IT5002 Tutorial 4

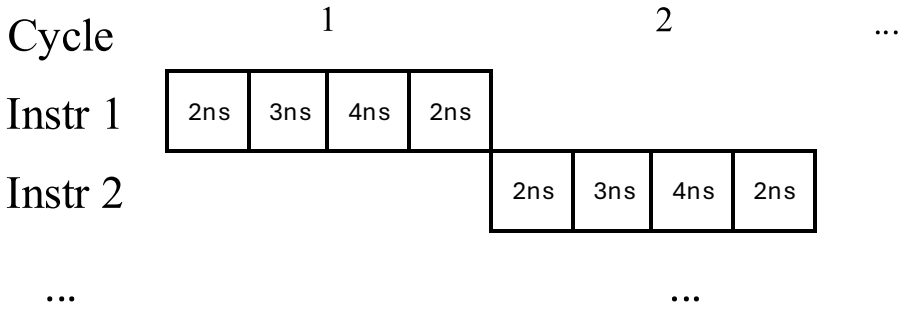
AY 2025/26 Semester 1

Prepared by Michael Yang

Slides adapted from Theodore Leebrant, Prof. Colin and Prof. Aaron

1. Suppose the four stages in some 4-stage pipeline take the following timing: 2ns, 3ns, 4ns, and 2ns. Given 1000 instructions, what is the speedup (in two decimal places) of the pipelined processor compared to the non-pipelined single-cycle processor?

Non-pipelined

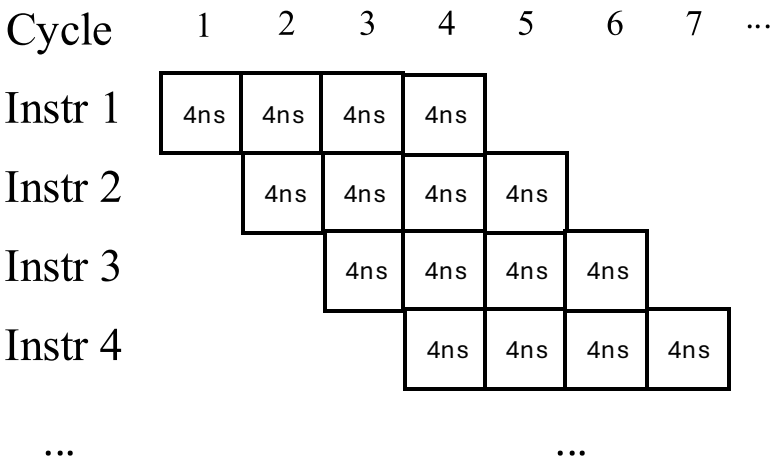


1 cycle takes 2+3+4+2 = 11ns

For 1000 instructions, it takes 11 x 1000 = 11000ns

Speedup = 11000 / 4012 = 2.74 times

Pipelined



Time to complete 1 cycle = Time taken by longest stage = 4ns

3 cycles to fill up pipeline => 3 x 4 = 12ns

1000 instructions => 1000 x 4 = 4000ns. Total = 12 + 4000 = 4012ns

Another way to see it: the 1000^{th} instruction starts at cycle 1000, ends at 1003

2. Suppose the pipeline registers (also known as pipeline latches) store the following information:

IF/ID (Instr Fetch/Instr Decode)	
PC+4	
OpCode	
Rs	
Rt	
Rd	
Funct	
Imm(16)	

ID/EX (Instr Decode/Execute)	
MemToReg	
RegWrite	
MemRead	
MemWrite	
Branch	
RegDest	
ALUSrc	
ALUop	
PC+4	
RD1/Op1	
RD2/Op2	
Rt	
Rd	
Imm(32)	

EX/MEM (Execute/Memory)	
MemToReg	
RegWrite	
MemRead	
MemWrite	
Branch	
BranchTarget	
isZero?	
ALURes	
RD2/Op2	
RegDest (DstRNum)	

MEM/WB (Memory/Write Back)	
MemToReg	
RegWrite	
MemRes	
ALURes	
RegDest (DstRNum)	

Registers 1 to 31 are initialized to 101 + their register number, e.g. [$1] = 102. You can mark unused pipeline registers with X.

Fill the contents of the pipeline registers for the following instructions:

```
0x8df80000  # lw $24, 0($15)    #Inst.Addr = 0x100

0x1023000C  # beq $1, $3, 12    #Inst.Addr = 0x100

0x0285c822  # sub $25, $20, $5 #Inst.Addr = 0x100
```

0x8df80000 = lw $24, 0($15) Instr Addr = 0x100

IF/ID stores:

Data

PC+4 = 0x104

Opcode = 0x23

Rs = $15

Rt = $24

Rd = X

Funct = X

Imm(16) = 0

ID/EX stores:

Control Signals

MemToReg = 1

RegWrite = 1

MemRead = 1

MemWrite = 0

Branch = 0

RegDest = 0

ALUSrc = 1

ALUop = 00

Data

PC+4 = 0x104

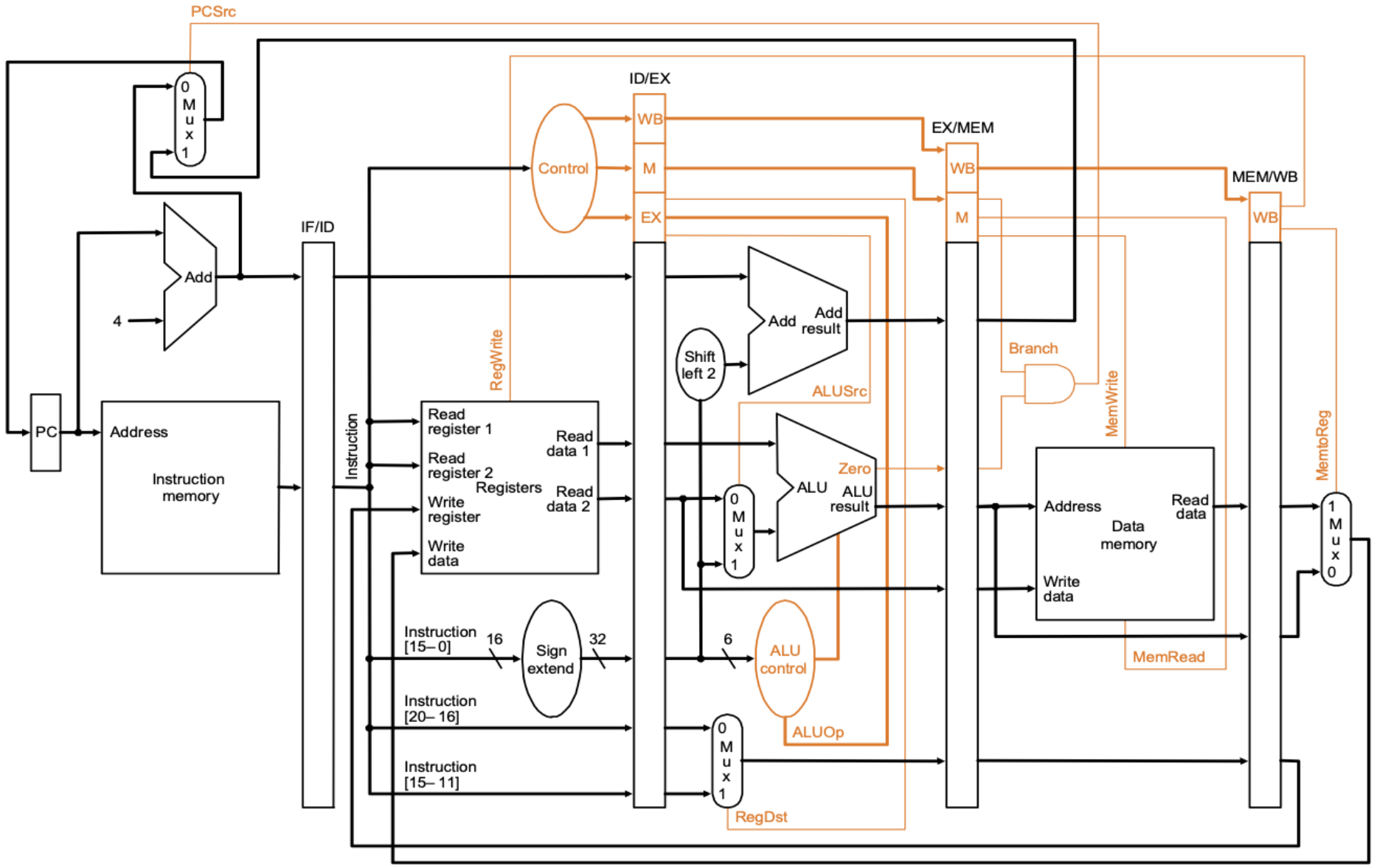
RD1/Op1 = 116

RD2/Op2 = X

Rt = $24

Rd = X

Imm(32) = 0



EX/MEM stores:

Control Signals

MemToReg = 1

RegWrite = 1

MemRead = 1

MemWrite = 0

Branch = 0

BranchTarget = X

isZero? = X

Data

ALURes = 116

RD2 = X

DstRNum = $24

MEM/WB stores:

Control Signals

MemToReg = 1

RegWrite = 1

Data

MemRes = Mem(116)

ALURes = X

DstRNum = $24

`0x8df80000 = lw $24, 0($15)` `Instr Addr = 0x100`

IF/ID (Instr Fetch/Instr Decode)	
PC+4	0x104
OpCode	0x23
Rs	$15
Rt	$24
Rd	X
Funct	X
Imm(16)	0

ID/EX (Instr Decode/Execute)	
MemToReg	1
RegWrite	1
MemRead	1
MemWrite	0
Branch	0
RegDest	0
ALUSrc	1
ALUop	00
PC+4	0x104
RD1/Op1	116
RD2/Op2	X
Rt	$24
Rd	X
Imm(32)	0

EX/MEM (Execute/Memory)	
MemToReg	1
RegWrite	1
MemRead	1
MemWrite	0
Branch	0
BranchTarget	X
isZero?	X
ALURes	116
RD2/Op2	X
DstRNum	$24

MEM/WB (Memory/Write Back)	
MemToReg	1
RegWrite	1
MemRes	Mem(116)
ALURes	X
DstRNum	$24

0x1023000C = beq $1, $3, 12 Instr Addr = 0x100

IF/ID stores:

Data

PC+4 = 0x104
Opcode = 0x4
Rs = $1
Rt = $3
Rd = X
Funct = X
Imm(16) = 12

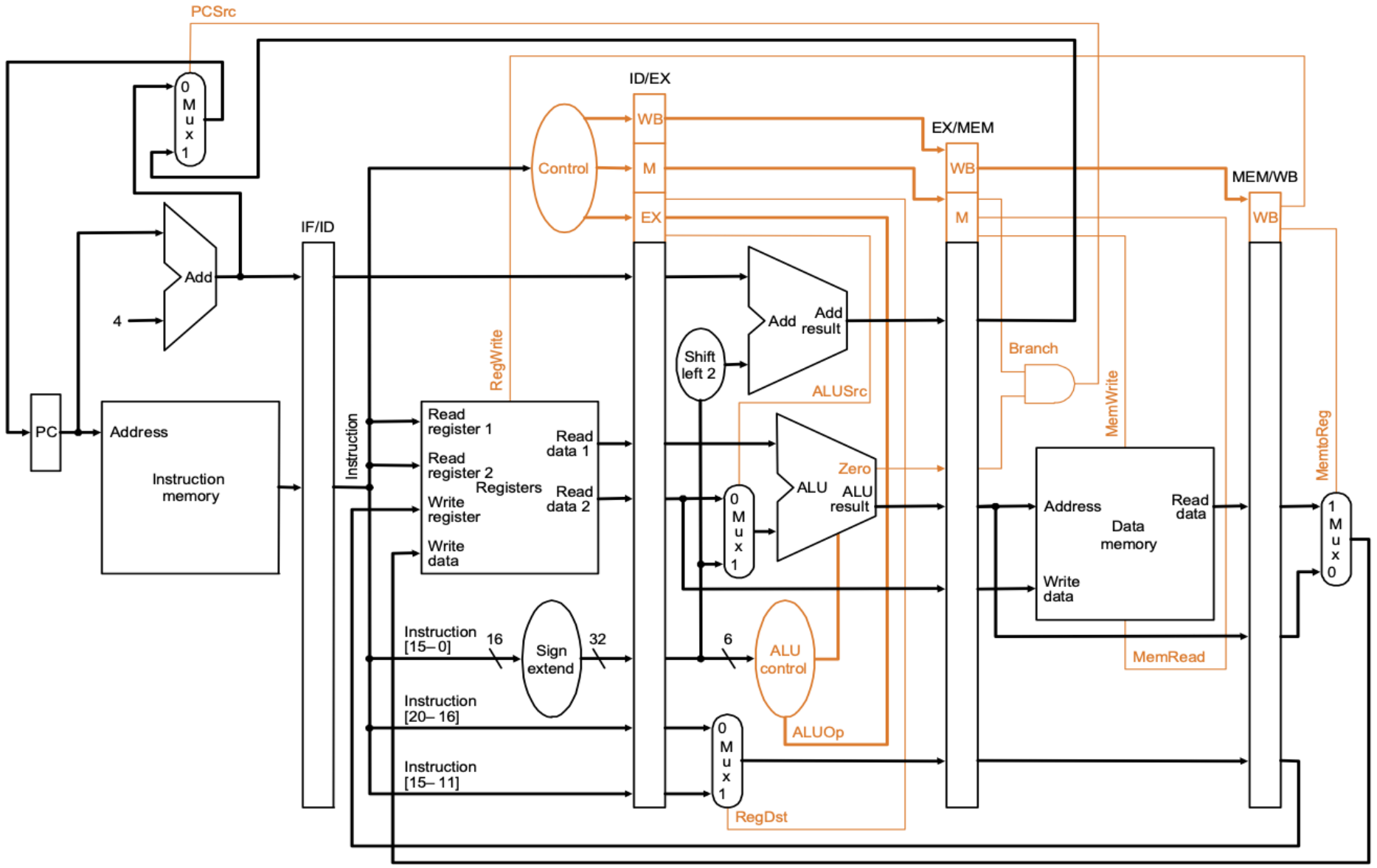
ID/EX stores:

Control Signals

MemToReg = X
RegWrite = 0
MemRead = 0
MemWrite = 0
Branch = 1
RegDest = X
ALUSrc = 0
ALUop= 01

Data

PC+4 = 0x104
RD1/Op1 = 102
RD2/Op2 = 104
Rt = X
Rd = X
Imm(32) = 12



EX/MEM stores:

Control Signals

MemToReg = X
RegWrite = 0
MemRead = 0
MemWrite = 0
Branch = 1
BranchTarget = 0x134
isZero? = 0

Data

ALURes = -2
RD2 = X
DstRNum = X

MEM/WB stores:

Control Signals

MemToReg = X
RegWrite = 0

Data

MemRes = X
ALURes = X
DstRNum = X

0x1023000C = beq $1, $3, 12 Instr Addr = 0x100

IF/ID (Instr Fetch/Instr Decode)	
PC+4	0x104
OpCode	0x4
Rs	$1
Rt	$3
Rd	X
Funct	X
Imm(16)	12

ID/EX (Instr Decode/Execute)	
MemToReg	X
RegWrite	0
MemRead	0
MemWrite	0
Branch	1
RegDest	X
ALUSrc	0
ALUop	01
PC+4	0x104
RD1/Op1	102
RD2/Op2	104
Rt	X
Rd	X
Imm(32)	12

EX/MEM (Execute/Memory)	
MemToReg	X
RegWrite	0
MemRead	0
MemWrite	0
Branch	1
BranchTarget	0x134
isZero?	0
ALURes	-2
RD2/Op2	X
RegDest (DstRNum)	X

MEM/WB (Memory/Write Back)	
MemToReg	X
RegWrite	0
MemRes	X
ALURes	X
RegDest (DstRNum)	X

0x0285c822 = sub $25, $20, $5 Instr Addr = 0x100

IF/ID stores:

Data

PC+4 = 0x104

Opcode = 0x0

Rs = $20

Rt = $5

Rd = $25

Funct = 22

Imm(16) = X

ID/EX stores:

Control Signals

MemToReg = 0

RegWrite = 1

MemRead = 0

MemWrite = 0

Branch = 0

RegDest = 1

ALUSrc = 0

ALUop = 10

Data

PC+4 = 0x104

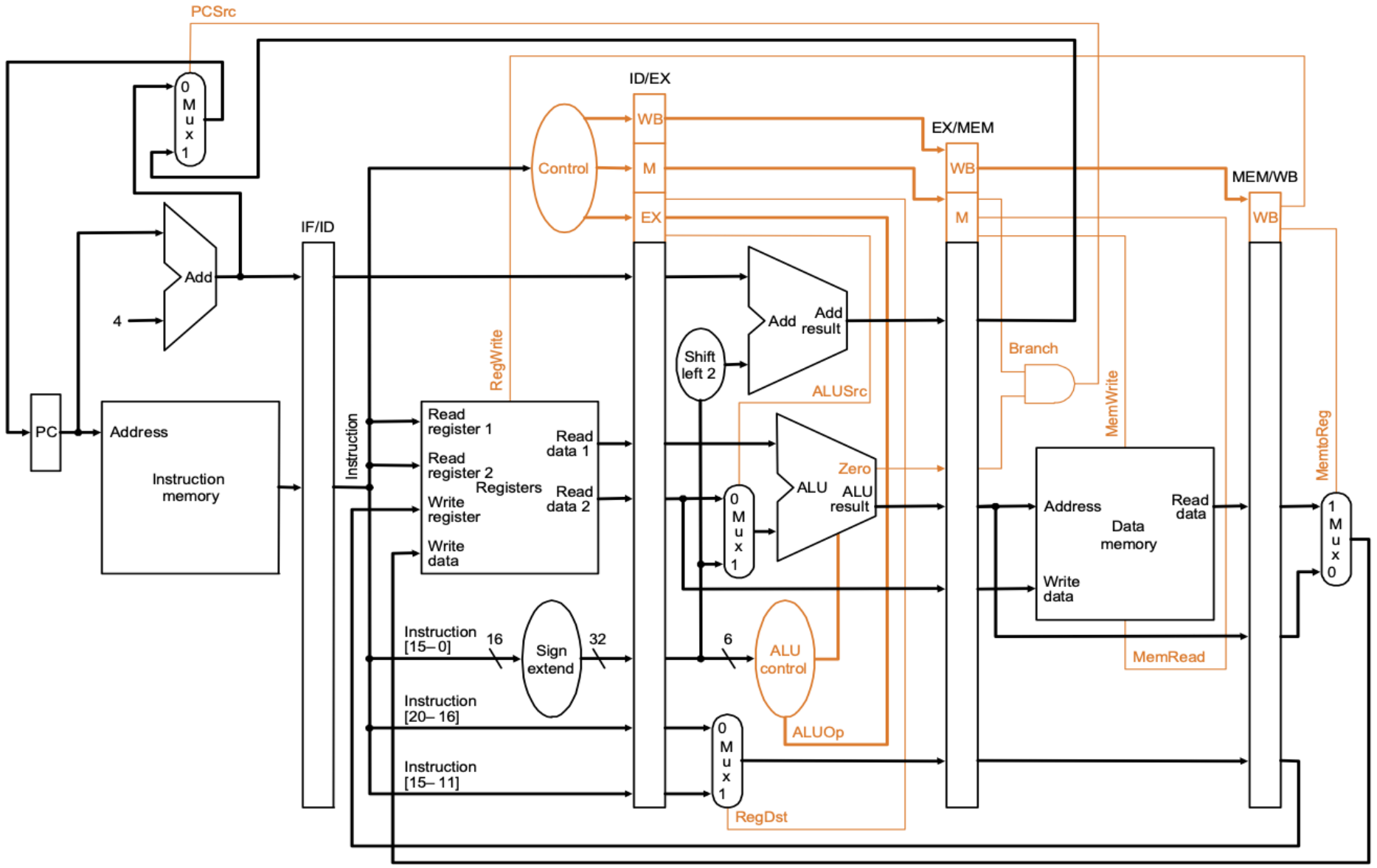
RD1/Op1 = 121

RD2/Op2 = 106

Rt = X

Rd = $25

Imm(32) = X



EX/MEM stores:

Control Signals

MemToReg = 0

RegWrite = 1

MemRead = 0

MemWrite = 0

Branch = 0

BranchTarget = X

isZero? = X

Data

ALURes = 15

RD2 = X

DstRNum = $25

MEM/WB stores:

Control Signals

MemToReg = 0

RegWrite = 1

Data

MemRes = X

ALURes = 15

DstRNum = $25

0x0285c822 = sub $25, $20, $5 Instr Addr = 0x100

IF/ID (Instr Fetch/Instr Decode)	
PC+4	0x104
OpCode	0x0
Rs	$20
Rt	$5
Rd	$25
Funct	22
Imm(16)	X

ID/EX (Instr Decode/Execute)	
MemToReg	0
RegWrite	1
MemRead	0
MemWrite	0
Branch	0
RegDest	1
ALUSrc	0
ALUop	10
PC+4	0x104
RD1/Op1	121
RD2/Op2	106
Rt	X
Rd	$25
Imm(32)	X

EX/MEM (Execute/Memory)	
MemToReg	0
RegWrite	1
MemRead	0
MemWrite	0
Branch	0
BranchTarget	X
isZero?	X
ALURes	15
RD2/Op2	X
RegDest (DstRNum)	$25

MEM/WB (Memory/Write Back)	
MemToReg	0
RegWrite	1
MemRes	X
ALURes	15
RegDest (DstRNum)	$25

3. Given the following three formulas (See Lecture #9, Section 5 Performance),

$$CT_{seq} = \sum_{k=1}^{N} T_k$$

$$CT_{pipeline} = \max(T_k) + T_d$$

$$Speedup_{pipeline} = \frac{CT_{seq} \times InstNum}{CT_{pipeline} \times (N + InstNum - 1)}$$

For each of the following processor parameters, calculate CT_{seq}, $CT_{pipeline}$ and $Speedup_{pipeline}$ (to two decimal places) for 10 instructions and for 10 million instructions.

	Stages Timing (for 5 stages, in ps)	Latency of pipeline register (in ps)
a.	300, 100, 200, 300, 100 (slow memory)	0
b.	200, 200, 200, 200, 200	40
c.	200, 200, 200, 200, 200 (ideal)	0

a. Stage Timing for 5 stages (in ps): 300, 100, 200, 300, 100. Latency of pipeline register: **0**ps

$CT_{seq} = 300 + 100 + 200 + 300 + 100 = 1000ps$

$CT_{pipeline} = \max(300, 100, 200, 300, 100) = 300ps$

Speedup (10 instructions) = (1000*10) / (300*14) = 2.38

Speedup (10 million instructions) = $(1000*10^7) / (300*(10^7+4)) = 3.33$

b. Stage Timing for 5 stages (in ps): 200, 200, 200, 200, 200. Latency of pipeline register: **40**ps

$CT_{seq} = 200 + 200 + 200 + 200 + 200 = 1000ps$

$CT_{pipeline} = \max(200+40, 200+40, 200+40, 200+40, 200+40) = 240ps$

Speedup (10 instructions) = $(1000*10) / (240*14) = 2.98$

Speedup (10 million instructions) = $(1000*10^7) / (240*(10^7+4)) = 4.17$

c. Stage Timing for 5 stages (in ps): 200, 200, 200, 200, 200. Latency of pipeline register: **0**ps

CT_{seq} = 200 + 200 + 200 + 200 + 200 = 1000ps

$CT_{pipeline}$ = max(200, 200, 200, 200, 200) = 200ps

Speedup (10 instructions) = (1000*10) / (200*14) = 3.57

Speedup (10 million instructions) = $(1000*10^7) / (200*(10^7+4))$ = 5.00

Slides uploaded to https://github.com/michaelyql/IT5002

Email: e1121035@u.nus.edu

Anonymous feedback: https://bit.ly/feedback-michael

(or scan the QR below)