

Architecture avancée : TD6

Exercice 1 : Structures de contrôle

Sujet

Ecrire le code assembleur ARM correspondant aux extraits de code C suivants :

```
if (MaVar == 4) {  
    // exécute une tâche...  
} else {  
    // exécute une autre tâche...  
}  
// le programme continue...
```

FIGURE 1 – Extrait 1 de code C.

```
for (int i = 0; i < 5; ++i) {  
    // tâche à l'intérieur de la boucle  
}  
// le programme continue...
```

FIGURE 2 – Extrait 2 de code C.

```
int FonctionAUnParametre (int param)  
{ return param + 1; }
```

FIGURE 3 – Extrait 3 de code C.

```
LDR    R0, MaVar  
CMP    R0, #4  
BNE    else  
;Tâche 1  
B continue  
else  
;Tâche2  
continue  
;Suite  
  
for  
MOV    R0, #0  
CMP    R0, #5  
BGE    done  
;Tâche
```

```

        ADD    R0, R0, #1
        B      for
done

```

- ADDS / SUBS : S met à jour les drapeaux

```

        MOV    R0, #8
        B      FonctionAUnParametre
fonctionAUnParametre
;param dans R0
        ADD    R0, R0, #1
        BX     LR
        ...
        ...
        LOAD   R0, #42
        BL     fonctionAUnParametre

```

Exercice 2 : Tableaux

Sujet

Traduire le programme suivant en assembleur, en utilisant uniquement des branchements

```

Max = X[0];
Imin = 0;
Imax = 0;
For (i=1; i< N; i++)
{
    if (X[i] <Min){
        Min= X[i];
        Imin = i;}
    if (X[i] >Max){
        Max= X[i];
        Imax} = i;}}

```

FIGURE 4 – Pseudo-code.

Résolution

```

Max      LDR    R1, [R0]
Imin     MOV    R2, #0
Imax     MOV    R3, #0

```

```

i      MOV    R4, #1
      MOV    R6, #4
      MOV    R7, #0
      LDR    R5, N

for
      CMP    R4, R5
      BEQ    done
      LDR    R7, [R0, R6]
      CMP    R7, R9
      BGE    skip1
      LDR    R9, [R0, R6]
      ADR    R2, R4

skip1
      CMP    R7, R1
      BLE    skip2
      LDR    R1, [R0, R6]
      ADR    R3, R4

skip2
      ADD    R4, R4, #1
      ADD    R6, R6, #4
      B      for

done
      ...

```