

Les Entrées/Sorties (I/Os)



GIF-1001 Ordinateurs: Structure et Applications
Jean-François Lalonde

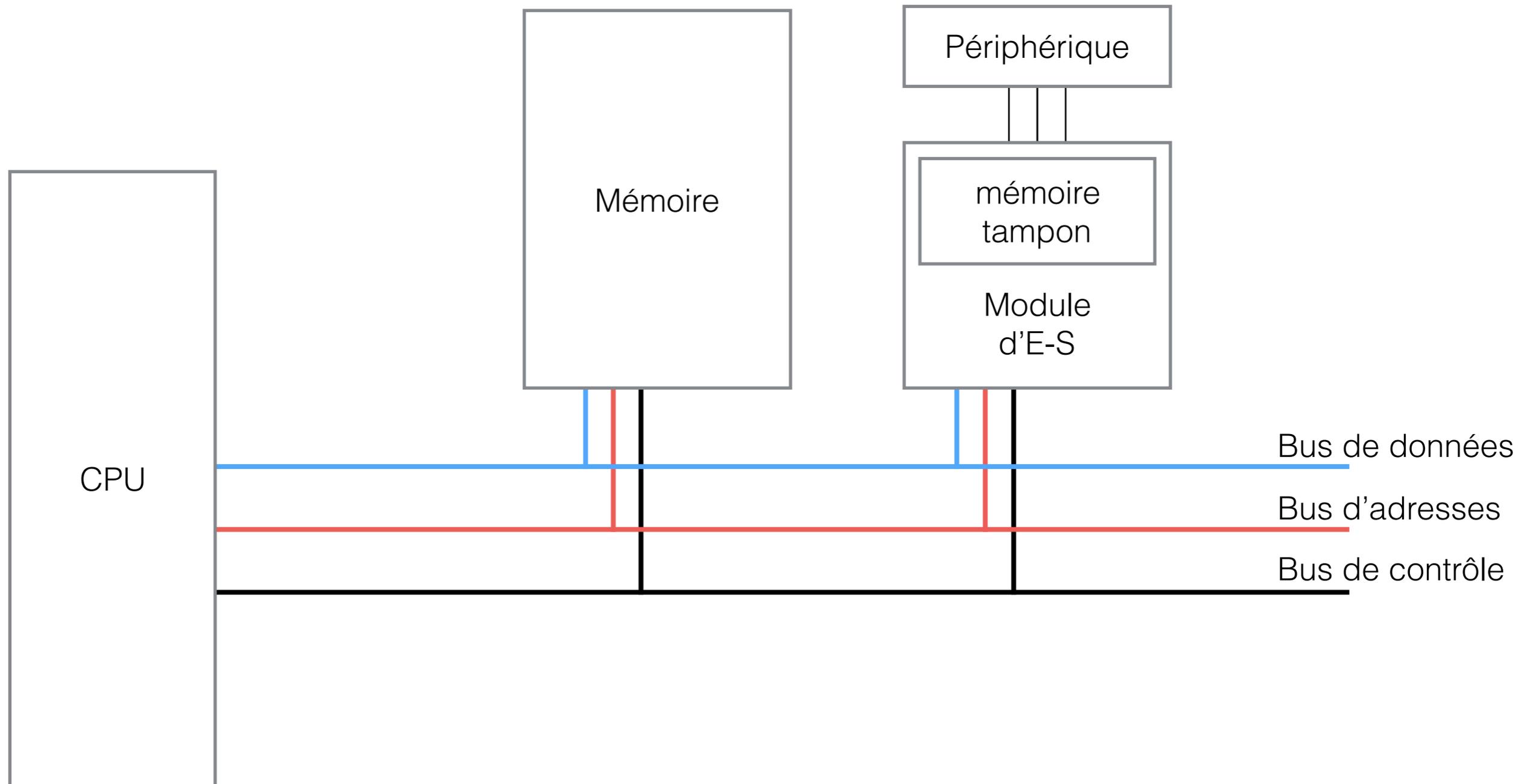
Entrées-sorties

- Prenons un exemple très courant, soit la copie d'un bloc de données provenant d'un périphérique (ex: le disque dur) vers la RAM.
- Le transfert est divisé en deux étapes principales:
 1. La **requête**: le micro-processeur informe le contrôleur de périphérique qu'il aura besoin d'un bloc de données.
 - Les données doivent être transférées du périphérique vers la mémoire tampon de son contrôleur. Ce transfert est géré par le contrôleur.
 - Cela peut prendre du temps!
 2. Le **transfert**: lorsque le périphérique est prêt, les données sont ensuite copiées du contrôleur de périphérique (mémoire tampon) vers la RAM.

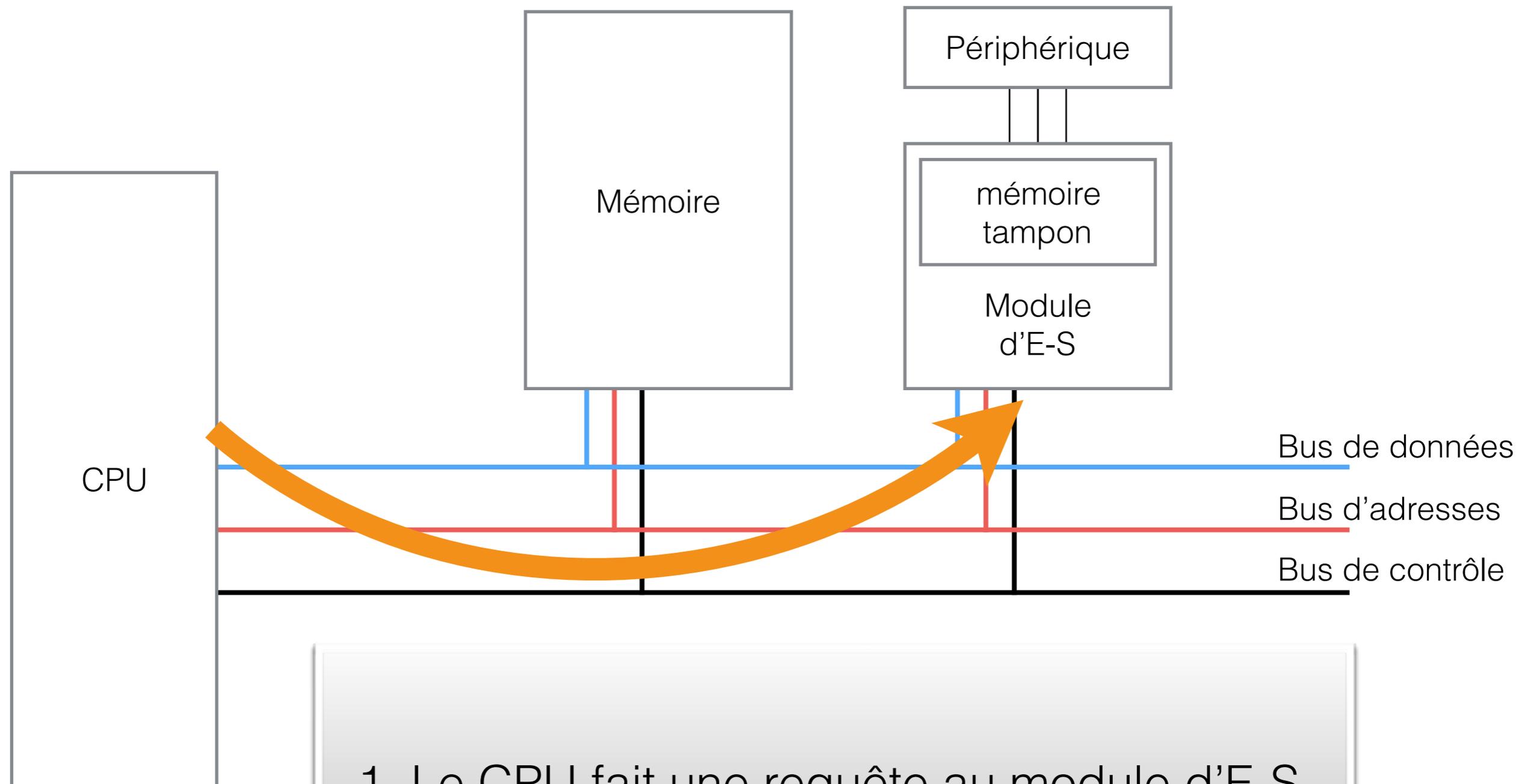
Entrées-sorties

- Il existe trois techniques principales pour effectuer un tel transfert:
 - E/S programmées;
 - E/S avec interruptions;
 - le *Direct Memory Access (DMA)*.

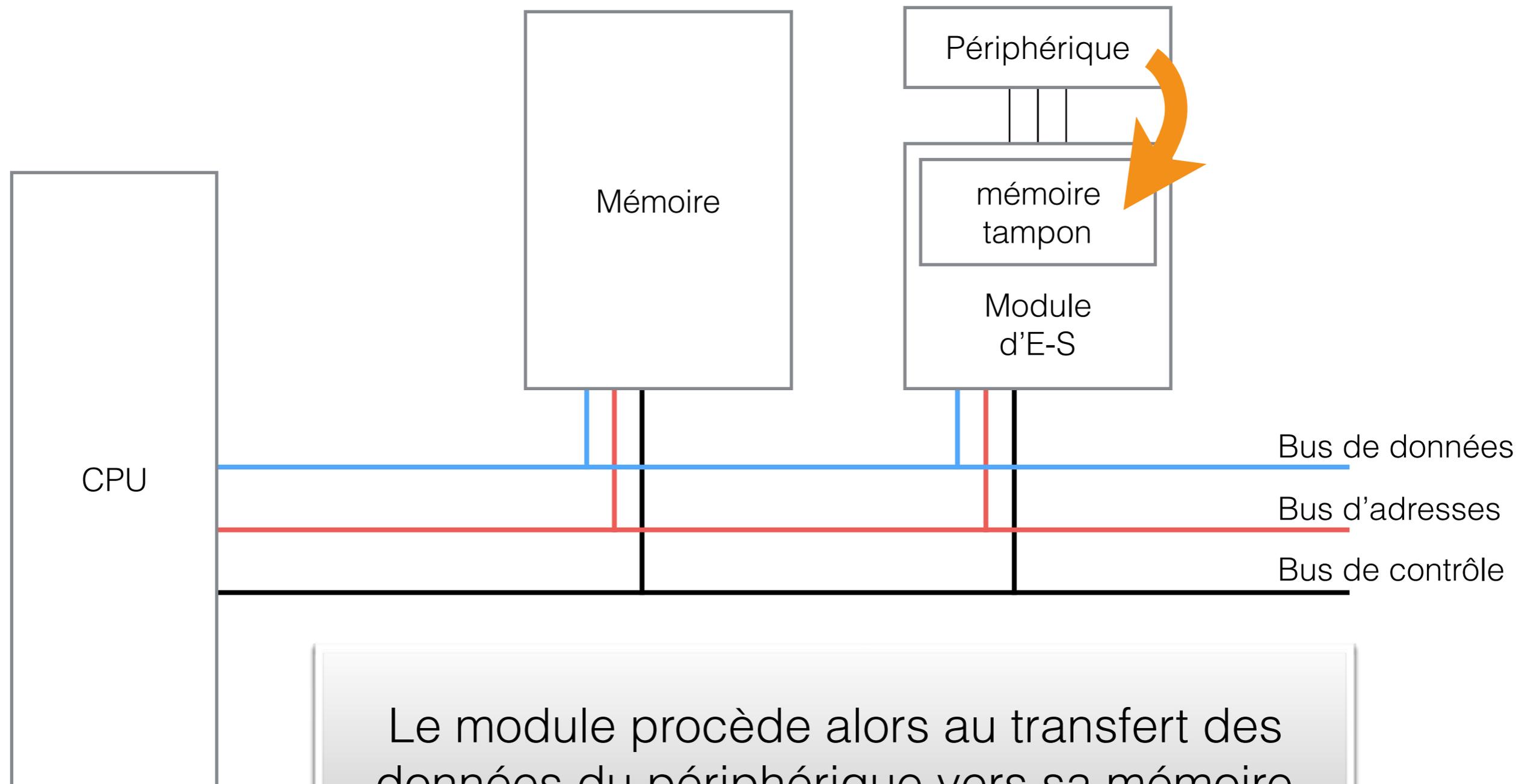
Entrées-sorties programmées



Entrées-sorties programmées

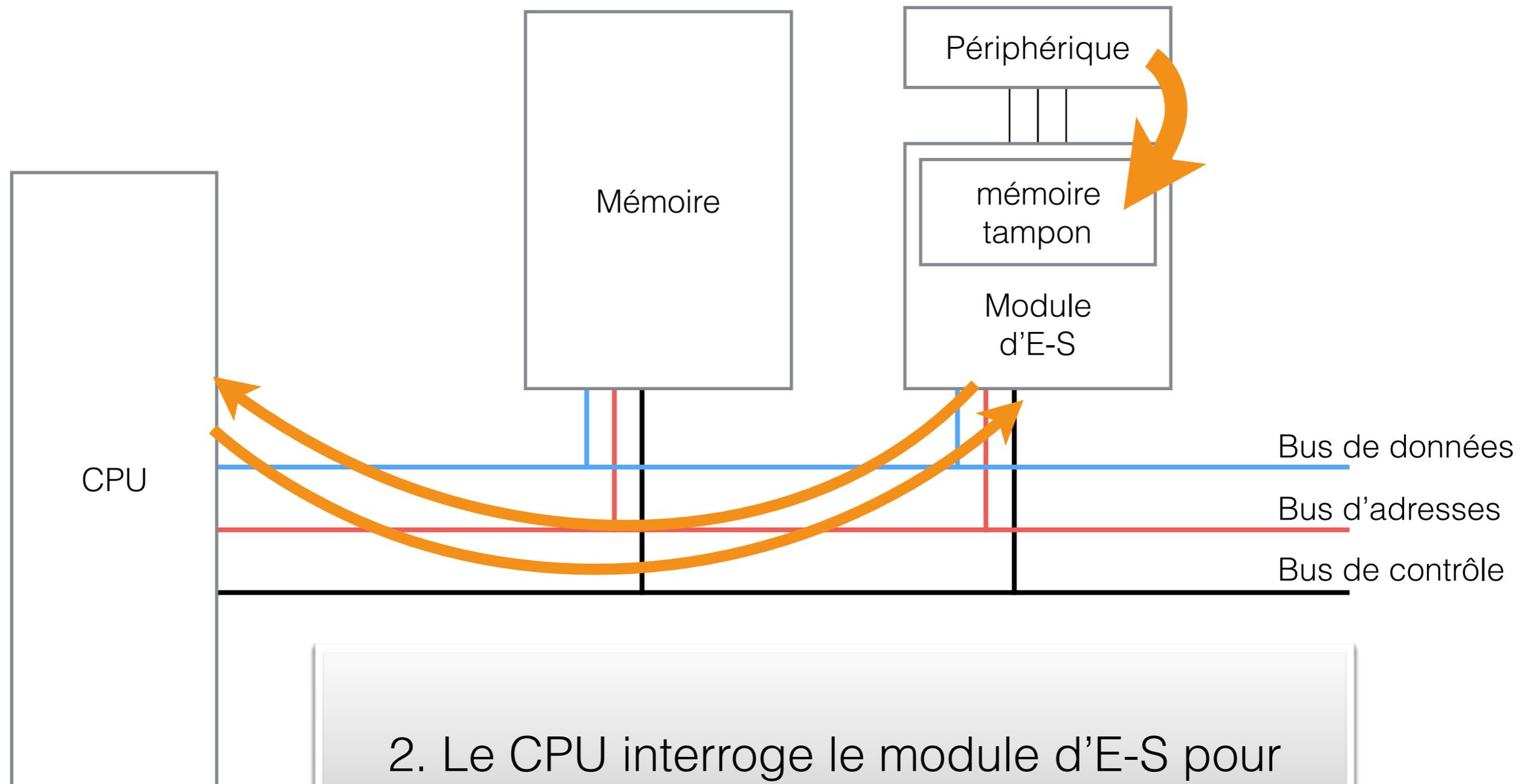


Entrées-sorties programmées

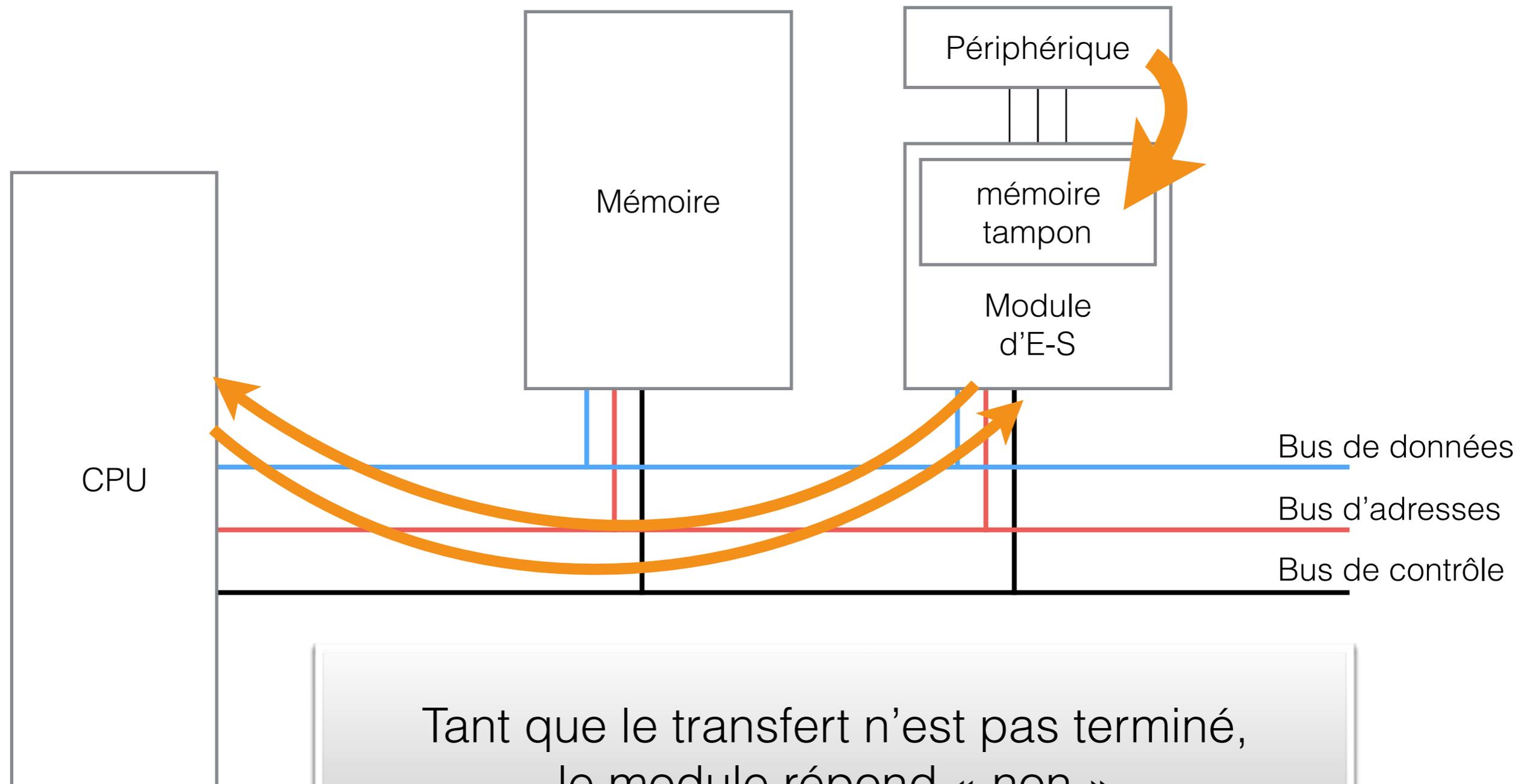


Le module procède alors au transfert des données du périphérique vers sa mémoire tampon.

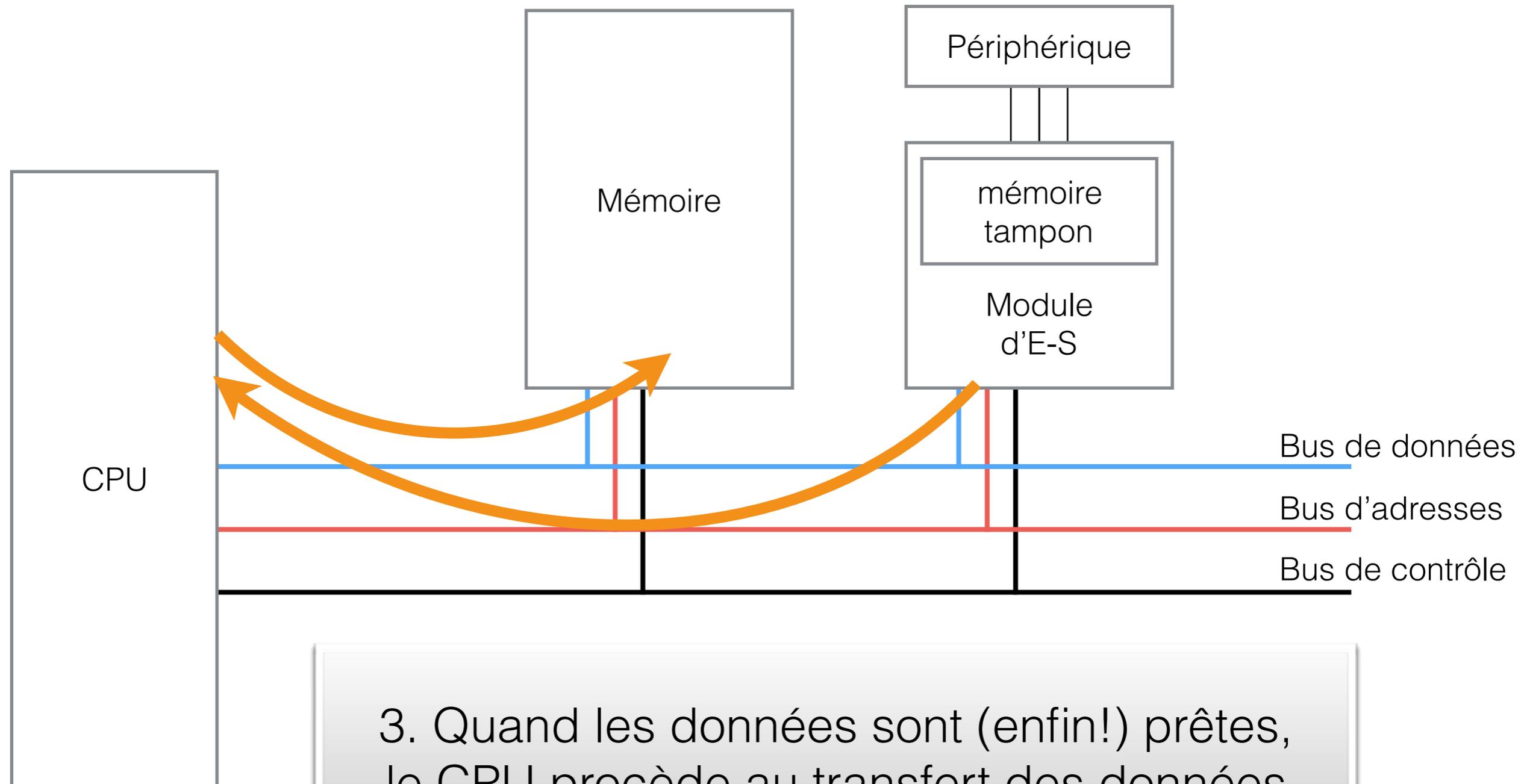
Entrées-sorties programmées



Entrées-sorties programmées



Entrées-sorties programmées



3. Quand les données sont (enfin!) prêtes, le CPU procède au transfert des données vers la RAM avec des LDR/STR.

Entrées-sorties programmées

- Pour accéder à un périphérique, on exécute les étapes suivantes:
 1. Envoyer une requête au périphérique
 1. Le périphérique démarre la requête—ce qui peut être long à compléter!
 2. Attendre que la requête soit finie en interrogeant les registres de statut du périphérique constamment (*polling*)
 3. Lorsque la requête est complétée, effectuer la copie mémoire
 1. La copie est effectuée entre la mémoire tampon du périphérique vers la RAM

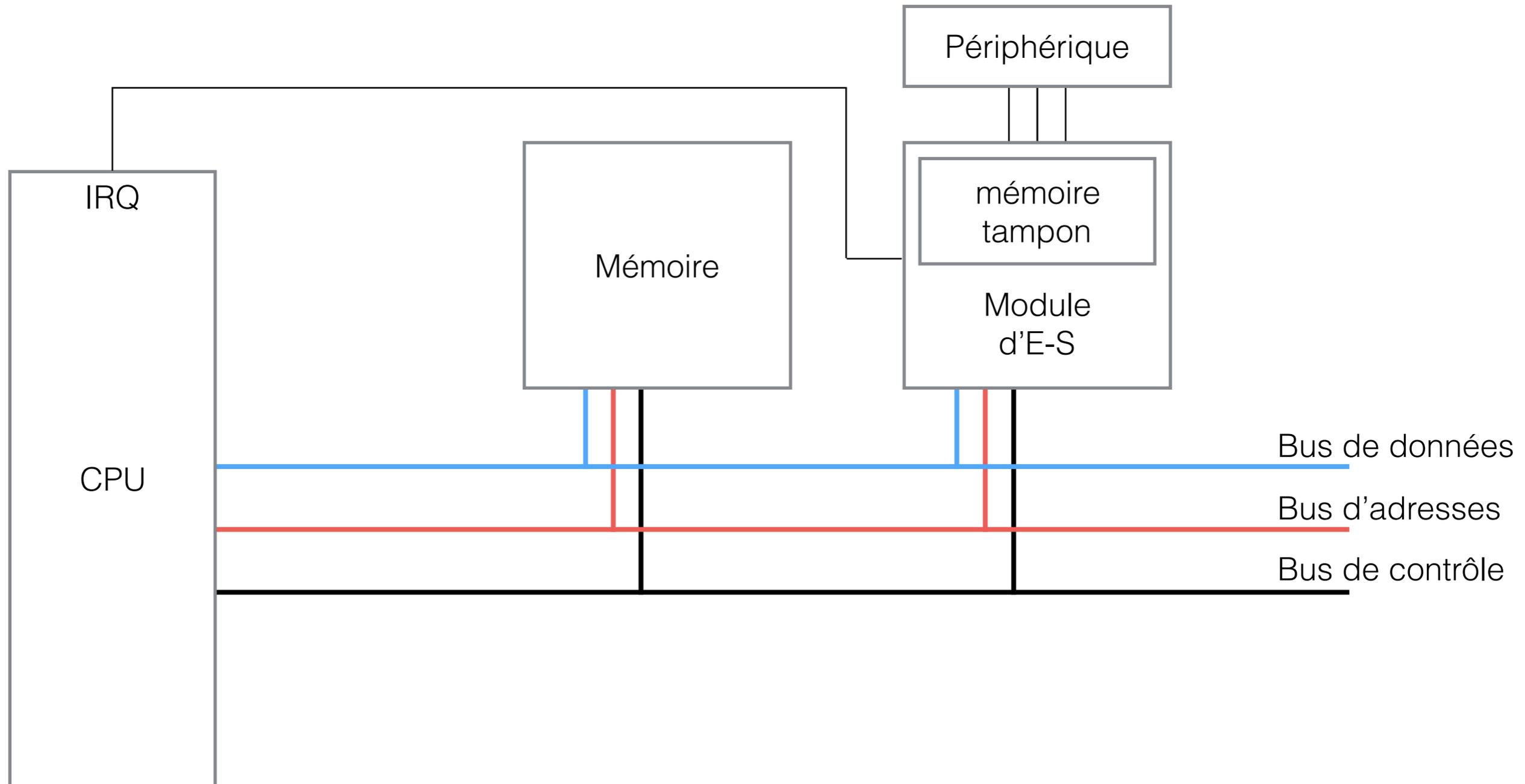
Entrées-sorties programmées

- Avantage?
 - C'est **simple**. Pas besoin de mécanisme supplémentaire autre que les bus que nous avons déjà.
- Désavantage?
 - C'est **inefficace**.
 - Le micro-processeur est utilisé pour interroger le périphérique alors qu'il pourrait faire autre chose pendant ce temps.
 - Le microprocesseur doit gérer le transfert de données lui-même: il ne peut faire autre chose durant ce temps.

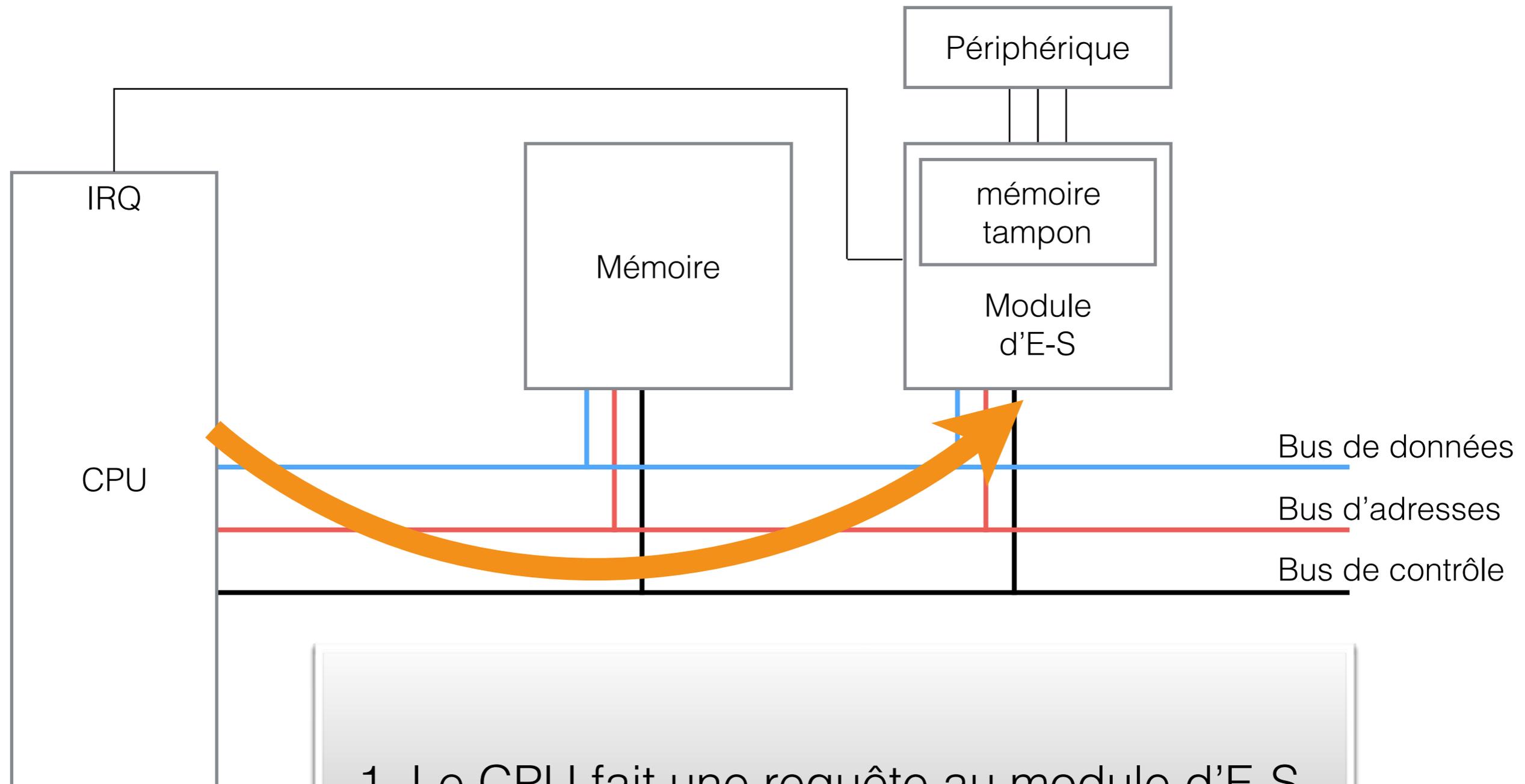
Entrées-Sorties par interruptions

- Plutôt que d'interroger le module d'E-S constamment, il émet une interruption lorsqu'il est prêt à effectuer le transfert en RAM.
- Pendant ce temps, le microprocesseur peut faire autre chose plutôt que de l'interroger constamment!

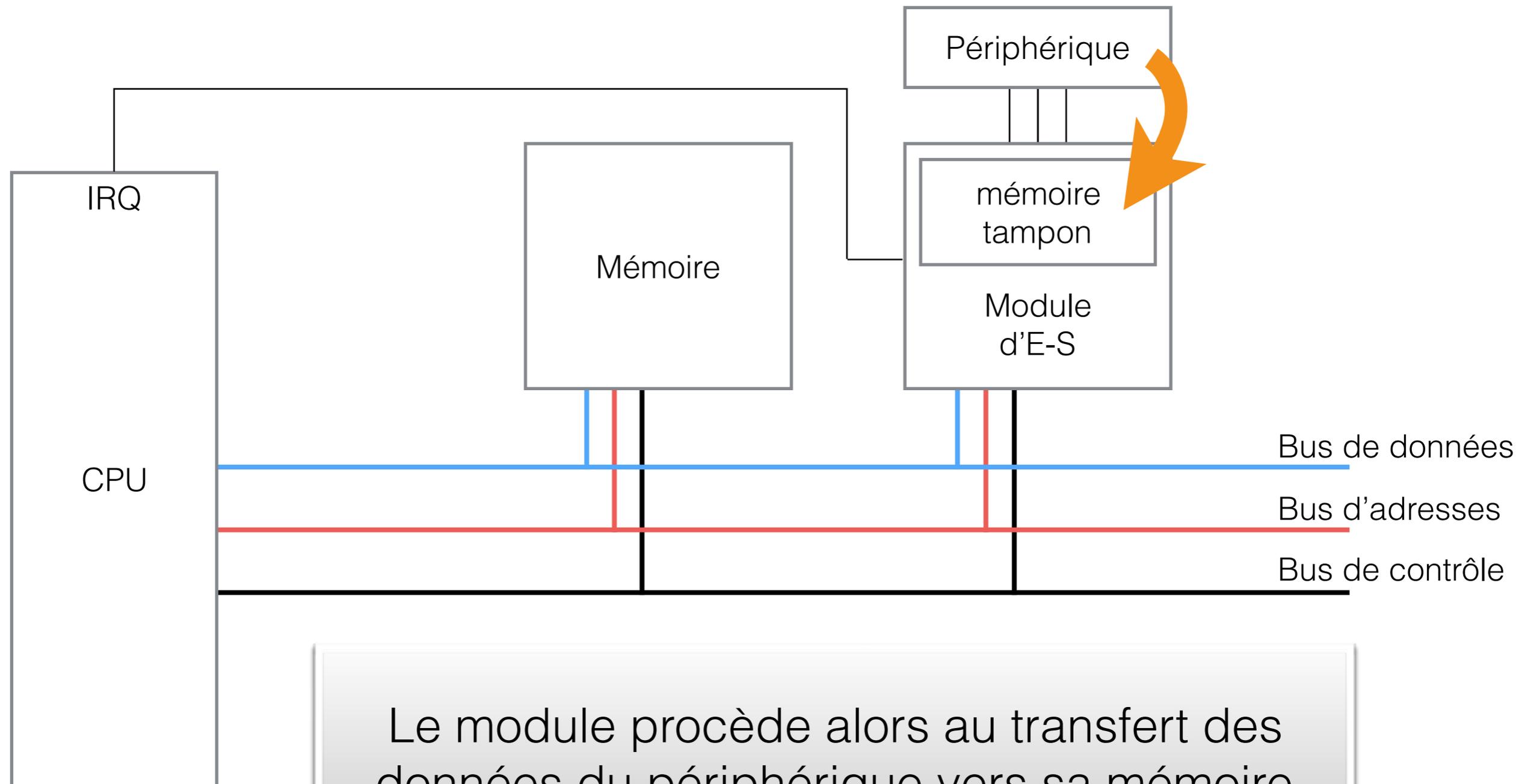
Entrées-sorties par interruptions



Entrées-sorties par interruptions

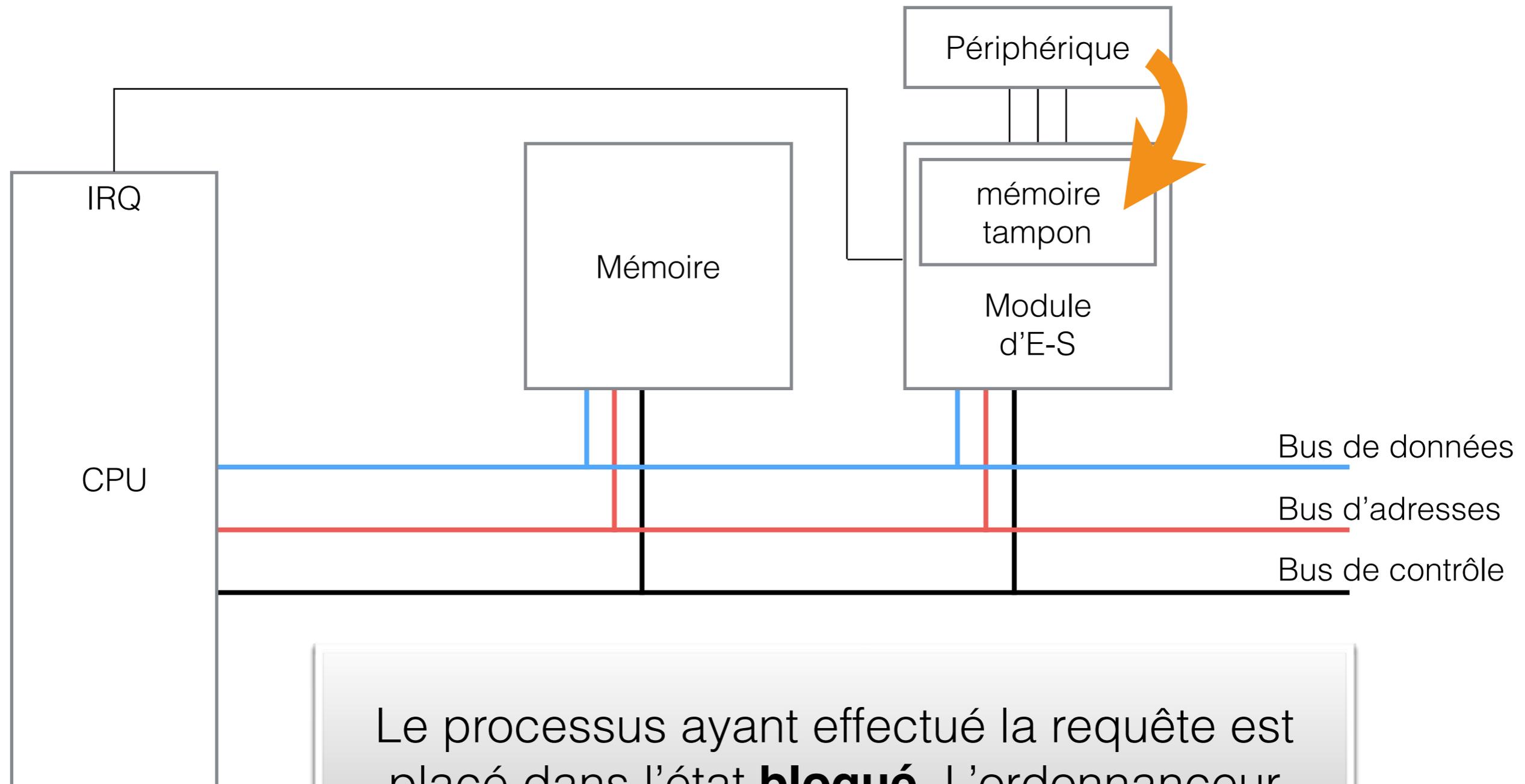


Entrées-sorties par interruptions



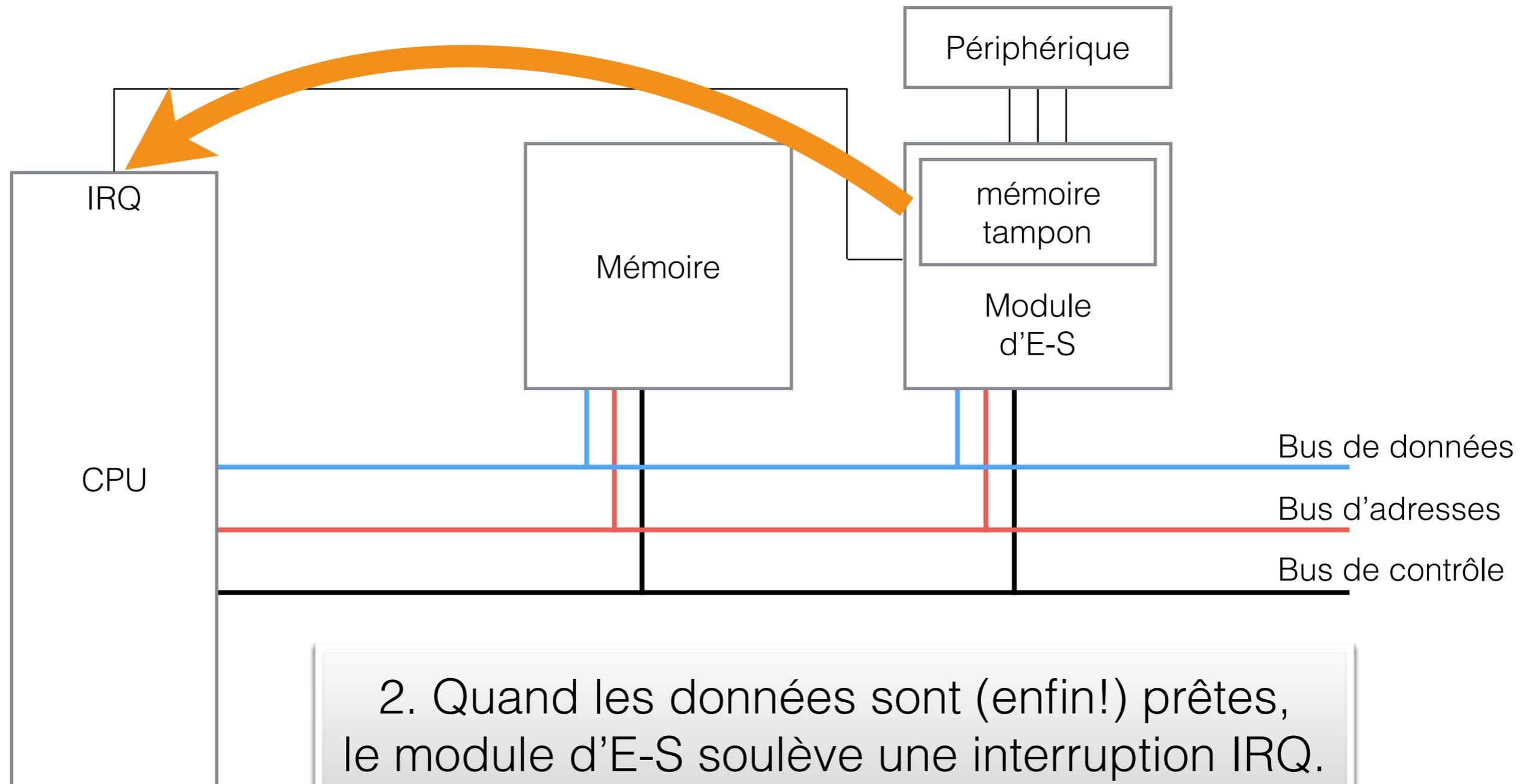
Le module procède alors au transfert des données du périphérique vers sa mémoire tampon.

Entrées-sorties par interruptions



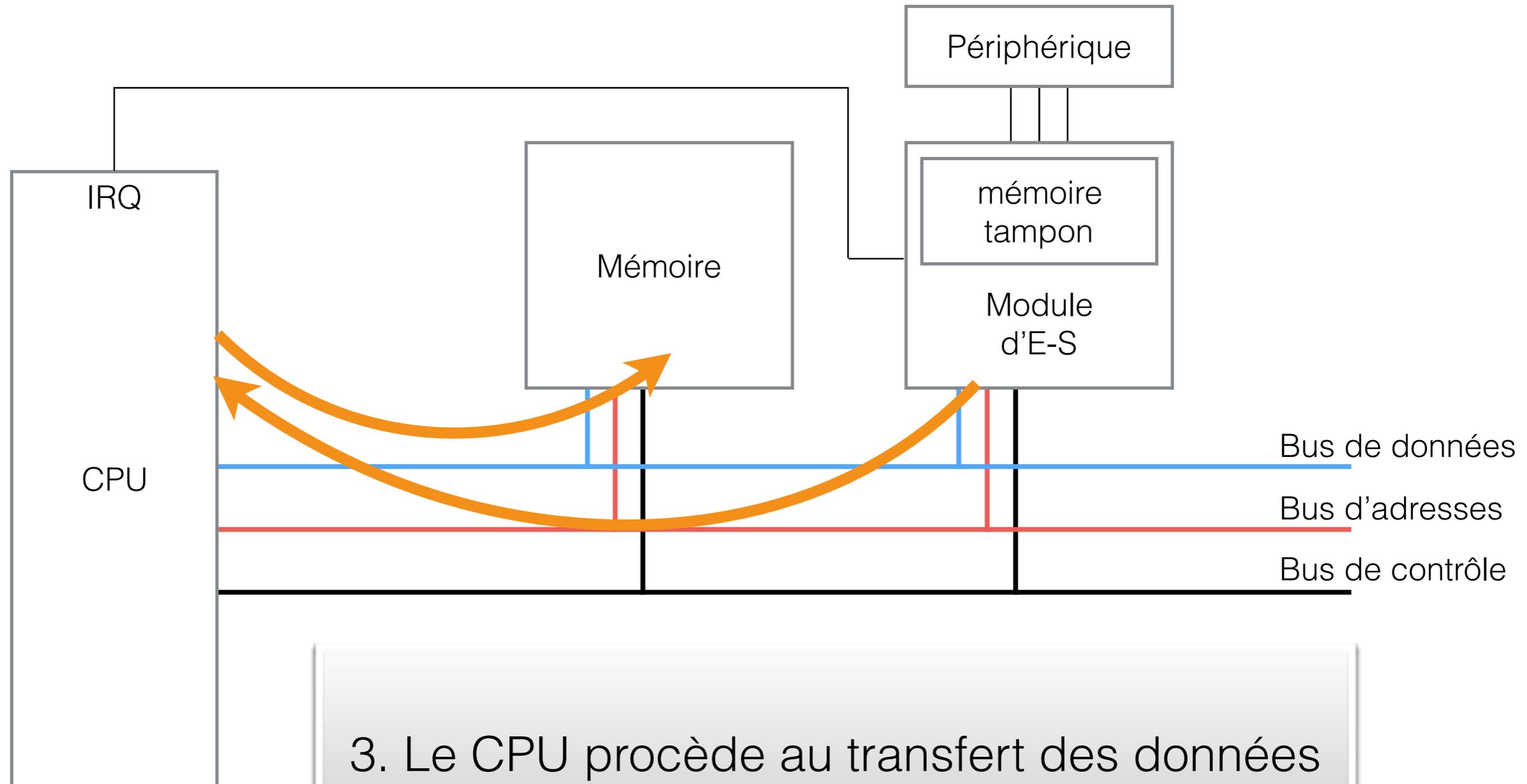
Le processus ayant effectué la requête est placé dans l'état **bloqué**. L'ordonnanceur sélectionne un autre processus à exécuter.

Entrées-sorties par interruptions



2. Quand les données sont (enfin!) prêtes, le module d'E-S soulève une interruption IRQ. Le système d'exploitation place le processus ayant effectué la requête dans l'état **prêt**.

Entrées-sorties par interruptions



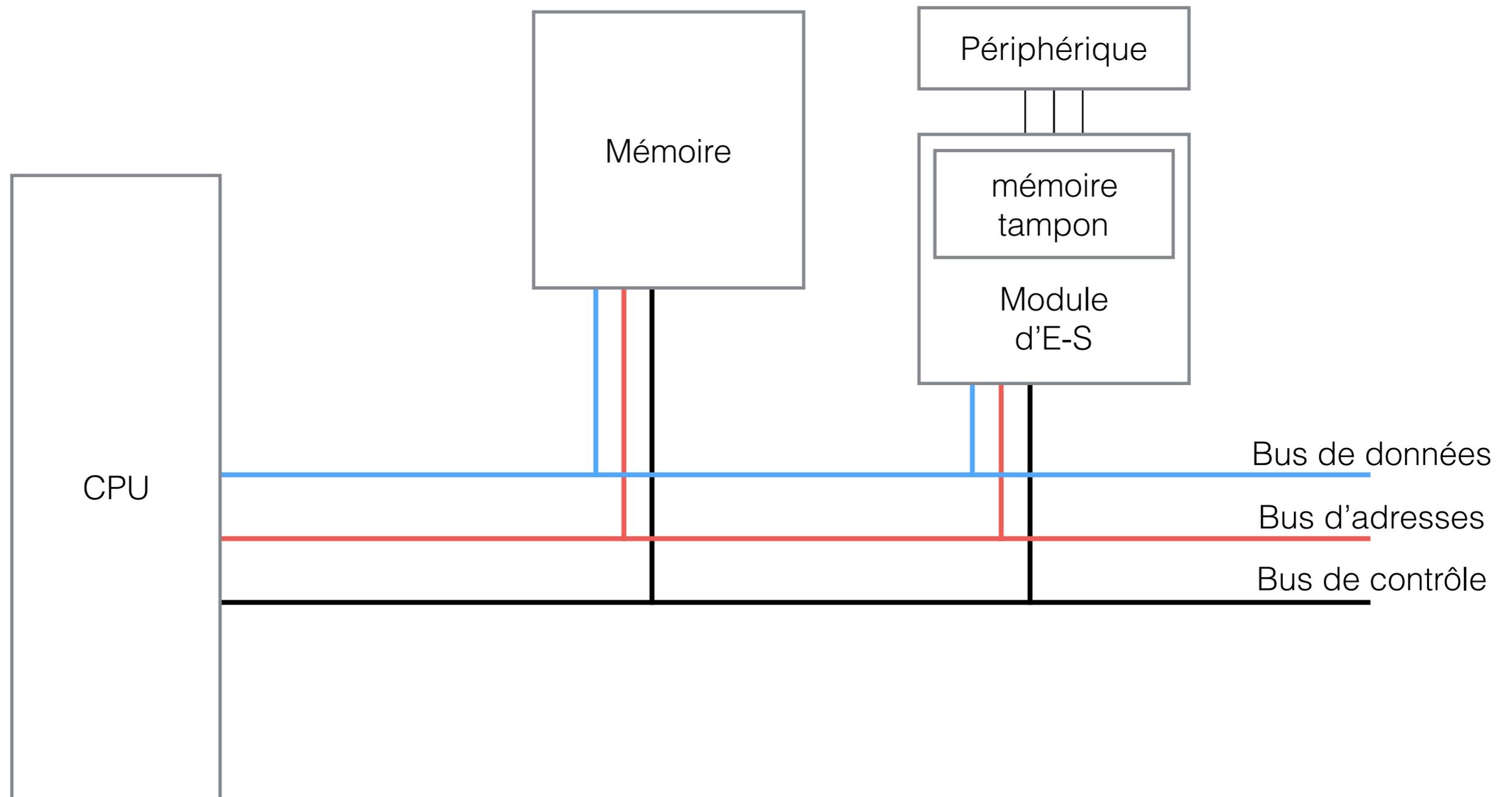
Entrées-sorties par interruptions

- Pour accéder à un périphérique, on exécute les étapes suivantes:
 1. Envoyer une requête au périphérique
 1. Le périphérique démarre la requête—ce qui peut être long à compléter!
 2. Pendant ce temps, le microprocesseur peut faire autre chose.
 2. Lorsque la requête est terminée, le périphérique informe le microprocesseur en soulevant une interruption.
 3. Lorsque la requête est complétée, effectuer la copie mémoire
 1. La copie est effectuée entre la mémoire tampon du périphérique vers la RAM

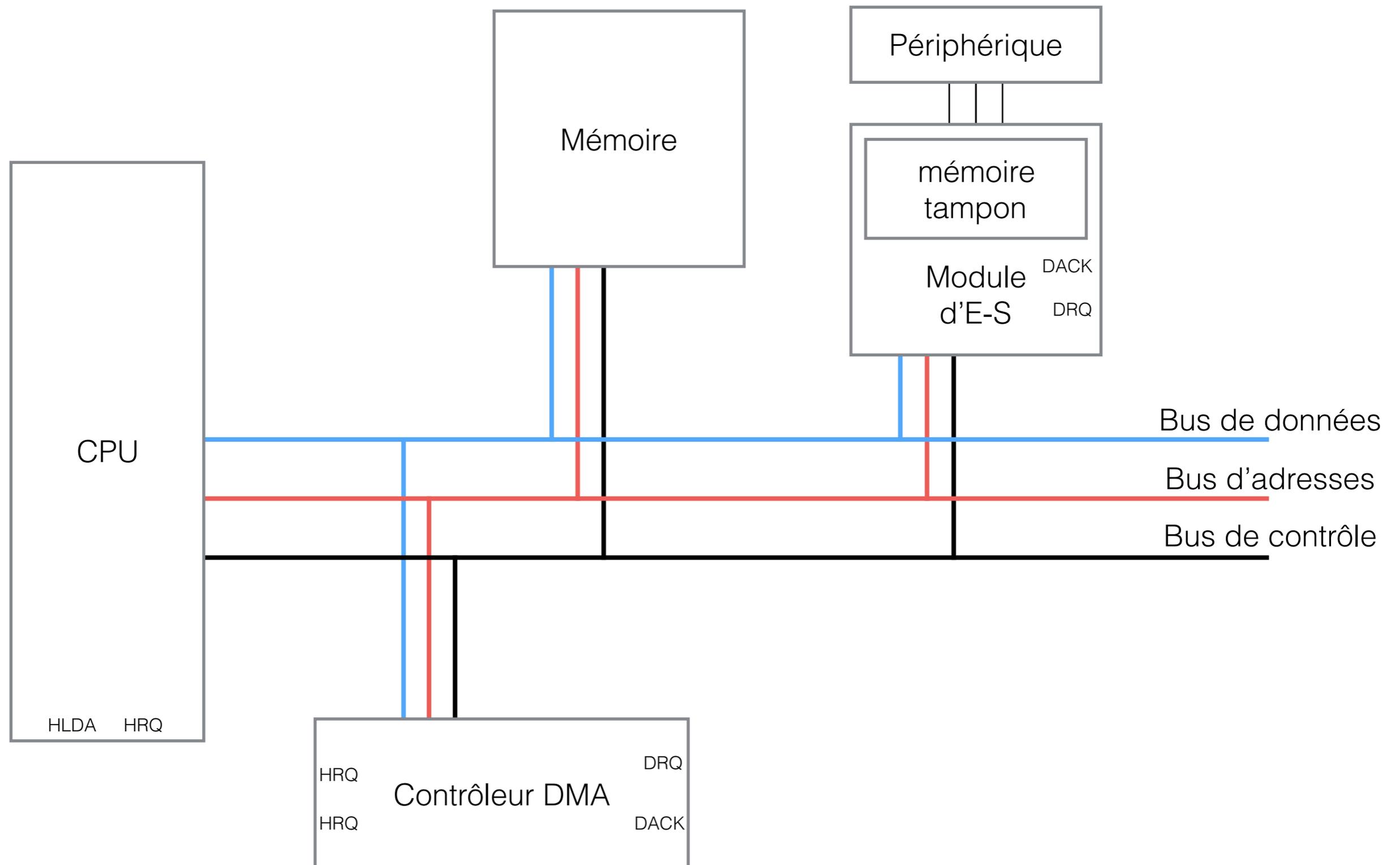
Entrées-sorties par interruptions

- Avantages
 - Plus **efficace**. Plus besoin d'attendre après le périphérique, c'est lui qui dit au microprocesseur quand il est prêt
- Désavantages?
 - Plus **complexe**. Nécessite des circuits et mécanismes supplémentaires pour gérer les interruptions.
 - Transfert **inefficace**. Le microprocesseur doit gérer le transfert de données lui-même: il ne peut faire autre chose durant ce temps.

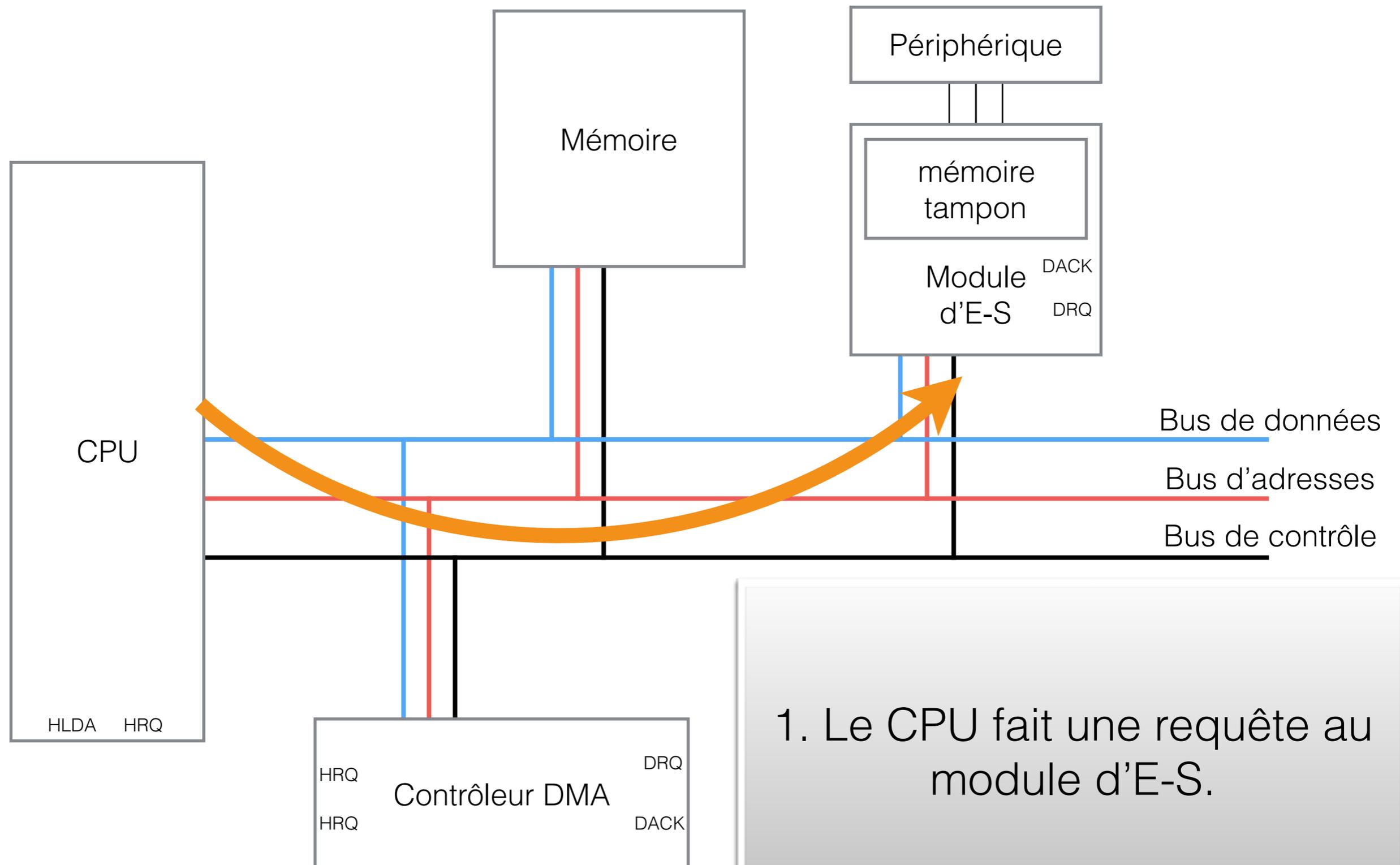
Direct Memory Access (DMA)



Direct Memory Access (DMA)

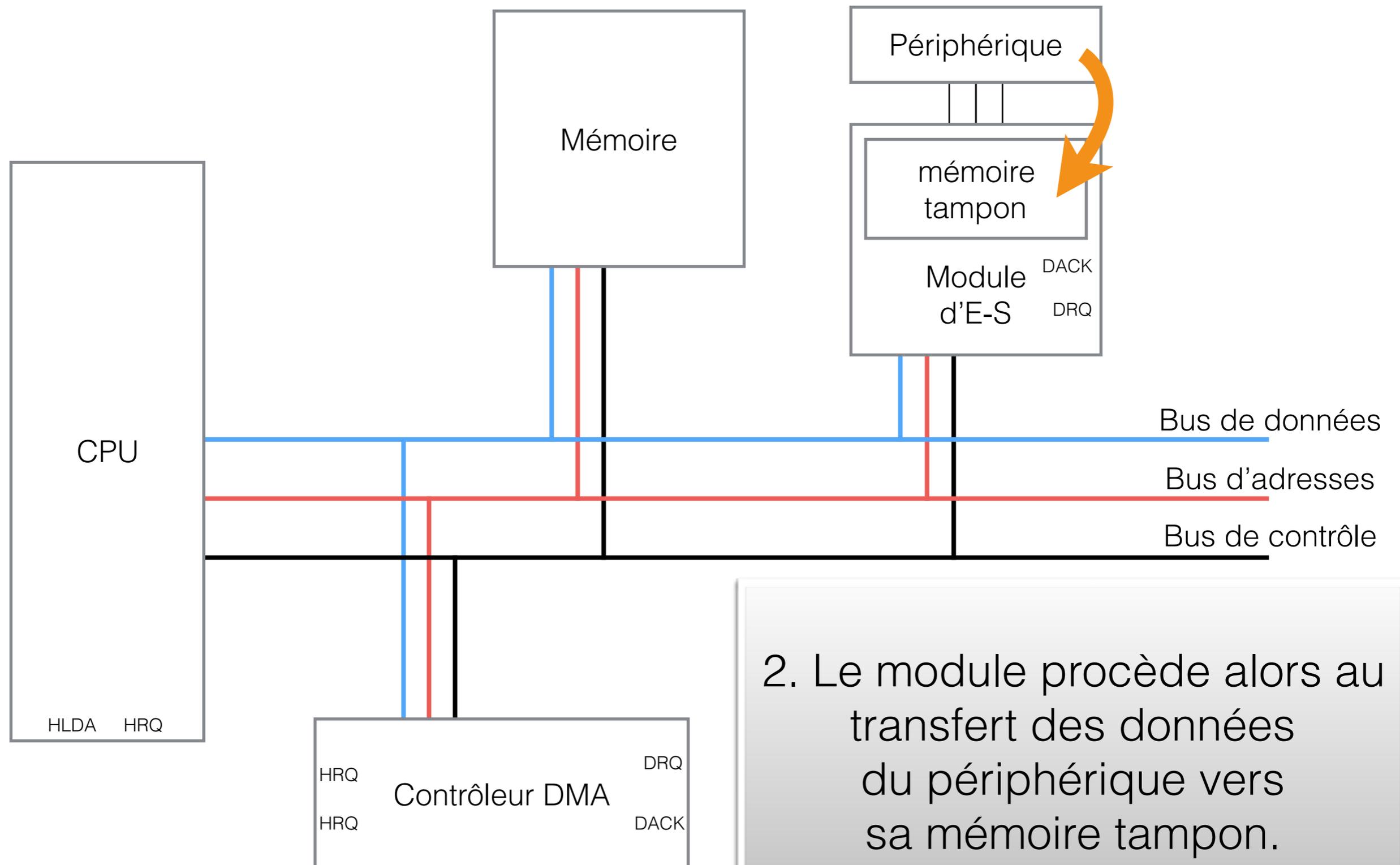


Direct Memory Access (DMA)

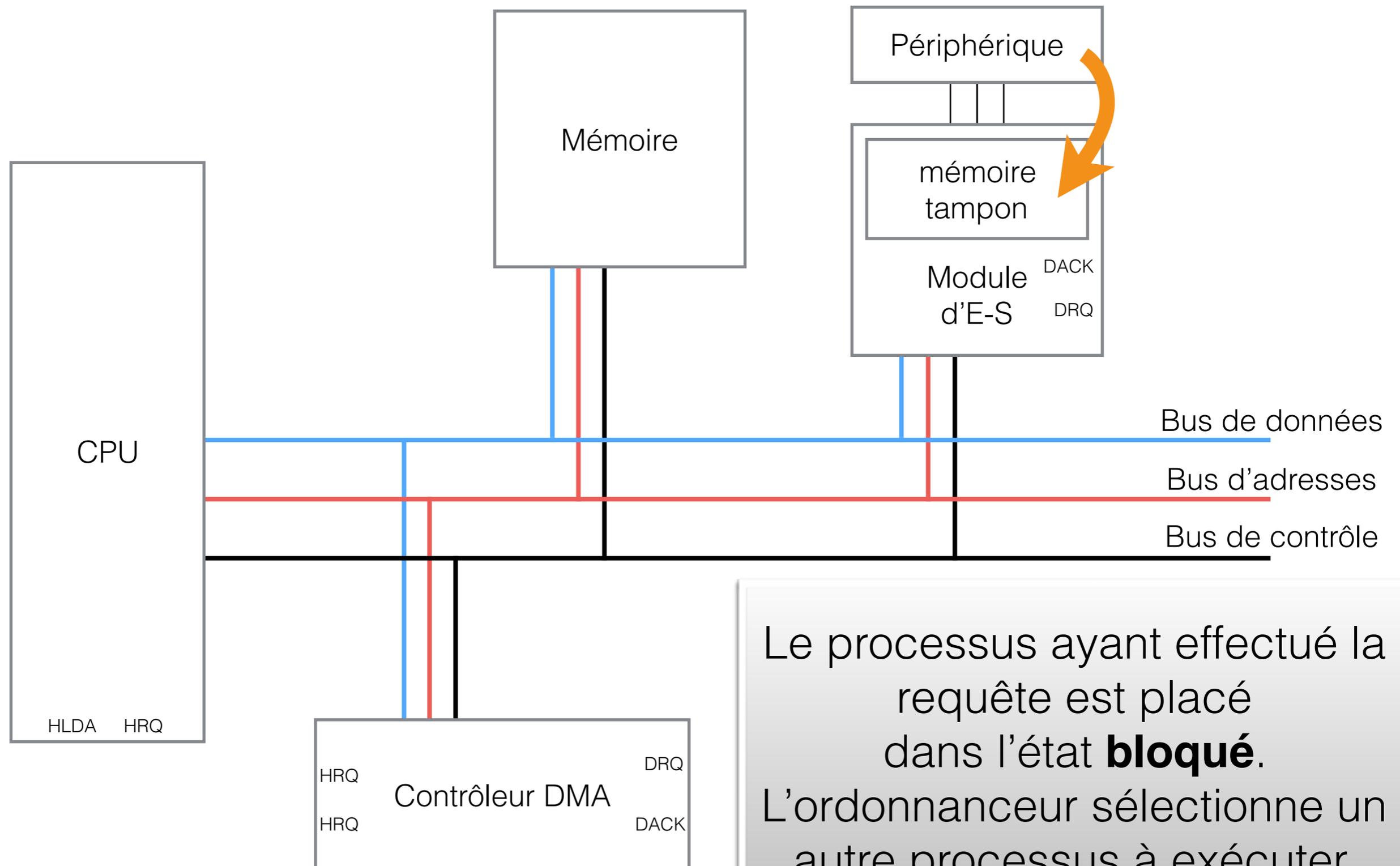


1. Le CPU fait une requête au module d'E-S.

Direct Memory Access (DMA)

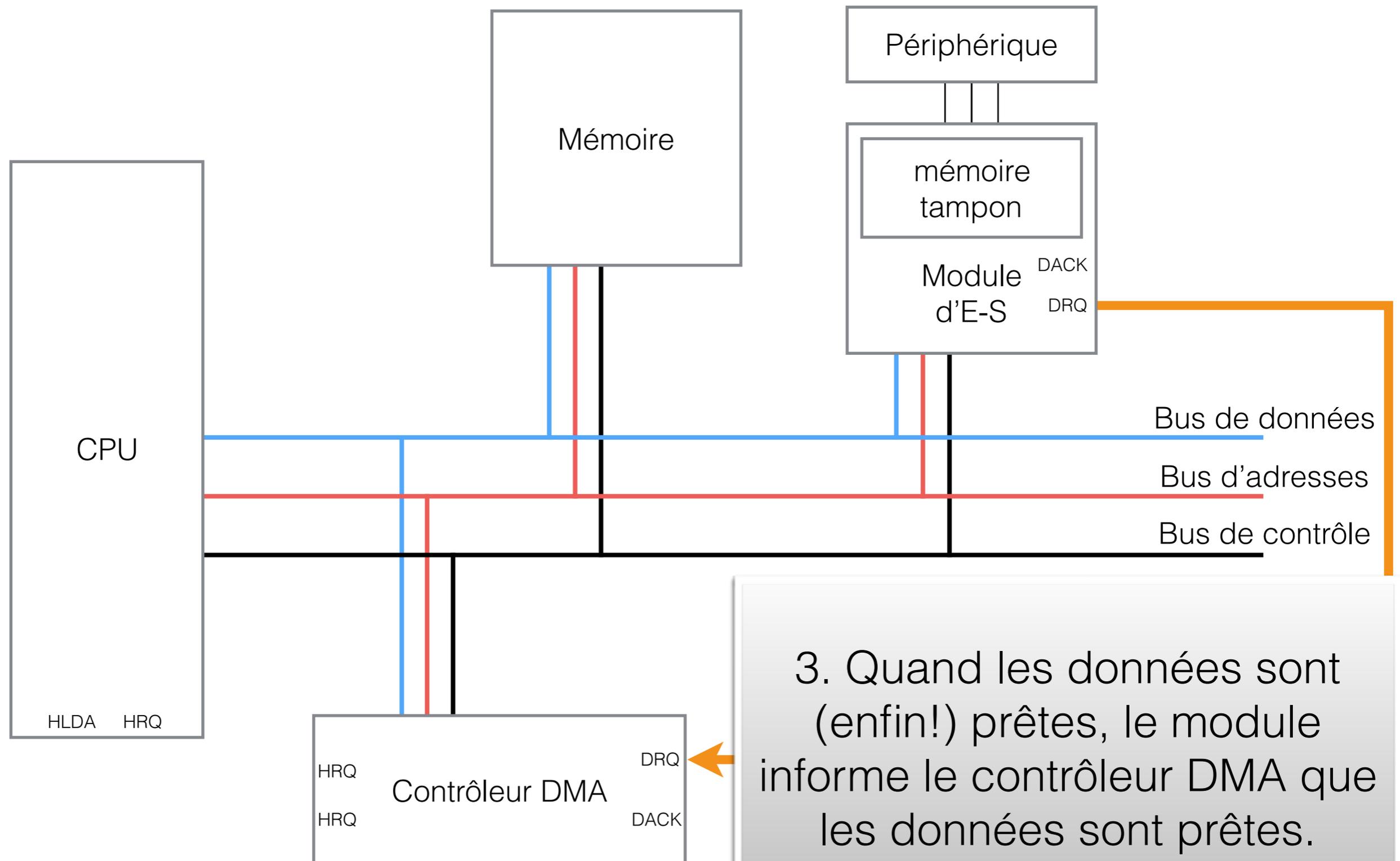


Direct Memory Access (DMA)

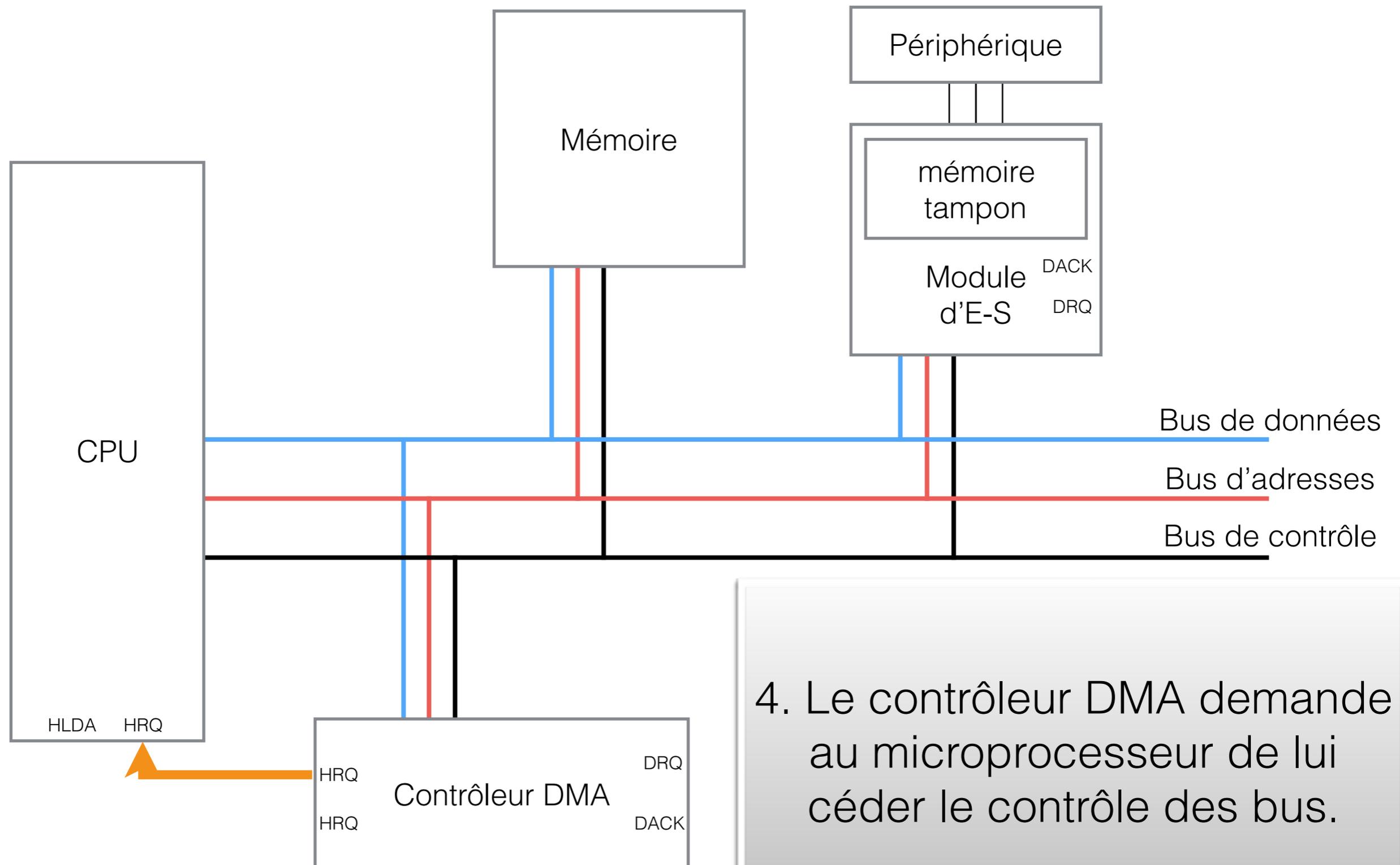


Le processus ayant effectué la requête est placé dans l'état **bloqué**.
L'ordonnanceur sélectionne un autre processus à exécuter.

Direct Memory Access (DMA)

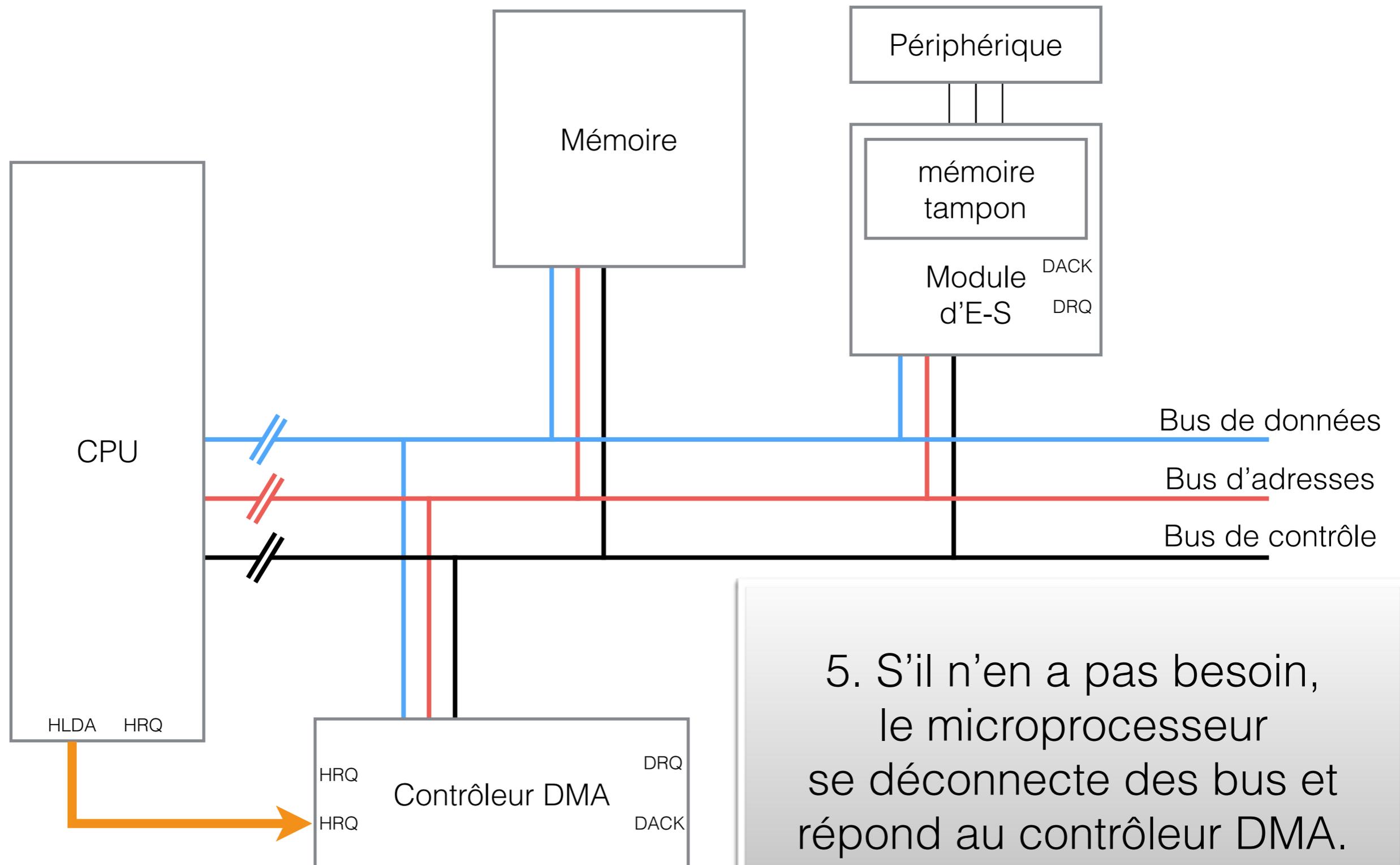


Direct Memory Access (DMA)

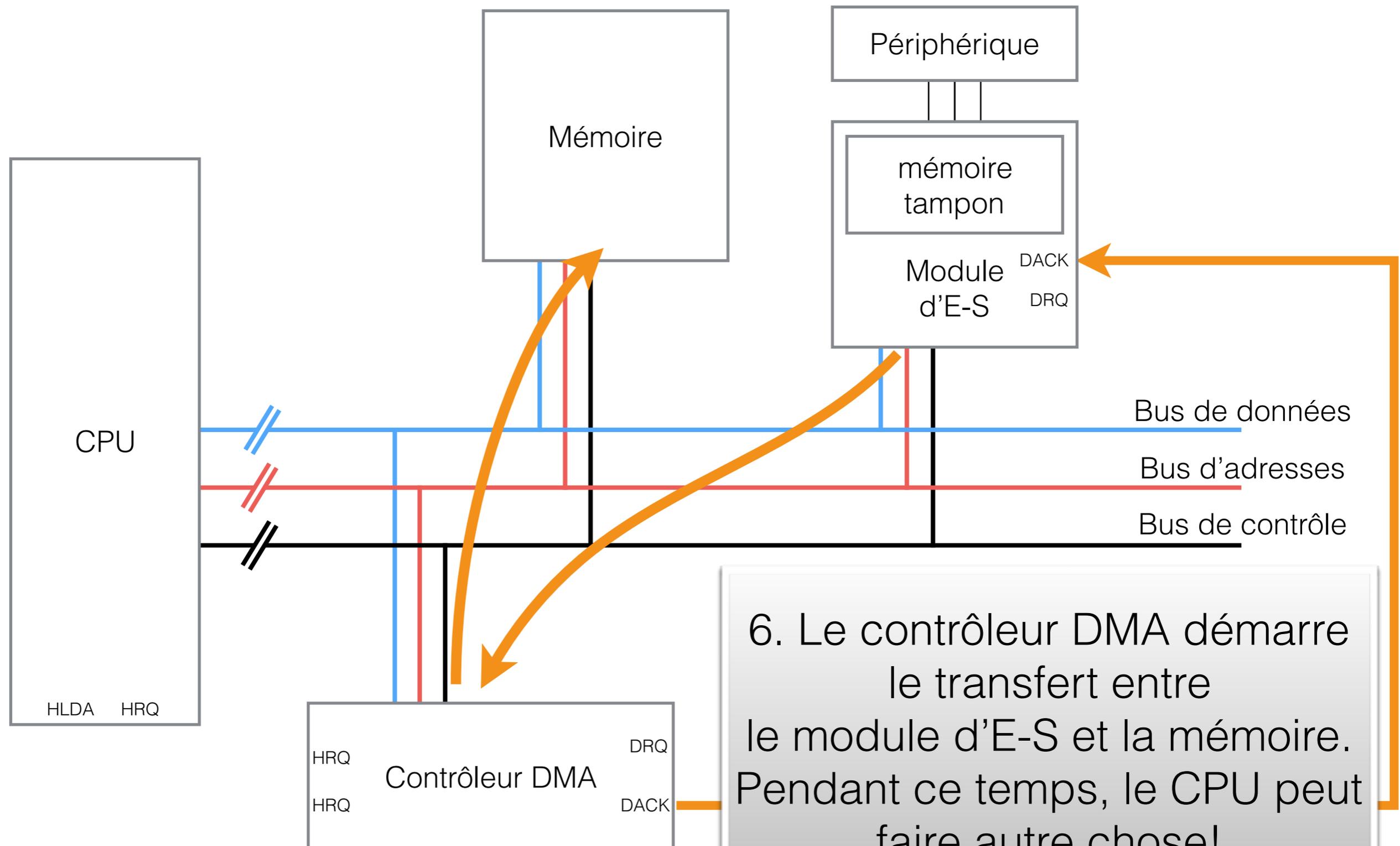


4. Le contrôleur DMA demande au microprocesseur de lui céder le contrôle des bus.

Direct Memory Access (DMA)



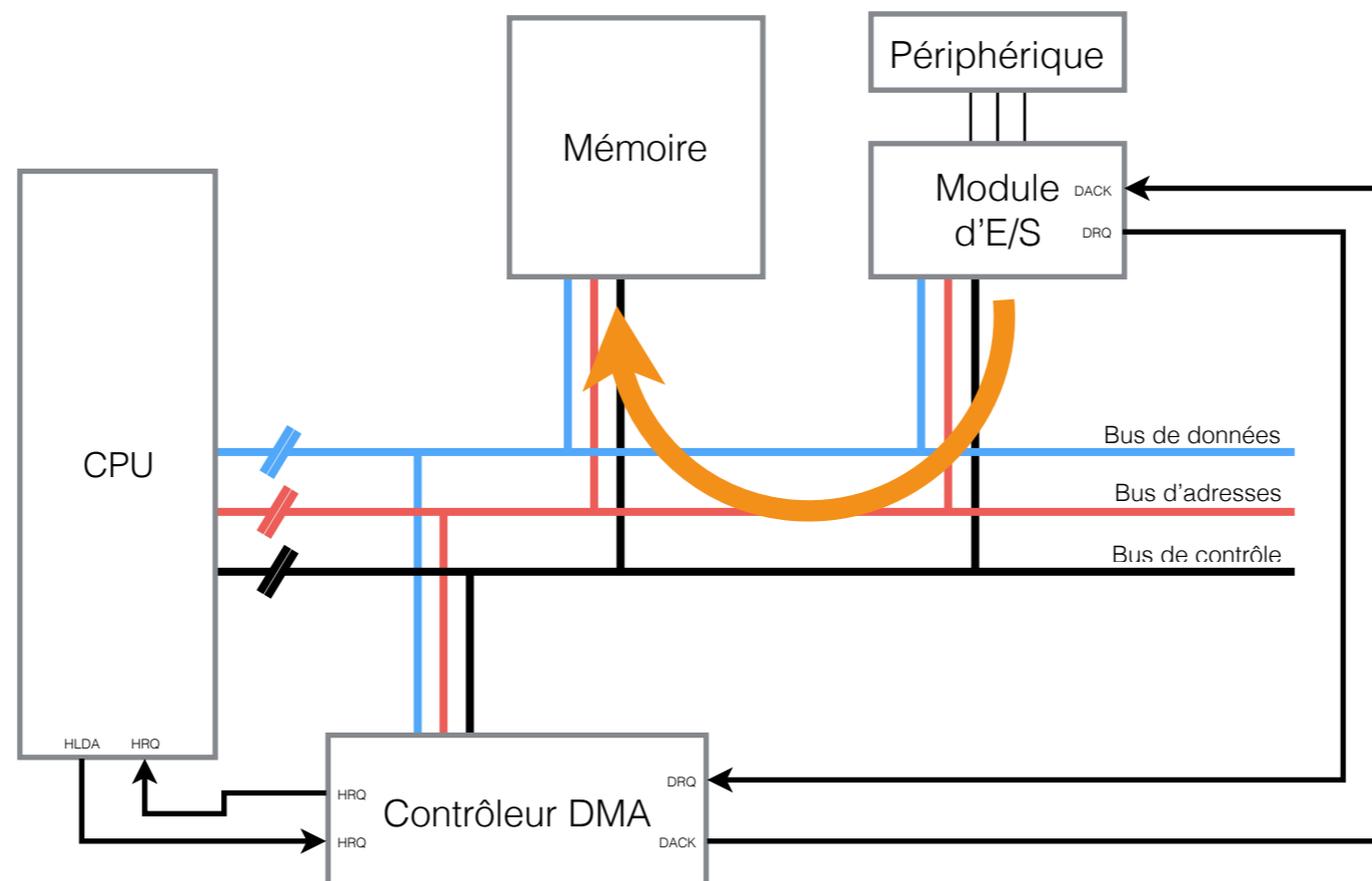
Direct Memory Access (DMA)



6. Le contrôleur DMA démarre le transfert entre le module d'E-S et la mémoire. Pendant ce temps, le CPU peut faire autre chose!

Direct Memory Access (DMA)

- Le Direct Memory Access (DMA) est un transfert de données direct entre un périphérique et la mémoire ou vice versa, effectué *sans intervention du microprocesseur*.
- Le contrôleur de DMA est un circuit intégré qui gère le transfert par DMA.
- Le transfert par DMA se fait par bloc de données, les données étant à des adresses contigües de la mémoire et provenant habituellement du même périphérique.



Direct Memory Access (DMA)

- Pour accéder à un périphérique, on exécute les étapes suivantes:
 1. Envoyer une requête au périphérique
 1. Le périphérique démarre la requête—ce qui peut être long à compléter!
 2. Pendant ce temps, le microprocesseur peut faire autre chose.
 2. Lorsque la requête est terminée, le périphérique informe le contrôleur de DMA
 1. Le contrôleur de DMA demande au CPU s'il peut contrôler les bus.
 3. Le DMA gère le transfert mémoire
 1. La copie est effectuée entre la mémoire tampon du périphérique vers la RAM
 2. Pendant ce temps, le microprocesseur peut faire autre chose.

Direct Memory Access (DMA)

- Avantages
 - Beaucoup plus **efficace**. Le microprocesseur n'a pas à attendre après le périphérique, et n'a pas à gérer le transfert mémoire.
- Désavantages?
 - Plus **complexe**. Nécessite une communication entre le CPU et le contrôleur de DMA.

Direct Memory Access (DMA)

- Les transferts par DMA sont configurés par des instructions exécutées par le microprocesseur. Habituellement, le microprocesseur configure le transfert de DMA en exécutant ces instructions. Au minimum, les instructions indiquent:
 - Le périphérique visé
 - La plage de mémoire visée (l'adresse de départ et le nombre de données à transférer)
 - La vitesse de transfert, la taille des données transférées et le mode de transfert.
 - Le mode de transfert est la façon dont sera déclenchée le transfert. Seul le mode de transfert "unique"—un seul bloc de données—sera vu dans le cadre du cours.
- Une fois la configuration effectuée, le transfert par DMA débute automatiquement lors d'un signal de périphérique ou lorsque le microprocesseur exécute une instruction initiant le transfert.
- Lors de l'initialisation d'un transfert par DMA, le contrôleur de DMA négocie l'accès au bus de données avec le microprocesseur.

Direct Memory Access (DMA)

- Lors d'un transfert par DMA, le contrôleur de DMA gère les bus de contrôle d'adresse et de données pour effectuer le transfert.
 - Le microprocesseur et le contrôleur de DMA ne peuvent contrôler les bus en même temps.
 - Le transfert s'effectue habituellement avec un compteur sur le bus d'adresse. Lors du transfert de chaque données, le compteur d'adresse est incrémenté à partir de l'adresse de base du bloc de mémoire visé.
- Le contrôleur de DMA possède souvent un tampon de mémoire pour emmagasiner les données échangées entre la mémoire et un périphérique. Le transfert se fait en deux temps: lecture de données puis écriture de données.
- Le transfert par DMA est beaucoup plus rapide que le transfert avec des instructions LDR/STR et il soulage le microprocesseur de cette tâche:
 - Entre le transfert de chaque données, le microprocesseur n'a pas à lire et exécuter des instructions.
 - Le transfert se fait par bloc, directement de la mémoire au périphérique ou vice versa: les données ne sont pas entreposées temporairement dans des registres du microprocesseur.
 - Pendant le transfert, le microprocesseur peut effectuer d'autres tâches.

Direct Memory Access (DMA)

- Dans les systèmes modernes, le microprocesseur n'est pas arrêté pendant le transfert par DMA.
- Cependant, le microprocesseur et le contrôleur de DMA pourraient avoir besoin des bus en même temps. Que faire?
 - Deux bus d'accès à la mémoire peuvent permettre au contrôleur de DMA et au microprocesseur d'accéder simultanément à la mémoire.
 - L'accès au bus de données par le contrôleur de DMA est subordonné à celui du microprocesseur: le contrôleur de DMA effectue le transfert entre chaque moment d'utilisation du bus par le microprocesseur.
 - Les caches (que nous verrons la semaine prochaine) contiennent des données utilisées par le microprocesseur alors que le transfert par DMA modifie la mémoire seulement.
- Le DMA est utilisé abondamment dans vos ordinateurs
 - transfert de données entre la mémoire et le disque dur
 - transfert de données vers la carte graphique ou la carte de son, etc.

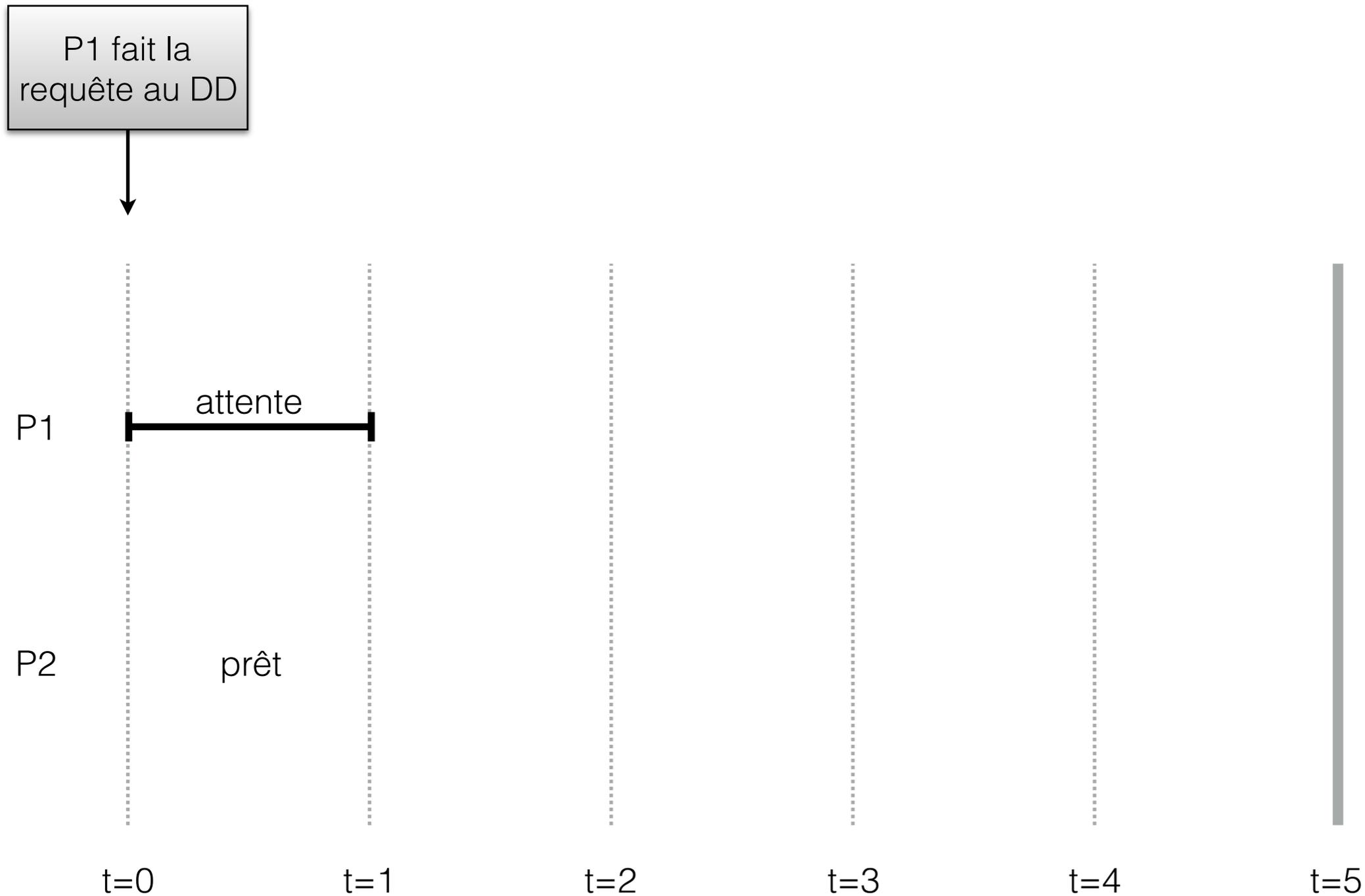
Entrées-sorties: 3 stratégies

Stratégie	1. Requête	2. Transfert
Programmées	Le CPU interroge le contrôleur d'E-S.	Effectué par le CPU .
Interruptions	Le contrôleur d'E-S soulève une interruption.	Effectué par le CPU .
DMA	Le contrôleur d'E-S informe le contrôleur de DMA.	Effectué par le contrôleur de DMA.

Exemple: E/S & ordonnancement

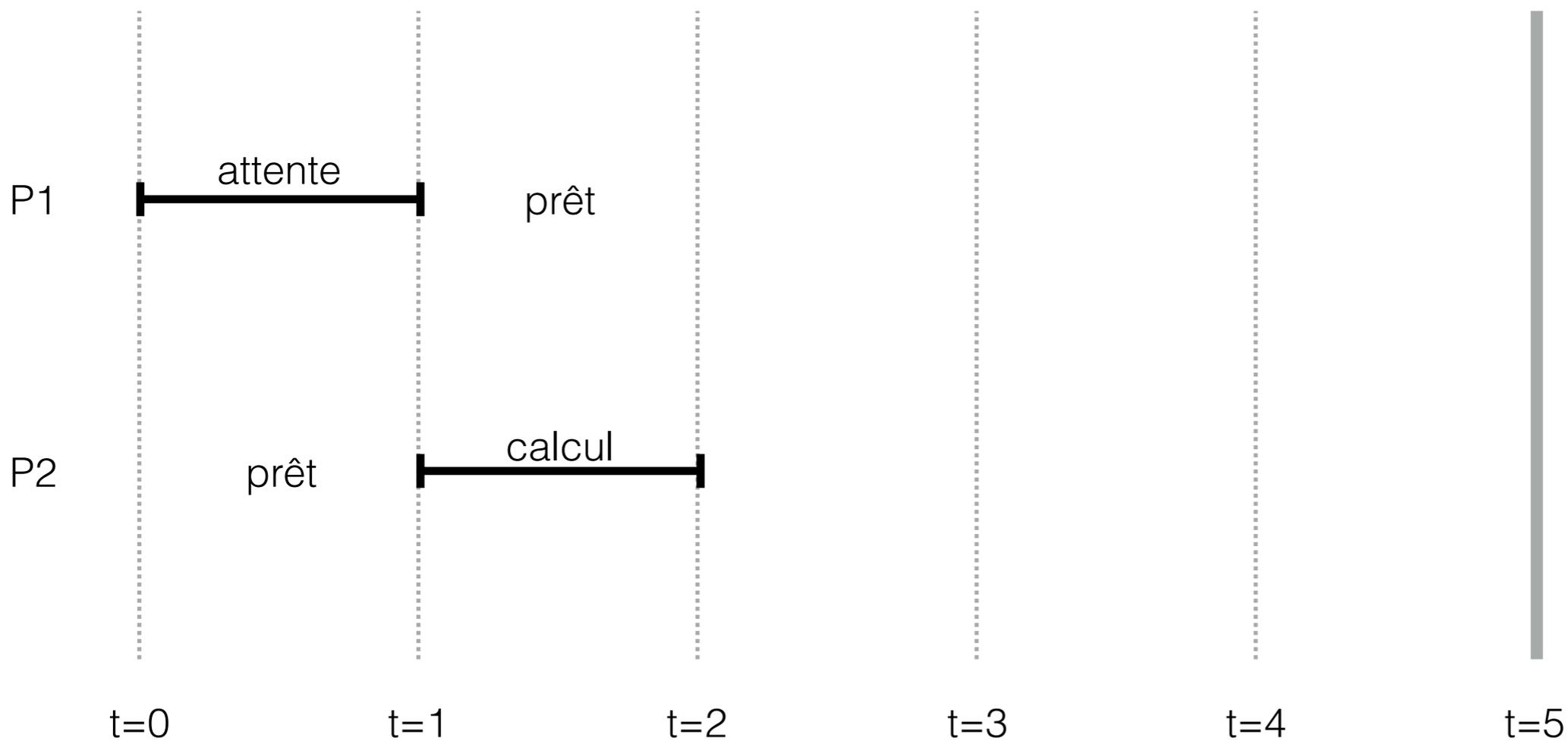
- Supposons un ordinateur possédant les caractéristiques suivantes:
 - L'algorithme d'ordonnancement employé est le "tourniquet"
 - Le « tick » d'horloge survient à toutes les millisecondes (ms)
- Deux processus, P1 et P2, sont admis. Ces deux processus possèdent les caractéristiques suivantes:
 - P1: commence par demander un transfert de données entre le disque dur et la RAM. Lorsque le transfert est complété, le processus poursuit avec des calculs sur les données transférées.
 - P2: Effectue des calculs sur les données déjà en mémoire.
- Nous faisons les hypothèses suivantes:
 - Le temps d'exécution de l'algorithme d'ordonnancement et le temps d'accès des données en RAM est de 0ms (nous pouvons les ignorer);
 - Le temps de réponse du disque dur à une requête de transfert est de 2.5ms;
 - Une fois la réponse du disque dur reçue, le temps de transfert des données est de 1ms;
 - Une interruption IRQ ne fait que modifier l'état du processus qui a fait la requête et ne démarre pas le transfert.
- Question: combien de temps est passé au calcul (par P1 et P2) pour chacune des 3 stratégies de gestion des entrées-sorties vues dans le cours, dans les 5 premières millisecondes d'exécution?

Exemple: E/S programmées



Exemple: E/S programmées

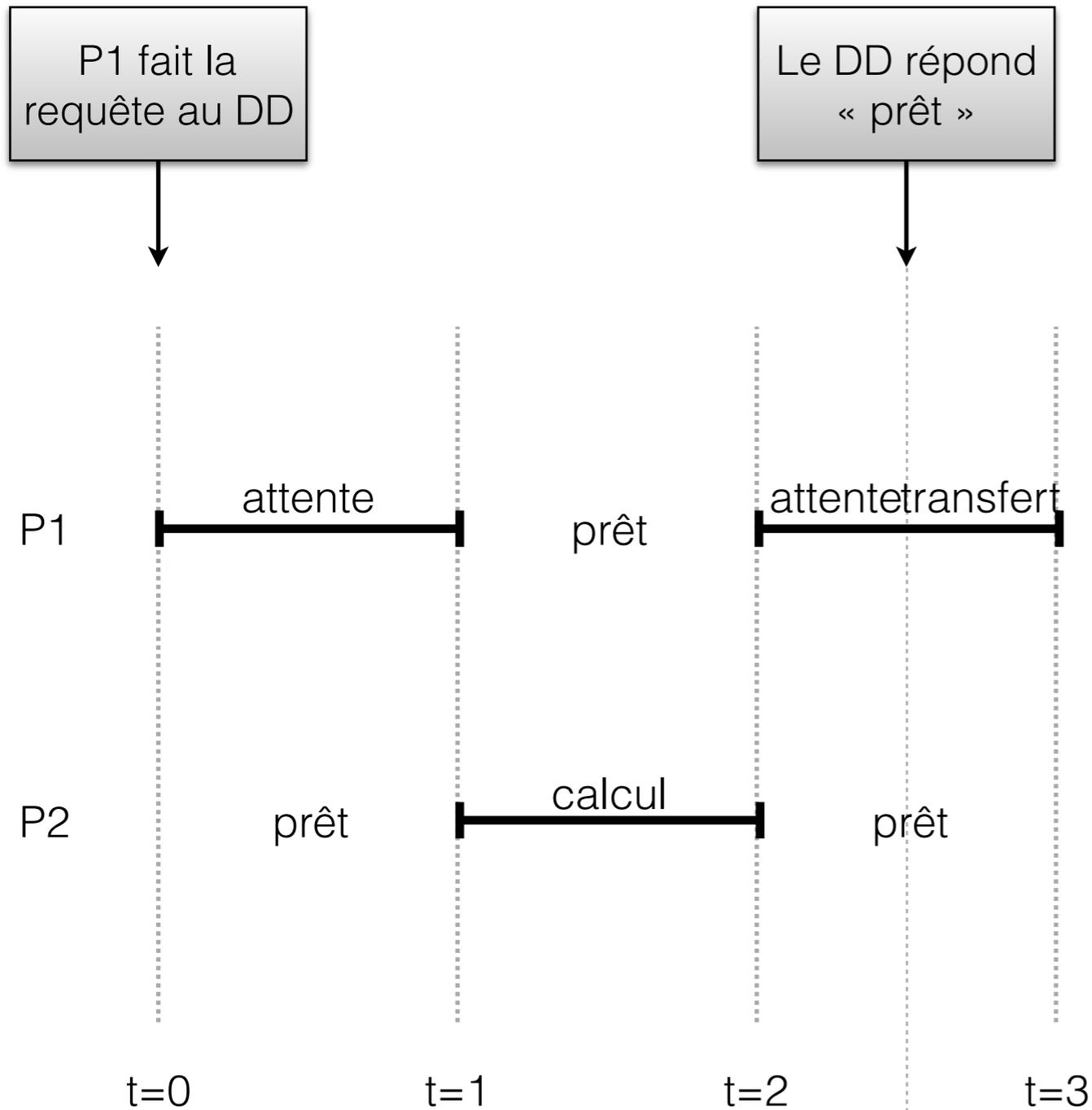
P1 fait la
requête au DD



Rappel

Le temps de réponse du disque dur à une requête de transfert est de 2.5ms.

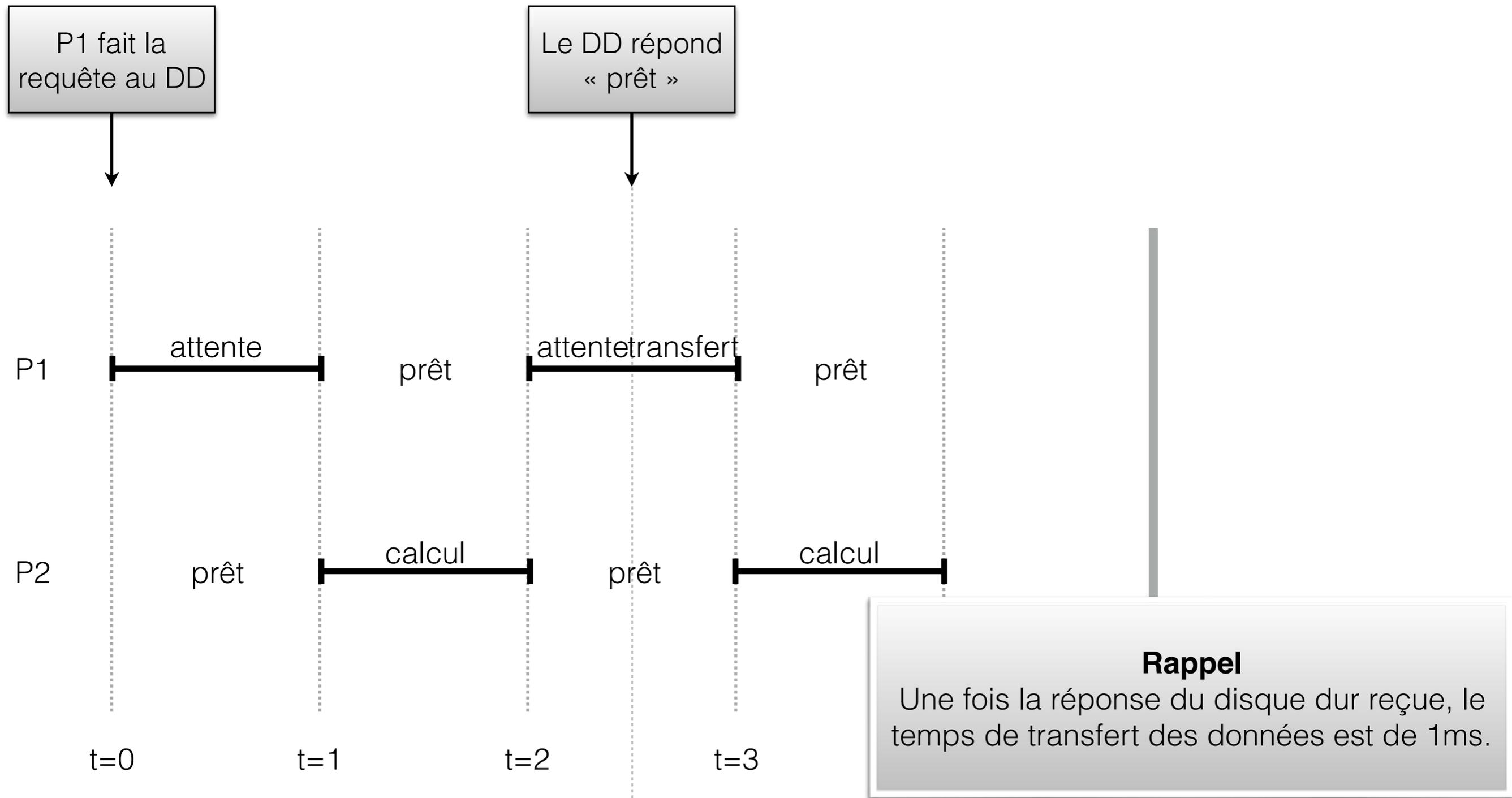
Exemple: E/S programmées



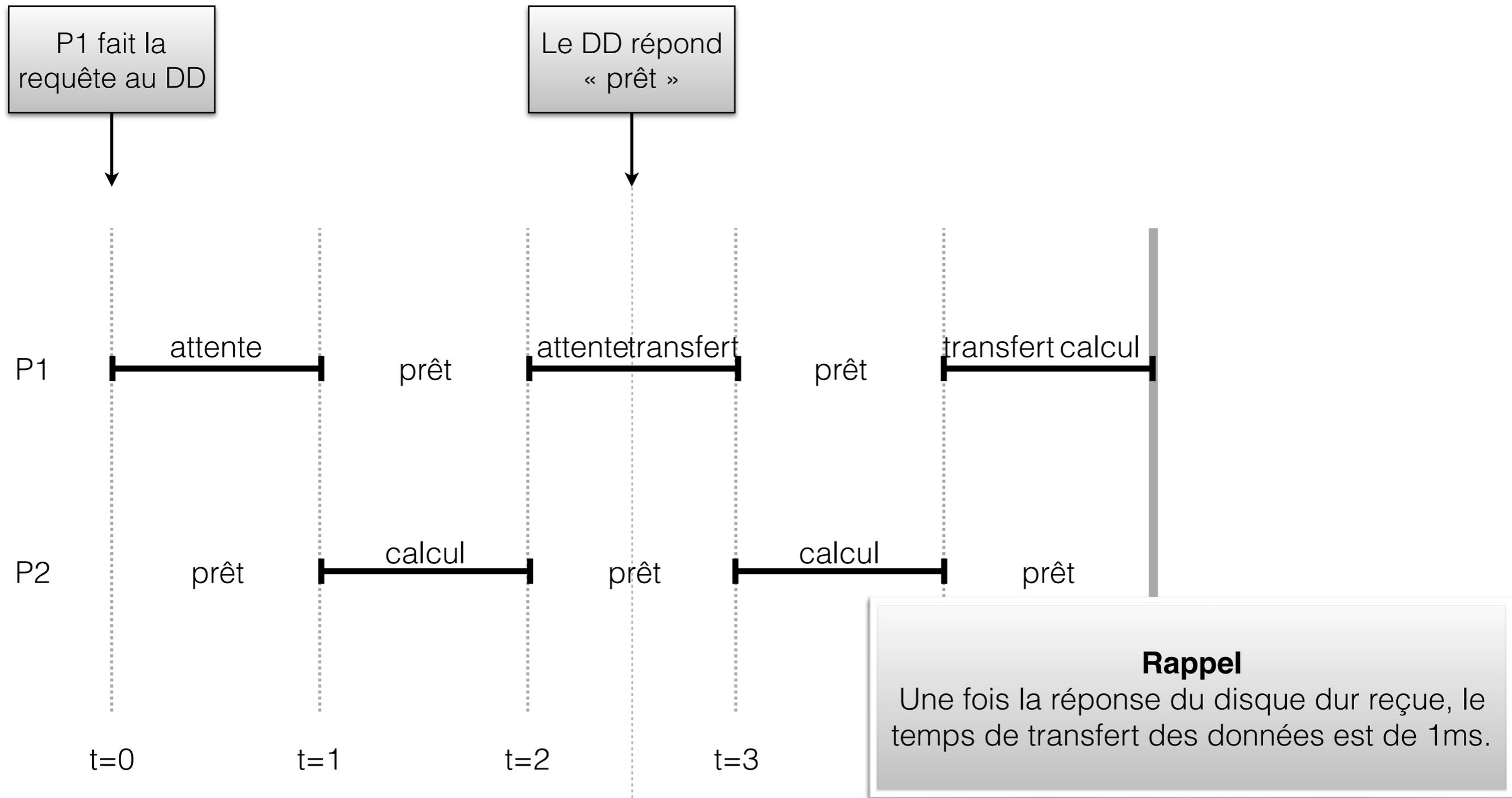
Rappel
Le temps de réponse du disque dur à une requête de transfert est de 2.5ms.

Rappel
Une fois la réponse du disque dur reçue, le temps de transfert des données est de 1ms.

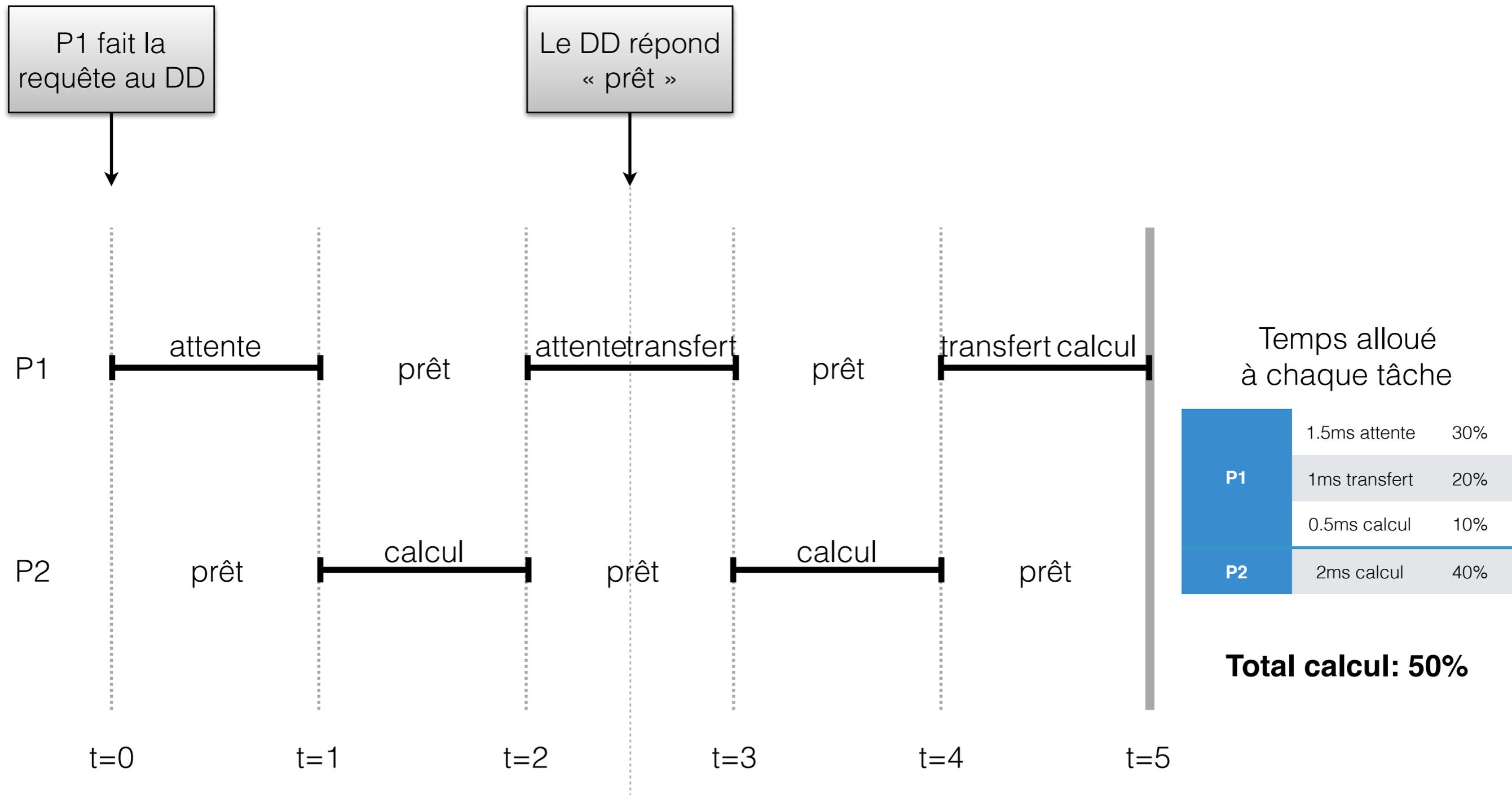
Exemple: E/S programmées



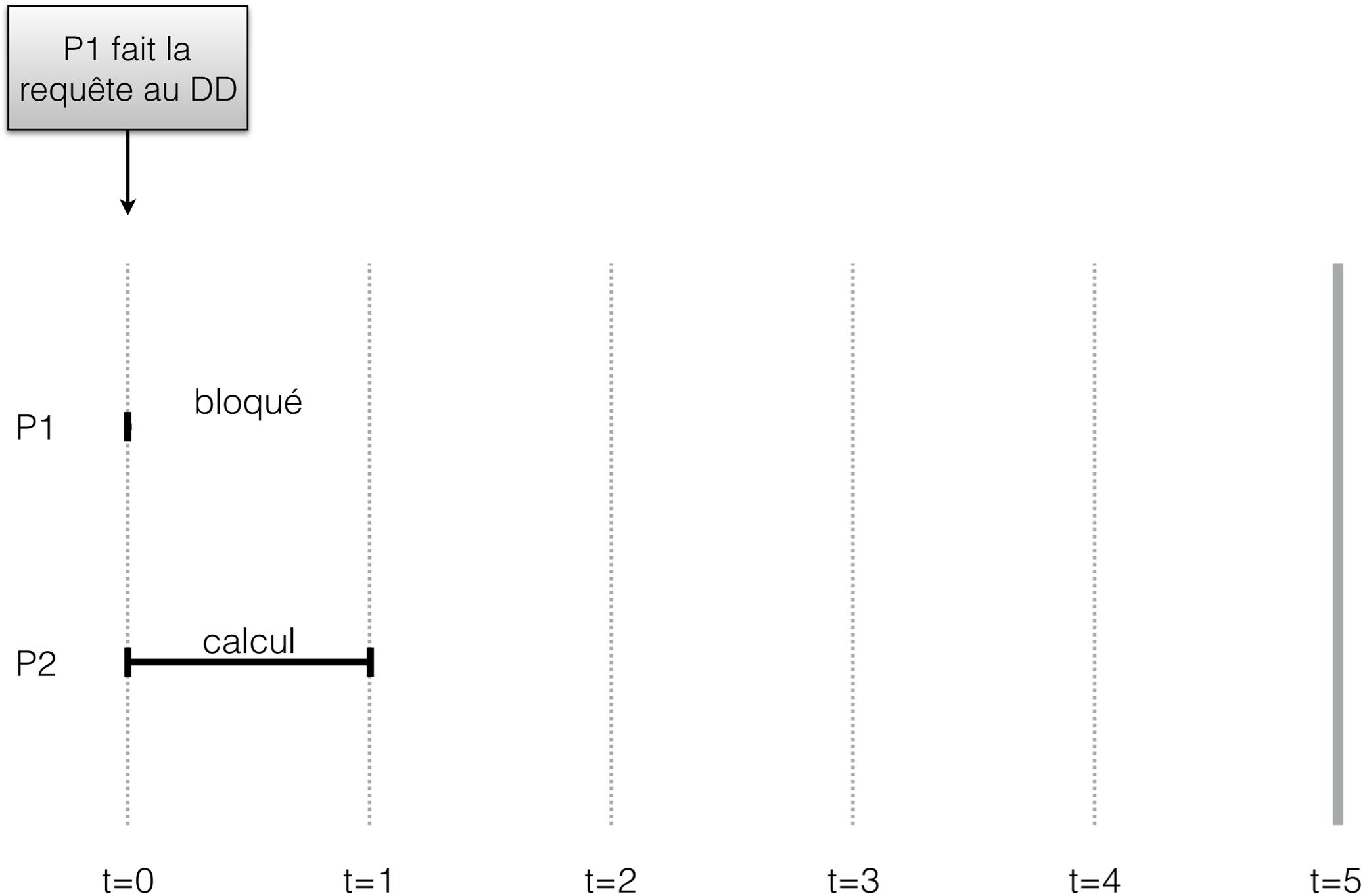
Exemple: E/S programmées



Exemple: E/S programmées

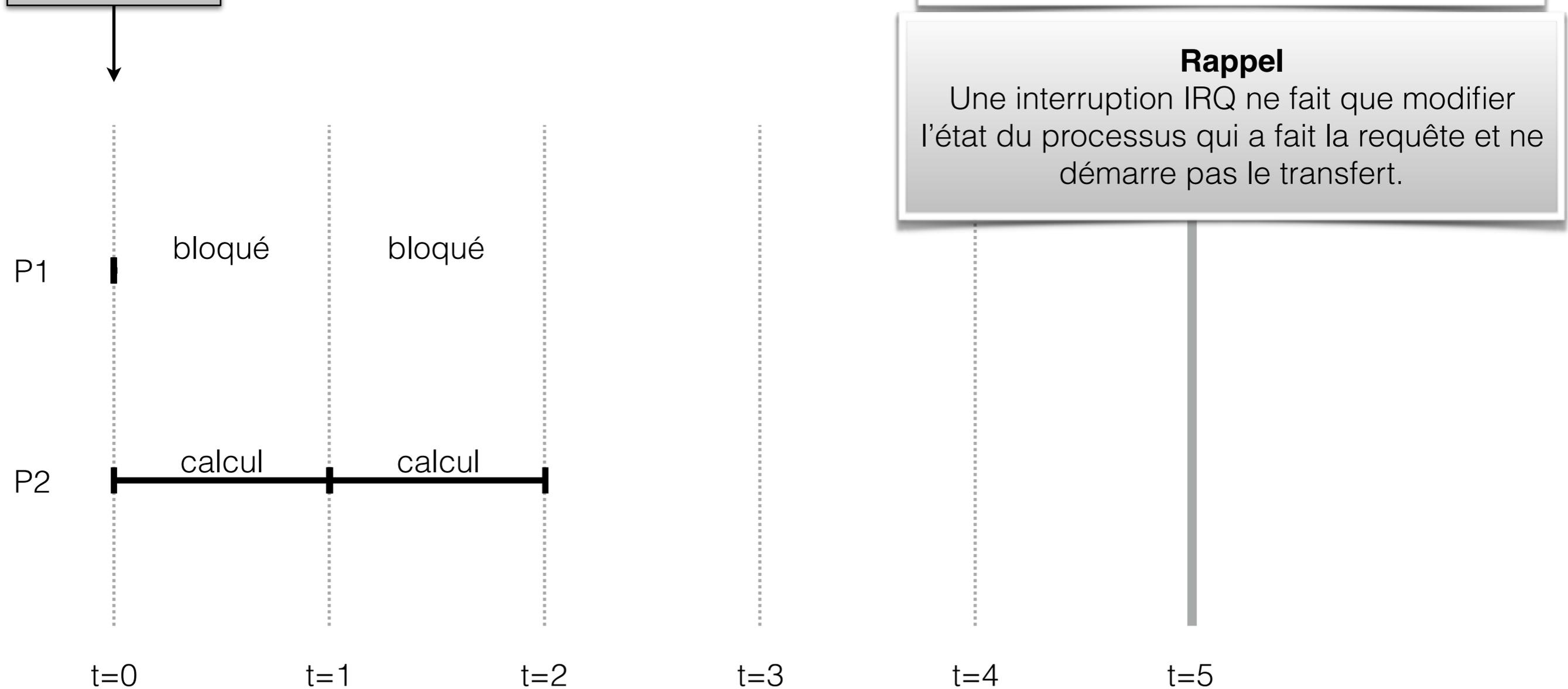


Exemple: E/S par interruptions



Exemple: E/S par interruptions

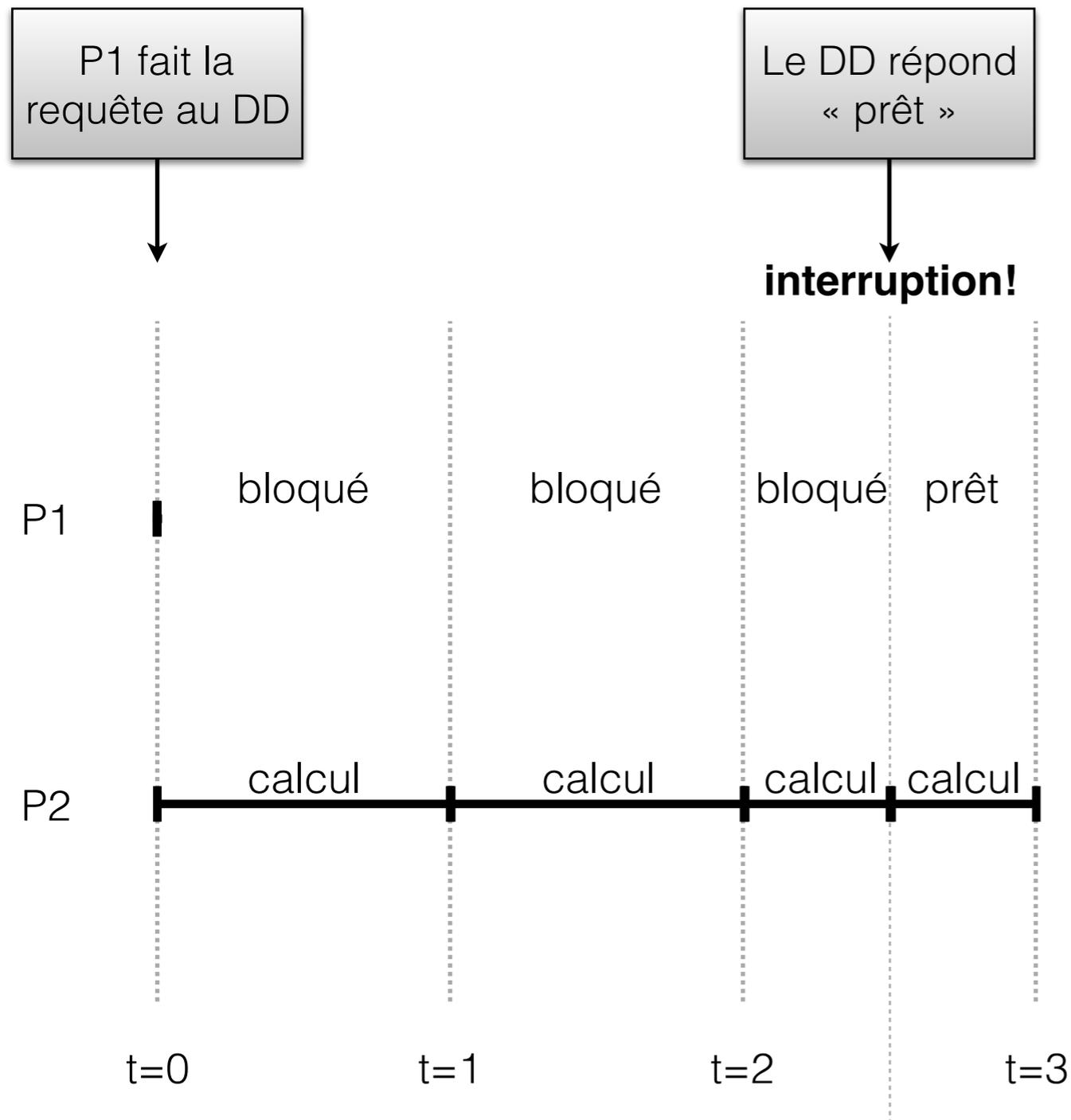
P1 fait la
requête au DD



Rappel
Le temps de réponse du disque dur à une requête de transfert est de 2.5ms.

Rappel
Une interruption IRQ ne fait que modifier l'état du processus qui a fait la requête et ne démarre pas le transfert.

Exemple: E/S par interruptions

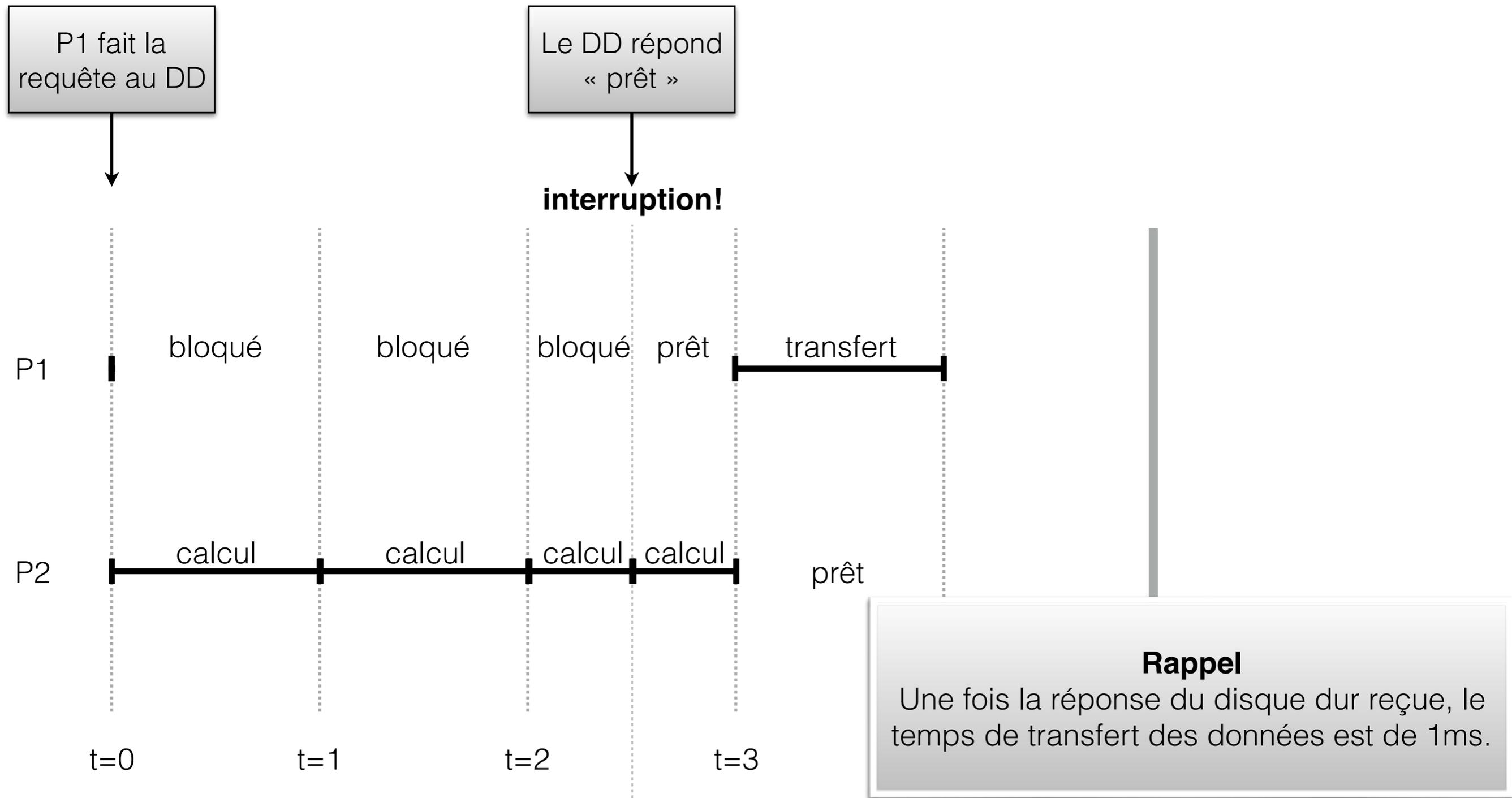


Rappel
Le temps de réponse du disque dur à une requête de transfert est de 2.5ms.

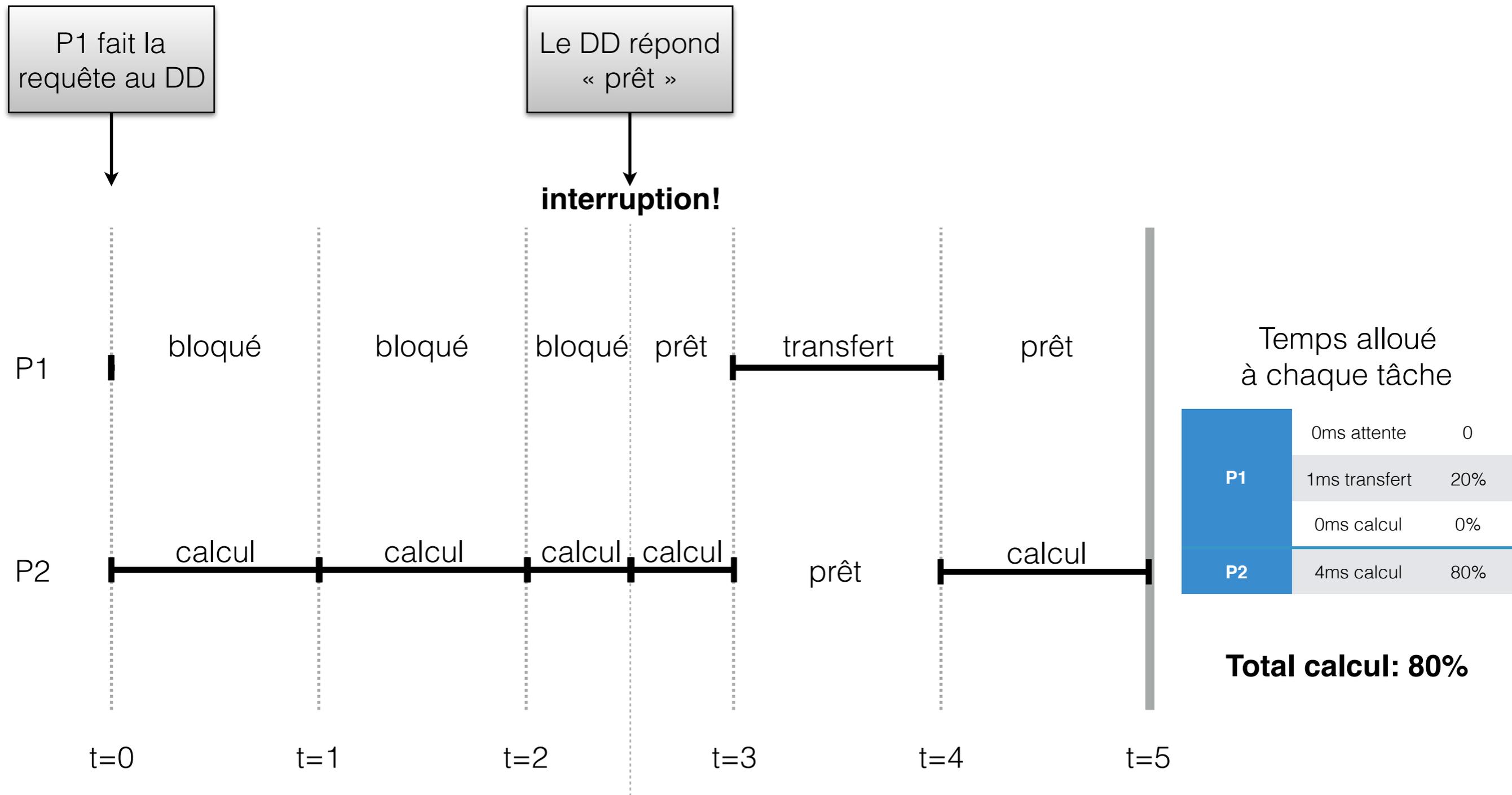
Rappel
Une interruption IRQ ne fait que modifier l'état du processus qui a fait la requête et ne démarre pas le transfert.

Rappel
Une fois la réponse du disque dur reçue, le temps de transfert des données est de 1ms.

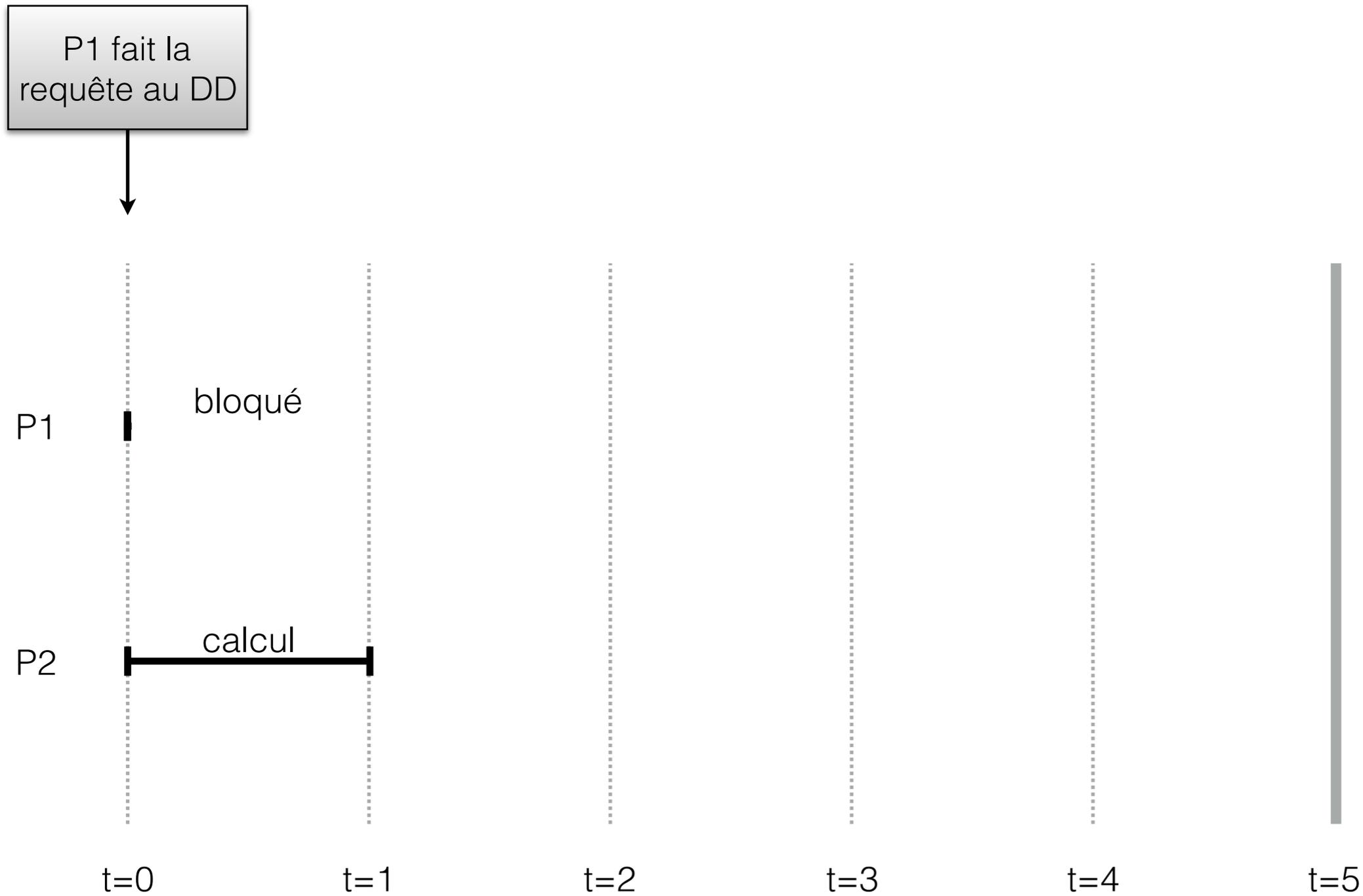
Exemple: E/S par interruptions



Exemple: E/S par interruptions

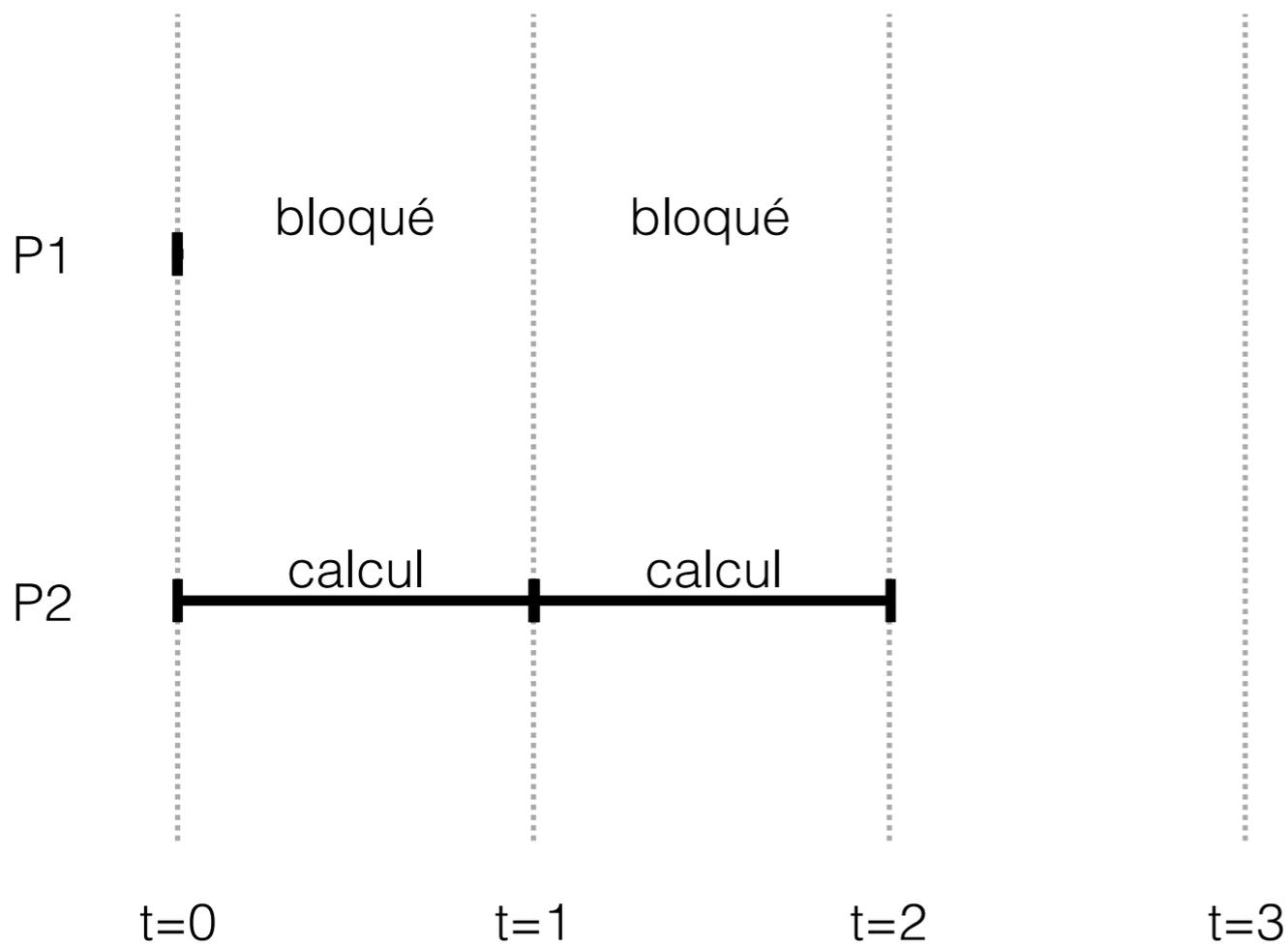


Exemple: E/S par DMA



Exemple: E/S par DMA

P1 fait la
requête au DD



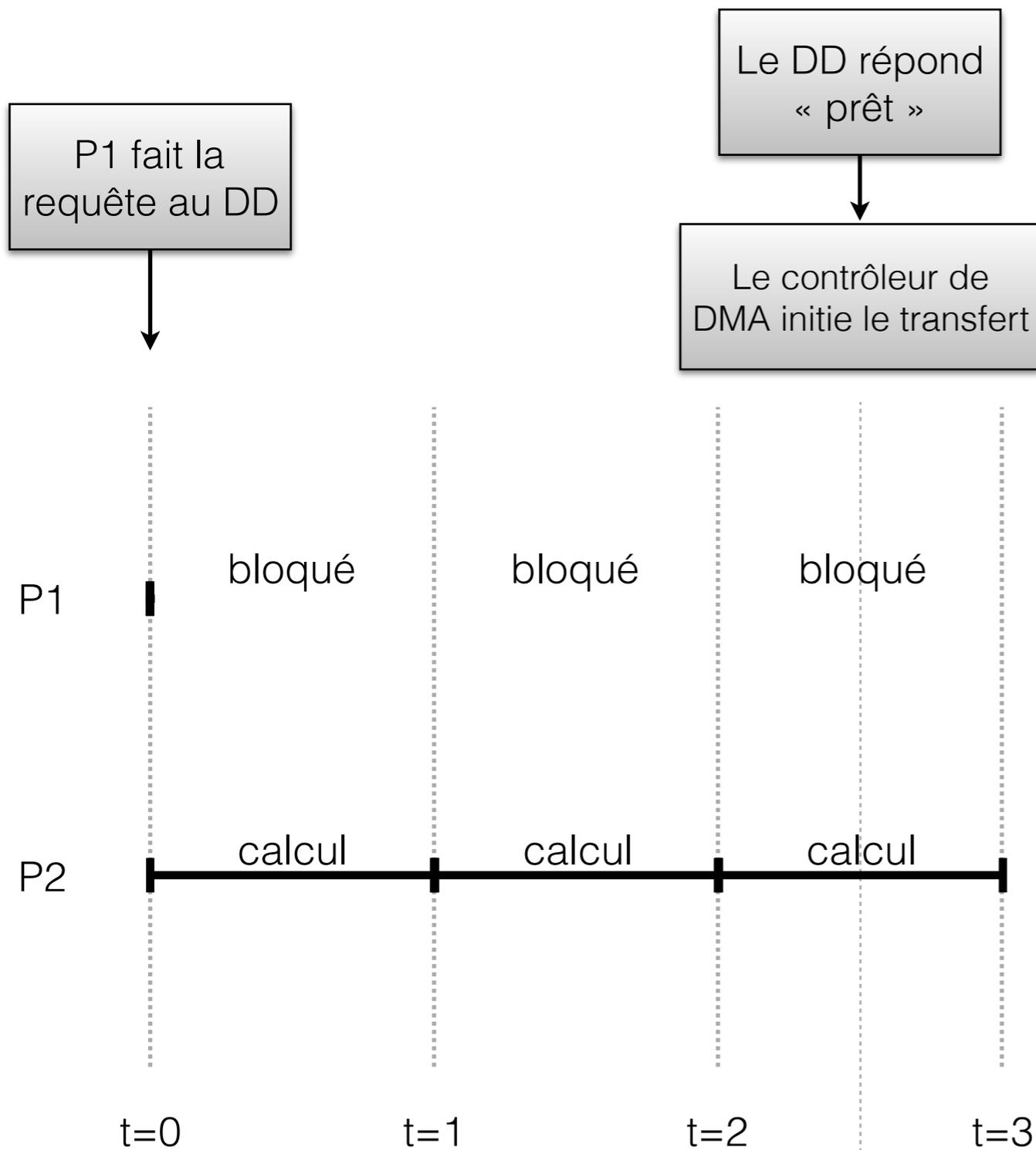
Rappel

Le temps de réponse du disque dur à une requête de transfert est de 2.5ms.

Rappel

Une fois la réponse du disque dur reçue, le temps de transfert des données est de 1ms.

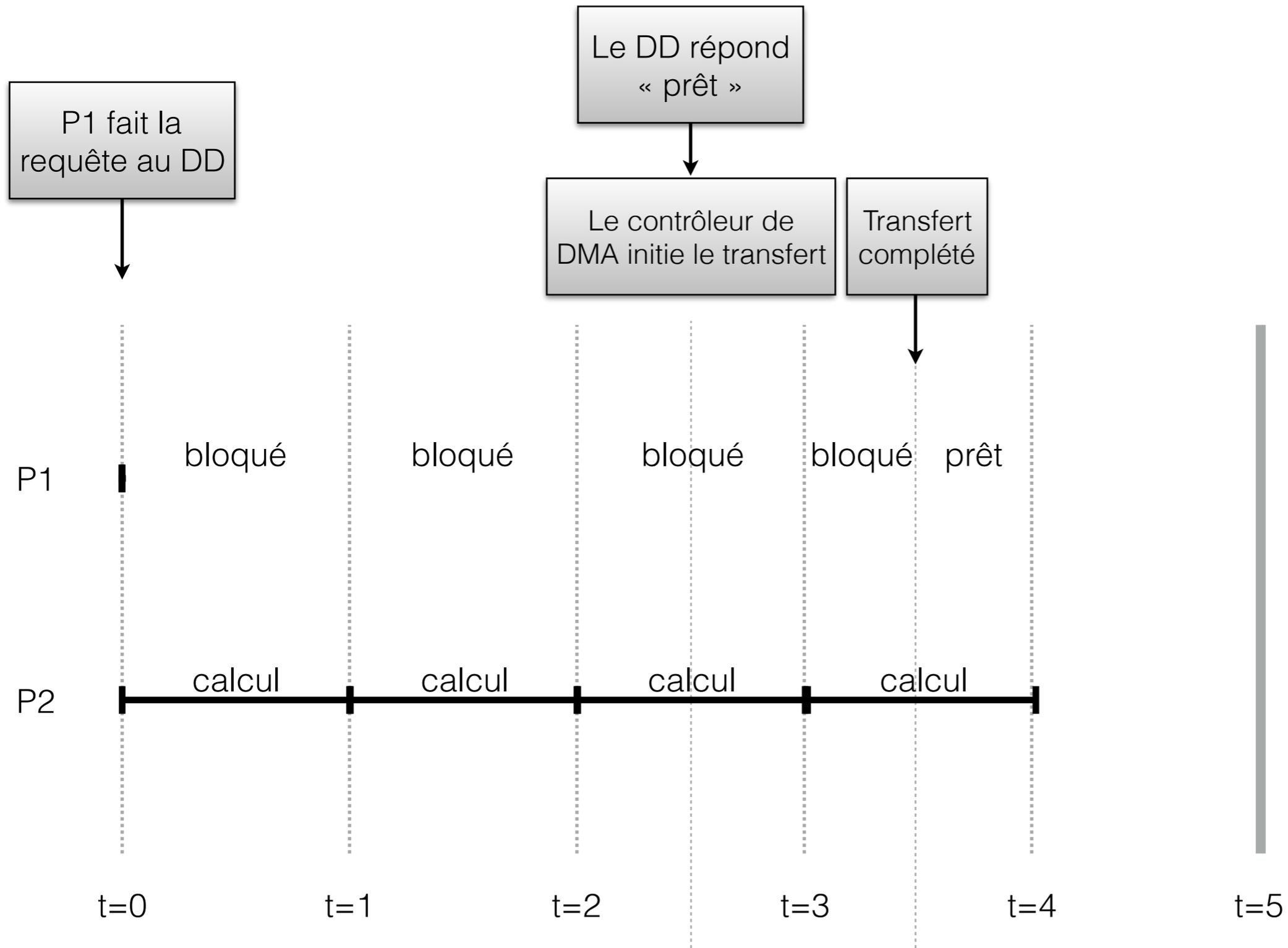
Exemple: E/S par DMA



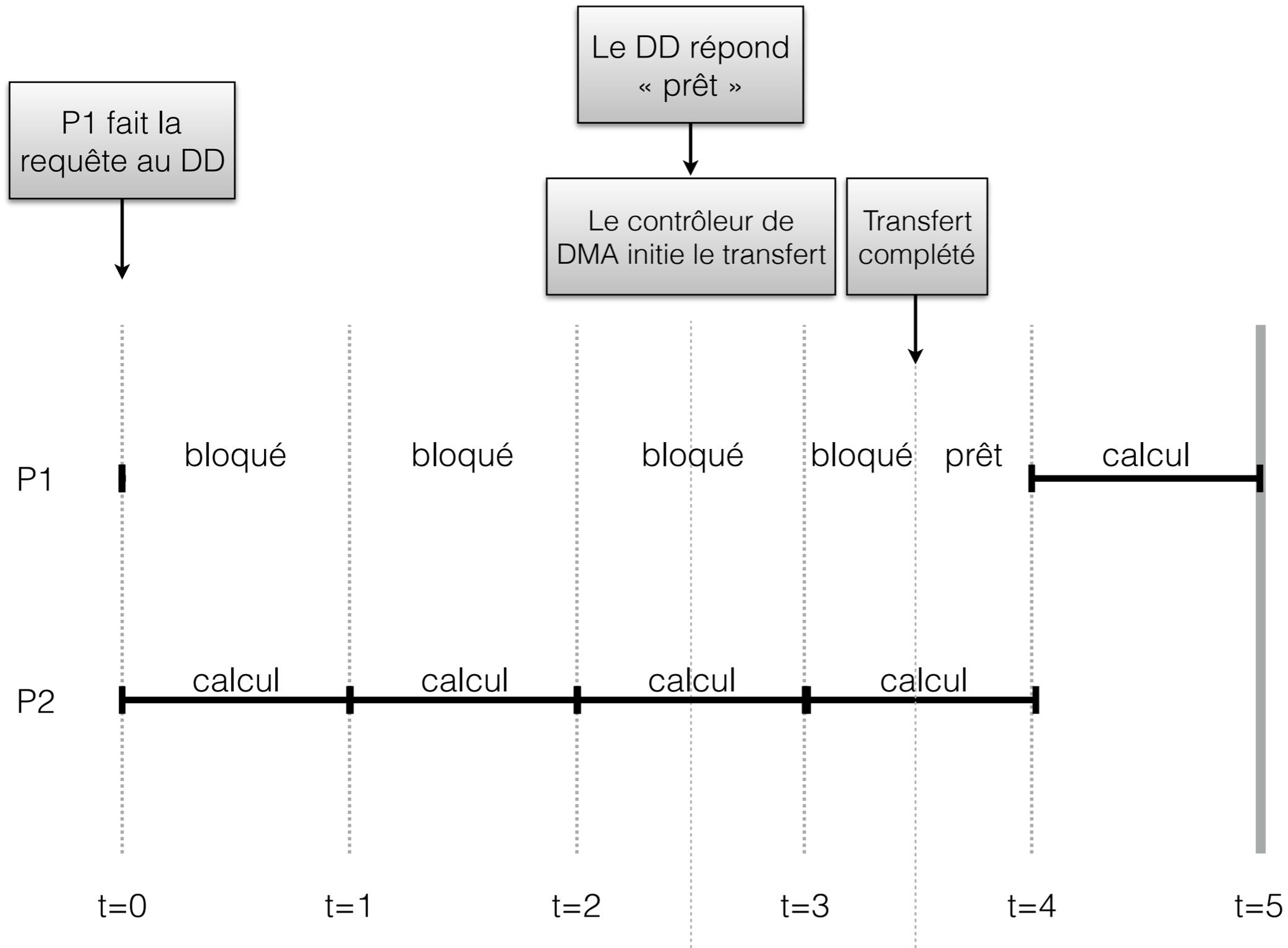
Rappel
Le temps de réponse du disque dur à une requête de transfert est de 2.5ms.

Rappel
Une fois la réponse du disque dur reçue, le temps de transfert des données est de 1ms.

Exemple: E/S par DMA



Exemple: E/S par DMA



Exemple: E/S par DMA

