Name: Ruoyan Wu RCS ID: wur7 @rpi.edu

△ CSCI 2500 — Computer Organization △
Fall 2018 Quiz 5 (November 9, 2018)

Please silence and put away all laptops, notes, books, phones, electronic devices, etc. This quiz is designed to take 25 minutes; therefore, for 50% extra time, the expected time is 38 minutes and 100% extra time is 50 minutes. Questions will not be answered except when there is a glaring mistake or ambiguity in a question. Please do your best to interpret and answer each question.

1. **(16 POINTS)** Convert each of the 16-bit binary values shown below into its equivalent hexadecimal and decimal values (i.e., convert base 2 to base 16 and base 10). Assume these are two's complement signed integers.

   (a) 0001 0000 0001 1011_{two} ==> 0x 101B $_{hex}$ ==> 4123 $_{ten}$ (1+2+8+16)

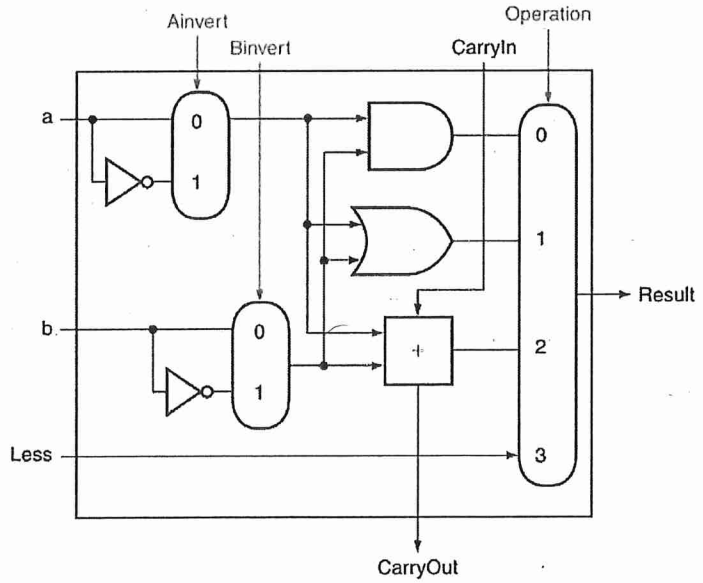
   (b) 0010 0000 0010 1010_{two} ==> 0x 202A $_{hex}$ ==> 8234 $_{ten}$ (32, 8, 2)

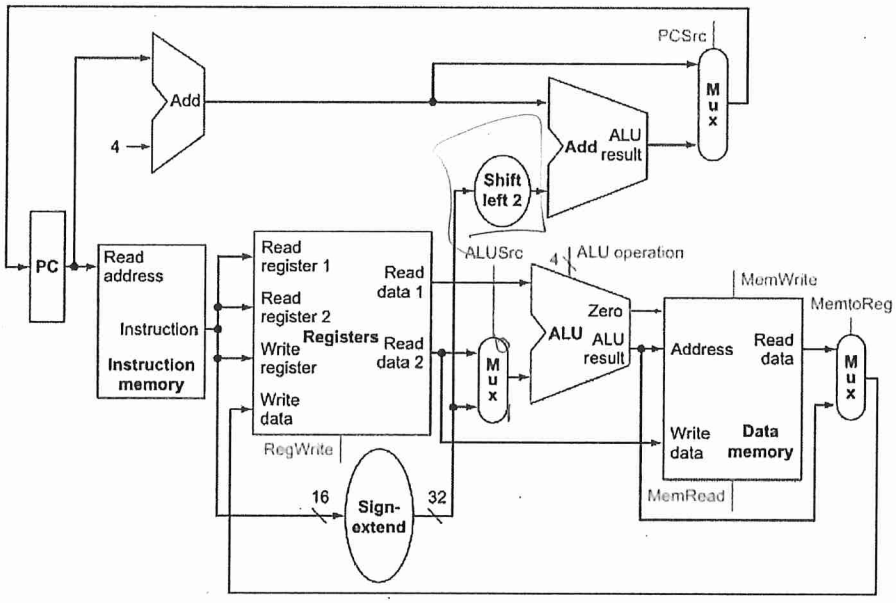
   (c) 1111 0111 1111 1110_{two} ==> 0x F7FE $_{hex}$ ==> -2050 $_{ten}$

   (d) 1111 1111 1111 1111_{two} ==> 0x FFFF $_{hex}$ ==> -1 $_{ten}$

2. **(24 POINTS)** Given the 1-bit ALU shown below, fill in the table by specifying the inputs and control lines for each given operation. More specifically, each cell of the table should contain one or more bits, with each bit specified as 0, 1, or δ for a "don't care" input (i.e., the bit does not affect the instruction). Note that inputs a and b are both 1-bit inputs.

Desired 1-Bit Result	Less	Ainvert	Binvert	CarryIn	Operation
a AND b	δ	0	0	δ	00 → 0
a OR b	δ	0	0	δ	01 → 1
a + b	δ	0	0	0	10 → 2
a - b	δ	0	1	0	10 → 2





3. **(30 POINTS)** What is the significance of the hard-wired **Shift left 2** step of the MIPS architecture diagram above? Describe the type(s) of instructions this would be applicable to.

The shift left 2 is for multiply constant by 4 to get offset. and is a re-routes wires. The shift left 2 here enable large branch offset.

Instructions this would be applicable to: beq.

4. **(30 POINTS)** The `ALUSrc` control bit feeds into a multiplexor (**Mux**) between **Registers** and the **ALU**. Describe the dataflow and an example instruction when `ALUSrc` is 0. Next, describe the dataflow and an example instruction when `ALUSrc` is 1.

When ALUSrc is 0 means select the sign-extended 16 bit offset. When ALUSrc is 1 means, select the second register file output.

Should be a 1 for load and store. and 0 for everything else.