

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES

“CUAUTITLAN”



DISEÑO DE UN PROBADOR DE CIRCUITOS  
INTEGRADOS DIGITALES

T E S I S

QUE PARA OBTENER EL TITULO DE:  
INGENIERO MECANICO ELECTRICISTA  
P R E S E N T A

ALFONSO FLORES TAPIA

DIRECTOR DE TESIS: M. EN C JUAN ANTONIO NAVARRO M.  
PREMIO NACIONAL DE CIENCIA Y TECNOLOGIA BANAMEX  
CUAUTITLAN IZCALLI, EDO. DE MEX. 1985



Universidad Nacional  
Autónoma de México



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

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

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

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

# TESIS CON FALLA DE ORIGEN

## PROLOGO.

El objetivo principal del desarrollo de ésta tesis, fue el de colaborar con el Laboratorio de Electrónica de la FES-C, con un sistema que pueda ser útil dentro de la misma además de cumplir con el requisito para obtener el título de Licenciado en Ingeniería Mecánica Eléctrica.

Hoy en día el Ingeniero o diseñador de sistemas digitales cuenta con una nueva herramienta, el microprocesador, que en combinación con los circuitos integrados digitales, dispositivos periféricos y memorias han originado lo que son las microcomputadoras, las cuales pueden ser aplicadas en la industria, en las escuelas, en comunicaciones, en el hogar, automotriz, etc.

El ingeniero puede utilizar las Características de programación de los microprocesadores y de los circuitos que lo auxilian para desarrollar dispositivos electrónicos en una forma más fácil y más sofisticados que usando únicamente circuitos electrónicos. El microprocesador puede bajar el costo e incrementar la flexibilidad de los dispositivos electrónicos. Sin embargo, la fase de programación puede tomar un largo proceso de aprendizaje para aquellas personas acostumbradas a trabajar con circuitos integrados digitales y transistores únicamente.

El diseño del Probador de Circuitos Integrados Digitales, sigue el mismo proceso, encontrándose a través del desarrollo con dispositivos como los ya mencionados, casi en su mayoría.

El Capítulo 1 se aboca al estudio de los circuitos integrados digitales, permitiendo al lector conocer la funcionalidad de un gran número de éstos. En el Capítulo 2 se da el aspecto físico del Probador, que pudiera tener como sistema, describiéndose las partes que la componen; así como, los conceptos básicos para su diseño. El Capítulo 3 trata específicamente de la circuitería (hardware) que se necesitó para su desarrollo, analizando cada uno de los circuitos. Este capítulo emplea además un sistema de microprocesador llamada: Microcomputadora.

En el Capítulo 4 se describe el diseño del programa utilizado por el Probador, que al igual que en el Capítulo 3 es basado en una micro-computadora con microprocesador Z80.

Al final de los capítulos se encuentran 2 apéndices, conteniendo el manual del usuario y las referencias bibliográficas.

Por este mismo medio, quiero hacer patente mi agradecimiento al -- Ing. José Luis Rivera López, a la Ing. Lourdes Clares Fuentes y en especial al M. en C. Juan Antonio Navarro Martínez, por la ayuda que me ---- brindaron en la elaboración de esta tesis.

POR MI RAZA HABLARA EL ESPIRITU, atontamente El Autor.

# I N D I C E

## CAPITULO 1:

### INTRODUCCION - CIRCUITOS INTEGRADOS DIGITALES

1.0	Generalidades.....	1
1.1	Antecedentes históricos.....	2
1.2	Generalidades de los circuitos integrados digitales..	3
1.3	Familias lógicas.....	7
1.3.A	RTL.....	7
1.3.B	DTL.....	9
1.3.C	TTL.....	9
1.3.D	HPL.....	11
1.3.E	ECL.....	12
1.3.F	C-MOS.....	13
1.4	Estudio de la familia lógica TTL.....	17
1.4.1	Compuertas lógicas.....	24
1.4.2	Multiplexores.....	27
1.4.3	Decodificadores.....	31
1.4.4	Circuitos aritméticos.....	34
1.4.5	Schmitt triggers.....	35
1.4.6	Comparadores de magnitud.....	35
1.4.7	Checadores y generadores de paridad.....	35
1.4.8	Flip-flops y latch.....	35
1.4.9	Contadores.....	39
1.4.A	Shift-Registers.....	42

## CAPITULO 2:

### DEFINICION DEL PROBADOR DE CIRCUITOS INTEGRADOS DIGITALES.

2.1	Generalidades.....	49
2.2	Definición del probador.....	49
2.3	Descripción del probador.....	50
2.4	Vista macroscópica del probador.....	50

2.5	Indicadores alfanúmericos.....	52
2.5.1	Indicadores con diodos emisores de luz.....	53
2.6	Introducción a las computadoras.....	57
2.6.1	Estructura de una computadora digital.....	58
2.6.2	Microprocesador y microcomputadora.....	60
2.7	Interfaces de entrada y salida.....	61
2.7.1	Señales de control de transferencia.....	61
2.7.2	Interfaz.....	63
2.7.3	Selección del dispositivo periférico.....	67
2.8	Dispositivos periféricos de entrada.....	67
2.8.1	Interruptores.....	69
2.8.2	Interface con teclados.....	71
2.8.3	Teclado codificado.....	72
2.9	Dispositivos periféricos de salida.....	75
2.9.1	Interfazando LEDs y lámparas.....	75
2.10	Generación de un caracter en una matriz de diodos LED.	78
2.11	Sistema de almacenamiento de datos.....	79

## CAPITULO 3:

### DESCRIPCION DEL HARDWARE.

3.1	Introducción al circuito.....	82
3.2	Circuito principal.....	84
3.3	Circuito interface.....	91
3.3.1	Interfaz Z80-PIO.....	92
3.3.2	Configuración de los circuitos integrados a prueba....	111
3.4	Tarjeta control de displays.....	114
3.4.1	Introducción.....	114
3.4.2	Usos de la tarjeta.....	114
3.4.3	Instructivo de operación.....	115
3.4.4	Funcionamiento a bloques.....	121

## CAPITULO 4:

### DISÑO DEL PROGRAMA NECESARIO.

4.1	Definición del Software.....	123
-----	------------------------------	-----

4.2	Conceptos básicos de programación.....	123
4.2.1	Operación.....	123
4.2.2	Instrucción.....	124
4.2.3	Programa.....	124
4.2.4	Lenguaje de máquina.....	125
4.2.5	Sistema hexadecimal.....	126
4.2.6	Mnemónico.....	128
4.2.7	Programa ensamblador.....	128
4.3	Actividades de rutina.....	129
4.3.1	Representación lógica.....	130
4.4	Estructura del programa.....	130
4.5	Equipo de prueba utilizado.....	175
CONCLUSIONES.....		178
APENDICE A:		
MANUAL DEL USUARIO.		
A.1	Introducción.....	180
A.2	Descripción del equipo.....	180
A.3	Modo de operación del sistema.....	181
APENDICE B:		
BIBLIOGRAFIA.....		194



# CAPITULO 1

## INTRODUCCION - CIRCUITOS INTEGRADOS DIGITALES.

### 1.0 GENERALIDADES

El uso extensivo que ha tenido la Electrónica en los últimos años, ha creado la necesidad de miniaturizar los componentes. Al principio --- cuando se usaban elementos discretos para implementar diferentes funciones lógicas, se ocupaba mucho tiempo, dinero y espacio para lograr hacer algún diseño importante. Gracias a los circuitos integrados, hoy en día la gran mayoría de los sistemas utilizados hace años caracterizados por el gran espacio que ocupaban, se han visto reducidos a un volumen mínimo.

Como la presente tesis, además de tratar concretamente el como --- probar un circuito integrado digital TTL de la serie 7400, da al lector los conceptos básicos necesarios de los circuitos integrados. Por lo --- que, primeramente y antes de empezar a definir los conceptos de éstos, --- comenzaremos por definir lo que es un sistema digital.

Se llama sistema digital, a las señales que tienen valores discretos únicamente. Si las señales discretas solo tienen 2 valores, entonces el sistema digital se llama binario.

En los circuitos digitales la información se encuentra en estados discretos (dígitos). Los sistemas digitales binarios usan como herramienta el álgebra de Boole, ésta es la expresión matemática de la lógica de Aristóteles, por lo que también reciben el nombre de circuitos lógicos.

### 1.1 ANTECEDENTES HISTORICOS.

Alrededor de los años 1955, los dispositivos electrónicos empleados en la construcción de sistemas digitales eran diodos semiconductores y tubos de vacío. Los diodos son relativamente pequeños, de dimensiones-

de décimas de pulgada, y consumen poca potencia. Por otro lado, los tubos de vacío son físicamente grandes, de dimensiones de pulgadas y consumen relativamente cantidades grandes de potencia, del orden de algunos watts. Aunque con diodos y resistencias se podían fabricar grandes cantidades de compuertas, fué necesario utilizar con bastante frecuencia -- tubos de vacío. La situación mejoró notablemente con el advenimiento del transistor en los años 1950. El transistor, que en general sustituye al tubo de vacío, consume mucho menos potencia y lo mismo que el diodo semiconductor, un transistor individual encapsulado tiene dimensiones físicas de décimas de pulgada.

Hasta aproximadamente el año 1965 las compuertas lógicas y sistemas digitales se construían con dispositivos semiconductores encapsulados individualmente y resistores. Aquellos se fabrican aplicando procesos físico-químicos a la superficie de una oblea muy pura de silicio. -- Los detalles de fabricación de semiconductores no nos conciernen, excepto para indicar que un dispositivo semiconductor típico tiene dimensiones, en la superficie de la oblea de silicio, que son del orden de varias milésimas de pulgada. En un dispositivo encapsulado individualmente, la mayor parte del espacio físico no corresponde para las conexiones eléctricas. Este es el inicio de una serie de desarrollo tecnológicos -- (que continúan actualmente y aún no se vislumbra su fin) que conducen a un tipo de dispositivo semiconductor llamado CIRCUITO INTEGRADO (CI), ya que en él están fabricados muchos transistores y diodos, es decir, integrados en una oblea de silicio; en la misma estructura están integrados los resistores y aún las interconexiones que necesitan para fabricar una compuerta completa o muchas compuertas o aún un sistema digital elaborado.

Las pastillas integradas (chips) comercialmente disponibles se encuentran en pastillas de integración en pequeña escala (SSI small-scale-integrated), de integración en media escala (MSI medium-scale-integrated), de integración en gran escala (LSI large-scale-integrated) y dispositivos de integración en muy grande escala (VLSI very large-scale-integrated). Por convenio, la denominación SSI se aplica a las pastillas -- con menos de 12 compuertas lógicas, MSI para el rango de 13 a 99 compuertas, LSI para el rango de 100 a 1000 compuertas y VLSI para más de --

1000 compuertas.

El origen de los conceptos manejador en sistemas digitales se pierden en la antigüedad. Sin duda, la gente primitiva podía contar, y el ábaco que puede competir favorablemente con una calculadora mecánica, tiene como mínimo 2500 años de antigüedad. Una calculadora digital mecánica, que podría sumar y estar, se construyó hacia el año 1600. Un dispositivo mecánico que incorpora algunos conceptos utilizados en los ordenadores actuales se diseñó en los comienzos del siglo XIX. Hacia 1930 se disponía de gran cantidad de calculadoras de mesa, y por lo tanto baratas, que podían sumar, restar, multiplicar y dividir. En la década de los 40, los circuitos electrónicos con tubos de vacío confieren a los sistemas digitales una velocidad que no alcanzan con los dispositivos mecánicos, pero esos sistemas serán físicamente grandes y consumirán gran cantidad de potencia. En la década de los 50 aparecieron los transistores y los circuitos integrados comenzaron a desarrollarse en la década de los 60. Al comienzo de los 80 la circuitería integrada se ha desarrollado hasta el punto que una pastilla integrada que disipe 1 W y se mantenga fácilmente en la palma de la mano, pueda realizar el trabajo de sistemas, que 30 años antes ocupaban una habitación completa de equipos y disipan miles de watts.

## 1.2 GENERALIDADES DE LOS CIRCUITOS INTEGRADOS DIGITALES.

Para implementar las funciones lógicas, existen varios tipos de circuitos integrados, que son, o fueron comerciales, unos optimizados para algunas características y otros para otras. A cada tipo se le denomina "Familia Lógica". Las familias lógicas principales son:

Resistor Transistor Logic	RTL
Diode Transistor Logic	DTL
Transistor Transistor Logic	TTL ( $T^2 L$ )
High-Threshold Logic	HTL
Emitter Coupled Logic	ECL
Complimentary Symmetry	
Metal Oxide Semiconductor	C-MOS (COS/MOS)

Otras familias que se nombran pero no son tan comerciales son la -

$I^2L$ ,  $T^3L$ , etc. Antes de pasar a estudiar las Características principales de cada familia lógica definiremos algunos términos en común.

FAN-IN.- Es el máximo número de entradas que puede tener una compuerta lógica sin afectar sus características. Por lo general es de 8, aunque en algunas familias llega a 5.

FAN-OUT. Es el máximo número de compuertas que se puede conectar a la salida de una compuerta lógica. En algunas familias es un número grande (más de 300) y en otras no llega a más de 10.

NIVELES LOGICOS.- Bajo circunstancias normales sólo hay dos estados lógicos: Cero o Uno. El cero se refiere al voltaje más negativo y el uno al más positivo. La nomenclatura para el cero es  $V_0$  (Low) y para el uno es  $V_h$  (High). Algunos circuitos digitales tienen tres estados, (tri-state) donde el tercero se refiere a un circuito abierto, este será estudiado mas adelante.

TIEMPO DE RESPUESTA.- (Time delay o propagation delay) es el tiempo que transcurre desde la aplicación de una entrada hasta que la salida es correcta. Hay dos distintos retrasos: Uno cuando la salida "va" a cero y otro cuando va a uno. Responden de distinta forma cuando la salida sube que cuando baja. El símbolo  $t_{pd}^+$  es el tiempo que pasa desde que la entrada llega el 50% de  $V_h$  hasta que la salida también llega al 50% de  $V_h$ , pero cuando va a uno;  $t_{pd}^-$  se define igual que el anterior pero cuando la salida va a cero.

Si  $t_{pd}^+$  se suma con  $t_{pd}^-$  y se le saca la mitad, tenemos lo que se llama  $t_{av}$ , el tiempo promedio de propagación o respuesta. El  $t_{av}$  varía con el voltaje de alimentación, el fan-out y la temperatura. El tiempo que transcurre desde  $0.9V_h$  hasta  $0.1V_h$  cuando la onda va de bajada se llama el "fall time" y cuando va de subida, el "rise time".

INMUNIDAD AL RUIDO (Noise Immunity).- Es la máxima amplitud de ruido que no causa salidas erróneas cuando es añadida a la señal de entrada. Hay básicamente dos tipos de ruidos que deben ser considerados en los sistemas lógicos. El de DC, es un valor de estado estable que difiere de los niveles lógicos de voltaje para el cero y el uno, por ejemplo:  $V_h$  es 5 Volts y el Voltaje de entrada a una compuerta no llega a los 5 o

se pasa. El otro ruido es el de AC que es debido a variaciones de alguna fuente de voltaje.

La inmunidad al ruido de DC de un circuito digital es la habilidad de ese circuito a mantener su estado lógico en presencia del ruido. La - inmunidad al ruido de DC se expresa así:

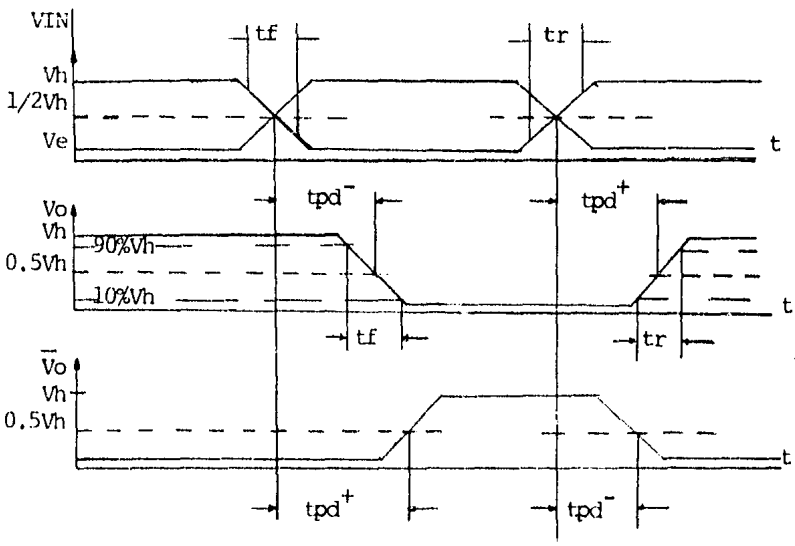


Figura 1.1. Gráfica del tiempo de respuesta.

$$NL = V_{il \text{ max}} - V_{OL \text{ max}}$$

$$NI = V_{OH \text{ min}} - V_{IH \text{ min}}$$

En donde:

$NL$  = Inmunidad al ruido de entrada cuando  $V_{in} = 0$

$NI$  = Inmunidad al ruido de entrada cuando  $V_{in} = 1$

$V_{iL}$  = Máximo  $V_{in}$  que se considera como 0

$V_{iH}$  = Mínimo  $V_{in}$  que se considera como 1

$V_{OL}$  = Máximo  $V_O$  que da al circuito como 0

$V_{OH}$  = Mínimo  $V_O$  que da al circuito como un 1

Por ejemplo, un fabricante de circuitos integrados TTL asegura que en la entrada debe haber de 0 a 0.8 volts máximo para que la compuerta - vea un cero en la entrada y de 2 a 5 volts para que vea un uno.

En cuando a datos de salida dice que, cuando hay un uno lógico, el mínimo voltaje es 2.8V y el máximo para un cero es 0.4V. La inmunidad - al ruido de este circuito es:

$$NL = 0.8 - 0.4 = 0.4V$$

$$NH = 2.8 - 2.0 = 0.8V$$

Los valores que da el fabricante de VIL, VIH, VOL y VOH muchas veces no son los valores en los peores casos de operación. Esto hay que -- tenerlo en cuenta para no hacer suposiciones incorrectas.

La inmunidad al ruido es un factor del Fan-out, Vcc y la temperatura. El primer factor que interviene en la generación del ruido de DC es la caída de voltaje por las líneas de alimentación.

El ruido de AC consiste en pulsos de poca duración que son generados por transientes de alta frecuencia y puede darse en las líneas de -- alimentación o en las líneas de datos. Los transientes causados por la - alta velocidad de cambio de las señales lógicas, también contribuyen a - la generación de ruido de AC.

La capacidad propia de los circuitos de entrada contribuye favorablemente a eliminar el ruido de AC. Este ruido es expresado en términos- de amplitud y duración.

Las principales fuentes de ruido AC son las líneas largas de datos y de alimentación, que actúan como antenas durante los transientes oca-- sionados al apagar una lámpara fluorescente, un motor o por transientes - propios de la fuente de alimentación de potencia.

LIMITES DE TEMPERATURA DE TRABAJO.- Son los límites (máximo y mí-- nimo) de temperatura para los cuales se asegura que el circuito integra- do funcionará correctamente. Por lo general los hay de tipo comercial -- que trabajan de 0 a 70°C y los del tipo militar que trabajan de -55°C a +125°C.

IMPEDANCIA DE SALIDA.- La habilidad de manejar cargas capacitivas- es una característica importante porque de aquí depende el tiempo de --- respuesta del circuito integrado y del sistema digital en sí. Si la im--

pedancia de salida es baja, la constante de tiempo es baja también, dando por resultado una buena respuesta a frecuencia.

PROTECCION CONTRA CORTOS CIRCUITOS.- Es la habilidad del circuito-integrado a que una o varias salidas sean puestas al potencial de tierra o Vcc sin que sufran daños. Algunas familias lógicas están protegidas -- contra cortos circuitos.

TOLERANCIA DE LA FUENTE DE ALIMENTACION.- Una vez diseñado todo el circuito lógico viene el problema de diseñar fuente de voltaje.

Algunas familias aceptan 10% de variación de Vcc y otras como los-CMOS aceptan voltajes de alimentación de 3 a 18 volts. El diseño de la -- fuente dependerá mucho de la familia que se use para implementar.

TIPO DE ENVOLTURA.- La más comercial de todas es la llamada DIP -- (Dual In line Package) y la hay de 4 a 60 patas de conexión. Los envuelven en plástico, o en cerámica. Otro empaque es el llamado FLAT, ocupa -- menos espacio que el DIP pero es más difícil de maniobrar. Algunos, pero muy raros, vienen encapsulados en el T0-5 de cápsula de metal. Suelen -- venderse también en CHIPS, o sea, el circuito integrado (CI) solo, (el sustrato de silicio sin ninguna envoltura), los CHIPS se utilizan para -- hacer circuitos híbridos. (en un sólo empaque hay varios CI'S para for-- mar una función determinada).

### 1.3 FAMILIAS LOGICAS.

1.3.A. RTL.- Resistor Transistor Logic, fue la primera familia lógica comercial y su proceso de fabricación es bastante simple. Se compone de resistencias y transistores y tiene unas desventajas respecto a -- otras familias lógicas, el Fan-out se ve reducido así como la inmunidad al ruido y la frecuencia de operación. En la Fig. 1.2 se muestra el circuito de una compuerta RTL NOR.

Cuando una de las dos o las dos entradas son uno, el voltaje de -- salida es cero y no fluye corriente hacia afuera del transistor. VOL es practicamente cero ( $V_{CE(SAT)}$ ). Si las dos entradas son cero, entonces el voltaje de salida es igual al de la fuente, si no está conectado a nada, pero si ésta salida maneja más compuertas entonces fluye una corriente -- por  $R_c$  y por  $R_s$  de la carga que se conectó. Si se siguen aumentando car-

gas, entonces la corriente que pasa  $R_C$  ya no será suficiente para proporcionar la  $I_B$  necesaria, a todas las bases de los transistores conectados, para mandarlos a saturación. Para cada carga que se conecta  $V_{OH}$  baja, hasta que llega a ser una indeterminación. El Fan-out para los RTL es de 5 normalmente. Claramente se ve que entre más compuertas se coloquen a la salida, la inmunidad al ruido disminuye ya que  $V_{OH}$  disminuye, normalmente es 0,2 volts de tal forma que si en un RTL se conectan las 5 cargas la inmunidad al ruido baja hasta 0,1 volts siendo bastante crítico.

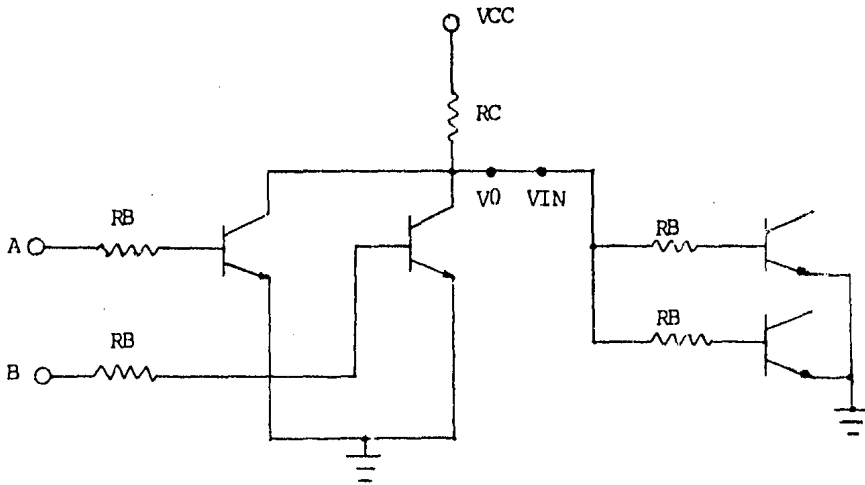


Figura 1.2. Compuerta RTL NOR.

La potencia de disipación de un RTL es 20 mW promedio por compuerta, esta potencia es considerada como mediana comparada con otras familias lógicas. El tiempo de propagación es algo grande, principalmente se debe a que la resistencia de la base, junto con el condensador propio de la unión BE, tienen una constante de tiempo grande. El tiempo de respuesta de los RTL anda entre 20 y 40 nanosegundos (ns).

La gran ventaja de los RTL es el precio. Son baratos pero tienden-



a desaparecer y por lo general ya no se usan en diseños de equipo nuevo.

1.3.B. DTL.- Diode Transistor Logic, son similares a los RTL con la diferencia que en vez de tener entradas a una resistencia las tienen a unos diodos. En la Fig. 1.3 se muestra el circuito de una compuerta -- DTL NAND.

Cuando A y B están en VH o abiertas, la corriente circula de Vcc - por R1 y R2 saturando el transistor Q1, luego pasa por D3 y por último - por Q2 a tierra. Al saturar Q2 el voltaje de salida es cero y si están - conectadas otras compuertas, hace que el Vin de esa compuerta sea  $\approx 0$ . - Regresemos a D1 y D2 para ver que pasa cuando están conectadas a tierra.

Si A ó B o las dos entradas están conectadas a tierra V1 es  $\approx V_d - (0.7)$ , si V1 = 0.7 entonces Q1 se encuentra en corte porque para saturar Q1 se necesita que V1 sea cuando menos  $V_{EBQ1} + V_{D3} + \dots + V_{EBQ2} = 0.7 \times 3 = 2.1$  Volts. Así que, si A ó B están a tierra Q1 está en corte y por lo tanto Q2 también, haciendo que VC sea el voltaje de la fuente (Vcc) y el fan-out no variará este voltaje porque todos los diodos conectados a la salida estarán polarizados inversamente.

El problema aquí no es el fan-out cuando la salida vale uno sino - cuando vale cero, porque en este caso, por cada diodo conectado circula una corriente que idealizando será igual a  $V_{cc}/(R1+R2)$ . Entonces el --- transistor Q2 debe tener capacidad de drenar estas corrientes. El máximo fan-out para los DTL normales es de 10. Tambien hay drivers con un --- fan-out hasta de 30.

La potencia de disipación es de unos 5 mW por compuerta. El tiempo de propagación anda en unos 20 ns por compuerta (velocidad media). Al - igual que los RTL se consideran de bajo costo pero hoy en día, debido a la muy alta producción de los TTL, el precio de éstos está siendo más -- bajo que los anteriores. Para nuevos diseños tampoco son recomendables.- Suelen tener una entrada extra sin diodo, de tal forma que se le puede - aumentar el fan-in hasta 20 colocando más diodos.

1.3.C TTL.- Transistor-Transistor-Logic.- Básicamente son iguales a los DTL a excepción que en vez de diodos se usan transistores con va-- rios emisores pero el circuito en sí viene a ser casi igual en operación. Las diferencias son que el transistor de entrada equivale a los diodos -

de entrada, y al transistor Q1 de los DTL (Fig. 1.3). La unión BC de Q1-estará conduciendo siempre que no conduzca ninguno de los emisores, así-se comporta como un diodo polarizado directamente. En la Fig. 1.4 se --muestra el circuito de una compuerta NAND TTL.

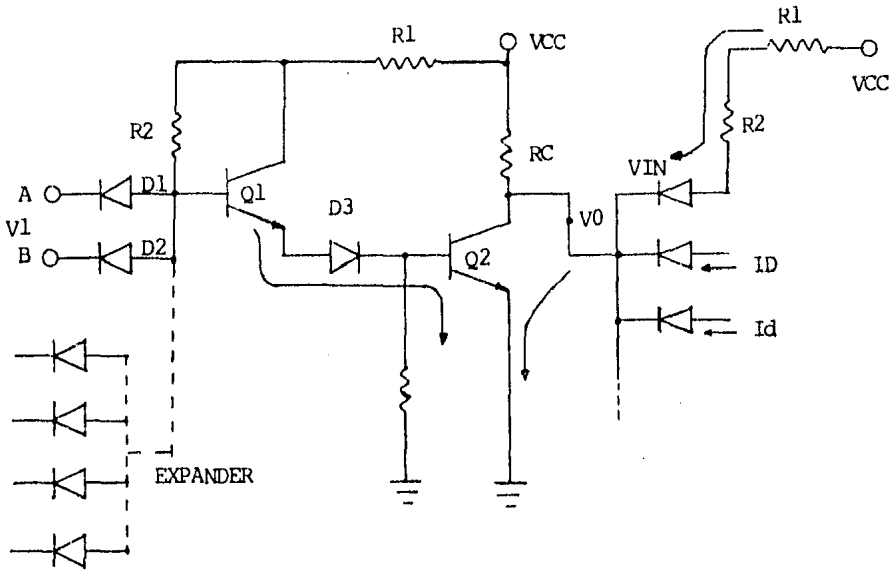


Figura 1.3. Compuerta DTL NAND.

Si A y B están abiertas o al positivo, por las uniones B-E de Q1 -no pasará corriente y la unión B-C del mismo transistor estará polariza--do directamente proporcionando corriente de base a Q2 y a Q3 haciendo --que dos transistores estén en saturación. En este caso, los puntos X y Y se encuentran prácticamente al mismo voltaje de tal forma que el tran---sistor Q4 esté en corte (para que Q4 se sature se necesitarán 1.4 volts--entre Vo y Y por que hay que vencer la barrera de potencial de la unión--B-E de Q4 y la del diodo D1).

El fan-out de las compuertas TTL es de 10 y está limitado, al ----

igual que el de los DTL, por la capacidad de manejar corriente de Q3. Si A ó B o las dos entradas estuvieran a cero volts entonces ya no fluiría corriente de base por Q2 porque el voltaje necesario en Z, para que la hubiera, sería igual a 2 veces 0.7 (Uniones E-B de Q2 y Q3) y si los emisores se encuentran a tierra el voltaje en Z baja a aproximadamente 0.7V. En este caso Q4 sí puede estar en saturación ya que como Q2 está en corte hace que  $V_y \approx V_{cc}$ . A la configuración de los transistores de salida se le llama TOTEM-POLE.

La inmunidad al ruido es de 0.4 volts, típicamente llega hasta 0.8 dependiendo de la configuración y del fabricante. El fan-out es de 10 pero hay drivers que elevan hasta 30 la capacidad de cargas a la salida.

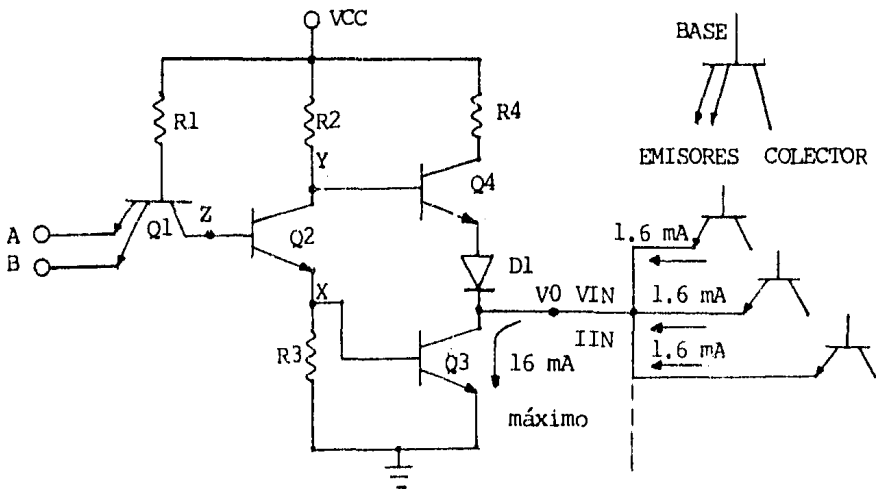


Figura 1.4. Compuerta NAND TTL.

1.3.D. HTL.- High Threshold Logic, son muy similares a los DTL con la diferencia que tienen un diámetro entre Q1 y Q3.

La inmunidad al ruido es  $NH=NL=4.5V$ . Estos circuitos se usan para interface (circuito acoplador) entre un sistema y otro por su gran inmunidad al ruido. El fan-out típico es de 10 compuertas. El tiempo de propagación es de 30 ns y su potencia de disipación es de más de 30 mW por-

compuerta. No son circuitos muy comerciales y con la aparición de los -- C-MOS lo más probable es que desaparezcan. En la Fig. 1.5 se muestra el circuito de una compuerta NAND HTL.

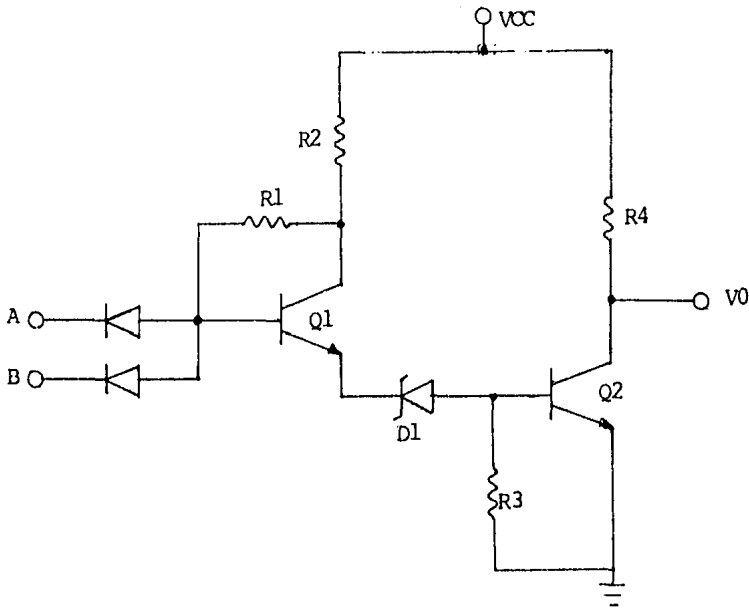


Figura 1.5. Compuerta NAND HTL.

1.3.E. ECL.- Emitter Coupled Logic. Esta familia tiene grandes desventajas como la potencia de disipación, su poca inmunidad al ruido y el costo pero son los más rápidos hasta el momento. Su voltaje de operación es de  $-5.2V$ , los voltajes de entrada son:  $V_{iL} = -1.45V$  y  $V_{iH} = 1.1V$ . Los voltajes de salida son:  $V_{oL} = -16$  y  $V_{oH} = -0.98 V$  dando una inmunidad de  $-N_1 = 0.12$  y  $N_2 = 0.15$ . El fan-in puede ser aumentado hasta 20 conectando transistores entre los puntos X y Y. El fan-out es considerablemente alto 20 o 25 compuertas. La potencia de disipación es de  $25 mW$  por compuerta para respuestas de  $2 ns$  y de  $75mW$  para un  $ns$ . Una ventaja es que en el mismo circuito tenemos salidas complementarias, evitando así el --



Voltaje de operación-----cualquiera.

Tiempo de respuesta-----que tienda a cero.

Impedancia de salida-----que tienda a cero.

Casi todas las características mencionadas las cumplen los C-MOS - donde la potencia en estado estable es de 0.01mW por compuerta y de 1mW a 1MHz. El fan-out es de unos 300.

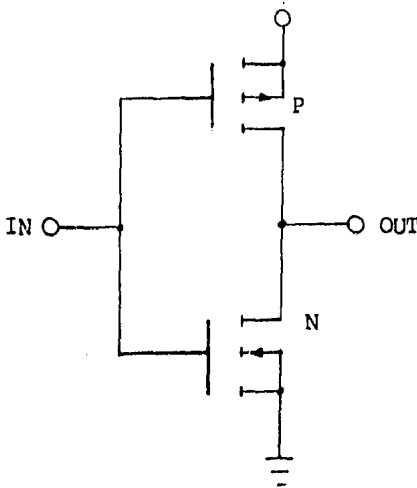


Figura 1.7. Inversor C-MOS.

El costo es muy bajo por lo simple de su fabricación, hoy en día - son comparables con los TTL. Su inmunidad al ruido es del 45%. El voltaje de operación es de 3 a 18 Volts. La impedancia de salida es menor de 1000 ohms y su única inconveniencia, el tiempo de respuesta es de unos - 25 a 35 ns, dependiendo del voltaje de operación.

En la Fig. 1.7 está mostrado un inversor C-MOS. El FET de arriba - es canal P y el de abajo es canal N. Cuando la entrada es un uno lógico - el canal P, entre drain y source, se comporta como una resistencia de -- 750 ohms. Entonces, con un uno a la entrada, la salida está a tierra a - través de una Resistencia de 750 ohms. Cuando  $V_{in} = 0$  el FET de abajo es - un circuito al positivo de la fuente.

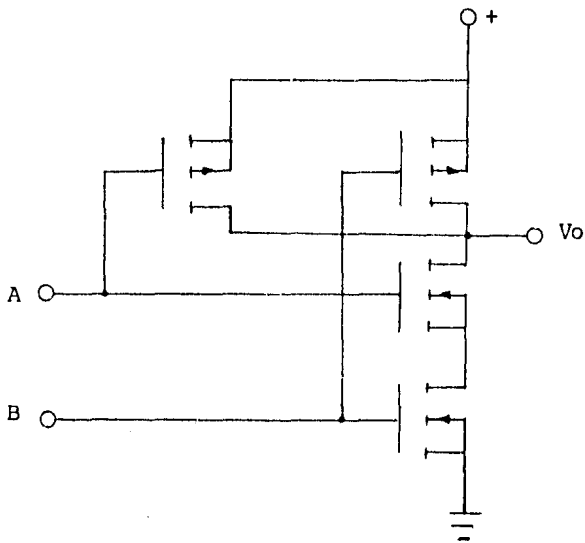


Figura 1.8. Compuerta NAND C-MOS.

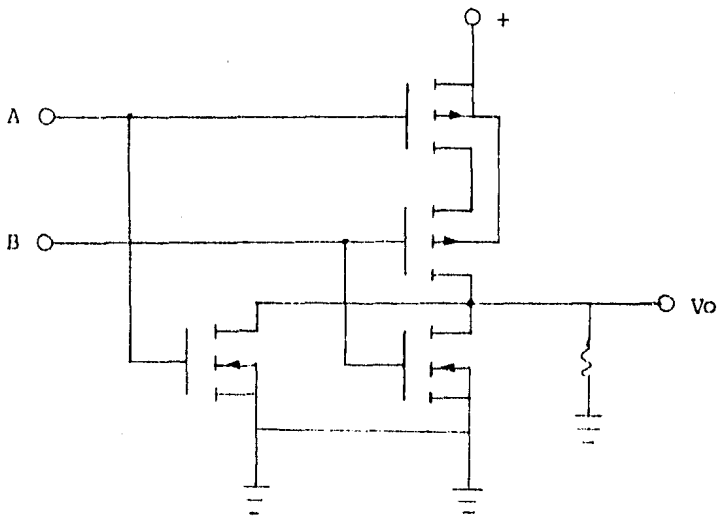


Figura 1.9. Compuerta NOR C-MOS.

CARACTERISTICAS	RTL	DTL	TTL	HTL	ECL	MOS	CMOS
SERIE	900	930	74/54		10000		4000
DISIPACION DE POTENCIA POR COMP. (mW)	20	10	1 a 25	30	35	1	10
FAN OUT	5	10	10	10	25	10	HASTA 300
FAN IN	5	10	8	5	5	8	8
MARGEN DE RUIDO NMh/NMl	REGULAR	BUENO	MUY BUENO	EXCELENTE	BUENO	REGULAR	MUY BUENO
FUENTE (Vcc) (v)	+ 3.6 a 5	+ 5	+ 5	+ 5	-5.2	-12	3 a 18
TIEMPO DE RESPUESTA (PROPAGATION DELAY TIME) (ns)	20	30	6 a 30	30 a 90	2	250	25
NIVELES LOGICOS 0/1 (v)	0.2/1.6	0.2/3.0	0.2/3.3		1.6/0.15	1.0/1.1	0/10

Tabla 1.0. Características típicas de las familias lógicas.



En la NAND (Fig. 1.8), con cualquier entrada que esté a cero, uno de los dos FET'S de arriba está en su estado de baja impedancia dando un uno a la salida. Se necesita que las dos entradas estén a uno para que los 2 transistores de abajo estén en su estado de baja impedancia y den cero a la salida. La compuerta NOR (Fig. 1.9) para que de un uno se necesita que tanto A como B sean cero.

#### 1.4 ESTUDIO DE LA FAMILIA LÓGICA TTL.

Una vez que se han definido los términos en común y características principales de cada familia lógica, incluyendo la de TTL, se hará un estudio más amplio de ésta por ser la más comercial y por cumplir con los requerimientos necesarios para el diseño como ya se mencionó. Pero la razón principal de este estudio, es dar a conocer al lector una idea general de lo que es y hace esta familia como CIRCUITOS INTEGRADOS (CIs), ya que la presente tesis trata precisamente de la prueba de estos.

La compañía Texas Instrument Inc. desarrolló principalmente esta familia, aunque ahora también la construyen otros fabricantes. Para la familia TTL, la compañía Texas Instrument Inc. anota en las pastillas (CIs) la denominación SN para indicar "Semiconductor network"; otros fabricantes utilizan denominaciones distintas, por ejemplo, DM (digital monolithic). Existen dos series en esta familia, una que se identifica por la familia 54 y la otra por el 74. La serie 54 se emplea para aplicaciones militares exigencias más restrictivas. Esta serie opera entre  $-55$  y  $125^{\circ}\text{C}$ . La serie 74 es la versión industrial de bajo costo con rango de temperatura de  $0$  a  $70^{\circ}\text{C}$ , su voltaje de operación es de 5 Volts  $\pm 5\%$ , el voltaje de operación de la serie 54 es de 5 Volts también, pero con un  $10\%$  de variación. A esta serie militar se le ha puesto el número 5400 en vez del 7400 para distinguir únicamente, ya que son compatible por pata, por esto se entiende que la configuración física no cambia, las patas de entrada y salida de la serie 7400 y la 5400 son las mismas.

En el año 1966, cuando la compañía Texas Instrument introdujo esta familia, se contaba con números desde el 7400 hasta los 74200 aunque hoy ya hay bastantes números que pasan de los 200. Para más información con-

sultar la referencia, manual The TTL Data Book (apéndice B).

Debido al posible compromiso potencia-velocidad ya que los tran---sistores pueden fabricarse como dispositivos estándar o Schottky. (Las - formas más obvias de eliminar los problemas de retardo de saturación es- simplemente no permitir al transistor entrar en saturación. Una forma de lograr esto es conectando un diodo Schottky entre la base y el colector- del transistor. A este transistor junto con el diodo se conoce como --- "TRANSISTOR SCHOTTKY" y tiene el símbolo especial que se ilustra en la - Fig. 1.A.

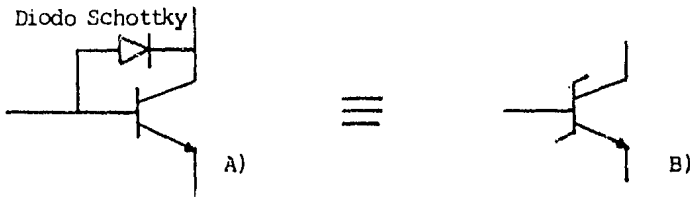


Figura 1.A. Transistor Schottky.

Dentro de la familia TTL se encuentran 5 series de CIs de alta ve- locidad y alta disipación (serie H), los schottky (serie S), los scho--- ttky de baja disipación (serie LS), baja velocidad y baja disipación --- (Serie L), lo mismo que CIs estándar, cuyas características se dan en la tabla 1.1.

Siempre que se esté usando un CI de la familia TTL es conveniente- que las entradas que no se utilicen estén conectadas a VCC para evitar - ruidos que hagan cambiar su estado lógico de la salida.

Entre la gran variedad de compuertas que hay de la serie TTL hay - unas llamadas "OPEN COLECTOR" (colector abierto). Por ejemplo la 7401 -- tiene un diagrama como el de la Fig. 1.10. El colector literalmente está abierto, no tiene ninguna carga conectada, de tal manera que si con un - voltímetro se mide  $V_o$ , para cualquier combinación de entradas, no marca- rá nada. Esta configuración es muy útil para implementar la "AND ALAM--- BRADA".

Supongamos varia salidas de diferentes compuertas pero todas de -- colector abierto. Si todos los colectores los alambremos en corto cir---

SERIES	COMPUERTAS			FLIP-FLOPS
	PRODUCTO VEL-POT.	TIEMPO DE RET. DE PROPAG.	DISIP. DE POTENCIA	RANGO DE FRECS. DE ENT. DE RELOJ
54LS/74LS	19 $\mu$ J	9.5 ns	2 mW	dc a 45 MHz
54L/74L	33 $\mu$ J	33 ns	1 mW	dc a 3 MHz
54S/74S	57 $\mu$ J	13 ns	19 mW	dc a 125 MHz
54/74	100 $\mu$ J	10 ns	10 mW	dc a 35 MHz
54H/74H	132 $\mu$ J	6 ns	22 mW	dc a 50 MHz

Tabla 1.1. Características típicas de la familia 54/74 SSI.

cuito y luego le conectamos a este punto una resistencia al positivo, -- vamos a tener una función lógica extra.

La salida de cada compuerta está marcada con diferente letra, como se muestra en la Fig. 1.11. Cualquiera que sea pero hace que  $V_o$  sea ce-- ro. Esta función lógica es la AND, que hace que haya un uno, sólo que -- todos sean uno. Cuando hay compuertas con colector abierto se simboliza-- como en la Fig. 1.12a, el equivalente sería el de la Fig. 1.12b.

Los diodos que aparecen en las entradas de la compuerta de la Fig. 1.10, los tienen todos los elementos de la serie TTL. Les llaman los --- "Input Clamping Diodes". Sirven para evitar que, por alguna razón, apa--- rezca un voltaje inverso en las entradas de las compuertas.

Hay que tener en cuenta que existen varias series TTL como la --- 7400, es decir, no sólo la serie 7400 es TTL, hay otras que tienen dife--

rentes fabricantes como National, signetics, Motorola, Intel, etc.

La gran variedad de compuertas y circuitos de pequeña y mediana -- escala de integración que se encuentra en la serie comercial no se en-- cuenta tan fácilmente para las familias de alta velocidad o baja potencia.

Los fabricantes de esta familia lógica establecen que para la entrada se debe tener entre 0 y 0.8 Volts para que el circuito entienda -- que hay un cero lógico. Para tener un uno lógico, es necesario un voltaje entre 2 y 5 Volts. Garantizan que todos los circuitos tendrán a la -- salida, entre 2.4 y 5 Volts cuando se trate de un uno lógico (Fig. --- 1.13).

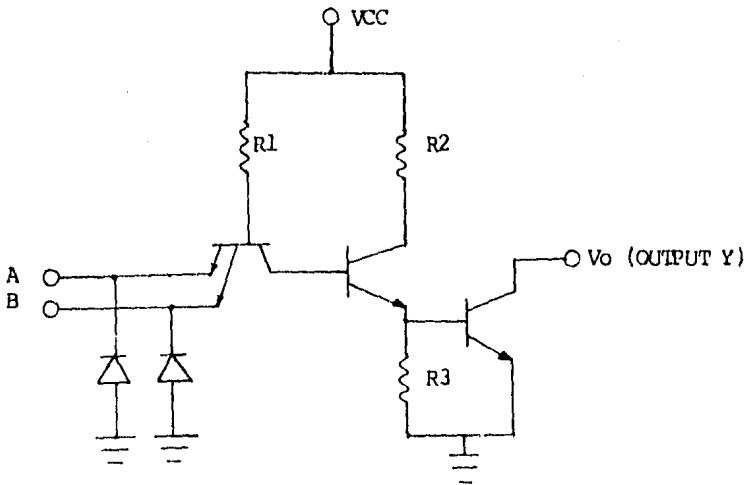


Figura 1.10. Compuerta TTL No. 7401 NAND de colector abierto.

Hay que tener presente que al dejar una entrada sin conectar, significa un uno lógico. Es muy recomendable no dejar entradas volando, si no se usan, deben conectarse a Vcc (de referencia) o a tierra.

Dado que las entradas de los TTL's siempre son los emisores de un transistor, la corriente fluye al ser conectadas a cero volts, sale del circuito, es por esto que, cuando las características dicen "corriente --

de entrada para un cero lógico", ésta siempre es negativa, y es igual a  $-1.6 \text{ mA}$  máximo (Fig. 1.14). La corriente que entra al circuito cuando una de sus entradas es conectada a VCC, se garantiza que es menor de  $40 \mu\text{A}$ .

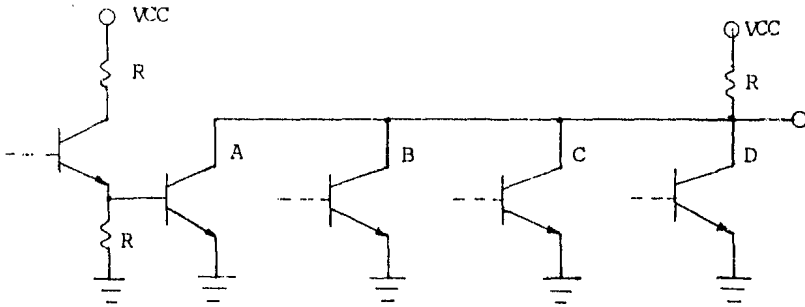


Figura 1.11. Compuertas cualesquiera con colector abierto.

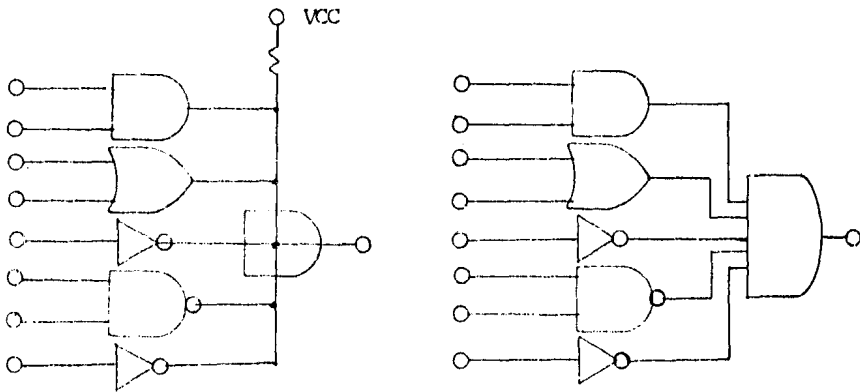


Figura 1.12. a) "Wired AND" b) Equivalente si no se tuvieran salidas con colector abierto.

La configuración totem-pole cuenta con un transistor a la salida cuando se trata de drenar corriente (sink current) y tiene una resistencia

cia en serie con un transistor y un diodo cuando se trata de proporcionar corriente (source current) como el de la Fig. 1.15. Esto hace que la capacidad de corriente de drain y de source sean diferentes. Los TTL --- pueden drenar hasta 16 mA, pero sólo pueden proporcionar unos 500  $\mu$ A -- (esto depende del circuito TTL en especial).

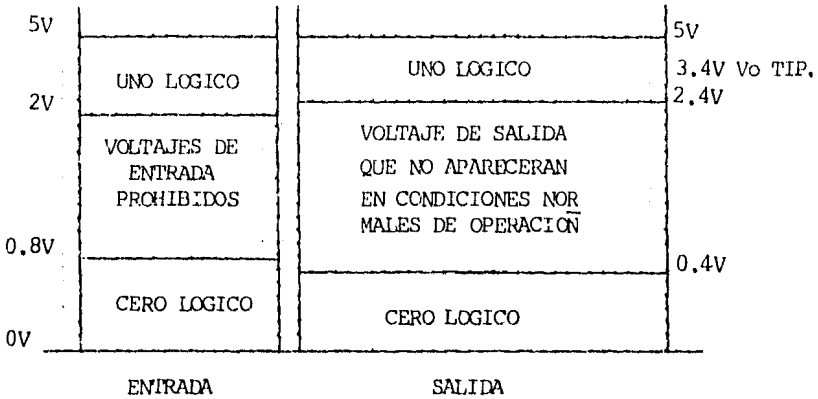


Figura 1.13. Valores de voltajes en las entradas y salidas de un C.I. TTL.

Viéndolo de otra forma: si colocamos una resistencia entre  $V_{cc}$  y - la salida de un TTL, la mínima resistencia que se puede colocar será:

Una resistencia más pequeña hará que el transistor de abajo se sobrecargue, porque pasará más corriente de la permitida.

Si colocamos una resistencia entre la salida y tierra de un TTL -- que tenga una "source current" de 500  $\mu$ A, la mínima resistencia que podremos tener será el voltaje de salida para un uno lógico (3.4V típico) - dividido entre máxima corriente de source:

$$\frac{V_o}{I} = \frac{3.4V}{500\mu A} = 6,800 \Omega \text{ (ohms)}$$

La resistencia encontrada es sólo una aproximación, porque ni

el  $V_o$  es 3.4 ni la  $I$  es de  $50\mu A$ , ya que estos valores dependen del CI en concreto. Simplemente es bueno para referencia y tratar de no colocar una resistencia menor.

El tiempo de propagación en la mayoría de las compuertas es de unos 15 ns. Para circuitos de funciones especiales como: contadores, convertidores de códigos, selectores de datos, etc., siempre son mayores.

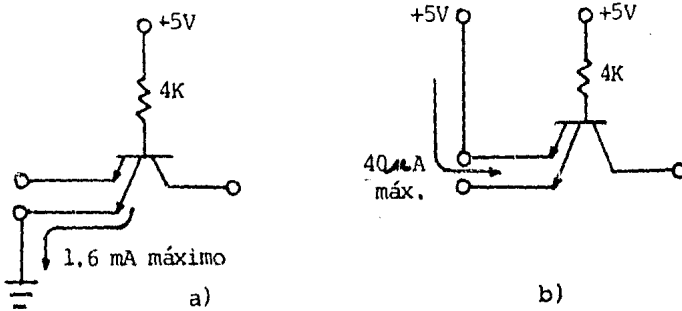


Figura 1.14. La corriente de entrada a un circuito TTL es de  $-1.6$  mA para un cero lógico(a) y de  $40\mu A$  para un uno lógico (b).

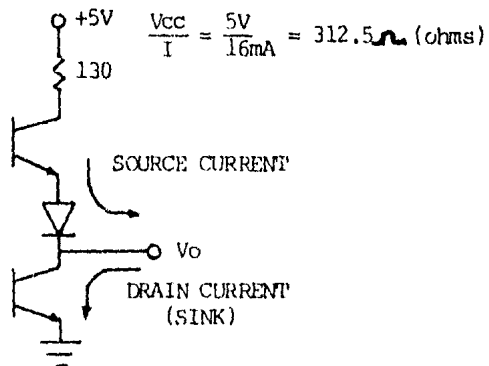


Figura 1.15. La corriente de Source que puede proporcionar un TTL, no es igual a la corriente de sink.

El fan-out de los TTL es normalmente de 10; esto quiere decir que el transistor de salida acepta que pasen hasta 16mA (cada entrada necesita drenar 1.6mA cuando se conecta a un cero lógico). Hay circuitos - que aceptan hasta 30 cargas y son llamados "BUFFERS" en algunas ocasiones y "DRIVERS" en otras. El término Buffer es también usado para describir la amplificación de una señal la cual debe manejar muchas cargas. Los CIs que manejan grandes cargas son llamados Buffers o Drivers.

#### 1.4.1 COMPUERTAS LOGICAS.

Dentro de las compuertas lógicas podemos distinguir tres tipos. - Las que tienen un totem-pole a la salida, los de colector abierto y los buffers. Los primeros dos se discutieron anteriormente. Casi todos los buffers son de colector abierto y los hay principalmente de tres tipos: los que aceptan sólo 15 volts entre colector y emisor, los de 30V y --- aquellos que sólo aceptan 5V pero el fan-out es mayor de lo normal. Hay Buffers que cuentan con una resistencia de pull-up a la salida y hay -- otros que ya se están usando y que su salida es de tri-state (tres estados).

Estos circuitos son usados como interfaz entre los TTL y los MOS o cualquier otro tipo de cargas que ocupen más voltaje para sus niveles lógicos. El fan-out suele ser de 25 o 30 cargas (40 o 48mA) para los -- tres tipos.

La tabla 1.3 es una guía útil para encontrar fácilmente el número del circuito que se necesita. En ella están las compuertas comerciales-TTL 7400 con la configuración de salida, el número de compuertas por -- paquete, cuántas entradas tiene la compuerta, etc. Al final del capítulo se da una lista de circuitos por orden numérico.

Dado que al principio de los CIs todavía no se estandarizaba el - tipo de lógica que se iba a usar, algunos fabricantes les dan nombres a las compuertas lógicas como "Quadruple 2-input Positive Nand Gate" --- (7400). Lo de "positive" se refiere a que la lógica que se usa es positiva (un cero es el menor voltaje y un uno es el mayor voltaje).

Dentro de las compuertas que se estudiarán en este punto, encontramos varios grupos que se han dividido como sigue: NANDS, ANDS, NORS, ORS, Inversores y buffers (Tabla 1.3).



Hay varias compuertas NANDS e inversores que en realidad son buffers, pero se han puesto junto con su función especial, es decir, con todas las compuertas NANDS. Independientemente del tipo de salida que tengan, se ha hecho un sólo grupo. Lo mismo para los inversores. Como buffers sólo se han agrupado aquellos circuitos que no realizan ninguna función lógica.

En la columna que dice "Voltaje de salida en Volts" cuando aparece "2.4 min" se refiere a que, como su configuración de salida es un totem-pole, el mínimo voltaje que el fabricante asegura es 2.4 Cuando son de colector abierto pero sólo se deben usar para hacer compuertas alambradas ("WIRED AND"), entonces su máximo voltaje a la salida es de 5 volts. Cuando las compuertas se pueden usar como buffers, los voltajes y corrientes permitidos en el transistor de salida son mayores.

La columna de "corriente máxima de consumo en mA" es la máxima corriente que el circuito integrado le pide a la fuente Vcc con todas sus alidas en cero. Este dato difiere de fabricante a fabricante y debe servirnos únicamente para comparar un circuito integrado con otro desde el punto de vista de consumo de corriente. Lo recomendable para determinar el consumo total en mA del circuito final, es armar el prototipo y medir la corriente que proporciona Vss en el peor de los casos. El calcular la corriente máxima de consumo es muy tardado y es un trabajo medio inútil, ya que de cualquier manera se debe armar el prototipo en el laboratorio, antes de diseñar la fuente de voltaje.

El tiempo de propagación también difiere entre fabricante y fabricante. Este dato nos sirve para comparar la velocidad de respuesta entre unas configuraciones y otras.

TABLA 1.3 GUIA DE SELECCION DE COMPUERTAS LOGICAS POR ORDEN DE FUNCION, CONFIGURACION A LA SALIDA Y NUMERO DE ENTRADAS POR COMPUERTA.

DESCRIPCION	NUMERO	CONFIGURACION A LA SALIDA. (1)	ENTRADAS POR COMPUERTA.	COMPUERTAS POR PAQUETE.	FAN-OUT.	VOLTAJE DE SALIDA EN VOLTS.	CORRIENTE DE CONSUMO MAXIMA EN mA.	TIEMPO DE PROPAGACION EN ns.
NAND	7400	T.P.	2	4	10	2.4 min.	22	10

NAND	7410	T.P.	3	3	10	2,4 mín.	16	10
NAND	7420	T.P.	4	2	10	2,4 mín.	11	10
NAND	7430	T.P.	8	1	10	2,4 mín.	6	10
NAND	7437	T.P.	2	4	30	2,4 mín.	54	11
NAND	7440	T.P.	4	2	30	2,4 mín.	27	11
NAND	7401(2)	O.C.	2	4	10	5 máx.	22	22
NAND	7403(2)	O.C.	2	4	10	5 máx.	22	22
NAND	7426	O.C.	2	4	10	15 máx.	22	14
NAND	7438(2)	O.C.	2	4	30	5 máx.	54	13
NAND	7439(2)	O.C.	2	4	30	5 máx.	54	13
AND	7408	T.P.	2	4	10	2,4 mín.	26	15
AND	7411	T.P.	3	3	10	2,4 mín.	20	15
AND	7421	T.P.	4	2	10	2,4 mín.	13	15
AND	7409	O.C.	2	4	10	5 máx.	26	19
NOR	7402	T.P.	2	4	10	2,4 mín.	27	10
NOR	7427	T.P.	3	3	10	2,4 mín.	26	9
OR	7432	T.P.	2	4	10	2,4 mín.	38	12
INVERSOR	7404	T.P.	1	6	10	2,4 mín.	33	10
INVERSOR	7405	O.C.	1	6	10	2,4 mín.	33	22
INVERSOR	7406	O.C.	1	6	25	30 máx.	42	13
INVERSOR	7416	O.C.	1	6	25	15 máx.	42	13
INVERSOR	74366	T.S.	1	6	20	2,4 mín.	77	11
INVERSOR	74368	T.S.	1	6	20	2,4 mín.	77	11
BUFFER	7407	O.C.	1	6	25	30 máx.	41	13
BUFFER	7417	O.C.	1	6	25	15 máx.	41	13
BUFFER	74125	T.S.	1	4	10	2,4 mín.	54	10
BUFFER	74126	T.S.	1	4	10	2,4 mín.	62	10
BUFFER	74365	T.S.	1	6	20	2,4 mín.	85	12
BUFFER	74367	T.S.	1	6	20	2,4 mín.	85	12

NOTA 1.-- T.P. = Totem Pole; O.C. = Open Collector; T.S. = Tri-state.

NOTA 2.-- El 7401 y el 7403, así como el 7438 y el 7439 son funcionalmente iguales, sólo difieren en el arreglo de las patas de sa-

lida.

#### 1.4.1.a.- COMPUERTAS LOGICAS MAS ELABORADAS.

Mencionaremos solamente tres. La 7425 (Dual 4-Input NOR Gate --- with Strobe) que viene a ser una compuerta NOR doble, de cuatro entradas con un control adicional (Strobe). Si el strobe es uno, se comporta como una NOR de cuatro entradas. Si el strobe es cero, la salida es uno independientemente de las entradas.

El CI. 7451 se llama, en inglés "Dual 2-Wide, 2-Input-AND-OR-INVERT Gates". Es un circuito que tiene la posibilidad de implementar la función  $F = \overline{AB+CD}$  dos veces, (es doble).

El CI 7454 realiza la función  $U = \overline{AB+CD+IJ+KL}$  y le llaman "4 Wide - AND-OR-INVERT Gates".

Cuando se tienen funciones lógicas a implementar, como las que se resuelven estos tres CI's es recomendable usarlos para que el número -- total de paquetes usados sea menor, pero los tres son fácilmente sustituibles por compuertas elementales de las que se vieron anteriormente.

#### 1.4.1.b.- COMPUERTAS EXPANDIBLES.

Hay comercialmente tres tipos de CI's que tienen la posibilidad - de admitir más entradas, haciendo uso de un "Expander".

Por ejemplo, el CI 7423 tiene un par de compuertas NOR como el -- 7425, a excepción de que una de las dos tiene posibilidades de expan--- derse usando el CI 7460.

A la compuerta 7460 se le llama "EXPANDER" y a la 7423 "Expandable, Dual 4-Input NOR Gates with Strobe".

Otro CI expandible es el 7450 llamado "Expandable Dual 2-Wide, -- 2-Input, AND-OR-INVERT Gates". Este circuito es similar al 7451 con la diferencia de que éste se puede usar el expander (7460) implementando - la función:

$$U = \overline{AB+CD+EFGH}$$

El CI 7453 es similar al 7454. Al igual que los anteriores puede- expanderse si se usa junto con el 7460.

#### 1.4.2.- MULTIPLEXORES (SELECTORES DE DATOS).

Un multiplexor realiza la operación indicada en la Fig. 1.16, donde disponemos de una serie de líneas de entrada, y una sola línea de salida. La posición del conmutador de posición múltiple que es contralable-determinará la señal de entrada que aparezca en la línea de salida. Un multiplexor, consta de varias entradas y una salida, y mediante un mecanismo de elección, una entrada se transfiere a la salida.

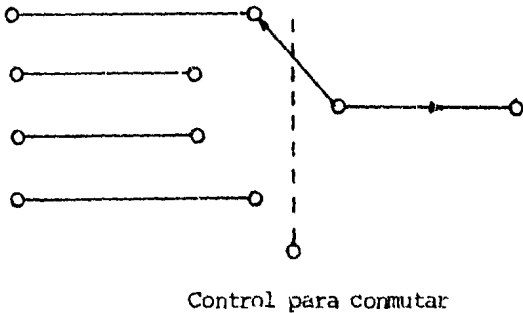


Figura 1.16. Función de un multiplexor (selector).

Los selectores de datos se usan para transformar información que se tiene en paralelo a información en serie. Con un contador binario en las señales de control, podemos transmitir la información que aparece en 8 líneas, por una sola. El receptor puede ser un demultiplexor o un Shift Register para después sacar la información en paralelo otra vez. Esto se puede usar para mandar información por teléfono de una computadora a otra, estando físicamente lejos una de la otra, Fig. 1.17.

El Primero de los selectores de datos que veremos es el 74150. Es un "Data Selector/Multiplexor" de 16 líneas a una; vienen en un paquete de 24 patas distribuidas de la siguiente forma: 16 son para los datos de entrada, una para Vcc, otra para tierra. Cuatro son para las entradas de control. Una es la salida y hay una más llamada "strobe". Si strobe es uno lógico, la salida es también uno sin importar en qué estado se encuentren todas las demás entradas. Si Strobe es cero lógico, la salida será igual al inverso de aquella entrada que marquen las lí-

neas de control.

El 74151 es muy similar al 74150, sólo que es de 8 líneas a una.- La salida tiene posibilidades de que se tome invertida (tiene dos salidas, F y  $\bar{F}$ ) y también cuenta con una línea de strobe.

El 74152 es igual al 151 pero no cuenta ni con la salida complementaria ni con las líneas de strobe. El 74251 se comporta exactamente igual que el 74151 a excepción que sus salidas son tri-state y son controladas por el strobe. Si éste es uno lógico, la salida es de alta impedancia, si es cero lógico, la salida o su complemento serán dependientes de las entradas y líneas de control.

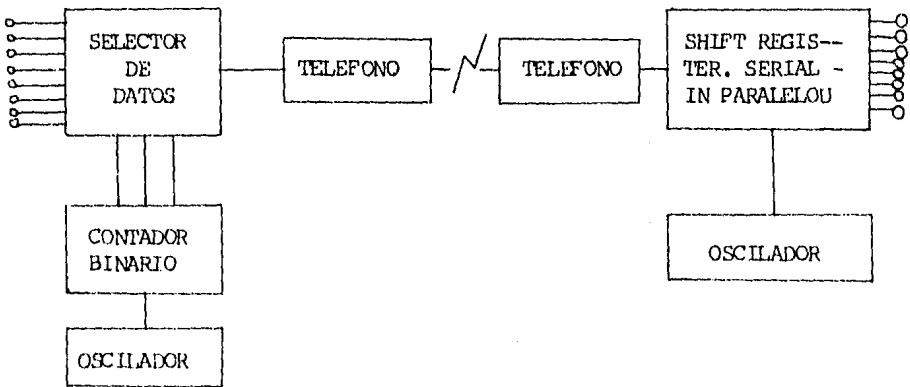


Figura 1.17. Una aplicación de los selectores de datos.

El Primero de los selectores de datos que veremos es el 74150. Es un "Data Selector/Multiplexor" de 16 líneas a una; vienen en un paquete de 24 patas distribuidas de la siguiente forma: 16 son para los datos - de entrada, una para  $V_{cc}$ , otra para tierra. Cuatro son para las entradas de control. Una es la salida y hay una más llamada "strobe". Si --- strobe es uno lógico, la salida es también uno sin importar en qué estado se encuentren todas las demás entradas. Si Strobe es cero lógico, - la salida será igual al inverso de aquella entrada que marquen las líneas de control.

El 74151 es muy similar al 74150, sólo que es de 8 líneas a una.- La salida tiene posibilidades de que se tome invertida (tiene dos salidas, F y  $\bar{F}$ ) y también cuenta con una línea de strobe.

El 74152 es igual al 151 pero no cuenta ni con la salida complementaria ni con las líneas de strobe. El 74251 se comporta exactamente igual que el 74151 a excepción que sus salidas son tri-state y son controladas por el strobe. Si éste es uno lógico, la salida es de alta impedancia, si es cero lógico, la salida o su complemento serán dependientes de las entradas y líneas de control.

El 74153 es un selector de datos de cuatro líneas a una y es doble (Dual 4-Line to 1-Line Data Selector/Multiplexers), tiene cuatro líneas de control, son las mismas para los dos selectores.

El 74157 es un selector cuádruple de dos líneas a una línea (4 -- polos, 2 tiros) y en inglés se llama "Quad 2-Line to 1-Line Data Selector/Multiplexer". Tiene 8 entradas de datos y 4 salidas, una señal de control y un strobe. El 158 es idéntico al 157, a excepción de que las salidas son complementadas.

Hasta aquí hemos analizado los selectores de datos o multiplexores (multiplexers), los equivalentes con interruptores de estos circuitos están mostrados en la Fig. 1.18.

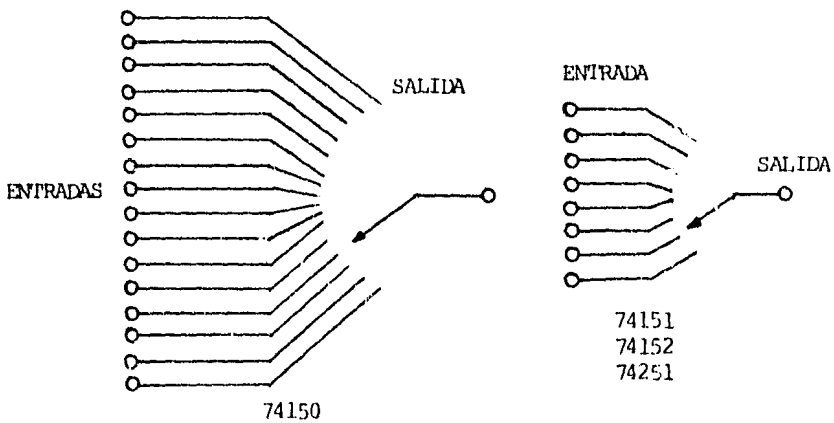


Figura 1.18. Equivalentes con interruptores de los CI's vistos - como selectores de datos/multiplexers.

### 1.4.3.- DECODIFICADORES.

Estos circuitos, lo que hacen es pasar una información de un código a otro, los hay de binario a decimal, de exceso 3 a decimal, de BCD a 7 segmentos, etc. Algunos de ellos aparte de ser decodificadores son drivers, los transistores de salida tienen suficiente capacidad de voltaje y/o corriente para manejar displays u otros elementos. En la tabla 1.4 se muestran algunos de los códigos BCD empleados más en sistemas digitales.

NUMERO BINARIO	8421	AIKEN 2421	EXCESO 3	GRAY EXCESO 3
0 0 0 0	0	0		
0 0 0 1	1	1		
0 0 1 0	2	2		0
0 0 1 1	3	3	0	
0 1 0 0	4	4	1	4
0 1 0 1	5		2	3
0 1 1 0	6		3	1
0 1 1 1	7		4	2
1 0 0 0	8		5	
1 0 0 1	9		6	
1 0 1 0			7	9
1 0 1 1		5	8	
1 1 0 0		6	9	5
1 1 0 1		7		6
1 1 1 0		8		8
1 1 1 1		9		7

Tabla 1.4. Códigos BCD de cuatro bits.

#### 1.4.3.a.- DECODIFICADORES DE BINARIO A DECIMAL.

El 74154 decodifica la información binaria que se le presente a -

las entradas (DCBA) a decimal. Cuenta con 4 líneas para las entradas en binario y 16 para la salida en decimal.

En los circuitos llamados "Multiplexers" tenemos varias líneas de entrada y una de salida. El "DEMULTIPLEXER" es similar pero, en este caso tenemos una línea de entrada y varias de salida. El 74154 se usa también como demultiplexer, aparte de ser un decodificador de binario a decimal.

El 74155 y 156 son iguales excepto que el primero tiene a la salida un totem-pole y el segundo es de colector abierto. En inglés se llama "Dual 2-Line to 4-Line Decoders/Demultiplexers". Se puede usar este circuito de cuatro formas diferentes:

- a).- Decodificador de dos líneas a cuatro.
- b).- Demultiplexer de 2 polos 4 tiros.
- c).- Decodificador de tres líneas a ocho.
- d).- Demultiplexer de un polo 8 tiros.

#### 1.4.3.b.- DECODIFICADORES DE BCD A DECIMAL.

El 7442 es parecido al 74154 usado como decodificador. El 74154 tenía cuatro líneas de entrada y 16 de salida. El 7442 tiene las mismas líneas de entrada, pero sólo 10 de salida. Convierte BCD a decimal. Este circuito no cuenta con señales de strobe.

El 7445 y el 74145 son iguales al 7442, pero son de colector abierto. El primero acepta 30 volts a la salida y el segundo 15 volts. Los dos tienen capacidad de 80 mA a la salida.

El 7441 es parecido al 7442, en realidad desempeña la misma función lógica, pero además de decodificador de BCD a decimal, es driver. Está especialmente diseñado para manejar los displays de Burroughs llamados "Nixie". Las salidas cuentan con transistores de algo voltaje; entre emisor y colector se pueden aplicar hasta 70 volts. Lógicamente son de colector abierto. La tabla de verdad es la misma que para el 7442. El 7441 es igual al 74141 pero éste consume la mitad de la potencia.

#### 1.4.3.c.- DECODIFICADORES DE BCD A 7 SEGMENTOS.

Se estudiarán primero dos de ellos que se diferencian básicamente-



por su salida; el 7446 es de colector abierto y tiene capacidad de 30 - volts, el 7447 también de colector abierto pero sólo maneja 15 volts.

La función lógica es igual para los dos CI's. Tiene cuatro entradas para los datos en BCD y siete salidas para los segmentos. Se usan - para manejar displays de ánodo común y tienen capacidad de drenar hasta 40 mA.

El 7448 es muy parecido a los dos anteriores. Su configuración es la misma (son compatibles pata por pata). La diferencia entre uno y --- otro es que los anteriores son colector abierto y éste tiene interna--- mente una resistencia "pull-up" de 2 Kohms.

#### 1.4.3.d.- DECODIFICADORES DE DECIMAL A BINARIO.

A estos circuitos en inglés se les llama "Encoders". El 74147 --- convierte una señal codificada en decimal a binario, dándole prioridad al bit mayor, por ejemplo si en las líneas de entrada (9) está activa - la línea del 6 y del 8 la salida será un ocho binario, si estuvieran -- activadas las líneas 1, 5 y 6, la salida sería un seis binario. En in-- glés se llama a este circuito "10-Line to 4-Line Priority Encoder". En realidad son sólo 9 líneas de entrada, pero cuando todas están activa-- das, el circuito "entiende" que la línea cero está activada, así que -- implícitamente está la línea cero.

El 74148 es llamado "8-Line to 3-Line Priority Encoder". Cuenta - con ocho líneas de entrada, del cero al siete y además tienen un "Ena-- ble Input" a fin de poder expandirse. Su salidas son cinco, tres de --- ellas son para dar la señal binaria y las otras dos son para usarse --- cuando se colocan varios circuitos para expansión.

#### 1.4.3.e.- DECODIFICADORES ESPECIALES.

Un par de circuitos muy interesantes son el 74184 y el 74185. El - primero es un convertidor de BCD a binario de 6 bits (variable lógica - de dos estados 0 ó 1) y el segundo es un convertidor de binario a BCD.- Estos CI's salieron de la programación de una ROM x 8 (CI 7+88). Ambos - CI's se pueden usar en cascada para convertir un número de n bits de -- BCD, a binario (74184 o de binario a BCD (74185). Los dos tienen cinco - líneas de entrada (25=32) y ocho de salida además de un strobe.

El 7443 es un convertidor de exceso tres a decimal y 7444 convierte exceso tres Gray a decimal. Los dos cuentan con cuatro líneas de entrada y diez de salida y vienen en paquetes de 16 patas.

1.4.4.- CIRCUITOS ARITMETICOS.

El 7486 es un CI que tiene cuatro compuertas OR exclusivas. Implementa la función  $F = \overline{A}B + A\overline{B} = A \oplus B$ .

Dentro de los circuitos más elaborados que nos sirven para efectuar sumas lógicas, encontramos el 7480. Es un sumador completo de un bit. A las entradas hay unas ANDs de tal forma que el sumador completo hará la operación  $\overline{A_1}A_2A_3 + \overline{B_1}B_2B_3 + C_n$  dando por resultado la suma de  $A+B+C_n$  en . En la salida tenemos tres líneas. El carry, la suma y su complemento.

Una característica propia de este circuito es que sus entradas son DTL y sus salidas TTL. Además, las salidas de  $A_1A_2$  y  $B_1B_2$  pueden usarse como entradas si  $A_1, A_2, B_1$  y  $B_2$  se mandan a Vcc.

El 7483 "4-Bit Binary Full Adder (look Ahead Carry)" es un sumador completo de 4 bits. Cuenta con nueve entradas que son la palabra A:  $A_1, A_2, A_3$  y  $A_4$ ; la palabra B:  $B_1, B_2, B_3$  y  $B_4$ ; y Carry de entrada ( $C_n$ ). Las salidas son  $S_1, S_2, S_3, S_4$  y el Carry de salida ( $C_4$ ). Su diagrama de bloques está en la Fig. 1.19.

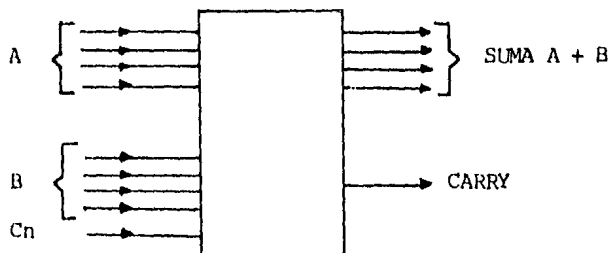


Figura 1.19. Diagrama de bloques del sumador de 4 bits 7483.

#### 1.4.5.- SCHMITT TRIGGERS (CUADRADOR DE ONDAS).

Se cuenta con cuatro configuraciones. El 7413 que es un "Dual --- NAND Schmitt trigger" (dos compuertas NAND de cuatro entradas cada una); el 74132 que es un "Quad NAND Schmitt trigger" (cuatro compuertas NAND de dos entradas cada una); el 74232 "Quad NOR Schmitt trigger" y el --- 7414 "HEX Schmitt trigger" (seis inversores).

Cuando se tienen señales que sus tiempos de subida o bajada son muy lentos es necesario usar un schmitt trigger ya que los contadores o flip-flops no responden cuando se cambian de cero a uno o de uno a cero muy lentamente.

#### 1.4.6.- COMPARADORES DE MAGNITUD.

El 7485 es un comparador de cuatro bits. Cuenta con tres salidas:  $A > B$ ,  $A < B$  y  $A = B$ , dando un uno en la correspondiente. Se puede poner en serie para comparar dos números de N bits, usando tres entradas adicionales que se conectan con las salidas del comparador de la derecha.

#### 1.4.7.- CHECADORES Y GENERADORES DE PARIDAD.

Cuando se transmiten datos en binario de una parte a otra es probable que en el trayecto se generen ruidos haciendo que un determinado bit que es un cero lógico al ser transmitido se reciba como un uno lógico en el receptor, acarreando errores que pueden causar muchos problemas. Para evitar, o cuando menos detectar este tipo de errores, se inventaron los generadores de paridad y por consiguiente, los chequeadores de la misma.

El CI 74180 es un "8 bit ODD/EVEN parity generator checker. Tiene posibilidades de generar y chequear paridad ya sea par o impar.

#### 1.4.8.- FLIP-FLOPS Y LATCH.

Un elemento de memoria es cualquier dispositivo que puede almacenar niveles lógicos 0 y 1 (bits), de tal manera que un nivel o un grupo de niveles pueden ser accedidos posteriormente o alterados al recibir otros niveles.

Un "Flip-Flop" es un circuito que puede almacenar dos estados ló-

gicos (1 y 0) y tiene la capacidad de cambiar de un estado a otro con la aplicación de una señal de control (reloj), y permanecer en ese estado después de recibir la señal de control.

Existen un número de tipos diferentes de flip-flops como se mencionarán en este punto. El más simple de ellos, Fig. 1.20, tiene las siguientes líneas:

- 1).- Una entrada de dato.
- 2).- Una entrada de la señal de reloj.
- 3).- Una salida Q que presenta el estado almacenado.
- 4).- Una salida  $\bar{Q}$  con el estado opuesto a Q.
- 5).- Una entrada (PRESENT) para obligar al flip-flop que su salida Q tome el estado 1.
- 6).- Una entrada (CLEAR) para obligar al flip-flop que su salida Q tome el estado 0.

Este circuito se conoce como "D flip-flop". El estado de la Q es el mismo estado que tenía la entrada cuando recibió el último pulso de reloj en su entrada CLK. Por ejemplo, si la entrada D tiene el nivel lógico 1 cuando su entrada CLK recibe un pulso de reloj, la salida toma el nivel 1.

Frecuentemente la información se debe conservar durante cierto tiempo antes de que un dispositivo periférico o una microcomputadora lo reciba. En situaciones como ésta se deben utilizar ciertos dispositivos conocidos como "Latch". El latch es un circuito (tales como los flip-flops) que se utilizan para almacenar estados lógicos.

La diferencia entre uno y otro está en que el latch pasa la información a la salida si el clock está en uno, y el D flip-flop para la información únicamente cuando el reloj cambia de estado, de cero a uno en unos tipos (positive going) o de uno a cero (negative going). Esto quiere decir que en un tipo D la entrada puede estar cambiando aún cuando el reloj esté en uno y la salida no cambia.

A continuación se da una clasificación de flip-flops.

a) J-K. El 7470 contiene un flip-flop J-K. La entrada de J-K es por medio de una compuerta AND de tal forma que  $J=J_1.J_2.J_3$  y  $K=K_1.K_2.K_3$ . También cuenta con entradas para Preset y Clear que son independientes de los estados de J y K, pero deben ser aplicados sólo cuando -

el reloj está en Cero.

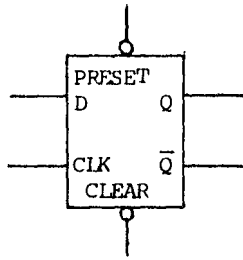


Figura 1.20. D flip-flop.

El 7472 es prácticamente igual al 7470, sólo que  $J=J_1.J_2.J_3$  y  $K=K_1.K_2.K_3$ . Son pata por pata compatibles uno con otro.

El 7473 es doble. J y K entran directas al flip-flop y no cuentan con Preset. Tiene 14 patas.

El 7476 es como el 7473 pero sí cuenta con Preset. Tiene 16 patas.

El 74107 es funcionalmente igual al 7473 pero el arreglo de las patas de salida es distinto.

El 74109 se parece al 7476. Este cuenta con preset y clear, y viene también en 16 patas. Las diferencias fundamentales son que tiene inversor en la entrada de K y cambia de estado con la subida del reloj. Si se juntan J y K se convierte en un D flip-flop.

En la tabla 1.5 aparece una guía de selección de JK's.

b) Tipo D y Latches. El 7474 es un D flip-flop doble con preset y clear que pasa la información a la salida cuando el reloj va a uno. Tiene salidas Q y  $\bar{Q}$ .

El 7475 es llamado "Quadruple Biestable Latch" tiene cuatro flip-flops del tipo latch. Para la información de la entrada a la salida cuando el reloj está en uno. Si el reloj pasa a cero, la salida se queda con los datos que tenía cuando estaba en uno. No cuenta con preset ni clear. Viene en 14 patas y para los 4 latches sólo tiene dos entradas de reloj.

El 74100 es parecido al 7475, sólo que éste tiene 8 latches. Se -

llama "Dual 4 bit Bilestable Latch" y está dividido en dos secciones de 4 latches cada una. Cada sección es manejada con una señal de reloj y no tiene salidas complementarias. Viene en 24 patas, y al igual que el 7475 la salida (Q) seguirá los datos de la entrada "D" mientras el reloj esté en uno lógico y se quedará (Q) con la información que tenga -- (D) cuando el reloj baje a cero.

NUMERO	J	K	CLR	PR	No. PATAS	No. Ffs	CAMBIO
7470	$J1J2\bar{J3}$	$K1K2\bar{K3}$	SI	SI	14	1	↑
7472	$J1J2J3$	$K1K2K3$	SI	SI	14	1	↓
7473	J	K	SI	NO	14	2	↓
7476	J	K	SI	SI	16	2	↓
74107	J	K	SI	NO	14	2	↓
74109	J	$\bar{K}$	SI	SI	16	2	↑

↑ = El cambio de las salidas ocurre en la transición del reloj de 0 a 1.

↓ = El cambio de las salidas ocurre en la transición del reloj de 1 a 0.

Tabla 1.5. Tabla de selección de los flip-flops JK de la serie 7400.

El 74173 tiene la salida de tres-estados y cuenta con cuatro ---- flip-flops del tipo D. Es manejado con una sola línea de reloj y una -- línea para clear independiente del reloj. Los datos de entrada, sólo -- pueden pasar si las dos líneas de "Data Enable Inputs" están en cero y la salida estará en el estado de alta impedancia cuando en alguna o en las dos líneas de "Output Control" haya un uno lógico. El cambio de las salidas Q se realiza cuando el reloj pase de 0 a 1.

El 74174 tiene 6 flip-flops con reloj y clear comunes para todos.

Sólo cuenta con la salida Q y viene en 16 patas; la salida cambia cuando el reloj pasa de 0 a 1.

El 74175 es similar al 74174 pero sólo tiene 4 flip-flops y cuenta con Q y  $\bar{Q}$  a la salida. Viene en 16 patas. También éste es "Positive-going-edge-triggered".

En la tabla 1.6 se muestra una guía de selección de flip-flops -- tipo D y Latches.

NUMERO	TIPO	# DE Ffs.	# DE PATAS	CAMBIO	Q	CLR	PR	# DE ENTRADAS	
								DE CLK	DE CLR
7474	D	2	14	↑	SI	SI	NO	2	2
7475	LATCH	4	16	↓	SI	NO	NO	2	-
74100	LATCH	8	24	↓	NO	NO	NO	2	-
74173	D	4	16	↑	NO	SI	NO	1	1
74174	D	6	16	↑	NO	SI	NO	1	1
74175	D	4	16	↑	SI	SI	NO	1	1

Tabla 1.6. Selección de los flip-flops tipo D y Latches.

c) S-R. El único flip-flop del tipo S-R es el 74279. Contiene ocho compuertas NAND alambradas de tal forma que hacen 4 S-R's. Dos de ellos tienen entradas dobles de S haciendo que se vaya a uno con cualquiera de las dos que pase a cero. Como en todos los S-R, las dos entradas no pueden estar en cero al mismo tiempo.

#### 1.4.9.- CONTADORES.

Los contadores son sistemas secuenciales con una sola entrada de impulsos, cuyo estado interno en cada instante representa el número de impulsos que se han aplicado a aquélla. El número de estados internos ha de ser igual al máximo número de impulsos que se desean contar, más uno, que será el estado inicial el cual indicará la ausencia de impulsos.

Para la realización de los contadores se utilizan elementos bistables sincronizados por flancos, que poseen dos estados internos (bistable D o latch).

A continuación se describen cada uno de los contadores.

7490. Es un contador de décadas. Tiene cuatro flip-flops arreglados de tal forma que puede contar del cero al 9 y regresar al cero - (contador en BCD), o se puede usar para dividir la frecuencia de entrada entre 10.

Tiene dos partes, una de ellas es el flip-flop A y la otra los -- flipflops B, C y D. Para usarse como un contador BCD, los pulsos de -- reloj deben entrar por el "A input" y la salida A se debe conectar al -- "B input". Las salidas ABCD corresponden a la cuenta en BCD de los pulsos del reloj, donde  $A=2^0$ ;  $B=2^1$ ;  $C=2^2$  y  $D=2^3$ .

Si se ocupa tener separadamente un divisor entre 5 y un divisor - entre 2, se puede lograr si se toman como partes separadas las dos secciones del 7490.

Tiene dos líneas de Reset comunes para los cuatro flip-flops para mandar a ceros la salida. Si cualquiera de ellas está en cero, el contador cuenta normalmente, pero si ambas están en uno, entonces la salida será cero. De la misma forma cuenta con dos líneas para mandar la -- salida a 9 en BCD (1001). Cualquiera de ellas que esté en cero, el contador cuenta. Si los cuatro resets están en unos, la salida será 9 en - BCD.

El 7492 tiene también cuatro flip-flops, pero arreglados de tal - forma que cuentan del cero al once y regresan a cero. Se usan para di-- vidir una frecuencia entre 12 o también se pueden usar para dividir entre 6 y entre dos separadamente. Cuenta con dos líneas para reset a cero, las cuales deben estar en uno para que las salidas estén en cero -- lógico. No tiene resets a nueves. La frecuencia de operación es de 18 - megahertz y consume 32 mA.

El 7493 viene a ser igual que los anteriores, pero éste cuenta -- hasta 15, es decir, es un contador binario. Este circuito tampoco tiene resets a 9, y sus valores de frecuencia máxima de operación y consumo - de corriente son iguales a los dos anteriores (18 MHz y 32 mA). La salida se verá incrementada cuando el reloj pase de uno a cero.

En la serie 7400 se tienen cuatro contadores muy parecidos. Son - el 74160, 74161, 74162 y 74163. Estos contadores son sincrónicos, es de-- cir, todas las salidas cambian cuando el reloj cambia de (0 a 1). Tie--



nen la característica de que se pueden programar para empezar en un número determinado. Su máxima frecuencia de operación es de 35 MHz y el consumo de 63 mA (valores típicos que dependen del fabricante).

Una versión entre el 7490 y el 74160 es el 74176. El diagrama de bloques del 74176 es igual al del 7490 pero tiene la posibilidad de programarse como el 74160. El 176 BCD y el 177 es binario, parecido al 7493 y al 74161. La frecuencia de operación anda en 25 MHz. y el consumo en unos 35 mA.

Funcionalmente el 74176 es igual al 74196 y el 74177 es igual al 74197. Unos son sustitutos exactos de los otros en la mayoría de las aplicaciones.

El 74190 es un contador BCD, programable, sincrónico (el reloj va a los cuatro flip-flops directamente) diferente a los vistos anteriormente porque puede contar hacia arriba (0, 1, ..., 9, 0) o hacia abajo (2, 1, 0, 9, 8, 7, ...). Si la entrada "UP/DOWN" está en Cero cuenta hacia arriba. Si está en uno, cuenta hacia abajo.

Su frecuencia típica de operación es de 20 MHz. y el consumo de corriente anda en los 65 mA.

El 74191 es igual al 74190 sólo que es un contador binario, es decir, cuenta del 0 al 15. Tiene las mismas patas de entrada y de salida que el 190. Los pulsos de RCC y M se dan en la transición de 15 a 0 o de 0 a 15.

El 74192 es un contador en BCD, programable, sincrónico, puede contar hacia arriba o hacia abajo, tiene clear pero no enable y también dos entradas de reloj, una de ellas para que cuente hacia arriba y otra para contar hacia abajo, (nunca deben estar los dos en cero). El contador cambia de estado cuando el reloj (hacia arriba o hacia abajo) pasa de cero a uno.

El 74193 es la versión en binario del 74192. Todo es igual excepto que cuenta del 0 al 15 en vez de contar del 0 al 9. Los pulsos de CA se dan en la transición del 15 al 0 y el pulso en BO se obtiene cuando pasa del 0 al 15.

La tabla 1.7 nos muestra la selección de contadores de la familia 7400.

NUMERO	TIPO	CLR	PROGRAM.	UP/DOWN	ENABLE	RELOJ	CLR	CAM
7490	BCD	SI	A NUEVE	NO	0	2	2	↓
7492	12	SI	NO	NO	0	2	2	↓
7493	BINARIO	SI	NO	NO	0	2	2	↓
74160	BCD	SI	SI	NO	2	1	1	↑
74161	BINARIO	SI	SI	NO	2	1	1	↑
74162	BCD	SI	SI	NO	2	1	1	↑
74163	BINARIO	SI	SI	NO	2	1	1	↑
74176/196	BCD	SI	SI	NO	0	2	1	↑
74177/197	BINARIO	SI	SI	NO	0	2	1	↑
74190	BCD	NO	SI	SI	1	1	1	↑
74191	BINARIO	NO	SI	SI	1	1	1	↑
74192	BCD	SI	SI	SI	0	2	1	↑
74193	BINARIO	SI	SI	SI	0	2	1	↑

Tabla 1.7. Selección de contadores.

#### 1.4.A.- SHIFT-REGISTERS. (REGISTROS DE CORRIMIENTO).

Un Shift-Register (ShR) es un grupo de flip-flops JK o D alambrados en cascada y se usan para almacenar información o para desplazarla de un lado a otro y para otras muchas aplicaciones.

El 7491 es un SR de 8 bits. La información entra en serie y sale en serie (Serial in-Serial out) es el más sencillo de todos los de la serie 7400. Los datos de entrada pasan a través de una compuerta AND de tal forma que por cualquiera de las dos líneas (A, B) se le puede meter información. Tiene salida complementaria.

El 74164 es muy parecido al 7491, es de 8 bits y su diagrama es prácticamente igual pero tiene salidas en cada uno de los flip-flops -- siendo entonces un "Serial in-Paralel Out"; pero si no se usa la salida de todos los flip-flops, sino únicamente la salida de la derecha, entonces es también un "Serial in-Serial out". Cuenta con un Clear independiente del reloj y directo a todos los flip-flops.

El 7496 es un SR de 5 bits muy parecido al 7491. Es "Serial in---

Serial out", la entrada no pasa por una AND como en el 91, y clear común e independiente del reloj y además se puede restablecer cuando el "Preset Enable" (PE) se encuentran en uno. Si primero se aplica el clear y luego se presetea se puede convertir en un "Paralel in-Serial out". Hay que tener cuidado en que este circuito no es precisamente un SR de la forma "Paralel in-Paralel out" porque si en la entrada de un preset hay un uno y se quiere escribir un cero no se puede hacer, a menos que se le aplique el clear antes. La salida cambia cuando el reloj pasa de cero a uno.

El 7494 es un SR de 4 bits. Casi igual en funcionamiento al 7496. Para que funcione como "Paralel in-Serial out" debe aplicarse un pulso (uno lógico) en CLEAR ya que con los presets sólo se puede cambiar de cero a uno, pero no al revés. A diferencia con el 7496, éste tiene 8 líneas de entrada para los presets pudiendo seleccionar cuatro de ellas con las dos entradas para este fin. Manteniendo los presets en cero, se le puede usar como "Serial in-Serial out" dándole los datos por la línea marcada como ENTRADA. La salida cambia de estado cuando CLK pasa de cero a uno. Tanto dos presets como clear deben estar en cero lógico cuando el reloj pasa de cero a uno.

El 74165 sí es realmente un "paralel in-Serial out" de 8 bits. Los datos que se apliquen en las 8 entradas de datos son grabados en los flip-flops, si la entrada marcada con "LOAD" está en cero. Si todas las entradas de los datos se ponen a cero, "LOAD" sirve como clear y entonces por la pata marcada con "Serial in" entran los datos para que se pueda usar como un ShR de 8 bits "Serial in-Serial out".

El 74166 hace lo mismo que el 165 pero las entradas están arregladas de otra forma y además, la entrada "LOAD" esta sincronizada con el reloj. Si se usa como "Paralel in-Serial out", "LOAD" debe estar en cero y cuando el reloj pasa de cero a uno, los ocho flip-flops se cargan con los datos de las ocho líneas de entrada y cada vez que el reloj pasa de cero a uno, los datos se recorren hacia la derecha. Si "LOAD" está en uno, entonces no pueden pasar los datos de las señales en paralelo, pero sí, los datos que aparezcan en la entrada "Serial in".

El 74195 es casi igual al 166. Es de cuatro bits. Sólo tiene una entrada de reloj que pasa la información cuando va de cero a uno. Tiene

un clear que puede actuar independiente de cualquier entrada. La única diferencia funcional respecto al 166 es que tiene dos entradas para los datos cuando se usa "serial in-Serial out". La entrada real de datos en serie depende del estado del primer flip-flop. Si el flip-flop A tiene un uno, entra el dato que esté en K. Si el flip-flop tiene un cero, entonces entra el dato que esté en J.

El 74199 difiere en dos cosas con el 166. La primera es que el 199 es de 8 bits y la segunda es que tiene dos entradas para el reloj a través de una OR. Para su funcionamiento léase lo escrito del 74166 y 74195.

Los SR vistos son unidireccionales, es decir, la información siempre se desplaza hacia la derecha. Dentro de la serie 7400 hay 3 llamados bidireccionales, la información se puede desplazar hacia la derecha o hacia la izquierda. El 7495 es uno de estos. Tiene tres modos de información:

a) Si "MODE CONTROL" (MC) está en cero. Se comporta como un SR "Serial in-Serial out" de cuatro bits. La información cambia cuando el reloj 1 pasa de 1 a 0. La entrada es por A y la salida es por QD. En este caso, el reloj 2 y las entradas en paralelo quedan inhibidas.

b) Si MC está en uno. Se comporta como un SR "Paralel in-Paralel-out". Los datos entran en paralelo por ABC y D y pasan a las salidas QA, QB, QC y QD cuando el reloj 2 pasa de uno a cero. En este caso las entradas para datos en serie quedan inhibidas

c) Si MC está en uno y las salidas se conectan a las entradas de la izquierda (QD a C, QC a B, etc) y la entrada en serie es por D, entonces se comporta como un "Serial in-Paralel out" con desplazamiento a la izquierda cada vez que el reloj 2 pasa de uno a cero.

El 74194 es más fácil de usar que el 7495. También es de 4 bits, pero sólo tiene una entrada de reloj que actúa cuando cambia de cero a uno. Cuenta también con un clear y dos entradas para el "MODE CONTROL"- S1 y S2. Su manera de operar es como sigue:

a) Si  $S_1=0$  y  $S_2=0$ , inhibe el reloj. No hace nada.

b) Si  $S_1=1$  y  $S_2=1$ , se comporta como un SR "Paralel in-Paralel-out". Las entradas A, B, C y D pasan a las salidas cuando el reloj cambia de 0 a 1. (las entradas en serie se inhiben).

c) Si  $S_1=0$  y  $S_2=1$  la información que está en la entrada R se desplaza a la derecha cada vez que el reloj pasa de 0 a 1. Las entradas en paralelo quedan inhibidas (A, B, C y D).

d) Si  $S_1=1$  y  $S_2=0$ , entonces la información que está en la entrada "L" se desplaza a la izquierda cada vez que el reloj pasa de 0 a 1. Las entradas en paralelo quedan inhibidas.

El 74190 sólo difiere del 194 en que es un SR de 8 bits. Todo lo mencionado para el 194 se aplica al 198.

La tabla 1.8 nos muestra la selección de Shift Registers que nos puede servir de guía para escoger el más adecuado en el diseño que se esté haciendo.

NUMERO	TIPO	# BITS	# SAL.	ENT. DE CLK	CLR	BIDIREC.	# PATA'S
7491	SISO	8	2	1	NO	NO	14
7494	PIPO-SISO	4	1	1	SI	NO	16
7495	PIPO-SIPO	4	2	2	NO	SI	14
7496	PIPO-SIP	5	5	1	SI	NO	16
74164	SIPO	8	8	1	SI	NO	14
74165	PISO-SISO	8	2	2	NO	NO	16
74166	PISO-SISO	8	1	2	SI	NO	16
74194	PIPO-SIPO	4	4	1	SI	SI	16
74195	PIPO-SIPO	4	5	1	SI	NO	16
74198	PIPO-SIPO	8	8	1	SI	SI	24
74199	PIPO-SIPO	8	8	2	SI	NO	24

Tabla 1.8. Selección de los Shift-Registers de la familia 7400.

NOTA: SISO = Serial In-Serial Out.

SIPO = Serial In-Paralel Out.

PISO = Paralel In-Serial Out.

PIPO = Paralel In-Paralel Out.

Cuando un Shift-Register tiene su salida en paralelo, implícitamente la tiene en serie también, tomando la salida del flip-flop de la derecha.

Una vez que se ha hecho una descripción de una manera muy general

de una gran mayoría de CI's de la serie 7400, a continuación se da una lista de estos en orden numérico.

#### CIRCUITOS INTEGRADOS TTL.

NUMERO	DESCRIPCION
7400	Cuatro NANDs de dos entradas.
7401	Cuatro NANDs de dos entradas de colector abierto.
7402	Cuatro NORs de dos entradas.
7403	Cuatro NANDs de dos entradas de colector abierto.
7404	Seis inversores.
7405	Seis inversores con colector abierto.
7406	Seis inversores de colector abierto 30 V.
7407	Seis Buffers de colector abierto 30 V.
7408	Cuatro ANDs de dos entradas.
7409	Cuatro ANDs de dos entradas y colector abierto.
7410	Tres NANDs de tres entradas.
7411	Tres ANDs de tres entradas.
7413	Schmitt Trigger. Dos NANDs de cuatro entradas.
7414	Schmitt Trigger. Seis inversores.
7416	Seis inversores.
7417	Seis inversores.
7420	Dos NANDs de cuatro entradas.
7421	Dos ANDs de cuatro entradas.
7423	Dos NOR de cuatro entradas expandibles.
7425	Dos NORs de cuatro entradas con strobe.
7426	Cuatro NANDs de dos entradas con colector abierto.
7427	NOR triple de tres entradas.
7430	Una NAND de ocho entradas.
7432	OR cuádruple de dos entradas.
7437	Cuatro NANDs de dos entradas.
7438	Cuatro NANDs de dos entradas con colector abierto.
7439	Cuatro NANDs de dos entradas con colector abierto.
7440	Dos NANDs de cuatro entradas.
7441	Decodificador de binario a decimal. Colector ---- abierto 70 V.
7442	Decodificador BCD a decimal.
7443	Decodificador 4 a 10 líneas (exceso 3 a decimal).
7444	Decodificador de exceso tres Gray a decimal.
7445	Decodificador/Driver BCD a decimal.
7446	Decodificador/Driver 7 segmentos. 30 V.
7447	Decodificador/Driver 7 segmentos. 15 V.
7448	Decodificador BCD a 7 segmentos Pull-up.
7450	Dos ANDs de dos entradas a una OR. Doble y Expandi-

ble,

7451 Dos ANDs de dos entradas a una OR. Doble Cuatro -- ANDs de dos entradas a una OR, Expandible.

7453 Cuatro ANDs de dos entradas a una OR, Expandible.

7454 Cuatro ANDs de dos entradas a una OR.

7460 Expander.

7470 Flip-flop J-K.

7472 Flip-flop J-K.

7473 Flip-flop J-K doble.

7474 Flip-flop D doble.

7375 Cuatro latches.

7476 Flip-flop J-K doble.

7480 Sumador de un bit.

7483 Sumador de cuatro bits.

7485 Comparador de magnitud de cuatro bits.

7486 Cuatro OR EXCLUSIVAS de dos entradas.

7488 ROM de 256 bits.

7489 RAM de 64 bits (16X4).

7490 Contador de décadas.

7491 Shift Register de ocho bits.

7492 Contador divisor por 12.

7493 Contador binario.

7494 Shift Register de cuatro bits.

7495 Shift Register de cuatro bits bidireccional.

7496 Shift Register de cinco bits.

74100 Dos latches de cuatro bits.

74107 Flip-flop J-K doble.

74109 Flip-flop J-K doble.

74121 Multivibrador monoestable.

74122 Multivibrador monoestable con clear.

74123 Multivibrador monoestable con clear, Doble.

74125 Cuatro buffers, Tri-state.

74126 Cuatro buffers, Tri-state.

74132 Schmitt trigger, Cuatro NANDs de dos entradas.

74141 Decodificador/Driver BCD a decimal, Colector ---- abierto, 70 V.

74145 Decodificador/Driver BCD a decimal, Colector ---- abierto, 15 V.

74147 Decodificador de decimal a binario con prioridad.- 8 líneas a 3.

74148 Decodificador de decimal a binario con prioridad.- 10 líneas a 4.

74150 Selector de datos-Multiplexer, 16 líneas a una.

74151 Selector de datos-Multiplexer, 8 líneas a una.

74152 Selector de datos-Multiplexer, 8 líneas a una

74153 Selector de datos-Multiplexer, 4 líneas a una Doble.

74154 Decodificador-Demultiplexer, 4 líneas a 16.

74155 Decodificador-Demultiplexer, 2 líneas a una, Doble.

74156 Decodificador-Demultiplexer, 2 líneas a 4, Doble.

74157 Selector de datos-Multiplexer, 2 líneas a 1, Cua--

74158           druple.  
           Selector de datos-Multiplexer. 2 líneas a l. Cua--  
           druple.  
 74160           Contador en BCD programable.  
 74161           Contador en binario programable.  
 74162           Contador en BCD programable.  
 74163           Contador en binario programable.  
 74164           Shift Register de ocho bits.  
 74165           Shift Register de ocho bits.  
 74166           Shift Register de ocho bits.  
 74170           RAM de 16 bits (4X4).  
 74173           Flip-flop D. Cuádruple.  
 74174           Flip-flop D. Séxtuple.  
 74175           Flip-flop D. Cuadruple.  
 74176           Contador en BCD programable.  
 74177           Contador en binario programable.  
 74180           Generador o checador de paridad.  
 74181           Unidad aritmética de alta velocidad.  
 74182           Generador de carga de alta veolocidad.  
 74184           Decodificador de BCD a binario.  
 74185           Decodificador de binario a BCD.  
 74187           ROM.  
 74190           Contador en BCD programable y bidireccional.  
 74191           Contador en binario programable y bidireccional.  
 74192           Contador en BCD programable y bidireccional.  
 74193           Contador en binario programable y bidireccional.  
 74194           Shift Register de cuatro bits bidireccional.  
 74195           Shift Register de cuatro bits.  
 74196           Contador en BCD programable.  
 74197           Contador en binario programable.  
 74198           Shift Register de ocho bits bidireccional.  
 74199           Shift Register de ocho bits.  
 74232           Schmitt Trigger. Cuatro NORs de dos entradas.  
 74251           Selector de datos-multiplexer, 16 líneas a l. ---  
           Tres-estados.  
 74279           Flip-flop SR. Cuadruple.  
 74365           Seis buffers.  
 74366           Seis buffers inversores.  
 74367           Seis buffers.  
 74368           Seis buffers inversores.



## CAPITULO 2

### DEFINICION DEL PROBADOR DE CIRCUITOS INTEGRADOS DIGITALES.

#### 2.1 GENERALIDADES.

Una vez que se ha descrito la funcionalidad de algunos circuitos integrados digitales (ver tabla al final del capítulo 1) y antes de explicar la circuitería (Hardware) y el programa (Software) necesarios para el funcionamiento del Probador, se hará una pequeña descripción de lo que se pretende haga el sistema y el aspecto físico que tendría. Además a través de este capítulo se definen términos y dispositivos usados en el diseño, permitiendo así una fácil comprensión y entendimiento del mismo.

#### 2.2 DEFINICION DEL PROBADOR.

El Probador de Circuitos Integrados Digitales lo podemos definir como un sistema digital diseñado para probar CIs digitales TTL de la serie 7400 de 14 y 16 patas multi-entradas/multi-salidas (en base a una pequeña microcomputadora), usando combinaciones lógicas. Por combinación lógica mediremos que los CIs de estados lógicos en sus entradas producen valores lógicos estáticos en su patas de salida. Primeros ejemplos de lógica combinatorial son las compuertas estandar: AND, NAND, OR y NOR. El término combinatorial es usado para distinguir básicamente su función lógica estática a las propiedades más dinamicas de sistemas lógicos secuenciales como: FLIP-FLOPS y CONTADORES. Esto quiere decir, que de acuerdo a la configuración del circuito a probar, se tendrán el número de entradas y salidas y por consiguiente las combinaciones. Por combinación lógica se conocerá el estado lógico que se pretende a su salida y con esto el resultado del mismo.

El principal y único objetivo contemplado fué el siguiente:

DISEÑAR UN INSTRUMENTO QUE PERMITA PROBAR, SI UN CIRCUITO INTEGRADO DIGITAL ESTÁ O NO DAÑADO. EVITANDO ASÍ EMPLEAR TIEMPO PREVIO A LAS PRÁCTICAS DE ELECTRONICA PARA LA PRUEBA DEL ESTADO DE LOS COMPONENTES. PERMITIENDO CON ESTO, ELIMINAR CIOS DIGITALES DEFECTUOSOS DE CIENITOS COMPRENDIDOS EN EL LABORATORIO.

### 2.3 DESCRIPCION DEL PROBADOR.

La Fig. 2.1 muestra el aspecto externo físico del Probador de Circuitos Integrados Digitales, como se puede observar se distinguen 4 elementos en conjunto que son: el interruptor de encendido y el de reinicialización; los exhibidores luminosos (displays alfanuméricos, matriz de 7X5); el teclado y 7 bases (wire wrap sockets) de 14 y 16 patas.

1) Interruptores de encendido y reinicialización.- Con el interruptor encendido se energiza el sistema del Probador para su funcionamiento. El interruptor de reinicialización (reset) da la señal a todo el sistema para reiniciar su operación.

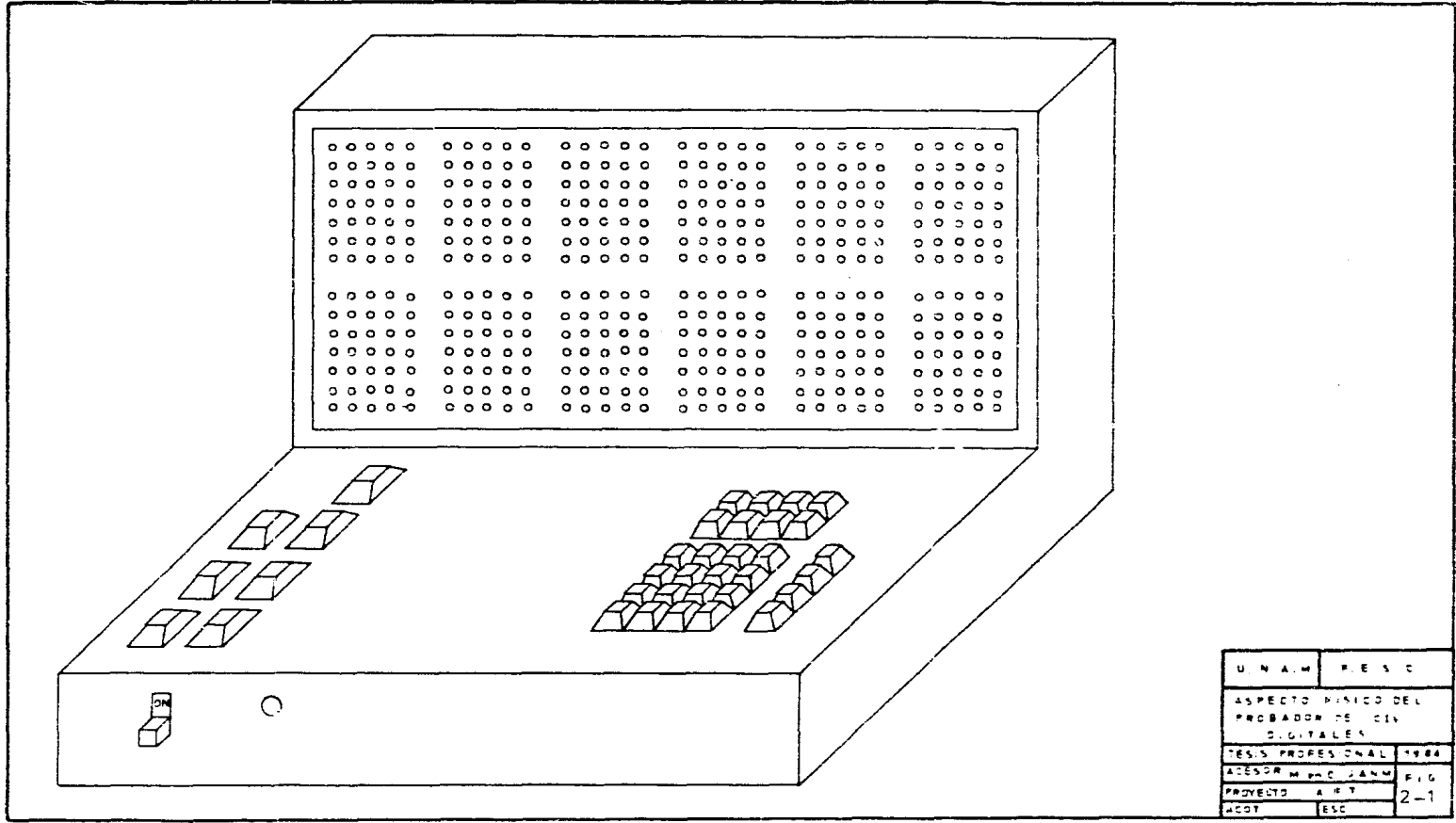
2) Exhibidores luminosos.- Son para mostrar o exhibir el mensaje al probar un CI, así como para mostrar el número del CI a probar deseado por el usuario.

3) El teclado.- Por medio de este, se introducen los comandos necesarios del programa para la prueba del CI, así como los números de los mismos.

4) Bases de prueba (wire wrap sockets).- Estas están compuestas de dos grupos; una se compone de 4 y es para los CIs de 14 patas únicamente, el segundo grupo la componen 3 y son para los CIs de 16 patas, formando un total de 7 bases. Además estos están ordenado de acuerdo a su alimentación, se recomienda al lector para mayor información ver el Manual del Usuario.

El orden que se ha seguido en la descripción de los elementos externos del Probador que se acaban de mencionar, o sea el orden numérico, es el mismo que tienen asignados en la Fig. 2.1.

### 2.4 VISTA MACROSCOPICA DEL PROBADOR.



U N A M	F E S C
ASPECTO FÍSICO DEL PROBADOR DE CIV DIGITALES	
TESIS PROFESIONAL 1984	
ASESOR M. en C. JANM	FIG
PROYECTO A. I. T.	2-1
ACOT.	ESC

Una de las principales ventajas de usar, un microprocesador, es la de permitir gran flexibilidad en los sistemas en los que son incluida y con la finalidad de aprovechar esta flexibilidad se diseñó el lenguaje de comunicación Usuario/Probador de CIs Digitales, esto es, el Probador realiza sus operaciones por medio de un programa. Permitiendo con esto, la posibilidad de ampliar sus aplicaciones, es decir, el número de CIs a probar en base a las necesidades del laboratorio.

El manejo del Probador viene siendo relativamente sencillo, únicamente se requiere la comunicación hombre-Probador para realizar la prueba de los CIs deseados.

Además de las partes externas ya mencionadas en el apartado 2.4 es conveniente dar un programa general del Probador internamente. El probador cuenta con una pequeña microcomputadora internamente, la cual esta compuesta por una sola tarjeta de circuito impreso llamada: Z80 Starter Kit. Los componentes más importantes del "Starter" son:

- a) Un Z80-CPU
- b) Un Z80-PIO
- c) Un Z80-CTC (no usada en el Probador)
- d) Un kiloByte de RAM
- e) Dos kiloByte de ROM
- f) Tres exhibidores dobles de 7 segmentos.
- g) Un cristal oscilador de 3.9936 MHz
- h) Un teclado

Por otra parte y para darle mayor visualidad al Probador es utilizada una Tarjeta de Control de exhibidores, la cual controla los 12 exhibidores mostrados en la Fig. 2.1, la finalidad de estos exhibidores es mostrar el número completo del CI a probar, los cuales son introducidos por medio del teclado, una vez que se ha insertado en la base correspondiente el CI a probar, así como el resultado obtenido en la prueba.

NOTA.- Ver capítulo 3 y 4 para mayor información.

## 2.5 INDICADORES ALFANUMERICOS.

Con el uso creciente de sistemas digitales en la vida común del -

hombre, han aparecido en los últimos años indicadores alfanuméricos --- (que hace uso de letras y números) de distintas tecnologías. Debido a - la gran variedad de estas es necesario conocer sus principales caracte- rísticas a fin de hacer una elección adecuada del indicador para un --- problema de instrumentación específica.

A continuación se describe un tipo de indicadores.

### 2.5.1 INDICADORES CON DIODOS EMISORES DE LUZ (LIGHT-EMITTING-DIODE).

Un diodo emisor de luz, comúnmente llamado LED por su nombre en - inglés, es un dispositivo semiconductor que emite luz cuando se polari- za en forma directa. Forma los segmentos de la mayoría de los exhibido- res.

El símbolo de un LED es mostrado en la Fig. 2.2. Los extremos -- del LED -anodo y cátodo son los mismos que un diodo ordinario (ver Fig. 2.3).

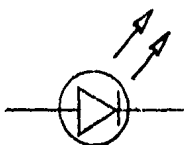


Figura 2.2. Símbolo del LED (Light-Emitting Diodo).

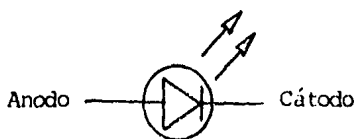


Figura 2.3. Anodo y cátodo del LED.

Cuando trabajamos con LEDs, es necesario que una resistencia limi- tadora de corriente sea provista para restringir la corriente a través-

del LED a un nivel que no lo queme y también su intensidad luminica sea suficiente para ser observado sin ningún problema. Un simple circuito - de un LED con una resistencia limitadora de 330 ohm es mostrada en la - Fig. 2.4.

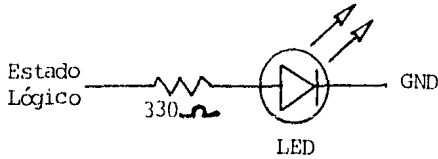
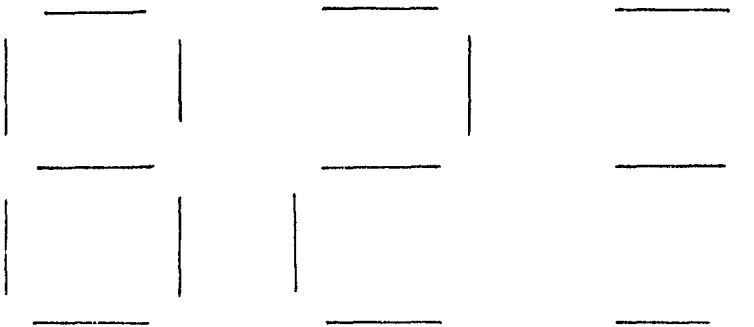


Figura 2.4. Circuito controlador del LED.

Para producir caracteres numéricos se emplean segmentos, mientras que para producir letras y números deben emplearse arreglos de puntos - (LEDs) de 5X7. Ver Fig. 2.5.

0 0 0 0 0	X X X X X	0 X X X 0
0 0 0 0 0	X 0 0 0 X	X 0 0 0 X
0 0 0 0 0	X 0 0 0 X	0 0 0 0 X
0 0 0 0 0	X X X X X	0 X X X 0
0 0 0 0 0	X 0 0 0 X	X 0 0 0 0
0 0 0 0 0	X 0 0 0 X	X 0 0 0 0
0 0 0 0 0	X 0 0 0 X	X X X X X

(a) ALFANUMERICO



(b) NUMERICO

Figura 2.5. Ejemplo de formación de caracteres con segmentos y - puntos (matrices).

Los indicadores con LEDs se fabrican generalmente para caracteres numéricos con 7 segmentos y para caracteres alfanuméricos con arreglos de puntos de 5x7.

Los caracteres con LED con por lo general de pequeñas dimensiones. Sin embargo sus pequeñas dimensiones, los hacen adecuados para equipo portátil.

La mayor parte de los LEDs emiten luz roja, amarilla o verde.

Los LEDs tienen un umbral (valor de una magnitud física a partir del cual empiezan a ser observables sus efectos) de conducción bien definidos lo cual hace que los indicadores que se formen con ellos sean apropiados para trabajar en tiempo compartido. El umbral de conducción de un LED está alrededor de 1.7 V, lo que los hace compatibles con la mayoría de circuitos lógicos integrados.

Es también posible hacer LEDs en barras y otras formas que pueden ser usados para otro tipo de exhibidores. El dispositivo o exhibidor luminoso (display, en inglés) cuenta con siete patas terminales por donde reciben la información codificada del número decimal a presentar, estos también se encuentran disponibles en dos configuraciones como el de la Fig. 2.7a configuración cátodo común y Fig. 2.7b configuración ánodo común.

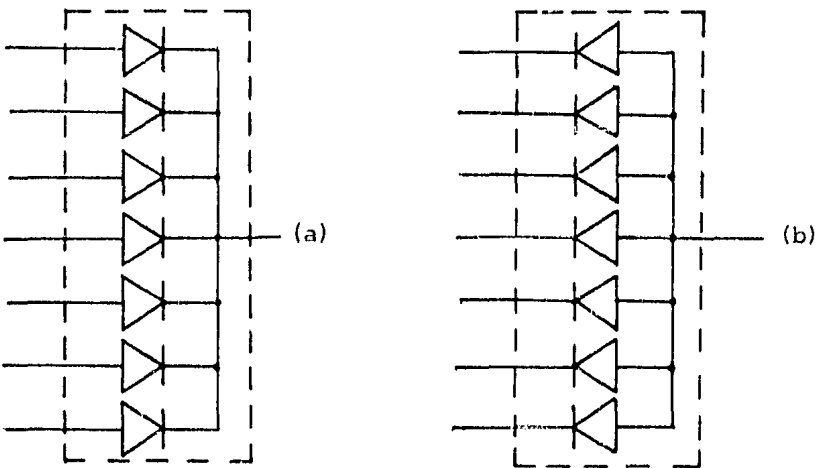


Figura 2.6. Configuración interna del Display.

El exhibidor de 7 segmentos mostrado en la Fig. 2.8 es un ejemplo de siete LEDs dormados dentro de una barra. Cada una de las barras es -- asociada con una letra, a, b, c, d, e, f y g. Usted puede fácilmente -- ver que cualquier número hexadecimal puede ser obtenido por iluminación a cierta combinación de segmentos. La correspondencia entre dígitos hexadecimales y combinación de segmentos literales es como sigue:

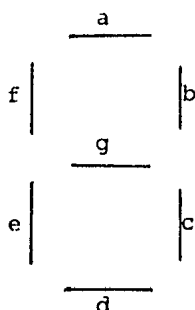


Figura 2.7. Display de 7 segmentos.

<u>Digito Hex.</u>	<u>Combinación de segmentos literales.</u>
0	a,b,c,d,e,f
1	b,c
2	a,b,d,e,g
3	a,b,c,d,g
4	b,c,f,g
5	a,c,d,f,g
6	a,c,d,e,f,g
7	a,b,c
8	a,b,c,d,e,f,g
9	a,b,c,f,g
A	a,b,c,e,f,g
b	c,d,e,f,g
C	a,d,e,f
d	b,c,d,e,g
E	a,d,e,f,g
F	a,e,f,g

Cuando una pata terminal recibe un nivel alto el segmento correspondiente se enciende. Utilizando un puerto de salida de 8 bits se puede enviar información a un display. Las salidas del puerto se pueden -- conectar con las patas terminales de la siguiente manera:



<u>Bits</u>	<u>Segmentos</u>
0	a
1	b
2	c
3	d
4	e
5	f
6	g

Con esta forma de conexión, para generar el número 0 (cero) se --  
 tienen que enviar a nivel alto las terminales del display con excepción  
 de la terminal g (bit 6 del puerto), es decir enviando el código 3FH al  
 puerto. La tabla 2.1 muestra los códigos para generar los números en el  
 display. Algunos displays requieren nivel bajo en las terminales para -  
 activarse.

DIGITO	CODIGO
0	3F
1	06
2	5B
3	4F
4	66
5	6D
6	7D
7	07
8	7F
9	6F

Tabla 2.1. Códigos en hexadecimal de los números decimales.

## 2.6 INTRODUCCION A LAS COMPUTADORAS.

Antes de iniciar con el estudio de los dispositivos de entrada y-  
 salida con los que cuenta el Probador es conveniente y de suma impor---  
 tancia, hace un análisis de lo que son las computadoras.

Al hablar de computadoras se está haciendo referencia a máquinas-electrónicas, esto es, máquinas cuyas funciones se efectúan utilizando circuitos electrónicos.

Existen dos tipos básicos de computadoras: digitales y analógicas. Aquí sólo se hace referencia a las computadoras digitales. Una computadora analógica trabaja con señales electrónicas continuas, mientras que las computadoras digitales trabajan con señales electrónicas discretas.

La principal característica de las computadoras digitales es su velocidad. Esta velocidad se logra debido a que están construidas con componentes electrónicos de alta velocidad. Estos componentes se utilizan para formar circuitos que realizan funciones más complejas y que operan en base a señales de dos niveles o estados (binario); nivel lógico 1 y nivel lógico 0. Estas máquinas operan con circuitos binarios, por lo que para operar a altas velocidades deben ejecutar las operaciones aritméticas y lógicas utilizando el sistema de números binarios.

### 2.6.1 ESTRUCTURA DE UNA COMPUTADORA DIGITAL.

La computadora digital está formada por cuatro partes principales como se muestra en la Fig. 2.8.

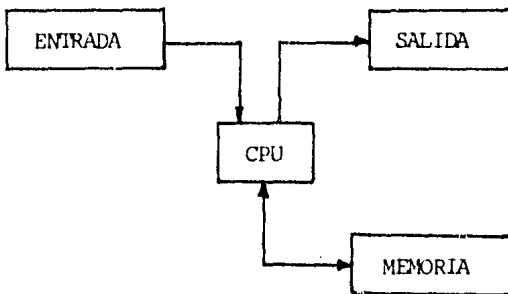


Figura 2.8. Diagrama a bloques de una computadora.

A continuación se realiza una breve descripción de cada una de --

estas partes.

### 1) UNIDAD CENTRAL DE PROCESOS (CPU).

La CPU es la parte principal de la computadora y se puede dividir por sus funciones en la UNIDAD DE CONTROL y en la UNIDAD ARITMETICA Y LOGICA (ALU, Arithmetic Logic Unit).

#### a) UNIDAD DE CONTROL.

Todas las acciones dentro de la computadora deben estar sincronizadas y seguir las instrucciones de un programa. La unidad de control recibe las instrucciones codificadas en binario desde la memoria, y decide cuando, como y que operaciones se deben ejecutar para realizar cada instrucción. Conoce cuando se termina la ejecución de una instrucción e indica cual es la que se debe ejecutar a continuación. Se considera a la Unidad de Control como el cerebro de la computadora.

#### b) UNIDAD ARITMETICA Y LOGICA.

Esta unidad es la que ejecuta realmente el trabajo de procesamiento. La ALU puede recibir datos y efectúa con ellos operaciones aritméticas, lógicas, de comparación y corrimiento, entre otras. La ALU tiene algunos registros para almacenar los datos sobre los que va a realizar las operaciones; el número exacto de estos registros depende de cada computadora en particular. El registro principal de la ALU se conoce como Acumulador. Generalmente, al comienzo de una operación, el Acumulador contiene uno de los operandos, y al final de la operación, contiene el resultado.

### 2) MEMORIAS.

Esta unidad contiene las instrucciones y los datos que se van a procesar en la CPU. La computadora digital obedece fielmente lo que le ordenan las instrucciones.

La memoria se divide, por su ubicación, en dos áreas principales, propia y auxiliar. Las memorias propias están construidas principalmente de semiconductores y núcleos magnéticos. La CPU tiene acceso directo a la memoria propia y es desde esta memoria de donde la CPU obtiene los códigos de las instrucciones. Este tipo de memoria normalmente se encuentra en el mismo gabinete.

Las memorias auxiliares son dispositivos de almacenamiento masivo de instrucciones y de datos, pero no pueden ser accedidos directamente-

por la CPU. Los programas que se almacenan en este tipo de memorias se tienen que enviar primero a la memoria propia de la computadora antes de intentar su ejecución. Entre los dispositivos mas usuales se encuentran el disco magnético, cinta magnética y diskettes.

### 3) UNIDADES DE ENTRADA.

Los dispositivos de entrada se usan para alimentar los datos necesarios para los cálculos, lo mismo que los programas que le indican a la computadora las operaciones a desarrollar. Los requerimientos de las unidades de entrada varían de una computadora a otra. Las unidades de entrada más comunes son: lectora de tarjetas, cinta de papel perforada, teclado, cinta magnética, disco magnético, diskette, tubos de rayos catódicos (CRT) y teletipo.

### 4) UNIDAD DE SALIDA.

Los resultados de las operaciones de la computadora se proporcionan al usuario por medio de las unidades de salida. Los dispositivos más comunes son: el teletipo, el CRT, cinta perforada, impresor de líneas, cinta magnética, disco magnético y diskette.

## 2.6.2 MICROPROCESADOR Y MICROCOMPUTADORA.

El Microprocesador es una Unidad Central de Proceso (CPU) implementado en un chip o circuito integrado LSI de uso general, el cual debe realizar un gran número de funciones diferentes, siendo controlado por una serie de instrucciones ordenadas secuencialmente. La mayoría de los Microprocesadores vienen implementados en un integrado de 40 pines terminales.

A la par con el desarrollo de los Microprocesadores, los fabricantes también han desarrollado otros circuitos integrados tales como memorias, interfaces de entrada/salida, decodificadores, buffers, que permiten al diseñador una gran facilidad en el desarrollo de nuevos productos y aplicaciones.

Se define como Microcomputadora a la computadora digital en la que la CPU está formada por un Microprocesador, y sus componentes auxiliares por integrados que forman la familia del Microprocesador (memoria, interfaces, decodificadores, etc.). La Fig. 2.9 ilustra un diagrama

ma de una Microcomputadora en base al microprocesador Z-80.

## 2.7 INTERFACES DE ENTRADA Y SALIDA.

Los sistemas de computación deben contar, al menos, con un dispositivo periférico de entrada que le permitan al usuario enviar información a la CPU y con dispositivo de salida para que la CPU le pueda mostrar los resultados de los procesamientos ordenados. En este capítulo se presentan algunas técnicas para interfazar una Microcomputadora en base al Z-80 con algunos dispositivos de entrada y salida.

Tradicionalmente cuando se habla de dispositivos periféricos de entrada y de salida se hace referencia a las lectoras de tarjetas, unidades de cinta magnética, CRT, teletipos, etc. Sin embargo, un circuito integrado, tales como los registros de corrimiento, contadores, memorias, etc. También pueden considerarse como dispositivos de entrada y de salida.

La CPU-Z80 cuenta con dos instrucciones para realizar la transferencia de datos con los periféricos: IN y OUT. La cantidad de periféricos de entrada y salida que el Z-80 puede manejar con estas dos instrucciones es de 512 en total. Para efectuar la transferencia de datos con la memoria la Z-80 cuenta con las instrucciones LD.

### 2.7.1 SEÑALES DE CONTROL DE TRANSFERENCIA.

La Z-80 utiliza cuatro señales para el control de la transferencia de datos entre los registros de la CPU o memoria y dispositivos periféricos o memoria. Estas señales son:

- 1)  $\overline{RD}$ .- Se habilita para indicar una lectura de datos.
- 2)  $\overline{WR}$ .- Se habilita para indicar una escritura de datos.
- 3)  $\overline{MEMRQ}$ .- (Memory Request) indica que se está haciendo una solicitud a memoria ya sea para lectura o escritura.
- 4)  $\overline{IORQ}$ .- (Input/Output Request) indica que se está haciendo una solicitud a un dispositivo de E/S ya sea para lectura o escritura.

Estas 4 señales de control se habilitan para cuatro combinaciones de acuerdo a los ciclos de máquina de leer memoria, escribir en memoria, entrada y salida.

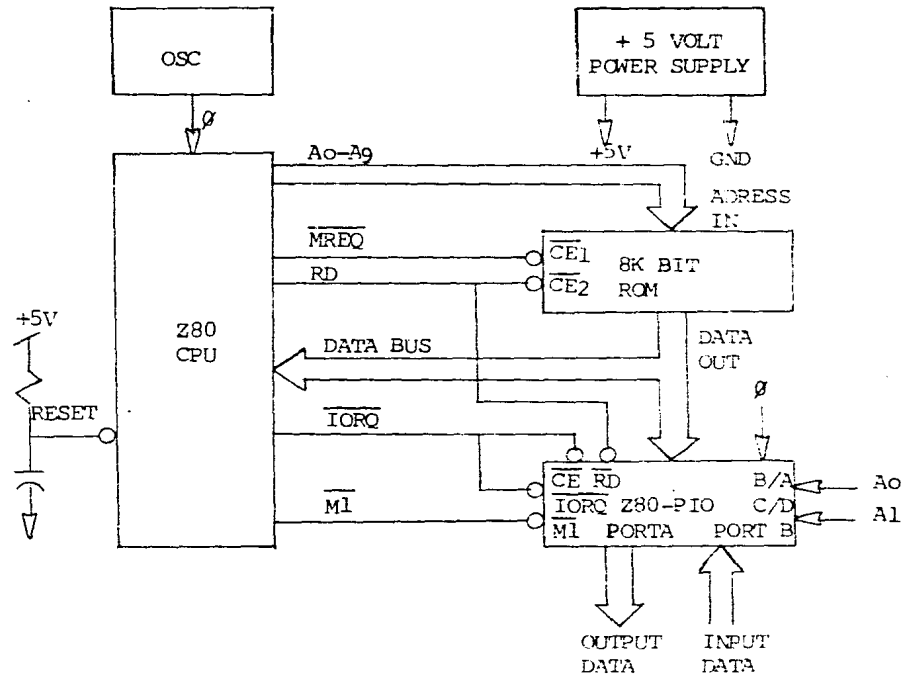


Figura 2.9. Diagrama a bloques de una microcomputadora.

- 1) Leer memoria.-  $\overline{RD}$  y  $\overline{MREQ}$ .
- 2) Escribir en memoria.-  $\overline{WR}$  y  $\overline{MREQ}$ .
- 3) Entrada.-  $\overline{RD}$  y  $\overline{IORQ}$ .
- 4) Salida.-  $\overline{WR}$  y  $\overline{IORQ}$ .

El circuito de la Fig. 2.10 realiza las combinaciones de estas --- señales para generar otras que se utilizan en el bus S-100 estandar.

## 2.7.2 INTERFAZ.

Interfaz se define como la unión de miembros de un grupo en la -- manera que pueden funcionar en una forma compatible y ordenada. Inter-- fazar con una computadora se define como la sincronización de la trans-- misión de datos digitales entre la computadora y los dispositivos ex--- ternos, incluyendo dispositivos periféricos de Entrada/Salida y memo--- rias.

Normalmente los dispositivos de E/S son mas lentos que las compu-- tadoras. De aquí que una computadora tiene que esperar a que el dispo-- sitivo se encuentre listo para recibir o para transmitir un nuevo dato.

Las computadoras se comunican con los dispositivos periféricos o-- externos por medio de sus buses de dirección, datos y control. La Fig.- 2.11 ilustra los buses de una microcomputadora con diferentes disposi-- tivos periféricos. Este modo de operación se conoce con el nombre de -- línea compartida, en donde cada dispositivo conectado a los buses del - sistema se deben comportar como si fuera el único dispositivo conectado al sistema.

Esta condición se logra con el uso de las interfaces, los cuales-- deben cumplir con los siguientes requisitos.

- a).- Decodificar el código de selección que envía la computadora-- cuando ejecuta una instrucción de entrada IN o de salida OUT y responder sólo si el código es idéntico a el.
- b).- Decodificar los códigos de los comandos que recibe de la --- computadora y generar las señales de control para efectuar-- las operaciones ordenadas. Algunos dispositivos requieren de información adicional además de las señales de control,  $\overline{RD}$ ,  $\overline{WR}$  e  $\overline{IORQ}$ . Por ejemplo, las interfaces del teletipo o el -- CRT (Tubo de Rayos Catódicos) necesitan información tales --

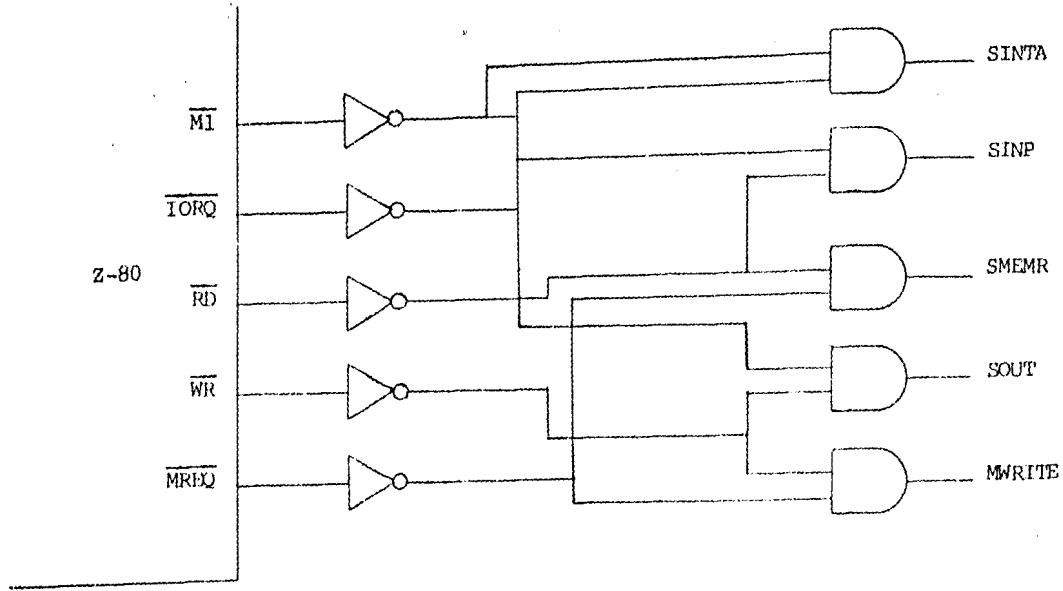


Figura 2.10. Señales de control de la memoria y periférico.



como, velocidad de transmisión, número de bit, etc. Una vez recibida la información necesaria, la interfaz realiza su función de transmisión y recepción de datos.

- c).- Enviar a la computadora la información que describe el estado del dispositivo periférico. Esta información se conoce como banderas de estado, por ejemplo, transmisor listo, receptor listo.
- d).- Efectuar la transferencia de datos entre la computadora y el dispositivo periférico a través de la propia interfaz. Debe contar con un registro de datos para la entrada y salida de datos.

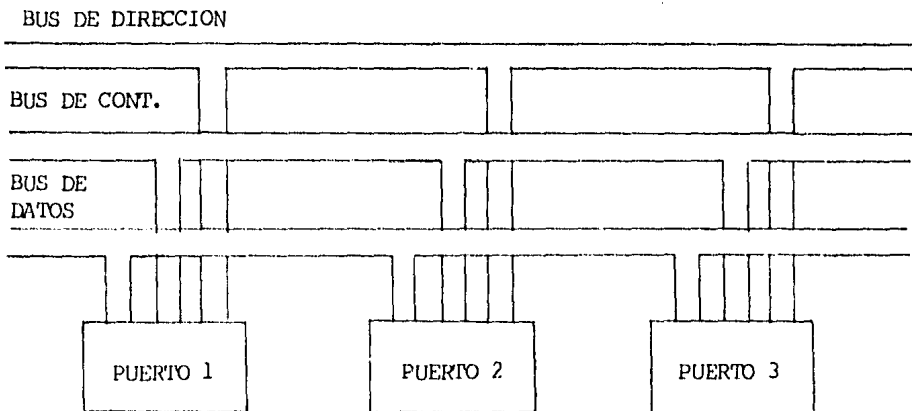


Figura 2.11. Línea compartida.

La Fig. 2.12 ilustra la relación computadora/interfaz/periférico. En el proceso de entrada el periférico deposita el dato en un registro de datos de la interfaz y después la computadora lee el registro. En el proceso de salida la computadora carga el dato en un registro de datos de la interfaz y después la interfaz transmite el dato al periférico.

Como se puede observar una interfaz puede tener uno o más registros para almacenar la información que recibe de acuerdo a las funcio--

nes del periférico. Una interfaz puede requerir un registro para recibir comandos, un registro para proporcionar las banderas de estado y un registro para la transferencia de datos entre la interfaz/periférico y para la transferencia computadora/interfaz (la transferencia de datos se realiza entre el Acumulador y los registros).

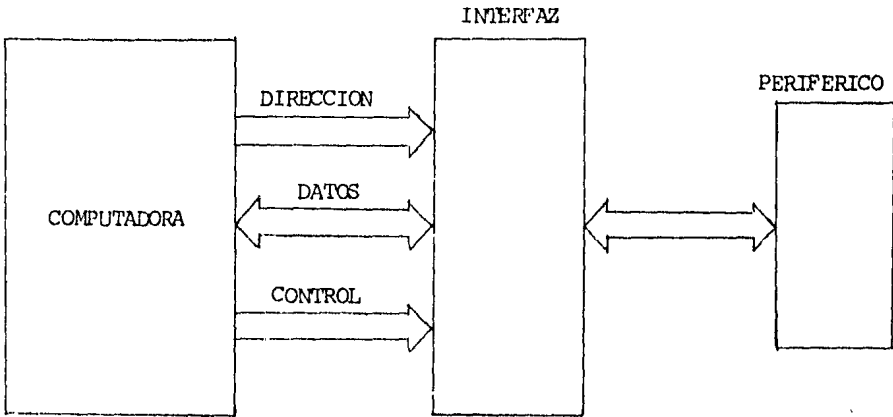


Figura 2.12. Relación computadora/interfaz/periférico.

A estos registros se les da el nombre de "Puertos". Por lo tanto, una interfaz puede contar con varios puertos dependiendo del dispositivo al que esta controlando. La interfaz de un disco tendrá más puertos que la interfaz de un teletipo, debido a que el disco requiere de más información para su funcionamiento correcto. Cada puerto debe tener su propio código de selección o número de puerto.

Es importante señalar que la computadora únicamente tiene relación con la interfaz, no tiene comunicación directa con el periférico. Es la interfaz quién tiene relación directa con el periférico.

Existen interfaces programables y no-programables. Las interfaces no programables son las interfaces cuyas características de funcionamiento se especifican por circuitos (Hardware) y las programables son -

las que por sus características de funcionamiento se especifican por -- circuitos y programación.

Cuando el programador tiene que utilizar interfaces programables, debe conocer cuales son los comandos que se deben ordenar a la interfaz para programarlo adecuadamente. Una vez definido lo anterior, el pro--- gramador o usuario tiene que preocuparse únicamente de utilizar los nú--- meros correctos de los puertos para la transmisión de datos con los pe--- riféricos.

### 2.7.3 SELECCION DEL DISPOSITIVO PERIFERICO.

Un pulso de "selección de dispositivo" es un pulso de sincroniza--- ción generado por la interfaz para sincronizar la transferencia de da--- tos entre la computadora y un dispositivo periférico de entrada o de -- salida específico. El término de "selección de dispositivo" se asocia -- con los términos de "selección de integrado" o de "habilitar integrado" que se utilizan en los integrados de memoria. Cada interfaz debe tener un selector de código de selección que le permita generar el pulso de - selección de dispositivo cada vez que la CPU envía por el bus de direc--- ción su código de selección. Al generarse el pulso de selección de dis--- positivo la interfaz queda habilitada para recibir las señales de con--- trol de la CPU para el periférico.

La Fig. 2.13 nos muestra un selector con código de selección 5AH. Cuando el valor en el bus de dirección es de 5AH la salida de la com--- puerta NAND pasa a nivel alto, indicando a la interfaz que la computa--- dora se va a comunicar con ella. Este tipo de circuito selector debe -- ser parte de la interfaz periférico.

Una forma más simple de generar los códigos de selección es usando un circuito decodificador tales como el 8205, 74LS138 y el 74154. -- Los cuáles los primeros dos son decodificadores de 3 a 8 y el tercero - de 4 a 16 respectivamente. La Fig. 2.14 ilustra un circuito para selec--- cionar 8 dispositivos. Este circuito decodificador puede ser parte de - la microcomputadora y las interfaces.

## 2.8 DISPOSITIVOS PERIFERICOS DE ENTRADA.

Cuando se conecta o conectamos dispositivos periféricos a la CPU,

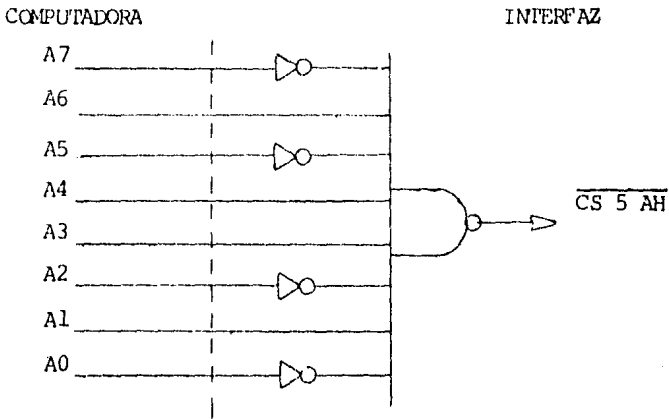


Figura 2.13. Selector del código 5AH.

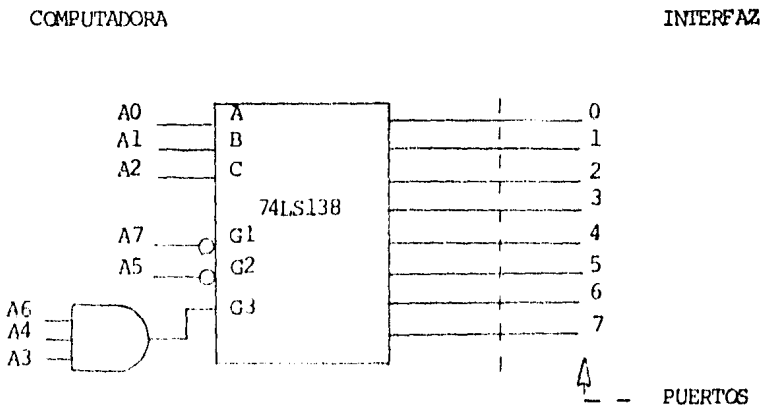


Figura 2.14. Selector para 8 puertos.

decimos que estamos interfazando a la CPU, a el "mundo real". En otras palabras, estamos conectando a la computadora dispositivos que realizan alguna función. Después de todo, la CPU no es nada mas que un "procesamiento" de datos de entrada y salida. En este apartado se verá como conectar interruptores a la computadora.

Todos estos dispositivos son diseñados para usarse con puertos de entrada y salida en paralelo.

### 2.8.1 INTERRUPTORES.

Una de las maneras mas sencillas de establecer operaciones de entrada de información al microprocesador es por medio de interruptores.- El microprocesador podrá leer la señal, si encuentra en las líneas de entrada del puerto, en donde el contenido de esta señal estará constituida por el estado en que se encuentra el interruptor de cada línea -- del puerto correspondiente a cada bit de la señal. En la Fig. 2.15a se ilustra un simple interruptor de entrada. El interruptor esta conectado a la línea del bit 0 de un puerto de entrada en paralelo. La Fig. 2.15b también ilustra el uso de un interruptor pushbutton (contacto momentaneo).

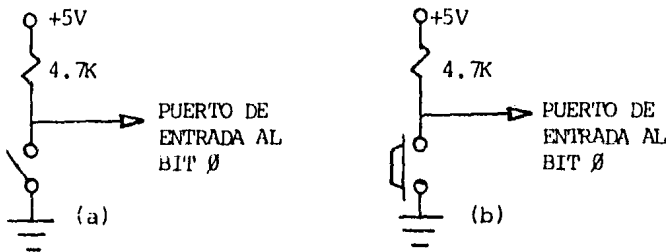


Figura 2.15. Conexión de interruptores a un puerto de entrada.

Cuando usamos este tipo de circuitos de interruptores y resistencias pull-up, la posición "lógica" de un interruptor es invertido. Un interruptor abierto (OFF), será leído por la computadora como un 1 lógico y un interruptor cerrado (ON), será leído como un 0 lógico. Esto -

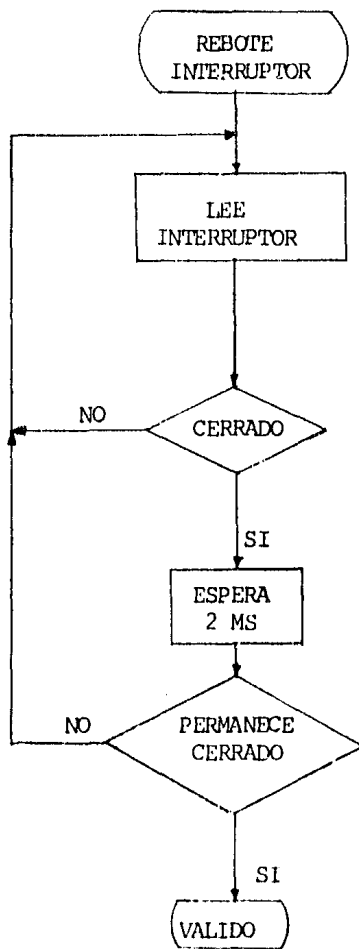


Figura 2.16. Diagrama de flujo de una rutina de rebote de interruptor.

puede causar confusión porque normalmente conocemos a un "ON" como un "1" y "OFF" como un "0". Para cambiar los estados lógicos se pueden usar inversores después de los interruptores o inversores en software. El principio de funcionamiento es el mismo para los demás interruptores que se conectan a las demás líneas del puerto de entrada, de esta manera, el microprocesador sentirá las señales digitales en paralelo que se

encuentran en el puerto de entrada, determinando y analizando su contenido si se trata de una señal de control, dirección o de datos.

Los interruptores tienen un ruido asociado con su operación llamado "rebote en interruptores". Este rebote es producido por el contacto mecánico de los interruptores cuando se cierran. El interruptor puede ir de ON a OFF en tiempo de milisegundos. Por ejemplo, el rebote en los contactos del teclado de una terminal causaría que muchos caracteres fueran generados cuando únicamente una es oprimida una sola vez.

Este rebote en los interruptores puede ser eliminado por una variedad de circuitos (Hardware). Sin embargo, el método más económico para eliminar este rebote es por medio de un programa (software). Este rebote puede ser realizado al sensar un interruptor al cerrarse 2 veces, separado por un intervalo de tiempo apropiado. Un intervalo de tiempo de unos pocos milisegundos es usualmente suficiente. El diagrama de flujo para el programa rebote en interruptores es como se muestra en la Fig. 2.16.

Si se utiliza la lógica cableada (Hardware) para eliminar el ruido o rebote, se emplean circuitos llamados multivibradores conocidos como "ONE SHOT".

### 2.8.2 INTERFACE CON TECLADOS.

Los teclados alfanuméricos típicos, como los usados en terminales de video, emplean 50 o más interruptores. Estos interruptores pueden ser conectados directamente a los puertos en paralelo. Sin embargo, este llevaría seis o más puertos, cada uno de los cuales habría de ser sensada para determinar cuál tecla ha sido cerrada. La interface del circuito electrónico, y por consiguiente el costo, puede ser sustancialmente reducido por una técnica llamada matriz-sensora.

La misma situación existe en sistemas que emplean un número grande de interruptores sensores. Por ejemplo, en sistemas de seguridad con interruptores tipo sensores en puertas y ventanas, detectores de fuego y humo, etc. Pueden tener varios interruptores de entrada a la computadora. En este caso la cantidad de alambrado, también como interface electrónico, muchas veces puede ser reducido por un sistema de matriz-sensora.

El esquema básico para el sistema de la matriz-sensora es ilustrada en la Fig. 2.17.

Esta consiste de una matriz alambreado de interruptores con contactos normalmente abiertos en cada intersección. En este caso hay 64 interruptores individuales con contactos. Estos interruptores pueden ser teclas o un teclado alfanumérico, o cualquier dispositivo con interruptores de tipo cerrado. Únicamente se necesitan dos puertos de E/S: uno de salida y uno de entrada. Este reduce el número de puertos de los 8 previos a únicamente dos, y el número de alambres de los 65 previos a únicamente 16.

Es necesario un programa para sensar los interruptores de la matriz y determinar si un interruptor ha sido cerrado, y si es así, cual de ellos. También este programa debe tener una rutina para evitar el rebote de los contactos del interruptor, ya que de lo contrario existe el riesgo de tener algún ruido que interfiera en las señales que se envían. Otro problema y que debe resolver el programa, es cuando se oprime más de una tecla; en este caso la rutina sensora determinará si se oprimieron dos o más renglones o si se oprimieron más de dos columnas.

### 2.8.3 TECLADO CODIFICADO.

El ejemplo previo mostrado como interface de una matriz de teclas con interruptores (teclado sin codificar) con un mínimo de hardware. El CPU esta a la izquierda para hacer la mayor parte del trabajo, muchos cambios fueron hechos para llevar un hardware simple. Sin embargo, hay varios circuitos integrados de "TECLADOS CODIFICADORES" que realizan todo el sensado, decodificando funciones y detectando errores en hardware. Estos circuitos integrados proveen ocho bits de datos y un estrobo de dato disponible para la CPU.

El circuito de la Fig. 2.18 muestra un interface típico de teclado codificador.

Usted puede obtener el circuito integrado del teclado codificador y un teclado sin codificar y alambrear la matriz por si mismo; sin embargo, los circuitos integrados codificadores son usualmente disponibles en teclados ASCII (American Standard Code Information Interchange-Código normal americano para intercambio de información.- Código bina-



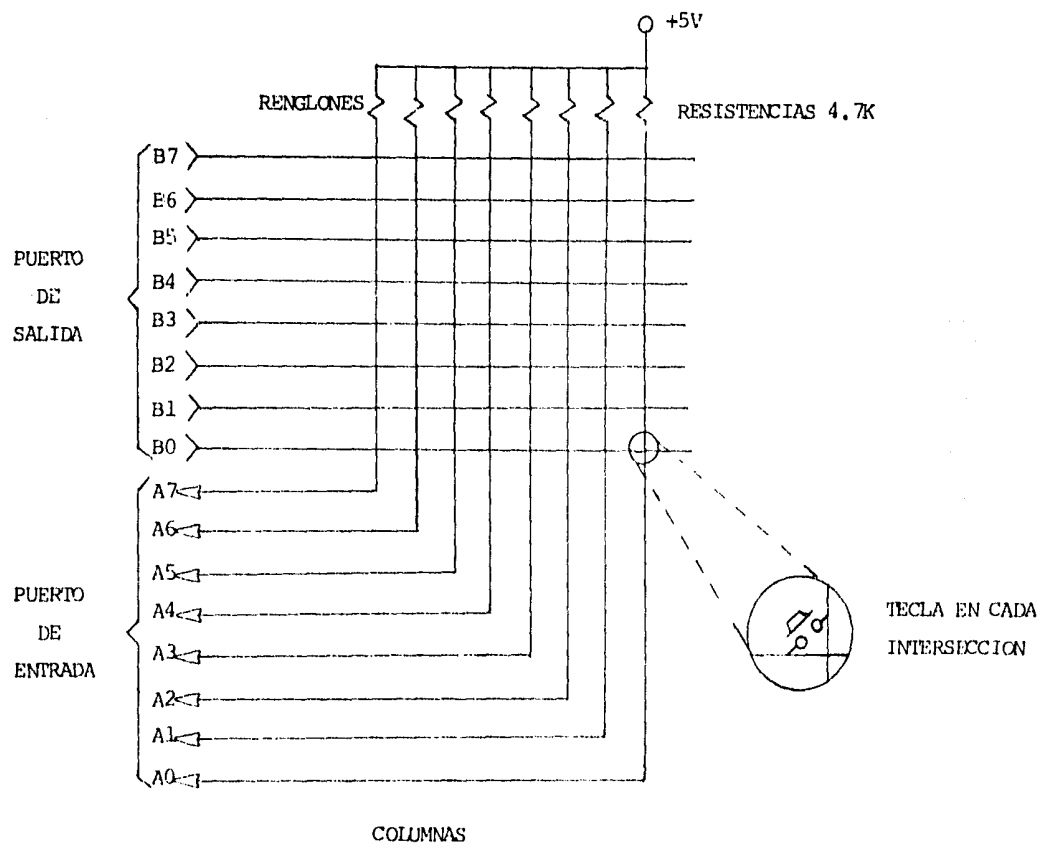


Figura 2.17. Sistema de Matriz sensora de un teclado.

TECLADO DE INTERRUPTORES

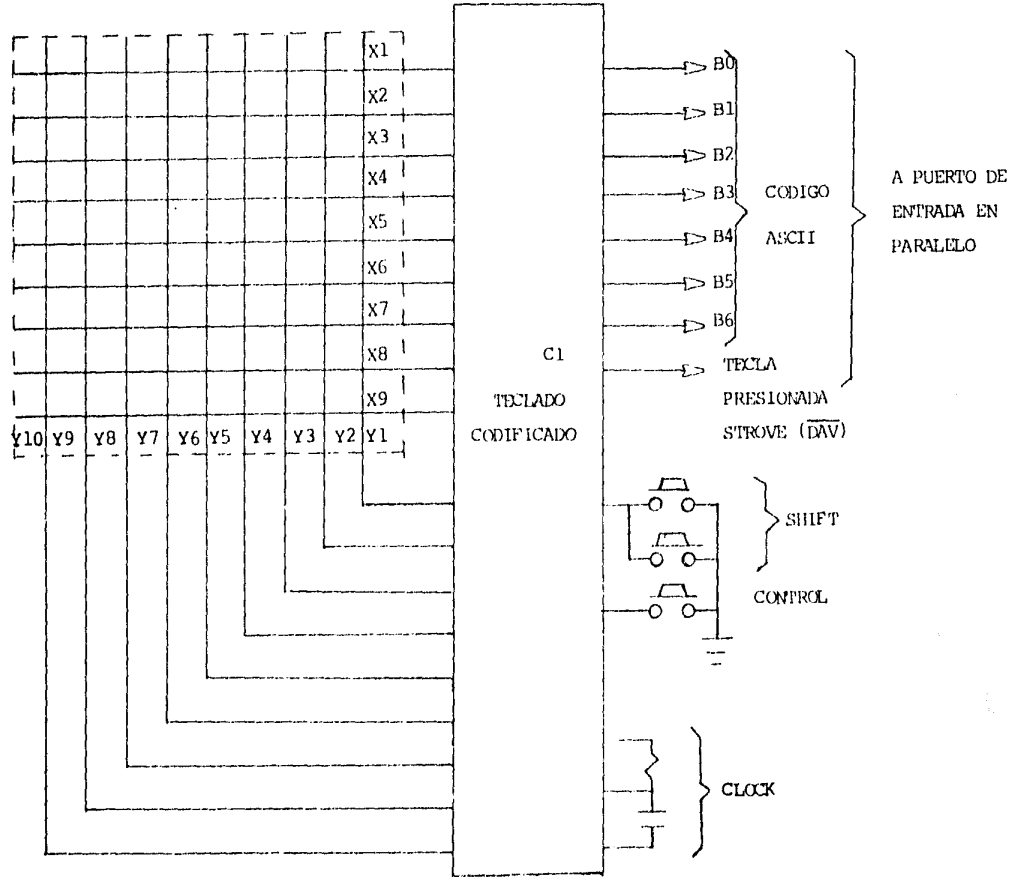


Figura 2.18. Circuito típico de teclado codificado.

rio desarrollado por la American Standards Association (ASA), con el cual se puede representar, en 7 bit, hasta 128 comandos, números y letras y caracteres especiales. Este código es el mas empleado para la transmisión de datos) comerciales. En otro caso, el circuito integrado o teclado codificador puede ser conectado a la computadora usando un circuito latch para el puerto de entrada en paralelo.

### 2.9 DISPOSITIVOS PERIFERICOS DE SALIDA.

En el apartado anterior se mencionó como conectar dispositivos de entrada a una computadora.

En este apartado mostraremos como conectar dispositivos periféricos a la salida de una microcomputadora. Estos dispositivos serán conectados directamente a la microcomputadora.

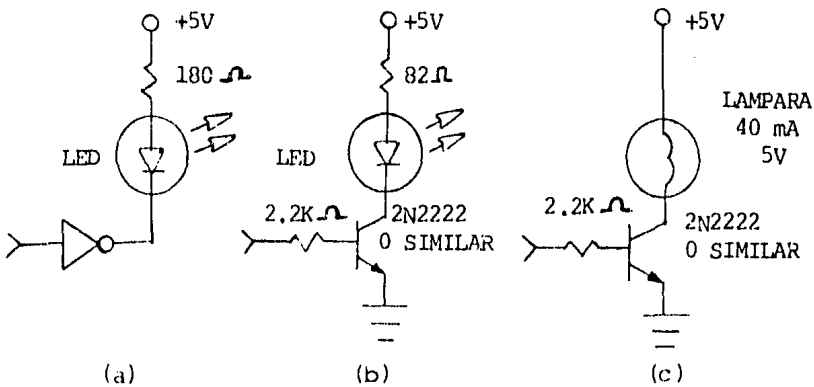


Figura 2.19. Interface con LEDs y lámparas.

#### 2.9.1 INTERFAZANDO LEDs Y LAMPARAS.

Las lámparas son conectadas a la CPU sirviendo como indicadores de estados o eventos. Muy a menudo son usados LEDs. Actualmente, un LED puede ser conectado directamente a un puerto de salida, como se muestra en la Fig. 2.19a. En la Fig. 2.19a el LED puede ser escrito cuando el puerto de salida está en bajo. Cuando el LED esta en ON (encendido), típicamente pasan 16 mA a través del LED. Esta es la máxima capacidad de absorber corriente para una compuerta estandar TTL. Sin embargo el brillo del LED es menor que el deseado. Un buffer de alta-corriente,

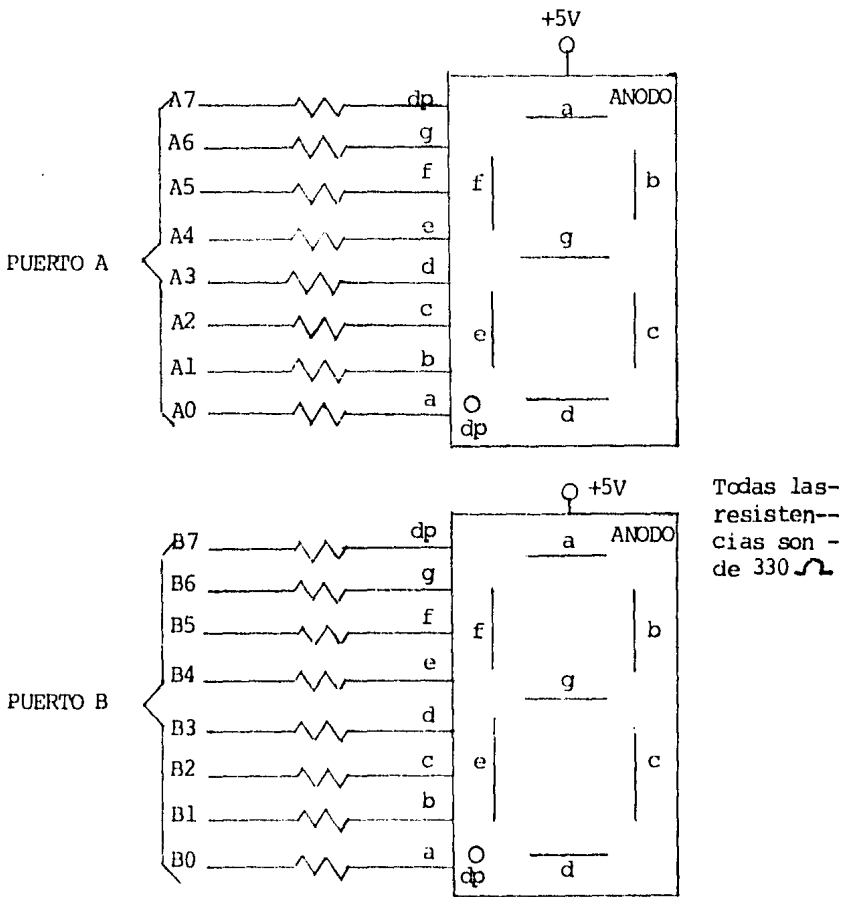


Figura 2.20. Conexión de displays de 7 segmentos a puertos de salida.

tal como el 7437 puede ser usado para absorber hasta 48 mA. Para disminuir el valor de la resistencia limitadora de corriente a 82 ohms, la corriente del LED es incrementada a 32 mA y el brillo es aumentado.

Una alternativa usando un circuito integrado buffer es usando un LED manejado con transistor, como se muestra en la Fig. 2.19b. Especialmente el mismo circuito puede ser usado para manejar una lámpara incandescente, como se muestra en la Fig. 2.19c.

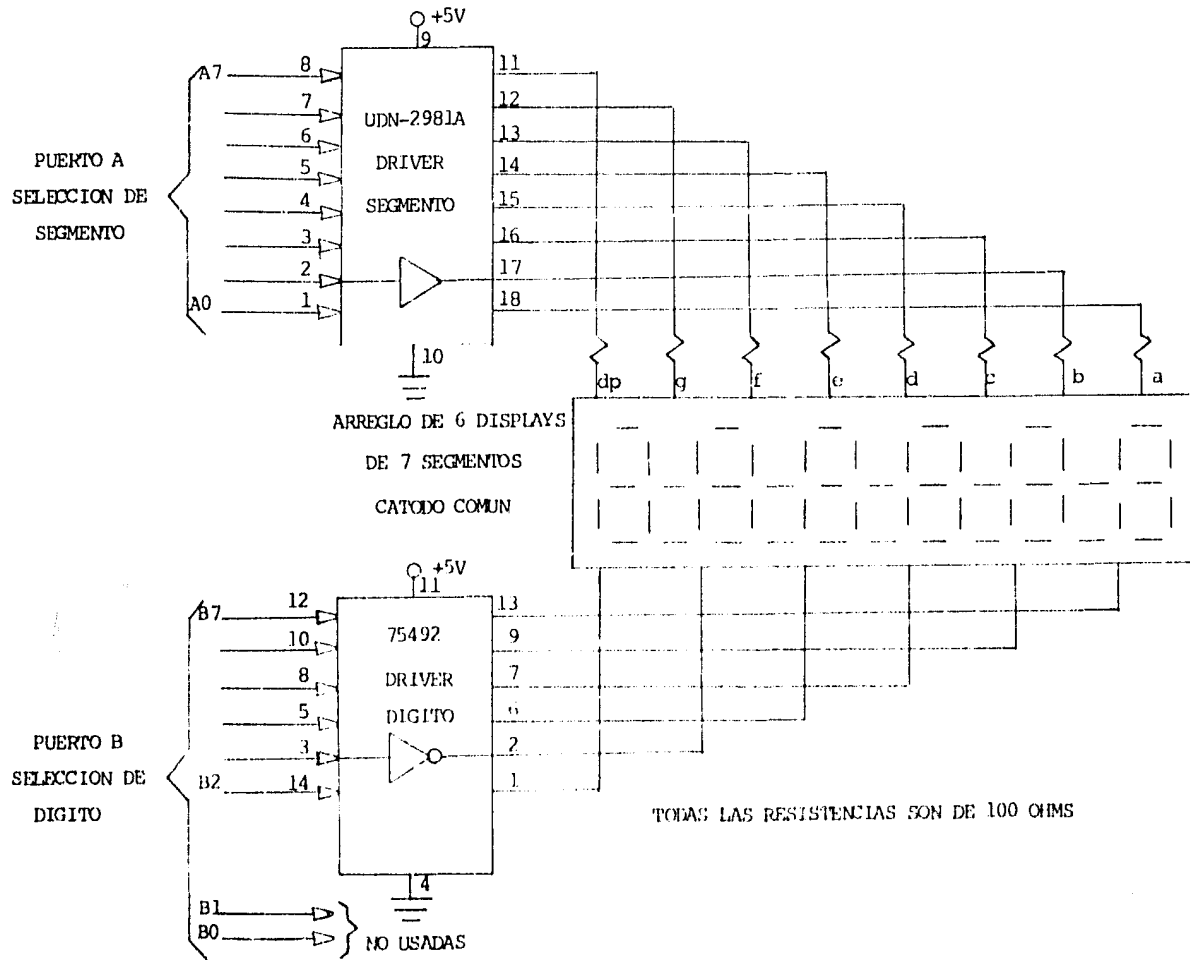


Figura 2.21. Multiplexaje de un arreglo de Display de 6 dígitos.

Los displays LED de 7 segmentos son también frecuentemente conectados a los puertos de salida de una microcomputadora. Un ejemplo típico es el mostrado en la Fig. 2.20. Cada uno de los displays es conectado a un puerto de salida. Para los caracteres del display, los códigos del carácter son cargados en los registros de cada uno de los puertos.

Cuando muchos displays de 7 segmentos son usados el puerto del -- circuito empieza a ser muy complejo. En tal caso vale la pena considerar el multiplexaje del display. Por ejemplo, para manejar displays de 7 segmentos se requerirán de seis puertos separados y 48 resistencias. Multiplexando esto se reduciría a dos puertos y ocho resistencias. Un circuito típico de displays multiplexados es mostrado en la Fig. 2.21. Este emplea un tablero o panel de displays de 7 segmentos cuyos segmentos son conectados en paralelo.

Esto reduce, el alambrado a un mínimo. Los circuitos integrados -- UDN-2981A Sprague y 75492 Texas Instruments son específicamente designadas para esta aplicación. El 2981A contiene driver de ocho segmentos y el 75492 contiene 6 seleccionadores de dígito. Dos puertos de salida son empleados para almacenarlos, uno para las salidas de segmentos y el otro para seleccionar los dígitos.

El procedimiento de multiplexaje es para examinar un dígito del -- display en un tiempo. Como cada uno de los dígitos es habilitado en secuencia, es provisto un código de segmento apropiado. Por lo que es necesario un programa que controle el procedimiento entero. En el programa; los códigos para los caracteres a ser exhibidos son cargados dentro de la memoria con la dirección de los displays. Cada uno de los códigos del segmento es entonces buscado en la memoria y enviado al puerto del segmento tal como dígito seleccionado. Después de examinar todos los -- dígitos, el proceso es repetido.

## 2.10 GENERACION DE UN CARACTER EN UNA MATRIZ DE DIODOS LED.

El propósito fundamental para este caso es generar una letra en -- una matriz de diodos "LED" de 5X7 elementos. El siguiente diagrama a -- bloques muestra el funcionamiento de un sistema que puede llevar a cabo esta tarea.

Dependiendo de la letra que se vaya a generar en la matriz, se --

programan 35 direcciones en la memoria con la información correspon-  
 diente a los diodos que deben estar encendidos o apagados para obtener  
 este carácter. La información así programada es recirculada y extraída  
 en serie para alimentar un registro del tipo "entrada serie salida pa-  
 ralelo" que almacenará en un circuito "latch" palabras de siete bits, -  
 que a su vez controlarán el encendido y apagado de cada renglón de la -  
 matriz de diodos. La unidad lógica sincronizará este proceso a través -  
 de un decodificador a cinco líneas para que en forma cíclica vaya acti-  
 vando cada columna en el momento en que la información que le corres-  
 ponda aparezca a la salida del "latch". A una frecuencia suficientemen-  
 te elevada (1 KHz) se obtendrá la ilusión óptica de la presencia de la  
 letra deseada en la matriz.

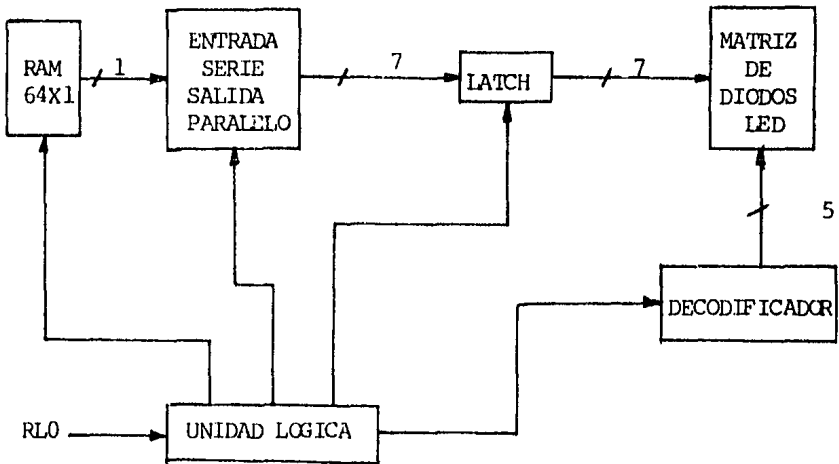


Figura 2.25. Diagrama a bloques de una matriz de diodos LED.

Es de hacer notar que el circuito clave para implementar este ---  
 circuito, es el correspondiente al de la unidad lógica que genera las -  
 fases del reloj necesarias para controlar cada una de las actividades -  
 de los bloques en forma sincronizada.

## 2.11 SISTEMAS DE ALMACENAMIENTO DE DATOS.

El punto principal es, elaborar un sistema alrededor de una memo-

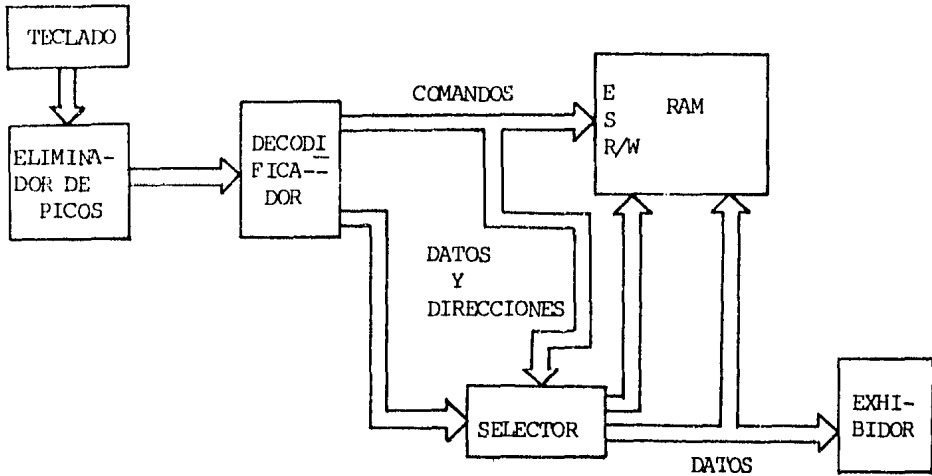


Figura 2.26. Diagrama a bloques del sistema.

ria RAM que adquiera, almacene y permita la posibilidad de mostrar una serie de datos que le sean provistos a través de un teclado. El sistema a bloques tendría la estructura mostrada en la Fig. 2.26.

De el decodificador salen un par de grupo de líneas: las de comandos que manejan la habilitación, lectura y escritura de la memoria, y las líneas de datos y direcciones que son las mismas pero que tienen que pasar a través de un circuito selector para poder distribuirlos, en los momentos adecuados, en los "Buses" de datos y direcciones de la memoria. Esta distribución se efectúa con la ayuda de una línea especial del comando obtenido del teclado, que determina cuando se direcciona y cuando se envía un dato para almacenar. Como el "bus" de datos de la memoria es bidireccional es necesario que el selector de datos cuente con un circuito de lógica de tres estados a su salida para que pueda manejar directamente la memoria.

El sistema presentado puede variar considerablemente en su magnitud dependiendo del número de datos que se desee manejar. Por esto, es de recomendar que el número de ellos especificado sea relativamente pe-



queño para que el sistema pueda ser construido con una cantidad razonable de unidades lógicas. Cien datos como máximo podría ser la especificación.

## CAPITULO 3

### DESCRIPCION DEL HARDWARE.

#### 3.1. INTRODUCCION AL CIRCUITO.

En este capítulo se presenta la circuitería necesaria para el --- diseño del Probador de Circuitos Integrados Digitales. Es importante -- mencionar, que existe un programa (software), el cual será presentado - en el siguiente capítulo, éste es compatible con el Hardware presentado aquí, por lo que no se tendrá ningún problema de adaptación a éste.

Mucho se viene hablando de Hardware, por lo que es de suma im--- portancia definirlo más claramente antes de continuar con la descrip--- ción.

El término Hardware se define como: Todos los circuitos básicos - que componen a una microcomputadora tales como: memorias, interfaces, - decodificadores, microprocesador, etc. Esto es en si lo que muchos han definido como "Logica Alanbrada".

Dentro del sistema se distinguen los siguientes circuitos funda--- mentales (Fig. 3.1).

1.- Circuito Principal.- El cual incluye la CPU, la memoria, los dispositivos de control y un teclado.

2.- Los exhibidores luminosos.- Son los que muestran al usuario - en el momento de prueba el mensaje dado por éste, así como los comandos que se introducen para la misma.

3.- La fuente de alimentación. Es la que se encarga de proveer al sistema del Voltaje apropiado para su operación.

La explicación de como opera el sistema se dá en el Apéndice A -- (Manual del Usuario), en este capítulo únicamente se expondrán los cir-

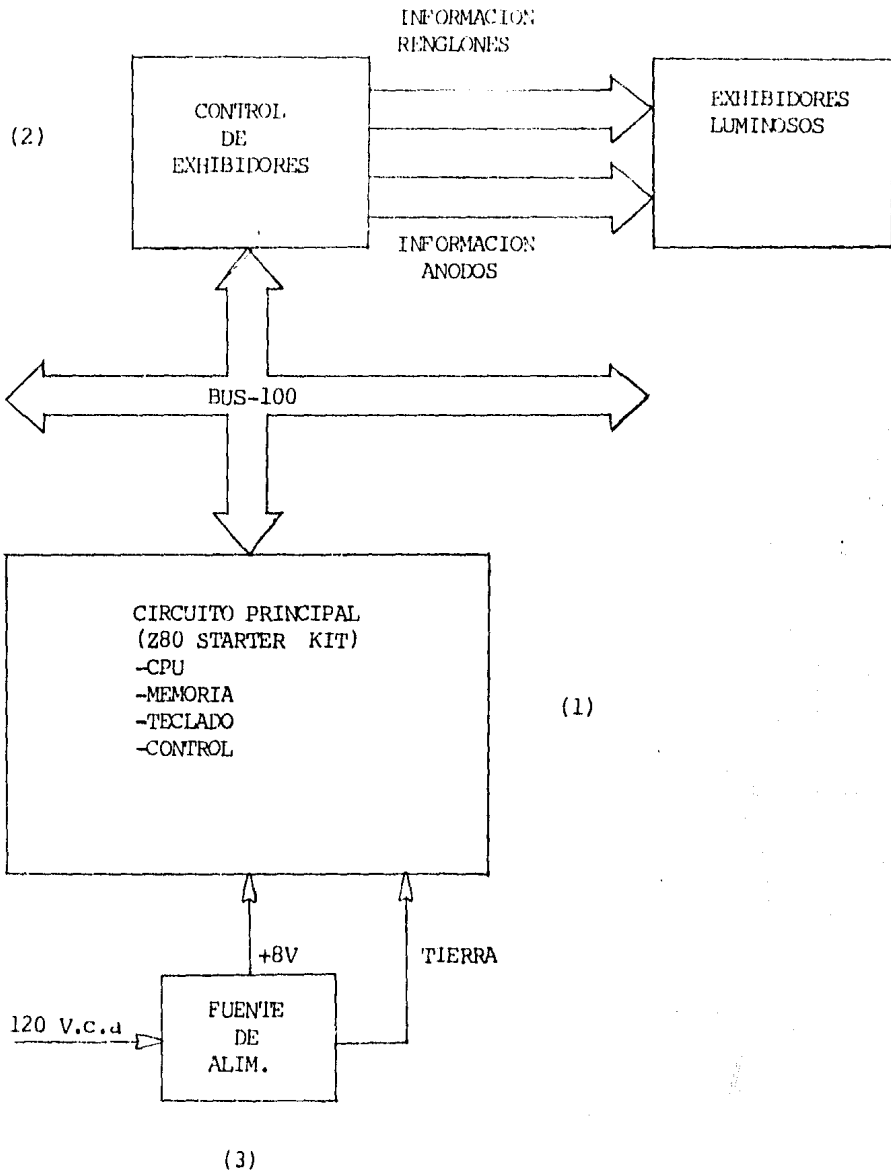


Figura 3.1. Diagrama a bloques del Hardware del Probador de Circuitos Integrados Digitales.

cuitos.

### 3.2 CIRCUITO PRINCIPAL.

Primeramente empezaremos por definir este sistema. Este es una pequeña microcomputadora producto de la compañía norteamericana SD --- Systems, denominado "Z80 Starter Kit" diseñada principalmente para la enseñanza de los fundamentos de la microcomputación, aunque, como dice el fabricante, está dirigido a los entusiastas de la computación, a los experimentadores en la electrónica, a los radioaficionados y a los fabricantes de equipo original industrial. Esta microcomputadora es barata, sencilla y fácil de adquirir.

El Z80 Starter Kit (que a partir de ahora lo llamaremos simplemente "Starter") es una sola tarjeta de circuito impreso (Fig. 3.2), -- con dimensiones de 30.5 cm. de largo, por 30.5 cm. de ancho y 5 cm. de alto, en la cual van montados directamente los diversos componentes --- propios del circuito principal, contando además con un área para que el propietario pueda expandir el circuito a placer y un par de conectores elementales para BUS S-100.

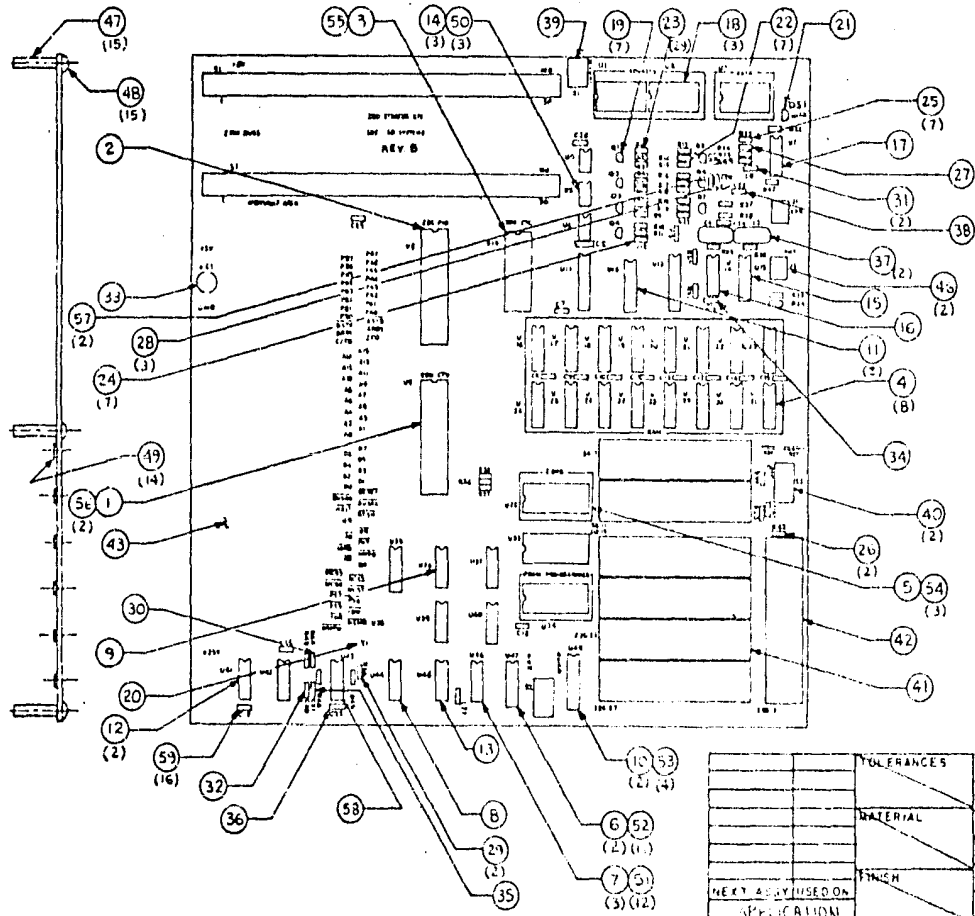
Refiriéndose a la Fig. 3.2, los componentes más importantes del Starter son:

a) Una Z80-CPU (número 1 en la Fig. 3.2).- El microprocesador --- Z-80 es una unidad central de procesos (CPU), fabricado en un solo circuito integrado de 40 patas con técnica de alta escala de integración - (LSI), utilizando el proceso de canal N para compuertas MOS.

b) Un Z80-PIO (2).- Es un circuito integrado que contiene dos --- puertos entrada/salida paralelo, de 8 bit cada uno, totalmente progra-- mables y compatibles con la Z80-CPU.

c) Un Z80-CTC (3).- Es un circuito integrado que contiene 4 cro--- nómetro/contadores programables y que se accesan como puertos E/S. Este circuito integrado es especialmente útil como base de tiempo para relojes hechos por Software, pudiendo trabajar independientemente de la CPU y solo accesarla mediante interrupciones cuando un cronómetro/contador-- llegue a un número preestablecido.

d) Un kiloByte de RAM (4).- Está formada por 8 circuitos integra--



REVISION			
LTR	DESCRIPTION	DATE	APP

85

TOLERANCES		DRAWN BY D.O.		SEE SEPARATE B.O.M. 0100080	
MATERIAL		CHECKED BY		SD SYSTEMS GARLAND TEXAS	
FINISH		DESIGN ENGR		Z80 STARTER KIT ASSEMBLY	
NEXT ASSEMBLY USED ON APPLICATION		APPROVED		SIZE B	CONTROL IDENTIFYING NO C100080
				SCALE 1:1	REV 1
				SHEET 1 OF 4	

Figura 3.2. Localización de componentes en el Z80 Starter Kit. SD Systems, 1978.

dos de memoria de lectura/escritura, expandible directamente a 16 circuitos integrados, para un total de 2 KByte de Ram.

e) Dos KiloByte de ROM (5).- Es un solo circuito integrado programado de fábrica con un programa base llamado monitor, el cual le sirve originalmente al Starter para permitir al usuario introducir sus propios programas en RAM, depurarlos, ejecutarlos y guardarlos, si lo desea, en una grabadora magnetofónica ordinaria de "cassettes". El Starter cuenta con 2 bases para poder aumentar el programa base hasta 4 kByte más, con circuitos integrados de EPROM, donde se deposita el Software del Probador.

f) Tres exhibidores dobles de 7 segmentos (18).- Sirven originalmente para las funciones del programa monitor, exhibiéndose en ellos direcciones o datos con números hexadecimales.

g) Un cristal oscilador de 3.9936 MHz (20).- Es el que fija con precisión la frecuencia de oscilación del reloj del sistema. La frecuencia del reloj del sistema es exactamente la mitad de la frecuencia del cristal (1.9968 MHz).

h) Un teclado (41).- Está formado por 28 teclas y le sirve al Starter para las funciones del programa monitor, programando en lenguaje de máquina en formato hexadecimale (en el extremo derecho de la Fig. 3.4- se puede ver el teclado, con la asignación original de las teclas).

Apoyando a los componentes principales, se encuentran varios circuitos integrados, interruptores, conectores y componentes discretos, los cuales ejercen diversas funciones en el Starter.

Para ampliar el conocimiento sobre el Z80 Starter Kit, se recomienda consultar la referencia B Apéndice B.

Los diagramas del circuito electrónico del Starter se pueden ver en las Figs. 3.3 y 3.4.

Descripción del circuito de la Fig. 3.3.

1) La Z80-CPU (U9, a la izquierda del diagrama) se interconecta con los otros componentes vía el bus de direcciones (A0-A15), el bus de datos (D0-D7), las entradas  $\overline{\text{BUSRQ}}$ ,  $\overline{\text{NMI}}$ ,  $\overline{\text{INT}}$ ,  $\overline{\text{WAIT}}$ ,  $\overline{\text{RESET}}$  y CK, y las salidas  $\overline{\text{BUSAk}}$ ,  $\overline{\text{FRESH}}$ ,  $\overline{\text{HREQ}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{MI}}$ ,  $\overline{\text{TORQ}}$  y  $\overline{\text{WR}}$ .

2) Los circuitos integrados de RAM ( al centro, de izquierda a de-

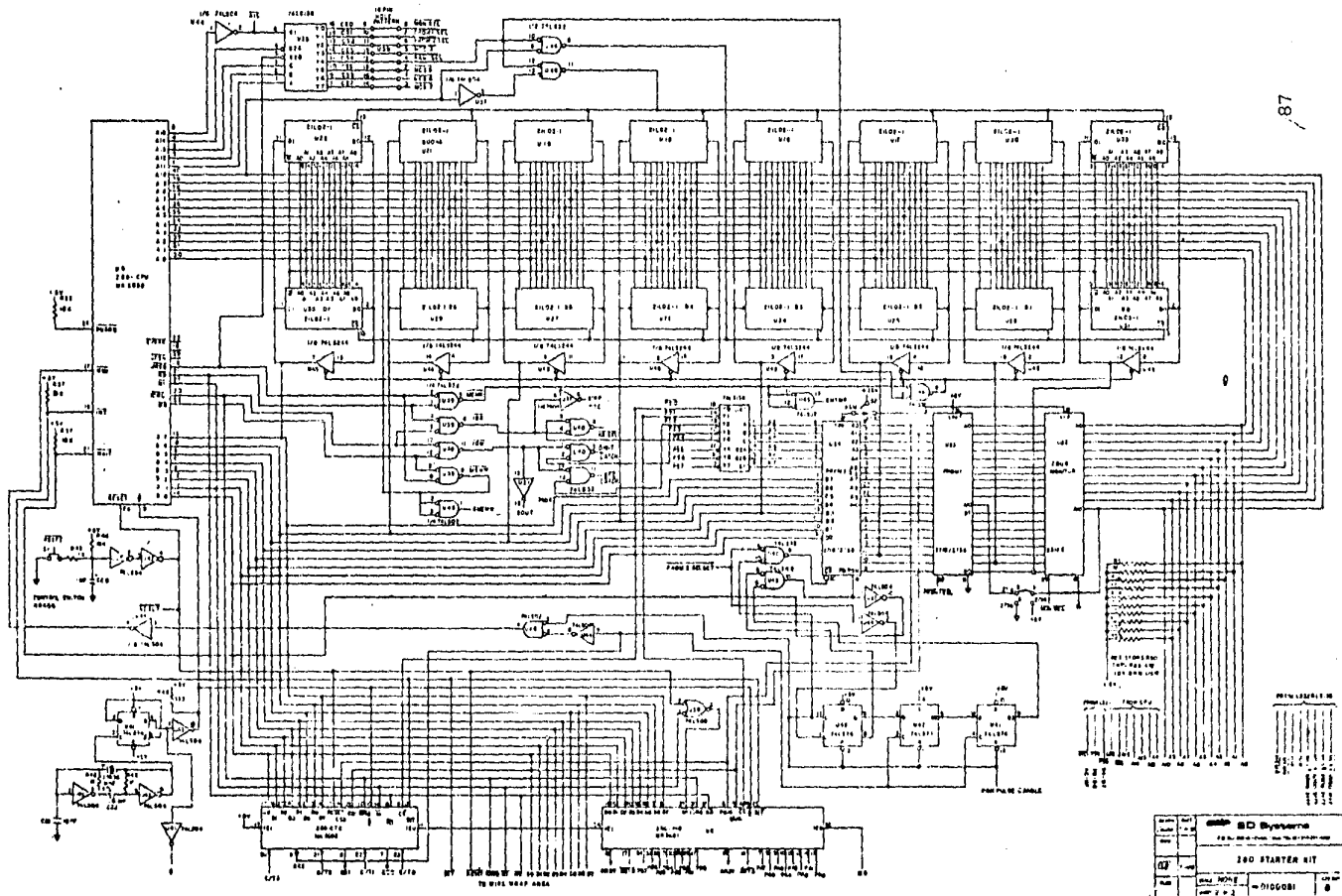


Figura 3.3. Circuito del Starter; CPU, MEMORIA, CTC, PIO Y RELOJ SD SYSTEMS. 1978.

recha, U30, U29, U27, U26, U25, U28 y U31) se conectan directamente a -- las líneas A0-A9 del bus de direcciones e indirectamente a las líneas -- A10-A15 por medio de los circuitos de selección (1/6 U44, U35, 1/2 U46 y 1/6 U37). Se conectan al bus de datos vía 8 buffer de 3 estados, con--- trolados por las señales de  $\overline{MREQ}$  y  $\overline{RD}$ , por medio de 1/4 U39 y 1/4 U46. - La lectura/escritura de la RAM se controla directamente por la terminal-  $\overline{WR}$ .

3) La ROM del programa monitor (U32, a la derecha del diagrama), al igual que las EPROM de expansión (U33 y U34) se controlan y accesan de - manera similar a los circuitos integrados de RAM, excepto que la cone--- xión al bus de datos es directa, al igual que la línea A10 del bus de -- direcciones, además de no llevar conexión con  $\overline{WR}$  y que U34 tiene conec-- tado un sistema primitivo para programar EPROM (1/2 U45, 1/6 U37, 1/2 -- U44, U42, 1/2 U41 y 1/4 U10).

4) El Z80-CTC (U10, abajo a la izquierda) se conecta directamente al bus de datos, directamente de A0 y A1 del bus de direcciones e indi-- rectamente a las líneas A2-A7, mediante el selector (U11). Adicionalmente, se conecta a las líneas  $\overline{RESET}$ , CK,  $\overline{IORQ}$ ,  $\overline{MI}$ ,  $\overline{RD}$  e  $\overline{INT}$  de la CPU y con el Z80-PIO (U8) mediante la línea de prioridad de interrupciones IE0. El -- CTC tiene otras líneas de control (C/TO-C/T3, ZCO-ZC2) que el usuario -- puede conectar a conveniencia.

5) El Z80-PIO (U8, abajo al centro) se conecta al sistema de igual manera que el Z80-CTC, siendo la conexión con éste vía la terminal de -- prioridad IE1. El PIO tiene disponibles sus 20 líneas externas (PA0-P47,  $\overline{ASTB}$ , ARDY, PBO-PB7,  $\overline{BSTB}$  y BRDY), para que el usuario utilice sus dos - puertos paralelo a conveniencia (ver apartado 3.3.1 de este capítulo).

6) El reloj del sistema (ángulo inferior izquierdo del diagrama) - está formado por 4 inversores (2/3 U43), un biestable "D" funcionando -- como divisor entre dos (1/2 U41), el cristal de 3.9936 MHz (Y1), 2 re--- sistencias de 1 Kohm (R45 y R48), un capacitor de 0.01 uF (C22) y un ca- pacitor de 10) pF (C21). Con el total de estos componentes, el reloj del sistema dá una frecuencia de 1.9968 MHz.

7) Dos de los seis inversores de U43 (debajo de la CPU), el inte--- rruptor S1, las resistencias R43 y R44, y el capacitor C20, forman el --



circuito que provee la señal de RESET a todo el sistema. Este circuito - tiene dos maneras de dar dicha señal: una, al alimentar voltaje por primera vez al sistema, la otra, al oprimir el interruptor S1 en cualquier momento que desee, forzando al sistema a reinicializarse.

8) Finalmente, los circuitos integrados U39, 1/2 U45, 3/4 U40 y 1/3 U37 producen algunas de las señales de control para el bus S-100 y para el circuito de la Fig. 3.4.

Descripción del circuito de la Fig. 3.4.

1) Los 6 exhibidores luminosos de 7 segmentos, empacados de dos en dos (arriba al centro, U1, U2 y U3), son del tipo LED de cátodo común. Los segmentos de cada exhibidor están conectados en paralelo entre sí y conectados al bus de datos vía un biestable "D" óctuple (U12), que sirve para almacenar la información hacia los segmentos, y 7 transistores, --- mientras que los cátodos están conectados también al bus de datos por -- medio de otro biestable "D" óctuple (U11) y 6 manejadores de periférico (U4, U5 y U6). A su vez, los biestables "D" son seleccionados por 1/2 -- U40 (Fig. 3.3). Así, los exhibidores trabajan multiplexados o en "tiempo compartido", encendiéndose sólo uno de los seis a la vez, por espacio de unos cuantos microsegundos, cíclicamente.

2) El teclado (S4-S31, a la derecha del diagrama) es trabajado en forma matricial, recibiendo datos de barrido en renglones y dando datos de encuentro en columnas. Los renglones son accésados por el mismo biestable "D" óctuple (U11) de los cátodos de los exhibidores, mientras que los datos de las columnas se conectan al bus de datos vía 5 buffer de 3-estados (5/8 U13, abajo a la izquierda), controlados por 1/4 U40 (Fig. - 3.3). Las cuatro teclas del extremo derecho generan una interrupción no-mascarable cuando son oprimidas, por medio de un monoestable constituido por 1/2 U33, C5 y R32, y un canal del Z80-CTC (1/4 U10, Fig. 3.3). Esta característica puede ser eliminada al desconectar el puente que une a -- U13 con la línea común de dichas teclas.

3) El circuito para guardar y recuperar datos y programas en cinta magnetofónica (abajo, al centro) consta de dos partes: la receptora y la transmisora, controladas por la CPU como interface tipo Kansas City --- Standard, con una velocidad de 300 Baud (ver la referencia 8 del apéndice



ce B).

El sistema de recepción está formado por el conector J1, un circuito limitador de amplitud (C23, C24, CR1, CR2, R24, R25, R26, R27, R28, R49 y 1/4 U7), un indicador de nivel de señal de audio (C6, Ds1, R22, R47 y 1/4 U7) y un detector-decodificador de frecuencia (C4, R29, 1/2 U14 y 1/2 U15), conectado al bus de datos vía dos buffer de 3 estados (1/4 U13). El circuito transmisor está formado por un generador de frecuencia (1/4 U10, Fig. 3.3), un divisor entre dos (1/2 U14) y un filtro de salida (C16, R31), el cual se conecta a la grabadora de "cassette" mediante el conector J2.

Al pie de la Fig. 3.4 se ve la asignación de terminales del bus S-100 que trae normalmente el Starter, el cual es una versión reducida del bus S-100 común.

### 3.3 CIRCUITO INTERFACE.

Dado que el Probador de Circuitos Integrados Digitales como ya se mencionó en el capítulo 2, probará circuitos integrados de 14 y 16 pata--tas. Se pensó en el circuito programable Z80-PIO (circuito integrado que compone al Starter), el cual contiene dos puertos entrada/salida paralelo, de 8 bit, cada uno totalmente programables y compatibles con la ---Z80-CPU, estos 2 puertos se encuentran disponibles al igual que las lí--neas de handshake (éstas últimas no utilizadas para nuestro caso) para -interfazar mecanismos en paralelo. Estas líneas podrán ser llevadas a el área de wire wrap para conectar la circuitería o dispositivos necesari--os. El área de wire wrap cuenta con espacio para 25-30 circuitos integrados para experimentos adicionales, expansión de memoria, interface --video, etc.

Las señales de la Z80-CPU y del Z80-PIO decodificadas en el Star--ter, han sido puestas cerca del área de wire wrap logrando con esto una rápida y fácil conexión de los experimentos a realizar, en ésta área las conexiones son a base de alambrado, tensadas una de la otra, por lo que se recomienda, usar pins o wire wrap sockets (base con patas largas).

Antes de mencionar la forma de como fué interfazado el Z80-PIO con

las 7 bases de los circuitos integrados a probar dentro del Starter, es importante dar una descripción detallada de lo que es y hace como tal, y por ser el integrado que por sus características fué utilizado para el diseño del Probador como se verá en el apartado 3.3.1.

### 3.3.1 INTERFAZ Z80-PIO.

El Z80-Parallel I/O (PIO) es un circuito programable, que provee 2 dispositivos de puerto compatibles TTL para comunicar un dispositivo periférico y el microprocesador Z80.

El microprocesador Z80 puede programar el comportamiento del Z80-PIO dentro de un amplio rango de requerimientos de dispositivos periféricos sin necesidad de lógica adicional, pudiendo así conectar el microprocesador Z80 gran diversidad de dispositivos como son teclados, impresoras, lectoras perforadoras de papel, programadoras de PROM, etc.

El Z80-PIO utiliza un sustrato tipo n siendo encapsulado en un chip de 40 patas.

Entre sus principales características están:

- 2 puertos bidireccionales de 8 bits independientes con transferencia de control mediante handshake.

- Interrupciones manejadas mediante el handshake para rápida respuesta.

- 4 tipos diferentes de operación seleccionables incluyendo:

- Byte output

- Byte input

- Byte bidireccional bus (disponible sólo en un puerto (A)).

- Bit de control

Todos con posibilidad de interrumpir.

- Lógica de prioridad de interrupciones para vectoriado de interrupción sin necesidad de lógica externa (Daisy Chain).

- Sus salidas son capaces de manejar transistores Darlington.

- Sus salidas son totalmente compatibles con lógica TTL.

Alimentación única de 5V y una fase de CK requerida.

La arquitectura del Z80-PIO es mostrada en la Fig. 3.5 donde se --

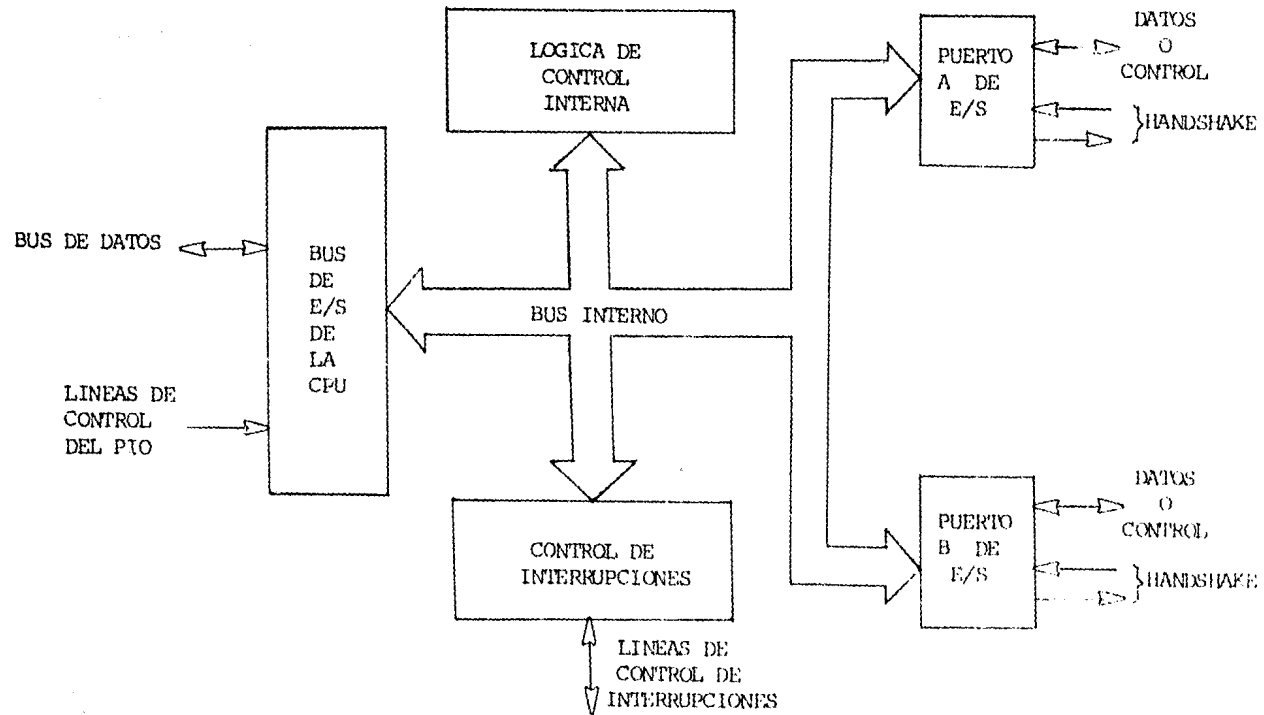
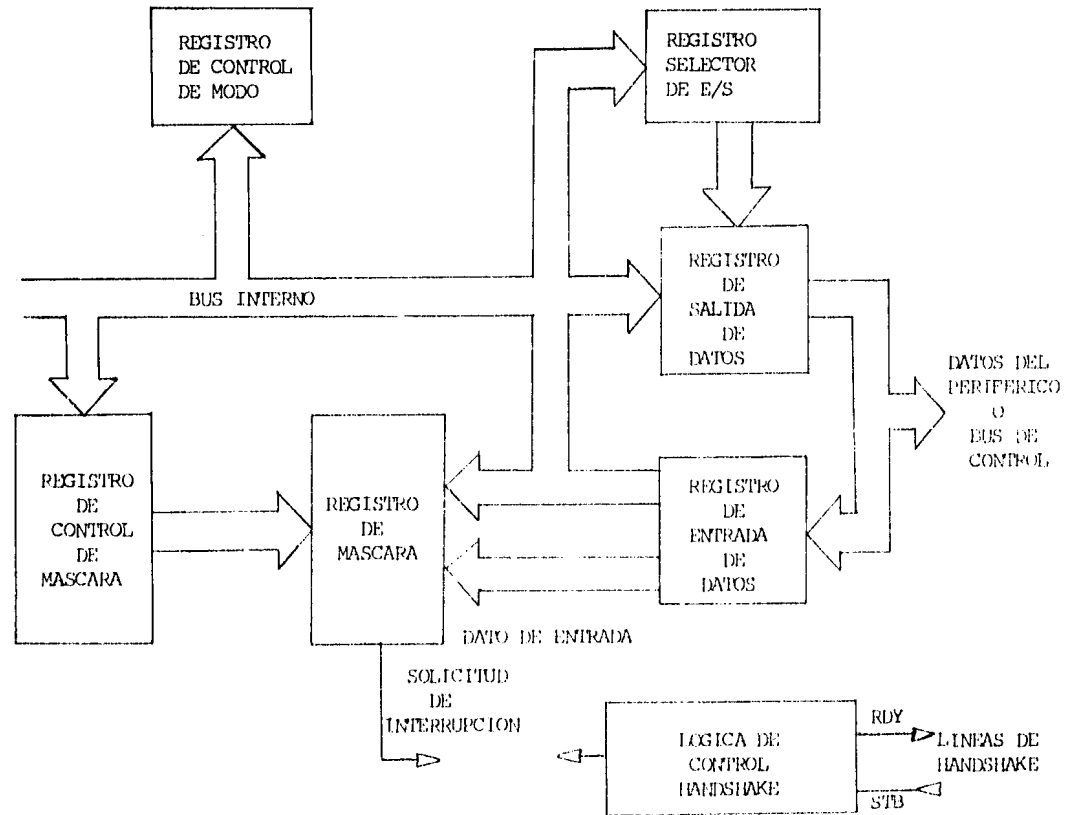


Figura 3.5. Diagrama a bloques del Z80-PIO.



puede observar que el Z80-PIO consiste de una interface con el bus del microprocesador Z80, 2 puertos denominados A y B, una lógica interna para sincronización de las unidades y una lógica de interrupciones estando estos módulos interconectados mediante un bus interno.

La lógica de cada puerto de entrada y salida esta compuesta de 6 registros con la lógica de control del handshake como se muestra en la Fig. 3.6.

Los registros son:

- + Registro de datos de entrada (8 bits)
- + Registro de datos de salida (8 bits)
- + Registros de control entrada-salida (8 bits)
- + Registro de mascara (8 bits)
- + Registro de control de mascara (2 bits)
- + Registro de control de mod (2 bits)

Usos de los registros.

- Registro de control de modo.

En este registro la CPU almacena el modo en que debe operar el PIO entre:

- Byte de entrada
  - Byte de salida
  - Byte bidireccional (únicamente en A)
  - Bit de control
- Registro de datos de salida,

Todos los datos que el microprocesador debe transferir al dispositivo periférico, los deberá escribir en este registro, del cual los leerá el periférico con auxilio de las líneas del handshake en el caso de Byte de salida y Byte bidireccional o bien el periférico los leerá directamente en el caso de bit de control.

- Registro de datos de entrada. Todos los datos que se transfieran entre el periférico y el microprocesador deberán pasar por este registro del cual el microprocesador leerá los datos de entrada que el periférico previamente escribió mediante las líneas del handshake en el caso de Byte de entrada y Byte bidireccional o bien colocó directamente en el caso de bit de control.

Los registros restantes son empleados únicamente en el caso de que el PIO opere en el modo de bit de control como se indica.

- Registros de control entrada/salida.

En este registro el microprocesador almacena un dato que indica que bit opera como entrada y que bit opera como salida pudiéndose definir éstos en forma indistinta.

- Registros de mascara.

En este registro el microprocesador almacena un dato que indica cual o cuales de las líneas de entrada podrán generar interrupción.

- Registro de control de mascara.

En este registro el microprocesador almacena un dato que indicará si los bits facultados a interrumpir lo realizarán con un estado de 0 ó 1 lógico, y si se generan las interrupciones cuando todos los bits facultados interrumpan (función AND) o bien cuando cualquiera de ellos lo hagan (función OR).

La lógica de interrupción contiene todos los protocolos necesarios para solicitar al microprocesador una interrupción, así como, proporcionar el vector de interrupción cuando el microprocesador contesta la interrupción, además de contar con una lógica que permite determinar la prioridad del PIO con respecto a otros dispositivos de acuerdo con su colocación física en el alambrado, proporcionando así, dos líneas para el control de esta prioridad.

Cuando interrumpen tanto el puerto A como el puerto B de un mismo PIO se le da más prioridad al A sobre el B, además el PIO está provisto de una lógica que permita a otros PIO's o cualquier otro dispositivo de mayor prioridad, interrumpir aunque no se haya terminado de ejecutar su rutina de servicio una vez que este ha interrumpido, pero no permite a las de menor prioridad interrumpir sino hasta que se de por terminada su rutina de servicio.

Cuando una interrupción es aceptada por la CPU en modo 2. El dispositivo que interrumpe debe proporcionar en vector de 8 bit al microprocesador, este vector es usado para apuntar las localidades, la memoria donde se encuentra almacenada la dirección de inicio de la rutina correspondiente, para formar el apuntador a las localidades de memoria,-



el microprocesador genera los 16 bits de este apuntador con los 8 bits del registro I y los 8 del vector, siendo los más significativos los del registro I y los menos los del vector, cada puerto (A o B) del PIO tiene un vector propio pudiendo ser diferentes con la única restricción de que el bit menos significativo es 0. El PIO, con el fin de saber cuando terminan de ejecutar su rutina de servicio, esta sensando el código de las instrucciones que el microprocesador, fecha y compara dicho código con el de la instrucción RETI, y cuando el microprocesador lo ha fechado da por terminado su rutina.

Descripción de las señales del Z80-PIO (Fig. 3.7).

La Fig. 3.7 es un diagrama de la configuración del chip Z80-PIO.

DO-D7.- Conexión al bus de datos del Z80-CPU (bidireccional, tres-estados).

Estas líneas son usadas para transferir información del Z80-CPU al PIO o bién del PIO al Z80-CPU.

B/A SEL.- Selecciona puerto A/B (entrada, activa en 1).

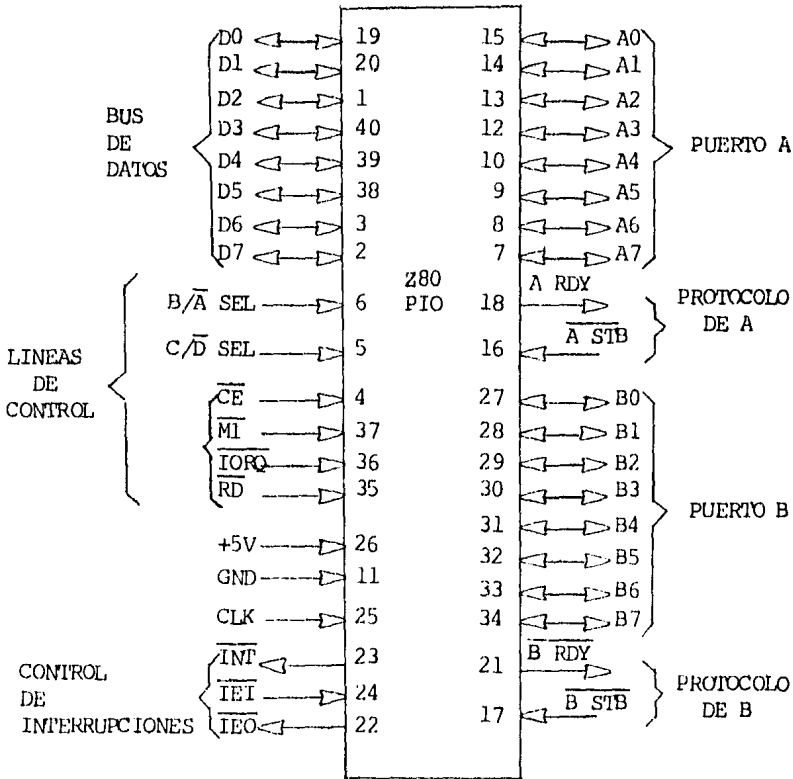
Esta señal, define en que puerto se esta comunicando al microprocesador ya sea para lectura o escritura. Si es 0 se refiere al puerto A y si es 1 se refiere al B, es común que se conecte a la línea A0 del bus de direcciones del microprocesador para seleccionar de esta manera el puerto con que se comunica el microprocesador.

C/D SEL.- Selecciona control o dato (entrada, activa en 1).

Mediante esta señal el microprocesador define si el dato transferido entre él mismo y el PIO será un dato para control del PIO o un dato destinado al periférico que se conecta a través del PIO, cuando toma el valor de 1, se asume que el dato esta destinado a algún registro de control del PIO y cuando esta en 0 asume que el dato esta destinado al periférico, es común que ésta se conecte directamente a la señal A del bus de direcciones del microprocesador, permitiendo así, con la dirección del puerto de escritura seleccionar entre dato de control o dato de información.

$\overline{CE}$ .- Chip Enable, habilitador de integrado (entrada, activa en 0).

Cuando esta señal se activa (en 0) durante un ciclo de escritura en puerto asume el PIO que la información es para él, ya sea dato o con-



◆ Figura 3.7. Señales de la interfaz Z80-PPIO.

trol en el puerto A o B, y cuando se activa en un ciclo de lectura en -- puerto asume que la lectura será de él, por lo que coloca en el bus de - datos un dato, en sistemas pequeños (que requieran más de 6 dispositivos de este tipo, es común que se conecte a alguna de las líneas A2 hasta A7 y en sistemas más grandes de 7 hasta 26 dispositivos) se conectará a las líneas A2 hasta A7 a través de un decodificador.

$\emptyset$ .- Sistema de Clock (entrada).

El Z80-PIO usa la señal  $\emptyset$  del sistema con el fin de sincronizar -- sus señales internas, ésta es de una sola fase.

$\overline{MI}$ .- Ciclo de máquina uno (entrada, activa en 0).

Esta señal proveniente del microprocesador es usada como una señal de sincronía del PIO con el fin de que este pueda detectar los ciclos -- Fetch del microprocesador cuando ocurre en compañía de la señal RD y detectar así el fin de la rutina, también es empleada para que el PIO sepa cuando le contestan la interrupción, y de esta manera cuando la señal  $\overline{MI}$  se activa junto con la señal  $\overline{IORQ}$  proporciona el vector de interrupción.

Otro uso es proporcionar una señal con reset en el caso de que se active  $\overline{MI}$  en ausencia tanto de  $\overline{RD}$  como de  $\overline{IORQ}$ .

$\overline{IORQ}$ .- Requerimiento de entrada/salida proveniente del Z80-CPU --- (entrada, activa en 0).

• Esta señal es usada junto con las señales A/B, C/D y  $\overline{CE}$  para definir cuando existe transferencia de información entre el microprocesador y el PIO y con la señal  $\overline{MI}$  para saber cuando contestan la interrupción - del PIO.

$\overline{RD}$ .- Señal de lectura proveniente del Z80-CPU (entrada activa en - 0).

Esta señal es empleada junto con las líneas A/B, C/D y  $\overline{CE}$  para definir si la transferencia es del PIO al microprocesador (activa en 0) o bien del microprocesador al PIO (activa en 1).

$\overline{INT}$ .- Solicitud de interrupción (activa en 0 y de colector abierto).

Esta señal se pone en estado bajo cuando solicita una interrupción, y es de colector abierto con el fin de no requerir lógica adicional --- cuando se tiene más de un dispositivo que interrumpe.

IEI.- Habilitador de interrupciones entrada (entrada, activo en 1).

Esta señal es requerida para definir la prioridad de las interrupciones, cuando esta en 1 el PIO puede interrumpir y cuando esta en 0 el PIO no puede interrumpir, permitiendo así a un PIO interrumpir o no según la prioridad de este en el alambrado.

IEO.- Habilitador de interrupciones salida (salida, activa en 1).

Esta señal también es empleada por el PIO para definir la prioridad de interrupciones cuando esta en 1, quiere decir, que el PIO no ha interrumpido ni ningún otro de mayor prioridad, por lo que los de menor prioridad están facultados para interrumpir, cuando está en cero, quiere decir, que el PIO ha interrumpido y/o uno de mayor prioridad por lo que los de menor prioridad no deberán interrumpir.

AO-A7.- Bus del puerto A (bidireccional de tercer estado).

Este bus es empleado para transferir información entre el puerto A del PIO y el dispositivo periférico conectado a él. AO es el bit menor - significativo del bus de datos del puerto A.

A STB.- Almacenar puerto A dato del dispositivo periférico (entrada, activa en 0).

La función de esta señal depende del modo de operación del puerto-A como se indica.

1.- Modo de salida: Un flanco positivo en esta señal indica que el periférico contesta el recibo del dato proporcionado por el puerto A y el PIO puede interrumpir para que el microprocesador proporcione otro -- dato.

2.- Modo de entrada: Esta señal es empleada por el dispositivo periférico para almacenar un dato en el puerto A cuando se termina el período de escritura en el puerto de entrada A (flanco positivo en A STB) - el PIO puede interrumpir para que el microprocesador lea el dato almacenado en el puerto A.

3.- Modo bidireccional: Cuando esta señal se activa el dato almacenado en el registro de salida del puerto A es localizado en las líneas de AO-A7 para que el periférico pueda leer el dato, un flanco positivo - en esta señal indica que el periférico ha terminado de leer el dato por lo que las líneas de AO-A7 se ponen en tercer estado además de quitar el

PIO facultado para solicitar nueva interrupción.

4.- Modo de control, en este modo la señal  $\overline{A\ STB}$  no es empleada.

A RDY.- Registro A listo (salida, activa en 1).

El significado de esta señal depende del modo de operación del --- puerto A del PIO como se indica.

1.- Modo de salida: Esta señal se activa para indicar al periférico que en el puerto de salida A se ha depositado un dato y las líneas de A0-A7 están listas para que en ellas se lea el dato cambiando a 0 cuando la señal  $\overline{A\ STB}$  cambia a 1.

2.- Modo de entrada: Esta señal se activa cuando el registro de -- entrada del puerto A esta vacío (una vez que se ha programado en tal modo o bien una vez que el microprocesador ha leído el dato) y por tal motivo esta listo para aceptar un dato del periférico y se desactivará --- cuando haya leído un dato del periférico mismo que se presentará cuando se desactiva la señal  $\overline{A\ STB}$ .

3.- Modo bidireccional: Esta señal es activa cuando un dato esta - almacenado en el puerto de salida y espera que el periférico lo lea, en este modo el dato no es localizado en las líneas de A0-A7 hasta que ocurra la señal  $\overline{A\ STB}$ .

4.- Modo de control: Esta señal no es empleada y el PIO lo pone en 0 lógico mientras permanece en este modo de operación.

B0-B7.- Bus del puerto B (bidireccional, tres estados).

Tienen la misma función que las líneas A0-A7, salvo que se trata - del puerto B. El puerto B es capaz de suministrar 1.5V a 1.5 mA para manejar transistores Darlington. B0 es el bit menos significativo del bus.

B STB.- Strobe del puerto B (entrada, activa en 0).

Esta señal tiene la misma función en el puerto B que la señal ---  $\overline{A\ STB}$  en el puerto A, salvo en el caso de que el puerto A sea programado en el modo 3 (bidireccional) caso en el que es empleado para almacenar - datos del periférico A conectado en el puerto A y por tal motivo cuando el puerto A se emplea en modo bidireccional sólo se podrá usar el puerto B en modo 4 (bit de control).

B RDY.- Registro B Listo (salida, activa en 1).

Esta señal tiene un uso similar a la señal A RDY, salvo en el caso

de que el puerto A sea programado en modo bidireccional, caso en el que sirve para indicar que el registro de entrada del puerto A esta vacío.

#### PROGRAMACION DEL PIO.

RESET: El Z80-PIO entra en estado de reset cuando se le aplica la alimentación, el estado de reset fuerza las siguientes funciones.

1.- Ambos puertos de mascarar son puestos de manera tal que ningún bit puede interrumpir en modo de control.

2.- Las líneas del bus de datos de los puertos, son puestos en alta impedancia y se activa la señal READY (0) y el modo 1 es automáticamente seleccionado.

3.- El registro del vector no es reseteado.

4.- Ambos bits de interrupción permitida son reseteados.

5.- Ambos puertos de salida son reseteados en adición con el reset automático, el PIO puede ser reseteado tanto por Hardware como por Software, por Hardware con la aparición de la señal  $\overline{M\bar{I}}$  sin la presencia de  $\overline{IORQ}$  y  $\overline{RD}$  por lo que es necesario implementar la señal  $\overline{M\bar{I}}$  PIO si se quiere que obedezca al reset como se indica en la Fig. 3.8.



Figura 3.8. Implementación de la señal  $\overline{M\bar{I}}$  PIO.

El reset por Software se puede lograr como se indica mas adelante.

#### CARGADO DE VECTOR DE INTERRUPCION.

El Z80-PIO ha sido diseñado para operar con el modo 2 la interrupción del Z80-CPU. Este modo requiere que un vector sea proporcionado por el dispositivo que ha interrumpido cuando el microprocesador le conteste la interrupción, colocando el dispositivo de mayor prioridad en el bus de datos en el momento que se contesta la interrupción, éste vector podrá ser almacenado en el puerto de control correspondiente cuando se ---

carga en él un dato con el siguiente formato.

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

El bit menos significativo deberá ser 0.

SELECCION DEL MODO DE OPERACION.

El puerto A del PIO puede ser operado en cualquiera de sus 4 modos de operación, 0 salida, 1 entrada, 2 bidireccional y 3 de control. - Correspondiéndole a cada modo un número de 0 a 3.

El modo de operación se define escribiendo en el puerto de control un Byte con el siguiente formado.

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1

Modo seleccionado. Indicador de selección del modo de operación.

Modo	M1	M0
Salida 0	0	0
Entrada	0	1
Bidireccional 2	1	0
Control 3	1	1

Los bits D7 y D6 indican el No. de modo seleccionado, los bits D5- y D4 son ignorados y los bits D3-D0 deberán estar en 1111 para indicar - que se está escribiendo el modo de operación.

#### MODO 0.

Cuando se selecciona el modo 0 se habilita cualquier dato escrito en el puerto de salida por la CPU y es localizado en el bus de comunicación con el periférico, el contenido del puerto de salida puede ser cambiado en cualquier tiempo por la CPU simplemente escribiendo en el puerto un nuevo dato, y el contenido del registro de salida puede ser leído por la CPU con la ejecución de una instrucción INPUT (entrada) que ejecute el microprocesador en la dirección del puerto.

Si es escrito por el microprocesador un dato en un puerto programado en el modo 0 se activa la señal READY del handshake para informar - al periférico que puede recoger un dato, esta señal continua activada -- hasta que la señal de atrapado (strobe) proveniente del periférico sea - recibido, el flanco de terminación del strobe provoca que se desactive - la línea READY y el PIO solicite una interrupción si éstas han sido ha-- bilitadas para el puerto en cuestión, esta simple forma de protocolo es-- empleado en la mayoría de dispositivos periféricos.

#### MODO 1.

Si es seleccionado el modo 1, se define la operación de entrada al inicio de la operación del handshake, la CPU solamente deberá realizar - operaciones de lectura en el puerto, en este modo se activará la señal - de READY para indicar al periférico que puede almacenar datos en el re-- gistro de entrada vacío.

El periférico almacenará el dato activando la señal de strobe hacia con el flanco de fin de escritura en la línea de strobe, causa una inte-- rrupción (si éstas están habilitadas) y es desactivada la señal de ---- READY. Es posible almacenar datos en el registro de entrada sin hacer -- caso del estado de la señal READY, si esto ocurre se presenta una condi-- ción de sobreflujo de datos.

#### MODO 2.

El modo 2 es un tipo de transferencia de datos bidireccionales em-



pleando las 4 líneas del handshake, únicamente el puerto A puede ser --- usado en el modo 2, en la operación del modo 2 son empleadas las señales del handshake del puerto A para el control del puerto de salida y las -- señales del handshake del puerto B para el control del puerto de entra-- da. Esto en ambas señales A RDY y B RDY, pueden ser activadas simultá--- neamente. La única diferencia de operación del modo 2 con respecto a los modos 0 y 1, es que el modo 0 difiere en la operación de salida del modo 2 en que el dato es localizado en el bus de transferencia únicamente con la presencia de la señal A STB y no constantemente como ocurre en el mo-- do 0.

### MODO 3.

El modo 3 implica entrada y salida de datos sin protocolo, no se - utilizan las líneas de protocolo. En el modo 3 cada pata terminal de los puertos A y B se deber definir como entrada o como salida. Una vez que - se especifica el modo 3, los datos se pueden leer o escribir en un puer-- to en cualquier momento. Las señales de protocolo no están activas; las-- señales STB no se usan y las señales RDY siempre tienen nivel bajo. Los-- datos de salida al puerto afectarán únicamente a las líneas programadas-- como salidas, mientras que los datos de entrada se cargarán en todas las líneas, incluyendo aquellas programadas como salidas. Si todas las ter-- minales de un puerto se programan para una misma dirección, entonces el-- puerto se puede usar como un simple puerto de entrada o de salida de 8 - bits en paralelo.

Cuando seleccionamos este modo, la próxima palabra de control de--- berá ponerse en el Registro I/O para indicar que líneas serán tomadas -- como entradas y cuales como salidas, esto es:

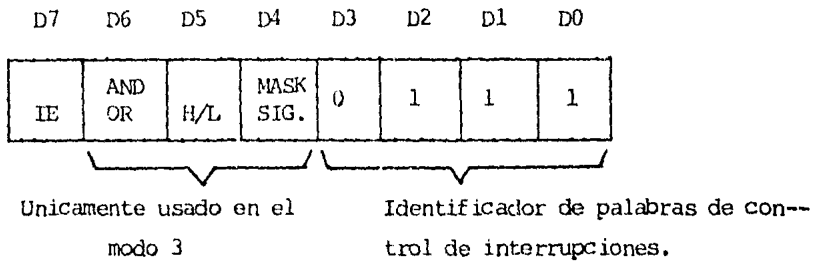
I/O = 1; poner bit a la entrada.

I/O = 0; poner bit a la salida.

D7	D6	D5	D4	D3	D2	D1	D0
I/07	I/06	I/05	I/04	I/03	I/02	I/01	I/00

CONJUNTO DE PALABRAS DE CONTROL PARA INTERRUPCION.

La palabra de control de interrupciones para el puerto tiene el siguiente formato:



Si el bit = 7 el Flip-Flop habilitador de interrupciones en 1 y el puerto puede generar interrupciones, si D7 = 0 el Flip-Flop habilitador de interrupciones, es puesto en 0 y no serán generadas las interrupciones.

Si una interrupción ocurre cuando el Flip-Flop habilitador de interrupciones esta en 0, la interrupción es almacenada internamente y esta se indicará al microprocesador cuando el Flip-Flop habilitador sea puesto en 1. Los bits D6, D5 y D4 son usados únicamente con el modo de operación 3, sin embargo, poniendo el bit 4 (enl) en la palabra de control de interrupciones durante cualquier modo de operación causará que cualquier interrupción pendiente sea reseteada. Estos 3 bits son usados para definir los criterios de interrupción en el modo 3 cuando cualquier grupo de líneas de entrada/salida tengan un estado definido D6 (AND/OR) define la función lógica que se formará con las líneas monitoreadas, si el bit 6 = 1 se especifica la función AND y se le asigna 0, se especifica la función OR esto es cuando D6 = 1, se genera una interrupción sólo cuando todas las líneas con facultad de interrumpir lo hagan y cuando D6 = 0 se genera la interrupción cuando cualquier línea facultada para interrumpir lo haga.

El bit D5 define el estado con que las líneas monitoreadas interrumpirán si el bit D5 = 1, las líneas interrumpirán cuando estén en 1 lógico y cuando D5 = 0 interrumpirán con 0 lógico.

Si el bit D4 = 1 la siguiente palabra de control escrita en el PIO define la mascara de interrupción como sigue:

D7	D6	D5	D4	D3	D2	D1	D0
MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0

Unicamente las líneas en que el bit de la mascara es 0 serán monitoreadas para generar una interrupción.

El Flip-Flop habilitador puede ser modificado sin alterar los bits 6, 5 y 4 con la siguiente palabra de control.

D7	D6	D5	D4	D3	D2	D1	D0
IE	X	X	X	0	0	1	1

### MODO 3.

Este modo no emplea las señales del handshake pudiéndose ejecutar instrucciones de lectura o escritura en cualquier tiempo.

Una interrupción se generará si éstas son habilitadas y además se cumple con la condición de la mascara (8 bits de mascara y 2 de estado y función).

Otra interrupción no será generada hasta que ocurra un cambio en la función de mascara. Esto es, la función debe cambiar de falso a verdadero para que ocurra una interrupción.

Una vez que se ha descrito al Z80-PIO, daremos a conocer la forma de como fué interfazado el PIO con las bases de los circuitos integrados a probar, como se muestra en la Fig. 3.9.

Para nuestro caso, como lo único que nos importa es la entrada y salida de datos para probar los circuitos integrados sólo se utilizaron los 2 puertos E/S paralelo programándolo en el Modo 3 (ver el apartado 3.3).

Como se puede ver en la Fig. 3.9, se encuentran disponibles 7 "bases" (wire wrap sockets). Estas bases, que aunque en la Fig. 3.9 no se pueden apreciar bien se encuentran comprendidos en dos grupos: el primer grupo cuenta con 4 bases de 14 patas y el segundo cuenta con 3 bases de 16 patas. El lector podrá hacerse aquí una pregunta. Porque 7 bases?, si sólo se van a probar dos tipos de circuitos integrados, de 14

y 16 patas y por consiguiente sólo se utilizarían dos. La respuesta es - la siguiente: la mayor parte de los CIs TTL a probar llevan la alimentación en las patas 7 (GND) y 14 (VCC) para CIs de 14 patas y 8 (GND) y 16 (VCC), para CIs de 16 patas, pero existen otros (en ambos tipos de CIs)- que su alimentación no es a través de éstas, lo que hace que sus líneas- de alimentación no sean comunes para todos con respecto a patas. Sólo 3 pares de bases coinciden como se muestra en la Fig. 3.9, esto es, la base 1 y 2 coinciden en VCC (pata 4), la base 1 con la 3 en GND (pata 10), la base 3, 5 y 6 en VCC (pata 5) y las demás líneas de alimentación no - son comunes a ninguna otra, por lo que al igual que las obras se conec-- tan a las líneas de alimentación del sistema más cercanas.

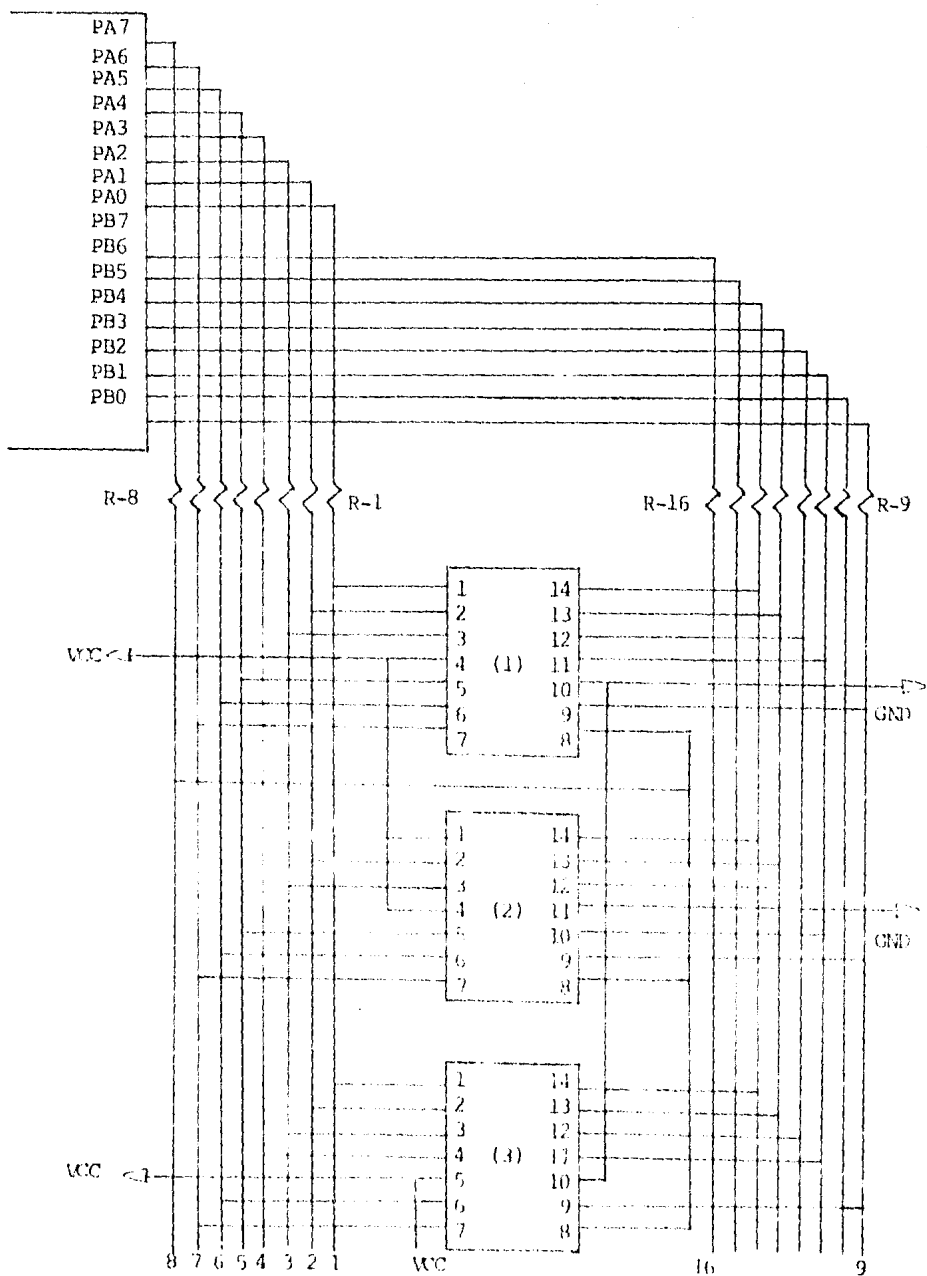
La relación de CIs de acuerdo a su alimentación se da el Apéndice- A (Manual del Usuario).

Los dos puertos del PIO, llamados puerto A y puerto B, son alam--- brados para las 16 posiciones de patas de las bases de las mismas, que - como ya se mencionó sirven como la posición prueba.

Un CI de 16 patas llena las 16 posiciones enteras de prueba, ya -- que el PIO tiene 2 puertos E/S de 8 bits independientes, haciendo un total de 16 bits, un CI de 14 patas es puesto a través de las patas 1-7 -- (con PA0-PA6) y 9-14 (con PA7 y PB0-PB5).

La Fig. 3.9 muestra la correspondencia entre las líneas de puerto- de datos del PIO, PA0-PA7 y PB0-PB7 y número de patas para CIs a prueba- de 14 y 16 patas, respectivamente.

Otras de las cosas que se puede observar en la Fig. 3.9 es el uso de una resistencia R entre cada una de las líneas de ambos puertos del - PIO y las patas de los CIs a probar (14 y 16 patas). Estas son u operan- como una medida de seguridad, tanto para el Z80-PIO como para los CIs a probar. El análisis se hizo en base a experiencias y un ejemplo es el -- siguiente: si tenemos a la salida de un CI cualquiera (en este caso el - de una compuerta AND) un 1 lógico y en la entrada del PIO un 0 lógico o viciversa, como se muestra en la Fig. 3.10, la corriente circulará de 1 a 0 (en ambos casos), dañando con esto al PIO o a el CI a probar. Por lo tanto, para evitar posibles averías y que vendría a causarnos problemas- muy graves, se hizo un estudio de lo que sería el método de prueba y por lo tanto previendo las posibles fallas del usuario, esto es, en el caso-



Todas las resistencias son de 3.3K (R1-R16)

Figura 2.2. Interfaz Z80-PPIO con las bases de los CIs a probar.

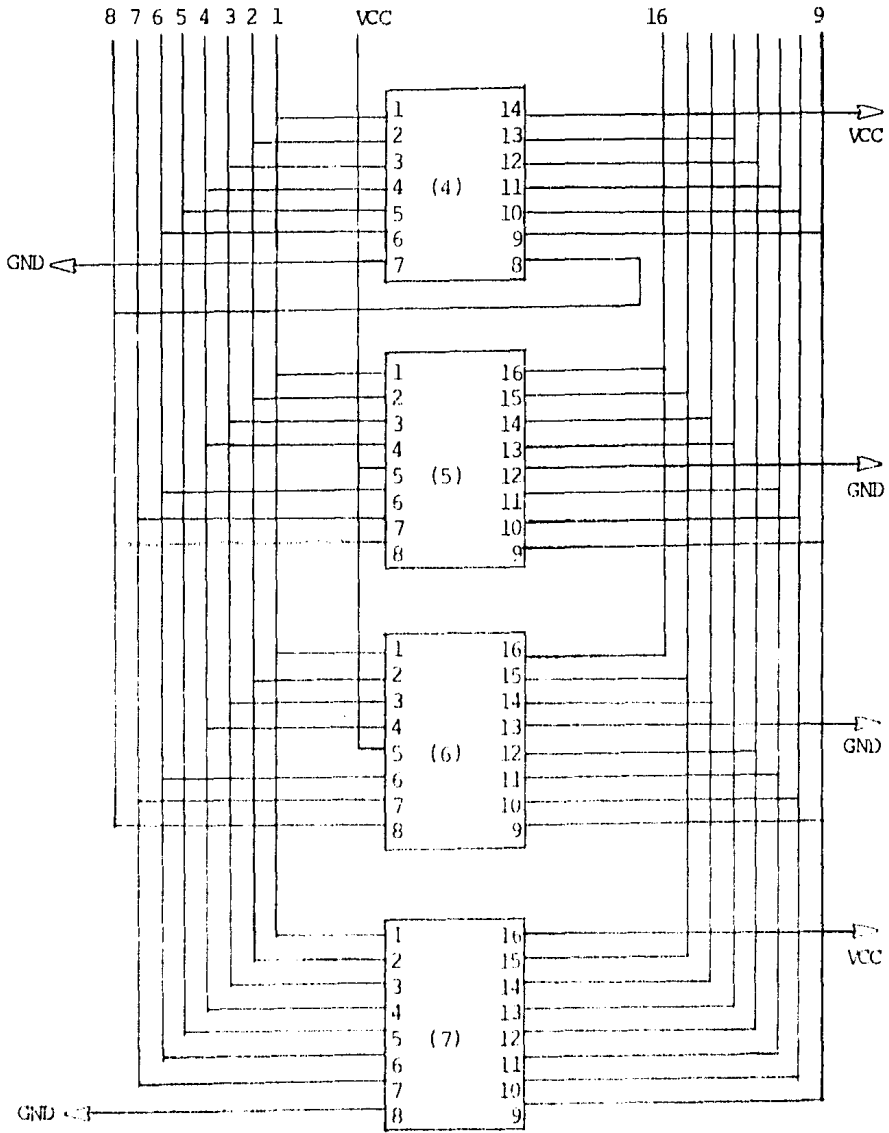


Figura 3.9. (CONTINUACION)

de que el CI a prueba no fuera el deseado por este, ya sea por no estar muy claro el número del CI o por equivocación.

El valor de la resistencia (el mismo valor para la de todas las -- líneas, ya que realizan la misma función) se calculó en relación a lo -- siguiente: como la diferencia de potencial es + 5 V entre PIO y CI a --- probar y la corriente máxima que puede consumir el PIO es 1.5 mA, por lo tanto el valor de la resistencia R es:

$$R = \frac{5 \text{ V}}{1.5 \text{ mA}} = 3.2 \text{ K}\Omega$$

y por lo tanto para R0-R7 y R8-R15

$$R = 3.3 \text{ K}\Omega \text{ (valor comercial)}$$

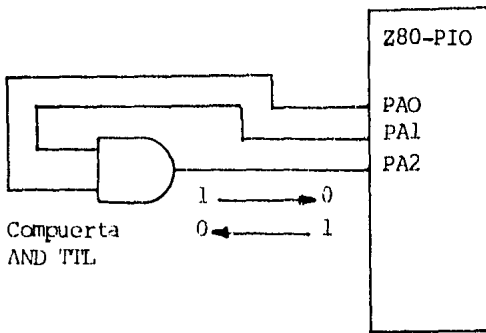


Figura 3.10. Configuración salida (CI) y entrada (PIO).

### 3.3.2 CONFIGURACION DE LOS CIRCUITOS INTEGRADOS A PRUEBA.

Es conveniente también mencionar los pasos que se siguieron para la configuración de los CIs a prueba.

Como ya se mencionó, ambos puertos A y B del PIO son operados en --

el MODO 3, esto es, si un puerto del Z80-PIO se programa en el MODO 3, entonces, el siguiente byte que se envíe a ese puerto después del Código de control se considerará como la palabra de control que especifica la Máscara de Dirección de la transferencia para cada una de las patas terminales del puerto. Un nivel 1 en un bit de la palabra implica que la pata terminal correspondiente es de entrada, mientras que un cero -- implica que es de salida. El número del bit de la palabra corresponde -- al número de línea del puerto, es decir, el bit 0 tiene correspondencia con la línea 0, el bit 1 con la línea 1, etc.

Los datos de salida al puerto afectan sólo a las líneas programadas como salidas, mientras que los datos de entrada se cargan en todas las líneas, incluyendo las programadas como salidas. Por consiguiente -- cada una de las 16 líneas de datos del puerto es configurado por una -- vía de entrada o salida de byte de control. Llamando al significado de los bits puestos en la configuración I/O del byte de control del PIO -- como sigue:

Bit DN= $\emptyset$  en el puerto A de I/O, el byte de control implica que la línea PAN es una línea de salida.

Bit DN=1 en el puerto A de I/O, el byte de control implica que la línea PAN es una línea de entrada.

Bit DN= $\emptyset$  en el puerto B de I/O, el byte de control implica que la línea PBN es una línea de salida.

Bit DN=1 en el puerto B de I/O, el byte de control implica que la línea PBN es una línea de entrada.

La configuración E/S de la palabra o byte de control es un parámetro especificado para el uso del Probador de CIs Digitales para describir las características importantes del CI a ser probado.

El siguiente procedimiento es empleado para asociar la palabra de control con las propiedades de un CI.

1.- Las patas de entrada en el CI bajo prueba son consideradas -- como salidas de la microcomputadora. Por lo tanto, las líneas del puerto del PIO son asignadas a valores de  $\emptyset$  lógico en la configuración de la palabra control.

2.- Las patas de salida en el CI bajo prueba son consideradas como entradas de la microcomputadora. Por lo tanto, las líneas--



del puerto del PIO son asignadas a valores de 1 lógico en la configuración de la palabra control.

- 3.- Las patas de potencia en el CI, +5V (VCC) y tierra (GND) son consideradas como salidas de la microcomputadora. Por lo tanto, las líneas del puerto del PIO son asignadas a valores de 0 lógico en la configuración de la palabra de control. Notando que las patas de potencia deben estar correctamente a +5V y tierra.
- 4.- Las patas "NO CONECTADAS" (NC) en el CI bajo prueba son consideradas como salidas a la microcomputadora. Por lo tanto, sus bits en la palabra de control son puestos en 0 lógico.
- 5.- Para los CIs de 14 patas bajo prueba, los bits PB6 y PB7 son cada uno consideradas como salidas a la microcomputadora. Por lo tanto asigna un 0 lógico en la configuración de la palabra de control

Una vez habiendo determinado las propiedades de los CIs a probar o sea, los bits de cada una de las 2 configuraciones de la palabra de control, una para el puerto A y la otra para el puerto B. Los transmisores usan esta información para probar el CI por la vía de dos localidades de memoria. Una localidad carga la configuración de la palabra de control en el puerto A y la otra en el puerto B.

El siguiente punto es dar a conocer la descripción de las tablas de prueba y de las partes que la componen.

- A) Definición de E/S del PIO. Esto es, la palabra de control para programar el PIO, un cero (0) indica que es salida del PIO y entrada al CI, un uno (1) indica que es entrada del PIO y salida del CI.
- B) Condiciones Iniciales. Estas son para operar algunos CIs, ya que la mayoría de los CIs a probar no las necesitan. También debe especificarse el número de condiciones iniciales.
- C) PRUEBA. Aquí es donde se va a probar el CI, por lo tanto aquí están contenidos los datos de entrada y salida (PIO), las salidas indican la entrada a el CI y las entradas indican la salida del CI, también aquí se especifica el número de pruebas a realizar.

### 3.4 TARJETA CONTROL DE DISPLAYS (CONT-DIS).

#### 3.4.1 INTRODUCCION.

La tarjeta para controlar Displays está diseñada para manejar la información de 64 displays del tipo 7 segmentos ánodo común o bien 16 del tipo matriz de 7X5.

Esto permite gran flexibilidad en diversas aplicaciones.

Dos ventajas principales de esta tarjeta son:

a).- La compatibilidad con computadoras que posean el estandar -- industrial del BUS-S100.

b).- La posibilidad de cambiar direcciones para el control debido a un conjunto de pequeños interruptores (dip switch).

La alimentación que requiere esta tarjeta es de +8.0 volts solamente.

#### 3.4.2 USOS DE LA TARJETA.

La tarjeta de controlar displays (cont-dis) permite conectar la CPU con 64 displays de 7 segmentos de ánodo común o bien 16 de 7X5.

La utilidad de esta tarjeta es amplia puesto que la información que pueden proporcionar, la maneja fácilmente la CPU ya que la ve como una memoria de 64 Bytes.

En el caso del Probador de Circuitos Integrados Digitales, aunque los comandos que se introducen a el programa principal (para la realización de la prueba), son comandos con variables numéricas y por lo tanto sin ningún problema para ser mostrados por los displays que traconsigo el Starter. El problema radica al realizarse la operación de prueba, dar como resultado el mensaje de la misma, siendo éstos resultados los siguientes: BUENO, MALO Y DESCONOCIDO. Esto es, dependiendo del estado en que se encuentre el CI.

Como se puede observar el máximo número de caracteres de uno de los mensajes (DESCONOCIDO) es 11, mensaje que con los exhibidores del Starter no se tendría una exhibición del todo completa en toda su forma.

Además de que los exhibidores del Starter (Fig. 3.4) es relativamente lento y exige mucha atención por parte de la CPU y la memoria de la misma, ya que el demultiplexor de cada uno de los exhibidores corre-

por cuenta del circuito principal.

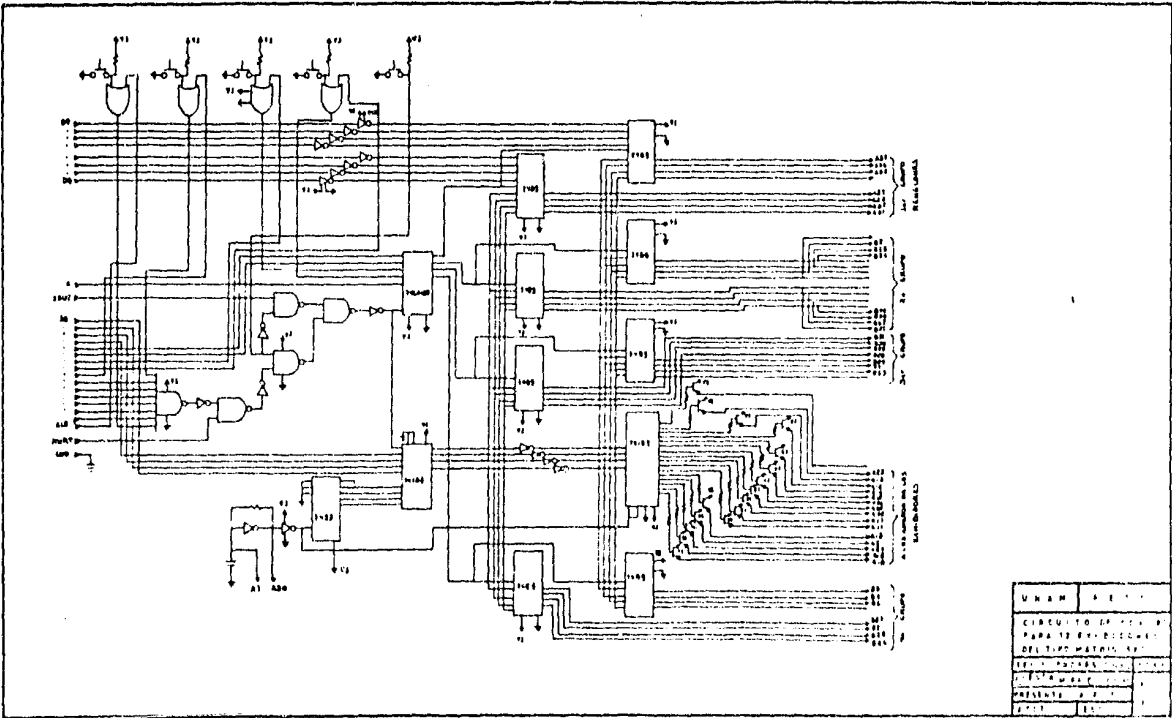
Por lo que tomando estas desventajas se pensó en el uso de la --- tarjeta de controlar displays (cont-dis) por las características de --- aplicación para manejar información. Conectando a éste una tarjeta con 12 displays del tipo matriz de 7x5 mediante dos conectores de 40 patas- (A y B), permitiendo así mostrar al usuario el mensaje en forma comple- ta del CI a ser probado.

El circuito para controlar los exhibidores para conectarse a un - bus-S100, se presenta en la Fig. 3.11, mientras que la conexión de los exhibidores se ve en la Fig. 3.12.

### 3.4.3 INSTRUCTIVO DE OPERACION.

Para operar la tarjeta de controlar displays es necesario tener - un programa con el cual se indica lo que se desea escribir, dicho pro- grama requiere la dirección de memoria correspondiente de los displays- (para nuestro caso, ver capítulo 4), la cual puede ser modificada debi- do a que un conjunto de pequeños interruptores toman valores de 0 ó 1 - lógico para determinar la dirección que uno elija, como se muestra en - la siguiente tabla (Tabla 3.1).

SW2	SW4	SW5	SW6	SW7	DIRECCIONES
0	0	0	0	1	FF00-FF3F
0	0	0	1	1	FF80-FFBE
0	0	1	0	1	FF40-FF7E
0	0	1	1	1	FFC0-FFFE
0	1	0	0	1	FE00-FE3E
0	1	0	1	1	FE80-FEBE
0	1	1	0	1	FE40-FE7E
0	1	1	1	1	FEC0-FEFE
1	0	0	0	1	7F00-7F3F
1	0	0	1	1	7F80-7FBE
1	0	1	0	1	7F40-7F7E
1	0	1	1	1	7FC0-7FFE
1	1	0	0	1	7E00-7E3E
1	1	0	1	1	7E80-7EBE



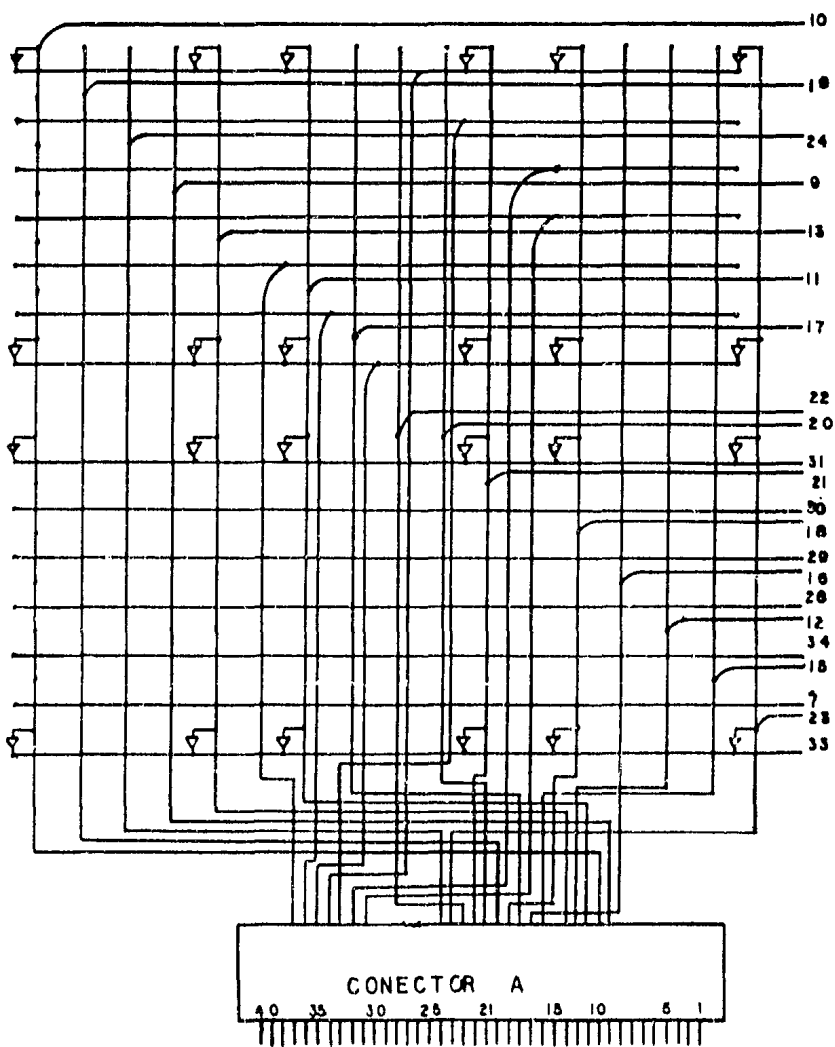
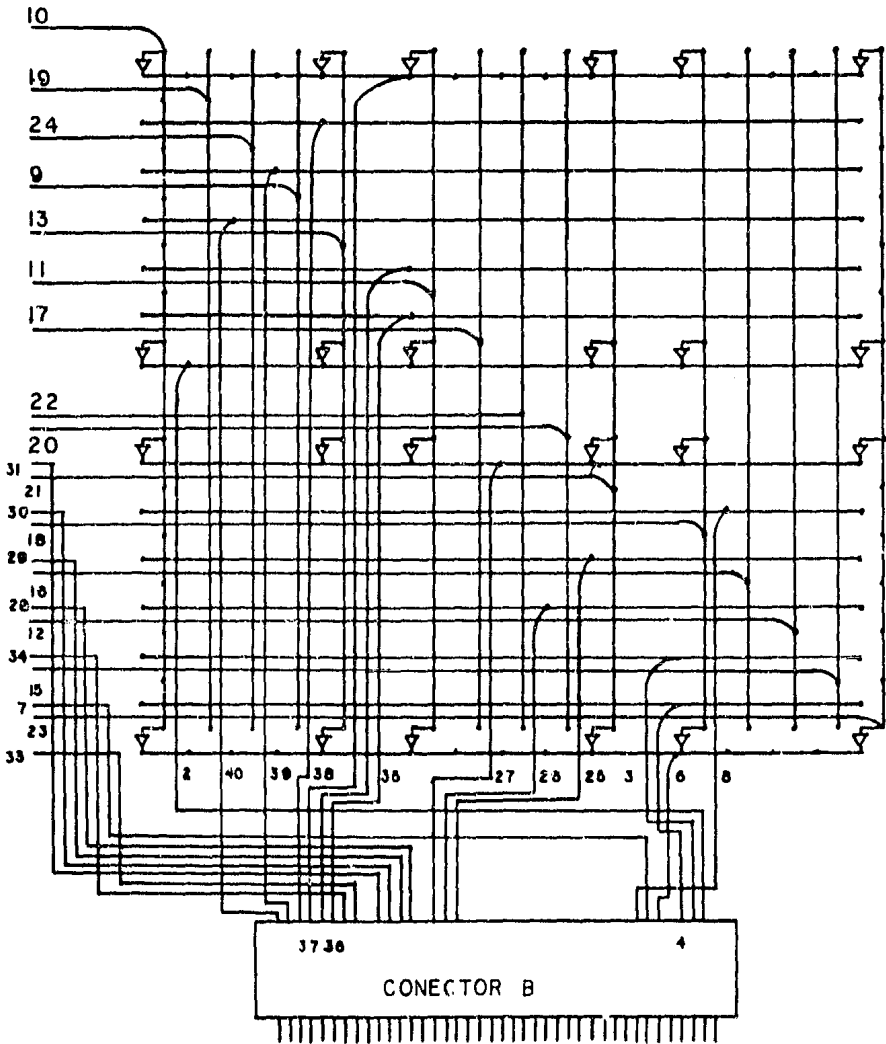


FIG. 3.12. CONEXION DE EXHIBIDORES. FES-C. UNAM.

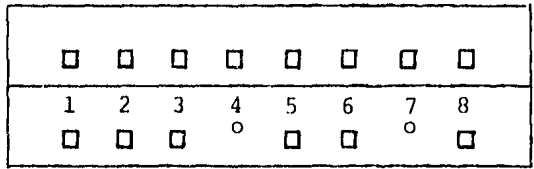


CONTINUACION

1	1	1	0	1	7E40-7E7E
1	1	1	1	1	7EC0-7EFE

Tabla 3.1 Posición de los interruptores y dirección correspondiente para el Display Matricial.

NOTA: Un interruptor está en 1 lógico si se oprime la parte inferior de este, como se muestra en la Fig. 3.13, para la dirección FE00-FF3F.



o.- indica sonido (1 lógico).

Figura 3.13. Posición de los interruptores en el DIP SWITCH para la dirección FE00-FE3F.

Para indicar el dato a escribir en el Display se considera que -- cada segmento del display corresponde a un bit del dato escrito como se indica en la Tabla 3.2.

Segmento	Bit
a	0
b	1
c	2
d	3
e	4
f	5
g	6
dp	7

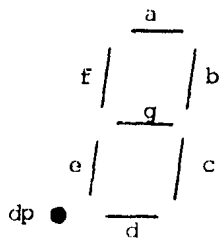
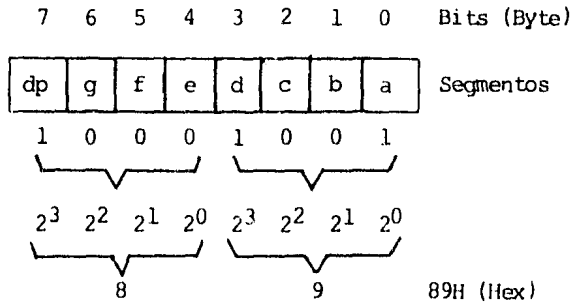


Tabla 3.2. Relación segmento del Display a un bit del dato escrito.

Una consideración que se debe hacer antes de continuar con el modo de operación es la siguiente: un segmento se enciende si se escribe un "0" lógico en el bit correspondiente y se apaga cuando se escribe un "1" lógico.

Ejemplo: Cuando se quiere escribir H O L A en los displays se hace un programa que contenga lo siguiente:

- Se inicialice la memoria que corresponda a los displays.
- Se cargue en el registro A el número que previamente se obtenga de cada letra, es decir, para H son los segmentos: f, g, b, e y c, por lo tanto:



Por lo que se cargaría 89 para escribir H y así sucesivamente con cualquier letra que se desee escribir en los displays.

Para indicar el dato a escribir en un Display del tipo matriz de 7x5 (como el utilizado por nosotros), el procedimiento es el mismo del ejemplo anterior, el cual utiliza Displays de 7 segmentos, con la diferencia de que al escribir un número o letra se cargarían cinco datos -- hexadecimales, ya que cada columna es una palabra de 8 bit y por consiguiente a cada una le corresponde una dirección, considerando el LED superior como el bit 0 y el bit inferior como el bit 6, como se muestra en la Tabla 3.3.

Como se puede ver el bit 7 no le corresponde ningún LED de las -- columnas, por lo que puede tomar cualquier valor 0 ó 1 lógico.



		DIRECCION				
		0	1	2	3	4
Bit	0	0	0	0	0	0
"	1	0	0	0	0	0
"	2	0	0	0	0	0
"	3	0	0	0	0	0
"	4	0	0	0	0	0
"	5	0	0	0	0	0
"	6	0	0	0	0	0
"	7					

Tabla 3.3 Relación bits-columnas (Dirección) de la matriz de -- LEDs de 7x5.

Como se puede ver el bit: 7 no le corresponde ningún LED de las -- columnas, por lo que puede tomar cualquier valor 0 ó 1 lógico.

#### 3.4.4 FUNCIONAMIENTO A BLOQUES.

En la Fig. 3.14 se muestra el diagrama a bloques de la tarjeta -- (cont-dis) y en la Fig. 3.11 se muestra el diagrama del circuito.

- Las líneas del bus de control provienen del Bus S-100. Se activa la MEMWR para acceder a la localidad de memoria requerida en la es--critura.

- El bus de direcciones corresponde a las 16 líneas, con las cuales se indica la dirección de memoria en donde se encuentra la informa--ción.

- Con las líneas A15-A8 del bus de direcciones se selecciona la --operación de la tarjeta entre escritura en la memoria o bien de refres--co de displays.

- Con las líneas A0 a la A5 se selecciona la localidad de memoria del nuevo dato.

- La operación de refresco tienen como mostrar en los displays o columnas el contenido de las localidades de memoria, acción que es mul--tiplexada en tiempo y espacio.

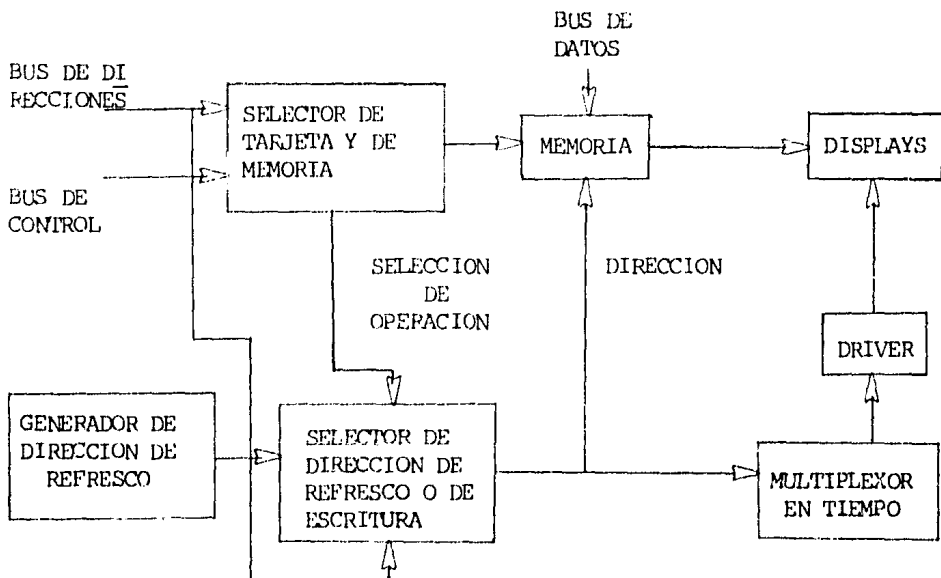


Figura 3.14 Diagrama a bloques de la tarjeta (cont-dis).

- Para realizar la función de refresco de los displays se tiene un oscilador que genera la dirección de refresco (de 0 a 15) la cual se deshabilita durante la escritura.

- Existen 16 tiempos de refresco correspondiendo cada uno de estos tiempos a 4 displays o columnas necesarios para tener 48 líneas; 16 indican a que grupo de displays o columnas se le concede tiempo de refresco y 4 grupos de 8 líneas que contienen la información de los segmentos o LEDs por encender en cada uno de los 4 displays o columnas.

## CAPITULO 4

### DISEÑO DEL PROGRAMA NECESARIO.

#### 4.1 DEFINICION DEL SOFTWARE.

En este capítulo se definen algunos conceptos básicos de programación (Software), antes de analizar y describir el diseño del programa principal del Probador de Circuitos Integrados Digitales y rutinas que la componen. Por lo que primeramente es conveniente definir el término Software mas detalladamente que la ya mencionada en capítulos anteriores como: "Lógica Programada". Considerando además que en los capítulos 2 y 3 se dan y hacen mención de los circuitos básicos (Hardware) que componen a una microcomputadora.

SOFTWARE.- Es la totalidad de programas y rutinas usadas para extender la capacidad de computadoras, tal como compiladores, rutinas y subrutinas. Contrastadas con el Hardware.

#### 4.2 CONCEPTOS BASICOS DE PROGRAMACION.

##### 4.2.1 OPERACION

Una operación es una acción específica que realiza el microprocesador siempre que ejecuta una instrucción.

Existen diferentes operaciones que puede realizar el microprocesador. Las operaciones asociadas con el microprocesador Z-80 son:

- 1) Transferencia de información.
- 2) Operaciones Aritméticas.
- 3) Operaciones Lógicas.
- 4) Subrutinas.

- 5) Entrada y Salida.
- 6) Incremento y Decremento.
- 7) Saltos.
- 8) Probar, poner y limpiar bits.
- 9) Otras, como ajuste a decimal e interrupciones.

#### 4.2.2 INSTRUCCION.

Una instrucción es un patrón de bits que ordena a la microcomputadora a ejecutar una operación específica. En la referencia 5 del apéndice B se pueden ver las instrucciones de la Z-80 con sus respectivos códigos.

El grupo de instrucciones de un microprocesador es simplemente un conjunto de patrones de bits que generan una serie de acciones definidas en la CPU. Cada microprocesador tiene un número de instrucciones en particular y cada instrucción de ésta un patrón. Algunas instrucciones se encuentran implementadas en la mayoría de los microprocesadores pero probablemente tendrán diferente patrón de bits y/o mnemónicos (nombre de las instrucciones).

Las instrucciones se obtienen de la memoria durante un ciclo ---- Fetch y se cargan en el registro de instrucción, pasando a continuación al Decodificador de Instrucciones para su interpretación y general una acción específica. Cada instrucción o patrón de bits genera una acción invariable, por ejemplo, el microprocesador Z-80 interpreta al patrón de bits  $\beta 1111000$  como la instrucción para

"Mover el contenido del registro B al Acumulador".

El microprocesador (como cualquier otra computadora) únicamente reconoce patrones binarios como instrucciones o datos no reconoce números octales, hexadecimales o decimales.

#### 4.2.3 PROGRAMA.

Un programa es un grupo de instrucciones organizadas en una forma secuencial que comandan a la CPU a realizar funciones más complejas que las que se logran con las simples instrucciones. La organización adecuada de las instrucciones permite al programador ordenar la ejecución-

de muchos y diferentes cálculos o procesamiento con los datos de su interés y obtener los resultados en una forma que le es útil. En algunos casos, se requiere de complicadas funciones matemáticas, mientras que en otros casos se requieren de funciones sencillas.

Un programa puede estar formada no sólo por instrucciones sino -- también por datos y direcciones los cuales se requieren para realizar -- las tareas especificadas por las instrucciones. Todos los microprocesadores ejecutan las instrucciones secuencialmente a menos que una ins--- trucción de salto cambie la secuencia de ejecución.

#### 4.2.4 LENGUAJE DE MAQUINA.

En síntesis, se puede decir que un programa está formado por un -- grupo de patrones binarios. Por ejemplo, para sumar los datos de las -- localidades 1500H y 3020 y almacenar el resultado en la localidad 3000H el programa puede ser:

```
0011 1010
0000 0000
0001 0101
0100 0111
0011 1010
0010 0000
0011 0000
1000 0000
0011 0010
0000 0000
0011 0000
```

A estos patrones binarios se le conoce como "lenguaje de máquina" ya que es el único lenguaje que entienden las computadoras.

La palabra de la 2-80 es de 8 bits y con 8 bits se pueden generar 256 patrones binarios diferentes.

Existen algunos inconvenientes para el usuario cuando trabaja con el lenguaje de máquina. Entre ellos se puede mencionar:

- 1.- Es muy difícil para el programador entender o depurar los --- programas.

- 2.- El proceso de cargar el programa en la computadora es muy --- lento, ya que tiene que alimentar bit a bit (se recomienda -- ver apartado 4.2.5).
- 3.- El programa no describe la tarea que realiza en una forma fá- cil para el usuario.
- 4.- Los programas largos son cansados para su elaboración.
- 5.- Existe una gran posibilidad de que el programador se equivo-- que en la redacción del programa y una gran dificultad en en- contrar los errores.

Si en el programa anterior existiere un bit erróneo sería labo--- rioso detectarlo y corregirlo.

#### 4.2.5 SISTEMA HEXADECIMAL.

Aunque las computadoras manejan cantidades binarias las personas-- no. El programador puede recurrir al auxilio del sistema hexadecimal -- para manejar el lenguaje de máquina, los códigos de operación. Este --- sistema consiste en agrupar los dígitos binarios de cuatro en cuatro y-- "representarlos" por un dígito del 0 al 9 ó una letra de la A a la F -- dependiendo el valor de los cuatro bits. La tabla 4.1 muestra la con--- versión de Binario a Hexadecimal y viceversa. Se le da el nombre de he- xadecimal debido a que con cuatro bits se pueden realizar 16 combina--- ciones (base 16). El programa anterior quedaría en la forma siguiente - usando el sistema hexadecimal.

3AH  
 00H  
 15H  
 47H  
 3AH  
 20H  
 30H  
 80H  
 32H  
 00H  
 30H

Utilizando el sistema hexadecimal para representar los patrones binarios de las instrucciones, el programador tiene una probabilidad -- menor de equivocarse, que trabajando con los patrones binarios de las -- instrucciones. Se acostumbra poner la letra H (mayúscula) después de -- una cantidad hexadecimal para identificar que es una cantidad hexadecimal.

BINARIO	HEX
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Tabla 4.1 Conversión Binario/Hexadecimal.

El problema ahora es como cargar los números hexadecimales ya que el microprocesador únicamente entiende patrones binarios. La solución -- es la de contar con un método para convertir números hexadecimales a -- binarios. Esta solución se logra con los programas llamados "Cargadores hexadecimales" y que generalmente los proporcionan los fabricantes de -- microcomputadoras.

El procedimiento para cargar un programa en hexadecimal consiste -- primero en llamar a ejecución el "Cargador hexadecimal" y posteriormente -- alimentar los números hexadecimales a la microcomputadora por medio--

de una terminal de entrada para que los procese el cargador y los cargue en la memoria.

#### 4.2.6 MNEMONICO.

Un método de mejorar el manejo de las instrucciones para el programador es el de asignar un nombre a cada instrucción. Por ejemplo, a la instrucción de decrementar nombrarla como "DEC" y a la de mover un dato nombrarla como "LD". A los nombres de las instrucciones se les conoce como "mnemónicos". Los mnemónicos se escogen de tal manera que orienten en forma fácil al programador la función que realiza la instrucción que representa. Es más fácil recordar que INC A es el mnemónico de la instrucción para incrementar en uno el contenido del Acumulador que recordar que 3CH es el código de operación de la misma instrucción.

Cada fabricante de un microprocesador proporciona una serie de mnemónicos para la serie de instrucciones del microprocesador.

El programa anterior queda en la forma siguiente al usar los mnemónicos.

```
LD  A, (1500H)
LD  B,A
LD  A, (3020H)
ADD B
LD  (3000H),A
```

Esta representación del programa es más comprensible que los anteriores usando dígitos hexadecimales y patrones binarios. En este caso ya se puede identificar que partes son instrucciones y que partes son direcciones o datos. La representación de las instrucciones de un programa usando mnemónicos se le conoce como "programa fuente".

#### 4.2.7 PROGRAMA ENSAMBLADOR.

Una vez que el programador tiene en papel su programa utilizando mnemónicos se le presenta el problema de cargarlo en la memoria. Una forma de resolver esto es "traduciendo" cada instrucción a su código de operación hexadecimal y utilizando después el "Cargador hexadecimal".



El ejemplo anterior quedaría.

3A0015H	LD	A, (1500H)
47H	LD	B,A
3A2030H	LD	A, (3020H)
80H	ADD	B
320030H		3000H

Al proceso de traducir o convertir cada mnemónico de las instrucciones de un programa a su código de operación se conoce como "ensamblar el programa" y al resultado se le conoce como "programa objeto".

El proceso de ensamblar un programa instrucción por instrucción es tedioso y por lo mismo se pueden cometer errores. Esto se complica debido a que existen instrucciones de una, dos tres y cuatro palabras.

Para solucionar este problema los fabricantes de microcomputadoras proporcionan un programa que permite traducir los mnemónicos de las instrucciones a los patrones binarios o códigos de operación. Este programa se conoce como "Ensamblador". En otras palabras, el programa Ensamblador traduce un programa fuente a un programa objeto. Cualquier programa fuente que se va a ensamblar utilizando el programa "Ensamblador" debe cumplir con ciertas reglas (gramaticales). Estas reglas constituyen al "lenguaje ensamblador". El lenguaje ensamblador no lo constituyen los mnemónicos.

#### 4.3 ACTIVIDADES DE RUTINA.

Todas nuestras actividades diarias están comprendidas por una secuencia de instrucciones que permiten realizarse paso a paso con un exitoso fin. Una receta en un libro de cocina sería un excelente ejemplo, la preparación de un platillo complicado se divide en pasos simples y comprensibles para cualquier persona con experiencia en cocina. Levantar el teléfono y marcar el número, también involucra un conjunto de pasos o instrucciones que nos permite hacer uso de este medio de comunicación. O también por ejemplo el estudiante de nuestra escuela puede necesitar que le den un conjunto de instrucciones que le indique como llegar a la biblioteca; después de esto, también tendrá que seguir ciertas instrucciones para poder localizar el libro que le interese. Es

fácil pensar en miles de situaciones análogas en las que se siguen actividades diarias, algunas de las cuales ya se han transformado en rutinas, o si no se está en una situación familiar estas actividades son esfuerzos conscientes que tenemos que llevar a cabo.

#### 4.3.1 REPRESENTACION LOGICA.

A continuación se presenta una lista de instrucciones que comprenden el proceso cotidiano de cambiar una llanta baja.

- 1.- Levantar el carro con el gato.
- 2.- Quitar los tornillos.
- 3.- Quitar la rueda.
- 4.- Poner la llanta de refacción.
- 5.- Apretar los tornillos.
- 6.- Bajar el gato.

Podrían aumentarse más instrucciones a este proceso, por ejemplo, extraer los materiales de la cajuela, colocar el gato, quitar los tapones de las ruedas y aflojar los tornillos antes de levantar el carro.

Cada una de las instrucciones puede pasar a ser parte de un esquema, que represente la actividad en su forma más lógica, que ahorre las palabras para fijar un orden de éstas en el proceso de la actividad.

De esta misma manera es explicada la DEFINICION del programa, que es parte primera de las tres partes que la componen. Esta parte es de gran importancia ya que permite al lector o usuario, así como a cualquier persona que no tenga conocimiento del mismo comprender el programa, ya que ésta parte está escrito siguiendo los mismos procedimientos o analogías que se han mencionado en este apartado.

#### 4.4 ESTRUCTURA DEL PROGRAMA.

Una vez que se han definido los conceptos básicos utilizados para una mejor comprensión del programa, se analizará la estructura que se siguió para el desarrollo del programa principal y las rutinas que la componen.

En la Fig. 4.1 se muestra la estructura del programa.

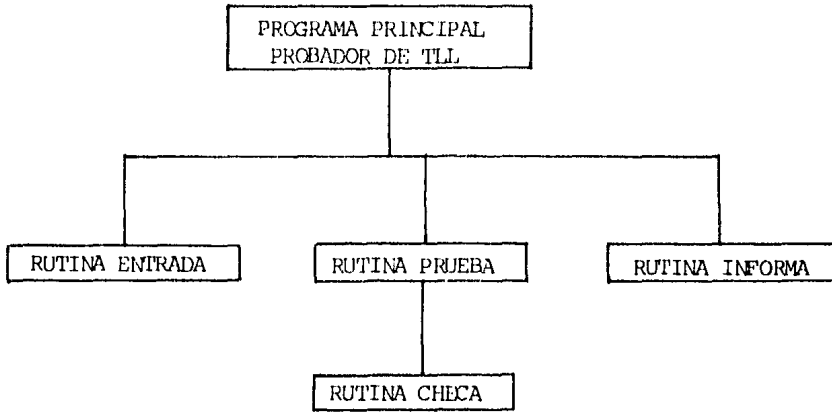


Figura 4.1. Estructura del programa "Probador de TTL".

Debido a que es muy factible equivocarse cuando se trabaja en --- lenguaje de máquina y por consiguiente encontrar los errores y corre--- girlo. El programa principal al igual que las rutinas ha sido dividido--- en tres etapas: DEFINICION, SEUDO-CODIGO Y CODIGO.

a) DEFINICION.- Aquí como su nombre lo dice se define con la ma--- yor naturalidad de lo que se pretende haga el --- programa, de tal manera que el único objetivo es que sea entendido por cualquier persona que no -- tenga conocimiento alguno de programación, obvia--- mente existen términos no usuales en la vida co--- mún. Esta etapa comprende los siguientes puntos.

- 1) Nombre de la rutina.
- 2) Quién la realizó.
- 3) Fecha.
- 4) Objetivo de la misma.
- 5) Datos de entrada (cuando la llaman).
- 6) Datos de salida (cuando termina).
- 7) Procedimiento (como hace la función).

8) Datos (requeridos por los CIs, así como los datos que le proporcionan las rutinas que --- llaman).

b) SEUDO-CODIGO.- Esta es una forma más técnica, basándose en la DEFINICION y por lo tanto condición suficiente y necesaria para el desarrollo del CODIGO, además es de gran importancia ésta etapa ya que en ella se puede detectar más fácilmente el error del programa.

c) CODIGO.- Esta es formada en un lenguaje más técnico que el anterior, Aquí se especifica con mnemónicos la operación que se debe ejecutar, estando contenidos direcciones y datos e identificándose cada parte.

A continuación se describe la función que realiza cada una de las rutinas de acuerdo a las primeras dos etapas ya mencionadas, esto es, para que el lector tenga una noción más general de como funciona el programa. Notando que en el listado dado en este mismo apartado, se mencionan de nueva cuenta, esto es debido a que todo el programa se encuentra almacenado en la microcomputadora marca Cromenco Inc., los cuales fueron confeccionados gracias a las capacidades de impresión y edición de CDOS.

#### DEFINICION :

; PROGRAMA: PROBADOR DE TIL 'PTLL'.

; PROGRAMA PRINCIPAL (PPRIN).

; ELABORO: A.F.T.

; OBJETIVO: ADMINISTRAR LA SECUENCIA ENTRADA DE INFORMACION, PRUEBA Y ENTREGA DE RESULTADO.

; DATOS DE ENTRADA: NINGUNA.

; DATOS DE SALIDA: NINGUNO.

; PROCEDIMIENTO: GENERA UN LOOP QUE HA DE TERMINAR CUANDO SE LE DE POR-COMANDO UNA 'F' SEGUIDA DE UN 'CR'. DENTRO DEL LOOP --

```

;           REALIZA LO SIGUIENTE: SOLICITA SE EJECUTE LA RUTINA -
;           ENTRADA Y VERIFICA QUE EL DATO SEA DE 'F' 'CR' Y EN -
;           TAL CASO SE DA POR TERMINADO. PERO DE LO CONTRARIO --
;           SOLICITA SEA EJECUTADA LA RUTINA PRUEBA Y ENSEGUIDA -
;           INFORMA.
; DATOS DE ENTRADA:
; RUTINA ENTRADA: NINGUNO.
; RUTINA PRUEBA: UNA CADENA CON 10 CARACTERES.
; RUTINA INFORMA: DIRECCION DEL MENSAJE A ESCRIBIR.
; DATOS DE SALIDA:
; RUTINA ENTRADA: CADENA CON 10 CARACTERES.
; RUTINA PRUEBA: DIRECCION DEL MENSAJE A ESCRIBIR.
; RUTINA INFORMA: NINGUNA.
; //////////////////////////////////////
; NOMBRE: ENTRADA 'ENTRA'.
; ELABORO: A.F.T.
; OBJETIVO: OBTENER DE LA TERMINAL UNA CADENA DE 1 a 10 CARACTERES QUE-
;           PROPORCIONA EL OPERARIO, DONDE SE GENERAN 10 CARACTERES, EN
;           TAL CASO COMPLETANDO CON BLANCOS.
; DATOS DE ENTRADA: NINGUNO.
; DATOS DE SALIDA: UNA CADENA CON 10 CARACTERES.
; PROCEDIMIENTO: AL COMPAADOR DE CARACTERES LE ASIGNA Ø, A LA CADENA LE-
;           ASIGNA 10 BLANCOS. AL APUNTAOR DE CADENA LO PONE ---
;           APUNTADO AL INICIO DE LA LOCALIDAD PARA CARACTERES.
;           GENERA UN LOOP QUE TERMINARA CUANDO EL COMEADOR DE CA-
;           RACTERES SEA IGUAL A 1Ø O BIEN EL CARACTER LEIDO SEA -
;           UN 'CR'. DENTRO DEL LOOP REALIZA LO SIGUIENTE:
;           OBTIENE UN CARACTER DE LA CONSOLA MEDIANTE EL SISTEMA-
;           OPERATIVO, GUARDANDOLO EN LA LOCALIDAD APUNTAADA POR EL
;           APUNTAOR DE CARACTERES. INCREMENTA EL APUNTAOR DE --

```

```

;          CADENA Y AL CONTADOR DE CARACTERES EN L.
;
; //////////////////////////////////////
; NOMBRE: PRUEBA 'PRUE'.
;
; ELABORO: A.F.T.
;
; OBJETIVO: PROBAR CIRCUITOS INTEGRADOS CON UNA CADENA DE 1 A 1Ø CARAC-
;          TERES DE UNA TABLA DE CIRCUITOS INIEGRADOS CONOCIDOS.
;
; DATOS DE ENTRADA: UNA CADENA CON 1Ø CARACTERES.
;
; DATOS DE SALIDA: DIRECCION DEL MENSAJE A ESCRIBIR.
;
; PROCEDIMIENTO: SE SELECCIONARA LA CADENA DE 1Ø CARACTERES DADO POR UN
;                COMANDO QUE LA MICROCOMPUTADORA DEBE CONOCER PARA PRO-
;                GRAMARLO ADECUADAMENTE. ESTE LO BUSCARA DENTRO DE LA --
;                TABLA DE CIRCUITOS CONOCIDOS, LA CUAL ESTA CONTENIDA --
;                EN UNA LOCALIDAD DE LA MISMA SI ESTA SE ENCUENTRA EN --
;                LA TABLA, ENTONCES SERA ENVIADA A UNA TABLA DE DIREC--
;                CIONES DE LAS TABLAS DE PRUEBA. SI NO LO ENCUENTRA ES-
;                TA NO REQUERIRA DE UNA DIRECCION EN LA TABLA DE DIREC-
;                CIONES. DE LO CONTRARIO PASARA A LA TABLA DE PRUEBA --
;                CON LA DIRECCION YA ESPECIFICA PARA CADA UNO DE LOS --
;                CIRCUITOS INTEGRADOS CONOCIDOS A PROBAR.
;                A CONTINUACION SE DA LA TABLA DE CIRCUITOS INTEGRADOS--
;                CONOCIDOS A PROBAR. UNA VEZ DADO ESTO SE LLAMARA A LA
;                RUTINA CHECA, LA CUAL VERIFICARA LOS RESULTADOS CON --
;                LAS TABLAS DE PRUEBA, ESTO ES, SI SON IGUALES O DIFE--
;                RENTES.
;
; //////////////////////////////////////
; NOMBRE: CHECA 'CHECA'.
;
; ELABORO: A.F.T.
;
; OBJETIVO: VERIFICAR EL RESULTADO DEL CIRCUITO A PROBAR CON LAS TABLAS
;          DE PRUEBA.
;
; DATOS DE ENTRADA: LA DIRECCION DE LAS TABLAS DE PRUEBA.

```

```

; DATOS DE SALIDA: REL RESULTADO DEL CIRCUITO INTEGRADO A PRUEBA.

; PROCEDIMIENTO: PRIMERAMENTE DEFINIR LAS ENTRADAS Y SALIDAS DEL PIO, -
;                 ESTO ES, LA PALABRA DE CONTROL PARA PROGRAMARLO Y ADE-
;                 MAS SELECCIONARLO PARA EL MODO 3 QUE IMPLICA ENTRADA Y
;                 SALIDA DE DATOS SIN PROTOCOLO.
;                 EL SIGUIENTE PASO SON LAS CONDICIONES INICIALES PARA -
;                 PODER OPERARLO. NOTANDO QUE EXISTEN CIRCUITOS QUE NO -
;                 LAS NECESITAN, ESTAS SE PODRAN VER EN LAS TABLAS CON -
;                 LOS CIRCUITOS QUE LAS UTILICEN.
;                 EL TERCER PASO SON LAS TABLAS DE PRUEBA Y EL NUMERO DE
;                 PRUEBAS, EN LA CUAL ESTAN CONTENIDAS LOS DATOS DE EN-
;                 TRADA Y SALIDA DEL CIRCUITO INTEGRADO. EN ESTE PUNTO -
;                 LAS SALIDAS (PIO) SON ENTRADAS DEL CIRCUITO INTEGRADO-
;                 Y SON LOS QUE NOS VAN A AYUDAR A PROBARLO, LAS ENTRA--
;                 DAS (PIO) SON SALIDAS DEL CIRCUITO, ESTOS SON LOS RE--
;                 SULTADOS VERDADEROS LOS CUALES SE VAN A COMPARAR CON -
;                 LOS RESULTADOS QUE NOS DE AL PROBARLO. SI EL RESULTADO
;                 ES IGUAL AL DE LA TABLA, EL RESULTADO SE TOMARA COMO -
;                 BUENO DE LO CONTRARIO SE TOMARA COMO MALO O DESCONOCI-
;                 DO.
;                 UNA VEZ CONOCIDO EL RESULTADO NOS INFORMARA.

; NOMBRE: INFORMA 'INFOR'.

; ELABORO: A.F.T.

; OBJETIVO: INFORMAR AL USUARIO EL RESULTADO DE LA PRUEBA DEL CIRCUITO-
;           INTEGRADO DESEADO.

; DATOS DE ENTRADA: DIRECCION DEL MENSAJE A ESCRIBIR.

; DATOS DE SALIDA: NINGUNA.

; PROCEDIMIENTO: UNA VEZ DADA LA DIRECCION DEL MENSAJE A ESCRIBIR, ESTE
;                 NOS INFORMARA EL RESULTADO COMO: BUENO, MALO Y DESCO--
;                 NOCIDO.

; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;
; SEUDO-CODIGO,

```

```

;      (PPRIN)
; INICIO
;      1.- FIN:=FALSO
;
;      2.- REPITE
;
;      A.- ENTRADA
;
;      B.- SI COMANDO(INICIO):='F' Y COMANDO(INICIO+1):
;          = 'CR'
;
;          1.- ENTONCES
;
;              A.- FIN:=VERDADERO
;
;          2.- DE LO CONTRARIO
;
;              A.- PRUEBA
;
;              B.- INFORMA
;
;      3.- HASTA FIN:=VERDADERO
;
; FIN

```

```

;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

```

```

;      SEUDO-CODIGO
;
;      (ENTRA)
; INICIO
;      1.- CONTADOR DE CARACTERES:=9
;
;      2.- CADENA(0...9):=BLANCO
;
;      3.- APUNTAJOR DE CADENA:=INICIO DE CADENA
;
;      4.- REPITE
;
;          A.- OBTEN CARACTER
;
;          B.- SI CARACTER:='CR'
;
;              1.- ENTONCES CONTADOR DE CARACTERES:=1
;
;          C.- CADENA(APUNTAJOR DE CADENA):=CARACTER
;
;          D.- APUNTAJOR DE CADENA:=APUNTAJOR DE CADENA+1

```



```

;      E.- CONTADOR DE CARACTERES:=CONTADOR DE CARACTERES-1
;      5.- HASTA CONTADOR DE CARACTERES=0
; FIN
;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;      SEUDO-CODIGO
;      (ENTRA)
; INICIO
;      1.- CONTADOR DE CARACTERES:=9
;      2.- CADENA(0...9):=BLANCO
;      3.- APUNTADOR DE CADENA:=INICIO DE CADENA
;      4.- REPITE
;      A.- OBTEN CARACTER
;      B.- SI CARACTER:='CR'
;          1.- ENTONCES CONTADOR DE CARACTERES:-1
;      C.- CADENA(APUNTADOR DE CADENA):=CARACTER
;      D.- APUNTADOR DE CADENA:=APUNTADOR DE CADENA+1
;      E.- CONTADOR DE CARACTERES:=CONTADOR DE CARACTERES-1
;      5.- HASTA CONTADOR DE CARACTERES=0
; FIN
;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;      SEUDO-CODIGO
;      (PRUE)
; INICIO
;      1.- APUNTADOR:=INICIO TABLA CIR. CONOCIDOS
;      2.- ENCONTRADO:=FALSO
;      3.- REPITE

```

```

;           A.- SI TABLA CIR. CONOCIDOS (APUNTADOR)=CADENA
;
;           1.- ENTONCES
;
;           A.- ENCONTRADO:=VERDADERO
;
;           2.- DE LO CONTRARIO
;
;           A.- APUNTADOR:=APUNTADOR+1
;
;           4.- HASTA ENCONTRADO = VERDADERO O APUNTADOR = FIN DE TABLA+1
;
;           5.- SI ENCONTRADO = VERDADERO
;
;           A.- ENTONCES
;
;           1.- EJECUTA CHECA
;
;           B.- DE LO CONTRARIO
;
;           1.- MENSAJE:=DESCONOCIDO
; FIN

```

```

;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

```

```

;           SEUDO-CODIGO
;
;           (CHECA)

```

```

; INICIO

```

```

;           1.- MALO:=FALSO
;
;           2.- ESCRIBE 0FFH EN EL PUERTO DE CONTROL A DEL PIO (MODO 3)
;
;           3.- ESCRIBE MEMORIA(DIRECCION DE TABLA) EN PUERTO DE CONTROL A-
;           DEL PIO
;
;           4.- DIRECCION DE TABLA:=DIRECCION DE TABLA+1
;
;           5.- ESCRIBE 0FFH EN EL PUERTO DE CONTROL B DEL PIO
;
;           6.- ESCRIBE MEMORIA(DIRECCION DE TABLA) EN PUERTO DE CONTROL B-
;           DEL PIO
;
;           7.- DIRECCION DE TABLA:=DIRECCION DE TABLA+1
;
;           8.- CONTADOR:=MEMORIA(DIRECCION DE TABLA)
;
;           9.- MIENTRAS CONTADOR 0 EJECUTA

```

```

;           A.- DIRECCION DE TABLA:=DIRECCION DE TABLA+1
;
;           B.- ESCRIBE MEMORIA(DIRECCION DE TABLA) EN PUERTO DE DATOS
;             A DEL PIO
;
;           C.- DIRECCION DE TABLA:=DIRECCION DE TABLA+1
;
;           D.- ESCRIBE MEMORIA(DIRECCION DE TABLA) EN PUERTO DE DATOS
;             B DEL PIO
;
;           E.- CONTADOR:-CONTADOR-1
;
; 10.- DIRECCION DE TABLA:=DIRECCION DE TABLA+1
;
; 11.- CONTADOR:=MEMORIA(DIRECCION DE TABLA)
;
; 12.- MIENTRAS CONTADOR Ø O MALO=VERDADERO EJECUTA
;
;           A.- DIRECCION DE TABLA:=DIRECCION DE TABLA+1
;
;           B.- ESCRIBE MEMORIA (DIRECCION DE TABLA) EN PUERTO DE DATOS
;             A DEL PIO
;
;           C.- DIRECCION DE TABLA:=DIRECCION DE TABLA+1
;
;           D.- ESCRIBE MEMORIA(DIRECCION DE TABLA) EN PUERTO DE DATOS
;             B DEL PIO
;
;           E.- LEE PUERTO A DE DATOS DEL PIO EN ENTRADA
;
;           F.- DIRECCION DE TABLA:=DIRECCION DE TABLA-1
;
;           G.- SI ENTRADA MEMORIA (DIRECCION DE TABLA)
;
;             1.- ENTONCES MALO:=VERDADERO
;
;           H.- LEE PUERTO B DE DATOS DEL PIO EN ENTRADA
;
;           I.- DIRECCION DE TABLA:=DIRECCION DE TABLA+1
;
;           J.- SI ENTRADA MEMORIA (DIRECCION DE TABLA)
;
; 13.- SI MALO:=FALSO
;
;           A.- ENTONCES: MENSAJE:=MALO
;
;           B.- DE LO CONTRARIO: MENSAJE:=BUENO
;
; FIN

```

```

;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

```

```
; SEUDO-CODIGO
; (INFOR)
; INICIO
; 1.- APUNTADOR DISPLAY:=INICIO DISPLAY
; 2.- REPITE
; A.- DISPLAY(APUNT DISPLAY):=MENSAJE(DIRECCION DEL MENSAJE)
; B.- APUNTADOR DISPLAY:=APUNTADOR DISPLAY+1
; C.- DIRECCION DEL MENSAJE=DIRECCION DEL MENSAJE+1
; 3.- HASTA DIRECCION DEL MENSAJE=FIN DEL MENSAJE+1
; FIN
; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

FORNEMCO 386 MACRO ASSEMBLER VERSION 03.07  
 \*\*\* IN \*\*\*

```

0000: 0001      ORG 0000      ,ORIGEN ROM
0002
0003
0004      UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
0005      FACULTAD DE ESTUDIOS SUPERIORES "CUAUTITLAN"
0006      INGENIERIA MECANICA Y ELECTRICA
0007
0008      TESIS "DISEÑO DE UN PROBADOR DE CIRCUITOS
0009      INTEGRADOS DIGITALES"
0010
0011      ELABORO ALFONSO FLORES TAPIA
0012
0013      EL SIGUIENTE PROGRAMA CONSTITUYE EL SOFTWARE
0014      DEL PROBADOR Y ELABORADO DE TAL FORMA QUE --
0015      PUEDA SER EMPLEADO POR EL C-80 STARTER SYSTEM
0016
0017
0018
0019      PROGRAMA PRINCIPAL
0020
0021      NOMBRE PROBADOR DE TTL P TTL
0022      ELABORO A F T
0023
0024      OBJETIVO ADMINISTRAR LA SECUENCIA ENTRADA DE
0025      INFORMACION PRUEBA Y ENTREGA DE RESULTADO.
0026
0027      DATOS DE ENTRADA NINGUNO
0028      DATOS DE SALIDA NINGUNO
0029
0030      PROCEDIMIENTO GENERA UN LOOP QUE HA DE TERMINAR
0031      CUANDO SE LE DE POR COMANDO UNA "F" SEGUIDA DE -
0032      UN "CP"
0033      SOLICITA SE EJECUTE LA Rutina ENTRADA Y VERIFICA
0034      QUE EL DATO SEA DE "F" "CP" Y EN TAL CASO SE DA-
0035      POR TERMINADO Y PERO DE LO CONTRARIO SOLICITA SEA
0036      EJECUTADA LA Rutina PRUEBA Y ENSEGUIDA INFORMA
0037
0038      DATOS DE ENTRADA
0039      Rutina ENTRADA NINGUNO
0040      Rutina PRUEBA UNA CADENA CON 10 CARACTERES
0041      Rutina INFORMAR DIRECCION DEL MENSAJE A ESCRIBIR
0042
0043      DATOS DE SALIDA
0044      Rutina ENTRADA UNA CADENA CON 10 CARACTERES
0045      Rutina PRUEBA DIRECCION DEL MENSAJE A ESCRIBIR

```

```

0046 ,      RITUNA INFORMA. NINGUNO
0047
0048
0049 ,      SEUDO-CODIGO
0050 ,      (FTTL)
0051
0052 , INICIO
0053 ,      0 -LIMPIA DISPLAY
0054 ,      1 -FIN -FALSO
0055 ,      2 -REPITE

```

CROMENCO Z80 MACRO ASSEMBLER VERSION 02.07

+++ IN +++

```

0056 ,      A -ENTRADA
0057 ,      B -SI COMANDO[INICIO]-'F' Y COMANDO[INICIO+1]-'CR'
0058 ,      1 -ENTONCES
0059 ,      A -FIN -VERDADERO
0060 ,      2 -DE LO CONTRARIO
0061 ,      A -PRUEBA
0062 ,      B -INFORMA
0063 ,      3 -HASTA FIN -VERDADERO
0064 , FIN
0065 , START
0066 , FTTL , INICIO
0800 0810 0067 LD B,BI ,0.-
0802 2100FE 0068 LD HL,0FE00H
0800 0E0F 0069 LD A,0FFH
0807 77 0070 LOOP LD HL,A
0808 2D 0071 INC HL
0809 10F0 0072 DJNZ LOOP
0073 ,1.-
0074 ,2.-
0806 005008 0075 FTTL1 CALL ENTRA ,2.A.-
080E 2E46 0076 LD A,'F' ,2.B.-
0810 BE 0077 CP A,HL)
0811 021008 R 0078 JP NZ,FTTL2
0814 2D 0079 INC HL
0816 2E20 0080 LD A,' '
0817 BE 0081 CP A,HL)
0818 2B 0082 DEC HL
0083 ,2.B.1.-
0819 0A0000 0084 JP 0,0 ,2.B.1.A.-
0085 ,2.B.2.-
0811 00E10E 0086 CALL PAUE ,2.B.3.A.-
081F 00710 0087 CALL INFOR ,2.B.2.B.-
0822 005008 R 0088 JP FTTL1 ,3.-
0089 ,
0090 , RUTINA ENTRADA
0091
0092 , NONESE ENTRADA ENTRA'
0093 , ELABORO. A.F.T

```

```

0094
0095 , OBJETIVO. OBTENER DE LA TERMINAL UNA CADENA
0096 , DE 1 A 10 CARACTERES QUE PROPORCIONA EL OPE-
0097 , RARIO, DONDE 5 GENERAN 10 CARACTERES, EN TAL
0098 , CASO COMPLETANDO CON BLANCOS.
0099
0100 , DATOS DE ENTRADA. NINGUNO
0101 , DATOS DE SALIDA. UNA CADENA CON 10 CARACTERES
0102
0103 , PROCEDIMIENTO. AL CONTADOR DE CARACTERES LE -
0104 , ASIGNA 0. A LA CADENA LE ASIGNA 10 BLANCOS. -
0105 , AL APUNTAOR DE CADENA LO PONE APUNTADO AL --
0106 , INICIO DE LA LOCALIDAD PARA CARACTERES.
0107 , GENERA UN LOOP QUE TERMINARA CUANDO EL CONTA-
0108 , DOR DE CARACTERES SEA IGUAL A 10 O BIEN EL CA-
0109 , RACTER LEIDO SEA UN 'CR'.
0110 , DENTRO DEL LOOP REALIZA LO SIGUIENTE. OBTIENE

```

CRONEMICO 280 MACRO ASSEMBLER VERSION 02 07

\*\*\* IN \*\*\*

```

0111 , UN CARACTER DE LA CONSOLA MEDIANTE EL SISTEMA
0112 , OPERATIVO. GUARDANDOLO EN LA LOCALIDAD APUNTA-
0113 , DA POR EL APUNTAOR DE CADENA. INCREMENTA EL-
0114 , APUNTAOR DE CADENA Y AL CONTADOR DE CARACTE-
0115 , RES EN 1
0116
0117
0118 , SEUDO-CODIGO
0119 , ENTRAR
0120
0121 , INICIO
0122
0123 , 0 -APUNTAOR DISPLAY OFEGGH
0124
0125 , 1 -CONTADOR DE CARACTERES 0
0126 , 2 -CADENA(0..9) BLANCO
0127 , 3 -APUNTAOR DE CADENA -INICIO DE CADENA
0128 , 4 -REPITE
0129 , A -OBTEN CARACTER
0130 , B -SI CARACTER 'CR'
0131 , 1 -MENTONCES CONTADOR DE CARACTERES +=1
0132 , C -CADENA(APUNTAOR DE CADENA) CARACTER
0133 , D -APUNTAOR DE CADENA APUNTAOR DE CADENA+1
0134 , E -CONTADOR DE CARACTERES CONTADOR DE CARACTERES+1
0135 , 0 -HASTA CONTADOR DE CARACTERES 0
0136 , FIN
0137 ENTRAR LD HL OFFFH 10 -
0138 LD A,OFFFH HL
0139 LD B,B 1 -
0140 LD A,A 10 -
0141 LD A,CADENA

```

```

0800 0100FE
0801 010000
0802 0609
0803 0800
0804 010000
0805 010000

```

```

0822 77      0142 ENTRAC LD (HL),A
0823 25      0143      INC HL
0824 10FC    0144      DJNZ ENTRAC
0826 0609    0145      LD B,9
0828 210220  0146      LD HL,CADENA      .D.-
                                .4.-
082B ED      0148      PUSH HL
082C C0EFO9  0149      CALL CARAC      .4.A.-
082F E1      0150      POP HL
0840 FE00    0151      CP A,00H      .4.B.-
0842 C01908  R 0152      JP NZ,ENTAC
0843 0601    0153      LD B,1
0847 3E20    0154      LD A,0
0849 77      0155      ENTAC LD (HL),A      .4.C.-
084A 25      0156      INC HL      .4.D.-
084B 10EE    0157      DJNZ ENTRAM    .4.E.-.D.-
                                .FIN
                                MODIFICACION CODIGO DE SERIE 74
084D 210220  0160      LD HL,CADENA
0850 7E      0161      LD A,(HL)
0851 FE27    0162      CP A,77
0852 00      0163      RET NZ
0854 25      0164      INC HL
0855 7E      0165      LD A,(HL)
DRAGHECO Z80 MACRO ASSEMBLER VERSION 03 87
*** IN ***

0856 FE24    0166      CP A,74
0858 C0E008  R 0167      JP C,TIFOH
085B 2B      0168      DEC HL
085C 09      0169      RET
085D 25      0170      TIFOH INC HL
085E 7E      0171      LD A,(HL)
085F FE18    0172      CP A,18
0861 C0E808  R 0173      JP NZ,TIFOL
0864 3E21    0174      LD A,1
0866 77      0175      LD (HL),A
0867 09      0176      RET
0868 F810    0177      TIFOL CP A,10
086A C0E708  R 0178      JP NZ,TIFOS
086D 25      0179      INC HL
086E 7E      0180      LD A,(HL)
086F FE00    0181      CP A,0
0871 C0E908  R 0182      JP C,TIFLS
0874 3E20    0183      LD A,0
0876 2B      0184      DEC HL
0877 77      0185      LD (HL),A
0878 09      0186      RET
                                TIFLS
0879 3E20    0187      LD A,0
087B 77      0188      LD (HL),A

```



```

0870 C9      0190      RET
0870 FE00      0191      TIP03 CP A, 'S'
087F 028602  R 0192      JP NZ, NORMAL
0882 FE04      0193      LD A, '4'
0884 77      0194      LD (HL), A
0885 C9      0195      RET
0886 38      0196      NORMAL DEC HL
0887 FE10      0197      LD A, '0'
0889 77      0198      LD (HL), A
088A C9      0199      RET
088B 38      0200      ASCII DB 'X' , NO IMPRINIBLE
088C 48      0201      DB 'H' , H
088D 4C      0202      DB 'L' , L
088E 50      0203      DB 'S' , S
088F 54      0204      DB 'O' , O
0890 58      0205      DB 'N' , NO IMPRINIBLE
0891 5C      0206      DB 'F' , F
0892 60      0207      DB 'D' , D
0893 64      0208      DB 'C' , C
0894 68      0209      DB '1' , 1
0895 6C      0210      DB 'X' , NO IMPRINIBLE
0896 70      0211      DB 'N' , NO IMPRINIBLE
0897 74      0212      DB '6' , 6
0898 78      0213      DB 'O' , O
0899 7C      0214      DB '4' , 4
089A 80      0215      DB 'X' , NO IMPRINIBLE
089B 84      0216      DB 0DH , CR
089C 88      0217      DB '9' , 9
089D 8C      0218      DB '8' , 8
089E 90      0219      DB '7' , 7
089F 94      0220      DB 'X' , TECLA NO EXISTENTE

```

CROMENCO 360 MACRO ASSEMBLER VERSION 03 07  
 \*\*\* IN \*\*\*

```

08A0 58      0221      DB 'X' , NO IMPRINIBLE
08A1 5C      0222      DB 'X' , NO IMPRINIBLE
08A2 60      0223      DB 'X' , NO IMPRINIBLE
08A3 64      0224      DB 'X' , NO IMPRINIBLE
08A4 68      0225      DB 'X' , TECLA NO EXISTENTE
08A5 6C      0226      DB 'X' , NO IMPRINIBLE
08A6 70      0227      DB 'X' , NO IMPRINIBLE
08A7 74      0228      DB 'X' , NO IMPRINIBLE
08A8 78      0229      DB 'X' , NO IMPRINIBLE

```

```

08A9 F1FE8EFA 0230      CODIGO DE OPCH, OFEH, OARH, OFAH, OFCH , NI
08AA 00F7F7F7 0231      DB 00CH, 0FTH, 0FTH, 0FTH, 00CH , H
08AB 00F0F0F0 0232      DB 00CH, 0FPH, 0FPH, 0FTH, 00FA , L
08AC 090E0E0E 0233      DB 00RH, 00RH, 00RH, 00RH, 00EH , S
08AD 010E0E0E 0234      DB 00LH, 00FH, 00FH, 00RH, 00FH , O
08AE F1FE8EFA 0235      DB 0FCH, 0FEH, 0ARH, 0FAH, 0FCH , NI

```

0807	00F6F6F6	0238	DB	000H, 0F6H, 0F6H, 0F6H, 0FEH	.F
0808	00E8E8E8	0239	DB	000H, 0E8H, 0E8H, 0E8H, 0C9H	.2
0809	00C6C6C6	0240	DB	000H, 0C6H, 0C6H, 0C6H, 0C9H	.3
080A	FF00000F	0241	DB	0FFH, 0C0H, 000H, 0C0H, 0FFH	.1
080B	FDFEAF6A	0242	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
080C	FDFEAF6A	0243	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
080D	C15E0E0E	0244	DB	0C1H, 0E0H, 0E0H, 0E0H, 0CEH	.6
080E	D87ADADA	0245	DB	0C8H, 0DAH, 0DAH, 0DAH, 0C6H	.5
080F	E7E8E000	0246	DB	0E7H, 0E8H, 0E0H, 000H, 0EFH	.4
0810	FDFEAF6A	0247	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
0811	FFFFFFF	0248	DB	0FFH, 0FFH, 0FFH, 0FFH, 0FFH	.CR
0812	D9E6E6E6	0249	DB	0C9H, 0E6H, 0E6H, 0C6H, 0C1H	.9
0813	C9E6E6E6	0250	DB	0C9H, 0E6H, 0E6H, 0C6H, 0C9H	.8
0814	FE0EFAFA	0251	DB	0FEH, 0E0H, 0F6H, 0FAH, 0FCH	.7
0815	FDFEAF6A	0252	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NE
0816	FDFEAF6A	0253	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
0817	FDFEAF6A	0254	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
0818	FDFEAF6A	0255	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
0819	FDFEAF6A	0256	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
081A	FDFEAF6A	0257	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NE
081B	FDFEAF6A	0258	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
081C	FDFEAF6A	0259	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
081D	FDFEAF6A	0260	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
081E	FDFEAF6A	0261	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
081F	FDFEAF6A	0262	DB	0FDH, 0FEH, 0AEH, 0FAH, 0FDH	.NI
		0263			
081F		0263	DEL	000 0A4H	RETARDO EN MONITOR
		0264			
		0265			
		0266			
		0267			
0827	0A0000	0268	CARAC	LD HL, (DIRDISP)	
0828	2E0F	0269		LD A, 0FH	
0829	80	0270		AND A, 1	
082A	0F	0271		LD L, A	
082B	2E0F	0272		LD A, 0FEH	
082C	87	0273		LD H, A	
082D	2E0000	0274		LD (DIRDISP), HL	
082E	00	0275		PUSH BC	
CROMBENCIO 080 MACRO ASSEMBLER VERSION 02 07					
*** IN ***					
082F	000000	0276		CALL TELLR	
0830	01	0277		POP BC	
0831	07	0278		PUSH HL	
0832	05	0279		PUSH BC	
0833	0D	0280		PUSH AF	
0834	0A00	0281		LD A, 0	
0835	00	0282		CP A, 0	
0836	00, 00	0283		JP NC, 00H	
0837	0000	0284		LD B, 0H	
0838	0A0000	0285		LD HL, 0A000H	

095F	3EFF	0286		LD A,6FFH	
0961	77	0287	OTRAS.	LD (HL),A	
0962	21	0288		INC HL	
0963	16FC	0289		DJNZ,OTRAS	
0965	F1	0290	SAL.	POP AF	
0966	C1	0291		POP BC	
0967	E1	0292		POP HL	
		0293			
0968	006A09	0294		CALL IMPRIME	
096B	C9	0295		RET	
096C	004F06	0296	TECLA.	CALL DEL	, ESPERA 20 MSE
096F	DE7F	0297		LD A,7FH	, RUTINA TECLA
0971	0288	0298		OUT 3BH,A	, APAGA DISP DE
0973	DE3F	0299		LD A,3FH	, BUSCA TECLA E
0975	DE8C	0300		OUT 3CH,A	
0977	0B90	0301		IN A,30H	, OBTEN DATO DE
0979	E61F	0302		AND A,1FH	
097B	FE1F	0303		CP A,1FH	, SI HAY TECLA
097D	28ED	0304		JR Z,TECLA	
097F	004F06	0305		CALL DEL	, ESPERA 20 MSE
0982	0E8C	0306		LD C,3CH	, INDICA PUERTO
0984	0E01	0307		LD B,01	, INICIALIZA RE
0986	ED41	0308	TECLA1.	OUT (C),B	, INDICA RENGLO
0988	0B90	0309		IN A,90H	, OBTEN DATO DE
098A	E61F	0310		AND A,1FH	, ENMASCARA ENT
098C	FE1F	0311		CP A,1FH	, LA TECLA ESTA
098E	029C09	R 0312		JF NC,SALIDA	, SI ESTA SAL D
0991	CB20	0313		SLA B	, SI NO ACTUALI
0993	DE28	0314		LD A,40	, YA SE BUSCO E
0995	88	0315		CP A,B	
0996	006A09	R 0316		JF NC,TECLA1	, SI NO BUSCA E
0999	028C09	R 0317		JF TECLA	, SI YA BUSCA N
099C	4F	0318	SALIDA.	LD C,A	, SALVA DATO DE
099D	0B90	0319	SALIDA1.	IN A,90H	, OBTEN DATO DE
099F	E61F	0320		AND A,1FH	, ENMASCARA COL
09A1	FE1F	0321		CP A,1FH	, NO ESTA LA TE
09A3	029C09	R 0322		JF NC,SALIDA1	, NO ESPERA F
09A6	110000	0323		LD DE,0	, DECODIFICA NU
09A9	210000	0324		LD HL,0000	
09AC	CB18	0325	DECO1	RR B	
09AE	0AB409	R 0326		JF C,DECO2	
09B1	13	0327		ADD HL,DE	
09B3	18F8	0328		JR DECO1	
09B4	CB19	0329	DECO2	RR C	
09B6	D0	0330		RET NC	

CROMENCO 280 MACRO ASSEMBLER VERSION 03.07  
 \*\*\* IN \*\*\*

09B7	20	0331		INC HL	
09B8	18FA	0332		JR DECO2	
		0333			

096A	E5	0034	IMPRIME. PUSH HL
096B	E5	0035	PUSH HL
096C	D1	0036	POP DE
096D	29	0037	ADD HL, HL
096E	29	0038	ADD HL, HL
096F	19	0039	ADD HL, DE
09C0	11A908	0040	LD DE, CODIGO
09C1	19	0041	ADD HL, DE
09C4	ED080000	0042	LD DE, (DIRDISP)
09C5	7E	0043	LD A, E
09C9	E60F	0044	AND 0FH
09CB	F80F	0045	CP A, 0FH
09CD	00109	R 0046	JP NZ, OTRA
09D0	1C	0047	INC DE
09D1	010000	0048	OTRA LD BC, 00
09D4	ED60	0049	LDIR
09D6	ED010010	0050	LD (DIRDISP), DE
09DA	E1	0051	POP HL
09DB	11808	0052	LD DE, ASCII
09DE	19	0053	ADD HL, DE
09DF	7E	0054	LD A, (HL)
09E0	09	0055	RET
		0056	
		0057	
		0058	, NONERE PRUEBA "PRUE"
		0059	
		0060	, ELABORO. A F T
		0061	
		0062	, OBJETIVO. PROBAR CIRCUITOS INTEGRADOS CON UNA CADENA DE
		0063	
		0064	1 A 10 CARACTERES DE UNA TABLA DE CIRCUITOS -
		0065	INTEGRADOS CONOCIDOS
		0066	
		0067	, DATOS DE ENTRADA. UNA CADENA CON 10 CARACTERES
		0068	, DATOS DE SALIDA. DIRECCION DEL MENSAJE A ESCRIBIR.
		0069	
		0070	, PROCEDIMIENTO SE SELECCIONARA LA CADENA DE 10 CARACTE-
		0071	RES CADA POR UN COMANDO QUE LA MICROCOM-
		0072	PUTADORA DEBE CONOCER PARA PROGRAMARLO -
		0073	ACORDADAMENTE ESTE LO BUSCARA DENTRO DE-
		0074	LA TABLA DE CIRCUITOS CONOCIDOS. LA CUAL
		0075	ESTA CONTENIDA EN UNA LOCALIDAD DE LA -
		0076	MISMA SI ESTA SE ENCUENTRA EN LA TABLA. -
		0077	ENTONCES SERA ENVIADA A UNA TABLA DE DI-
		0078	RECCIONES DE LAS TABLAS DE PRUEBA SI NO-
		0079	LO ENCUENTRA ESTA NO REQUERIRA DE UNA DI-
		0080	RECCION EN LA TABLA DE DIRECCIONES DE LO
		0081	CONTRARIO PASARA A LA TABLA DE PRUEBA -
		0082	CON LA DIRECCION YA ESPECIFICA PARA CADA
		0083	UNA DE LOS CIRCUITOS INTEGRADOS CONOCI-
		0084	DOS A PROBAR

0385 ,  
 CROMEMCO Z80 MACRO ASSEMBLER VERSION 03.07  
 \*\*\* IN \*\*\*

A CONTINUACION SE DA LA TABLA DE CIRCUI--

0386 , TOS INTEGRADOS CONOCIDOS A PROBAR.  
 0387 , UNA VEZ CADA ESTO SE LLAMARA A LA RUTINA-  
 0388 , CHECA, LA CUAL VERIFICARA LOS RESULTADOS-  
 0389 , CON LAS TABLAS DE PRUEBA, ESTO ES, SI SON-  
 0390 , IGUALES O DIFERENTES.  
 0391 , NOTA. LAS 5 SERIES QUE DISPONE LA FAMILIA TTL, EN LA TABLA  
 0392 , SIGUIENTE TOMAN LOS VALORES.

0393 ,	0394 ,	SERIE	VALOR
0395 ,	ESTANDAR		0
0396 ,	H		1
0397 ,	L		2
0398 ,	LS		3
0399 ,	S		4

0400 TCC.	0401	DB	1000
09E1 00000000	0401	DB	1000
09E0 01000000	0402	DB	1100
09E9 02000000	0403	DB	1200
09E8 03000000	0404	DB	1300
09F1 04000000	0405	DB	1400
09F0 05000100	0406	DB	1501
09F9 06000100	0407	DB	1601
09F8 07000100	0408	DB	1701
0A01 08000200	0409	DB	1802
0A00 09000200	0410	DB	1902
0A09 0A000200	0411	DB	2002
0A08 0B000200	0412	DB	2102
0A11 0C000300	0413	DB	2203
0A10 0D000300	0414	DB	2303
0A19 0E000300	0415	DB	2403
0A18 0F000300	0416	DB	2503
0A21 10000400	0417	DB	2604
0A20 11000400	0418	DB	2704
0A29 12000400	0419	DB	2804
0A28 13000400	0420	DB	2904
0A31 14000500	0421	DB	3004
0A30 15000500	0422	DB	3105
0A39 16000500	0423	DB	3205
0A38 17000500	0424	DB	3305
0A41 18000600	0425	DB	3406
0A40 19000600	0426	DB	3506
0A49 1A000700	0427	DB	3607
0A48 1B000700	0428	DB	3707
0A51 1C000800	0429	DB	3808
0A50 1D000800	0430	DB	3908
0A59 1E000900	0431	DB	4009
0A58 1F000900	0432	DB	4109

0A61	34D03920	0433	DB	'409
0A65	30B13020	0434	DB	'010
0A69	31D13020	0435	DB	'110
0A6D	33D13020	0436	DB	'210
0A71	35D13020	0437	DB	'310
0A75	34B13020	0438	DB	'410
0A79	31D13120	0439	DB	'111
0A7D	33D13120	0440	DB	'311

CROMENCO Z80 MACRO ASSEMBLER VERSION 03 07  
 \*\*\* IN \*\*\*

0A81	34D13120	0441	DB	'411
0A85	30B13220	0442	DB	'012
0A89	30B13320	0443	DB	'212
0A8D	30B13420	0444	DB	'013
0A91	30B13520	0445	DB	'313
0A95	30B13620	0446	DB	'014
0A99	30B13720	0447	DB	'214
0A9D	34B13820	0448	DB	'115
0AA1	32B13920	0449	DB	'315
0AA5	34B13A20	0450	DB	'415
0AA9	30B13B20	0451	DB	'016
0AAD	30B13C20	0452	DB	'017
0AB1	30B13D20	0453	DB	'220
0AB5	31B20020	0454	DB	'120
0AB9	30B13E20	0455	DB	'220
0ABB	30B13F20	0456	DB	'020
0ABF	34B20020	0457	DB	'420
0AC3	31B20120	0458	DB	'121
0AC7	30B20220	0459	DB	'221
0ACB	30B20320	0460	DB	'022
0AD3	31B20420	0461	DB	'122
0AD7	30B20520	0462	DB	'222
0ADA	31B20620	0463	DB	'422
0ADD	30B20720	0464	DB	'023
0AE1	30B20820	0465	DB	'223
0AE5	30B20920	0466	DB	'024
0AE9	30B20A20	0467	DB	'227
0AED	30B20B20	0468	DB	'027
0AF1	30B20C20	0469	DB	'228
0AF5	30B20D20	0470	DB	'028
0AF9	30B20E20	0471	DB	'229
0AFD	31B20F20	0472	DB	'129
0B01	30B21020	0473	DB	'229
0B05	30B21120	0474	DB	'029
0B09	31B21220	0475	DB	'429
0B0D	30B21320	0476	DB	'029
0B11	30B21420	0477	DB	'229
0B15	30B21520	0478	DB	'029
0B19	30B21620	0479	DB	'229
0B1D	30B21720	0480	DB	'029

0621	00333720	0481	DB '037 '
0625	00333720	0482	DB '037 '
0629	04333720	0483	DB '437 '
0633	00333820	0484	DB '038 '
0637	00333820	0485	DB '038 '
0641	04333820	0486	DB '438 '
0645	00343020	0487	DB '040 '
0649	01343020	0488	DB '140 '
0653	00343020	0489	DB '040 '
0657	04343020	0490	DB '440 '
0661	00343220	0491	DB '042 '
0665	02343220	0492	DB '242 '
0669	00343220	0493	DB '042 '
0673	00343220	0494	DB '043 '
0677	00343220	0495	DB '043 '

CROMEMCO Z60 MACRO ASSEMBLER VERSION 02.07  
 \*\*\* IN \*\*\*

0680	00343420	0496	DB '044 '
0684	00343420	0497	DB '044 '
0688	00343620	0498	DB '045 '
0692	00343620	0499	DB '046 '
0696	00343620	0500	DB '046 '
0700	00343720	0501	DB '047 '
0704	00343720	0502	DB '047 '
0708	00343720	0503	DB '047 '
0712	00343820	0504	DB '048 '
0716	00343820	0505	DB '048 '
0720	00343920	0506	DB '049 '
0724	00343920	0507	DB '049 '
0728	00350120	0508	DB '051 '
0732	01350120	0509	DB '151 '
0736	04350120	0510	DB '451 '
0740	00350120	0511	DB '051 '
0744	01350120	0512	DB '151 '
0748	00350120	0513	DB '054 '
0752	01350120	0514	DB '154 '
0756	00350120	0515	DB '054 '
0760	00350120	0516	DB '054 '
0764	04350120	0517	DB '454 '
0768	04350320	0518	DB '455 '
0772	00350320	0519	DB '058 '
0776	01350320	0520	DB '158 '
0780	00350320	0521	DB '058 '
0784	00350320	0522	DB '072 '
0788	01350320	0523	DB '172 '
0792	00350320	0524	DB '072 '
0796	00350320	0525	DB '072 '
0800	01350320	0526	DB '172 '
0804	00350320	0527	DB '075 '
0808	00350320	0528	DB '075 '

06E1	00370400	0029	06	'074
06E5	01270400	0030	06	'174
06E9	02270400	0031	06	'274
06ED	03270400	0032	06	'374
06F1	04270400	0033	06	'474
06F5	05270000	0034	06	'075
06F9	06270000	0035	06	'275
06FD	07270000	0036	06	'375
0C01	08270600	0037	06	'076
0C05	01270600	0038	06	'176
0C09	03270600	0039	06	'376
0C0D	04270600	0040	06	'476
0C11	05270600	0041	06	'576
0C15	06270600	0042	06	'676
0C19	08280000	0043	06	'080
0C1D	09280000	0044	06	'180
0C21	0A280000	0045	06	'280
0C25	0B280000	0046	06	'380
0C29	0C280600	0047	06	'086
0C2D	0D280600	0048	06	'186
0C31	0E280600	0049	06	'286
0C35	0F280600	0050	06	'386

ROMENCO P&O MACRO ASSEMBLER VERSION 03.07  
 \*\*\* IN \*\*\*

0C39	01290000	0051	06	'187
0C3D	02290000	0052	06	'290
0C41	03290000	0053	06	'390
0C45	04290000	0054	06	'490
0C49	05290000	0055	06	'590
0C4D	06290000	0056	06	'690
0C51	08290000	0057	06	'093
0C55	09290000	0058	06	'193
0C59	0A290000	0059	06	'293
0C5D	0B290400	0060	06	'394
0C61	0C290400	0061	06	'494
0C65	0D290000	0062	06	'594
0C69	0E290000	0063	06	'694
0C6D	0F290600	0064	06	'096
0C71	01290600	0065	06	'196
0C75	02290600	0066	06	'296
0C79	03290000	0067	06	'310
0C7D	04290000	0068	06	'410
0C81	05290000	0069	06	'510
0C85	06290000	0070	06	'610
0C89	08290000	0071	06	'710
0C8D	09290000	0072	06	'810
0C91	0A290000	0073	06	'910
0C95	0B290000	0074	06	'010
0C99	0C290400	0075	06	'114
0C9D	0D290400	0076	06	'214



0CA1	00B13426	0077	DB	'0148'
0CA2	00B13428	0078	DB	'0148'
0CA9	00B1302D	0079	DB	'0103'
0CAD	00B13033	0080	DB	'0103'
0CB1	00B13033	0081	DB	'0103'
0CB5	0421302D	0082	DB	'4103'
0CB9	00B13030	0083	DB	'0103'
0CC0	00B13030	0084	DB	'0103'
0CC1	00B13036	0085	DB	'0106'
0CC5	00B13036	0086	DB	'0106'
0CC9	00B13037	0087	DB	'0107'
0CC0	00B13037	0088	DB	'0107'
0CD1	00B13037	0089	DB	'0107'
0CD5	04B13037	0090	DB	'4107'
0CD9	00B13038	0091	DB	'0108'
0CDB	04B13038	0092	DB	'4108'

0093 FINTAB  
0094  
0095 DIRECCIONES DE LAS TABLAS DE PRUEBA  
0096  
0097  
0098  
0099 TABLAS

0CE1	AC0E	0600	DW	TAB1
0CE3	AC0E	0601	DW	TAB1
0CE5	AC0E	0602	DW	TAB1
0CE7	AC0E	0603	DW	TAB1
0CE9	AC0E	0604	DW	TAB1
0CEE	E90E	0605	DW	TAB2

CRONENCO 230 MACRO ASSEMBLER VERSION 03.07  
\*\*\* IN \*\*\*

0CE0	E90E	0606	DW	TAB2
0CE7	AC0E	0607	DW	TAB1
0CF1	C00E	0608	DW	TAB3
0CF5	C00E	0609	DW	TAB3
0CF0	C00E	0610	DW	TAB3
0CF7	C00E	0611	DW	TAB3
0CF9	AC0E	0612	DW	TAB1
0CF6	AC0E	0613	DW	TAB1
0CFD	AC0E	0614	DW	TAB1
0CFF	AC0E	0615	DW	TAB1
0D01	D10E	0616	DW	TAB4
0D0C	D10E	0617	DW	TAB4
0D05	D10E	0618	DW	TAB4
0D07	D10E	0619	DW	TAB4
0D09	D10E	0620	DW	TAB4
0D0E	D10E	0621	DW	TAB4
0D0D	D10E	0622	DW	TAB4
0D0F	D10E	0623	DW	TAB4
0D11	D10E	0624	DW	TAB4

0013	D10E	0620	DW	TAB4
0015	E90E	0626	DW	TAB5
0017	E1GE	0627	DW	TAB6
0019	E10E	0628	DW	TAB6
001B	E10E	0629	DW	TAB6
001D	E10E	0630	DW	TAB6
001F	E10E	0631	DW	TAB6
0021	E10E	0632	DW	TAB6
0023	ED0E	0633	DW	TAB7
0025	ED0E	0634	DW	TAB7
0027	ED0E	0635	DW	TAB7
0029	ED0E	0636	DW	TAB7
002B	ED0E	0637	DW	TAB7
002D	010F	0638	DW	TAB8
002F	010F	0639	DW	TAB8
0031	010F	0640	DW	TAB8
0033	ED0E	0641	DW	TAB7
0035	ED0E	0642	DW	TAB7
0037	150F	0643	DW	TAB9
0039	150F	0644	DW	TAB9
003B	D10E	0645	DW	TAB4
003D	D10E	0646	DW	TAB4
003F	010F	0647	DW	TAB8
0041	010F	0648	DW	TAB8
0043	010F	0649	DW	TAB8
0045	D10E	0650	DW	TAB4
0047	050E	0651	DW	TAB5
0049	150F	0652	DW	TAB9
004B	150F	0653	DW	TAB9
004D	150F	0654	DW	TAB9
004F	150F	0655	DW	TAB9
0051	150F	0656	DW	TAB9
0053	190F	0657	DW	TAB10
0055	790F	0658	DW	TAB10
0057	150F	0659	DW	TAB9
0059	150F	0660	DW	TAB9

CRONHEND DSD MACRO ASSEMBLER VERSION 02 07

\*\*\* IN \*\*\*

005B	150F	0661	DW	TAB9
005D	150F	0662	DW	TAB9
005F	500F	0663	DW	TAB11
0061	AD0E	0664	DW	TAB1
0063	AD0E	0665	DW	TAB1
0065	810F	0666	DW	TAB12
0067	810F	0667	DW	TAB12
0069	050E	0668	DW	TAB5
006B	050E	0669	DW	TAB5
006D	500F	0670	DW	TAB12
006F	500F	0671	DW	TAB12
0071	500F	0672	DW	TAB12

0073	350F	0673	DW	TAB13
0075	350F	0674	DW	TAB13
0077	630F	0675	DW	TAB14
0079	830F	0676	DW	TAB14
007B	630F	0677	DW	TAB14
007D	C50E	0678	DW	TAB3
007F	C50E	0679	DW	TAB3
0081	AD0E	0680	DW	TAB1
0083	AD0E	0681	DW	TAB1
0085	AD0E	0682	DW	TAB1
0087	AD0E	0683	DW	TAB1
0089	AD0E	0684	DW	TAB1
008B	AD0E	0685	DW	TAB1
008D	150F	0686	DW	TAB9
008F	150F	0687	DW	TAB9
0091	150F	0688	DW	TAB9
0093	150F	0689	DW	TAB9
0095	C50F	0690	DW	TAB15
0097	C50F	0691	DW	TAB15
0099	C50F	0692	DW	TAB15
009B	D50F	0693	DW	TAB16
009D	D50F	0694	DW	TAB16
009F	F50F	0695	DW	TAB17
00A1	F50F	0696	DW	TAB17
00A3	0010	0697	DW	TAB18
00A5	2510	0698	DW	TAB19
00A7	2510	0699	DW	TAB19
00A9	2510	0700	DW	TAB19
00AB	2510	0701	DW	TAB19
00AD	2510	0702	DW	TAB19
00AF	2010	0703	DW	TAB20
00B1	2010	0704	DW	TAB20
00B3	2510	0705	DW	TAB21
00B5	2510	0706	DW	TAB21
00B7	6010	0707	DW	TAB22
00B9	6010	0708	DW	TAB22
00BB	6010	0709	DW	TAB22
00BD	3110	0710	DW	TAB23
00BF	3110	0711	DW	TAB23
00C1	8510	0712	DW	TAB24
00C3	0510	0713	DW	TAB25
00C5	8F10	0714	DW	TAB26
00C7	EF10	0715	DW	TAB26

CRONENCO 280 MACRO ASSEMBLER VERSION 03 07

\*\*\* IN \*\*\*

00C9	0511	0716	DW	TAB27
00CB	1011	0717	DW	TAB28
00CD	1511	0718	DW	TAB29
00CF	1011	0719	DW	TAB30
00D1	4111	0720	DW	TAB31

0003	7711	0721	DW	TAB32
0005	7711	0722	DW	TAB32
0007	7711	0723	DW	TAB32
0009	8F11	0724	DW	TAB33
0008	8F11	0725	DW	TAB33
0000	8F11	0726	DW	TAB33
000F	8D11	0727	DW	TAB34
00E1	8D11	0728	DW	TAB35
00E3	8D11	0729	DW	TAB35
00E5	8D11	0730	DW	TAB35
00E7	8D11	0731	DW	TAB35
00E9	8D11	0732	DW	TAB35
00EB	CF11	0733	DW	TAB36
00ED	CF11	0734	DW	TAB36
00EF	CF11	0735	DW	TAB36
00F1	0911	0736	DW	TAB37
00F3	0911	0737	DW	TAB37
00F5	F111	0738	DW	TAB38
00F7	6612	0739	DW	TAB39
00F9	6E12	0740	DW	TAB39
00FB	2B12	0741	DW	TAB40
00FD	2C12	0742	DW	TAB41
00FF	3112	0743	DW	TAB42
0E01	7512	0744	DW	TAB43
0E03	7512	0745	DW	TAB43
0E05	9912	0746	DW	TAB44
0E07	9912	0747	DW	TAB44
0E09	9912	0748	DW	TAB44
0E0B	AD12	0749	DW	TAB45
0E0D	E112	0750	DW	TAB46
0E0F	BD12	0751	DW	TAB47
0E11	BD12	0752	DW	TAB47
0E13	BD12	0753	DW	TAB47
0E15	ED12	0754	DW	TAB48
0E17	ED12	0755	DW	TAB48
0E19	2512	0756	DW	TAB49
0E1B	2512	0757	DW	TAB49
0E1D	2E12	0758	DW	TAB50
0E1F	2E12	0759	DW	TAB51
0E21	2712	0760	DW	TAB52
0E23	2712	0761	DW	TAB52
0E25	0F12	0762	DW	TAB53
0E27	F712	0763	DW	TAB54
0E29	F712	0764	DW	TAB54
0E2B	F712	0765	DW	TAB54
0E2D	8C14	0766	DW	TAB55
0E2F	8C14	0767	DW	TAB55
0E31	1914	0768	DW	TAB56
0E33	1914	0769	DW	TAB56
0E35	2514	0770	DW	TAB57

+++ IN +++

0E37	3D14	0771	DW	TAB57
0E39	3F14	0772	DW	TAB58
0E3B	3F14	0773	DW	TAB58
0E3D	4D14	0774	DW	TAB59
0E3F	4D14	0775	DW	TAB59
0E41	6D14	0776	DW	TAB60
0E43	6D14	0777	DW	TAB60
0E45	7D14	0778	DW	TAB61
0E47	7D14	0779	DW	TAB61
0E49	7D14	0780	DW	TAB61
0E4B	7D14	0781	DW	TAB61
0E4D	9D14	0782	DW	TAB62
0E4F	9D14	0783	DW	TAB62
0E51	AD14	0784	DW	TAB62
0E53	AD14	0785	DW	TAB62
0E55	8914	0786	DW	TAB64
0E57	8914	0787	DW	TAB64
0E59	8914	0788	DW	TAB64
0E5B	8914	0789	DW	TAB64
0E5D	0714	0790	DW	TAB65
0E5F	0714	0791	DW	TAB65
		0792		SEUDO CODIGO
		0793		
		0794		RUTINA FRUERA
		0795		
		0796		INICIO
		0797		AFUNTOR INICIO TABLA CIR. CONOCIDOS
		0798		0 -ENCENTRADO -FALSO
		0799		1 -REPITE
		0800		A -SI TABLA CIR. CONOCIDOSI AFUNTORI-CADENA
		0801		1 -ENTONCES
		0802		A -ENCENTRADO VERDADERO
		0803		2 -DE LO CONTRARIO
		0804		A -AFUNTORI -AFUNTORI+1
		0805		1 -HASTA ENCENTRADO VERDADERO O AFUNTORI
		0806		FIN DE TABLA+1
		0807		0 -SI ENCENTRADO VERDADERO
		0808		A -ENTONCES
		0809		1 -EJECUTA CHECK
		0810		B -DE LO CONTRARIO
		0811		1 -MENSAJE DESCONOCIDO
		0812		FIN
		0813		
		0814		
		0815		CODIGO
		0816		
		0817		VARIABLES
		0818		AFUNTORI
		0819		ENCENTRADO
				REG DE
				REG. C

		0820	,	CADENA		MEMORIA APUNTADO POR HL
		0821	,	MENSAJE		HL
		0822				
		0823				
		0824		FRUE.		
0E61	11E109	0825		LD	DE, TCC	, 1. -
CROMENCO Z80 MACRO ASSEMBLER VERSION 03. 07						
+++ IN +++						
0E64	0E00	0826		LD	C, 0	, 2. -
		0827	REP.			, 3. -
0E66	0E04	0828		LD	B, 4	, 3. A. -
0E68	E0	0829		PUSH	HL	
0E69	D0	082A		PUSH	DE	
0E6A	1A	082B	REP1.	LD	A, (DE)	
0E6B	3E	082C		CF	A, (HL)	
0E6C	1D	082D		INC	DE	
0E6D	2D	082E		INC	HL	
0E6E	C27A0E	R 082F		JP	NZ, FRUSAC	
0E71	10F7	0830		OUND	REP1	
0E73	D1	0831		POP	DE	
0E74	E1	0832		POP	HL	
		0833				, D. A. 1. -
0E75	0E7F	0834		LD	C, 0FFH	
0E77	0E810E	R 0835		JP	FRU1	
		0836	FRUSAC			, D. A. 2. -
0E7A	D10100	0837		LD	HL, 4	
0E7D	D1	0838		POP	DE	
0E7E	19	0839		ADD	HL, DE	
0E7F	EB	083A		EM	DE, HL	
0E80	E1	083B		POP	HL	
0E81	E0	083C	FRU1.	PUSH	HL	, 4. -
0E82	D1E100	083D		LD	HL, FINTAB	
0E85	87	083E		CA	A, A	ALINPIA BANDERA CY
0E86	E002	083F		SEC	HL, DE	
0E88	E1	0840		POP	HL	
0E89	0A8A0E	R 0841		JP	C, FRUSB	
0E8C	DE00	0842		LD	A, 0	
0E8E	E9	0843		CF	A, C	
0E8F	0A8E0E	R 0844		JP	C, REP	
		0845	CAMBIO DE DIRECCION DE TABLAS			
0E91	EB	0846		EM	DE, HL	
0E93	11E109	0847		LD	DE, TCC	
0E96	87	0848		CA	A, A	
0E97	E002	0849		SEC	HL, DE	
0E99	CE00	084A		FRU1	A	
0E9E	CE10	084B		FRU1	L	
0E9D	11E100	084C		LD	DE, TABLAS	
0E9F	19	084D		ADD	HL, DE	
0EA1	3E	084E		LD	E, (HL)	
0EA2	2D	084F		INC	HL	

```

06AD 06      0668      LD 0, (HL)
06A1 EB      0669      EX DE, HL
                0670      , 0. -
                0671      , 0. A. -
                0672
06AD 000514  0673      CALL  CHECA      , 0. A. 1. -
                0674
06A8 09      0675      RET
                0676      FRUGB.      , 0. B. -
06A9 010015  0677      LD      HL, DECONO  , 0. B. 1. -
06AC 09      0678      RET
                0679
                0680

```

CROMHECO 130 MACRO ASSEMBLER VERSION 03. 07

+++ IN +++

```

0681      ,      RUTINA CHECA
0682
0683
0684      , ELABORA F T
0685      , OBJETIVO VERIFICAR EL RESULTADO DEL CIRCUITO A PROBAR CON LAS
0686      , TABLAS DE PRUEBA
0687      , DATOS DE ENTRADA LA DIRECCION DE LAS TABLAS DE PRUEBA
0688      , DATOS DE SALIDA EL RESULTADO DEL CIRCUITO INTEGRADO A PRUEBA
0689      , PROCEDIMIENTO PRIMAMENTE DEDINIR LAS ENTRADAS Y SALIDAS DEL
0690      , PIO. ESTO ES, LA PALABRA DE CONTROL PARA PROGRAMARLO -
0691      , Y ADEMAS SELECCIONARLO PARA EL MODO D QUE IMPLICA EN--
0692      , TRADA Y SALIDA DE DATOS SIN PROTOCOLO
0693      , EL SIGUIENTE PASO SON LAS CONDICIONES INICIALES PARA-
0694      , PODER OPERARLO NOTANDO QUE EXISTEN CIRCUITOS QUE NO LAS
0695      , NECESITAN ESTAS SE PODRAN VER EN LAS TABLAS CON LOS CIR-
0696      , CUITOS QUE LAS UTILISEN
0697      , EL TERCER PASO SON LAS TABLAS DE PRUEBA Y EL NUMERO DE
0698      , PRUEBAS, EN LA CUAL ESTAN CONTENIDAS LOS DATOS DE ENTRA-
0699      , DA Y SALIDA DEL CIRCUITO INTEGRADO EN ESTE PUNTO LAS SA-
0700      , LIDAS (PIO) SON ENTRADAS DEL CIRCUITO INTEGRADO Y SON -
0701      , LOS QUE NOS VAN AYUDAR A PROBARLO. LAS ENTRADAS (PIO) -
0702      , SON SALIDAS DEL CIRCUITO. ESTOS SON LOS RESULTADOS VER-
0703      , DADEROS, LOS CUALES SE VAN A COMPARAR CON LOS RESULTADOS
0704      , QUE NOS DE AL PROBARLO, SI EL RESULTADO ES IGUAL AL DE -
0705      , LA TABLA, EL RESULTADO SE TOMARA COMO BUENO DE LO CONTRA-
0706      , RIO SE TOMARA COMO MALO O DESCONOCIDO
0707      , UNA VEZ CONOCIDO EL RESULTADO NOS INFORMARA
0708
0709
0710      TAB1.
0711      DB 0A1H, 004H, 000H, 001H, 0A1H, 0E4H
0712      DB 0A0H, 0E0H, 0E6H, 0FAH, 016H
0713      DB 0F6H
0714      TAB2.
0715      DB 0A9H, 012H, 0A0H, 0A4H, 0A9H, 0F2H

```

0E6F	9EF62DF8	0916		DB 036H, 0F6H, 02DH, 0FEH, 0E6H
0EC4	ED	0917		DB 0EDH
0EC3	09120004	0918	TABD	DB 003H, 012H, 00AH, 004H, 009H, 0F2H
0EC6	32E424E9	0919		DB 092H, 0E4H, 024H, 0E9H, 0E6H
0ED0	ED	0920		DB 0EDH
		0921		
0ED1	AA9A0602	0922	TAB4	DB 0AAH, 06AH, 000H, 06CH, 0AAH, 0EAH
0ED7	15F5	0923		DB 61CH, 0F5H
		0924		
0ED9	AA9A0602	0925	TAB5	DB 0AAH, 0AAH, 000H, 062H, 000H, 0E0H
0EDF	BFFF	0926		DB 0EFH, 0FFH
		0927		
0EE1	A1040004	0928	TAB6	DB 0A4H, 004H, 000H, 004H, 000H, 0EDH
0EE7	09E912F2	0929		DB 003H, 0E9H, 012H, 0F2H, 0BFH
0EEC	FF	0930		DB 0FFH
		0931		
0EED	A0030003	0932	TAB7	DB 0ACH, 003H, 000H, 003H, 0ACH, 0E6H
0EFD	A5E9A8EA	0933		DB 0ACH, 0E9H, 0AAH, 0EAH, 0AFH
0EFS	EBE0FC63	0934		DB 0E6H, 0E0H, 0FCH, 0E6H, 0FCH
0EFD	BAFE1FF7	0935		DB 0BAH, 0FEH, 01FH, 0F7H

CROMEMCO 286 MACRO ASSEMBLER VERSION 02.07

\*\*\* IN \*\*\*

		0936		
0F01	A0030003	0937	TAB8	DB 0ACH, 003H, 000H, 003H, 0ACH, 0E6H
0F07	03E10A82	0938		DB 00CH, 0E1H, 00AH, 0E2H, 06FH
0F0C	E310F415	0939		DB 0E2H, 010H, 0F4H, 015H, 0F5H
0F11	1AF6BFFF	0940		DB 01AH, 0F6H, 0BFH, 0FFH
		0941		
0F13	A0030010	0942	TAB9	DB 0ACH, 003H, 000H, 010H, 0AAH, 0E4H
0F1B	A5E9A8EA	0943		DB 0ACH, 0E9H, 0AAH, 0EAH, 0AFH
0F20	A7A9E6A9	0944		DB 0E7H, 0ACH, 0E6H, 0E6H, 0E6H
0F23	A5E9A8EA	0945		DB 0AAH, 0E9H, 0AFH, 0E6H, 0E4H
0F2A	F8E0FC63	0946		DB 0F4H, 0E6H, 0FCH, 0E6H, 0FCH
0F2F	07F7B0C0	0947		DB 0E7H, 0F7H, 0E6H, 0FCH, 0E6H
0F34	F0E6E21F	0948		DB 0FCH, 0E6H, 0FEH, 01FH, 0FFH
		0949		
0F39	A0030010	0950	TAB10	DB 0ACH, 003H, 000H, 010H, 004H, 0E4H
0F3F	03E0C6B5	0951		DB 003H, 0E6H, 006H, 0E6H, 007H
0F44	A7A9E6A9	0952		DB 0E7H, 00CH, 0E6H, 006H, 0E6H
0F49	0E6E6E6F	0953		DB 00E, 0E6H, 0AFH, 0E6H, 014H
0F4E	F410F016	0954		DB 0F4H, 010H, 0FCH, 016H, 0E6H
0F53	17F717F0	0955		DB 017H, 0F7H, 01CH, 0F7H, 01CH
0F58	F0E6E21F	0956		DB 0FCH, 01E, 0FEH, 0AFH, 0FFH
		0957		
0F5F	A0030010	0958	TAB11	DB 0ACH, 003H, 000H, 010H, 0AAH, 0E4H
0F65	A5E9A8EA	0959		DB 0ACH, 0E9H, 0AAH, 0EAH, 0AFH
0F6B	E70C6E0D	0960		DB 0E6H, 0F7H, 0E6H, 0E6H, 010H
0F70	0E6E6E6F	0961		DB 0E6H, 0E6H, 0AFH, 0E6H, 014H
0F77	F410F016	0962		DB 0F4H, 010H, 0FCH, 016H, 0E6H
0F7D	17F717F0	0963		DB 017H, 0F7H, 01CH, 0F7H, 01CH



0F7C	FC1EF1F	0964		00 0F0H, 01EH, 0FEH, 01FH, 0FFH
		0965		
0F81	ACC9A003	0966	TAB12	00 0A0H, 002H, 000H, 003H, 0A0H, 0E5H
0F87	00E10AEE	0967		00 000H, 0E1H, 00AH, 0E2H, 00FH
0F8C	E310F410	0968		00 0E2H, 010H, 0F4H, 010H, 0F0H
0F91	1AF61FF7	0969		00 01AH, 0F6H, 01FH, 0F7H
		0970		
0F9D	30000010	0971	TAB1D	00 000H, 000H, 000H, 010H, 000H, 0F0H
0FA6	81F182F3	0972		00 001H, 0F0H, 002H, 0F0H, 002H
0FA0	F030F037	0973		00 0F0H, 002H, 0F0H, 002H, 0F0H
0FA3	80F13FF3	0974		00 000H, 0F0H, 000H, 0F0H, 002H
0FAA	F030F037	0975		00 0F0H, 002H, 0F0H, 002H, 0F0H
0FAD	80F72FF7	0976		00 0A0H, 0F7H, 00FH, 0F7H, 00FH
0F84	F000FF1F	0977		00 0F0H, 000H, 0FFH, 0F0H, 0FFH
		0978		
0FB9	A4040004	0979	TAB14	00 0A4H, 004H, 000H, 004H, 000H, 0E0H
0FBF	ACE0E0F6	0980		00 0A0H, 0E0H, 0E4H, 0F0H, 00FH
0FC4	FF	0981		00 0FFH
		0982		
0FC0	7F07000A	0983	TAB10	00 07FH, 007H, 000H, 00AH, 07EH, 007H
0FC8	700770A7	0984		00 070H, 007H, 070H, 0A7H, 077H
0FC0	E70F270F	0985		00 0E7H, 00FH, 097H, 00FH, 007H
0FC0	0F0770FF6	0986		00 00FH, 007H, 07FH, 0F0H, 07FH
0FC4	807F0E	0987		00 000H, 07FH, 000H
		0988		
0FC0	7F07000A	0989	TAB16	00 07FH, 007H, 000H, 00AH, 07EH, 007H
0FC8	700770A7	0990		00 070H, 007H, 070H, 0A7H, 077H
CRASHEND0 D80 HALFC ASSEMBLER VERSION 00 07				
*** IN ***				
0FB3	070FF03F	0991		00 0E7H, 00FH, 0F7H, 00FH, 00FH
0FB0	0F0770A7	0992		00 00FH, 00FH, 07FH, 00FH, 07FH
0FB0	E07F9B	0993		00 0E0H, 07FH, 000H
		0994		
0FB0	7F07000A	0995	TAB17	00 07FH, 007H, 000H, 00AH, 07EH, 007H
0FB0	700770A7	0996		00 070H, 007H, 070H, 0A7H, 077H
1000	070F270F	0997		00 007H, 00FH, 097H, 00FH, 007H
1000	0F0770FF6	0998		00 00FH, 007H, 07FH, 0F0H, 07FH
100A	807F0E	0999		00 0E0H, 07FH, 000H
		1000		
1000	7F07000A	1001	TAB18	00 07FH, 007H, 000H, 00AH, 07EH, 007H
1010	700770A7	1002		00 070H, 007H, 070H, 0A7H, 077H
1010	E70F270F	1003		00 0E7H, 00FH, 097H, 00FH, 007H
1010	0F0770FF6	1004		00 00FH, 007H, 07FH, 0F0H, 07FH
1010	807F0E	1005		00 000H, 07FH, 000H
		1006		
1000	007F000A	1007	TAB19	00 000H, 07FH, 000H, 00AH, 010H, 0A0H
100B	00F10004	1008		00 070H, 0F0H, 010H, 004H, 000H
1070	011E900E	1009		00 001H, 01EH, 002H, 00EH, 009H
1000	1F900F0D	1010		00 01FH, 000H, 00FH, 0E0H, 070H
107A	007000	1011		00 0A0H, 070H, 002H

		1012		
102D	007F000A	101D	TAB20	DB 000H, 07FH, 000H, 000H, 01CH, 0DFH
104D	008C108E	1014		DB 00CH, 0ECH, 01CH, 0E8H, 05CH
1048	8E1EECDE	1015		DB 08EH, 01EH, 0ECH, 00EH, 0F6H
1040	1FE70F9C	1016		DB 01FH, 0E7H, 00FH, 09CH, 0E2H
1052	FF70FC	1017		DB 0FFH, 07CH, 0FCH
		1018		
1055	A01F000A	1019	TAB21	DB 0A0H, 01FH, 000H, 009H, 0A4H, 0F7H
105E	14E3A0EE	1020		DB 014H, 0E3H, 0A0H, 0EEH, 090H
1060	EF06FB96	1021		DB 0EFH, 006H, 0F6H, 096H, 0F0H
1065	A7F917E7	1022		DB 0A7H, 0F9H, 017H, 0E7H, 0A0H
106A	FF10FF	1023		DB 0FFH, 01CH, 0FFH
		1024		
106D	A0000010	1025	TAB22	DB 0A0H, 000H, 000H, 010H, 0A0H, 0E0H
107D	A0E0A1FC	1026		DB 0A0H, 0E0H, 0A4H, 0F0H, 007H
1078	FC8E0A8	1027		DB 0F0H, 0A0H, 0E0H, 0A0H, 0E0H
1070	ACF00FFD	1028		DB 0A0H, 0F0H, 00FH, 0F0H, 0E0H
1082	EEB3EE64	1029		DB 0EEH, 0B0H, 0EEH, 0E4H, 0FEH
1087	17FE18EF	1030		DB 017H, 0FFH, 013H, 0EFH, 018H
108C	EF10FF1F	1031		DB 0EFH, 01CH, 0FFH, 01FH, 0FFH
		1032		
1091	A0B00010	1033	TAB23	DB 0A0H, 000H, 000H, 010H, 0A0H, 0E0H
1097	A0E0A1E8	1034		DB 0A0H, 0E0H, 0A4H, 0E8H, 0E7H
109C	E8A3F0A8	1035		DB 0E8H, 0A0H, 0F0H, 0A0H, 0F0H
10A1	ACF00FF3	1036		DB 0A0H, 0F0H, 00FH, 0F0H, 0E0H
10A7	E19E1E64	1037		DB 0E1H, 0E0H, 0E1H, 0E4H, 0E0H
10AB	87E998F1	1038		DB 0E7H, 0E9H, 098H, 0F4H, 098H
10B0	F19CF91F	1039		DB 0F1H, 0A0H, 0F9H, 01FH, 0F9H
		1040		
10E0	A000000E	1041	TAB24	DB 0A0H, 000H, 000H, 00EH, 0A0H, 0E0H
10E6	A0E0A1E0	1042		DB 0A0H, 0E0H, 0A0H, 0E0H, 0A0H
10CC	E0A1E0A0	1043		DB 0E0H, 0A1H, 0E0H, 0A0H, 0E0H
10C0	8E0A0E0	1044		DB 0E0H, 0E0H, 0A4H, 0E0H, 0A0H
10CA	FC0E0F38	1045		DB 0F0H, 0E0H, 0E0H, 0E0H, 0E0H
CROMENCO D80 MACRO ASSEMBLER VERSION 02 07				
*** IN ***				
10CF	2E0010FC	1046		DB 0E0H, 0E0H, 0E1H, 0F0H, 0E0H
10D4	FF	1047		DB 0FFH
		1048		
10DD	8000000E	1049	TAB25	DB 0E0H, 000H, 000H, 00EH, 0E0H, 0E0H
10E6	00F00E0	1050		DB 0E0H, 0E0H, 0E0H, 0E0H, 0E0H
10E0	E0A1E0F0	1051		DB 0E0H, 0E4H, 0E0H, 0E0H, 0F0H
10ED	00F00E0	1052		DB 0E0H, 0E0H, 0E0H, 0E0H, 0E0H
10EA	E00000CF	1053		DB 0E0H, 000H, 0F0H, 0E0H, 0FFH
		1054		
10EF	A0B0000E	1055	TAB26	DB 0A0H, 0A0H, 000H, 000H, 0A0H, 0E0H
10F0	80F00E0	1056		DB 0E0H, 0E0H, 0F0H, 0E0H, 0A0H
10FA	F0A0F0A0	1057		DB 0E4H, 0A0H, 0E0H, 0E0H, 0F0H
10FF	80E00E0	1058		DB 0E0H, 0E0H, 0A0H, 0E0H, 0E0H
1104	FF	1059		DB 0FFH

		1060		
1100	0000000A	1061	TAB07	0B 000H, 000H, 000H, 000H, 000H, 000H
1106	00E100E0	1062		0B 000H, 001H, 000H, 000H, 000H
1110	0000E090	1063		0B 000H, 000H, 000H, 000H, 001H
1115	00E100E0	1064		0B 000H, 000H, 000H, 000H, 001H
111A	000FFF	1065		0B 000H, 000H, 000H
		1066		
1110	0000000A	1067	TAB08	0B 000H, 000H, 000H, 000H, 000H, 000H
1100	00E100E0	1068		0B 000H, 001H, 000H, 000H, 000H
1106	0000E090	1069		0B 000H, 000H, 000H, 000H, 001H
1120	00E100E0	1070		0B 000H, 000H, 000H, 000H, 001H
1125	000FFF	1071		0B 000H, 000H, 000H
		1072		
1120	A000000A	1073	TAB09	0B 000H, 000H, 000H, 000H, 001H, 000H
1125	A000000F	1074		0B 000H, 000H, 000H, 000H, 000H
1130	A0000000	1075		0B 000H, 000H, 000H, 000H, 000H
1137	A000000F	1076		0B 000H, 000H, 000H, 000H, 000H
114A	A0000000	1077		0B 000H, 000H, 000H
		1078		
1147	A0000009	1079	TAB0A	0B 000H, 000H, 000H, 000H, 000H, 000H
1157	A000000E	1080		0B 000H, 000H, 000H, 000H, 000H
115A	A0000000	1081		0B 000H, 000H, 000H, 000H, 000H
1160	A000000F	1082		0B 000H, 000H, 000H, 000H, 000H
1162	00	1083		0B 000H
		1084		
1167	A0000008	1085	TAB01	0B 000H, 000H, 000H, 000H, 000H, 000H
1177	A000000F	1086		0B 001H, 000H, 000H, 000H, 000H
1182	A000000F	1087		0B 000H, 000H, 000H, 000H, 000H
1170	A000000F	1088		0B 000H, 000H, 000H, 000H
		1089		
1177	A000000A	1090	TAB0B	0B 000H, 000H, 000H, 000H, 000H, 000H
1170	A000000F	1091		0B 001H, 000H, 000H, 000H, 000H
1182	A000000F	1092		0B 000H, 000H, 000H, 000H, 000H
1187	A000000F	1093		0B 000H, 000H, 000H, 000H, 000H
1190	A000000F	1094		0B 000H, 000H, 000H
		1095		
119F	A000000A	1096	TAB0C	0B 000H, 000H, 000H, 000H, 000H, 000H
1190	A0000000	1097		0B 000H, 000H, 000H, 000H, 000H
119A	A000000F	1098		0B 000H, 000H, 000H, 000H, 000H
119F	A0000000	1099		0B 000H, 000H, 000H, 000H, 000H
11A4	00	1100		0B 000H
REVISION 7.00 TAB01: A000000A: REVISION 7.00				
11A7	A000000A	1101	TAB01	0B 000H, 000H, 000H, 000H, 000H, 000H
11A6	A0000000	1102		0B 000H, 000H, 000H, 000H, 000H
11A0	A000000F	1103		0B 000H, 000H, 000H, 000H, 000H
11A5	A000000F	1104		0B 000H, 000H, 000H, 000H, 000H
11A8	A0000000	1105		0B 000H, 000H, 000H
11BD	A000000F	1106	TAB0D	0B 000H, 001H, 000H, 000H, 001H, 001H
1100	A000000A	1107		0B 000H, 000H, 000H, 000H, 000H

1108	FF19FDAC	1108	CB	0FFH	019H	0FCH	0ACH	0F6H
1100	95F2	1109	CB	0A9H	0F2H			
110F	51C70002	1110	TAB26	DB	021H	0E7H	000H	003H
1103	7ED31002	1111		DB	07EH	003H	010H	0C2H
1109	0020000A	1112	TAB27	DB	000H	056H	000H	00AH
110F	5C20F72C	1113		DB	02CH	022H	0F7H	02CH
11E4	22FFC00E	1114		DB	02CH	0FFH	0C2H	00EH
11E9	F70006AA	1115		DB	0F7H	0C2H	0F6H	0AAH
11EE	AE0E0C0	1116		DB	0A9H	00EH	0C0H	
11F1	0020000E	1117	TAB28	DB	000H	056H	000H	00AH
11F7	5C20F72C	1118		DB	02CH	022H	0F7H	02CH
11FD	22FFC00E	1119		DB	02CH	0FFH	0C2H	00EH
1201	F70006AA	1120		DB	0F7H	0C2H	006H	0AAH
1208	AE0E0C0F	1121		DB	0A9H	00EH	0C0H	0FFH
120E	7E00000A	1122	TAB29	DB	02EH	000H	000H	00AH
1211	4AF011FE	1123		DB	014H	0F2H	014H	0FEH
1218	F810FF0A	1124		DB	0FAH	01CH	0FAH	0FEH
121E	AD0E00FA	1125		DB	0A9H	0FEH	020H	0FAH
1220	FFA8FE	1126		DB	0FFH	0A9H	0FEH	
1223	50190006	1127	TAB30	DB	000H	019H	000H	0FEH
1229	5AF01002	1128		DB	02AH	00FH	020H	00AH
122E	F8F000FF	1129		DB	0FAH	00FH	00EH	00EH
1232	F8F000FF	1130		DB	0FEH	0F0H	0FAH	0FEH
1238	88FF00FF	1131		DB	0FEH	0FEH	0F0H	0FFH
123E	7E000006	1132	TAB31	DB	02AH	000H	000H	00AH
1244	F8F010FE	1133		DB	02AH	0C1H	019H	0FEH
1248	F810000F	1134		DB	0FAH	01EH	0E0H	0A9H
124E	00E000FA	1135		DB	02CH	0FEH	020H	0FAH
1251	81000010	1136	TAB32	DB	001H	00AH	000H	010H
1257	EE010001	1137		DB	0E9H	001H	0E0H	001H
1260	00E000FF	1138		DB	00AH	0E9H	0E9H	0E9H
1261	00E00001	1139		DB	0E9H	0E9H	0E9H	0E9H
1265	00E00001	1140		DB	00AH	0E9H	00AH	00FH
1268	00E000FE	1141		DB	0E9H	0C2H	0E9H	0FEH
1270	00E000FE	1142		DB	0FEH	0E1H	0FEH	0E9H
1277	00E00010	1143	TAB33	DB	02CH	0E1H	000H	010H
127E	10071000	1144		DB	01EH	00CH	019H	00CH
1280	00E00007	1145		DB	0A9H	0E1H	010H	0E7H
1283	00E0000E	1146		DB	0E9H	010H	0E9H	070H
128A	00E0000A	1147		DB	02CH	0FEH	0C2H	0FAH
128F	00E00000	1148		DB	00AH	0A9H	010H	00CH
1294	00E00007	1149		DB	0A9H	01EH	0A0H	0FEH
1299	00E00001	1150	TAB34	DB	0A9H	0FAH	000H	000H
129F	AE0E00FA	1151		DB	0A9H	0E0H	0E9H	01EH
12A4	FE	1152		DB	0FEH			
12AD	00E00001	1153	TAB35	DB	00AH	0FAH	000H	000H
12B6	00E00007	1154		DB	0A9H	0E0H	0E9H	0E9H
12BA	FE	1155		DB	0FEH			

REPERMITS THE USER TO CONTROL THE OPERATION OF THE  
 \*\*\* IN \*\*\*

1281	34090004	1156	TAB16.	DB	024H, 009H, 000H, 004H, 02CH, 0EDH
1287	0FFFA0E0	1157		DB	0DFH, 0FFH, 0ACH, 0EDH, 039H
128C	E1	1158		DB	0E4H
128E	00000207	1159	TAB17.	DB	000H, 000H, 000H, 007H, 020H
12C2	01C00900	1160		DB	001H, 020H, 009H, 000H, 009H
12C7	01030000	1161		DB	001H, 029H, 000H, 000H, 0E1H
12CC	00000931	1162		DB	020H, 000H, 009H, 0E1H, 029H
12D1	00040104	1163		DB	000H, 004H, 001H, 004H, 000H
12D6	0819FD98	1164		DB	000H, 019H, 0FDH, 098H, 0D4H
12DB	99F49600	1165		DB	099H, 0F4H, 096H, 0DDH, 099H
12E0	F0430040	1166		DB	0FDH, 019H, 000H, 019H, 0FDH
12E5	F9FD99F0	1167		DB	0F9H, 0FDH, 099H, 0FDH, 017H
12EA	F011F0	1168		DB	0F0H, 011H, 0F0H
12ED	0000027F	1169	TAB18	DB	000H, 000H, 000H, 07FH, 0F0H
12F2	1FF0181E	1170		DB	01FH, 0F0H, 018H, 01EH, 000H
12F7	1EFD1F03	1171		DB	01EH, 0F0H, 01FH, 0D3H, 01FH
12FC	F01E041E	1172		DB	0F0H, 01EH, 0D4H, 01EH, 0F4H
1301	1FD01FF0	1173		DB	01FH, 000H, 01FH, 0F0H, 09EH
1306	019EF19F	1174		DB	0D1H, 09EH, 0F1H, 09FH, 0E9H
130B	9FF91E01	1175		DB	09FH, 0F9H, 01EH, 0D1H, 01EH
1310	F11FD094F	1176		DB	0F1H, 01FH, 009H, 01FH, 0F9H
1315	1E041EEF1	1177		DB	01EH, 0D4H, 01EH, 0F1H, 01FH
131A	F04FF09F	1178		DB	000H, 01FH, 0F0H, 09EH, 000H
131F	F0F09F08	1179		DB	09EH, 0F0H, 09FH, 000H, 09FH
1324	F0	1180		DB	0F0H
1329	0000027F	1181	TAB19	DB	000H, 000H, 000H, 07FH, 0F0H
132E	F9F02078	1182		DB	079H, 0F0H, 009H, 079H, 0D0H
1333	F0F079F8	1183		DB	079H, 0F0H, 079H, 009H, 079H
1338	F0700178	1184		DB	0F0H, 079H, 0D1H, 079H, 0F1H
133D	F90799F9	1185		DB	079H, 009H, 079H, 0F9H, 0F0H
1342	10F0F0F9	1186		DB	000H, 0F0H, 0F0H, 0F9H, 0D9H
1347	F0F0F001	1187		DB	0F0H, 0F0H, 0F0H, 0D1H, 0F0H
134C	F1F009F9	1188		DB	0F1H, 0F9H, 0D9H, 0F0H, 0F0H
1350	F00478F4	1189		DB	079H, 0D4H, 079H, 0F4H, 079H
1355	0078F078	1190		DB	000H, 079H, 0F0H, 079H, 0D0H
135A	F0F07800	1191		DB	079H, 0F0H, 079H, 0D0H, 079H
135F	F0F004F0	1192		DB	0FDH, 0F0H, 0D4H, 0F0H, 0F4H
1364	F000F0F0	1193		DB	0F9H, 000H, 0F0H, 0F0H, 0F0H
1369	00F0F0F9	1194		DB	000H, 0F0H, 0F0H, 0F9H, 000H
136E	F0F0	1195		DB	0F0H, 0F0H
1373	001E00FF	1196	TAB20	DB	000H, 01EH, 000H, 0FFH, 0EDH
1378	F0E00000	1197		DB	0F0H, 0EDH, 000H, 070H, 000H
137D	F0E0F000	1198		DB	070H, 0EDH, 0F0H, 000H, 0F0H
1382	F0700170	1199		DB	0F0H, 070H, 0D1H, 070H, 0E1H
1387	F004F0F1	1200		DB	0F0H, 0D1H, 0F0H, 0F1H, 070H
138C	007000F0	1201		DB	000H, 070H, 0EDH, 0F0H, 0D0H
1391	F0F0F000	1202		DB	0F0H, 0F0H, 070H, 000H, 070H
1396	00F000F0	1203		DB	0EDH, 0F0H, 000H, 0F0H, 0F0H
139B	F00F0000	1204		DB	070H, 000H, 070H, 0EDH, 0F0H
13A0	00F0F0F0	1205		DB	000H, 0F0H, 0F0H, 070H, 009H
13A5	F0E0F000	1206		DB	070H, 0EDH, 0F0H, 0D9H, 0F0H

1284	F970C87C	1287	DB 0F3H, 07CH, 0CAH, 07CH, 0E9H
1289	F08AF0FA	1288	DB 0FCH, 0DAH, 0FCH, 0FAH, 07CH
128E	C870EBFC	1289	DB 0CEH, 07CH, 0E9H, 0FCH, 0D6H
1293	F0FB	1210	DB 0FCH, 0FBH

CROMEMCO Z80 MACRO ASSEMBLER VERSION 02 07

\*\*\* IN \*\*\*

12E5	00010007	1211	TABD1	DB 000H, 001H, 000H, 007H, 010H, 002H
12E8	0FF01001	1212		DB 0CFH, 0FCH, 010H, 001H, 013H
12C0	D198D1D0	1213		DB 0D1H, 093H, 0D1H, 0C0H, 0D1H
12C5	9801	1214		DB 09CH, 0D1H
12C7	001E000A	1215	TABD2	DB 000H, 01EH, 00CH, 00AH, 0A3H, 0E0H
12C0	00E00A1E1	1216		DB 003H, 0E0H, 0D1H, 0E1H, 0E0H
12D2	F000E320	1217		DB 0FCH, 00CH, 0E3H, 0C0H, 0E4H
12D7	00E200E1	1218		DB 0CCH, 0FCH, 00CH, 0E1H, 0A0H
12D0	00A0E1	1219		DB 0E0H, 0A0H, 0E1H
12CF	001B000A	1220	TABD3	DB 0CCH, 01EH, 00CH, 00AH, 0A3H, 0C0H
12E5	20001000	1221		DB 0E3H, 0C0H, 043H, 0C0H, 0C3H
12EA	D0000C0C	1222		DB 0C0H, 0C3H, 0C0H, 0C3H, 0C0H
12EF	00C11000	1223		DB 0C3H, 0C1H, 043H, 0C0H, 0A3H
12F1	00E300	1224		DB 0C0H, 0E3H, 0C0H
12F7	00700009	1225	TABD4	DB 000H, 07EH, 00CH, 003H, 010H, 000H
12FD	1000FEFA	1226		DB 010H, 00CH, 0FEH, 0FAH, 0A0H
1400	F70AF816	1227		DB 0FCH, 00AH, 0FAH, 01FH, 0A4H
1407	119D1091	1228		DB 011H, 0A3H, 010H, 01FH, 011H
1400	0H	1229		DB 00AH
1405	A0040001	1230	TABD5	DB 0A1H, 0D1H, 00CH, 0D1H, 00CH, 0E0H
141D	A0E10AFA	1231		DB 0A1H, 0EDH, 0FAH, 0FAH, 0E0H
141F	FF	1232		DB 0FFH
1419	A0040001	1233	TABD6	DB 0A1H, 0D1H, 00CH, 0D1H, 0A4H, 0E4H
141F	00E30AFA	1234		DB 0AFA, 0E0H, 0E0H, 0FAH, 0E0H
1404	FF	1235		DB 0FFH
1420	00FA000E	1236	TABD7	DB 0A3H, 0FFH, 00CH, 00EH, 07FH, 00AH
142E	00FF00FF	1237		DB 00AH, 0FFH, 040H, 0FFH, 0C0H
142A	0FA10FA0	1238		DB 0EFA, 0E1H, 0FAH, 0E0H, 0E0H
142D	0EFA0AFA	1239		DB 0E0H, 07FH, 0E1H, 0E0H, 0E0H
142F	FA0FA0E0	1240		DB 0FAH, 0E0H, 0FAH, 0E0H, 0E0H
143F	00F00000	1241	TABD8	DB 07FH, 00FH, 000H, 0C0H, 07FH, 00FH
1440	00000A0E	1242		DB 07CH, 0A0H, 0A0H, 0A0H, 0C0H
144A	00100E	1243		DB 0A0H, 00EH, 0FFH
1440	0001000A	1244	TABD9	DB 000H, 011H, 00CH, 00AH, 07FH, 0FFH
1457	0700000F	1245		DB 07FH, 0C0H, 0E0H, 0E0H, 01FH
145A	FA0E0E00	1246		DB 0FAH, 01EH, 0FFH, 0C0H, 0E0H
1450	0E0FA0E0	1247		DB 0E0H, 0FFH, 07FH, 0E0H, 07FH
1462	000EFA	1248		DB 0E0H, 07FH, 0A0H
1462	00E1000A	1249	TABDA	DB 0A0H, 0E1H, 0C0H, 00AH, 07FH, 0FFH
146E	0E0E0E0E	1250		DB 0A0H, 0E0H, 07FH, 0E0H, 0E0H
147A	0F000FA	1251		DB 0C0H, 0C0H, 0FAH, 0E0H, 0E0H
147E	0700000F	1252		DB 07FH, 00CH, 0C0H, 0C0H, 07FH
147A	0A0E00	1253		DB 0A0H, 0E0H, 01H
147E	00010009	1254	TABDB	DB 0A0H, 0D1H, 00CH, 00CH, 0E0H, 0E0H

148D	1C9C7C9F	1255		DB 01CH, 09CH, 07CH, 09FH, 02CH
148E	8A7CBF36	1256		DB 06AH, 07CH, 06FH, 026H, 026H
148F	7E3F2AAE	1257		DB 07EH, 09FH, 03AH, 0A2H, 07EH
1490	8F	1258		DB 06FH
149D	780F0306	1259	TAB6C	DB 073H, 00FH, 000H, 006H, 07FH, 0FFH
1499	393E099C	1260		DB 023H, 08EH, 009H, 09DH, 06DH
149E	8E70977E	1261		DB 086H, 07CH, 097H, 07EH, 0FFH
14AD	780F000F	1262	TAB6C	DB 073H, 00FH, 000H, 009H, 07FH, 0FFH
14A9	788E789C	1263		DB 073H, 08EH, 073H, 09DH, 07CH
14AE	8E7097D9	1264		DB 086H, 07CH, 097H, 029H, 0CFH
14B0	09DF6DCD	1265		DB 009H, 00FH, 056H, 0CFH, 07CH

CRONEMCO D80 MACRO ASSEMBLER VERSION 02.07  
 \*\*\* IN \*\*\*

14B8	DF	1266		CB 06FH
14B9	4806000D	1267	TAB64	CB 048H, 009H, 000H, 003H, 027H, 0F6H
14BF	049076EF	1268		DB 024H, 05CH, 07EH, 06FH, 012H
14C1	84778F	1269		DB 0A1H, 07FH, 06FH
14C7	4806000C	1270	TAB60	DB 048H, 009H, 000H, 003H, 07FH, 0FFH
14C0	8C9E566A	1271		DB 05CH, 05EH, 056H, 066H, 05EH
14C0	8C9766	1272		CB 0A0H, 027H, 06FH
		1273		
		1274		
		1275		IF PREG-OCORRIDA PUTINA CHECK
		1276		INICIO
		1277		1. -HALO FALSO
		1278		2. -ESCRIBE 06FH EN EL PUERTO DE CONTROL A DEL PIO (MODO D)
		1279		3. -ESCRIBE MEMORIA (DIRECCION DE TABLA) EN PUERTO DE CONTROL
		1280		4. A DEL PIO
		1281		5. -DIRECCION DE TABLA (DIRECCION DE TABLA+1
		1282		6. -ESCRIBE 09FH EN EL PUERTO DE CONTROL B DEL PIO
		1283		7. -ESCRIBE MEMORIA (DIRECCION DE TABLA) EN PUERTO DE CONTROL
		1284		8. B DEL PIO
		1285		9. -DIRECCION DE TABLA (DIRECCION DE TABLA+1
		1286		10. -CONTADOR MEMORIA (DIRECCION DE TABLA)
		1287		11. -MIENTRAS CONTADOR ES EJECUTA
		1288		12. A -DIRECCION DE TABLA (DIRECCION DE TABLA+1
		1289		13. B -ESCRIBE MEMORIA (DIRECCION DE TABLA) EN PUERTO DE DA--
		1290		14. TOE A DEL PIO
		1291		15. C -DIRECCION DE TABLA (DIRECCION DE TABLA+1
		1292		16. D -ESCRIBE MEMORIA (DIRECCION DE TABLA) EN PUERTO DE DA--
		1293		17. TOE B DEL PIO
		1294		18. E -CONTADOR
		1295		19. F -CONTADOR (CONTADOR-1
		1296		20. -DIRECCION DE TABLA (DIRECCION DE TABLA+1
		1297		21. -CONTADOR MEMORIA (DIRECCION DE TABLA)
		1298		22. -MIENTRAS CONTADOR O HALO VERDADERO EJECUTA
		1299		23. A -DIRECCION DE TABLA (DIRECCION DE TABLA+1
		1300		24. B -ESCRIBE MEMORIA (DIRECCION DE TABLA) EN PUERTO DE DA--
		1301		25. TOE A DEL PIO
		1302		26. C -DIRECCION DE TABLA (DIRECCION DE TABLA+1

1303 , D.-ESCRIBE MEMORIA (DIRECCION DE TABLA) EN PUERTO DE DA--  
 1304 , TOS B DEL PIO  
 1305 , E.-LEE PUERTO A DE DATOS DEL PIO EN ENTRADA  
 1306 , F.-DIRECCION DE TABLA.-DIRECCION DE TABLA-1  
 1307 , G.-SI ENTRADA<MEMORIA (DIRECCION DE TABLA)  
 1308 , 1.-ENTONCES MALO. VERDADERO  
 1309 , H.-LEE PUERTO B DE DATOS DEL PIO EN ENTRADA  
 1310 , I.-DIRECCION DE TABLA.-DIRECCION DE TABLA+1  
 1311 , J.-SI ENTRADA<MEMORIA (DIRECCION DE TABLA)  
 1312 , 1D.-SI MALO.-FALSO  
 1313 , A.-ENTONCES. MENSAJE.-MALO  
 1314 , B.-DE LO CONTRARIO. MENSAJE.-BUENO  
 1315  
 1316  
 1317 , CODIGO DE LA RutINA CHECA  
 1318 , VARIABLES DE LA RutINA  
 1319 , MALO FN REGISTRO C  
 1320 , DIRECCION DE TABLA REGISTRO HL

CROMENCO 280 MACRO ASSEMBLER VERSION 02.07

\*\*\* IN \*\*\*

	1321	.CONTADOR EN REGISTRO B		
	1322	.ENTRADA EN REGISTRO A		
	1323	.MENSAJE EN REGISTRO HL		
	1324			
	1325	.DIRECCIONES DE LOS PUERTOS		
(0062)	1326	CTROLA EQU 32H		
(0063)	1327	CTROLB EQU 33H		
(0060)	1328	DATOR EQU 30H		
(0061)	1329	DATOB EQU 31H		
	1330			
	1331	CHECA.		
14D5	1332	LD A,0	.1-	
14D7	1333	LD C,A		
14D8	1334	LD A,OFFH	.2-	
14DA	1335	OUT CTROLA,A		
14DC	1336	LD A,(HL)	.3-	
14DE	1337	OUT CTROLA,A		
14DF	1338	INC HL	.4-	
14E0	1339	LD A,OFFH	.5-	
14E2	1340	OUT CTROLB,A		
14E4	1341	LD A,(HL)	.6-	
14E6	1342	OUT CTROLB,A		
14E7	1343	INC HL	.7-	
14E8	1344	LD A,(HL)	.8-	
14E9	1345	LD B,A		
	1346	CHECAS.		
14EA	1347	LD A,0	.9-	
14EB	1348	CP A,B		
14ED	1349	JP Z,CHECAS		
14F0	1350	INC HL	.9A-	



14F1	7E	1351		LD A, <HL>	.98. -
14F2	D380	1352		OUT DATOR, A	
14F4	23	1353		INC HL	.9C. -
14F5	7E	1354		LD A, <HL>	.9D. -
14F6	D381	1355		OUT DATOB, A	
14F8	85	1356		DEC B	.9E. -
14F9	CDEA14	R 1357		JF CHECAS	
14FC	23	1358	CHECA9.	INC HL	.10. -
14FD	7E	1359		LD A, <HL>	.11. -
14FE	47	1360		LD B, A	
		1361	CHECA11.		
14FF	DE00	1362		LD A, 0	.12. -
1501	83	1363		CF A, B	
1502	CA2713	R 1364		JF 3, CHECA12	
1505	83	1365		CF A, C	
1506	CA2713	R 1366		JF N3, CHECA12	
1509	23	1367		INC HL	.12A. -
150A	7E	1368		LD A, <HL>	.12B. -
150B	5F	1369		LD E, A	
150C	D380	1370		OUT DATOR, A	
150E	13	1371		INC HL	.12C. -
150F	7E	1372		LD A, <HL>	.12D. -
1510	57	1373		LD D, A	
1511	D381	1374		OUT DATOB, A	
1512	D380	1375		IN A, DATOR	.12E. -

CRONENCO 280 MACRO ASSEMBLER VERSION 03 07  
 \*\*\* IN \*\*\*

1515	8F	1376		CF A, E	.12G. -
1516	CA1B15	R 1377		JF 3, CHECAH	
1519	8E01	1378		LD C, 01	.12G. 1. -
151B	D681	1379	CHECAH.	IN A, DATOB	.12H. -
151D	8A	1380		CF A, D	.12J. -
151E	CA2D15	R 1381		JF 3, CHECAK	
1521	8E01	1382		LD C, 01	.12J. 1. -
1523	85	1383	CHECAK.	DEC B	.12K. -
1524	CDFF14	R 1384		JF CHECA11	
1527	DE00	1385	CHECA12	LD A, 0	.1D. -
1529	83	1386		CF A, C	
152A	CA2D15	R 1387		JF 3, CHECAB	
152B	218015	1388		LD HL, HALO	.12A. -
152C	CD2A15	R 1389		JF FIN	
152D	218015	1390	CHECAB.	LD HL, BUENO	.12B. -
152E	03	1391	FIN	RET	
		1392			
		1393			
		1394		NONBRE. RUTINA INFORMIA	
		1395			
		1396		ELABORO. A F. T	
		1397			
		1398		OBJETIVO. INFORMAR AL USUARIO EL RESULTADO DE LA PRUEBA	

```

1399 ,      DEL CIRCUITO INTEGRADO DESEADO
1400
1401 ,      DATOS DE ENTRADA  DIRECCION DEL MENSAJE A ESCRIBIR
1402
1403 ,      DATOS DE SALIDA  NINGUNA
1404
1405 ,      PROCEDIMIENTO. UNA VEZ DADA LA DIRECCION DEL MENSAJE A -
1406 ,      ESCRIBIR, ESTE NOS INFORMARA EL RESULTADO COMO. BUE
1407 ,      NO, MALO Y DESCONOCIDO
1408
1409
1410 ,      SEUDO-CODIGO
1411
1412 ,      (INFOR)
1413
1414
1415 ,      INICIO
1416 ,      1. -APUNTADOR DISPLAY -INICIO DISPLAY
1417 ,      2. -REFITE
1418 ,      A. -DISPLAY(APUNT DISPLAY) -MENSAJE(DIRECCION DEL
1419 ,      MENSAJE)
1420 ,      B. -APUNTADOR DISPLAY APUNTADOR DISPLAY+1
1421 ,      C. -DIRECCION DEL MENSAJE DIRECCION DEL MENSAJE+1
1422 ,      D. -HASTA DIRECCION DEL MENSAJE -FIN DEL MENSAJE+1
1423 ,      FIN
1424
1425
1426 ,      CODIGO
1427
1428 INFOR
1429
1337 1100FE 1430 LD DE, OFE0CH
CAGENEMCO 230 MACRO ASSEMBLER VERSION 03 07
*** IN ***

133A 016400 1431 LD 80, 064H
133B 8060 1432 LDIR
133C 09 1433 RET
1434
1435
1340 00E0E0E 1436 BUENO 0E 00H, 006H, 006H, 006H, 019H
1341 00E0E0E 1437 0E 006H, 006H, 006H, 006H, 006H
1342 00E0E0E 1438 0E 00H, 006H, 006H, 006H, 00EH, 0FFH
1343 00E0E0E 1439 0E 00H, 006H, 006H, 007H, 00H
1344 00E0E0E 1440 0E 001H, 00EH, 00EH, 00EH, 001H
1345 00E0E0E 1441 0E 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
1346 00E0E0E 1442 0E 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
1347 00E0E0E 1443 0E 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
1348 00E0E0E 1444 0E 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
1349 00E0E0E 1445 0E 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
1350 00E0E0E 1446 0E 0FFH, 0FFH, 0FFH, 0FFH, 0FFH

```

157A	FFFFFFFF	1447		DB	OFFH, OFFH, OFFH, OFFH, OFFH, OFFH
		1448			
1580	00F0F3FD	1449	HALO.	DB	000H, 0FDH, 0F5H, 0FDH, 000H
1585	00F6F6F6	1450		DB	000H, 0F6H, 0F6H, 0F6H, 000H
158A	00F0F0F0	1451		DB	000H, 0F0H, 0F0H, 0F0H, 0F0H, 0FFH
1590	C1E0E0E0	1452		DB	0C1H, 0E0H, 0E0H, 0E0H, 0C1H
1595	FFFFFFFF	1453		DB	OFFH, OFFH, OFFH, OFFH, OFFH
159A	FFFFFFFF	1454		DB	OFFH, OFFH, OFFH, OFFH, OFFH, OFFH
15A0	FFFFFFFF	1455		DB	OFFH, OFFH, OFFH, OFFH, OFFH
15A5	FFFFFFFF	1456		DB	OFFH, OFFH, OFFH, OFFH, OFFH
15AA	FFFFFFFF	1457		DB	OFFH, OFFH, OFFH, OFFH, OFFH, OFFH
15B0	FFFFFFFF	1458		DB	OFFH, OFFH, OFFH, OFFH, OFFH
15B5	FFFFFFFF	1459		DB	OFFH, OFFH, OFFH, OFFH, OFFH
15BA	FFFFFFFF	1460		DB	OFFH, OFFH, OFFH, OFFH, OFFH, OFFH
		1461			
15C0	00E0E0E0	1462	SECOND.	DB	000H, 0E0H, 0E0H, 000H, 0E0H
15C5	00D6D6D6	1463		DB	000H, 0D6H, 0D6H, 0D6H, 0E0H
15CA	09D6D6D6	1464		DB	0D9H, 0D6H, 0D6H, 0D6H, 0E0H, 0FFH
15D0	C1E0E0E0	1465		DB	0C1H, 0E0H, 0E0H, 0E0H, 0C0H
15D5	C1E0E0E0	1466		DB	0C1H, 0E0H, 0E0H, 0E0H, 0C1H
15DA	00F0F0F0	1467		DB	000H, 0FDH, 0F0H, 0F0H, 000H, 0FFH
15E0	C1E0E0E0	1468		DB	0C1H, 0E0H, 0E0H, 0E0H, 0C1H
15E5	C1E0E0E0	1469		DB	0C1H, 0E0H, 0E0H, 0E0H, 000H
15EA	FF0E0E0E	1470		DB	OFFH, 0E0H, 000H, 0E0H, 0FFH, 0FFH
15F0	00E0E0E0	1471		DB	000H, 0E0H, 0E0H, 000H, 0E0H
15F5	C1E0E0E0	1472		DB	0C1H, 0E0H, 0E0H, 0E0H, 0C1H
15FA	FFFFFFFF	1473		DB	OFFH, OFFH, OFFH, OFFH, OFFH, OFFH
		1474	ORIGEN DE DONA DE MEMORIA RAM		
	10000-	1475	ORG 0000H		
0000	100000	1476	DIRECTIF DS 0		
0000	100140	1477	ORGENA DS 00		
0010	100000	1478	END	START	

ERRORS 0  
RANGE COUNT 05

CACHENCO D80 MACRO ASSEMBLER VERSION 03.07

\*\*\* IN \*\*\*

SYMBOL	VALUE	DEFN	REFERENCES
ASCTI	088E	0100	0101
BUENO	1010	1475	1000
ORGENA	2000	1477	0141 0146 0160
CARAC	09DF	0028	0145
CHECA	1100	1001	0870
CHECAL1	11FF	1001	1004
CHECAL2	1027	1000	1004 1006
CHECAS	11EA	1010	1007
CHECAP	14FC	1008	1042
CHECAB	1000	1000	1007
CHEFLAH	101E	1009	1001

```
CHECAK      1523   1583   1584
CODIGO      09A9   0932   0940
CTRLA      0962   1326   1330 1337
CTRLB      0963   1327   1340 1342
DATOA      0960   1328   1352 1370 1375
DATOB      0961   1329   1353 1374 1379
DECO1      09AC   0925   0936
DECO2      09B4   0929   0926 0932
DECOND     1500   1462   0977
DEL        0E4F   0265   0296 0200
DIRDISF    0000   1476   0108 0258 0274 0242 0300
ENT1C      0E49   0155   0152
ENTRA      0925   0137   0075
ENTRA2     0932   0142   0144
ENTRA4     092B   0147   0157
FIN        1506   1391   1389
FINTAB     0CE1   0592   0649
IMPRIME    09EA   0924   0994
INFOR      1037   1428   0987
LOOP       0607   0670   0072
HALO      1530   1449   1388
NORMAL     0936   0196   0192
OTRA      09C1   0D43   0D46
OTRAS      0961   0267   0289
FRUDAC     0ETA   0842   0825
FRUH       0E51   0818   0841
FRUCB      0EA9   0876   0852
FRUC      0E61   0824   0086
FTTL       0900   0068
FTTL1     0608   0675   0083
FTTL2     0810   0685   0076
REP        0E66   0827   0836
REF1       09EA   0821   0826
SAL        0960   0120   0081
SALIDA     0960   0118   0112
SALIDA1    0960   0319   0520
START      0900   0060   1478
TAB1       0EAD   0916   0600 0601 0602 0603 0604 0607 0612 0613 0614 0616 0664 0665 0680
          0690
TAB10      0F09   0900   0607 0608
TAB11      0F0C   0908   0660
TAB12      0F31   0926   0666 0667
TAB13      0F0D   0971   0670 0671 0672 0673 0674
CRONOMETRO 080 MACRO ASSEMBLER VERSION 02 07
+++ IN +++
```

```
SYMBOL      VALUE      DEFN      REFERENCES
TAB14       0F09   0979   0675 0676 0677
TAB15       0F0C   0960   0690 0691 0692
TAB16       0F0E   0965   0697 0694
TAB17       0F0D   0990   0690 0696
```

TAB18	100D	1001	0697						
TAB19	102D	1007	0698 0699	0700	0701	0702			
TAB2	0E69	0914	0605 0606						
TAB20	103D	1013	0703 0704						
TAB21	105D	1019	0705 0706						
TAB22	106D	1025	0707 0708	0709					
TAB23	1091	1033	0710 0711						
TAB24	10B5	1041	0712						
TAB25	10C5	1049	0713						
TAB26	10EF	1055	0714 0715						
TAB27	1105	1061	0716						
TAB28	111D	1067	0717						
TAB2F	1135	1073	0718						
TAB3	0ECC	0918	0608 0609	0610 0611	0668	0669	0678	0679	
TAB30	114D	1079	0719						
TAB31	116D	1085	0720						
TAB32	1177	1090	0721 0722	0723					
TAB33	118F	1096	0724 0725	0726					
TAB34	11AC	1101	0727						
TAB35	11B0	1106	0728 0729	0730 0731	0732				
TAB36	11CF	1110	0733 0734	0735					
TAB37	11D9	1118	0736 0737						
TAB38	11F1	1117	0738						
TAB39	120E	1121	0739 0740						
TAB4	0E01	0921	0616 0617	0618 0619	0620 0621	0622 0623	0624 0625	0640 0646	0650
TAB40	122E	1127	0741						
TAB41	123D	1132	0742						
TAB4C	1251	1128	0743						
TAB4E	127D	1142	0744 0745						
TAB44	1299	1150	0746 0747	0748					
TAB4D	12AC	1152	0749						
TAB4C	12B1	1156	0750						
TAB47	12B0	1159	0751 0752	0753					
TAB48	12E0	1169	0754 0755						
TAB49	12C0	1161	0756 0757						
TAB5	0ED9	0925	0626 0631						
TAB50	136D	1196	0758						
TAB51	13E5	1211	0759						
TAB52	1307	1215	0760 0761						
TAB53	131F	1210	0762						
TAB54	13F7	1225	0763 0764	0765					
TAB55	143D	1230	0766 0767						
TAB56	1419	1233	0768 0769						
TAB57	1425	1236	0770 0771						
TAB58	143F	1241	0772 0773						
TAB59	144D	1244	0774 0775						
TAB6	0EE1	0928	0627 0628	0629 0630	0631	0632			
TAB60	1485	1249	0776 0777						
TAB61	147D	1254	0778 0779	0780 0781					
TAB62	149D	1259	0782 0783						

\*\*\* JH \*\*\*

SYMBOL	VALUE	DEFN	REFERENCES
TAB63	1483	1262	0784 0785
TAB64	1489	1267	0786 0787 0788 0789
TAB65	1407	1270	0790 0791
TAB7	0EED	0932	0633 0634 0635 0636 0637 0641 0642
TAB8	0F01	0937	0638 0639 0640 0647 0648 0649
TAB9	0F15	0942	0642 0644 0652 0653 0654 0655 0656 0659 0660 0661 0662 0666 0687
TABLAS	0CE1	0599	0664
TCC	09E1	0480	0820 0839
TECLA	036C	0296	0276 0304 0317
TECLA1	0936	0508	0D16
TIFL5	0679	0187	0182
TIFOH	0650	0170	0167
TIFOL	0668	0177	0173
TIFOS	067D	0191	0178

#### 4.5 EQUIPO DE PRUEBA UTILIZADO.

El desarrollo e implementación de estos programas fue elaborado y probado en un equipo de desarrollo ensamblado en México con partes fabricadas por Cromenco Inc., el cual está equipado con un sistema de bus S-100, que es un conector con 100 terminales, para conectar él, y en paralelo, las tarjetas de circuito impreso son diferentes componentes para realizar las diferentes funciones de una microcomputadora.

El bus S-100 agrupa diferentes señales de mando, las cuales pueden ser agrupadas en 4 categorías: señales de alimentación, señales de datos, señales de dirección y señales de reloj y control. Para más información consultar la referencia manual ZPU Z70-CPU.

Para establecer la comunicación con la microcomputadora, se utilizarán una consola video-teclado, marca Televidco Systems, Inc., modelo TVI-920C, una unidad de discos flexibles (floppy disk), marca Cromenco, modelo PFD, para dos discos de 8" (20.3 cm) y una impresora de línea marca Centronics, modelo 779.

La microcomputadora utilizada aloja actualmente 5 tarjetas de circuitos fabricados por Cromenco, todos compatibles con el bus S-100 del gabinete principal. A continuación se da una breve reseña de dichas tarjetas.

CPU (Cromenco ZPU).- Esta tarjeta utiliza el microprocesador Z-80a, el cual opera confiablemente con un reloj de hasta 4 Mhz. La tarjeta ZPU ofrece la capacidad de salto automático al encendido a cualquiera de 16 localidades de memoria, seleccionadas por interruptores, el funcionamiento a 2 Mhz o a 4 Mhz de frecuencia, un generador de estados de espera para igualar la velocidad del microprocesador al tiempo de acceso de dispositivos de entrada externos relativamente más lentos, circuitería para el monitoreo de dirección y otras ventajas como la superioridad de programación y el refrescado de RAM dinámica sin adición de componentes.

MEMORIAS (Cromenco 64 KZ).- La tarjeta 64 KZ contiene 65,536 Byte (64 kByte) de memoria de lectura/escritura dinámica (RAM dinámica), con un tiempo máximo de acceso real de 250 nsegs. Esto la hace operable a la velocidad de 4 Mhz del Z-80A sin necesidad de tiempos de espera.

El diseño de esta tarjeta permite una expansión de memoria de --- hasta 512 kByte, en bloques de 32 kByte, seleccionables por medio de -- interruptores. La selección de los bloques también se puede efectuar -- por programa, teniendo también la facilidad de habilitar o deshabilitar automáticamente después de un "reset" del sistema.

PUERTOS ENTRADA/SALIDA (Cromenco D+7A I/O).- Esta tarjeta da una forma fácil y rápida de interfazar analógica y digitalmente a la CPU. - Este módulo tiene 7 canales de conversión analógica-digital de 8 bits y siete de conversión digital-analógica, con un tiempo de conversión 5.5 microsegundos.

Adicionalmente, la tarjeta tiene un puerto digital paralelo de -- salida y uno de entrada, ambos de 8 bit. Así, hay 8 puertos E/S en la - tarjeta, cuya dirección de acceso es seleccionada por 5 conectores in--tercambiables (jumpers) en la misma tarjeta.

CONTROLADOR DE DISCO (Cromenco 4FDC).- El controlador de disco --- Cromenco está diseñado para interfazar unidades de discos de 5" u 8" a la microcomputadora a través del bus S-100, con el fin de almacenar y - recobrar programas extensos en discos magnéticos. Adicionalmente, la -- 4FDC contiene un canal de E/S serie, con velocidades de comunicación -- seleccionables por programa desde 110 hasta 76,800 Baud. Este canal se--rie se emplea generalmente para la comunicación entre la consola video--teclado y la CPU. El controlador 4 FDC contiene también un kByte de ROM programada con el sistema residente de operador de discos (RDOS).

El RDOS permite al operador realizar todas las operaciones de --- disco esenciales desde el teclado de la consola. También posee un pro--grama para cargar en memoria el sistema operativo de disco Cromenco --- (CDOS) desde un disco.

INTERFACE CON IMPRESORAS (Cromenco PRI).- La interface paralelo - para impresoras, Cromenco PRI, permite la operación simultánea de una - impresora de matriz de puntos o una impresora con calidad de imprenta.

La dirección de acceso a este puerto de salida puede ser selec--- cionada por conexiones de alambre en la misma tarjeta.

SISTEMA OPERATIVO (CDOS).- El sistema operativo de discos Cromen--co (CDOS) es el programa que mantiene activo el sistema en conjunto.



La función de CLOS es mantener comunicación de E/S con la unidad de discos, para poder crear y leer archivos, con la consola, para escribir (programar) y leer, y con la impresora. Esto lo efectúa CLOS con llamadas a subprogramas por medio de nombres simbólicos que introduce el programador desde la consola. CLOS ofrece también la facilidad de poder llamar desde un programa, a subrutinas que posee, como multiplicación, división, obtención de caracteres desde el teclado e impresión de caracteres en la pantalla de la consola.

## CONCLUSIONES

Dar solución a los problemas es una tarea primordial del hombre, para su desarrollo mismo y el de la sociedad, a medida que da solución a éstas da un paso más al avance económico y tecnológico, generando con esto nuevas fuentes de trabajo.

Hoy en día el sorprendente auge que ha tomado la electrónica digital ha sido de gran importancia para el desarrollo de grandes potencias, de tal forma que aparatos o dispositivos que antes ocupaban grandes dimensiones de espacio han sido miniaturizados con la aparición de una gran cantidad de computadoras, las cuales tienen acceso en todas las actividades desarrolladas por el ser humano.

El desarrollo del diseño del Probador de Circuitos Integrados Digitales, viene a ser parte de una mínima parte de los conocimientos adquiridos a lo largo de la carrera, y digo mínima, ya que prácticamente es un área la que cubre el diseño del sistema ya descrito y es la que corresponde al Laboratorio de Electrónica, la cual en opinión muy personal, hoy por hoy, es uno de los mejores laboratorios a nivel académico-práctico en lo que se refiere a la carrera. Dando como resultado que los estudiantes salgan con un nivel acorde a las necesidades del país y de la industria y además comparado con el de otras universidades o institutos. Si bien es cierto que no todo es perfecto, aquí al menos se cumplen con los objetivos.

Es claro el ejemplo, que con su empeño y trabajo este grupo de académicos, tratan de dar una imagen a esta facultad a través de la carrera de Ingeniería Mecánica Eléctrica, como una institución que cumple con los programas requeridos para el avance tecnológico de la industria mexicana.

Si al igual que ésta, todas las demás áreas trataran de establecer los parámetros necesarios para la superación misma, hoy la carrera podría ser comparada con la de institutos de gran prestigio en el país.

Todas estas deficiencias encontradas, traen como consecuencia que

los trabajos de tesis sean tomadas por la gran mayoría de los egresados de las distintas carreras como un mero trámite para obtener el título y por consiguiente que muchas de éstas sean copias originales de otras ya realizadas.

Algo completamente diferente, es el concepto que tengo de las --- mismas, ya que a mi me permitió reafirmar conocimientos, que por una -- razón u otra las había olvidado y las cuales son de gran importancia, - además me permitió conocer temas nuevos durante y para el desarrollo -- del sistema. Por otra parte en mi estancia en el laboratorio aprendí de temas nuevos en lo que se refiere a la electrónica digital y otras ma-- terias, trayendo como consecuencia estar más actualizado y además se--- quir los ejemplos que de muchos de ellos aprendí.

Como es natural y como seres humanos que somos, estamos sujetos a cometer errores en cualquier trabajo que realicemos, por lo que se --- agradece al lector todas las criticas que se hagan al respecto, para el perfeccionamiento de la misma al buen desarrollo.

## APENDICE A

### MANUAL DEL USUARIO

#### A.1 INTRODUCCION.

El Probador de Circuitos Integrados Digitales ha sido diseñado -- como su nombre lo dice, para probar un cierto número de CIs digitales - TTL, de los cuales al final de este apéndice se da un listado. La se-- lección de los circuitos integrados a probar se hizo en base a las ne-- cesidades propias del Laboratorio de Electrónica, es decir, los más --- utilizados en la elaboración de prácticas y en otras aplicaciones como son: pruebas de circuitos realizados por alumnos de tesis, diseño de -- profesores, etc. Evitando con esto la pérdida de tiempo en la elabora-- ción de las mismas; permitiendo además tener más control sobre el mane-- jo de estos considerando que su costo es alto y el número de alumnos -- también.

#### A.2 DESCRIPCION DEL EQUIPO.

El Probador de Circuitos Integrados Digitales se implementó en un pequeño gabinete como el que se muestra en la Fig. A.1.

Como se puede observar consta de un panel de 12 exhibidores del - tipo matriz de LEDs 5X7 (2), teclado (3), interruptores de apagado/en-- cendido e inicialización (1) y 7 bases comprendidas en dos grupos (4 de 14 patas y 3 de 16). Cada uno tiene una función específica dentro del - Probador como a continuación se menciona.

- 1) Interruptores de apagado/encendido y de inicialización.- El -- primero es el que se encarga de energizar al sistema y el se--

gundo es el que da la señal a todo el sistema para inicializar su operación.

- 2) Panel de exhibidores luminosos (matriz de LEDs 5X7).- Son donde se muestran los números de los CIs a probar, así como los resultados de los mismos al realizarse la prueba. En cada matriz de LEDs deberá aparecer un número o letra, esto es, cuando el usuario pruebe algún circuito deberá meter el número del CI. Una vez que lo ha desplegado estaremos listos para recibir el mensaje de acuerdo al estado en que se encuentre.
- 3) Teclado.- Por medio de este, se introducen los números de CIs a probar así como los comandos necesarios para su ejecución.
- 4) Bases de prueba.- Aquí es donde se insertan los CIs a probar, observando que se encuentran clasificados en dos grupos; en el primer grupo de la derecha se encuentran cuatro y corresponden a los CIs de 14 patas, ordenados de acuerdo a su alimentación; en el segundo grupo a la izquierda se encuentran tres y corresponden a los CIs de 16 patas, ordenados también de acuerdo a su alimentación.

### A.3 MODO DE OPERACION DEL SISTEMA.

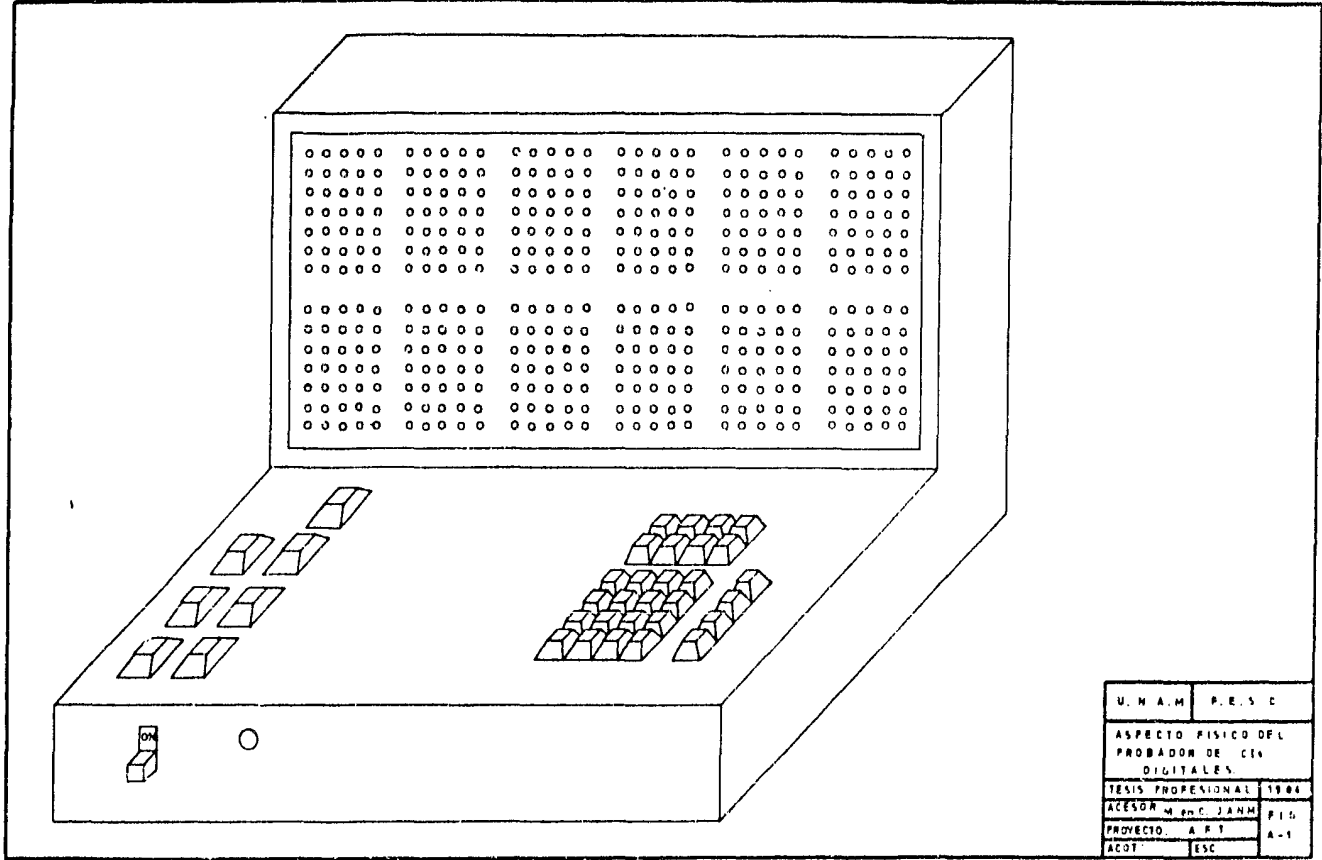
En esta parte se da a conocer al usuario la manera de operar al Probador de CIs Digitales.

i) Primeramente el usuario deberá tener el cuidado de alimentar al sistema a través de la línea y enseguida energizar al sistema por medio del interruptor de apagado/encendido.

ii) Como siguiente paso, inicializar el sistema originando la señal para su operación.

iii) En tercer término insertar el CI en la base que le corresponda de acuerdo a su alimentación y número de patas. Se recomienda ver Fig. A.2 para no incurrir en errores, en donde se muestra únicamente la posición de las bases de acuerdo a lo ya mencionado, esto se hace para mayor ayuda al usuario, recordando que éstas se pueden ver también en la Fig. A.1.

Es importante también señalar, que en el listado que al final de este apéndice se da de los CIs a probar, se incluye el número de base -



U. N. A. M.	P. E. S. C.
ASPECTO FISICO DEL PROBADOR DE C.I. DIGITALES.	
TESIS PROFESIONAL 1984	
ACCESOR M. P. C. JANM	FIG.
PROYECTO. A. P. 7	A-1
ACOT.	ESC.

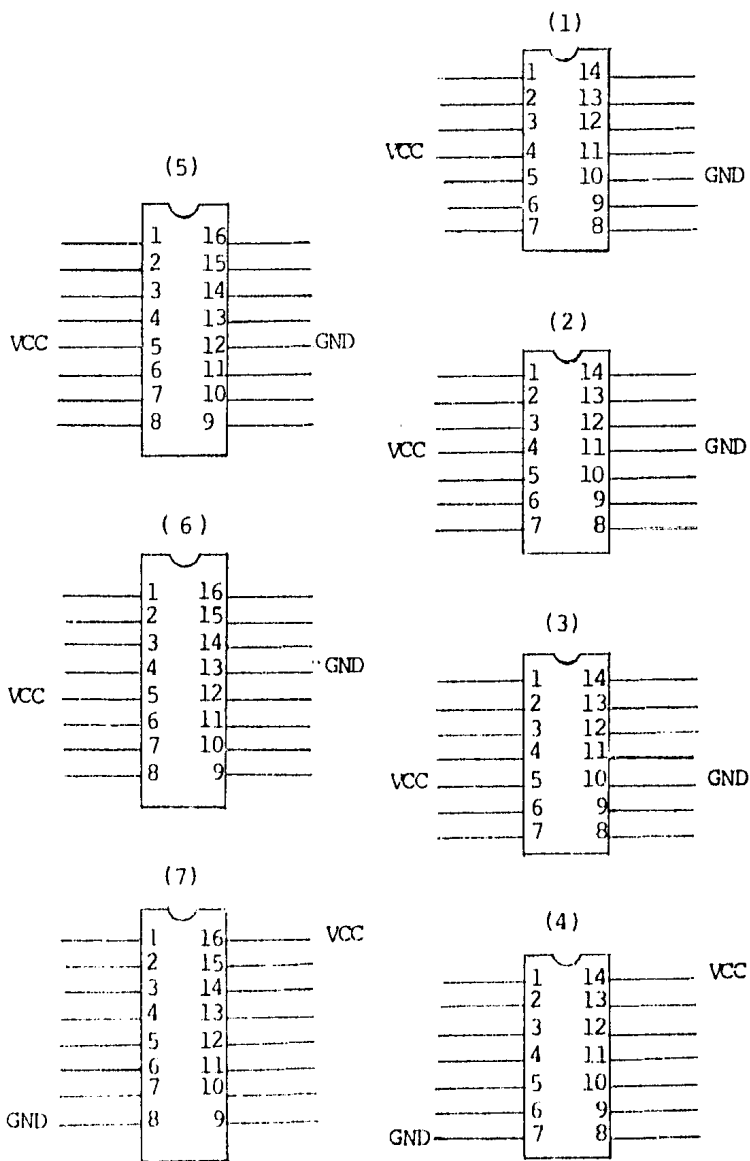


Figura A.2. Posición de las bases de prueba del Probador de acuerdo a su alimentación y número de patas.

que le corresponde a cada uno.

iv) El siguiente paso es meter la información necesaria para la prueba del CI, es decir, el número del CI a probar por medio del teclado; oprimiendo únicamente el número en la tecla correspondiente hasta completar el deseado, siendo estos desplegados cada vez que se mete una letra o número. Una vez metido esta información se oprimira la tecla -- que se encuentra en el extremo superior derecho denominada por "CR" --- (ver Fig. A.3). Al mismo tiempo se debe poner atención en el panel de -- exhibidores para visualizar si la información que esta metiendo es la -- correcta, de lo contrario el sistema le contestara como desconocido y -- por consiguiente deberá corregir comenzando de nueva cuenta su ejecu--- ción, la cual una vez realizada nos mostrara el resultado obtenido en -- la prueba, con los siguientes mensajes:

a) BUENO.- Si el CI no se encuentra averiado.

b) MALO.- Si éste se encuentra averiado.

c) DESCONOCIDO.- Si el CI que se desea probar no se encuentra en el listado de los CIs a probar conocidos.

Como se podrá observar el método de operación no es difícil, solo se pide al usuario seguir los pasos correctos, evitando con esto ave--- riar al sistema y por lo tanto ahorro de tiempo en la prueba.

A continuación se da la tabla citada anteriormente.

Tabla A.1 Listado general de CIs a probar en orden numérico y -- número de base correspondiente.

NUMERO	No. BASE	DESCRIPCION
7400	4	
74H00	4	
74L00	4	COMPUERTA NAND CUADRUPLA DE DOS ENTRADAS
74LS00	4	
74S00	4	
7401	4	COMPUERTA NAND CUADRUPLA DE DOS ENTRADAS, CO---
74LS01	4	LECTOR ABIEPTO
74H01	4	COMPUERTA NAND CUADRUPLA DE DOS ENTRADAS, CO---



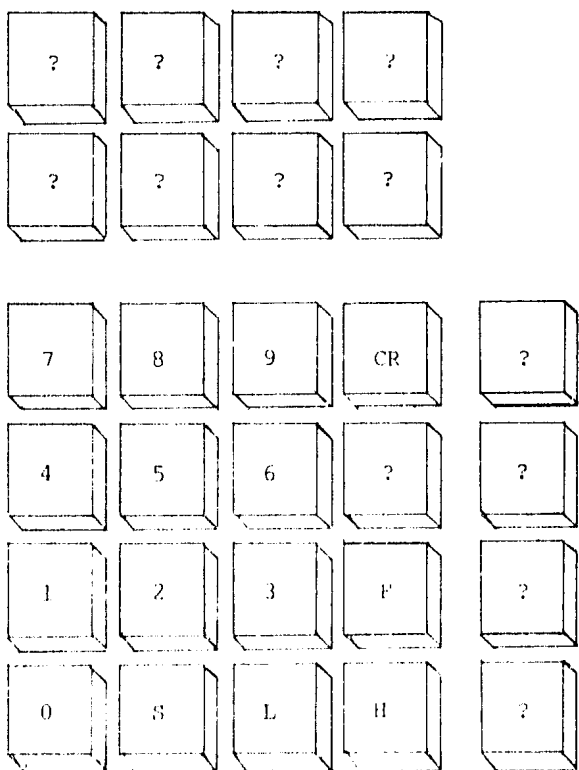


Figura A.s. Teclado del Probador de CIs Digitales.

		LECTOR ABIERTO.
7402	4	
74L02	4	NOR CUADRUPLE DE DOS ENTRADAS.
74LS02	4	
74S02	4	
7403	4	
74L03	4	NAND CUADRUPLE DE DOS ENTRADAS, COL. AB.
74LS03	4	
74S03	4	
7404	4	
74H04	4	
74L04	4	INVERSOR SEXTUPLO
74LS04	4	
74S04	4	
7405	4	
74H05	4	INVERSOR SEXTUPLO, COL. AB.
74LS05	4	
74S05	4	
7406	4	
7406	4	INVERSOR SEXTUPLO BUFFER/DRIVER, COL. AB.
7407	4	BUFFER/DRIVER SEXTUPLO ALTO VOLTAJE, COL. AB.
7408	4	
74LS08	4	AND CUADRUPLE DOS ENTRADAS.
74S08	4	
7409	4	
74LS09	4	AND CUADRUPLE DOS ENTRADAS, COL. AB.
74S09	4	
7410	4	
74H10	4	
74L10	4	NAND TRIPLE TRES ENTRADAS.
74LS10	4	
74S10	4	
74H11	4	
74LS11	4	AND TRIPLE TRES ENTRADAS.
74S11	4	

7412	4	NAND TRIPLE DE TRES ENTRADAS, COL. AB.
74LS12	4	
7413	4	DOBLE SCHMITT TRIGGER CUATRO ENTRADAS NAND.
74LS13	4	
7414	4	INVERSOR SEXTUPLO SCHMITT TRIGGER.
74LS14	4	
74H15	4	
74LS15	4	AND TRIPLE TRES ENTRADAS, COL. AB.
74S15	4	
7416	4	INVERSOR SEXTUPLO BUFFER/DRIVER, COL. AB.
7417	4	BUFFER/DRIVER SEXTUPLO ALTO VOLTAJE, COL. AB.
7420	4	
74H20	4	
74L20	4	NAND DOBLE CUATRO ENTRADAS.
74LS20	4	
74S20	4	
74H21	4	AND DOBLE CUATRO ENTRADAS.
74LS21	4	
7422	4	
74H22	4	NAND DOBLE DE CUATRO ENTRADAS, COL. AB.
74LS22	4	
74S22	4	
7425	4	NOR DOBLE DE CUATRO ENTRADAS Y ESTROBE.
7426	4	NAND CUADRUPLE DOS ENTRADAS E INTERFAZ ALTO ---
74LS26	4	VOLTAJE, COL. AB.
7427	4	NOR TRIPLE DE TRES ENTRADAS.
74LS27	4	
7428	4	BUFFER NOR CUADRUPLE DOS ENTRADAS.
74LS28	4	
7430	4	
74H30	4	
74L30	4	NAND OCHO ENTRADAS.
74LS30	4	
74S30	4	
7432	4	

74LS32	4	OR CUADRUPLE DOS ENTRADAS.
74S32	4	
7433	4	BUFFER NOR CUADRUPLE DOS ENTRADAS, COL. AB.
74LS33	4	
7437	4	
74LS37	4	BUFFER CUADRUPLE NAND DOS ENTRADAS.
74S37	4	
7438	4	
74LS38	4	BUFFER NAND CUADRUPLE DOS ENTRADAS, COL. AB.
7440	4	
74H40	4	BUFFER NAND DOBLE CUATRO ENTRADAS.
74LS40	4	
74S40	4	
7442	7	
74L42	7	DECODIFICADOR BCD A DECIMAL.
74LS42	7	
7443	7	DECODIFICADOR 4 A 10 LINEAS (EXCESO 3 A DECIMAL)
74L43	7	
7444	7	DECODIFICADOR EXCESO 3 GRAY A DECIMAL.
74L44	7	
7445	7	DECODIFICADOR/DRIVER BCD A DECIMAL.
7446	7	DECODIFICADOR/DRIVER 7 SEGMENTOS.
74L46	7	
7447	7	DECODIFICADOR/DRIVER 7 SEGMENTOS.
74L47	7	
74LS47	7	
7448	7	DECODIFICADOR BCD A 7 SEGMENTOS PULL-UP.
74LS48	7	
7449	4	DECODIFICADOR/DRIVER BCD A 7 SEGMENTOS, COL. AB.
74LS49	4	
7451	4	
74H51	4	COMPUERTAS AND Y NOR.
74S51	4	
74L51	4	COMPUERTAS AND Y NOR.
74LS51	4	

7454	4	CUADRUPLE AND-OR INVERSOR DE DOS ENTRADAS.
74H54	4	CUADRUPLE AND-OR INVERSOR.
74L54	4	CUADRUPLE AND-OR INVERSOR.
74LS54	4	
74S64	4	COMPUERTAS AND-OR-INVERSOR SALIDA TOTEM-POLE.
74S65	4	COMPUERTAS AND-OR-INVERSOR SALIDA COL. AB.
7470	4	FLIP-FLOP CON PRESET Y CLEAR JK.
74H71	4	FLIP-FLOP JK MAESTRO ESCLAVO CON PRESET.
74L71	4	FLIP-FLOP JK MAESTRO ESCLAVO CON PRESET Y CLEAR.
7472	4	
74H72	4	FLIP-FLOP JK MAESTRO ESCLAVO.
74L72	4	
7473	2	
74H73	2	FLIP-FLOP JK MAESTRO ESCLAVO.
74L73	2	
74LS73	2	
7474	4	
74H74	4	
74L74	4	FLIP-FLOP D DOBLE.
74LS74	4	
74S74	4	
7475	5	
74L75	5	LATCH DE CUATRO BITS.
74LS75	5	
7476	6	
74H76	6	FLIP-FLOP JK DOBLE.
74LS76	6	
74H78	4	FLIP-FLOP JK DOBLE CON PRESET, CLEAR Y CLOCK -- COMUN.
74L78	2	FLIP-FLOP JK DOBLE CON PRESET, CLEAR Y CLOCK -- COMUN.
74LS78	2	COMUN.
7480	4	SUMADOR COMPLETO.
7482	2	SUMADOR COMPLETO DE DOS BITS.
7483	5	SUMADOR COMPLETO DE CUATRO BITS.
74LS83	5	

7486	4	
74LS86	4	OR EXCLUSIVA DE DOS ENTRADAS.
74S86	4	
74L86	4	
74H87	4	COMPLEMENTO/VERDADERO 4 BIT.
7490	3	
74L90	3	CONTADOR DE DECADAS.
74LS90	3	
7492	3	CONTADOR DIVISOR POR 12.
74LS92	3	
7493	3	CONTADOR BINARIO DE 4 BITS.
74LS93	3	
74L93	2	
7494	5	SHIFT REGISTER DE 4 BITS.
7495	4	SHIFT REGISTER DE 4 BITS DERECHO-IZQUIERDO.
74LS95	4	
74L95	2	
7496	5	
74L96	5	SHIFT REGISTER DE 4 BITS.
74LS96	5	
74125	4	BUFFER CUADRUPLE CON SALIDA DE 3 ESTADOS.
74LS125	4	
74126	4	BUFFER CUADRUPLE CON SALIDA DE 3 ESTADOS.
74LS126	4	
74LS138	7	DECODIFICADOR/DEMÚLTIPLEXOR DE 3-8 LINEAS.
74S138	7	
74LS139	7	DOBLE DECODIFICADOR/DEMÚLTIPLEXOR DE 2-4 LI-----
74S139	7	NEAS.
74147	7	CODIFICADOR BCD 10 LINEAS A 4.
74LS147	7	
74148	7	CODIFICADOR 8-3 OCTAL.
74LS148	7	
74153	7	
74LS153	7	SELECTOR/MÚLTIPLEXOR DOBLE DE 4 A 1.
74LS153	7	

74S153	7	
74155	7	DECODIFICADOR/DEMULTIPLEXOR DOBLE SALIDA TOTEM-
74LS155	7	POLE.
74156	7	DECODIFICADOR/DEMULTIPLEXOR DOBLE COL. AB.
74LS156	7	
74157	7	
74L157	7	SELECTOR/MULTIPLEXOR CUADRUPLE 2 A 1.
74LS157	7	
74S157	7	
74LS158	7	SELECTOR/MULTIPLEXOR CUADRUPLE 2 A 1.
74S158	7	

Cabe mencionar que el Probador de Circuitos Integrados Digitales no está limitado a probar únicamente los mencionados, sino que el número puede ir aumentando en relación a las necesidades propias del laboratorio sin necesidad de cambiar el programa, lo único que debe hacer el usuario es meter las tablas de prueba de los CIs (ver configuración que se da en el apartado 3.3.1) y por consiguiente deberá cambiar las tablas de direcciones y las de circuitos conocidos.

Es de suma importancia, dar a conocer la forma de como el usuario puede aumentar el No. de CIs a probar; por lo que a continuación se dan los pasos a seguir.

a).- Verificar si el CI que se desea integrar a la lista de CIs conocidos realiza la misma función que alguno de estos; si es igual, no se realizará la tabla de prueba (esto es de acuerdo a los pasos mencionados en el apartado 3.3.2). Por ejemplo, si se quisiera meter el 74425, nos daríamos cuenta que este CI realiza la misma función que el CI --- 74125. En caso de que el CI no cumpla con esto se tendrá que realizar la tabla de prueba.

b).- Una vez que se ha realizado lo ya mencionado se hará lo siguiente: Si el CI a meter cumple con lo primero mencionado en el inciso a, sólo tendrá que meter el No. del CI en la tabla de circuitos integrados conocidos considerando lo siguiente:

CI ESTANDAR = 0

" H = 1  
 " L = 2  
 " LS = 3  
 " S = 4

Considerando lo mencionado el CI 74425 quedaría como DB 'Ø425'.

Estos circuitos siguen el orden numérico, por lo que debiera de -- tener cuidado en meterlo en su lugar correspondiente.

Al igual que en la tabla de CIs conocidos, en la tabla de direc-- ciones de prueba se meta el número de tabla que le corresponde, para el caso anterior le corresponde la tabla 55 (DW TAB55), cuidando que el mismo lugar que le correspondió en la tabla de CIs conocidos sea el que le corresponda en la tabla de direcciones de prueba.

Cuando el CI que se desea integrar no se parece a ninguno de los ya contenidos en cuando a función se generará la tabla de prueba, co--- rrespondiéndole para este caso; la tabla No. 66 y así sucesivamente, -- anotando obviamente el No. de CI (DB ' ' y DW TAB---) en las ta- blas y orden correspondiente.

Para realizar las tablas el procedimiento es el siguiente:

Para el CI 74LS266

PUERTO B								PUERTO B								
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	E/S del PIO
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Patas del CI a probar.
1	1	1	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Palabra de control para --- mar el PIO.
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Condiciones iniciales.
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	Ø	Ø	Número de pruebas.
1	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	
1	1	1	Ø	1	1	1	Ø	1	Ø	Ø	1	1	1	Ø	1	Prueba.
1	1	1	1	Ø	1	1	1	Ø	Ø	1	Ø	1	1	1	Ø	
1	1	1	1	1	Ø	Ø	1	1	Ø	1	1	Ø	Ø	1	1	



TABLA EN HEXADECIMAL:

CONTROL PUERTO A  
 CONTROL PUERTO B

0C
E6
00
04
00
E0
9D
EE
2E
F7
B3
F9

Programando el PIO.

No. de condiciones iniciales.

No. de pruebas.

DATOS    PUERTO A  
 "        "    B  
 "        "    A  
 "        "    B  
 "        "    A  
 "        "    B  
 "        "    A  
 "        "    B

PRUEBA

APENDICE B.

REFERENCIAS BIBLIOGRAFICAS.

- 1 -García Narcia, Octavio F.  
"Microprocesadores- Z80 e Interfaces"  
ESIME, IPN; México, 1981.
- 2 -García Narcia, Octavio F.  
"Programación Z80"  
EXIME, IPN (2a Edición); México, 1982.
- 3 -Levine, Morris E.  
"Digital Theory and Practice Using Integrated  
Circuits"  
Prentice-Hall, Inc.; New Jersey, 1978.
- 4 Mandado, Enrique.  
"Sistemas Electrónicos Digitales"  
Ed. Marcombo (4a. Edición), Barcelona 1983.
- 5 -Mostek, Corp.  
"MK 3880 & MK 3880-4 (Z80-CPU & Z80A-CPU) Técnica Manual"  
Mostek, Corp.; Carrollton, Texas, 1977.
- 6 -Nichols, Joseph C.  
-Nichols, Elizabeth A.  
-Rony, Peter R.  
"Interfacing Digital Circuits with the Z-80 CPU, PIO, and CTC --  
Chips"
- 7 -Peatman, John P.  
"Computer-Based Design"  
McGraw-Hill Book Co.; New York, 1981.
- 8 -S Systems.  
"Z80 Starter Kit Operacional Manual"  
SD Systems; Dallas, Texas, 1978.
- 9 -Sol Libes and Mark Garetz.

"Intefacing To S-100/IEEE 696 Microcomputers"

Osborne/McGraw-Hill, California, 1981

10 - Taub, Herbert.

"Circuitos Digitales y Microprocesadores"

McGraw-Hill,

11 -Texas Instruments, Inc.

"The TTL Data Book For Design Engineers"

Texas Instruments, Inc.; U.S.A., 1981.

12 -Zilog, Inc.

"Z80-PIO & Z80A-PIO Technical Manual"

Zilog, Inc.; Cupertino, California, 1977.