

# Sistema d'Entrades i Sortides

Els Interfícies d'Entrada/Sortida els podem dividir en dos parts:

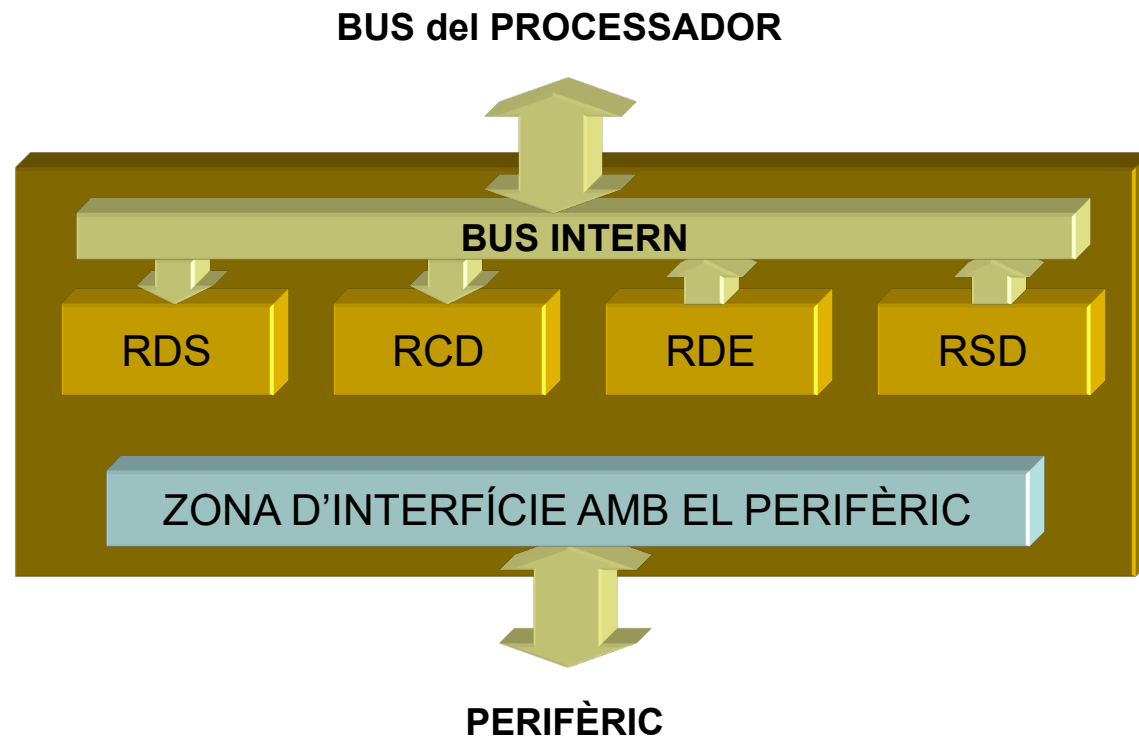
- CIRCUIT D'INTERFÍCIE (HARDWARE):
  - Adapten nivells de senyal entre dispositius.
  - Sincronitzen les transferències entre el sistema de busos del computador i les connexions dels perifèrics.
- PROGRAMES D'INTERFÍCIE (SOFTWARE)\*, 2 funcions:
  - Inicialització de la interfície d'E/S. Habitualment consisteix en programar registres de control i llegir registres de *status*.
  - Controlar les transferències d'informació (comunicació).

\*Això és el que normalment anomenem *drivers*.

## Interfície d'E/S Genèric

Qualsevol Interfície d'Entrada/Sortida el podrem descriure com el següent

- RDS (Registre de Dades de Sortida)
- RDE (Registre de Dades d'Entrada)
- RCD (Registre de Control de Perifèric)
- RSD (Registre de Status de Perifèric)
- Zona d'interfície amb el Perifèric, depèn del perifèric al que connectem



## Exemple d'un Interfície d'Impressora

## INTERFICIE HARDWARE

**RDE:** No hi ha. L'impressora és un dispositiu de sortida.

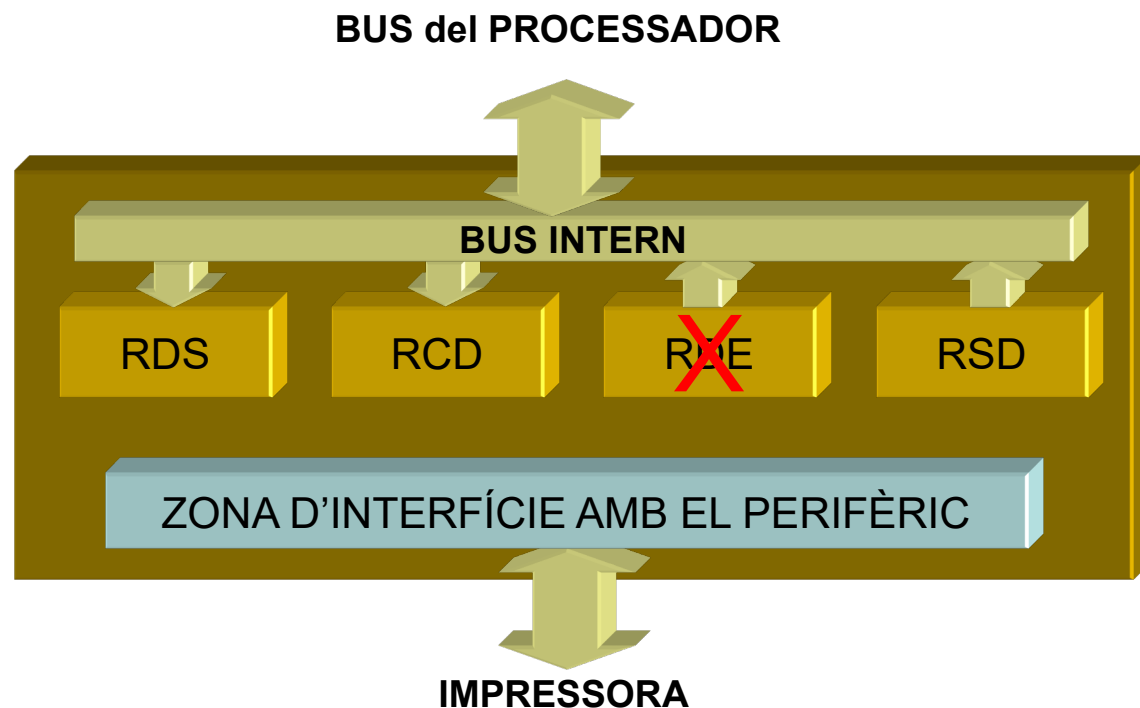
**RDS:** Port de 8 bits, on el sistema escriu el codi ASCII del caràcter que vol imprimir.

**RSD:**

- Bit Select: Indica impressora activa.
- Bit PE: *Paper Empty*.
- Bit Busy: Indica impressora ocupada
- Bit Ack: Indica que la impressora ha rebut la informació.

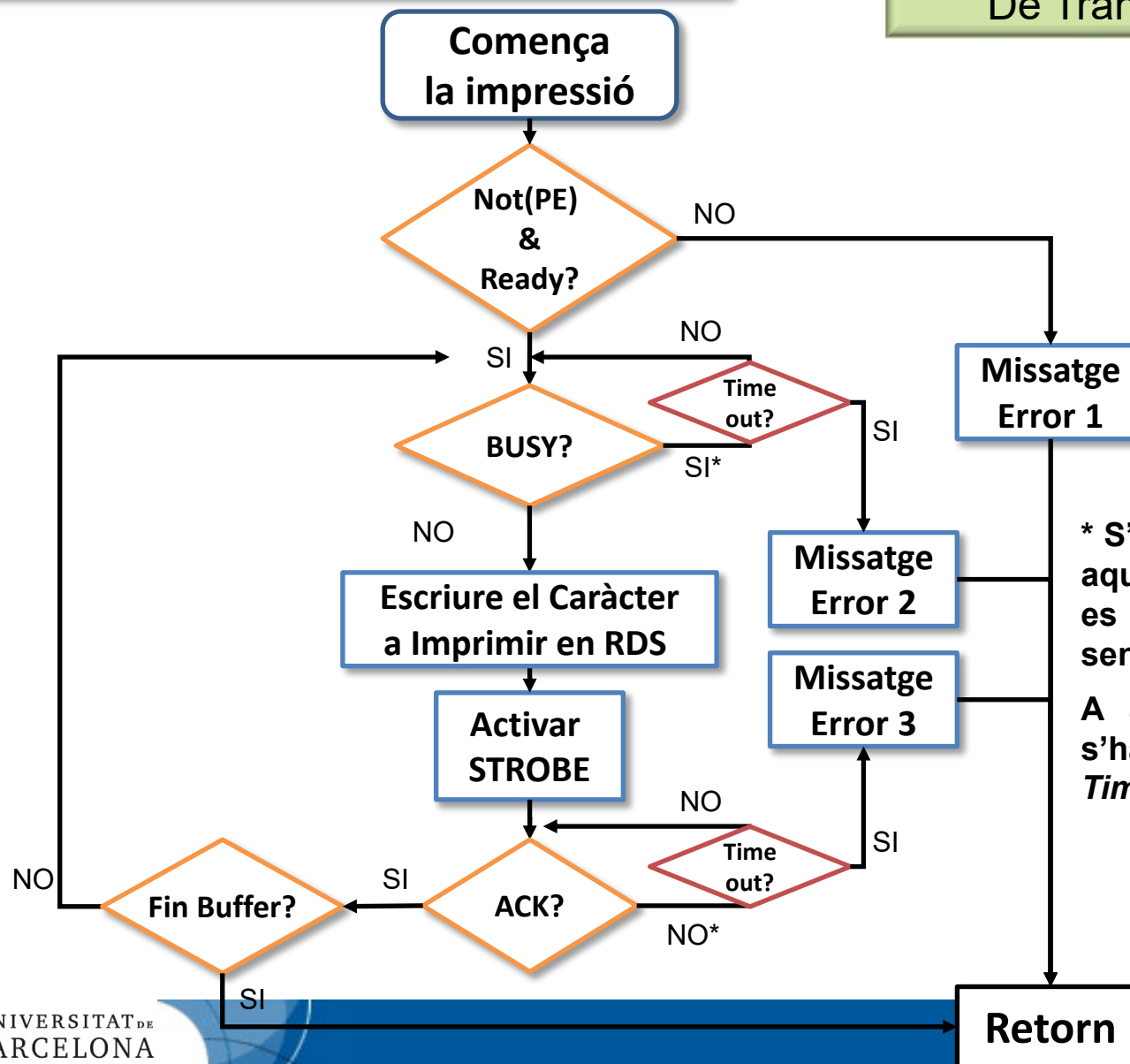
**RCD:**

- Bit Reset: Quan s'activa es inicialitza la impressora.
- Bit Strobe: Informa a la impressora que té una dada per imprimir.

**Zona d'interfície amb el Perifèric:**

- Bus de 8 dades de sortida.
- Línia d'entrada de BUSY.
- Línia d'entrada ACK.
- Línies d'entrada PE i Select.
- Línia de sortida de STROBE.
- Línia de sortida de Reset.

## Exemple d'un Interfície d'Impressora

INTERFICIE SOFTWARE  
De Transferència

\* S'ha d'anar amb compte a aquestes situacions perquè es pot entrar a un bucle sense sortida.

A aquests casos sempre s'ha d'afegir una rutina de *Timeout*.

## Mètodes d'Entrada/Sortida

Es solen distingir 3 tècniques d'entrada/sortida (E/S):

1. E/S mitjançant Programa.
2. E/S amb Interrupcions.
3. E/S mitjançant Accés Directe a Memòria (DMA).

I els criteris que s'han de tenir en compte per triar quina/es és/són més adient/s són:

- Temps Disponible de CPU.
- Urgència amb la que el perifèric s'ha de comunicar.
- Velocitat de Transferència.

## E/S per Programa

- És el programa qui pren la iniciativa de realitzar una transferència d'E/S.
- Es diu que les operacions d'E/S es fan de forma SÍNCRONA amb l'execució del programa.
- Es solen escriure unes funcions que crida el programa de forma seqüencial a l'execució. Habitualment compren 2 parts:
  1. Inicialització del interfície (escriptura de registres de control i lectura de registres de *status*).
  2. Realització de la transferència de dades. A més de escriure/llegir dades a registres RDS/RDE o *buffers*, molts cops també s'han de gestionar registres de control i *status*).

Avantatges	Inconvenients
Senzill	Pèrdua de temps de la CPU
Econòmic	Lent a l'hora d'atendre una petició de transferència

## E/S per Interrupció

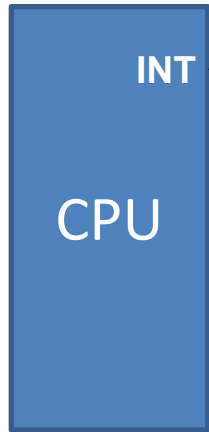
- Es el perifèric qui pren la iniciativa de realitzar una transferència d'E/S. Per fer això, el interfície d'E/S del perifèric ha d'activar un senyal de la CPU de petició d'Interrupció (INT).
- Es diu que les operacions d'E/S es fan de forma ASÍNCRONA amb l'execució del programa.
- Igual que abans, hem d'escriure funcions per Inicialitzar i per fer les Transferències. La diferència està en que ara la crida a la funció de transferència la fa el perifèric.

Avantatges	Inconvenients
Ràpid a l'hora d'atendre una petició de transferència.	Necessita hardware adicional.
La CPU no ha de perdre temps explorant perifèrics.	



## E/S per Interrupció

Seqüència típica d'una petició d'interrupció:



- El processador habitualment estarà executant programes.
- Quan el perifèric necessita ser atès, es a dir fer transferència de dades, el seu interfície d'E/S activa el senyal de petició d'interrupció del processador.
- Si s'accepta la petició d'interrupció, llavors:
  1. Es finalitza el que està fent (instrucció, cicle de bus...).
  2. Guarda a la pila la posició del Comptador de Programa (PC) i el registre de *status*.
  3. Executa la rutina d'atenció a la interrupció (IHR\*).
  4. Retorna al punt en que havia deixat l'execució del programa (PC i *status*).

\*IHR: *Interrupt Handle Routine*. També es diu ISR, *Interrupt Service Routine*.

## E/S per Interrupció

Exemple d'un interfície per a un Teclat

**RDS:** Aquí no fa falta, no el posarem.

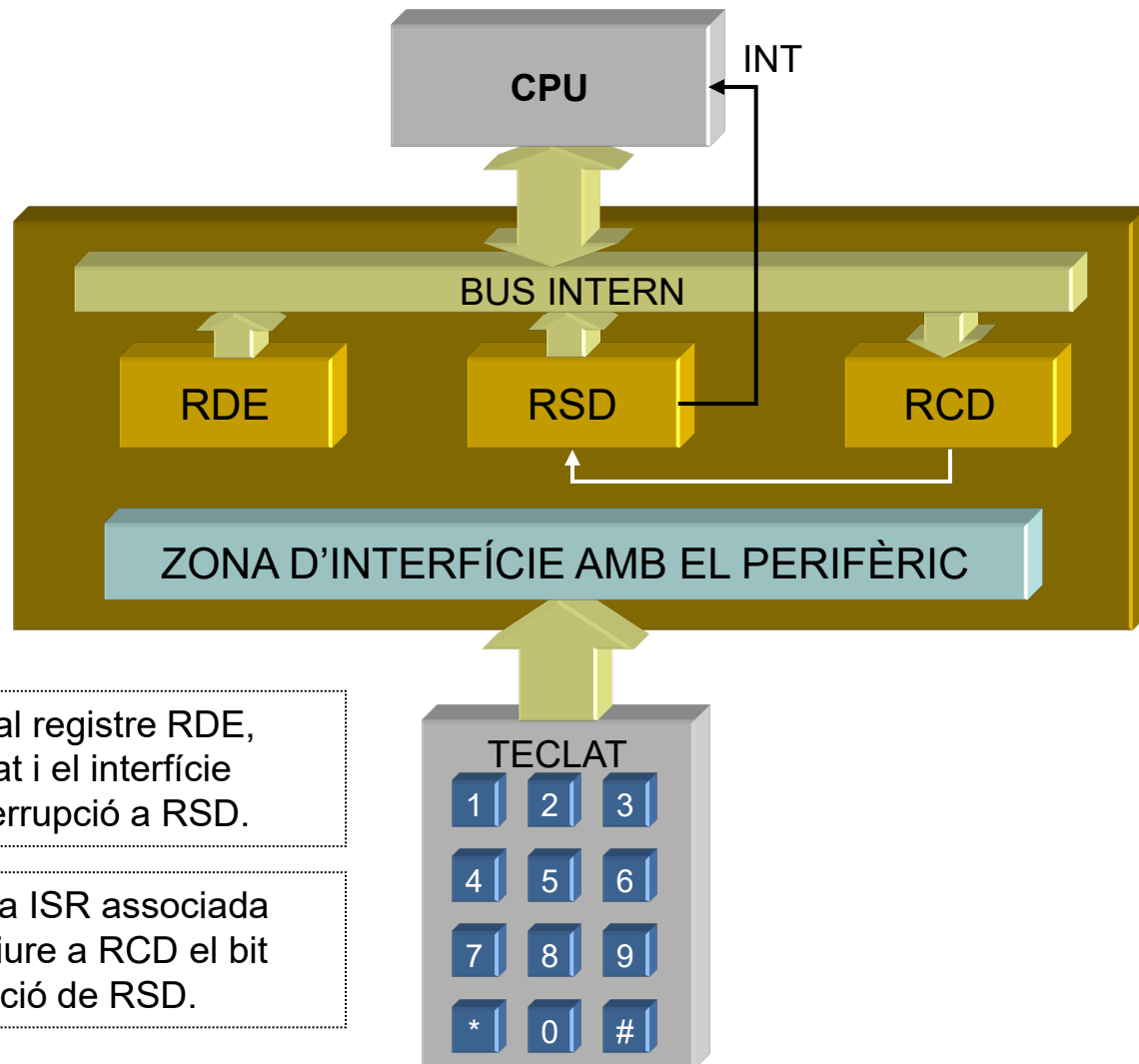
**RDE:** S'escriurà aquí el codi de tecla de al tecla que hem pitjat.

**RSD:** Serà un bit que s'activa quan pitgem una tecla i que servirà per demanar interrupció al processador

**RCD:** Només un bit, el farem servir per esborrar el de Status de petició d'interrupció.

Quan pitgem una tecla, el seu codi es fica al registre RDE, mitjançant la lògica d'interfície entre el teclat i el interfície d'E/S. A més, s'activa el bit de petició d'interrupció a RSD.

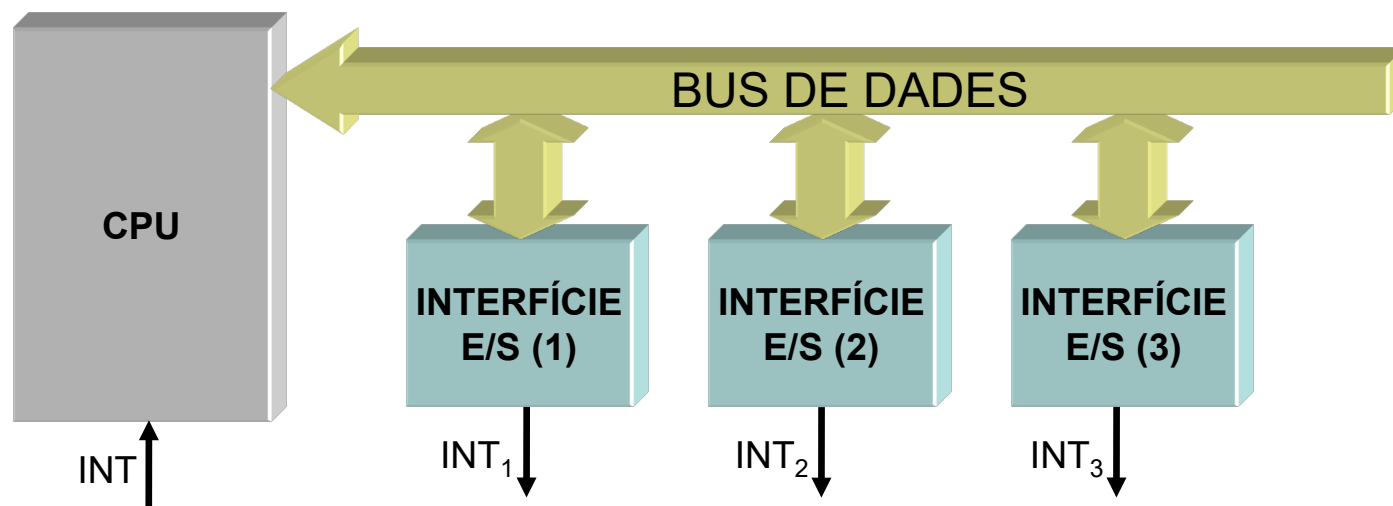
Si la CPU accepta l'interrupció, executarà la ISR associada que llegirà el codi de RDE. També a d'escriure a RCD el bit permet desactiva el bit de petició d'interrupció de RSD.

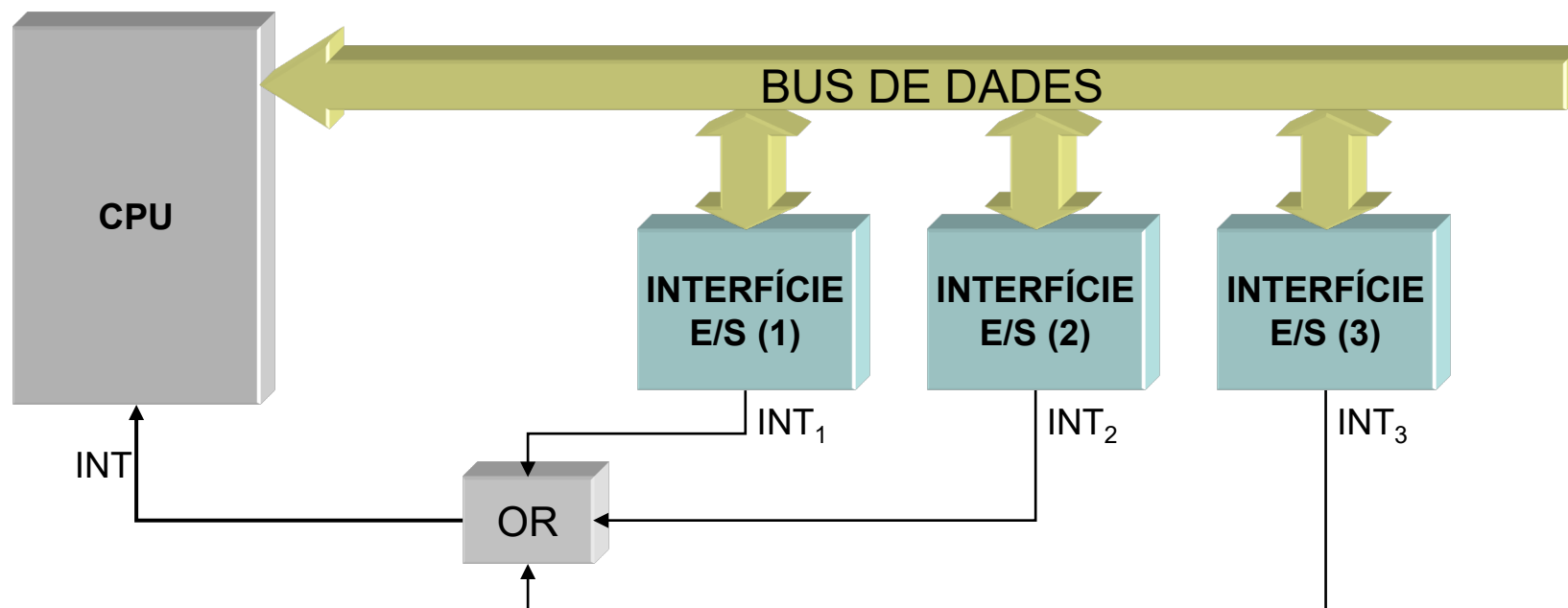


## Interrupcions Múltiples

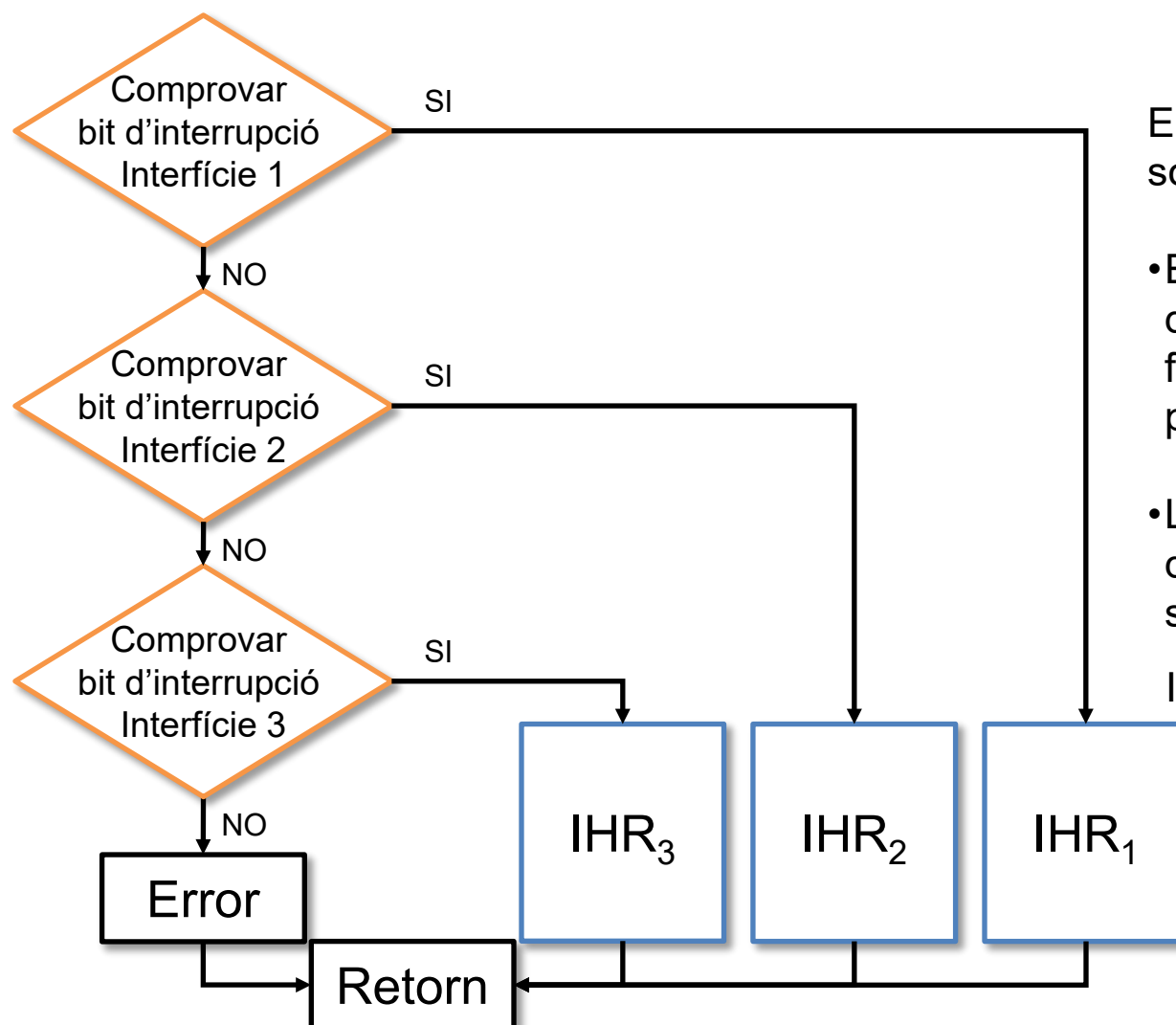
Una situació molt habitual serà que a un ordinador tinguem més d'una font d'interrupció. Sens plantegen llavors diversos “problemes”:

- Si el número d'entrades de petició d'interrupció del processador és menor que el de dispositius que poden demanar, tenim un problema de *hardware*. El processador ha de saber quin dispositiu li demana la interrupció per executar la  $IHR_i$  corresponent.
- Quan dos o més dispositius demanin interrupció, s'ha d'establir algun sistema de prioritats.



Interrupcions Múltiples: Reconeixement per Sondeig (*Polling*)

Mitjançant una porta OR fem que totes les sortides de petició d'interrupció dels diferents interfícies d'E/S dels perifèrics, vagin a l'entrada d'interrupció del processador. Un cop es genera una interrupció per part de qualsevol del interfícies, la ISR inicial ha de detectar quin ha estat el interfície que ha demanat la interrupció.

Interrupcions Múltiples: Reconeixement per Sondeig (*Polling*)

Els dos problemes que ens sorgien queden solucionats:

- El software imposa les prioritats de cada font d'interruptió. En funció de quina consultem primer.
- La font d'on prové la interrupció queda especificada pel mateix software.

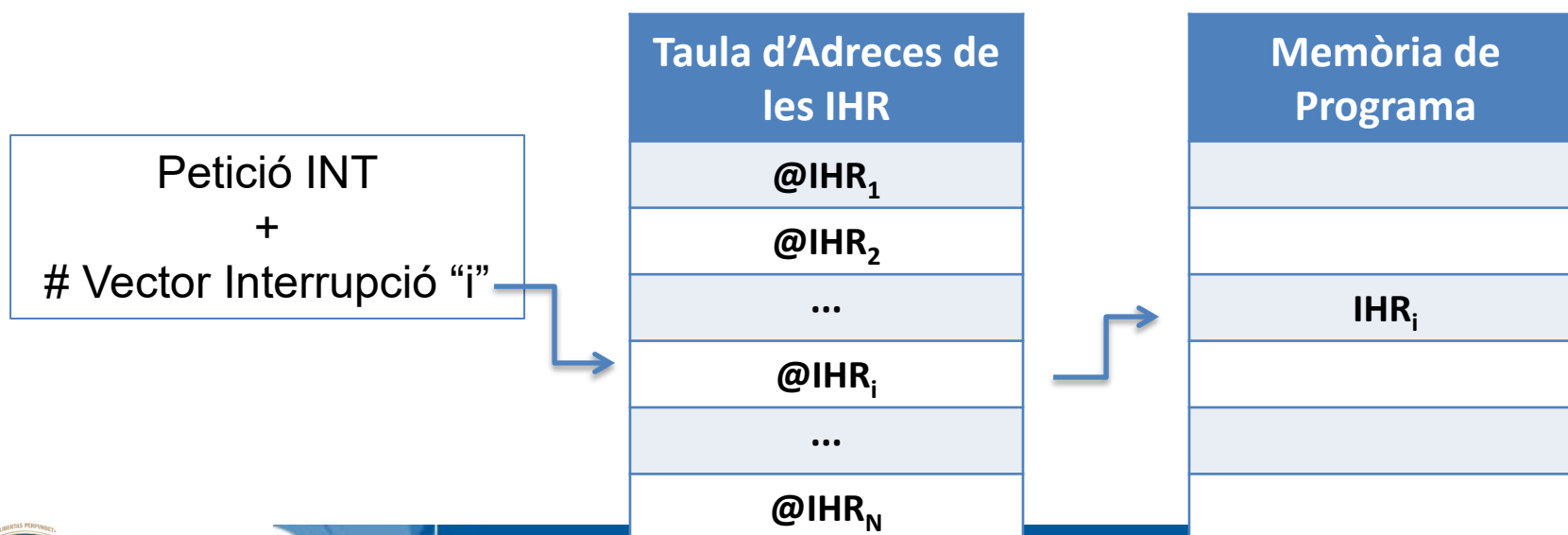
Inconvenient:

- Les interrupcions del final de la "cadena" triguen més en ser ateses.

## Interrupcions Múltiples: Solucions Hardware

Aquestes solucions es basen en que la identificació de la font d'interrupció, i moltes vegades les prioritats i emmascaraments, es fa per hardware.

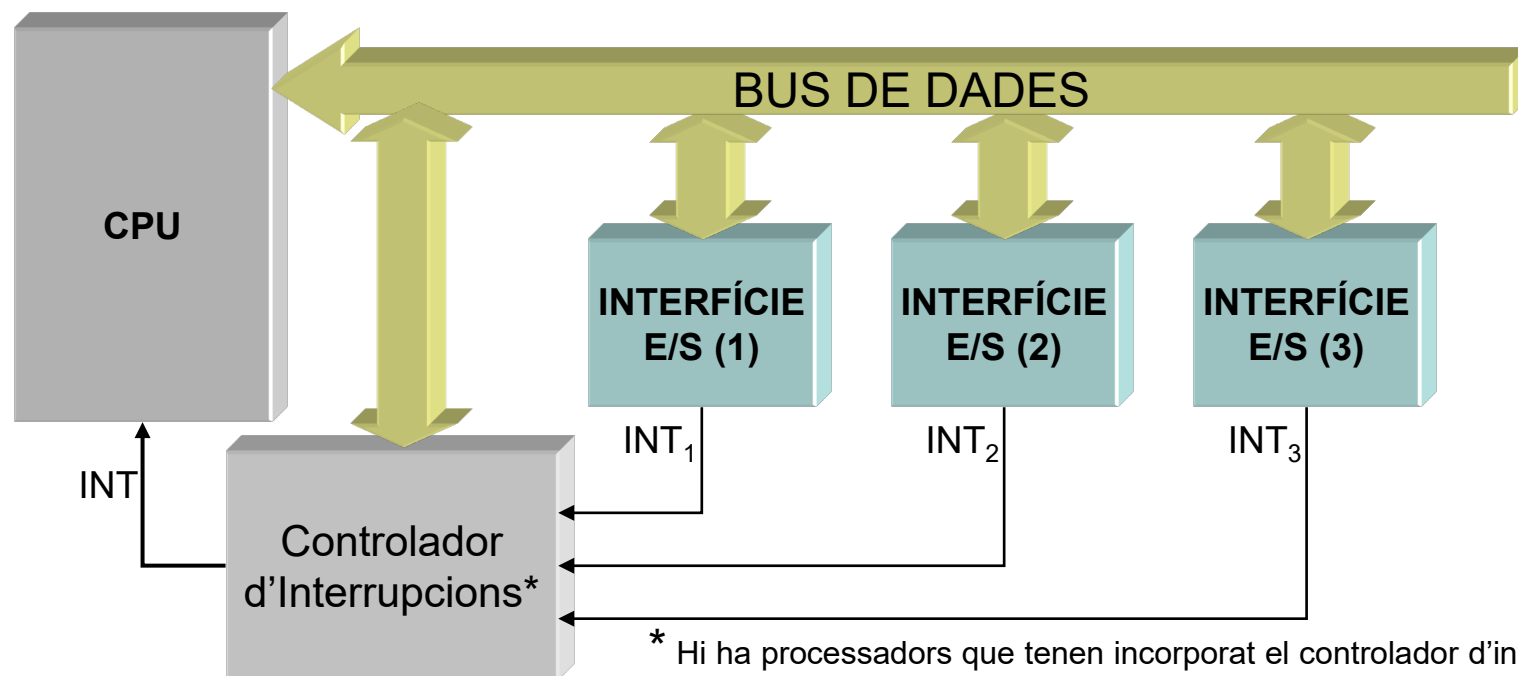
- Cada dispositiu que pot generar interrupcions té associat un identificador, anomenat Vector d'Interrupcions, que ha de proporcionar al processador quan li demana una interrupció.
- Cada vector d'interrupció té associada una posició a una taula on es guarden les adreces de les rutines d'atenció a les interrupcions ( $IHR_i$ ).



## Interrupcions Múltiples: Solucions Hardware

Es solen dividir en dos tipus:

- 1. Vectoritzades:** És la interfície, o un controlador d'interrupcions extern, el que proporciona el vector d'interrupció.
- 2. Autovectoritzades:** El processador té incorporat un controlador d'interrupcions que proporciona el vector d'interrupció.

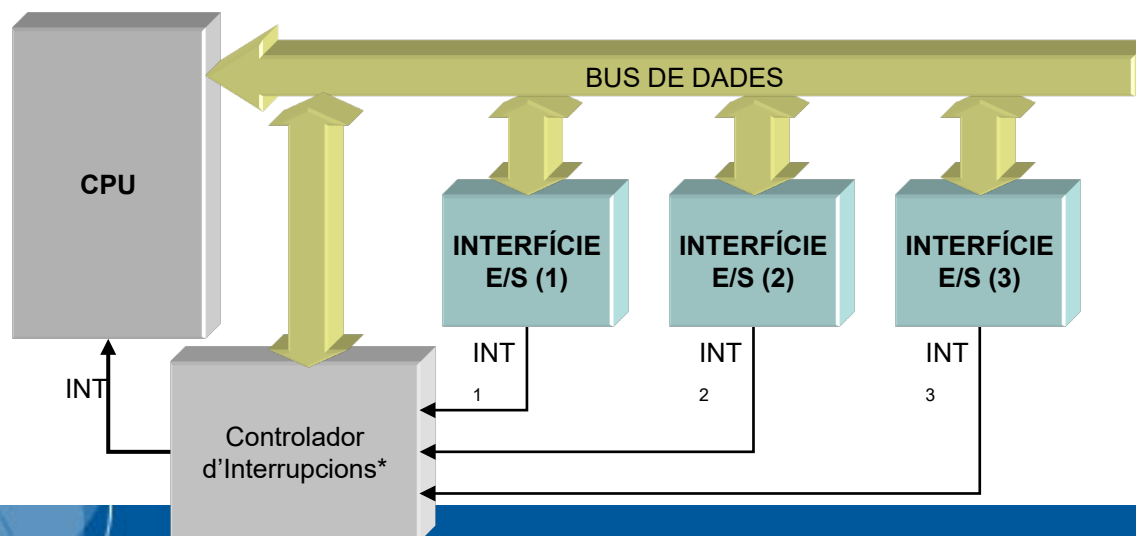


\* Hi ha processadors que tenen incorporat el controlador d'interrupcions. Processadors de Motorola, o els embeded (empotrats).

## Controlador d'Interrupcions

Les funcions dels sistemes que tenen Controlador d'Interrupcions són:

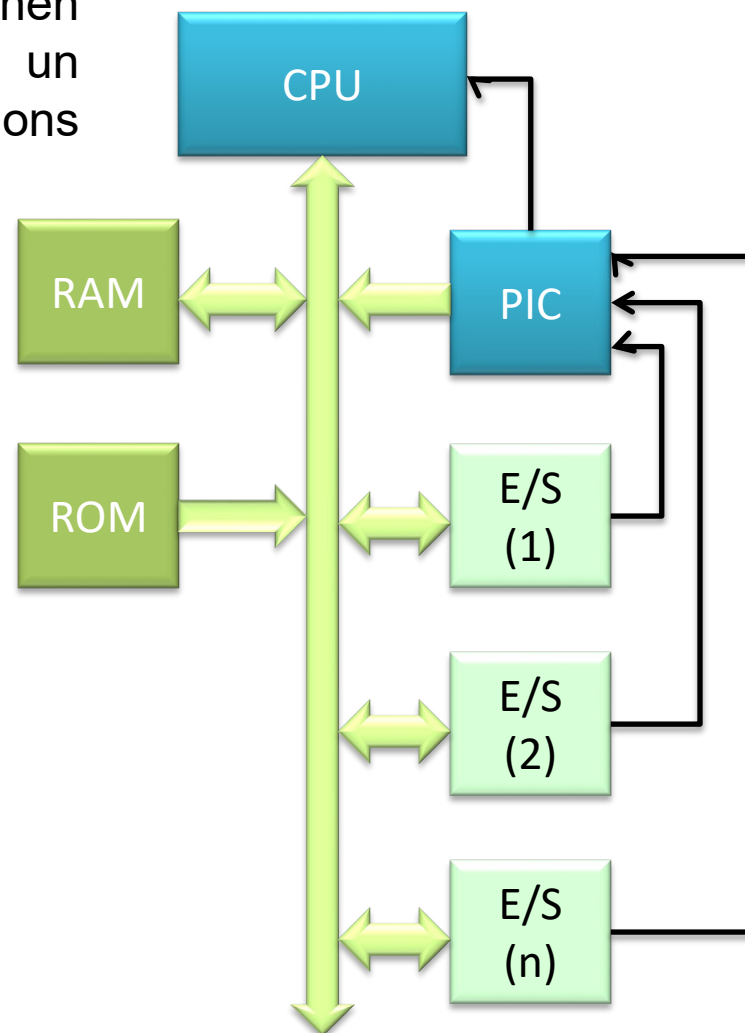
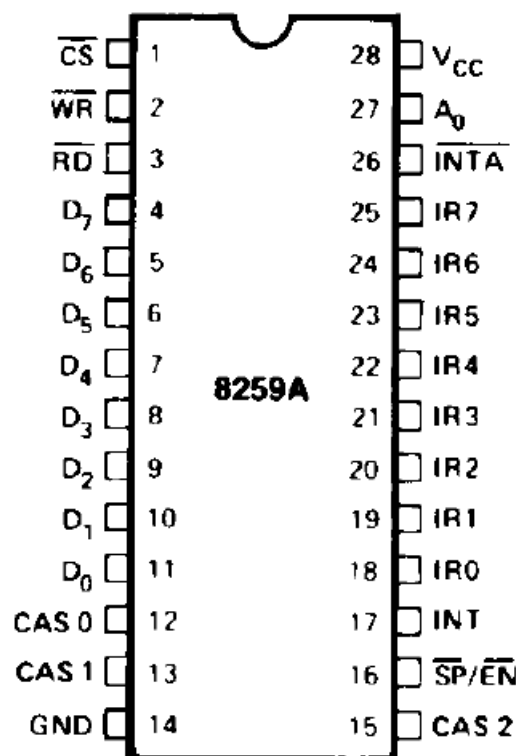
- **Generar el Vector d'Interrupció** per a cada font.
- **Gestionar les Prioritats** de cada font d'interrupció.
- Gestionar l'habilitació/Inhabilitació (Emmascarament) de cada font d'Interrupció.
- Si és extern al processador, tenen varies entrades d'interrupció i només una sortida que és la que activarà el senyal d'interrupció del processador.





## Exemple de Gestió d'Interrupcions (Intel 80x86) amb PIC8259A

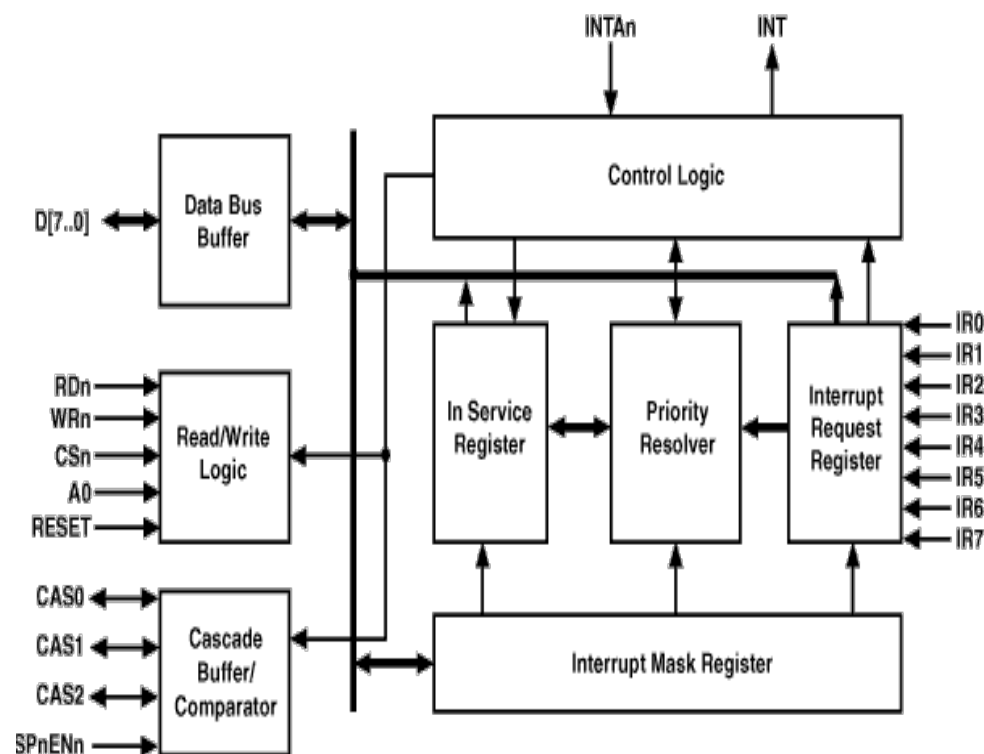
Aquests processadors de Intel, només tenen una entrada d'Interrupció (INT#). Necessiten un controlador extern per gestionar les interrupcions externes.



Peripheral Interrupt Controller (PIC8259) de Intel.

## Exemple de Gestió d'Interrupcions (Intel 80x86) amb PIC8259A

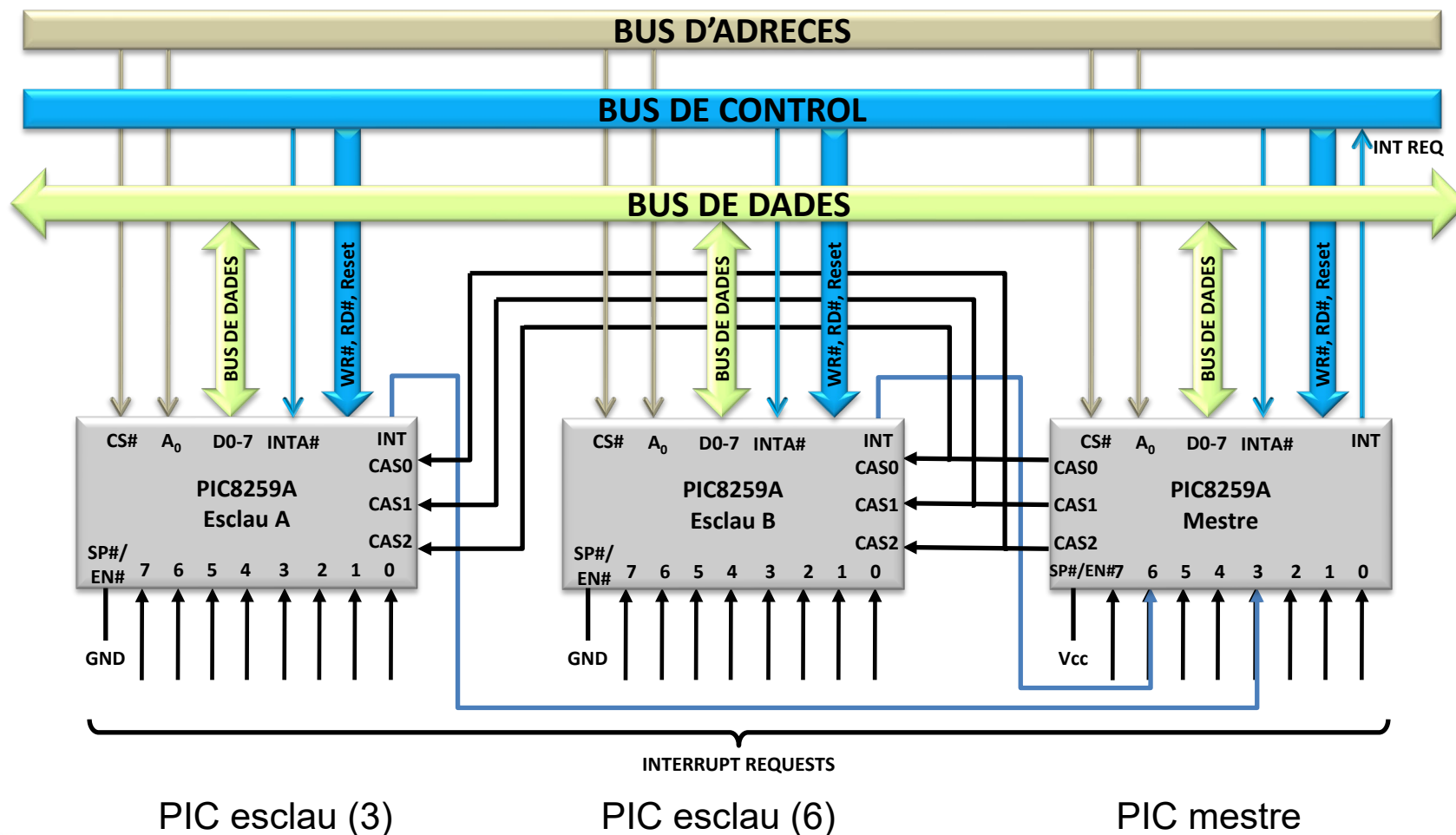
1. IRR: Registra les entrades que demanen interrupció.
2. Mitjançant el IMR i el "Priority Resolver" es determina si s'accepta la interrupció per ser demandada al processador.
3. Si és així es registra al ISR.
4. A més, s'activarà el senyal de petició d'interrupció INT de la CPU i començarà el procés entre el PIC i la CPU per gestionar l'atenció de la interrupció.



Les línies  $CAS_i$  i la  $SP_n/EN_n$  serveixen per connectar diversos PICs en mode cascada.

## Exemple de Gestió d'Interrupcions (Intel 80x86) amb PIC8259A

Connectant varis PIC en cascada podem ampliar el nombre d'entrades de petició d'interrupció. A l'exemple hem posat 3 PICs, el que ens permet tenir 22 entrades.



## Exemple de Gestió d'Interrupcions (Intel 80x86) amb PIC8259A

1. Un o més dispositius demanen interrupció
2. S'activen les entrades IRR del PIC mestre a partir dels esclaus (o del dispositiu)
3. El PIC mestre activa INT la CPU
4. La CPU activa INTA per indicar que accepta la interrupció
5. El PIC mestre indica a CAS0-2 el codi de l'esclau al que li reconeix la petició
6. La CPU envia el segon INTA
7. El PIC esclau/mestre posa al bus de dades el vector d'interrupció
8. La CPU llegeix el vector d'interrupció i el converteix a vector d'adreça (x4)
9. La CPU executa la IHR....

## DMA (Direct Memory Access): Introducció

Suposem que volem transferir un bloc de dades de Memòria a un perifèric. Per exemple per imprimir un document, desar-les al disc dur... Els processadors acostumen a tenir instruccions que faciliten aquestes tasques. Veiem l'exemple d'un Motorola 68000:

INSTRUCCIÓ	QUE FA...	Cicles de rellotge
<b>MOVE #N,D0</b>	Nombre de dades a transferir a D0.	Aquestes són d'inicialització, i només s'executen una vegada. Per tant, no és molt important el que triguin
<b>MOVE #Buffer,A0</b>	Adreça d'on es troben les dades a memòria a A0.	
<b>MOVE #Port,A1</b>	Adreça del port del perifèric.	
<b>Bucle:</b>		
<b>MOVE.W (A0)+,A1</b>	Transfereix una dada al port i apunta a la següent.	<b>8 Tc</b>
<b>DBF D0,Bucle</b>	Decrementa D0 i salta a bucle si no és = a 0.	<b>10 Tc</b>

Si comptem el cicles de rellotge de cada bucle (que transfereix una paraula) tenim que són 18 cicles de rellotge, que haurem de multiplicar per N paraules que volem transferir per calcular el temps total.

Les tècniques d'Accés Directe a Memòria (DMA) permeten transferir una paraula en 2 cicles, o fins i tot en només 1 cicle, de rellotge.

## DMA (Direct Memory Access): Introducció

Per fer transferències DMA es necessita un hardware específic, en concret un Controlador d'Accés Directe a Memòria (CDMA).

- El CDMA és un dispositiu Mestre (Master) i per tant té capacitat per gestionar els busos del sistema, direccionant tant la memòria principal com els interfícies d'E/S.
- Abans de qualsevol transferència DMA el processador ha d'inicialitzar el CDMA, de forma semblant a la primera part del programa anterior:
  - ✓ Nombre de dades a transferir.
  - ✓ Origen de les dades (Bloc de Memòria o Perifèric).
  - ✓ Destí de les dades (Zona de Memòria o Perifèric).
- Un cop inicialitzat el CDMA, aquest pot demanar el començament de les transferències per exemple demanant una interrupció al processador quan el perifèric estigui llest o li demani.
- A partir d'aquí, s'ha de posar d'acord amb el processador per a compartir els busos.

## Protocols de Gestió de Bus durant les Transferències DMA

Degut a que tant el Processador com el Controlador de DMA han de fer us dels busos, s'han d'establir protocols per gestionar l'ús d'aquests busos entre ells. Hi ha diferents protocols:

- 1. ATURADA DEL PROCESSADOR.**
- 2. ROBAMENT DE CICLE.**
- 3. BUS COMPARTIT.**

De fet, aquest potser un altre paràmetre que ha de programar el processador quan inicialitzar el CDMA per fer les transferències (a més del número de dades, origen i destí).

## Protocol DMA: ATURADA DEL PROCESSADOR



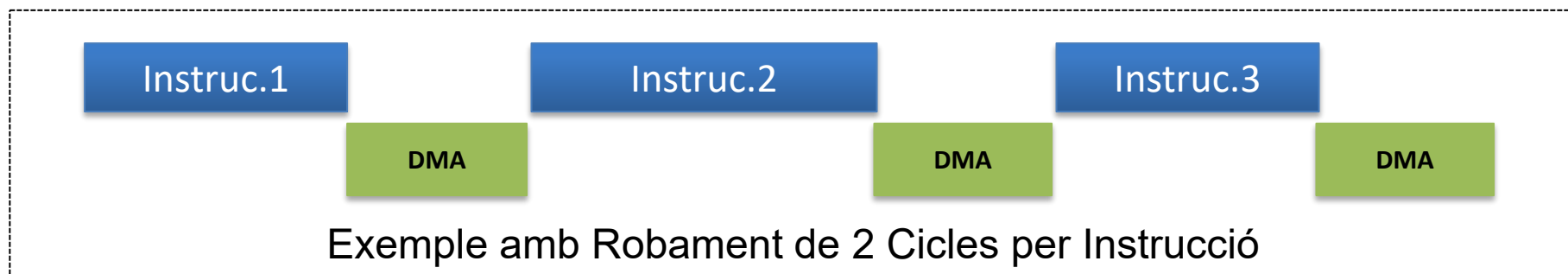
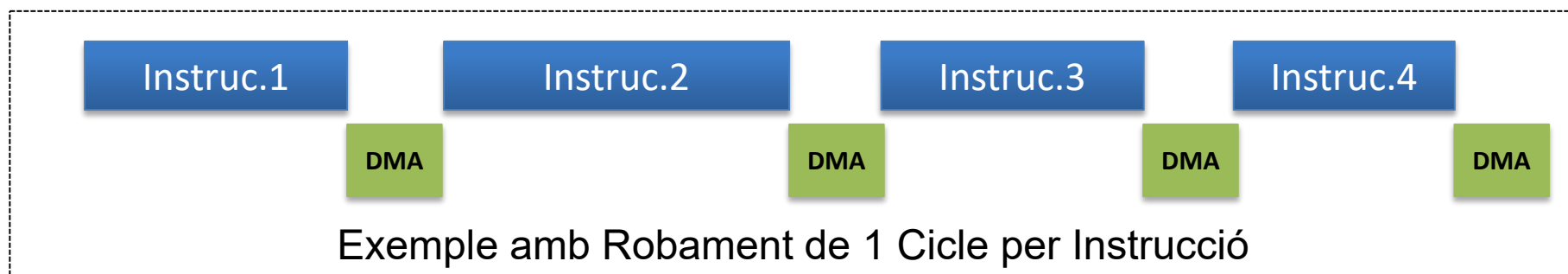
El mètode d'**Aturada del Processador** és el més senzill i habitual, consisteix en que durant les transferències de DMA el processador para la seva activitat i deixa que el Controlador de DMA gestioni els busos durant tota la transferència.

**Avantatge:** Les transferències DMA es fan el més ràpid possible.

**Inconvenient:** El processador para la seva activitat.



## Protocol DMA: ROBAMENT DE CICLE



**Avantatge:** Es continua executant el programa i es fan les transferències.

**Inconvenient:** No fem la transferència del DMA a màxima velocitat, ni tampoc l'execució del programa.

## Protocol DMA: BUS COMPARTIT



Mètode de Bus Compartit

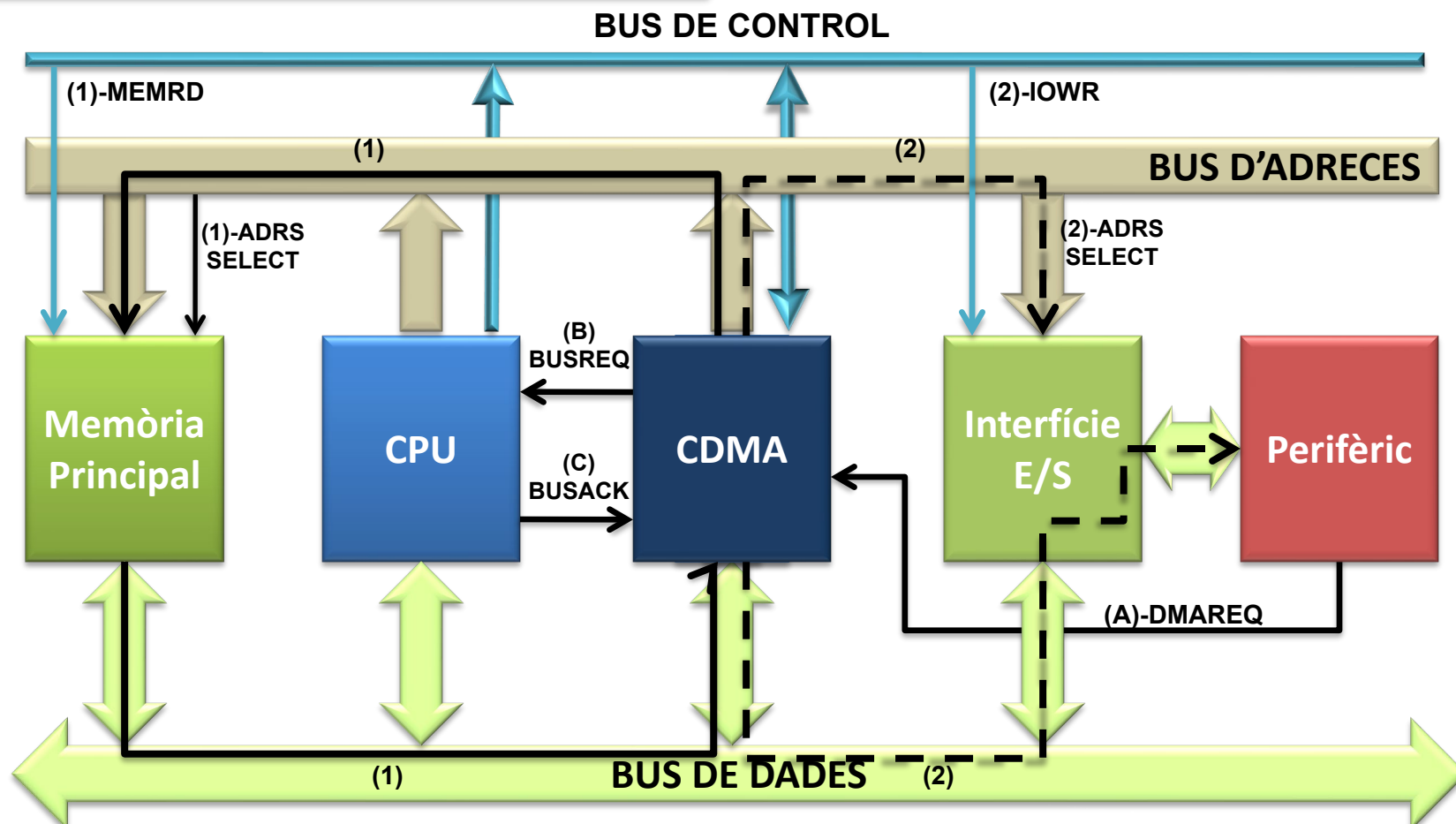
A aquest mètode, el processador no para l'execució del programa. Durant l'execució de cada instrucció poden haver cicles de rellotge en que no es faci servir els busos, aquests cicles són els que fa servir el DMA per fer les transferències.

**Avantatge:** No es perd temps d'execució de programa.

**Inconvenient:** No fem les transferències DMA a màxima velocitat.

## Transferències DMA en 2 Cicles

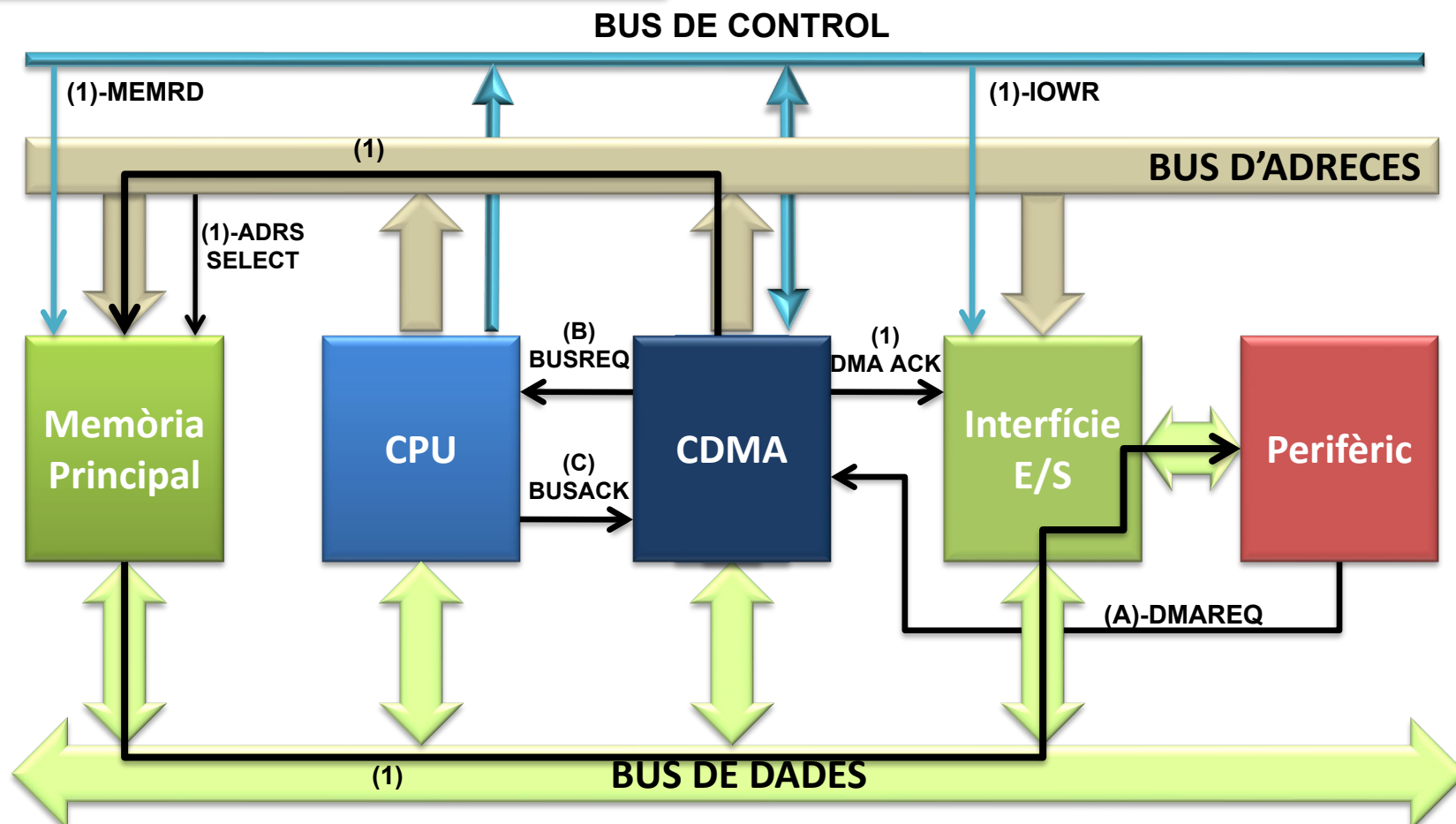
Ex.: Transf. bloc de memòria a perifèric.



- Comencen les transferències quan el perifèric demana al CDMA que li “faci cas” (A), aquest demana al processador el control dels busos (B) i aquest li cedeix (C).
- Per a cada dada que es transfereix, Cicle (1) el CDMA adreça de la memòria per llegir una dada que es guarda al CDMA; Cicle (2) el CDMA adreça el perifèric i li envia la dada.

## Transferències DMA en 1 Cicle

Ex.: Transferència bloc de memòria a perifèric.



- Comencen les transferències quan el perifèric demana al CDMA que li "faci cas" (A), aquest demana al processador el control dels busos (B) i aquest li cedeix (C).
- Per a cada dada que es transfereix, el CDMA adreça la memòria per llegir una dada que va directament al perifèric a través del seu CANAL de DMA (DMA ACK).

## Comparació DMA en 1 Cicle i 2 Cicles

Tipus DMA	Avantatges	Inconvenients
1 Cicle	Només triga un cicle de rellotge de bus en fer una transferència.	<ul style="list-style-type: none"> <li>Només serveix quan l'origen o destí és un perifèric. No permet transferències entre zones de memòria.</li> <li>Es complica una miqueta més el hardware ja que s'han de gestionar canals de DMA.*</li> </ul>
2 Cicles	Pot fer Transferències de una zona de memòria a un altre zona de memòria.	Necessita 2 cicles de rellotge de bus per fer una transferència.

\* Els canals de DMA sorgeixen perquè la transferència de cada dada es fa només en un cicle. Hem de definir un origen i un destí, el bus d'adreces només pot especificar un d'ells, el que fa referència a la posició de memòria. Per activar el interfície d'E/S del perifèric definim uns senyals específics que gestiona el CDMA i que els activa directament. Qualsevol perifèric que pugui treballar amb DMA ha de tenir un canal DMA connectat al CDMA.