

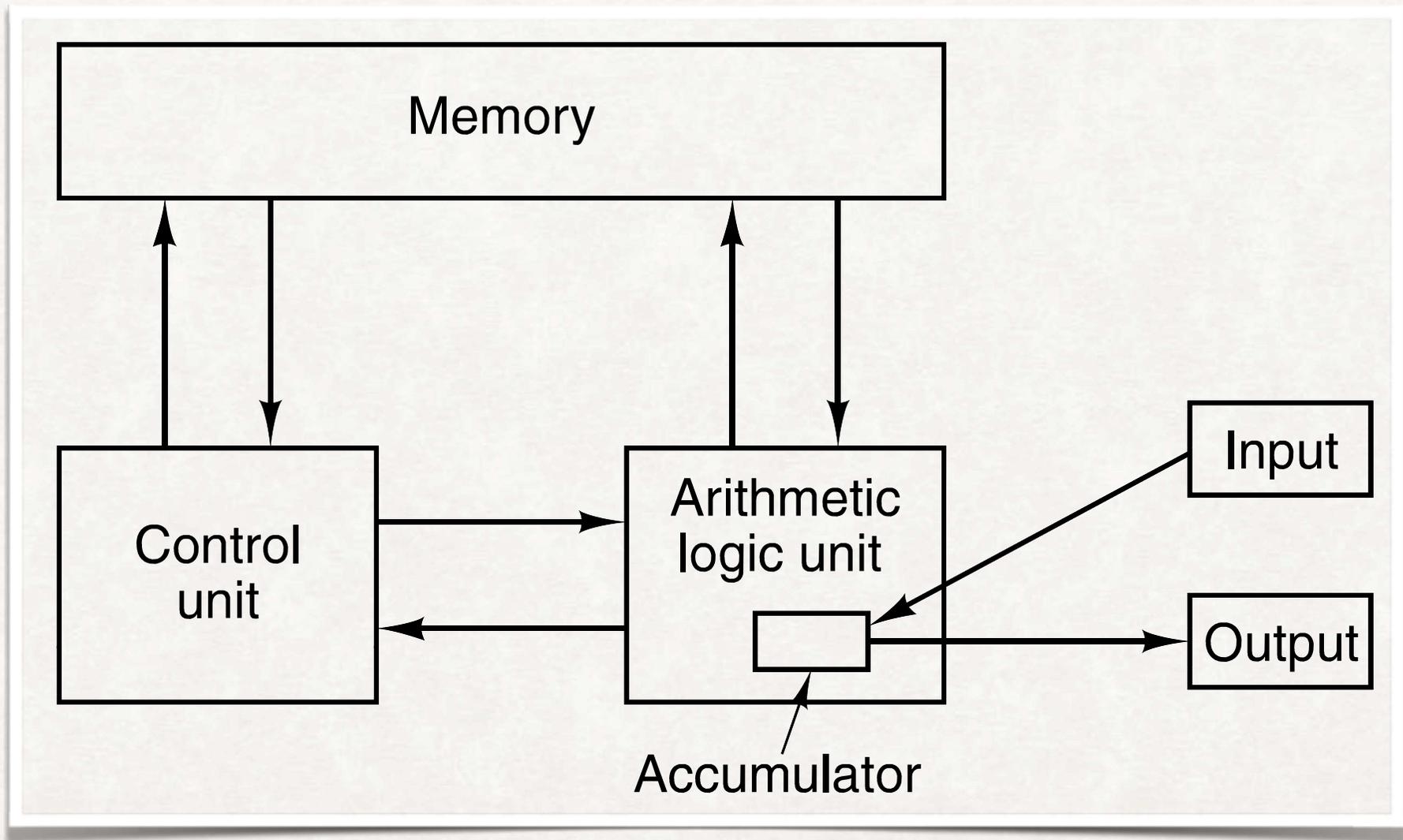
# TEMA 1

# FUNDAMENTOS



# FUNDAMENTOS

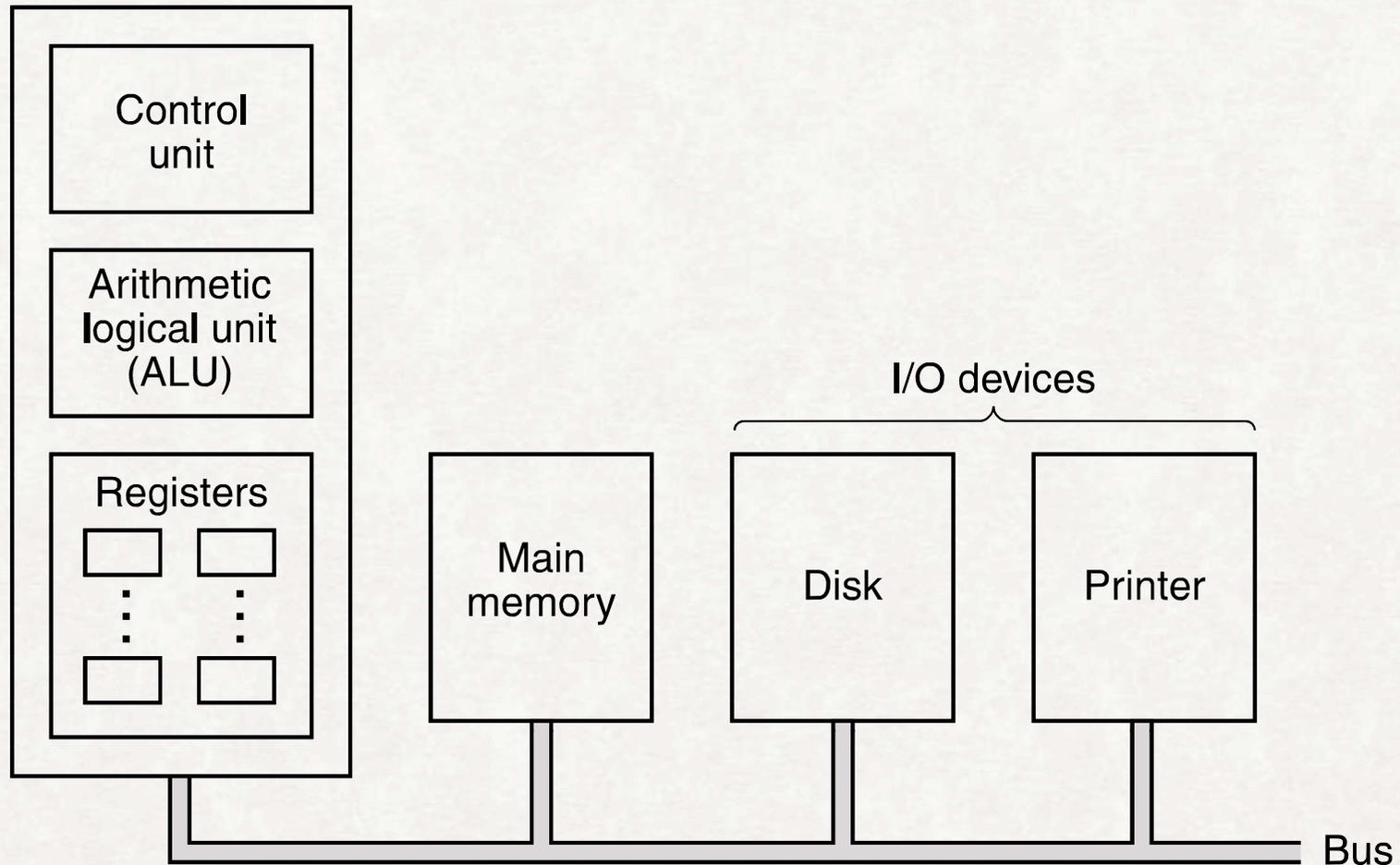
## MÁQUINA DE VON NEUMANN



# FUNDAMENTOS

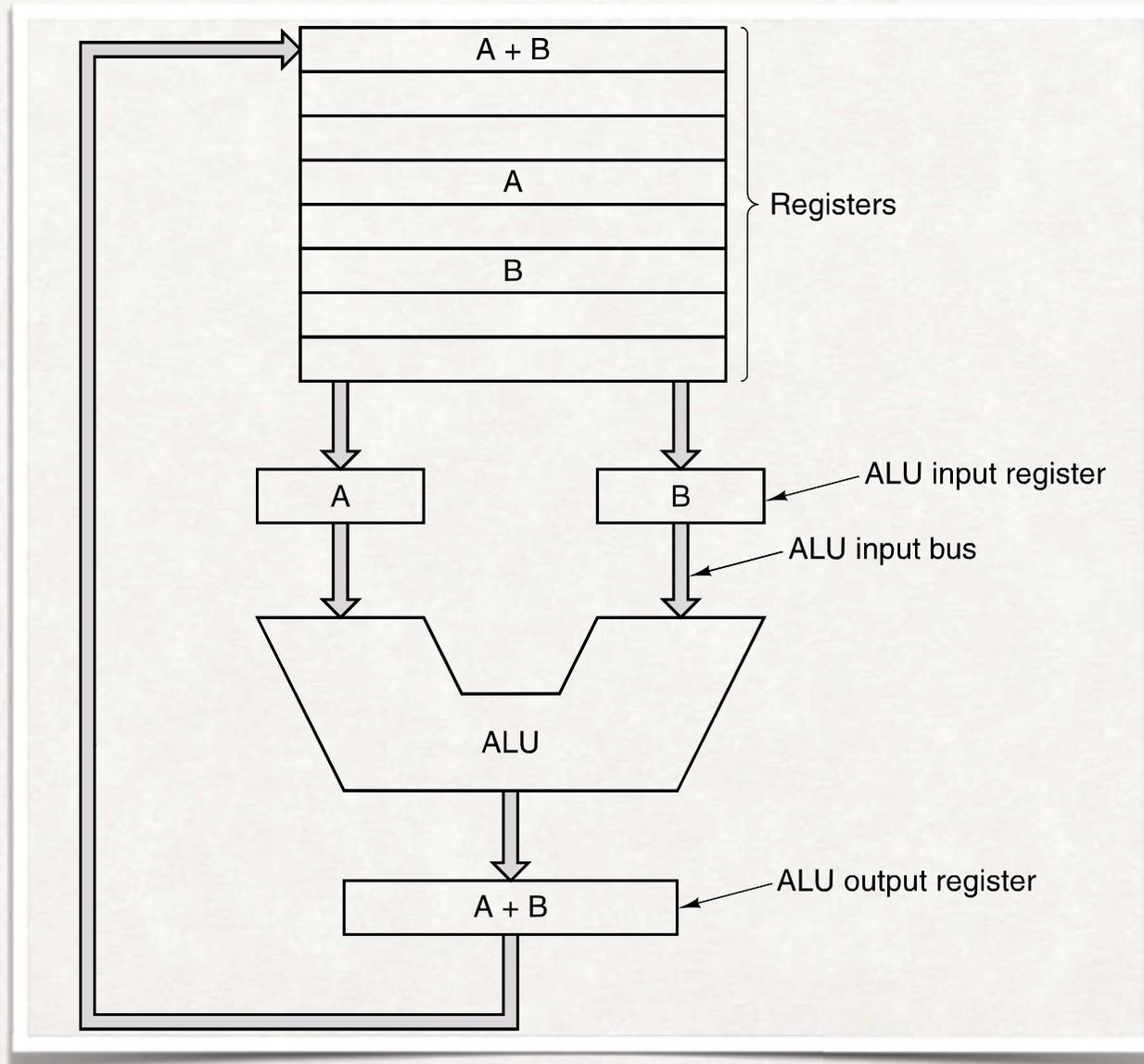
## ORGANIZACIÓN GENERAL DE UN COMPUTADOR

Central processing unit (CPU)



# PROCESADORES

## FLUJO DE LOS DATOS EN LA CPU



# PROCESADORES

## EL CICLO DE INSTRUCCIÓN

1. La siguiente instrucción se trae (*fetch*) desde la dirección de memoria apuntada por el registro contador de programa (PC) y se almacena en el registro de instrucciones (IR).
2. Se modifica el PC para que apunte a la siguiente instrucción en memoria.
3. Se descodifica la instrucción y se determina su tipo y la ubicación de sus operandos (registros o memoria)
4. Si algún operando se ubica en memoria, se calcula su posición, se obtiene el dato y se almacena en un registro de la CPU.
5. Se ejecuta la instrucción. Si el resultado debe almacenarse en memoria, se transfiere a la posición apropiada.
6. Vuelta al primer paso.

# PROCESADORES

## CPU MICROPROGRAMADA

- Cuenta con un *intérprete* que descompone la ejecución de cada instrucción en una secuencia de *microinstrucciones* mucho más sencillas, llamado microprograma.
- Los microprogramas se hallan almacenados en memoria rápida de sólo lectura (interna de la CPU).
- Se consigue:
  - Mayor flexibilidad (es posible añadir instrucciones, mejorarlas o corregirlas)
  - Permite diseñar y comprobar nuevas CPUs más rápidamente.
- Uno de los primeros  $\mu$ P microprogramados fue el MC68000.

# PROCESADORES

## RISC Y CISC

- RISC: Reduced Instruction Set Computer
  - Instrucciones que se puedan ejecutar y, especialmente, descodificar rápidamente.
  - Al simplificar la CPU, es posible acelerar considerablemente su funcionamiento.
  - En realidad, que haya pocas instrucciones no es lo más importante.
- CISC: Complex Instruction Set Computer
  - Muchos tipos de instrucciones, pero al final sólo un subconjunto de ellas se emplea con frecuencia.
  - Existen instrucciones que realizan operaciones relativamente complejas.
- CPUs Intel modernas: CISC microprogramadas con microinstrucciones RISC.

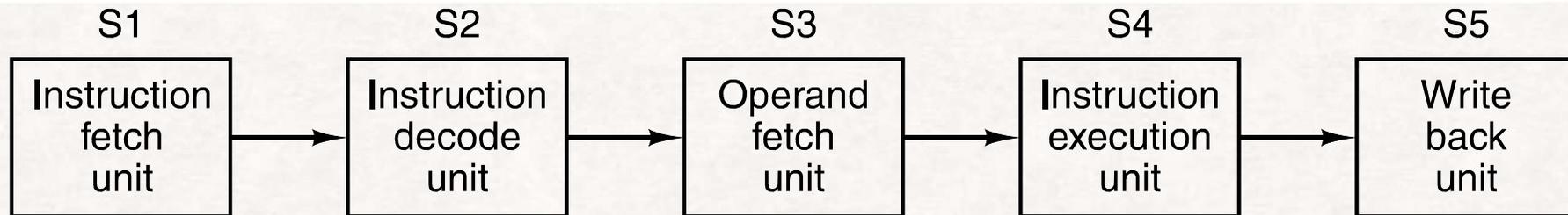
# PROCESADORES

## PRINCIPIOS DE DISEÑO DE LAS CPU<sub>s</sub> MODERNAS

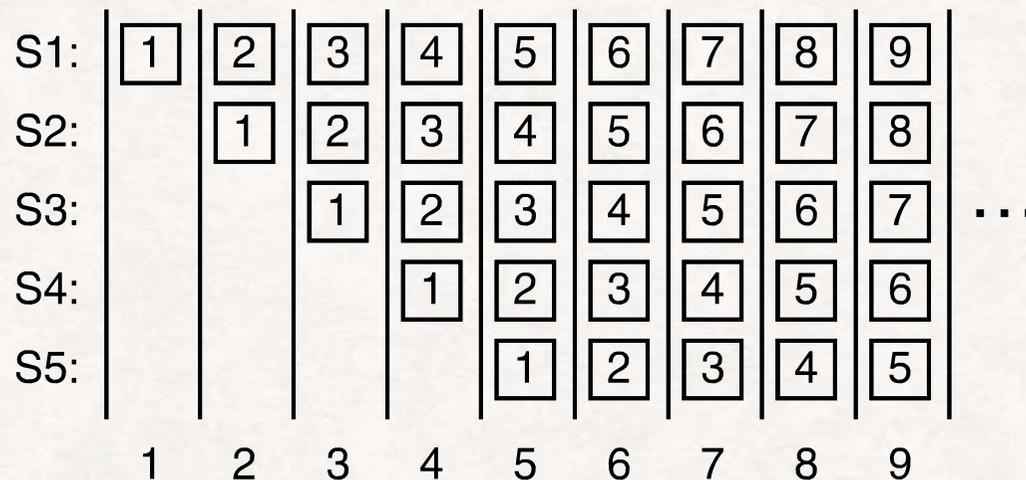
- Ejecución directa: bien RISC o bien CISC microprogramadas con intérprete RISC.
- Maximizar el ritmo al que se *lanzan (issue)* las instrucciones. Uso de paralelización: mientras una instrucción se ejecuta, otras pueden estar en *fetch*, decodificación, lectura de operandos, etc.
- Decodificación de instrucciones eficiente: códigos regulares, de longitud fija, y con pocos campos.
- Instrucciones dedicadas en exclusiva a movimientos de datos entre registros y memoria (*loads y stores*), puesto que el acceso a memoria es lo más lento. Las demás instrucciones siempre accederán a operandos en registros.
- Proporcionar una gran cantidad de registros, para reducir en lo posible los accesos a memoria, a costa de aumentar el movimiento de datos entre registros (mucho más rápido).

# PROCESADORES

## ARQUITECTURA ESCALAR CON PIPELINE DE CINCO FASES



(a)

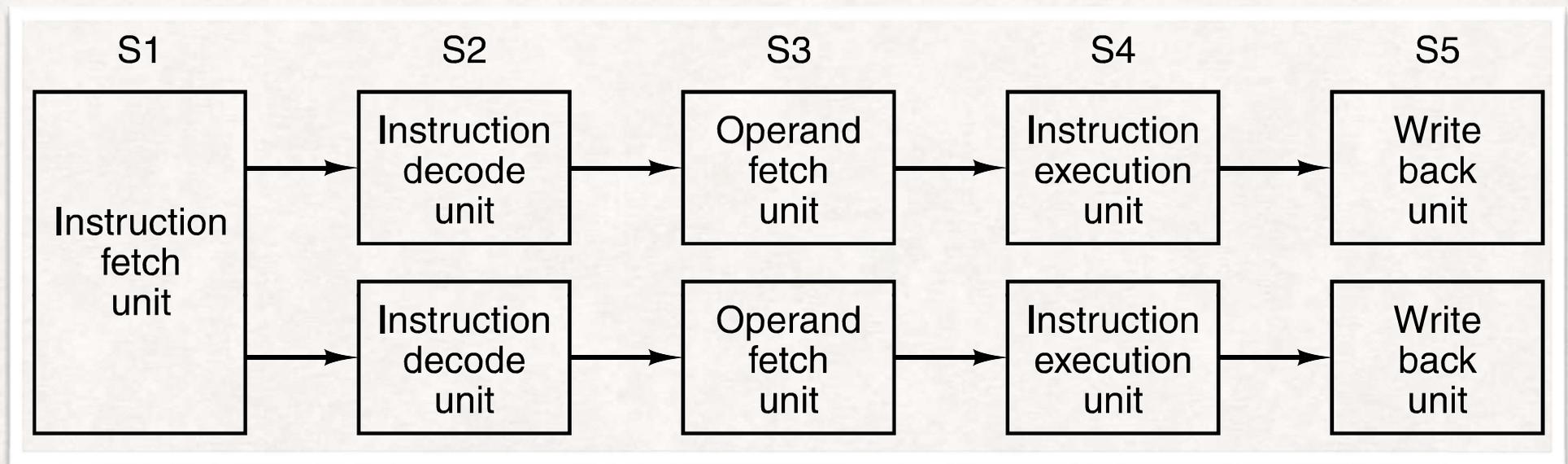


Time →

(b)

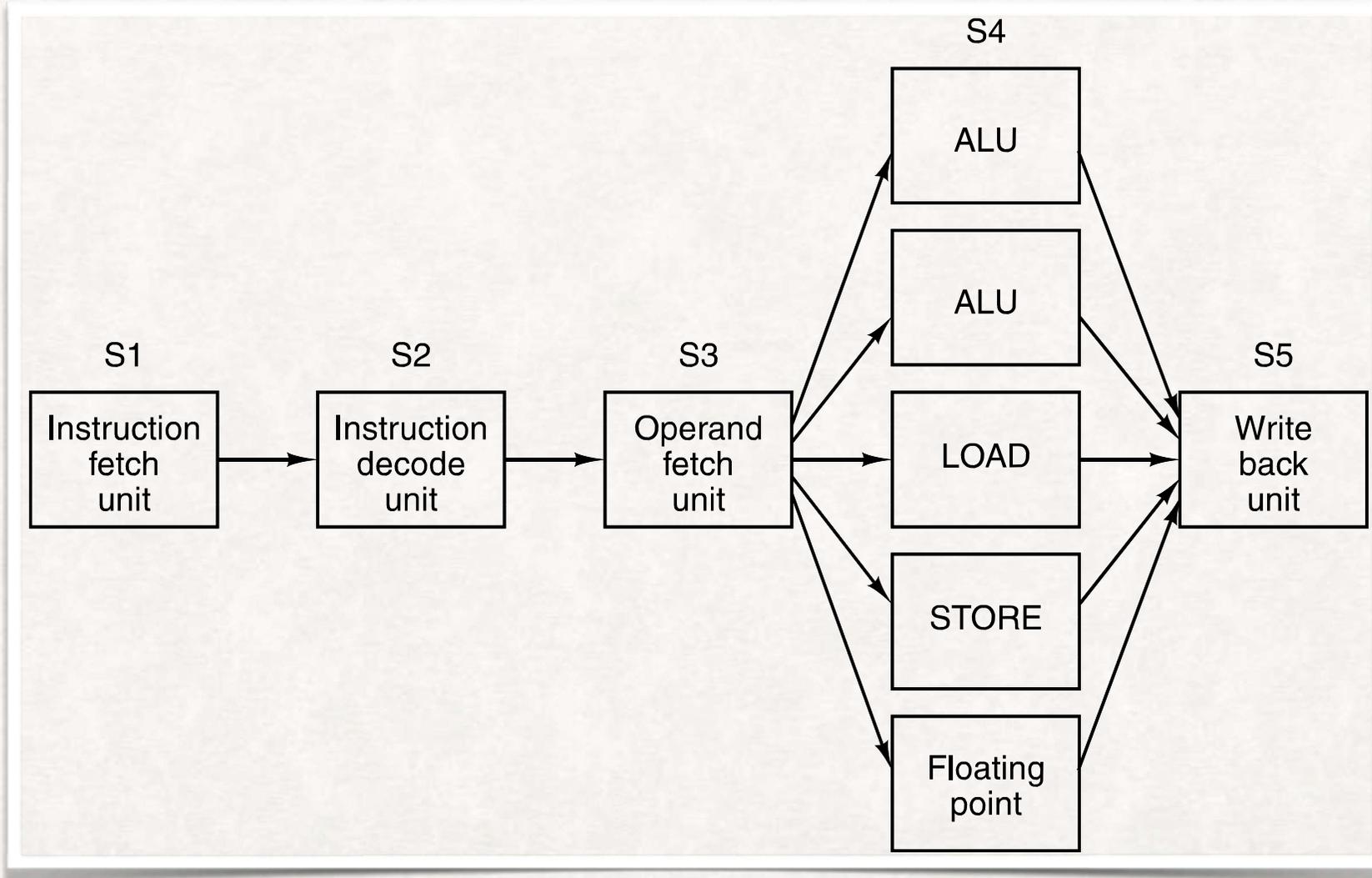
# PROCESADORES

## ARQUITECTURA SUPERESCALAR CON DOS PIPELINES



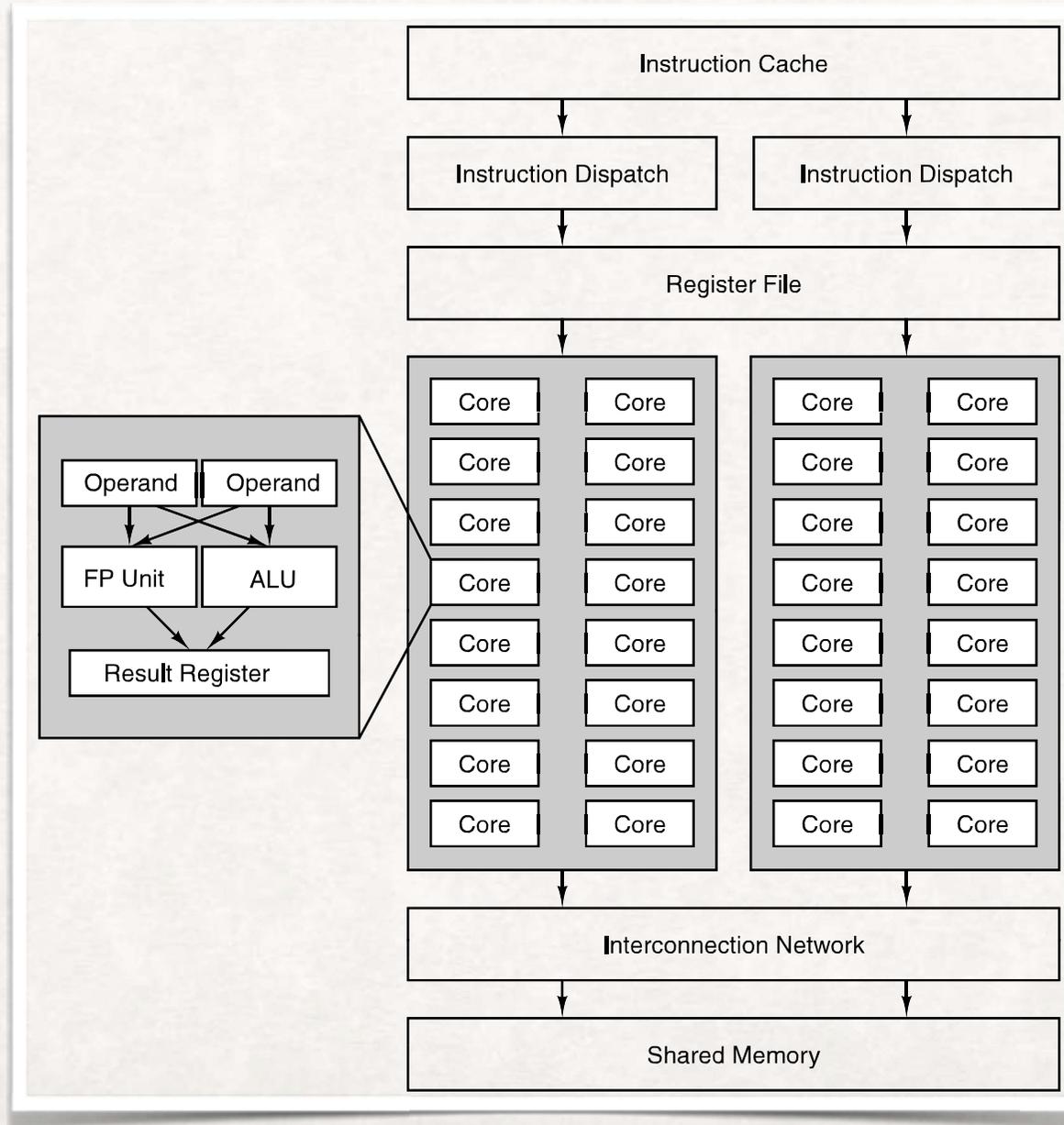
# PROCESADORES

## ARQUITECTURA SUPERSCALAR CON CINCO UNIDADES DE EJECUCIÓN



# PROCESADORES

## SINGLE INSTRUCTION MULTIPLE DATA (SIMD)



GPU  
nVidia Fermi:

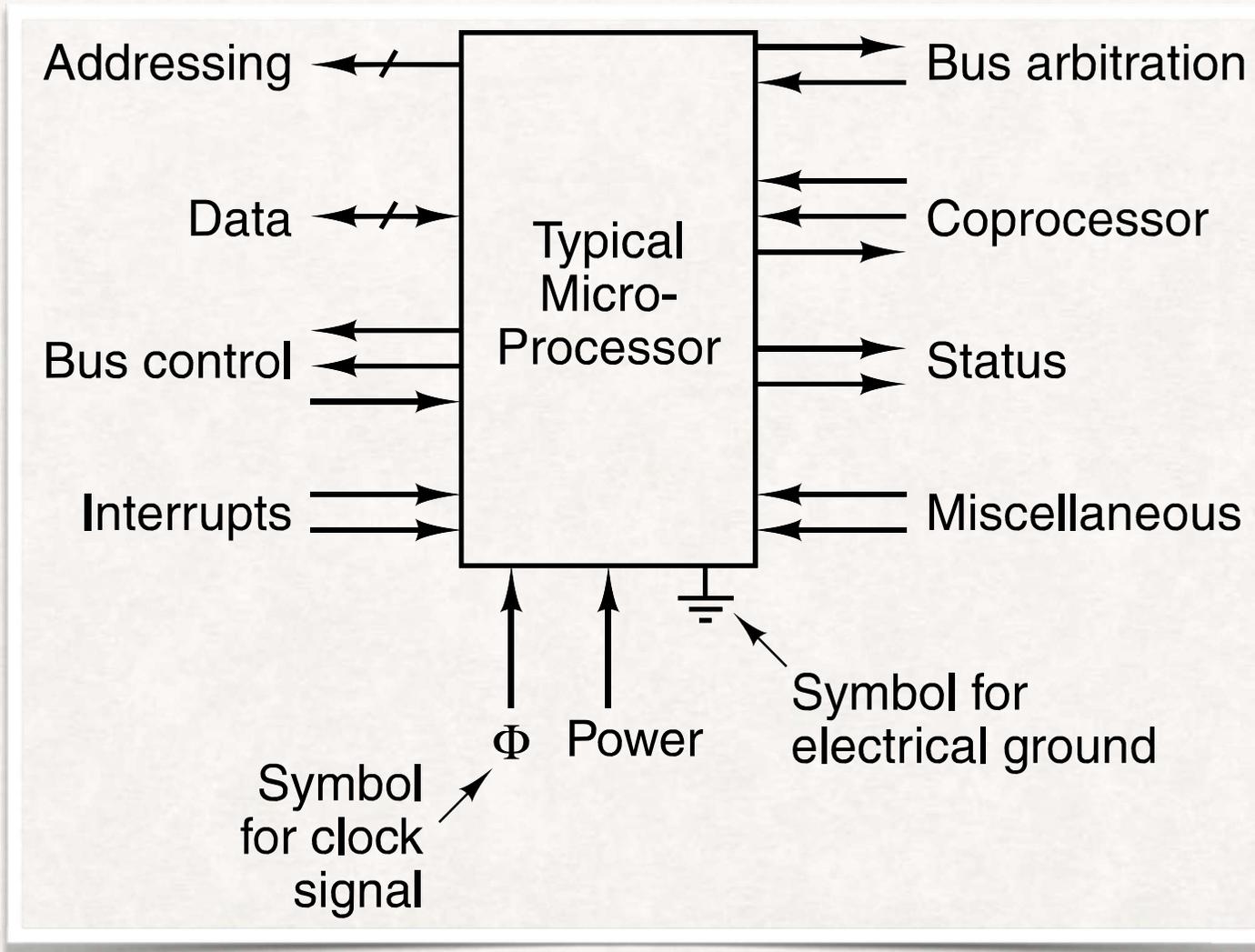
Intel:  
instrucciones SSE  
(Streaming SIMD  
Extension)

# CPU CHIPS Y BUSES

- La gran mayoría de CPUs se empaquetan en un único microchip.
- Existen también chips microcontroladores (CPU+E/S+memoria) y *single-chip computers* o SoCs (System-on-a-Chip).
- Todo CPU chip dispone de pins para interactuar con otros dispositivos del computador, típicamente clasificados en *buses*:
  - Direcciones: selección de elementos con los que transferir los datos (posiciones de memoria, registros de dispositivos de E/S, etc.)
  - Datos: líneas a través de las cuales se realiza la transferencia de los datos (en ambos sentidos).
  - Control: control de transferencias de datos, interrupciones, arbitración de acceso a los buses, estado, etc.

# CPU CHIPS Y BUSES

## ESQUEMA GENÉRICO



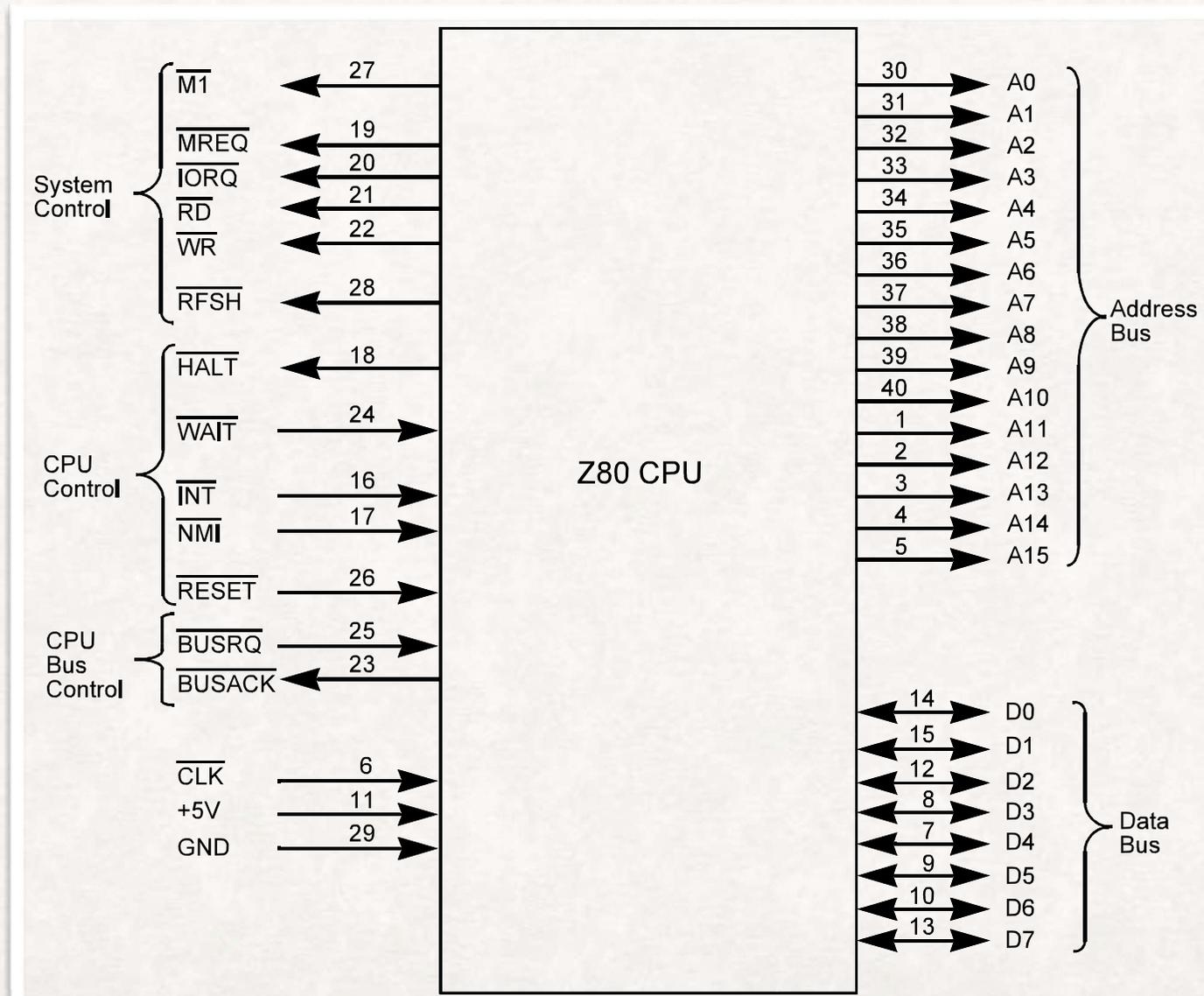
# CPU CHIPS Y BUSES

## BUSES Y ESPACIO DE DIRECCIONAMIENTO

- La anchura de un bus consiste en el número de líneas que lo componen.
- La anchura del bus de datos determina el tamaño máximo en bits (longitud de palabra o *word length*) que la CPU puede transferir en una sola operación de lectura o escritura.
- La anchura del bus de direcciones determina el tamaño del *espacio de direccionamiento (address space)*: número de elementos direccionables independientemente ( $2^n$  para un bus de  $n$  líneas).
- La anchura del bus de datos determina el nº de bits de cada elemento del espacio de direcciones.

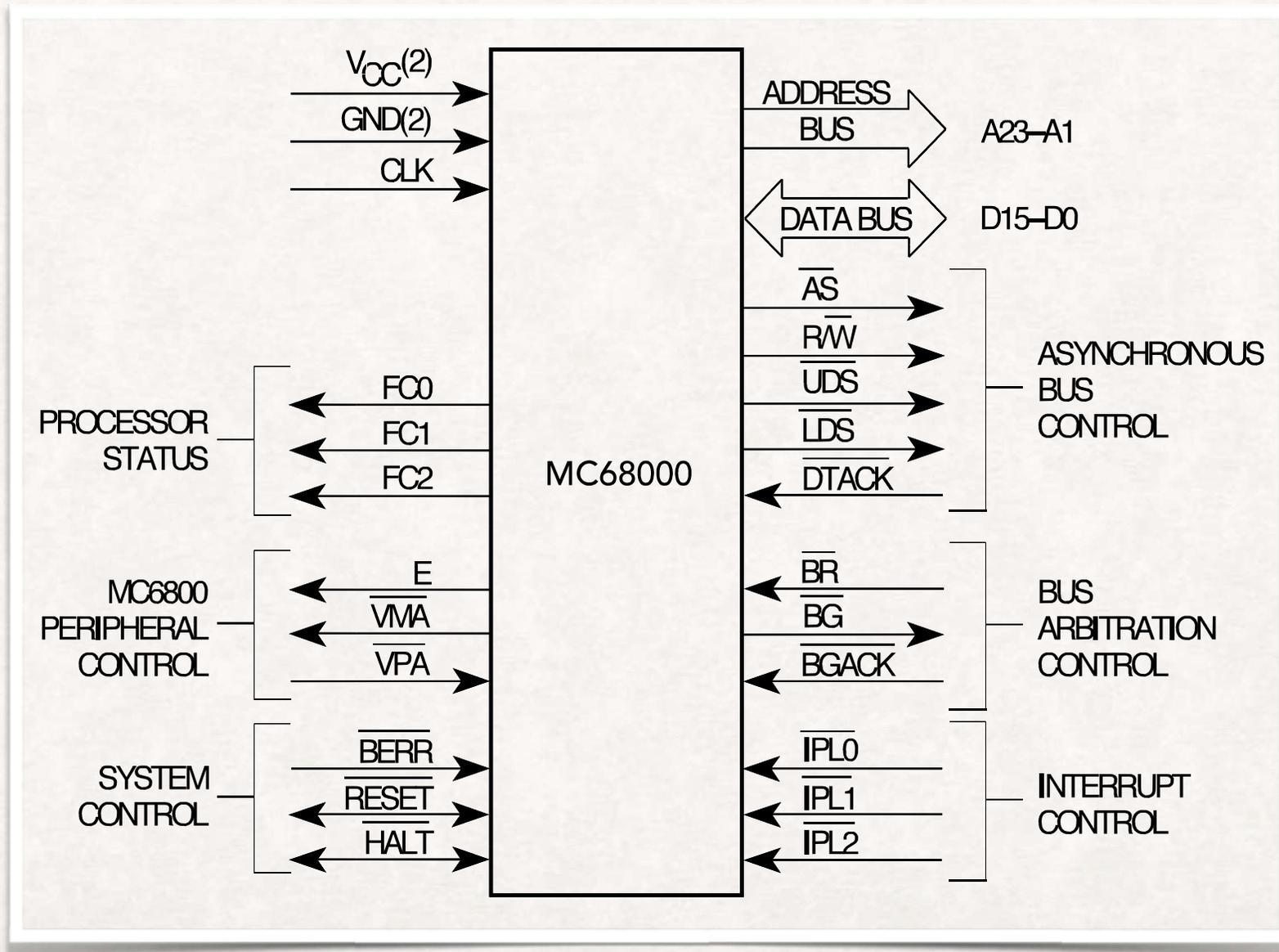
# CPU CHIPS Y BUSES

## ZILOG Z-80 (1976)



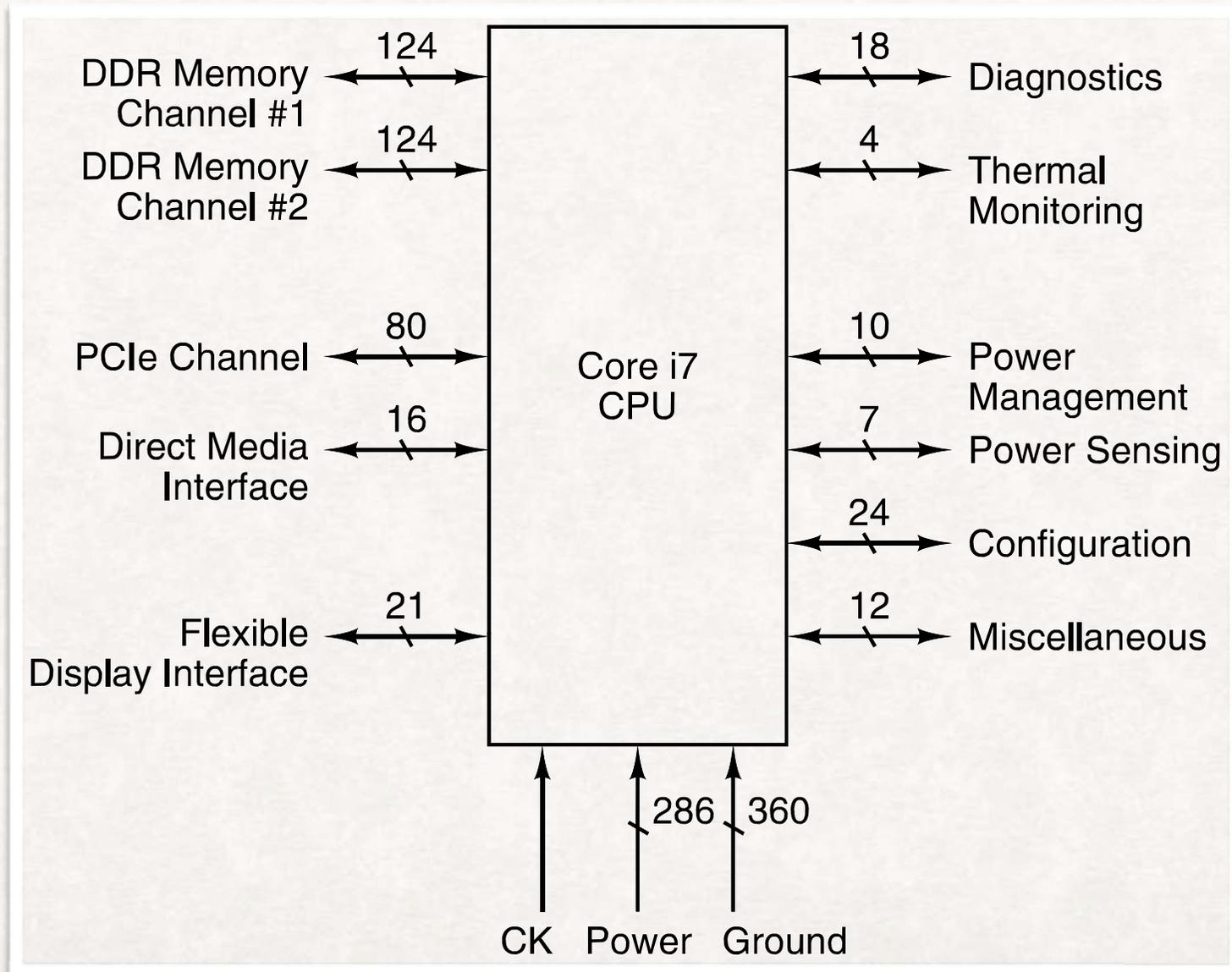
# CPU CHIPS Y BUSES

MOTOROLA MC68000 (1980)



# CPU CHIPS Y BUSES

INTEL CORE i7-2600 "SANDY BRIDGE" (2011)



# CICLOS DE LECTURA Y ESCRITURA

- Se trata de las operaciones de transferencia de datos entre la CPU y otros dispositivos del computador (memoria, E/S, etc.) a través del bus de datos.
  - Lectura: CPU ← dispositivo
  - Escritura: CPU → dispositivo
- Cada ciclo de lectura y escritura se realiza con el elemento del espacio de direcciones cuya dirección es especificada por la CPU en su bus de direcciones.
- En los ciclos de lectura y escritura están involucradas ciertas señales del bus de control (ejemplos: señales con la que la CPU indica si el ciclo es de lectura o de escritura, que ya ha colocado la dirección en el bus, reloj, etc.)
- Pueden ser *síncronos* (cada paso se realiza en instantes determinados sincronizados con la señal de reloj) o *asíncronos* (regulados mediante un protocolo entre la CPU y el dispositivo implicado).

# CICLOS DE LECTURA Y ESCRITURA

## CICLO DE LECTURA

1. La CPU coloca la dirección en su bus de direcciones.
2. La CPU indica que el ciclo es de lectura.
3. La CPU indica que el estado del bus de direcciones es válido.
4. El dispositivo implicado deposita el dato en el bus de datos.
5. La CPU recoge el dato desde el bus de datos.
6. La CPU da por finalizado el ciclo desactivando las señales que se activaron en los pasos 2 y 3.
7. Se procede con el siguiente ciclo de lectura o de escritura.

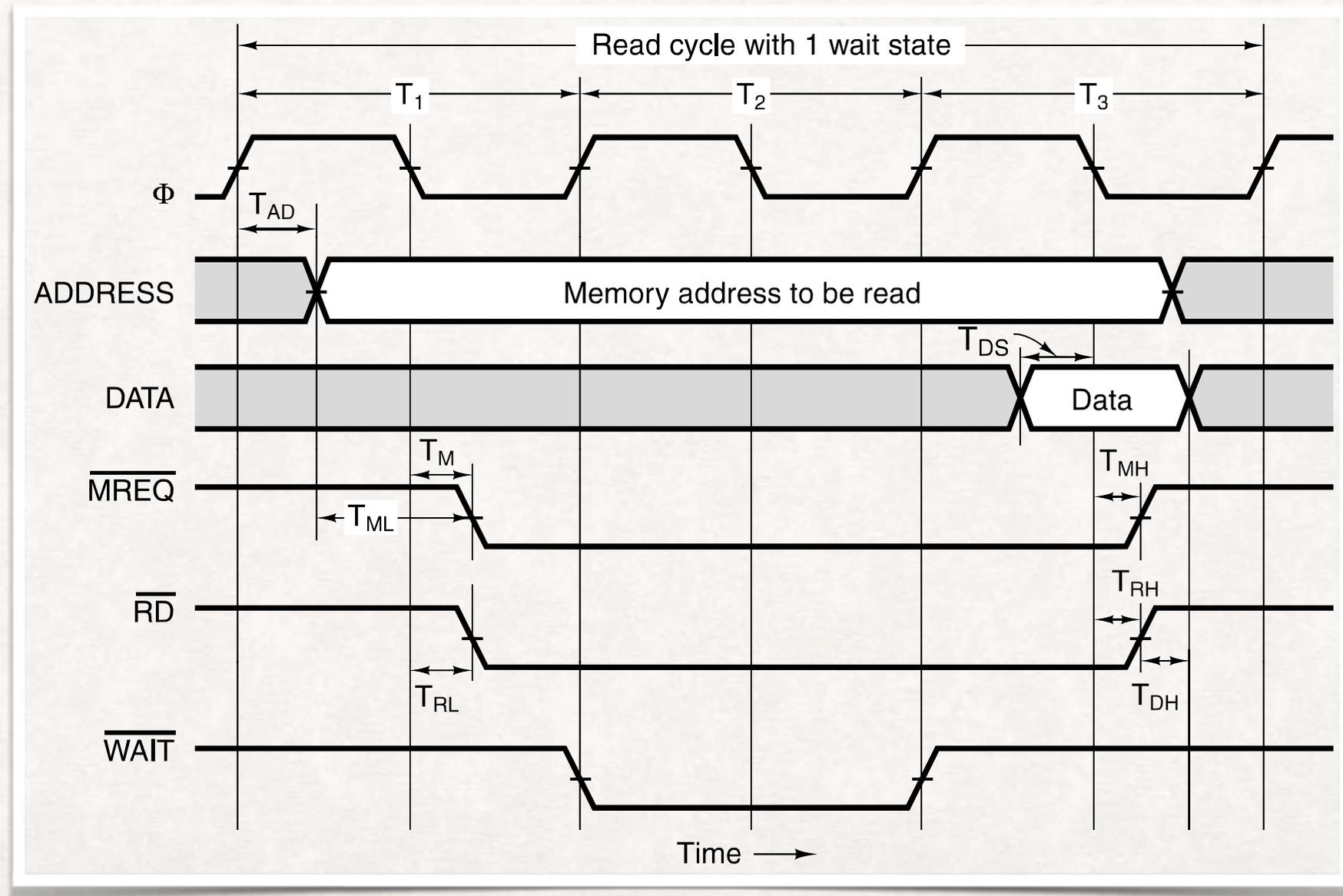
# CICLOS DE LECTURA Y ESCRITURA

## CICLO DE ESCRITURA

1. La CPU coloca la dirección en su bus de direcciones.
2. La CPU coloca el dato en su bus de datos.
3. La CPU indica que el ciclo es de escritura.
4. La CPU indica que el estado del bus de direcciones es válido.
5. El dispositivo implicado recoge el dato desde el bus de datos.
6. La CPU da por finalizado el ciclo desactivando las señales que se activaron en los pasos 2 y 3, y retirando el dato del bus de datos.
7. Se procede con el siguiente ciclo de lectura o de escritura.

# CICLOS DE LECTURA Y ESCRITURA

## EJEMPLO: CICLO DE LECTURA SÍNCRONO (I)



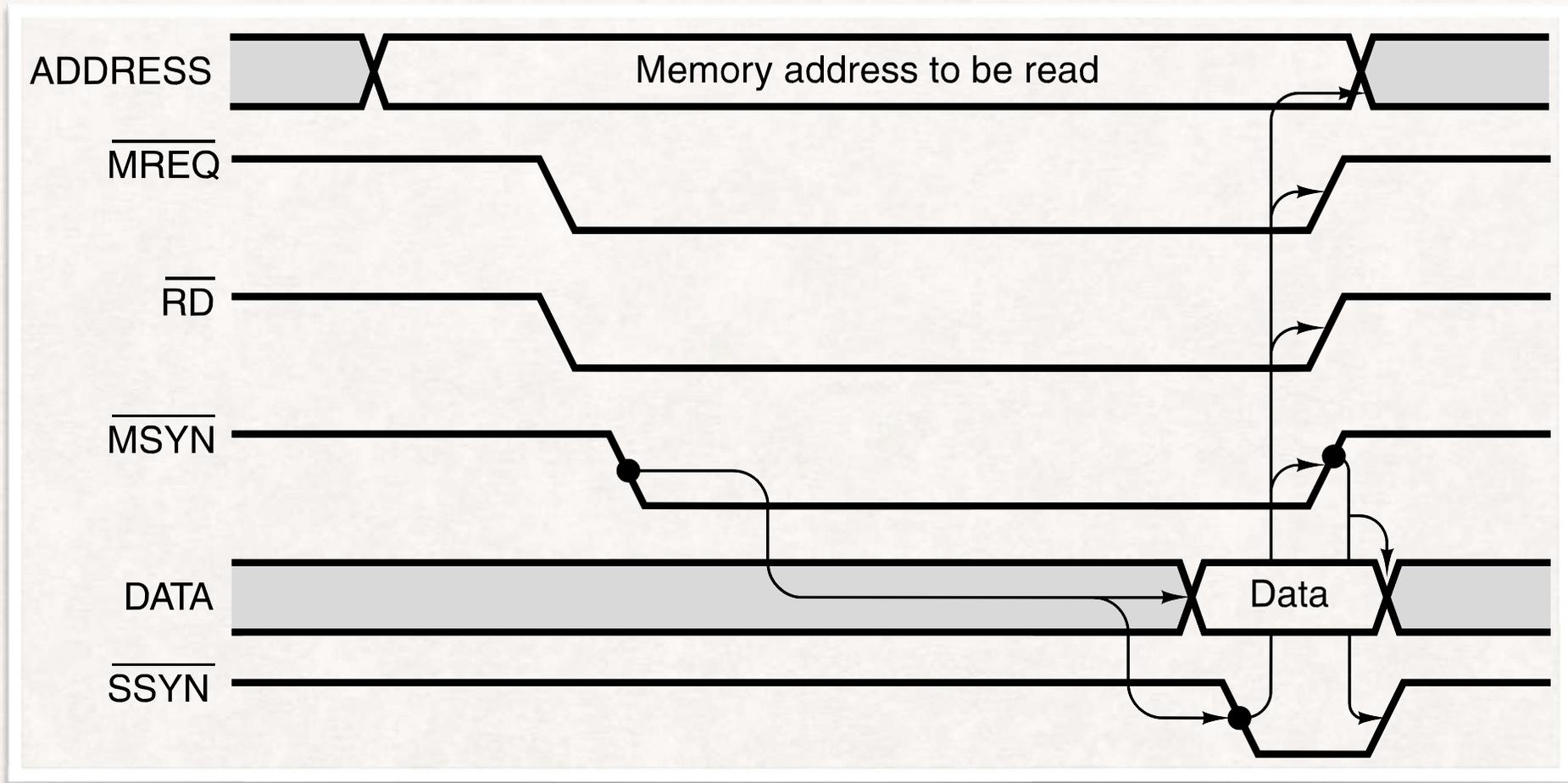
# CICLOS DE LECTURA Y ESCRITURA

## EJEMPLO: CICLO DE LECTURA SÍNCRONO (II)

Symbol	Parameter	Min	Max	Unit
$T_{AD}$	Address output delay		4	nsec
$T_{ML}$	Address stable prior to $\overline{MREQ}$	2		nsec
$T_M$	$\overline{MREQ}$ delay from falling edge of $\Phi$ in $T_1$		3	nsec
$T_{RL}$	RD delay from falling edge of $\Phi$ in $T_1$		3	nsec
$T_{DS}$	Data setup time prior to falling edge of $\Phi$	2		nsec
$T_{MH}$	$\overline{MREQ}$ delay from falling edge of $\Phi$ in $T_3$		3	nsec
$T_{RH}$	$\overline{RD}$ delay from falling edge of $\Phi$ in $T_3$		3	nsec
$T_{DH}$	Data hold time from negation of $\overline{RD}$	0		nsec

# CICLOS DE LECTURA Y ESCRITURA

EJEMPLO: CICLO DE LECTURA ASÍNCRONO



$\overline{\text{MSYN}}$  tiene una función similar a  $\overline{\text{LDS}}/\overline{\text{UDS}}$  en el MC68000

$\overline{\text{SSYN}}$  tiene una función similar a  $\overline{\text{DTACK}}$  en el MC68000