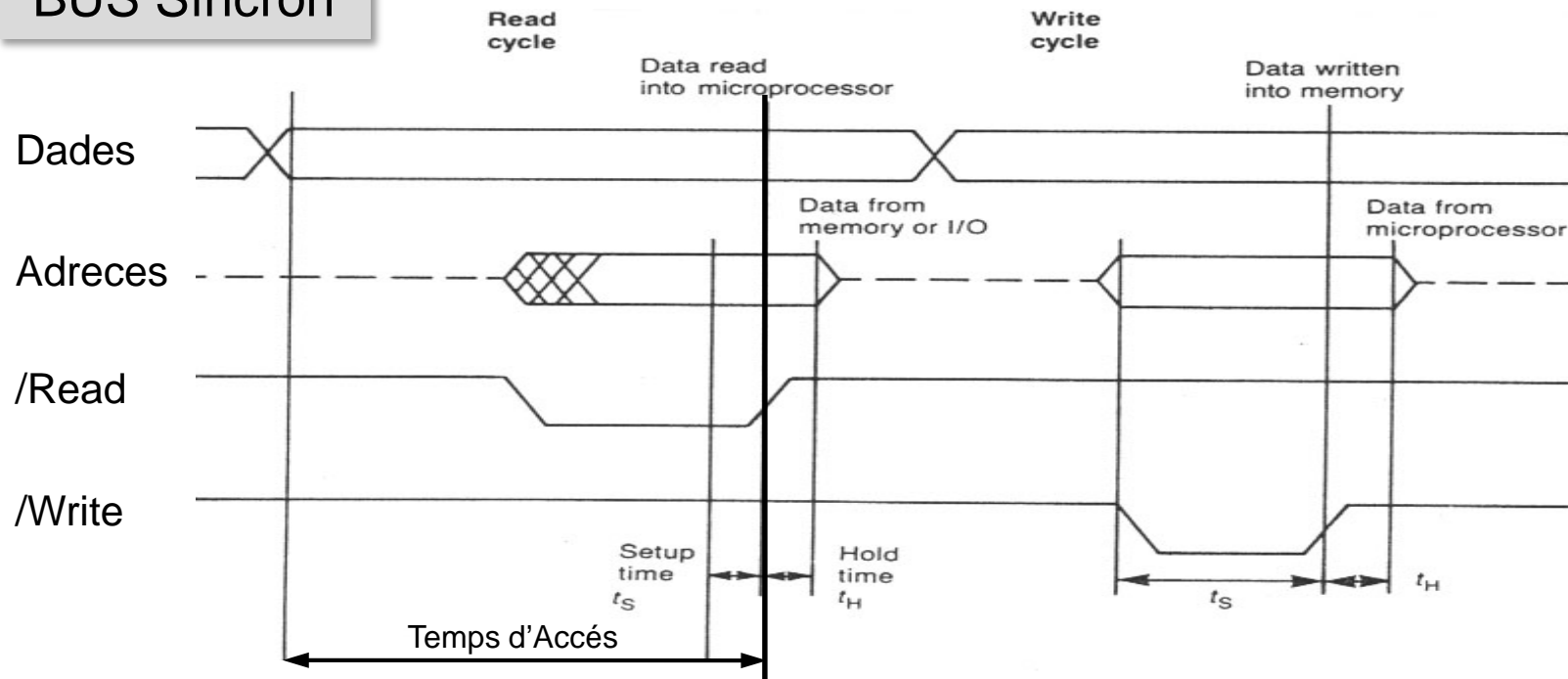


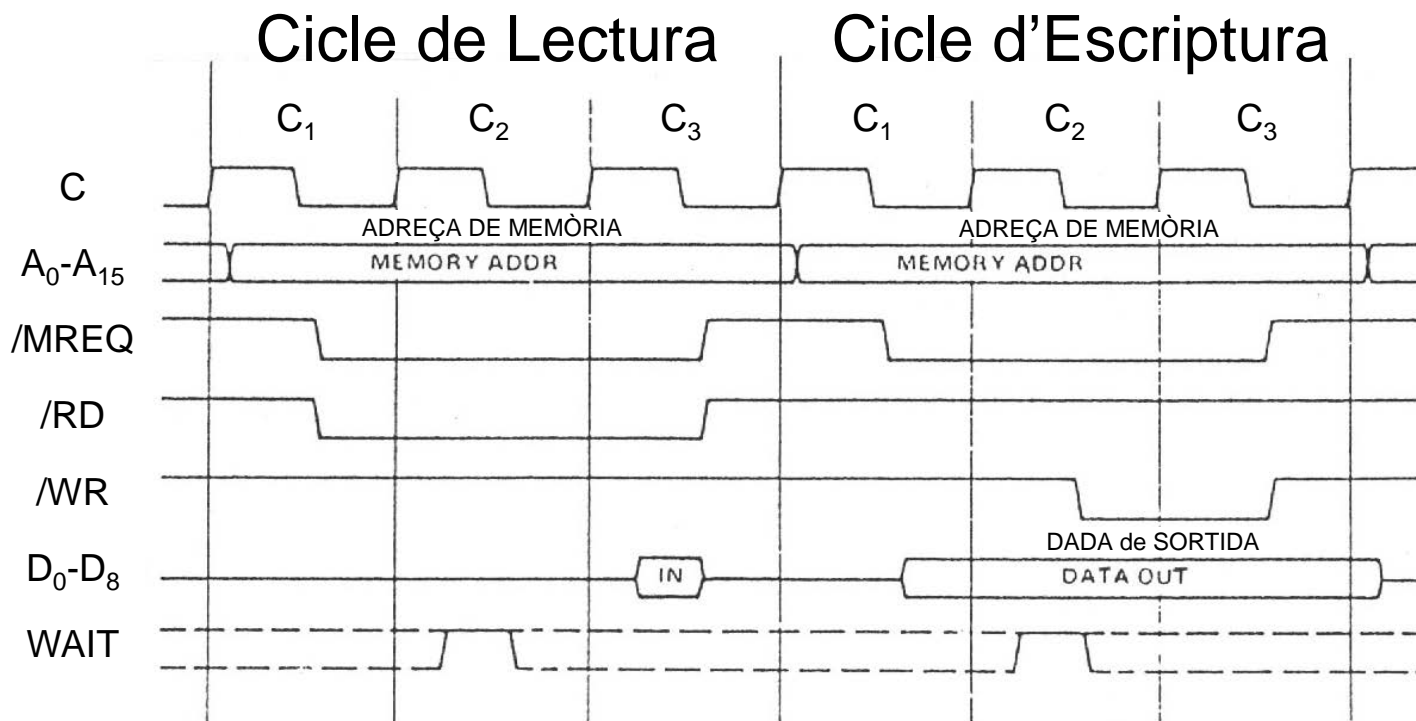
# BUSos

## BUS Síncron



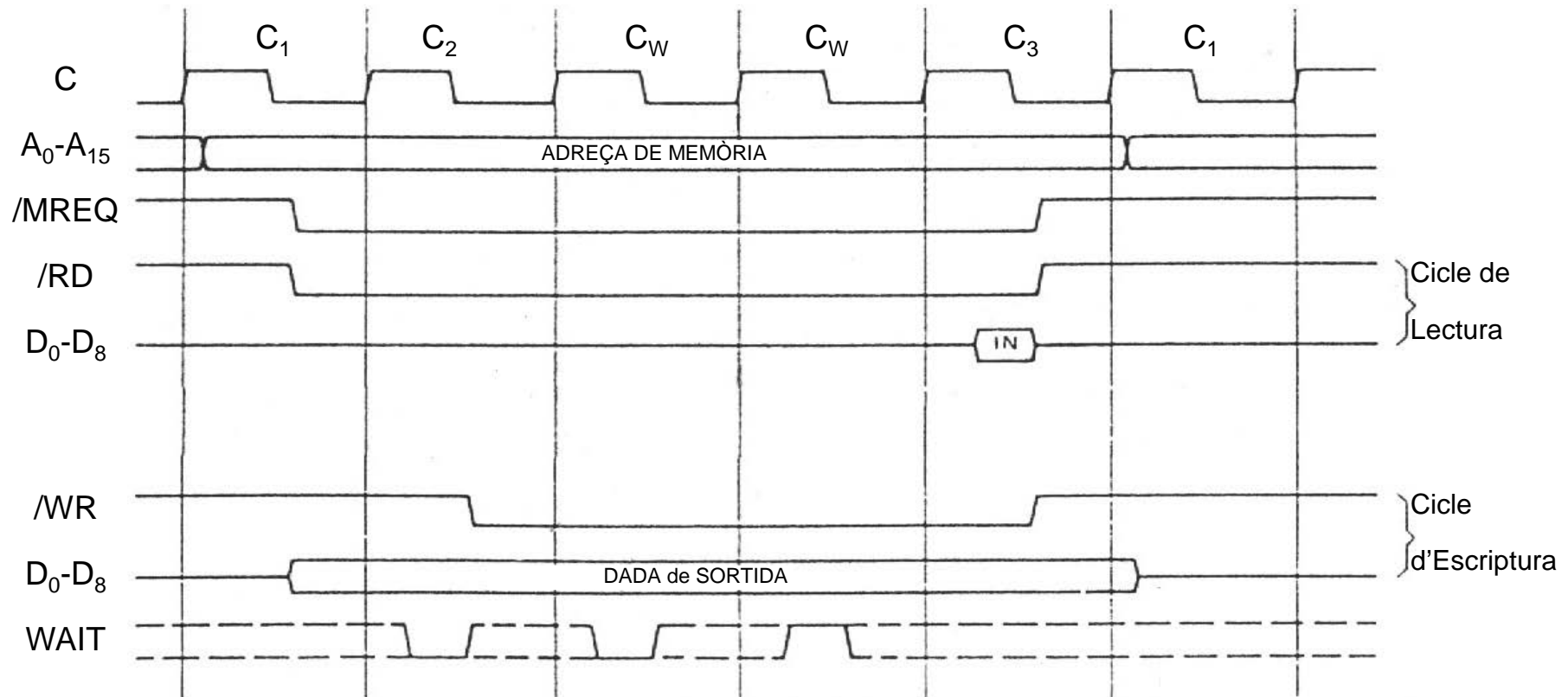
- Temps de SETUP ( $t_S$ ): temps mínim que ha de romandre el senyal estable abans de ser admès).
- Temps de Manteniment (HOLD  $t_H$ ): temps mínim que ha de romandre el senyal estable després de ser admès.
- Temps d'Accés: temps des de que és comença el la petició/enviament de dada i el moment en que es llegeix/recull la dada.
- Cicle de Bus: Nombre mínim de cicles de rellotge necessaris per fer una transferència.

## Exemple de BUS Síncron



- En aquest cas, el Cicle de Bus és de 3 cicles de rellotge (3C).
  - Cicle de Lectura.
  - Cicle d'Escriptura.
- } En aquest cas cicle de lectura=cicle d'escriptura
- Cicle de Bus vindria donat pel temps necessari per a comunicar-se amb el dispositiu més lent.

## Exemple de BUS Síncron



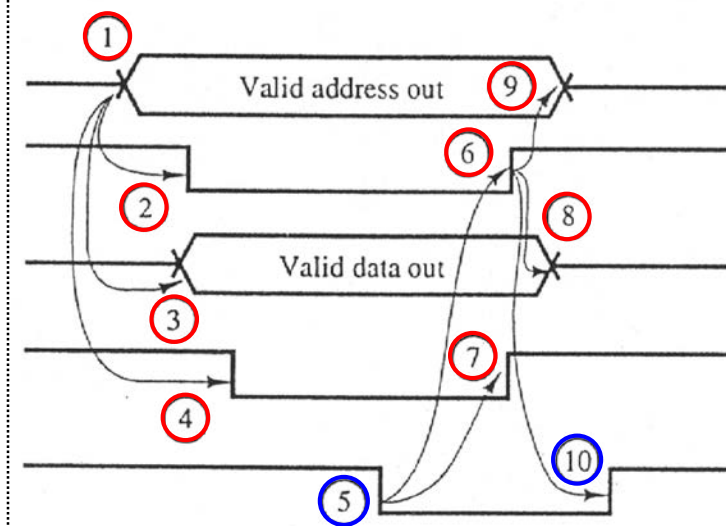
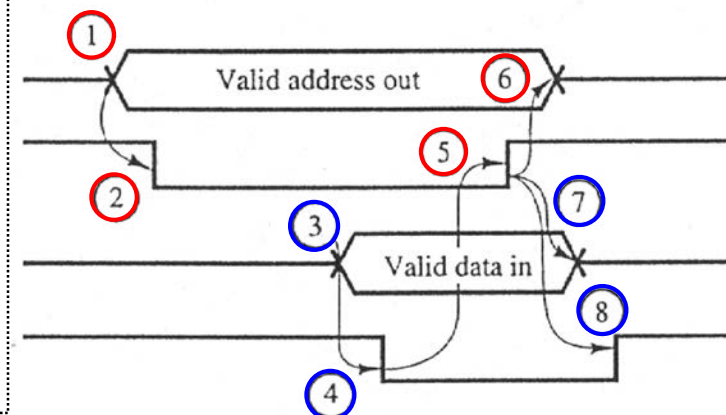
### Exemple de Bus Síncron amb cicles d'espera

- Per evitar pèrdues innecessàries de temps s'implementen protocols de cicles d'espera (wait states)
- Els dispositius lents introdueixen senyals per obligar a que el cicle de bus duri més.

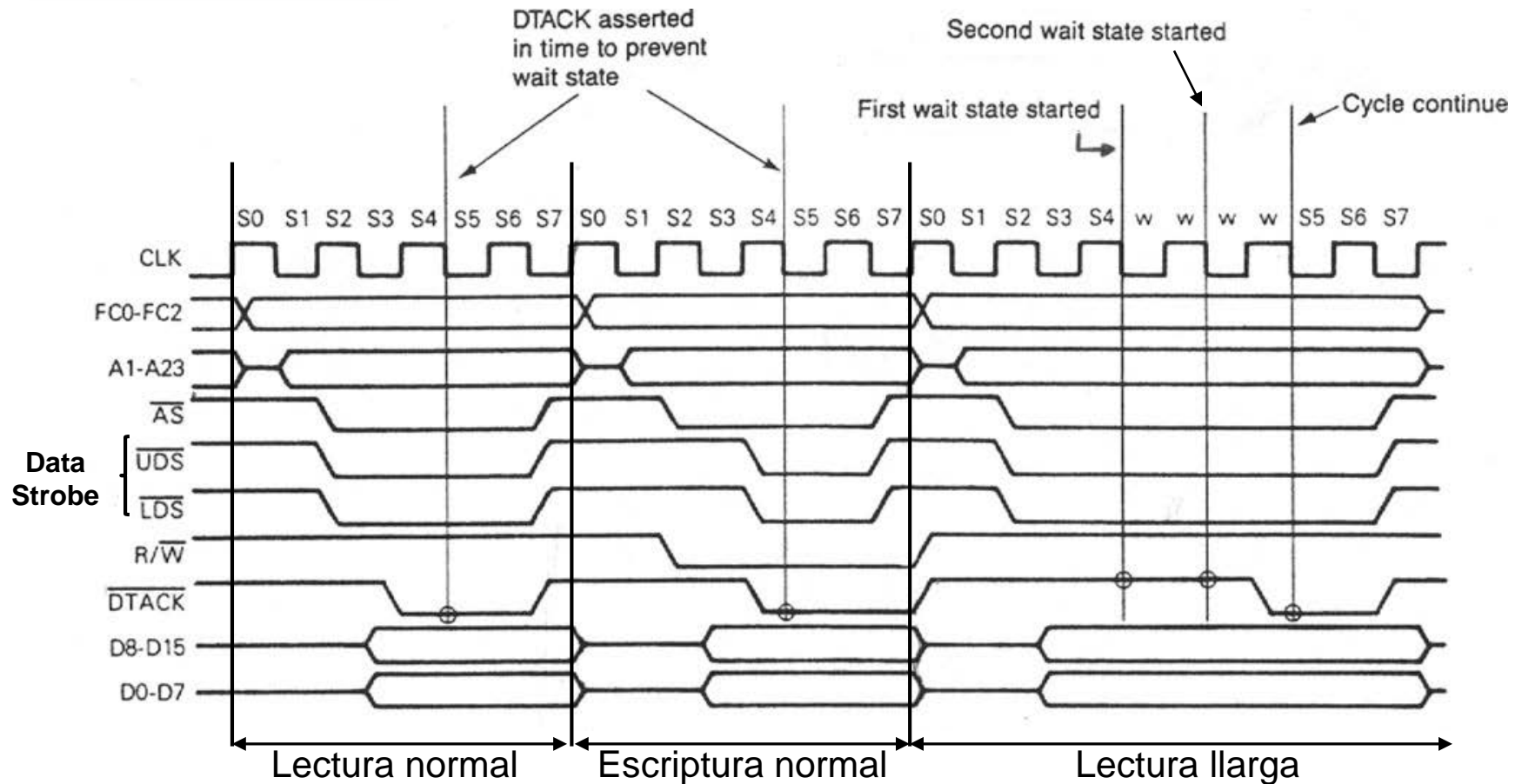
## BUS Asíncron

- El Mestre posa l'adreça al bus.
- El Mestre indica que ha posat l'adreça (**Adress Strobe**).
- L'Esclau posa la dada al bus.
- L'Esclau activa l'**Acknowledge**.
- El Mestre desactiva l'**Adress Strobe**.
- El Mestre ja no treu les adreces.
- L'Esclau ja no treu les dades.
- L'Esclau desactiva l'**Acknowledge**.

- El Mestre posa l'adreça al bus.
- El Mestre indica que ha posat l'adreça (**Adress Strobe**).
- El Mestre posa la dada al bus.
- El Mestre indica que ha posat les dades (**Data Strobe**).
- L'Esclau activa l'**Acknowledge**.
- El Mestre desactiva l'**Adress Strobe**.
- El Mestre desactiva el **Data Strobe**.
- El Mestre deixa de treure les dades.
- El Mestre deixa de treure les adreces.
- L'Esclau desactiva l'**Acknowledge**.

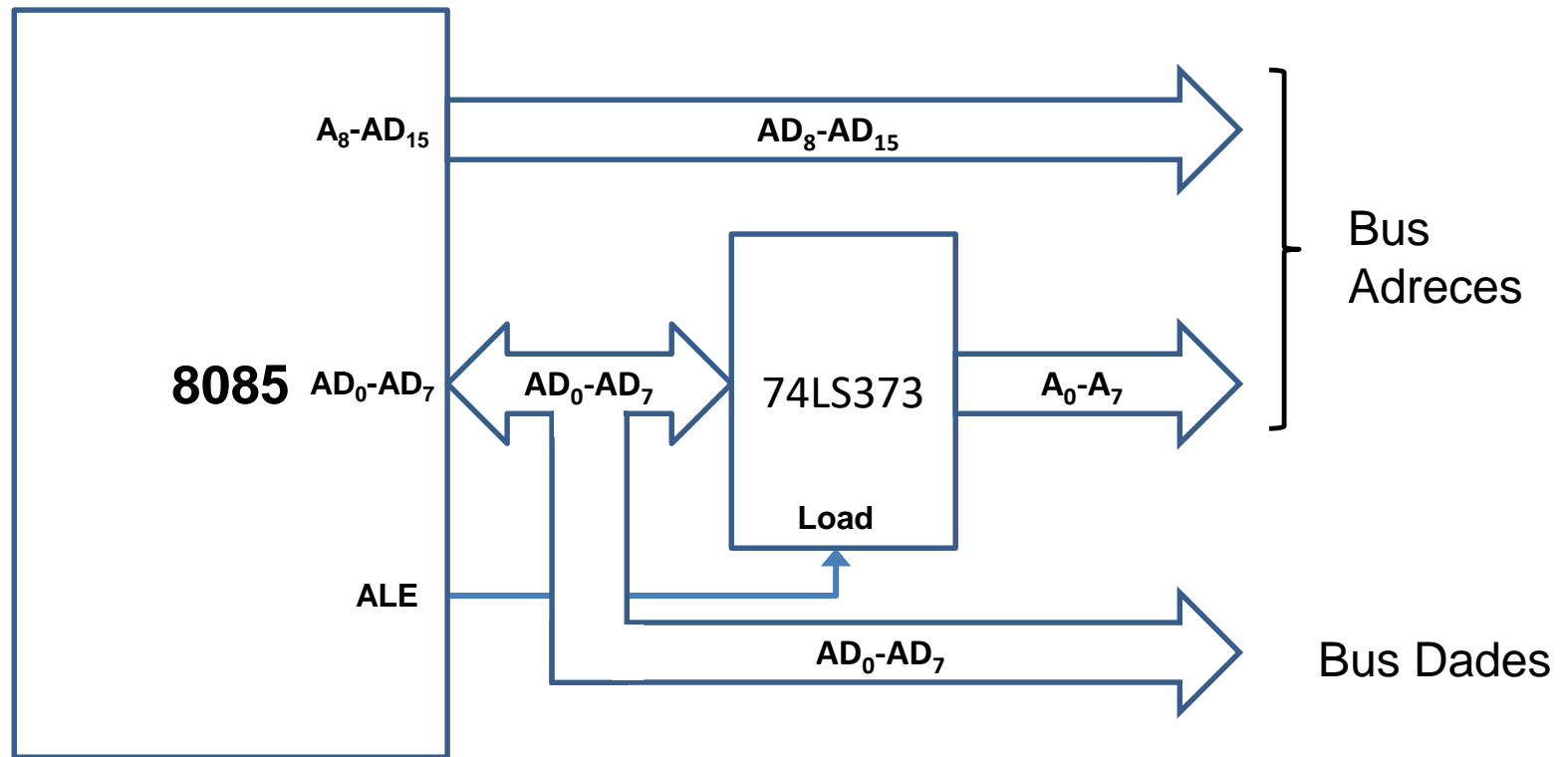


## BUS Asíncron



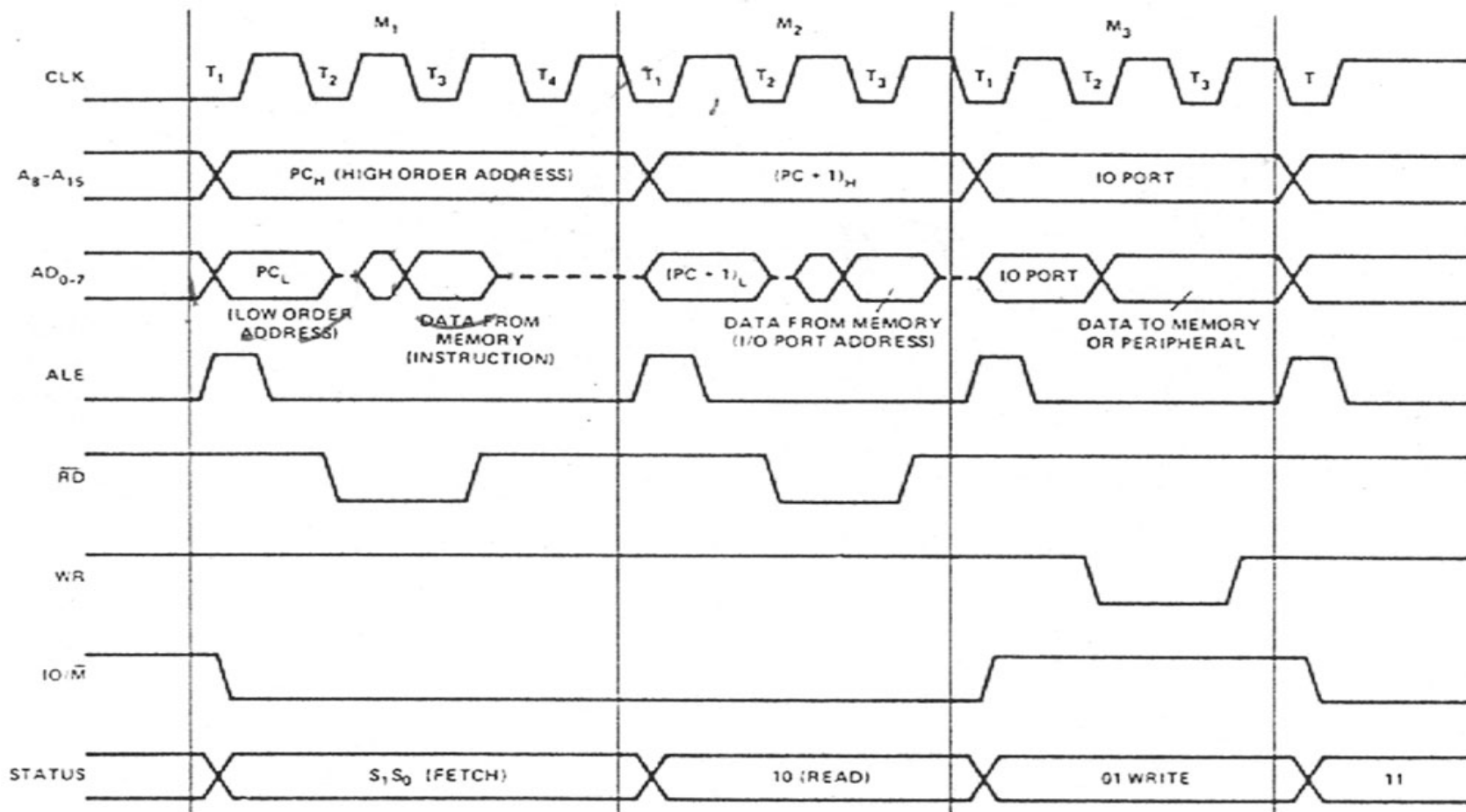
## Exemple de Bus Asícron amb cicles d'espera

# BUS Multiplexat



Exemple de Bus Multiplexat (8085)

# BUS Multiplexat

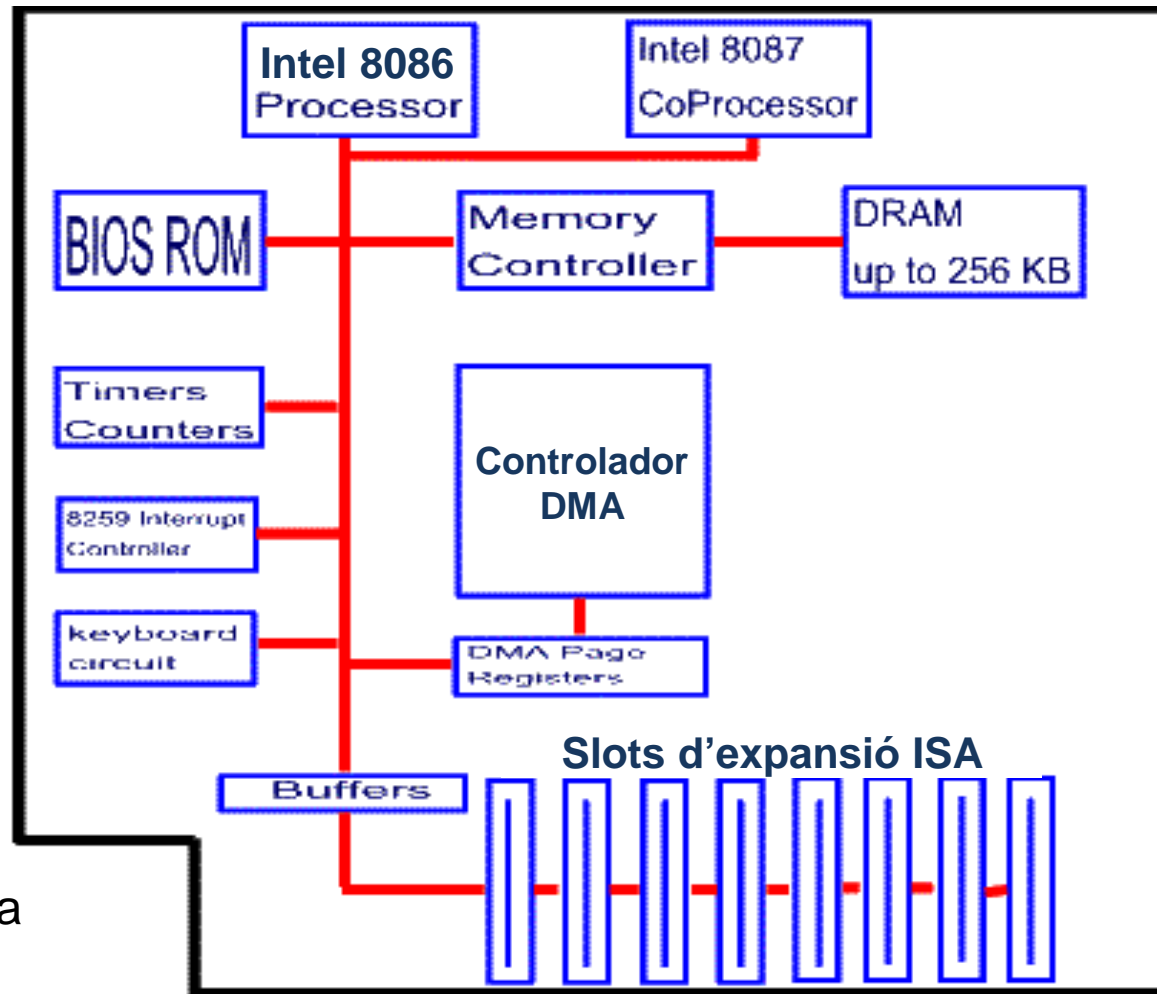


Exemple de Bus Multiplexat (8085)



# Exemple de sistema de BUS dels primers PCs

ISA: Industry Standard Architecture

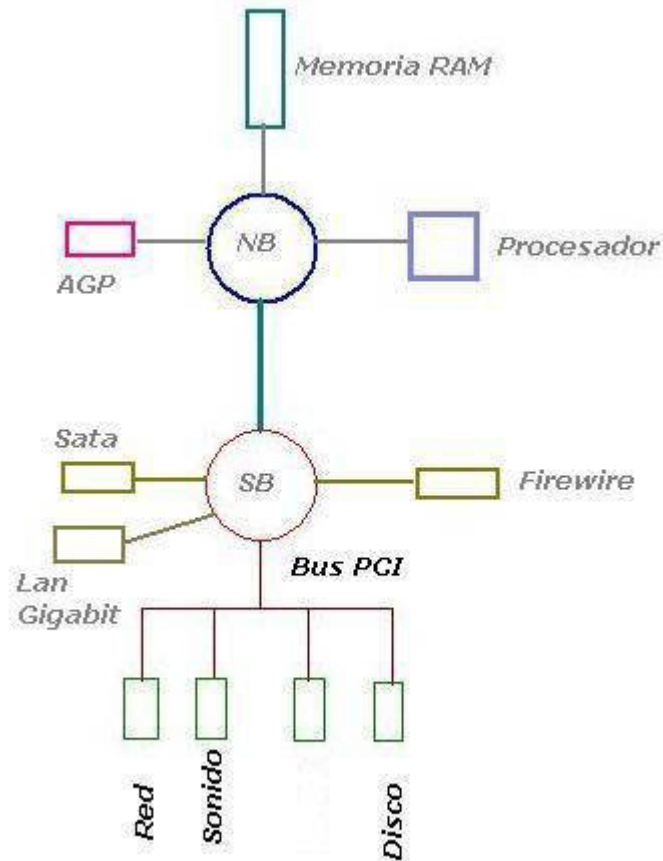


Arquitectura  
d'un PC

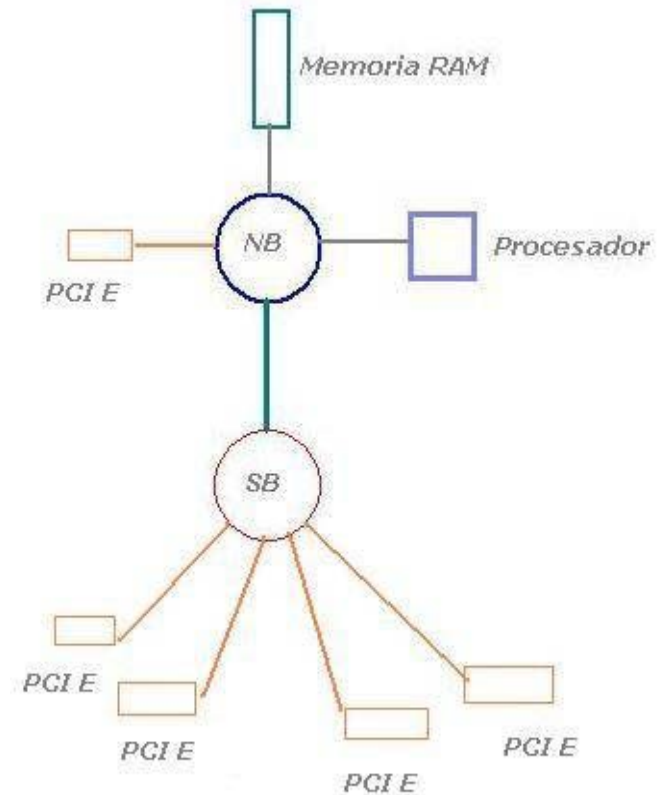
# Característiques principals de diferents “BUSos” “Estàndard”

BUS	Any	Amplada	Tipus	Velocitat	Ample Banda màxim
ISA	1982	8 bits	Síncron	4.77MHz	2MB/s
ISA (AT)	1984	16 bits	Síncron	8MHz	8MB/s
EISA	1988	32 bits	Síncron	8MHz	32MB/s
PCI 1.0	1993	32 bits	Asíncron/Multiplexat	33MHz	133MB/s
PCI 2.2	1998?	64 bits	Asíncron/Multiplexat	66MHz	533MB/s
AGP* 1x	1997	32 bits		66MHz	266MB/s
AGP* 8x	2002	32 bits		533MHz	2.13GB/s
PCI-Express 1x - x16	2004	Sèrie (Canals)	Síncron	133MHz-2128MHz	200MB/s-6.4GB/s (8GB/s)

# PCI Express: Arquitectura



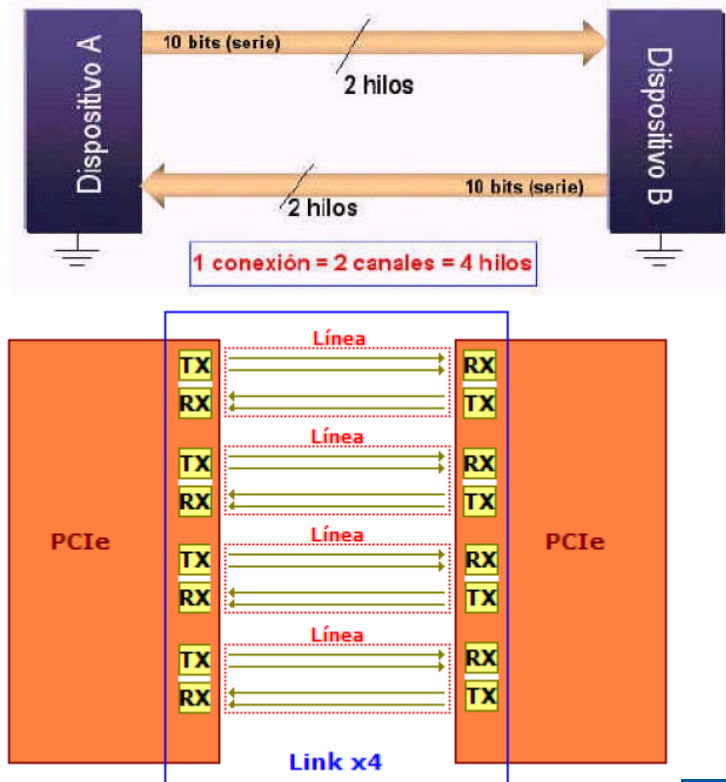
Arquitectura en BUS del PCI



Arquitectura punt a punt  
(en estrella) del PCIe.

## PCI Express: Capa Física

- El bus PCIe està estructurat com enllaços sèrie punt a punt, cada enllaç és *full duplex*.
- Un canal PCI Express ofereix inicialment una velocitat de 2,5 Gbits/s en cada sentit.
- Cada ruta utilitza dos parells de fils, transmissió i recepció, fent d'aquesta manera un rendiment efectiu de 200MBytes/s en cada direcció.
- Cada slot d'expansió porta 1, 2, 4, 8, 16 o 32 enllaços de dades. 32 enllaços donen el màxim d'ample de banda, 8 GB/s en cada sentit. El nombre d'enllaços s'escriu amb el prefix "x".



### PCI Express Example Connectors

**x1**

**BANDWIDTH**  
Single direction: 2.5 Gbps/200 MBps  
Dual Directions: 5 Gbps/400 MBps



**x4**

**BANDWIDTH**  
Single direction: 10 Gbps/800 MBps  
Dual Directions: 20 Gbps/1.6 GBps



**x8**

**BANDWIDTH** Single direction: 20 Gbps/1.6 GBps  
Dual Directions: 40 Gbps/3.2 GBps



**x16**

**BANDWIDTH** Single direction: 40 Gbps/3.2 GBps  
Dual Directions: 80 Gbps/6.4 GBps

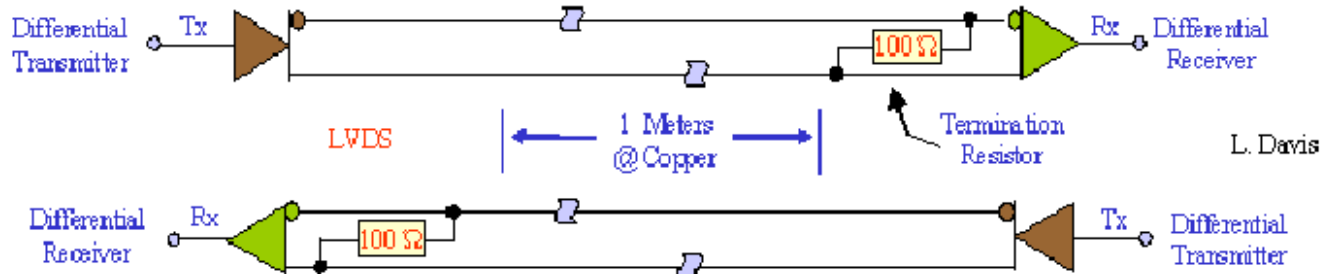


Source: IBM

©2005 HowStuffWorks

## PCI Express: Capa Física

- A nivell elèctric, cada enllaç consisteix en dos connectors LVDS EIA-644 (Senyals diferencials de baix voltatge) amb emissor i receptor. En total presenta 4 connectors de 655 Mbps màxims, és a dir, uns 2,5 Gbps com els fabricants senyalen per les versions 1.0 i 1.1.



Respecte el nivell físic de la connexió, existeixen diferents connectors en funció del nombre d'enllaços emprats com podem veure a continuació:

<b>Enllaços</b>	<b>Connectors</b>	<b>Ample de banda</b>	<b>Possibles usos</b>
x1	36	5 Gbps	Gigabit Ethernet
x4	64	20 Gbps	10 Gigabit Ethernet, SAS i SCSI
x8	96	40 Gbps	Perifèrics diversos
x16	164	80 Gbps	Targetes gràfiques

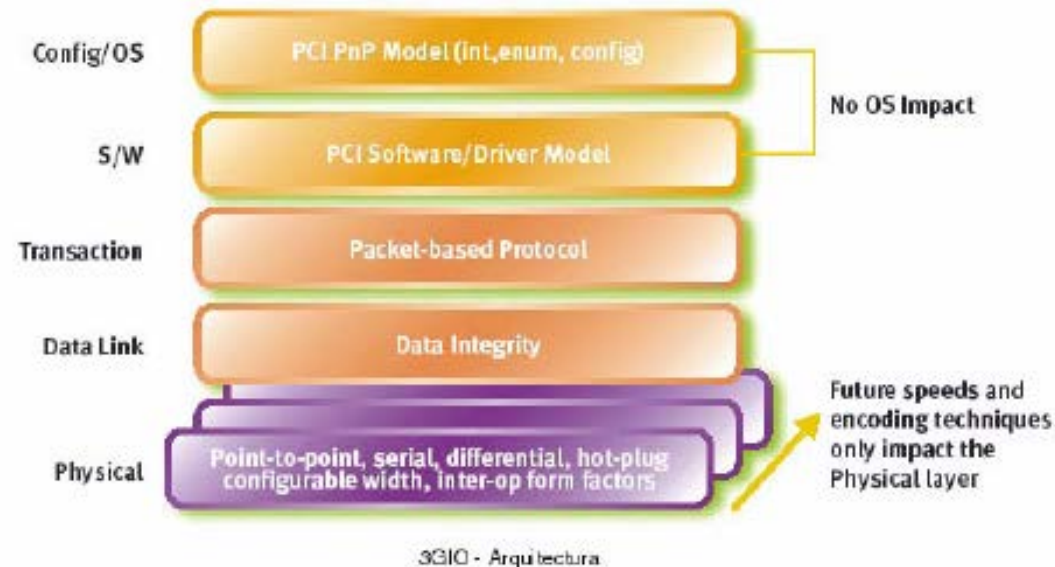
Per calcular l'ample de banda, els fabricants empen el següent càlcul:

$$\text{Nombre d'enllaços} \cdot 2 \text{ (doble banda)} \cdot 2,5 \text{ Gbps/línia}$$

## PCI Express: Protocol

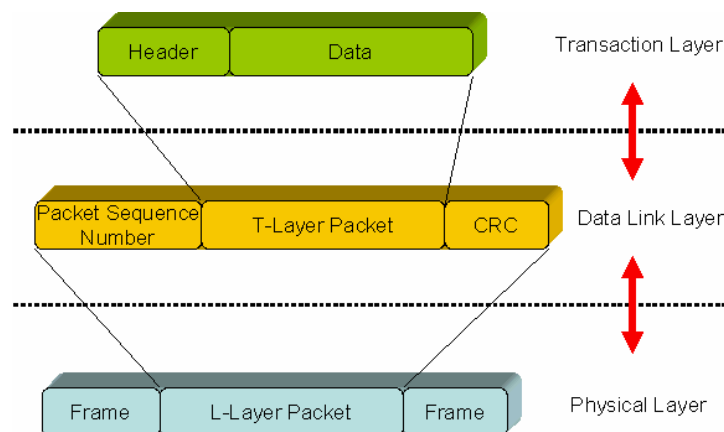
El protocol es divideix en 5 capes:

- La **capa física** fa la connexió entre els dos dispositius que s'han de transferir informació.
- La **capa d'enllaç (Data Link)** és l'encarregada de garantir la fiabilitat i integritat de les dades de cada paquet. A més del número de seqüència i el CRC, un protocol de control de flux garanteix que només es transmeten paquets quan hi ha un *buffer* disponible per rebre a l'altre costat. Els paquets corruptes es retransmeten automàticament.
- La **capa de transacció** crea els paquets a partir de les peticions de la capa de software a la d'enllaç, implementant-les com a transaccions. Cada paquet té un identificador únic, suportant adreçaments de 32 y 64 bits.
- Les **capes de software** són la clau per aconseguir la compatibilitat software. La inicialització i el *runtime* no ha canviat respecte al PCI ja que es vol que els sistemes operatius puguin fer servir PCI Express sense necessitat de modificacions. Els dispositius són enumerats de forma que el sistema operatiu els pugui trobar i assignar-los recursos, mentre que el *runtime* reutilitza el model *load-store* i *shared-memory* del PCI.

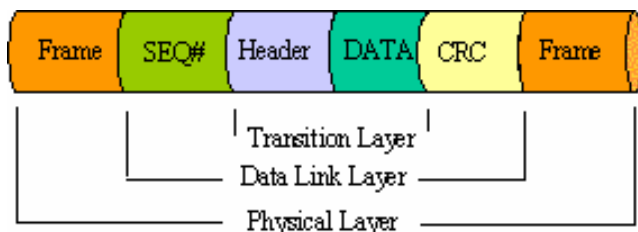


# PCI Express: Protocol

Com hem dit, el protocol bàsic es divideix en tres capes: física (*Physical Layer*), dades (*Data Link Layer*) i transacció (*Transaction Layer*), amb la següent estructura:



Trama amb tots els nivells ordenats i una taula amb les característiques de cada segment:



Element	Nivell	Funció	Bytes
Frame	Inici Físic	Inici	1
SEQ#	Inici dades	Número de la seqüència de dades	2
Header	Transacció	Capçalera de la transacció	16 a 20
DATA		Camp de dades	0 a 4960
CRC	Fi dades	Comprovació de redundància local per cicles (LCRC)	4
Frame	Fi Físic	Fi	1

## PCI Express: Conclusions

### Consum energètic

- Baix consum degut a les baixes tensions de funcionament.
- Implementa funcions d'estalvi d'energia.

### Estalvi de costos

- La transmissió sèrie estalvia moltes línies i per tant potetes als circuits integrats, pistes a les plaques de circuits impresos, i fils als cables permeten connectors més petits.

### Compatibilitat amb PCI

- Tot i que és evident la incompatibilitat hardware, es manté una compatibilitat amb PCI a les capes software, el que permet abaratir els costos d'implementació dels controladors de dispositius, anomenats popularment com a *drivers*.

### Connexió amb calent, *hot-plug*.

### Canvi amb calent, *hot-swap*.

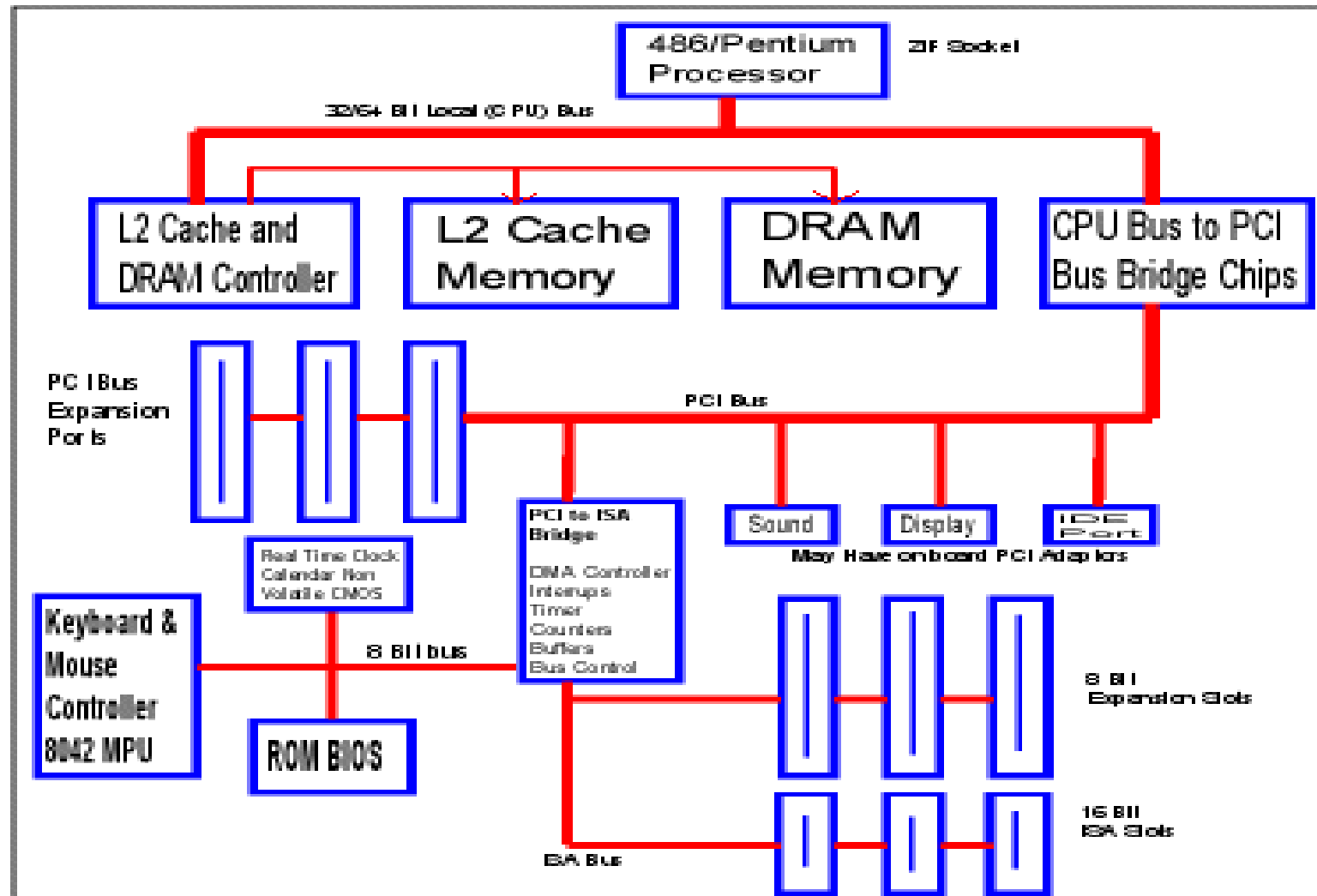
### Integritat de la senyal

- Al disminuir el nombre de pistes, permet prendre mesures contra les interferències Electromagnètiques de forma més senzilla i eficaç.
- Les senyals diferencials disminueixen els problemes amb el soroll.

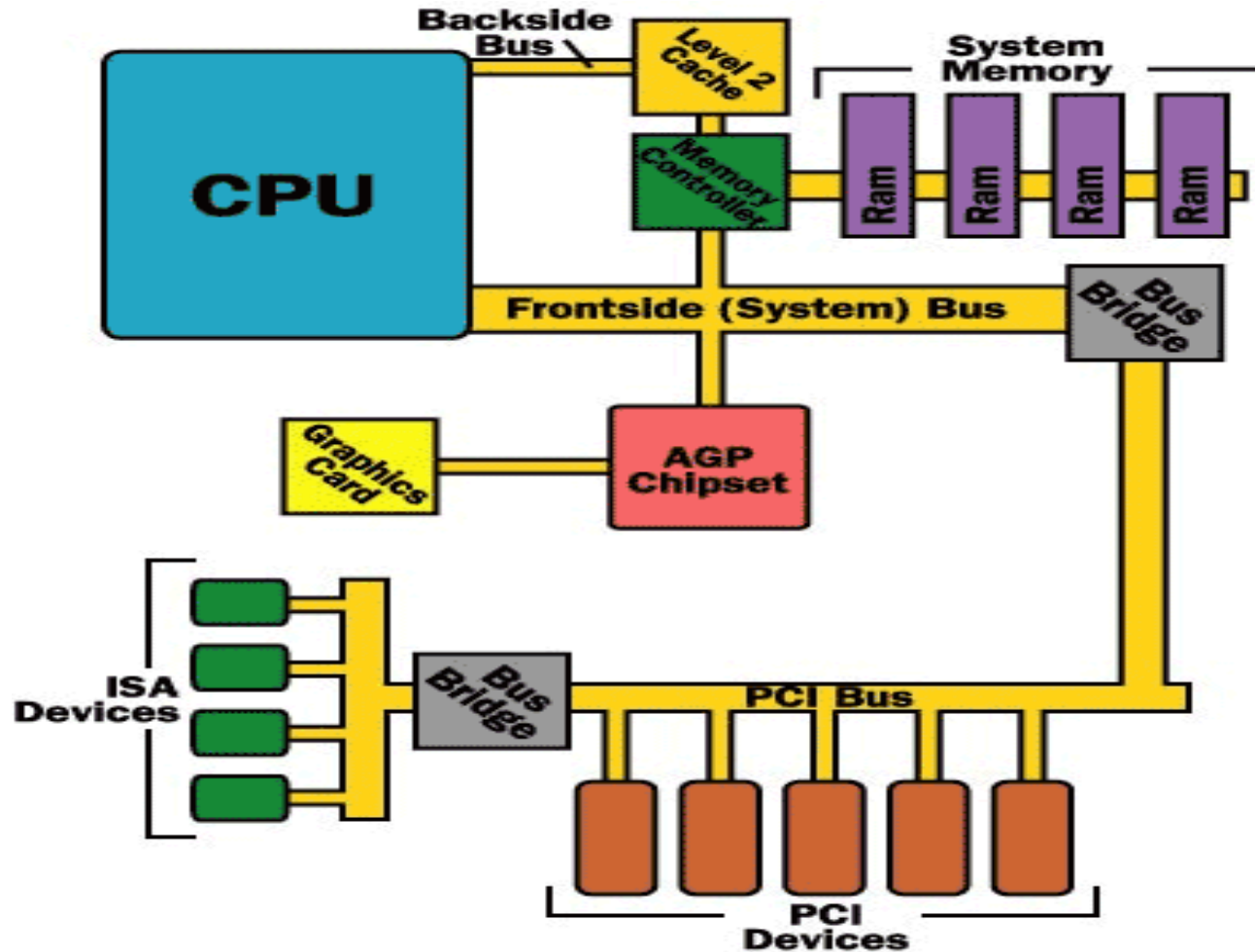
### Té gestió d'errors.



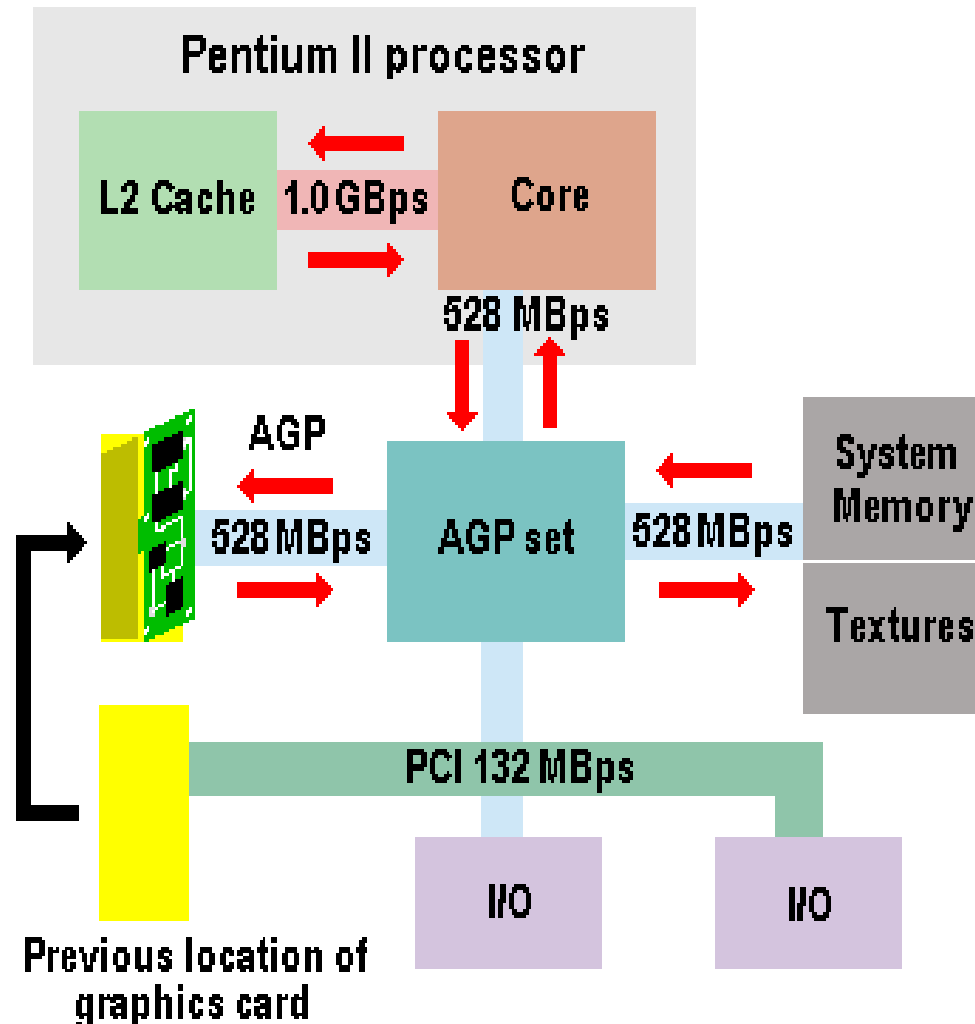
# Exemple de sistema amb BUS PCI-ISA



Exemple de sistema amb BUS PCI-AGP i caché de nivell 2 externa



# Exemple de sistemes amb diferents nivells BUSos mitjançant Chipset



## Exemple de sistemes amb BUSos gestionats per chipset: MCH i ICH

El processador té un bus que el comunica amb el MCH mitjançant el **FSB** (*Front Side BUS*).

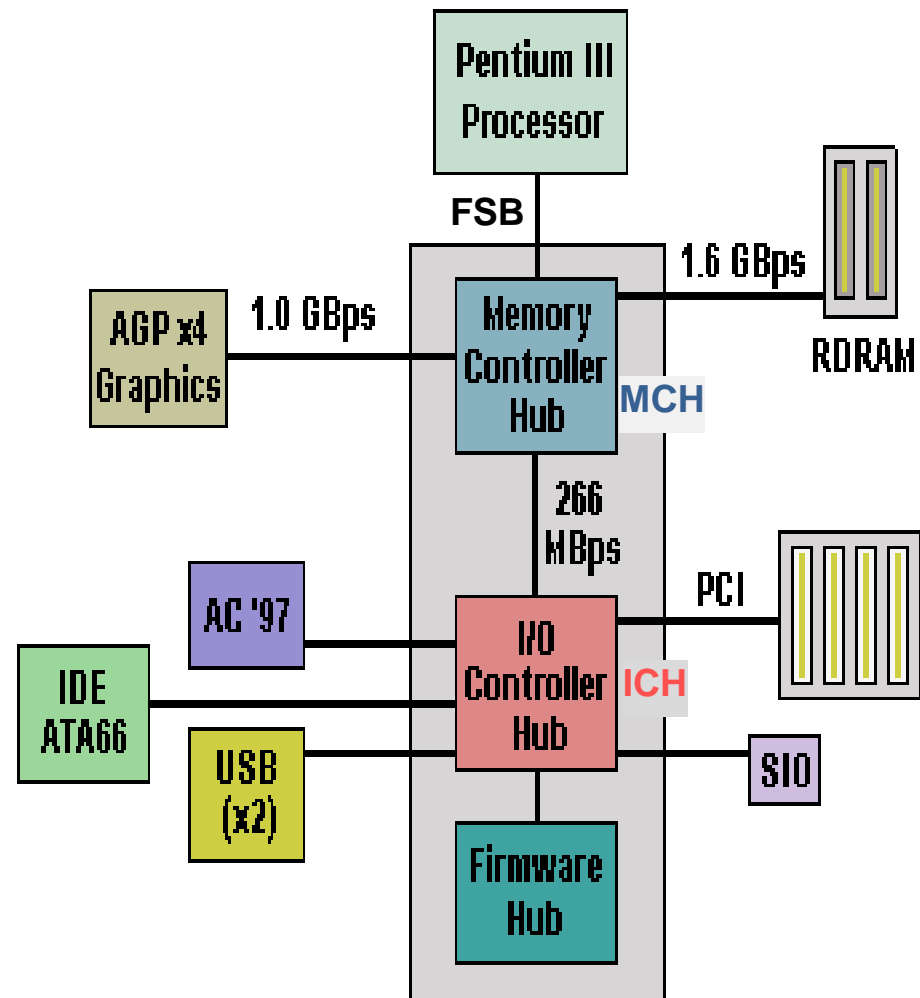
**MCH:** *Memory Controller HUB*.

El **MCH** comunica amb:

- La memòria principal mitjançant el “BUS de Memòria” (en aquest cas a 133 MHz, 128 Bit = 1.6 GB/s).
- La tarja gràfica (aquí un port AGP 4x).
- El ICH mitjançant el BUS “**Interlink**” (a 266 MB/s).

**ICH:** *I/O Controller Hub*. Aquest element del chipset es comunica amb tots els dispositius d'E/S (menys el video): Bus PCI, discs durs i òptics (PATA), dispositius USB, mòdems, àudio, ethernet.. I amb el FWH.

*Firmware HUB*. EEPROM amb la BIOS.



## CHIPSET

- Situat a la Placa Base, el *Chipset* consta d'un conjunt de xips que fan la majoria de tasques de transferència d'informació entre els elements que formen el computador.
- Les funcions del *chipset* es divideixen en dos grups:
  - El “*Northbridge*” o MCH (*Memory Controller HUB*): proper a la CPU.
  - El “*Southbridge*” o ICH (*Input/Output Controller HUB*): proper als perifèrics.
- Entre els dos hi ha una connexió (*Interlink*) a molt alta velocitat. I sobretot entre el “*Northbridge*” i la CPU.

## CHIPSET: NORTHBRIDGE o MHC

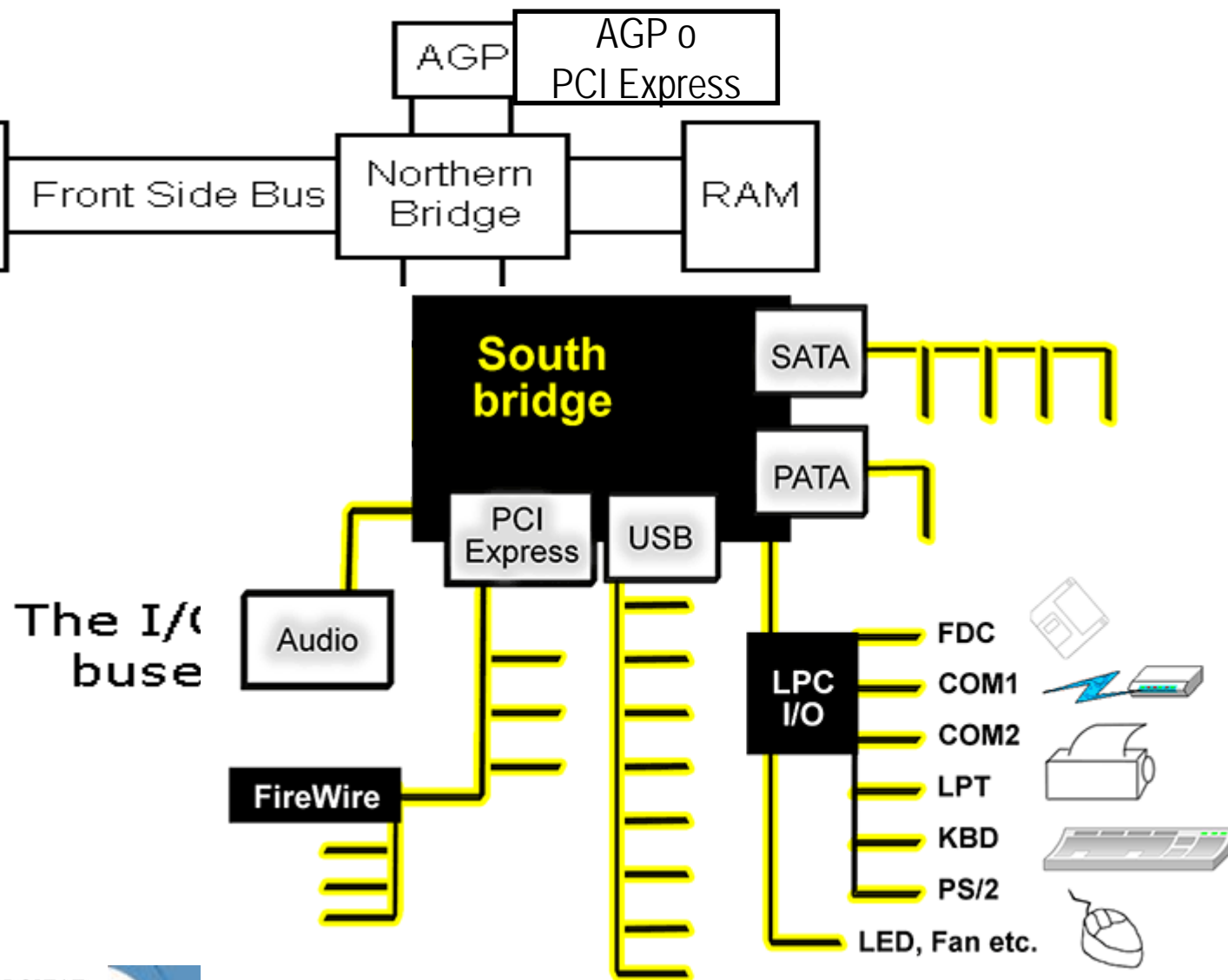
- S'encarrega de les tasques crítiques amb temps.
- S'ha de preocupar de proporcionar les dades als elements que treballen a més alta velocitat:
  - Transferència de dades entre CPU i RAM\*.
  - Transferències de dades entre RAM i Sistema Gràfic/Video (PCI-Express o AGP).
  - Transferències de dades amb el “*Southbridge o ICH*”

\* Per les CPUs de Intel fins al core 2, per AMD i intel core i5-i7 és diferent.

## CHIPSET: SOUTHBRIDGE o IHC

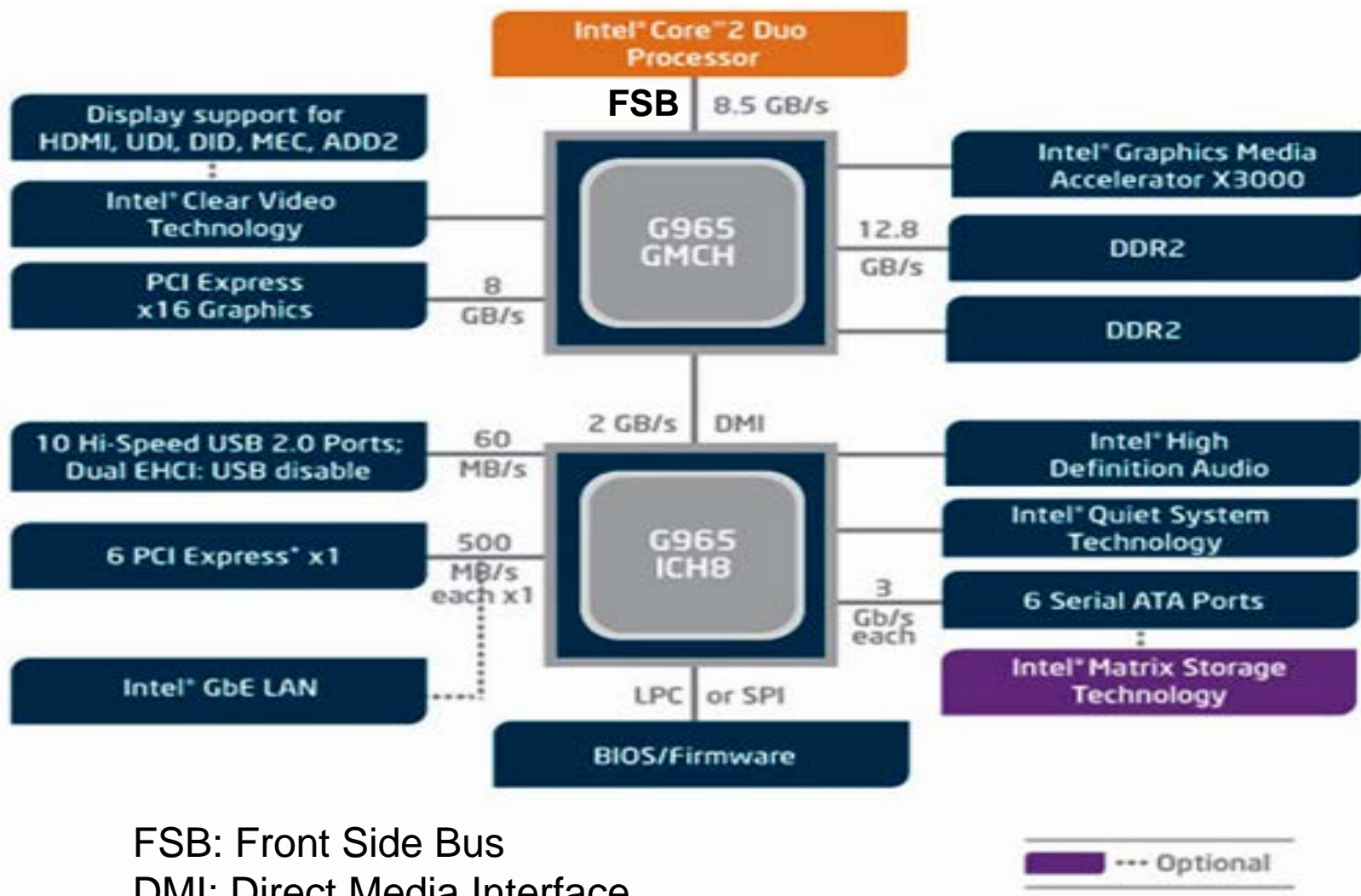
- Proporciona suport a una ampla varietat de dispositius.
- Per exemple per a busos secundaris:
  - Dispositius que es comuniquen amb el bus PCI.
  - SATA: per connexió de Disc Durs i CD/DVD.
  - PATA o IDE: pel mateix.
  - USB: per connectar impressores, escàners, tabletes gràfiques, càmeres...
  - PS/2: Teclats i ratolins.
  - Audio.
  - Ethernet....
- A més, connecta amb la memòria del Firmware.

# Estructural genèrica de BUSOs d'un sistema





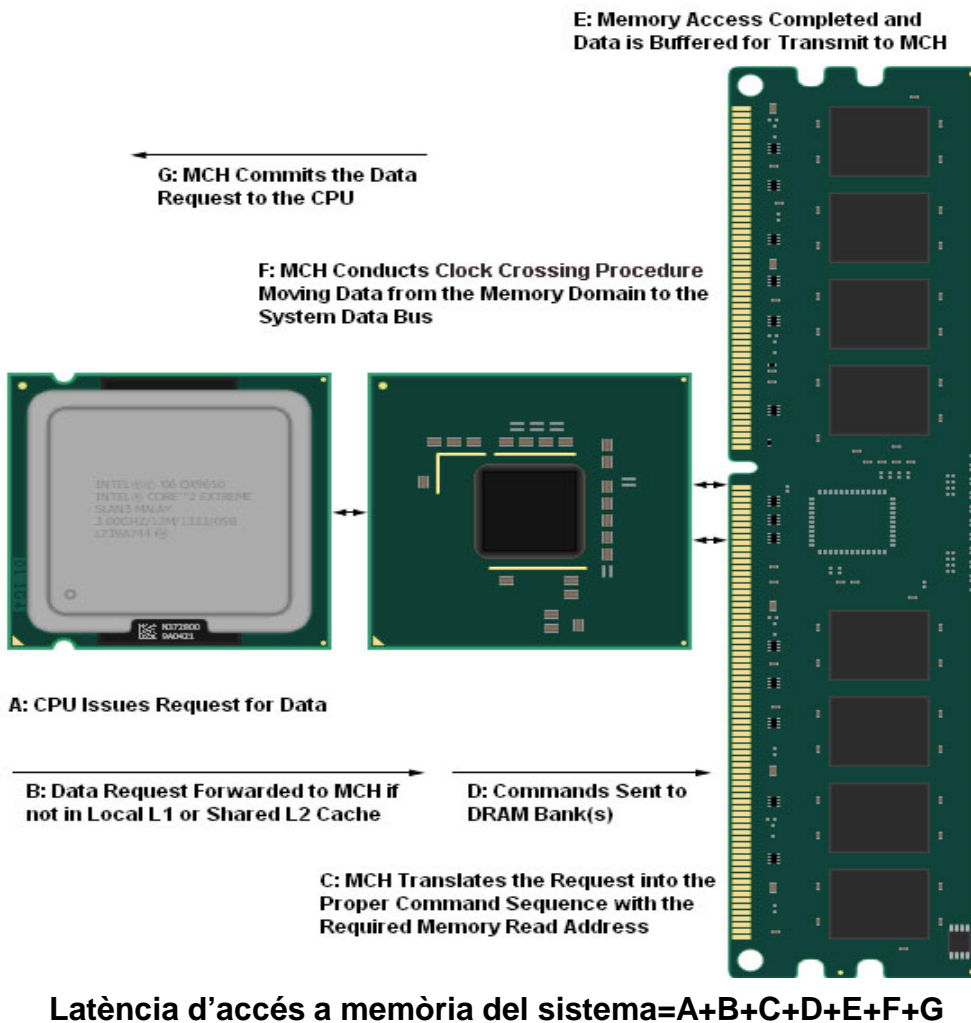
# Exemple de sistema amb arquitectura intel Core 2



FSB: Front Side Bus

DMI: Direct Media Interface

# Problema de l'accés a la memòria principal mitjançant el FSB i el MCH



Transferències entre processador i memòria principal amb l'esquema FSB+MCH. Les transferències es fan mitjançant el *NorthBridge* del chipset.

## Latències no dependents de la Memòria

- A. CPU fa una petició d'informació.
- B. Si no és a la caché, la petició es trasllada al MCH.
- C. MCH trasllada la petició en la seqüència de comandaments adients per la memòria.

## Latències que depenen de la Memòria

- D. S'envien els comandaments als banks de DRAM
- E. Es completa l'accés a memòria i es "registren" les dades per ser transmesos al MCH.

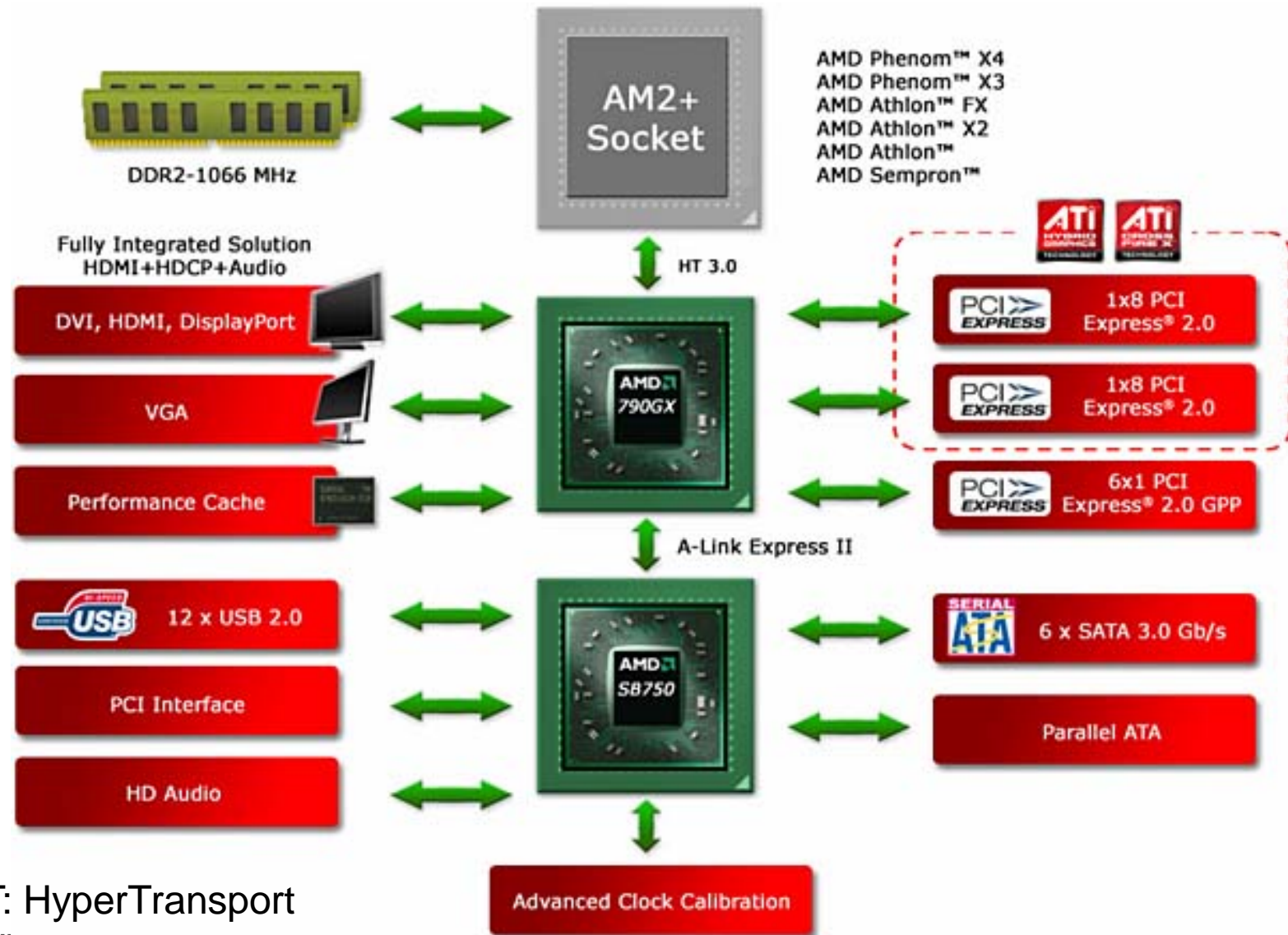
## Latències no dependents de la Memòria

- F. El MCH mou les dades de la memòria al FSB.
- G. MCH commits the data request to CPU

Només una tercera part de la latència total és funció de les temporitzacions de la memòria i de la seva velocitat de bus, la resta depèn del FSB de la placa base i el processador.

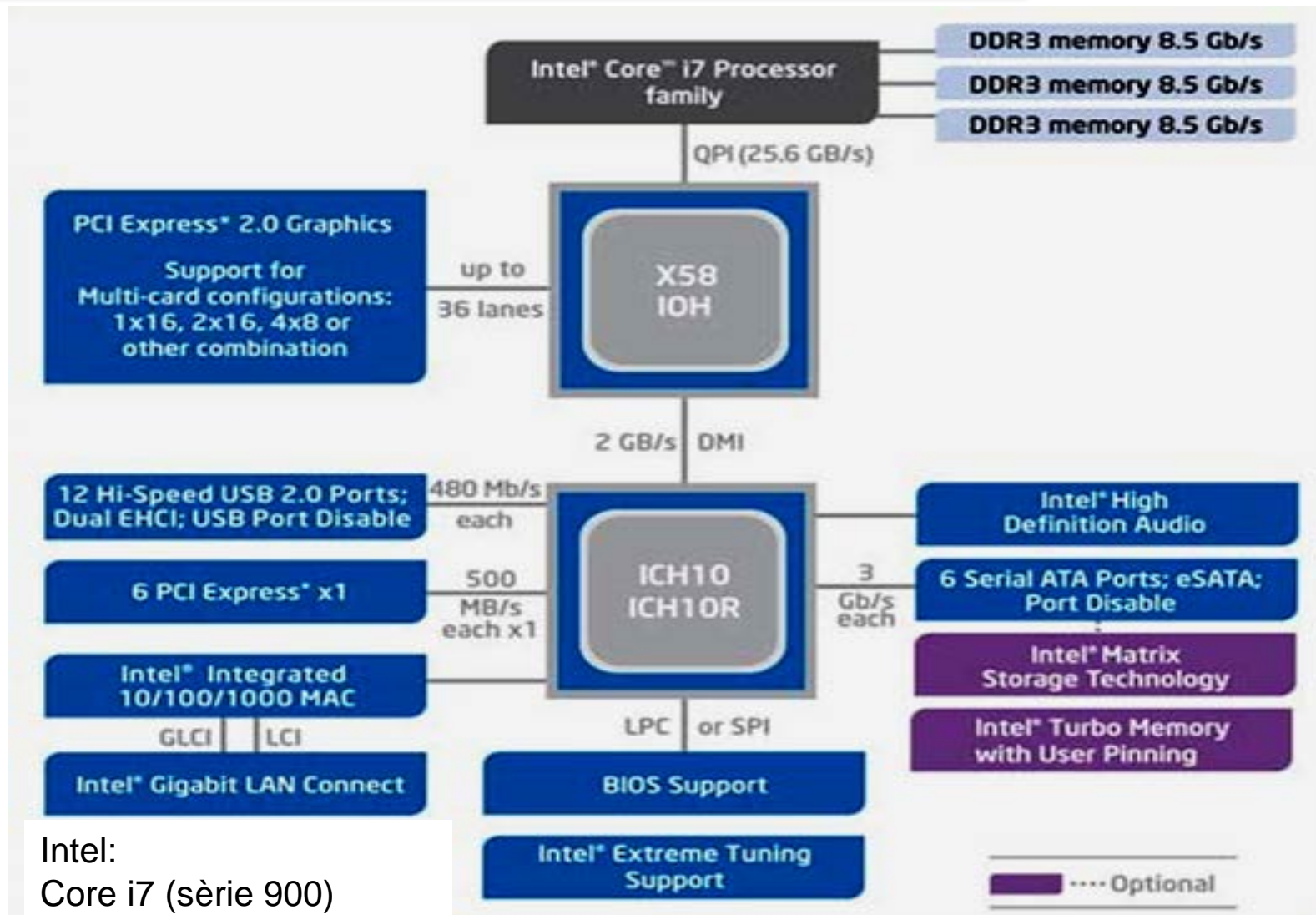
**Això fa que si canviem una memòria per una el doble ràpida només es noti una petita millora**

# Exemple de sistema amb AMD



HT: HyperTransport

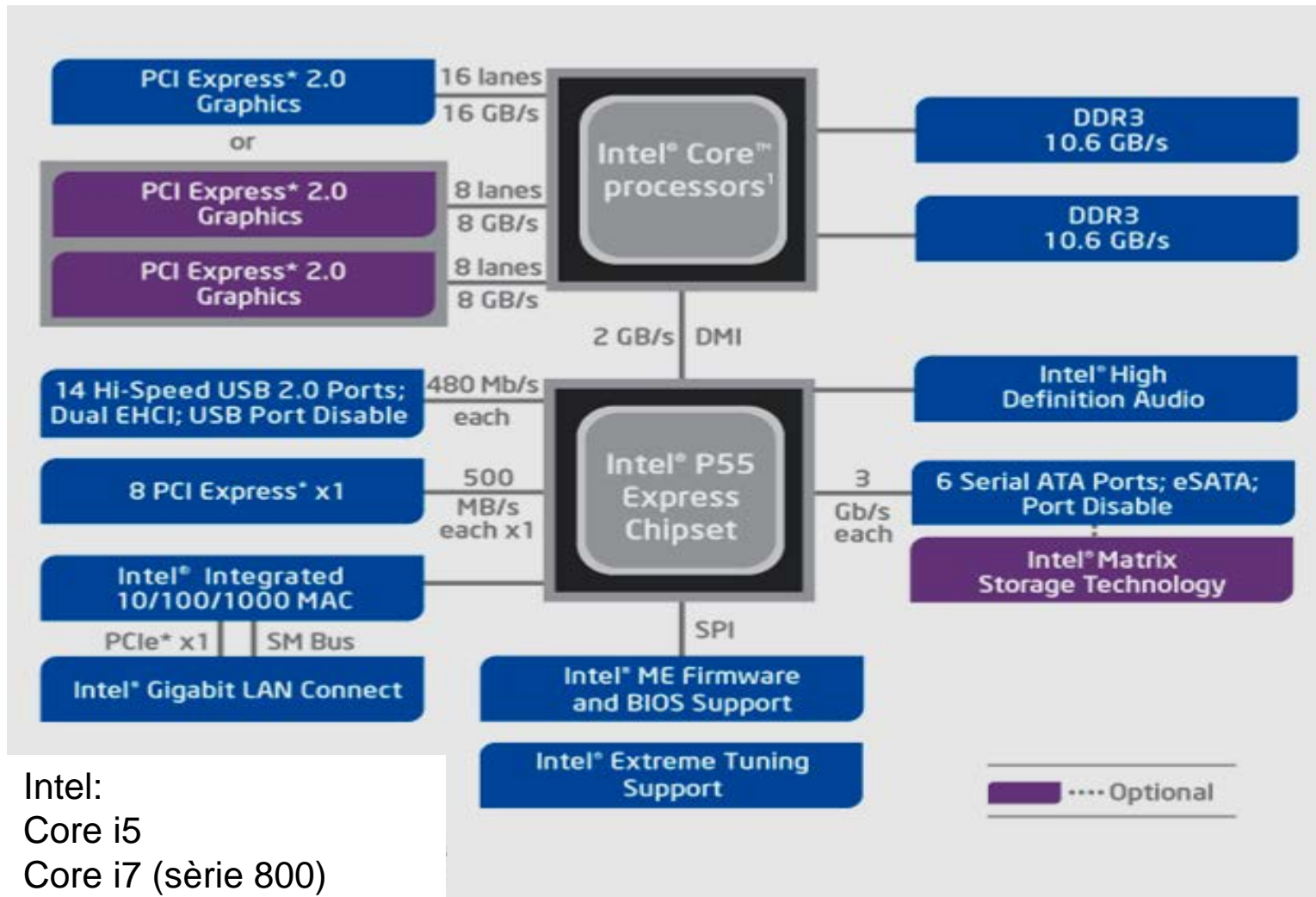
# Exemple de sistema amb arquitectura intel NEHALEM



Intel:  
Core i7 (sèrie 900)



# Exemple de sistema amb arquitectura intel Lynnfield i Havendale



Intel:  
Core i5  
Core i7 (sèrie 800)

## Comparació QPI i HyperTransport

<b>Versió Intel QuickPath Interconnect</b>	<b>Año</b>	<b>Frecuencia max.</b>	<b>Ancho enlace max.</b>	<b>Ancho de banda agregado max. (bidireccional)</b>	<b>Ancho de banda max. a 16-Bit (unidireccional)*</b>
1.0	2008	3.2 GHz	20 Bit	25.6 GB/s	12.8 GB/s

<b>Versió HyperTransport</b>	<b>Año</b>	<b>Max. Frecuencia HT</b>	<b>Max. Ancho enlace</b>	<b>Max. Ancho de banda agregado (bidireccional)</b>	<b>Max. Bandwidth a 16-Bit (unidirectional)*</b>
1.0	2001	800 MHz	32 Bit	12.8 GB/s	3.2 GB/s
1.1	2002	800 MHz	32 Bit	12.8 GB/s	3.2 GB/s
2.0	2004	1.4 GHz	32 Bit	22.4 GB/s	5.6 GB/s
3.0	2006	2.6 GHz	32 Bit	41.6 GB/s	10.4 GB/s
3.1	2008	3.2 GHz	32 Bit	51.2 GB/s	12.8 GB/s