JOINT INSTITUTE
交大密西根学院

Name: 胡炳城
Student ID: 516021910219
Course Code: VE370
Date:

Homework 8

4.16.1. (a) lw $1, 40($6)

IF/ID: ① PC+4
② fetched Ins from Ins memory.

ID/EX: ① WB, MEM, EX (RegDst, ALUOp, AluSrs//, Branch, memRead, MemWrite//RegWrite, MemtoReg)
② PC+4
③ ~~rd~~ $rs, $rt
④ ~~Singe~~ Signed Ins[15:0]
⑤ rs, rt, rd

EX/MEM: ① WB, MEM
② PC+4 + Signed extention (New PC)
③ zero
④ ALU out
⑤ ~~Data to write to DMem.~~ 疑问：这个是不是 Read Reg 2 ? 看图不是
⑤ Rt or Rd (According to RegDst). (Desination)

MEM/WB ① WB
② Data read from Dmem (or Random if no mem read)
③ ALU out
④ Rt or Rd (Destination to write on Reg).

4.16.2/3

	need to be read	Actually read	EX	MEM
a.	$6	$6, $1	40+$6	load from mem
b.	$5	$5, twice	$5+$5	×

~~Q3.~~ 4.18.3

PCSrc =0 (only ~~I type~~, beq make it 1)

In EX : faster to determin ~~what~~ whether beq is read.
Such that can save one clock sycle for faulse.

Not In Ex. May increase clock syde time.

4.20.~~1~~

a. I1 : lw $\boxed{\$1}$, 40($2) forwarding

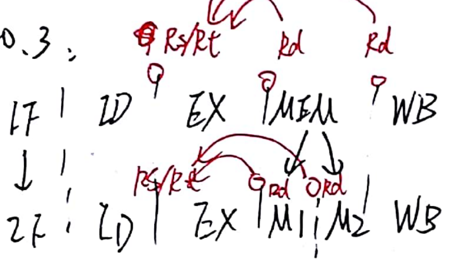
I2 : add $\boxed{\$2}$, $3, $3 forwarding

I3 : add $\boxed{\$1}$, $\boxed{\$1}$, $\boxed{\$2}$

I4 : sw $\boxed{\$1}$, Hazard 20($\boxed{\$2}$).

R after W	~~R~~ W after R	W after W
\$1 (I3, I1)	\$2 (I2, I1)	\$1 (I1, I3).
\$2 (I3, I2),		
\$1 (I3, 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 .

4.16.4

a、 Loop

待 2. add \$5, \$5, \$8		WB				
2: add \$6, \$6, \$8		~~Mem~~	WB			
2: sw \$1, 20(\$5)		EX	Mem	~~WB~~		
2~~1~~. beq		ID	EX	~~Mem~~	~~WB~~	
全 3. ~~add~~ lw	fetch the first	$\boxed{\text{IF}}$	ID	EX	~~Mem~~	WB
3. add			IF	ID	EX	~~M~~E
3. add				IF	ID	EX
3. sw					IF	ID
3. beq.						IF

$\boxed{\text{IF}}$ not include

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

4.16.5

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~~ ~~0~~ → 要写入Reg时为1：（R型，lw）

ALUOP = 10

ALUSrc = 0 → 仅在用 sign-extention 为1 （lw，sw，addi）

MEM： Branch = 0

Mem Read = 0

Mem Write = 0

WB： RegWrite = 1

Mem2Reg = 1

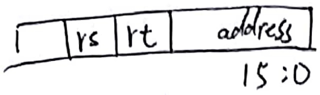
4.7.3 add - beq - lw - sw

means: ALU, shift Add, D-mem

And This means D-mem is the largest, so the answer is the same as 4.7.2.

4.8.1

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

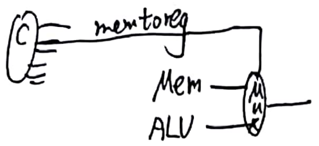


So we can test stuck at 0 by make address to be all 0 except bit 7:

addi $t0, $t0, 128 (128 is 10000000)

Then we check $t0 to see wether bit 7 is zero.

(b) Control unit, Mem to Reg:



Only when "lw" Mem to Reg =1.

lw $t0 $zero, if $t0 ≠ 0, stuck at zero.

~~4~~ 4.8.2

(a): $t0 origin at 128, then addi $t0, ~~$t0~~ $zero, ~~$zero~~ 0

(b) No. not relibe. 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 4.

4.11.2

(a) $\underline{100011}$ 00010 00011 00000 00000 $\underset{func}{\underline{010000}}$

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

so ~~4~~ $\underline{\text{ALU op}} = \overset{\text{look up table}}{\underline{00}}$, $\overset{func}{\text{Ins}} = \boxed{010000}$

(b) $\underline{000100}_2 = 2^2 = 4 =$ beq

So ALU op = 01, Ins = 001100

~~Q3.~~ 4.18.3

PCSrc = 0 (only ~~I-type~~, beq make it 1)

In EX: faster to determin ~~what~~ whether beq is real.
Such that can save one clock sycle for faulse.

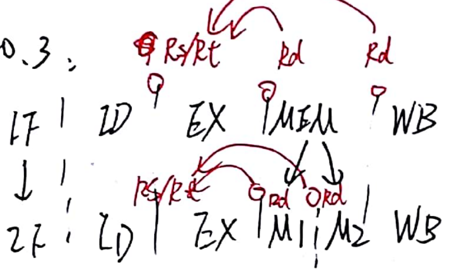
Not In EX: May increase clock sycle time.

4.20.~~1~~

			RafterW	Wafter R	Wafter W
a. I1 :	lw	$\boxed{\$1}$, 40($2)	$1 (I3, I4)	$2 (I2, I1)	$1 (I1, I3).
I2 :	add	$\boxed{\$2}$, $3, $3	$2 (I3, I2),		
I3 .	add	$\boxed{\$1}$, $\boxed{\$1}$, $\boxed{\$2}$	$1 (I3, I4).		
I4 :	SW:	$\boxed{\$1}$, 20($\boxed{\$2}$) .	$2 (I2, I4).		

forwarding
forwarding
Hazard

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 .