



Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

SIMULADOR DE MICROCOMPUTADORA

MC68705 - P3

TESIS PROFESIONAL

Que para obtener el Título de
INGENIERO MECANICO ELECTRICISTA

P r e s e n t a

L U W E I



México, D. F.

1984



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

I N D I C E

PAG.

CAPITULO I

MICROPROCESADORES Y MICROCOMPUTADORAS

1.1	DESARROLLO DE MICROPROCESADORES Y MICROCOMPUTADORA <u>S</u>	1
1.2	HARDWARE	3
1.2.1	UNIDAD DE CONTROL	5
1.2.2	MEMORIA DE CONTROL	7
1.2.3	REGISTROS DE TRABAJO	8
1.2.4	UNIDAD ARITMETICA Y LOGICA	9
1.2.5	UNIDAD DE CONTROL DE CANALES	10
1.2.6	MEMORIA INTERNA	10
1.2.7	RELOJ	11
1.2.8	MEMORIA	11
1.2.9	DISPOSITIVOS DE ENTRADA/SALIDA E INTERFACES	15
1.2.10	CANAL EXTERNO	18
1.3	SOFTWARE	20
1.3.1	TIPO DE INSTRUCCION	21
1.3.2	MODOS DE DIRECCIONAMIENTO	23

CAPITULO II

MICROCOMPUTADORA MC68705 - P3

2.1	DESCRIPCION GENERAL	26
-----	---------------------	----

	PAG.
2.2 HARDWARE.	26
2.2.1 MEMORIA.	28
2.2.2 UNIDAD CENTRAL DE PROCESAMIENTO.	28
2.2.3 REGISTROS.	29
2.2.4 CONTADOR DE EVENTOS.	31
2.2.5 INTERRUPCION.	35
2.2.6 ENTRADAS/SALIDAS.	36
2.3 SOFTWARE.	37
2.3.1 TIPO DE INSTRUCCION	37
2.3.2 MODOS DE DIRECCIONAMIENTO.	38

CAPITULO III

SIMULADOR DE MICROCOMPUTADORA MC68705-P3

3.1 INTRODUCCION AL SIMULADOR SIM05.	41
3.2 DESCRIPCION DE COMANDOS DEL SIM05.	42
3.2.1 COMANDOS.	42
3.2.2 SUBCOMANDOS.	46
3.3 SIMULACION DE INSTRUCCIONES.	46
3.4 SIMULACION DE PUERTOS.	47
3.5 SIMULACION DE INTERRUPCIONES.	47
3.6 GUIA DEL USO DEL SIM05.	48
3.7 CONCLUSION.	53

GRAFICAS

APENDICE I

APENDICE II

BIBLIOGRAFIA

C A P I T U L O I

MICROPROCESADORES Y MICROCOMPUTADORAS

1.1. DESARROLLO DE MICROPROCESADORES Y MICROCOMPUTADORAS.

El microprocesador es uno de los más excitantes desarrollos tecnológicos desde la aparición del transistor en 1948. Se ha predicho que este dispositivo no solamente revolucionaría el campo de la electrónica digital, sino también tendría una gran influencia sobre el modo de la vida de las generaciones presentes y futuras. El primer microprocesador, el INTEL 4004, fué fabricado en 1971, se desarrolló principalmente para aplicación de calculadoras. Es un circuito integrado monolítico (IC), en el cual se empleó la integración a gran escala (LSI) con tecnología de Metal-Oxido Semiconductor (MOS). Pronto, el 4004 fué seguido por una variedad de microprocesadores que se produjeron por muchos otros fabricantes de semiconductores. La mayoría de los microprocesadores usa tecnología LSI empleando MOS tanto de tipo P (PMOS) como de tipo N (NMOS), MOS complementario (CMOS), NMOS con alta densidad (HMOS) o procesos bipolares.

La introducción del microprocesador ha causado un cam

bio dramático en el diseño de sistemas digitales. Con el método tradicional, llamado lógica alambrada, se diseñan sistemas usando bloques lógicos individuales, tales como compuertas, multiplexores, codificadores, decodificadores, FLIP-FLOP's, registros, registros de corrimiento, contadores etc. Cada aplicación requiere un diseño único y existe una similitud muy pequeña entre diferentes sistemas. Una vez que el circuito está construido, su función es muy difícil de cambiar.

En cambio, el microprocesador forma un sistema de control de propósito general, el cual puede utilizarse para una amplia variedad de aplicaciones con pequeñas modificaciones al sistema. Además, en sistemas complejos, un microprocesador es más barato en general, que circuitos de lógica alambrada.

Las microcomputadoras, las cuales constan principalmente de un microprocesador, memoria y dispositivos de Entrada /Salida (I/O), están siendo utilizadas en tareas las cuales antes se realizaban por minicomputadoras. Muchas nuevas aplicaciones han aparecido donde las minicomputadoras no están económicamente justificadas. Las limitaciones de funcionamiento tales como velocidad, capacidad etc. limitan la competencia de microcomputadoras de la primera y segunda generación en áreas donde las minicomputadoras se están empleando. Sin embargo, estas limitaciones están disminuyendo a medida que se desarrollan tecnologías más sofisticadas.

1.2. HARDWARE

Existen dos aspectos diferentes de un sistema de microcomputadora: los componentes efectivos llamados "hardware" y los programas llamados "software".

Una microcomputadora está construida por elementos -- eléctricos, mecánicos y magnéticos que se consideran como componentes efectivos. Ella manipula números, o bien, información binaria bajo una secuencia de pasos organizados, la cual se denomina programa. Cada paso en la secuencia se llama instrucción. Las microcomputadoras, como todas las computadoras, son máquinas que poseen las siguientes características:

1. Un medio de entrada, por el cual los datos e instrucciones se pueden entrar.

2. Una memoria, de la cual se pueden obtener los datos e instrucciones, y donde se pueden almacenar los resultados en un orden deseado.

3. Una sección de cálculo, capaz de realizar operaciones aritméticas y lógicas sobre los datos de la memoria.

4. Un mecanismo de decisión, por medio del cual se pueden seleccionar cursos alternativos de acción en base a --

los resultados de cómputo.

5. Un medio de salida, por el cual se pueden entregar los resultados al usuario.

Las máquinas que satisfacen estas cinco propiedades se conocen como computadoras de HARVARD. Si, además de estas propiedades, las instrucciones se almacenan en la misma forma que los datos en la memoria, y cada uno es igualmente accesible a la sección de cálculo, entonces, las instrucciones se tratan como los datos, y la máquina puede modificar sus instrucciones. Tal máquina se conoce como computadora de VON NEUMAN o de PRINCETON.

La figura 1-1 muestra un diagrama de distribución general de una microcomputadora con las características más comunes. Se entiende que una microcomputadora particular puede no tener todas estas características. El grupo de bloques en el lado derecho se considera generalmente como unidad central de procesamiento (CPU). La CPU puede colocarse en un IC simple o puede dividirse en varios IC's. Varios componentes de la CPU están conectados juntos por líneas llamadas canales internos. Las líneas que conectan la CPU al resto de la computadora se llaman canales externos. El conjunto de IC's que contiene la unidad de control, memoria de control, unidad aritmética y lógica (ALU), y registros de trabajo se denomina como-

sección de procesamiento de la microcomputadora. La memoria principal o externa cuyo propósito es almacenar las instrucciones y datos, está mostrada en dos bloques: uno es la memoria de tipo ROM, y otro es la de tipo RAM, las cuales son dos tipos principales de memoria que se pueden incluir en una microcomputadora.

1.2.1. Unidad de control

El propósito de la unidad de control es traer la instrucción de la memoria principal, interpretarla y luego controlar su ejecución. Dependiendo del diseño, la unidad de control contiene alguna combinación de los siguientes registros:

1. Contador de programa

El contador de programa (PC) retiene la dirección de la localidad de la memoria principal, desde cual la siguiente instrucción será traída. El conjunto de instrucciones (programa) está normalmente guardado secuencialmente en la memoria principal, y la microcomputadora le da acceso secuencialmente también. Por esta razón, el PC se incrementa durante la ejecución de una instrucción. Sin embargo, en muchas ocasiones, es deseable cambiar la secuencia de las instrucciones, dependiendo del dato recibido y el resultado de la operación previa. Esto se puede llevar a cabo por ramificación de la localidad de memoria deseada. Tal paso requiere que el PC esta-

blezca directamente la localidad deseada en lugar del incremento.

2. Registro de instrucción

El registro de instrucción (IR) recibe la instrucción traída de la memoria externa, y la retiene mientras se está decodificando y ejecutando la instrucción.

3. Registro de condición

El registro de condición (CR) no es necesariamente un registro, puede ser una colección de FLIP-FLOP's individuales llamados banderas, o códigos de condición (CC). Estas banderas se usan para indicar el estado presente de la CPU y las características importantes de los resultados de la instrucción previa. El número y propósito de estas banderas varía de una computadora a la otra, pero típicamente se usan para indicar resultado cero, resultado negativo, acarreo, paridad etc.

4. Apuntador de pila

La mayoría de las microcomputadoras utiliza un mecanismo denominado pila. Una pila consiste en un conjunto de registros internos como una forma de memoria interna, o una porción de la memoria principal. La pila se usa para almacenar temporalmente la información importante cuando las subru-

tina o rutinas de servicio se ejecutan. La mayoría de las pila opera de forma que el último dato que entra es el primero que sale (LIFO). Por eso, el apuntador de pila (SP) retiene la dirección de la cima de la pila.

Después de que la ejecución de una instrucción se complete, la dirección de la siguiente instrucción se obtiene -- del PC. La siguiente instrucción se trae de la memoria y se coloca en el IR. La unidad de control entonces decodifica la instrucción y dirige su ejecución. Al mismo tiempo, la unidad de control establece en el PC la dirección de la siguiente instrucción. Esta dirección está determinada por el tipo de instrucción que está ejecutando y por los contenidos presentes del CR. Una vez que la ejecución de la instrucción -- presente está completa, la unidad de control regresa al PC para obtener la dirección de la siguiente instrucción, y tal ciclo se repite.

1.2.2 Memoria de control.

Si una CPU incluye una memoria de control, entonces -- la memoria de control se usa para retener un conjunto de instrucciones llamado microcódigos. Algunas microcomputadoras -- están construidas de modo que el usuario puede programarlas -- en dos niveles diferentes. El primer nivel consiste en ins-- trucciones que son capaces de realizar solamente las operaciou

nes más elementales dentro de la computadora, y se les denomina microinstrucciones de la computadora. Las microinstrucciones están agrupadas juntas para formar las operaciones más útiles, las cuales se denominan normalmente como macroinstrucciones, o simplemente instrucciones. Si la arquitectura de la CPU está basada en una memoria de control, las macroinstrucciones en el IR dirigen la unidad de control a una secuencia de microcódigos que corresponden a esa instrucción, y la unidad de control controla la ejecución de estas microinstrucciones. Si la memoria de control está en un IC separado, es posible que el usuario desarrolle su propio conjunto de microinstrucciones por programación de esta memoria. Tal programación se llama microprogramación.

1.2.3 Registros de trabajo.

Todas las computadoras contienen un cierto número de registros de trabajo, los cuales caen en una o ambas de las siguientes categorías:

1. Acumulador

Registro que sirve como un borrador para operaciones aritméticas y lógicas, El acumulador sirve también como una entrada de la ALU. Frecuentemente en las microcomputadoras, todos los datos de entrada y salida tienen que pasar por el -

acumulador. Así, éste es un registro clave. Algunas micro-computadoras tienen dos o más acumuladores, esto hace a la microcomputadora más flexible y permite que ciertas tareas sean complementadas más eficientemente.

2. Registro de dirección.

Registro que ayuda el direccionamiento de los datos e instrucciones en la memoria principal.

Un registro que puede realizar ambas funciones con igual facilidad se llama registro de propósito general. Generalmente, los contenidos de los registros de trabajo se necesitan almacenar temporalmente en la memoria principal o en la pila, cuando la CPU está sirviendo a otros dispositivos, subrutinas e interrupciones.

1.2.4 Unidad aritmética y lógica.

La unidad aritmética y lógica (ALU) realiza operaciones aritméticas y lógicas sobre los contenidos de los registros de trabajo, el PC, el SP, y/o la memoria principal, y guarda los resultados en el lugar indicado por la instrucción, generalmente en un acumulador. Ella también enciende y apaga las banderas apropiadas durante la ejecución de un programa.

Algunos diseños dependen de lo que se llama rebanada de bits. En una CPU de este tipo, la ALU y los registros de trabajo están contruidos por varias rebanadas de bits que están en paralelo, cada rebanada está incluida en un IC, el cual contiene unos registros que almacenan grupos de bits, generalmente dos o cuatro bits, y tiene la lógica necesaria para realizar varias operaciones aritméticas y lógicas sobre ellos. Diversos IC's puestos en paralelo forman lo que se llama registro de ALU (RALU). El concepto de rebanada de bits le da al usuario la oportunidad de diseñar parcialmente la CPU, se puede seleccionar el número de bits que se almacenan en los registros de trabajo y se manipulan con la ALU.

1.2.5 Unidad de control de canal.

La unidad de control de canales se usa para coordinar las actividades de la CPU con aquellos dispositivos del mundo exterior. En las microcomputadoras, algunas de las funciones de control, tales como interrupciones externas, pueden estar controladas por IC's separados.

1.2.6 Memoria interna

Pocas microcomputadoras tienen memorias internas en la CPU, las cuales consisten en un conjunto de registros que incluyen una pila interna. La diferencia conceptual entre re-

gistros de propósito general y memoria interna depende de la forma de direccionamiento. La pila permite guardar temporalmente los datos o direcciones. Cuando una información binaria entra a la pila, todos los datos previamente guardados en la pila se mueven una localidad hacia abajo. Esto se conoce como una operación PUSH. Cuando una información se retira de la pila, todos los datos se mueven una localidad hacia arriba. Esto se conoce como una operación POP. Un parámetro importante de una pila es el número de registros incluidos en la pila. Si uno intenta poner más información binaria en la pila que los registros existentes, la primera información se pierde.

1.2.7 Reloj.

Un reloj es un generador de pulsos. Estos pulsos son utilizados para coordinar las actividades de la computadora. Si el diseño es tal que el reloj está separado del IC del microprocesador, entonces los usuarios pueden elegir un reloj con la frecuencia deseada. La velocidad con la cual la computadora trabaja depende de esta frecuencia y todos los microprocesadores tienen una frecuencia máxima que establezca un límite superior de su velocidad.

1.2.8 Memoria.

La memoria principal (simplemente memoria) se usa para

almacenar las macroinstrucciones y datos. La memoria consiste en un gran número de bits, los cuales están organizados en grupos llamados palabras o bytes. A cada palabra de la memoria se le asocia una dirección, con la cual la CPU puede tener el acceso de las palabras de la memoria. Una dirección es una combinación única de bits que es transmitida a la memoria por vía de líneas de dirección de un canal externo. El número de bits en una palabra varía de una computadora a la otra y es el resultado de las consideraciones fundamentales del diseño que incluye el formato de instrucción. Los números típicos de bits de una palabra son 8, 12, y 16, pero el número de bits en un byte es comúnmente 8. La razón de tener estos números de bits en las palabras se debe a códigos externos, tales como el código de ASCII que son códigos de 6, 7, y 8 bits. También algunos registros de dispositivos de I/O son de 6 o 8 bits. Sin embargo, el significado más aceptado de "palabra" es el grupo más grande de bits que pueden transmitirse simultáneamente entre la CPU y la memoria.

Uno de los atributos importantes de una memoria es la posibilidad de retener sus contenidos cuando su fuente está apagada. Una memoria que puede hacer esto se dice que es no volátil, y en caso contrario, se dice que es volátil. El tipo más conocido de la memoria no volátil es la memoria de núcleo magnético, y la memoria volátil de semiconductor más utilizada es la tecnología MOS.

Las memorias están también clasificadas de acuerdo -- con sus capacidades de leer y escribir, y caen en dos categorías principales:

1. Memoria de sólo lectura

La memoria de sólo lectura (ROM), como su nombre implica, sólo puede leer sus contenidos establecidos y es difícil cambiarlos, excepto con algún equipo especial. Básicamente existen cuatro tipos de ROM, y estos tipos están distinguidos por el medio de programar sus contenidos. En el primer tipo de ROM, los contenidos están determinados por una operación de enmascaramiento, el cual se realiza durante la fabricación del IC. Tal memoria no puede alterarse por el usuario y se refiere simplemente como ROM. Los contenidos del segundo tipo se pueden establecer por el usuario, si dispone del equipo adecuado, y se llama ROM programable (PROM). Una vez que la PROM está programada, sus contenidos nunca pueden cambiarse. El tercer y cuarto tipos se pueden no solamente programar por el usuario, sino también alterar muchas veces usando equipo especial. Estos se llaman ROM programable y borrrable (EPROM), y ROM programable y alterable eléctricamente -- (EAPROM). Obviamente, todas las ROM's son no volátiles.

Cuando se diseña una microcomputadora para una aplicación particular y existe un conjunto de instrucciones y/o - -

constantes, que no necesitan alterarse después de la instalación de la computadora, tal información se puede guardar en una ROM. Las memorias de control son ROM's.

Las PROM's están construidas de matrices de diodos implantados en IC's. Mediante las conexiones externas, ciertos diodos seleccionados se pueden quemar, y con esto la matriz de diodos queda programada permanentemente.

Las EPROM's se pueden programar por inyección de cargas y se borran mediante la luz ultravioleta. Las EPROM's se usan principalmente durante el desarrollo y etapas de prueba de un diseño, y se reemplazan por las ROM's o PROM's, cuando el diseño está completo. Las EPROM's se consideran como memorias no volátiles, pero pueden perderse sus contenidos con el transcurso del tiempo.

Las EAPROM's se programan como las PROM's, pero sus contenidos no son permanentes. Una EAPROM difiere de una EPROM en que la EAPROM puede reprogramarse por computadora, usando circuitos de lectura y escritura que están propiamente diseñados.

2. Memoria de acceso aleatorio.

La memoria de acceso aleatorio (RAM) es el nombre - -

comunmente usado para las memorias que se pueden leer y escribir, aunque su prefijo "acceso aleatorio" no es un nombre propio. En algunas aplicaciones en las cuales la microcomputadora está utilizada como un simple controlador, no necesita el uso de RAM's. Pero en muchas aplicaciones, se necesitan meter y almacenar más datos y estos datos son indispensables para la operación de la máquina, por lo tanto, las RAM's pueden ser utilizadas para guardar dichos datos. Por razones de confiabilidad operativa, bajo costo, y flexibilidad, la mayoría de los diseños incluye ambas memorias ROM y RAM. RAM puede ser no volátil, tal como la memoria de núcleo, o volátil, tal como la de MOS.

1.2.9 Dispositivos de Entrada/Salida e interfaces

Existe una variedad muy amplia de dispositivos de Entrada/Salida (I/O) o periféricos que pueden conectarse a una microcomputadora. Los dispositivos típicos periféricos son: monitores de despliegue de Tubo de Rayo Catodico (CRT) denominados terminales, diskette, convertidores Analógico/Digital (A/D) y Digital/Analógico (D/A), lector de cintas, impresora, graficadora etc.

La gran diversidad de aplicaciones de las microcomputadoras da por resultado que los periféricos utilizados en los sistemas de microcomputadora varien radicalmente de un

sistema al otro. Una microcomputadora puede utilizarse como un simple controlador, el cual incluye solamente un microprocesador y otros elementos electrónicos menores. Una microcomputadora puede también servirse como un sistema de manejo de datos, el cual puede requerir, además de una microcomputadora, convertidores A/D y D/A, aparatos digitales de I/O, terminal, graficadora o impresora, unidades de disco y cinta, y otros más.

Toda transferencia de datos, excepto la cual se realiza dentro de la CPU, se hace sobre un o más canales. Todos los dispositivos de I/O y la memoria principal entonces, deben estar de alguna manera conectados con estos canales. Esto se lleva a cabo por medio de interfaces y controladores. Un controlador es un circuito que se necesita para iniciar los comandos que envía la microcomputadora a los periféricos, y conocer las condiciones de los periféricos. En muchos casos, el controlador es una parte integral del periférico. Una interfaz es un circuito necesario para conectar el periférico y su circuito de control con el canal apropiado. La interfaz tiene que realizar alguna combinación de las siguientes funciones:

1. Disponer las condiciones del periférico a la microcomputadora.
2. Disponer "buffers" para guardar datos que están

entrando a la microcomputadora del periférico.

3. Transmitir comandos de la microcomputadora al periférico.
4. Disponer "buffers" para guardar datos que están saliendo de la microcomputadora al periférico.
5. Emitir a la microcomputadora una señal cuando una operación está completa.
6. Emitir a la microcomputadora una señal cuando un error ha ocurrido durante una operación.

La transferencia de datos entre un dispositivo de I/O y la CPU o la memoria puede clasificarse de acuerdo con la cantidad de datos como sigue:

Transferencia de palabras o byte -- Se trata de una transferencia en la cual sólo un byte o una palabra de información se transmite por un comando dado a la microcomputadora. Un terminal es un ejemplo de dispositivo que normalmente opera este tipo de transferencias.

Transferencia en bloque -- Se trata de una transferencia en la cual bloque de información se transmite por un simple comando a la microcomputadora. Tal transferencia tiene lugar entre los periféricos y la memoria, y se llama comunmente acceso directo de memoria (DMA). La mayoría de los dispositivos periféricos que requieren alta velocidad de transfe-

rencia son dispositivos de DMA, porque la CPU no puede ser capaz de mover la información tan rápido como los de DMA.

1.2.10 Canal externo.

Una microcomputadora puede tener un canal para el uso de memoria principalmente, y otros para el uso de periféricos, o tener un sólo canal para ambos propósitos. Si tiene solamente un canal externo, se conoce comúnmente como canal del sistema. Las líneas de un canal pueden clasificarse como se describe abajo. La figura 1-2 ilustra las características más importantes de un canal externo.

Líneas de dato.- Estas líneas son utilizadas para transferir información. Cuando la información se transfiere con la memoria, dicha información puede ser datos o instrucciones. Cuando la información se transfiere con los dispositivos de I/O, esta información puede ser: datos, condiciones del periférico, comandos de la microcomputadora o información de interrupción. El número de líneas de dato en un canal determina el número de bits que se pueden transferir simultáneamente, y por lo tanto, influye directamente en la velocidad de transferencias. En general, el número de las líneas de dato es el mismo que el de palabra, pero algunas veces, es la mitad del tamaño de palabra. Entonces, se necesitan dos transferencias para transferir una palabra. Una microcomputa

dora que tienen n líneas de dato se llama normalmente microcomputadora de n bits.

Líneas de dirección.- Las líneas de dirección se usan para transmitir las combinaciones de bits, las cuales se están decodificadas como direcciones que se conectan con el canal. En el caso de un módulo de memoria sólo, la interfaz de la memoria debe estar diseñada para reconocer todas las direcciones de las palabras de la memoria. En el caso de un sistema de memoria donde varios módulos de memoria pueden estar conectados al mismo canal, la interfaz de la memoria está diseñada en general, de modo que unos los primeros bits de la dirección sirvan para seleccionar los módulos, y el resto de los bits se usan para buscar una localidad particular dentro del módulo. Si los dispositivos periféricos de I/O se reparten un canal con la memoria, algunas de las direcciones deben estar reservadas para sus registros de interfaz.

En una arquitectura de canal múltiple, la mayoría de los dispositivos de I/O está conectada con diferentes canales denominados canales de I/O, el canal que se usa por los módulos de memoria se llama canal de memoria.

Finalmente, las líneas de dirección y las de dato pueden ser las mismas. En esta situación, las direcciones y los datos deben usarlas alternativamente, y la CPU y las interfa-

ces tienen que estar diseñadas para coordinar las actividades del canal.

Líneas de control.- Independientemente del arreglo de los canales que se usa, una cierta cantidad de información de control debe pasarse por unas líneas del canal llamadas líneas de control, entre la CPU, los módulos de memoria, y las interfaces. La comunicación de la información de control tiene una forma de "handshake". Esta información incluye en general, alguna combinación de las siguientes:

1. Petición del uso de canal, la cual se realiza por varias interfaces conectadas con el canal.
2. Concesión del uso de canal, la cual se realiza -- por un circuito de prioridad que está predeterminada.
3. Señal de interrupción, la cual indica eventos externos que requieren la atención de la CPU.
4. Señal de tiempo, la cual coordina las transferencias de datos y direcciones que tienen lugar en el canal.
5. Señales que indican fallas de máquina y fuente.

1.3 SOFTWARE.

Todas las operaciones que se realizan dentro de una -

microcomputadora, están dirigidas por una serie de funciones individuales. Una instrucción es una unidad de información, la cual indica a la computadora qué operación se va a realizar. Un conjunto de instrucciones que se usa para llevar a cabo una tarea computacional completa se llama programa. Un programa o una colección de programas, se llama en general -- "software".

1.3.1 Tipo de instrucción.

Las operaciones básicas que se realizan por cualquiera microcomputadora pueden dividirse en las siguientes categorías. Debido a la limitación del tamaño de las microcomputadoras, una microcomputadora puede no tener un repertorio suficiente de instrucciones capaz de realizar todas estas operaciones.

Transferencia.- Transferir información entre registros, memoria, pila, y la CPU.

Aritméticas.- Adición, substracción y negación. Estas operaciones caen en tres tipos:

1. Con acarreo.
2. Sin acarreo.
3. Decimal (o BCD).

Lógicas.- OR, AND, OR exclusivo, y complementación.

Corrimiento y rotación.- Corrimiento y rotación hacia la derecha y la izquierda, y corrimientos y rotaciones de palabras múltiples.

Función de índice y cuenta.- Incremento y decremento.

Manipulación de bit.- Selectivamente encender, apagar, y probar los bits dentro de una palabra o byte.

Ramificación.- Ordinariamente las instrucciones se toman de las localidades consecutivas de la memoria. Sin embargo, algunas veces es necesario saltar fuera de esta secuencia ordinaria. Las instrucciones de ramificación realizan esta función, y caen en tres categorías principales:

1. Ramificación incondicional, en la cual se sucede la rama sin tomar en cuenta las condiciones de la CPU.
2. Ramificación condicional, la rama se sucede o no dependiendo de alguna condición de las banderas del CR.
3. Llamado y regreso de subrutinas, una dirección de regreso se guarda en la pila cuando se llama a una subrutina, y se usa esta dirección para regresar de la subrutina. Los llamados y regresos pue

den ser condicionales o incondicionales.

Enlace.- Se trata de una combinación de incremento o decremento, comparación, prueba, y ramificación, su propósito es ejecutar un segmento de programa en repetición.

Comunicación y transferencia de I/O.- Indicar y realizar transferencia de los datos de I/O, probar las condiciones de I/O, dar comandos a I/O.

1.3.2. Modos de direccionamiento.

A fin de que todos los componentes de una microcomputadora puedan comunicarse, la microcomputadora debe tener algunos medios para identificar las localidades individuales de memoria principal, pila, registros de la CPU y registros de interfaz. Esto se hace por asignación de dirección a todas las localidades.

Debido a que la computadora usa generalmente la memoria para almacenar tanto instrucciones como datos, la microcomputadora usa diversos métodos, llamados modos de direccionamiento, para dar acceso a la información de la memoria. -- Los modos de direccionamiento más importantes están definidos abajo. Estas definiciones no son estándares y la terminología puede variar de un manual al otro.

Direccionamiento directo.- La dirección está especificada como parte de la instrucción.

Direccionamiento de registro.- Una forma de direccionamiento directo, en la cual el operando está en un registro y la dirección del registro es parte de la instrucción.

Direccionamiento indirecto.- La dirección está en una localidad cuya dirección está especificada en una parte de la instrucción. Esta localidad puede ser una localidad de memoria o un registro, en este caso, se usa la frase direccionamiento indirecto de registro.

Direccionamiento de base.- La dirección se forma sumando un número especificado al contenido de una localidad de memoria o registro. El direccionamiento de base puede usarse junto con el direccionamiento indirecto.

Direccionamiento de página cero.- Una forma de direccionamiento indirecto en la cual la dirección especificada en la instrucción es la dirección de una localidad de página cero, y el contenido de esta localidad es la dirección deseada o efectiva.

Direccionamiento relativo.- La dirección es la suma de un número especificado y el contenido presente del PC. El

número es generalmente una parte de la instrucción, pero también puede estar contenido en un registro de trabajo.

Direccionamiento de índice.- Este modo de direccionamiento es similar al relativo. La diferencia es que la dirección efectiva es la suma de un número especificado y el contenido del registro de índice en lugar del PC.

Direccionamiento inmediato.- En este modo de direccionamiento, el operando está dado directamente por la instrucción.

En la programación de una microcomputadora, existen numerosos programas de apoyo para facilitar al usuario la creación, corrección, revisión, ejecución del programa, y ligamiento de varios programas, tales como editor, ensamblador, monitor, cargador y ligador. Además de estos programas, existen otros programas importantes del sistema que facilita al usuario el manejo de dispositivos de I/O, y se llama sistema operativo. Todos estos programas desempeñan un papel sumamente importantes en el desarrollo del software de una microcomputadora. Las descripciones detalladas de estos programas pueden ser consultadas en los libros que tratan del desarrollo software de la computadora.

C A P I T U L O 2

MICROCOMPUTADORA MC68705-P3

2.1. DESCRIPCION GENERAL

La unidad de microcomputadora (MCU) MC68705-P3 es un miembro EPROM de la familia M6805 de microcomputadoras con bajo costo en un IC sólo. La EPROM que es programable por el usuario permite cambios de programa y aplicaciones de bajo volumen en comparación con las versiones programadas por máscara desde la fábrica. Las versiones de EPROM reducen también los costos de desarrollo y tiempo de vuelta para evaluación de prototipo de las versiones de ROM de máscara. Esta microcomputadora de 8 bits contiene una unidad de procesamiento central (CPU), un reloj incluido en el mismo IC, EPROM, ROM para bootstrap, RAM, Entrada/Salida (I/O) y un contador de eventos.

2.2. HARDWARE

La figura 2-1 muestra la arquitectura de la microcomputadora MC68705-P3 en diagrama de bloques, las señales de entrada y salida en el diagrama se describen a continuación:

TINT.- Esta línea permite a un evento externo interrumpir el procesador en forma asincrónica. También puede usarse esta línea como una entrada de encuesta usando las instrucciones BIL y EIH.

XTAL y EXTL.- Las conexiones de estas líneas proporcionan al oscilador del reloj que está incluido en el IC. Dependiendo del bit CLK del registro de opciones enmascaradas (MOR), puede conectarse a estas líneas, un oscilador de cristal, un resistor o una señal externa.

TIMER.- Esta línea se usa como una entrada externa para controlar el contador de eventos interno. Esta línea también detecta un nivel de voltaje alto que se usa para iniciar el programa de bootstrap.

RESET.- Esta línea se utiliza para reestablecer la MCU.

Vpp.- Esta línea se usa cuando se desea programar EPROM. Para programar la EPROM, se aplica un voltaje de programación y en operación normal, esta línea se conecta a la fuente de polarización.

Líneas de I/o (PA0-PA7, PB0-PB7, PC0-PC7).- Estas 20 líneas están arregladas dentro de dos puertos de 8 bits (A y B) y un puerto de 4 bits (C). Todas estas líneas pueden pro-

gramarse como entradas o salidas.

2.2.1. Memoria.

La figura 2-2 muestra la configuración de memoria de la MCU. La MCU es capaz de operar con 2048 bytes de memoria y registros de I/O con el PC. En la MCU68705-P3 se implementan 2041 bytes de estas localidades en la siguiente forma: 1804 bytes EPROM para el usuario, 115 byte ROM para bootstrap, 112 bytes RAM para el usuario, un registro de opciones enmascaradas (MOR), un registro de control de programa, y ocho bytes para I/O.

La EPROM se localiza en dos áreas. La área principal de la EPROM está en localidades de memoria de \$080 a \$783. La segunda área que es de 8 bytes se reserva para vectores de interrupción y reset, y está en localidades de memoria de \$7F8 a \$7FF. La MCU usa 9 de las 16 localidades más bajas de la memoria para el control de programa, puertos de I/O, dirección de puertos de I/O (DDR), y el contador de eventos (TIMER). El MOR está en la localidad de memoria \$784. Los 112 bytes RAM para el usuario incluyen 31 bytes como máximo para la pila.

2.2.2. Unidad central de procesamiento.

La CPU de la familia M6805 se implementa independien-

temente de la configuración de I/O o memoria. Consecuentemente, la CPU puede tratarse como un procesador central independiente que está comunicado con I/O y memoria por vía de canales internos de dato, dirección y control.

2.2.3. Registros.

La CPU de la familia M6805 contiene cinco registros disponibles para el programador. La figura 2-3 muestra estos registros:

Acumulador (A): El acumulador es un registro de propósito general de 8 bits y se usa para retener operandos y resultados del cálculo aritmético o manipulación de datos.

Registro de índice (X): El registro de índice es de 8 bits, y se usa para el modo de direccionamiento de índice. El registro de índice puede usarse para manipulación de datos y puede usarse también como una área de almacén temporal.

Contador de programa (PC): El contador de programa es de 11 bits que contiene la dirección de la siguiente instrucción para la ejecución de dicha instrucción.

Apuntador de pila (SP): El apuntador de pila es de 11 bits que contiene la dirección de la siguiente localidad

libre de la pila. Durante el reestablecimiento de la MCU o la ejecución de instrucción RSP, el SP se carga con la localidad \$07F. Los seis bits más significativos del SP son permanentemente 000011. Las subrutinas e interrupciones pueden aprovechar las localidades bajas \$061 (31 bytes máximos), lo cual permite al programador usar hasta 15 llamados de subrutina (son menos si ocurre interrupción).

Registro de condición (CR): El registro de códigos de condición es de 5 bits donde 4 bits se usan para indicar los resultados de la instrucción que acaba de ejecutarse. Estos bits pueden probarse individualmente por programa y tomar una acción específica dependiendo de las condiciones del resultado. Cada uno de los 5 bits se explica a continuación:

1. Medio acarreo (H): Encienden durante operaciones de ADD y ADC para indicar un sobreflujo ocurrido entre bit 3 y 4.

2. Interrupción (I): Cuando este bit se enciende, el contador de eventos y el interruptor externo (INT) están deshabilitados. Si ocurre una petición de interrupción mientras este bit está encendido, se inhibe la interrupción y se realiza cuando este bit se apaga.

3. Negativo (N): Cuando este bit está encendido, indica que fué negativo el resultado de la última operación --

aritmética, lógica, o de manipulación de datos.

4. Cero (Z): Cuando este bit está encendido, indica que fué cero el resultado de la última operación aritmética, lógica o de manipulación de datos.

5. Acarreo (C): Cuando este bit está encendido, indica que ocurrió un acarreo durante la última operación aritmética. Este bit también se afecta durante la prueba de bit o con instrucciones de corrimiento y rotación.

2.2.4. Contador de eventos.

El contador de eventos (TIMER) de la MC68705-P3 consiste en un contador programable de 8 bits, el cual se conduce por una establecedora de escala programable de 7 bits. Varias fuentes de reloj para el TIMER pueden ser seleccionadas, y diferentes modos de trabajo del TIMER pueden ser también -- utilizados por vía de programar el registro de control del -- TIMER (TCR) y/o el MOR. La configuración de estos dos registros se describen a continuación:

1.- Registro de control del contador de eventos.

La configuración del TCR se determina por el nivel lógico del bit 6 (TOPT) del MOR. Existen dos configuraciones del TCR, una es para TOPT=1 y la otra, para TOPT=0. Si TOPT=1,

el TCR emula la MC6805-P2 o MC6805-P4, y cuando TOPT=0, él da el control al TCR por programa. La descripción de cada bit - del TCR se presenta bajo.

b7	b6	b5	b4	b3	b2	b1	b0
TIR	TIM	1	1	1	1	1	1

\$009

TCR con TOPT=1 del MOR (MC6805-P2/P4 emulación)

b7	b6	b5	b4	b3	b2	b1	b0
TIR	TIM	TIN	TIE	PSC	PS2	PS1	PS0

\$009

TCR con TOPT=0 del MOR (TIMER programable)

b7, TIR: Petición de interrupción del TIMER - Se usa para iniciar la interrupción del TIMER.

1= Se enciende cuando el registro de datos del TIMER cambia a cero.

0= Se apaga por reestablecimiento externo o programa.

b6, TIM: Máscara de interrupción del TIMER - Se usa para inhibir la interrupción del TIMER, cuando él está en lógica "1".

1 = Se enciende por reestablecimiento externo o programa.

0 = Se apaga por programa.

b5, TIN: Exterior o interior - Selecciona la fuente de reloj siendo o externa TIMER (línea 7) o interna Ø2.

1 = Selecciona la fuente de reloj externa.

0 = Selecciona la fuente de reloj interna Ø2.

b4, TIE: Habilitación externa - Se usa para habilitar la línea externa TIMER o el reloj interno (si TIN=0) sin tomar en cuenta el estado de la línea TIMER, cuando TOPT=1, TIE está siempre en lógica "1".

1 = Habilita la línea externa TIMER.

0 = Deshabilita la línea externa TIMER.

b3, PSC: Apagamiento de la establecedora de escala - Cuando TOPT=0, genera un pulso, si se escribe un "1" al PSC, el cual apaga la establecedora de escala, cuando TOPT=1, no tiene ningún efecto.

b2, PS2: Selección de escala de establecedora de escala.
 B1, PSI. Estos bits están decodificados para seleccionar. Una de las 8 escalas de las establecedoras.

2. Registro de opciones enmascaradas.

El MOR está implementado en la EPROM, y contiene ceros antes de la programación. Cuando se usa el MOR para emu-

lar la MC6805-P2/P4, 5 de los 8 bits del MOR se usan en conjunción de la establecedora de escala. En otros casos, el bit 7 se usa para seleccionar el tipo de reloj, y los bits 3 y 4 no se usan. Los bits 0, 1 y 2 determinan la escala del TIMER. El bit 5 determina la fuente de reloj del TIMER. El valor del bit TOPT (6) se programa para configurar el TCR. La descripción de los bits del MOR se presenta a continuación:

b7	b6	b5	b4	b3	b2	b1	b0
CLK	TOPT	CLS			P2	P1	P0

\$784

b7, CLK: Tipo de reloj de oscilador.

1 = Reloj RC

0 = Reloj de cristal.

b6, TOPT: Opción de TIMER.

1 = Tipo de TIMER o establecedora de escala de la MC6805-P2/P4. Los bits 5, 2, 1 y 0 del MOR determinan las opciones enmascaradas equivalentes de la MC6805-P2/P4.

0 = Todos los bits del TCR están implementados como un TIMER programable. Los bits 5, 4, 2, 1 y 0 del MOR se establecen los valores iniciales de los bits del TCR correspondientes.

b5, CLS: Fuente de reloj del TIMER.

1 = Línea externa TIMER.

0 = Reloj interno $\phi/2$.

b4: No se usa si TOPT = 1, y se establece el valor inicial del bit TIE del TCR, si TOPT=0.

b3: No se usa.

b2, P2: Opción de escala - Se utilizan en la misma forma, P1. Que la de los bits del TCR correspondientes, b0, P0.

2.2.5. Interrupción.

La MC68705-P3 puede ser interrumpida por tres caminos diferentes: el interruptor externo \overline{INT} , la interrupción del TIMER, o la instrucción de la interrupción software (SWI). Cuando alguna interrupción ocurre: la instrucción presente se completa, se suspende el procesamiento corriente, se mete el estado presente de la CPU a la pila, se enciende el bit I del CR, la dirección de la rutina de interrupción se obtiene de la dirección del propio vector de interrupción, y se ejecuta la rutina de interrupción. La rutina de interrupción debe terminarse con la instrucción de regreso de la interrupción (RTI), la cual permite a la CPU continuar el procesamiento del programa interrumpido.

Cuando la instrucción presente está completa, el procesador chequea todas las interrupciones hardware suspensas, y si no está enmascarada alguna interrupción, atiende la interrupción, en otros casos, sigue ejecutando la siguiente instrucción. Las interrupciones enmascaradas se guardan para servicios posteriores.

Si una interrupción externa y una del TIMER ambas están suspensas, la externa se sirve primero. La SWI se ejecuta como una instrucción ordinaria. La interrupción del TIMER se sirve solamente cuando el bit TIR está encendido, y el TIM está apagado. Durante el servicio de interrupción del TIMER, el bit TIR debe apagarse por el programa de la rutina de interrupción, para que se inhíba otra petición de interrupción del TIMER.

2.2.6. Entrada/Salida.

Existen 20 líneas de I/O, la línea \overline{INT} también puede ser otra línea de entrada con las instrucciones BIL y BIH. -- Las líneas de puertos A, B y C pueden programarse como entradas o salidas, dependiendo del estado del registro DDR correspondiente. La programación de los puertos de I/O se lleva a cabo por escribir una lógica "1" como salida o una lógica "0" como entrada a los bits del DDR correspondientes. El estado de los puertos se muestra en la siguiente tabla.

DDR bit	Dato de salida bit	Estado de salida	Entrada a MCU
1	0	0	0
1	1	1	1
0	X	alta impedancia	Línea

2.3. SOFTWARE

La MCU de MC68705-P3 posee un conjunto de instrucciones eficiente, y 10 modos de direccionamiento potentes, por lo tanto, facilita al usuario la programación, o bien las aplicaciones de software.

2.3.1. Tipo de instrucción.

La MCU de MC68705-P3 tiene un conjunto de 59 instrucciones básicas, las cuales derivan 207 códigos de operación utilizables cuando se combinan los 10 modos de direccionamiento. Todas las instrucciones y sus códigos de operación están presentados en apéndice I.

Este conjunto de instrucciones incluye casi todas las categorías de operaciones básicas, descritas en la sección 1.3.1, excepto la de enlace. Vale la pena mencionar que en esta microcomputadora, la operación de comunicación y transfe

rencia de I/O es igual a la de transferencia descrita en la sección 1.3.1, y se combina la operación de ramificación con la manipulación de bits, es decir, además de que cualquier bit de la memoria de página cero puede probarse, ocurre también la ramificación de programa dependiendo de las condiciones del resultado de prueba, con las instrucciones BRSET y BRCLR.

2.3.2. Modos de direccionamiento.

La MCU de MC68705-P3 tiene 10 modos de direccionamiento, los cuales son: directo, inmediato, extendido, relativo, inherente, de índice sin offset, con offset de 8 bits y con offset de 16 bits, BSC y BTB. Es necesario declarar que el direccionamiento inmediato y el inherente son equivalentes al indirecto e inmediato que se han definido en la sección 1.3.2, respectivamente, y existe una diferencia entre el modo de direccionamiento de índice de esta microcomputadora y el de direccionamiento de índice definido en la sección 1.3.2. En esta MCU, existen tres modos de direccionamiento de índice diferentes. Uno es el sin offset, el cual significa que la dirección efectiva está contenida en el X, el con offset de 8 bits significa que la dirección efectiva es la suma del contenido del X y un número de 8 bits especificado, y el con offset de 16 bits significa que la dirección efectiva es la suma del contenido del X y un número de 16 bits especificado. El modo

de direccionamiento de BSC y BTB sirve especialmente para las manipulaciones de bits, BSC es para establecer y borrar un bit de una memoria de la página cero, y BTB es para probar un bit de la memoria y ramificarse dependiendo del resultado de prueba.

C A P I T U L O 3

SIMULADOR DE MICROCOMPUTADORA MC68705-P3

La simulación de una microcomputadora consiste en el proceso de ejecutar un programa, el cual está escrito en instrucciones de la microcomputadora simulada, en una máquina diferente. La arquitectura de la microcomputadora está simulada por un bloque de memoria en la máquina principal. Eso es, el PC, las banderas CC, los registros de trabajo tales como el acumulador A, el registro de índice X, el apuntador de pila SP, y la memoria de la microcomputadora están representadas por localidades de memoria de la máquina principal. Durante la simulación, los contenidos de estas localidades deben ser los mismos que los de la microcomputadora simulada. Cada instrucción de la microcomputadora se simula por varias instrucciones de la máquina principal. Algunos simuladores se combinan con un ensamblador cruzado para que el usuario pueda no solamente ejecutar y verificar su programa, sino también desarrollar su programa en otra máquina diferente.

La simulación de cualquiera máquina es diferente, pero, un simulador debe en general, realizar las siguientes funciones:

1. Perseguir la ejecución de un programa.
2. Desplegar los contenidos de los registros de trabajo y la memoria.
3. Modificar los contenidos de los registros de trabajo y la memoria.

3.1. INTRODUCCION AL SIMULADOR SIM05

El SIM05 es un simulador que simula la microcomputadora MC68705 en la máquina CROMEMCO con el sistema CDOS. La entrada del SIM05 es un programa ensamblado en códigos de máquina de la MC68705, el cual está grabado en un diskette con su propio nombre del archivo. Como todos los simuladores, el SIM05 es una herramienta efectiva para ejecutar y revisar programas escritos en lenguaje de máquina de la MC68705 en las máquinas CROMEMCO bajo el sistema CDOS.

El SIM05 consiste principalmente en tres partes: programa principal, subrutinas de comandos y subrutinas de ejecución de instrucciones. El programa principal realiza funciones tales como manejo de archivos, manejo de los comandos. Su diagrama de flujo está mostrado en la figura 3-1. Las subrutinas de comandos realizan funciones especificadas por los comandos del SIM05, las cuales son ejecución de programa, despliegue de memoria y registros, modificación de memoria y registros etc. Las descripciones detalladas de cada comando se

presentan en las siguientes secciones. Las subrutinas de ejecución de instrucciones tienen tareas que completan 207 instrucciones de la MC68705. Se utiliza estas subrutinas para la ejecución de los programas del usuario. Es obvio que dentro del SIM05, existen otras subrutinas de apoyo las cuales realizan funciones tales como conversión de números, manejo de pantalla, indicación de errores, etc.

3.2. DESCRIPCION DE COMANDOS DEL SIM05

En cualquier simulador, siempre existen unos comandos, los cuales son disponibles para el usuario. Aunque estos comandos se varían de un simulador al otro, sus funciones principales son la ejecución y revisión del programa del usuario con efectividad. El SIM05 posee 11 comandos mayores y 2-subcomandos, y el usuario puede usarlos para efectuar las funciones deseadas fácilmente. Las descripciones de estos comandos se presentan abajo. Es conveniente declarar que dentro de las descripciones se usa símbolos / y XXXX, para indicar el regreso de carro y números hexadecimales de 4 dígitos, respectivamente.

3.2.1. Comandos.

1. / - Ejecución por paso

/ : ejecutar una instrucción.

2. B - Puntos de verificación

a).B/: Desplegar todos los puntos de verificación permanentes, máximo 12.

b).B XXXX1,...,XXXX12/ : Establecer o modificar los puntos de verificación permanentes deseados.

3. E - Salida

E/ : Salir del SIM05 y regresar al sistema CDOS.

4. G - Ejecución.

a).G/: Ejecutar el programa desde la dirección presente del PC, hasta la dirección indicada por algún punto de verificación permanente o la última del programa. También el usuario puede suspender la ejecución del programa por entrar el subcomando S sin despliegue de los registros de la MC68705.

b). G XXXX/: Realizar la misma función que la del G/ excepto que se empieza la ejecución del programa de la dirección indicada por XXXX.

c).G XXXX1, XXXX2/: Realizar la misma función que la del G/, excepto que la ejecución se empieza de la dirección - indicada por XXXX1 y se añade otro punto de verificación temporal establecido por XXXX2.

5. L - Lista

L/: Desplegar los comandos disponibles de SIM05.

6. M - Memoria

a). M/: Desplegar 320 localidades de memoria empezando por la dirección intrínseca 0080H.

b). M XXXX/: Realizar la misma función que la del M/, excepto que se empieza el despliegue de la localidad indicada por XXXX.

7. N - Ejecución de varias instrucciones.

N XX/: Ejecutar XX instrucciones.

8. O - Otro.

O/: Ejecutar otro programa del usuario.

9. R - Registros.

R/: Desplegar todos los registros de la MC68705, los cuales son:

a). Registro de las banderas (C).

b). Acumulador (A).

c). Registro de índice (X).

d). Contador de programa (PC).

e). Apuntador de pila (SP).

f). Dato de puerto A (PA).

- g). Dato de puerto B (PB).
- h). Dato de puerto C (PC).
- i). Dirección de puerto A (DA).
- j). Dirección de puerto B (DB).
- k). Dirección de puerto C (DC).
- l). Dato de TIMER (TD).
- m). Control de TIMER (TC).
- n). Opciones enmascaradas (MO).
- o). Interrupción de TIMER (TI).
- p). Interrupción externa (EI).
- q). Interrupción por software (SWI).
- r).- Reset (RES).

10. SM - Substituye Memoria

a). SM/: Sustituir los contenidos de la memoria desde la dirección de inicio intrínseca 0080H hasta la última intrínseca 0783H. También puede terminar el usuario la sustitución por el subcomando T.

b). SM XXXX/: Realizar la misma función que la del SM, excepto que la dirección de inicio sea XXXX.

c). SM XXXX1, XXXX2/: Realizar la misma función que la del SM, excepto que la dirección de inicio sea XXXX1 y la final sea XXXX2.

11. SR - Substituye Registros

SR/ : Sustituir los contenidos de los registros mencionados en el comando R. El usuario puede terminar la sustitución por el subcomando T.

3.2.2 Subcomandos

1. S/ : Suspender la ejecución del programa y usarlo solamente en conjunción del comando G.
2. T/ : Terminar la sustitución de los contenidos de la memoria o los registros y usarlo solamente en conjunción del comando SM o SR.

3.3 SIMULACION DE INSTRUCCIONES

La simulación de las instrucciones de la MC68705 se lleva a cabo por 207 subrutinas, las cuales ejecutan todas las instrucciones de la MCU. La mayoría de estas subrutinas son sencillas, y no necesitan la intervención del usuario, excepto las instrucciones BIL y BIH. Para que el SIMOS ejecute la instrucción BIH, el usuario debe apretar la tecla I que está simulada como la línea TNT, y ejecute la BIL, el usuario debe dejar dicha tecla en libre.

3.4 SIMULACION DE ENTRADA /SALIDA

En la MC68705, la transferencia de los datos de I/O se realiza en la misma forma que la de los datos de la memoria. Entonces, durante la escritura y lectura de los datos de I/O, se aparece primero en la pantalla, la especificación de los puertos que están usando, y luego se le pide al usuario entrar los datos del puerto, los cuales están trabajando como entradas por: bit*

?
 donde* indica la línea del puerto correspondiente. El usuario entonces debe escribir 1 o 0 a la pantalla para especificar los datos externos. Si no escribe nada el usuario o escribió mal, se vuelve a pedir al usuario los datos, hasta que entre el usuario los datos necesarios correctamente.

3.5 SIMULACION DE INTERRUPCIONES

Las interrupciones externas y del TIMER se simulan -- por la tecla I y C, respectivamente. El número de apretón de la tecla I, realizado por el usuario se guarda en un registro interno. Al terminar la ejecución de cada instrucción, el -- SIM05 verifica el bit I del CR, si I=1, sigue la ejecución -- del programa corriente, si I=0 y el contenido del registro interno no es cero, el SIM05 decrementa el contenido de dicho registro y realiza la interrupción externa. Cuando el TIMER de la MCU trabaja en el modo programable y con el reloj exter

no, es decir, el bit TOPT del MOR está en lógica "0" y el bit TIE, en lógica "1", cada vez que el usuario aprieta la tecla C, el SIMOS decrementa el contenido del contador. La interrupción del TIMER ocurre bajo las siguientes condiciones:

1. El bit I del CR está en lógica "0"
2. No existe la interrupción externa.
3. El bit TIR del TCR está en lógica "1"
4. El bit TIM del TCR está en lógica "0".

3.6 GUIA DEL USO DEL SIMOS

Al fin de usar el simulador SIMOS, el usuario debe leer bien las instrucciones del uso del SIMOS las cuales están descritas a continuación:

1. Entrar la clave del SIMOS la cual es SIMOS.

Al terminar de meter la clave del SIMOS se aparecen unos mensajes en la pantalla y se le pregunta al usuario:

¿Cuál es el nombre del archivo?

2. Entrar nombre del archivo.

El formato del nombre del archivo es I: Nom. Ext, donde I; es la indicación del drive donde está el diskette del usuario, en nuestro caso I solamente puede ser A o B.

Nom: es el nombre del archivo del usuario y puede --
ser 8 caracteres por lo máximo.

Ext: es la extensión del archivo y puede tener por --
lo máximo 3 caracteres.

Al terminar de entrar el nombre del archivo, se le --
pregunta:

¿Cuál es la última dirección de su programa?

3. Entrar la última dirección de su programa.

Como todos los programas del usuario se almacenan en --
la EPROM de la MC68705, esta dirección debe ser mayor que --
0080H y menor que 0783H. Al terminar este paso, se aparece --
una lista de los comandos como si hubiera entrado el comando --
L y se inspira por:

4. Entrar comando

Este comando debe ser uno de los 11 comandos disponi --
bles dependiendo de la función que desea el usuario realizar.
El uso de la mayoría de los comandos es simplemente entrar el
comando correspondiente, pero es necesario describir el uso --
de los comandos SM y SR con más detalle.

a). SM.

Al entrar el comando SM, SM XXXX o SM XXXXI, XXXX2. - Se aparece primero la dirección de la localidad desde la cual se empieza la sustitución y su contenido, por ejemplo:

0080: A6

Y luego se inspira por:

?

Si el usuario quiere modificar el contenido de esta localidad, entre el dato deseado. Si no entre / y se aparece en seguida la dirección de la siguiente localidad y su contenido, y de nuevo se inspira por ?, Así sucesivamente hasta encontrar la dirección final de la memoria o el subcomando T.

b). SR

Al entrar el comando SR, primeramente se despliega to dos los registros como si hubiera entrado el comando R y se aparece:

CC=

?

Lo cual es el registro de las banderas. Para modificar los contenidos de este registro, el usuario debe meter -- 1's o 0's dependiendo de cuál bandera quiere el usuario encender o apagar, bajo el orden de las banderas, que es CZNIH. - Si alguna bandera no requiere la modificación, puede dejar un

espacio en lugar donde está la bandera correspondiente, por ejemplo:

CC=

?10 1

Esto significa que se encienden las banderas C y H, se apaga la bandera Z y se dejan las banderas N y I como anterior. Al terminar la modificación de las banderas, se aparecen los siguientes registros en forma similar, es decir:

Reg=

?

Donde Reg es la abreviación de los registros. Si el usuario quiere modificar sus contenidos, entre los datos deseados, si no, entra /. Así sucesivamente hasta el último registro que es RES o hasta encontrar el subcomando T.

Se merece la pena también mencionar que cada vez que se suspende la ejecución del programa, lo cual se realiza por el comando G o N, se aparecen los contenidos de todos los registros, a menos que se suspenda la ejecución del programa por el subcomando S. Para salir del SIMOS y regresar al CDOS, existen dos remedios: uno es el uso del comando E y otro es el uso de las teclas C y CTRL simultáneamente.

Durante el uso del SIMOS, pueden aparecerse unas men-

sajes las cuales indican los errores provocados por el usuario. En seguida describimos estos mensajes y sus significados.

1. No se encuentra el archivo

Indica que el archivo que se introduce por el usuario no está en el diskette especificado.

2. Inválida la dirección

Indica que la última dirección del programa que se entra por el usuario, está fuera de la EPROM de la MC68705.

3. Inválido el comando

Indica que el comando se entra por el usuario tiene un error de sintaxis o las direcciones o datos que se escriben por el usuario no están en hexadecimal. Nótese que si un número hexadecimal es de 4 bits o de 2 bits, y el usuario escribió dicho número de 2 bits o de 4 bits, respectivamente, se aparecerá también este error.

4. La dirección XXXX es inválida.

Indica que las direcciones utilizadas en algunos comandos o en el programa del usuario están fuera de la zona disponible.

5. Error en el programa.

Indica los errores de la programación del usuario los cuales pueden ser instrucciones falsas, saltos invalidos, sobrecarga de pila etc. En el apéndice II, se da un ejemplo -- del uso del SIMOS.

3.7 CONCLUSIONES

Como se ha dicho anteriormente que ningún simulador - es igual, y sus comandos disponibles se varían de un simula-- dor al otro. EL SIMOS también es único. Aunque el SIMOS es simple, él es útil, es fácil usarlo y posee casi todas las -- funciones importantes de un simulador. Además, el SIMOS es flexible para su modificación, es decir, es fácil modificar - las subrutinas existentes y agregar otras subrutinas para me-- jorar y sofisticar sus funciones.

G R A F I C A S

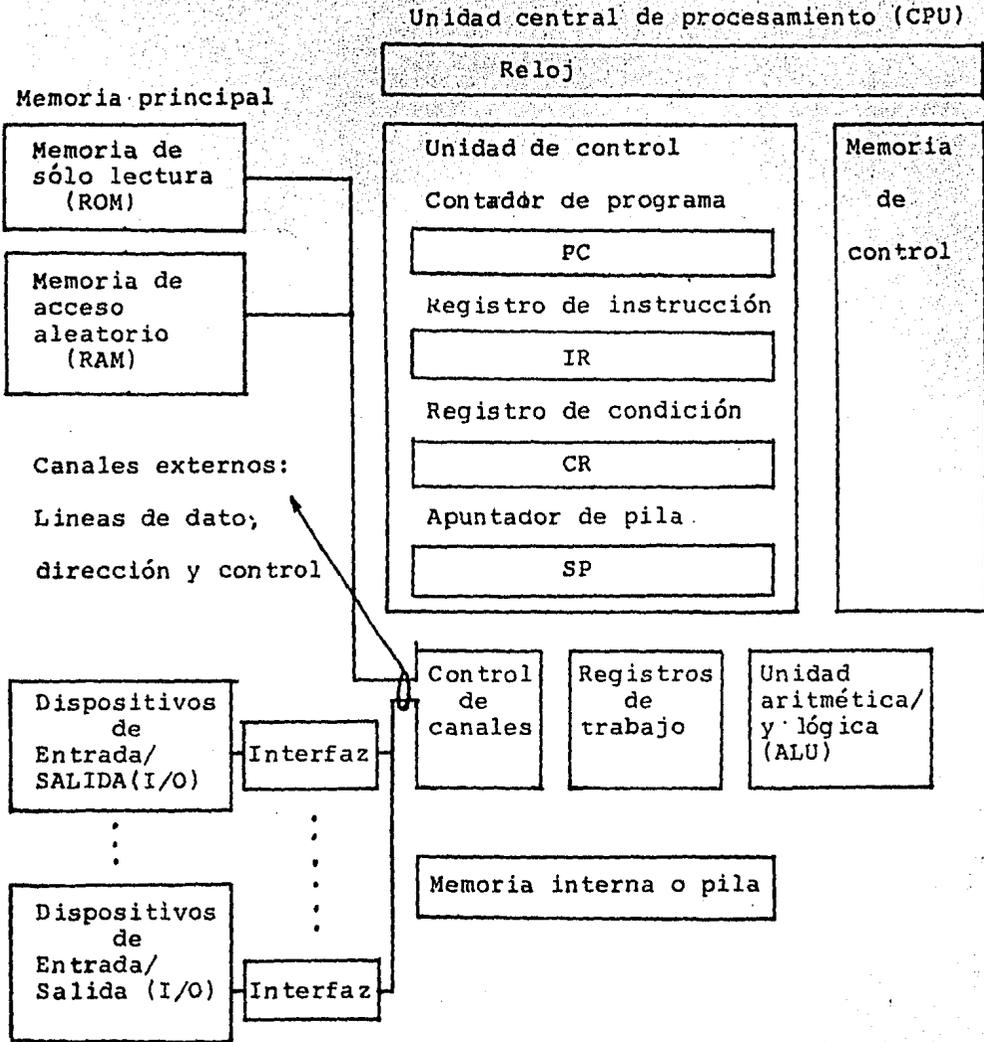


Figura 1-1 Distribución general de una microcomputadora.

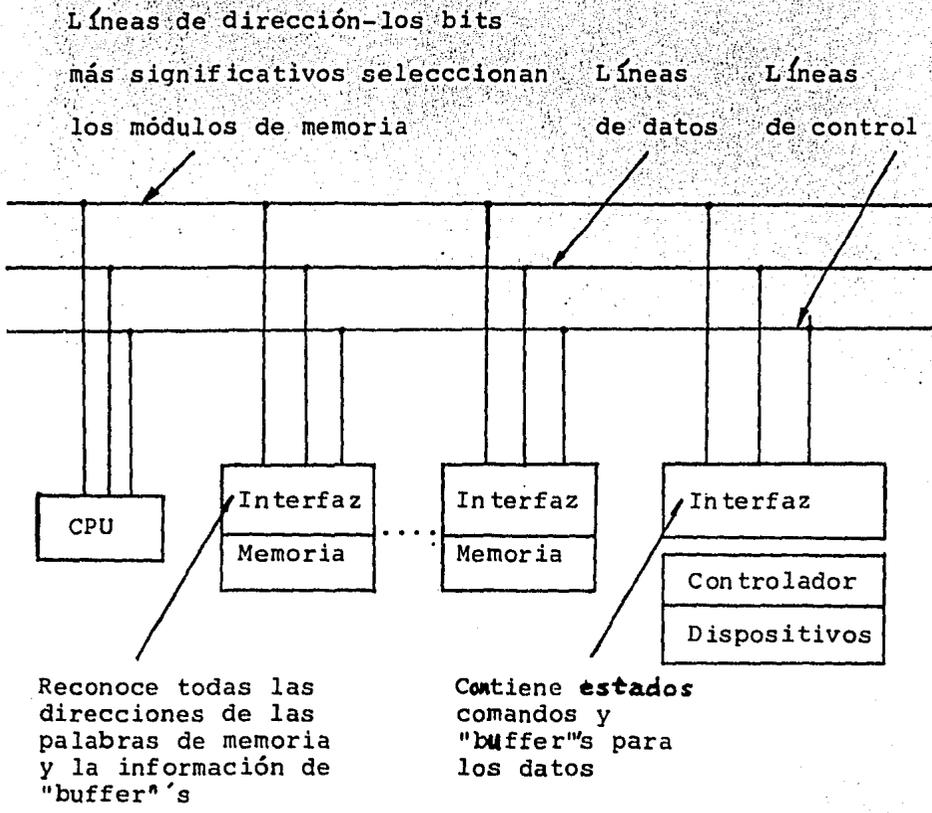


Figura 1-2 Características más importantes de un canal externo.

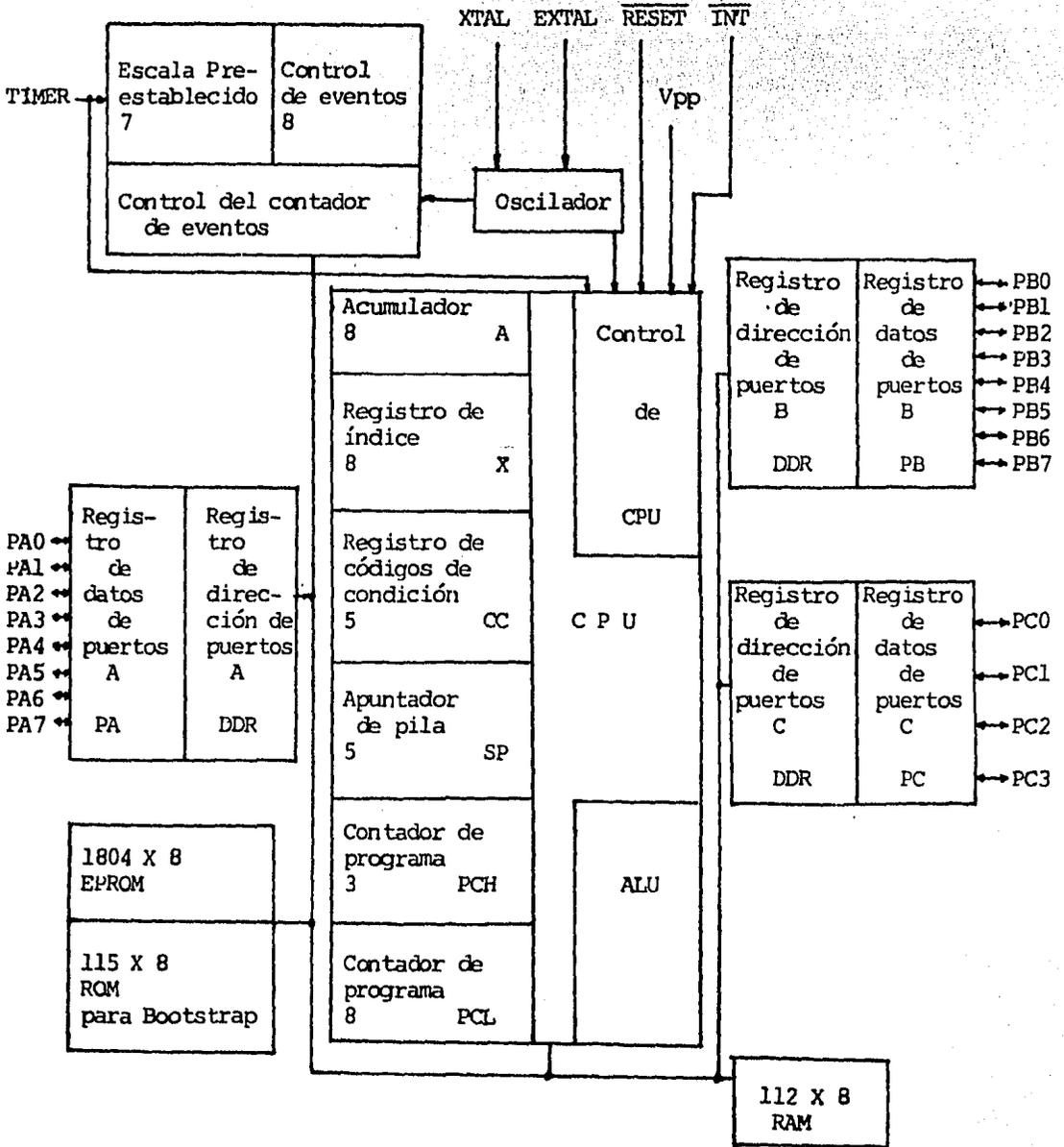


Figura 2-1 .: Arquitectura de la MC68705-P3

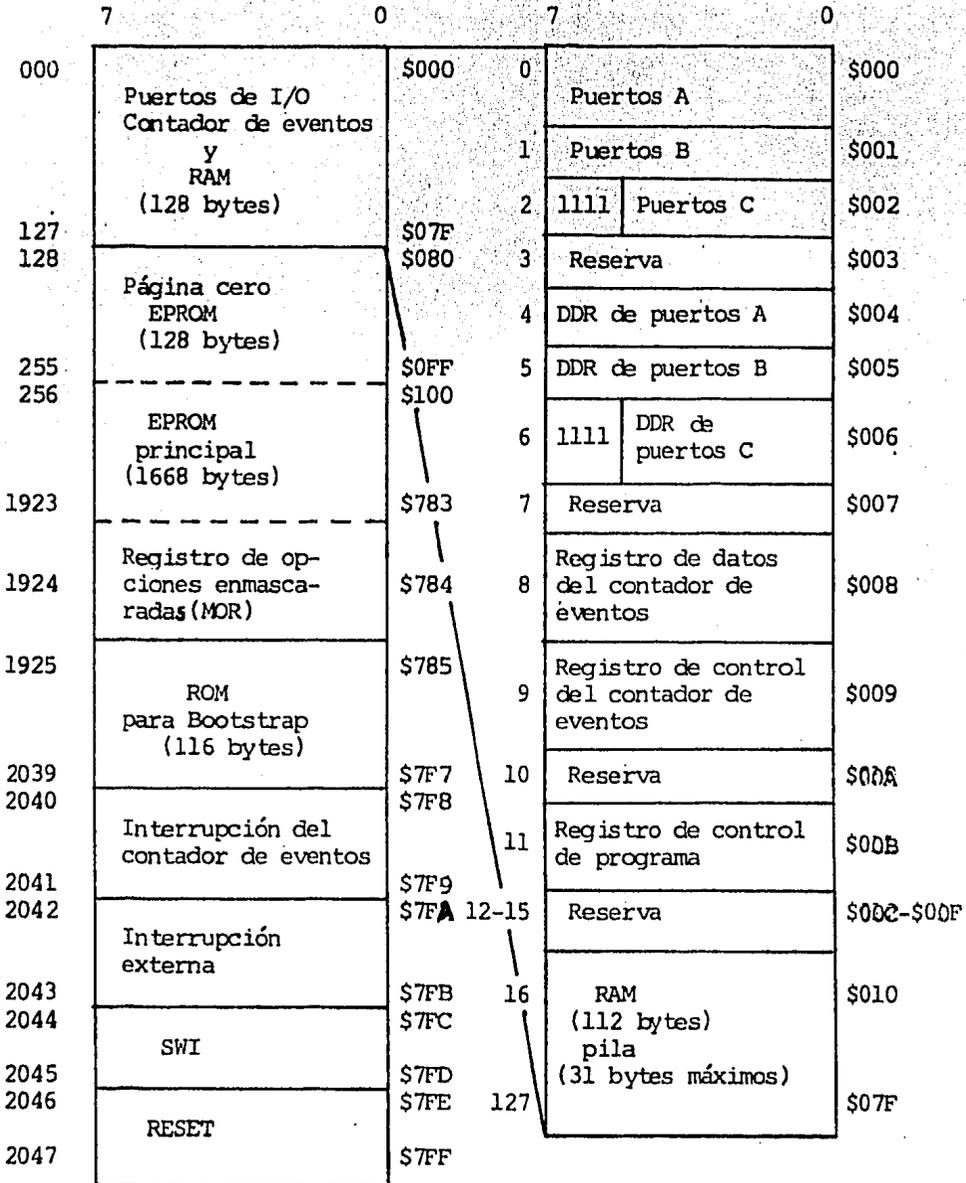


Figura 2-2 Configuración de memoria de la MC68075-P3

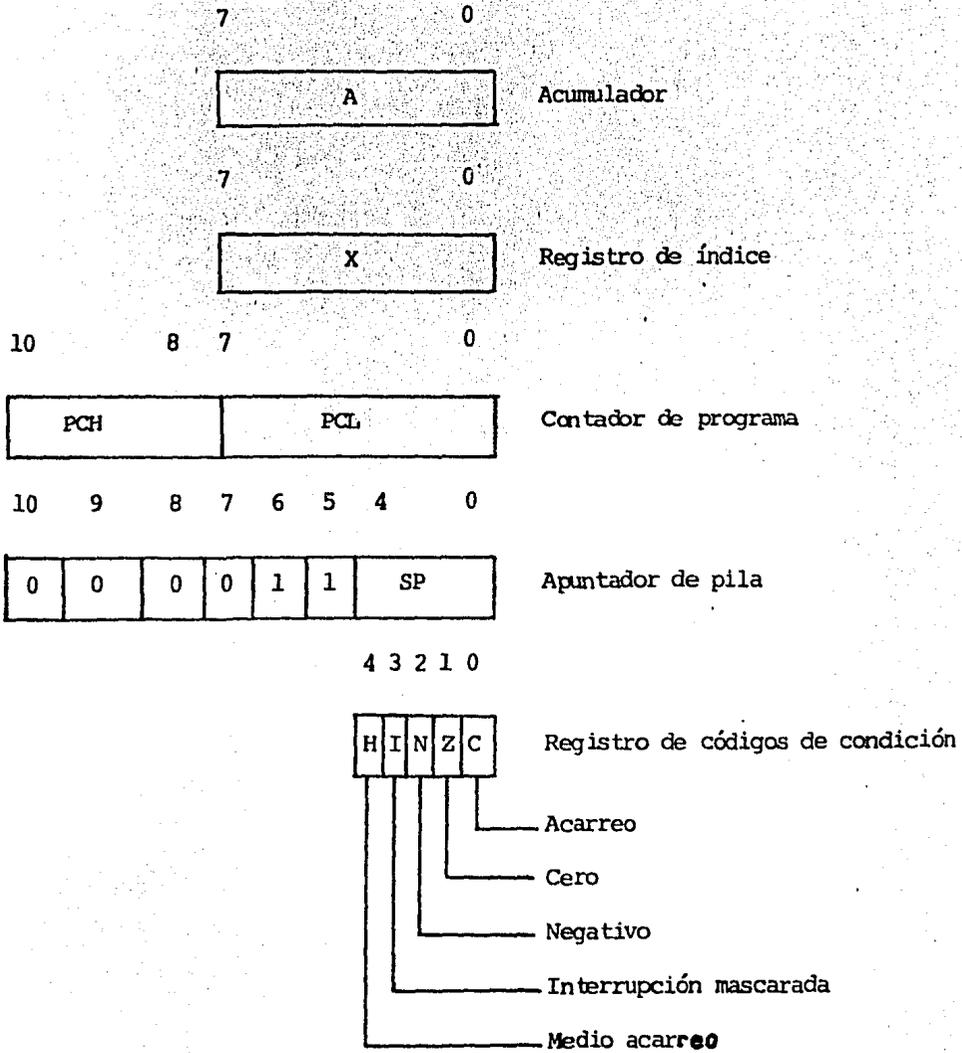


Figura 2-3 Registros de la MC68705-P3

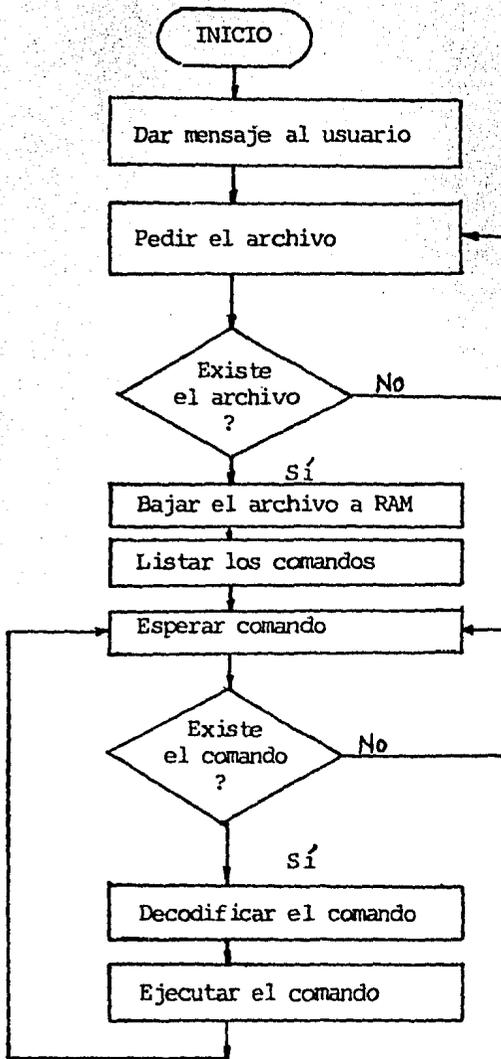


Figura 3-1 Diagrama de flujo de programa principal

APENDICE I

INSTRUCTION SET NUMERICAL LISTING

OP	MNEM	MODE	HMOS	CMOS	#
00	BRSET0	BTB	10	5	3
01	BRCLR0	BTB	10	5	3
02	BRSET1	BTB	10	5	3
03	BRCLR1	BTB	10	5	3
04	BRSET2	BTB	10	5	3
05	BRCLR2	BTB	10	5	3
06	BRSET3	BTB	10	5	3
07	BRCLR3	BTB	10	5	3
08	BRSET4	BTB	10	5	3
09	BRCLR4	BTB	10	5	3
0A	BRSET5	BTB	10	5	3
0B	BRCLR5	BTB	10	5	3
0C	BRSET6	BTB	10	5	3
0D	BRCLR6	BTB	10	5	3
0E	BRSET7	BTB	10	5	3
0F	BRCLR7	BTB	10	5	3
10	BSET0	BSC	7	5	2
11	BCLR0	BSC	7	5	2
12	BSET1	BSC	7	5	2
13	BCLR1	BSC	7	5	2
14	BSET2	BSC	7	5	2
15	BCLR2	BSC	7	5	2
16	BSET3	BSC	7	5	2
17	BCLR3	BSC	7	5	2
18	BSET4	BSC	7	5	2
19	BCLR4	BSC	7	5	2
1A	BSET5	BSC	7	5	2
1B	BCLR5	BSC	7	5	2
1C	BSET6	BSC	7	5	2
1D	BCLR6	BSC	7	5	2
1E	BSET7	BSC	7	5	2
1F	BCLR7	BSC	7	5	2
20	BRA	REL	4	3	2
21	BRN	REL	4	3	2
22	BHI	REL	4	3	2
23	BLS	REL	4	3	2
24	BCC	REL	4	3	2
25	BCS	REL	4	3	2
26	BNE	REL	4	3	2
27	BEQ	REL	4	3	2
28	BHCC	REL	4	3	2
29	BHCS	REL	4	3	2
2A	BPL	REL	4	3	2
2B	BMI	REL	4	3	2
2C	BMC	REL	4	3	2

INSTRUCTION SET NUMERICAL LISTING (CONTINUED)

OP	MNEM	MODE	HMOS	CMOS	#
2D	BMS	REL	4	3	2
2E	BIL	REL	4	3	2
2F	BIH	REL	4	3	2
30	NEG	DIR	6	5	2
33	COM	DIR	6	5	2
34	LSR	DIR	6	5	2
36	ROR	DIR	6	5	2
37	ASR	DIR	6	5	2
38	LSL	DIR	6	5	2
39	ROL	DIR	6	5	2
3A	DEC	DIR	6	5	2
3C	INC	DIR	6	5	2
3D	TST	DIR	6	4	2
3F	CLR	DIR	6	5	2
40	NEGA	INH	4	3	1
43	COMA	INH	4	3	1
44	LSRA	INH	4	3	1
46	RORA	INH	4	3	1
47	ASRA	INH	4	3	1
48	LSLA	INH	4	3	1
49	ROLA	INH	4	3	1
4A	DECA	INH	4	3	1
4C	INCA	INH	4	3	1
4D	TSTA	INH	4	3	1
4F	CLRA	INH	4	3	1
50	NEGX	INH	4	3	1
53	COMX	INH	4	3	1
54	LSRX	INH	4	3	1
56	RORX	INH	4	3	1
57	ASRX	INH	4	3	1
58	LSLX	INH	4	3	1
59	ROLX	INH	4	3	1
5A	DECX	INH	4	3	1
5C	INCX	INH	4	3	1
5D	TSTX	INH	4	3	1
5F	CLR X	INH	4	3	1
60	NEG	IX1	7	6	2
63	COM	IX1	7	6	2
64	LSR	IX1	7	6	2
66	ROR	IX1	7	6	2
67	ASR	IX1	7	6	2
68	LSL	IX1	7	6	2
69	ROL	IX1	7	6	2
6A	DEC	IX1	7	6	2
6C	INC	IX1	7	6	2

INSTRUCTION SET NUMERICAL LISTING (CONTINUED)

OP	MNEM	MODE	H MOS	CMOS	#
6D	TST	IX1	7	5	2
6F	CLR	IX1	7	6	2
70	NEG	IX	6	5	1
73	COM	IX	6	5	1
74	LSR	IX	6	5	1
76	ROR	IX	6	5	1
77	ASR	IX	6	5	1
78	LSL	IX	6	5	1
79	ROL	IX	6	5	1
7A	DEC	IX	6	5	1
7C	INC	IX	6	5	1
7D	TST	IX	6	4	1
7F	CLR	IX	6	5	1
80	RTI	INH	9	9	1
81	RTS	INH	6	6	1
83	SWI	INH	11	10	1
8E	STOP	INH	—	2	1
8F	WAIT	INH	—	2	1
97	TAX	INH	2	2	1
98	CLC	INH	2	2	1
99	SEC	INH	2	2	1
9A	CLI	INH	2	2	1
9B	SEI	INH	2	2	1
9C	RSF	INH	2	2	1
9D	NOP	INH	2	2	1
9F	TXA	INH	2	2	1
A0	SUB	IMM	2	2	2
A1	CMP	IMM	2	2	2
A2	SBC	IMM	2	2	2
A3	CPX	IMM	2	2	2
A4	AND	IMM	2	2	2
A5	BIT	IMM	2	2	2
A6	LDA	IMM	2	2	2
A8	EOR	IMM	2	2	2
A9	ADC	IMM	2	2	2
AA	ORA	IMM	2	2	2
AB	ADD	IMM	2	2	2
AD	BSR	IMM	8	6	2
AE	LDX	IMM	2	2	2
B0	SUB	DIR	4	3	2
B1	CMP	DIR	4	3	2
B2	SBC	DIR	4	3	2
B3	CPX	DIR	4	3	2
B4	AND	DIR	4	3	2
B5	BIT	DIR	4	3	2
B6	LDA	DIR	4	3	2

INSTRUCTION SET NUMERICAL LISTING (CONTINUED)

OP	MNEM	MODE	HMOS	CMOS	#
B7	STA	DIR	4	4	2
B8	EOR	DIR	4	3	2
B9	ADC	DIR	4	3	2
BA	ORA	DIR	4	3	2
BB	ADD	DIR	4	3	2
BC	JMP	DIR	3	3	2
BD	JSR	DIR	7	5	2
BE	LDX	DIR	4	3	2
BF	STX	DIR	5	4	2
C0	SUB	EXT	5	4	3
C1	CMP	EXT	5	4	3
C2	SBC	EXT	5	4	3
C3	CPX	EXT	5	4	3
C4	AND	EXT	5	4	3
C5	BIT	EXT	5	4	3
C6	LDA	EXT	5	4	3
C7	STA	EXT	6	5	3
C8	EOR	EXT	5	4	3
C9	ADC	EXT	5	4	3
CA	ORA	EXT	5	4	3
CB	ADD	EXT	5	4	3
CC	JMP	EXT	4	4	3
CD	JSR	EXT	8	6	3
CE	LDX	EXT	4	4	3
CF	STX	EXT	5	5	3
D0	SUB	IX2	6	5	3
D1	CMP	IX2	6	5	3
D2	SBC	IX2	6	5	3
D3	CPX	IX2	6	5	3
D4	AND	IX2	6	5	3
D5	BIT	IX2	6	5	3
D6	LDA	IX2	6	5	3
D7	STA	IX2	7	6	3
D8	EOR	IX2	6	5	3
D9	ADC	IX2	6	5	3
DA	ORA	IX2	6	5	3
DB	ADD	IX2	6	5	3
DC	JMP	IX2	5	5	3
DD	JSR	IX2	9	7	3
DE	LDX	IX2	6	5	3
DF	STX	IX2	7	6	3
E0	SUB	IX1	5	4	2
E1	CMP	IX1	5	4	2
E2	SBC	IX1	5	4	2
E3	CPX	IX1	5	4	2

INSTRUCTION SET NUMERICAL LISTING (CONCLUDED)

OP	MNEM	MODE	H MOS	CMOS	#
E4	AND	IX1	5	4	2
E5	BIT	IX1	5	4	2
E6	LDA	IX1	5	4	2
E7	STA	IX1	6	5	2
E8	EOR	IX1	5	4	2
E9	ADC	IX1	5	4	2
EA	ORA	IX1	5	4	2
EB	ADD	IX1	5	4	2
EC	JMP	IX1	4	4	2
ED	JSR	IX1	8	6	2
EE	LDX	IX1	5	4	2
EF	STX	IX1	6	5	2
F0	SUB	IX	4	3	1
F1	CMP	IX	4	3	1
F2	SBC	IX	4	3	1
F3	CPX	IX	4	3	1
F4	AND	IX	4	3	1
F5	BIT	IX	4	3	1
F6	LDA	IX	4	3	1
F7	STA	IX	5	4	1
F8	EOR	IX	4	3	1
F9	ADC	IX	4	3	1
FA	ORA	IX	4	3	1
FB	ADD	IX	4	3	1
FC	JMP	IX	3	3	1
FD	JSR	IX	7	5	1
FE	LDX	IX	4	3	1
FF	STX	IX	5	4	1

SIM05

APENDICE II

Hola! Como esta usted!

SIM05 es un simulador de microcomputadora MC68705.
Lea bien los comandos legales antes de usar SIM05!

Cual es nombre de su archivo ?

DAA.COM

Cual es la ultima direccion de su programa?

00A0

B/, B XXXX1, ..., XXXX2/, E/, G/, G XXXX/, G XXXX1, XXXX2/, N XX/,
L/, M/, M XXXX/, O/, R/, SM/, SM XXXX/, SM XXXX1, XXXX2/, SR/,
!M

0080: 25 04 A1 99 23 08 40 A0 60 40 AD 02 99 81 28 03
0090: AB 06 B1 AB 06 29 02 A0 06 B1 A6 2F AB E5 AD E0
00A0: 9D 03 0A 40 4A 00 32 05 4C 2F 55 B9 55 AE 04 EC
00B0: E0 00 04 9E 1A
00C0: 1A
00D0: 1A
00E0: 1A
00F0: 1A
0100: EB C3 20 28 3A 68 2F A7 20 12 2A C6 48 7E 3C 37
0110: C8 EB 21 0B 00 19 22 C6 48 EB 18 04 CD 37 31 D8
0120: 11 C2 3F 06 08 CD EF 27 01 08 00 20 0A 09 3A E4
0130: 3F AE CB 7F C8 ED 42 ED B0 7E 32 E4 3F 97 3C C9
0140: 78 01 00 00 21 BD 4C 08 D5 C5 E5 72 23 73 11 0D
0150: 00 19 EB 21 23 00 19 7D 12 13 7C 12 13 21 F2 39
0160: 01 21 00 ED B0 E1 D1 D5 E5 01 0D 00 09 CD 00 39
0170: D1 D5 CD 70 31 C1 E1 D1 19 C5 E3 C1 24 08 3D 20
0180: C6 C9 26 08 2E 03 22 DE 3F 11 BF 4C 62 6B 3A 68
0190: 2F 4F 0D 14 D5 1A 3C 28 06 CD C0 0B 38 01 EB D1
01A0: 0D 20 F0 7E 3C 37 C8 E5 01 0B 00 11 D0 48 ED B0
01B0: D1 1B 1B CD 70 31 21 D0 48 A7 C9 D5 1A 47 13 1A

!R

I A=00 X=00 PC=0080 SP=007F PA=00 PB=00 PC=00 AD=00 BD=00 CD=00
TD=FF TC=40 MD=00 TI=0000 II=0000 SWT=0000 RES=0000

!SR

I A=00 X=00 PC=0080 SP=007F PA=00 PB=00 PC=00 AD=00 BD=00 CD=00
TD=FF TC=40 MD=00 TI=0000 II=0000 SWT=0000 RES=0000

CC=

?

A=

?

X=

?

PC=

?009A

SP=

?T

!B

0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000

!B 0080,0086,008E,0093,0099,009E

0080,0086,008E,0093,0099,009E,0000,0000,0000,0000,0000,0000

!

I A=2F X=00 PC=009C SP=007F PA=00 PB=00 PC=00 AD=00 BD=00 CD=00
TD=FF TC=40 MD=00 TI=0000 II=0000 SWT=0000 RES=0000

!N 05

NI A=8C X=00 PC=00B9 SP=007D PA=00 PB=00 PC=00 AD=00 BD=00 CD=00
TD=FF TC=40 MO=00 TI=0000 II=0000 SWT=0000 RES=0000

!L
B/, B XXXX1,....,XXXX2/, E/, G/, G XXXX/, G XXXX1,XXXX2/, N XX/,
L/, M/, M XXXX/, O/, R/, SM/, SM XXXX/, SM XXXX1,XXXX2/, SR/,
!SM 00B2,00B4

00B2: A1

?

00B3: 99

?

00B4: 23

?

!M 0093

0093: AB 06 29 02 A0 06 81 A6 2F AB E5 AD E0 9D 03 0A
00A3: 40 4A 00 32 05 4C 2F 55 B9 55 AE 04 EC E0 00 04
00B3: 9E 1A
00C3: 1A
00D3: 1A
00E3: 1A
00F3: 1A EB C3 20
0103: 28 3A 68 2F A7 20 12 2A C6 48 7E 3C 37 C8 EB 21
0113: 0B 00 19 22 C6 48 EB 18 04 CD 37 31 D8 11 C2 3F
0123: 06 08 CD EF 27 01 08 00 20 0A 09 3A E4 3F AE CB
0133: 7F C8 ED 42 ED B0 7E 32 E4 3F 97 3C C9 78 01 00
0143: 00 21 BD 4C 08 D5 C5 E5 72 23 73 11 0D 00 19 EB
0153: 21 23 00 19 7D 12 13 7C 12 13 21 F2 39 01 21 00
0163: ED B0 E1 D1 D5 E5 01 0D 00 09 CD 00 39 D1 D5 CD
0173: 70 31 C1 E1 D1 19 C5 E3 C1 24 08 3D 20 C6 C9 26
0183: 08 2E 03 22 DE 3F 11 BF 4C 62 6B 3A 68 2F 4F 0D
0193: 14 D5 1A 3C 28 06 CD C0 0B 38 01 EB D1 0D 20 F0
01A3: 7E 3C 37 C8 E5 01 0B 00 11 D0 48 ED B0 D1 1B 1B
01B3: CD 70 31 21 D0 48 A7 C9 D5 1A 47 13 1A 4F 13 B0
01C3: 20 05 3E FF 12 D1 C9 D5 0B 79 1B 12 78 1B 12 D1
!G 009A

C I A=14 X=00 PC=009E SP=007D PA=00 PB=00 PC=00 AD=00 BD=00 CD=00
TD=FF TC=40 MO=00 TI=0000 II=0000 SWT=0000 RES=0000

!G 00B0,00B8

C I A=14 X=00 PC=0086 SP=007D PA=00 PB=00 PC=00 AD=00 BD=00 CD=00
TD=FF TC=40 MO=00 TI=0000 II=0000 SWT=0000 RES=0000

!

C NI A=EC X=00 PC=00B7 SP=007D PA=00 PB=00 PC=00 AD=00 BD=00 CD=00
TD=FF TC=40 MO=00 TI=0000 II=0000 SWT=0000 RES=0000

!D

Cual es nombre de su archivo ?

MULTIPLY.COM

Cual es la ultima direccion de su programa?

00AA

B/, B XXXX1,....,XXXX2/, E/, G/, G XXXX/, G XXXX1,XXXX2/, N XX/,
L/, M/, M XXXX/, O/, R/, SM/, SM XXXX/, SM XXXX1,XXXX2/, SR/,
!D

Cual es nombre de su archivo ?

B:DIVIDE.COM

Cual es la ultima direccion de su programa?

00FF

B/, B XXXX1,....,XXXX2/, E/, G/, G XXXX/, G XXXX1,XXXX2/, N XX/,
L/, M/, M XXXX/, O/, R/, SM/, SM XXXX/, SM XXXX1,XXXX2/, SR/,
!ABC

Invalido el comando

!E

Suerte! Adios!

B.

BIBLIOGRAFIA

1. Glenn A. Gibsoo / Yu-- Cherg Lin
"Microcomputers for engineers and Scientists"
Ed. Prentice - Hall, Inc., 1980
2. Lance A. Leventhal
"Introduction to microprocessors: Software, Hardware. -
Programming".
Ed. Prentice - Hall, Inc., 1978.
3. Motorola
"M6805 HMOS M146805 CMOS Family, microcomputer / micropro
cesser User's Manual"
2 Edición. Ed. Prentice Hall, Inc., 1983.
4. Rafiquzzaman, Mohamed.
"Microprocessors and microcomputer development Systems"
Hasper Row Publishers, Inc.
5. William Barden, JR.
"The Z-80 microcomputer Handbook"
Howard W. Sams Co., Inc., 1978.
6. "Cromemco CDOS instruction manual"
CROMENCO, Inc.
7. "Cromemco Z-80 macro assembler instruction manual"
CROMENCO, Inc.