**Lab 5: 32-bit Single-cycle Datapath and Controller Design**

**Overview:** In a series of laboratory exercises you will incrementally implement a 32-bit processor that supports a basic set of operations. In Task 0, you implement the datapath components required by the processor. In this part and later on, you will put together a datapath using these components and design a state machine that controls this datapath. Datapath will be configured by the controller to execute a specific operation in a single clock cycle. This way the datapath will be able to execute different operations in each cycle. You will eventually execute matrix multiplication code on your datapath to validate the functionality of your datapath.

**Lab 5 schedule overview (Tentative – subject to change)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Score** | **Start date** | **Due date** |
| Task 0 | 125 points | Week of  Oct 22 – Oct 26 | The first 60 minutes of your lab session during the week of Oct 29 – Nov 2, 2018 |
| Pre-lab of Task 1 |  | Week of  Oct 29 – Nov 2 | The **first 30 minutes** of **your** **lab session** during the **week of Nov 5- Nov 9, 2018** |
| **Task 1** | **325 points** | **As soon as you finish Task 0** | **By 4.45 pm on Friday Nov 16th, 2018** |
| Task 2 | 150 points | As soon as you finish Task 1 | By 4.45 pm on Thursday Nov 29th, 2018 |
| Task 3 | 100 points | As soon as you finish Task 2 | By 11.45 am on Wednesday May 2nd, 2018 |
| **Total** | **700 points** |  |  |

**Lab 5 - Task 1 (325 points)**

**Starts:**  as soon as you finish Task 0

**Demo:** See below

**References:**

* Lecture Notes: Verilog ppt slides and Verilog chapter on Zybook
* “single\_cycle\_datapth\_Task1” on D2L under Lab 5

**Objectives:**

* Practice writing Datapath and Controller and the testbench
* Continue using the ***post-synthesis simulation*** for your design

**Pre-lab** (the first 30 minutes of your lab session during the week of **Nov 5- Nov 9, 2018 or before**)

**See PowerPoint slides “single\_cycle\_datapth\_Task1” on D2L under Lab 5.**

**(10 pts)** Complete Slides 10 and 12 of “single\_cycle\_datapth\_Task1”

Read slides 2 – 9 and complete the table on Slide 10 (Exercise 1)

Read slide 11 and complete the table on Slide 12 (Exercise 1 cont.)

**Assignments for Lab 5 Task 1:**

**See PowerPoint slides “single\_cycle\_datapth\_Task1” on D2L under Lab 5.**

**(315 pts)** Write Verilog codes (see Slide 13 of “single\_cycle\_datapth\_Task1” ppt) to implement

* Controller: Using the tables generated in Exercise 1 (ppt slides 9 and 11), implement (write Verilog code) a controller for the given Datapath. The controller (see its block diagram on slide 4) has 2 inputs: 6-bit opcode and 6-bit func and its outputs are control signals listed in the table of Exercise 1 (ppt slides 10 and 12).
* Datapath: Using slide 3, write the Verilog code to implement Datapath (structural way – connecting several modules that you already implemented together as shown on Slide 3). It has two inputs: Clock, Reset and 1 output: the output from the rightmost 2x1 32-bit mux on slide 3.

The above Datapath is NOT complete (it does not support shift operations yet) – read slide 14 and complete the Datapath

* Integrate Datapath and Controller: Call/Instantiate Controller in the Datapath code.
* Run behavioral simulation.

Before running the simulation,

- Read slide 15 and make sure that your Instruction memory has what is required as stated in slide 15

- Write a testbench.

Make sure that RegFile[8] shows the correct answers for each instruction in the given code before continue to the next step.

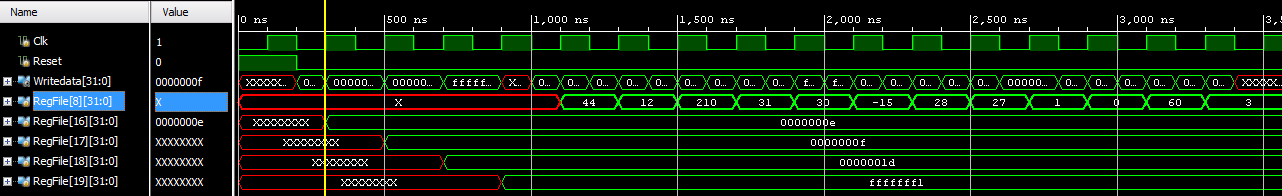
If your processor is working, you should have the following waveform (see Figure 1)

* Run Post-synthesis Functional simulation.

Note: See point distribution on slide 16.

* 30% deduction if your circuit work only in the behavioral simulation.
* Maximum of 25% of the total score will be given if your processor does not function at all.

**Figure 1: Output from Behavioral simulation**

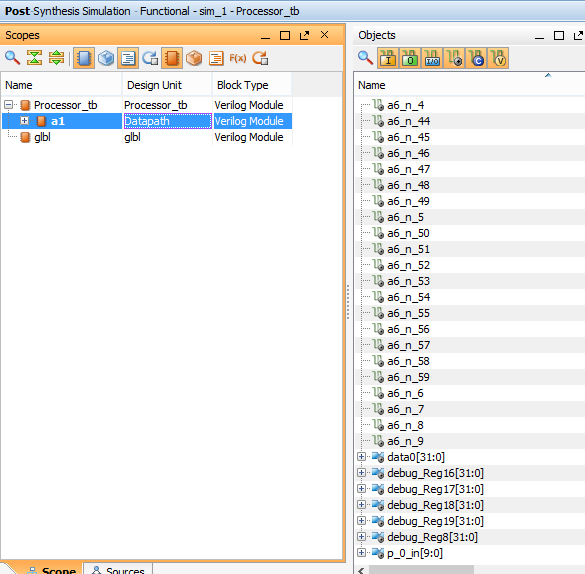


If your waveform show 0 (in place of X) - it is okay.

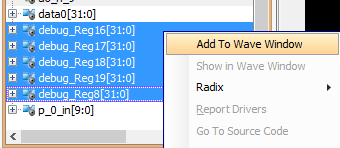
For **Post-synthesis Functional simulation**, we have to retain the signals such as RegFile8, 16, 17, 18 and 19 so that we can see their waveforms.

Download the zipped folder “retaining\_signals\_in\_post\_synthesis\_simulation\_guide”

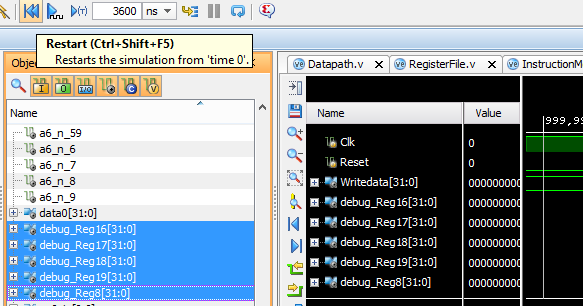
* Read “README” file
* Open “RegisterFile” in the folder to modify your RegisterFile.v file accordingly.
* Open “Datapath\_top” in the folder to add the necessary signals to your Datapath..v file
* Run synthesis -> Run Post-synthesis Functional simulation.
* After you run it, you should be able to see the following signals under Objects window. If not, click on and then look for those signals again



Then select all 5 signals (or one at a time) and add them to the waveform window. You can drag them to the waveform window or use “Add to Wave Window”.



Then click on **Restart** button  -> then  to rerun your simulation.



If working, you should see the following for the correct answer.

