CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Pipeline Processing

Lecture No.- 03

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Instruction Pipeline

Topic Pipeline Hazard

Topic Operand Forwarding

Topics to be Covered

- **Topic** Pipeline Hazards
- **Topic** Data Hazard Classification
- **Topic** CPI in Pipeline

Ques)

IF — 1

ID / OF — 1

EX — 1

WB — 1

R1 ← R2 * R3

R4 ← R1 − R6

R5 ← R4 + R8

no. of cycles needed if no operand forwarding

no. of stalls due to immediate data dependency

= phase no. of WB − phase no. of OF

= 4 − 2

= 2

n = 3

k = 4

no. of cycles w/o hazard = 4 + 3 − 1 = 6

stalls due to data dependency = 2 * 2 = 4

Total = 10

[NAT]

Ans = 8

#Q. Consider a pipelined processor with the following four stages:

IF: Instruction Fetch ID: Instruction Decode and Operand Fetch

EX: Execute WB: Write Back

The IF, ID, and WB stages take one clock cycle each to complete the operation. The number of clock cycles for the EX stage depends on the instruction. The ADD and SUB instructions need 1 clock cycle and the MUL instruction needs 3 clock cycle in the EX stage. Operand forwarding is used in the pipelined processor. What is the number of clock cycles taken to complete the following sequence of instructions?

ADD R2, R1, R0	R2←R1+R0
MUL R4, R3, R2	R4←R3*R2
SUB R6, R5, R4	R6←R5-R4

ALU to ALU data dependency but operand forwarding will eliminate all problems.

Solution

IF — 1
ID — 1
EX
WB — 1

ADD	SUB	MUL
1	1	3

1	2	3	4	5	6	7	(8)	9	10
IF	ID	EX	WB						
	IF	ID	EX	EX	EX	WB			
		IF	ID	—	—	EX	WB		

ADD
MUL
SUB

Method 2:-

IF — 1
ID — 1
EX
WB — 1

	ADD	SUB	MUL
	1	1	3
extra			2
			-

$n = 3$
$k = 4$

no. of cycles needed w/o hazard $= k + n - 1 = 4 + 3 - 1 = 6$
stalls due to 1 MUL instn $= 2$
Total $= 8$

Ques)

seq. of instⁿ ⇒ no. of cycles = ?

ADD
SUB
DIV
MUL

$k = 4$
$n = 4$

$= 14$ Ans.

no. of cycles w/o hazard $= 4+4-1 = 7$
stalls due to structural hazard
$= 1+2+4 = 7$

Total $= 14$

[NAT]

Ans = 12

#Q. Consider a pipelined processor with the following four stages:

IF: Instruction Fetch ID: Instruction Decode and Operand Fetch

EX: Execute WB: Write Back

The IF, ID, and WB stages take one clock cycle each to complete the operation. The number of clock cycles for the EX stage depends on the instruction. The ADD and SUB instructions need 1 clock cycle and the MUL instruction needs 3 clock cycle in the EX stage. Operand forwarding is not used in the pipelined processor. What is the number of clock cycles taken to complete the following sequence of instructions?

ADD R2, R1, R0	R2←R1+R0
MUL R4, R3, R2	R4←R3*R2
SUB R6, R5, R4	R6←R5-R4

Solution

stalls due to each immediate data dependency = 4 − 2 = 2

no. of cycles without hazards = 4 + 3 − 1 = 6

stalls due to structural hazard = 2

〃 data hazards = 2 * 2 = 4

Total = 12

[NAT]

stalls due to each immediate data dependency = 5 − 3 = 2

#Q. IF – 1 cycle
ID – 1 cycle
OF – 1 cycle
EX ----------
WB – 1 cycle

	ADD	SUB	MUL
EX	1	1	3
stalls	0	0	2

ADD type 7 instructions
SUB type 10 instructions
MUL type 5 instructions

$n = 7 + 10 + 5 = 22$
$k = 5$

Total Cycles with and without operand forwarding? → 46

↳ 5 immediate data dependencies

36

with operand forwarding :-

cycles w/o hazard = 5 + 22 − 1 = 26
stalls = 5 * 2 = 10
Total = 36

without operand forwarding :-

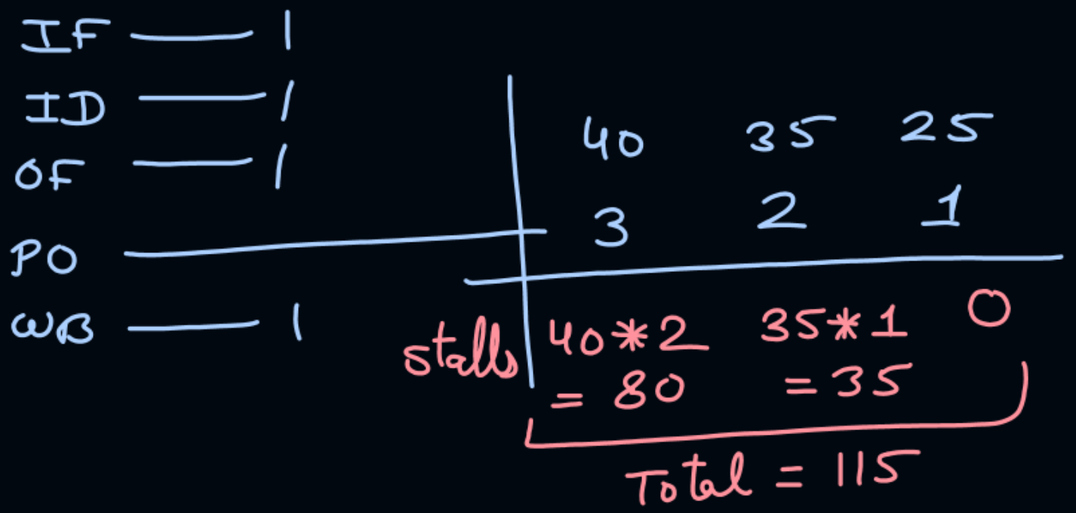
cycles w/o hazard = 5 + 22 − 1 = 26
stalls due to structural hazard = 10
stalls —11— data hazards = 5 * 2 = 10
Total = 46

[NAT] GATE-2018

Ans = 219

#Q. The instruction pipeline of a RISC processor has the following stages: Instruction Fetch (IF), Instruction Decode (ID), Operand Fetch (OF), Perform Operation (PO) and Writeback (WB), The IF, ID, OF and WB stages take 1 clock cycle each for every instruction. Consider a sequence of 100 instructions. In the PO stage, 40 instructions take 3 clock cycles each, 35 instructions take 2 clock cycles each, and the remaining 25 instructions take 1 clock cycle each. Assume that there are no data hazards and no control hazards.

The number of clock cycles required for completion of execution of the sequence of instruction is 219 ?



w/o hazard no. of cycles = 5 + 100 − 1 = 104

stalls	= 115
Total	219

[NAT]

Ans = 23

#Q. Consider a 4-stage pipeline processor. The number of cycles needed by the four instructions I1, I2, I3, I4 in stages S1, S2, S3, S4 is shown below:

	S_1	S_2	S_3	S_4
I1	2	1	1	1
I2	1	3	2	2
I3	2	1	1	3
I4	1	2	2	2

What is the number of cycles needed to execute the following loop?

For (i = 1 to 2) (I1; I2; I3; I4)

Solution

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I1	S1	S1	S2	S3	S4										
I2		–	S1	S2	S2	S2	S3	S3	S4	S4					
I3				S1	S1	–	S2	–	S3	–	S4	S4	S4		
I4															
I1															
I2															
I3															
I4															

	S_1	S_2	S_3	S_4
I1	2	1	1	1
I2	1	3	2	2
I3	2	1	1	3
I4	1	2	2	2

Solution

	S1	S2	S3	S4
I1	2	3	4	5
I2	3	6	8	10
I3	5	7	9	13
I4	6	9	11	15
I1	8	10	12	16
I2	9	13	15	18
I3	11	14	16	21
I4	12	16	18	23

	S_1	S_2	S_3	S_4
I1	2	1	1	1
I2	1	3	2	2
I3	2	1	1	3
I4	1	2	2	2

Topic : Hazard Classification

Assume that there are two instructions i and j, and i is executed before j

1. RAW (Read After write)
2. WAW (write — " —)
3. WAR (write after Read)

Topic : Read After Write (RAW)

→ True data dependency

j tries to read a source before *i* writes it. So *j* incorrectly gets the old value.

i : $R1 \leftarrow R2 + R3$

j : $R6 \leftarrow R1 * R5$

Soln ⇒ operand forwarding

⇓

only ALU to ALU RAW hazards

Topic : Write After Write (WAW)

→ write dependency
→ False dependency

j tries to write an operand before it is written by i

i : $R1 \leftarrow R2 + R3$
j : $R1 \leftarrow R5 * R6$

} Soln ⇒ Register Renaming (How Soln)

⇓

i : $R1 \leftarrow R2 + R3$
j : $R7 \leftarrow R5 * R6$

Topic : Write After Read (WAR)

→ Anti-dependency
→ False dependency

j tries to write an operand before *i* reads it, hence *i* incorrectly gets new value

i : R1 ← R2 + R3
j : R2 ← R5 * R6

} Soln ⇒ Reg. Renaming

i : R1 ← R2 + R3
j : R8 ← R5 * R6

[NAT]

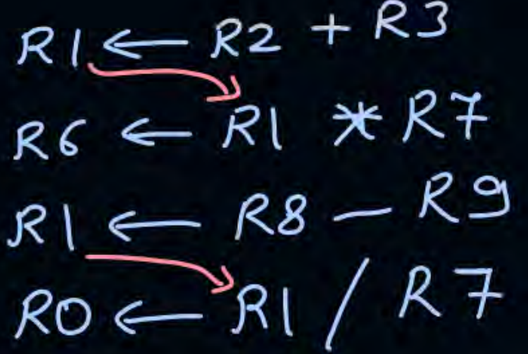
4 1 3

#Q. Count the number of RAW, WAW and WAR dependencies?

ADD R2, R1, R0	R2←R1+R0
MUL R4, R3, R2	R4←R3*R2
SUB R6, R5, R4	R6←R5-R4
ADD R6, R7, R8	R6←R7+R8
MUL R7, R1, R2	R7←R1*R2
SUB R1, R3, R4	R1←R3-R4

No. of RAW dependencies:-

Ans = 2



[MSQ]

B, C

#Q. An instruction format has the following structure

Instruction number: opcode destination reg, source reg-1, source reg-2

Consider the following sequence of instruction to be executed in a pipelined processor

I1: DIV R3, R1, R2	R3 ← R1 / R2
I2: SUB R5, R3, R4	R5 ← R3 − R4
I3: ADD R3, R5, R6	R3 ← R5 + R6
I4: MUL R7, R3, R8	R7 ← R3 * R8

Which of the following statement is/are TRUE?

A. There is a WAW dependency on R3 between I3 and I4

B. There is a WAW dependency on R3 between I1 and I3

C. There is a RAW dependency on R3 between I1 and I2

D. There is a RAW dependency on R3 between I2 and I3

[MSQ]

#Q. For a pipelined CPU with a single ALU, consider the following situations

I. The J + 1st instruction uses the result of the Jth instruction as an operand → data hazard

II. The execution of a conditional jump instruction → branch hazard

III. The Jth and J + 1st instructions require the ALU at the same time → structural hazard

Which of the above can cause a hazard

A. I and II only

B. II and III only

C. III only

D. ✓ All the three

[NAT]

#Q. Register renaming is done in pipelined processors:

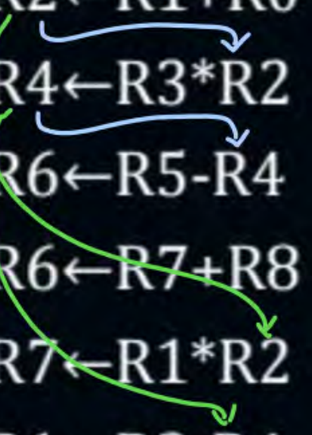
a) as an alternative to register allocation at compile time
b) for efficient access to function parameters and local variables
c) to handle certain kinds of hazards
d) as part of address translation

[NAT]

#Q. Consider the following sequence of instructions, which is to be executed on a 5 stage pipeline: IF, ID, OF, EX and WB. Each stage takes one cycle for each instruction. Total Cycles required to execute this sequence

1. When pipeline uses operand forwarding = 10 cycles
2. When pipeline does not use operand forwarding ⇒ 14 cycles

ADD R2, R1, R0	R2←R1+R0
MUL R4, R3, R2	R4←R3*R2
SUB R6, R5, R4	R6←R5-R4
ADD R6, R7, R8	R6←R7+R8
MUL R7, R1, R2	R7←R1*R2
SUB R1, R3, R4	R1←R3-R4



$k = 5$
$n = 6$

no. of cycles w/o hazard = $5 + 6 - 1 = 10$ cycles

---

stalls due to each immediate data dependency = WB phase no − OF phase no

$= 5 - 3$

$= 2$

---

for 2 data dependency, stalls = $2 * 2 = 4$

Total = $10 + 4 = 14$

[NAT]

#Q. Consider the following sequence of instructions, which is to be executed on a 5 stage pipeline: IF, ID, OF, EX and WB. Each stage takes one cycle for each instruction. Total Cycles required to execute this sequence

1. When pipeline uses operand forwarding ⇒ 12
2. When pipeline does not use operand forwarding ⇒ 14

ADD R2, R1, R0	R2←R1+R0
MUL R4, R3, R2	R4←R3*R2
LOAD R7, (1000)	R7←M[1000]
ADD R6, R7, R8	R6←R7+R8
MUL R7, R1, R2	R7←R1*R2
SUB R1, R3, R4	R1←R3-R4

$k = 5$

$n = 6$

w/o hazards cycles $= 5 + 6 - 1 = 10$

stalls due to mem to ALU data dependency $= 2$

12

---

w/o operand forwarding,

stalls $= 2 * 2 = 4$

Total $= 10 + 4 = 14$

2 mins Summary

Topic Pipeline Hazards

Topic Data Hazard Classification

Topic CPI in Pipeline

Happy Learning

THANK - YOU