THE UNIVERSITY OF KANSAS

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

EECS 645 – Computer Architecture

Fall 2017

Homework 09 (MIPS CU)

Student Name: Student ID:

MIPS Control Unit (CU)

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

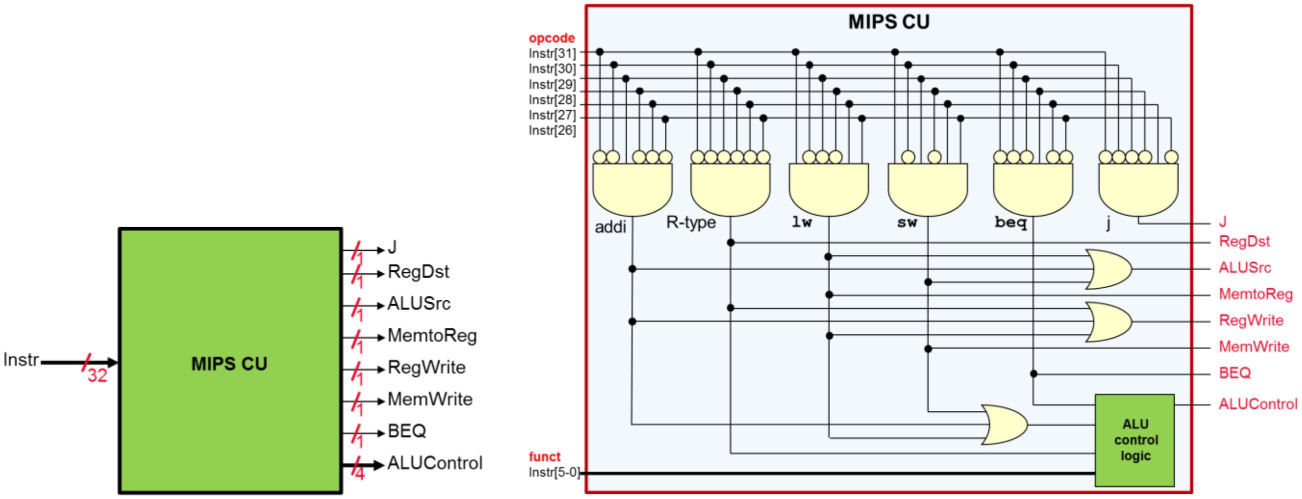
a) 7 Arithmetic/Logical instructions: *add, sub, and, or, nor, slt, addi*
b) 2 Memory reference: *lw, sw*
c) 2 Control transfer: *beq, j*

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 → *n_bits_instr* bits)
- Outputs
  - Register destination (RegDst → 1 bit)
  - Register write enable (RegWrite → 1 bit)
  - ALU source (ALUSrc → 1 bit)
  - ALU control (ALUControl → $\lceil \log_2(n\_functions\_alu) \rceil$ bits)
  - Memory write enable (MemWrite → 1 bit)
  - Memory read enable (MemRead → 1 bit)
  - Memory to register (MemToReg → 1 bit)
  - Conditional branch on equal (BEQ → 1 bit)
  - Unconditional branch (J → 1 bit)

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

(1-a) Functionality



(1-b) Interface

(1-c) Structure

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 “HW09_MIPS_CU.zip” from blackboard and extract its contents.
  2) Rename the folder “HW09_MIPS_CU” to “HW09_MIPS_CU_<your last name>”, for example “HW09_MIPS_CU_El-Araby”.
  3) Launch Vivado and create a new project, for example “vivado_project”, with the default settings under the following directory “\HW09_MIPS_CU_<your last name>” resulting in the following project directory “\HW09_MIPS_CU_<your last name>\vivado_project\”
  4) Add to the project the VHDL design and simulation source files from the folders; “\HW09_MIPS_CU_<your last name>\design_sources” and “\HW09_MIPS_CU_<your last name>\simulation_sources” respectively.
  5) Edit the VHDL file in the folder “\HW09_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 “\HW09_MIPS_CU_<your last name>” to “HW09_MIPS_CU_<your last name>.zip”, for example “HW09_MIPS_CU_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.

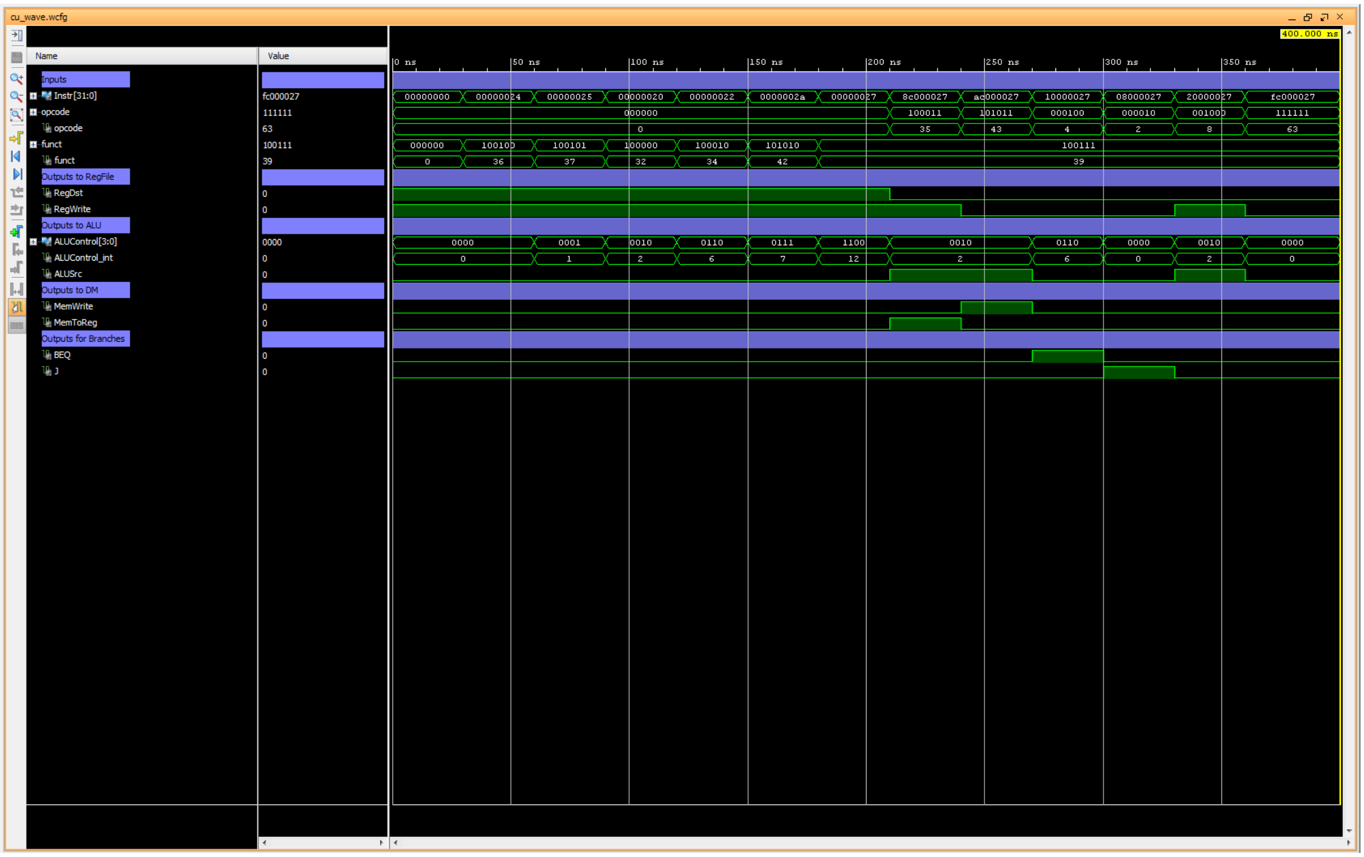

cu_wave.wcfg
Name
Value
400.000 ns
0 ns
50 ns
100 ns
150 ns
200 ns
250 ns
300 ns
350 ns
Inputs
Instr[31:0] fc000027 00000000 00000024 00000025 00000020 00000022 0000002a 00000027 8c000027 ac000027 10000027 08000027 20000027 fc000027
opcode 111111 000000 100011 101011 000100 000010 001000 111111
opcode 63 0 35 43 4 2 8 63
funct 100111 000000 100100 100101 100000 100010 101010 100111
funct 39 0 36 37 32 34 42 39
Outputs to RegFile
RegDst 0
RegWrite 0
Outputs to ALU
ALUControl[3:0] 0000 0000 0001 0010 0110 0111 1100 0010 0110 0000 0010 0000
ALUControl_int 0 0 1 2 6 7 12 2 6 0 2 0
ALUSrc 0
Outputs to DM
MemWrite 0
MemToReg 0
Outputs for Branches
BEQ 0
J 0

Figure 2. Snapshot of Correct Waveform Configuration for MIPS CU