

TP4 : projet Machine de Moore

Rendez ce projet noté à la fin de la 7e séance TP à votre enseignant de TP avec un rapport incluant des dessins de machines de Moore et des fichiers mémoire. Rendez des réalisations partielles commentées, si vous n'arrivez pas au bout de certains exercices.

Dans ce projet, vous allez programmer une machine de Moore pour réaliser différentes opérations arithmétiques avec un circuit séquentiel. Votre base est un projet tp4moore.tar.gz.

Ce projet comprend :

- Le module main avec la calculatrice deux entrées que vous pouvez modifier (aussi pendant la simulation), un interrupteur reset et une sortie. Les affichages sont décimaux avec une LED pour désigner le signe. Les entrées se font en codage 32 bits complément à 2. Un LED après la calculatrice signifie que le calcul a terminé.
 - Le module calculatrice comprend
 - Un banc de 16 registres de 32 bits, dont R0 et R1 les deux entrées de la calculatrice et R2, R3 liés à des constantes que vous pouvez choisir pour des fins de programmation de la machine. Les 12 autres registres sont modifiables. Le banc de registres permet de choisir deux registres de sortie par index et un registre à écrire (si souhaité).
 - Une UAL avec 8 opérations et deux entrées supplémentaires pour modifier l'addition. Elle dispose également de deux sorties drapeaux, une sortie test à zéro et une sortie pour des résultats négatifs dans le complément à 2.
 - Une unité de contrôle. Cette dernière dispose d'un registre 5 bits pour l'état de la machine (jusqu'à 32 états). Elle utilise 2 bits à l'entrée (les drapeaux de l'UAL) et elle génère 20bits à la sortie. Deux mémoires représentent séparément la fonction de successeur de la machine et les « instructions » sorties par la machine. Elle contrôle ainsi l'UAL et le banc de registres :
 - les index des registres d'entrée de l'UAL (2x 4 bits) ;
 - l'index d'un registre à écrire (si on choisit d'écrire) ;
 - un bit WEN pour choisir d'écrire ou non ;
 - les entrées de contrôle de l'UAL (opération, cin, invb) ;
 - l'état suivant de la machine (7 bits, soit 2 chiffres hexadécimal) ;
- Les mémoires mortes sont liées à deux fichiers mémoire listant simplement les 128 lignes en codes hexadécimaux (pour les successeurs) et 32 lignes (pour les sorties par état). Les fichiers fournis `simpleaddsucc.mem` et `simpleaddsortie.mem` réalisent avec deux états une simple opération d'addition entre les deux entrées.

Exercice 1 Lisez tous les modules du projet et essayez de comprendre intégralement leur fonction. Répondez aux questions suivantes :

1. Est-ce dangereux de sélectionner un des quatre pseudoregistres pour l'écriture ?
2. Décodez les « instructions » de `simpleadd.mem`.
3. Par quelles sorties du module contrôle peut-on réaliser une affectation $R_i=0$, $R_i=-1$ sans utiliser R2,R3 ? Comment réaliser une affectation $R_i=1$?
4. Quelles sont les options pour réaliser une décrémentation $R_i=R_i-1$?
5. Comment peut-on réaliser un test $R_i < R_j$ ou $R_i \leq R_j$?

Exercice 2 Réalisez et testez la machine vue en séance TD qui réalise une multiplication simple des deux entrées, en supposant qu'elles soient positives, dans le style du programme suivant :

```

// R0 = x, R1 = y, par défaut
//R4=0;
//R5=0;
while(R5!=R1){
    R4=R4+R0;
    R5=R5+1;
}
while(1){
    // émettre R4 et done=1
}

```

Exercice 3 Remplacez pour la multiplication simple les mémoires par des circuits combinatoires réalisés à partir de portes logiques.

Exercice 4 Concevez et réalisez un algorithme simple (par soustraction successive), puis une machine qui réalise une division (entière) de nombres positifs.

Exercice 5 Concevez et réalisez un algorithme et une machine plus efficace (par décalage et par soustraction successive) pour la division entière.

Exercice 6 (Bonus) La représentation BCD (binary coded decimal) représente un nombre décimal avec 4 bits par chiffre. Par exemple, le codage du nombre décimal 125 en BCD est «0001 0010 0101 », sa représentation binaire est « 0000 0111 1101 ».

Réalisez une machine qui réalise une conversion BCD (4 bits par chiffre décimal) vers binaire. Vous pouvez tester cette machine en saisissant les entrées hexadécimales avec des chiffres 0-9 seulement. Le résultat affiché (qui utilise un décodage pour un affichage décimal) doit correspondre à l'entrée.