



19
2
8

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Escuela Nacional de Estudios Profesionales
"ARAGON"

MICROPROCESADORES

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

Ricardo López Mijares



MEXICO, D. F.

1993

TESIS CON
FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

I.- INTRODUCCION A LOS MICROPROCESADORES Y MICROCOMPUTADORAS

1.1	Elementos de una microcomputadora	3
	El microprocesador	3
	Las memorias	5
	Interfaces	7
	Comunicación entre los módulos	9

II.- ESTRUCTURA DE UN MICROCOMPUTADOR

2.1	Las señales	13
2.2	Los elementos internos	14
2.3	El funcionamiento interno: los diagramas de estado	19
	El ciclo Fetch	20
2.4	Estudio detallado de un microprocesador de 8 bits	20

III.- ESTRUCTURA DE LOS ELEMENTOS DE MEMORIA

3.1	Estado actual de la tecnología	26
3.2	Estructura de los elementos de memoria	27
3.3	Estudio detallado de varios casos (RAM, ROM)	40
	RAM NMC2148	40
	RAM NMC6164	40
	EPROM MM2716	42

IV.- INTRODUCCION A LA PROGRAMACION DE MICROPROCESADORES

4.1	Direccionamiento de memoria y E/S	48
4.2	Banderas y Saltos	52
4.3	Conteo y lazos de tiempo	55
4.4	Funciones aritméticas	60
4.5	Conceptos avanzados	60
4.6	Interrupciones	70

V.- PROGRAMACION DEL MICROPROCESADOR

5.1	Lenguaje máquina y ensambladores	79
	Instrucciones aritméticas	80
	Instrucciones lógicas	80
	Instrucciones de transferencia de datos	80
	Instrucciones de bifurcación	80
	Instrucciones de llamada de subrutina	81
	Instrucciones varias	81
5.2	Programa monitor	81
5.3	Registros, Memorias y Buses	82
	Registro	82
	Memoria	83
	Buses	84
	Ejemplos de programación	85

VI.- INTERFAZ DE MICROCOMPUTADORAS CON DISPOSITIVOS EXTERNOS

6.1	Uso de líneas programadas Entrada/Salida para el control de dispositivos	123
6.2	Control de dispositivos con técnicas de programación	128
6.3	Interfase RS-232 y lazo de corriente de 20 mA	129

VII.- COMUNICACION SERIE ASINCRONA Y SINCRONA

7.1	Estructura de la comunicación asíncrona	137
7.2	Interfases asíncronas: estudio del UART y similares	138
7.3	Estructura de la comunicación serie síncrona	142
7.4	Interfases síncronas: los USART y similares	143
7.5	Comunicación síncrona compleja: Protocolos	147

VIII.- COMUNICACION PARALELO

8.1	Estructura de la comunicación paralelo	152
8.2	El saludo (handshake), tipos	152
8.3	Interfases integrados	153
	PIA	156
	PIO	156
	PPI	160
8.4	Interfases universales UPI y similares	161

IX.- APLICACIONES DE LOS MICROPROCESADORES

9.1	Aplicación de los microprocesadores en instrumentación y control	165
	Comunicaciones	165
	Computadoras	166
	Industria	166
	Equipo de negocios	167
	Transportación	167
	Consumidor/Comercial	168

INDICE DE FIGURAS

FIGURA I-1	Interfaces principales en una microcomputadora	8
FIGURA II-1	Elementos de un microprocesador básico	15
FIGURA II-2	Arquitectura de una microcomputadora	17
FIGURA II-3	Secuencia del ciclo Fetch	21
FIGURA III-1	Estructura de una computadora	28
FIGURA III-2	Estructura de una memoria RAM	30
FIGURA III-3	Estructura de una memoria ROM bipolar	32
FIGURA III-4	Unidad de programación de memorias PROM	34
FIGURA III-5	Celda de memoria RAM con seis transistores	35
FIGURA III-6	Conexión de celdas en una memoria RAM	37
FIGURA III-7	Celda de almacenamiento dinámico	39
FIGURA III-8	Diagrama de la memoria RAM NCM2148	41
FIGURA III-9	Diagrama de la memoria RAM NCM6164	43
FIGURA III-10	Diagrama de la memoria EPROM MM2716	44
FIGURA IV-1	Instrucción "colocación de la bandera de acarreo" (STC)	50
FIGURA IV-2	Registros de propósito general del microprocesador 8080/8085 y su registro de estado	53
FIGURA IV-3	Hardware para un teclado	57
FIGURA IV-4	Diagrama de flujo para una subrutina de tiempo	59
FIGURA IV-5	Circuitos ALU TTL	61
FIGURA IV-6	Circuito de interfaz representativo para un sistema basado en el 8080 y con dispositivos de E/S administrados con interrupciones	72
FIGURA IV-7	Interfaz paralela 8212 de Intel	73
FIGURA IV-8	El CI de E/S 8212 como puerto de salida	74
FIGURA IV-9	Estructura general de una rutina de servicio de interrupciones	76
FIGURA V-1	Instrucción "suma A a los datos" (ADI)	86
FIGURA V-2	Instrucción "suma L a A" (ADD L)	87

FIGURA V-3	Instrucción "suma indirecta de registro" (ADD M)	94
FIGURA V-4	Instrucción "resta los datos de A" (SUI)	95
FIGURA V-5	Instrucción "compara A con L" (CMP L)	98
FIGURA V-6	Instrucción "AND inmediata" (ANI)	99
FIGURA V-7	Instrucción "liga A con L mediante un OR" (ORAL)	101
FIGURA V-8	Instrucción "rotar hacia la derecha por medio de acarreo" (RAR)	103
FIGURA V-9	Instrucción "mueve A a L" (MOVE L,A)	105
FIGURA V-10	Instrucción "mueve HL a SP" (SPHL)	106
FIGURA V-11	Instrucción "carga inmediata de HL" (LXI H)	107
FIGURA V-12	Instrucción "carga directa de HL" (LHLD)	108
FIGURA V-13	Instrucción "almacenamiento directo de A" (STA)	110
FIGURA V-14	Instrucción "carga inmediata de HL" (IN)	111
FIGURA V-15	Instrucción "brinco incondicional" (JMP)	113
FIGURA V-16	Instrucción "brinca si es cero" (JZ)	114
FIGURA V-17	Ejemplo de una subrutina	116
FIGURA V-18	Instrucción "llamado de subrutina" (CALL)	117
FIGURA V-19	Instrucción "regreso de subrutina" (RET)	119
FIGURA VI-1	Instrucciones de Entrada/Salida	124
FIGURA VI-2	Temporización de señales	127
FIGURA VI-3	Ejemplo de comunicación entre un DTE y un DCE	131
FIGURA VI-4	Interfaz RS-232C y Lazo de Corriente	134
FIGURA VII-1	Operaciones de lectura y escritura asíncronas	139
FIGURA VII-2	Organización del circuito Intel 8251 (USART)	141
FIGURA VII-3	Utilización del CI 8251 USART para interfaz con un terminal CRT	146
FIGURA VII-4	Protocolos de enlace de datos entre un DTE y un DCE	148
FIGURA VIII-1	Interfaz de E/S en paralelo de la serie 6800 de Motorola, la PIA 6821 ..	157
FIGURA VIII-2	Interfaz Periférica Programable 8255	159
FIGURA VIII-3	Circuito de Entrada/Salida Universal	162

PREFACIO

El Trabajo de Tesis, denominado "Microprocesadores", fué realizado con el propósito de dar apoyo a la materia que lleva el mismo nombre y que conforma el nuevo Plan de Estudios de la carrera de Ingeniería Mecánica Eléctrica de la Escuela Nacional de Estudios Profesionales, Plantel Aragón.

Este Plan de Estudios propuesto por el Consejo Técnico y aprobado por las autoridades de la Universidad Nacional Autónoma de México entró en vigor al dar inicio el semestre 92-1.

La elaboración de esta Tesis tiene como finalidad poner en manos del alumno los temas fundamentales recopilados de textos que contienen los últimos avances en microprocesadores y electrónica, y han sido estructurados en forma concreta y operacional, más que abstracta y deductiva, tomando lo más importante para que el estudiante domine conceptos, técnicas de diseño de arquitecturas y rutinas de programación.

El texto está descrito en forma integrada, de tal manera que los capítulos están complementados entre sí, mostrándose a continuación una descripción de cada uno.

CAPITULO I

INTRODUCCION A LOS MICROPROCESADORES Y MICROCOMPUTADORAS

Se dará una reseña histórica propia del microprocesador así como los alcances que se espera tener con este dispositivo.

CAPITULO II

ESTRUCTURA DE UNA MICROCOMPUTADORA

Se manejarán conceptos de uso continuo desde el punto de vista electrónico y adquirir una idea más clara del uso del microprocesador.

CAPITULO III

ESTRUCTURA DE LOS ELEMENTOS DE MEMORIA

Se estudiará a uno de los dispositivos de más importancia para poder desarrollar arquitecturas electrónicas de utilidad. Este elemento lleva la responsabilidad de almacenar toda la información que se maneja en estos sistemas.

CAPITULO IV

INTRODUCCION A LA PROGRAMACION DE MICROPROCESADORES

Se detallarán las partes de que está conformado el microprocesador, dándose las bases para poderlo programar.

CAPITULO V

PROGRAMACION DEL MICROPROCESADOR

Se enlistarán todas las instrucciones que maneja un microprocesador de 8 bits y se ejemplificarán formas de usarlas para programación.

CAPITULO VI

INTERFAZ DE MICROCOMPUTADORAS CON DISPOSITIVOS EXTERNOS

Se darán a conocer las formas de decodificación de direcciones para las memorias, puertos y otros periféricos.

CAPITULO VII

COMUNICACION SERIE ASINCRONA Y SINCRONA

Se estudiará la comunicación serie entre bloques, dándose referencia a elementos propios de este tipo de comunicación, así como sus ventajas.

CAPITULO VIII

COMUNICACION PARALELO

Se explicará la forma de comunicación en paralelo, los dispositivos que involucra y las ventajas que ofrece al usuario.

CAPITULO IX

APLICACIONES DE LOS MICROPROCESADORES

Se dará una serie de ejemplos prácticos reales en los que se emplea el microprocesador.

Se espera que el empleo de este texto proporcione al estudiante y al profesional un conocimiento tangible y real en lo que respecta a la arquitectura, la programación y el diseño de sistemas basados en microprocesadores; sin embargo, ningún curso o libro podrán sustituir a la experiencia y práctica, pero los conceptos fundamentales son esenciales para adquirirla.

EL AUTOR

CAPITULO I

INTRODUCCION A LOS MICROPROCESADORES Y MICROCOMPUTADORAS

INTRODUCCION

El estudio de la microcomputadora, sus partes y funcionamiento ha dado lugar a innumerables investigaciones, artículos y libros. En este capítulo se muestran las partes que la componen.

Pero ¿qué es una microcomputadora?, es un sistema completo, incluyendo la unidad central, la memoria, las interfaces de entrada, salida y la alimentación, todo en un mínimo de espacio.

La necesidad de manejar grandes cantidades de información a menor velocidad, dió lugar a la creación del microprocesador, que no es mas que un circuito integrado o "chip" (llamado también pastilla o encapsulado), con gran densidad de integración (LSI), especializado en el tratamiento y la gestión de datos. Un microprocesador es por lo tanto el corazón del microordenador, aunque mida algunos milímetros a los lados.

Existen muchos fabricantes de microprocesadores y entre los más conocidos se encuentran ZILOG, INTEL, MOTOROLA, etc. Las clases de microprocesadores se caracterizan por el número de bits tratados en paralelo: 4, 8, 16 y 32 bits.

1.1 ELEMENTOS DE UNA MICROCOMPUTADORA

EL MICROPROCESADOR

Un microprocesador es el centro de las operaciones y transferencias de todos los datos de una computadora pequeña que se construye utilizando circuitería LSI (Larga Escala de Integración) y en una sola pastilla por lo general.

Un microprocesador no es una computadora, sino parte de la computadora misma.

La adición de una memoria externa más los circuitos de entrada/salida (E/S) para operar equipo periférico harán una microcomputadora.

Los microprocesadores son circuitos muy elaborados en los que se han integrado unos 5,000 componentes electrónicos, dispuestos en cientos, e incluso miles de compuertas. Estas compuertas se *organizan* en registros, controladores, contadores, etc. Estos dispositivos son capaces de ejecutar casi 200 instrucciones distintas.

El corazón del microprocesador es la unidad Aritmético-Lógica (ALU), algunos registros y la circuitería de control necesaria para el procesamiento de instrucciones. Un microprocesador maneja y procesa datos aritméticos como lógicos en un modo paralelo (todos los bits de la palabra simultáneamente), bajo el control de un programa.

El microprocesador controla todas las unidades del sistema empleando las líneas de control. El bus de direcciones selecciona un lugar en la memoria, puerto de entrada o puerto de salida. El bus de datos, constituye una vía de 2 sentidos para la transferencia de datos, ya sea hacia la unidad de procesamiento o de la misma unidad. El microprocesador envía o recibe datos provenientes de la memoria haciendo uso del bus de datos.

El microprocesador siempre tiene una secuencia, captar-decodificar-ejecutar. Primero capta la instrucción de la memoria de programa, luego decodifica la instrucción. Enseguida ejecuta la orden.

En las computadoras los bits se agrupan en palabras, las palabras en la memoria de programa son interpretadas por el microprocesador, en secuencia, una a la vez.

Una característica importante de cualquier microprocesador es el tamaño de su acumulador*. En general, los acumuladores son de 8 bits de capacidad. Así que el tamaño de la palabra** de un microprocesador es de ocho bits, es decir, un byte. Un microprocesador de 16 bits tiene una longitud de palabra de 2 bytes, o 16 dígitos binarios.

La gran ventaja que se ha encontrado en este dispositivo es que maneja demasiada información en períodos de tiempo mínimos, a grandes velocidades y todo en un solo encapsulado.

Se debe resaltar el hecho de que los microprocesadores no sólo se utilizan en las microcomputadoras, sino en una gran variedad de aplicaciones como son: dispositivos periféricos, automóviles, relojes, termómetros, videojuegos, lavadoras, sistemas de alarma, básculas, sistemas de comunicación, etc. El prefijo micro significa tamaño pequeño y bajo costo.

Las investigaciones sobre amplificadores electrónicos y conmutadores fueron inicialmente motivadas por las industrias de comunicaciones, principalmente de radio, televisión y de telefonía.

Hasta la década de los 50's la mayoría de los sistemas eran analógicos en lugar de digitales. Con la aparición de las computadoras electrónicas digitales en los 40's, comienza a incrementarse la utilización de técnicas de diseño digital. Con la proliferación de las computadoras en los 60's y 70's, se comienzan a utilizar grandes cantidades de CI's, y el desarrollo de CI's de gran densidad para el mercado de computadoras, se incrementa con rapidez.

En 1970 aparecen las primeras pastillas LSI en forma de unidades de memoria para computadoras. En 1971 Intel Corporation introdujo el CI considerado como el primer microprocesador comercial, el 4004.

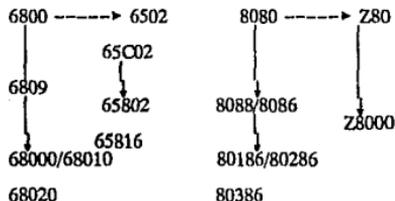
El 4004 fue seguido por otra serie de microprocesadores de varios fabricantes. Cuando el volumen de producción se incrementó los precios de los microprocesadores cayeron sustancialmente. Esto estimuló su demanda, ya que eran mucho más potentes y flexibles.

(*) El acumulador es un registro de propósito general de ocho bits que es un operador para la mayoría de las instrucciones aritméticas, lógicas, de carga, almacenamiento y de entrada/salida.

(**) Palabra es un grupo de bits que se procesa como un número simple o instrucción por un microprocesador no volátil, por lo que la información que ahí se guarda, se conserva aún cuando cese el suministro de energía.

Para los microprocesadores existía un enorme mercado, así que a finales de 1975, al menos 40 tipos diferentes de microprocesadores estaban en producción. En 1972 Intel introdujo el primer microprocesador de 8 bits, el 8008, empleaba la misma tecnología que el 4004. El 8008 fue superado por un microprocesador más rápido y más potente, el 8080 el cual fue comercializado en 1973. El 8080 era extremadamente útil, llegando a ser uno de los microprocesadores de 8 bits más usados e imitados. Otro de los primeros microprocesadores populares fue el Motorola 6800 disponible a partir de 1974, que también era de 8 bits.

La mayor parte de los microprocesadores utilizados hoy en día se pueden dividir en dos grupos, los 6's y los 8's. (Esta idea la originó el columnista de computadores John Dvorak en una columna del primer número de la revista MacUser). Los 8's lo forman los microprocesadores que se desarrollaron a partir del chip de Intel 8080 --el 8085, Z-80, 8088, 8086, 80186, 80286, 80386 y ahora el 80486. Los 6's están formados por los microprocesadores que se desarrollaron a partir del chip Motorola 6800 --el 6502, 6809, 68000, 68010, 68020, 65C02, 65802, y el 65816. Un árbol comparativo se muestra a continuación:



MEMORIAS

En la circuitería digital hay cierto tipo de dispositivos que contienen memoria. Cuando se aplica una entrada a tal circuito, la salida cambiará su estado, pero permanecerá en el nuevo estado aún después de retirar la entrada. Esta propiedad de retener su respuesta a una entrada momentánea se llama memoria.

Un microprocesador típico no contiene memoria y no es capaz de realizar funciones de entrada y salida de una computadora; para ello es necesario conectar otros circuitos integrados que realicen estas funciones. Estos circuitos son los llamados memorias.

Una memoria sólo para lectura (ROM), es un medio de almacenamiento para palabras y su contenido no puede ser normalmente alterado, es decir, es un dispositivo de almacenamiento.

Por esta razón, la ROM se utiliza para almacenar instrucciones (programas) o tablas de datos que deben estar siempre disponibles para el microprocesador.

Este tipo de memoria ROM se puede encontrar en cuatro formas distintas: La ROM estándar que está programada por el fabricante.

La memoria PROM (Programmable Read-Only Memory), se programa permanentemente por el usuario mediante equipo especial, es programada sólo una vez. La memoria EPROM (Erasable Programmable Read-Only Memory), puede ser programada y borrada por el usuario. Los datos almacenados en este tipo de memoria pueden ser borrados acercando rayos ultravioleta de alta intensidad a través de una ventana transparente especial en la parte superior del circuito integrado. La memoria EAROM (Electrically Alterable Read-Only Memory), es borrada y programada por el usuario con equipo especial, borrada eléctricamente y no con luz ultravioleta.

Una memoria para lectura - escritura (RAM). En este tipo de memoria, el almacenamiento es volátil, es decir, su contenido se pierde cuando el circuito es desenergizado. Este tipo de memoria se utiliza en las microcomputadoras para el almacenamiento temporal de los programas de usuarios y de los datos. En una memoria de acceso aleatorio cualquier posición localidad de almacenamiento puede ser escrita o leída en cierto tiempo, llamado tiempo de acceso.

La programación se conoce como escritura sobre la memoria. Copiar los datos de la memoria sin destruir su contenido se conoce como lectura de la memoria.

La ROM es una memoria permanente que contiene el programa monitor del sistema. Tiene entradas de direccionamiento junto con el "chip selector" (CS) y las líneas de entrada de lectura disponible. La ROM tiene también ocho salidas de tres estados conectadas al bus de datos*. Cada palabra, por lo tanto, es de ocho bits de amplitud.

(*) El bus de datos es una vía de dos sentidos que utiliza el microprocesador para enviar o recibir datos de la memoria.

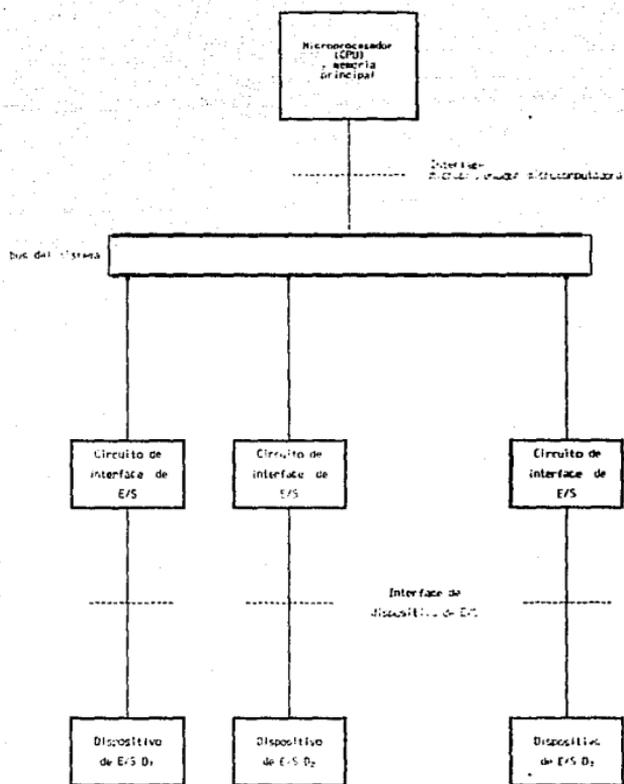
La RAM es el dispositivo de almacenamiento de lectura/escritura (R/W) temporal. Tiene entradas de dirección junto con un selector de chip (CS) y las entradas disponibles de lectura y escritura. La RAM tiene ocho salidas de tres estados conectadas al bus de datos. La RAM da entrada, salida y almacenamiento de datos como palabras de ocho bits.

INTERFACES

La mayoría de los microprocesadores tienen poco valor funcional en sí mismos, la mayoría de ellos no contienen una memoria substancial y pocos tienen puertos de entrada y salida que se conectan directamente a los dispositivos periféricos. Los microprocesadores operan como parte de un sistema. La conexión de las partes dentro de este sistema, se conoce como "interfazado". El nombre "interfaz" es un término general para especificar la frontera o punto de contacto entre dos partes de un sistema. En sistemas digitales usualmente con él se hace referencia al conjunto de puntos de conexión de señales que el sistema o cualquiera de sus componentes presenta al exterior. El verbo "interconectar" o la frase "realizar una interfaz", significa enlazar dos o más componentes o sistemas a través de sus respectivos puntos de conexión o enlace físico o lógico, de tal forma que entre ellos pueda transferirse información. En un sistema con microprocesadores existen principalmente dos tipos de interfaz: la interfaz microprocesador o microcomputadora (figura I-1), corresponde al bus del sistema, y las interfaces presentadas por los dispositivos de E/S del sistema.

La última depende de los dispositivos de E/S con que están asociadas, y varían en complejidad. Para conectar un dispositivo de E/S a un microprocesador, normalmente entre el dispositivo y el bus del sistema se incluye un circuito de interfaz de E/S. Este circuito sirve para acoplar los formatos de las señales y las características de temporización de la interfaz del microprocesador con los de interfaz del dispositivo de E/S.

El problema de interconexión de un microprocesador tiene dos aspectos principales -hardware y software-. Los problemas relacionados con la interfaz hardware consiste en relacionar los circuitos de interfaz más adecuados, interconectar los hilos adecuados y asegurar que todas las



Interfaces principales en una microcomputadora

Figura I-1

señales de interfaz tienen las características eléctricas pertinentes (nivel de señal de voltaje y corriente, etc).

El aspecto software de la interfaz implica la obtención de programas denominados programas de E/S, estos controlan la transferencia de datos hacia y procedente de los dispositivos de E/S. Este flujo de datos existe entre los dispositivos de E/S y la CPU (microprocesador). En la mayoría de los casos, sin embargo, las operaciones de E/S llevan consigo la transferencia de datos entre los dispositivos de E/S y la memoria principal. Los pasos individuales de una operación de E/S, por ejemplo, entrada o salida de una palabra de datos, son bastante sencillos, y pueden ser especificados utilizando el repertorio de instrucciones del microprocesador, tales como el 8080/8085, disponen de instrucciones, denominadas instrucciones de E/S, que tratan de facilitar la construcción de programas de E/S.

COMUNICACION ENTRE LOS MODULOS

La interacción entre la mayoría de los componentes de una microcomputadora requiere la transferencia de información sobre líneas de comunicación o buses. Un bus es un conjunto de líneas asociadas utilizadas para transferir datos digitales de un lugar a otro.

El camino principal de comunicación de una microcomputadora es el "bus" del sistema, al que están conectados la CPU, los circuitos ROM y RAM que forman la memoria principal, y los circuitos de interfaz para los dispositivos de entrada/salida. Se denomina "bus" paralelo porque los datos son transmitidos a través del "bus" del sistema con n bits a la vez, donde n es el número de bits de una palabra de datos de la memoria principal o de la CPU. Microprocesadores tales como el 6800 y el 8080/8085 tienen un tamaño de palabra de ocho bits, que se transmiten simultáneamente a través del bus del sistema.

Puesto que cada posición de memoria principal y cada puerta de E/S es potencialmente una fuente o destino de datos, por el bus del sistema deben suministrarse un gran número de direcciones. Entonces normalmente en el bus del sistema se incluye un bus específico de direcciones que puede transmitir todos los bits de una palabra de dirección. Por último, se tiene

un conjunto de líneas de control para sincronización, control de acceso al bus y señales especiales de estados tales como interrupciones.

En una microcomputadora el control del bus del sistema es responsabilidad de la CPU, con lo que la estructura del bus básicamente viene determinada por las líneas conectadas a la propia CPU. Los dispositivos de E/S toman el control del bus de sistema cedido por la CPU mediante un controlador de acceso directo a memoria DMA (Direct Memory Access), que transfiere datos directamente entre los dispositivos de E/S y la memoria principal sin la asistencia de la CPU. El DMA utiliza señales de control de acceso al bus para comunicarse con la CPU y obtener de ella el control del bus del sistema.

CAPITULO II

ESTRUCTURA DE UNA MICROCOMPUTADORA

INTRODUCCION

En el funcionamiento del microprocesador se encuentran implicados varios aspectos. Para que éste interactúe con los dispositivos periféricos, éstos deben de recibir señales del primero.

Con el avance de la tecnología electrónica la naturaleza de las señales se volvió digital, y éstas se identifican como 1 y 0, aunque en realidad varían dentro de un rango de valores mínimo. Como consecuencia de este tipo de identificación a estos valores se les denomina binarios.

La arquitectura interna de un microprocesador es muy diversa, dependiendo del tipo de microprocesador que se trate. En general, puede decirse que los dispositivos que funcionan en tal arquitectura son los siguientes: Registro de datos interno, Contador de programa, Registro de datos, Unidad aritmético-lógica (ALU), Decodificador de instrucciones, Unidad de tiempo y control y Buses.

El diagrama de estados es un diagrama que visualiza una sucesión de tareas y acciones la cual va a ser manejada por el microprocesador.

2.1 LAS SEÑALES

Los dispositivos se comunican entre sí a través de señales del sistema empleando líneas de interconexión. Estas señales son de naturaleza eléctrica y contienen información codificada.

Los primeros sistemas electrónicos eran analógicos, en los que las señales de entrada y salida, variaban de forma continua en un cierto margen de valores.

A medida que la tecnología electrónica evolucionó, los sistemas digitales aparecieron y se hicieron más comunes. En sistemas cuyos enlaces son caminos de transmisión de señales, usualmente los componentes son dispositivos que procesan o transforman información. El sistema completo puede ser contemplado como un dispositivo de procesamiento de datos. Un componente (que puede ser el sistema completo), recibe un conjunto de "señales de entrada", y el componente que envía estas señales genera un conjunto de "señales de salida", que puede transmitir a distintos componentes en tiempos determinados. El comportamiento del dispositivo se conoce sí, para cualquier señal de entrada del sistema, existe la correspondiente señal de salida producida por el dispositivo.

Una de las diferencias principales de los sistemas digitales es que las señales de cada línea, llamadas señales digitales, toman un conjunto definido de valores. Un ejemplo es una puerta lógica electrónica, en donde las tensiones de entrada y salida se definen en uno de dos márgenes: por ejemplo, un margen bajo de 0.0 a 1.0 V y un margen alto de 5.0 V.

Las señales digitales que se definen para uno de los dos márgenes de valores se llaman señales binarias o bits. Dado que las señales binarias toman uno solo de dos valores, es frecuente utilizar los números 0 y 1 para representar estos niveles. Debido a que la operación del microprocesador debe estar acompañada por la operación de los dispositivos conectados a él, se requiere que ciertas señales de control y temporización sean proporcionadas por el microprocesador a dichos dispositivos, además, el microprocesador debe estar monitoreando ciertas señales de entrada, las cuales intervienen en su operación.

Las señales están estrechamente ligadas con las funciones de cada pin en todos los microprocesadores, aunque todos difieren entre sí en su estructura física (las funciones de los

pinos, los nombres de las funciones o señales, etc).

2.2 LOS ELEMENTOS INTERNOS

(ARQUITECTURA)

A la organización de un sistema se le denomina arquitectura. Cuando un microprocesador ejecuta un programa, realiza diferentes funciones utilizando las diferentes partes de que está constituido. Los diferentes chips que existen hoy en el mercado tienen diferentes diseños y funciones, pero todos tienen típicamente al menos lo siguiente (observar la figura II-1):

1.- *Un registro de datos interno*, es una memoria de lectura-escritura. Se le conoce normalmente como el acumulador. Es un registro de propósito general de ocho bits, y es el foco de la mayoría de las instrucciones aritméticas, lógicas, de carga, de almacenamiento y de entrada/salida.

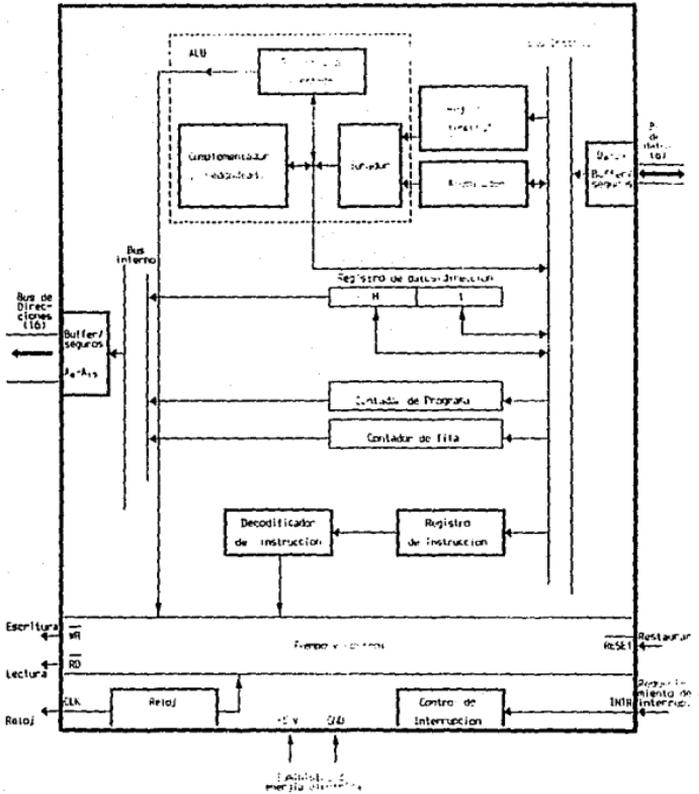
2.- *Un contador de programa*, contiene la dirección de la siguiente instrucción que será seleccionada de la memoria. Dado que las instrucciones del programa normalmente son ejecutadas en secuencia, el contador de programa casi siempre cuenta hacia arriba, a menos que se le indique lo contrario. Muchos microprocesadores comunes tienen un contador de programa de 16 bits que tendrá acceso a 64 K palabras de memoria a través del bus de direcciones.

3.- *Un registro de estado*, es una posición de memoria interna de lectura-escritura que muestra ciertos resultados de operaciones recientes.

4.- *La unidad aritmético-lógica (ALU)*, realiza operaciones aritméticas, lógicas y de rotación que afectan los registros de estado (banderas). Los resultados de la sección ALU se colocan de regreso en el acumulador a través del bus interno. El registro temporal y el acumulador son considerados muchas veces partes de la ALU. Las condiciones de banderas se alimentan de regreso a la unidad de tiempo y control.

5.- *Un decodificador de instrucciones*, toma el dato que representa a una instrucción (denominado código de operación) y lo traduce en acciones que deben realizar los diferentes

MICROPROCESADOR



Elementos de un microprocesador básico

Figura II-1

elementos internos del microprocesador, en otras palabras, interpreta el contenido del registro de instrucción, determina el programa exacto que va a seguirse en la ejecución de la instrucción en su totalidad y dirige la sección de control de acuerdo con esto.

6.- *Unidad de tiempo y control*, recibe señales del decodificador de instrucción para determinar la naturaleza de la instrucción que será ejecutada. La información del registro de estado también está disponible para la bifurcación condicional. Las señales de tiempo y de control se envían a todas las partes del microprocesador para coordinar la ejecución de las instrucciones. También se generan las señales de control externas.

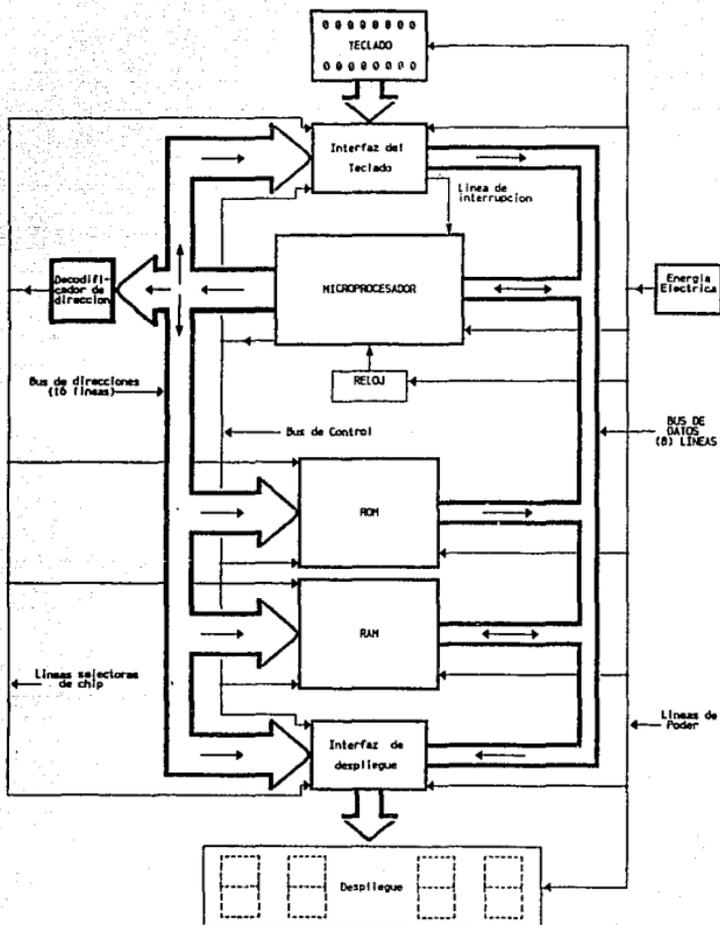
7.- *Buses*, que son conductores que transportan valores de voltaje de una posición a otra. Estos valores de voltaje son el conectado (representado por un 1) o desconectado (representado por un 0). Uno de los buses contiene el valor del dato que se está utilizando en un instante determinado y se le conoce como bus de datos. Otro contiene el valor de una dirección en la memoria del microcomputador donde se encuentra o se va a colocar el dato, y se le conoce como bus de direcciones.

La figura II-2 muestra la arquitectura de una microcomputadora. En el centro de todas las operaciones está el microprocesador. La unidad microprocesadora necesita conexiones de energía eléctrica y de reloj. El reloj puede ser un circuito separado o estar dentro del "chip" microprocesador. Un microprocesador ordinario puede tener 16 líneas de dirección que forman un bus de dirección de un sentido. También las acostumbradas ocho líneas de datos que conectan a un bus de datos de doble sentido.

La arquitectura de la microcomputadora en la figura II-2, muestra dos tipos de memoria que se utilizan en este sistema.

La ROM se utiliza para almacenar programas y otra información permanente, está diseñada para que se puedan leer los datos almacenados, pero no para reescribirla o para cambiarla durante la operación normal del sistema. El programa grabado en estas memorias es residente y no borrable. Esta memoria tiene entradas de direccionamiento, entrada de lectura y salidas al bus de datos.

También se muestra una memoria de lectura y escritura (RAM), para el almacenamiento de datos durante el funcionamiento del sistema, con sus entradas provenientes del bus de direcciones



Arquitectura de una microcomputadora

Figura II-2

y sus salidas/entradas al bus de datos.

En la figura también se observa un teclado como dispositivo de entrada. Las conexiones de energía eléctrica del teclado se muestran junto con las líneas de datos a un circuito integrado especial llamado interfaz del teclado. El circuito de interfaz almacena datos y coordina las entradas al teclado. En el momento apropiado, la interfaz interrumpe a la unidad microprocesadora por medio de la línea para tal efecto. Esta señal de interrupción ocasiona que el microprocesador:

- 1) termine la ejecución de la instrucción actual.

- 2) suspenda la operación normal.

- 3) brinque hacia un grupo especial de instrucciones en su programa monitor que maneja la entrada de datos desde el teclado.

El circuito de interfaz del teclado tiene dirección y entradas de control para activar la unidad. Cuando se activa, la interfaz del teclado coloca los datos en el bus correspondiente. El microprocesador acepta los nuevos datos de entrada a través del bus. Cuando las entradas de interfaz de tres estados no se activan, regresan a su estado de alta impedancia.

La microcomputadora diagramada utiliza como salida un grupo de despliegue de siete segmentos. Un circuito especial de interfaz de despliegue o circuito integrado se usa para almacenar los datos y manejar los despliegues. Cuando es activada, la interfaz acepta datos fuera del bus y los almacena. Luego, la interfaz maneja los despliegues en forma continua mostrando visualmente los datos almacenados en la interfaz de despliegue.

Las 16 líneas del bus de dirección puede contener 65,536 patrones diferentes de ceros y unos (2¹⁶). Las líneas de dirección del bus pueden ser conectadas a diversos dispositivos tales como RAM, ROM e interfaces. Para encender o habilitar solamente el dispositivo correcto, un decodificador de dirección muestrea los datos en el bus de dirección. La combinación lógica del decodificador de dirección activa el chip selector apropiado y de esta forma habilita el dispositivo correcto. Para simplificar el circuito, no todas las 16 líneas del bus de direcciones van hacia el decodificador de direcciones, memorias o interfaces.

La principal unidad de funcionamiento de cualquier sistema de computadora es la unidad central

de proceso (CPU).

Las principales funciones de la CPU de una microcomputadora son:

- 1.- Seleccionar, decodificar y ejecutar instrucciones de programa en el orden adecuado.
- 2.- Transferir datos hacia y desde la memoria, y hacia y desde las secciones de entrada/salida.
- 3.- Responder a las interrupciones externas.
- 4.- Proporcionar las señales de control y de tiempo necesarias para la totalidad del sistema.

2.3 EL FUNCIONAMIENTO INTERNO

LOS DIAGRAMAS DE ESTADO

Los diagramas de estado permiten conocer cuál es o será la secuencia de los pasos a seguir del microprocesador en una tarea definida por un programa.

Un estado es un tiempo en el que la memoria accesa una posición de acuerdo a la secuencia que se tenga en el programa que puede estar instalado en una memoria RAM o memoria ROM.

Un microprocesador está constituido principalmente por contadores y registros de desplazamiento, por lo que manda datos a las posiciones de la memoria de datos y a la memoria de programa, que primeramente se retiene en algún registro determinado, respetando siempre la secuencia en la ejecución del programa.

Los dos términos más comunes usados en la descripción de tiempos internos son los estados y los ciclos de máquina o ciclos Fetch. Un estado es la unidad más pequeña en la actividad secuencial y es definido como el tiempo o intervalo en un pulso positivo de reloj. Durante un estado, el microprocesador actúa con una función específica semejante al incremento en el contador de programa o transfiriendo datos hacia un registro. Una instrucción puede ejecutar muchos estados, en un rango de 10 a 20. Los estados son controlados por ciclos de máquina, los cuales intervienen cuando la memoria o los dispositivos de E/S son accedados. Sólo una dirección

es generada por ciclo de máquina, y un ciclo de máquina normalmente realiza de 3 a 6 estados. Por ejemplo, una instrucción que accesa tres veces la memoria puede tener como mínimo 3 ciclos de máquina.

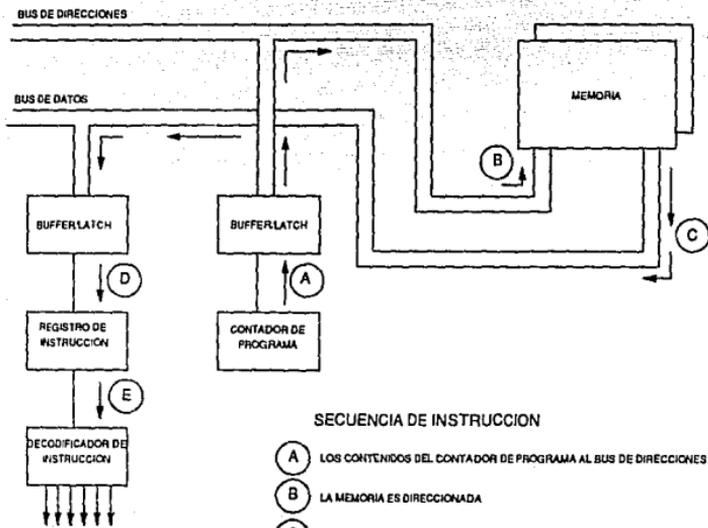
CICLO FETCH

En un programa, el ciclo de instrucciones básico es inicializado por el movimiento del contador del programa en el bus de direcciones. Este valor representa la dirección de la siguiente instrucción a ser ejecutada. Cuando este valor es puesto en el bus, permite una palabra de memoria representando la instrucción a ser leída fuera en el bus de datos. Esta operación es llamada el ciclo Fetch.

Los contenidos del registro de instrucciones variarán con cada tipo de instrucción, y son transmitidos al decodificador de instrucciones, inhabilitando la instrucción a ser ejecutada. Durante el ciclo el contador del programa es incrementado preparándose para la siguiente instrucción. La secuencia Fetch se muestra en la fig II-3. Cuando una instrucción llama a un operando para ser recuperado de la memoria, los Fetch adicionales de la memoria son requeridos. Si una operación va a ser modificada entre registros que no requieren acceso a memoria, entonces la porción de ejecución de la instrucción es más pequeña. Así, el tiempo de ejecución de una instrucción varía de acuerdo al número de tiempos que el microprocesador tiene de interfaz con la memoria. En suma, la secuencia global de instrucción puede ser una operación de 2 pasos: La adquisición de una instrucción con duración constante y la ejecución de la instrucción sobre un tiempo de proceso variable.

2.4 ESTUDIO DE UN MICROPROCESADOR DE 8 BITS (CASO GENERAL)

El hecho de que un microprocesador sea de ocho bits, indica que su bus de datos es de ocho bits, independientemente de que su bus de direcciones sea de ocho ó 16 bits.



SECUENCIA DE INSTRUCCION

- (A) LOS CONTENIDOS DEL CONTADOR DE PROGRAMA AL BUS DE DIRECCIONES
- (B) LA MEMORIA ES DIRECCIONADA
- (C) SE LEE LA INSTRUCCION Y SE COLOCA EN EL BUS DE DATOS
- (D) LA INSTRUCCION SE TRANSFIERE AL REGISTRO DE INSTRUCCIONES
- (E) LA INSTRUCCION ES DECODIFICADA

Secuencia del Ciclo Fetch

Figura 11-3

Se habla de un bus de datos y de un bus de direcciones.

Un sistema cuyo centro de control es un microprocesador, siempre va a estar conectado a una serie de dispositivos que complementan a dicho sistema. Estos elementos complementarios pueden ser: las memorias, un reloj y los interfaces.

El bus de datos es bidireccional, es decir, que siempre habrá comunicación por medio de éste hacia el exterior del microprocesador y del exterior al microprocesador. El bus de direcciones es unidireccional. Un microprocesador actual consta de 16 bits para este bus, y esto implica 2^{16} (65,536) direcciones de memoria. Este bus se conecta a todos los dispositivos externos al microprocesador (teclado, despliegue, puerto serie o paralelo, unidades de disco flexible, unidades de cinta, impresoras, etc), y de acuerdo a la instrucción que presente el estado actual del microprocesador, se activa cualquiera de los elementos anteriores.

La memoria que utiliza un sistema con microprocesador se divide en dos sectores: la memoria de programa y la memoria de datos.

La memoria de programa es aquella sección en la que se escribe un programa (que puede ser en lenguaje ensamblador), y es ahí donde se dan las instrucciones a realizar por el microprocesador.

La dirección en la memoria de programa se conecta al bus de direcciones y en esa dirección se guarda una instrucción o dato que se obtendrá por medio del bus de datos.

En forma general, para una instrucción (o dato) del microprocesador, se requiere de una dirección para alojarlo, siempre que sucede esto, existe un dispositivo interno del microprocesador que se llama "contador de programa (PC)", cuya función es incrementar en uno la dirección actual para tener otra en donde almacenar el siguiente dato.

La memoria de datos es una sección especial de la misma memoria y es el complemento de la memoria de programa, ya que esta memoria de datos se utiliza principalmente para guardar el resultado de la operación realizada, debido a una instrucción en la memoria de programa.

Generalmente todos los microprocesadores incluyen, aparte del acumulador (A) y del contador de programa (PC), registros de uso general, como lo son el apuntador de fila (SP), cuya función es la de encontrar un área de la memoria vacía y ahí almacenar un bloque de instrucciones que

proviene del programa actual por medio de una subrutina y su función es mantener los datos hasta que el programa lo decida. La fila es conocida también como memoria de primeras entradas y últimas salidas (FILO) o últimas entradas, primeras salidas (LIFO), y los registros H y L de uso general.

CAPITULO III

ESTRUCTURA DE LOS ELEMENTOS DE MEMORIA

INTRODUCCION

Un sistema procesador de datos (microcomputadora), requiere de un elemento fundamental para realizar todas sus funciones ya que estos datos deben estar latentes durante el tiempo que el usuario disponga del equipo y además pueda almacenar la información que fué procesada. Este dispositivo es la memoria.

Existen principalmente dos tipos de memorias: memoria RAM (Random Access Memory) y memoria ROM (Read Only Memory), de los cuales se derivan otros.

Una memoria básicamente está formada por celdas. La memoria RAM almacena información en estas celdas mientras el equipo esté encendido. La memoria ROM guarda la información aún después de apagar el equipo.

3.1 ESTADO ACTUAL DE LA TECNOLOGIA

La denominación integración a gran escala LSI (Large Scale Integration), se refiere a circuitos integrados que contienen entre 1,000 y 10,000 transistores. Esta densidad de componentes es suficiente como para que en un solo CI se pueda construir un sistema digital completo.

La tecnología preferida para fabricar circuitos LSI es la MOS (Metal-Oxido Semiconductor), como consecuencia de la alta densidad de componentes que se puede alcanzar. Los circuitos LSI están orientados a palabras en cuanto que las señales se agrupan típicamente en palabras que se transmiten a través de buses y se almacenan en registros. Incluso las palabras se llegan a agrupar en bloques de palabras correspondientes o conjuntos de programas o de datos; tales bloques se pueden considerar como unidades fundamentales de información al nivel LSI.

En la fabricación de computadoras y otros grandes sistemas digitales se utilizan en gran número circuitos integrados de memoria, diseñadas para almacenar bloques de información. Una memoria típica es organizada como una matriz de N bistables* de m bits, con lo que se pueden almacenar N palabras de m bits. A cada palabra almacenada en la memoria se le asigna una dirección única.

Dándole a la memoria la dirección adecuada, de donde se extrae o lee cualquier palabra que se desee, independientemente de su posición física dentro de la memoria. Esta propiedad de los circuitos de memoria se conoce como acceso aleatorio. Esto contrasta con el modo de operación acceso serie que caracteriza a los registros de desplazamiento y a las memorias no electrónicas tales como las cintas magnéticas y las unidades de disco, en las que para extraer un determinado dato, previamente hay que explorar otros datos no deseados. Hay dos tipos principales de memorias semiconductoras: ROM y RAM. Una memoria ROM (Read-Only Memory), está diseñada para que se lean los datos que almacena, pero no para reescribirla o para cambiarla, durante la operación normal del sistema. En el otro tipo de memoria, RAM (Random-Access Memory), se puede leer y escribir durante su operación normal. Las ROM se utilizan para almacenar programas y otra información permanente.

(*) Bistable indica que ama dos estados, y es el caso particular de un flip-flop.

Las RAM se utilizan como memorias temporales para almacenar resultados intermedios y datos similares no permanentes.

Utilizando tecnología LSI se ha producido una gran variedad de circuitos integrados de propósito general y propósito especial.

En la categoría de propósito general se encuentran los microprocesadores. Los circuitos de propósito especial para el procesamiento de datos que ejecutan un conjunto pequeño fijo de operaciones.

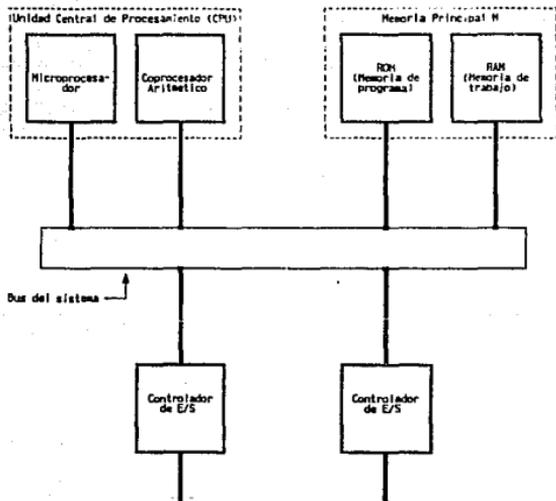
3.2 ESTRUCTURA DE LOS ELEMENTOS DE MEMORIA

Existe una gran variedad de tecnologías diferentes para almacenar información binaria, y sus características físicas varían dentro de un amplio rango. Los ejemplos incluyen circuitos electrónicos semiconductores, núcleos magnéticos de ferrita, discos magnéticos, tarjetas perforadas y papel impreso. Con frecuencia, una computadora contiene unidades de memoria que utilizan varias tecnologías diferentes, consecuencia del hecho de que ninguna tecnología reúne todas las características deseables para el almacenamiento de la información. En particular, las memorias baratas suelen resultar lentas (la velocidad de las memorias puede medirse por el tiempo requerido para realizar una operación de lectura o escritura), mientras que las memorias rápidas resultan caras. Una computadora de propósito general con la estructura de la figura III-1 puede tener los tres tipos de memoria siguientes:

1.- Una memoria pequeña pero muy rápida, constituida por entre ocho y 16 registros y que forman parte de la CPU; a veces se conoce como archivo de registros.

2.- Una memoria principal mayor, pero algo más lenta que la anterior, formada por circuitos ROM y RAM; la capacidad de la memoria se mide usualmente en Kilobytes, donde un Kilobyte, vale $2^{10} = 1,024$ palabras de 8 bits.

3.- Una memoria secundaria mayor y más lenta que las anteriores, constituida típicamente por discos magnéticos y unidades de cintas, con una capacidad medida en millones de bytes (megabytes). Las memorias secundarias están ligadas a las computadoras como dispositivos de



A los dispositivos de E/S

Figura III-1

E/S.

Los anteriores subsistemas de memoria están organizados en forma jerárquica que permite que la CPU interactúe principalmente con su archivo de registros y su memoria principal.

De cuando en cuando se transfieren bloques de información entre las memorias principal y secundaria, con lo que la CPU accede de forma indirecta a la memoria secundaria a través de la memoria principal.

En la figura III-2 se bosqueja la estructura de una memoria que utiliza el método de almacenamiento y de captación denominado acceso aleatorio. Esta memoria puede ser ROM o RAM. La mayor parte del área del CI de memoria está ocupada por una matriz de celdas de almacenamiento de 1 bit. Se utiliza un grupo de m celdas para almacenar una palabra de m bits, a la que se le asigna una combinación única de p bits siendo $2^p = N$. Las direcciones se suministran a la memoria mediante un bus de direcciones de memoria de p bits. La dirección se decodifica mediante un decodificador de 1 entre N (el decodificador de direcciones) que activa una línea seleccionada S , haciendo que se seleccione la celda correspondiente para un acceso externo. En el caso ROM, los contenidos de las m celdas seleccionadas se transfieren al bus de datos de m bits, con lo que se hacen accesibles a dispositivos externos tal como una CPU.

Una RAM requiere de una línea de control adicional denominada línea WE (write-enable: habilitación de escritura), que indica si se va a realizar una operación de lectura o de escritura con la celda direccionada. Una operación de lectura ($WE=0$) es en todo similar a la de una ROM. En una operación de escritura ($WE=1$) se transfieren los datos desde el bus externo de datos a las celdas direccionadas, sustituyendo sus contenidos anteriores. El BUS de datos de una RAM es un conjunto de m líneas direccionales; en una ROM basta un bus de datos unidireccional (de salida). Finalmente los CI ROM y RAM, tiene una línea general de habilitación denominada CS (chip-select: selección de pastilla), ésta debe estar activada para que la memoria responda a cualquier demanda de lectura y escritura.

La celda básica de la mayoría de las ROM es un simple conmutador semiconductor que está permanentemente conectado o desconectado, generando como consecuencia un bit de datos constante 1 ó 0 cada vez que es accedido. Una ROM debe ser previamente programada para fijar

Líneas de selección de celda

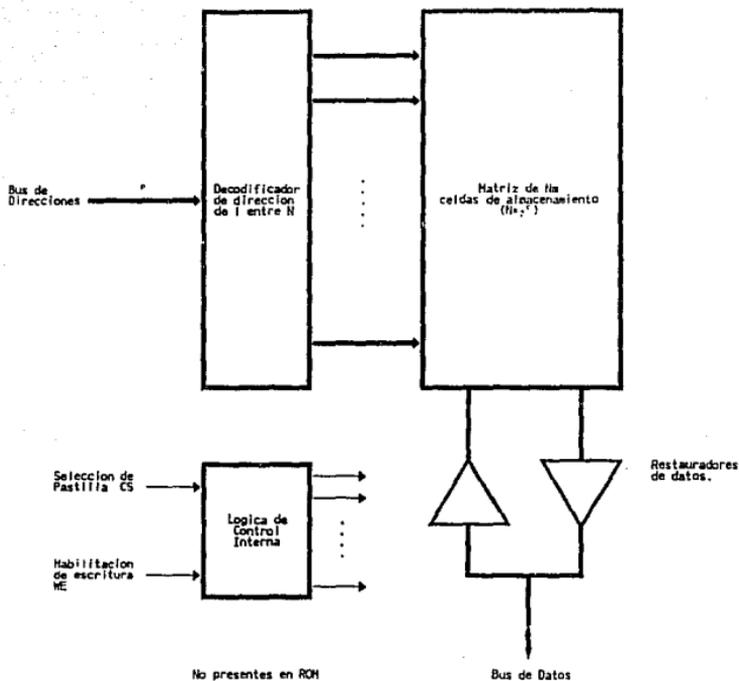


FIGURA III-2

que datos se almacenan en sus celdas de memoria. Esto se logra en el proceso de fabricación de los CI adaptando adecuadamente contactos en las celdas en las que se desee que constituyan un conmutador cerrado, y omitiendo los contactos cuando se quiera al conmutador abierto. Este proceso se conoce como programación con máscaras, pues el contenido de la ROM está determinado por la máscara que se utilice en alguna fase del proceso de fabricación. Otra clase de ROM se fabrica incluyendo un contacto fusible en cada celda, que se puede abrir o "fundir" selectivamente haciendo pasar una corriente eléctrica suficiente por ellos, por lo que estas ROM se pueden programar eléctricamente una vez que el CI se ha fabricado y empaquetado. Se observa en la figura III-3 la estructura de una ROM bipolar de este tipo.

Cada celda de almacenamiento consta de un transistor que, cuando se asigna la dirección apropiada, proporciona una señal a través de un contacto fusible a alguna línea de salida de datos D_i , esta señal vale V_{cc} , que representa el valor lógico 1, si el fusible está intacto, y vale 0 en cualquier otro caso. Los métodos de programación anteriores son irreversibles, de forma que una vez que una ROM ha sido programada, su contenido no puede cambiarse posteriormente.

Hay otra clase de memorias de solo lectura conocidas como memorias de sólo lectura y programables (EPROM o PROM), que se pueden programar muchas veces. En los tipos más comunes de PROM, un 1 o un 0 se indica por la presencia o ausencia de carga almacenada en un tipo especial de transistor MOS cuyo electrodo de puerta está encapsulado en dióxido de silicio, que es un excelente aislante.

Como consecuencia, una carga eléctrica que almacene puede mantenerse en la puerta durante un período de tiempo muy largo (muchos años), incluso a pesar de que se extraiga una pequeña cantidad de carga cada vez que se realiza una lectura. Para programar las PROM por celdas con transistores de este estilo, se aplica durante un tiempo fijo una tensión grande (30 volts o más), entre los terminales fuente y drenador de una celda descargada (se trata de transistores tipo FET -Transistor de Efecto de Campo-), de manera que aparecen electrones con la suficiente energía como para penetrar el óxido aislante y concentrarse en el electrodo puerta enterrado. Para borrar o descargar una celda se expone sencillamente la superficie de la celda a luz ultravioleta. Esto hace que temporalmente se rompa el aislamiento que hay alrededor de la puerta enterrada,

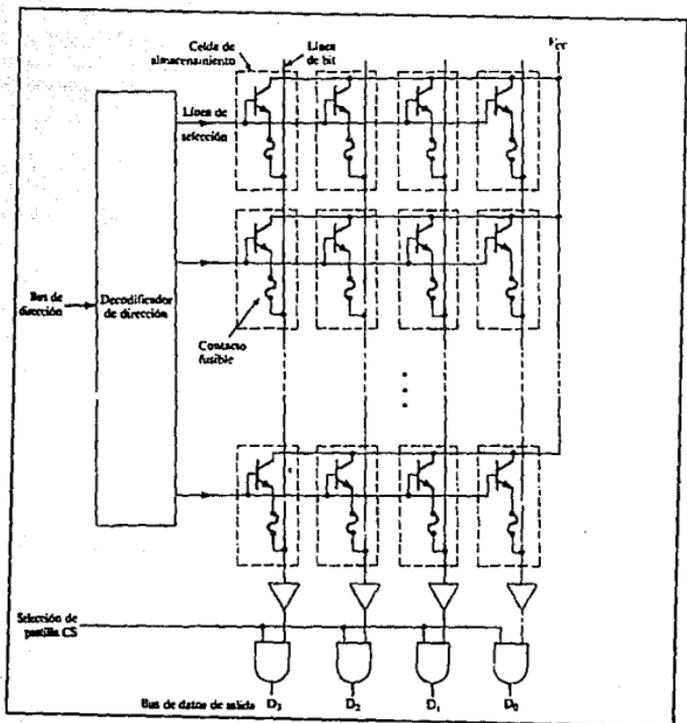


Figura III-3

permitiendo con ello que emigre la carga almacenada, con lo que la celda retorna al valor 0. Las PROM se programan cómodamente con una unidad de programación de PROM que es una unidad estándar en los sistemas de desarrollo de microprocesadores, como se puede observar en la figura III-4. Las PROM son particularmente útiles durante el desarrollo de un nuevo sistema basado en un microprocesador para almacenar el Software de control del sistema. Este software se puede cambiar o sustituir para realizar cambios en el diseño o para corregir errores.

Los circuitos utilizados para realizar las celdas RAM son más complejos que las ROM, dado que las celdas RAM deben permitir el que se escriba en ellas (programarlas), así como que se pueda leer durante su operación normal. La complejidad de las celdas RAM varía entre 1 y 6 transistores. Hay un considerable interés económico en que las celdas sean lo más simples que se pueda, resultando las memorias pequeñas, fiables y baratas. A partir de 1980 empezaron a producirse de forma masiva circuitos integrados RAM con una capacidad de almacenamiento de 64K bits (2^{16}) o más. En la figura III-5 se tiene una celda RAM con seis transistores, que es básicamente un biestable sin reloj formado por dos inversores MOS acoplados entre sí. Los datos almacenados están representados por el par de señales complementarias Q y \bar{Q} ($Q=1$ y $\bar{Q}=0$). Para minimizar el número de interconexiones externas de la celda, lo que constituye una importante consideración en diseño LSI, se utiliza un par de líneas bidireccionales B_0 y B_1 , conocidas como líneas de bit (o bit/sense), para transferir datos de o hacia la celda. Se utiliza una única línea de select S (también conocida como línea de palabra) para seleccionar la celda tanto para lectura como para escritura. Cada par de transistores T_1 , T_{R1} y T_2 , T_{R2} forman un inversor nMOS con T_{R1} y T_{R2} actuando como resistencias. Los otros dos transistores T_3 y T_4 se utilizan como conmutadores de E/S para posibilitar la lectura o la escritura de las líneas de bit.

Cuando no va a ser accedida, la celda RAM tiene su línea select S al valor 0. En consecuencia, los transistores T_3 y T_4 no conducen, desconectando efectivamente las celdas de las líneas de bit. Las líneas de bit propiamente dichas se mantienen normalmente al valor lógico 1 mediante resistencias externas de desplazamiento de nivel.

En una operación de lectura, la celda se selecciona poniendo S al valor 1, lo que pone a T_3 y T_4 en conducción y con ello se conectan las señales almacenadas Q y \bar{Q} a las líneas de bit B_0 y

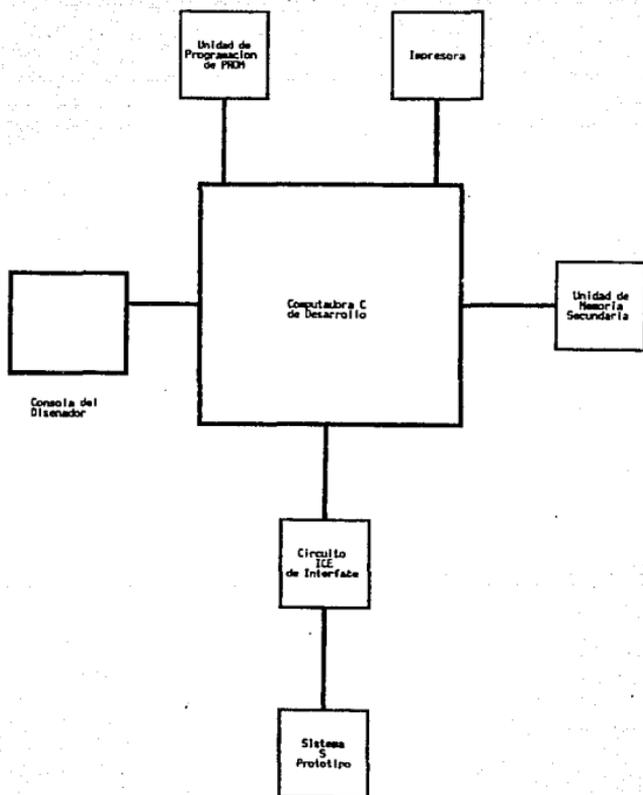


Figura III-4

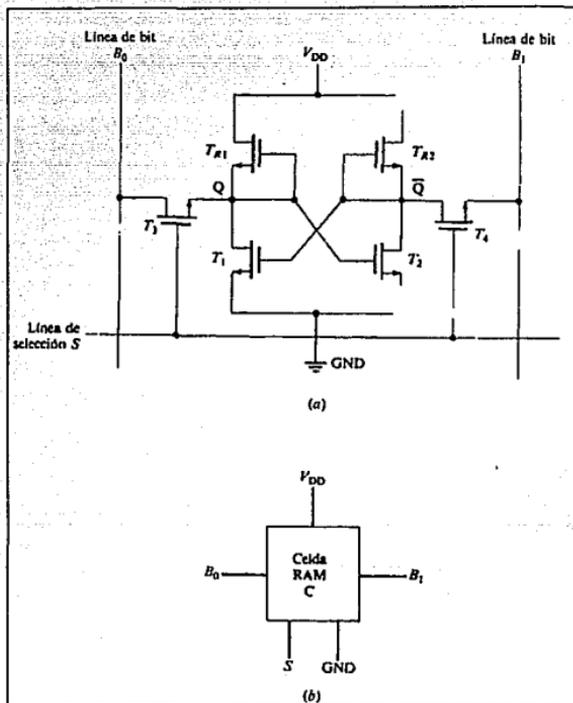


Figura III-5

B_1 , respectivamente. Si $Q=1$, o sea que la celda almacena un 1, entonces Q desplaza a B_1 de su estado normal 1 (débil) al estado 0.

El pulso 0 resultante en B_1 es procesado por un amplificador externo denominado amplificador de lectura o de sentido, lo que produce una señal de salida de valor 1 a la línea externa de datos apropiada de la RAM. Como $Q=B_0=1$, no hay cambio en la línea B_0 . De forma similar, cuando la celda almacena un 0, $\bar{Q}=0$, Q produce una señal 0 en la línea B_0 , mientras que B_1 permanece inalterado. Por tanto, controlando las señales en cualquiera de las líneas de bit, B_0 por ejemplo, se puede leer el estado de la celda. La celda no afecta a su estado, y consecuentemente este tipo de celdas se conoce como de lectura no destructiva.

Para escribir nuevos datos en la celda RAM se requieren circuitos de control que pueden forzar temporalmente las líneas de bit al valor 0, (si se quiere escribir un 1 en una celda que actualmente almacena un 0). Como en una operación de lectura S se pone a 1 haciendo que T_2 y T_1 conduzcan; entonces a B_1 se aplica un 0 (fuerte), mientras que a B_0 se aplica un 1 (débil). Esta combinación de entradas hace que el biestable de almacenamiento cambie de estado (T_2 pasa a conducción y T_1 deja de conducir), dando origen al nuevo estado $Q=B_0=1$ y $\bar{Q}=B_1=0$. Este estado permanece en la celda cuando la línea de selección vuelve a 0. Si la celda almacenaba ya un 1, entonces al escribir un 1 no afecta a la celda. Para escribir un 0 en la celda, S se pone a 1 y las líneas de bit B_0 y B_1 se fuerzan a 0 y a 1, respectivamente.

En la figura III-6 se muestra como se conectan las celdas de almacenamiento para construir una RAM completa. Se utiliza lógica triestado para permitir que el mismo conjunto de líneas externas sirvan tanto de líneas de entrada como de salida. Durante una operación de escritura, se necesita que $WE=1$, los amplificadores de lectura están deshabilitados. Esto hace que las líneas de salida asuman el estado Z de alta impedancia, desconectándolas del bus de datos. Durante una operación de lectura, $WE=0$, los amplificadores triestado de escritura están deshabilitados, de forma que no tienen efecto alguno sobre las líneas de bits.

El tipo anterior de RAM se denomina estática porque, salvo cuando se escriben nuevos datos en la memoria, los datos almacenados se mantienen sin alteraciones por un tiempo indefinido.

Hay otra clase de RAM conocidas como dinámicas, que tienen la propiedad de que los datos

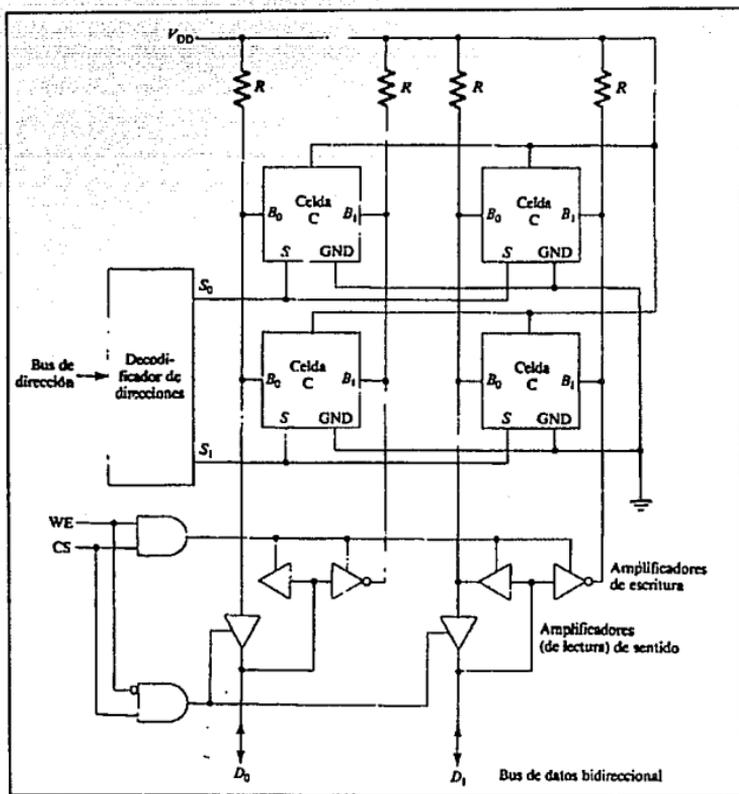


Figura III-6

almacenados decaen o se desvanecen espontáneamente y deben ser restaurados en intervalos regulares (refresco). Una celda se refresca simplemente ejecutando una operación de lectura de esa celda; el ciclo de lectura incluye una reescritura automática que restaura cualquier valor de carga máxima que haya. Las RAM dinámicas se refrescan mediante un circuito de control externo que genere secuencialmente todas las direcciones, especificando una operación de lectura para cada dirección; los datos que vayan apareciendo en el bus de datos de la RAM se ignoran. En un sistema basado en un microprocesador, el refresco se realiza frecuentemente mientras la RAM está inactiva, y por tanto el impacto de esta operación es bastante pequeño. Las RAM estáticas no requieren refresco, por lo que son de utilización más sencilla. Sin embargo, el que sus celdas de almacenamiento sean más complejas, implica que en un CI se pueden colocar menos celdas de almacenamiento para un tamaño y una tecnología dadas. En consecuencia, las pastillas RAM con mayor capacidad de almacenamiento utilizan usualmente circuitos dinámicos en lugar de estáticos.

Una celda de almacenamiento dinámico puede construirse con un capacitor y un transistor. El capacitor C es el elemento de almacenamiento, mientras que el transistor T actúa como un conmutador para cargar y descargar C a través de una única línea de bit B. Se utiliza una línea de selección S del decodificador de direcciones para conmutar T de corte a conducción y viceversa, con lo que C se carga o descarga a voluntad.

Esto se muestra en la figura III-7.

Para escribir un bit de datos D en una celda RAM dinámica, se aplica D a la línea de bit B. Entonces se pone a 1 la línea de selección S, haciendo que C se cargue si $D=1$, o se descargue si $D=0$.

Por tanto, se almacena el valor de D en la forma de carga cero o de carga máxima en C. Para leer el contenido de la celda, B se convierte en una línea de salida, y de nuevo se selecciona la celda poniendo a S al valor 1. Si C está cargado, almacenando un 1, entonces se descarga a través de la línea B, produciendo un pulso de corriente que es detectado por un amplificador de salida, y haciendo que aparezca un 1 en la línea de salida de datos correspondiente de la RAM. Si C está descargado antes de que se inicie la lectura, en B no se produce pulso de corriente,

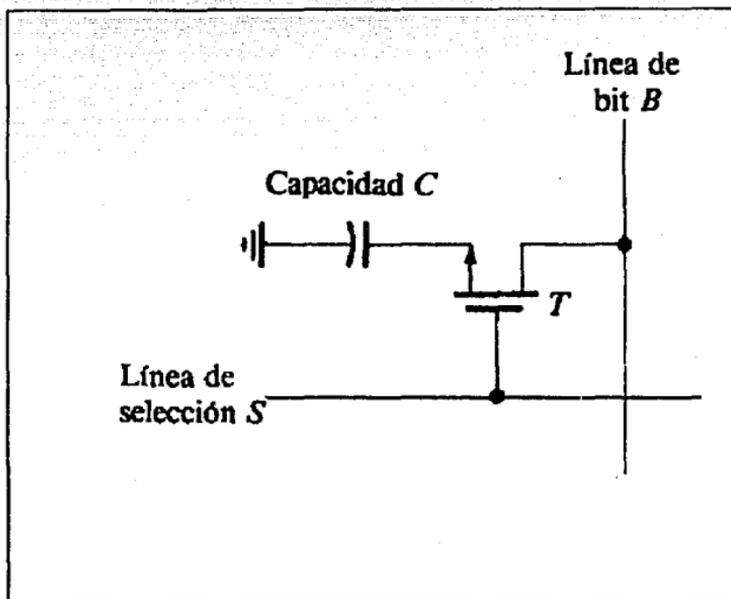


Figura III-7

generándose un cero en la salida de datos. La lectura es destructiva, dado que al leer una celda supone dejarla en el estado 0. Si la celda almacenaba originalmente un 1, la operación de lectura debe ir seguida inmediatamente de una escritura de un 1, restaurando el estado original. Esta etapa extra de escritura la realizan automáticamente los circuitos de control de la RAM, y por tanto esta cualidad de ser destructivo el proceso de lectura queda oculta al usuario.

3.3 ESTUDIO DETALLADO DE VARIOS CASOS (RAM, ROM)

MEMORIA RAM NMC2148

Esta memoria RAM es del tipo estático de la familia NMOS.

Tiene una capacidad de 4,096 bits, es decir, 1,024 palabras de 4 bits.

Anteriormente se comentó que las RAM están organizadas como matrices de N por m bits, donde N es el número de posiciones de almacenamiento de palabras direccionables y m el tamaño de las palabras, por lo tanto, esta memoria tiene 1,024 posiciones disponibles o direcciones de almacenamiento para palabras de 4 bits. Obsérvese la figura III-8.

Esta memoria se presenta en un paquete CI de 18 pines.

Los pines 9 y 18 son de polarización (5V ∞), los pines 1-7, y 15-17, son las entradas de direccionamiento, con las que se habilita una localidad determinada. Estos pines por lo general se conectan a el bus de direcciones del microprocesador. Los pines 11 a 14 son las de control de I/O (Input/Output), son bidireccionales y por medio de estas se forman palabras de 4.

Los pines 8 (Chip Select CS) y 10 (Write Enable WE), determinan si la memoria ha de leer o escribir, basándose en la tabla de verdad correspondiente que se presenta en la figura.

MEMORIA RAM NMC6164

Esta memoria pertenece a la familia CMOS y es capaz de almacenar 8,192 palabras de 8 bits cada una. Su funcionamiento se describe de acuerdo a la figura III-9.

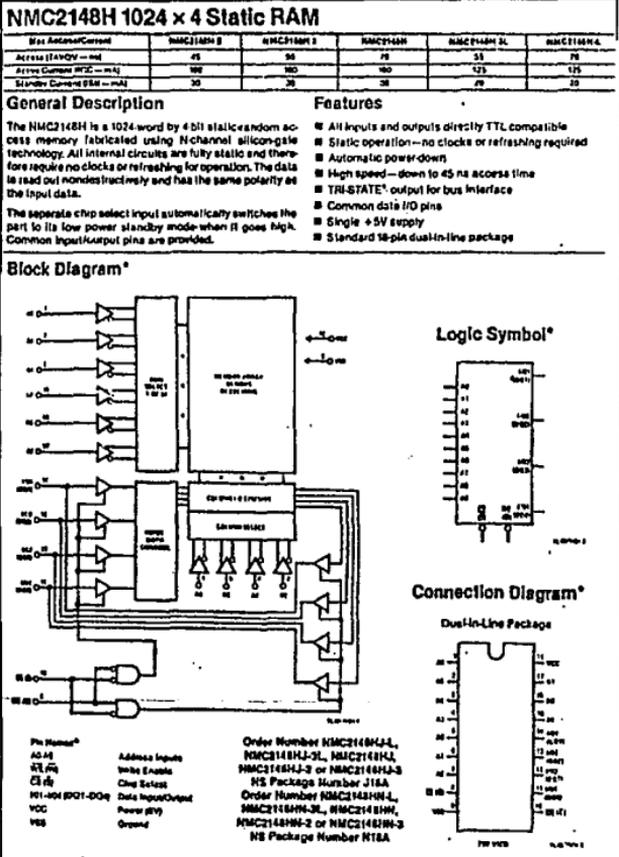


Figura III-8

La memoria se presenta en una pastilla CI de 28 pines.

El pin 1 indica un NC (No Conexión), los pines 14 y 28 son de polarización, (5 V). Los pines A₀ a A₁₂ (13 pines), son las entradas para habilitar las direcciones, es decir, es el arreglo de celdillas que permite acceder 2¹³ (8,192) localidades de memoria.

Los pines 11, 12, 13, 15-19, son las entradas de datos que provienen del microprocesador, o en su defecto por un arreglo microcontrolado, las cuales son bidireccionales, es decir, que a través de ellas se puede leer o escribir (I/O). Nótese que son 8 pines, por cada pin existe un bit, por lo que el conjunto de 8 pines es un byte, que en un momento dado se almacena en una localidad de la misma memoria.

Los pines 20 (CS₁), 22 (OE), 26 (CS₂) y 27 (WE) funcionan de acuerdo a la tabla de verdad que se muestra en la misma figura III-10 en la que se observa como disponer a la memoria para lectura, escritura o para deshabilitación.

MEMORIA EPROM MM2716

Obsérvese la figura III-10, en la que se presenta a la memoria en su paquete CI de 24 pines y su tabla de verdad. Esta memoria ROM es CMOS, y tiene la cualidad de ser programada, borrada y ser reprogramada cuantas veces sea necesario, pero no durante su operación, requiriéndose un equipo especial.

El CI 2716 es capaz de almacenar palabras de 8 bits en 2,048 direcciones (2¹¹) disponibles, por lo que se tienen 11 pines para direccionamiento de memoria (1-8, 19, 22 y 23). Esta memoria a diferencia de la RAM, sólo permite lectura de datos. Cuando la memoria accesa una localidad determinada, y se habilita el pin 20 (Output Enable OE), con un 0 siempre, los pines 9-11, 13-17 revelan el dato guardado en esa dirección. Estos pines de salida (Output), se conectan al bus de datos del microprocesador. Por medio de el pin 18 (Chip Enable CE) se habilita con un 0 para permitir lectura de datos (complemento de OE), y con un 1 para deshabilitación. El pin 21 (Vpp) se utiliza para el proceso de programación de la memoria; para la EPROM 2716 (National), se requieren 25V CD, polarización que cambia de acuerdo al tipo de memoria y

MM2716 16,384-Bit (2048 x 8) UV Erasable PROM

Part Number/Order Number	MM2716	MM2716-1	MM2716E
Active Time (ns)	450	500	450
V _{CC} Power Supply	5V ± 5%	5V ± 10%	5V ± 5%

General Description

The MM2716 is a high speed 16K UV erasable and electrically reprogrammable EPROM, ideally suited for applications where fast turnaround and pattern experimentation are important requirements.

The MM2716 is packaged in a 24-pin dual in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit patterns. A new pattern can then be written into the device by following the programming procedure.

This EPROM is fabricated with the reliable, high volume, time proven, N-channel MOS silicon gate technology.

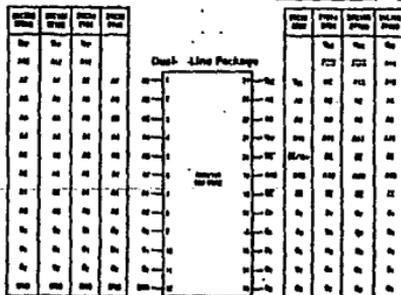
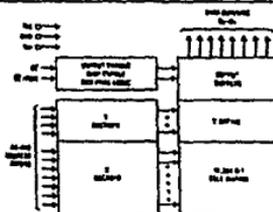
Features

- Access time down to 350 ns
- Low power consumption
 - Active power: 825 mW max
 - Standby power: 132 mW max (75% savings)
- Single 5V power supply
- Extended temperature range available (MM2716E), -40°C to +85°C, 450 ns ± 5% power supply
- Pin compatible to Matsushita's higher density EPROMs
- Static—no clock required
- TTL compatible inputs/outputs
- Tri-State™ output

Block and Connection Diagrams

Pin Names

Pin Name	Address
A0-A14	Addresses
CE	Chip Enable
OE	Output Enable
Q ₀ -Q ₇	Outputs
PGM	Program
NC	No Connect



MS Package Number J24A-G

Note: Matsushita's vertical compatible EPROM pin configurations are shown in the block subjected to the MM2716 pin.

Figura III-10

compañía que la fabrique. Cuando la memoria ya ha sido programada, es necesario que, cuando está funcionando dentro de un sistema dado, se polarice a 5V CD siempre.

CAPITULO IV

INTRODUCCION A LA PROGRAMACION DEL MICROPROCESADOR

INTRODUCCION

Los microprocesadores por sí solos no pueden generar función alguna, requieren de una arquitectura confiable y de programación.

La arquitectura en los microprocesadores principalmente está formada por dispositivos electrónicos como memorias, registros, generadores de pulsos, contadores, y líneas de conexión o buses.

La programación en los microprocesadores varía según el modelo, la versión y la marca. Todos los microprocesadores contemplan instrucciones aritméticas, lógicas, de transferencia de datos, de bifurcación, de llamada de subrutina, entre otras, y de acuerdo a la versión, es la velocidad de ejecución de estos.

4.1 DIRECCIONAMIENTO DE MEMORIA Y E/S

MODOS DE DIRECCIONAMIENTO.

Una instrucción especifica una operación y un conjunto de operandos. La operación está determinada por un código de operación, que usualmente es uno de los campos más a la izquierda del formato de la instrucción, tanto en lenguaje máquina como en lenguaje ensamblador. El número de operaciones básicas que una computadora puede realizar es relativamente pequeño. Sin embargo, el número de posibles instrucciones se incrementa enormemente cuando se tienen en cuenta los modos de direccionamiento de los operandos y los tipos de datos que llegan a utilizarse con cada código de operación.

En instrucciones en lenguaje máquina, el código de operación especifica todos los modos de direccionamiento y tipos de datos utilizados. Decodificando el código de operación, la CPU puede concluir cómo obtener los operandos y qué acciones realizar con los mismos. En el lenguaje ensamblador, los mnemónicos de los códigos de operación definen los tipos de operaciones básicas.

Considerando la instrucción "suma" del microprocesador, uno de los dos números que serán sumados se localiza en el acumulador.

Pero ¿de dónde viene el segundo número y cómo se consulta?, Los diversos métodos de consulta son conocidos como modos de direccionamiento del microprocesador. Cada microprocesador tiene sus propios métodos de consulta o modos de direccionamiento lo que deben ser bien entendidos por el programador.

Los modos de direccionamiento listados en el conjunto de instrucciones de las tablas anteriores son:

- 1.- Inherente
- 2.- De registro
- 3.- Inmediato
- 4.- Directo

5.- Indirecto de registro

Los primeros dos modos de direccionamiento de la lista (inherente y de registro) tienen que ver con los operandos que se localizan dentro de la CPU. Los últimos tres modos de direccionamiento (Inmediato, Directo e Indirecto de registro) tienen que ver con los operandos que se localizan fuera de la CPU, en posiciones de memoria o puertos de E/S.

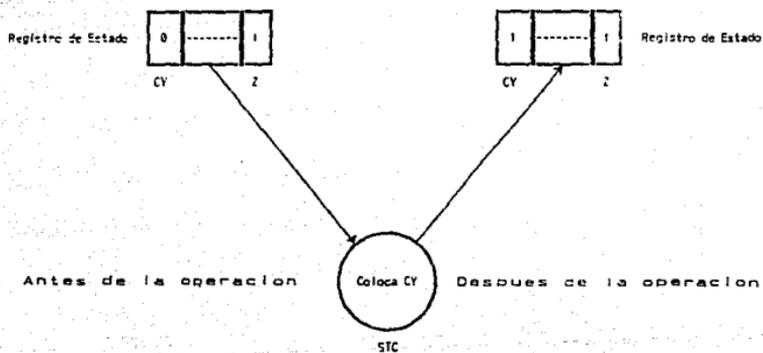
Las instrucciones inherentes son siempre instrucciones de 1 byte porque no se requieren mayores datos para completar la operación. Todas las acciones ocurren en la CPU. La instrucción inherente "coloca bandera de acarreo" (STC) se toma como ejemplo y se diagrama en la figura IV-1. No necesita consultar datos o direcciones de otros registros de la CPU, posiciones de memoria o puertos de E/S. La instrucción STC coloca la bandera de acarreo en 1 (CY=1), sin afectar otros registros o banderas.

En las instrucciones de registro el operando se consulta de un registro interno de la CPU. La operación "resta L de A" (SUB L) es una instrucción de este tipo. Las instrucciones de registro (como las instrucciones inherentes) son siempre instrucciones de 1 byte, dado que no se requieren más datos o direcciones de fuera de la CPU. Todas las acciones ocurren dentro de la CPU. Para la instrucción SUB L, el minuendo se puede almacenar en el acumulador y el substraendo en el registro L (o viceversa). Después de la operación interna de resta, la diferencia se transfiere al acumulador.

En las instrucciones inmediatas, el operando viene del siguiente byte (o a veces de los siguientes dos bytes) de la memoria de programa. El operando continúa inmediatamente después del código de operación de la instrucción. Por ejemplo, la instrucción "carga SP con datos" (LXI SP), del grupo de instrucciones de transferencia de datos es de este tipo; esta es una instrucción de 3 bytes. Generalmente las operaciones inmediatas se especifican con dos o tres bytes. La instrucción LXI SP utiliza dos bytes que se consultan de la memoria de programa y se cargan al apuntador de fila (SP). El tercer byte es el código de operación.

Las instrucciones de modo inmediato son convenientes para cargar los valores iniciales a los registros de la CPU o al apuntador de fila.

En las instrucciones directas, el segundo y tercer byte de memoria de programa apuntan



Colocación de la bandera de acarreo

Figura IV-1

directamente a la dirección del operando en la memoria de datos. El segundo y tercer byte de la memoria de programa son direccionados en una instrucción directa, mientras que son operandos en direccionamiento inmediato. La operación "entra a A del puerto de posición LOC a" (IN), del grupo de transferencia de datos es de este tipo. Es una instrucción de dos bytes. Las operaciones directas se especifican con dos o tres bytes. En la instrucción IN los contenidos del puerto de entrada se transfieren al acumulador. El primer byte es su código de operación y el segundo byte apunta una dirección en el programa, de donde se sacará el dato a ser transferido al registro A.

En las instrucciones indirectas de registro el par de registro especial HL apunta a la dirección del operando en la memoria de datos. La instrucción "carga LOC (H y L) a A" (MOV A,M), del grupo de instrucciones de transferencia de datos es de este tipo; es una instrucción de 1 byte. Las instrucciones indirectas de registro siempre son de 1 byte. El par de registros HL conocido como registro de datos/dirección en el microprocesador, apunta una dirección de la memoria de datos y el dato que está almacenado en esa dirección se transfiere por medio de esta instrucción al acumulador. Otros microprocesadores tienen variaciones en el modo de direccionamiento directo conocido con nombres tales como página cero o direccionamiento de página de base, direccionamiento absoluto, direccionamiento de página presente y direccionamiento extendido.

En la tabla III del capítulo V, se observan dos operaciones de transferencia de datos IN y OUT. Por medio de la instrucción IN se puede tener comunicación de un módulo externo a la unidad de proceso. Primeramente la memoria de programa accesa una dirección de 8 bits que es donde se localiza la fuente de los datos que serán transferidos y que es identificado como un puerto de entrada. Al ejecutar esta instrucción, los datos entran al microprocesador y se almacenan en el acumulador.

La instrucción OUT funciona es el sentido contrario de la instrucción IN. Aquí el acumulador es la fuente de los datos y el destino es un puerto de salida que direcciona la memoria de programa. Estas dos instrucciones se utilizan para tener comunicación con dispositivos periféricos y la CPU.

4.2 BANDERAS Y SALTOS.

La Unidad aritmético-Lógico (ALU) contiene al registro de estado que también se denomina registro de código de condición o banderas y es realmente un grupo de multivibradores (MVB), individuales que pueden ser establecidos o restaurados con base en las condiciones creadas por la última operación de la ALU.

El registro de códigos de condición (o de estados), almacena la información que describe los resultados de operaciones anteriores e información de sí, o no, tendrá el procesador conocimiento de una petición de interrupción enmascarable. Esta información se almacena asignando un bit en el registro a cada elemento de información.

Los bits individuales del registro de estado se conocen como bits de control o indicadores de estado, dependiendo de su función. Cada bit, como todos los bits de la memoria del microcomputador, tiene dos estados, el estado 1 y el estado 0.

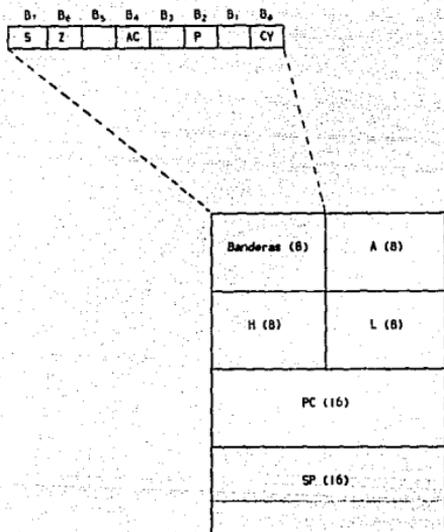
Los MBV individuales o banderas, incluyen indicadores de 0 (cero), resultado negativo, acarreo del bit más significativo (MSB), etc. Las banderas sirven para la toma de decisiones cuando se utilizan instrucciones subsecuentes de bifurcación.

En la figura IV-2 se presenta un resumen de los registros disponibles en el microprocesador 8080/8085. Aquí se observa el registro de estado de 8 bits que contiene las banderas. En la visión expandida del registro de estado, una bandera de acarreo (CY) se muestra en la posición B₇, mientras que la bandera 0 (Z) se localiza en la posición B₀. El 8080/8085 suministra cinco banderas.

Estas banderas dirigen la salida de la ALU y cada una sirve para almacenar un bit de salida de la ALU, que puede ser necesario en el desarrollo posterior del programa.

Las cinco banderas de este microprocesador son:

- 1.- La bandera de acarreo (CY), se coloca o se restaura mediante operaciones aritméticas. Su estado es probado por medio de instrucciones de programa. La sobresaturación de una suma de ocho bits hará que la bandera de acarreo sea colocada en 1. En la resta la bandera de acarreo actúa como una bandera de "prestado". La colocación de la bandera indica que el minuendo es



Registro de propósito general del microprocesador 8080/8085 y su registro de estado (banderas)

Figura IV-2

menor que el sustraendo.

2.- La bandera de cero (Z), se coloca si el resultado de ciertas instrucciones dé 0. La bandera de 0 se blanquea si el resultado no es 0.

3.- La bandera de signo (S), se coloca en la condición del bit más significativo del acumulador que sigue a la ejecución de las instrucciones aritméticas o lógicas. Estas instrucciones utilizan el bit más significativo de datos para representar el signo del número contenido en el acumulador. Una bandera de signo colocado (uno), representa un número negativo, mientras que una bandera restaurada (cero), significa un número positivo.

4.- La bandera auxiliar de acarreo (AC), indica una sobresaturación o el acarreo del bit 3 del acumulador, de la misma manera que una bandera de acarreo indica una sobresaturación desde el bit 7. Esta bandera es comúnmente utilizada en la aritmética BCD (Decimal Codificado en Binario).

5.- La bandera de paridad (P), prueba el número de bits 1 en el acumulador. Si el acumulador almacena un número par de unos, se dice que existe paridad par y la bandera de paridad se coloca en 1, sin embargo, si el acumulador almacena un número non de unos (llamado paridad non), la bandera de paridad del 8080/8085 se coloca en 0. Por ejemplo, si una instrucción ADD deja en el acumulador el resultado 00110011₂ la bandera de paridad se colocará en 1 porque existe un número par (4) de unos. Si la suma fuera 10101110₂ la bandera P sería restaurada a 0 porque existe un número non (5) de unos en el acumulador.

Los saltos van de acuerdo a la lógica de un programa. Cuando se tiene un diagrama de estados, derivado de un diagrama de flujo, se presentan bloques de condición, en donde sólo se tienen dos opciones: sí o no, que en realidad es la lógica de unos y ceros, que son los que habilitarán o deshabilitarán las banderas mencionadas anteriormente.

Las instrucciones de bifurcación para el microprocesador 8080/8085 se presentan en la tabla-IV (capítulo V). Estas operaciones afectan la ejecución en la secuencia de un programa. Su modo de direccionamiento es inmediato, por lo que, cuando en la ejecución de un programa se encuentra cualquiera de estas cinco instrucciones, alteran al contador de programa (PC). El contador de programa generalmente está secuenciado hacia arriba un número a la vez, pero con

una instrucción de brinco, el programa salta de ese bloque actual a otro sin considerar al PC.

Otro tipo de salto es que presentan las instrucciones de subrutina. La subrutina es un bloque independiente de la secuencia de un programa, y también altera la secuencia del PC, pero a diferencia de las instrucciones de bifurcación, al terminar la subrutina, se regresa a la dirección inmediata del programa principal, de donde se llamó esta instrucción.

4.3 CONTEO Y LAZOS DE TIEMPO

La CPU tiene el control global de la computadora, y es responsable de la captación, interpretación y ejecución de secuencias de instrucciones, es decir, de programas de computadora.

La acción de captar y decodificar las instrucciones la realiza una parte de la CPU denominada unidad I (unidad de instrucciones). Las instrucciones son ejecutadas por la unidad E (unidad de ejecución) de la CPU que contiene la ALU y un conjunto de registros de trabajo y auxiliares. La memoria principal M almacena instrucciones y datos a procesar por la CPU.

La unidad I es responsable de llevar la cuenta de las posiciones o direcciones de las instrucciones en M necesarias en cada instante para la CPU. Las instrucciones se captan y se ejecutan normalmente según la secuencia con la que se almacenaron en M. La unidad I contiene un registro especial (realmente es un contador), llamado contador del programa o PC (program counter), que contiene la dirección de la siguiente instrucción, o porción de instrucción, que requiere la CPU. Tras cada operación de captación de instrucción, el PC se incrementa automáticamente para construir la dirección de memoria requerida para captar la siguiente instrucción.

Una instrucción que representa un tiempo en un programa es la que se muestra en la tabla V (capítulo V), de operaciones varias, y que es la instrucción de no operación, que sólo hace que el PC se incremente.

Esta operación se utiliza para consumir tiempo en aquellas secciones de los programas en que sea necesario.

Como un ejemplo se puede suponer que se quieren transmitir caracteres alfanuméricos a una

máquina de escribir electromecánica. Cada tecla de la máquina debe encajar con el hardware de la figura IV-3. Para escribir un caracter se debe usar una instrucción OUT, donde el byte de la instrucción contiene la dirección de la tecla que va a ser golpeada, (si la máquina de escribir responde al código ASCII, los bytes de dirección son simplemente los códigos ASCII de los caracteres que se van a imprimir). La instrucción OUT pone en set al cerrojo, éste activa un electroimán, que a su vez empuja la tecla de escritura. Se debe mantener activado el electroimán el tiempo suficiente para que la relativamente lenta tecla mecánica complete su respuesta. Se puede emplear una segunda instrucción IN para poner en reset el cerrojo y librar la tecla.

Puede ser necesario también mantener en set el cerrojo durante muchas decenas de milisegundos, un tiempo que es muy grande comparado con la duración de un ciclo de reloj y con el tiempo requerido para buscar y ejecutar una instrucción. (Tiempo de ciclo de reloj para el 8080 está en el rango de 0.5 a 2.0 μ s, y aún la instrucción más complicada requiere solamente 18 ciclos de reloj).

Por lo que se debe interponer un programa entre las instrucciones OUT e IN, cuyo único propósito sea permitir el paso del tiempo.

Estos programas de retraso de tiempo se utilizan frecuentemente en microprocesadores con el equipo periférico apropiado asociado que generalmente aparecen como subrutinas.

Así un programa típico llamado durante la operación de una tecla de máquina de escribir sería:

ROTULO	CONTENIDO	COMENTARIO
	OUT	Pone en SET al cerrojo al pulsar la tecla de dirección.
	CALL [TIMING]	Llama incondicionalmente a la subrutina que comienza en la dirección rotulada con la bandera "TIMING".
RELEASE	IN	Pone en RESET el cerrojo al liberar la tecla.

En una rutina de tiempo se comienza por cargar un número en un registro. Entonces se decrementa el registro. La instrucción de decrementación es seguida por una instrucción de salto condicional.

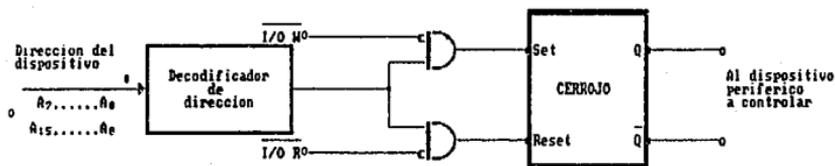


Figura IV-3

Si una vez decrementado el registro no está en cero, se regresa a la instrucción de decrementación (volver atrás), pero cuando el registro lee cero avanza a la siguiente instrucción que es de RETURN incondicional al programa principal "RELEASE". El tiempo de retardo se determina por el número colocado en el registro y la velocidad de reloj. Si el tiempo de retardo determinado por un único registro no fuera lo bastante grande, se pueden usar dos registros. El segundo registro se decrementaría en 1 cada vez que el primer registro completa un ciclo de decrementación desde su contenido inicial a cero. Así, el número total de operaciones de decrementación necesarias para borrar ambos registros depende del producto (no la suma) de los dos registros. Solamente cuando los dos registros se borran se termina con el lazo de subrutina. Un diagrama de flujo para la subrutina se muestra en la figura IV-4. Aquí se supone que los registros involucrados son los registros D y E.

El programa de subrutina es como sigue:

ROTULO	CONTENIDO	COMENTARIO
TIMING	MVI E NE	Mueve el registro NE al registro E.
LOAD	MVI D	Mueve el registro ND al registro D.
WAIT	DCR D	Decrementa el registro D.
	JNZ [WAIT]	Si el registro D no es cero, bifurca a la instrucción de decrementación en la dirección indicada con la bandera "WAIT". En otro caso continúa a la siguiente instrucción:
	DCR E	Decrementa el registro E.
	JNZ [LOAD]	Si el registro E no es cero, salta a la instrucción en la dirección indicada con la bandera "LOAD" para poner ND en el registro D.
	RET	Vuelve incondicionalmente al programa principal.

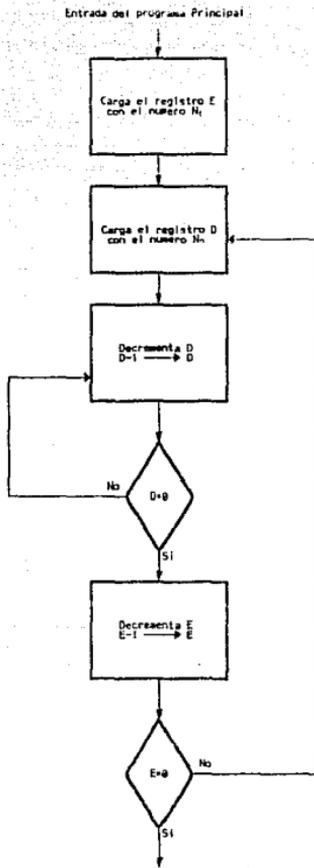


Diagrama de Flujo para una subrutina de tiempo

Figura IV-4

4.4 FUNCIONES ARITMETICAS

En la tabla I se listan las instrucciones de Suma, Resta, Decrementa, Incrementa y Compara, en las que se observa que el registro acumulador es el encargado de realizarlas, además del par de registros HL. Estos registros se encuentran dentro de la ALU y se definen como circuitos de conmutación de almacenamiento temporal a alta velocidad, ideados para realizar las operaciones antes mencionadas.

En la figura IV-5 aparecen dos circuitos ALU de la serie 74xx.

Estos CI multifunción están diseñados para procesar palabras de 4 bits que representen datos tanto numéricos como lógicos. El 74181 puede realizar 16 funciones aritméticas, incluyendo la suma, la resta, la comparación de valores, y 16 operaciones digitales, incluyendo las operaciones de compuertas para palabras AND, OR, NAND, NOR y OR-EXCLUSIVA. Un bus de control de 5 bits S selecciona la operación a realizar en cada instante por el CI.

El 74381 es una versión simplificada del 74181. Tiene un repertorio de 8 operaciones aritmético-lógicas descritas en la tabla VII. Entre las operaciones se incluye la suma, dos formas de resta, tres operaciones lógicas estándar y dos operaciones que ponen al bus de salida de datos Z a 0000 o a 1111.

4.5 CONCEPTOS AVANZADOS

ABSOLUTE LOADER (cargador absoluto). Programa destinado a cargar datos en una dirección numérica específica.

ACCESS TIME (tiempo de acceso). Tiempo empleado para buscar una palabra en memoria.

ACUMULADOR. Registro en el que se acumula el resultado de una operación aritmética o lógica.

ACKNOWLEDGE (respuesta afirmativa). Señal de control utilizada para completar una secuencia de confirmación de una secuencia de intercambio de indicativos y señales de control.

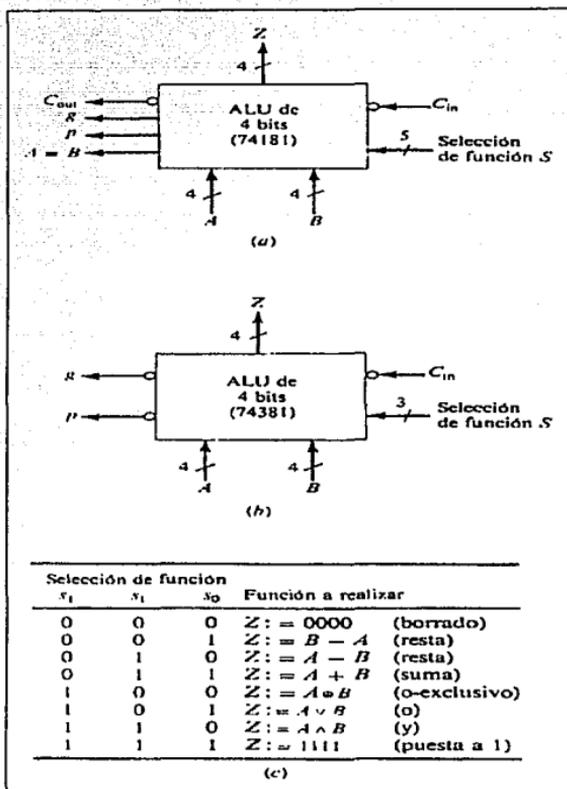


Figura IV-5

A/D (analógico-digital). Conversión de las tensiones y corrientes analógicas procedentes de un sensor para su representación digital en sistemas informáticos.

ADDRESS (dirección). Número que indica la posición de una palabra en memoria.

ALPHANUMERIC (alfanumérico). Conjunto de todos los caracteres alfabéticos y numéricos.

ALU (unidad aritmético-lógica). Parte de la CPU que realiza la ejecución de operaciones tanto lógicas como aritméticas.

ASCII (código americano normalizado para el intercambio de información). Código de caracteres utilizado por la mayoría de los equipos de cómputo.

ASSEMBLER (ensamblador). Programa que toma la forma mnemónica del lenguaje simbólico y lo convierte en código objeto binario para su ejecución.

ASYNCHRONOUS (asíncrono). Evento o dispositivo que no es síncrono con la temporización de la CPU.

BATCH (lote). Colección de documentos o registros agrupados con la finalidad de procesarlos como una sola unidad. Durante este proceso, no es posible realizar comunicaciones interactivas entre el programa y el usuario.

BAUD (baudío). Bits por segundo. Actualmente, unidad binaria de transmisión de información por segundo.

BCD (decimal codificado en binario). Representación de los números decimales en la que cada dígito se representa mediante 4 bits.

BI-DIRECTIONAL (bidireccional). Flujo de datos que puede circular por un cable en cualquier sentido.

BINARY CELL (célula binaria). Celdilla de almacenamiento capaz de retener un dígito binario.

BINARY NUMBER (número binario). Cantidad representada en un código binario, mediante unos y ceros.

BINARY TO DECIMAL CONVERSION (conversión binario a decimal). Proceso de conversión de un número binario a su equivalente decimal.

BIESTABLE (bistable). Dispositivo que siempre se encuentra en uno de dos estados posibles.

BIT Contracción de binary digit (dígito binario). Unidad de información que puede adoptar dos valores o estados distintos.

BIT RATE (velocidad de transmisión). Velocidad expresada en bits por segundo. Se abrevia bps.

BUFFER (memoria intermedia). Cualquier memoria destinada a retener información durante un tiempo determinado, hasta el momento de la transferencia adecuada.

BUS (conductor común). Conexiones por donde las señales van desde unos orígenes a unos destinatarios.

BYTE (octeto). Término que representa una porción medible de dígitos binarios consecutivos.

CALL (llamada). Instrucción utilizada para transferir la secuencia de ejecución de programa a una subrutina o subprograma.

CHARACTER (carácter). Cada uno de los símbolos convencionales adoptados para representar, aisladamente o relacionados entre sí, la información.

CLEAR (borrar, restaurar). Acción de suprimir los datos de una memoria o de unas posiciones de memoria, sustituyéndolos por otros predeterminados.

CLOCK (reloj). Dispositivo que genera una base de tiempos utilizada para proporcionar los impulsos secuenciales básicos para las operaciones de un ordenador, cuya forma de trabajo es secuencial.

CMOS (MOS complementario). Tecnología que se caracteriza por poseer muy bajo consumo. Esta tecnología requiere un canal p y un transistor de canal n, tiene la velocidad e integración intermedias entre NMOS y PMOS. Presenta inmunidad al ruido.

CONTROL UNIT (unidad de control). Modulo encargado de buscar y decodificar instrucciones. Esta unidad requiere de un registro de instrucciones, un contador de programa y genera las señales de control y dirige el bus o barra de control.

CPS (caracteres por segundo).

CPU (unidad central de proceso). Modulo encargado de buscar, decodificar y ejecutar instrucciones. Incorpora una unidad de control, una unidad aritmética y lógica (ALU) y dispositivos afines (registros, reloj, etc).

CPU TIME (tiempo de la CPU). Período de tiempo en el que la CPU se dedica a la ejecución de instrucciones.

DATA COMMUNICATION (comunicación de datos). Proceso consistente en transmitir y recibir datos que comprende operaciones tales como codificación, decodificación, etc.

DAV (datos disponibles). Uno de los cinco bits de estado de un UART estándar.

D-BUS (barra D). Barra de destino interno en una CPU que va de la ALU a los registros.

DIRECT ACCESS (acceso directo). Característica de ciertas memorias en las que los datos almacenados en ellas pueden serlo en una secuencia de direccionamiento que no guarda criterio alguno y cuyo acceso se puede realizar en forma directa, reduciéndose así el tiempo invertido.

DISABLE (deshabilitación). Supresión o anulación de un dispositivo de interrupción.

DYNAMIC MEMORY (memoria dinámica). Memoria RAM MOS que utiliza circuitos dinámicos. Cada bit se almacena en forma de carga en un transistor MOS.

EDGE (flanco). Intervalo de tiempo durante el cual una señal lógica pasa del estado 0 al estado 1, o viceversa.

EIA RS232C. Acoplamiento serie estándar para comunicaciones asíncronas.

ENABLE (habilitación). Restauración al estado operativo de un dispositivo de desinhibición.

ENVIRONMENT (configuración). Estado de todos los registros, ubicaciones de memoria y de otras condiciones operativas de un sistema.

EPROM (memoria pasiva ROM programable). Se refiere a las memorias PROM que pueden borrarse por exposición a los rayos ultravioleta.

ERASE. Concepto empleado para expresar la idea de suprimir unos datos almacenados sobre un soporte magnético, devolviendo cada uno de los elementos magnetizados a su estado nulo original, o incluso, a otro diferente al existente en ese momento.

FAN-IN (cargabilidad de entrada). Carga eléctrica presentada por una entrada a una salida.

FAN-OUT (cargabilidad de salida). Carga eléctrica que puede excitar una salida. Normalmente se expresa como el número de entradas que pueden ser excitadas.

FIFO (primero en entrar - primero en salir). Estructura en la cual los datos se depositan en un extremo y se extraen del otro. Las FIFO se utilizan como medio de conexión de dos dispositivos que trabajan en forma asíncrona y a distintas velocidades.

FIRMWARE (soporte lógico inalterable). Programa depositado en ROM.

FLAG (señalizador). Caracter que señala en acacimiento de alguna condición, tal como el final de una palabra.

GATE (puerta). Circuito provisto de una única señal de salida, cuyo valor depende del estado de las señales de entrada.

GPB (barra de acoplamiento de aplicación general). Nombre utilizado por la normativa de acoplamiento de barras IEEE-488-1975.

HALF-DUPLEX (semiduplex). Técnica empleada en comunicaciones en la que los datos pueden transmitirse en una sola dirección a la vez.

HANDSHAKING (intercambio de indicativos y señales de control). Técnica de sincronización de comunicación entre dos modems o terminales de datos.

HARDWARE. Conjunto de los dispositivos mecánicos, eléctricos y electrónicos que comprenden la estructura de un sistema de cómputo.

HEXADECIMAL (hexadecimal). Representación de los números en base 16.

IEEE (Instituto de Ingenieros Eléctricos y Electrónicos).

INSTRUCTION (instrucción). Secuencia única en un programa. Esta secuencia puede ser extraída de memoria, decodificada y ejecutada por la CPU. Pueden ser aritméticas, lógicas, de operación con registros, memorias y dispositivos de E/S u operaciones de control específicas.

INTERRUPT (interrupción). Señal de atención enviada por un dispositivo de E/S o chip a una MPU para romper el funcionamiento normal de la rutina que se esta ejecutando. Señal que origina la pausa.

I/O (entrada/salida, E/S). Líneas o dispositivos empleados para obtener o visualizar la información en el exterior.

LIFO (último en entrar, primero en salir, pila). Término que describe un método de extracción de elementos o artículos de una lista ordenada en cola.

LINK (enlace). Indicador de dirección al próximo elemento de una lista o de un bloque contíguo de un fichero.

LOADER (cargador). Programa especial que toma la información de un formato binario y la pone en memoria.

MACHINE CODE (código de máquina). Sistema de codificación adoptado en el diseño de un ordenador para representar su juego de instrucciones.

MACRO INSTRUCTION (macroinstrucción). Instrucción individual que se escribe como parte de un lenguaje fuente y que cuando se compila en un programa en código de máquina, genera varias instrucciones en código de máquina.

MAIN FRAME (unidad central). En un principio comprendía la estructura principal de una CPU, en la que estaban instalados la unidad aritmética y la circuitería lógica asociada. Actualmente se refiere al propio procesador central.

MAIN MEMORY (memoria principal). Memoria interna de un ordenador, es decir, almacenamiento de acceso inmediato a diferencia de cualquier almacenamiento auxiliar que pueda formar parte del sistema del ordenador.

MAIN PROGRAM (programa principal). Parte central de un programa que generalmente transfiere el control a otras subrutinas, según la naturaleza de los datos que se estén procesando.

MEMORY (memoria). Dispositivo en el que se pueden introducir datos, donde se mantienen y donde pueden retirarse posteriormente. Cualquier dispositivo capaz de almacenar datos.

MICROINSTRUCTION (microinstrucción). Instrucción que forma parte de una sección de microcodificación.

MNEMONIC (mnemónico). A los nombres de las instrucciones se les conoce como mnemónicos. Cada instrucción está codificada con un patron de bits único que es reconocido y

ejecutado por el microprocesador.

NIBBLE. Conjunto de 4 bits.

OBJECT CODE (código objeto). Salida de un compilador o de un ensamblador, que por propia naturaleza, es un código ejecutable de máquina o resulta apropiada para que sea procesada con el fin de producir un código ejecutable de máquina.

OBJECT PROGRAM (programa objeto). Programa obtenido en lenguaje de máquina después de la compilación o ensamblaje de otro simbólico. Es un programa simbólico traducido a otro de máquina.

OPERATION (operación). Es una acción específica que realiza el microprocesador siempre que ejecuta una instrucción. La mayoría de las instrucciones que ejecuta la CPU son ordenadas por la memoria del programa. Los datos que se procesarán se almacenan en la memoria de datos. El grupo de instrucciones ejecutables por un microprocesador se llama conjunto de instrucciones y puede tener de ocho hasta 200 operaciones básicas.

OUTPUT UNIT (unidad de salida). Dispositivo o dispositivos que se integran en la configuración de un equipo de tratamiento automático de la información, cuya única finalidad es proporcionar la salida al proceso en curso.

PARALLEL (paralelo). Método de tratamiento simultáneo de todos los elementos de unidad de información.

PARALLEL TRANSFER (transferencia en paralelo). Método de transferencia de datos en la que la transmisión de éstos se realiza de forma simultánea.

PARITY BIT (bit de paridad). Bit de verificación o su complemento que se añade a los bits de un bloque de información para controlar la transferencia de dicho bloque entre las unidades del equipo, pero que no tiene valor como tal información.

PERIPHERAL CONTROL UNIT (unidad de control de periférico). Unidad encargada de controlar las operaciones de una o más unidades periféricas pertenecientes a un sistema de tratamiento automático de datos y que permanecen bajo el control de la CPU de dicho sistema.

PERIPHERAL UNIT (periférico). Máquina que puede funcionar bajo el control del ordenador. El equipo periférico consta de dispositivos de E/S y de almacenamiento.

PORT (vía de acceso). Parte de un procesador de datos que se dedica a un sólo canal de datos con el fin de recibir o transmitir éstos a uno o más dispositivos externos situados a distancia.

PROCESSOR (procesador). Cualquier dispositivo capaz de desarrollar operaciones con datos.

PROGRAM. Es un grupo de instrucciones organizadas en la forma secuencial que comandan a la CPU a realizar funciones más complejas que las que se logran con simples instrucciones. El programa y los datos son los dos tipos de información que deben introducirse a la computadora y almacenarse en memoria.

PROGRAMMING (programación). Proceso que comprende el diseño, escritura y prueba de un programa.

PROTOCOL (protocolo). Conjunto de normas que gobiernan la forma de comunicación entre dos equipos.

RAM (memoria de acceso aleatorio). Memoria que se presta para almacenar datos durante el tiempo en que esté energizada.

READ (leer). Acción de obtener datos almacenados en memoria.

REGISTER (registro). Dispositivo cuya función consiste en retener una información que se ha de tratar a continuación.

RESET (restaurar). Acción de poner en cero a un contador o devolver un indicador a alguna posición estable.

ROM (memoria de solo lectura). Dispositivo capaz de retener datos, los cuales no se pueden alterar por instrucciones de programa.

SERIAL TRANSFER (transferencia en serie). Transferencia sucesiva de una serie de elementos de información, de unidad a unidad o de terminal a terminal.

SOFTWARE. Conjunto de todos los elementos que intervienen en la programación y utilización de un ordenador o calculadora: programas, rutinas, formularios y manuales.

STACK (pila). Estructura LIFO que memoriza la información en orden cronológico.

STATE TABLE (tabla de estados). Lista de salidas de un circuito lógico basado en las

entradas y en previas salidas. Dicho circuito tiene memoria y no puede ser descrito por una simple tabla de verdad.

STATUS (estado). Condición actual de un dispositivo.

STATUS REGISTER (registro de estado). Registro utilizado para mantener la información de estado dentro de una unidad funcional, como una MPU. Un registro de estado de MPU típico proporciona indicación de acarreo, rebasamiento, signo, cero e interrupción. Puede incluir también paridad, desinhibición o máscara.

STEP (ejecutar un paso). Acción de hacer que un ordenador ejecute una operación.

TIME ACCESS (tiempo de acceso). Tiempo invertido en la búsqueda de una información situada en memoria y su transferencia a la ALU del ordenador.

TIMING (temporización). Proceso de cálculo y regularización de unas operaciones con respecto a un ritmo de tiempo.

TRANSFER (transferir). Acción de transmitir uno o varios datos de un punto a otro, escribiéndolos con idéntico contenido en el receptor. Mover información desde unas posiciones de memoria a otras.

TWO'S COMPLEMENT (complemento a dos). Método de expresar números binarios en donde el negativo de un número se genera complementando al número y añadiendo 1.

UART (transmisor receptor asíncrono universal). Convertidor serie-paralelo y paralelo-serie.

UNCONDITIONAL JUMP (salto incondicional). Salto que se produce en la ejecución de la secuencia normal de unas instrucciones al aparecer una instrucción de transferencia incondicional.

USART (transmisor receptor síncrono asíncrono universal). Convertidor serie-paralelo para comunicaciones de alta velocidad.

VOLATILE STORAGE (memoria volátil). Sistema de memoria en que los datos almacenados se pierden cuando se desconecta la corriente que alimenta al sistema.

WORD (palabra). Unidad lógica de información que puede tener cualquier número de bits, pero normalmente es de 4, 8 o 16. Un microprocesador de ocho bits transfiere y almacena todos

los datos en grupos de ocho bits, por medio de ocho conductores paralelos que en conjunto forman el bus de datos.

WRITE (escribir, grabar). Acción de transcribir datos en una forma de almacenamiento desde otra de almacenamiento distinta.

ZERO (cero). Condición de códigos que un ordenador reconoce como cero.

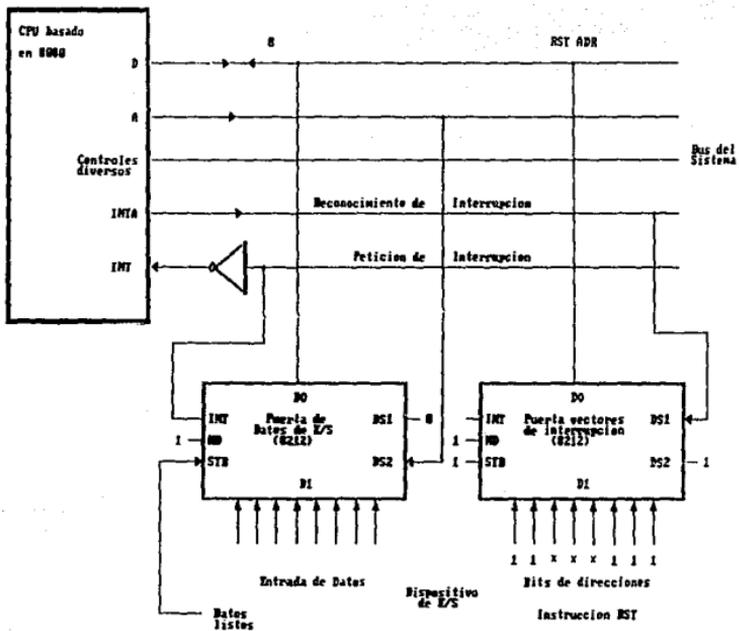
4.6 INTERRUPCIONES

Un mecanismo importante de comunicación entre los dispositivos de E/S y la CPU es la interrupción. Una interrupción permite a un dispositivo de E/S, pedir al CPU la suspensión temporal de la ejecución del programa actual y ramificarse a otro programa. Una razón típica para una petición de interrupción es la necesidad para un dispositivo de E/S de enviar datos o recibir datos de la CPU o de la memoria principal M. Frecuentemente se utiliza un par de líneas de control asíncronas, petición de interrupción y conformidad de interrupción, para temporizar la interrupción. Un dispositivo de E/S activa la línea de petición de interrupción para informar a la CPU de que hay una interrupción pendiente. Cuando responde a la petición de interrupción, lo que usualmente se realiza entre ciclos de instrucción, la CPU guarda el contenido del contador de programa como en una llamada a subrutina, y en el PC carga la dirección de interrupción, que es la dirección inicial de la rutina de interrupción. Normalmente se necesita una variedad de diferentes rutinas de interrupción para realizar todas las operaciones requeridas por los dispositivos de E/S. En microprocesadores tales como el 6800 de Motorola, la interrupción hace que el control del programa se transfiera a una dirección de interrupción fija, requiriendo por tanto que la CPU ejecute instrucciones adicionales para determinar que rutina utilizar de todas las posibles. Otros microprocesadores como el 8080/8085 de Intel, permiten al dispositivo de E/S interruptor transmitir las direcciones de interrupción (conocidas también como vectores de interrupción) directamente a la CPU, permitiendo por tanto una transferencia de control inmediata a la rutina de interrupción deseada. Esto se conoce como control de interrupción con vectores.

Los microprocesadores de la serie 8080/8085 presentan dispositivos de interrupción que generan

una instrucción completa de llamada tal como CALL ADR, en la que ADR es una dirección completa de memoria 16 bits; esto permite utilizar cualquier dirección de M como dirección de interrupción. La serie 8080/8085 tienen una instrucción especial de llamada de un byte RST (Comenzar o Reset), que puede usarse por dispositivos de entrada y salida para especificar una entre ocho direcciones de interrupción. El 8085 permite que el dispositivo de interrupción suministre una instrucción CALL completa de 3 bytes, esto permite utilizar cualquiera de las 64K posibles direcciones de memoria como dirección de interrupción. La figura IV-6 presenta un circuito de interfaz representativo para un sistema basado en un 8080 y con dispositivos de E/S administrados con interrupciones. La CPU y los dispositivos de E/S son conectados directamente a través de las líneas INT, de petición de interrupción, e INTA, de reconocimiento de interrupción. El circuito de interfaz esta compuesto por dos CI de E/S de uso general 8212 (sus características eléctricas se muestran en la figura IV-7), uno sirve de puerta de datos y el otro suministra las instrucciones RST que incorporan el vector de interrupción deseado. La entrada de control STB (habilitar) del 8212 se activa por una señal de control dato-listo que acompaña a la carga de datos en la puerta de datos del 8212, a continuación aquella activa la línea de salida INT. Como su nombre sugiere, INT está proyectada para conectarse con un inversor a la línea de petición de interrupción, INT del sistema. Así la carga de un nuevo dato en la puerta de datos envía automáticamente una señal de petición de interrupción a la CPU. Cuando la CPU está lista para responder a esta petición de interrupción, activa la línea de reconocimiento de interrupción INTA, que está unida a las entradas de selección (de pastilla) del dispositivo del segundo 8212. Las entradas de control de este CI están dispuestas para hacerlo actuar como una puerta de entrada (de datos) cuando se selecciona. Sus líneas de entrada de datos están cableadas a señales constantes que definen la instrucción RST ADR a ser usada. Cuando se activa INTA se selecciona el segundo 8212, causando que la instrucción RST ADR se sitúe en el bus de datos. La CPU entonces lee la instrucción del bus de datos y la procesa como cualquiera otra instrucción de un byte.

En la figura IV-8 se observa el modo de conexión del puerto E/S 8212 con la unidad microprocesadora.



Circuito de interfaz representativo para un sistema basado en el 8080 y con dispositivos E/S administrados con interrupciones

Figura IV-6

8212 8-BIT INPUT/OUTPUT PORT

- Fully Parallel 8-Bit Data Register and Buffer
- Service Request Flip-Flop for Interrupt Generation
- Low Input Load Current — .25mA Max.
- Three State Outputs
- Outputs Sink 15mA
- 3.65V Output High Voltage for Direct Interface to 8008, 8080A, or 8085A CPU
- Asynchronous Register Clear
- Replaces Buffers, Latches and Multiplexers in Microcomputer Systems
- Reduces System Package Count

The 8212 input/output port consists of an 8-bit latch with 3-state output buffers along with control and device selection logic. Also included is a service request flip-flop for the generation and control of interrupts to the microprocessor. The device is multimode in nature. It can be used to implement latches, gated buffers or multiplexers. Thus, all of the principal peripheral and input/output functions of a microcomputer system can be implemented with this device.

PIN CONFIGURATION



PIN NAMES

Pin No.	DATA IN
D0-D7	DATA IN
D0-D7	DATA OUT
D0-D7	DEVICE SELECT
INT	INTERRUPT
INT	ASYNCHRONOUS ACTIVE LOW
CLR	REGISTER ACTIVE LOW

LOGIC DIAGRAM

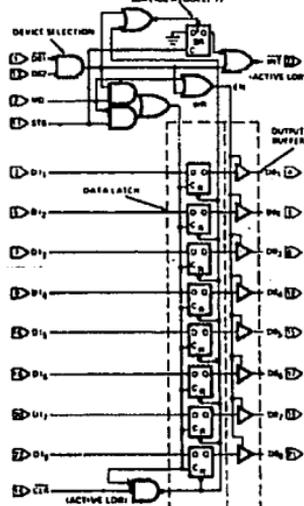
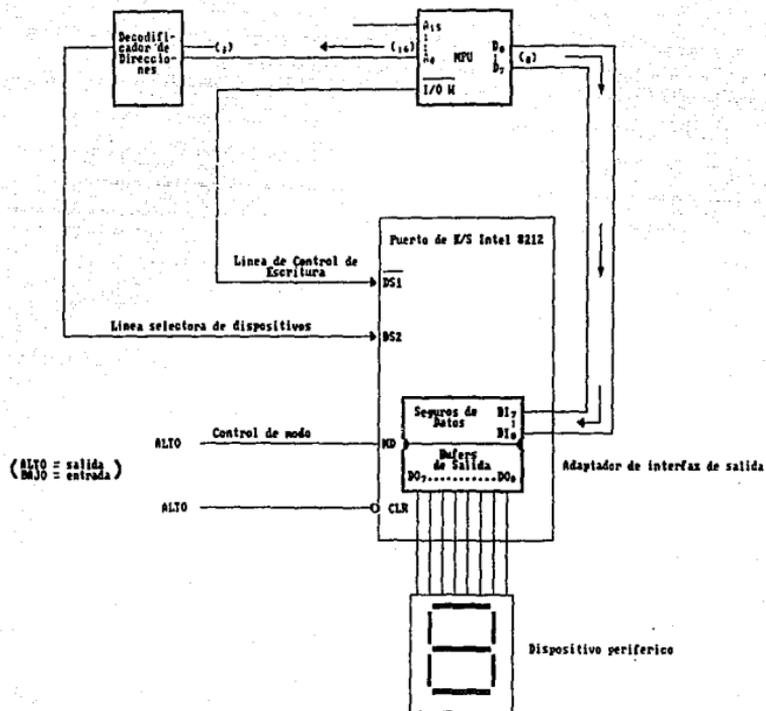


Figura IV-7



El CI de E/S 8212 como puerto de salida

Figura IV-8

El 8080/8085 tiene la instrucción DI (deshabilitación de interrupciones), que inhabilita la línea principal INT/INTR de petición de interrupción poniendo a uno un biestable de la máscara de interrupción INTE (interrupción habilitada). La instrucción interrupción habilitada pone a cero el biestable INTE, por tanto habilita el sistema de interrupciones.

La estructura general de una rutina de servicio de interrupción se describe en la figura IV-9. El punto de entrada de la rutina está determinado por el conjunto disponible de direcciones de interrupción. Si se desea hacer la rutina ininterrumpible, su primera instrucción debe inhabilitar todo o parte del sistema de interrupciones; en algún caso la deshabilitación de otras interrupciones se efectúa automáticamente por la CPU cuando responde a una instrucción. Después cualquier registro de la CPU que sea modificado por la rutina debe ser salvado; esto puede hacerse adecuadamente introduciendo los registros en cuestión en la zona de fila de la memoria principal. A continuación sigue el cuerpo principal de la rutina de interrupción. Normalmente se refiere a la transferencia de datos desde los dispositivos de E/S y cualquier operación de contabilidad asociada. La rutina termina restaurando cualquier registro de la CPU previamente salvado, habilitando cualquier línea de petición de interrupción previamente inhabilitada, y ejecutando finalmente una instrucción de retorno para establecer la ejecución del programa que originalmente fue interrumpido.

Punto de entrada
Inhabilitar otras interrupciones, si se requiere
Salvar todos los registros de la CPU que se modifiquen
Cuerpo principal de la rutina de servicio de interrupciones
Restaurar todos los registros de la CPU previamente salvados
Habilitar toda interrupcion previamente inhabilitada
Punto de salida (retorno)

Figura IV-9
76

CAPITULO V

PROGRAMACION DE MICROPROCESADORES

INTRODUCCION

El microprocesador es capaz de efectuar operaciones distintas, puede sumar y restar números, también efectuar operaciones lógicas, leer dispositivos de entrada y transmitir información a dispositivos de salida.

Las operaciones son acciones específicas que realiza el microprocesador y una instrucción es un "patrón de bits" que ordena al mismo ejecutar una operación específica. Cada microprocesador tiene determinado su conjunto de instrucciones. Las instrucciones se obtienen de la memoria durante un ciclo FETCH (ciclo de búsqueda) y se cargan al "registro de instrucciones" para su interpretación y poder generar acciones definidas.

Existen también las banderas que son indicadores de la salida de la ALU, y que son fundamentales para la secuencia y decisiones tomadas en un programa. Estas banderas indican también la entrada en acción de subrutinas, que son módulos independientes que pueden ejecutarse en el programa dirigiéndose a éstas por medio de un salto, precedido de una interrupción.

Las interrupciones también se refieren a dar paso a la ejecución de una función llevada a cabo por un dispositivo externo.

Al trabajar, el microprocesador cuantifica la instrucción ejecutada por medio de la unidad de instrucciones. Así como también el tiempo empleado en cada instrucción.

Finalmente, es importante conocer los conceptos que deben emplearse en el manejo de los microprocesadores.

5.1 LENGUAJE DE MAQUINA Y ENSAMBLADORES

Así como ha mejorado la tecnología física de las computadoras, también han evolucionado en software. Las primeras computadoras se programaban únicamente en lenguaje máquina, que es el código binario que ejecuta directamente la CPU. Una reducción significativa en la dificultad de programar se consiguió con la introducción de los lenguajes ensambladores, que son lenguajes de programación que permiten escribir instrucciones en forma simbólica utilizando nombres fácilmente recordables que dan a las operaciones y operandos de cada instrucción. Se necesita un programa especial llamado ensamblador para traducir el programa (fuente) escrito en lenguaje ensamblador a un programa (objeto) en lenguaje máquina antes de ser ejecutado.

Cuando la utilización de una computadora se incrementa, la dificultad de escribir programas aumenta también. Mientras que los lenguajes ensambladores son mucho más fáciles de usar que los lenguajes máquina, tienen la característica de que son inadecuados para muchas tareas de diseño de software. Los lenguajes ensambladores, similarmente a los lenguajes máquina, son específicos de la computadora, de forma que los programas escritos en lenguaje ensamblador para un tipo de computadora no pueden usarse en otro tipo de computadora. Los lenguajes ensambladores usualmente tienen cierto parecido al lenguaje (por ejemplo, inglés o el español) en el que el usuario de la computadora puede más fácilmente describir un problema. Un lenguaje ensamblador utiliza palabras y frases para representar los códigos de máquina del microprocesador.

Generalmente, una frase o instrucción en lenguaje ensamblador consistirá de 1 a 3 bytes de código de máquina.

En las siguientes hojas se enlistan los conjuntos de instrucciones de los microprocesadores 8080/8085 de Intel, junto con sus características principales, divididos en 7 categorías que son:

- 1.- Instrucciones aritméticas.
- 2.- Instrucciones lógicas.
- 3.- Instrucciones de transferencia de datos.
- 4.- Instrucciones de bifurcación.

5.- Instrucciones de llamado de subrutinas.

6.- Instrucciones varias.

INSTRUCCIONES ARITMETICAS.

Incluyen las instrucciones de suma, resta, incremento, decremento y comparación. El Acumulador se convierte en el registro básico para estas operaciones.

INSTRUCCIONES LOGICAS.

Incluyen AND, OR, OR Exclusivo, complemento (NOT) e instrucciones de rotación. El enfoque en la mayoría de las operaciones se encuentra en los contenidos del Acumulador. El modo de direccionamiento tiene que ver con cómo y dónde se encuentran otros datos en el sistema microprocesado.

INSTRUCCIONES DE TRANSFERENCIA DE DATOS.

Estas instrucciones incluyen entre otras el movimiento de registro, a registro, la carga de la memoria, el almacenamiento en memoria, la entrada y salida y las instrucciones de colocar la bandera de acarreo. Existen muchas instrucciones en este grupo, por lo que los datos pueden ser transferidos de una posición de memoria a cualquier registro o de cualquier registro a cualquier posición de memoria u otros registros. Una instrucción de transferencia de datos tiene una fuente de datos y un destino para ellos. El modo de direccionamiento tiene que ver con cómo y dónde se encuentra la fuente de estos datos.

INSTRUCCIONES DE BIFURCACION

La microcomputadora normalmente ejecuta instrucciones en orden secuencial. El contador de

programa de 16 bits del microprocesador siempre contiene la dirección de la siguiente instrucción que será rastreada de la memoria y ejecutada. El contador de programa normalmente está secuenciado hacia arriba, un número a la vez. Las instrucciones de bifurcación o brinco proporcionan un método de cambiar el valor del contador del programa; por lo tanto, alteran la secuencia normal de la ejecución del programa.

INSTRUCCIONES DE LLAMADO DE SUBRUTINAS

Son 2: **CALL**, que es de 3 bytes y se utiliza en el programa principal para hacer que el microprocesador brinque a una subrutina y la instrucción **RET**, de 1 byte, que se emplea para retornar al estado siguiente de la llamada de subrutina.

OPERACIONES VARIAS

Incluyen las instrucciones de introducción, extracción, no operación y alto.

5.2 PROGRAMA MONITOR

Este programa reside en la memoria del sistema y permite al usuario introducir órdenes a través de una terminal estándar, o a través de un teclado especial para ejecutar programas de aplicación o de desarrollo del sistema. Los programas monitores varían en complejidad desde programas sencillos en un computador de una tarjeta con un pequeño teclado y una presentación visual consistente en unos pocos dígitos, a programas complejos para sistemas de computador grande con diferentes terminales estándar y otros dispositivos periféricos.

En el momento de encender el microprocesador, éste inicia la ejecución de la instrucción indicada por el contador de programa (PC), (éste a su vez se carga aleatoriamente con una dirección que el usuario desconoce). Para evitar que esta situación quede fuera de control los fabricantes de computadoras proporcionan un método especial para cargar al PC con una

dirección donde se encuentra un programa de inicialización. Este programa se denomina monitor, el cual se comunica con el usuario a través de un dispositivo de E/S, generalmente son un display y un teclado. El usuario da órdenes al monitor a través de comandos en un lenguaje especial, mientras que el monitor responde con mensajes o ejecución de acciones.

De entrada, el monitor le indica al usuario que se encuentra listo para ejecutar el comando que se le indique a través de alguna marca especial (PROMPT), en este momento, el usuario tiene el control del sistema y por medio de los comandos el monitor controla todas las acciones de la computadora.

Las tareas que generalmente realiza un programa monitor son las siguientes:

- 1.- Cargar un programa objeto en la memoria principal.
- 2.- Ejecutar programas.
- 3.- Accesar y modificar los registros internos del microprocesador.
- 4.- Alojara datos en la unidad de memoria principal.

5.3 REGISTRO, MEMORIA Y BUSES

REGISTRO

En general un registro de n bits es una asociación de n flip-flop's. Los más frecuente es que los registros se construyan con flip-flop's tipo D disparados por flanco, pero también se utilizan otros tipos de flip-flop's, tales como los JK disparados por nivel o por flanco. Los flip-flop's D tienen la ventaja de requerir una sola línea de datos por cada bit almacenado, mientras que el JK necesita dos. A menos que se especifique otra cosa, las líneas de entrada y salida de datos de un registro R de n bits se suponen independientes, con lo que los n bits de una palabra se pueden transferir hacia o desde R simultáneamente o en paralelo, es decir, en un ciclo de reloj. Este modo de transferencia de datos se conoce como E/S paralelo frente al modo de E/S serie de transferencia de datos que se utiliza en los registros de desplazamiento. Por esta razón R se conoce en ocasiones como registro paralelo. Aunque los flip-flop's de R tienen líneas de datos

paralelas separadas, normalmente comparten las líneas de control de reloj, de clear y de preset (puesta a uno).

Un ejemplo de la agrupación de registros en un microprocesador se observa en el 8085, el cual tiene 8 registros direccionables de ocho bits. Seis de estos pueden emplearse como registros de ocho bits o pares de registros de 16 bits, y estos registros son:

EL ACUMULADOR. Conocido como registro A es el foco de todas las operaciones de acumulación que incluyen las instrucciones aritméticas, lógicas, de carga y almacenamiento y las de E/S. Es un registro de ocho bits.

REGISTROS DE PROPOSITO GENERAL. BC, DE y HL pueden usarse como seis registros de ocho bits o tres de 16 bits, según sea la instrucción que se está ejecutando. Un ejemplo, el par de registros HL (conocido como el apuntador de datos por Intel), sirven para apuntar direcciones.

EL CONTADOR DE PROGRAMA (PC). Apunta siempre a la posición de memoria de la siguiente instrucción que será ejecutada. Siempre contiene una dirección de 16 bits.

EL APUNTADOR DE FILA (SP). Es un apuntador de dirección de propósito especial (o apuntador de datos), que siempre apunta a la parte más alta de la fila en RAM. Es un registro de contador especializado de 16 bits que siempre almacena una dirección. La dirección en él es la localización de un grupo especial de posiciones de almacenamiento en la memoria de datos, a las cuales se les conoce como fila.

EL REGISTRO DE BANDERA. Contiene cinco banderas de un bit que incluyen la información del estado de la CPU. Estas banderas son utilizadas por instrucciones de brinco condicional, llamado y regreso de una subrutina.

MEMORIA.

Se comentó en el capítulo anterior que las memorias se disponen en forma matricial, de tal forma que se tienen N número de localidades o direcciones para palabras de m bits de longitud. La memoria se divide en memoria de programa y memoria de datos. En la memoria de programa,

obviamente se escriben las instrucciones que habilitan a los registros integrados en el microprocesador, y éstos a su vez direccionan las localidades de la memoria de datos, en donde se almacenan los datos a ser operados así como los resultados de cada operación. Después de ejecutada la instrucción, se regresa a la secuencia de la memoria de programa para poder realizar la siguiente operación.

Como se observa, al hacer un programa en un microprocesador, paralelamente se practica un juego con todos los registros incluidos en el sistema y guardando los datos de importancia en una infinidad de localidades de memoria y que en su momento serán nuevamente utilizados.

BUSES

BUS DE DATOS. Es un conjunto de conductores a través de los cuales se transfiere la información desde y hacia el microprocesador.

La longitud en bits de éste bus, es por lo general, igual a la longitud de palabra de la memoria principal.

Las instrucciones (códigos de operación) se transfieren por este bus debido a que en el momento de la transferencia el contenido de una localidad de memoria es considerada como un dato.

Al conjunto de bits que se transmiten desde la memoria principal hacia el microprocesador se les denominará datos, aunque realmente estos bits conformen una instrucción. La diferencia entre sí son datos o instrucciones, lo que se está transportando, lo determina la unidad de control a través de diferentes señales.

BUS DE DIRECCIONES. Es el canal por el cual el microprocesador selecciona la localidad de memoria o el puerto de E/S con el cual va a realizar un intercambio de información.

Los bits alojados en el área de datos se llamarán siempre datos, aunque están residentes en la memoria o se están transfiriendo.

El bus de direcciones en los microprocesadores de 8 bits es, generalmente, de 16 bits. Con esta cantidad de bits se pueden direccionar hasta 65,536 palabras de memoria (2^{16}).

BUS DE CONTROL. Por medio de este bus, el microprocesador transmite y recibe las señales adecuadas para la sincronización con el mundo externo. Es decir, si va a colocar un dato sobre la memoria debe indicarle al chip de memoria correspondiente que le va a enviar un dato y a su vez el circuito debe contestarle que está listo y en este momento se realiza la transferencia.

5.4 EJEMPLOS DE PROGRAMACION

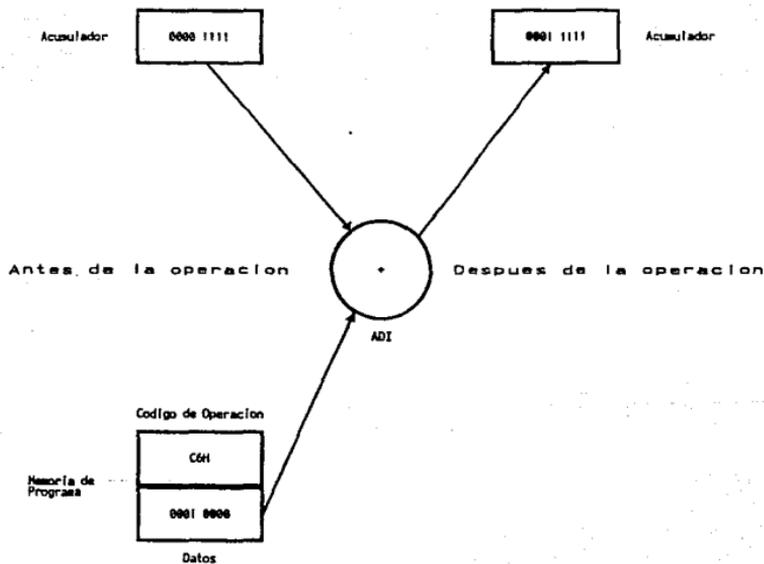
OPERACIONES ARITMETICAS: Existen cuatro instrucciones de suma. El acumulador (A), almacena uno de los números que han de ser añadidos. Cada instrucción de suma especifica una fuente diferente para el otro número que ha de ser sumado.

La primera instrucción mostrada en la tabla I, "suma inmediata de A" (ADI), es una instrucción de dos bytes. Su código de operación es el C6 hexadecimal (C6H), ocupa el primer byte de la memoria de programa, con los datos que han de ser sumados al acumulador en el segundo byte. Suponiendo que se quieren sumar dos números, uno de los cuales está almacenado en el acumulador (00001111₂), el otro se coloca en la memoria de programa en forma directa (00010000₂), utilizando el código de operación de la operación en proceso. Observar la figura V-1. Los datos inmediatos contenidos en la memoria de programa se suman a los almacenados en el acumulador después de la suma inmediata (ADI), el resultado se almacena en el acumulador.

La instrucción "Suma el registro L a A" (ADD L), necesita sólo un byte y es una instrucción de registro a registro. Un ejemplo se observa en la figura V-2, en la que los contenidos del acumulador (00001000₂) se suman a los del registro L (00000001₂). El resultado se almacena en el acumulador desplazando el dato anterior.

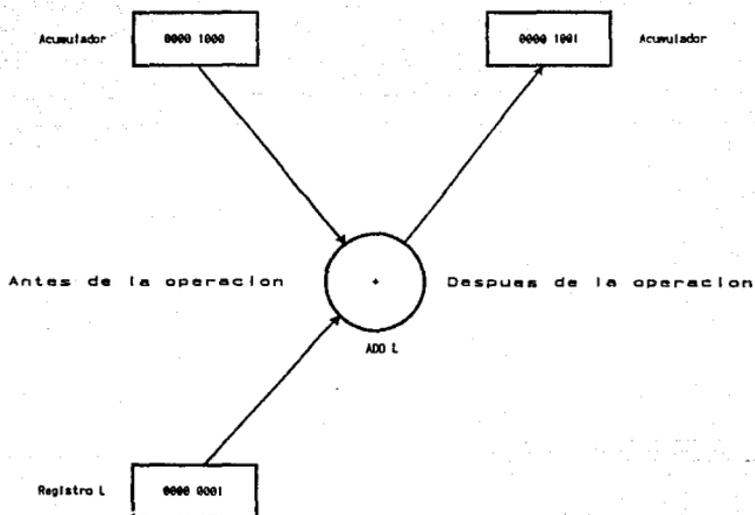
La instrucción "Suma el registro H a A" (ADD H), es similar a la anterior, el único cambio es el registro L por H.

La última instrucción de suma es "Suma indirecta de registro" (ADD M), necesita un byte de la memoria de programa. La localización de los datos que serán sumados es algo más complicada utilizando éste modo de direccionamiento indirecto de registro.



Suma A a los datos (ADI)

Figura V-1



Suma L a A (ADD L)

Figura V-2

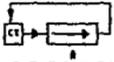
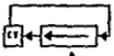
INSTRUCCIONES ARITMETICAS

DESCRIPCION DE LA OPERACION	MODO DE DIRECCIONAMIENTO	MEMORICO	COGEO DE OPERACION	BYTES	FORMULO DE LA INSTRUCCION	SIMBOLICO	CONDICIONES QUE SE AFECTAN
Suma A a los datos	INMEDIATO	ADI	06	2	Código de op. datos	$(A) \leftarrow (A) + (\text{byte } 2)$	Z, CY
Suma L a A	REGISTRO	ADD L	05	1	Código de op.	$(A) \leftarrow (A) + (L)$	Z, CY
Suma H a A	REGISTRO	ADD H	04	1	Código de op.	$(A) \leftarrow (A) + (H)$	Z, CY
Suma LOC(H y L) a A	INDIRECTO DE REGISTRO	ADD H	06	1	Código de op.	$(A) \leftarrow (A) + ((H)(L))$	Z, CY
Resta los datos de A	INMEDIATO	SUI	06	2	Código de op. datos	$(A) \leftarrow (A) - (\text{byte } 2)$	Z, CY
Resta L de A	REGISTRO	SUB L	95	1	Código de op.	$(A) \leftarrow (A) - (L)$	Z, CY
Resta H de A	REGISTRO	SUB H	94	1	Código de op.	$(A) \leftarrow (A) - (H)$	Z, CY
Resta LOC(H y L) de A	INDIRECTO DE REGISTRO	SUB H	96	1	Código de op.	$(A) \leftarrow (A) - ((H)(L))$	Z, CY
Incrementa A	REGISTRO	INC A	3C	1	Código de op.	$(A) \leftarrow (A) + 1$	Z
Incrementa HL	REGISTRO	INX H	23	1	Código de op.	$(HL) \leftarrow (HL) + 1$	
Decrementa A	REGISTRO	DCR A	3D	1	Código de op.	$(A) \leftarrow (A) - 1$	Z
Decrementa HL	REGISTRO	DCX H	2B	1	Código de op.	$(HL) \leftarrow (HL) - 1$	
Compara A con los datos	INMEDIATO	CPI	FE	2	Código de op. datos	$(A) - (\text{byte } 2)$	Z=1 if (A) < (byte 2) CY=1 if (A) < (byte 2)
Compara A con L	REGISTRO	CMP L	BD	1	Código de op.	$(A) - (L)$	Z=1 if (A) < (L) CY=1 if (A) < (L)
Compara A con H	REGISTRO	CMP H	BC	1	Código de op.	$(A) - (H)$	Z=1 if (A) < (H) CY=1 if (A) < (H)
Compara A con LOC(H y L)	INDIRECTO DE REGISTRO	CMP H	BE	1	Código de op.	$(A) - ((H)(L))$	Z=1 if (A) < ((H)(L)) CY=1 if (A) < ((H)(L))

(1) CONTENIDO DE DIRECCIONAMIENTO INDIRECTO DE REGISTROS

TABLA 1

INSTRUCCIONES LÓGICAS

DESCRIPCIÓN DE LA OPERACIÓN	MODO DE DIRECCIONAMIENTO	NUMERICO	CÓDIGO DE OPERACIÓN	BYTES	FORMA DE LA INSTRUCCIÓN	SIMBOLICO	EFECTOS QUE SE AFECTAN
Ligan A con los datos por medio de un AND	INMEDIATO	ANI	E6	2	[Codigo de op.] datos	$(A) \leftarrow (A) \cdot (\text{byte 2})$	Z CY es cero
Ligan A con L por medio de un AND	REGISTRO	ANA L	A5	1	[Codigo de op.]	$(A) \leftarrow (A) \cdot (L)$	Z CY es cero
Ligan A con H por medio de un AND	REGISTRO	ANA H	A4	1	[Codigo de op.]	$(A) \leftarrow (A) \cdot (H)$	Z CY es cero
Ligan A con L y H por medio de un AND	INDIRECTO DE REGISTRO	ANA B	A6	1	[Codigo de op.]	$(A) \leftarrow (A) \cdot ((B)(L))$	Z CY es cero
Ligan A con los datos por medio de un OR	INMEDIATO	ORI	F6	2	[Codigo de op.] datos	$(A) \leftarrow (A) + (\text{byte 2})$	Z CY es cero
Ligan A con L por medio de un OR	REGISTRO	OAL	B5	1	[Codigo de op.]	$(A) \leftarrow (A) + (L)$	Z CY es cero
Ligan A con H por medio de un OR	REGISTRO	OAH	B4	1	[Codigo de op.]	$(A) \leftarrow (A) + (H)$	Z CY es cero
Ligan A con L y H por medio de un OR	INDIRECTO DE REGISTRO	OAB	B6	1	[Codigo de op.]	$(A) \leftarrow (A) + ((B)(L))$	Z CY es cero
Ligan A con los datos por medio de un XOR	INMEDIATO	XRI	EE	2	[Codigo de op.] datos	$(A) \leftarrow (A) \oplus (\text{byte 2})$	Z CY es cero
Ligan A con A por medio de un XOR exclusivo	REGISTRO	XAA A	AF	1	[Codigo de op.]	$(A) \leftarrow (A) \oplus (A)$	Z=1 CY es cero
Ligan A con L por medio de un XOR exclusivo	REGISTRO	XAA L	AD	1	[Codigo de op.]	$(A) \leftarrow (A) \oplus (L)$	Z CY es cero
Ligan A con H por medio de un XOR exclusivo	REGISTRO	XAA H	AC	1	[Codigo de op.]	$(A) \leftarrow (A) \oplus (H)$	Z CY es cero
Ligan A con L y H con un XOR exclusivo	INDIRECTO DE REGISTRO	XAA B	AE	1	[Codigo de op.]	$(A) \leftarrow (A) \oplus ((B)(L))$	Z CY es cero
Complemento de A	INHERENTE	CMA	2F	1	[Codigo de op.]	$(A) \leftarrow \bar{A}$	
Rotar A hacia la derecha con un acarreo	INHERENTE	RA2	1F	1	[Codigo de op.]		CY
Rotar A hacia la izquierda con un acarreo	INHERENTE	RAL	17	1	[Codigo de op.]		CY

() contenido
() direccionamiento indirecto de registro
AND
OR
XOR

TABLA 2

INSTRUCCIONES DE TRANSFERENCIA DE DATOS

DESCRIPCION DE LA OPERACION	MODO DE DIRECCIONAMIENTO	MNEMONICO	CODIGO DE OPERACION	BYTES	FORMATO DE LA INSTRUCCION	SIMBOLOGICO	BANDERAS QUE SE AFECTAN
Nueva L a A	REGISTRO	NOU L,A	7D	1	Codigo de op.	(A) ← (L)	
Nueva H a A	REGISTRO	NOU A,H	7C	1	Codigo de op.	(A) ← (H)	
Nueva A a L	REGISTRO	NOU L,A	6F	1	Codigo de op.	(L) ← (A)	
Nueva A a H	REGISTRO	NOU H,A	67	1	Codigo de op.	(H) ← (A)	
Nueva HL a PC	REGISTRO	PCHL	E9	1	Codigo de op.	(PC) ← (HL)	
Nueva HL a SP	REGISTRO	SPHL	F9	1	Codigo de op.	(SP) ← (HL)	
Carga A con datos	INMEDIATO	HUI A	3E	2	Codigo de op. datos	(A) ← (byte 2)	
Carga L con datos	INMEDIATO	HUI L	2E	2	Codigo de op. datos	(A) ← (byte 2)	
Carga H con datos	INMEDIATO	HUI H	26	2	Codigo de op. datos	(A) ← (byte 2)	
Carga LOC(H y L) a A	INDIRECTO DE REGISTRO	NOU A,H	7E	1	Codigo de op.	(A) ← ((H)(L))	
Carga HL con datos	INMEDIATO	LVI H	21	3	Codigo de op. dato menor dato mayor	(L) ← (byte 2) (H) ← (byte 3)	
Carga SP con datos	INMEDIATO	LXI SP	31	3	Codigo de op. dato menor dato mayor	(SP) ← (byte 2+3)	
Carga HL de LOC aa	DIRECTO	LHLD	2A	3	Codigo de op. direc menor direc mayor	(L) ← (byte 2+3) (H) ← ((byte 2+3)+1)	
Carga A de LOC aa	DIRECTO	LDA	3A	3	Codigo de op. direc menor direc mayor	(A) ← (byte 2+3)	

() CONTENIDO DE
 (L) DIRECCIONAMIENTO INDIRECTO DE REGISTRO
 H) CONTENIDO DE PROGRAMA
 SP) APUNTADES DE PILA

INSTRUCCIONES DE TRANSFERENCIA DE DATOS (CONTINUACION)

Almacena A en LUC aa	DIRECTO	STA	32	3	Código de op. direc menor direc mayor	(Direccion) ← (A)	
Almacena KL en LUC aa	DIRECTO	SMLD	22	3	Código de op. direc menor direc mayor	(Direccion) ← (L) (Direccion +1) ← (H)	
Almacena A en LUC (H v L)	INDIRECTO DE REGISTRO	NOV H,A	77	1	Código de op.	((H)(L)) ← (A)	
Almacena L en LUC (H v L)	INDIRECTO DE REGISTRO	NOV H,L	75	1	Código de op.	((H)(L)) ← (L)	
Almacena H en LUC (H v L)	INDIRECTO DE REGISTRO	NOV H,H	74	1	Código de op.	((H)(L)) ← (H)	
Entrada de A al puerto en LUC a	DIRECTO	IN	DB	2	Código de op. dir de puerto	(A) ← (Direccion del puerto)	
Salida de A al puerto en LUC a	DIRECTO	OUT	D3	2	Código de op. dir de puerto	(Direccion del puerto) ← (A)	
Coloca bandera de aborteo	IMMEDIATA	STC	37	1	Código de op.	(CY) ← 1	CY es colocado en 1

() CONTENIDO DE
() DIRECCIONAMIENTO INDIRECTO DE REGISTRO
R CONTADOR DE PROGRAMAS
P AFUENTADOR DE FILA

INSTRUCCIONES DE LLAMADO Y REGRESO DE SUBROUTINAS

DESCRIPCION DE LA OPERACION	MODO DE DIRECCIONAMIENTO	ABNEMONICO	CODIGO DE OPERACION	BYTES	NUMERO DE LA INSTRUCCION	SIMBOLOGICO	NUMERO QUE SE AYUDA
Llama a la Subrutina en LUC aa	INMEDIATO/INDIRECTO DE REGISTRO	CALL	CD	3	Código de op. direc menor direc mayor	((SP) -1) ← (PC) ((SP) -2) ← (PC) (SP) ← (SP) -2 (PC) ← (Direccion)	
Regreso de la Subrutina	INDIRECTO DE REGISTRO	RET	C9	1	Código de op.	(PC) ← ((SP)) (PC) ← ((SP) +1) (SP) ← (SP) +2	

() CONTENIDO DE
() DIRECCIONAMIENTO INDIRECTO DE REGISTRO
R CONTADOR DE PROGRAMAS
P AFUENTADOR DE FILA

INTRODUCCIONES DE INSTRUCCIONES

DESCRIPCION DE LA OPERACION	MODO DE DIRECCIONAMIENTO	MNEMONICO	CODIGO DE OPERACION	BYTES	FORMATO DE LA INSTRUCCION	SIMBOLOGICO	BANDERAS QUE SE AFECTAN
Principio a LOC aa	INMEDIATO	JMF	CC	3	Codigo de op. direc menor direc mayor	(PC) ← (Direccion)	
Principio a LOC aa si es cero	INMEDIATO	JC	CA	3	Codigo de op. direc menor direc mayor	Si la bandera Z=1, entonces (PC) ← (Direccion)	
Principio a LOC aa si no es cero	INMEDIATO	JMC	CC	3	Codigo de op. direc menor direc mayor	Si la bandera Z=0, entonces (PC) ← (Direccion)	
Principio a LOC aa si se coloca la bandera de acarreo	INMEDIATO	JC	DA	3	Codigo de op. direc menor direc mayor	Si la bandera CF=1, entonces (PC) ← (Direccion)	
Principio a LOC aa si no se coloca la bandera de acarreo	INMEDIATO	JMC	DZ	3	Codigo de op. direc menor direc mayor	Si la bandera CF=0, entonces (PC) ← (Direccion)	

() CONTENIDO DE
PC CONTADOR DE PROGRAMA

INSTRUCCIONES VARIAS

DESCRIPCION DE LA OPERACION	MODO DE DIRECCIONAMIENTO	MNEMONICO	CODIGO DE OPERACION	BYTES	FORMATO DE LA INSTRUCCION	SIMBOLOGICO	BANDERAS QUE SE AFECTAN
Introduce A y banderas a la fila	INDIRECTO DE REGISTRO	PUSH PSW	FS	1	Codigo de op.	((SP) - 1) ← (A) ((SP) - 2) ← (banderas) (SP) ← ((SP) - 2)	
Introduce HL a la fila	INDIRECTO DE REGISTRO	PUSH H	ES	1	Codigo de op.	((SP) - 1) ← (HL) (SP) ← ((SP) - 2)	
Extrae A y banderas de la fila	INDIRECTO DE REGISTRO	POP PSW	F1	1	Codigo de op.	(banderas) ← ((SP)) (A) ← ((SP) + 1) (SP) ← ((SP) + 2)	
Extrae HL de la fila	INDIRECTO DE REGISTRO	POP H	E1	1	Codigo de op.	(L) ← ((SP)) (H) ← ((SP) + 1) (SP) ← ((SP) + 2)	
No Operacion	INHERENTE	KOP	00	1	Codigo de op.	(PC) ← ((PC) + 1)	
etc.	INHERENTE	HLT	76	1	Codigo de op.		

() CONTENIDO DE
() DIRECCIONAMIENTO INDIRECTO DE REGISTRO
() ACUMULADOR
P SP=7600H DE 71...

En la figura V-3 se observa que el par de registros HL almacena una dirección de 16 bits. El par de registros HL apunta entonces a la posición apropiada de memoria de datos (etiqueta LOC). Los contenidos de esta posición (00000011₂) se suman a los del acumulador (01100000₂). El resultado, como en todas las operaciones de suma, se almacena en el acumulador. Las instrucciones de registro indirecto utilizan un registro de 16 bits (normalmente un par de registros HL) como un apuntador de dirección.

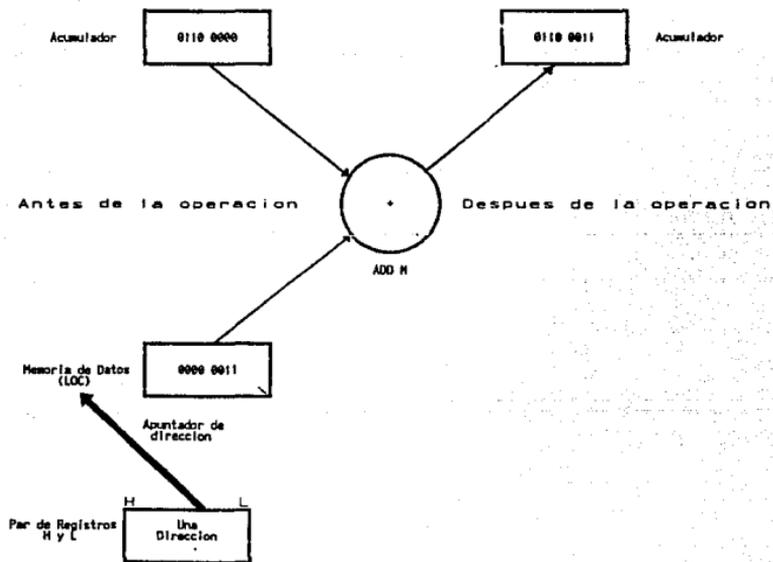
La notación simbólica para la instrucción ADD M se lee de derecha a izquierda. Del lado derecho de la flecha, el primer operando se identifica con los contenidos de A. En esta notación los paréntesis () significan "contenidos de". El signo + significa la operación que ha de ser realizada, que en este caso es la suma.

El doble paréntesis (()) denota una instrucción indirecta de registro. ((H)(L)) significa "el segundo operando reside en una posición de memoria apuntada por los contenidos del par de registros HL. La dirección de la flecha indica en donde se almacenará el resultado de la operación, que en este caso es el acumulador A.

Dentro de las instrucciones de resta, que son cuatro, el microprocesador utiliza el método de complemento a dos en el substraendo, ya que internamente no presenta hardware de resta. Un ejemplo de complemento a dos se presenta en la figura V-4. El número binario 00000001₂ está siendo restado de 00001001₂ (09H-01H = 08H). La resta binaria puede ser realizada sumando (09H-01H = 08H). La resta binaria puede ser realizada sumando el minuendo binario a la forma de complemento a 2 del substraendo sin considerar los residuos. El substraendo en este ejemplo es 00000001₂ que convertido a su forma complementaria de 2:

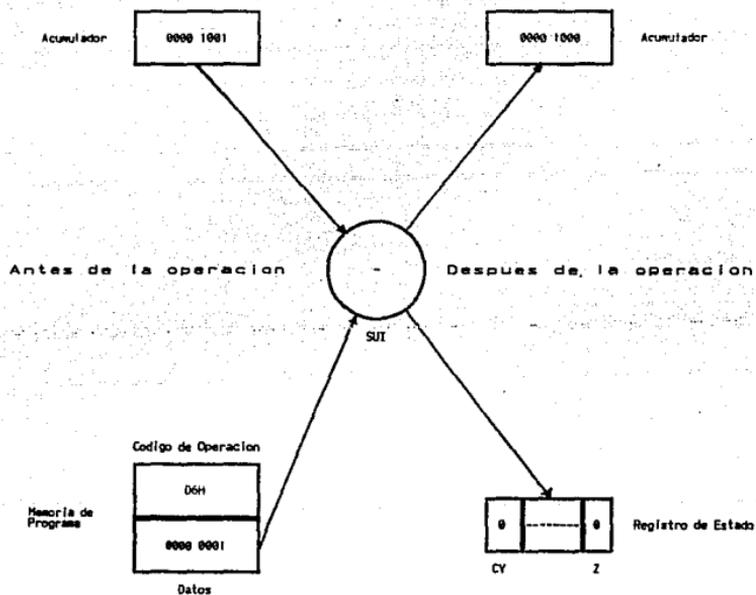
$$\begin{array}{r}
 00000001 \text{ -----} 11111110 \text{ complemento de 1} \\
 11111110 + 1 = 11111111 \text{ complemento de 2}
 \end{array}$$

El complemento de 2 (11111111) del substraendo se suma entonces al minuendo, y resulta una suma de 1 0000 1000. El 1 en la posición más significativa de la suma es un residuo y no es parte de la diferencia 00001000₂. El microprocesador utiliza los residuos para afectar la bandera



Suma indirecta de registro

Figura V-3



Resta los datos de A (SUI)

Figura V-4

de acarreo (o "prestado"). En la operación de resta, la CPU invierte el residuo y éste se convierte a los contenidos de la bandera de acarreo (CY). En el ejemplo, el residuo de 1 es invertido y restaura la bandera de acarreo a 0.

Cuando la bandera de acarreo es restaurada en un problema de resta, significa que no ocurrió ningún préstamo o que el minuendo es mayor que el substraendo.

Si en la operación anterior se utilizara la instrucción "resta inmediata" (SUI), los datos inmediatos del segundo byte de la memoria de programa (00000001₂) son restados de los contenidos del acumulador (00001001₂), la diferencia es 00001000₂ es transferida al acumulador después de la resta. La bandera de acarreo es restaurada a 0 (cero), lo que indica que no hubo ningún residuo o que el número del acumulador antes de la operación era mayor que el de la memoria. La bandera de 0 (cero) prueba los contenidos del acumulador después de la operación de resta.

Cuando el minuendo es menor que el substraendo, como ejemplo, restar 00000110₂ del número 00000101₂ (05H - 06H = FFH ó -1₁₀).

Primero se saca el complemento a dos del substraendo:

$$\begin{aligned} 00000110 & \text{-----} 11111001 \text{ complemento de 1} \\ 11111001 + 1 & = 11111010 \text{ complemento de 2} \end{aligned}$$

El minuendo (00000101₂), se suma entonces a la forma complementaria de 2 del substraendo (11111010₂), de lo que resulta una diferencia de 11111111₂ es la representación complementaria de 2 de -1₁₀. No hubo residuo en la suma, o un cero en el registro de residuo. Pero internamente el microprocesador, para la operación de resta, invierte este residuo resultando un 1 colocándolo en la bandera de acarreo CY. Si la bandera de acarreo está posicionada en 1, significa que el número del acumulador después de la resta es un número negativo. El número 11111111₂ con la bandera CY en 1, representan al número -1₁₀.

El microprocesador incluye cuatro instrucciones de comparación, estas instrucciones restan la memoria o el contenido del registro de los contenidos del acumulador, pero no cambian los

contenidos de ninguno de ellos. Se afectan las banderas por las instrucciones de comparación.

La instrucción "comparar el registro L" (CMP L), afecta las banderas CY (acarreo) y Z (cero). Si se tiene un número en el acumulador igual al que está en el registro L, la bandera CY se queda en 0 (cero), mientras que la bandera Z se coloca en 1. Estas condiciones se pueden observar en la columna de banderas de la tabla I. Si se diera el caso en que el acumulador fuera menor que el registro L, la bandera CY se colocará en 1 y la bandera Z se colocaría en 0. Un ejemplo se muestra en la figura V-5, al comparar con la instrucción CMP L el byte 00010000₂ contenido en el acumulador con el dato 00010000₂ contenido en el registro L.

Las instrucciones restantes de comparación funcionan de la misma manera. La instrucción CPI (Compara A con los datos) es de 2 bytes, uno es el código de operación (C6H) y el otro son los datos que serán comparados con cualquier otro dato, pero que no esté contenido dentro de los registros H y L.

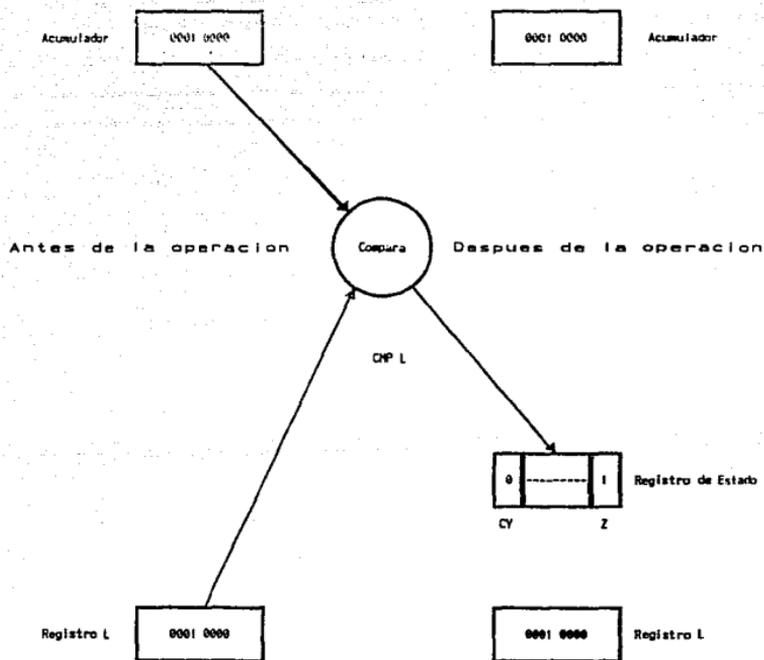
La instrucción CMP H funciona idénticamente a la instrucción CMP L.

Finalmente, la instrucción CMP M direcciona una localidad de memoria indicada por el par de registros H y L, donde se encuentra un dato para compararlo con el dato almacenado en el acumulador.

Las operaciones de incrementa y decrementa, se utilizan para acceder la localidad de memoria siguiente o anterior, según sea el caso, y leer o escribir datos en las direcciones que señalen el par de registros H y L. También se utilizan para incrementar o decrementar el dato almacenado en el acumulador.

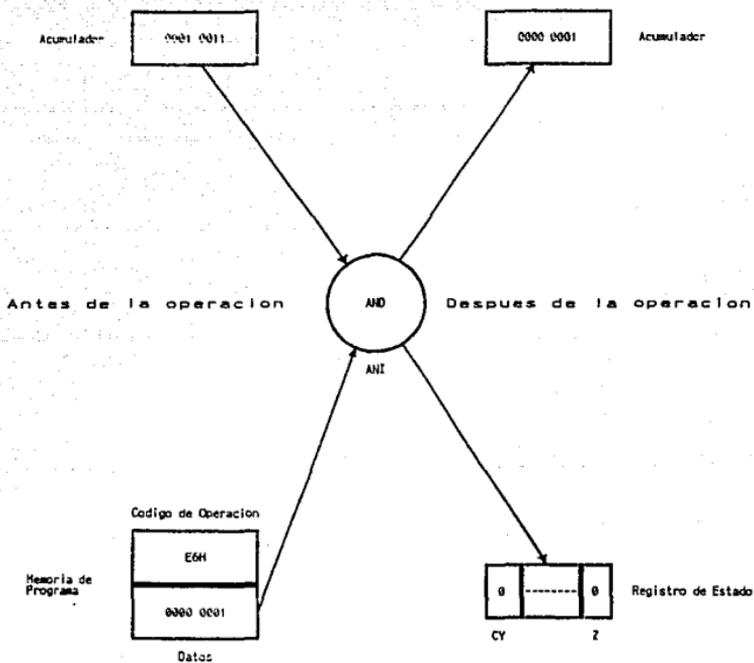
OPERACIONES LOGICAS: Incluyen AND, OR, OR Exclusivo, complemento (NOT) e instrucciones de rotación. Se pueden observar en la tabla II. El enfoque en la mayoría de las operaciones se encuentra en los contenidos del acumulador. Igual que con las instrucciones aritméticas, el modo de direccionamiento tiene que ver con cómo y dónde se encuentran otros datos en el sistema basado en microprocesador.

Un ejemplo se muestra en la figura V-6, con la instrucción ANI (AND inmediato). Los contenidos del acumulador (00010011₂) son ligados por medio de la operación AND con el segundo byte de datos de la memoria del programa (00000001₂). Los números son ligados



Compara A con L (CMP L)

Figura V-5



Instrucción AND Inmediata (ANI)

Figura V-6

mediante el AND bit por bit. En este ejemplo la salida será 10000001₂. El resultado de la operación AND se coloca en el acumulador. Para todas las operaciones AND del microprocesador, la bandera de acarreo está colocada en 0 (cero). La bandera de cero (Z), para estas instrucciones, funciona de la manera siguiente: Si el resultado de la operación AND es cero (0000000₂), la bandera Z se coloca en 1, si el resultado es diferente de cero (en el ejemplo fue 0000001₂), la bandera Z se coloca en cero.

Las instrucciones ANA L y ANA H, realizan la operación AND directamente con los datos del registro que se trate (L o H), con los datos del acumulador.

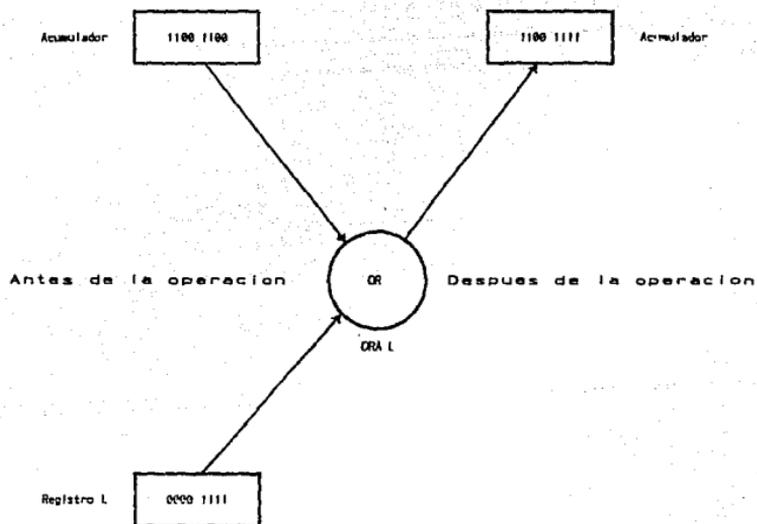
La instrucción "AND indirecta de registro" (ANA M), liga por medio de la instrucción AND (bit por bit) a los contenidos de la posición de memoria de datos apuntada por el par de registros HL, con los datos que registra el acumulador. El resultado se almacena en el acumulador, desplazando el dato anterior.

En esta operación se puede utilizar el número binario 0000001₂ como una "máscara", ésta puede servir para restaurar los 7 bits más significativos a 0 (cero), y también una forma de manejar la bandera Z.

Las cuatro instrucciones OR que se encuentran en la tabla II, unen los contenidos del acumulador a través de un OR, con alguna otra posición de memoria o registro. Al querer sumar dos números, empleando la instrucción ORA L, por ejemplo 11001100₂ guardado en el acumulador, con el número 00001111₂ contenido en el registro L en este caso. Los números son ligados mediante un OR, bit por bit, resultando el número 11001111₂ que se almacena en el acumulador.

Como se observa en la figura V-7.

Existen cinco instrucciones OR exclusivo. La operación XRA A ocasiona que los contenidos del acumulador sean ligados a través de un OR exclusivo, bit por bit, consigo mismo, de acuerdo a la tabla de verdad conocida para esta operación.. Aplicar la instrucción OR exclusivo a cualquier número consigo mismo ocasiona un resultado de 0000000₂. La bandera Z está siempre colocada en 1, indicando ceros en el acumulador. La bandera CY permanece, al igual que en la AND y la OR, en 0 (cero).



Instrucción de ligar A con L mediante un OR (ORA L)

Figura V-7

Las instrucciones RAR y RAL se utilizan para rotar los contenidos del acumulador hacia la derecha o a la izquierda respectivamente, empleando la técnica de acarreo. Si se tiene en el acumulador el dato 00110001_2 y se aplica la instrucción RAR, los contenidos serán rotados a una posición a la derecha con el bit menos significativo (1 en este ejemplo) transferido a la posición de acarreo de bits (bandera CY). La bandera Z no interviene en esta operación. Inicialmente la bandera CY almacenaba un 0 (en este ejemplo), con la instrucción RAR este acarreo entrará en la posición más significativa del acumulador. Después de la operación el acumulador almacena el dato 00011000_2 , colocándose la bandera de acarreo en 1. Ejemplificado el procedimiento anterior en la figura V-8.

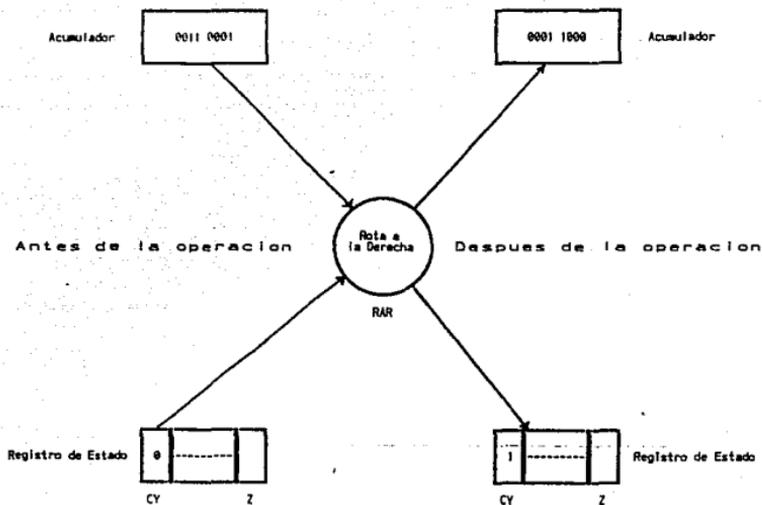
Con una o más instrucciones de rotación, cualquier posición de bit puede ser probada y puede colocarse o restaurarse una bandera de acarreo. La bandera de acarreo puede probarse entonces a través de una instrucción de bifurcación condicional (JZ, JNZ, JC, JNC). Otro uso de las instrucciones de rotación es la prueba de paridad. Una paridad non sería un número binario con una cantidad non de 1's en la palabra.

El modo de direccionamiento de las instrucciones de rotación es inherente, ya que operan solamente en el acumulador y no necesitan un operando de otro registro o posición de memoria.

La operación "complemento de A" (CMA), se utiliza únicamente para invertir, uno a uno, cada bit de la palabra almacenada en el acumulador.

Las instrucciones lógicas sirven para realizar manipulaciones de álgebra booleana sobre las variables. También se utilizan para colocar o blanquear bits específicos dentro de una palabra, o para probar o comparar bits.

INSTRUCCIONES DE TRANSFERENCIA DE DATOS: Estas incluyen el movimiento de registro a registro, la carga de la memoria, el almacenamiento en memoria, la entrada y salida y las instrucciones de colocar la bandera de acarreo. La mayoría de los programas contienen más instrucciones de transferencia de datos que otros tipos de operación. Todas estas instrucciones trabajan sin afectar o ser afectadas por las banderas, a excepción de la instrucción STC "coloca bandera de acarreo". Existen muchas instrucciones en este grupo, por lo que los datos pueden ser transferidos de una posición de memoria a cualquier registro o de cualquier registro a cualquier



Rotar A hacia la derecha por medio de acarreo

Figura V-8

posición de memoria u otro registro. Los microprocesadores con más registros en la CPU tienen más instrucciones de transferencia de datos. Cada instrucción de transferencia de datos tendrá una fuente de datos y un destino para ellos. El modo de direccionamiento tiene que ver con cómo y dónde se encuentra la fuente de estos datos.

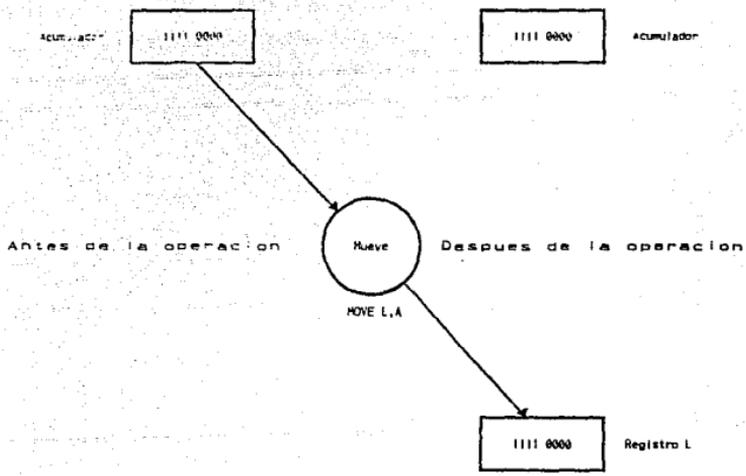
La primera operación de la tabla III (MOV A,L), mueve los datos del registro L al registro A (acumulador). Es una convención de Intel Corporation el hecho de que en el mnemónico se escriba primero el destino y después la fuente (A,L). Ver la figura V-9.

Todas las instrucciones utilizadas para mover datos, funcionan con el mismo principio de la instrucción MOV A,L.

EN la instrucción "mueve HL a SP" (SPHL), la fuente de los datos es el par de registros HL de 16 bits, mientras que el destino es el apuntador de fila (Stack Pointer SP), también de 16 bits. Esto se observa en la figura V-10.

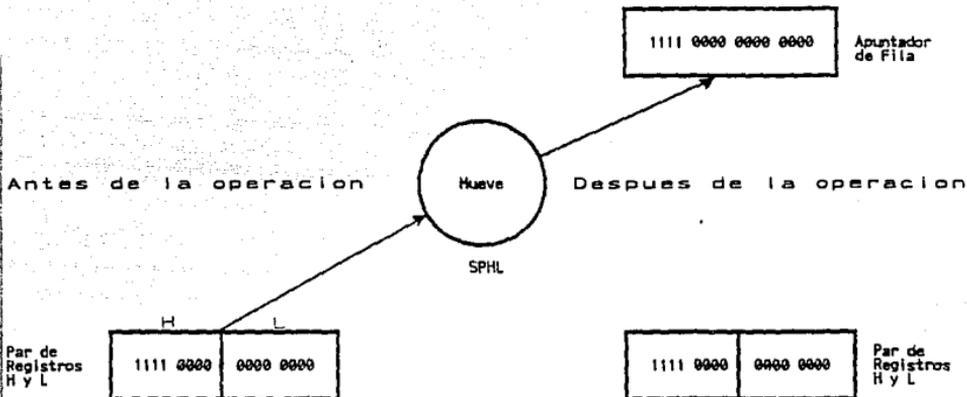
Existen cinco instrucciones de "carga inmediata", que son muchas veces utilizadas para colocarle un valor inicial al registro del microprocesador antes del programa. Un ejemplo se muestra con la instrucción "carga HL con datos", (LXI H), como se muestra en la figura V-11 y que requiere de 3 bytes. El par de registros HL de 16 bits va a ser cargado con datos inmediatos de la memoria de programa. El primer byte es el código de operación (21H), mientras que los siguientes 2 bytes de memoria de programa son bytes de datos. El segundo byte almacena los datos de menor orden, cargado en el registro L (LOW). El tercer byte almacena los datos de orden alto, cargados en el registro H (HIGH).

La instrucción "carga directa HL" (LHLD) utiliza el modo de direccionamiento directo. El segundo y tercer byte de la memoria del programa son direcciones de 16 bits de las posiciones de memoria de datos, en donde se encontrarán los datos que serán cargados. En la figura V-12 el código de operación de esta instrucción es el 2AH, utilizado por el primer byte. La CPU ensambla los siguientes dos bytes del programa dentro de una dirección de 16 bits (mostrada en la figura con líneas punteadas). Esta dirección de 16 bits sirve como un apuntador de la dirección para ubicar una posición (LOC) en la memoria de datos. Este byte de memoria de datos se carga entonces dentro del registro L. Los contenidos de la siguiente posición de memoria de datos



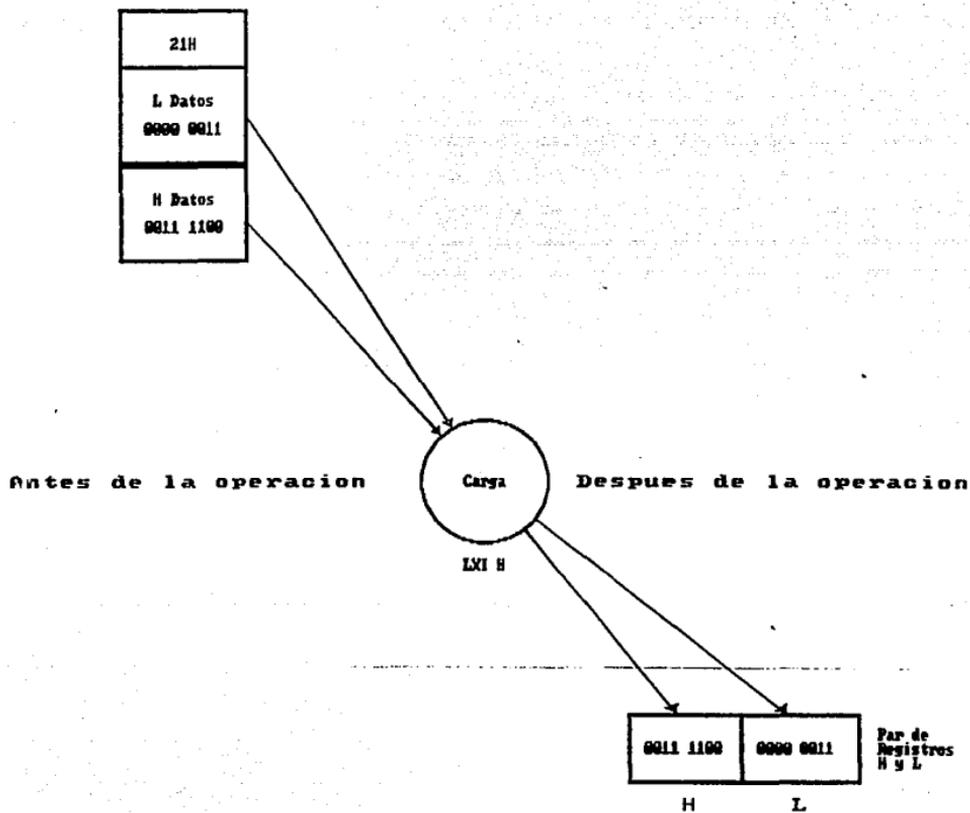
Instrucción Mueve A a L (MOVE L, A)

Figura V-9



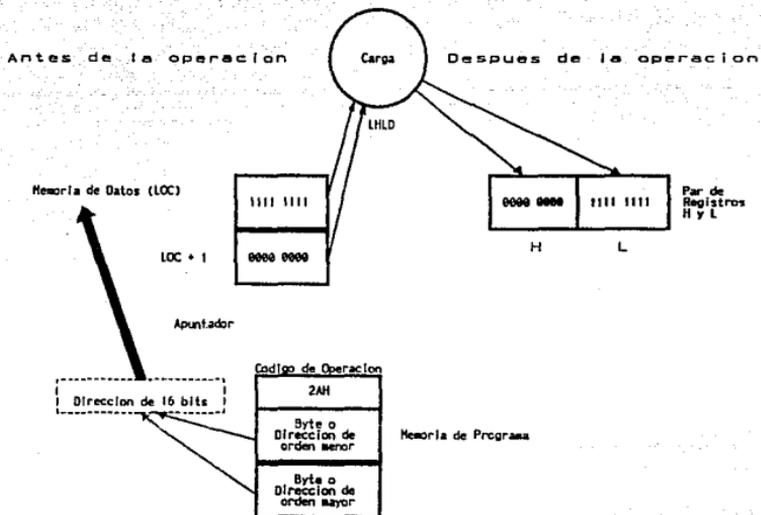
Instrucción Mueve HL a SP (L, A)

Figura V-10



Instrucción Carga Inmediata de HL

Figura V-11



Instrucción Carga Directa de HL

Figura V-12

(etiquetada como LOC + 1 en este ejemplo), se cargan posteriormente en el registro H.

El microprocesador maneja cinco instrucciones de almacenamiento, las que se utilizan para depositar los resultados y otros datos en memoria. Un ejemplo se presenta al utilizar la instrucción "almacenamiento directo de A" (STA) (figura V-13). Los contenidos del acumulador se almacenan en la posición de memoria de datos (LOC) que apunta a ella a través de la dirección de 16 bits formada por el segundo y tercer bytes de la instrucción. Después de la operación de almacenamiento, tanto la posición de memoria de datos (LOC) como el acumulador almacenan los mismos datos 11000000₂ en este ejemplo).

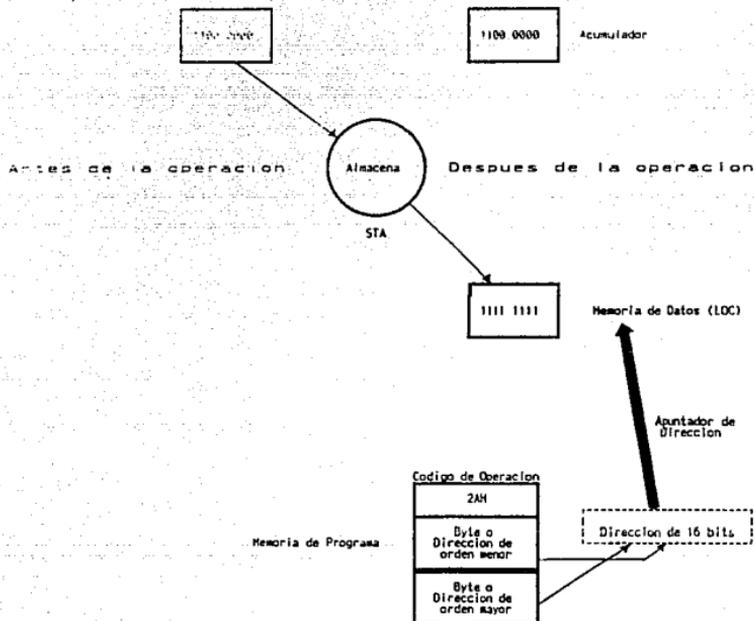
Otras instrucciones de transferencia de datos que son de importancia son las de entrada y salida (IN y OUT). La instrucción IN es semejante a la operación de carga. La fuente de los datos que serán transferidos es un puerto de entrada identificado por un número binario de ocho bits (0-255₁₀). El destino es el acumulador de la CPU. Un ejemplo se muestra en la figura V-14. Los datos del puerto de entrada que son apuntados por el segundo byte en la instrucción se transfieren al acumulador del microprocesador. En el ejemplo, 00001111₂ es la entrada y es transferida al acumulador desde un puerto identificado como LOC.

La instrucción OUT es exactamente el procedimiento inverso de la instrucción IN. Aquí la fuente es el acumulador y el destino es el puerto de salida, que es direccionado por la memoria de programa.

Dependiendo de si la comunicación es serie o paralelo, el microprocesador envía señales a la interfaz respectiva, para poder desplazar los datos en la forma conveniente.

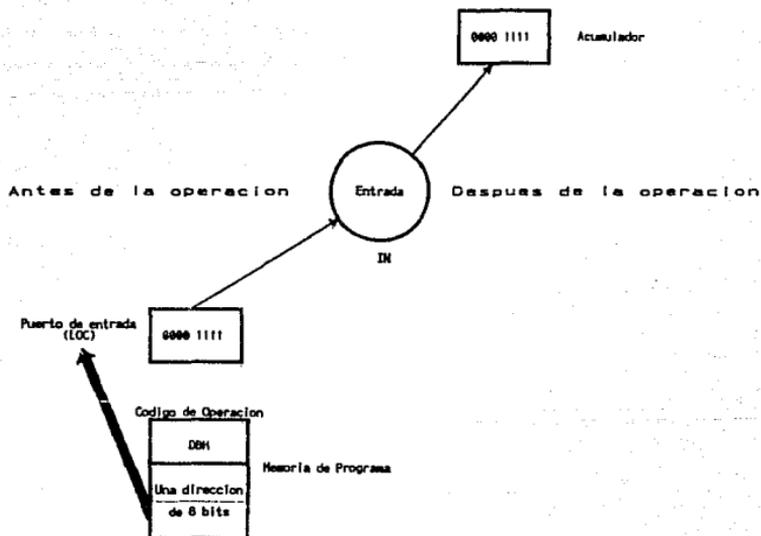
OPERACIONES DE BIFURCACION: Las operaciones de bifurcación se conocen también como instrucciones de brinco o instrucciones de control de transferencia.

La microcomputadora normalmente ejecuta instrucciones de orden secuencial. El contador de programa de 16 bits del microprocesador siempre contiene la dirección de la siguiente instrucción que será rastreada de la memoria y ejecutada. El contador de programa normalmente está secuenciado hacia arriba, un número a la vez. Las instrucciones de bifurcación o brinco proporcionan un método de cambiar el valor del contador de programa; por lo tanto, alteran la secuencia normal de la ejecución del programa.



Instrucción Almacenamiento Directo de A

Figura V-13



Instrucción de Carga Inmediata de HL

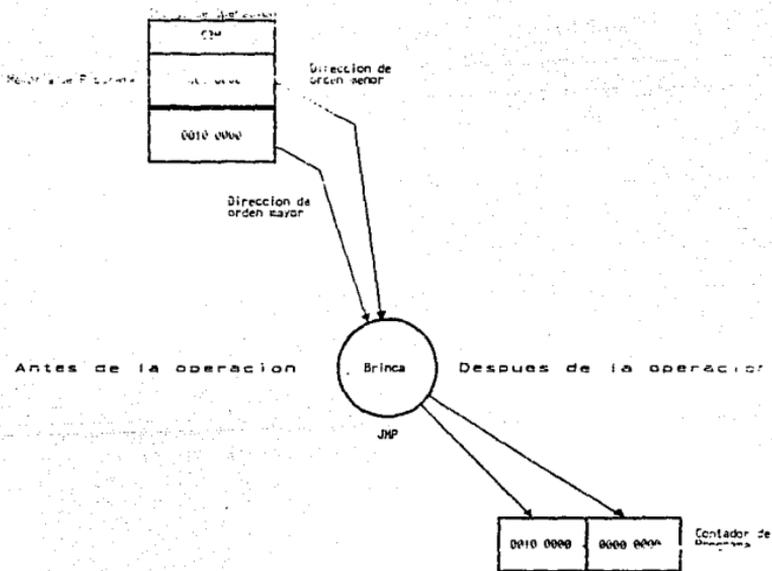
Figura V-14

Las operaciones de bifurcación son generalmente subdivididas en operaciones de brinco condicional y de brinco incondicional. La primera instrucción de la tabla III, la instrucción "brinco inmediato" (JMP), es de brinco incondicional. Es de 3 bytes y se usa para cargar una dirección específica dentro del contador del programa de la CPU. Un problema muestra para esta instrucción de brinco se ilustra en la figura V-15. Aquí la dirección 2000H está siendo cargada dentro del contador de programa. La información sobre la dirección sigue inmediatamente al código de operación y, por lo tanto, se llama direccionamiento inmediato. En la figura se observa que la parte de orden menor de la dirección de 16 bits se localiza en el segundo byte en memoria de programa, mientras que el tercer byte almacena la parte de orden mayor de la dirección.

La instrucción de bifurcación o brinco incondicional se puede emplear como un modo de cargar nueva información de direcciones en el contador de programa.

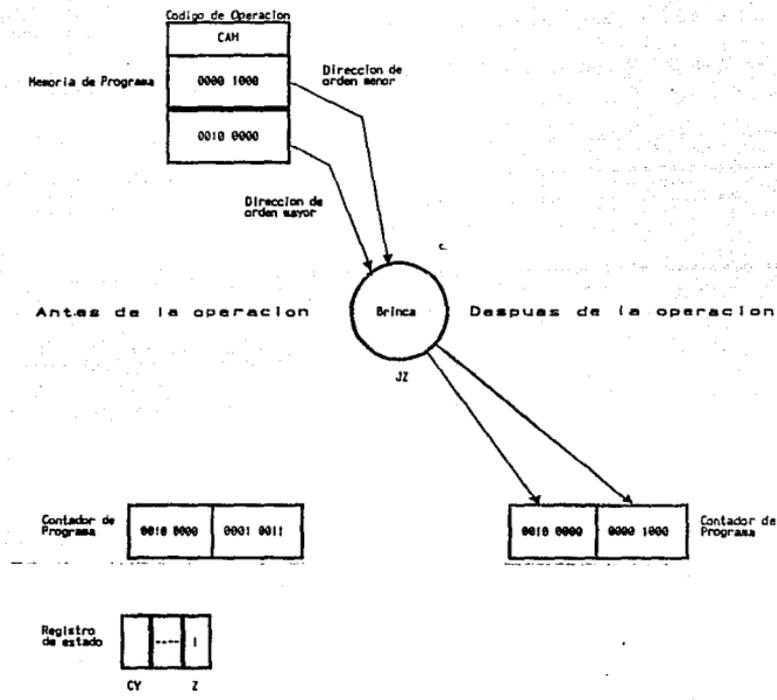
Las siguientes operaciones de bifurcación son instrucciones de brinco condicional. Estas ocasionarán la inmediata colocación de la dirección en el contador de programa "sólo si se satisface la condición especificada". De otra forma, el contador de programa simplemente seguirá su secuencia ascendente.

En la figura V-16 se diagrama un problema muestra que usa la instrucción "brinca si es 0" (JZ). En el ejemplo el contador de programa se encuentra en la dirección 2013H antes de la operación y continuará en la secuencia hacia arriba desde aquí, a menos que la bandera de ceros se coloque en uno ($Z=1$). La CPU evalúa la condición de la bandera de ceros en el registro de estado y encuentra que está colocada en uno. Esto significa que el último resultado de las operaciones aritméticas o lógicas fue un cero. Las condiciones son correctas para un brinco y el microprocesador carga el contador de programa con la nueva dirección de 2008H. Esta nueva dirección viene de la memoria de programa. Por lo tanto, esta instrucción utiliza direccionamiento inmediato. La siguiente instrucción que se ejecutará es la instrucción en la posición de memoria 2008H y no la 2013H, como podría esperarse. El microprocesador brinca a una nueva posición en la memoria del programa. Los brincos o bifurcaciones pueden hacerse hacia arriba o hacia abajo en la memoria del programa. El ejemplo indica un brinco hacia abajo, que es probablemente el más común.



Instrucción Brinca Incondicional (JMP)

Figura V-15



Instrucción Brinca si es cero (JZ)

Figura V-16

Las instrucciones de bifurcación o brinco se encuentran en casi cualquier programa del microprocesador. Son valiosas por que tienen una característica de toma de decisiones. Se utilizan para formar iteraciones en los programas.

OPERACIONES DE LLAMADO DE SUBROUTINAS Y DE REGRESO: Solamente se tienen dos instrucciones en este grupo. Las instrucciones de llamado y de regreso son siempre utilizadas en pares. La instrucción de llamado de 3 bytes se utiliza en el programa principal para hacer que el microprocesador brinque a una subrutina. En el ejemplo de la figura V-17, la subrutina es un grupo corto de instrucciones que ocasionan una demora de tiempo de un segundo. Cuando el microprocesador encuentra la primera instrucción de llamada en la posición 200DH, encuentra la dirección para el brinco en los siguientes dos bytes del programa. La dirección de la instrucción que sigue al llamado se almacena en la fila. El microprocesador brinca entonces al inicio de la subrutina en la posición 1000H, las instrucciones en la subrutina de retraso de tiempo de un segundo se realizan hasta que la CPU encuentra la instrucción de regreso. La dirección almacenada en la fila (2010H) se regresa al contador de programa y el microprocesador continua el programa principal en donde lo dejó cuando se llamo la subrutina por primera vez. El microprocesador ejecuta las instrucciones en orden hasta que encuentra la instrucción de segunda llamada en la dirección 2020H. El microprocesador almacena la dirección de la siguiente instrucción (2023H) en la fila y brinca a la subrutina en la dirección 1000H. Cuando nuevamente se completa la subrutina de retraso de tiempo, la instrucción de regreso extrae la dirección de la siguiente instrucción del programa principal (2023H) fuera de la fila y la transfiere al contador de programa. Una subrutina puede utilizarse varias veces en la ejecución de un programa.

La subrutina es almacenada en RAM o en ROM.

La instrucción de llamada combina las funciones de las operaciones de introducción y brinco. En la figura siguiente (V-18), se observa el uso de la instrucción de llamado. Primero introduce los valores actuales del contador de programa en la fila.

Enseguida el contador de programa debe ser cargado con una nueva dirección para desarrollar el brinco hacia la operación de la subrutina. La secuencia de acciones se sigue de acuerdo a los números encerrados en círculo:

Programa Principal

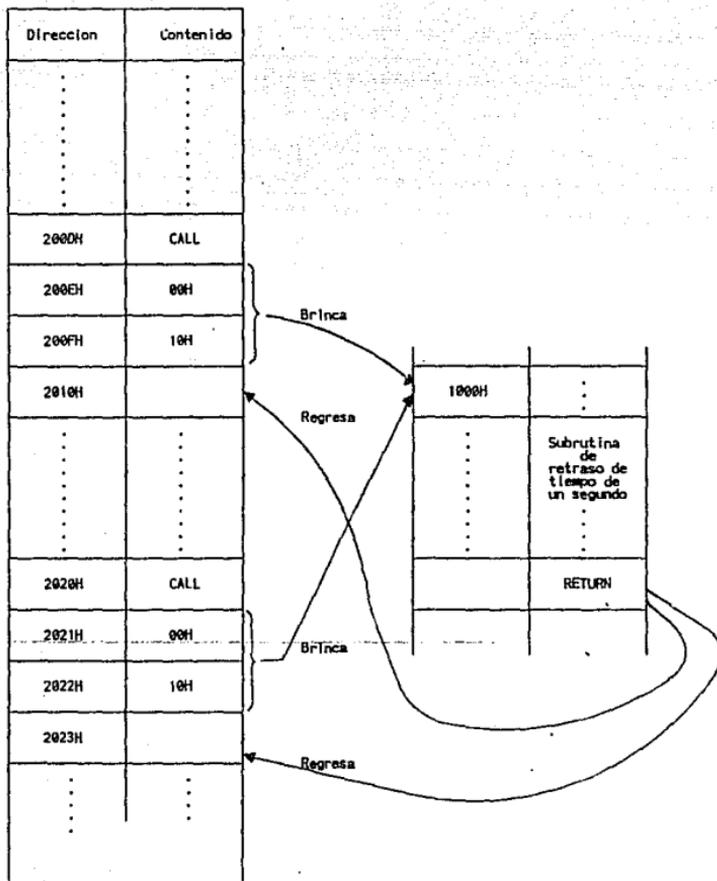
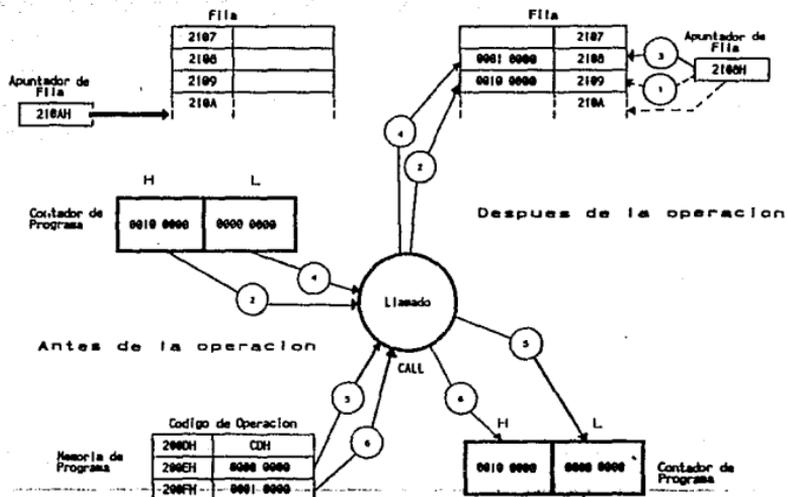


Figura V-17



Llamado de Subrutina (CALL)

Figura V-18

- 1.- El apuntador de fila decrece de 210AH a 2109H.
- 2.- El byte de orden mayor del contador de programa es introducido a la fila en la posición 2109H.
- 3.- El apuntador de fila decrece de 2109H a 2108H.
- 4.- El byte de orden menor del contador de programa es introducido en la fila a la posición 2108H.
- 5.- La dirección de orden menor se mueve del segundo byte de la memoria de programa al byte de orden menor del contador de programa.
- 6.- La dirección de orden mayor se mueve al tercer byte de la memoria del programa al byte de orden mayor del contador de programa.

El microprocesador debe ir ahora a la nueva dirección a la que apunta el contador de programa (1000H). Esta es la dirección de la primera instrucción de subrutina.

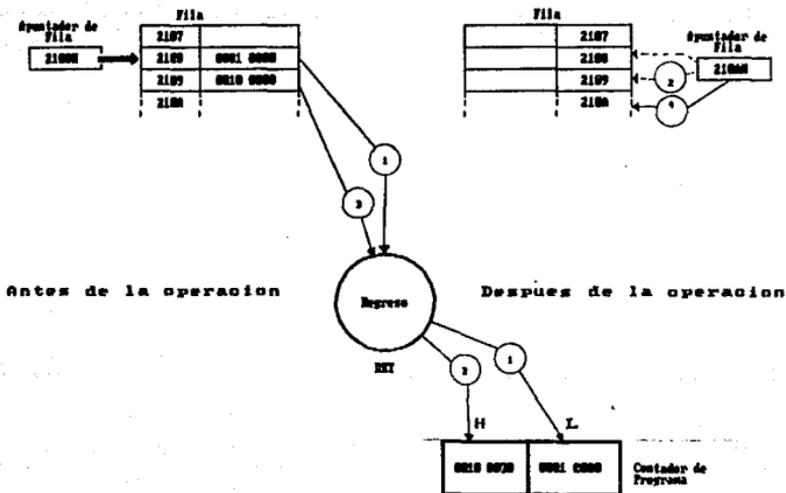
Utilizando el regreso de la instrucción de subrutina, se deben transferir los contenidos de la fila al contador de programa.

La secuencia se observa siguiendo los círculos numerados de la figura V-19:

- 1.- La parte más alta de la fila (posición 2108H) es extraída, transfiriendo los contenidos al byte de orden menor en el contador de programa.
- 2.- El apuntador de fila incrementa de 2108H a 2109H.
- 3.- La parte más alta de la fila (ahora la posición 2109H) es extraída, transfiriendo sus contenidos al byte de orden mayor del contador de programa.
- 4.- El apuntador de fila se incrementa de 2109H a 210AH.

El contador de programa almacena ahora la dirección de 16 bits (2010H) que va a ser rastreada de la memoria de programa.

OPERACIONES VARIAS: Las instrucciones que usan dispositivos de E/S como fuentes o destinos se conocen como instrucciones de entrada y salida (E/S). Hay dos tipos especiales de instrucciones conocidas como inserción (push) y extracción (pop), que se utilizan como auxiliares de las instrucciones de transferencia de datos, para transferir datos hacia o desde un área de memoria fila. La fila se realiza mediante un conjunto de posiciones consecutivas de memoria, a



Regreso de Subrutina (RET)

Figura V-19

las que se accede mediante un registro especial de direcciones, el puntero de la fila (Stack Pointer, SP). Una fila tiene la propiedad de que la última palabra escrita en la fila es la primera que se lee; este método de acceso se conoce como LIFO (last-in/first-out: último en entrar/primer en salir). Tras realizar una operación de inserción o de extracción, SP se ajusta automáticamente apuntando a la siguiente posición consecutiva de la fila. El extremo al que se añaden o del que se extraen datos se conoce como "cabecera" de la fila. Dado que el SP lleva automáticamente la cuenta de la cabecera de la fila, no es necesario incluir direccionamiento de la fila cuando se realiza una operación de inserción o de extracción.

Las instrucciones de introducción y de extracción son ampliamente usadas en la subrutina. Una instrucción "introduce A y banderas en fila (PUSH PSW), puede emplearse como la primera instrucción de la subrutina. Esto debe cubrir los contenidos del acumulador y las banderas de manera que no se vean afectadas por la subrutina. La instrucción "introduce A y banderas de la fila" (PUSH PSW). La parte PSW del mnemónico significa Program Status Word (Palabra de Estado del Programa), que en este caso son los contenidos del acumulador y de los registros de estado y banderas.

Es una instrucción de 1 byte, colocando al acumulador en la primera fila y luego empujando las banderas hacia la fila. Las instrucciones de introducción y de extracción se utilizan en pares porque lo que va en la fila utilizando una operación de introducción debe ser quitado con una operación de extracción.

La instrucción de "no operación" (NOP) no hace nada, excepto consumir uno o dos microsegundos (μs). Es una instrucción de 1 byte que sólo hace que el contador de programa se incremente; no afecta otros registros. Esta instrucción se utiliza como relleno cuando una o dos instrucciones de un programa se quitan durante la depuración. Efectúa un puente entre dos secciones de un programa, de manera que el microprocesador pueda avanzar de un segmento al otro. También puede servir para consumir tiempo en una iteración. La instrucción de "alto" (HLT) se utiliza al final de un programa para evitar que el microprocesador ejecute más instrucciones en la memoria, sólo una petición de restauración o de interrupción puede arrancarlo cuando esté en alto.

CAPITULO VI

INTERFAZ DE MICROCOMPUTADORAS CON DISPOSITIVOS EXTERNOS

INTRODUCCION

La interfaz es el medio natural entre el microprocesador y sus periféricos, o sea, lo que permite la comunicación entre el microprocesador y su entorno. Esta comunicación necesita del hardware, del software o de ambos a la vez. La comunicación se efectúa bien en paralelo, cuando todos los bits se envían al mismo tiempo, o bien en serie, cuando los bits se envían uno tras otro. Los dos principales tipos de interfaz paralelos son el Interfaz Compatible Centronics y el Interfaz IEEE488, que es utilizado para conectar los aparatos de medición. Los dos principales tipos de interfaz serie son el RS232C, empleado en impresoras, modems, fax, etc., y el interfaz de bucle de corriente (20 mA) que se utiliza para las comunicaciones a larga distancia. La comunicación en paralelo presenta algunas desventajas, pues aunque los datos se reciben a mayor velocidad, el hecho de que el microprocesador sea pequeño, limita el crecimiento en las conexiones. La comunicación serie, al presentar la ventaja de su conexión, disminuye su eficiencia en transmisión y recepción de datos.

La comunicación con los dispositivos que el microprocesador requiere para su funcionamiento, también se encuentran bajo el control de la programación, y para ello existen instrucciones específicas que se refieren a la entrada o salida del microprocesador.

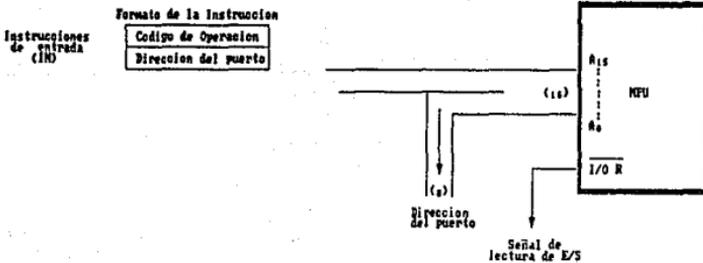
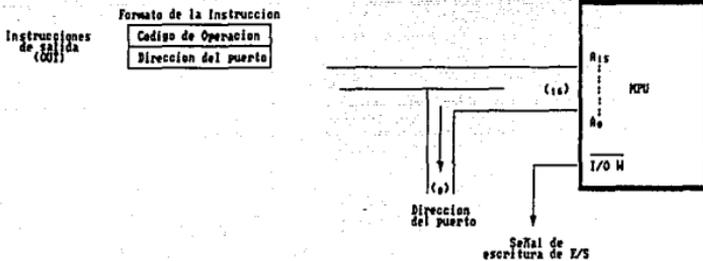
6.1 USO DE LINEAS PROGRAMADAS E/S PARA EL CONTROL DE DISPOSITIVOS

El proceso de unir un dispositivo de E/S a un sistema con microprocesador se denomina interfaz, e implica una fuerte interacción entre el diseño hardware y el diseño software.

Una operación de entrada o salida es el acto de transferencia de datos desde o hacia un dispositivo periférico seleccionado. El microprocesador es el foco de todas esas operaciones; por lo tanto, una entrada significa que los datos fluyen a la unidad microprocesadora y una salida significa que salen de ella. Aquellas posiciones en donde los datos entran o salen, suelen ser llamados puertos de entrada o salida.

En la tabla de instrucciones de transferencia de datos se observa que el microprocesador utiliza las instrucciones IN y OUT para transferir datos hacia los puertos de E/S y desde ellos. En la figura VI-1 se indican las instrucciones de transferencia de datos. La instrucción de salida que representa por el mnemónico OUT en los programas de lenguaje ensamblador, mientras que la instrucción de entrada utiliza el mnemónico IN. La longitud en bytes de la dirección del puerto puede seleccionar uno de 256 (2^8) puertos. La dirección del puerto viene de las ocho líneas de dirección menos significativas (A_0-A_7). La figura también muestra dos señales de control de salida adicionales al microprocesador. Cuando se utiliza la operación OUT, se usa una señal especial de escritura de E/S (I/O W). La operación IN también requiere el uso de una señal de salida especial llamada señal de lectura de E/S (I/O R). Ambas señales son activas en BAJO y se muestran al salir de los microprocesadores de la figura, se conoce como aislador o acumulador de E/S.

La transferencia de datos que utilizan las instrucciones IN y OUT se conocen como entradas/salidas controladas por el programa. Las instrucciones de programa controlan la transferencia de datos durante la operaciones de IN y OUT. Las transferencias de los datos pueden ser iniciadas por el dispositivo periférico diciendo: "estoy listo para enviar o recibir datos". Se emplean interrupciones en un dispositivo periférico para iniciar la acción en el microprocesador. Cuando este recibe una requisición de interrupción termina la instrucción que



Instrucción de Entrada/Salida

Figura VI-1

esté ejecutando y brinca a una rutina de servicio de interrupciones en la memoria de programa. Esta rutina puede incluir operaciones de entrada y salida.

Los principales componentes de una microcomputadora se comunican unos con otros a través de un conjunto de líneas compartidas conocido como bus del sistema (B). En un instante cualquiera sólo pueden estar conectados lógicamente dos dispositivos a B, uno que envía los datos y otro que los recibe.

Resulta particularmente importante la comunicación entre las siguientes parejas de unidades:

- 1.- La CPU y la memoria principal M.
- 2.- La CPU y los circuitos de interfaz para E/S.
- 3.- Los circuitos de interfaz para E/S y M.

A cada posición de la memoria M y a cada punto de acceso a los dispositivos de E/S se le asigna una dirección, que se utiliza para seleccionarlo como participante en las operaciones del bus. A la CPU no se le asigna dirección, pues se selecciona directamente mediante líneas de control específicas de B. En una microcomputadora sencilla todas las actividades del bus están iniciadas y controladas por la CPU. La mayoría de los microprocesadores se diseñan de manera que la CPU pueda ceder el control de B a determinados dispositivos de E/S, usualmente a través de circuitos de control de E/S especiales. Uno de tales circuitos es el controlador DMA (Acceso Directo a la Memoria), cuya finalidad principal es controlar la transferencia de datos entre los dispositivos de E/S como un DMA.

Una transferencia típica de datos a través del bus B del sistema en un microcomputadora supone la siguiente secuencia de acciones:

- 1.- Un dispositivo de control DEV1 tal como la CPU o un controlador DMA asociado con los dispositivos de E/S toma el control de B.
- 2.- El control DEV1 del bus coloca en las líneas apropiadas de B una palabra de dirección, que se utiliza para seleccionar un segundo dispositivo controlado DEV2 para participar en la transferencia de datos.
- 3.- Una vez que DEV2 ha sido activado, se inicia la transferencia de datos, cualquiera de ambos dispositivos puede actuar como fuente o como destino de los datos.

Usualmente se utilizan conjuntos separados de líneas de B, conocidos como buses de direcciones y de datos, para transmitir direcciones y datos, respectivamente. A veces las palabras de dirección y de dato se transmiten secuencialmente a través de líneas comunes de direcciones y datos, en cuyo caso se dice que las palabras de dirección y de datos están multiplexadas. Líneas de bus adicionales, conocidas colectivamente como líneas de control, se utilizan principalmente para iniciar y activar la transferencia de datos a través del bus.

Con frecuencia se utilizan para los buses líneas triestado, pues el tercer estado Z o de alta impedancia facilita enormemente el control de la transferencia a través del bus. Un dispositivo que esté conectado físicamente a una línea L, se puede desconectar lógicamente del bus llevándolo al estado Z. Cuando todas las conexiones de L están en estado Z, se dice que L está flotante. Un controlador de bus cede el control a otro controlador de bus haciendo que las líneas de dirección, datos y determinadas líneas de control floten, permitiendo por tanto que un dispositivo de control diferente tome el control.

Todas las comunicaciones a través del bus B deben temporizarse adecuadamente, de forma que los dispositivos de comunicación funcionen correctamente. Esto significa que los dispositivos de destino no tratarán de leer datos del bus hasta que todas las líneas de datos hayan alcanzado su valor correcto. De igual forma, los dispositivos fuente de datos no los eliminarán del bus hasta que hayan llegado a todos sus destinos. Ha de darse tiempo para permitir las fluctuaciones en los tiempos de transmisión de la señal, mediante un secuenciamiento adecuado de las señales del bus implicadas en la transferencia de datos. Por ejemplo, una palabra de datos D debe situarse en el bus un instante σ anterior a la activación de una línea de control S que indica la presencia de D. El retardo σ debe ser suficiente para garantizar que todos los bits de D lleguen a dispositivos de destino antes de que active la señal S; en otro caso el dispositivo de destino podría leer prematuramente la línea de datos y recibir valores de datos inadecuados. La temporización de las señales implicadas en una operación del bus pueden describirse mediante un cronograma como el de la figura VI-2, dibujado para que se muestre la secuencia según la cual cambian los valores de las señales, así como los retardos permisibles entre diferentes acciones.

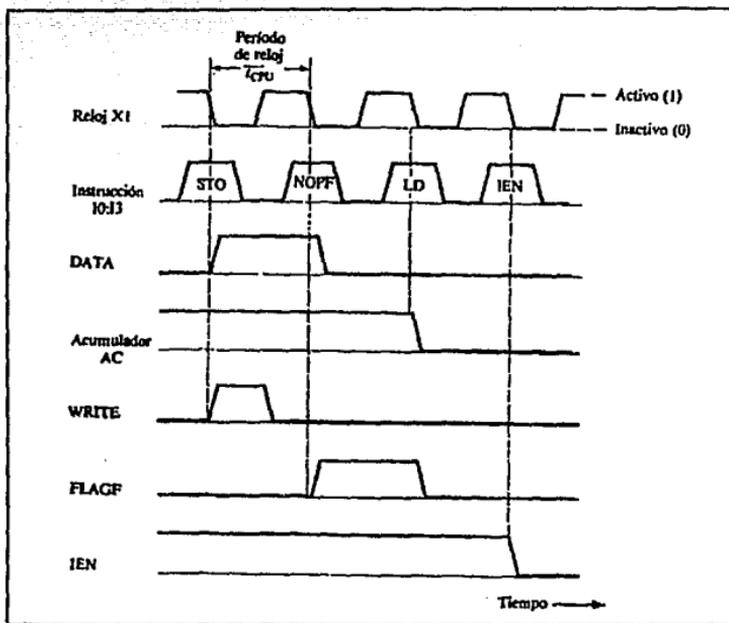


Figura VI-2

6.2 CONTROL DE DISPOSITIVOS CON TECNICAS DE PROGRAMACION

Una operación de entrada o salida es el acto de transferencia de datos desde o hacia un dispositivo periférico seleccionado. El microprocesador es el foco de todas esas operaciones; por lo tanto, una entrada significa que los datos fluyen a la MPU y una salida significa que salen de ella.

El microprocesador utiliza las instrucciones IN o OUT para transferir datos hacia los puertos de E/S y desde ellos. La instrucción de salida se representan por el mnemónico OUT en los programas de lenguaje ensamblador, mientras que la instrucción de entrada utiliza el mnemónico IN. Estas instrucciones utilizan su código de operación y un número de dispositivo o dirección del puerto. La longitud en bytes de la dirección del puerto puede seleccionar uno de 256 (2^8) puertos. La dirección del puerto viene de las ocho líneas de dirección menos significativas (A20-A27).

Cuando se utiliza la instrucción OUT, se utiliza una señal especial de escritura de E/S (I/O W). La operación IN también requiere el uso de una señal de salida especial llamada señal de lectura de entrada/salida (I/O R). Ambas señales son activadas en BAJO. A estas señales se les conoce como aislador o acumulador de entrada/salida.

Las transferencias de datos que utilizan las instrucciones IN y OUT se conocen como entradas/salidas controladas por el programa. Las instrucciones de programa controlan la transferencia de datos durante las operaciones de IN y OUT.

La entrada y salida controlada por el programa se presenta en dos técnicas: una es mediante las instrucciones IN y OUT y la otra trata las posiciones de entrada y salida como direcciones de memoria regulares. Esta técnica se conoce como entrada y salida de mapeo de memoria, con la cual se utilizan instrucciones de acceso regular a la memoria. Las señales de control usuales de escritura (WR) y de lectura (RD) sirven también para introducir o sacar datos. Cualquier instrucción de acceso a la memoria puede ser útil para introducir o sacar datos mediante la técnica de mapeo de memoria de E/S.

Esta técnica es probablemente la más común y puede utilizarse con cualquier microprocesador.

La técnica de aislamiento de E/S sólo puede aplicarse con microprocesadores que tienen instrucciones de IN y OUT por separado y salidas de control de escritura y lectura especiales de entrada/salida.

6.3 INTERFAZ RS-232 Y LAZO DE CORRIENTE 20 mA

Existen dos medios para transmitir datos: paralelo y serie.

Si el dato a transmitir está en octetos (8 bits), transmisión en paralelo, quiere decir que ocho cables separados deben llevar simultáneamente las ocho corrientes eléctricas representativas entre los dos puntos: Transmisor y Receptor. La ventaja de esta forma de transmisión de datos es que los ocho bits llegan a su destino a un mismo tiempo, la transmisión en paralelo de datos puede efectuarse a velocidades extremadamente altas. La desventaja es que para la transmisión en paralelo se necesitan, al menos, nueve cables: ocho para los bits de datos y uno para el circuito común. Usualmente, aun se requieren más cables para controlar el flujo de datos a través de la conexión. En forma global se pueden tener hasta 30 cables o más lo que implica un trabajo tedioso y caro. Otro problema asociado con la transmisión de datos en paralelo es para el manejo de los niveles de voltaje (estados 1 y 0). Cuando un nivel cambia de estado de uno a cero, o viceversa, lo hace muy rápidamente, en el orden de nanosegundos (una mil millonésima de segundo). Desafortunadamente, según el cable se va haciendo más largo sus propiedades eléctricas (capacidad e inductancia) restringen la rapidez con la que un bit puede cambiar entre cero y uno, y es posible la corrosión o pérdida de datos. Debido a esto, la velocidad inherente a la transmisión de datos en paralelo hace que dicha transmisión sea problemática a largas distancias.

Se necesita un método de transferencia de datos más resistente y menos caro. La alternativa para enviar todos los bits es enviarlos separados, uno después de otro. Al extremo receptor, se invierte el proceso y los bits individuales se reúnen en el octeto original. Con un único bit a transmitir de una vez, los datos pueden transferirse con un circuito eléctrico simple consistente de sólo dos cables. Este medio, conocido como transmisión en serie, reduce la complejidad y

gran parte del coste de la técnica en paralelo, además de que facilita la comunicación entre dos dispositivos a extremadas distancias utilizando la línea telefónica (vía modem).

La desventaja de la conexión en serie es que presenta una disminución en la eficiencia, al menos se tarda ocho veces más en transmitir los ocho bits individuales, uno tras otro, que en transmitirlos todos simultáneamente en paralelo. Afortunadamente, esta pérdida de velocidad no demuestra ser una limitación significativa en la mayoría de las aplicaciones ya que los dispositivos periféricos en serie típicos son, de alguna forma, lentos, al menos en comparación con la velocidad interna de los microprocesadores. Un dispositivo siempre consume tiempo en algún proceso mecánico: las impresoras están limitadas por la velocidad de sus cabezas de impresión; los modems por las restricciones en la frecuencia de las líneas telefónicas, y las unidades de disco por su lenta velocidad de rotación.

La conexión RS-232 es el medio principal mediante el cual se conecta un equipo auxiliar del microprocesador en serie. RS significa Recommended Standard (Modelo Recomendado).

En 1969, la EIA (Asociación de Industrias Electrónicas), los laboratorios Bell y los fabricantes de equipos de comunicaciones dieron a conocer la interfaz RS-232 desarrollada con un único propósito: Conexión entre un Equipo Terminal de Datos o DTE y un modem (Equipo de Comunicación de Datos o DCE) empleando un intercambio de datos binarios en serie.

En su forma más simple, donde sólo un sistema transmite y el otro recibe, la conexión RS-232 consta solamente de dos cables: uno para transmitir datos y el circuito común (referencia).

Generalmente esta interfaz utiliza, como característica mecánica de la conexión, al conector DB-25, el cual tiene forma de D, fabricado por la Organización Internacional de Estándares (ISO). Este conector presenta 25 localidades, una para cada cable. También existen los conectores DB-15 y DB-9, este último puede acoplarse bien en un sistema de E/S en serie de caso general, como el que se muestra en la figura VI-3, donde DTE es la fuente y DCE el destino.

Traducido a un lenguaje menos técnico puede quedar ejemplificado en el diálogo siguiente:

DTE -Estoy listo para enviarte algunos datos, ¿los aceptas?

DCE -Permíteme un momento. Ya estoy listo, envíamelos.

DTE -Mándame una señal para que me confirmes que estás listo y así te pueda mandar

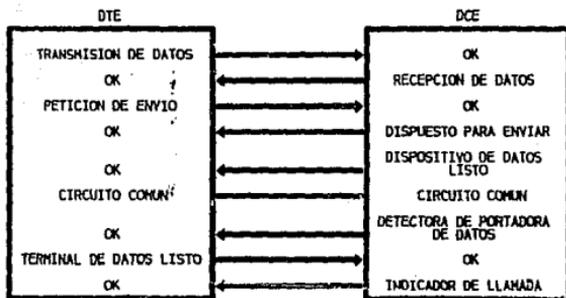


Figura VI-3

el primer dato.

DCE –Te mando dos señales; con una te digo que ya estoy listo en lo que respecta a mi hardware y con la otra te autorizo que me mandes el primer dato.

DCE –Quiero que protejas los datos que me vas a enviar con una portadora, mándame una señal cuando la generes.

DTE –¡Ya estoy listo!

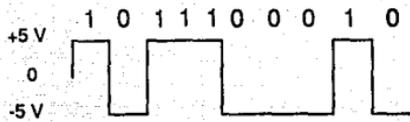
DCE –¡Permíteme un momento!, se me olvidaba decirte que habilitáramos el indicador de llamada para que detectemos alguna señal al otro lado de la red y podamos interrumpir nuestra comunicación en ese momento. Ya está. Mándame el primer dato.

Este diálogo permite una mayor comprensión para entender con más facilidad a los circuitos de E/S y en especial al componente en estudio. Como una observación, los dispositivos DTE y DCE deben siempre estar conectados con una línea común.

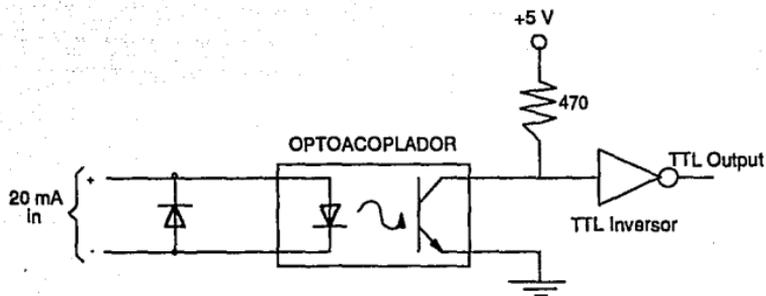
La interfaz RS-232C garantiza una distancia de 50 pies (15.244 metros), por lo que para distancias muy largas se debe emplear un lazo de corriente de 20 mA, para mantener los datos durante todo el trayecto. La capacidad máxima de transmisión de la RS-232C es sobre 20,000 bits por segundo. Para permitir más altos rangos en la transmisión de datos, la EIA ha designado a la interfaz RS-449 la cual permite rangos de 2 Megabits. En un futuro la EIA RS-449 puede reemplazar a la RS-232C.

Los sistemas digitales utilizan fuentes de voltaje para interfazarse con otros dispositivos. Cuando la distancia entre dispositivos es muy larga, el voltaje es reducido en el final de la recepción siendo ésta muy susceptible a fallar. Si se utiliza un lazo de corriente, las distancias entre dispositivos pueden pasar de 100 pies realizando la comunicación con muchas señales pequeñas. La terminal de la computadora tiene un circuito de no lazo de corriente, y el periférico tiene un circuito de lazo de corriente, el ASR-33, que trabaja en un rango de 20 mA a 60 mA de lazo de corriente, y estos niveles de corriente se convierten a niveles TTL mostrados en la figura VI-4a. En la figura VI-4b se muestra un optoacoplador integrado por un LED y un fototransistor. Cuando el fototransistor está saturado se genera un 1 con 20 mA de su fuente, y cuando está en corte, no hay corriente y se tiene un 0 lógico. El dispositivo TTL invierte el dato

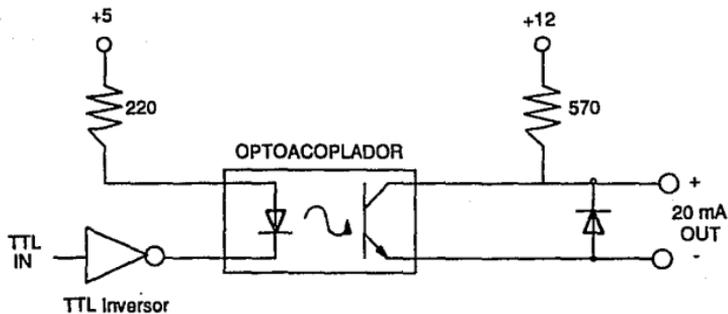
de la entrada y maneja el dato a su salida. Cuando se tiene la fuente de 20 mA enciende al optoacoplador y el inversor TTL invierte su entrada 1 a 0. En la figura VI-4c un 1 lógico a la entrada del dispositivo TTL, enciende al optoacoplador, utilizando los 20 mA de lazo para transmitir el dato.



(a)



(b)



(c)

Figura VI-4

CAPITULO VII

COMUNICACION SERIE ASINCRONA Y SINCRONA

INTRODUCCIÓN

La comunicación relaciona a dos o más ideas que finalmente se concretan en una conclusión. Cuando se necesita que una microcomputadora se comunique con dispositivos externos a grandes distancias, manejando una buena cantidad de datos, la única solución es utilizar dispositivos interfaces de característica serie con la gran ventaja de que puede emplearse la línea telefónica, y por lo tanto aprovechar equipos de actualidad como lo son modems y fax y así manejar la información en forma más completa.

La comunicación serie puede emplearse en forma síncrona o asíncrona según sea el caso. La comunicación serie entre dos dispositivos se lleva a cabo utilizando comandos denominados protocolos de comunicación.

7.1 ESTRUCTURA DE LA COMUNICACION ASINCRONA

La comunicación, ya sea síncrona o asíncrona tiene una finalidad: transferir e intercambiar datos entre ordenadores, (se entiende por ordenador una PC o terminal conectados a una red). Una red de ordenadores se interconecta a través de uno o varios caminos o medios de transmisión. La mayoría de las veces éste medio de transmisión es la línea telefónica, debido a su fácil accesibilidad.

Durante una operación de transmisión de datos, sólo pueden conectarse lógicamente dos unidades del bus del sistema. Una de las unidades tiene un control global sobre el bus del sistema, y frecuentemente se denomina administrador del bus o controlador del bus. Un controlador del bus puede enviar órdenes de transferencia de datos a través de las líneas de control del bus, y puede situar direcciones en el bus de direcciones. El otro dispositivo conectado lógicamente al bus tiene un papel más pasivo, y puede llamarse esclavo del bus. El dispositivo esclavo puede actuar como receptor de datos y decodificador de direcciones, y responde a órdenes del controlador del bus; no puede, sin embargo, enviar por sí mismo ni direcciones ni órdenes. Tanto el controlador como el esclavo pueden actuar como fuente de datos.

Cuando tanto los instantes de inicio y final de la operación del bus son impredecibles, se puede introducir una segunda señal de control cuyo nombre genérico es el de conformidad, para indicar la finalización. Por tanto, la operación está temporizada por dos señales de control preparado/demanda y conformidad. Este modo de comunicación se denomina totalmente asíncrono.

La transferencia asíncrona de datos es especialmente en buses de comunicación, ya que permite transferir eficientemente datos entre dispositivos con velocidades de comunicación muy diferentes. Está caracterizada por el uso de pares de señales de control de conformidad (petición/listo y reconocimiento), que se utilizan por cada unidad de comunicación para señalar la realización completa y satisfactoria de la transferencia de datos de una unidad a otra. Retrasando la activación o desactivación de una señal de conformidad, cada dispositivo puede hacer esperar al otro indefinidamente hasta que ha completado alguna acción que afecta a la

operación de transferencia. La figura VII-1 muestra las dos operaciones típicas de lectura y escritura asincrónicas. El controlador activa una orden de lectura o escritura, correspondiendo a una petición de dato o a una señal de dato listo, respectivamente, así señala al esclavo que la transferencia de datos indicada está teniendo lugar. El esclavo responde transfiriendo datos a o desde el bus de datos como se le ha ordenado, e indica que esta operación ha concluido activando la línea de reconocimiento. El controlador responde desactivando la señal de orden, que a continuación hace que la línea de reconocimiento se desactive.

En el caso de la operación de lectura de la figura anterior, la desactivación de la(s) línea(s) de órdenes indica al esclavo que el dato que este último le transmitió fue recibido satisfactoriamente por el controlador. En todo caso la fuente de datos es informada de la recepción de los datos transmitidos por la unidad de destino. Las relaciones causa-efecto entre las señales de conformidad se indican con flechas en la figura. Otro aspecto importante de la figura VII-1 temporización del bus es el hecho de que el controlador no deberá activar las líneas de órdenes leer/escribir hasta algún tiempo después de que haya situado la nueva dirección en el bus de direcciones. Esto se hace para asegurar que todos los bits de dirección que llegan al esclavo se han estabilizado en sus valores finales antes de decodificar la dirección. Similarmente, el controlador no debe eliminar la dirección en curso del bus de direcciones hasta después de que hayan desactivado las líneas de órdenes.

Las órdenes de transferencia de datos o líneas de control especifican la dirección de la transferencia de datos y son generalmente llamadas líneas de habilitación leer/escribir.

En el capítulo V se ejemplifican las instrucciones IN y OUT para comunicación de datos.

7.2 INTERFACES ASINCRONAS: ESTUDIO DEL UART Y SIMILARES

UART significa Universal Asynchronous Receiver/Transmitter (Receptor/Transmisor Asíncrono Universal). Virtualmente se lleva a cabo el proceso completo de E/S en serie mediante un único circuito integrado que es el UART. Antes de que se desarrollara este circuito, el proceso de E/S era controlado directamente por el mismo microprocesador del ordenador. Un programa era

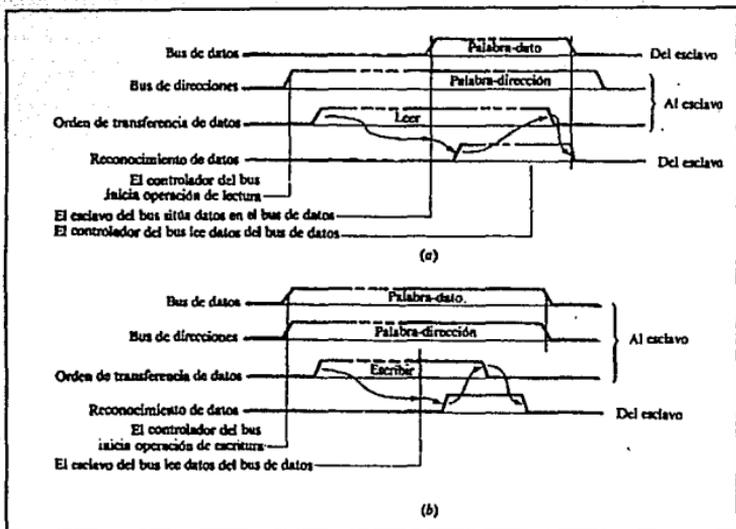


Figura VII-1

responsable incluso de los más mínimos detalles requeridos para convertir datos en paralelo en datos en serie, enviar los bits uno a uno a las correspondientes patillas de la conexión, y en el momento preciso y correcto. Debido a que tales programas, orientados a bits, gastan mucho tiempo del procesador y muchos recursos del ordenador, eran caros y tediosos de escribir. La E/S estaba reducida a un mínimo porque a los programadores no les gustaba escribirla y a los compradores no les gustaba pagar por esos programas.

La UART se conoce como un dispositivo de servicio debido a que releva al procesador del tedioso trabajo de las E/S en serie. En contraste con una E/S controlada por el procesador, un programa con UART no se ocupa de detalles. El programador trata a la UART como un buzón, dentro del cual se echan los caracteres de salida para ser despachados, o del que se sacan los caracteres de entrada.

Funcionalmente la UART reúne una sección de Transmisor y una sección de Receptor, en las cuales se llevan a cabo las siguientes funciones:

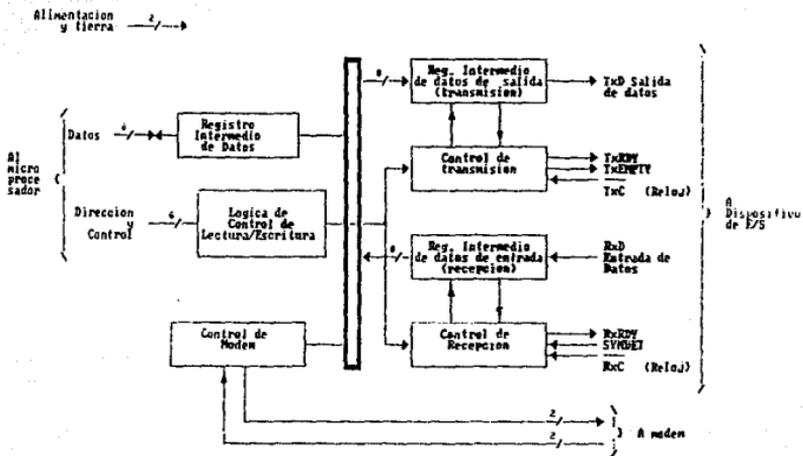
- Convierte los octetos (8 bits) que recibe el procesador en paralelo, en una corriente en serie de 8 bits, y viceversa, las corrientes en serie de 8 bits las convierte en octetos para entregarlas al procesador.

- Añade los necesarios bits de comienzo (start), parada (stop) y paridad a cada carácter que va a ser transmitido y los elimina de los caracteres que se reciben. Estos bits se utilizan para el control de errores en transmisión.

- Se asegura que los bits son enviados a la velocidad apropiada. Calcula el bit de la paridad de los caracteres recibidos e informa de los errores detectados.

Además hay una sección de Control y Estado, que, entre otras cosas, recibe el estado lógico de las patillas de acoplamiento, informando al procesador del estado de estas y, cuando un programa las llama, cambia el estado de las señales de acoplamiento de salida.

La figura VII-2 muestra estas tres secciones. La parte superior muestra la sección de Transmisor. Aquí, datos en paralelo provenientes de la derecha llegan al transmisor por el bus de datos. A la izquierda salen los bits de datos en serie. La parte de abajo del diagrama muestra la sección de Receptor. Los bits de entrada llegan al receptor desde la izquierda y son convertidos



Organización del circuito local RS232 (USART)

Figura VII-2

en datos en paralelo, y luego enviados por las líneas de datos de la derecha.

Es importante observar que tanto la sección del Receptor como la del Transmisor utilizan las mismas líneas de datos. Siempre que un carácter va a ser transmitido, la sección de Transmisor se conecta a las líneas de datos para tomar el carácter. Asimismo, cuando un carácter va a ser recibido y nuevamente convertido en un octeto, el Receptor se conecta a las líneas de datos mientras el octeto es transferido. Mediante comandos "de control" del software, la sección de Control/Estado automáticamente determina cuando las líneas de datos deben ser conectadas al transmisor o al receptor.

7.3 ESTRUCTURA DE LA COMUNICACION SERIE SINCRONA

Una vez establecido un camino de comunicación activo entre dos dispositivos, la transferencia de datos se temporiza mediante una o más señales de control. Una solución sencilla que requiere muy poco hardware es un reloj que funcione libremente como referencia temporal para todas las actividades del bus; ésta se conoce como comunicación síncrona. A todos los dispositivos conectados al bus del sistema se les puede conectar una señal de reloj sencilla, tal como el reloj de la CPU. Otra solución es que los dispositivos que se comunican tengan relojes independientes de la misma frecuencia, supuesto que los relojes se pueden mantener en sincronismo. En cada período de reloj la señal tiene un flanco de subida y otro de bajada, cada uno de los cuales se puede utilizar como punto de referencia para determinar en que instante debe subir o bajar la señal en otras líneas. La transmisión síncrona de los datos tiene la ventaja de que con una o dos señales de reloj bastan para temporizar todas las actividades del bus. El período de reloj está determinado por los retardos en los caminos de transmisión de la señal. Los dispositivos que se comunican han de tener velocidades de operación comparables; en otro caso el dispositivo más lento tiende a llevar al otro a un nivel de operación ineficiente y quizás inaceptable.

La comunicación síncrona requiere que los instantes de comienzo y de finalización de las operaciones del bus sean predecibles dentro de tolerancias razonables. Si sólo son predecibles los instantes iniciales, entonces se puede utilizar un modo de operación semisíncrono. En este modo

se utiliza una señal de control especial para iniciar la operación del bus, una transferencia de datos, por ejemplo. Una vez iniciada, la operación del bus se debe completar dentro de un período prefijado. Cuando la transferencia de datos semisíncrona está iniciada por el dispositivo de destino, la señal de control de inicio, denominada petición de datos. La señal de control de iniciación para una transferencia de datos iniciada por el dispositivo fuente tiene el nombre genérico de (datos) preparados o dispuestos (ready).

Para transmitir datos se debe fijar un control de velocidad, al que se le conoce como regulación de tiempo. Los datos transmitidos son sincronizados por medio de elementos de regulación de tiempo. Cuando se maneja la comunicación síncrona se tienen bits de inicio y parada, e indican el principio y final de la transmisión.

El factor principal para el establecimiento del límite de la velocidad es dado por la longitud de información a transmitirse.

El límite de velocidad es usualmente expresado en bits por segundo (bps), este valor es el número de 1's y 0's que pueden ser transmitidos durante el período de 1 segundo. Una vez que la velocidad es conocida, puede dividirse por el número de bits por carácter para determinar cuantos caracteres pueden ser transmitidos en un segundo. Las velocidades síncronas típicas son 1,200, 2,400, 9,600 y 19,200 bps. Si la velocidad de un medio de comunicación es de 2,400 bps, y la longitud de cada carácter es de 8 bits, aproximadamente 300 caracteres pueden ser transmitidos cada segundo. La norma de la RS-232 permite velocidades hasta de 20,000 bps.

En la mayor parte de comunicación de datos sobre líneas telefónicas se requieren velocidades superiores a los 1,200 bps, el modo síncrono de operación será entonces utilizado.

7.4 INTERFACES SINCRONAS: USART Y SIMILARES

Los correspondientes circuitos de interfaz de E/S serie (UART), son la ACIA 6850 (para el microprocesador 6800) y el circuito de interfaz de comunicaciones programable 8251.

El 8251 es llamado también USART (Universal Synchronous/Asynchronous Receiver/Transmitter, en español: Receptor/Transmisor Síncrono/Asíncrono Universal), porque

a diferencia del 6850, puede controlar transferencia de datos de E/S serie síncronas además de asíncronas. Una característica común de todos estos circuitos e E/S es que pueden programarse en línea (desde el programa) para satisfacer los requerimientos de interfaz de un amplio número de tipos diferentes de dispositivos de E/S.

En la figura VII-2 también se presenta la estructura de la USART 8251 de Intel.

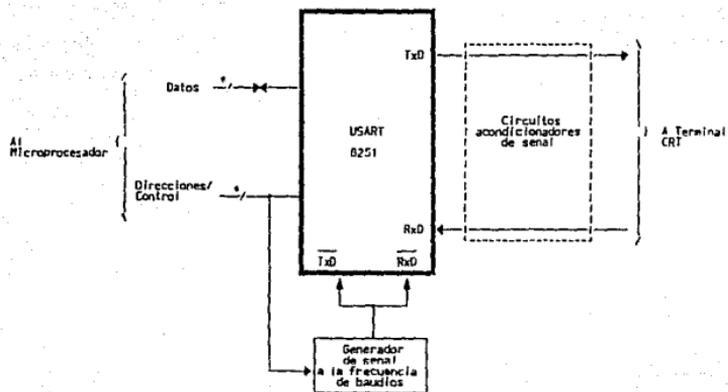
Este circuito tiene una interfaz serie que contiene una línea de datos TxD (Transmitir Datos), y una línea de entrada de datos RxD (Recibir Datos), y un conjunto de líneas de control asociadas. En una aplicación típica los caracteres alfabéticos codificados en binario se transfieren en serie entre el 8251 y un dispositivo de salida a través de las líneas TxD y RxD. El 8251 automáticamente efectúa una conversión de caracteres serie a paralelo durante las operaciones de salida. Se programa para administrar la transmisión de datos según las convenciones de muchos tipos de dispositivos serie de E/S y de buses normalizados de E/S serie como el RS-232.

El 8251 tiene dos modos de funcionamiento principales, denominados síncrono y asíncrono. La comunicación asíncrona es más frecuente con dispositivos de E/S de baja velocidad tales como monitores (CRT Tubo de Rayos Catódicos), e impresoras-teletipo. En este modo de funcionamiento el 8251 y el dispositivo de E/S que es conectado funcionan aproximadamente a la misma velocidad, pero son controlados por relojes independientes. Los bits de un carácter son transmitidos en serie a una velocidad predeterminada llamada número de baudios, en honor al inventor francés J. M. E. Baudot. Una velocidad de n baudios significa que se transmiten n señales distintas por segundos; n baudios corresponde a n bits/segundos, suponiendo que cada señal transmitida contiene un bit de información. Mientras que los bits de un carácter deben transmitirse a una velocidad específica, los caracteres individuales pueden estar separados por intervalos de tiempo arbitrarios. Esto se efectúa haciendo que delante y detrás de cada carácter vayan unos bits adicionales de sincronización denominados bits de comienzo y de parada, un proceso denominado enmarcado. Por ejemplo, si el estado de reposo de una línea serie de entrada de datos tal como RxD es 0, la llegada de un bit de comienzo 1 indica que está iniciándose la transmisión de un nuevo carácter. Al bit de comienzo le sigue una secuencia de 1 a 8 bits del carácter transmitido a los baudios prefijados. Finalmente, la transmisión de un bit de parada 0

permite al receptor volver al estado de reposo durante un período de tiempo indefinido. En consecuencia los requisitos de enmarcado de la comunicación de datos asíncrona implican una sobrecarga de al menos 2 bits de comienzo/parada por carácter transmitido. El 8251 puede programarse para insertar y eliminar automáticamente bits de comienzo/parada durante las operaciones de E/S asíncronas. La figura VII-3 muestra como el 8251 podría usarse como circuito de interfaz de E/S asíncrono a un terminal CRT. La transmisión de datos se efectúa con las líneas RxD y TxD; si fuese necesario deberían añadirse circuitos adaptadores de señal para acoplar los requisitos eléctricos del bus serie de la USART con el terminal CRT. Un circuito reloj denominado generador a la frecuencia de baudios sirve como referencia local de tiempos al 8251, y se conecta a las líneas de entrada de reloj de transmisión (TxC) y (RxC).

Cuando se usa para comunicación de datos síncrona, el 8251 recibe las señales de reloj directamente de los dispositivos de E/S a través de sus líneas TxC y RxC. Esta elimina la necesidad de enmarcar los caracteres, permitiendo obtener mayores velocidades en la transmisión de datos. Deben transmitirse de vez en cuando caracteres especiales de sincronización para permitir la identificación de las fronteras entre caracteres. En su modo síncrono, el 8251 puede programarse para detectar caracteres predefinidos de "sync" (de sincronismo) en el flujo de entrada e insertarlos en el flujo de salida.

El 8251 tiene una puerta de E/S correspondiente al registro de control (CR) a las que pueden enviarse órdenes de 8 bits desde la CPU. La CPU también puede captar un byte de estado procedente del 8251, que indica la disponibilidad del mismo para recibir o transmitir datos y cualquier situación de error detectada durante la transmisión de datos. El modo de funcionamiento del 8251 está determinado por un par de órdenes enviadas por la CPU. Una orden "modo" especifica el tipo de transferencia de datos (síncrona o asíncrona), los baudios, la longitud del carácter y los convenios de sincronización seguidos. Una segunda orden se usa para iniciar el 8251 y para comenzar las transferencias de E/S activando las líneas de control que van a los dispositivos de E/S. Varias líneas de control de E/S especiales se incluyen para facilitar la interfaz del 8251 a un modem.



Utilización del CI 8251 USART para interfaz con un terminal CRT

Figura VII-3

7.5 COMUNICACION SINCRONA COMPLEJA: PROTOCOLOS

El término protocolo se refiere a los procedimientos y a la lógica de manejo en el intercambio de datos de los equipos terminales de datos ETD. Los protocolos son acuerdos acerca de la forma en que se comunican entre sí los ETD y los dispositivos de comunicaciones, y pueden incluir regulaciones concretas que recomienden u obliguen a aplicar una técnica o convenio determinados. Por lo general, son varios los protocolos que cooperan para gestionar las comunicaciones. Existe, por ejemplo, un protocolo que se encarga de controlar el flujo de tráfico por cada canal de comunicación; suele existir un segundo protocolo que selecciona el mejor canal (entre varios) para que lo utilice el primer protocolo.

El primer protocolo entra dentro de la categoría de los protocolos de enlace o de línea (o control de enlace de datos). El segundo se conoce como protocolo de conmutación o encaminamiento.

Los ETD se comunican entre sí mediante las técnicas descritas en la figura VII-4. La mayoría de los protocolos que aparecen en la figura se conocen como protocolos de línea (enlace o canal) o controles de enlace de datos (DLC Data Link Control).

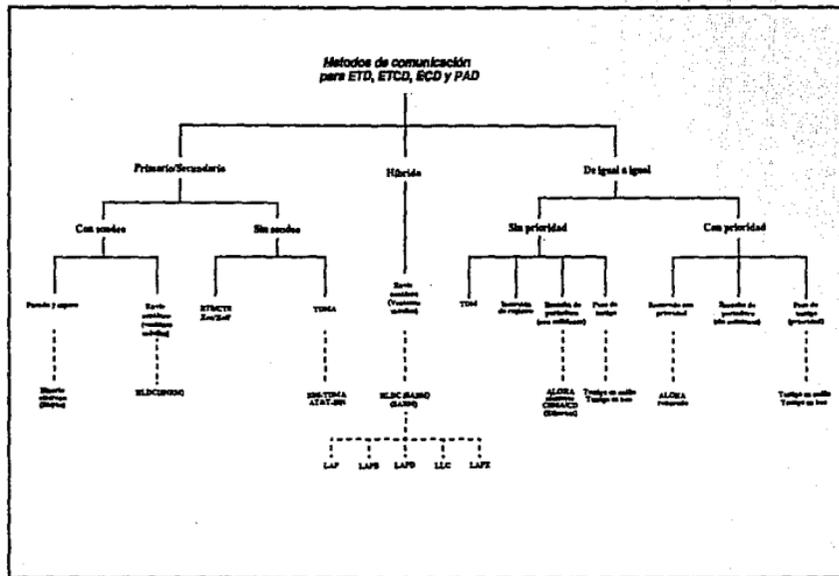
Reciben este nombre porque gobiernan el flujo de tráfico entre estaciones a través de un canal físico de comunicaciones.

Los protocolos de enlace de datos gestionan todo el tráfico de comunicaciones que atraviesa un canal. Así por ejemplo, si a un puerto de comunicación acceden varios usuarios, el DLC ha de garantizar que todos ellos puedan transportar sin errores sus datos por el canal hasta el nodo receptor. El DLC no suele tener en cuenta si los datos que transporta el canal proceden de múltiples usuarios.

Al gestionar un canal de comunicaciones, los protocolos de control de enlace de datos siguen varias etapas perfectamente ordenadas:

- 1) Establecimiento de enlace. Una vez que el ETD (Equipo de Terminación del Circuito de Datos, cuya misión es conectar los equipos ETD a la línea o canal de comunicaciones), ha conseguido una conexión física con el ETD remoto, el DLC "dialoga" con el DLC remoto para

Figura VII-4
148



asegurarse de que ambos sistemas están listos para intercambiar datos de usuario.

2) Transferencia de información. Ambas máquinas intercambian datos a través del enlace. El DLC comprueba todos los datos por si existe algún error en la transmisión, y envía validaciones de los mismos a la máquina que transmite.

3) Terminación de enlace. El DLC renuncia al control del enlace (canal), lo cual significa que no pueden transmitirse más datos hasta que se restablezca el enlace. Por lo general, un DLC mantiene activo el enlace siempre que la comunidad de usuarios desee enviar datos a través del mismo.

Un método muy utilizado para gestionar un canal de comunicaciones es el llamado protocolo primario/secundario (también maestro/esclavo). En esta técnica se designa un ETD, un ETCD o un ECD (Equipo de Conmutación de Datos, proporciona las funciones vitales de encaminamiento por la red, evitando los dispositivos y canales ocupados o fuera de servicio), como nodo principal del canal. Este nodo primario, por lo general un ordenador, controla las demás estaciones y determina si los dispositivos pueden comunicarse y, en caso afirmativo, cuando deben hacerlo.

El segundo método importante es el de igual a igual.

CAPITULO VIII

COMUNICACION PARALELO

INTRODUCCION

La comunicación en paralelo tiene como característica principal la alta velocidad de transmisión de datos, con la única limitante de que el medio de comunicación no sea demasiado larga. Este tipo de comunicación está controlado por una sincronización entre dos equipos, denominada saludo "Handshake". Este es un método de conocimiento de señales en el que se encuentran involucradas tres funciones distintas (talker, listener y controller).

Existen también interfaces en CI's que conectados con el microprocesador, forman una microcomputadora y son PLA, PIO, PPI y otros. Estos circuitos varían en cuanto a funciones, número de pines, temporización de señales, líneas de control, sentido de datos, etc. Se presentan algunos casos específicos de circuitos y su funcionamiento.

8.1 ESTRUCTURA DE LA COMUNICACION PARALELO

En la comunicación en paralelo, a diferencia de la comunicación serial, se presenta una alta velocidad de transferencia de datos, ya que 8 bits son lanzados simultáneamente, es decir al mismo tiempo, y que es muy usual en la comunicación de dispositivo a dispositivo por medio de buses dentro de una microcomputadora. También se pueden comunicar dispositivos periféricos no muy lejanos a la PC por medio de un cable paralelo, pudiendo ser todo en conjunto un equipo de escritorio.

Los buses paralelos transfieren todos los bits de información a través de líneas individuales en un mismo tiempo. Un bus paralelo incluye líneas de datos, direcciones y control. Un típico bus paralelo de 8 bits en un PC puede tener 8 líneas de datos, 16 líneas de dirección y de 5 a 12 líneas de control. Un microprocesador de 16 líneas de datos necesita de 16 a 24 líneas de dirección y muchas líneas de control.

Durante la transferencia en paralelo de datos, se presentan los siguientes pasos:

1.- Una dirección asignada a un periférico es puesta en el bus de direcciones de la PC, alertando al periférico para que acepte los datos por medio de una línea de control.

2.- Para una operación de salida (del CPU al periférico), los datos son puestos en el bus de datos y que por medio de una señal de habilitación de disponible son aceptados. Por medio de la interfaz, el periférico recibe y procesa los datos.

3.- Para una operación de entrada (periférico a CPU) se requiere una señal de petición de datos, para que la CPU los pueda transferir. Una vez que se ha llamado la atención de la CPU, se lleva a cabo el proceso de transmisión de datos descrito en los puntos anteriores.

8.2 EL SALUDO (HANDSHAKE): TIPOS

Esta señal significa un intercambio de indicativos y señales de control. Es una técnica de sincronización de comunicaciones entre dos modems o terminales de datos.

El saludo handshake es un método de conocimiento de señales patentado por Hewlett-Packard

junto con el bus IEEE-488-1975 paralelo. Este bus es un intento de conectividad de computadoras o calculadoras programables con instrumentos como voltímetros y generadores de señales. Algunas veces se denomina como GPIB o General Purpose Interfaz Bus (Interfaz de Propósito General).

Este bus puede conectarse hasta con 15 instrumentos conectados a un cable de más de 20 metros de largo, pero no más de 2 metros de cable por cada instrumento, es decir, no menos de 10 instrumentos.

Mientras que se considere esta norma, la información en bytes se comunica a velocidades altas hasta de 250,000 bytes por segundo.

Cada instrumento conectado al bus puede desarrollar las siguientes funciones:

- Un TALKER que es transmisor de datos a otros instrumentos.
- Un LISTENER que recibe datos provenientes de otros instrumentos.
- Un CONTROLLER que maneja a los TALKER y LISTENER.

Cada talker o listener pueden ser habilitados o deshabilitados por el controller en cualquier tiempo, lo cual define todas las actividades en el bus. Una computadora o una calculadora pueden servir como talker y listener.

El conector 488 tiene 16 líneas, agrupadas de la forma siguiente:

- 8 líneas de datos
- 5 líneas de control
- 3 líneas de handshake (saludo)

No tiene líneas de direcciones, mientras que el bus de datos pueda servir para transferir direcciones.

El saludo (handshake) es la clave para la comunicación en el bus. Antes de mirar a las tres funciones mencionadas anteriormente, el saludo considera un sencillo ejemplo utilizando solo dos líneas. La idea básica del saludo es simple. Un dispositivo pregunta: "¿Hola, cómo estas?", respondiendo el segundo: "Bien, y tengo algo para comunicarte", el primero contesta: "¡estoy listo!", y el segundo le dice: "OK, aquí está esto". Finalmente el primero responde "ya lo recibí".

En un bus el saludo hace dos líneas llamadas DAV (Data Valid) manejado por el talker, y

NDAC (Not Data Accepted) manejado por el listener; NDAC se activa con un bajo.

Suponiendo que el talker envía una serie de datos (bytes) a un listener. El saludo estudia una secuencia para que puedan entenderse, empezando en cualquier punto. Ocurren los siguientes pasos:

- 1.- Con $DAV=0$, el talker pone el byte D_0 en el bus de datos.
- 2.- Tan pronto como el talker ve que $NDAC=1$, hace que $DAV=1$ mostrándolo al listener como un dato válido. El talker debe mantener el dato en la línea hasta que sea $NDAC=0$.
- 3.- El listener acepta los datos y entonces hace a $NDAC=0$, manteniéndolo así hasta que ve $DAV=0$.
- 4.- El talker reconoce a $NDAC=0$, haciendo a $DAV=0$. Después se dispone para la próxima transferencia al colocar el dato D_1 en el bus de datos. El talker debe esperar que $NDAC=1$ antes de hacer a $DAV=1$.
- 5.- Cuando el listener ve que $DAV=0$, se encarga de hacer saber al talker que los datos han sido aceptados. De ahí que el listener haga al $NDAC=1$ en preparación para la siguiente transferencia.

Cuando todos los listeners conectados al bus reciben datos, cada uno hace su $NDAC=0$ para confirmar que recibieron datos. Las líneas NDAC individuales deben conectarse juntas para que el NDAC colectivo pueda transformarse a 0 sólo cuando un listener esté aceptando datos. El talker reconoce la señal colectiva NDAC y hace $DAV=0$, admitiendo que todos los listener han aceptado datos y muestra que los datos del bus han sido cambiados. Cada listener reconoce la señal $DAV=0$, poniéndose en $NDAC=1$.

La tercera línea del bus NRFD (Not Ready For Data), auxilia a coordinar las transferencias.

Como cada listener acepta datos, esto hace que $NRFD=1$ hasta que se establece $NDAC=0$. Como cada listener ve que $DAV=0$, esto genera que $NRFD=0$, mientras se hace $NDAC=1$. La señal colectiva NRFD se pondrá en 1 tan pronto como el primer listener reconozca a $DAV=1$. NRFD colectivo se pondrá en 0 cuando el último listener reconozca la línea $DAV=0$.

Ahora que se conoce el procedimiento del saludo, pueden estudiarse las líneas de control que usa el controlador, y que son las siguientes:

- * ATN (attention), distingue comandos o direcciones.
- * IFC (interfaz clear), fuerza a todos los instrumentos en un estado detenido.
- * REN (remote enable), selecciona si la operación es remota o es local.
- * SQR (service request), permite a un instrumento que solicite servicio con una interrupción.

* EOI (end or identify), nivela el último byte del mensaje del talker o, junto con ATN registran a los instrumentos con identificación. Cuando ATN=1, los primeros siete bits del bus de datos especifican comandos, el octavo puede ser usado para chequear la paridad. Por acuerdo general (aunque no convencional), se les asigna un cierto significado a estos comandos:

- * UNLISTEN, todos los instrumentos previamente establecidos como listener, son descartados de su función.

- * UNTALK, los talker son descartados de su función.

- * CLEAR, todos los instrumentos se llevan al estado cero.

El uso del bus puede volverse complicado. Puede ilustrarse con una secuencia de comandos, la cual el controlador actúa primero como un talker y después como un listener.

- 1.- El controlador inicializa la interfaz con IFC.
- 2.- El controlador inicializa todos los instrumentos, y con el saludo limpia los comandos.
- 3.- El controlador emite la dirección para un instrumento con ATN y el saludo lo convierte en talker.
- 4.- El controlador envía datos a el dispositivo con saludo.
- 5.- El controlador da el comando "unlisten" al dispositivo con saludo.
- 6.- El controlador emite direcciones al dispositivo con ATN y saluda a los que serán listener.
- 7.- El controlador pone al ATN en modo de datos y con la señal EOI finalmente recibe los datos.

8.3 INTERFACES INTEGRADOS: PIA, PIO, PPI Y SIMILARES

Las interconexiones entre los CI's que forman una microcomputadora están en gran medida

determinadas por las características del bus del sistema. Los circuitos de memoria y de interfaz de E/S tienen pines de datos y de direcciones que deben conectarse a las líneas correspondientes del bus del sistema.

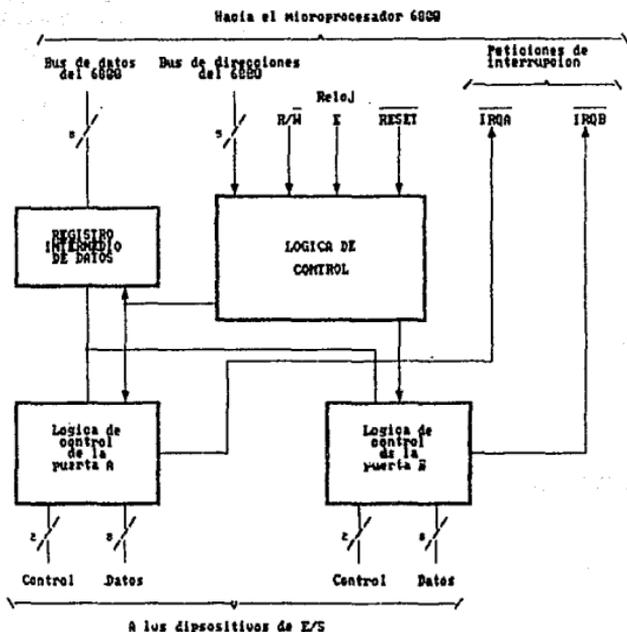
El número de señales que un dispositivo de E/S presenta a una microcomputadora y las características de temporización de estas señales, en otras palabras, la interfaz del dispositivo de E/S, pueden variar en un rango muy amplio. Los dispositivos de E/S serie y transmiten datos a razón cada vez, mientras que los dispositivos de E/S paralelo transmiten los datos a razón de una palabra cada vez (8 bits).

PIA

En la figura VIII-1, la PIA, (Adaptador Programable de Interfaz) 6821 de Motorola contiene dos compuertas independientes de E/S denominadas A y B. Cada puerta puede estar conectada a uno o varios dispositivos de E/S a través de diez líneas cuyo sentido (entrada o salida) se programan mediante una palabra de control del microprocesador a la PIA (en este caso del microprocesador 6800 de Motorola a la PIA 6821), donde se almacena en un registro de control. En una aplicación típica, 8 de las 10 líneas conectadas a una puerta del 6821 se pueden utilizar como líneas de datos para entrada o salida, mientras que las dos líneas restantes se utilizan como líneas de control (por ejemplo, como líneas de petición/conformidad). La PIA contiene registros de adaptación para el almacenamiento temporal de los datos que pasan a través de sus compuertas de E/S. También es capaz de enviar a la CPU señales de petición de interrupción.

PIO

El Circuito de I/O Paralelo (PIO), diseñado para el microprocesador Z-80, es un circuito de un conjunto de circuitos integrados fabricados para facilitar los interfaces de este procesador. El circuito PIO está diseñado para proporcionar dos compuertas, programables y compatibles a nivel TTL, para la transferencia de datos entre la CPU Z-80 y los dispositivos periféricos. Encapsulado



Interfaz de E/S en paralelo de la serie 6800 de Motorola, la PIA 6821

Figura VIII-1

en un chip de 40 patillas, el PIO proporciona las siguientes posibilidades necesitándose una muy pequeña circuitería externa:

- Dos compuertas independientes de 8 bits en paralelo, bidireccionales para el interfaz de periféricos, con control de la transferencia de datos del tipo "handshake".

- I/O mandadas por interrupción.

- Cuatro modos de funcionamiento seleccionables mediante software:

Modo 0 - Salida de Byte

Modo 1 - Entrada de Byte

Modo 2 - Byte bidireccional

- * Disponible sólo en la puerta A.

Modo 3 - Control de Bit

Cada modo de funcionamiento utiliza "handshaking", controlado por interrupción:

- Lógica de interrupción de prioridad encadenada en serie.

- Todas las entradas y salidas son completamente compatibles a nivel TTL.

La descripción de la pastilla PIO se muestra en la figura VIII-2.

D7-D0 - Bus de datos triestado bidireccional. D0 es el bit menos significativo.

B/A Sel - Selección de puerta B o puerta A. Un nivel bajo selecciona la puerta A, un nivel alto selecciona la puerta B.

C/D Sel - Control o selección de datos. Esta patilla define el tipo de transferencia de datos a ser realizado entre la CPU y el PIO. Un nivel alto durante una escritura de la CPU al PIO provoca que el bus de datos del Z-80 sea interpretado como un mando por la puerta seleccionada por la línea B/A. Un nivel bajo significa que el bus de datos está siendo utilizado para la transferencia de datos entre la CPU y el PIO.

CE - Chip Enable, habilitado con un bajo.

ϕ - Reloj del sistema (entrada).

M1 - Señal de ciclo de Máquina Uno de la CPU, (entrada, habilitada con bajo). Esta señal procedente de la CPU es utilizada como un pulso de sincronización para controlar varias operaciones internas del PIO.

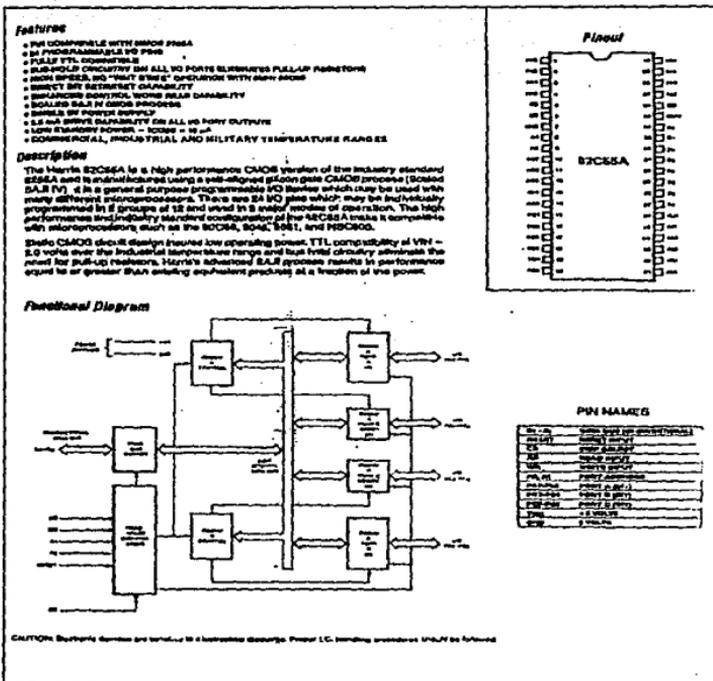


Figura VIII-2

IORQ - Petición de entrada/salida desde la CPU, activada con bajo. Se utiliza conjuntamente con las señales de selección B/A, C/D, CE y RD para transferir mandos y datos entre el Z-80 y el PIO.

RD - Estado de lectura desde la CPU, activado con bajo.

IEI - Habilitación de interruptores, activa con alto. Esta señal es utilizada para formar una cadena en serie de prioridades de interrupción cuando más de un dispositivo mandado por interrupciones está siendo utilizado.

IEO - Habilitación de interrupción de salida, activa en alto. Esta señal bloquea los dispositivos de menor prioridad para que no interrumpan a los dispositivos de mayor prioridad que están siendo servidos por la rutina de servicio de las interrupciones de la CPU.

INT - Petición de interrupción, activa en bajo.

A0-A7 - Bus de la puerta A, triestado, bidireccional. Este bus de 8 bits es utilizado para transferir datos y/o estados e información de control entre la puerta A del PIO y un dispositivo periférico.

A STB - Impulso de sincronización de la puerta A desde el dispositivo periférico, activada en bajo.

A RDY - Registro A preparado, salida activada en alto.

B0-B7 - Bus de la puerta B, bidireccional triestado.

B STB - Impulso de sincronización de la puerta B desde el dispositivo periférico. Entrada habilitada en bajo.

B RDY - Registro B preparado. Salida activada en alto.

PPI

La familia 8080/8085 contiene varios circuitos de interfaz de E/S programables. La pastilla 8255, Interfaz de Periféricos Programable (PPI), que se observa en la figura VIII-2, proporciona un conjunto de compuertas de E/S paralelas. Sirve para conectar dos o tres compuertas de E/S paralelo al bus del sistema del 8080/8085. De los 40 pines del 8255, hay 24 de E/S disponibles

para conectar dispositivos de E/S.

El sentido (entrada, salida o bidireccional) y la función (datos o control) de estos pines se determina durante la operación normal mediante una palabra de control de 8 bits que se transmite al 8255 desde la CPU. Se pueden especificar muchos modos de operación con esta palabra de control.

8.4 INTERFACES UNIVERSALES PROGRAMABLES: UPI Y SIMILARES

El circuito de entrada/salida universal 10696 está desarrollado por ROCKWELL para expandir la posibilidad de entradas/salidas de su microprocesador PPS. Según se puede observar en la figura VIII-3 está compuesto por varios bloques entre los que se destacan:

Zona de entradas, formada por 12 bits, agrupados en 3 grupos de 4, seleccionables.

Zona de amplificadores de salida formada asimismo por 12 bits en 3 grupos de 4.

Lógica de selección de los anteriores.

Decodificador de instrucciones.

Circuito lógico de selección de la unidad.

Circuito amplificador de reloj.

Para programar este circuito es preciso emitir 2 palabras de control desde la unidad central. La primera palabra es la señal de conexión de entrada/salida. La segunda palabra contiene la dirección del circuito periférico seleccionado y la instrucción que se le solicita. En esta segunda palabra, 4 bits indican la dirección del periférico, dirección que será comparada con las 4 entradas de preselección de que dispone, y los otros 4 bits especifican cuál de los 3 grupos de entrada o de los 3 grupos de salida va a ser activo.

Este circuito dispone de una decodificación propia de su dirección, de tal modo que sin necesidad de circuito externo pueden conectarse hasta 16 unidades distintas al mismo microprocesador, lo cual permite tener hasta 192 bits de entrada y otros 192 bits de salida.

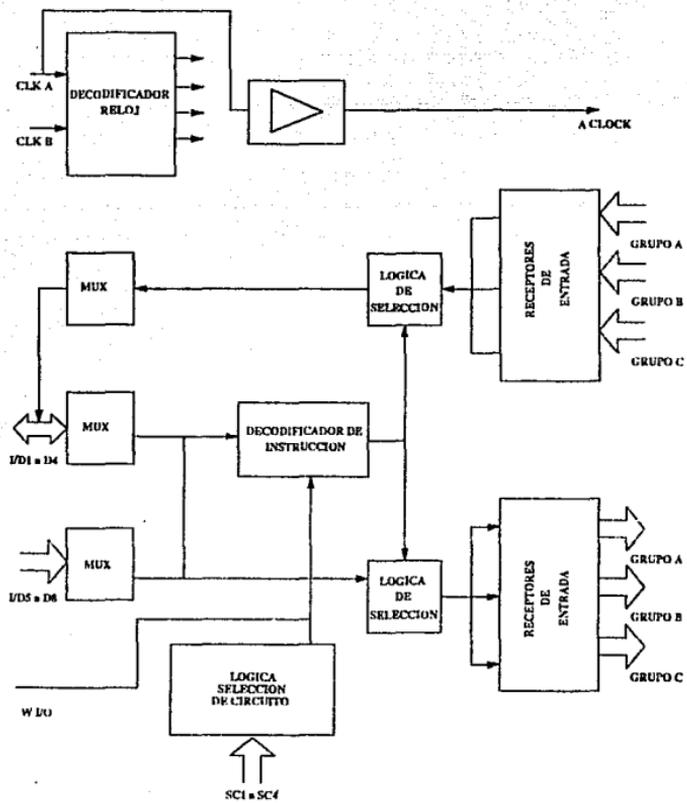


Figura VIII-3

CAPITULO IX

APLICACIONES DE LOS MICROPROCESADORES

INTRODUCCION

La necesidad que tiene el hombre para poder ejecutar trabajos con un alto grado de precisión, en tiempos relativamente pequeños y además en espacios muy reducidos, sólo la puede resolver un dispositivo: el microprocesador.

Adaptando el microprocesador a una buena arquitectura lógica programada, da como resultado un trabajo bien logrado y confiable.

En la actualidad, el microprocesador gobierna a casi todos los equipos eléctricos y mecánicos: relojes, motores, computadoras, cohetes, automóviles, satélites, equipos de medicina, etc, basándose únicamente en el concepto de "control".

9.1 APLICACION DE LOS MICROPROCESADORES EN INSTRUMENTACION Y CONTROL

Los microprocesadores han sido usados en contadores, equipo de prueba, osciloscopios, voltímetros digitales, puentes de capacitancias automáticos, analizadores de rayos X, analizadores de sangre, medidores de distancia, sintetizadores de frecuencia y sistemas de adquisición de datos.

Para instrumentos simples y conjuntos de pruebas, el microprocesador puede proveer programabilidad, manejo de los problemas de interfaz, entrada de datos simplificada, despliegue de datos y mensajes de alerta y de instrucción en forma conveniente y selección automática de parámetros. El microprocesador puede proveer también autopruueba y autocalibración, chequeo de consistencia de datos interno, comunicación con computadoras o con instrumentos controlados por ellas y promedio automático de lecturas. En general reemplaza circuitería estándar en aplicaciones simples. El microprocesador es más confiable, hace los instrumentos más flexibles y más fáciles de usar.

Cuando están involucrados sistemas de instrumentación y equipo de prueba grandes, el microprocesador puede reemplazar completa o parcialmente a una minicomputadora. El microprocesador es más barato, físicamente más pequeño, más fácil de proteger de condiciones ambientales severas, más confiable y consume menos energía que una minicomputadora.

COMUNICACIONES.

Los microprocesadores han sido usados en terminales, redes de minicomputadoras, unidades de conmutación de mensajes, repetidores, sistemas de almacenamiento y repetición, dispositivos de codificación, control de centrales telefónicas de baja capacidad, sistemas de comunicación portátiles y modems.

En área de comunicaciones, los microprocesadores pueden manejar rutinas de grandes computadoras, realizar conversiones o cálculos simples, detectar señales y ajustar referencias analógicas, hacer la interfaz de líneas que operan con diferentes velocidades o protocolos y

efectuar funciones de edición. Además ellos pueden, como en el área de prueba e instrumentación, proveer programabilidad, controlar despliegues (displays) y mensajes de operador, manejar teclados u otros dispositivos de entrada y permitir fácil familiarización e interconexión de equipo. Los microprocesadores pueden incrementar la seguridad y reducir los costos de las comunicaciones.

COMPUTADORAS.

Los microprocesadores son usados en las computadoras de pasatiempo y de negocios, en las CPU y controladores de E/S de microcomputadoras y en muchos periféricos de computadora como monitores, impresoras, discos magnéticos, etc.

El microprocesador es el centro de todas las operaciones cuando se trata de equipo de computo, ya que es él quien determina la función a realizar desde el primer momento en que se energiza.

El obedece toda la serie de pasos que se den en un programa (que puede ser de un lenguaje de alto nivel o de su propio lenguaje ensamblador), de tal forma que maneja todos los datos dándoles la ruta que requiere, es decir, que utiliza los buses, de datos y direcciones, involucrando a todo el sistema que lo rodea consistente de memorias, unidad aritmético-lógica, unidad de control, y unidad de entrada/salida la cual tiene gran importancia porque de ésta el microprocesador tiene comunicación con el mundo externo conformado por todos los periféricos existentes.

INDUSTRIA.

Los microprocesadores han sido usados en sistemas de monitoreo, cámaras inteligentes para control de calidad, sistemas de pesaje y de reparación en lotes automáticos, control de ensamblado de máquinas, controladores de máquinas, herramientas, estirado de metal, prensas dobladoras, medidores de flujo de gas, sistemas de gráficas y dibujo, terminales industriales y probadores automáticos.

La mayor ventaja de los microprocesadores en la industria son su bajo costo, alta confiabilidad y tolerancia a condiciones ambientales difíciles. Ellos pueden, por lo tanto, ser usados en aplicaciones donde las microcomputadoras podrían ser demasiado costosas o poco confiables. En la mayoría de las aplicaciones industriales, la limitación de la velocidad y el potencial de cálculo de éstos no son desventajas importantes. Los microprocesadores pueden también ser usados en sistemas distribuidos donde realizan tareas locales y preparan información para una computadora central; la inteligencia local permite un alto grado de tolerancia en fallas en las condiciones ambientales de una fábrica.

EQUIPO DE NEGOCIOS.

Los microprocesadores han sido usados en pequeñas computadoras de negocios, cajas registradoras, terminales de colección de datos, calculadoras programables, procesadores de cheques, cajeros automáticos, sistemas de cobros y terminales de procesamiento de palabras.

La mayor ventaja de los microprocesadores en equipo de negocios es su capacidad para proveer características interactivas a bajo costo. El microprocesador gobierna la mayoría de las tareas requeridas por archivadores, vendedores, cajeros y secretarías; puede hacer cálculos simples, trámites de registros y tener acceso a información de manera rápida y segura. Por otra parte, puede proveer instrucciones, indicaciones de alerta y mensajes de error para el operador. Así el equipo basado en microprocesador puede reducir errores, ahorrar costos de entrenamiento y simplificar cambios en procedimientos financieros, porcentaje de impuestos o políticas de venta.

TRANSPORTACION.

Los microprocesadores se usan bastante en semáforos y en sistemas móviles de comunicación y terminales. Han sido colocados en equipo de prueba para encendido electrónico y sistemas de inyección de combustible. Su aplicación en la industria automotriz incluye: controladores de encendido, sistemas de frenado antiderrapante, sistemas de diagnóstico de tablero, controladores

de temperatura interior y sistemas de control, y de despliegue digital.

CONSUMIDOR/COMERCIAL.

Los usos de microprocesadores en el área del consumidor incluyen juegos, calculadoras, computadoras domésticas, expendios de refrescos y dulces, equipos de estéreo, cajeros automáticos, etc.

La gran variedad de aplicaciones de los microprocesadores, demuestra las ventajas ofrecidas por estos dispositivos que cada día mejoran sus características. Esto hace que los sistemas basados en ellos sean cada vez más flexibles, más confiables, menos costosos, más precisos y más fáciles de usar.

CONCLUSIONES

En el trabajo de tesis denominado "microprocesadores" se conoció el funcionamiento de éstos, así como sus aplicaciones.

En su funcionamiento, se habló de las propiedades que tiene el microprocesador para manejar todas las acciones dentro de un sistema mínimo, respetando una secuencia lógica de captar-decodificar-ejecutar una instrucción, propia de la versión a la que pertenece; esto es, que cada microprocesador presenta distintas instrucciones de acuerdo a su versión y marca. También se dió el grupo completo de instrucciones del microprocesador 8080 de Intel para ejemplificar a la programación.

Se dijo que un sistema mínimo o arquitectura está compuesta de elementos básicos como son las memorias, registros, buses, interfaces y dispositivos digitales de complemento. Las memorias manejan datos y definen direcciones. Los registros son elementos de almacenamiento y retención de la información binaria. Los buses son los caminos por donde se envían las señales, donde viajan los datos y por los que se accesan direcciones. Los circuitos interfaces son dispositivos de comunicación con elementos externos al sistema mínimo mencionado.

También se dieron a conocer todas las partes internas que constituyen al microprocesador, como son la Unidad Aritmético Lógica (ALU), el Acumulador, el Contador de programa, el Decodificador de Instrucciones, los Registros internos, la Unidad de Tiempo y Control, etc., así como el funcionamiento de todos estos dentro del microprocesador en base a diagramas de estados y secuencias Fetch.

Se especificaron tipos de memorias. Se dijo que las ROM o de Sólo Lectura no pueden ser

grabadas más de una vez además de que son las que contienen el programa de inicio y reconocimiento de un sistema al momento de energizarlo, que las memorias RAM o de Acceso Aleatorio pueden ser grabadas más de una vez, pero que en el momento de quitarles su polarización pierden toda la información contenida. Se habló de los tipos de interfaz existentes: Serie y Paralelo. La interfaz serie RS232C comunica a dos o más dispositivos a grandes distancias auxiliándose de un lazo de corriente para mantener latentes los datos durante su trayectoria, los datos son enviados bit por bit. La interfaz paralela o Centronics conecta a dos o más dispositivos en pequeñas distancias enviando ocho bits en un mismo tiempo. Se diferencié la comunicación Serie en Síncrona y Asíncrona mencionándose modelos en cada caso.

En lo que respecta a las aplicaciones del microprocesador, se comentó que actualmente gran cantidad de equipos y sistemas en todas las áreas de trabajo están gobernados por éste, desde equipos caseros, automóviles, equipos de oficina y escuela, hasta equipos de carácter científico.

Todo el contenido de este trabajo se apoya en una buena cantidad de figuras representativas de los temas y tablas de complemento, para hacer entendibles los conceptos teóricos y poder generar finalmente aplicaciones prácticas de ideas teóricas.

BIBLIOGRAFIA

Hayes, John P.

DISEÑO DE SISTEMAS DIGITALES Y MICROPROCESADORES.

Mc Graw Hill, México, 1990.

Tokheim, Roger L.

FUNDAMENTOS DE LOS MICROPROCESADORES.

Mc. Graw Hill, México 1989.

Gault, James W.

SISTEMAS DIGITALES BASADOS EN MICROPROCESADORES.

Mc. Graw Hill, México 1983.

Fisher, Michael.

PROGRAMACION EN LENGUAJE ENSAMBLADOR.

Mc Graw Hill, México 1989.

Black, Ulyess.

REDES DE COMPUTADORAS.

Macrobít, México 1990.

Taub, Herbert.

CIRCUITOS DIGITALES Y MICROPROCESADORES.

Mc Graw Hill, México 1983.

Wiatrowsky, Claude A.

LOGIC CIRCUITS AND MICROCOMPUTERS SYSTEM.

Mc Graw Hill, EEUU 1980.

Morris, Mano.

DISEÑO DIGITAL.

Prentice Hall, México 1982.

Ramirez, Edward V.

MICROPROCESSING FUNDAMENTALS.

Mc. Graw Hill, EEUU 1980.

Peatman, John.

MICROCOMPUTERS BASE DESIGN.

Mc. Graw Hill, EEUU 1982.