Département informatique
2ème année Licence Informatique
Module : Architecture des Ordinateur

Série TD n° 5.1

Exercice N°1 : (Modèles d'accès à la mémoire)

On veut comparer l'efficacité mémoire de quatre styles différents de jeux d'instructions. Les styles sont:

a. **Accumulateur** : toutes les opérations interviennent entre un registre unique et une case mémoire.
b. **Mémoire – Mémoire** : les trois opérandes de chaque instruction sont en mémoire.
c. **Pile :** toutes les opérations interviennent sur le sommet d'une pile. Seuls l'empilement et le dépilement accèdent à la mémoire et toutes les autres instructions prennent leurs opérandes dans la pile et les remplacent par le résultat. L'implémentation utilise une pile pour les deux entrées supérieures. Les accès qui utilisent d'autres positions de la pile sont les références mémoires.
d. **Chargement – Rangement** : toutes les opérations interviennent dans les registres et les instructions registre à registre ont trois opérandes par instruction. Il y a 16 registres généraux et les spécificateurs de registres ont 4 bits.

Pour mesurer l'efficacité mémoire, les hypothèses sur les quatre jeux d'instructions sont les suivantes :

- Le code opération est toujours sur un octet.
- Toutes les adresses mémoires ont deux octets.
- Tous les opérandes donnés ont quatre octets.
- La longueur de toutes les instructions est un nombre entier d'octet.
- Les variables A, B, C et D sont initialement en mémoire.

Pour chaque architecture, proposer des mnémoniques de langage assembleur pour la séquence d'instructions en langage évolué.

A=B+C
B=B+C
D=B-A

Puis écrire le meilleur code assembleur équivalent

1. Calculer le nombre d'octets d'instructions lus et le nombre d'octets de données mémoire transféré.
2. Quelle architecture est la meilleure en fonction de la taille du code ?
3. Quelle architecture est la meilleure du point de vue bande passante mémoire totale nécessaire (code + données) ?

Exercice n°2 : **(Performance des processeurs)**

Voici la liste des instructions d'un ordinateur fictif :

04 instructions UAL	**02** instructions de chargement	**01** instruction de rangement	**01** instruction de branchement
▪ ADD Rd,Rs ▪ SUB Rd,Rs ▪ MUL Rd, Rs ▪ DIV Rd,Rs	▪ LOAD Rd, [Rs] ▪ LOAD Rd, Immédiat	▪ STORE [Rd], Rs	▪ BEQZ Rs, Immédiat

L'ordinateur possède 8 registres 32 bits et les valeurs immédiates sont encodées sur 32 bits.

1. Dans un premier temps, on utilise **un seul format** d'instruction de longueur fixe et à champ fixe. Énumérez les différents champs d'une instruction ainsi que le nombre de bits accordés à chaque champ. Quelle est la longueur totale d'une instruction ?

2. Afin de réduire la longueur des programmes, on utilise **deux formats** d'instructions de longueur fixe à champ fixe. Décrivez ces deux formats.
3. Si on suppose qu'un programme typique utilise une proportion égale de chaque instruction, combien vaut le rapport Longueur du programme avec 2 formats / Longueur du programme avec 1 format ?

Exercice n°3 : (Performance des processeurs)

L'analyse de l'exécution d'un programme sur une architecture séquentielle donne les résultats suivants :

Catégorie	Pourcentage	Cycles d'exécution
Contrôle	10%	3
Arithmétique et logique	50%	3
Chargement/rangement	40%	3,5

La répartition des chargements et rangements, selon les différents modes d'adressage est la Suivante :

Mode d'adressage	Pourcentage	Cycles d'exécution
Déplacement	40%	3
Indexé	35%	3
Indirect	25%	5

On modifie le compilateur pour qu'il génère un programme qui n'utilise pas le mode d'adressage indirect. Tous les cas de chargements/rangements utilisant ce mode seront remplacés par deux instructions de chargements/rangements utilisant les modes déplacement ou indexé.

Question :

- Calculez le gain ou la perte de performance par rapport à l'ancienne version, on utilisant la loi Amdahl [**Performance =Temps exécution ancien/ Temps exécution nouveau] ?** Commentez **?**

Tutorial Series No. 5.1

Exercise No. 1: (Memory Access Models)

We want to compare the memory efficiency of four different instruction set styles. The styles are:

a. **Accumulator**: All operations involve a single register and a memory location.
b. **Memory-to-Memory**: The three operands of each instruction are in memory.
c. **Stack**: All operations take place at the top of a stack. Only push and pop operations access memory, and all other instructions take their operands from the stack and replace them with the result. The implementation uses a stack for the two top inputs. Accesses that use other positions in the stack are considered memory references.
d. **Load-Store**: All operations occur in registers, and register-to-register instructions have three operands per instruction. There are 16 general-purpose registers, and the register specifiers are 4 bits.

To measure memory efficiency, the assumptions for the four instruction sets are as follows:

- The opcode is always 1 byte.
- All memory addresses are 2 bytes.
- All operands are 4 bytes.
- The length of all instructions is an integer number of bytes.
- The variables A, B, C, and D are initially in memory.

1. For each architecture, propose assembly language mnemonics for the following sequence of high-level instructions:

   A = B + C
   B = B + C
   D = B - A

   Then, write the best equivalent assembly code.
2. Calculate the number of instruction bytes read and the number of memory data bytes transferred.
3. Which architecture is the best in terms of code size?
4. Which architecture is the best in terms of total memory bandwidth required (code + data)?

Exercise No. 2: (Processor Performance)

Here is the list of instructions for a fictional computer:

04 ALU instructions	**02** Load instructions	**01** Store instruction	**01** Branchment instruction
▪ ADD Rd,Rs ▪ SUB Rd,Rs ▪ MUL Rd, Rs ▪ DIV Rd,Rs	▪ LOAD Rd, [Rs] ▪ LOAD Rd, Immédiat	▪ STORE [Rd], Rs	▪ BEQZ Rs, Immédiat

The computer has 8 32-bit registers, and immediate values are encoded on 32 bits.

1. **In the first case, a single fixed-length instruction format with fixed fields is used.** Enumerate the different fields of an instruction and the number of bits allocated to each field. What is the total length of an instruction?
2. **To reduce program length, two fixed-length instruction formats with fixed fields are used.**
   Describe these two formats.
3. **If we assume that a typical program uses an equal proportion of each instruction, what is the ratio of the program length with 2 formats to the program length with 1 format?**

Exercise No. 3: (Processor Performance)

The analysis of the execution of a program on a sequential architecture yields the following results:

Category	**Percentage**	**Execution Cycles**
Control	10%	3
Arithmétique et logique	50%	3
Load/Store	40%	3,5

The distribution of loads and stores, according to the different addressing modes, is as follows:

Addressing Modes	**Percentage**	**Execution Cycles**
Displacement	40%	3
Indexed	35%	3
Indirect	25%	5

We modify the compiler so that it generates a program that does not use the indirect addressing mode. All load/store operations using this mode will be replaced by two load/store instructions using the displacement or indexed modes.

Question:

1. Calculate the performance gain or loss compared to the old version, using **Amdahl's Law** [Performance = Old Execution Time / New Execution Time].
2. Comment on the result.