ECSE 425: Computer Organization and Architecture
VHDL Refresher: Pipelining

Due January 24, 2020, 11:59 PM

Introduction

The goal of this assignment is to build a pipeline for the following equation:

Output = ((a + b) * 42) - (c * d * (a - e))

The purpose of a pipeline is to subdivide a complex operation into several independent parts, thereby making it possible for each part to operate concurrently. A functioning pipeline will accept a new input each cycle, and once the first inputs have propagated through the pipeline stages, a new output will be produced each cycle.

Your VHDL module should have the following inputs and outputs:

- clk : in std_logic;
- a, b, c, d, e : in integer;
- op1, op2, op3, op4, op5, final_output : out integer

where

- clk is the clock
- a, b, c, d and e are inputs for the equation
- op1 is the intermediate result of a + b
- op2 is the intermediate result of op1 * 42
- op3 is the intermediate result of c * d
- op4 is the intermediate result of a – e
- op5 is the intermediate result of op3 * op4
- final_output is the result of op2 – op5

Example

Given the parameters [a = 1, b = 2, c = 3, d = 4, e = 5], the outputs at each stage should be:

- op1 = 3
- op2 = 126
- op3 = 12
- op4 = -4
- op5 = -48
- final_output = 174

Where to start

Three files are provided to you for this assignment:

- `pipeline.vhd`: You will implement your pipeline module in this file. Do not change the port map (entity).
- `pipeline_tb.vhd`: You will implement a complete testbench for your pipeline.
- `pipeline_tb.tcl`: You don't have to edit this file. It is a script to compile and run your testbench.

To compile, open ModelSim and change the directory (File, Change Directory) to the one containing those three files. In the Transcript section (ModelSim console), run the following command.

```
source pipeline_tb.tcl
```

If you don't have compilation errors, you should see the waves appear in the Wave section and your test results in the Transcript section.

Grading

Two aspects of your deliverable will be evaluated: (a) the correctness of your implementation, as evaluated by our testbench, and (b) the completeness of your testbench, with respect to test coverage.

In this case of this deliverable, test coverage is determined by the extent to which different combinations of pipeline state have been evaluated (e.g., as the pipeline fills, is full, and drains).

Hand In Procedure

Hand in, via MyCourses, in a single ZIP file:

- `pipeline.vhd`
- `pipeline_tb.vhd`