1

Chapitre III : GESTION DES E/S PHYSIQUES

Dr M.CHENAIT (mchenait@usthb.dz)

Dans ce chapitre…

Dr M.CHENAIT
(mchenait@usthb.dz)

1. Introduction.
2. Le matériel.
3. **Les mode de pilotage.**
4. E/S Tamponnées.
5. Traitement d'E/S simultanées.
6. Couches Logicielles d'E/S

Introduction.

Dr M.CHENAIT
(mchenait@usthb.dz)

- ✓ Le titre du chapitre …
- ✓ Le but du chapitre…
- ✓ La relation entre ce chapitre et ces prédécesseurs.

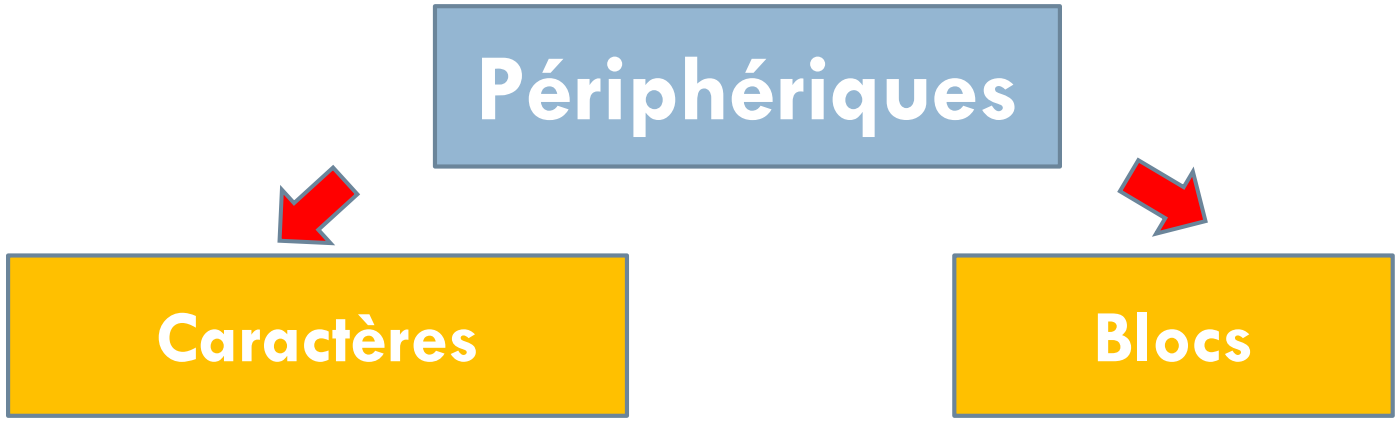
Le matériel

Dr M.CHENAIT
(mchenait@usthb.dz)

- ✓ Périphériques.
- ✓ Contrôleurs (interface, coupleur).
- ✓ Canaux.
- ✓ Bus.

Dr M.CHENAIT
(mchenait@usthb.dz)

Le matériel



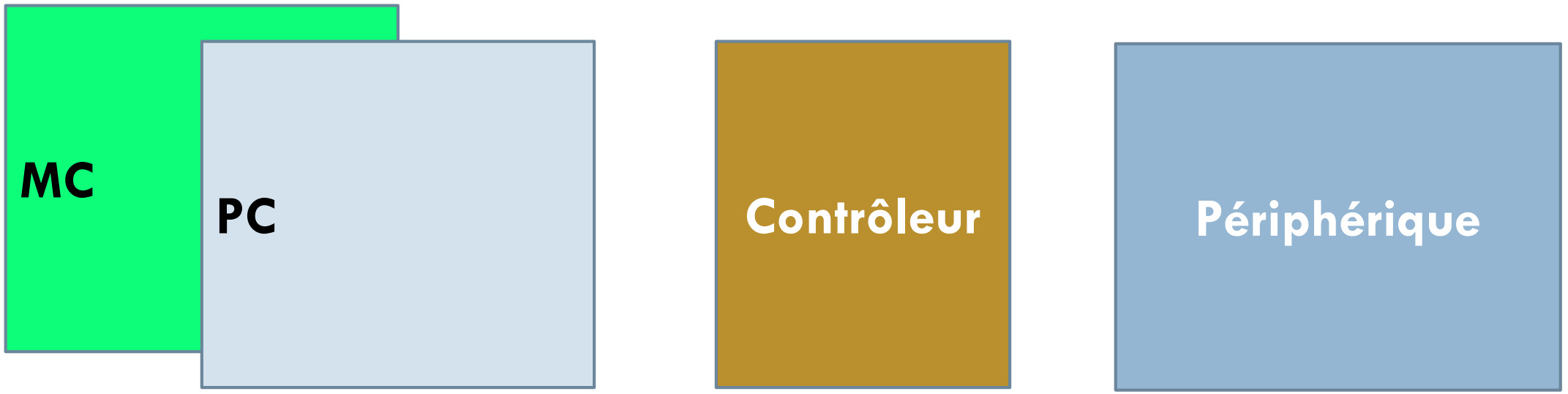
Ils envoient ou reçoivent les données octets par octets.
Exemple :
Le clavier, la souris, les imprimantes, les terminaux
Les données
Les données sont transmises les unes derrière les autres, on parle alors d'**accès séquentiel.**

Ils acceptent les données par blocs de taille fixe, chaque bloc ayant une adresse propre.
Exemple : les disques, la carte vidéo, …etc.
Les données
Le grand avantage par rapport aux périphériques caractères est
qu'il est possible d'aller lire ou écrire un bloc à tout moment, on parle alors d'**accès aléatoire (Random Access).**

Le matériel

Dr M.CHENAIT
(mchenait@usthb.dz)

Un contrôleur (Controller) est une unité spéciale, appelée aussi module d'E/S (I/O module) ou coupleur, qui sert d'interface entre le périphérique et l'UC+MC.



Le matériel

Dr M.CHENAIT
(mchenait@usthb.dz)

Un contrôleur dispose, pour chaque périphérique qu'il gère, de trois (03) types de registres :

- **Registres de données (Data Registers) : destinés à contenir les informations** échangées avec le périphérique. Ils peuvent être lus (entrée) ou écrits (sortie).
- **Registre d'état (State Register) : qui permet de décrire l'état courant du périphérique** (libre, en cours de transfert, erreur détectée,...).
- **Registre de contrôle (Control Register) : qui sert à préciser au coupleur ce qu'il doit** faire, et dans quelles conditions (vitesse, format des échanges,...).

Le matériel

Dr M.CHENAIT
(mchenait@usthb.dz)

Le rôles du contrôleur

- Lire ou écrire des données du périphérique.
- Lire ou écrire des données de l'UC/Mémoire.
- Contrôler le périphérique et lui faire exécuter des séquences de tâches.
- Tester le périphérique et détecter des erreurs.
- Mettre certaines données du périphérique ou de l'UC en mémoire tampon afin d'ajuster les vitesses de communication du contrôleur.

Le matériel

Dr M.CHENAIT
(mchenait@usthb.dz)

Canaux

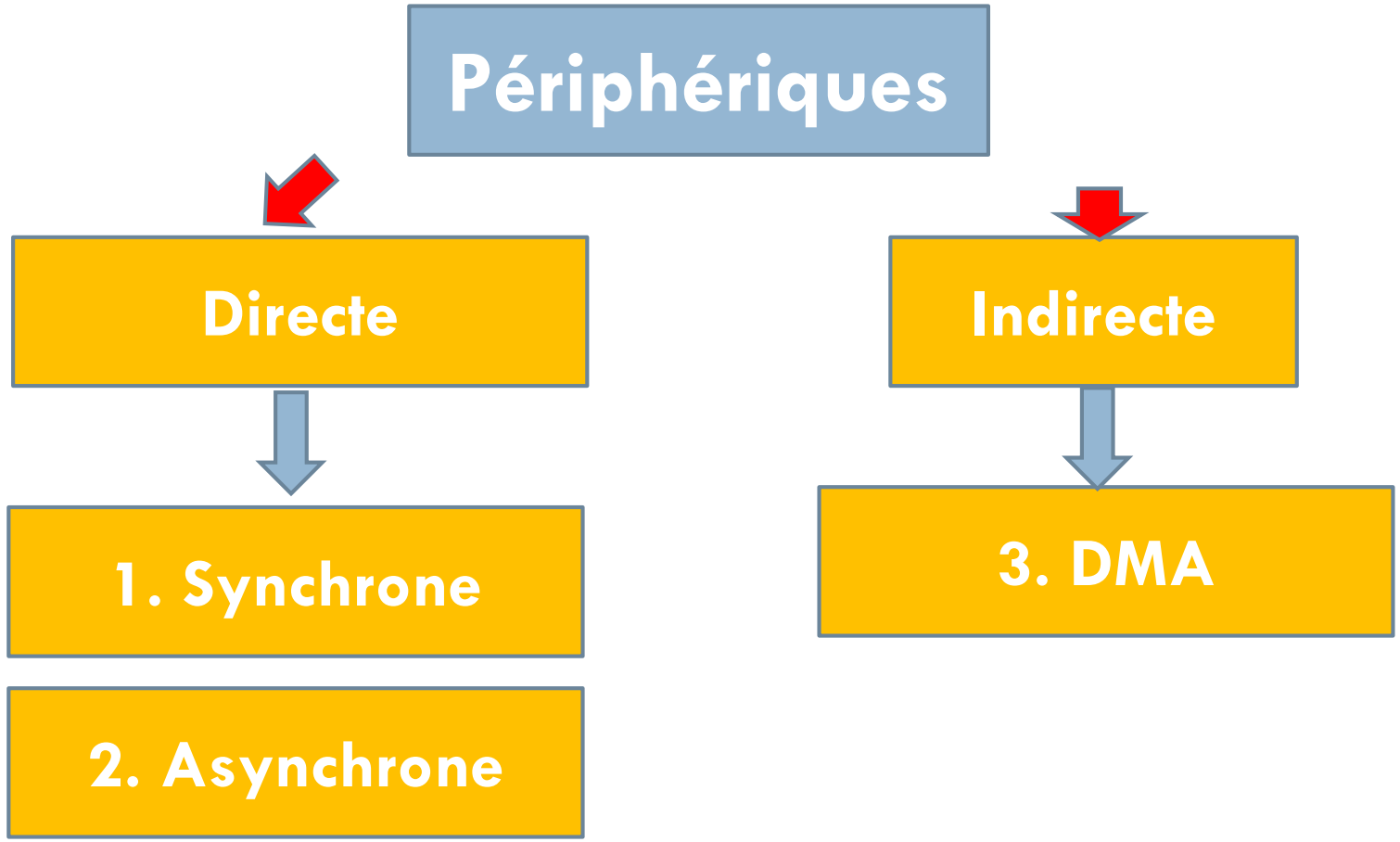
Sur les gros ordinateurs, des canaux d'E/S (**I/O Channels) allègent le travail du** processeur principal pour sa communication avec les contrôleurs (contrôle et synchronisation). Un canal d'E/S est un processeur spécialisé qui gère un ou plusieurs périphériques. (on va encore revoir ça)

Bus

- ✓ Les contrôleurs d'E/S sont connectés sur des bus, reliés à d'autres bus par des contrôleurs de bus.
- ✓ Le processeur et la mémoire sont eux-mêmes sur des bus.
- ✓ Chaque bus a ses propres caractéristiques (.Une vitesse de communication, un type de connecteur et un protocole qui décrit la façon dont sont échangées les données sur le bus)

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

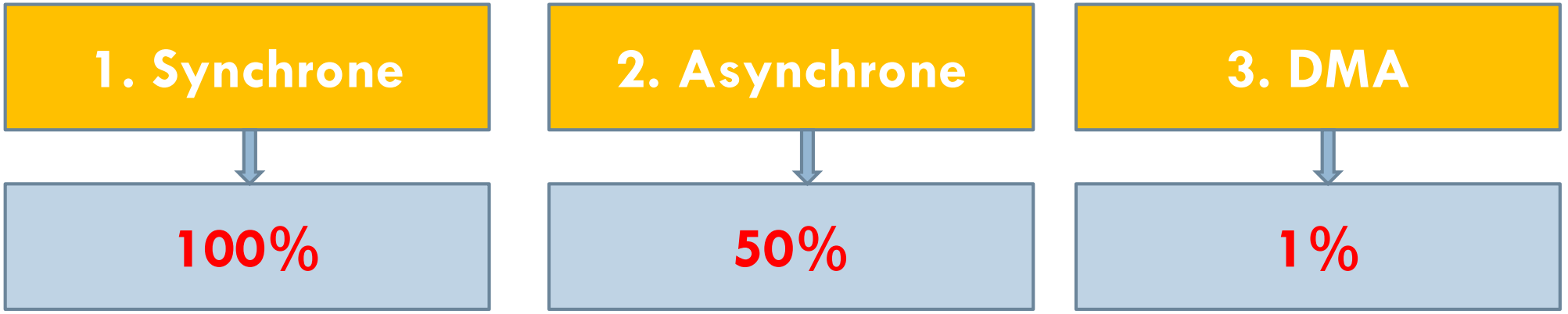


Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

Pilotage ??

La conduite du PC (pilote) envers l'opération d'E/S



Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

Explication dans l'ordre d'apparition de modes de pilotage

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

1. Mode synchrone (suivi à 100%)

Principe

Pour démarrer une opération d'E/S, le processeur (pilote) charge les informations nécessaires dans les registres appropriés du contrôleur de périphérique. Celui-ci examine à son tour le contenu de ces registres pour déterminer l'action à effectuer

(lecture, écriture, …etc.) et lance le transfert.

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

1. Mode synchrone (suivi à 100%)

Programme synchrone (pilote synchrone)

Pilote synchrone d'E/S

Début

- Initialise le transfert (sens du transfert : lecture, écriture, adresse du périphérique, …etc.).
- Vérifie la disponibilité du périphérique.
- Lance le transfert.
- Reste en attente (**active**) jusqu'à la fin du transfert.

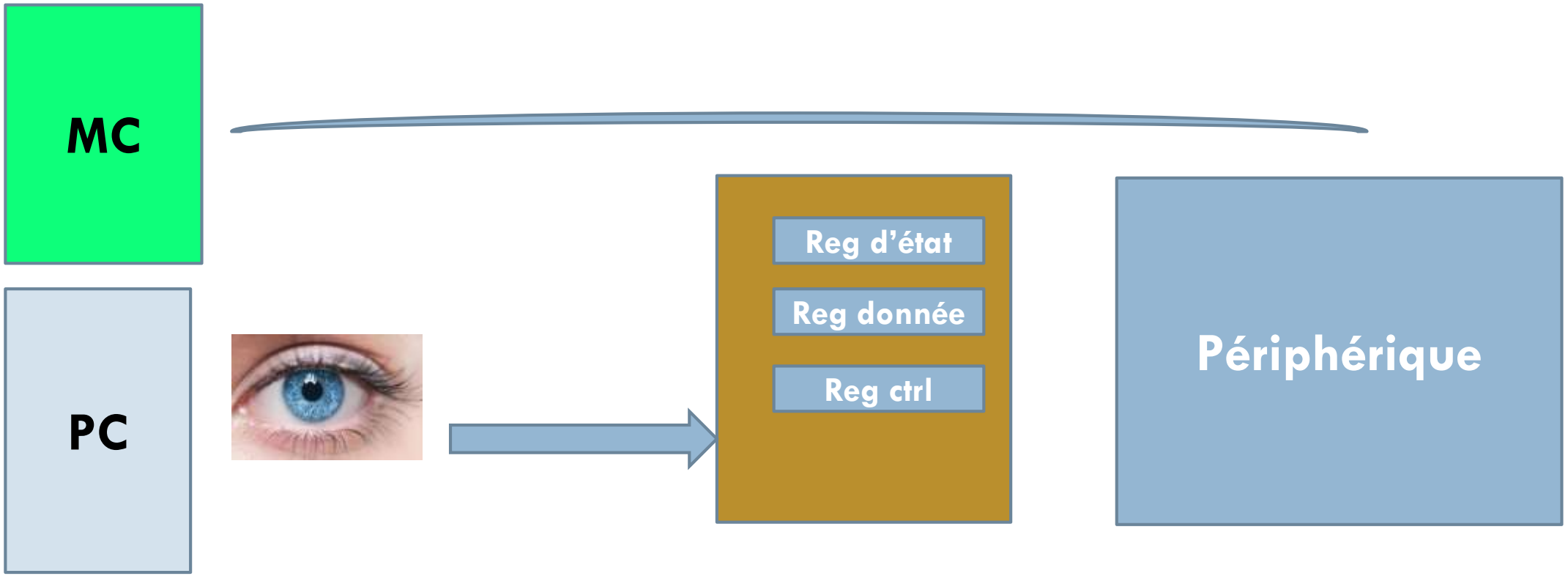
Fin.

Attente active

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

1. Mode synchrone (suivi à 100%)



Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

1. Mode synchrone (suivi à 100%)

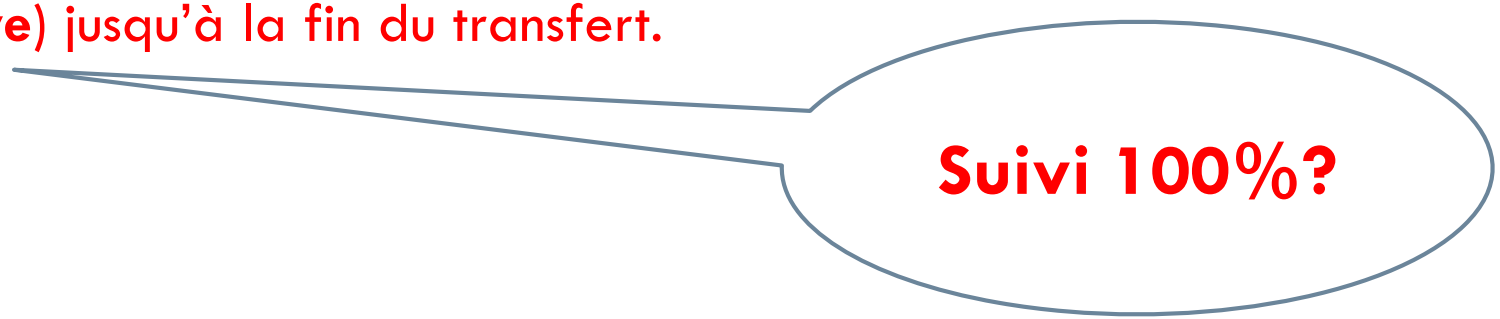
Programme synchrone (pilote synchrone)

Pilote synchrone d'E/S

Début

- Initialise le transfert (sens du transfert : lecture, écriture, adresse du périphérique, …etc.).
- Vérifie la disponibilité du périphérique.
- Lance le transfert.
- Reste en attente (**active**) jusqu'à la fin du transfert.

Fin.



Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

1. Mode synchrone (suivi à 100%)

Avantages ??

- On est arrivé à faire une E/S.
- Toutes les autres solution sont basées sur le synchrone.

Inconvénients ??

- Attente active (chômage PC)

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

2. Mode asynchrone (suivi à 50%)

Principe

Dans ce mode de pilotage, le processeur est libéré du contrôle la fin de transfert. Une interruption est générée par le coupleur du périphérique avertissant ainsi le processeur (pilote) de la fin du transfert.

Durant l'opération du transfert, le processeur peut exécuter un autre programme (scheduler).

Mode de pilotage

Dr M.CHENAIT
(mchenait@usthb.dz)

Comparaison de diagrammes

Synchrone (100%)			1er Cara		2e Cara		3e Cara	
	A(pgme user)	Pilote_syn ()	Pilote_syn () // ici la boucle active	Pilote_syn ()	Pilote_syn () // ici la boucle active	Pilote_syn	Pilote_syn () // ici la boucle active	Pilote_syn

Asynchrone (50%)			1er Cara		2e Cara		3e Cara	
	A(read(a,b,c)) pgme user)	Svc (dem E/S du premier car)	Sched()	**R**	Reprise Sched()	**R**	Reprise Sched()	**R**

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

2. Mode asynchrone (suivi à 50%)

- Programmes ??

SVC (dem E/S **du premier caractère**) + Routine de fin d'E/S du caractère

- Pourquoi « **du premier caractère** » **?**

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

2. Mode asynchrone (suivi à 50%) (le pilote asynchrone svc +R)

SVC (cause, ……)
<SC >
Switch cause of
Dem E/S (asynchrone): P-actif.etat := ″Bloqué″;
Enfiler (P-actif, FB) ;
Si périphérique est prêt **alors** lancer **le premier** caractère
Cpt=1;
sinon mettre la demande en attente // donner un tiquet
FSi
Lpsw(scheduler) ;
<RC>

Routine de Fin d'E/S d'un caractère
<SC >
Si cpt <= n **alors** lancer le caractère suivant
cpt++
Sinon // fin des caractères
Enfiler (P-actif, FA) ;
P-actif.etat := ″Prét″;
Débloquer les demandes en attente (si elles existent)
Reprendre le programme scheduler // **attention** ce n'est pas Lpsw(scheduler)
<RC>
.

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

2. Mode asynchrone (suivi à 50%)

Avantage

- Utilisation plus rationnelle du CPU. En effet, durant le transfert des caractères, le PCpeut exécuter d'autres traitements (scheduler).// élimination de l'attente active.

Inconvénient

- Perte de temps occasionnée par l'exécution des routines d'interruption, et des changements de contextes et programmes (la fréquente apparition de R).

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

3. Mode DMA (suivi à 1%)

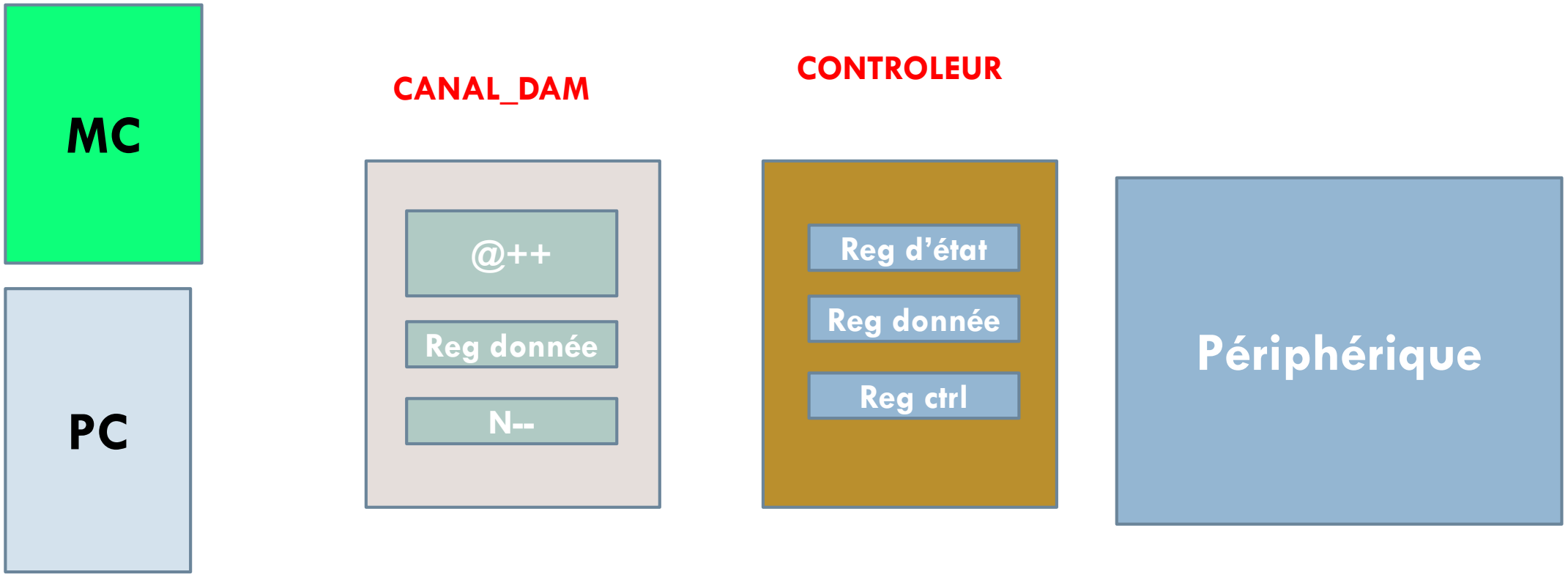
Pour éviter au PC d'intervenir à chaque transfert de caractère, on utilise un composant supplémentaire appelé**DMA (Direct Memory Access).**

Selon les architectures, le DMA est propre à un périphérique, au disque par exemple, ou partagé par plusieurs périphériques. Il peut être connecté entre un contrôleur de périphériques et le bus mémoire, permettant ainsi aux périphériques d'accéder à la mémoire sans passer par le CPU.

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

3. Mode DMA (Direct Access Memory)



Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

3. Mode DMA (suivi à 1%) // faites les rectifications nécessaires

SVC (cause, ……,n,……)
<SC >
Switch cause of
Dem E/S (asynchrone): P-actif.etat := ″Bloqué″;
Enfiler (P-actif, FB) ;
Si canal est prêt **alors**
initialiser le canal//init_canal()// remplir les registre du canal
SIO;// donner le feu vert pour commencer !
sinon mettre la demande en attente // donner un tiquet
FSi
Lpsw(scheduler) ;
<RC>

Routine de Fin d'E/S d'un caractère // Reg cpt =0

<SC >
Enfiler (P-actif, FA) ;
P-actif.etat := ″Prét″;
Débloquer les demandes en attente (si elles existent)
Reprendre le programme scheduler // **attention** ce n'est pas Lpsw(scheduler)
<RC>
.

Mode de pilotage

Dr M.CHENAIT
(mchenait@usthb.dz)

Comparaison de diagrammes

Asynchrone (50%)			**1er Cara**		**2e Cara**		**3e Cara**	
	A(read(a,b,c)) pgme user)	Svc (dem E/S du premier car)	Sched()	**R**	Reprise Sched()	**R**	Reprise Sched()	**R**

DMA (1%)			**1er Cara**	**2e Cara**	**3e Cara**			
	A(read(a,b,c)) pgme user)	Svc (dem E/S)	Sched()			**R**		

Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

3. Mode DMA (suivi à 1%)

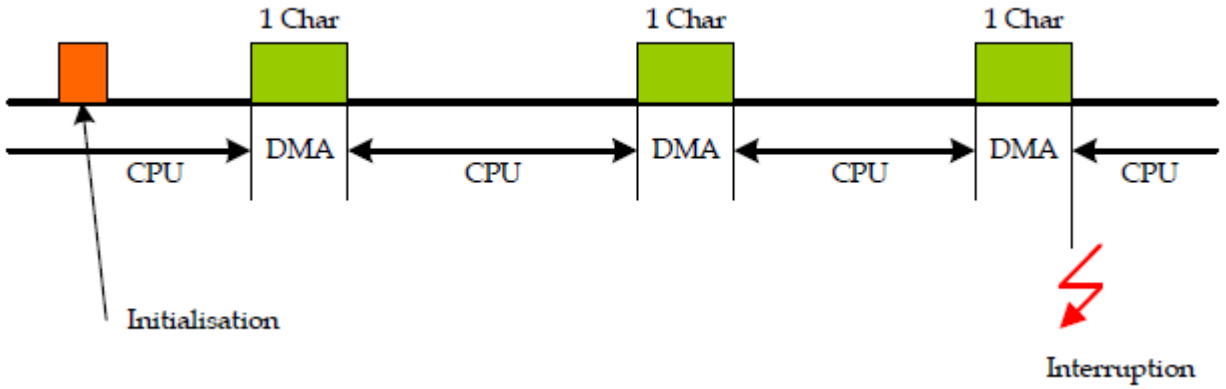
DMA et vol de cycles mémoire

Un mécanisme DMA nécessite :

La mise en place d'un chemin de passage entre le canal et la MC, matérialisé par un bus.

Un gestionnaire des **conflits d'accès à la MC entre le canal et le CPU.**

Pour cela, une technique de **vols de cycles (Cycle Stealing) est utilisée**



Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

3. Mode DMA (suivi à 1%)

Autres notions….

1. Traitement d'E/S simultanées

File bloquée propre à chaque périphérique.

2. E/S Tamponnées

Exemples

- Dans un échange avec un disque magnétique, l'essentiel du temps pris par l'échange vient du positionnement du bras et du délai rotationnel. Il est alors classique de ranger dans un tampon le contenu de toute une piste.
- La lecture de caractères au clavier peut se faire par anticipation ; c'est à dire que l'utilisateur peut frapper des caractères avant que le programme ait envoyé un ordre d'entrée. Les caractères frappés sont stockés dans le tampon et seront plus tard transférés dans une zone utilisateur.

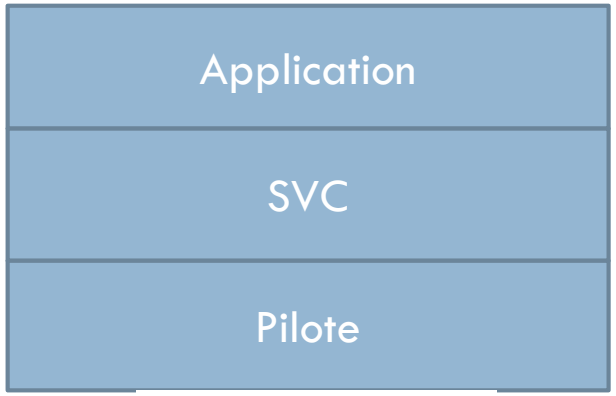
Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

3. Mode DMA (suivi à 1%)

Autres notions….

3. Les Couches logicielles
   3.1 Couche logicielles ??
   3.2 Intérêt de cette schématisation



Dr M.CHENAIT
(mchenait@usthb.dz)

Mode de pilotage

3. Mode DMA (suivi à 1%)

Autres notions….

3. Les Couches logicielles

- Voir le détail à la page 37 du cours + faire exo 3 de la serie 3