

# 605.611 - Foundations of Computer Architecture

## Assignment 07 - MIPS Datapath

Sabbir Ahmed

March 14, 2021

1. Consider the following instruction:

Instruction: `AND Rd, Rs, Rt`

Interpretation: `Reg[Rd] = Reg[Rs] AND Reg[Rt]`

- (a) What are the values of control signals generated by the control in Figure 4.2 for the above instruction?
  - (b) Which resources (blocks) perform a useful function for this instruction?
  - (c) Which resources (blocks) produce outputs, but their outputs are not used for this instruction? Which resources produce no outputs for this instruction?
2. When silicon chips are fabricated, defects in materials (e.g., silicon) and manufacturing errors can result in defective circuits. A very common defect is for one wire to affect the signal in another. This is called a cross-talk fault. A special class of cross-talk faults is when a signal is connected to a wire that has a constant logical value (e.g., a power supply wire). In this case we have a stuck-at-0 or a stuck-at-1 fault, and the affected signal always has a logical value of 0 or 1, respectively. The following problems refer to bit 0 of the Write Register input on the register file in Figure 4.24.
    - (a) Let us assume that processor testing is done by filling the PC, registers, and data and instruction memories with some values (you can choose which values), letting a single instruction execute, then reading the PC, memories, and registers. These values are then examined to determine if a particular fault is present. Can you design a test (values for PC, memories, and registers) that would determine if there is a stuck-at-0 fault on this signal?

- (b) Repeat 4.6.1 for a stuck-at-1 fault. Can you use a single test for both stuck-at-0 and stuck-at-1? If yes, explain how; if no, explain why not.
  - (c) If we know that the processor has a stuck-at-1 fault on this signal, is the processor still usable? To be usable, we must be able to convert any program that executes on a normal MIPS processor into a program that works on this processor. You can assume that there is enough free instruction memory and data memory to let you make the program longer and store additional data. Hint: the processor is usable if every instruction “broken” by this fault can be replaced with a sequence of “working” instructions that achieve the same effect.
  - (d) Repeat 4.6.1, but now the fault to test for is whether the “MemRead” control signal becomes 0 if RegDst control signal is 0, no fault otherwise.
  - (e) Repeat 4.6.4, but now the fault to test for is whether the “Jump” control signal becomes 0 if RegDst control signal is 0, no fault otherwise.
  - (f) Add unit in the upper right, the ALU result bit 0 and result bit 31 are stuck at 0 & 1. Note that this will always work correctly or always fail.
3. Program the microcode instruction, similar to the micro table we did in class, for an R instruction, I instruction, and `sll` instruction. Do the same for an `andi` instruction. What problems do you see for the instructions `sll` and `andi` instructions?