

Architecture des Ordinateurs et Systèmes d'Exploitation 2

Ahmed Wade

awade@ept.sn

Objectif général de l'E.C

Démystifier l'ordinateur via la connaissance de

- Architecture d'une machine de von Neumann
- Représentation des données et des programmes
- Structure et écriture de programme en langage assembleur
- Exécution d'un programme binaire
- Comprendre le rôle et le principe de fonctionnement des systèmes d'exploitation

Objectif général de l'E.C

Démystifier l'ordinateur via la connaissance de

- Architecture d'une machine de von Neumann
- Représentation des données et des programmes
- Structure et écriture de programme en langage assembleur
- Exécution d'un programme binaire
- Comprendre le rôle et le principe de fonctionnement des systèmes d'exploitation

Objectif spécifique de l'E.C

- À la fin de l'EC Archi et Système 1
 - Comprendre comment l'ordinateur traite les informations
 - Pouvoir construire des circuits combinatoires et séquentiels
 - Pouvoir modéliser et simuler une ALU
 - Connaître les commandes de base de Linux
- À la fin de l'EC Archi et Système 2
 - Comprendre l'organisation de la mémoire
 - Pouvoir écrire des programmes en langage assembleur
 - Savoir traduire un programme en binaire
 - Maîtriser les fonctionnalités de gestion des mécanismes de base (fichier, processus, mémoire et périphériques)

Objectif spécifique de l'E.C

- À la fin de l'EC Archi et Système 1
 - Comprendre comment l'ordinateur traite les informations
 - Pouvoir construire des circuits combinatoires et séquentiels
 - Pouvoir modéliser et simuler une ALU
 - Connaître les commandes de base de Linux
- À la fin de l'EC Archi et Système 2
 - Comprendre l'organisation de la mémoire
 - Pouvoir écrire des programmes en langage assembleur
 - Savoir traduire un programme en binaire
 - Maîtriser les fonctionnalités de gestion des mécanismes de base (fichier, processus, mémoire et périphériques)

Organisation

- 5 cours, 2 TD et 4 TP
- Il y aura un partiel au cours du semestre
- Un TP à rendre
- Un examen
- Note de l'E.C avec 50% Examen, 25% Partiel, 25% Projet
- Note Projet : TP à rendre + présence + participation

Stockage des informations : registre et mémoire

Sommaire

- 1 Introduction
- 2 Mémoire sur 1 bit
- 3 Registres
- 4 Architecture de la mémoire

Introduction

Dans ce cours, nous allons nous intéresser au stockage des informations dans l'ordinateur ainsi qu'à la localisation des unités de stockage.

Zones de stockage

Il existe différentes zones de stockage de l'information au sein de l'ordinateur : registres, mémoire principale, mémoires secondaires, etc. Chacun de ces espaces se différencie par des caractéristiques propres (capacité, adressage, méthode d'accès, performances, ...), qui déterminent leur utilisation.

Localisation

La localisation physique des mémoires influent directement sur leurs performances. Plus une mémoire est loin du processeur, plus les signaux électriques mettent du temps à l'atteindre.

Le stockage des données et instructions

Unités de stockage et localisation

- A tout instant, les **données et les instructions** du programme en cours d'exécution sont **stockées** en **mémoire**
- Une partie de ces informations est à un instant donnée dans le processeur : notamment instruction en cours d'exécution + une partie des données manipulées par le programme
- **Dans le processeur**, toute donnée (temporaire) est stockée dans des **registres**
- Transfert des informations entre la mémoire et le processeur via le bus

Sommaire

- 1 Introduction
- 2 Mémoire sur 1 bit
 - Circuits séquentiels
 - Exemples
- 3 Registres
- 4 Architecture de la mémoire

Circuits séquentiels

- Circuits combinatoires
- Circuits séquentiels

Circuits séquentiels

- Circuits combinatoires
 - Les sorties du circuit ne dépendent que des valeurs des entrées
- Circuits séquentiels

Circuits séquentiels

- Circuits combinatoires
 - Les sorties du circuit ne dépendent que des valeurs des entrées
- Circuits séquentiels
 - Ajout des notions d'état et de mémoire
 - Ajout de la notion de temps (horloge)

Circuits séquentiels

- Circuits combinatoires
 - Les sorties du circuit ne dépendent que des valeurs des entrées
- Circuits séquentiels
 - Ajout des notions d'état et de mémoire
 - Ajout de la notion de temps (horloge)
 - Les sorties du circuit dépendent
 - Des valeurs en entrée
 - De valeurs calculées précédemment
 - De l'état dans lequel on se trouve

Mémoire sur 1 bit

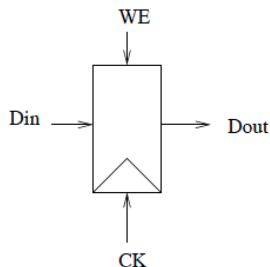
- Circuit séquentiel basique : mémoire sur 1 bit
 - Possède une variable binaire codée sur 1 bit
 - Sa valeur est conservée ou modifiable dans le temps
- Mémoires 1 bit peuvent être réalisées avec des bascules
 - Plusieurs types de bascules : RS, D, JK ...

Sommaire

- 1 Introduction
- 2 Mémoire sur 1 bit
- 3 Registres**
 - Registre 1 bit
 - Chronogramme
 - Stockage de donnée dans un registre
 - Les registres du processeur MIPS
- 4 Architecture de la mémoire

Registre

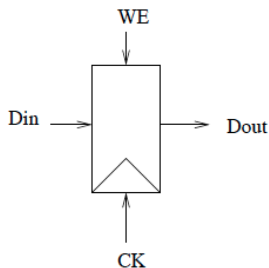
Registre 1 bit (Bascule D flip-flop 1 bit)



- CK désigne le signal d'horloge globale cadencant le processeur.
- WE signifie Write Enable
- Din signifie Data Input, Dout signifie Data Output

Fonctionnalité d'un registre mémorisant sur front montant

Registre 1 bit mémorisant sur front montant



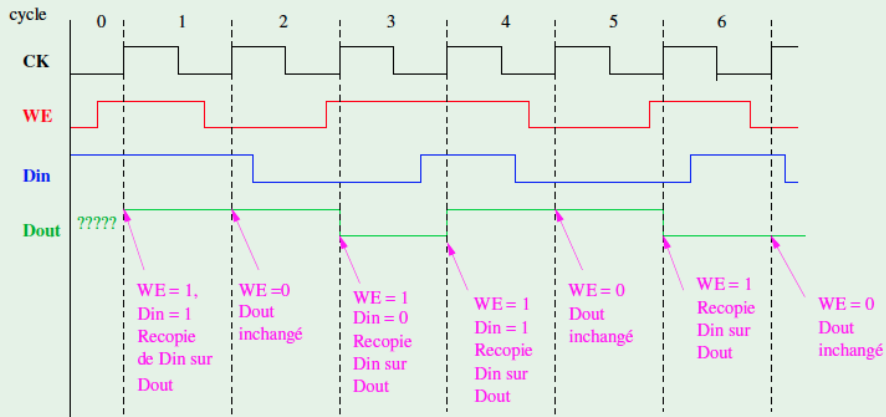
Lorsque CK passe de 0 à 1 (front montant)

- si $WE = 1$ alors la valeur de Din est recopiée sur Dout
- sinon ($WE = 0$) Dout reste inchangé (conserve la valeur précédente / effet mémoire).

A tout autre instant (front descendant donc), quel que soit WE, Dout est inchangé.

Exemple de chronogramme

Chronogramme des signaux entrée et sortie d'un registre 1 bit sur plusieurs cycles



Exemple de chronogramme

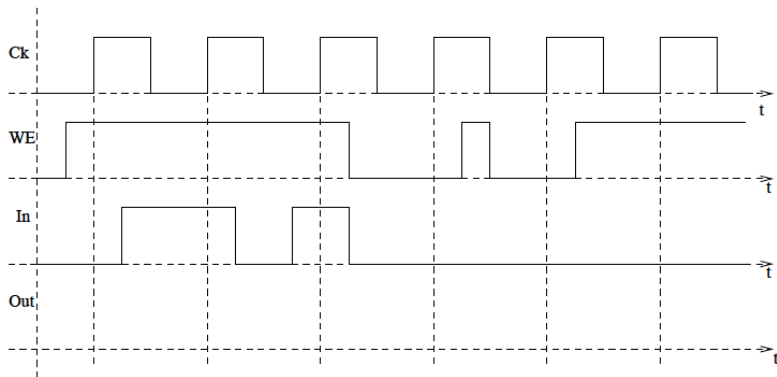


FIGURE 2 – Chronogramme d'un registre 1 bit

Registre n bits

Registre n bits

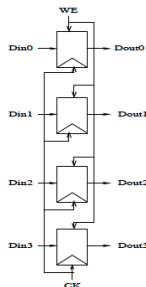
Un registre n bits est un élément de stockage qui permet la mémorisation de n bits en parallèle. Cet élément est constitué sur la base d'une mise en parallèle de n bascules mémorisant chacune 1 bit

Réalisation d'un registre n bits

Registre n bits

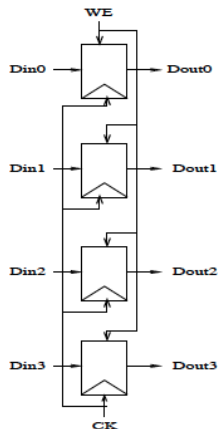
Pour réaliser un registre n bits, il suffit de mettre en parallèle n registres 1 bit et de connecter les signaux WE et CK de tous ces registres au même signal.

Registre 4 bits à partir de 4 registres 1 bits



Exemple de stockage d'un mot dans un registre

Comment stocker le mot 1100 dans le registre 4 bits ?

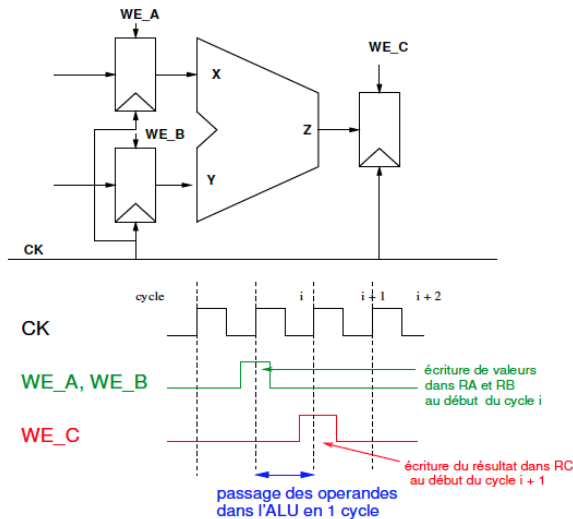


Lorsque CK passe de 0 à 1 (front montant)

- 1 Positionner les valeurs en entrée : $Din0 = 0$, $Din1 = 0$, $Din2 = 1$ et $Din3 = 1$
- 2 Positionner WE à 1
- 3 Au prochain front montant de CK (cycle suivant) on aura : $Dout = Din$ soit $Dout0 = 0$, $Dout1 = 0$, $Dout2 = 1$, $Dout3 = 1$
- 4 Si on change Din et si $WE = 0$ alors Dout vaudra toujours 1100.

Si 1) et 2) ont lieu au cycle i , alors 3 a lieu au cycle $i + 1$

Transfert d'informations dans l'ALU



Les registres d'un processeur

Les registres d'un processeur

- L'architecture du processeur définit le nombre, la taille et le nom des registres du processeur
- Certains peuvent être manipulés explicitement par le programmeur via des instructions

Manipulation des registres

Affectation d'une valeur à un registre par le biais d'instructions

- Placer explicitement une valeur dans R : $R \leftarrow 2$
- Placer le résultat d'un calcul dans R : $R \leftarrow A + B$
- Placer une donnée qui était stockée en mémoire : $R \leftarrow Mem[@]$

Les registres d'un processeur

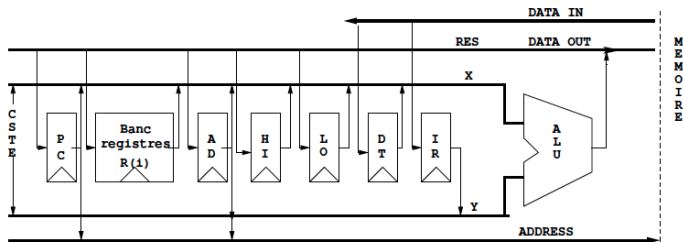
Les registres d'un processeur

- L'architecture du processeur définit le nombre, la taille et le nom des registres du processeur
- Certains peuvent être manipulés explicitement par le programmeur via des instructions

Manipulation des registres

Affectation d'une valeur à un registre par le biais d'instructions

- Placer explicitement une valeur dans R : $R \leftarrow 2$
- Placer le résultat d'un calcul dans R : $R \leftarrow A + B$
- Placer une donnée qui était stockée en mémoire : $R \leftarrow Mem[@]$

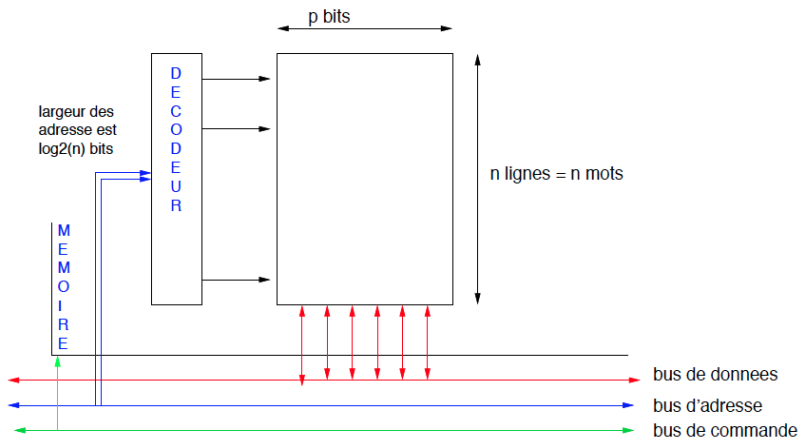


- Les registres du MIPS font 32 bits
- Les registres R_0, \dots, R_{31} sont des registres de travail et R_0 vaut toujours 0
- PC (Programme Counter) contient l'adresse de l'instruction en cours d'exécution (ou la suivante)
- Hi/Lo (High/Low) sont les registres pour le résultat d'opérations de multiplication ou de division

Sommaire

- 1 Introduction
- 2 Mémoire sur 1 bit
- 3 Registres
- 4 Architecture de la mémoire**
 - Organisation et capacité
 - Fonctionnement

Architecture de la mémoire



- La mémoire peut être vue comme :
 - un tableau de n lignes de p bits
 - une suite de n mots de p bits.
- Le numéro de ligne d'un mot définit son adresse :
 - adresse du 1er mot = 0, adresse du 2ème = 1, etc.
 - adresse du dernier = $n-1$
- Les mots sont en général des octets, soit $p = 8$: on dit que l'unité adressable est l'octet.

Capacité mémoire

Définition

La capacité mémoire correspond au nombre de mots qu'elle peut stocker, c'est un nombre d'octets.

Capacité de stockage

- 1 kilo-octets ou 1 ko = 2^{10} octets
- 1 mega-octets ou 1 Mo = 2^{20} octets
- 1 giga-octets ou 1 Go = 2^{30} octets

Exercice

On considère une mémoire de huit mots de quatre octets. L'unité adressable est l'octet.

- Quelle est la capacité de cette mémoire ?
- Donner la largeur des nappes de fils données et adresse ?

Rangement de mots de plusieurs octets

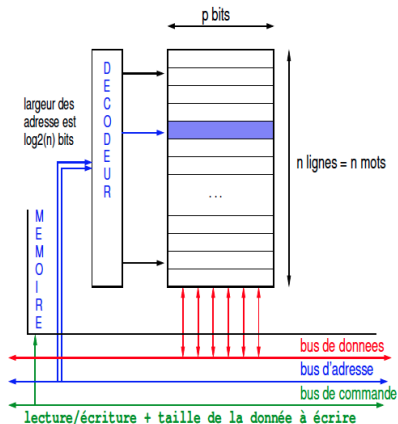
Petit/Gros boutien

- Soit $M = 0x o_3 o_2 o_1 o_0$ un mot de 4 octets rangé à l'adresse A .
- 2 rangements possibles :
 - Petit boutien : l'octet de poids faible est rangé à l'adresse la plus petite
 - Gros boutien : l'octet de poids fort est rangé à l'adresse la plus petite

Adresse	Petit boutien	Grand boutien
A	o_0	o_3
$A + 1$	o_1	o_2
$A + 2$	o_2	o_1
$A + 3$	o_3	o_0

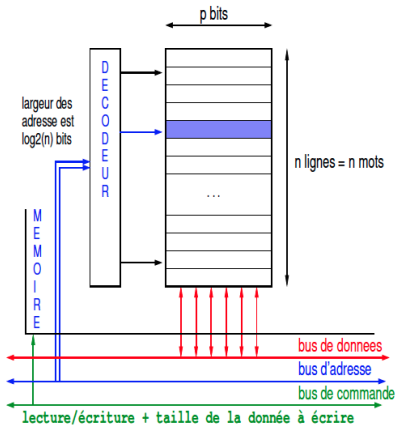
Fonctionnement de la mémoire

Lors d'un transfert



- le décodeur d'adresse sélectionne la ligne correspondant à l'adresse demandée sur le bus d'adresse,
- la commande indique l'opération = lecture ou écriture,
- lecture : les données lues sont mises sur le bus de données
- écriture : données à écrire sont celles présentes sur le bus de données

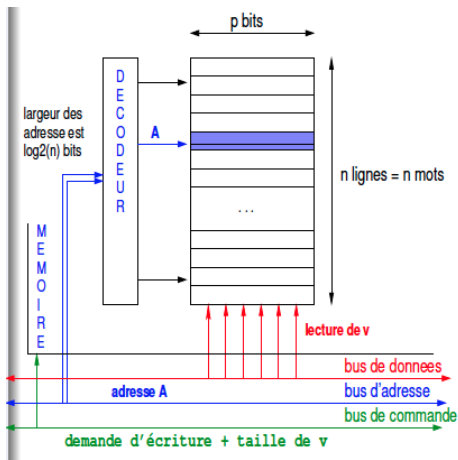
Transfert entre la mémoire et le processeur



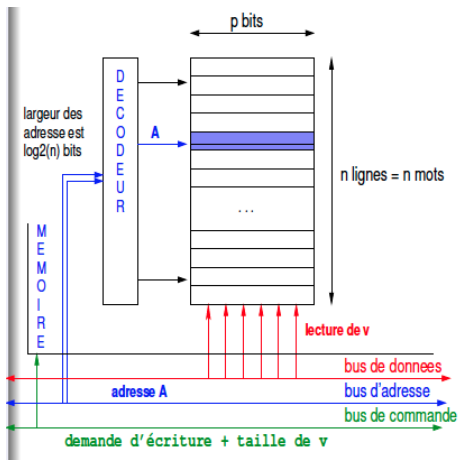
Le processeur initie les transferts de données entre le processeur et la mémoire et indique/donne à la mémoire :

- L'adresse du mot à transférer
- La taille du mot à transférer
- Le sens du transfert :
 - si processeur \longrightarrow mémoire alors c'est une écriture ou un store,
 - si mémoire \longrightarrow processeur alors c'est une lecture ou un load (chargement).
- La donnée à écrire si écriture

Protocole d'écriture de v à l'adresse A



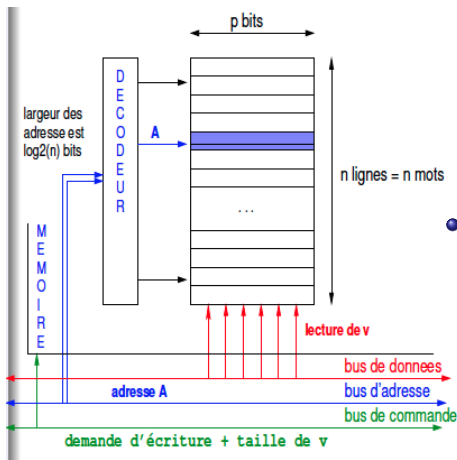
Protocole d'écriture de v à l'adresse A



Le processeur place

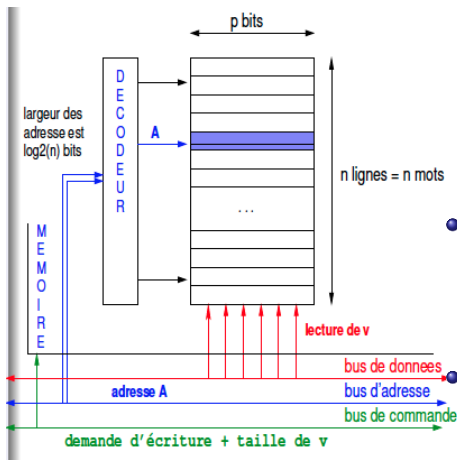
- A sur le bus d'adresse,
- v sur le bus de données,
- demande d'écriture sur le bus de commande
- la taille de la donnée v sur le bus de commande

Protocole d'écriture de v à l'adresse A



- Le processeur place
 - A sur le bus d'adresse,
 - v sur le bus de données,
 - demande d'écriture sur le bus de commande
 - la taille de la donnée v sur le bus de commande
- puis passe ensuite à l'instruction suivante si écriture non bloquante ou attend un acquittement de la mémoire sinon

Protocole d'écriture de v à l'adresse A

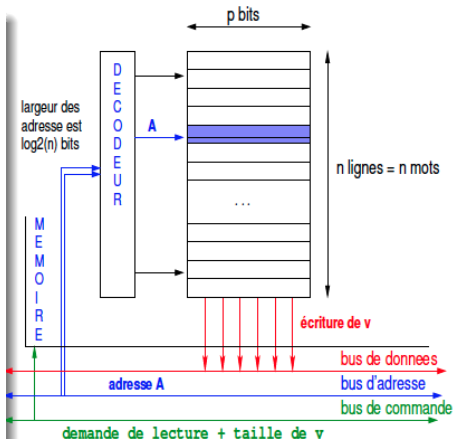


- Le processeur place
 - A sur le bus d'adresse,
 - v sur le bus de données,
 - demande d'écriture sur le bus de commande
 - la taille de la donnée v sur le bus de commande
- puis passe ensuite à l'instruction suivante si écriture non bloquante ou attend un acquittement de la mémoire sinon
- La mémoire lit l'adresse A sur le bus d'adresse, la commande d'écriture donc lit la donnée v sur le bus de donnée et place v à l'adresse A

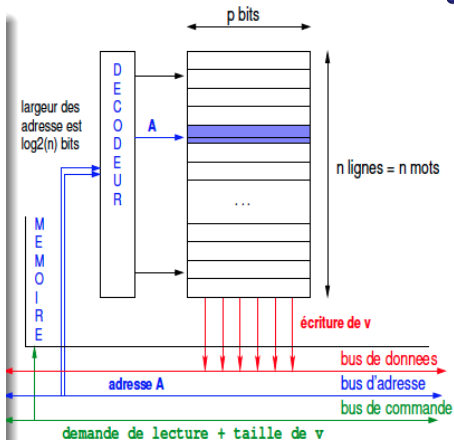
Effet d'une écriture

$\text{Mem}[A] = v$ jusqu'à la prochaine écriture à cette adresse ou à une des adresses occupée par le mot v .

Protocole de lecture d'un mot à l'adresse A

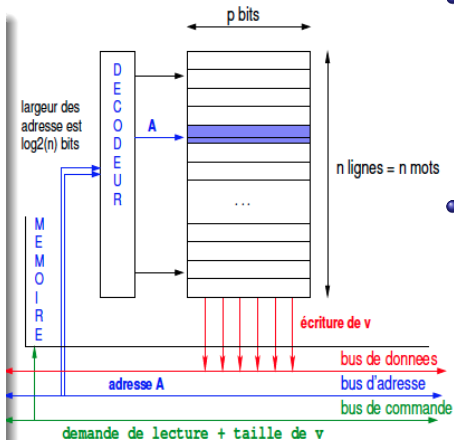


Protocole de lecture d'un mot à l'adresse A



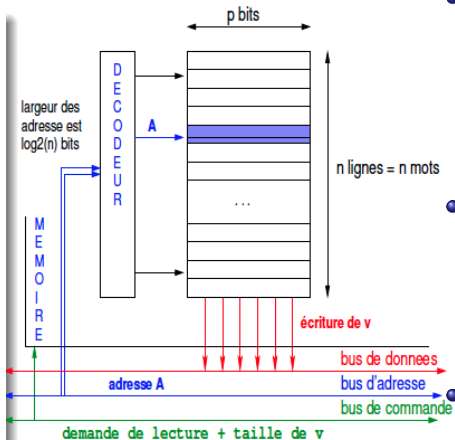
- Le processeur place
 - A sur le bus d'adresse,
 - demande de lecture sur le bus de commande
 - taille du mot à lire sur le bus de commande

Protocole de lecture d'un mot à l'adresse A



- Le processeur place
 - A sur le bus d'adresse,
 - demande de lecture sur le bus de commande
 - taille du mot à lire sur le bus de commande
- La mémoire lit l'adresse A sur le bus d'adresse et la demande de lecture, puis met le contenu de $\text{Mem}[A]$ sur le bus de données et acquitte le transfert.

Protocole de lecture d'un mot à l'adresse A



- Le processeur place
 - A sur le bus d'adresse,
 - demande de lecture sur le bus de commande
 - taille du mot à lire sur le bus de commande
- La mémoire lit l'adresse A sur le bus d'adresse et la demande de lecture, puis met le contenu de Mem[A] sur le bus de données et acquitte le transfert.
- Le processeur récupère v sur le bus de données et le place dans le registre destination

Effet d'une lecture

La valeur v lue (valeur de $\text{Mem}[A]$) est celle de la dernière écriture à cette adresse. La lecture n'est pas destructrice.

Écriture du mot 0xAABBCCDD à l'adresse 4

- Le processeur met
 - 4 sur le bus d'adresse,
 - commande d'écriture d'un mot de 4 octets sur le bus de commande
 - 0xAABBCCDD sur le bus de données
- La mémoire
 - lit l'adresse 4 (bus d'adresse) et la demande (écriture, 4 octets)
 - sélectionne les lignes 4, 5, 6 et 7
 - y écrit les 4 octets présents sur le bus de données

Effet

Adresse	Petit boutien	Grand boutien
0	?	?
1	?	?
2	?	?
3	?	?
4	DD	AA
5	CC	BB
6	BB	CC
7	AA	DD
8	?	?
9	?	?
10	?	?
...	?	?

Écriture du demi-mot 0x1234 à l'adresse 2 ?

Écriture du demi-mot 0x1234 à l'adresse 2

- Le processeur met
 - 2 sur le bus d'adresse,
 - commande d'écriture d'un mot de 2 octets sur le bus de commande
 - 0x1234 sur le bus de données
- La mémoire
 - lit l'adresse 2 (bus d'adresse) et la demande (écriture, 2 octets)
 - sélectionne les lignes 2, 3
 - y écrit les 2 octets présents sur le bus de données (de largeur 4 octets)

Effet

Adresse	Petit boutien	Grand boutien
0	?	?
1	?	?
2	34	12
3	12	34
4	DD	AA
5	CC	BB
6	BB	CC
7	AA	DD
8	?	?
9	?	?
10	?	?
...	?	?

Lecture d'un octet à l'adresse 3

- Le processeur met
 - 3 sur le bus d'adresse,
 - commande de lecture d'un mot de 1 octets sur le bus de commande
- La mémoire
 - lit l'adresse 3 (bus d'adresse) et la demande (lecture, 1 octets)
 - sélectionne la ligne 3
 - met sur le bus de données la valeur contenue dans cette ligne sur l'octet de poids faible du bus
 - acquitte le transfert
- Le processeur récupère la valeur (mise en registre)

Effet

Adresse	Petit boutien	Grand boutien
0	?	?
1	?	?
2	34	12
3	12	34
4	DD	AA
5	CC	BB
6	BB	CC
7	AA	DD
8	?	?
9	?	?
10	?	?
...	?	?

Lecture d'un demi-mot à l'adresse 4

Lecture d'un demi-mot à l'adresse 4

Effet

- Le processeur met
 - 4 sur le bus d'adresse,
 - commande de lecture d'un mot de 2 octets sur le bus de commande
- La mémoire
 - lit l'adresse 4 (bus d'adresse) et la demande (lecture, 2 octets)
 - sélectionne les lignes 4 et 5
 - met sur le bus de données les valeurs contenues dans ces lignes sur le bus
 - acquitte le transfert
- Le processeur récupère la valeur 0xCCDD (petit boutien) ou 0xAABB (grand boutien).

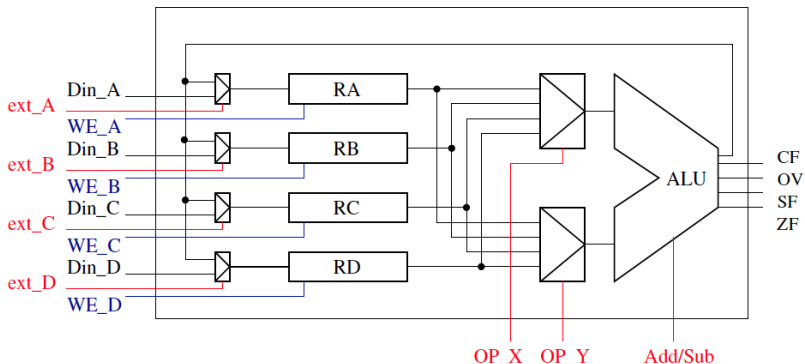
Adresse	Petit boutien	Grand boutien
0	?	?
1	?	?
2	34	12
3	12	34
4	DD	AA
5	CC	BB
6	BB	CC
7	AA	DD
8	?	?
9	?	?
10	?	?
...	?	?

Chemin de données

Notion de chemin de données

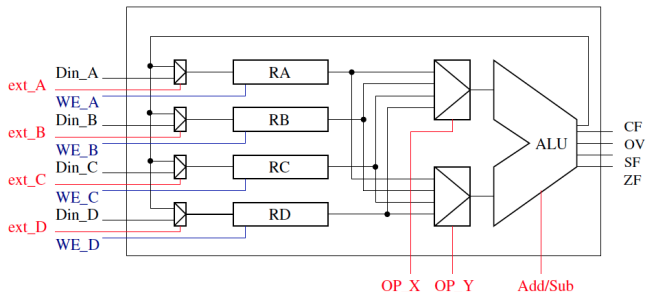
- Un chemin de données est un ensemble d'unités fonctionnelles électroniques (ALU, multiplexeurs, registres) qui effectuent des opérations de traitement de données.
- Le chemin de données permet de représenter le mouvement des données aux travers d'éléments le composant et au cours du temps.
- Les processeurs ont un chemin de données. Il est commandé par un séquenceur (partie contrôle) qui orchestre les mouvements de données pour réaliser les traitements dictés par l'exécution des instructions, lors de l'exécution d'un programme.

Exemple de chemin de données



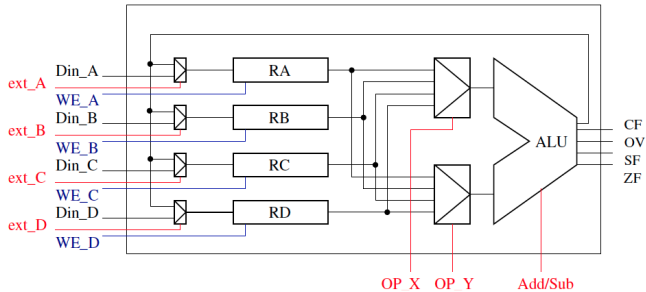
- CF : carryflow ou le bit de retenue sortante
- OV : overflow ou dépassement de capacité sur entiers relatifs
- SF : signflow ou bit de signe du résultat
- ZF : zeroflow ou signal indiquant si le résultat est nul

Exemple sur le chemin de données



Comment réaliser le calcul suivant : $RD \leftarrow 12 - 45 + 2$?
 Quelles sont les valeurs contenues dans les registres ?

Exemple sur le chemin de données



Calculer le maximum de deux nombres n et m ?