THE UNIVERSITY OF KANSAS

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

EECS 645 – Computer Architecture

Fall 2020

Homework 05 (MIPS Data Memory and Instruction Memory)

Student Name: Student ID:

MIPS Data Memory and Instruction Memory

Describe in behavioral VHDL both MIPS data memory and MIPS instruction memory as shown in Figures 1 and 2 with the following interface:

I. Data Memory (Single-Ported Read/Write memory)

- Generics
  - Data memory size/depth (*data_mem_depth* with default value of **117** locations)
  - Data memory width (*data_mem_width* with default value of 32 bits)
- Inputs
  - Clock (clk → 1 bit)
  - Asynchronous reset (rst → 1 bit)
  - Memory write enable (MemWrite → 1 bit)
    - 1 → enable writing to data memory,
    - 0 → disable writing to data memory
  - Address for read/write operand (A → *n_bits_address* bits = 32 bits)
  - Write Data for write operand (WD → *data_mem_width* bits)
- Outputs
  - Read Data for read operand (RD → *data_mem_width* bits)

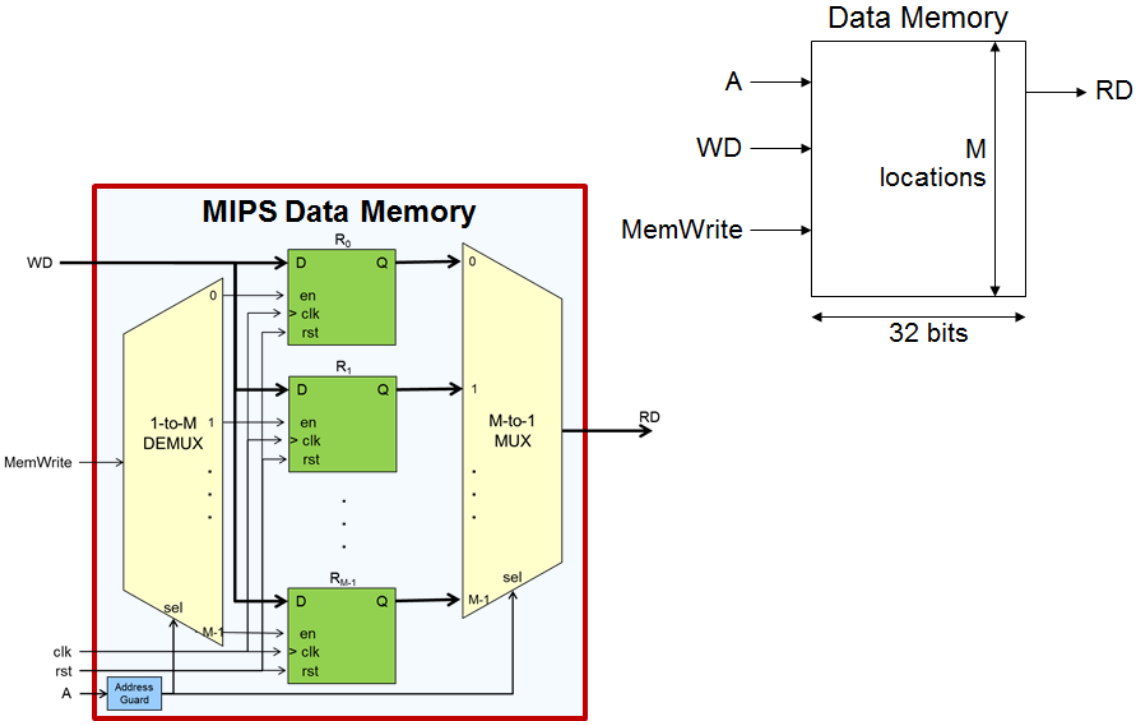

Figure 1. MIPS Data Memory

II. Instruction Memory (Single-Ported Read-Only Memory)

- Generics
  - Instruction memory size/depth (*instr_mem_depth* with default value of **39** locations)
  - Instruction memory width (*instr_mem_width* with default value of 32 bits)
- Inputs
  - Asynchronous reset for mimicking program load (rst → 1 bit)
  - Address for instruction read (A → *n_bits_address* bits = 32 bits)
- Outputs
  - Instruction fetched (Instr → *instr_mem_width* bits)

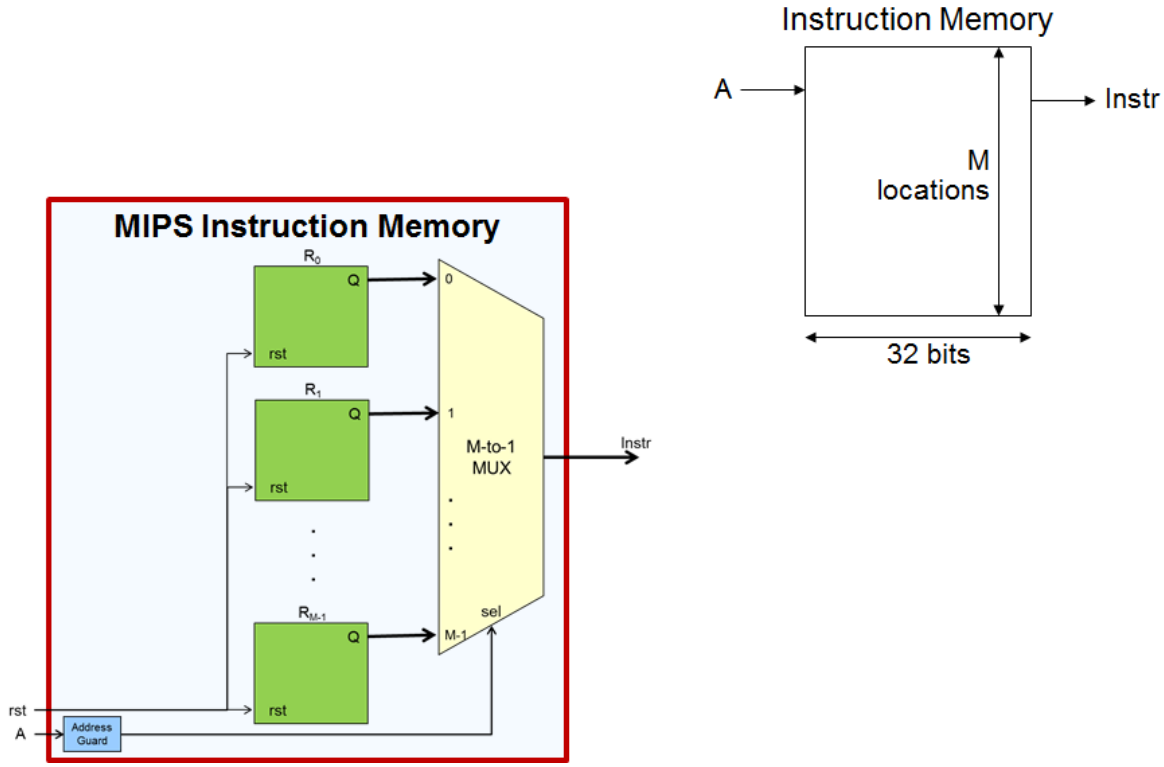

Figure 2. MIPS Instruction Memory

In your design:

- **Guard against address-spilling (address-overflow) where addresses could go out of range, e.g., exceeding the data/instruction memory size/depth.**

- In Vivado
  - Create a blank project
  - Add design and simulation source files
  - Run behavioral simulation
  - Your waveform configuration should be identical to the provided waveform snapshot, see Figure 2.
- Steps:
  1) Download the file "HW05_MIPS_MEM.zip" from blackboard and extract its contents.
  2) Rename the folder "HW05_MIPS_MEM" to "HW05_MIPS_MEM_<your last name>", for example "HW05_MIPS_MEM_El-Araby".
  3) Data memory
     a. Launch Vivado and create a new project, for example "vivado_project", with the default settings under the following directory "\HW05_MIPS_MEM_<your last name>\DataMem" resulting in the following project directory "\HW05_MIPS_MEM_<your last name>\DataMem\vivado_project\"
     b. Add to the project the VHDL design and simulation source files from the folders; "\HW05_MIPS_MEM_<your last name>\DataMem\design_sources" and "\HW05_MIPS_MEM_<your last name>\DataMem\simulation_sources" respectively.
     c. Edit the VHDL file in the folder "\HW05_MIPS_MEM_<your last name>\DataMem\design_sources\" according to your design such that it describes the required *MIPS data memory*.
  4) Instruction memory
     a. Launch Vivado and create a new project, for example "vivado_project", with the default settings under the following directory "\HW05_MIPS_MEM_<your last name>\InstrMem" resulting in the following project directory "\HW05_MIPS_MEM_<your last name>\InstrMem\vivado_project\"
     b. Add to the project the VHDL design and simulation source files from the folders; "\HW05_MIPS_MEM_<your last name>\InstrMem\design_sources" and "\HW05_MIPS_MEM_<your last name>\InstrMem\simulation_sources" respectively.
     c. Edit the VHDL file in the folder "\HW05_MIPS_MEM_<your last name>\InstrMem\design_sources\" according to your design such that it describes the required *MIPS instruction memory*.
  5) Set the simulation time to the proper time, e.g., ***300 ns***, and then launch Vivado Simulator.
  6) Verify the correctness of your design. Your waveform configuration should be identical to the waveform snapshot shown in Figures 3 and 4. You may go back to step 3 and/or 4 to correct your code until your design works properly as required.
  7) After you are done, compress the folder "\HW05_MIPS_MEM_<your last name>" to "HW05_MIPS_MEM_<your last name>.zip", for example "HW05_MIPS_MEM_El-Araby.zip" and upload it to blackboard before the due date and time.

Grade Distribution:

- Functional Correctness, i.e. correct source code → 75 / 100
- Proper Setup of Vivado Project → 25 / 100

NOTE:

Homework submission is a "Single Attempt", i.e., carefully review everything that you want to submit before hitting the "submit" button and make sure that you have uploaded all documents you want to submit and have not missed anything.

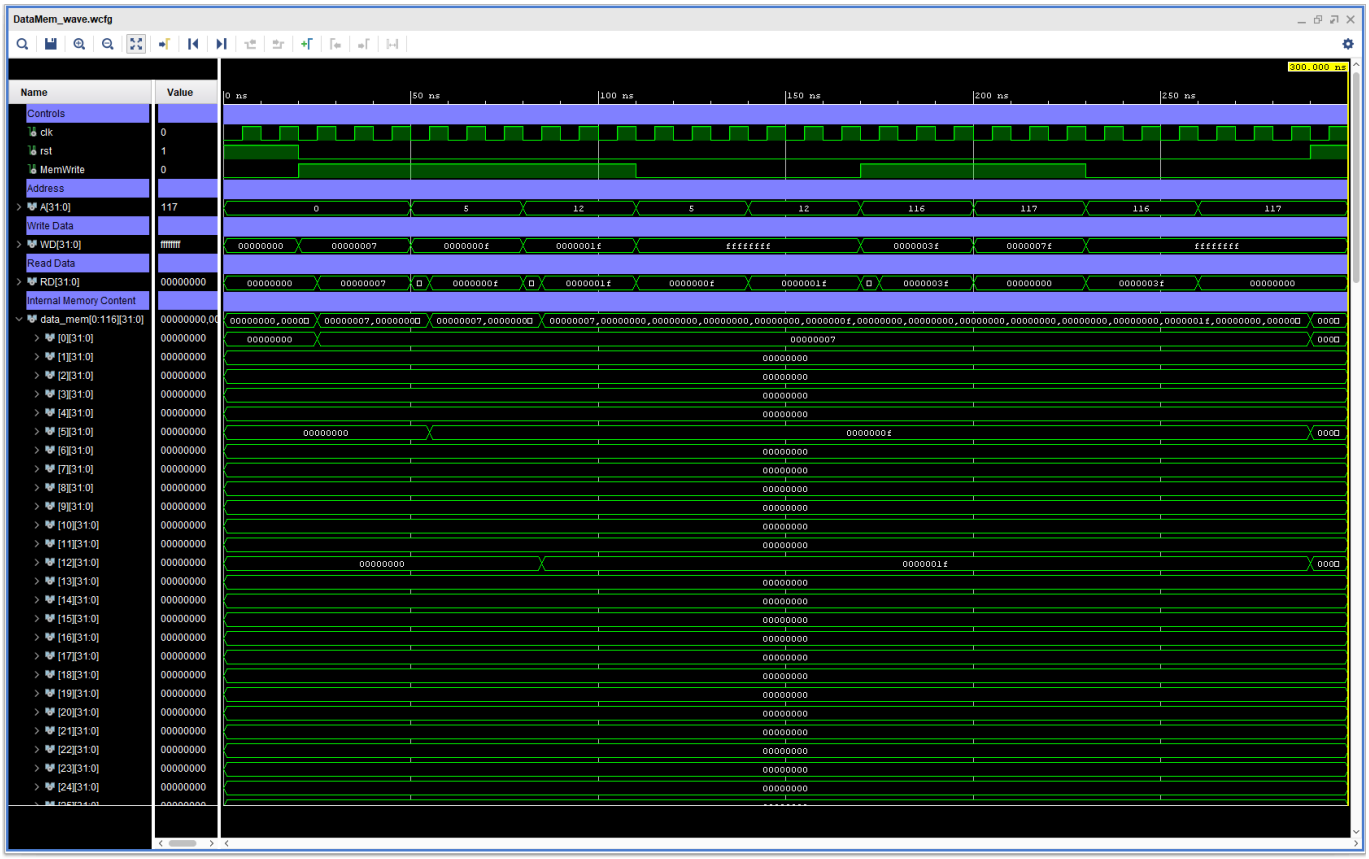

Figure 3. Snapshot of Correct Waveform Configuration for MIPS Data Memory

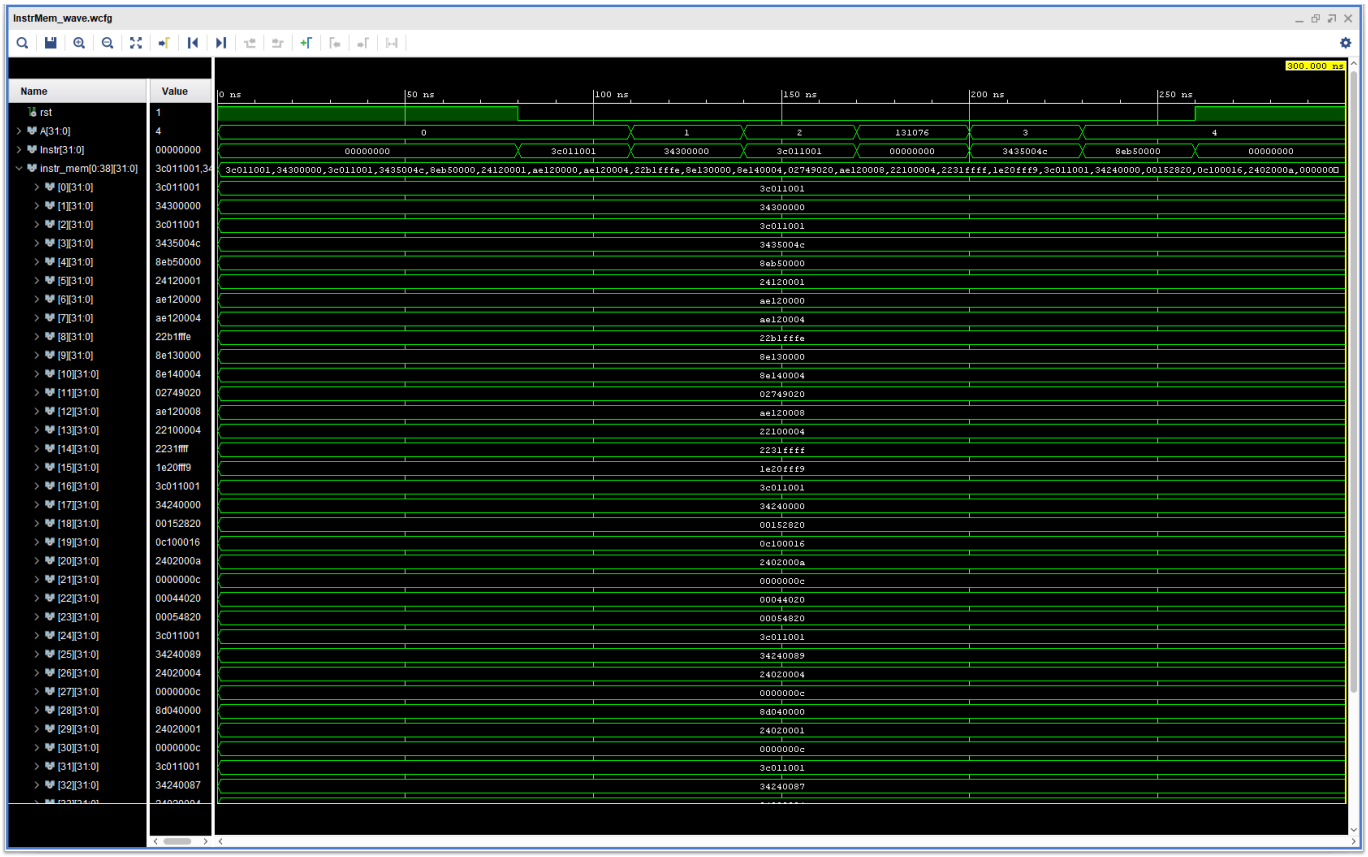

Figure 4. Snapshot of Correct Waveform Configuration for MIPS Instruction Memory