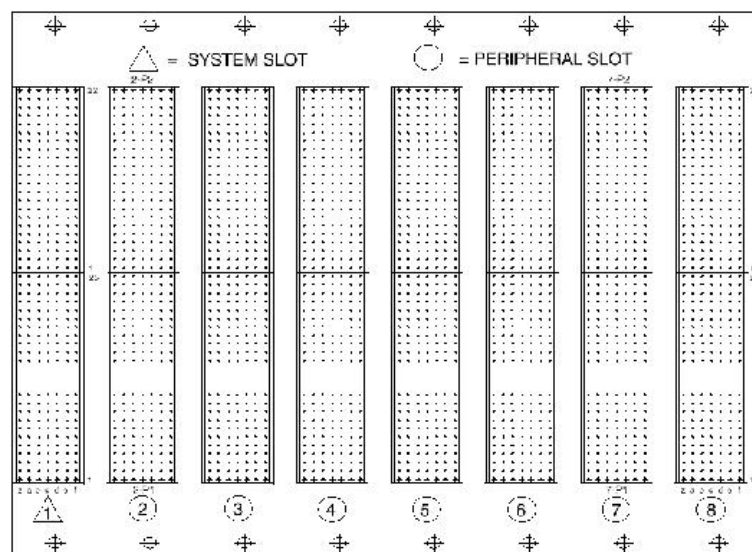


interconnexions

communication entre les différentes unités fonctionnelles



plan

1. bus : définition, structure et caractéristiques
2. synchronisation des échanges
3. techniques d'arbitrage
4. exemple du bus PCI

rappel

3 unités fonctionnelles

la mémoire

lecture, écriture d'un mot

l'unité d'entrées-sorties

à une adresse

le processeur

contrôle le système

elles commmunicquent les unes avec les autres

structure d'interconnexion: l'ensemble des chemins les connectant.

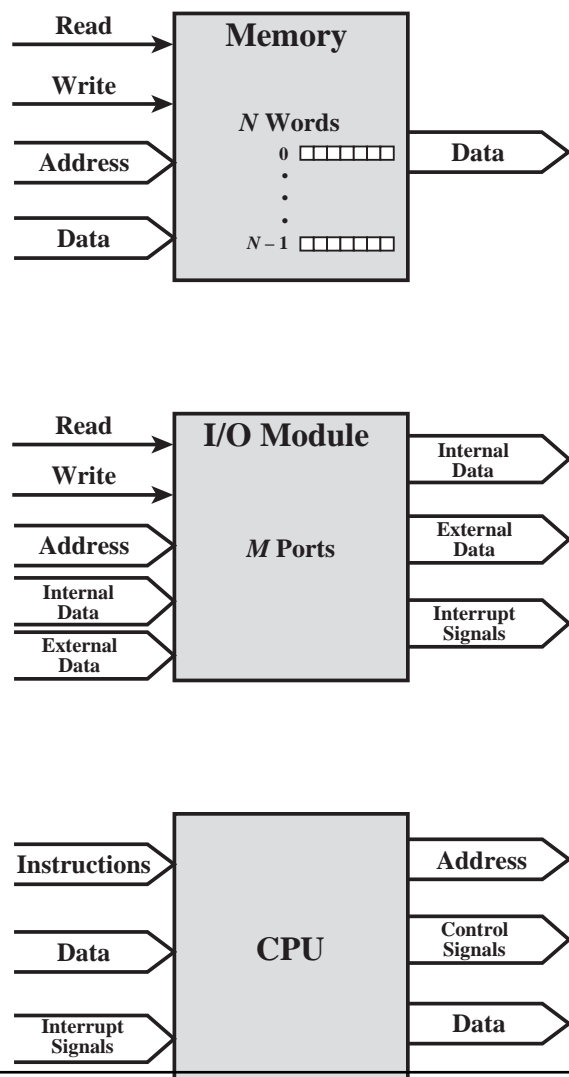


Figure 3.15 Computer Modules

type de transfert

mémoire vers processeur	lecture d'instructions
processeur vers mémoire	écriture de données
E/S vers processeur	lecture de données
	transmises par un périphériques
processeur vers E/S	transfert de données
	vers un périphérique
E/S vers mémoire	accès direct à la mémoire (DMA)
mémoire vers E/S	communication sans passer
	par le processeur
E/S vers E/S	ex. acquisition

bus

Définition : chemin *partagé* entre plusieurs unités

un seul équipement transmet à un instant donné

structure

50 à 100 lignes transmettant des signaux
chaque ligne possède une fonction propre

- les lignes de données (bus de données)
- les lignes d'adresses (bus d'adresse)
- les lignes de contrôle (bus de contrôle)

groupes de fonctions

- bus de données pour les mots de données
- bus d'adresse pour les adresses
- bus de contrôle pour l'accès et l'utilisation
 - des signaux de timing
 - validité des informations d'adresse ou de données
 - signaux de commandes
 - type d'opération à effectuer

signaux de contrôle

les plus courants

memory write écriture de la donnée sur le bus de données
à l'adresse mémoire du bus d'adresse

memory read lecture à l'adresse mémoire
indiquée sur le bus d'adresse

I/O write écriture de la donnée sur le bus de données
sur le port dont l'adresse est indiquée
sur le bus d'adresse

I/O read lecture sur le port dont l'adresse
est indiquée sur le bus d'adresse

signaux de contrôle

data ready	les données sont prêtes
data acknowledge	les données ont été reçues
bus busy	le bus est occupé
bus request	l'accès au bus est demandé
bus grant	l'accès au bus a été obtenu
clock	synchronise les transferts
reset	réinitialisation de tous les modules
...	des signaux gérant les interruptions

Fonctionnement schématique

Une transaction typique se compose de 3 parties

1. l'obtention du bus
2. l'envoi d'une adresse
3. l'envoi des données

type de transaction

transaction de type écriture

- obtention du bus
- transmission de données

transaction de type lecture

- obtention du bus
- requête au module destination
- attente des données

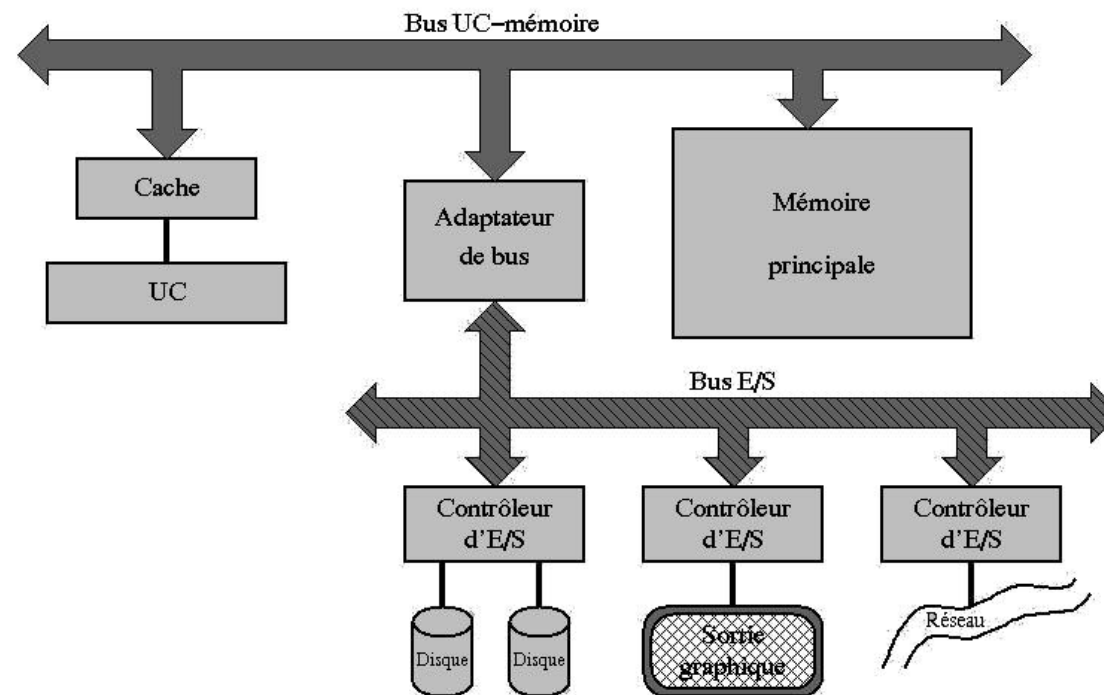
bus = goulot d'étranglement

plus il y a d'équipement connectés à un bus

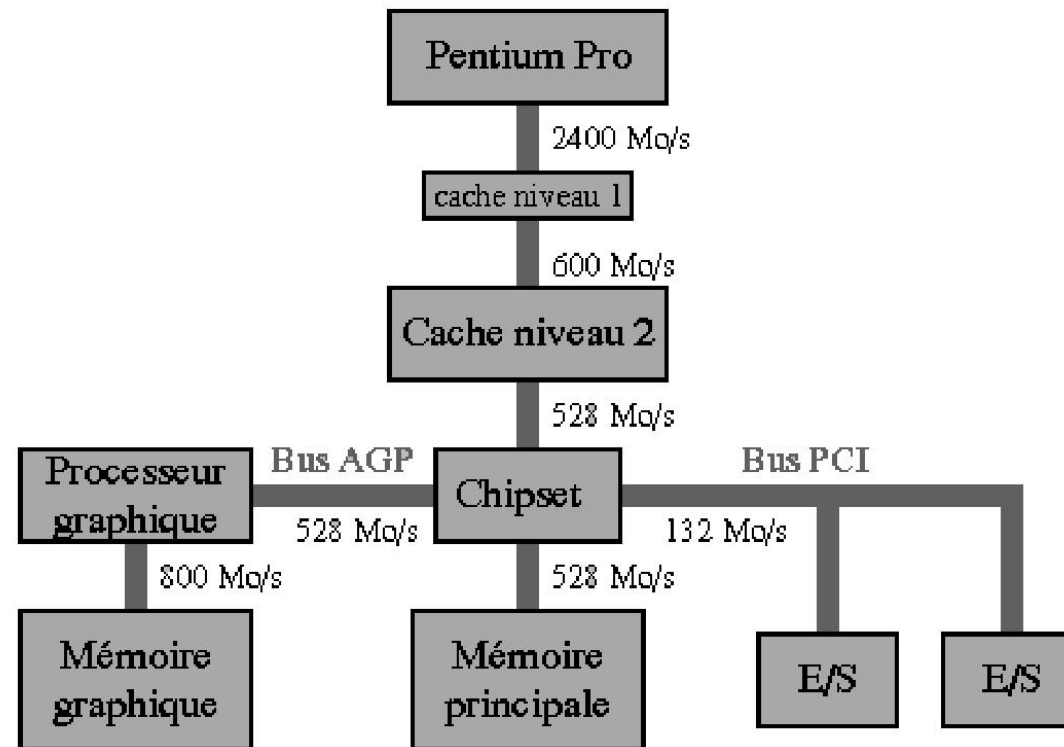
plus le bus est physiquement long

plus les performances décroissent

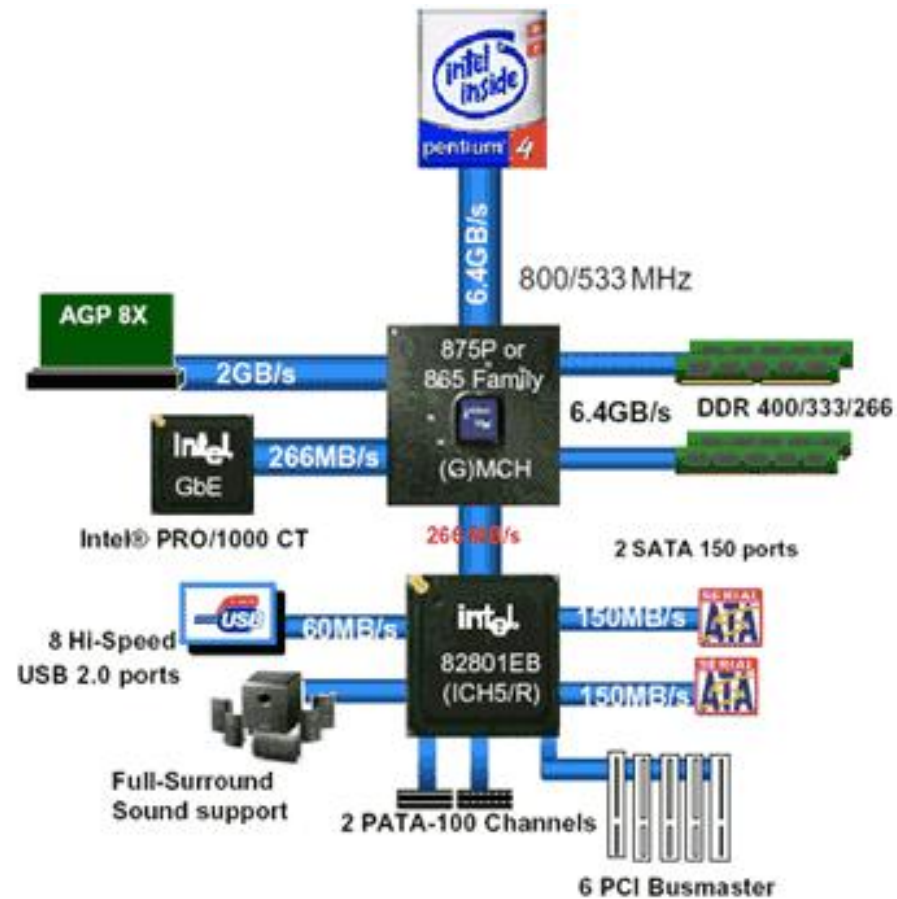
hiérarchie de bus



hiérarchie de bus



hiérarchie de bus



hiérarchie de bus

séparer communication processeur/mémoire de
communication E/S

étager en fonction de la performance des unités

2 familles de bus

- les bus UC-mémoire (bus système)
 - courts
 - rapides
 - les unités connectées sont connues dès la conception
- les bus E/S (bus d'extension)
 - plus long
 - moins rapides
 - offrant une gamme étendue de débits
 - souvent l'objet de normalisation

2 familles de bus

- les bus système

 - les composants conduisent la conception du bus

- les bus E/S

 - le bus dicte la conception des composants

Caractéristiques

- largeur de bus
- type de transfert
- synchronisation
- arbitrage
- performance

Largeur du bus

nombre d'information pouvant être envoyées en parallèle
plus la largeur est élevée, plus le débit est grand

elle influe sur

- le multiplexage des lignes données/adresse
- le nombre d'accès mémoire
- la taille de la capacité mémorielle

déterminante pour les performances du système

largeur et bus de données

bus peu large = multiplexage des données et adresses

largeur et bus de données

bus peu large = multiplexage des données et adresses

bus de données de 8 bits et instructions sur 16 bits

deux accès mémoire pour lire une instruction

largeur et bus de données

bus peu large = multiplexage des données et adresses

bus de données de 8 bits et instructions sur 16 bits

deux accès mémoire pour lire une instruction

Typiquement

- poids fort : sélection mémoire ou E/S
- poids faible : sélection port ou emplacement mémoire

Type de transfert de données

- écriture multiplexée
 1. adresse placée sur le bus
 2. données placées sur le bus
- lecture multiplexée
 1. l'adresse placée sur le bus
 2. temps d'accès aux données
 3. données placées sur le bus

Type de transfert de données

- lecture/écriture non multiplexée
 adresse et données placées simultanément sur le bus
- lecture-modification-écriture :
 1. lecture à une adresse
 2. écriture à cette adresse
- transfert de bloc de données :
 1. adresse placée sur le bus
 2. données d'adresses consécutives placées sur le bus

Synchronisation

La transmission peut être

- synchrone
- asynchrone

En général

- le bus système fonctionne de manière synchrone
- un bus d'E/S fonctionne de manière asynchrone

communication synchrone

horloge transmise sur les lignes de contrôle

définition d'un protocole pour émettre adresses et données

communication rapide, peu de logique de contrôle

émetteur et récepteur doivent fonctionner à la même fréquence

communication asynchrone

absence de référence à une horloge

échanges de signaux pour indiquer la progression de la communication

communication moins rapide, logique de contrôle plus importante

permet la communication entre composants hétérogènes

Technique d'arbitrage

maître du bus: un composant pouvant démarrer une transaction

esclave: un composant non maître

Il ne peut y avoir qu'un seul maître à la fois

en cas de plusieurs maîtres de bus potentiels un mécanisme d'arbitrage est nécessaire

Une transaction faisant intervenir plusieurs esclaves est appelée diffusion (broadcast).

Performances

définie par les critères suivants

- la bande passante
quantité d'informations échangées par unité de temps
- la latence
temps de réponse du bus à une requête de transfert
- la charge
nombre maximum d'unités pouvant être connectées
- la longueur physique du bus.

Synchronisation des échanges

3 manières différentes d'implanter le timing

- synchrone : les évènements ont lieu à des moments précis dans le temps
- asynchrone : les évènements peuvent avoir lieu à des moments arbitraires
- semi-asynchrone : les évènements peuvent avoir lieu de manière asynchrone lors des différentes phases d'une horloge

Communication synchrone

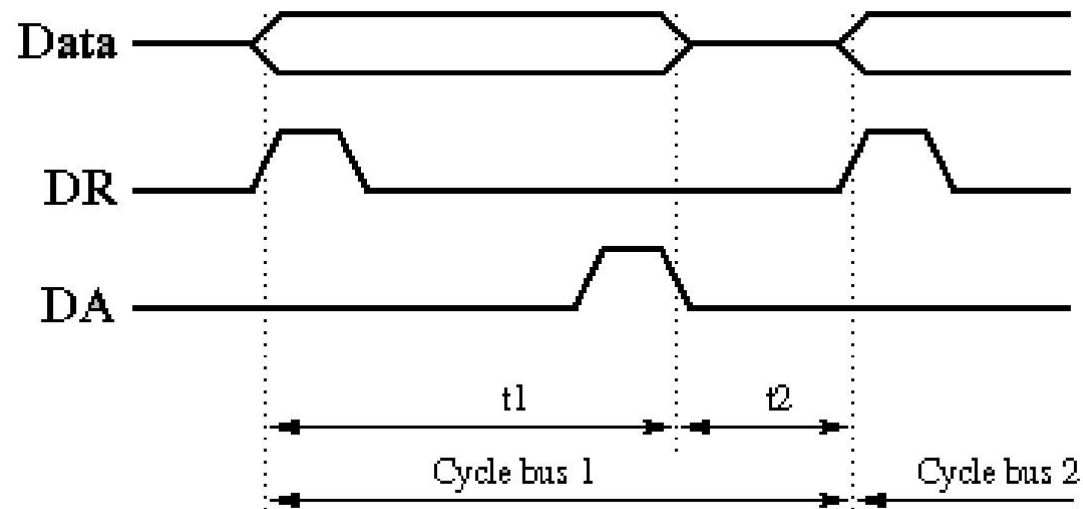
timing des opérations contrôlé par une horloge

pas de dialogue entre émetteur et récepteur pour le contrôle de l'échange

utilisation de 2 signaux générés par l'horloge

nom	utilisé par	les données
DR (Data Ready)	l'émetteur	sont placées sur le bus
DA (Data Acknowledge)	le récepteur	ont été reçues

chronogrammes



doit être adapté au récepteur le plus lent

Communication asynchrone

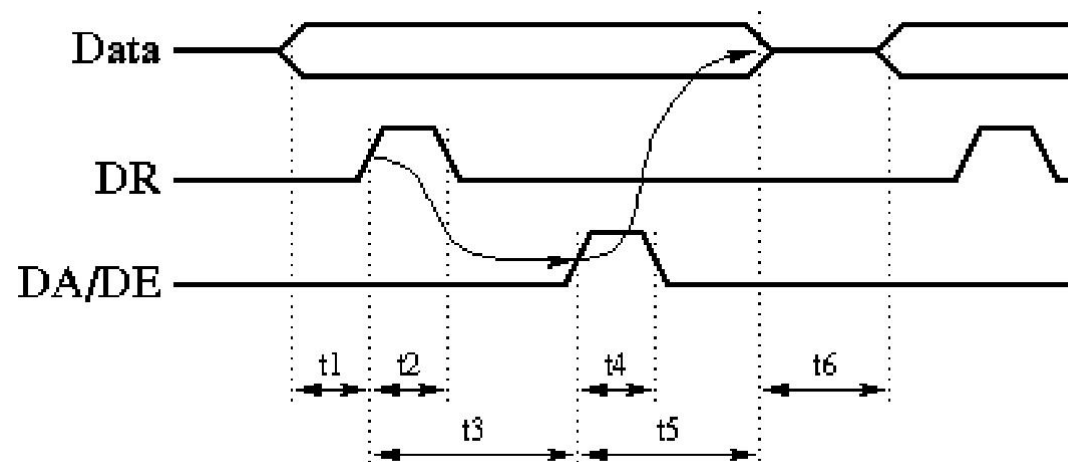
pas d'horloge fixe

signaux générés par les composants

trois type de protocoles asynchrone

- non-entrelacé
- semi-entrelacé
- complètement entrelacé

Transaction asynchrone non-entrelacée



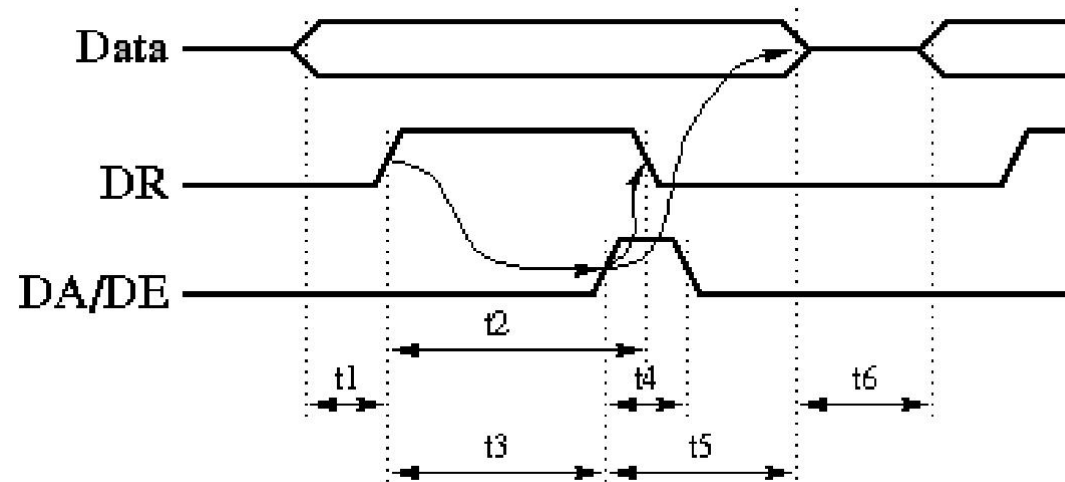
Transaction asynchrone non-entrelacée

supposons que l'émetteur soit très rapide

t_5 est inférieur à t_4

le prochain cycle de bus pourrait commencer avant la retombée du signal DA

Transaction asynchrone semi-entrelacée



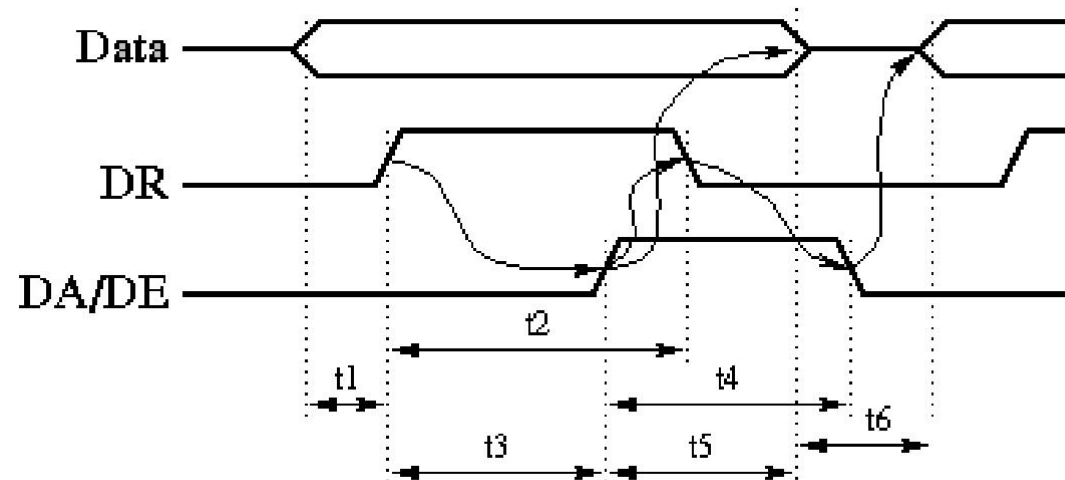
Transaction asynchrone semi-entrelacée

l'émetteur fait retomber le DR en réponse à DA.

problème partiellement résolu

l'émetteur peut commencer une nouvelle transaction trop tôt

Transaction asynchrone complètement entrelacée



L'émetteur est forcé de débiter une nouvelle transaction lorsque DA est retombé

handshake

protocole asynchrone complètement entrelacée

1. l'émetteur place les données sur le bus
2. l'émetteur lève le signal DR
3. le récepteur lit la donnée
4. le récepteur lève le signal DA
5. l'émetteur remet DR à zéro
6. l'émetteur retire sa donnée du bus
7. le récepteur remet DA à zéro

handshake

protocole très souvent utilisé

synchronise des composants de vitesses très différentes

time-out pour prendre en compte les pannes de composant

inconvenients

- communication ralentie par la transmission des signaux de contrôle
- sensible au bruit

communication semi-asynchrone

les transitions des signaux de contrôle ne peuvent avoir lieu qu'à des instants déterminés par une horloge

le temps entre deux transitions successives peut être variable

modes non-entrelacé, semi-entrelacé, et entrelacé

protocole moins sensible au bruit

Techniques d'arbitrage

garantir un seul maître de bus

2 types d'arbitrage

- statique
- dynamique

Arbitrage statique

les candidats deviennent maître à tour de rôle
dans un ordre fixé

inconvenient : le bus peut être inutilisé (non-opération)

avantage : simple à mettre en oeuvre

utilisé avec un protocole synchrone lorsqu'il y a peu de
maîtres potentiels

arbitrage dynamique

allocation du bus

- sur demande
- lorsqu'il est libre
- à un composant qui le demande

utilisation du signal BR (Bus Request)

arbitrage dynamique

comment répartir plusieurs demandeurs?

arbitrage dynamique

comment répartir plusieurs demandeurs ?

- suivant une priorité affectée de manière unique à chaque maître potentiel (bus d'E/S)

arbitrage dynamique

comment répartir plusieurs demandeurs ?

- suivant une priorité affectée de manière unique à chaque maître potentiel (bus d'E/S)
- de manière équitable
éviter qu'un demandeur de petite priorité soit constamment rejeté

arbitrage dynamique

comment répartir plusieurs demandeurs ?

- suivant une priorité affectée de manière unique à chaque maître potentiel (bus d'E/S)
- de manière équitable
éviter qu'un demandeur de petite priorité soit constamment rejeté
- en combinant les deux premières politiques
un choix équitable répartit deux demandes de même priorité

arbitrage dynamique

la libération du bus peut avoir lieu

– en fin de transaction

arbitrage dynamique

la libération du bus peut avoir lieu

- en fin de transaction
- sur demande

le maître conserve le bus jusqu'à une nouvelle demande
(exemple : le processeur qui demande le plus souvent)

arbitrage dynamique

la libération du bus peut avoir lieu

- en fin de transaction
- sur demande

le maître conserve le bus jusqu'à une nouvelle demande
(exemple : le processeur qui demande le plus souvent)

- par préemption

un module prioritaire peut devenir maître avant la fin
d'une transaction

Mécanismes matériels d'arbitres

un mécanisme d'arbitrage peut être

- distribuée
réparti sur l'ensemble des modules connectés au bus

Mécanismes matériels d'arbitres

un mécanisme d'arbitrage peut être

- distribuée
 - réparti sur l'ensemble des modules connectés au bus
- centralisée
 - sur un seul des modules connectés au bus
 - sur un module dédié appelé bus arbitre ou contrôleur de bus

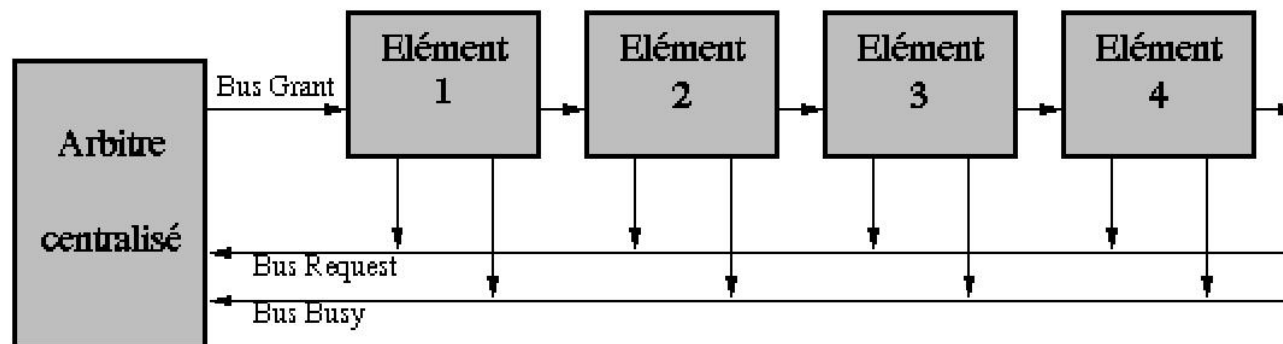
Mécanismes matériels d'arbitres

basée sur trois signaux de contrôle

nom	signale
BR (Bus Request)	demande de bus
BA/BG (Bus Acknowledge/Grant)	obtention du bus
BB (Bus Busy)	occupation du bus

Daisy chain

structure en guirlande



daisy chain

ordre de priorité imposé par la guirlande

1. les demandeurs émettent un BR
2. l'arbitre reçoit le OU logique des BR
3. si BR, l'arbitre envoie le BG à la guirlande
4. le demandeur de plus haute priorité lève BB
fait retomber BG, il devient maître du bus
5. en fin de transaction, le maître fait retomber BB

daisy chain

avantages

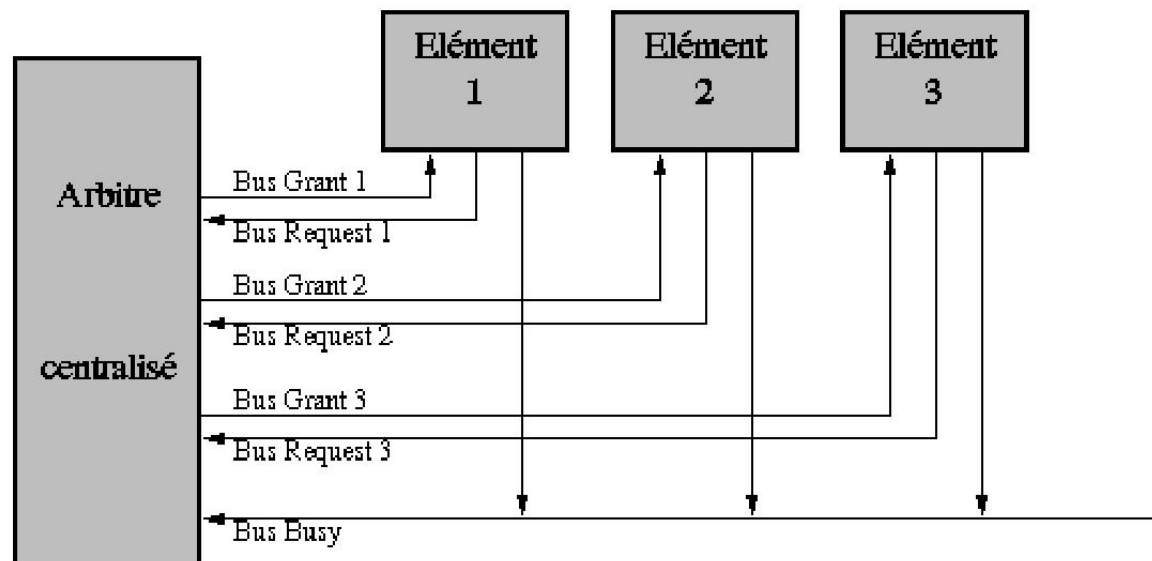
- simplicité de réalisation
- quasi-indépendante du nombre de modules connectés

inconvénients

- priorité statique
- lenteur de réponse de l'arbitre
(proportionnelle à la longueur de la guirlande)
- grande sensibilité aux pannes

Requête-autorisation

Chaque demandeur dispose de lignes BG et BR propre



requête-autorisation

avantages

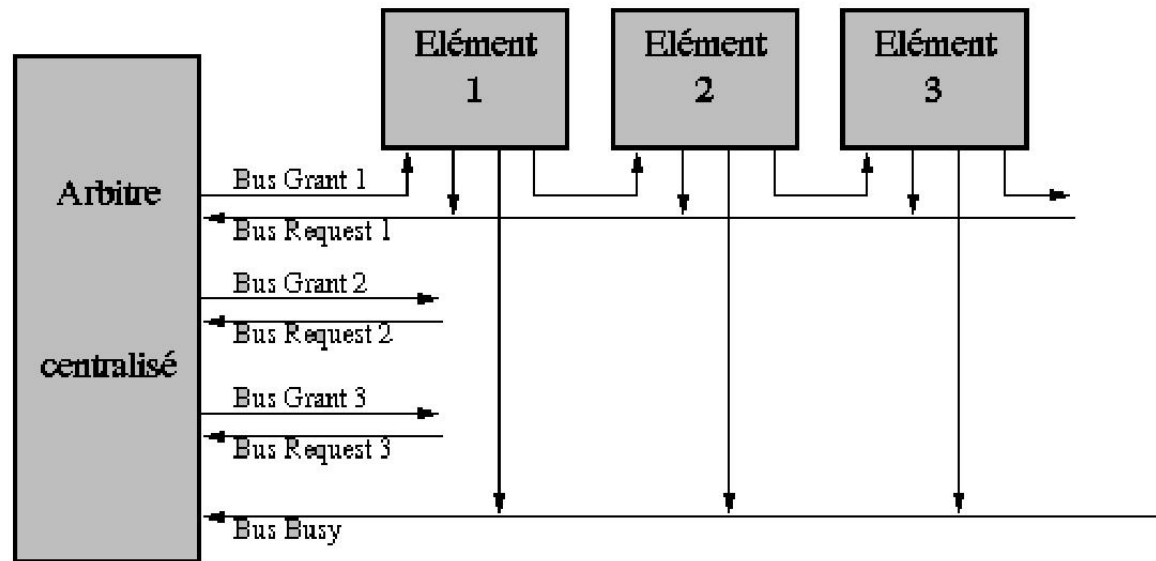
- absence de délai de réponse,
- insensibilité aux pannes d'un module
- priorité non statique

inconvénient

- multiplication des lignes de contrôle

Arbitrage mixte

combinaison des deux techniques précédentes



arbitrage mixte

organisation en séries de composants

- possédant des lignes BG et BR
- organisées en guirlande

arbitrage

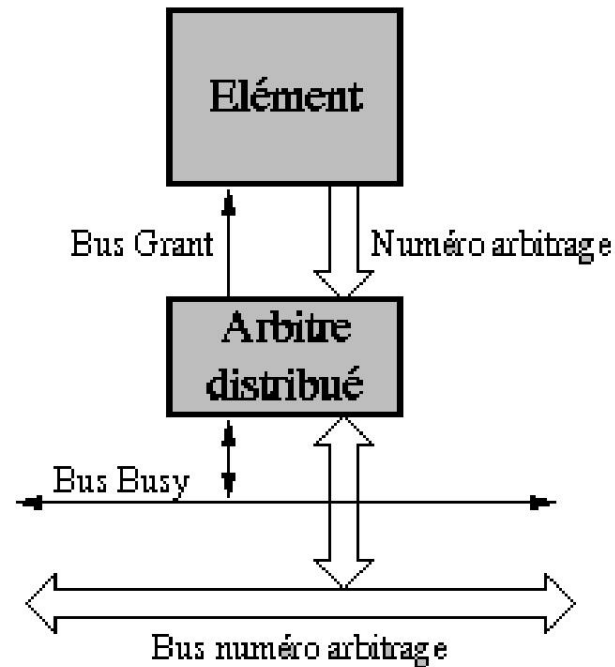
- différentes stratégies peuvent être implantées pour désigner la série
- la guirlande détermine le maître dans la série choisie

arbitrage décentralisé

l'arbitrage centralisé est très sensible aux pannes
un arbitre décentralisé diminue cette sensibilité
chaque module

- possède un circuit d'arbitrage
- possède un numéro de priorité unique
- dialogue avec son propre arbitre

arbitrage décentralisé



utilisé dans les configurations multiprocesseur

arbitrage décentralisé

lorsqu'un composant souhaite avoir le bus

1. il envoie son numéro de priorité à son arbitre
2. l'arbitre place ce numéro sur le bus numéro arbitrage
3. calcul du OU logique de tous les numéros de priorité des demandeurs
4. chaque arbitre compare ce numéro calculé au numéro du composant

arbitrage décentralisé

5. si les deux sont différents, l'arbitre retire son numéro, sinon il le maintient
6. il reste sur les lignes le numéro du composant pouvant être maître
7. l'arbitre signale à son composant qu'il est le nouveau maître (BG) et lève BB

Stratégies d'arbitrage

départager les demandeurs si la priorité n'est pas fixée statiquement

deux notions sont examinées

- le niveau de priorité
- l'ancienneté de la demande

Stratégie linéaire (L)

un numéro fixe est attribué à chaque demandeur

Les numéros sont rangés par ordre de priorité décroissante

Exemple : 4 L 3 L 1 L 2

le module le plus prioritaire est 4

le module le moins prioritaire est 2

simple à réaliser, mais risque de famine

Stratégie circulaire (R)

les numéros sont placés dans une liste circulaire

le demandeur dont le numéro est placé à droite du numéro du dernier maître devient maître à son tour

Exemple : 4 R 3 R 1 R 2

si 2 est maître du bus

le module le plus prioritaire est le module 4

plus difficile à réaliser mais évite la situation de famine

Stratégie cyclique (C)

pour chaque demandeur, on garde l'antériorité des demandes précédentes

l'arbitre applique une stratégie linéaire sur les modules classés par ordre d'antériorité décroissante

Exemple : 1 C 2 C 3 C 4

Stratégie cyclique (C)

si le plus anciennement servi est

- l'élément numéro 4
- puis l'élément numéro 1
- puis l'élément numéro 3
- et enfin l'élément numéro 2

alors l'élément le plus prioritaire sera le 4

Stratégie cyclique (C)

si le plus anciennement servi est

- l'élément numéro 4
- puis l'élément numéro 1
- puis l'élément numéro 3
- et enfin l'élément numéro 2

alors l'élément le plus prioritaire sera le 4

si seul le numéro 3 demande le bus il l'obtiendra

la priorité sera alors par ordre décroissant: 4, 1, 2 et 3

Stratégie multiple

combinaison de plusieurs stratégies

Exemple : 1 R (3 C 4) R (5 L 6)

5 sera le plus prioritaire si 3 ou 4 ont été servis en dernier

Exemple : le bus PCI

Peripheral Component Interconnect

standard développé par intel

utilisé pour les communications avec les E/S

processeur-indépendant, configurable, haut débit

très populaire et très utilisé

caractéristiques

version 2.1 :

largeur	32 ou 64 bits
cadencé à	66 Mhz
débit max	528 Mo/s
transfert	multiplexage adresses et données en rafale
timing	semi-synchrone
arbitrage	centralisé synchrone requête-autorisation caché

Transaction

des lignes de contrôle pour détailler

- type d'opération (lecture/écriture)
- destinataire (mémoire, E/S)

Une transaction consiste en

- une phase d'émission d'une adresse,
- une ou plusieurs phase d'émission de données

synchronisation des évènements sur fronts descendants

chronogrammes

FRAME	durée de la transaction
AD	adresse
C/BE	lecture/écriture
IRDY	initiator ready
TRDY	target ready
DEVSEL	récepteur

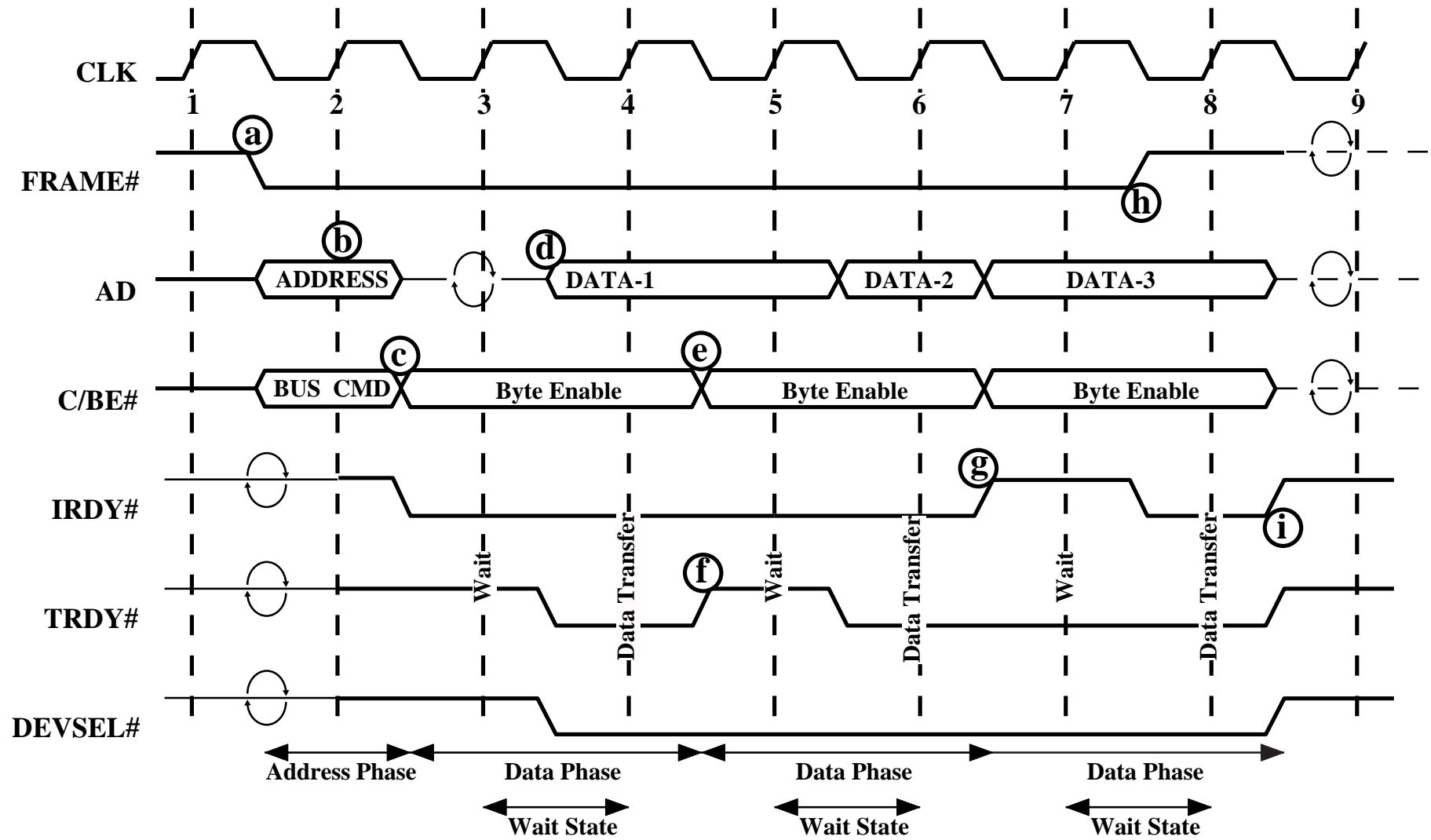


Figure 3.22 PCI Read Operation

Arbitrage

centralisé et reposant sur deux signaux BG et BR pour chaque module

la spécification ne fixe pas de stratégie

l'arbitre comprend plusieurs types de stratégies (FIFO, circulaire, fixation de priorité)

l'arbitrage est effectué pendant une transaction

chronogrammes

REQ	requête
GNT	autorisation
FRAME	
IRDY	
TRDY	cf. lecture
AD	

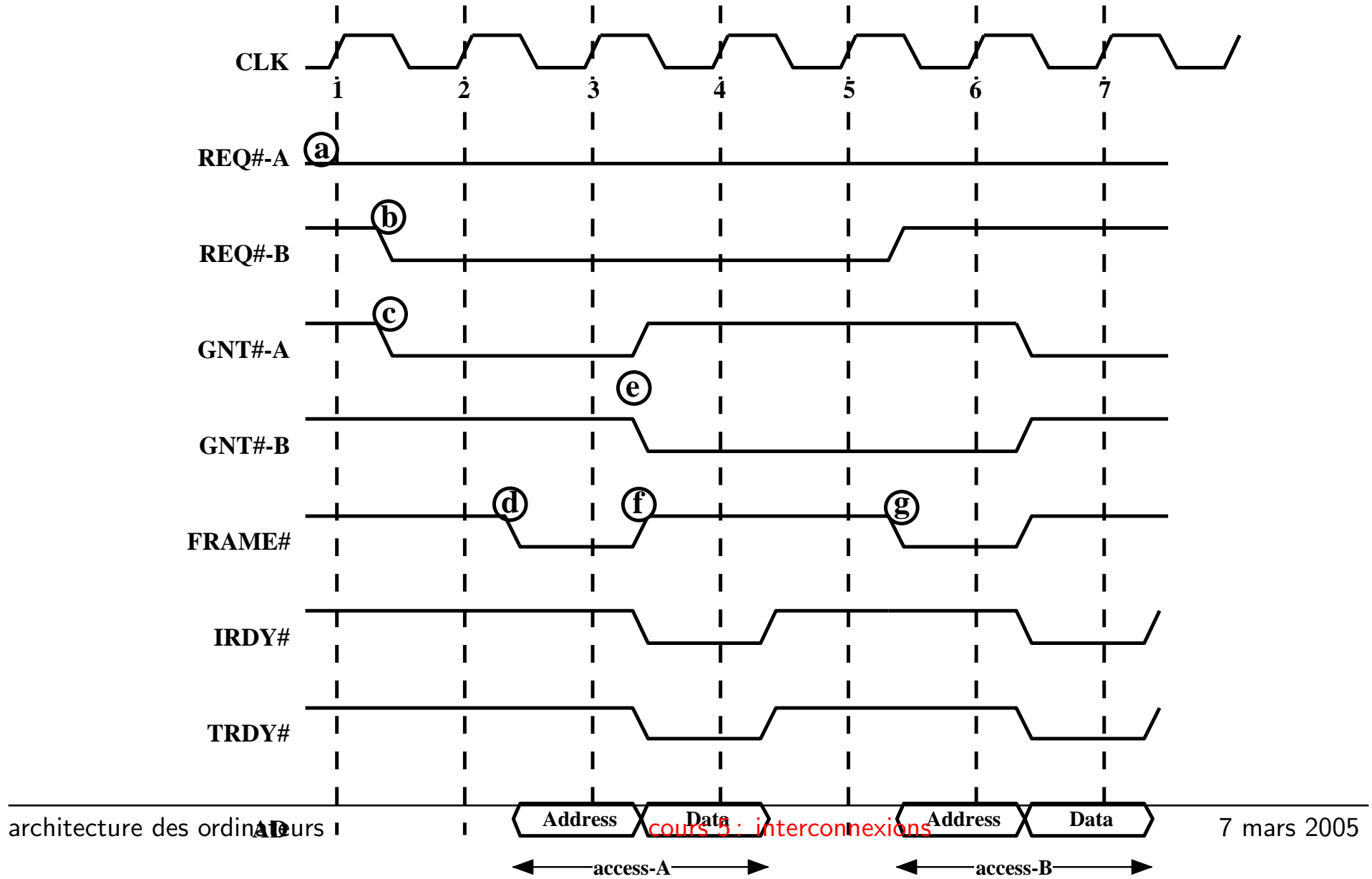


Figure 3.24 PCI Bus Arbitration Between Two Masters