

Busos de comunicació

- ❖ Introducció
- ❖ Definicions
- ❖ Jerarquia
- ❖ Protocols
- ❖ Busos normalitzats

Introducció

- **Objectiu:** analitzar els elements d'interconnexió que uneixen els diferents components d'un computador
- **Bus:** conjunt de línies de senyal que enllaça la CPU, la memòria i els perifèrics d'E/S, permetent la transferència d'informació entre ells
 - Línies de direcció: @ de memòria o de ports d'E/S
 - Línies de dades: informació a llegir o escriure
 - Línies de control:
 - Operacions a realitzar: lectura, escriptura, ...
 - Control de transferència: protocol i temporització
 - Arbitratge: determina quin dispositiu utilitza el bus (CPU, DMA, dispositius E/S, ...) en cas de conflicte
- **Funcionament** general: el dispositiu ha d'obtenir l'ús del bus i transferir la dada a través del bus, tot això seguint un determinat protocol
- **Operacions** en un bus: lectura / escriptura (dada / bloc), Read-Modify-Write, etc.

Definicions

- **Cicle de bus:** temps necessari per realitzar una transferència elemental d'una dada entre dos dispositius
 - Operació bàsica del bus
 - Etapes: petició del bus, arbitratge, adreçament, transferència, detecció d'errors, notificació
- **Amplada del bus:** nombre de línies de dades al bus
- **Ample de banda del bus:** paràmetre que expressa el nombre màxim d'elements d'informació (bytes) que es poden transmetre pel bus en unitat de temps
 - Exemple: 1 MB / s $\rightarrow 10^6$ bytes/s
 - Velocitat de transmissió: longitud, medi físic, lògica de control, etc.
- **Amplada de banda efectiva:** Quantitat real d'informació que es pot transmetre. En aquest cas es té en compte que per transmetre dades es necessita utilitzar cicles de bus en el protocol d'accés i d'arbitratge.
$$AB = ((n/8) / n^{\circ} \text{ de cicles}) * f$$
- **Protocol del bus:** pautes que han de seguir els dispositius connectats al bus per aconseguir una comunicació correcta

- **Amplada del bus:** Es defineix com el nombre de línies d'informació que té el bus. Es pot diferenciar entre busos sèrie: (USB, Firewire) i paral·lel (PCI, bus del sistema)
- **Longitud del bus:** pot oscil·lar entre menys d'un metre (bus del sistema, PCI), varis metres (bus SCSI) i fins a centenars de metres (Ethernet). Com més llarg sigui, menys velocitat i amplada.
- **Freqüència de funcionament** (pels busos síncrons): freqüència del senyal de rellotge que regeix les transferències.
- **Amplada de banda teòrica:** Quantitat d'informació que pot transmetre el bus. S'acostuma expressar en MBytes/s. $AB = n/8 * f$, on n és la quantitat de línies del bus de dades.
- **Amplada de banda efectiva:** Quantitat real d'informació que es pot transmetre. En aquest cas es té en compte que per transmetre dades es necessita utilitzar cicles de bus en el protocol d'accés i d'arbitratge.

$$AB = ((n/8) / n^{\circ} \text{ de cicles}) * f$$

Definicions

- Mestre i esclau:
 - Dispositiu mestre (màster): capaç d'iniciar una transferència de bus (és l'amo del bus)
 - Dispositiu esclau (slave): dispositiu passiu, que espera peticions
 - Exemples:

Mestre	Esclau	Operació
CPU	Memòria	Recerca d'instrucció i dades
CPU	Dispositiu d'E/S	Inicialitzar transferència
DMA	Memòria	Transferència de dades

- Relació dinàmica: la unitat A pot ser mestre en una transferència, però esclau en una altra. Per exemple, el controlador de DMA

Jerarquia

- **Problemes:**

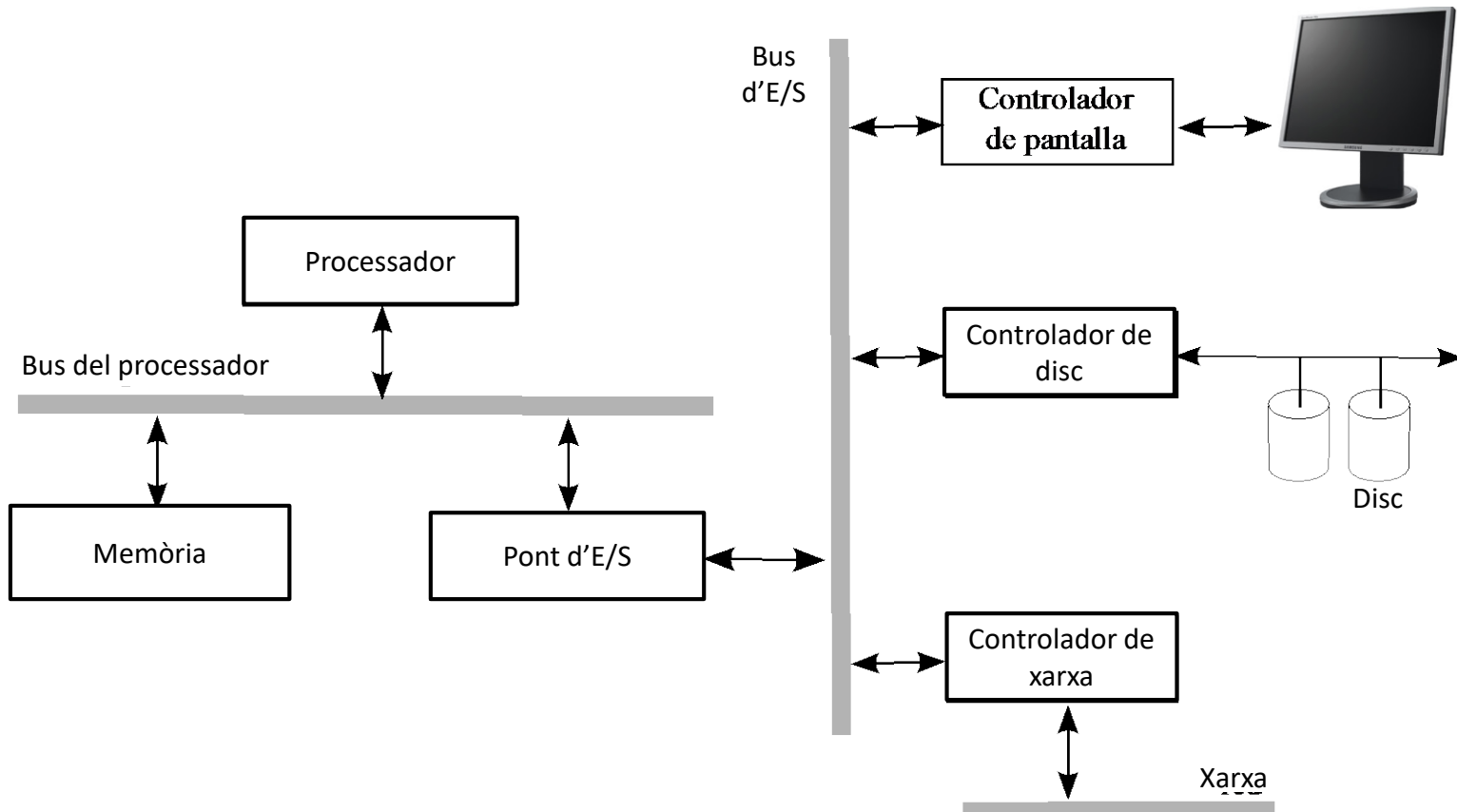
si es connecta un gran nombre de dispositius al bus:

- incrementa el retard de propagació dels senyals entre dispositius
- diferència entre les prestacions de cada dispositiu: velocitat de funcionament, necessitat d'ample de banda, etc.
- saturació de la capacitat del bus
 - bus únic: coll d'ampolla de l'ordinador

- **Solució:**

- utilitzar diversos busos organitzats jeràrquicament, cadascun d'ells pot tenir diferent amplada i velocitat de transmissió
- els dispositius amb el mateix nivell de prestacions comparteixen el mateix bus
- apropar a la CPU als dispositius amb millors prestacions

Esquema de busos en un sistema informàtic



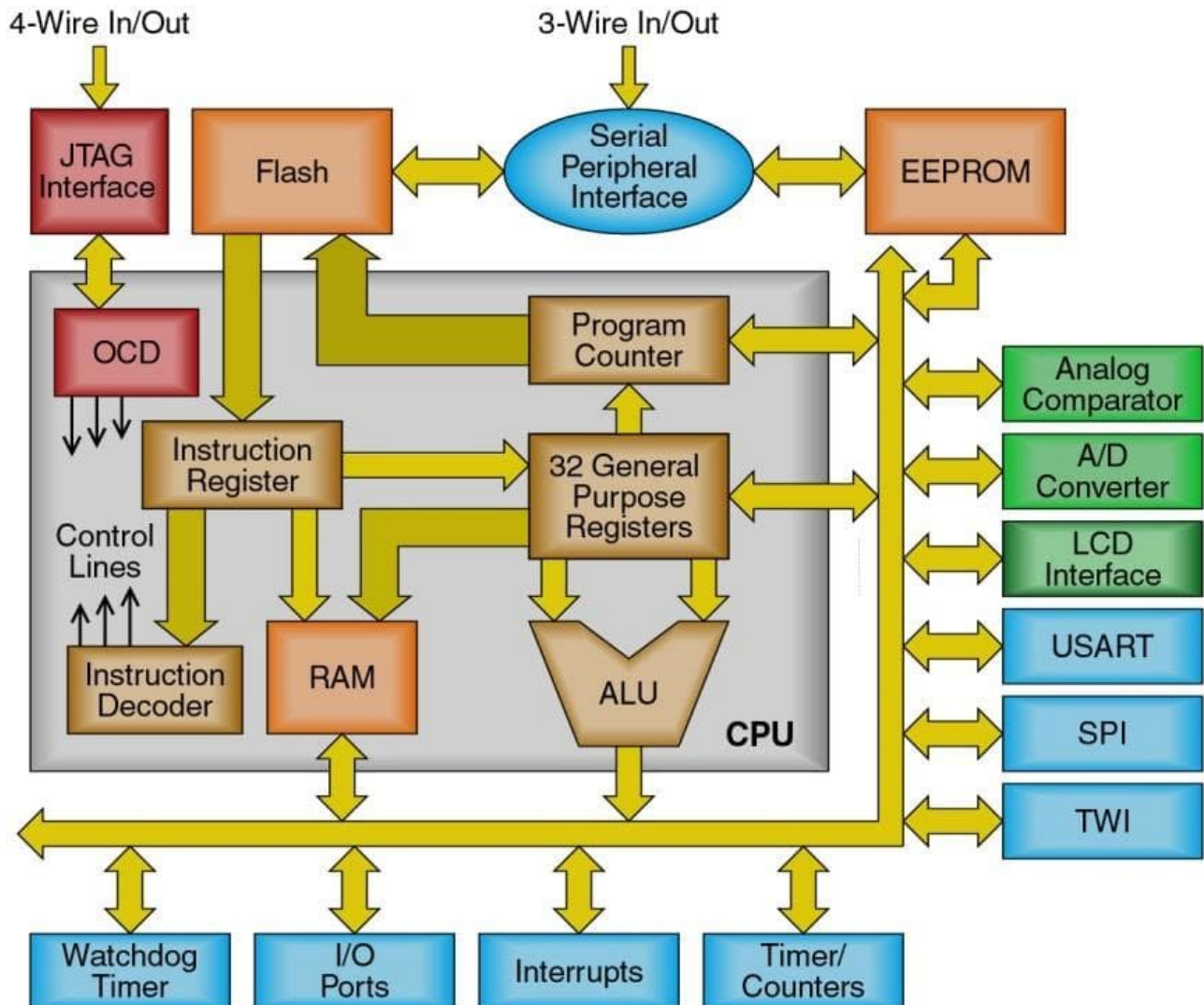
Jerarquia

- **Bus intern:** comunicació interna dins de la CPU
- **Bus del processador:** comunicació entre la CPU i la memòria cau externa. Exemple: bus 400 MHz del P4
 - Poca longitud i alta velocitat
 - Específics per a cada sistema (senyals de el processador)
- **Bus local:** busos per a la connexió processador/DRAM i dispositius d'E/S amb altes prestacions.

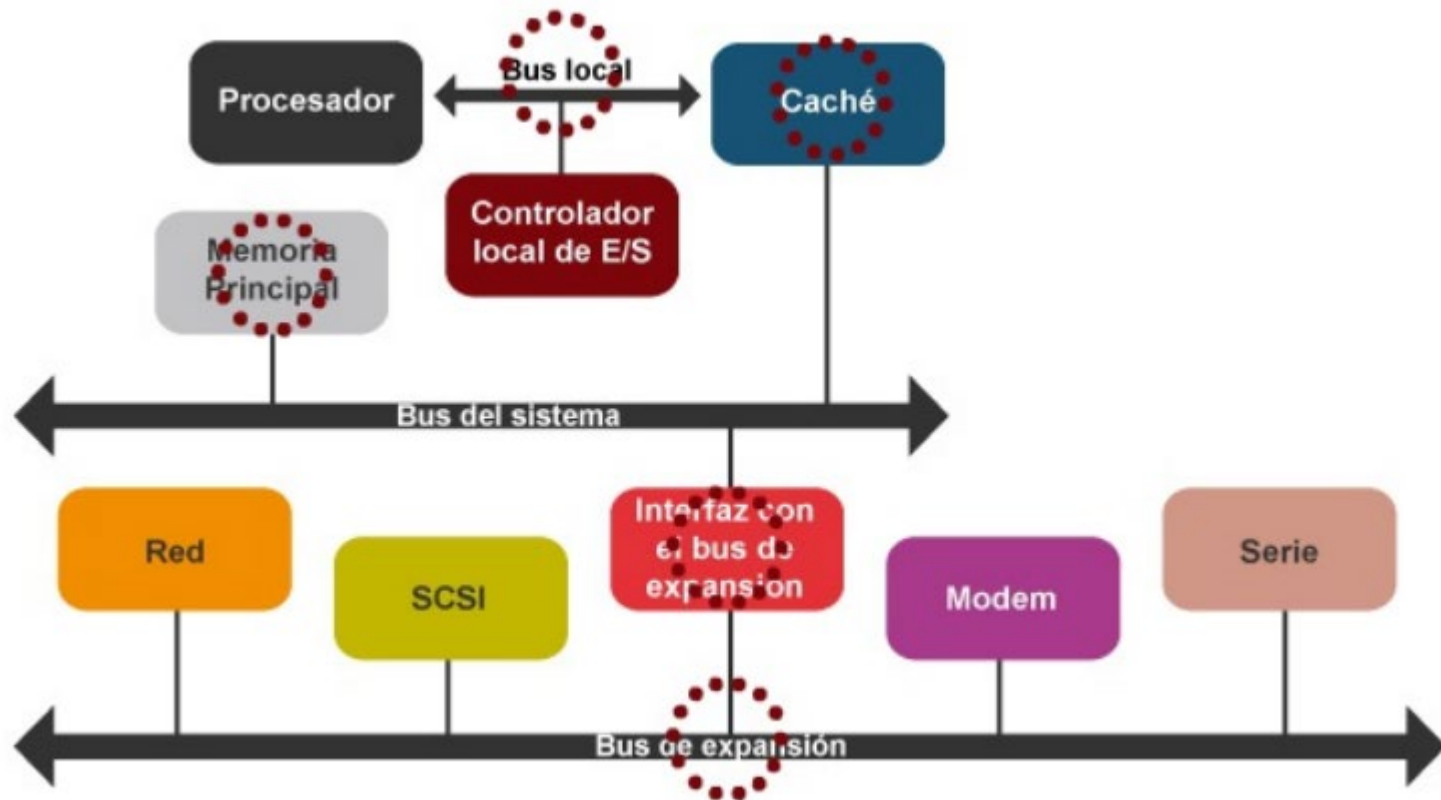
Poden ser:

- busos de caràcter general com el PCI,
- o dedicats, com IDE, SCSI, AGP, USB ...
- **Bus d'expansió:** connexió de dispositiu d'E/S amb menors prestacions (fax, port sèrie, mòdems, ...). Exemple: ISA, MCA
 - Gran nombre de dispositius diferents, amb ample de banda variable
- **Bus de sistema:** connexió entre diferents sistemes processadors que formen un mateix sistema. Exemple: VME

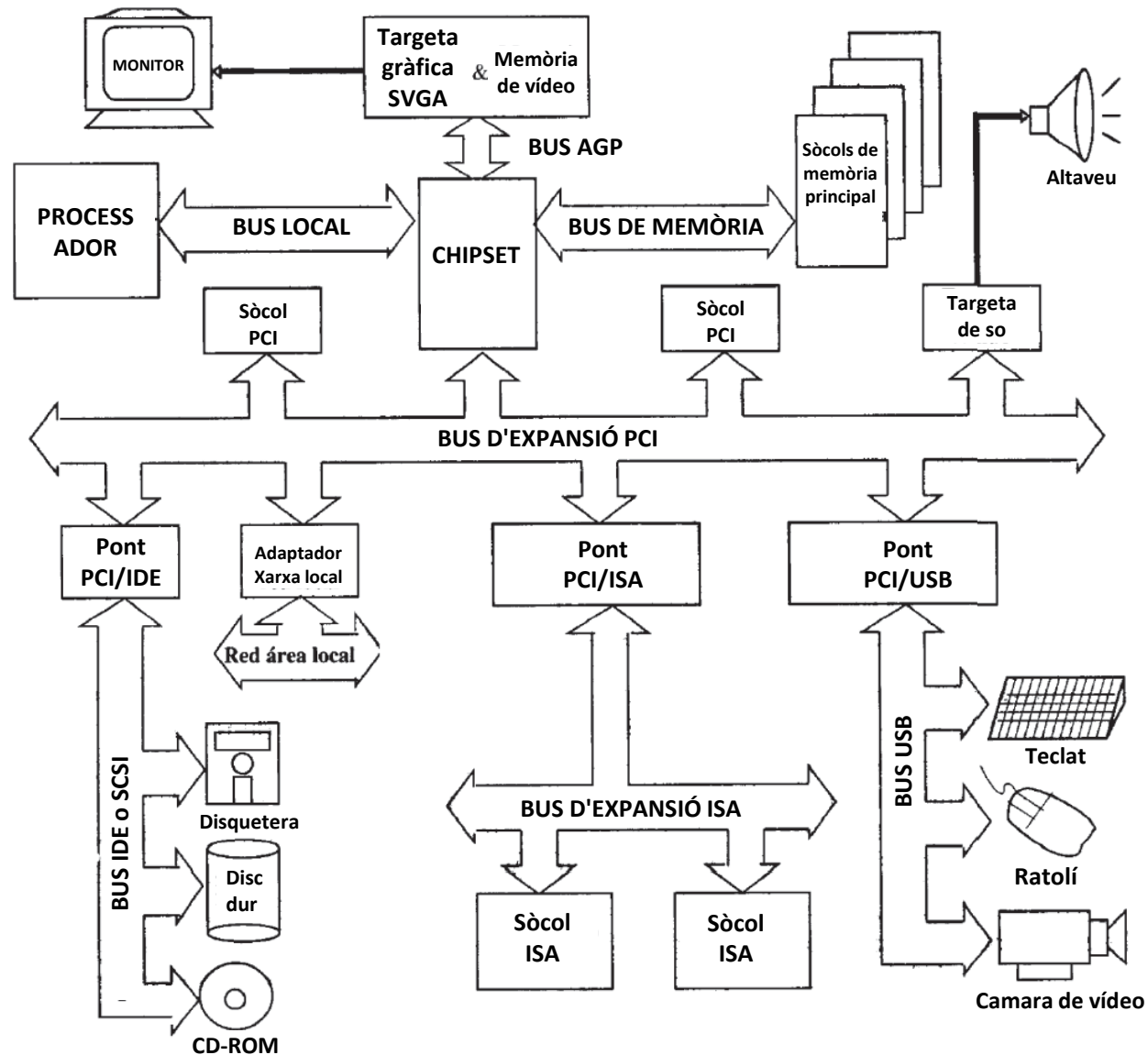
Sense jerarquia de busos



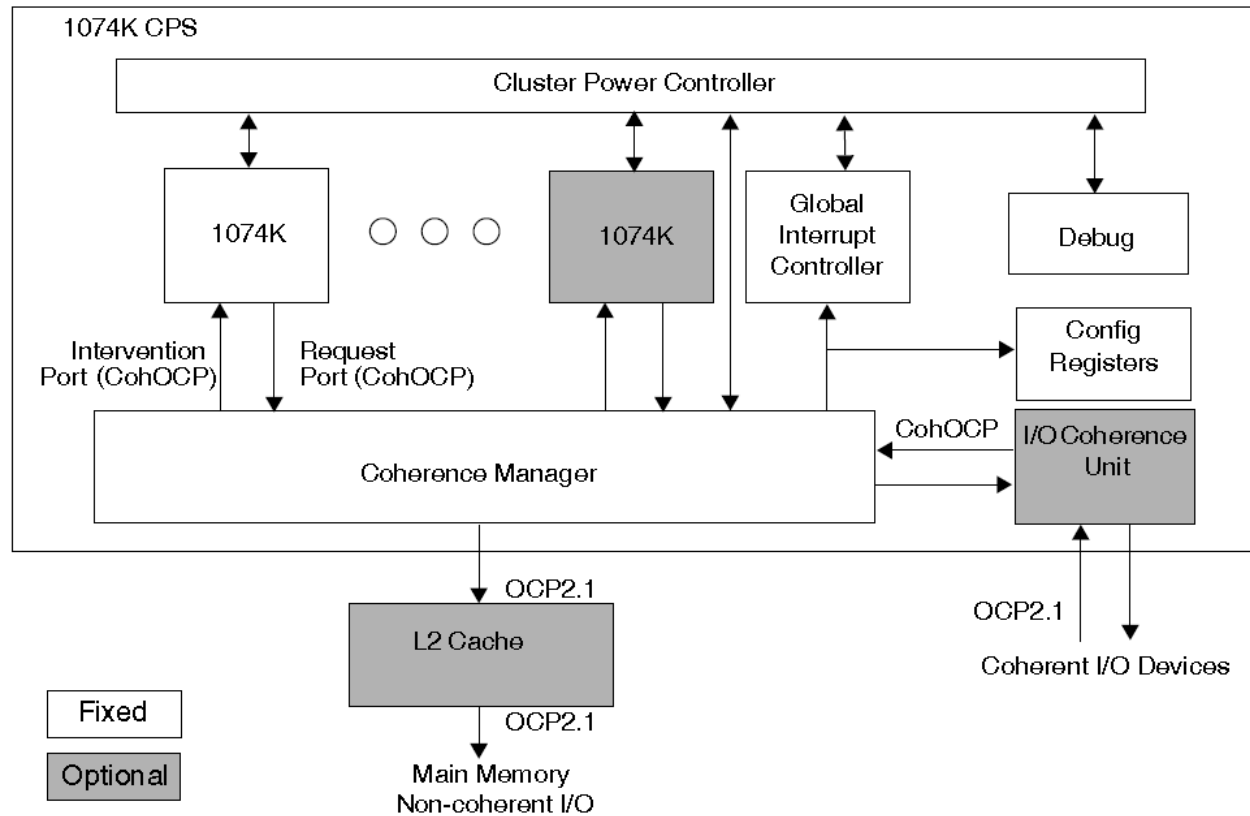
Concepte de bus: Jerarquia de buses



Concepte de bus: Jerarquia de busos

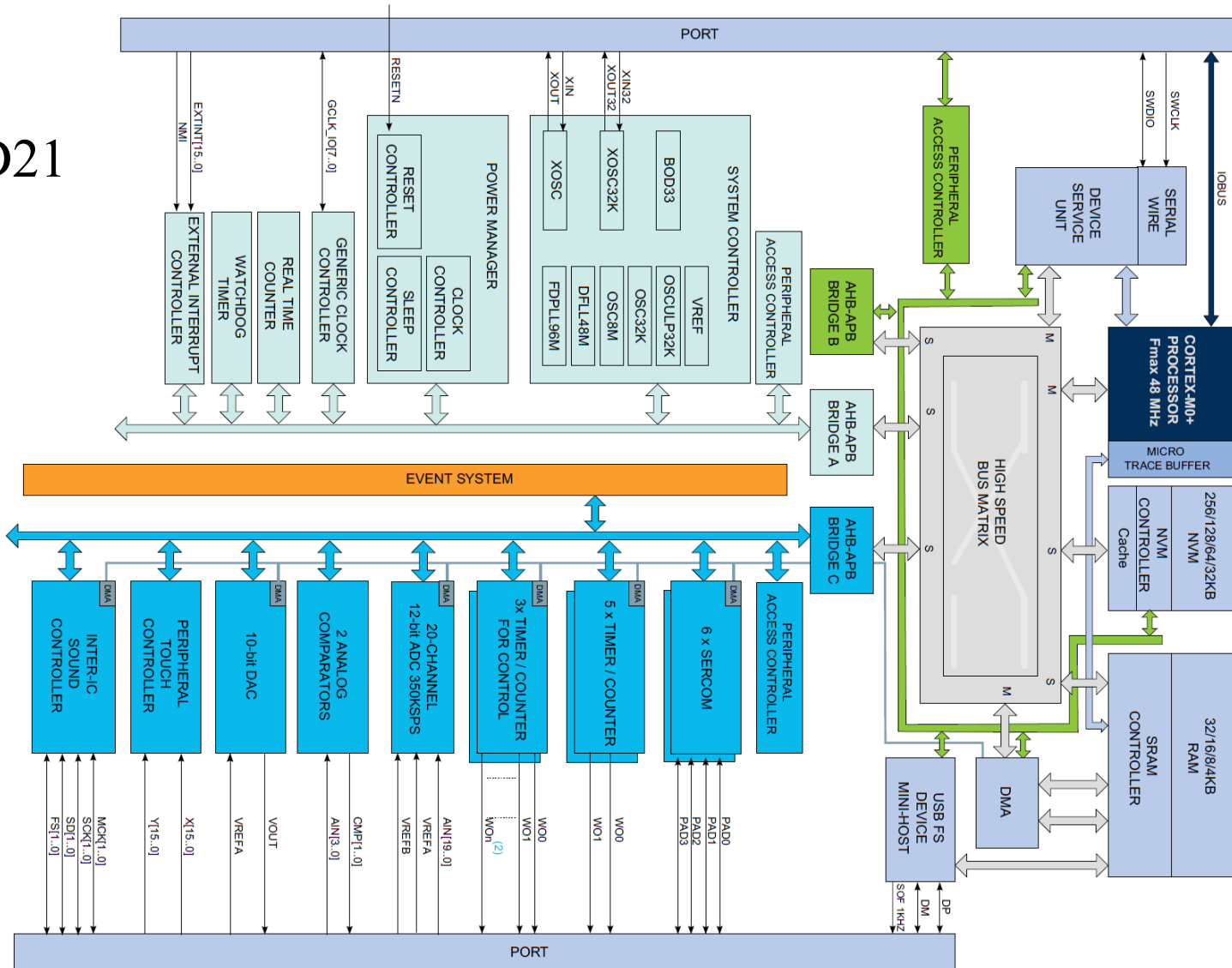


Concepte de bus: Jerarquia de busos



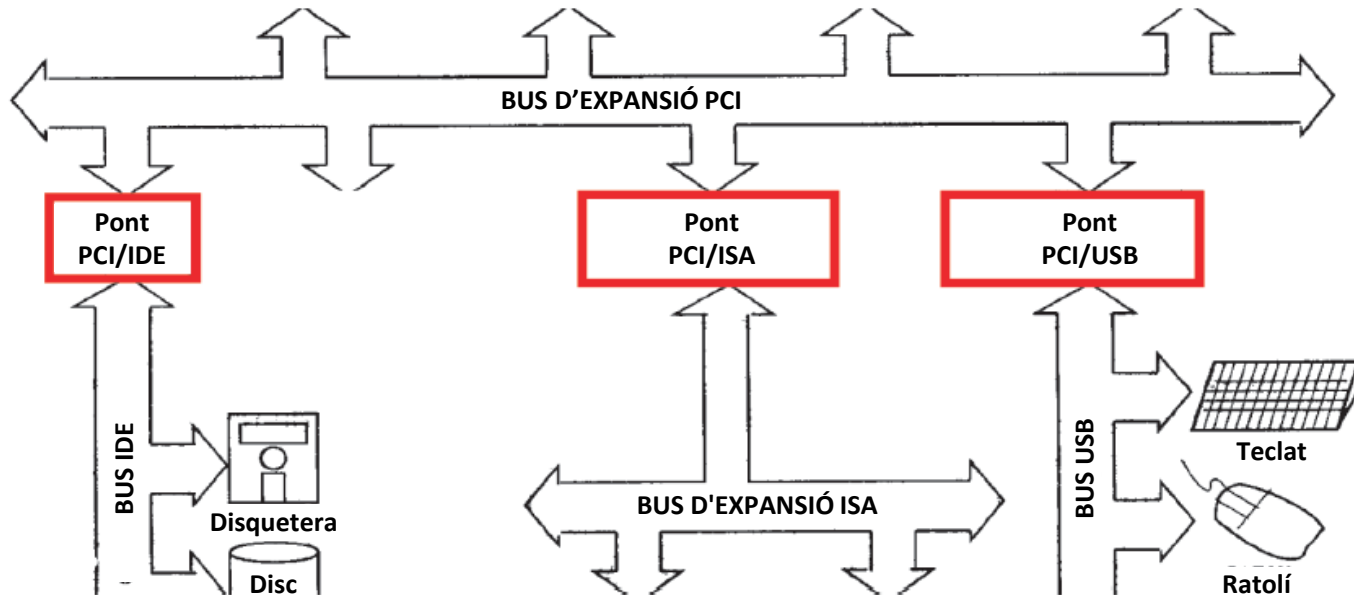
Concepte de bus: Jerarquia de busos

SAM D21



Concepte de bus: Ponts de connexió al bus

- Els **ponts** de connexió al bus connecten dos busos diferents i son capaços d'interpretar el contingut dels paquets que reben.



- **Funcions:**
 - Traducció: connecta busos diferents i tradueix el paquets. Ex: PCI → IDE
 - Gestió del tràfic: decideix si transmet un paquet pels altres extrems en funció de l'adreça del destinatari.

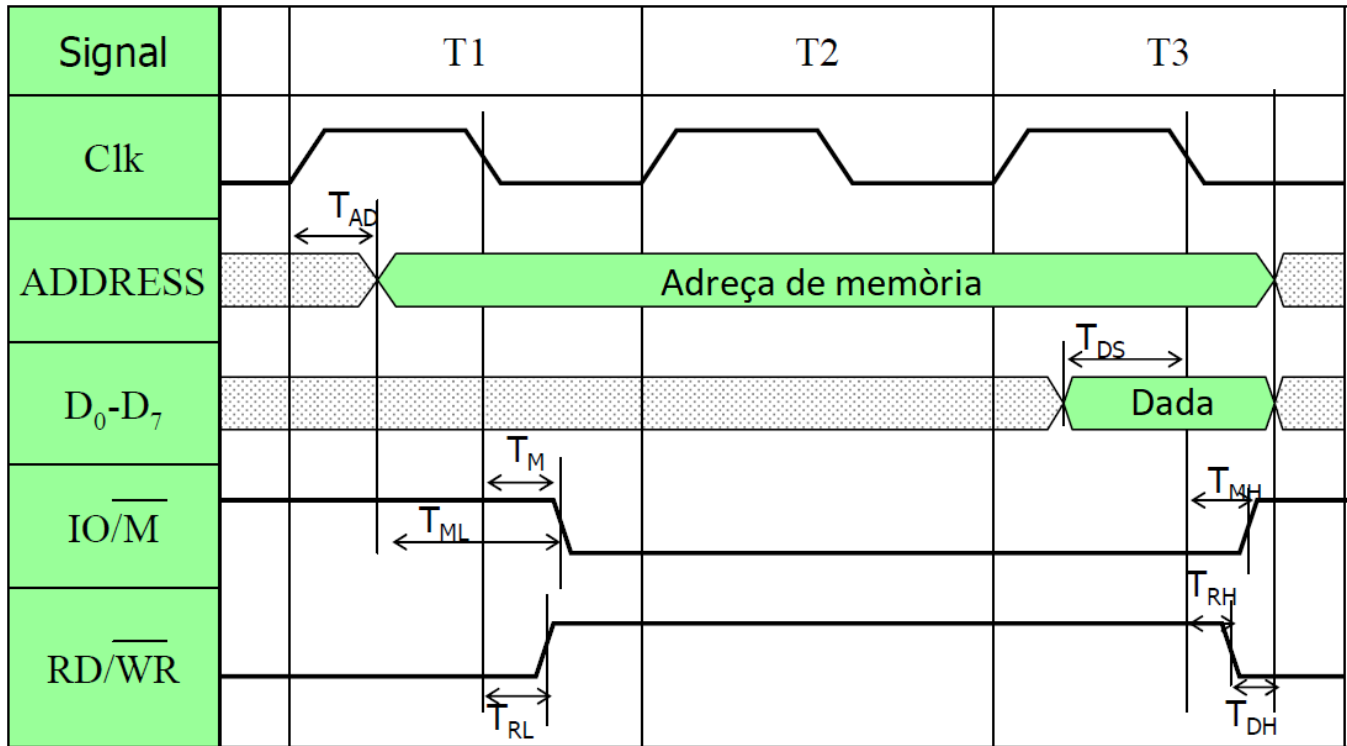
Protocols

- És la forma en la que es coordinen els esdeveniments al bus per garantir que la transmissió sigui correcta
- Dos tipus bàsics de protocols:
 - **síncron i asíncron**
- **Protocol síncron:** fàcil d'implementar, però amb poca flexibilitat (dispositius amb diferents velocitats). Busos curts. Major ample de banda.
- **Protocol asíncron:** permet adaptar la transferència a la velocitat del dispositiu, compagina dispositius lents i ràpids.

Bus síncron

- Transferències controlades per un senyal de rellotge al bus
- Una transferència necessita un nombre enter de cicles
- Exemple de protocol:
 - T_{AD} : (Address output delay) Defineix el temps màxim entre el flanc ascendent de rellotge del primer cicle i l'estabilització de l'adreça
 - T_{DS} : (Data setup time prior to falling edge of ϕ) Defineix el temps en que les dades llegides han d'estar estables abans del flanc de baixada de l'últim cicle
- T_M (MERQ delay from falling edge of ϕ in T_1) i T_{RL} (RD delay from falling edge of ϕ in T_1):
Senyalen el temps màxim en que els senyals IO/M i RD/WR s'han activar després del flanc de baixada del primer cicle

Bus síncron



T_M (MERQ delay from falling edge of ϕ in T_1) i T_{RL} (RD delay from falling edge of ϕ in T_1):
 Senyalen el temps màxim en que els senyals IO/M i RD/WR s'han activar després del flanc de baixada del primer cicle

Freqüència 4 MHz -> 250 ns (cicle de rellotge)

3 cicles -> 750 ns de cicle de lectura

Ample de banda: 1,33 Mbyte / s (1 byte / 750 ns)

- T_{AD} : 110 ns (màxim) T_{DS} : 50 ns (mínim)
- T_M i T_{RL} : 85 ns (màxim)
 - cicle 1 -> 125-85 = 40 ns
 - cicle 2 -> 250 ns
 - cicle 3 -> 125 -50 = 75 ns

la memòria disposa de 365 ns (en el pitjor dels casos) per a col·locar les dades al bus des que s'activa el senyal RD

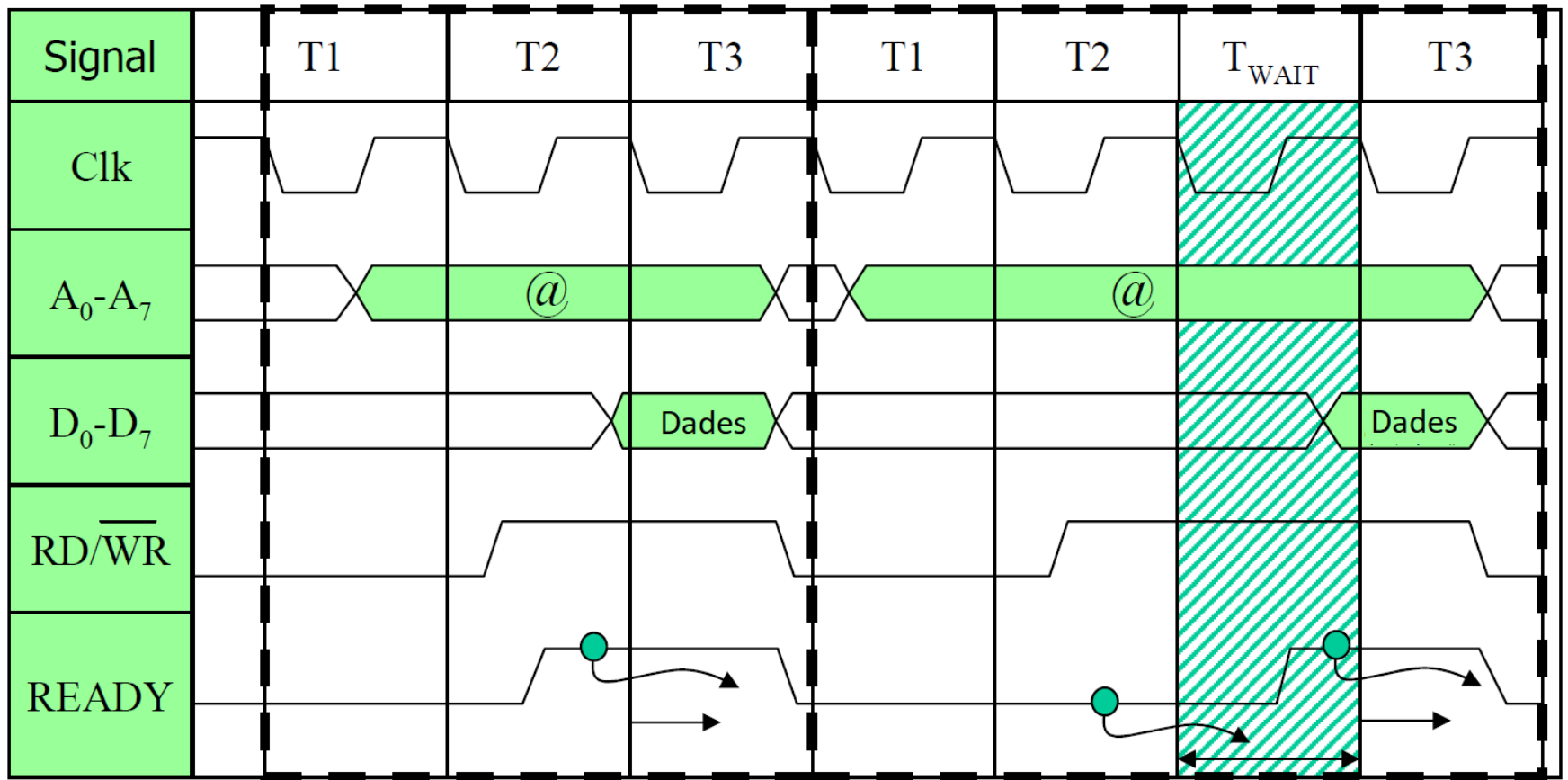
Bus síncron

- Que es pot fer per augmentar l'ample de banda del bus:
 - augmentar la freqüència de rellotge
 - problemes amb els dispositius -> exigeix un menor temps resposta
 - problema al bus (biaix del rellotge): assegurar l'estabilització dels senyals (cada senyal té el seu retard de propagació)
 - realitzar transferències en mode ràfega
 - el mestre indica a l'esclau el nombre de bytes a transferir
 - l'esclau, en lloc de tornar un byte al bus, col·loca al bus un byte en cada cicle fins arribar al nombre de bytes indicat
 - per l'exemple anterior, el temps per llegir un bloc de n bytes serà de $n + 2$ en lloc de $3n$

Bus semi síncron

- Les mateixes característiques que el bus síncron, però afegeix un nou senyal: READY (o també BUSY o WAIT)
- Quan un dispositiu no pot realitzar el seu treball en el temps prefixat, senyala aquesta situació al mestre amb la NO activació del senyal READY
 - es dediquen més cicles: cicles d'espera (wait state)
- Quan les dades estan disponibles, l'esclau activa el senyal READY
- El senyal READY permet ajustar el protocol a la velocitat de cada dispositiu. El nombre de cicles d'espera afegits depèn de el dispositiu
- Els protocols amb senyal de rellotge no aconsegueixen aprofitar totes les possibilitats del dispositiu.
 - En l'exemple anterior, una operació que necessita 550ns amb un protocol amb 250ns de temps de cicle, utilitzarà 3 cicles de rellotge (750ns)
 - pèrdua de rendiment

Bus semi síncron



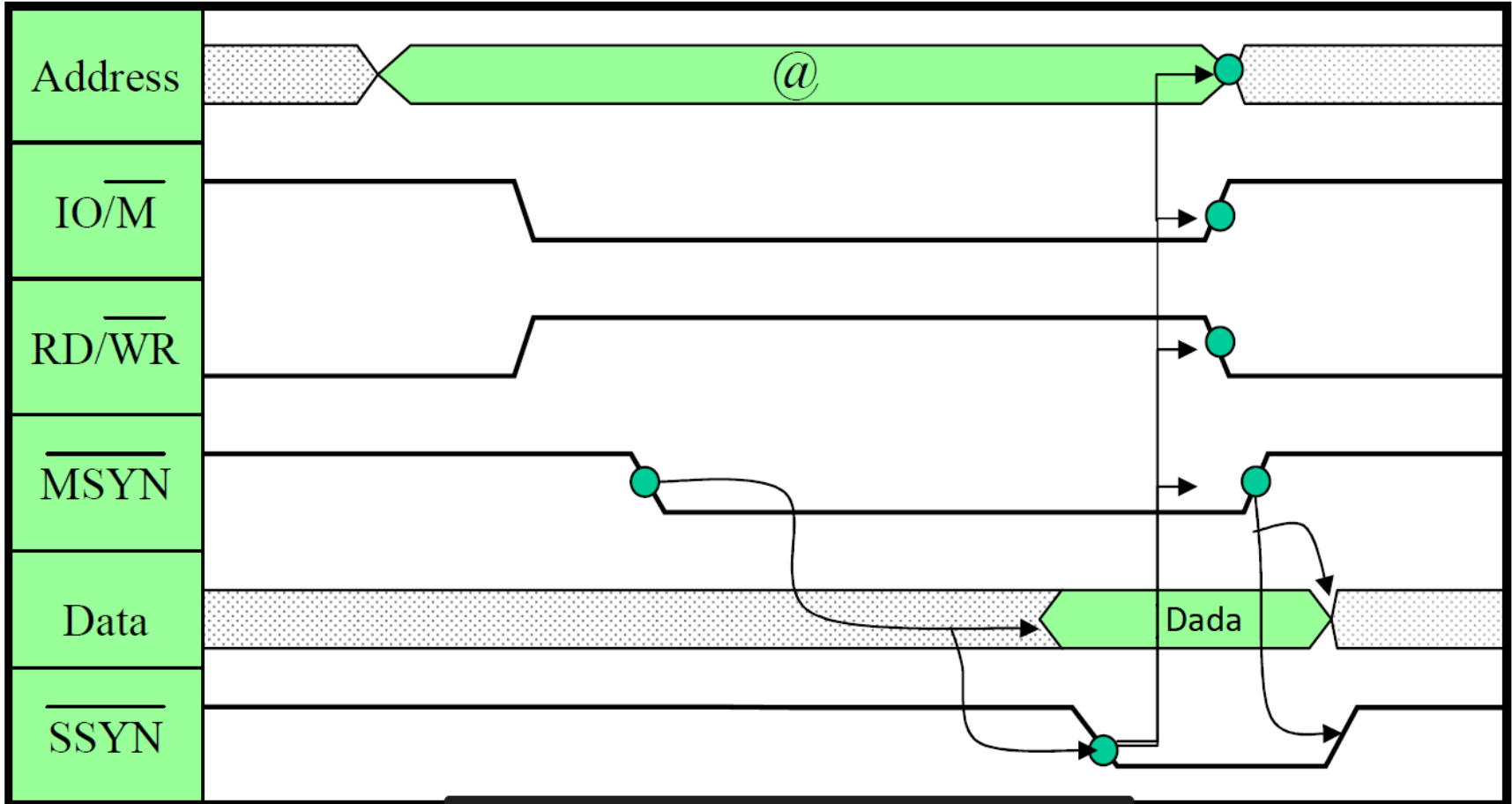
Sense cicles d'espera

Amb cicles d'espera

Bus asíncron

- No hi ha senyal de rellotge al bus. La comunicació entre dispositius es realitza mitjançant una "conversa" (handshake) entre tots dos, a partir de dos nous senyals: **MSYN** (màster synchronization) i **SSYN** (slave synchronization)
 - El mestre activa els senyals de direcció i control (p.e., lectura mem.)
 - Després d'un breu interval d'estabilització dels senyals, activa el senyal MSYN, indicant la presència de senyals de direcció i control vàlides
 - Després d'un temps no determinat a priori, l'esclau proporciona les dades i activa el senyal SSYN per indicar que les dades estan a punt
 - El mestre recull les dades i desactiva MSYN
 - L'esclau desactiva SSYN

Bus asíncron

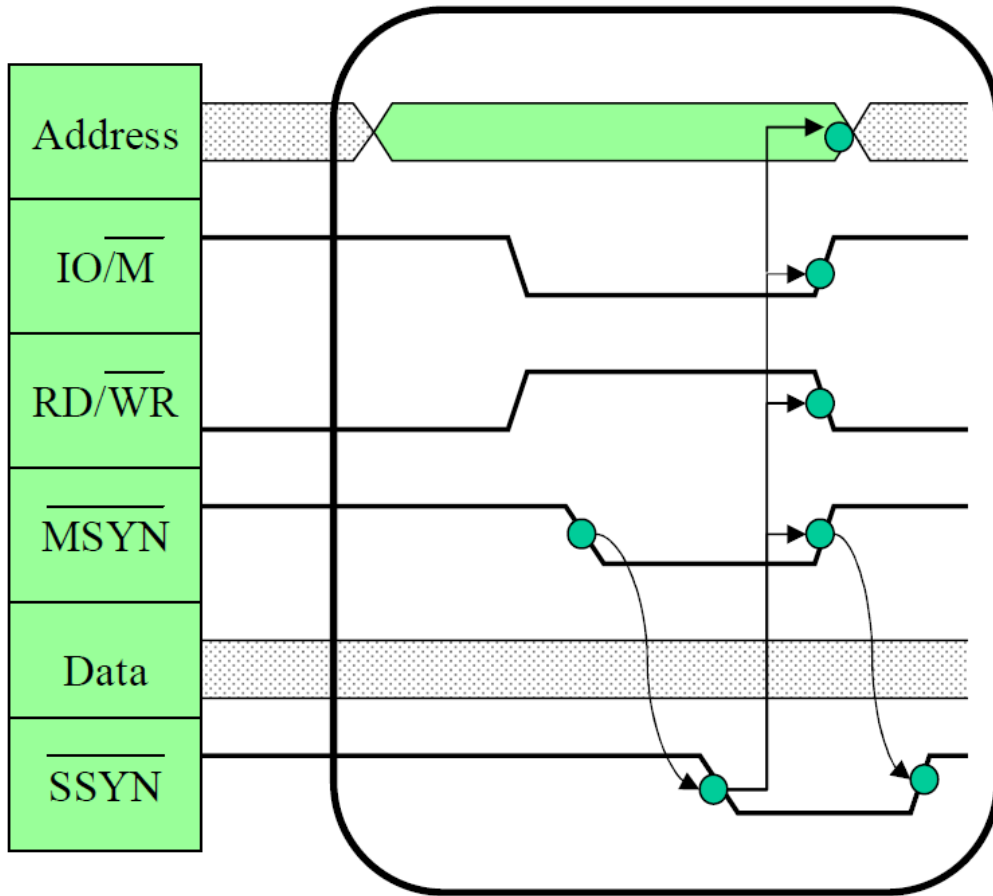




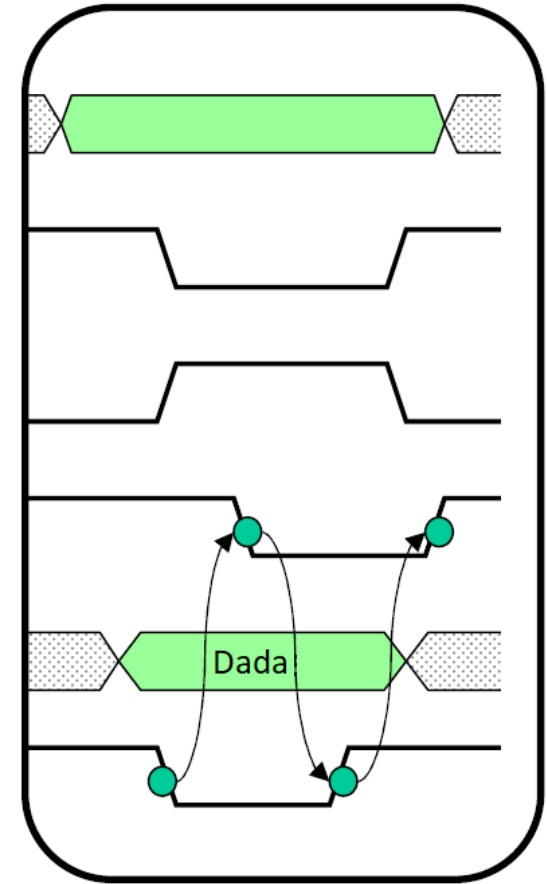
Bus asíncron de cicle partir

- Es distingeixen dues fases: la petició del mestre i la resposta de l'esclau
 - el temps intermedi entre les dues fases s'utilitza per a una altra transferència
- Senyals MSYN i SSYN (bus asíncron)
- Primera fase (per exemple, lectura de memòria):
 - El mestre envia l'adreça i activa els senyals IO/M, RD/WR, a més d'enviar el seu identificador
 - Després de l'interval d'estabilització dels senyals, activa el senyal MSYN
 - L'esclau activa SSYN, de manera que el mestre desactiva les senyals i es desconnecta. Finalment, l'esclau desactiva SSYN
- Segona fase: ara l'esclau anterior és el mestre del bus
 - Quan l'esclau té les dades llestos, s'inicia la transferència al bus: col·loca les dades al bus, col·loca l'identificador del mestre al bus i activa el senyal SSYN
 - El mestre recull les dades i activa el senyal MSYN
 - L'esclau desactiva SSYN, de manera que el mestre desactiva MSYN

Bus asíncron de cycle partir



Petició del mestre



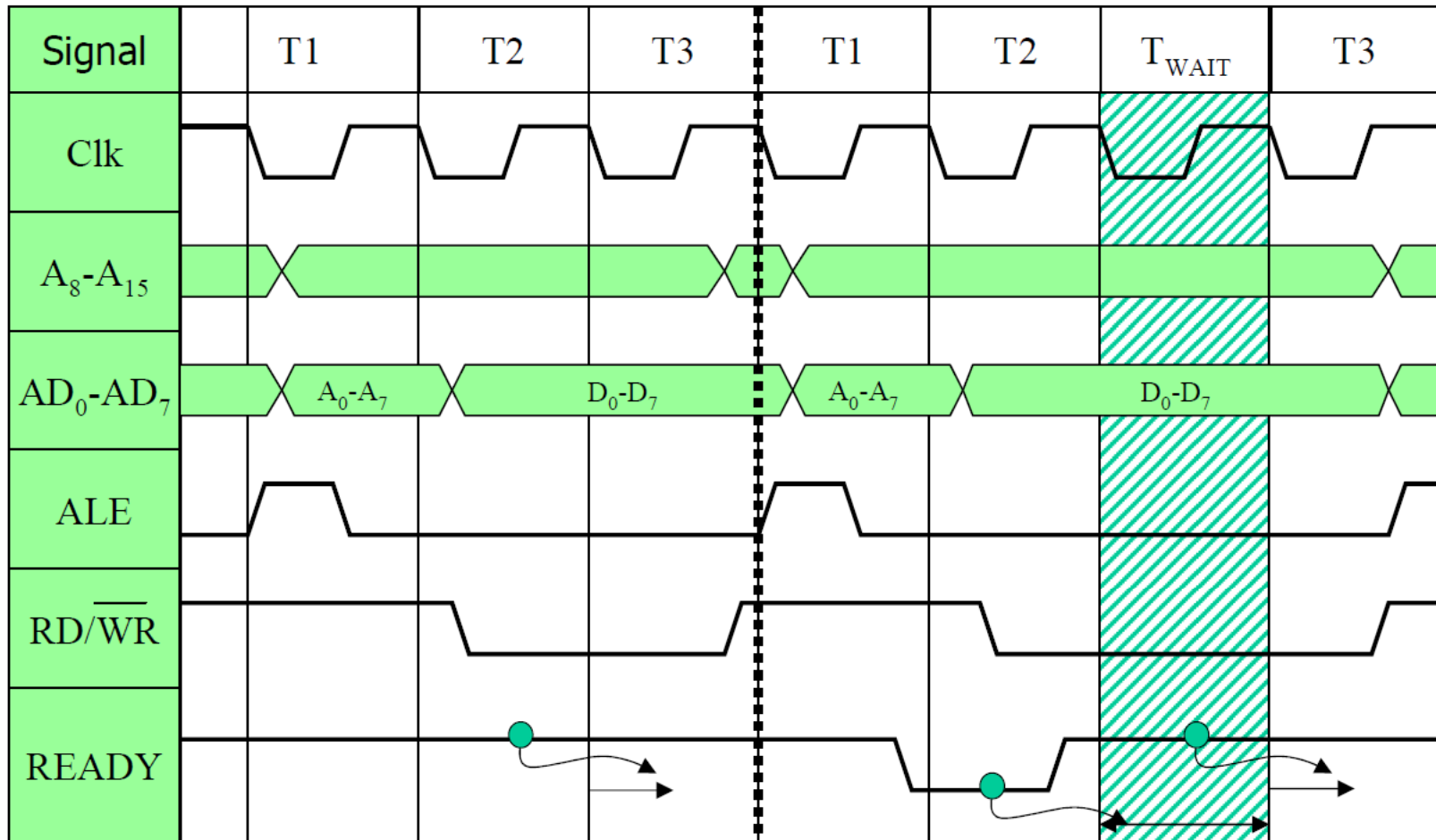
Resposta de l'esclau

Multiplexació de busos

- S'utilitzen les mateixes línies per enviar més d'un tipus de senyal: per exemple, adreces i dades
 - les línies de bus es multiplexen en el temps
- Nou senyal: ALE (Address Latch Enable), que indica el que en instant s'envia per les línies multiplexades
- La multiplexació de les línies de dades o direcció té com a objectiu augmentar les prestacions del bus amb poc cost:
 - Augmentar l'espai d'adreçament: si es multiplexen les línies de dades, es poden utilitzar aquestes línies per augmentar el nombre de bits per a les adreces
 - Augmentar l'ample de banda: si es multiplexen les línies de direcció, es poden utilitzar aquestes línies per a transferir dades simultàniament
- El maquinari és més senzill, però el protocol és més lent (no es pot enviar tota la informació en paral·lel)

Multiplexació de busos

Bus semisíncon multiplexat



Sense cicles d'espera

Amb cicles d'espera

Arbitratge del bus

Si hi ha diversos dispositius mestres en un bus:

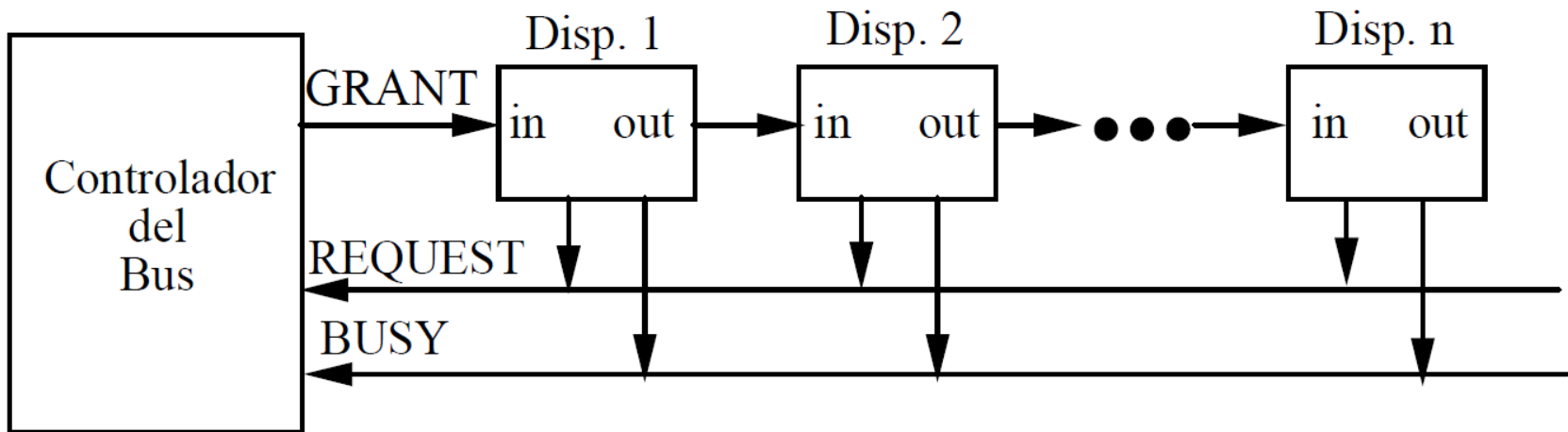
- Quin mestre pot utilitzar el bus en cas de peticions simultànies?
- Com es gestionen les prioritats en l'accés a l'autobús?

Tots els protocols treballen bàsicament amb 3 senyals:

- Petició de bus (Bus Request): l'activa el dispositiu que vol fer l'accés al bus
- Concessió de bus (Bus Grant): senyal que envia l'àrbitre del bus al dispositiu per indicar-li que té concedit l'ús del bus
- Bus ocupat (Busy): quan el dispositiu detecta que l'hi ha donat el control del bus activa aquest senyal per agafar-lo

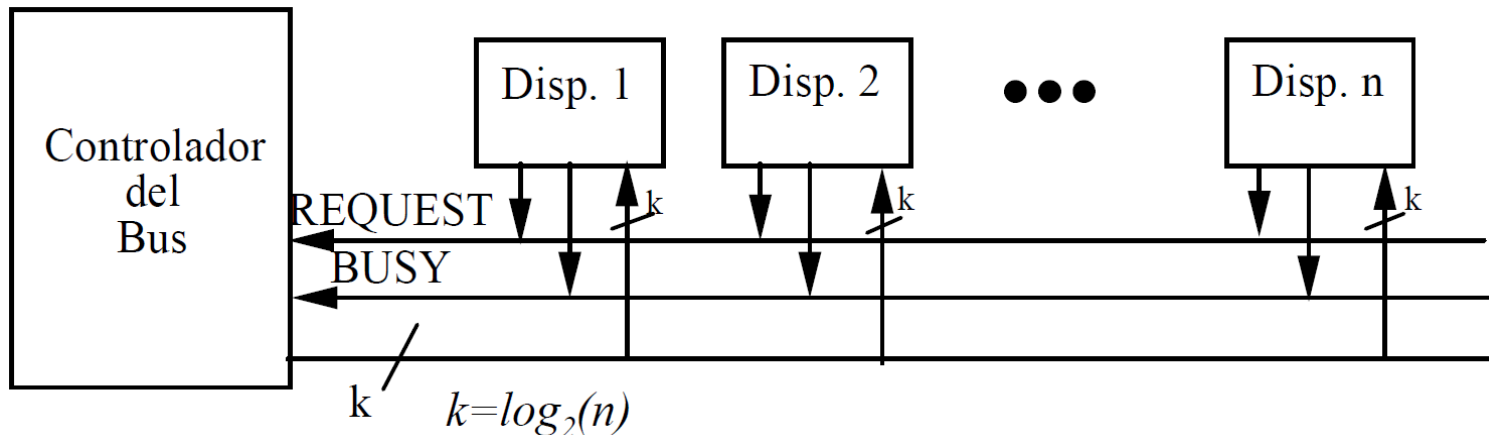
Arbitratge del bus (Daisy-Chain)

- El dispositiu activa Bus Request → l'àrbitre activa Bus Grant
- El senyal Bus Grant arriba a el primer dispositiu (in), que:
 - si no s'ha fet la petició, la passa a el següent (out)
 - si ha realitzat la petició, talla la propagació del senyal Bus Grant i activa el senyal Busy
- A l'acabar la transferència, desactiva el senyal Busy i allibera el bus
- Prioritat dels dispositius fixa: per proximitat a l'àrbitre del bus



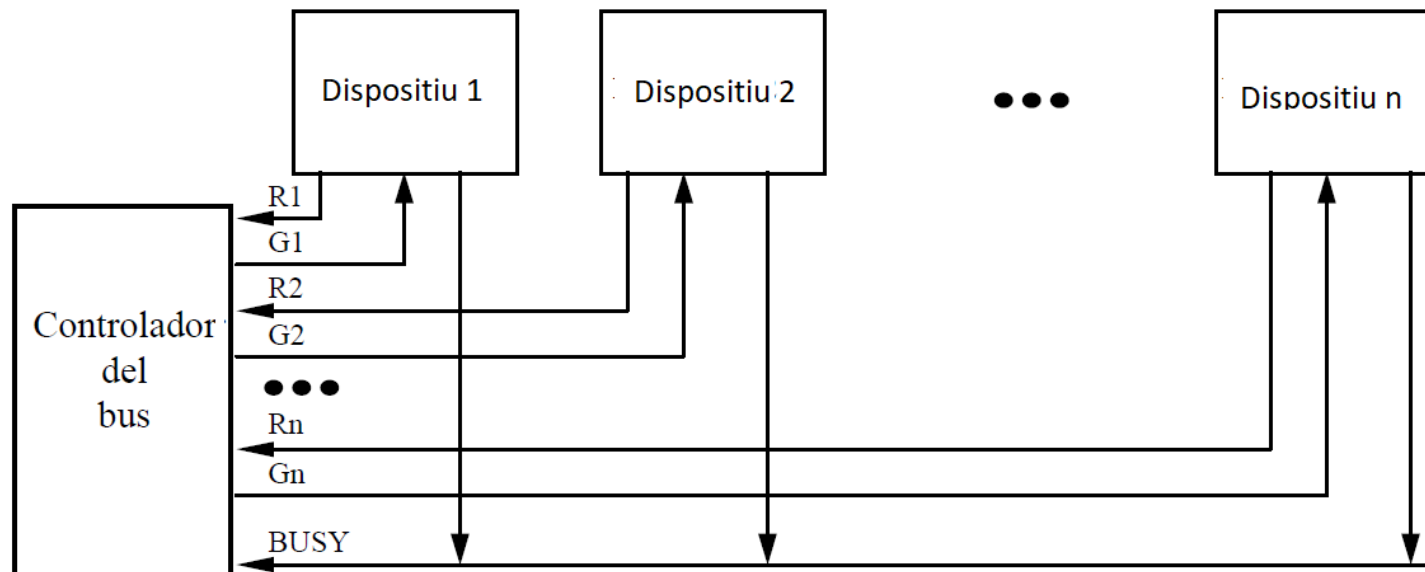
Arbitratge del bus (Enquesta)

- Quan el controlador detecta una petició (Bus Request), realitza una enquesta per determinar el dispositiu
 - envia el codi de cada dispositiu per mitjà de k senyals
 - quan el dispositiu que ha fet la petició detecta el seu codi, activa Busy. S'interromp l'enquesta
- El controlador pot iniciar una nova enquesta a partir de l'últim dispositiu atès (round robin) o des del principi
- Prioritat: depèn de l'ordre d'enquesta (pot ser dinàmica)



Arbitratge del bus (Peticions independents)

- Cada dispositiu té la seva línia de petició (R_i) i d'atenció (G_i), però tots utilitzen la mateixa línia Busy
- Un àrbitre centralitzat decideix a qui concedeix el control del bus d'acord a una determinada política de prioritats
- Avantatge: mètode ràpid. Exemple de bus: PCI
- Desavantatge: més línies en el bus (R_i, G_i)



Arbitratge del bus (Distribuit)

- Per autoselecció (exemple, SCSI):
 - A l'igual que en el mètode anterior, els dispositius sol·liciten l'ús del bus de forma independent
 - Els propis dispositius que demanen l'accés al bus determinen a qui se li concedeix el bus
 - Cada dispositiu posa al bus seu identificador
 - Els dispositius consulten els identificadors per determinar quin és el dispositiu més prioritari entre els que han fet la petició d'ús
- Per detecció de col·lisió (exemple, Ethernet):
 - També se sol·licita el bus de forma independent
 - Quan un dispositiu vol usar el bus, comprova si està ocupat; si no, comença la transmissió
 - Donat el retard en la propagació dels senyals, un altre pot "escoltar" el bus i no detectar que està ocupat comença la seva transmissió COL·LISIÓ
 - El dispositiu que transmet es queda escoltant i comprova si hi ha col·lisió quan detecta que el que transmet no coincideix amb el que escolta.
 - Si hi ha col·lisió, els dispositius implicats interrompen la transmissió i esperen un temps aleatori abans d'intentar de nou
 - repercussions: el nombre de dispositius ha de ser baix per no degradar el rendiment

Múltiples dels Bytes

Prefix del SI (SI)			Prefix binari (IEC 60027-2)		
Nom	Símbol	Múltiple	Nom	Símbol	Múltiple
kilobyte	kB	10^3 (o 2^{10})	kibibyte	KiB	2^{10}
megabyte	MB	10^6 (o 2^{20})	<u>mebibyte</u>	MiB	2^{20}
gigabyte	GB	10^9 (o 2^{30})	gibibyte	GiB	2^{30}
terabyte	TB	10^{12} (o 2^{40})	tebibyte	TiB	2^{40}
petabyte	PB	10^{15} (o 2^{50})	pebibyte	PiB	2^{50}
exabyte	EB	10^{18} (o 2^{60})	exbibyte	EiB	2^{60}
zettabyte	ZB	10^{21} (o 2^{70})	zebibyte	ZiB	2^{70}
yottabyte	YB	10^{24} (o 2^{80})	yobibyte	YiB	2^{80}

Donada la confusió que suposen aquestes dues possibilitats, el desembre de 1998 la IEC (International Electrotechnical Commission) va definir el que va anomenar com a prefixos binaris per eliminar aquesta ambigüitat. Amb aquests prefixos, la segona forma, 2^{20} , passaria a anomenar-se mebibyte.

- Es disposa d'un bus de dades i d'adreces de 32 bits, amb una temporització:
 - El primer cicle es posa l'adreça (i la dada si es tracta d'una escriptura)
 - El segon: latència d'un cicle.
 - El quart cicle: es realitza la lectura o es fa efectiva l'escriptura.
- A més a més, la memòria suporta el mode ràfega (es pot continuar llegint o escrivint fins un màxim de quatre paraules de 32 bits) sense haver d'iniciar una nova operació.

1. Quin és l'ampla de banda teòric del bus si funciona a 200 MHz?

En un cicle es poden transmetre 32 bits = 4 bytes. $f = 200 \text{ MHz} \rightarrow 200 \cdot 10^6 \text{ cicles} \rightarrow AB_T = 4 \times 200 \cdot 10^6 \text{ B/s} = 800 \text{ MB/s} = 762,939 \text{ MiB/s}$.

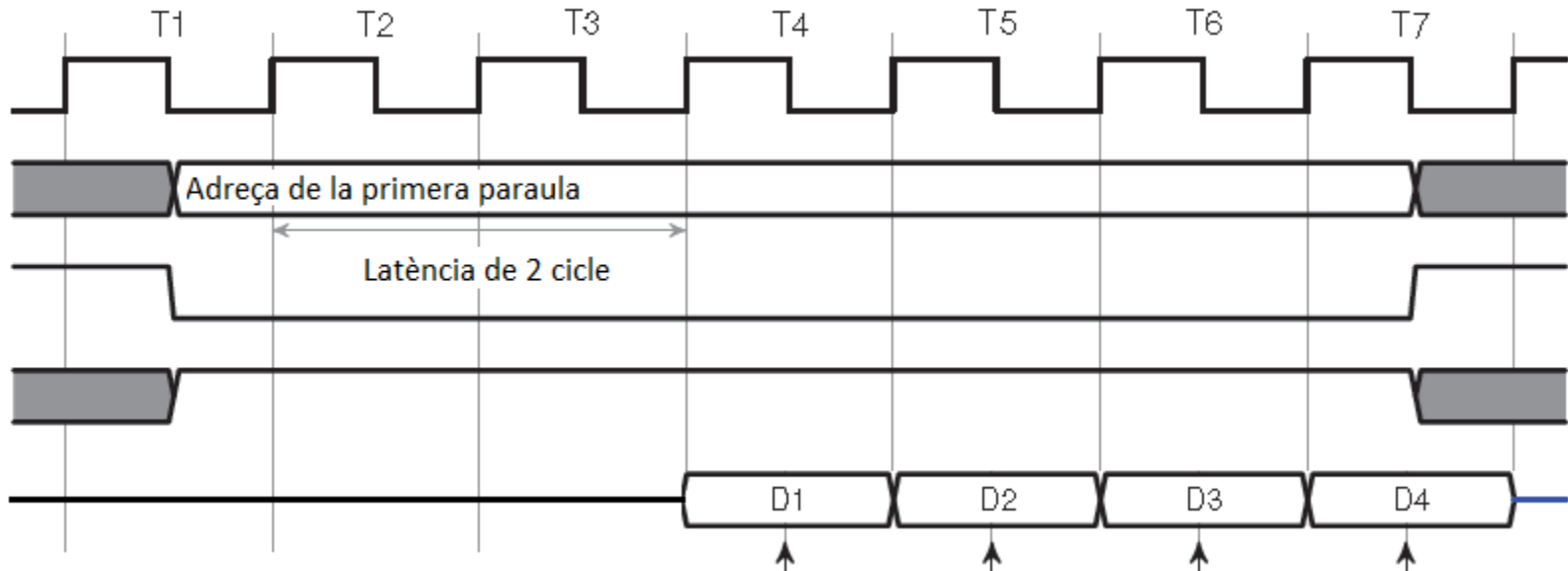
1. Quin és l'ampla de banda efectiu si no es considera el mode ràfega?

Cada transferència necessita 4 cicles (Adreça + latència de 2 + Operació)

Es transmeten 4 bytes en 4 cicles

$AB_E = (4 \text{ bytes}/4 \text{ cicles}) \times 200 \cdot 10^6 \text{ cicles/s} = 200 \cdot 10^6 \text{ B/s} = 200 \text{ MB/s} = 190,7348 \text{ MiB/s}$

3. Quina és l'ampla de banda efectiu si es considera el mode ràfega?



En una operació de 7 cicles es transmeten quatre paraules (16 bytes).

$$AB_E = 16 \text{ bytes} / 7 \text{ cicles} \times 200 \cdot 10^6 \text{ Hz/s} = 457,14 \cdot 10^6 \text{ Bytes/s} = 457,14 \text{ MB/s} = 435,96 \text{ MiB/s}$$

No arriba a l'amplada de banda teòric (800 MB/s), però dobla el valor del cas anterior.