

TD n°4

Micro-code (suite et fin) et Langage Machine

I. Traitement d'instructions LM

Ecrire en LMA les instructions LM ci-dessous.

Le LM utilisé est à 3 adresses (cf. l'aide mémoire).

On commencera par identifier le nombre d'opérandes d'entrée et la présence d'un résultat en sortie. Les éventuels opérandes d'entrée seront supposés déjà lus et stockés dans OP1 et OP2, la production d'un résultat s'opérera par stockage de ce résultat dans RES puis saut à l'adresse « sortie ».

Exercice 1 : Gestion de la pile système (pile descendante)

- PUSH
- POP
- NEWENV
- DESTENV

Exercice 2 : Transferts et calculs

- MOVE
- ADD
- SUB
- OPP

Exercice 3 : Rupture de flot

- GOTO
- JUMP
- ONGOTO (sans registre d'état)
- ONJUMP (sans registre d'état)
- CALL (en ajoutant un registre d'état – à discuter)
- RETURN

II. Programmation en LM

Ecrire en LM les programmes ci-dessous. Ils débiteront à l'adresse 200.

Les programmes seront écrits pour une machine 32 bits (donc 1 mot = 32 bits) disposant d'un registre d'état (et de l'instruction COMP) et de 6 registres libres (A, B, C, D, E, F).

La syntaxe utilisée est la suivante :

```
<Instruction> ::= <MotCode> [ <Adr1> ] [ <Adr2> ] [ <Adr3> ]
<MotCode>    ::= ADD, SUB, CALL, ... <RuptCond>
<RuptCond>   ::= <InstRC> <Cond>
<InstRC>     ::= ONGOTO, ONJUMP
<Cond>       ::= <, >, <=, >=, ==, !=
<Adr>        ::= <ModeAdr>:<ValeurNumériqueAdresse>
<ModeAdr>    ::= Im, Rg, Di, Ir, In, Lc, Ix, Li
```

pour Immédiat, Registre, Direct, Indirect par registre, Indirect, Local, Indexé, Local indirect.

Exercice 1 : Inversion de bit dans un mot mémoire

Cf. énoncé de l'exercice 1 du TD n°2.

Exercice 2 : Copie de bloc mémoire

Cf. énoncé de l'exercice 2 du TD n°2.