

NSI Première - Architecture - Résumé des mnémoniques ARMLite

qkzk

2022/02/14

Description simplifiée de l'assembleur ARMLite

ARMLite est une *machine* disposant d'une architecture de Von Neumann qui exécute du code saisi en mnémoniques.

- Elle dispose de 13 registres généraux : R0 à R12. On les notera Rx, Ry etc.
- Les nombres *littéraux* sont notés #val, par exemple #20 désigne le nombre 20, *vingt*.
- Sa mémoire est initialisée avec le code du programme.
- Les autres registres sont utilisés pour le fonctionnement interne, quelques bits indiquent l'état de la dernière opération

C'est donc un *émulateur* de processeur respectant une version légère du modèle ARM. On peut lui faire exécuter des programmes et interagir avec elle via une interface web dans laquelle on écrit de l'assembleur et exécute les instructions.

Instructions

Les instructions les plus couramment employées sont les suivantes :

1. Déplacement de contenu

- MOV Rx, val : déplace val dans le registre Rx. val peut-être un registre ou un littéral.
- LDR Rx, adresse : charge dans le registre Rx le contenu de l'adresse mémoire indiquée,
- STR Rx, adresse : écrit le contenu du registre Rx à l'adresse mémoire indiquée,

Attention, seules les adresses multiples de 4 sont valides. En effet, ARMLite fonctionne avec des *mots* de 4 octets et chaque octet dispose d'une adresse.

Donc LDR R0, 40 est valide mais LDR R0, 41 est invalide, car 41 n'est pas multiple de 4.

2. Opérations arithmétiques et logiques

- ADD Rx, Ry, val : effectue l'addition Ry + val et écrit le résultat dans Rx. val peut être un registre ou un littéral.
- Les instructions SUB, AND, ORR, EOR (respectivement soustraire, ET logique, OU logique, OU EXCLUSIF) ont un fonctionnement similaire.

3. Rupture de séquence

- CMP Rx, Ry : effectue la comparaison entre Rx et Ry
- BGT label : saute vers l'instruction précédée du nom label: SEULEMENT si la précédente comparaison était "plus grand que."
Par exemple :
 - si on a comparé 4 et 2, alors BGT label va sauter en label:,
 - si on a comparé 4 et 10, alors BGT label va passer à l'instruction suivante.
- BEQ label : saute vers l'instruction précédée du nom label: SEULEMENT si les valeurs comparées étaient égales,
- BLT label : saute seulement si la dernière comparaison était "plus petit que",
- B label : saute, *sans condition* vers label:,
- HALT : arrête la machine.

Exemple de programme

```
1|      MOV R1, #10      //      R1 contient 10
2|      MOV R2, #5       //      R2 contient 5
3|      ADD R3, R1, R2    //      R3 contient 5 + 10 = 15
4|      CMP R3, R1        //      Comparaison de 15 et 5
5|      BGT more          //      15 > 5 donc on va en "more", ligne 08.
6|      HALT              //      Fin du programme (ne sera pas exécuté dans l'exemple)
7|                          //      Rien, on va à la ligne
8|  more:                 //      pas de code après l'étiquette, on ne fait rien
9|      ADD R3, R1, R3    //      R3 contient 15 + 10 = 25
10|     HALT              //      Fin du programme
```