

# Computer Architecture - Homework #3

資工三 陳盈如 B05902118

## 3.47 [45] <§§3.6, 3.7>

Ans:

我會把 128-bit 的 register 切成四等份的 32-bit register，每一份 32-bit register 代表一個 sign\_in[]，利用平行運算減少執行的時間，再將四個 instruction 得到的結果算出 sign\_out[]，如此一來就會加快 program 執行的速度。

---

## 4.15.1 [5] <§4.4>

Ans:

新的 clock cycle time 大略會是  $\frac{4}{5} \times T_{original}$ ，因為原本有 5 個 stage 現在少掉 ALU 或 MEM，剩下 4 個 stage。

## 4.15.2 [10] <§4.4>

Ans:

- (1) 新的 CPU 會跑比較慢
- (2) 用課本各 stage 的時間算的話，假設總共有 100 個 instruction：

原本： $800 \times 100 = 80000$

新的： $600 \times 100 + 600 \times (25 + 11) = 81600$

## 4.15.3 [5] <§4.4>

Ans:

主要影響一個 program 執行速度的因素是 instruction 的數目

## 4.15.4 [5] <§4.4>

Ans:

我覺得原本的 CPU 比較好，它可以簡化整個 architecture 使速度比較快，而且從 4.15.2 的假設來看，新的 CPU 執行速度反而比較原本的還要慢，因此我會選擇原本的 CPU。

#### 4.21.1 [5] <§4.5>

Ans:

$$\frac{250 \times 1.4n}{300 \times 1.05n} = \frac{350}{315} \approx 1.1, \text{快 } 1.1 \text{ 倍, 兩個速度沒差多少。}$$

#### 4.21.2 [10] <§4.5>

Ans:

$$1.4n \times 250 = (n + x) \times 300 \rightarrow x \approx 0.167n$$

#### 4.21.3 [10] <§4.5>

Ans:

$$0.4175x \text{ (x 以 } 0.4n \text{ 代替)}$$

#### 4.21.4 [10] <§4.5>

Ans:

$$\text{有可能, } \frac{250 \times 1.4n}{250 \times \frac{0.075}{0.4} \times 8 \times \frac{6}{5} \times 1.075n} \approx \frac{350}{336} \approx 1.04, \text{因為算出來是有可能的, 當 program 只有很少的}$$

data hazard就有可能發生。

#### 4.21.5 [10] <§4.5>

Ans:

A program must have  $0.167n$  NOPs before it can possibly run faster on the pipeline with forwarding .

---

#### 4.27.1 [5] <§4.7>

Ans:

add -> NOP -> NOP -> ld -> ld -> NOP -> or -> NOP -> NOP -> sd

#### 4.27.2 [10] <§4.7>

Ans:

我們可以藉由新增 **addi x17, x17, 0** 來減少 NOP 的數量。

#### 4.27.3 [10] <§4.7>

Ans:

**or x13, x15, x13** 的 x13 會拿到舊的 x13, **sd x13, 0(x15)** 的 x13 也會拿到舊的 x13

#### 4.27.4 [20] <§4.7>

Ans:

	0	1	2	3	4	5	6
add x15, x12, x11	IF	ID	EX	MEM	WB		
ld x13, 4(x15)		IF	ID	EX	MEM	WB	
ld x12, 0(x2)			IF	ID	EX	MEM	WB
or x13, x15, x13				IF	ID	EX	MEM
sd x13, 0(x15)					IF	ID	EX
forwarding				ForwardA		ForwardB	ForwardB

#### 4.27.5 [10] <§4.7>

Ans: 新增 input: IF/ID RS, ID/EX RD, EX/MEM RD，檢查 IF/ID RS 和 ID/EX RD 是否不一樣，或者是 IF/ID RS 和 EX/MEM RD 是否不一樣，若這兩種情況有一種是一樣的話，就要將 output signal 的 PC Write 和 IF/ID Write 改成 0，把 ID/EX 改成 1。

#### 4.27.6 [20] <§4.7>

Ans:

	0	1	2	3	4
add x15, x12, x11	IF	ID	EX	MEM	WB
ld x13, 4(x15)		IF	bubble	bubble	ID
ld x12, 0(x2)			IF	ID	EX
or x13, x15, x13				IF	ID
sd x13, 0(x15)					IF
PC Write	1	1	0	0	1

#### 4.28.1 [10] <§4.8>

Ans:

一個 mispredicted 的 branch 會造成兩個 stall cycle，因此 extra CPI 是  $0.25 \times 0.55 \times 2 = 0.275$ 。

#### 4.28.2 [10] <§4.8>

Ans:

extra CPI 是  $0.25 \times 0.45 \times 2 = 0.225$

#### 4.28.3 [10] <§4.8>

Ans:

extra CPI 是  $0.25 \times 0.15 \times 2 = 0.075$

#### 4.28.4 [10] <§4.8>

Ans:

假設總共有  $n$  個 instruction，在 convert 以前總共有  $n + 0.25 \times 0.15 \times 2 \times n = 1.075n$  個 cycle，convert

之後有  $n - 0.125n + 0.125n + 0.125 \times 0.15 \times 2 \times n = 1.0375n$  個 cycle，速度增快為  $\frac{1.075}{1.0375} \approx 1.036$  倍。

#### 4.28.5 [10] <§4.8>

Ans:

在 convert 之後有  $n - 0.125n + 2 \times 0.125n + 0.125 \times 0.15 \times 2 \times n = 1.1625n$  個 cycle，因此速度減慢為

$\frac{1.1625}{1.075} \approx 0.925$  倍。

#### 4.28.6 [10] <§4.8>

Ans:

剩下 20% 的 branch instruction 的正確率為 25%

---

### 4.31.1 [30] <§4.10>

Ans:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
li x12, 0	IF	ID	EX	MEM	WB																				
jal ENT	IF	ID	EX	MEM	WB																				
bne x12, x13, TOP		IF	ID	EX	MEM	WB																			
slli x5, x12, 3			IF	ID	EX	MEM	WB																		
add x6, x10, x5				IF	ID	EX	MEM	WB																	
ld x7, 0(x6)					IF	ID	EX	MEM	WB																
ld x29, 8(x6)						IF	ID	EX	MEM	WB															
sub x30, x7, x29								IF	ID	EX	MEM	WB													
add x31, x11, x5									IF	ID	EX	MEM	WB												
sd x30, 0(x31)										IF	ID	EX	MEM	WB											
addi x12, x12, 2										IF	ID	EX	MEM	WB											
bne x12, x13, TOP											IF	ID	EX	MEM	WB										
slli x5, x12, 3												IF	ID	EX	MEM	WB									
add x6, x10, x5													IF	ID	EX	MEM	WB								
ld x7, 0(x6)														IF	ID	EX	MEM	WB							
ld x29, 8(x6)															IF	ID	EX	MEM	WB						
sub x30, x7, x29																	IF	ID	EX	MEM	WB				
add x31, x11, x5																		IF	ID	EX	MEM	WB			
sd x30, 0(x31)																			IF	ID	EX	MEM	WB		
addi x12, x12, 2																				IF	ID	EX	MEM	WB	
bne x12, x13, TOP																					IF	ID	EX	MEM	WB

### 4.31.2 [10] <§4.10>

Ans:

因為 loop 執行數千次 instruction，因此 loop 的 instruction 數會是影響執行時間的主要因素。又因為 **sd x30, 0(x31)** 和 **addi x12, x12, 2** 利用 two-issue processor 可以同時執行，所以速度會增快為大約 1.1 倍。

### 4.31.3 [10] <§4.10>

Ans:

	li x12, 0
	slli x13, x13, 3
	add x28, x10, x13
	jal ENT
TOP:	ld x7, 0(x10)
	ld x29, 8(x10)
	sub x30, x7, x29
	addi x10, x10, 16
	sd x30, 0(x11)
	addi x11, x11, 8
ENT:	bne x11, x28, TOP

### 4.31.4 [20] <§4.10>

Ans:

	ALU or branch instruction	Data transfer instruction
	slli x13, x13, 3	li x12, 0
	add x28, x10, x13	
	jal ENT	
TOP:		ld x29, 8(x10)
	sub x30, x7, x29	
	addi x10, x10, 16	sd x30, 0(x11)
	addi x11, x11, 8	
ENT:	bne x11, x28, TOP	ld x7, 0(x10)

### 4.31.5 [30] <§4.10>

Ans:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
li x12, 0	IF	ID	EX	MEM	WB																
slli x13, x13, 3	IF	ID	EX	MEM	WB																
add x28, x10, x13		IF	ID	EX	MEM	WB															
jal ENT			IF	ID	EX	MEM	WB														
bne x11, x28, TOP				IF	ID	EX	MEM	WB													
ld x7, 0(x10)				IF	ID	EX	MEM	WB													
ld x29, 8(x10)					IF	ID	EX	MEM	WB												
sub x30, x7, x29							IF	ID	EX	MEM	WB										
addi x10, x10, 16								IF	ID	EX	MEM	WB									
sd x30, 0(x11)								IF	ID	EX	MEM	WB									
addi x11, x11, 8									IF	ID	EX	MEM	WB								
bne x11, x28, TOP										IF	ID	EX	MEM	WB							
ld x7, 0(x10)										IF	ID	EX	MEM	WB							
ld x29, 8(x10)											IF	ID	EX	MEM	WB						
sub x30, x7, x29													IF	ID	EX	MEM	WB				
addi x10, x10, 16															IF	ID	EX	MEM	WB		
sd x30, 0(x11)																IF	ID	EX	MEM	WB	
addi x11, x11, 8																IF	ID	EX	MEM	WB	
bne x11, x28, TOP																	IF	ID	EX	MEM	WB

### 4.31.6 [10] <§4.10>

Ans:

因為在 loop 內的 **addi x10, x10, 16** 和 **sd x10, 0(x11)** 利用 two-issue processor 可以同時執行，並且將 loop 內的 **ld x7, 0(x10)** 移動到 ENT 再利用 two-issue processor 與 **bne x11, x28 TOP** 同時執行，因此速度增快為 1.4 倍。

### 4.31.7 [10] <§4.10>

Ans:

	li x12, 0
	slli x13, x13, 3
	add x28, x10, x13
	jal ENT
TOP:	ld x7, 0(x10)
	ld x29, 8(x10)
	ld x8, 16(x10)
	ld x9, 24(x10)
	sub x30, x7, x29
	sub x31, x8, x9
	sd x30, 0(x11)
	sd x31, 8(x11)
	addi x10, x10, 32
	addi x11, x11, 16
ENT:	bne x11, x28, TOP

### 4.31.8 [20] <§4.10>

Ans:

	ALU or branch instruction	Data transfer instruction
	slli x13, x13, 3	li x12, 0
	add x28, x10, x13	
	jal ENT	
TOP:		ld x29, 8(x10)
	sub x30, x7, x29	ld x8, 16(x10)
		ld x9, 24(x10)
	sub x31, x8, x9	sd x30, 0(x11)
	addi x10, x10, 32	sd x31, 8(x11)
	addi x11, x11, 16	
ENT:	bne x11, x28, TOP	ld x7, 0(x10)



### 4.31.9 [10] <§4.10>

Ans:

從 one-issue processor 改成 two-issue processor，速度加快 1.7 倍

### 4.31.10 [30] <§4.10>

Ans:

	slli x13, x13, 3	li x12, 0
	jal ENT	add x28, x10, x13
TOP:		ld x29, 8(x10)
	sub x30, x7, x29	ld x8, 16(x10)
		ld x9, 24(x10)
	sub x31, x8, x9	sd x30, 0(x11)
	addi x10, x10, 32	sd x31, 8(x11)
	addi x11, x11, 16	
ENT:	bne x11, x28, TOP	ld x7, 0(x10)

因為可以同時執行兩個 arithmetic/logic instructions，因此在 loop 之前減少了一個 cycle。