Grado en
Ingeniería Informática

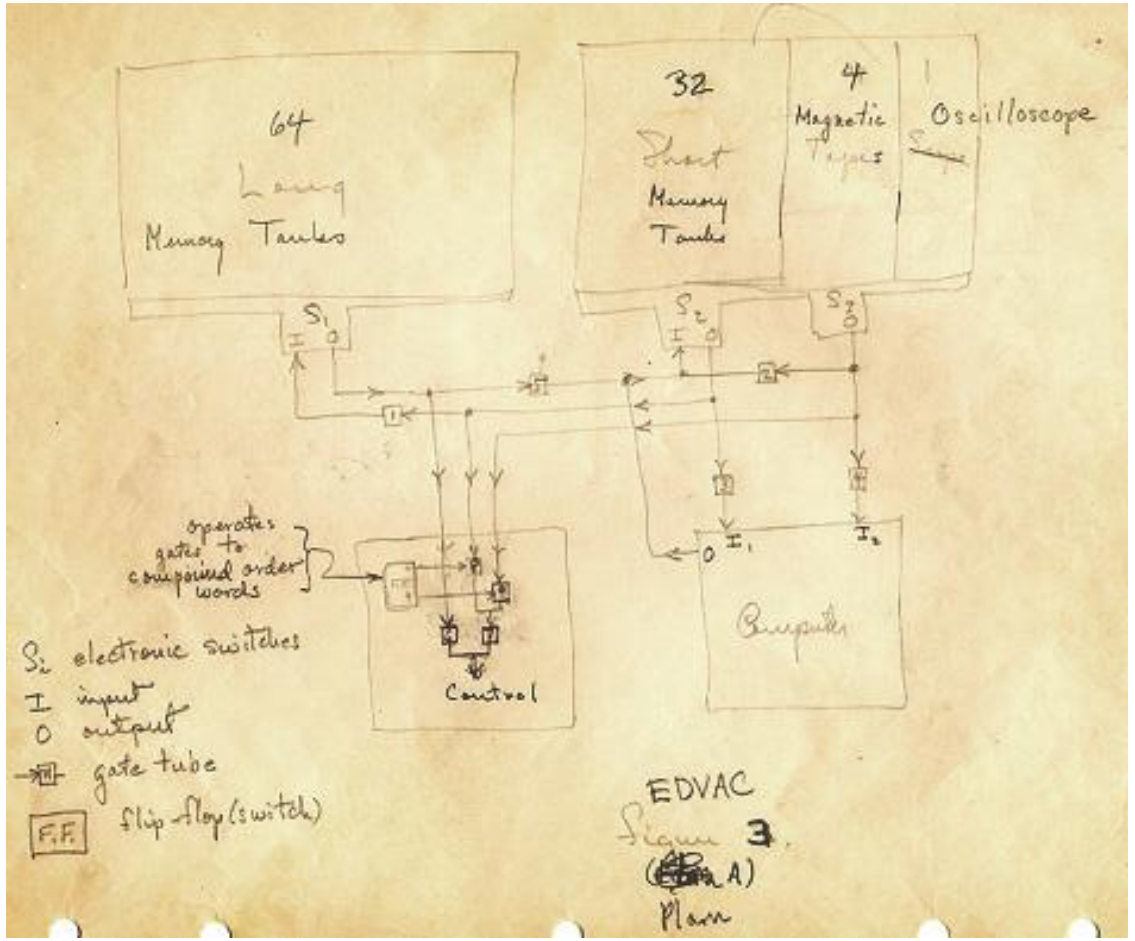
Tema 1: Introducción a los computadores

Unidad 2: Arquitectura Von Neumann

Rafael Casado González
Rosa María García Muñoz
María Teresa López Bonal
Universidad de Castilla–La Mancha

Arquitectura Von Neumann

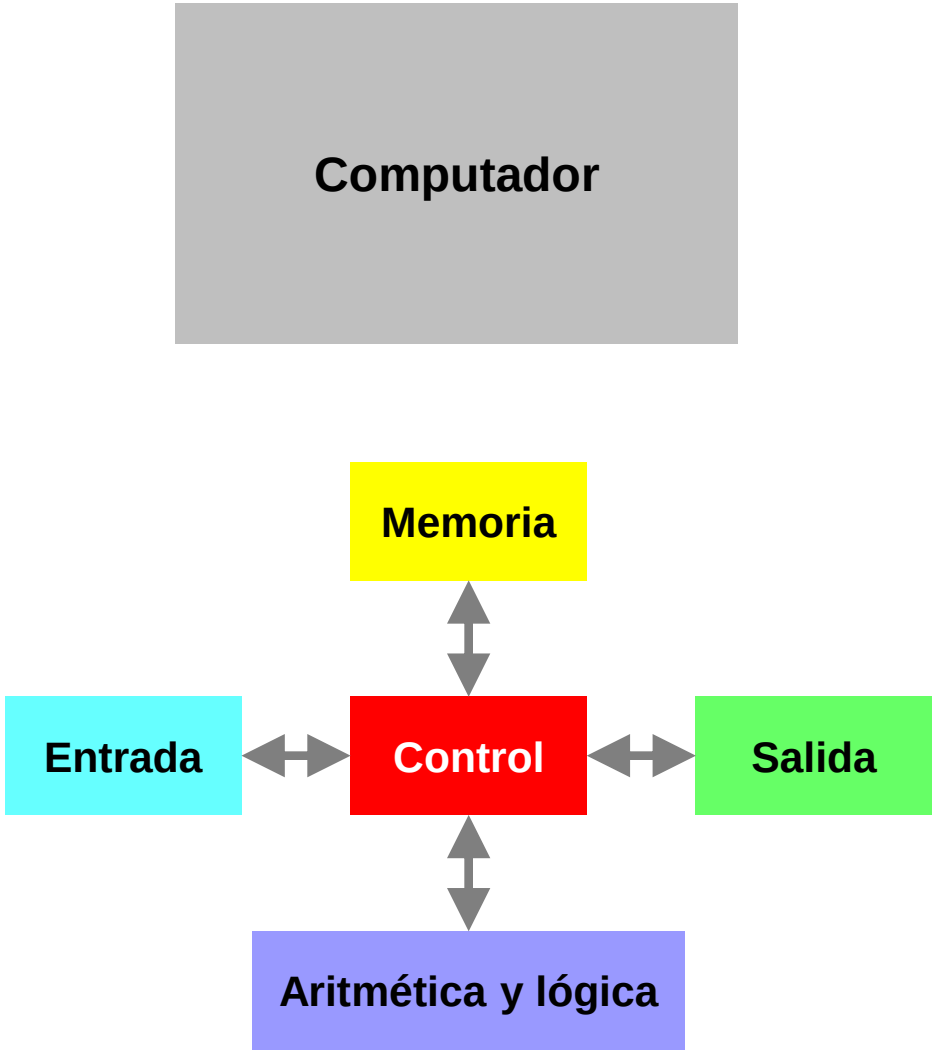
- La arquitectura de un computador define su **comportamiento funcional**
- El modelo básico de arquitectura empleado en los computadores fue establecido en 1945 por Von Neumann
  - Esta arquitectura (aunque evolucionada) se sigue empleando en la mayoría de los ordenadores actuales



Arquitectura Von Neumann

- Antes de Von Neumann
  - Los programas eran cableados
  - No existía una memoria

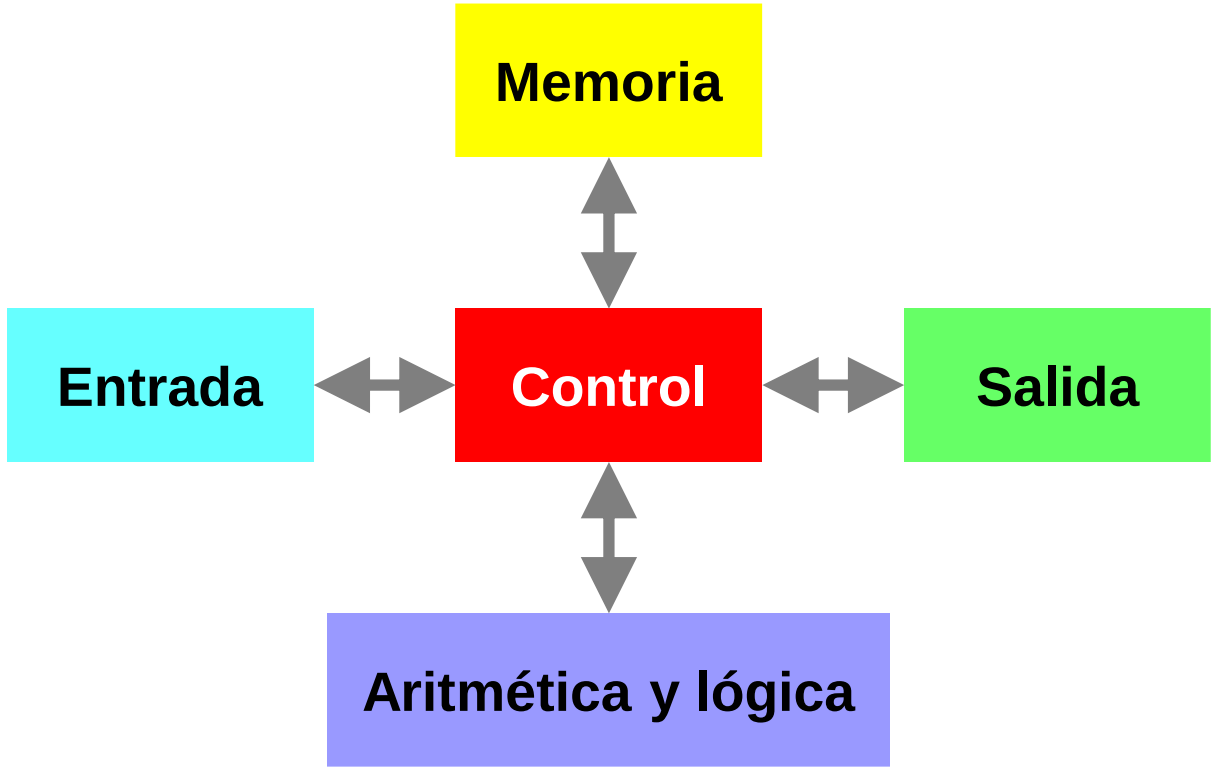
- Von Neumann introdujo
  - Diversas unidades **funcionales** independientes e interconectadas
  - Entre ellas una memoria para almacenar programas y datos



Arquitectura Von Neumann

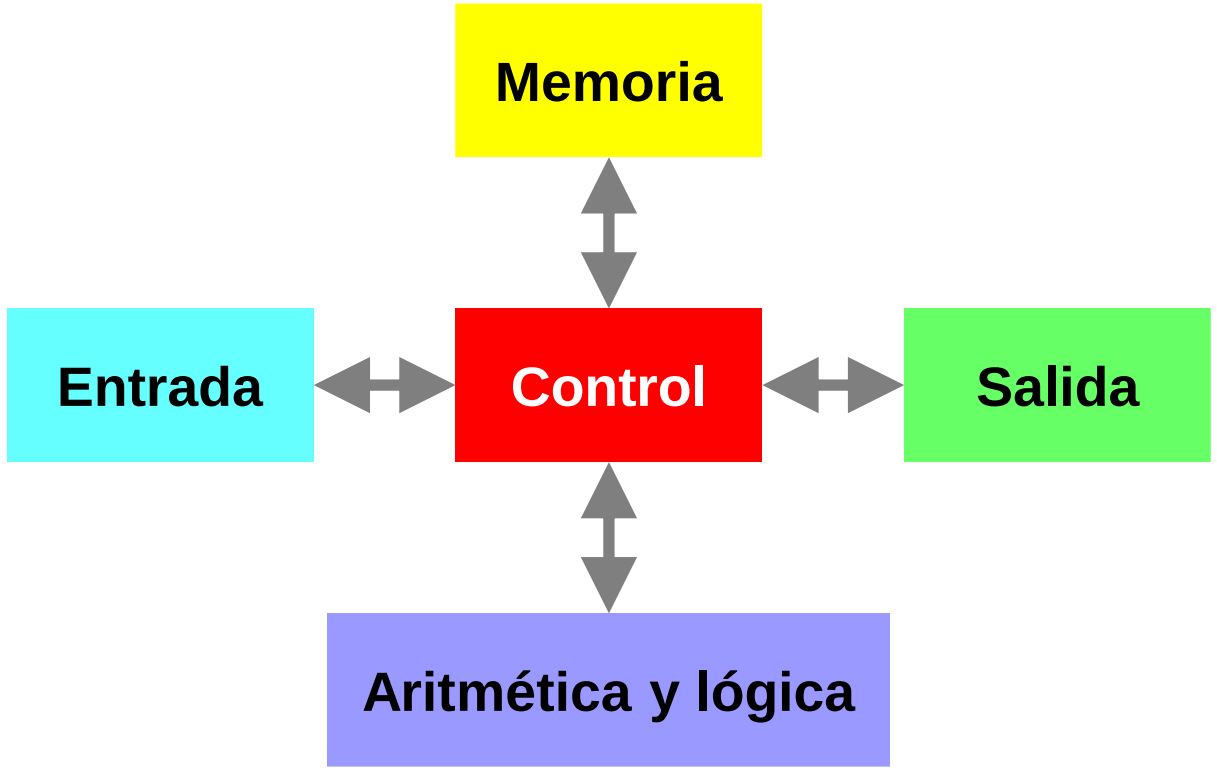
- Unidad de entrada
  - A través de ella, la computadora acepta los datos e instrucciones que provienen del exterior

- Unidad de salida
  - Proporciona mensajes y resultados al exterior de las operaciones que realiza la computadora



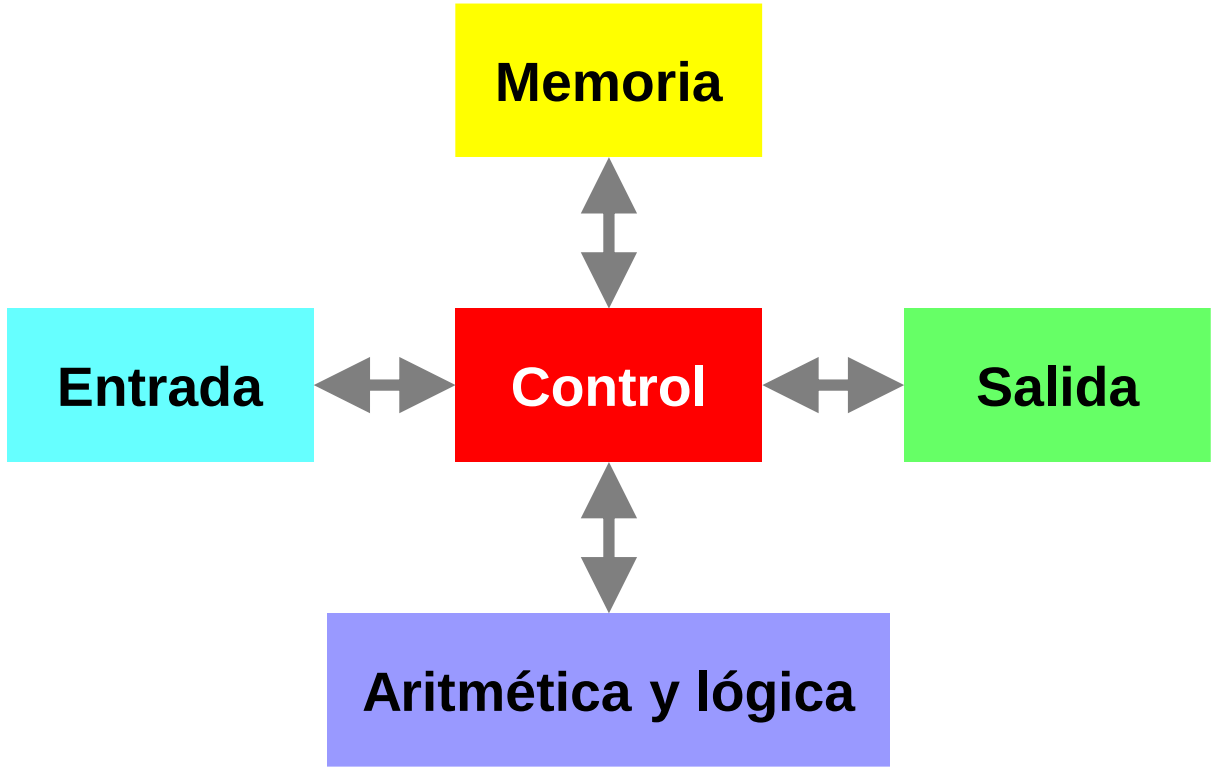
Arquitectura Von Neumann

- Las unidades de E/S incluyen
  - Interfaz con el usuario
    - monitor, teclado, ratón,…
  - Almacenamiento externo
    - Disco Duro, CD/DVD, Flash,…
  - Comunicaciones
    - Adaptadores de red
    - WiFi, Bluetooth,…



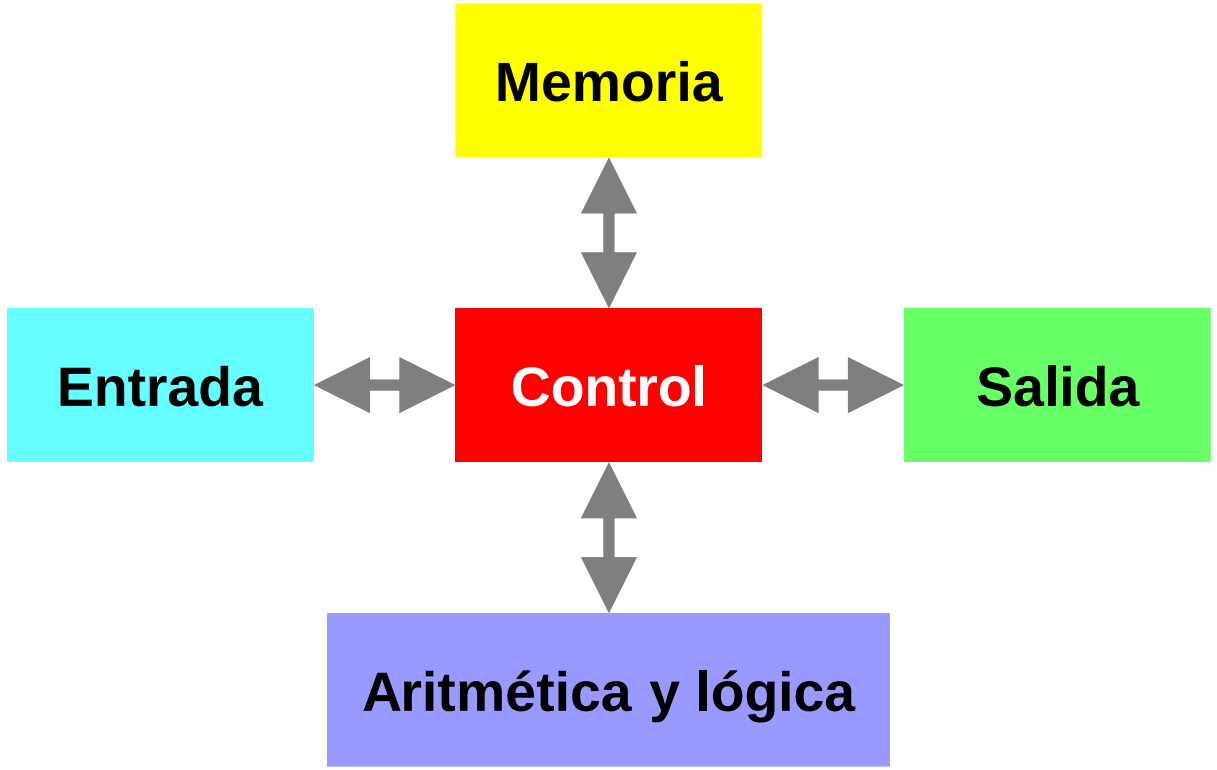
Arquitectura Von Neumann

- Unidad de memoria
  - Almacena
    - programas que gobiernan las operaciones a realizar por la computadora
    - datos necesarios para dichos cálculos
  - Diferentes tipos
    - Principal (rápida pero escasa)
    - Secundaria (Abundante pero lenta)



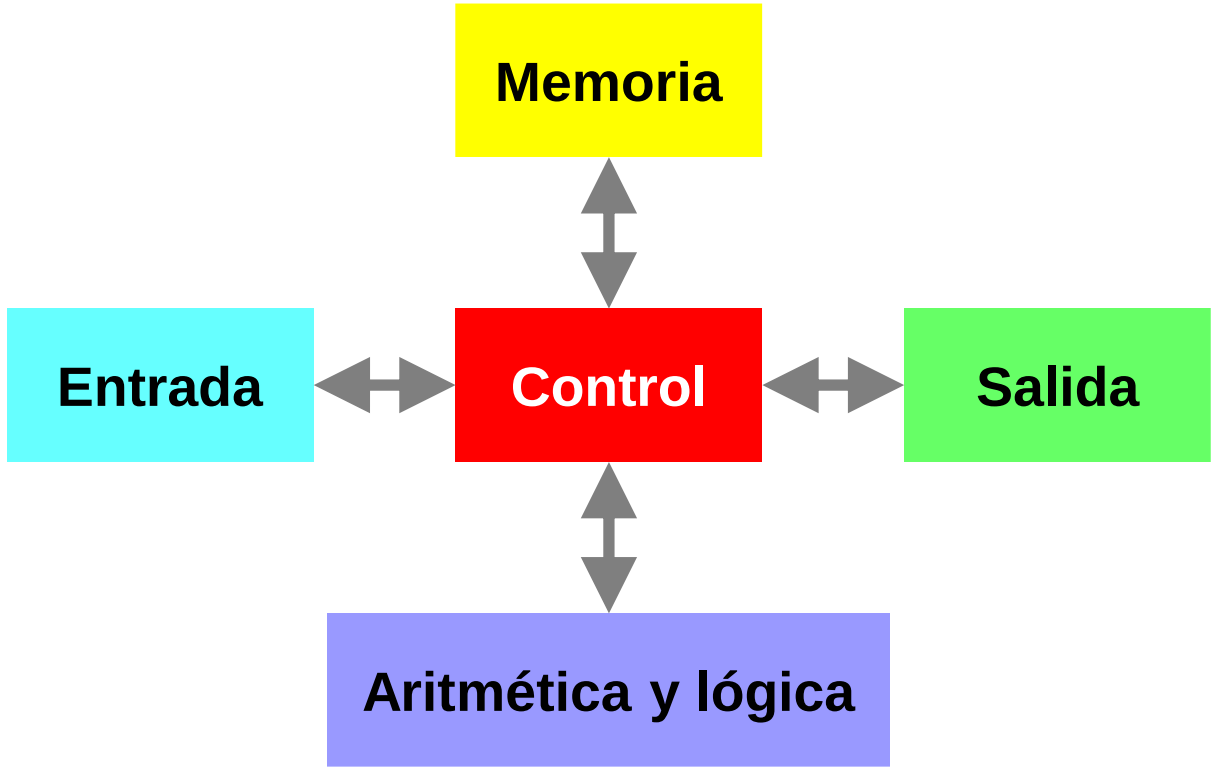
Arquitectura Von Neumann

- Unidad aritmético-lógica (ALU)
  - Contiene los circuitos encargados de realizar operaciones aritméticas
    - SUMA
    - RESTA
    - …
  - y operaciones lógicas
    - AND
    - OR
    - …

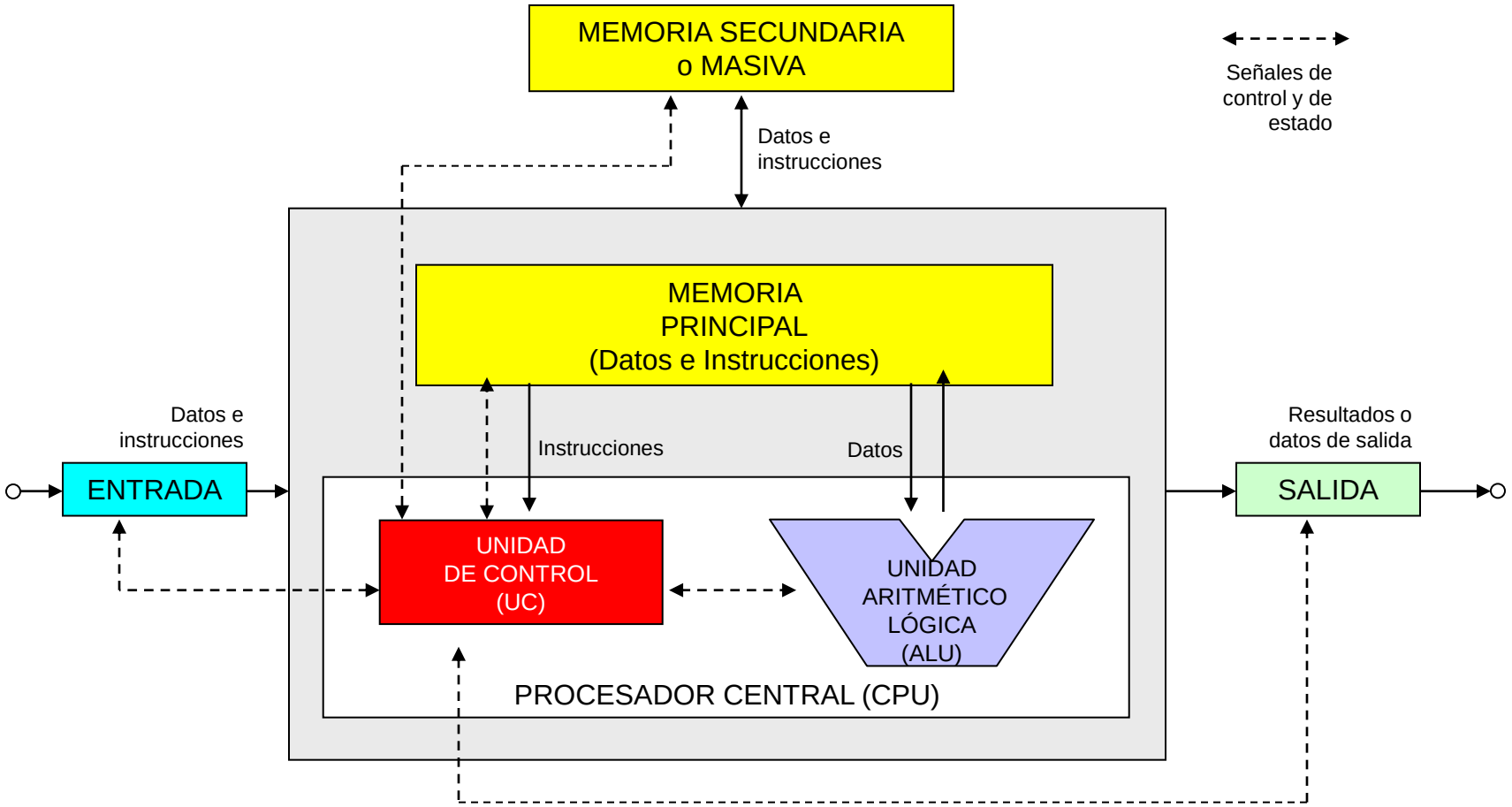


Arquitectura Von Neumann

- Unidad de control (UC)
  - Recibe señales de estado de las otras unidades
  - Envía señales de control a las otras unidades para coordinar su funcionamiento



Arquitectura Von Neumann



Arquitectura Von Neumann

- Ruta de datos
  - Conjunto de elementos que intervienen en el procesamiento de los datos durante la ejecución de las instrucciones
  - Unidades aritmético-lógicas (ALUs)
  - Buses
  - Registros

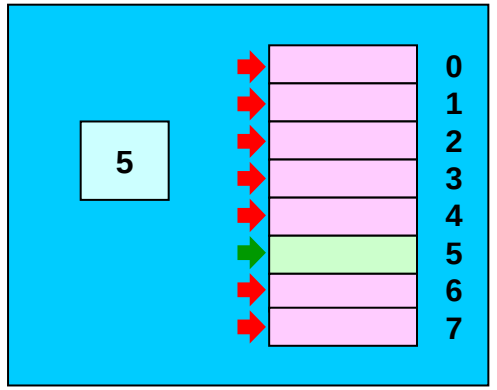
Unidad de memoria

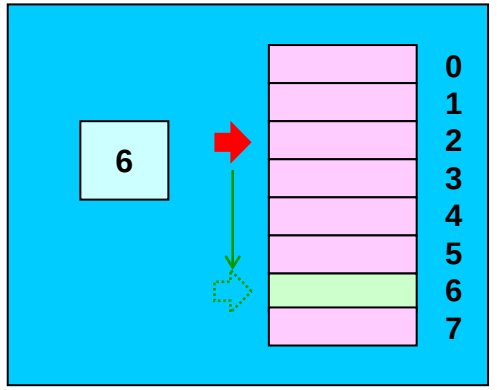
Unidades de almacenamiento

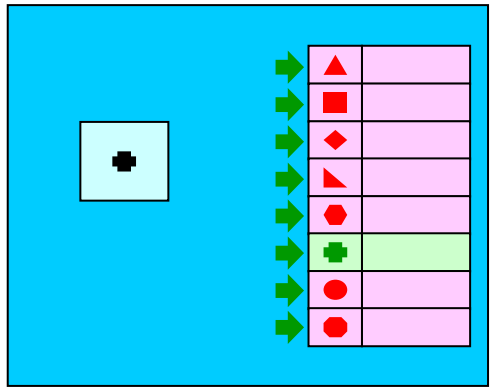
sistema internacional (SI)			sistema binario (ISO/IEC)				%
factor	nombre	abr.	factor	nombre	abr.	cantidad de bytes	
$(10^3)^1$	kilobyte	kB	$(2^{10})^1$	kibibyte	KiB	1.024	2.4%
$(10^3)^2$	megabyte	MB	$(2^{10})^2$	mebibyte	MiB	1.048.576	4.9%
$(10^3)^3$	gigabyte	GB	$(2^{10})^3$	gibibyte	GiB	1.073.741.824	7.4%
$(10^3)^4$	terabyte	TB	$(2^{10})^4$	tebibyte	TiB	1.099.511.627.776	10.0%
$(10^3)^5$	petabyte	PB	$(2^{10})^5$	pebibyte	PiB	1.125.899.906.842.624	12.6%
$(10^3)^6$	exabyte	EB	$(2^{10})^6$	exbibyte	EiB	1.152.921.504.606.846.976	15.3%
$(10^3)^7$	zetabyte	ZB	$(2^{10})^7$	zebibyte	ZiB	1.180.591.620.717.411.303.424	18.1%
$(10^3)^8$	yotabyte	YB	$(2^{10})^8$	yobibyte	YiB	1.208.925.819.614.629.174.706.176	20.9%

Unidad de memoria
Modo de acceso a la información

- Acceso aleatorio
  - Se accede a la posición indicada sin necesidad de pasar por todas las que hay delante
- Acceso secuencial
  - Para leer una posición de memoria es necesario pasar por todas las anteriores
- Memoria asociativa
  - En vez de usar la dirección se usa el propio dato a buscar (o parte del dato)







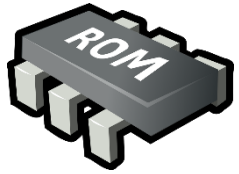
Unidad de memoria
Volatilidad de la información

- Memoria RAM
  (Random Access Memory)
  - Memoria de acceso aleatorio
  - Permite operaciones
    de lectura y escritura

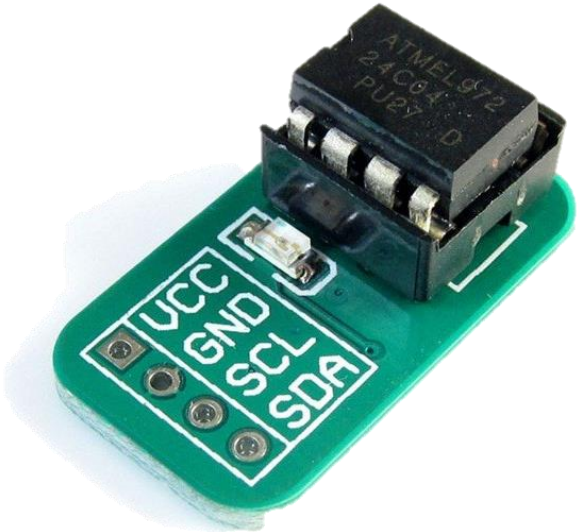
  - Su contenido permanece sólo
    mientras esté presente la tensión de
    alimentación (es volátil)

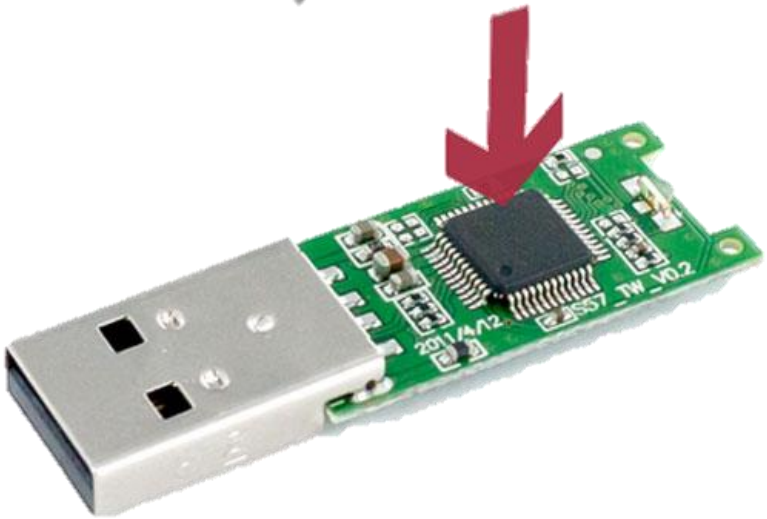
Unidad de memoria
Volatilidad de la información

- Memoria ROM
  (Read Only Memory)



  - Memoria de sólo lectura
    - No permite escritura
  - ¡Acceso aleatorio también!
  - Su contenido permanece sin tensión de alimentación
  - Variantes con escritura
    - PROM (Programmable ROM)
    - EPROM (Erasable PROM)
    - EEPROM (Electrically EPROM)
    - FLASH (EEPROM mejorada)



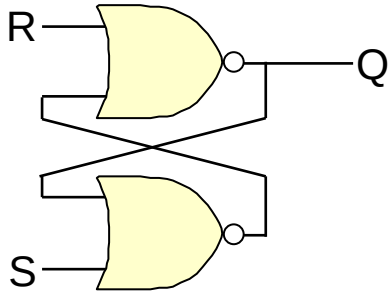


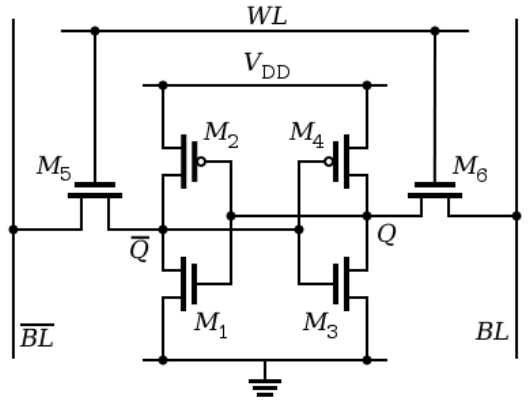
Unidad de memoria
Tecnologías

- La memoria está compuesta por celdas de 1 bit

- SRAM (Static RAM)
  - El valor almacenado en una celda se mantiene en un par de puertas inversoras
  - Tacceso = 2 – 25 nanoseg
  - Coste = 100 – 250 $/MByte
  - Tamaño = 4 a 6 transistores

Latch SR		
S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Oscilación

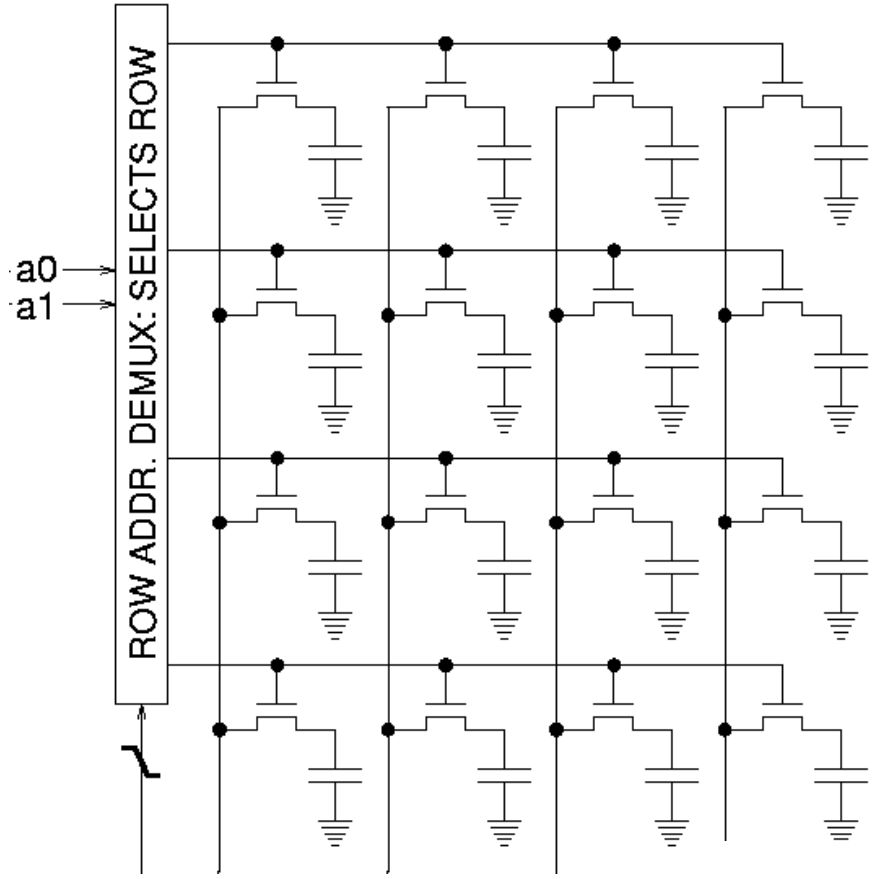




Pentium II
Memoria caché
de tecnología SRAM

Unidad de memoria
Tecnologías

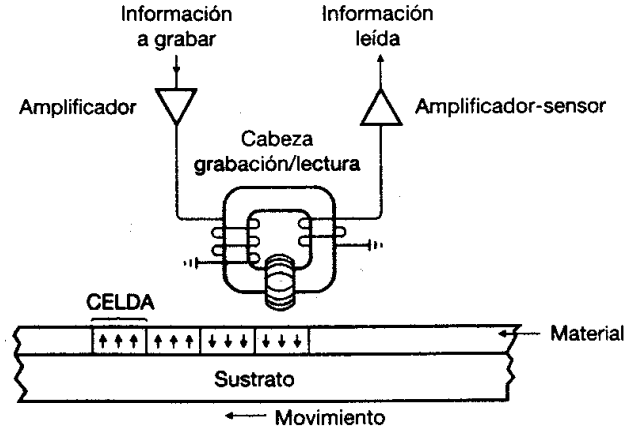
- DRAM (Dynamic RAM)
  - El valor almacenado en una celda se mantiene como una carga en un condensador
  - Requiere un refresco
  - Tacceso = 60 – 120 nanoseg
  - Coste = 5 – 10 $/MByte
  - Tamaño = 1 transistor

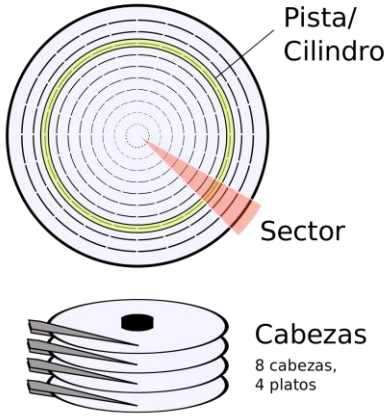


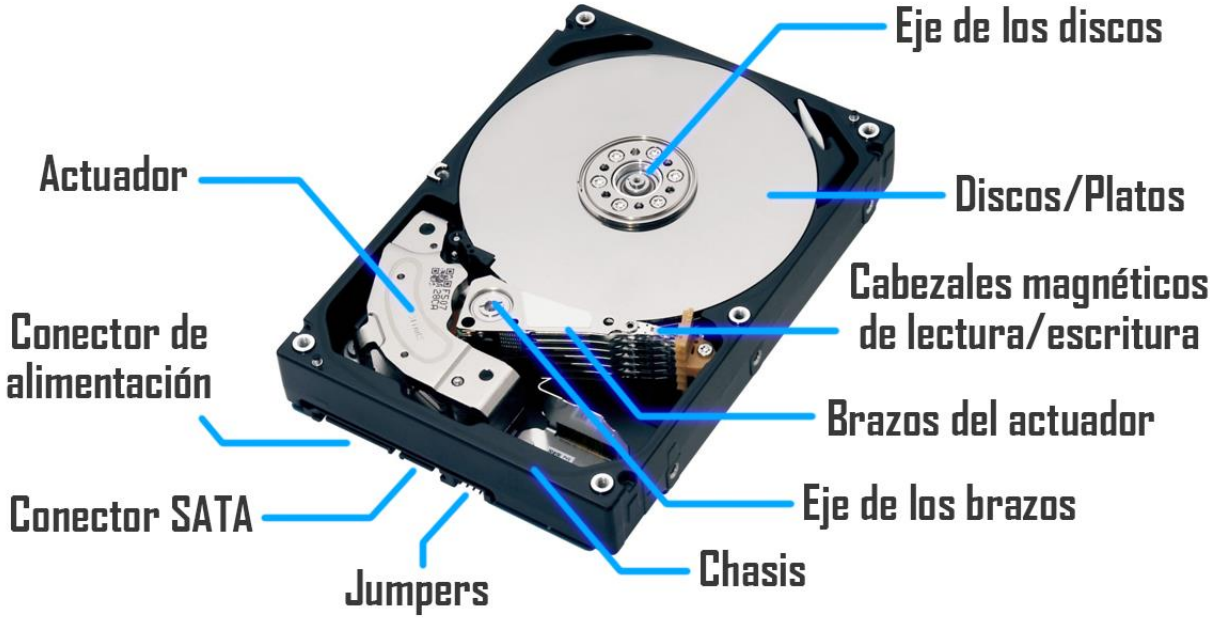
Módulos DIMM
1GB de memoria DRAM

Unidad de memoria
Tecnologías

- Discos magnéticos
  - El valor almacenado en una celda se mantiene como una superficie orientada magnéticamente
  - Tacceso = 10 – 20 miliseg
  - Coste = 2 – 5 $/GByte

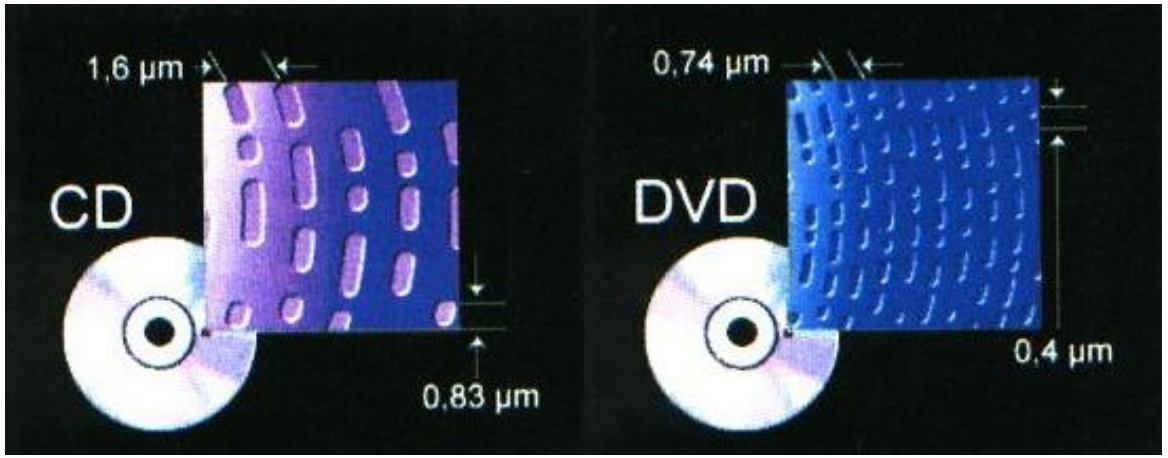


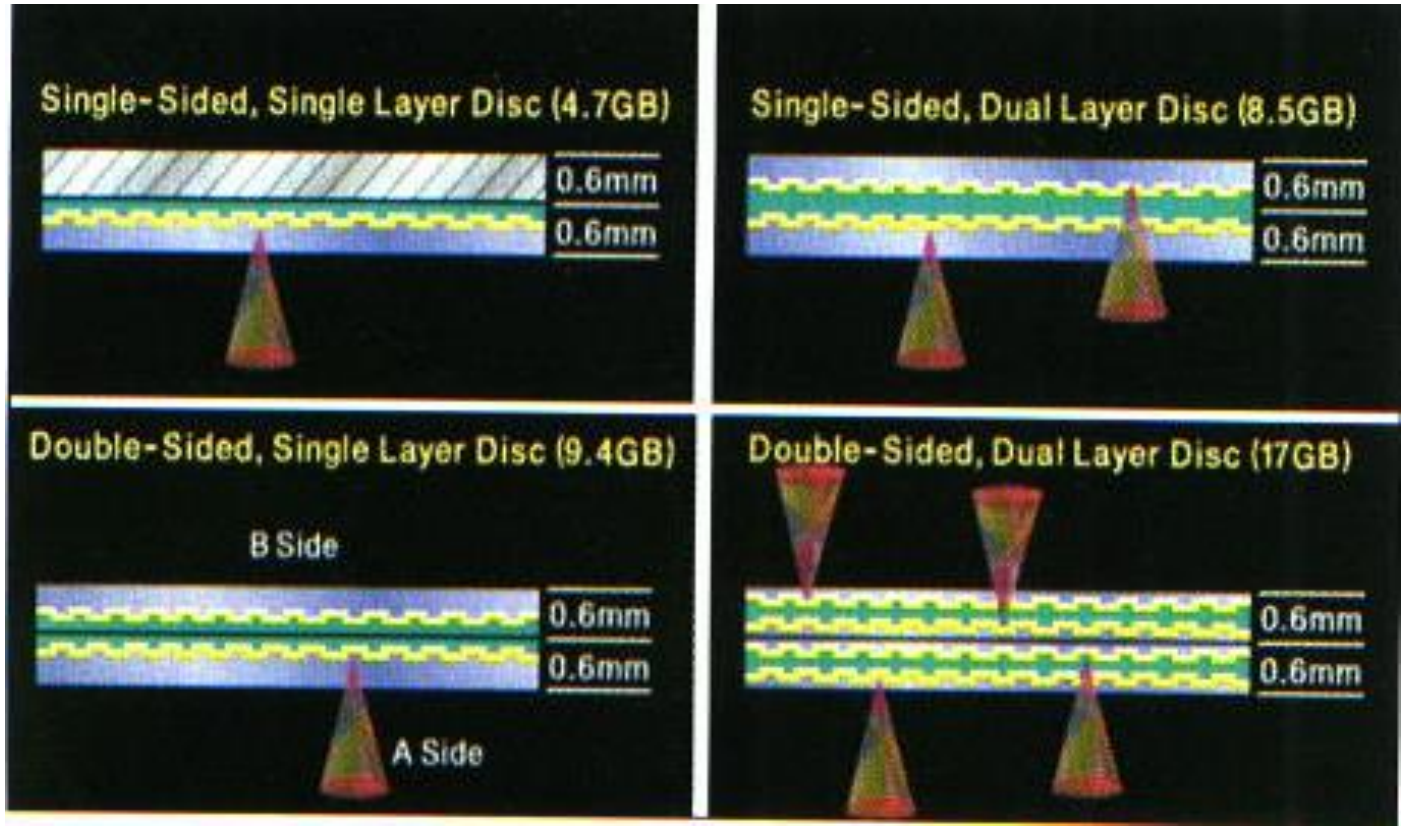




Unidad de memoria
Tecnologías

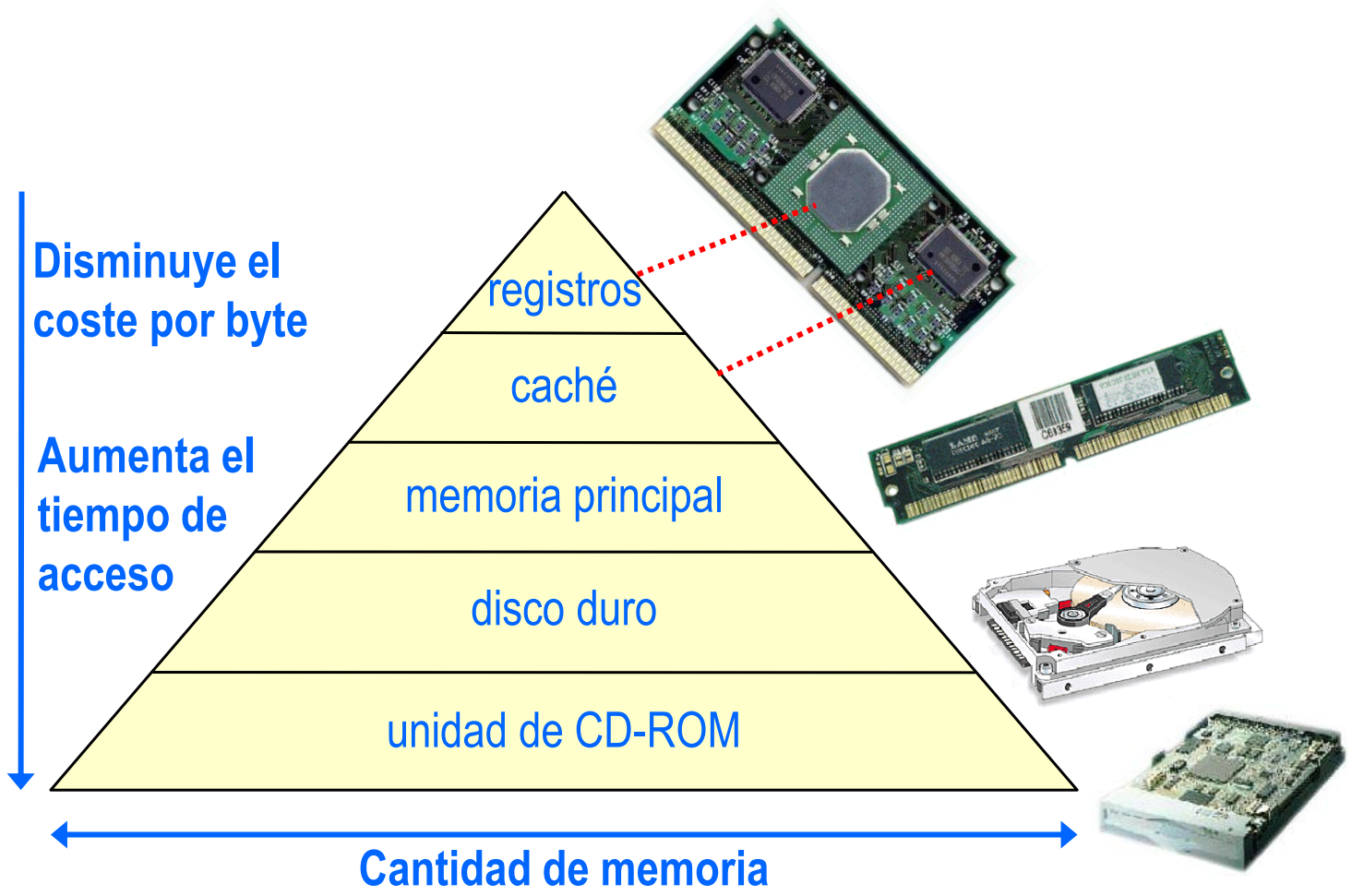
- Discos ópticos
  - El valor almacenado en una celda se define con la presencia o ausencia de un agujero en una superficie





Unidad de memoria
Jerarquía

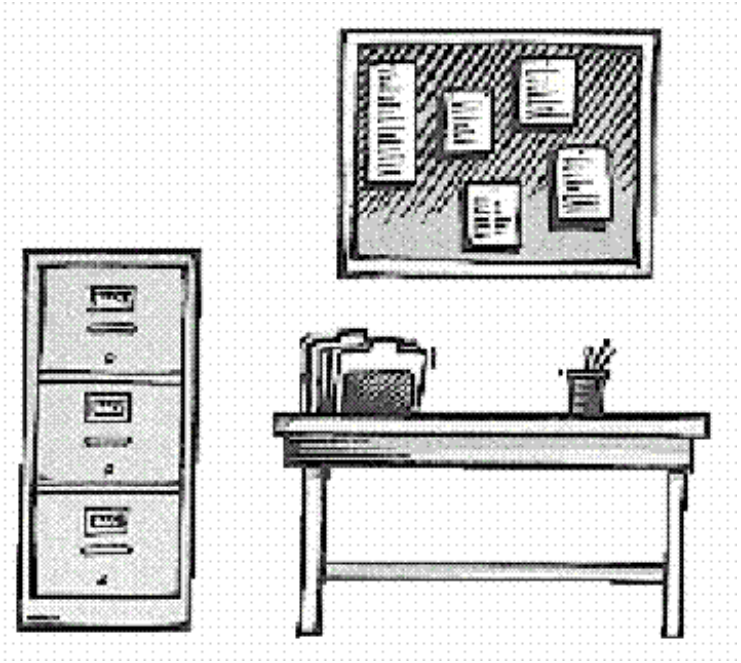
- La memoria depende de múltiples parámetros
  - Volatilidad
  - Velocidad
  - capacidad
  - precio
- Por ello, la memoria de un computador suele ser heterogénea, distribuida según la siguiente estructura jerárquica

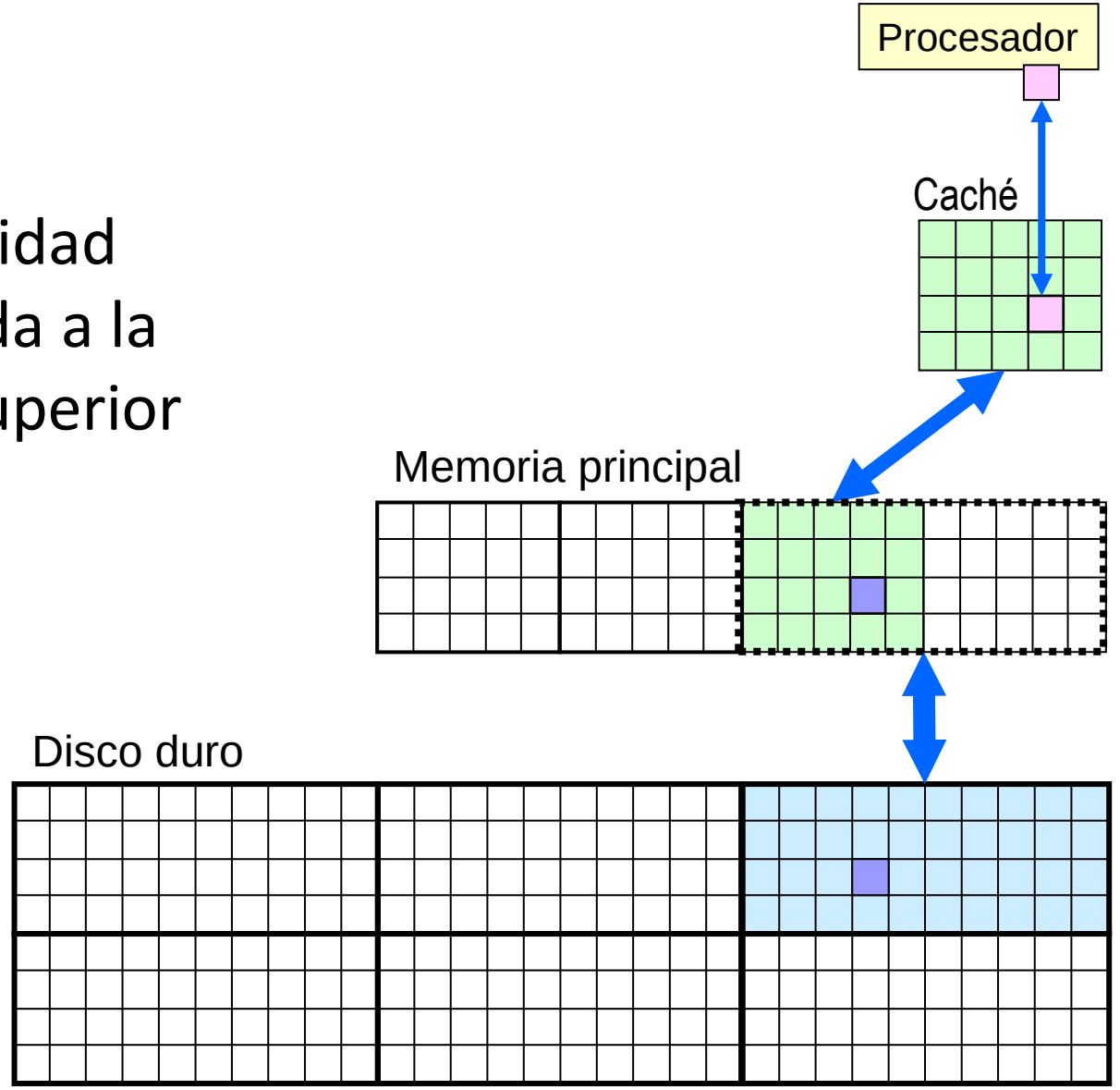


Unidad de memoria

Jerarquía

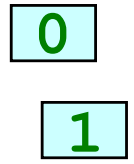
- Esta jerarquía proporciona la capacidad de la memoria de nivel inferior unida a la velocidad de la memoria de nivel superior



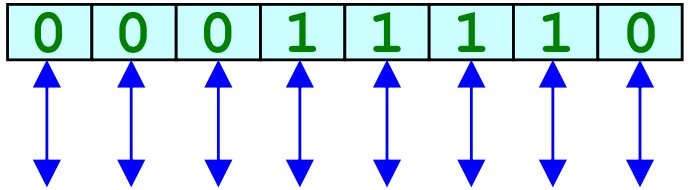


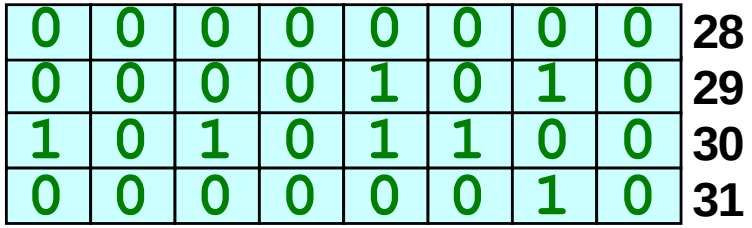
Unidad de memoria
Estructura interna

- Se estructura en celdas capaces de almacenar un bit



- Estas celdas se procesan en grupos de tamaño fijo denominados palabras
  - Transferidas en paralelo (en la misma operación)



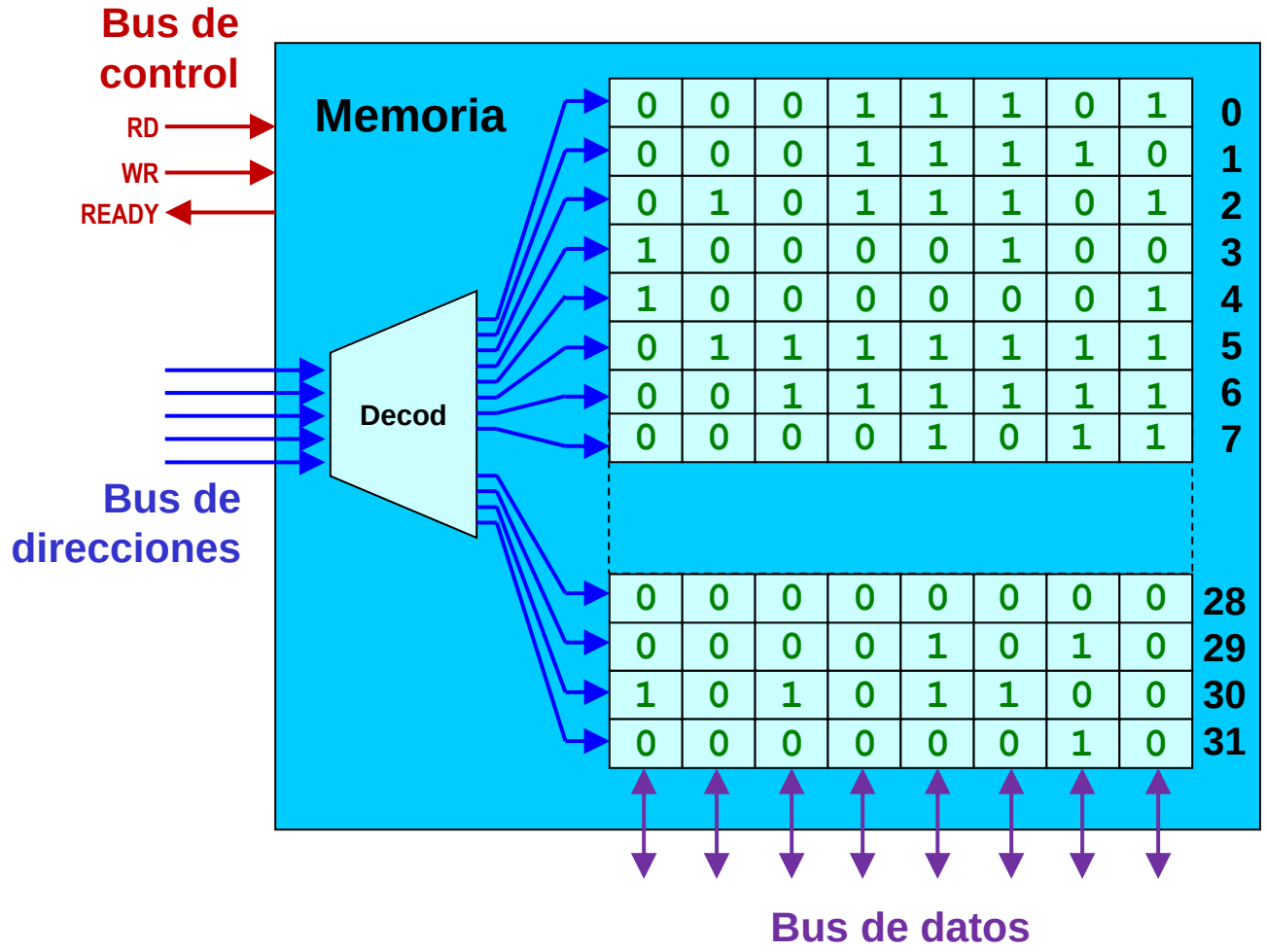


- Para poder acceder a las palabras, éstas se identifican por una dirección

Unidad de memoria

Estructura interna

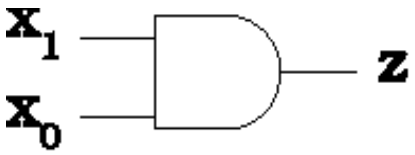
- Operación a realizar
  - Señal RD (lectura)
  - Señal WR (escritura)
- Indicación de operación terminada
  - Señal READY
- Estas líneas forman parte del bus de control



Unidad aritmético-lógica

Puertas lógicas

AND

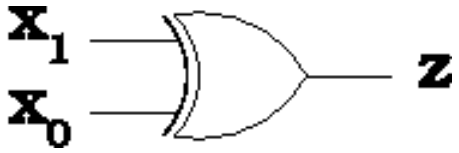


AND		
X0	X1	Z
0	0	0
0	1	0
1	0	0
1	1	1

OR

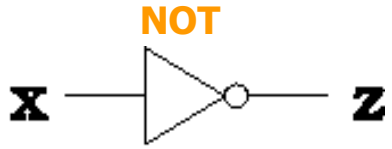
OR		
X0	X1	Z
0	0	0
0	1	1
1	0	1
1	1	1

XOR



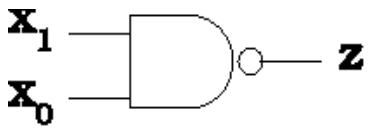
XOR		
X0	X1	Z
0	0	0
0	1	1
1	0	1
1	1	0

NOT

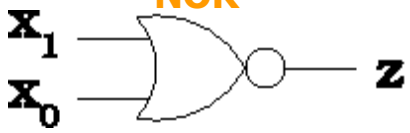


NOT	
X0	Z
0	1
1	0

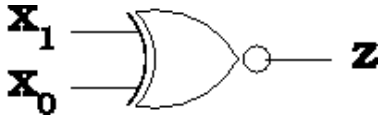
NAND



NOR

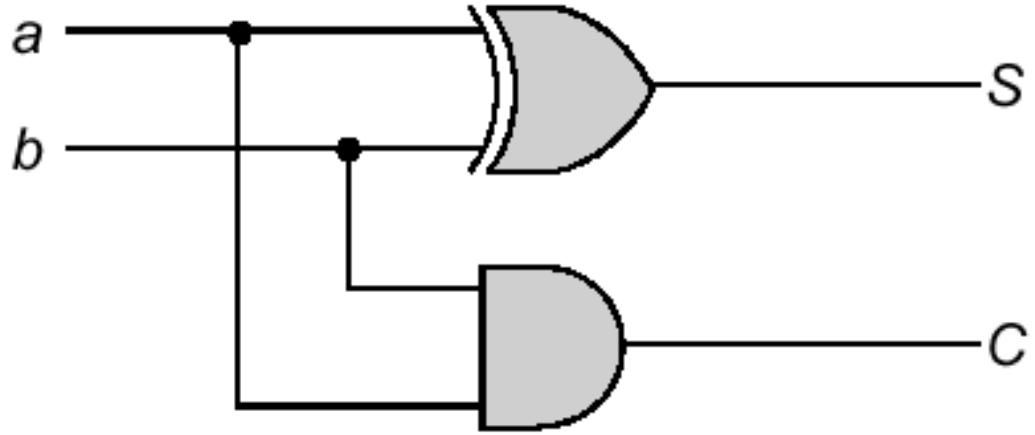


NXOR



Unidad aritmético-lógica

Circuito semisumador



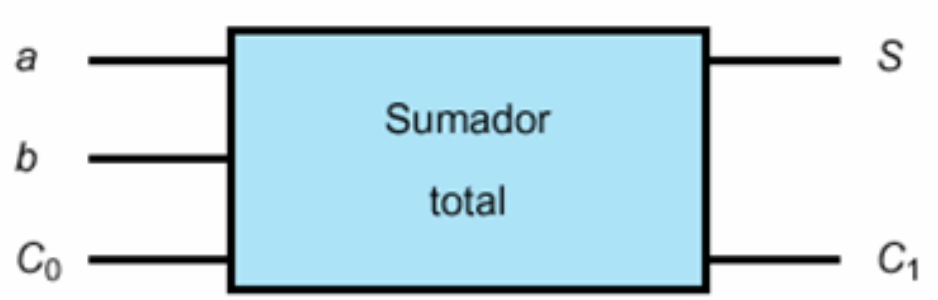
A	B	Suma (S)	Acarreo
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = a \oplus b$

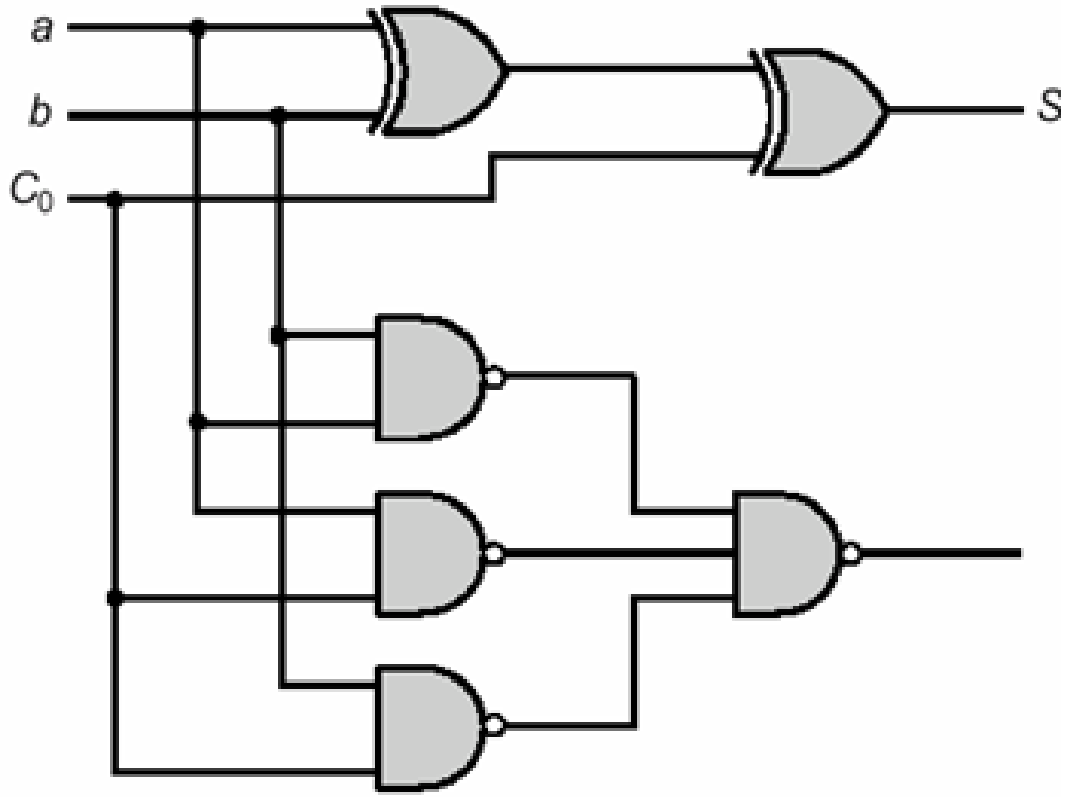
$C = a \cdot b$

Unidad aritmético-lógica

Sumador total

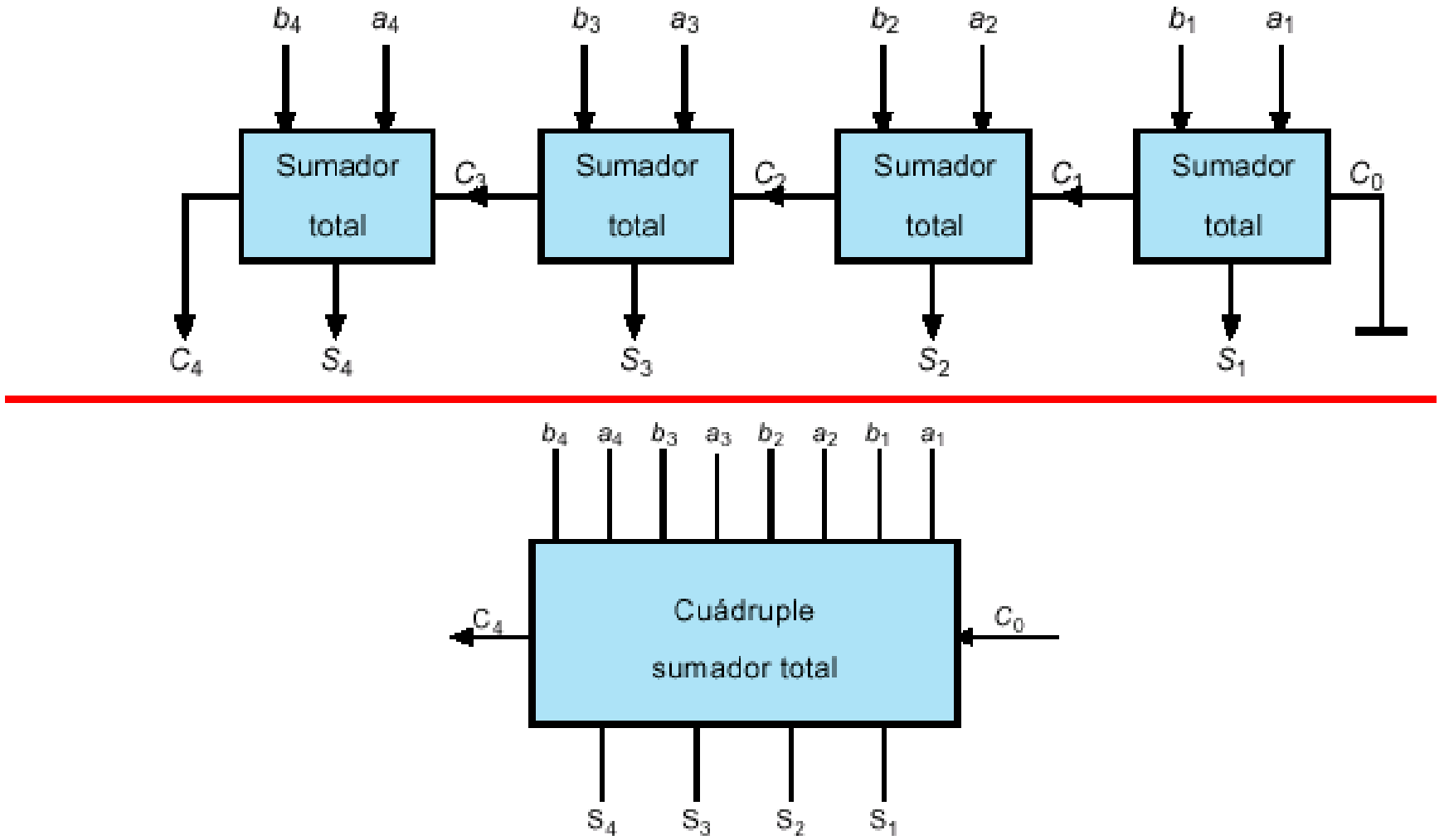


a	b	C_0	S	C_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Unidad aritmético-lógica

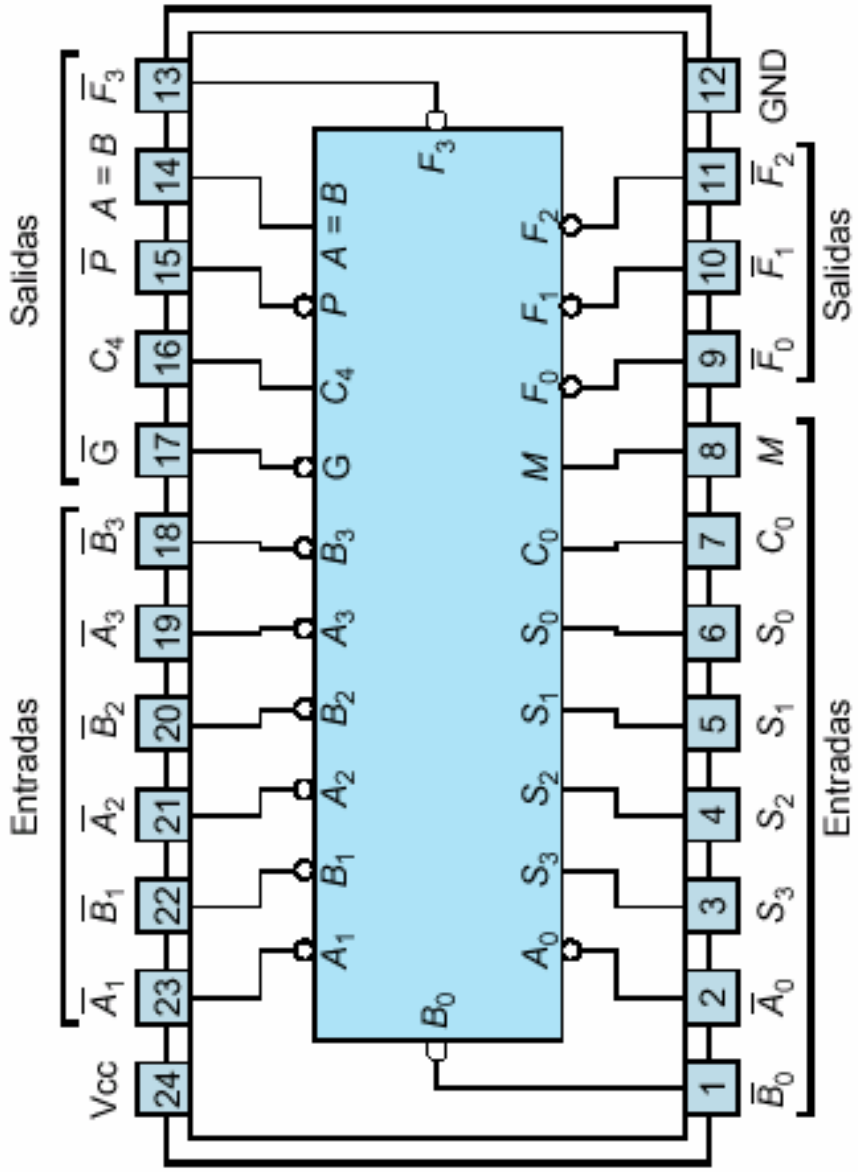
Sumador total



Unidad aritmético-lógica

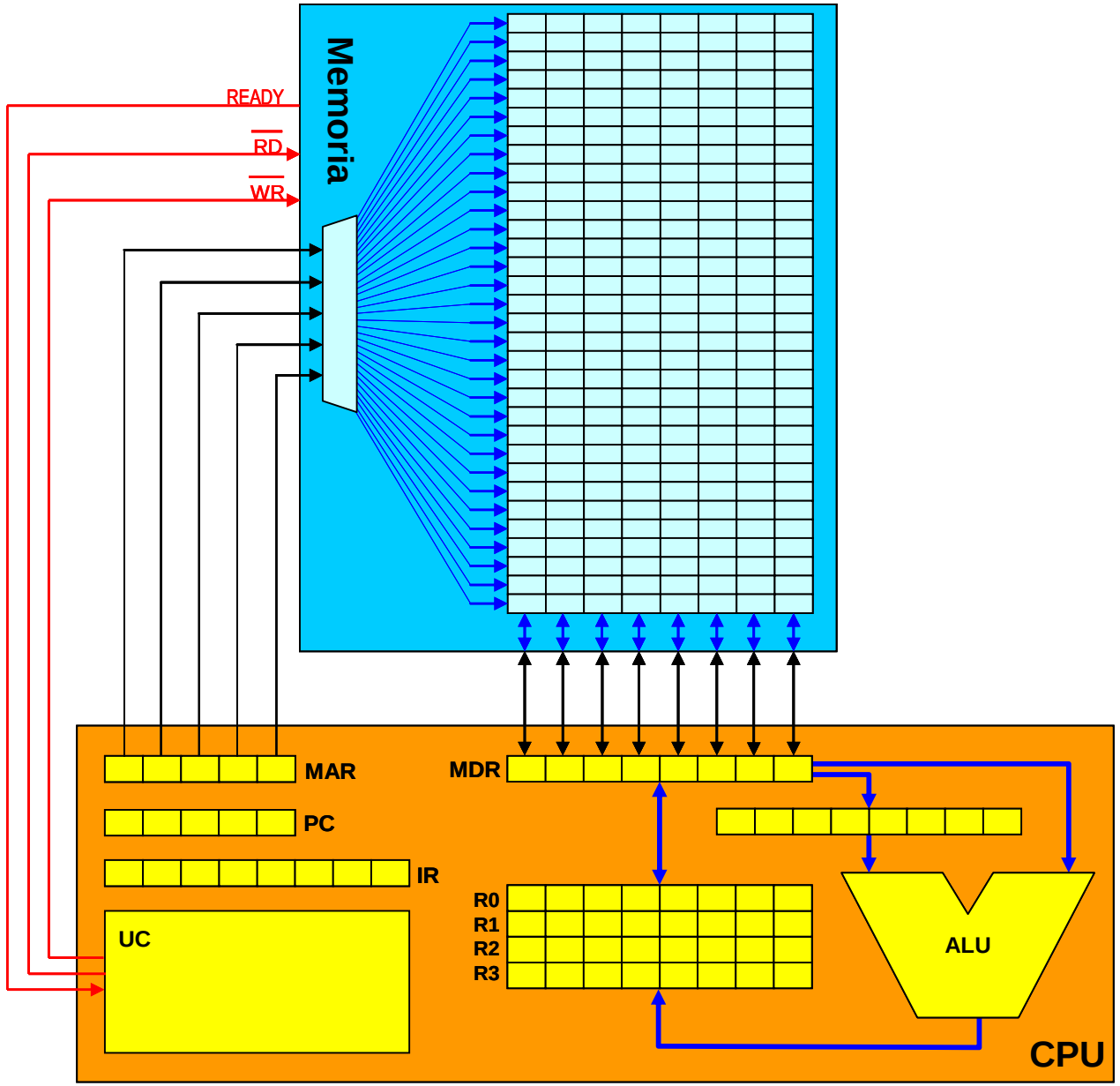
Ejemplo: ALU 74181

Selección				Funciones lógicas M=H	Operaciones aritméticas M=L	
S_3	S_2	S_1	S_0		C_0 = L (Sin acarreo)	C_{0-H} (Con acarreo)
L	L	L	L	/A	A menos 1	A
L	L	L	H	/(A.B)	AB menos 1	A.B
L	L	H	L	/A + B	A./B menos 1	A./B
L	L	H	H	1	Menos 1 (compl. A 2)	0 (cero)
L	H	L	L	/(A + B)	A más (A + /B)	A más (A + /B) más 1
L	H	L	H	/B	AB más (A + B)	A.B más (A+/B) más 1
L	H	H	L	/(A ⊕ B)	A menos B menos 1	A menos B
L	H	H	H	A + /B	A + /B	(A + /B) más 1
H	L	L	L	/A.B	A más (A + B)	A más (A + B) más 1
H	L	L	H	A ⊕ B	A más B	A más B más 1
H	L	H	L	B	A./B más (A + B)	A./B más (A + B) más 1
H	L	H	H	A + B	A + B	A + B más 1
H	H	L	L	0	A más A	A más A más 1
H	H	L	H	A./B	A.B más A	A.B más A más 1
H	H	H	L	A.B	A./B más A	A./B más A más 1
H	H	H	H	A	A	A más 1



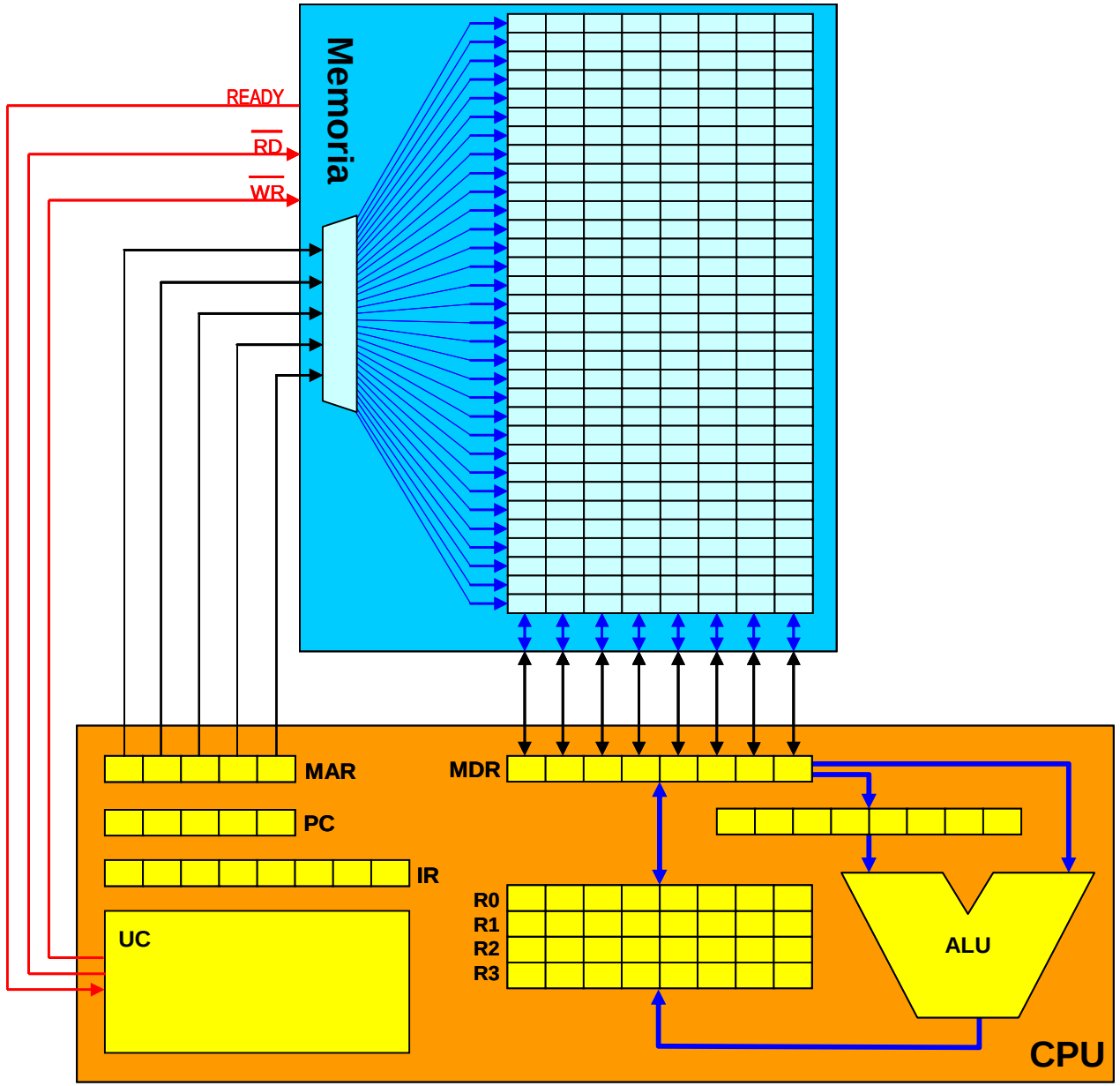
CPU
Unidad Central de Proceso

- Unidad aritmético-lógica
- Unidad de control
- Registros generales
  - Almacenamiento temporal de datos



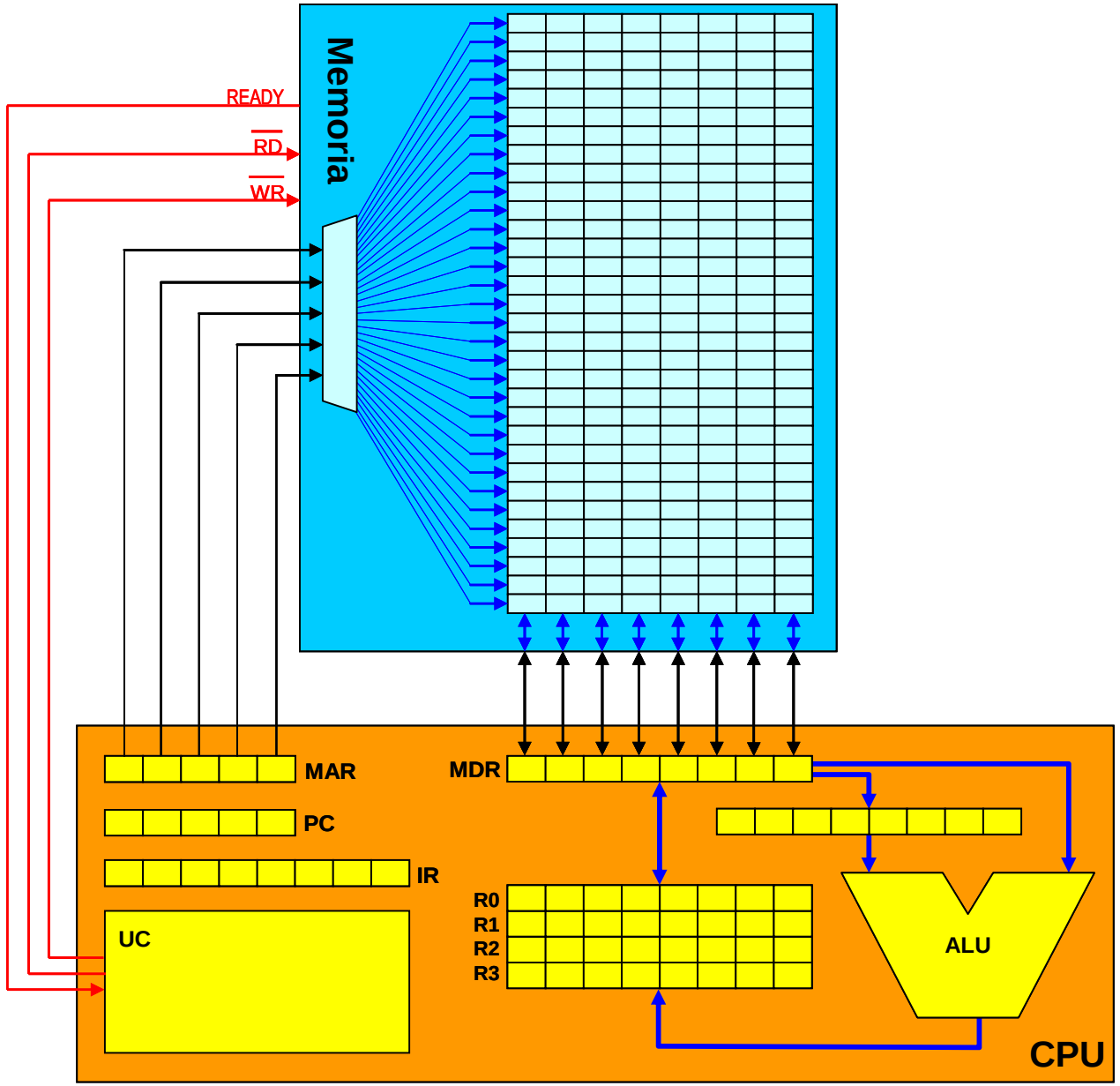
CPU
Unidad Central de Proceso

- Registros específicos
  - MAR (Memory Address Register, Registro de Direcciones de Memoria)
    - Dirección de memoria a la que se está accediendo
  - MDR (Memory Data Register, Registro de Datos de Memoria)
    - Datos leídos/escritos de/en memoria



CPU
Unidad Central de Proceso

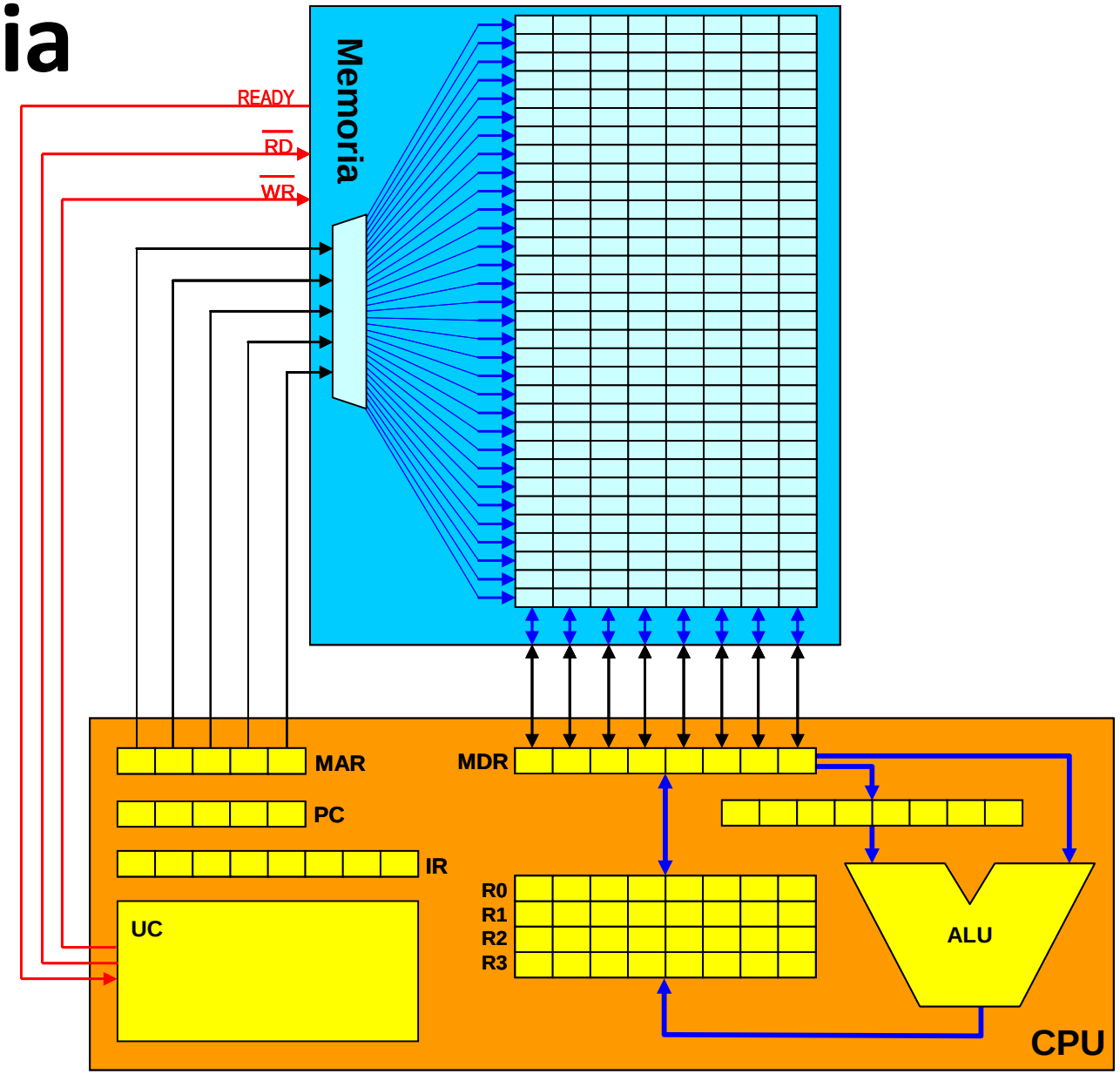
- Registros específicos
  - IR (Instruction Register, Registro de Instrucción)
    - Instrucción que se está ejecutando actualmente
  - PC (Program Counter, Contador de Programa)
    - Dirección de la siguiente instrucción que debe ejecutarse



Conexión CPU - memoria

Bus de direcciones

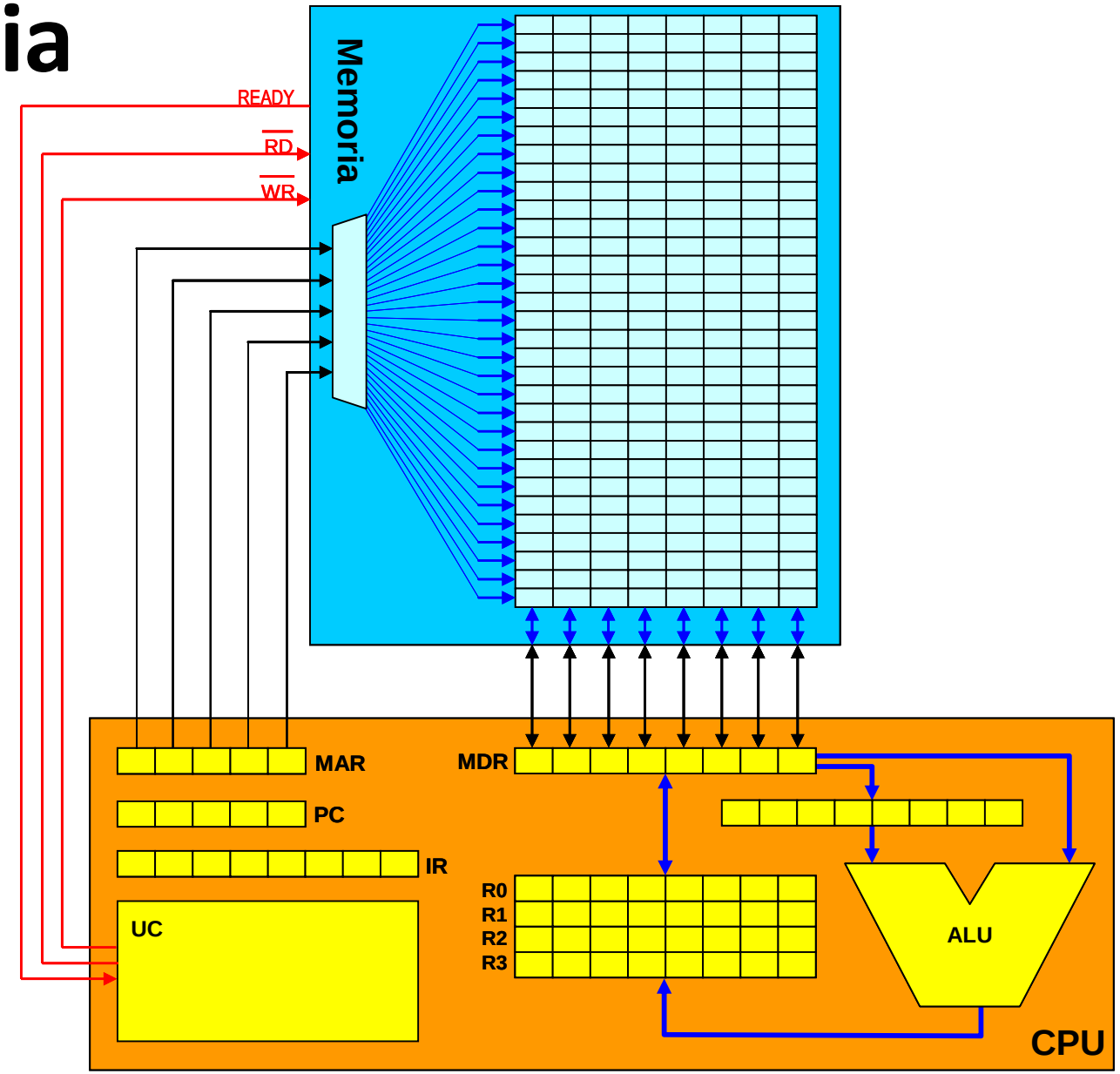
- Transporta la dirección de la posición de memoria (o puerto de periférico) que interviene en el tráfico de información
- Suele ser unidireccional
- Se conecta al registro MAR de la CPU



Conexión CPU - memoria

Bus de datos

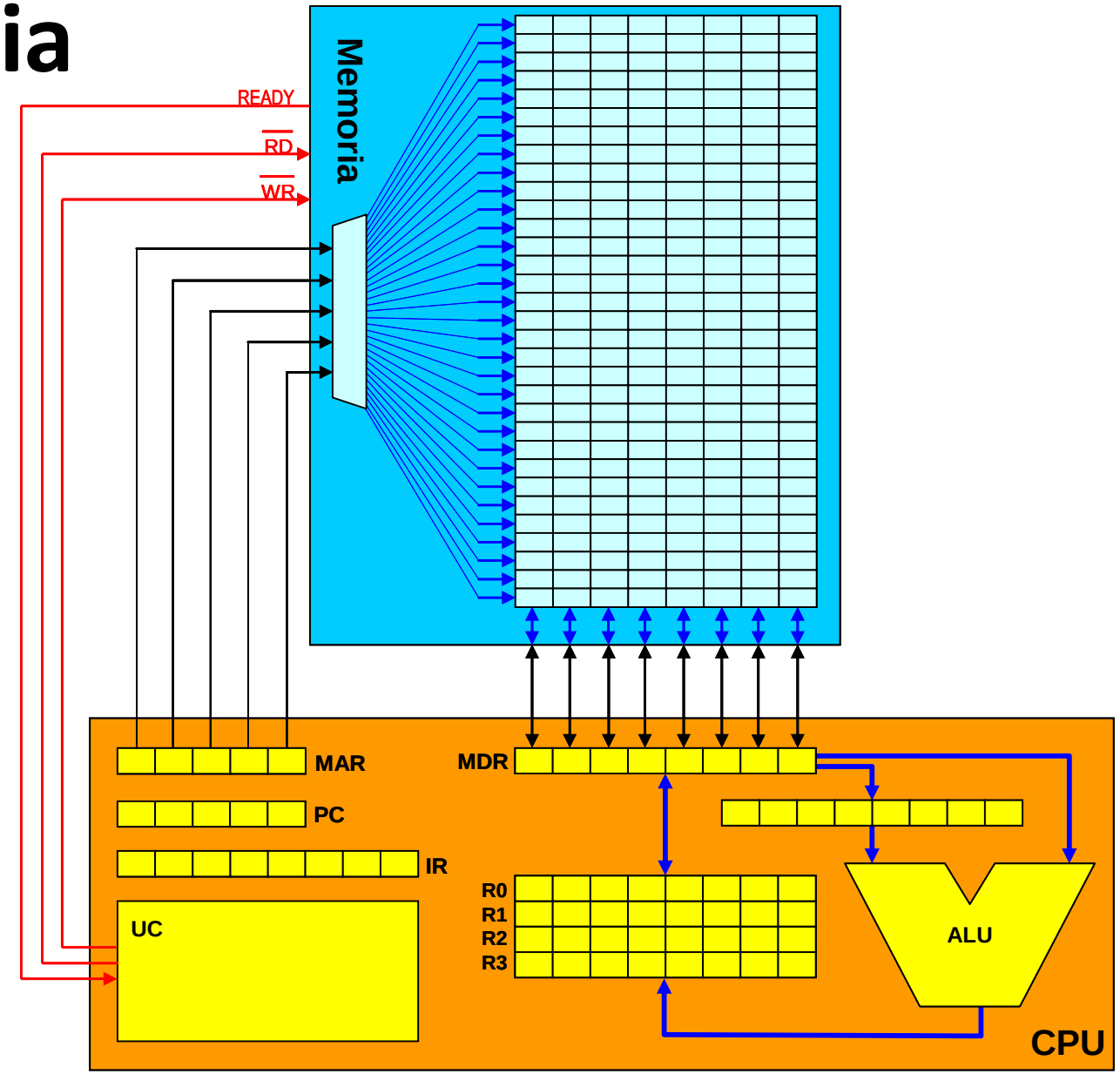
- Transporta los datos que se transfieren entre las unidades
- Suele ser bidireccional
- Se conecta al registro MDR de la CPU



Conexión CPU - memoria

Bus de control

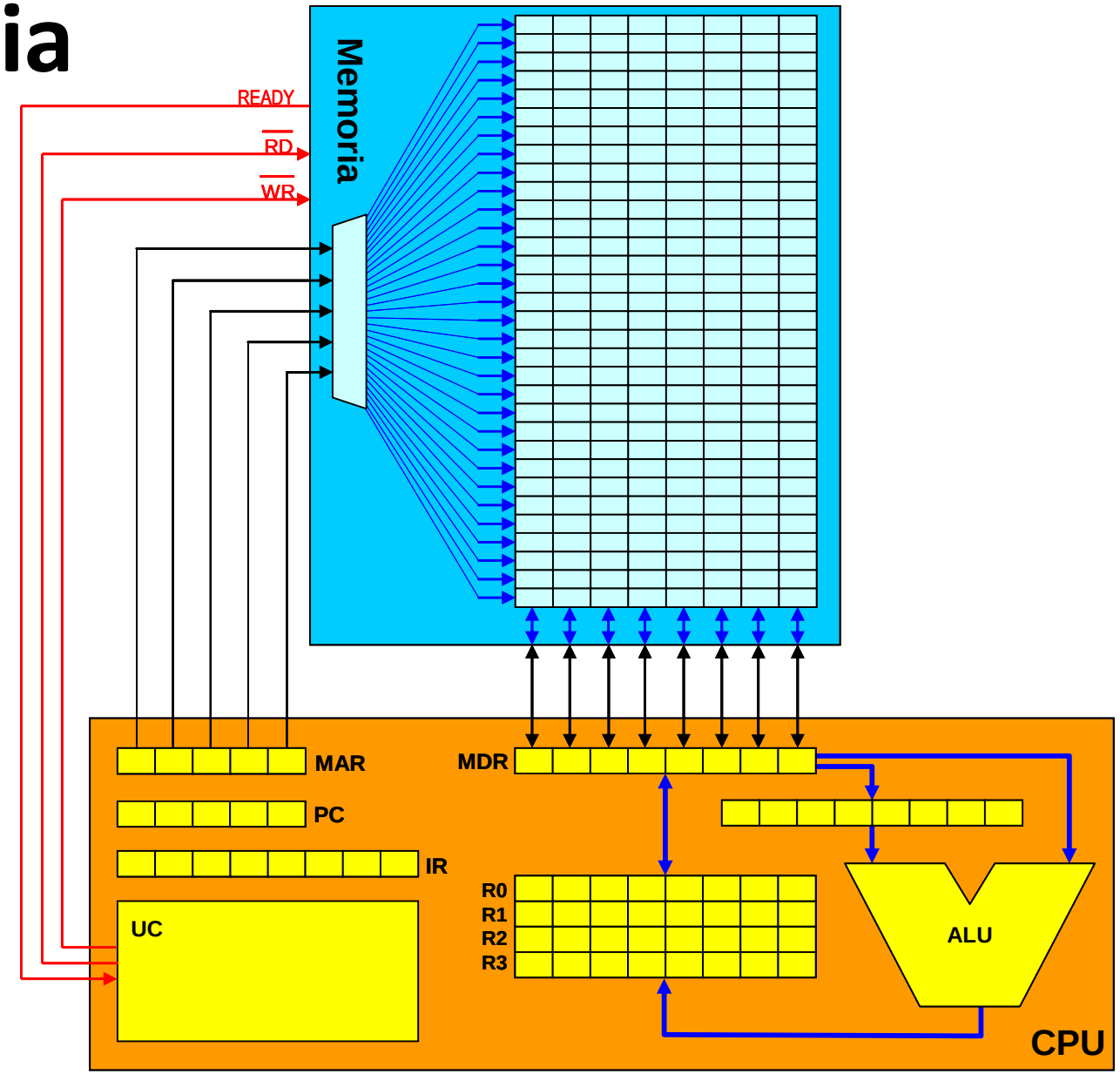
- Gestiona la comunicación indicando
  - La dirección de la transferencia de datos
  - Temporización de la transmisión
  - Posibles interrupciones
- Señales de control
  - parten de la CPU
- Señales de estado
  - llegan a la CPU



Conexión CPU - memoria
Operación de lectura

Memoria (espera señal)
4) Recoge la dirección de memoria del bus de direcciones
5) Decodifica la dirección
6) Copia palabra decodificada al bus de datos
7) Activa señal READY
8) Espera desactivación de RD
11) Desactiva READY

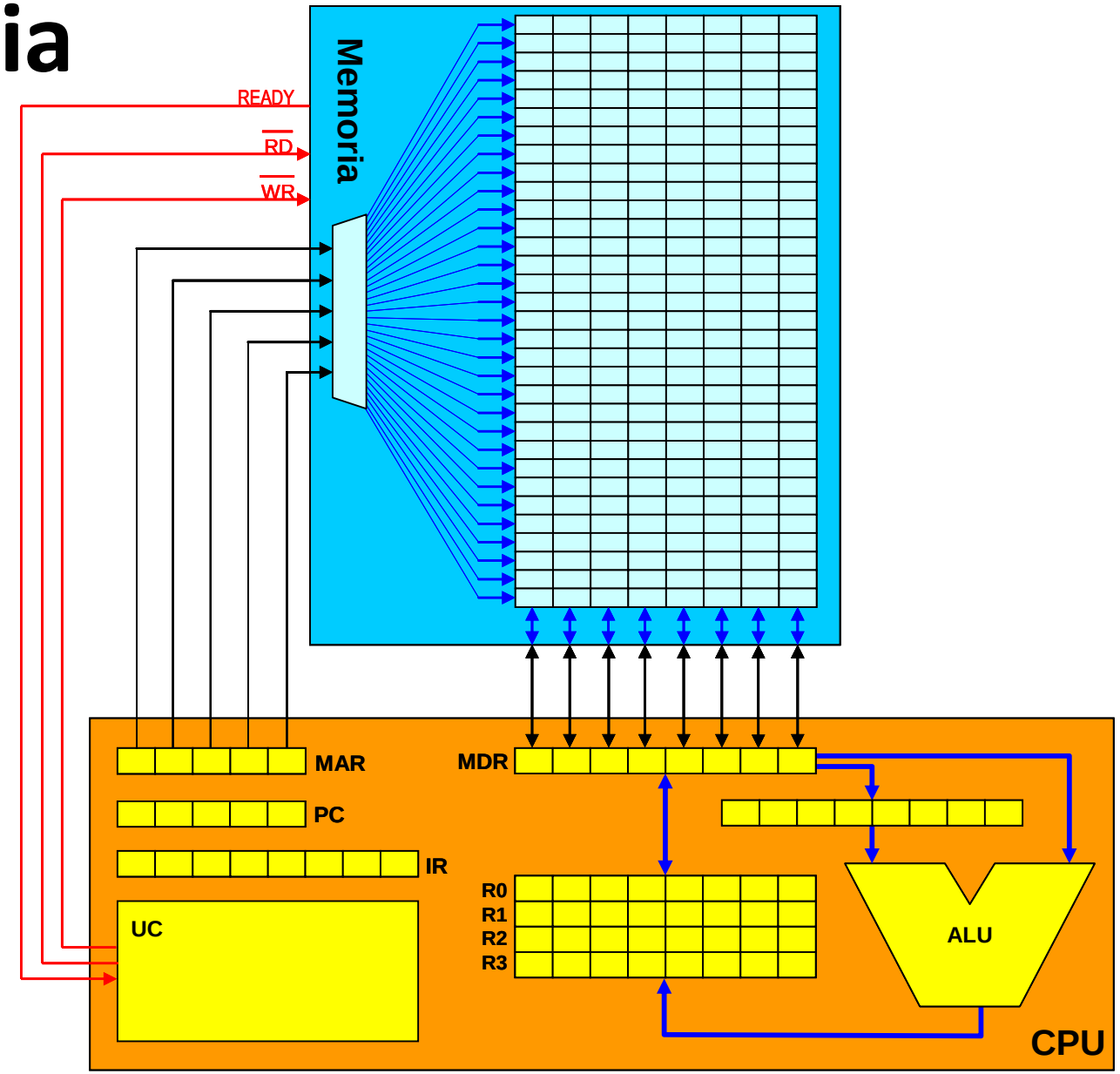
CPU
1) Pone la dirección de memoria a leer en el MAR
2) Activa la señal RD
3) Espera la señal READY
9) Recoge el dato del MDR
10) Desactiva la señal RD



Conexión CPU - memoria
Operación de escritura

Memoria (espera señal)
5) Recoge la dirección de memoria del bus de direcciones
6) Decodifica la dirección
7) Copia el contenido del bus de datos en la palabra decodificada
8) Activa señal READY
9) Espera desactivación de WR
11) Desactiva READY

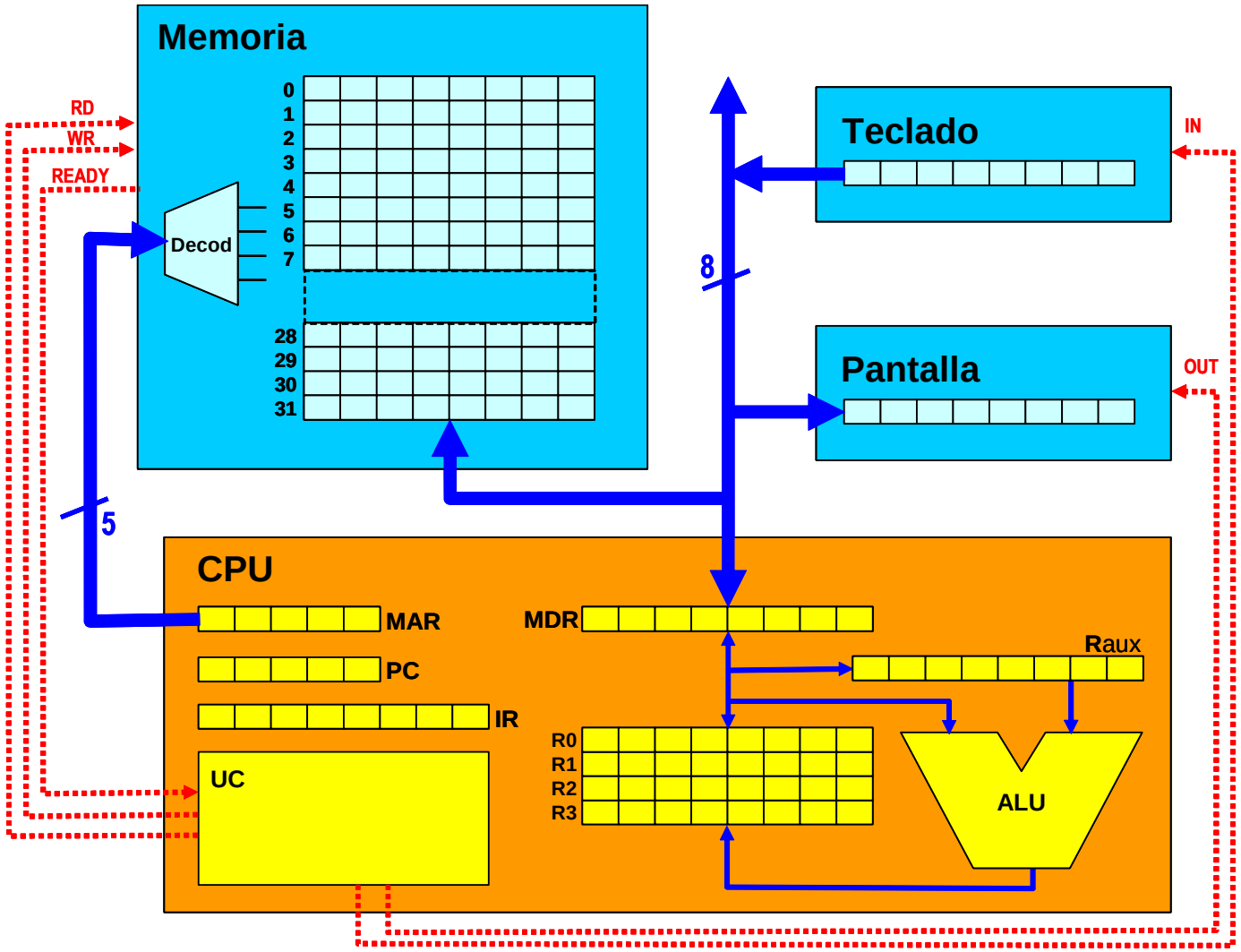
CPU
1) Pone la dirección de memoria a escribir en el MAR
2) Pone dato en MDR
3) Activa la señal WR
4) Espera la señal READY
10) Desactiva la señal WR



Ejecución de instrucciones

Ejemplo de computador

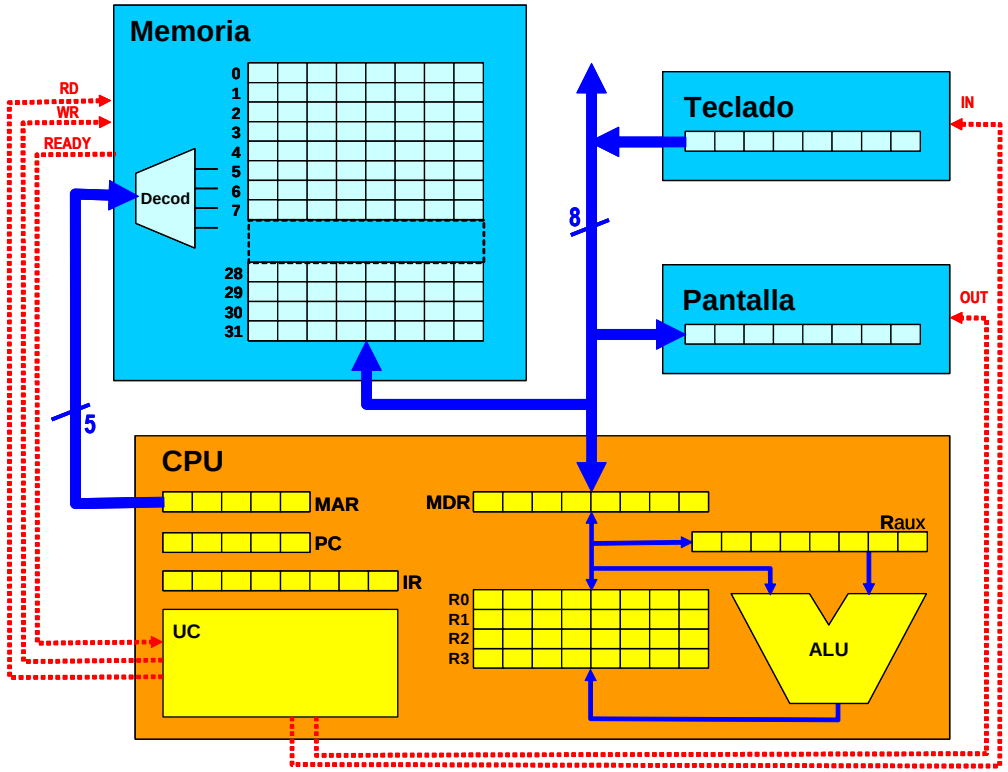
- Memoria de 32 palabras de 8 bits
- Entrada por teclado
- Salida por pantalla



Ejecución de instrucciones
Lenguaje máquina

- Instrucciones de 8 bits

Código nemotécnico	Instrucción máquina	Descripción
ENT M(m)	000mmmmm	memoria ← teclado
SAL M(m)	001mmmmm	pantalla ← memoria
CAR R0,M(m)	010mmmmm	Registro ← memoria
ALM M(m),R0	011mmmmm	memoria ← Registro
MOV Rx,Ry	100-xxyy	Rx ← Ry
SUM Rx,Ry	101-xxyy	Rx ← Rx+Ry

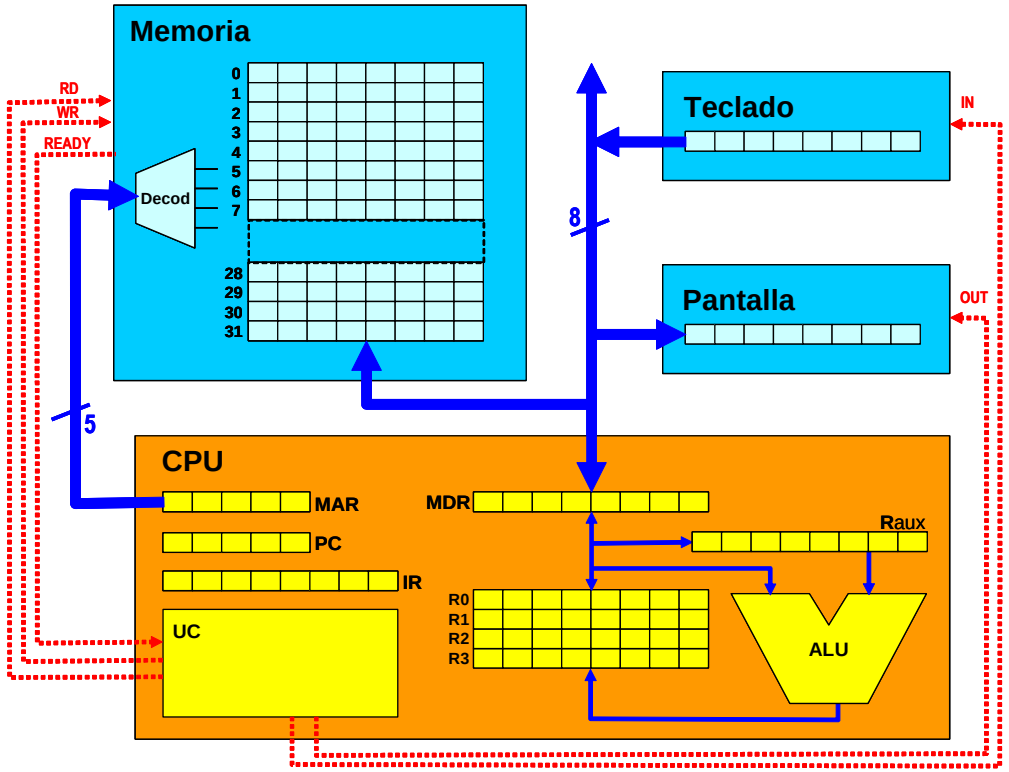


Ejecución de instrucciones

- Escribir un programa que sume 2 números tomados del teclado y saque el resultado por pantalla

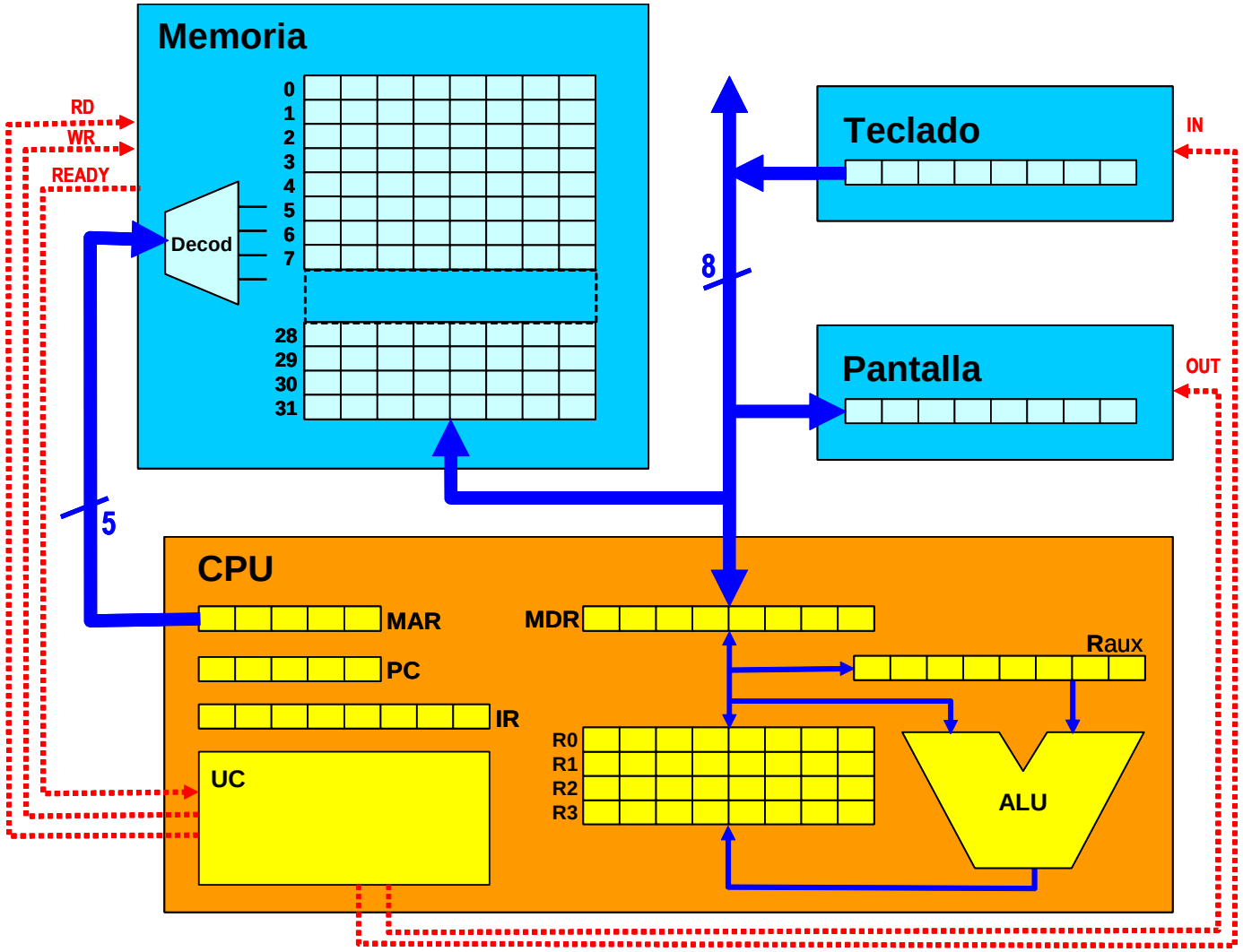
```
ENT M(29)          00011101
ENT M(30)          00011110
CAR R0,M(29)       01011101
MOV R1,R0          10000100
CAR R0,M(30)       01011110
SUM R0,R1          10100001
ALM M(31),R0       01111111
SAL M(31)          00111111
```

Código nemotécnico	Instrucción máquina	Descripción
ENT M(m)	000mmmmm	memoria ← teclado
SAL M(m)	001mmmmm	pantalla ← memoria
CAR R0,M(m)	010mmmmm	Registro ← memoria
ALM M(m),R0	011mmmmm	memoria ← Registro
MOV Rx,Ry	100-xxyy	Rx ← Ry
SUM Rx,Ry	101-xxyy	Rx ← Rx+Ry



Ejecución de instrucciones
Fases

- Fase 1: captación o búsqueda de instrucción
  - Se traslada una instrucción desde la memoria hasta el registro IR de la CPU
  - MAR ← PC, RD
  - PC ← PC+1
  - MDR ← M(MAR)
  - IR ← MDR



Ejecución de instrucciones
Fases

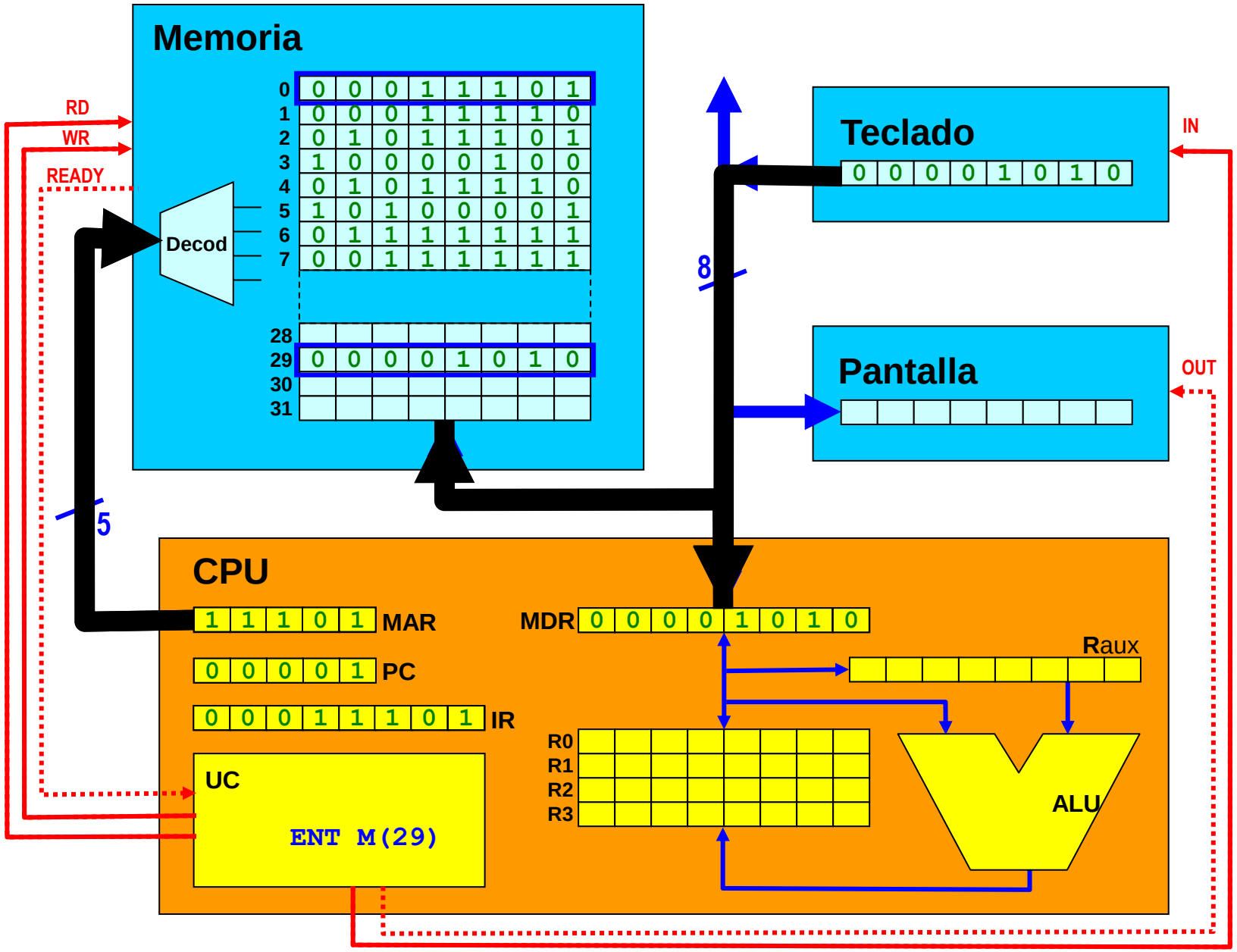
- Fase 2: decodificación de instrucción
  - La unidad de control interpreta la instrucción que se encuentra en el registro IR

- Fase 3: ejecución
  - La UC realiza las operaciones que implica la instrucción
    - Genera señales de control necesarias
  - En caso necesario, lee los operandos requeridos desde memoria
    - [Rn ← OPERANDO]

Ejecución de instrucciones
Carga de programas

- Para ejecutar un programa, primero éste debe introducirse en la memoria de la computadora
  - De esto se encarga un programa del sistema operativo denominado cargador
- Una vez cargado el programa, el SO indica al computador que pase el control a la primera instrucción del programa cargado
  - Inicializa el PC con la dirección de la primera instrucción del programa

Ejecución: 1ª instrucción



Ejecución: 3ª instrucción