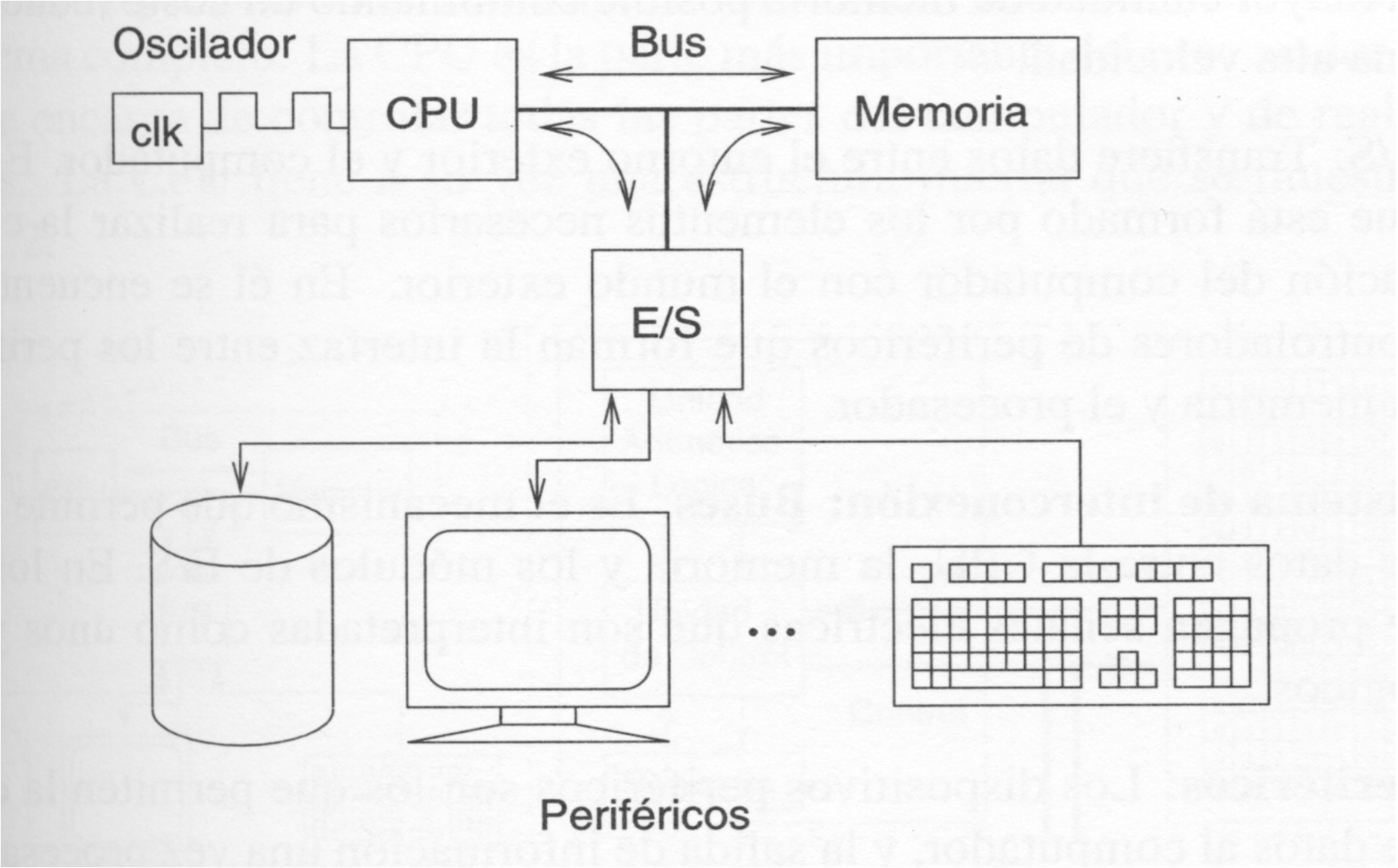
Contenidos

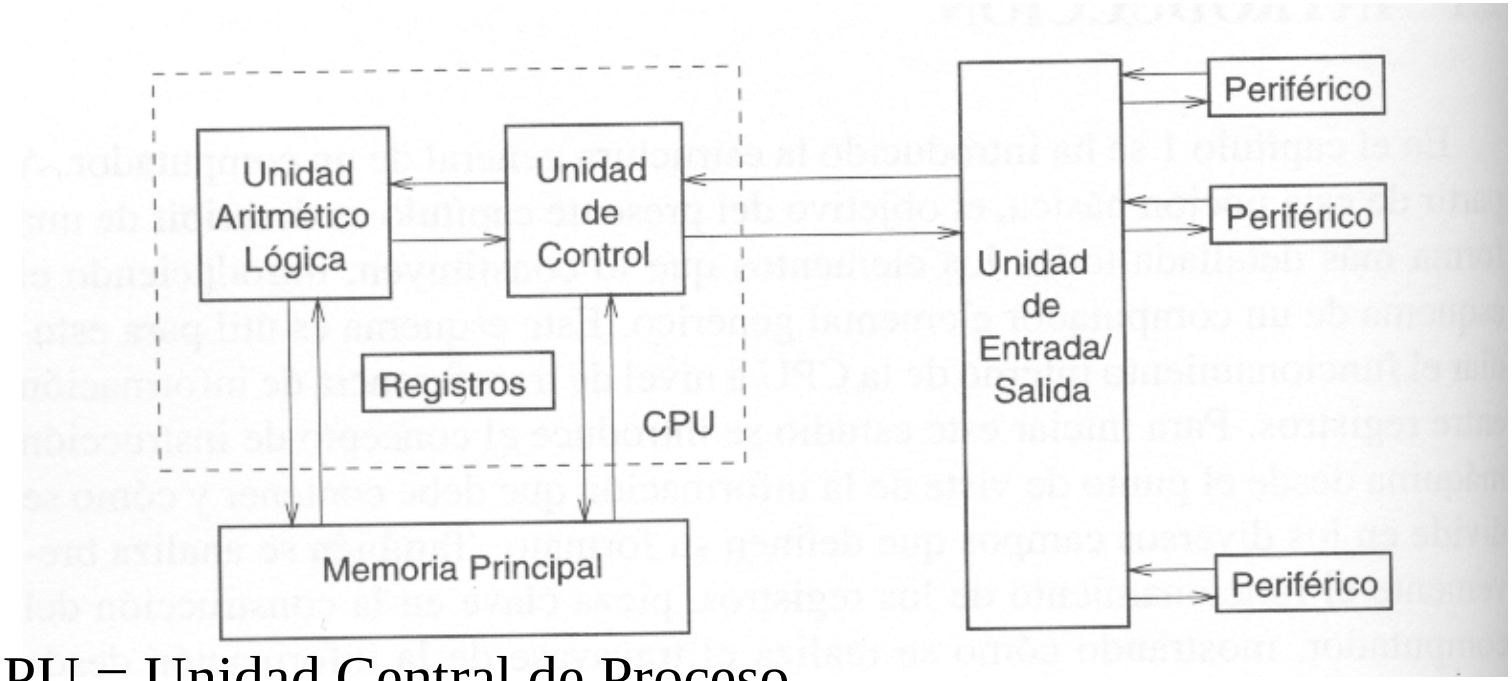
- Arquitectura de ordenadores (fundamentos teóricos)
  - Representación de la información
  - Estructura de un microprocesador
  - Memorias
  - Sistemas de E/S
- Elementos de un ordenador
  - Microprocesador
  - Placa base
  - Chipset
  - Memoria
  - Conexión de periféricos
  - BIOS
- Periféricos
  - Conexión al ordenador
  - Periféricos de entrada
  - Periféricos de salida
  - Periféricos de entrada/salida

Bibliografía

- “Arquitectura de computadores” J.M. Angulo y P. de Miguel. Paraninfo 1991
- “Estructura de computadores y periféricos” R.J. Martínez, J.A. Boluda y J.J. Pérez. Ra-Ma 2001
- “Arquitectura de ordenadores” M.A. De Miguel y T. Higuera. Ra-Ma 1996

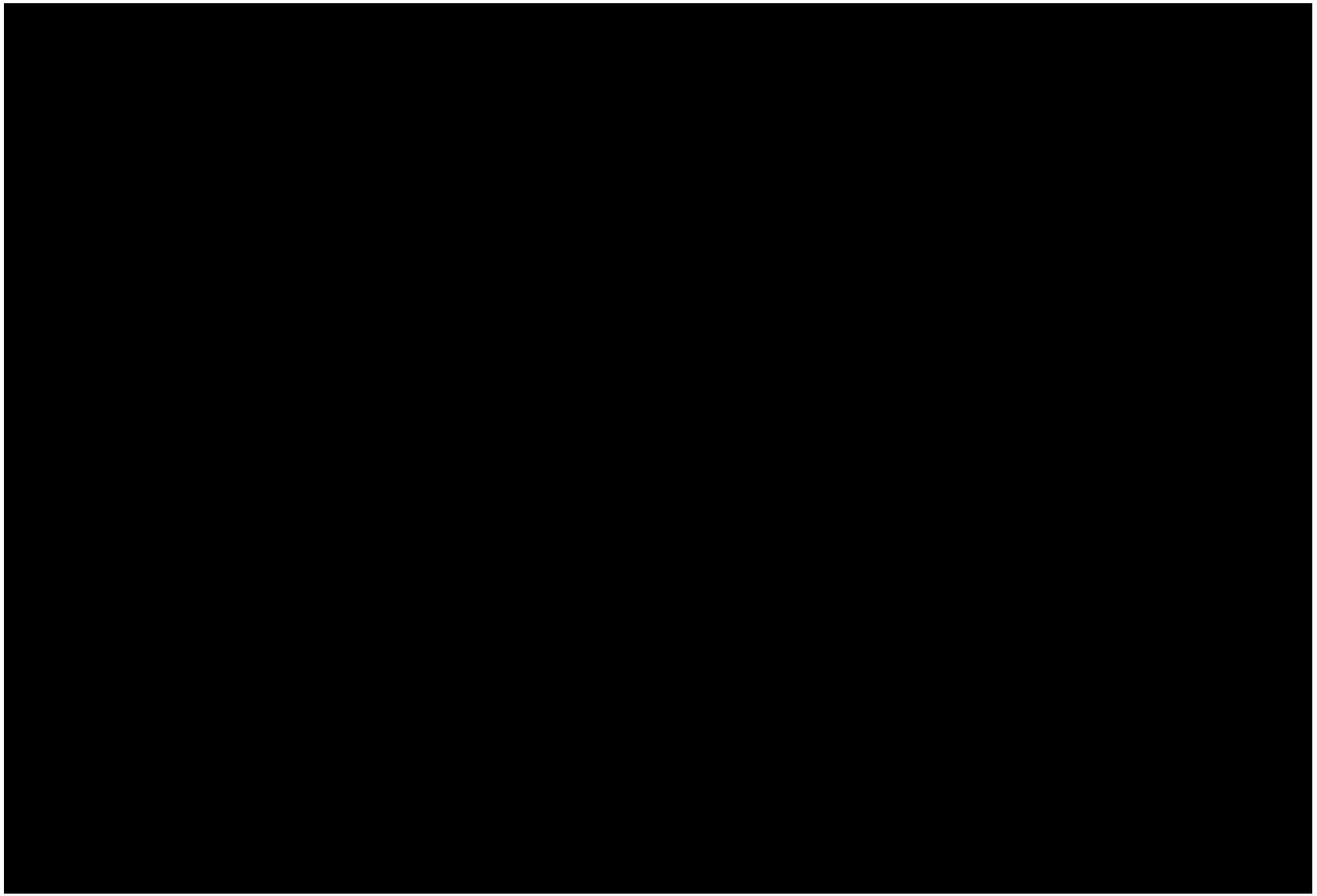
Estructura general de un ordenador (Von Neumann)





- CPU = Unidad Central de Proceso
  - Controla el flujo de datos, procesa los datos, lee y decodifica instrucciones, realiza cálculos aritméticos y lógicos, almacena datos temporales
- Memoria (RAM)
  - Almacena datos
- E/S
  - Transferencia de datos entre el ordenador y el exterior
- Buses
  - Interconexión entre todos los dispositivos. Transporte de datos

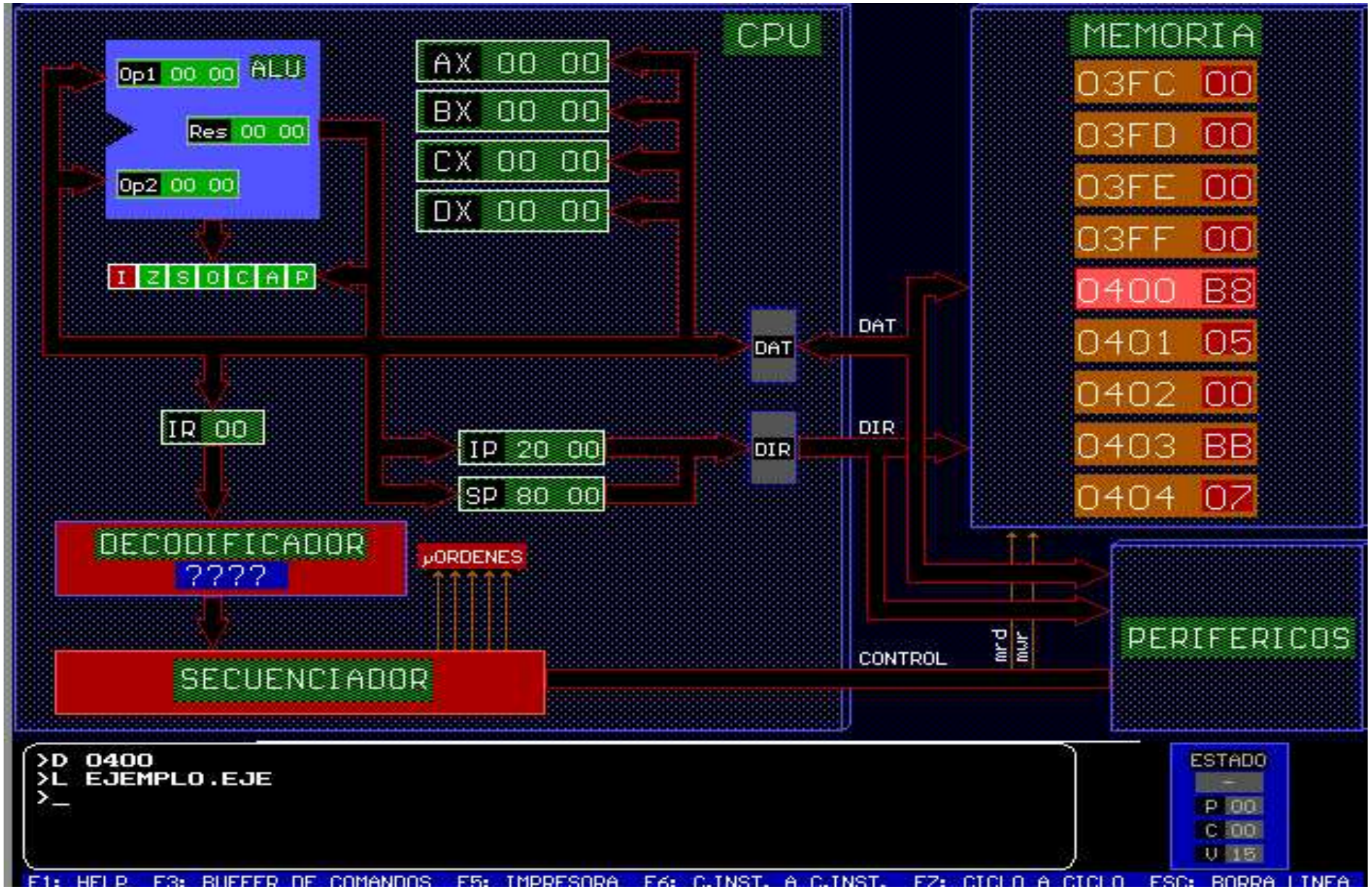
Buses



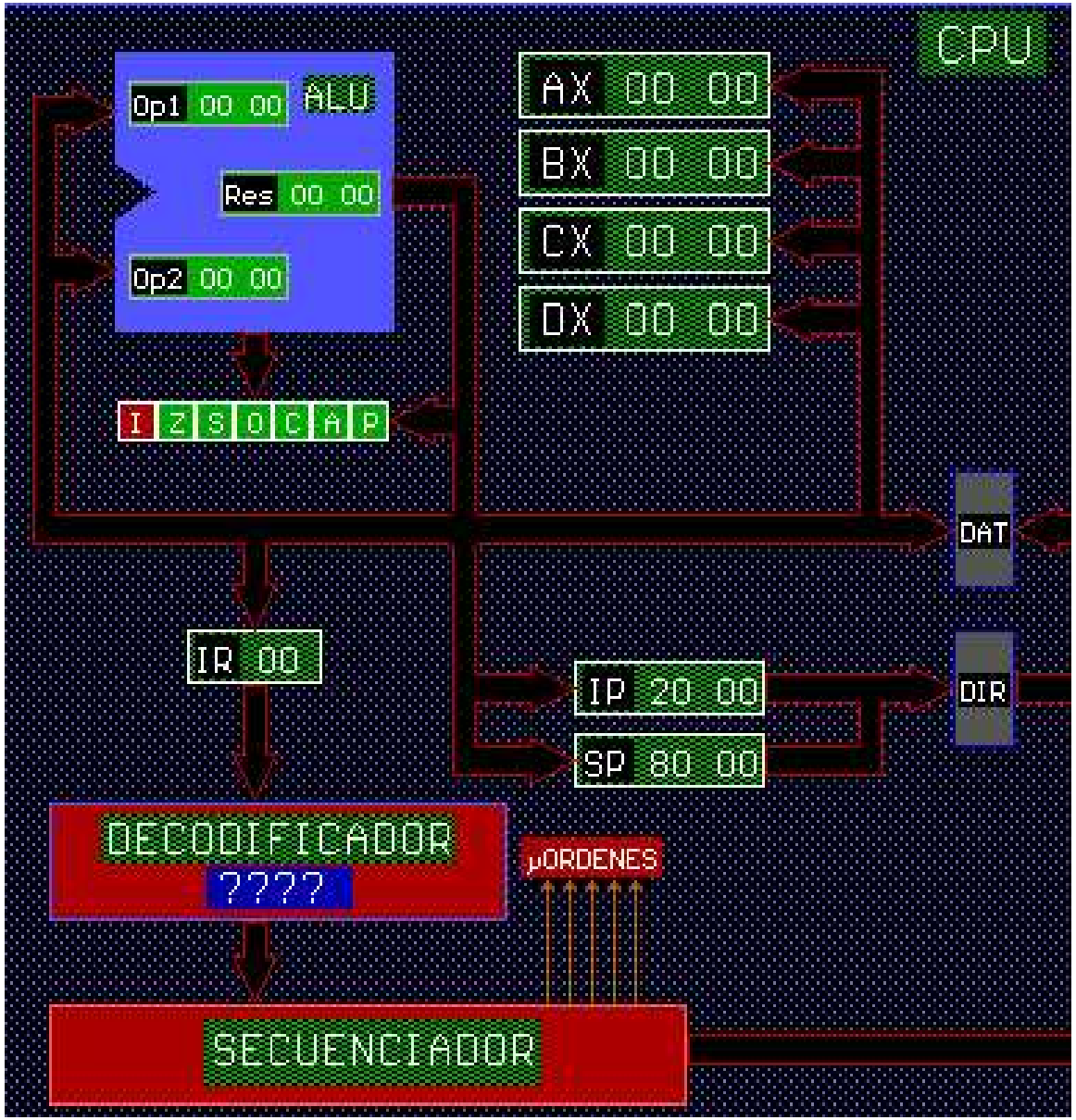
Ejecución de instrucciones

- Memoria -> datos + programa (instrucciones)
- Ejecución secuencial
- Ejecución de una instrucción
  - Ciclo de lectura del código de operación (fetch) -> se lee de la memoria el código de la instrucción y se almacena en el registro de instrucción (IR)
  - Decodificación de la instrucción
  - Lectura de los operandos (si hay) -> se buscan los operandos de la instrucción en memoria o de registros y se leen (se almacenan en la CPU)
  - Ejecución de la instrucción
  - Almacenamiento del resultado en memoria o registros
- La ejecución secuencial se puede romper por instrucciones de salto, interrupciones o excepciones

Ejemplo de ordenador (MSX88)



Ver: http://msx88.diatel.upm.es/

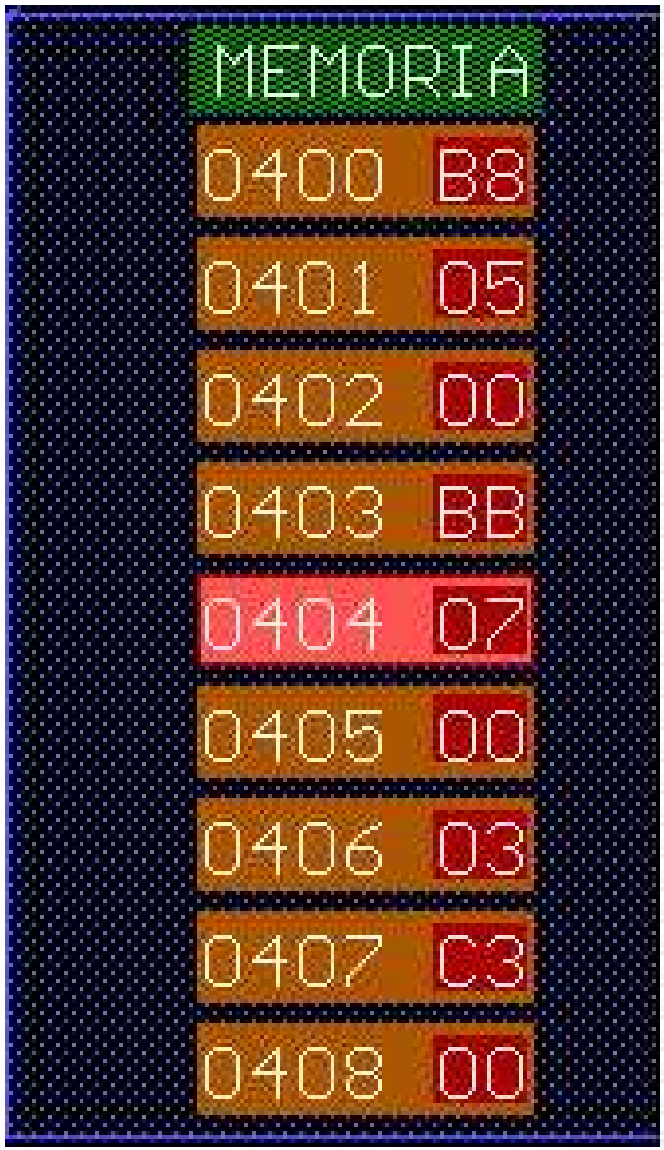


ELEMENTOS

- ALU: operaciones matemáticas y lógicas
- Registros (16 bits): almacenamiento de datos
- IR: registro de instrucción
- IP: contador de programa
- SP: puntero de pila
- Flags: señalizadores de estado
- Buses: datos, direcciones y control

La ALU

Programa almacenado en memoria

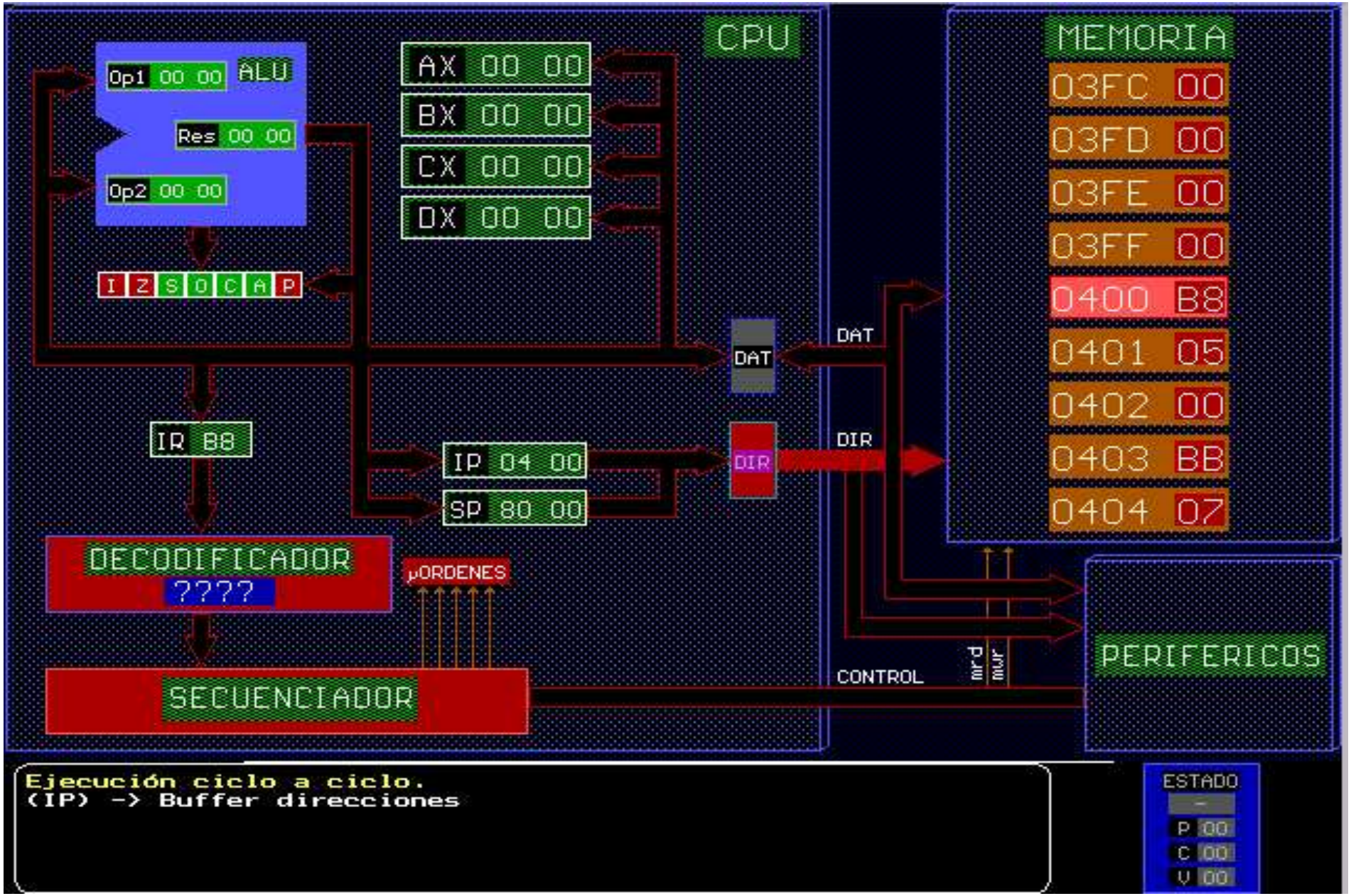


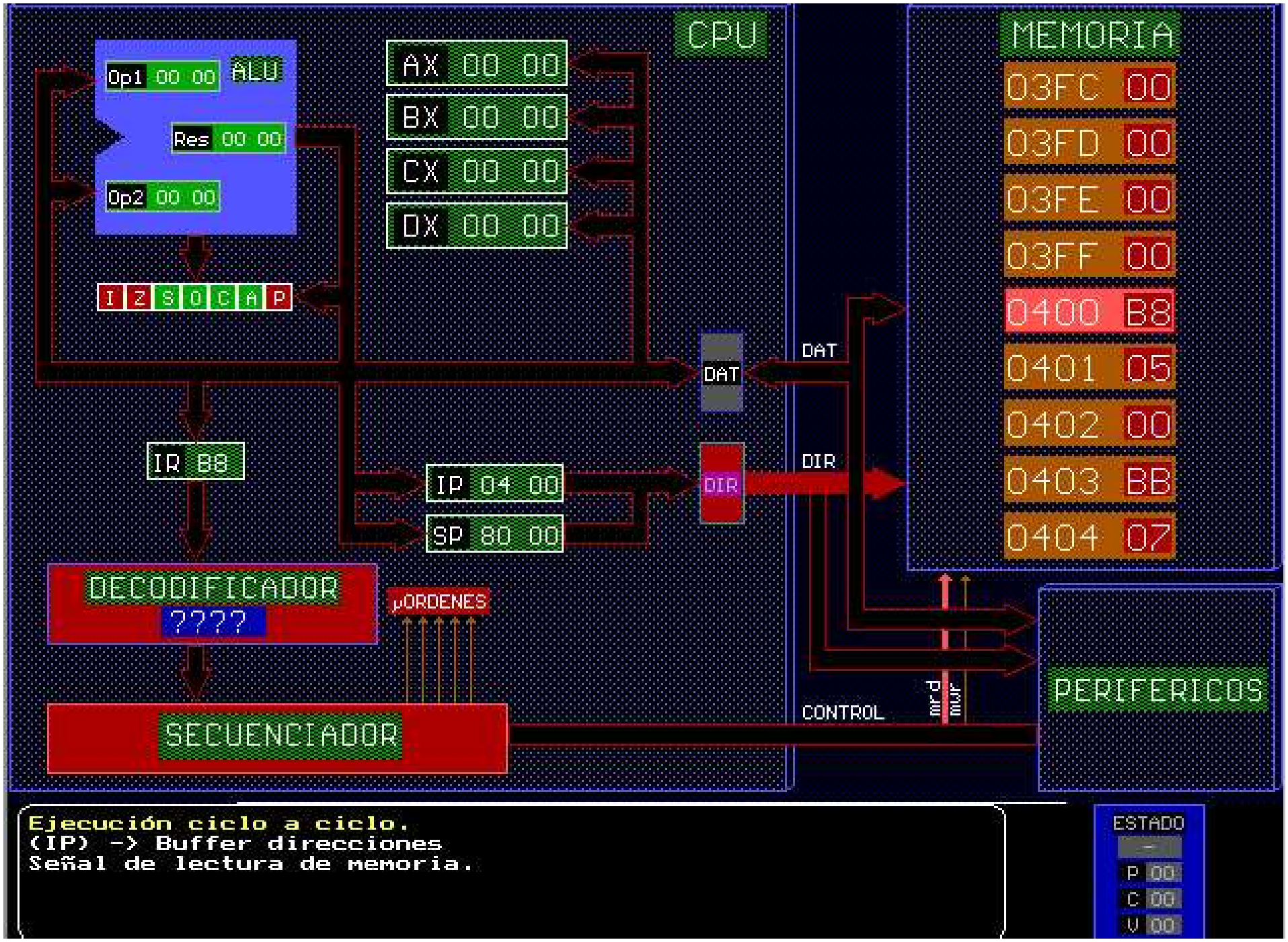
} MOV AX, 05

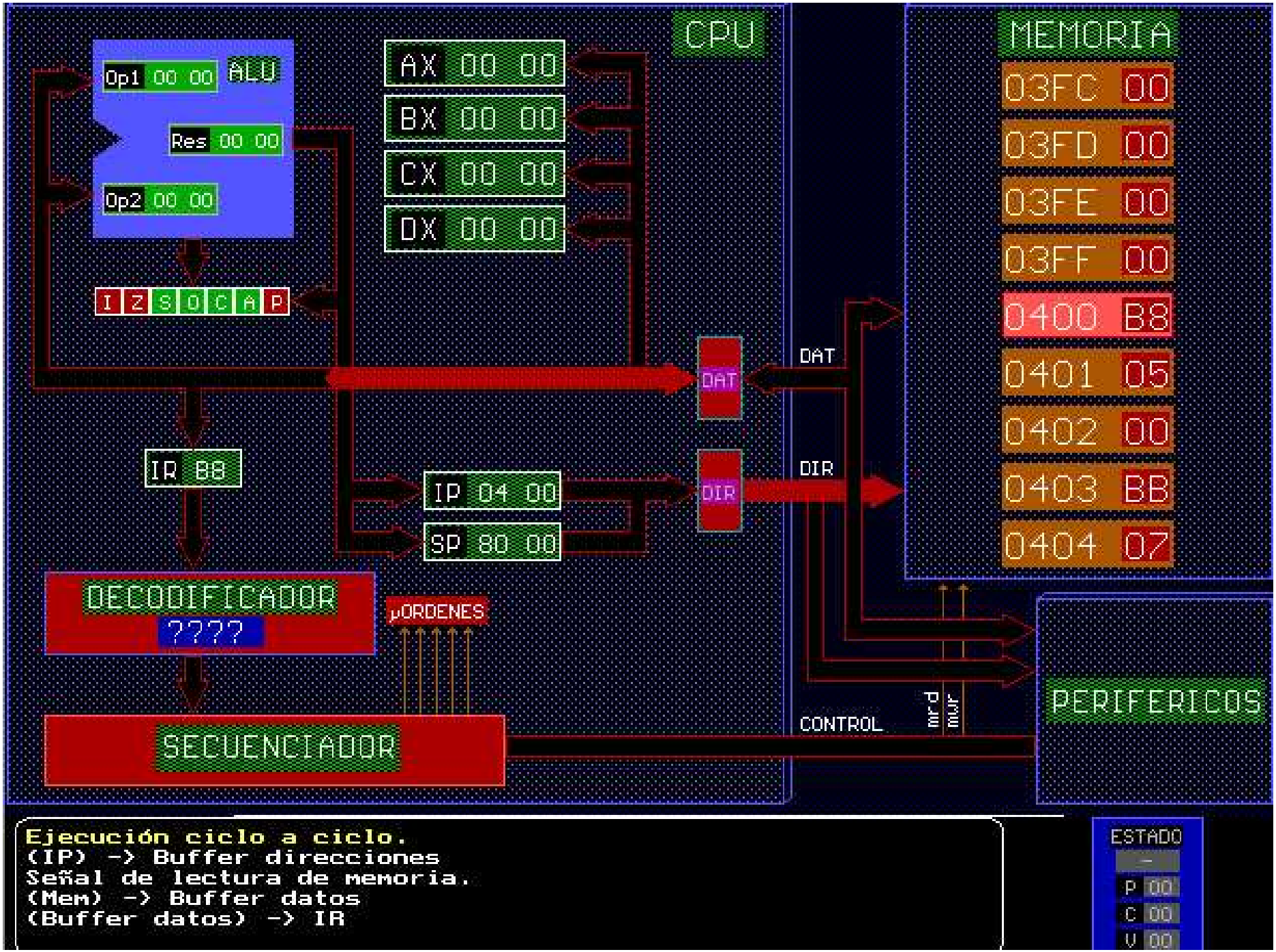
} MOV BX, 07

} ADD AX, BX

Flujo de ejecución de una instrucción




CPU
MEMORIA
ALU
Op1 00 00
Res 00 00
Op2 00 00
I Z S O C A P
AX 00 00
BX 00 00
CX 00 00
DX 00 00
03FC 00
03FD 00
03FE 00
03FF 00
0400 B8
0401 05
0402 00
0403 BB
0404 07
DAT
DAT
DIR
DIR
IR B8
IP 04 00
SP 80 00
DECODIFICADOR
????
µORDENES
SECUENCIADOR
CONTROL
mrd
mwr
PERIFERICOS
Ejecución ciclo a ciclo.
(IP) -> Buffer direcciones
Señal de lectura de memoria.
ESTADO
P 00
C 00
V 00


CPU
ALU
Op1 00 00
Res 00 00
Op2 00 00
I Z S O C A P
AX 00 00
BX 00 00
CX 00 00
DX 00 00
IR B8
IP 04 00
SP 80 00
DAT
DIR
MEMORIA
03FC 00
03FD 00
03FE 00
03FF 00
0400 B8
0401 05
0402 00
0403 BB
0404 07
DECODIFICADOR
????
µORDENES
SECUENCIADOR
CONTROL
mr
mw
PERIFERICOS
Ejecución ciclo a ciclo.
(IP) -> Buffer direcciones
Señal de lectura de memoria.
(Mem) -> Buffer datos
(Buffer datos) -> IR
ESTADO
P 00
C 00
V 00

EJEMPLO.ASM. ACCIONES REALIZADAS POR LA CPU EN LA EJECUCIÓN (flujo completo)

- Búsqueda del código de la 1ª operación
  - IP -> bus de direcciones
  - orden de lectura
  - (0400) -> bus de datos
  - (Bus de datos) -> IR
  - Decodificación
  - (IP) + 1 -> IP
- Carga del registro AX
  - IP -> bus de direcciones
  - orden de lectura
  - (0401) -> bus de datos
  - (bus de datos) -> AL
  - (IP) + 1 -> IP
  - IP -> bus de direcciones
  - orden de lectura
  - (0402) -> bus de datos
  - (bus de datos) -> AH
  - (IP) + 1 -> IP
- Búsqueda del código de la 2ª operación
  - IP -> bus de direcciones
  - orden de lectura
  - (0403) -> bus de datos
  - (Bus de datos) -> IR
  - Decodificación
  - (IP) + 1 -> IP

- Carga del registro BX
  - IP -> bus de direcciones
  - orden de lectura
  - (0404) -> bus de datos
  - (bus de datos) -> BL
  - (IP) + 1 -> IP
  - IP -> bus de direcciones
  - orden de lectura
  - (0405) -> bus de datos
  - (bus de datos) -> BH
  - (IP) + 1 -> IP
- Búsqueda del 1er byte de código de la 3ª operación
  - IP -> bus de direcciones
  - orden de lectura
  - (0406) -> bus de datos
  - (Bus de datos) -> IR
  - Decodificación
  - (IP) + 1 -> IP
- Búsqueda del 2º byte de código de la 3ª operación
  - IP -> bus de direcciones
  - orden de lectura
  - (0407) -> bus de datos
  - (Bus de datos) -> IR
  - Decodificación
  - (IP) + 1 -> IP

- Suma de AX y BX
  - (AX) -> b.datos interno -> ALU
  - (BX) -> b.datos interno -> ALU
  - Suma de datos -> b. datos interno -> AX
  - actualización de flags
  - (IP) + 1 -> IP

Tipos de instrucciones

- Transferencia de datos
- Aritméticas
- Lógicas
- Transferencia de control del programa
- Interrupción

- Instruciones de tres operandos -> se realiza operaciones con dos y el resultado se almacena en el tercero
- Instrucciones de dos operandos -> el resultado se almacena en uno de los dos operandos
- Instruciones con un operando -> el acumulador se utiliza como 2º operando y también para almacenar el resultado

Tipos de memoria de semiconductores

- De lectura y escritura (RAM)
  - Estáticas (SRAM)
  - Dinámicas o con refresco (DRAM)
- De solo lectura
  - ROM (Read Only Memory)
  - PROM (Programable ROM)
  - EPROM (Erasable PROM)
  - EEPROM (Electricaly Erasable PROM)
  - FLASH -> grabable electricamente, por comandos

Elementos típicos

- Bus de datos
- Bus de direcciones
- Señales de control (activas todas a nivel bajo)
  - OE -> activa la salida triestado de la memoria
  - CS o CE -> activa el chip
  - WE -> señal de escritura
  - RAS o CAS -> señales de refresco de las RAM dinámicas

Tecnologías de memoria RAM

Tecnología	Datos 1993		Datos 2001		Datos 2003	
	Tiempo acceso	Precio por MB (cents)	Tiempo de acceso	Precio por MB (cents)	Tiempo de acceso	Precio por MB (cents)
SRAM	8-35ns	60.000-24.000				
DRAM	90-120ns	1.500-3.000	Hasta 4ns	12 – 18	Hasta 2,5ns	9 – 10
Disco duro	10-20ms	60-120	4-10ms	0,24-0,48	~	0,09 – 0,1

Jerarquía de memorias

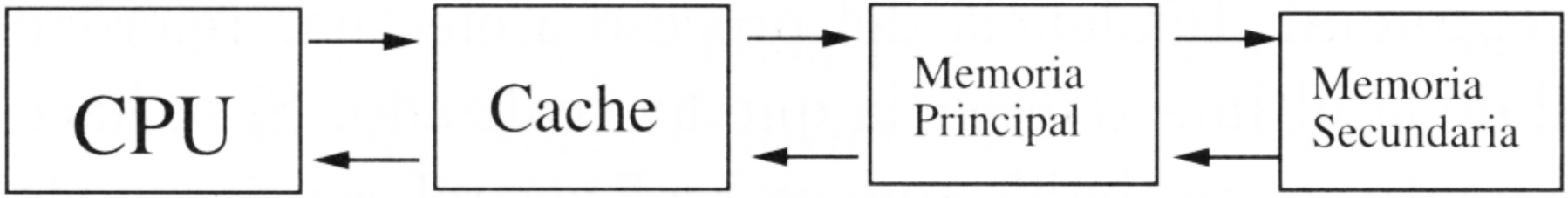
- Memoria cache (SRAM): hasta 1Mb
- Memoria física (DRAM): hasta 1Gb
- Memoria virtual (Disco): hasta 120Gb

Jerarquía de memorias

- Limite de velocidad de memoria:
  - Tecnología
  - Distancia “física” que deben recorrer los datos
- Distintos niveles de almacenamiento:
  - Memoria (cache) interna al micro -> SRAM muy rápida y escasa (y cara)
  - Memoria cache “externa” -> SRAM muy rápida y cara
  - Memoria “física” -> DRAM rápida y barata
  - Disco duro -> lento y muy barato

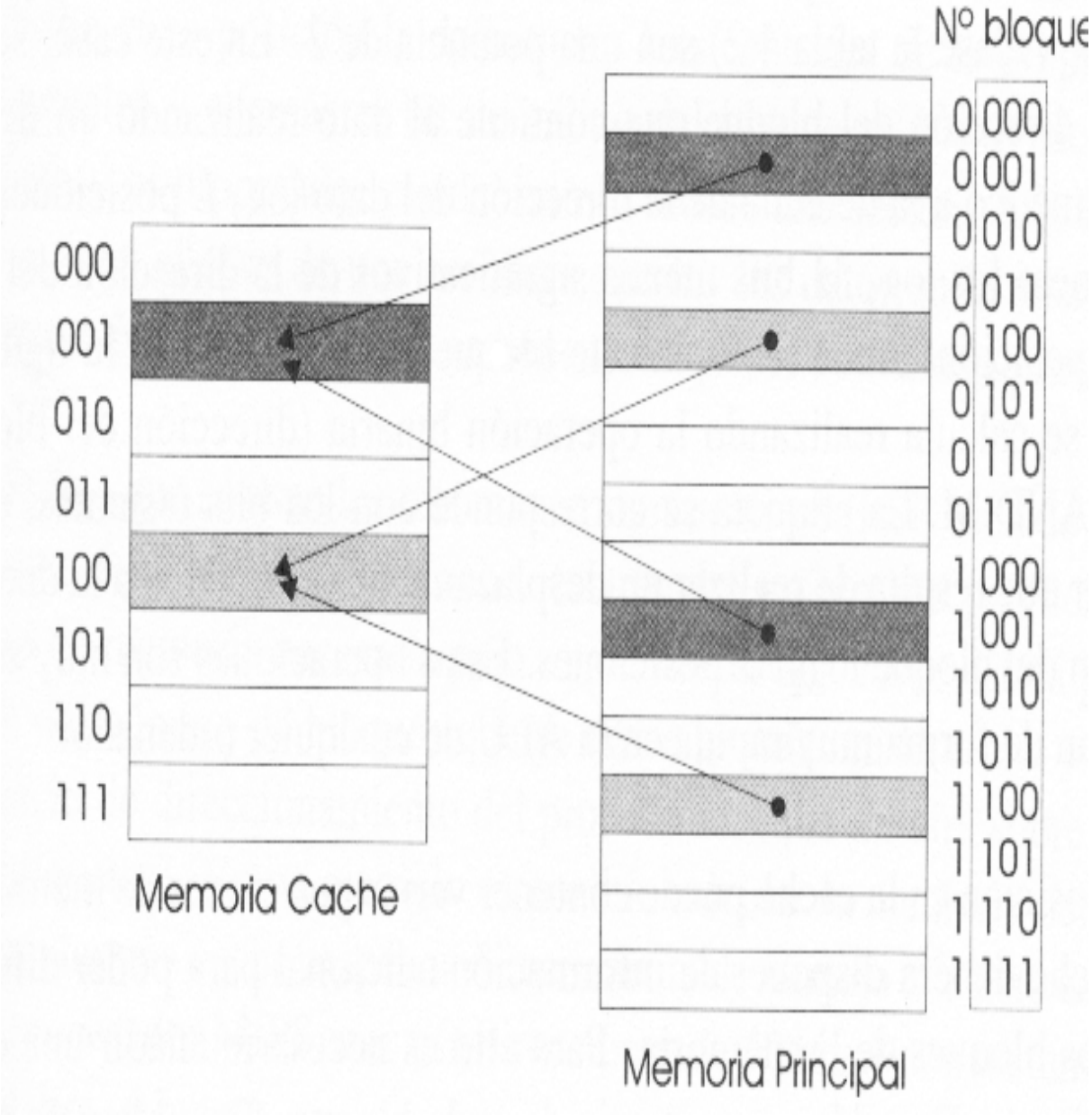
Jerarquía de memorias

- Objetivo en la jerarquía de memorias
  - Velocidad media muy alta (caché)
  - Tamaño de memoria muy alto (disco duro)
- Varios niveles de memoria:
  - Memoria caché SRAM pequeña -> alta velocidad
  - Memoria física DRAM mediana
  - Espacio en disco duro -> gran capacidad
- La jerarquía se basa en
  - Los programas suelen acceder a posiciones de memoria contíguas
  - Transferencia de bloques entre niveles de memoria

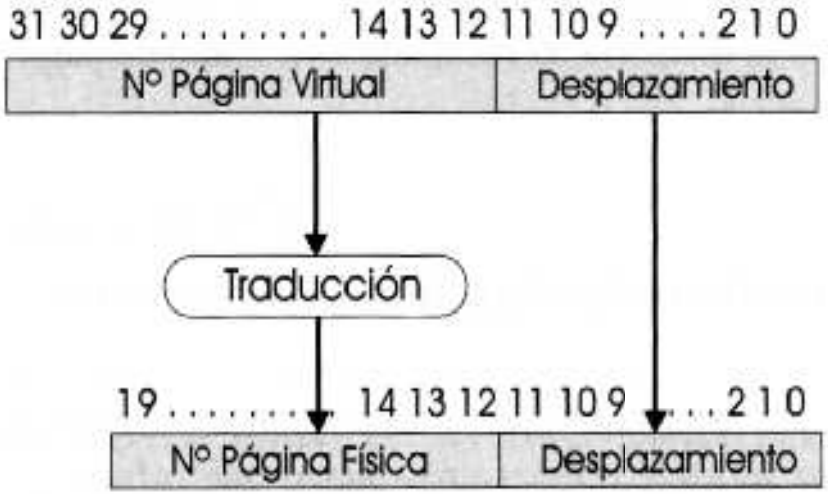


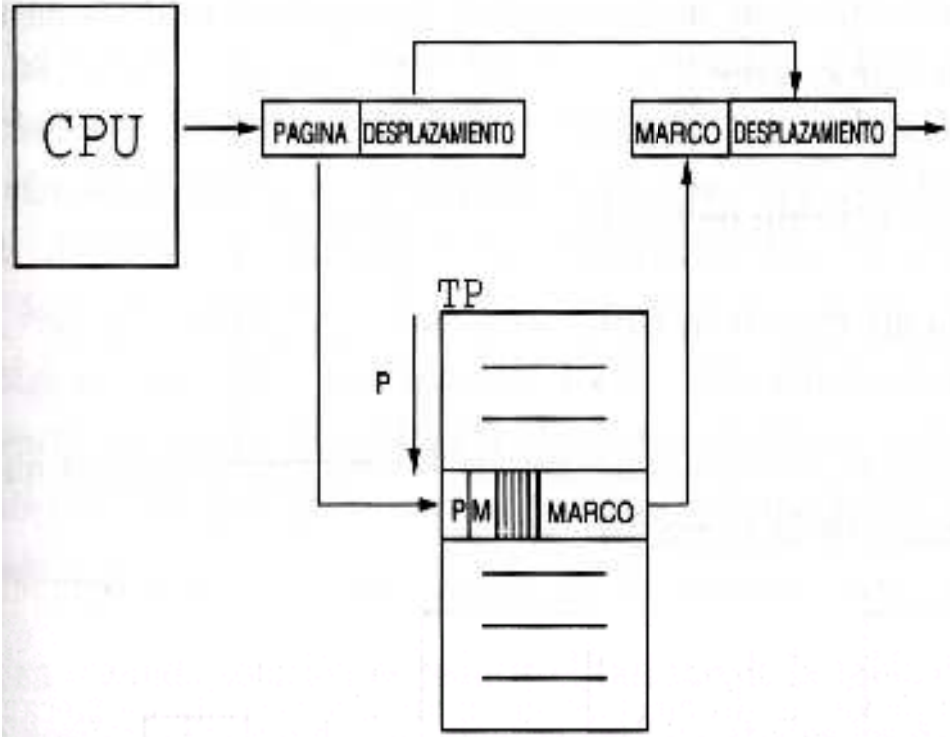
Memoria caché

- uP busca los datos en caché:
  - Acierto
  - Fallo de caché
- Accesos secuenciales dan alto porcentaje de aciertos -> velocidad media cercana a velocidad de caché
- Distintos tipos de caché
- 4 a 20 veces más rápida que la memoria física



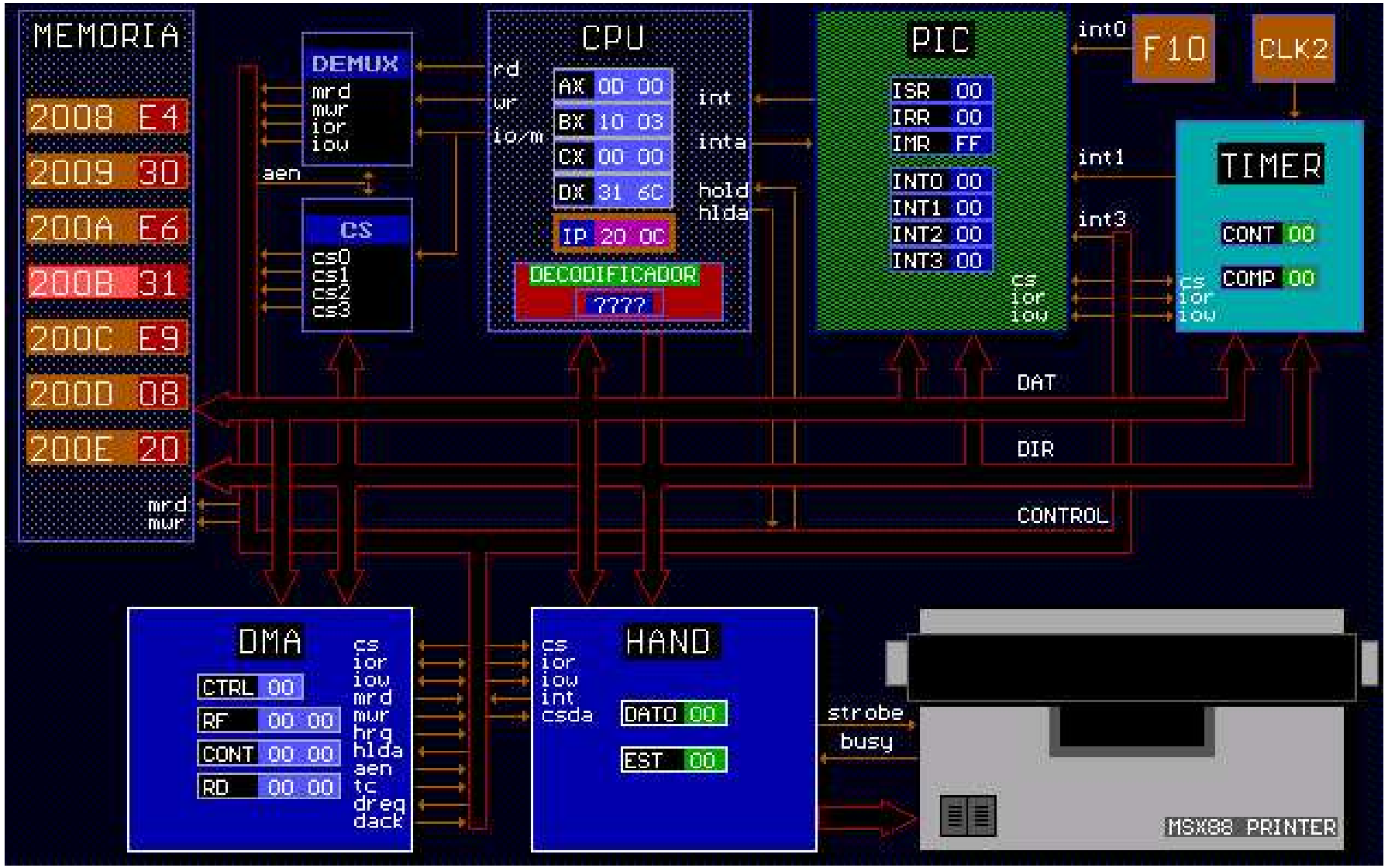
Memoria virtual





- Memoria paginada
  - Páginas de 4KB (12bits)
  - Reubicables
- Dirección física
  - 20 bits => 1MB
  - 256 páginas de 4KB
- Dirección virtual
  - 32 bits => 4GB
  - 1M páginas de 4KB -> **no** todas caben en memoria
  - Almacenamiento temporal en disco -> swap
- Tabla de páginas
  - Posición de las páginas
  - Presencia
- Accesos a memoria -> acierto/fallo

Dispositivos de E/S



- Comunicación con el exterior => E/S -> periféricos

Comunicación con los periféricos

- Direccionamiento
  - Bloques de E/S -> conexión CPU – periféricos
  - CPU - bloques E/S -> comunicación por buses (posiciones especiales de memoria o de E/S)
  - Bloques E/S – periféricos -> comunicación por conectores específicos. Se conectan al exterior mediante buffers
- Distintas formas de comunicación
  - Por sondeo (polling)
  - Por interrupción
  - Por acceso directo a memoria (DMA)

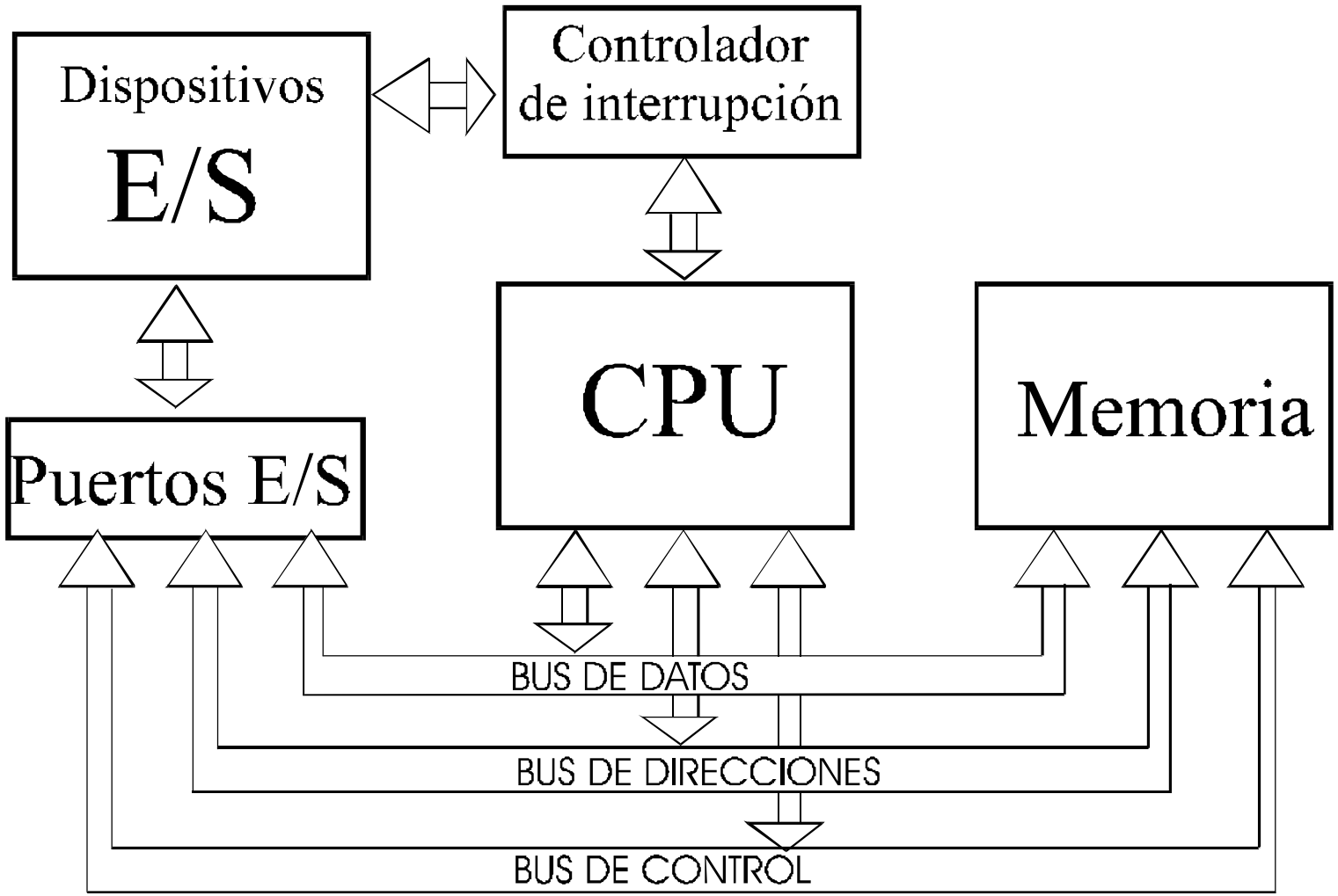
E/S por programa (polling)

- El programa debe sondear continuamente a los periféricos
- Se realiza por instrucciones de E/S
  - Específicas -> Mapa de memoria independiente
    - » IN = lectura de un puerto
    - » OUT = escritura en un puerto
  - Generales -> E/S mapeada en memoria
    - » MOV = lectura/escritura de puertos
- Problemas
  - Lentitud -> posible pérdida de datos
  - Desperdicio de tiempo de CPU

E/S por interrupción

- Necesidad de interrupciones -> evitar que la CPU tenga que estar pendiente de todos los periféricos
- Gestión de la interrupción
  - Cuando un periférico interrumpe a la CPU se corta la ejecución del programa y se ejecuta una rutina de tratamiento de la interrupción.
  - Permite la asignación de prioridades
- Dos modos:
  - Interrupción única
    - » Una línea de INT única
    - » Todos los periféricos se conectan a INT mediante una puerta OR
    - » Cuando se activa INT, la CPU comprueba todos los periféricos para ver cual es el peticionario de interrupción.
  - Interrupciones vectorizadas
    - » La línea de INT va conectada a un controlador de interrupción
    - » El controlador tiene una entrada de interrupción para cada dispositivo
    - » Cuando se activa INT la CPU se comunica con el controlador (ciclos de INTA) y lee un vector de interrupción
    - » Se especifica la rutina de tratamiento de la interrupción que especifique el vector de interrupción

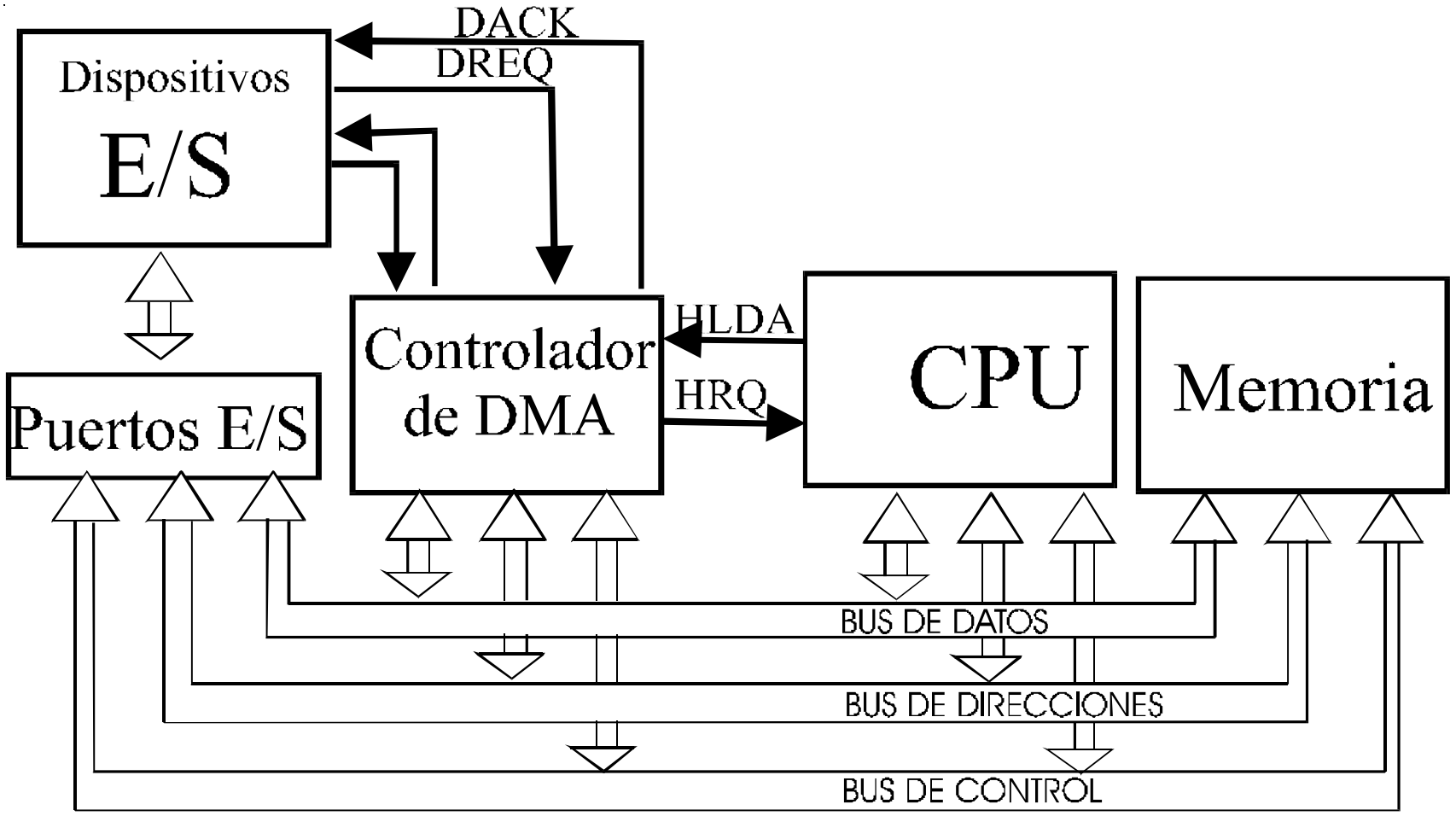
Conexión del controlador de interrupción en el sistema



E/S por DMA

- Necesidad de DMA -> liberar de trabajo a la CPU en grandes transferencias de datos
- Tipos de transferencia
  - E/S -> memoria
  - memoria -> E/S
  - memoria -> memoria
- Gestión de la transferencia
  - Se programa el controlador con las direcciones de origen y destino y el la cantidad de datos a transmitir
  - El controlador de DMA realiza la transferencia independientemente de la CPU
- Modos de funcionamiento
  - Con memoria multipuerta -> se reserva una puerta para la CPU y una puerta para cada canal de DMA
  - Por robo de ciclo -> el controlador de DMA toma el control de los buses para realizar las transferencias

E/S por DMA por robo de ciclo



DRQ/DACK -> petición de DMA por el periférico
HRQ/HLDA -> solicitud de buses al micro

Procesadores CISC y RISC

- Procesadores CISC (juego de instrucciones complejo)
  - Se utilizan cuando la CPU era más rápida que la memoria
  - Tienen un juego de instrucciones amplio
  - Las instrucciones son complejas (de alto nivel)
  - Las instrucciones se implementan con microprogramación
  - Utilizan mucho más la pila que los registros
- Procesadores RISC (juego de instrucciones reducido)
  - Se utilizan cuando la velocidad de la memoria alcanza a la de la CPU (aparición de la memoria caché)
  - Tienen pocas instrucciones (<50)
  - Las instrucciones son de bajo nivel
  - Las instrucciones se implementan por hardware
  - Se aumenta el número de registros para evitar los accesos a memoria
  - El formato de las instrucciones y su longitud están normalizados para favorecer el *pipeline*
  - Requieren el empleo de coprocesador, al ser el juego de instrucciones básico