Arden Diakhate-Palme (aqd)
Carlos Armendariz (carmenda)
Ankita Bhanjois (abhanjoi)

Lab 4: REPORT

1. **Describe your design in terms of organization & operation**
    a. We've taken the Lab 3 design and changed the IF stage to consider only one of the words fetched from I-MEM and associate the instruction with its corresponding PC. For the MEM stage, in order to account for the D-MEM access's 4 cycle latency, when we send an address *(addr)* to memory, we send a new address *(consecutive from addr)* to memory each cycle until the first access returns data. We then add this to the cache and the 4 other word pairs to the cache (once they return data). Once the requested data is made available, we cache it in a direct-mapped cache (B = 8, C = 2K). We stall IF, ID, EX and MEM until the two data words from the requested address appear from main memory *(data_load)* and are placed in the cache. So we are essentially stalling in the four mentioned stages until the cache hits. This stalling is only necessary to resolve RAW hazard where LW instructions are the producer. Due to the 1 cycle access latency of I-MEM, a branch misprediction incurs a misprediction penalty of 3. The rest of the processor design is comparable to our Lab 3 implementation. Note that from status check we abandoned this idea of adding an extra IF stage and the idea of having 4 additional MEM stages, and used stalling to compensate.
2. **Describe the critical path of your synthesized implementation. (Is it different from what you expected?) How does the clock frequency compare to your Lab 3 implementation?**
    a. The critical path of our implementation goes from the IF/ID pipeline registers to the multiplier to the EX/MEM pipeline register. Our clock frequency in Lab 4 is 4.33 ns which is slightly greater than what it was in Lab 3 (4.28 ns).
3. **Any special tricks you played that may not be obvious to the course staff?**
    a. We have expanded our forwarding logic to only forward a byte on an LB instruction or forward a half-word on a LH instruction. This prevents unnecessary bytes to be forwarded.
4. **Estimate the number of hours spent on planning the design in your mind or on paper**
    a. Planning the design in your mind or on paper: **6hrs**
    b. Capturing the design (that is, actually hacking in SystemVerilog): **24hrs**
    c. Testing the design (not including debug time): **12hrs**
    d. Debugging the design (actually fixing the bugs once found): **15hrs**

e. Analyzing the performance of the design (next item): **~1hr**

f. Net total of hours worked on the lab in general: ~60hrs