

Architecture des Ordinateurs et Systèmes d'Exploitation

Partie 1 : Architecture des Ordinateurs Pipeline Processeur

Fabrice BOISSIER & Elena KUSHNAREVA
2017/2018

fabrice.boissier@gmail.com
elena.kushnareva@malix.univ-paris1.fr

Pipeline Processeur

- Instruction :

la plus petite unité exécutable par le processeur (« atomique »)

Exemples :

- Déplacer un mot de 8/16/32/64 bits
- Additionner deux valeurs (+, -, x, /, OR, AND, XOR, ...)
- Tester une valeur/Mettre à jour les flags
- Sauter à une adresse précise dans le code (selon état flag)

Pipeline Processeur

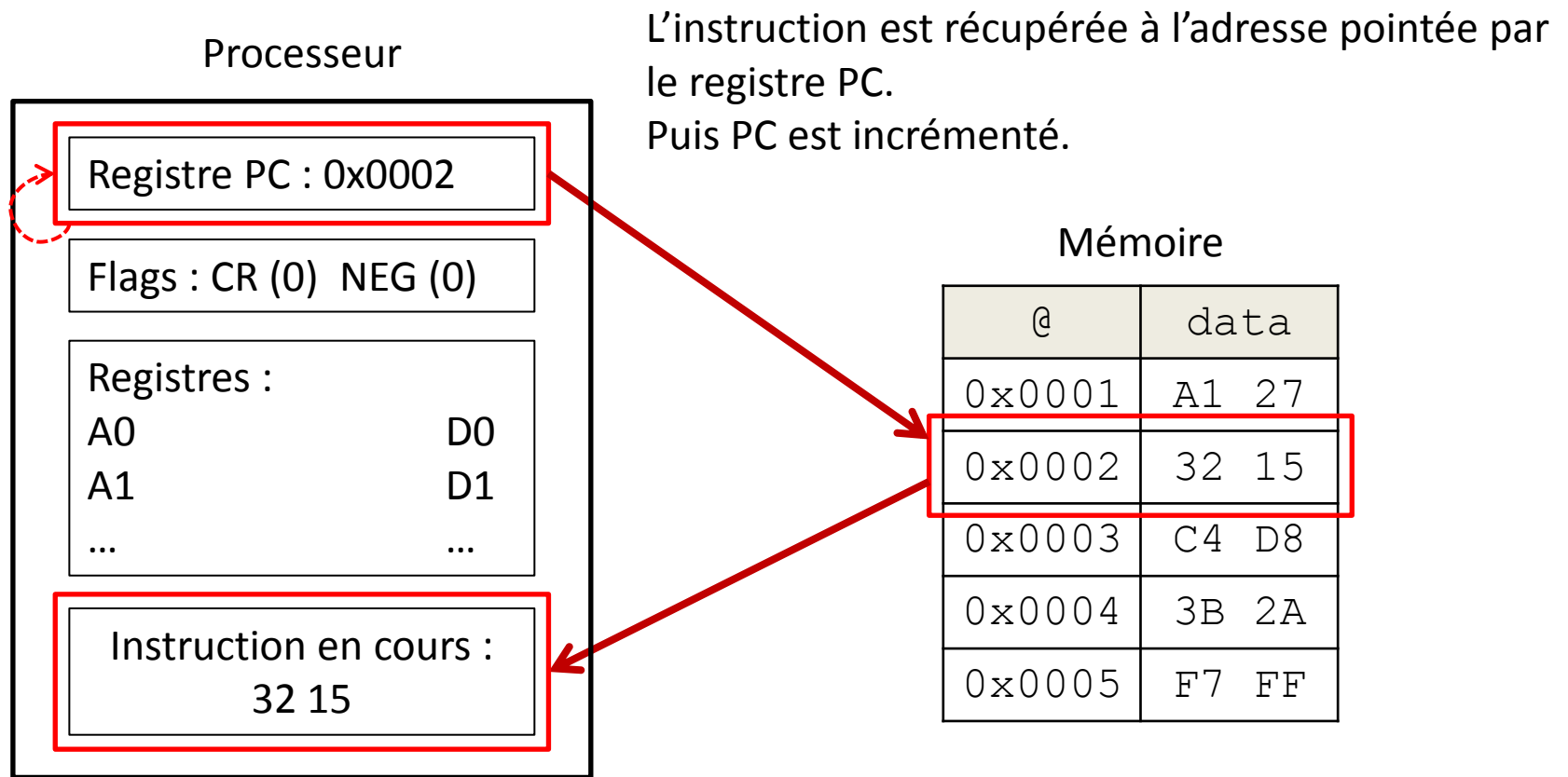
- Pipeline :

5 étapes pour exécuter une instruction

1. Instruction Fetch (récupération de l'instruction en mémoire)
2. Instruction Decode (décodage de l'instruction et des adresses)
3. Execute (récupération des paramètres & exécution de l'instruction)
4. Memory (écriture depuis ou vers la mémoire)
5. Write Back (écriture du résultat dans les registres)

Pipeline Processeur

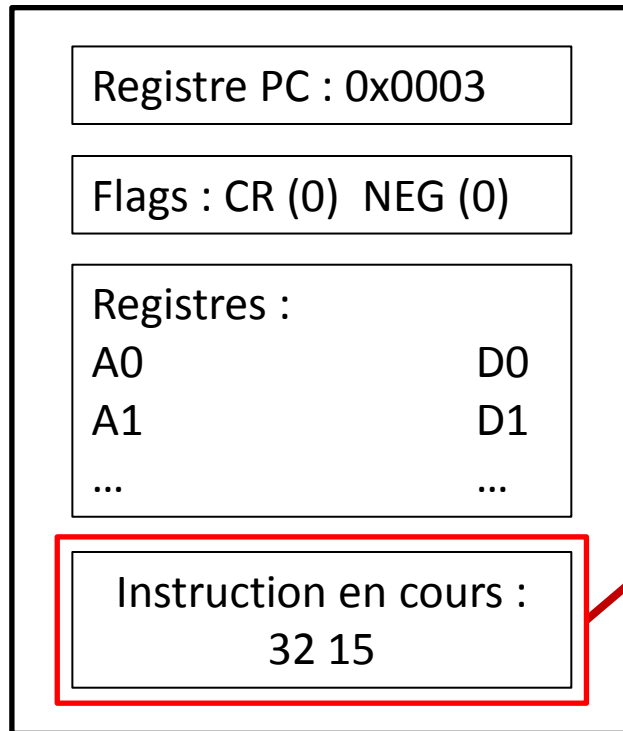
- 1 - Instruction Fetch



Pipeline Processeur

- 2 - Instruction Decode

Processeur



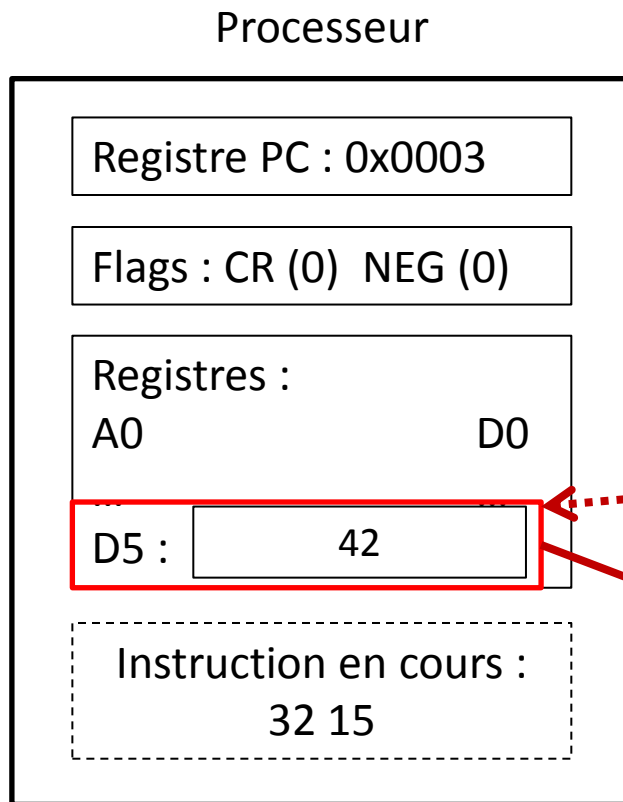
L'instruction est décodée selon un schéma défini par le fabricant du processeur.

32								15									
OP Code				Source						Destination							
0	0	1	1	0	0	1	0	0	0	0	1	0	1	0	1		
0	0	1	1	0	0	1	0	0	0	0	0	0	1	0	1		
ADD				8						D5							

32 15 ⇔ ADD 8, D5

Pipeline Processeur

- 3 - Execute



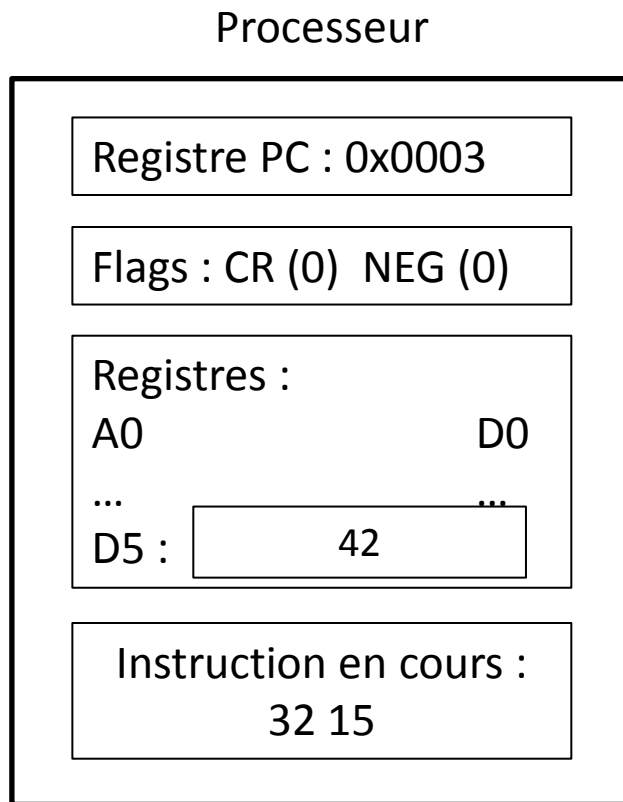
Les paramètres de l'instruction sont récupérés.
Puis l'instruction est exécutée (la zone concernée
du processeur s'active et lit les paramètres).

OP Code	Source	Destination
0 0 1 1	0 0 1 0 0 0	0 1 0 1 0 1
ADD	8	D5

$$42 + 8 = 50$$

Pipeline Processeur

- 4 - Memory



Lecture ou Ecriture en mémoire.
(selon le type de processeur : Execute peut lire en mémoire, et Memory peut écrire le résultat)

50

Mémoire

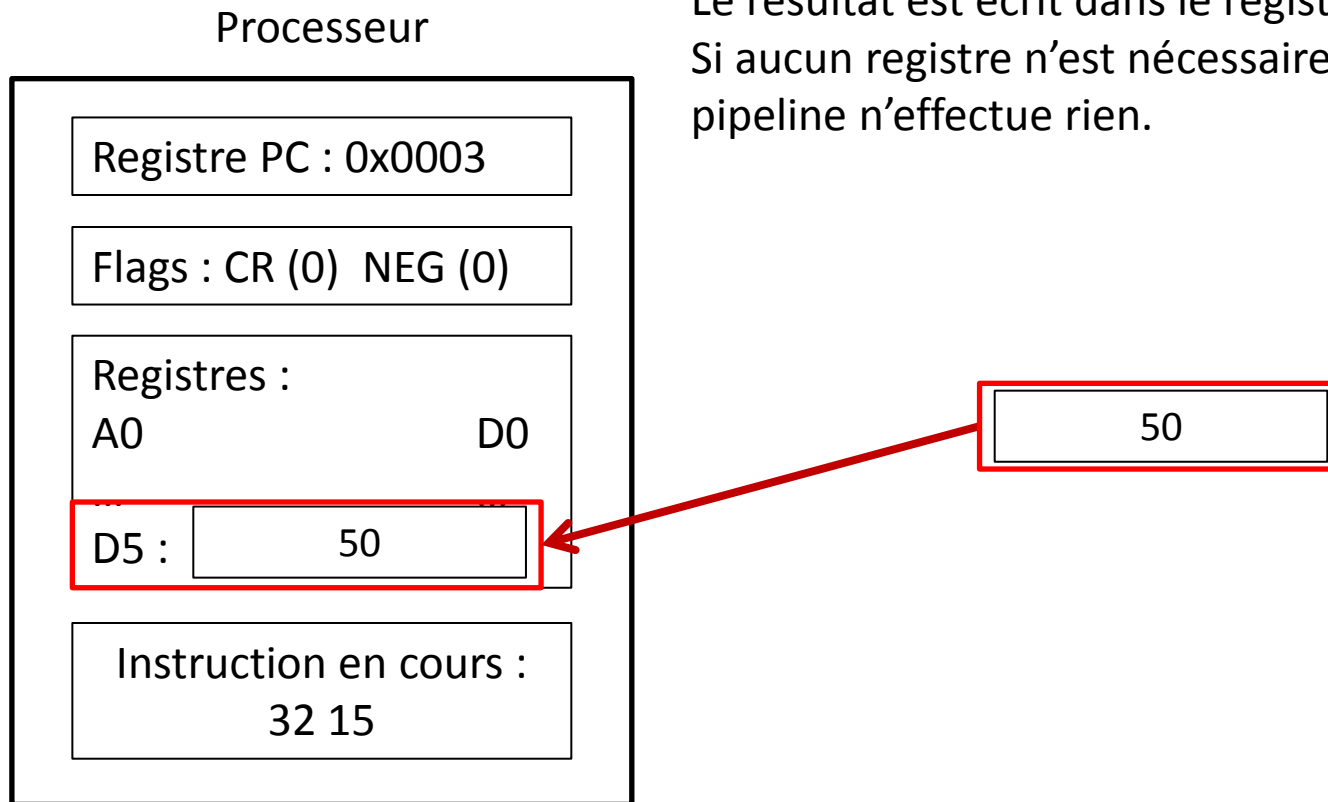
@	data
0x0001	A1 27
0x0002	32 15
0x0003	C4 D8
0x0004	3B 2A
0x0005	F7 FF

Dans l'exemple du ADD, aucune lecture ou écriture n'est faite.

Pipeline Processeur

- 5 – Write Back

Le résultat est écrit dans le registre désigné.
Si aucun registre n'est nécessaire, cet étage du pipeline n'effectue rien.



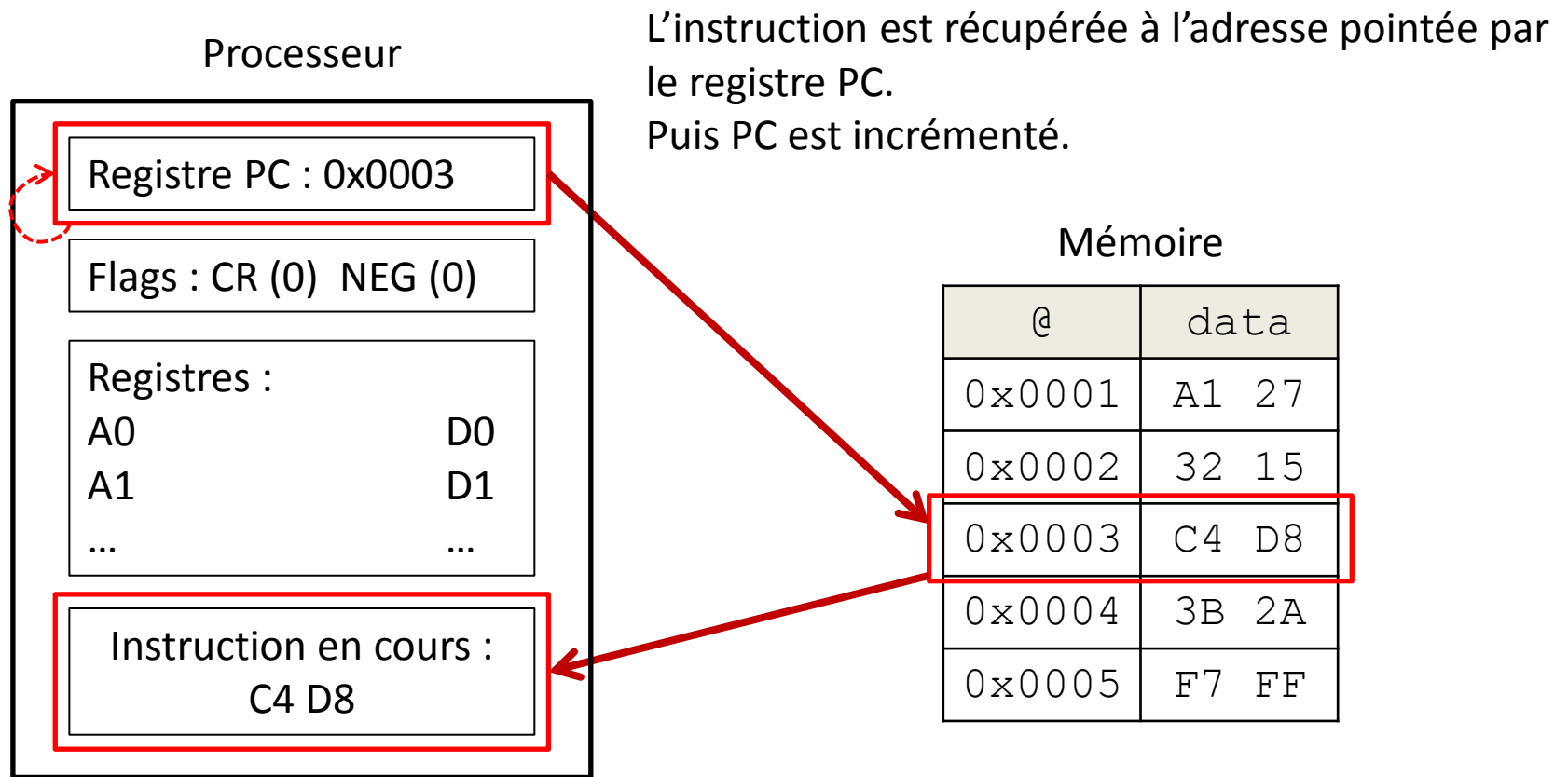
Pipeline Processeur

Fin du cycle !

(si une interruption a été reçue, elle est traitée)

Pipeline Processeur

- 1 - Instruction Fetch



Pipeline Processeur

...et ainsi de suite...

Pipeline Processeur

...indéfiniment ?

Pipeline Processeur

Oui, un processeur « tourne » indéfiniment sauf si :

- Instruction bloquante a été prévue par le fabricant
- On coupe le courant...

Pipeline Processeur

Comment stopper proprement un processeur ?

- Boucle logicielle qui ne fait rien + courant coupé
(NOP + JUMP)
- Interruption émise vers un composant externe qui stoppe le processeur (quartz stoppé, pin de désactivation, ...)


Soyons créatifs...

Architectures Harvard & Von Neumann

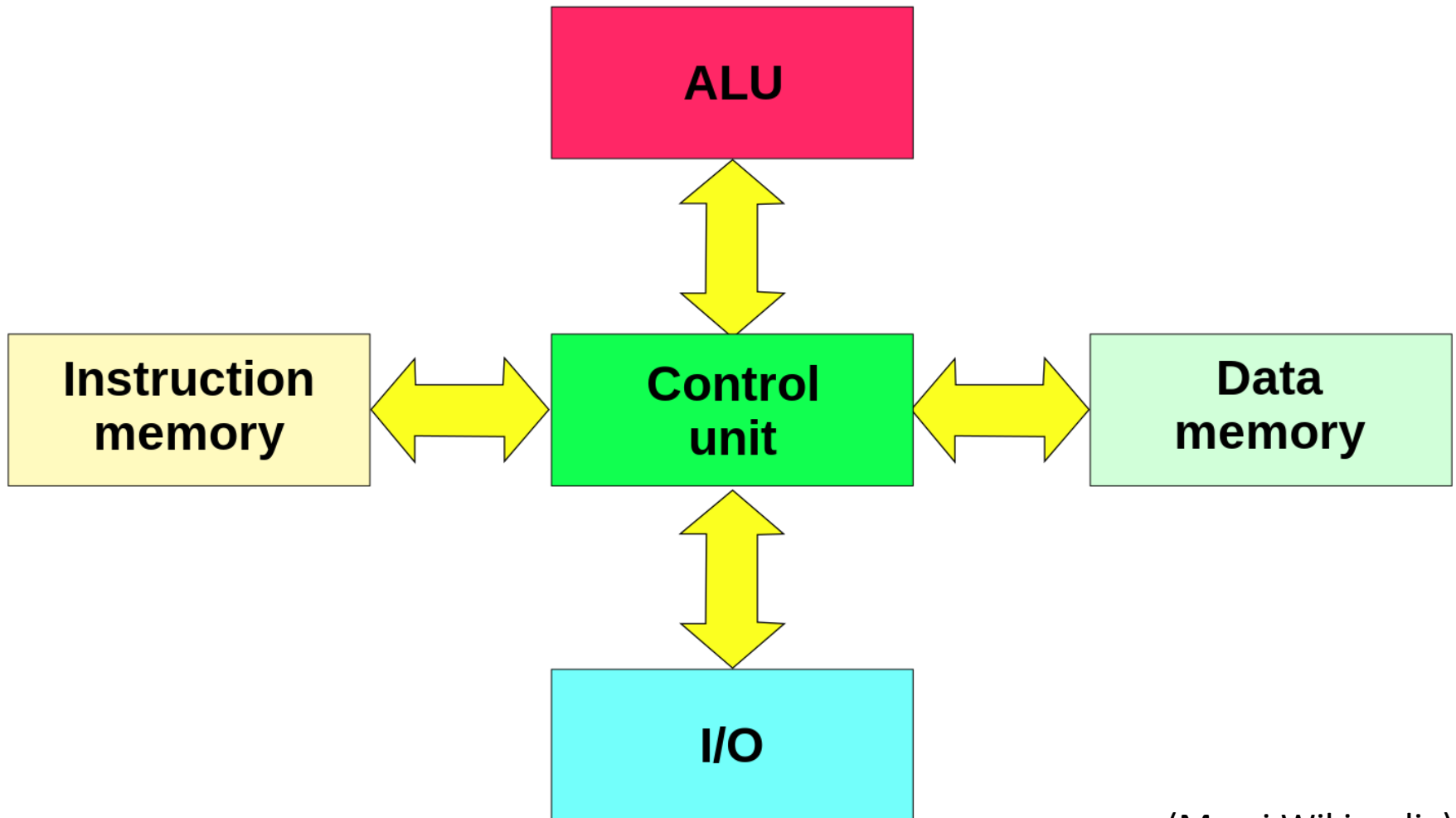
Architecture Harvard

- Séparation de la mémoire :
 - Un bus de données
 - Un bus d'instructions
 - Impossible au bus de données d'accéder à la mémoire instructions (et inversement)
- Possible d'exécuter une instruction ET lire/écrire des données en même temps

Architecture Von Neumann

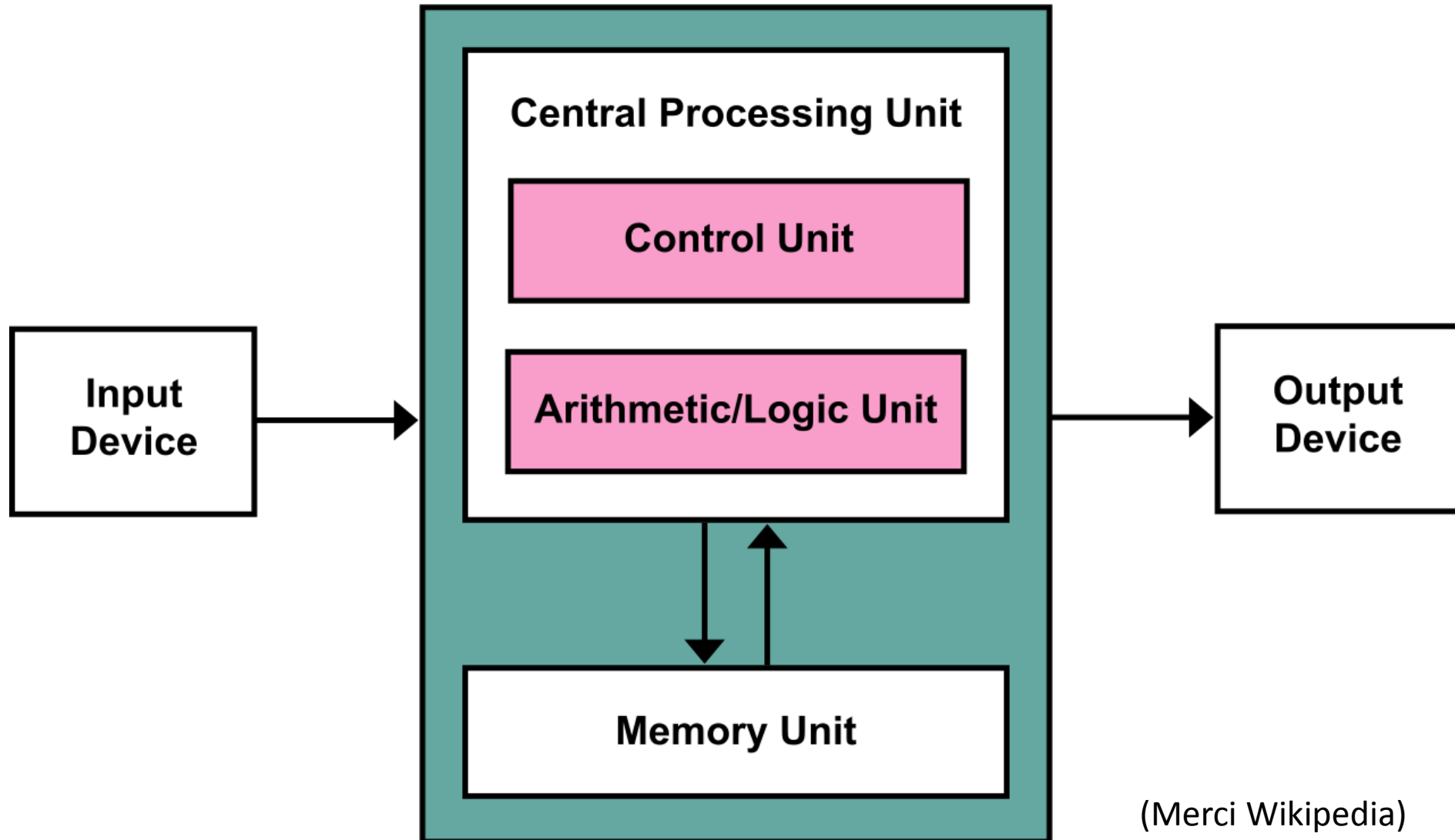
- Des entrées
- Des sorties
- Un processeur (avec registres) relié à de la mémoire
 - Instructions et données stockées au même endroit
- Les accès mémoire pour récupérer les instructions empêchent de lire/écrire des données au même moment

Architecture Harvard



(Merci Wikipedia)

Architecture Von Neumann



(Merci Wikipedia)

Architecture Processeurs

...Aujourd'hui...

...Architecture entre Harvard et Von Neumann...

Jeu d'Instructions CISC

- CISC
 - Complex Instruction Set Computing
 - Instructions « complexes » et longues à effectuer
 - Déplacement vers la mémoire inclus dans les opérations
 - Assembleur plus diversifié
 - Beaucoup d'instructions
 - Réduire au maximum la distance entre le bas niveau et le haut niveau

Jeu d'Instructions RISC

- RISC
 - Reduce Instruction Set Computing
 - Instructions « simples » et rapides à effectuer
 - Déplacement en mémoire nécessite une instruction
 - Assembleur moins diversifié...
 - Le développeur doit découper ses fonctions en instructions très petites... long et parfois difficile
 - Permettre au compilateur d'optimiser au maximum le code
 - Opération plus fines/petites

CISC/RISC

Exemple avec un CISC idéal, et un RISC réaliste :
ajouter deux valeurs se trouvant en mémoire, et les remettre
en mémoire

CISC

```
ADD @42, @67
```

RISC

```
MOVE @42, D0
```

```
MOVE @67, D1
```

```
ADD D0, D1
```

```
MOVE D1, @67
```

CISC/RISC

Exemple avec un CISC réaliste, et un RISC réaliste:
ajouter deux valeurs se trouvant en mémoire, et les remettre
en mémoire

CISC

```
MOVE @42, D0  
ADD D0, @67
```

RISC

```
MOVE @42, D0  
MOVE @67, D1  
ADD D0, D1  
MOVE D1, @67
```

Autres Jeux d'Instructions

- VLIW
 - Very Long Instruction Word
 - Instructions sur 128/256 bits
- EPIC
 - Explicitly Parallel Instruction Computing
- MISC
 - Minimal Instruction Set Computing

Jeux d'Instructions

...Aujourd'hui...

...RISC amélioré...

...voire assembleur/langage machine CISC
traduit en micro-instructions RISC dans le
pipeline...

...quelques tentatives de VLIW...

Autres Types de Processeurs

- DSP (Digital Signal Processor)
 - Traite des signaux (amplificateurs en musique)
 - Agit sur le niveau de tension ou intensité au lieu de bits
- Processeurs Vectoriels
 - Traite des tableaux/matrices très longues
 - Utilisés dans SuperOrdinateurs/SuperCalculateurs et dans les GPU