ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

"DISPOSITIVO PROBADOR DE CIRCUITOS INTEGRADOS DE LA FAMILIA CMOS"

IVAN RODRIGO AGUIRRE AYALA

TESIS PREVIA A LA OBTENCION DEL TITULO DE

INGENIERO EN ELECTRONICA Y TELECOMUNICACIONES

DICIEMBRE - 1996

Certificación:

Certifico que bajo mi dirección la presente tesis fue desarrollada en su totalidad por el señor Iván Rodrigo Aguirre Ayala.

Ing. Jaime Velarde G.
DIRECTOR DE TESIS

Dedicatoria

A Dios por permitirme culminar mi carrera. A la abnegada labor de mis padres y al apoyo y estímulo de Erika.

Agradecimiento

Al Ing. Jaime Velarde por su acertada dirección durante el desarrollo de la presente tesis.

INDICE

INTRODUCCION

La familia CMOS, a pesar de tener características que la singularizan de otras familias lógicas, no ha tenido una aceptación tan grande en nuestro medio como la tecnología TTL. Una razón para ello podría ser el cuidado con el que se debe manipular los circuitos integrados de esta familia, o también su costo superior comparado con otras tecnologías. Sin embargo, los circuitos integrados CMOS ofrecen varias ventajas que compensan el incremento de precio y de hecho el resultado final puede ser menos costoso que si se hubiera diseñado con otras familias.

Entre las características especiales que ofrecen los circuitos integrados CMOS, se tienen las siguientes:

- Baja disipación de potencia
- Bajos tiempos de propagación
- Inmunidad al ruido del 50% aproximadamente

Esto da la ventaja de poder diseñar nuestros dispositivos con fuentes de alimentación más pequeñas, pues la disipación de potencia es menor si se utiliza circuitos integrados CMOS. De la misma manera, seguramente no se requerirá de ventiliadores u otros métodos de

enfriamiento que posiblemente serían necesarios si se utiliza tecnología TTL. De lo dicho anteriormente se desprende la necesidad de incorporar la tecnología CMOS en nuevos diseños, pero también se hace necesario disponer de una forma rápida para comprobar el funcionamiento de circuitos integrados CMOS.

Es necesario también aprender a diseñar utilizando otras tecnologías y más que nada saber combinar las ventajas de cada una en una determinada aplicación.

La presente tesis tiene por objeto diseñar y construir un prototipo de un probador de circuitos integrados de la familia CMOS. El equipo de prueba esta basado en un microcontrolador del tipo 8751 que tiene por función aplicar los niveles lógicos a los pines del circuito integrado en prueba según los datos recibidos de un computador personal mediante el puerto serial.

El prototipo desarrollado permite:

- Probar las funciones lógicas de los circuitos integrados de la familia CMOS.
- Polarizar los circuitos integrados a probarse con los siguientes voltajes: +5V/GND, +5V/-5V, +9V/GND, +9V/-9V.

- Enlazar el equipo con un computador personal mediante puerto serial, para lo cual se desarrolla el software requerido a nivel de PC utilizando QBASIC para realizar las siguientes tareas:
- Seleccionar el tipo de circuito integrado
- Seleccionar los voltajes de polarización
- Obtener de una librería la tabla de verdad correspondiente a los circuitos integrados que pueden probarse
- Enviar al equipo los datos necesarios para poder realizar una prueba de funcionamiento.
- Presentar en la pantalla el resultado de la prueba.
- Permite actualizar la librería con nuevos circuitos integrados.

El equipo tiene la ventaja de polarizar con varios voltajes, positivos y negativos. Los probadores comerciales por lo general solamente polarizan a +5V y GND, mientras que muchas aplicaciones pueden estar fuera de este rango de voltaje. A pesar de ésto su tamaño no es muy grande, haciéndolo un equipo portátil y muy fácil de usar, pues se tiene un computador personal como interfaz entre el probador y el usuario.

La mayor dificultad en el diseño del equipo constituye la etapa de conversión de niveles lógicos de CMOS a TTL y viceversa. Esta etapa es indispensable si se desea polarizar con diferentes voltajes. La no existencia de convertidores bidireccionales obligó a la utilización de

interruptores analógicos para habilitar solamente uno de los convertidores de cada línea de datos a un determinado momento, dependiendo de su función.

Con el fin de limitar el número de circuitos integrados requeridos para la implementación del equipo se prefirió utilizar el microcontrolador 8751, pues al tener EPROM interna evita la necesidad de incorporar una EPROM externa para el programa del microprocesador, así como circuitos integrados adicionales.

Debido a que el microcontrolador del equipo de prueba realiza únicamente enrutamiento de datos y no procesamiento, basta con almacenar las características de los circuitos integrados a probarse en una librería en el computador personal. Este es el encargado de seleccionar y probar los circuitos integrados. Cualquier computador personal que tenga puerto serial puede ser utilizado con el equipo de prueba, lo que vuelve al probador muy versátil.

Finalmente se escogió la técnica de "wire-wrapping" por resultar más cómoda y rápida que la construcción de un circuito impreso. Además el resultado fue un equipo de menores dimensiones que lo que se hubiera tenido con otra técnica.

CAPITULO II

DESCRIPCION DEL EQUIPO DE PRUEBA

El equipo de prueba tiene por función verificar el funcionamiento lógico de circuitos integrados de la familia CMOS. Con "funcionamiento lógico" nos referimos a la tabla de verdad correspondiente a la serie del circuito integrado.

2.1.- TECNOLOGIA CMOS: ESTRUCTURA Y FUNCIONAMIENTO DE LOS CIRCUITOS INTEGRADOS

En muchos aspectos, la familia CMOS podría considerarse como la familia lógica ideal. Disipa poca potencia, su tiempo de propagación es relativamente bajo (aunque más alto que el de la familia TTL) y tiene una inmunidad al ruido igual al 50% de la variación lógica.

En primer lugar la familia CMOS disipa baja potencia. Típicamente, la disipación estática de potencia es de 10 nW por compuerta que es debido al flujo de corrientes de fuga. La potencia activa depende del voltaje de la fuente de poder, frecuencia, carga de salida y tiempo de subida, pero típicamente la disipación de una compuerta a 1Mhz con una carga de 50 pF es menor a 10 mW.

En segundo lugar los retardos de propagación en CMOS son cortos, aunque no se aproximan a cero. Dependiendo del voltaje de alimentación el retardo a través de una compuerta típica esta en el orden de 25 ns a 50 ns. Además, los tiempos de subida y bajada son controlados, tendiendo a ser rampas y no funciones de escalera. Típicamente, tiempos de subida y bajada tienden a ser de 20% a 40% más largos que los retardos de propagación.

Finalmente, la inmunidad al ruido se aproxima al 50%, siendo típicamente 45% del rango lógico total.

A pesar de que a nivel de componentes, la tecnología CMOS es más costosa que la tecnología TTL, el costo a nivel de sistema puede ser menor. Las fuentes de energía en un sistema CMOS son más baratas, pues se pueden construír más pequeñas y con menor regulación. Debido a las corrientes menores, el sistema de distribución de la fuente de energía puede ser más sencillo y por ende, más barato. No se requiere de ventiladores u otro equipo de enfriamiento pues la disipación es menor. Además, ya que se tiene tiempos de subida y bajada más largos, la transmisión de señales digitales se vuelve más sencilla, pudiéndose utilizar técnicas menos costosas.

Por último, no hay razón técnica alguna para que los precios CMOS no se aproximen a los precios TTL de hoy en día.

2.1.1 CARACTERISTICAS Y FUNCIONAMIENTO DE LA FAMILIA CMOS[3]

El circuito CMOS básico es el inversor mostrado en la figura 2.1. Consiste en dos transistores MOS de enriquecimiento, el superior de canal P y el inferior de canal N.

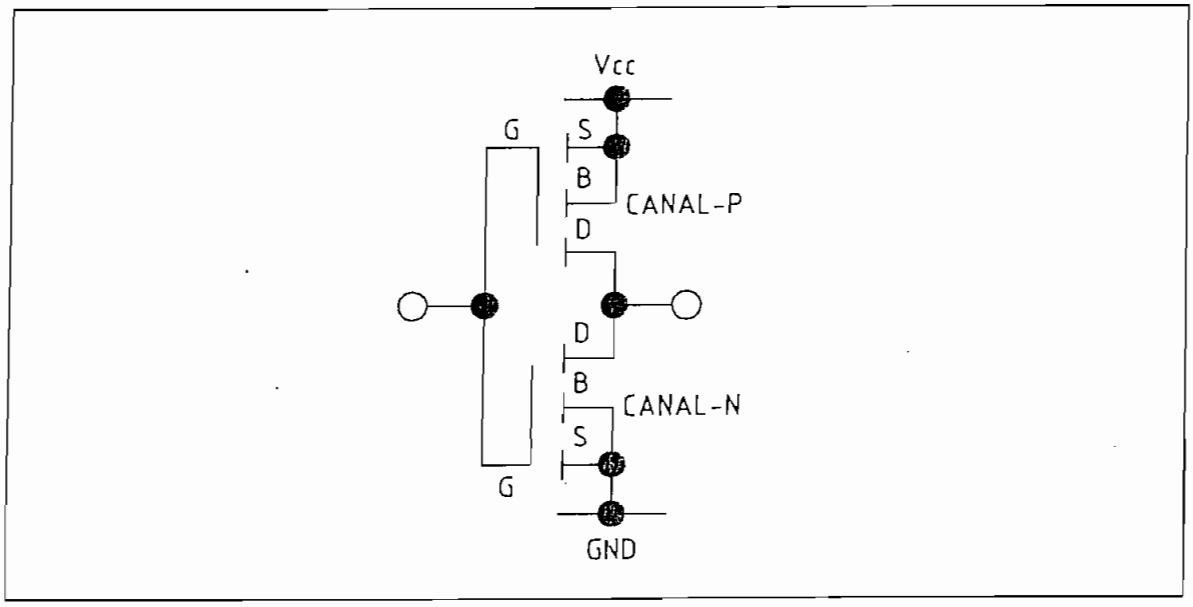

FIGURA 2.1

Las fuentes de poder para CMOS se denominan V_{DD} y V_{SS}, o V_{CC} y tierra dependiendo del fabricante. V_{DD} y V_{SS} vienen de la nomenclatura convencional de circuitos MOS y se refieren al drenaje

("drain") y fuente ("source"). Estos no se aplican directamente a CMOS ya que ambos sustratos son en realidad fuentes. V_{CC} y GND son, en cambio, rezagos de la lógica TTL.

Los niveles lógicos en un sistema CMOS son V_{CC} ("1" lógico) y tierra ("0" lógico). Dado que un transistor MOS "encendido" tiene virtualmente ninguna caída de voltaje en sus terminales si no hay corriente fluyendo a través de él, y dado que la impedancia de entrada de un dispositivo CMOS es tan alta (la característica de entrada de un transistor MOS es escencialmente capacitiva, asemejándose a una resistencia de 10^{12} Ω en paralelo con un capacitor de 5 pF) que los niveles lógicos encontrados en un sistema CMOS serán escencialmente iguales a las fuentes de voltaje.

Con respecto a las curvas características de transistores MOS, podemos ver en la figura 2.2 las correspondientes a transistores canal N y canal P de enriquecimiento. Si nos fijamos en la curva V_{GS} = 15V para el transistor canal N, podemos notar que para un voltaje V_{GS} constante, el transistor se comporta como una fuente de corriente para V_{DS} mayor que V_{GS} - V_T (V_T es el voltaje de umbral de un transistor MOS). Para V_{DS} menor que V_{GS} - V_T, el transistor se comporta escencialmente como una resistencia. También se debe notar que para V_{GS} menores hay curvas similares excepto que las magnitudas de las corrientes I_{DS} son significativamente más pequeñas y que, de hecho,

I_{DS} se incrementa aproximadamente de forma proporcional al cuadrado de V_{GS}. El transistor canal P exhibe características escencialmente idénticas, pero complementarias.

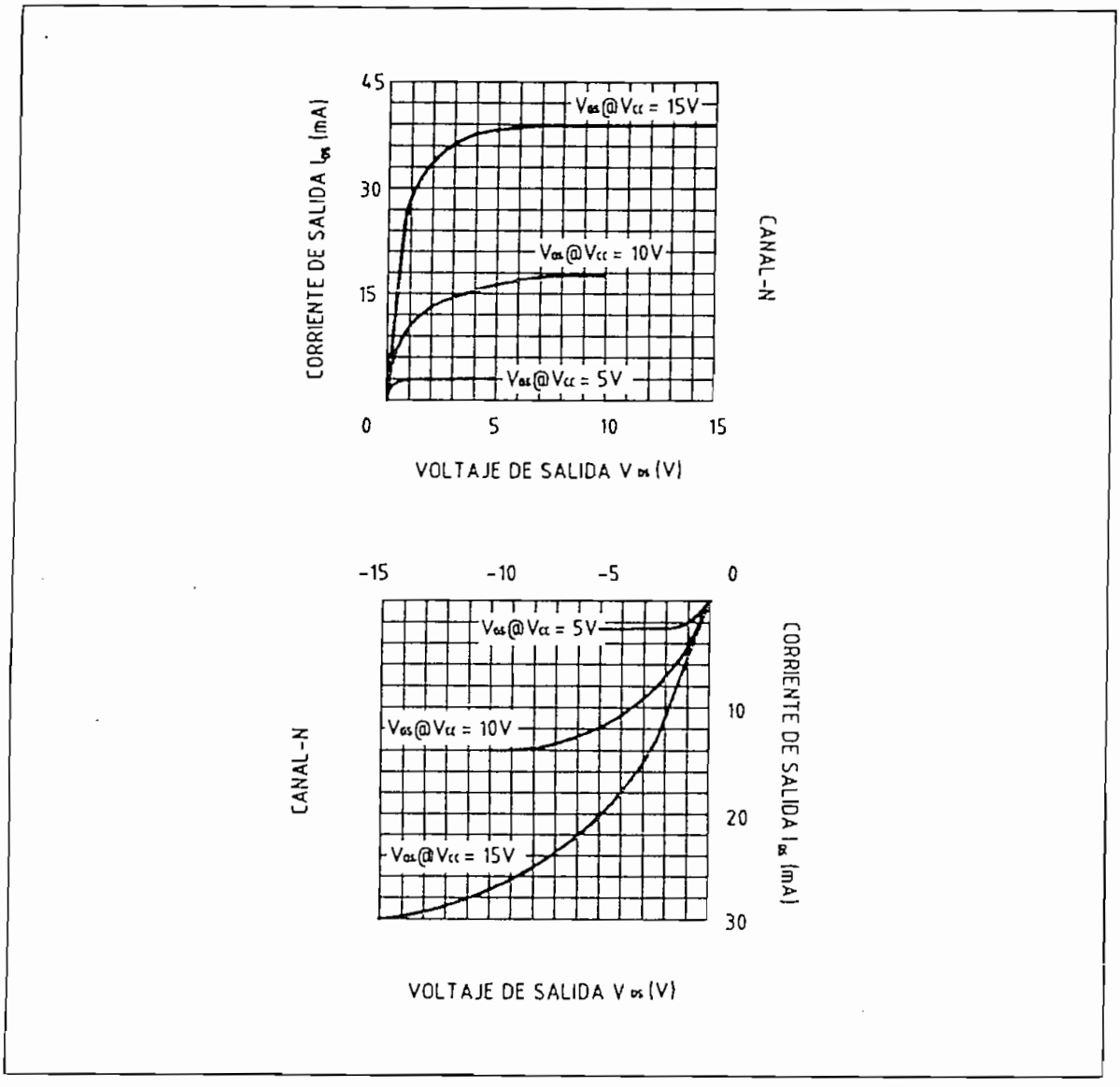

FIGURA 2.2

Si se intenta manejar una carga capacitiva con estos dispositivos, se puede ver que el cambio inicial de voltaje a través de la carga tendrá la forma de rampa debido a la característica de fuente de corriente seguido por un redondeo debido a la característica resistiva que empieza a dominar conforme V_{DS} se aproxima a cero. Refiriendo esto al inversor básico CMOS de la figura 2.1, conforme V_{DS} se hace cero, V_{OUT} se aproximará a V_{CC} o tierra dependiendo de cual transistor esta en conducción, el canal P o el canal N.

Si se incrementa V_{CC}, y por lo tanto también V_{GS}, el inversor debe manejar al capacitor en un rango mayor de voltaje. Sin embargo, para este mismo incremento de voltaje, la capacidad de corriente (I_{DS}) se ha incrementado aproximadamente con el cuadrado de V_{GS} y, por lo tanto, los tiempos de subida y retardos de propagación a través del inversor medidos en la figura 2.3 se han reducido.

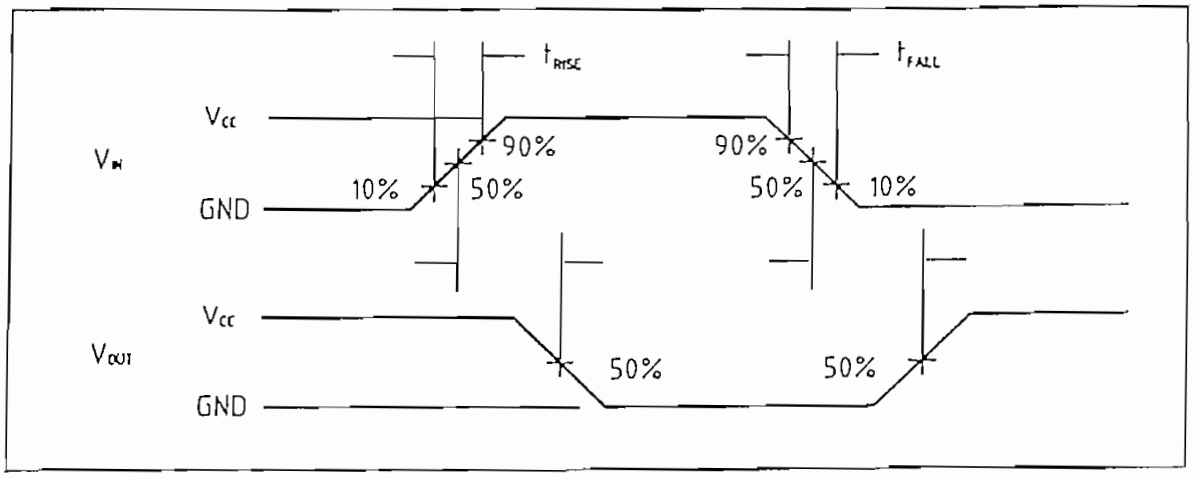

FIGURA 2.3

De lo anterior se ha visto que para un diseño dado, y por tanto carga capacitiva fija, incrementar el voltaje de la fuente incrementará la velocidad del sistema. Al incrementar V_{CC} se incrementa la velocidad pero también aumenta la disipación de potencia. Esto es verdad por dos razones. Primero, la potencia CV^2f se incrementa. Esta es la potencia disipada en un circuito CMOS, o cualquier circuito que maneja una carga capacitiva.

Para una carga capacitiva dada y frecuencia variable, la disipación de potencia se incrementa con el cuadrado del cambio de voltaje a través de la carga.

La segunda razón es que la potencia VI disipada en el circuito CMOS se incrementa con V_{CC} (para $V_{CC} > 2V_T$). Cada vez que el circuito conmuta, una corriente fluye momentáneamente de V_{CC} a tierra a través de ambos transistores de salida. Dado que los voltajes de umbral de los transistores no cambian con el incremento en V_{CC}, el rango de voltaje de entrada en el cual el transistor superior y el inferior conducen simultáneamente aumenta conforme V_{CC} aumenta. Al mismo tiempo, el V_{CC} más alto produce voltajes V_{GS} más altos que también incrementan la corriente I_{DS}. Si el tiempo de subida de la señal de entrada fuera cero, no se tuviera flujo de corriente desde V_{CC} a tierra en el circuito. Esta corriente fluye porque la señal de entrada

tiene un tiempo de subida finito y, por tanto, el voltaje en la entrada se demora un tiempo finito pasando por la región en que ambos transistores conducen simultáneamente. Obviamente, se debe mantener los tiempos de subida y bajada en un mínimo para que la potencia disipada VI no sea alta.

Con respecto a las características de transferencia, en la figura 2.4 se puede ver los varios casos cuando V_{CC} cambia. Se asumirá que los dos transistores en el inversor básico tienen características idénticas pero complementarias, además de voltajes de umbral iguales. Se asumirá, así mismo, $V_T = 2V$. Si V_{CC} es menor que el voltaje de umbral de 2V, ningún transistor puede entrar en conducción y el circuito no puede operar. Si V_{CC} es exactamente igual al voltaje de umbral entonces se esta en la curva mostrada en la figura 2.4a. Aparentemente se tiene una histéresis del 100%. Sin embargo, no es en realidad histéresis, pues ambos transistores de salida están apagados y el voltaje de salida esta fijada en la capacitancia de la compuerta de circuitos sucesivos. Si V_{CC} esta entre V_T y $2V_T$ (figura 2.4b), entonces se tiene menores "histéresis" conforme se aproxima a $V_{CC} = 2V_T$ (figura 2.4c). Cuando $V_{CC} = 2V_T$ no se tiene "histéresis" ni flujo de corriente durante la conmutación de los transistores superior e inferior. Cuando V_{CC} excede $2V_T$ las curvas de transferencia empiezan a redondearse (figura 2.4d). Mientras V_{IN} pasa por la región donde ambos transistores estan

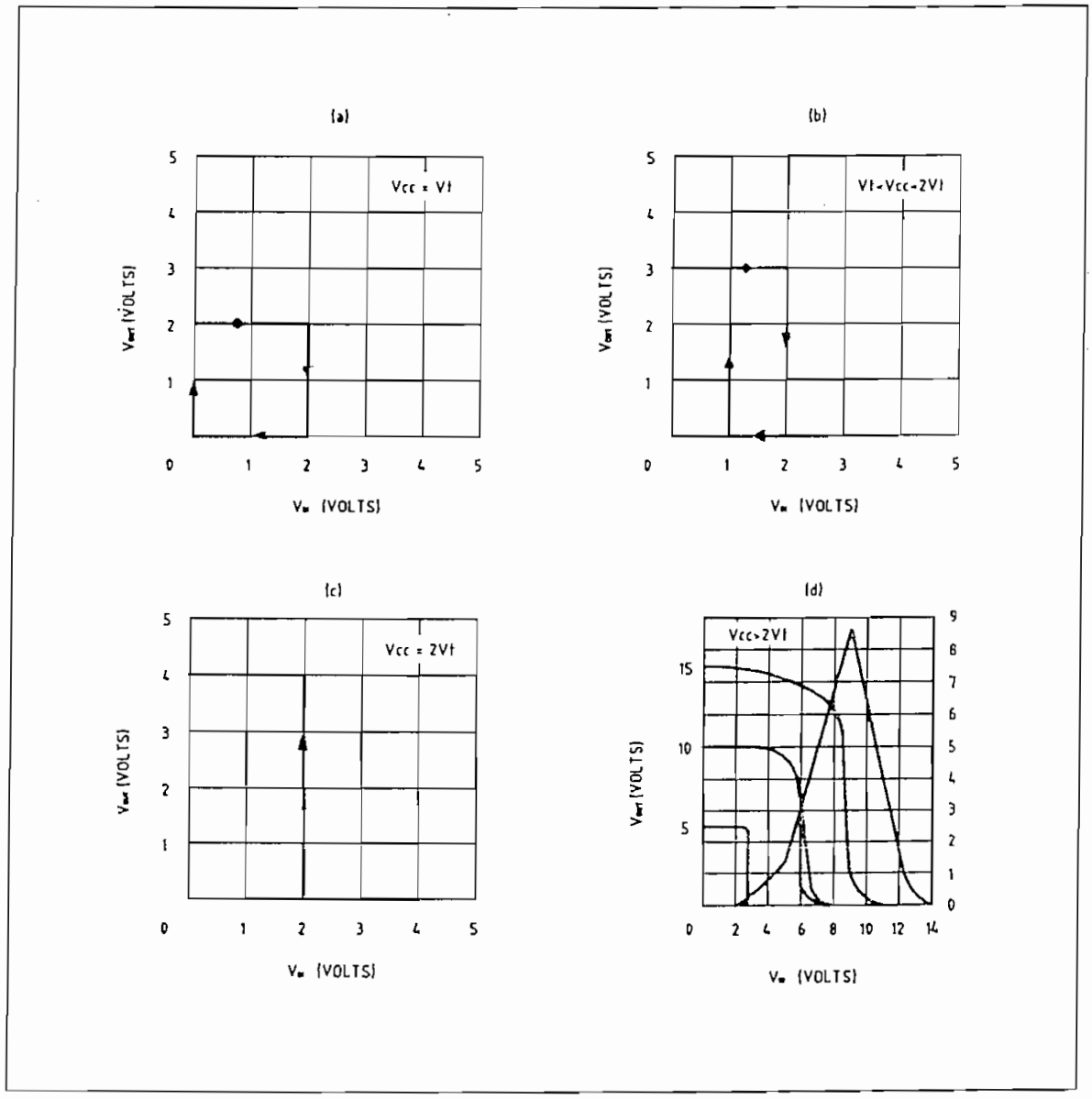

FIGURA 2.4

conduciendo, las corrientes que fluyen a través de los transistores causan caídas de tensión entre sus terminales, dando lugar a la característica redondeada.

Considerando el ruido en un sistema CMOS, se debe estudiar al menos dos tipos: inmunidad al ruido y margen de ruido.

Los circuitos CMOS tienen una inmunidad al ruido típica de 0.45 V_{CC}. Esto significa que una entrada espúrea de 0.45 V_{CC} o menos alejada de V_{CC} o tierra, típicamente no se propagará por el sistema como un nivel lógico erróneo. Esto no significa que no aparecerá una señal a la salida del primer circuito. De hecho, habrá una señal de salida como resultado de la entrada espúrea, pero será reducida en amplitud. Mientras esta señal se propaga a través del sistema, será atenuada aún más por cada circuito que pasa hasta que finalmente desaparezca. Típicamente, no cambiará ninguna señal al nivel lógico opuesto.

Normalmente los circutos CMOS tienen un margen de ruido de 1V DC sobre todo en el rango de voltaje y temperatura y con cualquier combinación de entradas. Esto es simplemente una variación de la especificación de inmunidad al ruido, solamente que ahora se ha seleccionado un juego específico de voltajes de entrada y salida. En otras palabras, la especificación dice que para que la salida de un circuito este dentro de 0.1 V_{CC} de un nivel lógico adecuado (V_{CC} o tierra), debe tenerse la entrada máximo 0.1 V_{CC} más 1V alejada del voltaje apropiado.

Gráficamente se tiene lo siguiente:

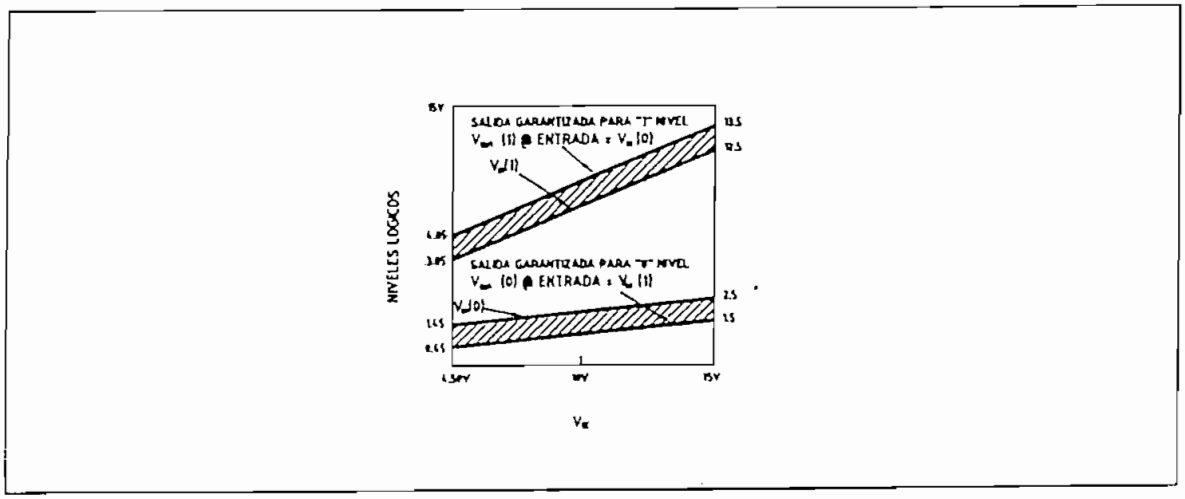

FIGURA 2.5

Esto es similar en naturaleza al margen de ruido estándar para TTL, que es 0.4V. (figura 2.6)

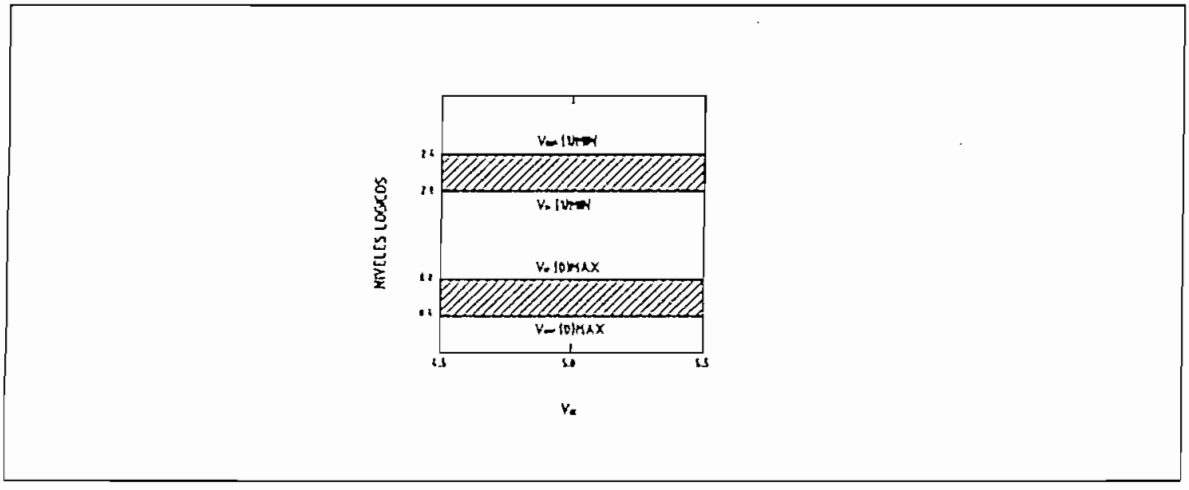

FIGURA 2.6

2.1.2.- DESCARGAS ESTATICAS Y SU EFECTO EN DISPOSITIVOS CMOS

Todos los dispositivos CMOS, que se componen de pares de MOSFETs canal N y canal P complementarios, son susceptibles a daños por descargas de energía electrostática entre dos pines cualesquiera. Esta sensitividad a la carga estática es debido al hecho de que la capacitancia de entrada de la compuerta (5 pF típicamente) en paralelo con una resistencia de entrada extremadamente alta, se presta para una alta impedancia de entrada y por tanto acumula cargas electrostáticas, siempre y cuando no se tomen medidas de precaución. Esta acumulación de voltaje en la compuerta puede fácilmente romper el delgado aislamiento óxido de la compuerta (1000 A), debajo de la porción metálica. Defectos locales como agujeros microscópicos pueden reducir sustancialmente la resistencia dieléctrica desde 8-10x10^6V/cm hasta 3-4x10^6V/cm. Este nuevo valor se vuelve, entonces, el factor limitante para determinar el voltaje que se puede aplicar sin problemas a las compuertas de dispositivos CMOS.

Cuando un voltaje más alto, resultado de una descarga estática, se aplica a un dispositivo, un daño permanente puede ocurrir. Tal daño puede ser un cortocircuito al sustrato, pin V_{DD}, pin V_{SS} o a la salida.

La electricidad estática esta siempre presente en cualquier entorno. Se genera cuando dos materiales diferentes se frotan entre sí. Una persona que camina en un pasillo puede generar una carga de miles de voltios. Una persona trabajando encima de un banco, deslizándose en la silla o frotando sus brazos contra la mesa puede desarrollar un potencial alto.

La tabla 2.1 muestra los resultados del trabajo realizado por Speakman[5] sobre potenciales estáticos generados en diferentes situaciones. La humedad ambiental relativa, por supuesto, tiene un gran efecto en la cantidad de carga estática desarrollada, pues la humedad forma una vía de descarga a tierra, reduciendo la acumulación estática.

CONDICION	PROMEDIO [V]	PICO [V]
PERSONA CAMINANDO SOBRE ALFOMBRA	12000	39000
PERSONA CAMINANDO SOBRE VINIL	4000	13000
PERSONA TRABAJANDO EN UNA MESA	500	3000
DIP DE 16 PINES EN CAJA PLASTICA	3500	12000
DIP DE 16 PINES EN TUBO PLASTICO PARA CHIPS	500	3000

TABLA 2.1.-Voltajes generados en 15%-30% de humedad relativa

Con el fin de proteger la compuerta de óxido contra niveles moderados de descarga electrostática, se añade circuitería de protección al integrado mismo. La figura 2.7 muestra la protección estándar utilizada en dispositivos CMOS. La resistencia serie de 200Ω usando una difusión P^+ ayuda a limitar la corriente cuando la entrada esta sujeta a una descarga de voltaje alto. Asociada con esta resistencia se tiene una red de diodos a V_{DD} que proteje contra transitorios positivos. Un diodo adicional a V_{SS} ayuda a descargar transitorios negativos.

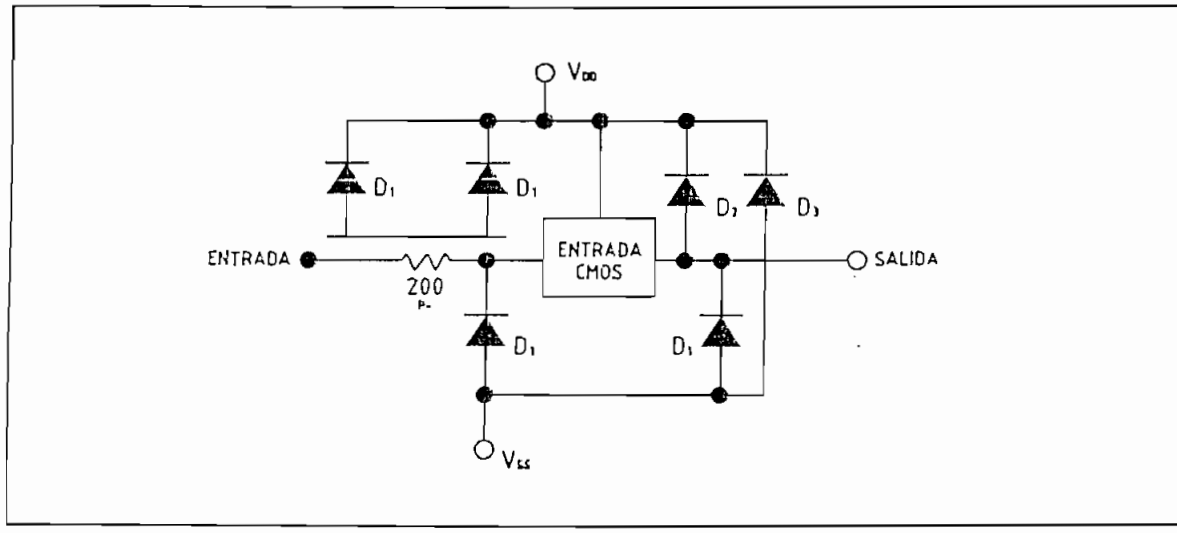

FIGURA 2.7

Finalmente se debe recalcar que al ser estas cargas estáticas también generadas por personas, se debe prestar especial cuidado al manipular tales dispositivos. Se debe evitar diferencias de potencial entre pines del circuito integrado. Al transportar o manipular dichos dispositivos se debe procurar utilizar espuma conductiva o rieles conductivos. En el mercado existen fundas protectoras de estática, por ejemplo de la

compañía 3M, que tienen baja resistividad ($\leq 10^4\Omega$/sq.) y son de poliéster con capa metálica. Estas fundas funcionan bajo el conocido principio de la jaula de Faraday.

2.2.- CONDICIONES A CONSIDERARSE EN EL DISEÑO DEL EQUIPO

Para el diseño del equipo de prueba es necesario tener en cuenta los siguientes requerimientos:

1.- Interfaz para comunicación serial de la norma EIA-RS-232.

2.- Fuentes de alimentación reguladas para voltajes de alimentación a los circuitos integrados.

3.- Convertidor bidireccional de niveles lógicos para cada pin del circuito integrado bajo prueba.

4.- Los pines del zócalo de pruebas deben ser capaces de polarizar o entregar niveles lógicos dependiendo de la función del pin del circuito integrado en prueba.

1.- La comunicación entre el computador personal y el probador de chips se realiza a través del puerto serial, por lo cual es necesario adecuar las señales de dicho puerto del computador a niveles TTL utilizados por el microcontrolador 8751, y viceversa.

2.- Tanto el microcontrolador como la mayoría de circuitos integrados utilizados en el equipo de prueba están polarizados con +5V y GND, a excepción de los que realizan la conversión de TTL a RS-232 que se encuentran polarizados con +/- 15V.

Como el probador de chips es capaz de polarizar al circuito integrado en prueba con diferentes voltajes, positivos y negativos, se debe disponer de reguladores de voltaje variables, los mismos que son controlados por el microcontrolador. En consecuencia se requiere de los siguientes voltajes para polarización y conversión a niveles RS-232:

- +/- 5V
- +/- 9V
- +/- 15V

3.- Debido a que el circuito integrado bajo prueba podrá estar polarizado con voltajes diferentes a los utilizados por los chips del equipo probador, se requiere de líneas bidireccionales que sean capaces de convertir niveles lógicos entre la circuitería del probador y el circuito integrado bajo prueba.

Como es posible imaginar, dependiendo del tipo de circuito integrado que se pruebe, cada pin tendrá su función particular. Por ejemplo, el

pin #14 en chips CMOS de 14 pines puede ser en ocasiones entrada para polarización positiva, mientras que en un chip de otra serie puede ser entrada o salida de datos. El equipo probador, por lo tanto, deberá fijar la dirección de cada línea de acuerdo a la información enviada por el computador personal.

4.- De lo anterior se desprende la necesidad de que los dispositivos de salida a los pines del zócalo de prueba sean capaces no solamente de entregar los niveles lógicos requeridos, sino además tener la capacidad de manejar corrientes de polarización por si requieren ser utilizados como tales.

2.3.- FUNCIONAMIENTO DEL EQUIPO: DIAGRAMA DE BLOQUES Y DESCRIPCION

El diagrama 2.1 presenta un esquema general del probador de circuitos integrados.

Como se mencionó anteriormente la selección del circuito integrado se la realiza en el computador personal. Así mismo, toda la información pertinente a los chips, sus tablas de verdad, pines de entrada y salida, etc, se encuentran en una librería manejada por el programa de pruebas del PC.

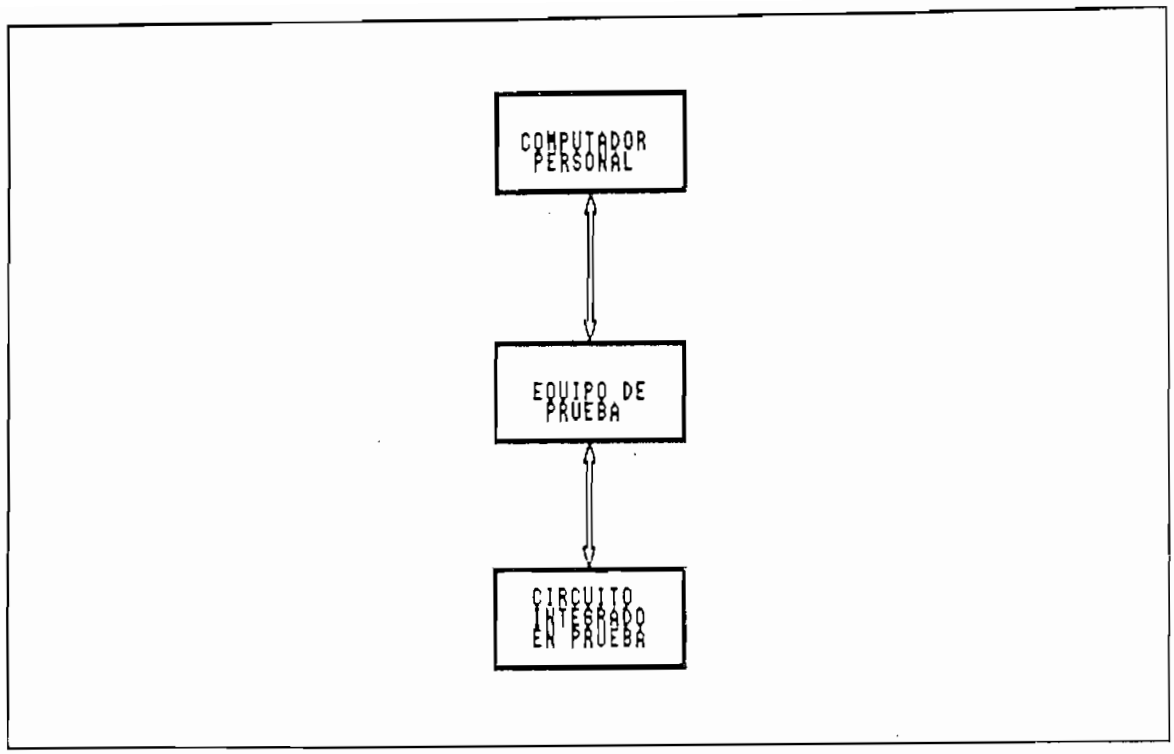

DIAGRAMA 2.1

Para aumentar un circuito integrado a la librería, se debe seguir un procedimiento en otro programa del PC, el CILIB.EXE, en donde se ingresan datos como serie, descripción, número de pines, entradas y salidas y la tabla de verdad.

El microcontrolador del equipo de prueba no es "inteligente", en otras palabras, no analiza información acerca del circuito integrado en prueba limitándose a enrutar los datos recibidos por el puerto serie a los puertos paralelos, y viceversa.

El equipo se basa en un microcontrolador 8751 para la recepción y enrutamiento de datos. Un arreglo de flip-flops e interruptores analógicos definen y mantienen el sentido de cada línea, sea como entrada de datos, polarización al chip o recepción de datos desde el circuito integrado.

Para convertir los niveles lógicos a diferentes voltajes según se requiera, se incorporó un arreglo de transistores bipolares de juntura que trabajan en corte o saturación dependiendo del estado lógico requerido. Estos transistores son controlados por circuitos integrados del tipo MS1488 que elevan el voltaje aplicado a la base de cada transistor al nivel necesario para asegurar su corte y saturación. Así mismo, dependiendo del signo de polarización que deberá tomar cada pin del zócalo de pruebas, se implementó en el arreglo dos tipos de transistores, NPN o PNP.

La fuente de poder es regulada y variable, estando controlada digitalmente por el microcontrolador. La fuente de 5V es independiente, mientras que las variables positiva y negativa comparten el transformador de tap central y el sistema de control. Los reguladores de voltaje y sistema de filtrado son, por supuesto, independientes para cada uno.

El programa de pruebas dispone de una opción de diagnóstico en donde se comprueba el voltaje existente en cada pin del zócalo para verificar el correcto funcionamiento del equipo.

El diagrama de bloques 2.2 muestra más detalladamente el funcionamiento del probador de circuitos integrados.

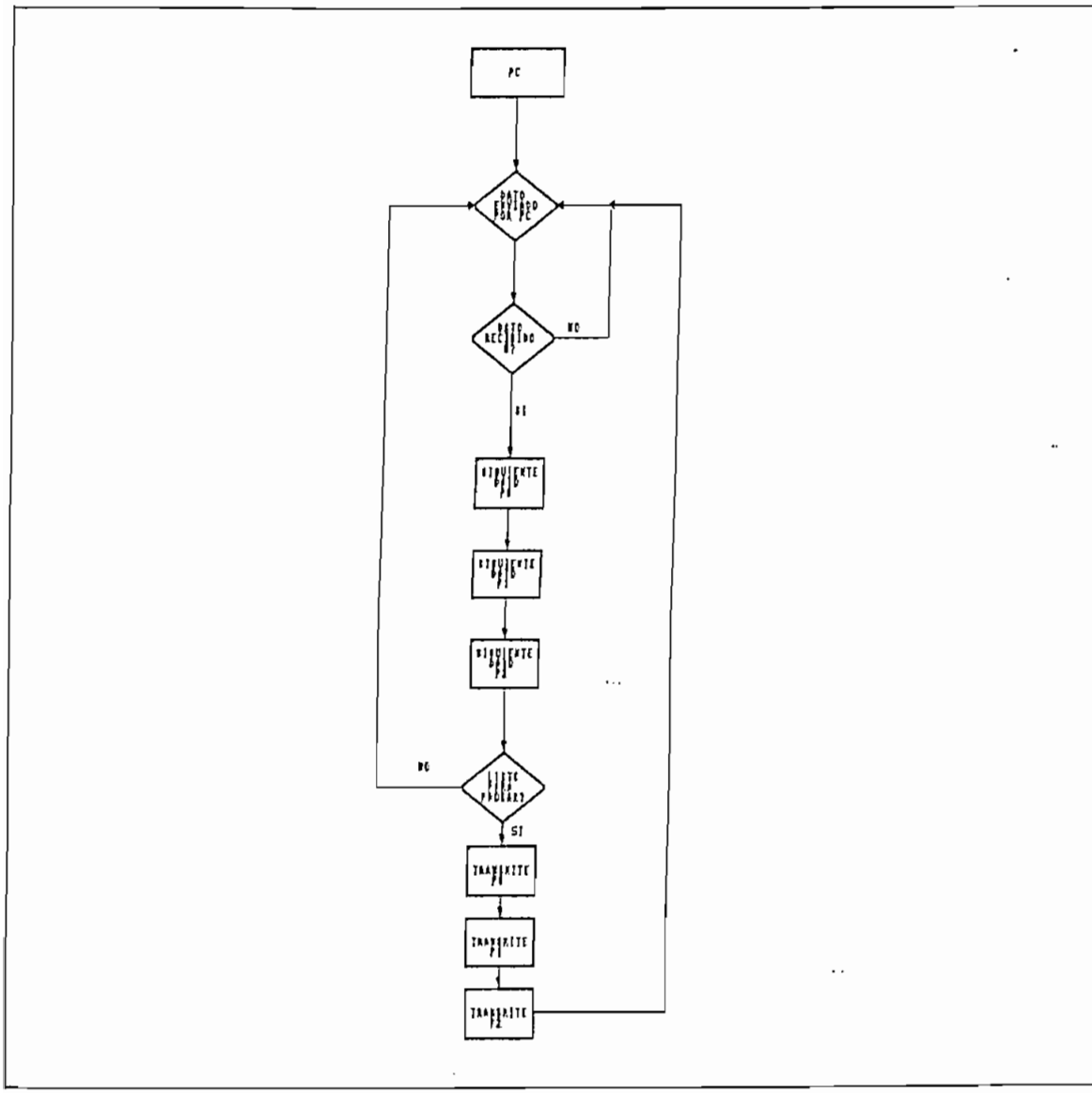

DIAGRAMA 2.2

CAPITULO III

HARDWARE

3.1.- DISEÑO DE LAS ETAPAS CIRCUITALES

Siendo el probador de circuitos integrados un equipo que maneja datos por el puerto serie y paralelo, así como operaciones lógicas, se requiere de un microcontrolador que sea capaz de realizar estas tareas en forma sencilla. El microcontrolador, el mismo que permitirá controlar por software la fuente regulada, enviar datos al chip en prueba y leer los resultados obtenidos, enviándolos luego mediante puerto serie al computador personal, será el INTEL 8751. Se seleccionó este microcontrolador por habérselo estudiado y utilizado en laboratorio ampliamente. Dispone de EPROM interna con una capacidad de 4Kbytes, suficiente para la presente aplicación. Además, esta característica simplifica en gran medida la circuitería del equipo pues no se requiere de memoria externa, latches, etc. El microcontrolador opera con un cristal externo de 7.3728 Mhz.

Al no utilizar memoria externa para el programa se dispone de todos los puertos para entrada/salida de datos y señales de control, eliminando la necesidad de ampliar los puertos.

Los puertos paralelos 0, 1 y 2 serán utilizados para control de las fuentes reguladas, de los flip-flops y envío y recepción de los datos de prueba y polarización. El puerto 3 se utiliza únicamente para la comunicación serial con el computador personal, específicamente se ocupan las líneas RX y TX solamente. El cuadro 3.1 muestra la función de cada pin de los puertos paralelos del microcontrolador:

P0.	0	1	2	3	4	5	6	7
PIN	39	38	37	36	35	34	33	32
USO	CLK	1L	V.R.	VSS	Z(20)	Z(19)	Z(18)	Z(17)
P1.	0	1	2	3	4	5	6	7
PIN	1	2	3	4	5	6	7	8
USO	Z(3)	Z(4)	Z(5)	Z(6)	Z(7)	Z(8)	Z(9)	Z(10)
P2.	0	1	2	3	4	5	6	7
PIN	21	22	23	24	25	26	27	28
USO	Z(2)	Z(1)	Z(11)	Z(12)	Z(13)	Z(14)	Z(15)	Z(16)
P3.	0	1	2	3	4	5	6	7
PIN	10	11	12	13	14	15	16	17
USO	RXD	TXD						

CUADRO 3.1.- Asignación de pines de puertos paralelos del uC

Para el microcontrolador se incorporó un circuito de reset basado en una red R-C. Para su diseño se tomó en cuenta el tiempo de carga del capacitor de esta red.

El tiempo de carga del capacitor es de:

$$t \approx RC$$

$$t \approx 8.2K \times 10\ uF \approx 82\ ms$$

La figura 3.1 muestra las conexiones del microcontrolador 8751.

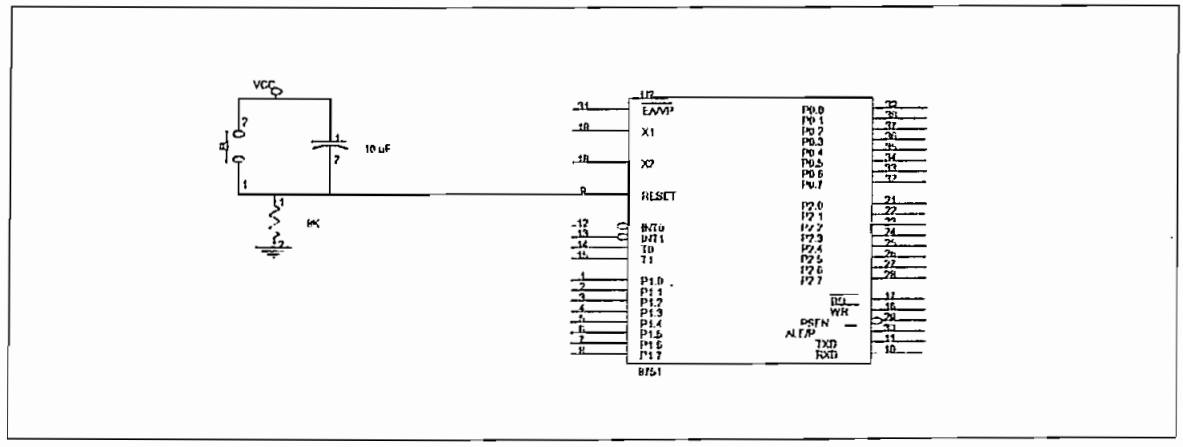

FIGURA 3.1

A continuación se procederá al diseño de los demás módulos mencionados anteriormente.

3.1.1.- INTERFAZ PARA COMUNICACION SERIAL ENTRE EL COMPUTADOR Y EL EQUIPO DE PRUEBA .

Como se puede notar en el diagrama de bloques del equipo, una de las etapas a considerarse es la comunicación entre el PC y el microcontrolador del equipo de pruebas.

Para mantener la circuitería lo más sencilla posible, así como ahorrar espacio al equipo, se utilizó un MC1488 y un MC1489. Con ésto se consiguió acoplar las señales de +12V y -12V del computador personal a señales TTL utilizadas por el microcontrolador. Como el MC1488 y el MC1489 son diseñados específicamente para este propósito, no se requiere de elementos adicionales para su operación.

3.1.2.- REGULADORES DE VOLTAJE

El equipo de pruebas requiere de fuentes reguladas para los siguientes propósitos:

- Polarizar el microcontrolador 8751 así como el resto de circuitos integrados utilizados en el equipo.
- Controlar el estado de conducción de los transistores de salida.
- Polarizar el circuito integrado en prueba.

El numeral 2.2 menciona los voltajes requeridos de las fuentes.

Como se mencionó anteriormente, la fuente de +5V es independiente de las demás. De hecho, utiliza un bobinado separado del transformador de entrada. El transformador utilizado dispone de varios bobinados, uno de ellos con tap central. Para la fuente de +5V se utilizó el bobinado de 110Vac a 9.45Vac, mientras que para las fuentes variables positiva y negativa se utilizó el de 15Vac de salida con tap central (figura 3.2 y 3.3). Todas las fuentes están conectadas a la misma referencia de tierra.

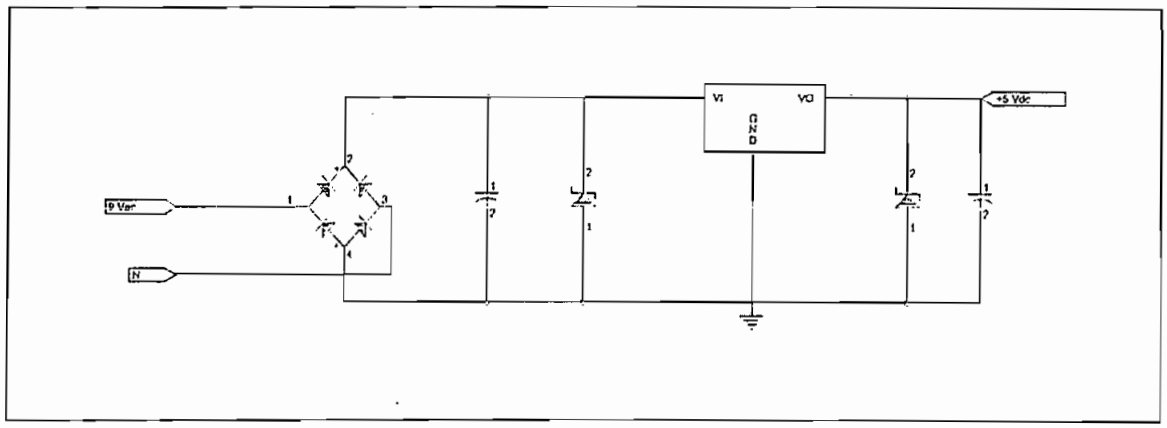

FIGURA 3.2

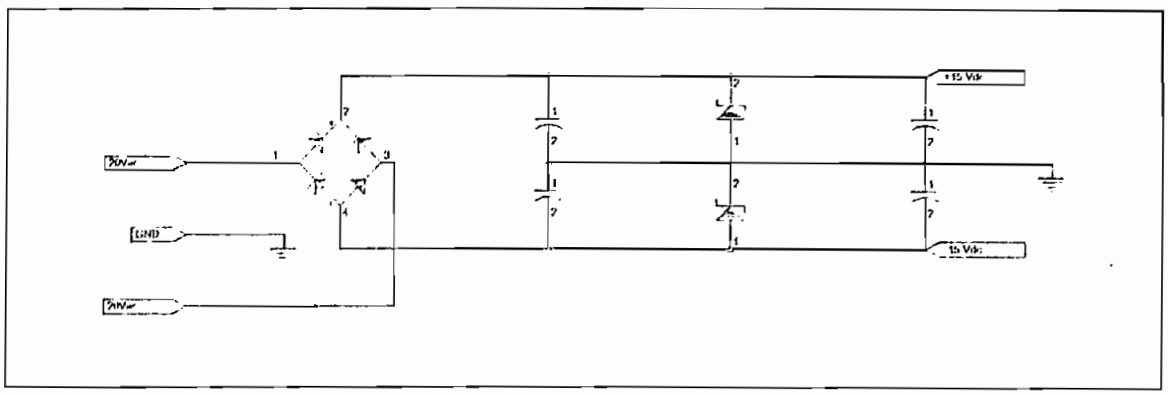

FIGURA 3.3

Para las fuentes variables se utilizaron reguladores de voltaje programables. Las señales de control provienen del microcontrolador y escogen entre +5V, -5V y +9V, -9V. La figura 3.4 muestra el esquemático correspondiente.

Como se puede ver, las compuertas utilizadas son del tipo "open collector"; cuando se aplica un alto en la entrada de las mismas se obtiene un bajo a la salida, en cambio cuando la entrada esta en bajo, la salida se acoplará al voltaje que se tenga en el terminal ADJ del regulador.

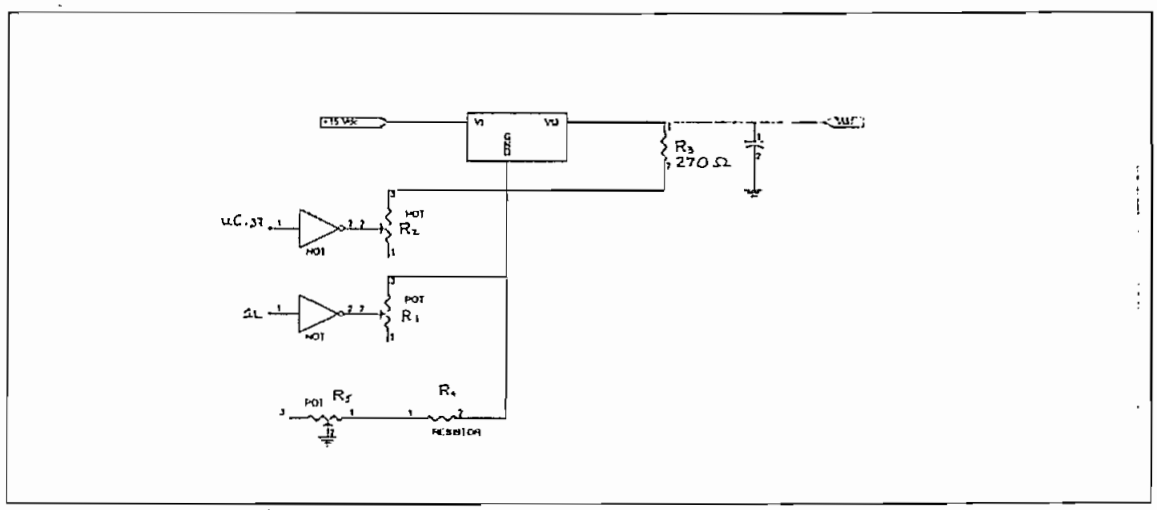

FIGURA 3.4

A continuación se incluyen los cálculos necesarios para diseñar la fuente a los voltajes requeridos.

Para la fuente positiva tenemos que:

$Vout = 1.25\ (1 + R_a/R_3)$

Si $R_3 = 270\ \Omega$ y $R_a = 2\ K\Omega$

Donde $R_a = R_4 + R_5$

Para Vout = 9 V tenemos que,

$R_a' = 1674\ \Omega$, donde $R_a' = R_1//R_a$

$1674 = (R_1.R_a)/(R_1+R_a)$

Así, $R_1 = 10.3\ K\Omega$

Para Vout = 5 V tenemos que,

$R_a'' = 810\ \Omega = R_1//R_a// R_2$

Así, $R_2 = 1568\ \Omega$

La fuente negativa es de características idénticas pero inversas a la positiva, por lo tanto los cálculos anteriores son también válidos para esta fuente. Sin embargo, se debe notar que la distribución de pines del regulador de voltaje negativo difiere del positivo.

3.1.3.- CONTROL DE LINEAS DE DATOS - CONVERSION DE NIVELES LOGICOS

Debido a que se necesita manejar niveles lógicos diferentes en la circuitería del probador y en el zócalo de pruebas, se requiere de un convertidor bidireccional de niveles lógicos. Bidireccional, porque

una determinada línea puede ser entrada para un tipo de circuito integrado o salida para otro. Comercialmente no hay un dispositivo semejante, por lo que se debe acoplar dos convertidores (de sentido opuesto) con algún sistema de control que solamente habilite uno de los dos a la vez de acuerdo a la necesidad. Puesto que se esta en capacidad de probar circuitos integrados de hasta 20 pines, se requerirá de 20 líneas independientes. Cada línea deberá tener su propio sistema de control para sus dos convertidores opuestos. Para solucionar el problema con la menor cantidad de líneas de control se optó por utilizar un flip-flop y un interruptor analógico por cada línea. Todos los flip-flops tienen sus entradas de reloj controladas por el mismo terminal del microcontrolador, definiéndose en el mismo instante el sentido de todas las líneas que van al chip en prueba. Dependiendo del estado del terminal Q de los flip-flops se habilitará o no el interruptor analógico, quedando la línea como lectura o escritura de datos respectivamente, como se puede ver en la figura 3.5.

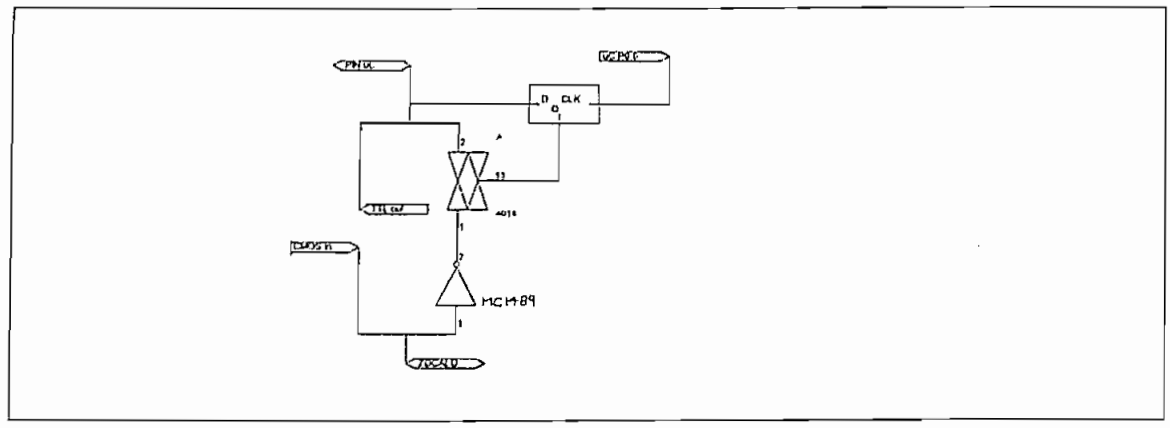

FIGURA 3.5

El procedimiento de selección de entrada o salida es el siguiente:

- Primeramente se fija el terminal D del flip-flop a 0L para inhabilitar el interruptor analógico y poder escribir al zócalo, o 1L para habilitar dicho interruptor y leer del zócalo. Debido a la forma en que estan conectados los transistores de salida, no es necesario tener un interruptor analógico en el convertidor TTL -> CMOS.
- Seguidamente se aplica un flanco positivo al terminal CLK del flip-flop para mover el dato desde D hasta Q, con lo que queda seleccionada la función. Esta selección no variará mientras no se aplique otro flanco positivo al CLK, pudiéndose utilizar el mismo terminal del microcontrolador para el resto de pruebas.

Para la conversión de niveles TTL a CMOS se utiliza un transistor bipolar de juntura por línea. El motivo de usar transistores es para dar a las salidas del zócalo la suficiente capacidad de corriente en caso tengan que cumplir las funciones de fuente de polarización. El estado de conducción de este transistor es controlado por una señal del MC1488 que a su vez convierte el nivel TTL de +5V / GND a -15V / +15V, con lo que se asegura el corte y saturación del transistor.

Debido a que los pines del zócalo necesitan en ocasiones polarizar positivamente y en ocasiones negativamente, se optó por utilizar transistores del tipo NPN o del tipo PNP, según el caso.

La figura 3.6 muestra las dos situaciones. Como se puede notar siempre va a haber un estado lógico presente, pero debido a la resistencia implementada, es posible para el circuito integrado sobrescribir un estado lógico sin afectar su funcionamiento.

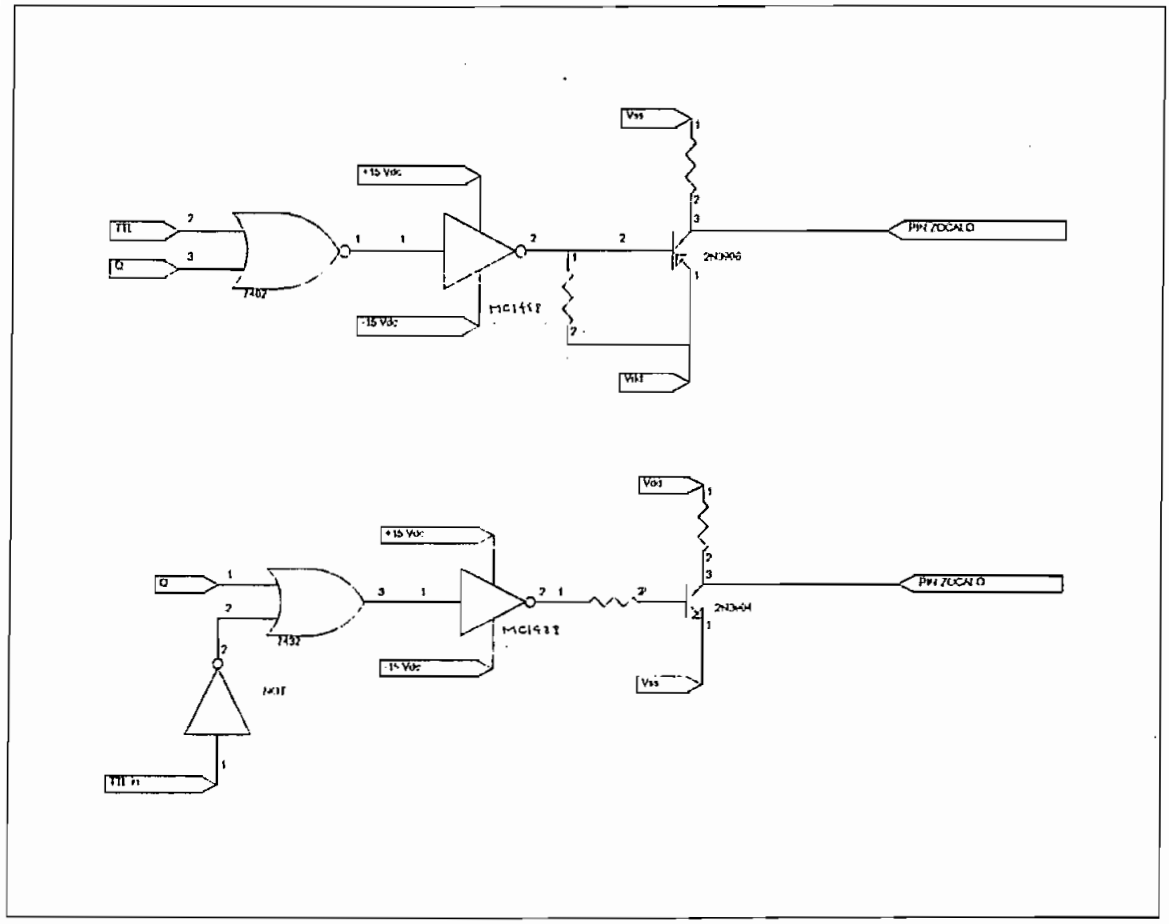

FIGURA 3.6

Para la conversión de CMOS a TTL se utilizó un arreglo de circuitos integrados MC1489, seguidos por el interruptor analógico mencionado anteriormente. La figura 3.5 muestra como se solucionó este problema.

El zócalo utilizado para las pruebas es único para todos los tipos de circuitos integrados y tiene 20 pines. El pin 10 del zócalo es común, habiendo que alinear el pin inferior derecho de los circuitos integrados a probarse con este terminal del zócalo. Por ejemplo, para un circuito integrado de 14 pines, el pin 7 se alineará con el pin 10 del zócalo antes de insertarlo.

3.2.- IMPLEMENTACION DEL CIRCUITO IMPRESO

Debido a que el diseño se orientó hacia un equipo fácil de transportar y usar, se vio la necesidad de mantener el tamaño lo más pequeño posible. La técnica de "wire-wrapping" resultó más práctica y a su vez más rápida de implementar que el uso de circuitos impresos. Estos últimos son más costosos y generalmente ocupan más espacio que una placa en "wire-wrapping".

En el presente caso se separó el circuito en dos placas independientes. Una correspondiente a la fuente de poder y otra para el circuito de prucbas.

El equipo requiere de solamente un interruptor de encendido y un pulsador de "reset". Además dispone de un LED para indicar que el equipo se encuentra encendido.

CAPITULO IV

SOFTWARE

4.1.- CONDICIONES A CUMPLIRSE

Como se mencionó anteriormente, el probador de circuitos integrados requiere de dos programas para el PC y uno para el microcontrolador. Los programas para el computador personal han sido creados en QBASIC, y el del microcontrolador en Assembler.

La librería que contiene la información de los circuitos integrados a probarse está dividida en campos, ocupando cada chip un registro del archivo de datos "CI.LIB" y cada pieza de información un campo de dicho archivo. El archivo de datos es secuencial, por lo que se requiere que la información sea grabada siempre en el mismo orden.

El programa de actualización de circuitos integrados, CILIB.EXE, permite realizar los siguientes procedimientos:

- Ingresar la numeración del nuevo circuito integrado, luego de lo cual el programa busca automáticamente en el archivo "CI.LIB" antes de proseguir con la actualización.

- Se guarda la siguiente información para cada circuito integrado: serie, descripción, pines que son entrada o salida, tabla de verdad.

El programa de prueba, CMOS.EXE, utiliza la información del archivo CI.LIB y la envía al equipo probador mediante el puerto serie. Este programa permite las siguientes opciones:

- Escoger el tipo de circuito integrado, ingresando únicamente la serie correspondiente.
- Escoger el voltaje de polarización deseado para el circuito integrado bajo prueba. Esta opción se presenta luego de la primera prueba, que se realiza siempre con +5V y GND. La razón para ésto es evitar la presencia de corrientes altas en caso que se polarice con otros voltajes y el circuito integrado tenga algún terminal cortocircuitado a Vdd o Vss.
- El equipo de prueba dispone, además, de una subrutina de verificación de funcionamiento para comprobar el estado del mismo. Todo lo que se requiere es de un voltímetro para verificar que los niveles presentes sean los indicados por el computador.

El programa del microcontrolador no realiza análisis alguno, pues se limita a recibir los datos por el puerto serie, enrutándolos hacia los puertos paralelos y viceversa. Con esto se evita tener que actualizar el programa del microcontrolador al aumentar el número de circuitos integrados a probarse.

Los diagramas siguientes muestran el funcionamiento en bloques de los diferentes programas. Como se puede ver, se mantuvo las pantallas y la utilización del programa lo más sencillos posible.

4.2.- DIAGRAMAS DE FLUJO

El diagrama 4.1 muestra el proceso en bloques que sigue el progama de actualización de librerías cuando se desea ingresar otro circuito integrado al banco de datos existente.

Primeramente, pide la serie del circuito integrado y busca dicha serie entre las existentes en la librería CI.LIB. Si no la encuentra prosigue con el ingreso de información, como es la descripción del chip, pines de entrada y salida y tabla de verdad. Si por otro lado, encuentra la serie en la librería, avisa al usuario que dicha serie ya fue ingresada con anterioridad.

Finalmente, luego de ingresar los datos correspondientes da la opción de grabarlos, desecharlos o iniciar otra vez.

El manual de instrucciones del equipo muestra con mayor detalle el manejo del probador y el uso del software.

El diagrama 4.2 se refiere al programa de pruebas propiamente. El proceso seguido por el software es esperar hasta que se ingrese una serie, luego la busca en CI.LIB y si la encuentra prosigue con el envió y recepción de datos. Si no lo encuentra pide escoger entre salir del programa o ingresar otra serie.

De la misma manera el programa permite continuar con otra polarización luego de la primera prueba, siempre y cuando haya sido satisfactoria. De lo contrario va a una pantalla de selección entre iniciar con otro circuito integrado o abandonar el programa.

El tercer programa implementado esta escrito en lenguaje Assembler y contiene las instrucciones para el microcontrolador 8751. Como se puede notar en el diagrama 4.3, el 8751 se limita a manejar los puertos serie y paralelo con la información recibida del computador personal.

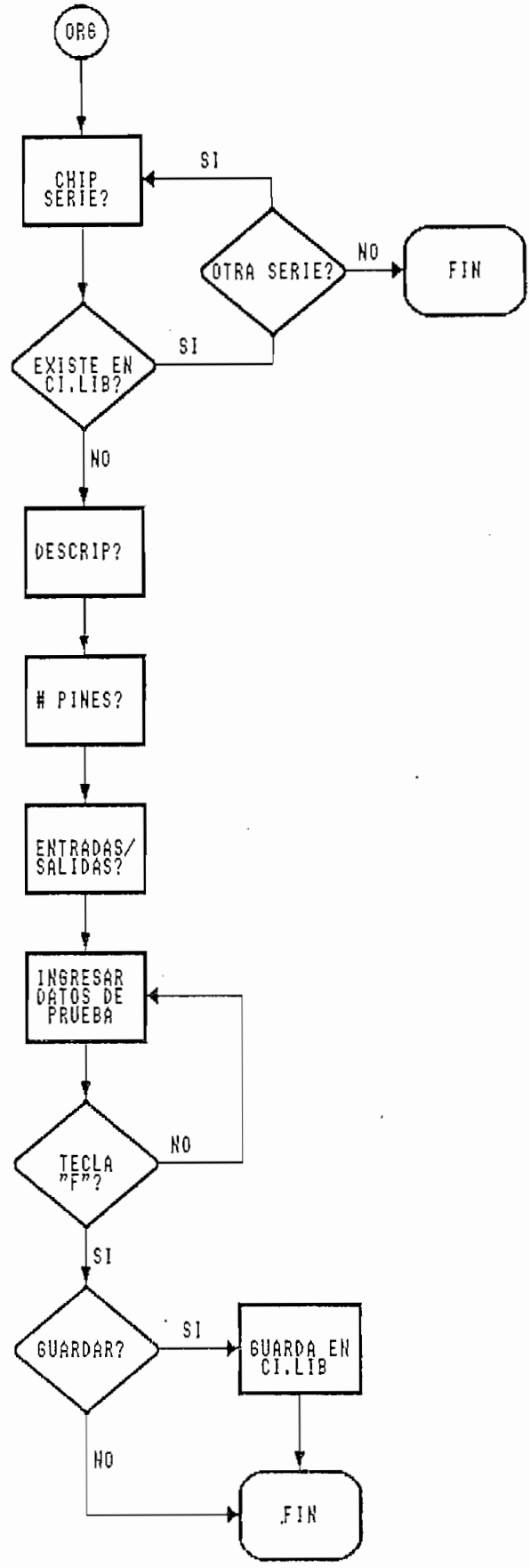

DIAGRAMA 4.1.- Programa de actualización del archivo CI.LIB

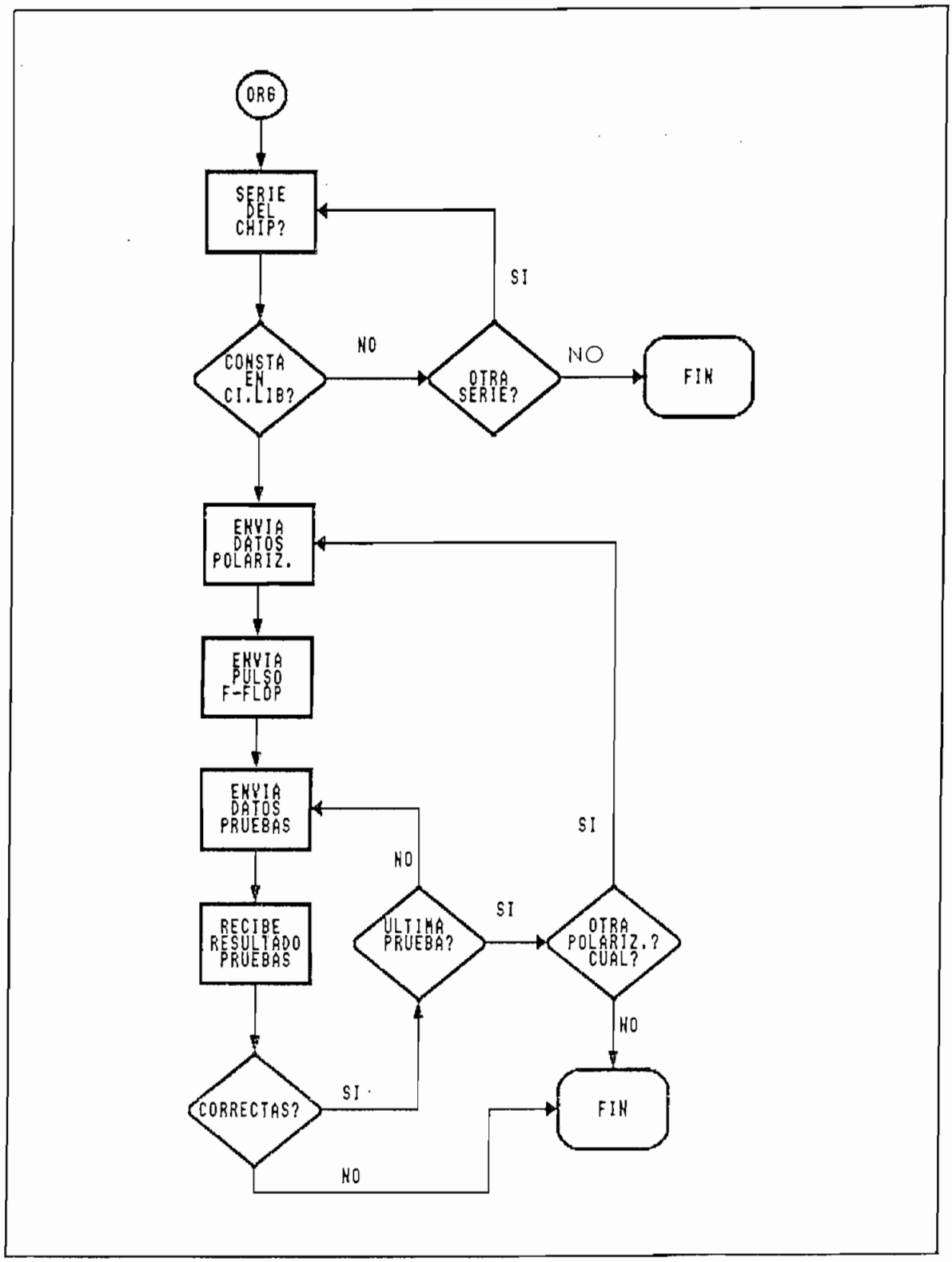

DIAGRAMA 4.2.- Programa de pruebas CMOS.EXE

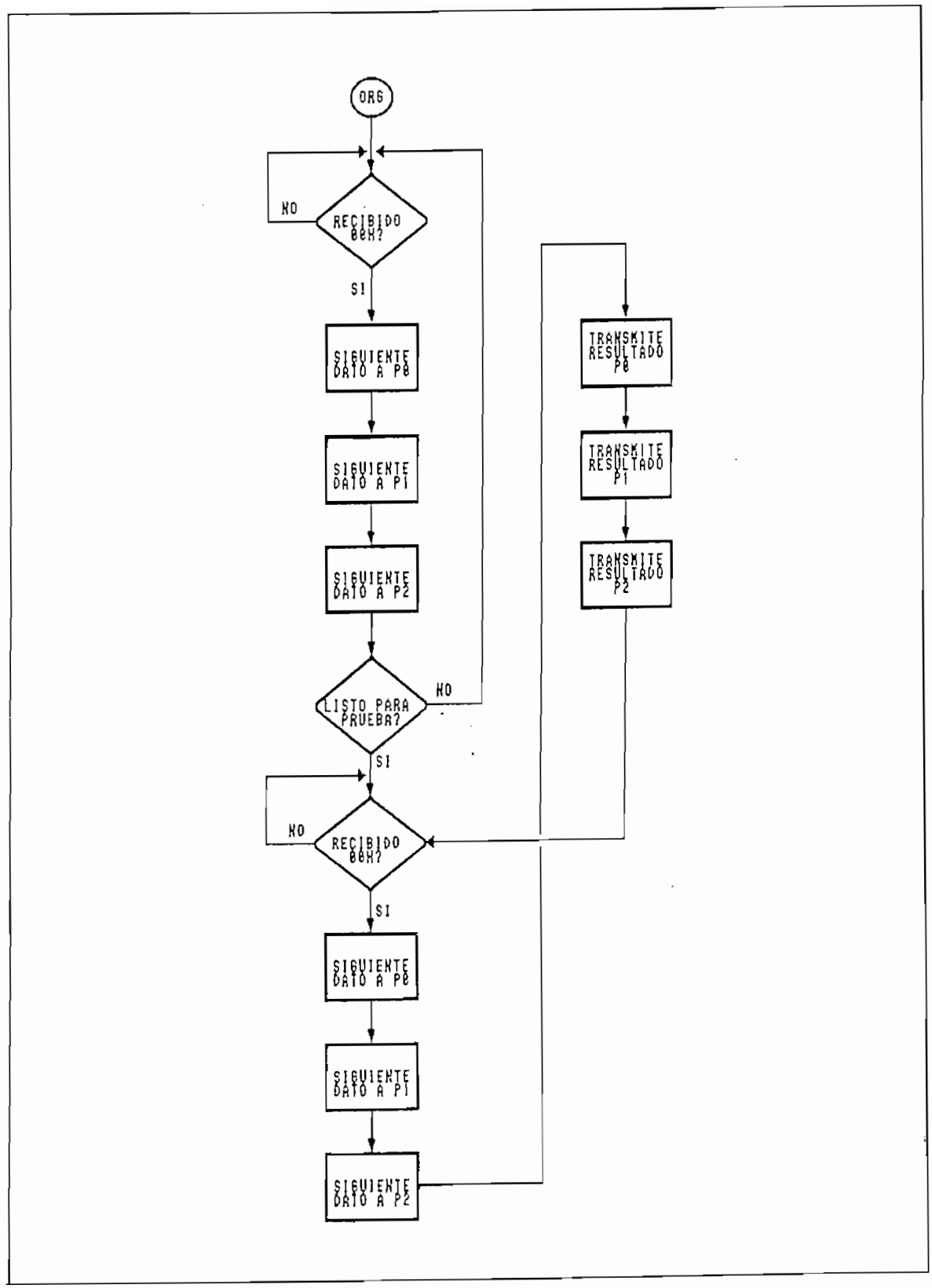

DIAGRAMA 4.3.- Programa en Assembler del 8751

4.3.- PROGRAMAS EN LENGUAJE DE ALTO NIVEL

4.3.1.- PROGRAMA DE AMPLIACION DE LA LIBRERIA CJ.LIB

```
DECLARE SUB MAIN ()
DECLARE SUB ERRORES ()
DECLARE SUB PANTALLA ()
DECLARE SUB PORTPRUEB ()
DECLARE SUB PORT16 ()
DECLARE SUB PORT18 ()
DECLARE SUB PORT20 ()
DECLARE SUB PRUEBAS16 ()
DECLARE SUB PRUEBAS18 ()
DECLARE SUB PRUEBAS20 ()
DECLARE SUB FIN ()
DECLARE SUB PRUEBAS14 ()
DECLARE SUB PORT14 ()
DECLARE SUB DATOS ()
DECLARE SUB ZOCALO20 ()
DECLARE SUB ZOCALO18 ()
DECLARE SUB ZOCALO16 ()
DECLARE SUB ZOCALO14 ()
DECLARE SUB CUAD3 (Y1!, X1!, Y2!, X2!, M1!, M2!, M3!, M4!, M5!, M6!, A!, B!)
COMMON PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
COMMON ZOC1, ZOC2, ZOC3, ZOC4, ZOC5, ZOC6, ZOC7, ZOC8, ZOC9, ZOC10, ZOC11, ZOC12,
ZOC13, ZOC14, ZOC15, ZOC16, ZOC17, ZOC18, ZOC19, ZOC20
COMMON POL1, POL2, POL3, POL4, POL5, POL6, POL7, POL8, POL9, POL10, POL11, POL12,
POL13, POL14, POL15, POL16, POL17, POL18, POL19, POL20
COMMON P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20
COMMON Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11, Z12, Z13, Z14, Z15, Z16, Z17, Z18, Z19, Z20
COMMON SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
COMMON P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
COMMON PINES$
COMMON ER, ERROR$
CLS
CLEAR
CALL MAIN
END

SUB CUAD3 (Y1, X1, Y2, X2, M1, M2, M3, M4, M5, M6, A, B)
H = X2 - X1
V = Y2 - Y1
COLOR A, B
LOCATE Y1, X1
FOR Z = 1 TO (X2 - X1)
LOCATE Y1, X1 + Z
PRINT CHR$(M1);
NEXT Z
LOCATE Y2, X1
FOR Z = 1 TO H
PRINT CHR$(M1);
```

```
NEXT Z
FOR Z = 1 TO (V - 1)
LOCATE Y1 + Z, X1
PRINT CHR$(M2);
LOCATE Y1 + Z, X2
PRINT CHR$(M2)
NEXT Z
LOCATE Y1, X1
PRINT CHR$(M3)
LOCATE Y2, X2
PRINT CHR$(M4)
LOCATE Y1, X1 + H
PRINT CHR$(M5)
LOCATE Y1 + V, X1
PRINT CHR$(M6)
END SUB

SUB DATOS
SHARED ZOC1, ZOC2, ZOC3, ZOC4, ZOC5, ZOC6, ZOC7, ZOC8, ZOC9, ZOC10, ZOC11, ZOC12,
ZOC13, ZOC14, ZOC15, ZOC16, ZOC17, ZOC18, ZOC19, ZOC20
SHARED POL1, POL2, POL3, POL4, POL5, POL6, POL7, POL8, POL9, POL10, POL11, POL12, POL13,
POL14, POL15, POL16, POL17, POL18, POL19, POL20
SHARED ZOCP0, ZOCP1, ZOCP2
SHARED POLP0, POLP1, POLP2
SHARED PINES$
ZOCP0 = 2 + ZOC17 + ZOC18 + ZOC19 + ZOC20
ZOCP1 = ZOC3 + ZOC4 + ZOC5 + ZOC6 + ZOC7 + ZOC8 + ZOC9 + ZOC10
ZOCP2 = ZOC1 + ZOC2 + ZOC11 + ZOC12 + ZOC13 + ZOC14 + ZOC15 + ZOC16
POLP0 = 2 + POL17 + POL18 + POL19 + POL20
POLP1 = POL3 + POL4 + POL5 + POL6 + POL7 + POL8 + POL9 + POL10
POLP2 = POL1 + POL2 + POL11 + POL12 + POL13 + POL14 + POL15 + POL16
IF PINES$ = "14" THEN CALL PORTPRUEB
IF PINES$ = "16" THEN CALL PORTPRUEB
IF PINES$ = "18" THEN CALL PORTPRUEB
IF PINES$ = "20" THEN CALL PORTPRUEB
END SUB

SUB ERRORES
SHARED ER, ERROR$
IF UCASE$(ERROR$) <> "E" AND UCASE$(ERROR$) <> "S" AND UCASE$(ERROR$) <> "+" AND
UCASE$(ERROR$) <> "-" THEN ER = 1
END SUB

SUB FIN
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
GUARDAR: LOCATE 1, 1
FOR M = 1 TO 24
    COLOR 0, 1
    PRINT "                                                    "
NEXT M
CALL CUAD3(8, 15, 14, 65, 196, 179, 218, 217, 191, 192, 11, 0)
```

```
COLOR 11, 0
LOCATE 9, 16
PRINT "                                    "
LOCATE 10, 16
PRINT "    DESEA GUARDAR ESTE CIRCUITO INTEGRADO    "
LOCATE 11, 16
PRINT "                                    "
LOCATE 12, 16
PRINT "    EN LA LIBRERIA DEL PROBADOR DE CIs?    "
LOCATE 13, 16
PRINT "                                    "
2000 LOCATE 22, 13
COLOR 4, 0
PRINT "                          "
LOCATE 22, 13
COLOR 4, 0
INPUT "Desea guardar y salir (S/N)"; COR$
IF UCASE$(COR$) = "N" THEN GOTO 3000
WRITE #1, SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, _
P10P2, P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2, P15P0,
P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
CLOSE #1
LOCATE 22, 13
COLOR 4, 0
INPUT "Desea iniciar otra vez (S/N)"; COR3$
IF UCASE$(COR3$) = "S" THEN CALL MAIN
3000 LOCATE 22, 13
COLOR 11, 0
PRINT "                          "
LOCATE 22, 13
COLOR 4, 0
INPUT "Desea salir sin guardar (S/N)"; COR$
IF UCASE$(COR$) = "S" THEN GOTO FIN1
GOTO 2000
FIN1: COLOR 7, 0
CLS
END
END SUB

SUB MAIN
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED ZOC1, ZOC2, ZOC3, ZOC4, ZOC5, ZOC6, ZOC7, ZOC8, ZOC9, ZOC10, ZOC11, ZOC12,
ZOC13, ZOC14, ZOC15, ZOC16, ZOC17, ZOC18, ZOC19, ZOC20
SHARED POL1, POL2, POL3, POL4, POL5, POL6, POL7, POL8, POL9, POL10, POL11, POL12, POL13,
POL14, POL15, POL16, POL17, POL18, POL19, POL20
SHARED P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20
SHARED Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11, Z12, Z13, Z14, Z15, Z16, Z17, Z18, Z19, Z20
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
SHARED PINES$
```

```
SHARED ER, ERROR$
CLS
LOCATE 1, 1
FOR M = 1 TO 24
      COLOR 0, 1
      PRINT "                                                    "
NEXT M
CALL CUAD3(8, 15, 14, 65, 196, 179, 218, 217, 191, 192, 11, 0)
COLOR 11, 0
LOCATE 9, 16
PRINT "                                    "
LOCATE 10, 16
PRINT "  IMPLEMENTACION DE NUEVOS CIRCUITOS INTEGRADOS  "
LOCATE 11, 16
PRINT "                                    "
LOCATE 12, 16
PRINT "          PARA EL PROBADOR DE CIs           "
LOCATE 13, 16
PRINT "                                    "
LOCATE 23, 40
COLOR 7, 0
PRINT "(Presione cualquier tecla)"
WHILE INKEY$ = ""
WEND
29 LOCATE 22, 13
COLOR 0, 1
PRINT "                                  "
LOCATE 23, 40
PRINT "                              "
COLOR 11, 0
LOCATE 10, 16
PRINT "          INGRESE LA NUMERACION DEL          "
LOCATE 12, 16
PRINT "          CIRCUITO INTEGRADO: CI #           "
LOCATE 12, 52
INPUT " ", AB$
OPEN "A:\CI.LIB" FOR INPUT AS #1
DO UNTIL EOF(1)
INPUT #1, SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, _
P10P2, P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2, P15P0,
P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
IF AB$ = SERIE$ THEN GOTO CIEXIST
LOOP
P1P0 = 0: P1P1 = 0: P1P2 = 0: P2P0 = 0: P2P1 = 0: P2P2 = 0: P3P0 = 0: P3P1 = 0: P3P2 = 0: P4P0 = 0:
P4P1 = 0: P4P2 = 0: P5P0 = 0: P5P1 = 0: P5P2 = 0: P6P0 = 0: P6P1 = 0: P6P2 = 0: P7P0 = 0: P7P1 = 0:
P7P2 = 0: P8P0 = 0: P8P1 = 0: P8P2 = 0: P9P0 = 0 _
: P9P1 = 0: P9P2 = 0: P10P0 = 0: P10P1 = 0: P10P2 = 0: P11P0 = 0: P11P1 = 0: P11P2 = 0: P12P0 = 0:
P12P1 = 0: P12P2 = 0: P13P0 = 0: P13P1 = 0: P13P2 = 0: P14P0 = 0: P14P1 = 0: P14P2 = 0: P15P0 = 0:
P15P1 = 0: P15P2 = 0: P16P0 = 0: P16P1 = 0: P16P2 _
= 0: P17P0 = 0: P17P1 = 0: P17P2 = 0: P18P0 = 0: P18P1 = 0: P18P2 = 0
SERIE$ = AB$
CLOSE #1
OPEN "A:\CI.LIB" FOR APPEND AS #1
2 LOCATE 22, 13
```

```
COLOR 0, 1
PRINT "                                    "
LOCATE 23, 40
COLOR 0, 1
PRINT "                                    "
3 COLOR 11, 0
LOCATE 10, 16
PRINT "       INGRESE LA DESCRIPCION DEL CI:          "
LOCATE 12, 16
PRINT "                                          "
LOCATE 12, 27
INPUT " ", DESCRIP$
28 LOCATE 23, 40
COLOR 0, 1
PRINT "                          "
31 COLOR 11, 0
LOCATE 10, 16
PRINT "       INGRESE EL NUMERO DE PINES DEL          "
LOCATE 12, 16
PRINT "       CIRCUITO INTEGRADO:  # PINES =          "
LOCATE 12, 56
INPUT " ", PINES$
SELECT CASE PINES$
      CASE "14"
            GOTO PIN14
      CASE "16"
            GOTO PIN16
      CASE "18"
            GOTO PIN18
      CASE "20"
            GOTO PIN20
      CASE ELSE
            GOTO 31
END SELECT
PIN14: CALL PANTALLA
60 LOCATE 22, 13
COLOR 11, 0
PRINT "                          "
LOCATE 12, 21
PRINT "            "
LOCATE 12, 21
ER = 0
PIN1$ = INPUT$(1)
PRINT PIN1$
ERROR$ = PIN1$
CALL ERRORES
IF ER = 1 GOTO 60
61 LOCATE 13, 21
PRINT "            "
LOCATE 13, 21
ER = 0
PIN2$ = INPUT$(1)
PRINT PIN2$
ERROR$ = PIN2$
CALL ERRORES
IF ER = 1 GOTO 61
```

```
62 LOCATE 14, 21
PRINT "          "
LOCATE 14, 21
ER = 0
PIN3$ = INPUT$(1)
PRINT PIN3$
ERROR$ = PIN3$
CALL ERRORES
IF ER = 1 GOTO 62
63 LOCATE 15, 21
PRINT "          "
LOCATE 15, 21
ER = 0
PIN4$ = INPUT$(1)
PRINT PIN4$
ERROR$ = PIN4$
CALL ERRORES
IF ER = 1 GOTO 63
64 LOCATE 16, 21
PRINT "          "
LOCATE 16, 21
ER = 0
PIN5$ = INPUT$(1)
PRINT PIN5$
ERROR$ = PIN5$
CALL ERRORES
IF ER = 1 GOTO 64
65 LOCATE 17, 21
PRINT "          "
LOCATE 17, 21
ER = 0
PIN6$ = INPUT$(1)
PRINT PIN6$
ERROR$ = PIN6$
CALL ERRORES
IF ER = 1 GOTO 65
66 LOCATE 18, 21
PRINT "          "
LOCATE 18, 21
ER = 0
PIN7$ = INPUT$(1)
PRINT PIN7$
ERROR$ = PIN7$
CALL ERRORES
IF ER = 1 GOTO 66
67 LOCATE 12, 50
PRINT "          "
LOCATE 12, 50
ER = 0
PIN8$ = INPUT$(1)
PRINT PIN8$
ERROR$ = PIN8$
CALL ERRORES
IF ER = 1 GOTO 67
68 LOCATE 13, 50
PRINT "          "
```

```
LOCATE 13, 50
ER = 0
PIN9$ = INPUT$(1)
PRINT PIN9$
ERROR$ = PIN9$
CALL ERRORES
IF ER = 1 GOTO 68
69 LOCATE 14, 50
PRINT "          "
LOCATE 14, 50
ER = 0
PIN10$ = INPUT$(1)
PRINT PIN10$
ERROR$ = PIN10$
CALL ERRORES
IF ER = 1 GOTO 69
70 LOCATE 15, 50
PRINT "          "
LOCATE 15, 50
ER = 0
PIN11$ = INPUT$(1)
PRINT PIN11$
ERROR$ = PIN11$
CALL ERRORES
IF ER = 1 GOTO 70
71 LOCATE 16, 50
PRINT "          "
LOCATE 16, 50
ER = 0
PIN12$ = INPUT$(1)
PRINT PIN12$
ERROR$ = PIN12$
CALL ERRORES
IF ER = 1 GOTO 71
72 LOCATE 17, 50
PRINT "          "
LOCATE 17, 50
ER = 0
PIN13$ = INPUT$(1)
PRINT PIN13$
ERROR$ = PIN13$
CALL ERRORES
IF ER = 1 GOTO 72
73 LOCATE 18, 50
PRINT "          "
LOCATE 18, 50
ER = 0
PIN14$ = INPUT$(1)
PRINT PIN14$
ERROR$ = PIN14$
CALL ERRORES
IF ER = 1 GOTO 73
74 LOCATE 22, 13
COLOR 4, 0
INPUT "Desea corregir (S/N)"; COR$
IF UCASE$(COR$) = "S" THEN GOTO 60
```

```
CALL ZOCALO14
PIN16: CALL PANTALLA
160 LOCATE 22, 13
COLOR 11, 0
PRINT "                    "
LOCATE 12, 21
PRINT "          "
LOCATE 12, 21
ER = 0
PIN1$ = INPUT$(1)
PRINT PIN1$
ERROR$ = PIN1$
CALL ERRORES
IF ER = 1 GOTO 160
161 LOCATE 13, 21
PRINT "          "
LOCATE 13, 21
ER = 0
PIN2$ = INPUT$(1)
PRINT PIN2$
ERROR$ = PIN2$
CALL ERRORES
IF ER = 1 GOTO 161
162 LOCATE 14, 21
PRINT "          "
LOCATE 14, 21
ER = 0
PIN3$ = INPUT$(1)
PRINT PIN3$
ERROR$ = PIN3$
CALL ERRORES
IF ER = 1 GOTO 162
163 LOCATE 15, 21
PRINT "          "
LOCATE 15, 21
ER = 0
PIN4$ = INPUT$(1)
PRINT PIN4$
ERROR$ = PIN4$
CALL ERRORES
IF ER = 1 GOTO 163
164 LOCATE 16, 21
PRINT "          "
LOCATE 16, 21
ER = 0
PIN5$ = INPUT$(1)
PRINT PIN5$
ERROR$ = PIN5$
CALL ERRORES
IF ER = 1 GOTO 164
165 LOCATE 17, 21
PRINT "          "
LOCATE 17, 21
ER = 0
PIN6$ = INPUT$(1)
PRINT PIN6$
```

```
ERROR$ = PIN6$
CALL ERRORES
IF ER = 1 GOTO 165
166 LOCATE 18, 21
PRINT "            "
LOCATE 18, 21
ER = 0
PIN7$ = INPUT$(1)
PRINT PIN7$
ERROR$ = PIN7$
CALL ERRORES
IF ER = 1 GOTO 166
167 LOCATE 19, 21
PRINT "            "
LOCATE 19, 21
ER = 0
PIN8$ = INPUT$(1)
PRINT PIN8$
ERROR$ = PIN8$
CALL ERRORES
IF ER = 1 GOTO 167
168 LOCATE 12, 50
PRINT "            "
LOCATE 12, 50
ER = 0
PIN9$ = INPUT$(1)
PRINT PIN9$
ERROR$ = PIN9$
CALL ERRORES
IF ER = 1 GOTO 168
169 LOCATE 13, 50
PRINT "            "
LOCATE 13, 50
ER = 0
PIN10$ = INPUT$(1)
PRINT PIN10$
ERROR$ = PIN10$
CALL ERRORES
IF ER = 1 GOTO 169
170 LOCATE 14, 50
PRINT "            "
LOCATE 14, 50
ER = 0
PIN11$ = INPUT$(1)
PRINT PIN11$
ERROR$ = PIN11$
CALL ERRORES
IF ER = 1 GOTO 170
171 LOCATE 15, 50
PRINT "            "
LOCATE 15, 50
ER = 0
PIN12$ = INPUT$(1)
PRINT PIN12$
ERROR$ = PIN12$
CALL ERRORES
```

```
IF ER = 1 GOTO 171
172 LOCATE 16, 50
PRINT "            "
LOCATE 16, 50
ER = 0
PIN13$ = INPUT$(1)
PRINT PIN13$
ERROR$ = PIN13$
CALL ERRORES
IF ER = 1 GOTO 172
173 LOCATE 17, 50
PRINT "            "
LOCATE 17, 50
ER = 0
PIN14$ = INPUT$(1)
PRINT PIN14$
ERROR$ = PIN14$
CALL ERRORES
IF ER = 1 GOTO 173
174 LOCATE 18, 50
PRINT "            "
LOCATE 18, 50
ER = 0
PIN15$ = INPUT$(1)
PRINT PIN15$
ERROR$ = PIN15$
CALL ERRORES
IF ER = 1 GOTO 174
175 LOCATE 19, 50
PRINT "            "
LOCATE 19, 50
ER = 0
PIN16$ = INPUT$(1)
PRINT PIN16$
ERROR$ = PIN16$
CALL ERRORES
IF ER = 1 GOTO 175
176 LOCATE 22, 13
COLOR 4, 0
INPUT "Desea corregir (S/N)"; COR$
IF UCASE$(COR$) = "S" THEN GOTO 160
CALL ZOCALO16
PIN18: CALL PANTALLA
260 LOCATE 22, 13
COLOR 11, 0
PRINT "                    "
LOCATE 12, 21
PRINT "            "
LOCATE 12, 21
ER = 0
PIN1$ = INPUT$(1)
PRINT PIN1$
ERROR$ = PIN1$
CALL ERRORES
IF ER = 1 GOTO 260
261 LOCATE 13, 21
```

```
PRINT "            "
LOCATE 13, 21
ER = 0
PIN2$ = INPUT$(1)
PRINT PIN2$
ERROR$ = PIN2$
CALL ERRORES
IF ER = 1 GOTO 261
262 LOCATE 14, 21
PRINT "            "
LOCATE 14, 21
ER = 0
PIN3$ = INPUT$(1)
PRINT PIN3$
ERROR$ = PIN3$
CALL ERRORES
IF ER = 1 GOTO 262
263 LOCATE 15, 21
PRINT "            "
LOCATE 15, 21
ER = 0
PIN4$ = INPUT$(1)
PRINT PIN4$
ERROR$ = PIN4$
CALL ERRORES
IF ER = 1 GOTO 263
264 LOCATE 16, 21
PRINT "            "
LOCATE 16, 21
ER = 0
PIN5$ = INPUT$(1)
PRINT PIN5$
ERROR$ = PIN5$
CALL ERRORES
IF ER = 1 GOTO 264
265 LOCATE 17, 21
PRINT "            "
LOCATE 17, 21
ER = 0
PIN6$ = INPUT$(1)
PRINT PIN6$
ERROR$ = PIN6$
CALL ERRORES
IF ER = 1 GOTO 265
266 LOCATE 18, 21
PRINT "            "
LOCATE 18, 21
ER = 0
PIN7$ = INPUT$(1)
PRINT PIN7$
ERROR$ = PIN7$
CALL ERRORES
IF ER = 1 GOTO 266
267 LOCATE 19, 21
PRINT "            "
LOCATE 19, 21
```

```
ER = 0
PIN8$ = INPUT$(1)
PRINT PIN8$
ERROR$ = PIN8$
CALL ERRORES
IF ER = 1 GOTO 267
268 LOCATE 20, 21
PRINT "          "
LOCATE 20, 21
ER = 0
PIN9$ = INPUT$(1)
PRINT PIN9$
ERROR$ = PIN9$
CALL ERRORES
IF ER = 1 GOTO 268
269 LOCATE 12, 50
PRINT "          "
LOCATE 12, 50
ER = 0
PIN10$ = INPUT$(1)
PRINT PIN10$
ERROR$ = PIN10$
CALL ERRORES
IF ER = 1 GOTO 269
270 LOCATE 13, 50
PRINT "          "
LOCATE 13, 50
ER = 0
PIN11$ = INPUT$(1)
PRINT PIN11$
ERROR$ = PIN11$
CALL ERRORES
IF ER = 1 GOTO 270
271 LOCATE 14, 50
PRINT "          "
LOCATE 14, 50
ER = 0
PIN12$ = INPUT$(1)
PRINT PIN12$
ERROR$ = PIN12$
CALL ERRORES
IF ER = 1 GOTO 271
272 LOCATE 15, 50
PRINT "          "
LOCATE 15, 50
ER = 0
PIN13$ = INPUT$(1)
PRINT PIN13$
ERROR$ = PIN13$
CALL ERRORES
IF ER = 1 GOTO 272
273 LOCATE 16, 50
PRINT "          "
LOCATE 16, 50
ER = 0
PIN14$ = INPUT$(1)
```

```
PRINT PIN14$
ERROR$ = PIN14$
CALL ERRORES
IF ER = 1 GOTO 273
274 LOCATE 17, 50
PRINT "            "
LOCATE 17, 50
ER = 0
PIN15$ = INPUT$(1)
PRINT PIN15$
ERROR$ = PIN15$
CALL ERRORES
IF ER = 1 GOTO 274
275 LOCATE 18, 50
PRINT "            "
LOCATE 18, 50
ER = 0
PIN16$ = INPUT$(1)
PRINT PIN16$
ERROR$ = PIN16$
CALL ERRORES
IF ER = 1 GOTO 275
276 LOCATE 19, 50
PRINT "            "
LOCATE 19, 50
ER = 0
PIN17$ = INPUT$(1)
PRINT PIN17$
ERROR$ = PIN17$
CALL ERRORES
IF ER = 1 GOTO 276
277 LOCATE 20, 50
PRINT "            "
LOCATE 20, 50
ER = 0
PIN18$ = INPUT$(1)
PRINT PIN18$
ERROR$ = PIN18$
CALL ERRORES
IF ER = 1 GOTO 277
278 LOCATE 22, 13
COLOR 4, 0
INPUT "Desea corregir (S/N)"; COR$
IF UCASE$(COR$) = "S" THEN GOTO 260
CALL ZOCALO18
PIN20: CALL PANTALLA
360 LOCATE 22, 13
COLOR 11, 0
PRINT "                    "
LOCATE 12, 21
PRINT "            "
LOCATE 12, 21
ER = 0
PIN1$ = INPUT$(1)
PRINT PIN1$
ERROR$ = PIN1$
```

```
CALL ERRORES
IF ER = 1 GOTO 360
361 LOCATE 13, 21
PRINT "            "
LOCATE 13, 21
ER = 0
PIN2$ = INPUT$(1)
PRINT PIN2$
ERROR$ = PIN2$
CALL ERRORES
IF ER = 1 GOTO 361
362 LOCATE 14, 21
PRINT "            "
LOCATE 14, 21
ER = 0
PIN3$ = INPUT$(1)
PRINT PIN3$
ERROR$ = PIN3$
CALL ERRORES
IF ER = 1 GOTO 362
363 LOCATE 15, 21
PRINT "            "
LOCATE 15, 21
ER = 0
PIN4$ = INPUT$(1)
PRINT PIN4$
ERROR$ = PIN4$
CALL ERRORES
IF ER = 1 GOTO 363
364 LOCATE 16, 21
PRINT "            "
LOCATE 16, 21
ER = 0
PIN5$ = INPUT$(1)
PRINT PIN5$
ERROR$ = PIN5$
CALL ERRORES
IF ER = 1 GOTO 364
365 LOCATE 17, 21
PRINT "            "
LOCATE 17, 21
ER = 0
PIN6$ = INPUT$(1)
PRINT PIN6$
ERROR$ = PIN6$
CALL ERRORES
IF ER = 1 GOTO 365
366 LOCATE 18, 21
PRINT "            "
LOCATE 18, 21
ER = 0
PIN7$ = INPUT$(1)
PRINT PIN7$
ERROR$ = PIN7$
CALL ERRORES
IF ER = 1 GOTO 366
```

```
367 LOCATE 19, 21
PRINT "          "
LOCATE 19, 21
ER = 0
PIN8$ = INPUT$(1)
PRINT PIN8$
ERROR$ = PIN8$
CALL ERRORES
IF ER = 1 GOTO 367
368 LOCATE 20, 21
PRINT "          "
LOCATE 20, 21
ER = 0
PIN9$ = INPUT$(1)
PRINT PIN9$
ERROR$ = PIN9$
CALL ERRORES
IF ER = 1 GOTO 368
369 LOCATE 21, 21
PRINT "          "
LOCATE 21, 21
ER = 0
PIN10$ = INPUT$(1)
PRINT PIN10$
ERROR$ = PIN10$
CALL ERRORES
IF ER = 1 GOTO 369
370 LOCATE 12, 50
PRINT "          "
LOCATE 12, 50
ER = 0
PIN11$ = INPUT$(1)
PRINT PIN11$
ERROR$ = PIN11$
CALL ERRORES
IF ER = 1 GOTO 370
371 LOCATE 13, 50
PRINT "          "
LOCATE 13, 50
ER = 0
PIN12$ = INPUT$(1)
PRINT PIN12$
ERROR$ = PIN12$
CALL ERRORES
IF ER = 1 GOTO 371
372 LOCATE 14, 50
PRINT "          "
LOCATE 14, 50
ER = 0
PIN13$ = INPUT$(1)
PRINT PIN13$
ERROR$ = PIN13$
CALL ERRORES
IF ER = 1 GOTO 372
373 LOCATE 15, 50
PRINT "          "
```

```
LOCATE 15, 50
ER = 0
PIN14$ = INPUT$(1)
PRINT PIN14$
ERROR$ = PIN14$
CALL ERRORES
IF ER = 1 GOTO 373
374 LOCATE 16, 50
PRINT "          "
LOCATE 16, 50
ER = 0
PIN15$ = INPUT$(1)
PRINT PIN15$
ERROR$ = PIN15$
CALL ERRORES
IF ER = 1 GOTO 374
375 LOCATE 17, 50
PRINT "          "
LOCATE 17, 50
ER = 0
PIN16$ = INPUT$(1)
PRINT PIN16$
ERROR$ = PIN16$
CALL ERRORES
IF ER = 1 GOTO 375
376 LOCATE 18, 50
PRINT "          "
LOCATE 18, 50
ER = 0
PIN17$ = INPUT$(1)
PRINT PIN17$
ERROR$ = PIN17$
CALL ERRORES
IF ER = 1 GOTO 376
377 LOCATE 19, 50
PRINT "          "
LOCATE 19, 50
ER = 0
PIN18$ = INPUT$(1)
PRINT PIN18$
ERROR$ = PIN18$
CALL ERRORES
IF ER = 1 GOTO 377
378 LOCATE 20, 50
PRINT "          "
LOCATE 20, 50
ER = 0
PIN19$ = INPUT$(1)
PRINT PIN19$
ERROR$ = PIN19$
CALL ERRORES
IF ER = 1 GOTO 378
379 LOCATE 21, 50
PRINT "          "
LOCATE 21, 50
ER = 0
```

```
PIN20$ = INPUT$(1)
PRINT PIN20$
ERROR$ = PIN20$
CALL ERRORES
IF ER = 1 GOTO 379
380 LOCATE 22, 13
COLOR 4, 0
INPUT "Desea corregir (S/N)"; COR$
IF UCASE$(COR$) = "S" THEN GOTO 360
CALL ZOCALO20
CIEXIST: COLOR 11, 0
LOCATE 10, 16
PRINT "     CI #     CONSTA EN LIBRERIA          "
LOCATE 12, 16
PRINT "     DESCRIPCION:                "
LOCATE 10, 28
PRINT SERIE$
LOCATE 12, 38
PRINT DESCRIP$
3002 LOCATE 22, 13
COLOR 11, 0
PRINT "                                "
LOCATE 22, 13
COLOR 4, 0
CLOSE #1
INPUT "Desea intentar otra serie (S/N)"; SER$
IF UCASE$(SER$) = "S" THEN GOTO 29
IF UCASE$(SER$) <> "S" THEN GOTO 3001
3001 LOCATE 22, 13
COLOR 11, 0
PRINT "                                "
LOCATE 22, 13
COLOR 4, 0
INPUT "Desea abandonar el programa (S/N)"; SER$
IF UCASE$(SER$) = "S" THEN GOTO FIN2
IF UCASE$(SER$) <> "S" THEN GOTO 3002
FIN2: COLOR 7, 0
CLS
END
END SUB

SUB PANTALLA
SHARED PINES$
CALL CUAD3(3, 7, 23, 73, 196, 179, 218, 217, 191, 192, 11, 0)
IF PINES$ = "14" THEN
    POS1$ = "PIN #1:"
    POS2$ = "PIN #2:"
    POS3$ = "PIN #3:"
    POS4$ = "PIN #4:"
    POS5$ = "PIN #5:"
    POS6$ = "PIN #6:"
    POS7$ = "PIN #7:"
    POS8$ = ""
    POS9$ = ""
    POS10$ = ""
    POS11$ = "PIN #8:"
```

```
        POS12$ = "PIN #9:"
        POS13$ = "PIN #10:"
        POS14$ = "PIN #11:"
        POS15$ = "PIN #12:"
        POS16$ = "PIN #13:"
        POS17$ = "PIN #14:"
        POS18$ = ""
        POS19$ = ""
        POS20$ = ""
    END IF
    IF PINES$ = "16" THEN
        POS1$ = "PIN #1:"
        POS2$ = "PIN #2:"
        POS3$ = "PIN #3:"
        POS4$ = "PIN #4:"
        POS5$ = "PIN #5:"
        POS6$ = "PIN #6:"
        POS7$ = "PIN #7:"
        POS8$ = "PIN #8:"
        POS9$ = ""
        POS10$ = ""
        POS11$ = "PIN #9:"
        POS12$ = "PIN #10:"
        POS13$ = "PIN #11:"
        POS14$ = "PIN #12:"
        POS15$ = "PIN #13:"
        POS16$ = "PIN #14:"
        POS17$ = "PIN #15:"
        POS18$ = "PIN #16:"
        POS19$ = ""
        POS20$ = ""
    END IF
    IF PINES$ = "18" THEN
        POS1$ = "PIN #1:"
        POS2$ = "PIN #2:"
        POS3$ = "PIN #3:"
        POS4$ = "PIN #4:"
        POS5$ = "PIN #5:"
        POS6$ = "PIN #6:"
        POS7$ = "PIN #7:"
        POS8$ = "PIN #8:"
        POS9$ = "PIN #9:"
        POS10$ = ""
        POS11$ = "PIN #10:"
        POS12$ = "PIN #11:"
        POS13$ = "PIN #12:"
        POS14$ = "PIN #13:"
        POS15$ = "PIN #14:"
        POS16$ = "PIN #15:"
        POS17$ = "PIN #16:"
        POS18$ = "PIN #17:"
        POS19$ = "PIN #18:"
        POS20$ = ""
    END IF
    IF PINES$ = "20" THEN
        POS1$ = "PIN #1:"
```

```
        POS2$ = "PIN #2:"
        POS3$ = "PIN #3:"
        POS4$ = "PIN #4:"
        POS5$ = "PIN #5:"
        POS6$ = "PIN #6:"
        POS7$ = "PIN #7:"
        POS8$ = "PIN #8:"
        POS9$ = "PIN #9:"
        POS10$ = "PIN #10:"
        POS11$ = "PIN #11:"
        POS12$ = "PIN #12:"
        POS13$ = "PIN #13:"
        POS14$ = "PIN #14:"
        POS15$ = "PIN #15:"
        POS16$ = "PIN #16:"
        POS17$ = "PIN #17:"
        POS18$ = "PIN #18:"
        POS19$ = "PIN #19:"
        POS20$ = "PIN #20:"
    END IF
    LOCATE 4, 8
    PRINT "                                                  "
    LOCATE 5, 8
    PRINT "  INGRESE LAS FUNCIONES DE LOS    PINES DEL CIRCUITO INTEGRADO   "
    LOCATE 5, 39
    PRINT PINES$
    LOCATE 6, 8
    PRINT "                                                  "
    LOCATE 7, 8
    PRINT "    E : ENTRADA DE DATOS       + : POLARIZACION POSITIVA      "
    LOCATE 8, 8
    PRINT "                                                  "
    LOCATE 9, 8
    PRINT "    S : SALIDA DE DATOS        - : POLARIZACION NEGATIVA      "
    LOCATE 10, 8
    PRINT "                                                  "
    LOCATE 11, 8
    PRINT "                                                  "
    LOCATE 12, 8
    PRINT "                                                  "
    LOCATE 12, 13
    PRINT POS1$
    LOCATE 12, 41
    PRINT POS11$
    LOCATE 13, 8
    PRINT "                                                  "
    LOCATE 13, 13
    PRINT POS2$
    LOCATE 13, 41
    PRINT POS12$
    LOCATE 14, 8
    PRINT "                                                  "
    LOCATE 14, 13
    PRINT POS3$
    LOCATE 14, 41
    PRINT POS13$
```

```
LOCATE 15, 8
PRINT "                                   "
LOCATE 15, 13
PRINT POS4$
LOCATE 15, 41
PRINT POS14$
LOCATE 16, 8
PRINT "                                   "
LOCATE 16, 13
PRINT POS5$
LOCATE 16, 41
PRINT POS15$
LOCATE 17, 8
PRINT "                                   "
LOCATE 17, 13
PRINT POS6$
LOCATE 17, 41
PRINT POS16$
LOCATE 18, 8
PRINT "                                   "
LOCATE 18, 13
PRINT POS7$
LOCATE 18, 41
PRINT POS17$
LOCATE 19, 8
PRINT "                                   "
LOCATE 19, 13
PRINT POS8$
LOCATE 19, 41
PRINT POS18$
LOCATE 20, 8
PRINT "                                   "
LOCATE 20, 13
PRINT POS9$
LOCATE 20, 41
PRINT POS19$
LOCATE 21, 8
PRINT "                                   "
LOCATE 21, 13
PRINT POS10$
LOCATE 21, 41
PRINT POS20$
LOCATE 22, 8
PRINT "                                   "
END SUB

SUB PORT14
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
COLOR 11, 0
LOCATE 4, 8
PRINT "                                   "
LOCATE 5, 8
```

```
PRINT "                    PRUEBA #                    "
LOCATE 6, 8
PRINT "                                                "
LOCATE 7, 8
PRINT "   INGRESE LOS NIVELES LOGICOS APLICADOS A LAS RESPECTIVAS    "
LOCATE 8, 8
PRINT "      ENTRADAS Y RESULTANTES EN LAS RESPECTIVAS SALIDAS       "
LOCATE 9, 8
PRINT "      INGRESE << F >> PARA TERMINAR EL INGRESO DE DATOS       "
LOCATE 10, 8
PRINT "                                                "
LOCATE 11, 8
PRINT "                                                "
LOCATE 12, 8
PRINT "   PIN #1:              PIN #8:                 "
LOCATE 13, 8
PRINT "   PIN #2:              PIN #9:                 "
LOCATE 14, 8
PRINT "   PIN #3:              PIN #10:                "
LOCATE 15, 8
PRINT "   PIN #4:              PIN #11:                "
LOCATE 16, 8
PRINT "   PIN #5:              PIN #12:                "
LOCATE 17, 8
PRINT "   PIN #6:              PIN #13:                "
LOCATE 18, 8
PRINT "   PIN #7:              PIN #14:                "
LOCATE 19, 8
PRINT "                                                "
LOCATE 20, 8
PRINT "                                                "
LOCATE 21, 8
PRINT "                                                "
LOCATE 22, 8
PRINT "                                                "
END SUB

SUB PORT16
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
COLOR 11, 0
LOCATE 4, 8
PRINT "                                                "
LOCATE 5, 8
PRINT "                    PRUEBA #                    "
LOCATE 6, 8
PRINT "                                                "
LOCATE 7, 8
PRINT "   INGRESE LOS NIVELES LOGICOS APLICADOS A LAS RESPECTIVAS    "
LOCATE 8, 8
PRINT "      ENTRADAS Y RESULTANTES EN LAS RESPECTIVAS SALIDAS       "
LOCATE 9, 8
PRINT "      INGRESE << F >> PARA TERMINAR EL INGRESO DE DATOS       "
```

```
LOCATE 10, 8
PRINT "                                    "
LOCATE 11, 8
PRINT "                                    "
LOCATE 12, 8
PRINT "   PIN #1:          PIN #9:         "
LOCATE 13, 8
PRINT "   PIN #2:          PIN #10:        "
LOCATE 14, 8
PRINT "   PIN #3:          PIN #11:        "
LOCATE 15, 8
PRINT "   PIN #4:          PIN #12:        "
LOCATE 16, 8
PRINT "   PIN #5:          PIN #13:        "
LOCATE 17, 8
PRINT "   PIN #6:          PIN #14:        "
LOCATE 18, 8
PRINT "   PIN #7:          PIN #15:        "
LOCATE 19, 8
PRINT "   PIN #8:          PIN #16:        "
LOCATE 20, 8
PRINT "                                    "
LOCATE 21, 8
PRINT "                                    "
LOCATE 22, 8
PRINT "                                    "
END SUB

SUB PORT18
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
COLOR 11, 0
LOCATE 4, 8
PRINT "                                    "
LOCATE 5, 8
PRINT "              PRUEBA #              "
LOCATE 6, 8
PRINT "                                    "
LOCATE 7, 8
PRINT "   INGRESE LOS NIVELES LOGICOS APLICADOS A LAS RESPECTIVAS   "
LOCATE 8, 8
PRINT "     ENTRADAS Y RESULTANTES EN LAS RESPECTIVAS SALIDAS      "
LOCATE 9, 8
PRINT "     INGRESE << F >> PARA TERMINAR EL INGRESO DE DATOS      "
LOCATE 10, 8
PRINT "                                    "
LOCATE 11, 8
PRINT "                                    "
LOCATE 12, 8
PRINT "   PIN #1:          PIN #10:        "
LOCATE 13, 8
PRINT "   PIN #2:          PIN #11:        "
LOCATE 14, 8
```

```
PRINT "    PIN #3:              PIN #12:              "
LOCATE 15, 8
PRINT "    PIN #4:              PIN #13:              "
LOCATE 16, 8
PRINT "    PIN #5:              PIN #14:              "
LOCATE 17, 8
PRINT "    PIN #6:              PIN #15:              "
LOCATE 18, 8
PRINT "    PIN #7:              PIN #16:              "
LOCATE 19, 8
PRINT "    PIN #8:              PIN #17:              "
LOCATE 20, 8
PRINT "    PIN #9:              PIN #18:              "
LOCATE 21, 8
PRINT "                                              "
LOCATE 22, 8
PRINT "                                              "
END SUB


SUB PORT20
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
COLOR 11, 0
LOCATE 4, 8
PRINT "                                              "
LOCATE 5, 8
PRINT "                PRUEBA #                      "
LOCATE 6, 8
PRINT "                                              "
LOCATE 7, 8
PRINT "    INGRESE LOS NIVELES LOGICOS APLICADOS A LAS RESPECTIVAS    "
LOCATE 8, 8
PRINT "      ENTRADAS Y RESULTANTES EN LAS RESPECTIVAS SALIDAS       "
LOCATE 9, 8
PRINT "      INGRESE << F >> PARA TERMINAR EL INGRESO DE DATOS       "
LOCATE 10, 8
PRINT "                                              "
LOCATE 11, 8
PRINT "                                              "
LOCATE 12, 8
PRINT "    PIN #1:              PIN #11:              "
LOCATE 13, 8
PRINT "    PIN #2:              PIN #12:              "
LOCATE 14, 8
PRINT "    PIN #3:              PIN #13:              "
LOCATE 15, 8
PRINT "    PIN #4:              PIN #14:              "
LOCATE 16, 8
PRINT "    PIN #5:              PIN #15:              "
LOCATE 17, 8
PRINT "    PIN #6:              PIN #16:              "
LOCATE 18, 8
```

```
PRINT "    PIN #7:              PIN #17:              "
LOCATE 19, 8
PRINT "    PIN #8:              PIN #18:              "
LOCATE 20, 8
PRINT "    PIN #9:              PIN #19:              "
LOCATE 21, 8
PRINT "    PIN #10:             PIN #20:              "
LOCATE 22, 8
PRINT "                                              "
END SUB

SUB PORTPRUEB
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2, P19P0, P19P1,
P19P2, P20P0, P20P1, P20P2, P21P0, P21P1, P21P2, P22P0, _
P22P1, P22P2, P23P0, P23P1, P23P2
SHARED Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11, Z12, Z13, Z14, Z15, Z16, Z17, Z18, Z19, Z20
SHARED P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED PINES$
CALL CUAD3(3, 7, 23, 73, 196, 179, 218, 217, 191, 192, 11, 0)
SELECT CASE PINES$
     CASE "14"
     CALL PORT14
     CASE "16"
     CALL PORT16
     CASE "18"
     CALL PORT18
     CASE "20"
     CALL PORT20
END SELECT
LOCATE 5, 43
PRINT "1"
SELECT CASE PINES$
     CASE "14"
     CALL PRUEBAS14
     CASE "16"
     CALL PRUEBAS16
     CASE "18"
     CALL PRUEBAS18
     CASE "20"
     CALL PRUEBAS20
END SELECT
P1P0 = 2 + Z17 + Z18 + Z19 + Z20
P1P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
P1P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
SELECT CASE PINES$
     CASE "14"
     CALL PORT14
     CASE "16"
     CALL PORT16
     CASE "18"
```

```
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
    PRINT "2"
    SELECT CASE PINES$
        CASE "14"
        CALL PRUEBAS14
        CASE "16"
        CALL PRUEBAS16
        CASE "18"
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P2P0 = 2 + Z17 + Z18 + Z19 + Z20
    P2P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
    P2P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
    SELECT CASE PINES$
        CASE "14"
        CALL PORT14
        CASE "16"
        CALL PORT16
        CASE "18"
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
    PRINT "3"
    SELECT CASE PINES$
        CASE "14"
        CALL PRUEBAS14
        CASE "16"
        CALL PRUEBAS16
        CASE "18"
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P3P0 = 2 + Z17 + Z18 + Z19 + Z20
    P3P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
    P3P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
    SELECT CASE PINES$
        CASE "14"
        CALL PORT14
        CASE "16"
        CALL PORT16
        CASE "18"
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
    PRINT "4"
```

```
SELECT CASE PINES$
      CASE "14"
      CALL PRUEBAS14
      CASE "16"
      CALL PRUEBAS16
      CASE "18"
      CALL PRUEBAS18
      CASE "20"
      CALL PRUEBAS20
END SELECT
P4P0 = 2 + Z17 + Z18 + Z19 + Z20
P4P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
P4P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
SELECT CASE PINES$
      CASE "14"
      CALL PORT14
      CASE "16"
      CALL PORT16
      CASE "18"
      CALL PORT18
      CASE "20"
      CALL PORT20
END SELECT
LOCATE 5, 43
PRINT "5"
SELECT CASE PINES$
      CASE "14"
      CALL PRUEBAS14
      CASE "16"
      CALL PRUEBAS16
      CASE "18"
      CALL PRUEBAS18
      CASE "20"
      CALL PRUEBAS20
END SELECT
P5P0 = 2 + Z17 + Z18 + Z19 + Z20
P5P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
P5P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
SELECT CASE PINES$
      CASE "14"
      CALL PORT14
      CASE "16"
      CALL PORT16
      CASE "18"
      CALL PORT18
      CASE "20"
      CALL PORT20
END SELECT
LOCATE 5, 43
PRINT "6"
SELECT CASE PINES$
      CASE "14"
      CALL PRUEBAS14
      CASE "16"
      CALL PRUEBAS16
      CASE "18"
```

```
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P6P0 = 2 + Z17 + Z18 + Z19 + Z20
    P6P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
    P6P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
    SELECT CASE PINES$
        CASE "14"
        CALL PORT14
        CASE "16"
        CALL PORT16
        CASE "18"
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
    PRINT "7"
    SELECT CASE PINES$
        CASE "14"
        CALL PRUEBAS14
        CASE "16"
        CALL PRUEBAS16
        CASE "18"
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P7P0 = 2 + Z17 + Z18 + Z19 + Z20
    P7P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
    P7P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
    SELECT CASE PINES$
        CASE "14"
        CALL PORT14
        CASE "16"
        CALL PORT16
        CASE "18"
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
    PRINT "8"
    SELECT CASE PINES$
        CASE "14"
        CALL PRUEBAS14
        CASE "16"
        CALL PRUEBAS16
        CASE "18"
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P8P0 = 2 + Z17 + Z18 + Z19 + Z20
    P8P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
```

```
P8P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
SELECT CASE PINES$
      CASE "14"
      CALL PORT14
      CASE "16"
      CALL PORT16
      CASE "18"
      CALL PORT18
      CASE "20"
      CALL PORT20
END SELECT
LOCATE 5, 43
PRINT "9"
SELECT CASE PINES$
      CASE "14"
      CALL PRUEBAS14
      CASE "16"
      CALL PRUEBAS16
      CASE "18"
      CALL PRUEBAS18
      CASE "20"
      CALL PRUEBAS20
END SELECT
P9P0 = 2 + Z17 + Z18 + Z19 + Z20
P9P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
P9P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
SELECT CASE PINES$
      CASE "14"
      CALL PORT14
      CASE "16"
      CALL PORT16
      CASE "18"
      CALL PORT18
      CASE "20"
      CALL PORT20
END SELECT
LOCATE 5, 43
PRINT "10"
SELECT CASE PINES$
      CASE "14"
      CALL PRUEBAS14
      CASE "16"
      CALL PRUEBAS16
      CASE "18"
      CALL PRUEBAS18
      CASE "20"
      CALL PRUEBAS20
END SELECT
P10P0 = 2 + Z17 + Z18 + Z19 + Z20
P10P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
P10P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
SELECT CASE PINES$
      CASE "14"
      CALL PORT14
      CASE "16"
      CALL PORT16
```

```
        CASE "18"
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
    PRINT "11"
    SELECT CASE PINES$
        CASE "14"
        CALL PRUEBAS14
        CASE "16"
        CALL PRUEBAS16
        CASE "18"
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P11P0 = 2 + Z17 + Z18 + Z19 + Z20
    P11P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
    P11P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
    SELECT CASE PINES$
        CASE "14"
        CALL PORT14
        CASE "16"
        CALL PORT16
        CASE "18"
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
    PRINT "12"
    SELECT CASE PINES$
        CASE "14"
        CALL PRUEBAS14
        CASE "16"
        CALL PRUEBAS16
        CASE "18"
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P12P0 = 2 + Z17 + Z18 + Z19 + Z20
    P12P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
    P12P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
    SELECT CASE PINES$
        CASE "14"
        CALL PORT14
        CASE "16"
        CALL PORT16
        CASE "18"
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
```

```
PRINT "13"
SELECT CASE PINES$
      CASE "14"
      CALL PRUEBAS14
      CASE "16"
      CALL PRUEBAS16
      CASE "18"
      CALL PRUEBAS18
      CASE "20"
      CALL PRUEBAS20
END SELECT
P13P0 = 2 + Z17 + Z18 + Z19 + Z20
P13P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
P13P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
SELECT CASE PINES$
      CASE "14"
      CALL PORT14
      CASE "16"
      CALL PORT16
      CASE "18"
      CALL PORT18
      CASE "20"
      CALL PORT20
END SELECT
LOCATE 5, 43
PRINT "14"
SELECT CASE PINES$
      CASE "14"
      CALL PRUEBAS14
      CASE "16"
      CALL PRUEBAS16
      CASE "18"
      CALL PRUEBAS18
      CASE "20"
      CALL PRUEBAS20
END SELECT
P14P0 = 2 + Z17 + Z18 + Z19 + Z20
P14P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
P14P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
SELECT CASE PINES$
      CASE "14"
      CALL PORT14
      CASE "16"
      CALL PORT16
      CASE "18"
      CALL PORT18
      CASE "20"
      CALL PORT20
END SELECT
LOCATE 5, 43
PRINT "15"
SELECT CASE PINES$
      CASE "14"
      CALL PRUEBAS14
      CASE "16"
      CALL PRUEBAS16
```

```
        CASE "18"
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P15P0 = 2 + Z17 + Z18 + Z19 + Z20
    P15P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
    P15P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
    SELECT CASE PINES$
        CASE "14"
        CALL PORT14
        CASE "16"
        CALL PORT16
        CASE "18"
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
    PRINT "16"
    SELECT CASE PINES$
        CASE "14"
        CALL PRUEBAS14
        CASE "16"
        CALL PRUEBAS16
        CASE "18"
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P16P0 = 2 + Z17 + Z18 + Z19 + Z20
    P16P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
    P16P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
    SELECT CASE PINES$
        CASE "14"
        CALL PORT14
        CASE "16"
        CALL PORT16
        CASE "18"
        CALL PORT18
        CASE "20"
        CALL PORT20
    END SELECT
    LOCATE 5, 43
    PRINT "17"
    SELECT CASE PINES$
        CASE "14"
        CALL PRUEBAS14
        CASE "16"
        CALL PRUEBAS16
        CASE "18"
        CALL PRUEBAS18
        CASE "20"
        CALL PRUEBAS20
    END SELECT
    P17P0 = 2 + Z17 + Z18 + Z19 + Z20
```

```
P17P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
P17P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
SELECT CASE PINES$
      CASE "14"
      CALL PORT14
      CASE "16"
      CALL PORT16
      CASE "18"
      CALL PORT18
      CASE "20"
      CALL PORT20
END SELECT
LOCATE 5, 43
PRINT "18"
SELECT CASE PINES$
      CASE "14"
      CALL PRUEBAS14
      CASE "16"
      CALL PRUEBAS16
      CASE "18"
      CALL PRUEBAS18
      CASE "20"
      CALL PRUEBAS20
END SELECT
P18P0 = 2 + Z17 + Z18 + Z19 + Z20
P18P1 = Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9 + Z10
P18P2 = Z1 + Z2 + Z11 + Z12 + Z13 + Z14 + Z15 + Z16
CALL FIN
END SUB

SUB PRUEBAS14
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20
SHARED Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11, Z12, Z13, Z14, Z15, Z16, Z17, Z18, Z19, Z20
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
1160 COLOR 11, 0
LOCATE 12, 21
PRINT "          "
LOCATE 12, 21
IF UCASE$(PIN1$) = "+" OR UCASE$(PIN1$) = "-" THEN
      PRINT PIN1$
      GOTO 1161
END IF
P1$ = INPUT$(1)
PRINT P1$
IF P1$ <> "1" AND P1$ <> "0" AND UCASE$(P1$) <> "F" THEN GOTO 1160
IF P1$ = "1" THEN Z4 = 0
IF P1$ = "0" THEN Z4 = 2
IF UCASE$(P1$) = "F" THEN CALL FIN
1161 LOCATE 13, 21
PRINT "          "
```

```
LOCATE 13, 21
IF UCASE$(PIN2$) = "+" OR UCASE$(PIN2$) = "-" THEN
      PRINT PIN2$
      GOTO 1162
END IF
P2$ = INPUT$(1)
PRINT P2$
IF P2$ <> "1" AND P2$ <> "0" AND UCASE$(P2$) <> "F" THEN GOTO 1161
IF P2$ = "1" THEN Z5 = 0
IF P2$ = "0" THEN Z5 = 4
IF UCASE$(P2$) = "F" THEN CALL FIN
1162 LOCATE 14, 21
PRINT "            "
LOCATE 14, 21
IF UCASE$(PIN3$) = "+" OR UCASE$(PIN3$) = "-" THEN
      PRINT PIN3$
      GOTO 1163
END IF
P3$ = INPUT$(1)
PRINT P3$
IF P3$ <> "1" AND P3$ <> "0" THEN GOTO 1162
IF P3$ = "1" THEN Z6 = 0
IF P3$ = "0" THEN Z6 = 8
1163 LOCATE 15, 21
PRINT "            "
LOCATE 15, 21
IF UCASE$(PIN4$) = "+" OR UCASE$(PIN4$) = "-" THEN
      PRINT PIN4$
      GOTO 1164
END IF
P4$ = INPUT$(1)
PRINT P4$
IF P4$ <> "1" AND P4$ <> "0" THEN GOTO 1163
IF P4$ = "1" THEN Z7 = 0
IF P4$ = "0" THEN Z7 = 16
1164 LOCATE 16, 21
PRINT "            "
LOCATE 16, 21
IF UCASE$(PIN5$) = "+" OR UCASE$(PIN5$) = "-" THEN
      PRINT PIN5$
      GOTO 1165
END IF
P5$ = INPUT$(1)
PRINT P5$
IF P5$ <> "1" AND P5$ <> "0" THEN GOTO 1164
IF P5$ = "1" THEN Z8 = 0
IF P5$ = "0" THEN Z8 = 32
1165 LOCATE 17, 21
PRINT "            "
LOCATE 17, 21
IF UCASE$(PIN6$) = "+" OR UCASE$(PIN6$) = "-" THEN
      PRINT PIN6$
      GOTO 1166
END IF
P6$ = INPUT$(1)
PRINT P6$
```

```
IF P6$ <> "1" AND P6$ <> "0" THEN GOTO 1165
IF P6$ = "1" THEN Z9 = 0
IF P6$ = "0" THEN Z9 = 64
1166 LOCATE 18, 21
PRINT "          "
LOCATE 18, 21
IF UCASE$(PIN7$) = "+" OR UCASE$(PIN7$) = "-" THEN
     PRINT PIN7$
     GOTO 1167
END IF
P7$ = INPUT$(1)
PRINT P7$
IF P7$ <> "1" AND P7$ <> "0" THEN GOTO 1166
IF P7$ = "1" THEN Z10 = 0
IF P7$ = "0" THEN Z10 = 128
1167 LOCATE 12, 50
PRINT "          "
LOCATE 12, 50
IF UCASE$(PIN8$) = "+" OR UCASE$(PIN8$) = "-" THEN
     PRINT PIN8$
     GOTO 1168
END IF
P8$ = INPUT$(1)
PRINT P8$
IF P8$ <> "1" AND P8$ <> "0" THEN GOTO 1167
IF P8$ = "1" THEN Z11 = 0
IF P8$ = "0" THEN Z11 = 4
1168 LOCATE 13, 50
PRINT "          "
LOCATE 13, 50
IF UCASE$(PIN9$) = "+" OR UCASE$(PIN9$) = "-" THEN
     PRINT PIN9$
     GOTO 1169
END IF
P9$ = INPUT$(1)
PRINT P9$
IF P9$ <> "1" AND P9$ <> "0" THEN GOTO 1168
IF P9$ = "1" THEN Z12 = 0
IF P9$ = "0" THEN Z12 = 8
1169 LOCATE 14, 50
PRINT "          "
LOCATE 14, 50
IF UCASE$(PIN10$) = "+" OR UCASE$(PIN10$) = "-" THEN
     PRINT PIN10$
     GOTO 1170
END IF
P10$ = INPUT$(1)
PRINT P10$
IF P10$ <> "1" AND P10$ <> "0" THEN GOTO 1169
IF P10$ = "1" THEN Z13 = 0
IF P10$ = "0" THEN Z13 = 16
1170 LOCATE 15, 50
PRINT "          "
LOCATE 15, 50
IF UCASE$(PIN11$) = "+" OR UCASE$(PIN11$) = "-" THEN
     PRINT PIN11$
```

```
      GOTO 1171
END IF
P11$ = INPUT$(1)
PRINT P11$
IF P11$ <> "1" AND P11$ <> "0" THEN GOTO 1170
IF P11$ = "1" THEN Z14 = 0
IF P11$ = "0" THEN Z14 = 32
1171 LOCATE 16, 50
PRINT "            "
LOCATE 16, 50
IF UCASE$(PIN12$) = "+" OR UCASE$(PIN12$) = "-" THEN
      PRINT PIN12$
      GOTO 1172
END IF
P12$ = INPUT$(1)
PRINT P12$
IF P12$ <> "1" AND P12$ <> "0" THEN GOTO 1171
IF P12$ = "1" THEN Z15 = 0
IF P12$ = "0" THEN Z15 = 64
1172 LOCATE 17, 50
PRINT "            "
LOCATE 17, 50
IF UCASE$(PIN13$) = "+" OR UCASE$(PIN13$) = "-" THEN
      PRINT PIN13$
      GOTO 1173
END IF
P13$ = INPUT$(1)
PRINT P13$
IF P13$ <> "1" AND P13$ <> "0" THEN GOTO 1172
IF P13$ = "1" THEN Z16 = 0
IF P13$ = "0" THEN Z16 = 128
1173 LOCATE 18, 50
PRINT "            "
LOCATE 18, 50
IF UCASE$(PIN14$) = "+" OR UCASE$(PIN14$) = "-" THEN
      PRINT PIN14$
      GOTO 1176
END IF
P14$ = INPUT$(1)
PRINT P14$
IF P14$ <> "1" AND P14$ <> "0" THEN GOTO 1173
IF P14$ = "1" THEN Z17 = 0
IF P14$ = "0" THEN Z17 = 128
1176 LOCATE 22, 13
COLOR 4, 0
INPUT "Desea corregir (S/N)"; COR1$
IF UCASE$(COR1$) = "S" THEN GOTO 1160
END SUB

SUB PRUEBAS16
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20
SHARED Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11, Z12, Z13, Z14, Z15, Z16, Z17, Z18, Z19, Z20
```

```
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
2160 COLOR 11, 0
LOCATE 12, 21
PRINT "          "
LOCATE 12, 21
IF UCASE$(PIN1$) = "+" OR UCASE$(PIN1$) = "-" THEN
    PRINT PIN1$
    GOTO 2161
END IF
P1$ = INPUT$(1)
PRINT P1$
IF P1$ <> "1" AND P1$ <> "0" AND UCASE$(P1$) <> "F" THEN GOTO 2160
IF P1$ = "1" THEN Z3 = 0
IF P1$ = "0" THEN Z3 = 1
IF UCASE$(P1$) = "F" THEN CALL FIN
2161 LOCATE 13, 21
PRINT "          "
LOCATE 13, 21
IF UCASE$(PIN2$) = "+" OR UCASE$(PIN2$) = "-" THEN
    PRINT PIN2$
    GOTO 2162
END IF
P2$ = INPUT$(1)
PRINT P2$
IF P2$ <> "1" AND P2$ <> "0" AND UCASE$(P2$) <> "F" THEN GOTO 2161
IF P2$ = "1" THEN Z4 = 0
IF P2$ = "0" THEN Z4 = 2
IF UCASE$(P2$) = "F" THEN CALL FIN
2162 LOCATE 14, 21
PRINT "          "
LOCATE 14, 21
IF UCASE$(PIN3$) = "+" OR UCASE$(PIN3$) = "-" THEN
    PRINT PIN3$
    GOTO 2163
END IF
P3$ = INPUT$(1)
PRINT P3$
IF P3$ <> "1" AND P3$ <> "0" THEN GOTO 2162
IF P3$ = "1" THEN Z5 = 0
IF P3$ = "0" THEN Z5 = 4
2163 LOCATE 15, 21
PRINT "          "
LOCATE 15, 21
IF UCASE$(PIN4$) = "+" OR UCASE$(PIN4$) = "-" THEN
    PRINT PIN4$
    GOTO 2164
END IF
P4$ = INPUT$(1)
PRINT P4$
IF P4$ <> "1" AND P4$ <> "0" THEN GOTO 2163
IF P4$ = "1" THEN Z6 = 0
IF P4$ = "0" THEN Z6 = 8
```

```
2164 LOCATE 16, 21
PRINT "            "
LOCATE 16, 21
IF UCASE$(PIN5$) = "+" OR UCASE$(PIN5$) = "-" THEN
      PRINT PIN5$
      GOTO 2165
END IF
P5$ = INPUT$(1)
PRINT P5$
IF P5$ <> "1" AND P5$ <> "0" THEN GOTO 2164
IF P5$ = "1" THEN Z7 = 0
IF P5$ = "0" THEN Z7 = 16
2165 LOCATE 17, 21
PRINT "            "
LOCATE 17, 21
IF UCASE$(PIN6$) = "+" OR UCASE$(PIN6$) = "-" THEN
      PRINT PIN6$
      GOTO 2166
END IF
P6$ = INPUT$(1)
PRINT P6$
IF P6$ <> "1" AND P6$ <> "0" THEN GOTO 2165
IF P6$ = "1" THEN Z8 = 0
IF P6$ = "0" THEN Z8 = 32
2166 LOCATE 18, 21
PRINT "            "
LOCATE 18, 21
IF UCASE$(PIN7$) = "+" OR UCASE$(PIN7$) = "-" THEN
      PRINT PIN7$
      GOTO 2167
END IF
P7$ = INPUT$(1)
PRINT P7$
IF P7$ <> "1" AND P7$ <> "0" THEN GOTO 2166
IF P7$ = "1" THEN Z9 = 0
IF P7$ = "0" THEN Z9 = 64
2167 LOCATE 19, 21
PRINT "            "
LOCATE 19, 21
IF UCASE$(PIN8$) = "+" OR UCASE$(PIN8$) = "-" THEN
      PRINT PIN8$
      GOTO 2168
END IF
P8$ = INPUT$(1)
PRINT P8$
IF P8$ <> "1" AND P8$ <> "0" THEN GOTO 2167
IF P8$ = "1" THEN Z10 = 0
IF P8$ = "0" THEN Z10 = 128
2168 LOCATE 12, 50
PRINT "            "
LOCATE 12, 50
IF UCASE$(PIN9$) = "+" OR UCASE$(PIN9$) = "-" THEN
      PRINT PIN9$
      GOTO 2169
END IF
P9$ = INPUT$(1)
```

```
PRINT P9$
IF P9$ <> "1" AND P9$ <> "0" THEN GOTO 2168
IF P9$ = "1" THEN Z11 = 0
IF P9$ = "0" THEN Z11 = 4
2169 LOCATE 13, 50
PRINT "          "
LOCATE 13, 50
IF UCASE$(PIN10$) = "+" OR UCASE$(PIN10$) = "-" THEN
    PRINT PIN10$
    GOTO 2170
END IF
P10$ = INPUT$(1)
PRINT P10$
IF P10$ <> "1" AND P10$ <> "0" THEN GOTO 2169
IF P10$ = "1" THEN Z12 = 0
IF P10$ = "0" THEN Z12 = 8
2170 LOCATE 14, 50
PRINT "          "
LOCATE 14, 50
IF UCASE$(PIN11$) = "+" OR UCASE$(PIN11$) = "-" THEN
    PRINT PIN11$
    GOTO 2171
END IF
P11$ = INPUT$(1)
PRINT P11$
IF P11$ <> "1" AND P11$ <> "0" THEN GOTO 2170
IF P11$ = "1" THEN Z13 = 0
IF P11$ = "0" THEN Z13 = 16
2171 LOCATE 15, 50
PRINT "          "
LOCATE 15, 50
IF UCASE$(PIN12$) = "+" OR UCASE$(PIN12$) = "-" THEN
    PRINT PIN12$
    GOTO 2172
END IF
P12$ = INPUT$(1)
PRINT P12$
IF P12$ <> "1" AND P12$ <> "0" THEN GOTO 2171
IF P12$ = "1" THEN Z14 = 0
IF P12$ = "0" THEN Z14 = 32
2172 LOCATE 16, 50
PRINT "          "
LOCATE 16, 50
IF UCASE$(PIN13$) = "+" OR UCASE$(PIN13$) = "-" THEN
    PRINT PIN13$
    GOTO 2173
END IF
P13$ = INPUT$(1)
PRINT P13$
IF P13$ <> "1" AND P13$ <> "0" THEN GOTO 2172
IF P13$ = "1" THEN Z15 = 0
IF P13$ = "0" THEN Z15 = 64
2173 LOCATE 17, 50
PRINT "          "
LOCATE 17, 50
IF UCASE$(PIN14$) = "+" OR UCASE$(PIN14$) = "-" THEN
```

```
      PRINT PIN14$
      GOTO 2174
END IF
P14$ = INPUT$(1)
PRINT P14$
IF P14$ <> "1" AND P14$ <> "0" THEN GOTO 2173
IF P14$ = "1" THEN Z16 = 0
IF P14$ = "0" THEN Z16 = 128
2174 LOCATE 18, 50
PRINT "          "
LOCATE 18, 50
IF UCASE$(PIN15$) = "+" OR UCASE$(PIN15$) = "-" THEN
      PRINT PIN15$
      GOTO 2175
END IF
P15$ = INPUT$(1)
PRINT P15$
IF P15$ <> "1" AND P15$ <> "0" THEN GOTO 2174
IF P15$ = "1" THEN Z17 = 0
IF P15$ = "0" THEN Z17 = 128
2175 LOCATE 19, 50
PRINT "          "
LOCATE 19, 50
IF UCASE$(PIN16$) = "+" OR UCASE$(PIN16$) = "-" THEN
      PRINT PIN16$
      GOTO 2186
END IF
P16$ = INPUT$(1)
PRINT P16$
IF P16$ <> "1" AND P16$ <> "0" THEN GOTO 2175
IF P16$ = "1" THEN Z18 = 0
IF P16$ = "0" THEN Z18 = 64
2186 LOCATE 22, 13
COLOR 4, 0
INPUT "Desea corregir (S/N)"; COR1$
IF UCASE$(COR1$) = "S" THEN GOTO 2160
END SUB

SUB PRUEBAS18
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20
SHARED Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11, Z12, Z13, Z14, Z15, Z16, Z17, Z18, Z19, Z20
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
3160 COLOR 11, 0
LOCATE 12, 21
PRINT "          "
LOCATE 12, 21
IF UCASE$(PIN1$) = "+" OR UCASE$(PIN1$) = "-" THEN
      PRINT PIN1$
      GOTO 3161
END IF
```

```
P1$ = INPUT$(1)
PRINT P1$
IF P1$ <> "1" AND P1$ <> "0" AND UCASE$(P1$) <> "F" THEN GOTO 3160
IF P1$ = "1" THEN Z2 = 0
IF P1$ = "0" THEN Z2 = 1
IF UCASE$(P1$) = "F" THEN CALL FIN
3161 LOCATE 13, 21
PRINT "            "
LOCATE 13, 21
IF UCASE$(PIN2$) = "+" OR UCASE$(PIN2$) = "-" THEN
      PRINT PIN2$
      GOTO 3162
END IF
P2$ = INPUT$(1)
PRINT P2$
IF P2$ <> "1" AND P2$ <> "0" AND UCASE$(P2$) <> "F" THEN GOTO 3161
IF P2$ = "1" THEN Z3 = 0
IF P2$ = "0" THEN Z3 = 1
IF UCASE$(P2$) = "F" THEN CALL FIN
3162 LOCATE 14, 21
PRINT "            "
LOCATE 14, 21
IF UCASE$(PIN3$) = "+" OR UCASE$(PIN3$) = "-" THEN
      PRINT PIN3$
      GOTO 3163
END IF
P3$ = INPUT$(1)
PRINT P3$
IF P3$ <> "1" AND P3$ <> "0" THEN GOTO 3162
IF P3$ = "1" THEN Z4 = 0
IF P3$ = "0" THEN Z4 = 2
3163 LOCATE 15, 21
PRINT "            "
LOCATE 15, 21
IF UCASE$(PIN4$) = "+" OR UCASE$(PIN4$) = "-" THEN
      PRINT PIN4$
      GOTO 3164
END IF
P4$ = INPUT$(1)
PRINT P4$
IF P4$ <> "1" AND P4$ <> "0" THEN GOTO 3163
IF P4$ = "1" THEN Z5 = 0
IF P4$ = "0" THEN Z5 = 4
3164 LOCATE 16, 21
PRINT "            "
LOCATE 16, 21
IF UCASE$(PIN5$) = "+" OR UCASE$(PIN5$) = "-" THEN
      PRINT PIN5$
      GOTO 3165
END IF
P5$ = INPUT$(1)
PRINT P5$
IF P5$ <> "1" AND P5$ <> "0" THEN GOTO 3164
IF P5$ = "1" THEN Z6 = 0
IF P5$ = "0" THEN Z6 = 8
3165 LOCATE 17, 21
```

```
PRINT "            "
LOCATE 17, 21
IF UCASE$(PIN6$) = "+" OR UCASE$(PIN6$) = "-" THEN
      PRINT PIN6$
      GOTO 3166
END IF
P6$ = INPUT$(1)
PRINT P6$
IF P6$ <> "1" AND P6$ <> "0" THEN GOTO 3165
IF P6$ = "1" THEN Z7 = 0
IF P6$ = "0" THEN Z7 = 16
3166 LOCATE 18, 21
PRINT "            "
LOCATE 18, 21
IF UCASE$(PIN7$) = "+" OR UCASE$(PIN7$) = "-" THEN
      PRINT PIN7$
      GOTO 3167
END IF
P7$ = INPUT$(1)
PRINT P7$
IF P7$ <> "1" AND P7$ <> "0" THEN GOTO 3166
IF P7$ = "1" THEN Z8 = 0
IF P7$ = "0" THEN Z8 = 32
3167 LOCATE 19, 21
PRINT "            "
LOCATE 19, 21
IF UCASE$(PIN8$) = "+" OR UCASE$(PIN8$) = "-" THEN
      PRINT PIN8$
      GOTO 3168
END IF
P8$ = INPUT$(1)
PRINT P8$
IF P8$ <> "1" AND P8$ <> "0" THEN GOTO 3167
IF P8$ = "1" THEN Z9 = 0
IF P8$ = "0" THEN Z9 = 64
3168 LOCATE 20, 21
PRINT "            "
LOCATE 20, 21
IF UCASE$(PIN9$) = "+" OR UCASE$(PIN9$) = "-" THEN
      PRINT PIN9$
      GOTO 3169
END IF
P9$ = INPUT$(1)
PRINT P9$
IF P9$ <> "1" AND P9$ <> "0" THEN GOTO 3168
IF P9$ = "1" THEN Z10 = 0
IF P9$ = "0" THEN Z10 = 128
3169 LOCATE 12, 50
PRINT "            "
LOCATE 12, 50
IF UCASE$(PIN10$) = "+" OR UCASE$(PIN10$) = "-" THEN
      PRINT PIN10$
      GOTO 3170
END IF
P10$ = INPUT$(1)
PRINT P10$
```

```
IF P10$ <> "1" AND P10$ <> "0" THEN GOTO 3169
IF P10$ = "1" THEN Z11 = 0
IF P10$ = "0" THEN Z11 = 4
3170 LOCATE 13, 50
PRINT "          "
LOCATE 13, 50
IF UCASE$(PIN11$) = "+" OR UCASE$(PIN11$) = "-" THEN
      PRINT PIN11$
      GOTO 3171
END IF
P11$ = INPUT$(1)
PRINT P11$
IF P11$ <> "1" AND P11$ <> "0" THEN GOTO 3170
IF P11$ = "1" THEN Z12 = 0
IF P11$ = "0" THEN Z12 = 8
3171 LOCATE 14, 50
PRINT "          "
LOCATE 14, 50
IF UCASE$(PIN12$) = "+" OR UCASE$(PIN12$) = "-" THEN
      PRINT PIN12$
      GOTO 3172
END IF
P12$ = INPUT$(1)
PRINT P12$
IF P12$ <> "1" AND P12$ <> "0" THEN GOTO 3171
IF P12$ = "1" THEN Z13 = 0
IF P12$ = "0" THEN Z13 = 16
3172 LOCATE 15, 50
PRINT "          "
LOCATE 15, 50
IF UCASE$(PIN13$) = "+" OR UCASE$(PIN13$) = "-" THEN
      PRINT PIN13$
      GOTO 3173
END IF
P12$ = INPUT$(1)
PRINT P12$
IF P13$ <> "1" AND P13$ <> "0" THEN GOTO 3172
IF P13$ = "1" THEN Z14 = 0
IF P13$ = "0" THEN Z14 = 32
3173 LOCATE 16, 50
PRINT "          "
LOCATE 16, 50
IF UCASE$(PIN14$) = "+" OR UCASE$(PIN14$) = "-" THEN
      PRINT PIN14$
      GOTO 3174
END IF
P14$ = INPUT$(1)
PRINT P14$
IF P14$ <> "1" AND P14$ <> "0" THEN GOTO 3173
IF P14$ = "1" THEN Z15 = 0
IF P14$ = "0" THEN Z15 = 64
3174 LOCATE 17, 50
PRINT "          "
LOCATE 17, 50
IF UCASE$(PIN15$) = "+" OR UCASE$(PIN15$) = "-" THEN
      PRINT PIN15$
```

```
      GOTO 3175
END IF
P15$ = INPUT$(1)
PRINT P15$
IF P15$ <> "1" AND P15$ <> "0" THEN GOTO 3174
IF P15$ = "1" THEN Z16 = 0
IF P15$ = "0" THEN Z16 = 128
3175 LOCATE 18, 50
PRINT "          "
LOCATE 18, 50
IF UCASE$(PIN16$) = "+" OR UCASE$(PIN16$) = "-" THEN
      PRINT PIN16$
      GOTO 3176
END IF
P16$ = INPUT$(1)
PRINT P16$
IF P16$ <> "1" AND P16$ <> "0" THEN GOTO 3175
IF P16$ = "1" THEN Z17 = 0
IF P16$ = "0" THEN Z17 = 128
3176 LOCATE 19, 50
PRINT "          "
LOCATE 19, 50
IF UCASE$(PIN17$) = "+" OR UCASE$(PIN17$) = "-" THEN
      PRINT PIN17$
      GOTO 3177
END IF
P17$ = INPUT$(1)
PRINT P17$
IF P17$ <> "1" AND P17$ <> "0" THEN GOTO 3176
IF P17$ = "1" THEN Z18 = 0
IF P17$ = "0" THEN Z18 = 64
3177 LOCATE 20, 50
PRINT "          "
LOCATE 20, 50
IF UCASE$(PIN18$) = "+" OR UCASE$(PIN18$) = "-" THEN
      PRINT PIN18$
      GOTO 3186
END IF
P18$ = INPUT$(1)
PRINT P18$
IF P18$ <> "1" AND P18$ <> "0" THEN GOTO 3177
IF P18$ = "1" THEN Z19 = 0
IF P18$ = "0" THEN Z19 = 32
3186 LOCATE 22, 13
COLOR 4, 0
INPUT "Desea corregir (S/N)"; COR1$
IF UCASE$(COR1$) = "S" THEN GOTO 3160
END SUB

SUB PRUEBAS20
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20
SHARED Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11, Z12, Z13, Z14, Z15, Z16, Z17, Z18, Z19, Z20
```

```
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2
SHARED P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2,
P15P0, P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
4160 COLOR 11, 0
LOCATE 12, 21
PRINT "          "
LOCATE 12, 21
IF UCASE$(PIN1$) = "+" OR UCASE$(PIN1$) = "-" THEN
    PRINT PIN1$
    GOTO 4161
END IF
P1$ = INPUT$(1)
PRINT P1$
IF P1$ <> "1" AND P1$ <> "0" AND UCASE$(P1$) <> "F" THEN GOTO 4160
IF P1$ = "1" THEN Z1 = 0
IF P1$ = "0" THEN Z1 = 2
IF UCASE$(P1$) = "F" THEN CALL FIN
4161 LOCATE 13, 21
PRINT "          "
LOCATE 13, 21
IF UCASE$(PIN2$) = "+" OR UCASE$(PIN2$) = "-" THEN
    PRINT PIN2$
    GOTO 4162
END IF
P2$ = INPUT$(1)
PRINT P2$
IF P2$ <> "1" AND P2$ <> "0" AND UCASE$(P2$) <> "F" THEN GOTO 4161
IF P2$ = "1" THEN Z2 = 0
IF P2$ = "0" THEN Z2 = 1
IF UCASE$(P2$) = "F" THEN CALL FIN
4162 LOCATE 14, 21
PRINT "          "
LOCATE 14, 21
IF UCASE$(PIN3$) = "+" OR UCASE$(PIN3$) = "-" THEN
    PRINT PIN3$
    GOTO 4163
END IF
P3$ = INPUT$(1)
PRINT P3$
IF P3$ <> "1" AND P3$ <> "0" THEN GOTO 4162
IF P3$ = "1" THEN Z3 = 0
IF P3$ = "0" THEN Z3 = 1
4163 LOCATE 15, 21
PRINT "          "
LOCATE 15, 21
IF UCASE$(PIN4$) = "+" OR UCASE$(PIN4$) = "-" THEN
    PRINT PIN4$
    GOTO 4164
END IF
P4$ = INPUT$(1)
PRINT P4$
IF P4$ <> "1" AND P4$ <> "0" THEN GOTO 4163
IF P4$ = "1" THEN Z4 = 0
IF P4$ = "0" THEN Z4 = 2
```

```
4164 LOCATE 16, 21
PRINT "           "
LOCATE 16, 21
IF UCASE$(PIN5$) = "+" OR UCASE$(PIN5$) = "-" THEN
      PRINT PIN5$
      GOTO 4165
END IF
P5$ = INPUT$(1)
PRINT P5$
IF P5$ <> "1" AND P5$ <> "0" THEN GOTO 4164
IF P5$ = "1" THEN Z5 = 0
IF P5$ = "0" THEN Z5 = 4
4165 LOCATE 17, 21
PRINT "           "
LOCATE 17, 21
IF UCASE$(PIN6$) = "+" OR UCASE$(PIN6$) = "-" THEN
      PRINT PIN6$
      GOTO 4166
END IF
P6$ = INPUT$(1)
PRINT P6$
IF P6$ <> "1" AND P6$ <> "0" THEN GOTO 4165
IF P6$ = "1" THEN Z6 = 0
IF P6$ = "0" THEN Z6 = 8
4166 LOCATE 18, 21
PRINT "           "
LOCATE 18, 21
IF UCASE$(PIN7$) = "+" OR UCASE$(PIN7$) = "-" THEN
      PRINT PIN7$
      GOTO 4167
END IF
P7$ = INPUT$(1)
PRINT P7$
IF P7$ <> "1" AND P7$ <> "0" THEN GOTO 4166
IF P7$ = "1" THEN Z7 = 0
IF P7$ = "0" THEN Z7 = 16
4167 LOCATE 19, 21
PRINT "           "
LOCATE 19, 21
IF UCASE$(PIN8$) = "+" OR UCASE$(PIN8$) = "-" THEN
      PRINT PIN8$
      GOTO 4168
END IF
P8$ = INPUT$(1)
PRINT P8$
IF P8$ <> "1" AND P8$ <> "0" THEN GOTO 4167
IF P8$ = "1" THEN Z8 = 0
IF P8$ = "0" THEN Z8 = 32
4168 LOCATE 20, 21
PRINT "           "
LOCATE 20, 21
IF UCASE$(PIN9$) = "+" OR UCASE$(PIN9$) = "-" THEN
      PRINT PIN9$
      GOTO 4169
END IF
P9$ = INPUT$(1)
```

```
PRINT P9$
IF P9$ <> "1" AND P9$ <> "0" THEN GOTO 4168
IF P9$ = "1" THEN Z9 = 0
IF P9$ = "0" THEN Z9 = 64
4169 LOCATE 21, 21
PRINT "          "
LOCATE 21, 21
IF UCASE$(PIN10$) = "+" OR UCASE$(PIN10$) = "-" THEN
    PRINT PIN10$
    GOTO 4170
END IF
P10$ = INPUT$(1)
PRINT P10$
IF P10$ <> "1" AND P10$ <> "0" THEN GOTO 4169
IF P10$ = "1" THEN Z10 = 0
IF P10$ = "0" THEN Z10 = 128
4170 LOCATE 12, 50
PRINT "          "
LOCATE 12, 50
IF UCASE$(PIN11$) = "+" OR UCASE$(PIN11$) = "-" THEN
    PRINT PIN11$
    GOTO 4171
END IF
P11$ = INPUT$(1)
PRINT P11$
IF P11$ <> "1" AND P11$ <> "0" THEN GOTO 4170
IF P11$ = "1" THEN Z11 = 0
IF P11$ = "0" THEN Z11 = 4
4171 LOCATE 13, 50
PRINT "          "
LOCATE 13, 50
IF UCASE$(PIN12$) = "+" OR UCASE$(PIN12$) = "-" THEN
    PRINT PIN12$
    GOTO 4172
END IF
P12$ = INPUT$(1)
PRINT P12$
IF P12$ <> "1" AND P12$ <> "0" THEN GOTO 4171
IF P12$ = "1" THEN Z12 = 0
IF P12$ = "0" THEN Z12 = 8
4172 LOCATE 14, 50
PRINT "          "
LOCATE 14, 50
IF UCASE$(PIN13$) = "+" OR UCASE$(PIN13$) = "-" THEN
    PRINT PIN13$
    GOTO 4173
END IF
P13$ = INPUT$(1)
PRINT P13$
IF P13$ <> "1" AND P13$ <> "0" THEN GOTO 4172
IF P13$ = "1" THEN Z13 = 0
IF P13$ = "0" THEN Z13 = 16
4173 LOCATE 15, 50
PRINT "          "
LOCATE 15, 50
IF UCASE$(PIN14$) = "+" OR UCASE$(PIN14$) = "-" THEN
```

```
      PRINT PIN14$
      GOTO 4174
END IF
P14$ = INPUT$(1)
PRINT P14$
IF P14$ <> "1" AND P14$ <> "0" THEN GOTO 4173
IF P14$ = "1" THEN Z14 = 0
IF P14$ = "0" THEN Z14 = 32
4174 LOCATE 16, 50
PRINT "           "
LOCATE 16, 50
IF UCASE$(PIN15$) = "+" OR UCASE$(PIN15$) = "-" THEN
      PRINT PIN15$
      GOTO 4175
END IF
P15$ = INPUT$(1)
PRINT P15$
IF P15$ <> "1" AND P15$ <> "0" THEN GOTO 4174
IF P15$ = "1" THEN Z15 = 0
IF P15$ = "0" THEN Z15 = 64
4175 LOCATE 17, 50
PRINT "           "
LOCATE 17, 50
IF UCASE$(PIN16$) = "+" OR UCASE$(PIN16$) = "-" THEN
      PRINT PIN16$
      GOTO 4176
END IF
P16$ = INPUT$(1)
PRINT P16$
IF P16$ <> "1" AND P16$ <> "0" THEN GOTO 4175
IF P16$ = "1" THEN Z16 = 0
IF P16$ = "0" THEN Z16 = 128
4176 LOCATE 18, 50
PRINT "           "
LOCATE 18, 50
IF UCASE$(PIN17$) = "+" OR UCASE$(PIN17$) = "-" THEN
      PRINT PIN17$
      GOTO 4177
END IF
P17$ = INPUT$(1)
PRINT P17$
IF P17$ <> "1" AND P17$ <> "0" THEN GOTO 4176
IF P17$ = "1" THEN Z17 = 0
IF P17$ = "0" THEN Z17 = 128
4177 LOCATE 19, 50
PRINT "           "
LOCATE 19, 50
IF UCASE$(PIN18$) = "+" OR UCASE$(PIN18$) = "-" THEN
      PRINT PIN18$
      GOTO 4186
END IF
P18$ = INPUT$(1)
PRINT P18$
IF P18$ <> "1" AND P18$ <> "0" THEN GOTO 4177
IF P18$ = "1" THEN Z18 = 0
IF P18$ = "0" THEN Z18 = 64
```

```
4178 LOCATE 20, 50
PRINT "          "
LOCATE 20, 50
IF UCASE$(PIN19$) = "+" OR UCASE$(PIN19$) = "-" THEN
      PRINT PIN19$
      GOTO 4179
END IF
P19$ = INPUT$(1)
PRINT P19$
IF P19$ <> "1" AND P19$ <> "0" THEN GOTO 4178
IF P19$ = "1" THEN Z19 = 0
IF P19$ = "0" THEN Z19 = 32
4179 LOCATE 21, 50
PRINT "          "
LOCATE 21, 50
IF UCASE$(PIN20$) = "+" OR UCASE$(PIN20$) = "-" THEN
      PRINT PIN20$
      GOTO 4186
END IF
P20$ = INPUT$(1)
PRINT P20$
IF P20$ <> "1" AND P20$ <> "0" THEN GOTO 4179
IF P20$ = "1" THEN Z20 = 0
IF P20$ = "0" THEN Z20 = 16
4186 LOCATE 22, 13
COLOR 4, 0
INPUT "Desea corregir (S/N)"; COR1$
IF UCASE$(COR1$) = "S" THEN GOTO 4160
END SUB

SUB ZOCALO14
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED ZOC1, ZOC2, ZOC3, ZOC4, ZOC5, ZOC6, ZOC7, ZOC8, ZOC9, ZOC10, ZOC11, ZOC12,
ZOC13, ZOC14, ZOC15, ZOC16, ZOC17, ZOC18, ZOC19, ZOC20
SHARED POL1, POL2, POL3, POL4, POL5, POL6, POL7, POL8, POL9, POL10, POL11, POL12, POL13,
POL14, POL15, POL16, POL17, POL18, POL19, POL20
SELECT CASE PIN1$
      CASE "E"
      ZOC4 = 0
      CASE "S"
      ZOC4 = 2
      CASE "+"
      ZOC4 = 0
      POL4 = 0
      CASE "-"
      ZOC4 = 0
      POL4 = 2
END SELECT
SELECT CASE PIN2$
      CASE "E"
      ZOC5 = 0
      CASE "S"
      ZOC5 = 4
      CASE "+"
      ZOC5 = 0
```

```
        POL5 = 0
        CASE "-"
        ZOC5 = 0
        POL5 = 4
    END SELECT
    IF UCASE$(PIN3$) = "E" THEN ZOC6 = 0
    IF UCASE$(PIN3$) = "S" THEN ZOC6 = 8
    IF UCASE$(PIN3$) = "+" THEN ZOC6 = 0
    IF UCASE$(PIN3$) = "+" THEN POL6 = 0
    IF UCASE$(PIN3$) = "-" THEN ZOC6 = 0
    IF UCASE$(PIN3$) = "-" THEN POL6 = 8
    IF UCASE$(PIN4$) = "E" THEN ZOC7 = 0
    IF UCASE$(PIN4$) = "S" THEN ZOC7 = 16
    IF UCASE$(PIN4$) = "+" THEN ZOC7 = 0
    IF UCASE$(PIN4$) = "+" THEN POL7 = 0
    IF UCASE$(PIN4$) = "-" THEN ZOC7 = 0
    IF UCASE$(PIN4$) = "-" THEN POL7 = 16
    IF UCASE$(PIN5$) = "E" THEN ZOC8 = 0
    IF UCASE$(PIN5$) = "S" THEN ZOC8 = 32
    IF UCASE$(PIN5$) = "+" THEN ZOC8 = 0
    IF UCASE$(PIN5$) = "+" THEN POL8 = 0
    IF UCASE$(PIN5$) = "-" THEN ZOC8 = 0
    IF UCASE$(PIN5$) = "-" THEN POL8 = 32
    IF UCASE$(PIN6$) = "E" THEN ZOC9 = 0
    IF UCASE$(PIN6$) = "S" THEN ZOC9 = 64
    IF UCASE$(PIN6$) = "+" THEN ZOC9 = 0
    IF UCASE$(PIN6$) = "+" THEN POL9 = 0
    IF UCASE$(PIN6$) = "-" THEN ZOC9 = 0
    IF UCASE$(PIN6$) = "-" THEN POL9 = 64
    IF UCASE$(PIN7$) = "E" THEN ZOC10 = 0
    IF UCASE$(PIN7$) = "S" THEN ZOC10 = 128
    IF UCASE$(PIN7$) = "+" THEN ZOC10 = 0
    IF UCASE$(PIN7$) = "+" THEN POL10 = 0
    IF UCASE$(PIN7$) = "-" THEN ZOC10 = 0
    IF UCASE$(PIN7$) = "-" THEN POL10 = 128
    IF UCASE$(PIN8$) = "E" THEN ZOC11 = 0
    IF UCASE$(PIN8$) = "S" THEN ZOC11 = 4
    IF UCASE$(PIN8$) = "+" THEN ZOC11 = 0
    IF UCASE$(PIN8$) = "+" THEN POL11 = 0
    IF UCASE$(PIN8$) = "-" THEN ZOC11 = 0
    IF UCASE$(PIN8$) = "-" THEN POL11 = 4
    IF UCASE$(PIN9$) = "E" THEN ZOC12 = 0
    IF UCASE$(PIN9$) = "S" THEN ZOC12 = 8
    IF UCASE$(PIN9$) = "+" THEN ZOC12 = 0
    IF UCASE$(PIN9$) = "+" THEN POL12 = 0
    IF UCASE$(PIN9$) = "-" THEN ZOC12 = 0
    IF UCASE$(PIN9$) = "-" THEN POL12 = 8
    IF UCASE$(PIN10$) = "E" THEN ZOC13 = 0
    IF UCASE$(PIN10$) = "S" THEN ZOC13 = 16
    IF UCASE$(PIN10$) = "+" THEN ZOC13 = 0
    IF UCASE$(PIN10$) = "+" THEN POL13 = 0
    IF UCASE$(PIN10$) = "-" THEN ZOC13 = 0
    IF UCASE$(PIN10$) = "-" THEN POL13 = 16
    IF UCASE$(PIN11$) = "E" THEN ZOC14 = 0
    IF UCASE$(PIN11$) = "S" THEN ZOC14 = 32
    IF UCASE$(PIN11$) = "+" THEN ZOC14 = 0
```

```
IF UCASE$(PIN11$) = "+" THEN POL14 = 0
IF UCASE$(PIN11$) = "-" THEN ZOC14 = 0
IF UCASE$(PIN11$) = "-" THEN POL14 = 32
IF UCASE$(PIN12$) = "E" THEN ZOC15 = 0
IF UCASE$(PIN12$) = "S" THEN ZOC15 = 64
IF UCASE$(PIN12$) = "+" THEN ZOC15 = 0
IF UCASE$(PIN12$) = "+" THEN POL15 = 0
IF UCASE$(PIN12$) = "-" THEN ZOC15 = 0
IF UCASE$(PIN12$) = "-" THEN POL15 = 64
IF UCASE$(PIN13$) = "E" THEN ZOC16 = 0
IF UCASE$(PIN13$) = "S" THEN ZOC16 = 128
IF UCASE$(PIN13$) = "+" THEN ZOC16 = 0
IF UCASE$(PIN13$) = "+" THEN POL16 = 0
IF UCASE$(PIN13$) = "-" THEN ZOC16 = 0
IF UCASE$(PIN13$) = "-" THEN POL16 = 128
IF UCASE$(PIN14$) = "E" THEN ZOC17 = 0
IF UCASE$(PIN14$) = "S" THEN ZOC17 = 128
IF UCASE$(PIN14$) = "+" THEN ZOC17 = 0
IF UCASE$(PIN14$) = "+" THEN POL17 = 0
IF UCASE$(PIN14$) = "-" THEN ZOC17 = 0
IF UCASE$(PIN14$) = "-" THEN POL17 = 128
CALL DATOS
END SUB

SUB ZOCALO16
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED ZOC1, ZOC2, ZOC3, ZOC4, ZOC5, ZOC6, ZOC7, ZOC8, ZOC9, ZOC10, ZOC11, ZOC12,
ZOC13, ZOC14, ZOC15, ZOC16, ZOC17, ZOC18, ZOC19, ZOC20
SHARED POL1, POL2, POL3, POL4, POL5, POL6, POL7, POL8, POL9, POL10, POL11, POL12, POL13,
POL14, POL15, POL16, POL17, POL18, POL19, POL20
IF UCASE$(PIN1$) = "E" THEN ZOC3 = 0
IF UCASE$(PIN1$) = "S" THEN ZOC3 = 1
IF UCASE$(PIN1$) = "+" THEN ZOC3 = 0: POL3 = 0
IF UCASE$(PIN1$) = "-" THEN ZOC3 = 0: POL3 = 1
IF UCASE$(PIN2$) = "E" THEN ZOC4 = 0
IF UCASE$(PIN2$) = "S" THEN ZOC4 = 2
IF UCASE$(PIN2$) = "+" THEN ZOC4 = 0: POL4 = 0
IF UCASE$(PIN2$) = "-" THEN ZOC4 = 0: POL4 = 2
IF UCASE$(PIN3$) = "E" THEN ZOC5 = 0
IF UCASE$(PIN3$) = "S" THEN ZOC5 = 4
IF UCASE$(PIN3$) = "+" THEN ZOC5 = 0: POL5 = 0
IF UCASE$(PIN3$) = "-" THEN ZOC5 = 0: POL5 = 4
IF UCASE$(PIN4$) = "E" THEN ZOC6 = 0
IF UCASE$(PIN4$) = "S" THEN ZOC6 = 8
IF UCASE$(PIN4$) = "+" THEN ZOC6 = 0: POL6 = 0
IF UCASE$(PIN4$) = "-" THEN ZOC6 = 0: POL6 = 8
IF UCASE$(PIN5$) = "E" THEN ZOC7 = 0
IF UCASE$(PIN5$) = "S" THEN ZOC7 = 16
IF UCASE$(PIN5$) = "+" THEN ZOC7 = 0: POL7 = 0
IF UCASE$(PIN5$) = "-" THEN ZOC7 = 0: POL7 = 16
IF UCASE$(PIN6$) = "E" THEN ZOC8 = 0
IF UCASE$(PIN6$) = "S" THEN ZOC8 = 32
IF UCASE$(PIN6$) = "+" THEN ZOC8 = 0: POL8 = 0
IF UCASE$(PIN6$) = "-" THEN ZOC8 = 0: POL8 = 32
IF UCASE$(PIN7$) = "E" THEN ZOC9 = 0
```

```
IF UCASE$(PIN7$) = "S" THEN ZOC9 = 64
IF UCASE$(PIN7$) = "+" THEN ZOC9 = 0: POL9 = 0
IF UCASE$(PIN7$) = "-" THEN ZOC9 = 0: POL9 = 64
IF UCASE$(PIN8$) = "E" THEN ZOC10 = 0
IF UCASE$(PIN8$) = "S" THEN ZOC10 = 128
IF UCASE$(PIN8$) = "+" THEN ZOC10 = 0: POL10 = 0
IF UCASE$(PIN8$) = "-" THEN ZOC10 = 0: POL10 = 128
IF UCASE$(PIN9$) = "E" THEN ZOC11 = 0
IF UCASE$(PIN9$) = "S" THEN ZOC11 = 4
IF UCASE$(PIN9$) = "+" THEN ZOC11 = 0: POL11 = 0
IF UCASE$(PIN9$) = "-" THEN ZOC11 = 0: POL11 = 4
IF UCASE$(PIN10$) = "E" THEN ZOC12 = 0
IF UCASE$(PIN10$) = "S" THEN ZOC12 = 8
IF UCASE$(PIN10$) = "+" THEN ZOC12 = 0: POL12 = 0
IF UCASE$(PIN10$) = "-" THEN ZOC12 = 0: POL12 = 8
IF UCASE$(PIN11$) = "E" THEN ZOC13 = 0
IF UCASE$(PIN11$) = "S" THEN ZOC13 = 16
IF UCASE$(PIN11$) = "+" THEN ZOC13 = 0: POL13 = 0
IF UCASE$(PIN11$) = "-" THEN ZOC13 = 0: POL13 = 16
IF UCASE$(PIN12$) = "E" THEN ZOC14 = 0
IF UCASE$(PIN12$) = "S" THEN ZOC14 = 32
IF UCASE$(PIN12$) = "+" THEN ZOC14 = 0: POL14 = 0
IF UCASE$(PIN12$) = "-" THEN ZOC14 = 0: POL14 = 32
IF UCASE$(PIN13$) = "E" THEN ZOC15 = 0
IF UCASE$(PIN13$) = "S" THEN ZOC15 = 64
IF UCASE$(PIN13$) = "+" THEN ZOC15 = 0: POL15 = 0
IF UCASE$(PIN13$) = "-" THEN ZOC15 = 0: POL15 = 64
IF UCASE$(PIN14$) = "E" THEN ZOC16 = 0
IF UCASE$(PIN14$) = "S" THEN ZOC16 = 128
IF UCASE$(PIN14$) = "+" THEN ZOC16 = 0: POL16 = 0
IF UCASE$(PIN14$) = "-" THEN ZOC16 = 0: POL16 = 128
IF UCASE$(PIN15$) = "E" THEN ZOC17 = 0
IF UCASE$(PIN15$) = "S" THEN ZOC17 = 128
IF UCASE$(PIN15$) = "+" THEN ZOC17 = 0: POL17 = 0
IF UCASE$(PIN15$) = "-" THEN ZOC17 = 0: POL17 = 128
IF UCASE$(PIN16$) = "E" THEN ZOC18 = 0
IF UCASE$(PIN16$) = "S" THEN ZOC18 = 64
IF UCASE$(PIN16$) = "+" THEN ZOC18 = 0: POL18 = 0
IF UCASE$(PIN16$) = "-" THEN ZOC18 = 0: POL18 = 64
CALL DATOS
END SUB

SUB ZOCALO18
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED ZOC1, ZOC2, ZOC3, ZOC4, ZOC5, ZOC6, ZOC7, ZOC8, ZOC9, ZOC10, ZOC11, ZOC12,
ZOC13, ZOC14, ZOC15, ZOC16, ZOC17, ZOC18, ZOC19, ZOC20
SHARED POL1, POL2, POL3, POL4, POL5, POL6, POL7, POL8, POL9, POL10, POL11, POL12, POL13,
POL14, POL15, POL16, POL17, POL18, POL19, POL20
IF UCASE$(PIN1$) = "E" THEN ZOC2 = 0
IF UCASE$(PIN1$) = "S" THEN ZOC2 = 1
IF UCASE$(PIN1$) = "+" THEN ZOC2 = 0: POL2 = 0
IF UCASE$(PIN1$) = "-" THEN ZOC2 = 0: POL2 = 1
IF UCASE$(PIN2$) = "E" THEN ZOC3 = 0
IF UCASE$(PIN2$) = "S" THEN ZOC3 = 1
IF UCASE$(PIN2$) = "+" THEN ZOC3 = 0: POL3 = 0
```

```
IF UCASE$(PIN2$) = "-" THEN ZOC3 = 0: POL3 = 1
IF UCASE$(PIN3$) = "E" THEN ZOC4 = 0
IF UCASE$(PIN3$) = "S" THEN ZOC4 = 2
IF UCASE$(PIN3$) = "+" THEN ZOC4 = 0: POL4 = 0
IF UCASE$(PIN3$) = "-" THEN ZOC4 = 0: POL4 = 2
IF UCASE$(PIN4$) = "E" THEN ZOC5 = 0
IF UCASE$(PIN4$) = "S" THEN ZOC5 = 4
IF UCASE$(PIN4$) = "+" THEN ZOC5 = 0: POL5 = 0
IF UCASE$(PIN4$) = "-" THEN ZOC5 = 0: POL5 = 4
IF UCASE$(PIN5$) = "E" THEN ZOC6 = 0
IF UCASE$(PIN5$) = "S" THEN ZOC6 = 8
IF UCASE$(PIN5$) = "+" THEN ZOC6 = 0: POL6 = 0
IF UCASE$(PIN5$) = "-" THEN ZOC6 = 0: POL6 = 8
IF UCASE$(PIN6$) = "E" THEN ZOC7 = 0
IF UCASE$(PIN6$) = "S" THEN ZOC7 = 16
IF UCASE$(PIN6$) = "+" THEN ZOC7 = 0: POL7 = 0
IF UCASE$(PIN6$) = "-" THEN ZOC7 = 0: POL7 = 16
IF UCASE$(PIN7$) = "E" THEN ZOC8 = 0
IF UCASE$(PIN7$) = "S" THEN ZOC8 = 32
IF UCASE$(PIN7$) = "+" THEN ZOC8 = 0: POL8 = 0
IF UCASE$(PIN7$) = "-" THEN ZOC8 = 0: POL8 = 32
IF UCASE$(PIN8$) = "E" THEN ZOC9 = 0
IF UCASE$(PIN8$) = "S" THEN ZOC9 = 64
IF UCASE$(PIN8$) = "+" THEN ZOC9 = 0: POL9 = 0
IF UCASE$(PIN8$) = "-" THEN ZOC9 = 0: POL9 = 64
IF UCASE$(PIN9$) = "E" THEN ZOC10 = 0
IF UCASE$(PIN9$) = "S" THEN ZOC10 = 128
IF UCASE$(PIN9$) = "+" THEN ZOC10 = 0: POL10 = 0
IF UCASE$(PIN9$) = "-" THEN ZOC10 = 0: POL10 = 128
IF UCASE$(PIN10$) = "E" THEN ZOC11 = 0
IF UCASE$(PIN10$) = "S" THEN ZOC11 = 4
IF UCASE$(PIN10$) = "+" THEN ZOC11 = 0: POL11 = 0
IF UCASE$(PIN10$) = "-" THEN ZOC11 = 0: POL11 = 4
IF UCASE$(PIN11$) = "E" THEN ZOC12 = 0
IF UCASE$(PIN11$) = "S" THEN ZOC12 = 8
IF UCASE$(PIN11$) = "+" THEN ZOC12 = 0: POL12 = 0
IF UCASE$(PIN11$) = "-" THEN ZOC12 = 0: POL12 = 8
IF UCASE$(PIN12$) = "E" THEN ZOC13 = 0
IF UCASE$(PIN12$) = "S" THEN ZOC13 = 16
IF UCASE$(PIN12$) = "+" THEN ZOC13 = 0: POL13 = 0
IF UCASE$(PIN12$) = "-" THEN ZOC13 = 0: POL13 = 16
IF UCASE$(PIN13$) = "E" THEN ZOC14 = 0
IF UCASE$(PIN13$) = "S" THEN ZOC14 = 32
IF UCASE$(PIN13$) = "+" THEN ZOC14 = 0: POL14 = 0
IF UCASE$(PIN13$) = "-" THEN ZOC14 = 0: POL14 = 32
IF UCASE$(PIN14$) = "E" THEN ZOC15 = 0
IF UCASE$(PIN14$) = "S" THEN ZOC15 = 64
IF UCASE$(PIN14$) = "+" THEN ZOC15 = 0: POL15 = 0
IF UCASE$(PIN14$) = "-" THEN ZOC15 = 0: POL15 = 64
IF UCASE$(PIN15$) = "E" THEN ZOC16 = 0
IF UCASE$(PIN15$) = "S" THEN ZOC16 = 128
IF UCASE$(PIN15$) = "+" THEN ZOC16 = 0: POL16 = 0
IF UCASE$(PIN15$) = "-" THEN ZOC16 = 0: POL16 = 128
IF UCASE$(PIN16$) = "E" THEN ZOC17 = 0
IF UCASE$(PIN16$) = "S" THEN ZOC17 = 128
IF UCASE$(PIN16$) = "+" THEN ZOC17 = 0: POL17 = 0
```

```
IF UCASE$(PIN16$) = "-" THEN ZOC17 = 0: POL17 = 128
IF UCASE$(PIN17$) = "E" THEN ZOC18 = 0
IF UCASE$(PIN17$) = "S" THEN ZOC18 = 64
IF UCASE$(PIN17$) = "+" THEN ZOC18 = 0: POL18 = 0
IF UCASE$(PIN17$) = "-" THEN ZOC18 = 0: POL18 = 64
IF UCASE$(PIN18$) = "E" THEN ZOC19 = 0
IF UCASE$(PIN18$) = "S" THEN ZOC19 = 32
IF UCASE$(PIN18$) = "+" THEN ZOC19 = 0: POL19 = 0
IF UCASE$(PIN18$) = "-" THEN ZOC19 = 0: POL19 = 32
CALL DATOS
END SUB

SUB ZOCALO20
SHARED PIN1$, PIN2$, PIN3$, PIN4$, PIN5$, PIN6$, PIN7$, PIN8$, PIN9$, PIN10$, PIN11$, PIN12$,
PIN13$, PIN14$, PIN15$, PIN16$, PIN17$, PIN18$, PIN19$, PIN20$
SHARED ZOC1, ZOC2, ZOC3, ZOC4, ZOC5, ZOC6, ZOC7, ZOC8, ZOC9, ZOC10, ZOC11, ZOC12,
ZOC13, ZOC14, ZOC15, ZOC16, ZOC17, ZOC18, ZOC19, ZOC20
SHARED POL1, POL2, POL3, POL4, POL5, POL6, POL7, POL8, POL9, POL10, POL11, POL12, POL13,
POL14, POL15, POL16, POL17, POL18, POL19, POL20
IF UCASE$(PIN1$) = "E" THEN ZOC1 = 0
IF UCASE$(PIN1$) = "S" THEN ZOC1 = 2
IF UCASE$(PIN1$) = "+" THEN ZOC1 = 0: POL1 = 0
IF UCASE$(PIN1$) = "-" THEN ZOC1 = 0: POL1 = 2
IF UCASE$(PIN2$) = "E" THEN ZOC2 = 0
IF UCASE$(PIN2$) = "S" THEN ZOC2 = 1
IF UCASE$(PIN2$) = "+" THEN ZOC2 = 0: POL2 = 0
IF UCASE$(PIN2$) = "-" THEN ZOC2 = 0: POL2 = 1
IF UCASE$(PIN3$) = "E" THEN ZOC3 = 0
IF UCASE$(PIN3$) = "S" THEN ZOC3 = 1
IF UCASE$(PIN3$) = "+" THEN ZOC3 = 0: POL3 = 0
IF UCASE$(PIN3$) = "-" THEN ZOC3 = 0: POL3 = 1
IF UCASE$(PIN4$) = "E" THEN ZOC4 = 0
IF UCASE$(PIN4$) = "S" THEN ZOC4 = 2
IF UCASE$(PIN4$) = "+" THEN ZOC4 = 0: POL4 = 0
IF UCASE$(PIN4$) = "-" THEN ZOC4 = 0: POL4 = 2
IF UCASE$(PIN5$) = "E" THEN ZOC5 = 0
IF UCASE$(PIN5$) = "S" THEN ZOC5 = 4
IF UCASE$(PIN5$) = "+" THEN ZOC5 = 0: POL5 = 0
IF UCASE$(PIN5$) = "-" THEN ZOC5 = 0: POL5 = 4
IF UCASE$(PIN6$) = "E" THEN ZOC6 = 0
IF UCASE$(PIN6$) = "S" THEN ZOC6 = 8
IF UCASE$(PIN6$) = "+" THEN ZOC6 = 0: POL6 = 0
IF UCASE$(PIN6$) = "-" THEN ZOC6 = 0: POL6 = 8
IF UCASE$(PIN7$) = "E" THEN ZOC7 = 0
IF UCASE$(PIN7$) = "S" THEN ZOC7 = 16
IF UCASE$(PIN7$) = "+" THEN ZOC7 = 0: POL7 = 0
IF UCASE$(PIN7$) = "-" THEN ZOC7 = 0: POL7 = 16
IF UCASE$(PIN8$) = "E" THEN ZOC8 = 0
IF UCASE$(PIN8$) = "S" THEN ZOC8 = 32
IF UCASE$(PIN8$) = "+" THEN ZOC8 = 0: POL8 = 0
IF UCASE$(PIN8$) = "-" THEN ZOC8 = 0: POL8 = 32
IF UCASE$(PIN9$) = "E" THEN ZOC9 = 0
IF UCASE$(PIN9$) = "S" THEN ZOC9 = 64
IF UCASE$(PIN9$) = "+" THEN ZOC9 = 0: POL9 = 0
IF UCASE$(PIN9$) = "-" THEN ZOC9 = 0: POL9 = 64
IF UCASE$(PIN10$) = "E" THEN ZOC10 = 0
```

```
IF UCASE$(PIN10$) = "S" THEN ZOC10 = 128
IF UCASE$(PIN10$) = "+" THEN ZOC10 = 0: POL10 = 0
IF UCASE$(PIN10$) = "-" THEN ZOC10 = 0: POL10 = 128
IF UCASE$(PIN11$) = "E" THEN ZOC11 = 0
IF UCASE$(PIN11$) = "S" THEN ZOC11 = 4
IF UCASE$(PIN11$) = "+" THEN ZOC11 = 0: POL11 = 0
IF UCASE$(PIN11$) = "-" THEN ZOC11 = 0: POL11 = 4
IF UCASE$(PIN12$) = "E" THEN ZOC12 = 0
IF UCASE$(PIN12$) = "S" THEN ZOC12 = 8
IF UCASE$(PIN12$) = "+" THEN ZOC12 = 0: POL12 = 0
IF UCASE$(PIN12$) = "-" THEN ZOC12 = 0: POL12 = 8
IF UCASE$(PIN13$) = "E" THEN ZOC13 = 0
IF UCASE$(PIN13$) = "S" THEN ZOC13 = 16
IF UCASE$(PIN13$) = "+" THEN ZOC13 = 0: POL13 = 0
IF UCASE$(PIN13$) = "-" THEN ZOC13 = 0: POL13 = 16
IF UCASE$(PIN14$) = "E" THEN ZOC14 = 0
IF UCASE$(PIN14$) = "S" THEN ZOC14 = 32
IF UCASE$(PIN14$) = "+" THEN ZOC14 = 0: POL14 = 0
IF UCASE$(PIN14$) = "-" THEN ZOC14 = 0: POL14 = 32
IF UCASE$(PIN15$) = "E" THEN ZOC15 = 0
IF UCASE$(PIN15$) = "S" THEN ZOC15 = 64
IF UCASE$(PIN15$) = "+" THEN ZOC15 = 0: POL15 = 0
IF UCASE$(PIN15$) = "-" THEN ZOC15 = 0: POL15 = 64
IF UCASE$(PIN16$) = "E" THEN ZOC16 = 0
IF UCASE$(PIN16$) = "S" THEN ZOC16 = 128
IF UCASE$(PIN16$) = "+" THEN ZOC16 = 0: POL16 = 0
IF UCASE$(PIN16$) = "-" THEN ZOC16 = 0: POL16 = 128
IF UCASE$(PIN17$) = "E" THEN ZOC17 = 0
IF UCASE$(PIN17$) = "S" THEN ZOC17 = 128
IF UCASE$(PIN17$) = "+" THEN ZOC17 = 0: POL17 = 0
IF UCASE$(PIN17$) = "-" THEN ZOC17 = 0: POL17 = 128
IF UCASE$(PIN18$) = "E" THEN ZOC18 = 0
IF UCASE$(PIN18$) = "S" THEN ZOC18 = 64
IF UCASE$(PIN18$) = "+" THEN ZOC18 = 0: POL18 = 0
IF UCASE$(PIN18$) = "-" THEN ZOC18 = 0: POL18 = 64
IF UCASE$(PIN19$) = "E" THEN ZOC19 = 0
IF UCASE$(PIN19$) = "S" THEN ZOC19 = 32
IF UCASE$(PIN19$) = "+" THEN ZOC19 = 0: POL19 = 0
IF UCASE$(PIN19$) = "-" THEN ZOC19 = 0: POL19 = 32
IF UCASE$(PIN20$) = "E" THEN ZOC20 = 0
IF UCASE$(PIN20$) = "S" THEN ZOC20 = 16
IF UCASE$(PIN20$) = "+" THEN ZOC20 = 0: POL20 = 0
IF UCASE$(PIN20$) = "-" THEN ZOC20 = 0: POL20 = 16
CALL DATOS
END SUB
```

4.3.2.- PROGRAMA DE PRUEBAS - CMOS.EXE

```
DECLARE SUB MAIN ()
DECLARE SUB TXRX ()
DECLARE SUB PRUEBAS ()
DECLARE SUB PRUEBA ()
DECLARE SUB CUAD3 (Y1!, X1!, Y2!, X2!, M1!, M2!, M3!, M4!, M5!, M6!, A!, B!)
COMMON SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2 _
, P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2, P15P0,
P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
COMMON TX0, TX1, TX2, RX0, RX1, RX2
COMMON EE
COMMON AB$
COMMON REP$
CLS
CLEAR
CALL MAIN
END

SUB CUAD3 (Y1, X1, Y2, X2, M1, M2, M3, M4, M5, M6, A, B)
H = X2 - X1
V = Y2 - Y1
COLOR A, B
LOCATE Y1, X1
FOR Z = 1 TO (X2 - X1)
LOCATE Y1, X1 + Z
PRINT CHR$(M1);
NEXT Z
LOCATE Y2, X1
FOR Z = 1 TO H
PRINT CHR$(M1);
NEXT Z
FOR Z = 1 TO (V - 1)
LOCATE Y1 + Z, X1
PRINT CHR$(M2);
LOCATE Y1 + Z, X2
PRINT CHR$(M2)
NEXT Z
LOCATE Y1, X1
PRINT CHR$(M3)
LOCATE Y2, X2
PRINT CHR$(M4)
LOCATE Y1, X1 + H
PRINT CHR$(M5)
LOCATE Y1 + V, X1
PRINT CHR$(M6)
END SUB

35 SUB MAIN
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2 _
, P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2, P15P0,
P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
```

```
SHARED TX0, TX1, TX2, RX0, RX1, RX2
SHARED EESHARED ABS
SHARED REPS
CLS
CLEAR
CLOSE
EE = 0
LOCATE 1, 1
FOR M = 1 TO 24
      COLOR 0, 1
      PRINT "                                                    "
NEXT M
CALL CUAD3(8, 15, 14, 65, 196, 179, 218, 217, 191, 192, 11, 0)
COLOR 11, 0
LOCATE 9, 16
PRINT "                                  "
LOCATE 10, 16
PRINT "      PROBADOR DE CIRCUITOS INTEGRADOS      "
LOCATE 11, 16
PRINT "                                  "
LOCATE 12, 16
PRINT "           DE LA FAMILIA CMOS           "
LOCATE 13, 16
PRINT "                                  "
LOCATE 23, 40
COLOR 7, 0
PRINT "(Presione cualquier tecla)"
WHILE INKEY$ = ""
WEND
29 CLEAR
LOCATE 22, 13
COLOR 0, 1
PRINT "                          "
LOCATE 23, 40
PRINT "                          "
COLOR 11, 0
LOCATE 10, 16
PRINT "        INGRESE LA NUMERACION DEL          "
LOCATE 11, 16
PRINT "                                  "
LOCATE 12, 16
PRINT "        CIRCUITO INTEGRADO: CI #          "
LOCATE 12, 52
INPUT " ", ABS
IF ABS = "PRUEBA" THEN CALL PRUEBA
OPEN "A:\CI.LIB" FOR INPUT AS #1
DO UNTIL EOF(1)
INPUT #1, SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, _
P10P2, P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2, P15P0,
P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
IF ABS = SERIE$ THEN GOTO 2
IF EOF(1) THEN GOTO CINOEXIST
LOOP
CINOEXIST: COLOR 11, 0
```

```
LOCATE 10, 16
PRINT "                                    "
LOCATE 11, 16PRINT "       CI #     NO CONSTA EN LIBRERIA        "
LOCATE 11, 30
PRINT AB$
LOCATE 12, 16
PRINT "                                    "
LOCATE 22, 13
COLOR 4, 0
CLOSE #1
INPUT "Desea intentar otra serie (S/N)"; SER$
IF UCASE$(SER$) = "S" THEN GOTO 29
IF UCASE$(SER$) <> "S" THEN GOTO FIN2
2 LOCATE 22, 13
COLOR 0, 1
PRINT "                                "
LOCATE 23, 40
COLOR 0, 1
PRINT "                                "
3 COLOR 11, 0
LOCATE 10, 16
PRINT "    CI #    :                   "
LOCATE 10, 26
PRINT SERIE$
LOCATE 10, 33
PRINT DESCRIP$
LOCATE 12, 16
PRINT "    INSERTE EL CHIP Y PRESIONE <<ENTER>>       "
4 IF INKEY$ = "" THEN 4
CALL PRUEBAS
IF REP$ = "A" THEN GOTO 35
FIN2: COLOR 7, 0
CLS
END

END SUB

SUB PRUEBA
SHARED TX0, TX1, TX2, RX0, RX1, RX2
VOLPOL = 4
VOLPOLP$ = "+5V"
VOLPOLN$ = "GND"
12 TX1 = 2 + VOLPOL
  TX2 = 0
  TX3 = 0
  TX4 = 1 + 2 + VOLPOL
  TX5 = 0
  TX6 = 0
  RX0 = RX1 = RX2 = 0
LOCATE 1, 1
FOR M = 1 TO 24
     COLOR 0, 1
     PRINT "                                                    "
NEXT M
CALL CUAD3(8, 15, 14, 65, 196, 179, 218, 217, 191, 192, 11, 0)
LOCATE 22, 13
```

```
COLOR 0, 1
PRINT "                                "
LOCATE 23, 40
COLOR 0, 1
PRINT "                                "
COLOR 11, 0
LOCATE 9, 16
PRINT "                                        "
LOCATE 10, 16
PRINT "       VERIFICAR LA PRESENCIA DE            "
LOCATE 10, 53
PRINT VOLPOLP$
LOCATE 11, 16
PRINT "                                        "
LOCATE 12, 16
PRINT "       EN TODOS LOS PINES DEL ZOCALO        "
LOCATE 13, 16
PRINT "                                        "
    TX0 = TX1
    TX1 = TX2
    TX2 = TX3
CALL TXRX
    TX0 = TX4
    TX1 = TX5
    TX2 = TX6
CALL TXRX
SOUND 1300, 1
7 IF INKEY$ = "" THEN 7
LOCATE 10, 16
PRINT "       VERIFICAR LA PRESENCIA DE            "
LOCATE 10, 53
PRINT VOLPOLN$
    TX0 = 243 + VOLPOL
    TX1 = 255
    TX2 = 255
CALL TXRX
SOUND 1300, 1
14 IF INKEY$ = "" THEN 14
CALL CUAD3(8, 15, 14, 65, 196, 179, 218, 217, 191, 192, 11, 0)
COLOR 11, 0
LOCATE 9, 16
PRINT "                                        "
LOCATE 10, 16
PRINT "       PRUEBA OK? : DESEA INTENTAR          "
LOCATE 11, 16
PRINT "                                        "
LOCATE 12, 16
PRINT "       CON OTRA POLARIZACION? (S/N)         "
LOCATE 13, 16
PRINT "                                        "
SER$ = INPUT$(1)
IF UCASE$(SER$) = "S" THEN GOTO 19
IF UCASE$(SER$) <> "S" THEN GOTO FIN1
19 COLOR 11, 0
LOCATE 9, 16
PRINT "                                        "
```

```
LOCATE 10, 16
PRINT "      ESCOJA LA POLARIZACION DESEADA:      "
LOCATE 11, 16
PRINT "                                          "
LOCATE 12, 16
PRINT "      A: +5V/-5V  B:+9V/-9V  C:+9V/GND    "
LOCATE 13, 16
PRINT "                                          "
V1$ = INPUT$(1)
SELECT CASE V1$
     CASE "A"
     VOLPOL = 12
     VOLPOLP$ = "+5V"
     VOLPOLN$ = "-5V"
     CASE "B"
     VOLPOL = 8
     VOLPOLP$ = "+9V"
     VOLPOLN$ = "-9V"
     CASE "C"
     VOLPOL = 0
     VOLPOLP$ = "+9V"
     VOLPOLN$ = "GND"
END SELECT
GOTO 12
FIN1: COLOR 7, 0
CLS
END
END SUB

SUB PRUEBAS
SHARED SERIE$, DESCRIP$, ZOCP0, ZOCP1, ZOCP2, POLP0, POLP1, POLP2, P1P0, P1P1, P1P2,
P2P0, P2P1, P2P2, P3P0, P3P1, P3P2, P4P0, P4P1, P4P2, P5P0, P5P1, P5P2, P6P0, P6P1, P6P2, P7P0,
P7P1, P7P2, P8P0, P8P1, P8P2, P9P0, P9P1, P9P2, P10P0, P10P1, P10P2 _
, P11P0, P11P1, P11P2, P12P0, P12P1, P12P2, P13P0, P13P1, P13P2, P14P0, P14P1, P14P2, P15P0,
P15P1, P15P2, P16P0, P16P1, P16P2, P17P0, P17P1, P17P2, P18P0, P18P1, P18P2
SHARED TX0, TX1, TX2, RX0, RX1, RX2
SHARED REP$
VOLPOL = 4
VOLPOL$ = "+5V/GND"
PP0 = POLP0 - 2
PP1 = POLP1
PP2 = POLP2
10 LOCATE 1, 1
FOR M = 1 TO 24
     COLOR 0, 1
     PRINT "                                                  "
NEXT M
CALL CUAD3(8, 15, 14, 65, 196, 179, 218, 217, 191, 192, 11, 0)
LOCATE 22, 13
COLOR 0, 1
PRINT "                          "
LOCATE 23, 40
COLOR 0, 1
PRINT "                          "
COLOR 11, 0
LOCATE 9, 16
```

```
PRINT "                                    "
LOCATE 10, 16
PRINT "    CI #     :                      "
LOCATE 10, 26
PRINT SERIE$
LOCATE 10, 33
PRINT DESCRIP$
LOCATE 11, 16
PRINT "                                    "
LOCATE 13, 16
PRINT "                                    "
LOCATE 12, 16
COLOR 4, 0
PRINT "    Prueba de CI en progreso ...            "
LOCATE 12, 51
PRINT VOLPOL$
ES:    TX0 = ZOCP0
       TX1 = ZOCP1
       TX2 = ZOCP2
       RX0 = 0
CALL TXRX
TX0 = TX0 + 1
CALL TXRX
DATOSPOL: TX0 = POLP0 + VOLPOL + 1
        TX1 = POLP1
        TX2 = POLP2
        RX0 = 0
CALL TXRX
DATOSPRUEBA:
      TX0 = P1P0 + PP0 + VOLPOL + 1
      TX1 = P1P1 + PP1
      TX2 = P1P2 + PP2
      RX0 = TX0
      RX1 = TX1
      RX2 = TX2
CALL TXRX
      IF P2P0 = 0 AND P2P1 = 0 AND P2P2 = 0 THEN GOTO VOLTS
      TX0 = P2P0 + PP0 + VOLPOL + 1
      TX1 = P2P1 + PP1
      TX2 = P2P2 + PP2
      RX0 = TX0
      RX1 = TX1
      RX2 = TX2
CALL TXRX
      IF P3P0 = 0 AND P3P1 = 0 AND P3P2 = 0 THEN GOTO VOLTS
      TX0 = P3P0 + PP0 + VOLPOL + 1
      TX1 = P3P1 + PP1
      TX2 = P3P2 + PP2
      RX0 = TX0
      RX1 = TX1
      RX2 = TX2
CALL TXRX
      IF P4P0 = 0 AND P4P1 = 0 AND P4P2 = 0 THEN GOTO VOLTS
      TX0 = P4P0 + PP0 + VOLPOL + 1
      TX1 = P4P1 + PP1
      TX2 = P4P2 + PP2
```

```
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P5P0 = 0 AND P5P1 = 0 AND P5P2 = 0 THEN GOTO VOLTS
    TX0 = P5P0 + PP0 + VOLPOL + 1
    TX1 = P5P1 + PP1
    TX2 = P5P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P6P0 = 0 AND P6P1 = 0 AND P6P2 = 0 THEN GOTO VOLTS
    TX0 = P6P0 + PP0 + VOLPOL + 1
    TX1 = P6P1 + PP1
    TX2 = P6P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P7P0 = 0 AND P7P1 = 0 AND P7P2 = 0 THEN GOTO VOLTS
    TX0 = P7P0 + PP0 + VOLPOL + 1
    TX1 = P7P1 + PP1
    TX2 = P7P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P8P0 = 0 AND P8P1 = 0 AND P8P2 = 0 THEN GOTO VOLTS
    TX0 = P8P0 + PP0 + VOLPOL + 1
    TX1 = P8P1 + PP1
    TX2 = P8P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P9P0 = 0 AND P9P1 = 0 AND P9P2 = 0 THEN GOTO VOLTS
    TX0 = P9P0 + PP0 + VOLPOL + 1
    TX1 = P9P1 + PP1
    TX2 = P9P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P10P0 = 0 AND P10P1 = 0 AND P10P2 = 0 THEN GOTO VOLTS
    TX0 = P10P0 + PP0 + VOLPOL + 1
    TX1 = P10P1 + PP1
    TX2 = P10P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P11P0 = 0 AND P11P1 = 0 AND P11P2 = 0 THEN GOTO VOLTS
    TX0 = P11P0 + PP0 + VOLPOL + 1
    TX1 = P11P1 + PP1
    TX2 = P11P2 + PP2
```

```
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P12P0 = 0 AND P12P1 = 0 AND P12P2 = 0 THEN GOTO VOLTS
    TX0 = P12P0 + PP0 + VOLPOL + 1
    TX1 = P12P1 + PP1
    TX2 = P12P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P13P0 = 0 AND P13P1 = 0 AND P13P2 = 0 THEN GOTO VOLTS
    TX0 = P13P0 + PP0 + VOLPOL + 1
    TX1 = P13P1 + PP1
    TX2 = P13P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P14P0 = 0 AND P14P1 = 0 AND P14P2 = 0 THEN GOTO VOLTS
    TX0 = P14P0 + PP0 + VOLPOL + 1
    TX1 = P14P1 + PP1
    TX2 = P14P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P15P0 = 0 AND P15P1 = 0 AND P15P2 = 0 THEN GOTO VOLTS
    TX0 = P15P0 + PP0 + VOLPOL + 1
    TX1 = P15P1 + PP1
    TX2 = P15P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P16P0 = 0 AND P16P1 = 0 AND P16P2 = 0 THEN GOTO VOLTS
    TX0 = P16P0 + PP0 + VOLPOL + 1
    TX1 = P16P1 + PP1
    TX2 = P16P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P17P0 = 0 AND P17P1 = 0 AND P17P2 = 0 THEN GOTO VOLTS
    TX0 = P17P0 + PP0 + VOLPOL + 1
    TX1 = P17P1 + PP1
    TX2 = P17P2 + PP2
    RX0 = TX0
    RX1 = TX1
    RX2 = TX2
CALL TXRX
    IF P18P0 = 0 AND P18P1 = 0 AND P18P2 = 0 THEN GOTO VOLTS
    TX0 = P18P0 + PP0 + VOLPOL + 1
    TX1 = P18P1 + PP1
    TX2 = P18P2 + PP2
```

```
        RX0 = TX0
        RX1 = TX1
        RX2 = TX2
 CALL TXRX
 VOLTS: CALL CUAD3(8, 15, 14, 65, 196, 179, 218, 217, 191, 192, 11, 0)
 COLOR 11, 0
 LOCATE 9, 16
 PRINT "                                    "
 LOCATE 10, 16
 PRINT "         PRUEBA OK : DESEA INTENTAR          "
 LOCATE 11, 16
 PRINT "                                    "
 LOCATE 12, 16
 PRINT "         CON OTRA POLARIZACION? (S/N)        "
 LOCATE 13, 16
 PRINT "                                    "
 SOUND 800, 1
 SOUND 1300, 1
 SER$ = INPUT$(1)
 IF UCASE$(SER$) = "S" THEN GOTO 9
 IF UCASE$(SER$) <> "S" THEN GOTO FIN
 9 COLOR 11, 0
 LOCATE 9, 16
 PRINT "                                    "
 LOCATE 10, 16
 PRINT "       ESCOJA LA POLARIZACION DESEADA:       "
 LOCATE 11, 16
 PRINT "                                    "
 LOCATE 12, 16
 PRINT "       A: +5V/-5V  B:+9V/-9V  C:+9V/GND      "
 LOCATE 13, 16
 PRINT "                                    "
 V1$ = INPUT$(1)
 IF UCASE$(V1$) = "A" THEN VOLPOL = 12 AND VOLPOL$ = "+5V/-5V"
 IF UCASE$(V1$) = "B" THEN VOLPOL = 8 AND VOLPOL$ = "+9V/-9V"
 IF UCASE$(V1$) = "C" THEN VOLPOL = 0 AND VOLPOL$ = "+9V/GND"
 LOCATE 12, 16
 PRINT "                                    "
 LOCATE 10, 16
 PRINT "    PRESIONE EL BOTON <<RESET>> Y LUEGO      "
 LOCATE 12, 16
 PRINT "    PRESIONE <<ENTER>> PARA INICIAR PRUEBA    "
 5 IF INKEY$ = "" THEN 5
 GOTO 10
 FIN: LOCATE 9, 16
 PRINT "                                    "
 LOCATE 10, 16
 PRINT "                                    "
 LOCATE 11, 16
 PRINT "  DESEA PROBAR OTRO CIRCUITO INTEGRADO? (S/N)  "
 LOCATE 12, 16
 PRINT "                                    "
 LOCATE 13, 16
 PRINT "                                    "
 SER1$ = INPUT$(1)
 IF UCASE$(SER1$) = "S" THEN GOTO 34
```

```
IF UCASE$(SER1$) <> "S" THEN GOTO FIN3
FIN3: COLOR 7, 0
CLS
END
34 REP$ = "A"
END SUB

SUB TXRX
SHARED TX0, TX1, TX2, RX0, RX1, RX2
SHARED SERIE$, DESCRIP$
SHARED AB$
SHARED EE
CLOSE
OPEN "COM1:9600,N,8,1,CD,CS,DS,RS" FOR RANDOM AS #1
     PRINT #1, CHR$(0);
     FOR K = 1 TO 100
     NEXT K
     TEMPO = 0
STCK:  IF LOC(1) <> 0 THEN GOTO STOK
     TEMPO = TEMPO + 1
     IF TEMPO >= 3000 THEN GOTO ECOM
     GOTO STCK
STOK:  ST = ASC(INPUT$(1, 1))
     IF ST = 0 THEN GOTO TXDAT0
     IF ST <> 0 THEN GOTO ECOM
TXDAT0: PRINT #1, CHR$(TX0);
     FOR K = 1 TO 100
     NEXT K
     TEMPO = 0
TXCK0:  IF LOC(1) <> 0 THEN GOTO TXOK0
     TEMPO = TEMPO + 1
     IF TEMPO >= 3000 THEN GOTO ECOM
     GOTO TXCK0
TXOK0:  P0CK = ASC(INPUT$(1, 1))
     IF P0CK = TX0 THEN GOTO TXDAT1
     IF P0CK <> TX0 THEN GOTO ECOM
TXDAT1: PRINT #1, CHR$(TX1);
     FOR K = 1 TO 100
     NEXT K
     TEMPO = 0
TXCK1:  IF LOC(1) <> 0 THEN GOTO TXOK1
     TEMPO = TEMPO + 1
     IF TEMPO >= 3000 THEN GOTO ECOM
     GOTO TXCK1
TXOK1:  P1CK = ASC(INPUT$(1, 1))
     IF P1CK = TX1 THEN GOTO TXDAT2
     IF P1CK <> TX1 THEN GOTO ECOM
TXDAT2: PRINT #1, CHR$(TX2);
     FOR K = 1 TO 100
     NEXT K
     TEMPO = 0
TXCK2:  IF LOC(1) <> 0 THEN GOTO TXOK2
     TEMPO = TEMPO + 1
     IF TEMPO >= 3000 THEN GOTO ECOM
     GOTO TXCK2
TXOK2:  P2CK = ASC(INPUT$(1, 1))
```

```
      IF P2CK = TX2 THEN GOTO VERIF
      IF P2CK <> TX2 THEN GOTO ECOM
VERIF:  IF RX0 = 0 THEN GOTO 8
      PRINT #1, CHR$(0);
      FOR K = 1 TO 100
      NEXT K
      TEMPO = 0
STVER:  IF LOC(1) <> 0 THEN GOTO STVOK
      TEMPO = TEMPO + 1
      IF TEMPO >= 3000 THEN GOTO ECOM
      GOTO STVER
STVOK:  ST = ASC(INPUT$(1, 1))
      IF ST = 0 THEN GOTO VERX0
      IF ST <> 0 THEN GOTO ECOM
VERX0:  IF LOC(1) <> 0 THEN GOTO VERDAT0
      TEMPO = TEMPO + 1
      IF TEMPO >= 3000 THEN GOTO ECOM
      GOTO VERX0
VERDAT0: VRX0 = ASC(INPUT$(1, 1))
      IF VRX0 = RX0 THEN GOTO VERX1
      IF VRX0 <> RX0 THEN GOTO FALLA
VERX1:  IF LOC(1) <> 0 THEN GOTO VERDAT1
      TEMPO = TEMPO + 1
      IF TEMPO >= 3000 THEN GOTO ECOM
      GOTO VERX1
VERDAT1: VRX1 = ASC(INPUT$(1, 1))
      IF VRX1 = RX1 THEN GOTO VERX2
      IF VRX1 <> RX1 THEN GOTO FALLA
VERX2:  IF LOC(1) <> 0 THEN GOTO VERDAT2
      TEMPO = TEMPO + 1
      IF TEMPO >= 3000 THEN GOTO ECOM
      GOTO VERX2
VERDAT2: VRX2 = ASC(INPUT$(1, 1))
      IF VRX2 = RX2 THEN GOTO 8
      IF VRX2 <> RX2 THEN GOTO FALLA
ECOM:   EE = EE + 1
      IF EE = 3 THEN GOTO EFIN
      LOCATE 1, 1
      FOR M = 1 TO 24
      COLOR 0, 12
      PRINT "                                                    "
      NEXT M
      CALL CUAD3(8, 15, 14, 65, 196, 179, 218, 217, 191, 192, 11, 0)
      COLOR 12, 0
      LOCATE 9, 16
      PRINT "                                    "
      LOCATE 10, 16
      PRINT "                                    "
      LOCATE 11, 16
      PRINT "         ERROR EN LA COMUNICACION          "
      LOCATE 12, 16
      PRINT "                                    "
      LOCATE 13, 16
      PRINT "                                    "
      LOCATE 22, 40
      COLOR 7, 0
```

```
      PRINT "(presione cualquier tecla para reintentar)"
      SOUND 1300, 1
      SOUND 800, 1
      WHILE INKEY$ = ""
      WEND
      SOUND 800, 1
      SOUND 1300, 1
      CLS
      IF AB$ = "PRUEBA" THEN CALL PRUEBA
      CALL PRUEBAS
EFIN:  COLOR 12, 0
      LOCATE 9, 16
      PRINT "                                    "
      LOCATE 10, 16
      PRINT "        ERROR EN LA COMUNICACION          "
      LOCATE 11, 16
      PRINT "                                    "
      LOCATE 12, 16
      PRINT "        FAVOR VERIFICAR CONEXION          "
      LOCATE 13, 16
      PRINT "                                    "
      LOCATE 22, 40
      COLOR 7, 0
      PRINT "(presione cualquier tecla para salir"
      SOUND 1300, 1
      SOUND 800, 1
      WHILE INKEY$ = ""
      WEND
      COLOR 7, 0
      CLS
      END

FALLA: LOCATE 1, 1
FOR M = 1 TO 24
      COLOR 0, 1
      PRINT "                                                    "
NEXT M
CALL CUAD3(8, 15, 14, 65, 196, 179, 218, 217, 191, 192, 11, 0)
COLOR 11, 0
LOCATE 9, 16
PRINT "                                    "
LOCATE 10, 16
PRINT "                                    "
LOCATE 11, 16
PRINT "       CIRCUITO INTEGRADO DEFECTUOSO        "
LOCATE 12, 16
PRINT "                                    "
LOCATE 13, 16
PRINT "                                    "
SOUND 1300, 1
SOUND 800, 1
LOCATE 23, 40
COLOR 7, 0
INPUT "Desea probar otro circuito integrado (S/N)"; COR5$
IF UCASE$(COR5$) = "S" THEN CALL MAIN
COLOR 7, 0
```

```
CLS
END
8 END SUB
```

4.4.- PROGRAMA DEL EQUIPO (MICROCONTROLADOR 8751)

```
;PROBADOR DE CIRCUITOS INTEGRADOS CMOS
;IVAN R. AGUIRRE A.
;DICIEMBRE 1996
;_____________________________________
;ASIGNACION DE ETIQUETAS A MEMORIA RAM INTERNA

ZOCP0     EQU     7DH       ;ENTRADAS/SALIDAS P0
ZOCP1     EQU     7EH       ;ENTRADAS/SALIDAS P1
ZOCP2     EQU     7FH       ;ENTRADAS/SALIDAS P2
BSERIAL   BIT     00H       ;BANDERA SERIAL
RXSERIAL  EQU     7CH       ;RECEPCION SERIAL

;_____________________________________

          ORG     0000H
          LJMP    INICIO
          ORG     0023H
          LJMP    SERIAL
          ORG     0100H
          CLR     BSERIAL
INICIO:       MOV     IE, #10010000B        ;HABILITO INT.
SERIAL
          MOV     PCON, #00H
          MOV     TMOD, #00100001B      ;TIMER5
          MOV     SCON, #01010000B      ;SERIAL TX (8-B. UART)
          MOV     TH1, #0FEH            ;BAUD RATE=9600
          MOV     TL1, #0FEH            ;BAUD RATE=9600
          SETB    TR1
; ESPERO DATO DESDE PC
ST:       JNB     BSERIAL, ST
          CLR     BSERIAL
          MOV     A,RXSERIAL
          CJNE    A,#00H,ST
          MOV     SBUF,A
PP0:          JNB     BSERIAL,PP0
          CLR     BSERIAL
          MOV     A,RXSERIAL
          MOV     ZOCP0,A
          MOV     SBUF,A
          MOV     P0,A
PP1:          JNB     BSERIAL,PP1
          CLR     BSERIAL
          MOV     A,RXSERIAL
          MOV     ZOCP1,A
          MOV     SBUF,A
          MOV     P1,A
PP2:          JNB     BSERIAL,PP2
          CLR     BSERIAL
          MOV     A,RXSERIAL
          MOV     ZOCP2,A
          MOV     SBUF,A
          MOV     P2,A
          MOV     A,ZOCP1
          CJNE    A,ZOCP2,ST1
```

```
        SJMP    ST
ST1:    JNB     BSERIAL,ST1
        CLR     BSERIAL
        MOV     A,RXSERIAL
        CJNE    A,#00,ST1
        MOV     SBUF,A
PP01:   JNB     BSERIAL,PP01
        CLR     BSERIAL
        MOV     A,RXSERIAL
        MOV     SBUF,A
        MOV     P0,A
PP11:   JNB     BSERIAL,PP11
        CLR     BSERIAL
        MOV     A,RXSERIAL
        MOV     SBUF,A
        MOV     P1,A
PP21:   JNB     BSERIAL,PP21
        CLR     BSERIAL
        MOV     A,RXSERIAL
        MOV     SBUF,A
        MOV     P2,A
ST2:    JNB     BSERIAL,ST2
        CLR     BSERIAL
        MOV     A,RXSERIAL
        CJNE    A,#00,ST2
        MOV     SBUF,A
PP02:   JNB     BSERIAL,PP02
        CLR     BSERIAL
        MOV     A,RXSERIAL
        MOV     SBUF,A
        ORL     A,ZOCP0
        MOV     P0,A
PP12:   JNB     BSERIAL,PP12
        CLR     BSERIAL
        MOV     A,RXSERIAL
        MOV     SBUF,A
        ORL     A,ZOCP1
        MOV     P1,A
PP22:   JNB     BSERIAL,PP22
        CLR     BSERIAL
        MOV     A,RXSERIAL
        MOV     SBUF,A
        ORL     A,ZOCP2
        MOV     P2,A
ST3:    JNB     BSERIAL,ST3
        CLR     BSERIAL
        MOV     A,RXSERIAL
        CJNE    A,#00,ST3
        MOV     SBUF,A
PP03:   JNB     BSERIAL,PP03
        CLR     BSERIAL
        MOV     A,RXSERIAL
        MOV     SBUF,A
        ORL     A,ZOCP0
        MOV     P0,A
PP13:   JNB     BSERIAL,PP13
```

```
            CLR     BSERIAL
            MOV     A,RXSERIAL
            MOV     SBUF,A
            ORL     A,ZOCP1
            MOV     P1,A
PP23:       JNB     BSERIAL,PP23
            CLR     BSERIAL
            MOV     A,RXSERIAL
            MOV     SBUF,A
            ORL     A,ZOCP2
            MOV     P2,A
SR:         JNB     BSERIAL,SR
            CLR     BSERIAL
            MOV     A,RXSERIAL
            CJNE    A,#00,SR
            CLR     EA
            MOV     SBUF,A
TRAS0:      JNB     TI,TRAS0
            CLR     TI
RXP0:       MOV     A,P0
            MOV     SBUF,A
TRASP0:     JNB     TI,TRASP0
            CLR     TI
RXP1:       MOV     A,P1
            MOV     SBUF,A
TRASP1:     JNB     TI,TRASP1
            CLR     TI
RXP2:       MOV     A,P2
            MOV     SBUF,A
TRASP2:     JNB     TI,TRASP2
            CLR     TI
            SETB    EA
            SJMP    ST3

SERIAL:     JB      TI,FINSERIAL
            MOV     A,SBUF
            MOV     RXSERIAL,A
            SETB    BSERIAL
FINSERIAL:  CLR     TI
            CLR     RI
            RETI

            END
```

CAPITULO V

PRUEBAS Y RESULTADOS

5.1.- TIPOS DE CIRCUITOS INTEGRADOS PROBADOS.

Para probar el funcionamiento del equipo se procedió primeramente a ingresar algunos circuitos integrados en el archivo CI.LIB que constituye la librería en donde se tiene toda la información de dichos circuitos integrados. Se ha procurado escoger chips que representen a diferentes grupos dentro de la familia CMOS.

Entre los escogidos se tiene los siguientes:

- ECG4000, Dual 3-Input NOR Gate Plus Inverter
- ECG4001, Quad 2-Input NOR Gate
- ECG4002, Dual 4-Input NOR Gate
- ECG4008, 4-Bit Full Adder
- ECG4050, Hex Non-Inverting Buffer
- ECG74251, 8-Input Multiplexer; 3-State

El primer paso constituye ingresar las características de los circuitos integrados a probarse en la librería CI.LIB por medio del programa de ampliación de circuitos integrados CILIB.EXE.

A continuación se tiene las tablas de verdad de los circuitos integrados utilizados para las pruebas. Estas muestran los valores lógicos puestos y obtenidos en los respectivos pines, "+" indica polarización positiva y "-" indica polarización negativa. Además se incluye la línea de datos para cada circuito integrado tal como se almacena en el archivo CI.LIB:

ECG4000:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0	0	0	1	-	0	1	1	0	0	0	+
1	1	0	0	1	0	-	0	1	0	0	0	1	+
1	1	0	1	0	0	-	0	1	0	0	1	0	+
1	1	0	1	1	0	-	0	1	0	0	1	1	+
1	1	1	0	0	0	-	1	0	0	1	0	0	+
1	1	1	0	1	0	-	1	0	0	1	0	1	+
1	1	1	1	0	0	-	1	0	0	1	1	0	+
1	1	1	1	1	0	-	1	0	0	1	1	1	+

"4000","DUAL 3-INPUT NOR GATE + NOT",2,64,24,2,128,0,2,56,228,2, 88,116,2,104,180,2,72,52,2,112,216,2,80,88,2,96,152,2,64,24,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

ECG4001:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	1	1	0	0	-	0	0	1	1	0	0	+
0	1	0	0	0	1	-	0	1	0	0	0	1	+
1	0	0	0	1	0	-	1	0	0	0	1	0	+
1	1	0	0	1	1	-	1	1	0	0	1	1	+

"4001","QUAD 2-INPUT NOR GATE",2,24,48,2,128,0,2,102,204, 2,58, 116,2,92,184,2,24,48,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

ECG4002:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	0	0	1	-	1	0	0	0	0	1	+
0	0	0	0	1	1	-	1	0	0	0	1	0	+
0	0	0	1	0	1	-	1	0	0	1	0	0	+
0	0	0	1	1	1	-	1	0	0	1	1	0	+
0	0	1	0	0	1	-	1	0	1	0	0	0	+
0	0	1	0	1	1	-	1	0	1	0	1	0	+
0	0	1	1	0	1	-	1	0	1	1	0	0	+
0	0	1	1	1	1	-	1	0	1	1	1	0	+
0	1	0	0	0	1	-	1	1	0	0	0	0	+
0	1	0	0	1	1	-	1	1	0	0	1	0	+
0	1	0	1	0	1	-	1	1	0	1	0	0	+
0	1	0	1	1	1	-	1	1	0	1	1	0	+
0	1	1	0	0	1	-	1	1	1	0	0	0	+
0	1	1	0	1	1	-	1	1	1	0	1	0	+
0	1	1	1	0	1	-	1	1	1	1	0	0	+
0	1	1	1	1	1	-	1	1	1	1	1	0	+

"4002","DUAL 4-INPUT NOR
GATE",2,2,128,2,128,0,2,60,120,2,30,184,
2,46,216,2,14,152,2,54,232,2,22,168,2,38,200,2,6,136,2,58,240,2,26,1
76,2,42,208,2,10,144,2,50,224,2,18,160,2,34,192,2,2,128,0,0,0,0,0,0

ECG4008: Pines 8:Vss y 16:Vdd

1	2	3	4	5	6	7	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0	1	1	1	1	0	1
1	1	0	1	0	1	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0	1	0	1	0	0
1	0	0	1	1	0	0	1	1	0	1	0	1	1
0	0	1	0	1	0	1	1	0	0	0	0	1	1
1	1	0	1	0	1	0	1	0	0	0	0	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1

"4008","4-BIT FULL
ADDER",2,0,248,2,128,0,130,84,132,130,25,172,2,
43,132,2,102,80,2,43,120,130,84,120,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

ECG4050: Pines 8:Vss y 16:Vdd

1	2	3	4	5	6	7	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1

"4050","HEX NON-INVERTING
BUFFER",130,42,40,2,128,0,2,0,0,130,
126,188,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

ECG74251: Pines 8:Vss y 16:Vdd

1	2	3	4	5	6	7	9	10	11	12	13	14	15
1	1	1	1	0	0	1	1	1	1	1	1	1	1
0	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	0	0	0
0	1	0	0	1	0	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1	0	0	0	0
0	0	0	0	1	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	1	0	1	0	0	1	0
0	0	0	0	1	0	0	1	1	0	0	1	0	0
0	0	0	0	1	0	0	1	1	1	1	0	0	0

"74251","8-INPUT MULTIPLEXER; 3-
STATE",2,48,0,2,128,0,2,48,0,130,
103,252,130,107,236,130,109,244,130,110,228,2,111,248,130,111,104,
130,111,176,130,111,192,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0

Luego de actualizar la librería, se procedió a probar los circuitos integrados en el equipo. Cabe mencionar que la primera prueba la realiza el equipo polarizando con +5V y GND. Esto se hace con el fin de proteger al equipo en caso de que el circuito integrado tenga cortocircuitos a Vdd o Vss. Luego de realizada la primera prueba se puede intentar con otras polarizaciones como se vio en la sección de diseño del equipo. Aunque el equipo esta limitado a dieciocho

pruebas por circuito integrado, éstos resultan más que suficientes para la mayoría de circuitos integrados. Por ejemplo, para el caso del ECG4002 con compuertas de cuatro entradas bastan dieciséis pruebas para verificar toda su tabla de verdad. En la mayoría de los casos se requerirá de menos pruebas para cumplir con la tabla de un circuito integrado cualesquiera. La razón de esta limitación es que QBASIC solamente compila programas que no sobrepasen los 64Kbytes.

5.2.- RESULTADOS OBTENIDOS

En todos los casos se tuvieron resultados satisfactorios y no se nota anomalía alguna del equipo luego de estar encendido por bastante tiempo. El equipo sí es, sin embargo, sensible a errores de operación. Si no se siguen los procedimientos indicados en la pantalla se pueden tener errores en la comunicación por falla en el sincronismo entre el computador personal y el equipo de pruebas. El equipo no requiere de controles o botones para su funcionamiento, excepto por el botón de <<RESET>>, el que debe pulsarse cada vez que el computador lo indique.

El programa de pruebas es muy claro en indicar si el circuito integrado esta bueno o malo.

No se pudo probar circuitos integrados secuenciales, pues no se puede conocer con exactitud cual es el estado presente antes de realizar las pruebas. Esto se debe a la estructura del programa que escribe primero en los pines del circuito integrado antes de leer las salidas.

Las salidas de tres estados también constituyeron un problema particular, pues como el pin del microcontrolador debe estar en uno lógico para poder escribirse sobre él, la ausencia de tal escritura (alta impedancia) hace que se mantenga dicho estado, debiéndose leer uno lógico en el pin del microcontrolador. Pero si se diera el caso que por falla del circuito integrado esté poniendo un pin a cero lógico (el microcontrolador y el chip en prueba trabajan a niveles lógicos opuestos) el equipo no lo detectaría pues se sobreescribiría un uno lógico sobre el uno lógico existente ya en el pin.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

Debido a que se requería de veinte líneas hacia el zócalo de prueba y veinte líneas de control para habilitar sólo una dirección de cada línea de prueba, se decidió utilizar un arreglo de flip-flops e interruptores analógicos en vez de ampliar los pórticos. Esto resultó ser una opción muy sencilla de implementar, pues de todas formas se necesitaba de los interruptores analógicos, teniéndose que aumentar únicamente un flip-flop tipo D por línea.

También resultó más económico utilizar reguladores de voltaje programables, pues de lo contrario hubiera sido necesario construir una fuente separada para cada voltaje requerido. De igual forma se hubiera tenido que incorporar relés u otro tipo de control para seleccionar por software los voltajes deseados.

La utilización de un PC para manejar la información y servir de interfaz con el usuario, y un equipo separado para realizar las pruebas, permitió evitar la necesidad de incorporar teclados y pantallas adicionales, indispensable en caso de tener un equipo de prueba totalmente independiente. De la misma manera, el programa del microcontrolador 8751 resultó muy sencillo, pues éste realiza muy

pocas instrucciones, manejando los datos desde el puerto serial a los puertos paralelos y viceversa.

Si bien es cierto que no se hizo el intento de optimizar la velocidad de procesamiento y transmisión, en realidad no fue necesario, pues el equipo no requiere de más que unos segundos para probar un circuito integrado.

Tampoco se utilizaron líneas de hand-shake para la comunicación serial, pues la aplicación es muy sencilla. En este sentido la implementación de subrutinas de error en la comunicación permitieron una solución mucho más fácil que el uso de líneas de hand-shake.

El equipo tiene la ventaja de ser relativamente pequeño, lo que le permite ser transportado con facilidad de un lugar a otro, necesitándose únicamente un computador equipado con pórtico serial para poder operar el probador de circuitos integrados.

Es verdad que el equipo tiene sus limitaciones en cuanto a número de pruebas y tipos de circuitos integrados que puede probar, es sin embargo, un prototipo muy completo que ofrece una solución al problema de comprobar el funcionamiento de chips utilizados frecuentemente.

A excepción del problema de alta impedancia el hardware no es el que provoca estas limitaciones, sino la manera como fue enfocado el software desde un comienzo. Es posible perfeccionarlo cambiando la forma como se realizan las pruebas y creando programas paralelos con el fin de lograr un equipo de prueba mucho más completo y útil. Estas modificaciones por sí solas constituyen material suficiente para una nueva tesis de grado.

El alcance de la presente tesis de grado es probar circuitos integrados de la familia CMOS únicamente, sin embargo, el equipo es capaz de probar circuitos integrados de otras familias como, por ejemplo, TTL. La forma como se diseño el hardware no ofrece limitación alguna en este aspecto. Además, el software permite ampliar la librería asignando la serie, descripción y características del chip que se requiera probar, sea éste TTL o CMOS. Eso sí, si se intenta probar un circuito integrado TTL se debe tener la precaución de no probar con otra polarización que no sea +5V / GND.

El presente equipo puede, de hecho, usarse en otras aplicaciones. Por ejemplo, si se desea comprobar la operación de un circuito combinacional cualquiera que sumadas sus entradas y salidas no sobrepasen las veinte, se puede añadir las características de dicho

circuito a la librería CI.LIB y conectar las respectivas salidas y entradas al zócalo de pruebas.

En cuanto al costo, sin ahondar demasiado podemos afirmar que se ha invertido menos de cien dólares en componentes. Esto lo vuelve un equipo relativamente económico si consideramos que probadores comerciales que trabajan a únicamente 5V y GND tienen precios similares o superiores al costo de fabricación del presente equipo. Por supuesto, tratándose de un prototipo únicamente, se terminó por utilizar la técnica de "wire-wrapping" y no con circuitos impresos, lo que hubiese encarecido de alguna manera su construcción.

BIBLIOGRAFIA

1) GONZALEZ J., Introducción a los Microcontroladores, Madrid, España, 1992.

2) MORRIS MANO M., Lógica Digital y Diseño de Computadores, México D.F., México, 1982.

3) NATIONAL SEMICONDUCTOR, CMOS Logic Databook, Santa Clara CA, USA, 1988.

4) PHILLIPS ECG, Master Replacement Guide, Williamsport, USA, 1994.

5) SPEAKMAN T., A Model for the Failure of Bipolar Silicon Integrated Circuits Subjected to ESD, USA, 1974.