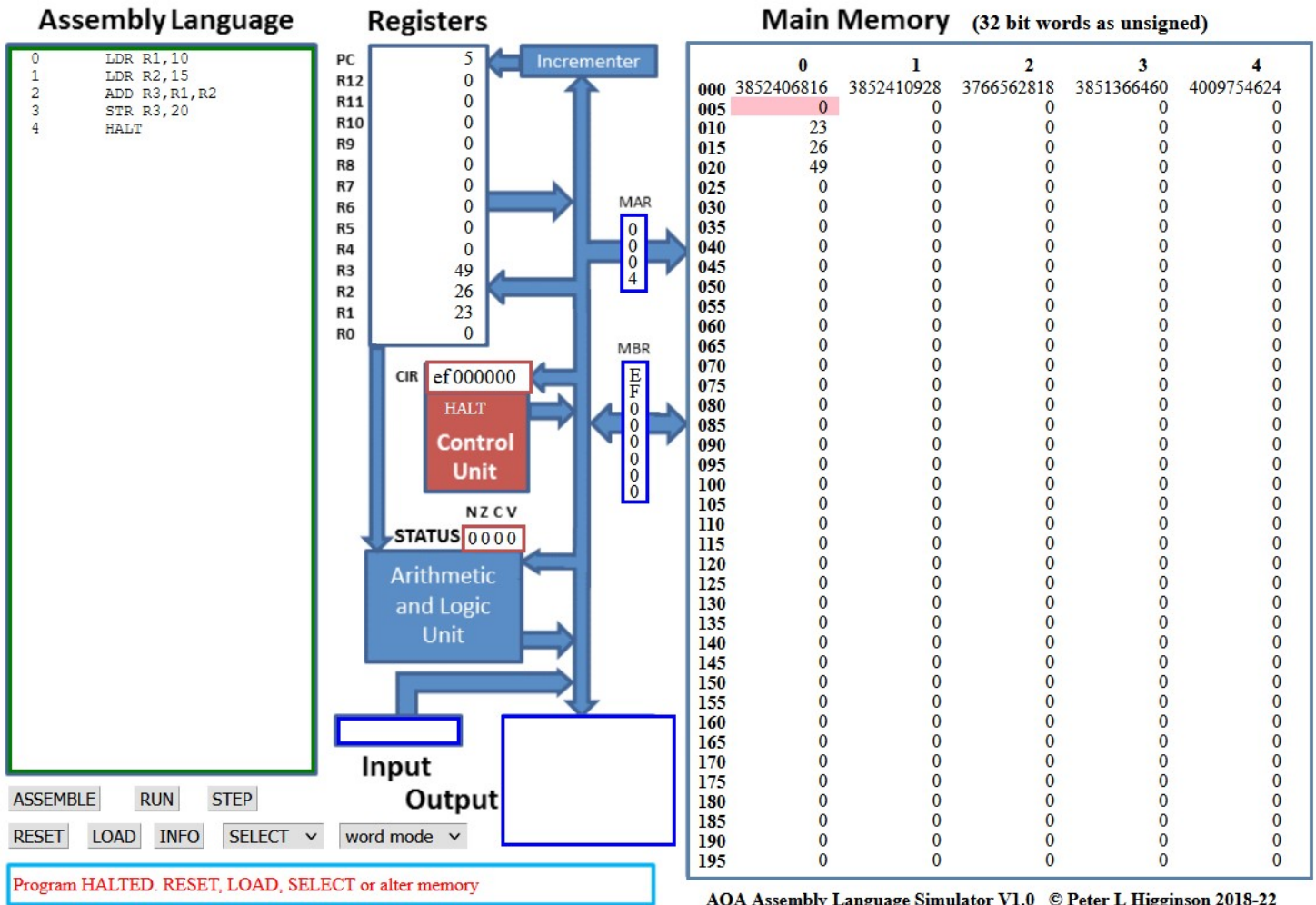


TP : Langage machine

Il existe sur le web un simulateur pour réaliser des instructions en langage assembleur/machine à l'adresse : <https://www.peterhigginson.co.uk/AQA/>



On donne ci-dessous les **principales instructions** du langage assembleur :

L'opérande <opérande> est une valeur désignée par #n ou Rm pour utiliser le contenu du registre Rm.

- LDR Rd, <mem ref> : Charge la valeur stockée dans l'emplacement mémoire spécifié par <mem ref> dans le registre d.
- STR Rd, <mem ref> : Stocke la valeur du registre d dans l'emplacement mémoire spécifié par <mem ref>.
- ADD Rd, Rn, <opérande> : Ajoute la valeur spécifiée dans <opérande> à la valeur dans le registre n et stocke le résultat dans le registre d.
- SUB Rd, Rn, <opérande> : Soustrait la valeur spécifiée par <opérande> de la valeur dans le registre n et stocke le résultat dans le registre d.
- MOV Rd, <opérande> : Copie la valeur spécifiée par <opérande> dans le registre d.
- CMP Rn, <opérande> : Compare la valeur stockée dans le registre n avec la valeur spécifiée par <opérande>.

- B <label> : Branche toujours sur l'instruction à la position <label> dans le programme.
- B <condition> <label> : Branche conditionnellement à l'instruction à la position <label> dans le programme si la dernière comparaison répond aux critères spécifiés par la <condition>. Les valeurs possibles pour <condition> et leur signification sont : EQ pour Égal à, NE pour Non égal à, GT pour Supérieur à et LT pour Inférieur à.
- AND Rd, Rn, <opérande> : Effectue une opération ET logique au niveau du bit entre la valeur dans le registre n et la valeur spécifiée par <opérande> et stocke le résultat dans le registre d.
- ORR Rd, Rn, <opérande> : Effectue une opération OU logique au niveau du bit entre la valeur dans le registre n et la valeur spécifiée par <opérande> et stocke le résultat dans le registre d.
- EOR Rd, Rn, <opérande> : Effectue une opération logique binaire ou exclusif (XOR) entre la valeur dans le registre n et la valeur spécifiée par <opérande> et stocke le résultat dans le registre d.
- MVN Rd, <opérande> : Effectue une opération NON logique au niveau du bit sur la valeur spécifiée par <opérande> et stocke le résultat dans le registre d.
- LSL Rd, Rn, <opérande> : Décale logiquement à gauche la valeur stockée dans le registre n du nombre de bits spécifié par <opérande> et stocke le résultat dans le registre d.
- LSR Rd, Rn, <opérande> : Décale logiquement à droite la valeur stockée dans le registre n du nombre de bits spécifié par <opérande> et stocke le résultat dans le registre d.
- HALT : Arrête l'exécution du programme.

Langages

- 1) Qu'est-ce que le langage machine ?
- 2) Qu'est-ce que le langage assembleur ?
- 3) Comment un programme écrit en langage de haut niveau est-il transformé en langage machine ?

Langage assembleur

1) Décrire par des phrases ce que font les différentes instructions suivantes écrites en langage assembleur.

a) ADD R0, R1, #25

b) LDR R2, 64

c) MOV R3, #45

d) STR R4, 72

e) SUB R5, R2, R3

f) CMP R3, #25
BGT 15

2) Sur la capture de la première page, un programme en langage assembleur est donné. Que réalise ce programme ?

3) Écrire en langage assembleur correspondant les instructions décrites par les phrases suivantes :

— Place la valeur 15 dans le registre R0

— Place la valeur 7 dans le registre R1

— Additionne la valeur stockée dans le registre R0 et la valeur stockée dans le registre R1, le résultat est stocké dans le registre R5

— Place le contenu du registre R5 à l'adresse mémoire 125.

- Place la valeur 10 dans le registre R1
- Place la valeur stockée à l'adresse mémoire 125 dans le registre R0
- Soustrait la valeur stockée dans le registre R0 et la valeur stockée dans le registre R1, le résultat est stocké dans le registre R5
- Place le contenu du registre R5 à l'adresse mémoire 125.

4) Quelles sont les valeurs dans les différents registres à l'issu de ce programme ?

5) Saisir votre programme dans le simulateur et vérifier vos réponses.