Alexandria University

Faculty of Engineering

Division of Communications & Electronics

CSx35 Computer Architecture
Sheet 2

---

1. Modify the single-cycle MIPS processor to implement one of the following instructions. Mark up a copy of Figure 1 to indicate the changes to the datapath. Name any new control signals. Mark up a copy of Table 1 to show the changes to the main decoder. Describe any other changes that are required.

(a) *sll*
(b) *lui*
(c) *slti*
(d) *blez*

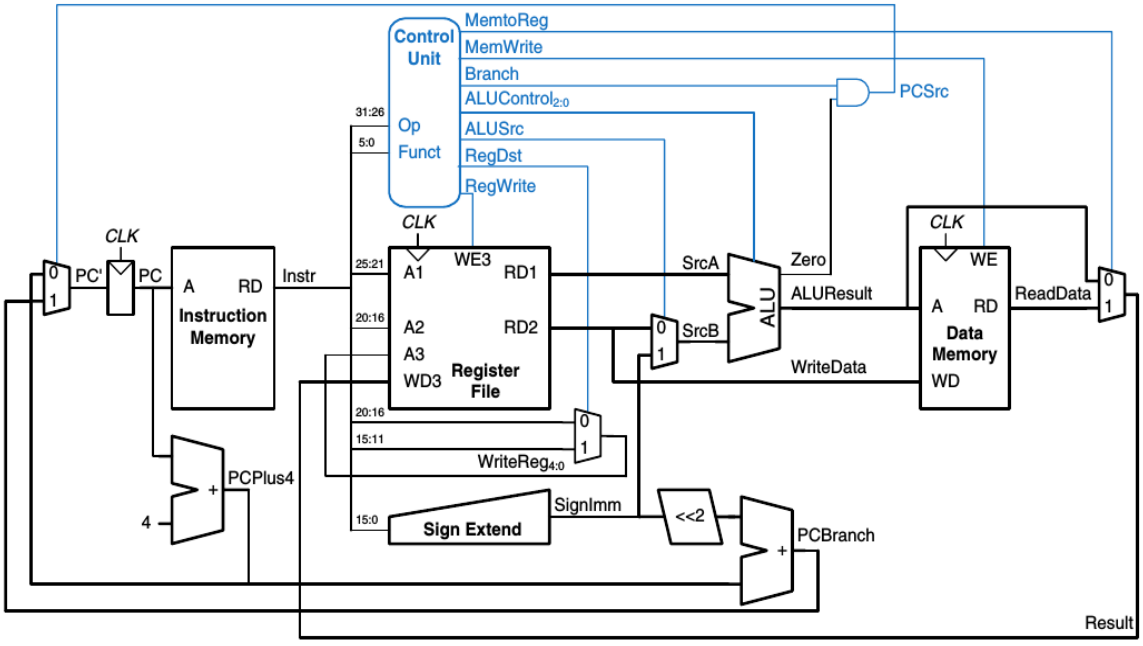

Figure 1: Single cycle MIPS processor

Instruction	Opcode	RegWrite	RegDst	ALUSrc	Branch	MemWrite	MemtoReg	ALUOp
R-type	000000	1	1	0	0	0	0	10
lw	100011	1	0	1	0	0	1	00
sw	101011	0	X	1	0	1	X	00
beq	000100	0	X	0	1	0	X	01

Table 1: Main decoder

2. Repeat Exercise 1 for the following MIPS instructions.

(a) *jal*
(b) *lh*
(c) *jr*
(d) *srl*

3. Many processor architectures have a load with post increment instruction, which updates the index register to point to the next memory word after completing the load.
lwinc $rt, imm($rs) is equivalent to the following two instructions:
lw $rt, imm($rs)
addi $rs, $rs, 4

Repeat Exercise 1 for the *lwinc* instruction. Is it possible to add the instruction without modifying the register file?

4. Your friend is a crack circuit designer. She has offered to redesign one of the units in the single-cycle MIPS processor to have half the delay. Using the delays from Table 2, which unit should she work on to obtain the greatest speedup of the overall processor, and what would the cycle time of the improved machine be?

Element	Parameter	Delay (ps)
register clk-to-Q	t_{pcq}	30
register setup	t_{setup}	20
multiplexer	t_{mux}	25
ALU	t_{ALU}	200
memory read	t_{mem}	250
register file read	t_{RFread}	150
register file setup	$t_{RFsetup}$	20

Table 2: Delay of circuit elements

5. Modify the multicycle MIPS processor to implement one of the following instructions. See Appendix B for a definition of the instructions. Mark up a copy of Figure 2 to indicate the changes to the datapath. Name any new control signals. Mark up a copy of Figure 3 to show the changes to the controller FSM. Describe any other changes that are required.
(a) *srlv*
(b) *ori*
(c) *xori*
(d) *jr*

6. Repeat Exercise 5 for the following MIPS instructions.
(a) *bne*
(b) *lb*
(c) *lbu*
(d) *andi*

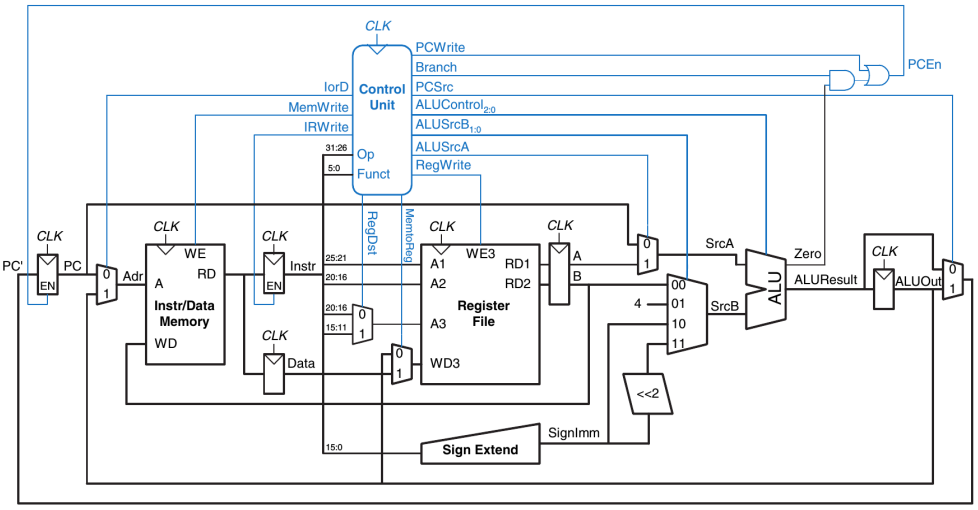

Figure 2: Multi-cycle MIPS

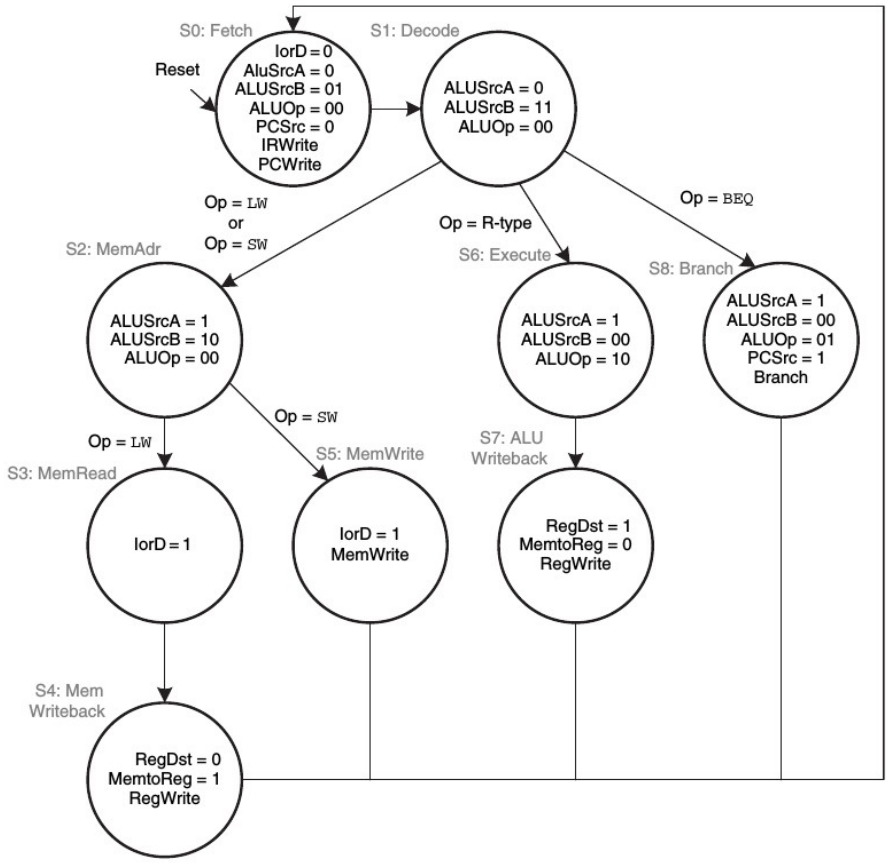

Figure 3: Multi-cycle MIPS controller FSM

7. Your friend, the crack circuit designer, has offered to redesign one of the units in the multicycle MIPS processor to be much faster. Using the delays from Table 2, which unit should she work on to obtain the greatest speedup of the overall processor? How fast should it be? (Making it faster than necessary is a waste of your friend's effort.) What is the cycle time of the improved processor?

8. Goliath Corp claims to have a patent on a three-ported register file. Rather than fighting Goliath in court, Ben Bitdiddle designs a new register file that has only a single read/write port (like the combined instruction and data memory). Redesign the MIPS multicycle datapath and controller to use his new register file.

9. What is the CPI of the redesigned multicycle MIPS processor from 8? Use the SPECINT2000 benchmark which consists of approximately 25% loads, 10% stores,11% branches, 2% jumps, and 52% R-type instructions.

10. How many cycles are required to run the following program on the multicycle MIPS processor? What is the CPI of this program?

```
    addi $s0, $0, done  # result = 5

while:
    beq $s0, $0, done   # if result > 0, execute while block
    addi $s0, $s0, -1   # while block: result=result-1
    j    while

done:
```

11. Repeat Exercise 10 for the following program.

```
        add   $s0, $0,  $0    # i = 0
        add   $s1, $0,  $0    # sum = 0
        addi  $t0, $0,  10    # $t0 = 10
    loop:
        slt   $t1, $s0, $t0   # if (i < 10), $t1 = 1, else $t1 = 0
        beq   $t1, $0,  done  # if $t1 == 0 (i >= 10), branch to done
        add   $s1, $s1, $s0   # sum = sum + i
        addi  $s0, $s0, 1     # increment i
        j loop
    done:
```

12. The pipelined MIPS processor is running the following program. Which registers are being written, and which are being read on the fifth cycle?

```
addi $s1, $s2, 5
sub  $t0, $t1, $t2
lw   $t3, 15($s1)
sw   $t5, 72($t0)
or   $t2, $s4, $s5
```

13. Repeat Exercise 12 for the following MIPS program. Recall that the pipelined MIPS processor has a hazard unit.

```
add  $s0, $t0, $t1
sub  $s1, $t2, $t3
and  $s2, $s0, $s1
or   $s3, $t4, $t5
slt  $s4, $s2, $s3
```

14. Using a diagram similar to Figure 4, show the forwarding and stalls needed to execute the following instructions on the pipelined MIPS processor.

```
add  $t0, $s0, $s1
sub  $t0, $t0, $s2
Iw   $t1, 60($t0)
and  $t2, $t1, $t0
```

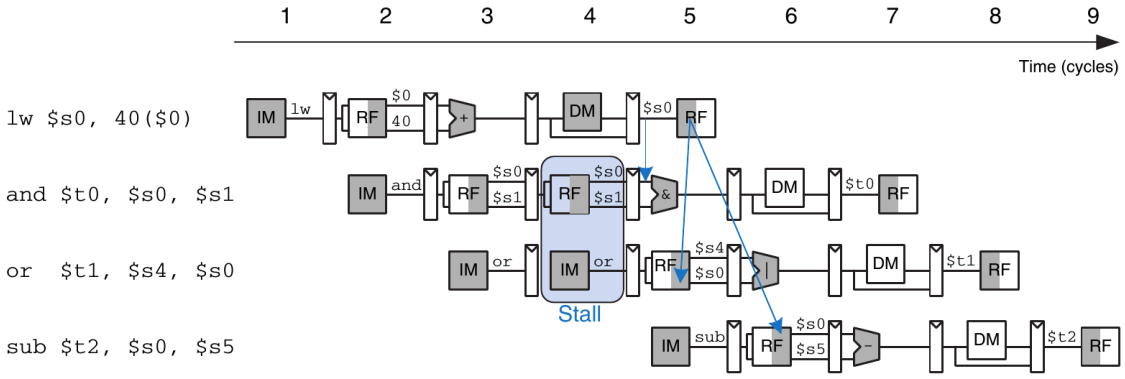

Figure 4: Pipeline diagram to stall hazards

15. Repeat Exercise 14 for the following instructions.

```
add  $t0, $s0, $s1
lw   $t1, 60($s2)
sub  $t2, $t0, $s3
and  $t3, $t1, $t0
```

16. How many cycles are required for the pipelined MIPS processor to issue all of the instructions for the program in Exercise 10? What is the CPI of the processor on this program?

17. Repeat Exercise 16 for the instructions of the program in Exercise 11.

18. Explain how to extend the pipelined MIPS processor to handle the *addi* instruction.

19. Explain how to extend the pipelined processor to handle the *j* instruction. Give particular attention to how the pipeline is flushed when a jump takes place.

20. Your friend, the crack circuit designer, has offered to redesign one of the units in the pipelined MIPS processor to be much faster. Which unit should she work on to obtain the greatest speedup of the overall processor? How fast should it be? (Making it faster than necessary is a waste of your friend's effort.) What is the cycle time of the improved processor?

Use the delays from Table 2 in addition to 40 ps for an equality comparator, 15 ps for an AND gate, 100 ps for a register file write,and 220 ps for a memory write.

21. Suppose the MIPS pipelined processor is divided into 10 stages of 400 ps each, including sequencing overhead. Assume the instruction mix of Exercise 9. Also assume that 50% of the loads are immediately followed by an instruction that uses the result, requiring six stalls, and that 30% of the branches are mispredicted. The target address of a branch or jump instruction is not computed until the end of the second stage. Calculate the average CPI and execution time of computing 100 billion instructions from the SPECINT2000 benchmark for this 10-stage pipelined processor.