

# Côté matériel

Au niveau matériel, on distingue plusieurs éléments :

- Le **périphérique d'entrée-sortie**, par exemple clavier, souris, écran, etc. On pourra le caractériser par sa nature ainsi que sa vitesse de transmission. Ces périphériques vont faire de la **signalisation** (ou **interruption**) pour signaler au système que sa tâche est terminée.
- Le **contrôleur** qui est l'interface entre le périphérique et le système d'exploitation. Le contrôleur permet au SE de contrôler le périphérique.

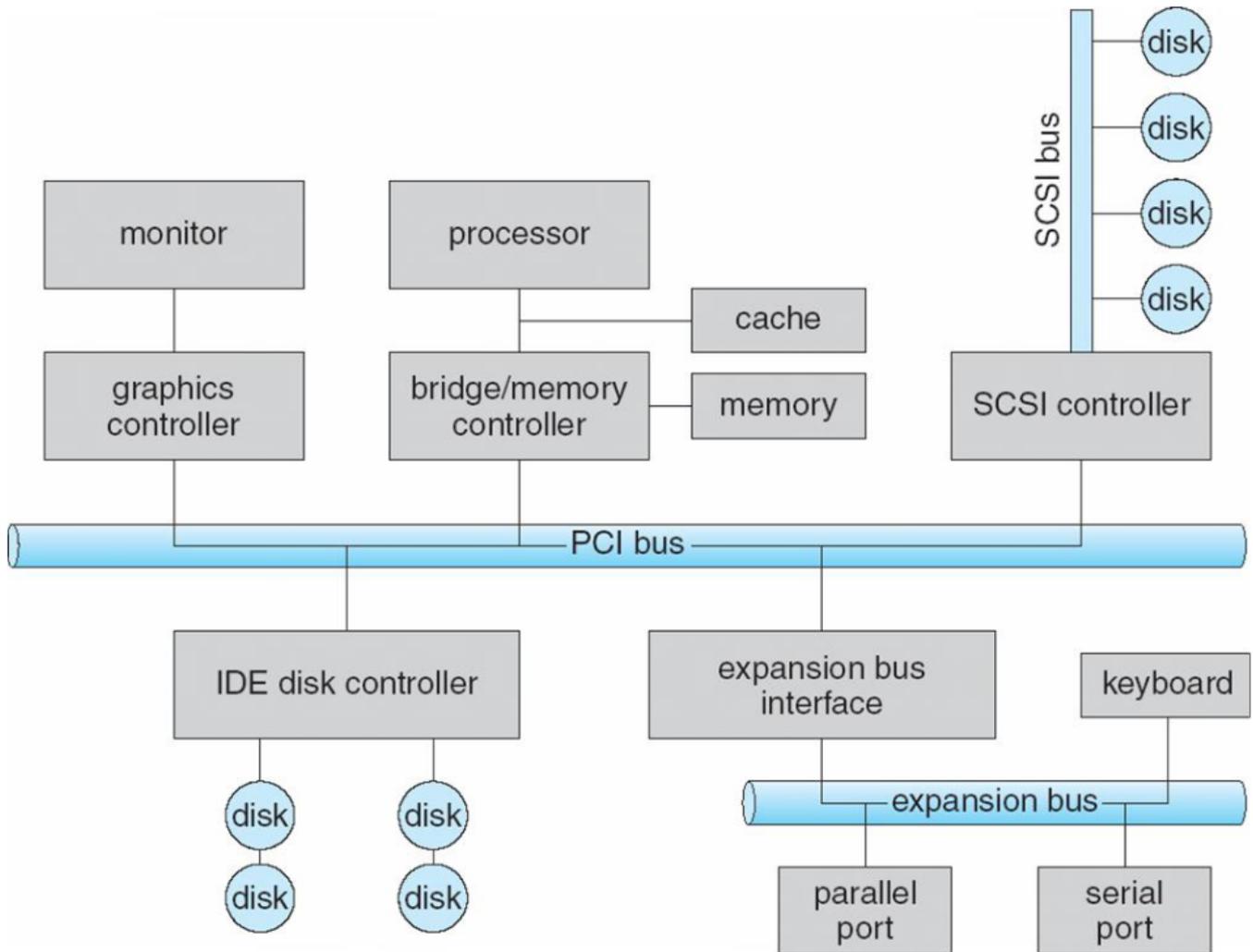
## Périphériques E/S

Il existe plusieurs types de périphériques d'E/S, ici, on va parler de deux types principaux.

Il y a les périphériques **blocs** où l'information est stockée dans des blocs de taille fixes et adressables. Un exemple de tels périphériques est un disque dur.

Il y a aussi les périphériques **caractères** où l'information est un flux de caractère sans tenir compte d'une structure quelconque, elle est donc non adressable. Par exemple, la carte réseau, clavier, souris, etc.

## Controlleur de périphérique



Le contrôleur est la partie électronique qui contrôle le matériel. Souvent, le périphérique dispose d'une partie mécanique et d'une partie électronique (exemple, un disque dur). Cependant, ce n'est pas toujours le cas, par exemple la carte graphique n'est pas attachée à l'écran.

Certaines cartes d'extensions sont même dissociées de la carte mère, c'est par exemple le cas avec les périphériques USB.

## Contrôle/commandes

Le système d'exploitation **commande** le périphérique au moyen du contrôleur, il n'interagit jamais directement avec le périphérique lui-même.

Le contrôleur dispose fréquemment d'un microprocesseur hautement adapté au périphérique, de registres qui permettent de commander le contrôleur et de mémoire pour faire les échanges (par exemple pour faire des buffers).

## Interruptions

Le matériel envoie des interruptions pour signaler que l'opération en cours est terminée. Le gestionnaire d'interruption du système d'exploitation va gérer l'interruption.

S'il n'y a qu'une seule interruption, le système gère l'interruption et choisit ensuite le processus suivant à démarrer.

S'il y a plusieurs interruptions, le système traite d'abord les interruptions urgentes en ignorant temporairement les autres.

## Fonctionnement

Le CPU sauvegarde le contexte du processus en cours et lance une routine (une fonction indépendante du reste du système) adaptée à l'interruption reçue. La routine à lancer est renseignée dans le **vecteur des interruptions** où pour chaque interruption un pointeur vers la routine à exécuter est renseignée.

Dès que l'interruption est traitée, l'état du contrôleur du périphérique est modifié. Le gestionnaire des interruptions est alors prêt à traiter les interruptions suivantes.

Une fois toutes les interruptions traitées, le système d'exploitation choisit le processus suivant à lancer.

## Dialogue via port d'E/S

Ce type de dialogue entre le système et le contrôleur consiste à échanger des données en écrivant ou lisant des données dans les registres du contrôleur.

Cette méthode n'est plus trop utilisée aujourd'hui.

## Dialogue via E/S mappé en mémoire

Dans cette méthode, on va faire correspondre une zone mémoire (de la mémoire centrale) aux registres du contrôleur. Ainsi, lorsque l'on lit dans la zone mémoire, on lit et/ou écrit dans les registres du contrôleur.

L'avantage de cette méthode est que l'on a juste à utiliser des instructions habituelles d'accès à la mémoire pour contrôler les périphériques.

Grâce à cette méthode, il n'y a donc pas besoin d'instructions assembleur particulières et il n'y a pas non plus besoin de mettre en place des mécanismes de protection particuliers, car il suffit d'interdire ou d'autoriser l'accès à la zone mémoire.

## Amélioration des performances via DMA

Si on utilise simplement le contrôleur d'un disque dur, lorsque l'on veut lire sur ce dernier, le contrôleur va lire un bloc à partir du périphérique, et le placer dans sa mémoire, ensuite, il va vérifier l'information, puis faire une interruption au système d'exploitation. Enfin, ce dernier va exécuter la routine adaptée et copier l'information dans sa mémoire centrale.

Le problème avec ce système est que le CPU est fort sollicité pour faire le transfert et ne peut donc rien faire d'autre. Une solution à cela est d'utiliser un contrôleur **DMA** ([Direct Memory Access](#)). C'est un microprocesseur spécialisé dans le transfert d'informations entre un contrôleur quelconque et la mémoire centrale.

Ce contrôleur DMA peut aussi bien se trouver sur certains périphériques que sur la carte mère.

Le contrôleur DMA possède, comme tout contrôleur, plusieurs registres. Les siens sont, l'adresse (pour indiquer où les informations doivent être écrites en mémoire), le compteur d'octets (pour indiquer quelle quantité doit y être écrite), le registre de contrôle (qui spécifie de quel périphérique il s'agit, si c'est une lecture ou une écriture et de l'unité du transfert (octet, mot, etc)).

Ainsi, en utilisant un contrôleur DMA, lorsque l'on veut lire sur un disque dur, le système d'exploitation va indiquer au DMA les données dont il a besoin, puis va demander au disque une lecture. Ensuite le contrôleur disque va lire et vérifier les blocs et c'est le DMA qui va se charger de transférer directement les informations dans la mémoire. Une fois le transfert terminé, le DMA fait une interruption au système d'exploitation.

De cette manière, le CPU peut faire d'autres choses durant le transfert

## Vol de cycle

Attention toutefois, lorsque le DMA travaille, il se peut qu'il entre en conflit avec le processus, car les mémoires ne peuvent faire qu'un accès par cycle.

Le DMA ne pouvant pas attendre aussi longtemps que le processeur, parce qu'il risque de perdre des informations, il a donc priorité d'accès à la mémoire. On dit alors qu'il y a un **vol de cycle** du processeur.

---

Revision #2

Created 5 January 2024 12:36:48 by SnowCode

Updated 6 January 2024 19:13:15 by SnowCode