**Project Deadlines and Demonstration Procedure**

There are three phases. First phase is the verification of pipelined datapath with forwarding and hazard detection. Second phase is the execution of the vbsme.s on your datapath. Third phase is the completion.

* **Phase 1 Deadline: Week of November 27 before your scheduled demonstration time slot**
  + **Submit all .v files “phase1” dropbox**
* **Phase 2 Deadline: Nov 29, before your lab section.** 
  + **Submit all .v files “phase2” dropbox**
* **Phase 3 Deadline: December 4, before your lab section.** 
  + **Submit all .v files on “phase3” dropbox.**

**Phase 1: (2000 points)**

* **Implement forwarding in ID, EXE, MEM stages**

Incrementally add forwarding units, show that each case is resolved with forwarding

Write a short program for each dependency case and test individually

* **Implement Hazard detection**

Identify all cases that require hazard detection (cases where forwarding cannot resolve the dependency) and implement.

* **Demonstration:**
  + Submit all .v files on D2L “phase1” folder before your scheduled demonstration time slot.
  + Functional verification based on post-routing simulation during the week of Nov 27
  + Your instruction memory must read from input file named "Instruction\_memory.txt".
    - Inside initial begin block of Instruction\_memory.v, use following initialization method:
      * $readmemh ("Instruction\_memory.txt", <memory register identifiers>)
  + Public test cases (project\_public\_test\_cases.s) and the Instruction\_memory.txt for those test cases are included in the “Project” folder.
    - File is generated by MIPS Helper and branch offset bug is fixed in this file
    - Note that these public test cases cover only a subset of dependency scenarios.
  + Before the demonstration you will be given the instruction memory ("**Instruction\_memory.txt**") with the private test cases.
  + You will generate the post-routing simulation using this instruction memory
  + **Waveform must show the following signals:** Reg\_writedata", pcresult, Hireg and Loreg as outputs to top module.
  + TAs will check your waveforms and will let you know about the failing cases.
  + Tams will have only one chance to re-demo with 75% of the lost points to recover for each case.
  + We will use public test-cases only if most of the private test cases fail.
  + Each demo will be conducted for 20 mins. You will have to select a slot for the demo. Please select one and insert comment on the demo slot which is suitable for all of your team members. Slots are available on first come first serve basis. Put your "d2l GROUP NUMBER" as comment in your slot. We will update sheet regularly.  Avoid selecting already booked slot. Slot sheet: <https://docs.google.com/spreadsheets/d/1fhdYJUleE97OxbHKipPMck0GMNAVKy22rZ5QSEnlXYo/edit?ts=59ff665f#gid=0>
  + All team members are required to attend the demo.
  + During the demo, **TA will log onto your system and download the group submission** from d2l. Then TA will take over the control of the system and will copy the instruction and data memory file from his/her flashdrive into newly created project, and will start the synthesis and implementation.
  + While the implementation is on-going, TA will ask questions about your implementation and details of pipelining to the team members and will take notes.
  + Once post-implementation simulation is ready, TA will check their waveform
  + After checking waveforms TA will explain what kind of forwarding or hazards are not taken care by the teams, if there is any. TA won’t be sharing actual testcases.
* **Offline Testing:**
  + Functionality of individual instructions on the pipelined datapath will be tested offline based on post-routing simulation. Your datapath should still be able to execute all the instructions.
* **Deliverable:**
  + Submit the following files (**deadline: before your scheduled demo slot)**
    - All verilog files: "\*.v"
  + Include the following notes under comment section during your submission
    - % effort
    - Number of pipeline stages:
      * We accept both four and five stage based pipelined datapaths
      * Indicate the number of pipeline stages in your design.
    - Branch decision and resolution stage:
      * Indicate the stage (DE or EX or MEM) where you are making branch decision.
    - If you don't include any note, we will **assume** that it is a 5-stage pipeline and branches are resolved during ID stage during the offline testing.
* **Penalty Conditions:**
  + private test cases on the pipelined datapath (2000 points)
    - If majority of the private test cases fail, then you can use public test cases for a maximum score of 800 points
  + Percent effort not reported (250 pt penalty)
  + Late submission (10% per day)
  + Submitting files in a folder or in compressed form (zip/tar). (20% penalty)
  + Changing the file name or extension. (20% penalty)
  + Failing to demonstrate (80% penalty)
  + Design works in behavioral simulation but fails to synthesize (70% penalty)
  + Design works in behavioral simulation, synthesizes with warnings but post-routing simulation fails (60% penalty)
  + Both team members must attend the demonstration

**Phase 2: (500 points)**

* **Objective:**
  + Execute your vbsme on the pipelined datapath (**due November 29 during your designated lab section**)
* **Eligibility**
  + At least 75% of the instructions are functional on the datapath
  + At least 75% of the private test cases for pipelining is passing
* **Deliverable:**
  + Submit the following files under “phase2” folder (**deadline: before your scheduled demo slot)**
    - All verilog files: "\*.v"
    - vbsme.s
* **Method**
  + A test case will be given at the beginning of the lab for your data memory
  + Prepare your instruction memory before the lab with your SAD routine
  + Prepare data memory with the new test case
  + Execute the program on the FPGA
  + The (X,Y) coordinates of the block of the current minimum SAD should be displayed on the FPGA
  + Display will start with (0,0) and each time a block with smaller SAD is found new coordinates should be displayed
  + Initialization part for the data memory will then be given to you by the **TA or ULA**
  + Set up behavioral simulation showing
    - **PC, Current minimum, X, and Y values**
  + TA will check both the FPGA and simulation
  + TA/ULA will collect your demonstration version of the datapath (.v) files and your vbsme.s
  + Offline validation to be conducted by TAs
* **Deliverable**:
  + Submit the following files (**before your designated lab section on Nov 29**)
    - All verilog files: "\*.v"
    - Data file used for initializing data and instruction memory (\*.txt)
    - .s (final form of your sad routine)
* **Penalty Conditions:**
  + Percent effort not reported (100 pt penalty)
  + Late submission (10% per day)
  + Submitting files in a folder or in compressed form (zip/tar). (20% penalty)
  + Changing the file name or extension. (20% penalty)
  + Failing to demonstrate (80% penalty)
  + Design works in behavioral simulation but fails to synthesize (70% penalty)
  + Design works in behavioral simulation, synthesizes with warnings but post-routing simulation fails (60% penalty)
  + Design works in post-routing simulation, but FPGA fails to display (25% penalty)
  + Both team members must attend the demonstration

**Phase 3: Competition (December 4, during your designated lab section)**

* Qualification for the competition
  + At least 75% of the instructions are functional on the single cycle datapath
  + Pipelined design passes at least 75% of the private test cases
  + vbsme executes on the pipelined datapath on the FPGA
* No pipelining demo will be taken.
  + Only competition
* Your files datapath (.v) and vbsme (.s) must be on D2L (**“phase3” folder**)
* You will then be given the initialization part for your data memory
  + Test case will be 64x64, 4x4
* Prepare behavioral simulation showing PC, Current minimum, X, and Y values
* Synthesize the design and prepare your post-routing resource utilization
* Program the FPGA
* Call your TA
  + Show that FPGA sequences through the coordinates and finally displays the correct X,Y coordinate
* Show your behavioral simulation waveform
  + TA will collect cycle count data (**A**): number of cycles it takes to run the program
* Show the post-routing data:
  + TA will collect Critical path delay(**B**) , BRAM usage(**C**), LUT usage(**D**), DSP usage(**E**)
* Results for A,B,C,D,E will be tabulated
  + Winners will be announced by the end of the last lab session.