

Architecture Von Neumann

Activité

En vous basant sur les sites ci-dessous, compléter le document d'étude sur l'architecture de Von Neumann.

1. [Le modèle d'architecture de Von Neumann](#)
2. [Mémoire et unité centrale, un couple dédié à l'exécution des programme](#)

Pour la partie Langage machine, on se référera au site wikipédia:

1. https://fr.wikipedia.org/wiki/Langage_machine
2. https://fr.wikipedia.org/wiki/Assembleur#Directives_du_langage_assembleur

Pour utiliser l'émulateur assembleur, voici quelques instructions.

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

Les registres Rm du processeur vont de R0 à R12.

- **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.

D'autres informations sur ces instructions sur les sites :

1. https://pixees.fr/informatiquelycee/n_site/nsi_prem_sim_cpu.html
2. <https://www.peterhigginson.co.uk/AQA/>

Document support de l'activité :

https://moodle.l-educdenormandie.fr/pluginfile.php/106782/mod_page/content/9/ActArchiVonNeumann.pdf