

## EE 457 Homework 5

### Redekopp

Name: \_\_\_\_\_

Score: \_\_\_\_\_ / 100\_

#### Single-Cycle CPU

The following exercises are taken from Hennessy and Patterson, CO&D 2<sup>nd</sup>, 3<sup>rd</sup>, and 4<sup>th</sup> Ed.

- 1.) (6 pts.) Review your class notes.
  - a. Is it required that the PC is an edge-sensitive register or can it be a level-sensitive latch? [**Must be edge-sensitive / Can be level-sensitive**]
  - b. Is it essential or is only desirable to have two READ ports and one WRITE port on the register file? [**Essential / Only Desirable but not essential**]
  - c. Is it required that the individual register in the register file are actually edge-sensitive registers or can they be level-sensitive latches? [**Must be edge-sensitive / Can be level-sensitive**]
  
- 2.) (**Exercise 5.2, 3rd Ed.**) (12 pts.) Consider the effect that a single stuck-at-0 fault (i.e. a defect during the manufacturing process causes that particular signal to be '0' regardless of the intended design) would have for the signals shown below, in the single-cycle datapath (Figure 1) Which instruction(s), if any, will not work correctly?

Place an 'X' in the entry in the table for instructions that may not work if the given signal is stuck-at-0.

Signal	R-Type	LW	SW	BEQ
<b>RegWrite</b>				
<b>ALUop1</b>				
<b>ALUop0</b>				
<b>Branch</b>				
<b>MemRead</b>				
<b>MemWrite</b>				

- 3.) (**Exercise 5.3, 3<sup>rd</sup> Ed.**) (12 pts.) Repeat the previous exercise but now assuming stuck-at-1 faults.

Place an 'X' in the entry in the table for instructions that may not work if the given signal is stuck-at-1.

Signal	R-Type	LW	SW	BEQ
<b>RegWrite</b>				
<b>ALUop1</b>				
<b>ALUop0</b>				
<b>Branch</b>				
<b>MemRead</b>				
<b>MemWrite</b>				

- 4.) (**Exercise 5.5 2<sup>nd</sup> Ed.**) (9 pts.) We wish to add the instruction *addi* (add immediate) to the single-cycle datapath. No datapath changes or new control signals are needed. Show the

necessary value of current control signals for the execution of the *addi*. If a control signal can be either 1 or 0, choose 0 as your answer.

	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUOp0
<b>addi</b>									

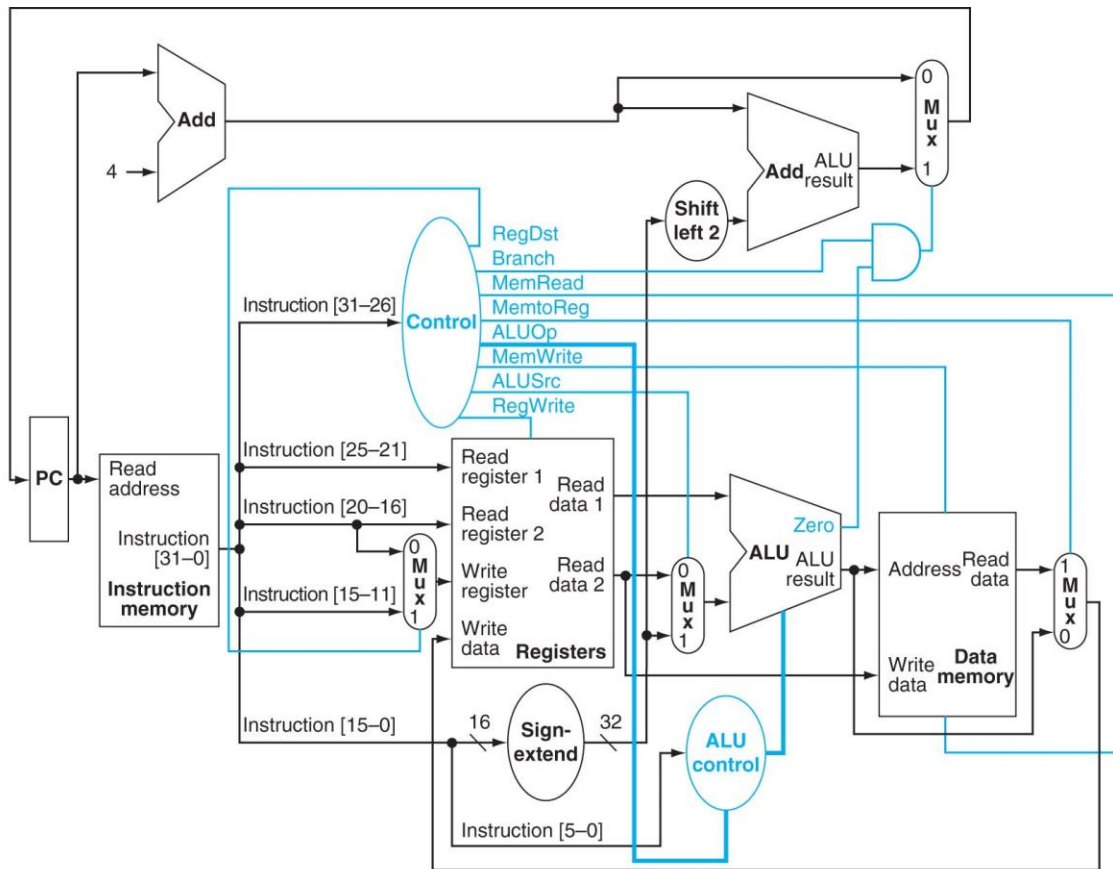


Figure 1 - Figure 4.17 (p. 322) 4th Ed.



6.) (Adapted from Exercise 5.13 and 3<sup>rd</sup> Ed.) (6 pts.) Examine the table of control signals (Figure 4.18 in the 4<sup>th</sup> Ed.) for the single-cycle datapath from Figure 4.17 in the 4<sup>th</sup> Ed. If we did NOT want to generate the control signal RegDst, which control signals (list all workable signals) could be used in its place to achieve equivalent operation? If we did not want to generate MemtoReg, which control signals could be used in its place (without modification) to achieve equivalent operation?

Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

Figure 3 - Figure 4.18 4th Ed.

Next question is on the following page.

Exercises 7.) - 9.) are drawn from Exercise 4.10 4<sup>th</sup> Ed. on page 4.18.  
 Consider only the times in row a.) of the tables and do not do problems for row b.

**Exercise 4.10** - In this exercise we examine how the clock cycle time of the processor affects the design of the control unit and vice versa. Problems in this exercise assume that the logic blocks used to implement the datapath have the following latencies (delays):

	I-Mem	ADD	Each Mux	ALU	Regs	D-Mem	Sign Extend	Shift-left-2	ALU Ctrl
a.	400 ps	100 ps	30 ps	120 ps	200 ps	350 ps	20 ps	0 ps	50 ps

7.) **Exercise 4.10.1 4<sup>th</sup> Ed. (p. 418)** (5 pts.) – To avoid lengthening the critical path of the datapath shown in Figure 4.24 (reprinted earlier in this HW but w/o any additional JAL logic), how much time can the control unit take to generate the MemWrite signal?

Note: You need to think about when the control unit can start generating MemWrite (hint: it's not at the beginning of the clock cycle) and when MemWrite is actually required. (Hint: since MemWrite is a write enable, it is needed at the end of the clock cycle...but you'll need to figure out the clock cycle by considering the longest delay of any instruction (i.e. consider all instructions: R-Type, lw, sw, beq, and j and their latencies; then pick the longest latency as your clock cycle time). You can find the time for the MemWrite by subtracting the clock cycle time minus the time when the control unit can start generating MemWrite.

Hint 1: Remember, we are treating MemWrite as a write enable. Thus it only has to arrive to the memory by the clock edge. The address and/or data is what is required for the 350ns to start.

Hint 2: When you find the clock cycle time for the longest instruction, be careful to consider muxes and when they are needed/used. If for a given instruction the mux will choose input 1, then consider when that value arrives. You need not be concerned with when input value 0 arrives since you aren't choosing it.

Hint 3: The given delay of the register file is for "reading". We assume that if data and write reg. # are present by the end of the clock cycle and regwrite=1 that writing happens "Immediately" (i.e. doesn't require an additional 350 ns).

	Time (in ps)
<b>Control Unit can start generating MemWrite</b>	
<b>Clock cycle time (longest path for any instruction)</b>	
<b>Time the control unit can take to generate MemWrite</b>	

