Homework 8 – ARM Pipelined Processor

Please read and follow the steps of this project closely. As always, don't forget to read the entire document and refer to the "What to Turn In" section before you begin.

Introduction

In this homework, you will design and build your own pipelined ARM processor that should match the design from the text. It should handle the following instructions: `ADD`, `SUB`, `AND`, and `ORR` (with register and immediate operands, but no shifts), `LDR` and `STR` (with positive immediate offset), and `B`.

The pipelined processor is divided into three units: the `controller`, `datapath`, and `hazard` units.

A new version of the SPI interface is also included and allows the transfer of information between the Raspberry PI and the DE0-Nano.

In this homework, you will implement the `hazard` module, add a new instruction and use the SPI interface.

Hazard Module

In the `hazard` module, the implementation of these signals is missing: `ForwardAE`, `ForwardBE`, `StallF`, `StallD`, `FlushD`, `FlushE`.

Study carefully the pipelined architecture and add the SystemVerilog code for these signals. Simulate your design on ModelSim and verify you code on INGInious.

Adding a new instruction

Choose a new instruction and update the architecture to implement it. Modify the assembler code to test this instruction and check you design on ModelSim.

Using the SPI interface

The new SPI interface is based on a FSM and includes two sets of 16 32-bit registers: the first set is used to transfer information from the PI to the ARM while the other set is for transfers from the ARM to the PI. Study this design and use Signal Tap to display some internal signals.

A fully functional configuration of Signal Tap is provided.

The Python program `MyARM_Pipelined.py` implements a simple demo:

- Register 0 : counting frequency
- Register 1 : 0x05
- Register 2 : 0x04
- Display the content of Register 3 : 0x09 = 0x05 + 0x04. This operation is done by the ARM.

Imagine a new function with more than two parameters and more than one output. Update the assembler code and the Python program to test it. The parameters are sent to the ARM by the PI. The ARM executes the operations and sends the results to the PI.

What to Turn In

Validate only the first task of this homework on INGInious You have to upload the modified content of the file `MyARM_Pipelined.sv`.

Write a detailed report based on the template. You have to follow this template carefully!

Upload on Moodle 2 files:

- A PDF file untitled "Homework8-Report-StudentLastName-StudentFirstName.pdf" that includes your report
- A ZIP file untitled "Homework8-Simu-StudentLastName-StudentFirstName.zip" that includes all the simulation files