Université Nice SOPHIA ANTIPOLIS

Architecture des ordinateurs

Cours (12x2h00) :

- Frédéric Mallet - fmallet@unice.fr

TP (12x2h00 - 2 groupes) :

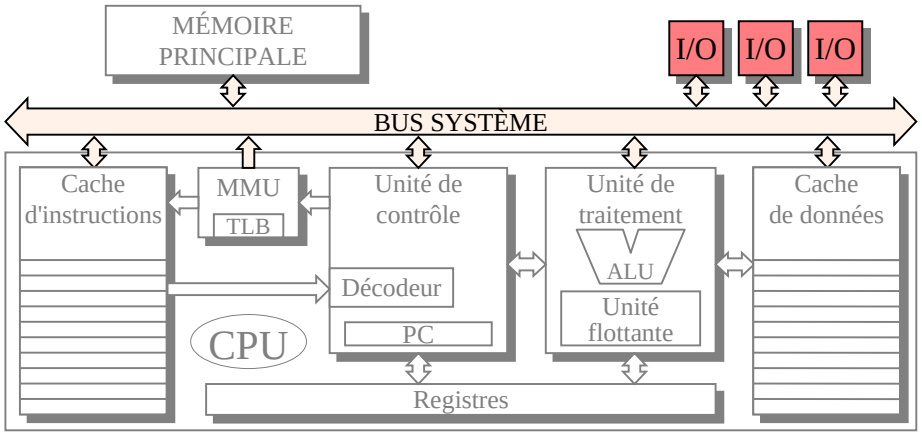
- Jean-Pierre Lips - Jean-Pierre.LIPS@unice.fr
- Christophe Delage – Christophe.Delage@sophia.inria.fr

http://deptinfo.unice.fr/~fmallet/archi

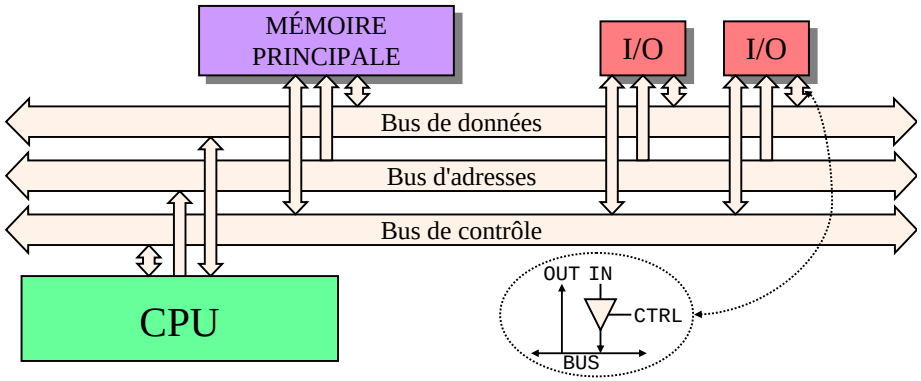
Structure du cours

- Performances : évolution et comparaison
- Codage de l 'information
- Fonctions logiques et éléments mémoires
- Systèmes à microprocesseurs
- La famille Intel - 80x86
- Conception d'architectures numériques - VHDL
- Éléments avancés (parallélisme, mémoire, ...)
  - **Périphériques et bus système**

Structure et gestion du bus



Maître - esclave



- Pour chaque **transaction** entre 2 dispositifs
  - Le bus a un **maître** (souvent le CPU), qui initie la transaction
  - Et un **esclave**

Classification des bus

Les bus peuvent être classifiés selon plusieurs paramètres:

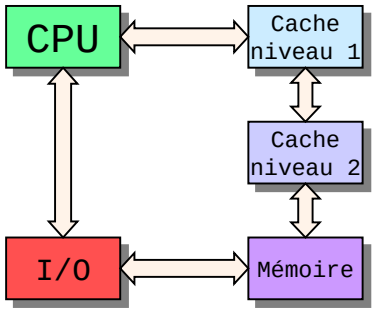
- Largeur du bus
  - Adresses
  - Données
- Type
  - Dédié
  - Multiplexé
- Timing
  - Synchrone
  - Asynchrone
- Type de transfert de données
  - Read
  - Write
  - Read-modify-write
  - Read-after-write
  - Block
- Méthode d'arbitrage
  - Centralisée
  - Distribuée

Largeur du bus système

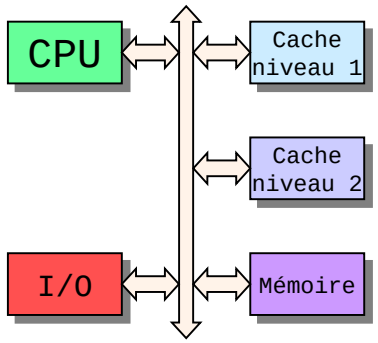
- La **taille** du bus de données est un élément clé pour la performance globale du système
  - Souvent le bus de données a la même taille que les registres du processeurs, sinon cela impacte les performances
  - E.g. bus 8 bits et registres 16 bits => 2 cycles de l'horloge du bus (# horloge processeur) pour chaque accès.
- La taille du bus d'adresse détermine la **capacité** maximale du système de mémoire
  - Le bus d'adresse peut être partagé avec celui de données (**multiplexage**).
  - Si il est **dédié** les performances sont meilleures.

Type de bus système

Dédié:

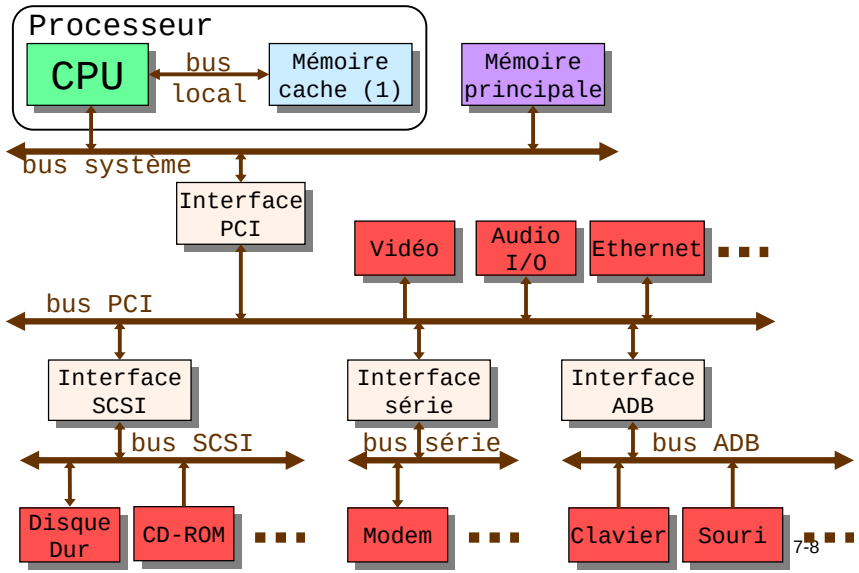


Multiplexé:



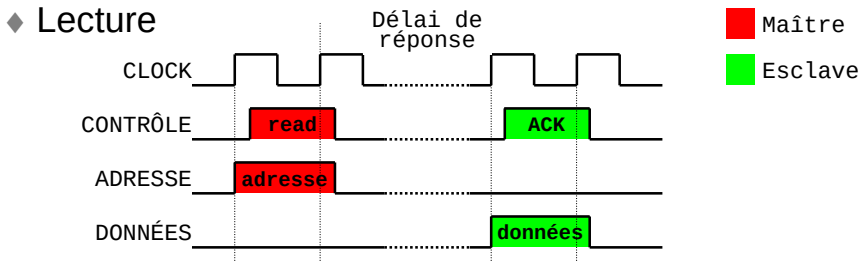
La fréquence d'opération du bus système dépend du nombre de dispositifs. Pour diminuer le délai de propagation, les bus sont, en général, organisés de façon **hiérarchique**.

Exemple - Apple Macintosh 7200

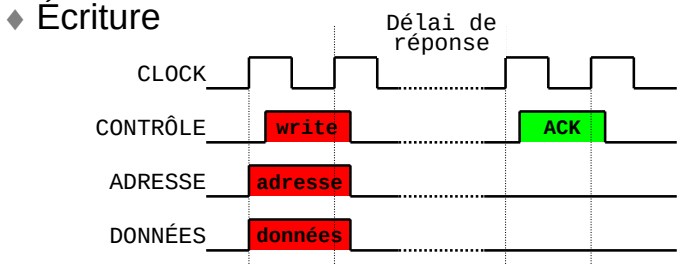


Les bus synchrones

- Lecture



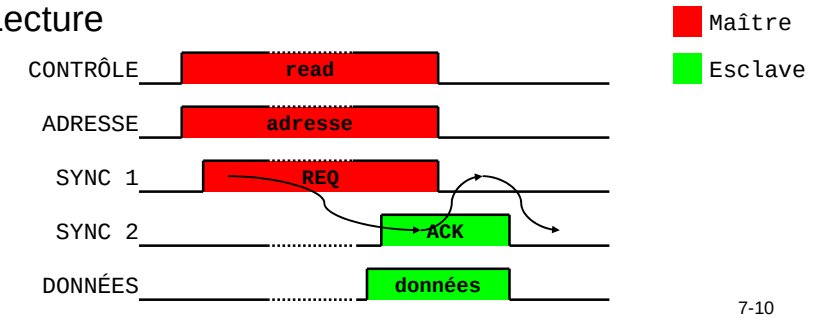
- Écriture



Les bus asynchrones

Les bus asynchrones ne nécessitent pas d'horloge. Ils utilisent des signaux spéciaux (ex. REQ, ACK) pour gérer les transferts des données (*handshaking*).
Le timing asynchrone est, en général, plus rapide que le timing synchrone. Son contrôle est toutefois plus difficile.

- Lecture



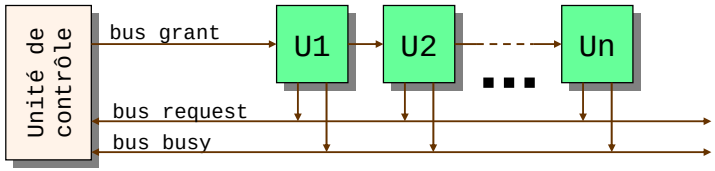
Types de transfert de données

Arbitrage du bus

- Si un dispositif souhaite communiquer avec un autre (par exemple, pour lui demander des données), il doit:
  1. Obtenir le contrôle du bus
  2. Envoyer une demande à l'autre dispositif
  3. Attendre les données
  4. Relâcher le bus
- L'**arbitrage** du bus est le mécanisme qui alloue le contrôle du bus aux dispositifs qui le demandent, évitant tout conflit.

Arbitrage du bus

- L'arbitrage peut être
  - **Distribué** : géré localement dans chaque dispositif,
  - **Centralisé** : géré par une unité de contrôle dédiée qui réalise un parmi plusieurs algorithmes possibles



- Daisy-chaining (SCSI, MIDI) :
  - Une ligne **unique** pour demander le bus à l'unité d'arbitrage, le numéro le plus faible est le plus prioritaire.

Exemple: le bus PCI

- Le bus **PCI** (Peripheral Component Interconnect) est un bus synchrone, avec un système d'arbitrage centralisé, développé par Intel (1990)
  - PCI 1.0 publié en 1992, PCI 2.0 publié en 1993
  - Remplace le bus **ISA** (Industry Standard Architecture) qui disparaît en 1998
- PCI 2.2
  - bus 32 bits à 33 MHz (soit une bande passante maxi de 133 Mo/s) (la plus répandue) ;
  - bus 64 bits à 66 MHz (soit une bande passante maxi de 528 Mo/s), utilisé sur certaines cartes mères professionnelles ou sur des serveurs

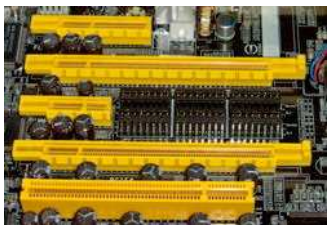
Exemple: le bus PCI

- PCI-X (réservé aux machines professionnelles) :
  - bus 64 bits à 133 MHz (bande passante max = 1066Mo/s)
  - PCI-X 2.0 : 266MHz (bande passante max=2133 Mo/s)
- **PCI Express**
  - Dérivé du *PCI*, destiné à le remplacer
  - Remplace également le bus AGP
- Mini PCI
  - dérivé du PCI 2.2 pour les ordinateurs portables

- PCI vs PCIe :
  - Contrairement au PCI, la bande passante est dédiée à chaque périphérique
  - Ligne 1-bit série bi-directionnelle (x1, x2, x4, x8, x16)
  - PCIe x16 (bande passante max = 8Go/s)

Exemple: le bus PCI

- Il y a 50 lignes obligatoires, divisées en 5 groupes:
  - Système (clock, reset, etc.)
  - Adresses et données (32-bit multiplexées)
  - Contrôle d'interface
  - Arbitrage
  - Erreur



http://fr.wikipedia.org

Exemple: le bus PCI

- Les 50 lignes optionnelles sont divisées en 4 groupes:
  - Interruption
  - Support du cache
  - Extension à 64 bits
  - JTAG/Boundary scan
- Les commandes possibles sont:
  - Interrupt acknowledge
  - Special cycle
  - I/O read
  - I/O write
  - Memory read
  - Memory read line
  - Memory read multiple
  - Memory write
  - Memory write and invalidate
  - Configuration read
  - Configuration write
  - Dual address cycle

Exemple 2 : le bus SCSI

- Le standard SCSI (*Small Computer System Interface*) permet la connexion de périphériques de type différent (disques ou ordinateurs)
  - Utilise un contrôleur (**adaptateur ou contrôleur SCSI**) généralement connecté au bus PCI
  - C'est un bus série
- La largeur du bus détermine le nombre d'unités physiques (8 bits = 8 unités)
  - Le contrôleur compte pour 1 unité
  - Déporte l'intelligence vers le périphérique ce qui libère le processeur d'une partie du travail

Le bus SCSI - adressage

- L'adressage se fait par
  - Un **identificateur** (numéro de périphérique, choisi par un cavalier : 1-7)
  - Un **LUN** (*Logical Unit Number*) qui identifie l'unité logique sur le périphérique, jusqu'à 8 (e.g. un lecteur de CD-ROM à +s tiroirs)
  - Un **numéro de carte**, pour distinguer les cartes lorsqu'il y a +s adaptateurs SCSI

Exemple 2 - Le bus SCSI

- Bus asymétrique (*Single Ended* - **SE**)
  - Chaque canal circule sur un fil
  - *Narrow* : 8 fils = 8 bits
  - *Wide* : 16 fils = 16 bits
  - Sensible aux interférences – distance limitée
- Bus différentiel (**DIFF**)
  - Deux fils par canal
  - Mode LVD (*low voltage differential*) : 3.3V
  - Mode HVD (*high voltage differential*) : 5V

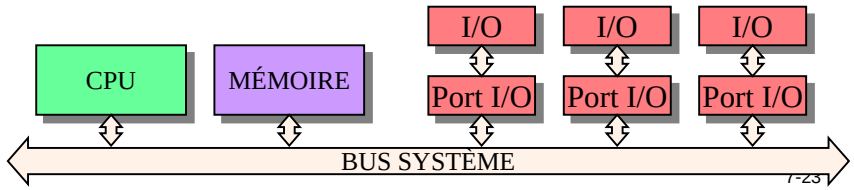
Norme SCSI

- 1986 : SCSI-1 bus à 4,77MHz, 8 bits, 5Mo/s
- 1992 : SCSI-2 (18 commandes communes)
  - *Wide SCSI-2* : bus de 16 bits, 10Mo/s
  - *Fast SCSI-2* et *Fast Wide SCSI-2* : mode synchrone, 10Mo/s et 20Mo/s
  - Fast-20 et Fast-40 : x2 et x4
- 2000 : SCSI-3 (nouvelles commandes)
  - *Ultra-80*, *Ultra-160*, *Ultra-320* et *Ultra-640*

Norme	Largeur du bus	Vitesse du bus	Bande passante	Connectique
SCSI-1 (*Fast-5 SCSI*)	8 bits	4.77 MHz	5 Mo/sec	50 broches (bus asymétrique ou différentiel)
SCSI-2 - Fast-10 SCSI	8 bits	10 MHz	10 Mo/sec	50 broches (bus asymétrique ou différentiel)
SCSI-2 - Wide	16 bits	10 MHz	20 Mo/sec	50 broches (bus asymétrique ou différentiel)
SCSI-2 - Fast Wide 32 bits	32 bits	10 MHz	40 Mo/sec	68 broches (bus asymétrique ou différentiel)
SCSI-2 - Ultra SCSI-2 (Fast-20 SCSI)	8 bits	20 MHz	20 Mo/sec	50 broches (bus asymétrique ou différentiel)
SCSI-2 - Ultra Wide SCSI-2	16 bits	20 MHz	40 Mo/sec	
SCSI-3 - Ultra-2 SCSI (Fast-40 SCSI)	8 bits	40 MHz	40 Mo/sec	
SCSI-3 - Ultra-2 Wide SCSI	16 bits	40 MHz	80 Mo/sec	68 broches (bus différentiel)
SCSI-3 - Ultra-160 (Ultra-3 SCSI ou Fast-80 SCSI)	16 bits	80 MHz	160 Mo/sec	68 broches (bus différentiel)
SCSI-3 - Ultra-320 (Ultra-4 SCSI ou Fast-160 SCSI)	16 bits	80 MHz DDR	320 Mo/sec	68 broches (bus différentiel)
SCSI-3 - Ultra-640 (Ultra-5 SCSI)	16	80 MHz QDR	640 Mo/sec	68 broches (bus différentiel)

Périphériques : port I/O

- Les dispositifs d'entrée/sortie (périphériques ou E/S ou I/O) ont des caractéristiques physiques très différentes de celles du processeur ou des mémoires (en vitesse et en timing): il faut dans ces cas un système d'interface.
- Afin d'uniformiser les accès aux périphériques, l'interface (port I/O) fournit également un ensemble de registres de données et de contrôle. Seulement deux opérations sont alors nécessaires, indépendamment du périphérique: la lecture et l'écriture d'un registre I/O.

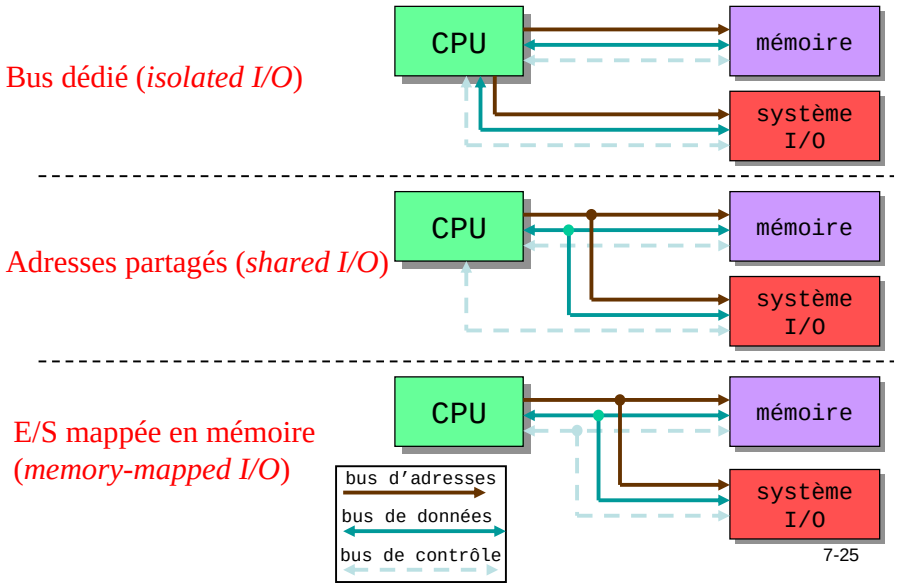


Classification des périphériques

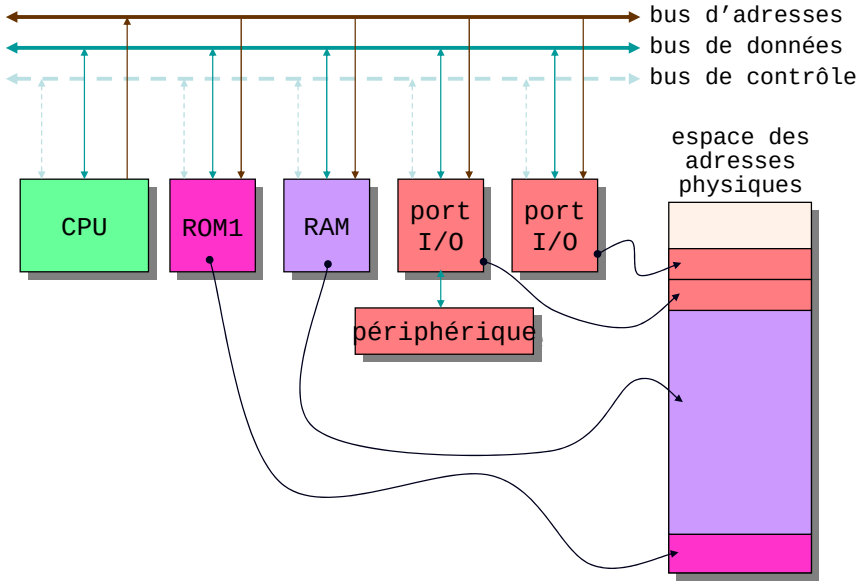
Les méthodes de gestion des périphériques peuvent être classifiées selon deux paramètres principaux:

- **Accès**
  - Bus dédié
  - Adresses partagées
  - E/S mappées en mémoire
- **Gestion**
  - Par interrogation
  - Par interruption
  - DMA
  - Processeurs d'E/S

Accès au périphériques



E/S mappées en mémoire



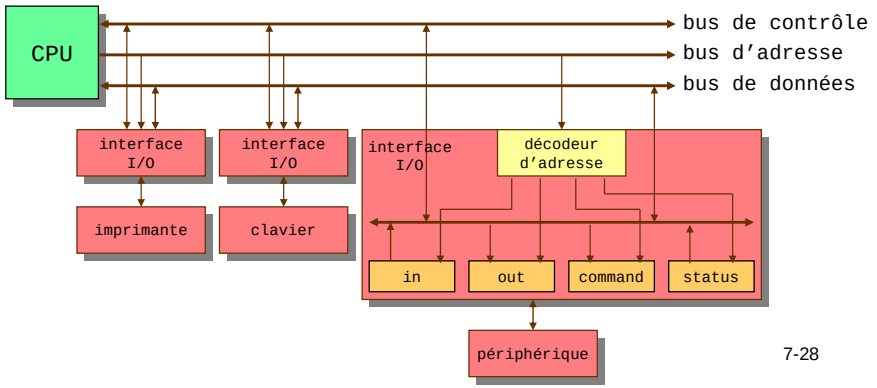
Gestion des périphériques

- Si les périphériques sont gérés **par interrogation**, le processeur doit les gérer explicitement. Il doit donc les consulter périodiquement pour déterminer leurs besoins.
- Si les périphériques sont gérés **par interruption**, ils sont capables de signaler au processeur qu'ils ont besoin d'attention. Le processeur peut alors déclencher la procédure appropriée.
- Si les périphériques ont **accès direct à la mémoire (DMA)**, ils sont capables de communiquer avec la mémoire sans la supervision du processeur, qui se limite à déclencher les transactions.
- Les **processeurs d'entrée/sortie** sont capables de gérer les transactions de façon "intelligente" (ex. à partir de programmes chargés par le système d'exploitation).

Gestion par interrogation

La gestion par interrogation demande très peu d'effort (et donc très peu de matériel) de la part des périphériques.

Le principal désavantage des I/O programmées vient du temps perdu par le processeur pendant le polling.

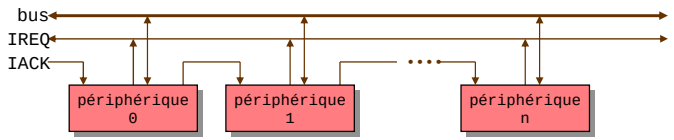


Gestion par interruption

- La gestion par interruption demande qu'un périphérique signale au processeur qu'il a besoin d'attention. Le processeur arrête l'exécution courante pour exécuter un code associé au périphérique.
- L'interruption est synchronisée par des signaux de *handshaking*:
  - le périphérique active IREQ
  - lorsque le processeur est prêt, il répond avec IACK
  - le périphérique s'identifie, de façon à ce que le processeur exécute la bonne procédure (*interrupt handler* ou *interrupt service routine*)

Gestion par interruption

- En général, il y a une seule ligne IREQ, partagée par tous les périphériques. Un seul périphérique doit recevoir le signal IACK. La solution courante est la chaîne de priorité (*daisy-chain*).



Accès direct à la mémoire (DMA)

- Pour accélérer les transferts, on peut permettre au périphérique de devenir maître du bus et d'accéder directement à la mémoire, ce qui est particulièrement avantageux lors des transferts de grands blocs.

- Pour implémenter cette fonction (**DMA** pour ***direct memory access***), l'interface du périphérique doit contenir un contrôleur DMA, un circuit capable de gérer les transferts de données.

Accès direct à la mémoire (DMA)

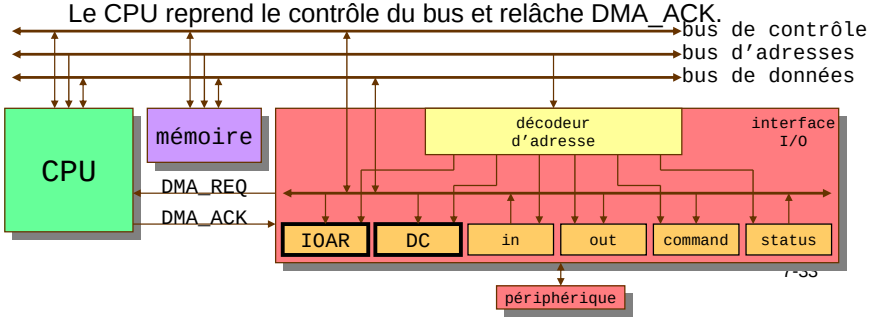
- Le contrôleur DMA contient deux registres, le registre d'adresse **IOAR** et le décompteur **DC**. IOAR stocke l'adresse du prochain mot à transférer. Il est incrémenté après chaque transfert. DC stocke le nombre de mots restant à transférer. Il est décrémenté après chaque transfert.

- Deux lignes de contrôle dédiées (***DMA_REQ*** et ***DMA_ACK***) reliant le processeur et le contrôleur DMA sont également nécessaires.

DMA - Implantation

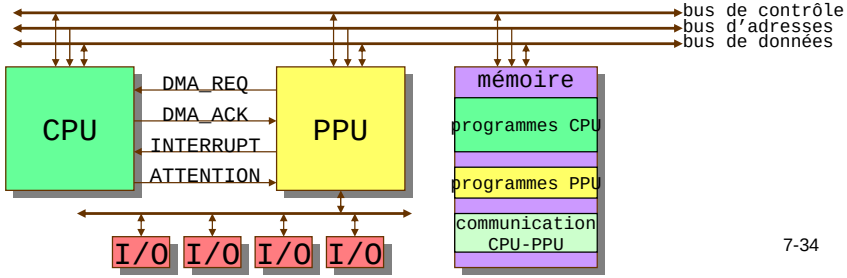
L'accès direct à la mémoire se déroule ainsi:

- Le CPU charge IOAR et DC.
- Quand le contrôleur DMA est prêt, il active la ligne DMA_REQ. Le CPU donne le contrôle du bus au contrôleur DMA et active DMA_ACK.
- Le périphérique et la mémoire échangent les données. DC est décrémenté et IOAR incrémenté après chaque transfert.
- Si DC est égal à 0, le contrôleur DMA relâche la ligne DMA_REQ. Le CPU reprend le contrôle du bus et relâche DMA_ACK.



Processeurs d'entrée/sortie

Les processeurs d'entrée/sortie (**PPU** ou *peripheral processing units*) sont l'extension du concept de DMA. Un processeur prend la place du contrôleur DMA et gère les transferts de données (généralement un seul processeur s'occupe de plusieurs périphériques).



Processeurs d'entrée/sortie

Ce type de processeur a un jeu d'instructions limité et fonctionne comme "esclave" du CPU. Il est capable d'exécuter des programmes fournis par le système d'exploitation. Une zone mémoire est normalement réservée pour la communication entre le CPU et le PPU.