

14/21

**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**  
**FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN**



**"DISEÑO Y CONSTRUCCION DE UN SISTEMA  
PROBADOR DE CIRCUITOS INTEGRADOS  
CONTROLADO POR UNA COMPUTADORA"**

**T E S I S**

QUE PARA OBTENER EL TITULO DE:  
**INGENIERO MECANICO ELECTRICO**  
P R E S E N T A N :  
JUAN ANTONIO RESENDIZ CAPETILLO  
GUSTAVO ALBERTO RUIZ TORRES

ASESOR: ING. JORGE BUENDIA GOMEZ

CUAUTITLAN IZCALLI, EDO. DE MEX.

**TESIS CON  
FALLA DE ORIGEN**

1997



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 esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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



UNIVERSIDAD NACIONAL  
AVENIDA DE  
MEXICO

FACULTAD DE ESTUDIOS SUPERIORES - CUAUTITLÁN  
UNIDAD DE LA ADMINISTRACION ESCOLAR  
DEPARTAMENTO DE EXAMENES PROFESIONALES

U. N. A. M.  
FACULTAD DE ESTUDIOS  
SUPERIORES - CUAUTITLÁN

DEPARTAMENTO DE  
EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

DR. JAIME KELLER TORRES  
DIRECTOR DE LA FES-CUAUTITLÁN  
P R E S E N T E .

AT'N: Ing. Rafael Rodríguez Ceballos  
Jefe del Departamento de Exámenes  
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:  
"Diseño y Construcción de un Sistema Probador de Circuitos Integrados Digitales Controlado por una Computadora".

que presenta el pasante: Juan Antonio Resendiz Capetillo  
con número de cuenta: 8819271-4 para obtener el TÍTULO de:  
Ingeniero Mecánico Electricista.

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .  
"POR MI RAZA HABLARA EL ESPIRITU"  
Cuautitlán Izcalli, Edo. de Méx., a 16 de Octubre de 1975

PRESIDENTE Ing. José Luis Rivera López  
VOCAL Ing. Ubaldo Ramírez Urizar  
SECRETARIO Ing. Jorge Buendía Gómez  
PRIMER SUPLENTE Ing. Blanca de la Peña Valencia  
SEGUNDO SUPLENTE Ing. Nicolde Calva Tapia

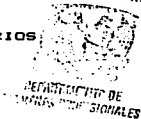
*[Handwritten signatures and dates]*  
07/14/96  
07/1986  
16/08/75  
11/21/96  
@Darin



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN N. A. R.  
UNIDAD DE LA ADMINISTRACION ESCOLAR FACULTAD DE ESTUDIOS  
DEPARTAMENTO DE EXAMENES PROFESIONALES SUPERIORES-CUAUTITLAN

ASUNTO: VOTOS APROBATORIOS



DR. JAIME KELLER TORRES  
DIRECTOR DE LA FES-CUAUTITLAN  
P R E S E N T E .

AT'N: Ing. Rafael Rodríguez Ceballos  
Jefe del Departamento de Exámenes  
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:

"Diseño y Construcción de un Sistema Probador de Circuitos Integrados Digitales Controlado por una Computadora"

que presenta el pasante: Gustavo Alberto Ruiz Torres  
con número de cuenta: 8836652-0 para obtener el TITULO de:  
Ingeniero Mecánico Electricista .

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .

"POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlan Izcalli, Edo. de Méx., a 16 de Octubre de 1996

PRESIDENTE Ing. José Luis Rivera López

VOCAL Ing. Ubaldo Ramírez Urizar

SECRETARIO Ing. Jorge Bucandía Gómez

PRIMER SUPLENTE Ing. Blanca de la Peña Valencia

SEGUNDO SUPLENTE Ing. Nicolée Galva Tapia

*J. Rivera* 02/10/96  
*U. Ramírez* 07/10/96  
*J. Bucandía* 16/10/96  
*B. de la Peña* 11/10/96  
*N. Galva*

## **AGRADECIMIENTOS**

**Esta tesis se la dedico a mis padres por su apoyo incondicional a lo largo de todos mis estudios, porque sin ellos no hubiera logrado finalizarlos.**

**A mis hermanos que siempre estuvieron cerca de mi.**

**Juan Antonio**

## **AGRADECIMIENTOS**

**A Dios el haberme permitido finalizar mis estudios.**

**A mi madre su paciencia y apoyo en todo momento ya que sin ella no hubiera sido posible el haber logrado una de las metas más importantes de mi vida.**

**A mis hermanas Kary y Mimi por su apoyo y comprensión en los momentos en que las necesite.**

**A mi padre, que es mi ejemplo a seguir.**

**Gustavo Alberto**

## **INTRODUCCION**

I.- Introducción.	1
II.- Características esenciales que debe tener la computadora para cumplir con el funcionamiento adecuado del sistema.	5
III.- Implementaron de un controlador de puertos paralelos utilizando un PPI8255.	8
IV.- Diseño y ensamble del sistema probador de circuitos integrados.	19
4.1.- Definición del sistema probador de circuitos integrados.	20
4.2.- Descripción interna del sistema probador de circuitos integrados.	20
4.3.- Fuente del sistema probador de circuitos integrados.	21
4.4.- Circuito de retención de datos.	24
4.5.- Circuito de potencia del integrado de prueba.	28
4.6.- Lectura de datos.	35
4.7.- Bloque de control.	38
V.- Funcionamiento del programa de control del sistema probador de circuitos integrados.	40
VI.- Funcionamiento del sistema probador de circuitos integrados.	57
VII.- Manual de usuario.	60
7.1.- Manual de funcionamiento del sistema probador de circuitos integrados lógicos utilizando los integrados ya almacenados dentro del sistema.	61
7.2.- Manual de funcionamiento del sistema probador de circuitos integrados sin tener que editar el circuito y sin que este exista en la lista de circuitos ya almacenados.	77
7.3.- Manual de funcionamiento del sistema probador de circuitos integrados lógicos para utilizar el editor de circuitos.	80
VIII.- Conclusiones.	90
Apéndices.	94
Apéndice A.	95
Bibliografía.	124

**DISEÑO Y CONSTRUCCION DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**INTRODUCCION**



## **INTRODUCCION**

Las fallas que presentan los circuitos integrados crean la necesidad de diseñar un sistema el cual sea capaz de probarlos de una forma confiable, debido a esto surge la idea de realizar un diseño que indique cuando un circuito integrado se encuentra funcionando de forma eficiente y en caso de que estuviera fallando alguna de sus partes o en su totalidad sea indicado a los usuarios de este diseño.

La utilidad de este proyecto se observa más en las necesidades presentadas en los laboratorios de las escuelas en las cuales se realizan prácticas con circuitos integrados, se necesita saber cuales están en buen funcionamiento, ya sea en algunos elementos internos que contiene el circuito integrado o en su totalidad, como sucede en algunos integrados que almacenan en su interior varios elementos, como es el caso del integrado 74LS00, el cual contiene cuatro compuertas NAND, y en el que puede darse el caso de que tres de las cuatro compuertas estén en buen estado, y para saber cual compuerta esta dañada se requiere invertir una cierta cantidad de tiempo que en momentos de realizar una práctica de laboratorio es indispensable, ya que se cuenta con el tiempo limitado.

El nombre de este proyecto es "PROBADOR DE CIRCUITOS INTEGRADOS DIGITALES POR MEDIO DE UNA COMPUTADORA PERSONAL" y su finalidad es

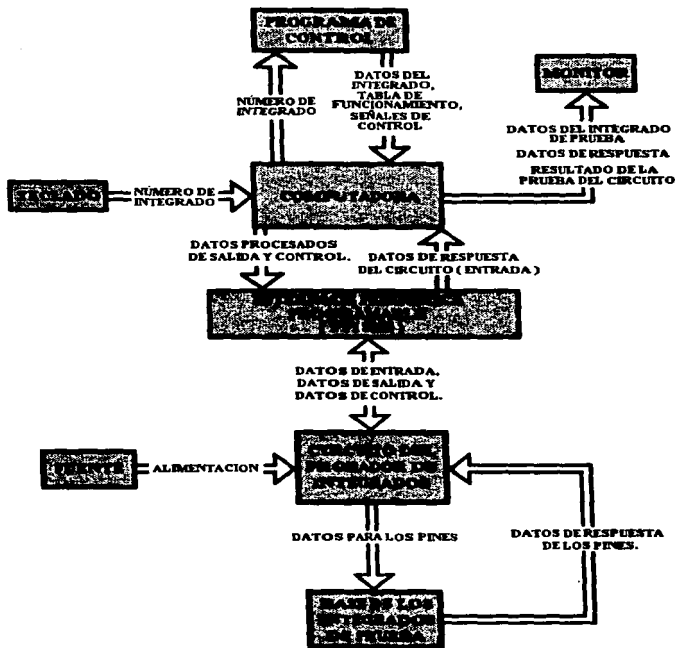
comprobar el funcionamiento correcto de los circuitos integrados digitales con un máximo de 24 pines, aunque para este diseño solo se incluirán los circuitos integrados de la serie TTL en sus diferentes clasificaciones según su tipo de salida los cuales son: totem pole, colector abierto, de tercer estado y la serie MOS en circuitos integrados lógicos, en caso de que se encuentre un error será indicado por el sistema, ya sea que este dañado totalmente o parcialmente.

La finalidad de este proyecto es aprovechar las ventajas que nos proporciona la computadora debido a que se diseñará una interface para controlar el sistema por medio de software, el funcionamiento y diseño de la interface será explicado posteriormente.

El diseño será dividido en varias secciones o bloques para su mejor estudio, una forma gráfica se observar estos bloques se encuentra en la figura 1, la cual divide el sistema en 6 partes las cuales se enumeran a continuación:

- 1 .- Una computadora ( con monitor y teclado ).
- 2 .- Una interface periférica programable.
- 3 .- Una fuente de alimentación.
- 4 .- El circuito del probador de integrados.
- 5 .- Una base para colocar los integrados de prueba.
- 6 .- Un programa de control.

Estos 6 bloques serán descritos en capítulos posteriores para su mejor comprensión.



**Figura 1. Diagrama de bloques del sistema probador de circuitos integrados por medio de una PC.**

**DISEÑO Y CONSTRUCCION DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**CARACTERISTICAS ESENCIALES QUE DEBE TENER LA  
COMPUTADORA PARA CUMPLIR CON EL FUNCIONAMIENTO  
ADECUADO DEL SISTEMA**

***CARACTERISTICAS FISICAS QUE DEBE CUMPLIR LA COMPUTADORA  
(HARDWARE)***

La computadora que se va a utilizar para instalar el probador de circuitos integrados debe cumplir las siguientes características: un procesador 286 o superior, un bus de expansión libre XT o AT, un monitor de resolución VGA color y una unidad de discos flexibles ( puede ser de 5 ¼ o de 3 ½ ) como requerimiento mínimo para poder operar el sistema probador de circuitos integrados.

Se considero como base el microprocesador 286 porque fue el primer sistema modelo en salir con sistema de vídeo VGA.

El bus de expansión puede ser cualquiera de los existentes ya sea XT o AT porque la tarjeta de interface solo requiere las señales de control que se generan en el bus XT que son las mismas de la primera sección del bus AT .

El sistema de vídeo fue basado en las necesidades del programa de control por ser el sistema VGA el que soporta 16 colores diferentes en un instante con resolución de 800 X 600 aunque se recomienda que sea monitor SVGA por su resolución que es de 1024 x 756 como las presentaciones gráficas son mejor apreciadas en monitores de mayor resolución y el programa de

control ocupa modo gráfico y maneja mas de dos colores se pide un monitor de las características antes mencionadas.

Refiriéndose al programa de control puede ser ejecutado en el mismo disco que lo contiene (refiriéndose al disquete de 3 ½ o 5 ¼ ), pero como el número de integrados que este pueda probar aumentara según las necesidades que el usuario requiriera de un mayor espacio por lo cual se recomienda instalarlo en un disco duro para no tener problemas con la extensión de las librerías, el tamaño del programa es de 350 KB y es el espacio mínimo requerido de disco duro aunque este espacio puede ir aumentando con el número de circuitos integrados a probar.

**DISEÑO Y CONSTRUCCION DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**IMPLEMENTACION DE UN CONTROLADOR DE PUERTOS  
PARALELOS UTILIZANDO UN PPI8255**

## INTERFACE CON EL PPI 8255

La necesidad de realizar una interface para controlar el sistema probador de circuitos integrados surge porque se requiere más de un puerto paralelo de ocho bits para controlarlo, es decir, se requieren tres puertos paralelos de ocho bits para poder controlar dicho sistema, y el puerto paralelo que trae normalmente la PC IBM o compatibles ( aunque es de ocho bits ) se utiliza para enviar datos a la impresora y las condiciones de la impresora entran por este puerto, y se requerirían de otros dos puertos paralelos de los cuales carece.

Para obtener tres puertos paralelos de ocho bits cada uno se utilizo el PPI 8255 (Programable Peripheral Interface ), se dice que es programable porque cada uno de los puertos que tiene puede ser programado para utilizarse como puerto de entrada ó salida, además se puede programar también en tres modos de uso, que son el MODO 0, el MODO 1 y el MODO 2, y la única diferencia entre estos es la forma en que van a trabajar los puertos A, B y C.

El PPI 8255 posee un bus bidireccional de datos de ocho bits, tres puertos paralelos de ocho bits, un bus de direcciones de dos bits, además internamente existe el puerto de control, que es donde se almacenan los datos de operación del circuito. El diagrama de pines del PPI 8255 se muestra en la figura 2.



Para poder programar el PPI 8255 dependiendo del modo en que van a operar se debe de tener en cuenta el diagrama de la figura 3, que muestra la forma de configurar el puerto de control para obtener las funciones requeridas de los puertos.

En donde para activar el PPI hay que darle el valor de uno lógico, de lo contrario este no recibirá ni dará dato alguno de los puertos, y para declarar el puerto será utilizado para recibir datos de entrada se asigna un uno lógico y si el puerto va a mandar los datos se le asigna un cero lógico.

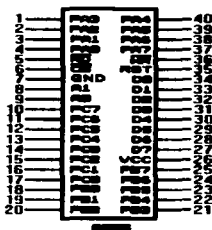
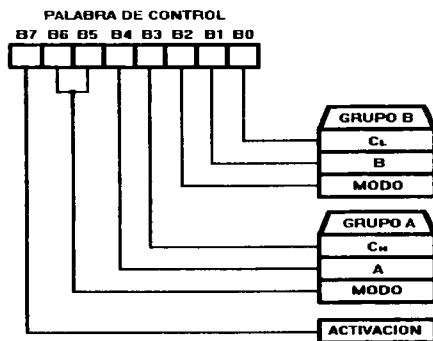


Figura 2. Configuración de pines del C.I. 8255

**NOTA:** Se debe tener en cuenta el modo de operación en que va a trabajar el PPI, pues en algunos modos el puerto C se convierte en puerto de control y solo recibe los datos del PPI que ya son definidos por el fabricante.



**Figura 3. Programación del PPI 8255**

Para definir el MODO de operación del PPI se utilizan los bits de modo usando la siguiente tabla :

00	Modo 0
01	Modo 1
1X	Modo 2

El bus de direcciones sirve para especificar a que puerto se van a mandar o recibir datos. El PPI contiene 4 puertos de ocho bits, por lo cual esta constituido por dos bits el bus de direcciones, y en el diagrama anterior están representadas por los pines A0 y A1.

Para comunicarle al PPI la palabra de control se utiliza la dirección 03 en hexadecimal en el bus de direcciones .

Para direccionar los puertos se utilizan las siguientes direcciones;

	A0	A1
Puerto A	0	0
Puerto B	0	1
Puerto C	1	0
Puerto de Control	1	1

El MODO 0 ó modo básico es el que se utilizó para desarrollar el sistema probador de circuitos integrados, ya que tiene la posibilidad de operar los puertos A ( de ocho bits ), B ( de ocho bits ), CBAJO ( de 4 bits ) y el CALTO (de 4 bits ) como puertos de entrada o de salida de datos. Para programar la forma de operación de los puertos se tiene que dar al puerto de control la palabra de control que nos va a definir la operación de los puertos del PPI8255.

Las palabras de control en este modo dependiendo de la forma de trabajo de puertos son las siguientes:

PUERTO A	PUERTO B	PUERTO C	PALABRA DE CONTROL	
			BIN	HEX
SALIDA	SALIDA	SALIDA	1000 0000	80
SALIDA	SALIDA	ENTRADA	1000 1001	89
SALIDA	ENTRADA	SALIDA	1000 1010	82
SALIDA	ENTRADA	ENTRADA	1000 1011	83
ENTRADA	SALIDA	SALIDA	1001 0000	90
ENTRADA	SALIDA	ENTRADA	1001 1001	99
ENTRADA	ENTRADA	SALIDA	1001 0010	92
ENTRADA	ENTRADA	ENTRADA	1001 1011	93

Para el desarrollo del proyecto se utilizo la palabra de control 82 hex.

El MODO 1 de operación se le conoce como handshake ( apretón de manos ) en el cual el puerto A y el puerto B funcionan como puertos de entrada / salida de ocho bits cada uno, y el puerto CALTO controla las funciones del puerto A y el puerto CBAJO controla las funciones del puerto B.

Las señales que recibe el puerto CALTO y el puerto CBAJO se muestran a continuación.

PUERTO B	ENTRADA	SALIDA
CBAJO	STB	OBF
	IBF	ACK
	INT	INTR

PUERTO A	ENTRADA	SALIDA
CALTO	STB	OBF
	IBF	ACK
	INT	INTR

El MODO 2 de operación del PPI es conocido por trabajar bidireccionalmente, aunque este modo de trabajo tiene como características el uso del puerto A como el puerto bidireccional, el puerto B es deshabilitado y no funciona en este modo y el puerto C se utiliza para controlar el funcionamiento del puerto A y para obtener información acerca del estado del PPI. Las funciones que realiza el puerto C en el MODO 2 para el control del puerto A se muestra a continuación.

PUERTO A	BIDIRECCIONAL
PUERTO CALTO Y CBAJO	INTR
	OBF
	ACK
	STBA
	IBFA

### LA TARJETA PARA EL PPI 8255.

El circuito utilizado para desarrollar la interface de la computadora se formo con un integrado 8255A-5 ( que es la interface de puertos paralelos ), un 74LS138 ( un demultiplexor de 3\*8 ), un bloque de jumpers de 2 líneas y ocho posiciones, y una tarjeta prototipo para el bus de la PC.

La ventaja de esta tarjeta para controlar el PPI 8255 es que se utiliza un multiplexor ( 74LS138 ) para habilitar el PPI en ocho distintas direcciones, lo cual nos da la seguridad de que el PPI no se activara cuando existan otras tarjetas que puedan activarse en esa misma dirección lógica, como pueden ser las tarjetas de sonido, el CD-ROM, etc.. Para definir que dirección se va a ocupar se tiene el bloque de jumpers, que es de dos líneas y ocho posiciones, y en cada posición distinta se tiene una dirección las cuales se muestran a continuación:

POSICIÓN DEL JUMPER	DIRECCIÓN	
	HEX	DEC
1	200	512
2	220	544
3	240	576
4	260	608
5	280	640
6	2A0	672
7	2C0	704
8	2E0	736

El desarrollo del circuito de la interface esta diseñado en base a las señales que trabajan en la ranura de expansión de la IBM PC de una maquina XT , la cual tiene las siguientes señales en sus terminales como se puede observar en la figura 4.

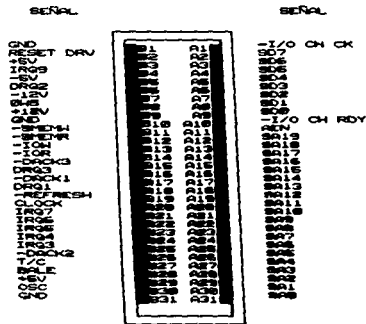


Figura 4. Señales en la ranura de expansión XT

Aunque las mismas señales que nos proporciona una computadora XT nos las proporciona una computadora con ranura de expansión de una máquina IBM PC AT, lo cual significa que esta tarjeta interface puede trabajar en una máquina XT como una AT. La ranura de expansión de una computadora AT (fig. 5) tiene los siguientes datos en sus terminales .

El desarrollo del circuito de la interface esta diseñado en base a las señales que trabajan en la ranura de expansión de la IBM PC de una maquina XT , la cual tiene las siguientes señales en sus terminales como se puede observar en la figura 4.

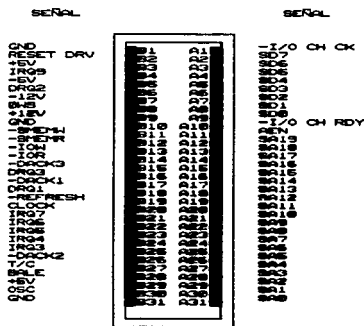


Figura 4. Señales en la ranura de expansión XT

Aunque las mismas señales que nos proporciona una computadora XT nos las proporciona una computadora con ranura de expansión de una máquina IBM PC AT, lo cual significa que esta tarjeta interface puede trabajar en una máquina XT como una AT. La ranura de expansión de una computadora AT (fig. 5) tiene los siguientes datos en sus terminales .



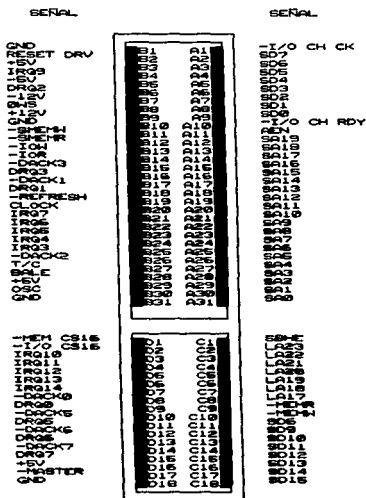
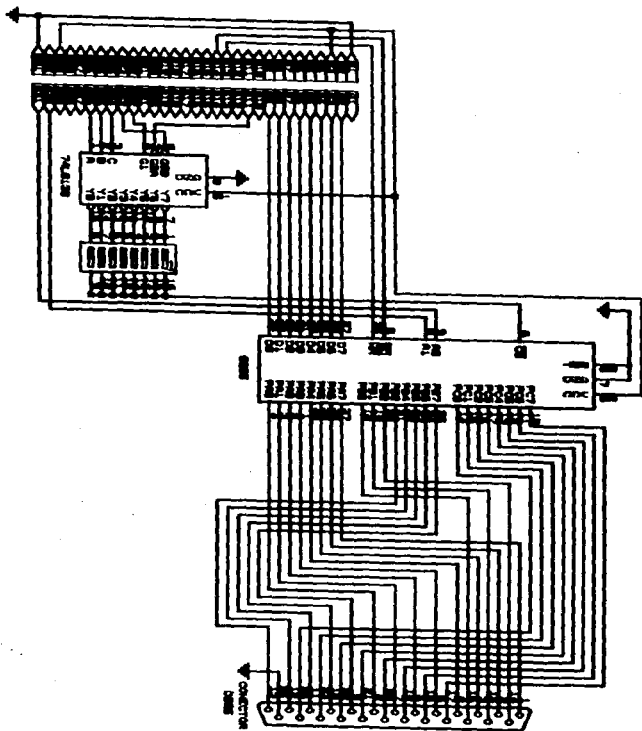


Figura 5. Señales en la ranura de expansión AT

Después de lo descrito anteriormente de como controlar a el C.I. 8255 para los propósitos de esta tesis se muestra en la figura 6 en diagrama completo de como se conecta el PPI8255 para generar tres puertos paralelos, los cuales son de utilidad para lograr el objetivo de esta tesis.



**DISEÑO Y CONSTRUCCION DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**DISEÑO Y ENSAMBLE DEL SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS**

## DEFINICIÓN DEL SISTEMA PROBADOR DE CIRCUITOS INTEGRADOS

La definición del sistema probador de circuitos integrados por medio de una computadora personal, se puede definir como un dispositivo digital, el cual comprobara el funcionamiento correcto de circuitos integrados digitales de hasta 24 pines, principalmente la serie TTL y la serie MOS, siendo estos integrados de múltiples entradas y/o salidas, ya sean totem pole, de colector abierto o de tercer estado. El correcto funcionamiento se comprueba en base a la tabla de verdad de cada circuito integrado y esto lo hará para cada combinación que exista en las entradas, obteniendo la(s) respuesta(s) de la(s) salida(s) de cada circuito integrado comprobando que sea la apropiada con respecto a la combinación de datos que esta siendo insertada en sus terminales de entrada tomando en cuenta la tabla de verdad.

## DESCRIPCION INTERNA DEL SISTEMA PROBADOR DE CIRCUITOS INTEGRADOS

El sistema probador consta de una interface la cual se encarga de enviar y recibir datos al circuito de prueba y el circuito principal.

El circuito principal es el que se encarga de retener los datos enviados por la computadora, proporcionar mayor corriente cuando el pin es de alimentación y leer los datos que proporciona el circuito integrado de prueba y esta constituido por los siguientes secciones:

- 1.- Fuente de alimentación.
- 2.- Bloque de retención de datos.
- 3.- Bloque de potencia.
- 4.- Bloque de lectura.
- 5.- Bloque de control.

## FUENTE DEL SISTEMA PROBADOR DE CIRCUITOS INTEGRADOS

La fuente de alimentación utilizada para el sistema probador de circuitos integrados fue diseñada con el propósito de no sobrecargar la fuente interna de la computadora, además esta fuente tiene la característica de activarse por medio de la computadora, es decir que se enciende cuando la computadora le envía una señal por medio del puerto C de la interface.

Para el diseño de la fuente se utilizó un transformador que tiene la capacidad de alimentar una carga que requiera hasta 1 amperé y una relación de 15 a 1, por lo cual si se tiene un voltaje en el primario en el transformador de 127 Vca ( rms ) ó 179.6051 V pico, en el secundario del transformador se obtiene un voltaje de 8.5 Vca ( rms ) ó 12 Vpico y después de pasar el puente rectificador de diodos se obtiene la señal rectificada, para tener un voltaje de rizo pequeño se

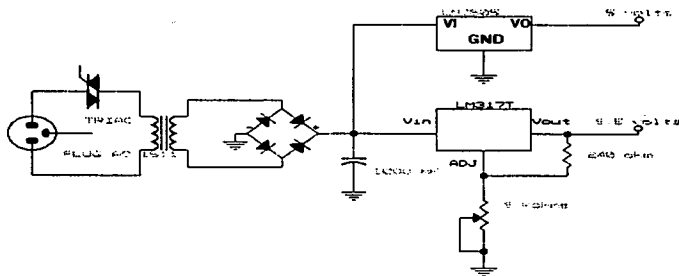


Figura 7. Diagrama de la fuente de alimentación.

coloca un capacitor electrolitico de 1000 microfaradios, con lo cual se obtiene un voltaje en el capacitor de 12 volts, y este voltaje es introducido a un regulador de voltaje programable ( LM 317 ) el cual tiene una capacidad de manejar máximo 1.5 amperes y puede realizar variaciones en su voltaje de salida desde 1.2 volts hasta 37 volts, en este caso la salida máxima de voltaje es de 12 volts por ser el voltaje que entrega el transformador. El diagrama de la fuente se observa en la figura 7.

Para controlar la activación de la fuente por medio de la computadora se ocupo un opto-triac y un triac, el cual tiene la siguiente configuración (figura 8).

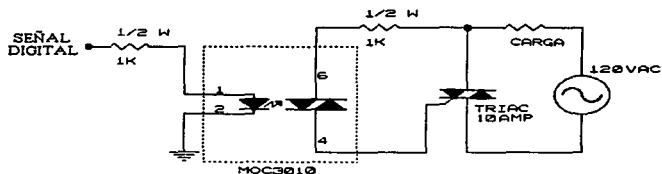


Figura 8. Diagrama del encendido electrónico por medio de un TRIAC.

Como se puede observar la señal digital dada por la computadora a través de la interface activa el led que se encuentra dentro del MOC3010, el cual activa el foto-triac que se encarga de controlar el triac la alimentación que se encarga de controlar el paso de la corriente hacia la carga que en este caso es el primario del transformador de la fuente.

El triac se eligió de acuerdo a las características de corriente y voltaje necesarias para la fuente de alimentación, en este caso fue de 10 amperes a 250 volts, son un poco altos para los valores reales pero hay que tomar en cuenta rangos de seguridad para evitar un deterioro rápido de este componente.

La necesidad de controlar la fuente de alimentación es debido a que si el sistema probador de circuitos integrados es encendido manualmente y este no se encuentra conectado a la computadora y como no tiene a la entrada los datos adecuados al sistema, se corre el riesgo de que este lea datos erróneos en la entrada, ya que por ser circuitos TTL los que lo constituyen estos

tienen la característica de tomar en cuenta como un uno lógico una terminal al aire, las cuales provoquen que se activen algunos circuitos que pueden quemarse, debido a su configuración.

La fuente de alimentación nos debe de proporcionar voltajes de 5.5 el cual sirve para alimentar los transistores y obtenemos en su salida hacia el pin del circuito integrado de prueba un voltaje aproximado de 5 volts.

Los 5 volts sirven para alimentar a todos los circuitos integrados que componen el diseño.

### CIRCUITO DE RETENCION DE DATOS

Este bloque esta constituido por un circuito integrado el cual nos permite retener los datos en forma de bits (1 lógico y 0 lógico ), ya que la computadora solo puede entregar ocho bits en el puerto de salida por cada instante, y por cada pin de el integrado a prueba se necesitan dos bits, cada integrado retiene ocho bits, se necesita de un integrado por cada cuatro pines.

El circuito integrado al que hacemos mención es el 74LS373 y las características que lo hacen indispensable para este circuito es que tiene un habilitador el cual cuenta de un inversor con schmitt trigger, que es el que se toma como señal de reloj para el funcionamiento de los flip flop tipo D que retienen los datos, este habilitador se activa en un 1 lógico y en este momento los datos que se tienen a la entrada de los flip flop se transfieren a la salida de estos, y al colocar el



habilitador en 0 lógico los datos se mantendrán fijos a la salida aunque en la entrada estén cambiando.

Haciendo mención a la tabla de verdad del integrado 74LS373 que corresponde a un integrado de 8 flip flop tipo "D", es la siguiente:

Dn	LE	OE	Qn
H	H	L	H
L	H	L	L
X	L	L	Qo
X	X	H	Z*

donde:

Dn = datos de entrada.

LE = latch enable

OE = habilitador de la salida.

Qn = datos de salida

H = nivel alto de voltaje (5 volts).

L = nivel bajo de voltaje (0 volts).

X = Puede ser cualquier valor (1 ó 0 lógico)

Z = alta impedancia

con esta tabla de verdad se puede observar el funcionamiento del integrado, dadas las señales de control se puede lograr que el integrado coloque sus salidas en alta impedancia, manteniendo el habilitador OE en 1 lógico y regresar al funcionamiento normal de las salidas colocándolo 0 lógico. Para lograr que los datos que tenemos a la entrada de los flip flop pasen a la salida, tenemos que utilizar el habilitador LE ("latch enable" habilitador de compuerta), la cual manteniéndola en bajo ( 0 lógico ) se mantiene el dato que se tenía cuando el habilitador estaba en alto ( 1 lógico ), una vez en esta posición ( 0 lógico ) el integrado no permite cambiar las salidas aunque cambien las entradas, y manteniéndola en estado alto nos permite el paso de los datos de entrada del integrado a la salida de este (aunque estas sean señales cambiantes).

Por lo mencionado anteriormente se escogió este integrado ya que permite retener los datos por etapas sin tener que emplear mas direcciones de puertos para enviar los datos porque se tendrían que emplear un total de 48 bits de una computadora para transferir todos los datos en una sola etapa.

La figura y configuración de pines del circuito integrado 74LS373 se muestra en la figura 9 y la forma en la que se conectan las entradas de este integrado para poder retener la 48 señales necesarias para probar un circuito integrado de 24 pines y queda estructurado como se puede observar en la figura 10.

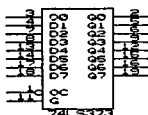


Figura 9. Configuración del C.I. 74LS373.

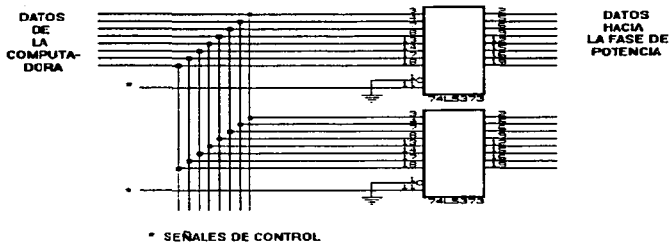


Figura 10. conexión de los C.I. 74LS373.

En la figura 10 se puede observar que los datos de entrada de cada circuito integrado están conectados en paralelo y lo que hace diferentes las señales de salida de cada circuito integrado es la señal de control, la cual selecciona que circuito integrado se va a habilitar para que los datos de entrada pasen a la salida.

### CIRCUITO DE POTENCIA DEL INTEGRADO DE PRUEBA.

Este etapa se encarga de proporcionar los datos al integrado de prueba, en forma de 1 lógico, 0 lógico, tierra, alimentación y en el caso de salida de colector abierto proporciona la resistencia con 5 volts para la salida.

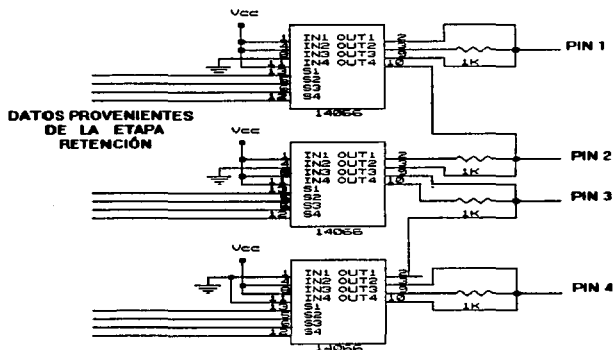


Figura 11. Conexión del integrado 14066.

En esta sección se hizo una primera prueba de funcionamiento con un circuito integrado con nomenclatura 14066, el cual son 4 interruptores digitales por cada circuito integrado y el circuito armado se muestra en la figura 11, en la cual se observa que se usaba un interruptor para

proporciona el 1 lógico o la alimentación del integrado, otro interruptor para el 0 lógico y la tierra y un último interruptor para aquellos integrados que son de colector abierto con su respectiva resistencia y voltaje.

Este circuito tuvo éxito en su funcionamiento mientras se probaron circuitos integrados de baja corriente como son compuertas lógicas totem pole, circuitos integrados CMOS pero al probar circuitos integrados que excedieran la corriente que manejaban los interruptores digitales los datos de salida se alteraban, proporcionando el probador datos erróneos por lo cual se obtenía un mensaje de error en el circuito integrado de prueba, debido ha este problema se opto por cambiar esta etapa por transistores los cuales pueden controlar mayor corriente.

El diseño del circuito que alimenta los pines del circuito integrado de prueba y que además sostiene los valores lógicos esta formado por dos transistores 2N2222 que son transistores de potencia para dar la corriente suficiente al circuito de prueba y además de algunas resistencias como se puede apreciar en la figura 12.

A simple vista puede que sea un poco difícil visualizar el funcionamiento de este circuito, pues proporciona un voltaje de alimentación de 5v con una corriente máxima, un nivel de voltaje de 5 volts para dar la señal de 1 lógico, y una señal de 0 volts para proporcionar al integrado la señal de cero lógico, además este circuito en ciertas circunstancias es inmune a los voltajes de respuesta en la salida de los C.I. de prueba, porque debido a la respuesta de los integrados esta puede ser de cinco volts ( 1 lógico ) ó cero volts ( cero lógico ), además de tener la capacidad de

proporcionar la resistencia adecuada y esta conectada a cinco volts para que los integrados de colector abierto puedan funcionar adecuadamente.

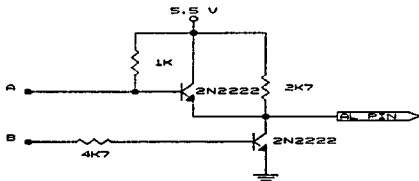


Figura 12. Circuito de potencia.

Para poder desarrollar una explicación más sencilla del circuito anterior, se puede tomar en cuenta que este se puede dividir en dos circuitos más simples.

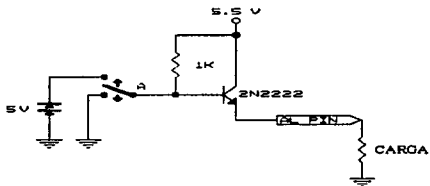


Figura 13. Análisis digital para proporcionar alimentación a los C.I.

Este circuito mostrado en la figura 13 sirve para proporcionar la alimentación al circuito de prueba el cual esta representado por la carga, el interruptor A representa las señales que nos da el circuito lógico ( un FLIP-FLOP D 74LS373 ) que controla el transistor, el cual nos proporciona una señal de cinco o cero volts, pero esta señal tiene una capacidad de corriente muy pequeña ( del orden de 12 mA ), la cual, para las características de alimentación de la mayor parte de los circuitos integrados es insuficiente y puede provocar un mal funcionamiento del circuito de prueba.

Suponiendo que primeramente el interruptor A se encuentra en la posición de 5 volts ( 1 lógico ), provoca que el transistor se encuentre en estado de saturación, y la resistencia de 1 kohm conectada a 5,5 volts sirve para ayudar a que el voltaje en la base del transistor sea de 5 volts, ya que como se sabe un uno lógico esta en el rango de voltajes de 2 volts a 5 volts. En este estado, el transistor proporciona en el emisor la corriente necesaria para alimentar el circuito de prueba, aparte que mantiene el nivel de voltaje de 5 volts, que es el nivel de alimentación para los circuitos TTL y estos trabajen adecuadamente.

Para el caso opuesto, cuando el interruptor A se encuentra en la posición de tierra ( 0 lógico ), la base del transistor se encuentra en un nivel de cero volts, lo que provoca que el transistor se encuentre en el estado de corte, en donde el flujo de corriente en el emisor del transistor se encuentra en cero, lo cual lleva a un estado de "inactivación" del transistor.

Para cuando este circuito se tiene que enfrentar a que en el emisor recibe la señal de respuesta del integrado de prueba, este debe tener el interruptor A en la posición de cero volts como se muestra en la figura 14, para que el transistor se encuentre en estado de corte, y entonces ya sea que el integrado de prueba proporcione un valor de cero lógico no se provoque una corriente excesiva del transistor a la salida del integrado de prueba ( tierra ) pudiendo provocar que se dañe el transistor o el circuito de prueba.

Para el caso que el integrado de prueba regresara como respuesta un uno lógico, este no presentaría problema alguno, ya que debido al "diodo" que forma la base y el emisor del transistor, no fluiría corriente alguna por ese sitio. Este también es el caso cuando el pin del circuito de prueba es de entrada, porque los datos de 1 y 0 lógicos se proporcionan por medio del otro transistor.

El segundo circuito que se forma para el análisis tiene la similitud al que se utiliza para formar las compuertas lógicas NOT, y se presenta como sigue (figura 15):

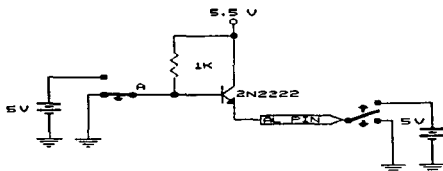


Figura 14.



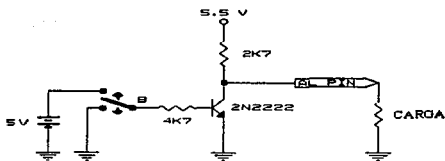


Figura 15.

Como se puede apreciar en la figura 15, el interruptor B muestra las señales lógicas que nos proporciona el circuito lógico que lo controla ( 74LS373 ). El tipo de señal a la salida del transistor proporciona valores de corriente y voltaje similares a un TTL, lo que lleva a pensar que si da una salida con valores a nivel TTL entonces para que se coloco.

La respuesta es sencilla, debido a que se necesita tener un punto en donde la corriente de alimentación del integrado fluya hacia tierra, ya que si se utiliza una terminal de un integrado lógico, puede sobrecargarse y dañarse, otra función de este circuito es proporcionar la resistencia que se debe colocar a los integrados de colector abierto, el cual un integrado TTL puede variar, ya que normalmente a la salida de estos hay una etapa de amplificación de corriente, la cual es formada por un transistor y una resistencia pequeña de la alimentación al colector del transistor.

Para tener una mejor idea del funcionamiento de este circuito supongamos que el interruptor B se encuentra en la posición de 5 volts ( 1 lógico ), el transistor se encuentra en estado de saturación, y por lo tanto el nivel de voltaje que existe en el pin del integrado es aproximadamente igual a cero volts, lo que nos proporciona una señal de 0 lógico al integrado de prueba, además si el interruptor B se encuentra en este estado y el integrado de prueba requiere que sea conectado a tierra ( GND ) proporciona un lugar por donde puede fluir la corriente en un cantidad mayor a un TTL .

Para cuando el interruptor se encuentra en la posición de 0 volts ( 0 lógico ) el transistor se encuentra en estado de corte, por lo que en el colector del transistor tenemos un valor de voltaje de 5 volts, este valor nos proporciona una señal de 1 lógico al integrado de prueba.

Para cuando este circuito se enfrenta a que el integrado de prueba va a responder por ese punto, el interruptor B se debe de encontrar en la posición de 0 volts.

Como se puede apreciar cuando la respuesta que da el integrado es de 5 volts ( 1 lógico ) en el colector del transistor hay un voltaje similar, por lo que al estar al mismo nivel no ocurre ninguna cosa anormal al circuito, y cuando el integrado responde en cero volts ( 0 lógico ) debido a la resistencia de 2.7 kohms el voltaje existente entre el colector y la alimentación es consumido por la resistencia. Para cuando se tiene un integrado de colector abierto, entonces la resistencia de 2.7 kohms funciona como la resistencia entre la salida del integrado de prueba y la alimentación.

## LECTURA DE DATOS

Este bloque se encarga de leer la respuesta de los integrados, ingresando los datos en grupos de 8 bits por lo que para esta sección se ocupa el circuito integrado 74LS245 que esta compuesto de 8 buffer's bidireccionales, la configuración del circuito integrado se muestra al figura 16 en la cual se observa la disposición de pines del circuito integrado faltándole únicamente el pin de tierra (10) y el de alimentación (20).

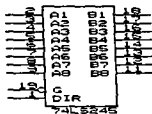


Figura 16. Diagrama esquemático de C.I. 74LS245.

Este integrado proporciona la opción de controlar la dirección por medio del pin 1 y un habilitador en el pin 19 el cual sirve para poner el integrado en tercer estado, a continuación se observara su tabla de verdad en la cual se vera el funcionamiento básico de este circuito integrado.

ENTRADAS		SALIDAS
HABILITADOR	DIRECCIÓN	
L	L	Datos del Bus B al Bus A
L	H	Datos del Bus A al Bus B
H	X	Bus en tercer estado

DONDE:

H = Nivel de voltaje alto.

L = Nivel de voltaje bajo.

X = Estado no relevante.

Como se puede observar la tabla de verdad se puede controlar la dirección del integrado ya sea en el sentido A-B o en el sentido B-A , y como se menciona anterior mente tiene un habilitador el cual permite controlar el funcionamiento de cada integrado por lo cual se pueden conectar todas las salidas en paralelo conectando el bit menos significativo de un integrado con el bit menos significativo del otro como se muestra en la figura 17.

Ahora se describe un poco el funcionamiento de este bloque, el cual se activa por medio de los habilitadores cuando estos reciben una señal en un nivel bajo de voltaje, entonces se dejan pasar los datos de las terminales A a las B, esto se hace secuencialmente, esto es uno por uno ya

que si se activaran dos integrados al mismo tiempo se podría producir un corto por este tipo de arreglo, la computadora compara la información con la tabla que tiene y esto determina si un integrado funciona correctamente o esta dañado.

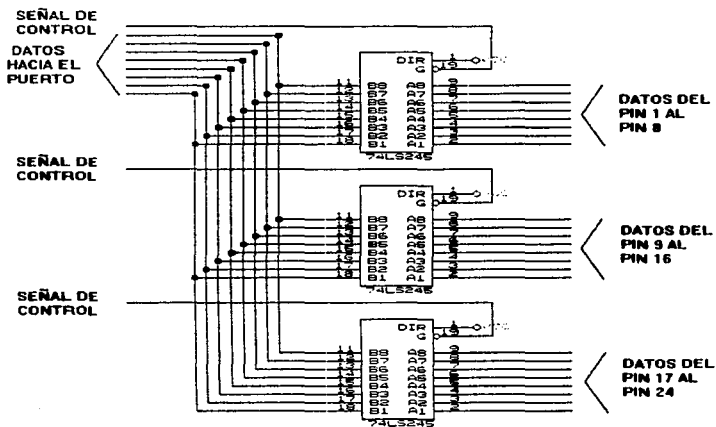


Figura 17.

<b>BLOQUE DE CONTROL</b>
--------------------------

Este bloque esta constituido por el puerto C de la interface y con ayuda de los integrados 74LS139 los cuales son dos multiplexores de 2 por 4 , el cual tiene la siguiente tabla de verdad

ENTRADAS			SALIDAS			
E	A0	A1	O0	O1	O2	O3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

**DONDE:**

**H = NIVEL ALTO DE VOLTAJE.**

**L = NIVEL BAJO DE VOLTAJE.**

**X = ESTADO SIN RELEVANCIA.**

Esta tabla nos muestra el funcionamiento del integrado ya que podemos ver que cuenta con un habilitador el cual necesita estar en bajo para el buen funcionamiento del integrado como multiplexor ya que de otro modo todas las salidas estarían en un nivel alto de voltaje.

Este integrado se conecta de tal manera que controle el bloque de retención de datos para ir seleccionando el integrado que mantendrá los datos mientras se realiza la comparación con la tabla de verdad de cada integrado, la forma de conexión se muestra en la figura 18.

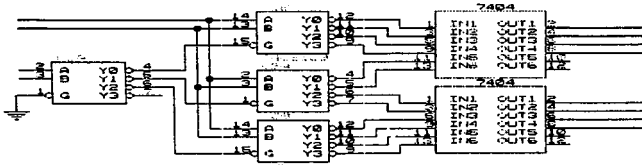


Figura 18. Bloque de control de la sección de retención de datos.

La otra parte del puerto C se encarga de controlar el bloque de lectura de datos, esto lo hace mediante los habilitadores que poseen los integrados base en esta sección y para esto no se necesita de ayuda de algún otro integrado ya que llega la señal directa de puerto a los integrados 74LS245.

**DISEÑO Y CONSTRUCCION DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**FUNCIONAMIENTO DEL PROGRAMA DE CONTROL DEL  
SISTEMA PROBADOR DE CIRCUITOS INTEGRADOS**



## **FUNCIONAMIENTO DEL PROGRAMA DE CONTROL**

El programa que controla el sistema probador de circuitos integrados por medio de una computadora personal fue desarrollado en lenguaje C. El programa total consta de 64 archivos, los cuales se diseñaron para controlar el sistema en general, para realizar pantallas de presentación y gráficos, pantallas de ayuda al usuario, etc.. El programa se decidió realizarlo con varios archivos o elementos debido a que es mucho más fácil realizar una parte del programa y comprobar el funcionamiento de este, que realizarlo todo en su totalidad, ya que si se presenta un error en un programa pequeño es mucho más fácil detectarlo que si el error se presenta en un programa grande, además cuando se tiene que cambiar el funcionamiento del programa se facilita realizar los cambios en un programa pequeño, y las variables utilizadas por este son más fáciles de visualizar.

El programa principal se llama **CITEST.EXE** el cual se encarga de dirigir y controlar la activación de los elementos restantes del programa. El programa **CITEST** controla el funcionamiento de los siguientes programas:

**Gráficos**

**Presenta.com \***

**Enciende.c**

**Marcopc.c**

**Window1.c**  
**Ayuda.c**  
**Busca.c**  
**Load\_int.c**  
**Checmenu.c**  
**Manual.c \***  
**Alerta.c**  
**Edicion.c \***  
**Rev\_lst.c**  
**Salir.c**

\* Para un mejor aprovechamiento de la memoria de la computadora se dividió este en tres partes, las cuales solo se activan cuando se requiere, como por ejemplo, el archivo "manual" solo se activa cuando se quiere dar al integrado de prueba valores manualmente, el archivo "edición" solo se activa cuando se quiere editar alguna tabla de datos para un nuevo integrado y el archivo "presenta" se activa cuando se inicia el programa, este solo controla la pantalla de presentación y la configuración, pero se elimina de la memoria cuando se pasa de este punto. Estos programas fueron originalmente creados como archivos con extensión "EXE", pero para diferenciarlos del programa principal y estos no fueran activados y provocar un mal funcionamiento del sistema se les cambio de extensión por "COM".

Algunos de estos subprogramas o subrutinas controlan otros programas más pequeños, y se explican a continuación cuales son estos programas y su funcionamiento dentro del programa principal.

El programa principal controla la dirección en la cual va a trabajar el PPI, controla también las opciones que constituyen el menú principal y codifica el número del integrado insertado por el usuario para ser buscado en la librería.

La subrutina de **GRÁFICOS** sirve para detectar la tarjeta de vídeo que tiene la computadora e inicializa el modo gráfico en caso que la tarjeta de gráficos sea **VGA**, y cargara las diferentes librerías en donde se albergan los diferentes tipos de letra utilizados en el programa. Para el caso que la tarjeta de vídeo no sea adecuada para soportar los gráficos que utiliza el programa de control no iniciara el programa y desplegara un mensaje mencionando el motivo de la falla.

El programa auxiliar **PRESENTA.COM** se utiliza para desplegar en pantalla la presentación del programa principal, y esta subrutina controla también el archivo **CONFIG.C** que es donde se guarda la configuración de operación de la tarjeta controladora con el PPI 8255 y es el programa donde se puede modificar la dirección de la misma al cambiar la posición del jumper. En el caso que no exista el archivo donde se guarda la configuración o esta no sea adecuada a cualquiera de las ocho posiciones, esta subrutina colocara el mensaje correspondiente y no correrá el programa.

La subrutina **ENCIENDE.C** es la subrutina que controla la inicialización del circuito del sistema probador de circuitos integrados, ya que como los flip-flop tipo "D" al encenderlos por primera vez no se sabe que tipo de datos tendrán, esta subrutina pide que se encienda el sistema y además pone todos los pines de la base de pruebas en un nivel de tierra ( cero lógico ). La subrutina enciende controla la subrutina **CLEAR\_PIN.C**, la cual esta mandando constantemente las señales de nivel de tierra desde el pin uno hasta el pin 28 y vuelve a iniciar. La subrutina **CLEAR\_PIN.C** se auxilia de la subrutina **MODO\_5.C** y **MODO\_0.C**. La subrutina **ENCIENDE.C** termina su ejecución cuando el sistema ya fue encendido y es presionada la tecla 'ENTER' para indicar a la computadora que este ha sido activado.

La subrutina **MODO\_5.C** sirve para recibir datos por el puerto B. Para que esta subrutina reciba datos por el puerto B del PPI8255 tiene que recibir primero la dirección de donde van a ser leídos los datos, es decir, se tiene que dar un dato al puerto C, para que por medio de este se activen los bits requeridos por la computadora para comprobar los datos presentes en los pines del integrado de prueba, y ya con este dato en el puerto C se lee el dato que se encuentran en el puerto B.

La subrutina **MODO\_0.C** sirven para enviar datos por el PPI 8255, es decir, esta subrutina envía datos al puerto C y al puerto A para colocar datos en los pines del integrado de prueba. Esta subrutina requiere para su funcionamiento que se le den las direcciones del puerto A y C, además de los datos que van a ser enviados por cada puerto.

La subrutina **MARCOPC.C** se encarga de dibujar en la pantalla el menú principal, en el cual se muestran las opciones generales que tiene el programa.

La subrutina **WINDOW1.C** se encarga de dibujar una ventana en la pantalla, esta puede ser de cualquier tamaño y en cualquier posición, solo hay que mandarle las coordenadas de donde se va a iniciar y donde va a finalizar. Esta ventana se utiliza para desplegar sobre ella los números de los circuitos integrados que tiene almacenados el sistema, así como los submenús dependiendo de cada selección que se realiza en el menú principal.

La subrutina **AYUDA.C** contiene información general de las funciones básicas de cada elemento del menú principal, algunas advertencias sobre su uso y quienes fueron los creadores del sistema, etc.. La información aparece en una ventana en la pantalla, y cambia esta dependiendo de la flecha presionada.

La subrutina **BUSCA.C** se emplea para buscar en el subdirectorio de librería el número de circuito integrado que se quiere probar, el cual si no se encuentra en este subdirectorio desplegará un mensaje en pantalla en el cual se menciona que el circuito integrado no existe. Este archivo controla la ejecución de la subrutina **LOAD\_INT.C**, el cual a su vez controla los gráficos de las compuertas lógicas que pueden pertenecer al integrado buscado, y las tablas de verdad de estos circuitos.

La subrutina **LOAD\_INT.C** recibe el numero del circuito integrado de prueba que se quiere probar o el numero de circuito integrado buscado, y si es una compuerta lógica este hará que se activen las subrutinas gráficas pertenecientes a las diferentes compuertas lógicas correspondientes al numero insertado, en caso que el numero del integrado que se inserte no corresponda con alguna compuerta lógica ya incluida este desplegara el gráfico de un circuito integrado. Para el caso que este buscando el numero de circuito integrado colocara en caso de ser un circuito insertado posteriormente, solo mostrara un gráfico del un integrado, y si el integrado no existe, entonces mostrara un mensaje expresando que ese integrado no existe en la libreria.

Las subrutinas que controla esta subrutina que son de tipo gráfico y/o de tablas de verdad son los siguientes:

**Amp-op.c**

**And1.c**

**Bufcontn.c**

**Buffer.c**

**Buflnv.c**

**Buflnvc.c**

**Cir-int.c**

**Flip-flop.c**

**Nand1.c**

**Nor1.c**

**Not1.c**  
**Or1.c**  
**And2inf.c**  
**And3inf.c**  
**And4inf.c**  
**And12inf.c**  
**Nand2inf.c**  
**Nand3inf.c**  
**Nand4inf.c**  
**Nand8inf.c**  
**Nor2inf.c**  
**Nor3inf.c**  
**Nor4inf.c**  
**Not\_inf.c**  
**Or2inf.c**

La subrutina **LOAD\_INT.C** controla también la subrutina **CHECMENU.C**, la cual es activada dependiendo si la subrutina **LOAD\_INT.C** fue activada desde el menú **BUSCAR** ó si fue llamada desde el menú de **CHECAR INTEGRADO**, que es donde se activa la subrutina.

La subrutina **CHECMENU.C** controla la activación de las subrutinas, presenta en pantalla un pequeño menú, en el cual se indica si se quiere iniciar la comprobación del circuito integrado de acuerdo con el número de circuito insertado ó si se quiere regresar al menú principal. Este

subprograma controla las siguientes subrutinas **INREADAT.C** y **CLEARPIN.C** del cual ya se explico cual es su funcionamiento.

La función **INREADAT.C** recibe el numero del integrado que se quiere probar, los números de los puertos en los que va a operar el **PPI 8255** y controlar cuando se van a activar las subrutinas correspondientes a la lectura de la tabla de datos del integrado de prueba, y las respuestas del mismo, y requerir por medio de los datos almacenados en los archivos de respuesta del integrado cuales son los pines con que va a responder el integrado y activar la subrutina correspondiente, la cual leerá desde el probador de circuitos el pin requerido para realizar la comparación de la respuesta del integrado con la respuesta que según la tabla de datos almacenada es la correcta.

Esta subrutina se apoya de las siguientes subrutinas:

**Lee\_dato.c**

**Ins\_dato.c**

**Det-blk.c**

**Outpin\_d.c**

**Compru2.c**

La subrutina **LEE\_DATO.C** lee el dato del archivo de entrada de datos o de respuesta de datos, para que esta subrutina funcione debe indicársele de que tipo de archivo se van a leer los datos, y la posición del dato dentro del archivo solicitado.



La subrutina **INS\_DATO.C** controla los datos que están siendo colocados en los pines del circuito integrado de prueba, solicita que sean leídos desde los archivos los datos requeridos para cada pin del circuito integrado de prueba utilizando la subrutina **LEE\_DATO.C**, controla que los datos sean cambiados dependiendo del número de combinación de datos que este en operación subrutina , controla la activación de la subrutina **INREADAT.C** para leer el dato presente en el pin de respuesta del circuito integrado de prueba y así realizar la comparación de la respuesta del integrado con la respuesta que debe tener el circuito según su tabla de verdad .

La subrutina **DET-BLK.C** detecta el número de bloques de pines (cada bloque consta de ocho pines) que esta ocupando el integrado de prueba, por ejemplo si un integrado tiene 14 pines, este utiliza dos bloque de pines para su prueba de funcionamiento. Esta subrutina se diferencia de la subrutina **detblock.c** debido a que esta detecta el número de bloques ocupados por el integrado ya que lee el archivo correspondiente al número del circuito integrado de prueba. Con esto se leen las combinaciones de datos dependiendo del número de bloques que ocupe el integrado.

La subrutina **OUTPIN\_D.C** tiene como función comprobar que los datos que se están leyendo desde los archivos donde se encuentran las tablas sean los adecuados para que estos puedan pasar al sistema probador de integrados, de lo contrario activara un mensaje de error, y no pasaran los datos al integrado de prueba. Para el caso que los datos sean los adecuados, estos datos son codificados para ser enviados a los pines del integrado, ya que cada vez que se envían datos para los pines se prueba, se envían de cuatro en cuatro, esto es, que cada vez que sale un

dato por el puerto A, este dato sirve para controlar las funciones de cuatro pines a la vez. Para poder enviar datos por puerto esta subrutina se apoya en la subrutina **MOD0\_0.C**.

La subrutina **COMPRU2.C** se encarga de comparar los datos almacenados en las tablas de funcionamiento almacenadas en la librería y los datos que responde el circuito integrado de prueba, y en caso de existir alguna diferencia, se activa una alarma y detiene al sistema para mostrar en que combinación de datos el circuito integrado de prueba fallo y en que compuerta apareció este error. Se encarga de cambiar el color del círculo de la compuerta en el que aparece el error para denotar que esta no funciona.

Al finalizar la operación de la subrutina **INREADAT.C** se activa la subrutina **CLEARPIN.C** la cual coloca a nivel de cero volts todas las terminales de la base de pruebas, para que si se quiere probar otro integrado, debido a la característica que se mantienen los datos debido a los flip-flop tipo "D" y al PPI 8255, no se dañe el siguiente circuito integrado que se quiera probar.

El programa auxiliar **MANUAL.COM** sirve para colocar datos a un integrado que no se encuentre en la lista de integrados a probar, este programa solo se activa cuando en el menú de **CHECAR INTEGRADO** se elige la opción manual. En este programa se pueden dar datos manualmente a los pines del integrado, solo debe tenerse cuidado de dar los datos adecuados a la tabla de verdad del integrado al cual se van a mandar los datos. Los datos dados aparecerán en la pantalla junto a la base de pruebas. Cada vez que se quiera dar una combinación de datos se tiene

que dar datos a las 28 terminales de la base, no solo a las que corresponden al integrado de pruebas.

Este programa se apoya en las siguientes subrutinas

**Decod\_in.c**

**Modo\_5.c**

**Modo\_0.c**

**Cod.c**

**Outerror.c**

**Outpin\_d.c**

**Clearpin.c**

La subrutina **DECOD\_IN.C** sirve para decodificar el dato que se está leyendo desde el puerto de entrada, el cual es de ocho bits, y solo regresa al sistema el valor lógico del pin solicitado por el sistema, por ejemplo, supongamos que en el puerto tiene el valor lógico 0100 0101 ( 45 hex ), y el sistema solo quiere saber el valor lógico del pin 3 más significativo, entonces la subrutina solo regresa al sistema el valor 1 lógico. Con esta subrutina el programa puede comparar pin por pin el dato que responde el integrado de prueba para ser comparado con el dato almacenado en su tabla de funcionamiento.

La subrutina **MODO\_5.C** sirve para recibir datos por el puerto B. Para que esta subrutina reciba datos por el puerto B del PPI8255 tiene que recibir primero la dirección de donde van a ser

leídos los datos, es decir, se tiene que dar un dato al puerto C, para que por medio de este se activen los bits requeridos por la computadora para comprobar los datos presentes en los pines del integrado de prueba, y ya con este dato en el puerto C se lee el dato que se encuentran en el puerto B.

La subrutina **MODO\_0.C** sirven para enviar datos por el PPI 8255, es decir, esta subrutina envía datos al puerto C y al puerto A para colocar datos en los pines del integrado de prueba. Esta subrutina requiere para su funcionamiento que se le den las direcciones del puerto A y C, además de los datos que van a ser enviados por cada puerto.

La subrutina **OUTERROR.C** sirve para colocar en pantalla el mensaje de error, en el que alguno de los datos insertados están fuera de los datos permitidos por el mismo sistema, y despliega una tabla en la que se muestran los datos que solo son aceptados.

La subrutina **OUTPIN\_D.C** tiene como función comprobar que los datos que se están leyendo desde los archivos donde se encuentran las tablas sean los adecuados para que estos puedan pasar al sistema probador de integrados, de lo contrario activara un mensaje de error, y no pasaran los datos al integrado de prueba. Para el caso que los datos sean los adecuados, estos datos son codificados para ser enviados a los pines del integrado, ya que cada vez que se envían datos para los pines se prueba, se envían de cuatro en cuatro, esto es, que cada vez que sale un

dato por el puerto A, este dato sirve para controlar las funciones de cuatro pines a la vez. Para poder enviar datos por puerto esta subrutina se apoya en la subrutina **MODO\_0.C**.

La subrutina **CLEARPIN.C** cada vez que se activa, se mandan los datos requeridos a la subrutina **MODO\_0.C**, y sirve para colocar datos en todos los pines de la base en cero lógico ( 0 volts ).

Cuando se termina la ejecución del programa auxiliar **MANUAL.COM** se libera la memoria que este ocupaba, debido a las características de la función en "C" llamada system.

La subrutina **ALERTA.C** despliega un mensaje, en el cual se expresa que se ha seleccionado la opción para editar nuevos archivos, los cuales sirven para almacenar las tablas de funcionamiento de los integrados que se quieren probar y que no se encuentran en la lista actual de los integrados que se puede probar su funcionamiento, y también despliega que estos datos deben ser insertados cuidadosamente, ya que si existe algún error en la inserción de estos, puede provocarse un mal funcionamiento tanto del sistema, como del integrado de prueba, y puede dañarse el sistema irremediablemente. Finalmente despliega la pregunta en la cual se dice si se desea continuar con el procedimiento de edición de integrados, en la cual si la respuesta es negativa, se regresara al menú principal, y si la respuesta es afirmativa iniciara el funcionamiento del subprograma **EDICION.COM**.

El programa auxiliar **EDICION.COM** se utiliza para editar nuevas librerías para los integrados que no existan en la librería actual. Este programa requiere de otras dos subrutinas

llamadas como **DEC\_NUM.C** y **DETBLOCK.C**, con este programa se crean las nuevas tablas de verdad de los nuevos integrados que se quieren probar, este programa verifica que el numero del nuevo integrado no sea igual a ninguno de los ya existentes, analiza el numero de pines que consta el nuevo integrado, ya que el numero de pines permitidos siempre debe ser par, y no debe rebasar de un limite máximo de 28 pines. Al estar realizando la edición de los datos este programa despliega en pantalla los datos que están siendo insertados en un gráfico en el cual se muestra la base de prueba de los integrados, y muestra cada dato que esta siendo insertado, y en caso que el pin sea de respuesta, se colocara un segundo dato el cual es el dato que el integrado debe responder a la combinación que esta siendo insertada. Este programa al terminar su ejecución desocupa la memoria utilizada por este, debido a la característica de la función `system` del lenguaje "C".

La subrutina **DEC\_NUM.C** lee los archivos temporales creados para los circuitos integrados nuevos, el cual lee bloque de datos de 3 bytes los cuales son utilizados para volver a acomodarlos, de tal modo que puedan ser utilizados por el programa **EDICION.COM** para ser almacenados en las tablas de funcionamiento del nuevo integrado.

La subrutina **DETBLOCK.C** recibe el numero de pines del nuevo circuito integrado y con este numero calcula el numero de bloques de pines utilizados por el nuevo circuito integrado.

El programa **EDICION.COM** crea dos archivos cada vez que se edita una nueva tabla de funcionamiento para un circuito integrado, uno es donde se almacenan el numero de pines y los datos que van a ser insertados a cada pin del circuito integrado de prueba, este archivo tiene la

extensión "IN". Mientras el otro archivo creado almacena el numero de compuertas que tiene el integrado, y los pines con que responde cada compuerta, además también se almacenan los datos que debe responder el circuito integrado después de cada combinación de datos, este archivo creado tiene la extensión "OUT".

Los datos almacenados en cada archivo son guardados de modo que cada tres bytes significan un numero, por ejemplo, si el archivo tiene la siguiente secuencia de números 001002020001, significa que esta guardando los siguientes valores 1,2,20,1. Dada la posición de cada tres bytes significan algún dato para el programa. Por ejemplo, el primer numero significa el numero de pin del circuito integrado, mientras el segundo numero es el valor lógico que va a ser asignado al pin del circuito integrado dependiendo de la tabla de funcionamiento del programa.

La subrutina **REV\_LIST.C** sirve para ver en pantalla los números de los integrados ya existentes en la librería, ya sean integrados TTL ó para integrados CMOS ( de los cuales solo se muestran los circuitos integrados que fueron originalmente adicionados al sistema ). También tiene una tercera opción, en la cual se despliegan todos los números de los integrados que están almacenados en la librería ( ya sean TTL, CMOS o circuitos integrados que fueron creados posteriormente en la librería ). Esta subrutina se auxilia de las subrutinas **TTL.C** y **CMOS.C** las cuales contienen los números de los integrados que se pueden probar, además también utiliza la subrutina **CLEARWIN.C**, la cual limpia la ventana donde están apareciendo los números de los integrados que se pueden probar.

La subrutina **SALIR.C** sirve para desplegar un aviso en el cual se despliega un mensaje para confirmar la finalización del programa y en caso afirmativo pedirá que sea apagado el sistema probador de circuitos integrados, para evitar un mal funcionamiento del mismo ó enviar datos erróneos a este que puedan dañar el sistema.



**DISEÑO Y CONSTRUCCION DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**FUNCIONAMIENTO DEL SISTEMA PROBADOR DE CIRCUITOS  
INTEGRADOS**

## FUNCIONAMIENTO

Una vez descrita cada sección por separado, el funcionamiento del probador de circuitos integrados será descrito con ayuda de bloques como se muestra el siguiente figura 1.

Este funcionamiento se basa principalmente en la interface de la computadora ya que es la que se encarga de enviar los datos que proporciona el programa a el sistema probador de circuitos integrados.

La forma de operar de este sistema se puede describir empezando por el programa que es el que se encarga de enviar los datos a el sistema, el funcionamiento de dicho sistema empieza cuando escogemos la opción de CHECAR INTEGRADO presionando la letra C, a continuación el numero 1 y luego insertando el numero del circuito integrado a probar, con esto se iniciara una serie de procesos con los que la computadora indicara si un circuito integrado funciona correctamente.

Esto se inicia cuando la computadora empieza a enviar datos por la interface en su puerto A y señales de control por el puerto C, estos datos mandados llegaran a los circuitos integrados 74LS373 los cuales se encargan de conservar los datos mientras se terminan de enviar todos, ya que como se menciona anteriormente no se pueden tener todos los datos en un mismo instante en las terminales del circuito integrado de prueba. posteriormente ya que los circuitos integrados

tienen todos los datos estos seleccionan la etapa de potencia que el dato que va a recibir el circuito integrado, estos datos pueden ser 1 lógico, 0 lógico, tierra, alimentación y en el caso de los de colector abierto alimentación por medio de una resistencia.

Una vez que el circuito integrado tiene todos los datos en sus terminales entonces se procede a leer estos datos lo cual lo hace por medio del circuito integrado 74LS245 y llegan a la interface por medio del puerto B seleccionando el grupo de 8 bits que va a leer, esta lectura se hará en grupos de 8 bits leyendo primero del pin 1 al 8, del 9 al 16 y del 16 al 24, esto lo puede hacer por medio de un control el cual es proporcionado por una parte del puerto C, ya que se leyeron los datos entonces el programa procederá a compararlos con los datos que ya tiene almacenado en la librería del integrado seleccionado, después de la comparación si se encuentra el circuito integrado mal en alguna de sus partes o en su totalidad el programa procederá enviando una señal roja y una alarma sonora indicando que parte del circuito integrado esta dañada.

Con lo mencionado anteriormente se da una breve explicación de como funciona el sistema probador de circuitos integrados, aunque esta explicación es muy general pero es el principio del funcionamiento de este prototipo el cual puede probar una amplia gama de circuitos integrados con la única condición de que estos sean digitales, esto es que tanto sus entradas como sus salidas necesiten datos digitales.

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**MANUAL DE USUARIO**

*MANUAL DE FUNCIONAMIENTO DEL SISTEMA PROBADOR DE CIRCUITOS  
INTEGRADOS LÓGICOS UTILIZANDO LOS INTEGRADOS YA ALMACENADOS  
DENTRO DEL SISTEMA.*

Al encender la computadora el PPI (Programmable Peripheral Interface) se activa iniciando todos sus puertos en el modo de entrada, lo cual significa que en las terminales del PPI no se encuentre ningún voltaje, y debido que el sistema probador de circuitos integrados por ser compuesto por circuitos lógicos TTL ( transistor transistor logic ) estos tengan "unos" ó "ceros" en sus terminales, además por estar compuesto el circuito del sistema probador de circuitos integrados por circuitos TTL, estos tienen la característica de que un pin al aire sea un uno lógico, por lo cual el sistema esta protegido para cuando se conecta el sistema antes de conectarlo a la computadora, previniendo así obtener datos erróneos los que podrían provocar un mal funcionamiento del sistema que lo dañaría irremediamente.

Al ejecutar el programa principal llamado CITEST.EXE este desplegará una pantalla (figura 19) en la cual se muestran el nombre del sistema y una opción, la cual dice "PRESIONA C PARA CONFIGURAR PUERTO", esta opción sirve para seleccionar la posición en la que se localiza el jumper en la tarjeta del PPI; esto es con la finalidad de tener varias opciones para direccionar los datos que va a enviar y recibir la computadora, sin que estos interfieran con cualquier otro componente instalado en la computadora, como pueden ser una tarjeta de red, la

tarjeta de sonido, etc., ya que puede suceder que alguno trabaje con la misma dirección de activación del PPI que controla el sistema probador de circuitos integrados. Para eliminar el problema de direcciones iguales existe sobre la tarjeta del PPI un jumper para que se pueda cambiar la dirección de control de la tarjeta del sistema, de lo contrario se provocaría el mal funcionamiento del Probador de Circuitos Lógicos, así como los accesorios que cuenta la computadora.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO	
FACULTAD DE ESTUDIOS SUPERIORES CUAUTEMOC	
SISTEMA PARA PROBAR CIRCUITOS INTEGRADOS	
Características principales	<ul style="list-style-type: none"> <li>• Operación automática</li> <li>• Operación manual</li> </ul>
Accesorios	<ul style="list-style-type: none"> <li>• Fuente de alimentación</li> <li>• Fuente de reloj</li> <li>• Fuente de reloj</li> <li>• Fuente de reloj</li> </ul>

Figura 19.

A continuación se muestra una tabla, en la cual se encuentran las direcciones lógicas en las que puede ser activado el PPI en caso que se encuentre interfiriendo con otros dispositivos de la computadora.

POSICIÓN DEL JUMPER	DIRECCION	
	HEX	DEC
1	200	512
2	220	544
3	240	576
4	260	608
5	280	640
6	2A0	672
7	2C0	704
8	2E0	736

Originalmente la tarjeta de control tiene el jumper en la tercera posición (figura 20), pero esta puede ser cambiada a cualquiera de las siete posiciones restantes, siempre y cuando la posición del jumper de la tarjeta del PPI8255 coincida con la posición marcada en el programa en la opción de configuración, de otro modo no se podrá controlar el sistema probador de circuitos integrados.

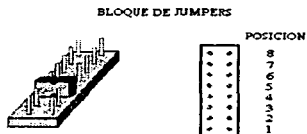


Figura 20.

**NOTA :** En el caso que la posición del jumper en la tarjeta no coincida con la posición que tiene almacenada el programa de control, al iniciar el programa y ejecutarse la prueba del sistema, este marcará que todos los pines en la base de pruebas están defectuosos, de tal modo que no se podrá avanzar más en el programa de control y regresará al sistema operativo.

### **CANBIANDO LA CONFIGURACIÓN DEL PUERTO.**

Para cambiar la dirección de activación del puerto (PPI), desde la pantalla inicial hay que presionar inmediatamente la tecla "C", y al hacer esto, nos llevará automáticamente a la pantalla de configuración del puerto paralelo, en la cual se muestra la posición del jumper desde que fue utilizado el programa por última vez, y también en donde aparece una pregunta en la que se pide confirmar si ha sido cambiada la posición del jumper en la tarjeta del PPI8255.

Si la respuesta es negativa se desplegará el mensaje "CONFIGURACIÓN CANCELADA" (figura 21) y el programa no realizará ningún cambio en la configuración del puerto y continuará el funcionamiento del programa con la configuración anterior.

Si la respuesta es afirmativa entonces pedirá que se inserte la posición del jumper en la que va a trabajar la tarjeta (figura 22). Aquí solo se van a aceptar los valores del 1 al 8, que son las únicas posiciones en las que puede ser puesto el jumper, en caso que sea colocada alguna distinta, el programa accionará una alarma y desplegará un mensaje de error.



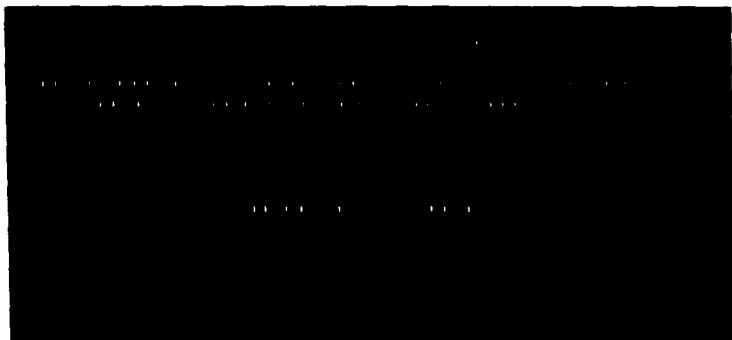


Figura 21.

En el caso de realizar una selección entre 1 y 8 el sistema preguntará si se esta seguro de el cambio que se va a realizar, y en caso de ser afirmativa la respuesta, será salvada la nueva posición en el archivo "CITEST.CFG", en caso contrario el cambio realizado será anulado y por lo tanto la posición anterior será la que siga operando el programa.

Para volver a entrar al modo de configuración del jumper se tendrá que volver a iniciar la ejecución del programa.

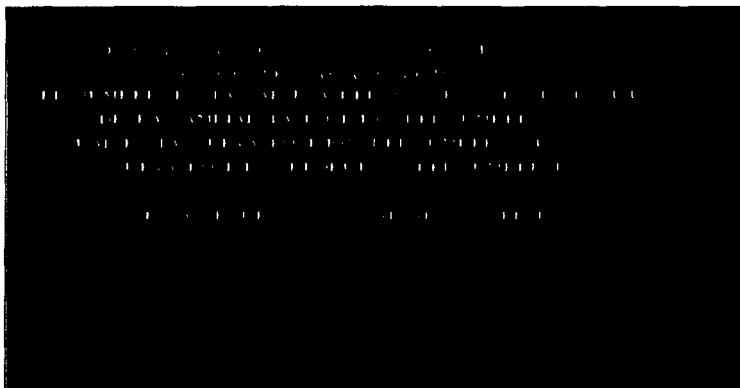


Figura 22.

**NOTA:**

**EN CASO DE SER ACTIVADO EL SISTEMA POR PRIMERA VEZ TIENE QUE SER CONFIGURADO EL SISTEMA, YA QUE CARECE DEL ARCHIVO EN EL QUE SE GUARDA LA CONFIGURACIÓN, Y EN DADO CASO QUE FALTE ESTE NO SE PODRÁ CORRER EL PROGRAMA YA QUE DESPLEGARA UN MENSAJE EN EL CUAL SE MENCIONA QUE SE CARECE DE ESTE ARCHIVO Y QUE HAY QUE CONFIGURARLO.**

## PRUEBA AUTOMÁTICA DE LOS PINES DE LA BASE DE PRUEBAS

La pantalla de la prueba automática de los pines de la base del sistema inicia con un mensaje recordando que debe retirarse cualquier integrado que se localiza en la base para poder iniciar la prueba (Figura 23).

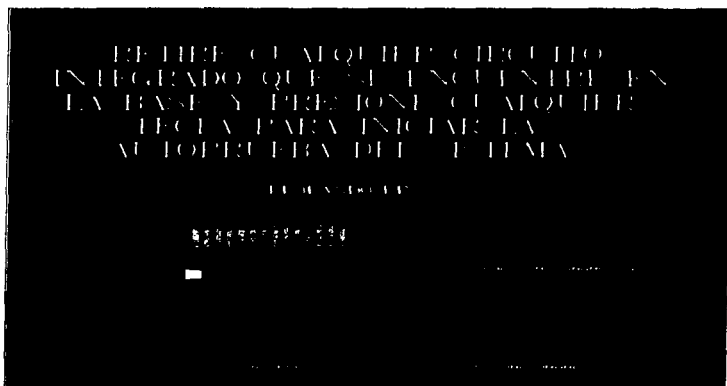


Figura 23.

Esta prueba tiene la función de comprobar el funcionamiento correcto de los pines de la base de pruebas donde se inserta el circuito integrado que se va a verificar; esta función envía datos a todos los pines que el programa puede manejar, ( el programa esta diseñado para controlar un total de 32 pines ).

La pantalla de prueba de pines muestra una base similar a la base donde se colocan los integrados de prueba, en la que se muestra por medio de colores el funcionamiento de los pines. Esto es, se coloca el pin de color verde cuando ya termino la prueba de ese pin y este esta funcionando correctamente, para el caso que el pin no funcione bien, este será mostrado en la pantalla de color rojo, esto puede ser síntoma de una falla eléctrica, o que el circuito que funciona con ese pin ó que ese pin no existe en la tarjeta (como puede ser el caso de los pines 15, 16, 17 y 18).

Para iniciar la prueba de pines de la base del sistema probador de circuitos integrados por medio de una PC, esta primeramente no debe tener ningún circuito montado sobre esta, ya que los datos que van a ser enviados hacia los pines no corresponden con alguna tabla de funcionamiento de los integrados, además que estos pueden provocar un mal funcionamiento del sistema y es posible que el integrado que se encuentre en el momento de la prueba de los pines se dañe, así como el circuito del sistema probador de integrados.

El resultado de la prueba de los pines de la base tiene que ser mayor o igual al 18.75% de pines en buen estado ( son un total de seis pines funcionando adecuadamente ), de lo contrario se mostrara en pantalla una ventana en la cual aparece el mensaje "EL SISTEMA NO ESTA FUNCIONANDO" (figura 24), también se exponen algunas de las causas posibles por las que se activo esta ventana y el modo de resolverlas, y después de presionar cualquier tecla se desactivara el programa y regresara la computadora al sistema operativo.



Figura 24.

#### **FUNCIONAMIENTO DEL PROGRAMA**

La pantalla siguiente después de la pantalla principal, ó después de la pantalla de configuración del puerto es la pantalla principal del programa (figura 25), la cual se muestra a continuación:

Como se puede apreciar esta pantalla consta de cinco opciones, las cuales son :

- Revisar Lista
- Checa Integrado
- Buscar
- Ayuda
- Salir



Figura 25.

### ***REVISAR LISTA***

En esta opción se activa presionando la tecla "R" , y al hacerlo esta nos mostrará una ventana con cuatro opciones (figura 26), las cuales son las siguientes;

1. - LISTA TTL.
2. - LISTA CMOS.
3. - REVISAR LA LISTA COMPLETA.

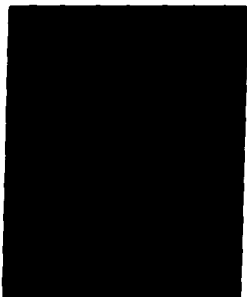


Figura 26.

Al seleccionar el número correspondiente con la información deseada, se nos mostrará la lista correspondiente a los integrados que están almacenados en el sistema listos para ser chequeados sin necesidad de editarlos.

Al seleccionar la opción 1 se pueden revisar los números de los integrados TTL ( lógica transistor transistor ) que ya están instalados dentro de la librería (figura 27).

Al seleccionar la opción 2 se pueden revisar los integrados CMOS ( integrados de oxido de metal ) que tiene almacenados el sistema en su librería (figura .

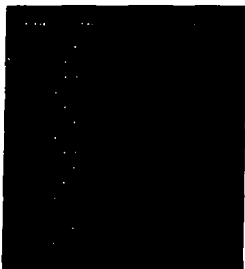


Figura 27.

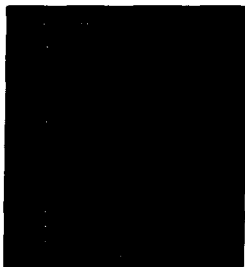


Figura 28.

Al seleccionar la opción 3 se pueden revisar los todos números de integrados que puede revisar el sistema, también aquí es el único lugar en donde se muestran los números de circuitos integrados editados por el usuario.



Para poder salir de esta opción y regresar al menú principal solo hay que presionar la tecla 'ESC'.

### ***CHECA INTEGRADO***

Esta opción se activa presionando la tecla "C", y nos muestra cuatro opciones (figura 29), las cuales son:

- 1. - INSERTA EL NUMERO DE INTEGRADO A REVISAR**
- 2. - INSERTA DATOS AL INTEGRADO DE PRUEBA  
MANUALMENTE**
- 3. - EDICIÓN**
- 4. - SALIR**



**Figura 29.**

En la opción ( 1 ) **INSERTA EL NÚMERO DE INTEGRADO A REVISAR** se pide el número del integrado que se va a checar, este no debe tener más de ocho caracteres, ya que si se rebasa este límite solo se buscará el integrado con los primeros ocho caracteres, los siguientes caracteres no serán tomado en cuenta para la búsqueda de los datos que van a probar el funcionamiento del integrado.

Después de haber insertado el número del integrado se preguntará si es correcto el número que se dio al programa, ya que puede equivocarse uno al teclearlo ó que se dude del número del integrado que va a probar, y si no es el mismo se puede quemar tanto el sistema que los prueba, como el integrado que se va a probar.

Si el número del integrado es aceptado como correcto por el usuario, la pantalla cambiará, y se mostrará en caso que sea un integrado ya incluido en el sistema , una compuerta lógica en caso de los circuitos lógicos o un esquema de un circuito integrado para los demás integrados como los puede ser para los contadores, decodificadores, multiplexores. etc.. Además que aparecerán dos opciones en las que se pide que se inicie el funcionamiento del sistema y otra en la que se puede regresar a la pantalla de selecciones inicial.

En caso que el número del integrado no exista, al iniciar la prueba del circuito integrado será desplegado un mensaje en el cual se mencionará que el número del integrado que se quiere probar no existe en la librería de circuitos integrados que se pueden revisar, y después de

presionar cualquier tecla se regresará a la pantalla principal para volver a realizar una nueva selección.

Para el caso que el número del integrado pedido este en la lista de integrados que se pueden checar se iniciará la comprobación del integrado después de presionar la tecla "I" (Iniciar) . Mostrándose en pantalla el número de pines que tiene el integrado de prueba, el número de compuertas que tiene el encapsulado, los datos que se están insertando a los pines del integrado siendo los valores de 0 para GND ó 1 lógico, 1 para 1 lógico, 2 para los pines de salida del integrado y 3 para el pin en donde se va a alimentar el integrado ( 5 volts ).

Al estar realizándose la prueba de funcionamiento, se mostrará también los pines que se están revisando, "SOLO SE REVISAN LOS PINES DE SALIDA DEL INTEGRADO", y de uno por uno, aunque algún integrado como los flip-flop tengan 2 pines con que responden, se muestra el valor que se esta leyendo desde la terminal o pin de respuesta y el valor que es el correcto ( VALOR OK ) que debe tener.

En caso que un pin no de la respuesta correcta, según su tabla de verdad sonara una alarma y la luz verde que tiene la compuerta cambiará por una luz roja, la cual nos señalará que esta defectuosa la compuerta (figura 30).

Al terminar la comprobación del funcionamiento del circuito integrado aparecerá en la parte inferior derecha de la pantalla el mensaje " COMPROBACIÓN FINALIZADA " y para regresar a la pantalla principal hay que presionar cualquier tecla.

Al terminar la comprobación del funcionamiento del circuito integrado aparecerá en la parte inferior derecha de la pantalla el mensaje " COMPROBACIÓN FINALIZADA " y para regresar a la pantalla principal hay que presionar cualquier tecla.



Figura 30.

**2. - MANUAL DE FUNCIONAMIENTO DEL SISTEMA PROBADOR DE CIRCUITOS SIN  
TENER QUE EDITAR EL CIRCUITO Y SIN QUE ESTE EXISTA EN LA LISTA DE CIRCUITOS  
YA ALMACENADOS**

En la opción ( 2 ) **INSERTAR DATOS AL INTEGRADO DE PRUEBA** **MANUALMENTE** se mandan datos a los primeros 14 pines de la base del integrado ( lado izquierdo de la base de integrados de prueba) y a los 14 últimos ( lado derecho ), donde se inicia siempre con todos los pines en un valor de cero lógico ó GND; esta acción se inicia al resonar la

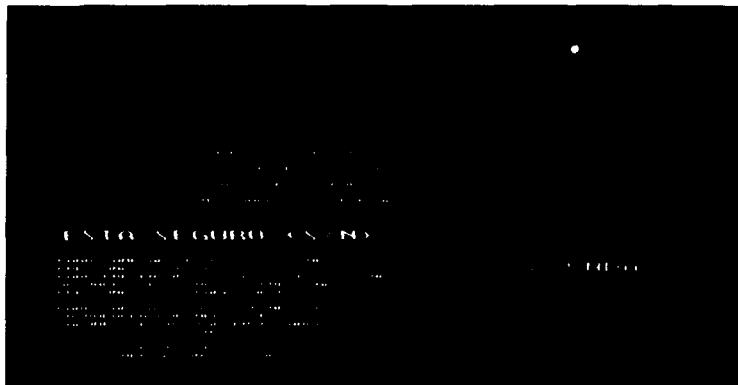


Figura 31.

tecla "C". entonces se pedirán los datos que van a ser insertados o puestos en la base del integrado de prueba. los pines serán pedidos según el orden de la base mostrada en la derecha de la pantalla, y estos serán insertados de uno por uno y respondiendo cada vez si esta correcto el dato insertado.

Los datos que están siendo insertados deben corresponder con la tabla amarilla que se muestra en la parte inferior izquierda de la pantalla, debido a que en caso que sea insertado un dato distinto y aceptarlo, aparecerá una pantalla indicando el error que ocurrió, así como permanecerán los datos de los pines sin cambiar de valor.

Estos datos insertados a los pines no se salvan en ningún archivo, y solo se mantienen estos hasta que estos son cambiados por la opción "C" ó se sale de esta opción.

La otra opción que esta en esta pantalla es la opción "R", la cual funciona para leer los datos que se encuentran en la base del integrado, esta solo lee "unos" y "ceros" , y no distingue si los datos son los que se mandaron poner o son los que esta respondiendo el integrado que se esta probando manualmente .

También existe una pantalla de ayuda que aparece presionando la tecla "F1", en la cual se menciona todo lo mencionado en los párrafos anteriores.

**ES IMPORTANTE SABER QUE INTEGRADO SE VA A PROBAR Y TENER LA SEGURIDAD DE EN QUE PINES SE VA A COLOCAR CADA DATO, DEBIDO A QUE EN**

CASO DE EQUIVOCARSE SE TIENE EL RIESGO DE QUEMAR EL PROBADOR DE CIRCUITOS INTEGRADOS, ASÍ COMO DE DAÑAR IRREMEDIABLEMENTE SU CIRCUITO.

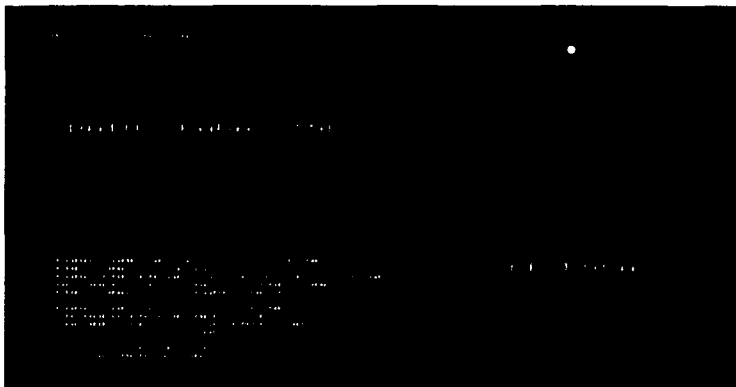


Figura 32.

Para salir de esta opción solo hay que presionar la tecla 'ESC' y entonces volverá al menú inicial.

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

**3 . - MANUAL DE FUNCIONAMIENTO DEL SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS LÓGICOS PARA UTILIZAR  
EL EDITOR DE CIRCUITOS.**

En la opción ( 3 ) **EDICIÓN** se pueden crear los archivos en los cuales se guardan las tablas de datos de entrada y salida de los integrados que se pueden probar y que no estén incluidos dentro de la lista de integrados, esto se activa al seleccionar la opción "3".

Al seleccionar la opción 3 inmediatamente aparece una pantalla de advertencia, en la cual se mencionan las características de la opción seleccionada y las consecuencias en caso de cometer algún error en la inserción de datos o de meter un número que pueda confundirse con otro integrado. Aparece también una pregunta en la cual si la respuesta es negativa se regresa al menú principal y si es afirmativa aparecerá la pantalla de edición.

En esta pantalla se tiene que presionar la tecla "E" para iniciar la edición del integrado que no se encuentra en la lista. Después de esto se pedirá que se inserte el número del integrado, el cual no debe exceder de ocho caracteres, y no debe ser igual a algún número de la lista de integrados existentes.

En caso de insertar un número igual a uno que ya estaba en la lista aparecerá un mensaje señalando que el número de integrado ya existe, y después de presionar una tecla aparecerá una opción en la cual se tiene que decidir si se sale de esta opción o se desea seleccionar otro número distinto del que se insertó para editar el integrado deseado.



**NO SE PUEDE SOBRESERIBIR UNA NUEVA TABLA DE DATOS DE UN INTEGRADO SOBRE UNA TABLA YA EXISTENTE.**

Al ser aceptado el número insertado todavía uno puede corregir este, respondiendo que no se esta seguro del nombre del integrado.

Inmediatamente después de aceptar el nombre del integrado se pregunta cual es el número de pines que tiene el integrado, y el cual debe ser un número par y no debe exceder de 28 pines, de lo contrario no se podrá continuar con el siguiente paso hasta que el dato insertado sea satisfactorio.

Continuando con la descripción del integrado se preguntará cuantas compuertas o componentes tiene el nuevo circuito, esto es, debe colocarse la cantidad de elementos internos que tiene el integrado, por ejemplo un integrado 74LS76 tiene dos flip-flop en un solo chip. Por lo tanto la respuesta al número de compuertas es de dos. El número máximo de compuertas por integrado es de doce.

En el siguiente paso se pregunta CUAL ES EL NUMERO DE PIN(ES) CON QUE RESPONDE LA COMPUERTA "NUMERO", esto se refiere a introducir el número de pin con el cual responde la compuerta "#", es decir, hay que poner el número de pin físico con que va a responder el integrado, por ejemplo, suponiendo que llamamos compuerta uno a la compuerta NAND del integrado 7400 la que sus entradas son las que tienen el pin 1 y el pin 2, entonces su pin de respuesta de este integrado será el pin 3 .

En los integrados ya incluidos se cuentan las compuertas comenzando desde la compuerta que esta más cerca del pin 1 del integrado hasta la ultima compuerta que es la que esta más cerca del ultimo pin.

Después se preguntará si existe otro pin de respuesta , esto es que si tenemos un convertidor de BINARIO A SIETE SEGMENTOS, nuestro elemento no tiene solo un pin de respuesta, sino que tiene SIETE, entonces se debe de responder que si existe otro pin de respuesta, y poner el siguiente número de pin. En caso que la respuesta sea negativa entonces se pedirá que se inserte el número de pin con que responde la compuerta "2" y volver a repetir el procedimiento mencionado anteriormente hasta que se llegue al número de compuertas insertado.

Ya que se insertaron todos los datos anteriores entonces se pedirá que se inserte la primera secuencia de datos que se van a dar al integrado, los cuales se deben de dar de acuerdo a la tabla de datos que se muestra en la pantalla, en donde 0 es GND, 1 es un uno lógico, 2 se marcan los pines que son de respuesta, 3 se usa para alimentar los pines con 5 volts ( este solo se usa para alimentar el circuito integrado de prueba, ya que proporciona un nivel de corriente mayor que cuando se le da una señal de un uno lógico ), y por ultimo el número 4 se utiliza para definir si ese pin se le va a dar un pulso de reloj, el cual inicia primero colocando un uno lógico y después le asigna un cero lógico, con lo cual se ahorra trabajo al no insertar dos veces los mismos datos a todos los pines restantes.

En el caso de poner el valor 2 que son los pines con los que responde el circuito integrado, debido al diseño del circuito del sistema probador de circuitos integrados, **ES INDISTINTO SI LA RESPUESTA DEL INTEGRADO ES DE SALIDA TOTEM POLE O DE COLECTOR ABIERTO.**



Figura 33.

Para el caso de integrados de tercer estado, cuando un circuito se encuentra en tercer estado ( ó alta impedancia a la salida del circuito ) el sistema solo leerá unos lógicos.

Después de insertar cada dato para cada pin se preguntará si se esta seguro de la información dada, ya que en caso de error se puede volver a insertar el dato correcto.

Después de completar un bloque de datos los cuales ocupan todos los pines del nuevo integrado se preguntará si es correcta la combinación de los datos dados, y en caso se existir algún error se borrara toda la información de esta combinación y se tendrá que volver a insertar, en caso de tener los datos insertados correctamente se preguntará a continuación si existe otra combinación de datos, esto se refiere a si se va a poner otra combinación de datos en todos los pines del integrados, y por lo tanto, habrá otro modo de respuesta para el integrado. Por ejemplo, el numero de distintas combinaciones que tiene un integrado 7400 es de cuatro ( que es un integrado con compuertas NAND ), las combinaciones de un 7404 es de dos ( que son compuertas NOT ).

Se puede dar el número requerido de combinaciones de datos en los nuevos integrados sin importar el tamaño, ya que estos datos están siendo almacenados en un archivo en el disco que se este usando.

Al finalizar la inserción de datos para el nuevo integrado se preguntará si los datos que fueron insertados son correctos. pues en dado caso de tener duda de estos, ó de existir algún error estos datos erróneos pueden dañar el sistema cuando sean probados los integrados con este nuevo número. Por lo tanto en caso de existir error se borrarán todos los datos insertados, y en caso de ser correctos estos serán codificados y almacenados en el directorio de LIBRERÍA para ser utilizados para la prueba de los nuevos integrados.

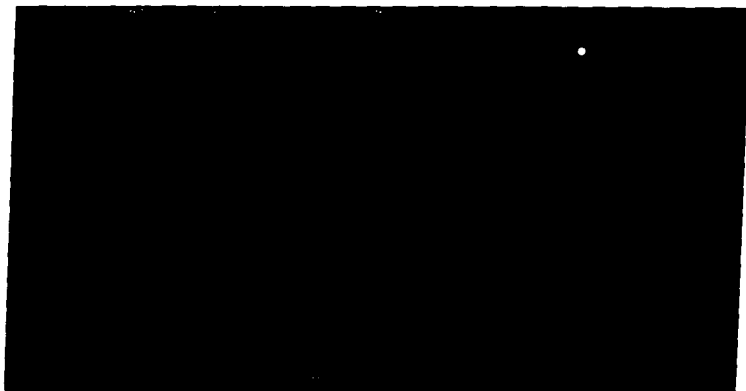


Figura 34.

**NOTA:** Para cuando sea dado de alta un número de integrado, el cual sea incorrecto el número de matrícula, ó este tenga errores en la tabla de funcionamiento y estos se termine el procedimiento completo anexándolos a la librería de circuitos integrados del sistema, se pueden borrar desde sistema operativo, cambiándose al directorio LIBRERÍA y ahí eliminando el numero de integrado equivocado con extensiones ".IN" y ".OUT".

Por ultimo la opción **(4) SALIR** regresa al menú principal.

**NOTA:** No todo el tiempo esta habilitada la tecla "ESC", solo funciona cuando esta escrita en color blanco. Para poder salir hay que continuar con el procedimiento de edición hasta que aparezca esta opción nuevamente en la parte inferior de la pantalla.

### ***BUSCAR***

En esta opción se activa presionando la letra "B" , y la cual tiene la función de dar el número de circuito integrado que no se sabe si esta en la lista de circuitos de prueba, y al dar el numero del integrado que se quiere probar siempre y cuando no exceda de ocho caracteres será buscado en la librería de circuitos, en caso que si exista pondrá en la pantalla el numero de circuito integrado, de lo contrario se desplegara el mensaje que el integrado insertado no existe.



Figura 35.

### ***AYUDA***

Esta opción se activa presionando la letra "A" , y la cual despliega una breve información de que función tiene cada menú de la pantalla principal, activándose cada una dependiendo del número seleccionado.

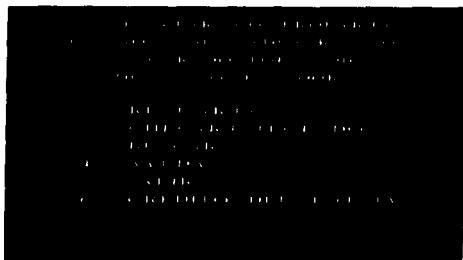


Figura 36.

### ***SALIR***

Esta opción se activa presionando la letra "S" , y la cual abre una ventana en la que se pregunta si se quiere salir al sistema operativo, si la respuesta es afirmativa entonces desplegara el mensaje de desconectar la interface y presionar la letra "D", esto con el fin de evitar cualquier problema con los datos del puerto paralelo que controla el sistema.

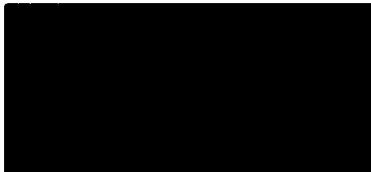


Figura 37.

### **EJEMPLO DE PRUEBA DE UN INTEGRADO**

Estando dentro del menú principal hay que presionar la letra "C" ( de la opción de Checar integrado ) , después hay que seleccionar la opción 1 , y dar respuesta afirmativa cuando se pregunta si uno esta seguro que se quiere realizar esa actividad.

A continuación aparecerá un recuadro y una indicación de insertar el numero de integrado, y es aquí donde se debe insertar el numero de integrado que se desea revisar, para este ejemplo será el integrado numerado como 74LS00. y después de haber tecleado el número hay que presionar la tecla "ENTER" e inmediatamente aparecerá un mensaje "Esta seguro (S/N)", al cual si no se cometió ningún error al teclearlo hay que teclear la tecla "S" ó "ENTER", lo cual hará que aparezca una pantalla con una compuerta lógica NAND y también aparecerá un cuadro en el cual se pregunta si se quiere "Iniciar" y otra opción es "Regresar al menú principal", en donde se tiene que presionar la tecla "I" para iniciar la prueba del integrado.



Antes de presionar la tecla "I" hay que insertar el integrado en la base de pruebas, fijándose que la muesca o marca que tiene el circuito integrado coincida con la marca que esta en la base de pruebas.

Ya realizado lo anterior hay que presionar la tecla "I" y entonces en la pantalla se desplegaran los datos del integrado, así como los datos que están siendo insertados al mismo, aparecerá también un determinado numero de renglones los cuales tienen la función de marcar el número de compuertas que tiene el integrado que se esta probando y enfrente de cada uno aparecerá un círculo verde, el cual nos indica que la compuerta esta en buen estado, y que cambiara a color rojo cuando la compuerta este defectuosa.

La prueba habrá finalizado cuando se muestre el mensaje "COMPROBACIÓN FINALIZADA", entonces se podrá retirar el integrado revisado de la base y se sabrá que compuertas están defectuosas.

Al presionar cualquier tecla después de terminada la comprobación volverá el sistema a la pantalla principal para volver a realizar otra selección.

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**CONCLUSIONES**

## CONCLUSIONES

Al realizar este proyecto para tesis no surgió solamente una conclusión, debido a que con ella se desarrollo una metodología, tanto para la descripción del proyecto, como también las habilidades para elaborar un circuito complejo basándose en el conocimiento de algunos de los circuitos sencillos aprendidos en el transcurso de la carrera. Así también se desarrollaron habilidades para utilizar otros circuitos, ya que en las primeras pruebas del sistema probador de circuitos integrados este no funcionaba adecuadamente, y después de analizar la causa de la falla, y verificar que esta no fuese por armado de circuito, ó por lógica de funcionamiento, se desarrollo otro circuito de mayor potencia para resolver el problema.

Otra conclusión que se obtuvo en el desarrollo del proyecto es que se debe de seguir siempre una metodología para desarrollar un proyecto, ya que si este se realiza en desorden se pierde información y tiempo, ya que hay que volver a buscar esta información perdida.

La metodología o pasos seguidos para desarrollar el presente proyecto de tesis fue el siguiente:

- I .- Seleccionar el proyecto para el desarrollo de la tesis.
- II .- Buscar información para desarrollar el proyecto seleccionado para tesis.
- III .- Depurar la información encontrada.
- IV .- Realizar pruebas con circuitos simples, producto de la información depurada.

- V .- Armar un prototipo del proyecto y probar su funcionamiento.
- VI .- Ajustar y corregir el funcionamiento erróneo del prototipo, documentando los cambios que se realizaron al proyecto y las causas por las cuales se realizaron esos cambios.
- VII .- Probar el funcionamiento adecuado del prototipo.
- VIII .- Recopilar la información total utilizada para la realización del proyecto, ampliarla y ordenarla.
- IX .- Conclusiones acerca del proyecto.

Esta lista de pasos en el desarrollo de la tesis fue de gran ayuda, ya que con esta se consiguió el desarrollo, la investigación y que sin ella no se hubiese podido llegar al funcionamiento correcto del proyecto denominado Probador de Circuitos Integrados por Medio de una Computadora en el lapso de tiempo predeterminado.

Otra conclusión obtenida con el desarrollo del Probador de Circuitos Integrados por Medio de una Computadora es que con este sistema se van a poder probar circuitos lógicos digitales (TTL, CMOS, RTL, etc.), y en si cualquier tipo de integrado al cual se le pueda asignar una tabla de funcionamiento similar a una tabla lógica de verdad, ya que el diseño del sistema solo puede mandar valores de voltaje de 5 volts y cero volts.

Este proyecto es una herramienta para agilizar y desarrollar con mayor eficacia las practicas desarrolladas en los laboratorios en las que se utilicen circuitos integrados lógicos de los cuales se dude su correcto funcionamiento.

La conclusión final acerca de este proyecto es que se puede desarrollar para probar cualquier tipo de circuito integrado y en si cualquier tipo de tarjetas, siempre y cuando sean desarrollados los circuitos necesarios para que el probador pueda generar señales analógicas, asi como direccionarlas sobre las diferentes terminales que van a ser conectados a los circuitos que van a ser probados, y también sean implementados los circuitos que reconozcan las señales analógicas, y estas sean insertadas a la computadora de manera digital, y en caso de ser necesario implementar los circuitos que proporcionen un mayor voltaje a la salida para probar circuitos de mayor capacidad de voltaje. En general, este proyecto puede ser la base para desarrollar otros dispositivos para comprobar el funcionamiento de cualquier circuito, siempre y cuando no se exceda de los limites de frecuencia que nos puede proporcionar la computadora.

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**APENDICES**

**APENDICE A**  
**LISTADO DEL PROGRAMA DE CONTROL**

**Listado del programa de control CITEST.C**

```

#include <stdlib.h>
#include <stdio.h>
#include <graphics.h>
#include <conio.h>
#include <dir.h>
#include "BELL.C"
#include "GRAFICOS.C"
#include "MARCOFC.C"
#include "ALERTA.C"
#include "AMP.OP.C"
#include "AND1.C"
#include "BUFFER.C"
#include "BUFINV.C"
#include "BUFCONTN.C"
#include "BUFINVEN.C"
#include "CIR.INT.C"
#include "CLEARPIN.C"
#include "FLIPFLOP.C"
#include "NAND1.C"
#include "NOR1.C"
#include "NOT1.C"
#include "OR1.C"
#include "TRSFORMA.C"
#include "CADENA.C"
#include "WINDOW1.C"
#include "CLEARWIN.C"
#include "SALIR.C"
#include "DECOD_IN.C"
#include "OUTPIN.C"
#include "MODO_0.C"
#include "MODO_3.C"
#include "LOAD_INT.C"
#include "REV_LIST.C"
#include "TTL.C"
#include "CMOS_LIST.C"
#include "AND3INF.C"
#include "AND3INF.C"
#include "AND4INF.C"
#include "NOT_INF.C"
#include "NAND2INF.C"
#include "NAND3INF.C"
#include "NAND4INF.C"
#include "NAND5INF.C"
#include "NAND12IF.C"

#include "NOR2INF.C"
#include "NOR3INF.C"
#include "NOR4INF.C"
#include "CHECKMENU.C"
#include "PIN_IN.C"
#include "RESUL_1.C"
#include "COD.C"
#include "OUTERROR.C"
#include "LEE_DATO.C"
#include "INREADAT.C"
#include "DET_BLK.C"
#include "INS_DATO.C"
#include "COMPRU2.C"
#include "GATE.C"
#include "TESTSYST.C"
#include "AYUDA.C"
int main(void)
{
FILE *CFG;
// char *endptr;
char cfgptr;
int PUERTO_A,PUERTO_B,PUERTO_C;
int x1,y1,x2,y2,a;
int alerta;
char r;
char FALLA;
GRAFICOS();
system("PRESENTA.COM");
/* PROGRAMACION DEL PPI :
PUERTO A SALIDA
PUERTO B ENTRADA
PUERTO C SALIDA */
if ((CFG = fopen("CITEST.CFG", "rt")) == NULL)
cleardevice();
settextstyle(TRIPLEX_FONT, HORIZ_DIR,4);
outtextxy(47,200,"NO EXISTE EL ARCHIVO
CITEST.CFG");
outtextxy(47,250," PRIMERO CONFIGURE EL
PUERTO");
BELL();
getch();
closegraph();
exit(1);}
cfgptr=fgetc(CFG);
fclose(CFG);
switch(cfgptr){

```

```

case '1':
    outportb (0x203,0x82);
    PUERTO_A=0x200;
    PUERTO_B=0x201;
    PUERTO_C=0x202;
    break;
case '2':
    outportb (0x223,0x82);
    PUERTO_A=0x220;
    PUERTO_B=0x221;
    PUERTO_C=0x222;
    break;
case '3':
    outportb (0x243,0x82);
    PUERTO_A=0x240;
    PUERTO_B=0x241;
    PUERTO_C=0x242;
    break;
case '4':
    outportb (0x263,0x82);
    PUERTO_A=0x260;
    PUERTO_B=0x261;
    PUERTO_C=0x262;
    break;
case '5':
    outportb (0x283,0x82);
    PUERTO_A=0x280;
    PUERTO_B=0x281;
    PUERTO_C=0x282;
    break;
case '6':
    outportb (0x2a3,0x82);
    PUERTO_A=0x2a0;
    PUERTO_B=0x2a1;
    PUERTO_C=0x2a2;
    break;
case '7':
    outportb (0x2c3,0x82);
    PUERTO_A=0x2c0;
    PUERTO_B=0x2c1;
    PUERTO_C=0x2c2;
    break;
case '8':
    outportb (0x2e3,0x82);
    PUERTO_A=0x2e0;
    PUERTO_B=0x2e1;
    PUERTO_C=0x2e2;
    break;
default:
    cleardevice();
    settextstyle(TRIPLEX_FONT,
HORIZ_DIR,4);
    outtextxy(47,200,"EL ARCHIVO
CTITEST.CFG ES ERRONEO");
    outtextxy(47,250," PRIMERO
RECONFIGURE EL PUERTO");
    BELL();
    getch();
    cclosegraph();
    exit(1);
}
FALLA=TESTSYST(PUERTO_A,PUERTO_B,PUE
RTO_C);
if(FALLA==1) (exit(1);)
inicio:
MARCOPC():
reinicio:
r=toupper(getch());
switch(r){
case 0x0:
    getch();
    BELL();
    goto reinicio;
case 'A':
    //AYUDA
    AYUDA();
    cleardevice();
    goto inicio;
case 'B':
    //BUSCAR
    x1=180,y1=100,x2=260,y2=200;
    WINDOW1(x1,y1,x2,y2);
    outtextxy(200,150,"Inserta numero de
integrado");
    getchxy(32,14).printf("No
CADENA(0,0,0,36,14,2);
    goto inicio;
case 'C':
    //CHECAR INTEGRADO
    x1=130,y1=80,x2=180,y2=300;
    WINDOW1(x1,y1,x2,y2);
    outtextxy(140,100,"1.-INSERTA EL
NUMERO");
    outtextxy(140,115," DEL INTEGRADO
QUE");
    outtextxy(140,130," SE VA A
REVISAR");
    outtextxy(140,150,"2.-INSERTA DATOS
AL");
    outtextxy(140,165," INTEGRADO DE
PRUEBA");
    outtextxy(140,180,"
MANUALMENTE.");
    outtextxy(140,195,"3.- EDICION.");
    outtextxy(140,210,"4.- SALIR.");
    outtextxy(155,225,"SELECCION : ");
    opcion1;
    setcolor(9);

```



```

SU*);          outtextxy(140,280," ESTA SEGURO DE
               outtextxy(140,300," OPCION (S/N)*);
               outtextxy(255,225,"1");
               outtextxy(255,225,"2");
               outtextxy(255,225,"3");
               outtextxy(255,225,"4");
               setcolor(15);
               a=(getch());
               if(a=='1') outtextxy(255,225,"1");
               if(a=='2') outtextxy(255,225,"2");
               if(a=='3') outtextxy(255,225,"3");
               if(a=='4') outtextxy(255,225,"4");
               if(a==0X1b) break;
               if(a==0X0) (BELL(),getch());goto
opcion1;
opcion1;      if(a<0X31||a>0X34){BELL();goto
SU*);          outtextxy(140,280," ESTA SEGURO DE
               outtextxy(140,300," OPCION (S/N)*);
               x1=toupper(getch());
               if(x1=='S'||x1==0xd){
                   setcolor(9);
                   outtextxy(140,280," ESTA
SEGURO DE SU*);
                   outtextxy(140,300," OPCION
(S/N)*);
                   setcolor(15);}
               else goto opcion1;
               switch(a){
                   case '1':
CADENA(A,PUERTO_A,PUERTO_B,PUERTO_C,24,
17,1);
                   break;
                   case '2':
                       cleardevice();
                       system("MANUAL.COM");
                       break;
                   case '3':
                       cleardevice();
                       alerta=ALERTA();
                       cleardevice();
if(alerta==1){BELL(),system("EDICION.COM");}
                   break;
                   case 0X1b:
                   case '4':break;
                   default : BELL();
                       break;
                       goto opcion1;}
                   case 'R':
//REVISION DE LISTA
               REV_LISTO;
               break;
               case 0x1b:
               case 'S':
                   //SALIR
                   x1=150,y1=120,x2=305,y2=200;
                   WINDOW1(x1,y1,x2,y2);
                   x1=SALIR(PUERTO_A,PUERTO_C);
                   if(x1==1){
                       cleardevice();
                       goto inicio;}
                   default: BELL();
                       goto reinicio.
                   }
                   cleardevice();
                   goto inicio.
                   }
               "BELL.C"
               void BELL(void)
               {
                   sound(1000);
                   delay(300);
                   nosound();
               }
               "GRAFICOS.C"
               // DECLARACION DEL PROTOTIPO
               void GRAFICOS (void);
               // FUNCION PARA DIBUJAR UN
               AMPLIFICADOR OPERACIONAL.
               void GRAFICOS()
               {
                   /* AUTODETECCION */
                   int gdriver = DETECT, gmode, errorcode;
                   /* INICIALIZACION DE GRAFICOS Y
VARIABLES */
                   initgraph(&gdriver, &gmode, "");
                   /* RESULTADO DE LA LECTURA DE
INICIALIZACION */
                   errorcode = graphresult();
                   if (errorcode != grOk) /* OCURRIO UN ERROR
*/
                   {
                       printf("ERROR DE GRAFICOS: %s\n",
grapherrormsg(errorcode));
                       printf("PRESIONE CUALQUIER TECLA
PARA LIBERAR.");
                       getch();
                       exit(1); /* FIN DEL CODIGO DE ERROR */
                   }
               }
               "MAROPC.C"

```

```

void MARCOPC()
{
// FONDO DE LA IMAGEN PRINCIPAL
setbkcolor(BLACK);
setcolor(WHITE);
settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
setfillstyle(1,1);
bar(1,33,638,478);
rectangle(2,32,639,479);
// MARCO DEL MENU DE OPCIONES
setfillstyle(1,7);
bar(2,4,638,30);
rectangle(1,4,639,30);
setcolor(4);
outtextxy(15,15,"R");
outtextxy(160,15,"C");
outtextxy(340,15,"B");
outtextxy(440,15,"A");
outtextxy(540,15,"S");
setcolor(0);
outtextxy(15,15," evisar lista");
outtextxy(160,15," hecar integrado");
outtextxy(340,15," usar");
outtextxy(440,15," yuda");
outtextxy(540,15," alir");
}

"ALERTA.C"
// FUNCION PARA PONER LA PANTALLA
ALERTA.
ALERTA(void)
{
int answ;
setbkcolor(RED);
// MARCO 1
rectangle(1,1,639,479);
// MARCO 2
rectangle(4,4,635,475);
settextstyle(TRIPLEX_FONT, HORIZ_DIR,8);
outtextxy(40,15,"IMPORTANTE ! !");
settextstyle(TRIPLEX_FONT, HORIZ_DIR,2);
outtextxy(30,100," HA SELECCIONADO LA
OPCION PARA CREAR ARCHIVOS.");
outtextxy(20,125," EN ESTA OPCION EL
USUARIO TENIENDO UN INTEGRADO EL");
outtextxy(20,150,"CUAL NO SE ENCUENTRE EN
LA LISTA DE INTEGRADOS EN LA");
outtextxy(20,175,"LIBRERIA. LO PUEDE EDITAR
EN ESTA OPCION.");
outtextxy(40,200," TENGA EN CUENTA QUE EL
SISTEMA SOLO PROPORCIONA");
outtextxy(20,225,"DATOS DIGITALES, POR LO
TANTO, EL NUEVO CIRCUITO");

outtextxy(20,250,"INTEGRADO DEBE
RESPONDER A ESTOS NIVELES DE
VOLTAJE.");
outtextxy(20,275," ASEGURESE DE TENER LA
TABLA DE FUNCIONAMIENTO DEL.");
outtextxy(20,300,"INTEGRADO QUE SE VA A
EDITAR.");
outtextxy(20,325," SE DEBEN METER LOS
DATOS CORRECTAMENTE DEL NUEVO");
outtextxy(20,350," INTEGRADO, SINO, EXISTE
EL RIESGO DE DETERIORAR");
outtextxy(20,375," IRREMEDIABLEMENTE EL
SISTEMA.");
outtextxy(150,425,"DESEA CONTINUAR
(S/N)?.");
answ=toupper(getch());
if(answ=="S"||answ=="O")return 1;
settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
setbkcolor(1);
return 0;
}

"AMP-OP.C"
// FUNCION PARA DIBUJAR UN
AMPLIFICADOR OPERACIONAL.
void AMP_OP()
{
int xi=0,yi=0;
setbkcolor(1);
// AMPLIFICADOR OPERACIONAL
line(xi+80,yi+120,xi+122,yi+100);
line(xi+80,yi+80,xi+122,yi+100);
line(xi+80,yi+80,xi+80,yi+120);
line(xi+122,yi+100,xi+134,yi+100);
line(xi+80,yi+90,xi+70,yi+90);
line(xi+80,yi+110,xi+70,yi+110);
settextstyle(SMALL_FONT, HORIZ_DIR,3);
outtextxy(xi+85,yi+86,"+");
outtextxy(xi+85,yi+106,"-");
}

"AND1.C"
// FUNCION PARA DIBUJAR UNA COMPUERTA
AND
void AND1()
{
int xi=0,yi=0;
setbkcolor(1);
/* AND*/
arc(xi+100,yi+100,270,90,20);
line(xi+100,yi+120,xi+80,yi+120);
line(xi+100,yi+80,xi+80,yi+80);
line(xi+80,yi+80,xi+80,yi+120);
line(xi+80,yi+110,xi+60,yi+110);
}

```

```

line(xi+80 ,yi+90 ,xi+60 ,yi+90);
line(xi+121,yi+100,xi+132,yi+100);
}

```

```

"BUFFER C"
// FUNCION PARA DIBUJAR UN BUFFER (tercer
estado).
void BUFFER()

```

```

{
int xi=0,yi=0;
setbkcolor(1);
// BUFFER (tercer estado)
line(xi+80 ,yi+120 ,xi+122,yi+100);
line(xi+80 ,yi+80 ,xi+122,yi+100);
line(xi+80 ,yi+80 ,xi+80 ,yi+120);
line(xi+80 ,yi+100,xi+60 ,yi+100);
line(xi+123,yi+100,xi+134,yi+100);
line(xi+60 ,yi+130 ,xi+105,yi+130);
line(xi+105,yi+130,xi+105,yi+110);
}

```

```

"BUFINV.C"
// FUNCION PARA DIBUJAR UN BUFFER (tercer
estado).
void BUFINV()

```

```

{
int xi=0,yi=0;
setbkcolor(1);
// BUFFER INVERSOR (tercer estado con control)
line(xi+80 ,yi+120,xi+122,yi+100);
line(xi+80 ,yi+80 ,xi+122,yi+100);
line(xi+80 ,yi+80 ,xi+80 ,yi+120);
line(xi+80 ,yi+100,xi+60 ,yi+100);
line(xi+128,yi+100,xi+134,yi+100);
line(xi+105,yi+130,xi+105,yi+110);
line(xi+60 ,yi+130,xi+105,yi+130);
circle(xi+125,yi+100,3);
}

```

```

"BUFCONTN.C"
// FUNCION PARA DIBUJAR UN BUFFER (tercer
estado con control negado).
void BUFCONTN()

```

```

{
int xi=0,yi=0;
setbkcolor(1);
// BUFFER (tercer estado con control negado)
line(xi+80 ,yi+120,xi+122,yi+100);
line(xi+80 ,yi+80 ,xi+122,yi+100);
line(xi+80 ,yi+80 ,xi+80 ,yi+120);
line(xi+80 ,yi+100,xi+60 ,yi+100);
line(xi+123,yi+100,xi+134,yi+100);
line(xi+60 ,yi+130,xi+105,yi+130);
line(xi+105,yi+130,xi+105,yi+116);
}

```

```

circle(xi+105,yi+112,3);
}

```

```

"BUFINV.CN C"
// FUNCION PARA DIBUJAR UN BUFFER
INVERSOR
// (tercer estado con control negado).
void BUFINV.CN()

```

```

{
int xi,yi,i;
// BUFFER INVERSOR (tercer estado con control
negado)
line(xi+80 ,yi+120,xi+122,yi+100);
line(xi+80 ,yi+80 ,xi+122,yi+100);
line(xi+80 ,yi+80 ,xi+80 ,yi+120);
line(xi+80 ,yi+100,xi+60 ,yi+100);
line(xi+128,yi+100,xi+134,yi+100);
line(xi+60 ,yi+130,xi+105,yi+130);
line(xi+105,yi+130,xi+105,yi+116);
circle(xi+105,yi+112,3);
circle(xi+125,yi+100,3);
}

```

```

"CIR-INT C"
// FUNCION PARA DIBUJAR UN CIRCUITO
INTEGRADO
void CIR_INT()

```

```

{
int xi=15,yi=30,i=0;
setbkcolor(1);
// CIRCUITO INTEGRADO
line(xi+130,yi+50 ,xi+190,yi+70 );
line(xi+130,yi+50 ,xi+60 ,yi+180);
line(xi+60 ,yi+180,xi+120,yi+200);
line(xi+120,yi+200,xi+190,yi+70 );
line(xi+60 ,yi+190,xi+120,yi+210);
line(xi+120,yi+210,xi+190,yi+80 );
line(xi+60 ,yi+190,xi+60 ,yi+180);
line(xi+120,yi+210,xi+120,yi+200);
line(xi+190,yi+80 ,xi+190,yi+70 );
//ramura
setfillstyle(1,15);
fillellipse(xi+97,yi+184,5,7);
line(xi+92,yi+180,xi+89,yi+188);
line(xi+93,yi+181,xi+90,yi+189);
line(xi+94,yi+182,xi+91,yi+189);
line(xi+95,yi+182,xi+92,yi+189);
//pines
for (i=0; i<7; i++){
//derechos
yi=yi+2;
line(xi+125,yi+195,xi+135,yi+197);
line(xi+128,yi+190,xi+138,yi+192);
line(xi+135,yi+197,xi+135,yi+203);
}
}

```

```

line(xi+138,yi+192,xi+138,yi+199);
line(xi+136,yi+203,xi+136,yi+210);
line(xi+137,yi+199,xi+137,yi+209);
// Izquierdos
line(xi+60,yi+173,xi+50,yi+171);
line(xi+63,yi+168,xi+53,yi+166);
line(xi+50,yi+171,xi+53,yi+166);
line(xi+50,yi+171,xi+53,yi+185);
line(xi+54,yi+173,xi+53,yi+186);
xi=xi+10;
yi=yi-20;
}
}

"CLEARPIN.C"
//LIMPIAR PINES
void CLEARPIN(PUERTO_A,PUERTO_C)
{
  int C=0,x=0;
  outputb (PUERTO_C,0Xf0);
  outputb (PUERTO_A,0Xaa);
  while(x<10){
    while(C<=16)
      { outputb (PUERTO_C,239+C);
        C++;
      }
    x++;
  }
}

"FLIPFLOP.C"
// FUNCION PARA DIBUJAR UN FLIP-FLOP
void FLIPFLOP()
{
  int xi=40,yi=15;
  setbkcolor(1);
// FLIP-FLOP
line(xi+60,yi+60,xi+100,yi+60);
line(xi+60,yi+120,xi+100,yi+120);
line(xi+60,yi+60,xi+60,yi+120);
line(xi+100,yi+60,xi+100,yi+120);
line(xi+60,yi+87,xi+68,yi+90);
line(xi+60,yi+93,xi+68,yi+90);
//RELOJ NEGADO
circle(xi+57,yi+90,3);
line(xi+54,yi+90,xi+45,yi+90);
//ENTRADA(S)
line(xi+60,yi+75,xi+45,yi+75);
line(xi+60,yi+105,xi+45,yi+105);
//SALIDA(S)
line(xi+100,yi+75,xi+115,yi+75);
line(xi+100,yi+105,xi+115,yi+105);
//PRESET Y CLEAR
line(xi+80,yi+60,xi+80,yi+50);

```

```

line(xi+80,yi+120,xi+80,yi+130);
}

```

```

"NaNDI.C"
//DECLARACION DEL PROTOTIPO
void NAND1(void);
// FUNCION PARA DIBUJAR UNA COMPUERTA
NAND.
void NAND1()
{

```

```

  int gdriver = DETECT, gmode, errorcode;
  int xi=0,yi=0;
  initgraph(&gdriver, &gmode, "");
  setbkcolor(1);
//NAND
arc(xi+100,yi+100,xi+270,90,20);
line(xi+100,yi+120,xi+80,yi+120);
line(xi+100,yi+80,xi+80,yi+80);
line(xi+80,yi+80,xi+80,yi+120);
line(xi+80,yi+110,xi+60,yi+110);
line(xi+80,yi+90,xi+60,yi+90);
circle(xi+125,yi+100,3);
line(xi+129,yi+100,xi+134,yi+100);
}

```

```

"NOR1.C"
// FUNCION PARA DIBUJAR UNA COMPUERTA
NOR.
void NOR1()
{

```

```

  int xi=60,yi=0;
  setbkcolor(1);
//NOR
arc(xi+150,yi+106,20,110,35);
arc(xi+150,yi+80,250,340,35);
arc(xi+108,yi+92,328,32,35);
line(xi+140,yi+105,xi+125,yi+105);
line(xi+140,yi+80,xi+125,yi+80);
circle(xi+186,yi+94,3);
line(xi+189,yi+94,xi+195,yi+94);
}

```

```

"NOT1.C"
// FUNCION PARA DIBUJAR UNA COMUERTA
NOT
void NOT1()
{

```

```

  int xi=0,yi=0;
  setbkcolor(1);
//NOT
line(xi+80,yi+120,xi+122,yi+100);
line(xi+80,yi+80,xi+122,yi+100);
line(xi+80,yi+80,xi+80,yi+120);
line(xi+80,yi+100,xi+60,yi+100);

```

```

circle(xi+125,yi+100,3);
line(xi+129,yi+100,xi+134,yi+100);
}

"OR1.C"
// FUNCION PARA DIBUJAR UNA COMPUERTA
OR
void OR1()
{
  int xi=-60,yi=0;
  setbcolor(1);
  //OR
  arc(xi+150,yi+106,20,110,35);
  arc(xi+150,yi+80,250,340,35);
  arc(xi+108,yi+92,328,32,35);
  line(xi+140,yi+105,xi+125,yi+105);
  line(xi+140,yi+80,xi+125,yi+80);
  line(xi+183,yi+94,xi+192,yi+94);
}

"TRSFORMA.C"
void TRSFORMA(char *string,int PUERTO_A,int
PUERTO_B,int PUERTO_C,int search,int checa)
{
  char ACTUAL[MAXDIR];
  char ext1[5]=".IN";
  int NUM_INT=0;
  char *p;
  char position,letra;
  char CADENA2[20]="";
  char
NUM0[2]="0",NUM1[2]="1",NUM2[2]="2",NUM3[
2]="3",NUM4[2]="4",
  char
NUM5[2]="5",NUM6[2]="6",NUM7[2]="7",NUM8[
2]="8",NUM9[2]="9";
  char *endptr;
  char CAD1[9]="",CAD2[9]="";
  char LETRA1[9]="",LETRA2[9]="";
  char cfgpto;
  int
largo,i,ptr0,ptr1,ptr2,ptr3,ptr4,ptr5,ptr6,ptr7,ptr8,ptr
9;
  int N_INT;
  strcpy(LETRA1,"");
  strcpy(LETRA2,"");
  strcpy(CADENA2,string);
  largo=strlen(CADENA2);
  for (i=0; i<8; i++){
  if(largo>0){
  strcpy(CAD1,"");
  strcpy(CADENA2);
  strcat(CAD1,CADENA2,largo);
  strcat(LETRA1,CAD1,1);
  ptr0=strnicmp(LETRA1,NUM0,2);
  ptr1=strnicmp(LETRA1,NUM1,2);
  ptr2=strnicmp(LETRA1,NUM2,2);
  ptr3=strnicmp(LETRA1,NUM3,2);
  ptr4=strnicmp(LETRA1,NUM4,2);
  ptr5=strnicmp(LETRA1,NUM5,2);
  ptr6=strnicmp(LETRA1,NUM6,2);
  ptr7=strnicmp(LETRA1,NUM7,2);
  ptr8=strnicmp(LETRA1,NUM8,2);
  ptr9=strnicmp(LETRA1,NUM9,2);
  if
(!ptr0||!ptr1||!ptr2||!ptr3||!ptr4||!ptr5||!ptr6||!ptr7||!ptr8||
!ptr9)
    strcat(LETRA2,LETRA1,1);
  strcpy(LETRA1,"");
  strcpy(CADENA2);
  strcpy(CAD2,"");
  largo--;
  strcat(CAD2,CADENA2,largo);
  strcpy(CADENA2,CAD2);
  }
  }
  strcpy(LETRA2);
  N_INT=atoi(LETRA2,&endptr,10);
  if (search==0)
  {LOAD_INT(N_INT,0,string,PUERTO_A,PUERTO
_B,PUERTO_C);}
  if (search==1){
  getchdir(0,ACTUAL);
  chdir("LIBRERIA");
  strcpy(CADENA2,"");
  strcat(CADENA2,string,8);
  strcat(CADENA2,ext1,4);
  p=searchpath(CADENA2);
  chdir("\");
  chdir(ACTUAL);
  if(p=NULL){
  cleardevice();
  settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
  outtextxy(165,150,"NO EXISTE EL
INTEGRADO");
  outtextxy(260,200,string);
  BELL();}
  else {
  LOAD_INT (NUM_INT,checa);
  settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
  outtextxy(400,150,string);
  outtextxy(280,50,"EL NUMERO DE
INTEGRADO");
  outtextxy(270,100,"INSERTADO ESTA
DISPONIBLE.");
  settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
  getch();
  cleardevice();}
}

```

```

}

"CADENA.C"
void
CADENA(PUERTO_A,PUERTO_B,PUERTO_C,L
OC_X,LOC_Y,ORJEN)
{
    char
    string[10],string2[10],caracter[2]="",help[10]="";
    char posicion,letra;
    int s,a;

    INTEGRADO
    if (ORJEN==1){
        gotoxy(24,17).printf(" ");
        setcolor(9);
        outtextxy(140,280,"ESTA SEGURO DEL");
        outtextxy(140,300,"NUMERO (S/N)");
        setcolor(13);
        outtextxy(140,240,"INSERTA NUMERO: ");
        gotoxy(24,17);
        strcpy(string,"");
        strcpy(string2,"");
        strcpy(caracter,"");
    }
    if (ORJEN==2){
        gotoxy(LOC_X,LOC_Y).printf(" ");
        strcpy(string,"");
        strcpy(string2,"");
        strcpy(caracter,"");
    }
    for (posicion=0,posicion<=8,posicion++){
        tecla:
        letra=toupper(getch());
        switch(letra){
            case 0x8: gotoxy(LOC_X,LOC_Y);
                    printf(" ");
                    strcpy(help,"");
                    strcat(help,string2,posicion-1);
                    strcpy(string2,help);
                    gotoxy(LOC_X,LOC_Y);
                    printf("%s",string2);
                    posicion--;
                    if (posicion<=0)posicion=0;
                    goto tecla;
            case 'A': strcat(caracter,"A");
                    break;
            case 'B': strcat(caracter,"B");
                    break;
            case 'C': strcat(caracter,"C");
                    break;
            case 'D': strcat(caracter,"D");
                    break;
            case 'E': strcat(caracter,"E");
                    break;
            case 'F': strcat(caracter,"F");
                    break;
            case 'G': strcat(caracter,"G");
                    break;
            case 'H': strcat(caracter,"H");
                    break;
            case 'I': strcat(caracter,"I");
                    break;
            case 'J': strcat(caracter,"J");
                    break;
            case 'K': strcat(caracter,"K");
                    break;
            case 'L': strcat(caracter,"L");
                    break;
            case 'M': strcat(caracter,"M");
                    break;
            case 'N': strcat(caracter,"N");
                    break;
            case 'O': strcat(caracter,"O");
                    break;
            case 'P': strcat(caracter,"P");
                    break;
            case 'Q': strcat(caracter,"Q");
                    break;
            case 'R': strcat(caracter,"R");
                    break;
            case 'S': strcat(caracter,"S");
                    break;
            case 'T': strcat(caracter,"T");
                    break;
            case 'U': strcat(caracter,"U");
                    break;
            case 'V': strcat(caracter,"V");
                    break;
            case 'W': strcat(caracter,"W");
                    break;
            case 'X': strcat(caracter,"X");
                    break;
            case 'Y': strcat(caracter,"Y");
                    break;
            case 'Z': strcat(caracter,"Z");
                    break;
            case '0': strcat(caracter,"0");
                    break;
            case '1': strcat(caracter,"1");
                    break;
            case '2': strcat(caracter,"2");
                    break;
            case '3': strcat(caracter,"3");
                    break;
            case '4': strcat(caracter,"4");
                    break;
            case '5': strcat(caracter,"5");
                    break;

```

```

case '6':strcat(caracter,"6");
    break;
case '7':strcat(caracter,"7");
    break;
case '8':strcat(caracter,"8");
    break;
case '9':strcat(caracter,"9");
    break;
case 0x1b: goto INTEGRADO;
case 0x00: getch();BELL();goto tecla;
case 0xd: goto end;
default: BELL();goto tecla;
if(strlen(string2)<8){strcat(string2,caracter);}
    else {BELL();posicion--;}
strcpy(caracter,"");
gotoxy(LOC_X,LOC_Y);
printf("%s",string2);
}
end;

if (ORIGEN==1){
    strcat(string,string2.8);
    outtextxy(140,280,"ESTA SEGURO DEL");
    outtextxy(140,300,"NUMERO ($N)");
    s=toupper(getche());
    if (s=="S"||s=="0xd) {
        TRSFORMA(string,PUERTO_A,PUERTO_B,PUER
TO_C,0,0);
    }
    else goto INTEGRADO;
}
if (ORIGEN==2){
    strcat(string,string2.8);
    TRSFORMA(string,0,0,0,1,1);
}
return;

"WINDOW1.C"
// FUNCION PARA DIBUJAR LA PANTALLA
PRINCIPAL
void WINDOW1(x1,y1,x2,y2)
{
    int punto[10];
    // MARCO DE LA NUEVA VENTANA
    setfillstyle(1,2);
    punto[0]= x1 ; punto[1]= y1 ;
    punto[2]=x1+x2; punto[3]= y1 ;
    punto[4]=x1+x2; punto[5]=y1+y2;
    punto[6]= x1 ; punto[7]=y1+y2;
    punto[8]= x1 ; punto[9]= y1 ;
    drawpoly(4,punto);
    fillpoly(4,punto);
    // FONDO DE LA NUEVA VENTANA
    setcolor(15);
    setfillstyle(1,9);

```

```

punto[0]=x1+3 ; punto[1]= y1+3 ;
punto[2]=x1+x2-3; punto[3]= y1+3 ;
punto[4]=x1+x2-3; punto[5]= y1+y2-3;
punto[6]=x1+3 ; punto[7]= y1+y2-3;
punto[8]=x1+3 ; punto[9]= y1+3 ;
drawpoly(4,punto);
fillpoly(4,punto);
}

"CLEARWIN.C"
//LIMPIAR PINES
void CLEARPIN(PUERTO_A,PUERTO_C)
{
    int C=0,x=0;
    outportb (PUERTO_C,0x0f);
    outportb (PUERTO_A,0xaa);
    while(x<10){
        while(C<=16)
            { outportb (PUERTO_C,239+C);
              C++;
            }
        x++;
    }
}

"SALIR.C"
//SALIR
SALIR(PUERTO_A,PUERTO_C)
{
    char S;
    settextstyle(TRIPLEX_FONT, HORIZ_DIR,2);
    outtextxy(157,150,"DESEA SALIR DEL
PROGRAMA");
    outtextxy(250,175,"( Si / No )");
    selecciona:
    S=toupper(getche());
    switch(S){
        case 0Xd :
            case 'S' :
                settextstyle(TRIPLEX_FONT,
HORIZ_DIR,1);
                setcolor(LIGHTGREEN);
                outtextxy(176,210,"PRESIONA LA
TECLA 'ENTER'");
                outtextxy(167,235,"PARA
DESACTIVAR EL SISTEMA");
                outtextxy(195,260,"O 'ESC' PARA
CANCELAR.");
                salida2:
                S=toupper(getche());
                if (S==0X1b)return 1;
                if (S==0Xd){
                    CLEARPIN(PUERTO_A,PUERTO_C);
                    outportb (PUERTO_C,0);
                    cleardevice();
                }
            }
    }
}

```

```

        closegraph();
        exit (1);
    } else (BELL();goto salida2;)
}
case 'N':return 1;
default: BELL();
        goto selecciona;
}
}

"DECOD_IN.C"
DECOD_IN (IN,C,D,MOD)
{
    int dato;
    int pin1,pin2,pin3,pin4,pin5,pin6,pin7,pin8;
    dato=IN;
    pin8=dato/128;
    if (dato>=128)dato=dato-128;
    pin7=dato/64;
    if (dato>=64)dato=dato-64;
    pin6=dato/32;
    if (dato>=32)dato=dato-32;
    pin5=dato/16;
    if (dato>=16)dato=dato-16;
    pin4=dato/8;
    if (dato>=8)dato=dato-8;
    pin3=dato/4;
    if (dato>=4)dato=dato-4;
    pin2=dato/2;
    if (dato>=2)dato=dato-2;
    pin1=dato/1;
    gotoxy(D-9,C+MOD);printf("%d log",pin1);
    gotoxy(D-9,C+1+MOD);printf("%d log",pin2);
    gotoxy(D-9,C+2+MOD);printf("%d log",pin3);
    gotoxy(D-9,C+3+MOD);printf("%d log",pin4);
    gotoxy(D-9,C+4+MOD);printf("%d log",pin5);
    gotoxy(D-9,C+5+MOD);printf("%d log",pin6);
    if(MOD==0){
        gotoxy(D-9,C+6);printf("%d log",pin7);
        gotoxy(D-9,C+7);printf("%d log",pin8);};
    return;
}

"OUTPIN_D.C"
int
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQUE,PUERTO_A,PUERTO_C)
{
    int DATO;
    if(PATA1>3){PATA1<0}(OUTERROR(),return;);
    if(PATA2>3){PATA2<0}(OUTERROR(),return;);
    if(PATA3>3){PATA3<0}(OUTERROR(),return;);
    if(PATA4>3){PATA4<0}(OUTERROR(),return;);
//PROGRAMACION DE LA PATA 1
    PATA1=cod(PATA1);

```

```

//PROGRAMACION DE LA PATA 2
    PATA2=cod(PATA2);
    PATA2=PATA2*4;
//PROGRAMACION DE LA PATA 3
    PATA3=cod(PATA3);
    PATA3=PATA3*16;
//PROGRAMACION DE LA PATA 4
    PATA4=cod(PATA4);
    PATA4=PATA4*64;
//RECOPIACION DE DATOS PARA LA SALIDA
    DATO=PATA1+PATA2+PATA3+PATA4;
    switch(BLOQUE){
        case 1:
            MOD0_0(DATO,0Xf0,PUERTO_A,PUERTO_C);
            break;
        case 2:
            MOD0_0(DATO,0Xf1,PUERTO_A,PUERTO_C);
            break;
        case 3:
            MOD0_0(DATO,0Xf2,PUERTO_A,PUERTO_C);
            break;
        case 4:
            MOD0_0(DATO,0Xf3,PUERTO_A,PUERTO_C);
            break;
        case 5:
            MOD0_0(DATO,0Xf4,PUERTO_A,PUERTO_C);
            break;
        case 6:
            MOD0_0(DATO,0Xf5,PUERTO_A,PUERTO_C);
            break;
        case 7:
            MOD0_0(DATO,0Xf6,PUERTO_A,PUERTO_C);
            break;
        case 8:
            MOD0_0(DATO,0Xf7,PUERTO_A,PUERTO_C);
            break;
        case 9:
            MOD0_0(DATO,0Xf8,PUERTO_A,PUERTO_C);
            break;
        case 10:
            MOD0_0(DATO,0Xf9,PUERTO_A,PUERTO_C);
            break;
        case 11:
            MOD0_0(DATO,0Xfa,PUERTO_A,PUERTO_C);
            break;
        case 12:
            MOD0_0(DATO,0Xfb,PUERTO_A,PUERTO_C);
            break;
    }
    return;
}

"MOD0_0.C"

```



```

//MODO 0
int MODO_0
(OUTA,OUTC,PUERTO_A,PUERTO_C)
{
    outputb(PUERTO_C,OUTC);
    outputb(PUERTO_A,OUTA);
    return 0;
}

"MODO_5_C"
//MODO 5
int MODO_5 (OUTC,PUERTO_B,PUERTO_C)
{
    int INB;
    outputb(PUERTO_C,OUTC);
    INB=inportb(PUERTO_B);
    return (INB);
}

"LOAD_INT_C"
int LOAD_INT
(N_INT,checa,string,PUERTO_A,PUERTO_B,PUE
RTO_C)
{
    cleardevice();
    switch (N_INT) {
        case 7400:
        case 7437:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND");
            NAND2INF();
            break;
        case 7401:
        case 7403:
        case 7426:
        case 7438:
        case 7439:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND");
            gotoxy(7,4);printf("COLECTOR
ABIERTO");
            NAND2INF();
            break;
        case 7410:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND");
            NAND3INF();
            break;
        case 7412:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND");
            NAND3INF();
            break;
        case 7420:
            NAND8INF();
            break;
        case 74134:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND 12");
            NAND12IF();
            break;
        case 74133:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND 13");
            NAND12IF();
            break;
        case 7402:
        case 7428:
            NOR1();
            gotoxy(7,3);printf("COMPUERTA
NOR");
            NOR2INF();
            break;
        case 7427:
            NOR1();
            gotoxy(7,3);printf("COMPUERTA
NOR");
            NOR3INF();
            break;
        case 7433:
        case 74128:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND");
            NAND3INF();
            break;
        case 7420:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND");
            NAND4INF();
            break;
        case 7422:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND");
            gotoxy(7,4);printf("COLECTOR
ABIERTO");
            NAND4INF();
            break;
        case 7430:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND");
            NAND8INF();
            break;
        case 74134:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND 12");
            NAND12IF();
            break;
        case 74133:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND 13");
            NAND12IF();
            break;
        case 7402:
        case 7428:
            NOR1();
            gotoxy(7,3);printf("COMPUERTA
NOR");
            NOR2INF();
            break;
        case 7427:
            NOR1();
            gotoxy(7,3);printf("COMPUERTA
NOR");
            NOR3INF();
            break;
        case 7433:
        case 74128:
            NAND1();
            gotoxy(7,3);printf("COMPUERTA
NAND");
            NAND3INF();
            break;
    }
}

```

```

case 74260:
    NOR1();
    gotoxy(7,3).printf("COMPUERTA
NOR");
    NOR4INF();
    break;
case 7404:
case 7405:
case 7406:
case 7407:
case 7416:
case 74901:
case 74902:
case 74903:
case 74904:
    NOT1();
    gotoxy(7,3).printf("COMPUERTA
NOT");
    NOT_INF();
    break;
case 7408:
case 7409:
    AND1();
    gotoxy(7,3).printf("COMPUERTA
AND");
    AND2INF();
    break;
case 7411:
    AND1();
    gotoxy(7,3).printf("COMPUERTA
AND");
    AND3INF();
    break;
case 7415:
    AND1();
    gotoxy(7,3).printf("COMPUERTA
AND");
    gotoxy(7,15).printf("COMPUERTA
CON");
    gotoxy(7,16).printf("SCHMITT
TRIGGER");
    break;
case 7421:
    AND1();
    gotoxy(7,3).printf("COMPUERTA
AND");
    AND4INF();
    break;
case 7441:
case 7446:
case 7447:
case 7448:
case 7449:
case 74248:

```

```

case 74249:
    CIR_INT0();
    gotoxy(7,2).printf("DECODIFICADOR");
    gotoxy(7,3).printf("BCD A 7
SEGMENTOS");
    break;
case 7442:
case 7445:
case 74141:
    CIR_INT0();
    gotoxy(7,2).printf("DECODIFICADOR");
    gotoxy(7,3).printf("BCD A DECIMAL");
    break;
case 7444:
    CIR_INT0();
    gotoxy(7,2).printf("DECODIFICADOR");
    gotoxy(7,3).printf("GRAY A
DECIMAL");
    break;
case 74142:
case 74143:
case 74144:
    CIR_INT0();
    gotoxy(9,2).printf("DECODIFICADOR");
    gotoxy(9,3).printf("BCD A DECIMAL");
    gotoxy(9,4).printf("Y 7 SEGMENTOS");
    break;
case 74147:
    CIR_INT0();
    gotoxy(7,2).printf("DECODIFICADOR");
    gotoxy(7,3).printf("DECIMAL A BCD");
    break;
case 74148:
case 74348:
    CIR_INT0();
    gotoxy(7,2).printf("DECODIFICADOR");
    gotoxy(7,3).printf("8 LINEAS A 3");
    gotoxy(7,4).printf("LINEAS OCTAL");
    break;
case 8368:
case 8370:
case 8374:
    CIR_INT0();
    gotoxy(7,2).printf("DECODIFICADOR");
    gotoxy(7,3).printf("HEXADECIMAL
A");
    gotoxy(7,4).printf("7 SEGMENTOS");

```

```

        break;
case 7474:
    FLIPFLOP);
gotoxy(7,3);printf("DUAL FLIP-FLOP
'D')");
        break;
case 7476:
    FLIPFLOP);
gotoxy(7,3);printf("DUAL FLIP-FLOP 'J-
K'");
        break;
case 74393:
    CIR_INT());
gotoxy(7,3);printf("DUAL CONTADOR
DE 4 BITS");
        break;
case 741:
    AMP_OP());
gotoxy(4,3);printf("AMPLIFICADOR
OPERACIONAL");
        break;
case 747:
    AMP_OP());
gotoxy(4,3);printf("DUAL
AMPLIFICADOR");
gotoxy(4,4);printf("OPERACIONAL");
        break;
case 339:
case 324:
    AMP_OP());
gotoxy(4,3);printf("CUADRUPLE
AMPLIFICADOR");
gotoxy(4,4);printf("OPERACIONAL");
        break;
default: CIR_INT());
}
rectangle(1,1,639,479);
rectangle(4,4,635,475);
line(245,4,245,475);
line(248,4,248,475);
line(248,280,634,280);
line(248,283,634,283);
if
(chaca==0)CHECMENU(string,PUERTO_A,PUERT
O_B,PUERTO_C);
return;
}
"REV_LIST.C"
//REVISION DE LA LISTA DE CIRCUITOS
INTEGRADOS DISPONIBLES
void REV_LIST (void)
{

```

```

char ACTUAL[MAXDIR];
int pantalla,tipo=0;
int x1,y1,x2,y2;
x1=10,y1=80,x2=200,y2=350;
WINDOW W1(x1,y1,x2,y2);
menu_sel;
outtextxy(40,120,"TIPO DE INTEGRADO");
outtextxy(20,155,"1 TTL (Transistor-");
outtextxy(40,170," Transistor-Logic");
outtextxy(20,195,"2 CMOS (Oxide Metal");
outtextxy(20,210," Semiconductor");
outtextxy(20,235,"3 VER TODA LA LISTA");
outtextxy(20,250," COMPLETA DE INT");
outtextxy(20,350," ELIJA UNA OPCION");
outtextxy(20,400," ESC PARA SALIR");
tipo=getch();
switch(tipo){
case '1'://LISTA TTL
    TTL(x1,y1,x2,y2);
    break;
case '2'://LISTA CMOS
    CMOS_LST(x1,y1,x2,y2);
    break;
case '3'://LISTA COMPLETA DE INTEGRADOS
    cleardevice();
    getcurdir(0,ACTUAL);
    chdir("LIBRERIA");
    gotoxy(1,1);
    system("DIR *.OUT/OW/P");
    getch();
    chdir("\");
    chdir(ACTUAL);
    return;
case 0x1b: return;
default : BELL();
    goto menu_sel;
}
}
"TTL.C"
//REVISION DE LA LISTA DE TTL
TTL(x1,y1,x2,y2)
{
int pag=0,ch;
select:
switch (pag){
case 0: CLEARWIN(x1,y1,x2,y2);
    outtextxy(40,130," 7400 74S03");
    outtextxy(40,150," 74C00 74C04");
    outtextxy(40,170," 74H00 74H04");
    outtextxy(40,190,"74LS00 74LS04");
    outtextxy(40,210," 74S00 74LS04");
    outtextxy(40,230," 7401 74S04");
    outtextxy(40,250,"74LS01 7405");
}

```

```

outtextxy(40,270," 74H01 74H05*);
outtextxy(40,290," 7402 74LS05*");
outtextxy(40,310," 74C02 74S05*");
outtextxy(40,330,"74LS02 7406*");
outtextxy(40,350," 74S02 7407*");
outtextxy(40,370," 7403 7408*);
outtextxy(40,390,"74LS03 74C08*);
break;
case 1: CLEARWIN(x1,y1,x2,y2);
outtextxy(40,130," 74H08 74LS11*");
outtextxy(40,150,"74LS08 74LS11*");
outtextxy(40,170," 74S08 7412*");
outtextxy(40,190," 7409 74LS12*");
outtextxy(40,210,"74LS09 7413*");
outtextxy(40,230," 74S09 74LS13*");
outtextxy(40,250," 7410 7414*");
outtextxy(40,270," 74C10 74C14*");
outtextxy(40,290," 74H10 74LS14*");
outtextxy(40,310,"74LS10 74LS15*");
outtextxy(40,330,"74LS10 74S15*");
outtextxy(40,350," 74S10 7416*");
outtextxy(40,370," 7411 7417*");
outtextxy(40,390," 74H11 7420*);
break;
case 2: CLEARWIN(x1,y1,x2,y2);
outtextxy(40,130," 74C20 74LS26*");
outtextxy(40,150," 74H20 7427*");
outtextxy(40,170,"74LS20 74LS27*");
outtextxy(40,190," 74S20 7428*");
outtextxy(40,210," 7421 74LS28*");
outtextxy(40,230," 74H21 7430*");
outtextxy(40,250,"74LS21 74C30*");
outtextxy(40,270," 7422 7432*");
outtextxy(40,290," 74H22 74LS30*");
outtextxy(40,310,"74LS22 74S30*");
outtextxy(40,330," 74S22 7432*");
outtextxy(40,350," 7423 74C32*");
outtextxy(40,370," 7425 74LS32*");
outtextxy(40,390," 7426 7433*");
break;
case 3: CLEARWIN(x1,y1,x2,y2);
outtextxy(40,130,"74LS33 7443*");
outtextxy(40,150," 7437 7444*");
outtextxy(40,170,"74LS37 7445*");
outtextxy(40,190," 7438 7446*");
outtextxy(40,210," 74LS38 7447*");
outtextxy(40,230," 7439 74LS47*");
outtextxy(40,250," 7440 7448*");
outtextxy(40,270," 74H40 74C48*");
outtextxy(40,290,"74LS40 74LS48*");
outtextxy(40,310," 74S40 74LS49*");
outtextxy(40,330," 7441 7450*");
outtextxy(40,350," 7442 74H50*");
outtextxy(40,370," 74C42 74H51*);

outtextxy(40,390,"74LS42 7451*");
break;
}
outtextxy(25,100,"PRESIONA LAS FLECHAS");
outtextxy(20,410," ESC PARA SALIR");
ch=getch();
if (ch==0x1b)return;
if (ch==0)ch=getch();
switch(ch){
case 0x4b: pag--;
break;
case 0x4d:
break;
case 0x50: pag++;
break;
}
if (pag>10)pag=10;
if (pag<0)pag=0;
goto select;
}

"CMOS_LST.C"
//REVISION DE LA LISTA DE CMOS
CMOS_LST(x1,y1,x2,y2)
{
int pag=0,ch;
select:
switch (pag){
case 0: CLEARWIN(x1,y1,x2,y2);
outtextxy(40,130,"4000 *4019*");
outtextxy(40,150,"4001 *4020*");
outtextxy(40,170,"4002 *4021*");
outtextxy(40,190,"*4006 *4022*");
outtextxy(40,210,"*4007 *4023*");
outtextxy(40,230,"*4008 *4024*");
outtextxy(40,250,"4011 *4025*");
outtextxy(40,270,"4012 *4026*");
outtextxy(40,290,"*4013 *4027*");
outtextxy(40,310,"*4014 *4028*");
outtextxy(40,330,"*4015 *4029*");
outtextxy(40,350,"*4016 *4030*");
outtextxy(40,370,"*4017 *4031*");
outtextxy(40,390,"*4018 *4032*");
break;
case 1: CLEARWIN(x1,y1,x2,y2);
outtextxy(40,130,"*4033 4050*");
outtextxy(40,150,"*4034 *4051*");
outtextxy(40,170,"*4035 *4052*");
outtextxy(40,190,"*4038 *4053*");
outtextxy(40,210,"*4040 *4055*");
outtextxy(40,230,"*4041 *4056*");
outtextxy(40,250,"*4042 *4060*");
outtextxy(40,270,"*4043 *4063*");
}

```

```

        outtextxy(40,290,"*4044 *4066");
        outtextxy(40,310,"*4045 *4067");
        outtextxy(40,330,"*4046 4068");
        outtextxy(40,350,"*4047 4069");
        outtextxy(40,370,"*4048 4070");
        outtextxy(40,390,"*4049 4071");
        break;
case 2: CLEARWIN(x1,y1,x2,y2);
        outtextxy(40,130,"4072");
        outtextxy(40,150,"4073");
        outtextxy(40,170,"4075");
        outtextxy(40,190,"*4076");
        outtextxy(40,210,"4077");
        outtextxy(40,230,"*4078");
        outtextxy(40,250,"4081");
        outtextxy(40,270,"*4082");
        outtextxy(40,290,"*4085");
        outtextxy(40,310,"*4086");
        outtextxy(40,330,"*4089");
        outtextxy(40,350,"*4093");
        outtextxy(40,370,"*NO SON
COMPUERTAS");
        outtextxy(40,390,"NO HAY MAS
DISPONIB.");
        break;
case 3: CLEARWIN(x1,y1,x2,y2);
        break;
        outtextxy(25,100,"PRESIONA LAS FLECHAS");
        outtextxy(20,410," ESC PARA SALIR");
        ch=getch();
        if(ch==0x1b)return;
        if(ch==0)ch=getch();
        switch(ch){
        case 0x4b: pag--;
        case 0x48: break;
        case 0x4d:
        case 0x50: pag++;
        case 0x54: break;
        }
        if(pag>3)pag=3;
        if(pag<0)pag=0;
        goto select;
}

"AND2INF.C"
//TABLA DE VERDAD DE LA COMPUERTA
AND DE DOS ENTRADAS
int AND2INF(void)
{
    settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
    outtextxy(70,230,"TABLA DE VERDAD");
    outtextxy(70,245," A B F");
}

        outtextxy(70,260," 0 0 0");
        outtextxy(70,275," 0 1 0");
        outtextxy(70,290," 1 0 0");
        outtextxy(70,305," 1 1 1");
        outtextxy(90,360,"NUMERO DE");
        outtextxy(90,375,"COMPUERTAS");
        outtextxy(90,390," 4");
        return ;
}

"NOT_INF.C"
//TABLA DE VERDAD DE LA COMPUERTA
AND DE DOS ENTRADAS
int NOT_INF(void)
{
    settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
    outtextxy(50,230,"TABLA DE VERDAD");
    outtextxy(50,245," A F");
    outtextxy(50,260," 0 1");
    outtextxy(50,275," 1 0");
    outtextxy(70,360,"NUMERO DE");
    outtextxy(70,375,"COMPUERTAS");
    outtextxy(70,390," 6");
    return ;
}

"NAND2INF.C"
//TABLA DE VERDAD DE LA COMPUERTA
NAND DE DOS ENTRADAS
int NAND2INF(void)
{
    settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
    outtextxy(70,230,"TABLA DE VERDAD");
    outtextxy(70,245," A B F");
    outtextxy(70,260," 0 0 1");
    outtextxy(70,275," 0 1 1");
    outtextxy(70,290," 1 0 1");
    outtextxy(70,305," 1 1 0");
    outtextxy(90,360,"NUMERO DE");
    outtextxy(90,375,"COMPUERTAS");
    outtextxy(90,390," 4");
    return ;
}

"NOR2INF.C"
//TABLA DE VERDAD DE LA COMPUERTA
NOR DE DOS ENTRADAS
int NOR2INF(void)
{
    settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
    outtextxy(70,230,"TABLA DE VERDAD");
    outtextxy(70,245," A B F");
    outtextxy(70,260," 0 0 1");
    outtextxy(70,275," 0 1 0");
}

```

```

outtextxy(70,290," 1 0 0");
outtextxy(70,305," 1 0");
outtextxy(90,360,"NUMERO DE");
outtextxy(90,375,"COMPUERTAS");
outtextxy(90,390," 4");
return ;
}

"CHECMENU.C"
//MENU DE VISAR INTEGRADO
CHECMENU(string,PUERTO_A,PUERTO_B,PUER
TO_C)
{
int opt;
setcolor(GREEN);
settextstyle(TRIPLEX_FONT, HORIZ_DIR,3);
outtextxy(320,310,"MENU DE OPCIONES");
outtextxy(325,350,"niciar revision");
outtextxy(325,380," egresar a la pantalla");
setcolor(LIGHTRED);
outtextxy(325,350,"I");
outtextxy(325,380,"R");
settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
setcolor(WHITE);
cmenu:
opt=coupper(getch());
switch(opt){
case
"R":return;
default :BELL; break;
}
goto cmenu;
}

"PIN_IN.C"
//DA EL VALOR LOGICO DEL PIN REQUERIDO
//int PIN_IN(NUM_PIN)
PIN_IN(NUM_PIN,TOTALPIN,PUERTO_B,PUER
TO_C)
{
int VALOR;
int
PIN1=0,PIN2=0,PIN3=0,PIN4=0,PIN5=0,PIN6=0,P
IN7=0,PIN8=0;
int PUERTO=LADOPIN,NUM_PIN2;
LADOPIN=TOTALPIN/2;

if (NUM_PIN<=LADOPIN)PUERTO=0;
else PUERTO=1;
if (PUERTO==0){
if (NUM_PIN==1&&NUM_PIN<=8)
VALOR=MODO_5(0x8f,PUERTO_B,PUERTO_C);
if (NUM_PIN==9&&NUM_PIN<=16)
VALOR=MODO_5(0x9f,PUERTO_B,PUERTO_C);
if (NUM_PIN==17&&NUM_PIN<=24)
VALOR=MODO_5(0xaf,PUERTO_B,PUERTO_C);
if (NUM_PIN==25&&NUM_PIN<=32)
VALOR=MODO_5(0xbf,PUERTO_B,PUERTO_C);
}
if (PUERTO==1){
NUM_PIN2=TOTALPIN-NUM_PIN+1;
if (NUM_PIN2==1&&NUM_PIN2<=8)
VALOR=MODO_5(0xaf,PUERTO_B,PUERTO_C);
if (NUM_PIN2==9&&NUM_PIN2<=16)
VALOR=MODO_5(0x9f,PUERTO_B,PUERTO_C);
if (NUM_PIN2==17&&NUM_PIN2<=24)
VALOR=MODO_5(0x8f,PUERTO_B,PUERTO_C);
if (NUM_PIN2==25&&NUM_PIN2<=32)
VALOR=MODO_5(0xbf,PUERTO_B,PUERTO_C);
}
if (PUERTO==0){
while(NUM_PIN>8)NUM_PIN=NUM_PIN-8;}
if (PUERTO==1){
while(NUM_PIN2>8)NUM_PIN2=NUM_PIN2-
8;}
//CALCULA LOS VALORES DE LOS PINES
if (VALOR==128) (PIN8=1;VALOR=VALOR-
128;

if (VALOR==64) (PIN7=1;VALOR=VALOR-64;
if (VALOR==32) (PIN6=1;VALOR=VALOR-32;
if (VALOR==16) (PIN5=1;VALOR=VALOR-16;
if (VALOR==8) (PIN4=1;VALOR=VALOR-8;
if (VALOR==4) (PIN3=1;VALOR=VALOR-4;
if (VALOR==2) (PIN2=1;VALOR=VALOR-2;
if (VALOR==1) (PIN1=1;
//REGRESA EL VALOR DEL PIN REQUERIDO
if (PUERTO==1)NUM_PIN=NUM_PIN2;
switch(NUM_PIN){
case 1: return PIN1;
case 2: return PIN2;
case 3: return PIN3;
case 4: return PIN4;
case 5: return PIN5;
case 6: return PIN6;
case 7: return PIN7;
case 8: return PIN8;
}
return 0;
}

```

```

"RESUL_1.C"
//RESULTADOS DE LAS COMPUERTAS
RESUL_1(OK1,OK2,OK3,OK4)
{
  if(OK1==0)outtextxy(440,60,"OK");
  else {outtextxy(480,60,"MAL 1");}
  if(OK2==0)outtextxy(440,90,"OK");
  else {outtextxy(480,90,"MAL 1");}
  if(OK3==0)outtextxy(440,120,"OK");
  else {outtextxy(480,120,"MAL 1");}
  if(OK4==0)outtextxy(440,150,"OK");
  else {outtextxy(480,150,"MAL 1");}

  outtextxy(330,30,"ESTADO DE LAS
  COMPUERTAS");
  outtextxy(300,60,"COMPUERTA 1");
  outtextxy(300,90,"COMPUERTA 2");
  outtextxy(300,120,"COMPUERTA 3");
  outtextxy(300,150,"COMPUERTA 4");
  setcolor(LIGHTRED);
  settxtstyle(TRIPLEX_FONT,HORIZ_DIR,3);
  outtextxy(300,410,"COMPROBACION
  FINALIZADA");
  settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
  return;
}

"COD.C"
int cod(DATO)
{
  if (DATO==0){DATO=2;return DATO;};
  if (DATO==1){DATO=0;return DATO;};
  if (DATO==2){DATO=0;return DATO;};
  if (DATO==3){DATO=1;return DATO;};
  return 0;
}

"OUTERROR.C"
OUTERROR(void)
{cleardevice();
setcolor(15);
rectangle(1,1,639,479);
rectangle(4,4,635,475);
settxtstyle(TRIPLEX_FONT,HORIZ_DIR,6);
outtextxy(110,50,"ERROR DE DATOS");
outtextxy(110,100," INSERTADOS");
settxtstyle(SANS_SERIF_FONT,HORIZ_DIR,3);
outtextxy(70,190,"SOLO DAR VALORES A LOS
PINES DE ACUERDO");
outtextxy(70,230,"A LA SIGUIENTE TABLA.");
setcolor(LIGHTRED);
outtextxy(70,270,"O - 0 LOGICO ( GND )");
outtextxy(70,310,"I - 1 LOGICO ( 5 VOLTS )");
}

```

```

outtextxy(70,350,"2 - PIN DE RESPUESTA DE
INT.");
outtextxy(70,390,"3 - ALIMENTACION ( 5 VOLTS
)");
settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
setcolor(15);
getch();
return;
}

"LEE_DATO.C"
long
LEE_DATO(NOMBRE,LEE_DIR,DATO,TIPO)
{
  FILE *in;
  char ext1[80]=".IN";
  char ext2[80]=".OUT";
  char NOMBRE2[80]="";
  long centenas,unidades,decenas;
  strncat(NOMBRE2,NOMBRE,9);
  if(TIPO==0)strncat(NOMBRE2,ext1,4)//
  ARCHIVOS EXTENSION .IN O DE ENTRADA
  if(TIPO==1)strncat(NOMBRE2,ext2,4)//
  ARCHIVOS EXTENSION .OUT O DE SALIDA
  if(((in = fopen(NOMBRE2, "r"))== NULL)
  return 1000;//CODIGO DE ARCHIVO
  INEXISTENTE O ERROR DE DISCO
  LEE_DIR_DATO=(LEE_DIR-DATO-1)*3;
  fseek(in,LEE_DIR_DATO,SEEK_CUR);
  centenas=(fgetc(in)-0x30)*100;
  decenas=(fgetc(in)-0x30)*10;
  unidades=fgetc(in)-0x30;
  fclose(in);
  return (unidades*decenas+centenas);
}

"INREADAT.C"
//ARCHIVO PARA LEER DATOS
ALMACENADOSZ EN LOS ARCHIVOS DE
ENTRADA Y SALIDA
INREADAT(string,PUERTO_A,PUERTO_B,PUER
TO,C)
{
  char ACTUAL[MAXDIR];
  char INTEGRADO[10];
  long DATOCOMP=0;
  int PRUEBA=0;
  int RESP=0,c;
  settxtstyle(TRIPLEX_FONT,HORIZ_DIR,1);
  strcpy(INTEGRADO,"");
  strncat(INTEGRADO,string,9);
  setcolor(YELLOW);
  outtextxy(410,8,INTEGRADO);
}

```

```

settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
getcurdir(0,ACTUAL);
chdir("LIBRERIA");
DATOCOMP=LEE_DATO(string,1,1);
setfillstyle(SOLID_FILL,LIGHTGREEN);
for (c=0; c<DATOCOMP+1; c++){
  if(c==1){
    outtextxy(280,150,"COMPUERTA 1");
    fillellipse(385,151,5,5);}
  if(c==2){
    outtextxy(280,165,"COMPUERTA 2");
    fillellipse(385,166,5,5);}
  if(c==3){
    outtextxy(280,180,"COMPUERTA 3");
    fillellipse(385,181,5,5);}
  if(c==4){
    outtextxy(280,195,"COMPUERTA 4");
    fillellipse(385,196,5,5);}
  if(c==5){
    outtextxy(280,210,"COMPUERTA 5");
    fillellipse(385,211,5,5);}
  if(c==6){
    outtextxy(280,225,"COMPUERTA 6");
    fillellipse(385,226,5,5);}
  if(c==7){
    outtextxy(420,150,"COMPUERTA 7");
    fillellipse(525,151,5,5);}
  if(c==8){
    outtextxy(420,165,"COMPUERTA 8");
    fillellipse(525,166,5,5);}
  if(c==9){
    outtextxy(420,180,"COMPUERTA 9");
    fillellipse(525,181,5,5);}
  if(c==10){
    outtextxy(420,195,"COMPUERTA 10");
    fillellipse(525,196,5,5);}
  if(c==11){
    outtextxy(420,210,"COMPUERTA 11");
    fillellipse(525,211,5,5);}
  if(c==12){
    outtextxy(420,225,"COMPUERTA 12");
    fillellipse(525,226,5,5.)}
}
outtextxy(300,245,"COMPUERTA EN BUEN
ESTADO");
outtextxy(300,260,"COMPUERTA
DEFECTUOSA");
setcolor(LIGHTGREEN);
fillellipse(280,245,5,5);
setfillstyle(SOLID_FILL,LIGHTRED);
fillellipse(280,260,5,5);
setcolor(YELLOW);
while(Resp==0){
  PRUEBA++;
  outtextxy(280,37,"NUMERO DE
COMBINACION");
  gotoxy(60,3);printf("%d",PRUEBA);
  Resp=INS_DATO(string,PRUEBA,PUERTO_A,PU
ERTO_C);
  if
  (Resp==0)COMPRUEBA(string,PRUEBA,PUERT
O_B,PUERTO_C);
  }
  chdir("V");
  chdir(ACTUAL);
  settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
  return 0;
}
"DET_BLK_C"
DETBLOQUE(pines)
{
  int BLOQUE;
  //DEC_NUM
  if ((pines/2)<=4)BLOQUE=1;
  if ((pines/2)>=5&&(pines/2)<=8)BLOQUE=2;
  if ((pines/2)>=9&&(pines/2)<=12)BLOQUE=3;
  if ((pines/2)>=13&&(pines/2)<=16)BLOQUE=4;
  if ((pines/2)>=17&&(pines/2)<=20)BLOQUE=5;
  if ((pines/2)>=21&&(pines/2)<=24)BLOQUE=6;
  if ((pines/2)>=25&&(pines/2)<=28)BLOQUE=7;
  if ((pines/2)>=29){gotoxy(20,10);printf("ERROR
DE PROGRAMACION");
  getch(); exit (1);}
  return BLOQUE;
}
"INS_DATO_C"
INS_DATO(string,PRUEBA,PUERTO_A,PUERTO_
C)
{
  int DIR;
  int BLOQUES,BLOQUE;
  long DATOIN=0;
  int
  PATA1,PATA2,PATA3,PATA4,PATA5,PATA6,PA
TA7,PATA8,PATA9,PATA10,PATA11;
  int PATA12,PATA13,PATA14,PATA15,PATA16;
  //DATOIN=LEE_DATO(NOMBRE_ARCHIVO,LE
E_DIR,DATO,TIPO '0-IN 1-OUT');
  DATOIN=LEE_DATO(string,1,0);
  if(DATOIN==1000){
    cleardevice();
    settextstyle(TRIPLEX_FONT, HORIZ_DIR,3);

```



```

outtextxy(165,150,"NO EXISTE EL
INTEGRADO");
outtextxy(110,200,"O NO SE PUEDE ABRIR EL
ARCHIVO");
settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
return 1;}
else {
outtextxy(280,53,"EL INTEGRADO TIENE
PINES");
gotoxy(57,4);printf("%d",DATOIN);}
BLOQUES=DETBLOQUE(DATOIN);
PRUEBA=PRUEBA-1;
DIR=1+(PRUEBA*8-BLOQUES);
BLOQUE=1;
outtextxy(280,67,"LOS DATOS POR BLOQUE
SON");
while(BLOQUE<=BLOQUES){
DIR++;
PATA1=LEE_DATO(string,DIR,0);
if(PATA1<=-1)return 1;
DIR++;
PATA2=LEE_DATO(string,DIR,0);
DIR++;
PATA3=LEE_DATO(string,DIR,0);
DIR++;
PATA4=LEE_DATO(string,DIR,0);
if(BLOQUE==1){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(36,7);printf("%d%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==2){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(41,7);printf("%d%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==3){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(46,7);printf("%d%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==4){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(51,7);printf("%d%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}

```

```

if(BLOQUE==5){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(56,7);printf("%d%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==6){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(61,7);printf("%d%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==7){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(66,7);printf("%d%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==8){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(71,7);printf("%d%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
BLOQUE++;
}
BLOQUE=9;
while(BLOQUE<=BLOQUES+8){
DIR++;
PATA1=LEE_DATO(string,DIR,0);
if(PATA1<=-1)return 1;
DIR++;
PATA2=LEE_DATO(string,DIR,0);
DIR++;
PATA3=LEE_DATO(string,DIR,0);
DIR++;
PATA4=LEE_DATO(string,DIR,0);
if(BLOQUE==1+8){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(36,6);printf("%d%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==2+8){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);

```

```

gotoxy(41,6);printf("%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==3+8){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(46,6);printf("%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==4+8){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(51,6);printf("%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==5+8){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(56,6);printf("%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==6+8){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(61,6);printf("%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==7+8){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(66,6);printf("%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
if(BLOQUE==8+8){
OUTPIN_D(PATA1,PATA2,PATA3,PATA4,BLOQ
UE,PUERTO_A,PUERTO_C);
gotoxy(71,6);printf("%d%d%d%d",PATA1,PATA2,
PATA3,PATA4);}
BLOQUE++;
}
//getch();
return 0;
}
"COMPRU2.C"
COMPRUEBA(string,PRUEBA,PUERTO_B,PUERT
O_C)

```

```

{
int TOTALPIN=0;
int dir=0,data,N_PRUEBA=0;
int datopin,RESP_COMP,RESP_OK,RESP_INT;
int G,gate,OKGATE;
int S,T,U;
while (!(PRUEBA==N_PRUEBA)){
dato=1000;
while (!(dato==777)){
dir++;
dato=LEE_DATO(string,dir,1);}
N_PRUEBA++;
}
NEXTPIN:
dir++;
dato=LEE_DATO(string,dir,1);
datopin=dato;
//DATOPIN ES EL NUMERO DEL PIN A
RESPONDER
if (datopin==777)return 0;
dir++;
RESP_OK=LEE_DATO(string,dir,1);
TOTALPIN=LEE_DATO(string,1,0);
//RESP_INT ES EL DATO QUE TIENE EL PIN DE
RESPUESTA.
RESP_INT=PIN_IN(datopin,TOTALPIN,PUERTO_
B,PUERTO_C);
outtextxy(280,116,"PIN QUE RESPONDE =");
gotoxy(56,8);printf(" ");
gotoxy(56,8);printf("%d",datopin);
outtextxy(280,132,"RESPUESTA REAL =");
gotoxy(54,9);printf(" ");
gotoxy(54,9);printf("%d",RESP_INT);
outtextxy(460,132,"TABLA <OK> =");
gotoxy(72,9);printf(" ");
gotoxy(72,9);printf("%d",RESP_OK);
if (!(RESP_OK==RESP_INT))
{G=0;
for(gate=1700;gate<3600;gate=gate+100){
G++;
//OKGATE RECIBE UN VALOR ENTRE 1700
Y 3600 PARA DECIR CUAL ES
//LA COMPUTERA QUE ESTA FALLANDO,
Y SI ESTA NO EXISTE,
//OKGATE ES IGUAL AL CERO.
OKGATE=GATE(string,gate,datopin);
if (OKGATE==0)gate=3700;
if (gate==OKGATE){
if(G==1)fillellipse(385,151,5,5);
if(G==2)fillellipse(385,166,5,5);
if(G==3)fillellipse(385,181,5,5);
if(G==4)fillellipse(385,196,5,5);
if(G==5)fillellipse(385,211,5,5);

```

```

if(G==6)filellipse(385,226,5,5);
if(G==7)filellipse(525,151,5,5);
if(G==8)filellipse(525,166,5,5);
if(G==9)filellipse(525,181,5,5);
if(G==10)filellipse(525,196,5,5);
if(G==11)filellipse(525,211,5,5);
if(G==12)filellipse(525,226,5,5);
/*ALARMA DE COMPUERTA
DEFECTUOSA*/
for (U=0;U<3; U++){
for (S=30;S<1000; S++){
for (T=0; T<1100; T++){
sound(S);}
nosound();
gate=3700;
}
}
}
goto NEXTPIN;
}
"GATE.C"
GATE(string,PARTE,dato,opin)
{
int dir2=0;
int dato=0,RESP_COMP;
//EN EL WHILE SE BUSCA SI EL NUMERO DE
PIN PERTENECE A UNA COMPUERTA
while ((dato==PARTE)){
dir2++;
dato=LEE_DATO(string,dir2,1);
if(dato==0)return 0;}
RESP_COMP=0;
//EN ESTA PARTE SE CHECA QUE EN LA
COMPUERTA EXISTA EL PIN QUE
//DEBE RESPONDER.
while(RESP_COMP<1000){
dir2++;
RESP_COMP=LEE_DATO(string,dir2,1);
if(RESP_COMP==dato)return PARTE;
if(RESP_COMP==0)return 0;}
return 1;
}

"TESTSYST.C"
//ARCHIVO PARA PROBAR EL
FUNCIONAMIENTO DE LOS
// PINES DE LA BASE DEL INTEGRADO DE
PRUEBAS.
TESTSYST(PUERTO_A,PUERTO_B,PUERTO_C)
{
int
C=0,TEST;//,PUERTO_A=0x240,PUERTO_B=0x24
1,PUERTO_C=0x242
int
X=161,Y=0,BLK=0,TESTPIN,FUN_SYS=0,SISTE
MA;
int x1=120,y1=85,x2=380,y2=305;
int CODE_CODEX=0;
GRAFICOS();
// 1010 0100 ceros (aa hex)
// 0000 0000 uno logico (00 hex)
// 0101 0101 alimentaciones (55 hex)
setbkcolor(BLUE);
// MARCO 1
rectangle(1,639,479);
// MARCO 2
rectangle(4,4,635,475);
settextstyle(TRIPLEX_FONT, HORIZ_DIR,4);
outtextxy(90,40,"RETIRE CUALQUIER
CIRCUITO");
outtextxy(40,70,"INTEGRADO QUE SE
ENCUENTRE EN");
outtextxy(50,100,"LA BASE Y PRESIONE
CUALQUIER");
outtextxy(110,130,"TECLA PARA INICIAR LA");
outtextxy(90,160,"AUTOPRUEBA DEL SISTEMA.
..");
settextstyle(TRIPLEX_FONT, HORIZ_DIR,1);
outtextxy(230,218,"PROBANDO PIN ");
settextstyle(DEFAULT_FONT,VERT_DIR,1);
outtextxy(171,405,"1");
outtextxy(241,405,"8");
outtextxy(321,405,"16");
outtextxy(171,250,"32");
outtextxy(241,250,"25");
outtextxy(321,250,"17");
rectangle(157,300,330,370);
rectangle(160,303,327,367);
bar(157,329,172,341);
getch();
CLEARPIN(PUERTO_A,PUERTO_C);
for (C=1, C<=32; C++){
if(C==5)BLK++;//getch();
if(C==9)BLK++;
if(C==13)BLK++;
if(C==17)BLK++;
if(C==21)BLK++;
if(C==25)BLK++;
if(C==29)BLK++;
if(C==33)BLK++;
if(C==37)BLK++;
if(C==41)BLK++;
if(C==45)BLK++;
if(C==49)BLK++;
if(C==53)BLK++;
if(C==57)BLK++;
if(C==61)BLK++;
}
}

```

```

TESTPIN=1;
gotoxy(49,13);printf("%d",C);
outportb (PUERTO_C,0X0D+BLK);
outportb (PUERTO_A,0Xaa);
TEST=PIN_IN(C,32,PUERTO_B,PUERTO_C);
if((TEST==0))TESTPIN=0;
outportb (PUERTO_C,0X0D+BLK);
outportb (PUERTO_A,0X00);
TEST=PIN_IN(C,32,PUERTO_B,PUERTO_C);
if((TEST==1))TESTPIN=0;
outportb (PUERTO_C,0X0D+BLK);
outportb (PUERTO_A,0Xaa);
outportb (PUERTO_C,0X0D+BLK);
outportb (PUERTO_A,0X55);
TEST=PIN_IN(C,32,PUERTO_B,PUERTO_C);
if((TEST==1))TESTPIN=0;
if (TESTPIN==0){
  if(C<=16)
  { setcolor(LIGHTRED);
    for(Y=1;Y<9;Y++)
    {line(C+Y+X,399,C+Y+X,371);}
    X=X+Y;
  }
  if(C>16)
  { setcolor(LIGHTRED);
    for(Y=1;Y<9;Y++)
    {line(C+Y+X-10,299,C+Y+X-10,271);}
    X=X-Y-2;
  }
}
else{
FUN_SYS++;
if(C<=16)
{ setcolor(LIGHTGREEN);
  for(Y=1;Y<9;Y++)
  {line(C+Y+X,399,C+Y+X,371);}
  X=X+Y;
}
if(C>16)
{ setcolor(LIGHTGREEN);
  for(Y=1;Y<9;Y++)
  {line(C+Y+X-10,299,C+Y+X-10,271);}
  X=X-Y-2;
}
}
}
SISTEMA=(FUN_SYS*100)/28;
setcolor(WHITE);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
gotoxy(74,21);printf("%d",SISTEMA);
outtextxy(390,325,"SISTEMA FUNCIONANDO AL
%");
outtextxy(400,450,"PIN FUNCIONANDO");
outtextxy(200,450,"PIN DEFECTUOSO");

```

```

if(SISTEMA==0){SISTEMA<=4}{
  delay(500);
  WINDOW1(x1,y1,x2,y2);
  settextstyle(TRIPLEX_FONT, HORIZ_DIR,4);
  setcolor(YELLOW);
  outtextxy(150,95,"EL SISTEMA NO ESTA");
  outtextxy(170,125," FUNCIONANDO !");
  settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
  setcolor(WHITE);
  outtextxy(150,180,"POSIBLES SOLUCIONES.");
  outtextxy(140,220,"- ASEGURE QUE EL
SISTEMA ESTE CONECTADO.");
  outtextxy(140,260,"- VERIFIQUE LA POSICION
DEL JUMPER EN LA");
  outtextxy(140,280,"- TARJETA
CORRESPONDA CON LA POSICION");
  outtextxy(140,300,"- DEFINIDA EN EL
PROGRAMA DE CONFIGURACION.");
  setcolor(LIGHTGREEN);
  outtextxy(160,340,"PRESIONE CUALQUIER
TECLA PARA SALIR");
  BELL();
  CODE=toupper(getch());
  if(CODE=='C'){
    CODEX=1;
    CODE=toupper(getch());}else(goto SALIDA);
  if(CODE=='U'){
    CODEX=CODEX+1;
    CODE=toupper(getch());}else(goto SALIDA);
  if(CODE=='S'){
    CODEX=CODEX+1;
    CODE=toupper(getch());}else(goto SALIDA);
  if(CODE=='T'){
    CODEX=CODEX+1;
    CODE=toupper(getch());}else(goto SALIDA);
  if(CODE=='A'){
    CODEX=CODEX+1;
    CODE=toupper(getch());}else(goto SALIDA);
  if(CODE=='V'){
    CODEX=CODEX+1;
    CODE=toupper(getch());}else(goto SALIDA);
  if(CODE=='O'){
    CODEX=CODEX+1;}else(goto SALIDA);
  if(CODEX==7)return 0;
  else{
    SALIDA:
    closegraph();
    return 1;}
}
else{
  setcolor(LIGHTGREEN);
  outtextxy(380,340,"VERIFICACION
FINALIZADA");
  setcolor(LIGHTRED);
  for(Y=1;Y<9;Y++){

```

```

line(180+Y,460,180+Y,435);
setcolor(LIGHTGREEN);
for(Y=1;Y<9;Y++)
line(380+Y,460,380+Y,435);
delay(1500);
closegraph();
return 0;
}

```

```

"AYUDA.C"
// AYUDA
// RUTINA PARA DAR LAS CARACTERISTICAS
DEL FUNCIONAMIENTO
// DEL SISTEMA Y EL MENU PRINCIPAL DEL
PROGRAMA.
void AYUDA(void)
{ int x1=120,y1=85,x2=380,y2=305;
  int PAGE;
  int E=0;
  MENU:
  WINDOW1(x1,y1,x2,y2);
  settextstyle(TRIPLEX_FONT, HORIZ_DIR,1);
  setcolor(YELLOW);
  outtextxy(x1+50,y1+20,"SISTEMA PARA
COMPROMISER EL.");
  outtextxy(x1+45,y1+40,"FUNCIONAMIENTO DE
CIRCUITOS");
  outtextxy(x1+75,y1+60,"INTEGRADOS POR
MEDIO");
  outtextxy(x1+80,y1+80,"DE UNA
COMPUTADORA.");
  setcolor(BLACK);
  outtextxy(x1+75,y1+120," . REVISAR LISTA.");
  outtextxy(x1+75,y1+140," . CHECAR
INTEGRADO.");
  outtextxy(x1+75,y1+160," . BUSCAR.");
  outtextxy(x1+75,y1+180," . AYUDA.");
  outtextxy(x1+75,y1+200," . SALIR.");
  outtextxy(x1+75,y1+220," . CREDITOS DEL
SISTEMA.");
  setcolor(LIGHTRED);
  outtextxy(x1+75,y1+120,"1");
  outtextxy(x1+75,y1+140,"2");
  outtextxy(x1+75,y1+160,"3");
  outtextxy(x1+75,y1+180,"4");
  outtextxy(x1+75,y1+200,"5");
  outtextxy(x1+75,y1+220,"6");
  settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
  setcolor(WHITE);
  outtextxy(x1+50,y1+270,"PRESIONE UNA
TECLA PARA SELECCIONAR.");
  outtextxy(x1+65,y1+285,"O 'ESC' PARA
REGRESAR AL MENU.");
  MENU:

```

```

E=getch();
switch(E){
case '1': WINRL1:
CLEARWIN(x1,y1,x2,y2);
settextstyle(TRIPLEX_FONT,
HORIZ_DIR,1);
setcolor(YELLOW);
outtextxy(x1+120,y1+20,"REVISAR
LISTA.");
settextstyle(DEFAULT_FONT,
HORIZ_DIR,1);
setcolor(WHITE);
outtextxy(x1+45,y1+50,"ESTA OPCION
SERVE PARA VER QUE CIRCUITOS");
outtextxy(x1+15,y1+65,"INTEGRADOS
PUEDEN SER CHECADOS POR EL.");
outtextxy(x1+15,y1+80,"SISTEMA.");
outtextxy(x1+45,y1+95,"EL MODO DE
USO ES SELECCIONANDO UN NUMERO");
outtextxy(x1+15,y1+110,"DE LAS
CUATRO OPCIONES QUE OFRECE EL MENU
DE.");
outtextxy(x1+15,y1+125,"REVISAR
LISTA.");
outtextxy(x1+15,y1+150,"1.- TTL (
INTEGRADOS DE LOGICA TRANSISTOR");
outtextxy(x1+15,y1+165,"
TRANSISTOR.");
outtextxy(x1+15,y1+180,"2.- CMOS (
INTEGRADOS DE OXIDO DE SILICIO);");
outtextxy(x1+15,y1+195,"3.- VER
TODA LA LISTA.");
outtextxy(x1+15,y1+210,"4.- SALIR.");
setcolor(LIGHTGREEN);
outtextxy(x1+30,y1+270,"PRESIONE
PAGE UP/DOWN PARA CAMBIAR DE");
outtextxy(x1+45,y1+285,"PAGINA O
'ESC' PARA REGRESAR AL MENU.");
  REACT:
  PAGE=getch();
  if(PAGE==OX||goto MENU;
  if(PAGE==0)goto REACT1;
  if(PAGE==O){
    PAGE=getch();
    if(PAGE==OX5||PAGE==OX50||PAGE==
OX4d)goto REACT1;}
  WINRL2:
  CLEARWIN(x1,y1,x2,y2);
  outtextxy(x1+45,y1+50,"LA OPCION 1
ACTIVA UNA VENTANA EN DONDE");
  outtextxy(x1+15,y1+65,"APARECEN
LOS NUMEROS DE CIRCUITOS LOGICOS");

```

```

outtextxy(x1+15,y1+80,"TTL QUE YA
ESTAN INCLUIDOS EN EL SISTEMA Y.");
outtextxy(x1+15,y1+95,"PARA
CAMBIAR DE PAGINA, SOLO SE TIENEN
QUE");
outtextxy(x1+15,y1+110,"PRESIONAR
LAS FLECHAS DEL TECLADO.");
outtextxy(x1+45,y1+140,"LA OPCION 2
MUESTRA UNA VENTANA DONDE SE");
outtextxy(x1+15,y1+155,"PUEDEN VER
LOS NUMEROS DE LOS CIRCUITOS DE");
outtextxy(x1+15,y1+170,"TIPO CMOS
QUE ESTAN INCLUIDOS EN LA LIBRERIA");
outtextxy(x1+15,y1+185,"DE
CIRCUITOS INTEGRADOS Y, PARA CAMBIAR
DE");
outtextxy(x1+15,y1+200,"PAGINA
SOLO SE PRESIONAN LAS FLECHAS DEL");
outtextxy(x1+15,y1+215,"TECLADO.");
setcolor(LIGHTGREEN);
outtextxy(x1+30,y1+270,"PRESIONE
PAGE UP/PG DOWN PARA CAMBIAR DE");
outtextxy(x1+45,y1+285,"PAGINA O
'ESC' PARA REGRESAR AL MENU.");
REACT2:
PAGE=getch();
if(PAGE==OX1b)goto MENU;
if!(PAGE==0)goto REACT2;
if(PAGE==0){
PAGE=getch();
if(PAGE==OX48||PAGE==OX49||PAGE==O
X4b)goto WINRL2;
goto REACT3;
case '2': WINCH1:
CLEARWIN(x1,y1,x2,y2);
settextstyle(TRIPLEX_FONT,
HORIZ_DIR,1);
setcolor(YELLOW);
outtextxy(x1+110,y1+20,"CHECAR
INTEGRADO.");
settextstyle(DEFAULT_FONT,
HORIZ_DIR,1);
setcolor(WHITE);
outtextxy(x1+45,y1+50,"ESTA
MUESTRA UN MENU EL CUAL NOS AYUDA
A");
outtextxy(x1+15,y1+65,"SELECCIONAR EL
MODO EN QUE VA PROBARSE EL");
outtextxy(x1+15,y1+80,"CIRCUITO
INTEGRADO CON EL SISTEMA.");
outtextxy(x1+45,y1+95,"LA FORMA DE
SELECCIONAR UNA OPCION ES");
outtextxy(x1+15,y1+110,"PRESIONANDO UN
NUMERO DE LOS CUATRO QUE SE");
outtextxy(x1+15,y1+125,"MUESTRAN
A CONTINUACION.");
outtextxy(x1+15,y1+150,"1.-
INSERTAR EL NUMERO DE INTEGRADO QUE
SE");
outtextxy(x1+15,y1+165,"VA A
REVISAR.");

```

```

outtextxy(x1+15,y1+180,"2-
INSERTAR DATOS AL INTEGRADO DE
PRUEBA");
outtextxy(x1+15,y1+195,"
MANUALMENTE.");
outtextxy(x1+15,y1+210,"3.-
EDICION.");
outtextxy(x1+15,y1+225,"4.- SALIR.");
setcolor(LIGHTGREEN);
outtextxy(x1+30,y1+270,"PRESIONE
PAGE UP/PG DOWN PARA CAMBIAR DE");
outtextxy(x1+45,y1+285,"PAGINA O
'ESC' PARA REGRESAR AL MENU.");
REACTCH1;
PAGE=getch();
if(PAGE==OX1b)goto MENU;
if((PAGE==0))goto REACTCH1;
if(PAGE==0){
PAGE=getch();
if((PAGE==OX51||PAGE==OX50||PAGE==
OX4d))goto REACTCH1;
WINCH2;
CLEARWIN(x1,y1,x2,y2);
outtextxy(x1+45,y1+30,"LA OPCION 1
SERVE PARA ACTIVAR LAS FUN.");
outtextxy(x1+15,y1+45,"CIONES QUE
HARAN QUE EL CIRCUITO INTEGRADO");
outtextxy(x1+15,y1+60,"QUE SE
ENCUENTRE EN LA BASE DEL SISTEMA
SEA");
outtextxy(x1+15,y1+75,"PROBADO SU
FUNCIONAMIENTO.");
outtextxy(x1+45,y1+90,"AL ELEGIR
ESTA FUNCION SE ACTIVARA UN");
outtextxy(x1+15,y1+105,"CUADRO EN
EL CUAL SE TIENE QUE COLOCAR EL");
outtextxy(x1+15,y1+120,"NUMERO
DEL CIRCUITO INTEGRADO QUE SE QUIERE");
outtextxy(x1+15,y1+135,"PROBAR SU
FUNCIONAMIENTO, Y AL ACEPTAR EL");
outtextxy(x1+15,y1+150,"NUMERO
APARECERA UNA PANTALLA EN LA
CUAL");
outtextxy(x1+15,y1+165,"APARECE
UN PEQUEÑO MENU EN EL CUAL INICIA");
outtextxy(x1+15,y1+180,"O CANCELA
LA COMPROBACION DEL INTEGRADO");
outtextxy(x1+15,y1+195,"QUE SE
ENCUENTRA EN LA BASE");
setcolor(LIGHTRED);
outtextxy(x1+45,y1+210,"PRECAUCION.");
setcolor(WHITE);
outtextxy(x1+45,y1+225,"ASEGURESE
QUE EL NUMERO DEL INTEGRADO");
outtextxy(x1+15,y1+240,"QUE SE
ENCUENTRA EN LA BASE DE PRUEBAS SEA");
outtextxy(x1+15,y1+255,"EL MISMO
QUE INSERTADO AL SISTEMA.");
setcolor(LIGHTGREEN);
outtextxy(x1+30,y1+270,"PRESIONE
PAGE UP/PG DOWN PARA CAMBIAR DE");
outtextxy(x1+45,y1+285,"PAGINA O
'ESC' PARA REGRESAR AL MENU.");
REACTCH2;
PAGE=getch();
if(PAGE==OX1b)goto MENU;
if((PAGE==0))goto REACTCH2;
if(PAGE==0){
PAGE=getch();
if(PAGE==OX51||PAGE==OX50||PAGE==0
X4d)goto WINCH3;
if(PAGE==OX48||PAGE==OX49||PAGE==0
X4b)goto WINCH1;
WINCH3;
CLEARWIN(x1,y1,x2,y2);
setcolor(WHITE);
outtextxy(x1+45,y1+20,"LA OPCION 2
SERVE PARA DAR DATOS AL");
outtextxy(x1+15,y1+35,"INTEGRADO
QUE SE QUIERE PROBAR MANUALMENTE");
outtextxy(x1+15,y1+50,"SELECCIONANDO LOS
DATOS DE ACUERDO A UNA");
outtextxy(x1+15,y1+65,"TABLA QUE
APARECE EN UN RECUADRO DE LA");
outtextxy(x1+15,y1+80,"PANTALLA
QUE APARECE. LOS DATOS QUE ESTAN");
outtextxy(x1+15,y1+95,"SIENDO
INTRODUCIDOS SE VAN MOSTRANDO A
UN");
outtextxy(x1+15,y1+110,"LADO DE
LOS PINES EN UNA BASE MOSTRADA EN");
outtextxy(x1+15,y1+125,"LA
PANTALLA.");
setcolor(LIGHTRED);
outtextxy(x1+45,y1+140,"PRECAUCION.");
setcolor(WHITE);
outtextxy(x1+15,y1+155,"ASEGURESE
QUE LOS DATOS DADOS AL CIRCUITO");
outtextxy(x1+15,y1+170,"INTEGRADO
EN LA BASE CORRESPONDAN CON LOS");
outtextxy(x1+15,y1+185,"DATOS
INSERTADOS AL SISTEMA, YA QUE SI SE");

```

```

outtextxy(x1+15,y1+200,"INSERTAN
DATOS ERRONEOS, PUEDE DETERIORARSE");
outtextxy(x1+15,y1+215,"EL SISTEMA
Y EL CIRCUITO DE PRUEBA.");
setcolor(LIGHTGREEN);
outtextxy(x1+30,y1+270,"PRESIONE
PAGE UP/PG DOWN PARA CAMBIAR DE");
outtextxy(x1+45,y1+285,"PAGINA O
'ESC' PARA REGRESAR AL MENU.");

```

```

REACTCH3:
PAGE=getch();
if(PAGE==0X1b)goto MENU;
if((PAGE==0))goto REACTCH3;
if(PAGE==0){
PAGE=getch();

```

```

if(PAGE==0X51||PAGE==0X50||PAGE==0
X4d)goto WINCH4;

```

```

if(PAGE==0X48||PAGE==0X49||PAGE==0
X4b)goto WINCH2;

```

```

WINCH4:
CLEARWIN(x1,y1,x2,y2);
setcolor(WHITE);

```

```

outtextxy(x1+45,y1+20,"LA OPCION 3
SIRVE PARA EDITAR TABLAS DE");
outtextxy(x1+15,y1+35,"DATOS PARA
CIRCUITOS INTEGRADOS QUE NO");
outtextxy(x1+15,y1+50,"EXISTAN EL
NUMERO EN LA LISTA DE INTEGRADOS");
outtextxy(x1+15,y1+65,"A PROBAR.
AL ELEGIR ESTA OPCION, APARECE UNA");
outtextxy(x1+15,y1+80,"PANTALLA,
EN LA CUAL SE PIDEN LAS CARACTER");
outtextxy(x1+15,y1+95,"RISTICAS DEL
NUEVO CIRCUITO INTEGRADO, ASI");
outtextxy(x1+15,y1+110,"COMO
TAMBIEN SE MUESTRAN LOS DATOS QUE
SE");

```

```

outtextxy(x1+15,y1+125,"ESTAN
INSERTANDO A UN LADO DEL PIN REQUE-");
outtextxy(x1+15,y1+140,"RIDO POR EL
SISTEMA SEGUN CORRESPONDE AL");
outtextxy(x1+15,y1+155,"NUMERO DE
PIN DEL NUEVO INTEGRADO.");
setcolor(LIGHTRED);

```

```

outtextxy(x1+45,y1+170,"PRECAUCION ");
setcolor(WHITE);
outtextxy(x1+45,y1+185,"ASEGURESE
QUE LOS DATOS INSERTADOS AL");
outtextxy(x1+15,y1+200,"SISTEMA
CORRESPONDAN CON LOS DATOS DE LA");

```

```

outtextxy(x1+15,y1+215,"TABLA DE
FUNCIONAMIENTO DEL INTEGRADO, YA");
outtextxy(x1+15,y1+230,"QUE SI
EXISTEN DATOS ERRONEOS EN LA NUEVA");
outtextxy(x1+15,y1+245,"TABLA DE
DATOS, PUEDEN DETERIORARSE TANTO");
outtextxy(x1+15,y1+260,"EL SISTEMA
COMO EL CIRCUITO DE PRUEBA.");
setcolor(LIGHTGREEN);

```

```

outtextxy(x1+30,y1+275,"PRESIONE
PAGE UP/PG DOWN PARA CAMBIAR DE");
outtextxy(x1+45,y1+290,"PAGINA O
'ESC' PARA REGRESAR AL MENU.");

```

```

REACTCH4:
PAGE=getch();
if(PAGE==0X1b)goto MENU;
if((PAGE==0))goto REACTCH4;
if(PAGE==0){
PAGE=getch();

```

```

if(PAGE==0X51||PAGE==0X50||PAGE==0
X4d)goto WINCH5;

```

```

if(PAGE==0X48||PAGE==0X49||PAGE==0
X4b)goto WINCH3;

```

```

WINCH5:
CLEARWIN(x1,y1,x2,y2);
setcolor(WHITE);

```

```

outtextxy(x1+45,y1+40,"LA OPCION 4
SIRVE PARA REGRESAL AL MENU");
outtextxy(x1+15,y1+55,"DE LA
PANTALLA PRINCIPAL.");
setcolor(LIGHTGREEN);
outtextxy(x1+30,y1+275,"PRESIONE
PAGE UP/PG DOWN PARA CAMBIAR DE");
outtextxy(x1+45,y1+290,"PAGINA O
'ESC' PARA REGRESAR AL MENU.");

```

```

REACTCH5:
PAGE=getch();
if(PAGE==0X1b)goto MENU;
if((PAGE==0))goto REACTCH5;
if(PAGE==0){
PAGE=getch();

```

```

if(PAGE==0X48||PAGE==0X49||PAGE==0
X4b)goto WINCH4;

```

```

if(PAGE==0X1b)goto REACTCH5;
case '3': CLEARWIN(x1,y1,x2,y2);
settextstyle(TRIPLE_X_FONT,
HORIZ_DIR,1);

```

```

setcolor(YELLOW);
outtextxy(x1+110,y1+20,"BUSCAR
INTEGRADO.");
settextstyle(DEFAULT_FONT,
HORIZ_DIR,1);

```



```

setcolor(WHITE);
outtextxy(x1+45,y1+50,"ESTA OPCION
ACTIVA UNA VENTANA EN LA QUE");
outtextxy(x1+15,y1+65,"SE TIENE QUE
COLOCAR EL NUMERO DE CIRCUITO");
outtextxy(x1+15,y1+80,"INTEGRADO
QUE QUIERE PROBARSE CON EL SISTEMA");
outtextxy(x1+15,y1+95,"Y QUE SE
DESCONOCE SI SE ENCUENTRA ESTE");
outtextxy(x1+15,y1+110,"NUMERO EN
LA LIBRERIA DE CIRCUITOS");
outtextxy(x1+35,y1+125,"DESPUES DE
INSERTAR EL NUMERO DE CIRCUITO");
outtextxy(x1+15,y1+140,"INTEGRADO
A BUSCAR, SI LA PANTALLA SE VUELVE");
outtextxy(x1+15,y1+155,"DE COLOR
AZUL Y COLOCA EL NUMERO INSERTADO");
;
outtextxy(x1+15,y1+170,"QUIERE
DECIR QUE EL INTEGRADO SI EXISTE EN");
outtextxy(x1+15,y1+185,"LA
LIBRERIA, PARA CUANDO NO EXISTE EL
NUMERO");
outtextxy(x1+15,y1+200,"BUSCADO,
LA PANTALLA SE VOLVERA NEGRA Y SE");

outtextxy(x1+15,y1+215,"DESPLGARA EL
MENSAJE *NO EXISTE EL NUMERO*");
outtextxy(x1+15,y1+230,"DE
CIRCUITO* Y COLOCARA TAMBIEN EL
NUMERO");
outtextxy(x1+15,y1+245,"DE
CIRCUITO INTEGRADO BUSCADO");
setcolor(LIGHTGREEN);
outtextxy(x1+45,y1+270,"PRESIONE
*ESC PARA REGRESAR AL MENU.");
RECT=1;
PAGE=getch();
if(PAGE==0x1b)goto MENU;
if(!!(PAGE==0x1b))goto REACTB1;
case '4': CLEARWIN(x1,y1,x2,y2);
settextstyle(TRIPLEX_FONT,
HORIZ_DIR,1);
setcolor(YELLOW);
outtextxy(x1+150,y1+20,"AYUDA.");
settextstyle(DEFAULT_FONT,
HORIZ_DIR,1);
setcolor(WHITE);
outtextxy(x1+45,y1+50,"ESTA OPCION
ACTIVA UNA VENTANA EN LA QUE");
outtextxy(x1+15,y1+65,"SE
DESPLIEGA UN MENSAJE, EL CUAL
PREGUNTA SI");
outtextxy(x1+15,y1+80,"SE QUIERE
SALIR DEL SISTEMA, Y AL RESPONDER");
outtextxy(x1+15,y1+95,"DE MODO
AFIRMATIVO APARECERA UN SEGUNDO");
outtextxy(x1+15,y1+110,"MENSAJE EN
EL CUAL SE DICE QUE SE PRESIONE");
outtextxy(x1+15,y1+125,"LA TECLA
ENTER PARA DESACTIVAR EL SISTEMA");

```

```

outtextxy(x1+15,y1+80,"EL NUMERO
CORRESPONDIENTE AL TEMA QUE SE");
outtextxy(x1+15,y1+95,"DESEA
OBTENER AYUDA HACE QUE APARESCA
LA");
outtextxy(x1+15,y1+110,
"INFORMACION ACERCA DEL TEMA
SELECCIONADO.");
outtextxy(x1+45,y1+125,"EN
ALGUNOS CASOS EXISTEN DATOS ACERCA
DE");
outtextxy(x1+15,y1+140,"ALGUNA
FUNCION DE VARIAS PAGINAS, Y PARA");
outtextxy(x1+15,y1+155,"CAMBIAR
DE PAGINA, SOLO SE TIENEN QUE");
outtextxy(x1+15,y1+170,"PRESIONAR
LAS FLECHAS, O LAS TECLAS PARA");
outtextxy(x1+15,y1+185,"AVANZAR O
RETROCEDER PAGINA, Y PARA REGRESAR");
outtextxy(x1+15,y1+200,"AL MENU
PRINCIPAL O SALIR DE CUALQUIER TEMA");
outtextxy(x1+15,y1+215,
"SELECCIONADO SOLO SE TIENE QUE
PRESIONAR LA");
outtextxy(x1+15,y1+230,"TECLA
*ESC.");
setcolor(LIGHTGREEN);
outtextxy(x1+45,y1+270,"PRESIONE
*ESC PARA REGRESAR AL MENU.");
RECTA1;
PAGE=getch();
if(PAGE==0x1b)goto MENU;
if(!!(PAGE==0x1b))goto REACTA1;
case '5': CLEARWIN(x1,y1,x2,y2);
settextstyle(TRIPLEX_FONT,
HORIZ_DIR,1);
setcolor(YELLOW);
outtextxy(x1+130,y1+20,"SALIR.");
settextstyle(DEFAULT_FONT,
HORIZ_DIR,1);
setcolor(WHITE);
outtextxy(x1+45,y1+50,"ESTA OPCION
ACTIVA UNA VENTANA EN LA QUE");
outtextxy(x1+15,y1+65,"SE
DESPLIEGA UN MENSAJE, EL CUAL
PREGUNTA SI");
outtextxy(x1+15,y1+80,"SE QUIERE
SALIR DEL SISTEMA, Y AL RESPONDER");
outtextxy(x1+15,y1+95,"DE MODO
AFIRMATIVO APARECERA UN SEGUNDO");
outtextxy(x1+15,y1+110,"MENSAJE EN
EL CUAL SE DICE QUE SE PRESIONE");
outtextxy(x1+15,y1+125,"LA TECLA
ENTER PARA DESACTIVAR EL SISTEMA");

```

```

outtextxy(x1+15,y1+140,"PRESIONAR
LA TECLA 'ESC' PARA CANCELAR LA");
outtextxy(x1+15,y1+155,"SALIDA DEL
SISTEMA Y REGRESAR A LA PANTALLA");
outtextxy(x1+15,y1+170,"DEL MENU
PRINCIPAL.");
setcolor(LIGHTGREEN);
outtextxy(x1+45,y1+270,"PRESIONE
'ESC' PARA REGRESAR AL MENU.");
REACTS1:
PAGE=getch();
if(PAGE==0x1b)goto MENU;
if(! (PAGE==0x1b))goto REACTS1;
case '6' : CLEARWIN(x1,y1,x2,y2);
settextstyle(TRIPLEX_FONT,
HORIZ_DIR,1);
delay(400);
setcolor(BLUE);

outtextxy(x1+38,y1+30,"UNIVERSIDAD
NACIONAL AUTONOMA");
setcolor(WHITE);

outtextxy(x1+35,y1+33,"UNIVERSIDAD
NACIONAL AUTONOMA");
delay(500);
setcolor(BLUE);
outtextxy(x1+138,y1+60,"DE
MEXICO.");
setcolor(WHITE);
outtextxy(x1+135,y1+63,"DE
MEXICO.");
setcolor(BLUE);
delay(1000);
outtextxy(x1+33,y1+120,"FACULTAD
DE ESTUDIOS SUPERIORES");
setcolor(WHITE);
outtextxy(x1+30,y1+123,"FACULTAD
DE ESTUDIOS SUPERIORES");
delay(500);
setcolor(BLUE);

outtextxy(x1+133,y1+150,"CUAUTITLAN.");
setcolor(WHITE);

outtextxy(x1+130,y1+153,"CUAUTITLAN.");
delay(1000);
setcolor(BLUE);
outtextxy(x1+43,y1+210,"SISTEMA
PROBADOR DE CIRCUITOS");
setcolor(WHITE);
outtextxy(x1+40,y1+213,"SISTEMA
PROBADOR DE CIRCUITOS");
delay(500);

setcolor(BLUE);
outtextxy(x1+33,y1+240,"INTEGRADOS POR
MEDIO DE UNA PC.");
setcolor(WHITE);

outtextxy(x1+30,y1+243,"INTEGRADOS POR
MEDIO DE UNA PC.");
delay(3000);
CLEARWIN(x1,y1,x2,y2);
delay(1000);
setcolor(BLUE);
outtextxy(x1+18,y1+30,"ESTE
SISTEMA FUE IMPLEMENTADO POR.");
setcolor(WHITE);
outtextxy(x1+15,y1+33,"ESTE
SISTEMA FUE IMPLEMENTADO POR.");
delay(1000);
setcolor(BLUE);
outtextxy(x1+48,y1+70,"GUSTAVO
ALBERTO RUIZ TORRES.");
setcolor(YELLOW);
outtextxy(x1+45,y1+73,"GUSTAVO
ALBERTO RUIZ TORRES.");
delay(1000);
setcolor(BLUE);
outtextxy(x1+28,y1+110,"JUAN
ANTONIO RESENDIZ CAPELILLO.");
setcolor(YELLOW);
outtextxy(x1+25,y1+113,"JUAN
ANTONIO RESENDIZ CAPELILLO.");
delay(1000);
setcolor(BLUE);
outtextxy(x1+18,y1+160,"PARA
OBTENER EL TITULO DE");
setcolor(WHITE);
outtextxy(x1+15,y1+163,"PARA
OBTENER EL TITULO DE");
delay(1000);
settextstyle(TRIPLEX_FONT,
HORIZ_DIR,3);
setcolor(BLUE);
outtextxy(x1+68,y1+210,"INGENIERIA
MECANICA");
setcolor(YELLOW);
outtextxy(x1+65,y1+213,"INGENIERIA
MECANICA");
delay(1000);
setcolor(BLUE);

outtextxy(x1+128,y1+250,"ELECTRICA");
setcolor(YELLOW);

outtextxy(x1+125,y1+253,"ELECTRICA");

```

```

delay(5000);
CLEARWIN(x1,y1,x2,y2);
settextstyle(TRIPLEX_FONT,
HORIZ_DIR,1);
delay(500);
setcolor(BLUE);

outtextxy(x1+38,y1+30,"PERTENECIENTES A LA
GENERACION");
setcolor(WHITE);

outtextxy(x1+35,y1+33,"PERTENECIENTES A LA
GENERACION");
delay(1000);
setcolor(BLUE);
outtextxy(x1+128,y1+60,"1991 -
1993");
setcolor(WHITE);
outtextxy(x1+125,y1+63,"1991 -
1993");
delay(1000);
setcolor(BLUE);
outtextxy(x1+38,y1+120,"EL ASESOR
DE LA TESIS FUE:");
setcolor(WHITE);
outtextxy(x1+35,y1+123,"EL ASESOR
DE LA TESIS FUE:");
delay(1000);
setcolor(BLUE);
outtextxy(x1+68,y1+150,"ING. JORGE
BUENDIA GOMEZ");
setcolor(WHITE);
outtextxy(x1+65,y1+153,"ING. JORGE
BUENDIA GOMEZ");
delay(1000);
setcolor(BLUE);
outtextxy(x1+168,y1+200,"1996");
setcolor(WHITE);
outtextxy(x1+165,y1+203,"1996");
delay(4000);
goto MENU;

case 0x1b: return;
case 0x0: getch();
default: BELL();
goto MENU;
}
)

```

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA PROBADOR DE  
CIRCUITOS INTEGRADOS DIGITALES CONTROLADO POR UNA  
COMPUTADORA**

**BIBLIOGRAFIA**

**BIBLIOGRAFIA****Lógica Digital y Diseño de Computadores****M. Morris Mano****Prentice Hall Hispanoamericana S. A., 1989****Principios Digitales****Segunda Edición****Robert L. Tokheim****Mc Graw Hill****Electrónica Digital Básica con Aplicaciones****John A. Dempsey****Editorial Alfa Omega, 1992****The Design of Digital Systems****John B. Peatman****Internacional Students Edition / Mc Graw Hill****Manual de Referencia ECG****Philips, 1991.**

**C Manual de Referencia**

Segunda Edición

Herbert Schildt

Mc Graw Hill

**Aplique C++**

Bruce Eckel

Osborne / Mc Graw Hill

**Tubo C++ Programming -An Objet Oriented Approach-**

Ben Ezzell

Addison Wesley

**Using Turbo C++**

Herbert Schildt

Borland Osborne / Mc Graw Hill, 1989

**Fast and LS TTL Data**

Logic Integrated Circuits Division

Second Printing

Motorola 1992