



**THE UNIVERSITY OF KANSAS**

**SCHOOL OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

EECS 645 – Computer Architecture

Spring 2023

Homework 08 (MIPS CU)

Student Name:

Student ID:

---

## MIPS Control Unit (CU)

Describe in **behavioral VHDL** the MIPS Control Unit (CU) that supports the following subset, i.e., 11 instructions, of MIPS ISA:

- 7 Arithmetic/Logical instructions: *add*, *sub*, *and*, *or*, *nor*, *slt*, *addi*
- 2 Memory reference: *lw*, *sw*
- 2 Control transfer: *beq*, *j*

The functionality, interface, and structure of the required MIPS CU are shown in Figure 1. The CU has the following interface:

- Generics
  - Instruction width ( $n\_bits\_instr = instr\_mem\_width$  with default value of 32)
  - Location of the operation (OP) code least significant bit in the instruction (*opcode\_start* with default value of 26)
  - Location of the operation (OP) code most significant bit in the instruction (*opcode\_end* with default value of 31)
  - Location of the function (funct) code least significant bit in the instruction (*funct\_start* with default value of 0)
  - Location of the function (funct) code most significant bit in the instruction (*funct\_end* with default value of 5)
  - Total number of supported ALU operations/functions ( $n\_functions\_alu$  with default value of 16)
- Inputs
  - Instruction fetched (Instr  $\rightarrow n\_bits\_instr$  bits)
- Outputs
  - Register destination (RegDst  $\rightarrow$  1 bit)
  - Register write enable (RegWrite  $\rightarrow$  1 bit)
  - ALU source (ALUSrc  $\rightarrow$  1 bit)
  - ALU control (ALUControl  $\rightarrow \lceil \log_2(n\_functions\_alu) \rceil$  bits)
  - Memory write enable (MemWrite  $\rightarrow$  1 bit)
  - Memory read enable (MemRead  $\rightarrow$  1 bit)
  - Memory to register (MemToReg  $\rightarrow$  1 bit)
  - Conditional branch on equal (BEQ  $\rightarrow$  1 bit)
  - Unconditional branch (J  $\rightarrow$  1 bit)

opcode	funct	ALUControl	RegDst	ALUSrc	MemToReg	RegWr	MemWr	BEQ	J
<b>R-type</b> $\equiv$ 000000	<b>AND</b> 100100	<b>0000</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>
	<b>OR</b> 100101	<b>0001</b>							
	<b>add</b> 100000	<b>0010</b>							
	<b>sub</b> 100010	<b>0110</b>							
	<b>slt</b> 101010	<b>0111</b>							
	<b>NOR</b> 100111	<b>1100</b>							
<b>lw</b> $\equiv$ 100011	xxxxxx	<b>0010</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>sw</b> $\equiv$ 101011	xxxxxx	<b>0010</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>
<b>beq</b> $\equiv$ 000100	xxxxxx	<b>0110</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>
<b>j</b> $\equiv$ 000010	xxxxxx	<b>0000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>
<b>addi</b> $\equiv$ 001000	xxxxxx	<b>0010</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>

(1-a) Functionality

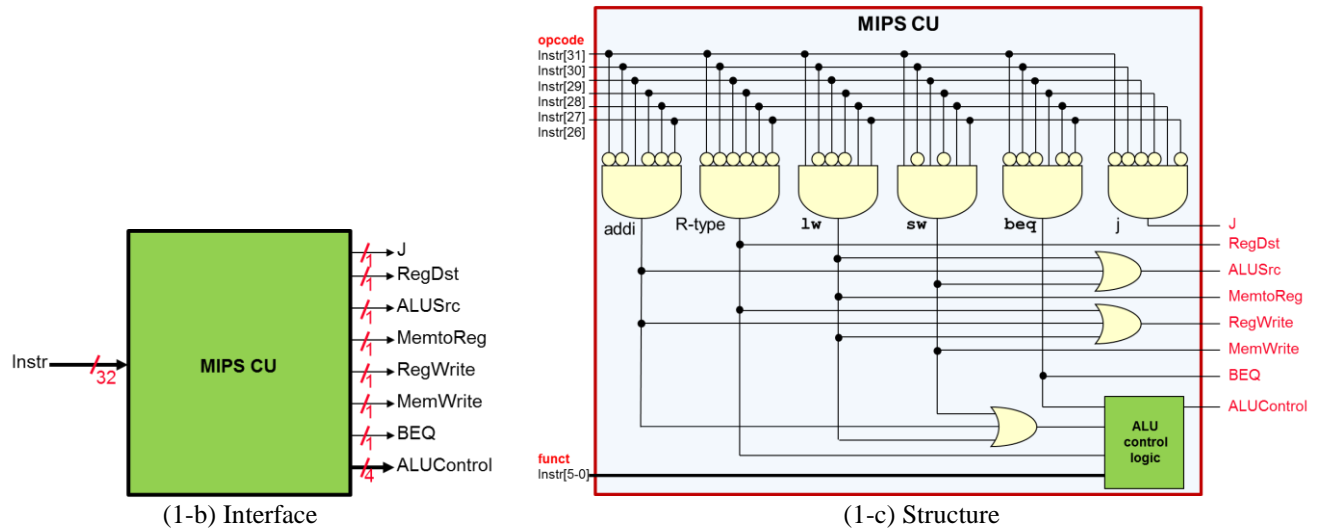


Figure 1. MIPS CU

- 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 “HW08\_MIPS\_CU.zip” from Canvas and extract its contents.
  - 2) Rename the folder “HW08\_MIPS\_CU” to “HW08\_MIPS\_CU\_<your last name>”, for example “HW08\_MIPS\_CU\_El-Araby”.
  - 3) Launch Vivado and create a new project, for example “vivado\_project”, with the default settings under the following directory “\HW08\_MIPS\_CU\_<your last name>” resulting in the following project directory “\HW08\_MIPS\_CU\_<your last name>\vivado\_project”
  - 4) Add to the project the VHDL design and simulation source files from the folders; “\HW08\_MIPS\_CU\_<your last name>\design\_sources” and “\HW08\_MIPS\_CU\_<your last name>\simulation\_sources” respectively.
  - 5) Edit the VHDL file in the folder “\HW08\_MIPS\_CU\_<your last name>\design\_sources” according to your design such that it describes the required *MIPS CU*.
  - 6) Set the simulation time to the proper time, e.g., **400 ns**, and then launch Vivado Simulator.
  - 7) Verify the correctness of your design. Your waveform configuration should be identical to the waveform snapshot shown in Figure 2. You may go back to step 5 to correct your code until your design works properly as required.
  - 8) After you are done, compress the folder “HW08\_MIPS\_CU\_<your last name>” to “HW08\_MIPS\_CU\_<your last name>.zip”, for example “HW08\_MIPS\_CU\_El-Araby.zip” and upload it to Canvas 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.

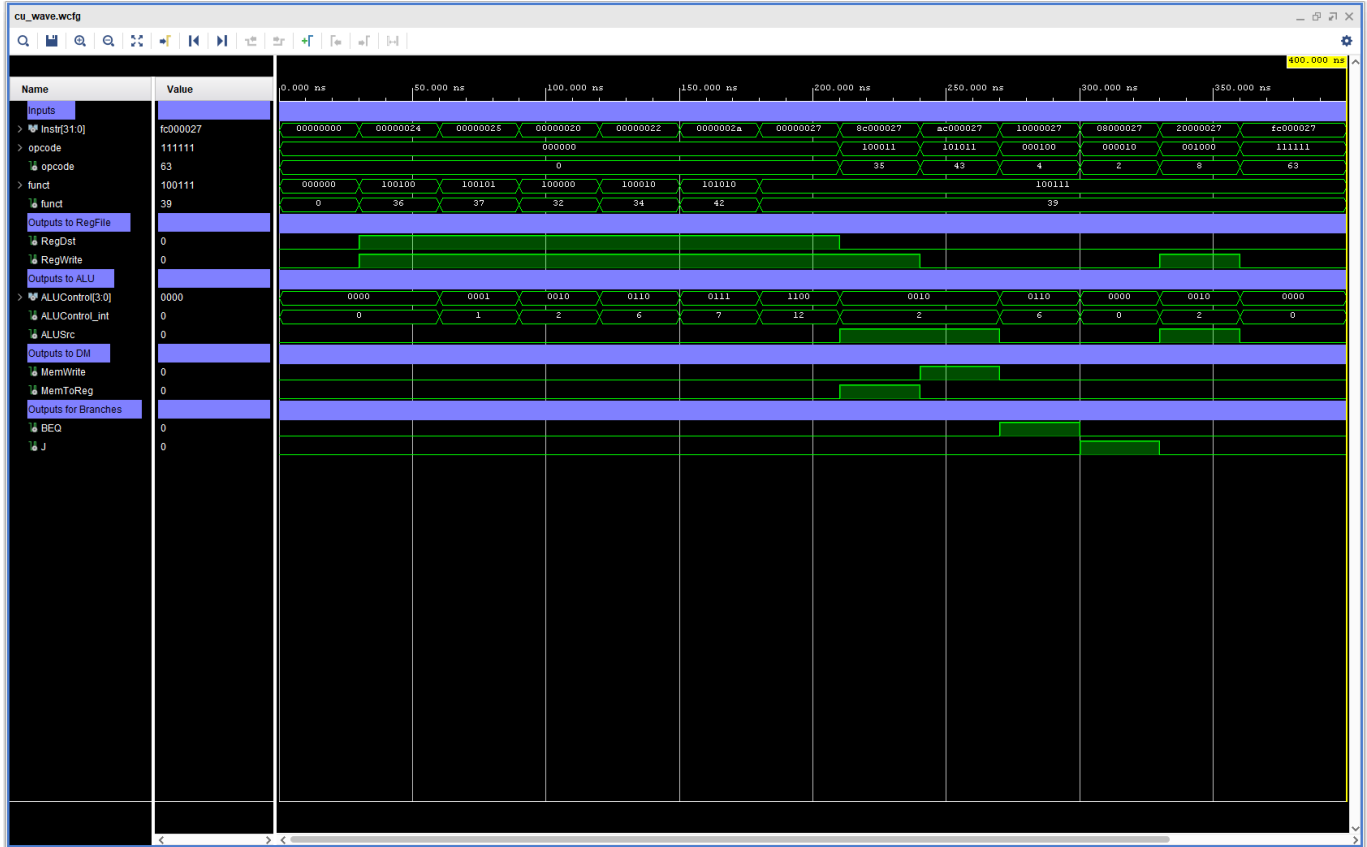


Figure 2. Snapshot of Correct Waveform Configuration for MIPS CU