

Solution for Tutorial 3, CSSE3001/7000

5.2

- a. RegWrite = 0: All R-format instructions, in addition to lw, will not work because these instructions will not be able to write their results to the register file.
- b. ALUop1 = 0: All R-format instructions except subtract will not work correctly because the ALU will perform subtract instead of the required ALU operation.
- c. ALUop0 = 0: beq instruction will not work because the ALU will perform addition instead of subtraction (see Figure 5.12), so the branch outcome may be wrong.
- d. Branch (or PCSrc) = 0: beq will not execute correctly. The branch instruction will always be not taken even when it should be taken.
- e. MemRead = 0: lw will not execute correctly because it will not be able to read data from memory.
- f. MemWrite = 0: sw will not work correctly because it will not be able to write to the data memory.

5.3

- a. RegWrite = 1: sw and beq should not write results to the register file. sw (beq) will overwrite a random register with either the store address (branch target) or random data from the memory data read port.
- b. ALUop0 = 1: lw and sw will not work correctly because they will perform subtraction instead of the addition necessary for address calculation.
- c. ALUop1 = 1: lw and sw will not work correctly. lw and sw will perform a random operation depending on the least significant bits of the address field instead of addition operation necessary for address calculation.
- d. Branch = 1: Instructions other than branches (beq) will not work correctly if the ALU Zero signal is raised. An R-format instruction that produces zero output will branch to a random address determined by its least significant 16 bits.
- e. MemRead = 1: All instructions will work correctly. (Data memory is always read, but memory data is never written to the register file except in the case of lw.)
- f. MemWrite = 1: Only sw will work correctly. The rest of instructions will store their results in the data memory, while they should not.

5.8

A modification to the datapath is necessary to allow the new PC to come from a register (Read data 1 port), and a new signal (e.g., JumpReg) to control it through a multiplexor as shown in Figure 5.42.

A new line should be added to the truth table in Figure 5.18 on page 308 to implement the jr instruction and a new column to produce the JumpReg signal.

5.9

A modification to the data path is necessary (see Figure 5.43) to feed the shamt field (instruction[10:6]) to the ALU in order to determine the shift amount. The instruction is in R-Format and is controlled according to the first line in Figure 5.18 on page 308.

The ALU will identify the sll operation by the ALUop field.

Figure 5.13 on page 302 should be modified to recognize the opcode of sll: the third line should be changed to 1X1X0000 0010 (to discriminate the add and sll functions), and a new line, inserted, for example, 1X0X0000 0011 (to define sll by the 0011 operation code).