

TP2

Les branchements et les subroutines

Application 1:

- 1- *Editer et assembler le programme .*
- 2- *Exécuter le programme pas à pas, en prélevant pour chaque instructions les registres et les cases mémoires affectées et leurs contenus.*

```
* TP22*
Que fait le programme
assembleur suivant :
DATA
V : DC.L 2
U : DC.L 9
T : DC.L 1
X : DC.L 4
CODE
MOVE.L X, D0
CMP.L #2 , D0
BNE SINON
MOVE.L U,T
BRA FIN
SINON : MOVE.L V,T
FIN : RTS
```

Application 2:

Exécuter le programme TP12 pas a pas, en résumant dans un tableau pour chaque instruction : Son code machine, et le contenu des cases mémoires et des registres affectées, ainsi que le contenu du registre d'état.

Dans le menu principal du simulateur ouvrez View memory et relevez le code machine de votre programme

```
* TP21 *
ORG $1000
x: dc.l 21      ; initialisation d'un
mot de 32 bits
str: dc.b "chaine"      ;
initialisation d'une chaîne
tab: ds.w 4      ; réservation de 4
mots de 16 bits
y: dc.w $1234
```

```
code
move.l x,d0
add.l #2,d0
lea Str,a0
move.b #'x',(a0)
lea tab,a1
move.w#$abcd,(a1)
rts
```

```
END $1000
```

Application 3:

- 1-Ecrire un programme qui reçoit une valeur dans D0 et qui retourne dans D1, 1 si D0 est impaire et 0 si D0 est paire.

MOVE.W D0,D1

AND.W #1,D1

- 2-Etudier ces deux programmes et interpréter le résultat après exécution de chaque instruction.

```
MOVE.L #$12345678,D0
MOVE.L #$00ABCDEF,D1
EOR.B D0,D1
EOR.B D1,D0
EOR.B D0,D1
```

```
ORG $A000
START TST.L D0
        BPL.S QUIT
NEG.L D0
QUIT
```

Attribuer la valeur -4 au registre D0 et prélever le résultat

Remarque : Noter les variations des différents drapeaux (Flags) du registre d'état pour chaque instruction