



East West University

Semester: Spring-2025

Course Title: Computer Architecture

Course Code: CSE360 **Sec:** 03

Assignment on Chapter 4

Submitted by-

Sheikh Sarafat Hossain

2022-3-60-109

Submitted to-

Md. Ezharul Islam, PhD

Professor

Department of Computer Science & Engineering

Date of Submission: 4th May 2025

Sub: _____

SAT SUN MON TUE WED THU FRI

DATE: / /

4.1.0

Reg-Write	Mem Read	ALU MUX	Mem Write	ALU OP	Reg MUX	Branch
0	0	1	1	10	X	0

ALUMUX is the control signal that controls ALU's MUX at the ALU input, 0(neg) selects the output at the register file, and 1 (Imm) select the immediate from the instruction word as the second input to the ALU. Reg Mux is the control signal that controls the MUX at the Data input to the register file. 0(ALU) selects the output at the ALU and 1(Mem) selects the output at memory.

4.1.30

The branch add produces the data memory as an input. This output is not used where the branch add; second the read pure register produces no output.

Sub: _____

SAT	SUN	MON	TUE	WED	THU	FRI
<input type="checkbox"/>						

DATE: / /

4.1.2:

The branch add unit does not perform a useful function for this instruction. It writes the part of the register.

4.2.1:

The given instruction memory, both registers read ports, the ALU to add, Rd, and Rs, data memory, and Write port in Registers.

4.2.2:

The instruction can be implemented using existing blocks.

4.2.3:

The instruction can be implemented without adding new control signal. It only requires control logic.

Sub: _____

DATE: / /

4.3.1°

$$\begin{aligned} \text{The latency of the path} &= (400 + 200 + 30 + 120 \\ &\quad + 350 + 30) \text{ ps} \\ &= 1130 \text{ ps} \end{aligned}$$

The original clock cycle time without improvement

is 1130 ps

$290 \text{ pp} =$

$$\text{New critical path} = (100 + 100 + 30 + 420 + 200 + 350 + 100) \text{ ps}$$

$= 1600 \text{ ps}$

The new clock cycle time with the improvement is 1600 ps .

4.3.2°

$$\text{Speed Up} = \frac{\text{Old clock cycle time}}{\text{New clock cycle time}} = \frac{1130}{1600} = 0.83$$

Which means it's actually have a slowdown.

Sub: _____

SAT SUN MON TUE WED THU FRI

DATE: / /

4.3.30

Without improvement, the total cost = $1000 + 30 + 10 + 120 + 200 + 2000 + 0.500 \text{ PS} = 3860 \text{ PS}$

With improvement, the total cost = $(1000 + 30 + 10 + 200 + 200 + 200 + 500) \text{ PS} = 440 \text{ PS}$

Instruction executed with improvement = 95%

(5% fewer instruction)

$$\frac{\text{Original cost}}{\text{Performance Ratio}} = \frac{\text{Total cost}}{\text{Ins. executed} \times \text{Clock cycle time}}$$

$$\frac{3860}{0.95} = \frac{3860}{0.95 \times 1600 \times 1.18} = 3.27$$

$$\frac{\text{Improved cost}}{\text{Performance ratio}} = \frac{440}{0.95 \times 1600} = 2.929$$

Sub: _____

DATE: / /

Relative cost: $\frac{4440}{3890} = 1.15$

$$\frac{\text{Cost}}{\text{Performance}} = \frac{1.15}{0.83} = 1.39$$

4.5.1:

The data memory is used by Tw and sw instructions, so $25\% + 10\% = 35\%$ of time is spent with bus.

4.5.2:

The sign extend circuit is actually computing a result in every cycle, but its output is ignored for ADD operation.

Now, $20\% + 25\% + 25\% + 10\% = 80\%$ of time is spent with bus.

4.7.1:

Sign extend	Jump Shift Left - 2
000000000000000010100	000110001000000000010100

Sub: _____

SAT	SUN	MON	TUE	WED	THU	FRI
<input type="checkbox"/>						

DATE: / /

4.2.2:

ALUop[1-9]	Instruction[50-0]
00	010100

4.2.4:

For each max, the formula of its data output during the execution of this instruction and these register values.

Wr reg MUX	ALU Max	mem ALU Max	Branch mux	Jump mux
20P0	20P0	PCTY	PCTY	PCTY

4.2.5:

For the ALU and the two add units, so their data input values, of 2 types

ALU	Add(pcty)	Add(Branch)
3 and 20	pc and 4	pcty, 20X4
		branch

SAT	SUN	MON	TUE	WED	THU	FRI
<input type="checkbox"/>						

Sub: _____

DATE: / /

4.9.1:

P.T.P

Explain

Sequence of instructions: $(S1) \text{LD}, S2, \text{ST}, M2$
 1st cycle OR R_1, R_2, R_3 (S1) $\Rightarrow, S1^M$ will
 2nd cycle - OR R_2, R_1, R_4 (S2) $\Rightarrow, S2^M$ will
 \Rightarrow OR R_1, R_1, R_2 $\Rightarrow, S2^M$ $\Rightarrow, S2^M$ b61.
 Here, Raw on R_1 from i_1 to i_2 and
 R_3 Raw from R_2 from i_2 to i_3 , WAR on
 R_2 from i_1 to i_2 . WAR on R_1 from
 i_1 to i_2 . WAR on R_1 from i_1 to i_3 .
 i_2 to i_3 . WAR on R_1 from i_1 to i_3 .

4.9.4:

The sequence of instruction takes 7 cycles for execution $(2+4) \times 180 \text{ ps} = 11 \times 180 = 1980 \text{ ps}$. The total execution time with forwarding is (2×240) $= 1680$. The speed up because of forwarding will be 1.18 as per the total execution.

Sub: _____

DATE: / /

4.10.1:P.C.P

SW r16, 12(r6) IF-ID-EX-MEM-WB

lw r16, 8(r6) IF-ID-EX-MEM-WB

beq r5, r4, lb1 IF-ID-EX-MEM-WB

add r5, r1, r4 IF-ID-EX-MEM-WB

slt r5, r15, r4 IF-ID-EX-MEM-WB

Total cycle 11. We cannot add NOPs to

the code to eliminate this hazard -

NOPs need to be fetched just like

any other instruction. So, structural

hazard and data hazard must

be addressed.

(PCIF) is bypassed after write

of PCIF to PCIF.

If PCIF is bypassed you have to use PCIF =

PCIF + PCIF * 2.0 + PCIF * 2.1 + PCIF * 2.2

• PCIF bypass

Sub: _____

DATE: / /

4.10.2%

The change only saves one cycle in an entire execution without data hazard.

Instruction executed = 5

cycle with 5 stages, $4+5=9$

cycle with 4 stages, $3+5=8$

$$\text{Speed up} = \frac{\text{cycle with 5 stages}}{\text{cycle with 4 stages}} = \frac{9}{8} = 1.125$$

4.10.3%

The speedup achieved is determined in the ID branch outcomes are determined where the branch stage, relative to the execution, where the branch outcomes are determined in the ID stages

each branch only cause one stall cycle

Instruction executed = 5, branches executed = 4

cycle with branches in EX, $4+5+(1 \times 2) = 11$

cycle with branches ID, $4+5+(1 \times 1) = 10$

$$\text{Speed up} = \frac{11}{10} = 1.10$$

SAT	SUN	MON	TUE	WED	THU	FRI
<input type="checkbox"/>						

Sub: _____

DATE: / /

4.10.4%

The no. of cycle for the 5 stages and 4 stage, Pipeline is already computed. The clock cycle time is equal to the latency of the longest latency stage,

If $I_F = 200 \text{ ps}$, $ID = 120 \text{ ps}$, $EXE = 15 \text{ ps}$,
 $MEM = 100 \text{ ps}$, $WB = 100 \text{ ps}$.

Cycle with 5 stage = 200 ps

Cycle with 4 stage = 210 ps

$$\text{Speedup} = \frac{9 \times 200}{8 \times 210} = 1.0781$$

4.10.5%

Assuming that the latency of ID stage increased by 50% and the latency of EXE stage decreased by 10 ps.

EXE stage decreased by 10 ps.

SAT SUN MON TUE WED THU FRI

Sub: _____

DATE: / /

New ID latency = 180 ps

New EXE latency = 190 ps

New cycle time = 200 ps

old cycle time = 200 ps

$$\text{Speed up} = \frac{11 \times 200}{110 \times 200} = 1.1$$

Q. 10.6

For each branch, the change does not affect execution time because it adds one additional stall.

cycle with branch in execution = $4 + 5 + (10 \times 2) = 11$

Execution time = $(11 \times 200) \text{ ps} = 2200 \text{ ps}$

cycle with branch in MEM = $4 + 5 + (1 \times 3) = 12$

Execution time (branch in MEM) = $12 \times 200 \text{ ps} = 2400 \text{ ps}$

$$\therefore \text{speed up} = \frac{\text{exe. time (in EX)}}{\text{exe. time (in MEM)}} = \frac{2200 \times 10^{12}}{2400 \times 10^{12}}$$

$$= 0.9167$$

$$= 0.92$$

SAT SUN MON TUE WED THU FRI

Sub: _____

DATE: / /

4.11.10

Pipeline execution diagram for the third iteration for the loop step when

lw r1, 0(r1) WB

lw r1, 0(r1) EX MEM WB

beg r1, r0, loop IF --- EX MEM WB

lw r1, 0(r1) IF ID EXE MEM WB

and r1, r1, r2 diff address IF-ID EXE WB

lw r1, 0(r1) Note last IF ID EXE

lw r1, 0(r1) (IF ID demand after step) IF, ID-

IF (EXE IF ID demand after step) IF, ID-

IF (IF ID demand) diff address

IF other

IF (IF ID demand) diff address

IF (IF ID demand) diff address

IF other

IF other

SAT	SUN	MON	TUE	WED	THU	FRI
<input type="checkbox"/>						

Sub: _____

DATE: / /

4.11.2:

We get from 4.11.2, the stalled = stages will be not doing useful work in one iteration it is 18. cycle the total cycle per iteration is 18. cycle in which all stage do useful work in one. so the percentage of cycles in which all stages do useful work is 0%.

4.12.18:

Dependences to the first next instruction results in 2 stall cycle, and the stall is also 2 cycles if the dependence is to both 1st and 2nd next iteration
 $\text{CPI} = 1 + 0.35 \times 2 + 0.15 \times 1 = 1.85$
 stall cycles, $\frac{0.85}{1.85} = 0.459$
 $\therefore 0.459 \times 100 = 45.9 = 46\%$

SAT	SUN	MON	TUE	WED	THU	FRI
<input type="checkbox"/>						

Sub: _____

DATE: / /

4.12.2^o

considering the full forwarding, the MEM stage of one instruction to the 1st next instruction cause RAN dependencies

$$CPI = 1 + 20\% = 1 + 0.20 = 1.20$$

$$\text{Stall cycle} = \frac{0.20}{1.20} = -1.67 \times 1.00 = 1.67 \text{ or } 17\%$$

4.12.3^o

Stall cycles of EXE to FMEM, 0.2 + 0.05 + 0.1 + 0.1

$$0.2 + 0.05 + 0.1 + 0.1 = 0.45$$

Exe to 2nd has no stall. $\frac{\text{MEM}}{\text{WB}}$

$$0.2 + \frac{0.05}{2.0} = 0.25 \text{ (25% stall)}$$

is better than $\frac{\text{EXE}}{\text{MEM}}$.

SAT SUN MON TUE WED THU FRI

Sub: _____

DATE: / /

4.12.40

clock cycle time without forwarding,

$$1.85 \times 150 = 277.5 \text{ ps}$$

clock cycle time with forwarding,

$$\begin{aligned} &= 1.2 \times 150 \\ &= 180 \text{ ps} \end{aligned}$$

$$\text{Speed up} = \frac{277.5}{180} = 1.54167.$$