

Présentation de notre simulateur

Nicolas Blanchard, Axel Davy et Marc Heinrich

21 octobre 2012

1 Comment exécuter notre compilateur

2 Options disponibles

3 Difficultés rencontrées

Après la première séance du TD, nous avons une fonction scheduler qui permettait de faire un tri topologique et de mettre ainsi dans l'ordre les instructions d'un fichier netlist source. Mais nous avons eu beaucoup de mal à avoir une fonction qui marche aussi pour les registres. En effet il faut que la fonction ne détecte pas de cycle combinatoire quand il y a une boucle causée par un registre. De plus si deux registres sont chacun connectés à la sortie de l'autre, on ne peut en exécuter un avant l'autre. Nous avons donc décidés d'enlever les liaisons des registres dans le graphe du tri topologique, ainsi le tri se fait sans considérer les registres, se qui coupe les éventuels cycles combinatoire qui auraient été repérés à cause d'une boucle de registre. Nous avons aussi fait en sorte que les registres donnent leurs sorties au début de l'exécution et capturent leur entrées à la fin de l'exécution. Pour cela nous avons mis les registres dans une liste pour donner leur sortie au début de l'exécution et nous les avons mis à la fin de la liste à exécuter pour qu'ils puissent capturer leur entrée. Comme le type "programme" était incomplet à notre goût on a créé un nouveau type Mprogramme : On souhaitait une liste des cas à traiter à part (comme les registres) et nous voulions que les équations soient composées de clés indiquant la valeur de leur arguments dans un tableau.