

Architectures matérielles — Découvrir Von Neumann

1 Architecture Von Neumann

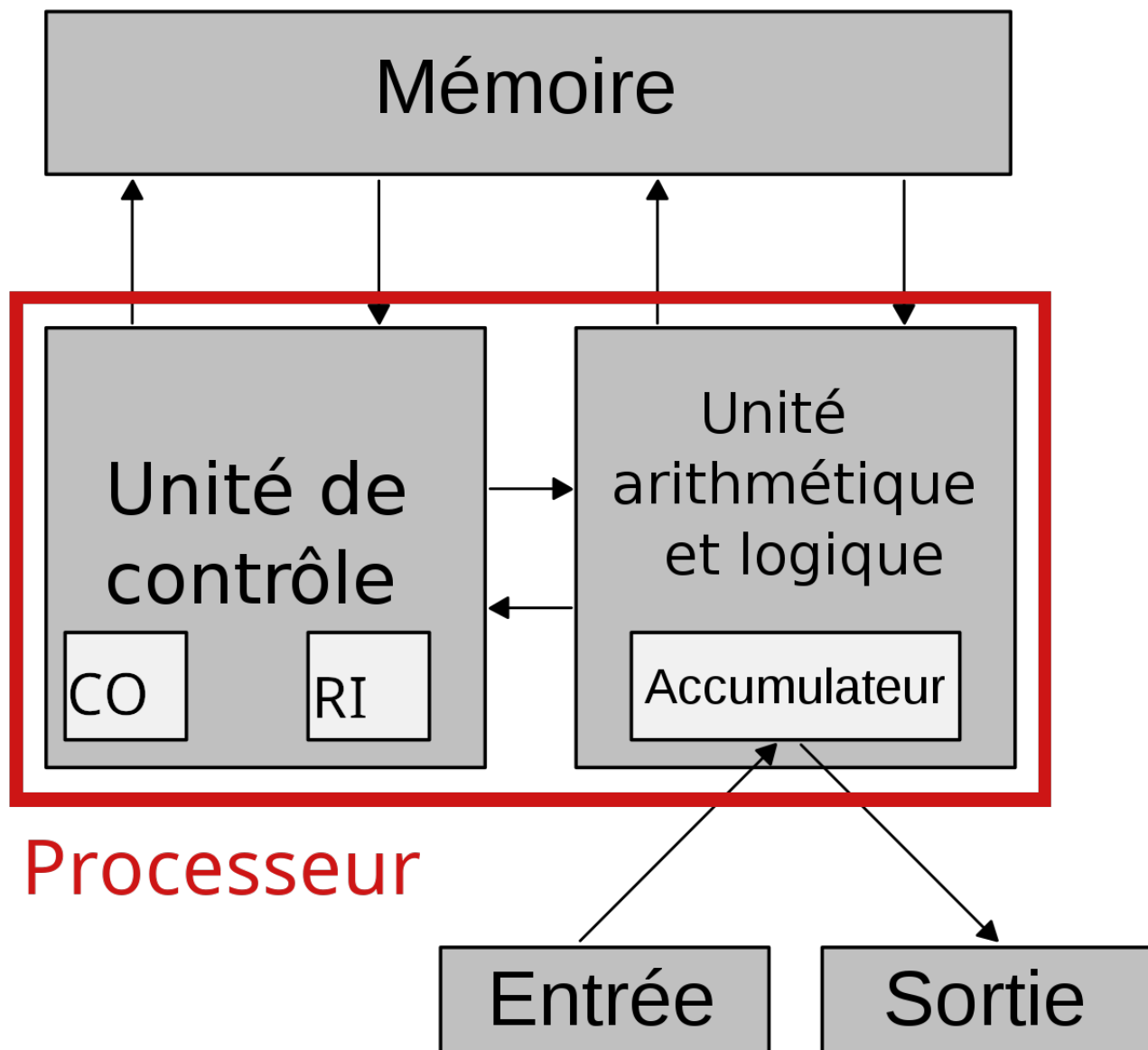


FIGURE 1 – Architecture Von Neumann. Les registres sont sur fond clair

2 La mémoire vive (ou RAM)

Il s'agit d'un tableau de bits (64 dans les ordinateurs récents).

Adresse (ici sur 4 bits)	valeur (ici sur 8 bits)
0000	0110 1010
0001	1111 1111
0010	1111 0000
...	...

3 Les Registres

Le processeur ne traite pas directement les données en RAM, il utilise pour cela des registres (de la mémoire directement incluse dans le processeur).

Parmi ces registres, nous allons utiliser les trois suivants :

RI le registre d'instruction ; il contient l'instruction en cours ;

CO le compteur ordinal ; il contient l'adresse de la prochaine instruction à exécuter ;

ACC l'accumulateur ; registre de l'UAL chargé de stocker une des opérandes avant l'exécution puis le résultat de l'opération.

4 Les instructions

Un processeur est capable d'exécuter des instructions.

DÉFINITION 1

Le **langage machine** est un ensemble d'instructions binaire que le processeur peut suivre.

Un **langage assembleur** est la version lisible (par les humain-e-s) de ce langage binaire.

Un langage assembleur est souvent spécifique un type de processeur.

Instruction machine	Instruction Assembleur	Effet
0000 aaaa	LD A	Lit le contenu de la cellule A ($ACC \leftarrow A$)
0001 aaaa	STO A	Enregistre dans la cellule A ($A \leftarrow ACC$)
0010 aaaa	ADD N	Ajoute le contenu de la cellule A ($ACC \leftarrow ACC + N$)
0011 aaaa	SUB N	Soustrait le contenu de la cellule A ($ACC \leftarrow ACC - N$)
0100 aaaa	JMP A	Saute à l'adresse A ($CO \leftarrow A$)
0101 aaaa	JMPZ A	Saute à l'adresse A ($CO \leftarrow A$) si $ACC = 0$
0110 aaaa	JMPP A	Saute à l'adresse A ($CO \leftarrow A$) si $ACC > 0$
0111 aaaa	JMPN A	Saute à l'adresse A ($CO \leftarrow A$) si $ACC < 0$

FIGURE 2 – Un exemple de langage machine, dont une opération est numérotée sur 4 bits et un paramètre sur 4 bits également.

#A signifie, la valeur de la case d'adresse A.

Voici un exemple d'état de la mémoire et des registres. Que va faire le processeur ainsi démarré ?

Adresse	Contenu	Étape	RI	CO	Acc
0	25	Initiale		8	
1	14	1			
...	...	2			
8	LD 0	3			
9	SUB 1	4			
10	STO 2				
11	END				

4.1 Exercice

Voici un exemple d'état de la mémoire et des registres. Complétez le tableau de droite pour simuler les actions du processeur et essayez de décrire ce que fait ce programme.

Adresse	Contenu	Étape	RI	CO	Acc
0	3	Initiale		8	
1	20	1			
...	...	2			
8	LD 0	3			
9	ADD 10	4			
10	STO 2				
11	END				

4.2 Exercice

Voici un exemple d'état de la mémoire et des registres. Complétez le tableau de droite pour simuler les actions du processeur et essayez de décrire ce que fait ce programme.

Adresse	Contenu	Étape	RI	CO	Acc
0	3	Initiale		8	
1	20	1			
...	...	2			
8	LD 0	3			
9	ADD 10	4			
10	JMP 8	5			
11	END	6			
		7			
		8			

4.3 Exercice

Bonus. Écrire un programme en langage assembleur qui récupère la valeur en RAM à l'adresse 0, à ajoute 1 et stock le résultat à l'adresse 0.

4.4 Exercice

Bonus. Voici un exemple d'état de la mémoire et des registres. Complétez le tableau de droite pour simuler les actions du processeur et essayez de décrire ce que fait ce programme.

Adresse	Contenu	Étape	RI	CO	Acc
0	3	Initiale		8	
1	20	1			
...	...	2			
8	LD 0	3			
9	SUB 1	4			
10	JUMPZ 13	5			
11	JUMP 9	6			
12	JMP 8	7			
13	STO 0	8			
14	END				

4.5 Exercice

Bonus. Écrire un programme en langage assembleur qui récupère la valeur en RAM à l'adresse 0, à ajoute 1 et stock le résultat à l'adresse 0.

4.6 Exercice

Bonus. Écrire un programme en langage assembleur qui récupère la valeur en RAM à l'adresse 0, à ajoute 1 si elle est négative (et retranche 1 si elle est positive) et stock le résultat à l'adresse 0.

4.7 Exercice

Bonus. Écrire un programme en langage assembleur qui récupère la valeur en RAM à l'adresse 0, à ajoute 1 si elle est négative (et retranche 1 si elle est positive) et stock le résultat à l'adresse 0, puis recommence jusqu'à ce que la valeur soit nulle.