

TEMA 5. ORGANIZACIÓN DE LA MEMORIA DE UN COMPUTADOR

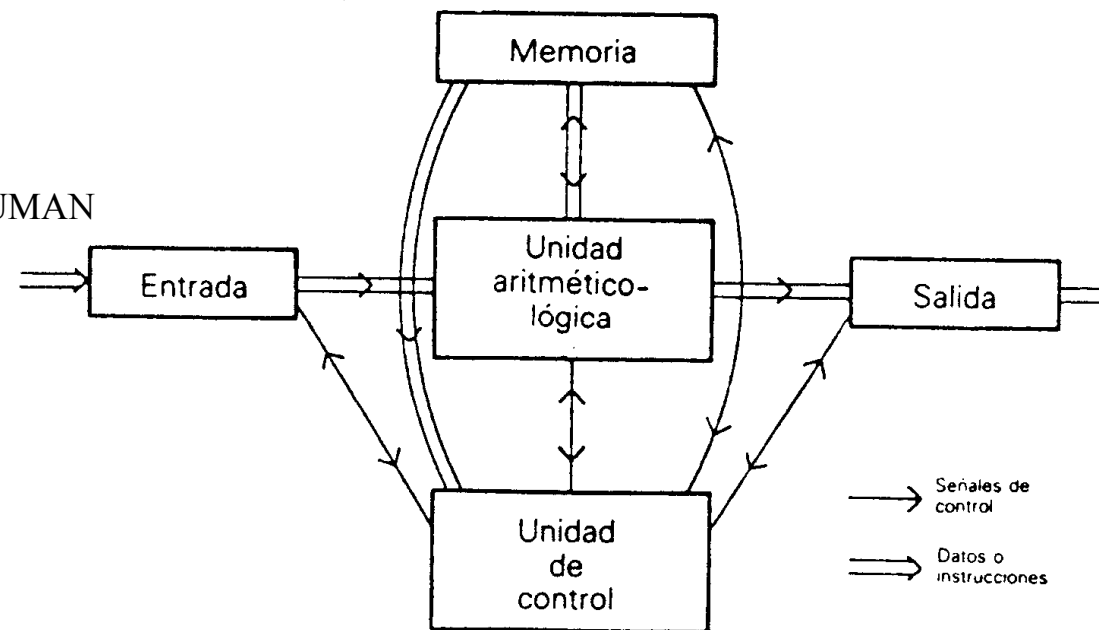
INTRODUCCIÓN

▪ Objetivos

- Analizar la organización de la memoria
 - Organización por niveles
 - Ventajas de cada uno de los niveles
 - Influencia sobre el rendimiento del computador
- Soporte Hardware para su implementación

▪ Visión global

- Componentes básicos de la Estructura VON NEUMAN
 - Unidad de Cálculo
 - Unidad de Control
 - Unidad de Memoria
 - Unidades de E/S



- **Papel de la Unidad de Memoria de un Computador**

- Almacenamiento de
 - Programas y Datos de usuario(s)
 - Programas y Datos del Sistema

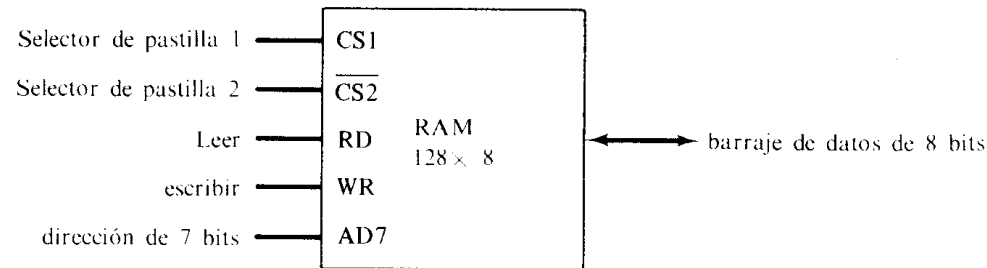
- **Características de la Memoria**

- Parámetros a tener en cuenta
 - Capacidad (Kbytes o Mbytes)
 - Velocidad de Acceso (miliseg, microseg. o nanoseg)
 - Costo (por bit o byte)
- Condiciones óptimas (imposibilidad de cumplir todas)
 - Gran capacidad (Gigabytes)
 - Rápidas (mínimo nanoseg.)
 - Baratas

- Una primera clasificación
 - Memoria Principal
 - * Acceso directo por la CPU
 - * Almacena los programas que se están ejecutando
 - * De tipo Semiconductora
 - Estáticas (SRAM) o Dinámicas (DRAM)
 - De Lectura/Escritura o de solo Lectura (ROM)
 - * Capacidad del orden de Megas
 - * Tiempos de acceso de decenas de nanosegundos
 - * Costo alto y proporcional a la velocidad de acceso
 - Memoria Secundaria o Auxiliar
 - * Acceso a través de Interfaces
 - * Almacena programas y datos no ejecutándose
 - * De tipo Magnético u óptico
 - Discos y Cintas
 - * Capacidades del orden de Gigas
 - * Tiempos de acceso de miliseg.
 - * Baratas y proporcional a la capacidad

▪ Organización mínima de la Memoria Principal

- Memoria de Acceso Aleatorio de R/W (RAM)
 - Líneas de dirección
 - Líneas de Datos (1/0)
 - Líneas de Habilitación (CS)
 - Línea de control Lectura/Escritura (R/W)



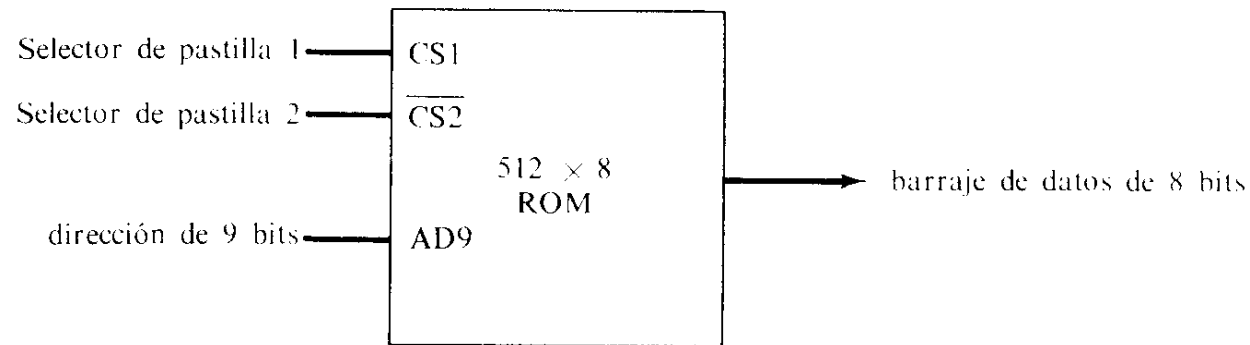
(a) Diagrama de bloques

CS1	$\overline{\text{CS2}}$	RD	WR	Función de memoria	Estado del barraje de datos
0	0	X	X	Inhibir	Alta impedancia
0	1	X	X	Inhibir	Alta impedancia
1	0	0	0	Inhibir	Alta impedancia
1	0	0	1	Leer	Entrada de datos al RAM
1	0	1	X	Escribir	Salida de datos al RAM
1	1	X	X	Inhibir	Alta impedancia

(b) Tabla de función

CIRCUITO RAM TÍPICO

- Memoria de Acceso Aleatorio de solo Lectura (ROM)
 - Líneas de dirección
 - Líneas de Datos (O)
 - Líneas de Habilitación (CS)



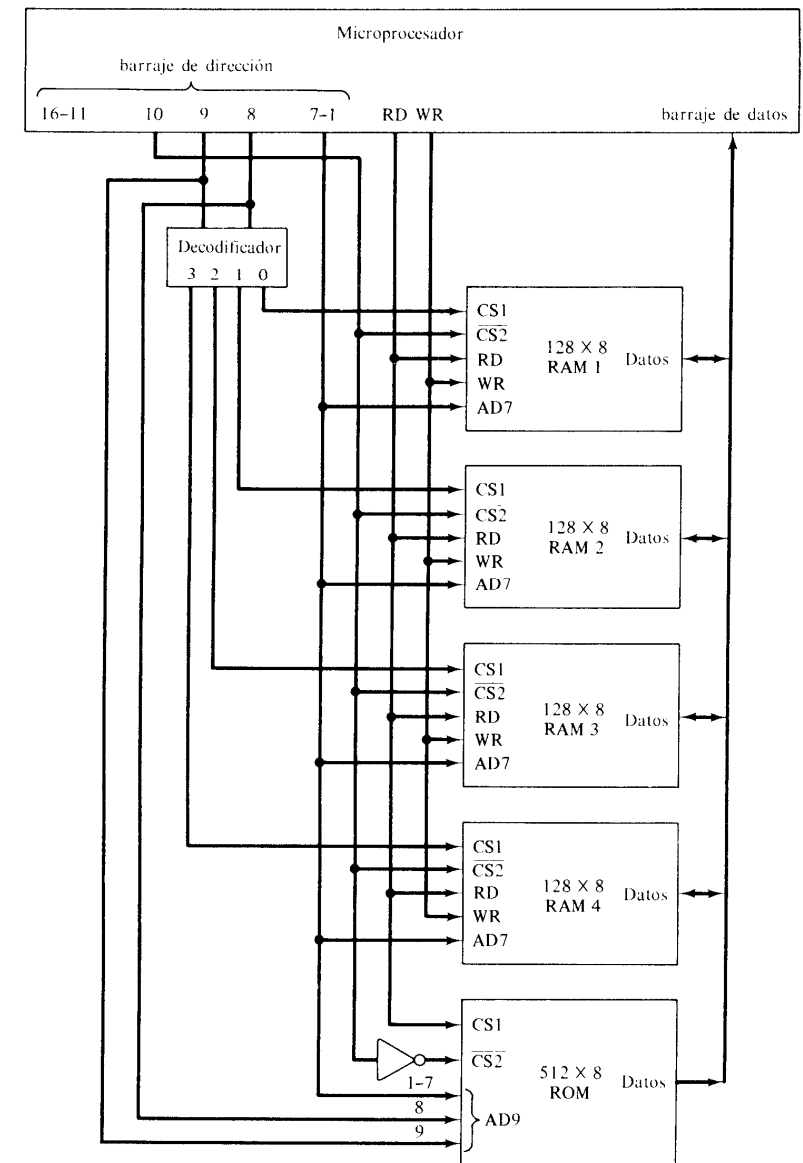
CIRCUITO ROM TÍPICO

- Mapa de direcciones

Componente	Dirección hexadecimal	barraje de dirección							
		10	9	8	7	6	5	4	3 2 1
RAM 1	0000-007F	0	0	0	x	x	x	x	x x x
RAM 2	0080-00FF	0	0	1	x	x	x	x	x x x
RAM 3	0100-017F	0	1	0	x	x	x	x	x x x
RAM 4	0180-01FF	0	1	1	x	x	x	x	x x x
ROM	0200-03FF	1	x	x	x	x	x	x	x x x

– Organizaciones de la Memoria para un Sistema Microprocesador

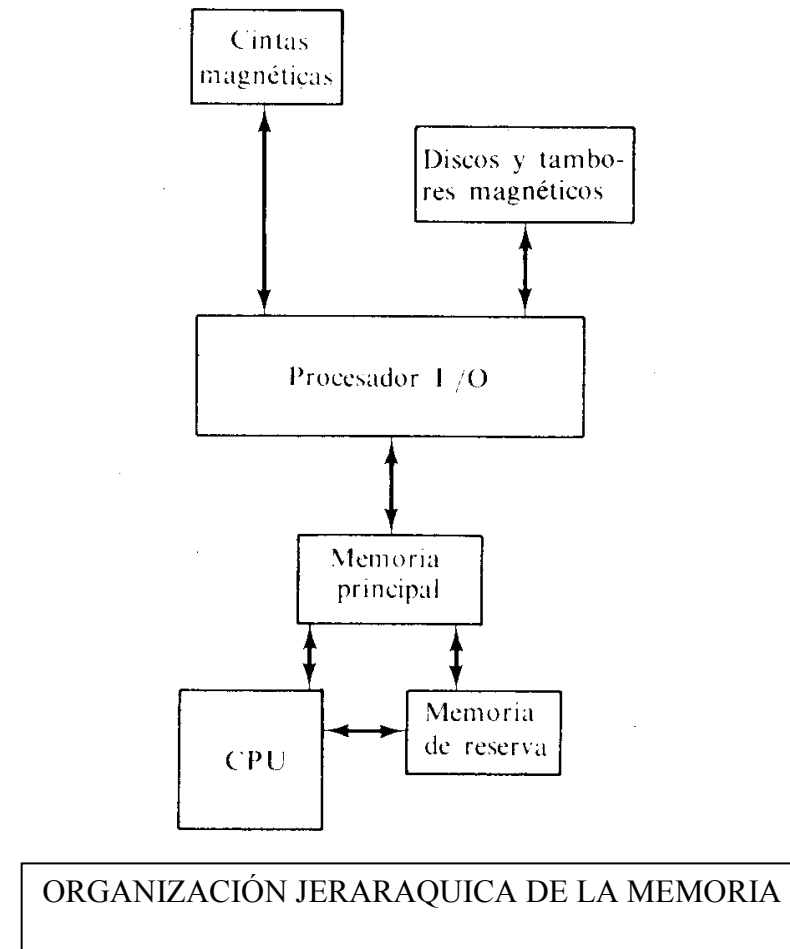
CONEXIONES DE MEMORIA AL MICROPROCESADOR



JERARQUÍA DE LA MEMORIA

- **Definición amplia de Memoria "Todo componente que almacena información en la computadora"**
 - Registros
 - Memoria Principal
 - Discos Magnéticos u ópticos
 - Cintas Magnéticas
 - etc...
- **Clasificación Jerárquica**
 - Tiene en cuenta varias cuestiones
 - Valores de sus parámetros (Capacidad, Rapidez y Costo)
 - Forma de acceso de la CPU (Directo o a través de Interfaces)
- **Memoria de Reserva (M. Caché)**
 - Almacenar la información que más se utiliza
 - Instrucciones y Datos
 - Debe ser muy rápida
 - No es muy grande por ser costosa
 - Se accede directamente desde la CPU y la M. Principal

- **Relación entre los parámetros de cada nivel Jerárquico**
 - A mayor nivel jerárquico mayor rapidez
 - A mayor nivel jerárquico mayor costo
 - A menor nivel jerárquico mayor capacidad
- **Necesidad de un Módulo Administrador de la Memoria (S.O.)**
 - La amplitud obliga a su optimización
 - Programas que necesitan más memoria de la existente
 - Complejidad en Sistemas Multiusuarios



MEMORIA ASOCIATIVA

"Memoria de acceso para la lectura por contenido y no por dirección (CAM)"

- **Diferencia con el tipo de memorias visto hasta ahora**
 - Acceso aleatorio por dirección
 - Acceso Secuencial
 - Tipo LIFO o FIFO
- **Casos en los que interesa este tipo de acceso**
 - Búsqueda en base de datos
 - Obtención de datos de una cuenta a partir del nombre
- **Tipo de búsqueda del contenido**
 - De forma secuencial
 - Barrido de direcciones y comparación de contenido
 - Lenta aunque sencilla y económica
 - * Memoria RAM, Contador y Comparador
 - De forma paralela
 - Comparación de todos los contenidos a la vez
 - Rápida pero compleja y costosa

- **Características que se pretenden de este tipo de Memoria**

- Acceso de lectura por contenido
 - por palabra entera
 - por campo en una palabra
- Muy rápidas
- El acceso para escritura puede ser
 - por dirección
 - se almacena en cualquier posición vacía

ORGANIZACIÓN DEL HARDWARE

- **La búsqueda debe ser rápida, lo que complica la circuitería**

- Incluir la comparación
- Incluir el estado de las posiciones
- etc..

- **Componentes de una Memoria Asociativa**

- Matriz de memoria que incluye
 - Celdas de almacenamiento (**m x n**)
 - Circuitería de comparación

- Registro de Argumento (**A**)
 - Contiene el dato o instrucción que se busca (**n**)
- Registro Clave (**K**)
 - Para emmascarar parte del argumento de búsqueda (**m**)
- Registro Equiparador (**M**)
 - Indica las posiciones que tienen ese contenido (**m**)

▪ **Ejemplo de búsqueda**

- **A** 101 111100
- **K** 111 000000
- palabra 1 100 111100 $M_1 = 0$
- palabra 2 101 000001 $M_2 = 1$
-

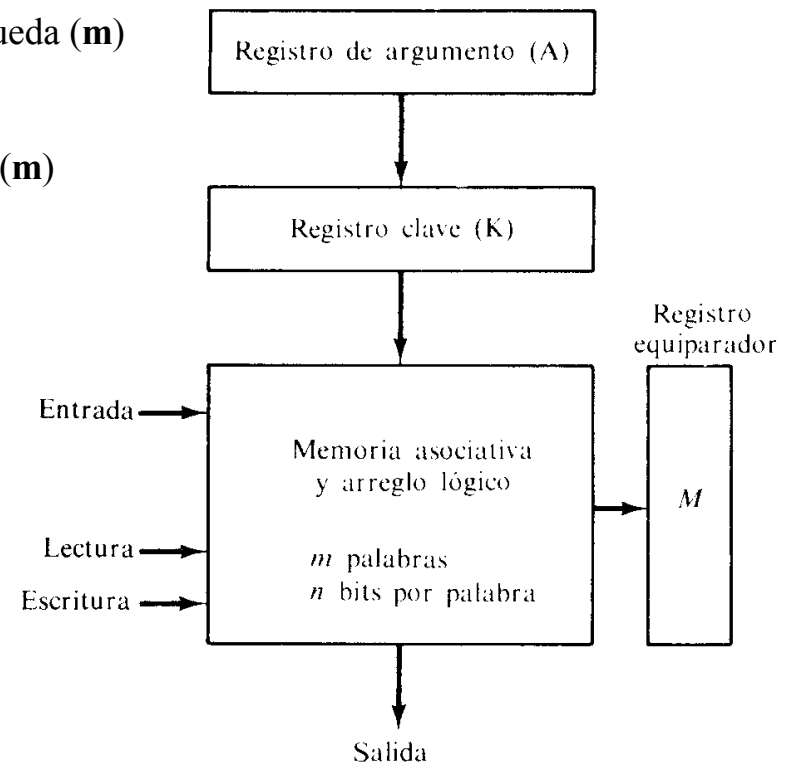
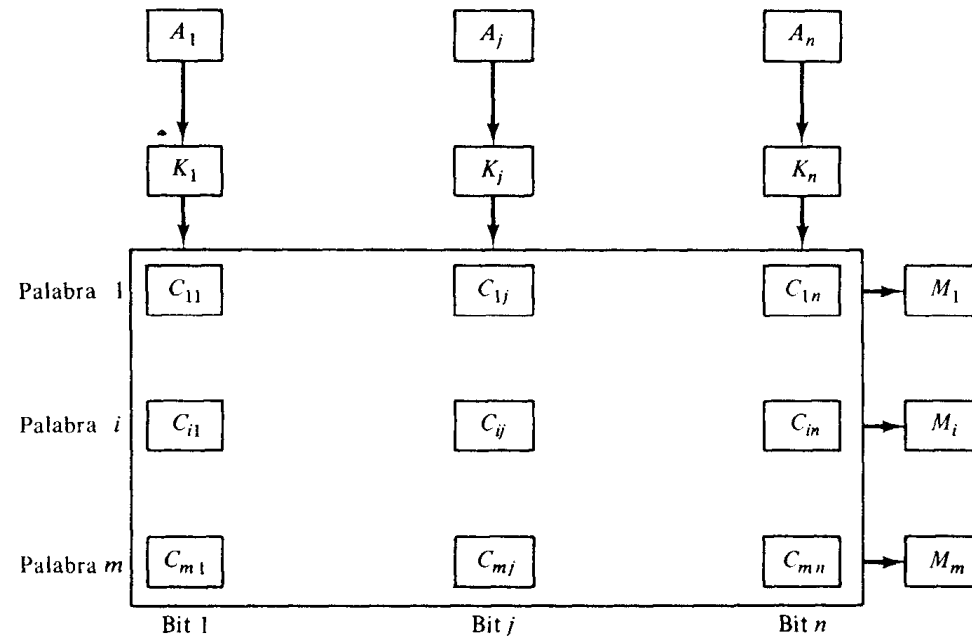


DIAGRAMA DE BLOQUES DE UNA MEMORIA ASOCIATIVA

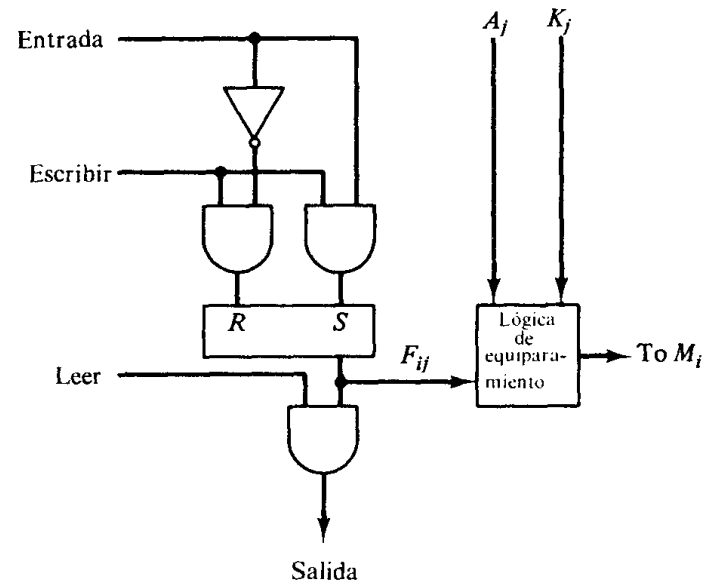
- Organización de la matriz

- Conjunto de celdas asociativas (C_{ij})
- Registros **A**, **K** y **M**



MATRIZ ASOCIATIVA DE **M** PALABRAS DE **N** CELDAS

- Estructura de la celda básica
 - Biestable de almacenamiento
 - Lógica de comparación y seguimiento
 - Lógica para escritura



CELDA DE MEMORIA ASOCIATIVA

▪ Lógica de comparación o equiparamiento

- Función de igualdad de dos bits: x_j

$$x_j = A_j F_{ij} + A'_j F'_{ij}$$

A_j = bit j del registro de Argumento

F_{ij} = bit j de la palabra i de almacenamiento

- Función de equiparamiento de la palabra almacenada

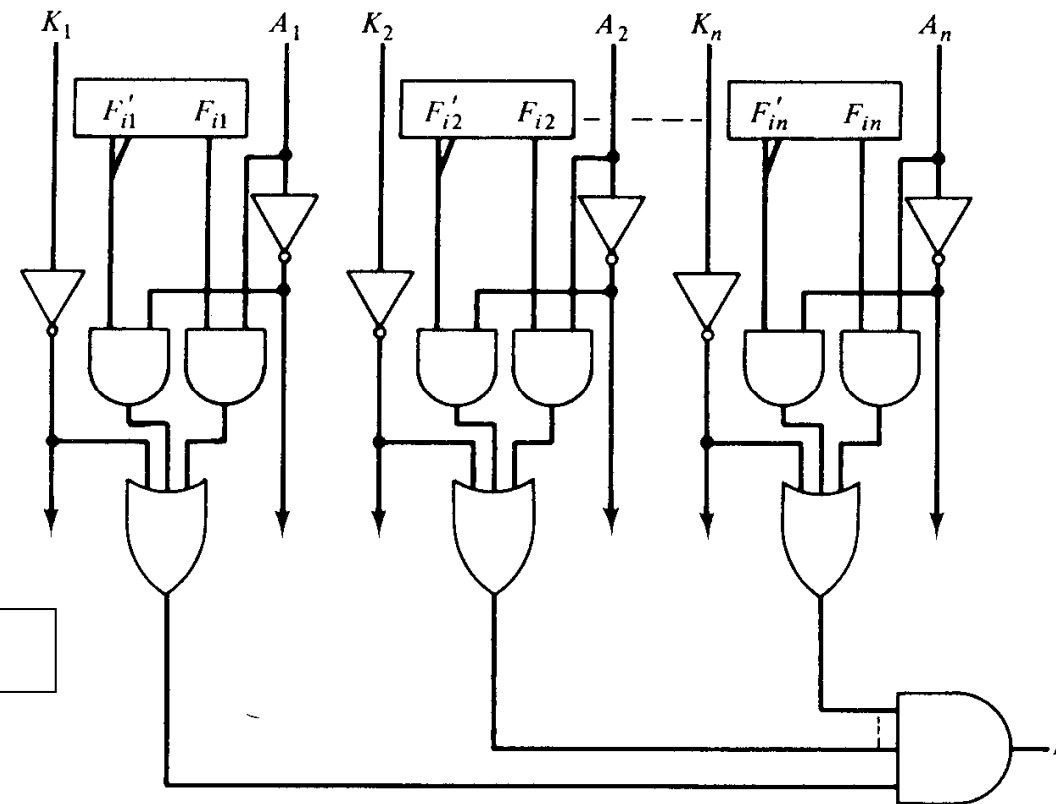
$$M_i = x_1 x_2 x_3 x_4 \dots x_n$$

M_i = bit i del Registro de Equiparamiento M

- Si se tiene en cuenta la Máscara

$$M_i = (x_1 + K'_1)(x_2 + K'_2)(x_3 + K'_3) \dots (x_n + K'_n)$$

K_j = bit j del Registro de enmascaramiento



LOGICA DE EQUIPARAMIENTO

▪ Operación de lectura

"Se busca una información cuya etiqueta coincida con la que se presenta"

- Se presenta la información de entrada
 - palabra de búsqueda (n) en el R. Argumento
 - palabra de máscara (n) en R. Clave
 - * El n° de bits a comparar siempre es menor que n
- Testeo de los bits del Registro M (**m**)
 - Si todos a 0 => No se encuentra (Fallo)
 - Algunos M_i a 1 => si se encuentra (Éxito)
 - Lectura secuencial de las posiciones con su M igual a 1
 - * Problema de lentitud
- Si solo puede estar en una posición; caso de que esté
 - Definición de la función de decisión **V**:

$$V = M_1 + M_2 + M_3 + \dots + M_m$$

M_i => bits del Registro de equiparamiento
 - Generación de la salida en las líneas de Datos

$$D_j = F_{1j}M_1 + F_{2j}M_2 + F_{3j}M_3 + \dots + F_{mj}M_m$$

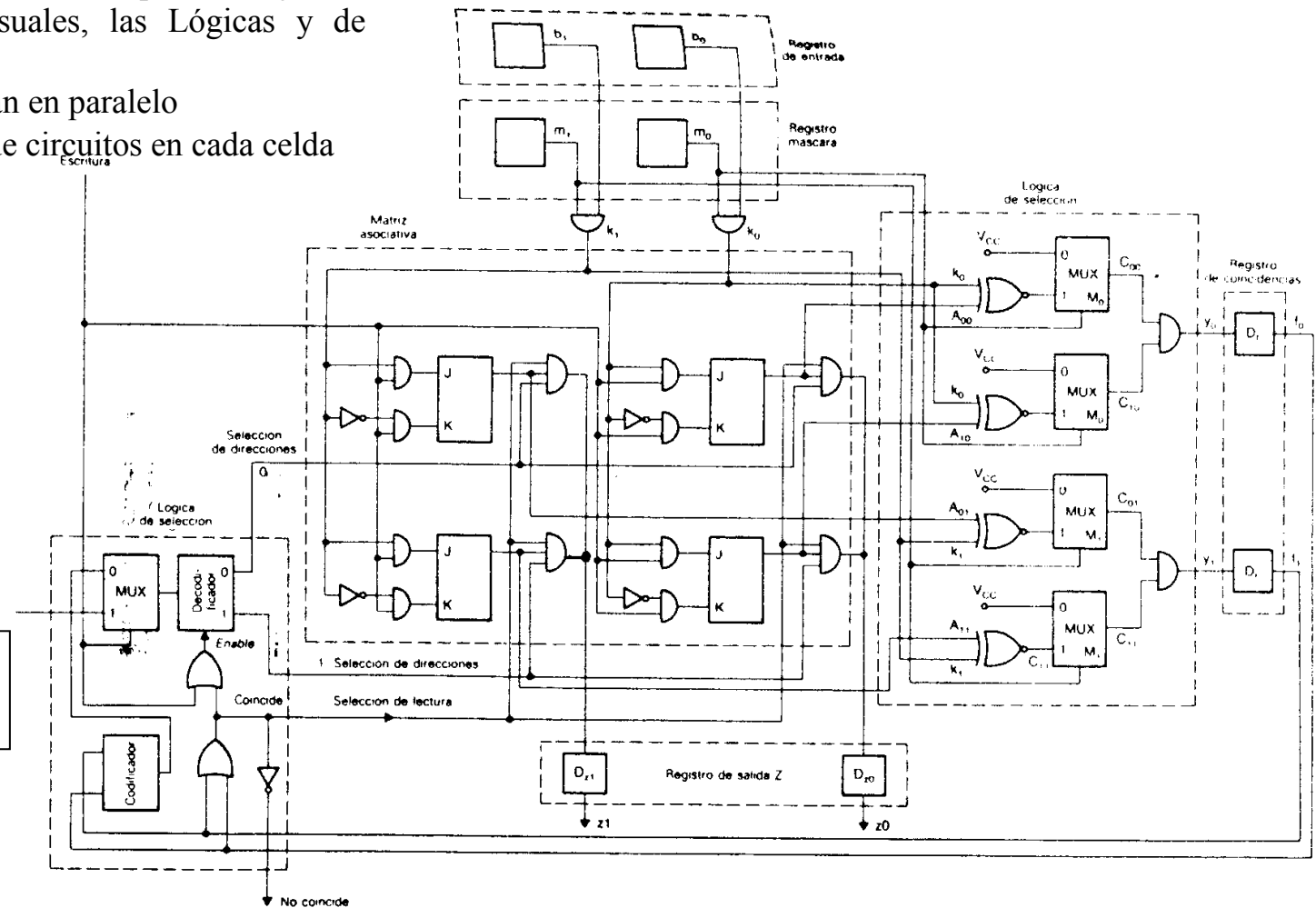
$j=1,2,3, \dots, n$
 - Si $V = 1$ se lee los datos en las líneas de salida
 - Si $V = 0$ se produce fallo de lectura

- **Operación de escritura**

- Necesidad de esta función para introducir la información
- Dependiendo de la aplicación puede hacerse
 - Direccionando las posiciones de forma secuencial
 - Similar a una de acceso aleatorio
 - Solo accediendo a aquellas posiciones vacías
 - * Registro de Rótulo **R**, de m bits (tantos como palabras)
 $R_i = 1 \Rightarrow$ la posición está ocupada
 $R_i = 0 \Rightarrow$ la posición está libre
 - * Se testean los bits de R hasta encontrar el primer 0
 - Se escribe la palabra
 - Se pone el bit de R a 1
 - Los **R_i** se deben usar también como máscara
 - * Impide la lectura de una información no válida

▪ Procesador Asociativo

- Computador cuya Memoria es de tipo Asociativo
 - Cada celda es asociada a sus vecinas para manejar las operaciones Aritméticas usuales, las Lógicas y de Desplazamiento
 - Las operaciones se computan en paralelo
 - Requerimientos excesivos de circuitos en cada celda



ORGANIZACIÓN DE UNA MEMORIA ASOCIATIVA DE 2*2 PALABRAS

MEMORIA VIRTUAL

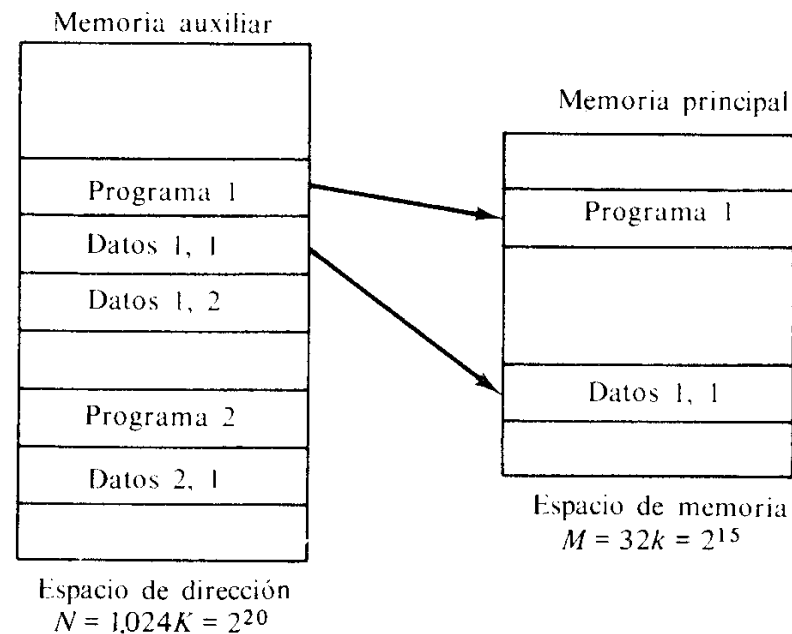
- **Sistemas de Programa Almacenado**
 - El programa debe estar en Memoria Principal
 - Su tamaño menor que la memoria física
 - Problemas con Sistemas multitareas y multiusuarios
 - Imposibilidad de disponer de tanta memoria
 - En un principio el programa está en Memoria Secundaria

- **Sistemas con Memoria Virtual**
 - Permite ejecutar programas de gran tamaño de la M. Secundaria
 - El Programa(s) está (n)
 - Completos (s) está(n) en M. Secundaria
 - Parte que se ejecuta en M. Principal
 - La CPU lanza direcciones virtuales para captar Inst. y/o Datos
 - Conversión a direcciones físicas
 - Necesidad de Hardware apropiado
 - Si no se encuentra en M. Principal
 - Traerlo desde M. Secundaria

-
- Inconveniente del costo temporal de acceso a M. Secundaria
 - Cuando se producen fallos de acceso
 - Es una combinación e software y hardware
 - Software de administración
 - Hardware de mapeo
 - **Espacio de Dirección y de Memoria**
 - Espacio de Dirección
 - Conjunto de direcciones virtuales, o direcciones que genera el programa
 - Espacio de Memoria
 - Conjunto de localizaciones, o direcciones físicas de la memoria
 - En los Sistemas de Programa Almacenado clásico coinciden
 - En los Sistemas de Memoria Virtual el Espacio de Dirección es mayor

▪ Ejemplo de un Computador con Memoria Virtual

- Memoria Principal (física) = 32K (15 bits de dirección)
- Memoria Auxiliar = 1.024K (20 bits de dirección)
 - Espacio de dirección $N = 1024\text{ K}$
 - Espacio de Memoria $M = 32\text{K}$
- Sistema multiprogramado
 - Se cargan varios trozos de Programas y Datos
 - * No se cargan siempre en la misma posición
 - * El Programa y los Datos no tienen que ser contiguos



RELACIÓN ENTRE ESPACIO DE DIRECCIÓN Y DE MEMORIA

- Necesidad de traducir direcciones virtuales a físicas
 - De acuerdo con la dirección virtual
 - De acuerdo con el lugar donde se ha cargado
- Existencia de una Tabla de Mapeo de Memoria
 - Se puede almacenar en M. separada
 - * Se necesita ese módulo
 - * Tiempo de acceso extra
 - Se puede almacenar en M. Principal
 - * Se necesitan dos accesos a memoria
 - * Mucho más lenta
 - Utilizar memoria muy rápida para realizar esa traducción

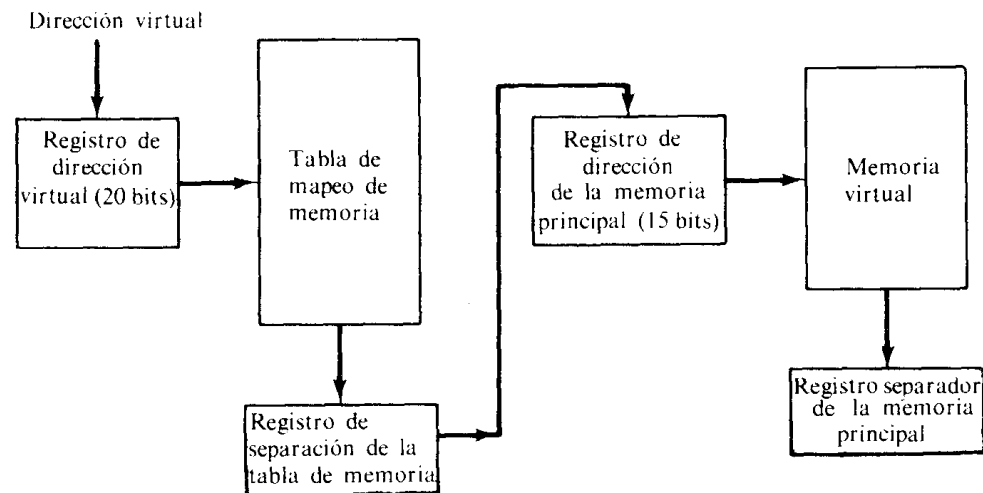


TABLA DE MEMORIA PARA MAPEAR UNA DIRECCIÓN VIRTUAL

▪ Mapeo de dirección

- División de los dos espacios en trozos
 - Página para el Espacio de Direcciones
 - Bloques para el Espacio de Memoria
 - Ambos tienen un tamaño fijo e igual
 - Los programas también se dividen en páginas
 - El traslado de M. Secundaria a Principal es por página
 - Los tamaños típicos de páginas son: 512B, 1KB, 2KB ó 4KB
 - Ejemplo: (página de 1K.)
 - Programa de 8 páginas (8 KB) => 13 bits de dirección
 - Memoria con 4 bloques (4 íCB) => 12 bits
 - Significado de una dirección virtual
 - p bits más significativos indican el n° de página
 - Resto de los bits la posición dentro de la página
 - En este caso $p = 3$ (los 3 msb)
- dirección: 1011000000001 ==> página 5, posición 513

Página 0
Página 1
Página 2
Página 3
Página 4
Página 5
Página 6
Página 7

Espacio de dirección
 $N = 8K = 2^{13}$

Bloque 0
Bloque 1
Bloque 2
Bloque 3

Espacio de memoria
 $M = 4K = 2^{12}$

- Mapeo mediante Tabla de página de Memoria
 - Cada posición (que indica no de página) contiene
 - * No de bloque en el que está
 - * bit de presencia (1 = SI, 0 = NO)
 - Solo se necesita traducir de página a bloque
 - * La búsqueda se hace accediendo a la dirección
 - La posición en la página y el bloque es la misma
 - La dirección física es el n° de bloque más la posición

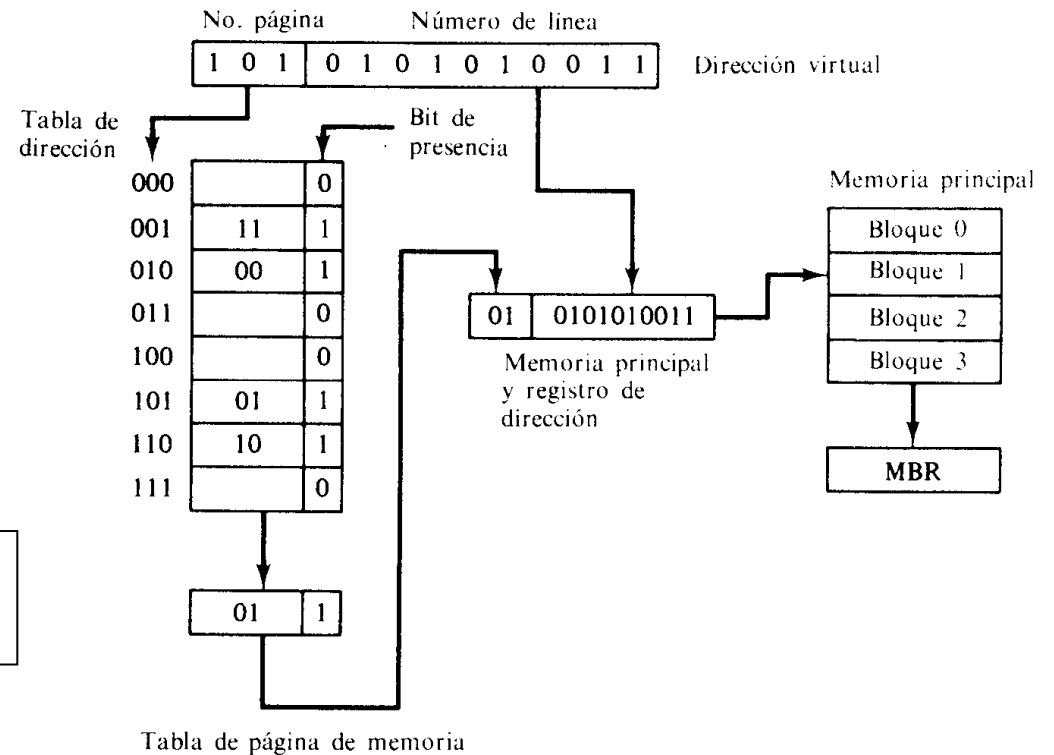


TABLA DE MEMORIA EN UN SISTEMA DE PÁGINA

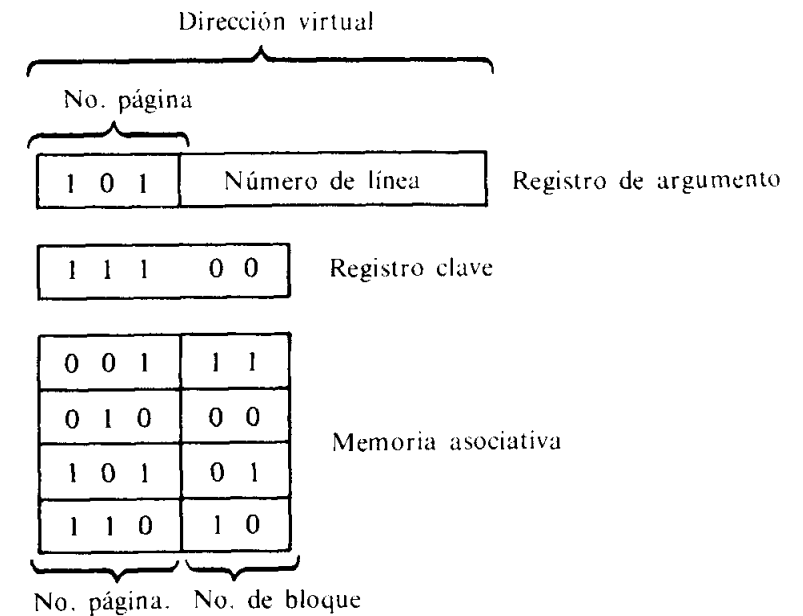
- Inconveniente del tamaño de la tabla
 - Se necesitan tantas posiciones como páginas
- Inconvenientes de la M. Virtual paginada
 - Penalización cuando la página no está en M. principal
 - * Acceso a Memoria Secundaria
 - * Algoritmos de reemplazo
- Mejora del tamaño de la tabla
 - La tabla tiene tantas posiciones como bloques (menor)
 - En cada una está el n° de página que se contiene
- Proceso de búsqueda
 - Se averigua si existe esa página y en que posición está
 - Si se hace secuencialmente es muy lenta

▪ Tabla de Página de Memoria Asociativa

Ineficiencia de memoria de acceso aleatorio para la tabla de página

- Con M. Asociativa es parecida a la vista últimamente
 - Tantas posiciones como bloques
 - La búsqueda se hace por contenido (asociativa)
 - * Muy rápida
 - Cada palabra contiene dos campos
 - * 1° indica el n° de página
 - * 2° indica el n° de bloque
 - Detección rápida si está presente una página
 - * SI ==> genera dirección física
 - n° de bloque mas posición en la página
 - * NO ==> genera un remplazo de página
 - algoritmos de remplazo
- Para el mismo ejemplo

TABLA DE PÁGINA DE MEMORIA ASOCIATIVA



Reemplazo de página

- El Software de administración de memoria decide
 - Que página de memoria hay que retirar
 - Cuándo una página es transferida de M. Auxiliar a Principal
 - En que lugar de M. Principal se coloca la página
- Carga inicial de unas páginas en M. Principal
 - Indicación en la tabla de página de su posición
- Fallo de página
 - Cuando se referencia una página que está en M. Auxiliar
- Acciones en presencia de un fallo de página
 - Suspensión de ejecución de programa (estado de espera)
 - Lanzar un proceso de E/S a un procesador de E/S
 - * Operación de traer la página necesaria a M. P.
 - Paso de estado de listo cuando se completa la E/S
- Pasos de transferencia de página de M. Auxiliar a Principal
 - Retirada de una página si la M. Principal está llena (transferencia a M. A. si ha cambiado)
 - * Algoritmos de reemplazo
 - Transferencia de la página y actualización de tabla
- Algoritmos de reemplazo
 - Tipo FIFO; elimina página que entró primero
 - * Existencia de una PILA (FIFO) de página
 - Tipo LRU; elimina la menos recientemente utilizada
 - * Asocia un contador a cada posición
 - Incremento periódico
 - A cero cada vez que se accede a la posición
 - * Se elige la que tiene el valor más alto

MEMORIA DE RESERVA

▪ Localidad de Referencia

"Las referencias a memoria en un intervalo de tiempo suelen estar confinadas en unas áreas localizadas de memoria"

- Los bucles
- Las subrutinas
- La secuencialidad propia de los programas
- Acceso a tablas de datos

▪ Forma de acelerar la ejecución de un Programa

- Colocando en una memoria rápida los trozos y datos activos de programa

▪ Memoria de Reserva

- Memoria colocada entre la CPU y M. Principal
- Contiene los trozos y datos activos de programa
- Tiene que ser más rápida que la M. P.
- Siempre es de menor capacidad que la M. P.

▪ Operación de la Memoria de Reserva

- La CPU lanza una dirección y busca en la M. de Reserva
 - Si está, toma su contenido y continúa
 - Si no está
 - * Accede a la M. Principal y toma el dato
 - * Trae de la M. Principal a la de Reserva un bloque (bloque que contiene la palabra accesada)

- **Diferencias entre M. Virtual y M. de Reserva**

- Memoria de Reserva
 - Tiene lo que se utiliza mas frecuentemente (datos e Instr.)
 - La CPU tiene acceso directo
 - Tiempo de acceso del orden de 5 a 10 veces mas pequeño
 - Tamaño de bloque pequeño (4 a 16 palabras)
 - La administración es totalmente Hardware
- Memoria Virtual
 - Tiene las partes de programa y datos no usados en ese momento
 - La CPU accede a través de un proceso de E/S
 - Tiempo de acceso del orden de 5 a 10 veces más grande
 - Tamaño de bloque grande (de 64 a 4K palabras)
 - La administración es parte Hardware y parte Software

- **Tasa de Acierto**

- Si la CPU encuentra la palabra en la M. de R. ----> Acierto
- Si la CPU no encuentra la palabra en la M. de R.—> Fallo

$$\textbf{Tasa Acierto} = n^{\circ} A / (A+F)$$

- **Ejemplo**

- M. Principal con tiempo de acceso 1000 ns
- M. de Reserva con tiempo de acceso de 100 ns
- Para una tasa de Acierto de 0.9 *tiempo de acceso medio de 200 ns*

- **Mapeo de Memoria**

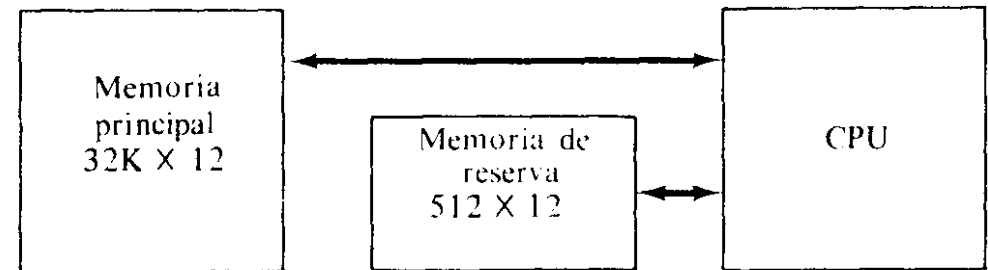
- Manera de relacionar las palabras de M. Principal y M. de Reserva
- Tipos
 - Mapeo Asociativo
 - Mapeo Directo
 - Mapeo Asociativo de Conjunto

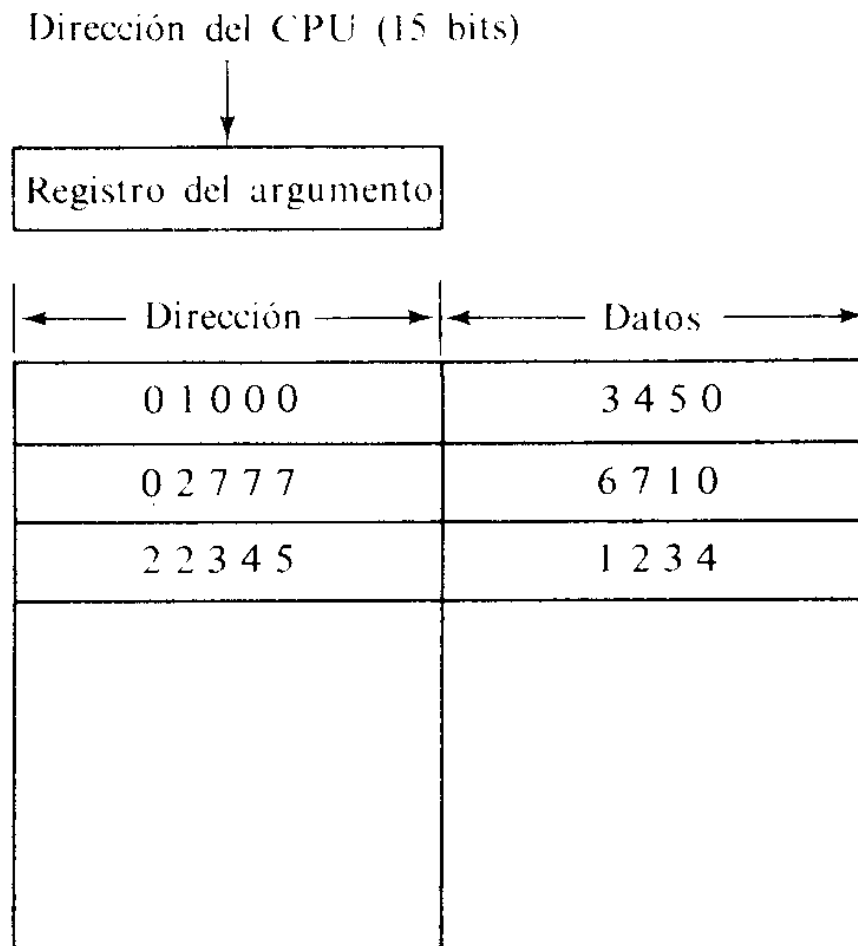
- **Ejemplo a utilizar**

EJEMPLO DE MEMORIA DE RESERVA

- **Mapeo Asociativo**

- Se implementa con una memoria Asociativa
 - Rápida pero costosa
- En cada posición de la M. de Reserva se almacena
 - Dirección
 - Datos de esa dirección
- La CPU lanza una dirección de 15 bits en el R. Argumento
 - Si se encuentra se toman los datos
 - Si no se encuentra
 - Se accede a M. Principal para traer dirección y datos
 - Algoritmo de reemplazo si la M. de R. está llena

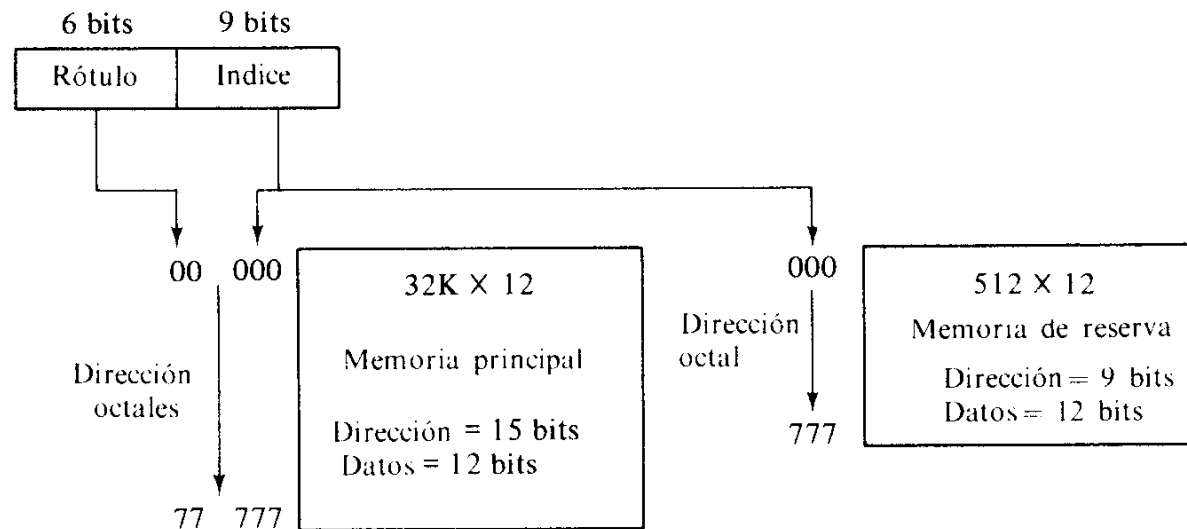




MAPEO ASOCIATIVO DE LA MEMORIA DE RESERVA

- **Mapeo directo**

- Se implementa con M. de acceso Aleatorio
 - Poco costosa y suficientemente rápida de acceso
 - No es muy eficiente
- La dirección se divide en dos campos
 - Campo índice (p.e. 9 bits)
 - * n° de bits que seleccionan cualquier posición de la M.R.
 - Campo rótulo o de identificación (ej. 6 bits)
- En cada posición M. de Reserva se almacena
 - Un campo para el rótulo
 - Resto para los datos
- Condición restrictiva de posicionar datos en la M. de Reserva
 - La dirección de M.R. es la que da los bits del campo Índice
 - No puede haber en M. R. dos direcciones del mismo Índice



RELACIONES DE DIRECCIONAMIENTO ENTRE M.P. Y M. R.

– La búsqueda

- Los bits del Índice seleccionan la dirección de M. R.
- Compara el rótulo de la dirección con rótulo en la M. R.
 - * Si coincide toma el dato
 - * Si no coincide se sustituye

—

Dirección de memoria	Datos de memoria
00000	1 2 2 0
00777	2 3 4 0
01000	3 4 5 0
01777	4 5 6 0
02000	5 6 7 0
02777	6 7 1 0

(a) Memoria principal

Dirección índice	Rótulo	Datos
000	0 0	1 2 2 0
777	0 2	6 7 1 0

(b) Memoria de reserva

MEMORIA DE RESERVA CON MAPEO DIRECTO

Tamaño de bloque

- Varias palabras en un mismo bloque
- Campos de una dirección
Rótulo-Índice(Bloque-Palabra)
- bits asociados a bloque
- bits asociados a palabra = posición en el bloque

– Búsqueda

- Se busca mediante los bits de Índice
- Se compara rótulos
 - * Si coinciden localiza la palabra
 - * Si no cambio el bloque completo

– Ventajas e inconvenientes

- Mayor probabilidad de acierto al tener N palabras seguidas
- Mayor tiempo de intercambio en caso de fallo

	Índice	Rótulo	Datos
Bloque 0	000	0 1	3 4 5 0
	007	0 1	6 5 7 8
Bloque 1	010		
	017		
Bloque 63	770	0 2	
	777	0 2	6 7 1 0

663

RótuloBloquePalabra

Índice

MAPEO DIRECTO CON BLOQUE DE 8 PALABRAS

▪ Mapeo Asociativo de Conjunto

- Toma ventajas del primero y del segundo
 - Parte con Mapeo Asociativo
 - Parte de Mapeo Directo
- Las posiciones de M.R. contienen varias palabras con distintos Rótulos
 - No es lo mismo que lo visto de bloque
- Búsqueda
 - Con el índice se selecciona la posición en M.R.
 - De manera Asociativa se comprueba si el rótulo esta
 - Si lo encuentra, toma el dato
 - Si no, lanza algoritmo de cambio (LRU o FIFO)

M. DE RESERVA CON MAPEO ASOCIATIVO
DE DOS CONJUNTOS

Indice	Rótulo	Datos	Rótulo	Datos
000	0 1	3 4 5 0	0 2	5 6 7 0
777	0 2	6 7 1 0	0 0	2 3 4 0

- **Escritura en la Memoria de Reserva**

- En un proceso de escritura en la M. de R. no esta involucrada la M.P.
 - Falta de coherencia en los datos al no coincidir
- Mecanismos de mantener la coherencia
 - Método de escritura directa
 - * Se actualiza en M. P. y en paralelo en M. de R.
 - * Mantiene la coherencia (necesaria en DMA)
 - Método de re-escritura
 - * Se actualiza solo la posición de M. de R.
 - * Se marca con un bit de bandera
 - * Al retirarla de la reserva se escribirá en M. P.
 - * Falta de coherencia

- **Iniciación de la Memoria de Reserva**

- Se parte de una situación de memoria vacía
 - bit asociado a cada posición indicando validez (0 = no válido)
 - Se va llenando a medida de la ocurrencia de fallos (1 = válido)

HARDWARE DE ADMINISTRACION DE LA MEMORIA

"Los Sistemas Multitareas y Multiusuarios implican una gran demanda de la Memoria del computador, lo que implica la necesidad de una administración"

- **Módulo de Manejo de Memoria**
 - Hardware de administración
 - Software de administración (Sistema Operativo)
- **Objetivos básicos de una Unidad de Administración de Memoria**
 - Facilidad de relocalización y traducción de D. Lógicas a Físicas
 - Compartición de programas comunes pos diferentes usuarios
 - Protección de acceso de la información
 - De otros usuarios
 - De las funciones del Sistema
- **Partición de los programas**
 - La página como unidad de partición
 - No tiene en cuenta coherencia lógica
 - p.e. parte una rutina en páginas distintas

- El segmento como Unidad de partición

"Conjunto de Instrucciones y/o Datos relacionados entre sí lógicamente"

- Se basa en cierta coherencia lógica
- p. e. una rutina, una matriz de datos, etc...
- Pueden ser generados por el programador o por el S.O.

- **Dirección lógica**

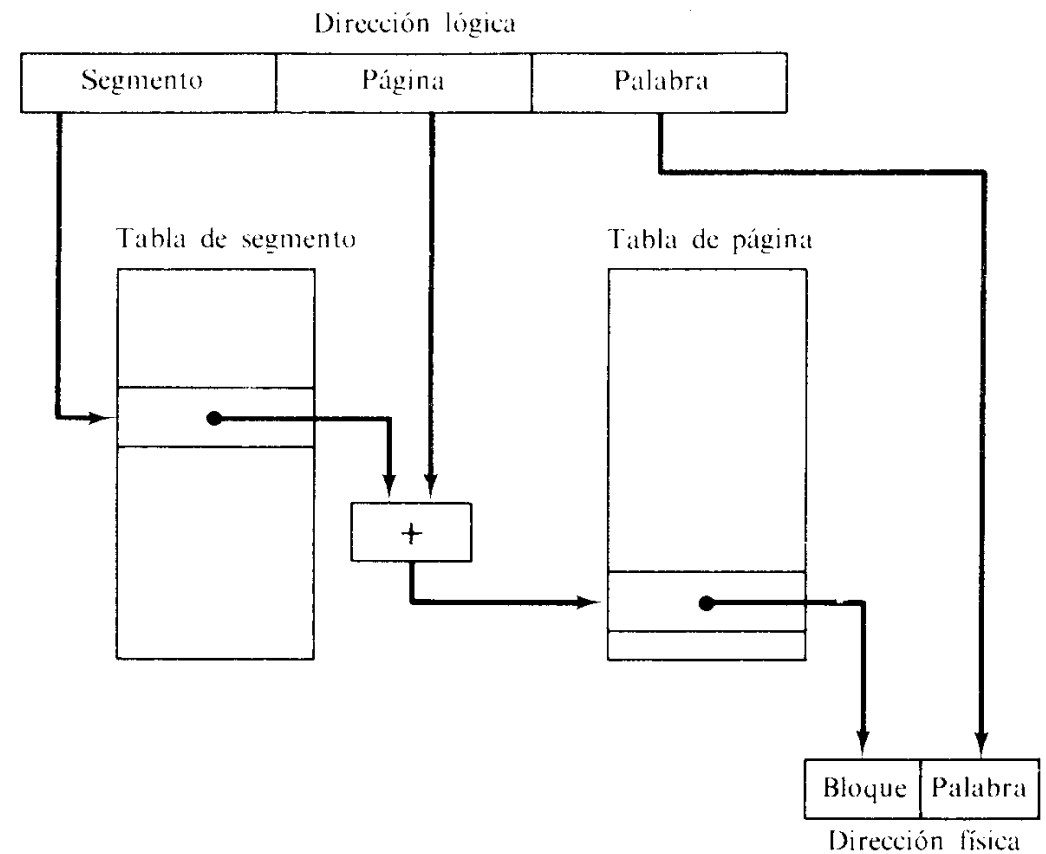
- Dirección generada por un programa segmentado
 - Similar a dirección virtual
- Diferencias con la dirección virtual
 - La longitud del segmento es variable, la de página es fija
 - La D. Lógica puede ser mayor, igual o menor que la D. Física

- **Mapeo de Direcciones lógicas a Direcciones Físicas**

- Mapeo de Página Segmentada
 - ***IBM370***
- Mapeo por Registros de Segmento
 - ***PDP 11, Z800: i80X86***

MAPEO DE PÁGINA SEGMENTADA

- **La longitud del segmento es variable**
 - Imposibilidad de predecir los bits necesarios de dirección
 - Se divide en páginas que sí tienen igual tamaño
 - La longitud de un segmento se asocia a un n°. al de páginas
- **Campos de la dirección lógica**
 - Campo de segmento (n° de segmento)
 - Campo de página (n° página del segmento)
 - Campo de palabra (posición en la página)



MAPEO DE DIRECCIÓN LÓGICA A FÍSICA

- **Mapeo de direcciones Lógicas a Físicas**

segmento-página-palabra =====> *bloque-palabra*

- **Tabla de segmento**

- Se direcciona por el n° de segmento
- Se obtiene la dirección de la página 0
 - Se le suma al dato anterior el n° de página
- Tabla de página
 - Se direcciona con el dato anterior
 - Se obtiene la dirección del bloque
- Ubicación de las tablas
 - En Memoria Aleatoria
 - * Dos memorias pequeñas adicionales
 - * En Memoria Principal
- Inconvenientes en M. Aleatoria
 - Cada dirección supone TRES accesos a memoria
 - Es muy lento
- En Memoria de tipo Asociativa
 - Como entrada de equiparación
 - * Segmento y página

- | Registro de argumento | | |
|-----------------------|--------|--------|
| Segmento | Página | Bloque |
| | | |

ÁREA DE CONOCIMIENTO DE ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES - DPTO. ELECTROTECNIA Y ELECTRÓNICA

- **Ejemplo**

- Dirección Lógica de 20 bits
 - 4 bits para n° segmento (de 0 a 15)
 - 8 para n° página en el segmento (de 0 a 255)
 - 8 para posición dentro de la página (de 0 a 255)
- Tamaño posible de los segmentos
 - Más pequeño: 1 sola página (256 palabras)
 - Más grande: 256 paginas (64K palabras)
- Dirección Física de 20 bits
 - 12 bits para el n° de bloque (de 0 a 2047)
 - 8 bits para posición dentro del bloque (de 0 a 255)
- Programa ejemplo
 - Un solo segmento de 5 páginas
 - Direcciones lógicas de la 60000 (hex) a la 604FF (hex)
 - Asignación de los bloques de memoria en la figura

Dirección hexadecimal	Número de página
60000	Página 0
60100	Página 1
60200	Página 2
60300	Página 3
60400	Página 4
604FF	

(a) Asignación de dirección lógica

Segmento	Página	Bloque
6	00	012
6	01	000
6	02	019
6	03	053
6	04	A61

(b) Página segmentada versus asignación de bloques de memoria.

ASIGNACIÓN DE MEMORIA LÓGICA A FÍSICA

Dirección lógica (en hexadecimal)

6	02	7E
---	----	----

TABLA DE SEGMENTOS Y PAGINAS, Y TABLA ASOCIATIVA

Tabla de segmento

0	
6	35
F	A3

Tabla de página

00	
35	012
36	000
37	019
38	053
39	A61
A3	012

Memoria física

00000	
000FF	Bloque 0
01200	
012FF	Bloque 12
01900	
0197E	← Palabra de 32 bits →
019FF	

(a) Tabla de mapeo de segmento y página

Segmento	Página	Bloque
6	02	019
6	04	A61

(b) Mapeo de memoria asociativa

- **Posibilidades del Sistema de Administración de Memoria**

- Asignar cualquier número de páginas a cada uno de los segmentos
- Cada página puede estar mapeada en cualquier bloque de la M. Física
- Las páginas pueden moverse a otros bloques en función de necesidades
 - Actualizar el n° de bloque en la Tabla de Página
- Los segmentos pueden crecer o decrecer sin afectar a los otros
- Segmentos diferentes pueden usar el mismo bloque
 - Compartir programas varios usuarios
 - En el ejemplo el bloque n° 12

MAPEO CON REGISTROS DE SEGMENTO

"Posibles situaciones en que la M. Física es mayor que la dirección que el programa puede direccionar directamente"

- N° de bits de la D. Física es mayor que el de la D. Lógica

PDP-11; 80X86; etc...

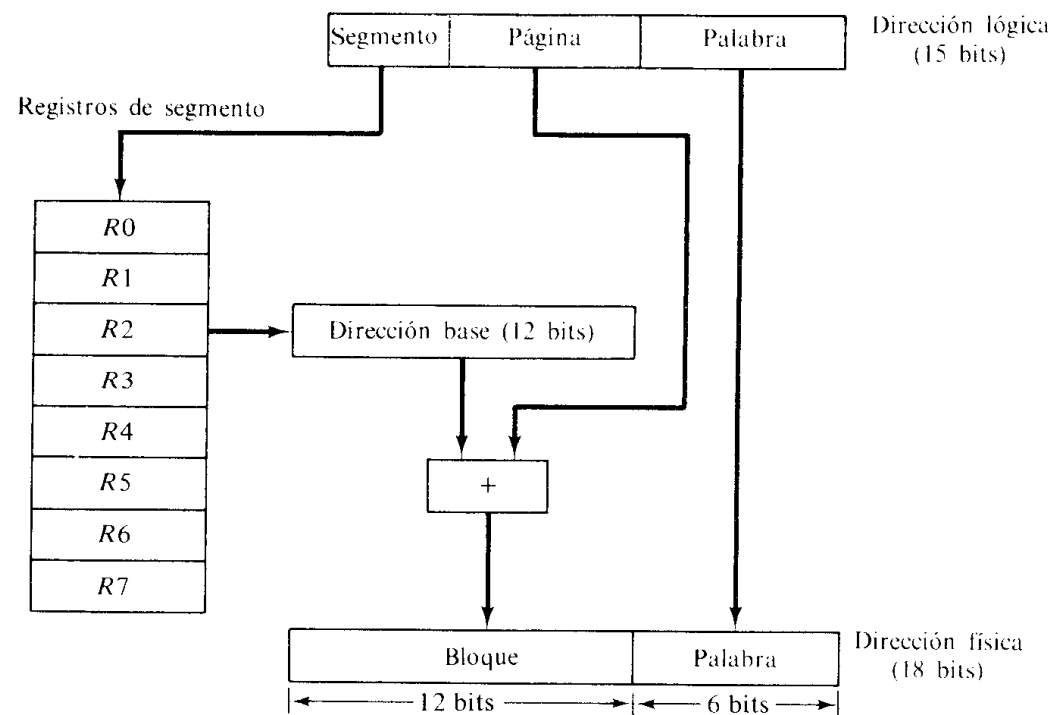
- **Mapeo de direcciones**

- Traduce dirección Lógica en Física
 - Mediante unos Registros de Segmentos
 - Deben de ser de acceso rápido (internos a la CPU)
- Campos de la dirección Lógica
 - Segmento: indica el registro referencia
 - Página: indica el n° de página
 - Palabra: indica posición en la página

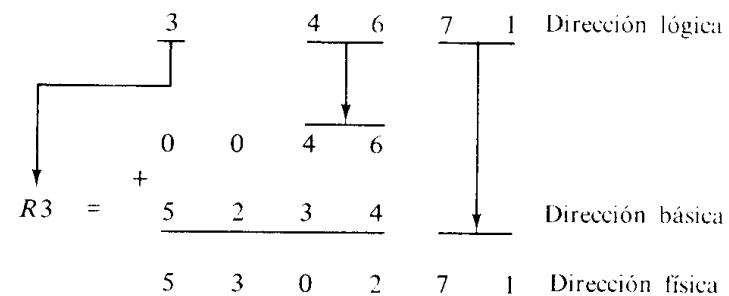
Contenido Reg. + n° pag. ==> n° de bloque

- Páginas seguidas corresponden a bloques seguidos

MAPEO CON REGISTROS DE SEGMENTO



(a) Mapeo de dirección de lógico a físico



(b) Ejemplo numérico (todos los números están en octal)

PROTECCIÓN DE LA MEMORIA

▪ Dónde se establece la protección

- En la Dirección Física
 - Cada Bloque tiene asignados unos bits de protección
 - Al mover una página de bloque se actualizan esos bits
- En la Dirección Lógica
 - Se incluye información en la tabla o Registro de segmento

▪ Descriptor :

"Contenido de cada entrada de una Tabla de Segmento o de un Registro de Segmento"

| Dirección base | Longitud | Protección |

- Campos de un descriptor típico
 - Dirección base
 - * dirección base de Tabla de página (página segmentada)
 - * dirección de bloque base (registro de segmento)
 - Longitud, da el tamaño del segmento
 - * Su comparación con n° de página evita accesos fuera
 - Protección; da derechos de acceso a ese segmento
 - * Para página segmentada cada una llevará su protección

- **Derechos de acceso típicos**
 - Lectura completa y privilegios de escritura
 - Solo lectura (protección de escritura)
 - Ejecución solamente (protección de programa)
 - Sistema solamente (protección de la operación del sistema)