



Initiation au langage machine

Le langage machine est la suite de bits qui est interprétée par le processeur de l'ordinateur lors de l'exécution d'un programme. C'est le langage natif du processeur, et le seul qui soit reconnu par celui-ci.

Un processeur ne peut comprendre un langage rentré par l'utilisateur, un programme python ou java par exemple.

Entre les langages que l'on connaît et que l'on implante et le langage machine, il existe des langages intermédiaires et notamment le langage assembleur qui est le langage de plus bas niveau que l'on peut encore appréhender et que le processeur sait traduire en langage machine.

Il existe plusieurs langages assembleurs. On va travailler avec un simulateur et donc de fait utiliser sa syntaxe.

Merci au [site](#) , excellentissime, de D.Roche , qui m'a permis de découvrir ce simulateur.

En langage assembleur, on peut rencontrer des calculs simples (addition, soustraction, multiplication), des affectations (placer telle valeur de la mémoire vive dans tel registre et vice versa) et enfin des instructions de saut qui permettent de coder des instructions conditionnelles par exemple.

Il y a trois possibilités pour accéder à un opérande (valeur sur laquelle va être opérée une opération) :

- Il est directement saisi
- Il est dans un registre : il faut alors indiquer le nom de ce registre
- Il est dans la mémoire vive : il faut indiquer son adresse.

Syntaxe du langage

Syntaxe	Signification	Syntaxe	Signification	Syntaxe	Signification
ADD	Addition	LDR	Affectation	HALT	Fin du programme
SUB	Soustraction	STR	Affectation		
MOV	Affectation	B	Aller à		

Comparaison :

CMP	Comparaison CMP R ₀ , #10 : compare 10 et la valeur du registre R ₀
BEQ	BEQ 20 : Va à l'adresse 20 en mémoire si R ₀ =10
BNE	BNE 20 : Va à l'adresse 20 en mémoire si R ₀ ≠10
BGT	Va à l'adresse 20 en mémoire si R ₀ > 10
BLT	Va à l'adresse 20 en mémoire si R ₀ < 10

Remarque : Pour les instructions de saut, on donnera un Label et nom un nombre .

Exemple : BEQ test.... Si la condition est vérifiée, le programme sautera les lignes jusqu'à rencontrer test.

Exemple d'instructions

MOV R₀, #17 : Affecte la valeur 17 au registre R₀

ADD R₁, R₀, #18. Ajoute 18 à la valeur du registre R₀ et stocke le résultat dans R₁.

LDR R₀, 20 : place la valeur stockée à l'adresse mémoire 20 dans le registre R₀

STR R₀, 20 : place la valeur stockée dans le registre R₀ à l'adresse mémoire 20

Voilà, vous avez quelques syntaxes simples et suffisantes pour la suite. Notez bien que si vous voulez désigner un nombre, le # est nécessaire, sinon vous ferez référence à une adresse de la mémoire.

Exercice 1 :

Ecrire en langage assembleur les instructions suivantes :

- Additionner 18 avec la valeur du registre R₀ et stocker le résultat dans R₁
- Placer la valeur stockée à l'adresse 20 dans le registre R₀, lui soustraire 10 puis stocker le résultat dans le registre R₁
- Voici une suite d'instructions :

```

0      MOV R0, #4
1      STR R0, 30
2      MOV R0, #8
3      STR R0, 31
4      LDR R0, 30
5      LDR R1, 31
6      SUB R2, R0, R1
7      STR R2, 158
8      HALT

```

1. Indiquer la signification de cette suite d'instructions. Quel est le contenu de R,R1,R2 à la fin de ces instructions ?
2. Ecrire un programme en langage python qui peut être traduit en langage assembleur avec le code ci-dessus. Les variables correspondent aux adresses mémoires. Par exemple x <-> adresse 30, y <-> adresse 31...

Exercice 2

Voici un code en langage python :

```

x = 4
y = 8
if(x == y):
    y = x-4
else:
    y = x+y

```

Compléter le code ci-contre.

```

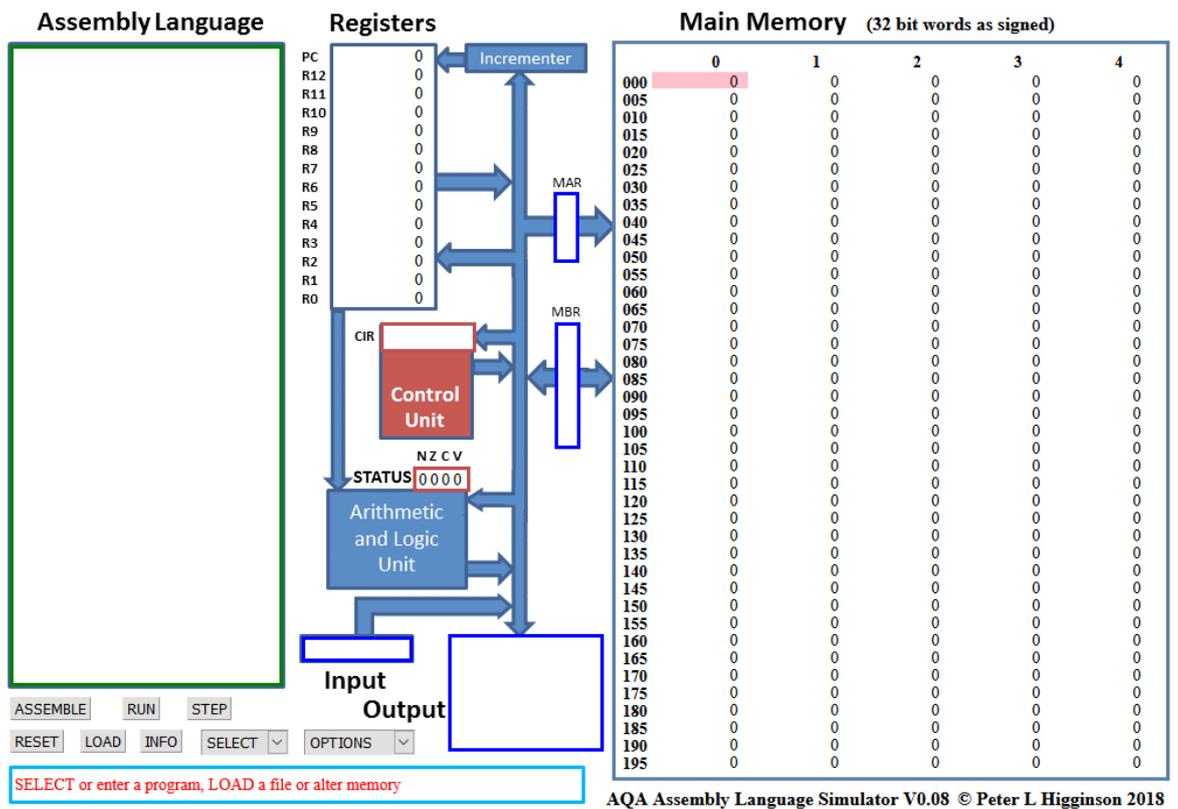
MOV ..., #4
STR R0,....|
MOV ..., #..
STR ....
LDR ...
LDR ....
CMP ....
BNE test
SUB ....
.
.
test:
ADD ....
.
.
HALT

```

Exercice 3

_Découverte du Simulateur en TD

<http://www.peterhigginson.co.uk/AQA/>



1. Tapez un code de deux ou trois lignes pour appréhender le fonctionnement du simulateur.
2. Recopier le code de l'exercice 2, en changeant les valeurs de x et de y pour que les deux cas soient testés.