Verilog
Design of Single-Cycle RISC-V Processor

1-Design a single-cycle processor datapath and control logic.

2-Run and debug logic simulations.

3-Your processor should implement the following subset of RV32I instruction set:

a. **R-Type**: add, sub, and, or
b. **I-Type**: addi, andi, ori, lw, jalr
c. **B-Type**: beq, bne
d. **J-Type**: jal
e. **S-Type**: sw

4-Use 1 ns clock period in your testbench.

5-Note that reading register-file/memory data is combinational, but writing is clocked.

6-The top module should be divided into two main modules: datapath and control logic. The Instruction and data memories should be connected to the top module in the testbench.

7-For simplicity, assume that the instruction memory is a 1kB word-addressable ROM and the data memory is a 1kB word-addressable RAM, i.e., you will only connect bits 9 to 2 in the address bus.

8-Use Venus to generate the machine code for an arbitrary sample program.

9-Then implement a simple cache system and integrate it with the RISC-V processor.

10-Re run your assembly code and compare simulation

Hint: **Read Section 7.3 Single-Cycle Processor in the textbook (DDCA by Harris and Harris) as it will be a good starting point**.

Reference Cache-controller:

1- https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/pages/c14/

2- https://slidetodoc.com/cache-performance-metrics-miss-rate-fraction-of-memory-2/

وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا