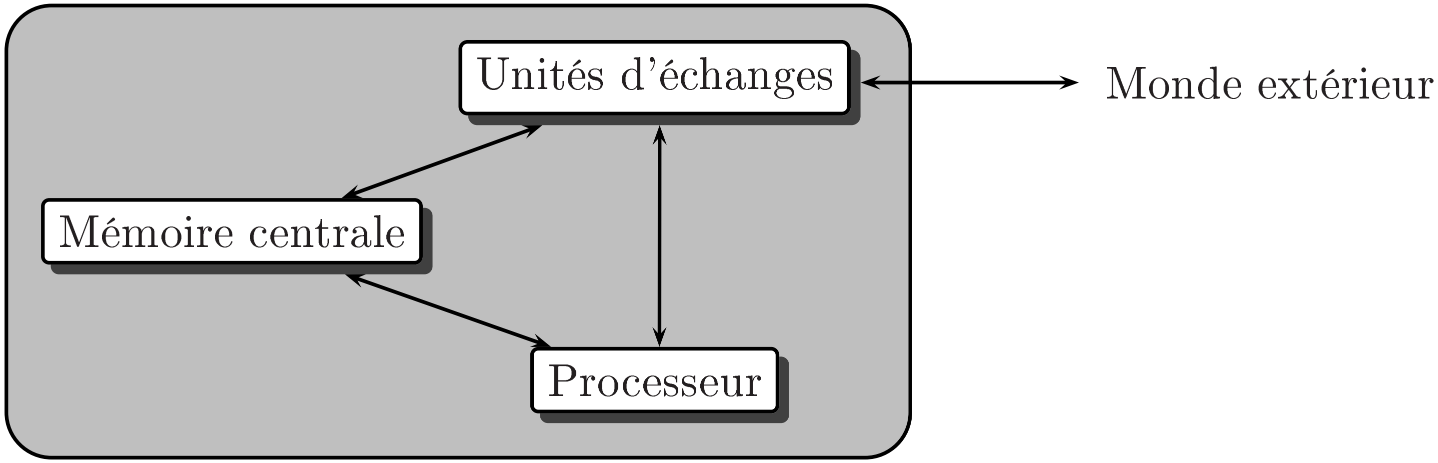
Architecture des Ordinateurs

Le processeur

Florent Hivert

Rappel : composants essentiels d'un ordinateur



1. Introduction
2. Composants du processeurs
3. Exemple d'exécution
4. Le Pipeline
5. RISC et CISC
6. Aléas et pipeline
7. Exemples de séquencement
8. Surface de la puce

Introduction : le processeur et son environnement
Notion de Tampon (anglais "Buffer")

Le processeur ne lit pas directement les données de la mémoire ou des périphérique : utilisation de tampon.

* Espace de stockage provisoire ;
* Plus rapide d'accès ;
* La mémoire sert de tampon au disque, le cache à la mémoire,
* les registres servent de tampon au cache et même : registre servant de tampons pour d'autres registres.

Introduction : le processeur et son environnement
Notion de contexte

Environnement dont a besoin un programme pour s'exécuter ou pour reprendre son exécution.

* Binaire du programme (au moins son adresse mémoire/disque) ;
* État courant des registres :
  - registres généraux ;
  - compteur ordinal ;
  - registres de drapeaux ;
  - registres de piles.

Tout programme en exécution possède un contexte qui lui est propre (y compris le système).

$\Longrightarrow$ changement de contexte.

Notion de mode de fonctionnement

Tout n'est pas faisable par tout les programmes.

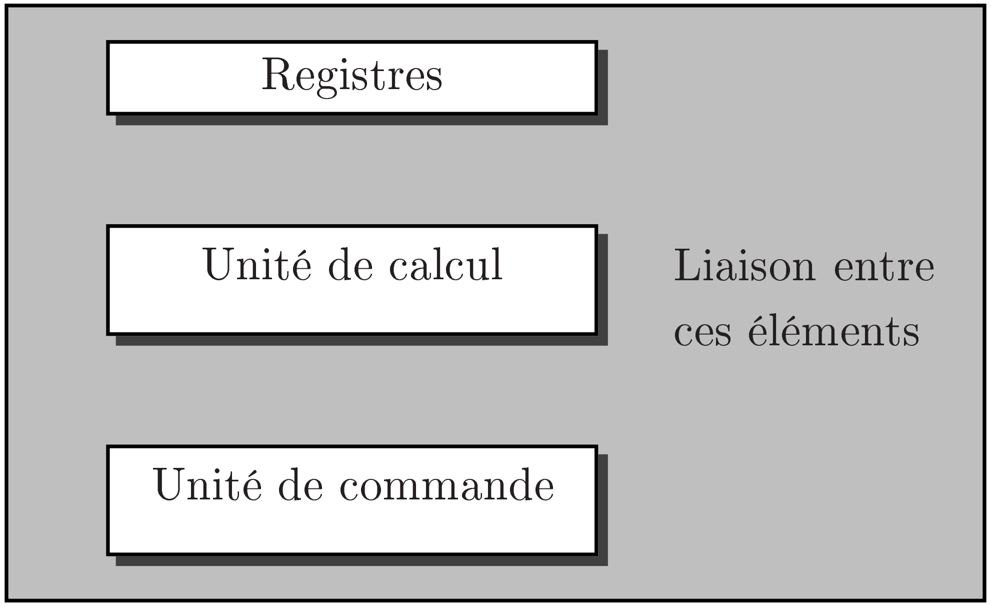
$\Longrightarrow$ droits d'accès (mode de fonctionnement).

Au moins deux modes :

1. mode privilégié, donne accès à tout, est réservé à une partie du système d'exploitation ;
2. mode normal, avec limitation des accès en mémoire et sur les ports d'entrée/sortie (protection des autres utilisateurs et du système).

Minimum pour garantir le bon fonctionnement de la machine contre les utilisateurs malveillants ou maladroits.

Les composants du processeur



Les composants du processeur : les registres

* Fonction identique à un emplacement mémoire ;

* Caractéristiques :

  - capacité fixe ;
  - accès très rapide ;
  - nombre limité ;
  - identification rapide dans l'instruction ;

* Actions usuelles :
  - chargement depuis mémoire, unité de calcul, autre registre ;
  - rangement stockage (lecture non modifiante vers mémoire ou autre registre)

Différents types de registres

Dépendant de l'architecture

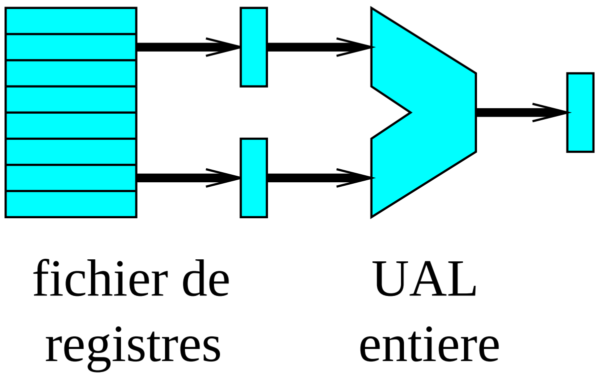
* Registres généraux (entiers, flottants, adresses) ;
  Tampon mémoire de 20 à 1024 ;
* Registres spécialisés d'adressage (souvent=entier car calcul adresse=calcul entier simple)
  Également registre de base inaccessible par l'utilisateur ;
* Registres de drapeaux (indicateurs de conditions, débordement, exception, problème matériel)

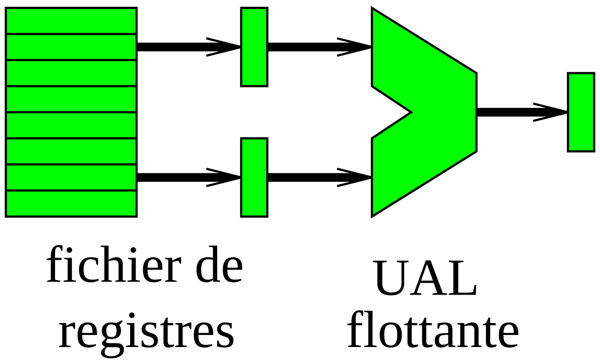
Fichier de registres

* Les processeur RISC : nombreux registres pour remplacer le manque d'instructions et la lenteur des accès mémoires.

* On regroupe les registres en fichier (bancs) proche des unités de calculs

* En général, chaque type de données (entier, flottant) dispose de son propre ensemble.

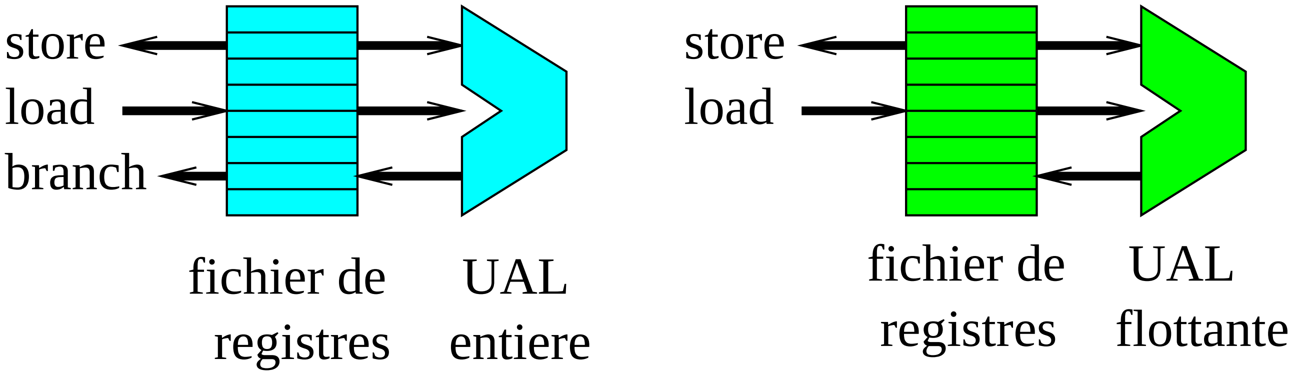




Registre : chemin d'accès

∗ On regroupe fichiers de registres et unités logiques. L'information communique avec ces unités par chemin d'accès

Exemple : DEC 21064



Exemple de registres

* Compteur Ordinal (Program Counter, Instruction Pointer) ;
* Accumulateur (registre de sortie des unités de calcul, inaccessible actuellement) ;
* Pointeurs de piles ;
* Registre de contextes (accès mémoire) ;
* E/S, interruption (inaccessible à l'utilisateur) ;

Fenêtre de Registre

Registre purement matériel inaccessibles au programmeur ;

Le processeur change les numéro des registres pour les rendre accessibles (appel de sous programme)

Exemple : SPARC Fenêtre de 24 registres :
- 8 partagés avec l'appelant ;
- 8 pour l'utilisation locale ;
- 8 partagés avec tout le programme ;

Inconvénient : prend le la place si inutile, interruption si plus de registre $\Longrightarrow$ le compilateur sait faire facilement.

Fenêtre de Registre : exemple

Passage d'arguments entre deux procédures :

Appelant	Appelé	Noms Locaux		Matériel
entrée		$Re_1 - Re_4$		$R_{10} - R_{13}$
locaux		$Ri_1 - Ri_6$		$R_{14} - R_{19}$
sortie	entrée	$Rs_1 - Rs_3$	$Re_1 - Re_3$	$R_{20} - R_{22}$
	locaux		$Ri_1 - Ri_4$	$R_{23} - R_{26}$
	sortie		$Rs_1 - Rs_3$	$R_{27} - R_{29}$

Les composants du processeur : Unité de calcul

Unité arithmétique et logique (UAL, "ALU" en anglais) : calcul sur le contenu des registres ;

Exemple : Addition de deux contenus, résultat Somme, retenue et indicateurs divers (débordement, résultat nul...).

Opérations possibles :

- addition/soustraction entière
- addition/soustraction flottante
- multiplication/division entière
- multiplication/division flottante
- racine carrée, ...

Souvent regroupées au sein d'une unité entière et d'une unité flottante avec chemins d'accès et registres distincts.

Les composants du processeur : Unité de commande

Chargé du déroulement des instructions :
- chargement des instructions ("fetch") ;
- décodage des instructions ;
- accès aux données ;
- séquencement des différentes phases.

Composition :
- Décodeur, séquenceur ;
- Zone de tampon d'instructions (registre, pile, file et cache) ;
- Compteur ordinal ("Program Counter") ;
- Générateur d'adresses ;

Communication entre modules

* Communication entre les différentes partie de l'ordinateur (processeur, mémoire, unités d'échanges) ou entre les composants interne du processeur ;

* Réalisée par des fils. Pour les données, regroupement en parallèle, on parle de Bus (mais plutôt autoroute) ;

* Vitesse du bus : débit théorique (en fait plus lent car partage) ;

* Nombre de bus dépendant de l'architecture :

Séparations possibles :	bus de données	bus de commandes
	bus d'adresses	bus d'entrées sorties

* bidirectionel ou dédoublé.

Séquencement du déroulement d'un instruction

* Le déroulement est découpé en phase : **Les micro-commandes**
  - Activation des différentes phases ;
  - Distribution des données à chaque phase.
* Deux solutions possibles :
  - Micro-programme : micro-commandes en mémoire morte (ROM) ;
  - Automate cablé (seulement si simple).
* Cadencement par une horloge (cycle d'horloge).

Phases du séquencement

* Chargement ("fetch") de l'instruction ;
* Modification du compteur ordinal ;
* Décodage de l'instruction ;
* Localisation mémoire des données (adresse ou registre) ;
* Chargement des données ;
* exécution des calculs ;
* Rangement des résultats en mémoire ou registre.

Plusieurs de ces phases peuvent être regroupées et avoir lieu simultanément.

Notion d'interruption

Détection aussi rapide que possible d'évènements pouvant perturber ou concourir au déroulement du programme :

- erreur de calcul arithmétique ;
- débordement d'adresses ;
- exécution d'instructions inconnue ou interdite ;
- appel au système d'exploitation ;
- opération d'entrée / sortie ;
- message d'un autre composant de l'ordinateur (processeur) ;
- écoulement d'un intervalle de temps programmé ;
- problème matérielle (coupure de courant).

Notion d'interruption

* Caractéristique :
  - matérielles ou logicielles ;
  - masquables ou non ;
  - fatale ou non ;
* Influence sur le déroulement de l'instruction : elle peuvent intervenir au milieu de l'exécution d'une instruction.

  On peut donc soit terminer l'instruction, soit l'interrompre.
* Il faudra restituer l'état antérieur, opération complexe.

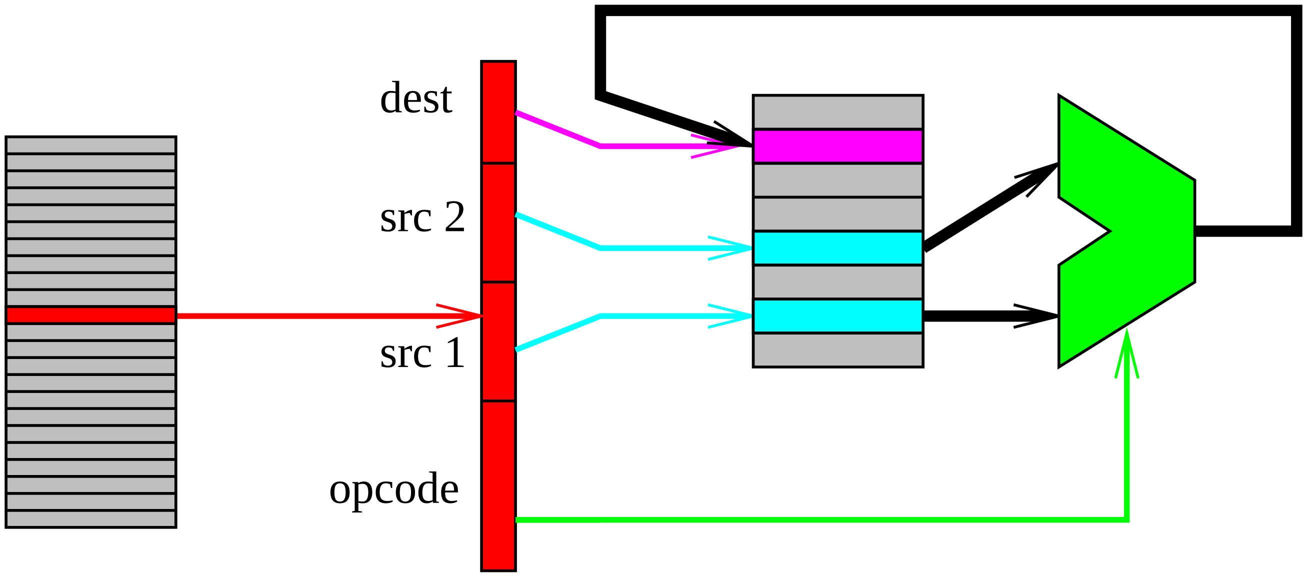
Parallélisme d'exécution : le pipeline

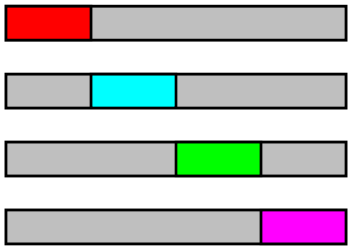
Il est difficile d'accélérer les différentes phases. La solution est de faire recouvrir partiellement les déroulement des instructions.

* Même type d'opération, données différentes : processeur vectoriel (pas toujours possible).
* Différent type d'instructions : Le pipeline

Pipeline : **Les phases du déroulement de l'instruction peuvent travailler indépendamment les unes des autres. Analogie : chaîne de montage automobile.**

Chemin de données d'une opération





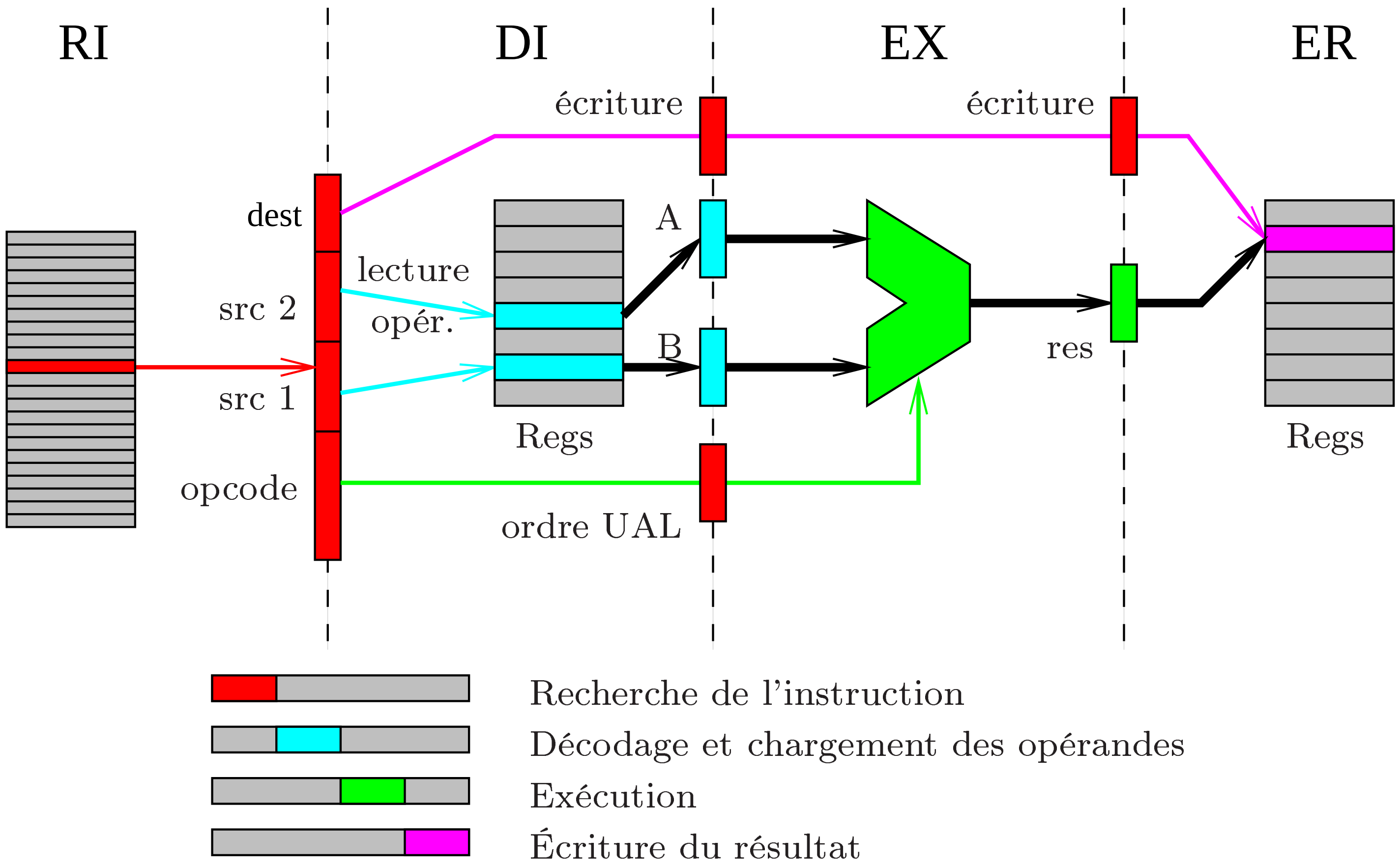
Recherche de l'instruction

Décodage et chargement des opérandes

Exécution

Écriture du résultat

Principe du pipeline



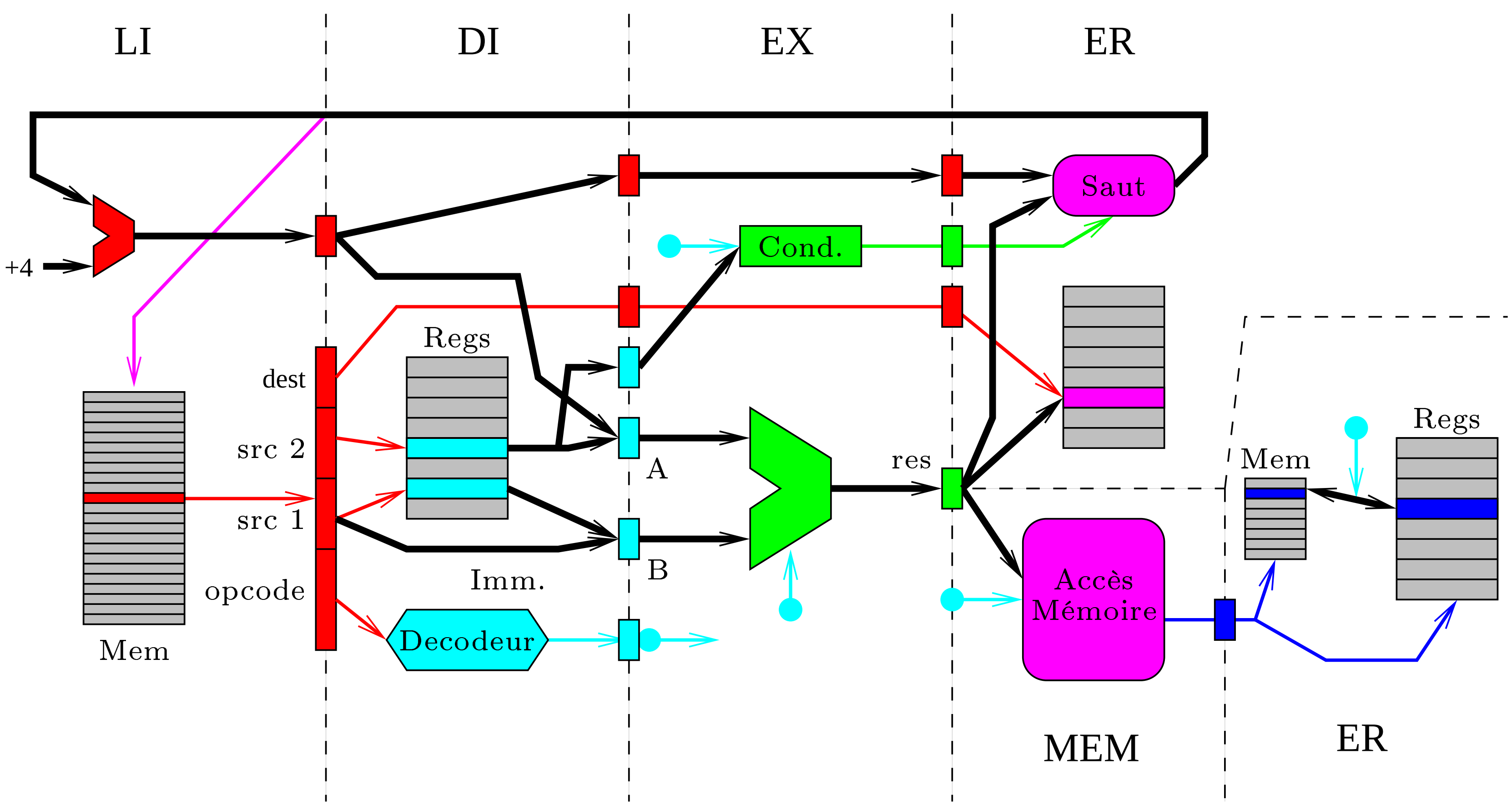
Pipeline et type d'instructions

En réalité, selon le type d'instructions, le pipeline est de longueur différente au sein du même processeur. On considère alors qu'il y a plusieurs pipeline logique (même si différentes instructions partages le même étage physique).

Trois type de pipeline :

1. le pipeline entier (calcul, comparaison...) ;
2. le pipeline flottant ;
3. le pipeline mémoire (chargement, rangement) ;

Les pipelines du processeur



Conséquences

* Problème de latence : la résultat n'est pas disponible pour l'instruction suivante.

  Solution : réordonancement des instructions.

* Problème des sauts : On ne sait pas à la fin du premier cycle quelle est l'instruction suivante.

  Solution : prédiction et déroulage des boucles.