



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

22

“ ARAGON ”

2ej

APUNTES PARA LA MATERIA:
ORGANIZACION DE COMPUTADORAS

T E S I S

Que para obtener el Título de:

INGENIERO EN COMPUTACION

Presenta:

SARA MONTOYA SANTOS

San Juan de Aragón, Estado de México 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.

CONTENIDO

INTRODUCCION

JUSTIFICACION

ANTECEDENTES

PLAN PROPUESTO

Capítulo I. TRANSFERENCIA ENTRE REGISTROS Y MICRO-OPERACIONES

1.1 Transferencia entre registros	1
1.2 Micro-operaciones lógicas	10
1.3 Micro-operaciones aritméticas	12
1.4 Micro-operaciones de corrimiento	15
1.5 Generaciones de función de control	16
1.6 Representación de datos aritméticos (Magnitud y signo, overflow, desplazamientos y punto flotante)	20

Capítulo II. DISEÑO LOGICO DE UN PROCESADOR

II.1 Organización del procesador	23
II.2 Unidad Aritmética y Lógica (ALU)	26
II.3 Registros	29
II.4 Buses (canales)	35

Capítulo III. DISEÑO DE LA LOGICA DE CONTROL

III.1 Organización del control	39
III.2 Control alambrado (hard-wired)	40
III.3 Control Microprogramado	48
III.4 Control con PLA (Program Logic Array)	52

Capítulo IV. ORGANIZACION DE LA MEMORIA

IV.1 Tecnología y características	60
IV.2 Memoria Virtual	67
IV.3 Memoria Cache	70
IV.4 Memoria Interleaved	74
IV.5 Memoria Asociativa	77

CONTENIDO

Capítulo V. ORGANIZACION DEL CONTROL MICROPROGRAMADO

V.1 Control microprogramado	82
V.2 Secuencia de las direcciones	86
V.3 Control de la Unidad Central de Proceso (CPU)	87
V.4 Control de formatos de palabra	90
V.5 Consideraciones de tiempo	92
V.6 Ventajas y desventajas de la microprogramación	93

Capítulo VI. DISEÑO DE UNA COMPUTADORA

VI.1 Configuración del sistema	100
VI.2 Instrucciones de la computadora	103
VI.3 Control y tiempo	105
VI.4 Ejecución de instrucciones	108
VI.5 Diseño de control	111
VI.6 Diseño de registro	115
VI.7 Diseño de la consola	118

Capítulo VII. ORGANIZACION DE ENTRADA/SALIDA

VII.1 Repaso de dispositivos periféricos	122
VII.2 Interfases de E/S	127
VII.3 Interfase asíncrona serie	129
VII.4 Acceso directo a memoria (DMA)	132
VII.5 Sistema de interrupciones	136
VII.6 Comunicación entre periféricos	140
VII.7 Procesadores E/S	144
VII.8 Periféricos	146

I N T R O D U C C I O N

El mundo de las computadoras es tan extenso y complejo que sería difícil contemplar en su totalidad el tema en un sólo libro; sin embargo, estos apuntes están enfocados al auxilio didáctico tanto de alumnos, como de profesores, y al lector en general.

Uno de los objetivos del presente trabajo es plasmar la organización de las computadoras digitales, para lograr diseñar desde el punto de vista lógico, los diferentes componentes de una computadora; así como su interfase con el mundo de los periféricos.

En el capítulo uno, se describe la transferencia entre registros y microoperaciones, con el fin de familiarizarse con el concepto microoperación y la realización de las instrucciones mediante ésta.

Posteriormente se contempla el diseño lógico de un procesador, lo cual coadyuvará al desarrollo y análisis de problemas relacionados con procesadores.

En seguida, se amplía el análisis del diseño lógico. Se trata del diseño de la lógica de control, en el cual se analiza el control alambrado, microprogramado y el PLA (Program Logic Array).

Uno de los capítulos más interesantes es el 4, en el cual se estudia la organización de las memorias y periféricos, con lo que se persigue ampliar, consolidar y enlazar conceptos

manejados anteriormente. Se plantean aspectos de las memorias virtual, cache, interleaved y asociativa.

En el siguiente capítulo, se aborda la organización del control microprogramado. En este punto nos evocaremos a conocer conceptos y aplicaciones del CPU (Control de la Unidad Central de Proceso), secuencia de direcciones y el control de palabras, con el objetivo de dominar el control de una computadora digital.

En seguida se analizan las memorias y periféricos. Se explica la configuración del sistema, control y tiempos, diseño de control, de registro y de la consola.

Por último se contempla la organización de entrada/salida, en donde se refuerzan conceptos manejados en los capítulos anteriores y se dan a conocer principios básicos de la programación y de los dispositivos de entrada/salida.

JUSTIFICACION

La industria mexicana cada vez más consciente de la importancia del uso de las computadoras, cada día exige más preparación de los Ingenieros y por ende mayor calidad en las tareas realizadas, para lo cual establece criterios y parámetros que les permite analizar perfiles, y decidir cuál o cuáles personas son aptas para realizar con éxito planes y proyectos que eleven la imagen tanto de la empresa como del País.

Por lo anterior, es necesario y urgente exigir y dar una buena preparación profesional, tanto del alumnado como del personal docente, para lo cual creemos que se deben de considerar materias relevantes dentro del plan de estudios, como lo es la Organización de las Computadoras; ya que en ellas se involucran temas tan trascendentes como lo es la estructura, diseño y funcionamiento interno de una computadora.

La finalidad de esta tesis es proporcionar conocimientos, que permitan una preparación profesional que eleve la calidad y nivel de competencia de nuestros estudiantes universitarios.

ANTECEDENTES DEL TRABAJO

Las computadoras han sido y serán una herramienta de trabajo imprescindible a nivel mundial.

Aunque en 1975 ya se podían encontrar computadoras en casi todas las organizaciones medianas y grandes, todavía eran demasiado costosas para grupos pequeños o individuos. No obstante, los avances en la Electrónica estaban a punto de introducir una categoría totalmente nueva de computadoras: máquinas de inventario con la capacidad de los antiguos gigantes y con precios al alcance de los individuos.

Actualmente, la gente está consciente de las ventajas que brindan las computadoras. Por otro lado, es impresionante el paso agigantado con que la Electrónica va avanzando, por lo cual es casi imposible mantenerse al nivel de desarrollo tecnológico de Países tan potentes como Japón, China y E.U.A, pero lo que sí podemos hacer es aprovechar la información que poseemos.

En todo momento de la historia de la computadoras, se han tomado en cuenta puntos importante como son: El saber las posibilidades y limitaciones de las computadoras (software y hardware), saber el manejo de las mismas, conocer el funcionamiento interno de ellas y por último apreciar la gran utilidad y facilidad que nos brindan.

PLAN PROPUESTO

Antes de entrar de lleno al desarrollo del tema, es conveniente proponer un plan, en el que se exponga la metodología para llevar a cabo los objetivos buscados en la tesis, así como la utilidad de la misma.

En primera instancia, se requiere que el alumno tenga conocimientos de Electrónica tanto Digital como Analógica; de Memorias; Minicomputadoras y Microprocesadores; así como también, haber tenido contacto físico con Circuitos Integrados, Memorias y Microprocesadores.

Una vez confirmado el anterior requisito, se recomienda la siguiente metodología:

- 1) Se proporcionará al alumno conceptos e ideas que reafirmen, o en su caso despierten su interés. Estos conceptos están enfocados al análisis de la transferencia entre registro y micro-operaciones, con el fin de que más adelante no se dificulte el lenguaje utilizado ni se tergiverse el mensaje proporcionado.
- 2) En seguida se analizará y estudiará la estructura de un Microprocesador, de una manera teórico-práctica. Esto coadyuvará a entender y, a la vez diseñar nuestros propios proyectos conectados y por supuesto controlados por un Microprocesador.
- 3) Una vez logrado lo anterior, se continúa con la exposición de formas de control de un Microprocesador, con lo que se busca versatilidad y calidad en el diseño de proyectos, y

sobre todo, analizar y estudiar internamente una computadora.

- 4) Al finalizar la exploración de un Microprocesador, se prosigue con el análisis de uno de los componentes más importantes conectados a ellos, que son las Memorias, sus diferentes tecnologías y características. Con ello se busca otorgar al alumno ideas y conceptos con los cuales debe de ser capaz de enlazar y decidir cual o cuales estructuras son mejores, en cuanto a costo, diseño y capacidad.
- 5) Posteriormente se pretende ampliar el tema del control microprogramado, ya anteriormente se había analizado y expuesto los diferentes tipos de control, en base a ello se considera que el control microprogramado, además de ser uno de los más usuales, es más práctico y ventajoso.
- 6) En este punto, se vincularán todos los temas analizados anteriormente, pues se trata del diseño de una computadora. En este momento se propone hacer participar al alumno con diseños y estructuras, auxiliándose de herramientas proporcionadas en el desarrollo del curso.
- 7) Finalmente, se analizará la organización de los dispositivos de Entrada/Salida; es uno de los temas más extensos, ya que refuerza conceptos e ideas vistas.

Se considera que para llevar a cabo los objetivos previstos, es necesario un Laboratorio, en el cual el alumno tenga contacto con las computadoras internamente, en el que se

les permita realizar prácticas que complementen lo visto en clase.

El desarrollo de estos temas busca principalmente el desenvolvimiento teórico-práctico del alumno en la industria, ya que se persigue darle una preparación profesional muy sólida, con la que desarrolle ideas y diseños que ayuden a la superación del País, y sobre todo ser capaz de dominar y por ende solucionar cualquier problema que se presente en el interior de una computadora.

I. TRANSFERENCIA ENTRE REGISTROS Y MICRO-OPERACIONES

I.1 TRANSFERENCIA ENTRE REGISTROS.

Antes de analizar el tema de la transferencia entre registros, es necesario considerar los siguientes conceptos:

a) CARACTERISTICAS DE UNA COMPUTADORA DIGITAL:

Una computadora digital es aquella que cuenta directamente los números (o dígitos) que representan numerales, letras u otros símbolos especiales, dentro de sus características principales se encuentran las siguientes:

- Divide el problema en elementos aritméticos.*
- Representa los números por medio de un patrón discreto codificado (datos digitales), tal como perforaciones de una tarjeta o la presencia de pulsos.*
- Las operaciones se efectúan empleando relativamente muchos dispositivos aritméticos intercambiables (sumadores, registros, acumuladores, etc.)*
- Se necesitan muchos dispositivos, por lo tanto, el costo es alto y la programación difícil.*
- Elementos idénticos usados en secuencia (principalmente agrupación en serie).*
- Precisión ilimitada (10^{12} o más).*

- El almacenamiento concentrado de información en el espacio de memoria es intercambiable e ilimitado en el tiempo.
- La computadora digital combina datos aritméticos, sin relación con el sistema que representa. El tiempo de las operaciones generalmente no tiene correspondencia con el tiempo real.
- Puede representar números, letras y caracteres alfanuméricos.
- Adecuada para manejar procesos discretos, datos estadísticos y problemas numéricos de naturaleza de negocios y científicos.

b) DEFINICIONES BASICAS.

Un sistema digital se define por los registros que contiene y las operaciones que hace con la información binaria almacenada en ellos, el número de micro-operaciones diferentes de un sistema dado es finito. La complejidad para el diseño es una secuencia de operaciones para lograr la tarea necesaria de procesamiento de datos. Esta abarca la formulación de las funciones de control o el desarrollo del micro-programa. Un sistema digital puede ser construido por medio de los circuitos MSI (Mediana Escala de Integración) tales como: Registros, decodificadores, ALU, memoria y multiplexores. Algunos sistemas digitales son adecuados por el diseño LSI (Baja Escala de Integración) con componentes tales como la unidad de proceso, el secuenciador del micro-programa y la

unidad de memoria. Estos sistemas pueden ser micro-programados para adecuarse a especificaciones requeridas y el método del micro-programa opera a nivel de transferencia entre registros y debe especificar cada micro-operación en el sistema.

c) UNIDAD CENTRAL DE PROCESAMIENTO (CPU).

Es aquella parte de un sistema digital ó un computador digital que configura las operaciones en el sistema. Está compuesta por un número de registros y de funciones digitales que conforman micro-operaciones aritméticas, lógicas, de desplazamiento y transferencia. La unidad de proceso se llama CPU cuando se combina con una unidad de control que supervisa la secuencia de micro-operaciones.

Una operación puede ser configurada en una unidad de proceso con una micro-operación sencilla o con una secuencia de microoperaciones. Por ejemplo, la multiplicación de dos números binarios almacenados en dos registros, puede ser configurada con un circuito combinacional, que realiza la operación por medio de compuertas. Tan pronto como las señales se propagan a través de las compuertas, el producto estará disponible y puede ser transferido a un registro de destino con un pulso de reloj sencillo.

Un computador ó CPU debe manipular no solamente datos, sino también códigos de instrucción y direcciones que vienen de la memoria. El registro que almacena y manipula el código de

operación de instrucciones se considera como parte de la unidad de control. Las funciones de control que inician la secuencia de operaciones consisten de señales de tiempo que le dan secuencia a las operaciones una por una. Una función de control es una variable binaria que en un estado binario inicia una operación y en el otro la inhibe.

d) REGISTROS.

Un registro, es un conjunto de elementos de información relacionados entre sí que se tratan como unidad; o bien un dispositivo capaz de almacenar una cantidad específica de datos.

Los registros de un computador digital pueden ser clasificados del tipo operacional o de almacenamiento de información binaria en sus FLIP-FLOP'S, y además tiene compuertas combinacionales capaces de realizar tareas de procesamiento de datos. Un registro de almacenamiento se usa solamente para el almacenamiento temporal de la información binaria y esta información no puede ser alterada cuando se transfiere adentro y afuera del registro.

UNIDAD LOGICA ARITMETICA (ALU). Una ALU es una función multioperacional digital de lógica combinatorial. Esta puede realizar un conjunto de operaciones aritméticas básicas y un conjunto de operaciones lógicas. El ALU tiene un número de líneas de selección para elegir una operación particular de la

unidad.

Las líneas de selección se decodifican dentro del ALU de manera que las k variables de selección pueden especificar hasta 2^k operaciones diferentes.

En su forma más simple, una computadora consiste en 5 partes principales e independientes desde el punto de vista funcional: Unidad de entrada, de memoria, aritmética y lógica, de salida y de control, como se muestra en la figura 1.1

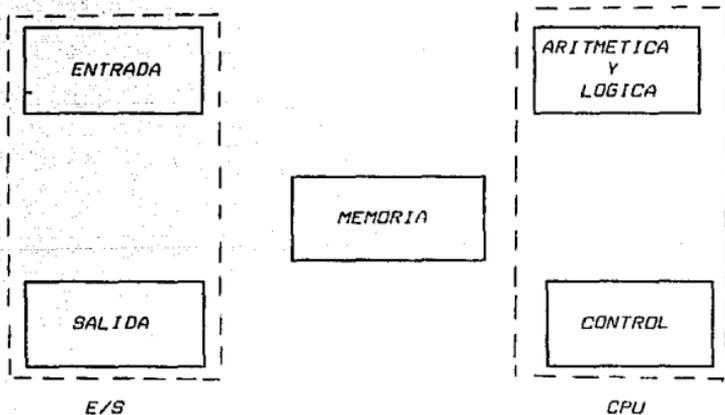


FIGURA 1.1 UNIDADES FUNCIONALES BASICAS DE UNA COMPUTADORA

La figura 1.2 muestra el diagrama de bloque de un ALU de 4 bits. Las 4 entradas de A se combinan con las 4 entradas de B para generar una operación en las salidas F . El terminal de selección de modo S_2 distingue entre las operaciones aritméticas y lógicas.

Las dos entradas de selección de función S_1 y S_0 especificando la operación aritmética ó lógica que se va a generar. Con tres variables aritméticas (con S_2 en un estado) y cuatro operaciones lógicas (con S_2 en otro estado). Los arrastres de entrada salida tienen significado solamente durante una operación aritmética.

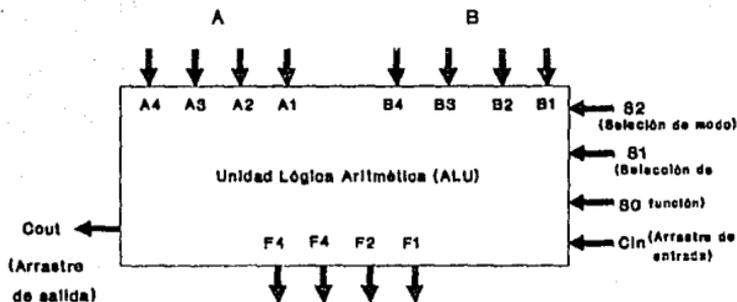


FIGURA 1.2

El arrastre de entrada en la posición menos significativa de un ALU se usa muy a menudo como una cuenta variable de selección que puede doblar el número de operaciones aritméticas. De esta manera, es posible generar cuatro operaciones más para un total de ocho operaciones aritméticas.

El ALU está compuesto por un número de registros y de funciones digitales, que conforman micro-operaciones aritméticas, lógicas, de desplazamiento y de transferencia. El ALU recibe la información de los registros y realiza una operación dada de la manera especificada por el control. El resultado de la operación se transfiere al registro de destino. Por definición, el ALU es un circuito combinacional; de manera que toda la operación de transferencia entre registros puede realizarse durante el intervalo de un pulso de reloj.

Una vez expuestos los conceptos anteriores; pasaremos a estudiar lo que es la transferencia entre registros.

En la sección anterior se especificó que dentro de una computadora la actividad se gobierna por medio de instrucciones. Para realizar una tarea dada, se almacena en la memoria principal un programa adecuado que consiste en un conjunto de instrucciones. Las instrucciones individuales se traen de la memoria al Procesador y éste ejecuta las instrucciones especificadas. Además de las instrucciones, es necesario utilizar como operandos algunos datos que también están almacenados en la memoria.

La mayoría de registros en un computador digital, son registros de memoria, a los cuales se transfiere la información para almacenamiento y de los cuales se obtiene la información necesaria para el procesamiento. Cuando se lleva a cabo el procesamiento de datos, la información de los registros seleccionados en la unidad de memoria se transfiere primero a

los registros operacionales en la unidad procesadora.

Los resultados intermedios y finales que se obtienen en los registros operacionales se transfieren de nuevo a los registros de memoria seleccionados. De manera similar, la información binaria recibida de los elementos de entrada se almacena primero en los registros de memoria. La información transferida a los elementos de salida se toman de los registros en la unidad de memoria.

Existen varios registros que se utilizan para almacenamiento temporal de datos. Hay dos registros que resultan de particular interés. El registro de instrucción (IR: Instruction Register) contiene la instrucción que se está ejecutando, su salida se encuentra a la disposición de los circuitos de control, los cuales generan las señales de sincronización que controlan los verdaderos circuitos de procesamiento necesarios para ejecutar la instrucción. El contador de programa (PC: Program Counter) es un registro que rastrea la ejecución de un programa, contiene la dirección de memoria de la instrucción que en ese momento se está ejecutando. Durante la ejecución de esta instrucción el contenido del PC se actualiza para que corresponda a la dirección de la siguiente instrucción que deba ejecutarse.

Además, hay 2 registros que facilitan la comunicación con la memoria principal. Estos son el registro de dirección de memoria (MAR: Memory Address Register) y el registro de datos de memoria (MDR: Memory Data Register). El MAR especifica la

palabra de memoria seleccionada. A cada palabra en la memoria se le asigna un número de identificación comenzando desde cero hasta el número máximo de palabras disponibles. Para comunicarse con una palabra de memoria específica, su número de localización o dirección se transfiere al registro de direcciones. El MDR contiene los datos que deban escribirse o leerse en esa dirección.

Los circuitos internos de la unidad de memoria aceptan esta dirección del registro y abren los caminos necesarios para seleccionar la palabra buscada. Un registro de dirección con n bits puede especificar hasta 2^n palabras de memoria.

Las 2 señales de control aplicadas a la unidad de memoria se llaman de lectura y escritura. Una señal de escritura especifica una función de transferencia saliente. Cada una es referenciada por la unidad de memoria. Después de aceptar una de las señales, los circuitos de control interno dentro de la unidad de memoria suministran la función deseada. Cierta tipo de unidades de almacenamiento, debido a las características de sus componentes, destruyan la información almacenada en una celda cuando se lea el bit de ella.

La información transferida hacia adentro y fuera de los registro en la memoria y al equipo externo, se comunica a través de un registro común llamado (Buffer Register) registro separador de memoria. Cuando la unidad de memoria recibe una señal de control de escritura, el control interno interpreta el contenido del registro separador como la configuración de bits

de la palabra que se va a almacenar en un registro de memoria.

En la figura 1.3 se muestra la forma en que se puede realizar la conexión entre la memoria principal y el procesador. Con una señal de control de lectura, el control interno envía la palabra del registro de memoria al registro separador. En cada caso, el contenido del registro de direcciones especifica el registro de memoria particular referenciado para escritura y lectura.

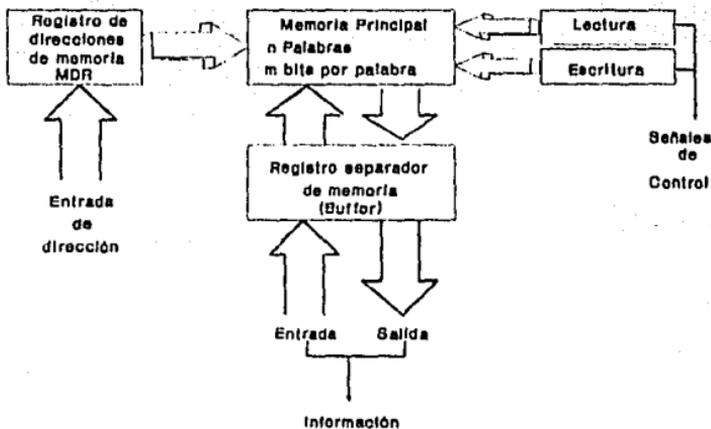


FIGURA 1.3

1.2 MICRO-OPERACIONES LÓGICAS.

Las micro-operaciones de transferencia entre registro no cambian el contenido de la información binaria, cuando esta pasa del registro fuente al registro de destino.

Todas las demás micro-operaciones cambian el contenido de la información durante la transferencia. Entre todas las operaciones posibles que pueden existir en un sistema digital, hay un conjunto básico del cual todas las demás operaciones puede obtenerse.

Las micro-operaciones lógicas especifican operaciones binarias para una cadena de bits almacenados en los registros. Estas operaciones consideran cada bit en los registros separadamente y lo tratan como una variable binaria.

La operación OR exclusiva se simboliza por medio de la siguiente proposición.

$$F \leftarrow A \oplus B$$

Esta especifica una operación lógica que considera cada par de bits en los registros como variables binarias. Las operaciones lógicas binarias pueden expresarse en términos de AND, OR y complemento. Se adoptarán símbolos especiales para estas 3 micro-operaciones, para distinguirlos de los símbolos correspondientes usados para expresar funciones de Boole. El símbolo \vee se usará para demostrar una micro-operación OR y el símbolo \wedge para demostrar un AND.

La micro-operación complemento es la misma que el complemento de 1 y usa una barra encima de la letra (o letras que denotan el registro).

Los símbolos para las 4 micro-operaciones lógicas se sumarizan en la Tabla 1.1.

Las micro-operaciones lógicas pueden configurarse fácilmente con un grupo de compuertas. El complemento de un registro de n bits se obtiene de n compuertas inversoras. La micro-operación AND se obtiene de un grupo de compuertas AND, cada una de las cuales recibe un par de bits de los 2 registros fuente. Las salidas de las compuertas AND se aplican a las entradas del registro de destino. La micro-operación OR requiere un grupo de compuertas OR dispuestas de manera similar.

Designación simbólica	Descripción
A ----- \bar{A}	Complementa todos los bits del registro A
F ----- $A \vee B$	Micro-operación OR lógica
F ----- $A \wedge B$	Micro-operación AND lógica
F ----- $A \oplus B$	Micro-operación OR exclusiva lógica

TABLA 1.1 MICROOPERACIONES LOGICAS

1.3 MICRO-OPERACIONES ARITMETICAS.

Las micro-operaciones aritméticas básicas son: Sumar, restar, complementar y desplazar. Todas las demás operaciones aritméticas pueden obtenerse de una variación o secuencia de estas micro-operaciones básicas. La micro-operación aritmética se define por la proposición:

$$F \leftarrow A + B$$

la cual significa una operación de suma. Esta establece que el contenido del registro A se va a sumar al contenido del registro B, y la suma se transfiere al registro F. Para configurar la proposición, se requiere de 3 registros A, B y F y la función digital que realiza la operación de suma, tal como un sumador paralelo. La sustracción aritmética implica la disponibilidad de un sustractor paralelo binario compuesto de circuitos sustractores completos conectados en cascada. La sustracción se configura a menudo por medio de la complementación y suma como se especifica a continuación:

$$F \leftarrow A + \bar{B} + 1$$

\bar{B} es el símbolo para el complemento de 1 en B. Al agregar 1 al complemento de 1, dará el complemento de 2 de B, se producirá A menos B.

Las micro-operaciones de incremento y decremento se simbolizan por una operación de más uno o menos uno ejecutados con los contenidos del registro. Estas micro-operaciones se configuran con un contador creciente o decreciente respectivamente.

Debe haber una relación directa entre las proposiciones escritas en un lenguaje de transferencia entre registros, y los registros y funciones digitales que se necesitan para su configuración. Para ilustrar lo anterior, considérese las 2 proposiciones siguientes:

$$T_2 : A \leftarrow A + B$$

$$T_3 : A \leftarrow A + 1$$

La variable de tiempo T_2 inicia una operación para sumar el contenido del registro B al contenido presente de A. La variable de tiempo T_5 incrementa el registro A. El incremento puede hacerse fácilmente con un contador, y la suma de dos números binarios puede generarse con un sumador en paralelo. La transferencia de la suma del sumador en paralelo al registro A, puede activarse con una entrada de carga al registro. Esto indica que el registro es un contador con capacidad de carga en paralelo. La configuración de las dos declaraciones se muestran en el diagrama de bloque en la figura 1.4. Un sumador paralelo recibe información de entrada de los registros A y B. Los bits suma del sumador paralelo se aplican a las entradas de A y la variable de tiempo T_2 carga la suma al registro A. La variable de tiempo T_5 incrementa el registro habilitando la entrada de incremento.

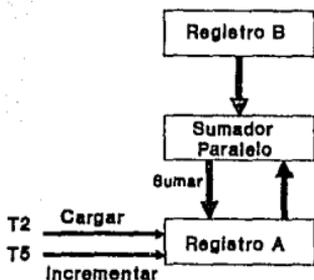


FIGURA 1.4 CONFIGURACION PARA LAS MICROOPERACIONES DE SUMA E INCREMENTO

La operación de multiplicación puede ser representada por el símbolo $*$ y la división por un $/$. Estas dos operaciones son operaciones aritméticas pero no se incluye en el conjunto básico de micro-operaciones, mostradas en la tabla 1.2. El único lugar donde estas operaciones pueden considerarse como micro-operaciones es en un sistema digital, en donde se configuran por medio de los circuitos combinacionales. En tal caso, las señales que ejecutan estas operaciones se propagan a través de las compuertas.

Designación simbólica	Descripción
F --- A + B	Contenido de A más B se transfiere a F
F --- A - B	Contenido de A menos B se transfiere a F
B --- B	Se complementa el registro B (complemento de 1)
B --- B + 1	Forman el complemento de 2 del contenido del registro B.
F --- A+B+1	A más el complemento de 2 de B se transfiere a F.
A --- A + 1	Incrementar el contenido de A en 1 (cuenta creciente)
A --- A - 1	Decrementar el contenido de A en 1 (cuenta decreciente).

TABLA 1.2 MICROOPERACIONES ARITMETICAS

1.4 MICRO-OPERACIONES DE CORRIMIENTO.

Las micro-operaciones de desplazamiento transfieren la información binaria entre registros en los computadores en serie. Se usan también en computadoras en paralelo para operaciones aritméticas, lógicas y de control.

Los registros pueden transferirse a la izquierda o a la derecha. No hay símbolos convencionales para las operaciones de desplazamiento. La tabla 1.3 muestra los símbolos para las operaciones de desplazamiento a la izquierda y a la derecha. Por ejemplo:

$$A \longleftarrow \text{Shl } A, \quad B \longleftarrow \text{Shr } B,$$

Son dos micro-operaciones que especifican un desplazamiento de 1 bit a la izquierda del registro A y 1 bit a la derecha del registro B. El símbolo del registro debe ser el mismo en ambos lados de la flecha como una operación de incremento.

Mientras los bits de un registro se desplazan, los flip-flops extremos reciben información de la entrada en serie. El flip-flop extremo está en la posición de extrema izquierda del registro durante una operación de desplazamiento a la derecha y a la posición de extrema izquierda durante una operación de desplazamiento a la izquierda.

Designación simbólica	Descripción
A ←--- Shl A	Registro A de desplazamiento a la izquierda
F ←--- Shl A	Registro A de desplazamiento a la derecha

TABLA 1.3 MICROOPERACIONES DE DESPLAZAMIENTO

1.5 GENERACIONES DE FUNCIONES DE CONTROL.

La operación adecuada de un micro-procesador requiere que se presenten ciertas señales de control y tiempo para

lograr funciones específicas y que otras señales de control sean medidas para determinar el estado del Microprocesador. Un conjunto típico de líneas de control disponibles en la mayoría de los Microprocesadores se muestra en la figura 1.5. Para completar el diagrama se muestra también el bus de datos, el bus de direcciones y el terminal de entrada de la fuente de poder a la unidad. Las necesidades de potencia de un Microprocesador particular se especifican por el nivel de voltaje y consumo de poder que debe suministrarse para operar el circuito integrado.

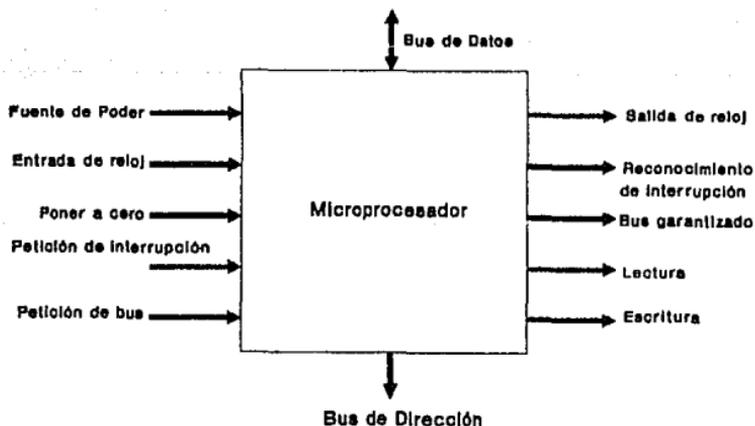


FIGURA 1. 5

El terminal de entrada de reloj es usado por el microcomputador para generar pulsos de reloj de multifase, y producir secuencias de tiempo y control para las funciones internas. Algunos Microprocesadores requieren un generador externo de pulsos de reloj para producir los pulsos. En este caso, el reloj de salida lo produce el generador de reloj en vez del Microprocesador en sí. Algunas unidades generan el pulso de reloj dentro de sí, pero requieren un cristal externo ó circuito para controlar la frecuencia del reloj. Los pulsos de reloj son usados por los módulos externos para sincronizar sus operaciones con las operaciones de Microprocesador.

El terminal de puesta a cero ó RESET, se usa para reposicionar o iniciar el Microprocesador después de haber activado la potencia ó en cualquier momento en que el usuario quiera comenzar el proceso. El efecto de la señal de puesta a cero es iniciar el Microprocesador, forzando una dirección dada al contador de programa. El programa comienza la ejecución con la primera instrucción en esa dirección.

La manera más simple de iniciar una puesta a cero es borrar el contador del programa y comenzar el programa desde la dirección cero. Algunos Microprocesadores responden a la señal de puesta a cero transfiriendo el contenido de un lugar de la memoria específico al contador del programa.

La requisición de interrupción (INTERRUPT) al Microprocesador, viene típicamente de un módulo de interconexión para informar al Microprocesador que esté listo para transferir

la información, cuando el Microprocesador recibe una requisición de interrupción; suspende la ejecución del programa corriente y se bifurca a un programa que sirve de módulo de interconexión. Al completar la rutina de servicio, el computador regresa al programa previo. La facilidad de interrupción, se incluye para producir un cambio en la secuencia del programa como resultado de las condiciones externas.

El terminal de entrada de BUS-REQUEST (requisición del bus), es una requisición al Microprocesador para suspender su operación y llevar todos los buses a su estado de mayor impedancia. Una vez reconocida la requisición, el Microprocesador responde habilitando la línea de salida de control de garantía de bus (BUS-GUARANTEED). Así, cuando un dispositivo externo desea transferir la información directamente a la memoria, este solicita que el Microprocesador abandone el control del bus común. Una vez que el bus sea inhabilitado por el Microprocesador, el dispositivo que originó la requisición toma control sobre el bus de direcciones y datos para producir las transferencias de memoria sin la intervención del procesador, esta característica se llama acceso directo de memoria (DMA-Direct Memory Access).

Lectura y escritura son líneas de control que informan el componente seleccionado por el bus de direcciones de la dirección de la transferencia esperada en el bus de datos. La línea de lectura interna a la unidad seleccionada que el bus de

datos está en modo de entrada y que el procesador aceptará datos del mismo.

La línea de escritura indica que el procesador está en el modo de salida y que los datos válidos están disponibles en el bus de datos. Cuando los buses están inhabilitados, las dos líneas de control estarán en el estado de alta impedancia; así la unidad externa que controla los buses puede especificar las operaciones de lectura y escritura. Existen otras posibilidades para el control de los buses. El bus de direcciones puede ser controlado con una línea adicional para indicar si la dirección es para una palabra de memoria o para una unidad de interconexión. Otra posibilidad es combinar las líneas de control de lectura y escritura en una línea que se denomina R/W. Cuando esta línea es "1" indica lectura y cuando es escritura es "0". Una segunda línea de control es necesaria para indicar cuando una dirección válida está en el bus de direcciones de manera que los componentes externos respondan a la línea R/W solamente cuando se solicita con una dirección válida.

1.6 REPRESENTACION DE DATOS ARITMETICOS (MAGNITUD Y SIGNO, OVERFLOW, DESPLAZAMIENTOS, PUNTO FLOTANTE)

Un registro con n flip-flops puede almacenar un número binario de n bits; cada flip-flop representa un dígito binario. Este representa la magnitud del mismo pero no da información acerca de su signo o la posición del punto binario.

El signo se necesita para operaciones aritméticas, ya que representa cuándo el número es positivo o negativo. La posición del punto decimal es necesario para representar enteros, fracciones o números enteros y fraccionarios mezclados.

El signo de un número es una cantidad discreta de información que tiene dos valores: más o menos. Estos dos valores pueden ser representados por un código de un bit. La convención es representar un más con un "0" y un menos con un "1". Para representar un número binario con signo, se necesitan $n=k+1$ flip-flops; k de ellos para la magnitud y uno para almacenar el signo del número.

Cuando un número binario de punto fijo es positivo, el signo se representa como "0" y la magnitud por un número binario positivo. Cuando el número es negativo, el signo se representa por "1" y el resto del número puede ser representado por cualquiera de las 3 maneras siguientes:

1. Signo-magnitud
2. Signo-complemento de 1
3. Signo-complemento de 2

SOBRECARGA (OVERFLOW). - Cuando dos números con n dígitos cada uno, se suman y la suma ocupa $n + 1$ dígitos, se dice que hay un desbordamiento por sobrecapacidad (overflow). Esto es verdadero para los números binarios o números decimales con o sin signo. Cuando se hace una suma con lápiz y papel, una sobrecapacidad no es un problema, ya que no hay limitaciones por el ancho de la página para escribir la suma.

Una sobrecapacidad es un problema en un computador digital, ya que las longitudes de todos los registros, incluyendo todos los registros de memoria son de longitud finita. Un resultado de $n + 1$ bits no puede acomodarse en un registro de longitud normalizada n . Por esta razón, muchos computadores comprueban la ocurrencia de la sobrecapacidad y cuando esto ocurre, ponen a "1" el flip-flop de sobrecapacidad para que el usuario verifique.

DESPLAZAMIENTOS ARITMETICOS. - Un desplazamiento aritmético, es una micro-operación que mueve un número binario con signo a la izquierda o a la derecha. Un movimiento aritmético a la izquierda multiplica un número con signo por 2. Un número aritmético a la izquierda divide el número por 2. Los desplazamientos aritméticos deben dejar el signo sin cambio, ya que el signo del número permanece igual cuando se multiplica o divide por 2.

PUNTO FLOTANTE. - La representación del punto flotante de los números necesita de registros. El primero representa un número con signo de punto fijo y el segundo la posición del punto del radical.

II. DISEÑO LÓGICO DE UN PROCESADOR

II.1 ORGANIZACIÓN DEL PROCESADOR

Las instrucciones que constituyen un programa que debe ejecutar una computadora, se cargan en localizaciones secuenciales de su memoria principal. Para ejecutar este programa la CPU trae una instrucción por vez y realiza las funciones especificadas. Las instrucciones se traen por medio de localizaciones efectivas de la memoria, hasta la ejecución de una ramificación o de una instrucción de salto.

La parte procesadora de un computador (CPU), se trata algunas veces como el canal de datos del mismo, porque el procesador formula los canales de transferencia de datos entre los registros de la unidad, la CPU lleva un control de la dirección de la localización de la memoria de la siguiente instrucción por medio del empleo de un registro de la CPU especial al que se denomina contador de programa (PC). Después de traer una instrucción, el contenido del PC se actualiza para apuntar la siguiente instrucción de la secuencia.

En una unidad procesadora bien organizada, los canales de datos se forman por medio de buses y otras líneas comunes. Las compuertas de control que forman los canales de datos, son esencialmente multiplexores y decodificadores, cuyas líneas de selección especifican el camino requerido.

Una organización se muestra en la figura 2.1. En este caso, la unidad aritmética y lógica (ALU) y todos los registros del CPU están conectados a través de un bus común. Desde luego, este bus es interno y no debe confundirse con el bus externo, o buses, que conectan a la CPU con la memoria y los dispositivos de E/S. El bus externo de memoria se muestra en la figura 2.1 conectado con la CPU por medio de los registros de datos y de dirección MDR y MAR. El número y funciones de los registros R0 al R(n-1) varían mucho de una máquina a otra. Puede ser para que el programador los emplee en operaciones generales, pero algunos de ellos pueden ser registros de aplicación especial, tales como registros índice o apuntadores de pila.

Los registros Y y Z son invisibles para el programador; esto es, no necesita preocuparse de su existencia, ya que nunca son referenciados en forma directa por instrucción alguna. Solo son utilizados por la CPU para almacenamiento temporal durante la ejecución de ciertas instrucciones. Sin embargo, nunca se utilizan para almacenar datos generados por una instrucción, para que luego los utilice otra instrucción.

En una Memoria de Acceso Aleatorio, la información está almacenada en localizaciones identificadas por sus direcciones. Para traer una palabra de información de la memoria, la CPU tiene que especificar la dirección de la localización de la memoria en donde está información esta almacenada y solicitar una operación de leer. Para realizar la traida desde la memoria, la CPU transfiere la dirección de la palabra de

información requerida al registro de dirección en la memoria (MAR: Memory Address Register), éste, está conectado a las líneas de dirección de las de memoria. Por lo tanto, la dirección de la palabra requerida se transfiere a la memoria principal. Mientras tanto, la CPU utiliza las líneas de control

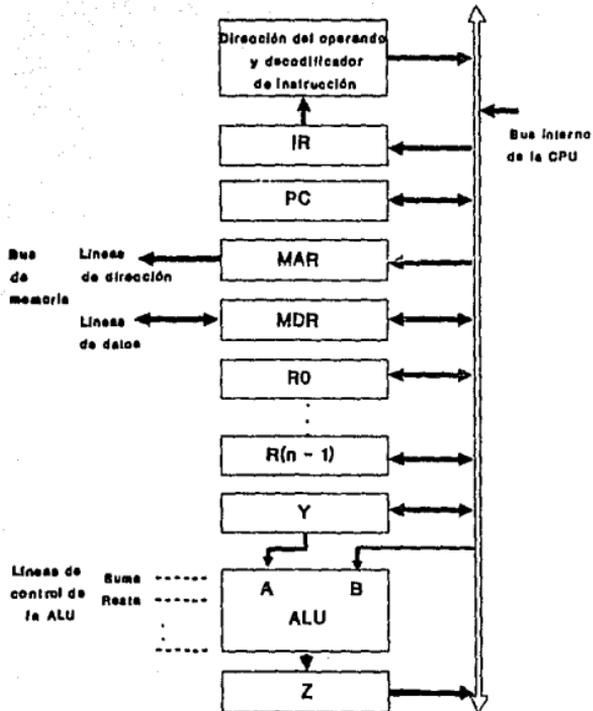


FIGURA 2.1 ORGANIZACIÓN DE BUS ÚNICO PARA LAS RUTAS DE DATOS DENTRO DE LA CPU.

del bus de memoria para indicar que se requiere una operación de leer. Lo normal es que después de emitir esta solicitud, la CPU espere hasta que reciba una respuesta de la memoria, que le informe que la función solicitada se ha concluido. Esto se logra por medio de su otra señal de control del bus de memoria, a la cual se denominará Función de Memoria Completa (MFC). La memoria pone esta señal en "1" para indicar que el contenido de la localización especificada de la memoria. Se considera que en cuanto la señal MFC sea igual a "1", la información que está en las líneas de datos está cargada en el MDR y por lo tanto está disponible para su uso dentro de la CPU.

El procedimiento para escribir una palabra en una localización dada de la memoria es semejante al empleador para leer la memoria. La única excepción, es que la palabra de datos que se va a escribir se carga en el MDR antes de que se emita el comando escribir.

II.2 UNIDAD ARITMETICA Y LOGICA (ALU)

Como se recordara en el capítulo anterior, en la parte de conceptos, se analizó y estudio el funcionamiento de un ALU. Se mencionó que un ALU es una función multioperación digital de lógica combinacional.

Al realizar una operación aritmética o lógica, debe recordarse que la ALU es en sí mismo un circuito combinatorio que no tiene almacenamiento interno. Por lo tanto, para realizar un adición, por ejemplo, los dos números que se van a

sumar deben estar listos en las dos entradas de la ALU simultáneamente. Se proporciona con este fin, el registro Y de la figura. 2.1 que se utiliza para conectar uno de los dos números, mientras que el otro es manejado por compuerta hacia el bus. El resultado se almacena en forma temporal en el registro Z. En la figura. 2.2 se muestra el diagrama lógico de una unidad lógica aritmética (ALU). Todas las operaciones aritméticas requieren dos números y producen un resultado. La multiplicación, por ejemplo, utiliza un multiplicando y un multiplicador para obtener un producto. Aunque todas las ALU deben ser capaces de dar datos y el resultado, en los distintos modelos se emplean diferentes técnicas de proceso y almacenamiento.

Además de las funciones aritméticas, la ALU se encarga de las operaciones lógicas. Estas operaciones son generalmente comparadores. Los circuitos de la unidad se emplean casi siempre para comparar dos números mediante una resta. El signo (positivo o negativo) y el valor de la diferencia indican al procesador que el número es mayor o menor que el primero, o igual a él.

Es frecuente que tanto la unidad de control, como la aritmética sean mucho más veloces en tiempo básico de ciclo que los otros dispositivos conectados a un sistema de cómputo. Por esto, resulta posible diseñar sistemas de cómputo relativamente complejos que contengan varios dispositivos externos controlados por un sólo procesador. Estos dispositivos pueden

ser terminales CRT, memorias en cinta y disco magnético, sensores, exhibidores, controladores mecánicos, etc.

Por lo general, una operación de la ALU es más veloz que una operación de acceso a la memoria. Esto significa que en una instrucción en que haya una operación de la ALU, el operando que deba traerse de la memoria no requerirá mucho más tiempo de ejecución que una instrucción que sólo translada el contenido de una localización de la memoria a otra.

En comparación, las operaciones de multiplicación y división son más complejas que las de adición o sustracción. Por lo común, estas operaciones se incluyen en el conjunto básico de instrucciones. Sin embargo, sus tiempos de ejecución pueden ser significativamente más lentos que los de otras instrucciones tales como sumar, mover, etc. Esto se debe a que están implantadas como una secuencia de pasos de adición y sustracción que realiza la ALU, controlada por un microprograma.

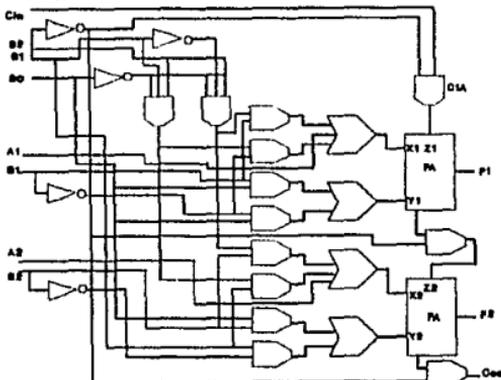


FIGURA 2.2 DIAGRAMA LOGICO DE UNA ALU

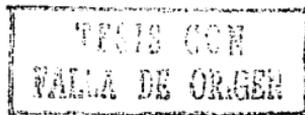
11.3 REGISTROS.

En el capítulo 1 se estudió y expuso el tema de los registros, sin embargo en este punto se analizará los registros acumulador, de condición y de desplazamiento.

Algunas unidades procesadoras separan un registro de otro y se le llama registro acumulador, abreviado AC o registro A. El nombre de este registro se deriva del proceso de adición aritmética que se encuentra en los computadores digitales. El proceso de sumar muchos números, se lleva a cabo almacenando inicialmente esos mismos en otros registros procesadores, o en la unidad de memoria del computador y borrando el acumulador a "0". Los números se agregan al acumulador uno a uno en orden consecutivo. El segundo número se agrega a los contenidos del acumulador y la suma formada de los números se agregan y se forma la suma total. Así, el registro acumula la suma paso a paso haciendo sumas secuenciales entre un número nuevo y la suma acumulada previamente.

El registro acumulador, en una unidad de proceso es un registro multi-propósito capaz de realizar no solamente la micro-operación de suma sino otras micro-operaciones de la misma forma. Las compuertas asociadas con un registro acumulador suministran todas las funciones digitales encontradas en un ALU.

La figura 2.3 muestra el diagrama de bloque de una unidad procesadora que emplea un registro acumulador.



En algunos casos, toda la unidad procesadora es justamente el registro acumulador y el ALU asociado. El registro en sí puede funcionar como un registro de desplazamiento para suministrar las micro-operaciones de desplazamiento. La entrada B suministra la otra fuente de información al ALU por el terminal A. El resultado de una operación se transfiere de nuevo al registro A y se reemplaza su contenido previo.

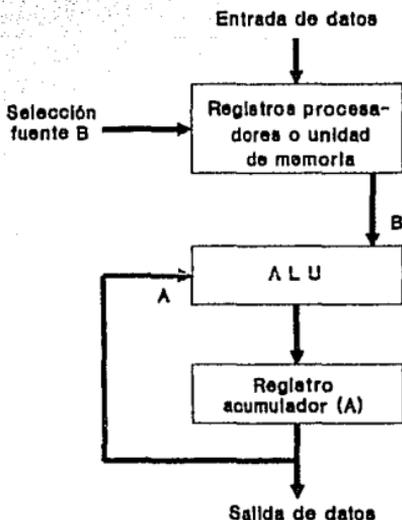


FIGURA 2.3 PROCESADOR CON UN REGISTRO ACUMULADOR

REGISTROS DE CONDICION.—Las magnitudes relativas de dos números pueden ser determinadas restando un número de otro y luego combinando ciertas condiciones de los bits en la diferencia resultante. Si los dos números están sin signo, las condiciones de los bits de algún interés, son el arrastre de salida y un resultado posible de cero. Si los dos números incluyen un bit de signo en la posición de mayor orden, las condiciones principales de los bits, son el signo del resultado, una indicación de cero y una condición de sobrecapacidad. Es conveniente algunas veces suplementar el ALU con un registro de condición donde se almacenan aquellas condiciones de los bits para análisis posterior.

La figura. 2.4 muestra un diagrama de bloque de un ALU de 8 bits con un registro de condición de 4 bits. Los cuatro bits de condición se simbolizan por medio C, S, Z y V. Los bits se ponen a uno ó cero como resultado de una operación realizada en el ALU.

1. El bit C se pone a uno si el arrastre de salida del ALU es "1" y se pone a cero (borrado) si el arrastre de salida es "0".
2. El bit S se pone a uno si el bit de mayor orden del resultado en la salida del ALU (bit de signo) es "1" y se pone a cero (borrado) si el bit de mayor orden es "0".
3. El bit Z se pone a uno si la salida del ALU contiene sólo ceros y se pone a cero (borrado) de otra manera.

$Z=1$ si el resultado es cero y $Z=0$ si el resultado es diferente de cero.

4. El bit V se pone a uno si la OR-exclusiva de los arrastres C_8 y C_9 es 1 y de otra manera se pone a cero (borrado). Esta es la condición de sobrecapacidad cuando los números están en la representación de signo-complemento de 2. Para el ALU de 8 bits, V se pone a uno si el resultado es mayor que 127 o mayor que -128 .

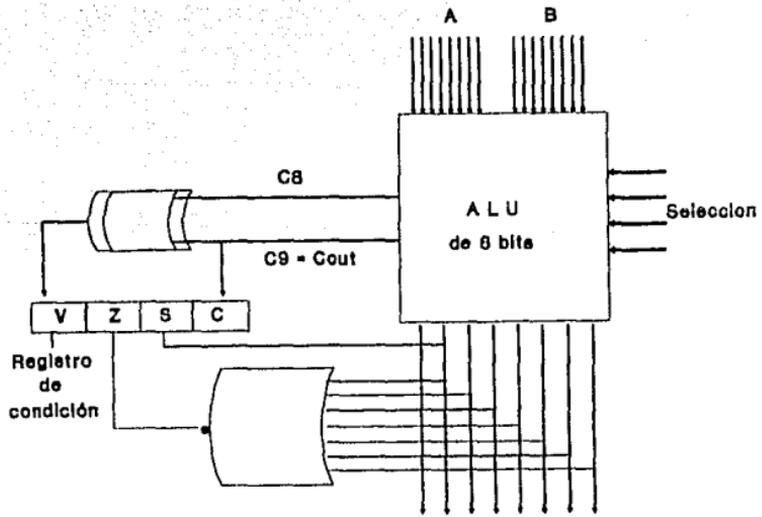


FIGURA 2.4 ACTIVACION DE LOS BITS EN UN REGISTRO DE ESTADO

REGISTRO DE DESPLAZAMIENTO.- La unidad de desplazamiento adjunta transfiere la salida del ALU al bus de salida. La unidad de desplazamiento puede transferir la información directamente sin un desplazamiento, o puede desplazar la información a la derecha o a la izquierda. Se debe tener alguna precaución para que algunas veces no haya transferencia del ALU al bus de salida. El registro de desplazamiento produce la micro-operación del desplazamiento comúnmente no disponible en un ALU.

Un circuito para un registro de desplazamiento es un registro de desplazamiento bidireccional con carga en paralelo. La información del ALU puede ser transferida al registro en paralelo, para luego desplazar a la derecha o a la izquierda. La transferencia de un registro fuente a un registro de destino puede hacerse con un pulso de reloj si se configura el registro de desplazamiento de lógica combinatorial, las señales del ALU al bus de salida se propagarán por las compuertas sin la necesidad de un pulso de reloj.

En la figura 2.5 se muestra un registro de desplazamiento. Las 2 variables de selección H_1 y H_0 aplicadas a los cuatro multiplexores seleccionan el tipo de operación en el registro de desplazamiento. Con $H_1, H_0 = 00$ no se ejecutan desplazamientos y las señales de F van directamente a las líneas de S . Las dos siguientes variables de selección causan una operación de desplazamiento a la derecha o a la izquierda.

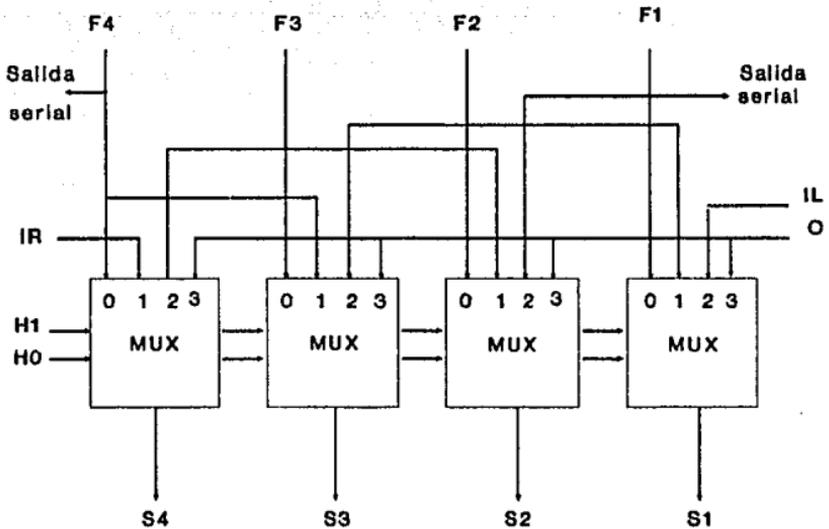


FIGURA 2.5 REGISTRO DE DESPLAZAMIENTO DE 4 BITS A BASE DE LOGICA COMBINACIONAL

II.4 BUSES (CANALES)

Las necesidades de comunicación en un sistema microcomputador aparecen a tres niveles distintos:

1) Comunicación entre los elementos constituyentes de un circuito integrado. Este sistema de comunicación constituye los buses internos, de estructura más o menos distinta para cada CPU considerada.

2) Comunicación entre los distintos subsistemas del microcomputador; Unidad Central de Proceso, y entrada y salida. El conjunto de las vías para este tipo de comunicación forman los buses externos, en general de carácter síncrono.

3) Comunicación con periféricos: Esta comunicación es a nivel de sistema Entrada/Salida con periférico. El conjunto de líneas de transmisión forman los buses periféricos, en general de carácter asíncrono.

La información desde o hacia la CPU y otros subsistemas puede ser de tres tipos: Direcciones, datos y control; por lo que funcionalmente se encontrarán tres tipos de buses en el microcomputador, aunque no estén físicamente siempre separados, ya que pueden estar multiplexados en el tiempo sobre líneas comunes. La interconexión de la CPU con el sistema de memoria y E/S se realiza mediante estos buses, tal como se ilustra en la figura. 2.6.

1) Bus de direcciones: Son líneas de la CPU a memoria y sistema de Entrada/Salidas, que sirven para enviar la dirección del elemento seleccionado para una transferencia.

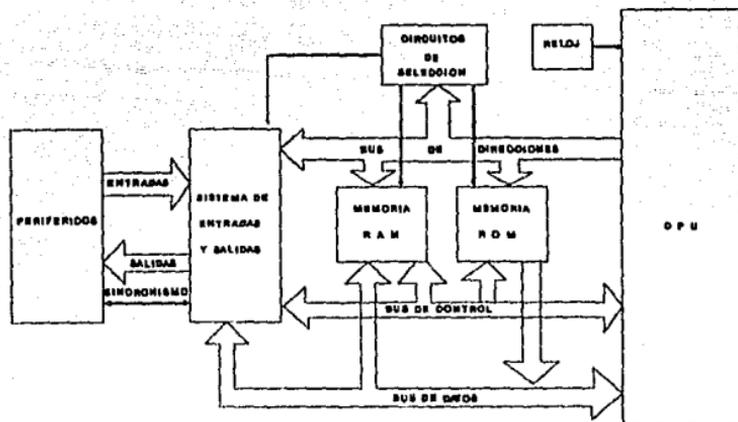


FIGURA 2.6 INTERCONEXION DE LOS SUBSISTEMAS DEL MICROCOMPUTADOR POR MEDIO DE BUSES COMPARTIDOS

Estas líneas serán siempre gobernadas por la CPU, salvo en el caso de transferencias por Acceso Directo a Memoria. Pueden compartir total o parcialmente las líneas físicas con el bus de datos.

2) *Bus de datos:* Son líneas bidireccionales (o dos buses unidireccionales independientes) para el envío de información (instrucciones y datos) entre subsistemas. Su anchura, en general, es la longitud de palabra del Microprocesador.

3) *Bus de control:* Son líneas de entrada o salida de la CPU de órdenes o de información de estado, que permiten coordinar la operación de todo el sistema.

El bus de control presenta variaciones notables de un Microprocesador a otro. Existen dos tendencias en cuanto a su realización física:

a) Dar más señales de control elementales, que pueden estar codificadas, y que deben ser interpretadas temporalmente respecto al estado en que se producen en cada ciclo de máquina. Esta estructura se observa principalmente en Microprocesadores de las primeras generaciones.

b) Dar un conjunto de señales completo y de multiplexado a las de memorias y periféricos y para que respondan de una forma pasiva, por lo que no son precisos circuitos especializados para la decodificación de las señales de control.

III. DISEÑO DE LA LÓGICA DE CONTROL

Los arquitectos de computadores, han buscado con insistencia nuevos enfoques para el diseño de máquinas de alto rendimiento. El flujo de datos (data flow) y la VLSI (Very Large Scale Integration) ofrecen dos planteamientos mutuamente compatibles, para el diseño de los futuros Supercomputadores.

Los conceptos de computación de flujo de control y de flujo de datos, se distinguen por el control de las secuencias de computación, que da lugar a dos representaciones de programa diferentes.

La relación entre la unidad de control y el procesador de datos en un sistema digital, se muestra en la figura 3.1. La parte del procesador de propósito general, puede consistir de registros individuales y funciones digitales asociadas.

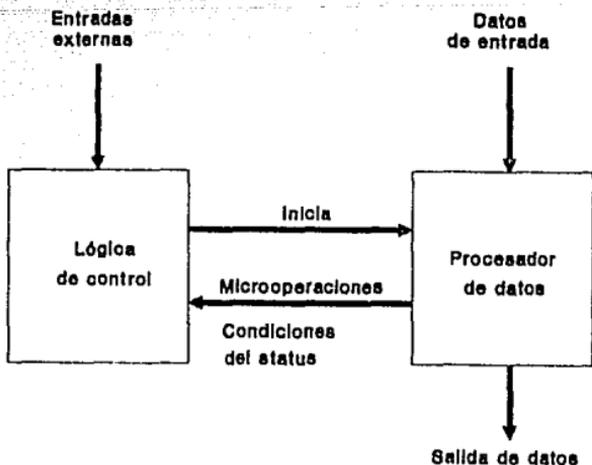


FIGURA 3.1 INTERACCIÓN ENTRE EL CONTROL Y EL PROCESADOR DE DATOS

III.1 ORGANIZACION DEL CONTROL

Los computadores de flujo de control, tienen una organización guiada por control. Esto significa, que el programa tiene control completo sobre el secuenciamiento de las instrucciones. En los computadores de flujo de control, se realizan las secuencias síncronas utilizando control centralizado.

El diseño de un sistema digital que requiere una secuencia de control, comienza con la suposición de la disponibilidad de variables de tiempo. Se diseña cada variable en la secuencia por medio de un estado, y luego se forma un diagrama de estado o una representación equivalente para la transición entre estados.

En este capítulo, se presentan tres configuraciones posibles para una unidad de control. Las diferentes configuraciones se presentan en forma de diagrama de bloque, para darle énfasis a las diferencias en organización.

A continuación, identificaremos las características especiales del modelo de flujo de control:

- Los datos se pasan entre las instrucciones a través de referencias a celdas de memoria compartida.
- El flujo de control es implícitamente secuencial, pero pueden utilizarse explícitamente operadores de control especiales para el paralelismo.

- En un entorno de control centralizado, se utilizan computadores de programa para secuenciar la ejecución de las instrucciones.

III.2 CONTROL ALAMBRADO (HARD-WIRED)

Este método resulta de usar circuitos SSI (Small Scale Integration) y MSI (Medium Scale Integration) para la configuración. Los diferentes circuitos se interconectan con alambres para formar una red de control. Una unidad de control configurada con elementos SSI y MSI se denota como un control de base de materiales interconectados. Si se necesitan alteraciones o modificaciones, los circuitos se deben alambrear de nuevo para cumplir con las especificaciones.

El diseño se lleva a cabo en cinco pasos consecutivos:

1. Se enuncia el problema.
2. Se asume una configuración inicial del equipo.
3. Se formula el algoritmo.
4. Se especifica la parte del procesador de datos.
5. Se diseña la lógica de control.

ENUNCIADO DEL PROBLEMA.- El problema es configurar con materiales la adición y la sustracción de dos números binarios, de punto fijo representados en forma de signo-magnitud.

CONFIGURACION DEL EQUIPO.- Los dos números binarios con signo al ser sumados o restados contienen n bits.

Las magnitudes de los números contienen $k = n - 1$ y se almacenan en los registros A y B. Los bits de signo se almacenan en los flip-flops A_s y B_s . La figura 3.2 muestra los registros y el equipo asociado.

El ALU realiza las operaciones aritméticas y el registro E de 1 bit sirve como flip-flop de sobrecapacidad. El arrastre de salida del ALU se transfiere al E.

Se asume que los dos números y sus signos, han sido transferidos a sus registros respectivos y que el resultado de la operación está disponible en los registros A y B.

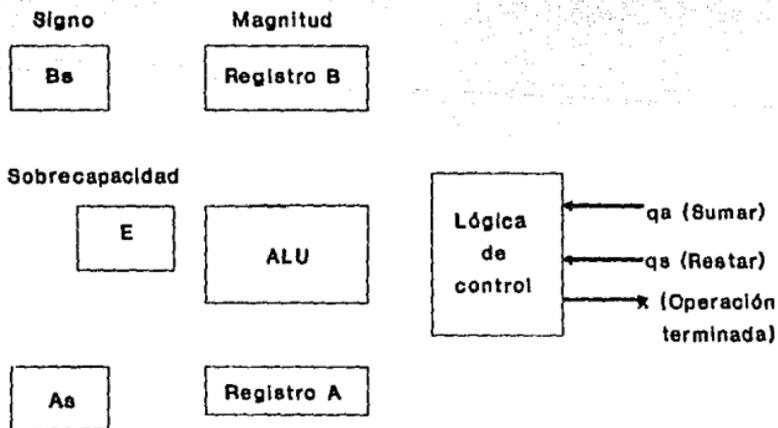


FIGURA 3.2 CONFIGURACION DEL REGISTRO PARA EL SUMADOR-SUSTRACTOR

Las señales de entrada en el control, especifican las operaciones de suma (q_+) y resta (q_-). La variable de salida X indica el final de la operación.

DEDUCCION DEL ALGORITMO.— La representación de números por medio de signo-magnitud es familiar, debido a que se usa para los cálculos aritméticos a lápiz y papel.

Primeramente, se designa la magnitud de dos números A y B . Cuando los números se suman o restan algebraicamente se encuentra que hay ocho condiciones diferentes para considerar, dependiendo del signo de los números y de la operación realizada. Las ocho condiciones pueden expresarse en forma compacta de la siguiente manera:

$$(+ - A) + - (+ - B)$$

El flujograma de la figura 3.3 muestra cómo se puede configurar una sustracción y una adición con signo-magnitud, con el equipo de la figura 3.2. Se inicia una operación con la entrada q_+ o la entrada q_- . La entrada q_+ inicia una operación de sustracción, de manera que se complementa el signo de B . La entrada q_- inicia una operación de suma y el signo de B se deja sin cambiar. El siguiente paso es comparar los dos signos. El bloque de decisión demarcado con $A_s:B_s$, simboliza esta decisión. Si los signos son iguales, se sigue por el camino demarcado por el símbolo $=$; de otra manera se tomará el camino

marcado por el símbolo f . El contenido de A se suma al contenido de B y la suma se transfiere a A en el caso de símbolos iguales. El valor del arrastre final en este caso, es una sobrecapacidad de manera que se hace el flip-flop E igual al arrastre de salida $Cout$. El circuito irá a su estado inicial y la salida X se convierte en 1.

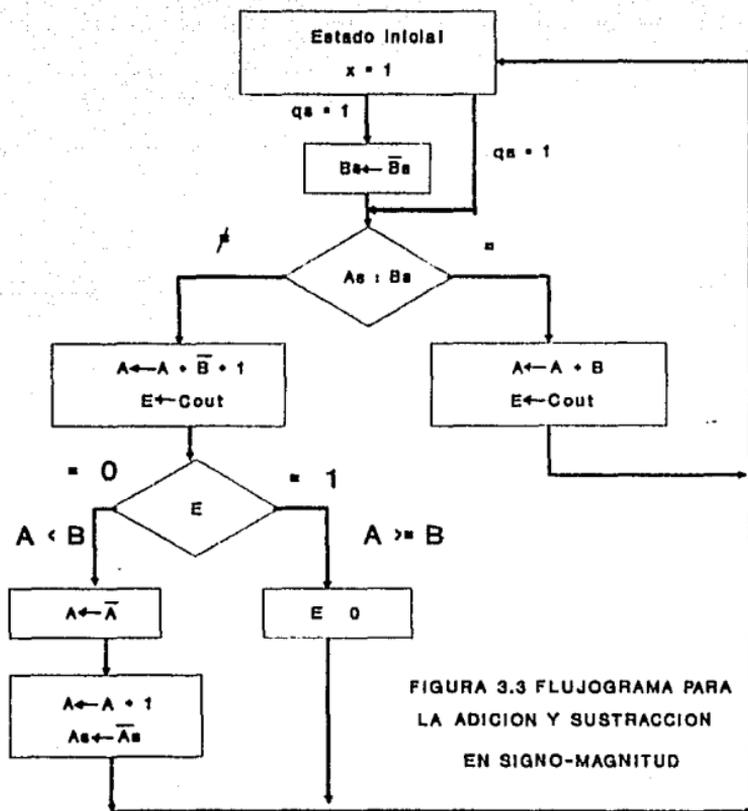


FIGURA 3.3 FLUJOGRAMA PARA LA ADICION Y SUSTRACCION EN SIGNO-MAGNITUD

ESPECIFICACION DEL PROCESADOR DE DATOS.-- El flujoograma del algoritmo, lista todas las micro-operaciones para la parte del procesador de datos del sistema. Las operaciones A y B pueden hacerse con el ALU. La figura 3.4 muestra el procesador de datos con las variables de control requeridas.

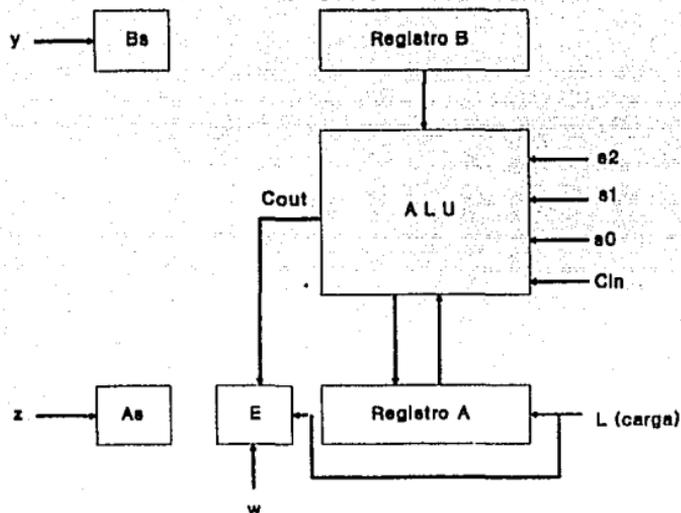
La variable L, carga la salida del ALU al registro A y también el arrastre de salida E. Las variables Y, Z y W complementan Bs, As y borran E respectivamente.

El diagrama de bloque de la lógica de control, se muestra en la figura 3.4 (b). El control recibe cinco entradas: Dos de los componentes externos y tres del procesador de datos. Para simplificar el diseño se define una nueva variable S:

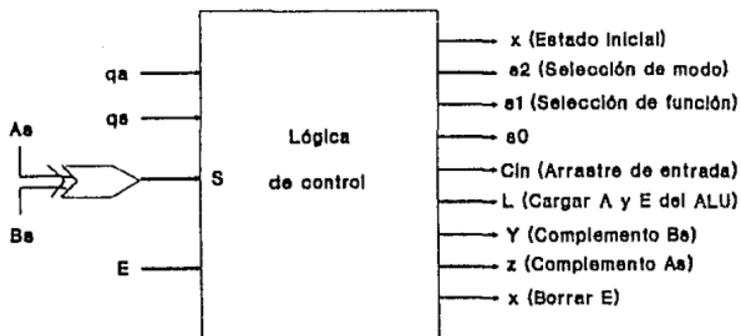
$$S = A_s + B_s$$

Esta variable da el resultado de la comparación entre dos bits de signo. La operación OR-exclusiva, es igual a 1 si los dos signos no son iguales, y es igual a 0 si los signos son ambos positivos o negativos.

El control suministra una salida X para el circuito externo. Esta selecciona también las operaciones en el ALU por medio de las cuatro variables de selección S_2 , S_1 , S_0 y Cin . Las otras cuatro salidas van a los registros en el procesador de datos como se especifica en el diagrama.



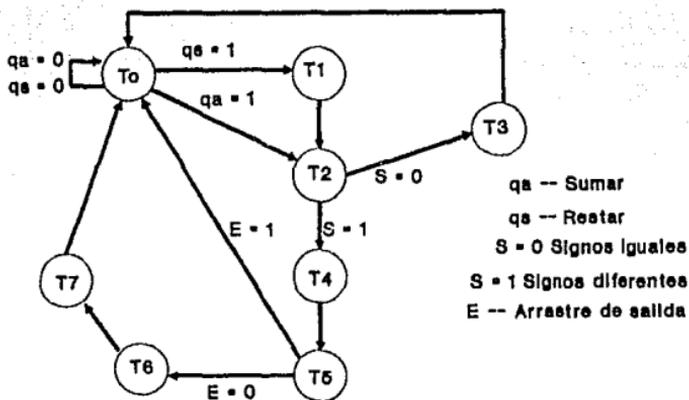
(a) REGISTROS PROCESADORES DE DATOS Y ALU



(b) DIAGRAMA DE BLOQUE DEL CONTROL

FIGURA 3.4 DIAGRAMA DE BLOQUE DE SISTEMA

DISEÑO DE LA LOGICA DE CONTROL. - Las salidas de control son función de los estados de control y se listan en la figura 3.5(b). Estas salidas se definen en el diagrama de bloque de la figura 3.4 (b). Los valores para las variables de selección se determinan a partir de la tabla 3.1.



(a) DIAGRAMA DE ESTADO

	X	S2	S1	S0	Cin	L	Y	Z	W
To: Estado Inicial $X = 1$	1	0	0	0	0	0	0	0	0
T1: $B_s \leftarrow \bar{B}_s$	0	0	0	0	0	0	1	0	0
T2: nada	0	0	0	0	0	0	0	0	0
T3: $A \leftarrow A + B$, $E \leftarrow \text{Cout}$	0	0	0	1	0	1	0	0	0
T4: $A \leftarrow A + \bar{B} + 1$, $E \leftarrow \text{Cout}$	0	0	1	0	1	1	0	0	0
T5: $E \leftarrow 0$	0	0	0	0	0	0	0	0	1
T6: $A \leftarrow \bar{A}$	0	1	1	1	0	1	0	0	0
T7: $A \leftarrow A + 1$, $A_s \leftarrow \bar{A}_s$	0	0	0	0	1	1	0	1	0

(b) SECUENCIA DE TRANSFERENCIA DEL REGISTRO
 FIGURA 3.5 DIAGRAMA DE ESTADO DE CONTROL
 Y SECUENCIA DE MICROOPERACIONES.

La variable L (cargar A), debe ser igual a "1" cada vez que la salida del ALU se transfiera al registro A. De otra manera L es "0" y las salidas del ALU no tendrían efecto sobre el registro.

Selección				Salida	Función
S ₂	S ₁	S ₀	C _{in}		
0	0	0	0	F = A	Transferir A
0	0	0	1	F = A+1	Incrementar A
0	0	1	0	F = A+B	Suma
0	0	1	1	F = A+B+1	Suma con arrastre
0	1	0	0	F = A-B-1	Resta con préstamo
0	1	0	1	F = A-B	Sustracción
0	1	1	0	F = A-1	Decrementar A
0	1	1	1	F = A	Transferir A
1	0	0	X	F = A \vee B	OR
1	0	1	X	F = A \oplus B	OR-exclusiva
1	1	0	X	F = A \wedge B	AND
1	1	1	X	F = \bar{A}	Complementar A

TABLA 3.1 TABLA DE FUNCION PARA UN ALU.

El control puede diseñarse usando un procedimiento clásico lógico secuencial. Este procedimiento requiere una tabla de estado con ocho estados, cuatro entradas y nueve salidas.

El circuito secuencial que se va a deducir de cada estado, no será fácil de obtener debido a la gran cantidad de variables.

III.3 CONTROL MICROPROGRAMADO.

En un microprograma de control, las variables de control que inician micro-operaciones, se almacenan en la memoria. La memoria de control es comúnmente una ROM, ya que la secuencia de control es permanente y no necesita alteración.

A continuación se proporcionan algunos terminos utilizados con frecuencia en este tema.

Se empezará por definir Palabra de Control (CW: Control Word), como una palabra cuyos bits individuales representan las diferentes señales de control. Las palabras de control de un microprograma por lo general se denominan microinstrucciones.

Considerese que los microprogramas correspondientes al conjunto de instrucciones de una computadora están almacenados en una memoria especial a la que se denomina la Memoria de Microprograma. La unidad de control puede generar las señales de control para cualquier instrucción, leyendo en forma secuencial las CW del microprograma correspondiente en la memoria de microprograma. Esto sugiere organizar la unidad de control según se muestra en la figura 3.6. Para leer en forma secuencial las palabras de control, a partir de la memoria de microprograma se emplea un Contador de Microprograma (μ PC).

El bloque etiquetado "generación de dirección inicial", es responsable de la carga de la dirección inicial del microprograma en el PC, cada vez que se carga una nueva instrucción en IR (Instruction Register).

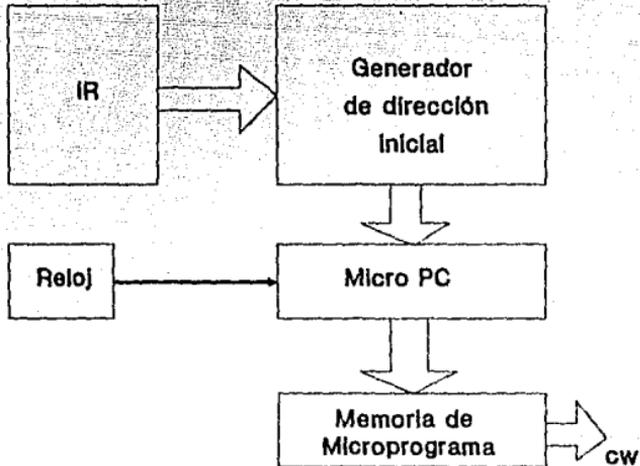


FIGURA 3.6 ORGANIZACION DE LA UNIDAD DE CONTROL PARA PERMITIR RAMIFICACION CONDICIONAL EN EL MICROPROGRAMA.

Después, el PC es incrementado en forma automática por el reloj. Provocando que se lean microinstrucciones sucesivas de la memoria.

Los controladores microprogramados utilizan como elemento básico un dispositivo programable (PROM, PAL, PLA), en el cual se graba un microcódigo, éste es el que contiene la secuencia de operaciones que debe realizar un controlador con arquitectura microprogramada; es decir un controlador microprogramado tiene una estructura definida, pero es posible cambiar la aplicación borrando los dispositivos programables y después volviéndolos a grabar.

Los controladores microprogramados tienen las siguientes características:

1. Un controlador microprogramado contiene una memoria dentro de su arquitectura de trabajo.
2. Se tiene un contador de programa PC, que permite direccionar la siguiente instrucción a ejecutarse.
3. Un microcontrolador, debe facilitar la ejecución de alguna instrucción desde cualquier localidad arbitraria.
4. La arquitectura microprogramada, debe permitir la ejecución de instrucciones de forma condicional o incondicional.

Para el diseño de esta arquitectura se requiere de ciertos mnemónicos, que son:

1. $PC \leftarrow PC + 1$: Cuenta Incondicionada (CI), la cual sólo se asigna si para pasar al siguiente estado no existe alguna condición.
2. $PC \leftarrow PC$; HOLD: $PC \leftarrow PC + 1$ CUENTA, ambos son una cuenta condicionada, esta ocurre en el caso contrario al anterior, sólo que cuando la bandera (flag) es falsa se toma como un HOLD, es decir permanece en el mismo estado, y si es verdadera, se hace una cuenta y pasa al siguiente estado.
3. $PC \leftarrow DIR$: Brinco Incondicional (BI), este mnemónico se asigna de manera similar a la cuenta incondicional, ya que cuando no existe condición para realizarse un brinco se presenta este.
4. $PC \leftarrow PC$ (HOLD); $PC \leftarrow DIR$, ambos significan brinco condicionado (BC), que aparece cuando para saltar al siguiente estado existe una condición, además si la bandera es falsa se mantiene en el estado.
5. $PC \leftarrow PC + 1$ cuenta condicionada (CC); $PC \leftarrow DIR$ brinco condicionado (BC), esta microinstrucción (C/BC) maneja dos situaciones que son cuenta con brinco condicionado, y la negación es porque la bandera es negativa o falsa.

Además de las microinstrucciones, este tipo de estructura utiliza una arquitectura de trabajo que se muestra en la fig.

3.7

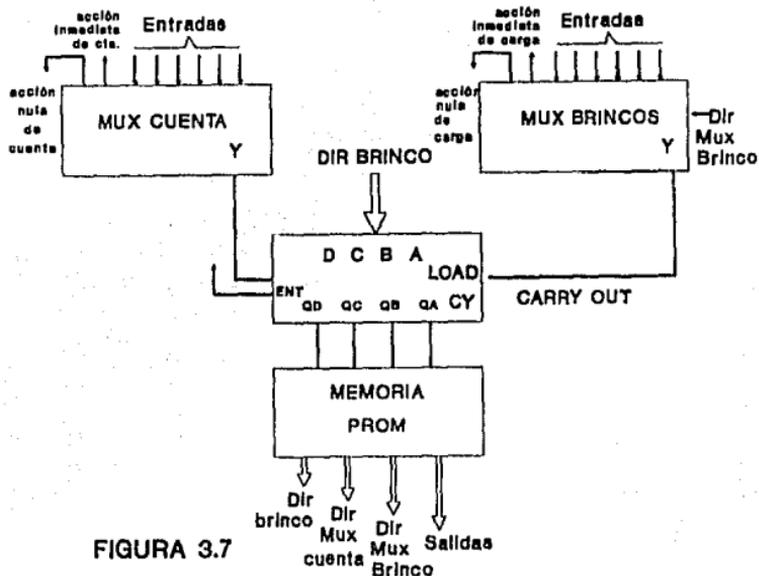


FIGURA 3.7

III.4 CONTROL CON P.L.A. (PROGRAM LOGIC ARRAY).

Un circuito combinacional, puede tener ocasionalmente condiciones de no importa. Cuando se configura con una ROM una condición de no importa, se convierte en una dirección de entrada que nunca ocurre. Las palabras en las direcciones de no importa, no necesitan ser programadas y pueden dejarse en su estado original (todas ceros o todos unos). El resultado es que no todos los patrones de bits disponibles en la ROM se usan, lo cual se considera como un desperdicio de equipo disponible.

Para aquellos casos en los cuales el número de condiciones de no importa es excesivo, es más económico usar un segundo tipo de componente LSI llamado Arreglo Lógico Programable (PLA). Un PLA es similar a una ROM en concepto; sin embargo, el PLA no produce la decodificación completa de las variables y no genera todos los términos mínimos como en una ROM. En un PLA, el decodificador se reemplaza mediante un grupo de compuertas AND, cada una de las cuales pueden ser programadas para generar un término producto de las variables de entrada.

El diseño de una unidad de control con un PLA, es muy similar al diseño que usa los métodos de registro de secuencia y decodificador. De hecho, el registro de secuencia en ambos métodos es el mismo. El PLA reemplaza esencialmente el decodificador y otros circuitos lógicos de decisión necesarios para la configuración de los componentes.

Para demostrar el procedimiento con un ejemplo, considérese el circuito de control para el multiplicador binario mostrado en la figura 3.8.

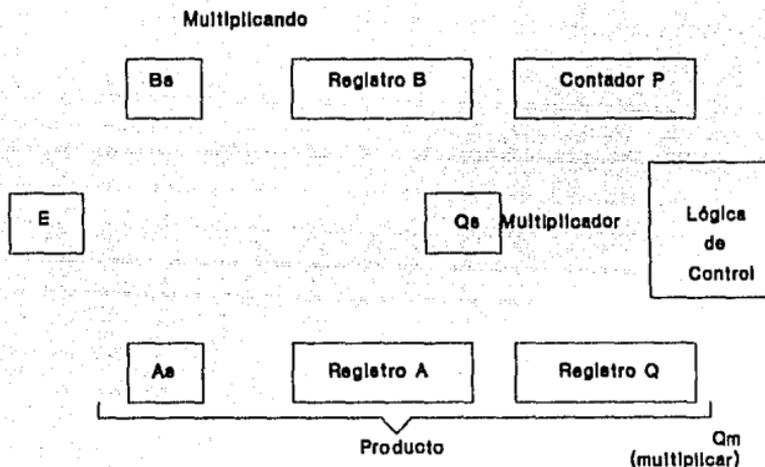
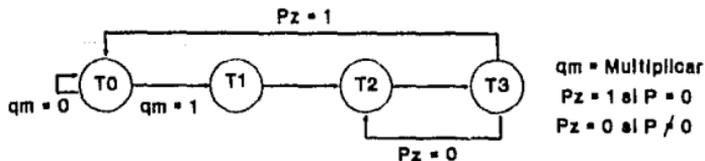


FIGURA 3.8 REGISTROS PARA EL MULTIPLICADOR BINARIO

Las especificaciones de control para el multiplicador binario se dan en la figura 3.9



(a) DIAGRAMA DE ESTADO

T0: Estado Inicial

T1: $Aa \text{ --- } Ba \oplus Qs, A \text{ --- } 0, E \text{ --- } 0, P \text{ --- } k$

Q1T2: $A \text{ --- } A + B, E \text{ --- } Cout$

T2: $P \text{ --- } P - 1$

T3: $AQ \text{ --- } ShrEAQ, E \text{ --- } 0$

(b) SECUENCIA DE TRANSFERENCIAS DEL REGISTRO

FIGURA 3.9 DIAGRAMA DE ESTADO DE CONTROL Y SECUENCIA DE MICROOPERACIONES DEL MULTIPLICADOR.

A partir de esta información se obtiene la tabla de estado 3.2.

Estado presente		Entradas			Estado siguiente		Salidas				
G_x	G_1	q_m	P_x	Q_1	G_x	G_1	T_0	T_1	T_x	L	T_3
0	0	0	X	X	0	0	1	0	0	0	0
0	0	1	X	X	0	1	1	0	0	0	0
0	1	X	X	X	1	0	0	1	0	0	0
1	0	X	X	0	1	1	0	0	1	0	0
1	0	X	X	1	1	1	0	0	1	1	0
1	1	X	0	X	1	0	0	0	0	0	1
1	1	X	1	X	0	0	0	0	0	0	1

TABLA 3.2 TABLA DE ESTADO PARA EL CIRCUITO DE CONTROL

El estado presente, se determina a partir de los flip-flops G_1 y G_x . Las variables de entrada para el circuito de control son q_m , P_x y Q_1 . El siguiente estado de G_1 y G_x puede ser una función de una de las entradas, o pueden ser independientes de cualquier entrada. Si una variable de entrada no influye en el estado siguiente, se marca con un X de no importa. Si el siguiente estado es una función de una entrada particular, el estado presente se repite en la tabla y a los estados siguientes se les asignan diferentes valores binarios. La tabla, lista también todas las salidas de control como una función del estado presente y de las condiciones de entrada.

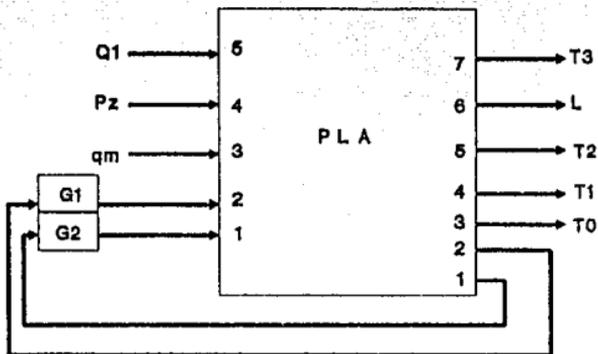
El diagrama de bloque de control PLA se muestra en la figura 3.10(a). el PLA se conecta al registro de secuencia con dos flip-flops G_1 y G_2 . Las entradas al PLA, son los valores del estado presente del registro de secuencia y las tres entradas externas. Las salidas del PLA, generan el estado siguiente para el registro de secuencia y las variables de salida de control. El estado presente del registro de secuencia, conjuntamente con las condiciones de entrada, determinan en un tiempo dado los valores de salida y del siguiente estado para el registro de secuencia.

Un PLA se especifica por medio del número de entradas, el número de términos de producto y el número de salidas. Para este caso, se tienen cinco entradas y siete salidas, el número de términos producto es una función del circuito que se desea configurar.

La tabla de programa del PLA se puede obtener directamente a partir de la tabla de estado, sin necesidad de procedimientos de simplificación. La tabla de programa del PLA en la figura 3.10(b) especifica siete términos producto, uno para cada fila en la tabla de estado. Las terminales de entrada y salida se marcan con números y las variables aplicadas a esas terminales numeradas se indican en el diagrama de bloque.

La unidad PLA en un control PLA, puede visualizarse como una memoria de control que almacena información de control para el sistema.

Las salidas del registro de secuencia conjuntamente con las entradas externas, podrían considerarse como una dirección para tal memoria de control. Las salidas suministran una palabra de control para el procesador de datos y la información del estado siguiente especifica un valor parcial para la siguiente dirección en la memoria de control.



(a) DIAGRAMA DE BLOQUE

Término del producto	Entradas					Salidas							Comentarios	
	1	2	3	4	5	1	2	3	4	5	6	7		
1	0	0	0	-	-	-	-	1	-	-	-	-	-	To = 1, qm = 0
2	0	0	1	-	-	-	1	1	-	-	-	-	-	To = 1, qm = 1
3	0	1	-	-	-	1	-	-	1	-	-	-	-	T1 = 1
4	1	0	-	-	0	1	1	-	-	1	-	-	-	Tz = 1, Q1 = 0
6	1	0	-	-	1	1	1	-	-	1	1	-	-	Tz = 1, Q1 = 1, L = 1
6	1	1	-	0	-	1	-	-	-	-	-	1	-	T3 = 1, Pz = 0
7	1	1	-	1	-	-	-	-	-	-	-	1	-	T3 = 1, Pz = 1

(b) TABLA DE PROGRAMA PLA
FIGURA 3.10 CONTROL DEL PLA PARA UN MULTIPLICADOR BINARIO

IV. ORGANIZACION DE LA MEMORIA

Para el implante de la memoria principal, se utilizan circuitos integrados de semiconductores, es posible obtenerlos en una amplia gama de velocidades con un consumo de energía muy bajo. Cuando se presentaron por primera vez, a fines de la década de 1960, eran mucho más costosos que las memorias de núcleos magnéticos que reemplazaron. Debido a los avances de la tecnología de VLSI (Very Large Scale Integration: Integración a Muy Gran Escala), el costo de las memorias semiconductoras ha descendido en forma notable.

Se pueden dividir las memorias semiconductoras en dos tipos: Bipolar y de MOS (Metal-Oxide Semiconductor: Semiconductor de Oxido-Metal).

Los chips de memoria, internamente están compuestos por celdas de memoria que almacenan un bit de información, éstas se organizan en forma de arreglos (ver figura 4.1). Una palabra de memoria está formada por un renglón de celdas conectadas a una línea común, conocida como línea de palabra que son manejadas por el decodificador de direcciones.

Las celdas de cada columna están conectadas por dos líneas, denominadas líneas de bit, a un circuito de detectar/escribir, éstos a su vez conectados a líneas de entrada/salida del chip. Los circuitos detectar/escribir durante una operación de leer, detectan o leen la información almacenada en las celdas seleccionadas por la línea de palabra,

y transmiten esta información a las líneas de salida de dato.

Durante una operación de escribir, reciben información de entrada y la almacenan en las celdas de la palabra seleccionada.

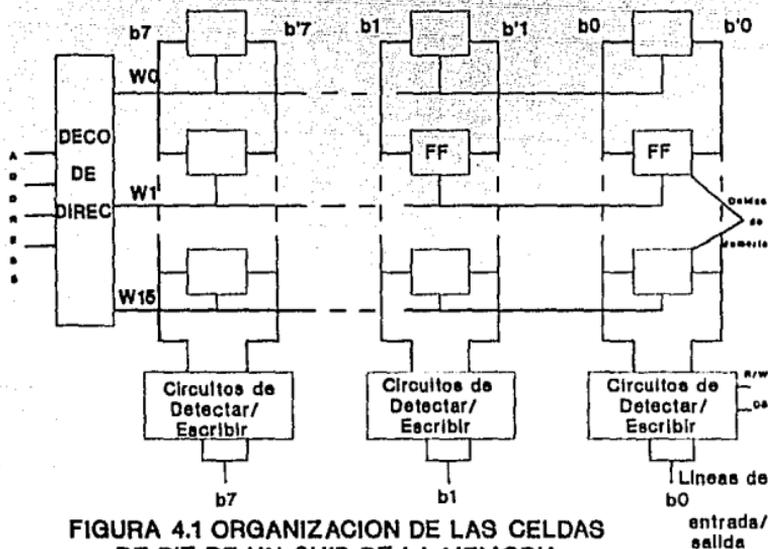


FIGURA 4.1 ORGANIZACION DE LAS CELDAS DE BIT DE UN CHIP DE LA MEMORIA

En la figura 4.1 se muestra un chip de memoria que consiste en 16 palabras de 8 bits cada una, este arreglo se denomina generalmente de 16 x 8, y tiene 128 bits. La entrada y salida de datos de cada circuito detectar/escribir están conectadas a una sola línea de datos bidireccional, ocupando 8 patas del chip. El direccionamiento de palabra ocupa 4 terminales y se puede observar también la entrada R/W (Read/Write: Leer/escribir), que sirve para especificar la operación requerida y la entrada CS (Chip Select: Selección de Chip), utilizada para seleccionar en un sistema de memoria de chips múltiples, un chip dado. El circuito de memoria tiene 14 terminales, puede fabricarse en forma de chip de 16 terminales, dejando dos terminales para suministro de energía y conexión a tierra.

IV.1 TECNOLOGIAS Y CARACTERISTICAS

Todo sistema ordenador precisa de memoria para:

- Almacenar las instrucciones que forman el programa.
- Almacenar datos y resultados intermedios.

La memoria está organizada en palabras, cada una con una única dirección. La longitud de palabra, en general, es la misma que la del Microprocesador y bus de datos.

Hay dos aspectos en la consideración de memorias para Microprocesadores en relación con el sistema de memoria de miniordenadores:



1) Los sistemas basados en Microprocesadores, deben ser compactos y baratos, por lo que la organización de memoria, en cuanto a eficiencia, cobra una vital importancia.

2) Las aplicaciones de los Microprocesadores son, típicamente de carácter dedicado, es decir, el programa es fijo y está grabado en memoria no volátiles y de acceso sólo para lectura.

Las memorias en una jerarquía pueden clasificarse en base a varios atributos. Un atributo típico es el método de acceso, que divide a las memorias en tres clases básicas: Las Memorias de Acceso Aleatorio (RAM: Random-Access Memory), las Memorias de Acceso Secuencial (SAM: Sequential Access Memory) y los dispositivos de almacenamiento de acceso directo (DASD: Direct-Access Storage Devices).

Otro atributo utilizado con frecuencia para clasificar memorias, es la velocidad o tiempo de acceso de la memoria. En la mayoría de los computadores, la jerarquía de memoria está organizada de modo que el nivel superior disponga de la velocidad de memoria más rápida y el nivel inferior la velocidad más lenta. La tabla 4.1 muestra las características de los dispositivos de memoria en una jerarquía de memoria.

Las memorias utilizadas en microcomputadores son, en general, de semiconductores y de acceso aleatorio. Otros tipos de memorias utilizadas para aplicaciones especiales son las memorias de acceso secuencial (registros de desplazamiento, dispositivos de acopio de carga (CCD) y memorias de burbuja magnética), caracterizadas por tener una gran capacidad de

almacenamiento y un tiempo de acceso relativamente elevado.

Finalmente, existen las memorias auxiliares de gran capacidad y bajo coste, como cintas magnéticas, discos y cassettes, que por su forma de acceso son consideradas como periféricos del microcomputadores.

Nivel i	Tipo de memoria	Tecnología	Tamaño típico S _i	Tiempo medio de acceso t _i	Unidad de transferencia
1	Cache	Bipolar, HMOS, ECL	2K-128K bytes	30-100 nseg	1 palabra
2	Memoria principal o primaria	MOS ferritas	4k-16M bytes	0.25-1 μ s 0.5-1 μ s	2-32 palabras
3	Memoria voluminosa (LCS, ECS)	Ferritas	64K-16M bytes	5-10 μ s	2-32 palabras
opcional)					
4	Disco de cabezas fijas o tambor	Magnética	8M-256M bytes	5-15 mseg	1K-4K bytes
5	Disco de brazo móvil	Magnética	8M-500M bytes	25-75mseg	4K bytes
6	Cinta	Magnética	50 M bytes	1-5 seg	1K-16K bytes

TABLA 4.1 CARACTERISTICA DE LOS DISPOSITIVOS DE MEMORIA EN UNA JERARQUIA DE MEMORIA.

Las memorias de semiconductores de acceso aleatorio, se fabrican utilizando diversas tecnologías (TTL, STTL, PMOS,

NMOS, CMOS, I²L, ECL) con características distintas en tiempos de acceso, posibilidades de integración y consumo.

Las memorias de sólo lectura (ROM), se utilizan en microcomputadoras para almacenar programas, tablas y constantes.

Existen tres tipos básicos de memorias ROM:

1) ROM de máscara: Son memorias programadas en el proceso de fabricación e inalterables posteriormente. Son las más económicas en coste por bit, pero sólo para grandes cantidades, ya que el coste fijo es elevado, por lo que son utilizadas en la producción de grandes series.

2) ROM programable (PROM): Son memorias que pueden ser programada por el usuario pero no modificadas, ya que el proceso de programación es irreversible. Son más aptas que las memorias ROM de máscara para pequeñas series.

3) ROM reprogramables (EPROM): Son memorias que pueden ser grabadas y borradas o modificadas posteriormente. Las memorias EPROM son, especialmente aptas para la realización de prototipos y pequeñas series.

Las memorias ROM existen con capacidades muy diversas, observándose un rápido crecimiento en los niveles de integración, llegando hasta el momento a capacidades de 64 kbits.

Las memorias de escritura y lectura (RWM o RAM), tienen como finalidad el almacenamiento de datos y resultados intermedios o también, se utilizan como memoria de programa

para zonas del mismo que deben ser modificadas para la fase de puesta a punto de un sistema, o para sistemas en los que es cargado al inicializar el sistema a partir de un medio auxiliar de memoria.

Las memorias RAM son volátiles, es decir, la información almacenada se pierde si falla la alimentación del sistema. Si es preciso salvar la información en caso de fallo de tensión, se deberá disponer de un sistema de baterías para alimentación de la RAM. En este caso, las RAM de tecnología CMOS son especialmente útiles por su bajo consumo.

Existen dos tipos de memoria RAM de Acceso Aleatorio:

1) RAM Estáticas: La información se almacena en un biestable, de forma que el contenido se mantiene mientras haya alimentación. La figura 4.2 muestra la organización de una memoria de $64 \text{ K} \times 16$ utilizando chips de memoria estática de $16 \text{ K} \times 1$.

2) RAM Dinámicas: La información se mantiene en forma de carga en la capacidad puerta-sustrato de un transistor MOS. Esta carga se disipa con el tiempo (del orden de milisegundos), por lo que es necesario un refresco periódico consistente en un ciclo de lectura/escritura en cada bit de memoria, realizado de forma paralela con todos los bits de una misma columna a la vez. La figura 4.3 muestra un diagrama a bloque de una unidad de memoria dinámica.

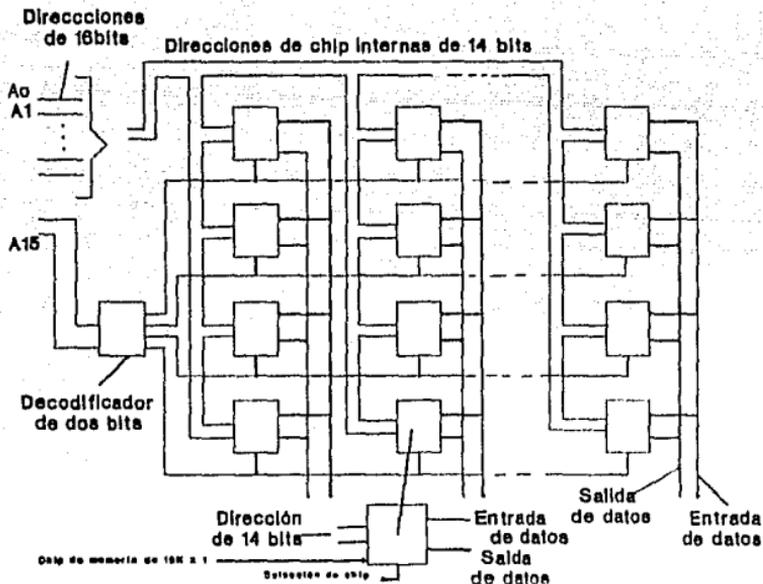


FIGURA 4.2 ORGANIZACION DE UNA MEMORIA DE 64K X 16 UTILIZANDO CHIPS DE MEMORIA ESTADICA DE 16K X 1

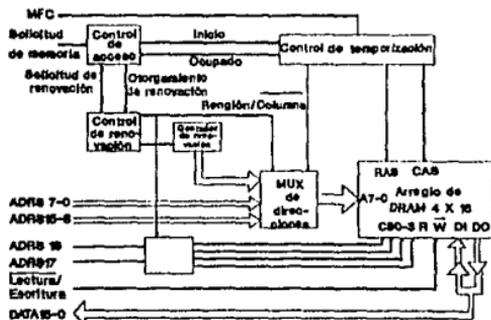


FIGURA 4.3 DIAGRAMA DE BLOQUE DE UNA UNIDAD DE MEMORIA DINAMICA DE 256K X 16

Las ventajas de la memoria dinámica, es que permiten una mayor densidad de integración (actualmente del orden de 64 k bits frente a 16 k en las estáticas), al precisar menos transistores por célula elemental de memoria, por lo que son más baratas que las RAM estáticas de la misma capacidad, y consumen menos potencia.

Existen dos tipos de estructura conceptualmente distintos en cuanto a la forma de considerar la memoria, en una microcomputadora:

1) Estructura Harvard: Cuando la memoria de programa (pasiva) recibe un tratamiento distinto de la memoria de datos (activa). Las diferencias pueden consistir en el modo de direccionamiento, de forma que las líneas de control (o alguna de ellas) que activan cada subsistema (ROM, RAM) no son comunes, pudiéndose llegar a una total diferenciación de los buses de comunicación, lo cual permitiría un cierto solapamiento entre los ciclos de búsqueda y ejecución de las instrucciones.

2) Estructura Princeton o Von Newman: Cuando el tratamiento de memoria es único, sin distinción entre datos e instrucciones. Este sistema es el más sencillo y el más flexible, y es el que emplean la mayoría de las microcomputadores.

Al interconectar la CPU con el sistema de memoria, se deberán considerar 3 tipos de información:

1) **Direcciones:** Sirven para seleccionar una posición de memoria determinada, y deberán mantenerse estables durante todo el ciclo de memoria (en lectura o escritura).

2) **Datos:** Son las líneas que forman el bus de datos y a través de ellas, se realiza la transferencia de información.

3) **Control:** Son las señales que activan y sincronizan la transferencia. Las señales de control a considerar en general son:

a) Líneas de selección.

b) Líneas de validación de transferencia.

c) Líneas de sincronización.

IV.2 MEMORIA VIRTUAL

En muchos computadores, los programadores advierten con frecuencia que alguno de sus programas más extensos, no pueden alojarse en memoria para su ejecución por falta de espacio. La solución habitual es introducir esquemas de administración, que asignen inteligentemente apropiadas porciones de memoria a los usuarios, para que sus programas se ejecuten eficazmente. En esta sección, se describe la utilización de la memoria virtual para alcanzar este objetivo.

Las técnicas generales de trasladar automáticamente el programa y, los bloques de datos necesarios a memoria principal (MP) física para su ejecución, se denominan técnicas de memoria virtual. Los programas, y por lo tanto la CPU, hacen referencia

a un espacio de instrucciones y de datos independiente del espacio físico de la MP. Las direcciones binarias que emite la CPU, ya sea para instrucciones o para datos, se conocen como direcciones virtuales o lógicas. El mecanismo que opera en estas direcciones virtuales y las traduce a localizaciones verdaderas de la jerarquía física; por lo general, se implanta empleando una combinación de componentes de Hardware y de Software. Si el resultado de traducir (o hacer el mapa de) una dirección virtual específica, es una localización de la MP física, el contenido de esa localización se utiliza de inmediato según se necesite. Por otra parte, si la localización no está en la MP, su contenido debe traerse a una localización adecuada de la MP y usarse después.

La memoria virtual proporciona a los programadores, la ilusión de que existe una memoria muy grande a su disposición, aún cuando la MP física disponible puede ser pequeña. Esta ilusión puede lograrse permitiendo que el programador opere en el espacio de nombres, mientras la arquitectura provee un mecanismo para traducir las direcciones (virtuales) generadas por el programa (en tiempo de ejecución), a direcciones de posiciones físicas.

La forma más fácil de método de traducción, se basa en considerar que todos los programas y datos están compuestos por páginas de longitud fija. Estas páginas son unidades básicas de bloques de palabras, que deben ocupar siempre localizaciones contiguas, ya sea que residan en la MP, o en almacenamiento

secundario. Lo común es que las páginas sean de 512 o 1024 palabras de longitud fija. Estas páginas son unidades básicas y constituyen la unidad básica de información que se traslada entre la memoria principal y el almacenamiento secundario, siempre que el mecanismo de traducción determine que se necesita un traslado. La idea de la memoria virtual tiene como objetivo principal servir de puente para la brecha que haya entre la MP, y el almacenamiento secundario. Por lo general, se implanta en parte usando técnicas de Software.

En la figura 4.4 se muestra en forma esquemática un método de traducción de dirección, basado en el concepto de páginas de longitud fija. Cada dirección virtual generada por la CPU, ya sea para una traida de instrucción o para una operación de traida/almacenamiento de operando, se interpreta como un número de página (bits de orden superior), seguido por un número de palabra (bit de orden inferior).

Una tabla de página de la MP especifica la localización de las páginas que en ese momento se encuentren en la MP. La dirección inicial de la tabla de páginas se conserva en el registro base de tabla de página. Al sumar el número de página al contenido de este registro, se obtiene la dirección de la notación correspondiente que tenga en la tabla de página. El contenido de esta localización, nombra al bloque de la MP en donde reside actualmente la página solicitada en la MP, la anotación de la tabla de página apunta al lugar en que se encuentra dicha página en el almacenamiento secundario.

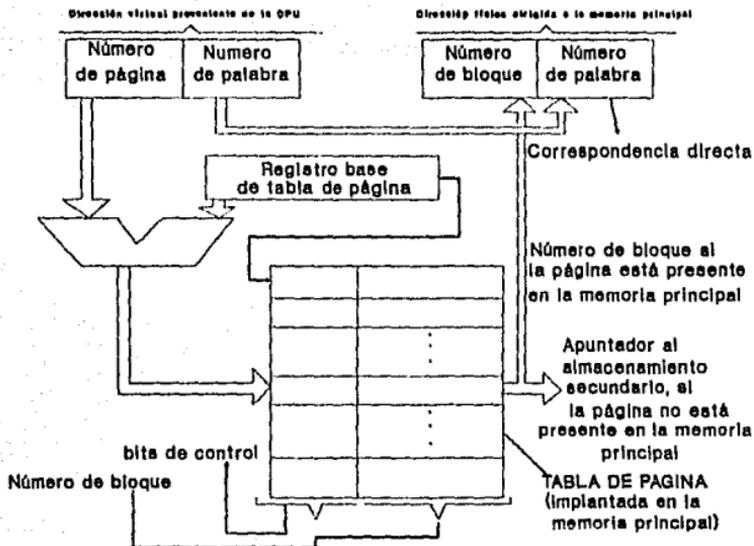


FIGURA 4.4 TRADUCCION DE DIRECCION DE LA MEMORIA VIRTUAL

Los bits de control indican si la página está o no presente en la MP. También pueden contener alguna información acerca de su uso pasado, etc., para fines de implantar el algoritmo de reemplazo de página.

IV.3 MEMORIA CACHE

Las memorias caché son memorias intermedias de alta velocidad que se insertan entre los procesadores y la MP para capturar aquellas porciones de los contenidos de MP que están actualmente en uso. También, pueden insertarse entre la MP y la memoria de masas. Como las memorias caché son típicamente entre

5 y 10 veces más rápidas que la MP, pueden reducir el tiempo de acceso efectivo a memoria si se diseñan e implementan cuidadosamente. En esta sección se discuten las características de las memorias caché y las diferentes estrategias de administración de tales memorias.

El éxito de las memorias caché puede atribuirse a la prioridad de localidad de referencias. La efectividad de la caché en capturar localidades se mide por la fracción asintótica de referencias de programas encontradas, en la caché, denominada tasa de aciertos "a".

La memoria caché consta generalmente de dos partes, el directorio caché (DC) y la memoria de acceso aleatorio (RAM). La porción de memoria está particionada en un número de bloques de igual tamaño, llamados Marcos de Bloque. El directorio, que está implementado generalmente por alguna forma de memoria asociativa, consta de las marcas de dirección de bloque y algunos bits de control tales como el bit "poluto"*, el bit "válido" y los bits de protección. Las marcas de dirección contienen las direcciones de bloque de los bloques, que están actualmente en la memoria caché.

Los bits de control se utilizan para administración y control de acceso de la caché. Por tanto, la caché contiene un

* EL BIT "POLUTO" ES EL EL QUE NOS INDICA CON UN "1", SI EXISTE ENLACE DE ENTRADA Y "0" SI ES ENLACE DE SALIDA

conjunto de parejas dirección-datos, cada una de las cuales consiste en la dirección de un bloque en MP y en una copia de los contenidos del bloque de MP correspondiente a esta dirección.

El directorio de caché, puede implementarse como una tabla de búsqueda implícita o explícita. En el directorio explícito, los datos referencias se extraen de la porción de memoria de la caché sólo después de haber buscado la marca de dirección correspondiente.

La operación caché es sencilla conceptualmente, como se ilustra en la figura 4.5, para una operación de lectura. El procesador genera una dirección virtual que se traduce a una dirección de memoria física, a través de una memoria de reserva de traducciones (MRT). Si hay acuerdo en MRT, se recupera la dirección de página física correspondiente para formar la dirección física, y se actualiza el estado de remplazamiento de las entradas en MRT.

Si la MRT no contiene la pareja de direcciones (virtual, física) precisa para la traducción, se transfiere la dirección virtual al traductor para determinar la dirección física.

En general, el diseño de una caché está sujeto a limitaciones y compromisos diferentes de los de la MP. Uno de los parámetros importantes en el diseño de una memoria caché, es la política de colocación que establece la correspondencia entre el bloque de MP y los de la caché. La caché podría estar particionada en varias cachés independientes para segregar

diferentes tipos de referencias. Una caché no particionada se dice que es homogénea. La caché puede ser multipuerto de modo que, puede hacerse dos o más solicitudes a la caché concurrentemente.

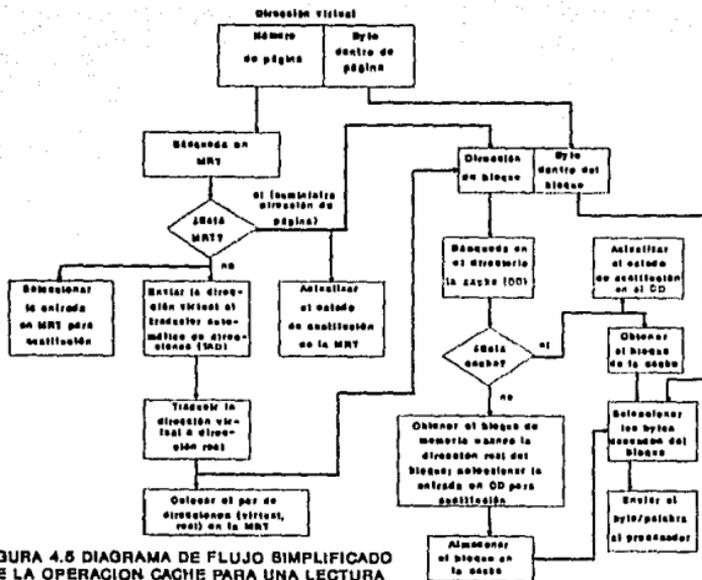


FIGURA 4.5 DIAGRAMA DE FLUJO SIMPLIFICADO DE LA OPERACION CACHE PARA UNA LECTURA

La caché se diseña generalmente para que sea transparente al usuario. En consecuencia, para localizar un elemento en la caché es necesario disponer de una función que convierta la dirección de MP en una posición de caché. Por uniformidad de referencia, tanto la caché como la MP están divididas en unidades de igual tamaño, llamadas bloque en la memoria y marcos de bloque en la caché.

IV.4 MEMORIA INTERLEAVED

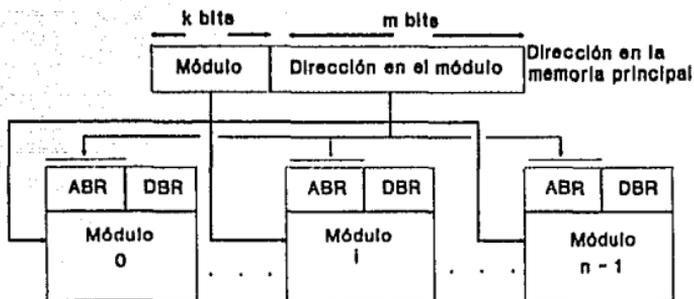
Si la MP está estructurada como un conjunto de módulos físicamente separadas, cada uno con su propio registro buffer de dirección (ABR: Address Buffer Register), y con su registro buffer de datos (DBR: Data Buffer Register), es posible que más de un módulo realice operaciones de leer o escribir al mismo tiempo.

La razón de transmisión de palabras promedio hacia o desde el sistema total de la MP puede entonces incrementarse. Se necesitarán controles extras, pero ya que a menudo la velocidad de la MP es el cuello de botella de las velocidades de cálculo, el gasto se justifica por lo general en las computadoras grandes.

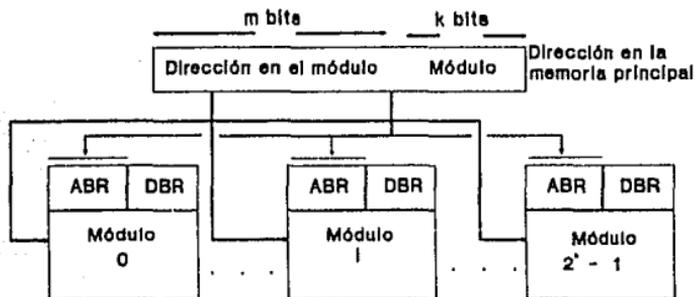
La forma en que las direcciones individuales están distribuidas a través de los módulos, es un factor crítico que determina el número de módulos promedio que pueden mantenerse ocupados conforme avancen los cálculos.

En la figura 4.6 se muestran dos métodos para la disposición general de la dirección. En el primer caso, la dirección de la MP generada por la CPU se decodifica como se muestre en la figura 4.6 (a). Los k bits de orden superior de la dirección nombran uno de los n módulos, y los m bits de orden inferior nombran una palabra específica de ese módulo. Si la CPU emite una solicitud de leer localizaciones consecutivas, como lo hace al traer de la memoria las instrucciones de un programa en línea recta, entonces la CPU sólo mantiene ocupado un módulo. Sin embargo, dispositivos con capacidad de acceso directo a la memoria (DMA: Direct Memory Access) pueden estar operando en otros módulos de la memoria.

La segunda y más efectiva forma de direccionar los módulos se muestra en la figura 4.6 (b). Se le conoce como entrelazamiento de la memoria (Interleaved). Los k bits de orden inferior de la dirección de la MP, seleccionan un módulo y los m bits de orden superior nombran una localización que está dentro de ese módulo. Por lo tanto, las direcciones consecutivas se localizan en módulos sucesivos. De esta forma, cualquier componente del sistema, por ejemplo, la CPU o un dispositivo de DMA, que genera solicitudes de acceso a localizaciones consecutivas de la MP, puede mantener ocupados varios al mismo tiempo. Esto da como resultado un promedio mayor de utilización del sistema de memoria como un todo.



a) PALABRAS CONSECUTIVAS EN UN MODULO



b) PALABRAS CONSECUTIVAS EN MODULOS CONSECUTIVOS

FIGURA 4.6 DIRECCIONAMIENTO EN SISTEMAS DE MEMORIA DE MODULOS MULTIPLES

En el sistema de la figura 4.6 (b) debe haber 2^m módulos. De otra forma, habrá espacios de localizaciones inexistentes en el espacio de direcciones de la MP. Esto presenta un punto

práctico. El primer sistema, el de la figura 4.6(a), es más flexible que el segundo, ya que pueden utilizarse con él cualquier número de módulos hasta 2^k . Por lo general, a los módulos se les asigna direcciones consecutivas en la MP, de 0 en adelante. Entonces, un sistema ya existente puede ampliarse simplemente agregando uno o más módulos, según se necesite. El segundo sistema debe tener siempre el conjunto completo de 2^k módulos y la falla de cualquier módulo afectaría a todas las áreas del espacio de dirección.

IV.5 MEMORIA ASOCIATIVA

Los datos almacenados en una memoria asociativa se direccionan por sus contenidos. En ese sentido, las memorias asociativas han sido conocidas como memorias direccionables por contenido, memorias de búsqueda paralela y memorias multiacceso. La principal ventaja de la memoria asociativa sobre la RAM, es su capacidad para realizar operaciones de búsqueda y comparaciones importantes, tales como almacenamiento y recuperación de bases de datos de rápida evolución, de imágenes, visión por computador e inteligencia artificial. El principal inconveniente de la memoria asociativa es su elevado Costo-Hardware. En la actualidad, el coste de la memoria asociativa es muy superior al de las memorias RAM.

En la figura 4.7 se presenta el modelo de la estructura de una MA básica. La matriz de la memoria asociativa está formada por n palabras con m bits por palabra. Cada celda de la matriz

$n \times m$ consta de un biestable asociado con algunas puertas lógicas para comparación de patrones y control de lectura-escritura. Esta estructura de lógica incluida en memoria, permite la lectura o escritura en paralelo de la matriz de memoria. Una sección de bit es una columna vertical de celdas que se hallan en la misma posición de bit de todas las palabras.

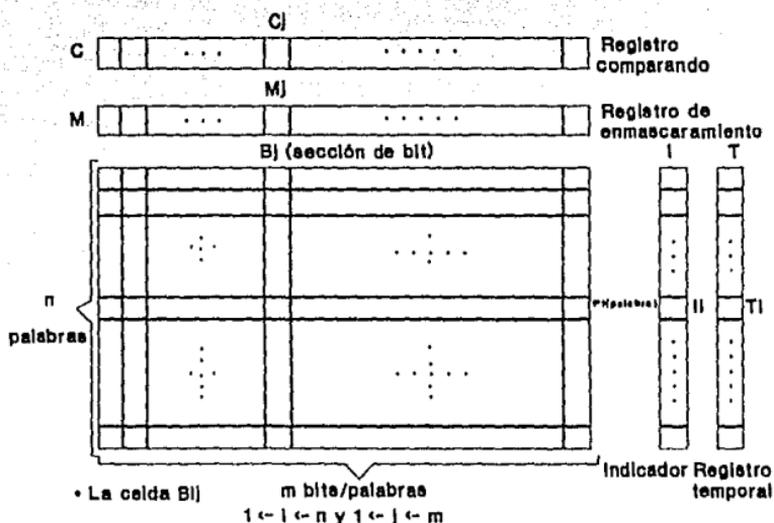


FIGURA 4.7 MATRIZ DE MEMORIA ASOCIATIVA Y REGISTROS DE TRABAJO

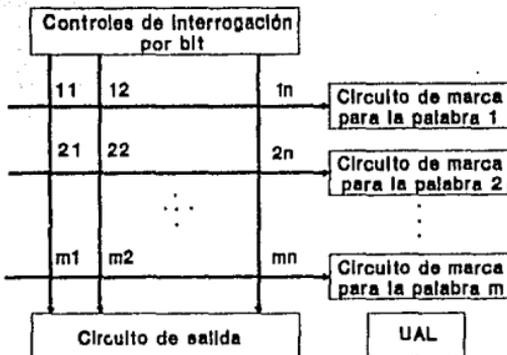
ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

En la práctica, la mayoría de las memorias asociativas tienen capacidad de realizar operaciones palabra-paralelo, es decir, todas las palabras de la memoria asociativa están implicadas en las operaciones de búsqueda en paralelo. Esto difiere drásticamente de las operaciones palabra-serie encontradas en las RAM. Dependiendo de cómo se vean afectadas las selecciones de bit en la operación, consideramos a continuación dos organizaciones diferentes de las memorias asociativas:

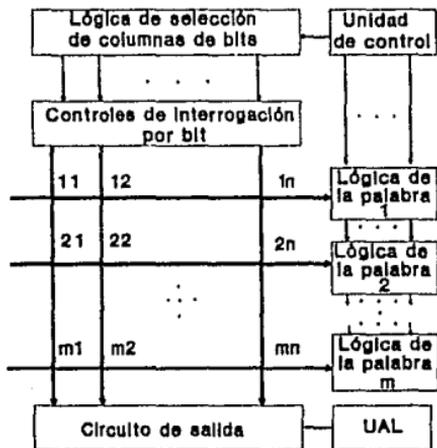
- Organización bit-paralelo.- En una organización bit-paralelo, el proceso de comparación se efectúan en modo paralelo por palabra y paralelo por bit. Todas las secciones de bit que no estén enmascaradas por la configuración de enmascaramiento, toman parte en el proceso de comparación. En esta organización se utilizan marcas de coincidencia para todas las palabras (figura 4.8a). Cada punto de cruce en la matriz es una celda de un bit. Esencialmente, la matriz de celdas está implicada en una operación de búsqueda.

- Organización bit-serie.- La organización de memoria de la figura 4.8 (b) funciona con una sección de bit cada vez a través de todas la palabras. La sección de bit concreta la selección de una unidad de control y lógica adicional. Las lecturas de celdas se utilizarán en las subsiguientes operaciones sobre sección de bit.

Las memorias asociativas se emplean principalmente en búsqueda y recuperación de información no numérica. La organización bit-serie requiere Hardware, pero su velocidad es menor. La organización bit-paralelo requiere lógica adicional para detección de palabras coincidentes, pero su velocidad es mayor.



a) ORGANIZACION BIT-PARALELO



b) ORGANIZACION BIT-SERIE

FIGURA 4.8 ORGANIZACION DE LA MEMORIA ASOCIATIVA

V. ORGANIZACION DEL CONTROL MICROPROGRAMADO.

En el capítulo III se presentó la posibilidad de generar señales de control en la CPU, que se conocen como control microprogramado por medio de Software. Se mencionó brevemente el enfoque general y algunos méritos relativos de tales diseños. En este capítulo, se examinará con mayor detalle esta opción.

Se iniciará con la premisa básica de que la microprogramación es un enfoque de Software, y por lo tanto puede manejarse utilizando estructuras de Hardware, afines a aquellas que se emplean para el secuenciamiento de programas ordinarios. Ya que las microinstrucciones van a ejecutarse en orden secuencial desde localizaciones consecutivas en la memoria de microprograma (también denominada almacenamiento de control), es lógico que se emplee un contador de microprograma (μ PC) para llevar el control de las direcciones.

El propósito de la unidad de control es iniciar una serie de pasos secuenciales de microoperaciones. Durante cualquier tiempo dado, se deben iniciar ciertas operaciones mientras que otras permanecen latentes. Así, las variables de control en un tiempo dado pueden ser representadas por una cadena de 1 ó 0 llamadas palabras de control. Como tales, dichas palabras de control pueden ser programadas para iniciar las diferentes componentes en el sistema de una manera organizada.

Una unidad de control cuyas variables de control se almacenan en una memoria, se llama unidad de control microprogramada. Cada palabra de control de memoria se llama microinstrucción y una secuencia de microinstrucciones se llama microprograma.

V.1 CONTROL MICROPROGRAMADO.

El uso del microprograma comprende la ubicación de todas las variables de control en palabras de la ROM, para usarias por medio de las unidades de control a través de operaciones sucesivas de lecturas. El contenido de la palabra en la ROM en una dirección dada especifica las microoperaciones del sistema. Las variables de control almacenadas en la memoria son leídas una a una, para iniciar la secuencia de microoperaciones del sistema.

Las palabras almacenadas en la memoria de control, son microinstrucciones y cada una de ellas especifica una o mas microoperaciones para los componentes en el sistema. Una vez que se ejecutan estas microoperaciones, la unidad de control debe determinar la siguiente dirección. Por tanto, unos pocos bits de la generación de la dirección para la siguiente microinstrucción. Así, una microinstrucción contiene bits para iniciar microoperaciones y bits para determinar la siguiente dirección para la memoria de control en sí misma.

Además de la memoria de control, una unidad de control de microprograma debe incluir circuitos especiales para seleccionar la siguiente dirección como se especifica por la microinstrucción. Estos circuitos y la configuración de los bits de microinstrucción almacenados en la memoria varían de una unidad a otra.

La inspección del diagrama de estado revela que la secuencia de direcciones de control del microprograma debe tener las siguientes cualidades:

1. Provisión para la carga de una dirección externa como resultado de la ocurrencia de las señales externas q_4 y q_5 .
2. Provisión para la sucesión consecutiva de direcciones.
3. Provisión para escoger entre dos direcciones como una función de los valores presentes de las variables de condición S y E.

Cada microinstrucción debe contener un número de bits para especificar la manera en que se selecciona la nueva dirección. La organización de la unidad de control del microprograma se muestra en la figura 5.1. La memoria de control es una ROM de 8 palabras por 14 bits. Los primeros nueve bits de una palabra de microinstrucción, contiene las variables de control que inician las microoperaciones. Los últimos cinco bits suministran información para seleccionar la siguiente dirección. El registro de direcciones de control (CAR: Control Address Register), almacena la dirección de la memoria de control. Este recibe un valor de entrada cuando se habilita su control de

carga; de otra manera se incrementa en 1.

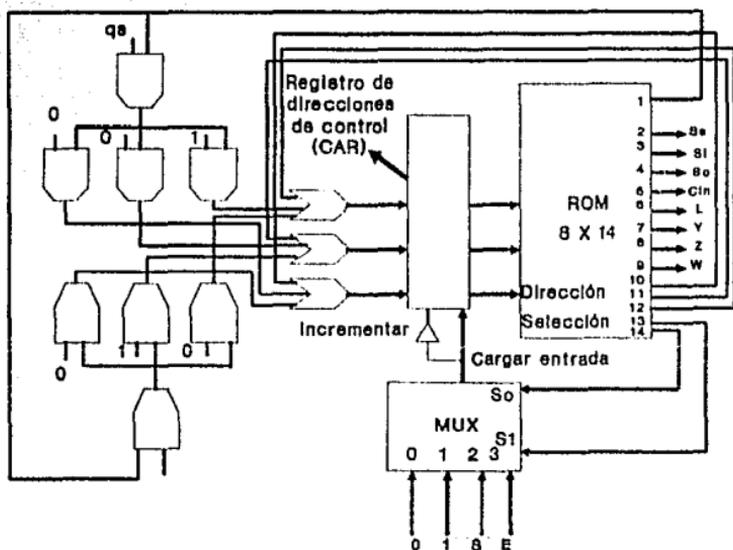
Un CAR es esencialmente un contador con capacidad de carga en paralelo.

Los bits 10, 11 y 12 de una microinstrucción, contienen una dirección para el CAR. Los bits 13 y 14 seleccionan una entrada para un multiplexor. El bit 1 suministra la condición de estado inicial denotada por la variable X y también habilita una dirección externa cuando q_0 o q_1 es igual a 1. Se estipula que cuando $X = 1$ el campo de dirección de la microinstrucción debe ser igual a 000. Entonces si $q_0 = 1$ el campo de dirección de la micro-instrucción debe ser igual a 000.

Entonces si $q_0 = 1$, la dirección 001 está presente en las entradas del CAR, pero si $q_1 = 1$, la dirección 010 se aplica al CAR. Si ambas entradas q_0 y q_1 son cero, la dirección cero de los bits 10, 11 y 12 es aplicada a las entradas del CAR. De esta manera, la memoria de control se mantiene en la dirección cero hasta que una variable externa se habilite.

El multiplexor (MUX) tiene cuatro entradas que se seleccionan con los bits 13 y 14 de la microinstrucción. Las funciones de los bits seleccionadas por el multiplexor se tabulan en la figura 5.1. Si los bits 13 y 14 son 00, se selecciona una entrada de multiplexor que es igual a "0". La salida del multiplexor es "0" y la entrada de incremento al CAR se habilita. Esta configuración incrementa el CAR para escoger la siguiente dirección en secuencia. Una salida de "1" es seleccionada por el multiplexor cuando los bits 13 y 14 son

iguales a 01. La salida del multiplexor es "1" y la entrada externa se carga al CAR. La variable de condición S es seleccionada cuando los bits 13 y 14 son iguales a 10. Si S = 1, la salida del multiplexor es 1 los bits de dirección de la micro-instrucción son cargados al CAR.



Bits de ROM 13 14	Función de selección del MUX
0 0	Incrementar el CAR
0 1	Cargar la entrada al CAR
1 0	Cargar las entradas al CAR si S = 1, incrementar el CAR si S = 0
1 1	Cargar las entradas al CAR si E = 1, incrementar el CAR si E = 0

FIGURA 5 . 1 DIAGRAMA DE BLOQUE DE CONTROL DEL MICROPROGRAMA

V.2 SECUENCIA DE LAS DIRECCIONES

Una vez que se establece la configuración de la unidad de control del microprograma, la tarea del diseñador es generar el microcódigo para la memoria de control.

Esta generación de código se llama microprogramación y es un proceso que determina la configuración de bits, para cada una de las palabras en la memoria de control.

El método de transferencia entre registros puede ser adoptado para desarrollar un microprograma. La secuencia de microoperación puede ser especificada con declaraciones de transferencia entre registros. No hay necesidad de listar las funciones de control con las variables de Boole, ya que en este caso, las variables de control son la palabras de control almacenadas en la memoria de control. En vez de una función de control, se especifica una dirección con cada proposición de transferencia entre registros. La dirección asociada con cada proposición simbólica, corresponde a la dirección donde la microinstrucción es almacenada en la memoria. La secuencia de una dirección a la siguiente, puede ser indicada por medio de proposiciones de control condicionales. Este tipo de proposiciones puede especificar una dirección a la cual va el control, dependiendo de las condiciones establecidas. Así, en vez de pensar en términos de "1" ó "0" que deben ser agregados a cada microinstrucción, es más conveniente pensar en términos de símbolos en el método de la transferencia entre registros.

Una vez que se ha establecido el microprograma simbólico, es posible trasladar las proposiciones de transferencia entre registros a su forma binaria.

La designación simbólica, es un método conveniente para desarrollar el microprograma de una manera que el usuario pueda leer y entender. Pero esta, no es la manera como el microprograma es almacenado en la memoria de control. El microprograma simbólico, debe traducirse a binario porque esta es la forma como va a la memoria. La traducción se hace dividiendo los bits de cada microinstrucción en sus partes funcionales, llamadas campos. Aquí se tienen tres partes funcionales, los bits 1 hasta 9 especifican la palabra de control para iniciar las microoperaciones. Los bits 10 hasta 12 especifican un campo de dirección y los bits 13 y 14 seleccionan una entrada del multiplexor. Por cada microinstrucción se deben escoger los bits adecuados, en los campos de microinstrucción correspondientes.

V.3 CONTROL DE LA UNIDAD CENTRAL DE PROCESO (CPU).

En una situación práctica, la organización de los materiales de una unidad de control del microprograma, debe tener una configuración de propósito general para adaptarse a una gran cantidad de situaciones. Una unidad de control de microprograma, debe tener una memoria de control suficiente como para almacenar microinstrucciones. Se debe hacer provisión

para incluir todas las variables de control posibles en el sistema, y no solamente para controlar un ALU. El multiplexor y los bits seleccionados, deben incluir todos los demás bits de condición posibles que se quieran comprobar en el sistema. Se debe tener una provisión para aceptar una dirección externa, para iniciar muchas operaciones en vez de dos operaciones solamente tales como suma y sustracción.

La principal ventaja del control del microprograma, es el hecho que una vez que se ha establecido la configuración de los materiales, no debe haber necesidad de cambios posteriores de las conexiones entre los componentes. Si se quiere establecer una secuencia de control diferente para el sistema, todo lo que se necesita es especificar un conjunto diferente de microinstrucciones para la memoria de control. La configuración con los materiales, no debe cambiar para las diferentes operaciones, el único cambio debe ser el microprograma que reside en la memoria de control.

Una organización del microprograma para controlar la unidad procesadora se muestra en la figura 5.2. Esta tiene una memoria de control de 64 palabras, con 26 bits por palabra. Para seleccionar 64 palabras se necesita una dirección de 6 bits. Para seleccionar 8 bits se necesitan 3 líneas de selección, para el multiplexor un bit de la microinstrucción seleccionada entre una dirección externa y el campo de dirección de la microinstrucción, sumando los 16 bits para seleccionar la microoperación en el procesador, se requiere un

total de 26 bits por cada microinstrucción.

La unidad procesadora se incluye en el diagrama, para mostrar sus conexiones a la unidad de control del microprograma. Los primeros 16 bits seleccionan la siguiente dirección para el control del registro de direcciones.

Los bits de condición del procesador, se aplican a las entradas del multiplexor. Se usan los dos valores; normal y de complemento, excepto para el bit de sobrecapacidad V. La entrada "0" del multiplexor se conecta a una constante binaria la cual es siempre "1". La entrada es seleccionada por medio de los bits 18, 19 y 20 en la microinstrucción. Esto causa una transferencia de información desde la salida del multiplexor "1" al CAR. La entrada al CAR es una función del bit 17 de la microinstrucción. Si el bit 17 es "1", el CAR recibe el campo de dirección de la microinstrucción si el bit 17 es cero se carga una dirección externa al CAR.

Para construir microprogramas correctos, es necesario especificar exactamente cómo el bit de condición es afectado por cada micro-operación en el procesador.

Los bits S (signo) y Z (cero) están afectados por todas las operaciones. Los bits C (arrastre) y V (sobrecapacidad) no cambian después de las siguientes operaciones del ALU:

1. Las cuatro operaciones OR, AND OR-exclusiva y complemento.
2. Las operaciones de incremento y decremento.

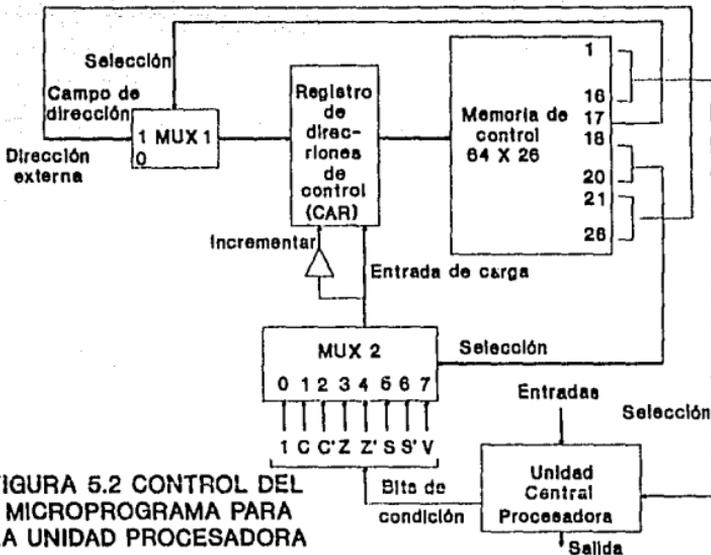


FIGURA 5.2 CONTROL DEL MICROPROGRAMA PARA LA UNIDAD PROCESADORA

V.4 CONTROL DE FORMATOS DE PALABRAS.

La figura 5.3 ilustra la configuración general de la unidad de control de microprograma. La memoria de control se asume como una ROM dentro de la cual se almacena permanentemente toda la información de control.

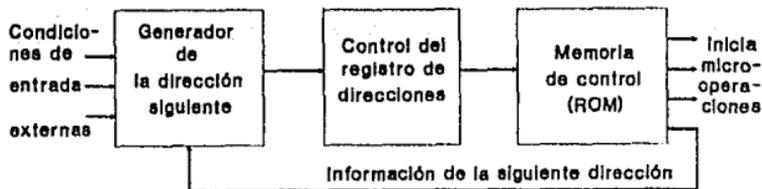


FIGURA 5.3 LOGICA DE CONTROL DEL MICROPROGRAMA

El registro de control de las direcciones de memoria, especifica la palabra de control leída de la memoria de control. Se debe tener en cuenta que una ROM, opera como un circuito combinatorial con el valor de la dirección como entrada, y la palabra correspondiente como salida. El contenido de la palabra especificada, permanece en los alambres de salida por el tiempo que el valor de la dirección permanece en el registro de dirección. No se necesita señal de lectura como en una memoria de acceso aleatorio. Una palabra que sale de la ROM, debe transferirse al registro separador, si el registro de direcciones cambia mientras que la palabra de ROM este aún en uso. Si pueden ocurrir simultáneamente un cambio en direcciones y en una palabra de ROM no es necesario un registro separador.

La palabra leída de una memoria de control representa una microinstrucción. La microinstrucción especifica una o más microoperaciones para los componentes del sistema. Una vez que se ejecuten las operaciones, la unidad de control debe determinar la siguiente dirección. La ubicación de la siguiente microinstrucción, podría ser la siguiente en secuencia o podría ser ubicada en otro lugar en la memoria de control. Por esta razón, es necesario usar algunos bits de la microinstrucción para controlar la generación de la dirección para la siguiente microinstrucción. La siguiente dirección, puede ser también una función de las condiciones de entrada externas. Mientras se ejecutan las microoperaciones, la siguiente dirección es computada en el circuito generador de la siguiente dirección y luego transferida (con el siguiente pulso de reloj) al registro de control de direcciones para leer la siguiente microinstrucción.

V.5 CONSIDERACIONES DE TIEMPO.

Si se ha tomado la decisión fundamental de emplear microprogramación, puede suceder que el diseñador desee producir además una máquina lo más veloz posible. Con este fin, es probable que el diseñador elija el almacenamiento de control más veloz de que disponga, y que utilice largas microinstrucciones a fin de ganar en forma simultánea tantas señales de control como sea posible. Pero existe otra opción:

En vez de seguir la secuencia lógica de traer una microinstrucción, ejecutarla y después traer la siguiente, es posible ahorrar tiempo trayendo antes la siguiente microinstrucción, mientras se está ejecutando la presente. De esta forma, la mayor parte del tiempo de ejecución puede suceder durante el tiempo de traida. Tales técnicas de hacer simultáneamente tantas operaciones como sea posible, para que las unidades de Hardware estén lo más ocupadas que sea posible, a menudo se conoce como "traida anticipada"

V.6 VENTAJAS Y DESVENTAJAS DE LA MICROPROGRAMACION.

VENTAJAS:

La selección de una organización específica, implica tener que equilibrar velocidades de ejecución y costo de implante. También se ve afectada por otros factores tales como la tecnología que se emplee, la flexibilidad de modificación, o el deseo de proporcionar algunas capacidades especiales, en el conjunto de instrucciones de la computadora.

El control microprogramado, proporciona considerable flexibilidad en el implante de conjuntos de instrucciones. También facilita la adición de nuevas instrucciones a máquinas ya existentes.

Cuando se presentó por primera vez el control microprogramado, resultaba mucho más lento que el control fijo, debido a la baja velocidad del almacenamiento ROM. Sin embargo,

los avances en la tecnología ROM han reducido las diferencias de velocidad. Como resultado, ahora el uso del control microprogramado es mucho más frecuente debido a su flexibilidad.

La función principal del control microprogramado, según se analizó antes, consiste en proporcionar un medio de control simple, flexible y relativamente barato para un computadora. Sin embargo, tiene otras posibilidades interesantes. Su flexibilidad en el manejo de los recursos dentro de una máquina, permite implantar diversas clases de instrucciones.

La flexibilidad que puede obtenerse con la microprogramación, posibilita diseñar la programación de rutas de datos del procesador, esto es, la ALU, los registros y las conexiones internas, de manera independiente de su conjunto de instrucciones. Además, es posible diseñar Hardware de secuenciamiento de microinstrucciones de aplicación general, que pueda emplearse junto con una amplia gama de configuraciones de ruta de datos y de conjuntos de instrucciones.

DESVENTAJAS:

El esquema de control microprogramado tiene una seria desventaja. Asignar bits individuales a cada señal de control, hace que las microinstrucciones sean largas, ya que el número de señales necesarias por lo general es bastante grande. La preparación de microprogramas puede resultar una tarea ardua. A menudo resulta difícil lograr la optimización del microcódigo.

En consecuencia, existe una gran necesidad de herramientas para ayuda a vencer estas dificultades. Se ha realizado mucho trabajo en el desarrollo de lenguajes de alto nivel para microprogramación. También, se ha logrado un avance considerable en el desarrollo de técnicas para la optimización del microcódigo.

La principal desventaja del control microprogramado, es la velocidad de operación más lenta inherente a la computadora. La traida de una microinstrucción del almacenamiento de control tarda mucho más, que la producción de señales de control equivalentes por medio de circuitos fijos. Resulta obvio que, las máquinas microprogramadas nunca serán capaces de alcanzar la velocidad de las máquinas de circuitos fijos.

VI. DISEÑO DE UNA COMPUTADORA

Los distintos grupos funcionales, o subsistemas, que debe poseer un ordenador se pueden resumir en los siguientes:

- a) Un sistema de entradas y salidas para establecer la comunicación con el mundo exterior (recepción de órdenes y datos, y entrega de resultados).
- b) Un sistema de memoria para la obtención de instrucciones (programa) y almacenamiento de información (datos).
- c) Una unidad de cálculo para la ejecución de operaciones aritméticas y lógicas.
- d) Una unidad de control capaz de coordinar el funcionamiento de todo el sistema y de tomar decisiones en función de resultados previos.

En la figura 6.1 se presenta en forma de diagrama de bloque la arquitectura típica de un ordenador de utilización general.

Un Microprocesador, es el resultado de la integración a alta escala de la unidad aritmética y lógica (ALU) y el subsistema de control, o gran parte del mismo, en una única unidad de control de proceso (CPU). Un Microcomputador es el ordenador completo que utiliza un Microprocesador. En algunos casos, este elemento base contendrá también parte de otros subsistemas, como líneas de entrada y salida y/o memoria de programa o de datos, siendo el caso límite de la integración de un microcomputador completo en un sólo circuito LSI.

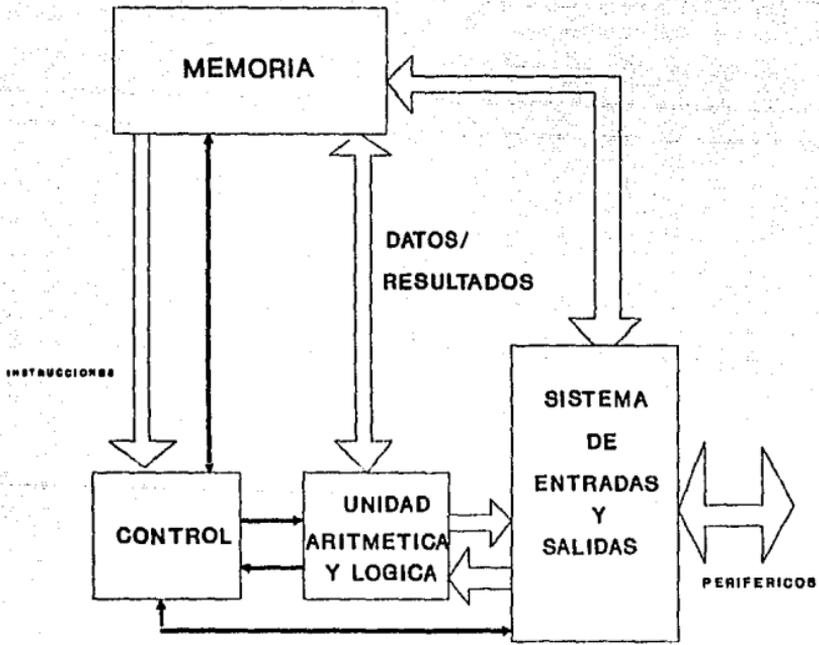


FIGURA 6.1 DIAGRAMA DE BLOQUES DE UN ORDENADOR DE UTILIZACION GENERAL

La arquitectura típica de los sistemas microcomputadores se presenta en la figura 6.2, en la que se muestran los distintos subsistemas enlazados entre sí, por unas únicas vías de comunicación, agrupadas funcionalmente en buses o barras de comunicación.

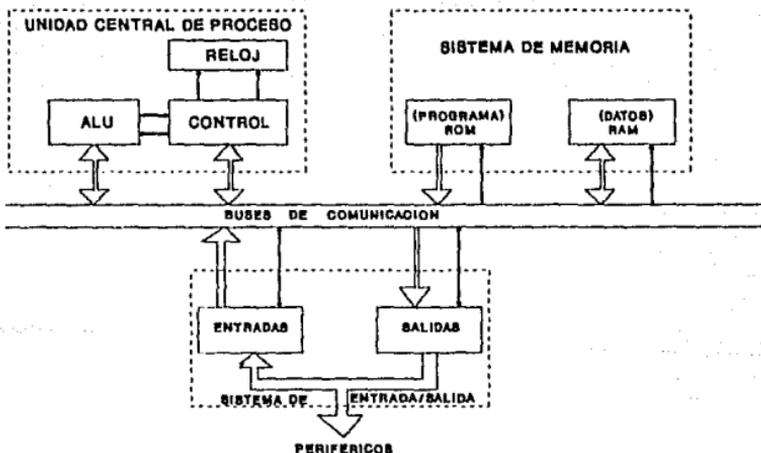


FIGURA 6.2 DIAGRAMA DE BLOQUES DE UN ORDENADOR CON ESTRUCTURA DE BUSES

Existen dos alternativas límites en cuanto a la organización completa de un microcomputador:

1) **ESTRUCTURA ORTOGONAL:** Partición del sistema en subsistemas disjuntos donde cada uno tiene una función específica. De esta forma, es posible aumentar la capacidad de cada subsistema, (ROM, RAM, accesos de entrada o de salida) de una forma independiente. La figura 6.3 muestra este tipo de estructura.

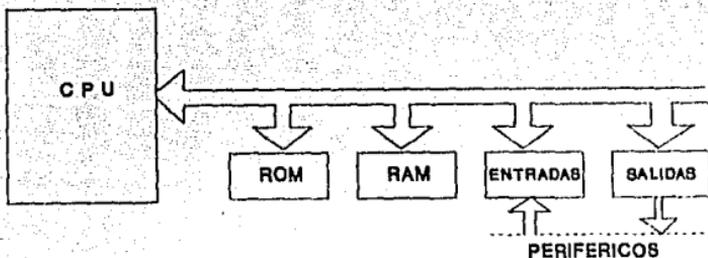


FIGURA 6.3 ESTRUCTURA ORTOGONAL

2) **ESTRUCTURA EMPAQUETADA:** (Packed).- Integración de varias funciones distintas en un mismo circuito integrado. Existen circuitos integrados con dos o más funciones: ROM, RAM, accesos de entrada y salida programables, temporizadores, circuitos de control de interrupciones, etc. De esta forma, el crecimiento de cada subsistema no es independiente. La figura 6.4 muestra este tipo de estructura.

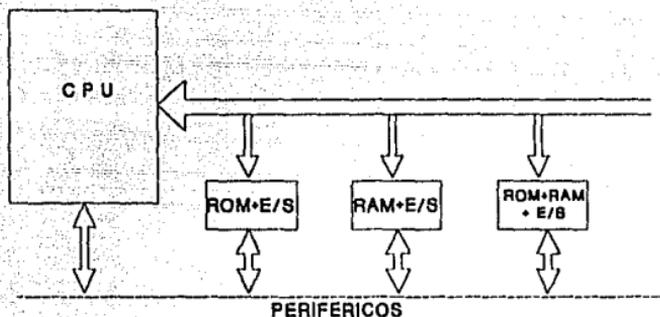


FIGURA 6.4 ESTRUCTURA EMPAQUETADA

En la práctica, los microcomputadores tienen una estructura mixta, tendiendo hacia la estructura ortogonal los sistemas más complejos, con requerimientos de gran capacidad de memoria y los accesos periféricos, y hacia una estructura más empaquetada los sistemas para aplicaciones menos complejas, por la ventaja que supone en la reducción del número de circuitos integrados.

VI.1 CONFIGURACION DEL SISTEMA

Según el tipo de Microprocesador utilizado, se precisará de más ó menos Hardware externo para formar una CPU completa. La tendencia de los modernos Microprocesadores es la de integrar todas las funciones de la CPU en un sólo circuito integrado.

La estructura interna de la unidad central de proceso, presenta variaciones notables al considerar distintos Microprocesadores, por lo que es difícil definir una estructura, no excesivamente simplificada, que se acerque a las distintas alternativas que se han desarrollado.

Además, de la unidad de cálculo, sistema decodificador de instrucciones y circuitos temporizadores, se encuentran en la CPU un conjunto de registros internos, en cantidad y con prestaciones variables según la CPU considerada:

- a) **MAR:** (Memory Address Register) Registro de direcciones. Contiene la dirección de la palabra de memoria (o accesos de E/S) seleccionado para una transferencia.
- b) **PC:** (Program Counter) Contador de programa. Contiene la dirección de la instrucción en curso.
- c) **IR:** (Instruction Register) Registro de Instrucción. Memoriza el código de la operación de la instrucción que se está ejecutando.
- d) **MDR:** (Memory Data Register) Registro de datos de entrada y salida de la CPU.
- e) **AC:** Acumulador. Es el registro base en operaciones aritméticas y lógicas. En general, es también registro fuente y destino de transferencias con memoria y sistema de E/S. Existen máquinas con más de un acumulador.
- f) **SR:** (State Register) Registro de Estado. Esta formado por un conjunto de biestables distribuidos por la CPU, y que se pueden tratar de una forma conjunta y/o indivi-

dualmente. Indican condiciones aritméticas (acarreo, desbordamiento, paridad, etc.) y de estado (máscaras de interrupción, peticiones pendientes, etc.).

g) SP: Puntero de stack. Utilizado para la gestión del stack (pila LIFO) en máquinas que permiten esta estructura.

h) Registros de Utilización General. Son registros internos, accesibles por programa, para el almacenamiento temporal, punteros de memoria, registros de índice, etc.

i) Registros Auxiliares para Uso Interno de la CPU. (Almacenamiento temporal) y no accesibles al programador.

La figura 6.5 muestra el diagrama de bloque de una computadora digital.

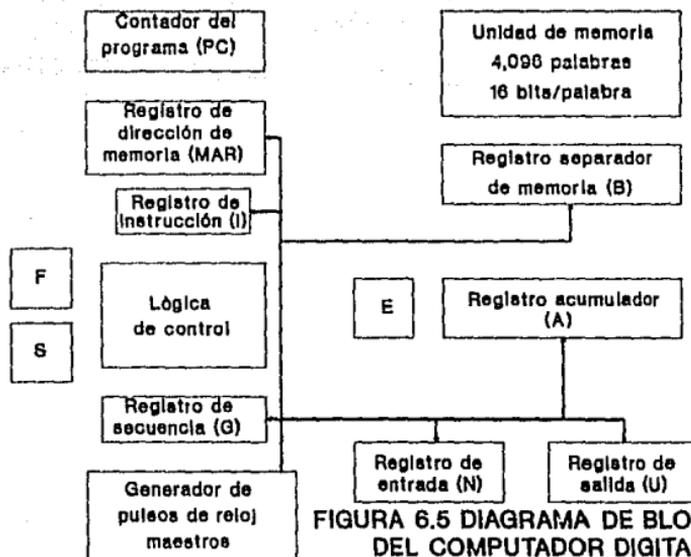


FIGURA 6.5 DIAGRAMA DE BLOQUE DEL COMPUTADOR DIGITAL

VI.2 INSTRUCCIONES DE LA COMPUTADORA.

El repertorio de instrucciones es muy variable en función del Microprocesador utilizado. Existe una gran diferencia entre los primeros Microprocesadores, con un conjunto de instrucciones muy limitado, y los modernos, con un conjunto de instrucciones potentes, cercano o superior al de muchos microcomputadores.

Por la función realizada, las instrucciones típicas de un Microprocesador se pueden dividir en varias categorías:

- 1) *Instrucciones de Transferencia:* Movimiento de datos entre registros. Los registros fuente y destino de la transferencia pueden pertenecer a la CPU (acumulador y registros auxiliares), a memoria o al sistema de E/S.
- 2) *Instrucciones Operativas:* Ejecución de operaciones lógicas o aritméticas entre dos operandos (suma, resta, multiplicación, etc.), sobre un sólo operando (incremento, complemento, borrado, etc.) o bit a bit (Y, OR, OR exclusiva, puesta a uno, complemento, etc.).
- 3) *Instrucciones de Control de Secuencia:* Saltos de programa y llamadas, y retornos de subrutinas, incondicionales o condicionales.
- 4) *Instrucciones de Control:* Instrucciones que afectan al registro de estado, sistema de interrupciones, cambio de bancos de registros, paro del sistema, instrucciones especiales de E/S, etc.

Algunos sistemas son microprogramables, lo cual posibilita un repertorio de instrucciones arbitrario.

Existen distintas formas de direccionar los registros, palabras de memoria y accesos de entrada y salida que intervienen en la ejecución de una instrucción. Las posibilidades de direccionamiento de un Microprocesador son importantes de tener en cuenta, por su repercusión en la velocidad de ejecución, flexibilidad de la programación y necesidades en la capacidad de memoria.

Una instrucción se puede considerar dividida en dos partes: Código de operación y operando (si es necesario). El operando, si existe, ocupa palabra(s) sucesiva(s) al código de operación. La información de la dirección se puede presentar de distintas formas:

- a) **Direccionamiento Implícito:** Cuando el mismo código de operación, indica la dirección del operando. Por la limitación que este método supone, en cuanto a capacidad, su utilización es restringida al direccionamiento de registros internos de la CPU y casos especiales de control de secuencia.
- b) **Direccionamiento Inmediato:** Cuando el operando no representa una dirección sino que es uno de los datos de la operación.
- c) **Direccionamiento Directo:** Cuando se indica de una forma absoluta, la dirección efectiva del operando o la dirección de salto, en caso de instrucciones de control de secuencia.

- d) **Direccionamiento Indirecto:** Cuando el operando indica una dirección donde está la dirección efectiva del operando. Puede ser posible un encadenamiento de direcciones hasta llegar a la dirección efectiva. Las direcciones pueden ser respecto a un registro interno, punteros de memoria o de stack, o respecto a memoria, según donde esté el puntero de la dirección efectiva.
- e) **Direccionamiento Relativo al Contador de Programa:** El operando indica un desplazamiento, positivo o negativo, respecto a la dirección de la instrucción en curso.
- f) **Direccionamiento Indexado:** El operando indica un desplazamiento respecto a un puntero de memoria en la CPU (registro de índice).
- g) **Direccionamiento por Página:** Cuando el operando indica la dirección efectiva en la página (memoria dividida en módulos de 256 palabras) en curso o respecto a la página cero (posiciones 0 a 255).

VI.3 CONTROL Y TIEMPO.

Todos las operaciones del computador, están sincronizadas por un generador de tiempo maestro, cuyos pulsos de reloj se aplican a todos lo flip-flops del sistema. Además, esta disponible cierto número de variables de tiempo en la unidad de control para darle secuencia a la operación en el orden adecuado.

Esas variables de tiempo se designan como t_0 , t_1 , t_2 y t_3 y se muestran en la figura 6.6. Los pulsos de reloj ocurren una

vez cada microsegundo (μs). Cada variable de tiempo es de $1 \mu s$ de duración y ocurre una vez cada $4 \mu s$. Se asume que el disparo de los flip-flops, ocurre durante el flanco negativo de los pulsos de reloj. Se puede controlar el pulso de reloj específico, que dispara el registro, aplicando una de las variables de tiempo al terminal de entrada de habilitación de un registro dado. Las variables de tiempo se repiten continuamente de manera que t_0 aparezca después de t_3 . Cuatro variables de tiempo, son suficientes para la ejecución de cualquier instrucción en el computador que consideramos aquí. En otras situaciones podría ser necesario emplear un número diferente de variables de tiempo.

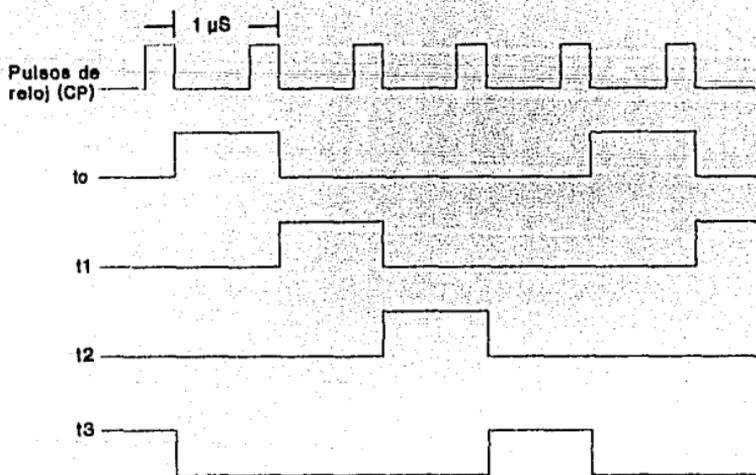


FIGURA 6.6 SEÑALES DE TIEMPO DEL COMPUTADOR

Se asume que el tiempo de acceso de memoria es menor que 1 μ s. Una operación de lectura o escritura de memoria puede iniciarse con una de las variables de tiempo cuando ésta se ponga alta. La operación de memoria se completará en el momento en que llegue el pulso siguiente de reloj.

El computador digital opera con pasos discretos controlados por las señales de tiempo, una instrucción leída de la memoria y ejecutada en los registros por medio de una secuencia de microoperaciones. Cuando el control recibe una instrucción, este genera las funciones de control adecuadas para las microoperaciones requeridas. En la figura 6.7 se muestra un diagrama de bloque de la lógica de control. Una instrucción que se lee de la memoria se coloca en el registro B separador de la memoria. La instrucción tiene un código de operación de 4 bits, designado por el símbolo OP. Si ésta es una instrucción de referencia de memoria tendrá una parte de dirección designada por símbolo AD. El código de operación se transfiere siempre al registro de instrucción I. El código de operación en I se decodifica en ocho salidas q_0 - q_7 , siendo el número suscrito igual al código hexadecimal para la operación. El registro G es un contador de 2 bits que cuenta continuamente los pulsos de reloj, durante el tiempo en el flip-flop S, de comienzo-parada, esté puesto a uno. Las salidas del registro G se decodifican en cuatro variables de tiempo t_0 - t_3 . El flip-flop F distingue entre los ciclos de búsqueda y de ejecución.

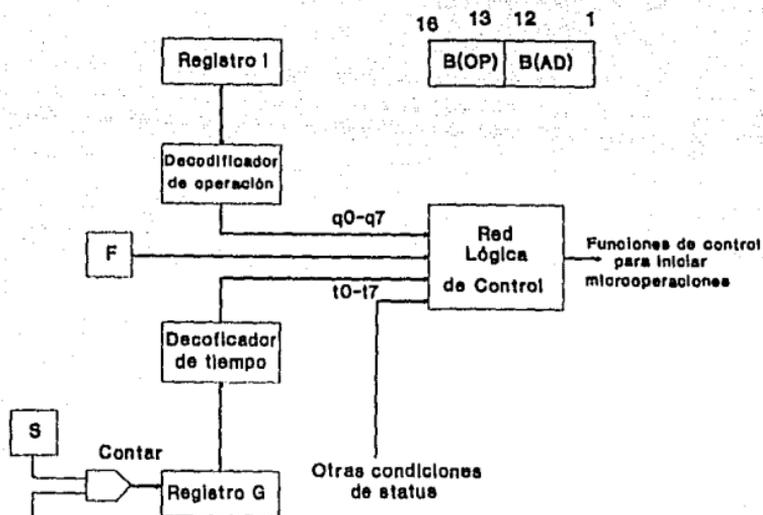


FIGURA 6.7 DIAGRAMA DE BLOQUE DE LA LOGICA DE CONTROL

VI.4 EJECUCION DE INSTRUCCIONES.

Se ha especificado la configuración del registro, el conjunto de instrucciones del computador, una secuencia de tiempo y la configuración de la unidad de control. En esta sección, se comienza con la fase de diseño del computador. El primer paso, es especificar las microoperaciones conjuntamente con las funciones de control necesarias para ejecutar cada instrucción de máquina.

Cuando en un microcomputador se realiza una transferencia con el sistema de E/S, ya sea de tipo elemental (palabra) o de bloques de información se deberá disponer de algún sistema que

sea capaz de iniciar, realizar y controlar el flujo de información.

Respecto a la forma de ejecución de la transferencia cabe distinguir entre dos alternativas diferenciadas:

1) *Transferencia por Programa:* Cuando el intercambio de información se realiza vía registros de la CPU, por medio de instrucciones de transferencia de datos con el sistema de E/S.

2) *Transferencia de Acceso Directo a Memoria:* Cuando se inhibe a la CPU de la tarea de control, realizándose la transferencia bajo control de un sistema externo (controlador de DMA) de forma que la comunicación se establece directamente entre memoria y periférico.

En caso de ejecutarse la transferencia por programa, se pueden considerar dos posibilidades en cuanto a la forma de control ó estrategia utilizada:

1) *Transferencia Síncrona:* Cuando se realiza por iniciativa del programa y sin consultar al periférico sobre su estado. Este método es el más sencillo en cuanto a necesidades de Software y Hardware (figura 6.8) precisas, pero su utilización estará restringida a la gestión de aquellos periféricos que siempre estén dispuestos a aceptar una nueva información (ejemplo: un visualizador alfanumérico) o a entregar los datos que se le piden.

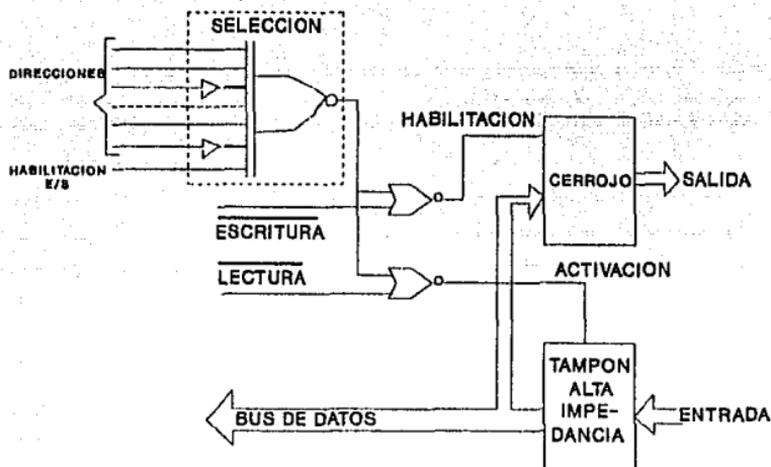


FIGURA 6.8 INTERFASE PARA UNA TRANSFERENCIA SINCRONA EN PARALELO

2) *Transferencia Asíncrona:* Cuando es preciso establecer un diálogo (Handshaking) entre CPU y periférico, a fin de controlar la ejecución de la transferencia en el instante y de la forma apropiadas. En este caso, cabe considerar dos alternativas en cuanto a la forma de control:

a) *Por Consulta de Estado:* Se pregunta al periférico sobre su disposición para aceptar la transferencia o su necesidad de servicio.

b) *Por Interrupción:* Se delega en el periférico, la facultad de avisar cuando está dispuesto o si precisa servicio.

En el método de transferencia síncrona se deberán establecer mecanismos de sincronismo del transvase de información entre microcomputador y periférico. En muchos, casos esta necesidad viene motivada por las diferencias relativas, en cuanto a velocidad, entre procesador (rápido) y periférico (lento).

VI.5 DISEÑO DE CONTROL.

La unidad de control del computador, genera las variables de control para los registros y unidad de memoria. En el capítulo III, se presentaron tres métodos para el diseño de la lógica de control: El control con componentes alambrados, el control PLA y el control del microprograma. La unidad de control del computador puede ser diseñada usando cualquiera de estos tres métodos.

* **CONTROL CON COMPONENTES ALAMBRADOS:** La organización de control presentado en la figura 6.6 es esencialmente una organización con componentes alambrados.

El registro G, en este caso, es un computador y el decodificador de tiempo entrega cuatro estados de control para el sistema . Un segundo decodificador se usa para el código de operación almacenado en el registro I.

El bloque de la red de lógica de control genera todas las funciones de control por el computador.

La configuración de la red de lógica de control de la figura 6.6, completa el diseño del control con componentes alambrados. Esta configuración, consiste de las compuertas combinacionales que generan las 24 funciones listadas en la tabla 6.1. Las funciones de Boole listadas como funciones de control, especifican las ecuaciones de Boole de las cuales se puede deducir el circuito combinacional.

* CONTROL PLA: El control PLA es similar al anterior, excepto que todos los circuitos combinacionales se configuran dentro del PLA. Los dos decodificadores se incluyen dentro de la configuración del PLA, ya que ellos son circuitos combinacionales. El número de salidas de control es 24.

La figura 6.8 muestra tres PLA y dos registros para la unidad de control. Los dos decodificadores no son necesarios aquí, ya que se configuran dentro del PLA. Nótese, que no hay conexiones de las salidas de cualquier PLA a las entradas del registro de secuencia G. Una conexión de realimentación, no es necesaria, porque el registro G es un contador y el siguiente estado se puede determinar a partir de la secuencia de cuenta continua.

Memoria de control

$R = F' t_1 + f(q_0 + q_1 + q_3) t_3$: $B \leftarrow M$ Leer de memoria
 $W = F(q_2 + q_3 + q_4) t_3$: $M \leftarrow B$ Escribir en memoria

Registro A

$a_1 = Fq_0 t_3$: $A \leftarrow B$ AND
 $a_2 = Fq_1 t_3$: $A \leftarrow +B$ Sumar
 $a_3 = rB_1 t_2$: $A \leftarrow 0$ Borrar
 $a_4 = rB_1 t_2$: $A \leftarrow A$ Complementar
 $a_5 = rB_0 t_2$: $A \leftarrow shr A, A_1, 0 \leftarrow E$ Desplazamiento a la derecha

$a_6 = rB_7 t_2$: $A \leftarrow shl A, A_1, 0 \leftarrow E$ Desplazamiento a la izquierda

$a_7 = rB_0 t_2$: $A \leftarrow A + 1$ Incrementa
 $a_8 = pB_{11} t_2$: $A_1 \leftarrow 0 \leftarrow N_{14} \leftarrow 0$ Transferencia

Registro B

$b_1 = Fq_2 t_2$: $B \leftarrow A$ Transferencia
 $b_2 = Fq_3 t_2$: $B \leftarrow B + 1$ Incremento
 $b_3 = Fq_4 t_2$: $B(AD) \leftarrow PC; B(OP) \leftarrow 0101$ Transferencia

Registro PC

$c_1 = F' t_1$
 $+ (q_3 B_2 + q_4) Ft_3$
 $+ (B_3 A'_{10} + B_4 A_{10} + B_5 A_2 + B_2 E') r$
 $+ (B_{12} N_{14} + B_{10} U_{14}) p$: $PC \leftarrow PC + 1$ Incremento
 $c_2 = q_0 t_3$: $PC \leftarrow B(AD)$ Transferencia
 $c_3 = Fq_4 t_2$: $PC \leftarrow MAR$ Transferencia

Registro MAR

$d_1 = F' t_0$: $MAR \leftarrow PC$ Transferencia
 $d_2 = Ft_0$: $MAR \leftarrow B(AD)$ Transferencia

Registro I

$i_1 = F' t_2$: $I \leftarrow B(OP)$ Transferencia

Flip-flop E

$e_1 = rB_{11} t_2$: $E \leftarrow 0$ Borrar
 $e_2 = rB_{10} t_2$: $E \leftarrow \bar{E}$ Complementar
 $a_2 = Fq_1 t_3$: $E \leftarrow arrastre$ Transferencia
 $a_5 = rB_0 t_2$: $E \leftarrow A_1$ Desplazamiento a la derecha
 $a_6 = rB_7 t_2$: $E \leftarrow A_{10}$ Desplazamiento a la izquierda

Flip-flop F

$f_1 = F'(q_0 + q_1 + q_2 + q_3 + q_4) T_3$: $F \leftarrow 1$ Poner a 1
 $f_2 = Ft_3$: $F \leftarrow 0$ Borrar

Flip-flop S

$s_1 = rB_1 t_2$: $S \leftarrow 0$ Borrar

Registro G

S : $G \leftarrow G + 1$ Contar

Registro U

$u_1 = pB_{11} t_2$: $U_{14} \leftarrow A_{14} \leftarrow 0, U_{14} \leftarrow 0$ Transferir

Registro N

$a_8 = pB_{11} t_2$: $N_{14} \leftarrow 0$ Borrar

TABLA 6.1 MICROOPERACIONES PARA LOS REGISTROS

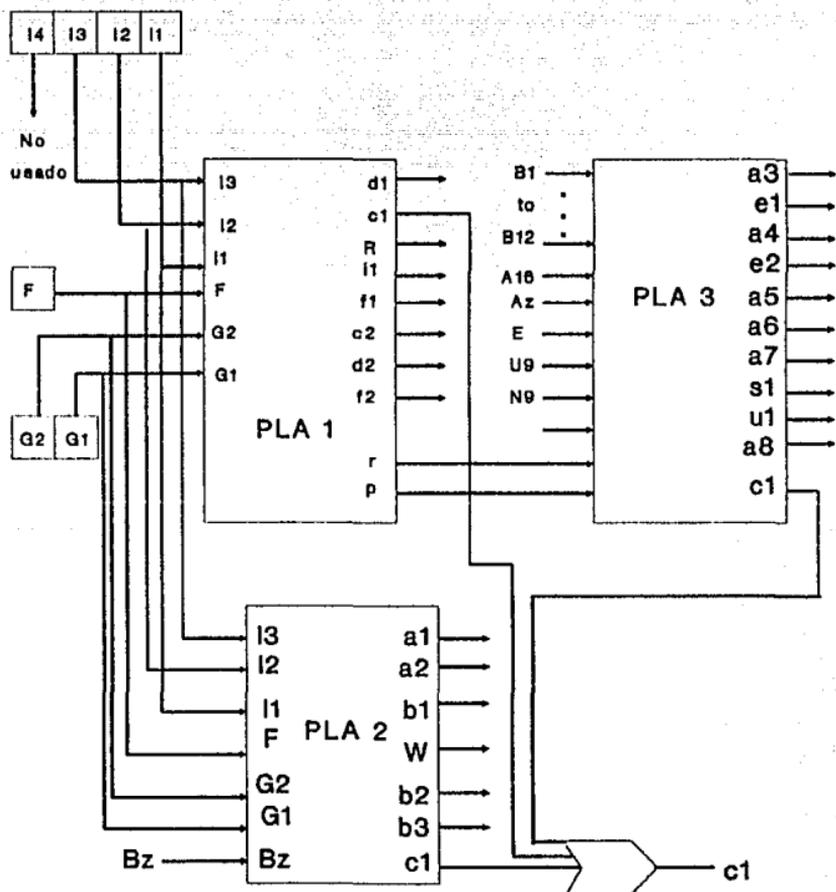


FIGURA 6.9 CONTROL PLA PARA COMPUTADOR

* **CONTROL DEL MICROPROGRAMA:** La organización de la unidad de control para el computador, es más adecuada para el control del PLA que para el control del microprograma, principalmente por la forma como fueron formuladas originalmente las instrucciones de referencia entre registros. La configuración del control del microprograma que se va a desarrollar aquí, configura las funciones de control para el ciclo de búsqueda y las instrucciones de referencia de memoria. Las operaciones de referencia de registro de entrada-salida, pueden configurarse más eficientemente con un control de compuertas interconectadas o un control PLA.

VI.6 DISEÑO DE REGISTRO.

El diseño de un sistema digital síncronico, sigue un procedimiento prescrito. A partir del conocimiento de las necesidades del sistema, se formula una red de control y se obtiene una lista de operaciones de transferencia entre registros del sistema. Una vez que se haya derivado esa lista, el resto del diseño es directo. Algunas instalaciones utilizan técnicas de automatización para el diseño de computador, para traducir las proposiciones de transferencia entre registros a un diagrama de circuitos compuestos de circuitos integrados. Para determinar el tipo de terminal de control que se debe tener en cada registro, es necesario obtener una lista de microoperaciones que afectan cada registro separadamente. Esto

se aplica también a las operaciones de lectura y escritura en la unidad de memoria.

La lista de microoperaciones dadas en la tabla 6.1 suministran la información necesaria para diseñar los registros del computador. Las operaciones que se van a realizar en cada registro, se demuestran claramente en las suposiciones listadas.

El diagrama de bloque que muestra los tipos de registros necesarios para el computador, se da en la figura 6.9, la unidad de memoria se incluye también para mostrar su conexión al procesador. La lógica de control presenta todas las variables de control de los registros. Las variables de control que se generan en la unidad de control, son aplicadas a los registros de la manera que se indica en el diagrama. Además de los registros, el procesador usa cuatro multiplexores para seleccionar de dos o más fuentes. Todos los registros y multiplexores son funciones MSI disponibles en circuitos integrados normales. Los tres flip-flops E, F y S y su correspondiente lógica combinacional, debe diseñarse con compuertas SSI y con flip-flops.

Todos los registros en el computador, excepto el registro A, requieren terminales de entrada de control de carga, incremento o de carga, e incremento juntos. Se puede escoger el uso de un contador MSI con carga en paralelo para todos los registros.

El registro A es el más complejo, porque realiza todas las tareas de procesamiento del computador. Este registro es un registro acumulador.

El registro de entrada N y el registro de salida V pueden ser parte de la interconexión normal de una teleimpresora. Los circuitos integrados que hacen interconexión con una unidad teleimpresora, están disponibles comercialmente y se les llama a menudo transmisores-receptores asincrónicos universales (UART).

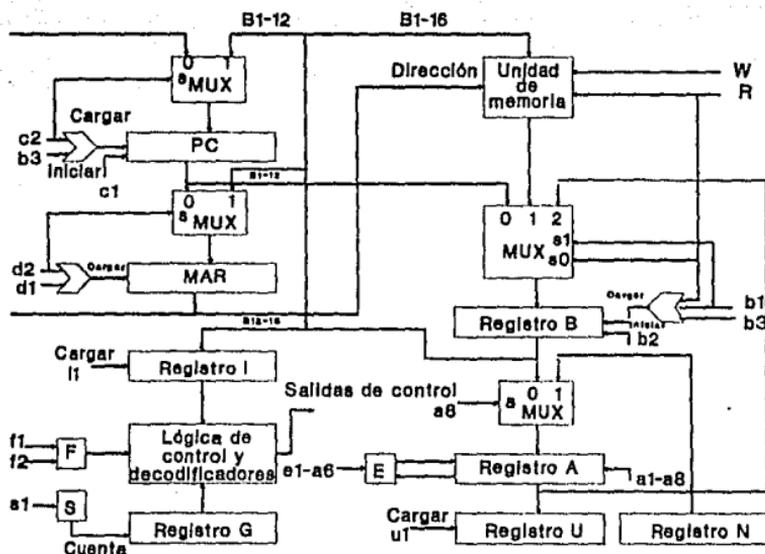


FIGURA 6.10 DIAGRAMA DE BLOQUE DETALLADO DEL COMPUTADOR

VI.7 DISEÑO DE LA CONSOLA.

Cualquier computador tiene un panel de control o consola con interruptores y lámparas, para permitir comunicación manual y visual entre el operador y el computador. Esta comunicación es necesario para comenzar la operación del computador (bootstrapping), y para propósitos de mantenimiento. Para completar, se enumera un conjunto de funciones de la consola útiles para el computador, aunque no se muestran los circuitos necesarios para configurar estas funciones.

Las lámparas indican al operador la condición de los registros del computador. La salida normal de un flip-flop conectado a una lámpara indicadora, causará que la lámpara alumbré cuando el flip-flop se pone a "1" y se apague cuando el flip-flop se borra. Los registros cuyas salidas van a ser observadas en la consola del conmutador son: A, B, PC, MAR, I, E, F y S. Cuando se muestra el número total del flip-flop se encuentra que es necesario 63 lámparas indicadoras.

Un conjunto de interruptores y sus funciones para la consola pueden incluir lo siguiente:

1. Dieciséis interruptores de palabra, para establecer manualmente los bits de una palabra.
2. Un interruptor de "comienzo" para preparar el flip-flop S. La señal de este interruptor borra el flip-flop F y el registro G.

3. El interruptor de "parada" para borrar el flip-flop S. Para asegurar que se complete esta instrucción, la señal que viene del interruptor se aplica conjuntamente con la función de Boole $(F + q_5 + q_6 + q_7 + q_8)$ a una compuerta AND antes de que se aplique para el borrado de la compuerta S.

4. Un interruptor de "carga-dirección" para transferir una dirección al registro PC. Cuando se activa este interruptor, el contenido de los 12 interruptores de "palabra" se transfiere al PC.

5. El interruptor de "depósito" para almacenar palabras en la memoria. Cuando se activa este interruptor, el contenido de PC se transfiere al MAR y se inicia el ciclo de memoria. Después de $1 \mu s$, el contenido de los 16 interruptores de "palabra" se transfiere al registro B y se incrementa el PC en 1.

VII. ORGANIZACION DE ENTRADA/SALIDA

El sistema de entrada y salida permite la comunicación del microcomputador con el mundo exterior.

Debido al tipo de aplicaciones, en general, en las que son utilizados los Microprocesadores, la capacidad y eficiencia en el tratamiento del sistema de E/S es una de las características principales de un sistema microcomputador.

Se denomina Interface al sistema Hardware-Software, que permite la comunicación con un periférico determinado, es decir, el conjunto de circuitos (Hardware) y programas (Software) que se utilizan para establecer la comunicación.

La forma concreta de realizar una interface, dependerá de las alternativas que se consideren. En principio, dentro del balance "Hard-Soft" se deberá potenciar el Software ya que, en general, el incremento en coste de memoria es inferior al incremento del Hardware preciso para realizar la misma función, además de la mayor flexibilidad que permite el Software. En otros casos, esta afirmación no estará justificada ya que el coste de programación puede ser decisivo. Por otra parte, las tareas de entrada y salida pueden consumir excesivo tiempo de máquina, por lo que puede ser precisa la utilización de circuitería externa, compleja y especializada, a fin de posibilitar la ejecución de estas tareas u otras de proceso que no se podrían ejecutar si el Microprocesador tuviese que hacerse cargo de toda la gestión y control de las operaciones

de transferencia.

Existen dos tipos de información en la comunicación Microprocesador-Periférico:

1) **DATOS:** Entrada de información para proceso y salida de resultados.

2) **CONTROL:** Salida de señales para el gobierno de los periféricos de entrada de información del estado de los mismos.

La flexibilidad en el tratamiento de esta información dependerá de varios factores:

a) Instrucciones del Microprocesador aplicables al sistema de E/S.

b) Posibilidad de delegación de las tareas de control de la transferencia, a circuitos de adaptación externos al Microprocesador y controladores de periféricos.

c) Técnicas de transferencia utilizadas. Las funciones que deberá realizar el sistema de interface o adaptación son:

- **Identificación de Direcciones:** A fin de establecer la conexión con el bus de datos y control cuando un acceso concreto de E/S es seleccionado.

- **Interpretación de Ordenes:** En general, las ordenes enviadas directamente al sistema de E/S por el Microprocesador, se reduce a señales de escritura y lectura; ya decodificadas o que precisan un pretratamiento sencillo. En otros casos, la estructura E/S es más compleja y se envían a los periféricos señales de control a partir de las cuales se debe interpretar

la operación a realizar (lectura, uscritura, borrado, test, activación, etc.).

- *Adaptación Física Entre Dos Sistemas, Microcomputador y Periférico:* Esta función es realizada por los transmisores y receptores de líneas, convertidores tensión-corriente y viceversa, optoisoladores, etc. Si la transmisión de los datos se realiza en serie, será preciso considerar, además, la conversión de formato de la información en ambos sentidos.

- *Temporización de la Transferencia a Fin de Controlar el Flujo de Información Entre Sistemas de Forma Ordenada y eficaz:* En esta función intervienen los protocolos de comunicación y las técnicas de transferencia.

VIS.1 REPASO DE DISPOSITIVOS PERIFERICOS.

Los periféricos se utilizan en un sistema microcomputador, para establecer la comunicación con el mundo exterior. Debido al incremento en los tipos de aplicaciones en los que se utilizan, y a la evolución tecnológica que ha permitido, el universo de los periféricos se ha ido desarrollando con gran rapidez, en parte, motivado precisamente por las necesidades crecientes de disponer de periféricos aptos para su utilización en sistemas basados en Microprocesadores.

Al considerar el tema de los periféricos adecuados para microcomputadores cobran una mayor importancia algunas de sus características, como:

a) **Costo:** En muchos casos el costo de un periférico y de la circuitería asociada para su control puede representar una parte muy significativa del costo total del sistema, e incluso ser superior a todo el resto, por lo que se hace más importante todavía, el poder disponer de periféricos baratos y sistemas de control sencillo, o la disponibilidad de circuitos LSI especializados para el control del periférico.

b) **Tamaño y Peso:** Muchos sistemas basados en Microcomputadores (terminales, equipo de instrumentación) son portátiles, por lo que el volumen y peso del equipo es un factor determinante.

c) **Consumo:** Característica bastante ligada al tamaño y peso del equipo, por el consumo asociado a las partes electromecánicas del mismo. Puede tener mucha importancia a la hora de diseñar las fuentes de alimentación, compactas y baratas.

d) **Fiabilidad:** Por la forma de operación de muchos sistemas microcomputadores, en comunicación continua con el mundo real, es importante disponer de equipos periféricos con índices de fiabilidad lo más próximos posibles al del propio microcomputador.

Los tipos de periféricos utilizados en sistemas microcomputadores son muy variados en función de la aplicación a la que son destinados, haciéndose difícil una enumeración exhaustiva de todos ellos.

En la figura 7.1 se pretende hacer una clasificación, bajo el criterio del elemento dialogante con el microcomputador.

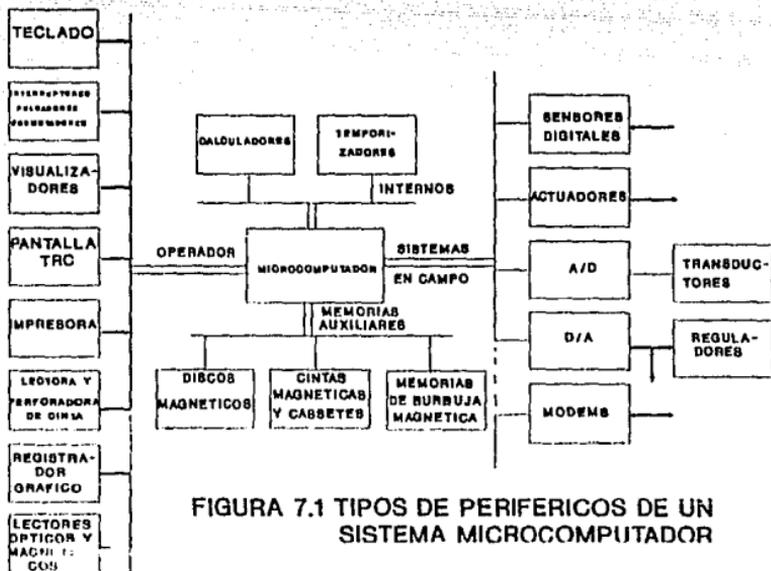


FIGURA 7.1 TIPOS DE PERIFERICOS DE UN SISTEMA MICROCOMPUTADOR

A continuación se consideran algunos periféricos:

1) *Periféricos Internos:* Son subsistemas del microcomputador, que no son en realidad periféricos, aunque pueden ser considerados como tales. Entre estos, se podrían destacar los circuitos utilizados para aumentar la potencia aritmética del sistema, como multiplicadores monolíticos y generadores de funciones, y los temporizadores programables que permitan la realización por Hardware, tales como el conteo de sucesos

TESIS CON
FALLA DE ORIGEN

externos, gestión de un reloj diario, reloj de tiempo real, etc.

2) *Interruptores, Pulsadores y Conmutadores:* Permiten entrar datos al sistema, seleccionar diversas opciones o modos de funcionamiento, lanzar ordenes de actuación, de una forma manual. La forma de establecer el contacto depende del tipo considerado: Contactos metálicos, de lámina, aceptación óptica, efecto capacitivo, magnéticos, por efecto Hall, etc.

3) *Teclados:* Es una forma de disponer de un conjunto de pulsadores, de forma compacta, para entrar información alfanumérica y de control al sistema. Presenta ventajas en cuanto a flexibilidad de operación, sencillez de manejo, coste y estética se refiere.

4) *Visualizadores:* Dan al operador de una forma visible, información sobre el sistema, pudiéndose realizar por diversos métodos ópticos (diodos emisores de luz, cristal líquido, incandescencia, fluorescencia) y electromecánicos. La información puede ser de tres tipos, según los visualizadores sean:

- *Binarios:* Indicadores individuales.
- *Númericos:* Permiten visualizar los dígitos de 0 a 9 y algunas letras.
- *Alfanuméricos:* Permiten visualizar todo el alfabeto y caracteres especiales.

Por la forma de control puede ser estáticos (información fija) y multiplexores (refresco periódico).

5) **Pantalla TRC:** Consiste en la incorporación de un sistema de control, a un tubo de rayos catódicos para disponer de un elemento de visualización alfanumérica. Un terminal TRC está constituido por una pantalla y teclado, y el Hardware/Software preciso para que su utilización sea flexible y sencilla.

6) **Impresoras:** Las impresoras más utilizadas en sistemas microcomputadores, son las de media y baja velocidad, numéricas o alfanuméricas y, en general, de escritura carácter a carácter. Para aplicaciones que precisen la escritura rápida de largos listados, se puede disponer de impresoras de líneas. Por el modo de escritura, pueden ser electromecánicas (de tambor, de cadena, de bola, de margarita, de matriz de agujas, térmicas, etc.). Algunas impresoras permiten también la generación de gráficas por puntos o continuas (registradores y plotters).

7) **Lecturas y Perforadoras de Cinta:** Son periféricos que permiten la lectura y escritura de información tomando como soporte físico una cinta perforada de papel o plástico. Son periféricos relativamente lentos, con velocidades de perforación de hasta 300 caracteres/seg y lectura del orden de 1000 caracteres/seg para lectoras ópticas y de efecto capacitivo.

8) **Memorias Auxiliares (de Masa):** Permiten disponer de un medio de memoria de gran capacidad a bajo costo. De los sistemas habituales, los discos rígidos y cintas magnéticas raramente son utilizados en sistemas microcomputadores,

empleándose ambos los discos flexibles (floppy disk) y cassettes digitales. El disco flexible es un sistema de relativamente bajo costo y bastante fiable.

9) Modems: Son sistemas para la conversión y reconversión de señales digitales en analógicas, de forma que se pueda enviar la información, en serie, a largas distancias por medio de líneas de transmisión. Será por tanto preciso el disponer de un sistema modulador/demodulador (modem) en cada terminal de línea.

VII.2 INTERFACES DE E/S

Varias familias de microcomputadores, disponen de circuitos integrados LSI compatibles con el resto de los elementos del sistema, a fin de evaluar el sistema de entrada/salida.

Las características fundamentales de estos subsistemas son:

1) **Facilidad de Interconexión a los Buses de Comunicación:** Por ser elementos compatibles con el resto del sistema, en cuanto a señales y temporizaciones, es muy fácil conectarlos a los buses, requiriendo en general, solamente un sistema de decodificación de direcciones.

2) **Posibilidad de Programación:** De esta forma, se logra disponer de un cierto número de opciones de funcionamiento. En general, la programación consiste en cargar de una forma determinada un registro interno de modo de operación, lo cual da, además, la posibilidad de una reprogramación dinámica, es

decir, en fase de operación. Esta posibilidad no siempre es utilizada, pero una programación inicial que permita adaptar el sistema a las necesidades particular de cada aplicación, amplia los campos de utilización de un determinado circuito, dando como resultado una modularidad y reducción en costo en comparación con el desarrollo de circuitos especiales para cada configuración.

3) *Inteligencia Distribuida:* La posibilidad de que estos circuitos posean una cierta capacidad de decisión para desarrollar funciones de control, permite descargar al Microprocesador de estas tareas, simplificando de esta forma el programa y/o posibilitando determinadas prestaciones del sistema que serían imposibles de conseguir, si la unidad central tuviera que hacerse cargo de toda la gestión del sistema de E/S.

Por la función que realizan, se pueden dividir en tres categorías:

1) *Interfaces Dedicadas:* Son circuitos especializados para una función específica, aunque en general permiten un cierto número de opciones. En esta categoría, se incluyen temporizadores, controladores de sistema (interrupciones y acceso directo a memoria) y controladores de periféricos (disco flexibles, pantallas, teclados, protocolos, etc.).

2) *Interfaces Generales:* Son circuitos de tipo más universal que los anteriores, y tienen como misión posibilitar la

transferencia y dialogo entre el Microprocesador y los perifericos.

Funcionalmente existen dos tipos distintos:

a) Interfaces para Transmisión en Paralelo: Permiten disponer de un numero elevado de lineas de entrada y salida, que permitiran el intercambio de informacion, asi como el envio de ordenes y recepcion de estado de una forma sencilla, flexible y rapida.

b) Interfaces para Transmisión en Serie: Permiten utilizar canales de transferencia serie de un modo transparente al Microprocesador, es decir, se descarga a la unidad central de las tareas de serialización y sincronización. Pueden ser sincronas y/o asincronas según permitan un modo u otro de transmisión.

3) Microcomputadores Especializados de Entrada y Salida: En principio, cualquier Microprocesador puede considerarse a modo de "esclavo" de otro, realizando tareas bajo control del "director". Estas tareas pueden ser cualesquiera y por tanto la gestión de entradas y salidas es una de ellas.

VII.3 INTERFACE ASINCRONA SERIE.

Para posibilitar este tipo de diálogo, será preciso que el periférico disponga de un registro de estado que suministre la información adecuada (libre, dato válido, etc.) a requerimiento del Microprocesador.

En el caso de control por consulta de estado, cabe considerar dos formas de establecerla (figura 7.2):

1) *Método de Bloqueo de Programa:* El procesador queda en bucle de espera, consultando continuamente el estado del periférico, en caso de que se precise efectuar una transferencia, con este método, y según sea el período de respuesta del periférico, se puede perder mucho tiempo consultando el estado, con lo que puede quedar considerablemente disminuida la potencia del sistema. En aplicaciones en tiempo real, este sistema, sin modificaciones, tiene una aplicación muy restringida.

2) *Método de Consulta Periódica:* Se interroga periódicamente la necesidad y posibilidad de efectuar una transferencia. De esta forma no se penaliza el tiempo de proceso, pero no se soluciona el problema de que, debido a que la consulta se realiza por iniciativa del programa, puede ocurrir que el tiempo existente entre el instante en que se precisa la transferencia, y la ejecución de la misma sea superior al intervalo dentro del cual el periférico necesita el servicio.

A fin de efectuar de una forma efectiva el diálogo entre emisor y receptor, (microcomputador o periférico en cada caso) se dispondrá de un Hardware adicional consistente en líneas de control (Handshaking), independientes de las líneas de datos, que indiquen al receptor cuando el emisor ha enviado una información y ésta es válida, y al emisor cuándo el receptor la ha recibido y está dispuesto para una nueva transferencia.

Como ejemplo se podría considerar el protocolo de comunicación que seguirían los dos sistemas representados en la figura 7.2:

a) *Entrada:* El periférico activa la señal DAV (dato valido) con lo que la interface carga el registro de entrada y responde desactivando la línea DAC (dato aceptado), indicando al periférico que no puede enviar otra información. Se modifica el registro de estado de forma que cuando es consultado por el Microprocesador, éste se entera de que existe una información para ser tratada. Cuando el Microprocesador lea el registro de datos, se activa DAC, a lo que el periférico contesta desactivando DAV, quedando el sistema dispuesto para una nueva transferencia.

b) *Salida:* En este caso, la señal DAV se activa cuando el Microprocesador escribe un el registro de salida. Cuando el periférico contesta mediante la activación de DAC, se modifica el registro de estado, de forma que el microprocesador pueda enterarse por consulta que ya se puede enviar otra información. La desviación de DAV por parte del microprocesador y la consiguiente de DAC por parte del periférico deja al sistema en el estado inicial.

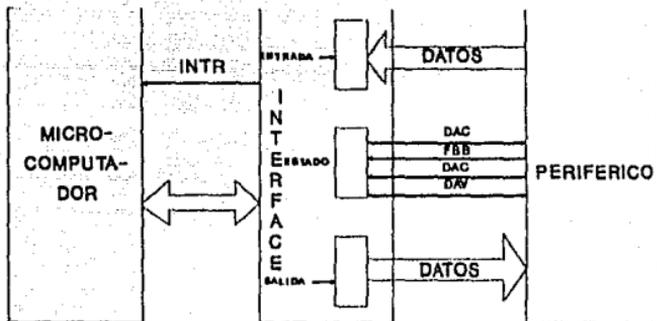


FIGURA 7.2 HARDWARE DE CONTROL ASINCRONO DE TRANSFERENCIAS

VII.4 ACCESO DIRECTO A MEMORIA (DMA).

La transferencia de datos entre un dispositivo de almacenamiento masivo, tal como el disco magnetico o cinta magnetica y el sistema de memoria, se limita a menudo por la velocidad del Microprocesador. Desconectado el procesador durante tal transferencia y dejando que el dispositivo periférico maneje la transferencia directamente a la memoria, mejoraría la velocidad de transferencia y se haria el sistema más eficiente. Esta técnica de transferencia se llama DMA (acceso directo de memoria). Durante la transferencia DMA el procesador estará inactivo de manera que no tenga control del

bus del sistema. Un controlador DMA acciona los buses para manejar la transferencia directamente entre, el dispositivo periférico y la memoria.

El Microprocesador puede hacer que quede inactivo de muchas maneras. El método más común es habilitar los buses mediante una señal de control especial.

El controlador DMA necesita los circuitos usados de una interconexión para comunicarse con el Microprocesador. Además, necesita un registro de direcciones, un registro contador de bytes y un grupo de líneas de direcciones. El registro y líneas de direcciones se usan para comunicación directa con el sistema RAM. El registro contador de palabras especifica el número de palabras que van a ser transferidas. La transferencia de datos se hace comúnmente, en forma directa entre el dispositivo periférico y la memoria bajo control del DMA.

La figura 7.3 muestra el diagrama de bloque de un controlador DMA típico. La unidad comunica con el Microprocesador vía el bus de datos y líneas de control. Los registros en el DMA son seleccionados para el Microprocesador por medio de unas líneas de direcciones, habilitando CS (selección de patillas) y RS (selección de registro). Las líneas RD y WR en el DMA son bidireccionales. Cuando BG = "0", el Microprocesador se comunica con el registro DMA a través de la barra de datos para leer o escribir en los registros DMA. Cuando BG = "1", el DMA puede comunicarse directamente con la memoria especificando una dirección en el bus de direcciones y

activando su control RD o WR. El DMA se comunica con un dispositivo periférico externo y a través de las líneas de requisición y reconocimiento.

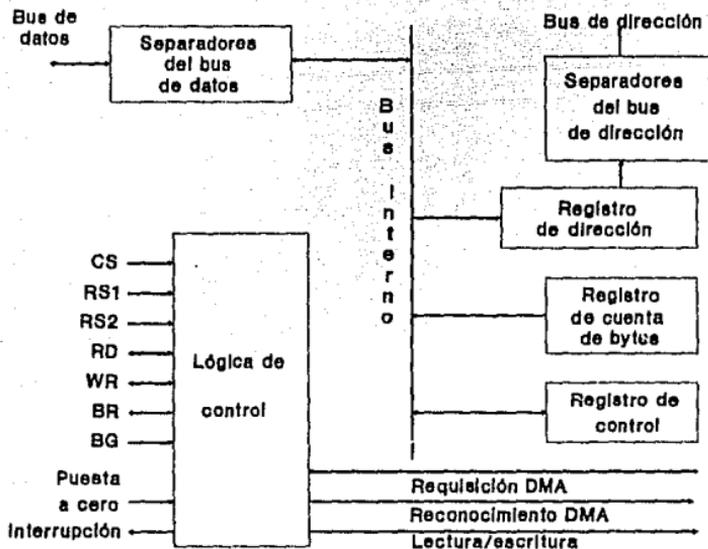


FIGURA 7.3 DIAGRAMA DE BLOQUE DEL CONTROLADOR DMA

El controlador del DMA incluye tres registros: Un registro de dirección, un registro contador de bytes y un registro de control. El registro de direcciones contiene 16 bits que especifican la posición deseada en la memoria. Los bits de la dirección pasan a través de un separador de buses y van a parar al bus de direcciones. El registro de direcciones se incrementa después de cada transferencia de byte DMA. El registro contador de bytes almacena el número de bytes que se van a transferir.

Este registro se decrementa después de la transferencia de cada byte y se comprueban internamente los ceros. El registro de control, especifica el modo de transferencia, bien sea hacia la memoria (escritura) o hacia fuera de ella (lectura). Todos los registros en el DMA actúan para el Microprocesador como una interconexión I/O. Así, el procesador puede leer o escribir en los registros DMA bajo el programa de control, vía el bus de datos.

El DMA se inicia primero por el Microprocesador. Después de ello, el DMA comienza y continúa la transferencia de datos entre la memoria y la unidad periférica, hasta que se transfiera un bloque completo. El proceso de inicio es esencialmente un programa que consiste de instrucciones E/S que incluyen la dirección DMA para seleccionar los registros particulares.

El Microprocesador inicia el DMA enviando la siguiente información a través del bus de datos:

1. La dirección de comienzo del bloque de memoria donde los datos están disponibles (para lectura) o donde los datos están almacenados (para escritura).
2. La cuenta de byte, la cual es el número de bytes en el bloque de memoria.
3. Los bits de control para especificar una transferencia de lectura o escritura.
4. Un bit de control para iniciar el DMA.

VII.5 SISTEMA DE INTERRUPCIONES.

Una interrupción es una primitiva de control externa de tipo Hardware, que permite "avisar" al Microprocesador de la ocurrencia de un suceso que precisa su atención y tratamiento. Las causas que puede genera una interrupción son diversas y no siempre están relacionadas con el sistema de entrada y salida. Así, se podrían considerar:

- Interrupciones por errores o averías de máquinas: Fallas de alimentación, errores de paridad en memoria, fin de cinta en una perforadora, etc.

- Interrupciones de reloj en sistemas operativos en tiempo real.

- Interrupciones externas de aviso de modificaciones importantes del mundo exterior.

- Interrupciones por causa de programa: Recepción de código de operación inexistentes, direccionamientos incorrectos, operaciones imposibles, desbordamiento, intento de escritura en zonas protegidas, etc.

- Interrupciones del sistema de entrada/salida: Avisos de fin de entrada de datos en transferencias por DMA, condiciones de periféricos libres o dispuestos, fin de temporizaciones, etc.

En muchos casos, existirán zonas de programas en las cuales no se podrá aceptar ninguna interrupción o solamente algunas de ellas, ya sea por temporizaciones críticas,

cuestiones de prioridades u otros motivos.

Para posibilitar esta opción, la mayoría de los Microprocesadores, disponen de instrucciones que permitan inhibir total o parcialmente el sistema de interrupciones. En este caso, en algunos Microprocesadores se recordará la petición a fin de serviría cuando se pueda; en otros, deberá ser el propio periférico quien "recuerde" la petición, manteniendo activa la señal hasta que sea servida.

En el control y gestión del sistema de interrupciones se deberán considerar diversos aspectos:

a) Las interrupciones deben ser servidas de manera que no imposibiliten de forma permanente el desarrollo de las tareas en proceso.

b) Debe haber algún método de permitir diversas interrupciones si existe más de un sistema que puede solicitarlas.

c) Si hay señales más importantes que otras o algunos periféricos requieren un servicio más rápido que otros, deberá existir un sistema jerárquico de interrupciones.

A partir del instante en que se solicita una interrupción, se desencadena una serie de acciones, algunas de forma automática y otras que deben ser realizadas por el programa.

En general, se termina de ejecutar la instrucción en curso, salvo en el caso de algunos Microprocesadores que disponen de instrucciones que precisan un tiempo de ejecución largo, como las de multiplicación, división y transferencias de

bloques de información, y permitan la interrupción en fase de ejecución.

Si la petición es aceptada, se inhibe total o parcialmente el sistema de interrupciones y se guarda el contador de programa en algún registro interno, en el "stock" o en una posición determinada de memoria. A continuación se pasa a la rutina de servicio en forma de salto vectorizado, o bien se genera un ciclo especial de lectura a fin de obtener la próxima instrucción suministrada en este caso por el propio periférico.

A partir de este instante el programa toma el control debiéndose realizar los siguientes pasos (figura 7.4):

a) Guardar el estado de la tarea en curso: Se deberán "salvar" todos aquellos registros que puedan ser modificados por la rutina de interrupción y que se precisen para poder reemprender la tarea interrumpida. En la mayoría de los Microprocesadores, se dispone de una estructura de stock para facilitar este tratamiento, que en algunos se realiza de forma automática. En algunos casos se dispone de la posibilidad de efectuar un cambio (swap) de registros operativos.

b) Identificación de la fuente de interrupción si puede existir más de un sistema o causas para solicitarla.

c) Rutina de servicio: Ejecución de las tareas asociadas con el servicio de una interrupción concreta. Constituye el tiempo realmente útil de todo el tratamiento.

d) Restitución del estado de la tarea interrumpida.

e) Rehabilitación en caso de que se precise hacerlo, del sistema de interrupciones.

f) Retorno al programa interrumpido.

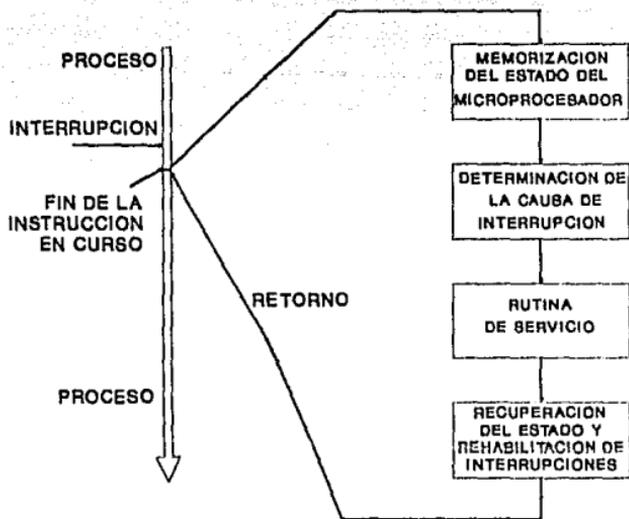


FIGURA 7.4 DIAGRAMA DE SERVICIO DE UNA INTERRUPCION

El método de control de transferencia por interrupciones consiste, básicamente, en que sea el propio periférico el que avise a la CPU de su disposición para aceptar una información (salida) o de que dispone de datos para entregar al Microprocesador (entrada).

En este caso, el Microprocesador interrumpirá la ejecución de la tarea que estuviese realizando, dará servicio al periférico y continuará con la tarea que quedó suspendida. De

esta forma, se penaliza al mínimo el tiempo de proceso con trabajos de gestión de transferencias.

El control por interrupciones permite:

- Un mejor aprovechamiento del sistema en general.
- Una respuesta rápida a un suceso externo.
- Sincronizar el mundo exterior con el proceso.
- Un cierto nivel de descentralización, ya que son los usuarios (periféricos) los encargados de "mover" el sistema.

La posibilidad de disponer de varias líneas de interrupción presenta las ventajas siguientes:

- Facilidad de identificación de la fuente de interrupción.
- Asignación de prioridades de una forma implícita.
- Posibilidad de enmascaramiento parcial del sistema de interrupciones.

VII.6 COMUNICACION ENTRE PERIFERICOS.

Los dispositivos periféricos estarán enlazados con la CPU siguiendo una de las estructuras siguientes:

1) Línea Compartida (Party-Line): Los diversos módulos de entrada y salida comparten en el tiempo más líneas únicas, tal como se indica en la figura 7.5a. En general, estas líneas son un subconjunto de los buses externos, por lo que el sistema periférico comparte el bus de direcciones, datos y control con el sistema de memoria. Esta estructura es la más utilizada, ya

que reduce significativamente el número de líneas que precisa la CPU para la gestión de E/S, y posibilita a la vez una concepción modular del sistema.

2) **Estructura Radial (Star):** En este caso cada periférico se conecta a la CPU a través de líneas dedicadas, tal como se ilustra en la figura 7.5b. Este sistema está muy limitado por la cantidad de terminales que se precisan en la CPU. Se utilizan sistemas en los que la CPU, posee algunas líneas de entrada/salida y en el caso de microcomputadores integrados en una sola pastilla, donde la incorporación de memoria ROM y RAM en el mismo circuito integrado que el resto de la CPU, permite disponer de terminales para funciones de E/S.

3) **Estructura de Cadena (Daisy-Chain):** En este caso las señales se propagan de un periférico a otro, tal como se encuentran en la figura 7.5c. Cuando un periférico concreto es seleccionado, bloquea la transmisión de las señales al resto del sistema. Este método se utiliza en algunos microcomputadores para disponer de un sistema jerárquico de interrupciones, estableciéndose los niveles según la posición en la cadena.

La gestión de entrada y salida, en su aspecto de programación puede considerarse a dos niveles:

1) **Instrucciones de Entrada y Salida:** Son tareas cuyas funciones están íntimamente ligadas al sistema de E/S, y entre las que se podrían destacar: Temporizaciones por programa, rutinas relacionadas por acceso directo a memoria, gestión de

tampones (buffers) de memoria para captura de datos, rutinas de empaquetamiento y desempaquetamiento de datos, rutinas de detección de errores, cambios de formato, etc.

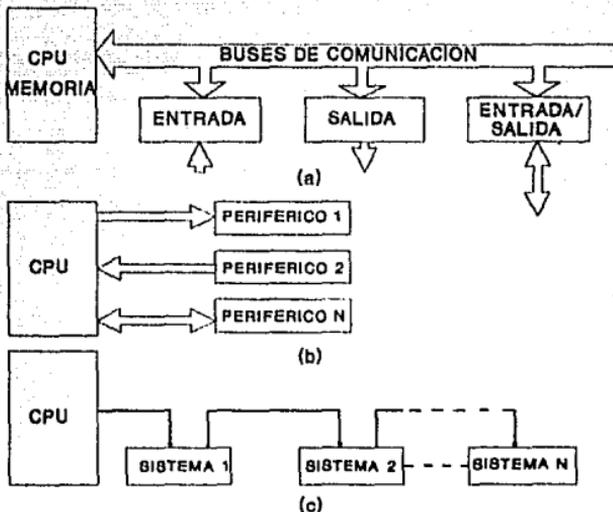
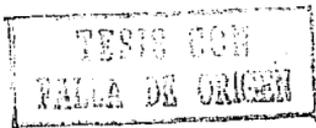


FIGURA 7.5 ESTRUCTURAS DE E/S a) LINEA COMPARTIDA b) ESTRELLA; c) CADENA

En la mayoría de los microcomputadores, no existen instrucciones específicas de E/S, ya que no se diferencia entre memoria y registros de E/S, por lo que las mismas instrucciones son aplicables a ambos sistemas. En otros microcomputadores, existen instrucciones que hacen referencia exclusiva al sistema de E/S y pueden ser de los tipos siguientes:



a) *Transferencia de Datos en Paralelo:* El código de operación indica si la operación es de entrada o de salida y, en general, indica de forma implícita el registro de la CPU (el acumulador en la mayoría de los casos), que interviene en la transferencia.

En máquinas con poca capacidad de direccionado, en cuanto a accesos de E/S, la misma palabra del código de operación puede indicar la dirección del registro periférico seleccionado. En máquinas de mayor capacidad de direccionamiento, o bien, el código de operación indica un registro de la CPU que sirve de base para un direccionamiento indirecto, o la dirección del periférico se da de forma directa en palabras sucesivas al código de operación.

b) *Transferencia, Seguida de un Test de una Línea Externa y Salto (skip) Condicional Según el Resultado del Test:* Este tipo de instrucciones facilita el diálogo con el periférico en transferencias programadas.

c) *Modificación:* Se ofrece la posibilidad de modificar algunos bits de un registro de salida, sin modificar el resto, con una sola instrucción. Consiste en realizar operaciones lógicas, (AND para la puesta a cero de un bit y OR para la puesta a uno), entre vías de acceso (ports) y datos inmediatos.

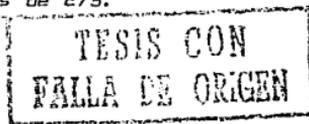
d) *Envío de órdenes especiales a periféricos:* En este caso, el código de operación contiene información que se envía al periférico, a fin de desencadenar una operación específica,

como puesta a uno, puesta a cero, complemento de biestables de control, etc.

e) Instrucciones de entrada y salida bit a bit: En máquinas con esta estructura de E/S, se posibilita el direccionamiento y acceso a un bit de entrada o de salida, además del acceso a una palabra completa.

VII.7 PROCESADORES E/S.

La solución lógica al programa de obtener máxima concurrencia en el procesamiento de la E/S, es disponer de un sistema de E/S inteligente que aisle a la CPU de los periféricos. La CPU sería entonces libre de proseguir a toda velocidad con su tarea primordial, de procesar el programa y manipular los datos internamente. El subsistema de E/S inteligente se obtiene mediante un procesador de E/S (PES). Básicamente, un procesador de E/S es aquel capaz, de ejecutar un pequeño repertorio de órdenes para atender la petición de E/S. La figura 7.6 ilustra los principales componentes arquitectónicos de un subsistema de E/S inteligente. Como se muestra en la figura, el procesador de E/S está asociado directamente al bus del sistema, y es responsable de la selección y extracción de las órdenes de E/S individuales contenidas en la memoria principal. El PES contiene generalmente, un procesador específicamente diseñado para el proceso de la E/S, y una serie de canales de E/S.



Los canales proporcionan una vía de comunicación entre el procesador de E/S y los controladores de dispositivos. También pueden existir canales de E/S sin PES, como se muestra en la figura 7.6.

En su forma más simple, y cuando está sólo, un canal puede ser un pequeño procesador que realiza operaciones de DMA para un reducido conjunto de dispositivos. Si el canal va incorporado dentro de un PES, se trata esencialmente de un componente pasivo sin capacidad por sí mismo de procesamiento lógico. Cuando el canal posee capacidad de proceso, se suele utilizar como PES.

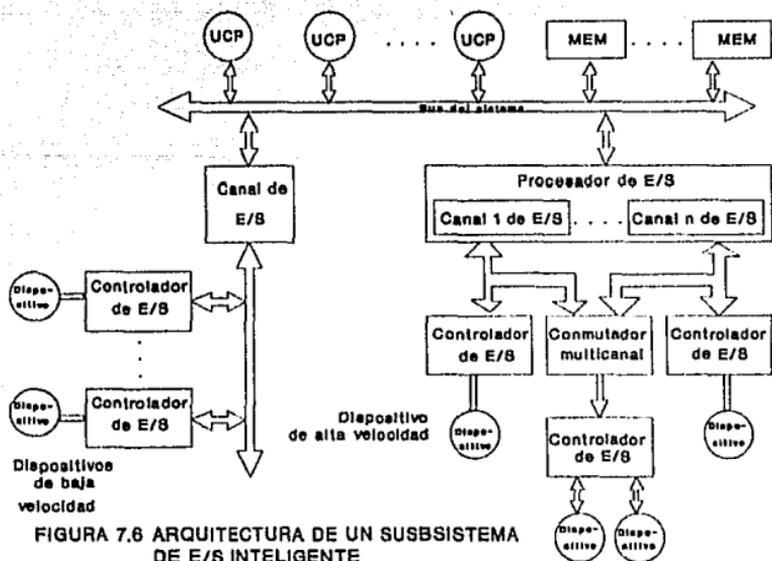


FIGURA 7.6 ARQUITECTURA DE UN SUBSISTEMA DE E/S INTELIGENTE

VII.B PERIFERICOS.

El rendimiento de un computador, puede ser afectado por trabajos limitados por cálculo o por trabajos limitados por E/S. A veces, la distinción entre el dispositivo y la interface asociada es confusa. La interface de E/S controla la operación del dispositivo periférico ligado a ella. Las operaciones de control se inician mediante órdenes desde la CPU. El repertorio de órdenes usadas para efectuar una transacción de E/S, se denomina controlador Software (Driver) de dispositivo. Las funciones de la interface, son almacenar y efectuar conversión de los datos a un formato requerido. También, detectar errores de transmisión, y solicitar la repetición de una transacción de E/S en caso de error. Además, la interface puede interrogar, iniciar y detener el dispositivo, de acuerdo con las órdenes emitidas por la CPU. En ciertos casos, la interface también puede interrogar a la CPU si el dispositivo solicita atención urgente. No todas las interfaces poseen estas capacidades, y existen muchas opciones de diseño dependiendo de las características del dispositivo. A continuación, reseñamos unos cuantos dispositivos y sus velocidades características.

Existen muchos tipos diferentes de dispositivos periféricos. La mayor parte de ellos son dispositivos electromecánicos y por tanto transfieren datos a una frecuencia limitada por la velocidad de los componentes electromecánicos. La tabla 7.1 muestra algunos dispositivos periféricos típicos.

Las memorias de burbujas, las unidades de disco, de tambor y de cinta magnética, son dispositivos de almacenamiento de masas que almacenan datos a bajo coste para su posterior recuperación. Las terminales de pantalla, son dispositivos de entrada y salida compuestos por teclados y tubos de rayos catódicos (TRC). El teclado actúa como entrada, mientras que el TRC es el elemento visualizador de la salida. En algunos casos en los que el TRC se sustituye por un impresora, los terminales se denominan teletipos.

DISPOSITIVO DE E/S	FUNCION	VELOCIDAD DE TRANSFERENCIA
Memorias de burbuja	Memoria de masas	300K, 4M cps*
Dispositivos de carga acoplada	Memoria de masas	500 K, 4M cps
Discos de cabeza fija de cabeza móvil flexibles		300K, 2M cps 30K, 1M cps 25K cps
Terminal de pantalla	Entrada-salida	10-300 cps
Impresora de líneas de impacto electrostática de chorro de tinta	Salida	100-3000 lpm+ 300-40000 lpm 100-3000 lpm
Unidad de cinta magnética de carretes (7.9 pistas)	Memoria de masas	15-300K cps
Cassette		10-400 cps

TABLA 7.1 ALGUNOS DISPOSITIVOS DE E/S

* Caracteres por segundo

+ Líneas por minuto

C O N C L U S I O N E S

Al finalizar el presente trabajo, y dados los elementos y conocimientos necesarios para ser capaz de entender y analizar la mayor parte de asuntos relacionados con la organización de una computadora, se puede concluir lo siguiente:

El conocer y dominar el ambiente interno de una computadora contribuye al desarrollo profesional de la gente en general, ya que con ello se puede comenzar un diseño propio de una máquina. Desafortunadamente en México no existen herramientas de apoyo necesarias para realizar físicamente dicho diseño.

Sin embargo, el hecho de conocer todos y cada uno de los componente es muy ventajoso, ya que con ello podemos diferenciar las características de los mismos (microprocesadores, memorias y periféricos en general).

Por otro lado, este trabajo nos proporciona una variedad de formas para el diseño de control, ventajas y desventajas entre los mismos, con lo cual podemos elegir el que se adapte a las necesidades de nuestro proyecto.

El hecho de saber las características de las diferentes memorias, material de fabricación, proceso de fabricación,

método de grabación, etc., nos permite conocer no solo el funcionamiento de una computadora, sino todo el mercado electrónico que se auxilie de las mismas, con ello nos da opción de elegir y asesorar, incluso modificar los aparatos.

Por último es una necesidad imperiosa, complementar todo este material con un laboratorio, en donde se lleven a cabo prácticas y proyectos.

B I B L I O G R A F I A

**Interconexión de Periféricos
a Microprocesadores**
José Mompín Poblet y varios
Edit. Publicaciones Marcombo
Mexico 1987

**Lógica Digital y Diseño
de Computadores**
M. Morris Mano
Edit. Prentice-Hall
Mexico 1988

Arquitectura de Computadoras
M. Morris Mano
Edit. Prentice-Hall
Mexico 1988

**Arquitectura de Computadoras
y Procesamiento en Paralelo**
Kay Hwang y Faye A. Briggs
Edit. MacGraw-Hill
Mexico 1988

Organización de Computadoras
V. Carl Hanacher, Zvonko G.
Vranesic, Safwat G. Zaky
Edit. McGraw-Hill
Mexico 1986

**80386/80286 (Programación en
Lenguaje Ensamblador)**
William Murray III y
Chris H. Papas
Edit. McGraw-Hill
España 1990

**Lenguaje Ensamblador para
Microcomputadoras IBM
J. Terry Godfrey
Edit. Prentice-Hall
Mexico 1991**

**Informática Presente y
Futuro
Donald H. Sanders
Edit. McGraw-Hill
México 1989**