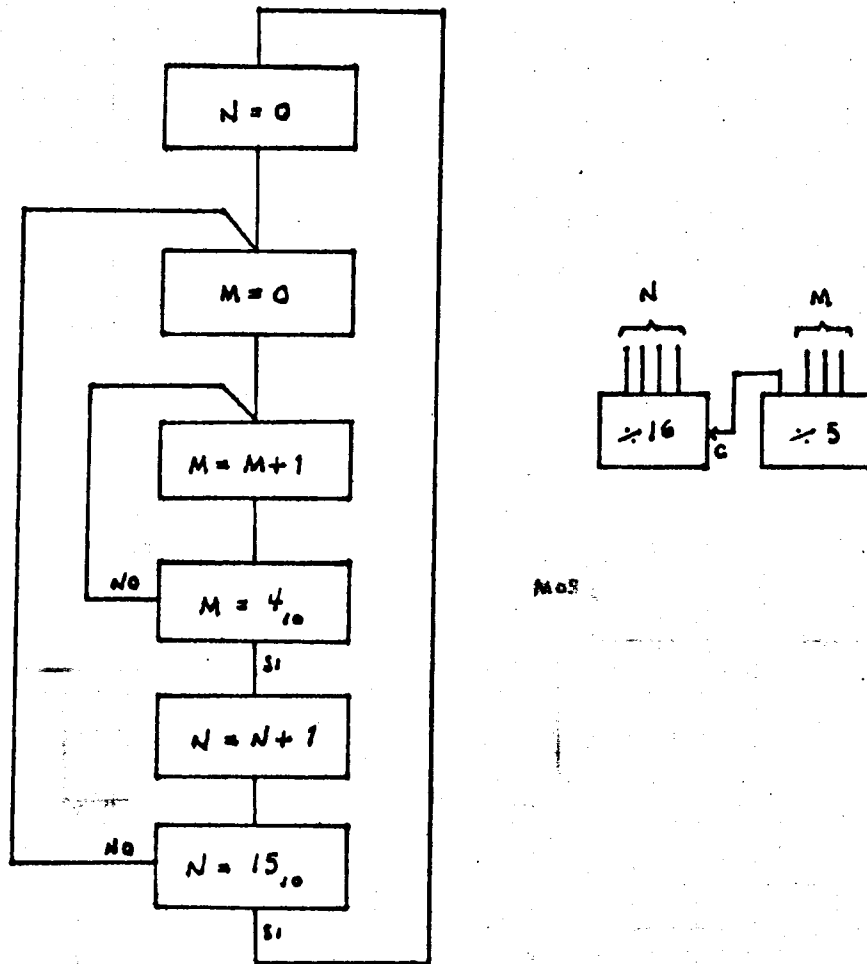


FIG. 3.5 DIAGRAMA DE FLUJO DE BARRIDO DE VARIOS MODULOS MULTIPLEXADOS.

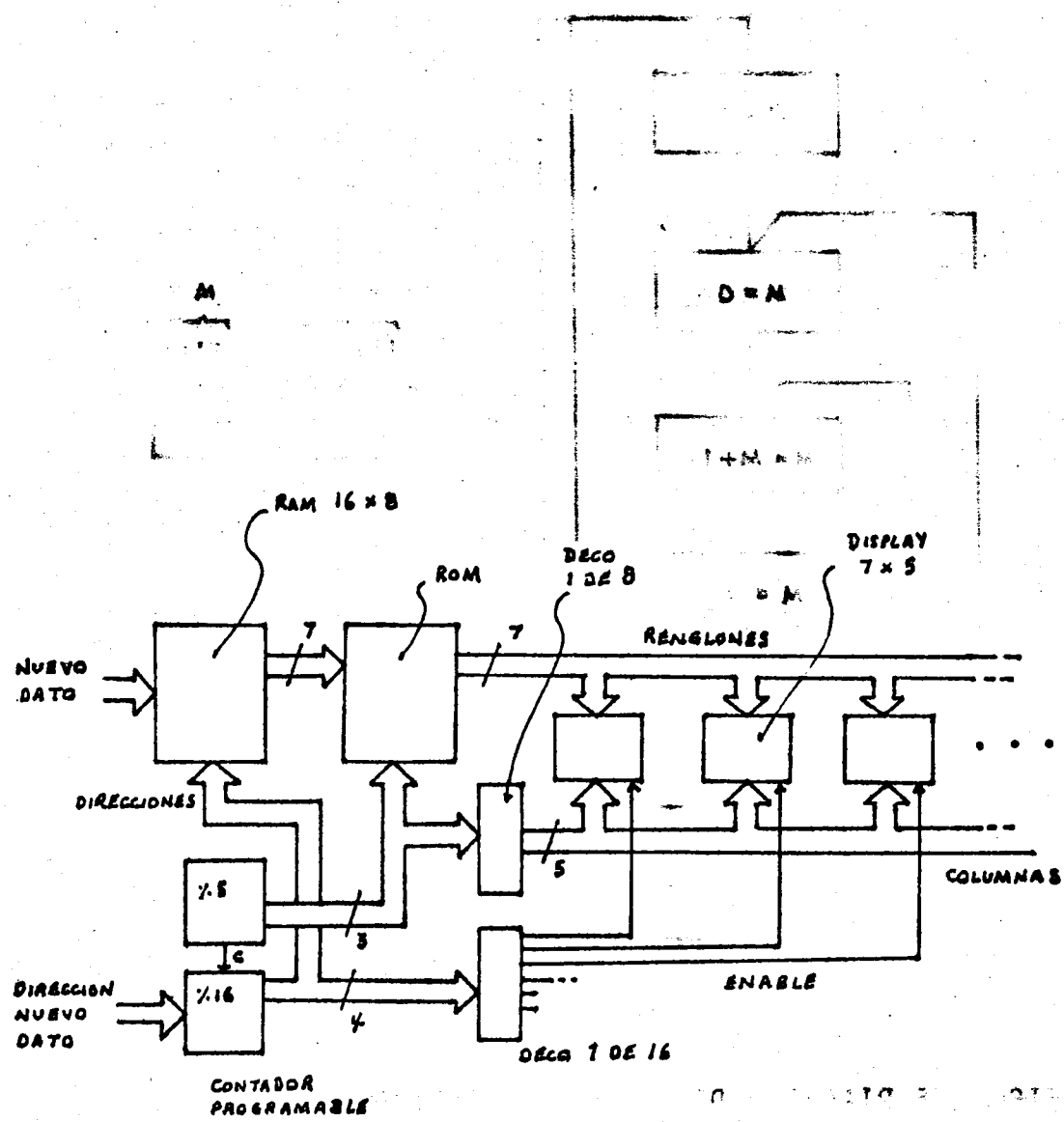


FIG. 3.6 MULTIPLEXAJE PARA DISPLAY MATRIZ DE PUNTOS

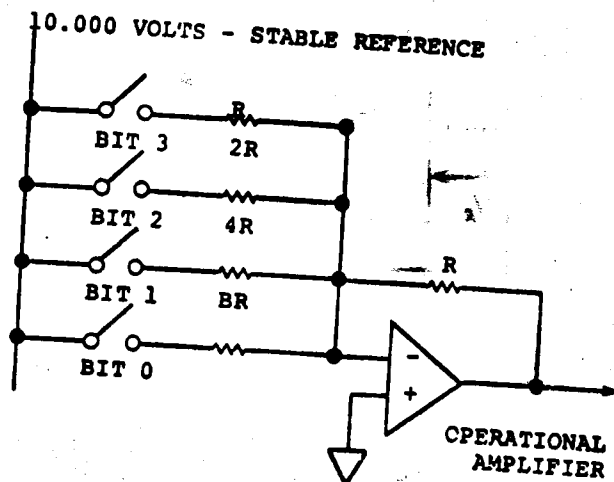


FIG. 4.1 CONVERSOR D-A SIMPLE

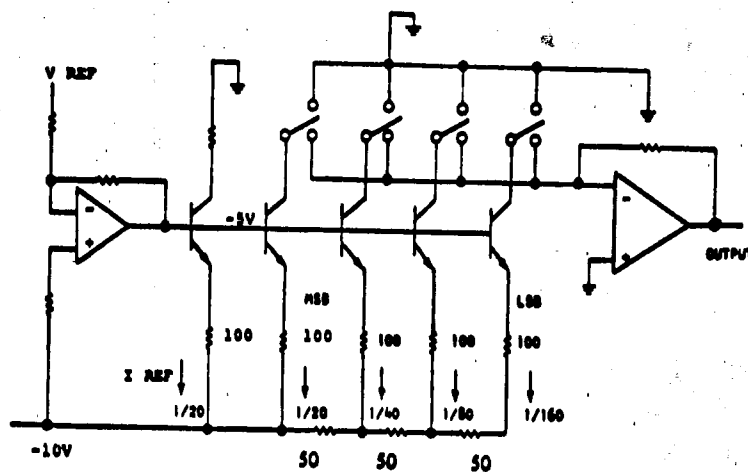


FIG. 4.2 DIAGRAMA DE UN DAC INTEGRADO

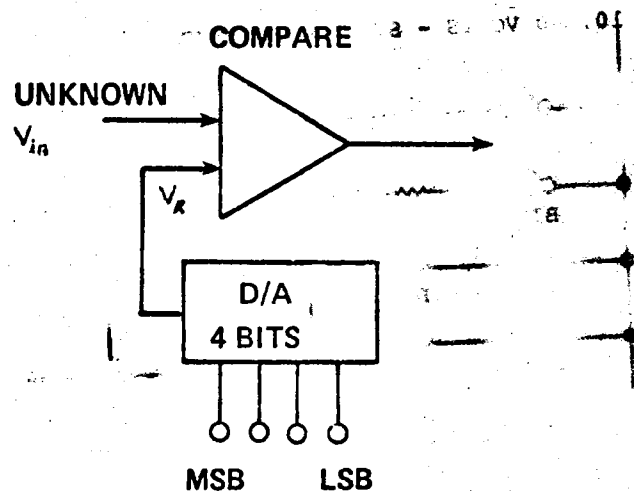


FIG. 4.3 BLOQUES BASICOS DEL ADC

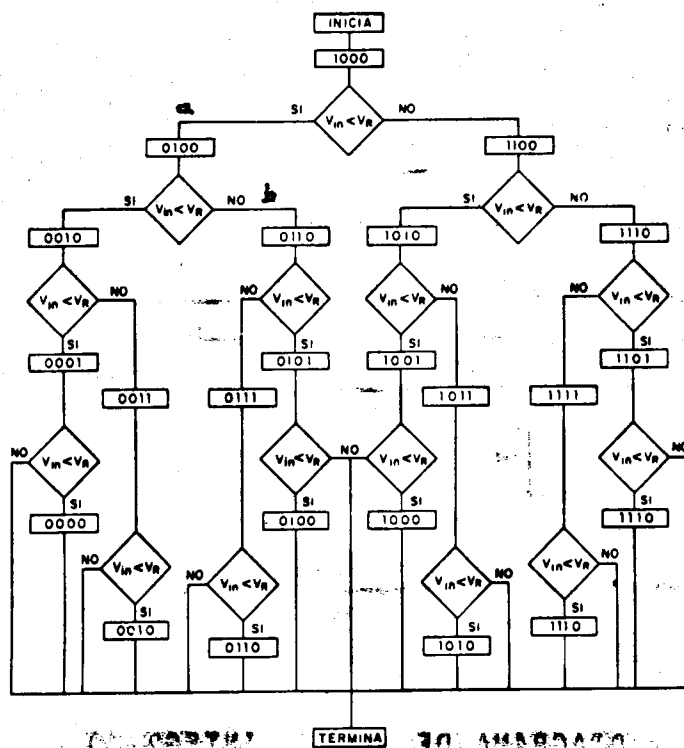


FIG. 4.4 DIAGRAMA RUTINA DE APROXIMACIONES SUCCESIVAS

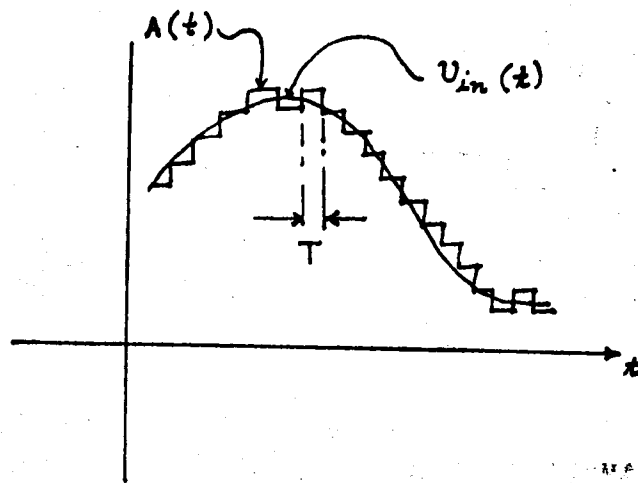


FIG. 4.5 VOLTAJES EN EL CONVERTOR SEGUIDOR

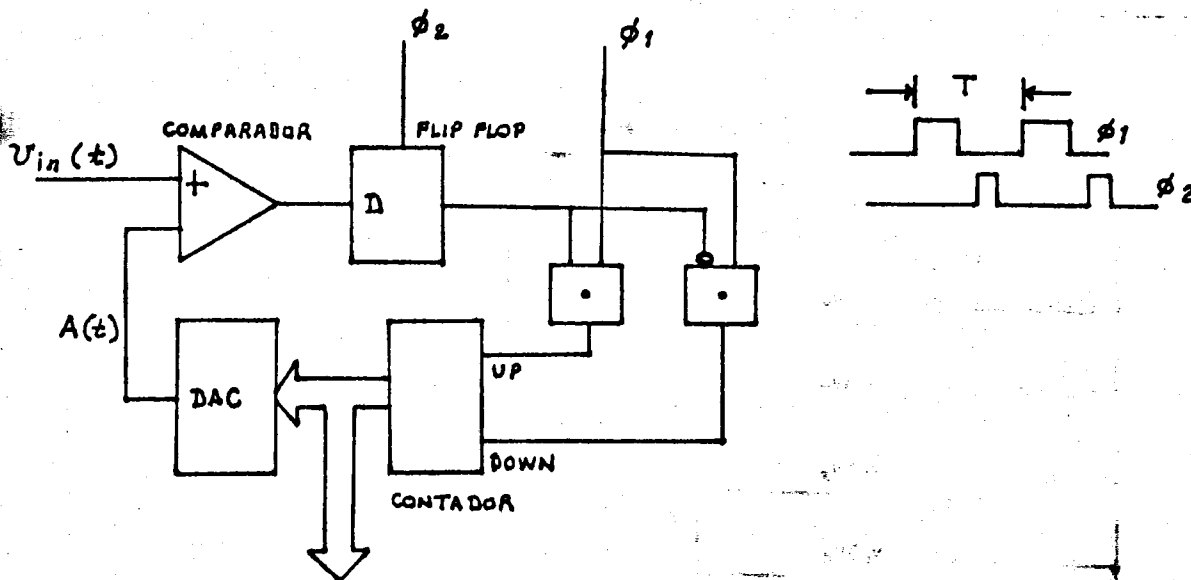


FIG. 4.6 CONVERTOR SEGUIDOR (TRACKING)



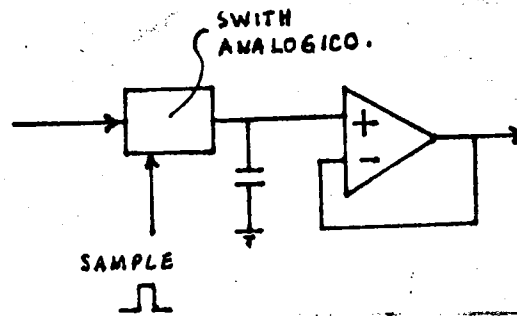


FIG. 4.7 SAMPLE &amp; HOLD

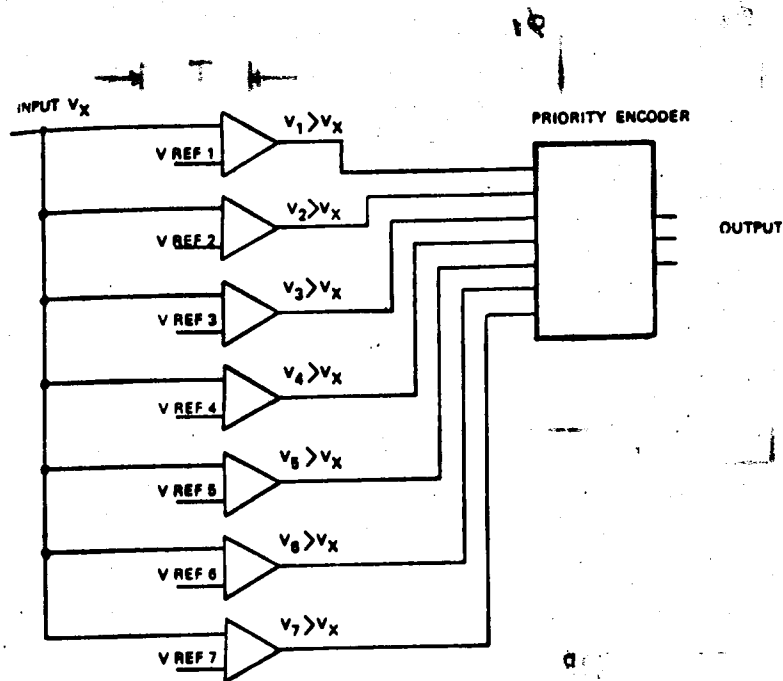


FIG. 4.8 COMPARACION DIRECTA

- GROUND
- XMIT DATA (TO COM EQUIPMENT)
- REC DATA (FROM COM)
- REQUEST TO SENT (TO COM)
- CLEAR TO SEND (FROM COM)
- DATA SET READY (FROM COM)
- DATA TERMINAL READY (TO COM)
- RING INDICATOR (FROM COM)
- RECEIVED LINE SIGNAL DETECTOR (FROM COM)
- SIGNAL QUALITY DETECTOR (FROM COM)
- DATA RATE SELECTOR (TO COM)
- DATA RATE SELECTOR (FROM COM)
- TRANSMITTER TIMING (TO COM)
- TRANSMITTER TIMING (FROM COM)
- RECEIVER TIMING (FROM COM)
- ♦ SECONDARY DATA AND REQUESTS

FIG. 5.1 SEÑALES DEL ESTANDAR RS 232-C

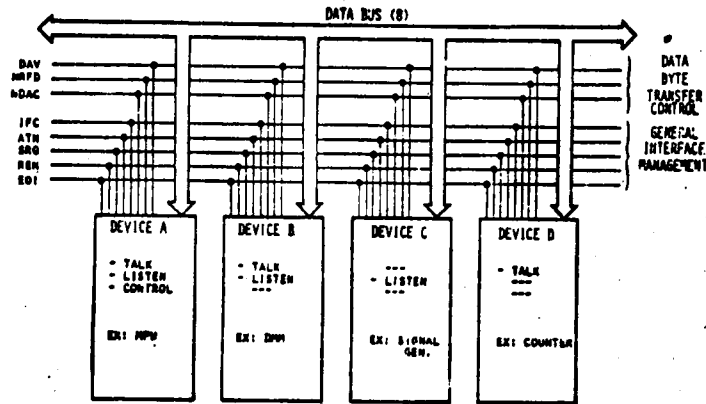


FIG. 5.2 ESTRUCTURA DEL ESTANDAR IEEE-488

FIG. 2.1. ESTÁNDAR

# INDICADORES VISUALES FOSFORESCENTES

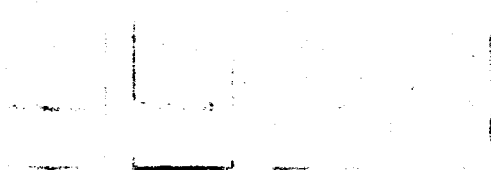


FIG. 2.2. ESTÁNDAR

## 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 termiónica (bulbos).

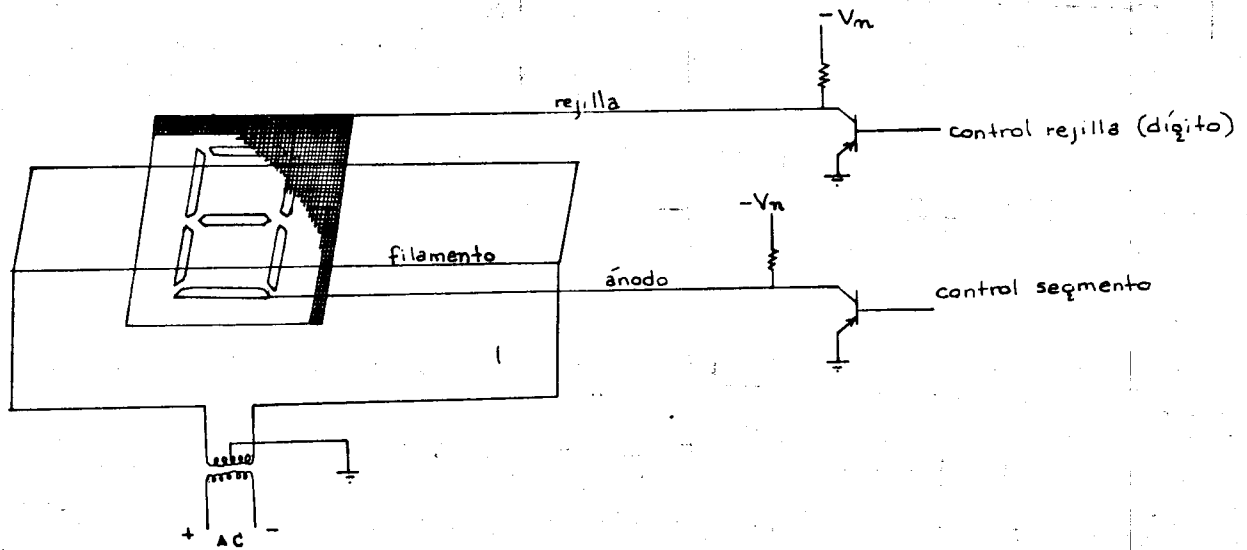
El filamento se calienta a una temperatura de aprox. 700°C. A esta temperatura emite suficientes electrones para proporcionar una luminosidad conveniente.

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).

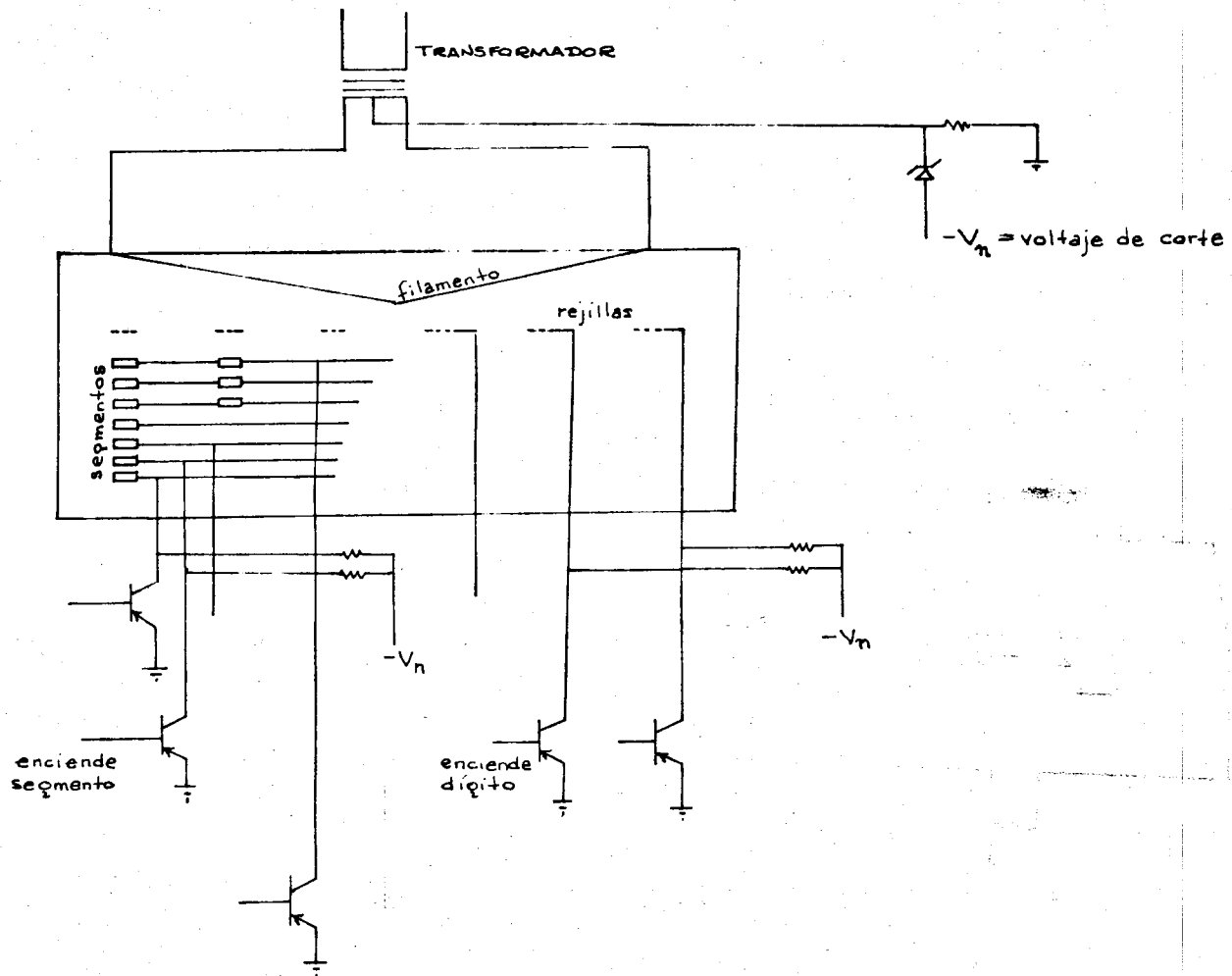
Cuando no existe diferencia de potencial en la rejilla y ánodo con respecto al cátodo, algunos electrones tienen, sin embargo, energía suficiente para alcanzar el ánodo y hacen brillar el indicador levemente.

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 DIGITO



CONTROL MULTIPLEXADO DE VARIOS DIGITOS

60. El control de video se realiza mediante un controlador de video que recibe las señales de video de los fabricantes de video y las envia a los fabricantes de video. El controlador de video es un dispositivo que permite a los fabricantes de video controlar el video de los fabricantes de video. El controlador de video es un dispositivo que permite a los fabricantes de video controlar el video de los fabricantes de video.

**CONTROLADOR DE VIDEO**

Este controlador de video es un dispositivo que permite a los fabricantes de video controlar el video de los fabricantes de video. El controlador de video es un dispositivo que permite a los fabricantes de video controlar el video de los fabricantes de video. El controlador de video es un dispositivo que permite a los fabricantes de video controlar el video de los fabricantes de video.

**JUAN B. MARTINEZ**

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

-iv-



## 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 fosforecentes (ver tema 'Indicadores visuales fosforecentes') 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 ima gen 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 frecuen cia 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 nones, mientras las líneas pares quedarán siempre en negro. De esta forma se 'refrescan' a una frecuencia de 60 HZ, solo las líneas nones 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 caracter.

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 caracteres x (8 líneas/caracter) = 192 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 caracter deseado. Dicho caracter 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 caracter 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.

## 8. 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 o 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 períodos 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.

## 10. 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 caracter. Sin embargo, funciones como el control del cursor y el 'scrolling', puede realizarlas 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 - Rodnay Laks,  
SYBEX

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

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

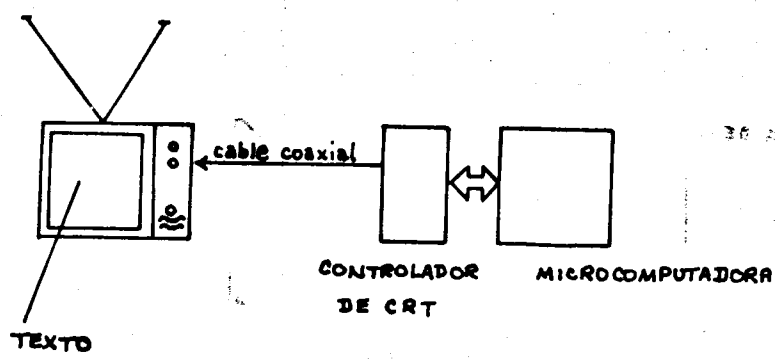


FIG 1 : ACOPLAMIENTO TV - MICROCOMPUTADORA

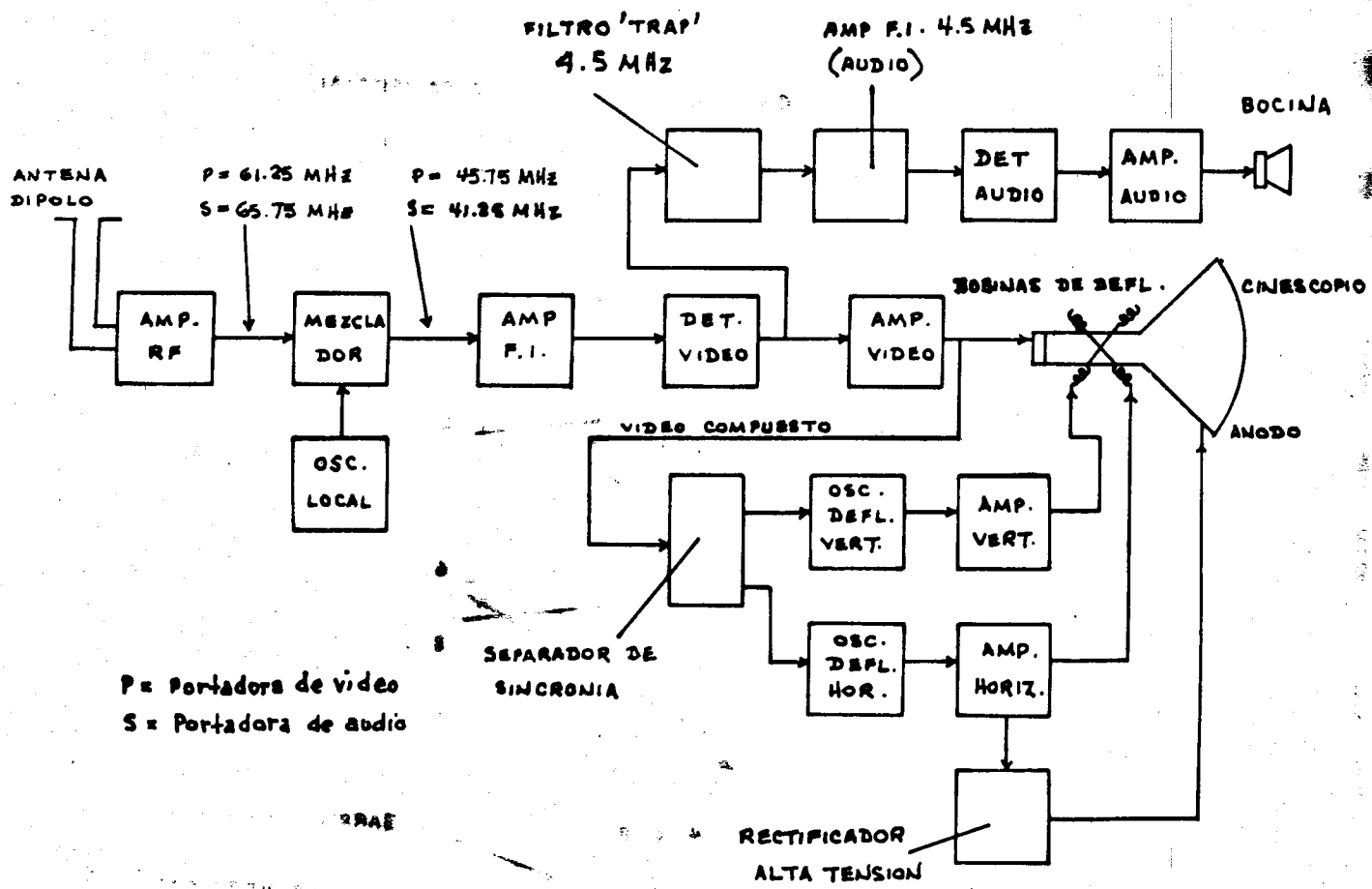


FIG 2 : DIAGRAMA DE BLOQUES DE UNA TELEVISION COMERCIAL.



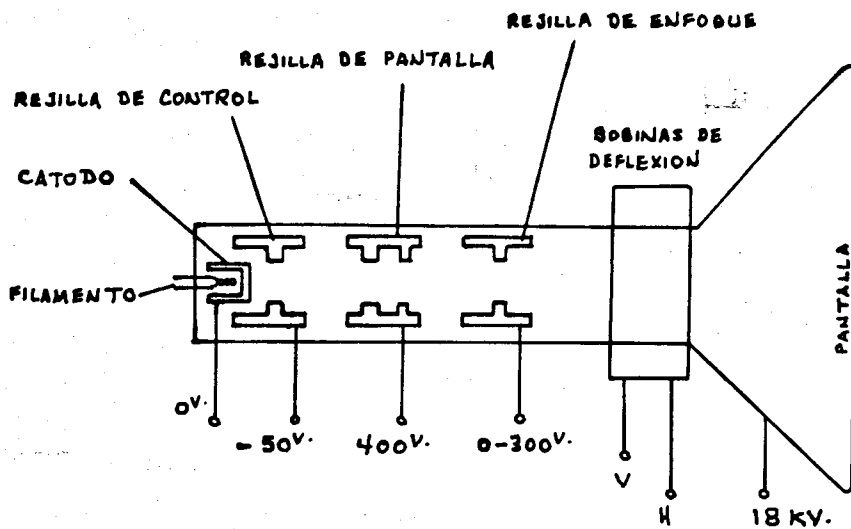


FIG. 3 : CINESCOPIO DE UNA TV. COMERCIAL

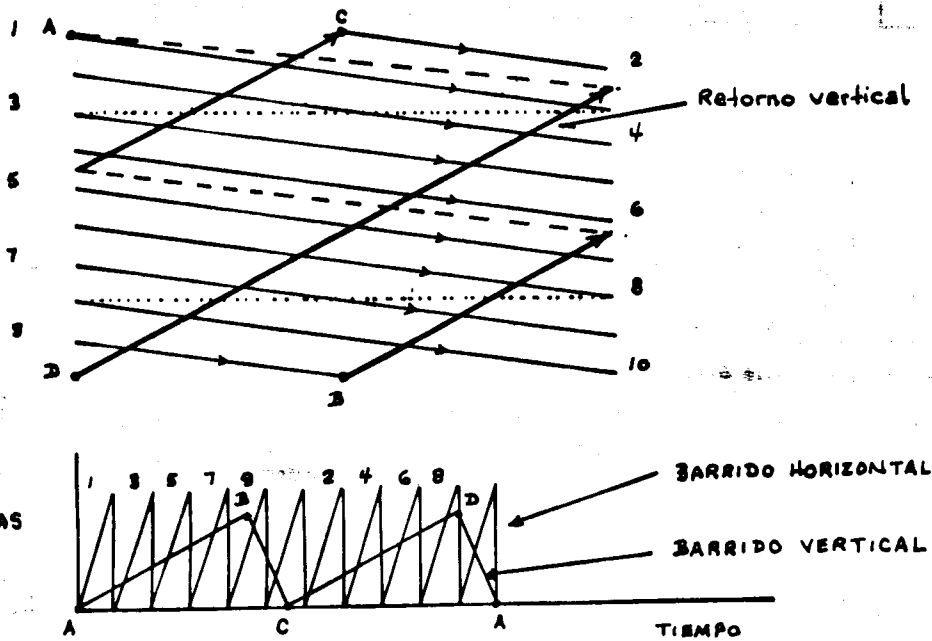


FIG 4 : FORMACION DE UNA IMAGEN ENTRELAZADA.

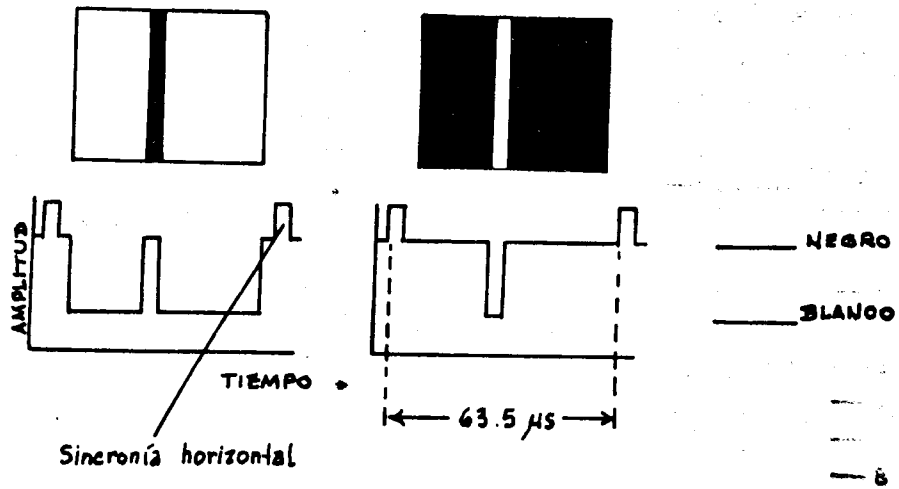


FIG 5 : SEÑAL DE VIDEO COMPUESTO Y SU CORRESPONDIENTE IMAGEN

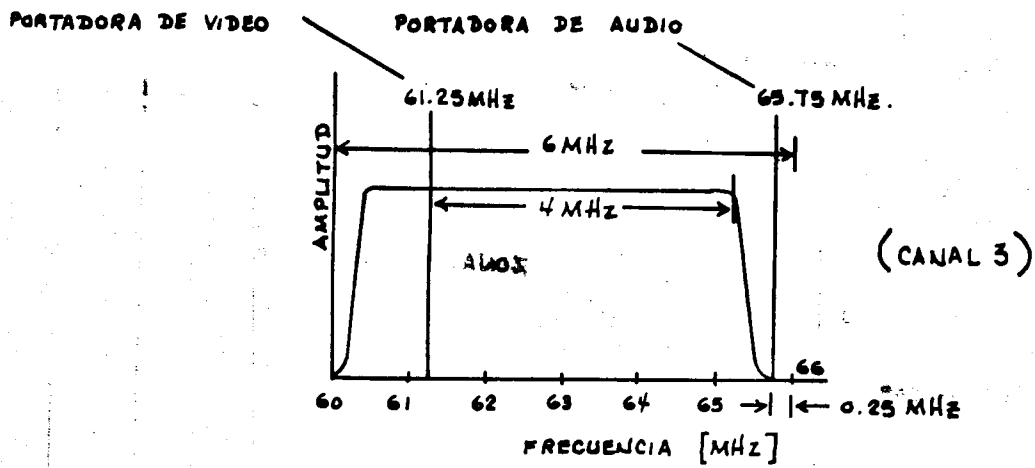


FIG 6 : CANAL ESTANDAR PARA TRASMISION POR TELEVISION.

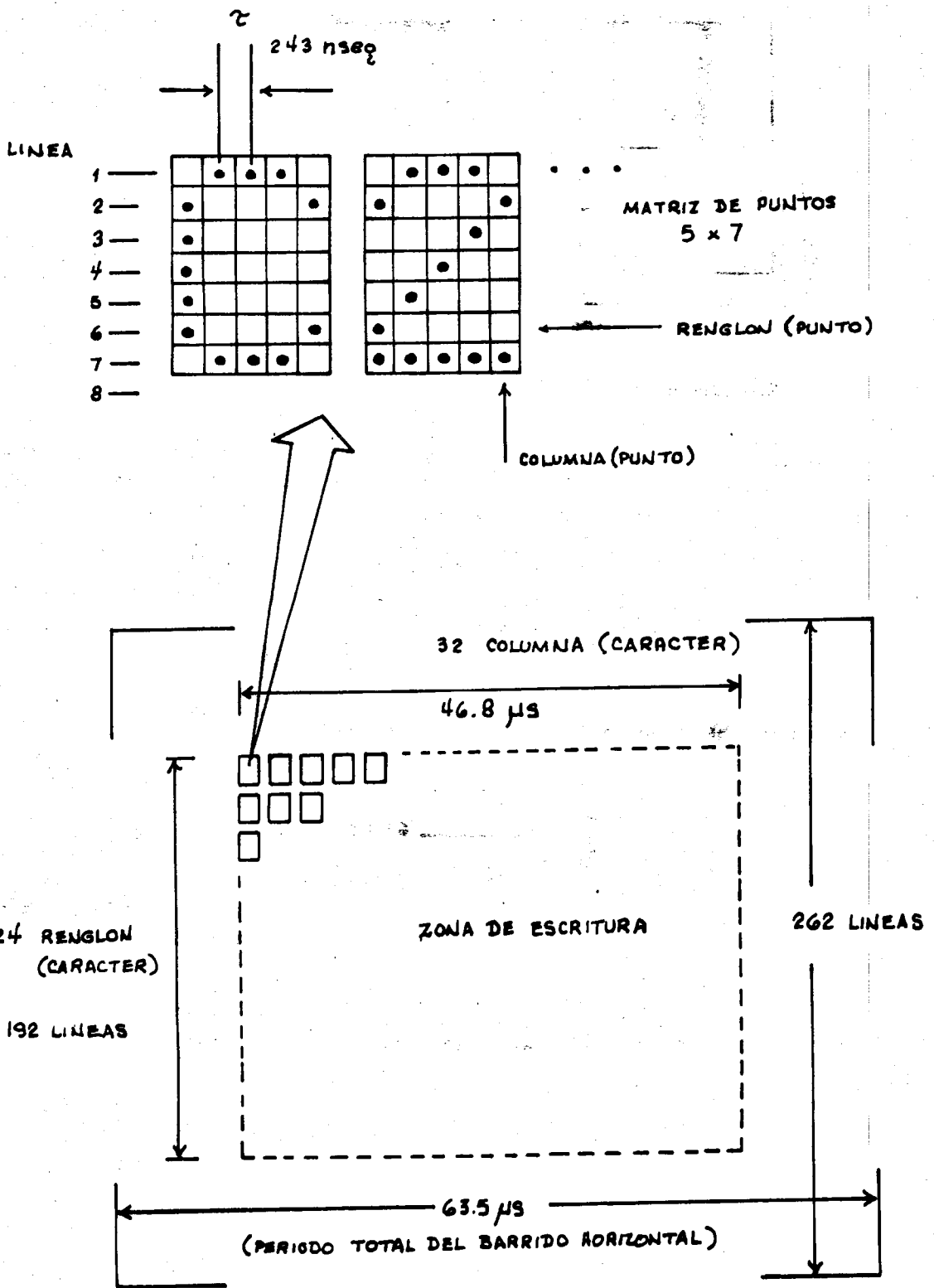


FIG 7 : FORMATO DE LA IMAGEN PARA ESCRITURA DE TEXTO .

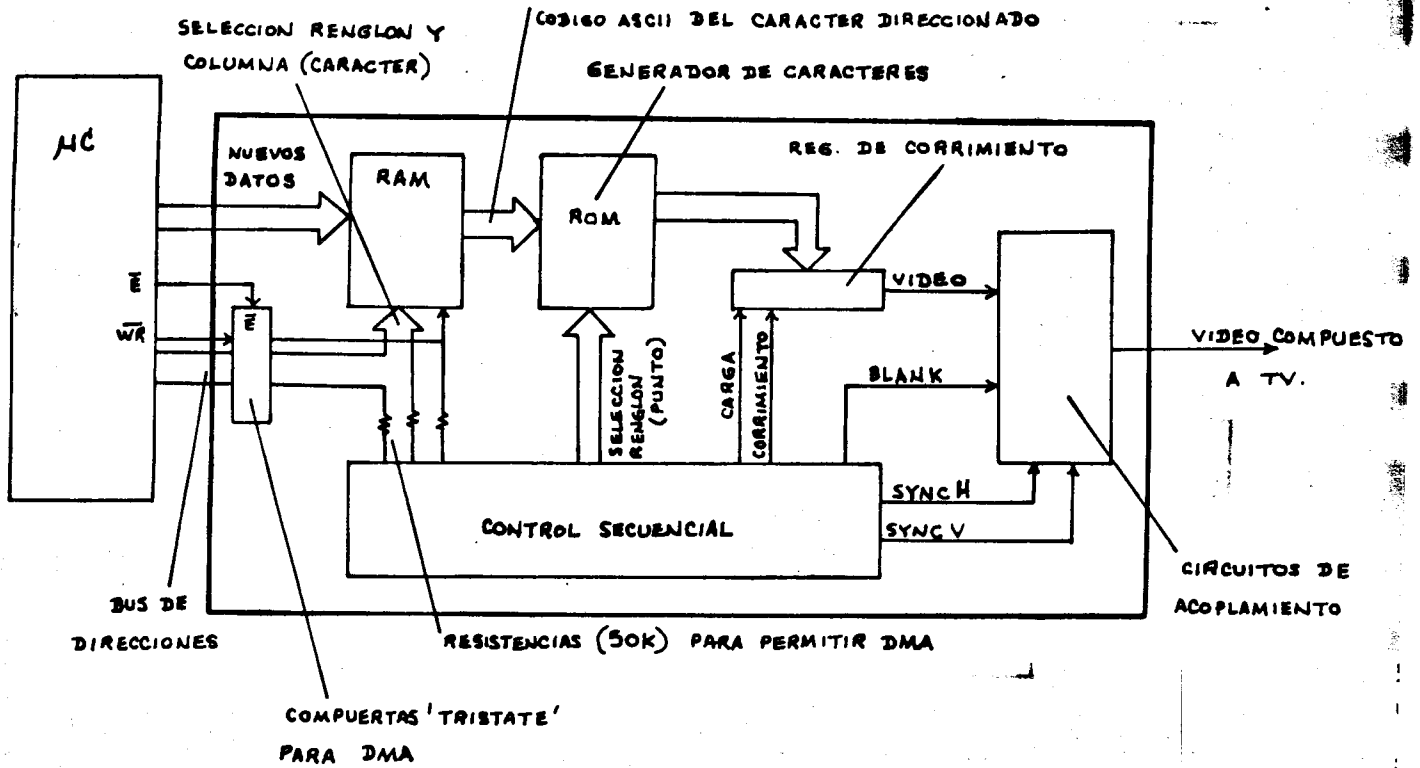


FIG 8 : DIAGRAMA DE BLOQUES DEL CONTROLADOR DE VIDEO.

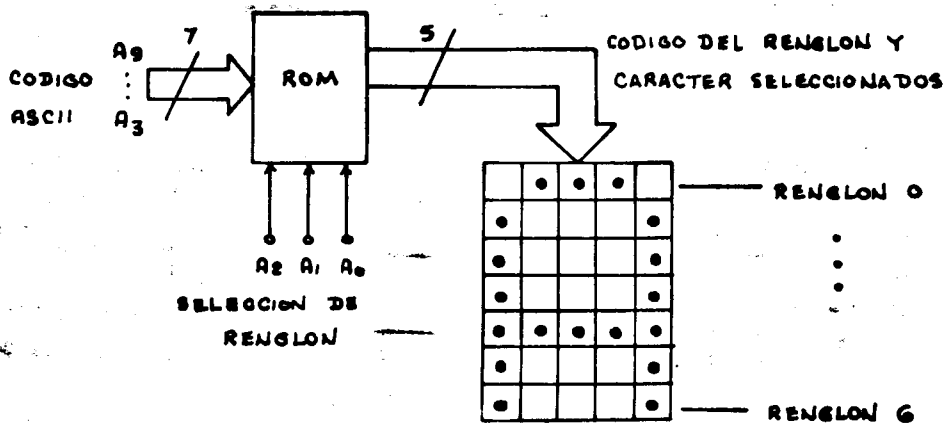


FIG 9 : GENERADOR DE CARACTERES

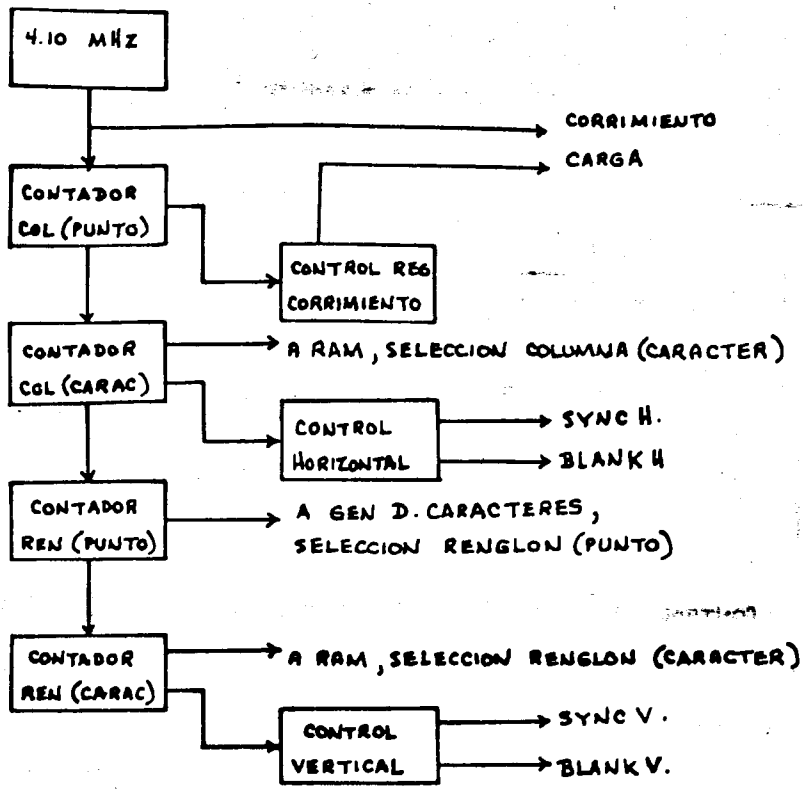


FIG 10 : CONTROL SECUENCIAL PARA EL CONTROLADOR DE VIDEO

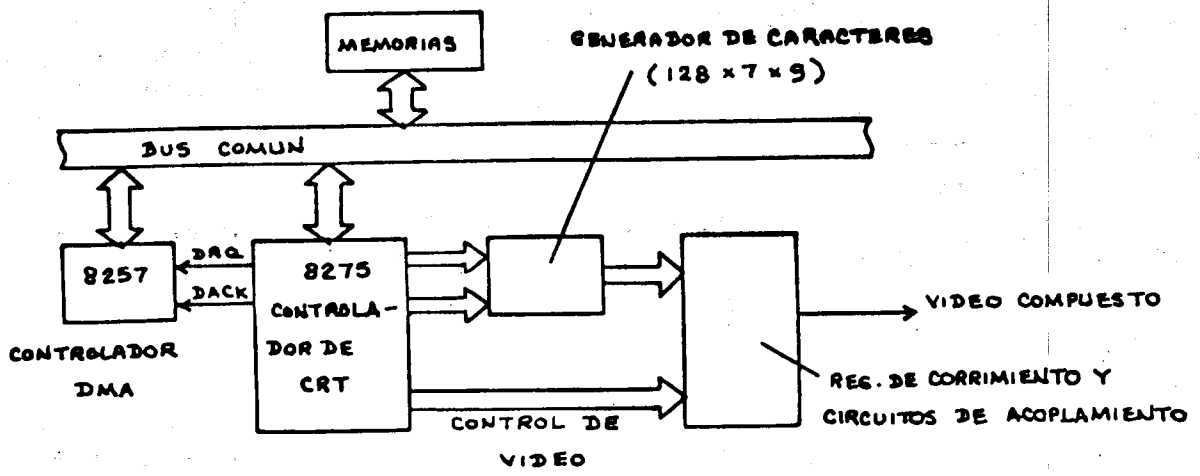


FIG 11 : INTEGRACION DE UN CONTROLADOR DE CRT EMPLEANDO EL CIRCUITO 8275 DE INTEL .

fecha de entrega de la obra  
 el presente proyecto de obra  
 para el año 1983.  
 el presente proyecto de obra  
 para el año 1983.  
 el presente proyecto de obra  
 para el año 1983.

**'LINEAS TELEFONICAS Y MODEMS'**

el presente proyecto de obra  
 para el año 1983.  
 el presente proyecto de obra  
 para el año 1983.  
 el presente proyecto de obra  
 para el año 1983.

Juan Martínez G.

Febrero 1983.

**FAC. DE INGENIERIA**  
**DOC. DE INVESTIGACION**

## 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:

FAC. DE INGENIERIA  
DOCUMENTACION

Pin.	Designación	Función	Abreviatura	Sentido E.D. ↔ módem
1	AA	Tierra de protección	GND	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.

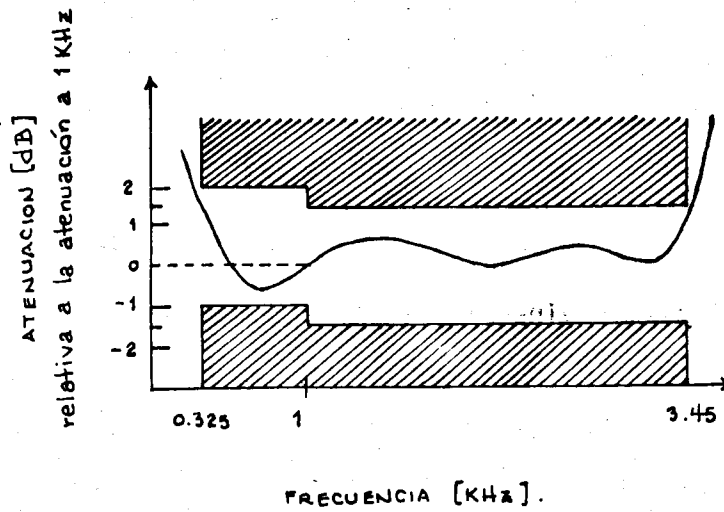


FIG. 1 : RESPUESTA EN FRECUENCIA TIPICA DE UNA LINEA TELEFONICA. LA ZONA ACHURADA SEÑALA LOS LIMITES OBSERVADOS ESTADISTICAMENTE.

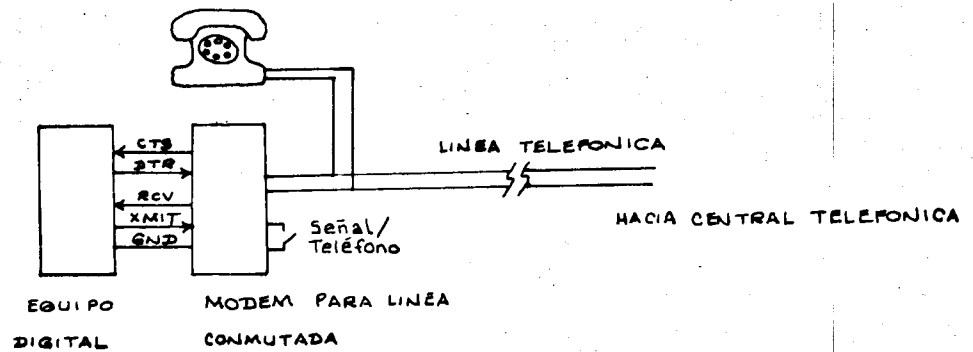


FIG 2 : ACOPLAMIENTO MODEM - EQUIPO DIGITAL EN TRASMISION POR LINEA CONMUTADA.

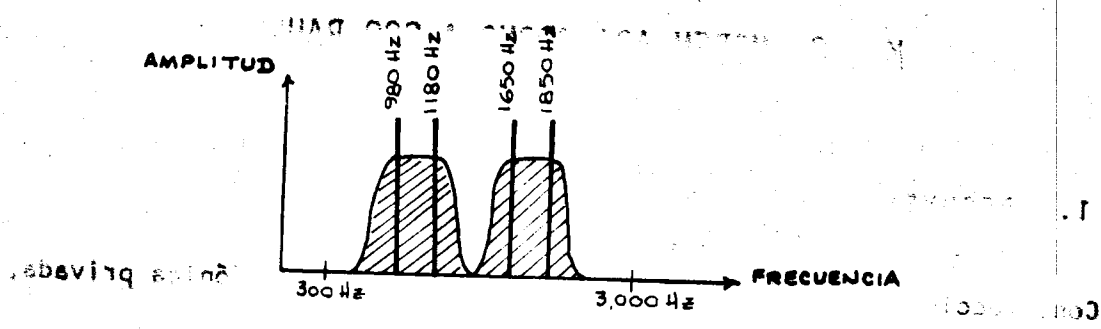


FIG 3 : ASIGNACION DE FRECUENCIAS EN UN MODEM DE 300 BAUDS , FULL DUPLEX , DE ACUERDO AL ESTANDAR CCITT V.21

VELOCIDAD [BITS POR SEGUNDO]	VEL. DE MODULACION [BAUDS]	MODULACION	BITS POR SIMBOLO	ANCHO DE BANDA [HERTZ]
2400	1200	DIFERENCIAL, 4 FASES	2	1,200
2400	800	DIFERENCIAL, 8 FASES	3	800
4800	1600	DIFERENCIAL, 4 FASES	3	1,600
9600	4800	DIFERENCIAL, 2 FASES 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  $f_c = 5\text{KHz}$ . 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  $600 \Omega$  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 trasmite 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

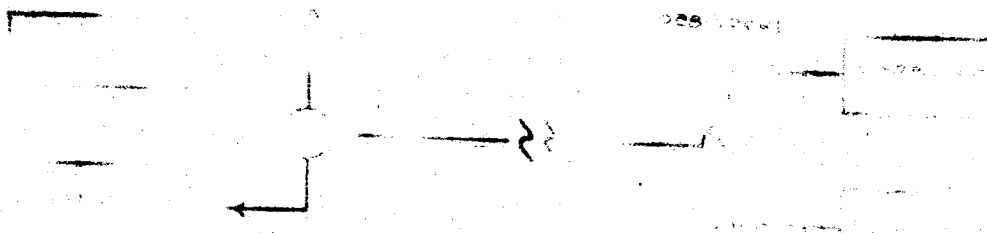
Acoplamiento a la línea telefónica: sólo para líneas privadas, M200 tiene impedancia de salida de  $600 \Omega$ .

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

854004

ASDAIT

025 10001



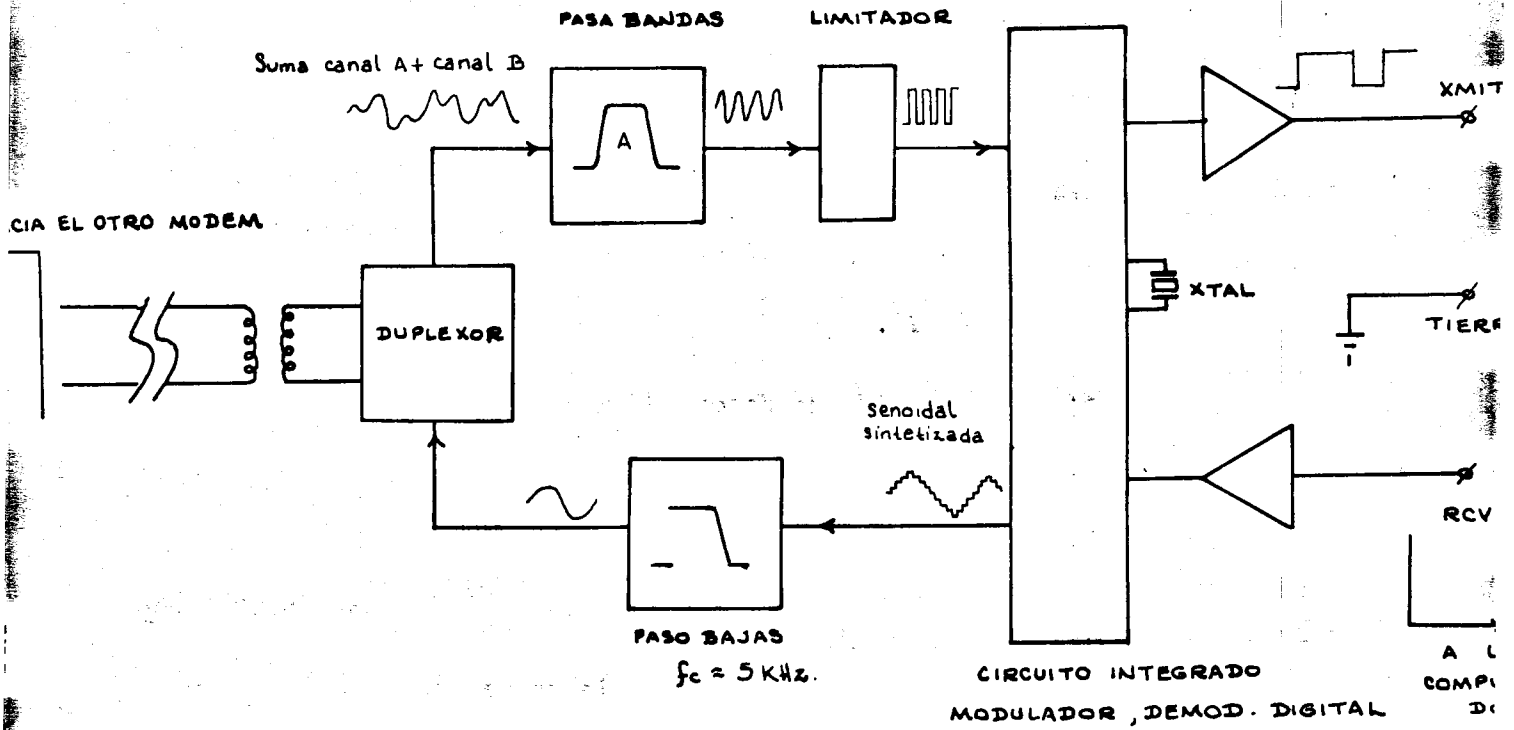


FIG 1: DIAGRAMA DE BLOQUES

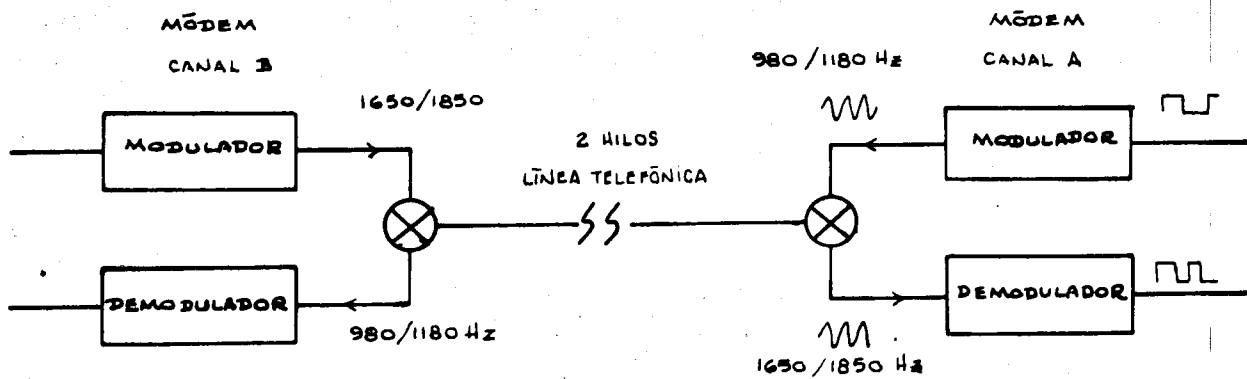


FIG 3: ASIGNACION DE FRECUENCIAS

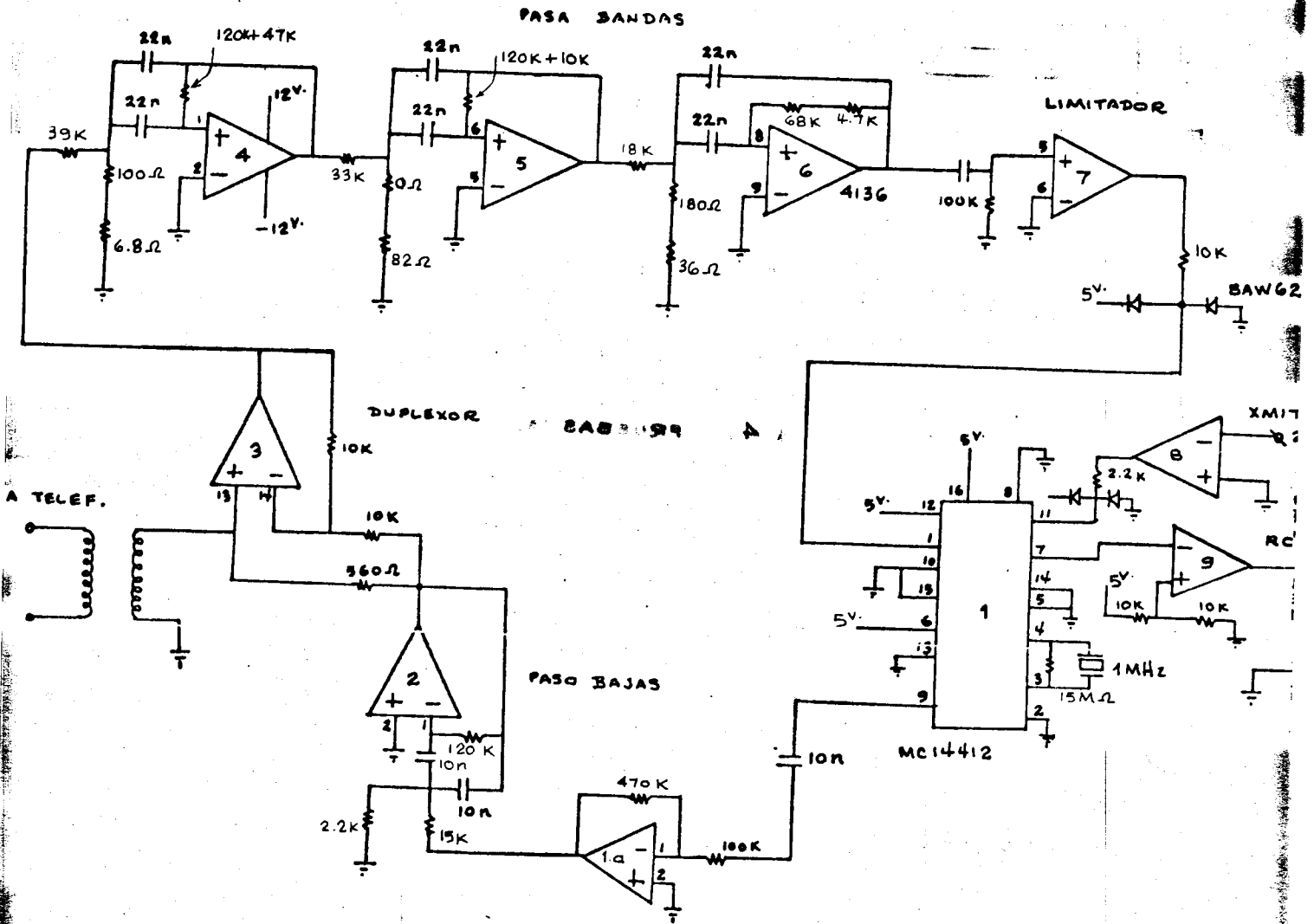


FIG 2: DIAGRAMA ELECTRÓNICO M-200  
CANAL B

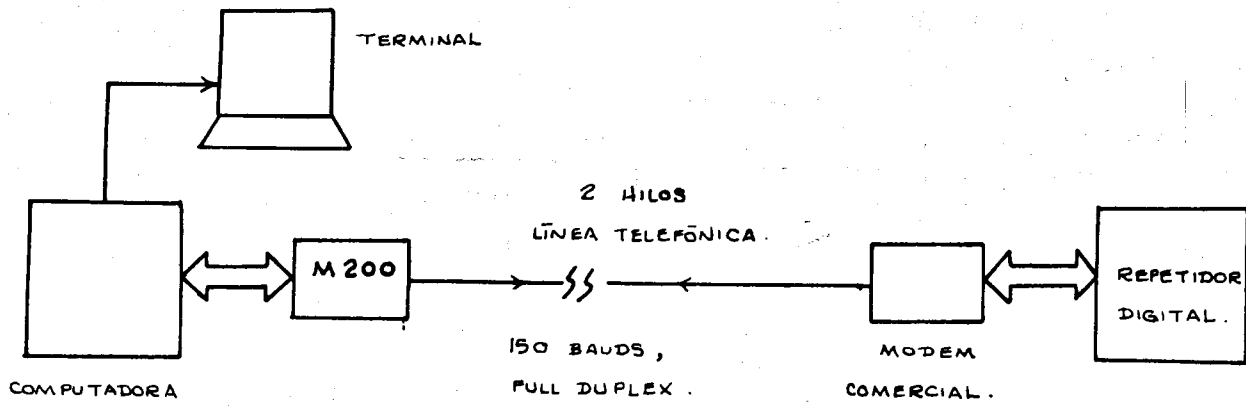


FIG 4. PRUEBAS AL M-200

"CARACTERISTICAS ESPECIFICAS DE MICROPROCESADORES  
DE 16 BITS"

FAC. DE INGENIERIA  
DOCUMENTACION

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 asíncronamente:
- 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 relocalizables.
- e)  $10^6$  bytes de memoria externa.

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

Zilog proporciona 2 versiones del microprocesador 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 relocalizables 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.

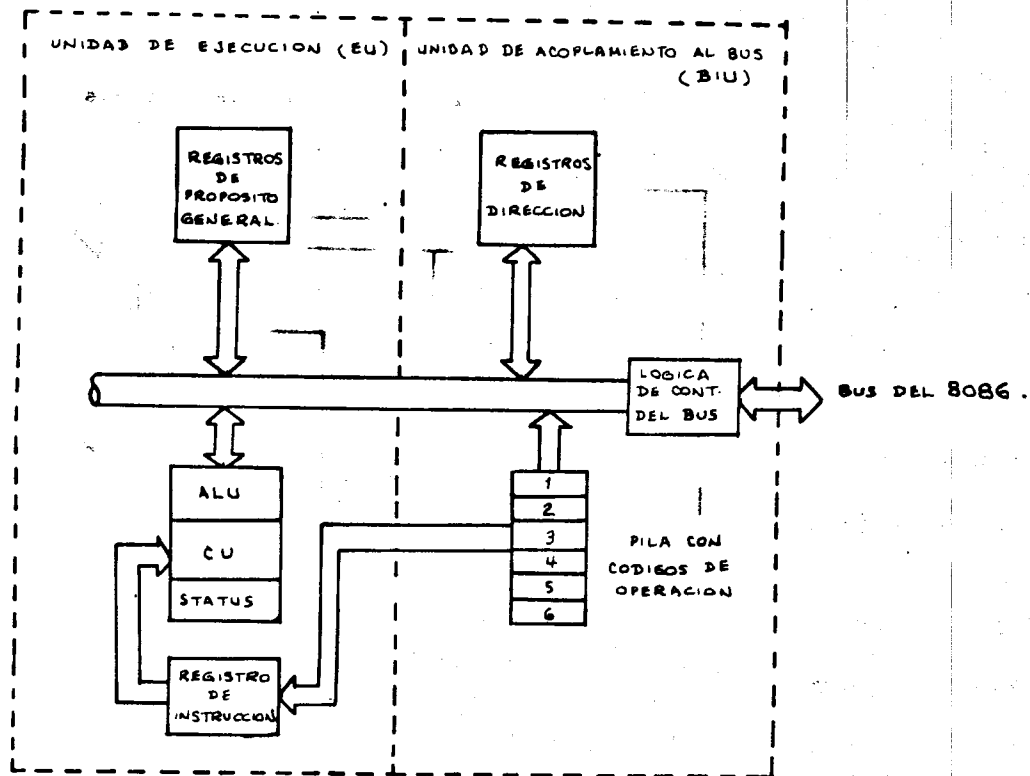


FIG 1. SISTEMA DE 'PIPELINE' EN EL 8086

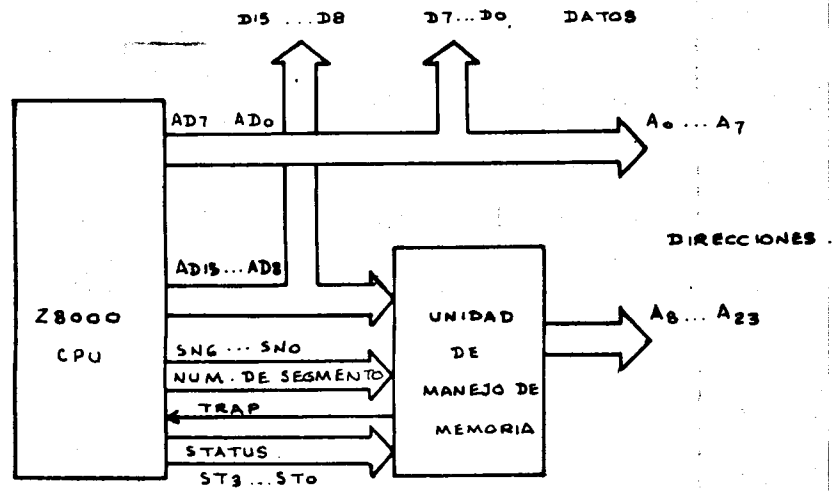


FIG 2 : ACOPLAMIENTO DEL Z8000 Y LA UNIDAD DE MANEJO DE MEMORIA.

ARQUITECTURA DE UN MICROPROCESADOR

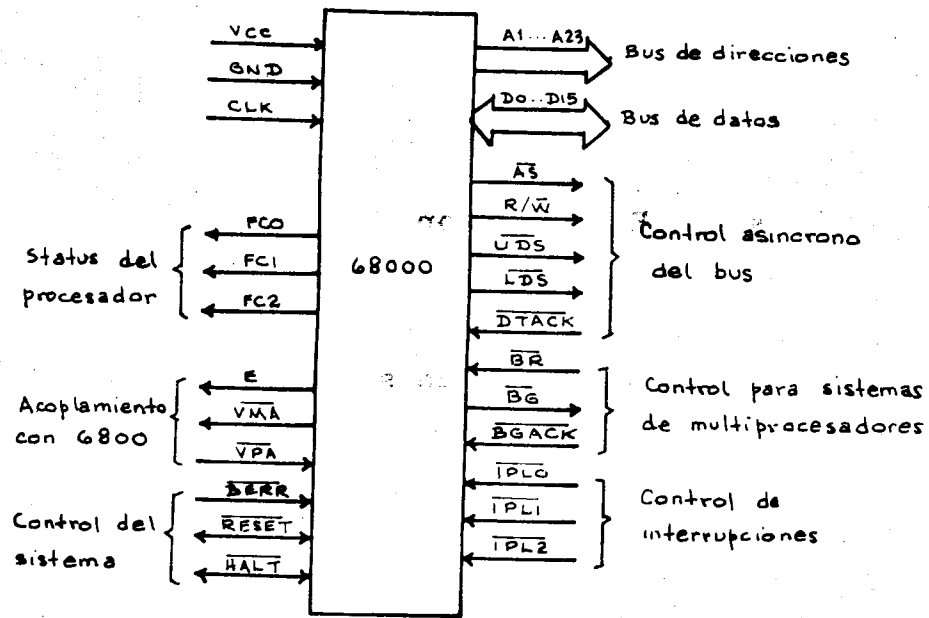
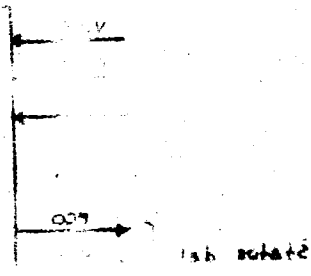


FIG 3. SEÑALES PRESENTES EN EL MICROPROCESADOR 68000



## 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



2 99 2

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, tratamientostérmicos, 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.
Linearidad	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 6 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:

$$V_o = \alpha T$$

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 Níquel-aluminio	-200° a 1250°	-5.51 a 51.05
E	Níquel-cromo cuproníquel	0° a 1000°	0 a 75.12
T	Cobre/cuproníquel	-200° a 400°	-5.28 a 20.80

El voltaje desarrollado en la junta de referencia (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 al 1%°C. En instrumental de precisión, sin embargo, se aproximan usualmente las curvas mostradas con varios segmentos lineales.

El mismo efecto termoeléctrico que <sup>se</sup> permite a los termopares generar 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 relativamen

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 isotermal. 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 vóltmetro 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 juntura 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 alambrada 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 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  $-100\text{K}\Omega/\text{°C}$ ) en la región de bajas temperaturas ( $-80\text{°C}$ ). En la fig 8, se muestra la respuesta de un termistor típico en la zona de  $0\text{°}$  a  $50\text{°C}$ .

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  $\pm 0.02\text{°C}$ , 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 junta fría en los termopares.

## 6. DISEÑO DE UN SISTEMA DE ADQUISICION 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 ( $V_o < 2 \text{ mV}$ ) 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  
 Zin: 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 conversor 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.

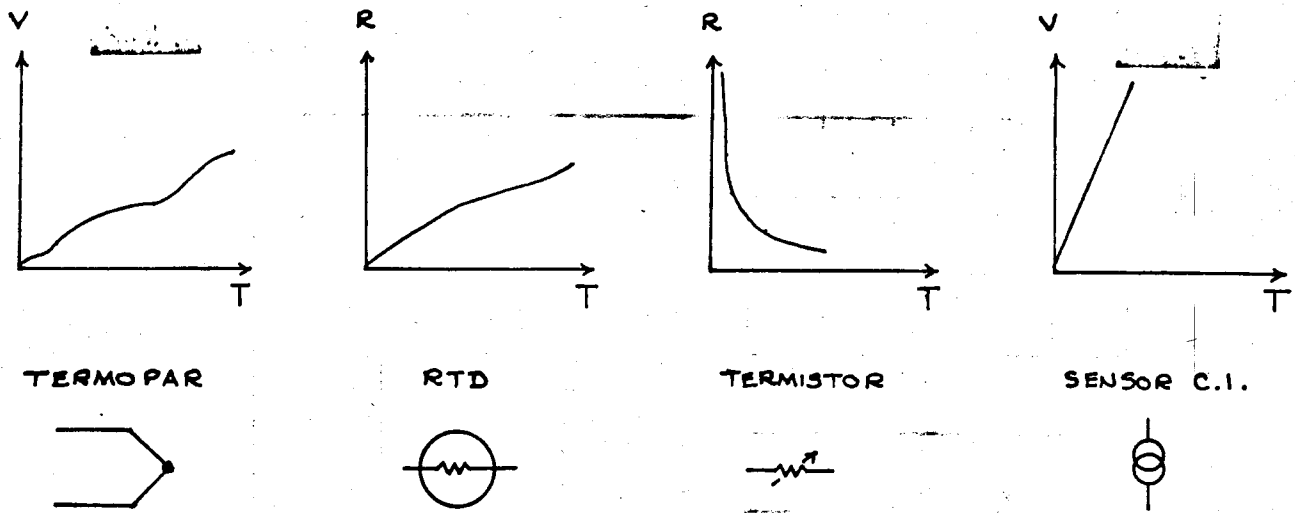


FIG. 1 : CURVAS DE TRANSFERENCIA PARA  
SENSORES DE TEMPERATURA.



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



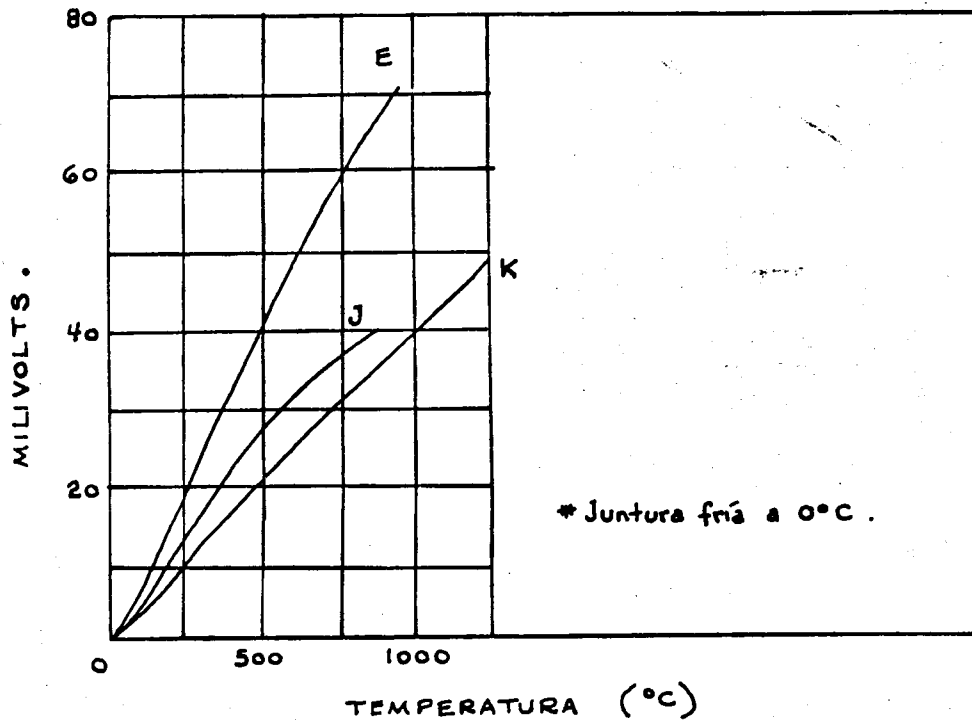


FIG 3. CURVAS DE TRANSFERENCIA DE TERMOPARES TIPOS E, J, K.

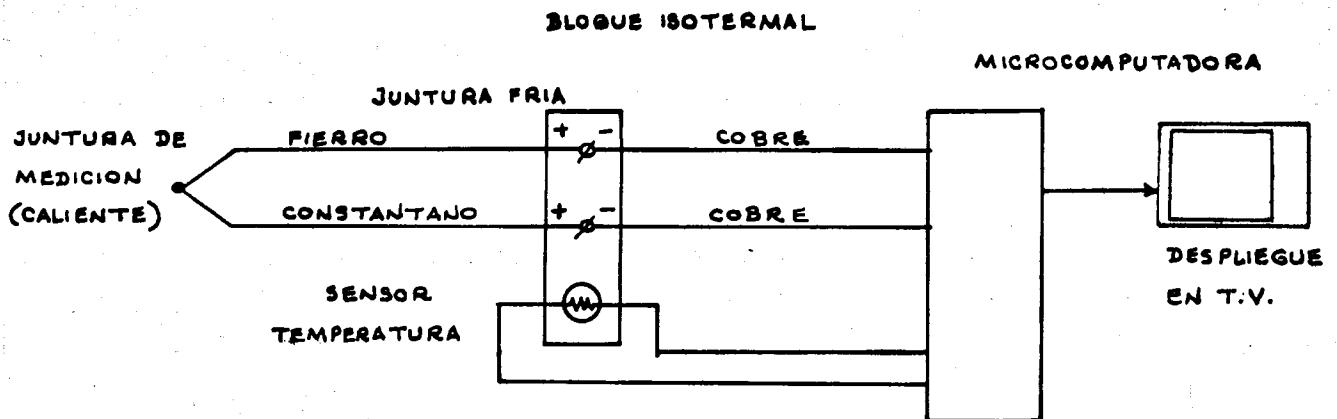


FIG 4. MEDICION DE TEMPERATURA CON COMPENSACION DEL VOLTAGE DE ERROR EN LA JUNTA FRIA.

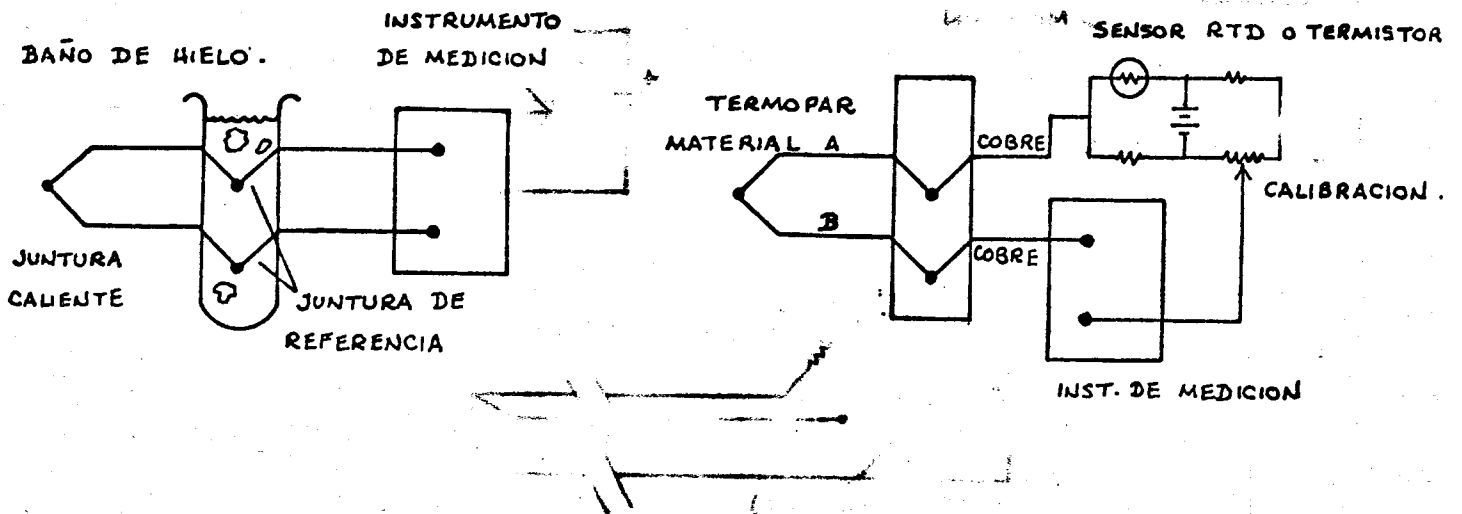


FIG 5. METODOS DE COMPENSACION DEL ERROR GENERADO EN LA JUNTURA FRIA.

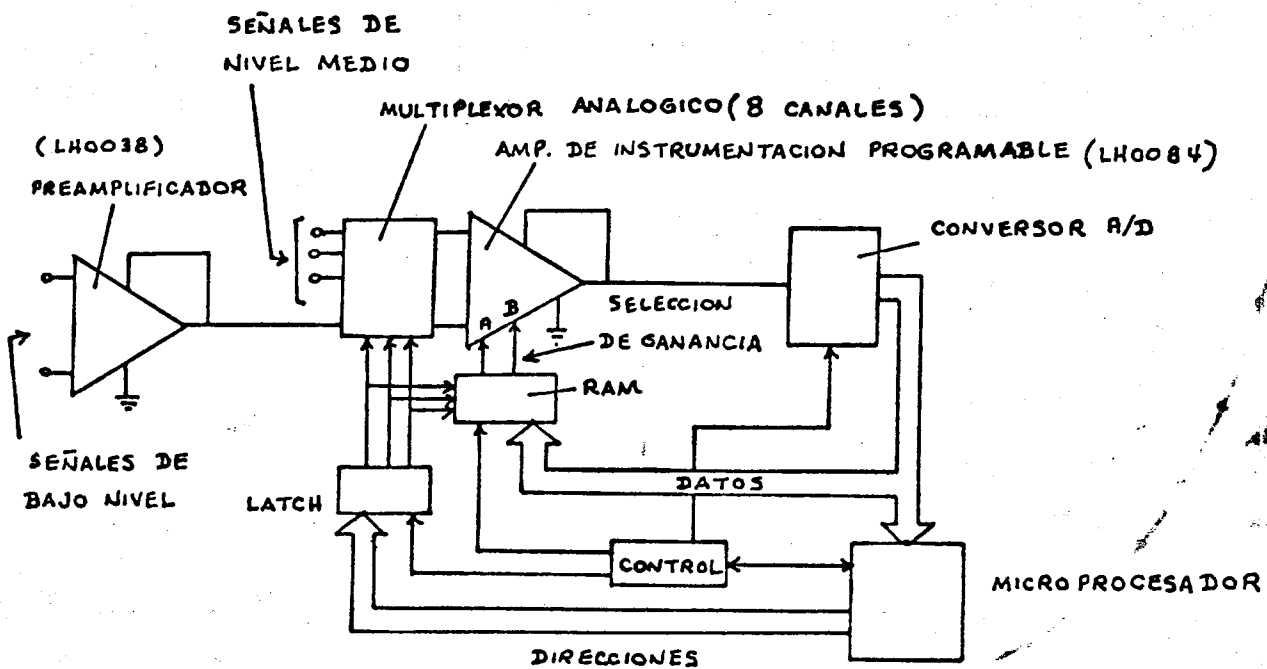


FIG 6. SISTEMA DE ADQUISICION DE DATOS

30 9M

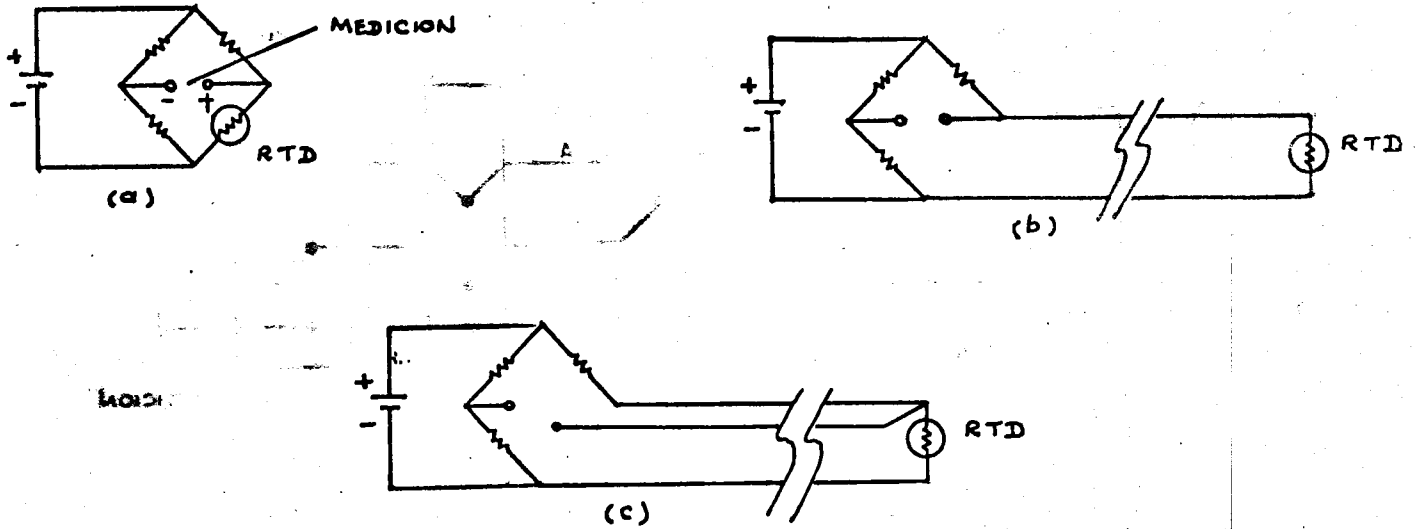


FIG 7. CONEXIONES DE SENSORES RTD

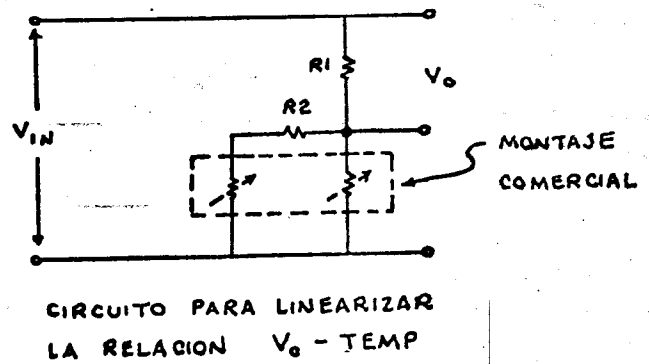
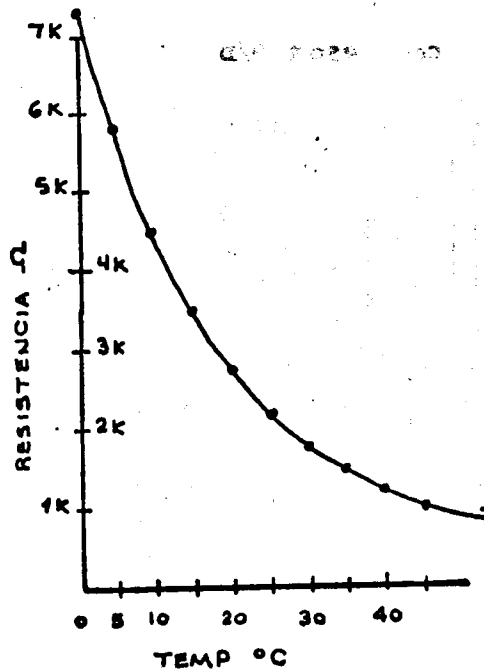


FIG 8. RESPUESTA TIPICA DEL TERMISTOR Y CIRCUITO PARA SALIDA LINEAL.

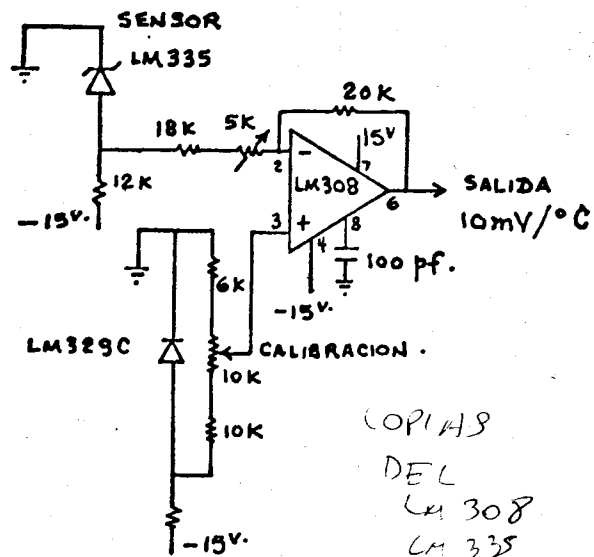
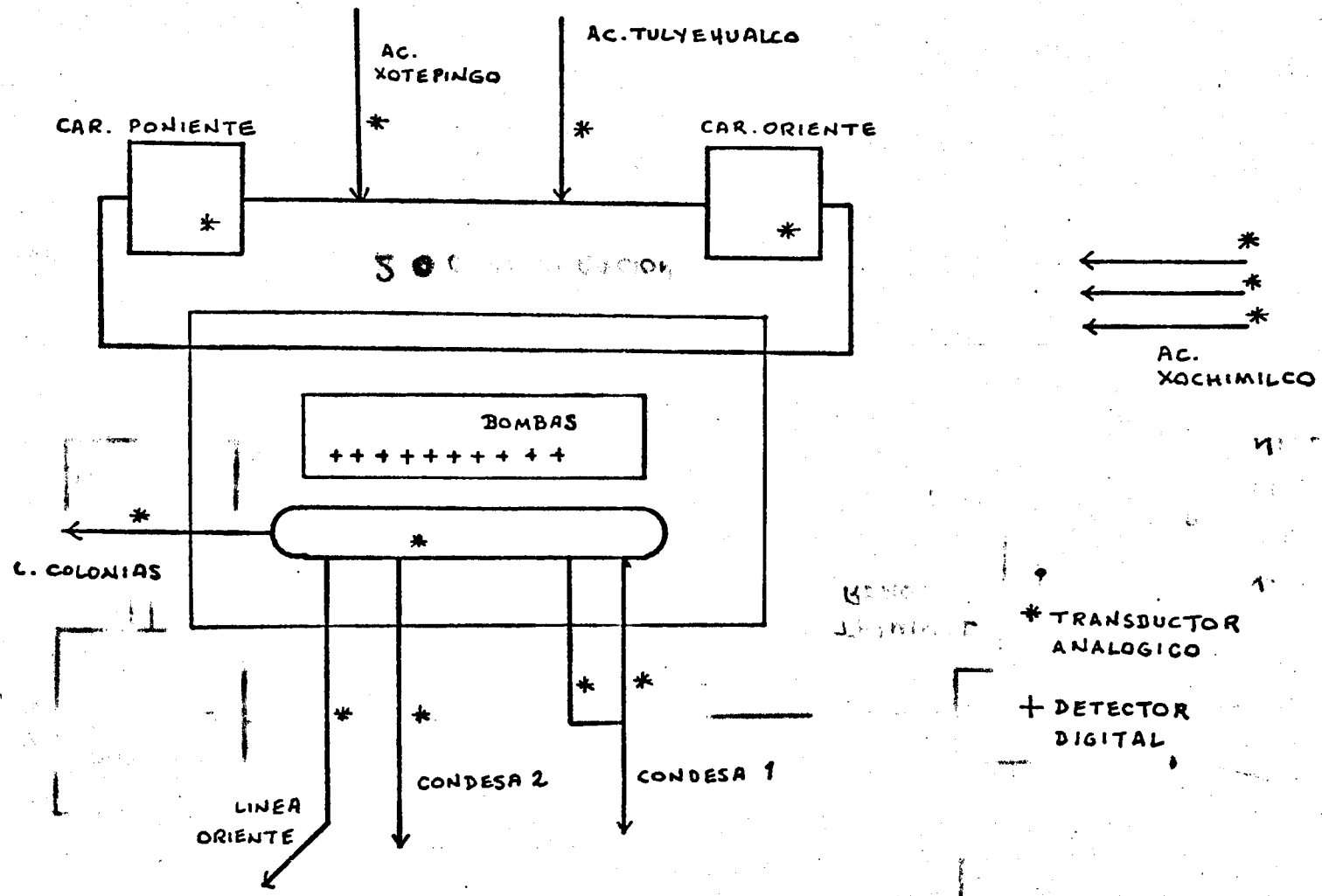


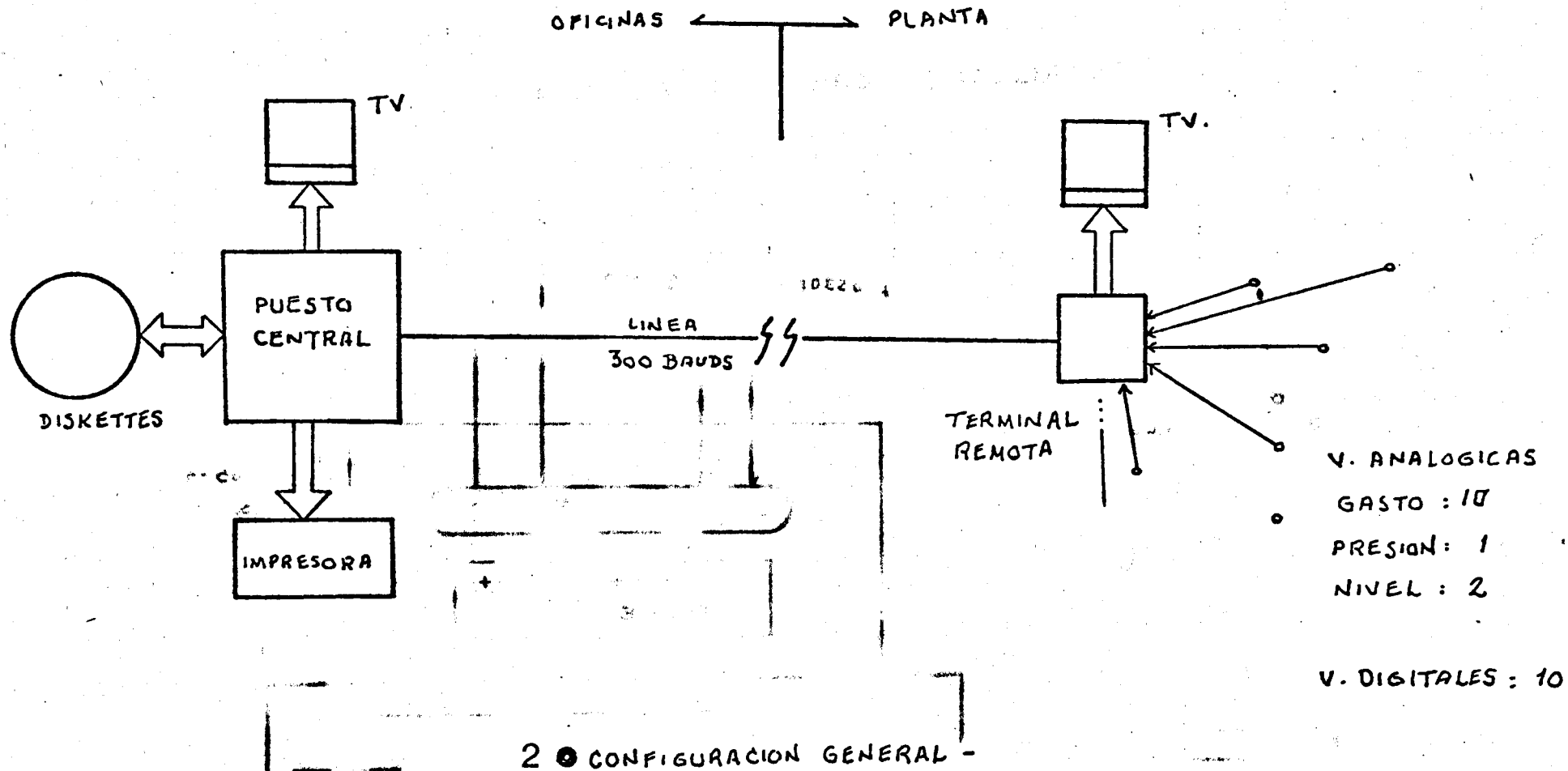
FIG 9 . CIRCUITO PARA SENSOR DE TEMPERATURA, CON SALIDA REFERIDA A TIERRA

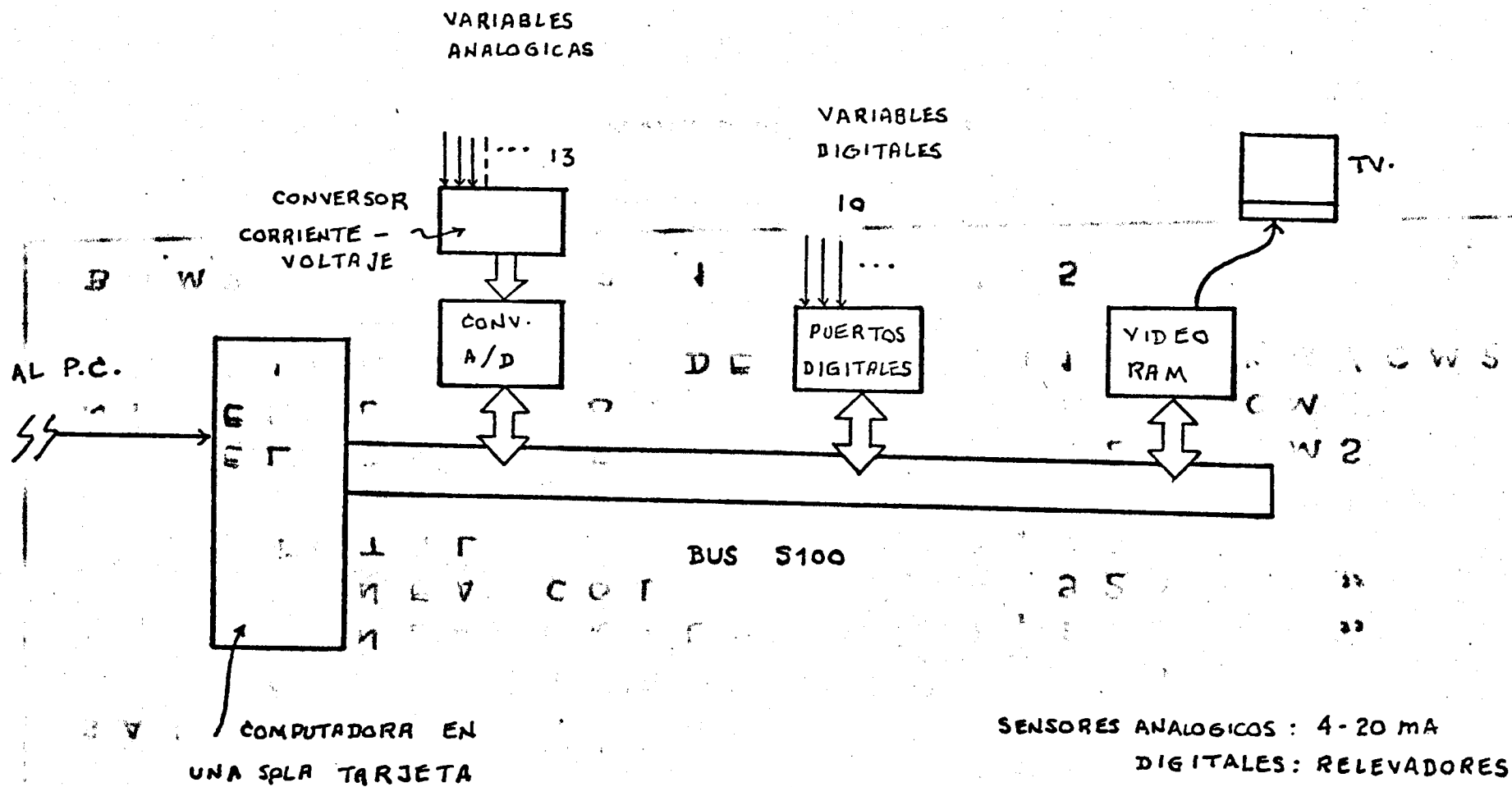
COPIAS  
DEL  
LM 308  
LM 335  
LM 329C

SISTEMA DIGITAL DE INFORMACION PARA LA PLANTA  
DE BOMBEO DE XOTEPINGO



1 ● PLANTA DE BOMBEO DE XOTEPINGO

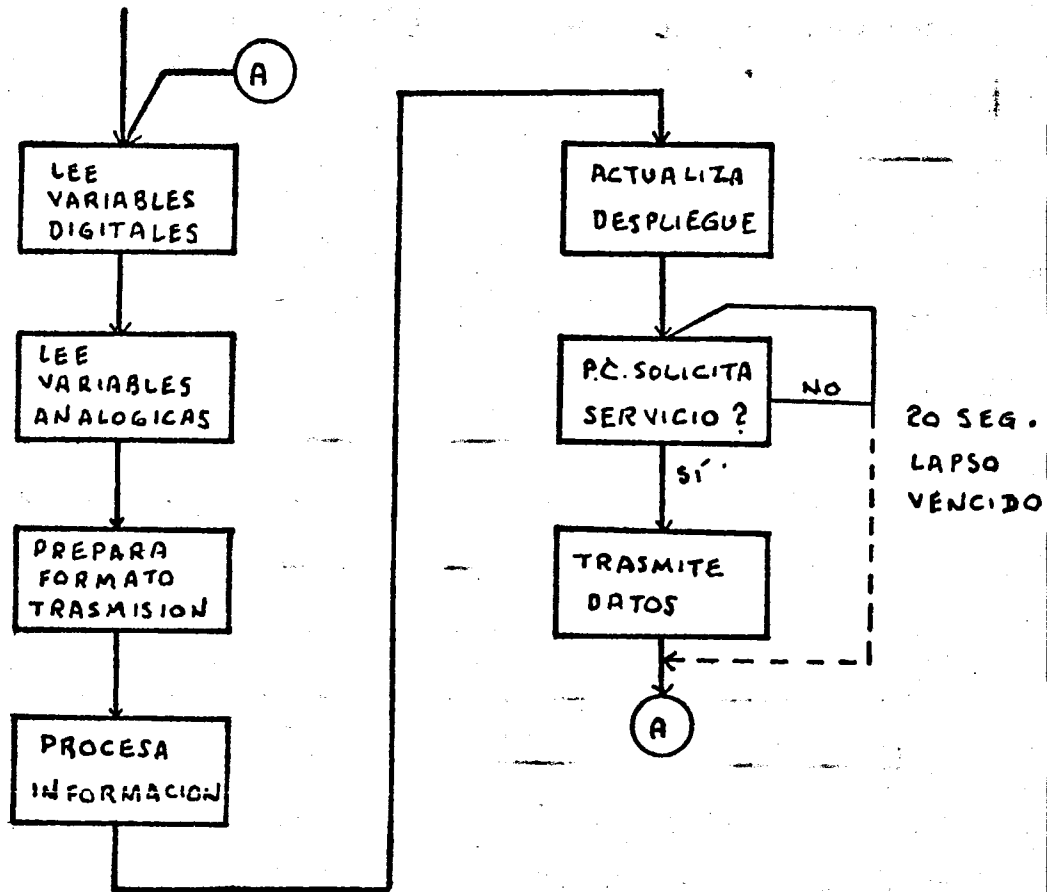




3 • DIAGRAMA TERMINAL REMOTA



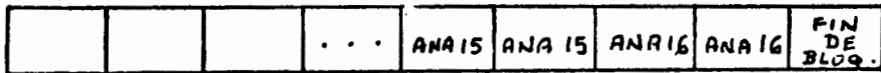
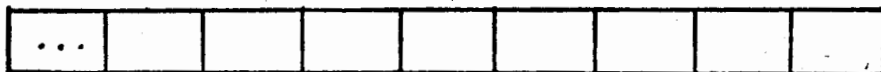
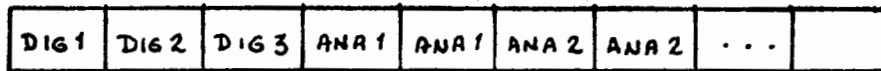
ENT :	AC . X O C H I M I L C O	7 6 0	L T S / S
	AC . T U L Y E H U A L C O	1 , 8 5 0	"
	A U X . X O T E P I N G O	1 , 0 0 0	"
	T O T A L	3 , 6 1 0	"
S A L :	C O N D E S A U N O	5 4 0	"
	C O N D E S A D O S	7 1 0	"
	L I N E A O R I E N T E	1 , 1 0 0	"
	L I N E A C O L O N I A	9 2 0	"
	T O T A L	3 , 2 9 0	
N I V E L	C A R . O R I E N T E	3 2 0	C M S
N I V E L	C A R . P O N I E N T E	6 6 0	C M S
P R E S I O N	A L A D E S C .	1 1 . 6	K G / C M 2
B O M B A S	E N G P . 1 . 2 . . . . . 4 . 5 . 6 . . . . . 9 .		



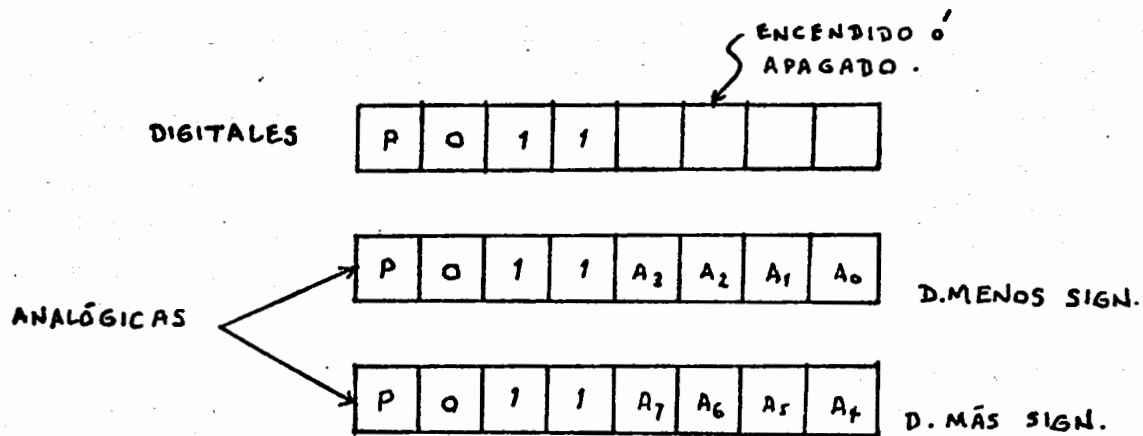
5 ● DIAGRAMA DE FLUJO  
PROGRAMA EN TERMINAL  
REMOTA.

TRASMISION : • FULL DUPLEX  
 • 300 BAUDS  
 • ASINCRONA

BLOQUE DE DATOS :



TOTAL 36 BYTES



6 • FORMATO DE TRASMISIÓN

RED TELEMETRICA DIGITAL

FAC. DE INGENIERIA  
DOCUMENTACION

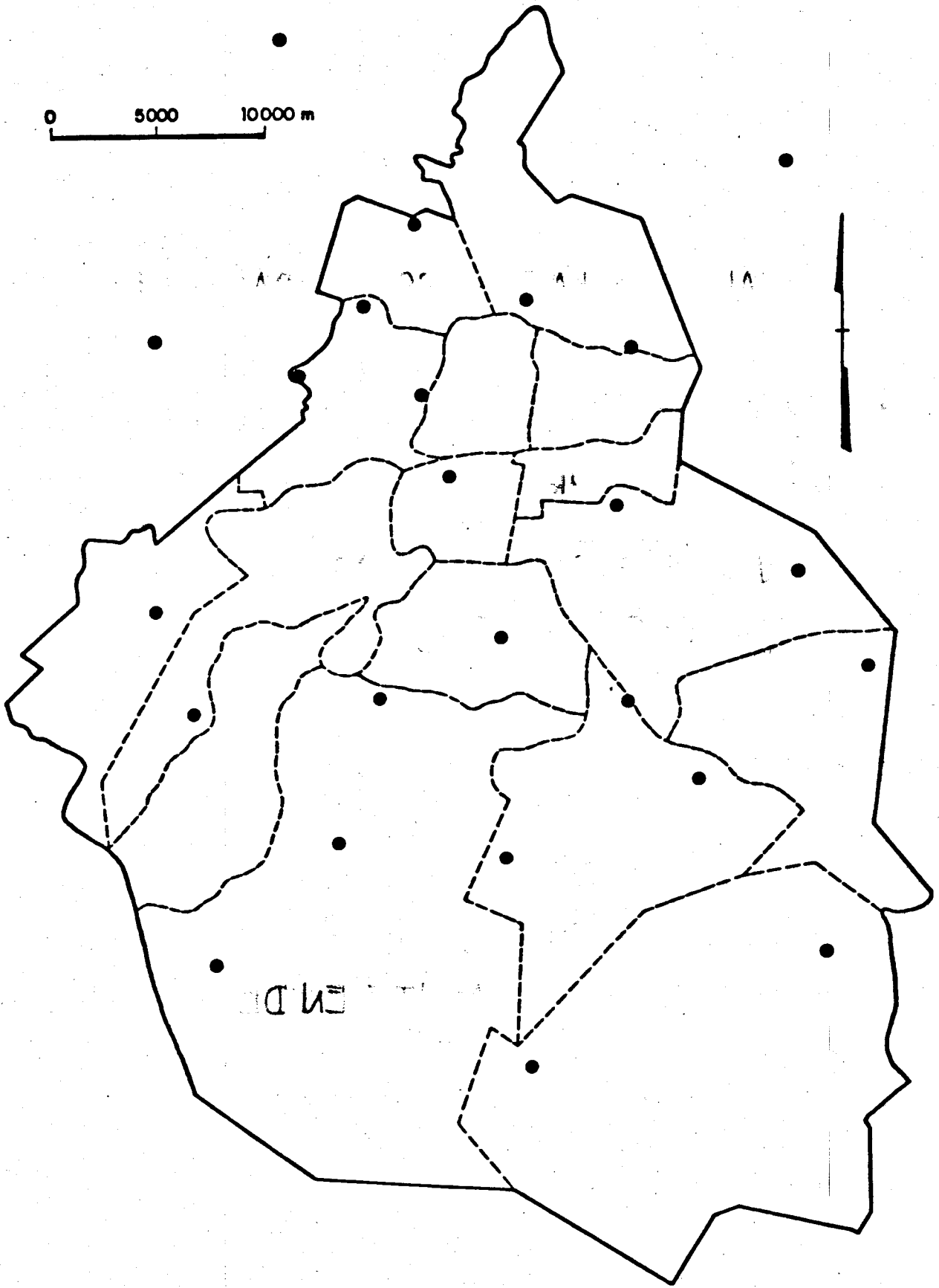
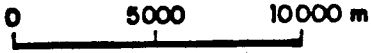
INSTITUTO DE INGENIERIA  
PATROCINIO : D.G.C.O.H. D.D.F.

1983

## 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

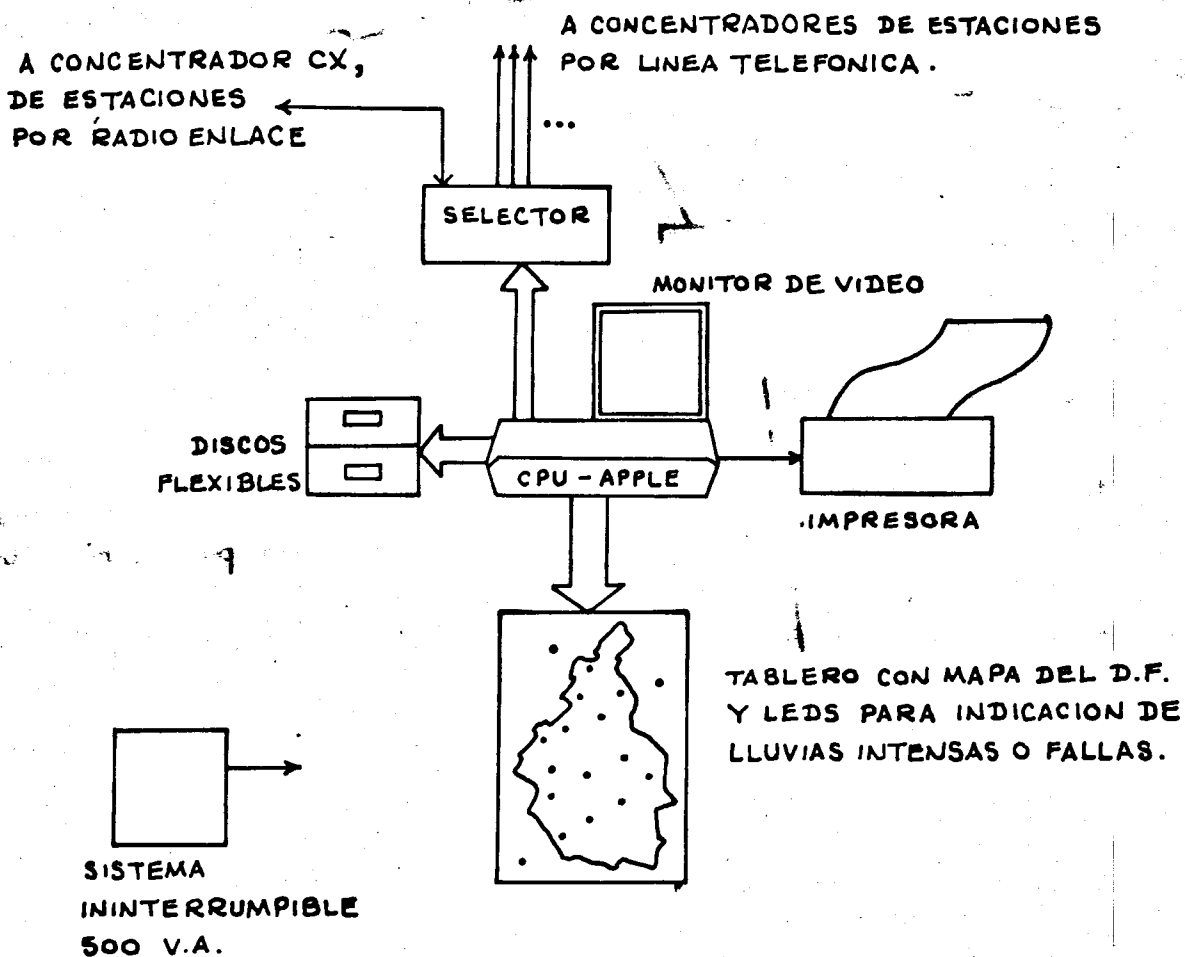
CANAL PARA LA TELECOMUNICACION :

- 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





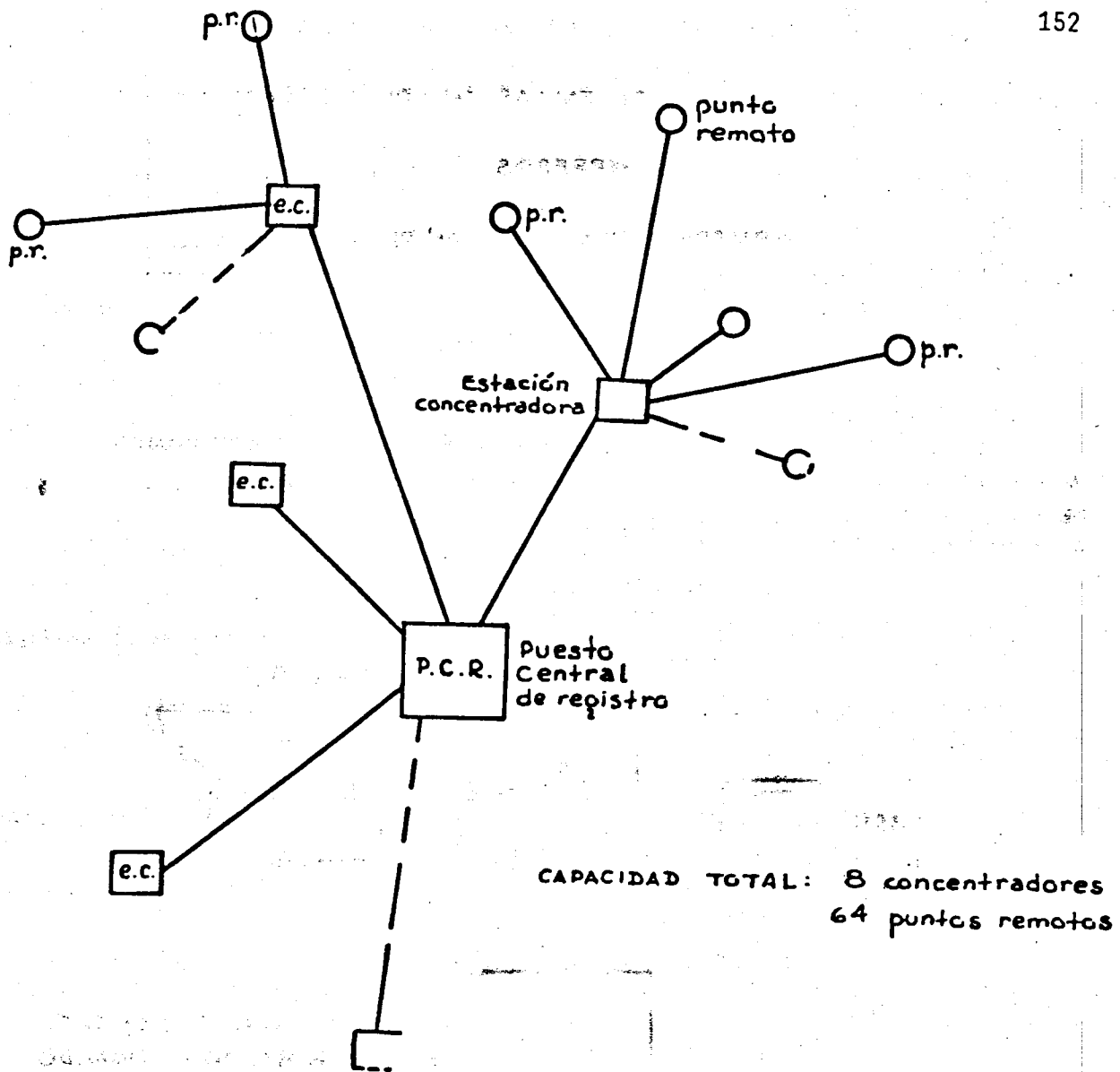
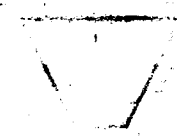


FIG 1 CONFIGURACION GENERAL DE LA RED. (VIA TELEFONICA)

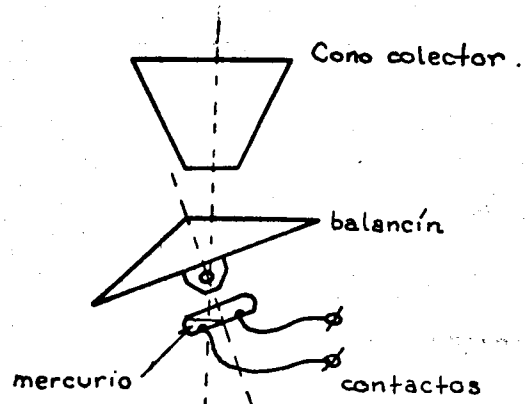
Figura 1. mm D.F.C. a = 2000



### 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)

en el suministro de energía

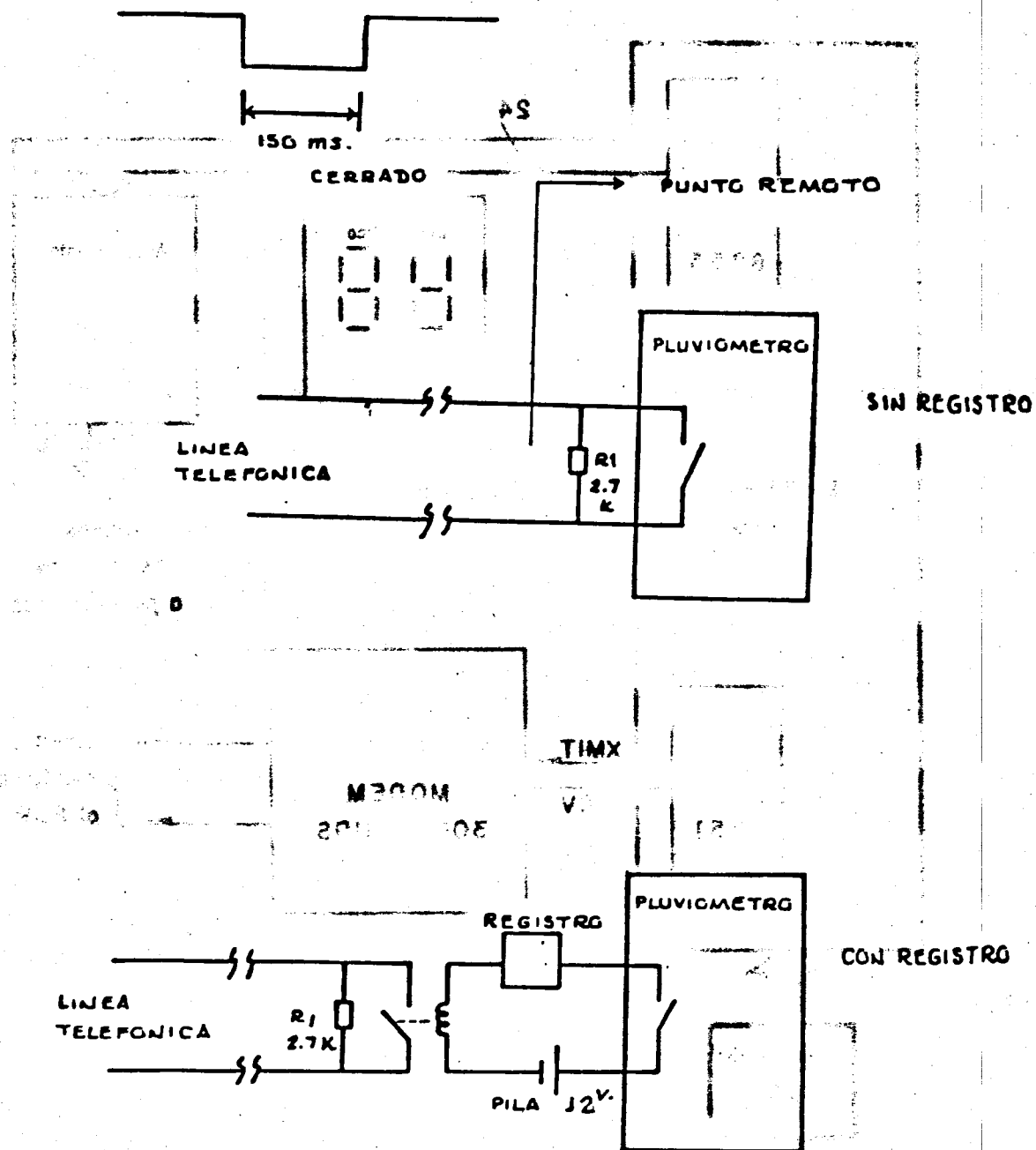


1 pulso = 0.254 mm. de lluvia .

PLUVIOMETRO

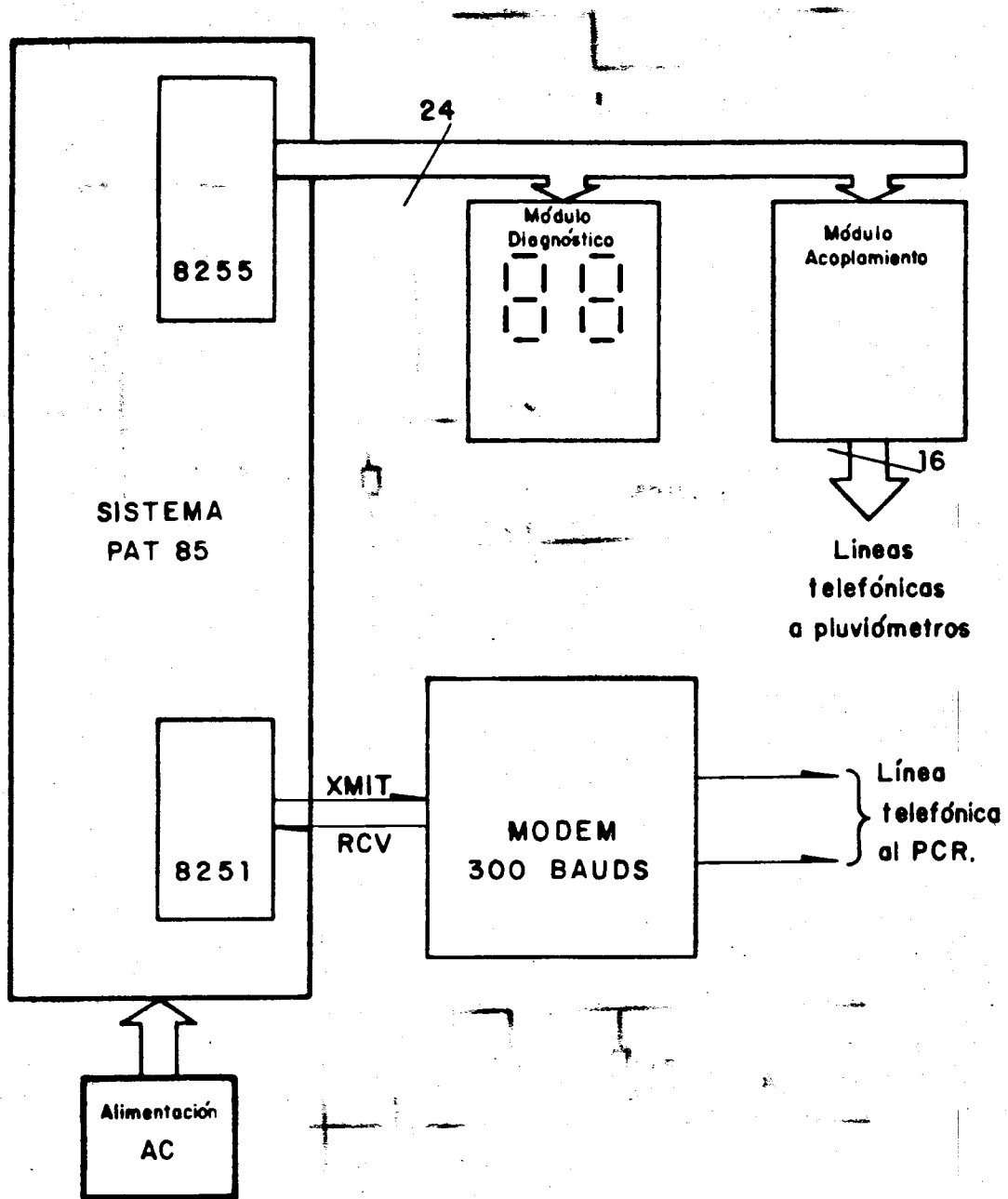
los puntos remotos son pasivos

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

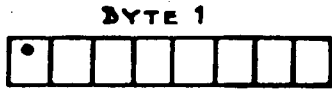


5

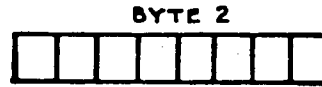
FIG 2: CONEXION DE LGS PLUVIGMETRGS



ESTADO DE  
UN CANAL :

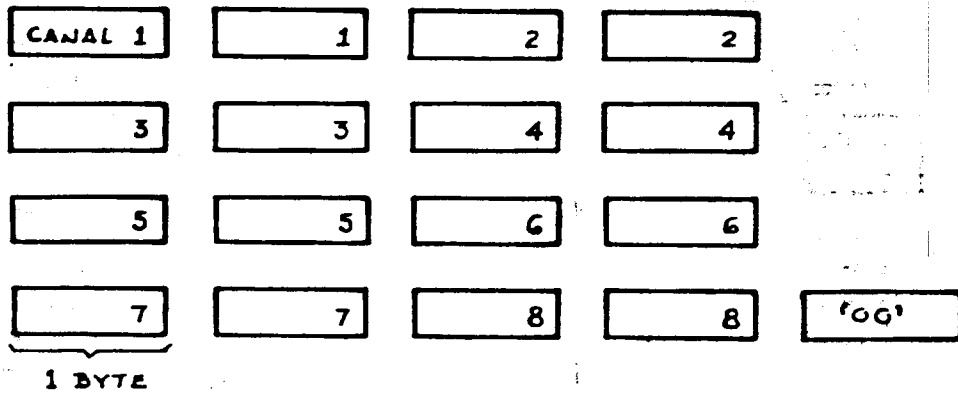


NO FALLA = 30 H  
FALLA = 31 H



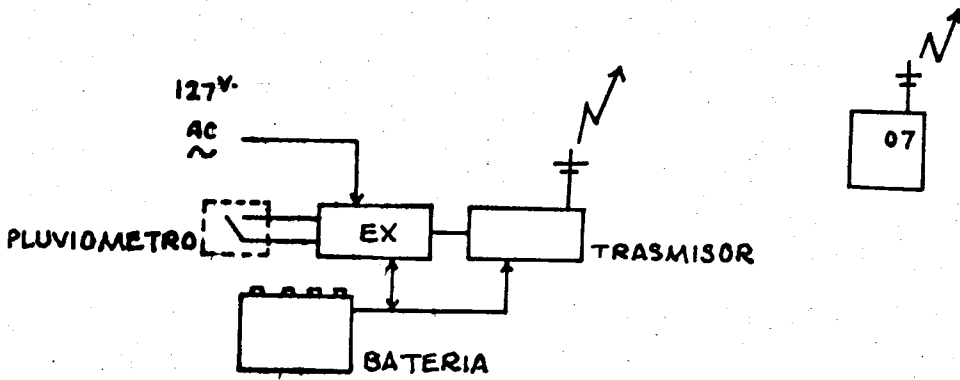
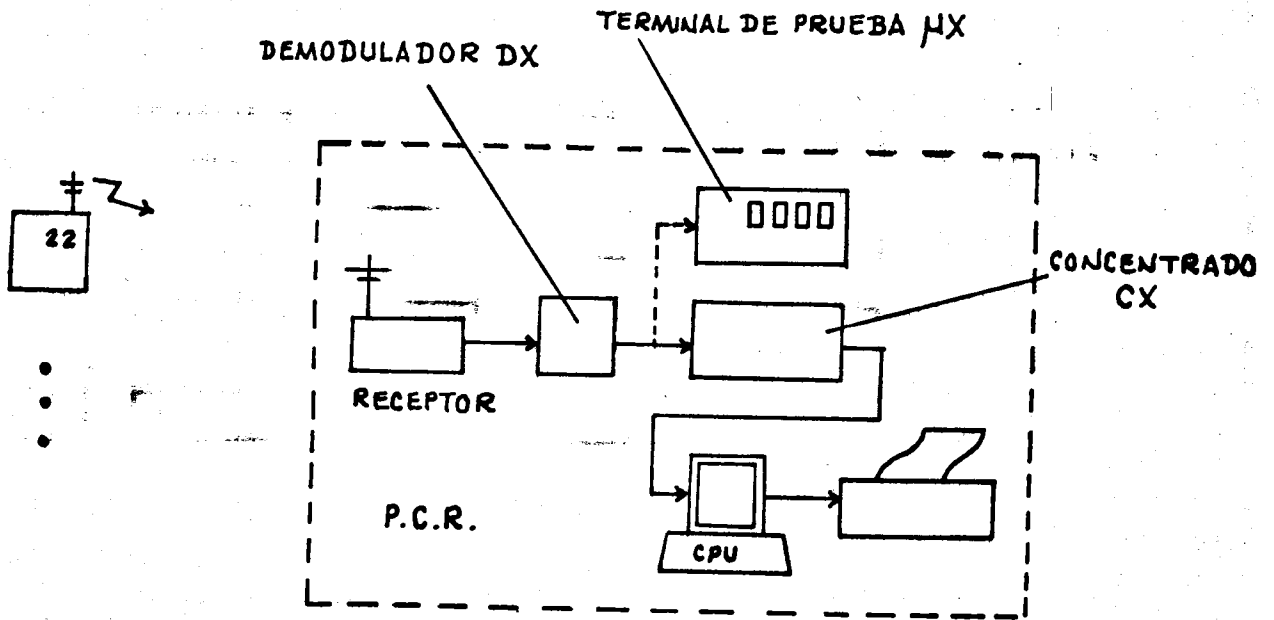
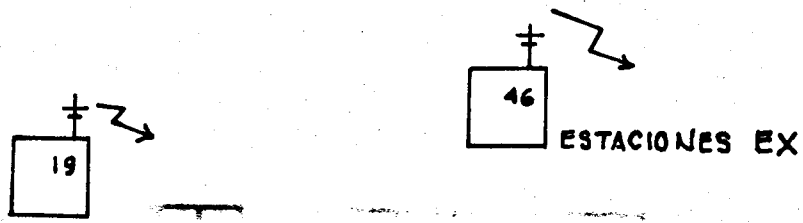
CUENTA ACUMULADA  
+ 30 H

ESTADO  
TOTAL :

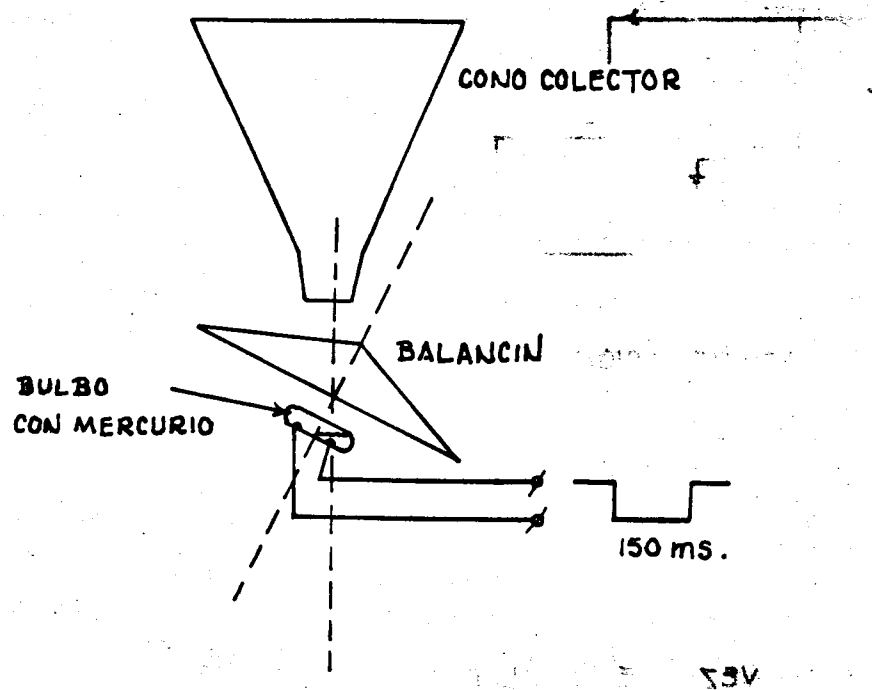


8.2

FIG 11 : ESTADO DE LA ESTACION  
CONCENTRADORA .



CONFIGURACION GENERAL  
(RED DE RADIOENLACE)

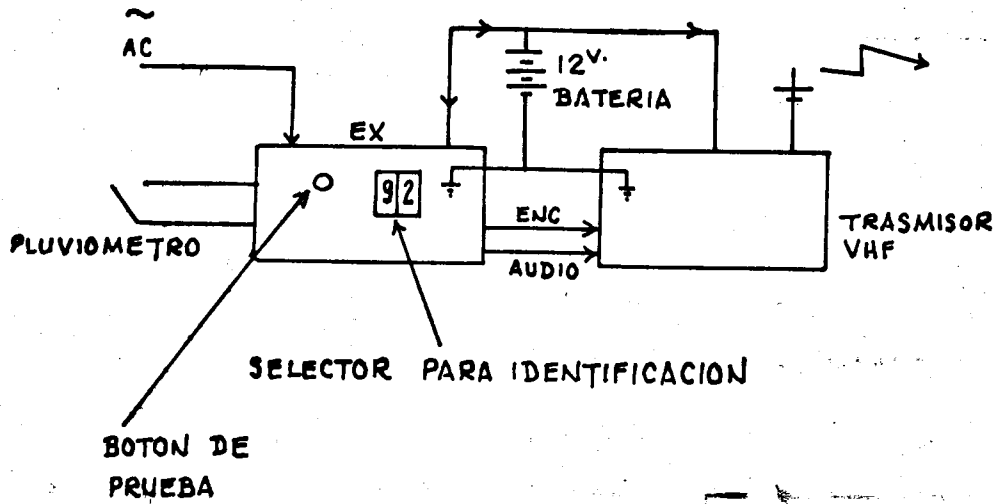


- CADA PULSO, EQUIVALENTE A 0.254 mm. DE LLUVIA.

- LA INFORMACION ES, DESDE SU ORIGEN, DIGITAL.

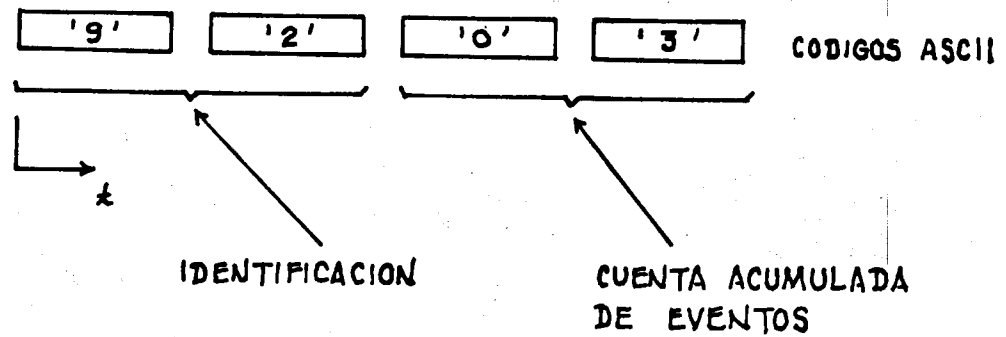
PLUVIOMETRO





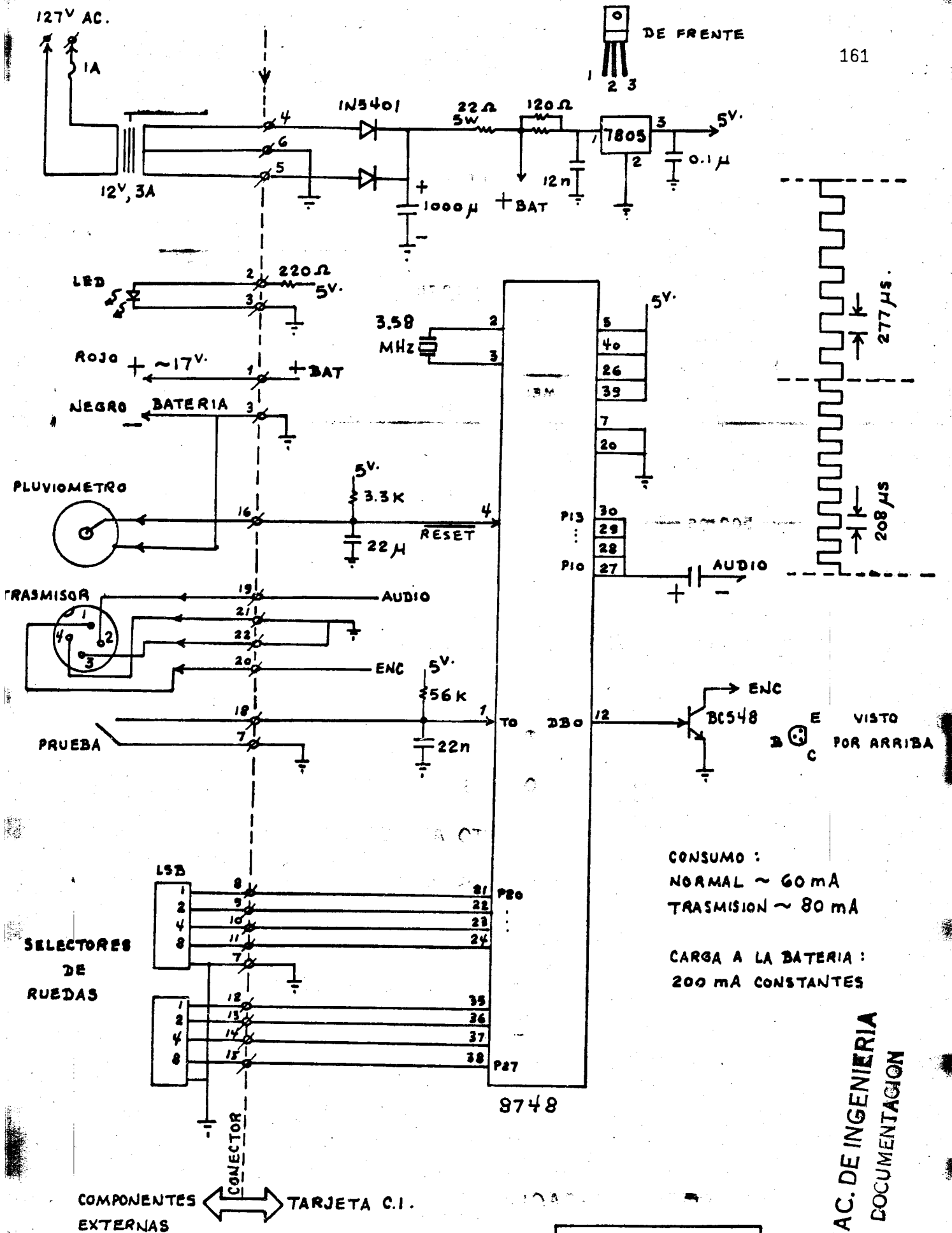
CADA VEZ QUE EN EL PLUVIOMETRO SE CIERRAN LOS CONTACTOS :

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



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

ESTACION EX

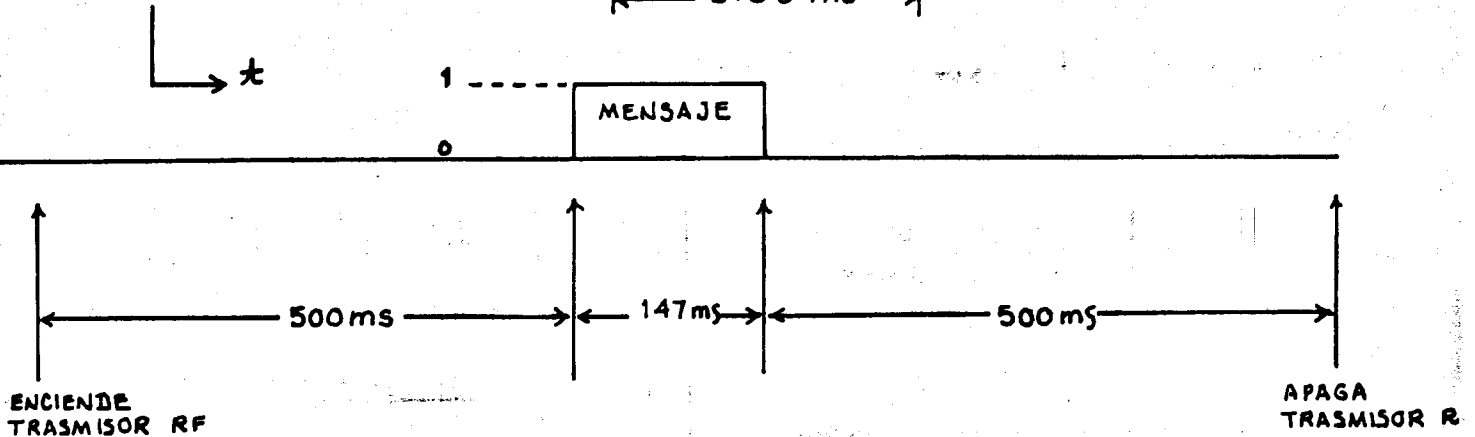
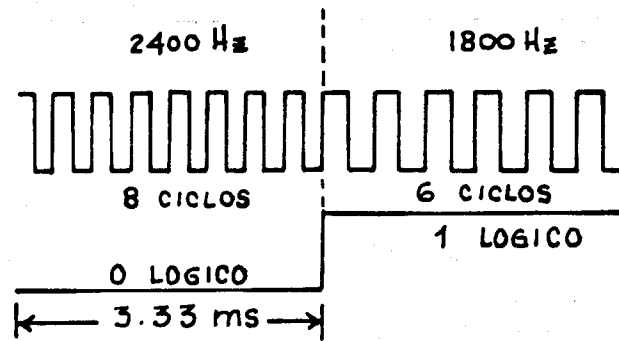


CONSUMO :  
 NORMAL ~ 60 mA  
 TRASMISION ~ 80 mA

CARGA A LA BATERIA :  
 200 mA CONSTANTES

ESTACION EX

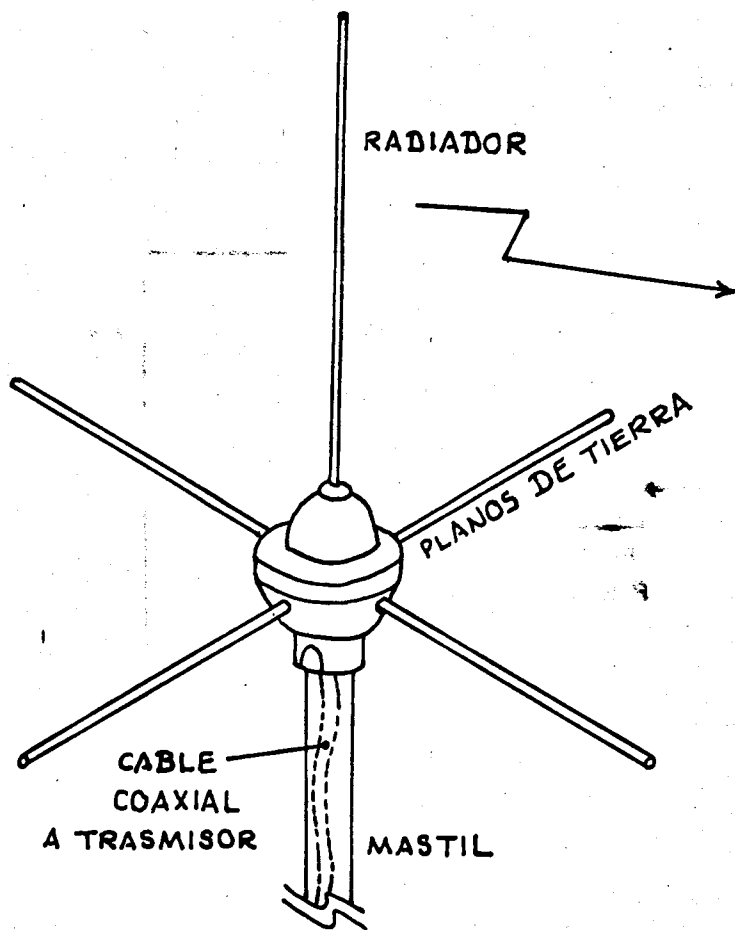
FAC. DE INGENIERIA  
 DOCUMENTACION



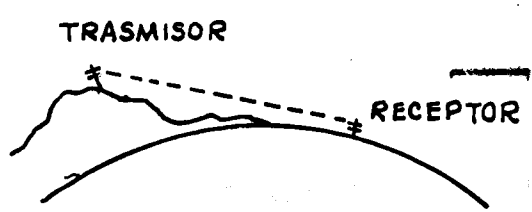
- MENSAJE :
- MODULACION FSK [1800 - 2400 Hz]
  - 300 BITS POR SEGUNDO
  - FORMATO ASINCRONO
  - CODIGOS ASCII.

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

ESPECIFICACION DEL MENSAJE



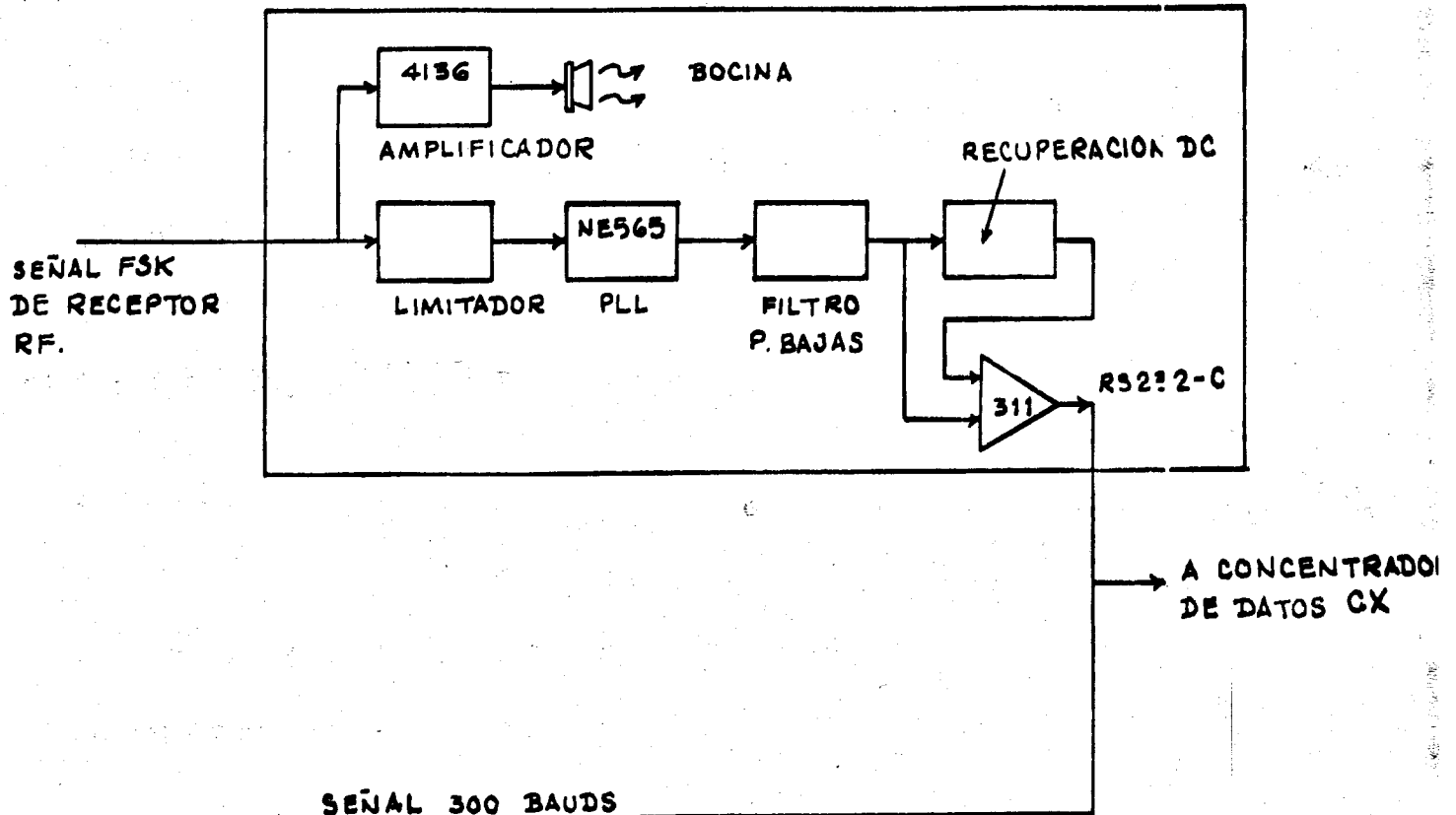
- OMNIDIRECCIONAL
- IMPEDANCIA 50Ω
- GANANCIA UNITARIA
- PATRON DE RAD. : 78°



- LINEA DE VISTA  
INDISPENSABLE

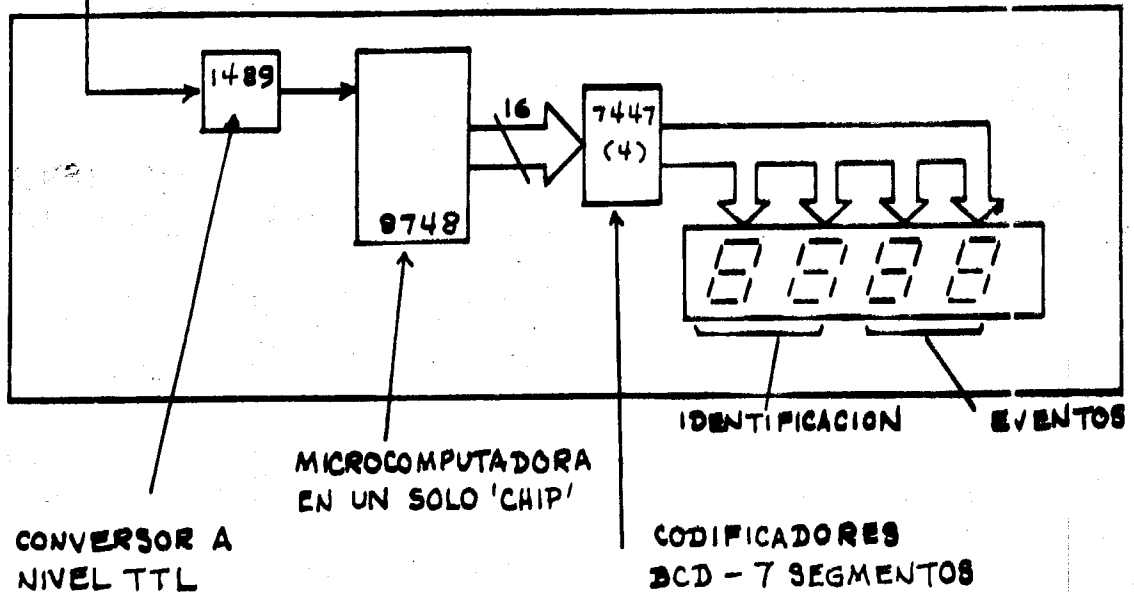
ENLACE RF

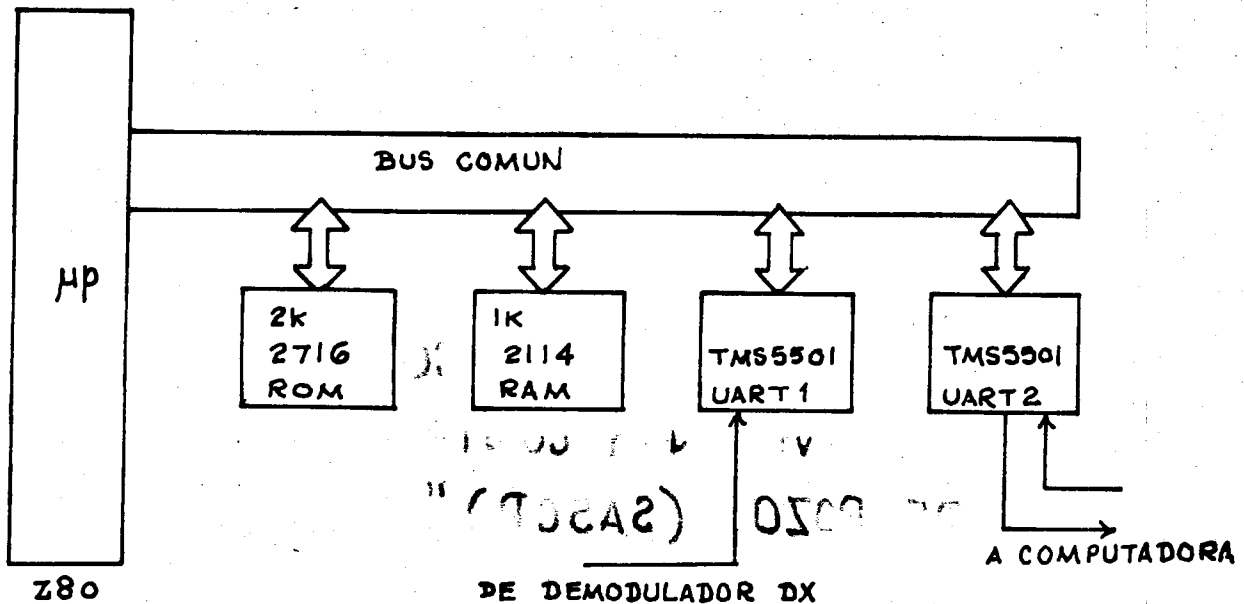
### DEMODULADOR DX



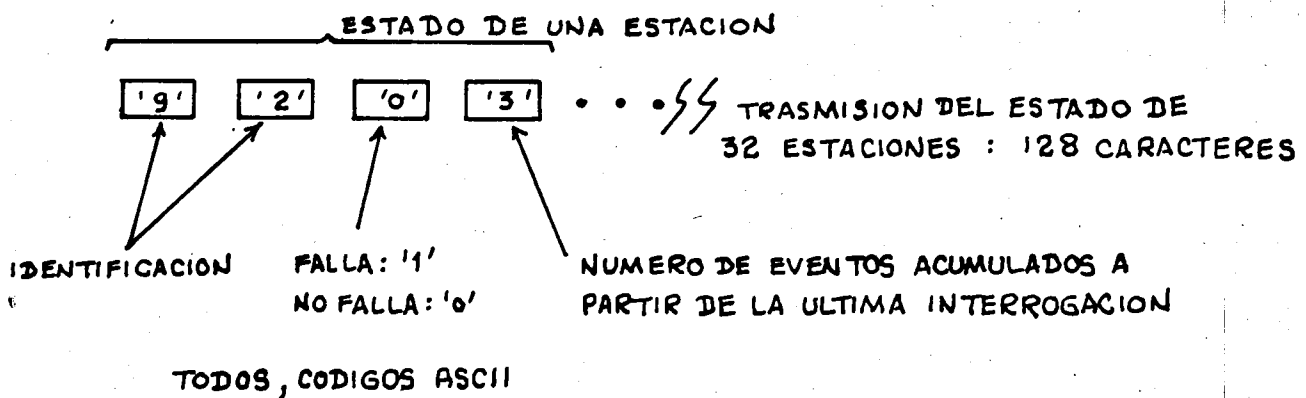
SEÑAL 300 BAUDS

### TERMINAL DE PRUEBA $\mu$ X





- 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 :

167

- \* 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 TELEFÓNICA PRIVADA .

A ESTE SISTEMA SE LE DENOMINÓ SASCP

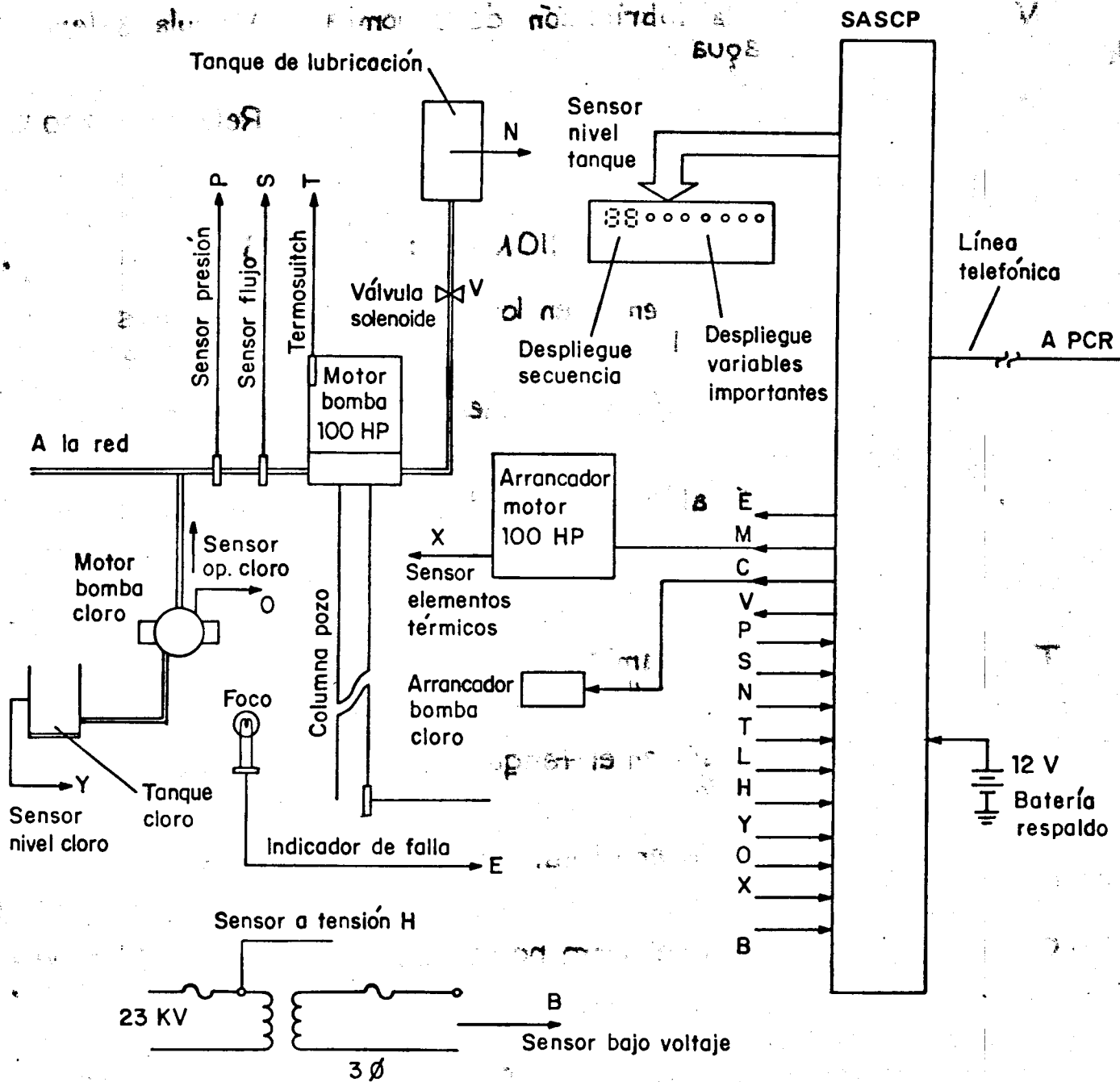


Fig 1 Diagrama instalación pozo municipal

## VARIABLE:

DISPOSITIVOS A  
OPERAR:

## ACTUADOR:

M	Motor bomba de agua (100 HP)	Arrancador 100 HI
C	Motor bomba cloro (3/4 HP)	Arrancador 3/4 H.F
V	Válvula lubricación de la bomba de agua	Válvula Solenoide
E	Foco rojo externo para indicación de falla.	Relevador 200 W.

## INTERRUPCIONES:

## SENSOR:

X A	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 m

## 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 autorestabiliza.

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  $IF = 0$  , no existe falla .

si  $IF = 1$  , existe alguna falla .

si  $\Phi = 0$  , no existe interrupción .

si  $\Phi = 1$  , existe alguna interrupción .

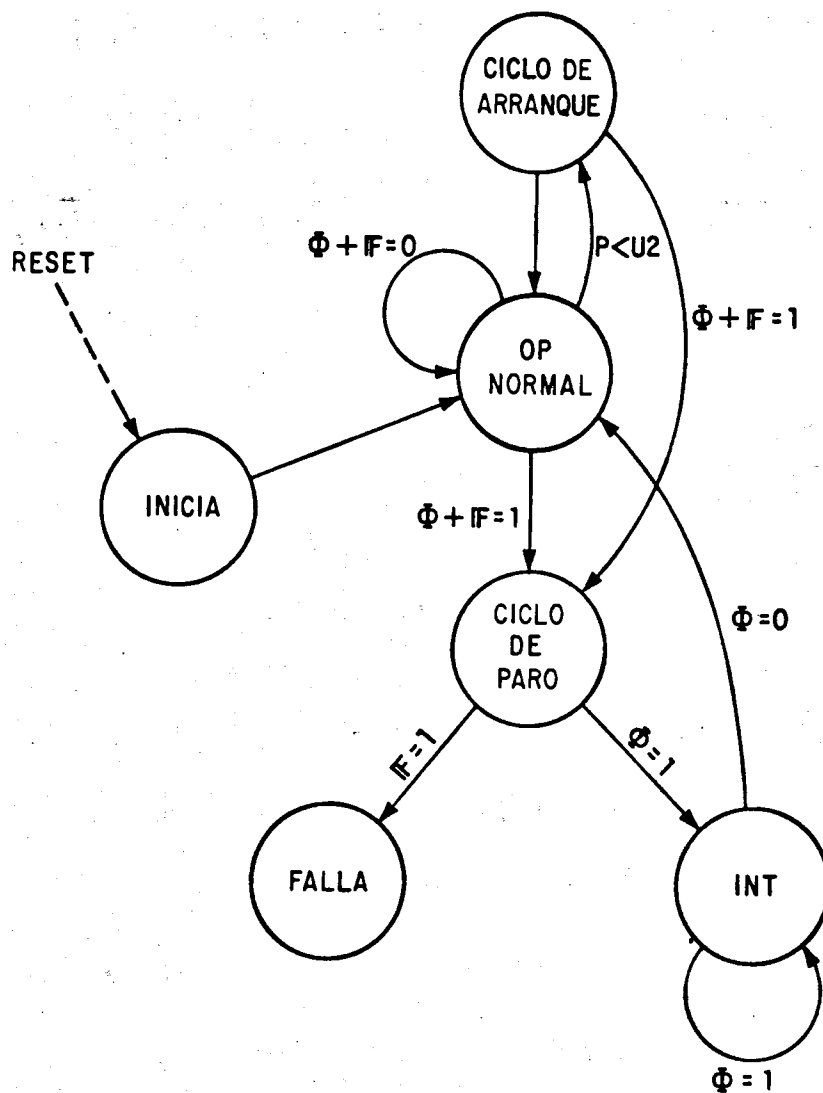
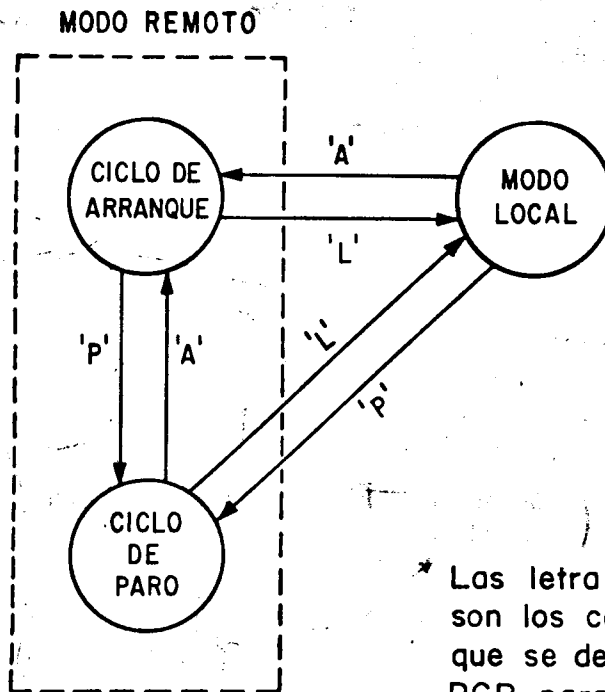


Fig 3 Diagrama de estados del controlador SASCP (modo local)



\* Las letras entre comilladas son los caracteres ASCII que se deben transmitir del PCR para pasar de un estado a otro

Fig 4 Diagrama de estados para el modo remoto de SASCP

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

El PCR Transmite  
(códigos ASCII):

'CNTR G'

'A'

'P'

'L'



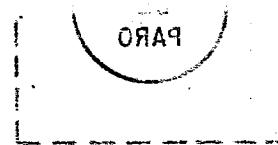
El SASCP responde:

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).



Las letras en cursiva son los caracteres ASCII que se deben transmitir del PCR al pozo.

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

FALLA 5

MOTOR	0	PRESION	19
COLOR	0	FLUJO	30
VALVULA	0	U. FLUJO	50

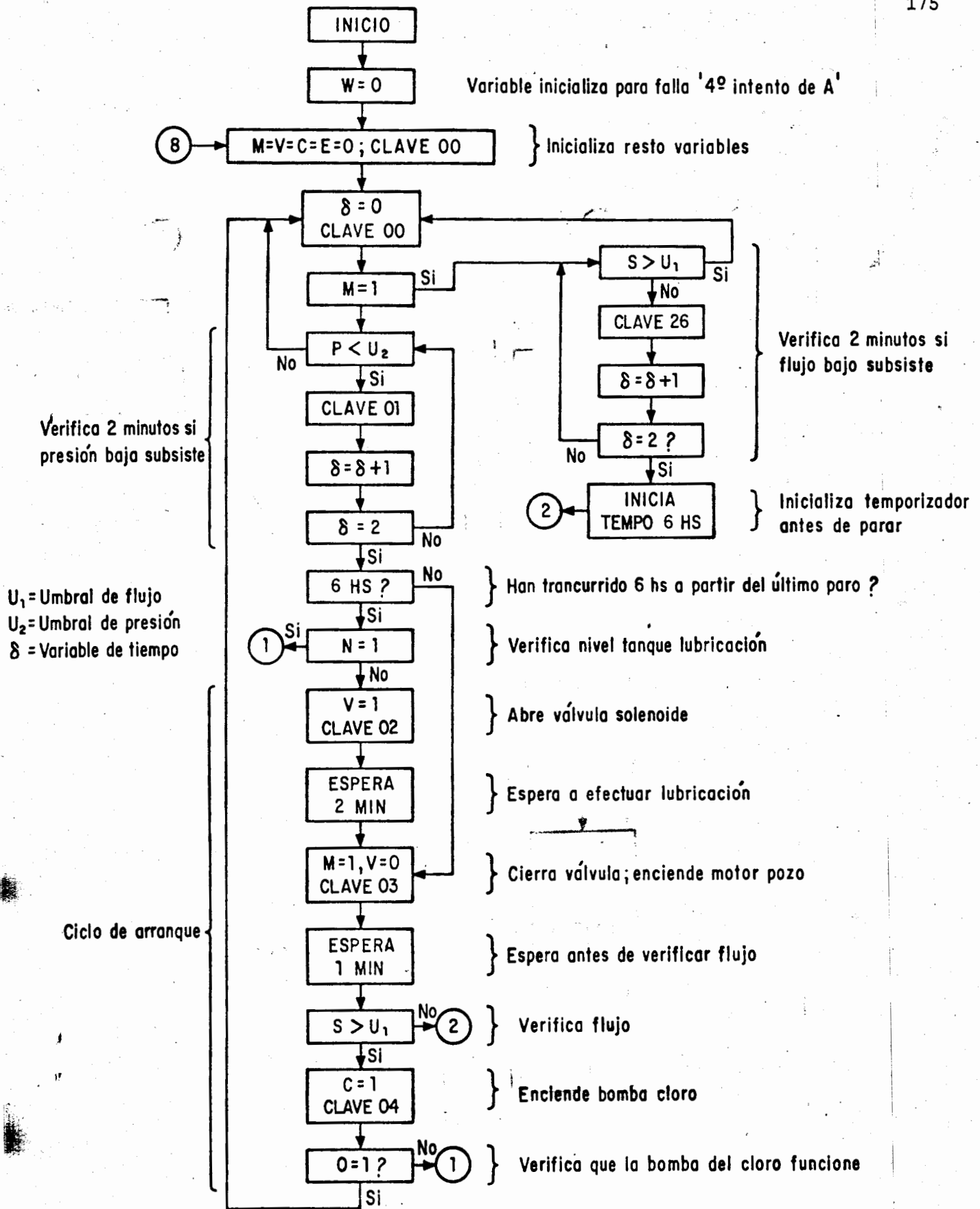


Fig 5 Diagrama de flujo, programa principal del SASCP



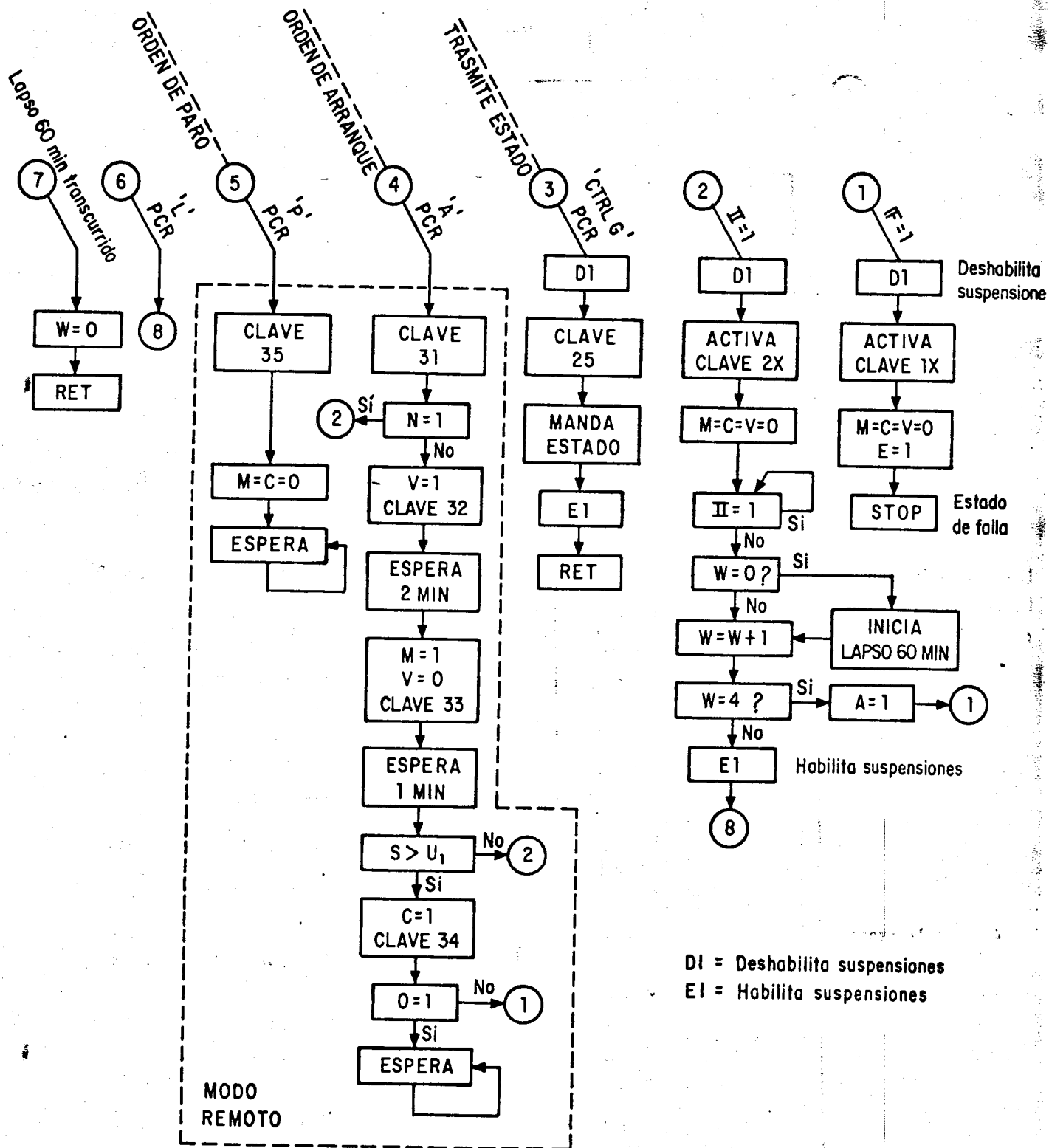


Fig 6 Diagrama de flujo, suspensiones en el programa del SASCP

DISEÑO ELECTRÓNICO DEL SISTEMA SASCP :

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

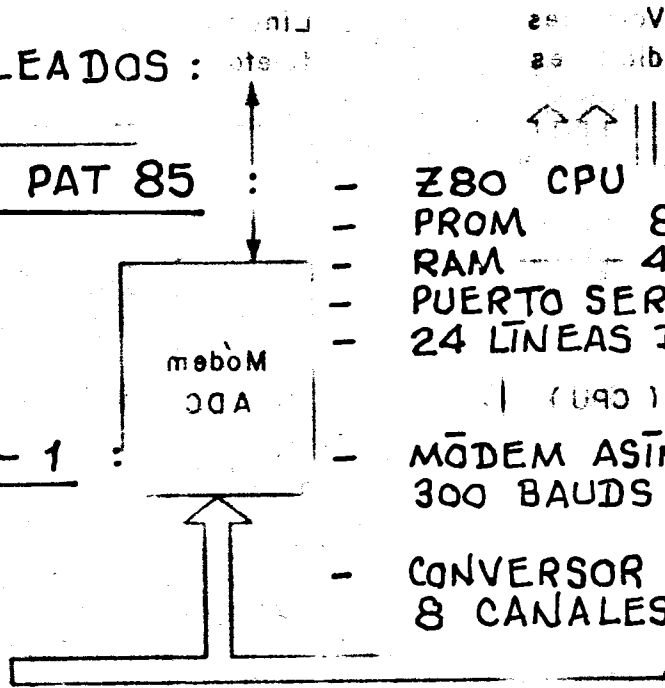
MÓDULOS EMPLEADOS :

TARJETA CPU PAT 85 :

- Z80 CPU
- PROM 8K
- RAM 4K
- PUERTO SERIE
- 24 LÍNEAS DE E/S

TARJETA MO - 1 :

- MÓDEM ASÍNCRONO 300 BAUDS , HALF DUPLEX
- CONVERTOR A/D DE 8 CANALES



BUS COMÚN :

OMNIBUS

SOPORTE DE PROGRAMACIÓN:-

INTÉRPRETE BASIC RESIDENTE (4K PROM)

902A2 robototnoo le ne "2.31110" ome 2 10 n

FUENTE DE PODER FO-1

- CON SOPORTE DE BATERIA 12V, CON CAP. PARA 24 HS.

DISEÑO ELECTRÓNICO DEL SISTEMA SASCP

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

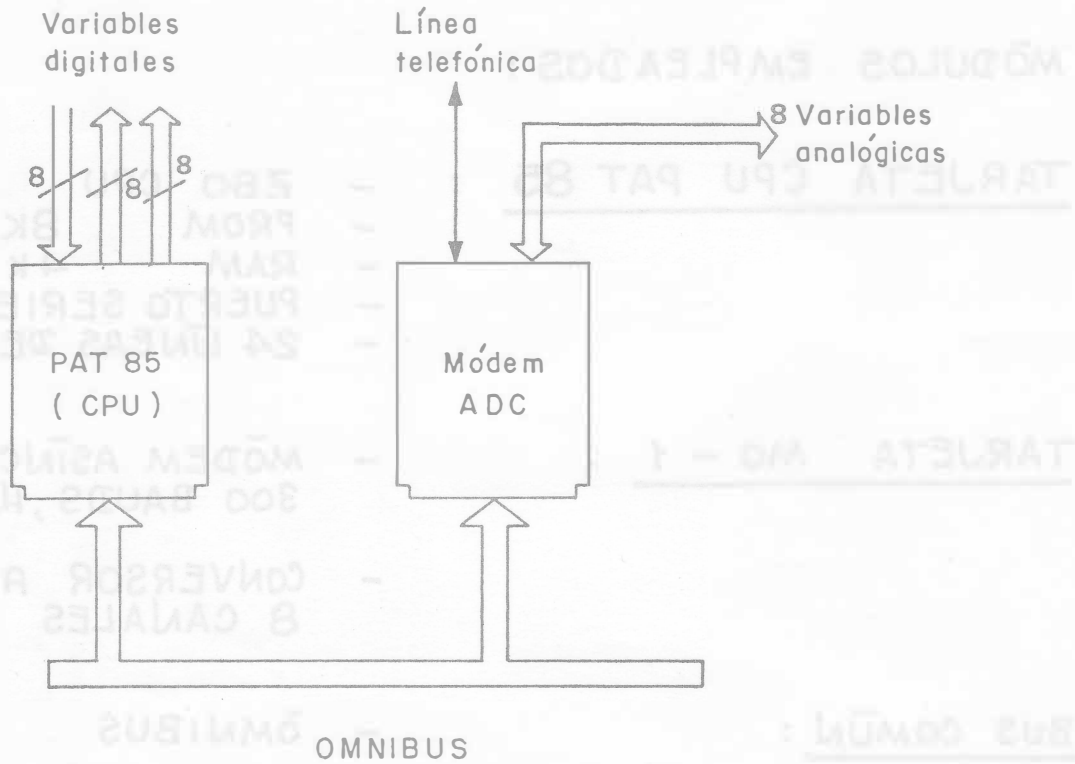


Fig 2 Configuración del sistema "OMNIBUS" en el controlador SASCP

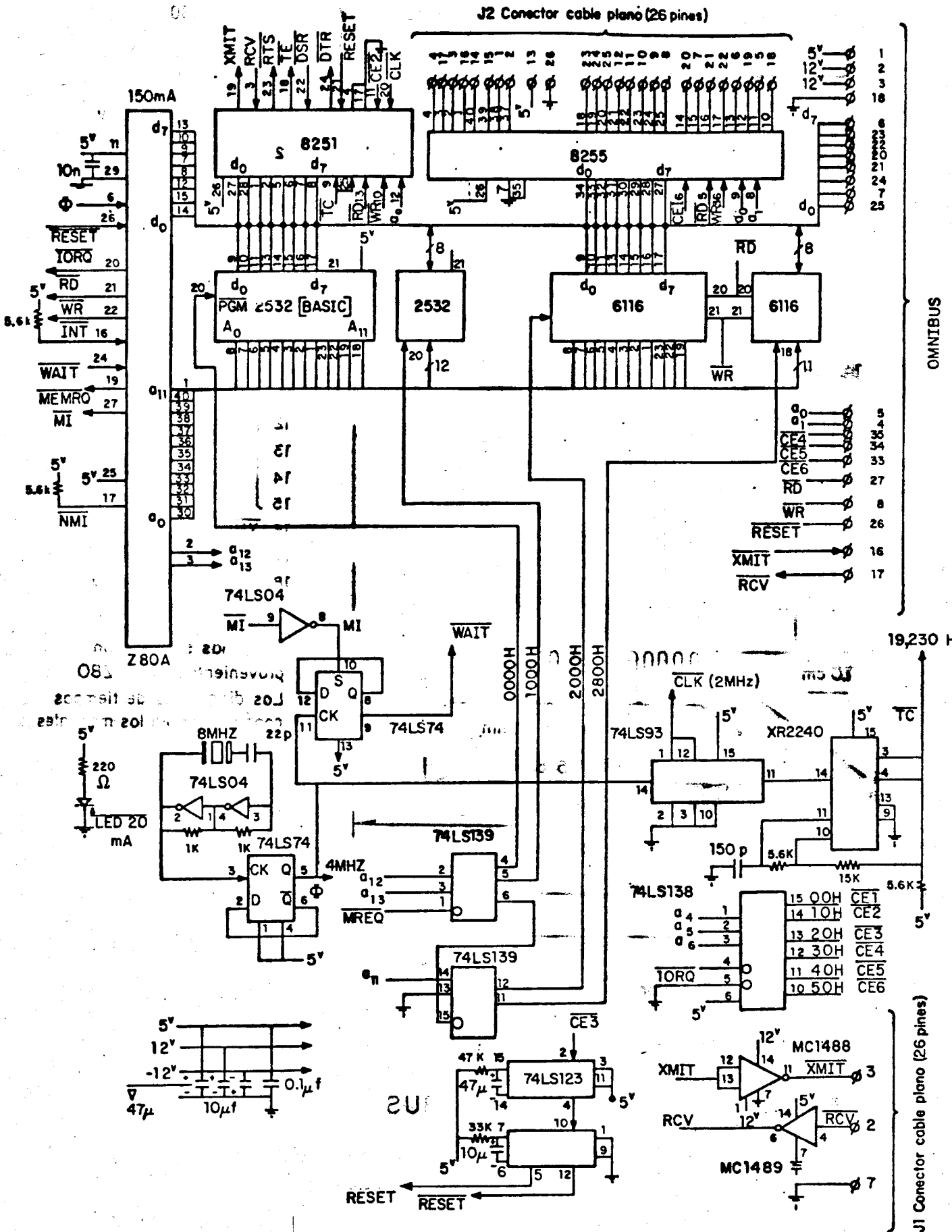
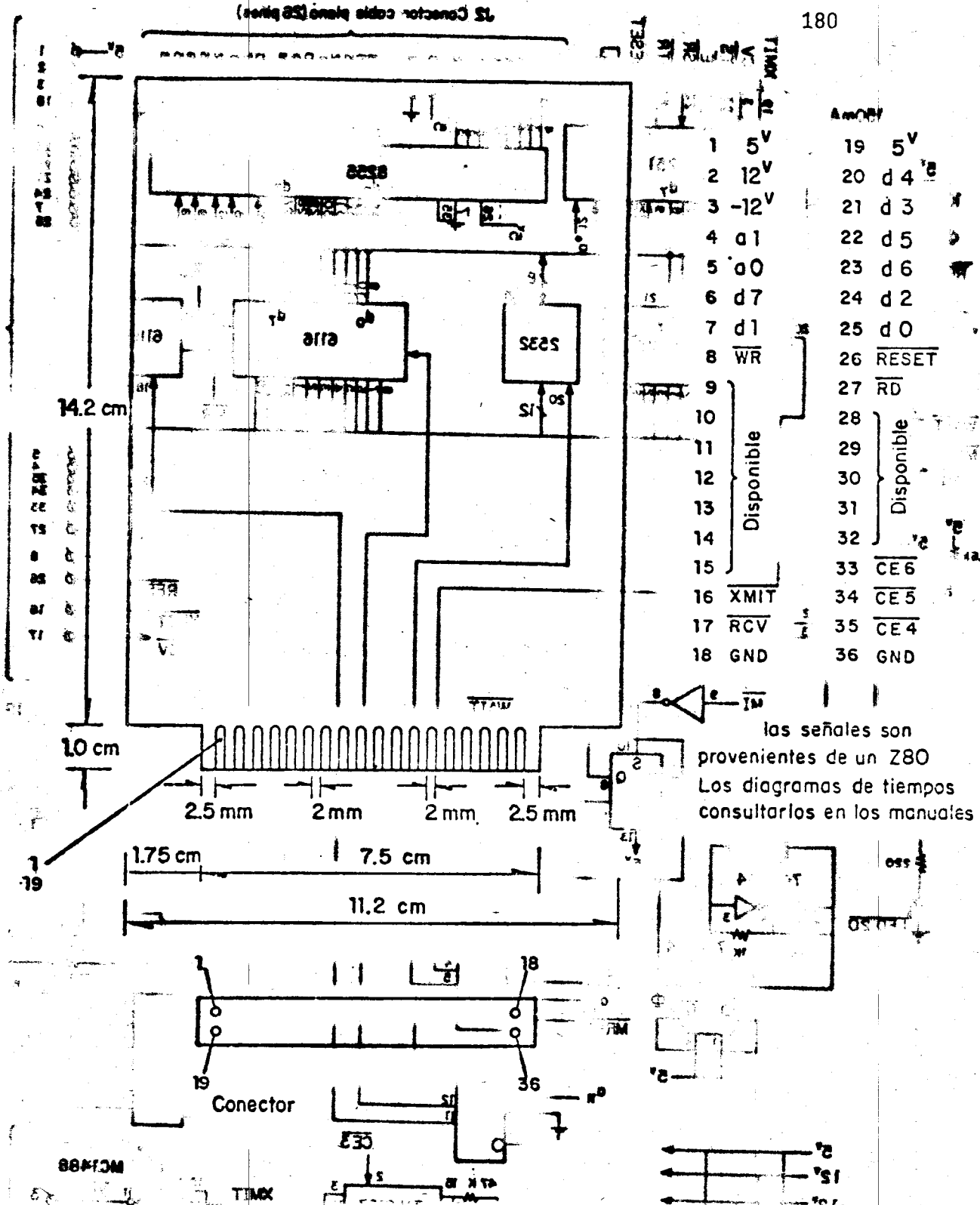


Fig A2 Sistema PAT 85



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

Fig A1 Módulo estándar OMNIBUS

EJERCICIO 1

haga las siguientes operaciones en binario utilizando representación complemento a 2 en 8 bits. Indique el signo overflow. Todos los números son decimales.

101 - 101  
101 - 101  
101 - 101

Con el formato indicado, realice la representación en binario de los números decimales dados

EJERCICIOS 1, 2, 3 Y 4.



101.101

fracción

entero

11,200.8



exp

mantisa

signo  
mantisa

FAC. DE INGENIERIA  
DOCUMENTACION

para un sistema de signo por decada (significando) en una suma de números en representación complemento a 2. Asigne las variables A como el bit del signo del primer sumando, B del segundo y C del resultado. Calcule y indique un programa para el 180 que cuente del 0 al 255 en binario y indique el resultado de cada incremento en el registro C. Calcule el tiempo de ejecución del programa si el pc se encuentra conectado a 3 MHz.

## EJERCICIO 1

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

-58  
-110

20  
-89

127  
-128

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 detecte 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.





## 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 inic:

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  $x/256$  SEG. 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 código  $XX_{16}$ , encontrarlo (dar su dirección) en una tabla de 256 caracteres.

7. Codifique las siguientes instrucciones.

Dirección

E1D

⋮

E5A

UNO:

⋮

DOS:

JR Z, 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 OAH. de E/S.

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

eventos:

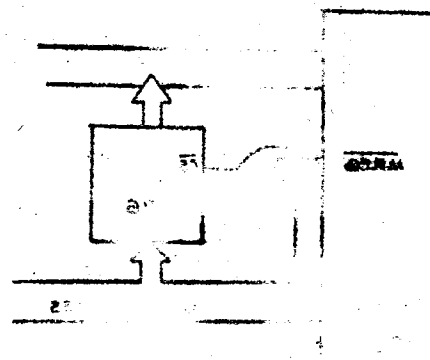
- El temporizador decrementa el contador una unidad cada  $64 \mu s$ .
- Cuando el contenido del contador es cero, el temporizador activa la línea  $\overline{INT}$  del Z80.
- Al recibir el reconocimiento de interrupción, el temporizador pone en el bus de datos el código CPH y regresa la línea de  $\overline{INT}$  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.



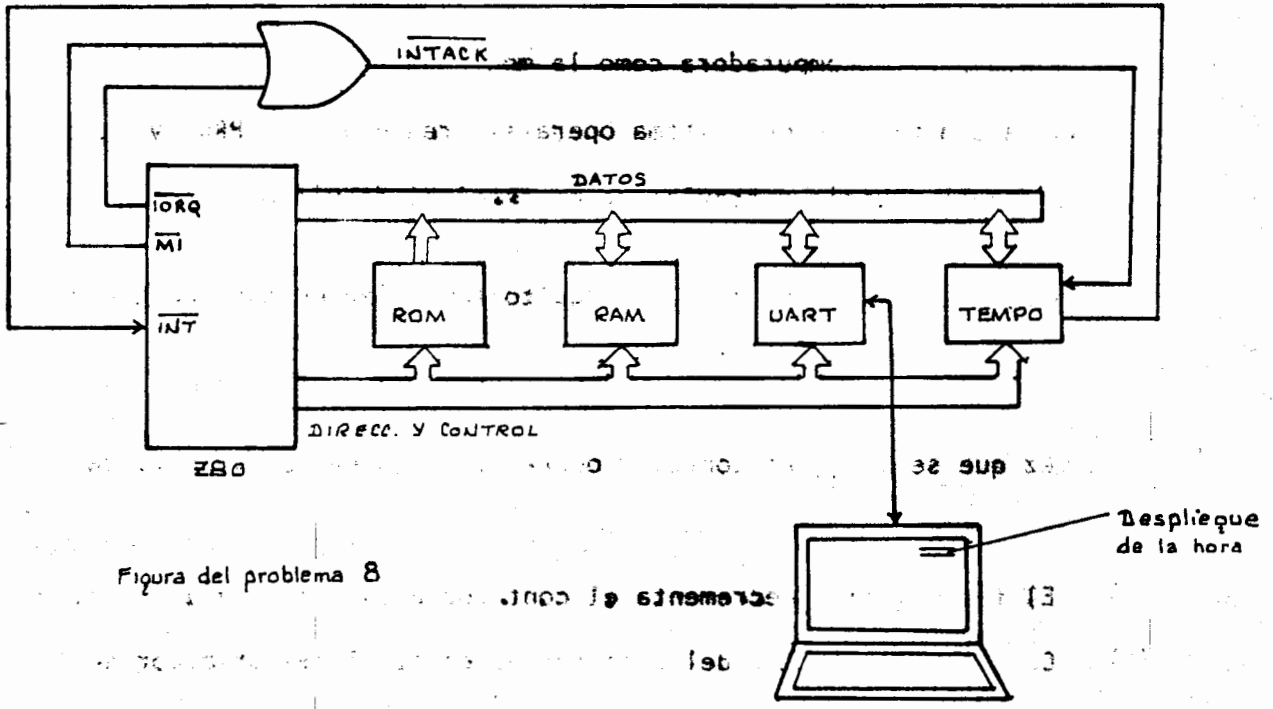
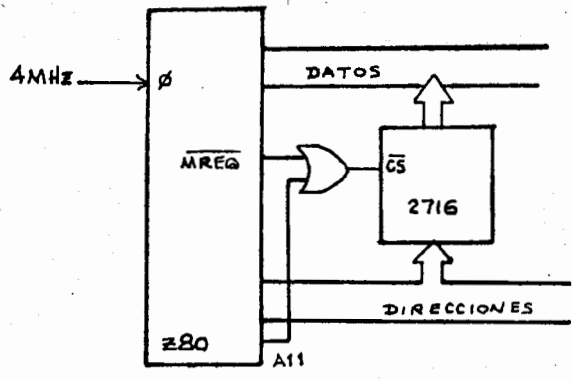


Figura del problema 8

9. Suponga un  $\mu p$  Z80 trabajando a 4MHz y conectado a una memoria ROM 2516 de la manera indicada en la figura.

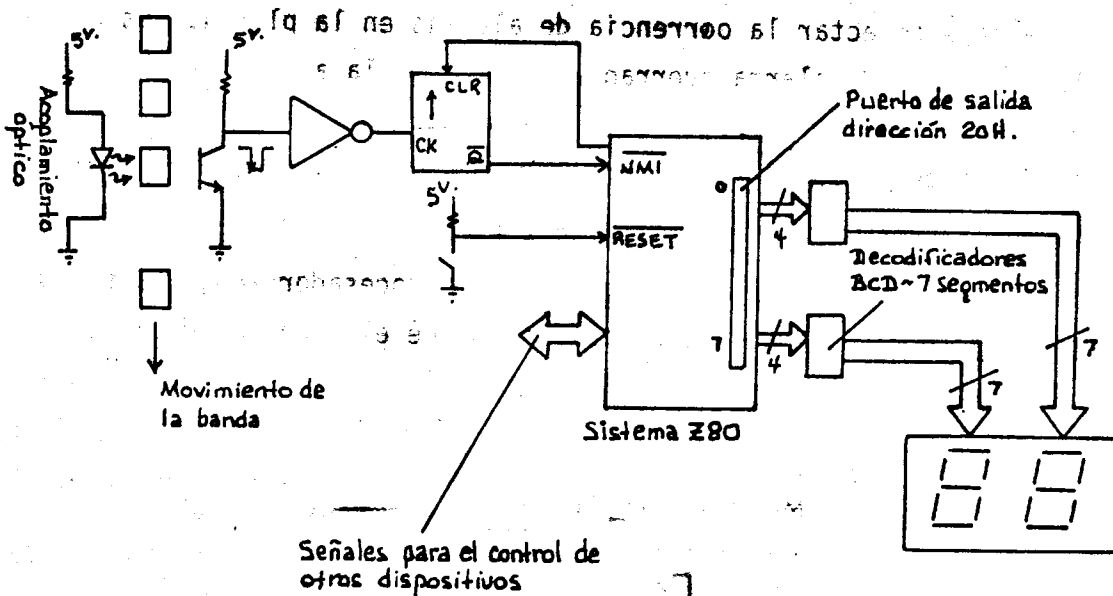
\* Diga si el circuito funciona correctamente durante los ciclos de lectura, en particular el ciclo 'FETCH'. Justifique su respuesta con diagramas de tiempos y proponga alternativas en caso que existan problemas.

\* Indique en que direcciones se habilita la memoria.

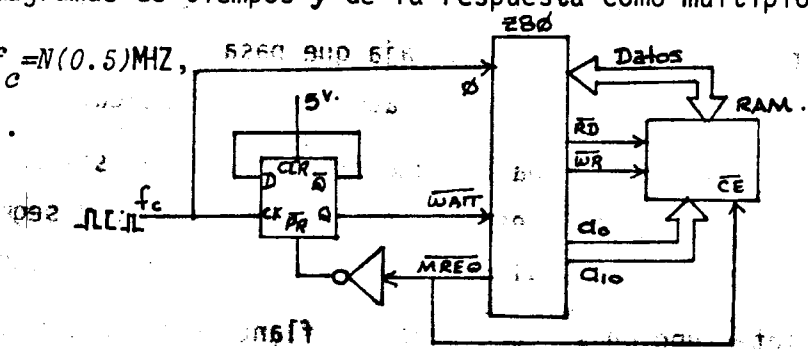


10. En una fábrica de tornillos, se tiene una microcomputadora realizando diversas tareas de control en tiempo real. Una de estas tareas, que se efectúa por interrupción, concierne la cuenta de cajas que pasan por la línea de producción: un detector óptico genera un pulso cada vez que una caja que pasa,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

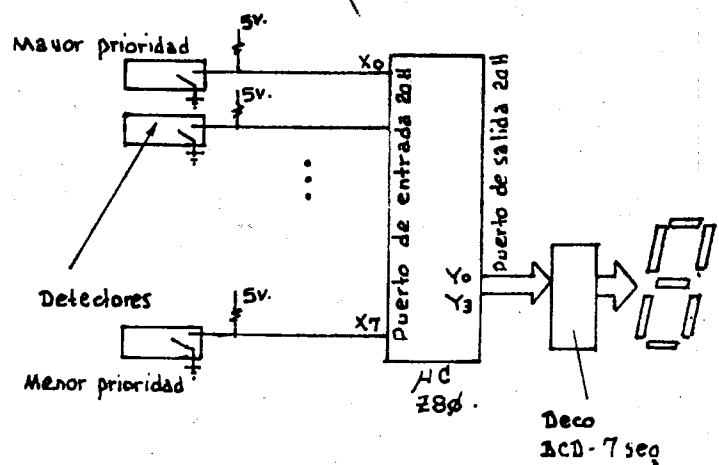


11. El Z80 de la figura debe ejecutar un programa almacenado en una memoria EPROM 2716 cuyo tiempo de acceso [máx] es de 500 nsec. Determine la frecuencia máxima a la que puede trabajar el microprocesador. Justifique su solución con diagramas de tiempos y de la respuesta como múltiplo de 0.5 MHz; es decir  $f_c = N(0.5)MHz$ , donde  $N=1, 2, 3...$



12. La microcomputadora de la figura se encuentra realizando tareas de control en una planta de sustancia químicas. Entre otras cosas, la  $\mu C$  ejecuta una subrutina que debe detectar la ocurrencia de alarmas en la planta. Los detectores de la condición de alarma cierran un switch si la alarma está activa. La  $\mu C$  debe detectar el cierre del switch y escribir en el módulo de 7 segmentos, el número de la alarma activa *con mayor prioridad*. Si no hay alarma activa, debe escribirse un cero en el despliegue.

Escriba, con un mínimo de instrucciones, para el procesador Z80, una subrutina que ejecute la tarea descrita arriba. Acompañe el programa de comentarios para facilitar su comprensión. No es necesario codificar las instrucciones.

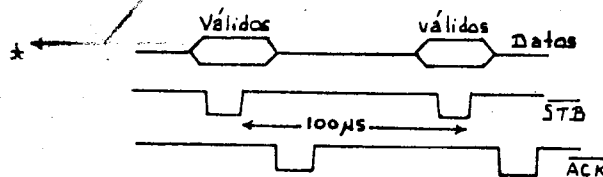
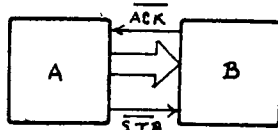


EJERCICIO 3

- 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:

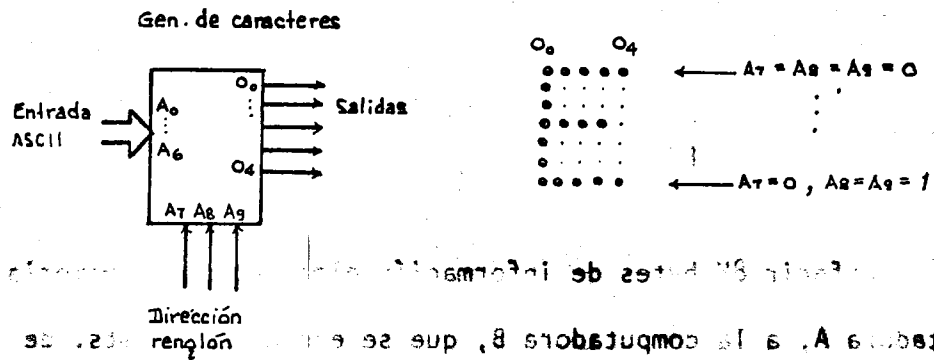
- Tiempo total de transmisión empleando un acoplamiento 'CENTRONIX' (vel. máx).
- Velocidad de transmisión serie (asíncrono) para igualar el tiempo de transmisión en (a).
- Circuitos y metros de cable empleados en cada caso.



Acoplamiento Centronix

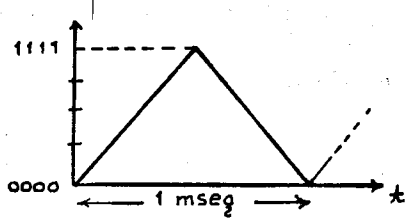
- La figura muestra un generador de caracteres para matriz de puntos de 5 x 7. Para el código de la E (45H) llene la siguiente tabla.

REGLON	DIRECCION [EN HEXADecimal]	CODIGO (0...0 <sub>4</sub> )
1		
2		
3		
4		
5		
6		
7		



los caracteres A a la computadora B, que se encarga de transmitirlos en serie (bit a bit) a la computadora A.

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  $\phi_1$  y  $\phi_2$ ?



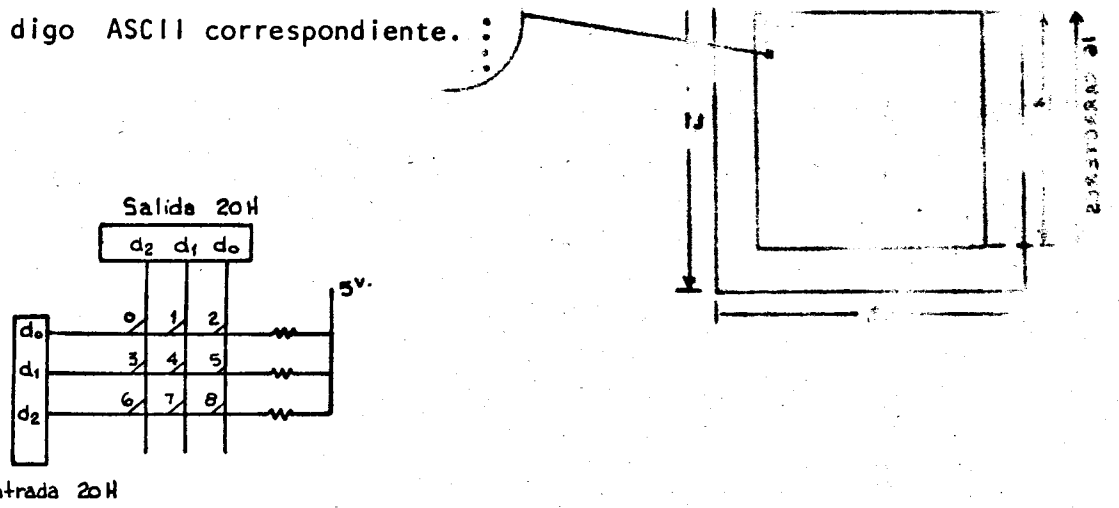
4. Explique claramente el funcionamiento del circuito de la fig. 3.6 (PERIF P/MICROPROC), indicando la función de cada módulo.
5. Basándose en la información contenida en el reporte 'Control de CRT' diga cual es el valor de N de cada contador divisor/N mostrado en la fig. 10.
6. Escriba un programa para el up Z80 que siga el algoritmo de aproximaciones sucesivas de la fig. 4.4 (PERIF P/up).

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  $d_0$ ).

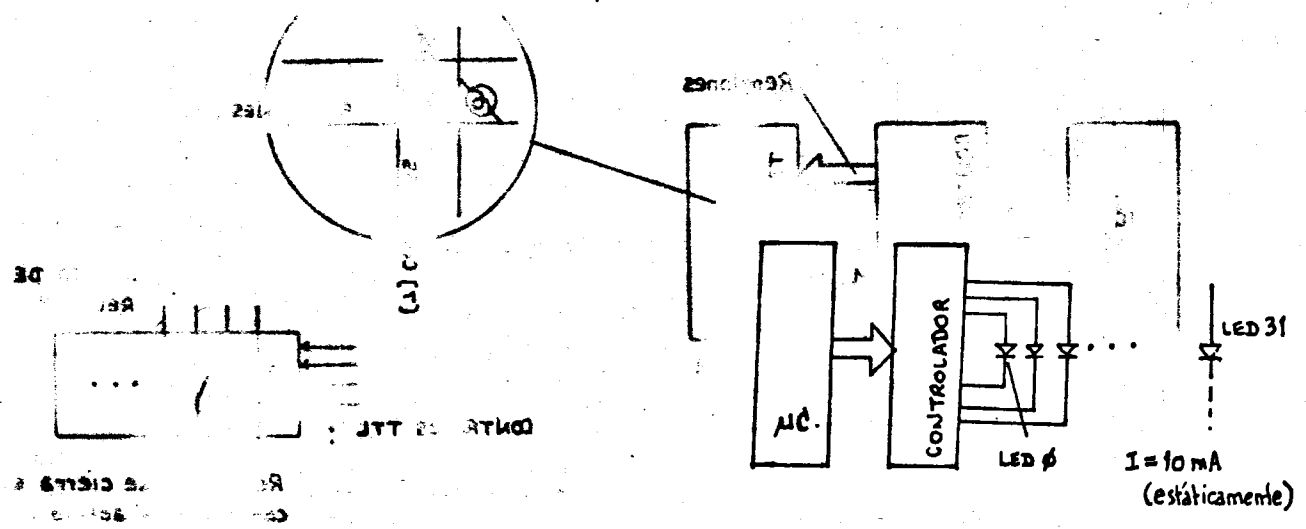
10. Se tiene un TV casera que tiene un formato de imagen como el que se indica para la zona de

7. Explique de manera resumida, el principio de funcionamiento de los despliegues fosforescentes ITRON.

8. Escriba un programa que identifique que tecla se oprimió en un teclado 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 código ASCII correspondiente.



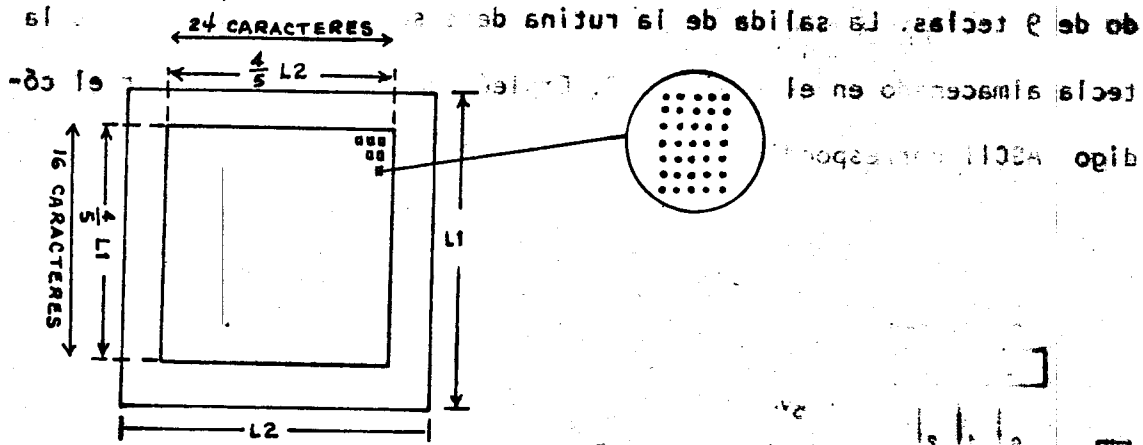
9. En un tablero para equipo de computación hay 32 LEDS en línea. Cada LED puede encenderse y apagarse independientemente del estado del resto. Diseñe un controlador que se acople a una microcomputadora y realice la tarea anterior usando circuitos multiplicados.



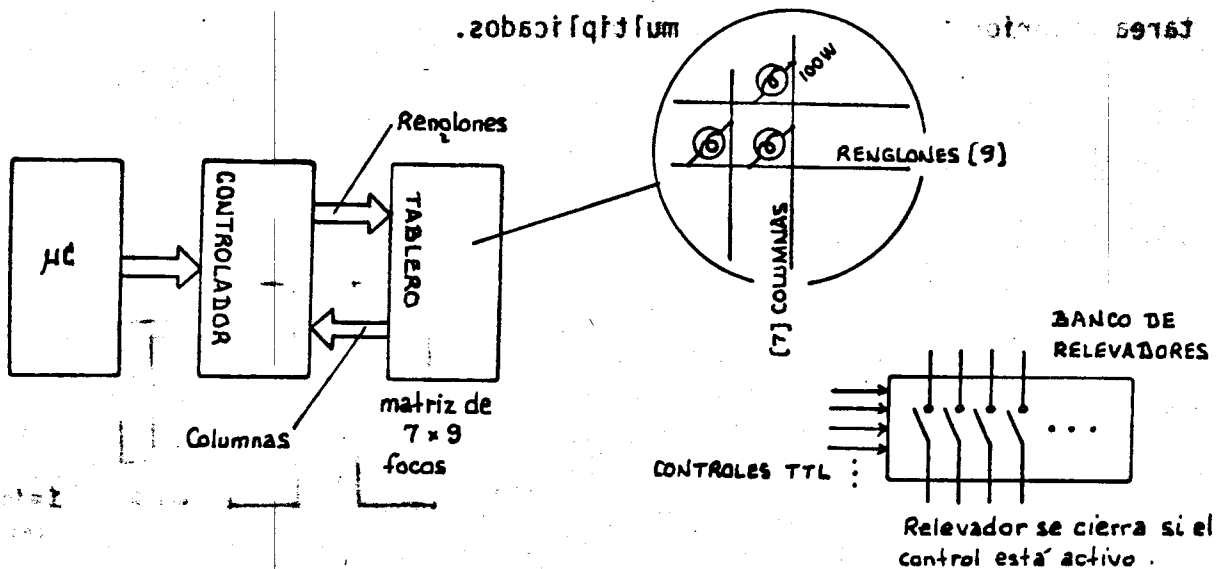


10. Se tiene una TV casera acoplada a un sistema de computación.  
 Si se emplea un formato como el que se indica para la zona de escritura,

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



11. 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.



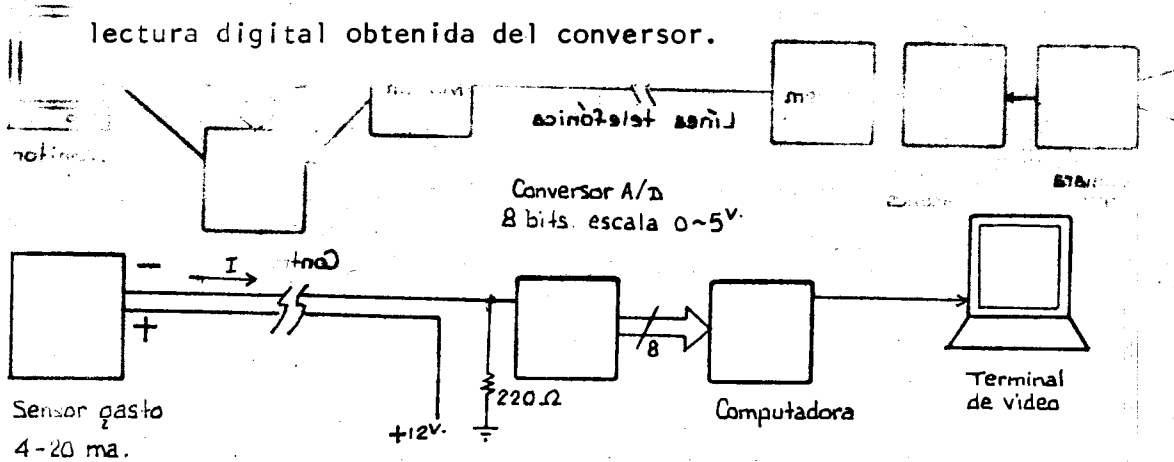
## EJERCICIO 4

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.



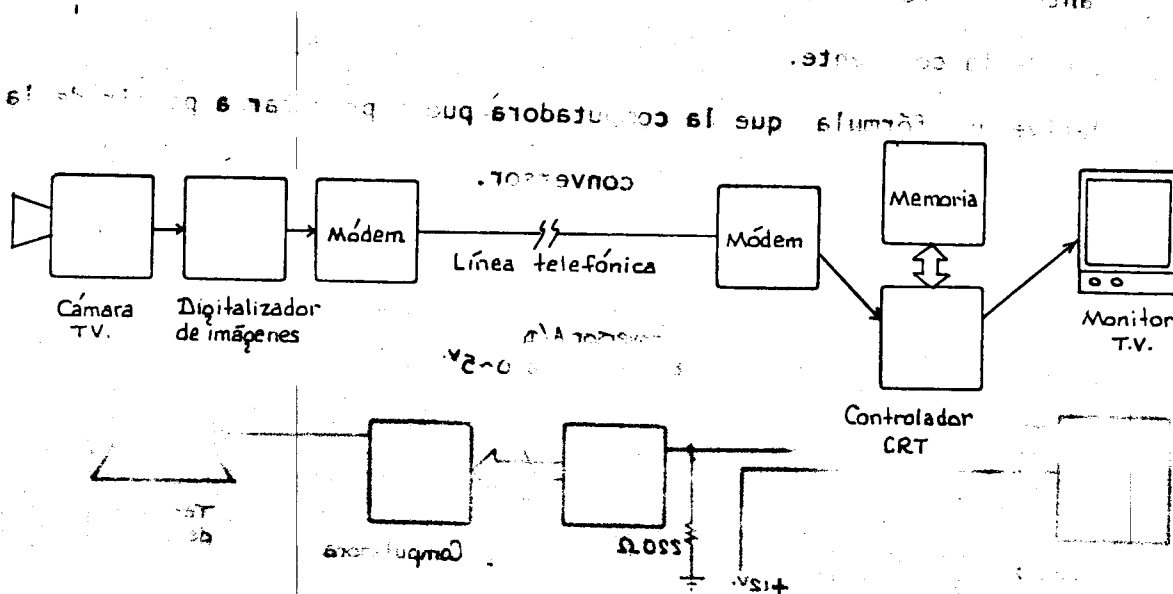
$$Q [M^3/SEG] = 0.57 \sqrt{1.25(I - 4)}$$

$$I \text{ en } [mA]$$

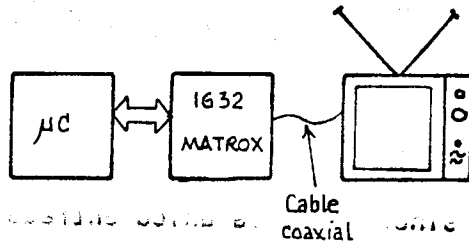
(último renglón).

EXERCICIO 4

4. Se tiene un sistema de transmisión de imágenes monocromáticas de TV. Cada imagen está formada por una matriz de 256 x 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

El control del motor es controlado por un sistema de video que genera una señal

TTL que automáticamente enciende o apaga el motor.

En un análisis completo del sistema incluido los siguientes puntos:

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.

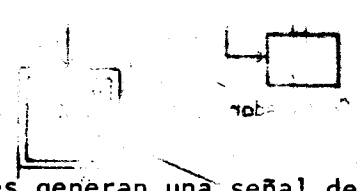
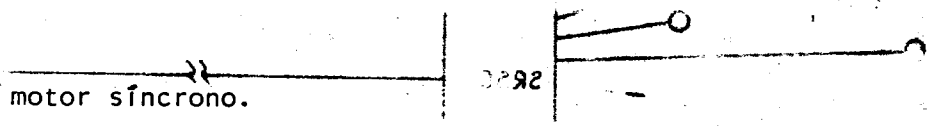
Sobre petición de la computadora central, el SRSC tendrá capacidad de:

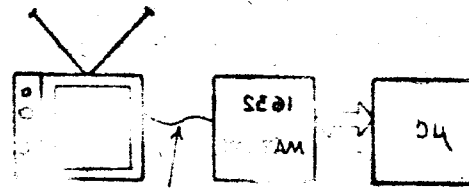
- Transmitir a la computadora central los valores de las ocho variables en forma codificada.
- Encender ó apagar el motor síncrono.

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.

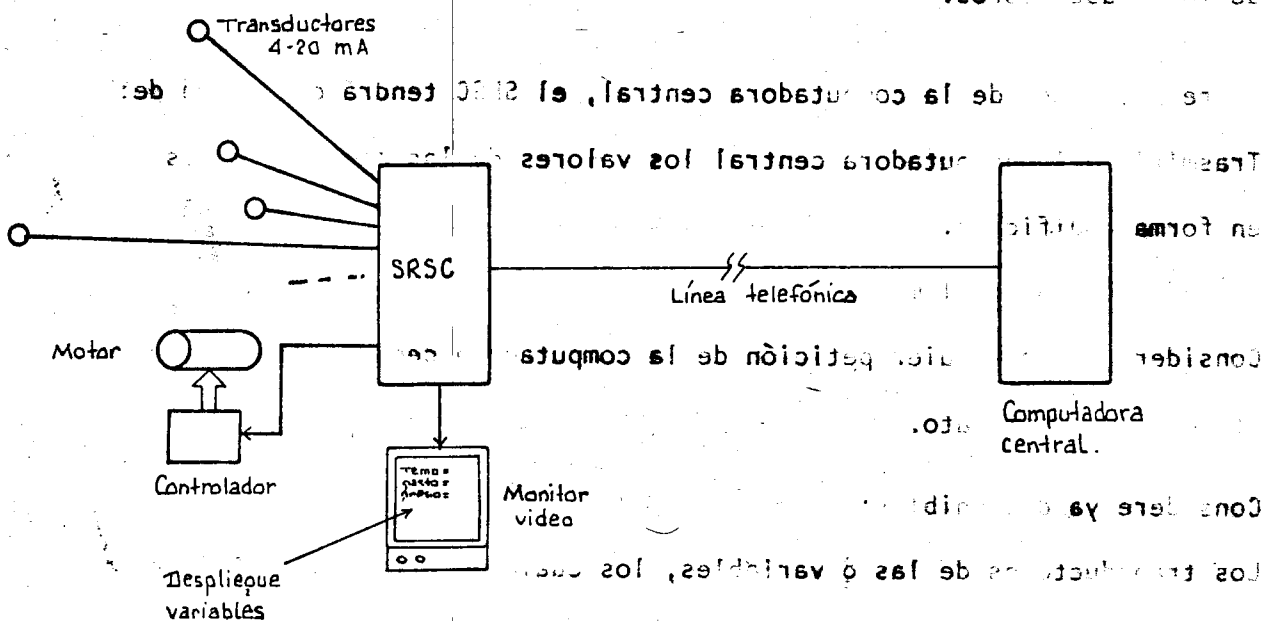




-El controlador del motor síncrono, cuya única entrada es una señal TTL que automáticamente enciende o apaga el motor.

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.

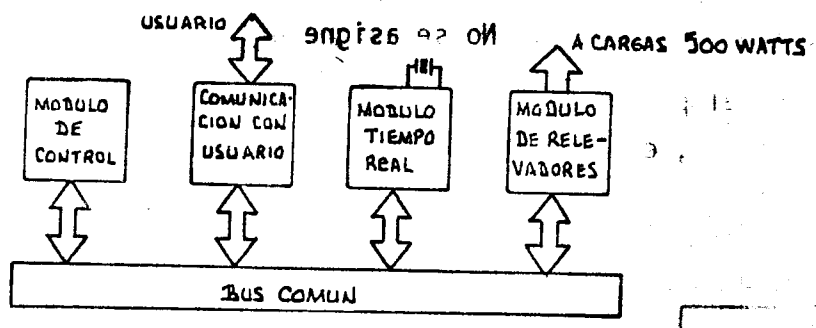


7. En una fábrica de productos plásticos se requiere de un temporizador industrial. Este equipo debe tener capacidad para encender y/o apagar cualesquiera de hasta 8 cargas de 500 watts (120 v. A.C.) cada una como máximo, en instantes previamente programados por el usuario y repetidos diariamente, con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



FAC. DE INGENIERIA  
DOCUMENTACION

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.

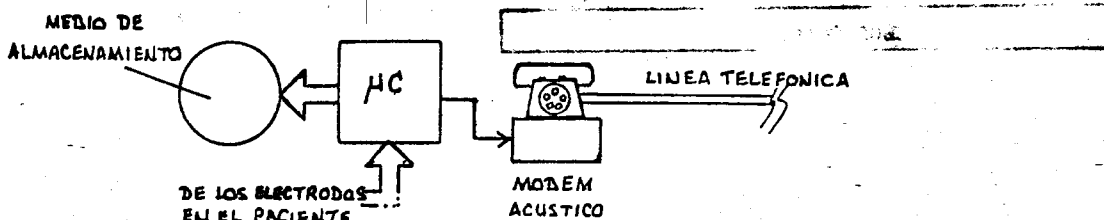
Posteriormente, ya en el mismo hospital, se haría la lectura directa del medio de almacenamiento, por una computadora que se encargaría de procesar la señal y dar eventualmente algún diagnóstico.

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 utilizando 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.



9. 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 pasos: el motor es manejado por un controlador que requiere de pulsos con nivel TTL a su entrada. Cada pulso hace avanzar el motor  $3.6^\circ$ , de tal forma que con 100 pulsos se tiene una revolución completa. Las velocidades de giro van 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..

Y 50091

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.

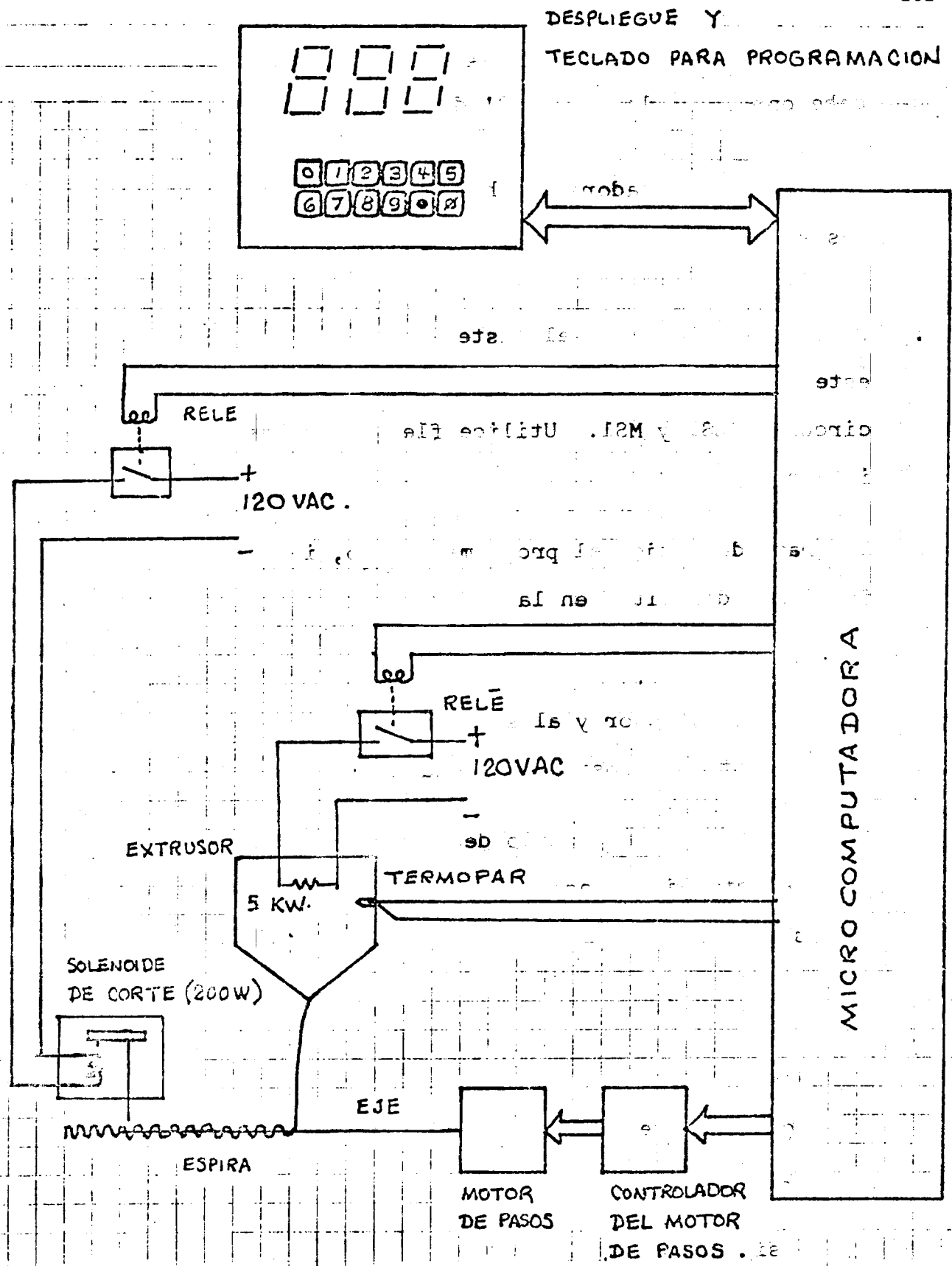


FIG 1 SISTEMA DE FABRICACION DE ESPIRALES DE PLASTICO

# 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

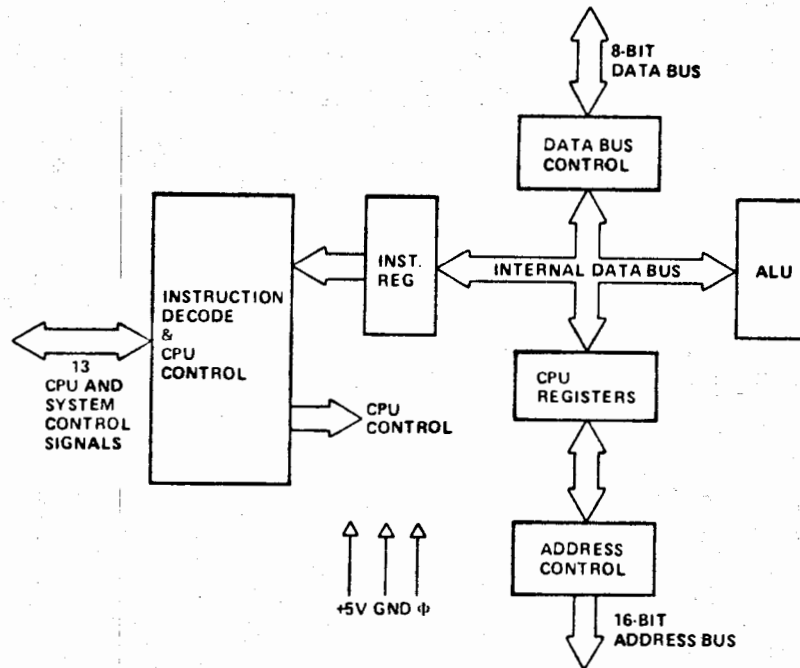


FIGURE 2.0-1

### 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.

## Z80-CPU REGISTER CONFIGURATION

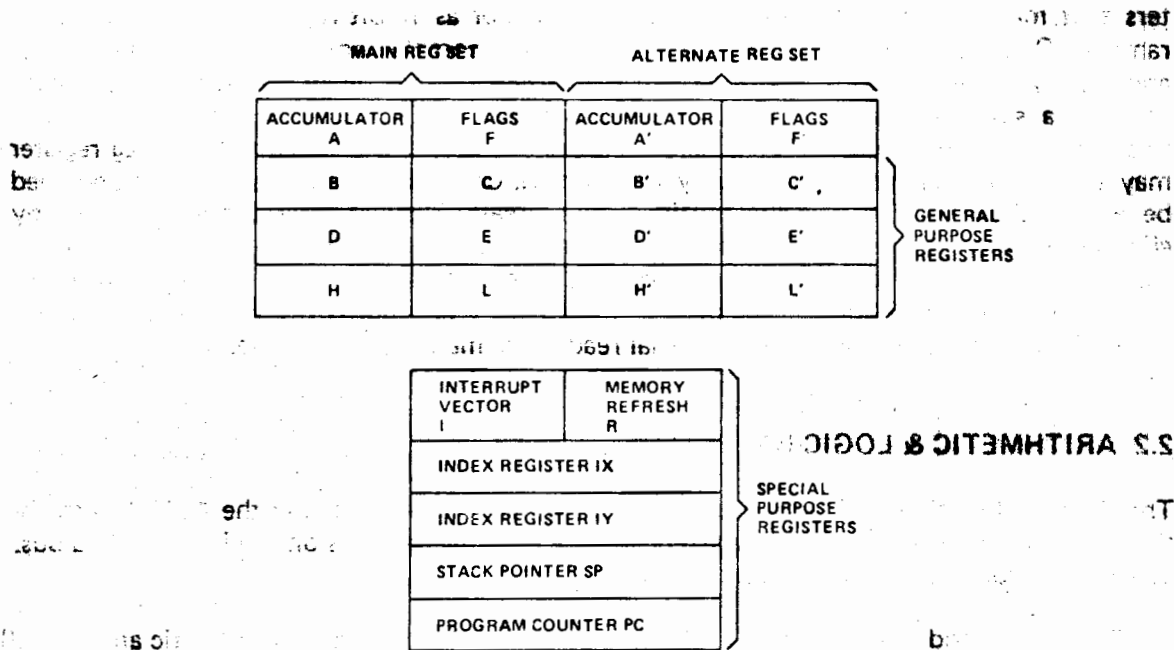


FIGURE 2.0-2

- 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.

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.

Z80 PIN CONFIGURATION

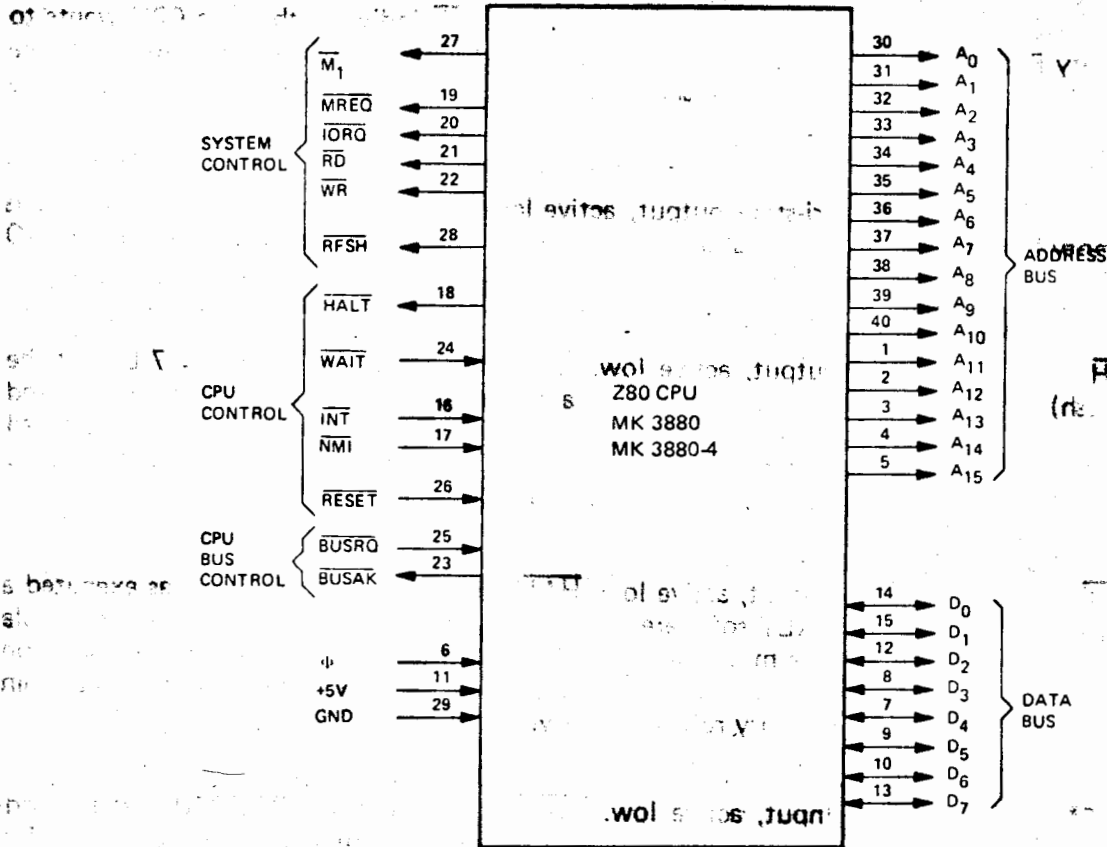


FIGURE 3.0-1

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.

$\overline{M}_1$   
(Machine Cycle one)

Output, active low.  $\overline{M}_1$  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,  $\overline{M}_1$  is generated as each op code byte is fetched. These two byte op-codes always begin with CBH, DDH, EDH, or FDH.  $\overline{M}_1$  also occurs with  $\overline{IORQ}$  to indicate an interrupt acknowledge cycle.

$\overline{MREQ}$   
(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.



**$\overline{\text{IORQ}}$**   
(Input/Output Request)

Tri-state output, active low. The  $\overline{\text{IORQ}}$  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  $\overline{\text{IORQ}}$  signal is also generated with an  $\overline{\text{M}_1}$  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  $\text{M}_1$  time while I/O operations never occur during  $\text{M}_1$  time.

**$\overline{\text{RD}}$**   
(Memory Read)

Tri-state output, active low.  $\overline{\text{RD}}$  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.

**$\overline{\text{WR}}$**   
(Memory Write)

Tri-state output, active low.  $\overline{\text{WR}}$  indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.

**$\overline{\text{RFSH}}$**   
(Refresh)

Output, active low.  $\overline{\text{RFSH}}$  indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and current  $\overline{\text{MREQ}}$  signal should be used to do a refresh read to all dynamic memories.  $\text{A}_7$  is a logic zero and the upper 8 bits of the Address Bus contains the I Register.

**$\overline{\text{HALT}}$**   
(Halt state)

Output, active low.  $\overline{\text{HALT}}$  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.

**$\overline{\text{WAIT}}^*$**   
(Wait)

Input, active low.  $\overline{\text{WAIT}}$  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.

**$\overline{\text{INT}}$**   
(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  $\overline{\text{BUSRQ}}$  signal is not active. When the CPU accepts the interrupt, an acknowledge signal ( $\overline{\text{IORQ}}$  during  $\text{M}_1$  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.

**$\overline{\text{NMI}}$**

Input, negative edge triggered. The non maskable interrupt request line has a higher priority than  $\overline{\text{INT}}$  and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop.  $\overline{\text{NMI}}$  automatically forces the Z80-CPU to restart to location  $0066\text{H}$ . 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 WAIT cycles can prevent the current instruction from ending, and that a  $\overline{\text{BUSRQ}}$  will override a  $\overline{\text{NMI}}$ .

(Reverse)

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

- 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.

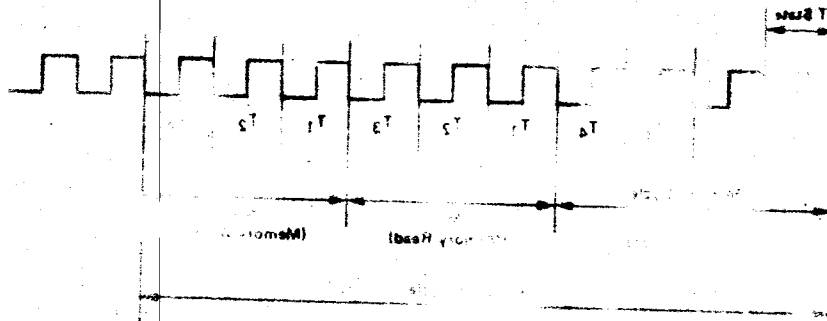
**BUSAK\***  
(Bus Acknowledge)

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.



All CPU timing can be predicted by the timing diagram. The timing diagram shows the relationship between the clock and the bus request (BUSRQ) and bus acknowledge (BUSAK) signals. The diagram also shows the timing of the address bus (A) and data bus (D) signals, which are active during the bus request period. The diagram includes labels for 'T1', 'T2', 'T3', 'T4', 'T5', and 'T6' representing different time intervals or machine cycles. A horizontal line at the bottom indicates the 'Memory Refresh' period, which is active during the bus request period.

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

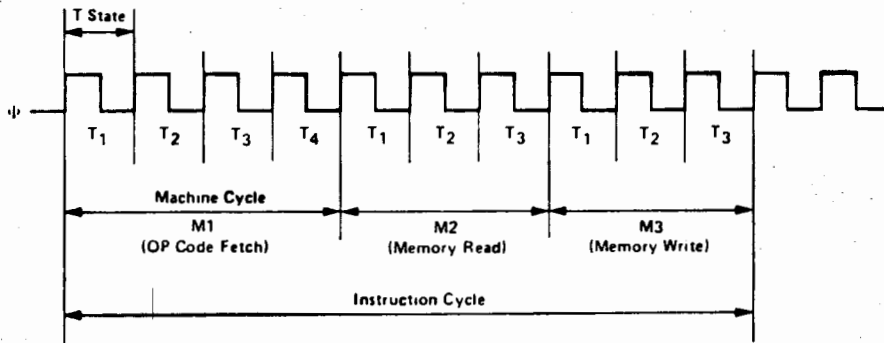


FIGURE 4.0-0

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

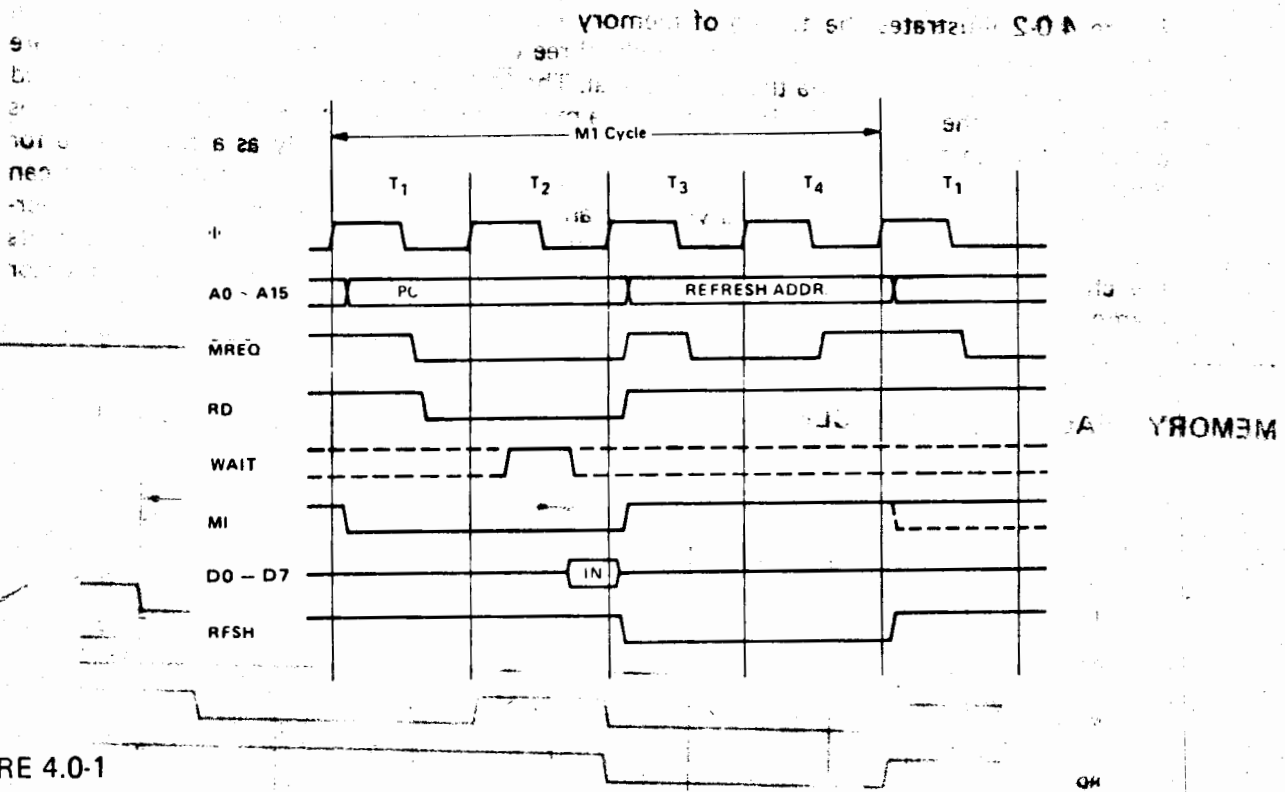


FIGURE 4.0-1

Figure 4.0-1A illustrates how the fetch cycle is delayed if the memory activates the  $\overline{WAIT}$  line. During T2 and every subsequent  $T_w$ , 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 4.0-1 shows the timing during

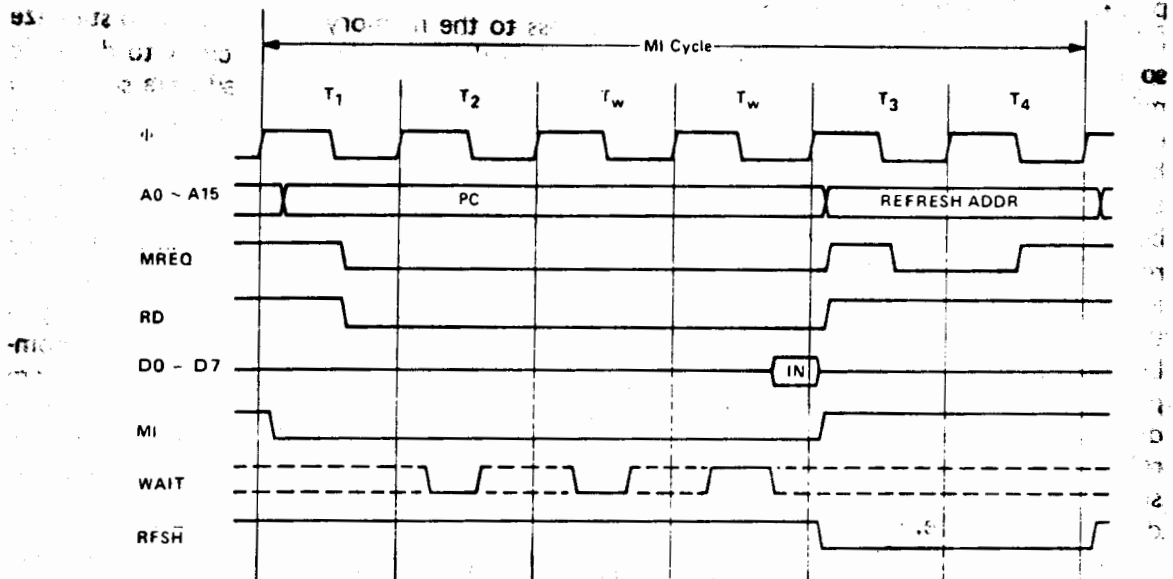


FIGURE 4.0-1A

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

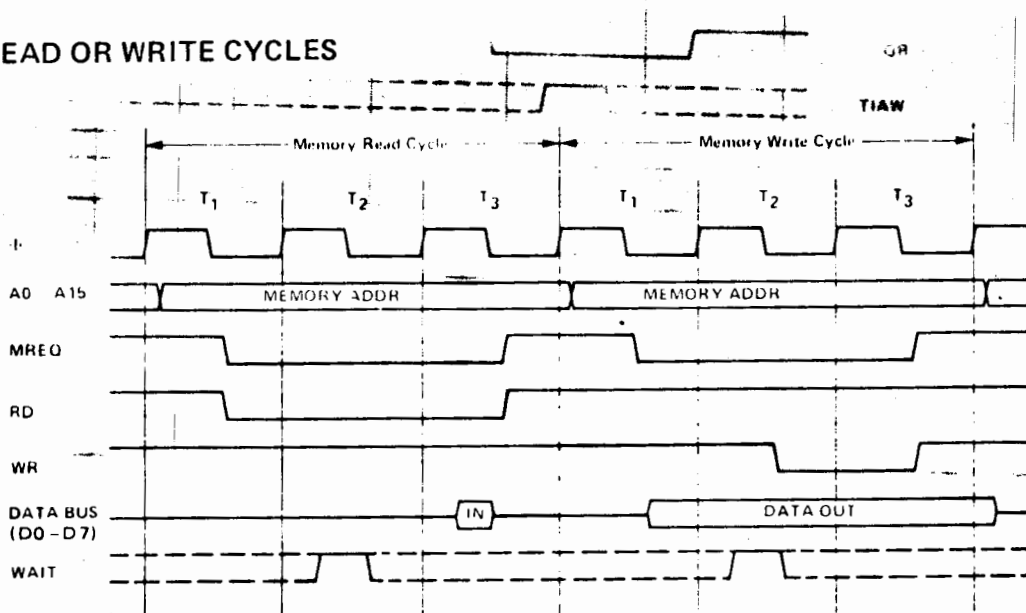


FIGURE 4.0-2

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

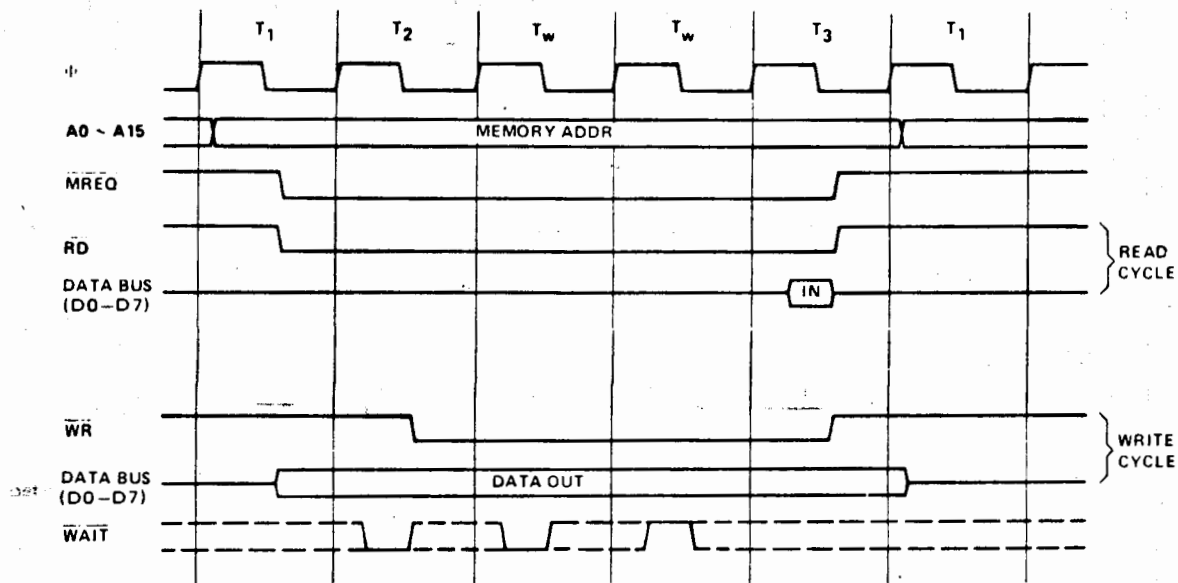


FIGURE 4.0-2A

### 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

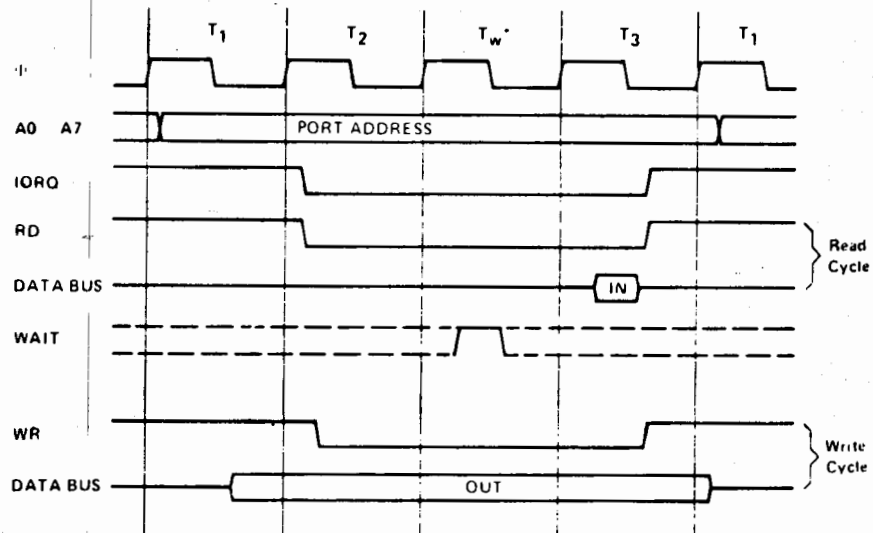


FIGURE 4.0-3

INPUT OR OUTPUT CYCLES WITH WAIT STATES

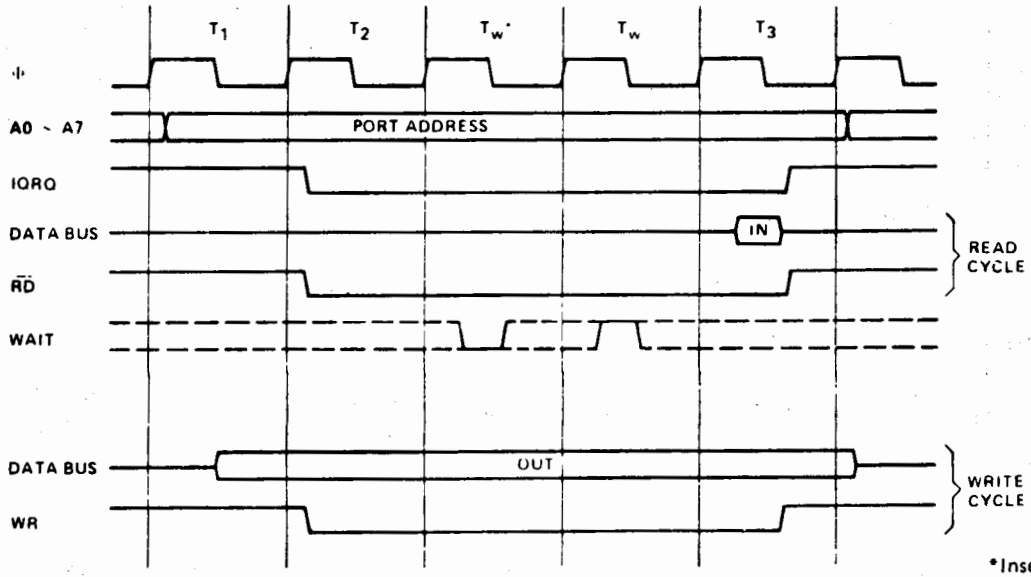


FIGURE 4.0-3A

**BUS REQUEST/ACKNOWLEDGE CYCLE**

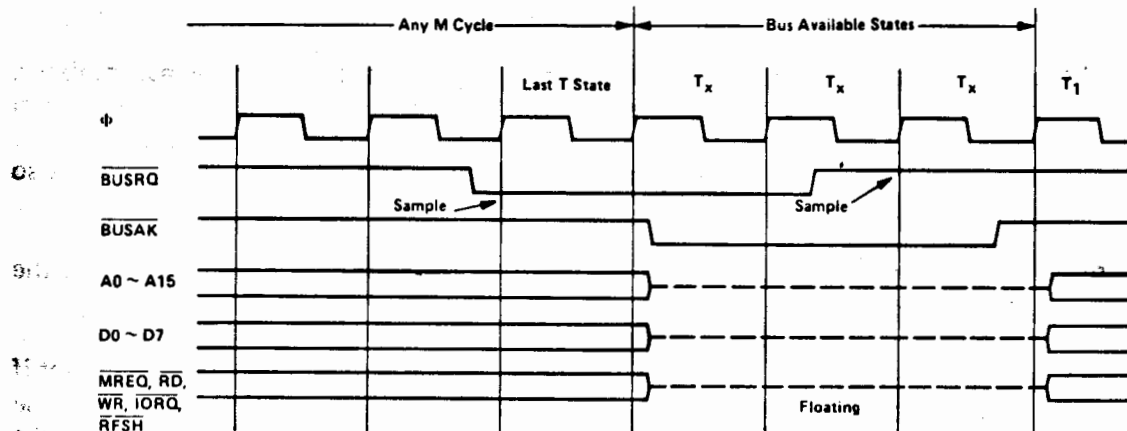
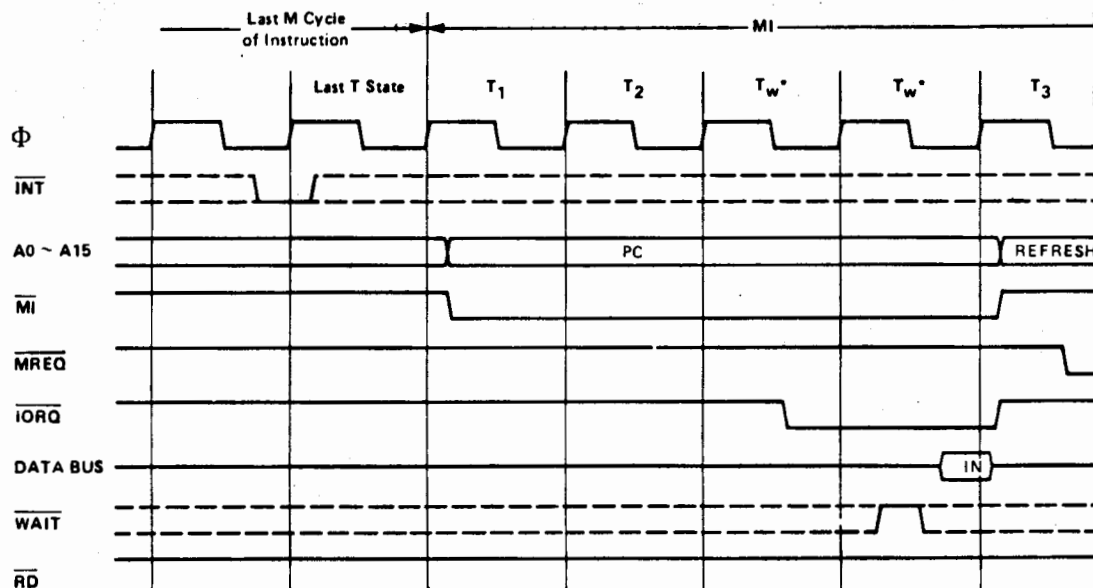


FIGURE 4.0-4

**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**



Mode 0 shown

FIGURE 4.0-5



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:

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

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:

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

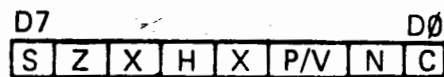
Notice that the answer is correct but the carry is set so that this flag can not be used as an overflow indicator. In this case the overflow would not be set.

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.

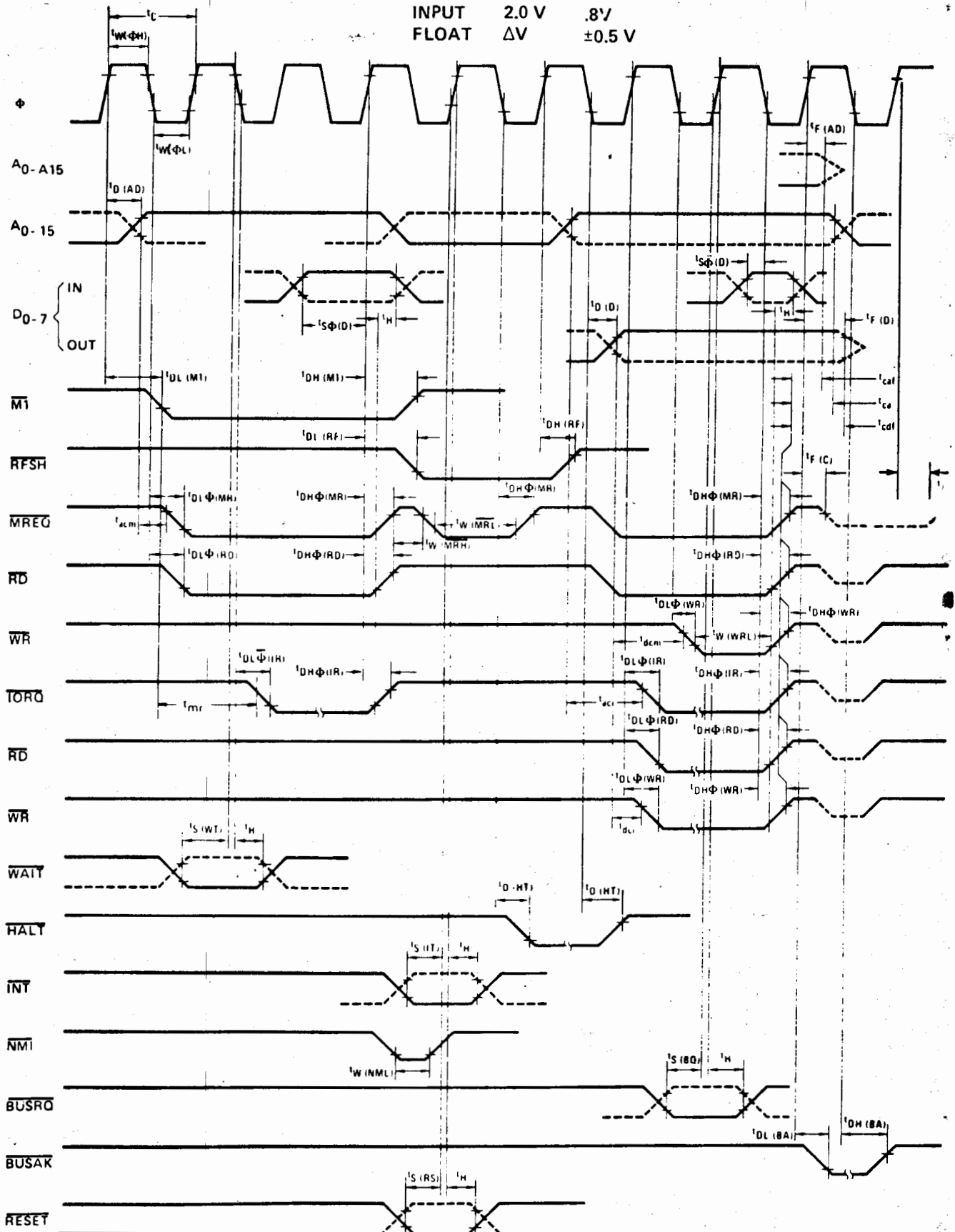
Table 6.0-1 lists how each flag bit is affected by various CPU instructions. In this table a '.' indicates that the instruction does not change the flag, an 'X' means that the flag goes to an indeterminate state, an '0' means that it is reset, a '1' means that it is set and the symbol  $\updownarrow$  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.

A.C. TIMING DIAGRAM

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	$V_{CC} - .6$	.8V
OUTPUT	2.0 V	.8V
INPUT	2.0 V	.8V
FLOAT	$\Delta V$	$\pm 0.5 V$



## 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	110		2	2	7	000 B 001 C
LD r, (HL)	r ← (HL)	•	•	X	•	X	•	•	•	01	r	110		1	2	7	010 D
LD r, (IX+d)	r ← (IX+d)	•	•	X	•	X	•	•	•	11	011	101	DD	3	5	19	011 E 100 H 101 L
LD r, (IY+d)	r ← (IY+d)	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19	111 A
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	DD	3	5	19	
LD (IY+d), r	(IY+d) ← r	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19	
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	DD	4	5	19	
LD (IY+d), n	(IY+d) ← n	•	•	X	•	X	•	•	•	11	111	101	FD	4	5	19	
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	
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	
LD A, I	A ← I	1	1	X	0	X	IFF	0	•	11	101	101	ED	2	2	9	
LD A, R	A ← R	1	1	X	0	X	IFF	0	•	11	101	101	ED	2	2	9	
LD I, A	I ← A	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	9	
LD R, A	R ← A	•	•	X	•	X	•	•	•	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,  
1 = flag is affected according to the result of the operation.

Table 7.0-1

## 16-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 dd, nn	dd ← nn	•	•	X	•	X	•	•	•	00	dd0	001		3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	11	011	101	DD	4	4	14	
										00	100	001	21				
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	11	111	101	FD	4	4	14	
										00	100	001	21				
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	X	•	X	•	•	•	00	101	010	2A	3	5	16	
LD dd, (nn)	dd <sub>H</sub> ← (nn+1) dd <sub>L</sub> ← (nn)	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20	
										01	dd1	011					
LD IX, (nn)	IX <sub>H</sub> ← (nn+1) IX <sub>L</sub> ← (nn)	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	20	
										00	101	010	2A				
LD IY, (nn)	IY <sub>H</sub> ← (nn+1) IY <sub>L</sub> ← (nn)	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	20	
										00	101	010	2A				
LD (nn), HL	(nn+1) ← H (nn) ← L	•	•	X	•	X	•	•	•	00	100	010	22	3	5	16	
LD (nn), dd	(nn+1) ← dd <sub>H</sub> (nn) ← dd <sub>L</sub>	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20	
										01	dd0	011					
LD (nn), IX	(nn+1) ← IX <sub>H</sub> (nn) ← IX <sub>L</sub>	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	20	
										00	100	010	22				
LD (nn), IY	(nn+1) ← IY <sub>H</sub> (nn) ← IY <sub>L</sub>	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	20	
										00	100	010	22				
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	
										11	111	001	F9				
LD SP, IY	SP ← IY	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	10	
										11	111	001	F9				
PUSH qq	(SP-2) ← qq <sub>L</sub> (SP-1) ← qq <sub>H</sub>	•	•	X	•	X	•	•	•	11	qq0	101		1	3	11	qq Pair 00 BC 01 DE 10 HL 11 AF
PUSH IX	(SP-2) ← IX <sub>L</sub> (SP-1) ← IX <sub>H</sub>	•	•	X	•	X	•	•	•	11	011	101	DD	2	4	15	
										11	100	101	E5				
PUSH IY	(SP-2) ← IY <sub>L</sub> (SP-1) ← IY <sub>H</sub>	•	•	X	•	X	•	•	•	11	111	101	FD	2	4	15	
										11	100	101	E5				
POP qq	qq <sub>H</sub> ← (SP+1) qq <sub>L</sub> ← (SP)	•	•	X	•	X	•	•	•	11	qq0	001		1	3	10	
POP IX	IX <sub>H</sub> ← (SP+1) IX <sub>L</sub> ← (SP)	•	•	X	•	X	•	•	•	11	011	101	DD	2	4	14	
										11	100	001	E1				
POP IY	IY <sub>H</sub> ← (SP+1) IY <sub>L</sub> ← (SP)	•	•	X	•	X	•	•	•	11	111	101	FD	2	4	14	
										11	100	001	E1				

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 M Cycles	No. of T States	Comments
		S	Z	H	P/V	N	C	76	543	210	Hex				
EX DE, HL	DE $\leftrightarrow$ HL	•	•	X	•	X	•	•	•	11 101 011	EB	1	1	4	Register bank and auxiliary register bank exchange
EX AF, AF'	AF $\leftrightarrow$ AF'	•	•	X	•	X	•	•	•	00 001 000	08	1	1	4	
EXX	(BC $\leftrightarrow$ BC') (DE $\leftrightarrow$ DE') (HL $\leftrightarrow$ HL')	•	•	X	•	X	•	•	•	11 011 001	D9	1	1	4	
EX (SP), HL	H $\leftrightarrow$ (SP+1) L $\leftrightarrow$ (SP)	•	•	X	•	X	•	•	•	11 100 011	E3	1	5	19	
EX (SP), IX	IX <sub>H</sub> $\leftrightarrow$ (SP+1) IX <sub>L</sub> $\leftrightarrow$ (SP)	•	•	X	•	X	•	•	•	11 011 101	DD	2	6	23	
EX (SP), IY	IY <sub>H</sub> $\leftrightarrow$ (SP+1) IY <sub>L</sub> $\leftrightarrow$ (SP)	•	•	X	•	X	•	•	•	11 111 101	FD	2	6	23	
LDI	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE+1 HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1	•	•	X	0	X	①	0	•	11 101 101	ED	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE+1 HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101	ED	2	5	21	If BC $\neq$ 0
										10 110 000	B0	2	4	16	If BC = 0
LDD	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE-1 HL $\leftarrow$ HL-1 BC $\leftarrow$ BC-1	•	•	X	0	X	①	0	•	11 101 101	ED	2	4	16	
										10 101 000	A8				
LDDR	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE-1 HL $\leftarrow$ HL-1 BC $\leftarrow$ BC-1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101	ED	2	5	21	If BC $\neq$ 0
										10 111 000	B8	2	4	16	If BC = 0
CPI	A $\leftarrow$ (HL) HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1	‡	‡	X	‡	X	①	1	•	11 101 101	ED	2	4	16	
										10 100 001	A1				
CPIR	A $\leftarrow$ (HL) HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1 Repeat until A = (HL) or BC = 0	‡	‡	X	‡	X	①	1	•	11 101 101	ED	2	5	21	If BC $\neq$ 0 and A $\neq$ (HL)
										10 110 001	B1	2	4	16	If BC = 0 or A = (HL)
CPD	A $\leftarrow$ (HL) HL $\leftarrow$ HL-1 BC $\leftarrow$ BC-1	‡	‡	X	‡	X	①	1	•	11 101 101	ED	2	4	16	
										10 101 001	A9				
CPDR	A $\leftarrow$ (HL) HL $\leftarrow$ HL-1 BC $\leftarrow$ BC-1 Repeat until A = (HL) or BC = 0	‡	‡	X	‡	X	①	1	•	11 101 101	ED	2	5	21	If BC $\neq$ 0 and A $\neq$ (HL)
										10 111 001	B9	2	4	16	If BC = 0 or A = (HL)

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 7.0-3

FAC. DE INGENIERIA

8-BIT ARITHMETIC AND LOGICAL 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						
ADD A, r	A ← A+r	†	†	X	†	X	V	0	†	10	000	r		1	1	4	r Reg.
ADD A, n	A ← A+n	†	†	X	†	X	V	0	†	11	000	110		2	2	7	000 B
												- n -					001 C
ADD A, (HL)	A ← A+(HL)	†	†	X	†	X	V	0	†	10	000	110		1	2	7	010 D
ADD A, (IX+d)	A ← A+(IX+d)	†	†	X	†	X	V	0	†	11	011	101	DD	3	5	19	011 E
												10 000 110					100 H
												- d -					101 L
ADD A, (IY+d)	A ← A+(IY+d)	†	†	X	†	X	V	0	†	11	111	101	FD	3	5	19	111 A
												10 000 110					
												- d -					
ADCA, s	A ← A+s+CY	†	†	X	†	X	V	0	†		001						s is any of r, n,
SUB s	A ← A-s	†	†	X	†	X	V	1	†		010						(HL), (IX+d),
SBC A, s	A ← A-s-CY	†	†	X	†	X	V	1	†		011						(IY+d) as shown for
AND s	A ← A ∧ s	†	†	X	1	X	P	0	0		100						ADD instruction.
DR s	A ← A ∨ s	†	†	X	0	X	P	0	0		110						The indicated bits
XOR s	A ← A ⊕ s	†	†	X	0	X	P	0	0		101						replace the 000 in
CP s	A ← s	†	†	X	†	X	V	1	†		111						the ADD set above.
INC r	r ← r+1	†	†	X	†	X	V	0	•	00	r	100		1	1	4	
INC (HL)	(HL) ← (HL)+1	†	†	X	†	X	V	0	•	00	110	100		1	3	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	†	†	X	†	X	V	0	•	11	011	101	DD	3	6	23	
												00 110 100					
												- d -					
INC (IY+d)	(IY+d) ← (IY+d)+1	†	†	X	†	X	V	0	•	11	111	101	FD	3	6	23	
												00 110 100					
												- d -					
DEC s	s ← s-1	†	†	X	†	X	V	1	•			101					s is any of r, (HL),
																	(IX+d), (IY+d) as
																	shown for INC.
																	DEC same format
																	and states as INC.
																	Replace 100 with
																	101 in DP 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: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.  
 † = flag is affected according to the result of the operation.

Table 7.0-4

## 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	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	1	X	•	1	•	00 101 111	2F	1	1	4	Complement accumulator (One's complement)
NEG	$A - \bar{A} + 1$	↓	↓	X	↓	X	V	1	↓	11 101 101 01 000 100	ED 44	2	2	8	Negate acc, (two's complement)
CCF	$CY - \bar{CY}$	•	•	X	X	X	•	0	↓	00 111 111	3F	1	1	4	Complement carry flag
SCF	$CY - 1$	•	•	X	0	X	•	0	1	00 110 111	37	1	1	4	Set carry flag
NOP	No operation	•	•	X	•	X	•	•	•	00 000 000	00	1	1	4	
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	FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	•	•	•	11 101 101 01 000 110	ED 46	2	2	8	
IM 1	Set interrupt mode 1	•	•	X	•	X	•	•	•	11 101 101 01 010 110	ED 56	2	2	8	
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11 101 101 01 011 110	ED 5E	2	2	8	

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-B1. ARITHMETIC GROUP

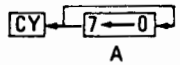
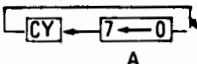
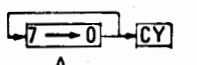
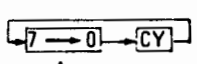
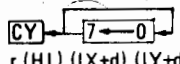
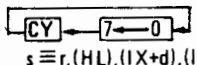
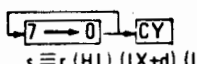
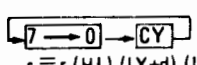
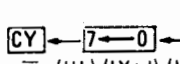
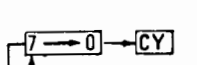
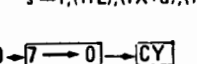
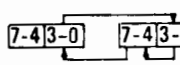
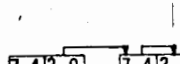
Mnemonic	Symbolic Operation	Flags							Op-Code				No. of Bytes	No. of M Cycles	No. of T States	Comments	
		S	Z		H	P/V	N	C	76	543	210	Hex					
ADD HL, ss	HL ← HL+ss	•	•	X	X	X	•	0	↓	00	ss1	001		1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL, ss	HL ← HL+ss+CY	↓	↓	X	X	X	V	0	↓	11	101	101	ED	2	4	15	01 DE 10 HL 11 SP
SBC HL, ss	HL ← HL-ss-CY	↓	↓	X	X	X	V	1	↓	11	101	101	ED	2	4	15	01 ss0 010
ADD IX, pp	IX ← IX+pp	•	•	X	X	X	•	0	↓	11	011	101	DD	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY+rr	•	•	X	X	X	•	0	↓	11	111	101	FD	2	4	15	rr Reg. 00 BC 01 DE 10 IY 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	DD	2	2	10	
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	10	
DEC ss	ss ← ss - 1	•	•	X	•	X	•	•	•	00	ss1	011		1	1	6	
DEC IX	IX ← IX - 1	•	•	X	•	X	•	•	•	11	011	101	DD	2	2	10	
DEC IY	IY ← IY - 1	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	10	
										00	101	011	2B				

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

ROTATE AND SHIFT 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	543210	Hex					
RLCA		•	•	X	0	X	•	0	†	00 000 111	07	1	1	4	Rotate left circular accumulator
RLA		•	•	X	0	X	•	0	†	00 010 111	17	1	1	4	Rotate left accumulator
RRCA		•	•	X	0	X	•	0	†	00 001 111	0F	1	1	4	Rotate right circular accumulator
RRA		•	•	X	0	X	•	0	†	00 011 111	1F	1	1	4	Rotate right accumulator
RLC r		†	†	X	0	X	P	0	†	11 001 011	CB	2	2	8	Rotate left circular register r
RLC (HL)		†	†	X	0	X	P	0	†	00 000 r 11 001 011	CB	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)		†	†	X	0	X	P	0	†	11 011 101 11 001 011 - d - 00 000 110	DD	4	6	23	
RLC (IY+d)		†	†	X	0	X	P	0	†	11 111 101 11 001 011 - d - 00 000 110	FD	4	6	23	
RL s		†	†	X	0	X	P	0	†	00 010					Instruction format and states are as shown for RLC's. To form new Op-Code replace 000 of RLC's with shown code
RRC s		†	†	X	0	X	P	0	†	001					
RR s		†	†	X	0	X	P	0	†	011					
SLA s		†	†	X	0	X	P	0	†	100					
SRA s		†	†	X	0	X	P	0	†	101					
SRL s		†	†	X	0	X	P	0	†	111					
RLD		†	†	X	0	X	P	0	•	11 101 101 01 101 111	ED 6F	2	5	18	
RRD		†	†	X	0	X	P	0	•	11 101 101 01 100 111	ED 67	2	5	18	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

## BIT SET, RESET AND TEST 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	r				Reg.		
BIT b, r	$Z - \bar{T}_b$	X	↓	X	1	X	X	0	•	11 001 011	CB	2	2	8	r	Reg.
										01 b r					000	B
BIT b, (HL)	$Z - \overline{(HL)}_b$	X	↓	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>	$Z - \overline{(IX+d)}_b$	X	↓	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
															b	Bit Tested
BIT b, (IY+d) <sub>b</sub>	$Z - \overline{(IY+d)}_b$	X	↓	X	1	X	X	0	•	11 111 101	FD	4	5	20	000	0
										11 001 011	CB				001	1
										- d -					010	2
										01 b 110					011	3
															100	4
															101	5
															110	6
															111	7
SET b, r	$r_b - 1$	•	•	X	•	X	•	•	•	11 001 011	CB	2	2	8		
										11 b r						
SET b, (HL)	$(HL)_b - 1$	•	•	X	•	X	•	•	•	11 001 011	CB	2	4	15		
										11 b 110						
SET b, (IX+d)	$(IX+d)_b - 1$	•	•	X	•	X	•	•	•	11 011 101	DD	4	6	23		
										11 001 011	CB					
										- d -						
										11 b 110						
SET b, (IY+d)	$(IY+d)_b - 1$	•	•	X	•	X	•	•	•	11 111 101	FD	4	6	23		
										11 001 011	CB					
										- d -						
										11 b 110						
RES b, s	$s_b - 0$ $s \equiv r, (HL), (IX+d),$	•	•	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  $s_b$  indicates bit b (0 to 7) or location s.

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.

## JUMP 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						
JP nn	PC - nn	•	•	X	•	X	•	•	•	11 000 011	C3	3	3	10			
										- n -							
										- n -							
JP cc, nn	If condition cc is true PC - nn, otherwise continue	•	•	X	•	X	•	•	•	11 cc 010		3	3	10	cc    Condition 000    NZ non zero 001    Z zero 010    NC non carry 011    C carry 100    PO parity odd 101    PE parity even 110    P sign positive 111    M sign negative		
										- n -							
										- n -							
										11 00 00							
JR e	PC - PC + e	•	•	X	•	X	•	•	•	00 011 000	18	2	3	12			
										- e-2 -							
JR C, e	If C = 0, continue If C = 1, PC - PC + e	•	•	X	•	X	•	•	•	00 111 000	38	2	2	7	If condition not met		
										- e-2 -		2	3	12	If condition is met		
JR NC, e	If C = 1, continue If C = 0, PC - PC + e	•	•	X	•	X	•	•	•	00 110 000	30	2	2	7	If condition not met		
										- e-2 -		2	3	12	If condition is met		
JR Z, e	If Z = 0, continue If Z = 1, PC - PC + e	•	•	X	•	X	•	•	•	00 101 000	28	2	2	7	If condition not met		
										- e-2 -		2	3	12	If condition is met		
JR NZ, e	If Z = 1, continue If Z = 0, PC - PC + e	•	•	X	•	X	•	•	•	00 100 000	20	2	2	7	If condition not met		
										- e-2 -		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			
										11 101 001	E9						
JP (IY)	PC - IY	•	•	X	•	X	•	•	•	11 111 101	FD	2	2	8			
										11 101 001	E9						
DJNZ, e	B - B-1 If B = 0, continue	•	•	X	•	X	•	•	•	00 010 000	10	2	2	8	If B = 0		
										- e-2 -							
	If B ≠ 0, PC - PC + e											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.

## CALL AND RETURN GROUP

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							
CALL nn	(SP-1) - PC <sub>H</sub> (SP-2) - PC <sub>L</sub> 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	C9	3	3	10	If cc is false		
										-	n	-		3	5	17	If cc is true		
RET	PC <sub>L</sub> - (SP) PC <sub>H</sub> - (SP+1)	•	•	X	•	X	•	•	•	11	001	001	C9	1	3	10			
RET cc	If condition cc is false continue, otherwise same as RET	•	•	X	•	X	•	•	•	11	cc	000	C9	1	1	5	If cc is false		
										-	n	-		1	3	11	If cc is true		
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11	101	101	ED	2	4	14	011	C	carry
										01	001	101					4D	100	PO
RETN <sup>1</sup>	Return from non maskable interrupt	•	•	X	•	X	•	•	•	11	101	101	ED	2	4	14	101	PE	parity even
										01	000	101					45	110	P
RST p	(SP-1) - PC <sub>H</sub> (SP-2) - PC <sub>L</sub> PC <sub>H</sub> - 0 PC <sub>L</sub> - p	•	•	X	•	X	•	•	•	11	t	111	C9	1	3	11	111	M	sign negative
										-	n	-							

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

<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.

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				
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11 011 011	DB	2	3	11	n to A <sub>0</sub> ~ A <sub>7</sub> Acc to A <sub>8</sub> ~ A <sub>15</sub>
IN r, (C)	r ← (C) if r = 110 only the flags will be affected	†	†	X	†	X	P	0	•	11 101 101 01 r 000	ED	2	3	12	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
INI	(HL) ← (C) B ← B - 1 HL ← HL + 1	X	†	X	X	X	X	1	•	11 101 101 10 100 010	ED A2	2	4	16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
INIR	(HL) ← (C) B ← B - 1 HL ← HL + 1 Repeat until B = 0	X	1	X	X	X	X	1	•	11 101 101 10 110 010	ED B2	2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
IND	(HL) ← (C) B ← B - 1 HL ← HL - 1	X	†	X	X	X	X	1	•	11 101 101 10 101 010	ED AA	2	4	16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
INDR	(HL) ← (C) B ← B - 1 HL ← HL - 1 Repeat until B = 0	X	1	X	X	X	X	1	•	11 101 101 10 111 010	ED BA	2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A <sub>0</sub> ~ A <sub>7</sub> 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> Acc to A <sub>8</sub> ~ A <sub>15</sub>
OUT (C), r	(C) → r	•	•	X	•	X	•	•	•	11 101 101 01 r 001	ED	2	3	12	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
OUTI	(C) ← (HL) B ← B - 1 HL ← HL + 1	X	†	X	X	X	X	1	•	11 101 101 10 100 011	ED A3	2	4	16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
OTIR	(C) ← (HL) B ← B - 1 HL ← HL + 1 Repeat until B = 0	X	1	X	X	X	X	1	•	11 101 101 10 110 011	ED B3	2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
OUTD	(C) ← (HL) B ← B - 1 HL ← HL - 1	X	†	X	X	X	X	1	•	11 101 101 10 101 011	ED AB	2	4	16	C to A <sub>0</sub> ~ A <sub>7</sub> B to A <sub>8</sub> ~ A <sub>15</sub>
OTDR	(C) ← (HL) B ← B - 1 HL ← HL - 1 Repeat until B = 0	X	1	X	X	X	X	1	•	11 101 101 10 111 011	ED BB	2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A <sub>0</sub> ~ A <sub>7</sub> 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

## 8.0 INTERRUPT RESPONSE

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 instruction 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.

Figure 8.0-1 is a summary of the effect of different instructions on the two enable flip flops.

## 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.

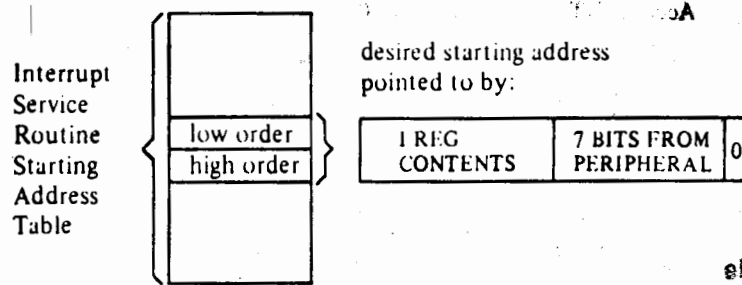
#### Mode 1

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.



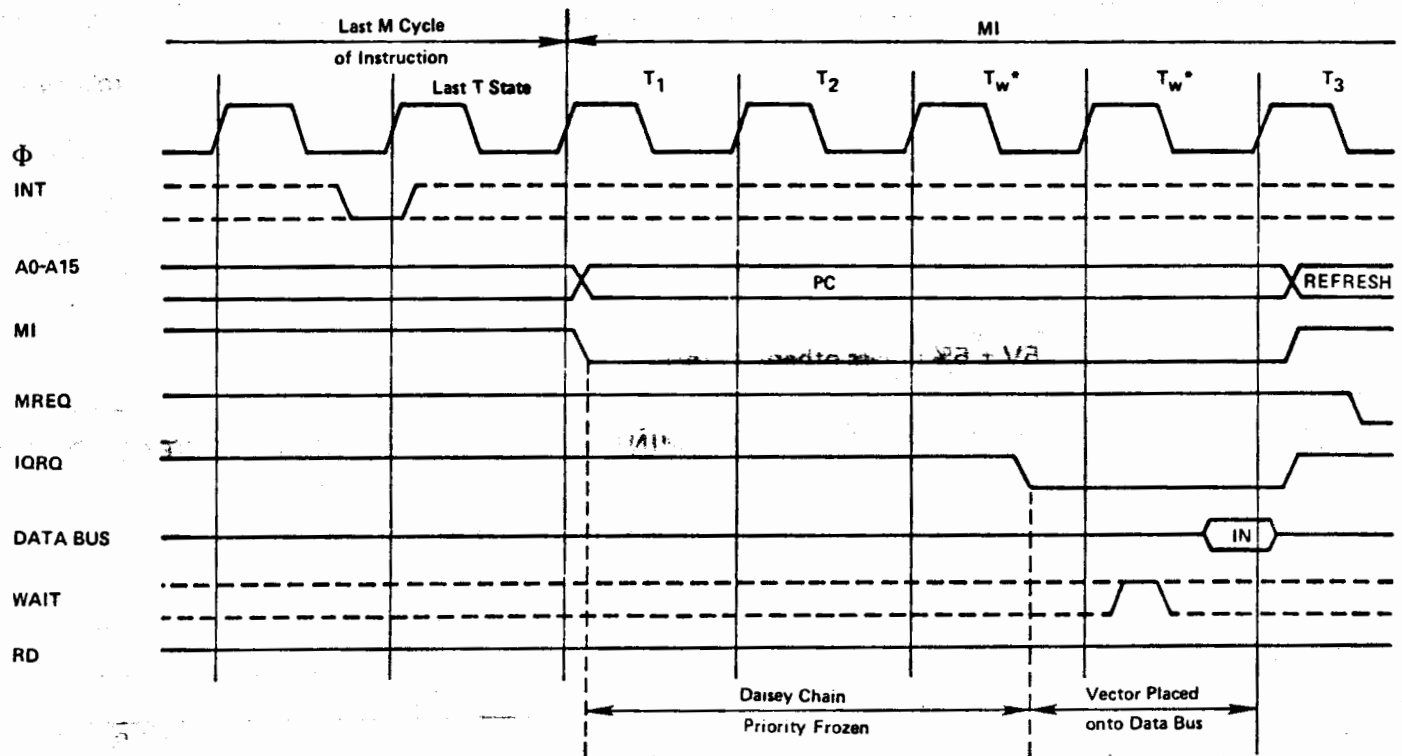
The first byte in the table is the least significant (low order) portion of the address. The programmer must obviously fill this table in with the desired addresses before any interrupts are to be accepted.

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 <sub>ILC</sub>	Clock Input Low Voltage	-0.3		0.8	V	
V <sub>IHC</sub>	Clock Input High Voltage	V <sub>CC</sub> -0.6		V <sub>CC</sub> +0.3	V	
V <sub>IL</sub>	Input Low Voltage	-0.3		0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0		V <sub>CC</sub>	V	
V <sub>OL</sub>	Output Low Voltage			0.4	V	I <sub>OL</sub> = 1.8mA
V <sub>OH</sub>	Output High Voltage	2.4			V	I <sub>OH</sub> = -250 μA
I <sub>CC</sub>	Power Supply Current			150*	mA	
I <sub>LI</sub>	Input Leakage Current			10	μA	V <sub>IN</sub> = 0 to V <sub>CC</sub>
I <sub>LOH</sub>	Tri-State Output Leakage Current in Float			10	μA	V <sub>OUT</sub> = 2.4 to V <sub>CC</sub>
I <sub>LOL</sub>	Tri-State Output Leakage Current in Float			-10	μA	V <sub>OUT</sub> = 0.4V
I <sub>LD</sub>	Data Bus Leakage Current in Input Mode			±10	μA	0 ≤ V <sub>IN</sub> ≤ V <sub>CC</sub>

\*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 <sub>Φ</sub>	Clock Capacitance	35	pF
C <sub>IN</sub>	Input Capacitance	5	pF
C <sub>OUT</sub>	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.

## A C CHARACTERISTICS

 $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = +5\text{V} \pm 5\%$ , Unless Otherwise Noted

SIGNAL	SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITION
$\Phi$	$t_c$	Clock Period	.4	[12]	$\mu\text{sec}$	
	$t_w(\Phi H)$	Clock Pulse Width, Clock High	180	(D)	nsec	
	$t_w(\Phi L)$	Clock Pulse Width, Clock Low	180	2000	nsec	
	$t_{r,f}$	Clock Rise and Fall Time		30	nsec	
A0-15	$t_D(AD)$	Address Output Delay		145	nsec	$C_L = 50\text{pF}$
	$t_F(AD)$	Delay to Float	[1]	110	nsec	
	$t_{acm}$	Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)			nsec	Except T3-M1
	$t_{aci}$	Address Stable Prior to $\overline{IORQ}$ , $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)	[2]		nsec	
D0-7	$t_D(D)$	Data Output Delay		230	nsec	$C_L = 50\text{pF}$
	$t_F(D)$	Delay to Float During Write Cycle		90	nsec	
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Clock During M1 Cycle	50		nsec	
	$t_{S\overline{\Phi}(D)}$	Data Setup Time to Falling Edge at Clock During M2 to M5	60		nsec	
	$t_{dcm}$	Data Stable Prior to $\overline{WR}$ (Memory Cycle)	[5]		nsec	
	$t_{dci}$	Data Stable Prior to $\overline{WR}$ (I/O Cycle)	[6]		nsec	
$\overline{MREQ}$	$t_{DL\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ Low		100	nsec	$C_L = 50\text{pF}$
	$t_{DH\Phi(MR)}$	$\overline{MREQ}$ Delay From Rising Edge of Clock, $\overline{MREQ}$ High		100	nsec	
	$t_{DH\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ High		100	nsec	
	$t_w(\overline{MRL})$	Pulse Width, $\overline{MREQ}$ Low	[8]		nsec	
	$t_w(\overline{MRH})$	Pulse Width, $\overline{MREQ}$ High	[9]		nsec	
$\overline{IORQ}$	$t_{DL\Phi(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ Low		90	nsec	$C_L = 50\text{pF}$
	$t_{DL\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ Low		110	nsec	
	$t_{DH\Phi(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ High		100	nsec	
	$t_{DH\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ High		110	nsec	
$\overline{RD}$	$t_{DL\Phi(RD)}$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ Low		100	nsec	$C_L = 50\text{pF}$
	$t_{DL\overline{\Phi}(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ Low		130	nsec	
	$t_{DH\Phi(RD)}$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ High		100	nsec	
	$t_{DH\overline{\Phi}(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ High		110	nsec	
$\overline{WR}$	$t_{DL\Phi(WR)}$	$\overline{WR}$ Delay From Rising Edge of Clock, $\overline{WR}$ Low		80	nsec	$C_L = 50\text{pF}$
	$t_{DL\overline{\Phi}(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ Low		90	nsec	
	$t_{DH\Phi(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ High		100	nsec	
	$t_w(\overline{WRL})$	Pulse Width, $\overline{WR}$ Low	[10]		nsec	

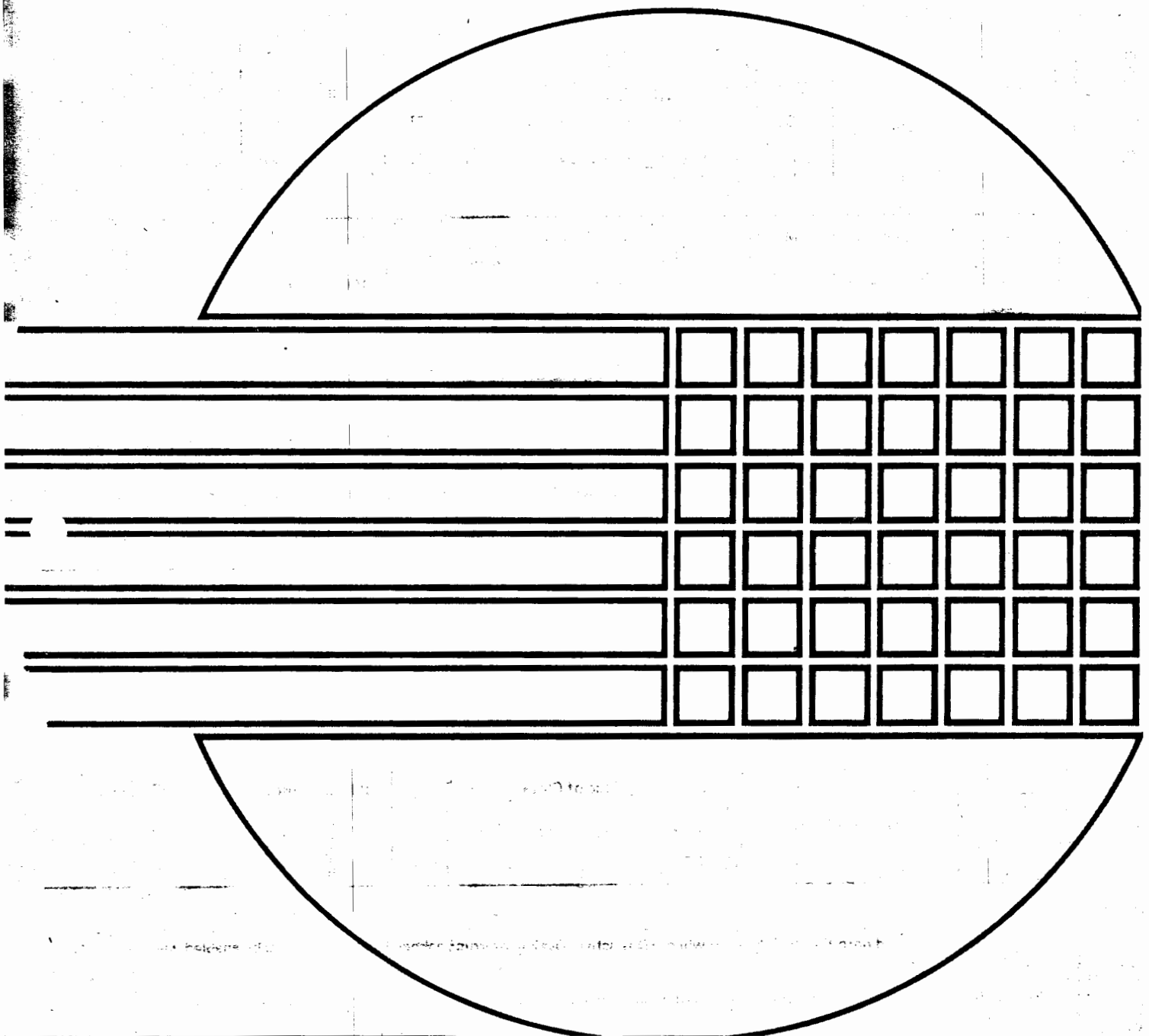
## NOTES:

A Data should be enabled onto the CPU data bus when  $\overline{RD}$  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 16,384-BIT ERASABLE AND REPROGRAMMABLE MOS ROM (2048X8) 2716



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

**2716**

PRELIMINARY SPECIFICATION

2716-1

**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.

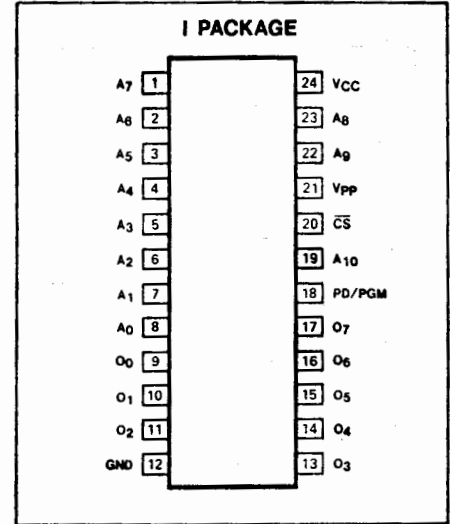
The 2716 operates from a single 5 volt power supply which makes it ideal for use with the newer high performance 5 volt microprocessors. A power down mode reduces power during standby to 25% that of operating power.

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.

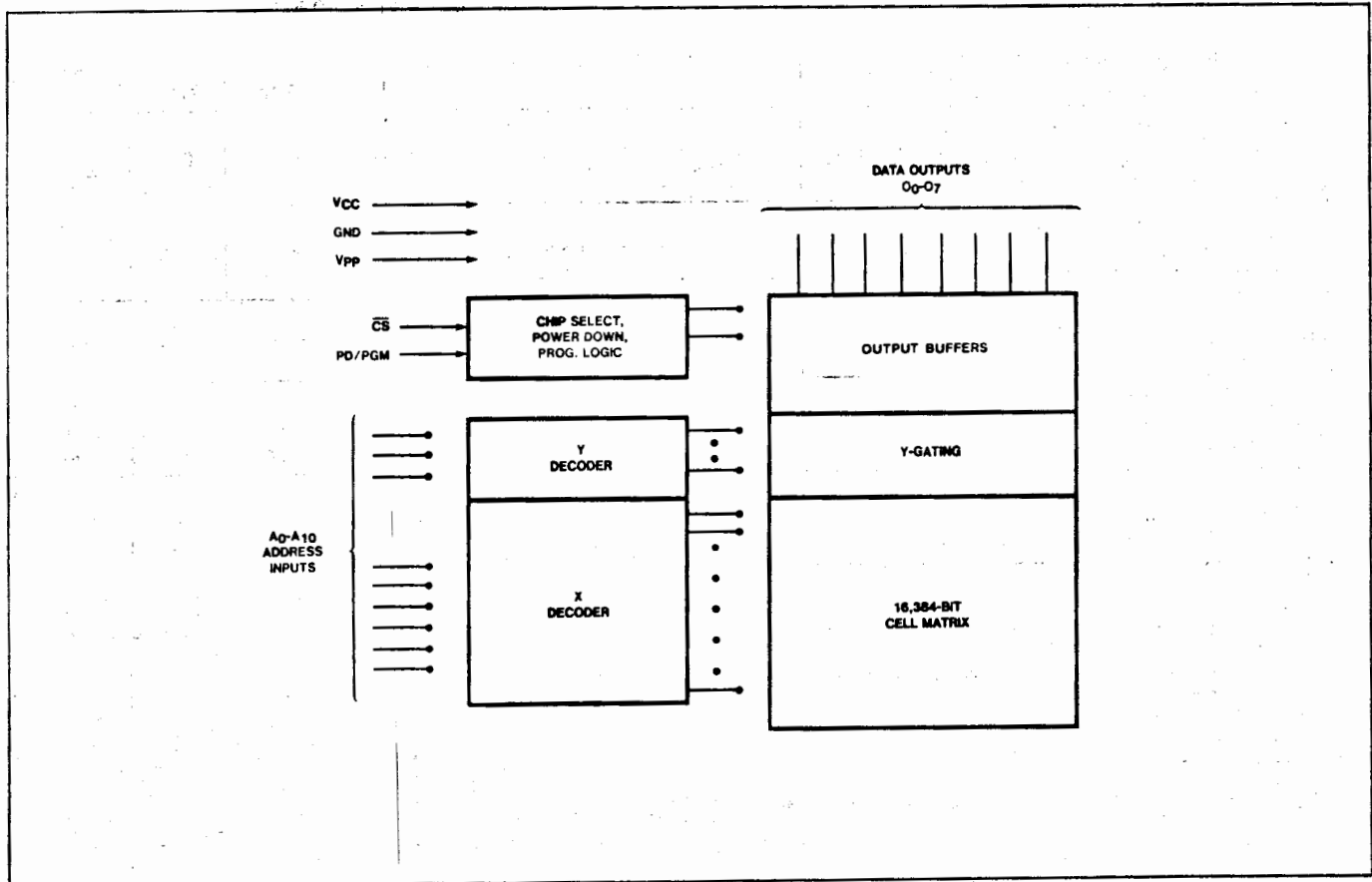
**FEATURES**

- 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 <sub>A</sub>	Temperature range	C°	
	Operating		-10 to 80
T <sub>STG</sub>	Storage	-65 to 125	
	All input or output voltages with respect to ground	-0.3 to 6	V
	V <sub>pp</sub> supply voltage with respect to ground	-0.3 to 28	V

## NOTE

1. Stresses above these 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 <sub>0</sub> -A <sub>10</sub>	Address inputs
18	PD/PGM	Power down/Program
20	$\overline{CS}$	Chip select
9-11, 13-17	O <sub>0</sub> -O <sub>7</sub>	Outputs
24	V <sub>CC</sub>	Power (+5V)
21	V <sub>pp</sub>	Program voltage (+25V)
12	GND	Ground

DC ELECTRICAL CHARACTERISTICS T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub><sup>2</sup> = +5V, ±5%, V<sub>pp</sub><sup>3</sup> = V<sub>CC</sub> ±0.6V<sup>4</sup>, unless otherwise specified.

PARAMETER	TEST CONDITIONS	LIMITS			UNIT
		Min	Typ <sup>5</sup>	Max	
V <sub>IL</sub>	Input voltage Low	-0.1		0.8	V
V <sub>IH</sub>	Input voltage High	2.2		V <sub>CC</sub> +1	V
V <sub>OL</sub>	Output voltage Low			0.45	V
V <sub>OH</sub>	Output voltage High			2.4	V
I <sub>LI</sub>	Input load current			10	μA
I <sub>LO</sub>	Output leakage current			10	μA
I <sub>pp</sub> <sup>3</sup>	V <sub>pp</sub> current			5	mA
I <sub>CC1</sub> <sup>3</sup>	V <sub>CC</sub> current (standby)		10	25	mA
I <sub>CC</sub> <sup>3</sup>	V <sub>CC</sub> current (active)		57	100	mA
C <sub>IN</sub>	Capacitance <sup>6</sup> Input			4	pF
C <sub>OUT</sub>	Capacitance <sup>6</sup> Output			6	pF

## NOTES

- V<sub>CC</sub> must be applied simultaneously or before V<sub>pp</sub> and removed simultaneously or after V<sub>pp</sub>.
- V<sub>pp</sub> may be connected directly to V<sub>CC</sub> except during programming. The V<sub>CC</sub> supply current would then be the sum of I<sub>CC</sub> and I<sub>pp</sub>.
- The tolerance of 0.6V allows the use of a driver circuit for switching the V<sub>pp</sub> supply pin from V<sub>CC</sub> in read to 25V for programming.
- Typical values are for T<sub>A</sub> = 25°C and nominal supply voltages.
- This parameter is only sampled and is not 100% tested.
- t<sub>ACC2</sub> is referenced to PD/PGM or the addresses, whichever occurs last.

**Table 4**  
**Timing: Data from Peripheral to Microprocessor**

$T_A = 25^\circ\text{C}$

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	COMM
$t_{ACSO}$	Address Bus Valid to Chip Select ON (CS=0)		40		ns	VDD=5V
$t_{CSR}$	Chip Select ON to Read Strobe	70			ns	
$t_{RD}$	Read Cycle Access Time from Read Strobe to Data Bus Valid		450	600	ns	CL=100pF
$t_{RH}$	Data hold time from trailing edge of Read Strobe	0		250	ns	
$t_{RA}$	Address Bus hold time from trailing edge of Read Strobe	50	500		ns	
$t_{ACSI}$	Address change to Chip Select OFF		40		ns	
$t_{AD}$	Address Bus valid to Data valid		850	1250	ns	CL=100pF
$t_{HZ}$	Time from trailing edge of Read Strobe until interface device bus drivers are in Tri-State mode	0		250	ns	
$t_{RW}$	Read Strobe Width			14	ns	

**Table 5**  
**Timing: Data from Microprocessor to Peripheral**

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	COMM
$t_{ACSO}$	Address Bus valid to Chip Select ON (CS=0)		40		ns	VDD=5V
$t_{CSW}$	Chip Select ON to Write Strobe	310	450		ns	
$t_{AW}$	Address Bus valid to Write Strobe	350			ns	
$t_{WW}$	Write Strobe Width	430			ns	
$t_{DW}$	Data Bus valid before Write Strobe	50			ns	
$t_{WA}$	Address Bus hold time following Write Strobe	100			ns	
$t_{WD}$	Data Bus hold time following Write Strobe	50			ns	
$t_{ACSI}$	Address change to Chip Select OFF (CS=1)		40		ns	

National Semiconductor Corporation  
2900 Semiconductor Drive, Santa Clara, California 95051 (408) 737-5000/TWX (910) 339-9240

National Semiconductor GmbH  
8080 Fuerstenfeldbruck Industriestrasse 10 West Germany, Tele 08141) 1371, 1373, 1374, 1484/Telex 527-649

National Semiconductor (UK) Ltd.  
Larkfield Industrial Estate, Greenock, Scotland, Telephone 0475-33251/Telex 778632

