

00362
3
2 ej



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE CIENCIAS**

**INSTRUMENTO PROGRAMABLE PARA PROBAR
DIFERENTES CIRCUITOS INTEGRADOS DIGITALES**

T E S I S
QUE PARA OBTENER EL GRADO DE :
Maestro en Ciencias (FISICA)
P R E S E N T A :
JORDI IÑAKI AUSTRICH SENDSIAN

**TESIS CON
FALLA DE ORIGEN**



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.

1.- INTRODUCCION.

Primero haré una pequeña reseña de cómo surgió la idea de construir un instrumento que tuviese las características necesarias y suficientes para realizar las pruebas pertinentes al Circuito Integrado Digital (CID) a probar. Luego, una idea muy general del por qué construir dos versiones del instrumento, lo cual quedará aclarado en el transcurso del desarrollo de esta tesis. Y, por último, una definición de cada una de las dos versiones y de utilizarlas, también, como material didáctico.

La idea de crear un instrumento que fuese capaz de poder probar diferentes circuitos integrados digitales, surgió a finales de los años 70's por medio de la M. en C. Gertrudis Kurz de Delara y del Ingeniero Mario Levin. Con el tiempo esta idea fué madurando poco a poco y se realizaron diferentes publicaciones al respecto (Ver Referencias). Tenían en mente el utilizar una microcomputadora como un instrumento capaz de poder llevar al cabo las pruebas pertinentes a diferentes circuitos integrados digitales, realizando ciertas adaptaciones a la microcomputadora para lograr el propósito de las pruebas.

Entre 1980 y 1983, el ingeniero Héctor Guerrero construyó un primer prototipo de lo que sería el instrumento. Este prototipo cumplía con las exigencias de poder probar diferentes circuitos integrados digitales, pero tenía un pequeño inconveniente: había que hacer muchas conexiones externas, algo parecido a un conmutador telefónico antiguo, y así poder conectar la fuente a cada circuito integrado digital y, asimismo, para las señales que son de entrada, las de salida y algunas otras, de acuerdo al circuito integrado digital que se fuera a probar.

En 1985 la maestra Kurz (entonces coordinadora del Laboratorio de Cibernética " Alejandro Medina ", q.e.p.d.), me propuso como tesis de maestría el que desarrollara y construyese este instrumento, contemplando la factibilidad de que no tuviese tanto conexionamiento externo. Viendo que sí era posible hacerlo, vislumbramos la posibilidad de crear dos instrumentos, digámoslo así, en uno sólo; de ésto surgió el nombre de Instrumento Programable/Programado, esto es, tendríamos dos versiones de dicho instrumento: El Instrumento Programable y el Instrumento Programado.

Por Instrumento Programable entenderemos a un dispositivo que debe ser programado por el usuario, para que éste le indique al instrumento cómo se realizará el conexionamiento apropiado para poder llevar al cabo la o las pruebas pertinentes al circuito integrado digital, esto es: el usuario le "dice" al instrumento que patitas (o terminales) se conectarán al Vcc y al GND, cuáles son entradas y cuáles son salidas y, también, que terminales son de selección, de control y de direccionamiento, si procede, para que el propio Instrumento Programable se encargue de llevar al

cabo dichos "conexionamientos". Una vez hecho ésto, el usuario le indicará al instrumento que realice la prueba (o pruebas, según sea el caso). Terminada la prueba, el instrumento desplegará en pantalla los resultados y el usuario, y sólo el usuario, será quien determine si el circuito integrado digital probado está en condiciones de buen funcionamiento o no. De aquí se desprende que el 'Programable' está pensado para ser utilizado por personas con amplios conocimientos en electrónica digital.

Definimos por Instrumento Programado al dispositivo que con sólo darle la nomenclatura apropiada del circuito integrado digital a probar, verifique si tiene o no el programa adecuado para realizar la o las pruebas a dicho circuito y, según sea el caso, proceda a probar o indique al usuario que no puede realizar la prueba. Esta verificación se hará automáticamente y los resultados se darán a través de un monitor o de una impresora, o de ambos, según lo requiera el usuario. Este dispositivo está pensado para que pueda ser utilizado, prácticamente, por cualquier persona.

Lo anterior ha sido posible gracias al gran avance tecnológico que ha sufrido en poco tiempo la industria electrónica digital (asimismo la electrónica analógica, pero esto va más allá de los fines de esta tesis), y que nos permite enfrentarnos con un mercado en el que hay una cantidad muy grande de diferentes circuitos integrados digitales, con los cuales se nos permite desarrollar, diseñar e implementar una serie de dispositivos electrónicos que son capaces de realizar diversas funciones, o funciones muy particulares, dependiendo de la aplicación que se requiera. Dentro de las funciones particulares tenemos, por ejemplo, a los controladores, como pueden ser para los semáforos, lavadoras, etc.; otros dispositivos son los detectores electrónicos que tienen algunas marcas de automóviles para indicar si están o no cerradas las puertas, o si se han dejado puestas las llaves en el sistema de ignición al momento de abrir una puerta, etc. Y como último ejemplo podemos citar al instrumento per se dentro del mundo de la electrónica: el osciloscopio, del cual hemos podido presenciar la transformación que ha habido del osciloscopio analógico al osciloscopio digital. No queriendo decir con esto que el osciloscopio analógico ya no se use, sino todo lo contrario y, además, nos está mostrando la gran importancia que están tomando los microprocesadores de 8 bits para el desarrollo e implementación de instrumentos programados/programables. Dentro de poco tendremos, también, instrumentos inteligentes utilizando estos microprocesadores (o microprocesadores más grandes, dependiendo del dispositivo que se realice).

Entre los dispositivos electrónicos que realizan muy diversas funciones tenemos a las bien conocidas computadoras, tan en boga en estos días, y que gracias a su gran versatilidad, se han podido utilizar en diferentes ramas, que cubren desde la escolaridad primaria hasta los vuelos espaciales, cumpliendo con sus funciones características o con funciones predeterminadas, dependiendo del uso que se les ha querido dar. Afortunadamente, ésto ha abierto ampliamente las puertas al desarrollo de una de estas ramas,

sumamente importante y que dentro de poco tiempo será tan fácil de utilizar como un destornillador, que es la instrumentación programable y, en nuestro caso, a la creación de un instrumento de laboratorio capaz de poder probar diferentes circuitos integrados digitales de tecnología TTL, incluyendo a RAM's estáticos, EPROM's y PIA's, cuyas características generales y particulares son tema de esta tesis, que se explicarán en los capítulos siguientes.

A continuación se dan los pormenores del por qué se creó tal dispositivo.

En términos generales, la idea para desarrollar un instrumento Programable/Programado es la de poder tener un equipo auxiliar para probar diferentes circuitos integrados digitales.

Hay varias razones para diseñar y construir un instrumento capaz de probar diferentes circuitos integrados digitales, a saber:

- a) No hay en el mercado un equipo que permita hacer pruebas a un sólo circuito integrado digital a la vez.
- b) El problema de tener que armar el conexiónamiento apropiado para llevar a cabo las pruebas a un solo circuito integrado digital. Este problema se acentúa mucho más cuando se tienen que probar varios circuitos integrados digitales diferentes.

El punto b) es bastante obvio y no requiere de mayor explicación.

El punto a) es donde se puede argüir que existen equipos tales como Analizadores de Estados Lógicos (Logic Analyzers) o Troubleshooters, etc. Para esto, diremos que:

- 1º) Los analizadores de estados lógicos (Logic Analyzers) son muy buenos y eficaces para llevar a cabo el rastreo de señales en algún equipo digital que se haya descompuesto y también son muy útiles para detectar fallas en sistemas completos que han estado funcionando, pero no para probar individualmente a cada circuito integrado digital, amén de que su precio es sumamente alto.
- 2º) Otros dispositivos, como los Troubleshooters, utilizan la técnica de la comparación, la cual no es muy fiable.
- 3º) Otra técnica es la de sustitución, la cual no es muy recomendable (pero efectiva en algunos casos, sobre todo cuando se trata de fallas en los microprocesadores).
- 4º) También se creó este instrumento para los requerimientos que tenía el Laboratorio de Cibernética *Alejandro Medina*, de la Facultad de Ciencias, U.N.A.M., y que muy posiblemente se tengan, no sólo en otros laboratorios de electrónica sino

también en talleres y la industria.

Estas son las razones por las cuales se ha decidido el diseño y la construcción de un instrumento que sea capaz, no solo de poder probar diferentes circuitos integrados digitales sino que, además, pueda diagnosticar si dicho circuito integrado digital está funcionando correctamente o no e indicar en que patita, o terminal, de salida del circuito no se está obteniendo la señal requerida; todo esto es lo que veremos en el transcurso de la descripción de todo lo que puede y no puede hacer este instrumento y, además, con la característica de que se puedan probar los circuitos integrados digitales en forma individual y con la certeza de que la o las pruebas que se hagan sobre cada uno de estos circuitos, sean satisfactorias para el físico, el ingeniero o el técnico en electrónica digital.

Esta tesis desarrollará el tema para el Instrumento Programable. Para el Instrumento Programado, consultar la tesis de licenciatura que dirigió al Sr. Roberto Serna Herrera con título de 'Una Aplicación de la Computación a la Instrumentación Programable: Analizador de Memorias Digitales'. (12).

Conforme se vaya desarrollando esta tesis, se podrá ver que el instrumento se ha hecho también con fines didácticos para las materias que el Laboratorio de Cibernética *Alejandro Medina* ofrece a los alumnos de la Facultad de Ciencias, ya que cada una de las tarjetas, explicadas más adelante, pueden tomarse como módulos para la impartición de clases de electrónica digital y/o de arquitectura de una microcomputadora basada en el microprocesador de Motorola MC6800 y dispositivos periféricos que cubren a la familia de la Motorola (PIA's, ACIA's, acopladores, etc.), en las materias de Circuitos Digitales con Laboratorio, Máquinas Digitales con Laboratorio y Seminario de Cibernética.

Esta tesis consta de 9 puntos, donde el primer punto es la Introducción y que el lector debe haberla leído ya. El segundo punto describe, en forma general, lo que es el instrumento en sí y en base a qué microprocesador se construyó. En el tercer punto se hace énfasis en la tarjeta que contiene las 3 bases de prueba y los 3 PIA's, puesto que, en cierta forma, es la parte medular de este instrumento para poder realizar las pruebas sin tanto conexonamiento como el que se mencionó al principio de la introducción. Sólo hay dos conexonamientos externos: el Vcc y el GND. El cuarto punto se refiere a todo el software (o programación) que requiere el Instrumento Programable, dando una explicación lo más clara que sea posible de lo que el programa hace. El quinto punto trata de las pruebas que realicé en el laboratorio y de los resultados que obtuve para cada uno de los circuitos que ahí se mencionan. El sexto punto da las conclusiones a las que llegué y de otras posibilidades que puede tener el instrumento, o alguna, o algunas de sus partes. Los tres últimos puntos, el séptimo, el octavo, y el noveno son los apéndices, un glosario de términos y la bibliografía, respectivamente.

2. - DESCRIPCION GENERAL DEL INSTRUMENTO.

El propósito fué el de diseñar, desarrollar, realizar e implementar un dispositivo capaz de poder probar el funcionamiento de diferentes Circuitos Integrados Digitales (C.I.D.). A este dispositivo se le ha dado el nombre de Instrumento Programable (I.P.) y no es más que una microcomputadora de 8 bits dedicada exclusivamente a la realización de pruebas de diferentes C.I.D. utilizando, además, una interfase especial de la cual hablaremos más adelante en el punto 3: INTERFASE PARA PRUEBAS DE DIFERENTES C.I.D. Para el desarrollo y construcción de este instrumento, nos basaremos en el microprocesador de la Motorola MC6800.

Como objetivo nos propusimos realizar un instrumento que tuviese la peculiaridad de poderse desarrollar en dos versiones, a saber: Una que llamamos Instrumento Programable (de la cual trata esta tesis), y a la otra la bautizamos como Instrumento Programado (el cual es tratado en una tesis que dirige, bibliografía # 12.).

El Instrumento Programable tiene la capacidad de poder ser programado de acuerdo a las indicaciones que el usuario dé para probar algún C.I.D., esto es: El usuario deberá dar todos los valores (los cuales se representan como ceros y unos lógicos) que dan lugar a las señales que activan al C.I.D. a probar y que el mismo usuario crea ameriten ser enviados a este C.I.D. Una vez dada toda la información pertinente del dispositivo que se va a probar, el I.P. realiza la prueba y muestra los resultados en la pantalla, donde el usuario puede determinar con facilidad si el C.I.D. probado está funcionando correctamente o no. Todo lo anterior se logrará cuando el usuario responda correctamente a todas las preguntas que el instrumento le hará. De aquí se desprende que esta versión del instrumento esta pensada sólo para gente que tenga muy amplios conocimientos en electrónica digital, que, por lo general, trabajan en laboratorios, talleres, etc.

El I.P. indica al usuario si desea realizar la misma prueba otra vez o si desea realizar una prueba diferente o probar algún otro C.I.D. Según la respuesta que dé el usuario, el programa procederá correctamente (se puede apreciar en el punto 4 y con mayor claridad en el punto 5: PRUEBAS Y RESULTADOS.). Esto da la ventaja de poder comprobar tantas veces como uno quiera si el C.I.D. está funcionando adecuadamente o no, con una misma prueba o con diferentes pruebas para el mismo C.I.D. Otra ventaja es la de poder probar tantos C.I.D. diferentes como uno quiera sin la necesidad de estar haciendo el alambrado correspondiente para cada uno de estos C.I.D.

Una desventaja que tiene el I.P. es que al finalizar cada prueba, el usuario debe de borrar los valores que se obtuvieron en la o las salidas que presente el C.I.D. Esto se logra dando el número de la terminal que representa a la salida del C.I.D. o pulsando la tecla espaciadora o simplemente oprimiendo la tecla del ENTER.

Otra desventaja del I.P. es cuando se desea probar un C.I.D. sincrónico (o sea, que utiliza una terminal para la señal del reloj), puesto que el usuario debe de simular la señal de reloj, es decir, deberá de cambiar de 0-lógico a 1-lógico y de 1-lógico a 0-lógico para que el dispositivo en prueba detecte, con la subida o la bajada del pulso de reloj (característica dada por el fabricante del C.I.D. a probar), cuándo debe de realizar la transición de acuerdo con los valores que el usuario ha dado.

Estas desventajas hacen que el manejo del I.P. sea un poco molesto, pero ambas deficiencias se pueden corregir a través de la programación (lo cual se hará para versiones futuras de este I.P.).

El Instrumento Programado solamente pedirá al usuario que dé la nomenclatura del C.I.D. que desea probar, esto es: para probar un QUAD-NAND, el usuario escribirá: 7400, y esta versión del instrumento se encargará de verificar que tenga la rutina que realice las pruebas suficientes y necesarias para el C.I.D. que el usuario desee probar. El mismo instrumento se encargará de comandar las señales adecuadas que probarán al C.I.D. e informará al usuario del resultado obtenido en dicha prueba. Este instrumento podrá ser utilizado casi por cualquier persona, puesto que el usuario sólo necesitará saber dar la nomenclatura de cada C.I.D. que vaya a probar y de cómo conectar o meter en la base correcta al C.I.D. A este dispositivo se le puede dar una finalidad más bien comercial.

En lo que se refiere al hardware que conformará a ambas versiones, diré que son exactamente iguales. La diferencia entre ambos instrumentos será en la parte del software o programación.

El Programa Principal del Instrumento Programable tiene la versatilidad de hacer preguntas al usuario y de obtener las respuestas de éste para que, así, se vayan conformando las señales necesarias y suficientes que deberán aplicarse al C.I.D. a probar. Con las preguntas que se le hacen al usuario y las indicaciones de procedimiento que se dan en ciertos puntos del programa es prácticamente innecesario tener un manual de operación para este I.P. (esto se puede apreciar mucho mejor en los puntos 4 y 5 de esta tesis).

El Programa Principal del Instrumento Programado solamente pedirá la nomenclatura del C.I.D. a probar y el resto corresponderá al Programa Principal, el cual verificará que exista la rutina de prueba de la nomenclatura dada y procederá según el resultado; esto es, si no existe la rutina para la nomenclatura dada se enviará un mensaje adecuado al usuario y, si existe, se procederá a hacer las pruebas pertinentes al C.I.D. a probar y el resultado de dicha o dichas pruebas será dado a conocer al usuario. Todo esto quiere decir que habrá un programa (o un conjunto de rutinas y subrutinas) para cada C.I.D. diferente para que, automáticamente, se realicen las pruebas al susodicho C.I.D. Se podrán probar tantos C.I.D. diferentes como rutinas y subrutinas

puedan entrar en 16 Kbytes de memoria EPROM.

Hemos hablado, en forma general, de las diferencias existentes en el software de ambas versiones. Ahora pasemos a ver como está conformado el hardware, el cual es el mismo para ambos instrumentos. La diferencia estriba en que el Programable utiliza menos memoria EPROM que el Programado.

De aquí en adelante sólo hablaré del Instrumento Programable.

Este instrumento está basado en el microprocesador de Motorola MC6800, junto con 16 Kbytes de memoria EPROM, 3 Kbytes de memoria RAM estática y 3 PIA's (Peripheral Interface Adapter) principalmente. Se desarrolló en el Laboratorio de Cibernética *Stefandro Medina*, en la Facultad de Ciencias, U.N.A.M.

Algunos de los problemas con los que nos enfrentamos para diseñar y construir un instrumento capaz de probar diferentes C.I.D. son:

- 1º) No todos los C.I.D. tienen el mismo número de patitas (terminales o pines).
- 2º) El conexionamiento de las terminales para Vcc y GND no es el mismo para todos los circuitos integrados digitales.
- 3º) La gran diferencia que hay entre las patitas que son entradas de señales y las terminales que son salidas de señales (en cada circuito integrado digital). Considerando en estos dos grandes grupos, entradas y salidas: a las que son de direccionamiento, las que son de control y las que son de datos.

El punto 1º) lo resolvemos poniendo tres bases diferentes: una de 16 terminales, otra de 24 terminales y la última de 40 terminales.

La solución de los puntos 2º) y 3º) presentan un grado de dificultad bastante alto, puesto que el poder seleccionar en la forma correcta a cada una de las patitas, o terminales, de la base donde estará colocado el C.I.D. y de cuáles son las entradas y cuáles las salidas del circuito a probar, así como la alimentación de Vcc y GND que le corresponda, no es tan trivial. Afortunadamente, existe un dispositivo electrónico que nos permite solucionar el problema en una forma bastante sencilla y simple, que es la PIA (Peripheral Interface Adapter, de Motorola y nomenclatura 6820 ó 6821). Este dispositivo es programable y nos permite usar sus puertos bidireccionales para seleccionarlos como entradas o como salidas, según sean las necesidades y, así, enviar las señales que serán entradas al C.I.D. y las que serán salidas; obteniendo con esto el resultado de la prueba hecha al C.I.D.

Ahora pasemos a ver como se integró al instrumento.

A excepción hecha de la tarjeta que contiene al microprocesador, todas las demás tarjetas fueron probadas con el sistema de

desarrollo *Exercicion 1A*, con el que cuenta el Laboratorio de Cibernética *Alejo Medina*, y debido a la utilización de este sistema de desarrollo, el Mother-Board del instrumento quedó como se muestra en la figura 2.1.

El I.P. está conformado por cinco tarjetas usando la técnica del Wire-Wrap para el conexionamiento entre los componentes electrónicos de cada tarjeta. Las cinco tarjetas son:

- 1^o) Tarjeta del Microprocesador (MPU) MC6800 de Motorola.
- 2^o) Tarjeta de Memoria EPROM para mantener la programación (Co los programas).
- 3^o) Tarjeta de Memoria RAM para poner el STACK y otras funciones.
- 4^o) Tarjeta del ACIA para comunicación entre instrumento y usuario.
- 5^o) Tarjeta de la interfase encargada de realizar los conexionamientos apropiados para probar a los circuitos integrados digitales.
- 6^o) Y la programación adecuada.

1^a) Tarjeta del Microprocesador (MPU) MC6800 de Motorola.

En la figura 2.2 se aprecian, por medio de los bloques, todos los componentes que están asociados al MPU. Esta figura muestra el contenido de la tarjeta que tiene toda la circuitería necesaria e indispensable para el Circuito del Reloj, del Reset y del Control, así como los acopladores (buffers) para las señales de direcciones y de los datos. Las señales de control, de datos y de direcciones están conectadas al bus del sistema o Mother-Board, para así poder realizar una comunicación adecuada entre esta tarjeta y las demás que conforman al instrumento Programable. Se utilizó su lenguaje ensamblador para todo el desarrollo de la programación que requiere el I.P. Para un mayor conocimiento y detalles de este microprocesador y de su conexionamiento, favor de consultar en la bibliografía las referencias siguientes: 2, 5, 7, 10, 24, 27, 28, 29, 30, 31 y 32; puesto que el objetivo de esta tesis no es el microprocesador en sí, sino su uso para formar al I.P.

En la figura 2.3 se muestran, un poco más detalladamente, los circuitos integrados que se han utilizado para acoplar las señales de dirección, de datos y de control junto con al MPU MC6800.

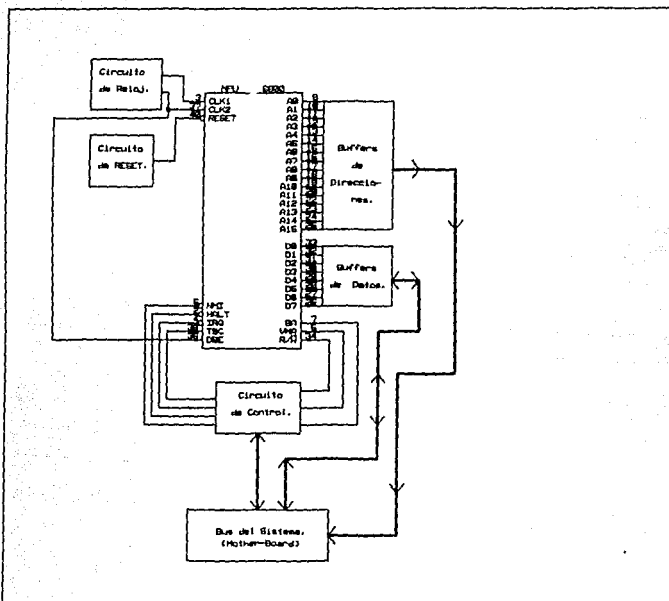
La figura 2.4 muestra cómo se construyó la circuitería que da lugar a la señal del Reset. Para que el microprocesador MC6800 se active con esta señal, es necesario que pasen, al menos, 8 ciclos de reloj. Esta señal del Reset fue calculada para que dure más de los 8 ciclos de reloj y, así, asegurar que el microprocesador se activará al serle aplicada esta señal, para que ejecute la rutina de inicialización dada por la dirección que se encuentra en las dos localidades de memoria más altas: \$FFFF y \$FFFE.

Asignación de Pines en el Mother Board.

Exerciser	Instrumento Pin	Función	Instrumento Pin	Función	Exerciser
A, B, C	A	+BV	1	+BV	1, 2, 3
	B	+CV	2	+CV	
AD	C	A0	3	N.C.	
\overline{U}	D	A1	4	BZ	J
U	E	A2	5	R/W	6
3B	F	A3	6	Reset	6
3B	H	A4	7	Hold	4
\overline{Y}	J	A5	8	\overline{ZPO}	D
\overline{E}	K	A6	9	NPC	E
37	L	A7	10	WNO	F
36	M	A8	11	TSC	
\overline{R}	N	A9	12	D0	P
\overline{D}	P	A10	13	D0	31
36	R	A11	14	D1	20
34	S	A12	15	D2	K
\overline{N}	T	A13	16	D3	H
\overline{H}	U	A14	17	D4	32
33	V	A15	18	D6	3B
	W	N.C.	19	D6	L
	X	N.C.	20	D7	J
W, X, Y	Y	Q0	21	Q0	41, 42, 43
	Z	Q1	22	Q1	

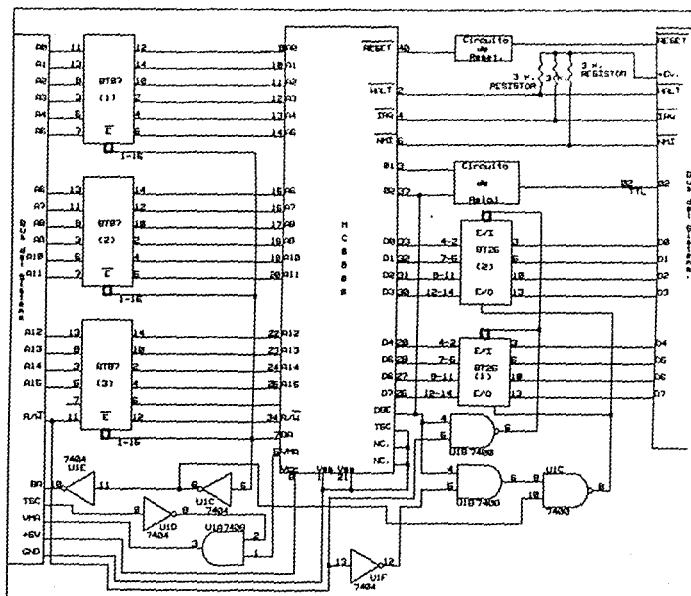
1

¹ FIGURA 2.1: Tabla que muestra la forma de asignar las terminales de los circuitos integrados digitales en el Mother-Board.



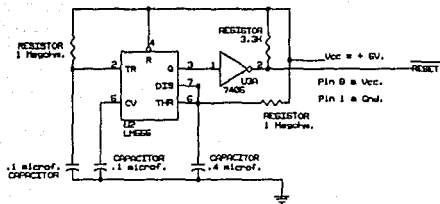
2

2 **FIGURA 2.2:** Por medio de los bloques se muestran todos los componentes que deben ir asociados con el microprocesador MC8800.



3

³ FIGURA 2.3: Se muestra el acoplamiento de las señales de dirección, de datos y de control con el microprocesador MC6800.



4

⁴ FIGURA 2.4: Circuitería que da lugar a la señal de RESET. Cuando es activada, dura más de 8 ciclos de reloj.

La figura 2.5 muestra claramente como se obtienen las señales de reloj que necesita el MPU MC6800, o sea, la señal $\phi 1$, $\phi 2$ y para el DBE (Data Bus Enable). El MPQ 6842 está diseñado para proveer la velocidad y voltajes de saturación necesarios en la circuitería del reloj para las entradas $\phi 1$ y $\phi 2$ del microprocesador MC6800.

2^a) Tarjeta de Memoria EPROM para mantener la Programación Co los Programas.

En la figura 2.6 se muestra la decodificación de las direcciones para poder seleccionar al EPROM (Erasable Programmable Read Only Memory) adecuado y obtener la información que se desea de la localidad de su memoria direccionada.

La figura 2.7 es la continuación de la figura 2.6. Aquí se muestran las ocho Memorias de sólo lectura, programables y borrables (EPROM's). Cada EPROM tiene una capacidad de almacenamiento de 2 Kbytes, dando un total de 16 Kbytes para guardar toda la programación pertinente; esto es, las rutinas y subrutinas necesarias para poder realizar correctamente las pruebas adecuadas al circuito integrado digital a probar, si se trata del instrumento Programado; o para efectuar las preguntas adecuadas y luego llevar al cabo las pruebas, si se trata del instrumento Programable, así como las rutinas de comunicación con el usuario a través de una pantalla o CRT. Ver figura 2.4.

3^a) Tarjeta de Memoria RAM para poner el STACK y para otras funciones.

En la figura 2.8 tenemos la decodificación de las direcciones apropiadas para seleccionar al RAM (Random Access Memory o Memoria de Acceso Aleatorio, que es de 3 Kbytes) adecuado y guardar ahí la información que se envía al C.I.D. a probar y también para guardar la información que el C.I.D. envía al I.P., recoger la información en la localidad de memoria apropiada y poder desplegar los resultados en la pantalla, para que el usuario verifique y ratifique los resultados así obtenidos. Ver figura 2.3. La figura 2.9 detalla el conexiónamiento para los 3 Kbytes de memoria RAM que tiene este I.P.

4^a) Tarjeta del ACIA para comunicación entre instrumento y usuario.

La figura 2.10 muestra la decodificación para poder seleccionar al ACIA (Asynchronous Communication Interface Adapter, Interfase Adaptadora de Comunicación Asíncrona). Este dispositivo permite entablar la comunicación entre el usuario y el instrumento por medio de un monitor o CRT y lograr así llevar al cabo correctamente las pruebas al C.I.D. con los valores apropiados, y comunicar al usuario los resultados a través del monitor o pantalla. Ver figura 2.3.

La figura 2.11 es la continuación de la figura 2.10 y se muestra el conexionamiento del ACIA con el mundo exterior (representado en los dos rectángulos del extremo derecho de la figura 2.11) y la circuitería que permite trabajar con el ACIA a 1200 bauds aproximadamente.

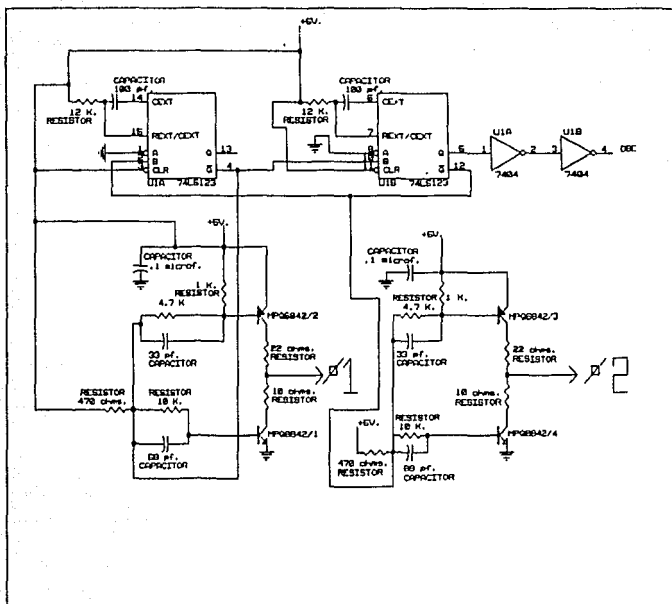
5^a) Tarjeta de la Interfase encargada de realizar los conexionamientos apropiados para probar a los circuitos integrados digitales.

Una interfase que llamo 'conmutadora', es la más importante del I.P. puesto que sin ella las pruebas a los diferentes C.I.D. no se podrían llevar al cabo, ya que se encarga de hacer las conexiones apropiadas entre las PIA's y la base donde estará inserto el C.I.D. a probar. Serán tres bases: Una base para C.I.D. que tengan 14 ó 16 terminales; otra para los que tengan 24 terminales y la tercera para los de 40 terminales. La descripción se realiza en el punto 3 de esta tesis: INTERFASE PARA PRUEBAS DE DIFERENTES C.I.D. Ver figura 2.5.

6^a) El programa que necesita este instrumento y que se detalla su algoritmo en el punto 4 de esta tesis.

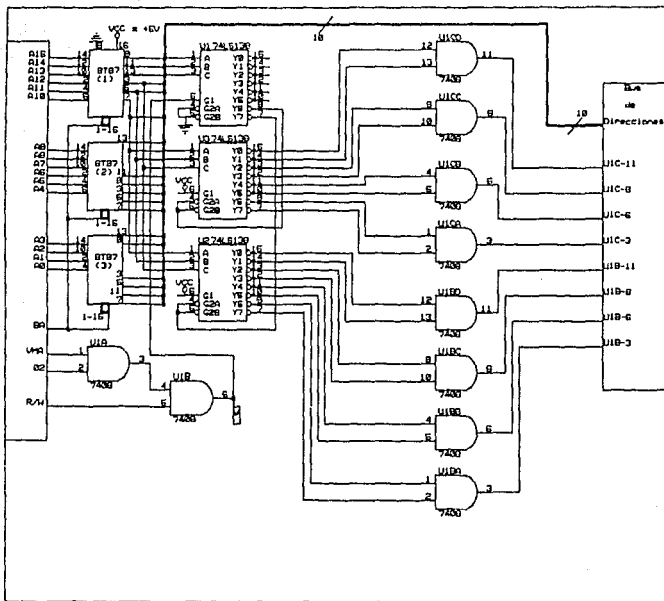
En cualquiera de las bases, el Vcc y el GND se tendrán que conectar externamente, y ésta será la única conexión externa que se necesitará hacer. Todas las demás conexiones, como entradas, salidas, direcciones, control, selección y/o datos, los hará el propio instrumento a través de las PIA's y las bases.

La finalidad de esta tesis no es la de explicar con detalle como están construidas estas tarjetas, salvo la quinta a la que se le dedica un capítulo completo, puesto que ya se dijo que es una microcomputadora dedicada y la construcción de esta se puede obtener de la bibliografía dada al final de esta tesis.



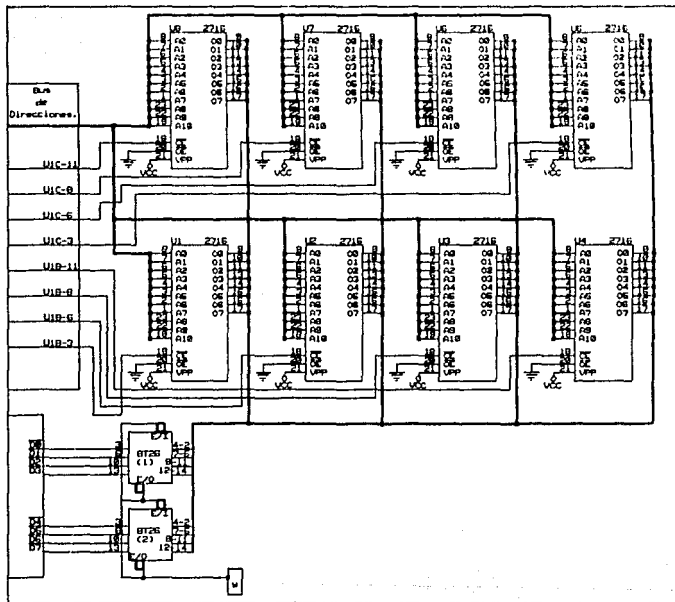
5

5. FIGURA 2.5. Obtención de las señales ϕ_1 , ϕ_2 y DBE (Data Bus Enable).



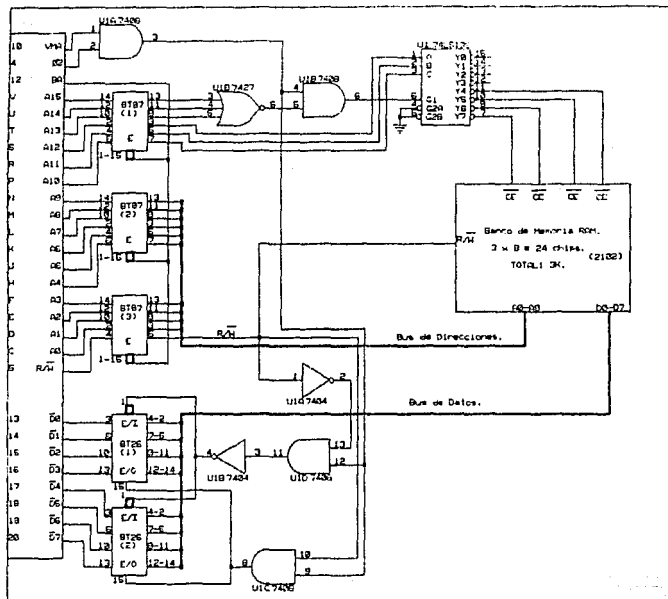
6

FIGURA 2.6. Decodificación de direcciones para seleccionar al EPPROM (Erasable Programmable Read Only Memory) apropiado.



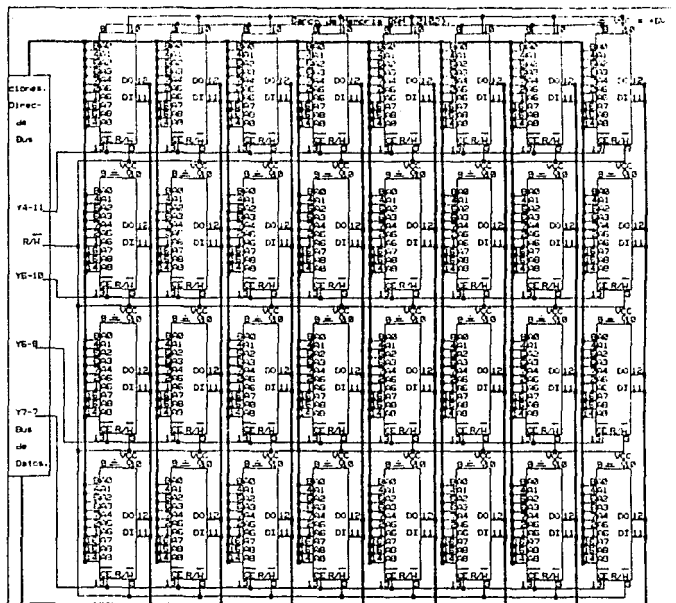
7

7 FIGURA 2.7. Continuación de la FIGURA 2.6, mostrando los 8 EPROM's.



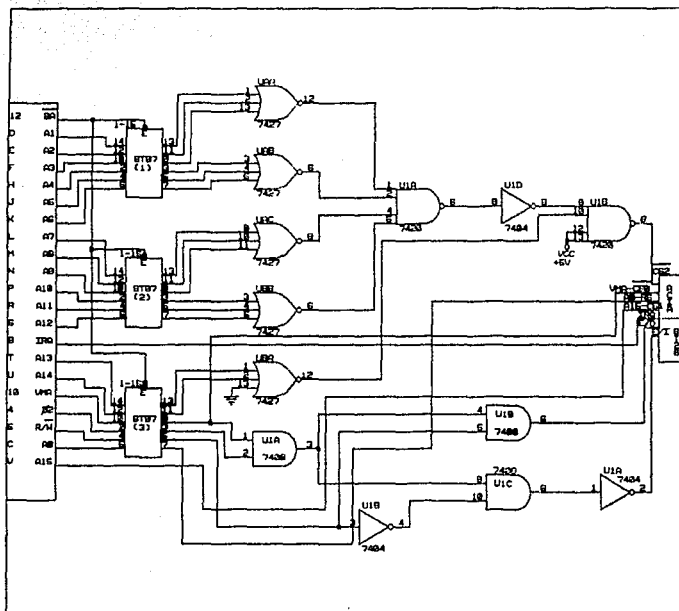
5

FIGURA 2.8. Decodificación de direcciones para seleccionar al PAM (Paradigm Access Memory) apropiado.



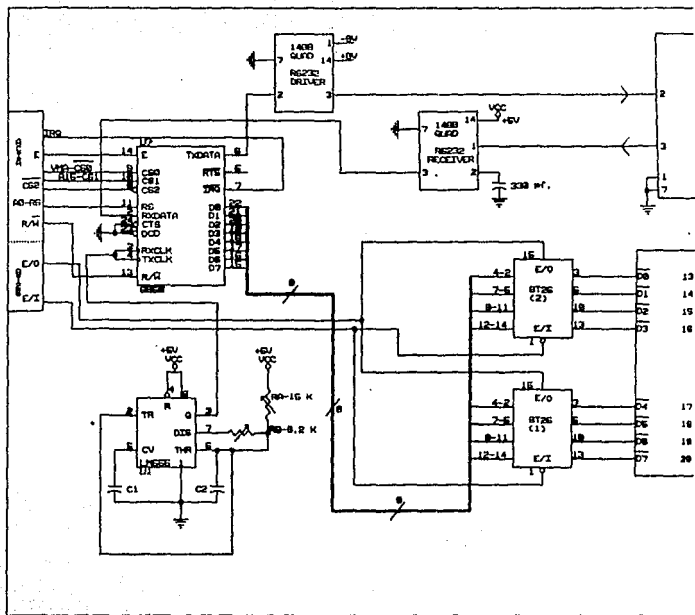
9

9 FIGURA 2.9. Detalla el conexionamiento de la memoria RAM.



10

10 FIGURA 2.10: Muestra la decodificación para seleccionar al ACIA (Asynchronous Communication Interface Adapter).



11

11 FIGURA 2.11: Conexión del ACIA y circuitería para trabajar a 1200 bauds aproximadamente.

3. - INTERFASE PARA PRUEBAS DE DIFERENTES C.I.D.

Tanto en la introducción como en la descripción del instrumento se ha dicho todo lo que es el instrumento programable *groose mode*. Aquí haré más énfasis en la parte del hardware que consideramos más importante, que es la interfase que dará lugar al conexiónamiento correcto de una de las tres bases que tendrá este dispositivo, para poder probar al C.I.D. deseado. Esta tarjeta amerita una descripción más detallada de su funcionamiento como interfase entre la microcomputadora y las bases donde se realizarán las pruebas pertinentes al circuito integrado digital que se desee probar, puesto que también veremos las dificultades con las que me enfrenté y como logré resolverlas.

En esta tarjeta se encuentran las bases donde se instalan los circuitos integrados digitales que se van a probar por medio del instrumento. Asimismo están las 3 PIA's (Peripheral Interface Adapter) con las cuales vamos a seleccionar a la base adecuada para poder realizar las pruebas pertinentes al circuito integrado digital que se desee probar.

Esta interfase para pruebas de diferentes C.I.D. es la tarjeta modular del I.P., ya que es en ésta donde se realizarán todas las pruebas a los diferentes C.I.D. Está compuesta por tres bases de palanca (bases que sujetan las terminales o pines o patitas de un C.I.D. por medio de una palanca) montadas sobre unas bases para wire-wrap. La primera base es de 16 terminales en donde se pueden probar C.I.D. de 14 ó 16 terminales. La segunda base es para C.I.D. de 24 terminales y la tercera base es para poder probar C.I.D. de 40 terminales.

A estas tres bases están conectadas tres dispositivos electrónicos conocidos como PIA's (Peripheral Interface Adapter; Interfases de entrada-salida con 2 puertos de 8 líneas cada uno; cada una de las 16 líneas puede programarse como entrada o como salida, según se requiera). Para las tres PIA's tenemos un total de 48 líneas, de las cuales se conectarán a las bases sólo 40 líneas. Ver figura 3.1. Las conexiones entre las PIA's y las bases es como sigue: Las 16 líneas de entrada - salida de la primera PIA están conectadas a las 16 terminales de la primera base y a las 16 primeras terminales de la segunda base así como también de la tercera base. El primer puerto de la segunda PIA (8 líneas) está conectado a las terminales de la segunda base que van de la terminal 17 a la 24 y en igual forma a la tercera base. El segundo puerto de la segunda PIA y el primer puerto de la tercera PIA (16 líneas en total) están conectados a las terminales de la tercera base, partiendo de la terminal 25 y terminando en la terminal 40 (las terminales en las bases están numeradas como en los C.I.D.). Este conexiónamiento se puede apreciar mejor viendo la figura 3.1. Con esta forma de conexiónamiento surgen dos pequeños inconvenientes: El Vcc y el GND deben conectarse en forma externa (manualmente) y, segundo, sólo se puede probar un C.I.D. a la vez, o sea,

forzosamente 2 de las 3 bases deben quedar libres de C.I.D. Con el software (Programa Principal) se programarán adecuadamente a las tres PIA's, seleccionando apropiadamente a los puertos que serán entradas a la(s) PIA(s) (salidas en el C.I.D. a probar) y los que serán salidas de la(s) PIA(s) (entradas al C.I.D. a probar). Ver figura 3.1.

En la figura 3.2 se muestra la forma en que se decodifica la señal de direccionamiento para poder seleccionar a cualquiera de las tres PIA's y así programarlas apropiadamente para la realización de las pruebas que vaya a efectuar el instrumento (comandado por el usuario, desde luego).

La figura 3.3 detalla la forma de conectar la unidad 825 para poder obtener la bidireccionalidad del bus de datos en esta interfase. La notación: PIA (33) 1, 2 y 3, por ejemplo, significa que en las tres PIA's esta línea se conecta a la terminal # 33 de cada una de las PIA's.

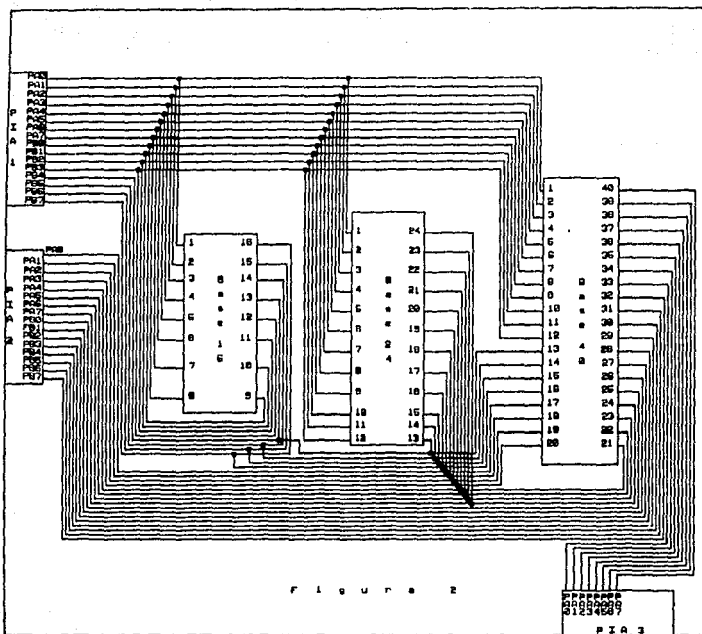
En la figura 3.4 podemos apreciar como los 16 puertos de entrada/salida que tiene la PIA-1 son conectados a la base de 16 terminales (o patitas). Un puerto de la PIA, digamos el PA0, va a la terminal 1 de la base de 16 terminales. La secuencia de conexionamiento es ascendente tanto en la PIA como en la base, como se puede apreciar en la figura 3.4. El cuadro de la derecha indica que estas 16 líneas se conectarán, en igual forma, a las terminales de las bases de 24 y de 40 patitas, esto es, de la 1 a la 16 en la base de 24 y de la 1 a la 16 en la base de 40 terminales.

En la figura 3.5 vemos que a la base de 24 terminales van conectados los puertos de la PIA-1 (PA0 a PB7) y que los puertos del PIA-2 (PA0 a PA7) se conectan en forma ascendente a la base de 24 terminales, a partir de la terminal # 17 hasta la 24 inclusive, tal y como se puede apreciar en esta figura 3.5.

En la figura 3.6 se muestra como los puertos PBO a PB7 de la PIA-2 se conectan a la base de 40 terminales, en forma ascendente de la terminal 25 a la 32. Las terminales 33 a 40 van conectadas al puerto PA0 a PA7, respectivamente, de la PIA-3.

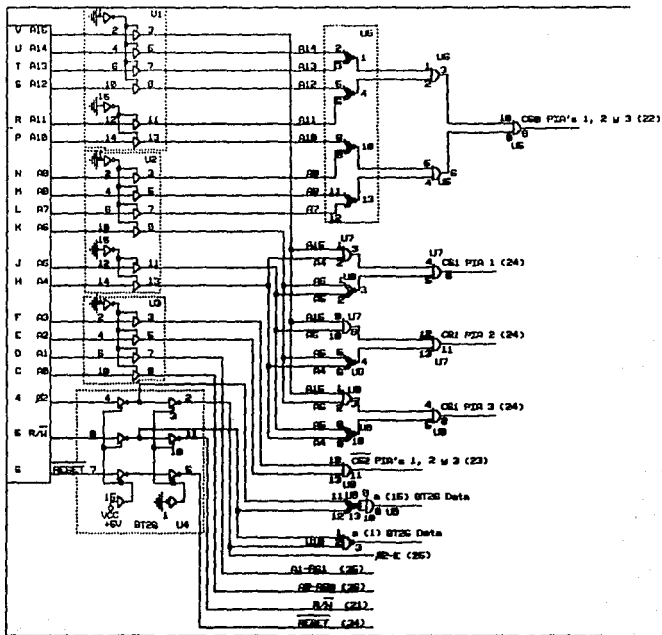
Es por esta forma de conectar las 3 PIA's a las tres bases que solamente se puede probar un sólo circuito integrado digital a la vez.

Esta misma interfase se puede realizar en una tarjeta que se conecte a cualquier PC, seleccionando algún puerto que por lo general todas las PC's tengan libre, y se desarrolle la programación pertinente quedando así como interfase compatible con una PC. También podría hacerse para cualquier otro tipo de computadora, por ejemplo: APPLE, ATARI, etc.



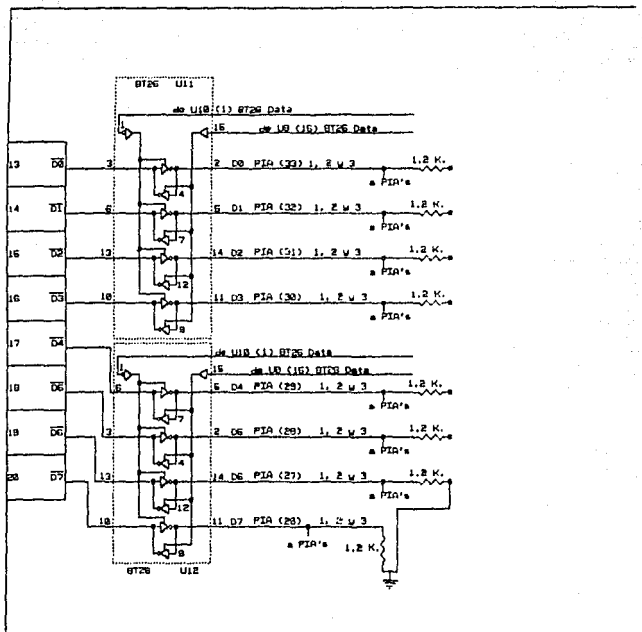
12

¹² **FIGURA 3.1:** Conexión de las tres PIA's (Peripheral Interface Adapter) con las tres bases para realizar pruebas.



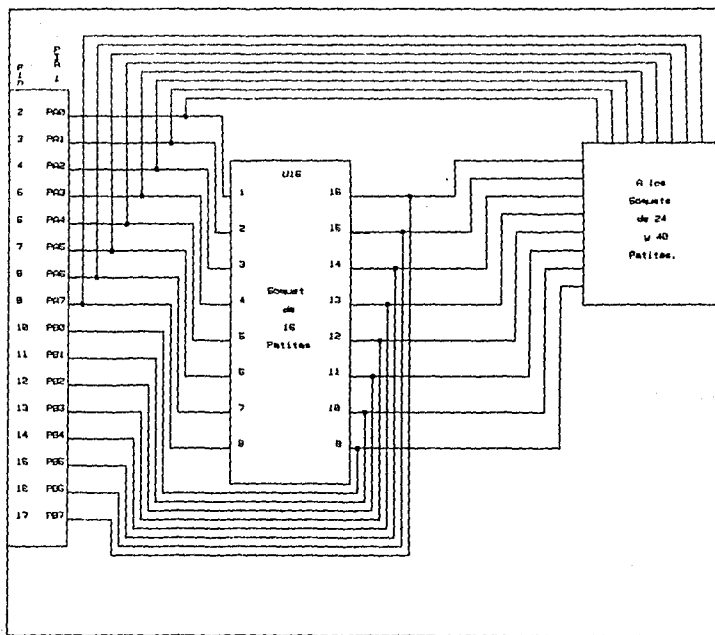
13

13 **FIGURA 3.2:** Decodificación de la dirección para seleccionar la PIA apropiada.



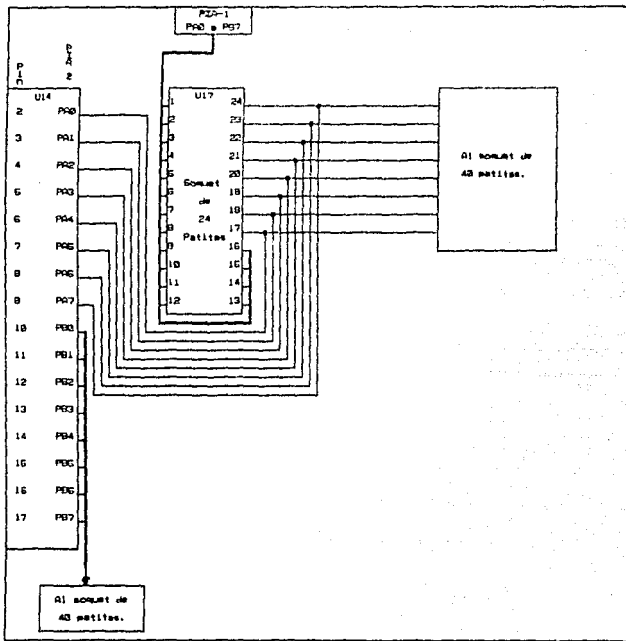
14

14 **FIGURA 3.3:** Detalle para obtener la bidireccionalidad en el bus de datos.



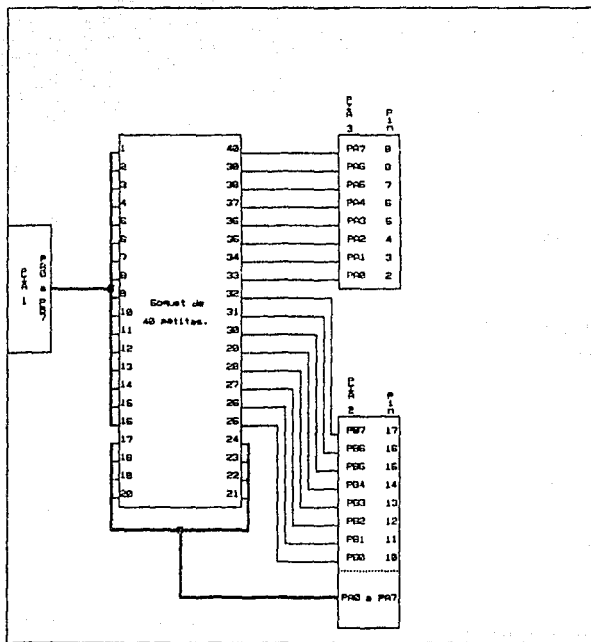
15

15 **FIGURA 3.4:** Detalla el conexionamiento de la PIA 1 con la base de 16 terminales.



16

16 FIGURA 3.5: Detalla el conexionamiento de la PIA 1 y PIA 2 con la base de 24 terminales.



17

17 **FIGURA 3.6:** Detalla el conexionamiento de las 3 PIA's con la base de 40 terminales.

El inconveniente que veo para hacer esta interfase para alguna computadora es que ésta quedaría restringida durante el tiempo que se usara esta interfase, por esta razón se hizo el I.P. que no es más que una computadora dedicada exclusivamente a la prueba de diferentes C.I.D. con un microprocesador menos poderoso que los usados en las PC's compatibles. Estos microprocesadores de 8 bits son los ideales para el desarrollo de instrumentos 'inteligentes' o dedicados a una sola labor o varias labores como es el caso de este I.P.

4. - ALGORITMO PARA EL INSTRUMENTO PROGRAMABLE.

Se detalla el algoritmo que generó al programa que contiene el I.P. con todos los mensajes que el usuario podrá ver en pantalla, esto es, suponiendo que también cometa errores y como corregirlos, prácticamente sin necesidad de un manual de cómo operar a este I.P. Se ennumeran los pasos de este algoritmo para facilitar el seguimiento de éste. Todos los mensajes que se envían a pantalla, para que el usuario los lea, se ponen con letras mayúsculas. Con las respuestas del usuario no hay problema puesto que se toman en cuenta las dos posibilidades: Respuesta Correcta y Respuesta Incorrecta. En el apéndice B se da el diagrama de flujo.

El algoritmo es como sigue:

1. - Se envía el siguiente mensaje de iniciación:

INSTRUMENTO PROGRAMABLE.

VERSION LABORATORIO.

2. - NUMERO TOTAL DE PATITAS: (14, 16, 24, O 40) ?

Si el número dado es el correcto, señalado entre paréntesis, se pasa al punto 3.

Si el número dado fué distinto a los señalados entre paréntesis, se envía el siguiente mensaje:

NUMERO TOTAL DE PATITAS: (14, 16, 24, O 40) ; EQUIVOCADO.
VOLVER A DARLO.

y el control pasa al punto 2.

Si se oprimió una tecla no numérica, el mensaje que se envía es el siguiente:

CARACTER INAPROPIADO.

y se queda aquí hasta que se dé el o los caracteres adecuados para poder pasar al punto 3.

3. - DAR EL NUMERO DE LA PATITA PARA Vcc.

Si el número para Vcc es el correcto el control pasa al punto número 4. Si dicho número está dentro del intervalo correcto (intervalo obtenido en el punto 2) pero no corresponde al Vcc sino a alguna otra señal, su corrección se podrá realizar en el punto 10.

Si se oprimió una tecla no numérica, el mensaje que se envía es el siguiente:

CARACTER INAPROPIADO.

y se queda en este paso hasta que el usuario dé los caracteres apropiados u oprima la tecla del ENTER o la letra F de terminación o Fin, y regresar al punto 3 si dió el ENTER o la F o pasar al punto 4 si escribió el o los números adecuados.

Si se oprimió la tecla del ENTER por error y no se dio información alguna o el número dado fué mayor a 40 decimal, entonces, se informa al usuario lo siguiente:

NO HAY DATOS O EL DATO ES MAYOR DE 40 DECIMAL.

y el control regresa al punto 3.

Si se excede el número total de terminales que se dió en el punto 2 entonces se indica lo siguiente:

EL VALOR DADO ES MAYOR QUE EL ESPECIFICADO: XX.
VOLVER A DARLO.

donde XX representa al número escrito en el punto 2 y el control regresa al punto 3.

El siguiente mensaje de error no debe de proceder aquí sino del punto 4 en adelante, pero se contempla, y es cuando se ha repetido el número de alguna de las terminales del C.I.D. que se vaya a probar:

SE HA REPETIDO EL NUMERO DE PATITA. VOLVER A DARLO.

y el control regresa al punto 3.

4. - DAR EL NUMERO DE LA PATITA PARA GND.

Si el número para GND es el correcto el control pasa al punto número 5. Si dicho número está dentro del intervalo correcto (intervalo obtenido en el punto 2) pero no corresponde al GND sino a alguna otra señal, su corrección se podrá realizar en el punto 10.

Si se oprimió una tecla no numérica, el mensaje que se envía es el siguiente:

CARACTER INAPROPIADO.

y se queda en este paso hasta que el usuario dé los caracteres apropiados u oprima la tecla del ENTER o la letra F de terminación o Fin, y regresar al punto 4 si dió el ENTER o la F o pasar al punto 5 si escribió el o los números adecuados.

Si se oprimió la tecla del ENTER por error y no se dió información alguna o el número dado fué mayor a 40 decimal, entonces, se informa al usuario lo siguiente:

NO HAY DATOS O EL DATO ES MAYOR DE 40 DECIMAL.

y el control regresa al punto 4.

Si se excede el número total de terminales que se dió en el punto 2 entonces se indica lo siguiente:

EL VALOR DADO ES MAYOR QUE EL ESPECIFICADO: XX.
VOLVER A DARLO.

donde XX representa al número escrito en el punto 2 y el control regresa al punto 4.

Cuando se ha repetido el número de alguna de las terminales del C.I.D. que se vaya a probar, se envía el mensaje:

SE HA REPETIDO EL NUMERO DE PATITA. VOLVER A DARLO.

y el control regresa al punto 4.

5. - HAY PATITAS DE DIRECCION (AO...AND) ?
RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

Si se oprime cualquier tecla distinta de la 'S', el control pasará al punto 6. Si la tecla oprimida fué la 'S', entonces, el programa da las siguientes indicaciones:

5.1. - PARA FINALIZAR OPRIMIR TECLA F.
5.2. - DAR EL NUMERO DE CADA PATITA QUE SEA DE DIRECCION.

Si el número está dentro del intervalo correcto (intervalo obtenido en el punto 2) pero no corresponde a una dirección sino a alguna otra señal, su corrección se podrá realizar en el punto 10. Se verifica si se oprimió la tecla 'F' y, así, el control pasa al punto 6, si no, continúa al punto 5.3 si no ha habido errores y si los ha habido, entonces se ve cual de los siguientes fueron:

Si se oprimió una tecla no numérica, el mensaje que se envía es el siguiente:

CARACTER INAPROPIADO.

y se queda en este paso hasta que el usuario dé los caracteres apropiados u oprima la tecla del ENTER o la letra F de terminación o Fin, y regresar al punto 5.2 si dió el ENTER o la F o pasar al punto 5.3 si escribió el o los números adecuados.

Si se oprimió la tecla del ENTER por error y no se dió información alguna o el número dado fué mayor a 40 decimal, entonces, se informa al usuario lo siguiente:

NO HAY DATOS O EL DATO ES MAYOR DE 40 DECIMAL.

y se verifica si el usuario ha oprimido la tecla 'F' con lo cual el control pasa al punto 6, si no el control regresa al punto 5.2.

Si se excede el número total de terminales que se dió en el punto 2 entonces se indica lo siguiente:

EL VALOR DADO ES MAYOR QUE EL ESPECIFICADO: XX.
VOLVER A DARLO.

donde XX representa al número escrito en el punto 2 y el control regresa al punto 5.2.

Cuando se ha repetido el número de alguna de las terminales del C.I.D. que se vaya a probar, se envía el mensaje:

FF HA REPETIDO EL NUMERO DE PATITA. VOLVER A DARLO.

y el control regresa al punto 5.2.

5.3. - DAR VALORES LOGICOS PARA PATITAS DE DIRECCION (O 1).

y se regresa al punto 5.2 hasta que el usuario oprima la tecla 'F' para indicar que terminó de dar la dirección donde empezará la prueba del C.I.D. y pasar el control al punto número 6.

Si se da un valor distinto al cero o al uno, entonces se envía el siguiente mensaje al usuario:

ESCRIBIR UN CERO O UN UNO.

y el control regresa al punto 5.3.

6. - HAY PATITAS DE CONTROL (CR/W, E, ETC.) ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

Si se oprime cualquier tecla distinta de la 'S', el control pasará al punto 7. Si la tecla oprimida fué la 'S', entonces, el programa da las siguientes indicaciones:

6.1. - PARA FINALIZAR OPRIMIR TECLA F.

6.2. - DAR EL NUMERO DE CADA PATITA QUE SEA DE CONTROL.

Si el número está dentro del intervalo correcto (intervalo obtenido en el punto 2) pero no corresponde a una señal de control sino a alguna otra señal, su corrección se podrá realizar en el punto 10. Se verifica si se oprimió la tecla 'F' y, así, el control pasa al punto 7, si no, continua al punto 6.3 si no ha habido errores y si los ha habido, entonces se ve cual de los siguientes fueron:

Si se oprimió una tecla no numérica, el mensaje que se envía es el siguiente:

CARACTER INAPROPIADO.

y se queda en este paso hasta que el usuario dé los caracteres apropiados u oprima la tecla del ENTER o la letra F de terminación o Fin, y regresar al punto 6.2 si dió el ENTER o la F o pasar al punto 6.3 si escribió el o los números adecuados.

Si se oprimió la tecla del ENTER por error y no se dió información alguna o el número dado fué mayor a '40 decimal, entonces, se informa al usuario lo siguiente:

NO HAY DATOS O EL DATO ES MAYOR DE 40 DECIMAL.

y se verifica si el usuario ha oprimido la tecla 'F' con lo cual el control pasa al punto 7, si no el control regresa al punto 6.2.

Si se excede el número total de terminales que se dió en el punto 2 entonces se indica lo siguiente:

EL VALOR DADO ES MAYOR QUE EL ESPECIFICADO: XX.
VOLVER A DARLO.

donde XX representa al número escrito en el punto 2 y el control regresa al punto 6.2.

Quando se ha repetido el número de alguna de las terminales del C.I.D. que se vaya a probar, se envía el mensaje:

SE HA REPETIDO EL NUMERO DE PATITA. VOLVER A DARLO.

y el control regresa al punto 6.2.

6.3. - DAR VALORES LOGICOS PARA PATITAS DE CONTROL (0 1).

y se regresa al punto 6.2 y así sucesivamente hasta que el usuario oprima la tecla 'F' para indicar que terminó de proporcionar la información pedida y pasar el control al punto número 7.

Si se da un valor distinto al cero o al uno, entonces se envía el siguiente mensaje al usuario:

ESCRIBIR UN CERO O UN UNO.

y el control regresa al punto 6.3.

7. - HAY PATITAS DE SELECCION (CSO...CSN) ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

Si se oprime cualquier tecla distinta de la 'S', el control pasará al punto 8. Si la tecla oprimida fué la 'S', entonces, el programa da las siguientes indicaciones:

7.1. - PARA FINALIZAR OPRIMIR TECLA F.

7.2. - DAR EL NUMERO DE CADA PATITA QUE SEA DE SELECCION.

Si el número está dentro del intervalo correcto (intervalo obtenido en el punto 2) pero no corresponde a una señal de selección sino a alguna otra señal, su corrección se podrá realizar en el punto 10. Se verifica si se oprimió la tecla 'F' y, así, el control pasa al punto 8, si no, continua al punto 7.3 si no ha habido errores y si los ha habido, entonces se ve cual de los siguientes fueron:

Si se oprimió una tecla no numérica, el mensaje que se envía es el siguiente:

CARACTER INAPROPIADO.

y se queda en este paso hasta que el usuario dé los caracteres apropiados u oprima la tecla del ENTER o la letra F de terminación o Fin, y regresar al punto 7.2 si dió el ENTER o la F o pasar al punto 7.3 si escribió el o los números adecuados.

Si se oprimió la tecla del ENTER por error y no se dió

información alguna o el número dado fué mayor a 40 decimal, entonces, se informa al usuario lo siguiente:
NO HAY DATOS O EL DATO ES MAYOR DE 40 DECIMAL.

y se verifica si el usuario ha oprimido la tecla 'F' con lo cual el control pasa al punto 8. si no el control regresa al punto 7.2.

Si se excede el número total de terminales que se dió en el punto 2 entonces se indica lo siguiente:

EL VALOR DADO ES MAYOR QUE EL ESPECIFICADO: XX.
VOLVER A DARLO.

donde XX representa al número escrito en el punto 2 y el control regresa al punto 7.2.

Cuando se ha repetido el número de alguna de las terminales del C.I.D. que se vaya a probar, se envía el mensaje:

SE HA REPETIDO EL NUMERO DE PATITA. VOLVER A DARLO.

y el control regresa al punto 7.2.

7.3. - DAR VALORES LOGICOS PARA PATITAS DE SELECCION (0 1).

y se regresa al punto 7.2 y así sucesivamente hasta que el usuario oprima la tecla 'F' para indicar que terminó de proporcionar la información pedida y pasar el control al punto número 8.

Si se da un valor distinto al cero o al uno, entonces se envía el siguiente mensaje al usuario:

ESCRIBIR UN CERO O UN UNO.

y el control regresa al punto 7.3.

8. - PARA FINALIZAR OPRIMIR TECLA F.

8.1. - DAR EL NUMERO DE CADA PATITA QUE SEA DE ENTRADA AL CIRCUITO.

Para los casos en que los C.I.D. no tengan señales de dirección, control o de selección, al punto 8.1 siempre llegará el usuario puesto que, por lo menos, un C.I.D. tiene entradas y salidas, las cuales forzosamente se tienen que dar. Si no se llegaran a dar los valores que correspondan a las terminales de entrada y a las de salida, o sea, no dar información alguna, entonces, el I.P. dará un mensaje de error fatal, a saber:

MEMORIA RAM DEL INSTRUMENTO EN PESIMO ESTADO.

!!!SE RECOMIENDA CAMBIAR MEMORIA RAM DEL INSTRUMENTO!!!

y se detiene la ejecución de este programa. Esto se hace en forma estrepitosa debido a que si no hay señales que enviar para realizar alguna prueba, podría dañarse al C.I.D. a probar, si hay alguno conectado, o al propio I.P. Es una pequeña medida de precaución. Si se ha dado toda la información requerida para realizar una prueba y se envía este mensaje, entonces, la memoria RAM del I.P. si está dañada y lo mejor será revisarla.

Si el número está dentro del intervalo correcto (intervalo obtenido en el punto 2) pero no corresponde a una señal de entrada sino a alguna otra señal, su corrección se podrá realizar en el punto 10. Se verifica si se oprimió la tecla 'F' y, así, el control pasa al punto 9, si no, continúa al punto 8.2 si no ha habido errores y si los ha habido, entonces se ve cual de los siguientes fueron:

Si se oprimió una tecla no numérica, el mensaje que se envía es el siguiente:

CARACTER INAPROPIADO.

y se queda en este paso hasta que el usuario dé los caracteres apropiados u oprima la tecla del ENTER o la letra F de terminación o Fin, y regresar al punto 8.1 si dió el ENTER o la F o pasar al punto 8.2 si escribió el o los números adecuados.

Si se oprimió la tecla del ENTER por error y no se dió información alguna o el número dado fué mayor a 40 decimal, entonces, se informa al usuario lo siguiente:

NO HAY DATOS O EL DATO ES MAYOR DE 40 DECIMAL.

y se verifica si el usuario ha oprimido la tecla 'F' con lo cual el control pasa al punto 9, si no el control regresa al punto 8.1.

Si se excede el número total de terminales que se dió en el punto 2 entonces se indica lo siguiente:

EL VALOR DADO ES MAYOR QUE EL ESPECIFICADO: XX.
VOLVER A DARLO.

donde XX representa al número escrito en el punto 2 y el control regresa al punto 8.1.

Cuando se ha repetido el número de alguna de las terminales del C.I.D. que se vaya a probar, se envía el mensaje:

SE HA REPETIDO EL NUMERO DE PATITA. VOLVER A DARLO.

y el control regresa al punto 8.1.

8.2. - DAR VALORES LOGICOS PARA PATITAS DE ENTRADA AL CIRCUITO (O 1).

y se regresa al punto 8.1 y así sucesivamente hasta que el usuario oprima la tecla 'F' para indicar que terminó de proporcionar la información pedida y pasar el control al punto número 9.

Si se da un valor distinto al cero o al uno, entonces se envía el siguiente mensaje al usuario:

ESCRIBIR UN CERO O UN UNO.

y el control regresa al punto 8.2.

9. - PARA FINALIZAR OPRIMIR TECLA F.

9.1. - DAR EL NUMERO DE CADA PATITA QUE SEA DE SALIDA DEL CIRCUITO.

Para los casos en que los C.I.D. no tengan señales de dirección, control o de selección, al punto 9.1 se llegará después de pasar por el 8.1, puesto que, por lo menos, un C.I.D. tiene entradas y salidas, las cuales forzosamente se tienen que dar. Si no se llegan a dar los valores que correspondan a las terminales de entrada y a las de salida, o sea, no dar información alguna, entonces, el I.P. dará un mensaje de error fatal, a saber:

MEMORIA RAM DEL INSTRUMENTO EN PESIMO ESTADO.
;;;SE RECOMIENDA CAMBIAR MEMORIA RAM DEL INSTRUMENTO;;;

y la ejecución de este programa es detenida. Esto se hace en forma estrepitosa debido a que si no hay señales que enviar para realizar alguna prueba, podría dañarse al C.I.D. a probar, si hay alguno conectado, o al propio I.P. Es una pequeña medida de precaución. Si se ha dado toda la información requerida para realizar una prueba y se envía este mensaje, entonces, la memoria RAM del I.P. si está dañada y lo mejor será revisarla.

Si el número está dentro del intervalo correcto (intervalo obtenido en el punto 2) pero no corresponde a una señal de salida sino a alguna otra señal, su corrección se podrá realizar en el punto 10. Se verifica si se oprimió la tecla 'F' y, así, el control pasa al punto 10, si no, continua al punto 10 si no ha habido errores y si los ha habido, entonces se ve cual de los siguientes fueron:

Si se oprimió una tecla no numérica, el mensaje que se envía es el siguiente:

CARACTER INAPROPIADO.

y se queda en este paso hasta que el usuario dé los caracteres apropiados u oprima la tecla del ENTER o la letra F de terminación o Fin, y regresar al punto 9.1 si dió el ENTER o la F o pasar al punto 10 si escribió el o los números adecuados.

Si se oprimió la tecla del ENTER por error y no se dió información alguna o el número dado fué mayor a 40 decimal, entonces, se informa al usuario lo siguiente:

NO HAY DATOS O EL DATO ES MAYOR DE 40 DECIMAL.

y se verifica si el usuario ha oprimido la tecla 'F' con lo cual el control pasa al punto 10, si no el control regresa al punto 9.1.

Si se excede el número total de terminales que se dió en el punto 2 entonces se indica lo siguiente:

EL VALOR DADO ES MAYOR QUE EL ESPECIFICADO: XX.
VOLVER A DARLO.

donde XX representa al número escrito en el punto 2 y el control regresa al punto 9.1.

Cuando se ha repetido el número de alguna de las terminales del C.I.D. que se vaya a probar, se envía el mensaje:

SE HA REPETIDO EL NUMERO DE PATITA. VOLVER A DARLO.

y el control regresa al punto 9.1.

Como esto corresponde a las terminales que son salidas en un C.I.D. que se vaya a probar, no debe asignarseles valor lógico alguno, puesto que éstas lo obtendrán de la prueba o las pruebas que se le realicen a dicho C.I.D.

Los siguientes 6 puntos son para enviar información a la pantalla y para que el usuario tome ciertas decisiones.

Si no se dió información a alguna o varias terminales del C.I.D. que se vaya a probar o, aún cuando se esté seguro de haber dado toda la información pertinente, resulte que el I.P. no está guardando la información en forma correcta, entonces y sólo entonces se enviará este error fatal:

MEMORIA RAM DEL INSTRUMENTO EN PESIMO ESTADO.

;;;SE RECOMIENDA CAMBIAR MEMORIA RAM DEL INSTRUMENTO;;;

Este mensaje tiene dos propósitos:

Primero: Informar de forma estrepitosa al usuario de si se le olvidó dar algún dato, puesto que este programa se ve incapacitado de poder cumplir correctamente con las funciones que se le han asignado al I.P. si falta algún dato, o varios, por lo cual se considera a este hecho como un error fatal y con este mensaje se le avisa al usuario de esta posible falta.

Segundo: Si el usuario dió todos los datos y está completamente seguro de que así fué, entonces, lo más probable es que la memoria RAM ya no esté funcionando apropiadamente. Se recomienda revisar esta memoria del I.P.

Al generarse este mensaje, la ejecución del programa se detiene por completo. Para reiniciarlo deberá oprimirse el botón del RESET.

10.- Se muestra al usuario la información que ha dado para que verifique si la dió correctamente, para así proceder a hacer las pruebas pertinentes al C.I.D., o si hubo algunos errores los pueda corregir antes de emprender las pruebas al C.I.D. en cuestión.

La información que se presenta al usuario es igual a la forma como se numeran las terminales o patitas de los C.I.D. de 14, 16, 24 y 40 pines. Los lugares donde no aparece el valor lógico, representan al Vcc o al GND o a las salidas que tiene el C.I.D. a probar.

Para este algoritmo simulamos la prueba de un C.I.D. cuádruple NAND de dos entradas cada compuerta. Las entradas están en las terminales 1 y 2, 4 y 5, 9 y 10, y 12 y 13. Las salidas son las patitas 3, 6, 8 y 11. El Vcc está en la 14 y el GND en la 7.

Los mensajes que se dan al usuario son:

10.1. - LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	
2.....	0	13.....	0
3.....		12.....	0
4.....	0	11.....	
5.....	0	10.....	0
6.....		9.....	0
7.....		8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

Si la respuesta es NO, el control pasa al punto 11. Si la respuesta es afirmativa, entonces:

10.2. - PARA FINALIZAR OPRIMIR TECLA F.

10.3. - PRIMERO DAR NUMERO DE PATITA. LUEGO DAR ESPACIO (CR) O VALOR LOGICO, SEGUN CONVENGA.

LAS SALIDAS DEL CIRCUITO EN PRUEBA DEBEN PONERSE EN BLANCO (DAR CR O ESPACIO).

PATITA NUMERO

Una vez dado el número de la terminal, se le pide al usuario que dé el nuevo valor lógico que deberá tener dicha patita o si es salida que la 'borre' con un espacio o un CR (ENTER), o sea:

10.4. - ESCRIBIR UN CERO O UN UNO. O UN ESPACIO O UN (CR)

Una vez dado el valor debido el control pasa nuevamente al punto 10.2 y así sucesivamente hasta que el usuario oprima la tecla 'F', con lo cual se llega al punto 10.1 otra vez, para que de esta forma el usuario verifique si todos los valores son correctos o hay que hacer alguna otra modificación, para lo cual volveríamos al punto 10.2 y siguientes. Si todo está correcto, entonces se responde con cualquier tecla distinta de la 'S' y así pasar al punto número 11.

Si se oprimió una tecla no numérica, el mensaje que se envía es el siguiente:

10.5. - CARACTER INAPROPIADO.

y se queda en este paso hasta que el usuario dé los caracteres apropiados u oprima la tecla del ENTER o la letra F de terminación o Fin, y regresar al punto 10.1 ó 10.2 ó 10.3 ó 10.4, según donde se haya cometido la falta, si dió el ENTER o la F; o pasar al punto 11 si escribió el o los números adecuados.

11.- ASEGURARSE QUE ESTE CORRECTAMENTE COLOCADO EL CIRCUITO A PROBAR.

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

Mientras la respuesta sea negativa, el punto 11 se repetirá.

Al momento de responder afirmativamente el control pasará al punto 12.

12. - SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

Si la respuesta es negativa, se pasa al punto número 13. Si la respuesta es afirmativa, se realiza la prueba y los resultados son mostrados al usuario de la manera siguiente:

12.1. - RESULTADOS DE LA PRUEBA.

12.2. - LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	1
2.....	0	13.....	0
3.....	1	12.....	0
4.....	0	11.....	1
5.....	0	10.....	0
6.....	1	9.....	0
7.....	0	8.....	1

12.3. - SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

Si la respuesta es negativa, se pasa al punto número 13. Si la respuesta es afirmativa, entonces se da al usuario la información anotada en los puntos 10.1 a 10.4 para que las salidas del C.I.D. queden en blanco. Una vez que el usuario oprima la tecla 'F', el I.P. le escribe la información siguiente:

12.4. - LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	1
2.....	0	13.....	0
3.....		12.....	0
4.....	0	11.....	
5.....	0	10.....	0
6.....		9.....	0
7.....	0	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

Si hubo algún error en la información dada, entonces se realizan los pasos marcados por los puntos 10.1 a 10.5 y una vez corregidos los errores, o si la respuesta es negativa, se continua con:

12.5. - SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

Si la respuesta es negativa, se pasa al punto número 13. Si la respuesta es afirmativa se vuelven a resolver los puntos 12.1

en adelante, hasta que el usuario quiera pasar al punto número 13.

13. - UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S. CUALQUIER OTRA TECLA INDICARA: NO.

Si la respuesta es negativa el control se envía al punto número 14. Si la respuesta es afirmativa, entonces se realizan los puntos 10.1 a 10.5, para dejar en blanco las terminales correspondientes a las salidas del C.I.D. y para cambiar los valores lógicos a las patitas que son entradas al C.I.D. que se está probando y así poder realizar una prueba diferente al mismo C.I.D. Una vez cambiados los valores de las terminales pertinentes, se muestra al usuario cómo quedaron esos cambios para que decida si se procede a hacer la prueba o si se necesitan hacer más cambios, con lo cual regresamos a los puntos 10.1 a 10.5. Si se procede a hacer la prueba, entonces, se repiten los puntos 12 al 13 inclusive.

14. - SE PRUEBA OTRO CIRCUITO ?

RESPUESTA: SI=S. CUALQUIER OTRA TECLA INDICARA: NO.

Si la respuesta es negativa se pasa al punto número 15. Si la respuesta es afirmativa, entonces, el control pasará al punto número 1, o sea, es como si se encendiese por primera vez al I.P.

15. - TEPMINAR LA SESION ?

RESPUESTA: SI=S. CUALQUIER OTRA TECLA INDICARA: NO.

Si la respuesta es negativa, el control pasará al punto número 1.

Del algoritmo detallado en las páginas precedentes hablaré de los pros y contras que tiene.

Dicho algoritmo permite hacer las pruebas a cualquier C.I.D. que contemple la tecnología TTL o compatible a ésta. Además, se pueden probar C.I.D. que tengan entre 14, 16, 24, ó 40 terminales y todo esto da un margen sumamente amplio de C.I.D. que pueden probarse con este I.P., lo cual es muy conveniente para tenerlo en un laboratorio, taller, fábrica, etc., en donde se necesiten hacer este tipo de pruebas y donde el personal que lo utilizará tenga los conocimientos necesarios y suficientes en electrónica.

Variando el software, esto es, creando otro programa adecuado, este mismo instrumento programable se puede convertir en un instrumento programado, o sea, un instrumento que sólo requerirá que el usuario dé la nomenclatura correcta del C.I.D. que desee

probar, conecte adecuadamente al C.I.D. en el socket o base correspondiente y conecte las señales de GND y Vcc apropiadamente, para que el instrumento programado verifique si tiene las rutinas de prueba pertinentes para el C.I.D. de la nomenclatura dada por el usuario y pueda realizar por sí sólo las pruebas al C.I.D. en cuestión y, una vez hechas éstas, dé los resultados al usuario en la terminal o monitor: Funciona Correctamente, o, No Funciona Correctamente, por ejemplo. Este instrumento programado es el que podría comercializarse.

Otro pro que tiene este I.P. es que puede utilizarse en la docencia, puesto que cada una de sus tarjetas puede servir como un módulo o tema dentro de un curso semestral en la materia de Máquinas Digitales con Laboratorio, por ejemplo.

Un punto que considero muy importante, y que da mucha fuerza a este I.P., es que puede ser tan versátil como interfaces de control diferentes se diseñen, según los requerimientos y, desde luego, con la programación correspondiente, debido a su amplia capacidad de almacenamiento en memoria EPROM. Como un ejemplo a esto último sería tener a este I.P. con las interfaces, transductores y programación apropiados para controlar diferentes puntos, cosas, aparatos, etc. dentro de una casa, como sería: la lavadora, la cafetera, el tostador de pan, la estufa, el horno (normal o de microondas), las luces de toda la casa o de sólo determinados lugares, etc.; de todos ellos el usuario podría programar su servicio, o sólo algunos, de acuerdo a los requerimientos que disponga o desee. Algo similar podría hacerse en un laboratorio o en un taller o en una fábrica, etc.

Otra de las ventajas que tiene este algoritmo es que va llevando de la 'mano' al usuario (pero repito, debe tener amplios conocimientos en electrónica), con lo cual uno podría prescindir de un manual para la operación de este I.P.

En cuanto a los contras que tiene el I.P. para probar diferentes C.I.D. tenemos dos principales: uno en hardware y el otro en software. El inconveniente que hay en hardware es el tener que conectar en forma externa y manualmente la alimentación al C.I.D. que se desee probar, esto es, el Vcc y el GND. Desde un principio había contemplado la forma de conectar el Vcc y GND por medio de hardware y software, pero en la práctica no funcionó muy bien, por lo que se descartó y se optó por construir el I.P. como se ha descrito.

Con respecto al inconveniente en software, tenemos que el usuario debe de estar dando los pulsos de reloj para que el C.I.D. secuencial que esté probando pueda funcionar apropiadamente. Esto se puede solucionar por software también, pero se hizo a propósito así para que el usuario siga con más detenimiento el comportamiento del C.I.D. secuencial que desee probar. Esto se aprecia mejor en el capítulo 'PRUEBAS Y RESULTADOS' en donde se da un ejemplo de cómo se prueba un C.I.D. secuencial y de cómo el usuario va dando las señales de reloj apropiadas.

Otra desventaja de este algoritmo es que el usuario debe poner en blanco a cada terminal que sea salida del C.I.D. que se vaya a probar. Aunque para cada prueba que se vaya a realizar se le informe al usuario de hacerlo, no deja de ser algo hastiante, por lo cual este hecho se considera como inconveniente. Desde luego que se puede corregir, y se hará en una versión futura, pero para este caso se tendría que modificar bastante el programa ya hecho.

El algoritmo presentado fué pensado para interactuar con una pantalla, de ahí el porqué se repiten muchos mensajes, lo cual no molesta mucho al usuario. Pero si se desea imprimir lo que aparece en pantalla, entonces, este algoritmo es pésimo, por lo cual no es recomendable hacer impresiones salvo para el caso que se necesite ver cierta información de las memorias que se prueben (ROM, PROM, EPROM, por ejemplo). Para todos los demás C.I.D. no se recomienda hacer impresiones de las pantallas que el I.P. dé.

5. - PRUEBAS Y RESULTADOS.

En esta parte se escribe una copia de dos pruebas hechas a dos diferentes C.I.D., a saber: una prueba consistirá en ver que opere correctamente un cuádruple NAND con dos entradas en cada una de las cuatro compuertas NAND, y la otra prueba es para un circuito secuencial que tiene dos Flip-Flops J-K. En el apéndice A se da una lista de todos los C.I.D. que se probaron en el laboratorio de Cibernética *Alejandro Medina*.

Para hacer más ágil la lectura no procederé a dar cada uno de los pasos que se vieron en el capítulo precedente, sino que primero daré la información de lo que se debe hacer y luego mostraré el resultado que se obtiene en la pantalla, después de haber efectuado algún o algunos cambios o de haber realizado la prueba al C.I.D. en cuestión. De esta forma evito poner todos los mensajes que da el I.P. y que haría muy tediosa la lectura de esta parte de la tesis, puesto que no es lo mismo escrito en papel que leer en pantalla y estar interactuando con lo expuesto en ésta.

El primer C.I.D. del cual obtendremos los resultados de las pruebas que se le hagan, es un SN74LS00 Cuádruple NAND, o sea, cuatro compuertas NAND de dos entradas y una salida cada una, como se puede apreciar en la figura 5.1. Las terminales de entrada son las numeradas como: 1 y 2, 4 y 5, 9 y 10, 12 y 13, (ver figura 5.1) para cada una de las cuatro compuertas y sus salidas son las numeradas como: 3, 6, 8 y 11 (ver figura 5.1). La terminal 7 es donde se conecta el común o tierra (GND) de la fuente, y en la terminal 14 el Vcc; estas dos últimas terminales se conectan en forma externa y manualmente. Las pruebas que se harán a cada una de las cuatro compuertas son las mostradas en la tabla de verdad 5.1. Los valores mostrados en la columna ENTRADAS son los que el usuario debe proporcionar al I.P. para que, al realizar la prueba y leer los resultados, dé las respuestas mostradas en la columna SALIDA y el usuario compare si son iguales, con lo cual, el C.I.D. está funcionando bien, si no, dicho C.I.D. estará funcionando mal.

Supongamos que se ha dado el valor de cero a cada una de las entradas de cada una de las cuatro compuertas NAND, con lo cual el I.P. nos dará la primera pantalla como:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	
2.....	0	13.....	0
3.....		12.....	0
4.....	0	11.....	
5.....	0	10.....	0
6.....		9.....	0
7.....		8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

ASEGURARSE QUE ESTE CORRECTAMENTE COLOCADO EL CIRCUITO A PROBAR.

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

Antes de proceder a dar los resultados cabe observar que todas las terminales que son salidas: 3, 6, 8 y 11 están en blanco y hay que mencionar que antes de realizar cualquier prueba, todas las terminales que sean salidas deberán ponerse en blanco porque de lo contrario el I.P. las considerará como entradas en lugar de salidas, y las lecturas así obtenidas serán erróneas. Por lo cual, en cada prueba que se vaya a realizar, al usuario siempre se le informa que debe de poner en blanco a las terminales que son salidas.

Las terminales 7 y 14 que son de la fuente de alimentación, GND y Vcc respectivamente, en la primera pantalla que se da de la información recibida, aparecerán en blanco. Una vez realizada la primera prueba, las terminales que son de fuente tendrán los valores lógicos correspondientes al GND (0-lógico) y al Vcc (1-lógico).

Una vez que el usuario oprimió la tecla 'S' a la pregunta de si se hace la prueba al C.I.D., el I.P. da la siguiente pantalla:

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

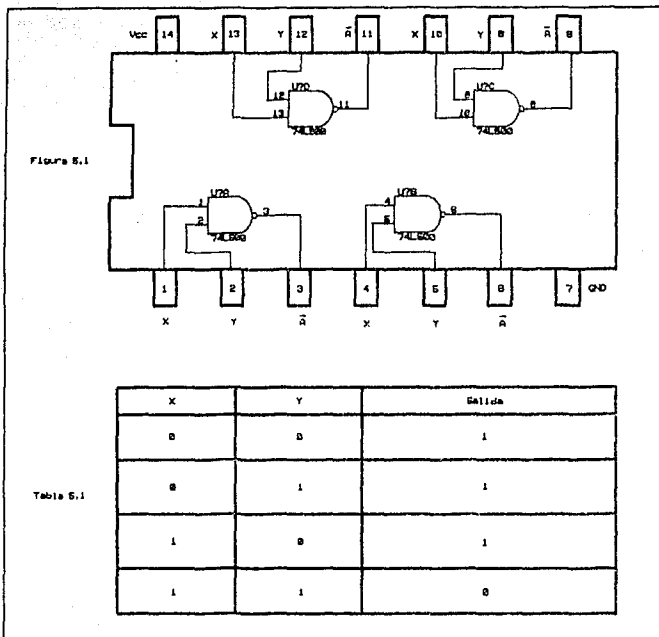
PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	1
2.....	0	13.....	0
3.....	1	12.....	0
4.....	0	11.....	1
5.....	0	10.....	0
6.....	1	9.....	0
7.....	0	8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

Aqui podemos apreciar que la respuesta en cada una de las cuatro computas es la correcta: terminales 3, 6, 8, y 11 tienen un 1-lógico, que es lo esperado para entradas con valor de 0-lógico, como se puede comparar con la tabla 5.1.



18

18 **FIGURA 5.1:** Diagrama del conexionamiento de cada compuerta NAND (74LS00)

TABLA 5.1: Tabla de Verdad de cada compuerta NAND (74LS00).

Como no queremos repetir la prueba sino realizar otra, se escribio la 'N' para así pasar al siguiente mensaje que da el I.P.:

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

Con esta respuesta el programa limpia la pantalla y reescribe la información dada en la pantalla líneas arriba, envía los mensajes que indican cómo salirse una vez dados los valores que realizarán una prueba diferente y que el usuario debe poner en blanco todas las terminales que sean salidas del C.I.D. a probar y cómo debe hacerlo. Una vez que el usuario terminó de dar los datos adecuados para realizar otra prueba, diferente a la anterior, el I.P. da la siguiente información:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	1
2.....	1	13.....	0
3.....		12.....	1
4.....	0	11.....	
5.....	1	10.....	0
6.....		9.....	1
7.....	0	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

Para agilizar, sólo voy a poner las pantallas correspondientes a las pruebas que faltan para terminar de probar al C.I.D., así como las pantallas de resultados que surgen de cada prueba, o sea:

RESULTADOS DE LA PRUEBA (última pantalla dada líneas arriba).

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	1
2.....	1	13.....	0
3.....	1	12.....	1
4.....	0	11.....	1
5.....	1	10.....	0
6.....	1	9.....	1
7.....	0	8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

Una vez hechos los cambios pertinentes, se tiene:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	1	14.....	1
2.....	0	13.....	1
3.....		12.....	0
4.....	1	11.....	
5.....	0	10.....	1
6.....		9.....	0
7.....	0	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	1	14.....	1
2.....	0	13.....	1
3.....	1	12.....	0
4.....	1	11.....	1
5.....	0	10.....	1
6.....	1	9.....	0
7.....	0	8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

Una vez realizados los cambios pertinentes para la última prueba,
se tiene:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	1	14.....	1
2.....	1	13.....	1
3.....		12.....	1
4.....	1	11.....	
5.....	1	10.....	1
6.....		9.....	1
7.....	0	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	1	14.....	1
2.....	1	13.....	1
3.....	0	12.....	1
4.....	1	11.....	0
5.....	1	10.....	1
6.....	0	9.....	1
7.....	0	8.....	0

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

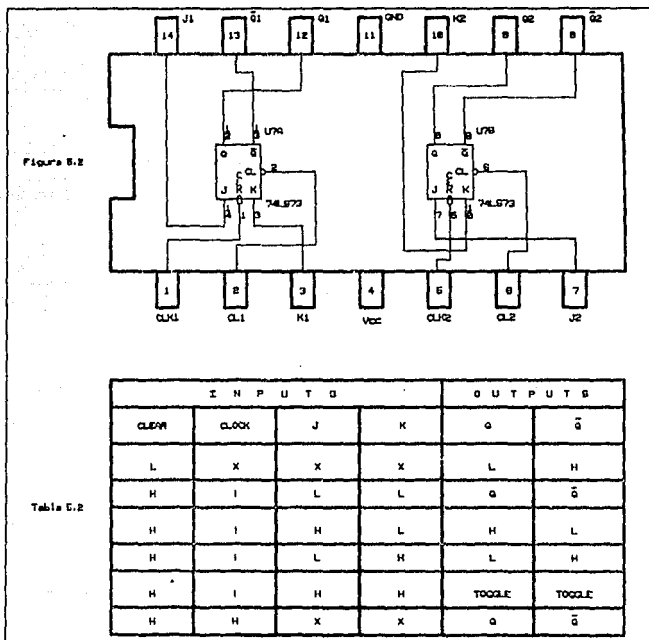
SE PRUEBA OTRO CIRCUITO ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

TERMINAR LA SESION ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

Ahora pasemos a probar al C.I.D. con nomenclatura SN7473, que es un doble Flip-Flop J-K con señal de Clear (o borrado) y señal de Clock (o reloj) para borrar y sincronizar, respectivamente, a cada uno de los dos Flip-Flops J-K. En la figura 5.2 se da la constitución de este C.I.D. y en la tabla 5.2 se pueden apreciar las señales que dan lugar a la operación de este par de flip-flops y de donde obtendremos la información para hacer las pruebas



19

¹⁹ **FIGURA 5.2:** Diagrama del conexionamiento de cada Flip-Flop J-K (74LS73).

TABLA 5.2: Tabla de Verdad para cada Flip-Flop J-K (74LS73).

pertinentes a este C.I.D., y donde podremos comparar los resultados de dichas pruebas que va a hacer el I.P.

Suponiendo que hemos dado ya toda la información adecuada al I.P. para realizar la primera prueba, tendremos como pantalla inicial lo siguiente:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	0
2.....	0	13.....	
3.....	0	12.....	
4.....		11.....	
5.....	0	10.....	0
6.....	0	9.....	
7.....	0	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

ASEGURARSE QUE ESTE CORRECTAMENTE COLOCADO EL CIRCUITO A PROBAR.

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

Recordar que por ser la primera pantalla antes de realizar la primera prueba, las señales de Vcc y GND están en blanco (terminales 4 y 11 para este C.I.D.) y que las demás patitas en blanco son las salidas Q's (afirmadas y negadas) de cada uno de los dos flip-flops J-K. Ver figura 5.2.

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	0
2.....	0	13.....	1
3.....	0	12.....	0
4.....	1	11.....	0
5.....	0	10.....	0
6.....	0	9.....	0
7.....	0	8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

De acuerdo a la tabla 5.2, los resultados obtenidos son los correctos, esto es, las Q's complementadas, terminales 13 y 8, deben tener un 1-lógico y las Q's afirmadas, terminales 9 y 12, un cero lógico; mientras la señal del Clear, terminales 2 y 6, estén en cero lógico e independiente del valor que tengan las entradas J y K, así como de la señal de reloj, terminales 1 y 5. Según la tabla 5.2, los flip-flops J-K realizan sus transiciones cuando detectan la bajada en las señales de reloj, por lo cual, para la siguiente prueba pondremos en uno lógico a las señales de reloj, así como también, a las señales del Clear (CLR) de ambos flip-flops y los resultados deberán ser los mismos que los señalados en la pantalla anterior, por lo tanto:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	1	14.....	0
2.....	1	13.....	
3.....	0	12.....	
4.....	1	11.....	0
5.....	1	10.....	0
6.....	1	9.....	
7.....	0	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

X

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	1	14.....	0
2.....	1	13.....	1
3.....	0	12.....	0
4.....	1	11.....	0
5.....	1	10.....	0
6.....	1	9.....	0
7.....	0	8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

Como era de esperarse, las Q's no sufrieron cambio alguno. Ahora pondremos a las señales de reloj con ceros lógicos, terminales 1 y 5, para que los flip-flops realicen su transición y mantendremos sin modificar las señales de CLR, terminales 2 y 6, en uno lógico de acuerdo a la tabla 5.2, con lo cual observaremos que no deberá haber cambio alguno en las salidas Q's de este C.I.D. Una vez hechas las modificaciones señaladas, se obtendrá la siguiente pantalla:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	0
2.....	1	13.....	
3.....	0	12.....	
4.....	1	11.....	0
5.....	0	10.....	0
6.....	1	9.....	
7.....	0	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	0
2.....	1	13.....	1
3.....	0	12.....	0
4.....	1	11.....	0
5.....	0	10.....	0
6.....	1	9.....	0
7.....	0	8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

Ahora activemos a los flip-flops J-K dando la señal de uno lógico a las terminales que representan a la J, 14 y 7 según la figura

5.2, y demos el mismo valor lógico a las señales de reloj, con lo cual las salidas Q's deberán permanecer sin cambio alguno puesto que la transición sólo debe realizarse cuando se detecta la bajada en el pulso (o en los pulsos) de reloj, por lo tanto, tendremos las dos siguientes pantallas:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	1	14.....	1
2.....	1	13.....	
3.....	0	12.....	
4.....	1	11.....	0
5.....	1	10.....	0
6.....	1	9.....	
7.....	1	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	1	14.....	1
2.....	1	13.....	1
3.....	0	12.....	0
4.....	1	11.....	0
5.....	1	10.....	0
6.....	1	9.....	0
7.....	1	8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

En el siguiente par de pantallas que se muestran se verá, en la primera, el cambio en las señales de reloj de uno lógico a cero lógico, con lo cual se debe realizar la transición en los flip-flops J-K, de 0 a 1 lógicos, en las terminales 9 y 12, y de 1 a 0 lógicos en las terminales 8 y 13 (ver la figura 5.2), por lo que:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	1
2.....	1	13.....	
3.....	0	12.....	
4.....	1	11.....	0
5.....	0	10.....	0
6.....	1	9.....	
7.....	1	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO,
N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO,
S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	1
2.....	1	13.....	0
3.....	0	12.....	1
4.....	1	11.....	0
5.....	0	10.....	0
6.....	1	9.....	1
7.....	1	8.....	0

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO,
N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO,
S

La siguiente prueba que se realizará será la de poner en unos lógicos a las terminales de reloj, 1 y 5, a las terminales de las J, 7 y 14, en ceros lógicos, y las terminales 3 y 10, las K, en unos lógicos, para así obtener las dos siguientes pantallas:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO
1.....	1
2.....	1
3.....	1
4.....	1
5.....	1
6.....	1
7.....	0

PATITA NUMERO	VALOR LOGICO
14.....	0
13.....	
12.....	
11.....	0
10.....	1
9.....	
8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO
1.....	1
2.....	1
3.....	1
4.....	1
5.....	1
6.....	1
7.....	0

PATITA NUMERO	VALOR LOGICO
14.....	0
13.....	0
12.....	1
11.....	0
10.....	1
9.....	1
8.....	0

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

El lector se preguntará que algo anda mal puesto que no se produjo la transición de unos a ceros lógicos en las terminales de salida Q's, 9 y 12; pero realmente está perfectamente bien esta respuesta del I.P. puesto que la realización de las transiciones se llevan al cabo cuando se detecta la bajada en el pulso de reloj, terminales 1 y 5, como se puede ver en la figura 5.2, y como se puede apreciar, los relojes contienen un uno lógico. Ahora procedemos a cambiar los valores para las señales de reloj, de uno a cero lógicos para así obtener la respuesta correcta, como se puede apreciar en las siguientes dos pantallas:

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	0
2.....	1	13.....	
3.....	1	12.....	
4.....	1	11.....	0
5.....	0	10.....	1
6.....	1	9.....	
7.....	0	8.....	

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	0
2.....	1	13.....	1
3.....	1	12.....	0
4.....	1	11.....	0
5.....	0	10.....	1
6.....	1	9.....	0
7.....	0	8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

A continuación se llevará al cabo la prueba conocida como 'TOGGLE' en los flip-flops J-K y significa que las salidas Q's van a estar conmutando su valor lógico, es decir, cambiando de ceros a unos lógicos y viceversa, con cada pulso de reloj y cuando ambas entradas, J y K, estén conectadas al valor uno lógico. Las dos primeras pantallas que se dan a continuación, mostrarán que no hay transición alguna debido a que las señales de reloj, terminales 1 y 5, están en uno lógico. Las dos siguientes pantallas mostrarán la transición de cada flip-flop puesto que las señales de reloj han cambiado de uno a cero lógicos. Estas cuatro pantallas se repetirán tantas veces como el usuario lo desee, por lo que no tiene caso repetir las, y con una sola vez que se pongan bastará.

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO
1.....	1
2.....	1
3.....	1
4.....	1
5.....	1
6.....	1
7.....	1

PATITA NUMERO	VALOR LOGICO
14.....	1
13.....	1
12.....	1
11.....	0
10.....	1
9.....	1
8.....	1

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO
1.....	1
2.....	1
3.....	1
4.....	1
5.....	1
6.....	1
7.....	1

PATITA NUMERO	VALOR LOGICO
14.....	1
13.....	1
12.....	0
11.....	0
10.....	1
9.....	0
8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
S

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO
1.....	0
2.....	1
3.....	1
4.....	1
5.....	0
6.....	1
7.....	1

PATITA NUMERO	VALOR LOGICO
14.....	1
13.....	1
12.....	1
11.....	0
10.....	1
9.....	1
8.....	1

HUBO ALGUN ERROR EN LA TABLA DADA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.
N

SE PROCEDE A HACER LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	0	14.....	1
2.....	1	13.....	0
3.....	1	12.....	1
4.....	1	11.....	0
5.....	0	10.....	1
6.....	1	9.....	1
7.....	1	8.....	0

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

Vamos a tomar estos últimos resultados de la prueba para realizar otra prueba que consistirá en dejar con un valor fijo de uno lógico a la terminal 1, que es el reloj de uno de los dos flip-flops J-K, e ir variando la señal de reloj de la terminal 5, o sea, del otro flip-flop y ver que resultados nos da el I.P. Para lo cual sólo pondremos la pantalla que nos dé el resultado (hacer comparaciones con la pantalla anterior, la dada líneas arriba), a saber:

RESULTADOS DE LA PRUEBA

LA INFORMACION RECIBIDA ES LA SIGUIENTE:

PATITA NUMERO	VALOR LOGICO	PATITA NUMERO	VALOR LOGICO
1.....	1	14.....	1
2.....	1	13.....	0
3.....	1	12.....	1
4.....	1	11.....	0
5.....	0	10.....	1
6.....	1	9.....	0
7.....	1	8.....	1

SE REPITE LA PRUEBA ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

N

UNA PRUEBA DIFERENTE ?

RESPUESTA: SI=S, CUALQUIER OTRA TECLA INDICARA: NO.

S

La respuesta es correcta puesto que la salida Q de la terminal 9 cambió del uno al cero lógicos, mientras que la salida Q de la terminal 12 permaneció inalterada, debido a que el C.I.D. no detectó la bajada en el pulso de reloj de la terminal 1, por mantenerla fija en el uno lógico.

Con las pruebas hechas podemos concluir que el C.I.D. SN7473 está funcionando correctamente y, además, se corrobora que este programa es sumamente tedioso cuando se trata de probar C.I.D. secuenciales, esto es, que tengan una señal de reloj, ya que hay que simularla en cada paso de cada prueba que se haga. Afortunadamente, no es tan tedioso al trabajarlo con un monitor puesto que las pantallas se despliegan con cierta rapidez; lo pesado es cuando se tiene que imprimir cada pantalla.

6.- CONCLUSIONES.

Concluiré esta tesis enumerando las ventajas y las desventajas que tiene este I.P. con lo cual uno se podrá dar cuenta de las mejoras que se le pueden hacer y mencionaré la posibilidad de tener casi cualquier tipo de instrumento, completamente diferente a éste, con sólo construir las interfases apropiadas y la programación que las manejen.

De las ventajas tenemos:

- 1.- Poder comprobar tantas veces como uno quiera si el C.I.D. está operando adecuadamente o no, con una misma prueba o con diferentes pruebas para el mismo C.I.D.
- 2.- Poder probar tantos C.I.D. diferentes de tecnología TTL o compatibles, como uno quiera, sin la necesidad de estar haciendo el alambrado correspondiente para cada C.I.D. diferente. Sólo se hace el alambrado de V_{cc} y GND, lo cual no da mucho problema.
- 3.- Tiene la ventaja de poderse utilizar dentro de la docencia, donde cada tarjeta sería un módulo o tema dentro de un curso.
- 4.- Construyendo interfases apropiadas y la programación para manejarlas, este I.P. puede utilizarse para realizar prácticas (en mecánica clásica, por ejemplo) o como una ayuda a éstas; o para controlar uno o varios experimentos, dentro de un laboratorio de docencia o de investigación; o de controlar uno o varios procesos dentro de alguna fábrica o industria, etc.
- 5.- Puede probar C.I.D. que tengan un total de terminales de 14, 16, 24 ó 40, dando un margen de C.I.D. a probar sumamente amplio; cosa que otros instrumentos parecidos a éste sólo están dedicados a un número fijo de terminales y no tan variado como el que tiene este I.P.
- 6.- Este I.P. puede implementarse en dos formas o versiones: La Programable, que es la que se discutió en esta tesis; y la Programada o Automática, que fué presentada el 5 de Abril de 1990 en la tesis que dirigí (bibliografía 12). Estas dos versiones se pueden también utilizar para lo expuesto en el punto número 4 de estas conclusiones.
- 7.- Se tiene la facilidad de aumentar la memoria EPROM y la memoria RAM hasta completar la capacidad máxima de direccionamiento que tiene el microprocesador de Motorola, MC6800, usado en este I.P., que es de 64 Kbytes. Ver figura 6.1.
- 8.- La comunicación que se entabla entre el usuario y este I.P.,

hace casi innecesario el uso de un manual para la operación de este I.P. Pero, vuelvo a repetir, el usuario debe tener amplios conocimientos en electrónica para usar esta versión de I.P.

De las desventajas tenemos:

1.- Tiene la desventaja de que al finalizar cada prueba, el usuario debe 'borrar' los valores que se obtuvieron en la o las salidas que presente el C.I.D.

2.- Tiene la desventaja de que el usuario debe simular los pulsos de reloj para los C.I.D. síncronos que se deseen probar.

3.- El Vcc y el GND deben conectarse externamente y manualmente, esto es, el usuario debe hacer el conexionamiento.

4.- Sólo puede probarse un C.I.D. a la vez.

APENDICE A.

Las referencias que se utilizaron para realizar las pruebas a todos los circuitos integrados digitales que se ennumeran más adelante, y se indica la página donde se encuentra cada uno de éstos, son los que están en la bibliografía con los números: 5 (para probar las PIA's), 32 (para probar las memorias EPROM 2716) y 8 (para el resto de los circuitos integrados digitales).

A continuación se enlista a los diferentes circuitos integrados digitales que se probaron dentro del Laboratorio de Cibernética *Alejandro Medina:*

- 1.- SN74LS00N QUADRUPLE 2-INPUT POSITIVE-NAND GATES.
Positive Logic: $Y = \overline{AB}$ Página 5-6. (BIBLIO. # 8).
- 2.- DM7402N QUADRUPLE 2-INPUT POSITIVE-NOR GATES.
Positive Logic: $Y = A+B$ Página 5-6. (BIBLIO. # 8).
- 3.- SN7404J HEX INVERTERS.
Positive Logic: $Y = \overline{A}$ Página 5-7. (BIBLIO. # 8).
- 4.- SN7408N QUADRUPLE 2-INPUT POSITIVE-AND GATES.
Positive Logic: $Y = AB$ Página 5-8. (BIBLIO. # 8).
- 5.- SN7410N TRIPLE 3-INPUT POSITIVE-NAND GATES.
Positive Logic: $Y = \overline{ABC}$ Página 5-8. (BIBLIO. # 8).
- 6.- SN74LS11N TRIPLE 3-INPUT POSITIVE-AND GATES.
Positive Logic: $Y = ABC$ Página 5-9. (BIBLIO. # 8).
- 7.- SN7413N DUAL 4-INPUT POSITIVE-NAND SCHMITT TRIGGERS.
Positive Logic: $Y = \overline{ABCD}$ Página 5-9. (BIBLIO. # 8).
- 8.- SN7414N HEX SCHMITT-TRIGGER INVERTERS.
Positive Logic: $Y = \overline{A}$ Página 5-9. (BIBLIO. # 8).
- 9.- DM74LS20N DUAL 4-INPUT POSITIVE-NAND GATES.
Positive Logic: $Y = \overline{ABCD}$ Página 5-10. (BIBLIO. # 8).
- 10.- 74LS21E DUAL 4-INPUT POSITIVE-AND GATES.
Positive Logic: $Y = ABCD$ Página 5-11. (BIBLIO. # 8).

- 11.- SN7425N DUAL 4-INPUT POSITIVE-NOR GATES WITH STROBE.
Positive Logic: $Y = \overline{G(A+B+C+D)}$ Página 5-11. (BIBLIO. # 8).
- 12.- SW7427N TRIPLE 3-INPUT POSITIVE-NOR GATES.
Positive Logic: $Y = \overline{A+B+C}$ Página 5-12. (BIBLIO. # 8).
- 13.- SN7430N 8-INPUT POSITIVE-NAND GATES.
Positive Logic: $Y = \overline{ABCDEFGH}$ Página 5-12. (BIBLIO. # 8).
- 14.- SN74S30N 8-INPUT POSITIVE-NAND GATES.
Positive Logic: $Y = \overline{ABCDEFGH}$ Página 5-12. (BIBLIO. # 8).
- 15.- DM74LS32N QUADRUPLE 2-INPUT POSITIVE-OR GATES.
Positive Logic: $Y = A+B$ Página 5-13. (BIBLIO. # 8).
- 16.- SN7437N QUADRUPLE 2-INPUT POSITIVE-NAND BUFFERS.
Positive Logic: $Y = \overline{AB}$ Página 5-13. (BIBLIO. # 8).
- 17.- SN7440N DUAL 4-INPUT POSITIVE-NAND BUFFERS.
Positive Logic: $Y = \overline{ABCD}$ Página 5-14. (BIBLIO. # 8).
- 18.- SN7450N DUAL 2-WIDE 2-INPUT AND-OR-INVERT GATES (ONE GATE EXPANDABLE).
Positive Logic: $Y = \overline{AB+CD+X}$ Página 5-16. (BIBLIO. # 8).
- 19.- F7451PC AND-OR-INVERT GATES.
Positive Logic: $Y = \overline{AB+CD}$ Página 5-16. (BIBLIO. # 8).
- 20.- SN7454N 4-WIDE AND-OR-INVERT GATES.
Positive Logic: $Y = \overline{AB+CD+EF+GH}$ Página 5-18. (BIBLIO. # 8).
- 21.- SN7470N AND-GATED J-K POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR.
Positive Logic: $J = J1 \bullet J2 \bullet \overline{J}$
 $K = K1 \bullet K2 \bullet \overline{K}$ Página 5-21. (BIBLIO. # 8).
- 22.- SN74LS73AN DUAL J-K FLIP-FLOPS WITH CLEAR.
Página 5-22. (BIBLIO. # 8).
- 23.- F-7474PC DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR.
Página 5-22. (BIBLIO. # 8).
- 24.- SN7486N QUADRUPLE 2-INPUT EXCLUSIVE-OR GATES.
Positive Logic: $Y = A \oplus B = \overline{AB} + \overline{A\overline{B}}$ Página 5-26. (BIBLIO. # 8).
- 25.- SN7490AN DECADE COUNTERS. DIVIDE-BY-TWO AND DIVIDE-BY FIVE.
Página 5-27. (BIBLIO. # 8).

26. - DM7491AN 8-BIT SHIFT REGISTERS. SERIAL-IN, SERIAL-CUT
GATED INPUT.
Página 5-28. (BIBLIO. # 8).
27. - SN7492AN DIVIDE-BY-TWELVE COUNTERS. DIVIDE-BY-TWO AND
DIVIDE-BY-SIX.
Página 5-28. (BIBLIO. # 8).
28. - SN74LS93H 4-BIT BINARY COUNTERS. DIVIDE-BY-TWO AND
DIVIDE-BY-EIGHT.
Página 5-28. (BIBLIO. # 8).
29. - SN7495AN 4-BIT SHIFT REGISTERS. PARALLEL IN/PARALLEL OUT
SHIFT REGISTER, SHIFT LEFT SERIAL INPUT.
Página 5-29. (BIBLIO. # 8).
30. - SN74LS107AN DUAL J-K FLIP-FLOPS WITH CLEAR.
Página 5-32. (BIBLIO. # 8).
31. - SN74LS125AN QUADRUPLE BUS BUFFER GATES WITH THREE STATE
OUTPUTS.
Positive Logic: Y=A Página 5-37. (BIBLIO. # 8).
32. - SN74164N 8-BIT PARALLEL OUTPUT SERIAL SHIFT REGISTERS.
ASYNCHRONOUS CLEAR.
Página 5-44. (BIBLIO. # 8).
33. - SN74LS164N 8-BIT PARALLEL OUTPUT SERIAL SHIFT REGISTERS.
ASYNCHRONOUS CLEAR.
Página 5-44. (BIBLIO. # 8).
34. - DM74176N PRESETABLE COUNTERS/LATCHES. DECADE
(BI-QUINARY).
Página 5-47. (BIBLIO. # 8).
35. - DM74180N 9-BIT ODD/EVEN PARITY GENERATORS/CHECKERS.
Página 5-47. (BIBLIO. # 8).
36. - DM74LS196N PRESETABLE COUNTERS/LATCHES.
DECADE/BI-QUINARY.
Página 5-51. (BIBLIO. # 8).
37. - DM74197N PRESETABLE COUNTERS/LATCHES. BINARY.
Página 5-51. (BIBLIO. # 8).

Estos han sido los circuitos integrados digitales que se han probado, constando cada uno de ellos de 14 terminales.

38. - 7442N 4 LINE-TO-10-LINE DECODERS. BCD-TO-DECIMAL.
 Página 5-14. (BIBLIO. # 8).
39. - DM74LS48N BCD-TO-SEVEN-SEGMENT DECODERS/DRIVERS. INTERNAL
 PULL-UP OUTPUTS.
 Página 5-15. (BIBLIO. # 8).
40. - SN7476N DUAL J-K FLIP-FLOPS WITH PRESET AND CLEAR.
 Página 5-23. (BIBLIO. # 8).
41. - SN7483AN 4-BIT BINARY FULL ADDERS WITH FAST CARRY.
 Página 5-25. (BIBLIO. # 8).
42. - SN7485N 4-BIT MAGNITUDE COMPARATORS.
 Página 5-26. (BIBLIO. # 8).
43. - SN7494N 4-BIT SHIFT REGISTERS. DUAL ASYNCHRONOUS PRESETS.
 Página 5-29. (BIBLIO. # 8).
44. - DM7496N 5-BIT SHIFT REGISTERS. ASYNCHRONOUS PRESET.
 Página 5-29. (BIBLIO. # 8).
45. - SN74109N DUAL J-K POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH
 PRESET AND CLEAR.
 Página 5-33. (BIBLIO. # 8).
46. - 74LS112N DUAL J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS WITH
 PRESET AND CLEAR.
 Página 5-34. (BIBLIO. # 8).
47. - SN74LS133N 13-INPUT POSITIVE-NAND GATES.
 Positive Logic: Y=ABCDEFGHIJKLM Página 5-38. (BIBLIO. # 8).
48. - SN74S151N 1-OF-8 DATA SELECTORS/MULTIPLEXERS.
 Página 5-41. (BIBLIO. # 8).
49. - 74157N QUAD 2-TO 1-LINE DATA SELECTORS/MULTIPLEXERS.
 NONINVERTED DATA OUTPUTS.
 Página 5-43. (BIBLIO. # 8).
50. - 74LS160E SYNCHRONOUS 4-BIT COUNTERS. DECADE, DIRECT
 CLEAR.
 Página 5-43. (BIBLIO. # 8).
51. - SN74LS163AN SYNCHRONOUS 4-BIT COUNTERS. BINARY
 SYNCHRONOUS CLEAR.
 Página 5-43. (BIBLIO. # 8).
52. - SN74165N PARALLEL-LOAD 8-BIT SHIFT REGISTERS WITH
 COMPLEMENTARY OUTPUTS.
 Página 5-44. (BIBLIO. # 8).
53. - SN74166N 8-BIT SHIFT REGISTERS. PARALLEL/SERIAL INPUT.
 SERIAL OUTPUT.
 Página 5-44. (BIBLIO. # 8).

54. - SN74184H CODE CONVERTERS. CASCADABLE TO N-BITS.
BCD-TO-BINARY.
Página 5-48. (BIBLIO. # 8).
55. - DM74189N 64-BIT RANDOM-ACCESS MEMORIES. 16 4-BIT WORDS.
THREE STATE OUTPUTS.
Página 5-49. (BIBLIO. # 8).
56. - DM74LS190N SYNCHRONOUS UP/DOWN COUNTERS. BCD.
Página 5-50. (BIBLIO. # 8).
57. - DM74LS191N SYNCHRONOUS UP/DOWN COUNTERS. BINARY.
Página 5-50. (BIBLIO. # 8).
58. - DM74LS192N SYNCHRONOUS UP/DOWN DUAL CLOCK COUNTERS. BCD
WITH CLEAR.
Página 5-50. (BIBLIO. # 8).
59. - DM74193N SYNCHRONOUS UP/DOWN DUAL CLOCK COUNTERS. BINARY
WITH CLEAR.
Página 5-50. (BIBLIO. # 8).
60. - 74LS194AE 4-BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTERS.
Página 5-50. (BIBLIO. # 8).
61. - DM74195N 4-BIT PARALLEL-ACCESS SHIFT REGISTERS.
Página 5-51. (BIBLIO. # 8).
62. - 74279N QUAD \bar{S} - \bar{R} LATCHES.
Página 5-59. (BIBLIO. # 8).
63. - SN74S298N 256-BIT PROGRAMMABLE READ-ONLY MEMORIES. 32
8-BIT WORDS. 3-STATE OUTPUTS.
Página 5-61. (BIBLIO. # 8).
64. - SN74LS367AN HEX BUS DRIVERS. NONINVERTED 3-STATE OUTPUTS
ORGANIZED TO FACILITATE HANDLING OF 4-BIT DATA.
Página 5-66. (BIBLIO. # 8).
65. - SN74LS175N QUAD D-TYPE FLIP-FLOPS. COMPLEMENTARY OUTPUTS.
COMMON DIRECT CLEAR.
Página 5-46. (BIBLIO. # 8).

Estos han sido los circuitos integrados digitales que se han probado, constando cada uno de ellos de 16 terminales.

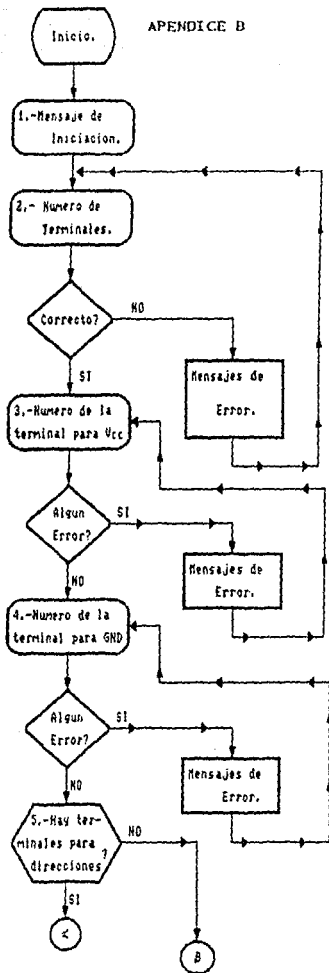
66. - SN74116N DUAL 4-BIT LATCHES.
Página 5-35. (BIBLIO. # 8).
67. - SN74143H COUNTERS/LATCHES/DECODERS/DRIVEPS. 15 mA.
CONSTANT CURRENT. 1-TO 5-V OUTPUT RANGE.
Página 5-40. (BIBLIO. # 8).
68. - DM74154H 4-LINE TO 16-LINE DECODERS/DEMULTIPLIXERS.
Página 5-42. (BIBLIO. # 8).
69. - SN74181H ARITHMETIC LOGIC UNITS/FUNCTION GENERATORS. 16
ARITHMETIC OPERATIONS. 16 LOGIC FUNCTIONS.
Página 5-48. (BIBLIO. # 8).
70. - DM74198N 8-BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTERS.
Página 5-51. (BIBLIO. # 8).
71. - SN74S412N MULTI-MODE BUFFERED 8-BIT LATCHES. 3-STATE
OUTPUTS. DIRECT CLEAR.
Página 5-71. (BIBLIO. # 8).
72. - MM2716Q ERASABLE PROGRAMMABLE READ-ONLY MEMORIES.
Páginas A4-18 a A4-22. (BIBLIO. # 32).

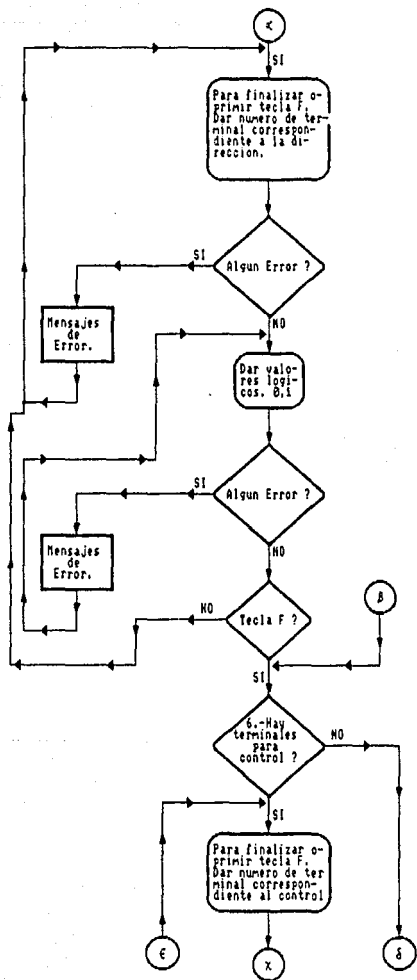
Estos han sido los circuitos integrados digitales que se han probado, constando cada uno de ellos de 24 terminales.

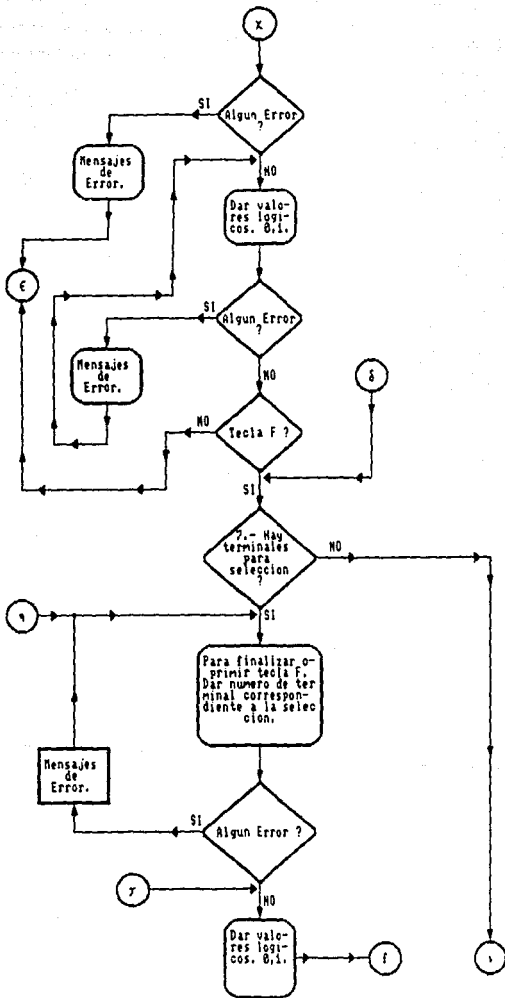
73. - S6820 PERIPHERAL INTERFACE ADAPTER (PIA).
Páginas 39 a 48. (BIBLIO. # 5).

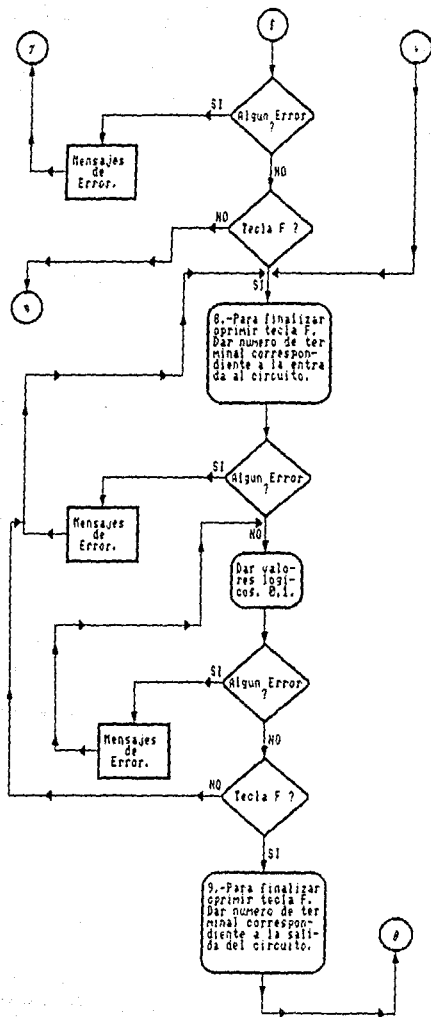
Es el único de 40 terminales: el PIA.

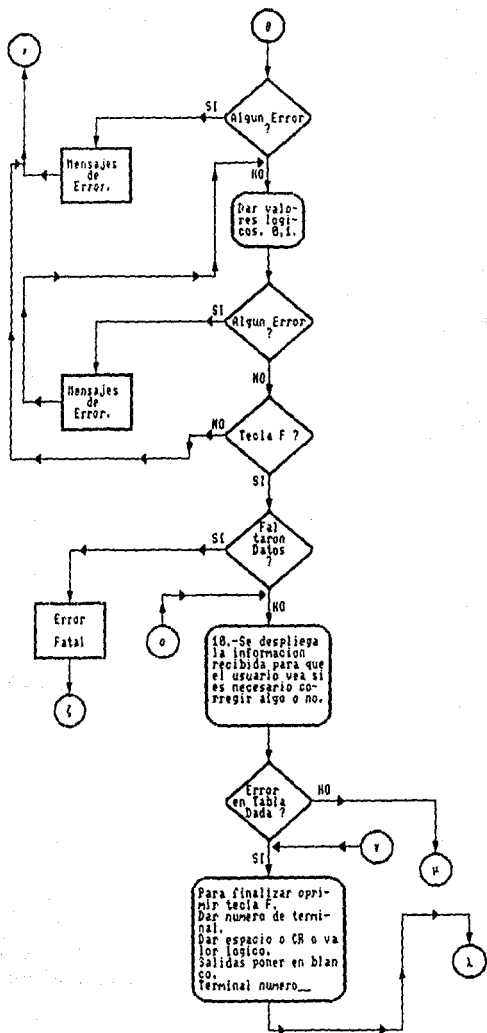
APENDICE B

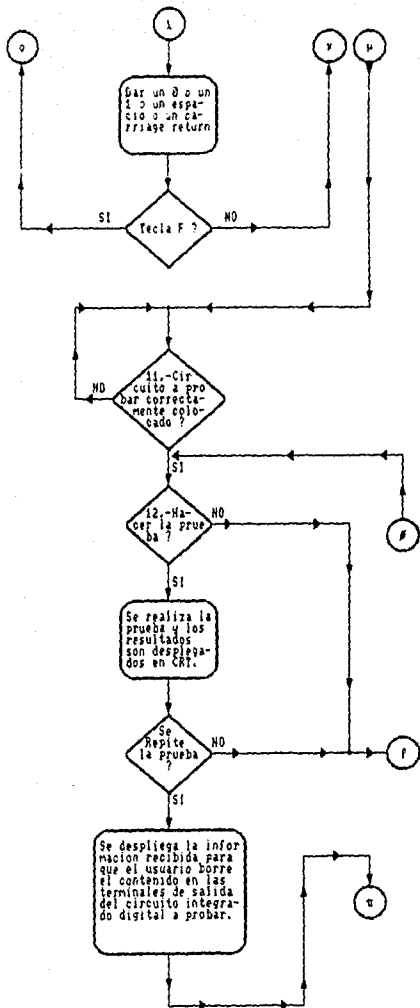


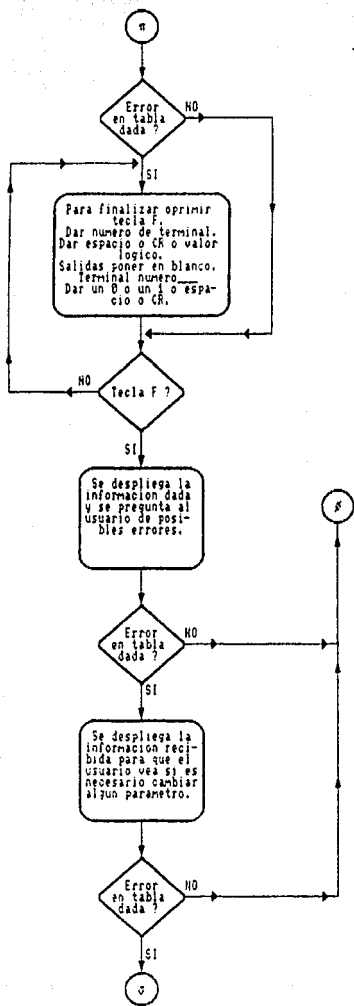


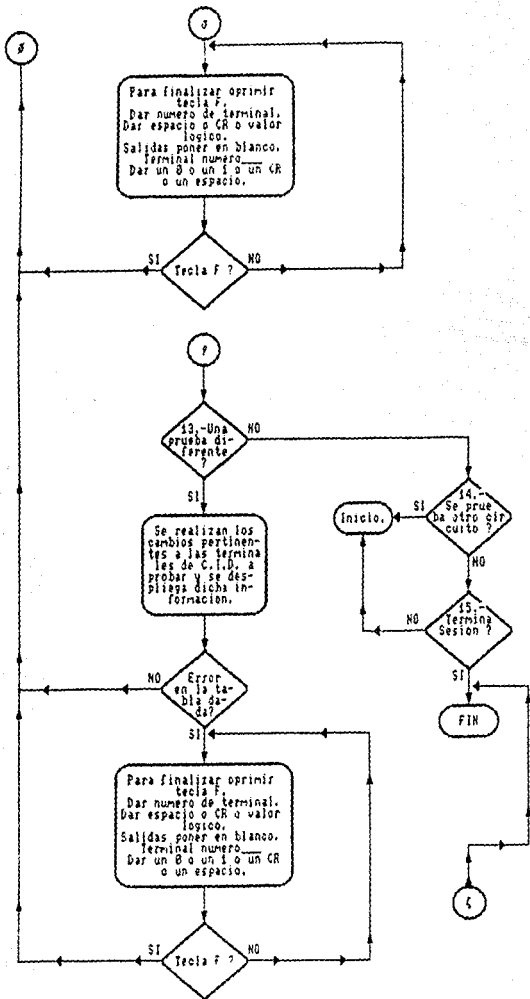












GLOSARIO.

BOPPAR: Indica la acción de poner en cero lógico las entradas, las salidas, los estados, etc. de un C.I.D.

CERO LOGICO: Se utiliza para indicar un nivel de voltaje entre los 0 y 1.9 volts.

C.I.D.: Siglas de: Circuito Integrado Digital.

I.P.: Siglas de: Instrumento Programable.

KBYTES: Abreviación de KILO-BYTES que especifica la capacidad de memoria que puede tener una computadora o un instrumento con memoria.

MICROCOMPUTADORA DE 8 BITS: Se designa así a la máquina que utiliza un microprocesador con 8 líneas de datos como máximo.

PATITAS: Ver TERMINALES.

PIN: Ver TERMINALES.

PINES: Ver TERMINALES.

TERMINALES: Son los puntos de conexión que tiene cada C.I.D., conocidos también como pin, pines o patitas.

UNO LOGICO: Se utiliza para indicar un nivel de voltaje entre los 2 y 5 volts.

BIBLIOGRAFIA.

1. - MEMOPY DATABOOK NATIONAL SEMICONDUCTOR.
2. - THE COMPLETE MOTOROLA MICROCOMPUTER DATA LIBRARY.
3. - TV TYPEWRITER COOKBOOK.
Don Lancaster.
4. - M6800 EXORCISER USER'S GUIDE.
5. - M6800 MICROCOMPUTER SYSTEM DESIGN DATA.
6. - TTL DATA BOOK FAIRCHILD SEMICONDUCTOR.
7. - MOTOROLA MICROPROCESSORS DATA MANUAL.
8. - THE TTL DATA BOOK FOR DESIGN ENGINEERS.
TEXAS INSTRUMENTS.
9. - LINEAR DATA BOOK.
NATIONAL.
10. - LINEAR INTEGRATED CIRCUITS.
MOTOROLA.
11. - THE LINEAR AND INTERFACE CIRCUITS DATA BOOK FOR DESIGN ENGINEERS.
TEXAS INSTRUMENTS INCORPORATED.
12. - UNA APLICACION DE LA COMPUTACION A LA INSTRUMENTACION PROGRAMABLE: ANALIZADOR DE MEMORIAS DIGITALES.
Tesis de Licenciatura en Fisica.
Roberto Serna Herrera. 1990.
13. - AN ENGINEERING APPROACH TO DIGITAL DESIGN.
William I. Fletcher.
Prentice-Hall, Inc.
14. - LOGIC CIRCUITS AND MICROCOMPUTERS SYSTEMS.
Claude A. Wiatrowski and Charles H. House.
McGraw-Hill.
15. - DIGITAL COMPUTER FUNDAMENTALS.
Thomas C. Bartee.
McGraw-Hill Kogakusha, Ltd.
16. - INTRODUCTION TO SWITCHING THEORY AND LOGICAL DESIGN.
Frederick J. Hill and Gerald P. Peterson.
Wiley International Edition.

79
ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

17. - TEORIA DE LAS CALCULADORAS NUMERICAS AUTOMATICAS.
Louis Nashelsky.
Editorial Alhambra, S.A.
18. - AN INTRODUCTION TO COMPUTEF LOGIC.
H. Troy Nagle Jr., B. D. Carroll and J. David Irwin.
Prentice-Hall, Inc.
19. - SISTEMAS ELECTRONICOS DIGITALES.
Enrique Mandado.
Marcombo Boixareu Editores.
20. - INSTRUMENTACION DIGITAL.
Asociación Mexicana de Ingenieros en Comunicaciones
Eléctricas y Electrónica. (AMICEE).
Editorial Limusa, México.
21. - COMPUTER LOGIC DESIGN.
M. Morris Mano.
Prentice-Hall, Inc.
22. - THE ART OF DIGITAL DESIGN. AN INTRODUCTION TO TOP-DOWN
DESIGN.
David Winkel and Franklin Prosser.
Prentice-Hall, Inc.
23. - TRANSISTORES EN TECNICA DIGITAL.
A. Koroncai y R. Alving.
Paraninfo. (Biblioteca Técnica Philips).
24. - MINI/MICROCOMPUTER HARDWARE DESIGN.
Gerge D. Kraft and Wing N. Toy
Prentice-Hall, Inc.
25. - THE DESIGN OF DIGITAL SYSTEMS.
John B. Peatman.
McGraw-Hill Kogakusha, Ltd.
26. - MICROCOMPUTER BASED DESIGN.
John B. Peatman.
McGeaw-Hill Kogakusha, Ltd.
27. - COMPUTER ORGANIZATION AND PROGRAMMING.
C. William Gear.
McGraw-Hill International Book Company.
28. - COMPUTER ARCHITECTURE AND ORGANIZATION.
John F. Hayes.
McGraw-Hill International Book Company.
29. - MICROPROCESSORS AND MICROCOMPUTERS.
Branko Soucek.
John Wiley and Sons.

30. - AN INTRODUCTION TO MICROCOMPUTERS.
VOLUME 1: BASIC CONCEPTS.
Published by: Adam Osborne and Associates, Inc.
31. - AN INTRODUCTION TO MICROCOMPUTERS.
VOLUME 2: SOME PEAL MICROPROCESSORS.
Published by: Adam Osborne and Associates, Inc.
32. - AN INTRODUCTION TO MICROCOMPUTERS.
VOLUME 3: SOME PEAL SUPPORT DEVICES.
Published by: Adam Osborne and Associates, Inc.
33. - INTRODUCTION TO MICROPROCESSOR SYSTEM DESIGN.
Harry Garland.
McGraw-Hill Series in Electrical Engineering.
34. - DIGITAL ELECTRONICS FOR SCIENTISTS.
H. V. Malmstadt and C. G. Enke.
W. A. Benjamin, Inc.
35. - DIGITAL INSTRUMENTATION.
A. J. Bouwens.
McGraw-Hill Book Company.
36. - CIRCUITOS ELECTRONICOS INTEGRADOS.
Asociación Mexicana de Ingenieros en Comunicaciones
Eléctricas y Electrónica. (AMICEE).
Editorial Limusa, México.
37. - SWITCHING AND FINITE AUTOMATA THEORY.
Zvi Kohavi.
McGraw-Hill Book Company.
38. - BUILT-IN TESTING OF MEMORY USING AN ON-CHIP COMPACT TESTING
SCHEME.
Kozo Kinoshita and Kewal K. Saluja.
IEEE Transactions on Computers; Vol. C-35; No. 10, October
1986; pags: 862-870.
39. - SINGLE-CHIP MICROCOMPUTER EXPANDS ITS MEMORY.
Harold W. Dozier and Robert S. Green.
Electronics; May 11, 1978; pags: 105-110.
40. - PROCESSOR-BASED TESTER GOES ON SITE TO ISOLATE BOARD FAULTS
AUTOMATICALLY.
Robert E. Anderson, and Robert G. Fulks, and Charles P.
Frusterio, and Frank S. Meade, and Donald E. Phelps.
Electronics; May 11, 1978; Pags: 111-117.
41. - M6800 MICROPROCESSOR PROGRAMMING MANUAL. MOTOROLA.