CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Pipeline Processing

Lecture No.- 02

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Synchronous Pipeline

Topic Latency & Throughput

Topic Instruction Pipeline

Topics to be Covered

Topic Instruction Pipeline

Topic Pipeline Hazard

Topic Operand Forwarding

Topic : Instruction Pipeline

Assume a 5-stage pipeline

IF:	Instruction Fetch
ID:	Instruction Decode & Address Calculation
OF:	Operand Fetch
EX:	Execution
WB:	Write Back

Topic : Instruction Pipeline

Clock Cycles

Instructions	1	2	3	4	5	6	7	8	9	10	11	12	13	14
I1	IF	ID	OF	EX	WB									
I2		IF	ID	OF	EX	WB								
I3			IF	ID	OF	EX	WB							
I4				IF	ID	OF	EX	WB						
I5					~~IF~~	–	–							
I6														
I7														
I8								IF	ID	OF	EX	WB		
I9									IF	ID	OF	EX	WB	

branch

Target

stall cycles
(extra cycles)

Topic : Instruction Pipeline

when CPU decodes branch inst^n then next inst^n is fetched after the condit^n is evaluated (the branch outcome is known)

no. of inst^{ns} executed $(n) = 6$ (I1 to I4, I8 to I9)

no. of segments $(k) = 5$

no. of cycles usually required $= k+n-1 = 5+6-1 = 10$

but because of branch inst^n, extra cycles needed $= 3$

Total $= 13$

In our example if I4 (branch instⁿ) does not take jump or branch.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
I4				IF	ID	OF	EX	WB									
I5					-	-	-	IF	ID	OF	EX	WB					
I6									IF	-	-	-	WB				
I7										IF	-	-	-	WB			
I8											IF	-	-	-	WB		
I9												IF	-	-	-	WB	

stalls

n = 9

stalls = 3

$$\text{no. of cycles} = (5 + 9 - 1) + 3$$

$$= \underline{\underline{16}}$$

If after i^{th} stage of branch instn. target is available

no. of stall cycles needed due to each branch instn = $i - 1$

[NAT]

Ans = 15

#Q. Consider a program which contains 50 instructions I1, I2 , I3 ..... I50. Further consider a 5-stage pipeline with stages as: Instruction Fetch, Decode, Operand Fetch, Execution and Write-Back. The program contains only 1 branch instruction which is instruction I5 and its target is instruction I48. If during the execution of the program the branch is taken then number of cycles required to execute this program in the given pipeline is ______ ?

no. of inst^{ns} executed$(n) = 5 + 3 = 8$ (I1 to I5, I48 to I50)

$k = 5$

stalls due to branch inst^n $= 4 - 1 = 3$

w/o branch no. of cycles $= k + n - 1 = 5 + 8 - 1 = 12$

stalls $= 3$

Total $= \underline{\underline{15}}$

[NAT]

#Q. In the above question if the pipeline has cycle time 10ns then total time required to execute the program is 150 ns?

$15 * 10 = 150\text{ns}$

[NAT]

Ans = 198

#Q. Consider a program which contains 200 instructions I1, I2 , I3 ..... I200. Further consider a 5-stage pipeline with stages as: Instruction Fetch, Decode, Operand Fetch, Execution and Write-Back. The program contains only 1 branch instruction which is instruction I6 and its target is instruction I16. If during the execution of the program the branch is taken then number of cycles required to execute this program in the given pipeline is ______ ?

$n = \left[(6-1+1)+(200-16+1)\right] = 191$

stalls = 4 − 1 = 3

k = 5

without stall = 5 + 191 − 1 = 195

stalls = 3

Total = 198

[NAT]

Ans = 207

#Q. Consider a program which contains 200 instructions I1, I2 , I3 ..... I200. Further consider a 5-stage pipeline with stages as: Instruction Fetch, Decode, Operand Fetch, Execution and Write-Back. The program contains only 1 branch instruction which is instruction I8 and its target is instruction I108. If during the execution of the program the branch is not taken then number of cycles required to execute this program in the given pipeline is __?

If branch not taken, $n = 200$

$k = 5$

stalls $= 4 - 1 = 3$

without stalls, no. of cycles $= 5 + 200 - 1 = 204$

stalls $= 3$

Total $= 207$

Topic : Pipeline Hazards

Situations that prevent the next instruction from being executing during its designated clock cycle

Topic : Pipeline Hazards

1. Structural Hazard / Resource Conflict
2. Data Hazard / Data Dependency
3. Control Hazard / Branch Difficulty

Topic : Structural Hazard

2 different segments try to use same resource at same time.

2 inst^{ns} ⇒ I1 ⇒ Multiplicat^n ⇒ takes 2 cycles in EX phase

I2 ⇒ Addition

I1	IF	ID	OF	EX	EX	WB	
I2		IF	ID	OF	—	EX	WB

due to structural hazard

There is no any solution of such structural hazard

Topic : Data Hazard or Data Dependency

Result of an instruction is used as input in next

I1 : $R1 \leftarrow R2 + R3$

I2 : $R5 \leftarrow R1 * R6$

IF	ID	OF	EX	WB			
	IF	ID	–	–	OF	EX	WB

2 stalls

R1 = ~~2~~ 8

R2 = 3

R3 = 5

R6 = 2

R5 = 16

Note :-

no. of stalls due to 2 next to next data dependency inst^{ns}

= phase no. of WB – phase no. of OF

I1 :- R1 ← R2 + R3

I2 :- R9 ← R8 ← independent instn

I3 :- R5 ← R1 * R6

```
IF  ID  OF  EX  WB
    IF  ID  OF  EX  WB
        IF  ID  -   OF  EX  WB
                |
                └─> stall
```

I1 :- R1 ← R2 + R3

I2 :- R9 ← R8

I3 :- R7 ← R15 + R12

} independent

I4 :- R5 ← R1 * R6

```
IF  ID  OF  EX  WB
    IF  ID  OF  EX  WB
        IF  ID  OF  EX  WB
            IF  ID  OF  EX  WB
```

sol^n of data dependency

S/w sol^n

(by compiler)

⇓

delayed Load

⇓

Insert independent or no operation inst^{ns} b/w dependent inst^{ns}.

H/w sol^n

- → Hardware interlock
- → operand forwarding (bypassing)

providing stalls to delay OF of dependent inst^n

operand forwarding :-

I1 :- R1 ← R2 + R3

I2 :- R5 ← R1 * R6

1	2	3	4	5	
IF	ID	OF	EX	WB	
	IF	ID	OF	EX	WB

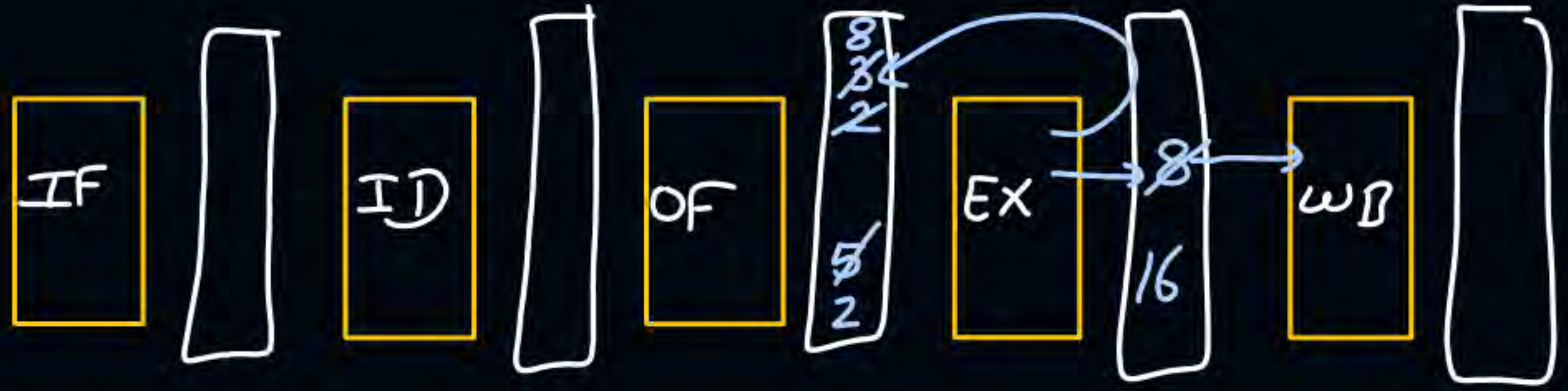
R1 = ~~2~~ 8

R2 = 3

R3 = 5

R6 = 2

R5 = ~~1~~ 16



ALU to ALU data dependency ⇒ operand forwarding eliminates all stalls

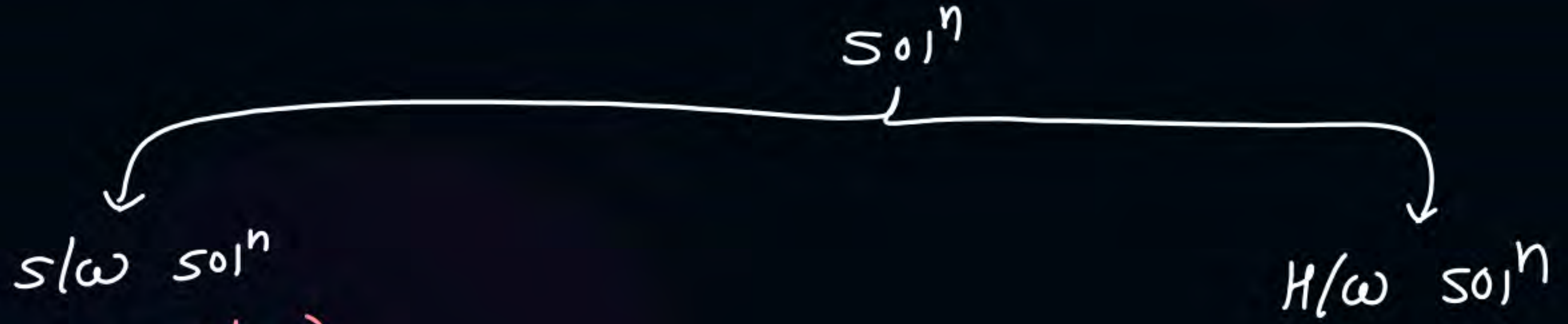
Load to ALU dependency ⇒ R1 ← Memory
R5 ← R1 * R6

stalls needed

ALU to store dependency ⇒ R1 ← R2 + R3
memory ← R1

Topic : Control Hazard or Branch Difficulty

Hazards because of branch instructions



sol^n

$S/W\ sol^n$

(by compiler)

delayed branch

Insert no operat^n or independent inst^{ns} after branch inst^n.

$H/W\ sol^n$

⇓

branch prediction

Branch prediction :-

Predicts if branch is taken or not before even branch outcome is known. and continues executn of instns as per the predictn.

If predictn is known to be wrong when branch outcome available then revert executn till branch instn and continue instn execution in correct directn.

2 mins Summary

Topic Instruction Pipeline

Topic Pipeline Hazard

Topic Operand Forwarding

Happy Learning

THANK - YOU