

86



**UNIVERSIDAD NACIONAL AUTONOMA
DE MÉXICO**

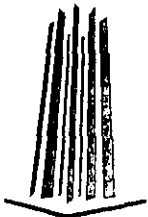
**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGÓN**

2970970

**MANIPULACIÓN Y ENVIO DE DATOS POR EL PUERTO
PARALELO A TRAVES DE UNA COMPUTADORA
PERSONAL, POR MEDIO DE UN PROGRAMA
COMPUTACIONAL**

T E S I S
QUE PARA OBTENER EL TÍTULO EN:
INGENIERIA MECANICA ELECTRICA
AREA : ELECTRICA - ELECTRONICA
P R E S E N T A :
ROSALES PEREA EDUARDO ALONZO

ASESOR:
LUNA ESCORZA PABLO



MÉXICO

2001



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

PAGINA

	INTRODUCCIÓN	1
CAPITULO I	PUERTO PARALELO E INTERFAZ	6
I.1	Registros del puerto paralelo	7
I.2	Los puertos paralelos	10
I.2.1	La interfaz de la computadora común	10
I.2.2	Líneas de tierras y de datos	12
I.2.3	Diálogo de datos	12
I.2.4	Diálogo de estado	14
I.3	El extremo del periférico	15
I.4	Dos trabajos convencionales de cableado	17
CAPITULO II	CIRCUITO PARA EL PROCESO DE DATOS	21
II.1	Circuitos lógicos	21
II.1.1	Circuitos para el proceso de datos	21
II.1.2	Circuitos TTL	22
II.2	Tipos de circuitos lógicos	23
II.3	Convertidores	24

PAGINA

II.3.1	Convertidores digital a analógico (D/A)	24
II.3.2	Convertidor analógico a Digital (A/D)	25
II.4	Codificación y decodificación de números binarios	28
II.4.1	Circuitos integrados CI codificadores y decodificadores	28
II.5	Registros	29
II.5.1	Registros de corrimiento	29
II.5.2	Registro con carga paralela	30
II.5.3	Registros con corrimiento	33
II.5.4	Registros con corrimiento bidireccional con carga paralela	35
II.6	Multiplexores y demultiplexores	36
II.7	Selección del circuito	40
II.8	Diseño	47
II.8.1	Pruebas	50
II.9	Datos	52
II.10	Circuito generador de pulsos de reloj	54
II.10.1	Pruebas	54

PAGINA

II.11	Circuito de entrada de datos	59
II.12	Características de los CI	60
II.12.1	CI 74151	60
II.12.2	CI 74125	61
II.13	El circuito y su funcionamiento	63
II.14	Fuentes de alimentación	68
II.14.1	Forma básica de una fuente de alimentación o de poder	69
II.14.2	Diodos rectificadores	70
II.14.3	Transformadores	71
II.14.4	Circuito rectificador de onda completa	72
II.14.5	Filtros para una fuente de alimentación	73
II.15	Fallas en las fuentes de alimentación	74
CAPITULO III	PROGRAMA Y ESTRUCTURA DE CONTROL BASIC	76
III.1	Determinación del problema	76
III.1.1	Definición del problema	76
III.1.2	Estudio de la factibilidad	77
III.1.3	Aprobación del proyecto	77

		PAGINA
III.2	Programa de control	77
III.2.1	¿Para qué es necesario el programa?	78
III.3	El lenguaje de programación Basic	78
III.4	Pruebas	85
III.5	Programa	86
III.6	Ejemplo	93
III.6.1	Programa para controlar un motor	93
III.6.2	Visual Basic	97
CAPITULO IV	DESARROLLO DE UNA APLICACION EN VISUAL BASIC	100
IV.1	Desarrollo de un fichero por medio de visual Basic	100
IV.2	Controles Activex	113
IV.3	Barra de herramientas	115
IV.4	Programa para interactuar con las entradas provenientes del puerto paralelo	129
IV.5	Control de salidas al puerto paralelo	137

	PAGINA
CONCLUSIONES	152
BIBLIOGRAFIA	153
GLOSARIO DE TERMINOS	157
APENDICE A (INTERFACES)	169
APENDICE B (TTL)	177

INTRODUCCIÓN

Es del conocimiento, para casi todo el mundo, que las computadoras son dispositivos sumamente útiles. Por lo que, si el lector es una persona que conoce de principios básicos en electrónica y que además sabe como comunicarse con ellas para su proyecto, pueden ser tan divertidas como funcionales. En este momento, algunos de los lectores que no están demasiado familiarizados con las computadoras podrían pensar que se darán explicaciones a nivel "científico espacial" sobre buses de computadoras, etc. Tranquilícese, de ninguna manera.

En cambio, me gustaría explicar como trabaja básicamente un puerto paralelo sencillo. Es sencillo trabajar con los puertos paralelos, que se pueden enganchar a casi cualquier proyecto que Ud. desee controlar con su computadora. El truco consiste en saber como trabajan.

Por lo tanto, y antes de comenzar a desarrollar nuestro tema, tendremos que mencionar algunas características básicas en electrónica en cuanto a lo que será necesario utilizar, para poder hacer que un usuario controle y manipule una computadora para que la misma controle la información dada por el usuario para ser utilizada en su proyecto.

Se hablará de un circuito que recibirá información proveniente de la computadora, pero para que sea controlado por la computadora será necesario saber como desarrollar un programa que controle datos¹ para ser enviados por el puerto paralelo² (lpt1) de un equipo de computo, para estados aleatorios que un usuario determine en un momento o variables de entrada que lleguen a ser introducidas a la computadora.

El circuito de control es la parte secundaria, la cual recibirá y mantendrá información proveniente de un equipo de computo, para eso es necesaria un programa con instrucciones que el usuario introduzca, siguiendo una serie de pasos para definir la funcionalidad del circuito, en diversas aplicaciones que el mismo usuario determine y conecte al circuito para ser operado por medio de la computadora.

¹ Datos: Nos referimos como dato a un conjunto de bits.

² Puerto Paralelo: Interfaz utilizada por la computadora principalmente para conectar a la impresora.

Uno de los objetivos a realizar es aplicar los conocimientos adquiridos en el área de sistemas digitales y diseño de sistemas digitales, no se intenta crear un producto nuevo, si no sobre la base de lo ya útil y conocido darle un direccionamiento a una posible aplicación hacia el exterior de una computadora.

Para diseñar el circuito será necesario tener bases de conocimiento digital, pero también debemos mencionar que la función del circuito será para realizar trabajos de almacenamiento temporal de datos, lectura de datos, puesto que para diseñarlo necesitaremos saber como trabajan las partes que componen nuestro circuito, dando como propósito dar una salida practica a los bits provenientes del puerto paralelo, controlando los estados de las entradas y salidas del programa, por lo que el programa será el encargado de controlar estas entradas y salidas que se dirigen hacia el circuito. Principalmente tendremos que saber como opera cada circuito integrado que es utilizado en el circuito, así como su forma de encendido y función que realiza. Todo esto se mencionara mas adelante.

El programa tiene como función principal dar seguimiento a los bits de entrada y salida que en determinado momento lleguen a ser establecidos. Este programa tiene como propósito que el circuito siga una serie de ordenes, que el usuario determine los pasos que el circuito ha de seguir de acuerdo a las necesidades del mismo, por lo que utilizaremos un lenguaje de programación fácil de entender y aplicar.

A fin de que se pueda aprovechar mejor esta maravillosa herramienta, la computadora, haremos una breve revisión sobre el desarrollo de los medios para dar instrucciones a esta, utilizando un lenguaje simple para poder manejar las bases técnicas y poder avanzar con seguridad para ser un usuario preparado.

Primeramente se mencionan las características de los puertos paralelos y su interfaz, así como su función y partes que lo integran, posteriormente desarrollaremos el circuito que será el encargado de recibir y enviar datos hacia la interfaz del puerto paralelo y finalmente programaremos la entrada y salida de datos del puerto paralelo en lenguaje Basic y Visual Basic.

OBJETIVO GENERAL

Recopilar como las computadoras que son dispositivos sumamente útiles, pueden ser funcionales para trabajar con el puerto paralelo de forma sencilla y practica, para ser acopladas a casi cualquier tipo de proyecto que se deseen controlar por medio de una computadora personal y una programación básica y sencilla, para mostrar como funcionan, mediante la programación en el lenguaje Basic y Visual Basic

OBJETIVO ESPECIFICO

Hablar de cómo podemos manipular las entradas y salidas de datos del puerto paralelo (Ipt1) de una computadora hacia un circuito que se encargue de administrar estos bits.

Teniendo en cuenta que se utilizaran circuitos integrados (familia TTL) por su facilidad de uso y de poder ser encontrados en tiendas de electrónica básica.

Además, por medio de un programa con instrucciones definidas se realizara tanto la entrada como la salida de bits de la computadora.

CIRCUITO DE SALIDA

El circuito de salida utilizara circuitos integrados TTL de tipo registro, demultiplexor, decodificador, etc.

Para poder obtener 8 bits provenientes del puerto paralelo de una computadora, estos bits serán distribuidos dentro del circuito puesto que este cuenta con diversos puntos de almacenamiento temporal (registros) de 4 bits que nos serán útiles para ser enviados a circuitos que puedan ser acoplados a estos 4 bits.

CIRCUITO DE ENTRADA

Nos referimos al circuito de entrada, como al circuito que recibirá las señales externas provenientes de una lógica combinacional, que se envían hacia la computadora para que obtengamos un control de procesos que sean ejecutados por el programa, este circuito también cuenta con circuitos digitales de la familia TTL, de bajo consumo de corriente, las señales que maneje serán digitales, y podrán ser manipuladas por medios de una multiplexión sencilla.

PROGRAMA

El programa será escrito mediante las bases de programación estructurada, ya que mediante éste podremos realizar la selección y utilización de diferentes bloques de instrucciones programados por el usuario para procesar las señales que él requiera, para esto se realizara mediante dos formas:

La 1ª. Será por un programa de selección bits de salida al puerto paralelo

La 2ª. Será la selección de entrada de bits provenientes del puerto paralelo

El programa será realizado tanto en lenguaje Basic como Visual Basic, para obtener un mejor entendimiento de las instrucciones, ventajas y desventajas con las que cuenta Basic y Visual Basic.

Por qué Basic

Para que una computadora pueda interpretar y procesar la información que se le da, mediante la unidad de entrada correcta, es necesario que este dada bajo ciertas reglas. Al conjunto de estas reglas se le conoce como lenguaje, existen diferentes tipos de lenguajes con diversas características para múltiples aplicaciones.

Basic requiere que la información que maneja este proporcionada como constante, variables o instrucciones para que sea interpretada correctamente.

Mediante el uso de este lenguaje de programación podemos encontrar las instrucciones necesarias para el propósito que buscamos, que es la salida y manipulación de datos para ser enviados al puerto paralelo (lpt1) de una computadora; Mediante la explicación de las sentencias e instrucciones con las que cuenta Basic, mostraremos de una forma sencilla como se puede llegar a realizar un programa para la salida y entrada de datos del puerto paralelo.

Usualmente es difícil encontrar computadoras que trabajen bajo simples sistemas operativos, actualmente hoy en día la mayoría de las computadoras cuentan con sistema operativo Windows, por lo que también hablaremos de las instrucciones de Visual Basic que trabaja en el ambiente Windows para realizar el mismo trabajo.

Por qué Visual Basic

Visual Basic es actualmente un lenguaje de programación popular en el mundo, y constituye un producto de interfaz gráfica de usuario¹, el cual sirve para crear aplicaciones Windows basado en el lenguaje Basic y en la programación orientada a objetos.

Visual Basic ha evolucionado a partir del lenguaje Basic original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows.

Visual Basic permite crear programas para su uso personal, para su grupo de trabajo, para una empresa, aplicaciones distribuidas a través de Internet, aplicaciones de bases de datos y otras muchas más.

¹ Interfaz gráfica de usuario: Ver glosario de Términos

CAPITULO I

PUERTO PARALELO E INTERFAZ

El puerto paralelo se utiliza generalmente en la PC para conectar a la impresora con la computadora, no obstante, esta interfaz es flexible y puede ser utilizada para una gran variedad de aplicaciones diversas, entre las que se encuentran la conexión de pequeños circuitos de diversos usos, así como la conexión de los ordenadores de forma directa.

La particularidad del puerto paralelo es que está diseñado para transmitir y recibir los datos de una computadora, de ocho en ocho bits (de byte en byte), lo que nos ofrece es fundamentalmente una mayor transferencia de información. Una diferencia del puerto paralelo con respecto al puerto serie es que utiliza niveles lógicos de 0, 1, que son cero y cinco volts positivos, respectivamente. En vez de las tensiones negativas y positivas entre 3 y 15 volts utilizadas por el puerto serie.

Con respecto del puerto paralelo observemos detenidamente la interconexión de ordenadores, observando cada uno de los contactos de la terminal de la interfaz del puerto paralelo, como son mencionados en la tabla 1.

TERMINAL DEL PUERTO PARALELO RS232 (DB-25)				
CONECTOR PC	SEÑAL	TIPO SEÑAL	CONECTOR IMPRESORA	DESCRIPCIÓN
1	/SRT	CONTROL	1	activación trasferencia de datos
2	D0	DATO	2	BIT de dato 0
3	D1	DATO	3	BIT de dato 1
4	D2	DATO	4	BIT de dato 2
5	D3	DATO	5	BIT de dato 3
6	D4	DATO	6	BIT de dato 4
7	D5	DATO	7	BIT de dato 5
8	D6	DATO	8	BIT de dato 6
9	D7	DATO	9	BIT de dato 7
10	/ACK	ESTADO	10	Transferencia de datos correcta

11	BSY	ESTADO	11	Impresora ocupada
12	PAP	ESTADO	12	Sin papel
13	OFON	ESTADO	13	Indicador de impresora en línea
14	/ALF	CONTROL	14	Alimentación de línea automática
15	/FEH	ESTADO	32	Error
16	/INI	CONTROL	31	Iniciación de impresora
17	/DSL	CONTROL	36	Iniciación de impresora
18-25	TIERRAS		19-30, 33	Tierra 0 V

TABLA 1
CONECTOR DEL PUERTO PARALELO RS232 (DB-25)

I.1 Registros del puerto paralelo

El puerto paralelo dispone de registros a través de los cuales se intercambia tanto los datos de información como las señales de control. Estos registros de tres tipos: Registro de datos, Registro de estado y Registro de control.

El registro de datos: Contiene la información que va a ser transferida a la interfaz o que ha sido recibida de la misma.

El registro de estados: El registro de Estados nos ofrece información del estado del dispositivo conectado al puerto paralelo (si está conectado o no, si reciben los datos correctamente, etcétera).

El registro de control: controla el comportamiento del dispositivo, así como la generación de interrupciones hardware.

El registro de datos y del registro de control son registros bidireccionales, mientras que el registro de estado es un registro de sólo lectura.

Cada puerto paralelo (de LPT1 a LPT4) dispone de una dirección para acceder a estos registros que está contenida en el área del BIOS.

El registro de datos se encuentra situado en el desplazamiento 00h de dicha dirección, el registro de estado en el desplazamiento 01h y el registro de control en el desplazamiento 02h. Esto significa que si nuestro puerto paralelo de la impresora LPT1 que suele encontrarse en el direccionamiento 378h, encontraremos al registro de estado se encontrara en el direccionamiento 379h y al registro de datos en 37Ah.

AREA DE DATOS DE LA BIOS PARA EL PUERTO PARALELO	
DIRECCIÓN	CONTENIDO
40:08	Dirección base Lpt1
40:0A	Dirección base Lpt2
40:0C	Dirección base Lpt3
40:0E	Dirección base Lpt4
40:78	Temporización Lpt1
40:79	Temporización Lpt2
40:7A	Temporización Lpt3
40:7B	Temporización Lpt4
40:11	Número de puertos paralelo (2 bits más significativos)

TABLA 2

AREA DE DATOS DE LA BIOS PARA EL PUERTO PARALELO

BITS DE LOS REGISTROS DEL PUERTO PARALELO

REGISTRO	OFFSET		7	6	5	4	3	2	1	0
Registro de datos	00h	Señal	D7	D6	D5	D4	D3	D2	D1	D0
		Patilla	9	8	7	6	5	4	3	2
Registro de estado	01h	Señal	/BSY	/ACK	PAP	OFON	/FEH	-	-	-
		Patilla	11	10	12	13	15	-	-	-
Registro de control	02h	Señal	-	-	-	IRQ	DSL	/INI	ALF	SRT
		Patilla	-	-	-	-	17	16	14	1

TABLA 3

BITS DE LOS REGISTROS DEL PUERTO PARALELO

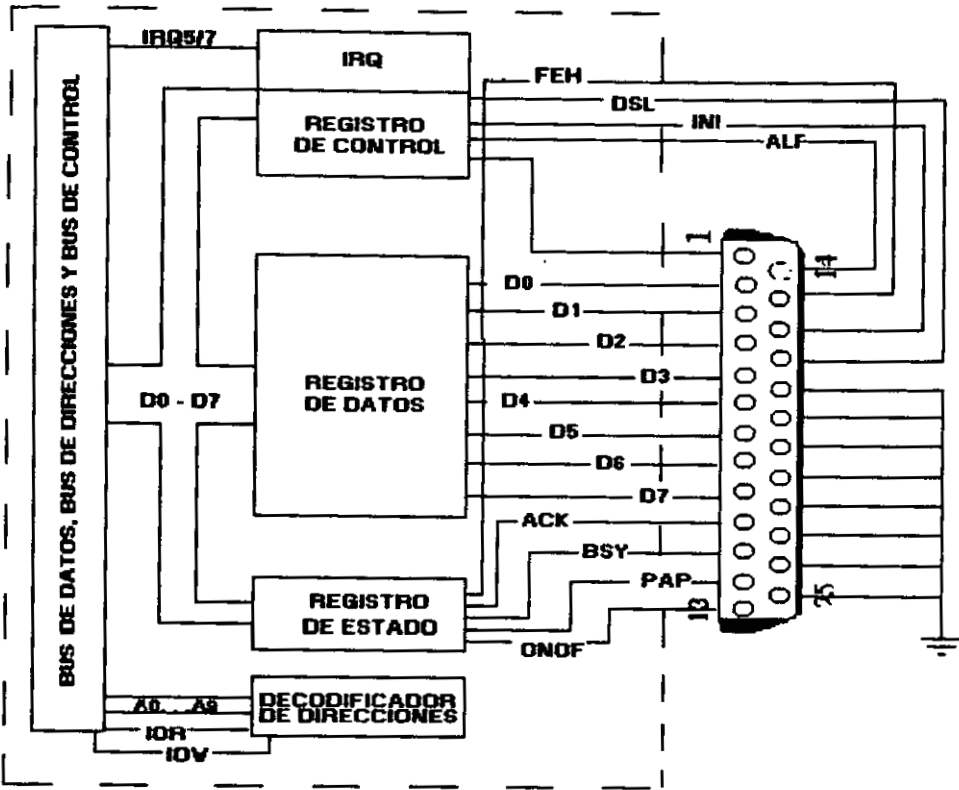


FIGURA 1: DIAGRAMA DE BLOQUE DE LA INTERFAZ PARALELA

De acuerdo al esquema anterior (figura 1) podemos observar los parámetros contenidos en la tabla 1.

INTERFAZ DEL PUERTO PARALELO

1.2 LOS PUERTOS PARALELOS

La computadora es un dispositivo útil, por medio del cual puede usarse para comunicarse a través del puerto paralelo, por lo cual se espera que sea tanto divertida como funcional. En este momento, algunos de los lectores que no están demasiado familiarizados con las computadoras podrían pensar como pueden llegar a ese fin, de manera sencilla para saber como trabajar con los puertos paralelos, que se pueden enganchar a casi cualquier proyecto que Ud. desee controlar con su computadora. El truco consiste en saber como trabajan.

Se presentará aquí un poco de información sobre las señales y el funcionamiento de los puertos paralelos, de modo que los más ambiciosos lectores podrán diseñar y construir proyectos con interfaces paralelas. Incluirá también algunos datos sobre los periféricos paralelos.

1.2.1 La interfaz de la computadora común

Puesto que las interfaces paralelas de la mayoría de las computadoras son casi idénticas, tomemos por ahora como ejemplo el más universal de ellos (que se encuentra en la mayoría de las tarjetas IBM-compatibles). La información debe ser suficiente para darles una buena base sobre el funcionamiento de cualquier interfaz paralela, las terminales pueden variar aunque casi todas las líneas de señal son las mismas, aunque no tenga una compatible.

NOTA: No trate de colgar equipos en paralelo sobre la base de las descripciones de los pines solamente. El hacer esto puede llegar a tener el riesgo de dañar sus equipos. Espere hasta que haya leído todas las precauciones que se recomiendan.

La mayoría de los puertos paralelos de la parte posterior de las computadoras IBM-compatibles poseen una terminal **DB-25**. Esto es extraño, puesto que esta terminal es el de serie que se utiliza para las interfaces serie. La terminal es normalmente hembra (tiene orificios en lugar de patitas) para distinguirlo de los conectores serie, que son habitualmente macho y que puede tener también la computadora.

La función de cada terminal del DB-25 se muestra en la Fig. 2.

Las señales que ocupan esas terminales se pueden dividir en cuatro grupos básicos: Tierras, salidas de datos, entradas de dialogo y salidas de dialogo. En la Fig. 2, las tierras se indican con círculos, las entradas de dialogo se indican con flechas que apuntan a la terminal y las salidas (tanto de datos como de dialogo) tienen flechas que apuntan hacia afuera de la terminal. (Note que algunas de las líneas tienen una abreviatura convencional que se indica entre paréntesis.)

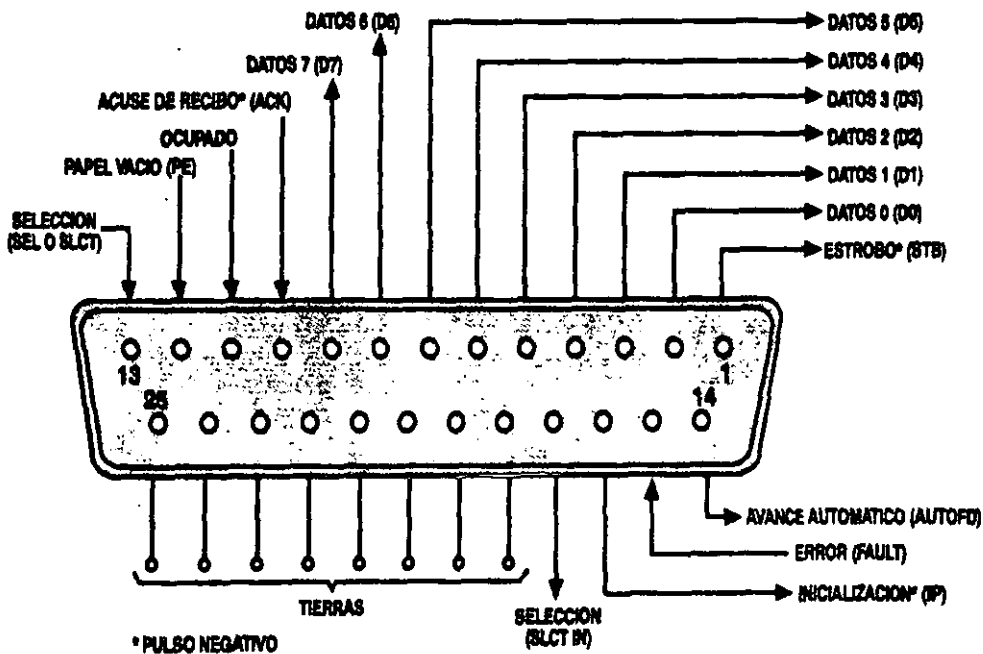


FIGURA 2: En la mayoría de las computadoras IBM compatibles (y algunas no compatibles) el puerto paralelo es como éste. Las flechas que apuntan hacia fuera del DB-25 son salidas, las que apuntan hacia adentro son entradas. Las terminales restantes, señaladas con círculos son tierras

1.2.2 Líneas de tierras y de datos

Las tierras cumplen dos funciones: la primera es que vinculan las tierras de señal de los dos dispositivos que se interconectan de modo que puedan compartir una tierra común como referencia para la señal.

La otra es que, puesto que, la conexión entre los dos dispositivos se realiza a menudo mediante un cable tipo cinta, las tierras (Llamadas muchas veces retornos de tierra en este contexto) actúan como blindajes de las líneas más importantes. Por ejemplo, el conductor conectado a la terminal 19 de un cable de cinta apantalla a la 6 de la 7, y viceversa. Esto impide que las señales D4 (cuya función veremos enseguida) afecten capacitivamente la línea D5, y viceversa. En los cables de calidad que no se hacen de tipo cinta, cada retorno de tierra se retuerce alrededor de una línea de señal formando un par trenzado, para proporcionar un poco de blindaje.

Como su nombre lo indica, la salida de datos transfiere información desde la computadora a un periférico en paralelo. Esto se hace con ocho bits (un byte) por vez utilizando las terminales de la 2 a la 9.

D0 se considera el BIT menos significativo (LSB) y D7 el más significativo (MSB), (Nota: algunas computadoras emplean las designaciones D1 -D8 en lugar de D0-D7); Algunos puertos de computadora no soportan el MSB. Por supuesto que en esos casos no debe diseñar sus proyectos de modo que lo necesiten. De la misma manera, algunos periféricos solo utilizan datos de 7 bits.

En tales circunstancias, el MSB se ignora o a veces se emplea como BIT de paridad. Va a necesitar un software bastante sofisticado para proporcionar bits de paridad, de modo que tal vez prefiera inhibir la verificación de paridad del dispositivo.

Los bits, como también las demás señales, se representan mediante niveles de tensión TTL convencionales: una señal entre 2.4 y 5 voltios es un nivel alto o 1 binario. Cualquier cosa entre 0.8 y 2.4 voltios se considera dato no valido o 0 binario.

1.2.3 Diálogo de datos

Puesto que la computadora es mucho más rápida que cualquier periférico con el que se comunique, puede fácilmente transmitir mas datos que los que el periférico puede manejar. Par ello, los periféricos utilizan señales especiales para decirle a la

computadora que detenga momentáneamente el envío de datos cuando tienen suficientes para trabajar. Esto le permite al periférico alcanzar a la computadora, que puede realizar otras tareas mientras tanto. Una vez que el periférico queda libre, le pide a la computadora que transmita más datos, y el proceso continúa.

Este juego computarizado de "luz roja, luz verde" que nos controlan el tránsito de datos se logra enviando señales por cables dedicados a ese propósito. El proceso de utilizar señales para controlar el flujo de datos se denomina diálogo (handshaking), de modo que las señales empleadas para ello se llaman "señales de diálogo".

Las señales de estrobo, ocupado y acuse de recibo son las señales de diálogo más importantes. Para ayudar a explicar cómo se relacionan y controlan el flujo de datos, observe la Figura 3.

Aquí se muestran las ocho líneas de datos concentradas como una sola línea en la parte superior. No se deje confundir, el valor de los bits individuales no tiene importancia. Lo que importa es el tiempo en que los datos sufren transiciones (representadas por las líneas cruzadas) y el tiempo en que permanecen constantes (las bandas).

Los datos que salen por las líneas D0-D7 comienzan a formarse en el tiempo t_1 y se establecen y quedan listos para utilizarse en el instante t_2 . Un momento después t_3 la computadora manda un pulso momentáneo negativo (llamado señal de "estrobo") al periférico, para indicar que los datos están listos y en espera en las líneas de datos. Luego de t_3 , el periférico puede responder en una de dos maneras: puede tirar de la línea ocupada hasta que este listo para más datos o puede esperar hasta que haya utilizado los nuevos datos y enviar entonces un pulso negativo de acuse de recibo a la computadora cuando desea más.

Cualquiera de las respuestas retiene a la computadora hasta que el periférico informe que está preparado. (Hay unos pocos periféricos que detienen a la computadora de ambas maneras, aunque hacerlo así es algo redundante). Luego que la línea ocupada se pone en bajo o se recibe un pulso de acuse de recibo, la computadora configura las líneas de datos para el siguiente byte, y se repite el procedimiento.

La línea ocupada se utiliza algunas veces para detener a la computadora por otras razones. Por ejemplo, si se acaba el papel de la impresora o se encuentra fuera de línea.

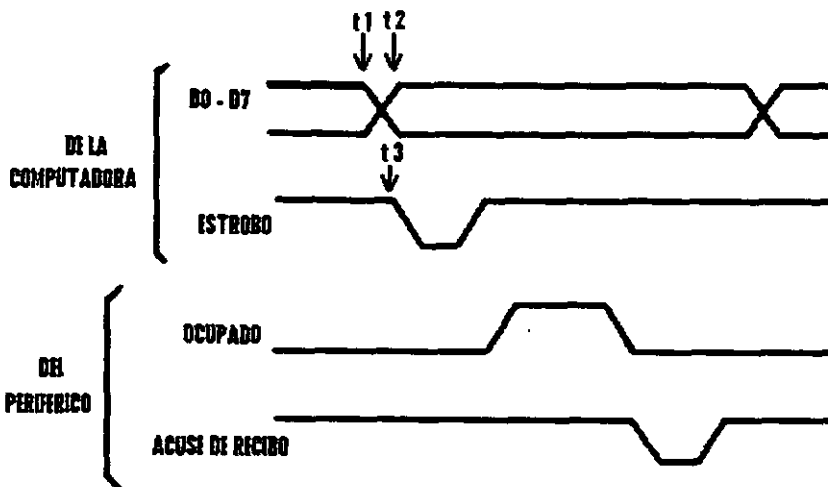


FIGURA 3: Una vez que la computadora configura los datos paralelos, se lo informa al periférico mediante un pulso de reloj negativo. El periférico responde poniendo en alto la línea de ocupado hasta que este listo para recibir más y / o espera hasta terminar con los datos y envía un pulso negativo de acuse de recibo.

1.2.4 Diálogo de estado

A lo largo de algunas líneas, a veces los periféricos paralelos (especialmente las impresoras) utilizan cables dedicados para indicar su estado. Puesto que el estado de un periférico puede afectar el flujo de datos, esto se puede considerar también una forma de dialogo. Por ejemplo, si una impresora, un graficador u oscilógrafo necesita informar a la computadora que se queda sin papel, puede hacerlo manteniendo alta la línea de papel vacío hasta que se aprovisione nuevamente. Esto impide que la computadora envíe datos al periférico cuando el dispositivo es incapaz de hacer algo con ellos. La mayoría de los puertos paralelos de tipo IBM soportan esta característica, pero no la soportan muchas otras computadoras domesticas. Asimismo, un periférico puede informar a la computadora que esta alimentado y en línea manteniendo alta la línea de "selección" de la terminal 13 (observe que hay dos líneas de selección en la figura 2, no las confunda); Esta es a veces una línea de señal necesaria porque algunos periféricos se pueden mantener alimentados pero fuera de línea, enviándoles un carácter especial de "deselección" (denominado DC1 o XON, que tiene el valor ASCII 17); Debe observar el manual del usuario de su periférico o el manual de programación para ver si contiene esa característica.

Un periférico puede hasta pedir ayuda sosteniendo baja la línea de error. Al igual que la línea ocupada, los periféricos de la impresora utilizan a veces la línea de error para indicar que simplemente están fuera de línea o que se acaba el papel.

La computadora puede hacer también requerimientos especiales o proporcionar datos de configuración enviando señales desde las restantes salidas de dialogo. (Tenga en cuenta que un periférico puede contener interruptores DIP que pueden configurarlo para que ignore los requerimientos de la computadora).

Por ejemplo, en algunos periféricos la característica de seleccion/deseleccion pueden habilitar e inhibir el puerto de la computadora. Para esos dispositivos, si la computadora mantiene alta la línea de salida de selección del terminal 17 (no confundirla con la línea de entrada de selección del terminal 13), la característica DC1 /DC3 se habilita. Al mantenerse esa línea baja, la característica se inhibe.

Asimismo, al mantener baja la línea de avance automático, la computadora solicita al periférico que acompañe cada retorno de carro con un avance de línea (es decir, la computadora informa al periférico que probablemente no enviara caracteres de avance de línea, de Modo que el periférico deberá agregarlos).

Por otra parte, si el computador envía un pulso negativo por la línea de inicialización (denominada técnicamente línea de entrada (input prime) o IP), el periférico que responde a esa línea se pondrá en cero (pasara a reset); Esto significa que el periférico adoptara cierta configuración por omisión y actúa normalmente como si recién se hubiera encendido. Tan poderosa como es, esta línea la soportan las computadoras IBM compatibles y muy pocas mas, puesto que hay a menudo mandatos especiales que se pueden enviar por las líneas de datos para lograr lo mismo.

1.3 El extremo del periférico

Si bien hemos tratado casi todas las señales que es probable que Ud. encuentre, hay aun algunas cosas que debe saber acerca de lo que hallara en el extremo del periférico del cable. La información adicional le ayudara a construir sus propios cables para periféricos ya construidos.

Un conector hembra de 36 conductores es la terminación mas adecuada que se encuentra en los periféricos paralelos. Las funciones convencionales de cada terminal de ese conector se muestran en la Fig. 4.

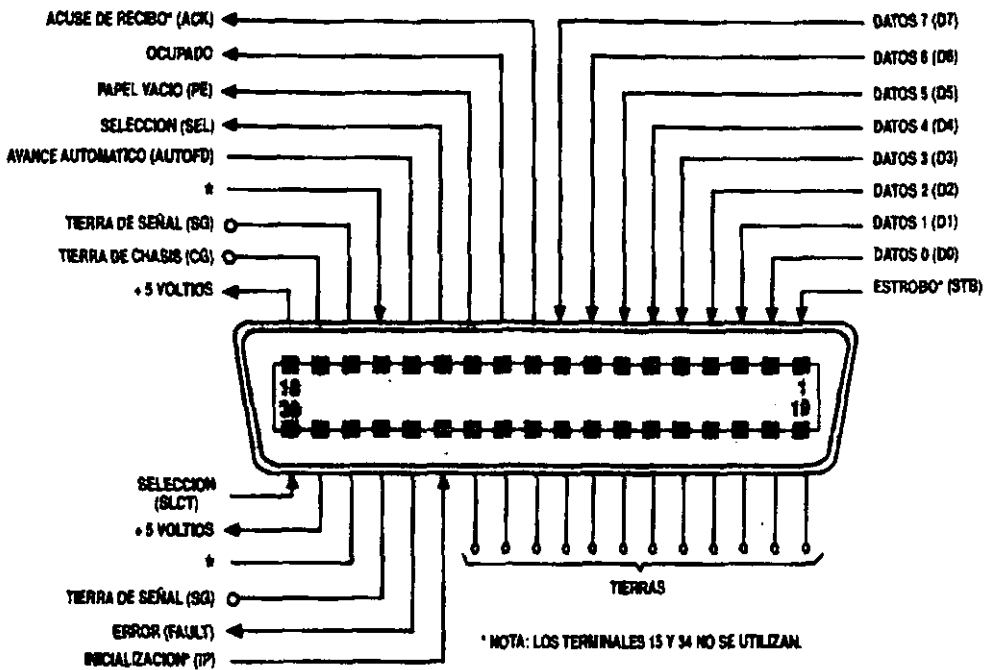


FIGURA 4: Esta es la asignación de terminales de un periférico compatible con Centronics.

Las funciones son las mismas que las del DB-25 ilustrado en la figura 3, pero hay unas cuantas tierras más y dos terminales de alimentación de 5 voltios.

Las flechas que señalan hacia el conector indican que el terminal es una entrada al periférico. Las flechas hacia afuera significan salidas del periférico, y los círculos son tierras.

Observe que este conector soporta unas cuantas funciones mas que el conector DB-25. Para nombrarlas, hay una tierra de chasis y dos líneas de 5 voltios. No todos los equipos paralelos las soportan. Tanto la tierra de chasis como las líneas de 5 V pueden causar inconvenientes si se conectan incorrectamente, de modo que mas adelante daré algunas indicaciones sobre como conectarlas en forma adecuada.

Las líneas de 5V las proporcionan algunos periféricos para mantener alta una línea de dialogo si es necesario. Por ejemplo, supongamos que un puerto de computadora

no genera un nivel alto para la línea de avance automático, pero el periférico necesita esa línea alta para funcionar correctamente. El preparador del cable puede resolver el problema con las funciones de terminales que se muestran (aunque se deben invertir las flechas de la Fig. 4) en lugar del conector DB-25 antes mencionado. Si descubre ese conector, tanto en la computadora como en el periférico, puede tentarse a utilizar un cable directo (uno que conecte terminal a terminal) ¡No lo haga! Puede dañar ambos equipos.

Explicare como construir el cable adecuado para esa situación, luego de presentar un par de indicaciones de seguridad.

PRECAUCION: Conectar dispositivos al puerto paralelo implica el riesgo de daños permanentes a la tarjeta madre de la PC, tenga siempre presente que aún los profesionales cometen errores, por lo tanto no está de más recomendarle extremo cuidado al trabajar en el puerto paralelo.

Sea precavido

Antes de lanzarse a cablear o construir proyectos de interfaz, es conveniente decir algunas cosas: se debe proceder con precaución cuando se trata de una interfaz paralela. Aunque las líneas de dialogo de los puertos paralelos son del tipo de colector abierto (es decir, se pueden cortocircuitar a tierra), las salidas de datos de una PC se pueden dañar con los cortocircuitos. Más aun, las tensiones mayores de 5V pueden dañar todas las líneas. La clave para la conexión segura de la interfaz entre equipos TTL es conocer las entradas y las salidas, de modo que pueda conectar la salida de un dispositivo a la entrada del otro, y viceversa. Si es prudente, debe efectuar siempre una doble comprobación de su cableado antes de alimentar los equipos.

I.4 Dos trabajos convencionales de cableado

El más sencillo trabajo paralelo que puede hacer es crear lo que llamare un "cable casi directo". Ese cable se puede utilizar para conectar una computadora a un periférico siempre que tengan conectores de 36 terminales de tipo paralelo y de acuerdo con la norma Centronics, Si ambos dispositivos tienen el conector correcto pero no esta seguro acerca de las funciones de las terminales, verifiquelas con un ohm-metro para ver si coinciden con la norma. Si no coinciden lo suficiente, tendrá que investigar su documentación para mas datos; un cable casi directo simplemente no funcionara.

Si parece que ambos dispositivos son compatibles con Centronics, conecte un cable que los conecte terminal a terminal excepto los 17, 18 y 35. Deje los tres sin conectar en ambos extremos. Los terminales 18 y 35 no se deben conectar entre los dispositivos porque pueden unir los buses de alimentación de ambos. Esto ocasionaría que las alimentaciones competirían por predominar, lo que, como mínimo, causara el quemado de un fusible. Podría también causar una falla fatal en ambos dispositivos.

Al dejar las patitas 17 de los dos equipos sin conectar, las tierras de sus chasis se mantienen aisladas entre si. El aislamiento mejora las posibilidades de que uno de los dispositivos sobreviva en la improbable eventualidad de que se produzca en el otro un cortocircuito interno que lleve su chasis a la tensión de línea antes de que un fusible tenga oportunidad de actuar. También permite que un dispositivo que tenga la tierra de su chasis conectada a su tierra de señal pueda conectarse con un equipo que debe tener esas dos tierras separadas.

La conexión de las terminales 15 y 34 es opcional, ya que esos terminales no los soporta la norma. Si utiliza conectores de desplazamiento de aislamiento (o IDC, como se denominan comúnmente) la desconexión de esos terminales causa más inconvenientes que beneficios. Si utiliza conectores de tipo de rebordado o de soldadura, ignore sencillamente esas terminales para ahorrar tiempo.

Para desconectar cualquier conexión cuando utiliza IDC, una simplemente los dos conectores de la forma en que lo haría normalmente, y rebane con cuidado una pequeña sección de cada cable no deseado con un cuchillo.

Otro trabajo común de cableado es conectar un puerto de computadora DB-25 a una terminación compatible con Centronics de 36 terminales. Si va a utilizar conectores de rebordado o soldadura, deberá hacer uno como el indicado en la tabla 4.

Conexiones estándar de cables IBM	
Conector DB-25	Conector tipo Centronics
1-14	1-14
15	32
16	31
17	36
19	19
25	30

TABLA 4
Conexiones estándar de cables IBM

Obsérvese en la tabla 4 que las conexiones de las terminales 1 -14 son conexiones de terminal a terminal. Si utiliza conectores IDC y cable tipo cinta de 25 conductores, puede aprovechar eso y crear un cable adecuado en pocos minutos. Comience posicionando el terminal 1 de ambos conectores sobre el mismo borde del conector de cinta (consulte la Fig. 5) antes de apretarlos. Con esto quedaran 11 terminales sin conectar en el conector Centronics, lo cual es correcto. El truco consiste en cortar una pequeña sección del cuarto cable hacia adentro a partir del cable de la terminal 1. El circuito abierto permitirá que la entrada de error de la computadora se encuentre en estado alto (estado "sin error") y ¡eso es todo!.

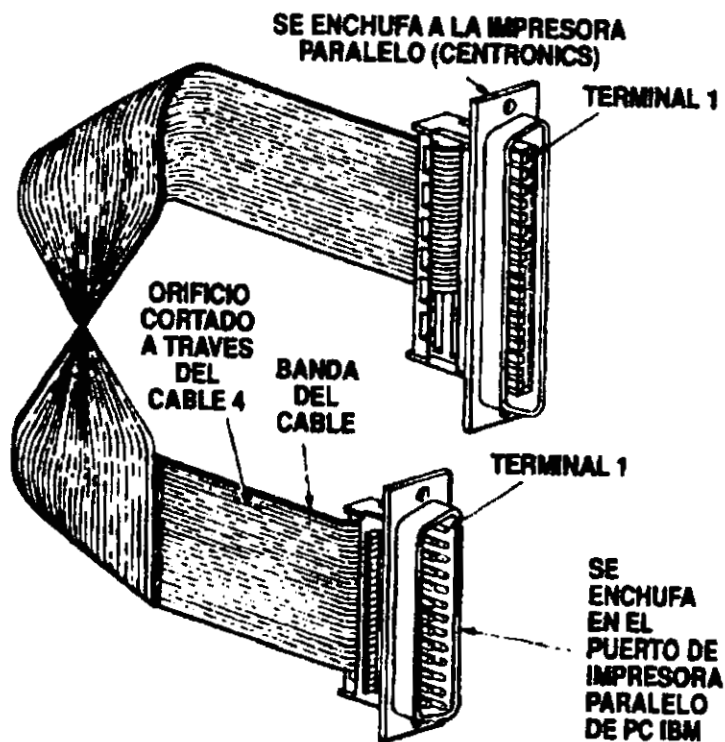


FIGURA 5: Es muy sencillo preparar un Cable para unir un puerto IBM compatible con un periférico compatible con Centronics utilizando dos conectores IDC y un poco de cable de cinta. El truco es cortar una sección del cable hacia adentro desde la terminal 1.

CAPITULO II

CIRCUITOS PARA EL PROCESO DE DATOS

II.1 CIRCUITOS LÓGICOS

La diferencia principal entre las operaciones analógicas y digitales es la manera de utilizar la recta de carga. En los circuitos analógicos se pueden usar puntos adyacentes en la recta de carga, por lo que la tensión de salida es una sucesión sin interrupciones o continua de valores. A causa de esto, la tensión de salida puede tener un número infinito de ellos. Una manera de obtener una operación analógica es aplicar una entrada senoidal. La tensión de entrada continuamente cambiante produce una tensión de salida que también varía continuamente.

Los circuitos digitales son diferentes. Casi todos ellos están diseñados para la operación de dos estados. Esto significa el uso de dos puntos no adyacentes en la recta de carga, que corresponden respectivamente a la saturación y al bloqueo o corte. En consecuencia, la tensión de salida tiene solamente dos estados (valores), que son alto o bajo. Una manera de obtener una operación digital es aplicar una entrada de onda cuadrada. Si es suficientemente grande, este tipo de entradas excita a los transistores en los estados de saturación y corte, produciendo una salida de dos estados.

El puerto paralelo maneja niveles de voltaje lógicos, 2.4 – 5 volts (1 lógico) y 0.8–2.3 volts (0 lógico).

II.1.1 Circuitos para el proceso de datos

Los circuitos lógicos que procesan datos binarios. Comienza por ser los multiplexores, los cuales son circuitos que pueden seleccionar una de varias entradas; Estos se emplean como diseño alternativo a la solución de suma de productos. A esto le siguen los demultiplexores, decodificadores, aunque también son CI digitales realizan funciones diferentes a la multiplexión, también dentro de los circuitos para procesar datos existen circuitos simples como son las compuertas lógicas que realizan diferentes funciones que pueden ser inversores, compuertas AND¹ y OR² o una combinación de estos.

¹ Compuerta AND: Su función a realizar es la de multiplicar números binarios.

² Compuertas OR: Su función es la suma de números binarios.

II.1.2 Circuitos TTL

En 1964 Texas Instrument introdujo la lógica transistor-transistor (TTL), que es una familia ampliamente utilizada de dispositivos digitales. La familia TTL es rápida, poco costosa y fácil de usar. Durante el transcurso de este capítulo se mencionan varios tipos de TTL utilizados en el proyecto.

Tecnologías y familias

Integración en pequeña escala (SSI) se refiere a circuitos integrados que tiene menos de 12 puertas en el mismo chip.

Integración en media escala(MSI) significa que hay de 12 a 100 puertas en cada chip.

Integración de gran escala (LSI) hace referencia a más de 100 puertas por chip.

Las dos técnicas básicas para la fabricación de circuitos integrados son la bipolar y la metal-oxido-semiconductor (MOS); En la primera se fabrican los transistores bipolares sobre un chip; en la segunda se fabrican transistores de efecto de campo MOS (MOSFET).

Familias bipolares

En la categoría bipolar se encuentran estas familias básicas:

- Lógica diodo-transistor (DTL)
- Lógica transistor-transistor (TTL)
- Lógica de emisor acoplado (ECL)

La DTL utiliza diodos y transistores; Este diseño, que en un tiempo fue muy popular, está ahora obsoleto. La familia TTL utiliza transistores casi exclusivamente; Se ha convertido en la familia más popular de chips de SSI y MSI, ECL, que es la familia lógica más rápida, se emplea en aplicaciones de alta velocidad.

Familia MOS

Estas familias pertenecen a la categoría MOS:

- PMOS MOSFET de canal p
- NMOS MOSFET de canal n
- CMOS MOSFET complementario

Familia PMOS es el tipo más antiguo y lento, esta actualmente obsoleto. NMOS domina el campo de LSI de los microprocesadores y memorias. CMOS, un dispositivo push-pull de MOSFET de canal n y canal p se emplea ampliamente cuando es necesario un consumo de baja potencia, como en las calculadoras de bolsillo.

II.2 Tipos de circuitos lógicos

Los circuitos integrados de compuertas lógicas contienen las propiedades de resistores, diodos, transistores de juntura bipolares y semiconductores de compuerta aislada o semiconductor de metal oxido de transistores de efecto de campo (IGFET o MOSFET);

Las compuertas usualmente están encapsuladas en grupos de dos o cuatro en el mismo CI. Cada compuerta puede tener de 2 a 8 entradas.

Las compuertas lógicas en los circuitos integrados se clasifican en familias de acuerdo con las funciones principales que se usan y como se conectan. Sus abreviaturas comunes se describen a continuación.

DTL La abreviatura DTL es para una lógica diodo-transistor. Los diodos se usan para la compuerta Y o la compuerta O, y los transistores sirven como inversores y amplificadores. En realidad, cada diodo puede hacerse como la juntura de emisor-base de un transistor integrado. La configuración básica DTL es la compuerta NO Y, la salida de la cual es BAJA cuando todas las entradas son ALTAS.

TTL La abreviatura TTL, es utilizada para la lógica transistor-transistor en circuitos integrados. En lugar de diodos para la compuerta, se usan para las entradas un transistor con múltiples emisores, como configuración básica es la compuerta NO-Y.

ECL Esta abreviatura es para la lógica de emisor-acoplado. Las señales de entrada se aplican a la base de los transistores se parados con resistencia R_E de emisor común para formar la compuerta. A este tipo también se le conoce como lógica en modo de corriente (CML), porque los transistores de entrada conmutan la corriente a través de R_E .

TTL Schottky El método TTL Schottky es la misma lógica de transistor-transistor, pero esta cuenta con un diodo Schottky para enclavar el voltaje de la junta base-colector para evitar la saturación del transistor, con lo cual se obtiene una rápida conmutación entre los estados de encendido y apagado.

CMOS Abreviatura que involucra a los transistores MOS de canal P y canal N. La ventaja de este tipo es que no necesita resistores, lo cual permite una muy alta densidad de compuertas en el bloque del CI, junto con una disipación de potencia baja.

II.3 CONVERTIDORES

II.3.1 Convertidores Digital a Analógico (D/A)

Los convertidores (D/A) digital a analógica y analógica a digital (A/D) forman dos aspectos muy importantes del proceso de datos digital. La conversión digital a analógica implica la traducción de la información digital a información equivalente en forma analógica. Por ejemplo, la salida de un sistema digital se podría cambiar a la forma analógica con la finalidad de excitar un registrador de pluma. Analógicamente, es necesaria una señal análoga para servomotores que exciten los brazos de cursor de un trazador de gráficos o plotter. En este respecto algunas veces se considera un convertidor D/A (DAC) como dispositivo decodificador.

La conversión de una señal digital a una señal analógica es un proceso sencillo que se considera más fácil que una conversión A/D, usualmente un convertidor de A/D forma parte integrante de un convertidor D/A.

Un tipo de convertidor digital-analógico es el que se muestra en la figura 6, este tipo de convertidor emplea una compuerta de aislamiento, que se utiliza para convertidores D TTL/A, en el que el voltaje máximo de salida es 15V y cuando se encuentra en estado BAJO, el valor de esta se encontrara entre 0.1 y 0.2 volts por encima de la referencia de tierra.

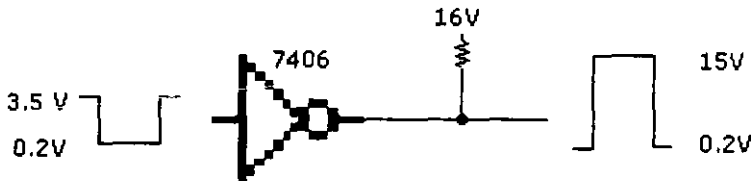


FIGURA 6: CONVERSION DE NIVELES DE VOLTAJE TTL A NIVELES D/A

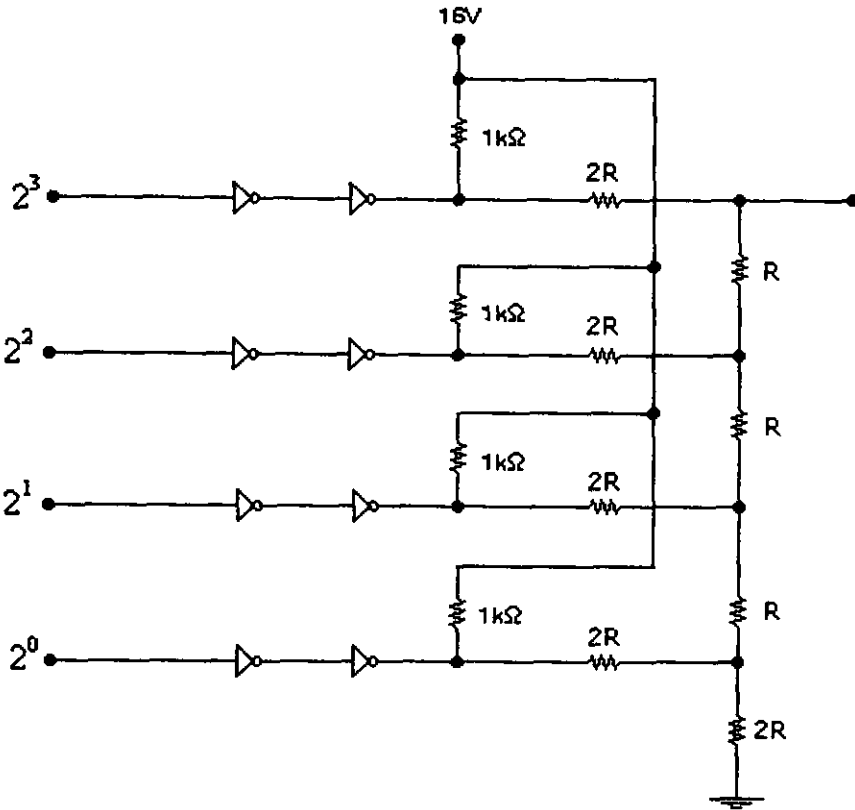


FIGURA 7: CONVERTIDOR D/A 2R TTL

II.3.2 Convertidor Analógico a Digital (A/D)

El proceso de cambiar una señal analógica a una señal equivalente digital se realiza mediante un convertidor A/D. Por ejemplo se utiliza un convertidor A/D para convertir las señales analógicas de salida de los transductores (medida de temperatura, presión etc.) en señales digitales equivalentes. Entonces estas señales estarán en forma adecuada para introducirlas en sistemas digitales. Un convertidor A/D (ADC) se denomina generalmente dispositivos codificador ya que se utiliza para codificar señales con el fin de introducirlas en un sistema digital.

Un ejemplo de este tipo de convertidores es el convertidor de cuenta ascendente y comparación, este tipo de arreglo utiliza un comparador de voltaje y un convertidor D/A, el comparador que utiliza es el LM339 y una red resistiva 2R

Las entradas del convertidor D/A son generadas por un contador binario o cualquier circuito que de cómo salidas una cuenta binaria de 0000 hasta 1111, las salidas del convertidor son tomadas en las terminales del 7493, el comparador de voltaje LM339 que se emplea para habilitar o inhabilitar la compuerta NAND que suministra la señal de reloj hacia el 7493.

Cuando en el caso del circuito de la figura se habilita al contador 7493 después de haber sido re-inicializado el contador comienza a generar salidas hasta que el voltaje de entrada al LM339 es igual o menor a la entrada de voltaje analógico en la terminal (-) del LM339, lo cual deshabilita a los pulsos de reloj que entran hacia el contador.

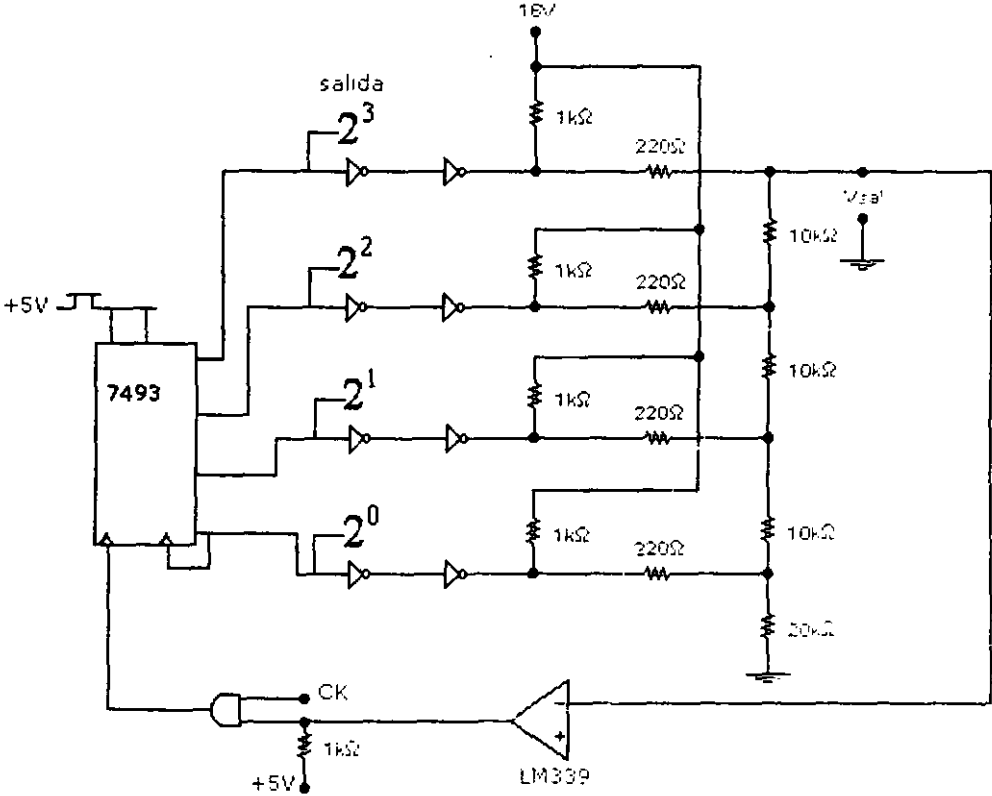


FIGURA 8: CONVERTIDOR A/D DE CUENTA ASCENDENTE Y COMPARACIÓN

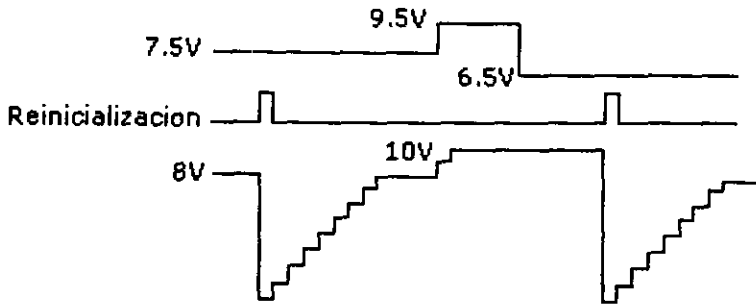


FIGURA 9: FORMAS DE ONDA DEL CONVERTIDOR D/A

Otro tipo de convertido Analógico a Digital es el convertidor A/D de aproximaciones sucesivas; Este tipo de convertidor también emplea un convertidor D/A y un comparador de voltaje; pero trabaja de forma diferente para determinar el número binario que es requerido en el convertidor D/A.

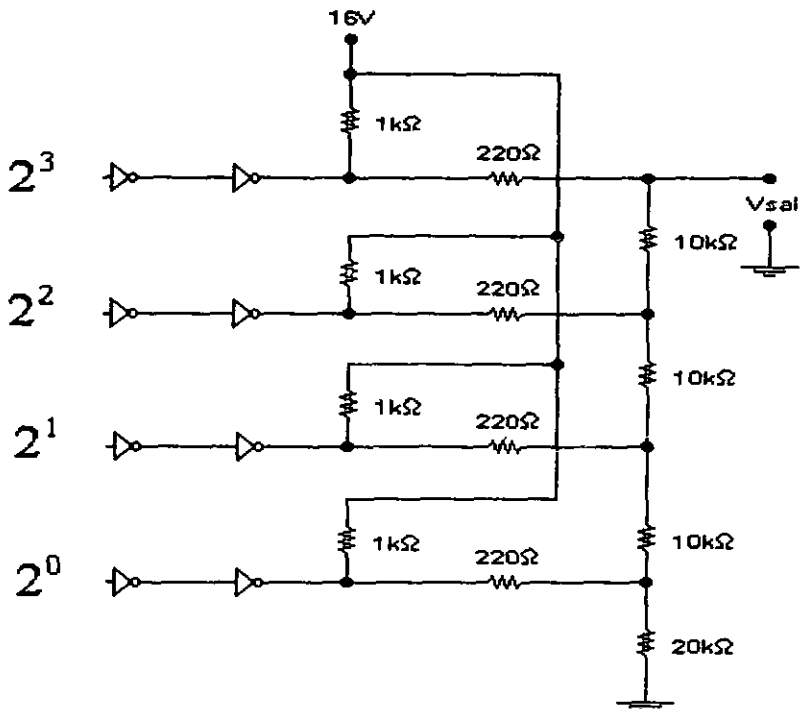


FIGURA 10: CONVERTIDOR A/D DE APROXIMACIONES SUCESIVAS

II.4 Codificación y decodificación de números binarios

La codificación significa cambiar los números decimales en forma binaria. En la decodificación la información digital se convierte otra vez en números decimales. El proceso de codificación puede usarse también para letras y símbolos.

II.4.1 Circuitos integrados CI codificadores y decodificadores

En los codificadores y decodificadores se usan generalmente en circuitos integrados. Los decodificadores convierten la entrada binaria a un código especial que enciende los segmentos individuales del indicador visual digital de 7 segmentos (Display).

El CI 7447 (FIGURA 11), es un ejemplo de circuito decodificador de una cuenta binaria a una conversión de código (BCD), y sus salidas son mostradas a través de un visualizador digital.

SN54/74LS47

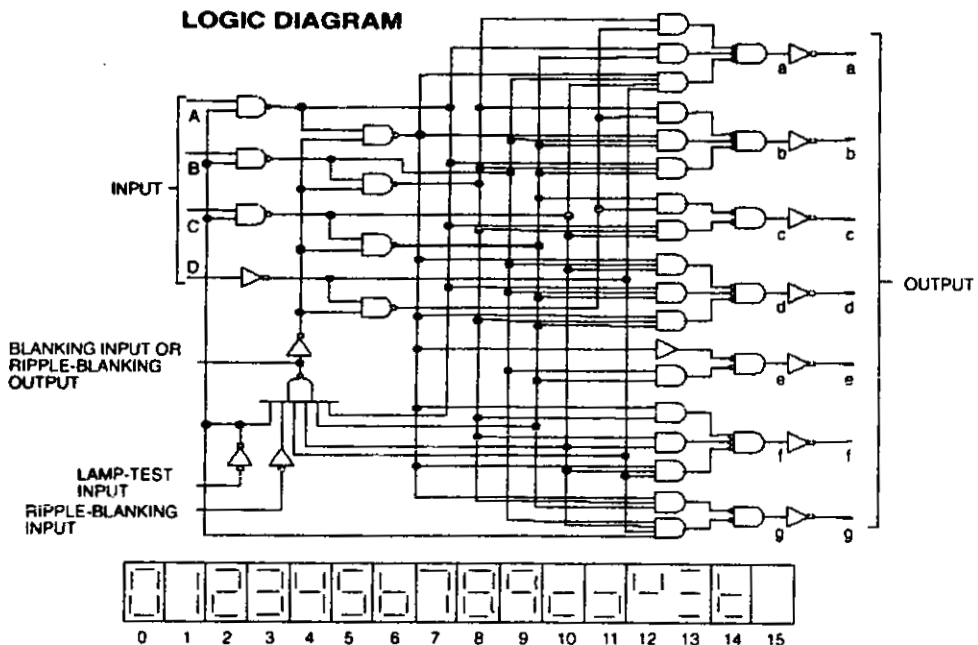


FIGURA 11: CI 74LS47, VISUALIZACION DE SALIDAS EN UN DISPLAY DE 7 SEGMENTOS

II.5 REGISTROS

II.5.1 Registros de corrimiento

Como ejemplo de la acción de corrimiento puede verse en el indicador o anunciador visual digital de una calculadora. Para meter el número 25, se oprime la tecla 2 y se suelta. El número 2 aparece en el anunciador visual. A continuación se oprime la tecla 5 y se libera; esto provoca que el número 2 se corra un lugar a la izquierda y que aparezca 25 en el anunciador visual.

El circuito de un registro de corrimiento consiste básicamente en flip-flops usualmente del tipo D con una terminal de reposición o borrado además de compuertas lógicas para funciones especiales. Se usan un flip-flop para cada BIT en la información binaria. El corrimiento significa transferir la información de cada flip-flop (FF) de su salida Q a la entrada del siguiente FF.

Un circuito secuencial temporizado consta de un grupo de flip-flops y compuertas combinacionales conectadas que forman una trayectoria de retroalimentación. Los flip-flops son secuenciales porque, cuando están ausentes, el circuito se reduce a un circuito combinacional puro (siempre que no haya trayectoria de retroalimentación).

Un circuito MSI que contiene celdas de almacenamiento en su interior es, por definición un circuito secuencial, uno de estos circuitos MSI son los registros.

Un *registro* es un grupo de celdas de almacenamiento binario adecuadas para mantener información binaria. Un grupo de flip-flops constituye un registro, ya que cada flip-flop es una celda binaria capaz de almacenar un BIT de información. Así pues tenemos que un conjunto de n flip-flops constituye un registro de información binaria de n bits.

Además de los flip-flops, un registro puede tener compuertas combinacionales que realicen ciertas tareas de procesamiento de datos. Dicho en otras palabras, un registro consta de un grupo de flip-flops y compuertas que efectúan su transición de información binaria.

Los flip-flops mantienen información binaria y las compuertas controlan cuando y cómo se transfiere información nueva al registro.

Existen varias disposiciones de registros de circuitos MSI. El registro más simple consta de flip-flops sin ninguna compuerta externa, como se muestra en la Fig. 12.

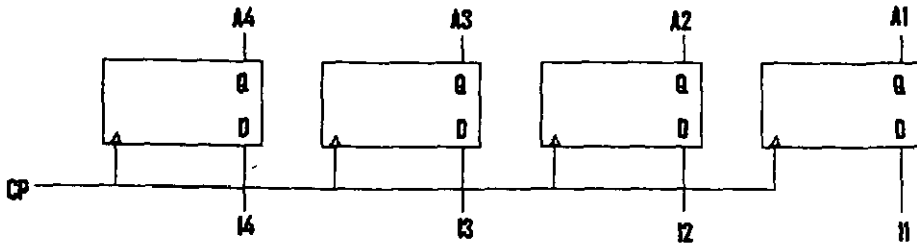


FIGURA 12: REGISTRO CON ARREGLO DE FLIP-FLOPS TIPO D

Se muestra a dicho registro construido por flip-flops tipo D y una entrada común de pulsos de reloj. La entrada de pulsos de reloj¹, CP, capacita todos los flip-flops, de modo que la información presente disponible en las cuatro entradas puede transferirse al registro de 4-BIT.

11.5.2 Registro con carga paralela

A La transferencia de bits de información nueva a un registro se le da el nombre de *cargar* el registro. Esta carga de bits se hace en paralelo si todos los bits en el registro se cargan en forma simultanea con un solo pulso de reloj. Un pulso aplicado a la entrada CP del registro mostrado en la figura 12, cargara a los bits en las cuatro entradas en paralelo hacia las cuatro salidas en paralelo.

Pero si se quiere dejar a los bits en el registro sin cambio alguno, se necesita inhibir el pulso de reloj, es decir, la entrada CP actúa como una señal de habilitación que controla la carga de bits nuevos hacia el registro. Cuando CP pasa a 1 la información de entrada se carga en el registro. Si CP permanece en 0, el contenido del registro no cambia. El cambio de estado en las salidas ocurre en el borde positivo del pulso \lrcorner de la señal CP.

La mayoría de los sistemas digitales tienen un reloj generador de un tren continuo de pulsos, estos pulsos de reloj se aplican a los flip-flops y registros en el sistema. La función de este generador de pulsos es actuar para suministrar un ritmo constante a todas las partes de sistema. Una señal mediante un control decide que pulsos tendrán efecto en un registro particular.

¹ Pulso de reloj (CP ó CK): Señal digital que cuenta con una serie de unos y ceros intercalados entre sí, que son transmitidos hacia una entrada de un circuito secuencial.

La anexión de una compuerta AND o de compuertas lógicas en la trayectoria de los pulsos de reloj produce retardos de propagación entre el reloj generador de pulsos y las entradas de reloj de los flip-flops. Para sincronizar el sistema en forma completa. Debe tenerse la seguridad de que todos los pulsos de reloj llegan al mismo tiempo a todas las entradas de todos los flip-flops, para que todos puedan cambiar en forma simultanea.

Un registro de 4 bits con una entrada de control de carga usando flip-flops RS se muestra en la figura 13.

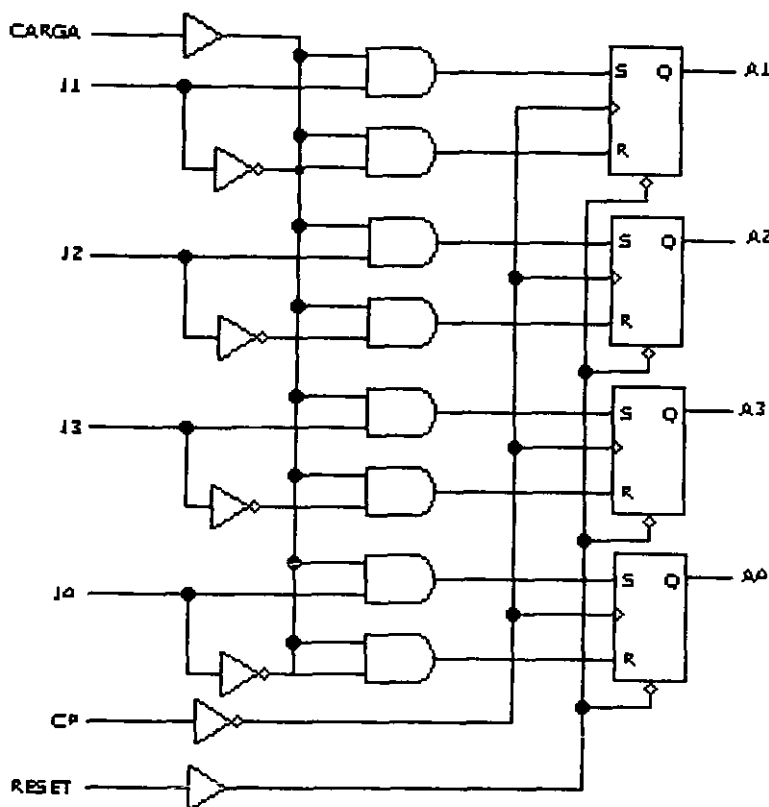


FIGURA 13: REGISTRO DE 4 BITS CON CARGA PARALELA CON ARREGLO DE FLIP-FLOPS RS

La entrada de reloj (CP) del registro recibe pulsos sincronizados continuos que se aplican a todos los flip-flops. El inversor en la trayectoria CP provoca que todos los flip-flops se disparen por el borde negativo de los pulsos que llegan a la entrada CP.

La entrada Despeje (Reset) va hacia una terminal que hace que todos los flip-flops a través de un buffer; Cuando esta terminal pasa a 0, el flip-flop se despeja en forma asíncrona, lo cual hace que todos los registros estén en 0 con lo cual obtenemos una señal de reset hacia el circuito.

La entrada de *carga* pasa a través de una compuerta buffer (para reducir el cargado) y a través de una serie de compuertas AND a las entradas R y S de cada flip-flop. Aunque siga existiendo un pulso de reloj en la entrada CP, la entrada de carga es la que controla y habilita la operación del registro. Las dos compuertas AND y el inversor asociado con cada CI determinan los valores de R y S. Por lo que si la entrada en la línea de carga es 0 en R y S no ocurre ningún cambio de estado por ningún pulso de reloj existente en la entrada CP, pero en cambio si la entrada de carga cambia a 1, las entradas I₁ hasta I₄ especifican que información binaria se carga en el registro en el siguiente pulso de reloj en la entrada CP.

Puesto que todos los bits de entrada se cargan en forma simultanea tenemos que a este tipo de transferencia se le conoce como *carga-paralela*. Un registro con carga paralela puede construirse también con flip-flops tipo D como se muestra en la figura 14.

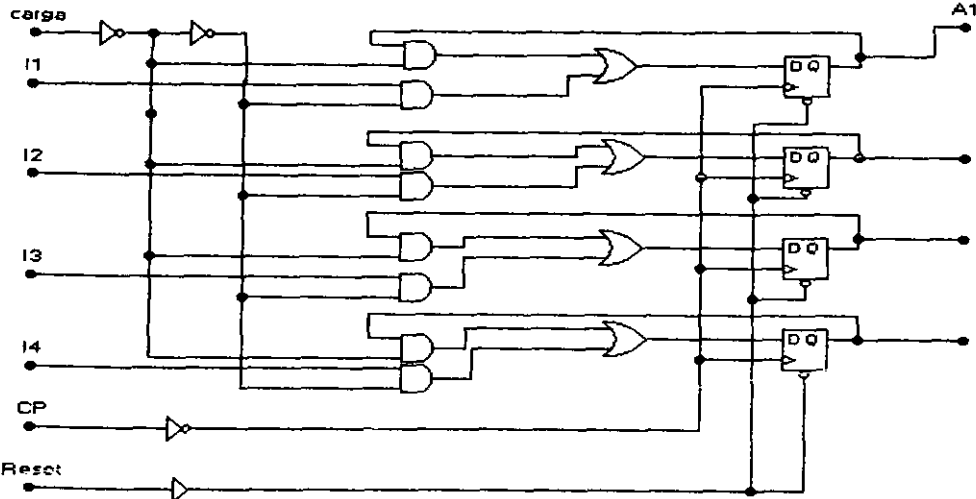


FIGURA 14: REGISTRO CON CARGA PARALELA CON ARREGLO DE FLIP-FLOPS D

Debido a que los registros están dispuestos como circuitos MSI a veces es necesario implementar una lógica secuencial que forme parte del registro; Donde el estado presente del registro y las entradas externas determinan el estado siguiente del registro y los valores de las salidas externas, por lo que parte del circuito combinacional determinan el estado siguiente y la otra parte otorga las salidas.

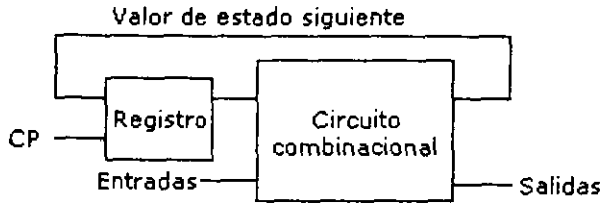


FIGURA 15: DIAGRAMA BLOQUES DE UN REGISTRO CON LOGICA COMBINACIONAL

II.5.3 Registros con Corrimiento

Se conoce como *registro con corrimiento* al registro capaz de correr su información binaria ya sea desplazándola hacia la derecha o a la izquierda, este tipo de registros consta de una cadena de flip-flops conectados en cascada, con las salidas de un flip-flop conectada a la entrada del siguiente flip-flop; Estos al recibir un pulso de reloj en forma paralela causa el corrimiento de un estado a otro. La salida Q de un flip-flop que se conecta a la entrada D del flip-flop a su derecha en cada pulso de reloj corre el contenido de registro una posición de BIT a la derecha. La entrada serial determina que bits se cargan al flip-flop que se encuentra al extremo izquierdo cuando se hace un corrimiento a la derecha.

Transferencia Serial: La transferencia serial ocurre cuando la información se transfiere y se manipula un BIT a la vez dentro de un sistema digital, la información es transferida de un registro a otro, corriendo los bits de un registro hacia el registro destino. La transferencia serial de un registro A hacia B se hace mediante los registros de corrimiento por lo que un flip-flop cede al siguiente el BIT que contiene hasta la salida seriada S0 y en la entrada seriada SI determina que BIT es el que entra para que no exista una pérdida de información.

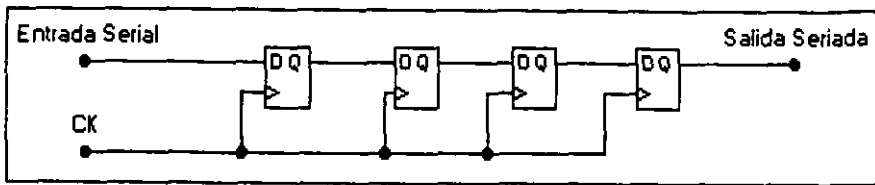


FIGURA 16: TRANSFERENCIA SERIAL CON FLIP-FLOPS D

La diferencia existente entre los dos modos de operación (serial y paralelo), existe en que en el modo paralelo, se esta disponible la información de todos los bits de un registro y todos los bits pueden llegar a ser transferidos en forma simultanea durante un pulso de reloj; En el modo serial, los registros tienen una sola entrada serial y una salida serial, la información en bits es transferido un BIT a la vez.

NOTA: Las computadoras pueden operar en modo serial, o en modo paralelo, o puede existir y también una combinación de ambos modos.

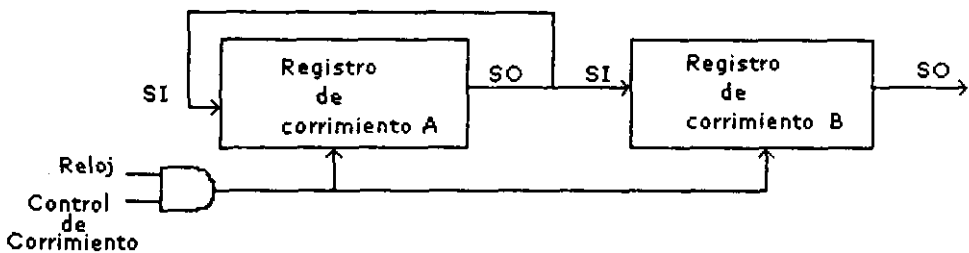


FIGURA 17: DIAGRAMA A BLOQUES

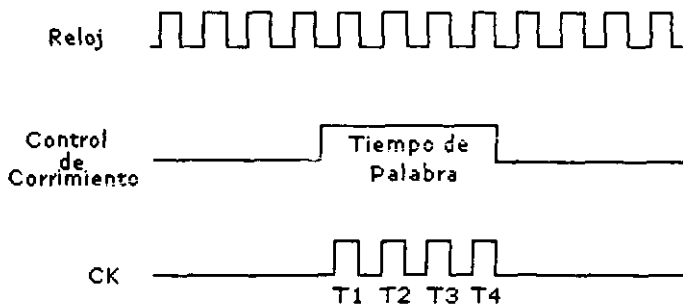


FIGURA 18: DIAGRAMA DE TEMPORIZADO

II.5.4 Registros con corrimiento bidireccional con carga paralela

Esta clase de registros puede usar datos seriales y paralelos para realizar una conversión de datos en paralelo a datos seriales y datos seriales a datos paralelos.

Dentro de estos registros proporcionan las terminales necesarias de entrada y salida para la transferencia en paralelo, que contienen las capacidades tanto de corrimiento a la derecha como corrimiento a la izquierda. Tienen como característica:

1. Control de despeje (reset)
2. una entrada de pulsos de reloj (CP), para la sincronización de las operaciones del registro
3. Un control de corrimiento a la derecha que habilita la operación de corrimiento a la derecha y las líneas de entrada y salida serial asociadas con este tipo de corrimiento
4. Un control de corrimiento a la izquierda que habilita la operación de corrimiento a la izquierda y las líneas de entrada y salida serial asociadas con este tipo de corrimiento
5. Un control de carga paralela que habilite n entradas de transferencia paralela
6. n numero de líneas de salidas en paralelo
7. Un estado de control que mantenga sin cambio la información contenida en el registro.

La figura 19 Muestra a un registro con todas las capacidades mencionadas. Un ejemplo de este tipo de registro es el CI 74194.

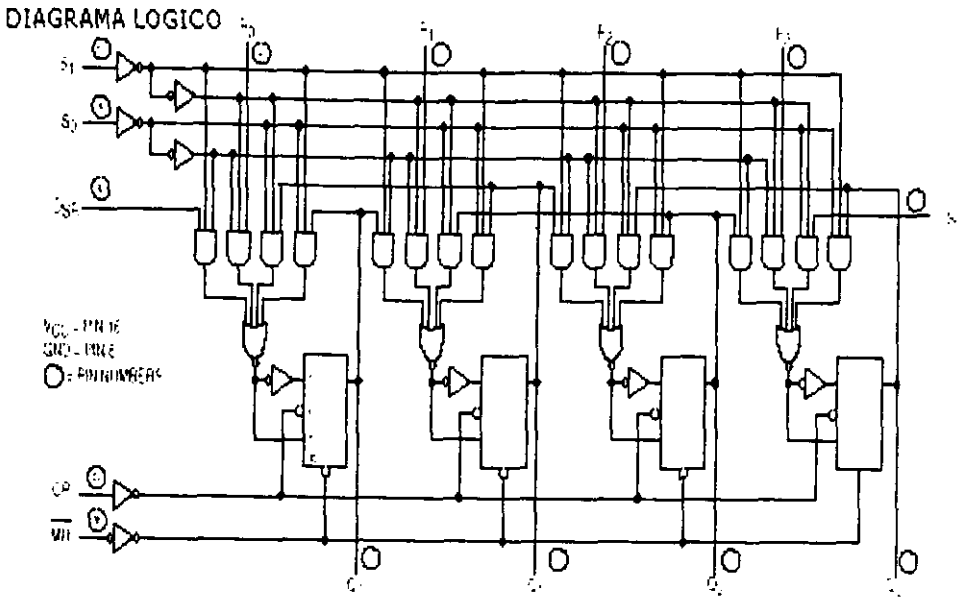


FIGURA 19: CI 74194

II.6 Multiplexores y demultiplexores

Los dispositivos conocidos como multiplexores y demultiplexores son los equivalentes electrónicos de un interruptor selector rotatorio de un polo y n numero de tiros.

La idea básica de cómo trabaja un multiplexor electrónico se muestra en la figura 20.

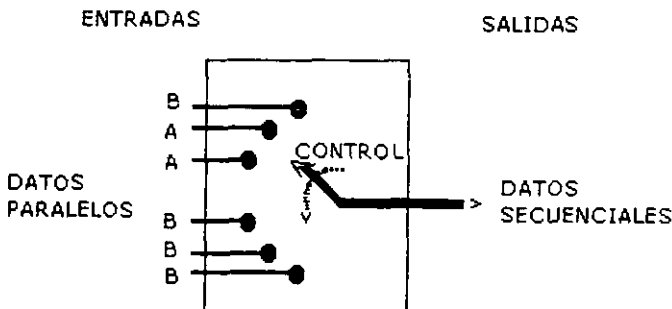


FIGURA 20: MULTIPLEXOR

Cualquiera de las entradas puede seleccionarse por el control. La información en esa entrada será transferida a la salida. La rotación rápida del control cambia la palabra de entrada en paralelo a información en serie. Varios bits tomados juntos forman una palabra. La palabra en la figura en forma paralela es BAABBB. Al multiplexor electrónico también se le llama selector de información.

La operación de un demultiplexor es lo opuesto del multiplexor. En la figura 21 se muestra una cadena de información en serie de bits transferida a las salidas correctas. Como resultado, la información en serie se cambia a información en paralelo. El demultiplexor también se llama distribuidor o decodificador.

Una de sus funciones es la transmisión de información por líneas telefónicas, cable o radio. En la transmisión es mejor tener la información en forma seriada, de modo que sólo se necesita una línea o canal, en el extremo receptor la información se restituye como información en paralelo, volviéndola a su forma original.

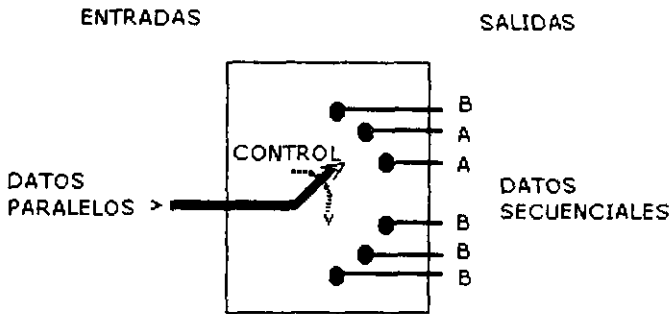


FIGURA 21: DEMÚLTIPLEXOR

Utilizamos al demultiplexor como interruptor digital permite hacer la conexión de una entrada con una de las muchas líneas de salidas posibles, la línea de salida que sea seleccionada podrá ser conectada con la línea de entrada, la cual es determinada por un número binario colocando digitalmente a la entrada con la salida seleccionada.

El multiplexor es un dispositivo que selecciona un canal de entrada por medio de una señal binaria, la cual coloca a la entrada seleccionada con una única salida del dispositivo, como se puede observar en las dos figuras anteriores un multiplexor es lo opuesto a un demultiplexor.

Como ejemplos de multiplexor y demultiplexores en CI tenemos al 74138 que es un demultiplexor 3 a 8¹ y al 7415 que es un multiplexor 8 a 1².

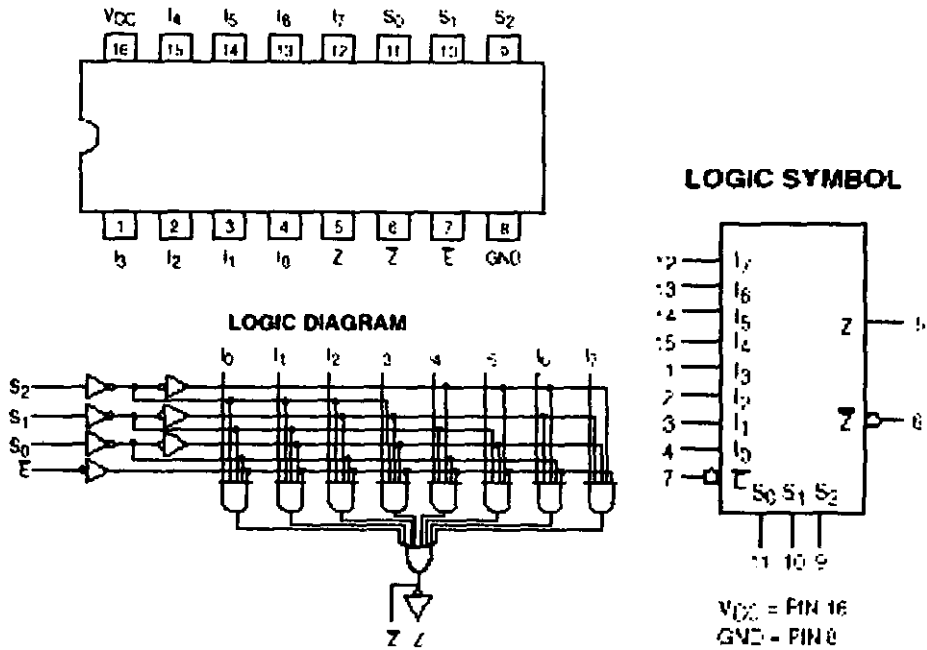


FIGURA 22: CI 74151

¹ 3 a 8: son 3 entradas que seleccionan una de 8 salidas

² 8 a 1: son 8 entradas donde se selecciona solamente una para ser dirigida hacia la salida

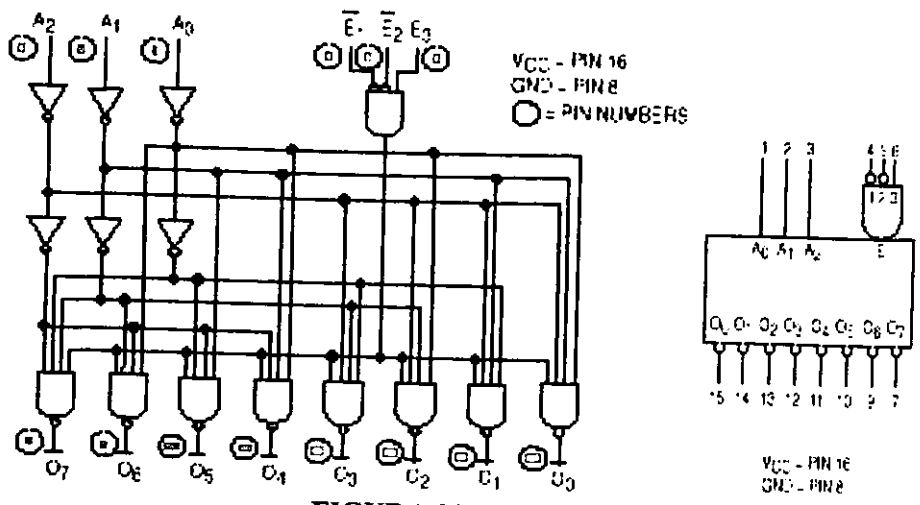


FIGURA 23: 74LS138

II.7 SELECCIÓN DEL CIRCUITO

A continuación mencionaremos un circuito que constituya la entrada y salidas de datos del puerto paralelo (LPT1).

Como se mencionó el circuito consta de CI TTL, ya que aparte de su facilidad de encontrarse, también son de bajo costo y contienen las características necesarias (bajo consumo de voltaje y corriente) para manejar bits de datos procedentes del puerto paralelo de una computadora.

Primeramente observando las características de los circuitos TTL que son necesarios para el circuito (71138,71194,7100...), dándonos cuenta de la manera de trabajar de cada uno de ellos.

Los 8 bits de salida que provienen del CPU a través del puerto paralelo, serán divididos en bits que utilizaremos como bits de direccionamiento, y bits de datos, para el circuito dado el siguiente arreglo:

1 bit bloqueo	3 bits de control	4 bits de datos
---------------	-------------------	-----------------

TABLA 5: DISTRIBUCIÓN DE BITS

Veamos lo que nos involucra nuestro circuito, su función, partes que lo constituyen y algunas de sus características eléctricas y de procesamiento digital, y como poder aplicar todas estas características.

El circuito de salida utilizara CI de la familia TTL, pero nos hemos de preguntar que clase de circuitos son esos, como son y como se polarizan¹; Hay que tener en cuenta que es necesario conectar al circuito a una fuente de voltaje para que funcione.

Como primer punto un circuito TTL tiene como codificación el número 74 seguido de 2 letras y de 2 a 3 números (depende el caso), que nos indican que clase de circuito es el que se esta utilizando;

74	XXX
----	-----

¹ Polarizar: Termino que indica que voltajes encienden y hacen trabajar un circuito.

Esto puede ser observado dentro de un manual de referencia, esta clase de clasificación puede buscarla en los manuales TTL de electrónica digital, donde se puede observar que la mayoría de estos se polarizan con 5 Volts de CD¹.

Dentro de la clasificación de estos circuitos podemos encontrar que pueden ser compuertas lógicas² de tipo inversor, AND, OR o combinación de ellos, y para saber como son solo observe la siguiente figura:

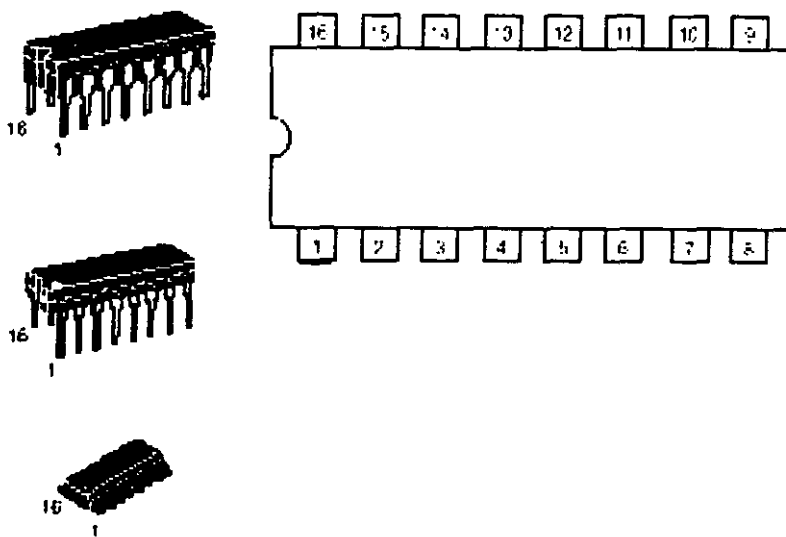


FIGURA 24: ENCAPSULADO DE UN CIRCUITO INTEGRADO (CHIP)

El pequeño hueco curvo, que se observa desde la parte superior del CI³, nos indica que el primer pin⁴ se encuentra en la parte inferior al circuito y la numeración de pines es como se ve en la figura, el número de pines puede llegar a variar dependiendo del circuito y la función que realiza.

¹ CD: Corriente Directa

² Compuerta Lógica: Son circuitos cuyo voltaje de salida puede predecirse de acuerdo con las condiciones de la entrada.

³ CI: Circuito Integrado

⁴ PIN: Terminal o patilla de un CI, que tiene una función específica.

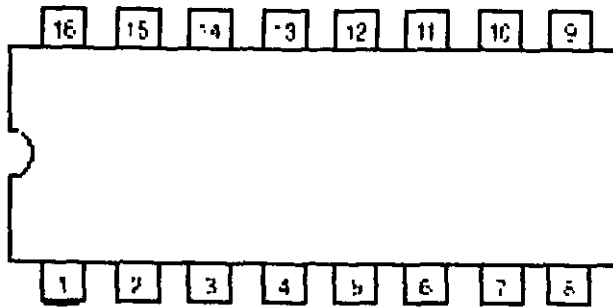


FIGURA 25: NUMERACIÓN DE PINES DE UN CI

Con esto podemos identificar un Integrado TTL, la colocación y disposición de sus pines, para comenzar a desarrollar el circuito.

Primero observaremos como los 3 bits son utilizados para manipular y direccionar a los 4 bits de datos;

Los 3 bits de direccionamiento entregados por la interfaz del puerto paralelo hacia el circuito, serán introducidos en un CI (74138) TTL

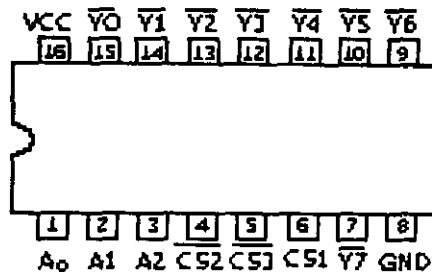
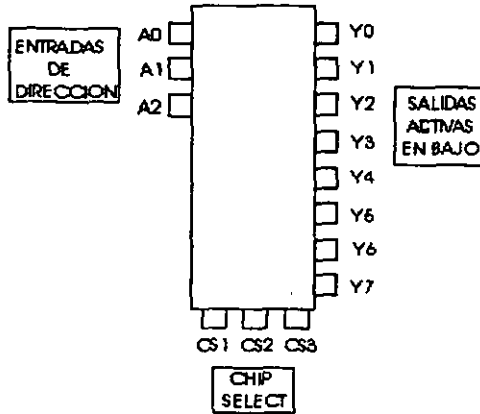


FIGURA 26: CI 74LS138

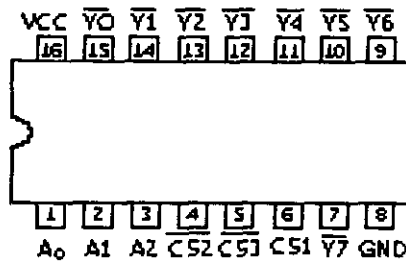
Su funcionamiento es explicado a continuación:

Observemos el diagrama lógico del circuito contenido en la Figura 27(A) y el diagrama de asignación de pines Figura 27(B)



**FIGURA 27(A): CI 74138
DIAGRAMA LOGICO**

DIAGRAMA DE ASINACION DE PINES



**FIGURA 27(B): CI 74138
DIAGRAMA DE ASIGNACIÓN DE PINES**

Tenemos dentro del diagrama lógico 3 líneas de dirección, 3 líneas de selección, y 7 líneas de salidas activadas en bajo, cada línea ocupada por un BIT¹.

Las líneas de selección CS₂, CS₃ son para que se habilite el chip deben estar a nivel bajo y la línea de selección CS₁ debe estar a nivel alto, cualquier cambio de nivel en estos pines dará como resultado, que no importando las entrada de A₀, A₁, A₂, las salidas Y_n se irán a nivel lógico alto.

¹ BIT Binary digiT (Dígito Binario)

Cuando en CS₁ hay un nivel lógico alto y en CS₂, CS₃ hay un nivel lógico bajo se habilita al chip y este acepta que se activen las salidas Y_n por medio de las entradas A₀, A₁, A₂, por ejemplo:

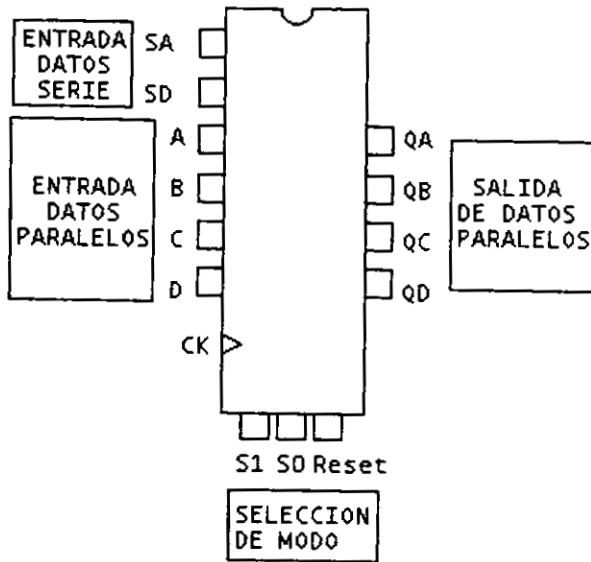
Si en A₀, A₁, A₂, entran los niveles lógicos 0,0,1 respectivamente tendremos que la salida activada correspondiente a esta entrada será la salida Y₁; Si en A₀, A₁, A₂, encontramos las entradas 1,0,1 respectivamente tendremos que la salida activada en bajo será la Y₅. Esto lo podemos observar con mas claridad en la tabla de funcionamiento del CI.

ENTRADAS						SALIDAS							
CS ₁	CS ₂	CS ₃	A ₂	A ₁	A ₀	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
X	X	H	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

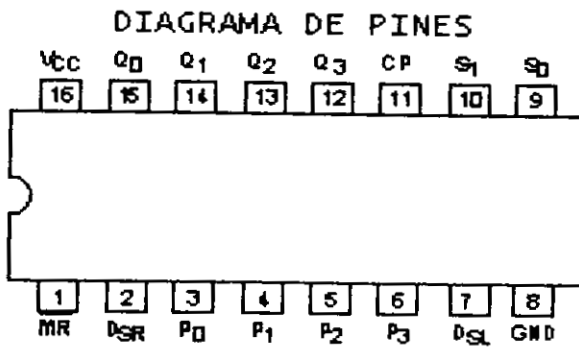
TABLA 6: (CI 74138) TABLA DE VERDAD

El CI 74194 será el circuito que recibirá los 4 bits de datos que serán direccionados por los 3 bits que activan las salidas del decodificador 74138, de este circuito (74194) que es un circuito de Registro de 4 - bits Bidireccionales, solo se utilizara las funciones de transmisión en paralelo y hold.

Para esto tenemos el siguiente diagrama lógico (figura 28A) y diagrama de asignación de pines (figura 28B)



**FIGURA 28A: CI 74194
DIAGRAMA LOGICO**



**FIGURA 28B: CI 74194
DIAGRAMA DE ASIGNACIÓN DE PINES**

Este circuito trabaja mediante la siguiente tabla de verdad:

MODO de SELECCION— TABLA de VERDAD

MODO DE OPERACION	ENTRADAS						SALIDAS			
	MR	S ₁	S ₀	D _{8R}	D _{8L}	P _n	Q ₀	Q ₁	Q ₂	Q ₃
RESET	L	X	X	X	X	X	L	L	L	L
HOLD	H	1	1	X	X	X	Q ₀	Q ₁	Q ₂	Q ₃
DESPLAZAMIENTO IZQ.	H	1	1	X	1	X	Q ₁	Q ₂	Q ₃	L
	H	1	1	X	1	X	Q ₁	Q ₂	Q ₃	H
DESPLAZAMIENTO DER.	H	1	1	1	X	X	L	Q ₀	Q ₁	Q ₂
	H	1	1	1	X	X	L	Q ₀	Q ₁	Q ₂
CARGA PARALELA	H	1	1	X	X	P _n	P ₀	P ₁	P ₂	P ₃

L = Nivel de Voltaje BAJO

H = Nivel de Voltaje ALTO

X = NO Importa

1 = nivel de voltaje BAJO en una transición de pulso de reloj de nivel BAJO hacia ALTO

h = nivel de voltaje ALTO en una transición de pulso de reloj de nivel BAJO hacia ALTO

TABLA 7: (CI 74194) TABLA DE VERDAD

Se utilizara de esta tabla las funciones de HOLD y Carga Paralela, que serán importantes en la aplicación de mantener o cargar datos desde el puerto paralelo hacia el circuito.

Otro circuito importante de ser mencionado es el 7404, que es un circuito inversor de 6 entradas, el cual tiene como función invertir las entradas lógicas que existan en él, es decir si entra un 1 lógico a la salida obtendremos un 0 lógico y viceversa.

La declaración de pines en el circuito es el siguiente:

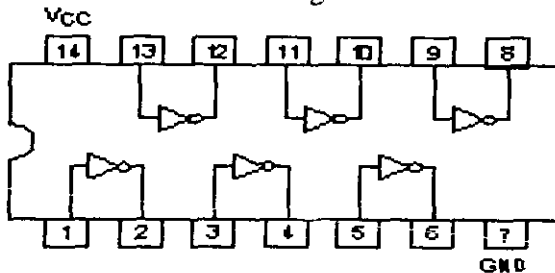


FIGURA 29: CI 7404 ASIGNACION DE PINES

II.8 DISEÑO

Dadas las características de estos circuitos TTL y de la interfaz RS232, podemos plantear y ensamblar un circuito que reciba los bits por el puerto paralelo:

Sabemos que tenemos como salida del puerto paralelo 8 bits de los cuales 3 bits serán utilizados para controlar a los demás bits procedentes del puerto paralelo;

Tenemos que al recibir a estos 3 bits que llamaremos de control, tendrán que ser recibidos por el circuito Integrado 74138, (circuito decodificador/demultiplexor), este se polarizara con 5 volts, según su diagrama de asignación de pines, conectaremos los 3 bits de control hacia las terminales (A_0 , A_1 , A_2) de los pines 1, 2, 3 respectivamente, las terminales CS_1 , CS_2 , CS_3 de los pines 6,4,5 serán conectados hacia valores de nivel lógico alto, bajo, bajo, respectivamente, para así poder habilitar el funcionamiento del chip.

Las salidas Y_n , serán enviadas hacia el circuito integrado 7404 para invertir los estados de las salidas Y_n , estas señales se dirigirán hacia las terminales 9 y 10 del circuito integrado 74194.

Dentro Del circuito integrado TTL 74194 que es un CI de Registro tendremos que de nuestro grupo de bits hemos considerado 4 bits de datos estos se conectaran en los pines de entrada P_0 , P_1 , P_2 , P_3 ;

De acuerdo con la tabla de funcionamiento de este circuito (74194) y la colocación de pines, después de haber sido polarizado a V_{cc} y GND los bits de datos que se encontraran en los pines asignados como P_0 , P_1 , P_2 , P_3 estarán en las terminales 3, 4, 5, 6 respectivamente, Las terminales 2 y 7 que son asignados como los pines de SR y SL se conectaran hacia GND, y los pines S_1 , S_0 , que se encuentran en las terminales 10 y 9 las conectaremos a una de las salidas del CI 74138 que fueron invertidas por el CI 7404 (una salida por cada 74194), con esto obtendremos las funciones de HOLD y CARGA PARALELA, cuando sean requeridas.

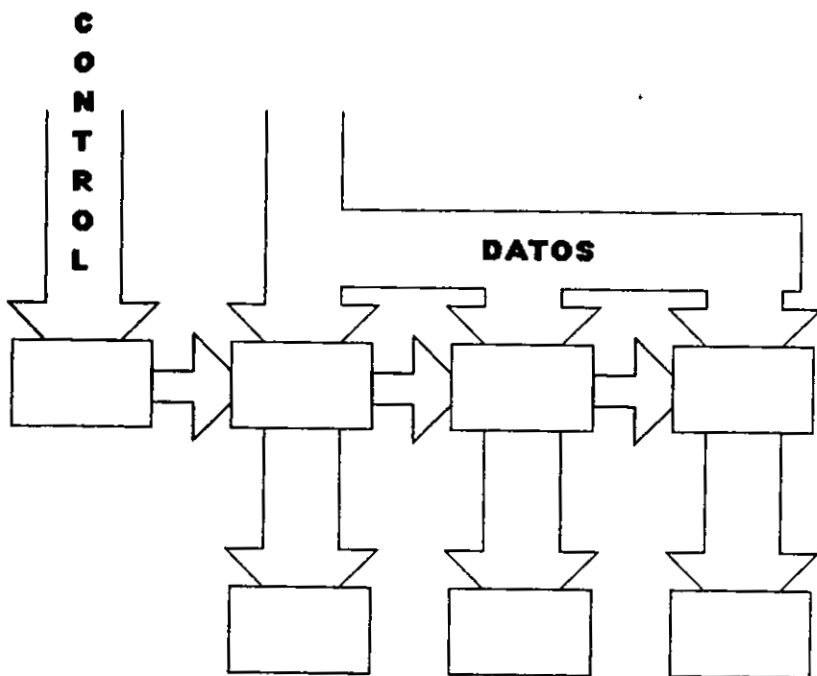


FIGURA 30: DIAGRAMA A BLOQUES PARA LA SALIDA DE BITS PROVENIENTES DEL PUERTO PARALELO

En el diagrama anterior (figura 30) muestra 2 bloques de entrada de bits: Un bloque de control y diversos bloques de entrada de datos; El bloque de control estará conformado por un circuito 74LS138.

De nuestra tabla de distribución de bits (tabla 8) utilizamos a los 3 bits de control obtenidos por medio de la interfaz del puerto paralelo

La distribución de los bits es la siguiente

1 BIT de bloqueo	3 bits de control	4 bits de información
X	X X X	X X X X

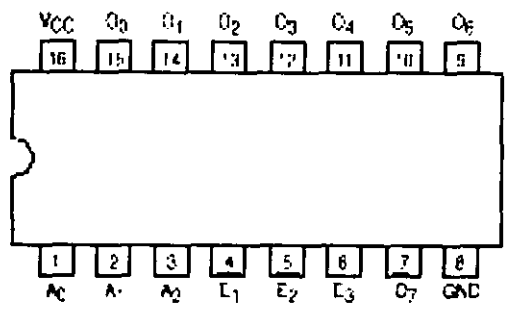
TABLA 8: DISTRIBUCIÓN DE LOS BITS DE DATOS PROVENIENTES DEL PUERTO PARALELO

En este arreglo el BIT menos significativo es el que se encuentra más a la derecha del bloque de bits y el BIT más significativo es el que se encuentra en el extremo izquierdo de la tabla.

Antes de ensamblar recuerde que no debe conectar ninguna fuente de voltaje, hasta haber terminado de realizar las conexiones de las terminales necesarias para que el CI funcione correctamente;

Para hacer las diferentes conexiones puede usar una protoboard¹, todas las conexiones se realizaran con cable para protoboard.

Inserte el chip en la protoboard con la cuña apuntando hacia el lado izquierdo. De acuerdo con el diagrama de asignación de pines:



- El pin 16 se conectará a los 5 Volts de CD
- El pin 18 se conecta hacia el polo negativo de la fuente de voltaje
- Los pines E₁, E₂ se conectarán a GND el pin E₃ se conecta a +5 Volts

FIGURA 31: CI 74138

¹ Protoboard: Tableta con pequeños huecos para colocar y hacer pruebas en los circuitos integrados.

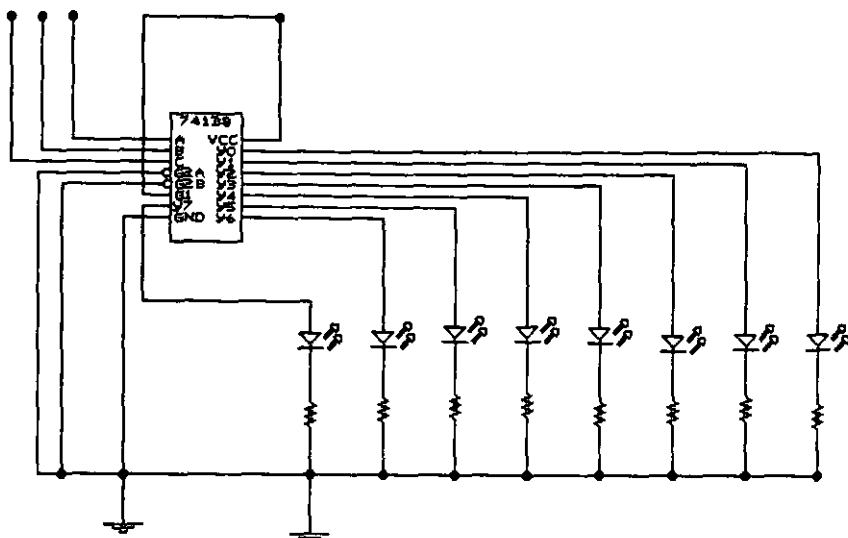


FIGURA 32: CONECCIONES PARA EL CI 74138

II.8.1 PRUEBAS

Para comprobar que se han efectuado correctamente bien las conexiones de todas las terminales de este CI, siga los siguientes pasos:

1. Conecte la batería de 5V (ó fuente de voltaje) como se ve en el diagrama de la figura 33.
2. Conecte como se muestra en el diagrama un diodo emisor de luz de cualquier color (**NO use diodos Infrarrojos**)
3. Conecte un Switch a las terminalcs A₀, A₁, A₃ como se observa en el diagrama
4. Conecte cada diodo a cada una de las terminales de salida junto con una resistencia de 330 ohms (simbología por color Naranja, Naranja, Café)
5. Con los Switches realice diferentes combinaciones de encendido y apagado.

Lo que se observara es que cuando cambie las combinaciones de los switches de apagado a encendido, uno y solo uno de los leds se encontrara de forma diferente a los demás.

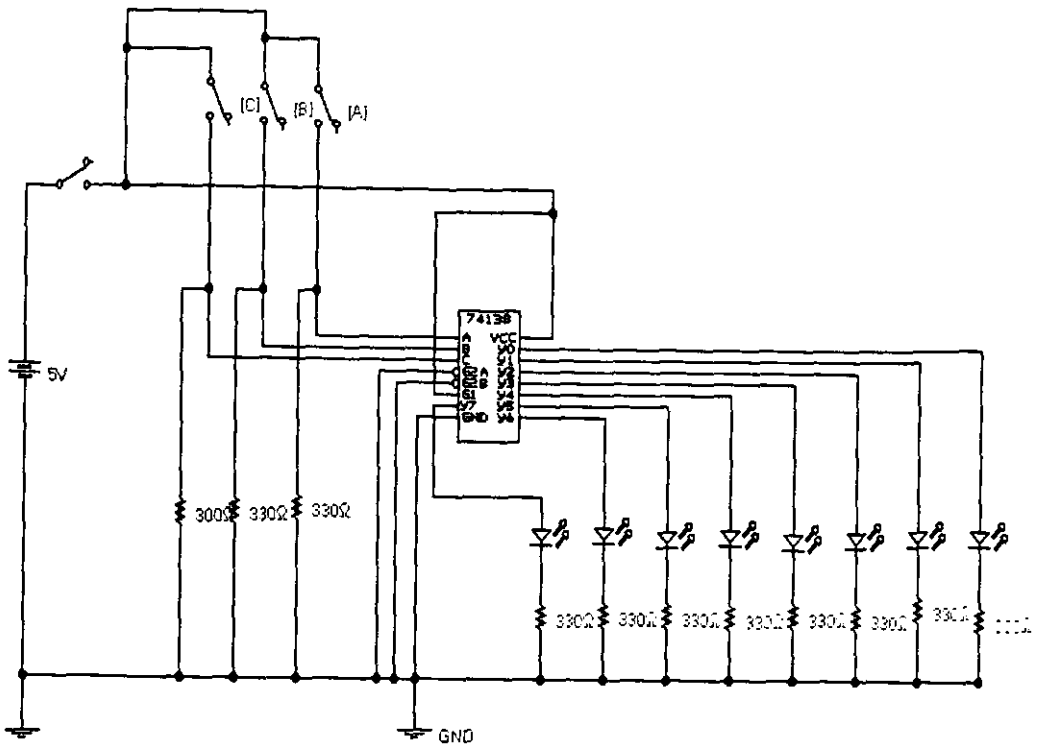


FIGURA 33: ARREGLO DE PUREBA PARA EL CI 74138

Lo que se observa al hacer las pruebas será la siguiente tabla

ENTRADAS						SALIDAS							
E ₁	E ₂	E ₃	A ₀	A ₁	A ₂	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇
H	H	H	H	L	L	H	L	H	H	H	H	L	L
L	L	H	H	L	L	L	L	L	L	L	L	L	L
L	L	H	L	H	L	L	L	L	L	L	L	L	L
L	L	H	H	H	L	L	L	L	L	L	L	L	L
L	L	H	L	L	H	L	L	L	L	L	L	L	L
L	L	H	H	L	H	L	L	L	L	L	L	L	L
L	L	H	L	H	H	L	L	L	L	L	L	L	L
L	L	H	H	H	H	L	L	L	L	L	L	L	L

H: NIVEL ALTO DE VOLTAJE
 L: NIVEL BAJO DE VOLTAJE
 X: NO IMPORTA

TABLA 9: CI 74138 (TABLA DE VERDAD)

II.9 DATOS

De acuerdo al diagrama a bloques, tenemos que la serie de 4 bits que provienen del puerto paralelo, los cuales hemos denominado como datos, como se muestra en la figura 34.

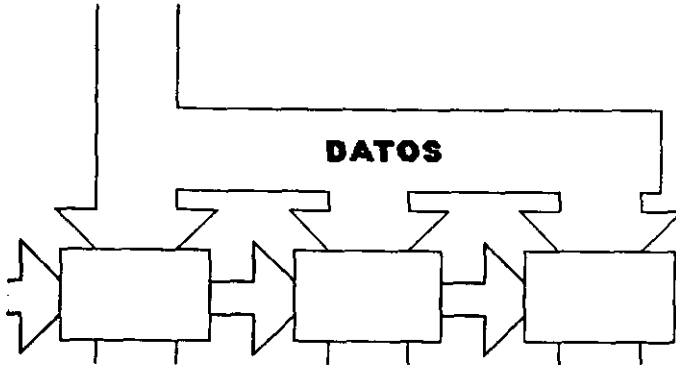
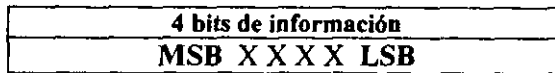


FIGURA 34: DIAGRAMA A BLOQUES PARA LA RECEPCIÓN DE DATOS

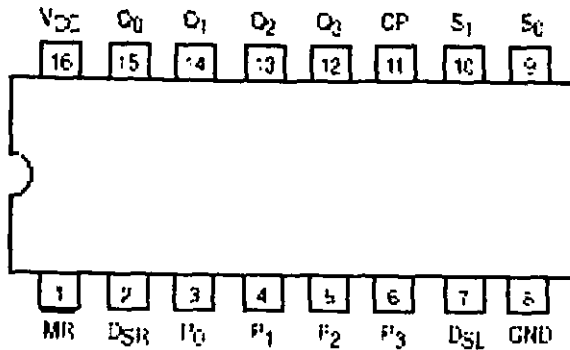
Estos datos son recibidos por el circuito mediante un CI TTL 74LS194, el cual es un registro bidireccional de 4 bits, controlado por los 3 bits de control que se han enviado hacia el CI 74138; A los 4 bits de datos que recibimos de nuestro puerto paralelo son colocados del BIT menos significativo hasta el BIT más significativo, hacia las entradas en paralelo del CI 74194, de acuerdo con el siguiente arreglo:



MSB: Bit más significativo

LSB: Bit menos significativo

Coloque el circuito sobre la protoboard, siguiendo las siguientes disposiciones:
Observando la colocación de pines de este circuito.



**FIGURA 35: CI 74194
ASIGNACION DE PINES**

1. Conecte el pin 16 (Vcc) hacia el positivo de la fuente de voltaje (recuerde que no debe encender la fuente de voltaje hasta conectar todos los pines necesarios para el funcionamiento de este circuito)
2. El pin 8 (GND) se conecta hacia el negativo de la fuente de voltaje
3. La terminal MR se conecta hacia Vcc
4. D_{SL} y D_{SR} se colocaran en GND
5. Las terminales P_0 , P_1 , P_2 , P_3 , se colocaran respectivamente con los bits del puesto paralelo, es decir, de izquierda a derecha.
6. S_1 , S_0 , se conectaran entre ellas y hacia una de las salidas del 7404
7. CP se conectara a un circuito generador de pulsos

II.10 Circuito generador de pulsos de reloj

Para este circuito necesita un CI 555 que es un multivibrador astable con el siguiente arreglo de resistencias para generar pulsos de reloj.

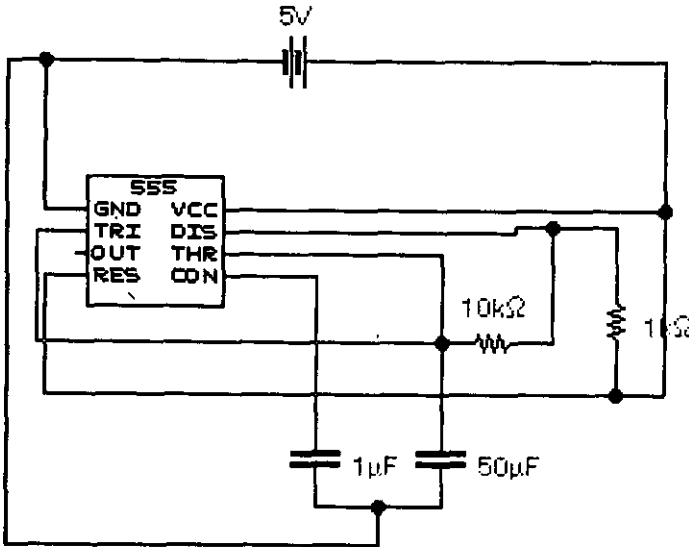


FIGURA 36: CIRCUITO GENERADOR DE PULSOS DE RELOJ (CI 555)

Con este arreglo obtendremos la siguiente salida:

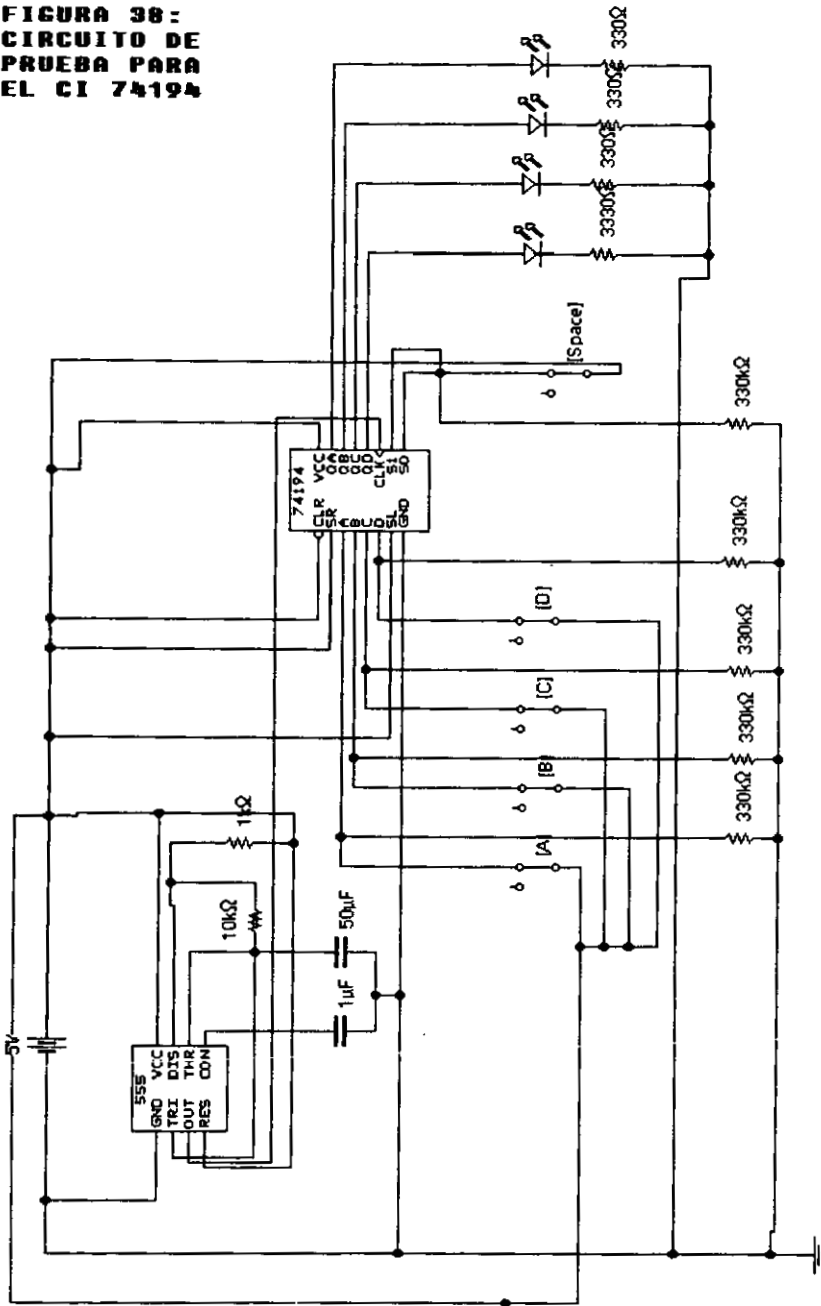


FIGURA 37: SEÑAL DE PULSOS DE RELOJ (CI 555)

II.10.1 Pruebas

Para realizar las pruebas al arreglo del circuito integrado 74194 debe conectar un diodo emisor de luz a cada una de las terminales de salida Q_n , junto con una resistencia de 330 ohms hacia el polo negativo (GND) de la fuente de voltaje. Esto lo podemos observar en el siguiente diagrama de la figura 38:

**FIGURA 38:
CIRCUITO DE
PRUEBA PARA
EL CI 74194**



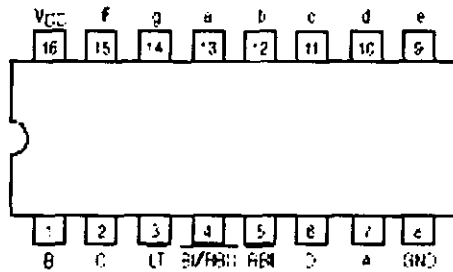
Este arreglo en el circuito nos producirá las siguientes salidas:

MODE	ENTRADAS						SALIDAS			
	MR	S ₁	S ₀	D _{SA}	D _{SL}	P _n	Q ₀	Q ₁	Q ₂	Q ₃
Reset	L	X	X	X	X	X	L	L	L	L
Hold	H	l	l	X	X	X	Q ₀	Q ₁	Q ₂	Q ₃
Parallel Load	H	h	h	X	X	P _n	P ₀	P ₁	P ₂	P ₃

TABLA 10: (CI 74194) TABLA DE VERDAD

Ya que se ha ensamblado los dos primeros circuitos, circuito de datos y circuito de control, desearíamos observar los datos enviados, principalmente al circuito de datos, para esto será necesario la utilización de un visualizador digital (display), y un circuito que convierta los datos para que estos puedan ser visualizados por medio del display.

El circuito que utilizaremos es el circuito TTL 74ls47 que de acuerdo a su forma de funcionamiento necesitamos un display de ánodo común; Su diagrama de distribución de pines se muestra en la figura 39.



**FIGURA 39: CI 7447
DIAGRAMA DE ASIGNACIÓN DE PINES**

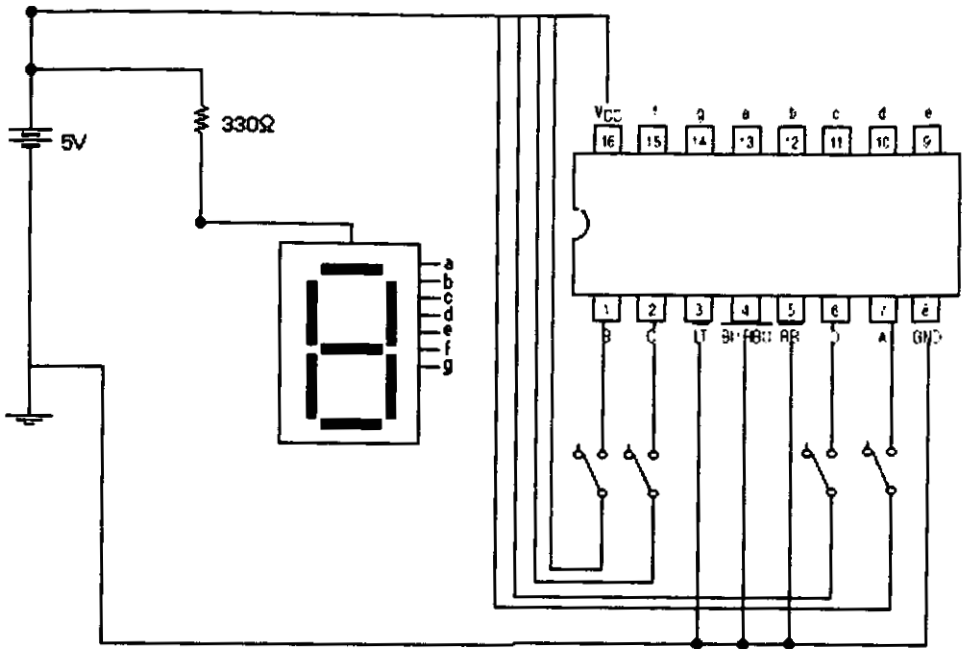
Para el funcionamiento en nuestro circuito los pines se colocaran de la siguiente manera:

1. Como primer paso se tiene que realizar las conexiones para polarizar al circuito, para esto se deberán de enviar a la terminal del pin 16 hacia Vcc (positivo de la fuente de alimentación), y la terminal del pin 8 hacia la terminal negativa de la fuente de voltaje.

2. El segundo paso será hacer que el circuito se encuentre habilitado para funcionar, para esto requerimos que: La terminal LT (pin 3), RBI (pin 4), BI/RBO (pin5), sean enviadas hacia nivel lógico alto, para lograr esto colocaremos estas terminales hacia Vcc.
3. Las terminales que recibirán los datos provenientes de la salida del CI 74194, son las 7,1,2,6, de esta manera observamos que el BIT menos significativo se encuentra en la terminal 7 y el más significativo se encuentra en la terminal 6, respectivamente.
4. Para visualizar los datos conectaremos las salidas de la letra a hasta la f hacia el display, que cuenta con terminales de entrada con el mismo nombre, por lo que cada terminal del CI 7447 se colocara con su respectiva letra de las entradas del display.
5. Para polarizar el display solo necesita conectar una de las terminales positivas junto con una resistencia de 220 o 330 ohms hacia la parte positiva de la fuente de voltaje que alimenta a nuestro circuito.

Para hacerle pruebas solo necesita hacer combinaciones en las terminales de entrada (7,1,2,6), para poder visualizar en el display que el circuito trabaja correctamente y los datos se mostraran en el visualizador digital.

El circuito queda de la siguiente manera:



Durante el proceso de prueba veremos que las salidas se observaran de la siguiente manera en el display

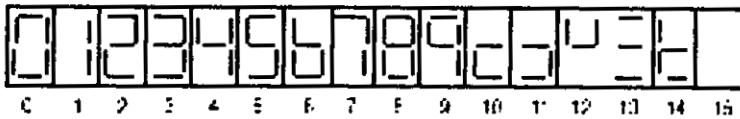


FIGURA 40: CIRCUITO DE PRUEBA PARA EL CI 7447

II.11 Circuito de entrada de datos

Ahora Bien teniendo el circuito de salida que puede controlar casi cualquier circuito que pueda ser acoplado a la salida de bits que obtenemos por medio de la distribución de bits del circuito de salida; Requerimos que se controle y verifique al circuito acoplado, así como los bits de datos, para esto es necesario realizar un circuito de entrada de datos.

El circuito de entrada tendrá el siguiente diagrama a bloques:

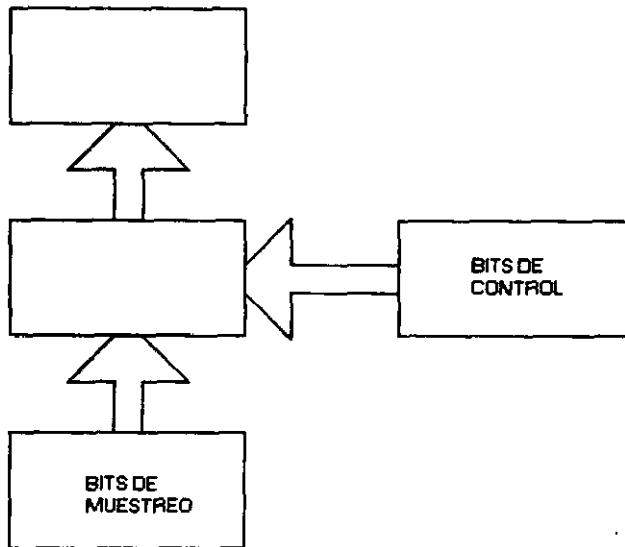


FIGURA 41: DIAGRAMA DE BLOQUES PARA LA RECEPCIÓN DE DATOS EN LA COMPUTADORA

Los bits de muestreo que son niveles de lógicos de voltaje alto o bajo, son enviados al ser seleccionados a través de un bloque de bits del circuito de salida, el cual nos habilita a un bloque del circuito de entrada para que estos bits sean registrados por la computadora como bits de muestreo.

Para que este circuito funcione de la manera como se expone en el diagrama a bloques necesita de los circuitos TTL como son buffer de 3 estados, multiplexores, registros, además de una lógica combinacional.

II.12 Características de los CI

Algunas de las características de los circuitos integrados que necesitaremos son las siguientes:

II.12.1 CI 74151

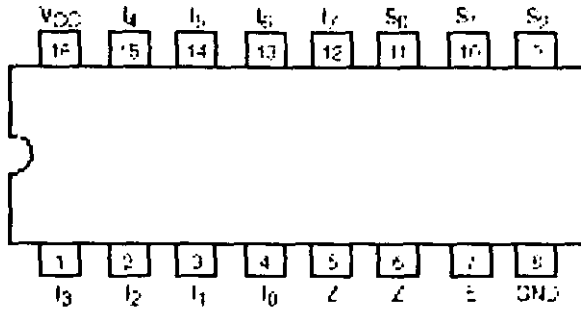


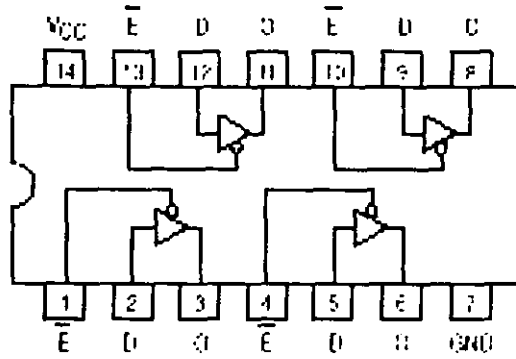
FIGURA 42: Diagrama de asignación de pines

E	S ₂	S ₁	S ₀	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	Z	Z
H	X	X	X	X	X	X	X	X	X	X	X	H	L
L	L	L	L	L	X	X	X	X	X	X	X	L	L
L	L	L	L	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	L	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	L	L
L	L	H	H	X	X	X	L	X	X	X	X	L	L
L	L	H	H	X	X	X	H	X	X	X	X	L	H
L	H	L	L	X	X	X	X	L	X	X	X	L	L
L	H	L	L	X	X	X	X	H	X	X	X	L	L
L	H	L	H	X	X	X	X	X	L	X	X	L	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	L	L
L	H	H	L	X	X	X	X	X	X	H	X	L	H
L	H	H	H	X	X	X	X	X	X	X	L	L	L
L	H	H	H	X	X	X	X	X	X	X	H	L	H

TABLA 11: (CI 74151) Tabla de verdad

El TTL 74LS151 (multiplexor de alta velocidad) proporciona, la habilidad de seleccionar un BIT de datos de un paquete de ocho bits de entrada.
 El 74LS151 puede usarse como un generador de una función universal, que genera cualquier función lógica de cuatro variables.

II.12.2 CI 74125



LS125A

**FIGURA 43: CI SN54/74LS125A
 DISPOSICIÓN DE PINES**

El circuito de la figura 43 es un buffer de 3 estados el cual trabaja con la siguiente tabla de verdad:

LS125A

INPUTS		OUTPUT
E	D	
I	I	I
L	H	H
H	X	(Z)

Donde Z = alta impedancia (off)

**TABLA 12: TABLA DE VERDAD
 CI 74LS125A**

De acuerdo con la tabla de verdad anterior (tabla 12) el circuito 74125 es utilizado para seleccionar uno y solo un BIT de entrada el cual podrá ser elegido mediante un bloque de nuestro circuito de salida de datos, con esto seleccionaremos el BIT de entrada que debe ser enviado a la salida del circuito para enviarlo hacia la interfaz del puerto paralelo.

El circuito para pruebas para el CI 74151 queda de la siguiente manera:

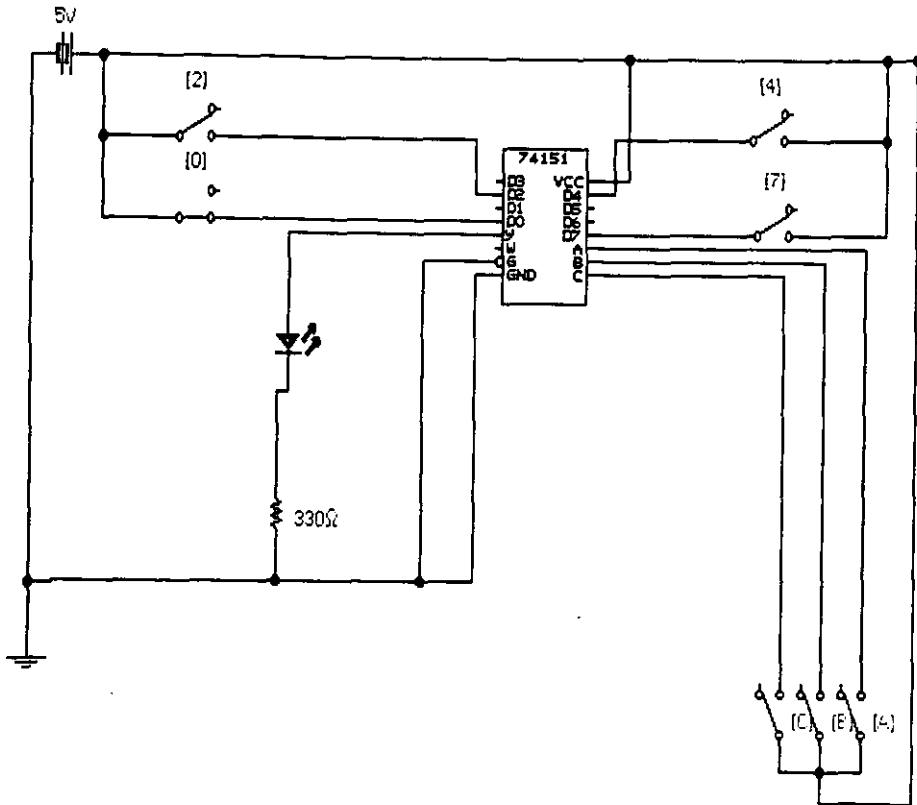


FIGURA 44: CIRCUITO DE PRUEBA PARA EL CI 74151

Las entradas A, B, C, serán conectadas a los datos transmitidos por un registro del circuito de salida, la terminal Y será enviada hacia la lógica combinacional de compuertas, pero antes se conectara hacia el buffer del 74125 que puede ser habilitado por un BIT de bloqueo de la trama de salida de la interfaz del Lpt1 (solo si es requerido de acuerdo con la tabla 8).

El circuito de salida se vera de la siguiente forma:

Figura 45-47

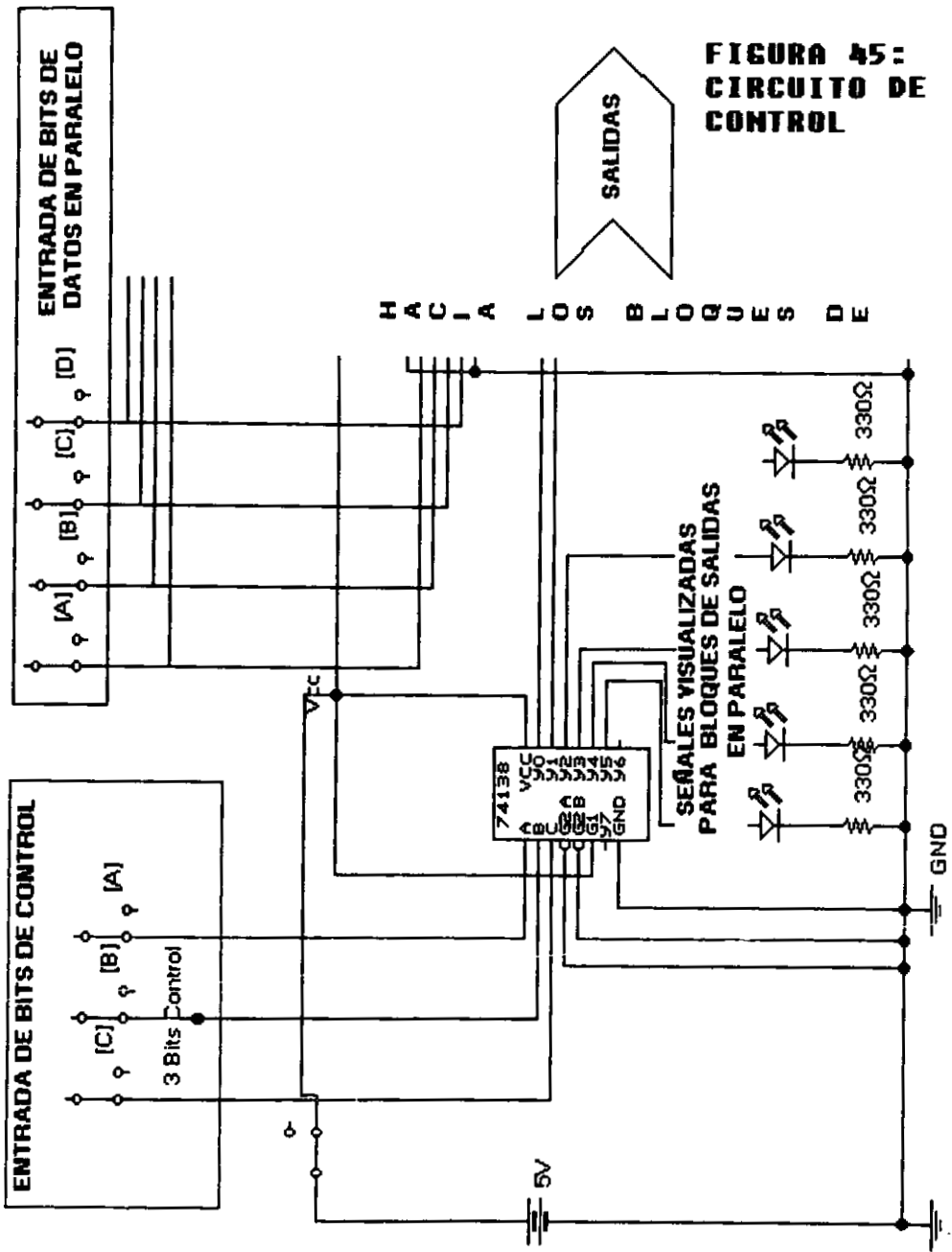
El arreglo final del circuito de entrada queda de la forma como se muestra en la figura 48:

Figura 48

II.13 EL CIRCUITO Y SU FUNCIONAMIENTO

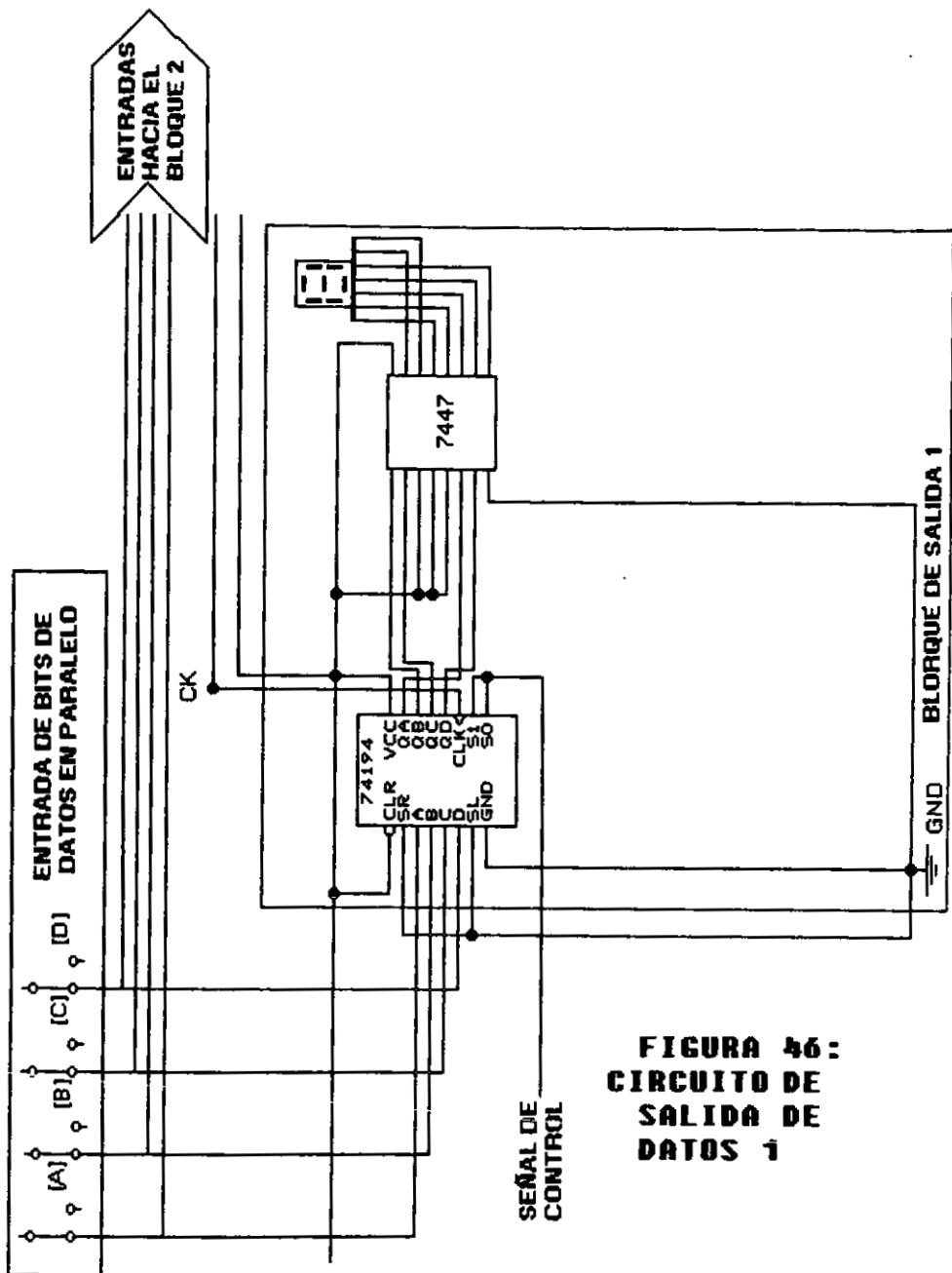
Observemos el funcionamiento del circuito completo:

1. Conecte a los dos circuitos como se ve en el diagrama de las figuras 45 a la 48 según el número de bloques que quiera.
2. Realizando una cuenta binaria en los bits de control, se observa que entre cada cuenta se realiza un cambio de los bits de datos en los registros que realizan la función de carga paralela, visualizando en el display el número que es transferido del registro hacia el display, con esto se comprueba la función de carga paralela de los registros del circuito de entrada.
3. Para seleccionar una entrada del circuito de entrada a la interfaz del puerto paralela, es colocando un número binario en las terminales ABC del CI 74151 para seleccionar una de las señales de la lógica combinatorial que sea requerida por la computadora.

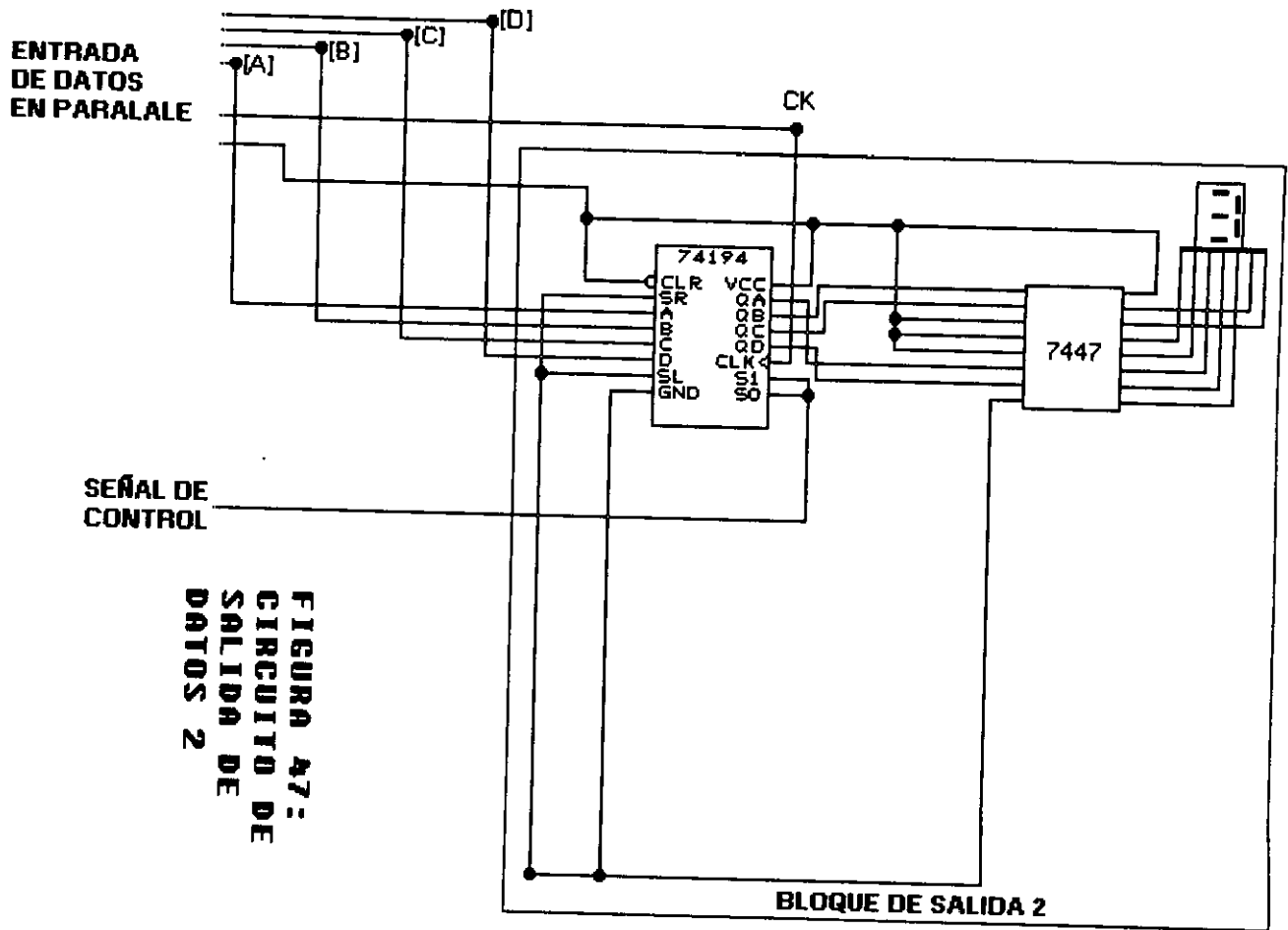


H A C I A L O S B L O Q U E S D E

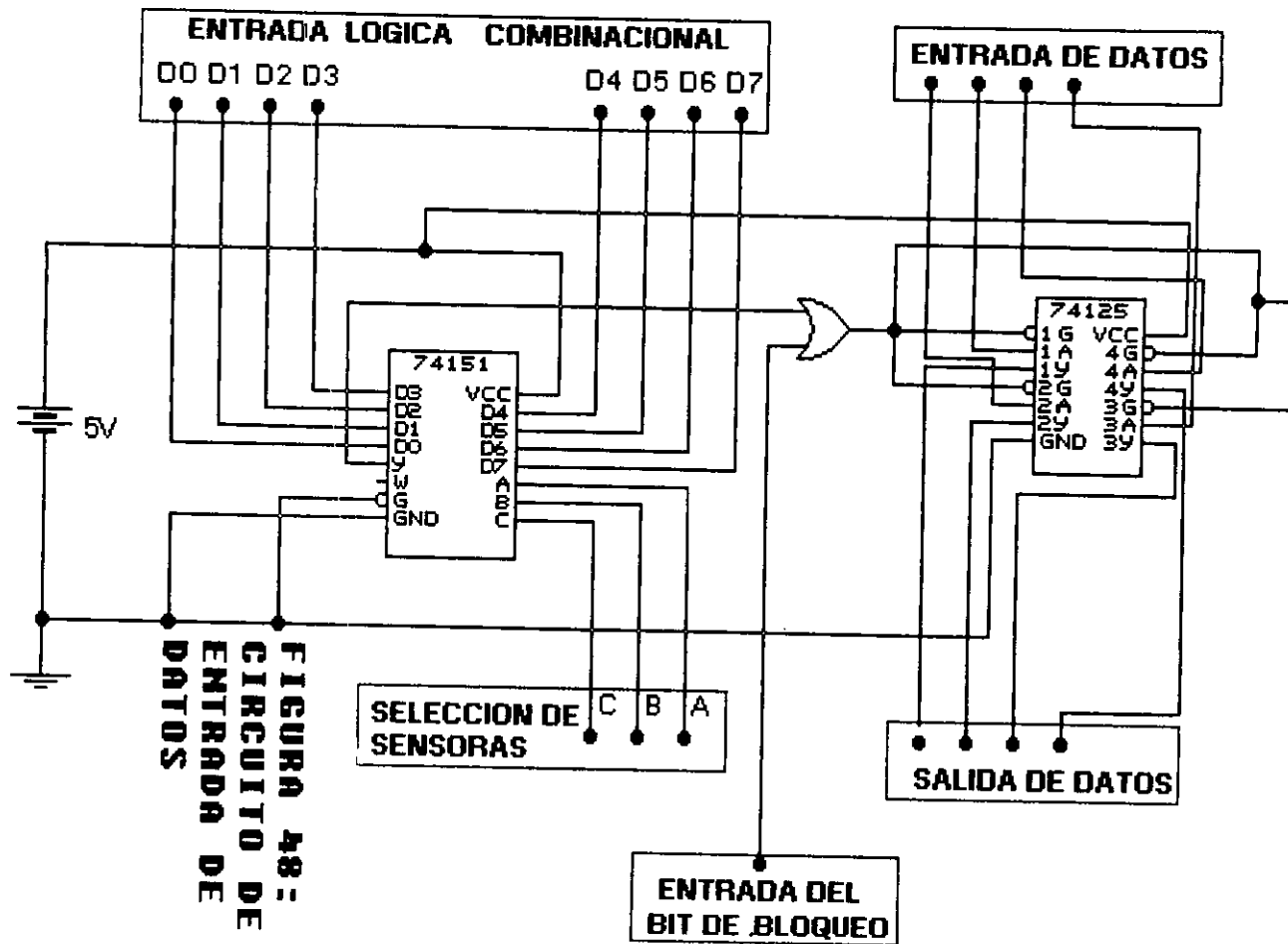
**FIGURA 45:
CIRCUITO DE
CONTROL**



**FIGURA 46:
CIRCUITO DE
SALIDA DE
DATOS 1**



**FIGURA 47:
CIRCUITO DE
SALIDA DE
DATOS 2**



**FIGURA 48:
CIRCUITO DE
ENTRADA DE
DATOS**

En todos los circuitos los led's, no son indispensables, solamente para comprobar salidas de los circuitos, así como los bloques de entrada a los registros, solo serán conectados cuantos y cuando sean requeridos, además los circuitos integrados 7447 y los display's solo son para verificar la salida de datos durante las cargas paralelas y pueden ser retirados para un menor consumo de corriente.

II.14 FUENTES DE ALIMENTACIÓN

Como todo circuito requiere de una fuente de voltaje y corriente para poder funcionar y realizar su trabajo, se menciona a continuación como es una fuente de voltaje.

Una fuente de alimentación o de poder convierte la entrada de ca de 1 línea de 60 Hz a una salida con voltaje de cc. Esta fuente de V^+ es necesaria para los amplificadores en equipo electrónico.

En la figura 49 se representa el esquema de bloques de los elementos que constituyen un equipo con circuitos integrados.

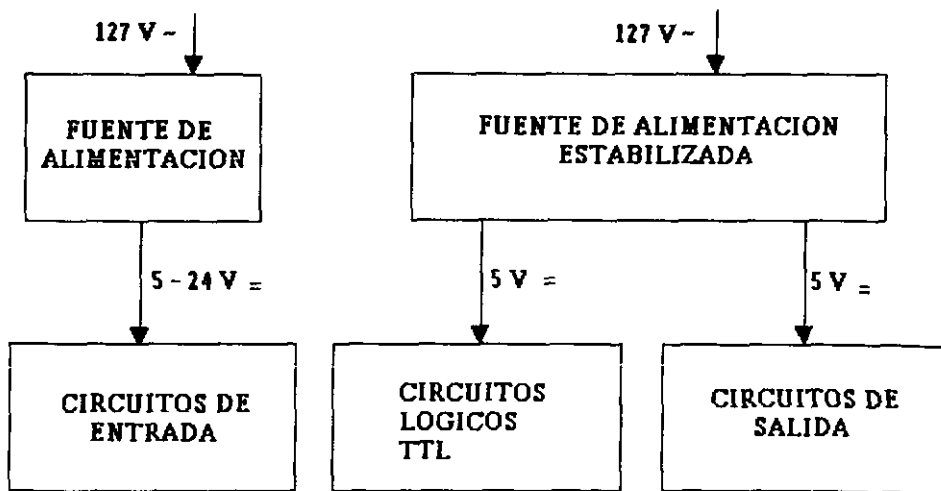


FIGURA 49: DIAGRAMA A BLOQUES PARA LA ALIMENTACIÓN DE CIRCUITOS

Circuito de entrada: Llamados también captadores, son los elementos que introducen las señales lógicas en el equipo. Pueden ser constituidos por interruptores, conmutadores, pulsadores, detectores, etc.

Circuito de salida: Los circuitos lógicos TTL operan con tensiones bajas (voltajes continuos de +5V) y corrientes muy pequeñas, lo que los hace inapropiados para ser aprovechados directamente en la gran mayoría de las aplicaciones, que operan con voltajes y corrientes más elevados.

Circuitos lógicos: En estos montajes se emplean exclusivamente en circuitos lógicos con una Lógica de Transistor-Transistor (familia TTL), que son los mas utilizados en las aplicaciones prácticas, dentro de estos circuitos se incluyen las compuertas lógicas, así como comparadores, codificadores, contadores, registros, etc.

Generalmente en un equipo con circuitos integrados se emplean según sea el caso, diversas tensiones continuas o alternas, por lo que debe disponer de fuentes de alimentación apropiadas.

El componente principal en la fuente de poder es el rectificador, el cual generalmente es un diodo de silicio. El diodo conduce solamente cuando está polarizado directamente. Una entrada de ca tiene semiciclos positivo y negativo y se convierte a una salida de cc con una polaridad constante.

II.14.1 Forma básica de una fuente de alimentación o de poder

Esencialmente solo se necesita un rectificador para convertir una entrada de ca en una salida de cc. Adicionalmente a esto se utilizan capacitores utilizados como filtros para suprimir y eliminar las variaciones pulsantes (rizos) de la salida de cc. Una fuente de cc tiene una sola polaridad pero puede cambiar de niveles de voltaje, además de esto puede utilizarse un regulador de voltaje de cc a la salida para evitar que la corriente cambie de valor, de no ser por esto el voltaje de cc tendería a disminuir conforme aumenta la demanda de corriente en la carga conectada a la salida de voltaje de cc.

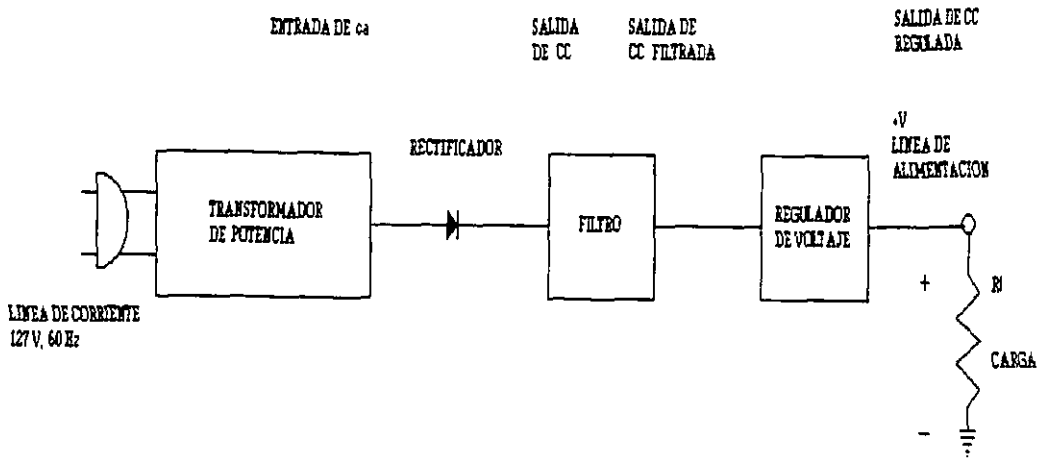


FIGURA 50: DIAGRAMA A BLOQUES DE UNA FUENTE DE VOLTAJE

II.14.2 Diodos rectificadores

Generalmente se utilizan diodos de silicio en la rectificación de las fuentes de alimentación. Los valores máximos de corriente van de 500 mA a 10 A. La ventaja que ofrecen es una muy pequeña caída de voltaje de aproximadamente 1 V.

La corriente que fluye dentro de un diodo es en un solo sentido, solamente cuando la entrada de ca produce un voltaje de polarización en directa.

El voltaje en directa para el diodo puede ser positivo en el ánodo o negativo en el cátodo. La entrada de ca puede suministrar este voltaje en uno de los semiciclos alternos, por lo que un solo diodo es un rectificado de media onda, por solo conducir un medio ciclo de la entrada de ca.

Pruebas del diodo rectificador

Un diodo semiconductor puede fácilmente ser probado con un ohmetro para ver si está abierto o en corto. Ya que un diodo cuenta con tan solo dos terminales se mide en el sentido de la polarización directa, y después en el sentido inverso, de acuerdo con la polaridad del ohmetro.

En la polarización inversa un diodo de silicio, la resistencia (R) debe ser muy pequeña, y en directa la resistencia (R) debe ser infinita ó muy grande. Si la resistencia es baja en ambas pruebas, el diodo está en corto; Si la resistencia es muy alta en ambos sentidos el diodo se encuentra abierto.



FIGURA 51: TERMINALES DE UN DIODO

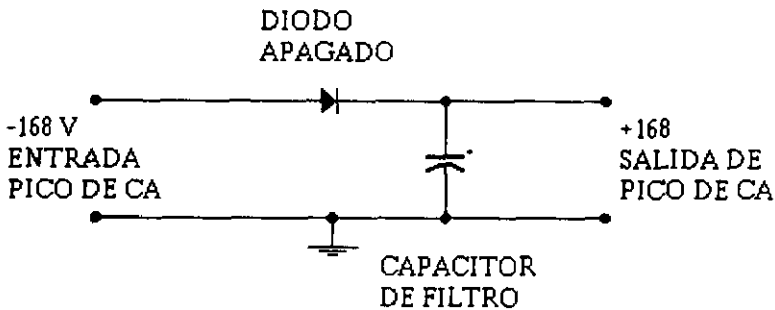


FIGURA 52: CIRCUITO DE VOLTAJE

11.4.3 Transformadores

Con el fin de obtener un valor deseado a la salida de cc, el voltaje de ca es necesario para el rectificador pueda diferir de los 127 V_{ac} de la línea de alimentación.

Utilizando un transformador de potencia, tenemos que debido a la inductancia mutua que existe entre L_p y L_s el voltaje de ca en el secundario aumentara o disminuirá su valor en proporción directa al numero de vueltas.

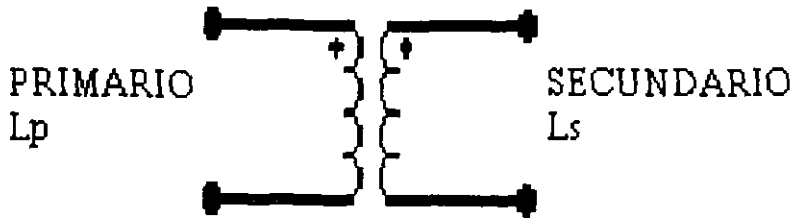


FIGURA 53: TRANSFORMADOR CON DOS DEVANADOS Y EL SECUNDARIO AISLADO

Para la comprobación de los devanados de un transformador puede utilizarse un ohmetro; Entre las terminales de los devanados del lado del primario de los transformadores de potencia que usualmente son de color negro, debe existir una resistencia de 5 a 50 Ω , mientras que la resistencia existente del secundario puede ser de 1 a 400 Ω , dependiendo si es elevador o reductor. Si existe un devanado abierto la lectura con el ohmetro será infinita.

II.14.4 Circuito rectificador de onda completa

El siguiente diagrama muestra un rectificador de onda completa, la salida del secundario es utilizado para el voltaje de cc de salida.

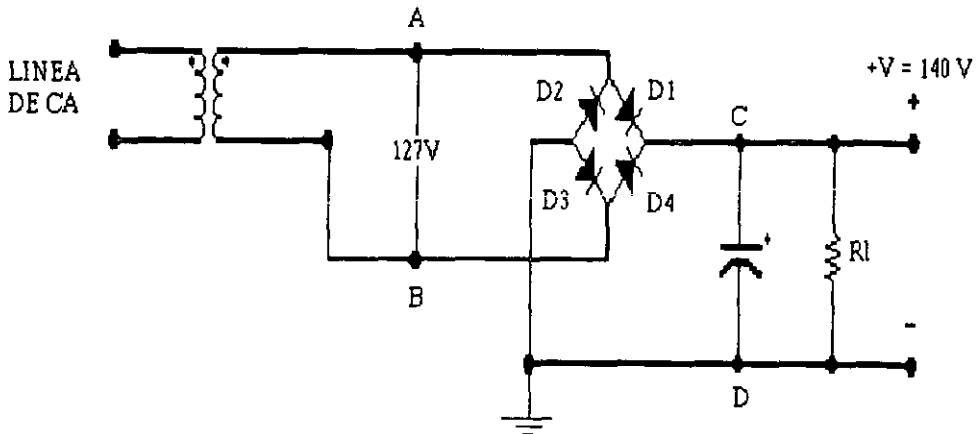


FIGURA 54: FUENTE DE VOLTAJE

Este circuito utiliza un transformador (T_1) de aislamiento, además cuenta con un puente rectificador con cuatro diodos, las cuatro terminales A, B, C, D, corresponden dos terminales para la entrada de ca, dos para la salida de cc.

II.14.5 Filtros para una fuente de alimentación

La salida obtenida de cc de un rectificador es unidireccional, sin la existencia de un filtro, pero aun existen variaciones en la amplitud, que es debida a una pulsación de ca superpuesta al nivel promedio de cc.

La función del filtro en la fuente es principalmente la de reducir la componente de rizado de ca en el voltaje de cc de salida, para así obtener un valor fijo de voltaje.

Para realizar el filtrado se realiza conectando capacitores en derivación, que tienen como función disminuir las componentes para la fuente de voltaje V^+ que deberá ser un valor fijo.

Los valores de los capacitores (C) usualmente son de 50 a 1000 μ F, se requiere que el tipo de capacitores sea electrolítico. El capacitor electrolítico debe conectarse con su polaridad correcta, su voltaje nominal debe ser ligeramente mayor que la salida de cc de la fuente de poder.

Al funcionar el filtro aplana el voltaje de salida porque el capacitor se opone a cualquier cambio en el voltaje. Además los capacitores deben ser conectados en paralelo con V^+ para no bloquear la salida de cc.

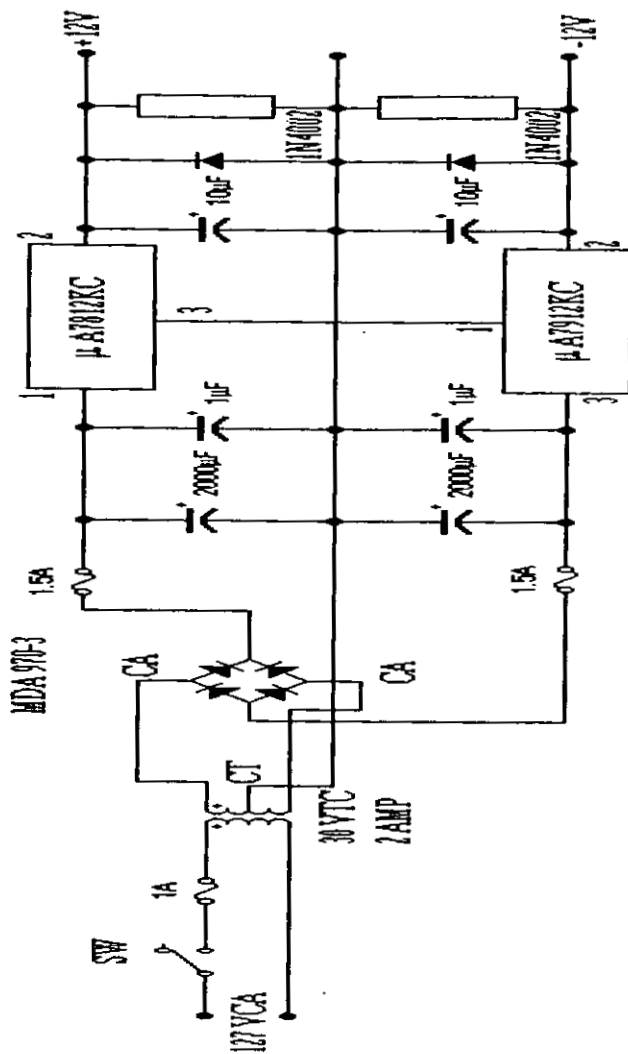
El porcentaje de rizado, que es una medida para identificar cuan efectivo es el filtro en reducir el rizado de ca en el voltaje de salida de cc, es mediante la siguiente formula.

$$\text{Factor de rizado \%} = (V_{\text{rizado}}/V_{\text{cc}}) \times 100$$

II.5 Fallas en las Fuentes de alimentación

Las fuentes de voltaje pueden presentar fallas si se operan con altos valores de voltaje y de corriente. Las fallas más comunes en este tipo de fuentes son:

1. *Rectificador defectuoso* Puesto que un solo diodo defectuoso puede originar que no exista salida de voltaje, aunque en la entrada de ca se encuentre normal.
2. *Filtro de entrada con capacitor abierto* Esto puede originar que la salida de +V sea baja, además que puede originar un excesivo zumbido, al usar capacitores electrolíticos pueden llegar a secarse con el tiempo haciendo que los capacitores se encuentre definitivamente abierto.
3. *Capacitor de filtro de entrada de en corto* Esto puede originar que no exista salida de voltaje con muy alta corriente en el rectificador y en el transformador de potencia, que puede originar que se quemen.
4. *Capacitor de filtro de salida abierto* Esta falla origina un zumbido sumamente alto sin afectar a la salida de voltaje.
5. *Capacitor de filtro de salida en corto* causara que exista una corriente excesiva en el circuito del rectificador incluyendo la resistencia o filtro, llegando a quemarse.
6. *Zumbido excesivo* Si obtenemos que la salida de voltaje sea cercana a la deseada, un zumbido producido es una falla de filtrado. En la mayor parte de los casos se corrige la dificultad remplazando a los capacitores que realizan la función de filtrado.



- Los radiadores de calor son de 200W para los reguladores de voltaje
- $R = 47K\Omega$

FIGURA 55: FUENTE DE VOLTAJE COMPLETA

CAPITULO III

PROGRAMA Y ESTRUCTURA DE CONTROL BASIC

INTRODUCCIÓN

La ingeniería de software se define como la disciplina tecnológica preocupada de la producción sistemática y mantenimiento de los productos de software que son desarrollados y modificados con el tiempo y dentro de un presupuesto definitivo. La ingeniería de software difiere de la programación tradicional porque se utilizan técnicas de ingeniería para especificar, diseñar, instrumentar, validar y mantener los productos dentro del tiempo y el presupuesto establecidos para el proyecto.

El termino *programador* se emplea para denominar al individuo preocupado por los detalles de la instrumentación, empaclado y modificación de los algoritmos y estructura de datos codificados en un lenguaje de programación particular.

III.1 Determinación del problema

III.1.1 Definición del problema: Todo elemento desarrollado por el hombre primero es una idea en su mente. Los sistemas computacionales, como otros productos de la tecnología, se desarrollan en respuesta a requerimientos detectados. La decisión de llevar a cabo el proyecto se basa, generalmente, en la explotación de todas las oportunidades para su uso.

Para definir el problema es necesario.

1. Desarrollar un enunciado definitivo del problema por resolver, incluir una descripción de la situación actual, restricciones del problema y de las metas que se logran, el enunciado del problema debe realizarse empleando tecnología del cliente.
2. Justificar una estrategia de solución computarizada para el problema.
3. Identificar las funciones por realizar, las restricciones, el subsistema de equipo electrónico, el subsistema del producto de programación y el del personal.
4. Determinar los objetivos y requisitos en el nivel del sistema, para el proceso de desarrollo y los productos finales.

5. Establecer criterios de alto nivel para la aceptación del sistema.

III.1.2 Estudio de la factibilidad

La factibilidad de cada estrategia de solución propuesta se puede establecer por el análisis de las restricciones de la solución. Estas establecen las fronteras del espacio de soluciones; el análisis de factibilidad determina si una estrategia propuesta es posible dentro de dichas fronteras. Una estrategia de solución es factible si las metas y requisitos del proyecto se pueden satisfacer dentro del tiempo, recursos y tecnología disponible.

Las técnicas para determinar la factibilidad de una estrategia de solución comprenden el estudio de casos, análisis del peor caso, simulaciones y construcción de prototipos.

III.1.3 Aprobación del proyecto

Algunas metas se aplican a todos los proyectos y productos. Por ejemplo, todo producto de programación debe ser útil, confiable, comprensible, y eficiente. Todo proceso de desarrollo debe producir productos finales a tiempo y dentro de los estimados de costo, y debe permitir que el personal del proyecto aprenda nuevas habilidades, otras metas, como entrega anticipada de subsistemas, y facilidad de uso para los no programadores, dependiendo de la situación particular.

III.2 PROGRAMA DE CONTROL

Ya ensamblado nuestro circuito de salida el cual se acoplará a nuestra computadora a través de la interfaz del puerto paralelo (DB-25), es conveniente realizar un programa que controle las salidas y entradas automáticamente.

En forma general las computadoras constan de un CPU, ALU, Unidad de salida, Unidad de entrada y Memoria; Para que una computadora pueda procesar la información que se le dé mediante la unidad de entrada, es necesario un lenguaje el cual contiene una serie de reglas y pasos a seguir, junto con instrucciones que el usuario define para que la computadora pueda realizar un proceso interno.

El lenguaje de programación que utilizaremos será el lenguaje Basic que contiene las instrucciones necesarias para poder dar salida a los bits que deseemos enviar hacia nuestro puerto paralelo.

El lenguaje Basic requiere que la información que se le proporcione este dada en forma de constantes, variables o instrucciones.

III.2.1 ¿Para qué es necesario el programa?

Los circuitos de capítulos anteriores, están expuestos de tal manera que cumplen su función en combinación con el ordenador. El circuito se encarga de las tareas electrónica, garantizadas por la correcta combinación de los componentes utilizados y del cableado correspondiente. El programa es el responsable del comportamiento del ordenador, este trabaja de la forma que sea necesaria con relación al funcionamiento del circuito en cuestión. Estas tareas de control entre el comportamiento del ordenador y las tareas que realiza el programa son distintas en cada circuito.

Una de las tareas al programar es poder influir en el comportamiento de dicho circuito, a través del programa que enviara información al circuito por medio del puerto paralelo, siendo una tarea importante la de programar al puerto paralelo.

Otra tarea consiste en preparar la información que debe ser enviada al circuito, de tal forma que se pueda transmitir por los diferentes bloques y el circuito pueda procesarla de manera correcta. Para esto deben realizarse cálculos, conversiones y otros procesos dentro del programa.

III.3 El lenguaje de programación Basic

El programa para el ordenador se crea con la ayuda de los lenguajes de programación. Para realizar el programa es necesario crear archivos de código fuente. Estos archivos contienen la información y los comandos que en último término debe ejecutar el programa.

Debemos de tener presente que para las estructuras Basic en programación necesitamos de los diagramas de flujo, así como las sentencias de control que estos involucran. **Sentencia de control** también denominadas como estructuras de control, permiten tomar decisiones y realizar un proceso n número de veces.

Dentro de las estructuras de control contamos con las siguientes:

1. **If... Then**
2. **If ... Then ... Else**
3. **For ... Next**
4. **Do ... Loop**
5. **Goto**

La primera estructura permite ejecutar condicionalmente una o más sentencias (instrucciones) y puede escribirse de la siguiente manera:

If condición **Then** *sentencia(s)* **Else** *sentencia(s)*

If condición **Then**

sentencia(s)

Else *sentencia(s)*

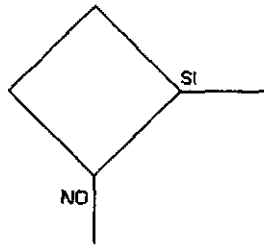


FIGURA 56: BLOQUE IF

La sentencia **For** da lugar a un lazo o bucle que permite ejecutar un conjunto de sentencias un cierto número de veces, a esto se le conoce como circuito contador y su proposición **For Next**:

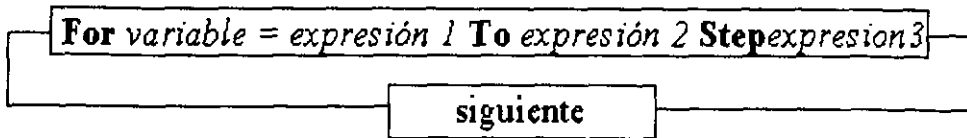
For *variable = expresión 1 To expresión 2 Step expresión 3*

Sentencias

Exit For

Sentencias

Next *variable, variable*



Borrar la pantalla:

CLS {{0 | 1 | 2}}

CLS Borra la ventana gráfica o la de texto. Si se ha establecido una ventana gráfica (usando **VIEW**), se borrará sólo la ventana gráfica. De lo contrario, se borrará la ventana de texto o toda la pantalla.

- CLS 0 Borra la pantalla, quitando todo el texto y los gráficos.
- CLS 1 Borra la ventana de gráficos, o la pantalla completa.
- CLS 2 Borra la ventana de texto.

Repetir un bloque de instrucciones el número de veces especificado.

```
FOR contador = inicio TO fin [STEP incremento]
  [bloque instrucciones]
NEXT [contador [,contador]...]
```

contador Una variable numérica utilizada como contador de bucle.

inicio y fin Los valores inicial y final del contador.

incremento El incremento con el que se cambia el contador cada vez que se ejecute el bucle.

Ejemplo:

```
FOR i% = 1 TO 15
  PRINT i%
NEXT i%
FOR i% = 7 to -6 STEP -3
  PRINT i%
NEXT i%
```

Realizar una bifurcación a una línea especificada.

GOTO línea

línea La etiqueta o el número de la siguiente línea que debe ser ejecutada.

DO...LOOP, SELECT CASE, IF...THEN...ELSE, SUB y FUNCTION ofrecen mejores formas de controlar el flujo de los programas.

GOTO también se utiliza como palabra clave en la instrucción ON ERROR.

HEX\$ devuelve una representación hexadecimal de cadena de un número.

OCT\$ devuelve una representación octal de cadena de un número.

HEX\$(expresión-numérica&)

OCT\$(expresión-numérica&)

expresión-numérica& Cualquier expresión numérica. La expresión se redondeada a un entero o entero largo antes de ser evaluada.

Ejemplo:

```
INPUT x
a$ = HEX$ (x)
b$ = OCT$ (x)
PRINT x; "decimal es "; a$; " hexadecimal y "; b$; " en octal."
```

Ejecuta una instrucción o bloque de instrucciones según las condiciones especificadas.

```
IF condición THEN
  [bloque instrucciones-1]
[ELSEIF condición THEN
  [bloque instrucciones-2]]...
[ELSE
  [bloque instrucciones-n]]
END IF
```

IF condición THEN instrucciones [ELSE instrucciones]

condición1 Cualquier expresión que puede ser evaluada como verdadera (no cero) o falsa (cero).

bloque instrucciones-1 Una o más instrucciones en una o más líneas.

bloque instrucciones-n

instrucciones Una o más instrucciones, separadas con el signo de dos puntos.

Ejemplo:

```
INPUT "¿1 € 2? ", i%
IF i% = 1 OR i% = 2 THEN
  PRINT "OK"
ELSE
  PRINT "Valor no válido"
END IF
```

Asigna el valor de una expresión a una variable.

[LET] variable = expresión

Variable Cualquier variable. Los nombres de variables pueden tener hasta 40 caracteres y deben comenzar con una letra. Los caracteres válidos son A-Z, 0-9 y el punto (.). No se pueden usar letras acentuadas.

expresión Cualquier expresión que produzca un valor al que puede ser asignada.

No es recomendable usar la palabra clave optativa LET. La instrucción de asignación variable = expresión realiza la misma acción con o sin LET.

LOCATE mueve el cursor en la pantalla a la posición especificada.

CSRLIN devuelve la posición actual de la fila donde se encuentra el cursor.

POS devuelve la posición actual de la columna donde se encuentre el cursor.

LOCATE [fila%] [,columna%] [,cursor%] [,inicio% [,fin%]]

CSRLIN

POS(expresión)

fila% y columna% El número de la fila y columna a la que se debe mover el cursor.

cursor% Especifica si el cursor esta visible:
0 = invisible, 1 = visible

inicio% y fin% Expresiones de enteros entre 0 y 31 que especifican la primera y última línea de exploración del cursor. Podrá cambiar el tamaño del cursor modificando las líneas de exploración.

expresión Cualquier expresión.

Ejemplo:

```
CLS
LOCATE 5, 5
MiLin% = CSRLIN
MiCol% = POS(0)
PRINT "posición 1 (Presione cualquier tecla)"
```

```
DO
LOOP WHILE INKEY$ = ""
LOCATE (MiLin% + 2), (MiCol% + 2)
PRINT "posición 2"
```

INP genera un byte leído del puerto (hardware) de Entrada y Salida (E/S).
OUT envía un byte a un puerto (hardware) de E/S.

INP(puerto%)
OUT puerto%, datos%

puerto% Un número entre 0 y 65,535 que identifica el puerto.
datos% Una expresión numérica entre 0 y 255 que puede ser enviada al puerto.

Ejemplo:

```
x% = INP(&H3FC)      'Leer COM1 Modem Control Register.
OUT &H3FC, (x% XOR 1) 'Cambiar BIT de Dato de la Terminal.
```

TIMER activa, desactiva o suspende la intercepción de eventos de **TIMER**. Si esta activada la intercepción de eventos, (**ON TIMER**) ir a una subrutina cada vez que transcurra el número de segundos especificado.

```
TIMER ON
TIMER OFF
TIMER STOP
ON TIMER(n%) GOSUB línea
```

TIMER ON Activa la intercepción de eventos de TIMER.
TIMER OFF Desactiva la intercepción de eventos de TIMER.
TIMER STOP Suspende la intercepción de eventos de TIMER.

Los eventos pueden ser procesados una vez que se active la intercepción de eventos a través de TIMER ON.

n% El número de segundos que deben transcurrir antes de que ON TIMER se bifurque a una subrutina de intercepción de eventos; Un valor entre 1 y 86,400 (24 horas).

línea La etiqueta o el número de la primera línea de la subrutina para intercepción de eventos.

Ejemplo:

```
ON TIMER(1) GOSUB Horaactualizada
TIMER ON
CLS
PRINT "Hora: "; TIMES$
HoraInicio = TIMER
WHILE TiempoTranscurrido < 10
    TiempoTranscurrido = TIMER - HoraInicio
WEND
END
```

```
HoraActualizada:
LOCATE 1, 7: PRINT TIMES$
RETURN
```

III.4 PRUEBAS

Una vez que se tienen acoplado a todo el circuito, con la interfaz del puerto paralelo hacia la computadora, tendremos que realizar la programación adecuada para poder manejar al circuito conectado al puerto y a los bits que deben ser enviados correctamente. Para verificar su funcionamiento enviaremos datos para verificar el funcionamiento correcto del circuito con los bits que sean entregados a la entrada de los registros.

NUMERO DECIMAL	NUMERO HEXADECIMAL
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
17	11
18	12
19	13
20	14
21	15
22	16
23	17
24	18
25	19
26	1A
27	1B
28	1C
29	1D
30	1E
.....

TABLA 13: CUENTA DECIMAL / HEXADECIMAL

Los datos que se leen por medio del teclado hacia el programa son en forma decimal mas sin embargo los datos que el paralelo envía son en forma hexadecimal para ello es necesario tener en cuenta la tabla anterior para identificar que datos son enviados al puerto paralelo.

Los programas que a continuación se muestran están escritos en lenguaje Basic.

III.5 PROGRAMA

Si el puerto paralelo LPT1 de la computadora esta identificado con el número hexadecimal 378.

```
10 LOCATE 10, 10: PRINT "n"  
   INPUT n           ' Petición del dato desde el teclado  
   OUT &H378, n      ' Instrucción de salida de datos hacia LPT1  
   x% = INP(&H378)   ' Instrucción para verificar desde la computadora  
                     ' si el dato ha sido enviado correctamente  
   LOCATE 20, 20: PRINT ; x% ' Muestra el dato enviado ha LPT1  
   GOTO 10           ' Salto hacia la etiqueta
```

El anterior bloque de instrucciones espera a que el usuario introduzca un dato decimal a la computadora para enviarlo hacia el puerto paralelo en forma hexadecimal.

El siguiente bloque de instrucciones permite enviar un dato al puerto paralelo después de transcurrido de 10 segundos.

```
ON TIMER(1) GOSUB HoraActualizada  
TIMER ON  
CLS  
PRINT "Hora: "; TIME$  
OUT &H378, 0  
OUT &H378, 0  
OUT &H378, 0  
HoraInicio = TIMER  
WHILE TiempoTranscurrido < 10  
   TiempoTranscurrido = TIMER - HoraInicio  
WEND  
END
```

HoraActualizada:

```
LOCATE 1, 7: PRINT TIMES$  
OUT &H378, 55  
RETURN
```

Bloque de instrucciones que durante 10 segundos espera a que el usuario envíe un dato al puerto paralelo designado por el usuario.

10

```
ON TIMER(1) GOSUB HoraActualizada  
TIMER ON  
CLS  
PRINT "Hora: "; TIMES$  
Horalnicio = TIMER  
WHILE TiempoTranscurrido < 10  
    TiempoTranscurrido = TIMER - Horalnicio  
WEND  
CLS 0  
LOCATE 1, 20: PRINT "DESEA CONTINUAR OPRIMA 1"  
INPUT E  
IF E = 1 THEN GOTO 10 ELSE  
END
```

HoraActualizada:

```
LOCATE 1, 7: PRINT TIMES$  
LOCATE 1, 15: PRINT "DAME N"  
INPUT N  
OUT &H378, N  
RETURN
```

Bloque de instrucciones que envía un dato a diferentes registros

```
CLS 0  
PRINT "INICIO"  
PRINT "DAME EL NUMERO DE REGISTROS"  
INPUT p  
E = 0  
N = 07
```

```

WHILE E < p
LET E = E + 1
PRINT ; E; N
HoraInicio = TIMER
  WHILE TiempoTranscurrido < 2
    TiempoTranscurrido = TIMER - HoraInicio
    OUT &H378, N
  WEND
LET N = N + 16
LET TiempoTranscurrido = 0
WEND
PRINT "FIN"
END

```

Bloque de instrucciones que envía los datos que asigne el usuario a diferentes registros de salida conectados al puerto paralelo

```

CLS 0
PRINT "DAME EL NUMERO DE REGISTROS"
INPUT N
LET E = 0
LET P = 0
WHILE E < N
  LOCATE 10, 10: PRINT "DAME EL DATO"
  INPUT D
  LET Z = P + D
  OUT &H378, Z
  LET E = E + 1
  LET P = P + 16
WEND
END

```

Bloque de instrucciones que envía un dato a diferentes registros, los mantiene un periodo de 5 segundos y posteriormente los borra (manda a los registros el dato 0)

```
CLS 0
PRINT "INICIO"
PRINT "DAME EL NUMERO DE REGISTROS"
INPUT p
E = 0
N = 17
WHILE E < p
LET E = E + 1
PRINT ; E; N
HoraInicio = TIMER
  WHILE TiempoTranscurrido < 5
    TiempoTranscurrido = TIMER - HoraInicio
    OUT &H378, N + 3
  WEND
HoraInicio = TIMER
  WHILE TiempoTranscurrido < 2
    TiempoTranscurrido = TIMER - HoraInicio
    OUT &H378, N
  WEND
LET N = N + 16
LET TiempoTranscurrido = 0
WEND
PRINT "FIN"
END
```

Bloque de instrucciones que permite al usuario enviar diferentes datos a diferentes registros, los datos son introducidos en secuencia para que el programa los envíe uno a uno a cada registro conectado al Puerto paralelo

```
CLS 0
PRINT "INICIO"
PRINT "DAME EL NUMERO DE REGISTROS"
INPUT p
E = 0
N = 0
```



```

FOR i% = 1 TO p
  PRINT "DAME EL DATO", i%
  INPUT D(i)
NEXT i%
WHILE E < p
  LET E = E + 1
  PRINT ; E; N
  HoraInicio = TIMER
  WHILE TiempoTranscurrido < 5
    TiempoTranscurrido = TIMER - HoraInicio
    OUT &H378, (N + D(i))
  WEND
  LET N = N + 16
  LET TiempoTranscurrido = 0
WEND
PRINT "FIN"
END

```

Bloque de instrucciones que envía datos que introduce el usuario a diferentes registros conectados al Puerto paralelo, manteniéndolos por dos 5 segundos y posteriormente los borra transcurrido ese tiempo

```

CLS 0
PRINT "INICIO"
PRINT "DAME EL NUMERO DE REGISTROS"
INPUT p
E = 0
N = 0
J = 0
FOR i% = 1 TO p
  PRINT "DAME EL DATO", i%
  LET J = J + 1
  INPUT D(J)
NEXT i%
LET J = 0

```

```

WHILE E < p
LET E = E + 1
LET J = J + 1
Horalnicio = TIMER
  WHILE TiempoTranscurrido < 5
    TiempoTranscurrido = TIMER - Horalnicio
    LET Z = (N + D(J))
    OUT &H378, Z
  WEND
LET TiempoTranscurrido = 0
WHILE TiempoTranscurrido < 2
  TiempoTranscurrido = TIMER - Horalnicio
  LET X = Z - D(J)
  OUT &H378, X
WEND
LET N = N + 16
LET TiempoTranscurrido = 0
WEND
PRINT "FIN"
END

```

Bloque de instrucciones que permite al usuario continuar dentro del programa o terminar con la ejecución del programa

```

CLS 0
10 --
--
--
--
--
--
--
--
--
PRITN "DESERA CONTINUAR OPRIMA EL 1"
INPUT d
IF d = 1 THEN GOTO 10
END

```

Con la base de este código fuente en Lenguaje Basic, se puede realizar diferentes programas de acuerdo a la unión de un bloque de instrucciones con otro bloque de instrucciones diferentes.

Ahora bien si se quiere programar en visual Basic, existe la problemática de que en este tipo de paquete de programación no existe explícitamente una estructura e instrucción de salida y entrada de datos para el puerto paralelo (LPT1, LPT2, etc), para poder realizar la programación gráfica de esta clase de programas (figura 57)

probador del puerto paralelo

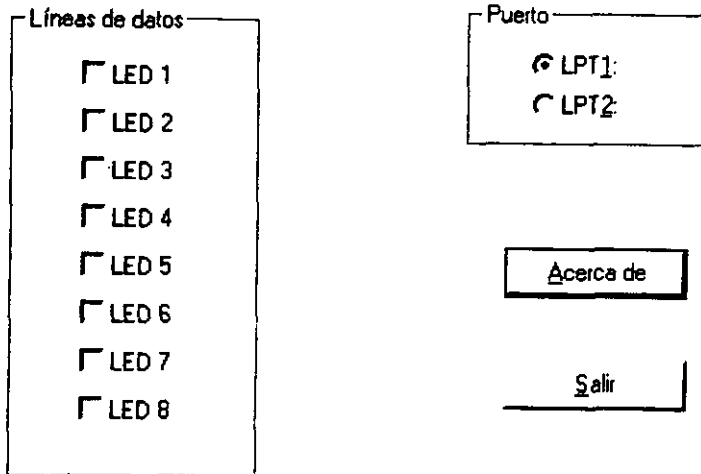


FIGURA 57: PROGRAMA EJECUTABLE DE VISUAL BASIC

se requiere de los pasos de llamado de biblioteca, más la inserción de código fuente que llame a la biblioteca de Windows de comunicación a puertos, para una mayor referencia sobre la aplicación de esta clase de programas puede consultar el libro:

Circuitos Electrónicos con la PC
Editorial: Alfa-Marcombo
Autor: Brend Zoller.

En este libro muestra diversos ejemplos de salida directa al puerto paralelo, además de incluir el código fuente de dichos ejemplos.

III.6 Ejemplo

Con lo anterior tiene el código fuente base para lograr obtener salidas hacia el circuito y distribuirlo a través de los registros conectados en paralelo, lo siguiente es direccionar a los bits de datos y mantener los datos dentro de los registros, con lo cual se logra liberar al programa de las funciones de salida de datos para el control constante, es decir, hacer que los registros mantengan la información que llega a ellos al ser direccionado por la parte de control del circuito, con lo cual obtenemos una distribución de los datos que llegan al circuito y podemos hacer que el programa realice otras funciones.

Ejemplo:

Ahora observemos un ejemplo de cómo podemos utilizar los bits de datos que son retenidos por medio de los registros:

III.6.1 Programa para controlar un motor:

De acuerdo al siguiente diagrama, el motor a controlar tiene como característica principal, la de ser controlado por medio de un convertidor D/A, el cual aumenta o disminuye su velocidad de acuerdo con el número binario que es entregado a la entrada del circuito, producto de la carga paralela realizada en un registro del circuito de salida.

Como la función principal del programa es entregar bits de datos al puerto paralelos (LPT1) y la función del circuito es mantenerlos, podemos enviar datos hacia el primer registro para operar un motor como se muestra en el diagrama de la figura 58; por lo tanto las instrucciones que nos sirven para este fin son:

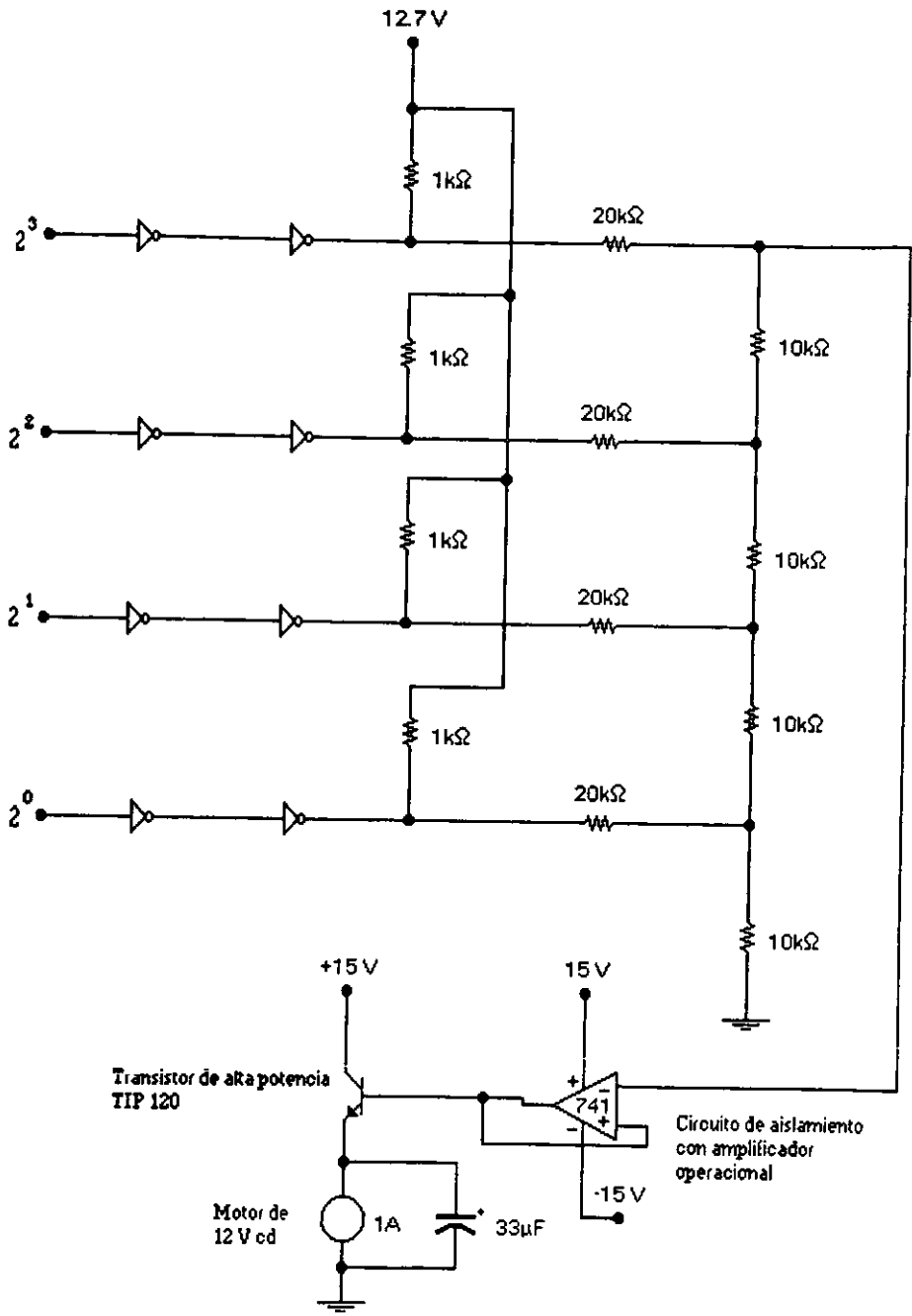


FIGURA 58: CIRCUITO PARA CONTROLAR UN MOTOR DE CD

Código fuente

```
CLS 0
'El número de registros es uno
LOCATE 5, 5: PRINT "dame el número de datos para el registro 2"
LOCATE 6, 5: INPUT n
K = 1
'Petición de los datos que se enviaran al puerto paralelo
PRINT "Datos para el registro 2"
FOR i% = 1 TO n
    LOCATE 10, 10: PRINT "Dame el dato:"; i%
    LOCATE 11, 10: INPUT d(K)
    LET K = K + 1
NEXT i%
K = 1
PRINT ; n
'Salida de datos al puerto paralelo

FOR j% = 1 TO n
    PRINT " el dato"; d(K)
    OUT &H378, d(K)
    LET K = K + 1
NEXT j%

'Instrucción que indica que hasta que el usuario oprima la tecla ESC para que termine
la ejecución del programa
10 PRINT "Presione Esc para salir..."
DO

'Petición del BIT de estado
20 x% = INP(&H379) 'Leer COM1
IF x% <> 0 THEN
    GOTO 10
ELSE
    GOTO 20
END IF

LOOP UNTIL INKEY$ = CHR$(27) '27 es el código ASCII para Esc.
END
```

Para controlar a dos motores podemos utilizar el siguiente código fuente en Basic:

```
CLS 0
```

```
'Salida de datos al registro 2
```

```
  HoraInicio = TIMER
```

```
  WHILE TiempoTranscurrido < 3
```

```
    TiempoTranscurrido = TIMER - HoraInicio
```

```
    OUT &H378, 21
```

```
  WEND
```

```
  HoraInicio = 0
```

```
'Salida de datos al registro 3
```

```
  HoraInicio = TIMER
```

```
  WHILE TiempoTranscurrido < 3
```

```
    TiempoTranscurrido = TIMER - HoraInicio
```

```
    OUT &H378, 37
```

```
  WEND
```

```
  HoraInicio = 0
```

```
'El programa espera a que el usuario presione ESC
```

```
'para finalizar el programa
```

```
  PRINT "Presione Esc para salir..."
```

```
  DO
```

```
  LOOP UNTIL INKEY$ = CHR$(27) '27 es el código ASCII para Esc.
```

```
'Al presionar ESC el programa borra los datos enviados a los
```

```
'registros 2 y 3
```

```
  HoraInicio = TIMER
```

```
  WHILE TiempoTranscurrido < 3
```

```
    TiempoTranscurrido = TIMER - HoraInicio
```

```
    OUT &H378, 16
```

```
  WEND
```

```
  HoraInicio = 0
```

```
  HoraInicio = TIMER
```

```
  WHILE TiempoTranscurrido < 3
```

```
    TiempoTranscurrido = TIMER - HoraInicio
```

```
    OUT &H378, 32
```

```
  WEND
```

```
  HoraInicio = 0
```

```
  PRINT "FIN"
```

```
  END
```

III.6.2 VISUAL BASIC

Las versiones de programación para Visual Basic en cuanto a las instrucciones para programar al puerto paralelo, tienen que utilizar una biblioteca de aplicación para Windows que es la que ejecuta las funciones de I / O (Input / Output) de acceso a puerto.

Documento para inpout32

Inpout32 es un DLL para el acceso del puerto en los programas de 32 bits

INFORMACIÓN IMPORTANTE

1. Use este DLL completamente. Escribiendo directamente a los puertos del hardware; Inpout32 fue desarrollado para permitir acceso hacia los puertos en el hardware de la computadora, como usted puede acostumbrarlo a acceder a cualquier hardware que es considerado como un puerto de I / O.
2. Use este DLL con los programas de 32 bits. Los programas de 16 bits requieren un DLL para 16 bits.
3. Windows permite el acceso del puerto directamente con Inpout32

inpout32.dll

Es un DLL que agrega instrucciones de entrada y salida (Inp y Out) de 32 bits que es elemento esencial para los programas en Visual.

inpout32.bas

Contiene las declaraciones para las funciones de entrada y salida (Inp y Out)

inpout32.vbp

Parte esencial para aplicaciones Visual que permiten la prueba de inpout32.dll

inpout32.frm

El formulario para inpout32.vbp

inpout32.dpr

Contiene el código fuente para el inpout32 (Delphi 2.0)

Cómo usar el inpout32:

1. Realice una copia inpout32.dll en su directorio de Windows predefinido.
2. Agregue inpout32.bas a su proyecto Básico Visual (Dentro del menú Archivo, Agregue el Archivo).
3. para escribir un byte a un puerto, se utiliza la siguiente sintaxis:
Out PortAddress, ByteToWrite

Para leer un byte de un puerto:

ByteRead = Inp(PortAddress)

Ejemplos:

Out &h378, &h55

ByteRead = Inp(&h378)

(La sintaxis es idéntica a Basic Inp y Out)

* * *

Cómo ejecutar el programa de prueba (inpout.vbp):

1. Ya realizada la copia inpout32.dll en su directorio de Windows predefinido.
2. Abra el proyecto inpout32.vbp.
3. PortAddress (la dirección del puerto definida).
3. Pulsando el botón de mando del programa es para escribir un valor al puerto. Cada click incrementa el valor entre 0 a 255.

CODIGO FUENTE

Option Explicit

Dim Value As Integer

Dim PortAddress As Integer

Private Sub cmdWriteToPort_Click()

'Escribe un byte en el Puerto , lo lee y vuelve a ejecutar

'y muestra el resultado

```

Out PortAddress, Value
Text1.Text = Inp(PortAddress)
Value = Value + 1
If Value = 255 Then Value = 0
End Sub
Private Sub Form_Load()
'Programa de prueba parainpout32.dll
Value = 0
'Cambie los valores para enviar y escribir en el Puerto deseado
'(direcciones usuales para el Puerto paralelo &h378, &h278, &h3BC)
PortAddress = &H278
End Sub

```

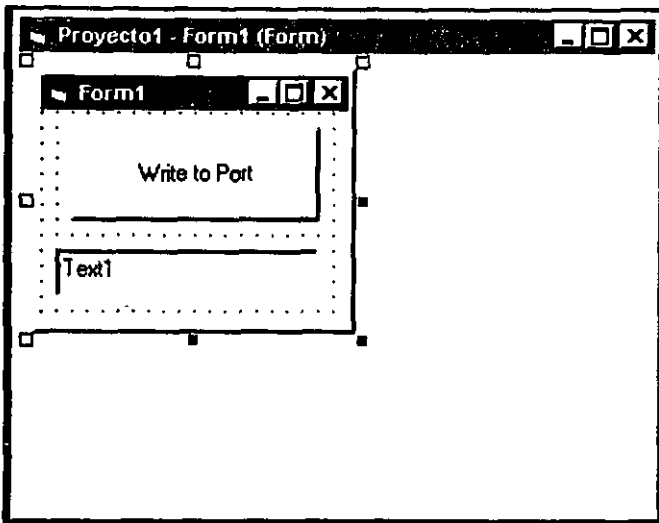


FIGURA 59: PROGRAMA PARA ESCRIBIR DATOS AL PUERTO PARALELO EN VISUAL BASIC

CAPITULO IV

DESARROLLO DE UNA APLICACION EN VISUAL BASIC

IV.1 DESARROLLO DE UN FICHERO POR MEDIO DE VISUAL BASIC

Para desarrollar una aplicación de registro tendremos que utilizar las herramientas de programación de Visual Basic.

Objetos

Un objeto es una entidad que tiene atributos particulares, los datos o propiedades, y unas formas de operar sobre ellos, los métodos o procedimientos. Cada objeto en Visual Basic está definido por una clase (class), entendiendo por clase un tipo de objeto.

Cuando es creada una aplicación en visual basic, trabaja con objetos y sobre un formulario se dibujan los controles; los formularios y los controles son objetos. Cada uno de estos objetos tienen asociadas un conjunto de propiedades.

Propiedades y métodos

De los objetos

Para referirse generalmente a una propiedad o a un método de un objeto es

Objeto.propiedad

Objeto.método

Ejemplo: Texto.text

Suma = Texto.text

Todos los controles tienen una propiedad que puede utilizar para almacenar o recuperar información

Crear objetos

La forma sencilla para crear un objeto dentro de la aplicación de visual basic es hacer doble clic en un control de la caja de herramientas. Pero también existe la forma de programar objetos haciendo referencia a ellos, o seleccionando al objeto en la caja de herramientas y dibujar las dimensiones del objeto sobre el formulario.

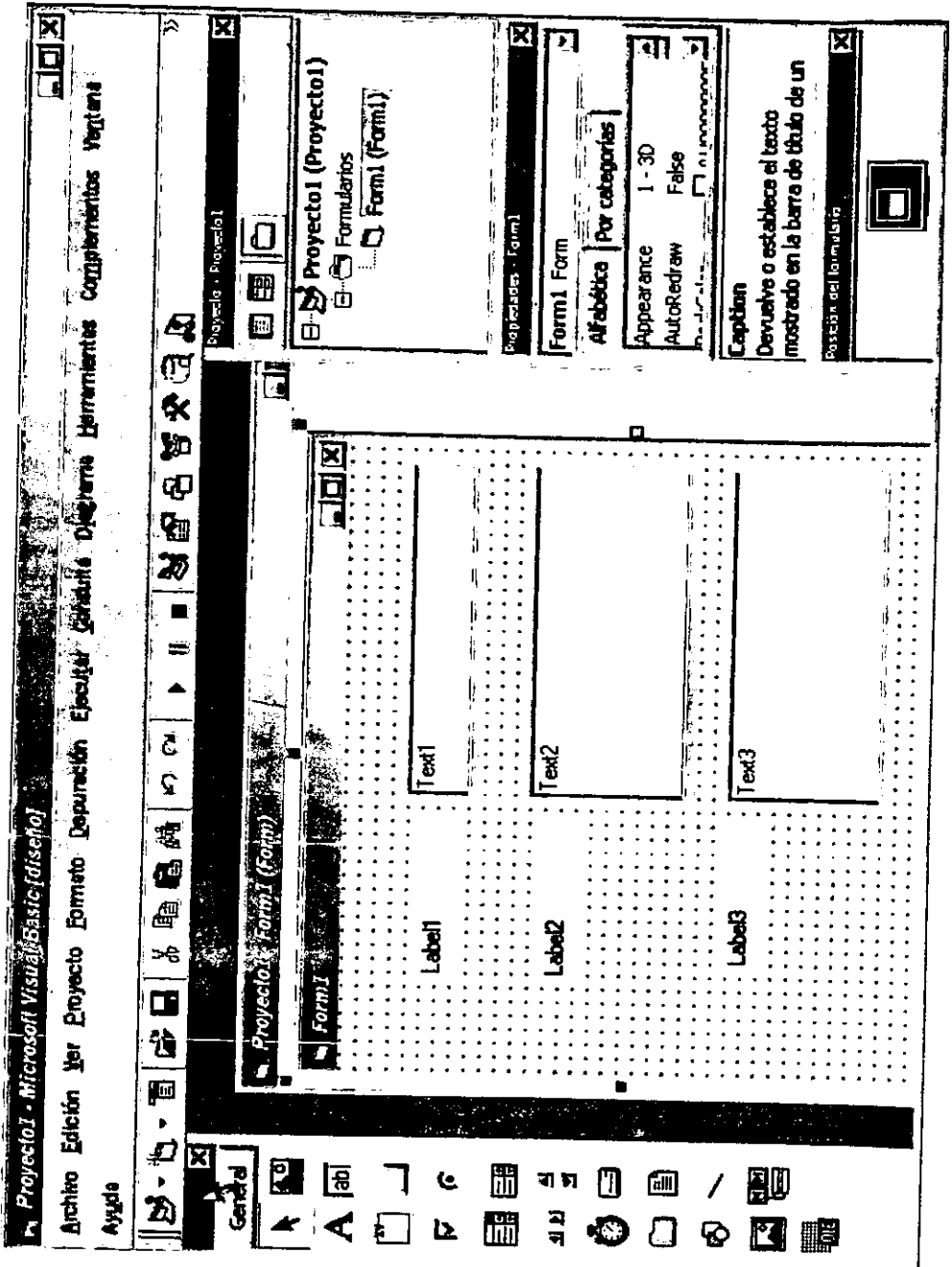


FIGURA 60: PAQUETE DE PROGRAMACIÓN VISUAL BASIC

Crear controles

Los controles, tales como caja de texto, etiquetas, listas, botones, son objetos gráficos que permiten introducir o extraer datos. El formulario más los controles forman una interfaz o medio de comunicación entre la aplicación gráfica y el usuario.

Para generar y escribir el código fuente de cada control solamente es necesario hacer doble clic en el objeto que se encuentre dibujado sobre un formulario.

¿Para que programar un fichero en Vbasic?

Un fichero en Vbasic es de gran utilidad por que en ellos se guardan bases de datos y las aplicaciones pueden ser guardadas como información central de datos para el usuario, con estos se pueden guardar tanto los bloques de instrucciones, así como sus referencias y especificaciones de cada función, para que cuando sean requeridos, puedan ser buscados y copiados a un archivo fuente de programación, que controle al circuito de salida.

Utilización de listas

Supongamos una pequeña base de datos para llevar la cuenta de los bloques de instrucciones ya utilizados, de los cuales, podamos obtener los siguientes datos:

Titulo del bloque de instrucciones
Referencia
Código fuente del bloque de instrucciones

Cada uno de estos datos elementales se denomina *campo*, y el conjunto de todos los campos referentes a un mismo libro recibe el nombre de *registro*.

La aplicación consta de una ventana principal que permite visualizar o introducir datos a un registro y de un menú que permite, buscar un determinado registro. Cuando el usuario seleccione este menú la orden de buscar registro aparecerá una caja de dialogo con una lista clasificada de los títulos de instrucciones, cuando el usuario haga un doble clic sobre uno de los títulos, los datos correspondientes a ese libro se visualizarán en la ventana principal.

Generalmente, una base de datos está clasificada por alguno de sus campos, para nuestro caso se encontrarán en el campo de - Título -.

Para diseñar esta aplicación, comenzamos con un nuevo proyecto en la aplicación de Visual Basic y asigne al formulario el nombre de Bloques de Instrucciones. Asignando la propiedad de **BorderStyle** el valor de 3 para evitar que se pueda redimensionar durante la ejecución y para que no se muestren los botones de maximizar y minimizar.

Continuamos con añadir 3 cajas de texto, a cada una de las cuales le asignaremos una etiqueta que los identifica.

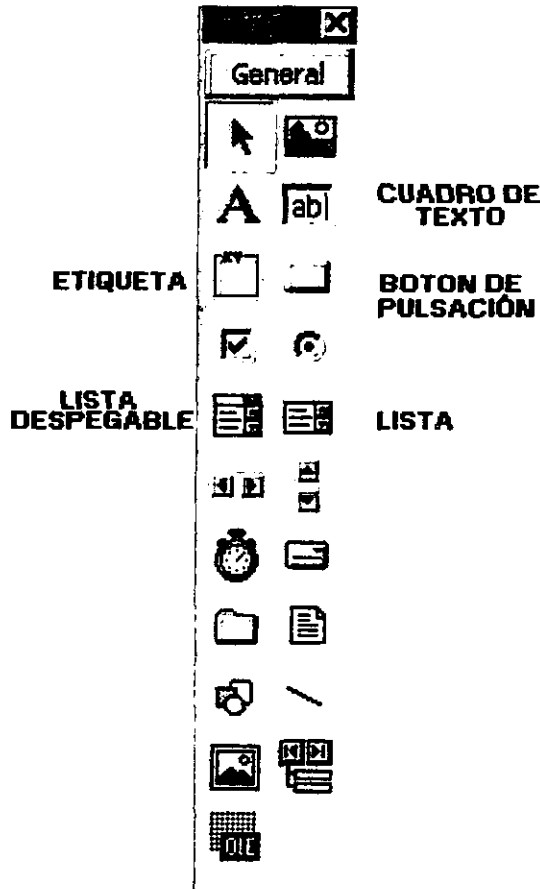


FIGURA 61: BARRA DE CONTROLES DE VISUAL BASIC

Las cajas de texto contendrán las siguientes propiedades:

Objeto	Propiedad	Valor
Etiqueta	Caption Name	Título: Label1
Caja de texto	Name	Título
Caja de texto	Name MultiLine ScrollBars	Referencia True 2
Etiqueta	Caption Name	Referencia: Label2
Caja de texto	Name MultiLine ScrollBars	Referencia True 2
Etiqueta	Caption Name	Instrucciones Label3
Caja de texto	Name MultiLine ScrollBars	Instrucciones True 2

**TABLA 14: CONTROLES Y PROPIEDADES
DE LA APLICACIÓN (VISUAL BASIC)**

Añada un menú con el nombre Fichero con las órdenes Añadir Registro, Buscar registro..., y salir. Para identificar estas órdenes, asigneles los nombres AñadirReg, BuscarReg y Salir, respectivamente. Esta opción se encuentra en el menú de herramientas.

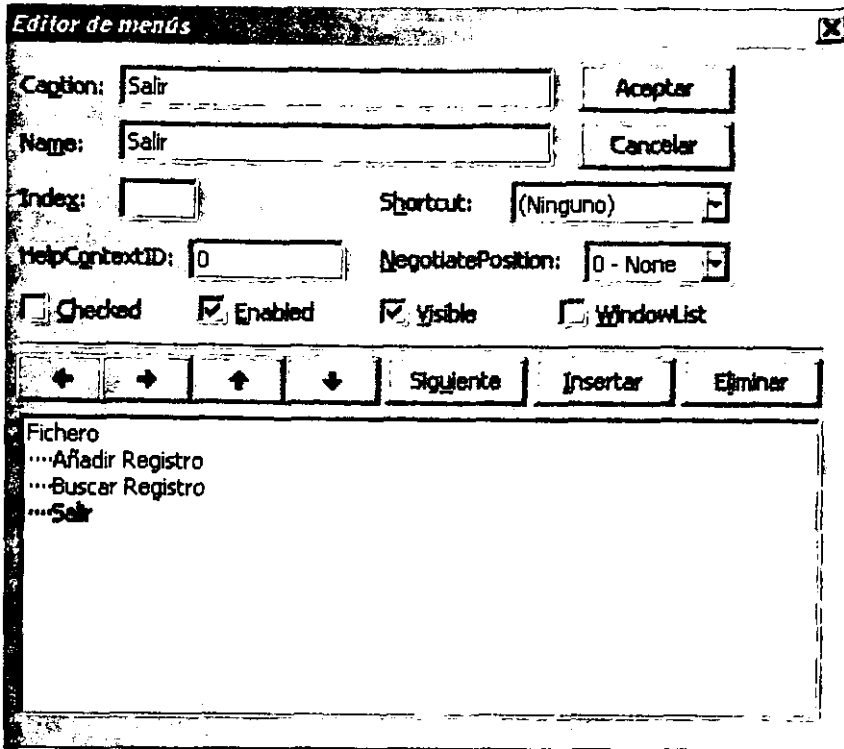


FIGURA 62: EDITOR DE MENUS

El trabajo realizado hasta ahora se guardara con el nombre de datos.frm y datos.vbp.

Seleccionar el texto de una caja de texto

Para cuando un usuario enfoca alguna caja de texto, Título, Referencia, Instrucciones, el texto de estas será seleccionado, con lo cual pueda ser remplazado automáticamente el texto de la caja seleccionada por un nuevo texto escrito, esto se realiza con la función **GotFocus**.

```
Private Sub Título_GotFocus( )
    'Seleccionar el texto de la caja
    Título.SelStart = 0
    Título.SelLength = Len(Título.Text)
End Sub
```


De la misma forma para las cajas de texto Referencia e Instrucciones.

Validar datos reteniendo el foco en el control

La propiedad **CausesValidation** se utiliza conjuntamente con el evento **Validate** para comprobar antes de permitir que el usuario se mueva a otro control.

```
Private Sub Título_Validate (Cancel As Boolean)
    'Si la Caja está vacía, mantener el foco
    If Len(Título.Text)=0 Then
        Cancel = True
        MsgBox "Tiene que escribir un título", , "Libros"
    End If
End Sub
```

Esta instrucción impide que el usuario haga clic sobre cualquier otro control.

Crear un módulo estándar

Nuestra base de datos va a estar formada por una matriz de registros denominada Libros en la que cada registro será del tipo *Registro*. Para almacenar estos datos en un fichero.

```
Type Registro
    Título As String * 50
    Referencia As String * 500
    Instrucciones As String * 500
End Type
Public Libros() As Registro
Public TotalRegs As Integer
```

Añada un nuevo módulo del menú proyecto y estas instrucciones serán colocadas dentro del siguiente módulo, guardando al módulo con el nombre de bibli.bas.

Añadir un Registro

El código correspondiente a la orden Añadir Registro del menú Fichero, es el siguiente. Cuando el usuario haga clic en esta orden, con lo cual el contenido de las cajas de texto será almacenado en la base de datos.

```

Private Sub AñadirReg_Click()
    TotalRegs = TotalRegs + 1
    'Asignar memoria para un registro más
    ReDim Preserve Libros(1 To TotalRegs)
    'Asignar los datos de las cajas a los campos del registro
    Libros(TotalRegs).Título = Título.Text
    Libros(TotalRegs).Referencia = Referencia.Text
    Libros(TotalRegs).Instrucciones = Instrucciones.Text
    'Añadir el título a la lista "ListaLibros"
    Form2.ListaLibros.AddItem Título.Text
End Sub

```

Buscar un registro

Para la función de buscar registro es necesario añadir un nuevo formulario, Form2, asignándole el nombre de Buscar Registro. Colocando la propiedad **BorderStyle** a valor 3 para evitar el cambio de tamaño durante su ejecución. Se añadirán a este formulario los controles de la tala 15.

Objeto	Propiedad	Valor
Lista	Name	ListaLibros
	Sorted	True
Botón de pulsación 1	Caption	Aceptar
	Name	Aceptar
	Default	True
Botón de pulsación 2	Caption	Cancelar
	Name	Cancelar
	Cancel	True
Botón de pulsación 3	Caption	Borrar
	Name	Borrar

TABLA 15: CONTROLES Y PROPIEDADES

Con la función Default en valor a True, la acción puede ejecutarse con la tecla Enter, sabiendo que solo puede haber en un formulario un botón con la propiedad default a True, en el botón cancelar la acción se puede ejecutar con la tecla Esc.

Para cuando la función de Buscar registro sea ejecutada, la caja de diálogo aparecerá enfocada con la función cancelar, con el siguiente código.

```
Private Sub BuscarReg_Click()  
    Form2.Show  
    Form2!Cancelar.SetFocus  
End Sub
```

Para visualizar el registro ligado al botón aceptar o hacer doble clic en el registro, es necesario el siguiente arreglo de instrucciones.

```
Private Sub Aceptar_Click()  
    Call VisualizarRegistro  
End Sub  
Private Sub ListaLibros_Db1Click()  
    Call VisualizarRegistro  
End Sub
```

Estas se escriben en el botón de aceptar y en la lista del formulario 2, (Form2)

Para cancelar la ventana de búsqueda en el botón de cancelar se escribe

```
Private Sub Cancelar_Click()  
    Form2.Hide  
End Sub
```

Para la orden de salir será:

```
Private Sub Salir_Click()  
    Unload Form2  
End  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    Unload Form2  
End Sub
```

El procedimiento para visualizar el registro tiene que realizar en primer lugar una búsqueda secuencial a partir del registro 1 para hallar la posición I en la que se encuentra el registro correspondiente al nombre buscado, una vez localizado, tiene que visualizarse los datos en las cajas de texto correspondientes del formulario 1. El procedimiento se añade al módulo bibli.bas

```
Public Sub VisualizarRegistro()  
    Dim I As Integer  
    For I = 1 To TotalRegs  
        If (RTrim(Libros(I).Título) =  
            RTrim(Form2.ListaLibros.Text)) Then  
            Exit For  
        End If  
    Next I  
    Form1.Título.Text = RTrim(Libros(I).Título)  
    Form1.Referencia.Text = RTrim(Libros(I).Referencia)  
    Form1.Instrucciones.Text = RTrim(Libros(I).Instrucciones)  
End Sub
```

Acceso a los elementos de una lista

Cuando la lista de registros es demasiado extensa y aproximarse a donde se encuentra el registro buscado con la letra inicial del nombre buscado, es necesario el siguiente bloque de instrucciones para que se realice el procedimiento indicado.

```
Private Sub ListaLibros_KeyUp(KeyCode As Integer, Shift As Integer)  
    Dim I As Integer  
    For I = 0 To TotalRegs - 1  
        If Left(ListaLibros.List(I), 1) = LCase(Chr(KeyCode)) Then  
            ListaLibros.ListIndex = I  
            Exit For  
        End If  
    Next I  
End Sub
```

Eliminar un registro

Cuando un registro ya no es necesario, se puede eliminar, con la función del botón de borrar, para esto cuando el usuario pulse el botón borrar, el elemento seleccionado de la lista tiene que borrarse de la lista de libros y de la matriz Libros. Este código se añade al botón de borrar del formulario 2.

```
Private Sub Borrar_Click()
Dim I As Integer, R As Integer
'Borrar el registro de la matriz
For I = 1 To TotalRegs
  If (RTrim(Libros(I).Titulo) =
    RTrim(Form2.ListaLibros.Text)) Then
    Exit For
  End If
Next I
If I > TotalRegs Then Exit Sub 'No encontrado
'Se borra
For R = I To TotalRegs - 1
  Libros(R) = Libros(R + 1)
Next R
TotalRegs = TotalRegs - 1
'Asignar memoria para un registro menos
If TotalRegs Then
  ReDim Preserve Libros(1 To TotalRegs)
Else
  Erase Libros
End If
'Borrar el registro de la lista
'Posición en la lista
R = Form2.ListaLibros.ListIndex
'Se borra
Form2.ListaLibros.RemoveItem R
End Sub
```

Inhabilitar controles

Finalmente cuando al ejecutarse el programa, tenemos que en determinados momentos de la ejecución la utilización de algunos controles no es necesaria esencialmente al comienzo de esté, por lo que es bueno impedir al usuario la utilización de dicho control.

Para esto escriba el siguiente código en el primer formulario.

```
Private Sub BuscarReg_Click()  
    Form2.Show  
    Form2!Cancelar.SetFocus  
    Form2!Aceptar.Enabled = False  
    Form2!Borrar.Enabled = False  
End Sub
```

Para el segundo formulario

```
Private Sub ListaLibros_Click()  
    Aceptar.Enabled = True  
    Borrar.Enabled = True  
End Sub
```

código para el botón borrar del formulario 2

```
Private Sub Borrar_Click()  
    ...  
    ...  
    ...  
'Asignar memoria para un registro menos  
If TotalRegs Then  
    ReDim Preserve Libros(1 To TotalRegs)  
Else  
    Erase Libros
```

<pre>Aceptar.Enabled = False Borrar.Enabled = False</pre>

```
End If
```

'Borrar el registro de la lista

...

...

End Sub

Cajas de diálogo comunes

El control *Microsoft Common Dialog* permite visualizar las cajas de diálogo más comúnmente empleadas en el diseño de aplicaciones, tales como Abrir y Guardar como, Imprimir, etc.

Para visualizar una de estas cajas de diálogo, realice los siguientes pasos:

1. Seleccione el control *diálogo común* en la caja de herramientas y dibújelo sobre el formulario, de no encontrarse, entonces hay que añadirlo, ejecutando la opción Componentes del menú Proyecto y eligiendo el control diálogo común de Microsoft (*Microsoft Common Dialog Control 6.0*)

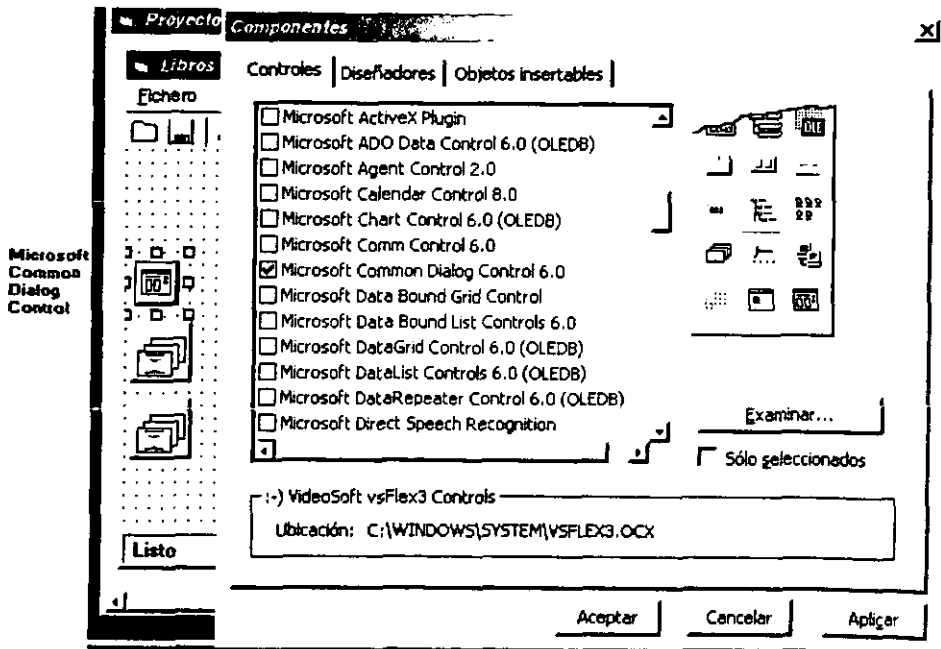


FIGURA 63: COMPONENTES PARA VISUAL BASIC

Para visualizar por ejemplo la caja de dialogo de Abrir se utiliza el comando:

CommonDialog1.ShowOpen

Método	Caja de dialogo
ShowOpen	Abrir
ShowSave	Guardar como
ShowColor	Color
ShowFont	Fuente
ShowPrinter	Impresora
ShowHelp	Invoca el motor de ayuda de Windows

TABLA 16: CUADROS DE DIALOGO COMUN

IV.2 Controles Activex

Uno de los controles intrínsecos de Visual Basic como las cajas de texto, los botones, están incluidos los controles activex que pueden ser agregados o quitados a la caja de herramientas.

Estos controles (Activex), existen como ficheros independientes con extensión *.ocx*, algunos de los controles Activex incluidos con todas las ediciones de Vbasic son: Fichas etiquetadas, barra de herramientas, barra de estados, lista de imágenes, etc.

Control Lista de imágenes

En ocasiones, observamos en las aplicaciones imágenes de forma relacionada con alguna operación.

Una lista de imágenes es una colección de imágenes del mismo tamaño accesibles por su índice.

Agregar una lista de imágenes

Para agregar el control de ImageList, se puede encontrar en el submenú de Componentes del menú Proyecto y añada la componente Activex denominada *Microsoft Windows Common Controls*.

Para agregar a la imagen que se requiere, primero tiene que visualizar el diálogo Páginas de propiedades del control *ImagenList*; Haciendo click sobre el control utilizando con el botón derecho y elija la orden propiedades del menú contextual que se utiliza.

1. Haga click en la ficha General y elija el tamaño de la imagen
2. Haga click en la ficha Imágenes.
3. Haga click en el botón insertar imagen y por medio de la caja de diálogo seleccione la imagen deseada.
4. Opcionalmente puede asignar un valor único a la propiedad Key y a la propiedad Tag.
5. Repita los pasos 3 y 4 hasta añadir las imágenes requeridas
6. Cuando haya terminado de click en el botón aceptar.

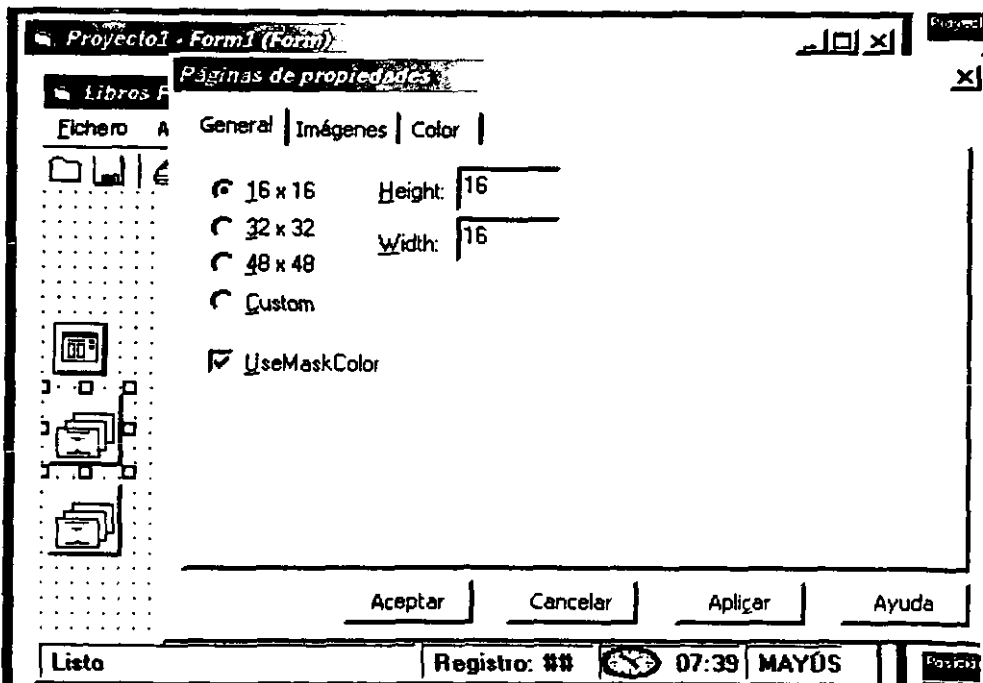


FIGURA 64: PROPIEDADES DE IMAGEN

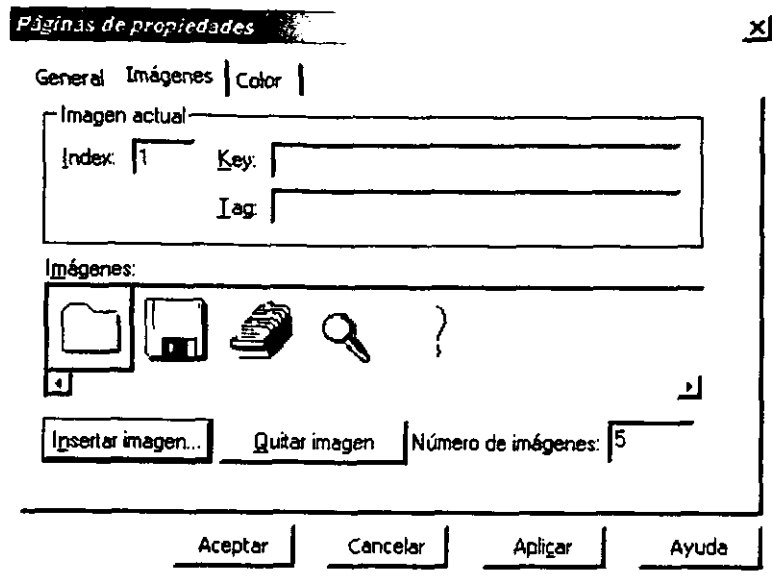
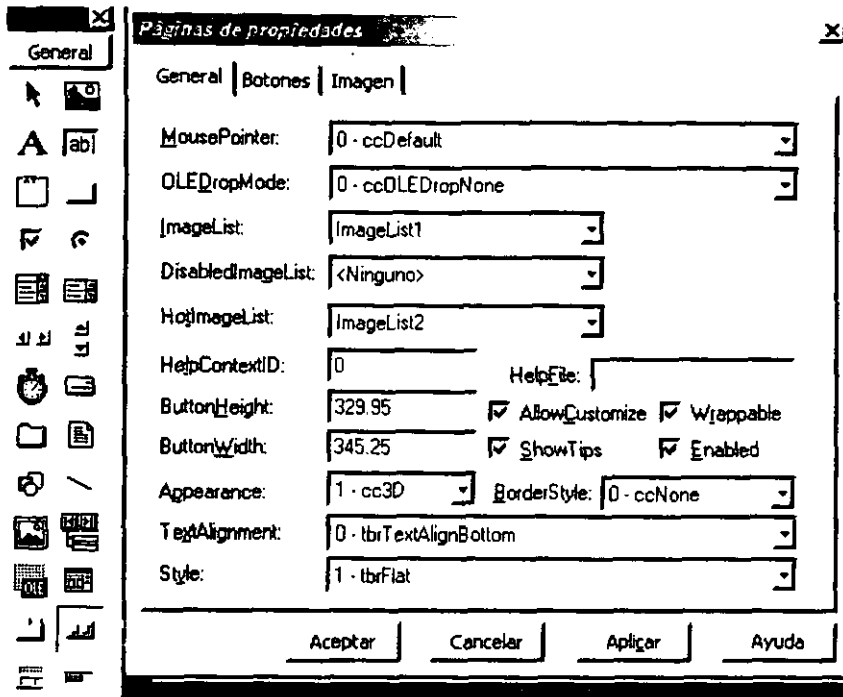


FIGURA 65: PAGINAS DE PROPIEDADES DE IMAGEN

IV.3 Barra de herramientas

Para la utilización de una barra de herramientas en un formulario es necesario la agregación del control *ToolBar* y un control *ImageList* a dicho formulario. Primero hay que visualizar la caja de diálogo Páginas de propiedades del control *ToolBar*, esto se hace realizando un click sobre el control con el botón derecho del ratón y eligiendo el menú propiedades, se siguen los siguientes pasos.

1. En la ficha General, seleccione *ImageList* y asígnele el nombre de *ImageList* que está utilizando.
2. En la ficha de Botones, haga click en el botón insertar botón y asigne valores a las propiedades del botón que está insertado; por ejemplo a *Caption* (título del botón), *Key* (Cadena que identifica unívocamente un miembro de una colección), *Style* (estilo del botón: *tbrDefault* para un botón de pulsación o *tbrSeraparador* para añadir un separador, el cual no requiere ninguna propiedad), *ToolTipText* (texto breve que informa acerca de la funcionalidad del botón), o *Image* (índice de la imagen de la lista de imágenes, que se asigna al botón).
3. Repita el paso anterior para añadir los botones que sean requeridos.
4. Cuando termine de añadir botones de click en el botón de aceptar.



**FIGURA 66: PROPIEDADES GENERALES
BARRA DE HERRAMIENTAS**

Para vincular un menú a un botón de la barra de herramientas, visualice la caja de diálogo Páginas de propiedades de la barra, haga clic en Botones y sitúe sobre el botón al que quiere vincular el menú, como se muestra en la figura 67, asignado el estilo tbrDropDown, escribiendo en la propiedad Text el título del elemento del botón, así sucesivamente hasta añadir todos los elementos del menú.

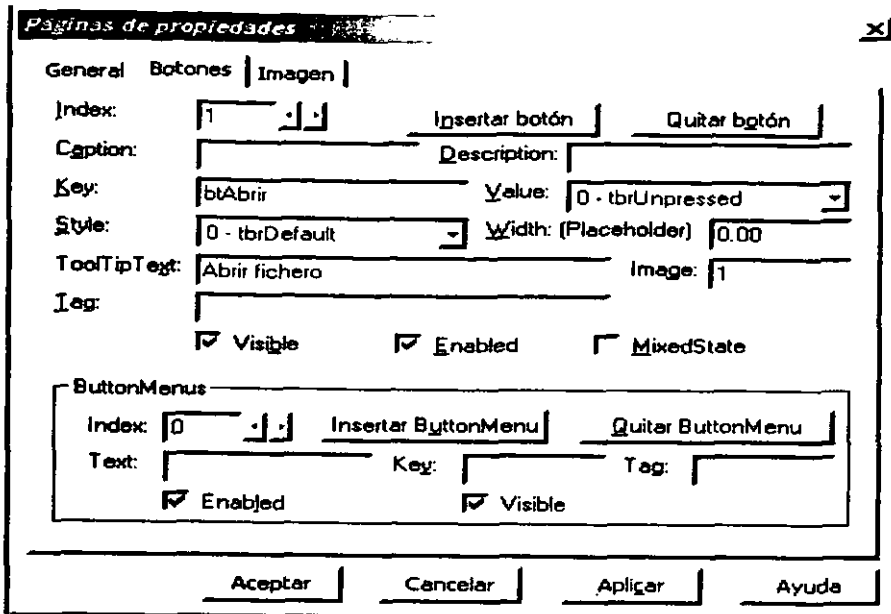


FIGURA 67: PROPIEDADES DE LOS BOTONES

Control barra de estado

Una barra de estado es una ventana, normalmente ubicada en la parte inferior de un formulario, con la cual se muestra información de ayuda al usuario. Para añadir rápidamente una barra de estado a una aplicación es utilizado el control StatusBar, el cual consta de uno a más paneles hasta un máximo de 16 contenidos de un grupo de paneles.

Para añadir, es necesario hacer un click con el botón derecho y elija la orden de propiedades en la ficha de Paneles haga click en Insertar Panel y asigne valores a las propiedades del panel que esta insertando.

Text, texto inicial que aparecerá en el panel

Key, cadena que identifica unívocamente un miembro de una colección

Style, estilo del panel

Bevel, especifica si el panel simulará una profundidad, un relieve, o estará plano

AutoSize, determina como el panel ajustara su tamaño cuando el tamaño del contenedor cambie.

El código completo de una aplicación de fichero se muestra a continuación:

Formulario 1

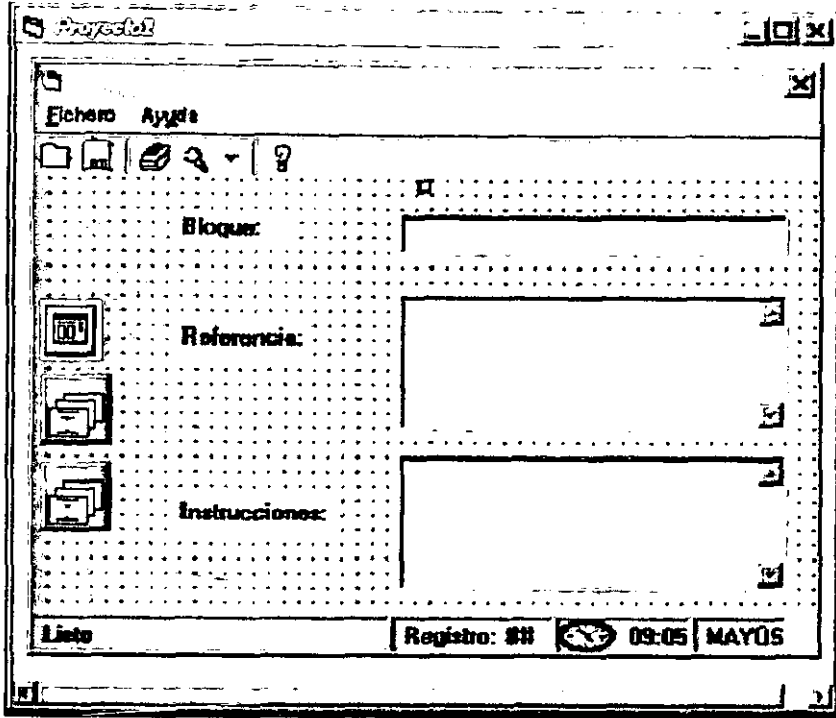


FIGURA 68: FORMULARIO 1
APLICACIÓN FICHERO

Option Explicit

```
Private Sub AbrirFichero()
```

```
Dim NumRegsFich As Integer, LongReg As Integer, I As Integer
```

```
Dim NumFichero1 As Integer
```

```
' Obtener un número libre de fichero
```

```
NumFichero1 = FreeFile
```

```
LongReg = Len(Libro)
```

```
On Error GoTo RutinaErrorAbrir
```

```
Open bdLibros For Random As #NumFichero1 Len = LongReg
```

```
NumRegsFich = LOF(NumFichero1) \ LongReg
```

```

' Borrar todas las entradas de la lista
Form2.ListaLibros.Clear
' Número de elementos de la lista
TotalRegs = NumRegsFich
' Añadir los títulos almacenados en el fichero a la lista
For I = 1 To TotalRegs
    Get #NumFichero1, I, Libro
    Form2.ListaLibros.AddItem Trim(Libro.Título)
Next I
' Visualizar el primer registro
Get #NumFichero1, 1, Libro
Form1!Título.Text = Trim(Libro.Título)
Form1!Referencia.Text = Trim(Libro.Referencia)
Form1!Instrucciones.Text = Trim(Libro.Instrucciones)
Close NumFichero1
StatusBar1.Panels("nreg").Text = "Registro: " & 1
SalirAbrir:
Exit Sub
RutinaErrorAbrir:
MsgBox "Error: " & Err.Description, 48, "Libros"
Close
Resume SalirAbrir
End Sub

```

```

Private Sub AñadirReg_Click()
    Dim mensaje As String
    If Len(Título.Text) = 0 Then
        'MsgBox "Tiene que escribir un título", , "Libros"
        mensaje = "Tiene que escribir un título"
        Form1.StatusBar1.Panels("Info").Text = mensaje
        Exit Sub
    End If
    Dim NumRegs As Integer
    Dim NumFichero1 As Integer
    ' Obtener un número libre de fichero
    NumFichero1 = FreeFile
    Libro.Título = Form1!Título.Text
    Libro.Referencia = Form1!Referencia.Text
    Libro.Instrucciones = Form1!Instrucciones.Text
    Open bdLibros For Random As #NumFichero1 Len = Len(Libro)

```

```

NumRegs = LOF(NumFichero1) \ Len(Libro)
Put #NumFichero1, NumRegs + 1, Libro
Close NumFichero1
Form2.ListaLibros.AddItem Título.Text
TotalRegs = TotalRegs + 1
StatusBar1.Panels("nreg").Text = "Registro: " & TotalRegs
End Sub

```

```

Private Sub AyudaAcercaDe_Click()
    AcercaDe.Show vbModal
End Sub

```

```

Private Sub Form_Load()
    bdLibros = "Temporal"
End Sub

```

```

Private Sub StatusBar1_PanelClick(ByVal Panel As MSCComctlLib.Panel)
    Static bFechaHora As Boolean
    bFechaHora = Not bFechaHora
    If Panel.Key = "FechaHora" And bFechaHora Then
        StatusBar1.Panels("FechaHora").Picture = Nothing
        StatusBar1.Panels("FechaHora").Style = sbrDate
    Else
        StatusBar1.Panels("FechaHora").Picture = _
            LoadPicture(App.Path & "\Reloj.ico")
        StatusBar1.Panels("FechaHora").Style = sbrTime
    End If
End Sub

```

```

Private Sub Título_GotFocus()
    ' Seleccionar el texto de la caja
    Título.SelStart = 0
    Título.SelLength = Len(Título.Text)
End Sub

```

```

Private Sub Instrucciones_GotFocus()
' Seleccionar el texto de la caja
Instrucciones.SelStart = 0
Instrucciones.SelLength = Len(Instrucciones.Text)
End Sub

```

```

Private Sub Referencia_GotFocus()
' Seleccionar el texto de la caja
Referencia.SelStart = 0
Referencia.SelLength = Len(Referencia.Text)
End Sub

```

```

Private Sub BuscarReg_Click()
Form2.Show
Form2.Cancelar.SetFocus
Form2!Aceptar.Enabled = False
Form2!Borrar.Enabled = False
End Sub

```

```

Private Sub FicheroAbrir_Click()
On Error GoTo ManipularErrorAbrir
' Generar un error cuando se pulse Cancelar
CommonDialog1.CancelError = True
CommonDialog1.Filter = "BD de libros (*.bdl)|*.bdl|" & _
    "Todos (*.*)|*.*"
' Filtro por defecto
CommonDialog1.FilterIndex = 1
' Visualizar la caja de diálogo Abrir
CommonDialog1.ShowOpen
' CommonDialog1.FileName contiene el camino y
' el nombre del fichero elegido
bdLibros = CommonDialog1.FileName
AbrirFichero
SalirAbrir:
Exit Sub
ManipularErrorAbrir:
' Manipular el error

```



```

If Err.Number = cdlCancel Then Exit Sub
MsgBox Err.Description
Resume SalirAbrir
End Sub

```

```

Private Sub FicheroGuardar_Click()
On Error GoTo ManipularErrorGuardar
' Generar un error cuando se pulse Cancelar
CommonDialog1.CancelError = True
CommonDialog1.Filter = "BD de libros (*.bdl)|*.bdl|" & _
    "Todos (*.*)|*.*"
' Filtro por defecto
CommonDialog1.FilterIndex = 1
' Visualizar la caja de diálogo Guardar como
CommonDialog1.ShowSave
' CommonDialog1.FileName contiene el camino y
' el nombre del fichero elegido
GuardarFichero
SalirGuardar:
Exit Sub
ManipularErrorGuardar:
' Manipular el error
If Err.Number = cdlCancel Then Exit Sub
MsgBox Err.Description
Resume SalirGuardar
End Sub

```

```

Private Sub GuardarFichero()
Dim I As Integer, Cadena As String, bdNueva As String
Dim NumFichero1 As Integer, NumFichero2 As Integer, R As Integer
On Error GoTo RutinaErrorGuardar
' Fichero especificado por el usuario para guardar los datos
bdNueva = CommonDialog1.FileName
' Renombrar el fichero viejo
Name bdLibros As "Temp" 'si "bdLibros" no existe, error 53
' Obtener un número libre de fichero
NumFichero1 = FreeFile
' Escribir en el fichero
Open bdNueva For Random As #NumFichero1 Len = Len(Libro)
If (LOF(NumFichero1) <> 0) Then

```

```

Cadena = "El fichero ya existe" & vbCrLf
Cadena = Cadena & "¿desea sobre escribirlo?"
If MsgBox(Cadena, 36, "Libros") = vbYes Then
    Close NumFichero1
    Kill bdNueva
    Open bdNueva For Random As #NumFichero1 Len = Len(Libro)
Else
    Close NumFichero1
    Exit Sub
End If
End If
' Obtener un número libre de fichero
NumFichero2 = FreeFile
' Escribir en el fichero los registros que tienen su
' correspondencia en la lista
Open "Temp" For Random As #NumFichero2 Len = Len(Libro)
For R = 1 To LOF(NumFichero2) \ Len(Libro)
    Get #NumFichero2, , Libro
    For I = 0 To TotalRegs - 1
        If (Trim(Libro.Título) = Trim(Form2.ListaLibros.List(I))) Then
            Put #NumFichero1, , Libro
        End If
    Next I
Next R
Close NumFichero1, NumFichero2
Kill "Temp"
bdLibros = CommonDialog1.FileName 'nuevo fichero creado
Form1.Título.SetFocus
SalirGuardar:
Exit Sub
RutinaErrorGuardar:
Select Case Err
    Case 53 'el fichero no existe
        Resume Next
    Case Else

```

```
    MsgBox "Error: " & Err.Description, 48, "Libros"  
End Select  
Resume SalirGuardar  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
    Unload Form2  
End Sub
```

```
Private Sub Salir_Click()  
    Unload Form2  
End  
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)  
    Select Case Button.Key  
        Case "btAbrir"  
            StatusBar1.Panels(1).Text = "Fichero cargado"  
            FicheroAbrir_Click  
        Case "btGuardar"  
            StatusBar1.Panels(1).Text = "Fichero guardado"  
            FicheroGuardar_Click  
        Case "btAñadir"  
            StatusBar1.Panels(1).Text = "Registro añadido"  
            AñadirReg_Click  
        Case "btBuscar"  
            StatusBar1.Panels(1).Text = "Diálogo buscar"  
            BuscarReg_Click  
        Case "btAyuda"  
            StatusBar1.Panels(1).Text = "Acerca de..."  
            AyudaAcercaDe_Click  
    End Select  
End Sub
```

```
Private Sub Toolbar1_ButtonMenuClick(ByVal MenuBoton As  
MSComctlLib.ButtonMenu)  
    ' Manipular el menú del botón "Buscar"  
    If TotalRegs = 0 Then Exit Sub  
    Dim I As Integer, NumFichero As Integer  
    ' Obtener un número libre de fichero
```

```

NumFichero = FreeFile
Open bdLibros For Random As #NumFichero Len = Len(Libro)
' Número de registro a leer
Select Case MenuBoton.Key
Case "elementoPrimero"
    I = 1
Case "elementoAnterior"
    I = Mid(StatusBar1.Panels("nreg").Text, 10) - 1
    If I = 0 Then I = 1
Case "elementoSiguiente"
    I = Mid(StatusBar1.Panels("nreg").Text, 10) + 1
    If I > TotalRegs Then I = TotalRegs
Case "elementoUltimo"
    I = TotalRegs
End Select
' Leer el registro
Get #NumFichero, I, Libro
Close NumFichero
' Visualizar registro
Form1.Título.Text = Trim(Libro.Título)
Form1.Referencia.Text = Trim(Libro.Referencia)
Form1.Instrucciones.Text = Trim(Libro.Instrucciones)
StatusBar1.Panels("nreg").Text = "Registro: " & I
End Sub

```

Formulario 2

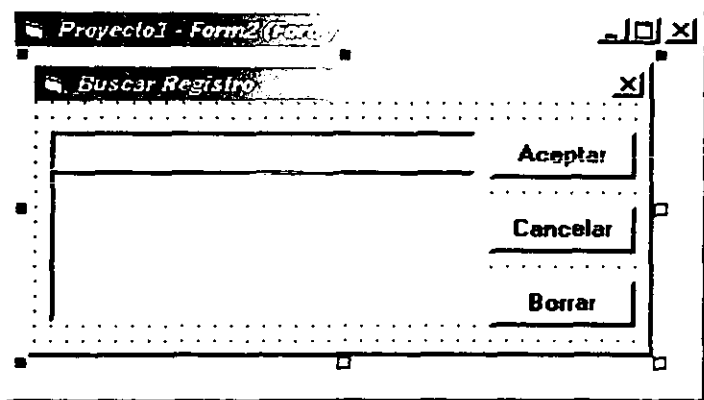


FIGURA 69: FORMULARIO 2 (REGISTRO)

Option Explicit

```
Private Sub Aceptar_Click()  
    Call VisualizarRegistro  
End Sub
```

```
Private Sub ListaLibros_Db1Click()  
    Call VisualizarRegistro  
End Sub
```

```
Private Sub Cancelar_Click()  
    Form2.Hide  
End Sub
```

```
Private Sub Borrar_Click()  
    Dim I As Integer  
    'Borrar el registro de la lista  
    'Posición en la lista  
    I = Form2!ListaLibros.ListIndex  
    'Se borra  
    Form2.ListaLibros.RemoveItem I  
    TotalRegs = Form2!ListaLibros.ListCount  
End Sub
```

```
Private Sub ListaLibros_Click()  
    Aceptar.Enabled = True  
    Borrar.Enabled = True  
End Sub
```

```
Private Sub ListaLibros_KeyUp(KeyCode As Integer, Shift As Integer)  
    Dim I As Integer  
    For I = 0 To TotalRegs - 1  
        If Left(ListaLibros.List(I), 1) = LCase(Chr(KeyCode)) Then  
            ListaLibros.ListIndex = I  
            Exit For  
        End If  
    Next I  
End Sub
```

Formulario 3

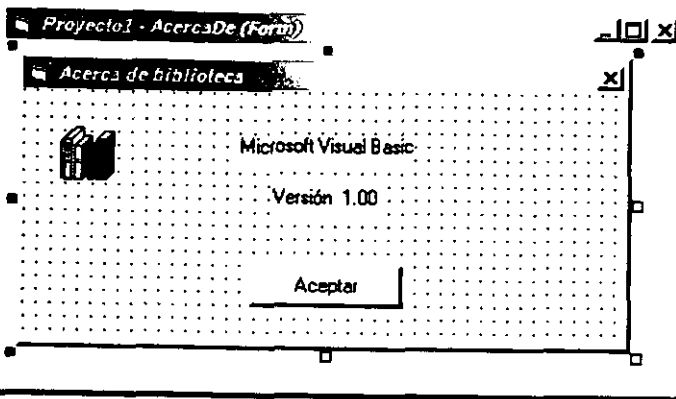


FIGURA 70: FORMULARIO 3 (PROYECTO REGISTRO)

Option Explicit

```
Private Sub Aceptar_Click()  
    AcercaDe.Hide  
End Sub
```

```
Private Sub Aceptar_KeyDown(KeyCode As Integer, Shift As Integer)  
    Aceptar.Value = True 'llama a Aceptar_Click  
End Sub
```

Modulo

Option Explicit

```
Type Registro  
    Titulo As String * 30  
    Autor As String * 30  
    Referencia As String * 240  
    Instrucciones As String * 240  
End Type  
Public Libro As Registro  
Public TotalRegs As Integer  
Public bdLibros As String
```

```

Public Sub VisualizarRegistro()
    Dim I As Integer, Encontrado As Boolean
    Dim NumFichero As Integer
    ' Obtener un número libre de fichero
    NumFichero = FreeFile
    Encontrado = False 'elemento no encontrado
    'Buscar el registro seleccionado
    Open bdLibros For Random As #NumFichero Len = Len(Libro)
    For I = 1 To LOF(NumFichero) \ Len(Libro)
        Get #NumFichero, I, Libro
        If (Trim(Libro.Título) = Trim(Form2.ListaLibros.Text)) Then
            Encontrado = True 'elemento encontrado
            Exit For
        End If
    Next I
    Close NumFichero
    If (Encontrado) Then
        'Visualizar registro
        Form1.Título.Text = Trim(Libro.Título)
        Form1.Referencia.Text = Trim(Libro.Referencia)
        Form1.Instrucciones.Text = Trim(Libro.Instrucciones)
        Form1.StatusBar1.Panels("nreg").Text = "Registro: " & I
    Else
        MsgBox "El libro especificado no se encuentra", 48, "Libros"
    End If
End Sub

```

IV.4 PROGRAMA PARA INTERACTUAR CON LAS ENTRADAS PROVENIENTES DEL PUERTO PARALELO

El siguiente programa tiene como finalidad mostrar una de las partes más interesantes de un programa, los elementos gráficos; En general los gráficos dentro de Visual Basic pueden ser manipulados por medio de diversas utilidades (Líneas, Figuras, Imagen).

Un control de imagen que es un área rectangular dentro de la cual se pueden cargar ficheros de imágenes. Para añadir una imagen durante la ejecución utilizamos la función **LoadPicture**. Esta función permite asignar un fichero jpg, gif, ico, etc. a la propiedad **Picture** de un formulario de una caja de imagen; Para cargar una imagen durante el inicio de la ejecución las instrucciones se escriben durante el procedimiento **Form_Load**, ejemplo:

```
Private Sub Form_Load( )  
    Image1.Picture = LoadPicture("imagen.bmp")  
End Sub
```

El programa despliega una serie de imágenes dentro de un cuadro de imágenes, las cuales son cargadas cuando se pulse un botón (IMÁGENES) que cuenta con las instrucciones de lectura de datos provenientes del puerto paralelo, y la función de **LoadPicture**, de acuerdo con el diagrama de flujo de la figura 71, dependiendo de la señal de entrada que se este manejando la salida de imágenes que se mostraran en la pantalla será diferente:

Primeramente se diseña en Visual Basic sobre el formulario el cuadro de controles correspondientes al programa; Este programa mostrara una serie de imágenes siempre y cuando exista una señal proveniente del puerto paralelo, según sea la petición que se este realizando, además de las variables que se manejen internamente dentro del programa.

La variable que se maneja en la entrada del puerto paralelo para este programa es num2 que provienen del circuito de entrada, (esta entrada se conecta al pin 15 del conector del puerto paralelo) para sea admitida hacia la computadora mediante la instrucción **INP** del programa; Si se requiere anexar más variables de entrada a través del puerto paralelo, solamente se necesita otro bloque del circuito de entrada que se conectara a los sucesivos pines del registro de estados del puerto paralelo (pines 10, 11, 12, 13).

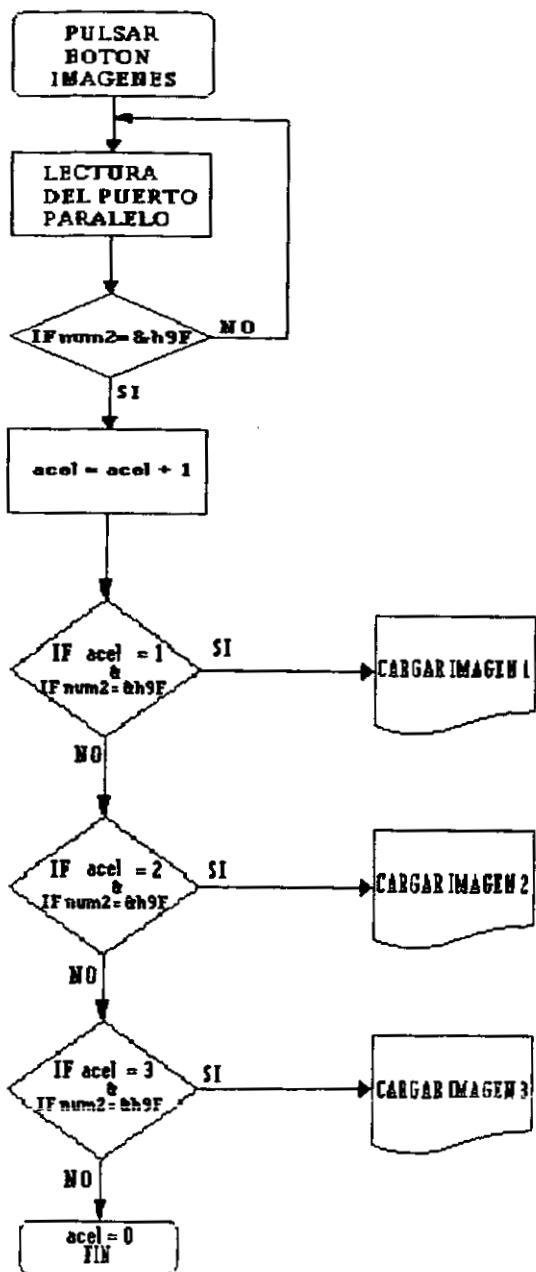
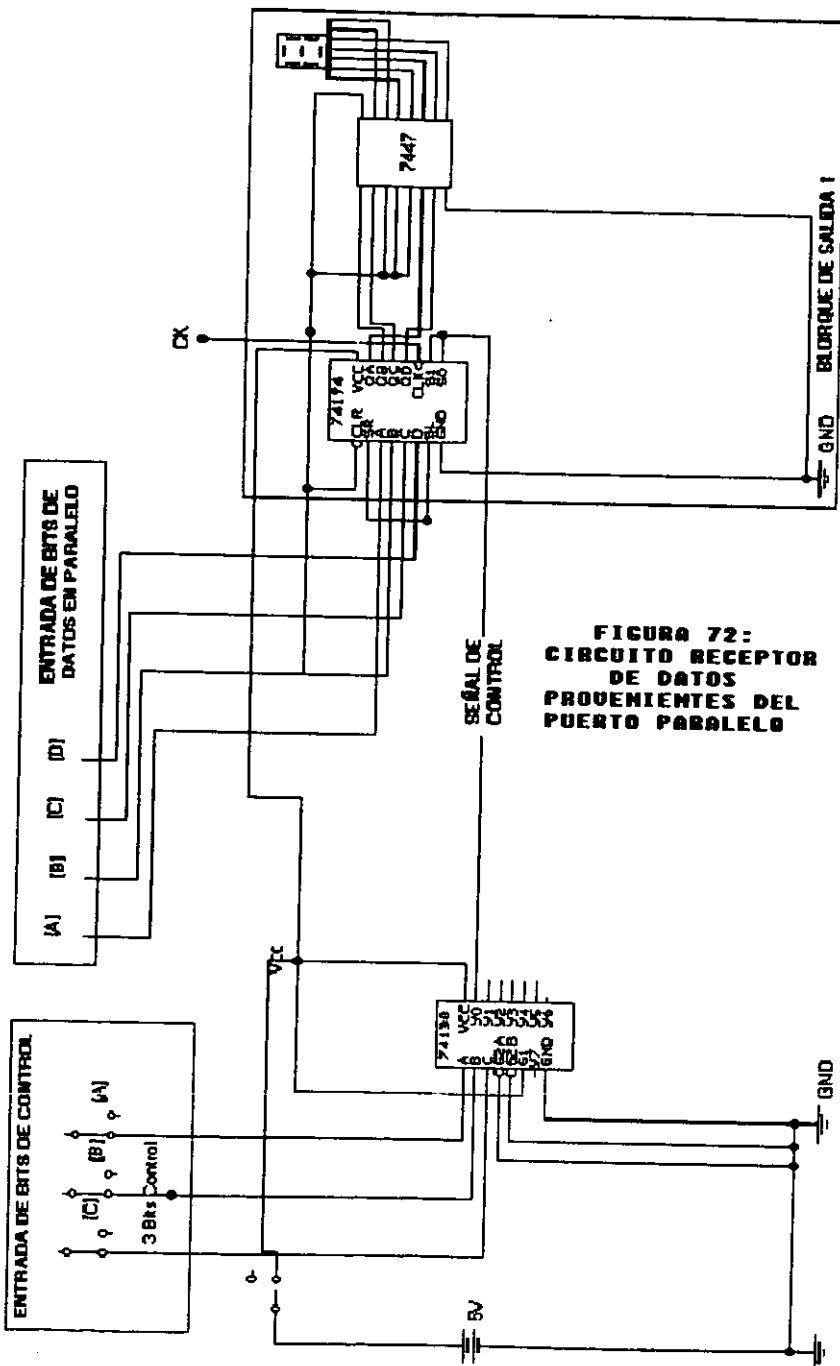


FIGURA 71: DIAGRAMA DE FLUJO
(PROGRAMA PARA ENTRADAS DEL PUERTO PARALELO)



**FIGURA 72:
CIRCUITO RECEPTOR
DE DATOS
PROVENIENTES DEL
PUERTO PARALELO**

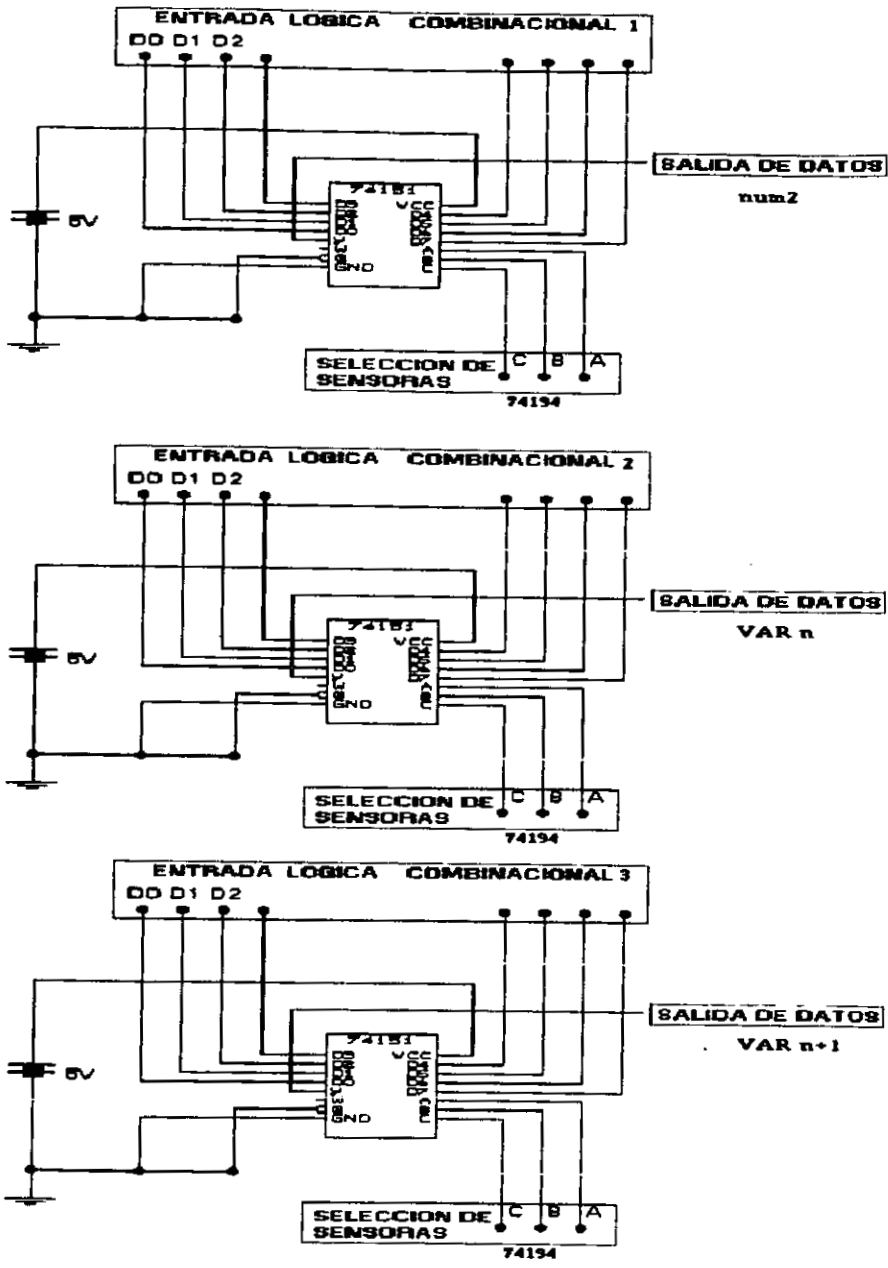


FIGURA 73: CIRCUITO PARA ENVIO DE DATOS AL PUERTO PARALELO

Antes de escribir el código fuente del programa es necesario crear los controles para el programa, mediante el editor de Visual Basic, en un nuevo proyecto agregando los controles y sus propiedades como se muestran en la tabla 17:

OBJETO	PROPIEDAD	VALOR
Boton de pulsación	Caption Nombre	SALIR Salir
Boton de pulsación	Caption Nombre	RESET Command1
Boton de pulsación	Caption Nombre	IMÁGENES Command2
Cuadro de texto	Caption Text	Text1 Text1
Cuadro de texto	Caption Text	Text2 Text2
Cuadro de texto	Caption Text	Text3 Text3
Cuadro de texto	Caption Text	Text4 Text4
Cuadro de texto	Caption Text	Text5 Text5
Imagen (Image)	Nombre BorderStyle	Image1 1
Etiqueta	Caption Nombre	378 Label1
Etiqueta	Caption Nombre	379 Label2
Etiqueta	Caption Nombre	num1 Label3
Etiqueta	Caption Nombre	num2 Label4
Etiqueta	Caption Nombre	acel Label5

TABLA 17: CUADRO DE CONTROLES Y PROPIEDADES

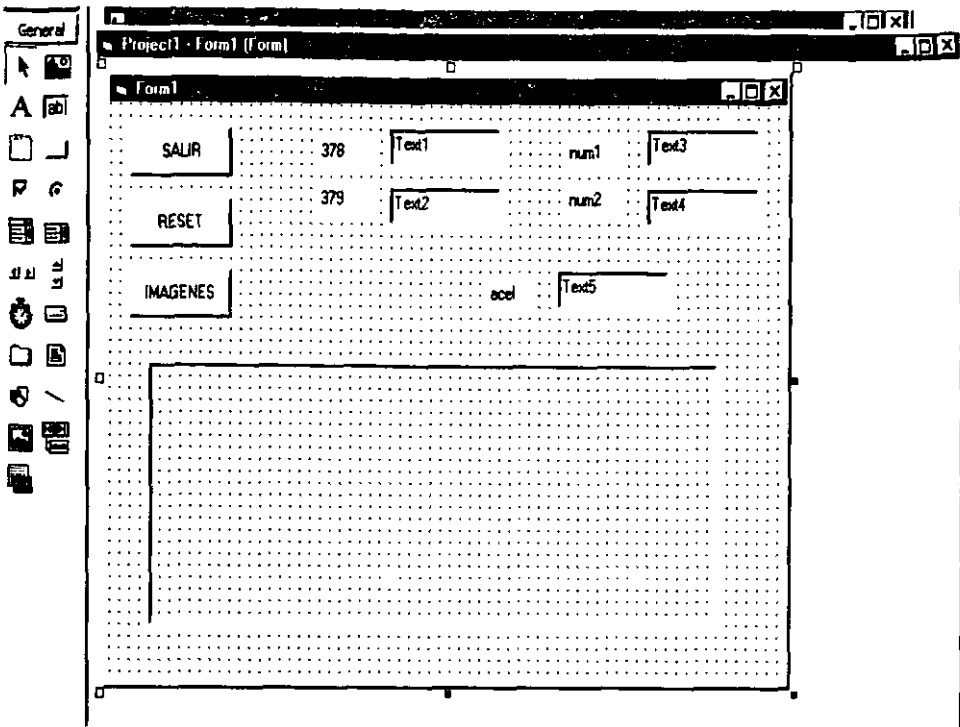


FIGURA 74: FORMULARIO DEL PROGRAMA PARA INTERACTUAR CON LAS ENTRADAS DEL PUERTO PARALELO

El código fuente para el programa es el siguiente, los cuales se agregaran a los a los botones de pulsación para ejecutar dichas instrucciones:

Option Explicit

'DECLARACION DE VARIABLES

Dim Value As Integer

Dim PortAddress As Integer

Dim num1, res, num2, acel, As Integer

Dim HoraInicio As Variant

Dim HoraParada As Variant

Dim Lapsus As Variant

'ESTAS INSTRUCCIONES SE EJECUTARAN INICIALMENTE,
'CUANDO SE EJECUTE EL PROGRAMA

Private Sub Form_Load()

'PROGRAMA PARA EVALUAR inpout32.dll

Value = 0

'CAMBIE ESTE VALOR POR EL NUMERO DEL PUERTO DONDE SE
'ESCRIBIRAN LOS DATOS:

'(USUALMENTE LAS DIRECCIONES DEL PUERTO PARALELO SON
'&h378, &h278, &h3BC)

PortAddress = &H378

'INSTRUCCION DE CARGA DE IMAGEN EN Image1

Image1.Picture = LoadPicture("C:\...\Dibujo.jpg")

Out PortAddress, &H11

End Sub

Private Sub Command1_Click()

Out PortAddress, &H11

End Sub

Private Sub Command2_Click()

GoTo 30

10 acel = 0

Out PortAddress, &H13

GoTo 20

30

Text1.Text = Inp(PortAddress)

Text2.Text = Inp(&H379)

'SE UTILIZA LA FUNCION VAL PARA ASEGURARNOS

'QUE TRABAJAMOS CON VALORES NUMERICOS

num1 = Val(Text1.Text)

num2 = Val(Text2.Text)

If num2 = &H9F Then acel = acel + 1

Text3.Text = Hex(num1) 'VALOR DE NUM1

Text4.Text = Hex(num2) 'VALOR DE NUM2

Text5.Text = (acel) 'valor de acel

If acel = 1 And num2 = &H9F Then Image1.Picture =
LoadPicture("C:\...\Dibujo1.jpg")

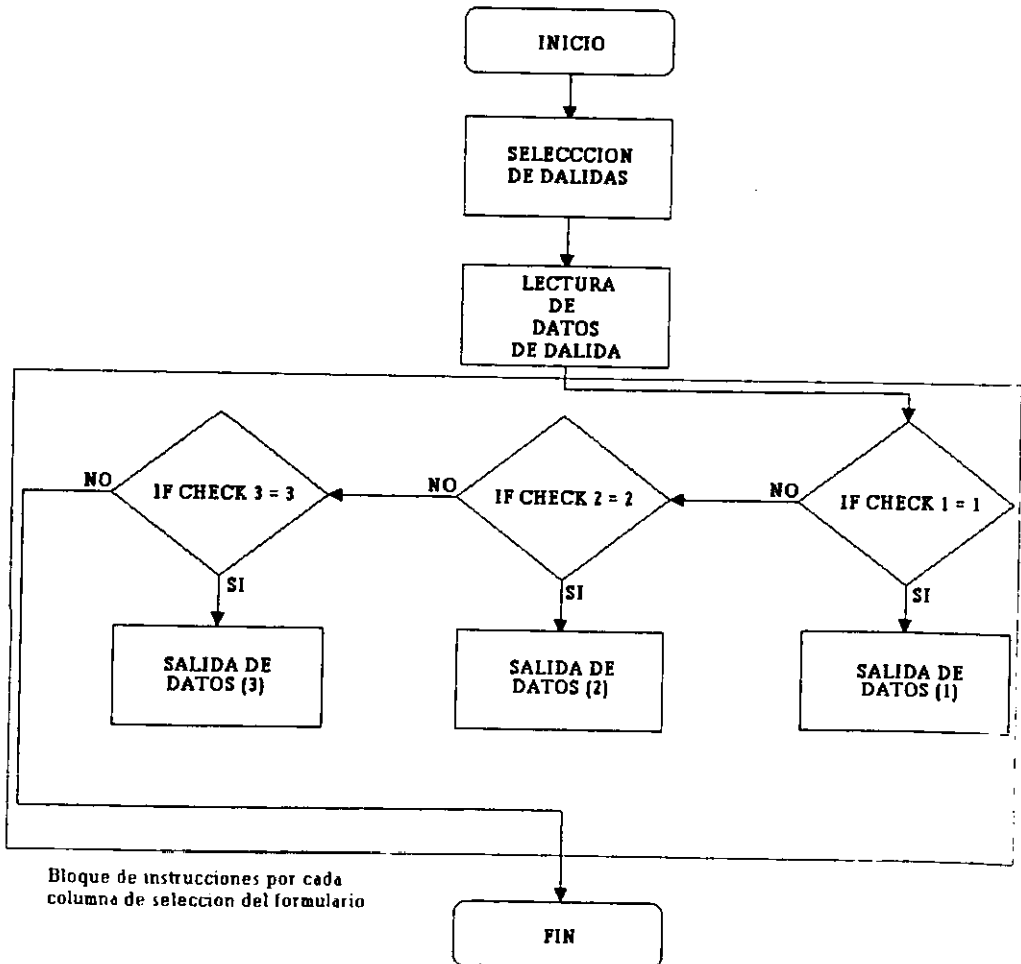
```
    If acel = 2 And num2 = &H9F Then Image1.Picture =  
LoadPicture("C:\...\Dibujo2.jpg")  
    If acel = 3 And num2 = &H9F Then Image1.Picture =  
LoadPicture("C:\...\Dibujo3.jpg")  
    If acel > 3 And num2 <> &HBF And num2 = &H9F Then GoTo 10  
20 End Sub
```

'CON ESTA INSTRUCCION SE LE INDICA QUE TERMINA LA EJECUCION
DEL PROGRAMA.

```
Private Sub salir_Click()  
    End  
End Sub
```

IV.5 CONTROL DE SALIDAS AL PUERTO PARALELO

Una Forma de utilizar las instrucciones de entrada y salida al puerto paralelo, es enviar datos que sean seleccionados a la salida del registro de datos del puerto paralelo hacia el bus de datos, los cuales se envían dentro de la interfaz del puerto paralelo hacia el circuito de salida el cual es el que distribuye o mantiene los datos recibidos por medio del CI 74138 y 74194.



**FIGURA 75: DIAGRAMA DE FLUJO
(PROGRAMA DE SALIDA DE DATOS AL PUERTO PARALELO)**

Para este programa es necesario agregar los siguientes controles dentro de un nuevo proyecto, junto con sus propiedades, como se muestra en la tabla 18.

OBJETO	PROPIEDAD	VALOR
Botón de pulsación	Nombre Caption	Command1 INICIAR
Botón de pulsación	Nombre Caption	reset RESET
Botón de pulsación	Nombre Caption	salir SALIR
Casilla de verificación (CheckBox)	Nombre Caption Value Enable	a1 Check1 0 True
Casilla de verificación	Nombre Caption Value Enable	a2 Check2 0 True
Casilla de verificación	Nombre Caption Value Enable	a3 Check3 0 True
Casilla de verificación	Nombre Caption Value Enable	a4 Check1 0 True
Casilla de verificación	Nombre Caption Value Enable	a5 Check2 0 True
Casilla de verificación	Nombre Caption Value Enable	a6 Check3 0 True
Casilla de verificación	Nombre Caption Value Enable	a7 Check1 0 True

Casilla de verificación	Nombre Caption Value Enable	a8 Check2 0 True
Casilla de verificación	Nombre Caption Value Enable	a9 Check3 0 True
Caja de texto	Nombre Caption	Text1 Text1
Caja de texto	Nombre Caption	Text2 Text2
Caja de texto	Nombre Caption	Text3 Text3
Caja de texto	Nombre Caption	tiempo1 tiempo1
Caja de texto	Nombre Caption	tiempo2 tiempo2
Caja de texto	Nombre Caption	tiempo3 tiempo3
Etiqueta (Label)	Nombre Caption	Label1 tiempo1
Etiqueta	Nombre Caption	Label2 Tiempo2
Etiqueta	Nombre Caption	Label3 Tiempo3
Etiqueta	Nombre Caption	Label4 Control de salidas

**TABLA 18: CONTROLES Y PROPIEDADES
PROGRAMA DE SALIDA DE DATOS POR EL PUERTO PARALELO**

La tabla anterior muestra los controles y sus nombres, así como las propiedades que deben ser seleccionadas de cada control dentro del formulario, el formulario para este proyecto queda como se muestra en la figura 76.

The image shows a screenshot of a Visual Basic form titled "SALIDAS A PUERTO" within a window named "Project1 - Form1 (Form)". The form is set against a dotted grid background. At the top, there is a title bar and a menu bar with "General". On the left side, there is a vertical toolbar with various icons for editing and formatting. The main area of the form contains a section titled "CONTROL DE SALIDAS" which is organized into three columns labeled "TIEMPO 1", "TIEMPO 2", and "TIEMPO 3". Each column contains three check boxes labeled "Check1", "Check2", and "Check3" respectively. Below the check boxes are three text boxes labeled "Text1", "Text2", and "Text3". At the bottom of the form, there are three buttons labeled "INICIAR", "RESET", and "SALIR". Below these buttons are three text boxes labeled "tiempo1", "tiempo2", and "tiempo3".

FIGURA 76: FORMULARIO SALIDAS A PUERTO

Quando se ejecute el anterior formulario es necesario que sean deshabilitados los cuadros check que no sean utilizados cuando un usuario de un click en cualquier cuadro Check, para esto es necesario cambiar la propiedad Enable en el código fuente de la siguiente manera:

`Checkn.Enabled = False`

Si se requiere volver ha habilitar las cajas Check solamente se cambiara el valor a True del cuadro que se requiera:

`Checkn.Enabled = True`

Después de enviar los datos hacia la interfaz del puerto paralelo es necesario mantenerlos durante unos segundos a la salida del registro de datos del puerto paralelo, para ejecutar este tipo de acciones solamente necesitamos las cajas de texto con el nombre tiempo y las siguientes instrucciones:

10

```
HoraInicio = Now  
tiempo1.Text = Format(HoraInicio, "ss")  
HoraParada = (tiempo1.Text + 2)  
If HoraParada > 60 Then GoTo 10  
tiempo2.Text = (HoraParada)
```

20

```
HoraInicio = Now  
HoraInicio = Format(HoraInicio, "ss")  
tiempo2.Text = (HoraParada)  
HoraParada = (tiempo2.Text)  
tiempo3.Text = (HoraInicio)  
If HoraInicio >= HoraParada Then GoTo 30  
GoTo 20
```

30

El código fuente para el formulario es el siguiente:

Option Explicit

Dim Value As Integer

Dim PortAddress As Integer

Private Sub Form_Load()

'PROGRAMA EVALUADO PARA inpout32.dll

Value = 0

'CAMBIE ESTE VALOR PARA EL NUMERO DE PUERTO SOBRE EL QUE SE
ESCRIBIRAN LOS DATOS

'(USUALMENTE LOS PUERTOS PARALELOS SON: &h378, &h278, &h3BC)

PortAddress = &H378

Out PortAddress, &H0

End Sub

'LAS SIGUIENTES INSTRUCCIONES DESABILITAN LAS CAJAS DE SELECCION

'CUANDO UNA SELECCION HA SIDO HECHA

Private Sub a1_Click()

If a1.Value = 1 Then

a2.Enabled = False

a3.Enabled = False

a2.Value = 0

a3.Value = 0

End If

End Sub

Private Sub a2_Click()

If a2.Value = 1 Then

a1.Enabled = False

a3.Enabled = False

a1.Value = 0

a3.Value = 0

End If

End Sub

Private Sub a3_Click()

If a3.Value = 1 Then

a1.Enabled = False

a2.Enabled = False

a1.Value = 0

a2.Value = 0

End If

End Sub

Private Sub a4_Click()

If a4.Value = 1 Then

a5.Enabled = False

a6.Enabled = False

a5.Value = 0

a6.Value = 0

End If

End Sub

```
Private Sub a5_Click()  
If a5.Value = 1 Then  
a4.Enabled = False  
a6.Enabled = False  
a4.Value = 0  
a6.Value = 0  
End If  
End Sub
```

```
Private Sub a6_Click()  
If a6.Value = 1 Then  
a4.Enabled = False  
a5.Enabled = False  
a4.Value = 0  
a5.Value = 0  
End If  
End Sub
```

```
Private Sub a7_Click()  
If a7.Value = 1 Then  
a8.Enabled = False  
a9.Enabled = False  
a8.Value = 0  
a9.Value = 0  
End If  
End Sub
```

```
Private Sub a8_Click()  
If a8.Value = 1 Then  
a7.Enabled = False  
a9.Enabled = False  
a7.Value = 0  
a9.Value = 0  
End If  
End Sub
```

```
Private Sub a9_Click()  
If a9.Value = 1 Then  
a7.Enabled = False  
a8.Enabled = False
```

```
a7.Value = 0
a8.Value = 0
End If
End Sub
```

'CUANDO SE PRESIONA EL BOTON INICIAR, SE EJECUTAN LAS SIGUIENTES INSTRUCCIONES
'ENVIANDO AL PUERTO PARALELO EL NUMERO DE LA SELECCION HECHA EN EL CUADRO
'DE CONTROLES

```
Private Sub Command1_Click()
Dim HoraInicio, HoraParada As Integer
a1.Enabled = False
a2.Enabled = False
a3.Enabled = False
a4.Enabled = False
a5.Enabled = False
a6.Enabled = False
a7.Enabled = False
a8.Enabled = False
a9.Enabled = False
If a1.Value = 1 Then Out PortAddress, &H14
If a2.Value = 1 Then Out PortAddress, &H24
If a3.Value = 1 Then Out PortAddress, &H34
If a1.Value = 1 Or a2.Value = 1 Or a3.Value = 1 Then
Text1.Text = "TRABAJANDO"
10
    HoraInicio = Now
    tiempo1.Text = Format(HoraInicio, "ss")
    HoraParada = (tiempo1.Text + 2)
    If HoraParada > 60 Then GoTo 10
    tiempo2.Text = (HoraParada)
20
    HoraInicio = Now
    HoraInicio = Format(HoraInicio, "ss")
    tiempo2.Text = (HoraParada)
    HoraParada = (tiempo2.Text)
    tiempo3.Text = (HoraInicio)
    If HoraInicio >= HoraParada Then GoTo 30
    GoTo 20
```

30

End If

If a1.Value = 0 And a2.Value = 0 And a3.Value = 0 Then Text1.Text = "NO OPERA"

If a4.Value = 1 Then Out PortAddress, &H14

If a5.Value = 1 Then Out PortAddress, &H24

If a6.Value = 1 Then Out PortAddress, &H34

If a4.Value = 1 Or a5.Value = 1 Or a6.Value = 1 Then

Text2.Text = "TRABAJANDO"

11

 Horalnicio = Now

 tiempo1.Text = Format(Horalnicio, "ss")

 HoraParada = (tiempo1.Text + 2)

 If HoraParada > 60 Then GoTo 11

 tiempo2.Text = (HoraParada)

21

 Horalnicio = Now

 Horalnicio = Format(Horalnicio, "ss")

 tiempo2.Text = (HoraParada)

 HoraParada = (tiempo2.Text)

 tiempo3.Text = (Horalnicio)

 If Horalnicio >= HoraParada Then GoTo 31

 GoTo 21

31

End If

If a4.Value = 0 And a5.Value = 0 And a6.Value = 0 Then Text2.Text = "NO OPERA"

If a7.Value = 1 Then Out PortAddress, &H14

If a8.Value = 1 Then Out PortAddress, &H24

If a9.Value = 1 Then Out PortAddress, &H34

If a7.Value = 1 Or a8.Value = 1 Or a9.Value = 1 Then

Text3.Text = "TRABAJANDO"

12 Horalnicio = Now

 tiempo1.Text = Format(Horalnicio, "ss")

 HoraParada = (tiempo1.Text + 2)

 If HoraParada > 60 Then GoTo 12

 tiempo2.Text = (HoraParada)

22

```
    HoraInicio = Now
    HoraInicio = Format(HoraInicio, "ss")
    tiempo2.Text = (HoraParada)
    HoraParada = (tiempo2.Text)
    tiempo3.Text = (HoraInicio)
    If HoraInicio >= HoraParada Then GoTo 32
    GoTo 22
```

32

End If

```
If a7.Value = 0 And a8.Value = 0 And a9.Value = 0 Then Text3.Text = "NO
OPERA"
```

End Sub

Private Sub reset_Click()

Out PortAddress, &H0

a1.Enabled = True

a2.Enabled = True

a3.Enabled = True

a4.Enabled = True

a5.Enabled = True

a6.Enabled = True

a7.Enabled = True

a8.Enabled = True

a9.Enabled = True

a1.Value = 0

a2.Value = 0

a3.Value = 0

a4.Value = 0

a5.Value = 0

a6.Value = 0

a7.Value = 0

a8.Value = 0

a9.Value = 0

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

```
tiempo1.Text = ""  
tiempo2.Text = ""  
tiempo3.Text = ""  
End Sub
```

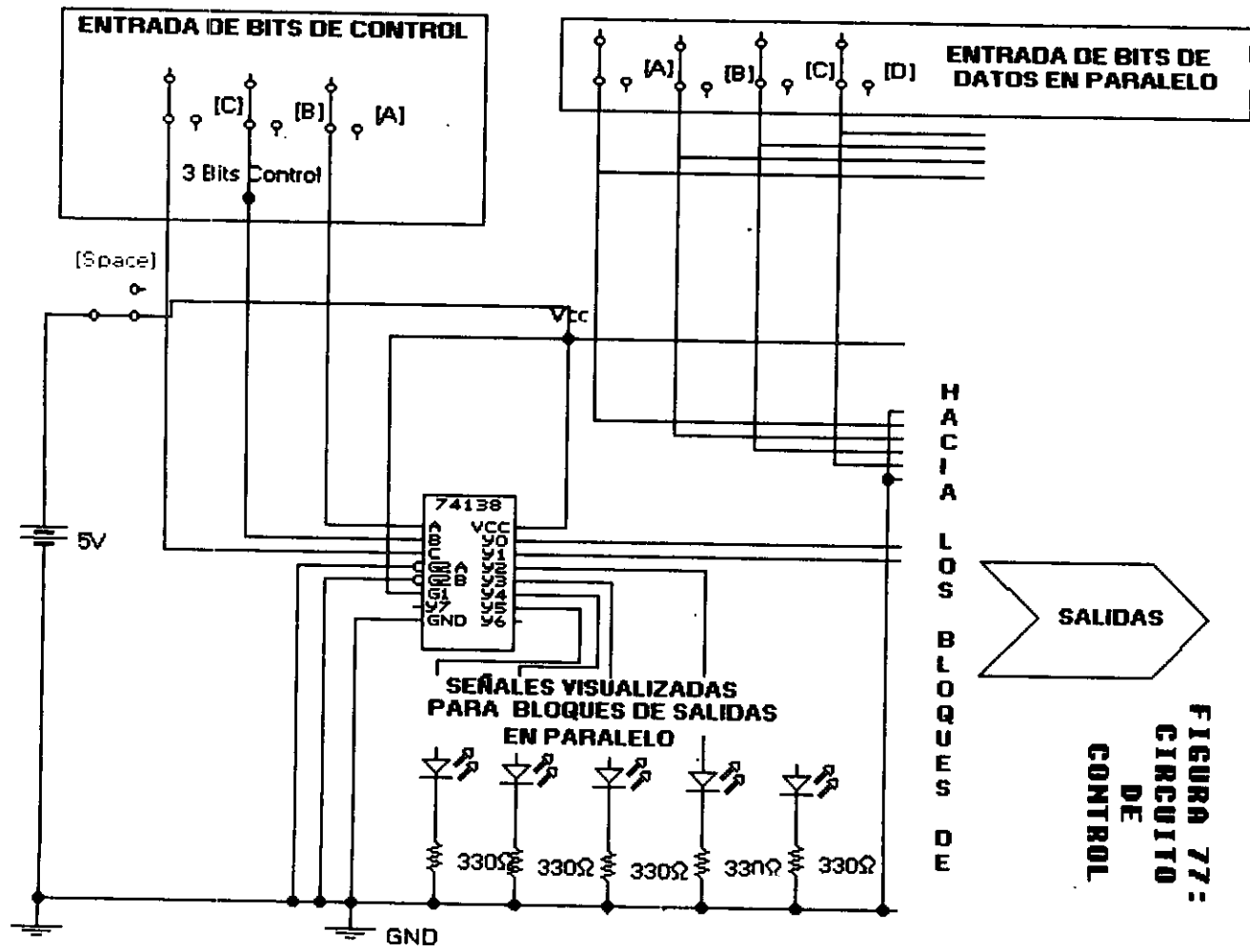
```
Private Sub salir_Click()  
End  
End Sub
```

Después de agregar el código fuente es necesario agregar un modulo con el siguiente código:

```
'Declare Inp and Out for port I/O  
Public Declare Function Inp Lib "inpout32.dll" _  
Alias "Inp32" (ByVal PortAddress As Integer) As Integer  
Public Declare Sub Out Lib "inpout32.dll" _  
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

Para agregar un modulo a nuestro proyecto puede ser cargado en el menú Proyecto, en el Sub-menú agregar nuevo modulo.

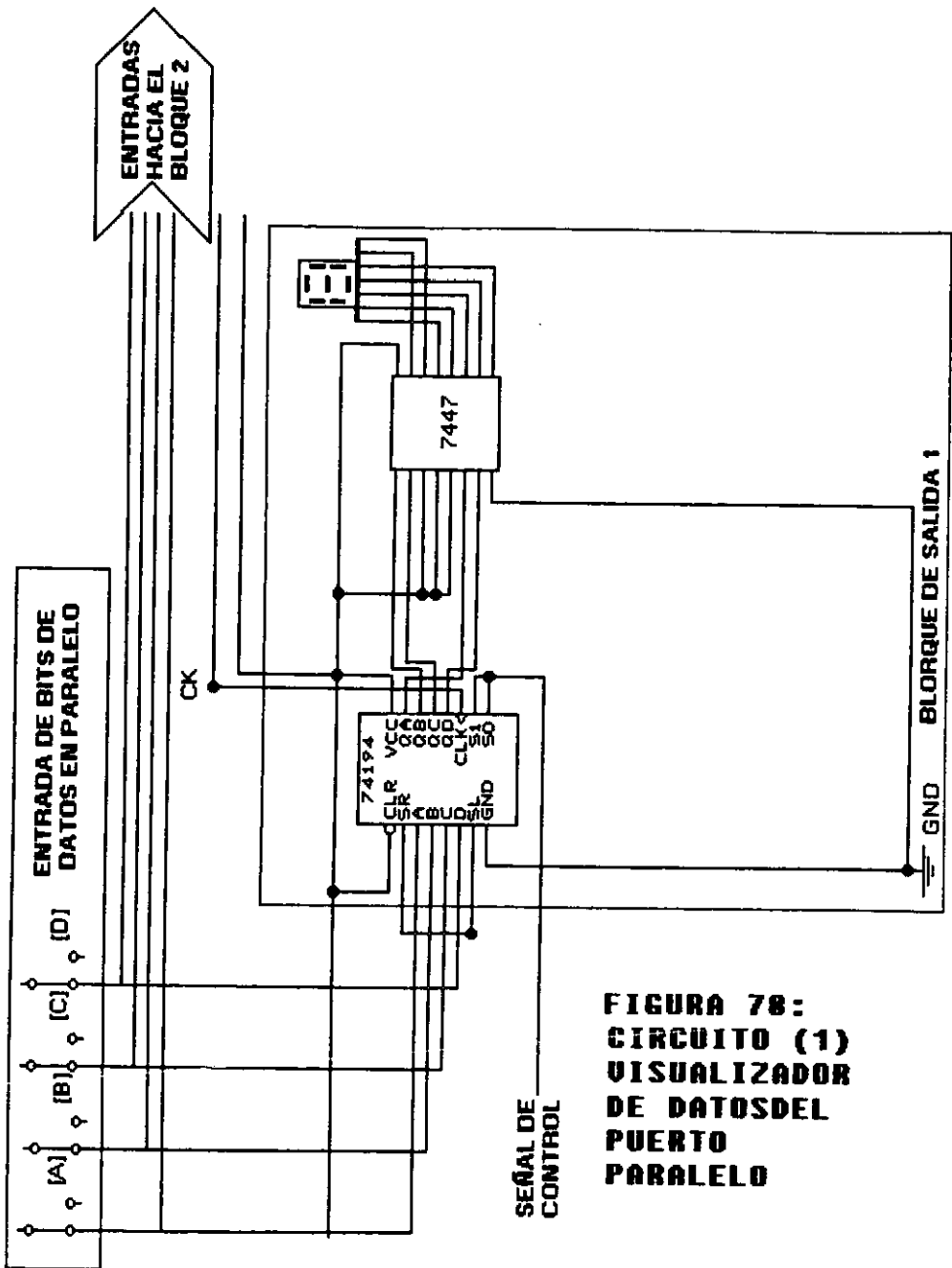
En las siguientes figuras (figuras 77 –80) se muestran los diagramas de los circuitos utilizados para este programa.



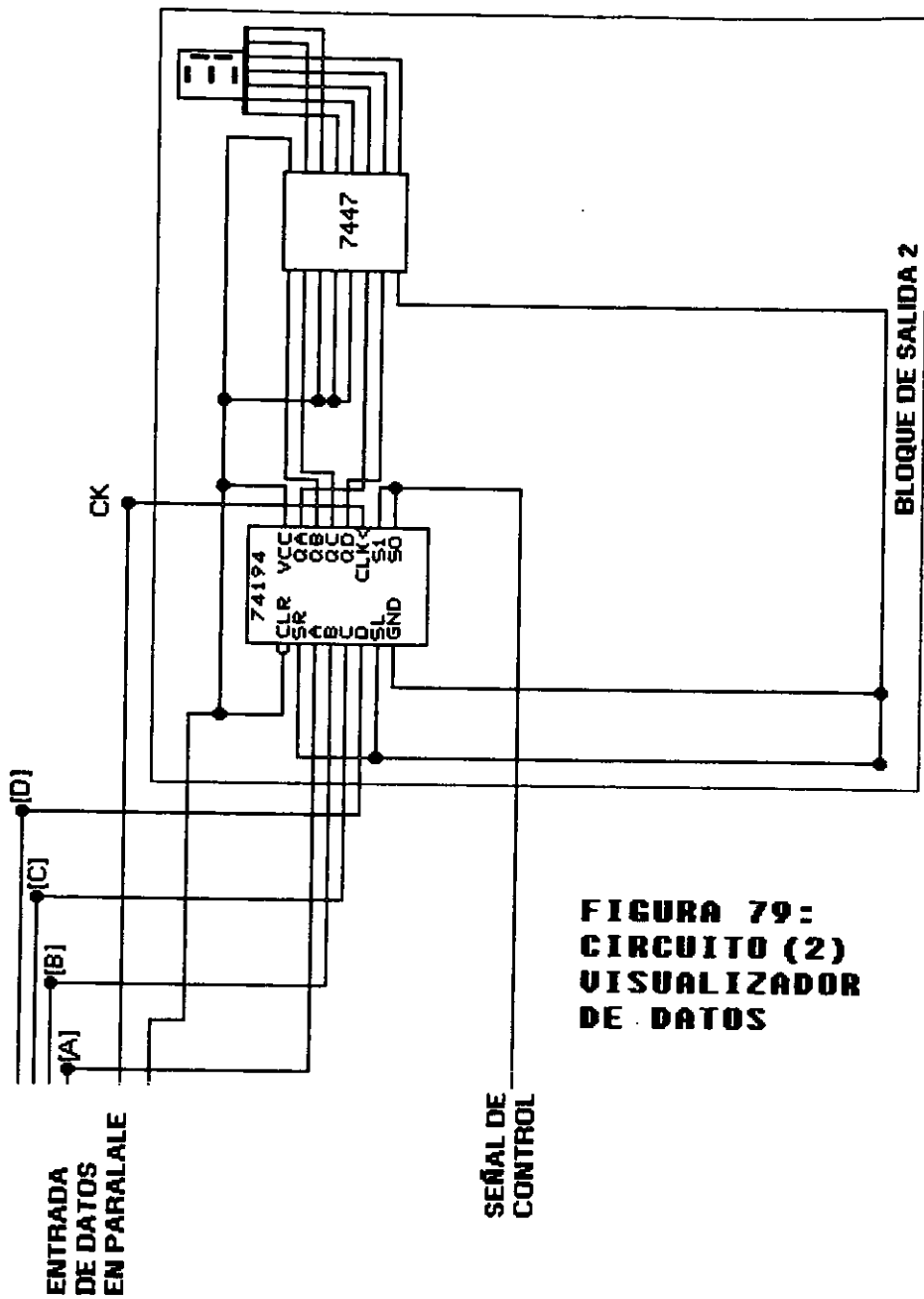
SALIDAS

HACIA LOS BLOQUES DE

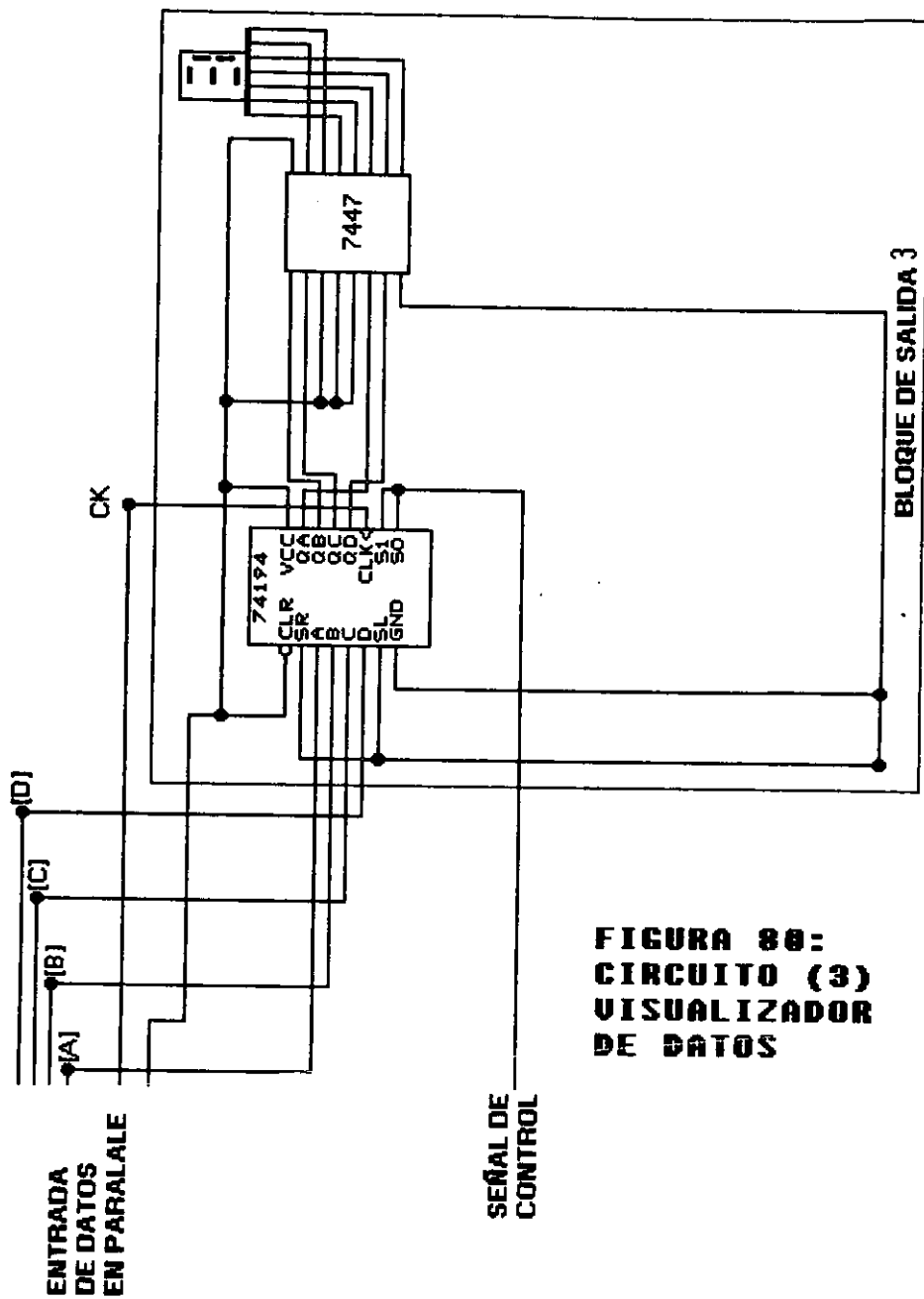
FIGURA 77:
CIRCUITO
DE
CONTROL



**FIGURA 78:
CIRCUITO (1)
VISUALIZADOR
DE DATOS DEL
PUERTO
PARALELO**



**FIGURA 79:
CIRCUITO (2)
VISUALIZADOR
DE DATOS**



**FIGURA 80:
CIRCUITO (3)
VISUALIZADOR
DE DATOS**

CONCLUSIONES

Como se ha mencionado las computadoras son elementos sumamente útiles, pueden comunicar datos al exterior de ellas siempre y cuando exista una manera de programar su sistema interno para obtener los datos deseados, de forma práctica y sencilla se ha mostrado como pueden obtenerse los datos provenientes del puerto paralelo, los cuales se reciben en un circuito de Entrada/Salida, por lo tanto conociendo las características técnicas de los elementos a desarrollar y teniendo una idea en la mente, es fácil utilizar un lenguaje de programación sencillo (Basic, Visual Basic, etc), para obtener las metas aplicadas al proyecto deseado y haciendo que éstos sean útiles y confiables.

En el pasado se programaba a las computadoras en lenguaje ensamblador, con el paso del tiempo se crearon los lenguajes de alto nivel como Basic que facilitaron el uso de instrucciones, hasta nuestros días que se utilizan los programas visuales, con los que es más fácil y practico la creación de controles y aplicaciones de casi cualquier índole de programación.

Durante la investigación de los temas de escritura al puerto paralelo, nos damos cuenta de que en la evolución del lenguaje Basic, existen muchas mejoras, puesto que en la programación Visual Basic, ya no es necesario la creación de controles, subrutinas y cuadros de controles, incluyendo sub-menús, pero existe la problemática de que aun en Visual Basic con su progreso para interfaces gráficas a usuario y con la facilidad de llamados a bibliotecas utilizadas por Windows se ha olvidado de utilizar algunas instrucciones como las instrucciones input y output para el puerto paralelo, los programas para Visual Basic descritos en los capítulos anteriores han sido probados en la versión de Visual Basic 6.

Con los circuitos podemos obtener las salidas del puerto paralelo, utilizando circuitos integrados TTL, puesto que son una familia de circuitos integrados de bajo costo, así como de bajo consumo de voltaje y corriente, por lo que el daño al puerto paralelo, es nulo. La utilización del circuito de salida y entrada de datos al puerto paralelo, realiza una gran función para la obtención de datos, siempre y cuando se utilicen niveles lógicos de voltaje, con una lógica combinacional binaria y/o hexadecimal, para un buen aprovechamiento y uso.

XIII BIBLIOGRAFIA

PROGRAMACION BASIC Y VISUAL BASIC

BASIC ESTRUCTURADO

Peter WORLAND

386 Págs. Rústica, Alfaomega

Circuitos Electrónicos con la PC

Brend Zoller

Septiembre 1999. Editorial: Alfa-Marcombo, 120 Págs.

Computación I

FOURNIER García, María de Lourdes

1999, Editorial Limusa, S.A. de C.V., 185 Págs.

CURSO DE PROGRAMACION QBASIC Y MS-DOS 5

Fco. Javier CEBALLOS

384 págs. Rústica, Coedición: Addison Wesley-Rama

De QBasic a Visual Basic. Curso de Programación

CEBALLOS, F.J.

Septiembre 1996. Rústica y CD ROM, 384 Págs.

Enciclopedia de Microsoft Visual Basic 6

CEBALLOS, F.J.

Noviembre 1999. Rústica y CD ROM, 1072 Págs.

Introducción Al Basic

P. LE BEUX

356 Págs. Rústica, Marcombo Importación

Introducción A Las Computadoras Y Al Procesamiento De Información

2ª. Edición

LONG, Larry.

Diciembre 1991. Prentice Hall Hispanoamericana, S.A., 447 Págs.

Visual Basic

Acceso Rápido

DITTRICH, Stefan

140 págs. Rústica, Coedición: Computec-Marcombo-Data Becker

Visual Basic. Programación cliente / servidor. 2ª edición actualizada a la versión 6.

GONZALEZ, A.

Julio 1999. Rústica y CD ROM, 624 Págs.

Visual Basic. Curso de programación. 2ª edición actualizada a la versión 6.

CEBALLOS, F.J.

Mayo 1999. Rústica y Disquete de 3'5, 528 Págs.

Visual Basic 5. Técnicas y aplicaciones.

EIDOS, Grupo.

Noviembre 1997. Rústica y CD ROM, 468 Págs.

Visual Basic 6 Curso de programación

CEVALLOS, Fco. Javier

528 Págs. Rústica, Coedición: Alfaomega-Rama, julio 1999

Quickbasic 4.5 Para Principiantes

H. TORNSDORF

288 págs. Rústica, Marcombo Importación

ELECTRÓNICA Y CIRCUITOS ELECTRONICOS

20 LECCIONES DE ELECTRONICA DIGITAL

M. BLANES

416 págs. Rústica, Marcombo Importación

20 Montajes Industriales con Circuitos Integrados

Ramírez, José

1998, Ceac, 220 Págs.

Circuiteria Básica En TTL

Arturo PAZ

128 págs. Rústica, Marcombo Importación

Circuitos Electrónicos y sus aplicaciones

GROB, Bernard

Junio 1996, McGraw-Hill, 535 Págs.

Configuración, Actualización Y Mantenimiento.

Software y Hardware de su PC

José A. CARBALLAR

476 págs. Coedición: Alfaomega-Rama

Diseño De Circuitos Y Sistemas Electrónicos

E. J. DEDE y J. ESPI

448 págs. Rústica, Marcombo

Diseño Digital

MANO, Morris

Prentice Hall, 325 Págs.

Electrónica Digital

Gerhard WOLF

328 Págs. Rústica, Marcombo Importación

Electrónica Digital Básica

John A. DEMPSEY

440 págs. Rústica, Alfaomega

El Gran Manual Del PC 2 (2a Edición)

Harald HAHN

444 págs. Rústica, 16 x 24 cm.

ISBN 84-267-1094-8

Marcombo Importación

Gran Libro De Las Impresoras De PC

Ralf OCKENFELDS

606 págs. Pasta dura, Marcombo Importación

Interconexión De Periféricos A Microprocesadores

Serie: MUNDO ELECTRÓNICO

240 págs. Rústica, Marcombo Importación

Principios y Aplicaciones Digitales

MALVINO, Albert P., y Donald P. Leach

Prentice Hall

INTERNET

[HTTP://WWW.MICROSOFT.COM/MSDN](http://WWW.MICROSOFT.COM/MSDN)

<http://www.lvr.com>

Jan Axelson, Lakeview Research (jaxelson@lvr.com).

GLOSARIO DE TERMINOS

Archivo

Conjunto de registros relacionados.

Almacenamiento primario

Area de la memoria en la que todos los programas y datos deben residir antes de que los programas puedan ejecutar o antes de que los datos puedan manipularse.

Almacenamiento secundario

Almacenamiento permanente de datos en disco y cinta magnéticos.

Basic

Lenguaje de programación de propósitos múltiples, muy utilizado en pequeños sistemas computacionales.

En informática, acrónimo de beginners all-purpose symbolic instruction code (código de instrucciones simbólicas de uso general para principiantes). Se trata de un lenguaje de programación de alto nivel desarrollado por los estadounidenses john kemeny y thomas kurtz en el dartmouth college a mediados de la década de 1960.

Bios¹

En informática, acrónimo de basic input/output system (sistema básico de entrada/salida), un conjunto de rutinas que trabajan estrechamente con el hardware de un ordenador o computadora para soportar la transferencia de información entre los elementos del sistema, como la memoria, los discos y el monitor. En los IBM y compatibles, el bios, o ROM bios, está incorporado en la memoria de sólo lectura (ROM) de la máquina. Aunque es fundamental para el funcionamiento, el bios es invisible a los usuarios de los equipos; los programadores sí pueden acceder a él.

¹"BIOS," *Enciclopedia Microsoft® Encarta® 2000*. © 1993-1999 Microsoft Corporation. Reservados todos los derechos.

Bit

Binary digit (dígito binario), que adquiere el valor 1 o 0 en el sistema numérico binario. En el procesamiento y almacenamiento informático un bit es la unidad de información más pequeña manipulada por el ordenador, físicamente es una celda de memoria de computadora; de un punto grabado magnéticamente en la superficie de un disco.

Bloque

Grupo de datos que se lee o se escribe en un dispositivo de E/S durante una operación.

Byte

Unidad de información que consta de 8 bits. Grupo de bits adyacentes que representan un carácter.

Carácter

Unidad de dato alfanumérico.

Cargar

Transferir programas o datos del almacenamiento secundario al primario

Ciclo

Secuencia en la que las instrucciones de un programa se ejecutan repetidamente hasta que se llega a una condición dada.

Circuito integrado

Pequeño circuito electrónico utilizado para realizar una función electrónica específica, como la amplificación. Se combina por lo general con otros componentes para formar un sistema más complejo y se fabrica mediante la difusión de impurezas en silicio monocristalino, que sirve como material semiconductor, o mediante la soldadura del silicio con un haz de flujo de electrones. Varios cientos de circuitos integrados idénticos se fabrican a la vez sobre una oblea de pocos centímetros de diámetro. Para interconectarlos con otros circuitos o componentes, los chips se montan en cápsulas que contienen conductores eléctricos externos. De esta forma se facilita su inserción en placas

En electrónica de consumo, los circuitos integrados han hecho posible el desarrollo de muchos nuevos productos, como computadoras y calculadoras personales, relojes digitales y videojuegos. Se han utilizado también para mejorar y rebajar el costo de muchos productos existentes.

Código

1. Reglas utilizadas para traducir la configuración de un bit a carácter alfanumérico
2. Proceso de compilación de instrucciones a una forma legible para la computadora
3. El mismo programa de computación.

Comando

Elemento que permite al usuario dirigir el curso de acción de la computadora.

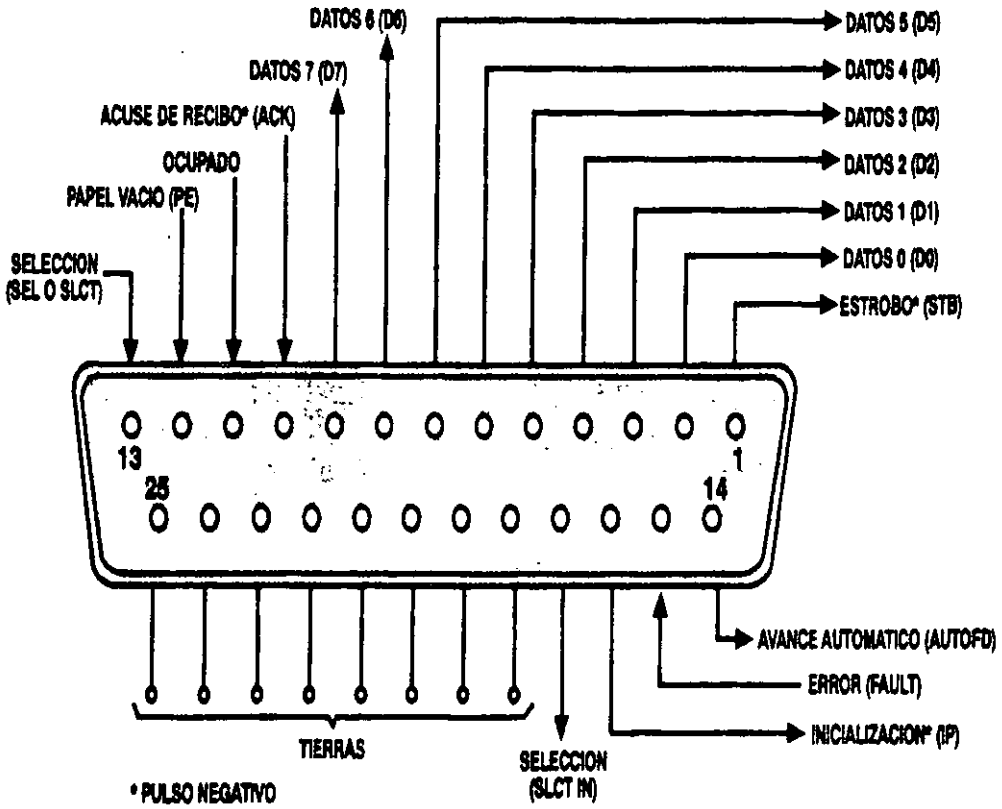
Comunicación de datos²

Intercambio de información entre computadoras. Sin apenas excepción alguna, los ordenadores modernos se basan en el concepto de dígitos binarios, denominados bits, que sólo pueden adoptar los valores 0 o 1. Todos los datos almacenados y procesados por una computadora tienen la forma de bits, por lo que la transferencia de datos entre máquinas implica enviar bits de un lado a otro. En principio resulta muy sencillo, ya que la señal está presente o ausente; por ejemplo, no existen los matices de tono y volumen que se aprecian en la comunicación de voz. En la práctica, sin embargo, las comunicaciones de datos son más complejas de lo que parecen. Una secuencia de dígitos enviados desde un ordenador debe volverse a transformar en una información significativa con independencia del retardo, ruido y corrupción que sufra en el trayecto.

Conector de bus de datos

Cualquiera de los diversos tipos de conectores utilizados para facilitar la entrada y salida en serie y en paralelo. El número que aparece detrás de las iniciales db, acrónimo de data bus (bus de datos), indica el número de líneas (cables) dentro del conector. Los conectores de bus de datos más comunes son el db-9, db-15, db-19, db-25, db-37 y db-50.

²"Comunicación de datos." *Enciclopedia Microsoft®. Encarta® 2000*. © 1993-1999 Microsoft Corporation. Reservados todos los derechos.



Convertidores digital y analógico


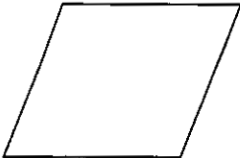
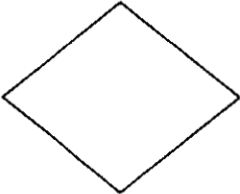
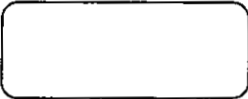
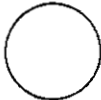
Son dispositivos que convierten la información de una forma a otra (Digital y analógica), que sirven para codificar valores numéricos en voltajes y viceversa.

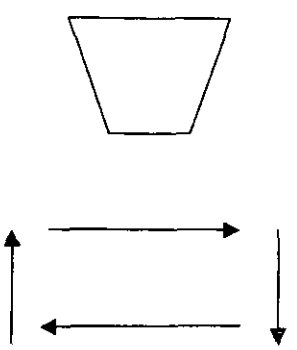
Depurar

Eliminar errores ocultos en un programa o sistema.

Diagrama de flujo

Diagrama que muestra el flujo de los datos, de información y de ejecución, que por medio de símbolos especializados conectados por líneas de flujo, representan la lógica de un sistema o programa.

Símbolo	Uso
	Proceso de la computadora
	Entrada / salida
	decisión
	Punto donde inicia o termina un procedimiento
	conector

Símbolo	Uso
	<div data-bbox="730 335 931 378" style="border: 1px solid black; padding: 2px; margin-bottom: 10px;">Proceso manual</div> <div data-bbox="730 505 931 562" style="border: 1px solid black; padding: 2px;">Líneas de flujo</div>

Digital

En contraposición a 'analógico' (continuo), forma de representar la información con valores numéricos discretos.

Dispositivos de entrada salida

Para introducir datos se emplean dispositivos muy diversos. La mayoría de las computadoras personales incluyen un teclado. El mouse, la bola de seguimiento o trackball y el joystick son otros dispositivos de entrada que permiten señalar, seleccionar y mover objetos en un monitor. Puede introducirse texto manuscrito deslizando por la pantalla un lápiz óptico, cuyos sensores convierten los movimientos del usuario en datos electrónicos. Las pantallas táctiles, en las que unos sensores de luz infrarroja localizan los dedos del usuario, se emplean en lugares en los que un teclado resultaría poco adecuado. El reconocimiento de voz es muy empleado en algunas aplicaciones, pero estos dispositivos de entrada son todavía imperfectos y sólo responden a un pequeño vocabulario de instrucciones.

Los dispositivos de salida más habituales son las impresoras y los monitores en color. La salida de audio también es corriente, así como las complejas conexiones con sintetizadores que producen una amplia gama de sonidos musicales.

Entrada

Datos que procesara un sistema computacional.

Interfaz

Punto en el que se establece una conexión entre dos elementos, que les permite trabajar juntos. En el campo de la informática se distinguen diversos tipos de interfaces que actúan a diversos niveles, desde las interfaces claramente visibles, que permiten a las personas comunicarse con los programas, hasta las imprescindibles interfaces hardware, a menudo invisibles, que conectan entre sí los dispositivos y componentes dentro de los ordenadores o computadoras. Las interfaces de usuario cuentan con el diseño gráfico, los comandos, mensajes y otros elementos que permiten a un usuario comunicarse con un programa.

Interfaz de instrucciones e interfaz grafica

El diálogo entre el usuario y la máquina suele realizarse a través de una interfaz de línea de comandos o de una interfaz gráfica de usuario (gui, siglas en inglés). Las interfaces de línea de comandos exigen que se introduzcan instrucciones breves mediante un teclado. Las gui emplean ventanas para organizar archivos y aplicaciones con iconos y menús que presentan listas de instrucciones. El usuario manipula directamente estos objetos visuales en el monitor señalándolos, seleccionándolos y arrastrándolos o moviéndolos con un mouse o una bola apuntadora.

El uso de las gui es más sencillo que el de las interfaces de línea de comandos. Sin embargo, la introducción de instrucciones con una gui es más lenta, por lo que las gui suelen tener la opción de emplear un sistema equivalente al de línea de instrucciones como alternativa rápida para los usuarios más expertos.

Interfaz de usuario

Conjunto de componentes empleados por los usuarios para comunicarse con las computadoras. El usuario dirige el funcionamiento de la máquina mediante instrucciones, denominadas genéricamente entradas. Las entradas se introducen mediante diversos dispositivos, por ejemplo un teclado, y se convierten en señales electrónicas que pueden ser procesadas por la computadora. Estas señales se transmiten a través de circuitos conocidos como bus, y son coordinadas y controladas por la unidad de proceso central y por un soporte lógico conocido como sistema operativo. Una vez que la PC ha ejecutado las instrucciones indicadas por el usuario, puede comunicar los resultados mediante señales electrónicas, o salidas, que se transmiten por el bus a uno o más dispositivos de salida, por ejemplo una impresora o un monitor.

Interfaz gráfica de usuario

En informática, tipo de visualización que permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el ratón.

Hardware

Todos los dispositivos físicos que integran a un sistema computacional

Hexadecimal

Sistema numérico cuya base es 16, se utiliza en los procesos de información y facilita al programador la condensación de resultados binarios y los hace más legibles.

Instrucción

Frase u oración de los lenguajes de programación que identifica una operación específica de computo por realizarse.

Led

Un diodo es un componente electrónico a través del cual la corriente pasa en un solo sentido. Los diodos emisores de luz (led, acrónimo de light-emitting diode) son semiconductores que generan luz al pasar una corriente a través de ellos. Se emplean en numerosos dispositivos comunes como en los displays que son utilizados en las calculadoras.

Lenguaje de programación

Lenguaje con el cual un programador comunica instrucciones a una computadora

Menú

Pantalla de una estación de trabajo con una lista de elecciones de procesamiento, a partir de la cual un usuario final puede seleccionar la que necesite.

Paralelo

En informática, se refiere al proceso o transferencia de datos de forma simultánea, en oposición al proceso o transferencia en serie. En la transferencia de datos en paralelo, la información se envía simultáneamente en grupos. Por ejemplo de ocho bits de un byte de datos sean transmitidos a la vez, a través de ocho hilos independientes de un cable.

Polarizar

En la mayoría de los casos, las propiedades de un dieléctrico son producto de la polarización de la sustancia. Al colocar un dieléctrico en un campo eléctrico, los electrones y protones que constituyen sus átomos se reorientarán a sí mismos.

Programa fuente

Código de un programa original.

Programa objeto

Programa a nivel de máquina que es resultado de la compilación de un programa fuente.

Programación orientada a objetos

En informática, un estilo de programación en el que un programa se contempla como un conjunto de objetos limitados, que a su vez, son colecciones independientes de estructuras de datos y rutinas que interactúan con otros objetos. Una clase define las estructuras de datos y rutinas de un objeto.

Puerto (informática)

Punto de accesos a un sistema que permite la comunicación entre la computadora y los periféricos.

En hardware de ordenadores o computadoras, el lugar donde se intercambian datos con otro dispositivo. Los microprocesadores disponen de puertos para enviar y recibir bits de datos. Estos puertos se utilizan generalmente como direcciones de memoria con dedicación exclusiva.

Registro

Conjunto de campos relacionados.

Pequeña área de almacenamiento de alta velocidad en la que se almacenan datos que pertenecen a la ejecución de una instrucción en particular.

Tiempo de acceso

Intervalo de tiempo que abarca el momento en que una computadora solicita la transferencia de datos de un dispositivo de almacenamiento secundario hasta el instante en que la operación se realiza.

Transmisión de datos

En informática, transmisión de información de un lugar a otro, tanto dentro de un ordenador o computadora, como de un dispositivo externo. La velocidad de transmisión de datos se denomina también coeficiente de transmisión o velocidad de transferencia de datos y suele medirse en bits por segundo (bps).

A cada dispositivo que necesita comunicarse con el microprocesador, se le asigna una dirección I/O, es decir un rango de direcciones.

Esta dirección puede no puede ser utilizada por ningún otro dispositivo.

Dirección I/O que se acostumbran en la PC:

RANGO HEXADECIMAL	USUARIO
00-0F	Controlador DMA 8327 #1
20-21	Controlador 8259A #1 de línea de interrupción programable
40-43	Cronómetro 8253
60-63	Controlador de Periféricos 8255 (XT únicamente)
60-64	Controlador de teclado (AT únicamente)
70-71	Dirección de acceso a RAM de configuraciones (AT únicamente)
80-8F	Registros de página DMA
A0-A1	Controlador programable de línea de Interrupción #2 8259 (AT)
A0-AF	Registro de mascara NMI (XT únicamente)
C0-DF	Controlador #2 para DMA 8237 (AT únicamente)
F0-FF	Coprocesador matemático (AT únicamente)
1F0-1F8	Controlador de disco duro (AT únicamente)
200-20F	Controlador de Joystick
210-217	Cnasis de expansión
238-23B	Mouse de Bus
23C-23F	Mouse de Bus alternativo
258	Tarjeta de memoria expandida (LIM)
278-27F	LPT2
2B0-2DF	EGA
2E0-2E7	GPIB (AT únicamente)
2E8-2EF	Puerto serie COM 4
2F8-2FF	Puerto serie COM 2
300-30F	Tarjeta LAN Ethernet
300-31F	Tarjeta prototipo

320-32F

330-33F

378-37F

380-38F

3A0-3AF

3B0-3BF

3BC-3BF

3D0-3DF

3E8-3EF

3F0-3F7

3F8-3FF

Controlador de disco duro
(XT únicamente)

Controlador de caja Bernoulli

Puerto de impresora LPT1

Tarjeta SDLC

Tarjeta BSC

Adaptador monocromático

LPT3

Adaptador gráfico en color

Puerto serie COM 3

Controlador de Diskette

Puerto serie COM 1.

APENDICE A

INTERFACES

INTERFAZ ST-506/412

- Interfaz entre controlador y periférico
- El separador de datos incluido en el controlador
- Usado en mover el motor de discos paso a paso (pista a pista)
- Permite controlar 4 discos y 16 cabezas
- No usa el servo
- El modelo 412 usa un buffer de señales para ajustarse a cualquier velocidad de motor
- Es un sistema barato
- Consta de 3 cables: J1, J2, J3
- J1 es de señales de control, J2 de datos y J3 de tensión
- J1 usa señales de configuración simple (0 o 5V) y J2 de configuración diferencial (diferencia entre alta y baja)
- Transmisión serie

INTERFAZ ESDI

- El separador de datos está incluido en el periférico lo que lo hace más rápido y con mayor rendimiento
- El controlador envía señales a través de la interfaz hacia dirección de periférico para mover el motor pista a pista (el periférico esta contenida en una dirección de memoria)
- El controlador puede interrogar a través de la interfaz al periférico acerca del número de cilindros, cabezas y número de bytes por pista
- Utilizado en pequeñas unidades de disco

INTERFAZ SMD

- Utilizado en grandes unidades de disco
- Estándar ANSI
- El separador de datos está en el periférico
- Tiene cables largos para adaptarse a grandes sistemas

INTERFAZ SCSI

- Interfaz (o bus) paralelo
- Interfaz entre la unidad básica y el controlador del periférico
- Es utilizable por casi todas las unidades básicas contando sólo con un pequeño adaptador
- Es estándar ANSI
- Soporta muchos tipos de periféricos
- Se puede usar de dos maneras:
 1. Como interfaz inteligente para conectar unidad básica y un solo periférico
 2. Como interfaz entre varias unidades básicas y varios controladores de periféricos
 - Se le pueden conectar 8 dispositivos (entre unidades y controladores)
 - Al menos debe haber una unidad básica y un controlador
 - Se utilizan 18 señales: 9 de control y 9 de datos (8 bits de datos y 1 bit de paridad)
 - Permite transmisión en modo simple (0 o 5V) que es más barato pero los cables son más cortos, y modo diferencial (diferencia entre alta y baja) que es más caro pero permite cables más largos
 - En los extremos del bus o interfaz debe haber un terminador que impida el rebote de las señales.

Wide SCSI

- Semejante a SCSI pero permite trabajar con 36 líneas de datos (32 bits de datos y 4 de paridad)

INTERFAZ Centronics

- Tipo paralelo
- Se suele emplear para dispositivos periféricos lentos (impresoras y plotters)
- Tiene un hardware muy sencillo
- El control recae en un programa de software de control debido a la simplicidad del hardware, por lo que es muy lento
- Bus de 17 líneas utilizables: 4 líneas de datos, 5 líneas de estado y 8 líneas de datos
- Líneas de datos unidireccionales que sólo permiten enviar datos al periférico y no recibir hacia el periférico
- Debido a su bajo costo es muy utilizado en PC's
- No permite distancias elevadas entre la unidad central y el periférico

- Debido a su utilización (al ocupar muy poco espacio) se ha modernizado y se ha convertido en estándar IEEE (se ha hecho bidireccional y más rápido)

VENTAJAS: Barato, Paralelo y Muy extendido

DESVENTAJAS: De corta distancia, Unidireccional, Punto a punto y Lento

INTERFAZ IEEE-1284

- Tipo paralelo
- Es una modernización del Centronics
- Permite flujo bidireccional de datos y es más rápido que el Centronics
- Usa 5 modos diferentes de utilización

Modo compatible Centronics

- Es barato y lento debido a que se usa un control por software
- Debido a su lentitud y ser unidireccional, se usa en impresoras de agujas y láser antiguas
- No en la norma IEEE se le dio una mejora que fue añadirle una memoria FIFO para guardar datos gestionables por hardware en vez de software y así ser más rápido (Centronics rápido o puerto paralelo con modo FIFO)

Modo Nibble

- Permite la recepción de datos y nos sólo el envío (bidireccional)
- Usa 4 líneas de estado para recibir nibbles (4 bits) por lo que para completar 1 byte necesita dos lecturas
- Las líneas de datos siguen siendo sólo unidireccionales (hacia el periférico)
- Debido a que no requiere circuitería especial, es compatible totalmente con cualquier Centronics

Modo Byte

- Consiste en modificar las líneas de datos para que se puedan usar bidireccionalmente
- Al no tener que partir el byte en dos nibbles, es mucho más rápido que en modo nibble

Modo EPP

- Permite el uso de las líneas de datos para datos y para direcciones de puerto, números de canal o comandos
- El control se efectúa por el propio hardware por lo que es más rápido que Centronics y Nibble
- La velocidad de transferencia es ajustable respecto al dispositivo controlado
- Permite controlar varios dispositivos mediante la técnica del interbloqueo
- Usa registros para poder direccionar incluso 32 bits

Modo ECP

- Usado sobre todo para impresoras y escáners
- Como los escáners suelen usar datos redundantes, permite la compresión de los datos
- Usa una cola FIFO y un canal DMA
- Incorpora direccionamiento lógico por lo que es posible usar varios periféricos en tiempo compartido y no sólo uno cada vez como en los otros modos
- Usa varios registros adicionales

Negociación de modo

- Al usarse una interfaz IEEE-1284, se debe interrogar al periférico sobre si admite todos los modos o sólo algunos. Esto lo hace mediante una línea de control
- Los modos Centronics y Nibble son admitidos en cualquier periférico tipo Centronics, por lo que no se requiere respuesta por parte del periférico. Las demás respuestas las recibe la interfaz por una línea de estado. Los modos Centronics y Nibble no envían respuesta.

DISPOSITIVOS DE ENTRADA

Teclados
Codificación
Software de control
El ratón

TECLADOS

Consisten en un conjunto de teclas que han de ser pulsadas por el dedo, y además algunos LED luminosos. Internamente es un circuito electrónico conectado al procesador. No son muy sofisticados ya que deben ser manejados por el hombre directamente y son la evolución de las máquinas de escribir.

El teclado consta básicamente del pulsador, que es un elemento electromecánico que al ser pulsado, produce un impulso eléctrico que es transmitido al procesador.

CODIFICACION

Cuando el número de teclas es grande, es necesario codificarlas. Hay diferentes formas de conectar las teclas al procesador:

Conexión a codificador

Se realiza cuando el número de teclas es pequeño y consiste en emplear un

codificador al que van a parar las señales de las teclas, este codificador da su salida que es luego tratada y enviada al procesador.

Conexión matricial

Se emplea cuando el número de teclas es más elevado y consiste en una matriz con dos codificadores. Además, este sistema emplea una EPROM en la que están codificados todos los códigos que pueden ser señalados. Esto permite la utilización de teclas de selección como ALT, CTRL y SHIFT. Debido al uso de la EPROM, la dirección señalada en ella debe mantenerse un cierto tiempo para poder ser leída.

SOFTWARE DE CONTROL

Para el control del teclado es preciso detectar cuando se pulsa más de una tecla a la vez y cuando se suelta una u otra, etc... En los sistemas conectados a hosts (controla el SO), el SO hace que se empleen las interrupciones cada vez que se pulse una tecla o se suelte y así poder actuar en consecuencia. Esto hace que el software controle al teclado.

- * Pulsación de dos teclas: cuando se pulsan dos teclas a la vez, sólo es utilizada la primera y todas las demás son rechazadas hasta que se libere a la primera. Sólo en el caso de que la primera sea soltada antes que la segunda, se detectará la Segunda.
- * Bloqueo total: el sistema se bloquea mientras haya más de una tecla pulsada.
- * Pulsación de n teclas: siempre se generará el código de la última tecla pulsada aunque la anterior siga también pulsada.

EL RATON

Los ratones más antiguos eran electromecánicos, luego salieron los opto-mecánicos y luego los ópticos.

Los ratones opto-mecánicos consisten en una bola de caucho o similar que tienen 2 ejes perpendiculares entre sí y otro oblicuo a ambos que hace que la bola no se separe de los 2 ejes. Además emplea un sistema de emisor-receptor óptico (láser de semiconductor o de gas) para cada eje. Usa una ruleta con rendijas entre el emisor y el receptor que es la que permite la detección del movimiento. Las rendijas son asimétricas para así poder detectar el sentido del movimiento de la bola.

Los ratones ópticos emiten luz a la alfombrilla y esta luz es reflejada y detectada por el ratón. Como la alfombrilla tiene zonas reflectantes y no reflectantes, la detección permite conocer el movimiento del ratón.

También hay otros tipos de ratones como ratones a distancia (son más cómodos pero deben permitir una comunicación visual con el procesador), ratón lápiz o trackball (es

un ratón boca arriba de forma que lo que giramos con el dedo es la bola. Debido al poco espacio que requieren, son muy usados en ordenadores portátiles).

PERIFERICOS DE INSTRUMENTACION Y CONTROL ADQUISICION DE DATOS

TRANSDUCTORES Y SEÑALES DE CAMPO

Los transductores son dispositivos que convierten una entrada de una cierta naturaleza física en una salida de otra naturaleza física (generalmente eléctrica).

La señal de salida no suele estar adaptada a su utilización por computadores, por lo que suele adaptarse.

Desde el punto de vista del tipo de señal de salida de un transductor, estos se dividen en:

1. Transductores de resistencia variable

Se suelen utilizar con potenciómetros y hay que tener en cuenta que al ser simplemente una resistencia, no suministran solos una señal de salida por lo que se les debe suministrar una tensión de entrada que por efectos de calentamiento puede salir distorsionada.

2. Transductores de reactancia variable

Suelen ser transductores capacitivos y miden pequeños desplazamientos, niveles de líquidos ya que son muy precisos y estables. Ya que su medida es en c.a., para su utilización en computadores se los debe pasar a c.c.

3. Transductores generadores de carga

Son generadores de corriente y se usan en celdas fotoeléctricas.

4. Transductores generadores de tensión

Son muy utilizados porque se pueden conectar directamente al computador (salvo una regulación de tensión) aunque su señal a distancia se puede ver afectada por ruidos

5. Transductores generadores de corriente

Se usan mucho debido a que estas señales no se ven influenciadas por ruido en la transmisión a alarga distancia.

6. Transductores digitales

Son muy usados en alarmas, interruptores, etc... y su utilización en computadores no requiere a lo sumo más que una adaptación de la tensión.

SISTEMAS DE ADQUISICION DE DATOS

Los sistemas digitales de control son los más utilizados debido a que no se ven influidos por el ruido, son baratos, son más precisos pero en cambio son más lentos que los analógicos.

Como las señales físicas suelen ser analógicas, es necesario que los sistemas de adquisición de datos las tomen y las conviertan en digitales para que sean disponibles para un computador.

La señal es tomada del mundo físico por el transductor y la pasa a tensión eléctrica. Luego es amplificada al nivel de 0 a 10 voltios por el amplificador. Después es filtrada para quitarle los ruidos o altas frecuencias. Después pasa a un multiplexor que hace posible que varias señales diferentes puedan ser utilizadas por el sistema. La salida del multiplexor va a un circuito de muestreo y retención que se encarga de tomar la señal y mantenerla el tiempo suficiente para que el conversor analógico/digital la pase a digital. Por último, el programador secuenciado (controlado por el computador) se encarga de controlar al multiplexor, al muestreador/retenedor y al conversor A/D.

Debido a que es necesario hacer un muestreo a la señal más rápido que ella misma para poder obtener su forma, es necesario que el muestreador/retenedor memorice sus lecturas hasta poder obtener una aproximación a la onda original para pasarla al conversor A/D.

La frecuencia de muestreo debe de ser al menos el doble que la frecuencia de la onda original.

Los circuitos de muestreo y retención se usan para mantener la señal el suficiente tiempo para que el conversor la pueda convertir. Constan generalmente de un interruptor y un condensador.

MODOS DE CONEXION DE UN SISTEMA DE ADQUISICION DE DATOS A UN ORDENADOR

1. Adquisición del valor más reciente: el convertidor A/D funciona continuamente y actualizando, conforme su velocidad de proceso se lo permite, su salida en un registro que será leído por el computador cuando lo necesite.

2. Comienza y espera: el computador ordena al convertidor que realice el proceso de conversión y espera hasta que el convertidor haya terminado y colocado su salida en un registro que será leído en ese momento por el computador. El inconveniente es que

el computador pierde tiempo en esperar.

3.Utilizar interrupciones: el computador inicia el proceso de conversión pero al disponer de interrupciones, puede seguir haciendo otras cosas hasta ser interrumpido por el conversor.

4.Utilizar acceso directo a memoria (DMA): se trata de transferir datos directamente del conversor a la memoria sin intervención directa del procesador.

LA INTERFAZ SERIE RS232

Se suele usar para enviar señales a distancia ya que muchas veces el transductor esta separado del computador y el RS232 se encarga de convertir la señal analógica a digital y enviarla en serie al computador.

Actualmente se usa para muchas cosas como conexión con módem o con impresora.

Puede usarse el RS232 como un sistema completo de adquisición de datos (transductor, amplificador, filtro activo, multiplexor analógico, muestreador/retenedor, programador/secuenciador y conversor A/D).

La conexión del RS232 al computador se hace usando un adaptador asincrono de comunicaciones.

Las ventajas son: está muy extendido, permite acercar la toma de datos de la interfaz y emitirlos vía interfaz al computador.

Las desventajas son: la transmisión es lenta al ser en serie.

El RS422 permite una mayor distancia de transmisión, tolera más el ruido (debido al desdoblamiento de la señal en ella y su complemento que se envían juntas y son reconstruidas por el receptor) y es más veloz la transmisión.

El RS423 permite enviar señales a muchos receptores a la vez.

LA INTERFAZ GPIB (IEEE-488)

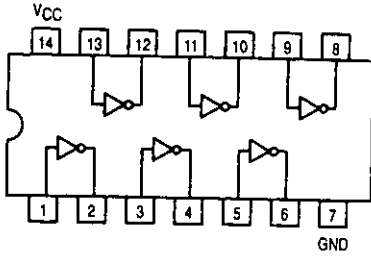
Es diseñado para que sea múltiplemente compatible. Sus características son:

- La transmisión es digital.
- El número de equipos a interconectar es 15.
- Las longitudes de transmisión no pueden exceder 20 metros o 2 metros por equipo.
- La velocidad de transmisión no puede exceder 1 Mb/seg
- Consta de 8 líneas de control, 8 líneas de datos bidireccionales y 8 líneas de masa.
- Cualquier unidad puede ser emisora, receptora o controladora (sólo puede haber un controlador en cada instante).
- La velocidad de transmisión se adapta al equipo más lento.
- La transmisión es en paralelo de byte en byte.

APENDICE B

TTL

HEX INVERTER



SN54/74LS04

HEX INVERTER
LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 632-08



N SUFFIX
PLASTIC
CASE 646-06



D SUFFIX
SOIC
CASE 751A-02

ORDERING INFORMATION

SN54LSXXJ	Ceramic
SN74LSXXN	Plastic
SN74LSXXD	SOIC

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
VCC	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
TA	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54			4.0	mA
		74			8.0	

SN54/74LS04

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions	
		Min	Typ	Max			
V_{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74		0.8			
V_{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	$V_{CC} = \text{MIN.}$ $I_{IN} = -18 \text{ mA}$	
V_{OH}	Output HIGH Voltage	54	2.5	3.5	V	$V_{CC} = \text{MIN.}$ $I_{OH} = \text{MAX.}$ $V_{IN} = V_{IH}$ or V_{IL} per Truth Table	
		74	2.7	3.5	V		
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	$I_{OL} = 4.0 \text{ mA}$ $V_{CC} = V_{CC} \text{ MIN.}$ $V_{IN} = V_{IL}$ or V_{IH} per Truth Table
		74		0.35	0.5	V	
I_{IH}	Input HIGH Current			20	μA	$V_{CC} = \text{MAX.}$ $V_{IN} = 2.7 \text{ V}$	
				0.1	mA	$V_{CC} = \text{MAX.}$ $V_{IN} = 7.0 \text{ V}$	
I_{IL}	Input LOW Current			-0.4	mA	$V_{CC} = \text{MAX.}$ $V_{IN} = 0.4 \text{ V}$	
I_{OS}	Short Circuit Current (Note 1)	-20		-100	mA	$V_{CC} = \text{MAX}$	
I_{CC}	Power Supply Current Total, Output HIGH Total, Output LOW			2.4	mA	$V_{CC} = \text{MAX}$	
				6.6			

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS ($T_A = 25^\circ\text{C}$)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
t_{PLH}	Turn-Off Delay, Input to Output		9.0	15	ns	$V_{CC} = 5.0 \text{ V}$ $C_L = 15 \text{ pF}$
t_{PHL}	Turn-On Delay, Input to Output		10	15	ns	

BCD TO 7-SEGMENT DECODER/DRIVER

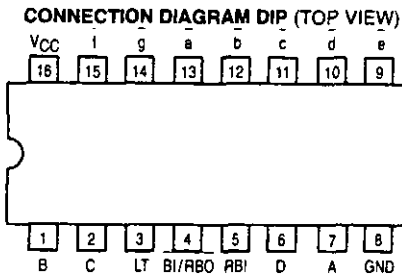
The SN54/74LS47 are Low Power Schottky BCD to 7-Segment Decoder/Drivers consisting of NAND gates, input buffers and seven AND-OR-INVERT gates. They offer active LOW, high sink current outputs for driving indicators directly. Seven NAND gates and one driver are connected in pairs to make BCD data and its complement available to the seven decoding AND-OR-INVERT gates. The remaining NAND gate and three input buffers provide lamp test, blanking input/ripple-blanking output and ripple-blanking input.

The circuits accept 4-bit binary-coded-decimal (BCD) and, depending on the state of the auxiliary inputs, decodes this data to drive a 7-segment display indicator. The relative positive-logic output levels, as well as conditions required at the auxiliary inputs, are shown in the truth tables. Output configurations of the SN54/74LS47 are designed to withstand the relatively high voltages required for 7-segment indicators.

These outputs will withstand 15 V with a maximum reverse current of 250 μ A. Indicator segments requiring up to 24 mA of current may be driven directly from the SN74LS47 high performance output transistors. Display patterns for BCD input counts above nine are unique symbols to authenticate input conditions.

The SN54/74LS47 incorporates automatic leading and/or trailing-edge zero-blanking control (RBI and RBO). Lamp test (LT) may be performed at any time which the BI/RBO node is a HIGH level. This device also contains an overriding blanking input (BI) which can be used to control the lamp intensity by varying the frequency and duty cycle of the BI input signal or to inhibit the outputs.

- Lamp Intensity Modulation Capability (BI/RBO)
- Open Collector Outputs
- Lamp Test Provision
- Leading/Trailing Zero Suppression
- Input Clamp Diodes Limit High-Speed Termination Effects


PIN NAMES

A, B, C, D	BCD Inputs
RBI	Ripple-Blanking Input
LT	Lamp-Test Input
BI/RBO	Blanking input or Ripple-Blanking Output
a, to g	Outputs

LOADING (Note a)

	HIGH	LOW
A, B, C, D	0.5 U.L.	0.25 U.L.
RBI	0.5 U.L.	0.25 U.L.
LT	0.5 U.L.	0.25 U.L.
BI/RBO	0.5 U.L.	0.75 U.L.
a, to g	1.2 U.L.	2.0 U.L.
	Open-Collector	15 (7.5) U.L.

NOTES:

a) 1 Unit Load (U.L.) = 40 μ A HIGH, 1.6 mA LOW.

b) Output current measured at $V_{OUT} = 0.5$ V

The Output LOW drive factor is 7.5 U.L. for Military (54) and 15 U.L. for Commercial (74) Temperature Ranges.

SN54/74LS47

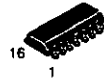
BCD TO 7-SEGMENT DECODER/DRIVER LOW POWER SCHOTTKY



**J SUFFIX
CERAMIC
CASE 620-09**



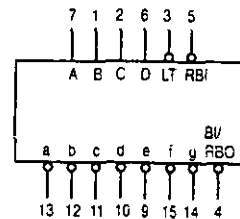
**N SUFFIX
PLASTIC
CASE 648-08**



**O SUFFIX
SOIC
CASE 751B-03**

ORDERING INFORMATION

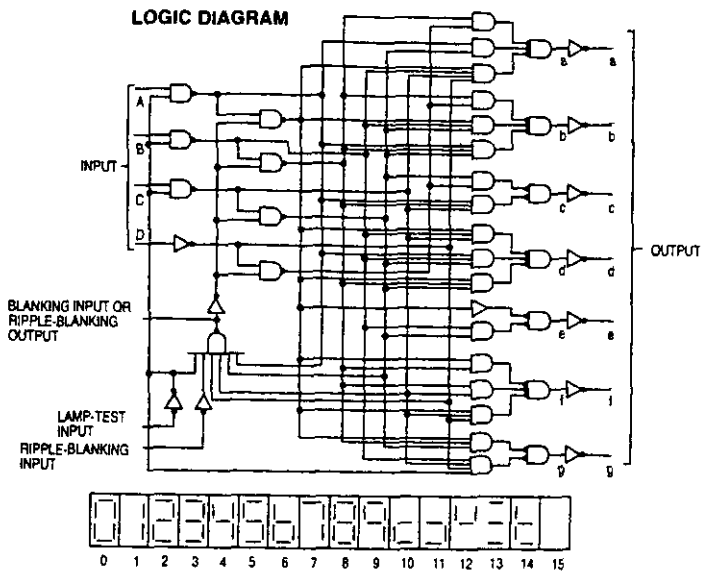
SN54LSXXJ	Ceramic
SN74LSXXN	Plastic
SN74LSXXD	SOIC

LOGIC SYMBOL


VCC = PIN 16
GND = PIN 8

SN54/74LS47

LOGIC DIAGRAM



NUMERICAL DESIGNATIONS — RESULTANT DISPLAYS

TRUTH TABLE

DECIMAL OR FUNCTION	INPUTS						OUTPUTS										NOTE
	LT	RBI	D	C	B	A	B _V /RBO	a	b	c	d	e	f	g			
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	A		
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H	A		
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L			
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L			
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L			
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L			
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L			
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H			
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L			
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L			
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L			
11	H	X	H	L	H	H	H	H	L	L	H	H	L	L			
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L			
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L			
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L			
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H			
B _I	X	X	X	X	X	X	L	H	H	H	H	H	H	H	B		
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H	C		
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L	D		

H = HIGH Voltage Level
 L = LOW Voltage Level
 X = Immaterial

NOTES

- (A) B_V/RBO is wire-AND logic serving as blanking input (BI) and/or ripple-blanking output (RBO). The blanking out (BI) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input (RBI) must be open or at a HIGH level if blanking of a decimal 0 is not desired. X = input may be HIGH or LOW
- (B) When a LOW level is applied to the blanking input (forced condition) all segment outputs go to a LOW level regardless of the state of any other input condition.
- (C) When ripple-blanking input (RBI) and inputs A, B, C, and D are at LOW level, with the lamp test input at HIGH level, all segment outputs go to a HIGH level and the ripple-blanking output (RBO) goes to a LOW level (response condition)
- (D) When the blanking input/ripple-blanking output (B_V/RBO) is open or held at a HIGH level, and a LOW level is applied to lamp test input, all segment outputs go to a LOW level.

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V _{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T _A	Operating Ambient Temperature Range	54 74	-55 0	25 25	125 70	°C
I _{OH}	Output Current — High BI/RBO	54, 74			-50	μA
I _{OL}	Output Current — Low BI/RBO BI/RBO	54 74			1.6 3.2	mA
V _{O (off)}	Off-State Output Voltage \bar{a} to \bar{g}	54, 74			15	V
I _{O (on)}	On-State Output Current \bar{a} to \bar{g} \bar{a} to \bar{g}	54 74			12 24	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Threshold Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Threshold Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IN} = -18 mA
V _{OH}	Output HIGH Voltage, BI/RBO	2.4	4.2		V	V _{CC} = MIN, I _{OH} = -50 μA, V _{IN} = V _{IH} or V _{IL} per Truth Table
V _{OL}	Output LOW Voltage BI/RBO	54, 74	0.25	0.4	V	I _{OL} = 1.6 mA V _{CC} = MIN, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	0.35	0.5		
I _{O (off)}	Off-State Output Current \bar{a} thru \bar{g}			250	μA	V _{CC} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table, V _{O (off)} = 15 V
V _{O (on)}	On-State Output Voltage \bar{a} thru \bar{g}	54, 74	0.25	0.4	V	I _{O (on)} = 12 mA V _{CC} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	0.35	0.5		
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
I _{IL}	Input LOW Current BI/RBO Any Input except BI/RBO			-1.2 -0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{OS BI/RBO}	Output Short Circuit Current (Note 1)	-0.3		-2.0	mA	V _{CC} = MAX, V _{OUT} = 0 V
I _{CC}	Power Supply Current		7.0	13	mA	V _{CC} = MAX

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
t _{PHL}	Propagation Delay, Address Input to Segment Output			100	ns	V _{CC} = 5.0 V C _L = 15 pF
t _{PLH}				100		
t _{PHL}	Propagation Delay, RBI Input To Segment Output			100	ns	
t _{PLH}				100		

AC WAVEFORMS

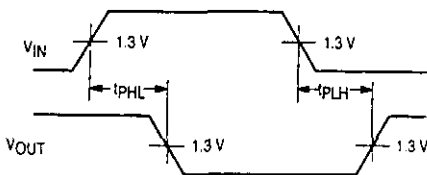


Figure 1

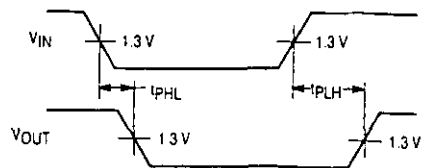


Figure 2



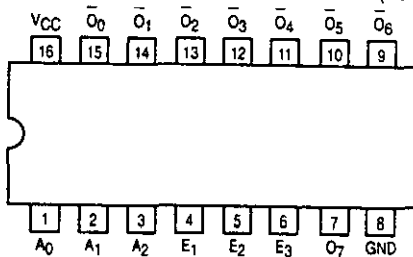
MOTOROLA

1-OF-8 DECODER/ DEMULTIPLEXER

The LSTTL/MSI SN54/74LS138 is a high speed 1-of-8 Decoder/Demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding. The multiple input enables allow parallel expansion to a 1-of-24 decoder using just three LS138 devices or to a 1-of-32 decoder using four LS138s and one inverter. The LS138 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

- Demultiplexing Capability
- Multiple Input Enable for Easy Expansion
- Typical Power Dissipation of 32 mW
- Active Low Mutually Exclusive Outputs
- Input Clamp Diodes Limit High Speed Termination Effects

CONNECTION DIAGRAM DIP (TOP VIEW)



NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

PIN NAMES

A ₀ -A ₂	Address Inputs
E ₁ , E ₂	Enable (Active LOW) Inputs
E ₃	Enable (Active HIGH) Input
O ₀ -O ₇	Active LOW Outputs (Note b)

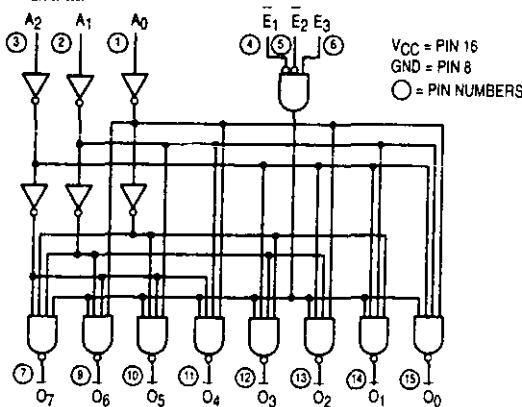
LOADING (Note a)

HIGH	LOW
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
10 U.L.	5 (2.5) U.L.

NOTES:

- a) 1 TTL Unit Load (U.L.) = 40 μA HIGH/1.6 mA LOW.
 b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

LOGIC DIAGRAM



SN54/74LS138

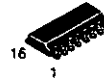
1-OF-8 DECODER/ DEMULTIPLEXER LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 620-09



N SUFFIX
PLASTIC
CASE 648-08

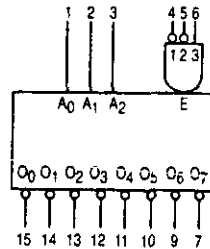


D SUFFIX
SOIC
CASE 751B-03

ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXD	SOIC

LOGIC SYMBOL



VCC = PIN 16
GND = PIN 8

SN54/74LS138

FUNCTIONAL DESCRIPTION

The LS138 is a high speed 1-of-8 Decoder/Demultiplexer fabricated with the low power Schottky barrier diode process. The decoder accepts three binary weighted inputs (A_0, A_1, A_2) and when enabled provides eight mutually exclusive active LOW Outputs (O_0-O_7). The LS138 features three Enable inputs, two active LOW (E_1, E_2) and one active HIGH (E_3). All outputs will be HIGH unless E_1 and E_2 are LOW and E_3 is HIGH. This multiple enable function allows easy parallel ex-

pansion of the device to a 1-of-32 (5 lines to 32 lines) decoder with just four LS138s and one inverter. (See Figure a.)

The LS138 can be used as an 8-output demultiplexer by using one of the active LOW Enable inputs as the data input and the other Enable inputs as strobes. The Enable inputs which are not used must be permanently tied to their appropriate active HIGH or active LOW state.

TRUTH TABLE

INPUTS						OUTPUTS							
E_1	E_2	E_3	A_0	A_1	A_2	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

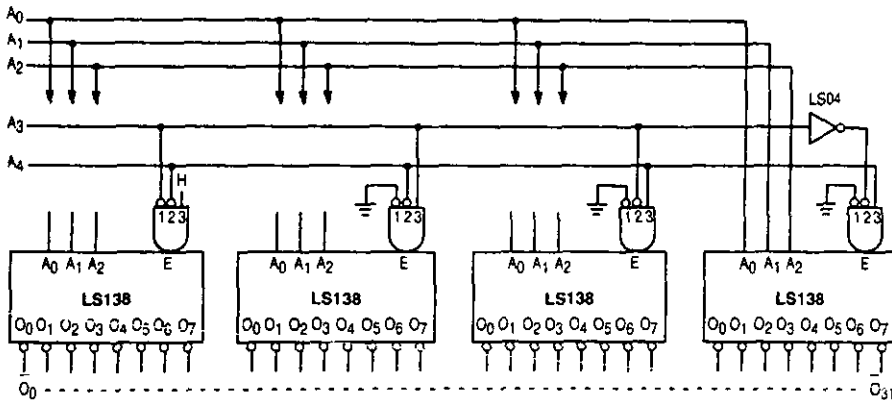


Figure a

SN54/74LS138

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V _{CC}	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T _A	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54			4.0	mA
		74			8.0	

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions	
		Min	Typ	Max			
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74		0.8			
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IN} = -18 mA	
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table	
		74	2.7	3.5	V		
V _{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I _{OL} = 4.0 mA V _{CC} = V _{CC} MIN, V _{IN} = V _{IL} or V _{IH} per Truth Table
		74		0.35	0.5	V	
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V	
				0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V	
I _{IL}	Input LOW Current			-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V	
I _{OS}	Short Circuit Current (Note 1)	-20		-100	mA	V _{CC} = MAX	
I _{CC}	Power Supply Current			10	mA	V _{CC} = MAX	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C)

Symbol	Parameter	Levels of Delay	Limits			Unit	Test Conditions
			Min	Typ	Max		
t _{PLH}	Propagation Delay Address to Output	2		13	20	ns	V _{CC} = 5.0 V C _L = 15 pF
t _{PHL}	Propagation Delay Address to Output	2		27	41		
t _{PLH}	Propagation Delay Address to Output	3		18	27		
t _{PHL}	Propagation Delay Address to Output	3		26	39		
t _{PLH}	Propagation Delay E ₁ or E ₂ Enable to Output	2		12	18		
t _{PHL}	Propagation Delay E ₁ or E ₂ Enable to Output	2		21	32		
t _{PLH}	Propagation Delay E ₃ Enable to Output	3		17	26	ns	
t _{PHL}	Propagation Delay E ₃ Enable to Output	3		25	38		

AC WAVEFORMS

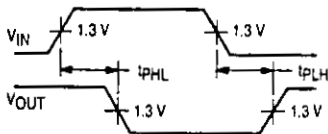


Figure 1

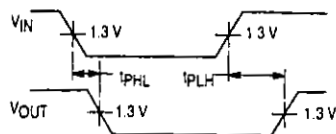
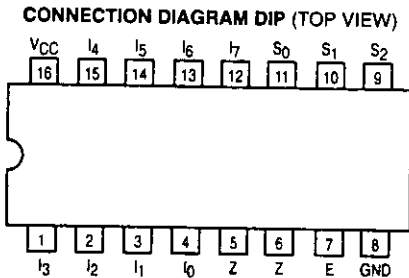


Figure 2

8-INPUT MULTIPLEXER

The TTL/MSI SN54/74LS151 is a high speed 8-input Digital Multiplexer. It provides, in one package, the ability to select one bit of data from up to eight sources. The LS151 can be used as a universal function generator to generate any logic function of four variables. Both assertion and negation outputs are provided.

- Schottky Process for High Speed
- Multifunction Capability
- On-Chip Select Logic Decoding
- Fully Buffered Complementary Outputs
- Input Clamp Diodes Limit High Speed Termination Effects



SN54/74LS151

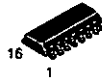
**8-INPUT MULTIPLEXER
LOW POWER SCHOTTKY**



**J SUFFIX
CERAMIC
CASE 620-09**



**N SUFFIX
PLASTIC
CASE 648-08**



**D SUFFIX
SOIC
CASE 751B-03**

ORDERING INFORMATION

- SN54LSXXXJ Ceramic
- SN74LSXXXN Plastic
- SN74LSXXXD SOIC

PIN NAMES

- S₀-S₂ Select Inputs
- E Enable (Active LOW) Input
- I₀-I₇ Multiplexer Inputs
- Z Multiplexer Output (Note b)
- Z Complementary Multiplexer Output (Note b)

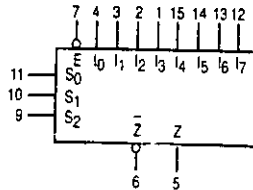
LOADING (Note a)

	HIGH	LOW
S ₀ -S ₂	0.5 U.L.	0.25 U.L.
E	0.5 U.L.	0.25 U.L.
I ₀ -I ₇	0.5 U.L.	0.25 U.L.
Z	10 U.L.	5 (2.5) U.L.
Z	10 U.L.	5 (2.5) U.L.

NOTES:

- a) 1 TTL Unit Load (U.L.) = 40 μA HIGH/1.6 mA LOW.
- b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges

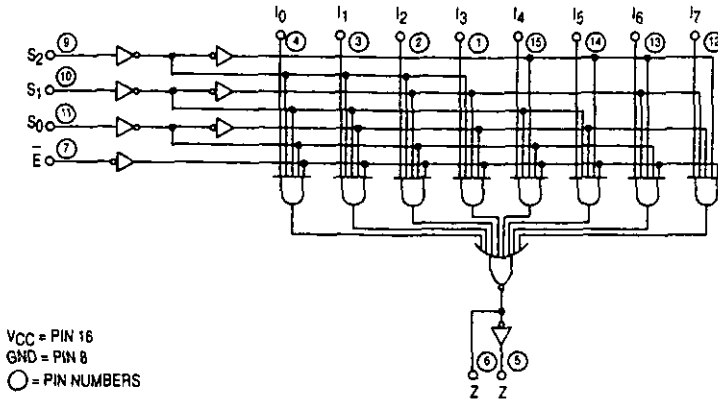
LOGIC SYMBOL



VCC = PIN 16
GND = PIN 8

SN54/74LS151

LOGIC DIAGRAM



FUNCTIONAL DESCRIPTION

The LS151 is a logical implementation of a single pole, 8-position switch with the switch position controlled by the state of three Select inputs, S_0 , S_1 , S_2 . Both assertion and negation outputs are provided. The Enable input (E) is active LOW. When it is not activated, the negation output is HIGH and the assertion output is LOW regardless of all other inputs. The logic function provided at the output is:

$$Z = \bar{E} \cdot (I_0 \cdot \bar{S}_0 \cdot \bar{S}_1 \cdot \bar{S}_2 + I_1 \cdot \bar{S}_0 \cdot \bar{S}_1 \cdot S_2 + I_2 \cdot \bar{S}_0 \cdot S_1 \cdot \bar{S}_2 + I_3 \cdot \bar{S}_0 \cdot S_1 \cdot S_2 + I_4 \cdot S_0 \cdot \bar{S}_1 \cdot \bar{S}_2 + I_5 \cdot S_0 \cdot \bar{S}_1 \cdot S_2 + I_6 \cdot S_0 \cdot S_1 \cdot \bar{S}_2 + I_7 \cdot S_0 \cdot S_1 \cdot S_2)$$

The LS151 provides the ability, in one package, to select from eight sources of data or control information. By proper manipulation of the inputs, the LS151 can provide any logic function of four variables and its negation.

TRUTH TABLE

E	S ₂	S ₁	S ₀	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	Z	Z
H	X	X	X	X	X	X	X	X	X	X	X	H	L
L	L	L	L	L	X	X	X	X	X	X	X	H	L
L	L	L	L	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	H	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	H	L
L	L	H	L	X	X	H	X	X	X	X	X	L	H
L	L	H	H	X	X	X	H	X	X	X	X	H	L
L	H	L	L	X	X	X	X	L	X	X	X	L	H
L	H	L	L	X	X	X	X	H	X	X	X	L	H
L	H	L	H	X	X	X	X	L	X	X	X	H	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	H	L
L	H	H	L	X	X	X	X	X	H	X	X	L	H
L	H	H	H	X	X	X	X	X	X	X	X	L	H
L	H	H	H	X	X	X	X	X	X	X	H	L	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V _{CC}	Supply Voltage		4.5	5.0	5.5	V
			74	5.0	5.25	
T _A	Operating Ambient Temperature Range		-55	25	125	°C
			74	25	70	
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low				4.0	mA
			74		8.0	

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions	
		Min	Typ	Max			
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74		0.8			
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IN} = -18 mA	
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table	
		74	2.7	3.5	V		
V _{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I _{OL} = 4.0 mA V _{CC} = V _{CC} MIN, V _{IN} = V _{IL} or V _{IH} per Truth Table
		74		0.35	0.5	V	
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V	
				0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V	
I _{IL}	Input LOW Current			-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V	
I _{OS}	Short Circuit Current (Note 1)	-20		-100	mA	V _{CC} = MAX	
I _{CC}	Power Supply Current			10	mA	V _{CC} = MAX	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
t _{PLH}	Propagation Delay Select to Output Z		27	43	ns	V _{CC} = 5.0 V C _L = 15 pF
t _{PHL}	Select to Output Z		18	30		
t _{PLH}	Propagation Delay Select to Output Z		14	23	ns	
t _{PHL}	Enable to Output Z		20	32		
t _{PLH}	Propagation Delay Enable to Output Z		26	42	ns	
t _{PHL}	Data to Output Z		20	32		
t _{PLH}	Propagation Delay Enable to Output Z		15	24	ns	
t _{PHL}	Data to Output Z		18	30		
t _{PLH}	Propagation Delay Data to Output Z		20	32	ns	
t _{PHL}	Data to Output Z		16	26		
t _{PLH}	Propagation Delay Data to Output Z		13	21	ns	
t _{PHL}	Data to Output Z		12	20		

AC WAVEFORMS

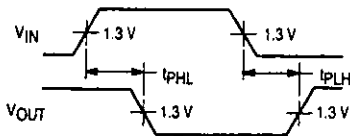


Figure 1

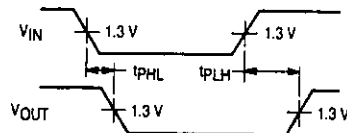
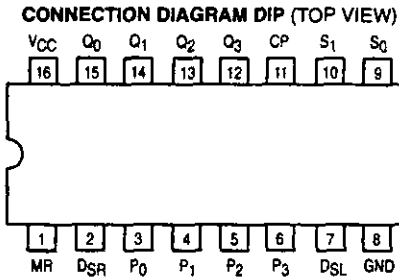


Figure 2

4-BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTER

The SN54/74LS194A is a High Speed 4-Bit Bidirectional Universal Shift Register. As a high speed multifunctional sequential building block, it is useful in a wide variety of applications. It may be used in serial-serial, shift left, shift right, serial-parallel, parallel-serial, and parallel-parallel data register transfers. The LS194A is similar in operation to the LS195A Universal Shift Register, with added features of shift left without external connections and hold (do nothing) modes of operation. It utilizes the Schottky diode clamped process to achieve high speeds and is fully compatible with all Motorola TTL families.

- Typical Shift Frequency of 36 MHz
- Asynchronous Master Reset
- Hold (Do Nothing) Mode
- Fully Synchronous Serial or Parallel Data Transfers
- Input Clamp Diodes Limit High Speed Termination Effects



PIN NAMES

S ₀ , S ₁	Mode Control Inputs
P ₀ –P ₃	Parallel Data Inputs
DSR	Serial (Shift Right) Data Input
DSL	Serial (Shift Left) Data Input
CP	Clock (Active HIGH Going Edge) Input
MR	Master Reset (Active LOW) Input
Q ₀ –Q ₃	Parallel Outputs (Note b)

LOADING (Note a)

	HIGH	LOW
S ₀ , S ₁	0.5 U.L.	0.25 U.L.
P ₀ –P ₃	0.5 U.L.	0.25 U.L.
DSR	0.5 U.L.	0.25 U.L.
DSL	0.5 U.L.	0.25 U.L.
CP	0.5 U.L.	0.25 U.L.
MR	0.5 U.L.	0.25 U.L.
Q ₀ –Q ₃	10 U.L.	5 (2.5) U.L.

NOTES:

- a. 1 TTL Unit Load (U.L.) = 40 μ A HIGH/1.6 mA LOW.
 b. The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

SN54/74LS194A

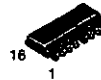
**4-BIT BIDIRECTIONAL
UNIVERSAL SHIFT REGISTER**
LOW POWER SCHOTTKY



**J SUFFIX
CERAMIC
CASE 620-09**



**N SUFFIX
PLASTIC
CASE 648-08**



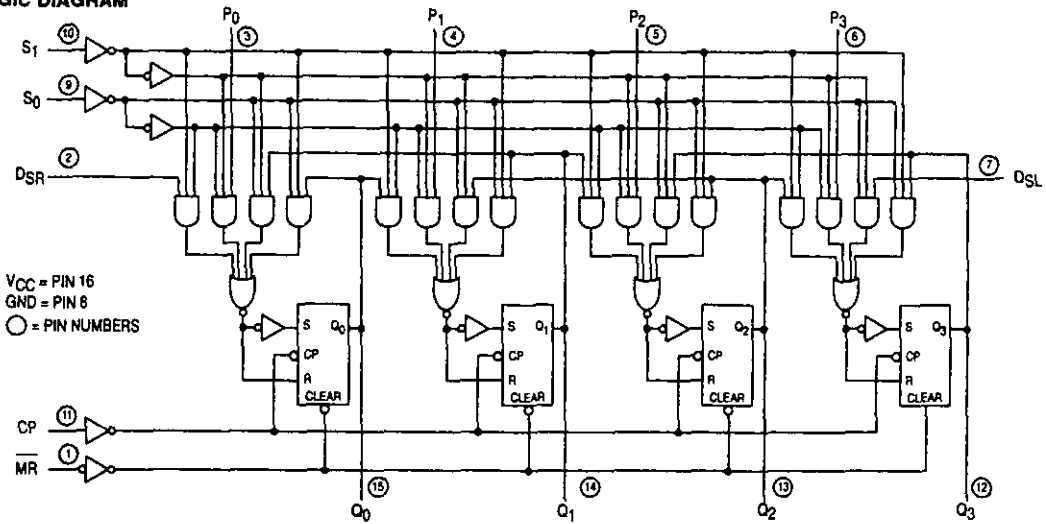
**D SUFFIX
SOIC
CASE 751B-03**

ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXD	SOIC

SN54/74LS194A

LOGIC DIAGRAM



FUNCTIONAL DESCRIPTION

The Logic Diagram and Truth Table indicate the functional characteristics of the LS194A 4-Bit Bidirectional Shift Register. The LS194A is similar in operation to the Motorola LS195A Universal Shift Register when used in serial or parallel data register transfers. Some of the common features of the two devices are described below:

All data and mode control inputs are edge-triggered, responding only to the LOW to HIGH transition of the Clock (CP). The only timing restriction, therefore, is that the mode control and selected data inputs must be stable one set-up time prior to the positive transition of the clock pulse.

The register is fully synchronous, with all operations taking place in less than 15 ns (typical) making the device especially useful for implementing very high speed CPUs, or the memory buffer registers.

The four parallel data inputs (P₀, P₁, P₂, P₃) are D-type inputs. When both S₀ and S₁ are HIGH, the data appearing on P₀, P₁, P₂, and P₃ inputs is transferred to the Q₀, Q₁, Q₂, and

Q₃ outputs respectively following the next LOW to HIGH transition of the clock.

The asynchronous Master Reset (MR), when LOW, overrides all other input conditions and forces the Q outputs LOW.

Special logic features of the LS194A design which increase the range of application are described below:

Two mode control inputs (S₀, S₁) determine the synchronous operation of the device. As shown in the Mode Selection Table, data can be entered and shifted from left to right (shift right, Q₀ → Q₁, etc.) or right to left (shift left, Q₃ → Q₂, etc.), or parallel data can be entered loading all four bits of the register simultaneously. When both S₀ and S₁ are LOW, the existing data is retained in a "do nothing" mode without restricting the HIGH to LOW clock transition.

D-type serial data inputs (DSR, DSL) are provided on both the first and last stages to allow multistage shift right or shift left data transfers without interfering with parallel load operation.

MODE SELECT — TRUTH TABLE

OPERATING MODE	INPUTS						OUTPUTS			
	MR	S ₁	S ₀	DSR	DSL	P _n	Q ₀	Q ₁	Q ₂	Q ₃
Reset	L	X	X	X	X	X	L	L	L	L
Hold	H	l	l	X	X	X	q ₀	q ₁	q ₂	q ₃
Shift Left	H	h	l	X	l	X	q ₁	q ₂	q ₃	L
	H	h	l	X	h	X	q ₁	q ₂	q ₃	H
Shift Right	H	l	h	l	X	X	L	q ₀	q ₁	q ₂
	H	l	h	h	X	X	H	q ₀	q ₁	q ₂
Parallel Load	H	h	h	X	X	P _n	P ₀	P ₁	P ₂	P ₃

L = LOW Voltage Level

H = HIGH Voltage Level

X = Don't Care

l = LOW voltage level one set-up time prior to the LOW to HIGH clock transition

h = HIGH voltage level one set-up time prior to the LOW to HIGH clock transition

P_n (q_n) = Lower case letters indicate the state of the referenced input (or output) one set-up time prior to the LOW to HIGH clock transition

SN54/74LS194A

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V _{CC}	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T _A	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54			4.0	mA
		74			8.0	

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IN} = -18 mA
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	2.7	3.5	V	
V _{OL}	Output LOW Voltage	54, 74	0.25	0.4	V	I _{OL} = 4.0 mA
		74	0.35	0.5	V	I _{OL} = 8.0 mA
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
				0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _{IL}	Input LOW Current			-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{OS}	Short Circuit Current (Note 1)	-20		-100	mA	V _{CC} = MAX
I _{CC}	Power Supply Current			23	mA	V _{CC} = MAX

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
f _{MAX}	Maximum Clock Frequency	25	36		MHz	V _{CC} = 5.0 V C _L = 15 pF
t _{PLH}	Propagation Delay, Clock to Output		14	22	ns	
t _{PHL}			17	26		
t _{PHL}	Propagation Delay, MR to Output		19	30	ns	