

2.2 Registres

Le bloc de registres contient 32 registres, dont l'organisation et le fonctionnement sont calqués sur le modèle MIPS. Ou pas. TODO

2.3 Control

Le bloc de contrôle, présent dans notre projet sous le nom de “Décodeur” a dans notre projet non seulement le rôle du bloc “Control” de l'architecture standard MIPS mais c'est aussi lui qui s'occupe de la synchronisation entre les différents cycles parcourus pour une instruction donnée. TODO à détailler

2.4 Instruction Memory

TODO

2.5 Data Memory

La mémoire va être implémentée par un device. TODO à détailler

3 Détails techniques

3.1 Format des instructions

TODO explication du *R*-format, *I*-format, format des jumps, etc.

3.2 ALU

Fonctionnement : Le bloc $\text{Alu}_{<m>}$ du code source prend 3 entrées :

- deux arguments a et b sur m bits,
- un sélecteur d'opération sur 4 bits,

et en fonction de la valeur du sélecteur, qui contient le “code” de l'opération voulue (0 pour And, 1 pour Or, etc.), il renvoie le résultat sur m bits, tronqué si besoin (les éventuelles retenues sont perdues, tous les calculs sont faits modulo 2^m).

Comment ça marche : Chaque opération de l'ALU est codée séparément. Puis, à la lecture de deux entiers, toutes les opérations possibles sont exécutées simultanément, et un filtre sélectionne le résultat voulu. Tout fonctionne en un seul cycle, et les circuits correspondant aux diverses opérations ont tous été codés récursivement.

3.3 Registres

TODO, etc.