

# Architecture des ordinateurs et systèmes d'exploitation

## TD 4: Une architecture simple

Arnaud Giersch

Benoît Meister

Nicolas Passat

On considère l'architecture 32 bits simplifiée composée des éléments suivants :

- Une UAL (Unité Arithmétique et Logique).
- Les registres :
  - $R_1, R_2, \dots, R_{32}$  (registres de données) ;
  - $R_{op1}$  et  $R_{op2}$  (registres opérandes de l'UAL) ;
  - $R_{res}$  (registre résultat de l'UAL) ;
  - Flags (registre des flags, composé de 4 bits : Z, C, O, S) ;
  - IP (registre adresse de l'instruction courante) ;
  - RI (registre contenant l'instruction courante) ;
  - RTA (registre tampon adresse, contenant l'adresse d'un mot à lire/écrire en mémoire) ;
  - RTD (registre tampon données, contenant un mot à lire/écrire en mémoire).
- Deux types de signaux :
  - R/W (signal déclenchant la lecture/écriture en mémoire) ;
  - E (signal activant l'UAL).
- Une mémoire pour les données et les instructions (mots de 4 octets).

Un jeu d'instructions réduit est associé à cette architecture :

Instruction	Syntaxe	Signification
Addition	add $R_x, R_y, R_z$	$R_x \leftarrow R_y + R_z$
Soustraction	sub $R_x, R_y, R_z$	$R_x \leftarrow R_y - R_z$
Et	and $R_x, R_y, R_z$	$R_x \leftarrow R_y \text{ and } R_z$
Ou	or $R_x, R_y, R_z$	$R_x \leftarrow R_y \text{ or } R_z$
Chargement	load $R_x, R$ ou Adresse	$R_x \leftarrow \text{Mem}(R \text{ ou Adresse})$
Enregistrement	store $R_x, R$ ou Adresse	$\text{Mem}(R \text{ ou Adresse}) \leftarrow R_x$
Comparaison	cmp $R_x, R_y$	$Z \leftarrow 0 \text{ si } R_x = R_y, 1 \text{ sinon}$
Branchement conditionnel	bz $R$ ou Adresse	Branchement si $Z = 0$
Branchement	ba $R$ ou Adresse	Branchement dans tous les cas

Remarques :

- Les instructions sont toutes sur 4 octets ;
- $R_\alpha$  correspond à un des registres  $R_1, \dots, R_{32}$  ;
- Adresse correspond à une adresse sur 16 bits. Comme les adresses doivent être sur 32 bits, les 16 bits de poids fort sont alors mis à 0.
- Comme les mots mémoire sont codés sur 4 octets, toutes les adresses de mots doivent être des multiples de 4.

1. Donnez la significations des flags Z, C, O, S.
2. Quelle est l'adresse la plus grande qui peut être utilisée ?
3. Quelle est la taille mémoire utilisable (i.e. adressable), en bits ?
4. Quel est le plus grand entier non signé (resp. signé) qui peut être manipulé ?
5. Pour chaque type d'instruction, donnez les registres concernés.
6. Donnez les étapes nécessaires pour chacune des instructions suivantes :

- load R<sub>5</sub>, 1016
- store R<sub>2</sub>, R<sub>4</sub>
- add R<sub>4</sub>, R<sub>1</sub>, R<sub>12</sub>
- sub R<sub>6</sub>, R<sub>6</sub>, R<sub>5</sub>
- and R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>
- or R<sub>1</sub>, R<sub>8</sub>, R<sub>11</sub>
- cmp R<sub>4</sub>, R<sub>6</sub>
- bz 4096
- ba 4000

7. On considère l'état mémoire suivant :

Adresse	Instructions	Adresse	Données
100	load R <sub>4</sub> , 0	0	1
104	load R <sub>1</sub> , 12	4	10
108	load R <sub>2</sub> , 4	8	5
112	load R <sub>3</sub> , 8	12	0
116	cmp R <sub>1</sub> , R <sub>2</sub>		
120	bz 148		
124	add R <sub>1</sub> , R <sub>1</sub> , R <sub>4</sub>		
128	add R <sub>5</sub> , R <sub>2</sub> , R <sub>5</sub>		
132	sub R <sub>6</sub> , R <sub>6</sub> , R <sub>3</sub>		
136	cmp R <sub>1</sub> , R <sub>2</sub>		
140	bz 148		
144	ba 124		
148	store R <sub>5</sub> , 4		
152	store R <sub>6</sub> , 8		

En supposant que tous les registres R<sub>1</sub>, ..., R<sub>32</sub> sont initialement à 0, donnez le contenu de ces registres ainsi que le contenu de la mémoire après exécution.