

ARC2 – TP 1

Léo Noël-Baron & Thierry Sampaio

17/01/2016

UC cassée

Le fichier `UC.bdf` décrit une UC non fonctionnelle; en tentant de l'utiliser, on constate que le cycle 1 (commun à toutes les instructions, deuxième cycle de **fetch**) ne déclenche pas le signal `ChIR`. Le registre d'instructions ne charge alors jamais d'instruction réelle, ce qui explique le dysfonctionnement. Il suffit de relier `ChIR` sur le compteur de cycles, au cycle 1, et le problème est réglé.

Cependant le processeur ne fonctionne toujours pas; en effet, le signal `R>Bus` n'est pas branché non plus. Il faut le relier judicieusement aux cycles 2, 3 et 5 en fonction des diverses commandes qui l'utilisent (`stw`, `ldw`, ...), et le processeur exécute finalement le programme chargé sans erreur.

UAL cassée

Le chargement du fichier `UAL.bdf` provoque de nouvelles erreurs. En examinant le schéma du circuit, on constate que certaines commandes (`add` et `inc4`) ne sont tout simplement pas implémentées. L'ajout des circuits adéquats corrige toutes les erreurs d'exécution; le NIOS est réparé.

Une instruction de copie

Ajouter une nouvelle instruction implique de la placer dans le tableau des OP-codes existants; dans notre cas, on peut choisir pour une instruction de copie registre à registre, le format R-type et prendre comme OPX n'importe quelle entrée vide du tableau. La syntaxe voulue est `mov rB, rA`; comme on ne peut pas sélectionner plusieurs registres à la fois, il faut faire transiter le contenu à copier dans le registre interne C du processeur. Ainsi, cette instruction se décompose en deux cycles :

Cycle 2		<code>C <- rA</code>	<code>selrA, ChC, R>Bus</code>
Cycle 3		<code>rB <- C</code>	<code>selrB, ChR, C>Bus</code>

Pour ajouter l'instruction au processeur, il suffirait de rajouter les branchements adéquats aux signaux mentionnés, et de relier l'OP-code choisi.