

Projet de simulateur de microprocesseur

Nathan Boyer, Matteo Torrents, Antoine Cuvelier, Grégoire Lecorre

11 décembre 2023

Introduction

On cherche à coder un simulateur de microprocesseur en Ocaml.

Pour cela, on va utiliser un simulateur de netlist qui exécutera des netlists écrites en miniJazz.

Cette netlist peut exécuter les opérations arithmétiques de base (and, or etc ...) sur un bit, manipuler des bus de bits (splice, concat ...)

De plus, l'instruction reg permet de se prendre la valeur du cycle précédent d'une variable. L'instruction RAM permet d'écrire et de lire dans la mémoire (par soucis de réalisme, on considérera qu'il n'existe qu'une seule RAM). L'instruction ROM, quant à elle, permet de lire le contenu d'un fichier composé de 0 et de 1

On va alors programmer un langage assembleur qui nous permettra de communiquer avec notre microprocesseur. Notamment, on programmera une horloge textuelle, puis une horloge graphique pour tester notre microprocesseur.

1 Les registres

Tous les calculs du microprocesseur seront faits sur des registres de 16 bits.

Il y a en tout 8 registres.

2 Représentation des instructions

Le microprocesseur lit les instructions qu'il doit exécuter depuis un fichier (lu avec l'instruction ROM)

Chaque instruction fait 16 bits, prend en argument 2 numéros de registre et une constante, et est découpée en

- 4 bits du code de l'instruction
- 2*3 bits de numéro de registre
- 6 bits de constante

3 Du langage assembleur à la ROM

Un parseur menhir lit les instructions et écrit dans un fichier lisible par la ROM les différentes instructions

Voici une liste des différentes instructions, de leur code en binaire, et de leur effet.