THE UNIVERSITY OF KANSAS

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

EECS 645 – Computer Architecture

Spring 2023

Homework 05 (MIPS ALU)

Student Name: Student ID:

MIPS ALU

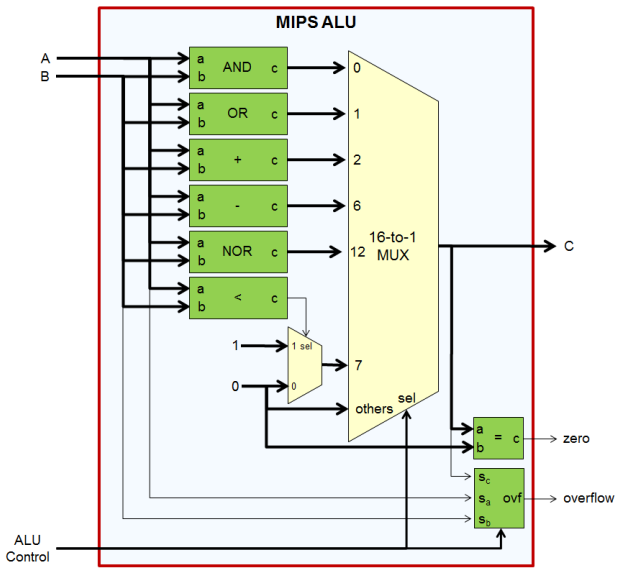
Describe in behavioral VHDL (using 2 concurrent VHDL processes) the MIPS ALU supporting signed arithmetic and up to 16 operations/functions, as shown in Figure 1. The ALU has the following interface:

- Generics
  - ALU data bits (*n_bits_alu* with default value of 32 bits)
  - Total number of supported operations/functions (*n_functions_alu* with default value of 16 functions)
- Inputs
  - First input operand (A → *n_bits_alu* bits = 32 bits)
  - Second input operand (B → *n_bits_alu* bits = 32 bits)
  - ALU control (ALUControl → $\lceil \log_2(n\_functions\_alu) \rceil$ bits)
- Outputs
  - Result output operand (C → *n_bits_alu* bits = 32 bits)
  - Zero flag (zero → 1 bit)
  - Overflow flag (overflow → 1 bit)

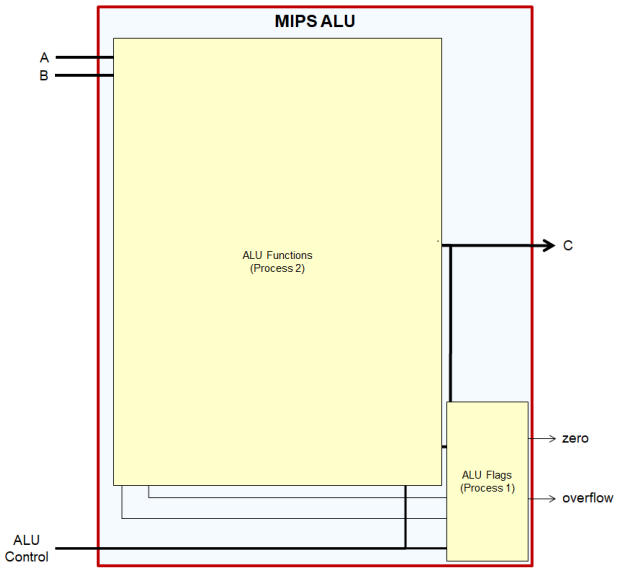
- ALU used for
  - R-type: ALU Function depends on funct field
  - Load/Store: ALU Function = add
  - Branch: ALU Function = subtract
- Operations/Functions
  - Up to 16 operations (we will implement only 6)
  - Signed / Unsigned arithmetic (we will implement only signed)
- Combinational circuit

ALU Control		Operation/Function		Zero Flag	Overflow Flag
0	0000	and	C = A and B	zero = 1 when ((A op B) = 0) else 0	overflow = 1 when ((A op B) is out of C range) else 0
1	0001	or	C = A or B		
2	0010	add	C = A + B		
6	0110	sub	C = A - B		
7	0111	slt	C = 1 when (A < B) else 0		
12	1100	nor	C = A nor B		
otherwise			C = 0		

(1-a) Specification & Interface



(1-b) Architecture



(1-c) VHDL Processes

Figure 1. MIPS ALU

- 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_ALU.zip" from Canvas and extract its contents.
  2) Rename the folder "HW05_MIPS_ALU" to "HW05_MIPS_ALU_<your last name>", for example "HW05_MIPS_ALU_El-Araby".
  3) Launch Vivado and create a new project, for example "vivado_project", with the default settings under the following directory "\HW05_MIPS_ALU_<your last name>" resulting in the following project directory "\HW05_MIPS_ALU_<your last name>\vivado_project\"
  4) Add to the project the VHDL design and simulation source files from the folders; "\HW05_MIPS_ALU_<your last name>\design_sources" and "\HW05_MIPS_ALU_<your last name>\simulation_sources" respectively.
  5) Edit the VHDL file in the folder "\HW05_MIPS_ALU_<your last name>\design_sources\" according to your design such that it describes the required *MIPS ALU*.
  6) Set the simulation time to the proper time, e.g., ***1,100 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 "\HW05_MIPS_ALU_<your last name>" to "HW05_MIPS_ALU_<your last name>.zip", for example "HW05_MIPS_ALU_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.

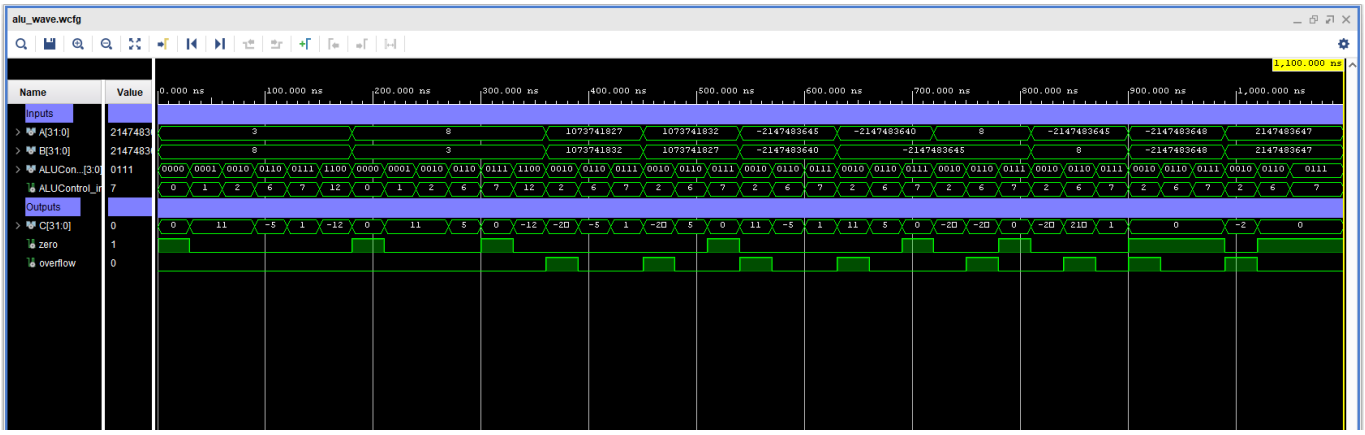


Figure 2. Snapshot of Correct Waveform Configuration for MIPS ALU