

Organisation de base d'une machine de von Neumann

Eduardo Sanchez
EPFL

Le logiciel

```
int data = 0x123456;  
int result = 0;  
int mask = 1;  
int count = 0;  
int temp;  
  
while (count < 32) {  
    temp = data & mask;  
    result = result + temp;  
    data = data >> 1;  
    count = count + 1;  
};  
  
/* result = 9 */
```

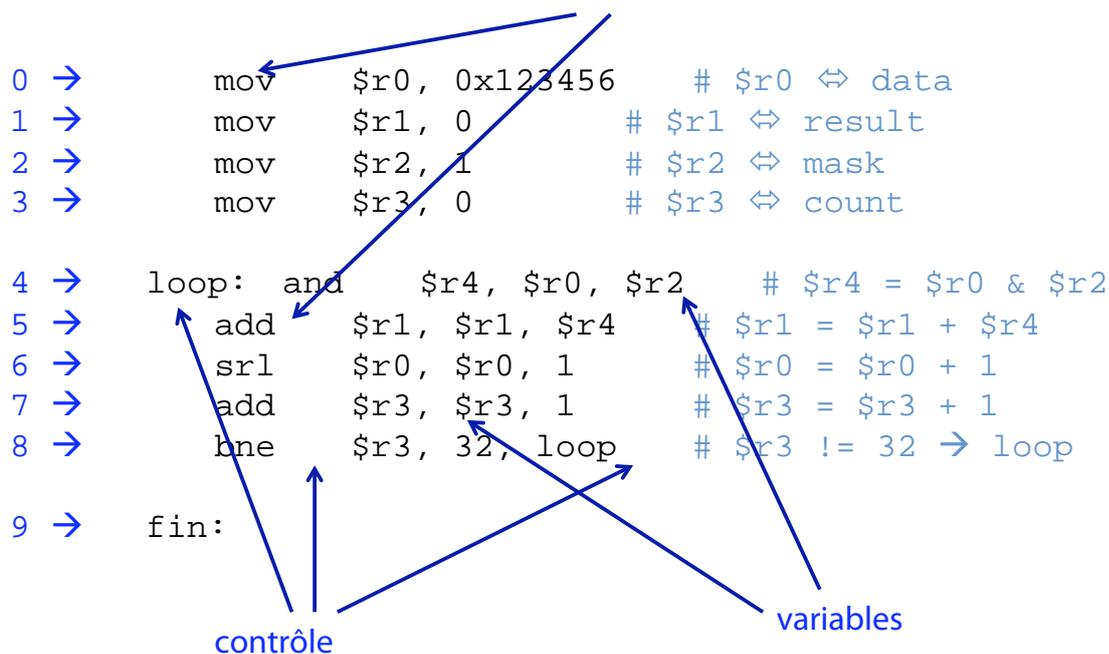
variables

data:	0x123456
result:	0
mask:	1
count:	0
temp:	undef

opérations et affectations

contrôle

opérations et affectations



Eduardo Sanchez

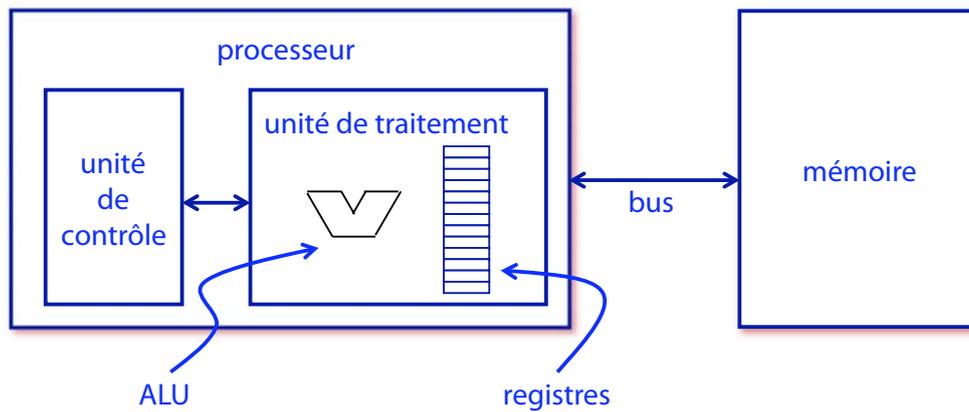
3

Architecture des ordinateurs

- Les données traitées par un ordinateur sont stockées dans sa **mémoire**
- L'élément de l'ordinateur qui réalise les opérations de traitement des données est le **processeur** ou CPU (*Central Processing Unit*)
- Le processeur peut être divisé en deux parties:
 - l'unité de traitement: ensemble d'opérateurs arithmétiques et logiques, groupés autour d'une ou plusieurs ALUs (*Arithmetic and Logic Unit*);
 - l'unité de contrôle: coordonnateur des différentes activités du processeur
- En plus, le processeur possède ses propres unités de stockage d'information, plus rapides que la mémoire, mais moins nombreuses: les **registres**

Eduardo Sanchez

4



- Le transfert des données entre la mémoire et le processeur se fait via un ensemble de lignes d'interconnexion: le bus
- Le processeur est capable de lire ou d'écrire une donnée dans la mémoire: il doit envoyer l'adresse de la donnée et un signal indiquant le type d'opération

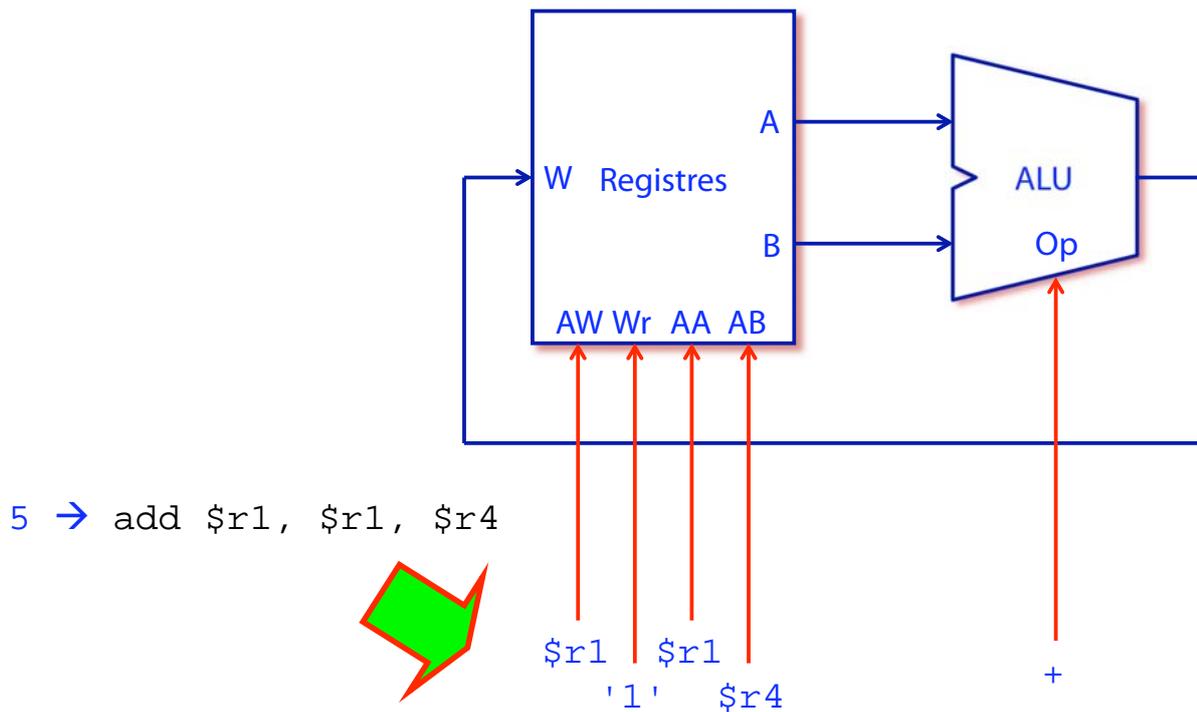
Eduardo Sanchez

5

- L'exécution d'une tâche, aussi simple soit-elle, implique une série d'opérations, réalisées dans l'unité de traitement et coordonnées par l'unité de contrôle, avec des transferts de données entre la mémoire et le processeur
- L'addition de deux nombres, par exemple, pourrait se faire en 5 pas:
 - chercher le premier nombre dans la mémoire et le placer dans un registre du processeur
 - chercher le deuxième nombre dans la mémoire et le placer dans un autre registre
 - activer l'additionneur avec les deux registres précédents comme sources; stocker le résultat dans un registre
 - sauver le résultat dans la mémoire
 - arrêter

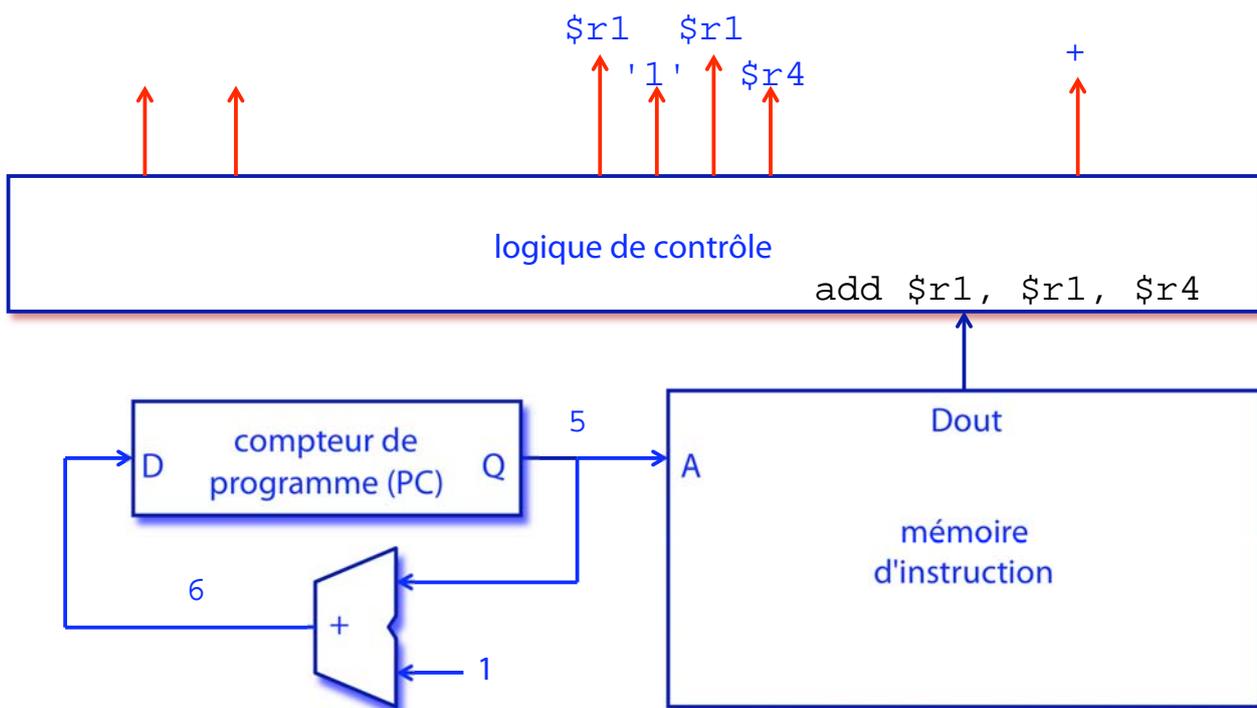
Eduardo Sanchez

6



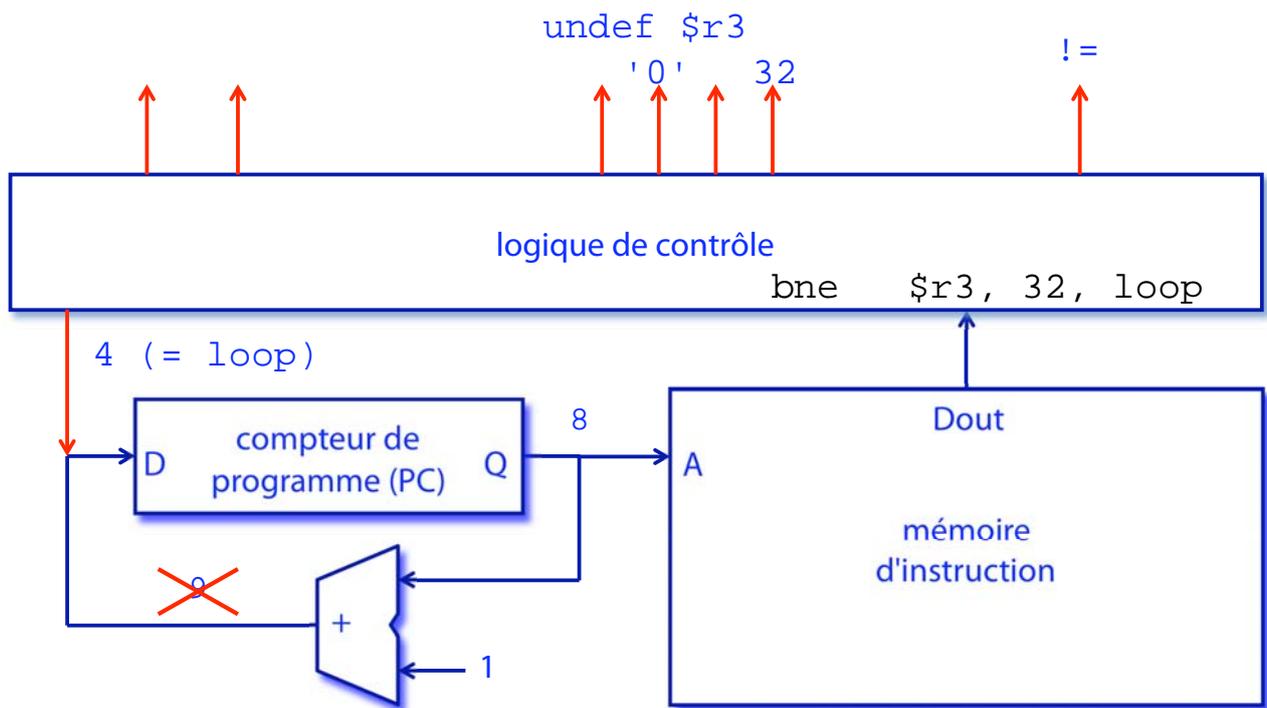
Eduardo Sanchez

7



Eduardo Sanchez

8



Eduardo Sanchez

9

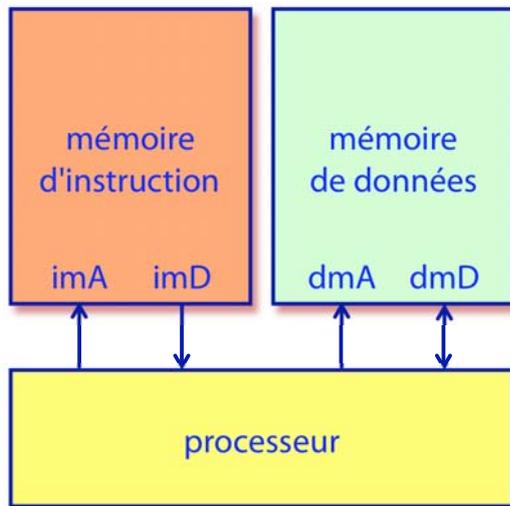
Architecture de von Neumann

- Dans les premiers ordinateurs, les différents pas nécessaires à l'exécution d'une tâche, le **programme**, étaient directement câblés dans l'unité de contrôle
- Un grand progrès a été effectué lorsque le programme, comme les données, a été codé et stocké dans la mémoire principale: c'est l'architecture appelée de von Neumann
- La fonction de l'unité de contrôle est de lire le programme de la mémoire, décoder les instructions et commander leur exécution
- Un changement de programme se fait maintenant par une simple réécriture de la mémoire

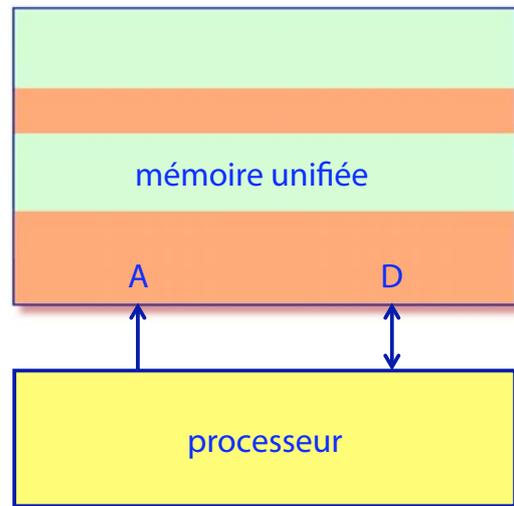
Eduardo Sanchez

10

architecture de Harvard

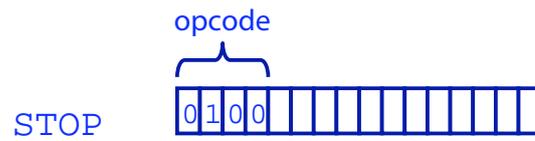
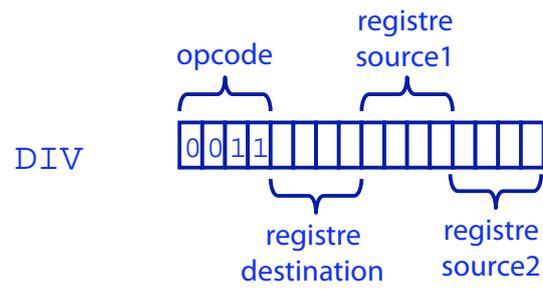


architecture de von Neumann



Langage machine

- Les processeurs doivent reconnaître des instructions codifiées sous la forme de groupes de bits
- L'ensemble des instructions reconnues par un processeur et son système de codage forment ce qu'on appelle le langage machine du processeur
- Il y a deux grandes familles de processeurs, selon la complexité de son langage machine:
 - processeurs **CISC** (*Complex Instruction Set Computer*). Exemple: Pentium
 - processeurs **RISC** (*Reduced Instruction Set Computer*). Exemples: Sparc, PowerPC, MIPS
- Il y a trois grands types d'instruction:
 - transfert de données
 - opérations arithmétiques/logiques
 - contrôle



- Notre programme codé serait alors:

1	LOAD	R0,M[0]	0000
2	LOAD	R1,M[1]	0101
3	JUMP	zéro,6	2006
4	DIV	R2,R0,R1	3201
5	STORE	M[2],R2	1202
6	STOP		4000

Exécution du programme

- Pour exécuter un programme, l'unité de contrôle du processeur doit lire chaque instruction, la décoder et ensuite l'exécuter. C'est le cycle exécuté sans arrêt par un processeur: *fetch-decode-execute*
- Pour cela, l'unité de contrôle dispose de deux registres spécialisés: le compteur de programme (PC) et le registre d'instruction (IR)
- Le PC contient l'adresse de la prochaine instruction à exécuter. Le IR contient le code de l'instruction en exécution
- Pour notre exemple précédent, le *fetch* implique la lecture de deux mots de la mémoire. Toute instruction, sauf le JUMP, fait donc une incrémentation par deux du PC

Exercice

- Supposez que vous voulez multiplier deux variables **a** et **b**, stockées dans les positions de mémoire **M[20]** et **M[21]**, respectivement, pour affecter cette valeur à la variable **toto**, stockée à la position de mémoire **M[40]**
- C'est-à-dire, vous voulez effectuer l'opération:
$$\text{toto} = \text{a} * \text{b}$$
ou:
$$\text{M}[40] = \text{M}[20] * \text{M}[21]$$

- Le processeur possède 8 registres (**R0...R7**). Le registre **R0** contient toujours la valeur 0
- Les instructions du langage machine du processeur sont:
 - **LOAD Rd, M[adr]** $Rd \leftarrow M[adr]$
 - **STORE M[adr], Rs** $M[adr] \leftarrow Rs$
 - **ADD Rd, Rs1, Rs2** $Rd \leftarrow Rs1 + Rs2$
 - **SUB Rd, Rs1, Rs2** $Rd \leftarrow Rs1 - Rs2$
 - **DEC R** $R \leftarrow R - 1$
 - **JUMP zero, adr** si zero alors sauter à adr
- Avant d'écrire le programme, nous devons trouver un algorithme réalisant la tâche voulue
- Un algorithme possible serait:

