



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**Facultad de Ingeniería**

**DISEÑO Y CONSTRUCCION DE UNA INTERFASE  
PARA UN MICROPROCESADOR.**

**T E S I S**

Que para obtener el título de:  
**INGENIERO MECANICO ELECTRICISTA**

**P r e s e n t a n :**

**JOSE LUIS MORENO LASTRA**

**EMILIO RAMIREZ RAMIREZ**

**México, D. F.**

**1979**

24  
101



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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

# TESIS CON FALLA DE ORIGEN

# INDICE

<i>Tema.</i>	<i>Página</i>
<b>- Introducción</b>	<b>1</b>
<b>Capítulo I.</b>	
<b>- Descripción del sistema.</b>	<b>5</b>
<b>Objetivos de la interfase.</b>	<b>7</b>
<b>Elección del microprocesador.</b>	<b>7</b>
<b>Selección de la calculadora electrónica.</b>	<b>8</b>
<b>Simulación del teclado.</b>	<b>10</b>
<b>Densificación y demultiplexaje del display.</b>	<b>10</b>
<b>Desplazamiento de los niveles de tensión.</b>	<b>11</b>
<b>Capítulo II.</b>	
<b>- Interfase del display.</b>	<b>12</b>
<b>Desplazamiento de los niveles de tensión.</b>	<b>12</b>
<b>Conversión de 7 segmentos a código B C D.</b>	<b>16</b>
<b>Discrecionalización de los dígitos.</b>	<b>25</b>
<b>Capítulo III.</b>	
<b>- Interfase del teclado.</b>	<b>28</b>
<b>Estructura del teclado</b>	<b>28</b>
<b>Simulación del teclado.</b>	<b>31</b>
<b>Circuito de interfase del teclado.</b>	<b>32</b>

#### Capítulo IV.

- Circuito de interfase.	36
Etapas de escritura.	36
Etapas de lectura.	37
Implementación física del circuito de interfase	40

#### Capítulo V.

Interfases periféricas programable.	44
Selección del modo de operación.	47
Modos de operación.	50
Definición de los formatos de entrada y salida.	57
Código de errores.	59

#### Capítulo VI.

- Descripción del programa.	61
Instrucciones del programa.	61
Ejemplo y simulación.	67

#### Apéndice II.

- Unidad central de procesamiento 8080.	90
Arquitectura del C T U - 8080.	90
El ciclo del procesador.	94
Secuencia de interrupción.	103
Secuencia de rotación.	105
Secuencia de alto.	107
Instrucciones del C P U - 8080.	109

*Índice B.*

*- La calculadora electrónica de bolsillo.*

*113*

*- Bibliografía.*

*129*

----- 0 -----

## INTRODUCCION

A finales de 1971 se comercializó el primer Microprocesador (el modelo 4004 - de la casa INTEL). En un principio solo fue utilizado por científicos, investigadores y técnicos de grandes empresas; actualmente ha alcanzado tal desarrollo que se está imponiendo con inusitada fuerza, a tal grado, que se cree que el Microprocesador está destinado a convulsionar el mundo de la electrónica.

En efecto los Microprocesadores están abriendo nuevos caminos facilitando el diseño de sistemas complejos ofreciendo una gran flexibilidad y posibilitando la introducción de nuevas opciones en los sistemas ya construidos.

Entre las muchas ventajas que proporcionan estos dispositivos cuenta la reducción de costos en innumerables aplicaciones en las dorsales de circuitos integrados - que pueden ser reemplazados por uno o varios chips. La reducción del cableado y la minimización del circuito supone una mayor fiabilidad, menor consumo de potencia y por último una gran facilidad para el diagnóstico de averías, reparación y mantenimiento de los equipos.

Los Microprocesadores tienen una notable repercusión en la concepción del diseño, en el tiempo y costo de desarrollo de nuevos sistemas, y consecuentemente de sus costos de fabricación.

Los dispositivos del Microprocesamiento están siendo utilizados en numerosas y variadas aplicaciones, cubriendo un amplio espectro que abarca desde las comunicaciones, la industria, los productos de consumo, la instrumentación y las computaciones. Dentro de esta gran gama de aplicaciones se pueden citar sistemas tan variados como el control de procesos, control robótico, terminales inteligentes, sistemas de ventas,

juegos electrónicos, elementos autocalibradores, equipos biomédicos, aplicaciones militares, control de vehículos, procesamiento de datos, aparatos electrodomésticos, etc.

En la mayoría de estas aplicaciones se requerirá desarrollar operaciones automáticas en mayor o menor complejidad. Estas operaciones las puede realizar el Microprocesador, así como una computadora electrónica, la diferencia estriba en la forma de proporcionar el programa a la máquina.

En el Microprocesador el programa se introduce en la memoria, y por consiguiente, es modificado o eliminado a voluntad sin tener que intervenir en la circuitería del mismo; es lo que se llama una máquina con programa grabado.

En una computadora, por el contrario, el programa viene fijado por la circuitería en forma de cableado (Hardware). Por tanto, no se puede modificar a posteriori. El fenómeno es diferente si el programa (o el microprograma) se graba en una memoria ROM o PROM.

Para que un Microprocesador realice todas las operaciones que pueden obtenerse de una computadora electrónica necesitamos suministrarle un conjunto de rutinas Software (programas) dichas rutinas serán complejas y tediosas y necesitaremos gran cantidad de memoria.

Otro método de obtener esto es aceptar una computadora o un Microprocesador, — necesitando además un pequeño programa, para que el microprocesador controle la computadora, y así, realice estas operaciones automáticas.

En este trabajo se presenta el diseño de una interfase implementada en Hardware que realiza la función de acoplamiento entre una computadora electrónica y un Micro



procesador. Esta Interfase evita la implementación en Software de las funciones aritméticas esenciales anteriormente relativamente a bajo costo.

El sistema "Microprocesador - Calculadora" tiene aplicación principalmente en aquellos proyectos en los cuales no son necesario una gran precisión ni velocidad de cálculo, debido a que los calculadores tienden a ser lentos; algunos calculadores apenas realizan diez multiplicaciones por segundo. El utilizar un calculador es enfaticada tal vez disminuya la lentitud del sistema, pero aumentaría el costo, volumen y complejidad de la Interfase a desarrollar. En cualquier caso el principio del diseño de la Interfase es el mismo, quedando a criterio del diseñador el utilizar un calculadora más o menos compleja dependiendo de los requerimientos de su sistema a implementar.

Para fines ilustrativos en el desarrollo de la Interfase se acoplamiento se ha empleado un calculadora que únicamente realiza las cuatro operaciones fundamentales (Suma, Resta, Multiplicación y División).

Este trabajo que se presenta como aporte del Seminario titulado "Diseño y Construcción de una Interfase para un Microprocesador", se encuentra dividido en seis capítulos y dos apéndices.

En el capítulo I se establecen las relaciones necesarias para acoplar un Microprocesador y un calculadora, así como la selección del Microprocesador y la calculadora utilizada. También se establecen los objetivos en el diseño de la Interfase.

El capítulo II se refiere al diseño de la Interfase respecto a la parte que corresponde al Display. Se diseña el circuito que convierte el código de siete segmentos a código BCD.

En el capítulo III se investiga acerca de como está constituido el teclado de la calculadora y se diseña el circuito que logra la simulación del mismo.

En el capítulo IV se describe el circuito total de Interfases, así como, las señales de control necesarias para establecer comunicación entre la calculadora y el Microprocesador.

Las características generales de la Interfase Paralela Programable (PP1 - 8255) son enunciadas en el capítulo V, así como la selección de su modo de operación; también es establecido el formato de Entrada / Salida y el código del teclado.

La descripción del programa que permite al C. F. U. controlar la calculadora, así como, el ensamble y simulación del mismo es realizada en el capítulo VI.

En el Apéndice A se da un resumen de las características principales del Microprocesador 8080 de INTEL, así como el grupo de instrucciones del mismo.

Por último, en el Apéndice B se trata de dar una breve descripción del funcionamiento básico de una calculadora electrónica.

## CAPITULO I

## DESCRIPCION DEL SISTEMA

Para llevar a cabo nuestro objetivo: establecer comunicación entre un Microprocesador y una calculadora electrónica requerimos hacerlo a través de dos interfaces como se muestra en la figura (1.1), la cual representa el micro sistema que utilizamos.

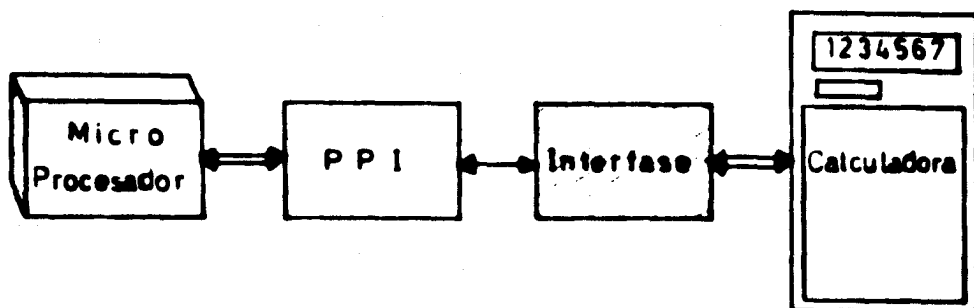


Fig.(1.1).- Sistema CPU - Calculadora.

El P. P. I. es una Interfase Paralela Programable que depende totalmente del Microprocesador, y puesto que este tipo de Interfase es ofrecido como un paquete estándar en casi todos los microprocesadores en diseño no será incluido en este trabajo. El PPI es un chip especializado en el control de periféricos, tiene como característica fundamental su programabilidad por Software y su conexión directa a los buses del Microprocesador sin chips auxiliares.

Cada uno de estos chips (PPI) tienen una serie de entradas y salidas que por -  
propósito pueden estar juntas o separadas como entradas, como salidas, en alta impo-

dancia, etc. al mismo tiempo pueden direccionarse y obtenerse desde el Microprocesador las líneas de Entrada y Salida, bit a bit.

De esta forma, según alguno de los diferentes modos de funcionamiento pueden actuar como salidas, como entradas, como buses bidireccionales, como señales de control "handshaking". Cuando actúan como salidas generan suficiente corriente para impulsar transistores darlington de potencia para impresoras o displays de alta tensión, etc.

Puesto que la Calculadora es un periférico para el Microprocesador el "enlace" entre estos será a través del PPI.

Para hacer posible la transferencia de datos entre el PPI y la Calculadora necesitamos de una INTERFASE, a la cual, llamaremos "Interfase de la Calculadora" --- (debido a que es función de la arquitectura de la Calculadora) y será desarrollado en el presente trabajo.

La rutina Software para el manejo de datos transferidos de la Interfase hacia el CPU, dependará del Microprocesador y del PPI. Es decir, se requiere desarrollar un programa, el cual, almacenado en PROM nos permitirá dirigir al CPU, inicializar el PPI en el modo de trabajo deseado, para de esta manera lograr la sincronización adecuada y establecer la comunicación deseada.

En el capítulo VI trataremos en forma detallada al programa mencionado, sistema operativo aquí que esta rutina es bastante más simple que las rutinas anteriores - que están reemplazadas por la calculadora. Esta rutina es básicamente del tipo Load y Store (Carga y Almacenamiento).

## 1.1.- OBJETIVOS DE LA INTERFASE.

El diseño de una INTERFASE entre un Microprocesador y una Calculadora Electrónica implica el desarrollo de un cierto procedimiento, y que para nuestros fines se asumimos de la siguiente manera:

- A).- Selección del Microprocesador.
- B).- Selección de la Calculadora Electrónica.
- C).- Simulación del teclado de la calculadora.
- D).- Demultiplexaje y decodificación del display de la calculadora.
- E).- Desplazamiento a niveles de voltajes adecuados.

### 1.1.1) Selección del Microprocesador:

Para la realización de este Seminario utilizamos el Microprocesador 8080 de INTEL disponible en el Laboratorio de Electrónica de la Facultad de Ingeniería.

El Microprocesador 8080 de Intel está implementado en un solo chip de LSI, el cual, consta de 40 terminales en doble línea. Este Microprocesador es fabricado utilizando tecnología de compuertas de silicio tipo MOS de canal N, de esta manera se ha obtenido este rápido (2 microsseg/ciclo) y poderoso (72 instrucciones básicas) -- MICROPROCESADOR 8080.

El Microprocesador 8080 tiene un bus de direcciones de 16 bits, un bus de datos de 8 bits y bidireccional completamente decodificado, salidas de control compatibles con TTL, memoria interna 64 Kbytes de memoria ROM y RAM combinadas, el 8080 puede direccionar hasta 256 puertos de entrada y 256 puertos de salida; por lo tanto, podemos virtualmente expandir ilimitadamente el sistema.

Cuando este Microprocesador (MP) es combinado con memoria ROM (memoria de or-

cesu aleatorio) y memorias ROM (memoria de lectura iricamente), circuitos entrada-salida y registros de corriente, se complementa el sistema microcomputador MCS-80.

El conjunto de instrucciones del 8080 incluye bifurcaciones condicionales, así como, aritméticas binaria y decimal, instrucciones lógicas, instrucciones de transferencia de datos de registro a registro, instrucciones de control de stack e instrucciones de referencia a memoria. En efecto el conjunto de instrucciones del 8080 es suficientemente poderoso para rivalizar con el funcionamiento de muchos de los más costosos microcomputadores. Debido a las características antes mencionadas podemos decir que este microprocesador es relativamente fácil de interfacer.

### 1.1.2) Selección de la Calculadora Electrónica:

La elección de la calculadora se debe hacer en base a una determinación realista de los resultados deseados. Los retardos computacionales prolongados resultan ser un factor molesto e intolerable aún en un sistema computacional personal. Para iniciar la elección de la calculadora se listan las funciones que se requerirán. Luego se miden los tiempos de retardo necesario para la sincronización requeridos por el sistema proyectado. Tenemos mucho cuidado de acumular los retardos resultantes del uso repetitivo de la función más lenta de la calculadora.

Naturalmente la elección de una calculadora analógica, la cual aun requiere una mayor precisión (para conseguirse la habilitación de un número mayor de líneas de dígitos) y numerosas funciones (como una operación mediante un switch independiente), incrementaría el costo del circuito de la interfase.

La calculadora elegida para este trabajo es la calculadora R U V 11 L 91 S, en-

yas características técnicas listamos a continuación:

- Modelo** .....915
- Denominación** ..... UA123
- Operaciones de cálculo** ..... Adición, Sustracción, Multiplicación, División, Raíz Cuadrada.
- Funciones especiales**..... Constante, Porcentaje Automático, Cambio de Signo, Constante  $\pi$ , Cambio de Registros.
- Memoria** ..... Una memoria independiente para almacenamiento, con tecla de entrada y reemplazo y con indicación de memoria activa.
- Capacidad** ..... Entrada: 8 dígitos  
Resultados: 8 dígitos  $\times 10^3$
- Sistema de la coma** ..... Coma flotante con desplazamiento automático hacia la derecha (sistema Underflow) y coma flotante/fija.
- Elementos de Construcción Electrónica**.... 1 -  $\text{MSI}$  -  $\text{LSI}$  - Chip.
- Suministro de potencia**..... 4 pilas secas, 1.5 Voltio.  
Micro, modelo Nv1, IEC R 03.
- Dimensiones** ..... 120 mm  $\times$  73 mm  $\times$  19 mm  
(largo  $\times$  ancho  $\times$  alto)
- Peso** ..... aprox. 160 g. (incl. pilas).

### 1.1.3) Simulación del Teclado de la Calculadora:

Simular el teclado de la calculadora significa el poder "escribir" en la calculadora los datos adecuados para cuando se requiere realizar determinada operación.

Cada uno de los diferentes modelos de calculadoras emplean diferentes métodos para obtener la entrada del teclado. Sin embargo, es casi general el caso, en el cual, el teclado consiste de switches J. P. J. T. Una interfase general se puede implementar a partir de esta consideración; simulando el cierre de switches en paralelo con los switches del teclado en estudio.

Por lo tanto, el primer paso es determinar el formato del teclado de la calculadora escogida. Los formatos de los teclados más comúnmente utilizados son los siguientes:

A).- Cada uno de los switches es independiente de los demás, requiriendo conectar a cada switch dos líneas desde el circuito explorador de la calculadora.

B).- Los switches son diodos codificados; existe una línea común para todos los switches y una línea común conectada a cada uno de ellos.

C).- Los switches son explorados de manera multiplexada.

### 1.1.4) Demultiplexaje y Decodificación del Display:

Se requiere demultiplexar y decodificar el Display de la calculadora para poder "leer" de ella los resultados parciales o totales de alguna operación efectuada en ella, bajo el control del microprocesador, y así, proporcionar a cada dígito, dígitos en código B C U direccionados a la entrada del CPU.

Los resultados que queremos "leer" de la calculadora estarán expresados como -



un código multiplexado en siete segmentos, por lo tanto, necesitamos convertir la información multiplexada en siete segmentos a información en código BCD direccional.

#### 1.1.5) Desplazamiento a Niveles de Voltaje adecuados:

En esta parte nos referimos al desplazamiento de niveles de voltaje requeridos para el acoplamiento entre la calculadora y el CPU y viceversa.

Como el CPU comunica en niveles TTL mientras que la calculadora opera con uno o más voltajes CMOS, por lo que, necesitamos traducir los voltajes de control del teclado y del display de nivel CMOS a nivel TTL requeridos para establecer comunicación con el CPU del Microprocesador.

## CAPITULO 11

## INTERFASE DEL DISPLAY

Los resultados que queremos "leer" de la calculadora estarán expresados como un código multiplexado en siete segmentos. Por lo tanto, necesitamos convertir la información multiplexada en siete segmentos e información codificada en B C D, y — trasladar los voltajes de control del display de la calculadora a voltajes TTL — seguidos para establecer la comunicación con el Microprocesador.

## 11.1.- DESPLAZAMIENTO DE NIVELES DE TENSION.

Los voltajes de control, con los cuales, funciona el display de la calculadora son los siguientes:

9° Lógico . . . . . - 17 Volts.

0° Lógico . . . . . - 18 Volts.

Así mismo, es más exacto decir, que tenemos un intervalo de voltajes para cada nivel que varía conforme aumentan (ó disminuyen) el número de dígitos que aparecen en el display.

En el nivel lógico "1", tenemos la variación en el intervalo de -17 a -5 V. Para el "0" lógico, nuestro intervalo variable es de -18 a -22 Volts.

El display de la calculadora tiene en su parte inferior diecisiete terminales, los cuales, corresponden cada uno de ellos a un segmento o a un dígito. Estas terminales las hemos identificado mediante un número, dicha numeración fue realizada de izquierda a derecha desde 1 a 17. La relación entre cada una de estas terminales y sus respectivos segmentos o dígitos se lista a continuación.

- 1 - Barra (apaga el display).
- 2 - Dígito  $D_8$  (dígito más a la izquierda).
- 3 - Segmento "g".
- 4 - Dígito  $D_7$ .
- 5 - Segmento "c".
- 6 - Dígito  $D_6$ .
- 7 - Segmento "b".
- 8 - Dígito  $D_5$ .
- 9 - Segmento "a".
- 10 - Dígito  $D_4$ .
- 11 - Punto Decimal.
- 12 - Dígito  $D_3$ .
- 13 - Segmento "f".
- 14 - Dígito  $D_2$ .
- 15 - Segmento "e".
- 16 - Dígito  $D_1$ .
- 17 - Segmento "d".
- 18 - Dígito  $D_0$  (dígito más a la derecha).

La figura (11.1) muestra el display de la calculadora con sus respectivos terminales.

Debido a las voltajes variables que representan a nuestros niveles lógicos, - tenemos que trabajar con varios circuitos (divisor de tensión con resistencias, circuitos con diodos y resistencias, etc), antes de encontrar la forma más apropiada - para relacionar los niveles lógicos de nuestra calculadora a niveles TTL.

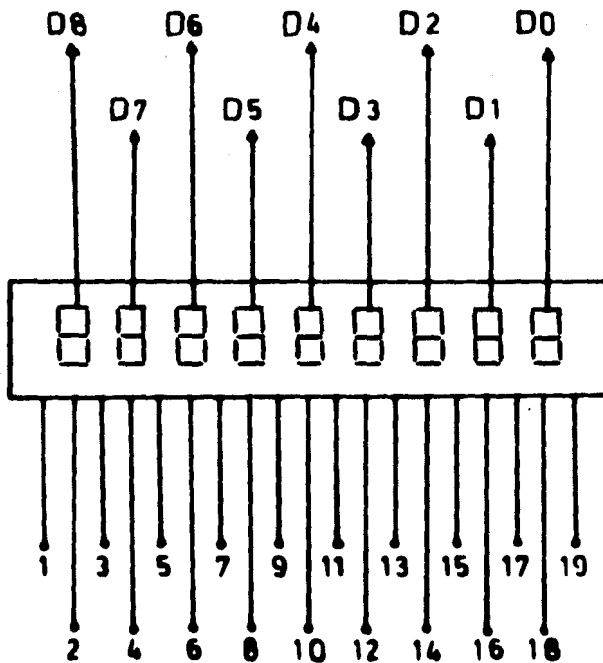


Fig (11.1).- Terminales y Dígitos del Display.

para el tratamiento de niveles encierrado utilizando Comparadores de Voltaje — (833), los cuales son cuatro comparadores integrados por chip y son absolutamente — binarios, cuando se les aplican los otros voltajes negativos que utiliza la calculadora.

El circuito que utilizamos para el tratamiento de niveles es muestra en la figura (11.2). En dicha figura observamos que los comparadores están polarizados por:

$$V_{CC} = 5 \text{ Voltos.}$$

$$V_{EE} = -30 \text{ Voltos.}$$

En la entrada inversora conectamos el voltaje de referencia que es de:

$$V_R = -17.5 \text{ Volts}$$

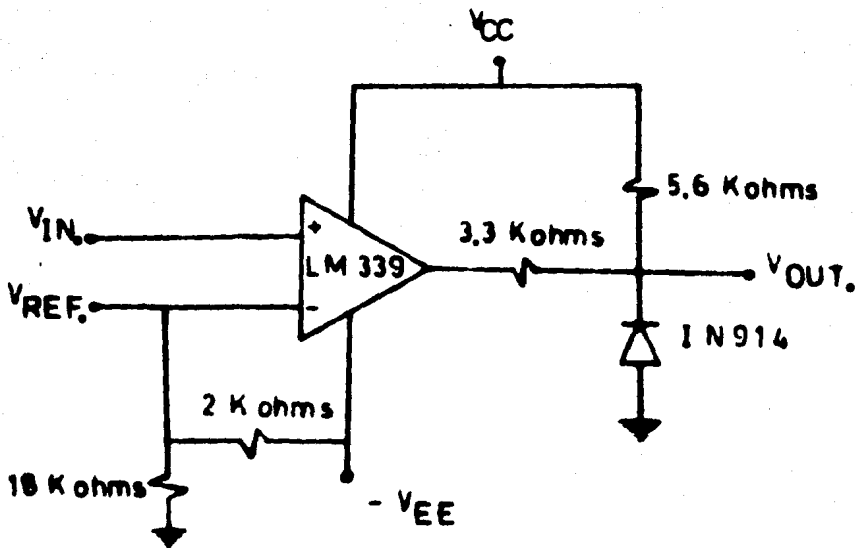


Fig 11.21.- Circuito Display de Niveles del Display.

Las entradas no inversoras de los comparadores las conectamos a las terminales de los segmentos del display.

Por lo tanto, cuando se ilumina el segmento al cual está conectado el comparador, a la entrada de este último tenemos entre  $-17 \text{ V}$  y  $-5 \text{ V}$  y a la salida del circuito obtenemos  $3 \text{ Volts}$ . Por otro lado, cuando el segmento no se encuentra iluminado tendremos a la entrada del comparador voltajes entre  $-18$  y  $-22 \text{ Volts}$  y a la salida aproximadamente  $0.7 \text{ Volts}$ . Leyendo de esta manera los niveles lógicos "1" y "0" avanzados a los circuitos TTL.

## 11.2.- CONVERSION DE 7-SEGMENTOS A CODIGO BCD.

Debido a que el C P U únicamente maneja información en código binario, es necesario convertir los datos del Display en forma de 7 - segmentos a código B C D — (Código Decimal Codificado en Binario).

Existen diversos métodos que pueden ser usados para convertir el código de siete - segmentos al formato B C D. Antes de escoger algún método en particular definiremos nuestros segmentos y la forma como están conectados los dígitos.

El ojo humano y por consiguiente el cerebro, requieren que los siete - segmentos tengan la forma que muestra la figura (11.3), para poder identificar a cada uno de los dígitos decimales. Pero la interfase para lograr la decodificación sólo requiere de cinco segmentos; a saber: a, b, e, f, g.

Los dígitos que identifican el circuito de interfase para su conversión a código B C D tienen la forma mostrada en la figura (11.4).

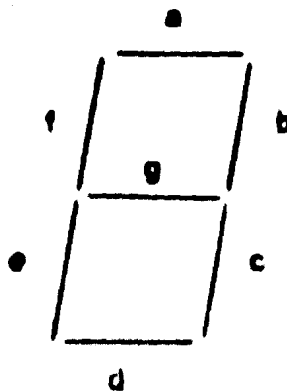


Fig (11.3).- Segmentos del Display.

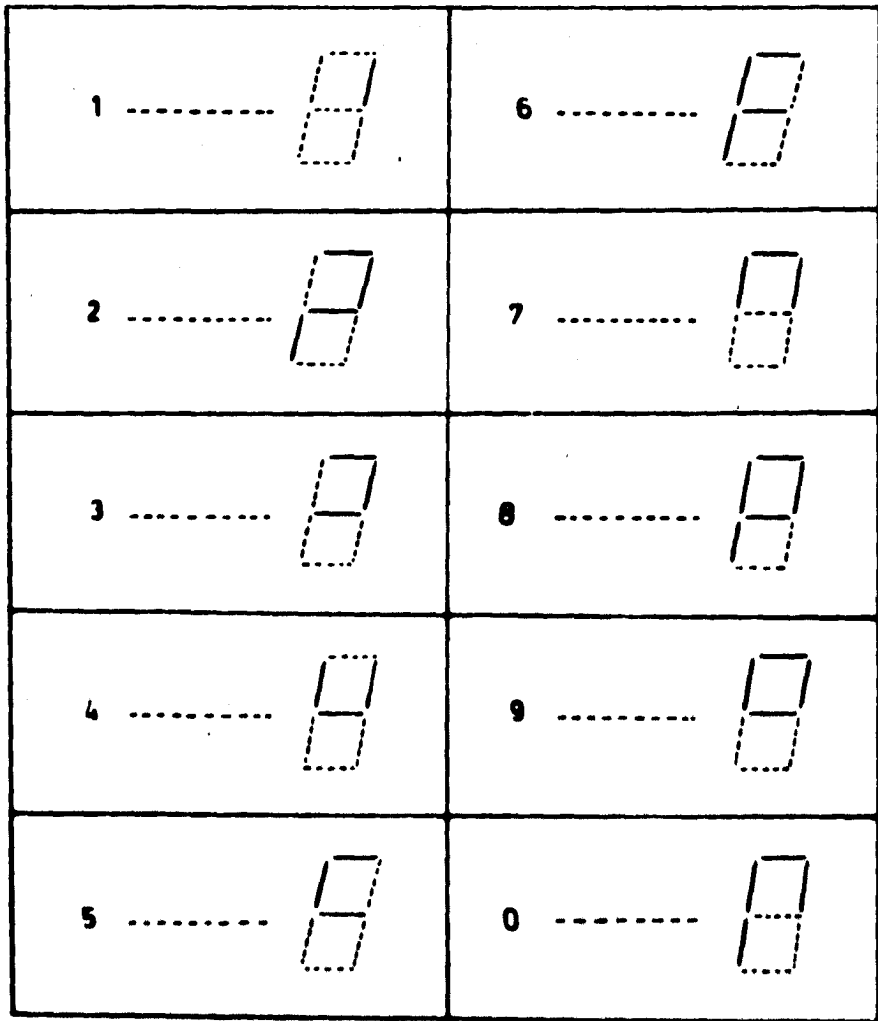


Fig 11.41.- Definición de Dígitos.

No existen circuitos integrados comerciales que realicen el cambio a código — B C D; pero si hay varios métodos que pueden utilizarse para tal fin. Uno de ellos — y el más común para obtener dicho circuito conversor, es el emplear técnicas tradicionales de diseño lógico (Algebras de Boole, Leyes de Morgan, Mapas de Karnaugh) — con compuertas NAND / NOR ó circuitos discretos, pero este método utilizaría varios chips y por lo tanto sería algo voluminoso.

Utilizando elementos MSI (Escala Media de Integración) la implementación del circuito conversor de código sería más compacto y quizás menos caro, con este tipo de dispositivos el diseño se basa principalmente en la observación minuciosa de las tablas de verdad y ecuaciones lógicas proporcionadas por los fabricantes.

Un dispositivo común utilizado en estos casos es el Multiplexor / Selector, el cual, consta de cierto número de entradas y salidas, entradas selectoras y terminales para sincronización con otros dispositivos de un circuito dado.

Para el caso de funciones lógicas de gran número de variables los Multiplexores simplifican el número de datos, conectando algunas variables de la función a sus entradas selectoras, por lo tanto, estas variables se eliminan reduciendo la complejidad de la función.

En nuestros casos, la función a implementar queda indicada en la tabla (1) donde cada uno de los bits ( $D_0, D_1, D_2, D_3$ ) del código B C D es función de cinco variables, que con los argumentos; a, b, c, f, g, del display utilizados por la interfase para la identificación de dígitos.

Para la implementación del circuito conversor de código se empleó el Multiplexor / Selector 3974/5, con el cual, se reduce a tres el número de variables del



código B C D.

El C. 1. SN74153 es un multiplexor dual de cuatro entradas, el cual selecciona dos datos de un bit provenientes de sus cuatro entradas mediante el control de sus dos entradas selectores (A, B). La sincronización con otros dispositivos se logra con las terminales  $G_1$ , por lo que debe existir un "cero" lógico en estas terminales ( $G_1, G_2$ ), para habilitar a cada multiplexor.

$\theta$	a	b	c	d	$\bar{D}_0$	$\bar{D}_1$	$\bar{D}_2$	$\bar{D}_3$
0	1	1	1	1	0	1	1	1
1	0	1	0	0	0	0	1	1
2	1	1	1	0	1	1	0	1
3	1	1	0	0	1	0	0	1
4	0	1	0	1	1	1	1	0
5	1	0	0	1	1	0	1	0
6	1	0	1	1	1	1	0	0
7	1	1	0	1	0	0	0	0
8	1	1	1	1	1	1	1	1
9	1	1	0	1	1	0	1	1

Tabla (11).- Código B C U (en función de 5 - segmentos).

(Este chip SN74153) hace posible generar dos funciones de tres variables, la tabla (12) nos muestra la tabla de verdad de este Multiplexor, y sus ecuaciones lógicas obtenidas del manual es:

$$Z_0 = (1_{10} \bar{a} \bar{b} - 1_{10} \bar{a} \bar{b} - 1_{20} \bar{a} \bar{b} - 1_{20} \bar{a} \bar{b}) \bar{E}_0$$

$$Z_b = I_0 \bar{A} \bar{B} - I_1 \bar{A} B - I_2 \bar{A} B - I_3 \bar{A} B \bar{E}_b$$

Donde:  $Z_{a,b}$  = Salidas

$I_i$  = Entradas

$A, B$  = Entradas Seleccionas

$E_{a,b}$  = Terminales de Sincronia

$A$	$B$	$I_0$	$I_1$	$I_2$	$I_3$	$E$	$Z$
X	X	X	X	X	X	1	0
0	0	0	X	X	X	0	0
0	0	1	X	X	X	0	1
0	1	X	0	X	X	0	0
0	1	X	1	X	X	0	1
1	0	X	X	0	X	0	0
1	0	X	X	1	X	0	1
1	1	X	X	X	0	0	0
1	1	X	X	X	1	0	1

Tabla (2).- Tabla de Verdad del SA74153.

Das chips SA74153 es representada para generar el código E C D, para cinco de -  
cinco bits, que están formados de tres variables si a las entradas seleccionas es las  
asignar dos variables de las cinco ya establecidas.

Por ser relativamente más sencillo, generamos primero el complemento del código  
BCD, obteniéndolo luego en su forma normal a través de Invertidores (SA7400).

A continuación se muestra como se genera el código  $\overline{BD}$  utilizando su tabla de verdad (Tabla 1), así como, las ecuaciones lógicas y tabla de verdad del Multiplexor.

Con el primer chip obtenemos los dos primeros bits ( $D_0, D_1$ ) del código  $\overline{BD}$ , — los cuales solo dependen de los segmentos e, f, g, si asignamos a las entradas selectoras de los multiplexores los segmentos a, b.

Para obtener el bit  $\overline{D}_0$  (así como todos los demás bits) debemos determinar las conexiones a cada uno de las entradas del Multiplexor respectivo. Para realizar esto obtenemos la tabla (3) observando en la tabla (1) como son los estados lógicos del bit para cada una de las combinaciones de las entradas selectoras (que son las que eligen la entrada que aparecerá a la salida del Multiplexor); así como los estados lógicos de los segmentos restantes determinando la relación existente entre el uno o algunos de ellos con respecto a la función lógica del bit a implementar.

Observando la tabla (3) establecemos:

- La combinación  $\overline{ab}$  no existe, esto es, cuando cualquiera de los dígitos que se iluminan en el display cuando suena alguno de los segmentos g y/o b se ilumina, por lo tanto, la entrada  $I_{00}$  del Multiplexor nunca será seleccionada y puede ser utilizada ó no, debido a ello la hemos conectado a la terminal positiva de la fuente de polarización ( $I_{00} \dots \dots \dots V$ ).

- La entrada  $I_{10}$  es seleccionada dos veces por la combinación de los segmentos  $\overline{ab}$  como se observa en la tabla 3a, al igual que la relación directa existente entre el bit  $\overline{D}_0$  y el segmento g; esto indica que este segmento debe ser conectado a la entrada  $I_{10}$  ( $I_{10} \dots \dots \dots a$ ).

- En la combinación  $\bar{a}b$ , la relación directa del bit  $\bar{D}_0$  es con los segmentos  $f$  y  $g$ , así como, con el bit  $D_2$ . Para en cierto modo asegurar la generación correcta de todo el código BCD asignamos el bit  $D_2$  a la entrada  $I_{2a}$  del Multiplexor — ( $I_{2a} \dots \dots \dots D_2$ ).

+ En la combinación  $a\bar{b}$  el segmento  $e$  es conectado a la entrada  $I_{3a}$  más está directamente relacionado con el bit  $\bar{D}_0$  ( $I_{3a} \dots \dots \dots e$ ).

Siguendo un criterio similar obtenemos el bit  $\bar{D}_1$  utilizando la tabla (3b) de la cual observamos:

- Al igual que en el bit  $\bar{D}_0$  no existe la combinación  $\bar{a}\bar{b}$  en las entradas selectoras, por lo que la entrada  $I_{0b}$  nunca será seleccionada pudiendo ser ó no utilizarse, también la conectamos a la terminal positiva de la fuente de polarización — ( $I_{0b} \dots \dots \dots +V$ ).

- Con la combinación  $a\bar{b}$  asignamos a la entrada el bit  $D_0$  por existir una correspondencia entre ambos bits ( $I_{1b} \dots \dots \dots D_0$ ).

- Cuando es seleccionada la entrada  $I_{2b}$ , mediante la combinación  $a\bar{b}$  deberá estar en el estado lógico "1" (en las dos ocasiones en que dicha combinación ocurre), por lo tanto, dicha entrada es conectada a la fuente de polarización ( $I_{2b} \dots \dots \dots +V$ ).

- Durante la combinación  $a\bar{b}$  el bit  $\bar{D}_1$  no tiene relación directa con alguno de los segmentos  $a, f, g$ , ó con los otros bits del código BCD; pero es posible generar mediante la operación lógica "Nó" entre el segmento  $f$  y el bit  $\bar{D}_2$  la cual debe ser conectada a la entrada  $I_{3b}$  ( $I_{3b} \dots \dots \dots f \cdot \bar{D}_2$ ).

D<sub>0</sub>:

a	b	$\bar{a}$	e	f	g
0	0	-	-	-	-
1	0	0	0	1	1
0	1	1	1	1	1
0	1	0	0	0	0
		1	0	1	1
		1	1	1	0
		1	1	0	1
1	1	0	0	0	1
		0	0	1	0
		1	1	1	1
		0	0	1	1

$$\begin{cases} I_0 = V_{CC} \\ I_1 = e \\ I_2 = f, g, \bar{D}_2 \\ I_3 = e \end{cases}$$

Tabla 3a

D<sub>1</sub>:

a	b	$\bar{a}$	e	f	g	$\bar{a} \cdot b$
0	0	-	-	-	-	-
1	0	1	0	1	1	1
0	1	0	1	1	1	0
0	1	1	0	0	0	1
		1	0	1	1	1
		1	1	1	0	1
		0	1	0	1	1
1	1	0	0	0	1	1
		0	0	1	0	0
		1	1	1	1	1
		1	0	1	1	1

$$\begin{cases} I_0 = -\bar{a} \bar{D}_2 \\ I_1 = D_0 \\ I_2 = V_{CC} \\ I_3 = f \cdot \bar{D}_2 \end{cases}$$

Tabla 3b

D<sub>2</sub>:

a	f	$\bar{a}$	b	e	g	$\bar{a}$
0	0	1	1	0	0	1
1	0	1	1	1	1	1
0	1	0	1	1	1	1
		1	1	1	0	1
		0	0	0	1	1
1	1	0	0	1	1	0
		0	1	0	0	0
		1	1	1	1	1
		1	1	0	1	1

$$\begin{cases} I_0 = V_{CC} \\ I_1 = b, g, V_{CC} \\ I_2 = Gnd. \\ I_3 = \overline{(e + f)} \cdot b \end{cases}$$

Tabla 3c

D<sub>3</sub>:

a	f	$\bar{a}$	b	e	g	$\bar{a}$
0	0	1	1	0	0	1
1	0	1	1	1	1	1
0	1	0	1	1	1	1
		1	1	1	0	1
		1	0	0	1	0
1	1	0	1	1	0	0
		0	1	1	1	1
		0	1	0	1	1

$$\begin{cases} I_0 = V_{CC} \\ I_1 = V_{CC} \\ I_2 = V_{CC} \\ I_3 = \bar{a} \cdot \bar{D}_2 \cdot b \cdot g \end{cases}$$

Tabla 3d

Para obtener los bits  $\bar{D}_2$  y  $\bar{D}_3$ , utilizamos un segundo chip 74153, conectando a sus entradas selectoras (A; B) los segmentos 'a', 'f'; quedando como variables los segmentos: b, e, g.

Continuando con criterio similar determinamos que cada una de las entradas del multiplexor deberá ser conectada de la forma siguiente.

Para el bit  $\bar{D}_2$ :

$$\begin{aligned} I_{0c} & \dots\dots\dots \rightarrow V_{CC} \\ I_{1c} & \dots\dots\dots 0 \text{ Volts (GND)} \\ I_{3c} & \dots\dots\dots \overline{(a + f) \cdot b} \end{aligned}$$

Para el bit  $\bar{D}_3$ :

$$\begin{aligned} I_{0d} & \dots\dots\dots \rightarrow V_{CC} \\ I_{1d} & \dots\dots\dots \rightarrow V_{CC} \\ I_{2d} & \dots\dots\dots \rightarrow V_{CC} \\ I_{3d} & \dots\dots\dots \overline{e \cdot D_2} \end{aligned}$$

Una vez, determinadas las conexiones correspondientes a cada una de las entradas de los multiplexores, en la fig. (11.5) se muestra el diagrama del circuito de codificación, los Circuitos Inversores 7410 que se encuentran en esta figura conectados a cada una de las salidas de los multiplexores con fines de convertir el complemento del código B C U a su forma normal.

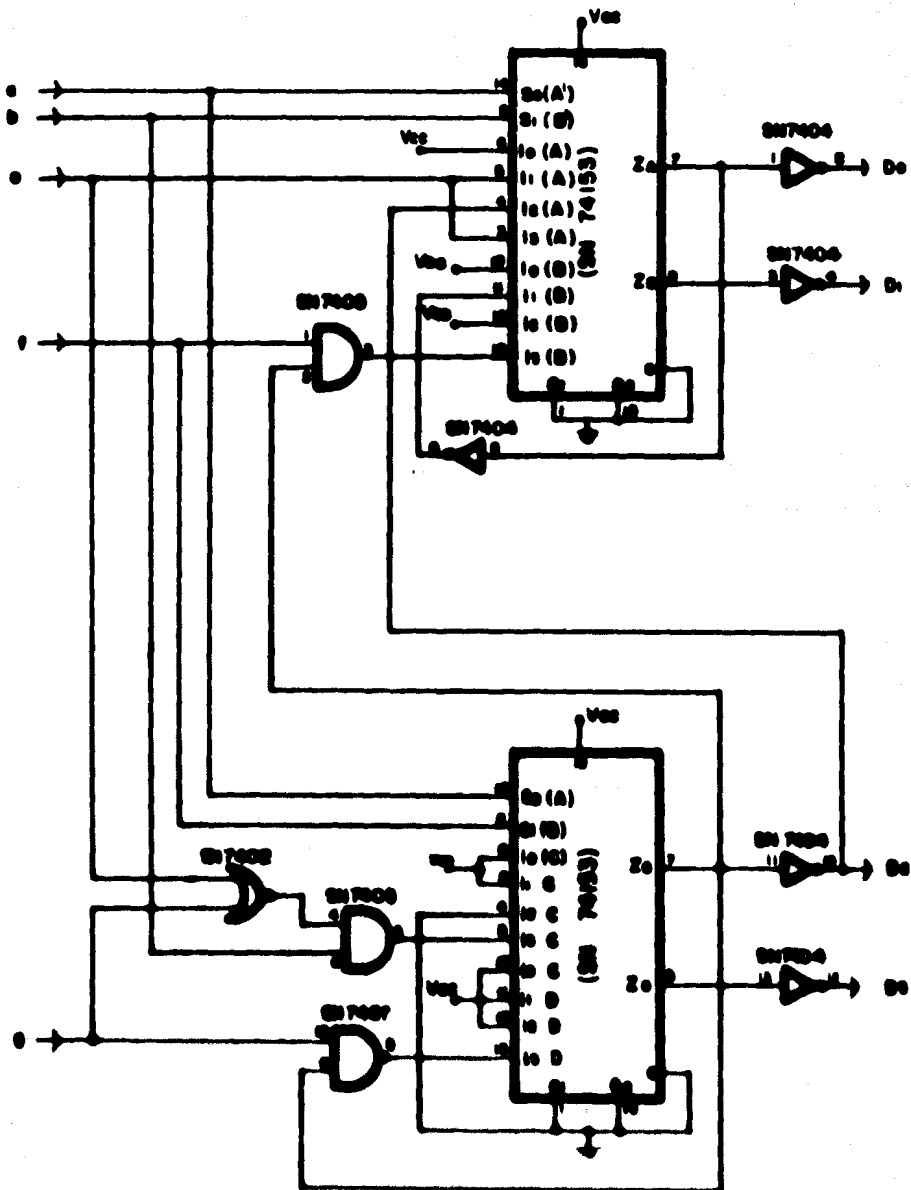


Fig. (2.6) CIRCUITO DE INTERFASE DEL DISPLAY

### 11.3).- DIRECCIONAMIENTO DE LOS DÍGITOS.

Para sincronizar la conversión del formato de siete - segmentos al código BCD se necesita direccionar el dígito a "leer" de la calculadora ( $D_0, D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8$ ).

Para lograr este direccionamiento se utiliza el Circuito Integrado SN74151, - que es un Multiplexor /Selector de Datos, El cual, mediante tres líneas de control nos selecciona una entrada determinada de las ocho existentes.

Estas 8 entradas provenientes de las terminales del display, al igual que las terminales de los segmentos son desplazadas primeramente, antes de ser conectadas - al Multiplexor /Selector. Estas terminales del Display se muestran en la fig (11.1).

Los bits  $B_5, B_6, B_7$  provenientes del puerto "B" del P. P. 1. (8255) nos dan la dirección del dígito a "leer".

Además el bit  $B_0$  nos indica si se realiza la operación de leer ( $B_0 = 1$ ) ó se - escribe ( $B_0 = 0$ ) en la calculadora. Cuando el bit  $B_0$  es "1" habilita al SN74151 para la selección del dígito, por lo tanto, se realiza la función de leer. El circuito selector de dígitos se muestra en la figura (11.6).



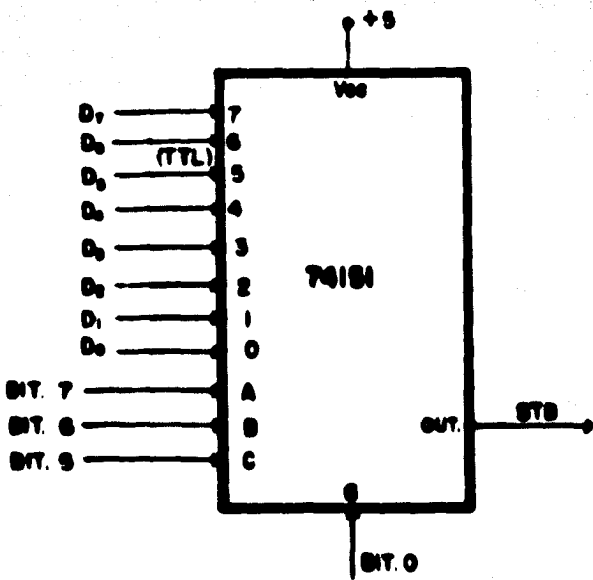


FIG. (3.6) CIRCUITO SELECTOR DE DIGITOS.

## CAPITULO III

## INTERFASE DEL TECLADO

Como Interfase del Teclado nos referimos a la parte del circuito, que es encargada de interpretar las señales generadas por el C P U para realizar una determinada operación en la computadora.

Las señales son recibidas a través de la Interfase Periférica Programable (P. P. I. - 8255) y consisten de ocho bits (un byte) que es la longitud de palabras del Microprocesador.

El byte de información proveniente del Microprocesador habilita el circuito — que simulará la operación de una tecla mediante un código preestablecido (en el capítulo V se describe dicho código), de acuerdo a la estructura del teclado.

## III.1).- ESTRUCTURA DEL TECLADO.

Para determinar la estructura del teclado seguimos dos métodos:

- i).- Analizando directamente el dibujo (las pistas) del circuito impreso del teclado.
- ii).- Mediante un Osciloscopio se determinaron las relaciones existentes entre las teclas presionadas y las terminales interconectadas.

Mediante el primer método y posteriormente confirmado por el segundo se obtuvieron las relaciones que se listan en la tabla 4.

En las embocaduras existen en el top del circuito impreso del teclado una serie de terminales, las cuales forman el punto de contacto para el teclado y las —

TECLAS	TERMINALES INTERCONECTADOS
CL	11 y 15
C	15 y 14
√	8 y 16
π	15 y 4
(-)	8 y 15
8	16 y 9
2Y	15 y 7
7	17 y 7
8	17 y 6
9	4 y 17
÷	16 y 11
⊖	9 y 15
4	11 y 17
5	17 y 9
6	17 y 8
X	12 y 16
⊙	15 y 6
1	16 y 17
2	17 y 13
3	17 y 12
-	13 y 16
→	4 y 16
0	15 y 13
.	12 y 15
=	16 y 7
•	14 y 16
⊞	16 y 6

Tabla 4.- Relación entre teclas y terminales interconectados.

LINEA COMUNA 15. TECLA	TERMINALES INTERCONECTADAS
π	4 y 15
⊖	6 y 15
X-V	7 y 15
(-)	8 y 15
CM	9 y 15
CL	11 y 15
.	12 y 15
0	13 y 15
C	14 y 15
<b>LINEA COMUNA 16.</b>	
B	4 y 16
B	6 y 16
=	7 y 16
√	8 y 16
B	9 y 16
+	11 y 16
X	12 y 16
-	13 y 16
+	14 y 16
<b>LINEA COMUNA 17.</b>	
9	4 y 17
8	6 y 17
7	7 y 17
6	8 y 17
5	9 y 17
4	11 y 17
3	12 y 17
2	13 y 17
1	14 y 17

Tabla 5.- Líneas Comunes.

enumeramos desde uno hasta diecinueve (de izquierda a derecha vista de frente), y estos son los números de las terminales interconectadas que aparecen en las tablas 4 y 5.

De la tabla 4 se observa que existen tres líneas comunes, las cuales son las terminales 15, 16 y 17.

A continuación reordenamos la tabla 4 agrupándola de acuerdo a las líneas comunes, y de esta manera, observas con mayor claridad la estructura del teclado en la tabla 5.

La tabla 5 muestra que la terminal 17 es común a todos los dígitos exceptuando el cero; La terminal 16 es común a las operaciones aritméticas, así como, a las funciones de memoria  $M_0$  y  $M_1$ ; y la terminal 15 es común a las funciones:  $\pi$ ,  $\theta_{\text{out}}$ ,  $X - Y$ ,  $-$ ,  $\text{On}$ ,  $\text{Cl}$ ,  $\cdot$ ,  $\circ$ ,  $C$ . De aquí en adelante al referirnos a las líneas comunes, lo haremos de la siguiente manera:

- La línea común 15 será la línea de las funciones especiales.
- Terminal 16 es la línea de operaciones aritméticas.
- Terminal 17 corresponde a la línea de dígitos.

### III.21.- SIMULACION DEL TECLADO.

Una vez determinado el formato del teclado pasamos a la simulación del mismo. Para lograr esto existen varias técnicas, primero por ejemplo conectamos relévoladores en paralelo con los switches del teclado, pero el costo y espacio requeridos por esto último lo hacen poco aceptable. Mediante switches implementados con transistores también podemos simular el presionar los teclas.

Sin embargo, existe una mejor solución la cual es menos cara y más compacta, - nos referimos a los Switches analógicos CMOS. Nos enfocamos por este último método para implementar la simulación del teclado.

El switch analógico CMOS emplea una compuerta de transmisión. La compuerta es el equivalente de un relé de estado sólido, cuando la línea de control está en "1" lógico la compuerta deja pasar señales analógicas bidireccionalmente presentando típicamente de 80 a 150 Ohms. La compuerta dejará pasar solamente aquellas señales cuya amplitud no exceda el rango entre sus fuentes de polarización. Cuando la línea de control está en "0" lógico la compuerta actúa como un circuito abierto --- 110<sup>12</sup> Ohms).

En las terminales de la calculadora existen voltajes comprendidos entre 0 y -22 Volts; por lo tanto el switch analógico deberá ser polarizado con estas tensiones. De esta manera aseguramos que cualquier posible nivel de voltaje presente en el teclado de la calculadora pasará a través de la compuerta simulándonos el cierre de un switch del teclado.

Los niveles de voltaje para la polarización de la compuerta determinan cuales deben ser los voltajes de control que nos produzcan las condiciones "on" y "off" - deseadas. Por lo tanto, para un "off" el nivel de voltaje de control deberá estar cercano a -22 Volts y para la condición "on" deberá estar cercano a 0 Volts.

### 111.3).- INTERFASE DEL TECLADO:

Para escribir en la calculadora, recibimos del puerto B de la Interfase Perifé-rica Programable (8255) dirigida por el CPU del 8080: ocho bits, los cua-

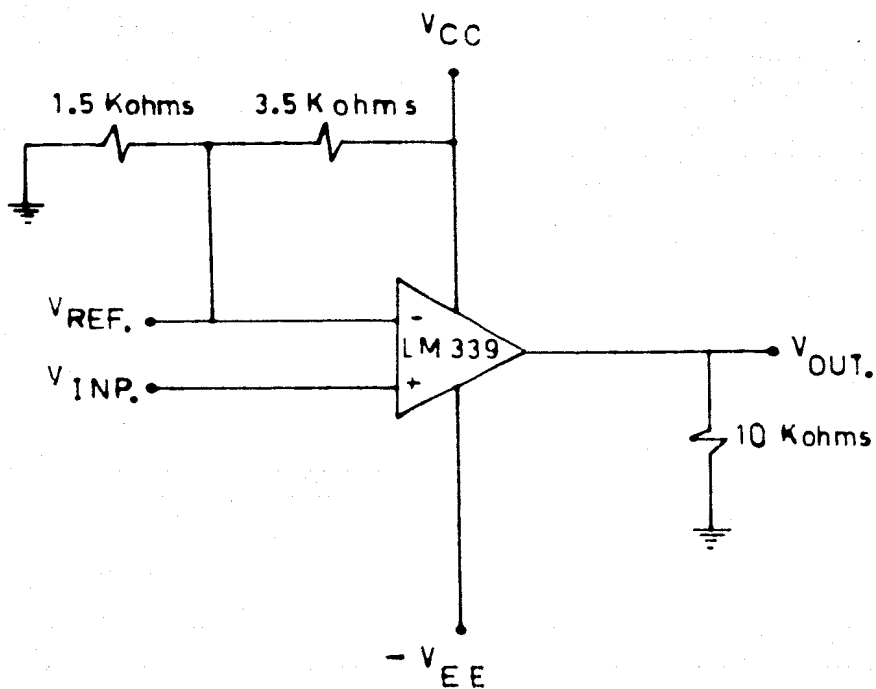
Las son ordenados en un formato; donde los bits  $B_1$ ,  $B_2$  y  $B_3$  seleccionan las líneas comunes obtenidas en la tabla 5. El bit  $B_0$  nos indica si se realiza la operación de leer o de escribir. Los bits  $B_4$ ,  $B_5$ ,  $B_6$  y  $B_7$  forman el código de la tecla o — opina en la calculadora.

En la figura (111.2) se muestra el circuito de Interfase del Teclado. En ella se observa que las líneas comunes son habilitadas por Flip - Flops tipo D (SA7474), los relojes de estos son sincronizados por los One - Shots (SA74121) y estos últimos son disparados por la señal OBF proveniente del PPI - 8255. La señal OBF aparecerá a la entrada del One - Shot cuando la calculadora se encuentre lista para recibir un dato proveniente del BUCO. Por lo tanto el Microprocesador antes de enviar otro dato esperará que transcurra el tiempo de respuesta de la calculadora, y esto será cuando la señal ACR sea baja.

Los bits  $B_4$ ,  $B_5$ ,  $B_6$  y  $B_7$  son utilizados por el SA74154 (Decodificador / Demultiplexor de cuatro a dieciséis líneas), para seleccionar una de las nueve líneas — comunes del teclado.

Los niveles de tensión TTL en la salida del Decodificador / Demultiplexor (SA74154) y de los flip - flops (SA7474) debemos convertirlos a niveles requeridos para la tensión de control en los Switches Analógicos CMOS (CD4016).

Para efectuar esta conversión de niveles de tensión empleamos Comparadores de Voltaje (LM33) como se muestra en la figura 111.1; la señal de entrada a los comparadores de voltaje es de cero y cinco voltos y las tensiones de salida obtenidas — son de -22 y 0 Valtos respectivamente.



$V_{CC} = 5$  Volts.

$V_{EE} = -22$  Volts.

$V_{inp.} = 0 - 5$  Volts

$V_{out.} = -22 - 0$  Volts.

Fig (111.1).- Circuito Desplumion de Niveles del Teclado.





## CAPITULO IV

## CIRCUITO DE INTERFASE

A través de los capítulos anteriores hemos desarrollado el circuito de interfase de la calculadora explicando por bloques funcionales, sin embargo, no se han mencionado las señales de control y sincronía que son requeridas por el sistema Microprocesador - Interfase - Calculadora. En este capítulo describiremos la integración de los bloques funcionales para formar el diagrama final del Circuito de Interfase que se muestra en la figura (IV.1).

## IV.1).- ETAPA DE ESCRITURA:

En el instante en que el Microprocesador requiere realizar alguna operación aritmética envía a la calculadora, a través del P P 1 - 8255 y de la Interfase, -- los datos necesarios codificados de la memoria que se muestra en la tabla (V.3). Estos datos serán recibidos en la Interfase por los Buffers / Drivers SN7407 que tienen como función proporcionar la corriente necesaria para los circuitos pulseros.

Para "byálmia" una fecha determinada necesitamos habilitar los switches de una línea no común, lo cual se realiza mediante el Decodificador / Demultiplexor SN74154 que dirige el switch correspondiente de acuerdo a la combinación de los bits  $P_4$ ,  $P_5$ ,  $P_6$  y  $P_7$ .

Los bits  $P_1$ ,  $P_2$  y  $P_3$  seleccionan el switch correspondiente a la línea común, -- esta información es retenida previamente en los Flip - Flop tipo D (SN7474).

Por cada byte de datos que envía el C P U, el P P 1 - 8255 proporciona la co-

La señal U B F, la cual le indica al periférico (en nuestro caso la Interfase) que le ha sido enviado un dato. Esta señal es utilizada para habilitar los flip - flops tipo D, los cuales liberan la información almacenada que acciona el switch de la línea común elegida. La señal U B F también es utilizada por el One - Shot SA74121(A) que genera el tiempo de espera necesario para la reacción de la Calculadora, después del cual, los bits previamente almacenados en los flip - flops (SA7474) son borrados, quedando de esta manera los flip - flops en condiciones de recibir el próximo dato que envíe el C P U.

La señal A C K generada por la Interfase le indica al C P U que sus datos han sido recibidos y están siendo procesados. Esta señal es generada por el One - Shot SA74121(B), una vez que ha sido recibida la señal U B F y habiendo transcurrido el tiempo de respuesta de la calculadora.

#### IV.2).- E T N P N D E L E C T U R A :

Una vez que el C P U ha enviado los datos necesarios para realizar la operación deseada, la calculadora procesa esta información y el resultado obtenido es enviado como respuesta hacia el Microprocesador.

El Microprocesador "lee" dígito a dígito el resultado que recibe del Display, para lo cual, envía al circuito de Interfase una palabra de control indicando al dígitos que desea "leer", esto es con la información contenida en los bits  $B_5$ ,  $B_6$  y  $B_7$ . En el circuito de Interfase estos bits son utilizados por el Multiplexor /Selector de datos SA74151 para elegir la línea correspondiente del dígito a leer; y a su vez genera la señal S T B en conjunción con los operandos  $a$  y  $b$ , - (datos para suministrar eficientemente la operación del dígito seleccionado en el display).

Con la señal  $\bar{S}T$  el Microprocesador acepta el código B C D del dígito leído. Cuando el SN74151 selecciona el dígito a leer por el C P U, su salida activará el Latch SN7475 (A) liberando el código B C U de dicho dígito, el cual, es aceptado por el puerto n del P P I - 8255; así como también liberará la indicación de punto decimal (P. D.) y entera (O V F) generadas por el Latch SN7475 (B). Las señales P. D. y O V F son obtenidas directamente de los terminales respectivos del display y decodificador a niveles T T L (como fue realizado en los segmentos y dígitos) antes de ser almacenados en el Latch.

El Microprocesador en la operación de lectura como de escritura envía una palabra de ocho bits hacia la Interfase, por la cual, es necesario que el circuito de Interfase sea capaz de reconocer si esta palabra es un dato a escribir en la calculadora ó es la palabra de control que indica el dígito a leer. Para determinar la función a realizar se utiliza el bit  $B_0$ , el cual, si es un "1" lógico indica el proceso de lectura, en caso de "0" lógico indicará que se realiza la escritura de un dato.

Cada vez que es leído un dígito por el C P U, el P P I le indicará al circuito de Interfase que el dato ha sido capturado y aceptado en su Latch de entrada, esto es un "1" lógico en terminal de salida I B F. Esta señal I B F es aplicada a la entrada "Clock" del flip - flop tipo D (SN7474), haciendo el dígito seleccionado - proveniente por el SN74151, y de esta manera iniciar el proceso de lectura del dígito siguiente.



#### IV.31.- IMPLEMENTACION FISICA DEL CIRCUITO DE INTERFASE:

El circuito de interfase fue implementado físicamente en 10 tarjetas, quedando dividido en 5 tarjetas para el display y 5 tarjetas para el teclado. A continuación explicamos la función principal de cada tarjeta.

##### A1.- Sección Del Display:

Las tarjetas A y C son los circuitos destinados a desplegar los niveles de tensión. La tarjeta A desplaza los niveles existentes en los segmentos, el punto decimal y la indicación de saturación; y la tarjeta C se encarga de lo mismo en las líneas que activan a los dígitos. Ambas tarjetas son idénticas dado que los voltajes existentes en toda la sección del display son iguales. Estas tarjetas están compuestas de los chips LM339, resistencias y diodos.

La tarjeta A contiene el circuito decodificación de 7 - segmentos a código BCD. Está compuesto por dos chips SA74153, un Latch SA7495, Compuertas AND SA7400, Inversoras SA7404 y un Chip SA7408.

En la tarjeta B se encuentra el circuito colector de dígitos formado por el chip SA7475, un chip de compuertas AND Buffer SA7437 y compuertas OR SA7432.

El circuito colector del punto decimal o indicación de saturación compuesto por un chip SA7475, se encuentra en la tarjeta E. También en esta tarjeta hay un chip SA7406 (Inversoras buffers - drivers) utilizados en las señales U B F y A C K.

##### B1.- Sección Del Teclado:

Las tarjetas B, G, H, I, J corresponden a esta sección. La tarjeta F contiene los buffers - drivers que reciben los pulsos e impulsan a los otros circuitos de se-

ta sección. También contiene a los flip - flops tipo D que activan a la línea común del teclado que es elegida.

La tarjeta G está formada por los One - Shots SN74121; uno que borra los datos almacenados temporalmente en los flip - flops tipo D, y otro que genera la señal A C K.

La tarjeta H selecciona una de las nueve líneas no comunes, por lo tanto contiene al Selector / Demultiplexor SN74154, así como a los inversores SN7406, que son utilizadas para convertir en "1" lógico la línea seleccionada y en "0" lógico las restantes.

La tarjeta I contiene los comparadores de voltaje LM339 que convierten la tensión TTL en niveles requeridos por los circuitos M O S (de la calculadora).

Finalmente la tarjeta J contiene los switches analógicos Q4016 (tres chips - que simulan la operación de una tecla de la calculadora).

Las tarjetas fueron colocadas en una base de mica de 0.35 x 0.25 x 0.15 metros, como se muestra en las fotografías 1 y 2. Habiendo designado a las tarjetas como se enciende anteriormente A...J (de derecha a izquierda).

En la fotografía 3 se muestra la calculadora utilizada en el trabajo y en la fotografía (4) se muestra una de las 10 tarjetas (la A), la cual mide igual que antes las demás 10 por 15 cm.

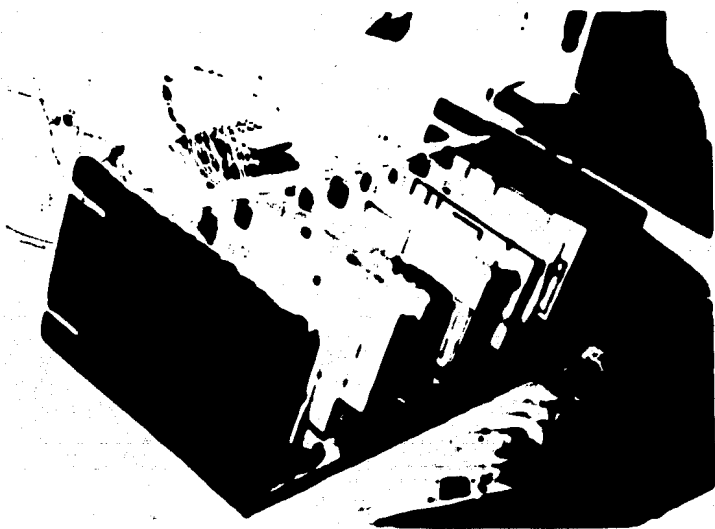
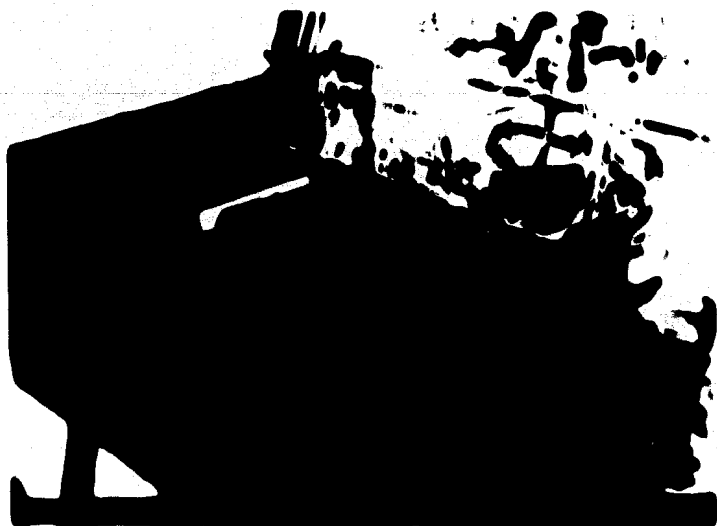


Fig. (1) Circuito de Inicijuse (Vista lateral izquierda).





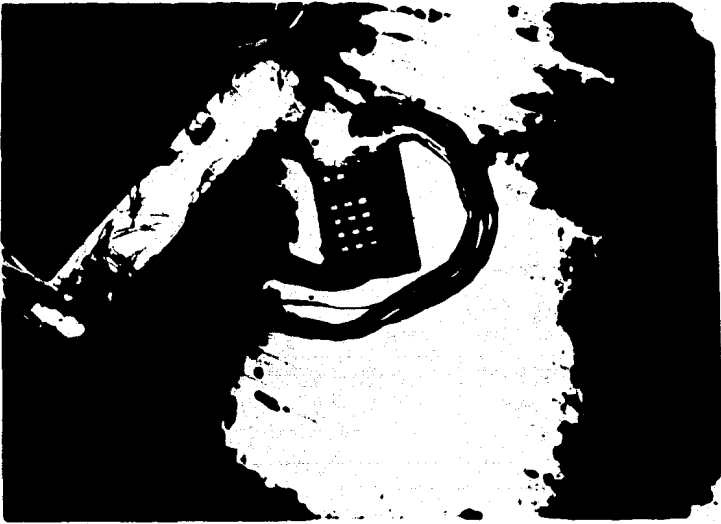


Foto (3) Calculadora electrónica.



Foto (4) Tarjeta de selección de las líneas de conexión.

### CAPITULO V

## INTERFASE PERIFERICA PROGRAMABLE

El 8255 es un dispositivo de interfase periférica programable debido a que su configuración funcional, es programada por el sistema Software de modo que no hay necesidad de incluir lógica externa a este dispositivo. La siguiente figura muestra en forma esquemática los bloques fundamentales que constituyen este dispositivo, y a continuación se describen sus rasgos fundamentales de acuerdo a cada bloque.

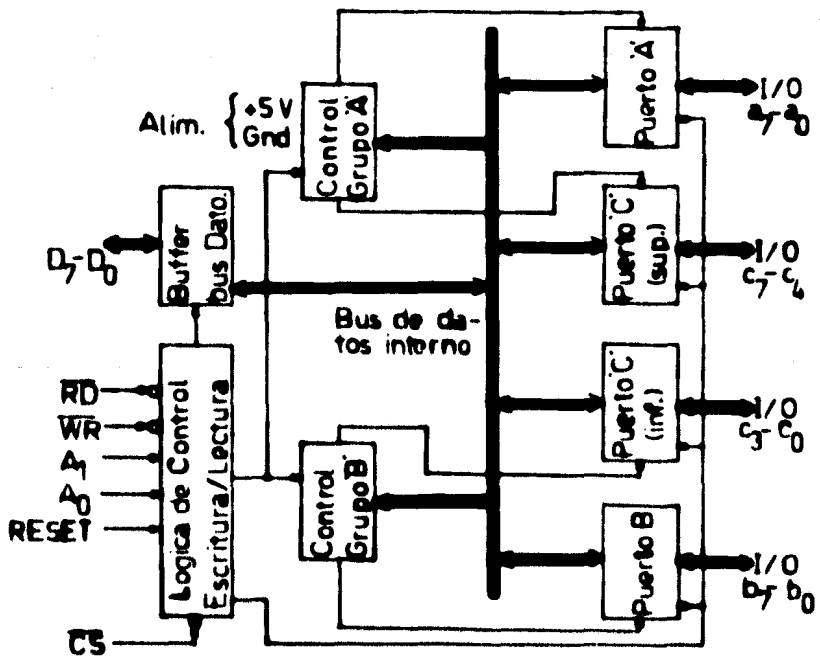


Fig. IV.11.- Esquema de la 8255.

El buffer bidireccional de tres estados asociados al bus, es usado como interfase entre el bus del sistema del 8080 y el 8255. El dato es transmitido o recibido por el buffer en la ejecución de las instrucciones de Entrada / Salida (I / O).

También pueden ser transmitidos a través del buffer, palabras de control e información de status.

La función del bloque marcado como Lógica de Control y Lectura / Escritura, es manejar todas las transferencias internas y externas tanto del dato como palabras de control o de status. Acepta entradas de las buses de control y dirección y a su vez, emite comandos a los dos grupos de control A y B. La terminal  $\overline{RD}$  (read) permite con un nivel bajo el envío de datos o información de status del 8255 al CP  $\overline{U}$  - del 8080 (en su bus de datos). En ocasiones, permite que el 8080 escriba un dato o palabras de control en el 8255. Las terminales  $A_0$  y  $A_1$  (para select 0 y para select 1) en conjunción con las entradas  $\overline{RD}$  y  $\overline{WR}$ , controlan la selección de uno de los tres puertos o al registro de palabras de control. Estas terminales son normalmente conectadas a los bits menos significativos del bus de dirección ( $A_0, A_1$ ).

La siguiente tabla de verdad de la combinación de estas cinco terminales identifica la operación que se ejecuta.

$A_1$	$A_0$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	OPERACION DE ENTRADA
0	0	0	1	0	PUERTO A → BUS DE DATOS
0	1	0	1	0	PUERTO B → BUS DE DATOS
1	0	0	1	0	PUERTO C → BUS DE DATOS
0	0	1	0	0	BUS DE DATOS → PUERTO A
0	1	1	0	0	BUS DE DATOS → PUERTO B
1	0	1	0	0	BUS DE DATOS → PUERTO C
1	1	1	0	0	BUS DE DATOS → CONTROL
X	X	X	X	1	BUS DE DATOS → DO. ESTADO
1	1	0	1	0	OPERACION LIBRE

La línea  $\overline{RESET}$  con un nivel alto, borra todos los registros internos incluyendo el registro de control y todos los puertos (A, B, C) están conectados en su modo de entrada.

La configuración funcional de cada uno de los puertos es programada por el sistema software. En esencia, el CPU - 8080 envía una palabra de control al 8255. La palabra de control contiene información del modo como será utilizado cada uno de los puertos, el activar o inhibir algún bit, etc; que inicializa la configuración funcional del 8255.

Cada uno de los bloques de control Lectura / Escritura, reciben palabras de control del bus de datos y envían los comandos apropiados a cada uno de sus puertos asociados.

El grupo A, comanda al puerto A y a los 4 bits superiores del puerto C ( $C_7 - C_4$ ).

El grupo B, comanda al puerto B y a los 4 bits inferiores del puerto C ( $C_3 - C_0$ ).

El registro de palabras de control sólo puede recibir palabras de control, es decir que no puede ser leído.

Puerto A.- Un "data output latch/buffer" y un "data input latch" ambos de 8 bits.

Puerto B.- Un "data input/output latch buffer" y un "data input buffer" ambos de 8 bits.

Puerto C.- Un "data output latch/buffer" y un "data input buffer" ambos de 8 bits. Este puerto puede ser dividido en dos puertos de 4 bits. Cada puerto de 4 -

bits contiene un latch y este puede ser usado para emitir las señales de control - en conjunción con los puertos A y B.

### V.1).- SELECCION DEL MODO:

Hay tres modos básicos de operación que pueden ser elegidos por medio del Software.

Modo 0. "input / output" básico.

Modo 1. "Stratol input / output".

Modo 2. Bus bidireccional.

Después que la señal de RESET es removida el 8255 permanece en modo de entrada sin ninguna iniciación adicional. Durante la ejecución de un programa cualquier modo puede elegirse usando tan solo una instrucción de "output". Esto permite a un único 8255 dar servicio a una variedad de dispositivos con una simple rutina - software.

Los pines del puerto A y el puerto B, pueden ser definidos separadamente, así como que el puerto C es dividido en dos funciones según lo requiere el puerto A y el puerto B. Todos los registros serán borrados siempre que sea cambiado el modo. Los modos pueden ser combinados de manera que en definición funcional puede ser exigido a cual cualquier estructura de I/O. La figura (V.2) esquematiza los tres modos utilizables y la palabras de control que los definen.

También por una palabra de control, cualquiera de los 8 bits del puerto C pueden ser disparados o borrados (usando una instrucción de salida). Cuando el puerto C está siendo usado para señales de status y control por los puertos A o B, todos los bits del puerto C pueden ser disparados o borrados por el uso de la operación ---

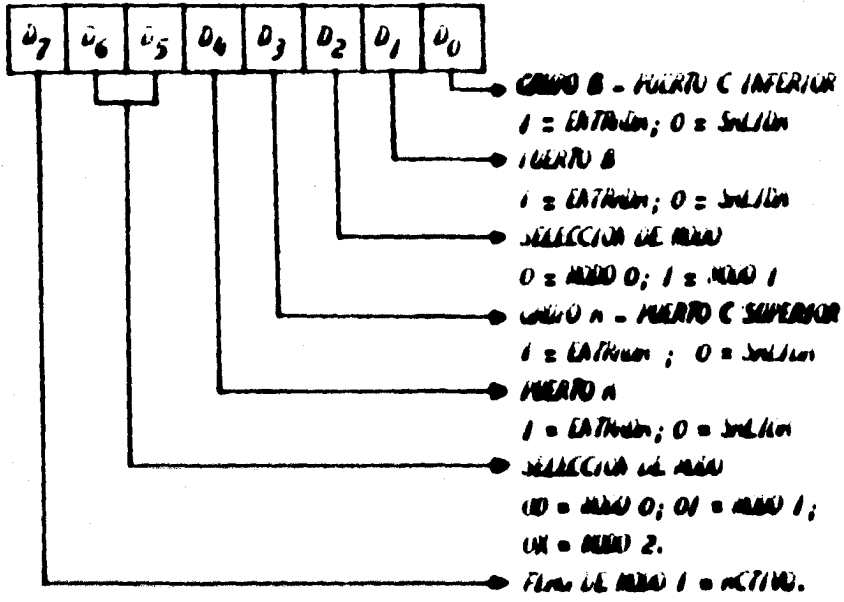
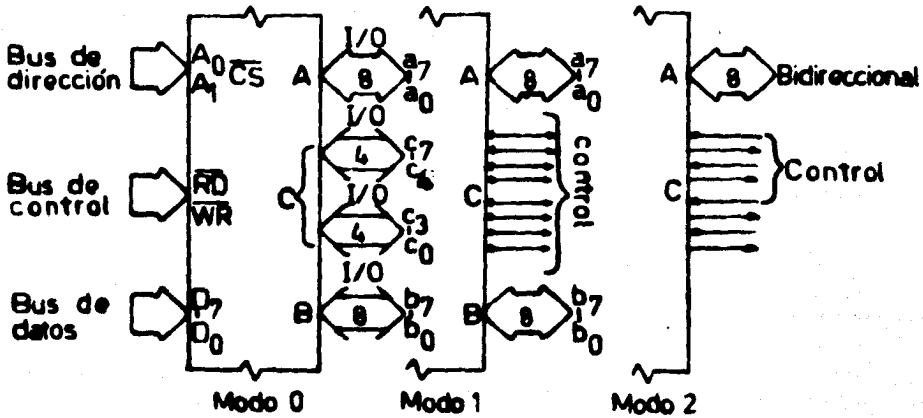


Fig. IV.2).- Modos de Operación y Palabras de Control del P. P. 1.- 8255.

"bit set / reset".

Cuando el 8255 está programado para operar en el Modo 1 o el Modo 2, son proporcionadas señales de control que pueden ser usadas como entradas al C P U, que hacen la petición de Interrupción. Estas señales, generadas en el puerto C, pueden ser inhibidas o permitidas por medio del "seteo" o "reseteo" del flip flop INTE asociado al dispositivo, usando tan solo la operación de "bit set/reset" del puerto C. Esta función permite al programador capacitar o no al dispositivo para que haga la interrupción del C P U sin afectar a ningún otro dispositivo en la estructura de interrupción.

al definir la palabra de control que establece el modo de seteo o reseteo cualquier bit del puerto C, se puede decir con relación a la manera de permitir o no la interrupción que:

Si (Bit Set) - el flip flop INTE es pyme set - Interrupción posible.

Si (Bit Reset) - el flip flop INTE es pyme reset - Interrupción no permitida.

La figura (V.3) muestra el formato de la palabra de control para la operación "bit set / reset".

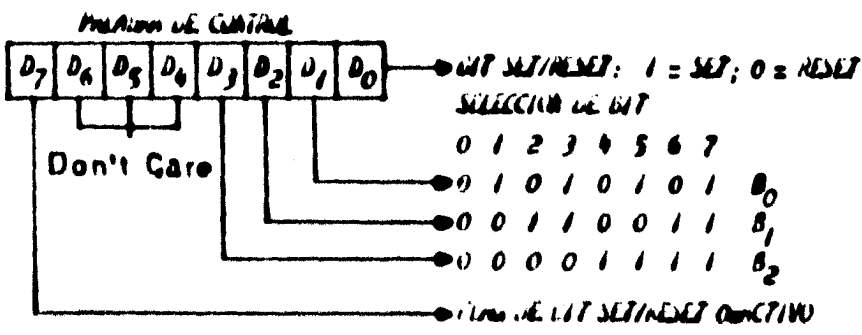


Fig (V.3).- Palabra de control para "Bit Set / Reset".

## V.2).- MODOS DE OPERACION DEL 8255.

### A) Modo 0:

Esta configuración funcional, proporciona una simple operación de Entrada/Salida, para cada uno de los tres puertos. No se requieren señales previas de reconocimiento para iniciar la transferencia de datos, sino que simplemente se escriben o se leen de algún puerto específico.

Puede definirse funcionalmente al Modo 0, por medio de las siguientes características:

- 1) Dos puertos de 8 bits y dos puertos de 4 bits.
- 2) Cualquier puerto puede ser de entrada o salida.
- 3) Los datos de salida son almacenados temporalmente en un circuito Latch.
- 4) Los datos de entrada no son almacenados en Latch.
- 5) Las 16 configuraciones posibles de entrada y salida se muestran en la tabla IV.11.

### B) Modo 1:

Esta configuración funcional proporciona señales que determinan el paso de algún dato o palabras o transferencias de/a un puerto específico. En este modo el puerto A y el puerto B usan las líneas del puerto C para generar o aceptar estas señales.

La definición funcional del Modo 1, puede resumirse de la siguiente manera:

- 1) Dos grupos Grupo A y Grupo B1.
- 2) Cada grupo contiene 8 bits de datos como puerto de Entrada / Salida y 4 bits de líneas de control de datos para el puerto.
- 3) El puerto de 8 bits de datos puede ser tanto de entrada como de salida, y - en estos casos los datos son almacenados temporalmente en un Latch.



n		B		GRUPO n		GRUPO B	
$u_4$	$u_3$	$D_1$	$D_0$	HUERTO n	HUERTO C	HUERTO B	HUERTO C
					Superior		Inferior
0	0	0	0	salida	salida	salida	salida
0	0	0	1	salida	salida	salida	entrada
0	0	1	0	salida	salida	entrada	salida
0	0	1	1	salida	salida	entrada	entrada
0	1	0	0	salida	entrada	salida	salida
0	1	0	1	salida	entrada	salida	entrada
0	1	1	0	salida	entrada	entrada	salida
0	1	1	1	salida	entrada	entrada	entrada
1	0	0	0	entrada	salida	salida	salida
1	0	0	1	entrada	salida	salida	entrada
1	0	1	0	entrada	salida	entrada	salida
1	0	1	1	entrada	salida	entrada	entrada
1	1	0	0	entrada	entrada	salida	salida
1	1	0	1	entrada	entrada	salida	entrada
1	1	1	0	entrada	entrada	entrada	salida
1	1	1	1	entrada	entrada	entrada	entrada

Tabla (V.11).- Anzo 0 - Definición de ruidos.

4) El puerto de 4 bits es usado para el control y la indicación del Status - del puerto de datos de 8 bits.

En la fig (V.4) se muestra la configuración que presenta el 8255 cuando está - siendo utilizado en Modo 1 y como puerto de Entrada.

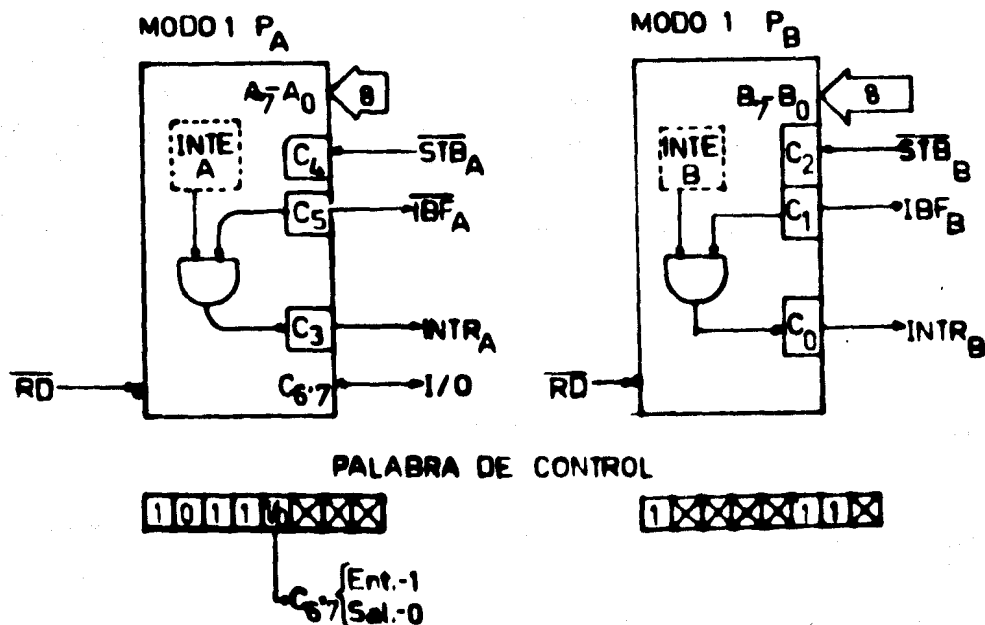


Fig (V.4).- 8255 en Modo 1 y Como Puerto de Entrada.

La entrada  $\overline{STB}$  (Strobe input) con un nivel bajo carga el dato en el Latch de entrada.

La salida  $\overline{IBF}$  (Input buffer full I/F) con un nivel alto indica que el dato - ha sido cargado en el Latch de entrada; en esencia, un reconocimiento de que el - Latch está disponible para un dato es indicado por la señal  $\overline{IBF}$  y es activo por el borde de caída de la entrada  $\overline{STB}$  y es inhibido por el borde de salida de la entrada  $\overline{RD}$ .

La salida  $\overline{INTR}$  se activa por el borde de subida de  $\overline{STB}$  si  $\overline{INT}$  es un "1" e  $\overline{INT}$  es "1" también. Se inhibe por el borde de bajada de  $\overline{RD}$ . Este procedimiento permite al dispositivo de entrada solicitar el servicio del CPU simplemente disparando su dato en el puerto.

Los flip-flops  $\overline{INTE A}$  e  $\overline{INTE B}$  (interrupt enable) pueden ser activados o inhibidos por medio de la operación bit set / reset.

$\overline{INTE A}$  es controlado por  $PC_4$ .

$\overline{INTE B}$  es controlado por  $PC_2$ .

La figura (IV.5) presenta la configuración del 8255 cuando está siendo utilizado en el Modo 1 y como puerto de salida.

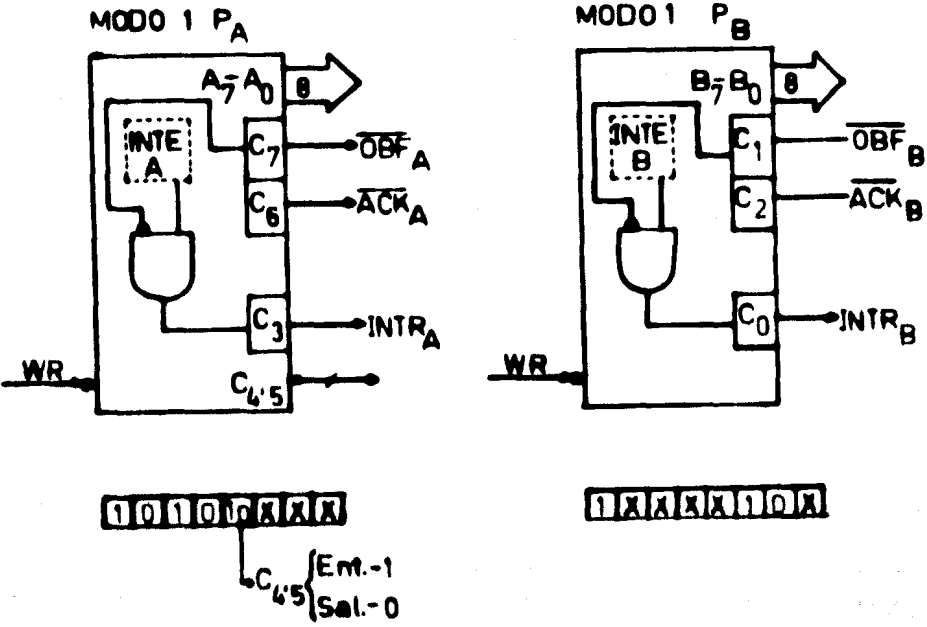


Fig (IV.5).- 8255 en Modo 1 y como Puerto de Salida.

La salida  $\overline{OBF}$  (output buffer full F/F) con un nivel bajo indica que el CPU se activa con el borde de subida de la entrada  $\overline{WR}$  y se inhibe con el borde de bajada de la señal  $\overline{ACK}$ .

La entrada  $\overline{ACK}$  (acknowledge input) con un nivel bajo informa al 8255 que el dato del puerto A o el puerto B ha sido aceptado. En esencia, es una respuesta del dispositivo periférico indicando que éste ha recibido el dato generado por el CPU.

La línea  $\overline{INTR}$  (interrupt request) con un nivel alto para interrumpir al CPU cuando un dispositivo de salida ha aceptado el dato transmitido por el CPU.  $\overline{INTR}$  es activado por el borde de subida de  $\overline{ACK}$  si  $\overline{OBF}$  es un "1" e  $\overline{INTR}$  es un "1" también. -  $\overline{INTR}$  es inhibido por el borde de bajada de  $\overline{WR}$ .

$\overline{INTR}$  A es controlado por la operación "bit out / read" de  $PC_0$ .

$\overline{INTR}$  B es controlado por la operación "bit out / read" de  $PC_2$ .

### C) Modo 2:

Esta configuración funcional proporciona un medio de comunicación con un dispositivo periférico en un bus único de 8 bits que puede tanto transmitir como recibir datos. Señales de disparo se proporcionan para mantener la disciplina de flujo de propiedad en el bus, de manera similar a la del modo 1. La generación de interrupción y la función de capacitante o no, también están disponibles.

La definición funcional para resumirlos de la siguiente manera:

- 1) Un controlador al grupo A.
- 2) Un puerto de comunicación bidireccional de 8 bits (Puerto A) y un puerto de control de 5 bits (Puerto C).
- 3) Tanto las entradas como las salidas con almacenamiento temporalmente en Lab. das.

4) El puerto de control de 5 bits (Puerto C) es usado para el control e información de status para el puerto bidireccional de 8 bits (Puerto A).

La figura (V.6) muestra la configuración que presenta el 8255 cuando está siendo utilizado en Modo 2.

La salida  $\overline{INTR}$  (interrupt request) con un nivel alto puede interrumpir al CPU tanto para la operación de entrada como de salida.

#### Operación de Salida:

La salida  $\overline{OBF}$  (output buffer full) con un nivel bajo, indicará que el CPU ha escrito el dato en el puerto A.

La entrada  $\overline{ACK}$  (acknowledge) con un nivel bajo permite la conexión del buffer de salida de 3 estados del puerto A para enviar el dato hacia afuera. De otro modo, el buffer de salida estará en su estado de alta impedancia.

El flip flop INTR 1 asociado con  $\overline{OBF}$  es controlado por la operación "bit set - reset" de  $\overline{PC_6}$ .

#### Operación de Entrada.

La entrada  $\overline{STB}$  (strobe input) con un nivel bajo carga el dato en el latch de entrada.

La salida  $\overline{IBF}$  (input buffer full I/B) con un nivel alto, indica que el dato ha sido cargado dentro del latch de entrada.

El flip - flop INTR 2 asociado con  $\overline{IBF}$  es controlado por la operación "bit set - reset" de  $\overline{PC_6}$ .

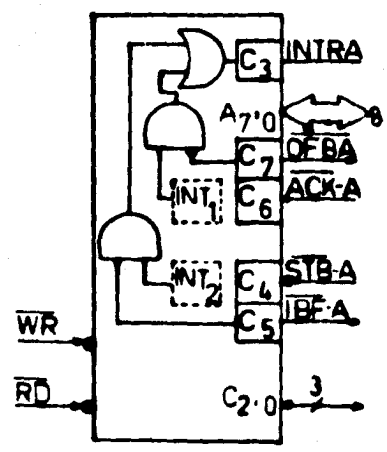
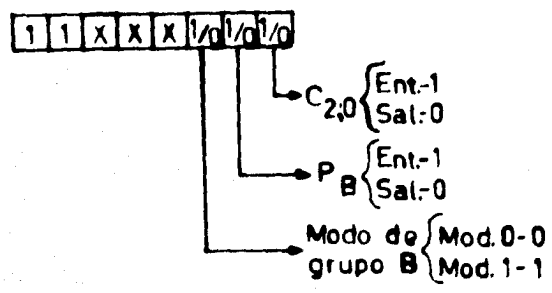


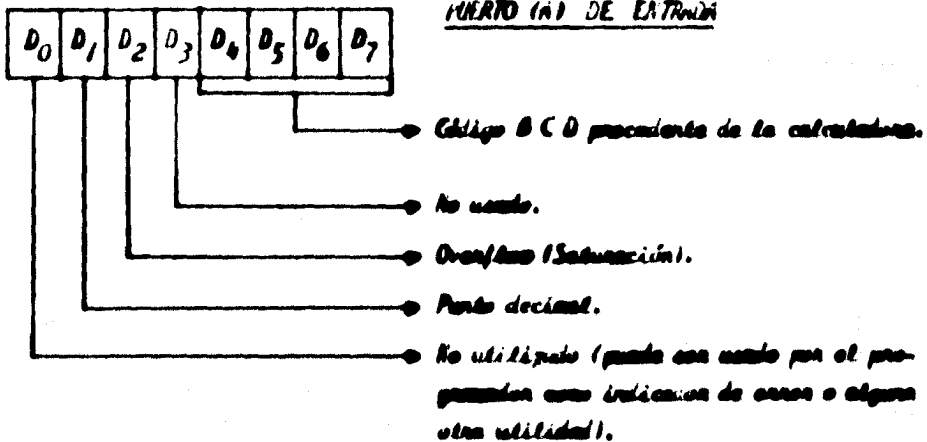
Fig (V.6)- 8255 en Modo "2".

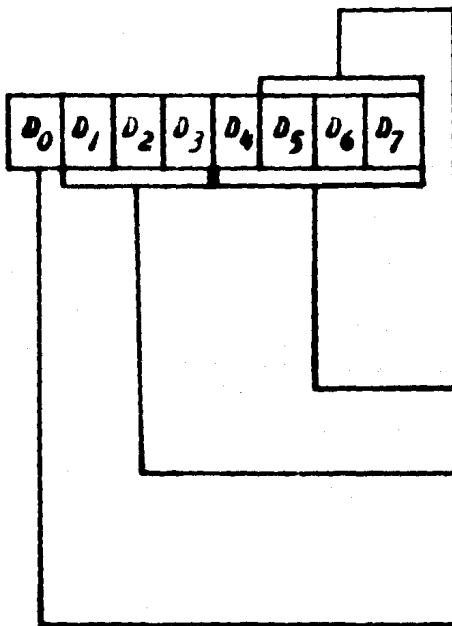
V.31.- DEFINICION DE LOS FORMATOS DE ENTRADA Y SALIDA.

Para lograr nuestros objetivos, utilizamos al P. P. 1. - 8255 en su modo 1 de operación, de esta manera (como ya se explicó) es posible definir al Puerto A como entrada de datos, o sea este puerto recibe la información que la calculadora envía hacia el C P U del Microprocesador, esta información corresponde al código BCD del dígito leído del Display, así mismo, el punto decimal e indicación de saturación en caso de existir.

También es posible definir el Puerto B, como puerto de salida, o sea, envía hacia la calculadora los datos provenientes del C P U. Dichos datos corresponden al código de la tecla que se desea "oprimir" o la dirección del dígito en caso de estar "leyendo" el display de la calculadora.

A continuación se describen los formatos de entrada y salida antes mencionados:





PUERTO (B) DE SALIDA

En caso de que se esté leyendo un dato de la calculadora, estos bits indican la dirección del dígito — que se lee.

Código B C D del dígito que se desea escribir en la calculadora.

Elección de cualquiera de las tres líneas comunes del teclado.

(1) Se lee un dato de la calculadora, direccionados por los bits 5, 6, 7.

(0) Se escribe un dato en la calculadora.

D<sub>1</sub> ————— Línea Común 15.

D<sub>2</sub> ————— Línea Común 16.

D<sub>3</sub> ————— Línea Común 17.



#### V.4).- CODIGO DE COMANDOS.

Para simular el teclado, el CPU del Microprocesador nos envía a través del puerto B del P. P. 1. (8255) un cierto código para cada una de las teclas.

Este código depende directamente del programador, en este caso lo formamos de la siguiente manera:

BITS	DIGITOS	OPERACIONES	FUNCIONES
4 5 6 7	Línea Común 17	Línea Común 16	Línea Común 15
0 0 0 0	2	-	0
0 0 0 1	1	•	C
0 0 1 0	3	R	.
0 0 1 1	4	+	CL
0 1 0 0	5	8	CB
0 1 0 1	6	√	(-)
0 1 1 0	7	=	LF
0 1 1 1	8	M	$\frac{0}{out}$
1 0 0 0	9	B	W

Tabla (V.2).- Selección de las Líneas de Comandos.

El código anterior es utilizado por el CPU para seleccionar las líneas de comandos del teclado. La selección de las líneas comunes se realiza por medio de los bits  $B_1$ ,  $B_2$  y  $B_3$ ; mientras el bit  $B_0$  "leemos" o "escribimos" en la calculadora. Por lo tanto, el Código de Comandos queda finalmente como sigue:

CODIGO DE CUBANOS

B <sub>0</sub>	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	TECLA
-	1	0	0	0	0	0	1	C
-	1	0	0	0	0	1	0	.
-	1	0	0	0	0	1	1	CL
-	1	0	0	0	1	0	0	CB
-	1	0	0	0	1	0	1	-
-	1	0	0	0	1	1	0	XY
-	1	0	0	0	1	1	1	FE
-	1	0	0	1	0	0	0	.
-	0	1	0	0	0	0	1	-
-	0	1	0	0	0	0	0	-
-	0	1	0	0	0	1	0	X
-	0	1	0	0	0	1	1	+
-	0	1	0	0	1	0	0	√
-	0	1	0	0	1	1	0	=
-	0	1	0	0	1	1	1	≠
-	0	1	0	1	0	0	0	≠
-	1	0	0	0	0	0	0	0
-	0	0	1	0	0	0	1	1
-	0	0	1	0	0	0	0	2
-	0	0	1	0	0	1	0	3
-	0	0	1	0	0	1	1	4
-	0	0	1	0	1	0	0	5
-	0	0	1	0	1	0	1	6
-	0	0	1	0	1	1	0	7
-	0	0	1	0	1	1	1	8
-	0	0	1	1	0	0	0	9

Tabla (V.3).

## CAPITULO VI

## DESCRIPCION DEL PROGRAMA

Como ya es sabido, se necesita desarrollar una rutina Software para lograr la transferencia de datos entre el Microprocesador y la Calculadora. Este pequeño programa, el cual será almacenado en una memoria ROM nos permitirá dirigir al CPU e inicializar al PPI en el modo de trabajo adecuado.

Por lo tanto, nuestro programa contendrá primordialmente las instrucciones para programar el 8255, las direcciones de memoria donde se encontrarán los datos que serán enviados a la calculadora, así como también las direcciones donde se almacenarán los resultados que la calculadora proporcionará.

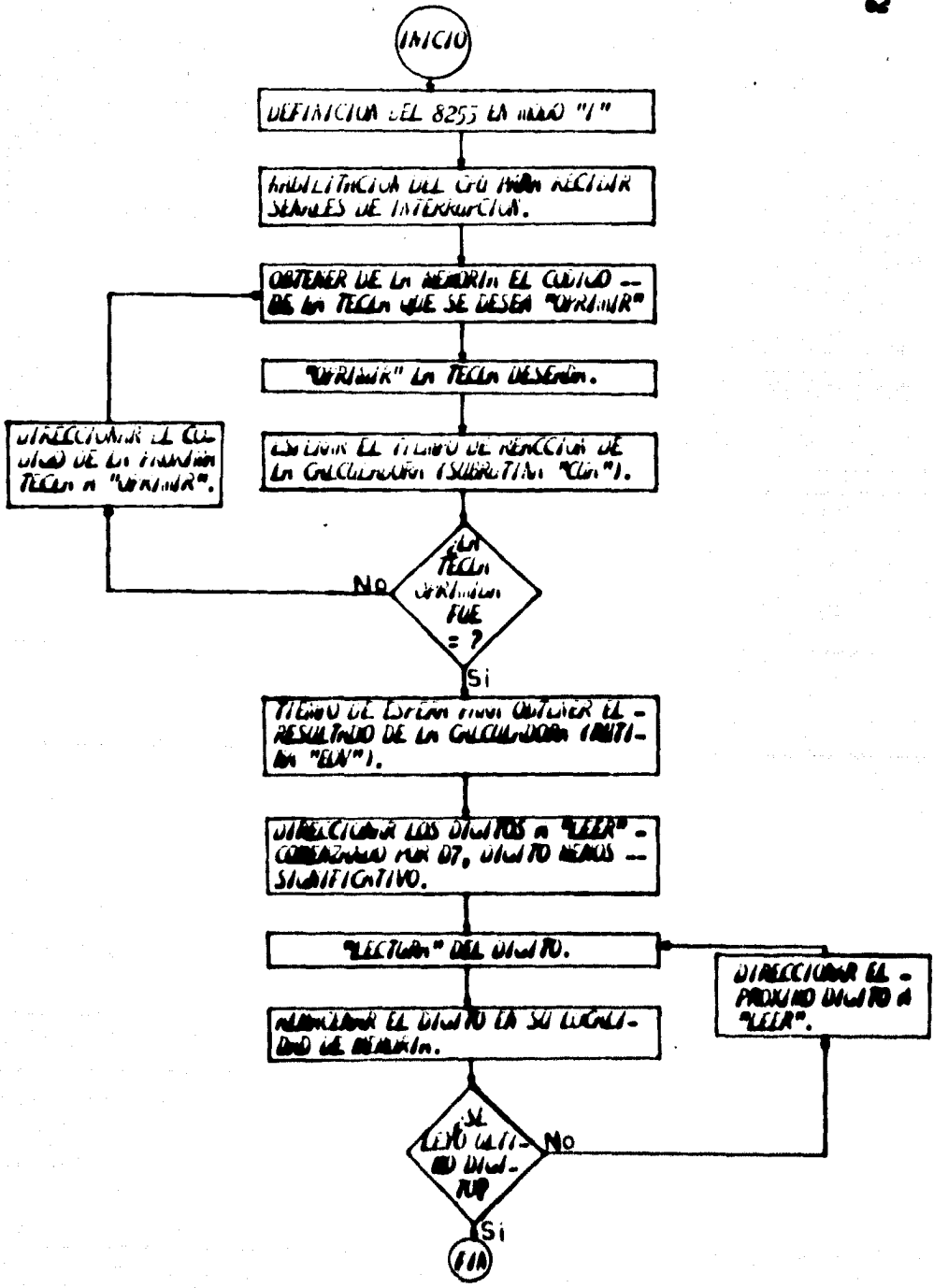
Para la descripción general del programa desarrollado utilizaremos el diagrama de flujo de la figura (VI.1); el cual está realizado para ejecutar las operaciones: Suma, Resta, Multiplicación y División exclusivamente.

En la página (70) se encuentra un listado del programa por instrucciones codificadas y con la dirección de memoria donde estarán almacenadas.

Al final del capítulo se anexa el listado del programa, así como su codificación, y también la simulación del mismo.

## VI.1) INSTRUCCIONES DEL PROGRAMA:

En continuación describiremos las instrucciones del programa. Para definir al P. P. I. (8255) en el modo "1" de operación utilizaremos las instrucciones cuyas direcciones de memoria son 1300H, 1301H, 1302H, 1303H y 1304H; esto es el modo en el cual, deberá trabajar haciendo como puente de entrada hacia el CPU al Puerto A -



y el B como punto de salida. Para esto se le da salida a la palabra de control — ODC (en código hexadecimal), asignándose el punto con la dirección 131 en la —  
 dirección ODC. Una vez finalizado esto, también se programa el 8255 para que no —  
 se rompan las series de interrupción (interrupción externa), con la palabra de —  
 control ODC, por lo tanto el bit Set / Reset (R<sub>2</sub>) del Puerto C será el que nos —  
 marque estas series.

La instrucción 130H carga en el registro par "0" del Cx el direccionamiento —  
 para que se envíen a la salida los datos que se envían a la calculadora y se la —  
 dirección 100H; para que la instrucción 130H incrementa el registro B a 100H; —  
 así es que en realidad la dirección de inicio donde deberá colocarse el primer da —  
 to o código de estado que será enviado a la calculadora es 100H.

La instrucción 130H carga en el acumulador el dato localizado en la direc —  
 ción de memoria 100H, que es el primer código de control que involucra el uso de —  
 las calculadoras. La instrucción 130H le da salida a este código de control que es el —  
 Puerto B del P. P. 1. para que "opere" la serie deseada.

La instrucción 131H realiza la Subrutina de Laplace "Dn" para que la calcula —  
 dos números a la operación de la serie. Para no perder el control exterior durante —  
 de la Subrutina de Laplace, dato es almacenado en el registro B del Cx por lo que —  
 dirección 1310.

La Subrutina de Laplace es necesaria, dado que en condiciones normales se en —  
 encuentran los algoritmos siempre de "recursión" para cada una de las series.

T E C N I C O S

+	0.12	.....
X	0.10	.....
+	0.05	.....
-	0.05	.....
√	0.16	.....
8 (Octal)	0.08	.....
8	0.12	.....
X	0.06	.....

La instrucción 1314 repasa hacia el acumulador el contenido ordenadamente el-

segundo en el registro D del C.U. Posteriormente la instrucción 1315 copia el-  
 to código de comando con el código de la tecla "igual" (27H = 00100111); la copia  
 acción se hace para saber si se ha formado de "operación" las teclas necesarias -  
 para realizar un cálculo determinado.

La instrucción 1314 copia en el registro par D del C.U. la dirección 1310H; -  
 en cada dirección se insiere el resultado obtenido de la calculadora  
 "registro" cada uno de los dígitos del Display de decimales e izquierda, utilizando  
 la misma clave de memoria AFD para almacenar estos dígitos.

La instrucción 1310 copia el acumulador con el dato que direccionará el pro-  
 ceso dígito a ser leído (que será el de la entrada derecha del Display), este da-  
 to cambia en el acumulador en  $1000 (1000)_{10}$  y el cual entra por las líneas  
 P<sub>7</sub> P<sub>6</sub> P<sub>5</sub> P<sub>4</sub> P<sub>3</sub> P<sub>2</sub> P<sub>1</sub> y P<sub>0</sub> (entradas del Circuito de Interfases) a las líneas  
 máximas P<sub>7</sub> P<sub>6</sub> P<sub>5</sub> P<sub>4</sub> P<sub>3</sub> P<sub>2</sub> P<sub>1</sub> P<sub>0</sub> del 274151, posteriormente con una dirección par e desde del -  
 255 punto B.

Las direcciones que envía el CPU para efectuar la lectura de cada uno de los dígitos de la calculadora, son las que se muestran a continuación:

DIRECCION	DIGITO DE LA CALCULADORA.
1000 0000 .....	D <sub>1</sub>
" 0001 .....	D <sub>2</sub>
" 0010 .....	D <sub>3</sub>
" 0011 .....	D <sub>4</sub>
" 0100 .....	D <sub>5</sub>
" 0101 .....	D <sub>6</sub>
" 0110 .....	D <sub>7</sub>
" 0111 .....	D <sub>8</sub>

Este bit indica que es una dirección para efectuar la lectura de los dígitos

Los terminales D<sub>0</sub>, D<sub>1</sub>, D<sub>2</sub>, D<sub>3</sub>, D<sub>4</sub>, D<sub>5</sub>, D<sub>6</sub> y D<sub>7</sub> (Circuito de Interfase) en el 8874151 tiene conectadas los terminales del pulso de borrado del Display de la calculadora e indican el dígito que es borrado de acuerdo a la dirección enviada por el CPU, como se mencionó antes.

La instrucción 1320 da salida a la dirección del dígito que se va a borrar y - en direcciones con 11H por que es el dato que sale por el puerto B.

La instrucción 1322 detiene al CPU hasta que se obtenga el dato del dígito - borrado en el Puerto A del 8255. El CPU sale del estado de alto (hi.L) al detectar al P. P. I. una señal de interrupción proporcionada por la Interfase.

La instrucción 1323 permite la entrada al dato del dígito borrado a través del puerto A del P. P. I. (8255), por eso es direcciones con 10H.

La instrucción 1325 almacena el dato del dígito leído en la localidad de memoria indicada por el registro D en el CU, y es la 1380H. Esta dirección deberá incrementarse para guardar los dígitos siguientes en las localidades de memoria subsiguientes (1381H, 1382H, ... , 1387H); hasta leer y almacenar los 8 dígitos, el incremento en las direcciones de los dígitos a leer se realiza por medio de la instrucción 1326.

La instrucción 1327 transfiere lo que contiene el registro H hacia el Acumulador, y como el registro H tenía en 80H que fue la dirección del primer dígito leído (entrada derecha), éste se incrementará con la instrucción 132D para direccionar al siguiente dígito y se comparará con 87H para determinar si se han leído los 8 dígitos (instrucción 132B), en caso de que aún no se haya leído el octavo dígito, se direcciona el próximo para leerlo y almacenarlo, y así hasta leer los 8 dígitos, la rutina para realizar éste se inicia en la instrucción 131F y finaliza en la instrucción 132E.

La subrutina "Uti" es de espera para dar tiempo a que la calculadora reaccione. En esta rutina se realizan los lapsos de espera de la siguiente forma:

001 n, 77H	-----	INSTRUCCIÓN 7 CICLOS.
002 n	-----	" 5 "
003 2E	-----	" 10 "

Estos lapsos como se ve en el programa ocupan desde la instrucción 1322 hasta 133C, en donde se realizan 84 lapsos de espera (54H = 84) que van desde la palabra 007H y se documentan a como 007H = 255). Todos estos lapsos de espera total de aproximadamente 0.160 seg., son necesarios, puesto que el tiempo de reacción de nuestras calculadoras es bastante grande.



## VI.2) ENSEMBLE Y SIMULACION:

Nuestro programa de interfase debemos almacenarlo en Memoria, para ello necesitamos codificarlo en Lenguaje de Máquina. Esto se logra mediante un programa Ensamblador.

El Programa Ensamblador que nos transforma nuestros "programa fuente" al Lenguaje de Máquina o "programa objeto", es el programa llamado "MNC - 80" (Micro Ensamblador del Microprocesador Intel 8080). Este programa requiere un Procesador con una longitud de palabra de 32 bits o mayor, así como un compilador standard - Fortran, y un mínimo de 2 archivos de entrada secuenciales y 4 de salida. Normalmente este programa requiere al menos de 11 Kbytes de Memoria.

El código ensamblado, obtenido por medio del programa "MNC - 80", se encuentra en el listado del Programa de Interfase iniciándose con la instrucción:

E MNC/80; FILE.FILE20 o TRJLETm

Proporcionaremos una tabla final del Código de Máquina.

Nuestro programa de interfase fue simulado para observar el desarrollo del mismo antes de almacenarlo en Memoria, mediante un programa simulador escrito en FORTRAN IV, llamado INTER/80. Este programa proporciona una simulación en Software del CPU del Microprocesador Intel 8080, junto con la ejecución de los comandos apropiados que nos ayudan a observar paso a paso el desarrollo del programa.

El INTER/80 acepta el código de máquina producido por el Ensamblador MNC-80, junto con la ejecución de los comandos desde una terminal de tiempo compartido. - Los comandos de ejecución nos permiten la manipulación de la Memoria simulada del sistema MNC-80 y los registros del CPU del 8080. Además operando instrucciones de

nuestra puede detenerse la ejecución del programa en puntos cruciales. Disponemos también de características de trap, las cuales permiten monitorizar la operación del CPU. El INTER/80 proporciona también referencias simbólicas para localidades de almacenamiento, tanto como referencias numéricas en diferentes bases.

Nuestro programa fue simulado en la terminal de tiempo compartido (Teletipo) de la DESFI. La simulación del Programa de Interfase se encuentra en el listado - del mismo conorgado con la instrucción:

E SIMULADOR/8080

En esta simulación, se modificó el tiempo de espera de la subrutina "CPU", - por limitación de tiempo y papel y por razones que no es necesario, en la observación del desarrollo de la simulación básica del programa. Las instrucciones modificadas fueron:

1332 CPU: MVI L, 9AH

1332 CPU: MVI L, 02H

1334 MD: MVI A, 07FH

1334 MD: MVI A, 03H

ruptura puede detenerse la ejecución del programa en puntos cruciales. Disponemos también de características de trap, las cuales permiten monitorizar la operación del CPU. El INTERP/80 proporciona también referencias simbólicas para localidades de almacenamiento, tanto como referencias numéricas en diferentes bases.

Nuestro programa fue simulado en la terminal de tiempo compartido (Teletipo) de la DESI. La simulación del Programa de Interface se encuentra en el listado - del mismo comenzando con la instrucción:

**E BSIMULADOR/8080**

En esta simulación, se modificó el tiempo de espera de la subrutina "CIA", - por limitación de tiempo y papel y por pensar que no es necesario, en la observación del desarrollo de la simulación básica del programa. Las instrucciones modificadas fueron:

1332 Cln: MVI E, 5AH

1332 Cln: MVI E, 02H

1334 MD: MVI A, 0FFH

1334 MD: MVI A, 03H

UNAM B6700:126 CANDE 28.1 UNAMI [09011401] YOU ARE TTYH016(24)  
ENTER USERCODE PLEASE

ES81  
ENTER PASSWORD PLEASE.

SESSION 7444 17:11:21 04/03/78

FILES  
(ES81) ON PACK

- . MAC
- . . 11 : FORTRAN
- . SLIC
- . . SLIC : FORTRANCODE
- . FILE7 : DATA
- . TACHO : JOBSYMBOL
- . OBJECT
- . . MAC
- . . . 80 : FORTRANCODE
- . TRACON : FORTRANCODE
- . TARJETA : DATA
- . SIMULADOR
- . . 8080 : FORTRANCODE

E MAC/800DEL  
E MAC/80IFILE FILE20-TARJETA  
BRUNING 7426  
BT

8080 MACRO ASSEMBLER, VER 2.1  
88

- 01-0 05-1
- 0C-0 0D-0
- 0E-0 0F-0
- 0I-1 0L-1
- 0M-0 0O-1
- 0P-1 0Q-0
- 0R-72 0S-1
- 0T-0 0U-72
- 07406 INU INDEX @ 014:00FC:08
- 0F-08 @ 00257200, 00000100, 040:0000:4.
- 0ET-20.0 F1-1.1 10-0.2

E MAC/80#FILE FILE20-TARJETA  
BRUNNING 7470  
BT

70

8080 MACRO ASSEMBLER, VER 2.1  
BT-1 BT-2

BEGIN

8080 MACRO ASSEMBLER, VER 2.1 ERRORS - 0 PAGE 1

1300		ORG	1300H
1300	3E5C	ORH:	HVI A,05CH
1302	D313		OUT 13H
1304	3E09		HVI A,09H
1306	D313		OUT 13H
1308	FB		EI
1309	010010		LXI B, 1000H
130C	03	FJA:	INX B
130D	0A		LDAX B
130E	D311		OUT 11H
1310	57		MOV D,A
1311	CD3213		CALL CUA
1314	7A		MOV A,D
1315	FE27		CFI 27H
1317	C20C13		JNZ FJA
131A	118013		LXI D,1380H
131D	3E5C		HVI A,5CH
131F	67	EDV:	MOV H,A
1320	D311		OUT 11H
1322	76		HLT
1323	DB1C		IN 1CH
1325	12		STAX D
1326	13		INX D
1327	7C		MOV A,H
1328	FE27		CFI 27H
132A	CA3113		JZ ROH
132D	3C		INR A
132E	CB1F13		JMF EDV
1331	76	ROH:	HLT
1332	1E02	CUA:	HVI E,02H (E, 02H)
1334	3E03	XDI:	HVI A,03H (A, 03FH)
1336	3D	XE:	DCR A
1337	C23613		JNZ XE
133A	7B		MOV A,E
133B	1D		DCR E
133C	C23413		JNZ XD
133F	09		REI
			END

NO PROGRAM ERRORS

6080 MACRO ASSEMBLER, VER 2.1 ERRORS - 0 PAGE 2

## SYMBOL TABLE

\* 01

A	0007	B	0000	C	0001	CUA	1332
D	0002	E	0003	EDV	131F	FJA	130C
GHR	1300 *	H	0004	L	0005	N	0006
FSM	0006	ROM	1331	SF	0006	XD	1334
XE	1336						

SET-1:37.7 FT-6.3 ID-2.4

•

FIDEL

L FILE22

\*FILE (ESS1)FILE21 ON PACK

```

100 001 BLOCK01 0
200 002 A      0000070
300 003 B      0000000
400 004 C      0000010
500 005 CUA    0114620
600 006 D      0000020
700 007 E      0000030
800 008 EDV    0114370
900 009 FJA    0114140
1000 010 GHR   0114000
1100 011 H     0000040
1200 012 L     0000050
1300 013 N     0000060
1400 014 FSM   0000060
1500 015 ROM   0114610
1600 016 SF    0000060
1700 017 XD    0114640
1800 018 XE    0114660
1900 •

```

```

2000 :101300003EBCD3133E09D313FB010010030AD311D3
2100 :1013100057CD32137AFE27C20C131180133E60671B
2200 :10132000831176DB1012137DFE670A31133CC31F26
2300 :1013300013761E023E033DC236137B1DC23413C911
2400 :00000000000
2500 •

```

•  
E S SIMULADOR/8080  
BRUNING 7467  
BT

INTERF/80 VERS 1.0  
BF-2  
LOAD 7.

64 LOAD OK

S FC1300H. DEL  
S FC-1300H. S SF-13FCH. S MEMORY 1001H-13H 23H 14H 27H.

SET OK  
SET OK  
SET OK

INFUT 10H. OUTFUT 13H. 11H.

INFUT OK  
OUTFUT OK  
TRACE 0 TO 10000.  
00.

TTIME

05:22 PM MONDAY, APRIL 3, 1978

TRACE OK

CYZBP	A	B	C	D	E	H	L	HL	SF	FC
*0000*	*000*	*000*	*000*	*000*	*000*	*000*	*000*	*00000*	05104*	404864
HVI A 188										
0000*	188	000	000	000	000	000	000	00000	05104*	404864
OUT 17										
FORT 17 - 188										
0000	188	000	000	000	000	000	000	00000	05104*	404868
HVI A 9										
0000*	009	000	000	000	000	000	000	00000	05104*	404870
OUT 17										
FORT 17 - 9										
0000	009	000	000	000	000	000	000	00000	05104*	404872
EI										
0000	009	000	000	000	000	000	000	00000	05104*	404873
LXI B 4096										
0000	009*	016	001	000	000	000	000	00000	05104*	404876
INX B										
0000	009	016*	001	000	000	000	000	00000	05104*	404877
LDAX B										
0000*	019	016	001	000	000	000	000	00000	05104*	404878
OUT 17										
FORT 17 - 19										
0000	019	016	001	000	000	000	000	00000	05104*	404880
MOV DA										
CYZBP	A	B	C	D	E	H	L	HL	SF	FC
0000	019	016	001*	019	000	000	000	00000	05104*	404881
CALL 4914										
0000	019	016	001	019	000	000	000	00000*	05102*	404914
HVI E 2										
0000	019	016	001	019*	002	000	000	00000	05102*	404916
HVI A 3										
0000*	003	016	001	019	002	000	000	00000	05102*	404918
DCR A										
0000*	002	016	001	019	002	000	000	00000	05102*	404919
JNZ 4918										
0000	002	016	001	019	002	000	000	00000	05102*	404918
DCR A										
0000*	001	016	001	019	002	000	000	00000	05102*	404919
JNZ 4918										
0000	001	016	001	019	002	000	000	00000	05102*	404918
DCR A										
00101*	000	016	001	019	002	000	000	00000	05102*	404919
JNZ 4918										
0101	000	016	001	019	002	000	000	00000	05102*	404922
MOV AE										
CYZBP	A	B	C	D	E	H	L	HL	SF	FC
0101*	002	016	001	019	002	000	000	00000	05102*	404923
DCR E										
0000	002	016	001	019*	001	000	000	00000	05102*	404924
JNZ 4914										
0000	002	016	001	019	001	000	000	00000	05102*	404916
HVI A 3										
0000*	003	016	001	019	001	000	000	00000	05102*	404918









MVI E 2  
 1011 039 016 004 039\*002 000 000 00000 05102\*04916  
 MVI A 3  
 1011\*003 016 004 039 002 000 000 00000 05102\*04916  
 DCR A  
 \*1000\*002 016 004 039 002 000 000 00000 05102\*04919  
 JNZ 4918  
 1000 002 016 004 039 002 000 000 00000 05102\*04918  
 DCR A  
 1000\*001 016 004 039 002 000 000 00000 05102\*04919  
 JNZ 4918  
 1000 001 016 004 039 002 000 000 00000 05102\*04918  
 DCR A  
 \*1101\*000 016 004 039 002 000 000 00000 05102\*04919  
 JNZ 4918  
 1101 000 016 004 039 002 000 000 00000 05102\*04922  
 MOV AE  
 CY2BF A B C D E H L HL SF FC  
 1101\*002 016 004 039 002 000 000 00000 05102\*04923  
 DCR E  
 \*1000 002 016 004 039\*001 000 000 00000 05102\*04924  
 JNZ 4918  
 1000 002 016 004 039 001 000 000 00000 05102\*04918  
 MVI A 3  
 1000\*003 016 004 039 001 000 000 00000 05102\*04918  
 DCR A  
 1000\*002 016 004 039 001 000 000 00000 05102\*04919  
 JNZ 4918  
 1000 002 016 004 039 001 000 000 00000 05102\*04918  
 DCR A  
 1000\*001 016 004 039 001 000 000 00000 05102\*04919  
 JNZ 4918  
 1000 001 016 004 039 001 000 000 00000 05102\*04918  
 DCR A  
 \*1101\*000 016 004 039 001 000 000 00000 05102\*04919  
 JNZ 4918  
 1101 000 016 004 039 001 000 000 00000 05102\*04922  
 MOV AE  
 CY2BF A B C D E H L HL SF FC  
 1101\*001 016 004 039 001 000 000 00000 05102\*04923  
 DCR E  
 1101 001 016 004 039\*000 000 000 00000 05102\*04924  
 JNZ 4918  
 1101 001 016 004 039 000 000 000 00000 05102\*04927

RET

1101 001 016 004 039 000 000 000 00000005104804884

MOV AD

1101039 016 004 039 000 000 000 000000 05104804885

CFI 39

00101 039 016 004 039 000 000 000 000000 05104804887

JWZ 4876

0101 039 016 004 039 000 000 000 000000 05104804890

LXI B 4992

0101 039 016 0040170128 000 000 000000 05104804893

HUI A 128

01010128 016 004 017 128 000 000 000000 05104804895

MOV MA

0101 128 016 004 017 1280128 000032768 05104804896

OUT 17

PORT 17 - 128

CYZSF A B C D E H L HL SF FC

0101 128 016 004 017 128 128 000 32768 05104804898

HLT

HLT CYCLE 1001

TIME

.

TIME-1001 (500.5 USEC)

INTER.

INTER OK

GO.

0101 128 016 004 019 128 128 000 32768 05104804899  
 IN 16  
 FORT 16...  
 02

01018002 016 004 019 128 128 000 32768 05104804901  
 HQF

0101 002 016 004 019 128 128 000 32768 05104 04901  
 STAX D

0101 002 016 004 019 128 128 000 32768 05104804902  
 INX D

0101 002 016 004 0198129 128 000 32768 05104804903  
 NOV AN

01018128 016 004 019 129 128 000 32768 05104804904  
 CFI 130

01011 128 016 004 019 129 128 000 32768 05104804906  
 JZ 4913

1011 128 016 004 019 129 128 000 32768 05104804909  
 INX A

10118129 016 004 019 129 128 000 32768 05104804910  
 JMF 4895

CY2SF A B C D E H L ML SF FC  
 1011 129 016 004 019 129 128 000 32768 05104804895

NOV MA  
 1011 129 016 004 019 1298129 000833024 05104804896

OUT 17  
 FORT 17 - 129

1011 129 016 004 019 129 129 000 33024 05104804878

MLT  
 MLT CYCLE 1080  
 TIME.

TIME-1080 (540 USED)

INTER.

INTER OK  
GO.

1011 129 016 004 019 129 129 000 33024 05104804879  
IN 16  
FORT 16...  
01

1011001 016 004 019 129 129 000 33024 05104804901

STAX B

1011 001 016 004 019 129 129 000 33024 05104804902

INX B

1011 001 016 004 019 130 129 000 33024 05104804903

NOU AH

10110129 016 004 019 130 129 000 33024 05104804904

CFI 135

1011 129 016 004 019 130 129 000 33024 05104804906

JZ 4913

1011 129 016 004 019 130 129 000 33024 05104804909

IWR A

CY28F A B C D E H L HL BF FC  
10110130 016 004 019 130 129 000 33024 05104804910

JMF 4875

1011 130 016 004 019 130 129 000 33024 05104804875

NOU HA

1011 130 016 004 019 130 130 000 33280 05104804876

OUT 17

FORT 17 - 130

1011 130 016 004 019 130 130 000 33280 05104804878

HLT

HLT CYCLE 1155

TIME.

TIME-1155 (577.5 USEC)

INTER.

INTER OK  
00.

1011 130 016 004 017 130 130 000 33280 05104404877  
IN 16  
FORT 16...  
0

10118000 016 004 017 130 130 000 33280 05104404701

STAX D

1011 000 016 004 017 130 130 000 33280 05104404702

INX D

1011 000 016 004 0178131 130 000 33280 05104404703

MOV AM

10118130 016 004 017 131 130 000 33280 05104404704

CFI 135

81010 130 016 004 017 131 130 000 33280 05104404706

JZ 4913

CY2SF A B C D E H L HL SF FC

1010 130 016 004 017 131 130 000 33280 05104404909

INA A

10108131 016 004 017 131 130 000 33280 05104404910

JWF 4875

1010 131 016 004 017 131 130 000 33280 05104404895

MOV MA

1010 131 016 004 017 1318131 000833536 05104404896

OUT 17

FORT 17 - 131

1010 131 016 004 017 131 131 000 33536 05104404898

HLT

HLT CYCLE 1230

TIME.

TIME-1230 (615 USEC)





INTER.

INTER OK  
OO.

1011 132 016 004 017 132 132 000 33792 05104804897  
IN 16  
FORT 16...  
C

1011000 016 004 017 132 132 000 33792 05104804901  
STAX D

1011 000 016 004 017 132 132 000 33792 05104804902

INX D

1011 000 016 004 017\*133 132 000 33792 05104804903

NOV AN

CY28F A B C D E H L HL SF FC

10110132 016 004 017 133 132 000 33792 05104804904

CFI 135

1010 132 016 004 017 133 132 000 33792 05104804906

JZ 4913

1010 132 016 004 017 133 132 000 33792 05104804909

IWR A

10100133 016 004 017 133 132 000 33792 05104804910

JWF 4895

1010 133 016 004 017 133 132 000 33792 05104804895

NOV MA

1010 133 016 004 017 133\*133 000\*34048 05104804896

GUT 17

FORT 17 - 133

1010 133 016 004 017 133 133 000 34048 05104804898

MLT

MLT CYCLE 1380

TIME.

TIME-1380 (490 USEC)

INTER.

INTER OK  
GO.

1010 133 016 004 017 133 133 000 34048 05104804877  
IN 16  
PORT 16...  
0

10100000 016 004 017 133 133 000 34048 05104804901  
STAX D

1010 000 016 004 017 133 133 000 34048 05104804902  
INX D

CYZBF A B C D E H L HL SF FC  
1010 000 016 004 017 134 133 000 34048 05104804903  
NDU AH

10100133 016 004 017 134 133 000 34048 05104804904  
CPI 135

1010 133 016 004 017 134 133 000 34048 05104804906  
JZ 4913

1010 133 016 004 017 134 133 000 34048 05104804909  
INR A

10100134 016 004 017 134 133 000 34048 05104804910  
JWF 4675

1010 134 016 004 017 134 133 000 34048 05104804895  
NDU MA

1010 134 016 004 017 134 134 000 34304 05104804896  
OUT 17

PORT 17 - 134  
1010 134 016 004 017 134 134 000 34304 05104804898

MAT  
MAT CYCLE 1455  
TIME.

TIME-1455 (727.5 USEC)

INTER.

INTER OK  
GO.

1010 133 016 004 019 133 133 000 34048 05104804899  
IM 16  
FORT 16...  
C

10108000 016 004 019 133 133 000 34048 05104804901  
STAX D

1010 000 016 004 019 133 133 000 34048 05104804902

INX D  
CYZBF A B C D E H L ML SF FC

1010 000 016 004 0198134 133 000 34048 05104804903

NOV AH

10108133 016 004 019 134 133 000 34048 05104804904

CPI 135

1010 133 016 004 019 134 133 000 34048 05104804906

JZ 4913

1010 133 016 004 019 134 133 000 34048 05104804909

INR A

10108134 016 004 019 134 133 000 34048 05104804910

JMF 4895

1010 134 016 004 019 134 133 000 34048 05104804895

NOV NA

1010 134 016 004 019 1348134 000834304 05104804896

OUT 17

FORT 17 - 134

1010 134 016 004 019 134 134 000 34304 05104804898

MLT

MLT CYCLE 1455

TIME.

TIME-1455 (727.5 USEC)

INTER.

INTER OR  
DU.

1010 134 016 004 017 134 134 000 34304 05104804877  
IN 16  
FORT 16...  
0

1010000 016 004 017 134 134 000 34304 05104804901  
STAX D  
CYZSF A B C D E H L HL SF FC  
1010 000 016 004 017 134 134 000 34304 05104804902  
INX D  
1010 000 016 004 017 135 134 000 34304 05104804903  
NOU AH  
10100134 016 004 017 135 134 000 34304 05104804904  
CFI 135  
1011 134 016 004 017 135 134 000 34304 05104804906  
JZ 4913  
1011 134 016 004 017 135 134 000 34304 05104804907  
INR A  
10110135 016 004 017 135 134 000 34304 05104804910  
JWF 4895  
1011 135 016 004 017 135 134 000 34304 05104804875  
NOU MA  
1011 135 016 004 017 135 135 000 34304 05104804876  
OUT 17  
FORT 17 - 135  
1011 135 016 004 017 135 135 000 34304 05104804878  
MLT  
MLT CYCLE 1030

ENTER.

ENTER OK  
G.

1011 135 016 004 019 135 135 000 34560 05104804897  
PORT 16...

ZSP	A	B	C	D	E	H	L	HL	SF	FC
1011	000	016	004	019	135	135	000	34560	05104804901	
TAX D										
1011	000	016	004	019	135	135	000	34560	05104804902	
TAX D										
1011	000	016	004	019	136	135	000	34560	05104804903	
OV AM										
1011	135	016	004	019	136	135	000	34560	05104804904	
FI 135										
0101	135	016	004	019	136	135	000	34560	05104804906	
Z 4713										
0101	135	016	004	019	136	135	000	34560	05104804913	

LT  
LT CYCLE 1575  
TIME

TIME-1575 (787.5 USEC)

DB

7487 OPERATOR USED @ 21E:0410:48  
0-DB @ 40237000, 41545000, 00069200, 00050200, 00103500, 00029400, 03A:0000:1.  
ET-18:20.2 FT-06.8 10-0.4

8  
E 8 SIMULADOR/8080  
8RUNNING 7666  
8?

INTERF/80 VERS 1.0  
8F-2  
LOAD 7.

64 LOAD OK

8 FC-1300H. 8 SF-13FOH. 8 MEMORY 1001H- 13H 23H 14H 27H.

SET OK  
SET OK  
SET OK  
INPUT 10H. OUTPUT 11H, 13H.

INPUT OK  
OUTPUT OK  
GO.

PORT 19 - 108  
PORT 19 - 9  
PORT 17 - 19  
PORT 17 - 35  
PORT 17 - 20  
PORT 17 - 37  
PORT 17 - 128  
HLT CYCLE 1001  
TIME.

TIME-1001 (500.3 USEC)  
INTER.

INTER OK  
GO.

PORT 16...  
02H

PORT 17 - 127  
HLT CYCLE 1080  
TIME.

TIME-1080 (540 USEC)

INTER.

INTER OK  
GO.

PORT 16...  
GIN

PORT 17 - 130  
MLT CYCLE 1155  
TIME.

TIME-1155 (577.5 USEC)  
INTER.

INTER OK  
GO.

PORT 16...  
GOH

PORT 17 - 131  
MLT CYCLE 1230  
TIME.

TIME-1230 (615 USEC)  
INTER.

INTER OK  
GO.

PORT 16...  
GOH

PORT 17 - 132  
MLT CYCLE 1305  
TIME.

TIME-1305 (652.5 USEC)



INTER.

INTER OK  
OO.

FORT 16...  
OOH

FORT 17 - 133  
HLT CYCLE 1380  
TIME.

TIME-1380 (690 USEC)  
INTER.

INTER OK  
OO.

FORT 16...  
OO

FORT 17 - 134  
HLT CYCLE 1400  
TIME.

TIME-1400 (727.5 USEC)  
INTER.

INTER OK  
OO.

FORT 16...  
OOH

FORT 17 - 135  
HLT CYCLE 1530  
TIME.

TIME-1530 (765 USEC)  
INTER.

INTER OK  
OO.

FORT 16...  
OOH

HLT CYCLE 1575  
TIME.

TIME-1575 (787.5 USEC)

## APENDICE A

### UNIDAD CENTRAL DE PROCESAMIENTO 8080

El 8080 es una unidad central de procesamiento (CPU) completa de 8 bits en paralelo para ser utilizado en sistemas digitales de computación de uso general. Está fabricado como un único chip con tecnología  $N^2$ MOS. El 8080 transfiere datos e información de su estado interno a través de un bus de datos bidireccional de tres estados ( $D_0$  a  $D_7$ ). Las direcciones de Memoria y dispositivos periféricos son transmitidas a través de un bus de direcciones de 16 bits en paralelo y de tres estados ( $A_0$  a  $A_{15}$ ). Seis salidas de control y tiempo (SYNC,  $\overline{MEMA}$ ,  $\overline{MEMT}$ ,  $\overline{MEMR}$ ,  $\overline{MEMW}$  e  $\overline{IATE}$ ) emanan del 8080, siendo de cuatro entradas de control ( $\overline{RSTW}$ ,  $\overline{HOLD}$ ,  $\overline{INT}$  y  $\overline{RES}$ ), cuatro entradas de polarización ( $+12V$ ,  $5V$ ,  $-5V$  y tierra), y dos entradas de reloj  $\phi_1$  y  $\phi_2$  en recibidas por el 8080.

#### A.1) ARQUITECTURA DEL CPU-8080:

El 8080 consiste de las siguientes unidades funcionales:

- Un arreglo de registros y lógica de direcciones.
- Una Unidad Aritmética y Lógica (ALU).
- Registros de instrucciones y estado de control.
- Buffer de bus de datos de tres estados bidireccional.

La figura (A.1) ilustra los bloques funcionales dentro del CPU - 8080

a.1a) ALU/STAD.

La sección de registros consiste de un arreglo de  $R \times N \times M$  celdas organizadas en seis registros de 16 bits.

- Contador de programa (PC).



- Stack pointer (S P).
- Seis registros de uso general arreglados en pares, conocidos como B, C, D, E, H, L.
- Un registro par temporal llamado W, Z.

El P C siempre mantiene la dirección de la próxima instrucción a ejecutar. El Stack Pointer contiene la dirección de la próxima localidad del Stack disponible en la memoria. El stack pointer puede ser inicializado para utilizar cualquier posición de memoria para escribir y leer en ella. El stack pointer es decrementado cuando un dato es introducido en el stack e incrementado cuando el dato es sacado del stack.

Los seis registros de propósito general pueden ser utilizados como registros individuales de 8 bits o como registros pares de 16 bits. El registro par temporal (W, Z) no es direccionable por el programador y es solo usado para la ejecución interna de las instrucciones.

Bytes de datos pueden ser transmitidos entre el bus interno y el arreglo de registros a través del Multiplexor selector de registros. Transferencias de 16 bits pueden tener lugar en el arreglo de registros y el Latch de direcciones o el circuito incrementación / decrementación. El Latch de direcciones o el circuito reciben datos de cualquiera de los tres registros pares y conduce la salida de los buses de dirección ( $A_0 - A_{15}$ ), así como también, el circuito incrementación / decrementación, este circuito recibe datos del Latch de direcciones y los envía al arreglo de registros. Los buses de 16 bits pueden ser incrementados o decrementados o simplemente ser transferidos de un registro a otro.

11.161 UNIDAD ARITMÉTICA Y LÓGICA.

La Unidad Aritmética y Lógica (ALU) contiene los siguientes registros:

- Un acumulador de 8 bits.
- Un registro temporal de 8 bits.
- Un registro de indicaciones de 5 bits: cero, paridad, signo, no
- Un registro.

- Un registro temporal de 8 bits (TRP).

Son ejecutadas operaciones aritméticas, lógicas y de notación en la ALU. La

ALU está organizada por el registro temporal, el acumulador temporal y el flag -

Stop de recursos (carry), el resultado de la operación puede ser transferido al bus

interno o al acumulador. La ALU también es alimentada por el registro de indicacio-

nes.

El registro temporal recibe información del bus interno y puede enviar todo o

parte de ella al ALU, el registro de indicaciones o al bus interno. El acumulador -

(ACC) puede ser cargado por la ALU y el bus interno; puede transferir datos a cada

estado del procesador y al bus interno. El contenido del acumulador y el flag de

recursos pueden ser usados para una conversión decimal durante la ejecución

de la instrucción ALU.

11.162 REGISTRO DE INSTRUCCIÓN Y CONTROL.

durante la búsqueda de una instrucción, el primer byte de esta instrucción -

el código de operación) es transferido del bus interno al registro de instrucción

de 8 bits.

El contenido del registro de instrucción es usado para el decodifi-

cación de instrucciones. La salida del decodificador, combinada con varias señales de

tiempo, proporcionan las señales de control para el arreglo de registros, la *RAM*, y el *buffer* de datos. Además las salidas del decodificador de instrucción y control, alimentan las señales de tiempo y control de estados, las cuales generan el estado y las señales de tiempo externas.

#### A.1d) BUFFER DEL BUS DE DATOS.

Este *buffer* bidireccional de tres estados y 8 bits es usado para aislar el bus interno del *CPU* del bus externo de datos ( $D_0$  a  $D_7$ ).

#### A.2).- EL CICLO DEL PROCESAMIENTO.

El ciclo de instrucción está definido como el tiempo requerido para buscar y ejecutar una instrucción. Durante la búsqueda de la instrucción, una instrucción octario (de uno, dos o tres bytes) es extraída de la Memoria y depositada en el registro de instrucción del *CU*. Durante la fase de ejecución, la instrucción es decodificada y traducida a actividades específicas de procesamiento.

Cada ciclo de instrucción consiste de uno, dos, tres, cuatro o cinco ciclos de *clock*. Un ciclo de *clock* es requerido para cada vez que el *CU* tiene acceso a la Memoria o a un punto de entrada / salida. La porción del ciclo de instrucción en el cual, se realiza la búsqueda de una instrucción, requiere de un *clock* de *clock* para cada byte a ser buscado.

La duración de la porción de ejecución del ciclo de instrucción depende de la clase de instrucción que se está buscando. Algunas instrucciones no requieren ningún ciclo de *clock* más que el necesario para buscar la instrucción, otras, sin embargo, requieren de ciclos adicionales para escribir o leer el dato en ó de la memoria ó del dispositivo de Entrada / Salida.



Inicio de un ciclo de irabucción puede ocurrir cualquiera de los diez ciclos

de siguiente siguiente:

1) Búsqueda (fatch,  $H_1$ ).

2) Lectura de memoria (memory read).

3) Escritura de memoria (memory write).

4) Lectura de atach (atch read).

5) Escritura de atach (atch write).

6) Entrada de datos (input).

7) Salida de datos (output).

8) Interrupción (interrupt).

9) Alto (halt).

10) Alto e interrupción (halt-interrupt).

Los ciclos de algunos durante el proceso de un ciclo de irabucción particular

los dependen de la clave de irabucción, con lo ya establecido de que el primer ciclo

de algunos es siempre de irabucción. El procesamiento idéntico de ciclo de algunos

en proceso mediante la transferencia de una palabra de 8 bits de "Estado" (estado en

el cual, se encuentran el Microprocesador) durante el primer estado de los los ciclo

claves de algunos.

La estructura de irabucción de Estado es mostrada por los líneas de de-

tao ( $D_0 - D_7$ ) durante el irabucción de MFC. Este dato dato son guardado en el

latch, y usado para determinar los estados de control e cláusulas enlance. La

Una (MFC) muestra la irabucción de Estado y como se distribuida en el bus de de-

tao del procesamiento.



## DEFINICIÓN DE INFORMACIÓN DE STATUS.

Simbolo	Bit	Definición.
INT <sup>B</sup>	D <sub>0</sub>	Señal de reconocimiento a una solicitud de interrupción. La señal será usada para habilitar una instrucción <i>INT</i> en el bus de datos — cuando es activada la línea <i>INTA</i> .
<u>UD</u>	D <sub>1</sub>	Indica que la operación del ciclo de bus en un proceso será una función de escritura en memoria ó salida ( <i>MEM = 0</i> ). De otra manera una operación de lectura o entrada será ejecutada.
STCK	D <sub>2</sub>	Indica que el bus de direcciones contiene la dirección del Stack Pointer (Indicador de Apilamiento) para introducir un dato en el Stack.
ALTA	D <sub>3</sub>	Señal de reconocimiento para la instrucción <i>INT</i> .
OUT	D <sub>4</sub>	Indica que el bus de direcciones contiene la dirección de un dispositivo de salida y el bus de datos contendrá el dato que sale cuando <i>UD</i> es activada.
D <sub>1</sub>	D <sub>5</sub>	Proporciona una señal para indicar que el <i>Ctr</i> está en un ciclo de búsqueda ( <i>SEARCH</i> ) del primer byte de la instrucción.
INT <sup>B</sup>	D <sub>6</sub>	Indica que el bus de direcciones contiene la dirección de un dispositivo de entrada y el dato que entra debe ser colocado en el bus de datos cuando es activada <i>INTA</i> .
MEM <sup>B</sup>	D <sub>7</sub>	Designa que el bus de datos será utilizado para leer un dato de la memoria.

(B) Estos bits de status pueden usarse para controlar el flujo de datos dentro del bus de datos al *MEM*.

Tabla (A.1) Definición de los Bits de Status del *8080*.

**STATUS WORD CHART**

**TYPE OF MACHINE CYCLE**

		1	2	3	4	5	6	7	8	9	10	11	12
D <sub>0</sub>	DATA	0	0	0	0	0	0	0	0	1	0	1	1
D <sub>1</sub>	DATA	1	1	0	1	0	1	0	1	0	1	1	1
D <sub>2</sub>	STATUS	0	0	0	1	1	0	0	0	0	0	0	0
D <sub>3</sub>	HLTA	0	0	0	0	0	0	0	0	0	1	1	1
D <sub>4</sub>	OUT	0	0	0	0	0	0	1	0	0	0	0	0
D <sub>5</sub>	IN	1	0	0	0	0	0	0	1	0	0	1	1
D <sub>6</sub>	IMP	0	0	0	0	0	1	0	0	0	0	0	0
D <sub>7</sub>	COND	1	1	0	1	0	0	0	0	0	1	0	0

● STATUS WORD

**FIG A.2. DEFINICION DEL BIT DE STATUS**

## n.26) SECUENCIA DE TRANSICIÓN DE ESTADOS:

Todo ciclo de máquina dentro de un ciclo de instrucción consiste de tres a cinco estados (denominados  $T_1, T_2, T_3, T_4, T_5$  o  $T_{ij}$ ). El número de estados depende de la instrucción que es ejecutada y en particular al ciclo de máquina dentro del ciclo de instrucción. En la figura (n.3) se muestra el diagrama de estados externos del CPU - SISO. El diagrama muestra que las líneas READY, HOLD e Interrupt son muestreadas durante el ciclo de máquina, puesto que las condiciones de estas líneas pueden modificar la secuencia de transición.

El CPU - SISO, no indica directamente su estado interno por la transmisión de una señal de salida de estado durante cada estado; En su lugar, el SISO proporciona salidas directas de control (HATE, HMIN, WMIN,  $\overline{WR}$  y WRT) para el uso de circuitos externos.

Recalcamos que el SISO pasa a través de al menos tres estados en todo ciclo de máquina, y cada estado está definido por las transiciones sucesivas de bajo a alto de la señal de reloj  $\Phi_1$ . La figura (n.4) muestra la relación de tiempos en un ciclo de máquina de bitparalelo (LIC) eficiente.

La señal SMC identifica al primer estado ( $T_1$ ) en todo ciclo de máquina, como se observa la señal SMC está relacionada con la transición de bajo a alto del reloj  $\Phi_2$  teniendo un retardo asociado. La información de Status está colocada en el bus de datos ( $D_0 - D_7$ ) durante el mismo intervalo. El sentido de las señales de Status es controlada también por la fase  $\Phi_2$  del reloj; el pulso de  $\Phi_2$  durante  $T_1$  también carga las líneas de dirección del pronombre  $A_0 - A_{15}$  y permanece estable hasta después del estado  $T_3$ ; esto da espacio suficiente de tiempo para leer el dato en la memoria.



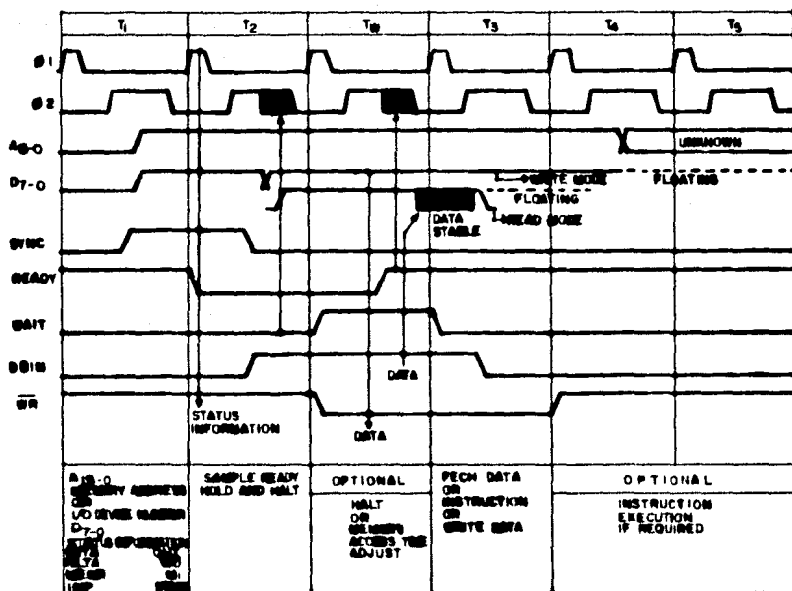


FIG. A.4. CICLO BÁSICO DE LA INSTRUCCIÓN

Una vez que el procesador ha enviado una dirección a la memoria, esta tiene la oportunidad de solicitar una espera ( $t_{wait}$ ), esto se hace por la memoria colocando la línea  $\overline{RDV}$  en estado lógico "0", esta línea permanecerá así hasta que la memoria responde. El intervalo de tiempo del estado  $T_2$  es siempre múltiple del periodo de reloj. El procesador responderá esta petición de espera entrando al estado  $T_3$  después del estado  $T_2$  en lugar de entrar directamente al estado  $T_3$  y esto es indicado por el procesador enviando la señal  $\overline{WAIT}$ , reconvirtiéndolo la solicitud de la memoria.

Los eventos que tienen lugar durante el estado  $T_3$  son determinados por la carga de ciclo de algunos en proceso. En un ciclo de  $t_{wait}$  (siempre  $t_{wait} < T_3$ ) el procesador en

termina el ciclo en su fase de datos con una destrucción. Durante un ciclo de lectura de memoria o del Stack, el dato en este bus es interpretado como una palabra de datos. Los datos de salida del procesador son colocados en este bus durante el ciclo de ejecución en memoria (Memory Buffer). Durante las operaciones de Entrada/Salida (I/O), el procesador puede transmitir o recibir datos de propósito a la operación de entrada o salida.

La tabla (n.2) indica las actividades asociadas a cada estado, y en concordancia con ella y a lo anteriormente descrito presentamos un diagrama de tiempo que muestra el ciclo de salida y otros que muestran un ciclo de entrada, figura (n.5).

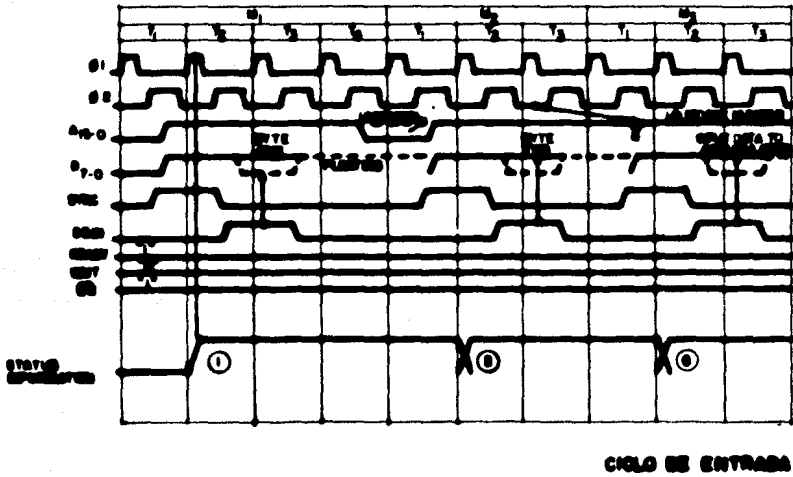
ACTIVIDAD ASOCIADA

ESTADO

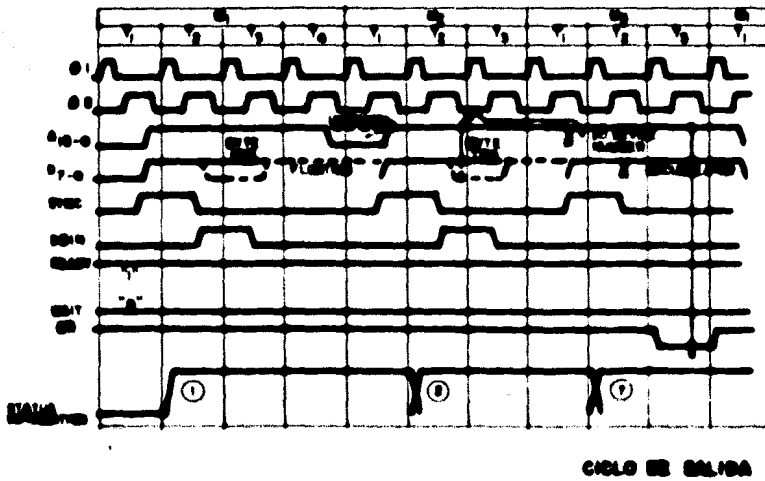
Una dirección de memoria o el número de un dispositivo de entrada es colocado en el bus de direcciones; se informa el bus de datos en relación con el bus de datos.  
 El Cu es activado (punto) las entradas  $R_{10}$ ,  $R_{11}$  y  $R_{12}$  pueden ser activadas la destrucción  $R_{11}$ .

El procedimiento es en estado de espera (WAIT) al  $R_{10}$  y  $R_{11}$  se baja o al la destrucción  $R_{11}$  ha sido ejecutada.  
 Un bus de instrucciones (ciclo de ejecución de programa), -  
 byte de datos (lectura de memoria, lectura de estado) o una destrucción de interrupción de interrupción en el  $R_{11}$  via al bus de datos; o un byte de datos (ciclo de ejecución de programa de datos) en estado de espera, o en estado de espera -

después de haberse ejecutado.  
 Los estados  $R_{10}$  y  $R_{11}$  son activados al la ejecución de una destrucción (punto) se requiere; al no, el  $R_{11}$  puede -  
 recibir un o estado. Los estados  $R_{10}$  y  $R_{11}$  son activados una destrucción de interrupción de interrupción.



CICLO DE ENTRADA



CICLO DE SALIDA

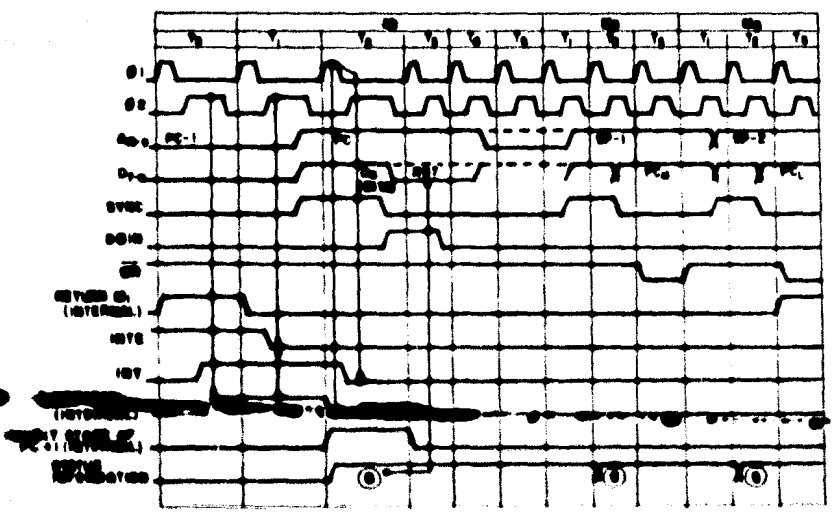
NOTA: ① Se refiere a la posición de Status de la Fig. A-2.

FIG. A-3

A.3).- SECUENCIA DE INTERRUPCION.

El 8080 es capaz de manipular solicitudes externas de Interrupción. Un dispositivo periférico puede iniciar una interrupción, simplemente elevando (colocando en estado lógico "1") la línea de Interrupción (INT) del procesador.

La entrada de interrupción (INT) es asíncrona, y por lo tanto, una petición — puede originarse en cualquier instante durante cualquier ciclo de instrucción. Mediante lógicas internas sincroniza la petición externa, de modo que se establece correspondencia con el reloj. La figura (A.6) esquematiza un circuito de Interrupción.



NOTA (N) Se refiere a la posición de Status de la Fig. A.5.

FIG A.6 SECUENCIA DE INTERRUPCION.



El conjunto de instrucciones del CPU preparadas para ejecución en el modo de interrupción, debe ser la instrucción de interrupción (INT), la que permite al sistema de instrucciones de interrupción (INT) en el modo de interrupción, para que se pueda ejecutar en el modo de interrupción. La dirección de ejecución de las instrucciones de interrupción (INT) en el modo de interrupción, debe ser la instrucción de interrupción (INT).

**El ciclo de interrupción en el modo de interrupción**

El ciclo de interrupción en el modo de interrupción, se inicia cuando se produce una interrupción. Durante el ciclo de interrupción, el sistema de instrucciones de interrupción (INT) en el modo de interrupción, debe ser la instrucción de interrupción (INT). El ciclo de interrupción en el modo de interrupción, se inicia cuando se produce una interrupción. Durante el ciclo de interrupción, el sistema de instrucciones de interrupción (INT) en el modo de interrupción, debe ser la instrucción de interrupción (INT). El ciclo de interrupción en el modo de interrupción, se inicia cuando se produce una interrupción. Durante el ciclo de interrupción, el sistema de instrucciones de interrupción (INT) en el modo de interrupción, debe ser la instrucción de interrupción (INT).

**Fin de la interrupción**

Como se observa, una petición de interrupción llega durante el tiempo en que la línea llamada "habilitación de interrupción" (INT) es alta y activa en coincidencia con  $\theta_2$  para excitar el flip-flop interno INT. Este evento ocurre durante el último estado del ciclo de interrupción, en el cual, es activada la interrupción, asegurando así, que cualquier interrupción en proceso sea completamente ejecutada en

dedicada a datos en: 8, 16, 24, 32, 40, 48, 50. Cualquiera de estas direcciones pueden ser usadas para almacenar la memoria (a) instrucción (ea) de una rutina de rutina para servir los requerimientos del dispositivo que interrumpe.

### 4.1.- SECUENCIA DE RETENCIÓN

El CPU - 8080 tiene medios que permiten el acceso directo a la memoria (LDR) - por algún otro dispositivo, mediante la aplicación de una señal HLL a la terminal oportuna en el procesador, un dispositivo externo puede ocasionar que el CPU que para sus operaciones normales y operaciones el control de los buses de datos y direcciones. El procesador responde a una petición de este tipo haciendo fletar sus buses. Al mismo tiempo el procesador reconoce la notación (HLL) colocando un "1" lógico en su terminal HLL. Durante la identificación de una notación, los buses de datos y direcciones están controlados por el control de la notación del cual se origina la petición operativa para transferir datos desde o hacia la memoria sin la intervención del procesador.

El ciclo de notación, el ciclo de interrupción, el ciclo de notación es sincronizado - ni igual que el ciclo de interrupción, ni igual que el ciclo de notación de tiempo que lo de interrumpe. Las figuras (1-6) y (1-6b) muestran los diagramas de tiempo que lo de interrumpe. Nótese el retardo existente entre la petición HLL y el HLL al mismo tiempo. También se observa en las figuras, como la configuración de HLL, HLL y el pulso de notación de interrupción. El diagrama de este tipo hace por que se observe la configuración de los buses de datos y direcciones (HLL) que el procesador hace fletar una vez que el procesador ha iniciado en el estado de notación (HLL).

El reconocimiento de la notación no hace fletar el dispositivo al inicio de que el procesador hace fletar una vez que se inicia la notación y cuando el procesador reconoce un HLL de interrupción de tipo HLL o fletar de interrupción de tipo HLL o fletar de interrupción de tipo HLL.



adriose. De otro modo, el reconocimiento es realizado en el inicio del estado siguiente a  $T_3$ .

Todo hace parecer que el procesador ha suspendido sus operaciones una vez que el bus de datos y direcciones está flotando, sin embargo, internamente algunas — funciones pueden continuar. Si una solicitud de MMU es reconocida en  $T_3$  y si el procesador está a la mitad de un ciclo de máquina, el cual requiere de cinco o — seis estados para completarse, el CPU procede a través de  $T_4$  y  $T_5$  antes de que vuelva la pausa del procesador, así el procesamiento interno es permitido transitar la transiencia externa de MMU, mejorando sobre la eficiencia y la velocidad de todo el sistema.

Una solicitud de MMU es terminada asincrónicamente cuando el dispositivo externo ha terminado su transferencia de datos. La salida MMU regresa a su estado — "bajo" lógico siguiente la transición de alto a bajo del primer pulso de reloj  $\Phi_1$ .

#### 4.31.- SECUENCIA DE MLTU:

Cuando una instrucción MLT es ejecutada, el CPU entra en el estado de alto — (MLT)  $T_{MLT}$ , después del estado  $T_2$  del primer ciclo máquina, como se muestra en la figura (4.31). Existen alto tres formas de que el CPU pueda salir del estado MLT.

- Un "1" lógico en la línea de RESET, siempre hará regresar al CPU al estado  $T_1$ ; en RESET siempre barra el PC.

- Una solicitud de MMU original que el CPU entra en el estado de notificación, como previamente fue descrito, cuando la línea MMU presenta un nivel bajo, el CPU regresa al estado MLT durante la transición de bajo a alto del primer pulso de  $\Phi_1$ .

- Una interrupción (o deca, un "1" en INT) siempre que INT es habilitado original que el CPU sale del estado MLT y entra en el estado  $T_1$ .

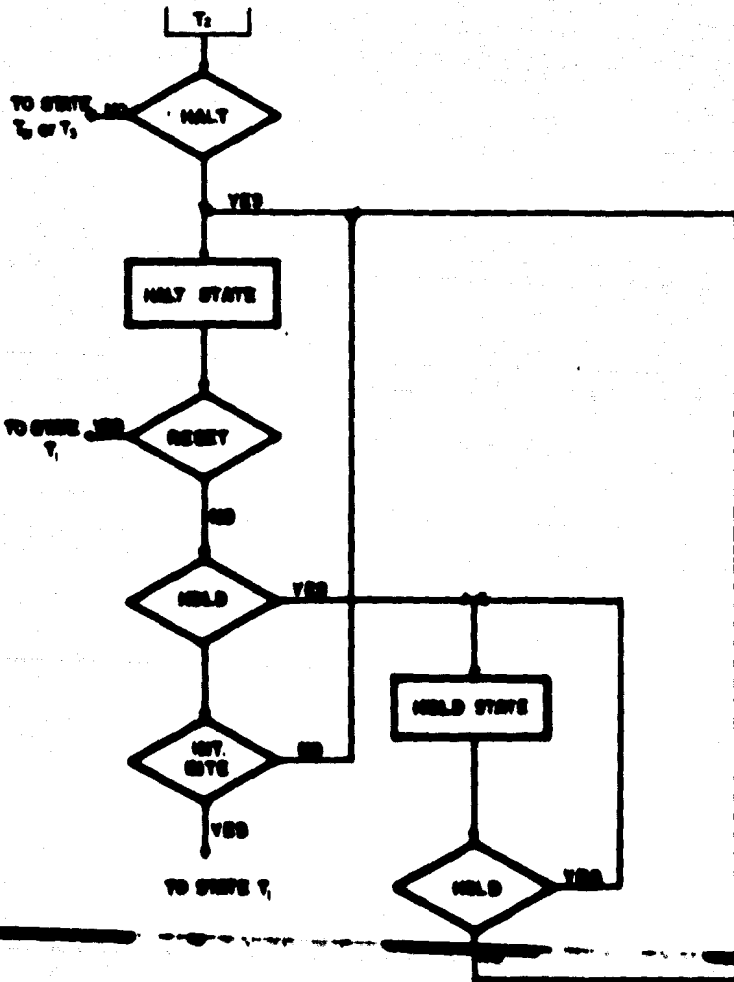


FIG. A-3 SEQUENCE HALT.

durante la transición de bajo a alto del próximo pulso del reloj  $\Phi_1$ .

**Nota:** El Flip - Flop deberá ser "disparado" cuando el 8080 es introducido en el estado de Hold, de otro modo, el 8080 sólo podrá salir del estado Hold emitiendo una señal de RESET.

## A.6) INSTRUCCIONES DEL 8080.

El grupo de instrucciones del 8080, incluye operaciones aritméticas y lógicas, las cuales son direccionadas en modo directo, indirecto e inmediato. Los grupos de instrucciones para traslado (MOV), carga (MVI) y almacenamiento de datos (STO), nos proporcionan la capacidad de intercambiar datos de 8 o 16 bits entre Memoria, los seis registros de trabajo y el Acumulador; utilizando modos de direccionamiento directo, indirecto e inmediato.

La capacidad de saltar hacia diferentes posiciones del programa es proporcionada por las instrucciones JMP, JMP Condicional, Jump Computado. También, mediante estos saltos condicionales e incondicionales llegamos a una subrutina y retornamos de ella.

Operaciones de doble precisión; tales como manipulación de Stack e instrucciones de doble precisión, las cuales aumentan la capacidad aritmética y el manejo de integridades del 8080. Las instrucciones del 8080, nos habilitan para incrementar y decrementar la memoria, los 6 registros de trabajo y el Acumulador, así como instrucciones para incrementar o decrementar los registros punteros y el Stack Pointer. También nos habilitan para rotar el contenido del acumulador hacia la derecha o izquierda junto con ó sin rotación del bit de signo.

Salidas de entrada o salida con parámetros utilizando instrucciones de memoria en

en puertos de Entrada / Salida (I/O) o mediante el direccionamiento directo que nos proporcionan las instrucciones de I/O.

El siguiente subgrupo de instrucciones especiales completa el grupo de las instrucciones del  $\text{CPU}$ :

- La instrucción **HALT**.
- La instrucción **INTI** para detener la operación del procesador.
- La instrucción **DAD** proporciona aritmética decimal.
- La instrucción **SFC** permite disparar el indicador de errores (Carry).
- La instrucción **CHK** permite complementar el indicador de errores.
- La instrucción **CLR** complementa el contenido del acumulador.
- La instrucción **MOV** intercambia el contenido de los registros pares de 16 bits directamente.

#### a.6a).- FUENTE DE INSTRUCCIONES Y DATOS:

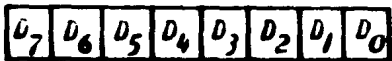
Los datos en el  $\text{CPU}$  están almacenados en forma de enteros binarios de 8 bits. La transferencia de datos al bus de datos del sistema está en el mismo formato.



**DATOS DE DATOS**

Las instrucciones del programa pueden ser de uno, dos o tres bytes de longitud. Las instrucciones de tres bytes se ejecutan en tres ciclos de reloj consecutivos desde el comienzo del programa. Las fuentes de las instrucciones, entonces dependen de la operación particular a ser ejecutada.

*Instrucciones de un byte.*



OP CODE

Registro a registro, referencia de memoria, aritméticos o lógicos, rotación, retorno, push, pop, instrucciones de interrupt.

*Instrucciones de dos bytes.*



OP CODE

Instrucciones de dato inmediato o en Entrada / Salida.



OPERAND 1

*Instrucciones de tres bytes.*



OP CODE

Instrucciones Jump, Call o de Load y Store directo.



OPERAND 1



OPERAND 2

*A continuación se viene un resumen de instrucciones del procesador 8080.*



**6800 INSTRUCTION SET**

**Summary of Processor Instructions**

Mnemonic	Description	Instruction Code (1)						Cycles	Mnemonic	Description	Instruction Code (1)						Cycles					
		Op	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>4</sub>	R <sub>5</sub>				Op	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>4</sub>	R <sub>5</sub>						
<b>MOVE, LOAD, AND STORE</b>																						
MOV R1:2	Move register to register	0	1	0	0	0	5	5	JPO	Jump on parity odd	1	1	0	0	0	1	0	10				
MOV R0:r	Move register to memory	0	1	1	0	0	5	5	7	PCHL	H & L to program counter	1	1	1	0	1	0	1	5			
MOV r:0	Move memory to register	0	1	0	0	0	1	1	0	7												
MOV r	Move immediate register	0	0	0	0	0	1	1	0	7	<b>CALL</b>											
MOV R0	Move immediate memory	0	0	1	1	0	1	1	0	10	CALL	Call unconditional	1	1	0	0	1	1	0	1	17	
LIR 0	Load immediate register Pair 0 & C	0	0	0	0	0	0	0	1	10	CC	Call on carry	1	1	0	1	1	1	0	0	11	17
LIR 0	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10	CNC	Call on no carry	1	1	0	1	0	1	0	0	11	17
LIR H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10	CZ	Call on zero	1	1	0	0	1	1	0	0	11	17
LIR H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10	CNZ	Call on no zero	1	1	0	0	1	0	0	0	11	17
STAR 0	Store A indirect	0	0	0	0	0	0	1	0	7	CP	Call on positive	1	1	1	0	1	0	0	0	11	17
STAR 0	Store A indirect	0	0	0	1	0	0	1	0	7	CM	Call on minus	1	1	1	1	1	0	0	0	11	17
LDA r 0	Load A indirect	0	0	0	0	1	0	1	0	7	CPE	Call on parity even	1	1	0	0	1	1	0	0	11	17
LDA r 0	Load A indirect	0	0	0	1	1	0	1	0	7	CPO	Call on parity odd	1	1	0	0	1	0	0	0	11	17
STA	Store A direct	0	0	1	1	0	0	1	0	13	<b>RETURN</b>											
LDA	Load A direct	0	0	1	1	0	1	0	1	13	RET	Return	1	1	0	1	0	0	1	0	1	10
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16	RC	Return on carry	1	1	0	1	1	0	0	0	5	11
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16	RNC	Return on no carry	1	1	0	1	0	0	0	0	5	11
CHNG	Exchange D & E H & L Registers	1	1	1	0	1	0	1	1	4	RZ	Return on zero	1	1	0	1	0	0	0	0	5	11
<b>STACK OPS</b>																						
PUSH 0	Push register Pair 0 & C on stack	1	1	0	0	0	1	0	1	11	RNZ	Return on no zero	1	1	0	1	0	0	0	0	5	11
PUSH 0	Push register Pair 0 & E on stack	1	1	0	1	0	1	0	1	11	RP	Return on positive	1	1	1	0	0	0	0	0	5	11
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	11	RM	Return on minus	1	1	1	1	0	0	0	0	5	11
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	11	RPE	Return on parity even	1	1	1	0	1	0	0	0	5	11
POP 0	Pop register Pair 0 & C off stack	1	1	0	0	0	0	1	1	10	RPO	Return on parity odd	1	1	1	0	0	1	0	0	5	11
POP 0	Pop register Pair 0 & E off stack	1	1	0	1	0	0	1	1	10	<b>START</b>											
POP H	Pop register Pair H & L off stack	1	1	0	0	0	0	1	1	10	ST	Restart	1	1	A	A	A	1	1	1	1	11
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10	<b>INCREMENT AND DECREMENT</b>											
STHL	Exchange top of stack H & L	1	1	1	0	0	0	1	1	10	INR r	Increment register	0	0	0	0	0	1	0	0	5	
SHL	H & L to stack pointer	1	1	1	1	0	0	1	1	5	DEC r	Decrement register	0	0	0	0	0	1	0	1	5	
L0I SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10	INR 0	Increment memory	0	0	1	0	1	0	0	1	10	
I0I SP	Increment stack pointer	0	0	1	1	0	0	1	1	5	DEC 0	Decrement memory	0	0	1	0	1	0	1	1	10	
DEC SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5	INR 0	Increment 0 & C registers	0	0	0	0	0	0	1	1	5	
<b>JUMP</b>																						
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10	INR 0	Increment D & E registers	0	0	0	1	0	0	1	1	5	
JC	Jump on carry	1	1	0	1	1	0	1	0	10	INR H	Increment H & L registers	0	0	1	0	0	0	1	1	5	
JNC	Jump on no carry	1	1	0	1	0	1	0	1	10	DEC 0	Decrement 0 & C	0	0	0	0	1	0	1	1	5	
JZ	Jump on zero	1	1	0	0	1	0	1	0	10	DEC 0	Decrement D & E	0	0	0	1	0	1	1	1	5	
JNZ	Jump on no zero	1	1	0	0	0	1	0	1	10	DEC H	Decrement H & L	0	0	1	0	0	0	1	1	5	
JP	Jump on positive	1	1	1	0	0	1	0	1	10	<b>ADD</b>											
JM	Jump on minus	1	1	1	1	0	1	0	1	10	ADD r	Add register to A	1	0	0	0	0	5	5	5	4	
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10	ADD r	Add register to A with carry	1	0	0	0	1	5	5	5	4	
JPO	Jump on parity odd	1	1	1	0	0	1	0	1	10	ADD 0	Add memory to A	1	0	0	0	1	1	0	7		
<b>ADD</b>																						
DAD A	Add H & C to H & L	0	0	0	1	0	0	0	1	10	ADD 0	Add memory to A with carry	1	0	0	0	1	1	0	7		
DAD r	Add R to H & L	0	0	1	1	0	0	1	10	ADI	Add immediate to A	1	1	0	0	0	1	0	7			
DAD H	Add H & L to H & L	0	0	1	0	0	0	1	10	ADI	Add immediate to A with carry	1	1	0	0	1	1	0	7			
DAD SP	Add stack pointer to H & L	0	0	1	1	0	0	1	10	<b>DATA</b>												

NOTES: 1. 000-555 0. 000-001 0. 010 0. 011 H. 100 L. 101 Memory 110 A. 111  
2. Two possible cycle times: 0. 12; multiple instruction cycles dependent on condition flags.

All trademarks Copyright Intel Corporation 1977

Table (A.3) Instructions and Cycles - 6800A

**6800 INSTRUCTION SET**

**Summary of Processor Instructions (Cont.)**

Mnemonic	Description	Instruction Format								Cycles
		0	1	2	3	4	5	6	7	
<b>SUBTRACT</b>										
SUB	Subtract register from A	1	0	0	1	0	5	5	5	4
SUB	Subtract register from A with borrow	1	0	0	1	1	5	5	5	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SUB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SB	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
<b>LOGICAL</b>										
ANA	And register with A	1	0	1	0	0	5	5	5	4
XRA	Exclusive Or register with A	1	0	1	0	1	5	5	5	4
ORA	Or register with A	1	0	1	1	0	5	5	5	4
CP	Compare register with A	1	0	1	1	1	5	5	5	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	0	7	
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CP M	Compare memory with A	1	0	1	1	1	1	1	0	7
AMB	And immediate with A	1	1	1	0	0	1	1	0	7
XB	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORB	Or immediate with A	1	1	1	1	0	1	1	0	7
CP	Compare immediate with A	1	1	1	1	1	1	1	0	7
<b>SHIFTS</b>										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
<b>STATUS</b>										
CSA	Component A	0	0	1	0	1	1	1	1	4
SIC	Set carry	0	0	1	1	0	1	1	1	4
CC	Component carry	0	0	1	1	1	1	1	1	4
SCA	Set decimal adjust A	0	0	1	0	0	1	1	1	4
<b>INPUT/OUTPUT</b>										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
<b>CONTROL</b>										
SI	Enable interrupts	1	1	1	1	0	1	1	0	0
DI	Disable interrupts	1	1	1	1	0	0	1	0	0
NOP	No operation	0	0	0	0	0	0	0	0	0
HALT	Halt	0	1	1	0	1	1	0	0	7

LA CULTURA ELECTRÓNICA DE LOS AÑOS 60

En los cinco años siguientes después de que aparecieron los computadores elec-

trónicos de bolsillo, llegó a pensarse en un posible desarrollo de la

observación de las nuevas aplicaciones de cálculo como multiplicación y división

de números complejos que aparecen en el display al instante de operar la te-

cla " = ". Sin embargo es posible, aunque sea poco de lo que han gozado de es-

ta esencial expansión y aceleración en sus capacidades de cálculo, han tenido --

no que una idea general de como ha evolucionado la cultura en su interior. La

bien conocido que el funcionamiento de una computadora electrónica está basado en

un lenguaje "chip" microelectrónico. Pero como están representadas para dicho fun-

cionamiento las mismas matemáticas y la organización lógica de la computadora?

En términos de hardware y de su principio de operación, la computadora electrónica

de bolsillo es consecuencia directa de las computadoras electrónicas de "efect-

iva" y de sus aplicaciones "profesionales de escritorio". Hace apenas una década

existieron tipos de estas funciones (adición, sustracción, multiplicación y di-

visión) de este tipo incorporadas en circuitos integrados especiales y en

un cambio de diseño. El desarrollo de la industria electrónica llegó a un estado

de tal manera que algunas funciones incorporadas en algunos más pequeños y de

menor costo para obtener un mayor espacio. Sin embargo, en estos momentos, a --

principio de 1965 la forma tradicional de computación electrónica está unido

de una manera bastante en un solo chip microelectrónico (C. I.). La historia

de este desarrollo data de 1967.

La historia en la tecnología de los semiconductores ha sido marcada por un ritmo y predecible incremento en la complejidad de los circuitos integrados cuando con la reducción de costos en toda la producción de semiconductores, incluyendo los LSI (Diseños Embebidos de Lógica), los cuales, son utilizados en la mayoría de las aplicaciones para implementar un diseño. Este tipo de evolución es a menudo interpretado como una "Ley de Moore". Basada en la experiencia ganada de un crecimiento constante en el volumen de producción. Alrededor de 1970 hubiera sido posible introducir más de la mitad de los tipos de chips de un solo chip tipo MSU - (Metal - Oxide - Semiconductor) a un precio menor de 100 Dólares.

Con esta tecnología a lo largo de un año se logró desarrollar una nueva generación de circuitos integrados de campo de acción (FPGA). Los cálculos típicos para un solo chip de (C) 1. para realizar todas las funciones de cálculo, así como, control de circuitos de salida. Algunos de los nuevos cálculos fueron reducidos de tamaño, integrados en un solo chip para su utilización, otros modelos fueron adaptados de estructuras existentes que operaban con conceptos antiguos. Uno de los casos típicos de esta reducción en los precios (bien abajo de los 100 Dólares), lo cual dio a estos dispositivos un mayor mercado. Ahora que la industria está eliminando de los sistemas de unidades este efecto en la curva de aprendizaje, el rango de los precios está más abajo de los 20 dólares. Los pequeños calculadores han evolucionado con rapidez y con los avances y aplicaciones que ninguna aplicación de campo de acción se esperaba a través los cálculos. No obstante es posible aplicar los principios de funcionamiento en base a una hipótesis calculadora de ciertos principios de funcionamiento de un chip más rápido. Este chip no es de última generación en dispositivos electrónicos, pero es la base de la tecnología representativa de los chips en dispositivos electrónicos.

En esta tecnología a lo largo de un año se logró desarrollar una nueva generación de circuitos integrados de campo de acción (FPGA). Los cálculos típicos para un solo chip de (C) 1. para realizar todas las funciones de cálculo, así como, control de circuitos de salida. Algunos de los nuevos cálculos fueron reducidos de tamaño, integrados en un solo chip para su utilización, otros modelos fueron adaptados de estructuras existentes que operaban con conceptos antiguos. Uno de los casos típicos de esta reducción en los precios (bien abajo de los 100 Dólares), lo cual dio a estos dispositivos un mayor mercado. Ahora que la industria está eliminando de los sistemas de unidades este efecto en la curva de aprendizaje, el rango de los precios está más abajo de los 20 dólares. Los pequeños calculadores han evolucionado con rapidez y con los avances y aplicaciones que ninguna aplicación de campo de acción se esperaba a través los cálculos. No obstante es posible aplicar los principios de funcionamiento en base a una hipótesis calculadora de ciertos principios de funcionamiento de un chip más rápido. Este chip no es de última generación en dispositivos electrónicos, pero es la base de la tecnología representativa de los chips en dispositivos electrónicos.



El control de la información en el sistema de control de la información se realiza a través de la información que se genera en el sistema de control de la información. La información que se genera en el sistema de control de la información se utiliza para el control de la información.

En el punto de vista de la información, el sistema de control de la información se puede considerar como un sistema de control de la información. El sistema de control de la información se puede considerar como un sistema de control de la información. El sistema de control de la información se puede considerar como un sistema de control de la información.

Este es el punto de vista.

El sistema de control de la información se puede considerar como un sistema de control de la información. El sistema de control de la información se puede considerar como un sistema de control de la información. El sistema de control de la información se puede considerar como un sistema de control de la información.

Este es el punto de vista de la información.

El sistema de control de la información se puede considerar como un sistema de control de la información. El sistema de control de la información se puede considerar como un sistema de control de la información. El sistema de control de la información se puede considerar como un sistema de control de la información.

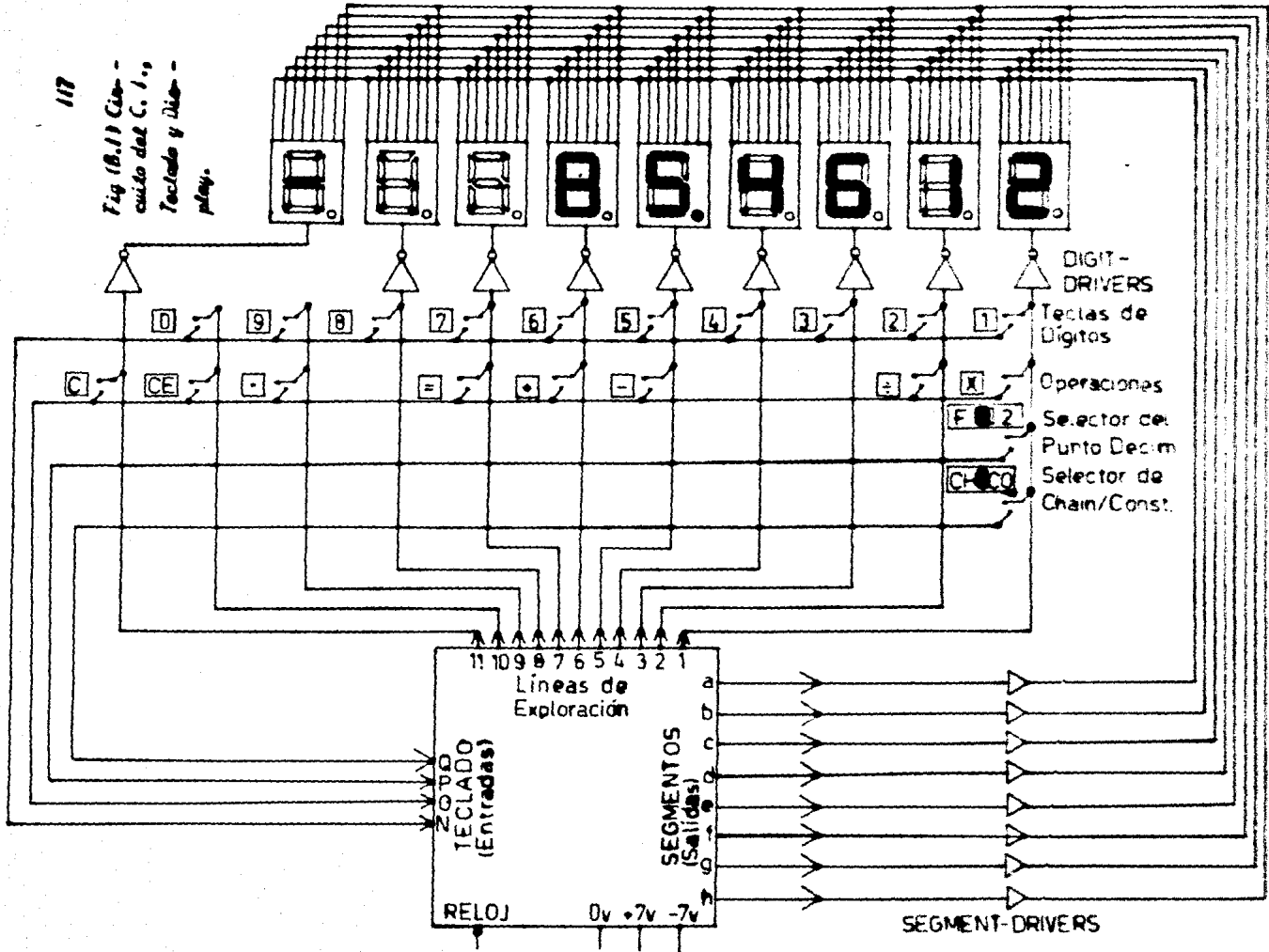
no el signo "menos", su punto decimal respectivo y los alícuotas para euros y naturas  
ción. En el caso de luminarias con terminales con conectores a la fuente de poder  
con el cableado de modo que el punto de conexión sea el mismo.

Las 25 terminales del chip también deben de aceptar la información enviada por  
el teclado. La calculadora tiene 18 teclas presionables y dos teclas de salida  
blancas (sin incluyendo el botón que enciende o apaga la calculadora). Cada uno de es-  
tas tiene dos terminales. Finalmente una tecla equivalente cuando el botón con-  
pulsado.

El problema de conexión 72 LEDs y 20 teclas de chip a través de 25 termina-  
les es solucionado como sigue. El teclado es conectado al chip a través de un conector  
de cuatro terminales de entrada, las cuales son designadas H, O, P y Q. Los LEDs de  
los segmentos son conectados a través de ocho terminales de salida las cuales son  
designadas por las letras a, b, c, d, e, f, g, h. En cualquier instante dado, el  
chip interconecta con un modo que un switch por cada línea de salida del teclado,  
e incluso no sólo de uno de los nueve segmentos de salida. Esto es posible para la  
solución de 11 líneas de "expansión", las cuales son designadas I, J, K, L, M, N, O, P, Q, R, S,  
T, U, V, W, X, Y, Z, AA, AB, AC, AD, AE, AF, AG, AH, AI, AJ, AK, AL, AM, AN, AO, AP, AQ, AR, AS, AT, AU, AV, AW, AX, AY, AZ, BA, BB, BC, BD, BE, BF, BG, BH, BI, BJ, BK, BL, BM, BN, BO, BP, BQ, BR, BS, BT, BU, BV, BW, BX, BY, BZ, CA, CB, CC, CD, CE, CF, CG, CH, CI, CJ, CK, CL, CM, CN, CO, CP, CQ, CR, CS, CT, CU, CV, CW, CX, CY, CZ, DA, DB, DC, DD, DE, DF, DG, DH, DI, DJ, DK, DL, DM, DN, DO, DP, DQ, DR, DS, DT, DU, DV, DW, DX, DY, DZ, EA, EB, EC, ED, EE, EF, EG, EH, EI, EJ, EK, EL, EM, EN, EO, EP, EQ, ER, ES, ET, EU, EV, EW, EX, EY, EZ, FA, FB, FC, FD, FE, FF, FG, FH, FI, FJ, FK, FL, FM, FN, FO, FP, FQ, FR, FS, FT, FU, FV, FW, FX, FY, FZ, GA, GB, GC, GD, GE, GF, GG, GH, GI, GJ, GK, GL, GM, GN, GO, GP, GQ, GR, GS, GT, GU, GV, GW, GX, GY, GZ, HA, HB, HC, HD, HE, HF, HG, HH, HI, HJ, HK, HL, HM, HN, HO, HP, HQ, HR, HS, HT, HU, HV, HW, HX, HY, HZ, IA, IB, IC, ID, IE, IF, IG, IH, II, IJ, IK, IL, IM, IN, IO, IP, IQ, IR, IS, IT, IU, IV, IW, IX, IY, IZ, JA, JB, JC, JD, JE, JF, JG, JH, JI, JJ, JK, JL, JM, JN, JO, JP, JQ, JR, JS, JT, JU, JV, JW, JX, JY, JZ, KA, KB, KC, KD, KE, KF, KG, KH, KI, KJ, KK, KL, KM, KN, KO, KP, KQ, KR, KS, KT, KU, KV, KW, KX, KY, KZ, LA, LB, LC, LD, LE, LF, LG, LH, LI, LJ, LK, LL, LM, LN, LO, LP, LQ, LR, LS, LT, LU, LV, LW, LX, LY, LZ, MA, MB, MC, MD, ME, MF, MG, MH, MI, MJ, MK, ML, MM, MN, MO, MP, MQ, MR, MS, MT, MU, MV, MW, MX, MY, MZ, NA, NB, NC, ND, NE, NF, NG, NH, NI, NJ, NK, NL, NM, NN, NO, NP, NQ, NR, NS, NT, NU, NV, NW, NX, NY, NZ, OA, OB, OC, OD, OE, OF, OG, OH, OI, OJ, OK, OL, OM, ON, OO, OP, OQ, OR, OS, OT, OU, OV, OW, OX, OY, OZ, PA, PB, PC, PD, PE, PF, PG, PH, PI, PJ, PK, PL, PM, PN, PO, PP, PQ, PR, PS, PT, PU, PV, PW, PX, PY, PZ, QA, QB, QC, QD, QE, QF, QG, QH, QI, QJ, QK, QL, QM, QN, QO, QP, QQ, QR, QS, QT, QU, QV, QW, QX, QY, QZ, RA, RB, RC, RD, RE, RF, RG, RH, RI, RJ, RK, RL, RM, RN, RO, RP, RQ, RR, RS, RT, RU, RV, RW, RX, RY, RZ, SA, SB, SC, SD, SE, SF, SG, SH, SI, SJ, SK, SL, SM, SN, SO, SP, SQ, SR, SS, ST, SU, SV, SW, SX, SY, SZ, TA, TB, TC, TD, TE, TF, TG, TH, TI, TJ, TK, TL, TM, TN, TO, TP, TQ, TR, TS, TT, TU, TV, TW, TX, TY, TZ, UA, UB, UC, UD, UE, UF, UG, UH, UI, UJ, UK, UL, UM, UN, UO, UP, UQ, UR, US, UT, UY, UZ, VA, VB, VC, VD, VE, VF, VG, VH, VI, VJ, VK, VL, VM, VN, VO, VP, VQ, VR, VS, VT, VU, VV, VW, VX, VY, VZ, WA, WB, WC, WD, WE, WF, WG, WH, WI, WJ, WK, WL, WM, WN, WO, WP, WQ, WR, WS, WT, WU, WV, WW, WX, WY, WZ, XA, XB, XC, XD, XE, XF, XG, XH, XI, XJ, XK, XL, XM, XN, XO, XP, XQ, XR, XS, XT, XU, XV, XW, XX, XY, XZ, YA, YB, YC, YD, YE, YF, YG, YH, YI, YJ, YK, YL, YM, YN, YO, YP, YQ, YR, YS, YT, YU, YV, YW, YX, YY, YZ, ZA, ZB, ZC, ZD, ZE, ZF, ZG, ZH, ZI, ZJ, ZK, ZL, ZM, ZN, ZO, ZP, ZQ, ZR, ZS, ZT, ZU, ZV, ZW, ZX, ZY, ZZ.

La solución de líneas en el chip, únicamente para la señal proveniente del  
teclado principal. Las líneas de expansión a la vez desde un teclado normal  
no como un teclado de este tipo, solamente interconecta a través de un teclado  
112 microcanales (55 canales del teclado). Existe un máximo de 6 canales entre el fi-  
nal de un canal y el inicio del siguiente. El teclado que un canal de expansión  
está completamente formado por el canal de expansión en un canal de este tipo.

Fig. (B.1) Circuito del C. 1, Teclado y Display.





El ejemplo de la que resulta de este ejemplo para algunas aplicaciones que se  
 líneas de exploración 2 y las medidas a, b, c, d, e, y, y cada una. El diagrama (fig.  
 con los (2) de exploración) líneas de exploración y el tipo de diagrama correspondiente.  
 de exploración las medidas de las exploraciones con el tipo de diagrama para el tipo de

El diagrama particular que se muestra en un tiempo dado en una o más posiciones  
 del diagrama se determinando por medio de las ocho direcciones de salida de las exp-  
 loras cada una durante un pulso de exploración. Cada una de las medidas de -  
 las exploraciones en un elemento de salida "segmento" líneas "segmento" de -  
 una". Cada exploración - datos se conecta al estado del LED en la misma posición en  
 las líneas del diagrama. (Como un ejemplo - datos se activado, conecta en -  
 estado correspondiente a la dirección particular de la fuente de tensión. Por lo que,  
 la medida / figura dada cada particular segment datos a medida del tipo LED, en  
 se debe a conectar a líneas a medida de un digit - a datos.

Como se puede con las líneas de exploración control 72 LED con ocho sen-  
 sadas de salida? Cada una de las nuevas líneas de exploración 11, 2, 3, 4, 5, 6, -  
 7, 8, 11) se conecta a un circuito inversor "invert - Invert" el cual, en cada  
 no el chip principal. El circuito es diseñado como un "Digit - Invert", cada dato  
 van a través de activado por un pulso de exploración, proporciona una trayectoria  
 de gran capacidad de corriente a través desde las ocho direcciones en cada sentido de -  
 datos. Por dato, la posición de un digit puede ser iluminado solamente durante el  
 tiempo en que la correspondiente línea de exploración se activada en cada ciclo de  
 exploración.

compruebe 427 ciclo de reloj, es decir aproximadamente 1.7 milisegundos.

Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso. Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso.

chip para la producción de un teclado de pulso.

El chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso. Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso. Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso.

Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso.

Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso. Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso. Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso.

Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso. Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso. Este artículo describe el chip para la producción de un teclado de pulso (un teclado de pulso) que se puede utilizar en un teclado de pulso.



La parte básica del sistema es el Sistema / Substrato, el cual es llamado por

destruido de un circuito.  
La figura (B.2). A continuación se describen dichos subatamas y se ejemplifican el  
construido en el sistema del chip principal con variados esquemas de conexión  
lógica y los algoritmos o rutinas mínimas. Los principios subatamas electrónicos  
hace también a nuevas hipotéticas estructuras, en la arquitectura u organización  
de la información. El teclado y el display son importantes, pero lo que realmente  
sea. En efecto, es un sistema general de procesamiento de información de conside-  
rable altura, realista, múltiple y diversa, efectúa un complejo proceso de auto-  
De lo tratado anteriormente resulta evidente que la cultura es el resultado de pro-

La parte básica del sistema es el Sistema / Substrato, el cual es llamado por  
destruido de un circuito.  
La figura (B.2). A continuación se describen dichos subatamas y se ejemplifican el  
construido en el sistema del chip principal con variados esquemas de conexión  
lógica y los algoritmos o rutinas mínimas. Los principios subatamas electrónicos  
hace también a nuevas hipotéticas estructuras, en la arquitectura u organización  
de la información. El teclado y el display son importantes, pero lo que realmente  
sea. En efecto, es un sistema general de procesamiento de información de conside-  
rable altura, realista, múltiple y diversa, efectúa un complejo proceso de auto-

La parte básica del sistema es el Sistema / Substrato, el cual es llamado por  
destruido de un circuito.  
La figura (B.2). A continuación se describen dichos subatamas y se ejemplifican el  
construido en el sistema del chip principal con variados esquemas de conexión  
lógica y los algoritmos o rutinas mínimas. Los principios subatamas electrónicos  
hace también a nuevas hipotéticas estructuras, en la arquitectura u organización  
de la información. El teclado y el display son importantes, pero lo que realmente  
sea. En efecto, es un sistema general de procesamiento de información de conside-  
rable altura, realista, múltiple y diversa, efectúa un complejo proceso de auto-

La parte básica del sistema es el Sistema / Substrato, el cual es llamado por  
destruido de un circuito.  
La figura (B.2). A continuación se describen dichos subatamas y se ejemplifican el  
construido en el sistema del chip principal con variados esquemas de conexión  
lógica y los algoritmos o rutinas mínimas. Los principios subatamas electrónicos  
hace también a nuevas hipotéticas estructuras, en la arquitectura u organización  
de la información. El teclado y el display son importantes, pero lo que realmente  
sea. En efecto, es un sistema general de procesamiento de información de conside-  
rable altura, realista, múltiple y diversa, efectúa un complejo proceso de auto-

La parte básica del sistema es el Sistema / Substrato, el cual es llamado por  
destruido de un circuito.  
La figura (B.2). A continuación se describen dichos subatamas y se ejemplifican el  
construido en el sistema del chip principal con variados esquemas de conexión  
lógica y los algoritmos o rutinas mínimas. Los principios subatamas electrónicos  
hace también a nuevas hipotéticas estructuras, en la arquitectura u organización  
de la información. El teclado y el display son importantes, pero lo que realmente  
sea. En efecto, es un sistema general de procesamiento de información de conside-  
rable altura, realista, múltiple y diversa, efectúa un complejo proceso de auto-

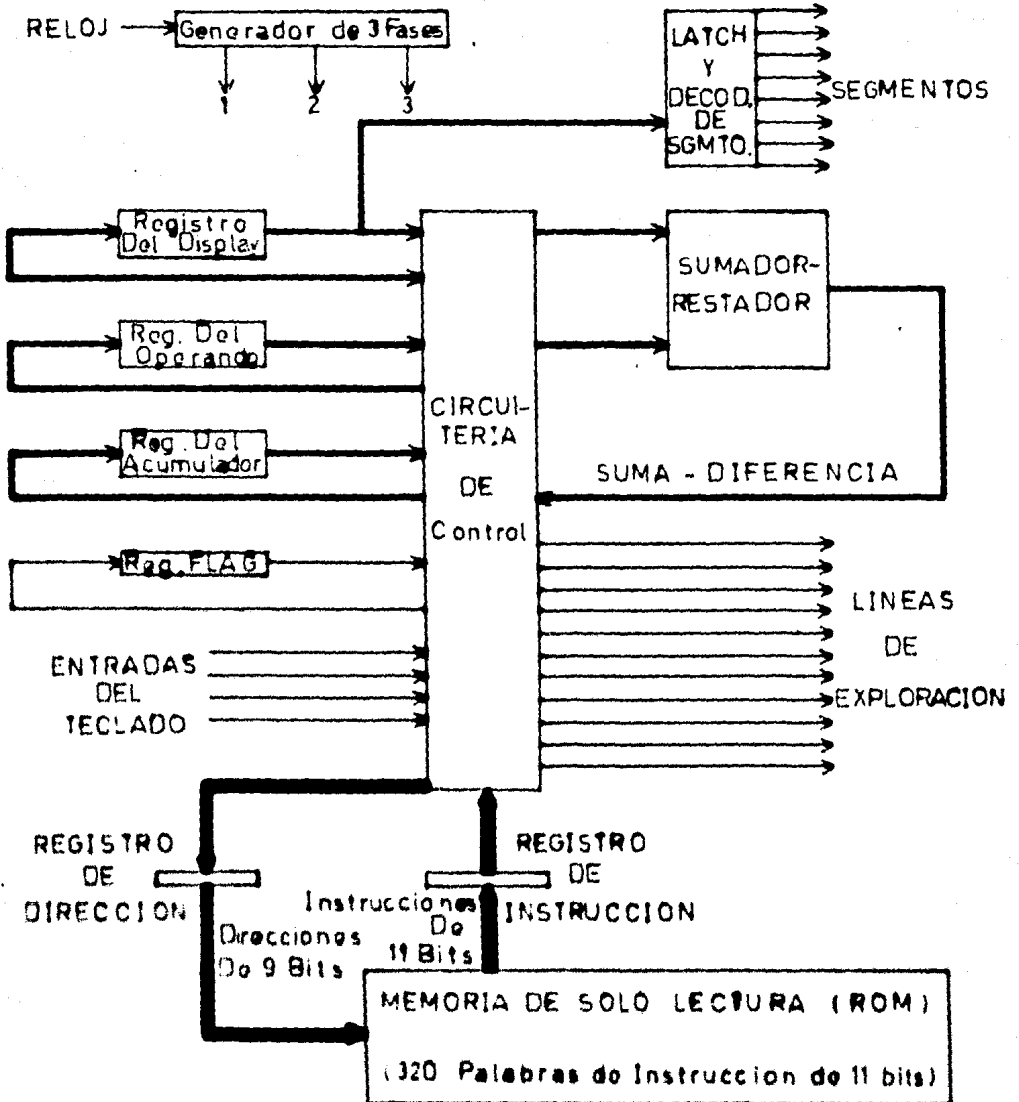
Los dos sistemas coinciden en el momento para una operación dada que -  
 generalmente cuando el registro del diagrama y del registro del operador, entonces  
 es el resultado en el registro del momento. Los dos registros coinciden en  
 registros de estado de 0 a 13 bits. La salida de los registros es efectuada por  
 elementos que se encuentran dentro del bloque. En la salida de los con-  
 tadores, estos elementos empujan los datos en los registros empujando  
 información sincronizada. Los dos registros son de hecho parte integral del "re-  
 gistro de estado" (dos registros) de un ciclo registro de contadores distintos de -  
 10 a 13 bits que se trata de registrarlos empujando cada una parte de ellos. El re-  
 gistro es un registro que los datos se manejan y se los maneja. El estado de  
 los contadores (con los datos de estado) se manejan en un registro de estado de  
 los contadores (con los datos de estado) se manejan en un registro de estado de  
 los contadores (con los datos de estado) se manejan en un registro de estado de

Para poder acceder a los datos para convertir el código a un formato en código  
 de 16 bits, este código debe ser complementado a "1" para luego ser usado. La salida de  
 cada y de cada un decodificador empuja cada salida de uno o más.

**funciones accesorias.**

Hay tres señales derivadas del reloj que llamamos "fases". En condiciones de la  
 señal del reloj principal, un elemento interno pulsa otras tres señales llamadas  
 "fases", de una en una en un ciclo repetitivo (de acuerdo al reloj), proporcionalmente a  
 un factor de control accesorial para ciertos sub-sistemas que deben desarrollarse en

hecho en un estado antes de que sean aceptadas por otros sub-sistemas.  
 cada nuevo conjunto de señales de salida características a un sub-sistema dado para  
 ver para sincronizar la actividad de los diferentes sub-sistemas y para asegurar que  
 un a todos los componentes del circuito de la calculadora. Los pulsos de reloj son



durante un ciclo de instrucción se completa precisamente una recirculación. El bloque de control también proporciona cierta lógica en la rutina, como interconectar un número entre registros cuando éstos son recirculados, y de reemplazar algunos

de los números recirculados en la salida del sumador. Además ciertas rutinas incluyen una extensión o un retardo de uno o dos estados (tres o seis pulsos de reloj). De esta manera la rutina y el circuito de retardo pueden imponer un retardo más largo o más corto en un registro, que es transmitido a los otros y así de esta manera recorrer el número contenido en un determinado registro en una u otra dirección con respecto a los demás registros. Finalmente una entrada a los circuitos de rutina es mediante un dígito sintetizado por otros circuitos en el bloque de control. Este es el recorrido de un dígito decodificado proveniente de una entrada del teclado, el cual puede iniciar su camino a través de los registros.

Al decodificador y a la circuitería de control y tiempo puede llamarse "Controlador", el cual regula a los demás subsistemas en el chip. El Controlador está dirigido por cualquiera de las trececientas veinte instrucciones. Cada una de las instrucciones tiene una longitud de palabra de 11 bits, las cuales son proporcionadas al Registro de Instrucción y después al Controlador, y provienen de la memoria ROM (Read Only Memory), así llamada por que contiene un conjunto de instrucciones operacionales que no es posible cambiar una vez establecida su configuración. Cada instrucción, obtenida de la Memoria ROM mediante direcciones de 9 bits, establece los reglas de operación que se aplican durante un Ciclo de Instrucción de 13 Estados (39 ciclos de reloj).

Durante cada ciclo de instrucción los registros de currículum pasan en desarrollo, no obstante una completa recirculación aún también una suma o una resta, así dicho operando es reemplazado. Hoyaduna se en relación a la recirculación de registros—





hacia el Sumador, el "2" como 0010 y el "3" como 0011. Allí se realiza la suma y el resultado, es colocado en el acumulador. Desde allí el resultado es desplegado hacia el registro del display para que luego sea desplegado.

Probablemente en cálculos más complejos que " $2 + 3$ " se requieran algunos pasos a través de los circuitos de rutina y de retardo y posiblemente también a través del Sumador. Supóngase que el primer dato de la operación es 25.6. El número 2 es introducido primero y aparece en el display con un punto decimal a su derecha. Cuando el número 5 es introducido, el 2 deberá desplazarse un lugar a la izquierda en el display, pero el punto decimal deberá permanecer en su lugar, a la extrema de derecha. Luego la tecla del punto decimal es presionada, sin producir algún cambio en el display, pero la señal de esta tecla es almacenada como una notación en el Flag-Register. Ahora cuando la tecla del número 6 es presionada, el punto decimal es desplazado un lugar hacia la izquierda al igual que el "2" y el "5". Para recordar el punto decimal se requiere que la calculadora realice un ciclo de memoria, el cual, es aplicado al dígito del punto decimal, que es uno de los 13 dígitos en el registro del display, y es captado para no perder de vista el punto decimal.

Si el siguiente comando del teclado es "Suma" y el número a ser sumado (a — 25.6) es 33.14, la primera acción a realizar es mover la línea del Punto Decimal a los dos registros involucrados, tal y como se haría en una suma manual. En otros ejemplos, los dígitos del punto decimal en los dos registros cuyos contenidos están — siendo sumados, deberán ser igualmente desplazados el 25.6 un lugar a la izquierda, moviendo un lugar al dígito del punto decimal (que sería 25.60). Después de haber — realizado lo anterior, los números pueden sumarse en un paso a través del Sumador.

Los ejemplos anteriores deberían ser suficientes para indicar la frecuencia de



circuitos de control, y que es solicitada por alguna instrucción en un programa escrito en lenguaje de máquina).

La operación de un microprograma en una computadora o una calculadora es semejante a un programa en el que generalmente se involucran fetching (alcanzar) una instrucción después de otra, desde una memoria de cierto tipo, de acuerdo con la dirección determinada parcialmente por el resultado de la operación anterior. El concepto de un programa, sin embargo ordinariamente denota un conjunto de instrucciones que pueden ser más fácilmente combinadas y alteradas externamente a la máquina, lo mismo que los microprogramas. La distinción es evidente en calculadoras programables, donde cada instrucción de un programa (posiblemente un programa hecho por el operador) típicamente lleva a una rutina microprogramada del tipo que se ha descrito. Esta distinción en algunas ocasiones es abstracta, cuando se considera en ejecución las calculadoras electrónicas otras funciones aparte de las cuatro operaciones básicas, tal como, obtener una raíz cuadrada, en  $Sov$  o calcular  $V^2$ . Tales funciones son generalmente ejecutadas mediante una secuencia programada (llevada a cabo a través de microprogramas) de sumas, restas, multiplicaciones y divisiones.

Un resultado de la estructura parecida a una computadora que tienen las partes de las calculadoras electrónicas de bolsillo es que, como con una computadora, una variedad de diferentes rutinas programadas pueden ser ejecutadas por la misma estructura del circuito electrónico. (Ordinariamente programas diferentes son colocados en una computadora mediante la lectura de instrucciones en la memoria, mientras que los microprogramas de una calculadora electrónica de bolsillo son cargados en su memoria ROM durante el montaje).

## BIBLIOGRAFIA

## Manuales:

- Manual del Usuario Sistema 8080.
  - Guía del Usuario M C S - 80.
  - Manual de Instrucciones y Programación
- Editado por I N T E L Co.

## Catálogos:

- Digital Circuit Integral.
- Interfases.

Edic: National Co.

- T T L Data Book.

Edic: Texas Instruments Inc.

## Libros:

- Microprocesador / Microprogramming Handbook.  
aut: Bruce Wood.  
Edic: Tab Book, F M T S Edition.
- Microprocesadores y Microcomputadores.  
Serie: Mundo Electrónico.  
Edic: Norberto Arizono Ediciones.
- Digital Design With Standard M S I and L S I.  
aut: Thomas R. Shallos.  
Edic: John Wiley & Sons.

- *Introduction To Switching Theory and Logical Design.*

Aut: Frederick J. Hill.

Coauthor: Gerald R. Peterson.

Edit: Wiley International Edition.

- *The Design Of Digital Systems*

Aut: John B. Panton.

Edit: Mc Graw - Hill.

- *Designing With T T L Integrated Circuits.*

Edit: Texas Instruments Inc.

International Student Edition.

- *Scientific American. March 1976.*