

計算機結構 HW4

b07902048 資工三 李宥霆

4.7

4.7.1

Register Read + I-Mem + Register File + Mux + ALU + Mux + Register Setup = $30 + 250 + 150 + 25 + 200 + 25 + 20 = 700\text{ps}$

4.7.2

Register Read + I-Mem + Register File + Mux + ALU + D-Mem + Mux + Register Setup = $30 + 250 + 150 + 25 + 200 + 250 + 25 + 20 = 950\text{ps}$

4.7.3

Register Read + I-Mem + Register File + Mux + ALU + D-Mem = $30 + 250 + 150 + 25 + 200 + 250 = 905\text{ps}$

4.7.4

Register Read + I-Mem + Register File + Mux + ALU + Single Gate + Mux + Register Setup = $30 + 250 + 150 + 25 + 200 + 5 + 25 + 20 = 705\text{ps}$

4.7.5

Register Read + I-Mem + Register File + Mux + ALU + Mux + Register Setup = $30 + 250 + 150 + 25 + 200 + 25 + 20 = 700\text{ps}$

4.7.6

950ps

4.27

4.27.1

```

add x15, x12, x11 //c = a + b
nop
nop
ld x13, 4(x15) //load d from c
ld x12, 0(x2)
nop
or x13, x15, x13 // d = c | d
nop
nop
sd x13, 0(x15) store d to c

```

4.27.2

rearrange無法使NOP變少

4.27.3

如果沒有hazard detection，在load後面馬上使用load出來的資料時就會出現錯誤，但因為上面的code中沒有出現此一情況，所以不會出現問題。

4.27.4

ins	1	2	3	4	5	6	7	8	9
add x15, x12, x11	IF	ID	EX	ME	WB				
ld x13, 4(x15)		IF	ID	EX	ME	WB			
ld x12, 0(x2)			IF	ID	EX	ME	WB		
or x13, x15, x13				IF	ID	EX	ME	WB	
sd x13, 0(x15)					IF	ID	EX	ME	WB

cycle	FowardA	FowardB
1	no	no
2	no	no
3	00	00
4	10	00
5	00	00
6	00	10
7	00	01

4.27.5

需要的input有：

1. ID/EX register的rd的值
2. EX/MEM register的rd的值
3. IF/ID register的rs1和rs2

舉上題的例子來說：

1. add跟第一個ld需要ID/EX的rd與IF/ID register的rs1
2. 第一個ld與or需要EX/MEM的rd與IF/ID register的rs2

但因為上述三項的第一項與第三項已經是hazard detector的input了，所以只需要增加第二項

需要的ouput有：

不須增加新的output

4.27.6

cycle	PC write	IF/ID write	Mux
1	1	1	0
2	1	1	0
3	0	0	1
4	0	0	1
5	1	1	0

4.28

4.28.1

$$1 + 0.25 \times 0.55 \times 1 = 1.1375 \text{倍}$$

4.28.2

$$1 + 0.25 \times 0.45 \times 1 = 1.1125 \text{倍}$$

4.28.3

$$1 + 0.25 \times 0.15 \times 1 = 1.0375 \text{倍}$$

4.28.4

$$1 + 0.25 \times 0.15 \times 1 \div 2 = 1.01875$$

$$\frac{1.0375}{1.01875} = 1.0184 \text{倍 speed up}$$

4.28.5

$$1 + 0.25 \times 0.15 \times 1 \div 2 + 0.25 \times 1 \div 2 = 1.18125$$

$$\frac{1.0375}{1.18125} = 0.907 \text{倍 speed up}$$

4.28.6

$$0.8 * (1) + 0.2 * (X) = 0.85$$

$$X = 0.25$$

4.29

4.29.1

always taken: 0.6

always not taken: 0.4

4.29.2

result: T, NT, T, T

predict: NT, NT, NT, NT

ans: 25%

4.29.3

到stable的狀態時，state transssition會是12112 12112 12112(1為深藍，2為淺藍)，而所有猜測都會是taken，因此準確度為60%

4.29.4

做出一個finite state machine M，有5個state(以下稱為A,B,C,D,E)，A,C,D代表T, B,E代表NT，而A->B->C->D->E->A... 依此下去，一開始從A出發，每次出現branch後就移到下一個state，這樣準確率就會是100%

4.29.5

若把patten反過來，準確率會變為0%(每次都猜錯)

4.29.6

一開始在state A上，看第一次的prediction是不是對的，如果是對的則繼續用finite state machine M，否則使用M'，M'是M的補集，A,C,D代表NT, B,E代表T，如此只需要一個warm up period就可以知道是那一個pattern。