TD1
Archeticture des ordinateurs

Rappel de représentation des nombres dans une machine

Soit les trois types suivant :

- `int` : les nombres entiers signés représentés en complément à 2 sur 32 bits.
- `float` : les nombres réels représentés en IEEE754 simple précision sur 32 bits.
- `double` : les nombres réels représentés en IEEE754 double précision sur 64 bits.

Etant donné le code C suivant :

```
int A=11 , B=145 , C;
float X= 53.75;
double Y= -13.5;
...
      C = A+B ;
```

1. Représenter en binaire les variables `A,B,C,X,Y`
2. Abréger la représentation des variables en Hexadécimal.

A	00000000 00000000 00000000 00001011	0x 00 00 00 0B
B	00000000 00000000 00000000 10010001	0x00000091
C	00000000 00000000 00000000 00000000	0x 00 00 00 00
X	01000010010101110000000000000000	0x 42570000
Y	11000000 00101011 00000000 00000000 00000000 00000000 00000000 00000000	0x C02B000000000000

3. Donner les intervalles des valeurs représentables pour les deux types : `int`

   L'intervalle de `int` est [-2 147 483 648 , 2 147 483 647]

Processeur / Mémoire centrale

4. Expliquer l'exécution de l'instruction `C = A+B` en prenant en compte les points suivants : (1) emplacement de stockage de l'instruction, (2) emplacement de stockage

des valeurs des variables, (3) transfert d'instructions et de valeurs entre mémoire / processeur, (4) et la manière de faire l'addition utilisant les registres.

(1) emplacement de stockage de l'instruction

Il faut rappeler que la mémoire centrale est divisée en plusieurs parties, comme la partie dédie au stockage des données et des instructions du système d'exploitation.

Dans notre contexte, on s'intéresse aux deux parties : (1) partie de stockage des instructions (les programmes eux-mêmes) (2) et la partie de stockage des données des programmes.

En général, les programmes (les instructions) sont stockés dans la partie instruction dans la mémoire centrale (appelée .text dans l'architecture MIPS). Donc, si on suppose que dans un moment donné, seulement notre programme qui est en état d'exécution, alors, l'instruction `C = A+B` est placé dans la position $d+(n-1)$, où d est l'adresse de la première position de la partie instruction, et n est le numéro de l'instruction `C = A+B` dans notre programme ($n^{ième}$ instruction). Par exemple, si notre instruction occupe la première position dans notre programme, elle va être stockée dans la position d dans la mémoire.

En réalité, l'unité de stockage des instructions est égale à 32 bits. Sachant que, dans l'architecture MIPS, les emplacements mémoire qui sont adressés ont une taille de 8 bits (octets). Il faudra donc utiliser 4 octets pour le stockage d'une instruction. Donc, dans ce cas d'architecture, l'instruction sera placée dans l'octet $d+\mathbf{4}(n-1)$.

(2) emplacement de stockage des valeurs des variables

Les valeurs des variables utilisées par le programme sont stockés dans la partie données dans la mémoire centrale (appelée .data dans l'architecture MIPS).
Tout est dépend du type de la variable utilisée, par exemple pour le type int, il faut avoir 4 octets (32 bits). Par contre pour un char (code ASCII), il suffit d'utiliser 8 bits.
Néanmoins, dans le cas du MIPS, l'unité de représentation de données ou d'instructions occupe **au minimum** 32 bits. Même si la donnée nécessite moins d'espace.

(3) transfert d'instructions et de valeurs entre mémoire / processeur, (4) et la manière de faire l'addition utilisant les registres.

a. Le processeur charge (transfert) l'instruction `C = A+B` à partir de la mémoire et la mettre sur processeur sur un registre de 32 bits appelé registre d'instructions (IR en MIPS).
b. Le processeur commence à analyser l'instruction (UCC) afin de commencer son exécution.
    i. Il transfert les valeurs de A et B à partir de la mémoire, et les mettre sur deux registre $r1 et $r2.
    ii. Le processeur (UCC) lance une commande au circuit additionneur afin qu'il fasse une addition des deux registre $r1 et $r2, puis garder le résultat sur une registre $r3,
    iii. Le résultat trouvé sera gardé (transféré de $r1 vers mémoire) dans la zone mémoire de la variable C.