ECE369A Computer Organization

Competition (Optional)
Due December 4 @12pm

Eligibility for the competition

- Not received two or more non-equal workload warning or
- Not marked off more than two times during demonstrations (with official warnings)
  - Also refer to lecture 1 slide # 18
- Lab 3: vbsme is running at least 75% of the private test cases correctly
- Labs 4-5: At least 75% of the instructions are functional on the datapath
- Lab6: Hazard detection passes at least 75% of the private test cases
- vbsme executes on the pipelined datapath on the FPGA
  - You are allowed make changes to the vbsme implementation through custom instructions or change of search pattern. However, the condition of the final minimum SAD results is still <= (less than or equal) than the current minimum SAD.
- Demonstration requires validation showing correct minimum SAD and the (X,Y) coordinate based on the post implementation simulation and showing the correct (X,Y) coordinates displayed on the FPGA. The correct (X,Y) coordinates correspond to the sub-block that results with the current minimum SAD value as the algorithm traverses the frame.

Potential Enhancement Ideas

- **Implement forwarding cases** needed for your vbsme.s code, show the number of cycles it takes your vbsme code to complete without forwarding and with forwarding.

- **Implement branch prediction** scheme and show the number of cycles it takes your vbsme code to complete with and without branch prediction

- **Implement custom instruction(s)**, and show the number of cycles saved before and after the custom instruction using your vbsme code.

- **Any other ideas:** Discussing with the instructor is highly recommended to make sure your design ideas are compatible for competition.

If your design is not the winner, there will be extra credit and the scale of the extra credit will depend on the complexity of the enhancement(s).

Preparing for Competition:

- It is **essential** to use Vivado 2022.2 to be able to conduct fair comparisons among the teams. Vivado version might impact the key performance metrics we will be collecting. If you are using a different Vivado version, you need to refer to **Lab 1 Resources** and make sure that you can verify your results on Vivado 2022.2 before the competition lab day.
- In your vbsme.s code, place an infinite loop at the end of the program (`exit: j exit`)
- Convert vbsme into hex/machine code, and initialize your instruction memory with that.
- The code will be tested for a single case of 64x64 frame and 4x4 window. Hence, if you are using the stack pointer, initialize the stack pointer register accordingly.
- When compiling your vbsme using MIPShelper, initialize a single test case of size 64x64, 4x4 using $readmemh
    - The test case input data for the competition will become available under Content > Labs > Lab8-Competition folder.
    - Both verilog and hex file formats of the 64x64 and 4x4 testcase will be available the day before the due date to implement your design and collect your performance metrics
        - In case your design involves memory level parallelization you will need to use the provided data and set up your memory structure for your design specific needs.
    - Please see files to be uploaded section below regarding the individual_group_results.xlsx file on how to collect performance related data.
- For fair critical path delay evaluation during competition, you **must** **REMOVE** the clk_div (clock divider) module from your design, and use the single generated clock for the entire design. Your critical path will be measured by looking at the path with the worst negative slack (WNS) in your Vivado implementation report (project summary). Make sure your board's display shows the v0 and v1 coordinates and stay constant after running the single test case. Without the clock divider it will run too fast to see changes on the display. You should stop execution by creating an **infinite loop (exit: j exit)** right after the test case.
    - **Critical path delay = Target Period - WNS.**
    - You are **NOT** allowed to use any synthesis/implementation strategies other than the Vivado synthesis/implementation defaults, and you are **NOT** allowed to make any modifications to the constraint file except the clock period, reset, x coordinate, y coordinate, and SAD connections.
    - Removing the clock divider from your design will probably lead to a failure in meeting a clock period of **10 ns**, if this is the case in your updated design, you need to **increase** your clock period **slightly (for example 11 ns, 12 ns, ... 20 ns, ...)** until you meet your timing target. Your current WNS can give you an idea about how much you should increase your clock period. **However,** increasing your target periods (if timing fails) **slightly** is very important to find the ideal and correct value of your critical path delay.

Files to be uploaded on D2L

Submit all necessary files to d2l before deadline in the fpga-ready configuration

- Submit instruction memory.mem, data memory.mem (of the competition test case), all design sources (.v), constraint file, and datapath testbench (that shows PC, WriteData , current minimum SAD and its coordinates) along with vbsme.s. Uploading your bitstream is also **recommended**, but optional.
- Submit competition design results in the provided file named “individual_group_results.xlsx”. **This file is available on D2L.** We will be DOUBLE-CHECKING these values during the demo. **All reported results must be after the implementation stages of your design, NOT synthesis.**

Measuring the performance

- Your program runtime will be equal to **# of simulation cycles * critical path delay**
    - **# of simulation cycles = end time of your program execution (when it enters the infinite loop)/simulation clock period**
    - **simulation clock period = 2 * clock delay (that you define in your datapath (or top module) testbench)**
    - Cycles are determined by the end of the vbsme routine, **not when the correct final value is displayed.** Lower is better!
    - **for example, if your testbench includes clk = ~clk; #20 then your simulation clock period is equal to 2*20 = 40 ns**
    - Your program runtime will be equal to the: **# (number) of simulation cycles * critical path delay**
    - Refer to the provided "individual_group_results.xlsx" file
- <u>**Note:**</u> You will have to reimplement & regenerate your bitstream files if we notice any **out-of-date** design messages in your Vivado project.

<u>**Demo Procedure:**</u>

- Only competition demos will be taken on December 4
- Before the demo, you should have your bitstream and post-implementation waveform for this lab test case set up and ready for the TAs to check. If not, it will be time consuming for the TAs to implement and generate the bitstream.
- TAs will check the minimum SAD and its coordinates on the post-implementation waveform and the correct (X,Y) coordinates on the FPGA board. Showing the SAD value on the board is optional
- The flow of the correct (X,Y) coordinates in successive iterations of the algorithm over the frame data must be displayed in the simulation waveform and the coordinate of the block that results with the minimum SAD should be displayed on the FPGA to be considered eligible for ranking in competition (by the end of program execution)
- TAs will be check your post-implementation resource utilization, WNS, and post-implementation simulation cycle counts, target clock period (in the constraint file) , and simulation clock period. These metrics will be compared with your "individual_group_results.xlsx" file.
- We will announce the winners within 24 hours. The number of competition categories will depend on the number of teams participating in the competition.
- TAs will collect information on technical contribution of each team member
    - Technical contribution is defined as having a direct impact on the performance obtained.
    - Activities such as writing a testbench, preparing test data or writing a test code are not qualified as technical contribution.