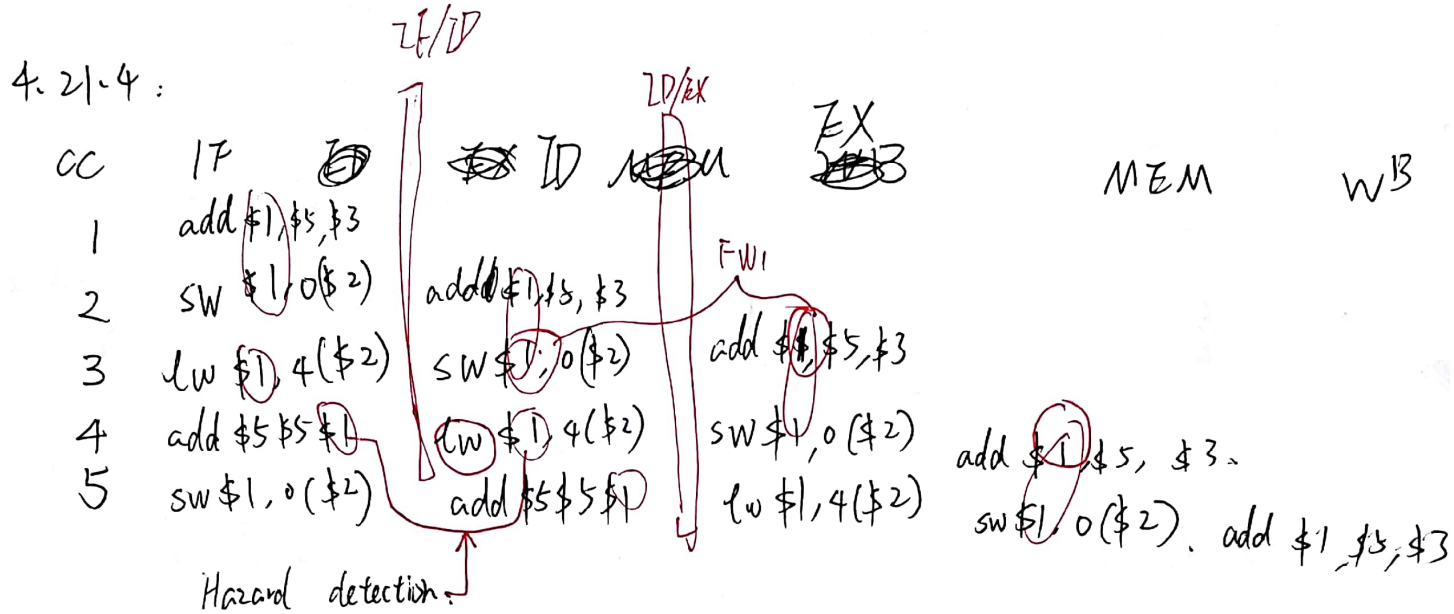
VE370

Homework 6

Bingcheng

516021910219

4.21.4:



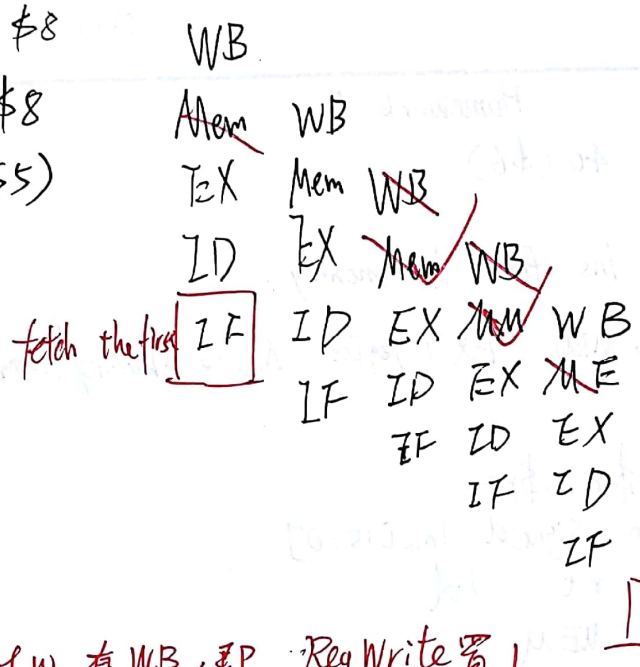
CC	Stall?	ForwardA	ForwardB
1	no	\	\
2	no	\	\
3	no	\	\
4	no	00	10
5	Yes		

4.16.4

a. Loop

将
- 2: add $5, $5, $8
- 2: add $6, $6, $8
- 2: sw $1, 20($5)
- 2. beq

全
- 3. lw
- 3. add
- 3. add
- 3. sw
- 3. beq.



仅 add, addi, lw 有 WB, 即 RegWrite 置 1
仅 sw 有 MemWrite, 即 MEM 有用
仅 lw 有 MemRead & MemtoReg 这两命令一定相同, 也有 Mem.
至于 beq: Branch & zero → PCSrc 在比较

14.16.5

如图. only $\frac{1}{5}$ clock cycles

4.16.6

PC+4,
Ins word for previous beq.

4.17_3

use data Mem: lw, sw,

4.18_1

for add: EX = { RegDst = 1, ALUOP = 10, ALUSrc = 0 }

要写入 Reg 时为 1: (R型, lw)

仅在用 sign-extention 为 1 (lw, sw, addi)

MEM = { Branch = 0, MemRead = 0, MemWrite = 0 }

WB = { RegWrite = 1, Mem2Reg = 1 }

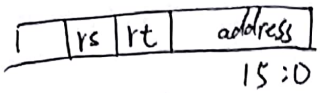
4.7.3 add - beq - lw - sw

means: ALU, shift Add, D-mem

And this means D-mem is the longest, so the answer is the same as 4.7.2.

4.8.1

(a) I-mem, out, bit 7.

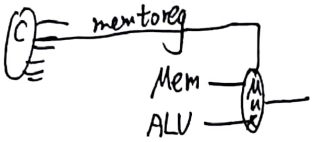


So we can test stuck at 0 by make address to be all 0 except bit 7:

addi $to, $to, 128 (128 is 10000000)

Then we check $to to see wether bit 7 is zero.

(b) Control unit, Mem to Reg:



Only when "lw" Mem to Reg = 1.

lw $to $zero, if $to ≠ 0, stuck at zero.

4.8.2

(a): $to origin at 128, then addi $to, $zero, 0

(b) No. not relible. If stuck at 1, then random write, may be the same as the value in register.

Because a signal can not be both 0 & 1. we can't.

4.11.2

(a) 10011 00010 00011 00000 0000 0|00000

op = 2^5+2^1+2^0 = 32+2+1 = 35 = lw

look up table - func

so ALU op = 00, Ins = 0|0000

(b) 00100₂ = 2^2 = 4 = beq

So ALU op = 01, Ins = 00100

4.18.3

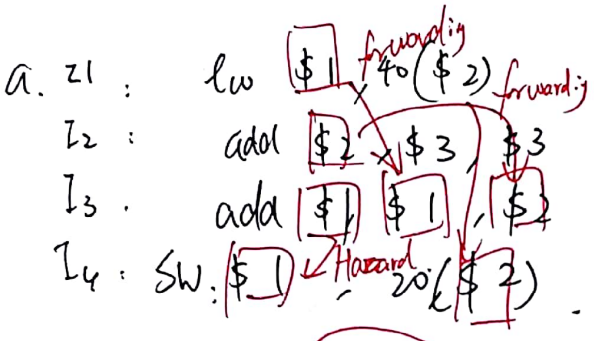
PCSrc = 0 (only J type, beq maybe 1)

In EX: faster to determin whether beq is real.

Such that can save one clock sycle for faulse.

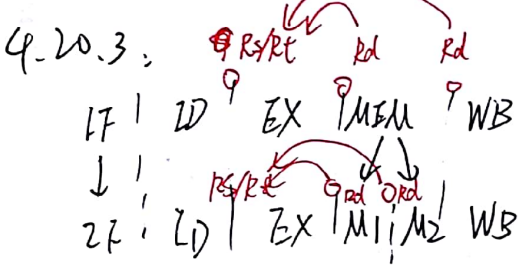
Not In Ex. May increase clock sycle time.

4.20.1



	RaferW	WaferR	WaferW
I1	$1 (I3, I4)	$2 (I2 I1)	$1 (I1, I3]
I2	$2 (I3, I2)		
I3	$1 (I3, I4)		
I4	$2 (I2, I4)		

4.20.3:



with forwarding.

($1) I1 to I3

4.20.5.

$0 = 0

$1 = 32

$2 = 2000

$3 = 1000

4.20.6

```
lw
add
nop
nop
add
nop
nop
sw.
```