

Dans le tableau suivant, <operande> peut soit être une *valeur immédiate* s'il commence par un # ou la valeur stockée dans d'un *registre* s'il commence par un R.

(ex : #25 est le nombre 25 ; R6 désigne le registre n°6).

Un <label> identifie une adresse en mémoire vive dans le programme assembleur. (ex : **fin**: HALT).

LDR Rd, <adresse>	Charge la valeur stockée à l'adresse mémoire <adresse> dans un registre (LoaD Register)
STR Rd, <adresse>	Place la valeur stockée dans le registre d en mémoire vive à l'adresse <adresse> (STore Register)
ADD Rd, Rn, <operande>	Additionne <opérande> et le registre n, place le résultat dans le registre d (ADD)
SUB Rd, Rn, <operande>	Soustrait <opérande> de la valeur stockée dans le registre n, place le résultat dans le registre d (SUBstract)
MOV Rd, <operande>	Copie <opérande> dans le registre d (MOVe)
B <label>	Branchement inconditionnel vers l'instruction située en mémoire vive à l'adresse identifiée par <label>
CMP Rn, <operande>	Compare le registre n et <opérande> (CoMPare). Précède une instruction de branchement conditionnel.
B<condition> <label> BEQ <label> BNE <label> BGT <label> BLT <label>	Branchement conditionnel à l'adresse identifiée par <label> si la dernière comparaison satisfait le critère spécifié par <condition>. Les valeurs possibles de <condition> sont : EQ : égal à (EQual), NE : différente (Not Equal) GT : supérieur (Greater Than), LT : inférieur (Less Than)
AND Rd, Rn, <operande>	ET logique bit à bit entre le registre n et <operande>, place le résultat dans le registre d.
ORR Rd, Rn, <operande>	OU logique bit à bit entre le registre n et <operande>, place le résultat dans le registre d.
EOR Rd, Rn, <operande>	OU Exclusif bit à bit entre le registre n et <operande>, place le résultat dans le registre d.
MVN Rd, <operande>	NON logique de la valeur <operande>, place le résultat dans le registre d.
LSL Rd, Rn, <operande>	(Logical Shift Left) : décale les bits du registre n vers la gauche en ajoutant un nombre de 0 à droite égal à <operande>, place le résultat dans le registre d.
LSR Rd, Rn, <operande>	(Logical Shift Right) : décale les bits du registre n vers la droite en ajoutant un nombre de 0 à gauche égal à <operande>, place le résultat dans le registre d.
INP Rd, 2	Lit un nombre depuis une Entrée et enregistre sa valeur dans le registre d. (INPut)
OUT Rd, <device>	Écrit la valeur du registre d dans une sortie. (OUTput). La valeur de <device> indique le type de sortie : 4 : nombre signé 5 : nombre non signé 6 : hexadécimal 7 : caractère ASCII
HALT	Arrête l'exécution du programme

Exemple de LSL : LSL R2, R1, 3. Si R1 vaut 01110111, R2 devient 10111000.

Exemple de LSR : LSR R2, R1, 3. Si R1 vaut 01110111, R2 devient 00001110.