

# Les Entrées et Sorties: Programmées, interruptions, et DMA



# Aujourd'hui

- Entrées-sorties
  - Programmées
  - Par interruptions
  - Le DMA

# Les Entrées/Sorties

- Entrées/Sorties (E/S, Input/Output ou I/O en anglais) désigne l'ensemble des transferts de données qui permettent au microprocesseur et à la mémoire de communiquer avec le reste du monde:
  - **entrée**: une donnée allant du monde extérieur vers le microprocesseur
  - **sortie**: une donnée allant du microprocesseur vers le monde extérieur.
- Les Entrées/Sorties proviennent des périphériques.
  - Un périphérique est un appareil qui interagit avec microprocesseur et la mémoire. Certains périphériques sont branchés à l'intérieur de l'ordinateur (disques durs, carte réseau,...) alors que d'autres sont branchés sur des interfaces externes de l'ordinateur (clavier, écrans, souris, etc.).

# Types de périphériques

Nom	Entrée ou sortie?	Vitesse des données	Contrôle	Type de données
Clavier	Entrée	Très faible	Externe & programmes	Caractères
Souris				
Souris tactile	Entrée	Faible	Externe	Caractères
Écran tactile				
Scanneur	Entrée	Moyenne	Externe et programme	Blocs
Voix	Entrée	Faible à moyenne	Externe et programme	Blocs
Son	Entrée-sortie	Moyenne	Programme	Blocs ou continu
USB	Entrée-sortie	Faible à très élevée	Externe et programme	Blocs
Réseau	Entrée-sortie	Élevée à très élevée	Externe et programme	Blocs
Imprimante	Sortie	Faible à moyenne	Programme	Blocs
Carte graphique	Sortie	Élevée	Programme	Continu
Disque flash	Stockage	Moyenne	Externe et programme	Blocs
Disque magnétique	Stockage	Moyenne	Programme	Blocs
SSD	Stockage	Moyenne à élevée	Programme	Blocs
Disque optique	Stockage	Moyenne à élevée	Externe et programme	Blocs ou continu
Tape magnétique	Stockage	Faible à moyenne	Externe et programme	Blocs ou continu

# Les Entrées/Sorties

- Pourquoi le traitement des E/S est-il complexe?
  - Les périphériques ont des modes de fonctionnement variés.
  - Les périphériques ont souvent leur propres formats de données.
  - Les périphériques ne vont pas à la même vitesse que le CPU. Beaucoup sont très lents par rapport à ce dernier alors que certains sont plus rapides...
- Pour chaque périphérique, il existe une unité spéciale appelée module d'E/S (I/O module) qui sert d'interface entre le périphérique et le microprocesseur.

# Modules d'Entrées/Sorties

- Les modules d'E/S sont des interfaces entre le microprocesseur et un périphérique spécifique.
- Ces modules sont habituellement appelés “contrôleurs”.
- Par exemple, le module d'E/S servant d'interface entre le microprocesseur et un disque dur sera appelé contrôleur de disque.

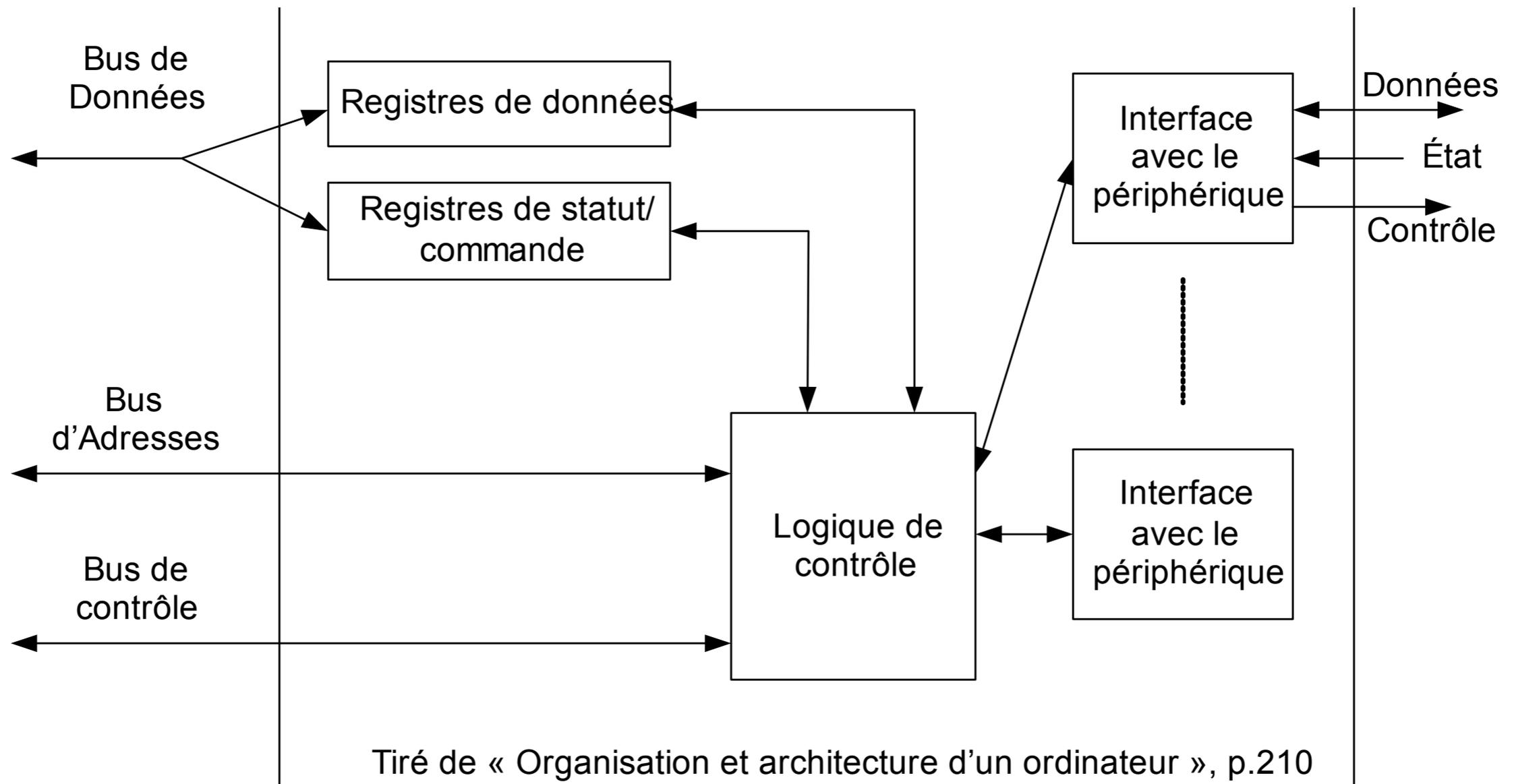
# Modules d'Entrées/Sorties

- Le module comprend:
  - de la mémoire tampon pour des données (sous forme de registres dans l'illustration),
  - une logique de contrôle pour décoder l'adresse et les lignes de contrôle (ou pour faire du DMA—à suivre)
  - une ou plusieurs interfaces avec un ou plusieurs périphériques.
- Certains module d'E/S sont très rudimentaires alors que d'autres sont très complexes (exemples de caractéristiques qui rendent le module complexe: DMA, support pour interruptions, programmes exécutés à l'intérieur du module, etc.).

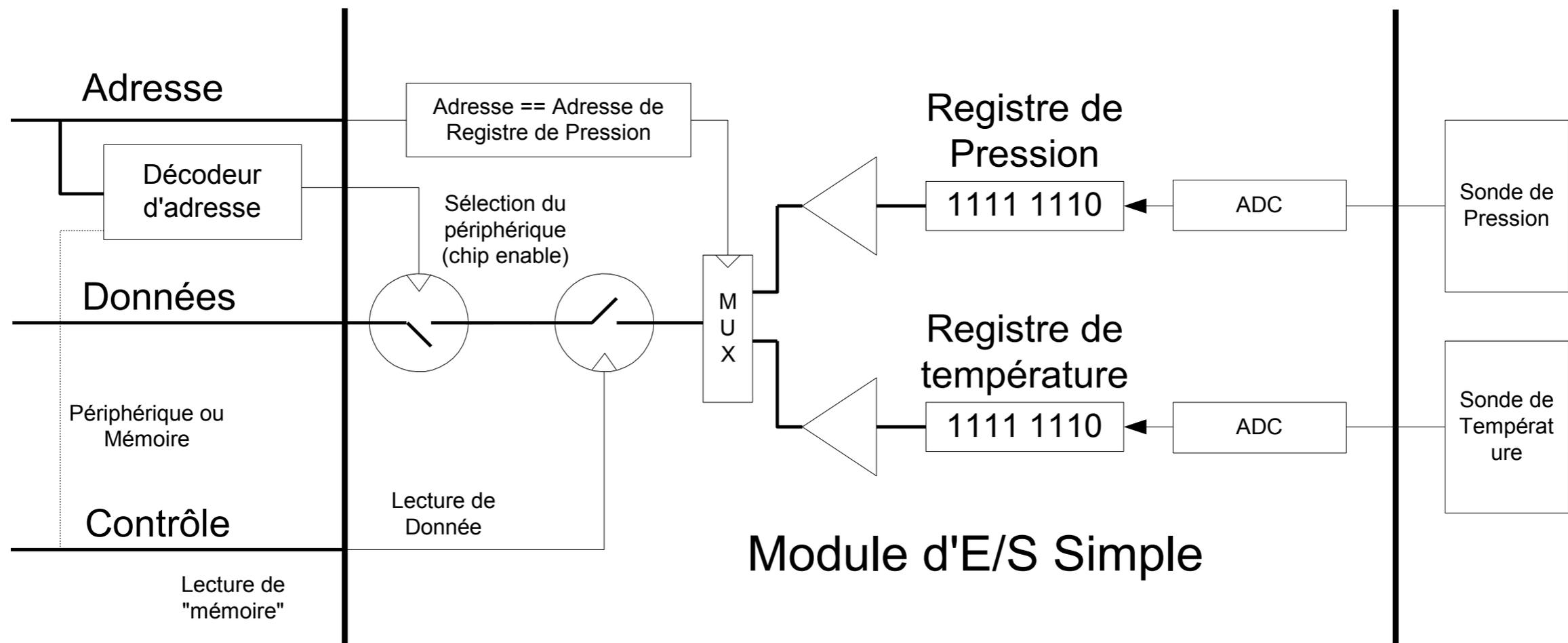
# Modules d'Entrées/Sorties

- Les modules d'E/S ont les fonctions principales:
  - Lire ou écrire des données du périphérique.
  - Lire ou écrire des données du Microprocesseur/Mémoire. Cela implique du décodage d'adresses, de données et de lignes de contrôle. Certains modules d'E/S doivent générer des interruptions ou accéder directement à la 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 du microprocesseur en mémoire tampon afin d'ajuster les vitesses de communication.

# Modules d'Entrées/Sorties

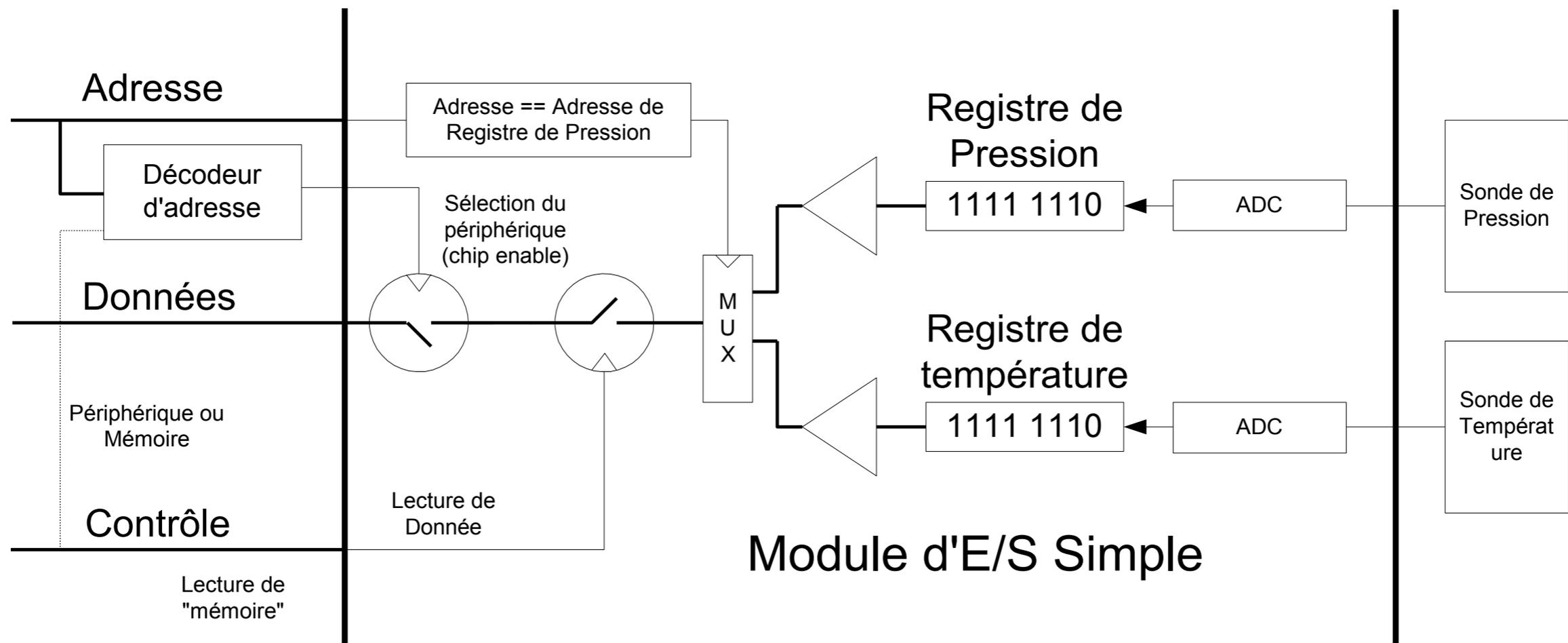


# Exemple — module d'entrées-sorties



- Lors d'une instruction `LDR R0, [R1]` (où R1 contient l'adresse du capteur de pression), les étapes suivantes surviennent:
  - met l'adresse sur le bus d'adresse: le décodeur d'adresse active le module d'E/S via la broche "chip enable", et le MUX sélectionne le registre de pression
  - met le bus de contrôle en lecture
  - la donnée présente dans le registre de pression est placée sur le bus de données, et donc lue par le microprocesseur

# Exemple — module d'entrées-sorties



- Est-ce que le module d'E/S "comprend" l'instruction LDR ou STR?
  - non! il ne fait que réagir aux signaux de contrôle
- Est-ce que le microprocesseur sait comment fonctionne la sonde de pression ou de température?
  - non! c'est le rôle du module d'E/S d'interagir avec les sondes
- Pour le microprocesseur, quelle est la différence entre accéder à une donnée en mémoire RAM, vs. une donnée d'une de ces sondes?
  - aucune différence, dans ce cas nous sommes en "memory-mapped I/O" (comme avec l'architecture ARM), les périphériques ont une adresse, tout comme les emplacements en mémoire

# Entrées/Sorties

- Il existe trois techniques principales pour communiquer à partir du CPU/Mémoire vers un périphérique à travers un module de I/O:
  - E/S programmées
  - E/S avec interruptions
  - le DMA

# Entrées/Sorties programmées

- Pour accéder à un périphérique, le microprocesseur exécute le programme suivant:
  1. Vérifier si le périphérique est prêt
  2. Envoyer une requête au périphérique
  3. Attendre que la requête soit finie en interrogeant les registres de statut du périphérique (polling)
  4. Lorsque la requête est complétée, lire le mot disponible dans le module d'E/S
  5. Recommencer les opérations 2—5 tant que le programme n'est pas terminé...
- Avantages? Inconvénients?

# Entrées/Sorties programmées

- Consiste à concevoir un programme, exécuté par le microprocesseur, qui communiquera avec le module d'E/S afin d'obtenir des données du périphérique.
- Le microproc. peut tester, lire, écrire ou contrôler le périphérique à travers le module d'E/S.
- L'avantage?
  - C'est très simple!
- Désavantages?
  - C'est **lent**. Le microprocesseur doit attendre après le périphérique avant de passer à la prochaine instruction!
  - Il peut y avoir des tâches complexes au niveau matériel qui exigent un contrôle très rapide et très poussé du périphérique. Dans ce cas, il peut devenir impossible de faire un programme qui pourra contrôler le périphérique.

# Entrées/Sorties avec interruptions

- Une façon d'éliminer les délais d'attente du microprocesseur est d'utiliser des interruptions.
- La méthode d'E/S avec interruption consiste à accéder au module d'E/S lors d'interruptions provenant de ce dernier afin de signaler un évènement particulier (la fin d'une tâche par exemple!).

# Éentrées/Sorties avec interruptions

- Rappel de ce qui se produit lorsqu'une interruption survient:
  - Le périphérique émet un signal d'interruption géré par le contrôleur d'interruptions. Ce dernier achemine le signal au microprocesseur.
  - Le microprocesseur termine l'instruction en cours.
  - Le microprocesseur vérifie s'il doit traiter l'interruption en fonction de sa priorité et de ses registres internes. Finalement, il envoie un accusé réception au contrôleur d'interruptions.
  - Le microprocesseur sauvegarde l'emplacement du programme en cours (l'adresse de retour) et toute l'information nécessaire pour reprendre l'exécution du programme en cours (drapeaux par exemple).
  - Le microprocesseur détermine l'adresse de la routine à exécuter afin de traiter l'interruption.
  - Le microprocesseur appelle la routine de traitement de l'interruption (ISR).
  - La routine de traitement de l'interruption est exécutée. En temps normal, l'ISR sauvegarde les registres utilisés dans l'ISR sur la pile et l'ISR se termine avec une instruction de retour vers le programme exécuté avant l'interruption.

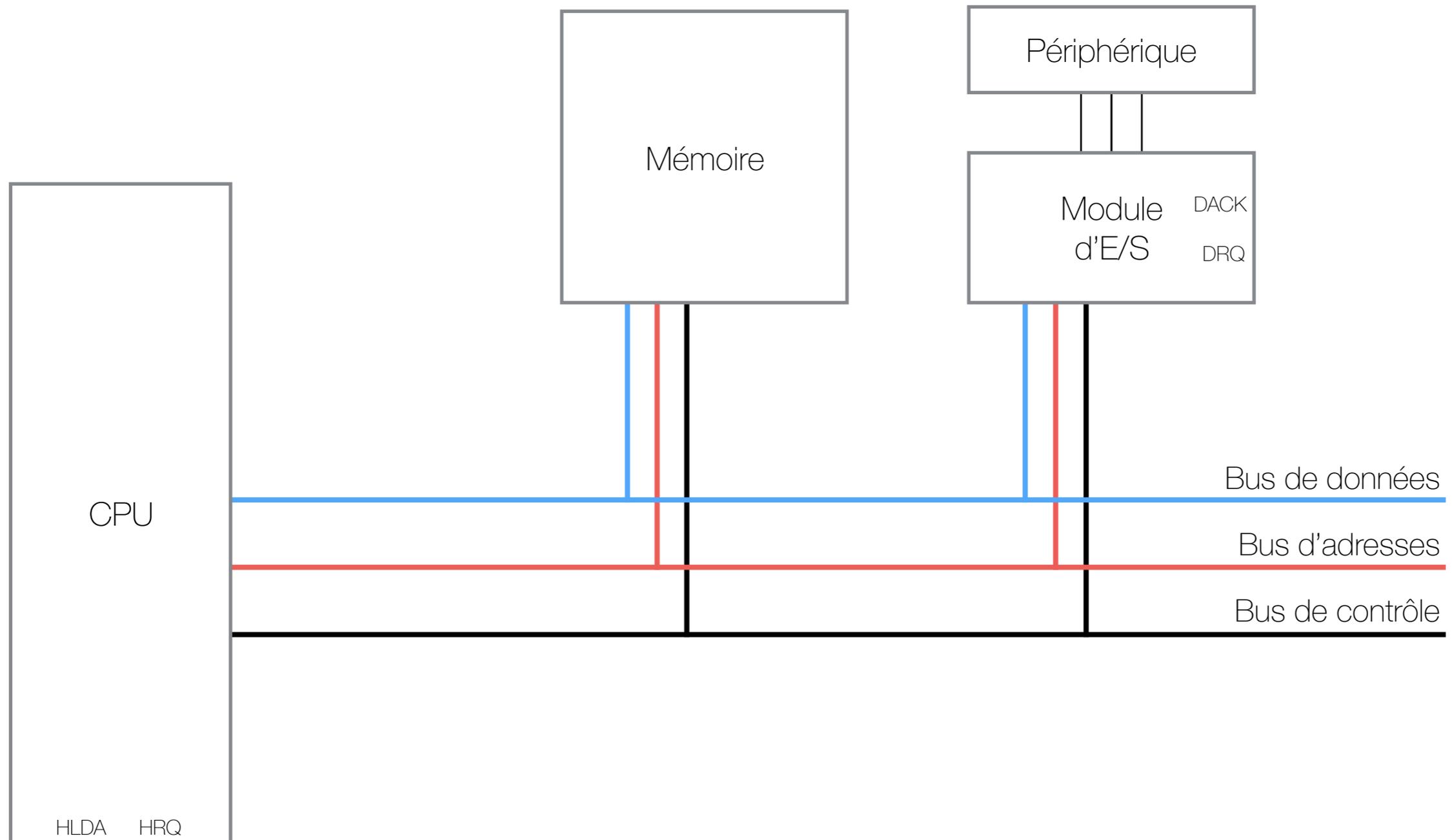
# Interruptions multiples, types et priorités

- Lorsque plusieurs périphériques peuvent faire des interruptions, il faut identifier qui fait l'interruption et déterminer quelle interruption est la plus prioritaire.
- Il existe quatre méthodes de détection de la source d'interruption:
  - Plusieurs lignes d'interruption. Le microprocesseur détermine qui a fait l'interruption en fonction de la ligne activée.
    - Requier trop de ligne pour les interruptions!
  - Interrogation logicielle (software polling). Lors d'interruption, un programme est exécuté afin de demander à chaque périphérique s'il a généré l'interruption.
    - Prend beaucoup de temps.
  - Chaînage en série (interruption matérielle vectorisée). Lors d'interruption, le périphérique ayant fait l'interruption mettra un numéro (ou vecteur) sur le bus de donnée après l'accusé réception du microprocesseur.
    - Cette technique est employée par la plupart des microprocesseurs.
  - Arbitrage de bus. Comme le chaînage en série, sauf que le périphérique prend le contrôle du bus de données avant de générer l'interruption.

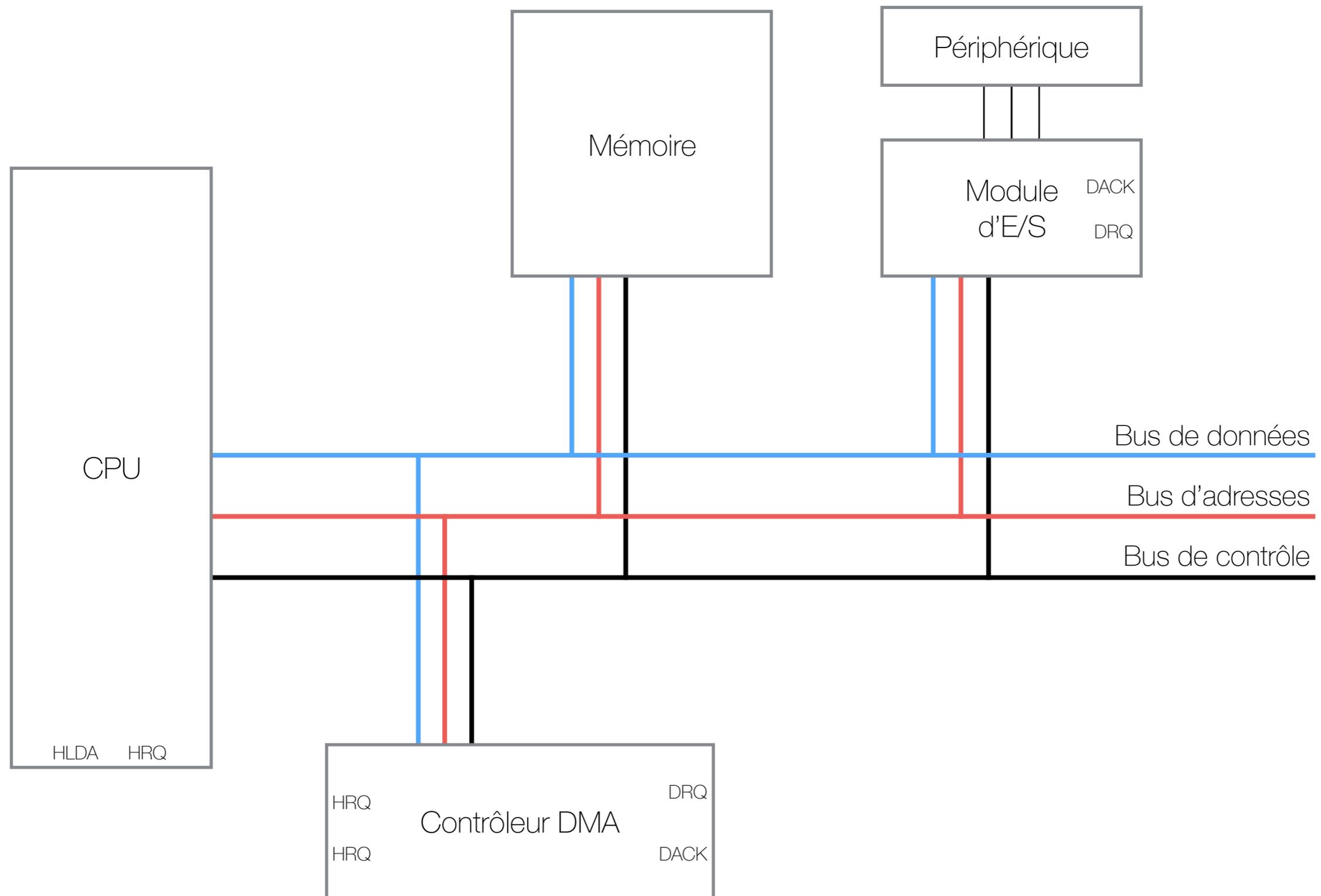
# Interruptions

- Avantages?
  - plus besoin d'attendre après le périphérique, c'est lui qui dit au microprocesseur quand il est prêt
- Désavantages?
  - la vitesse de transfert est limitée par la vitesse à laquelle le microprocesseur peut traiter ces instructions
    - rappelons-nous: le microprocesseur peut être occupé à autre chose lorsqu'il reçoit une interruption!
  - le microprocesseur doit gérer le transfert de données lui-même: il ne peut faire autre chose durant ce temps.
  - plusieurs instructions doivent être exécutées pour chaque transfert de données

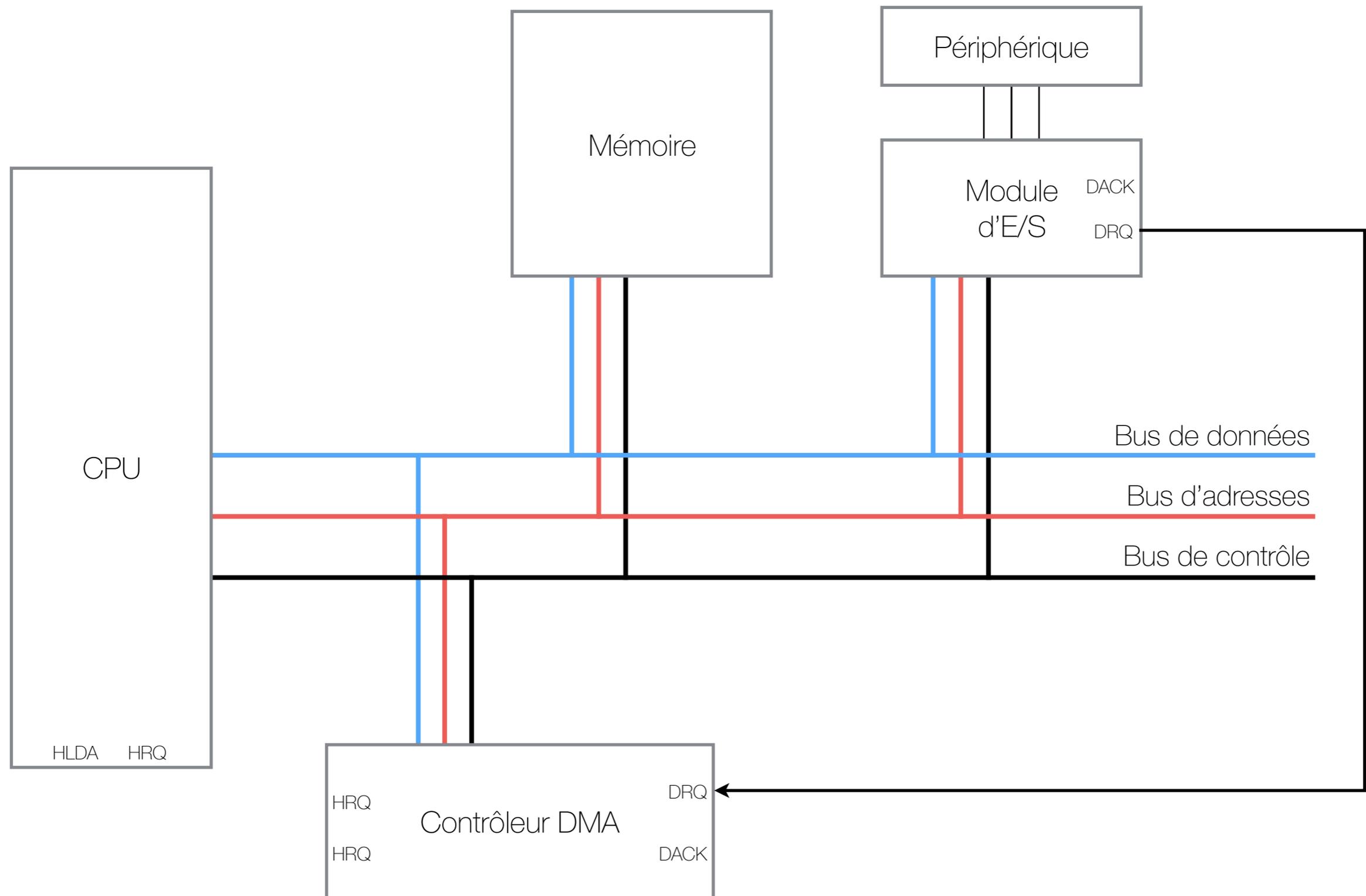
# Direct Memory Access (DMA)



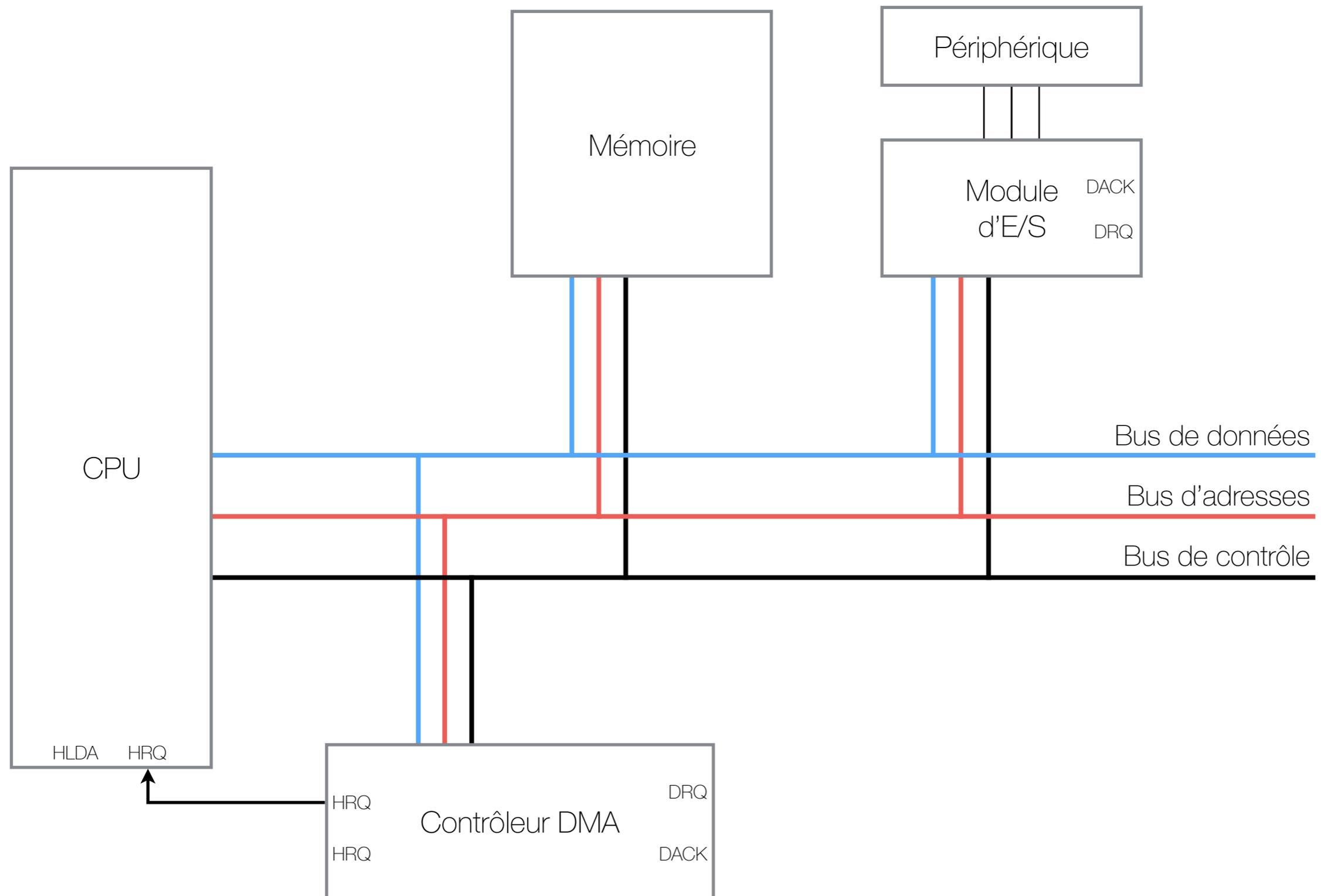
# Direct Memory Access (DMA)



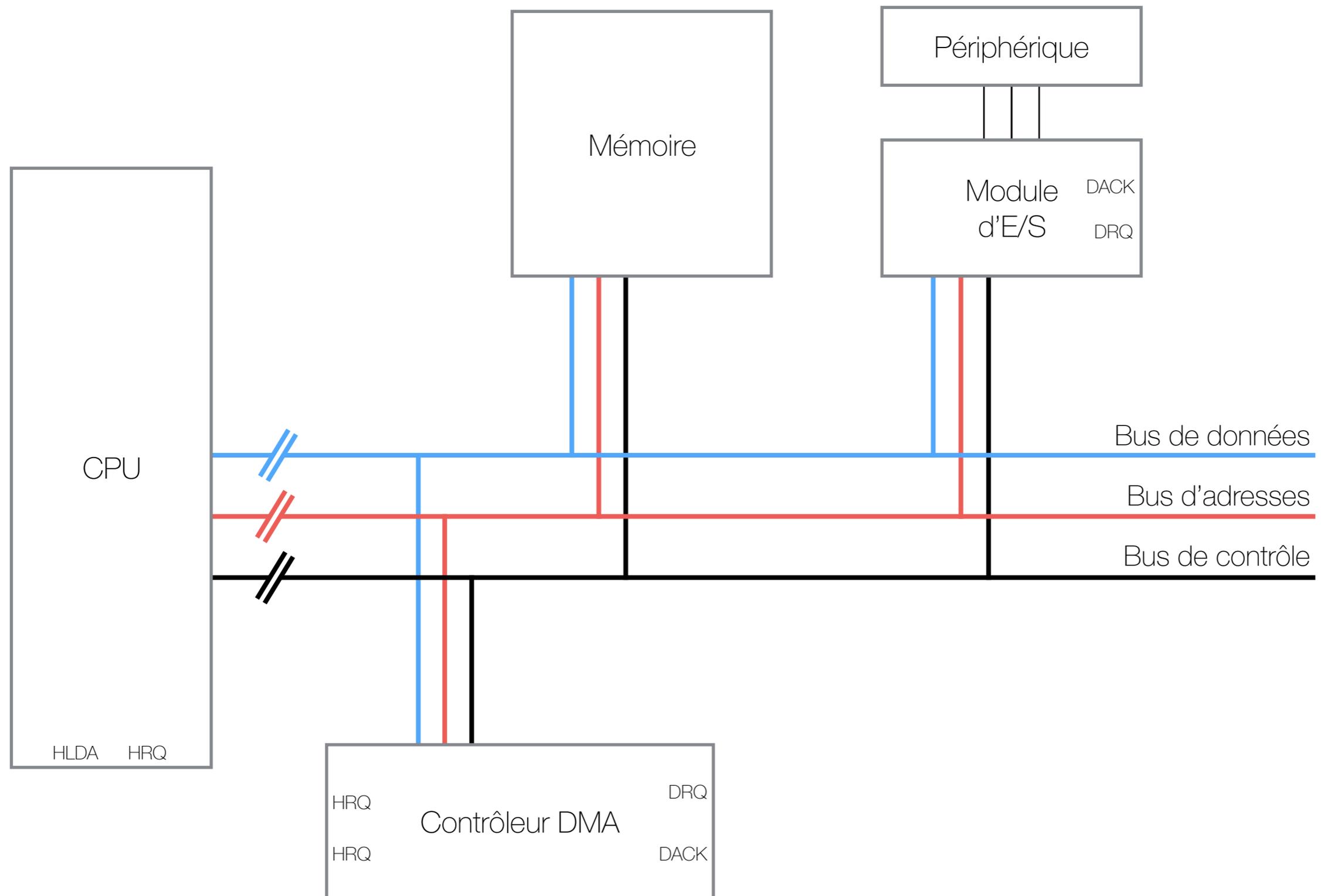
# Direct Memory Access (DMA)



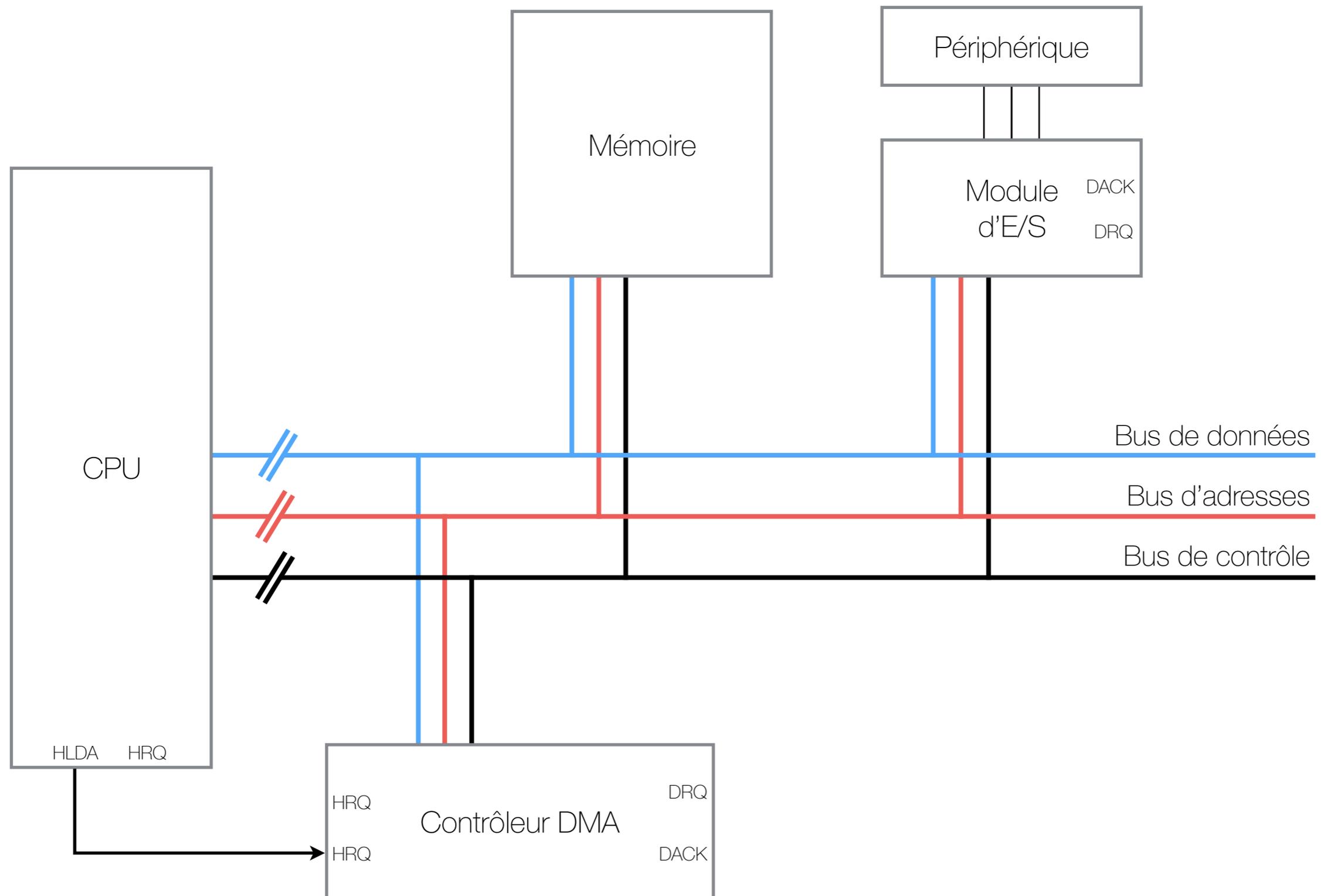
# Direct Memory Access (DMA)



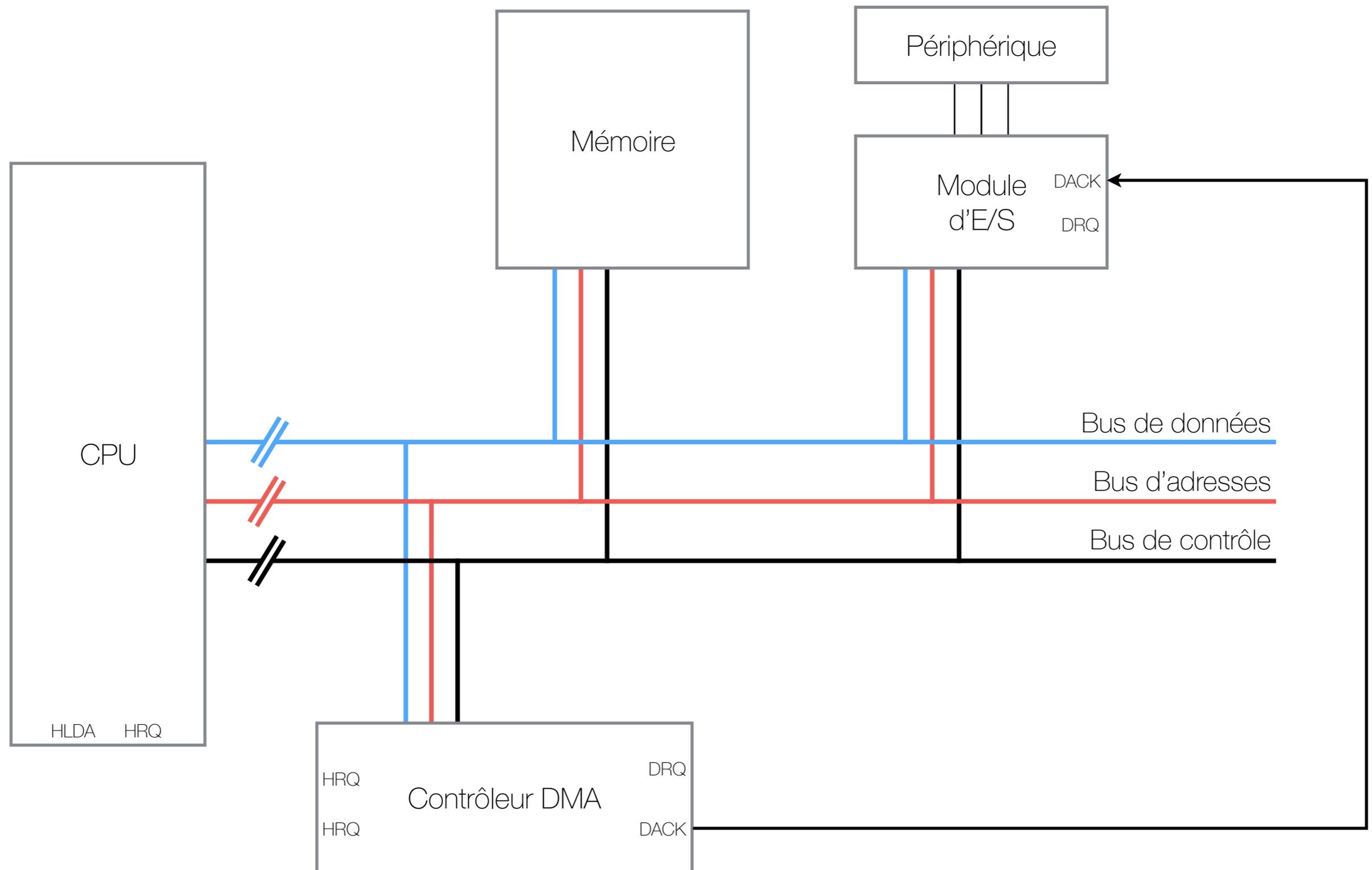
# Direct Memory Access (DMA)



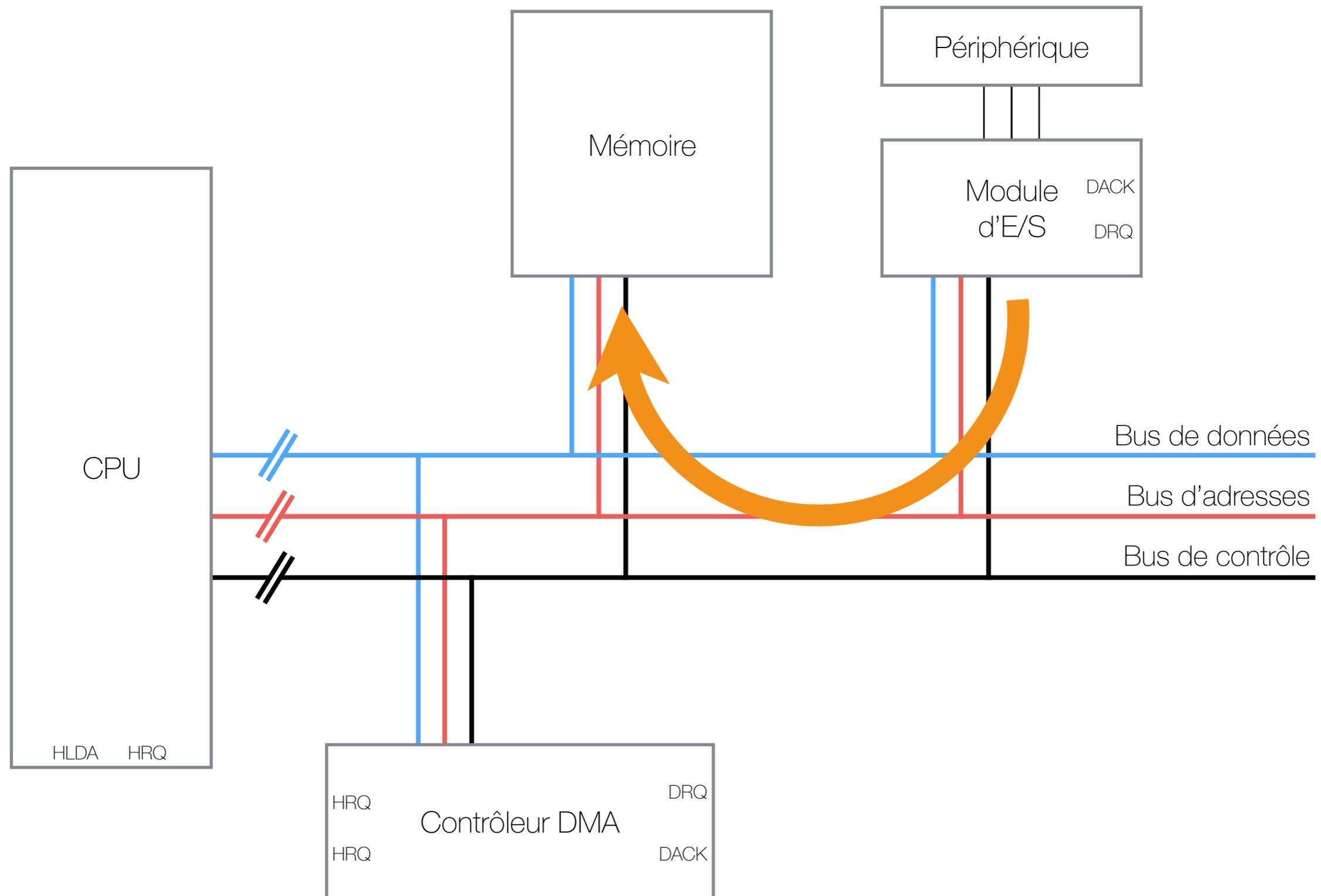
# Direct Memory Access (DMA)



# Direct Memory Access (DMA)

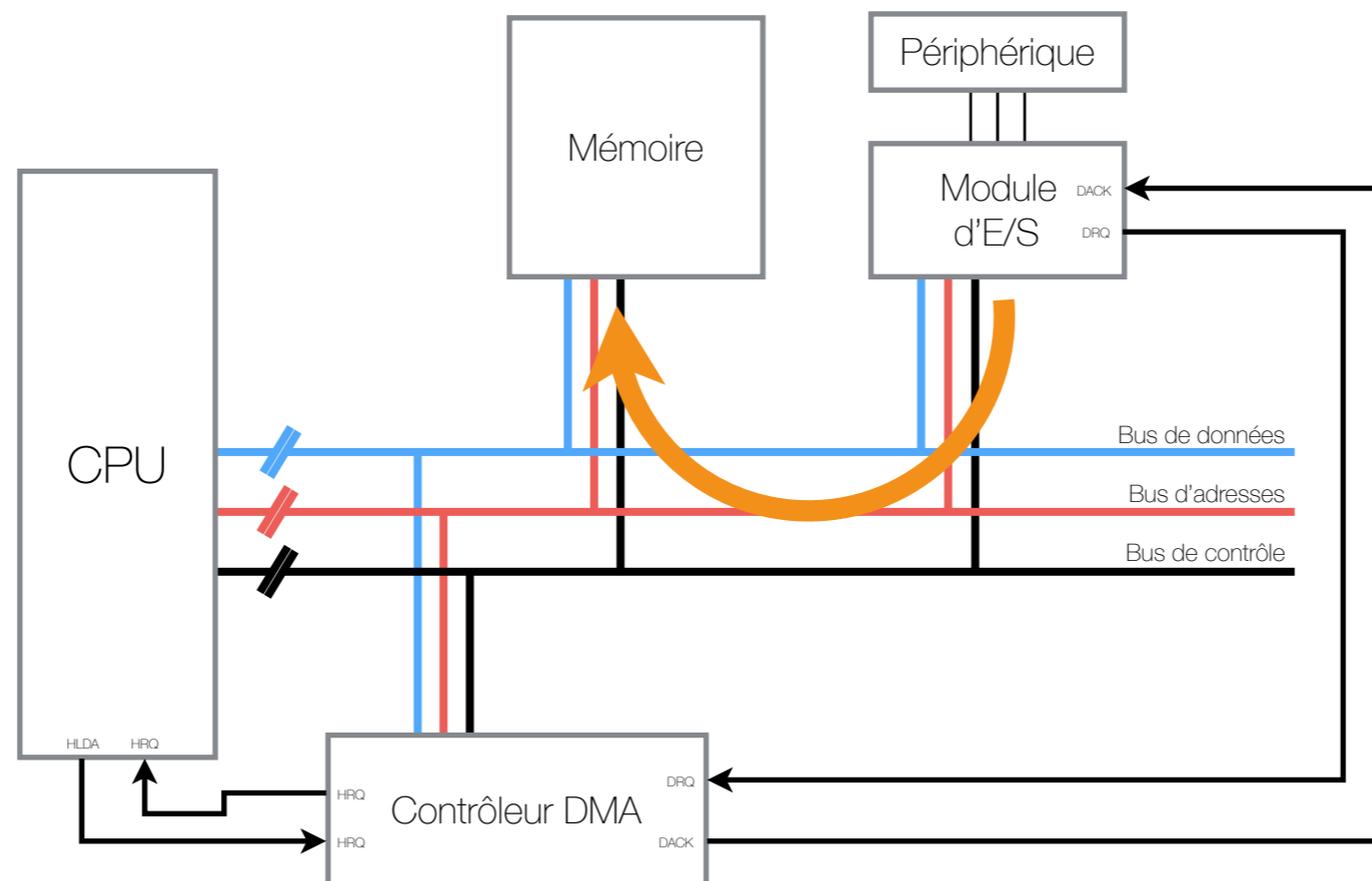


# Direct Memory Access (DMA)



# 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)

- 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. Dans l'exemple de la page précédente (qui est un vieil exemple), un périphérique demande un transfert par DMA en activant DRQ2. Ensuite, le contrôleur de DMA (8237) active la ligne HRQ pour demander un accès au bus de données au microprocesseur. Enfin, le microprocesseur active HLDA pour permettre le transfert.

# 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)

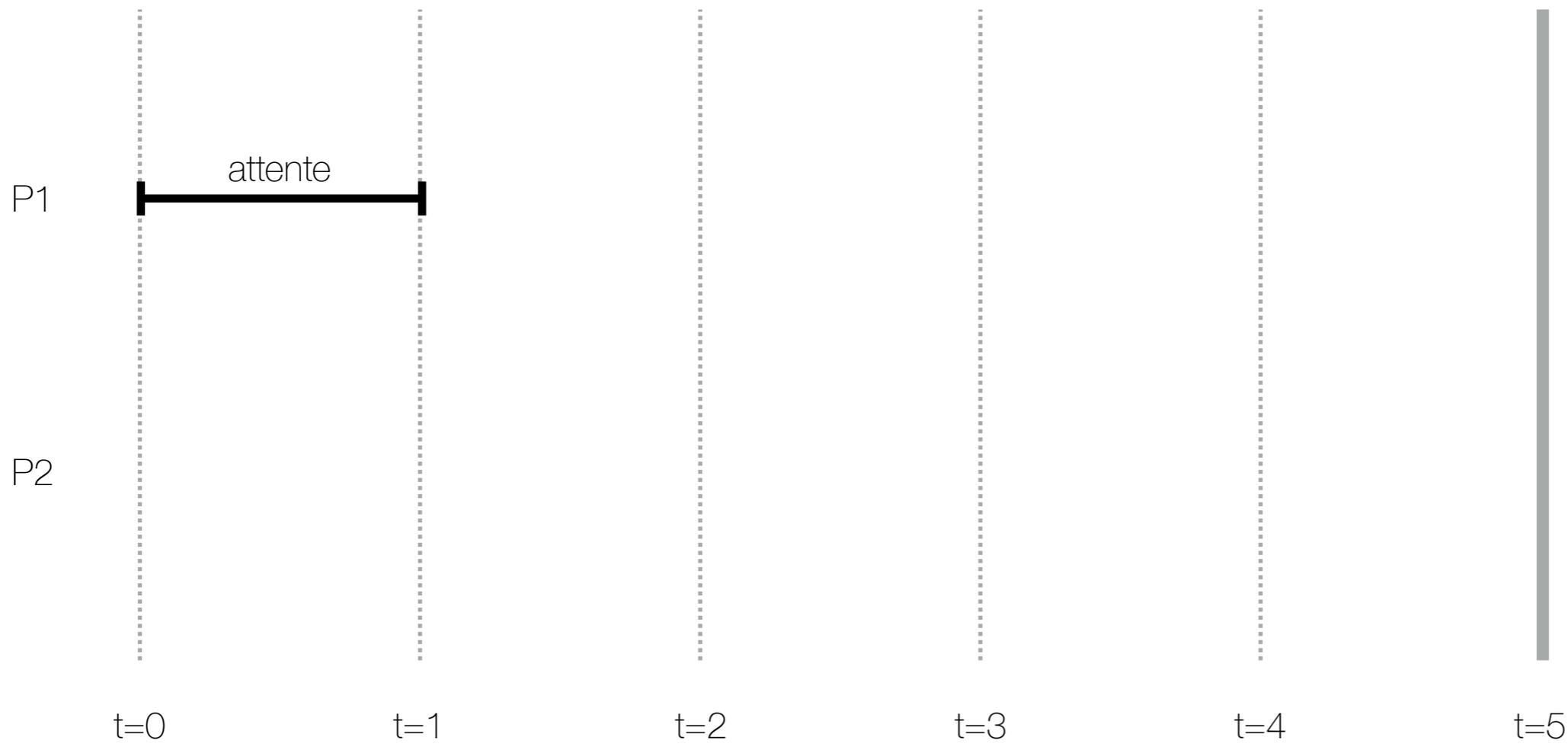
- Dans les systèmes modernes, le microprocesseur n'est pas arrêté pendant le transfert par DMA.
- Cependant, le microprocesseur et le contrôleur de DMA pourraient avoir besoin des bus en même temps. Que faire?
  - Deux bus d'accès à la mémoire peuvent permettre au contrôleur de DMA et au microprocesseur 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 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.

# 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 l'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 est plus prioritaire que l'interruption de l'horloge d'ordonnancement;
- 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?

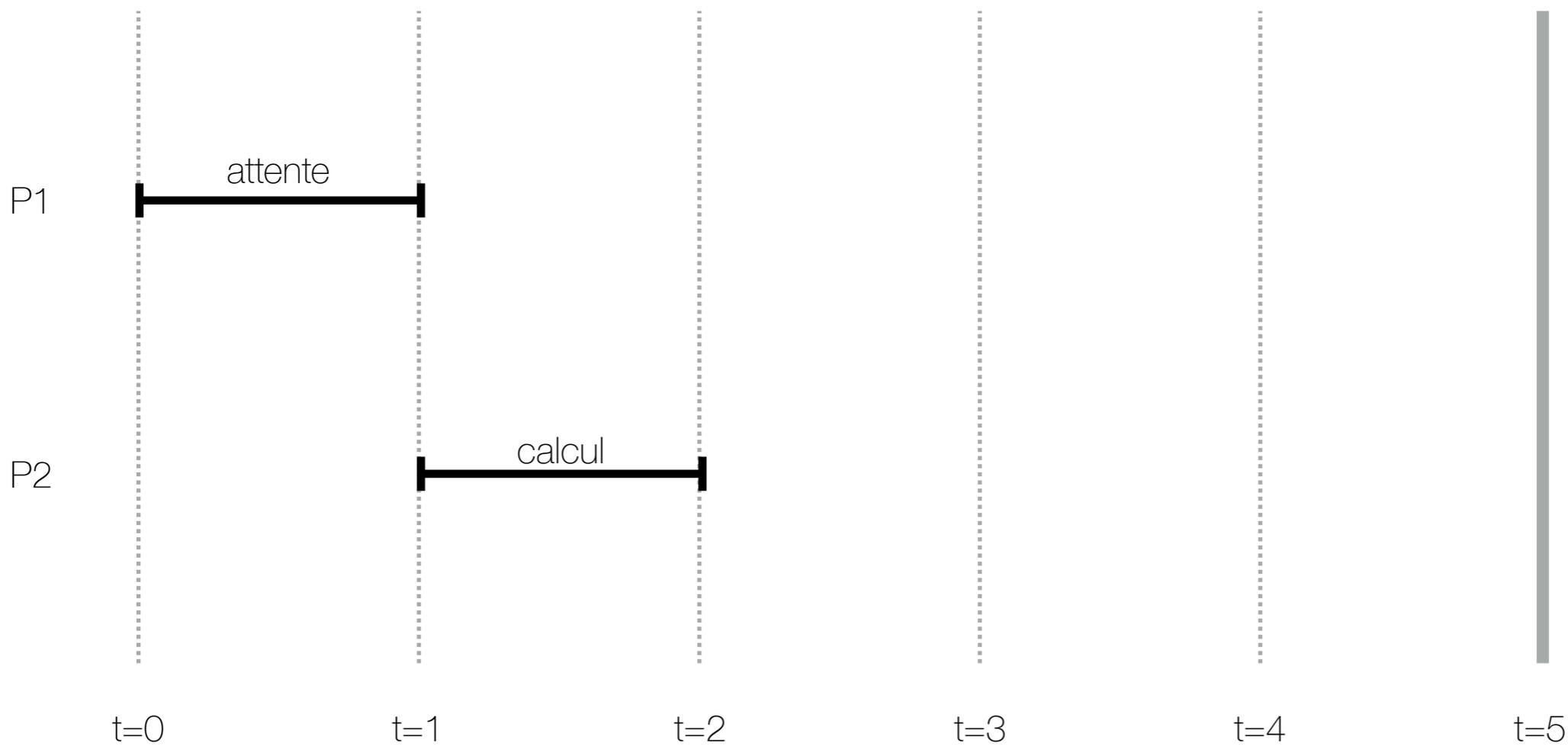
# Exemple: E/S programmées

P1 fait la requête  
au DD

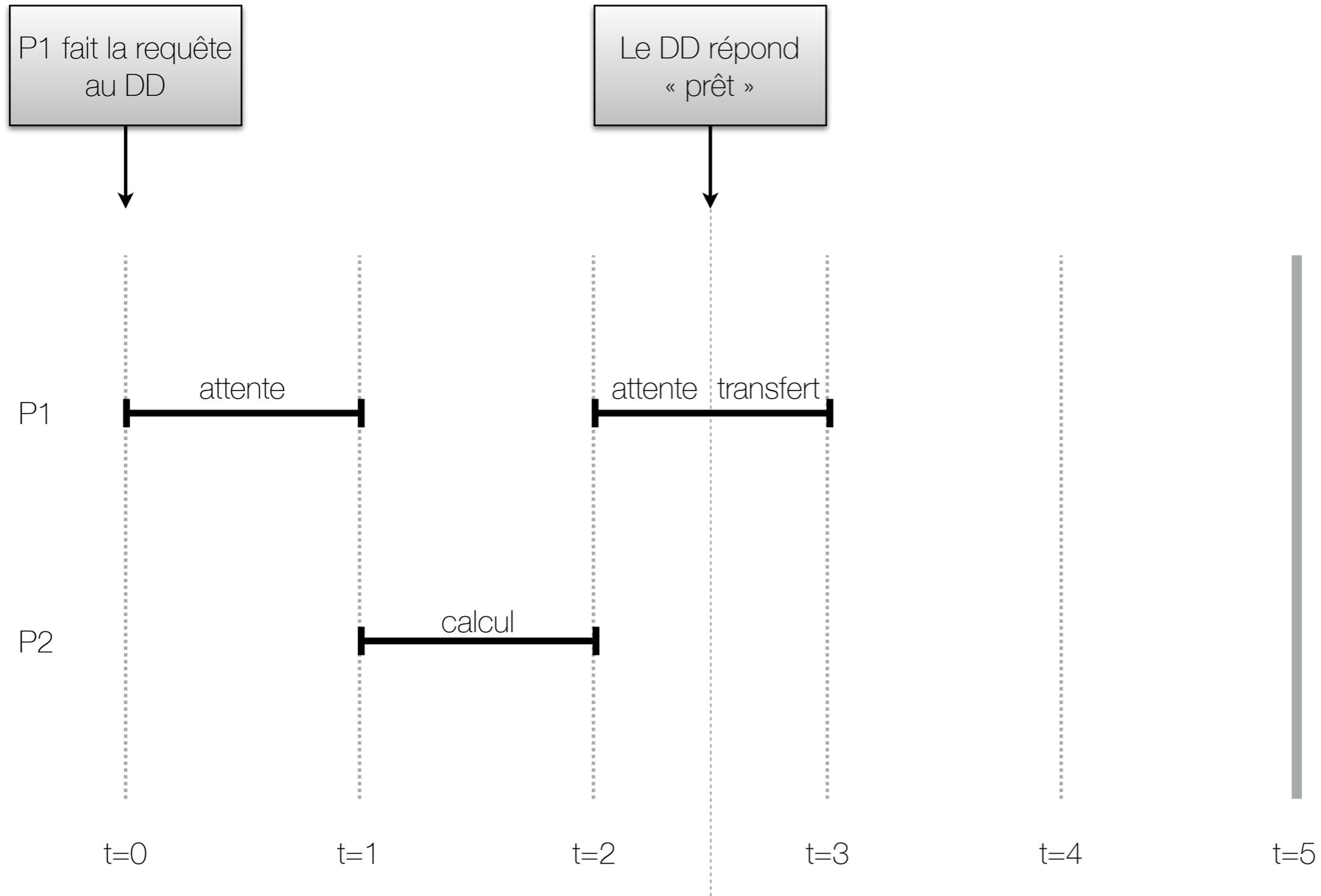


# Exemple: E/S programmées

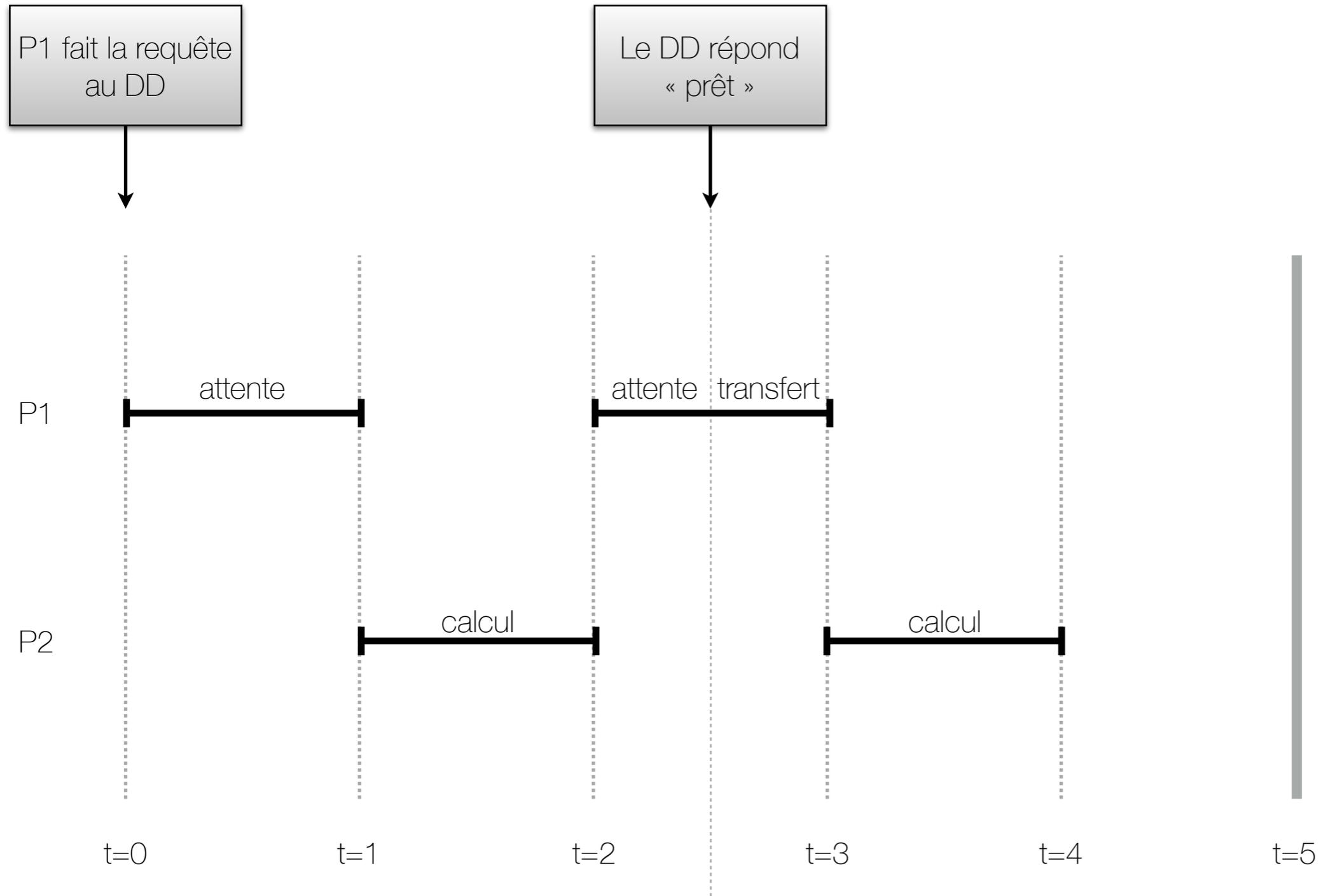
P1 fait la requête  
au DD



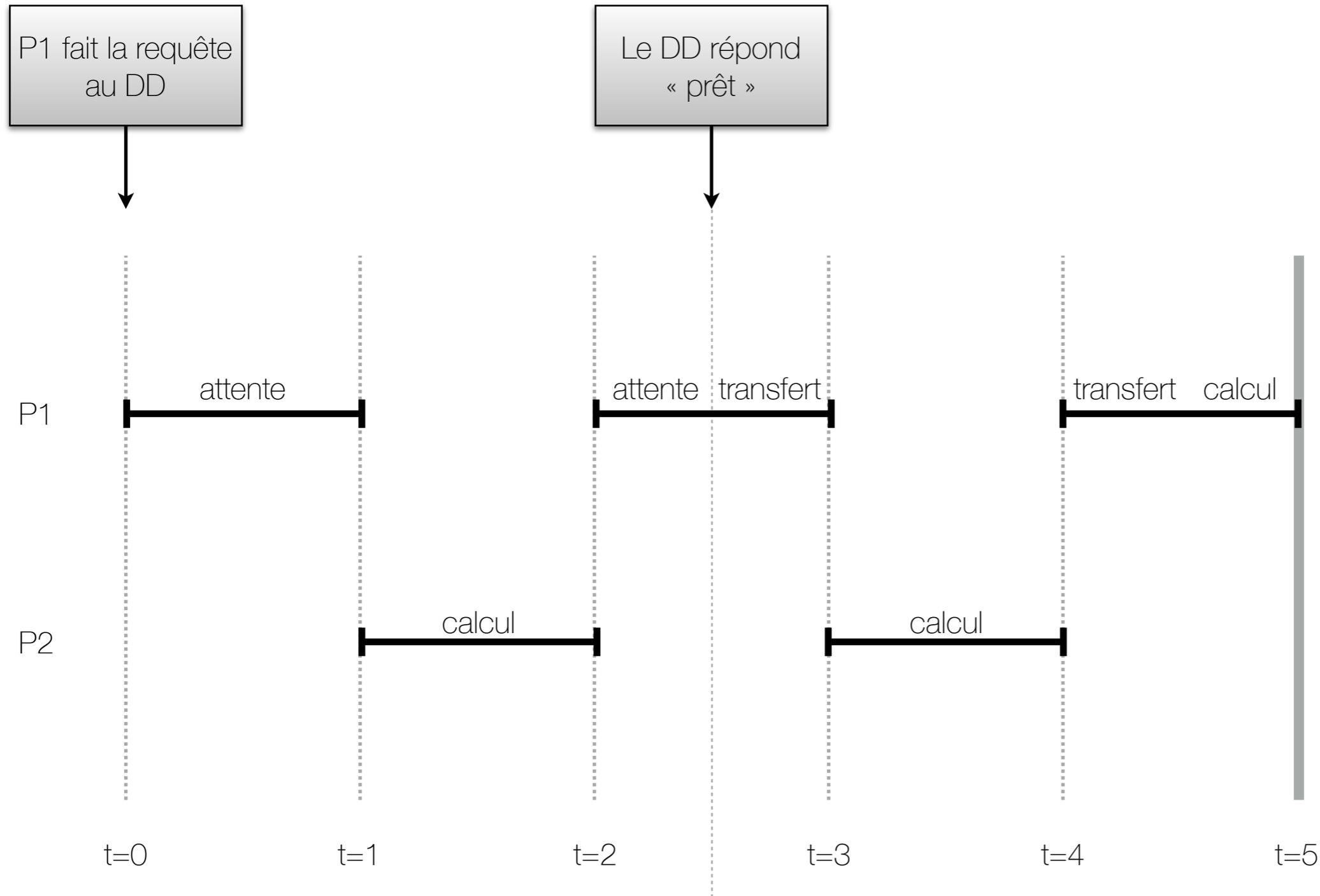
# Exemple: E/S programmées



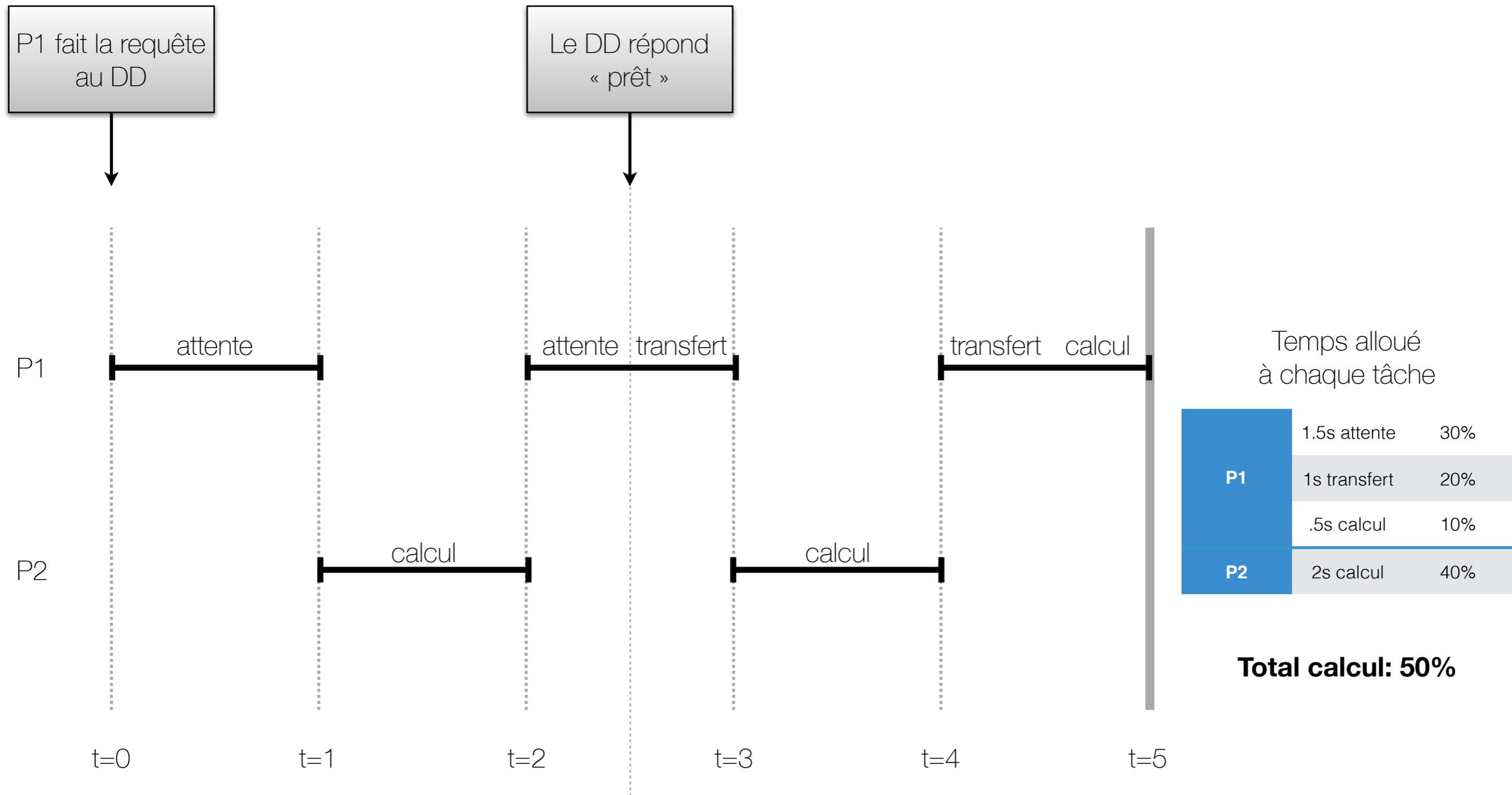
# Exemple: E/S programmées



# Exemple: E/S programmées

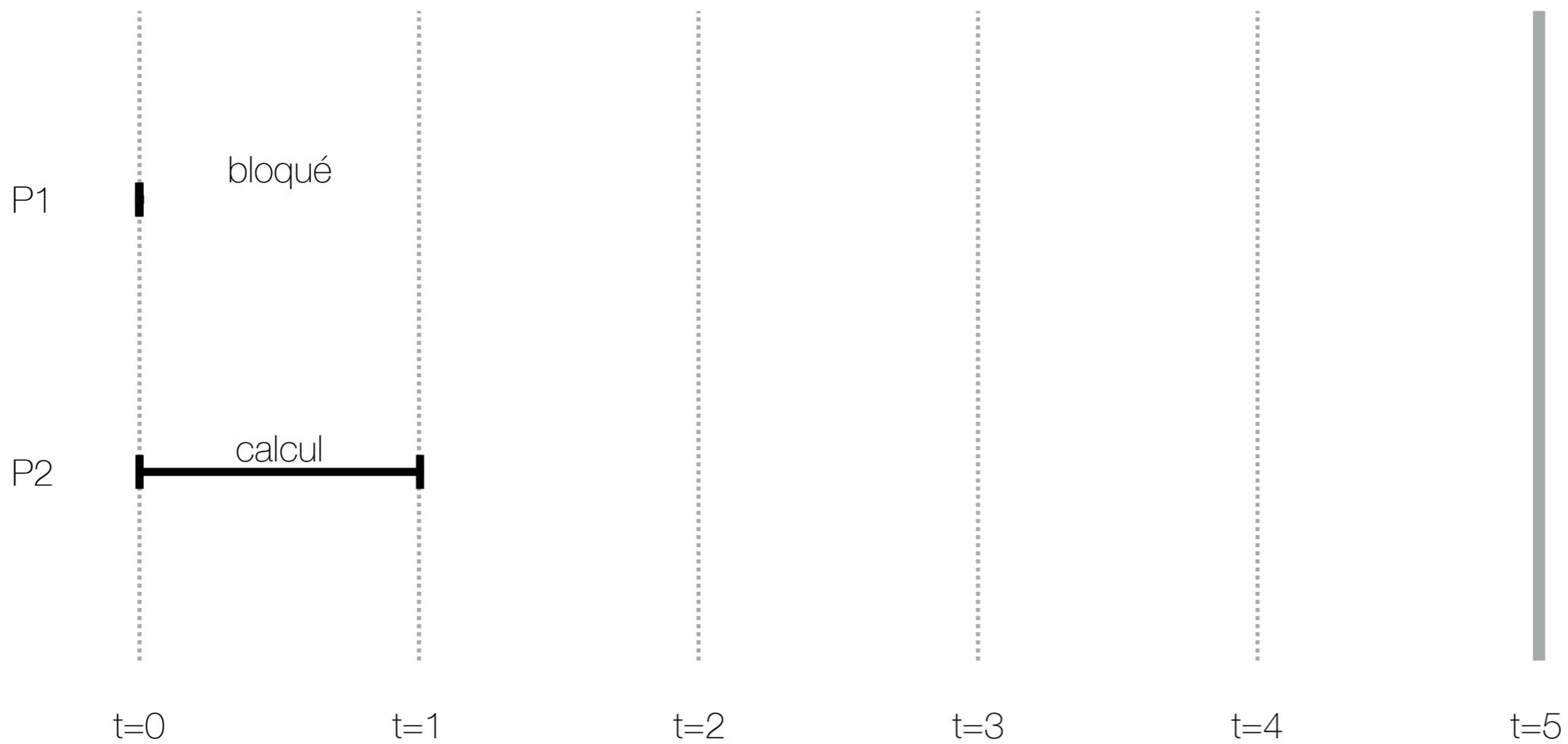


# Exemple: E/S programmées



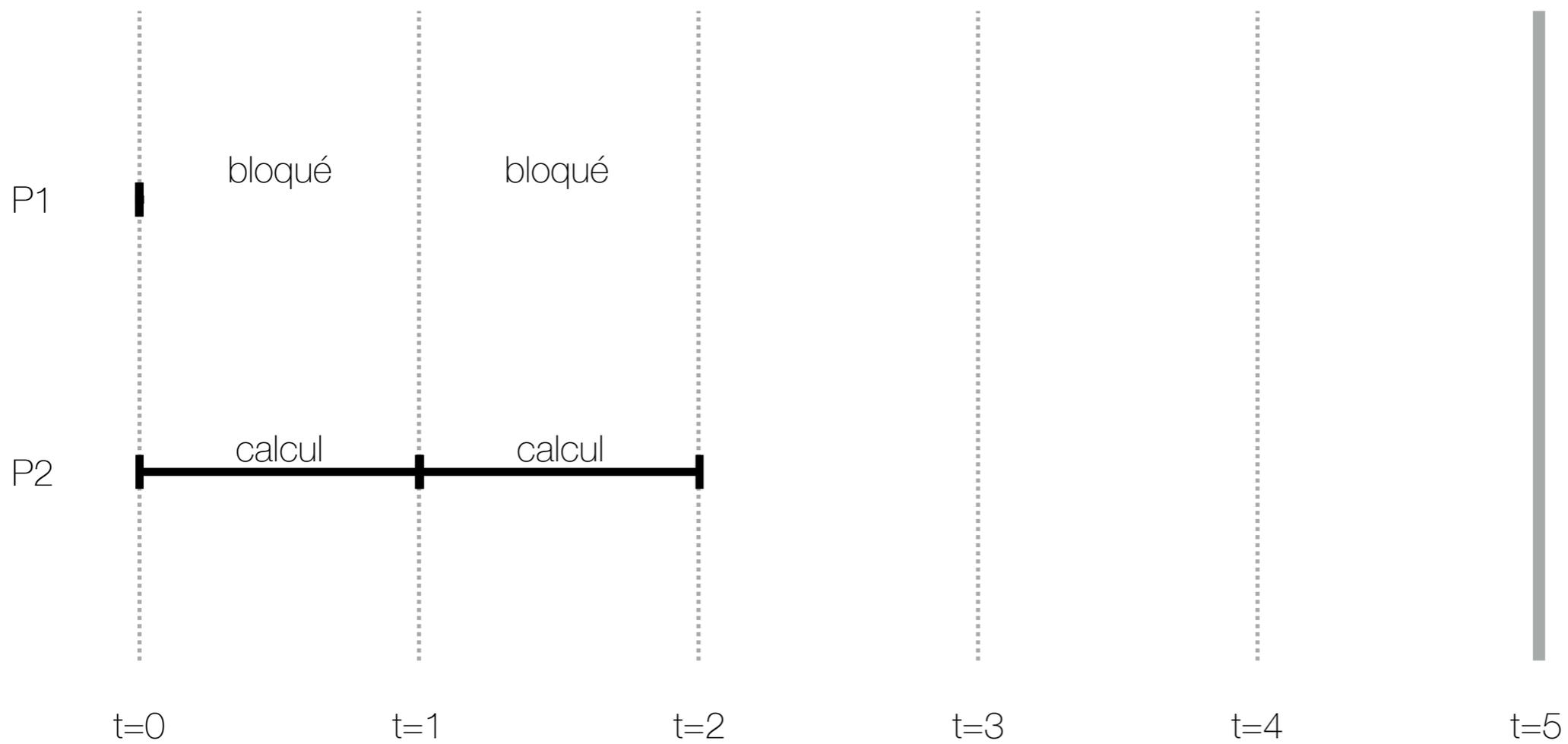
# Exemple: E/S par interruptions

P1 fait la requête  
au DD

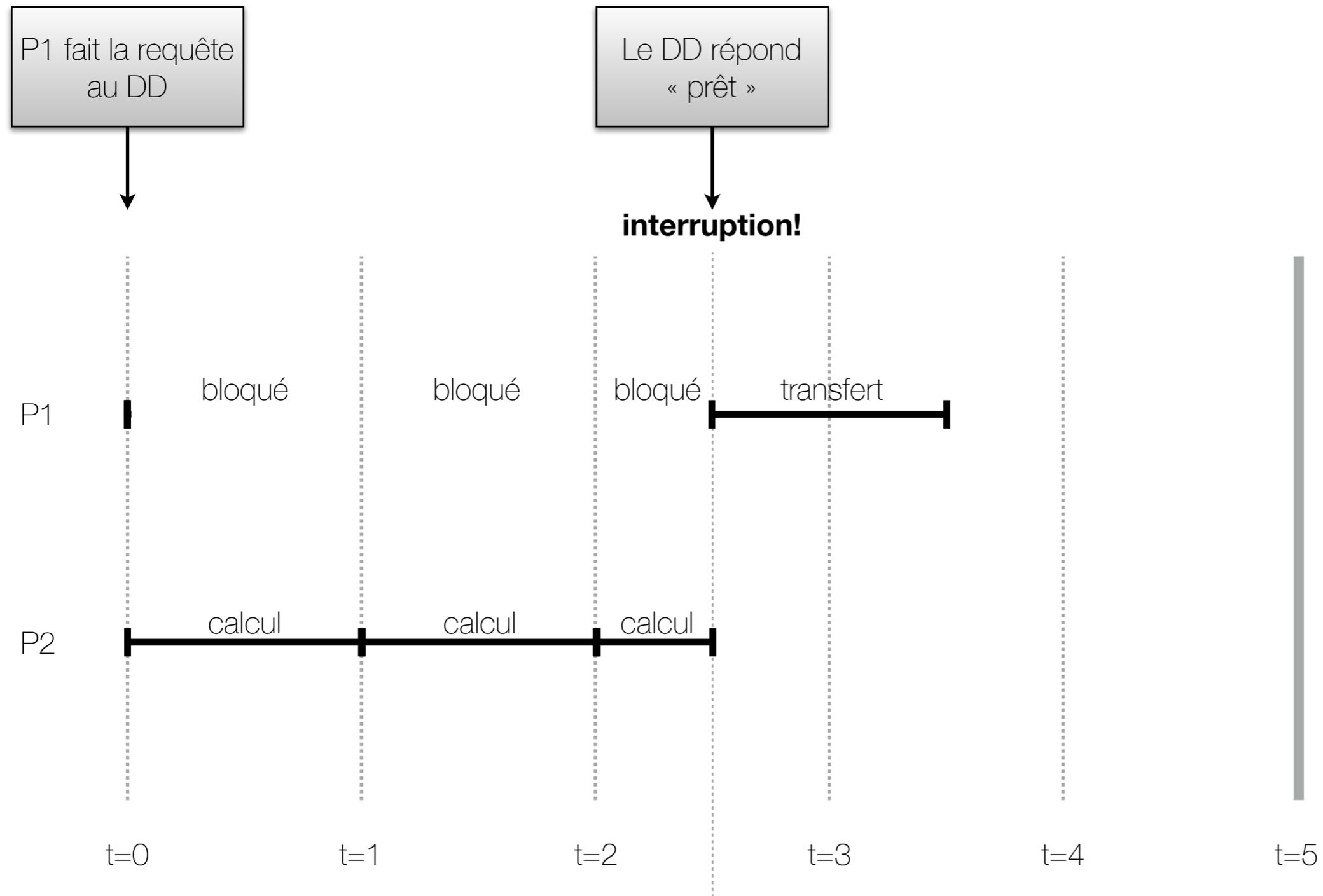


# Exemple: E/S par interruptions

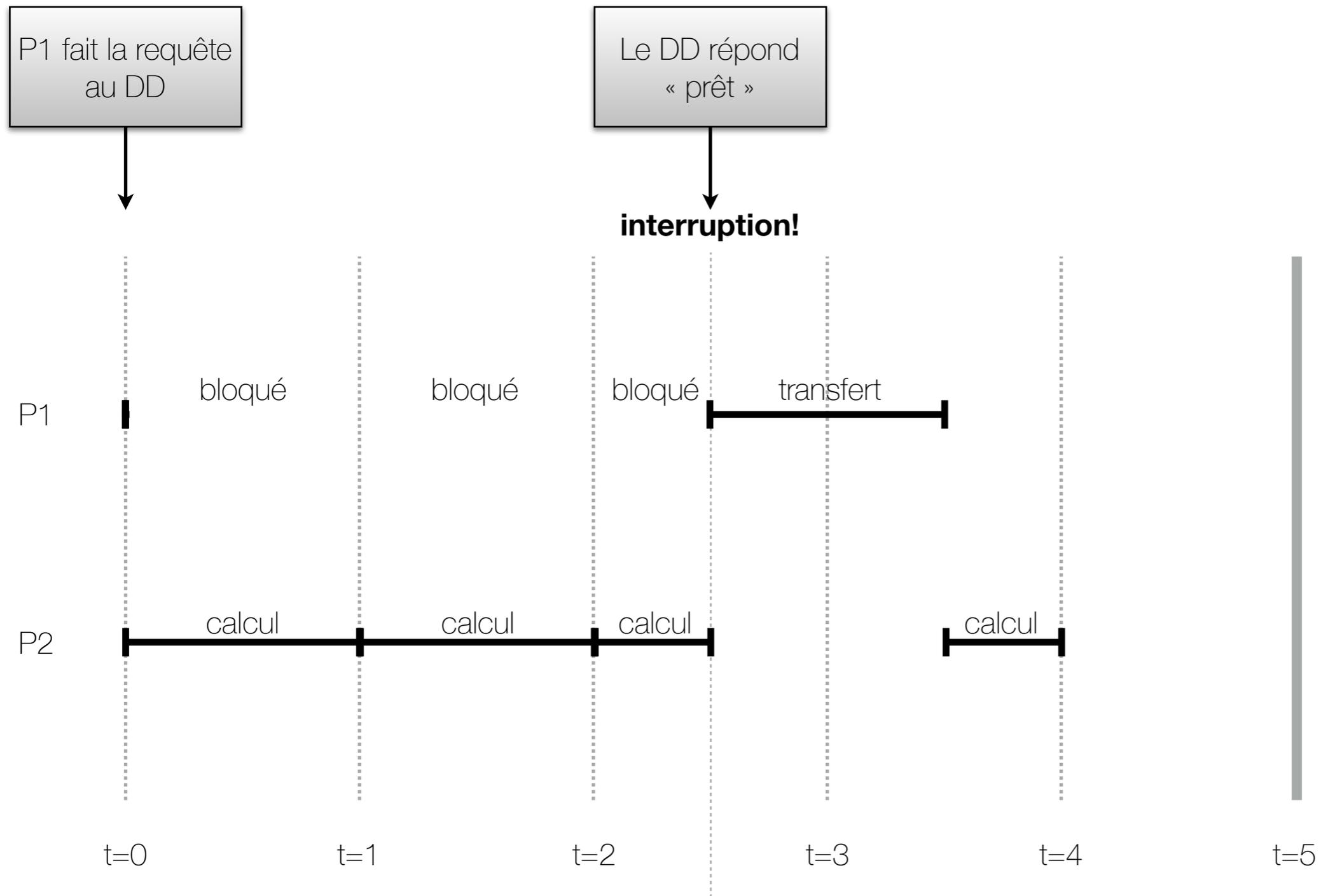
P1 fait la requête  
au DD



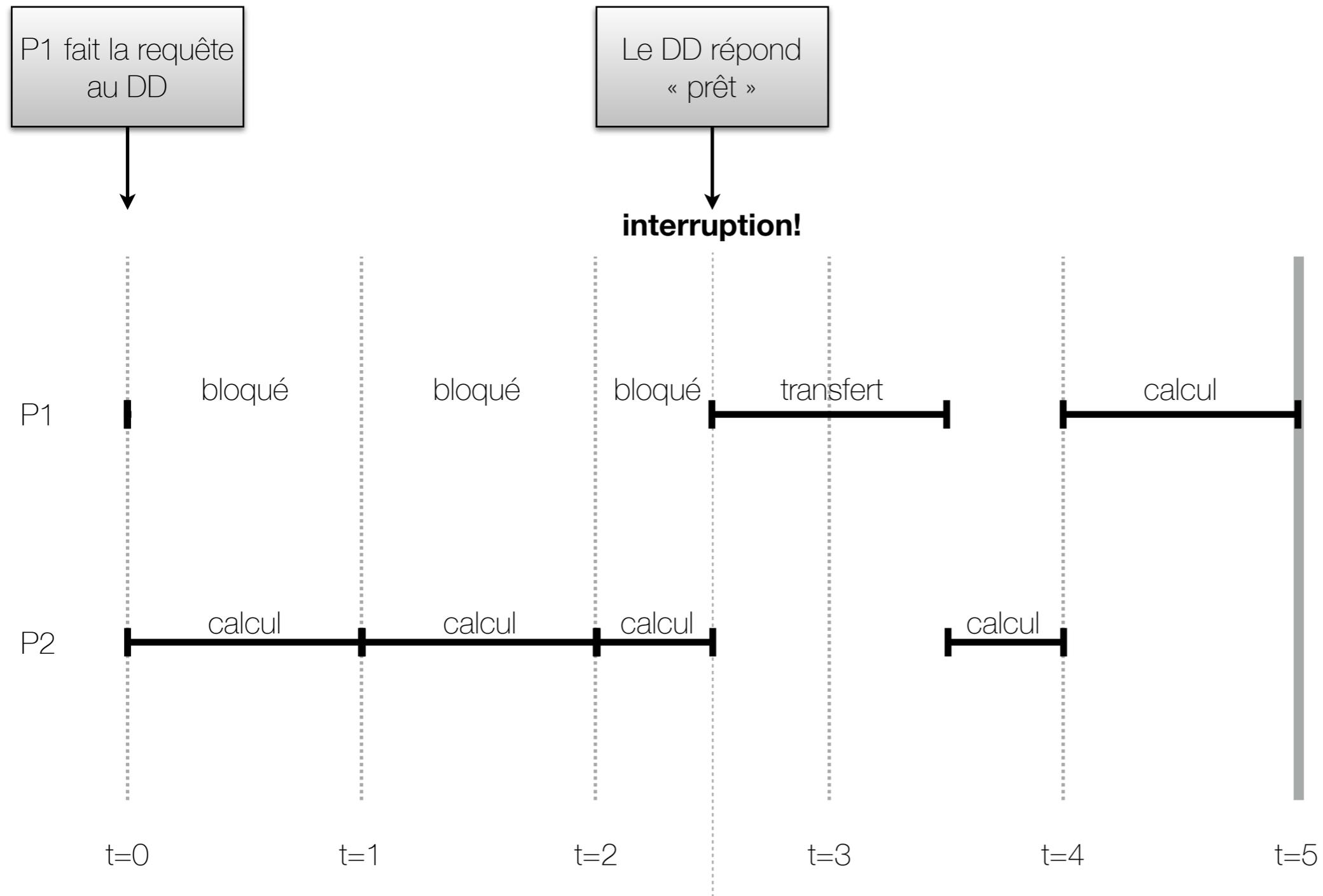
# Exemple: E/S par interruptions



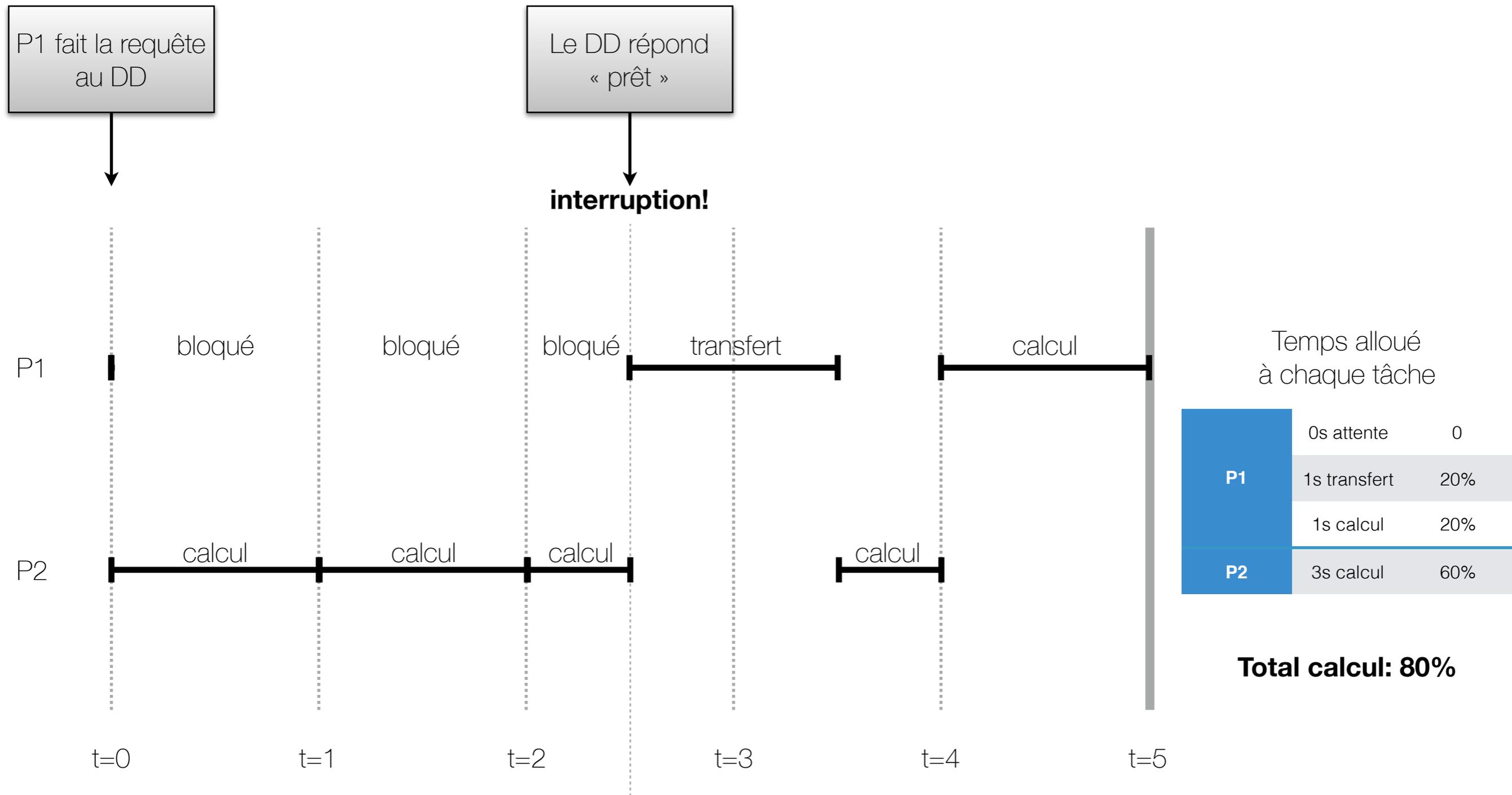
# Exemple: E/S par interruptions



# Exemple: E/S par interruptions

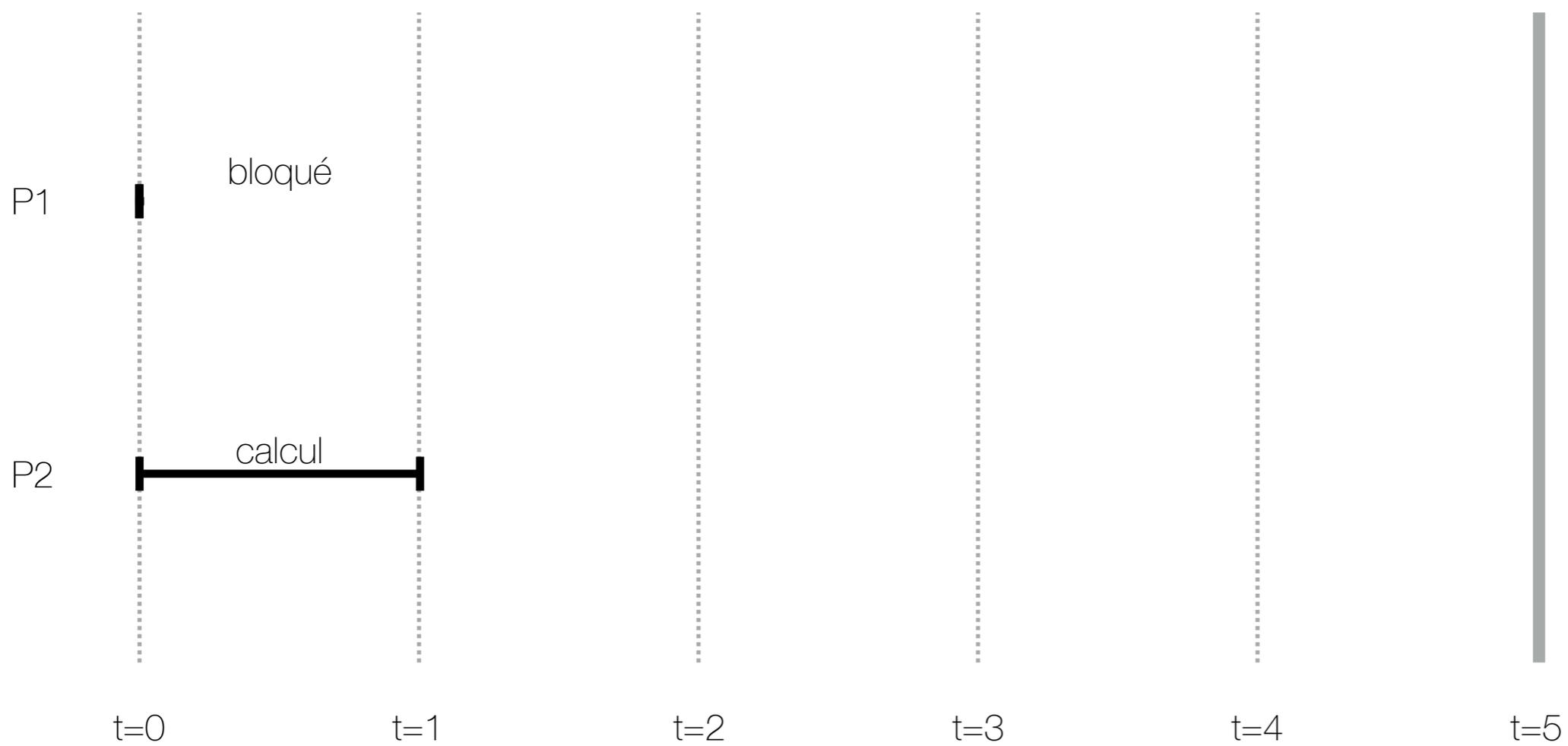


# Exemple: E/S par interruptions



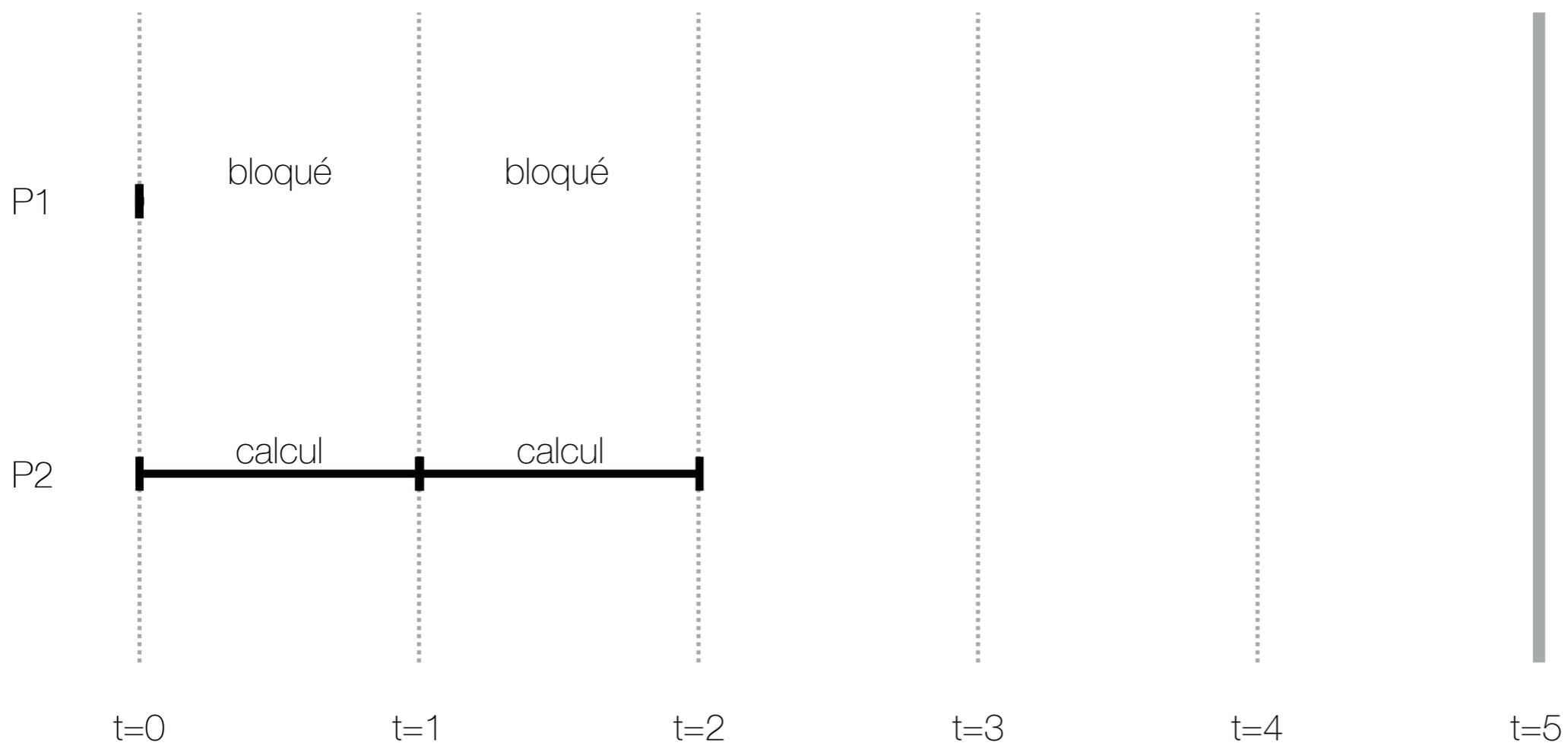
# Exemple: E/S par DMA

P1 fait la requête au **contrôleur de DMA**

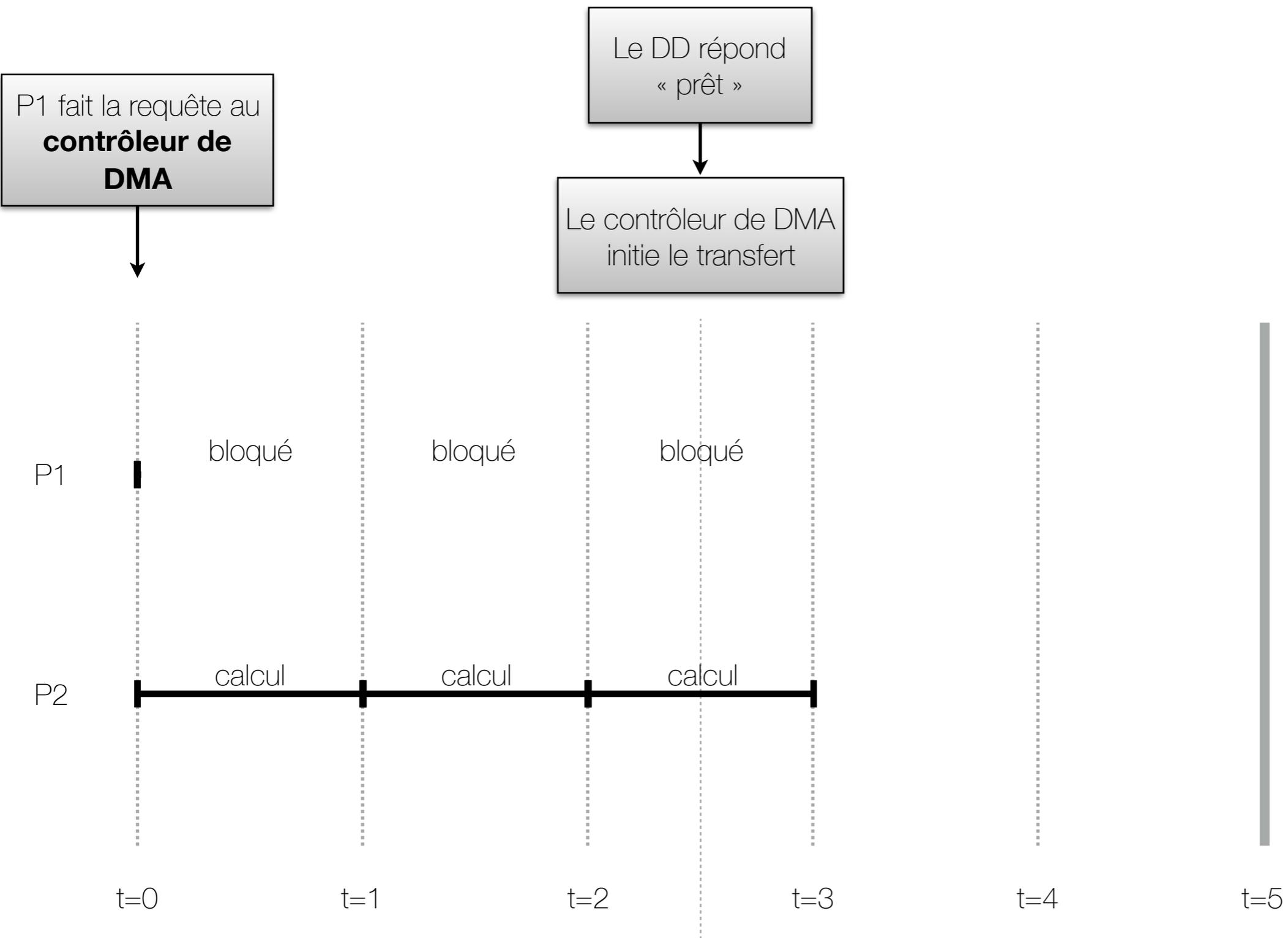


# Exemple: E/S par DMA

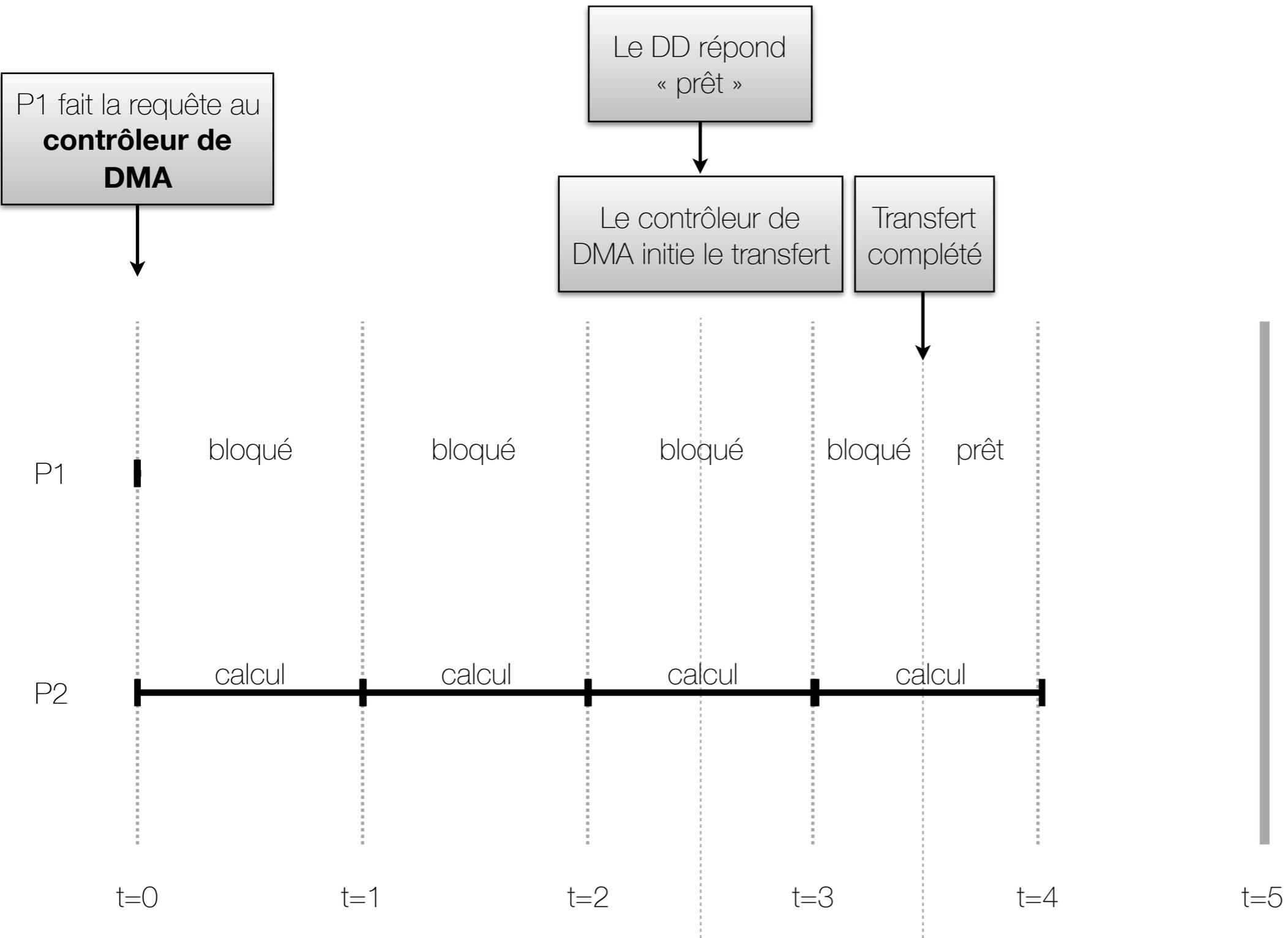
P1 fait la requête au **contrôleur de DMA**



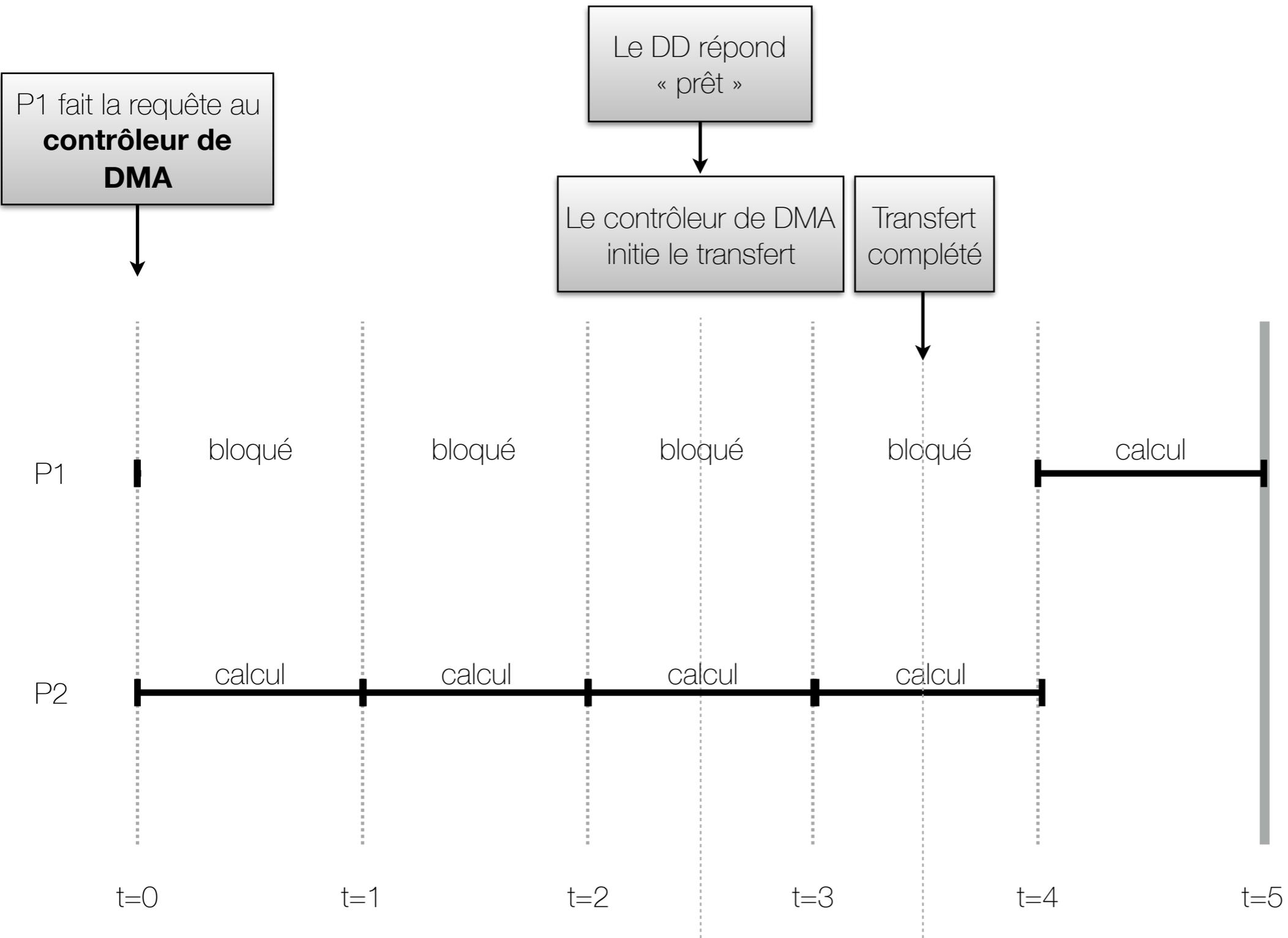
# Exemple: E/S par DMA



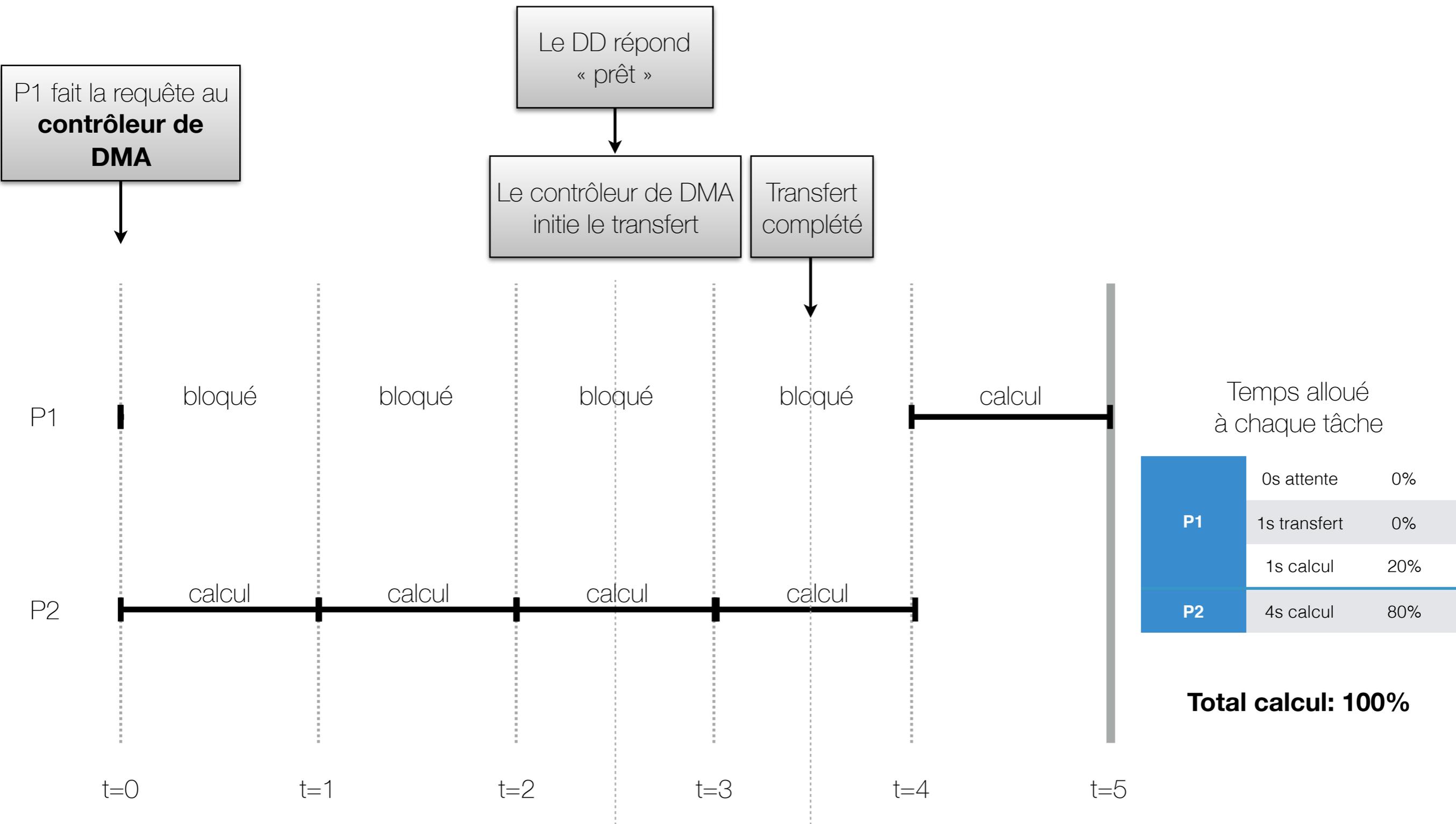
# Exemple: E/S par DMA



# Exemple: E/S par DMA



# Exemple: E/S par DMA



# À retenir

- Contrôleur de périphérique: pourquoi?
- entrées-sorties:
  - programmées
  - par interruptions
  - par DMA

# Références et exercices

- Références
  - Iv Englander: Chapitres 9, 11.1 et 11.2