20181210 유광호

4.7

I Mem / D mem	Register file	Mux	ALU	Adder
250ps	150ps	25ps	180ps	140ps

Single gate	Register Read	Register setup	Sign extend	Control
5ps	30ps	30ps	60ps	50ps

4.7.1

(PC) adder control

Register read + I mem + (read) register file + mux + ALU + mux + reg setup

sgate + mux

30 250 150 25 180 25 30

Latency of R type : 690 ps

4.7.2

(PC) adder control sign extend

Register read + I mem + reg file + MUX + ALU + Dmem + mux + reg setup

sgate + mux

30 250 150 25 180 250 25 30

Latency of lw : 940 ps

4.7.3

adder control sign extend sgate + mux

Register read + I mem + reg file + MUX + ALU + Dmem

30 250 150 25 180 250

Latency of sw : 885 ps

4.7.4

adder control sign extend adder sgate + mux (pc update)

Reg read + I mem + reg file + mux + ALU + sgate + mux + reg setup

30 250 150 25 180 5 25 30

Latency of beq : 695 ps

4.7.5

adder control sign extend adder

Register read + I mem + reg file + mux + ALU + mux + reg setup

30 250 150 25 180 25 30

Latency of shift i type : 690ps

4.7.6

Minimum clock period (critical path) : 940 ps

4.8 R type I type lw sw beq

25% 27% 25% 11% 12%

4.8.1 lw + sw 36%

4.8.2 100%

4.8.3 I type + lw + sw + beq 75%

4.8.4 old CPU : 940ps

new CPU : $0.25 \times 690 + 0.27 \times 690 + 0.25 \times 940 + 0.11 \times 885 + 0.12 \times 695 = 774.55$ps

21% faster

4.16 IF ID EX MEM WB

250ps 350ps 150ps 400ps 200ps

ALU/Logic Jump/Branch Load Store

45% 15% 25% 15%

4.16.1 pipelined : 400ps non-pipelined : 1350ps

4.16.2 pipelined : $5 \times 400 = 2000$ps non-pipelined : 1350ps

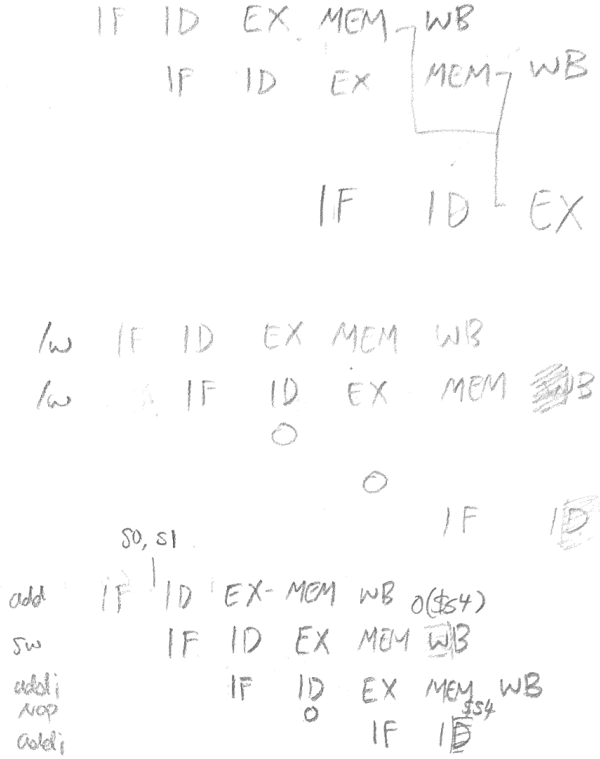
4.16.3 MEM stage $\max(IF, ID, EX, \frac{MEM}{2}, WB) = 350$ps

4.16.4 reg write = 1 when r format, lw

45 + 25 = 70%

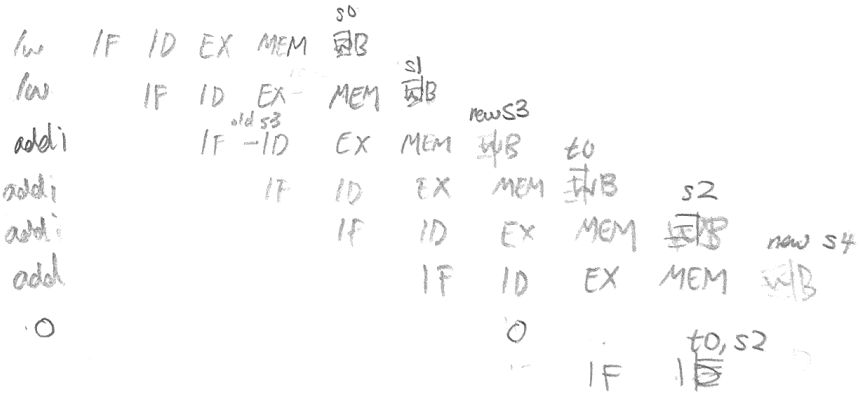
4.25.1

```
LOOP:
    lw $s0, 0($s3)
    lw $s1, 4($s3)
    NOP
    NOP
    add $s2, $s0, $s1
    sw $s2, 0($s4)
    addi $s3, $s3, -8
    NOP
    addi $s4 $s4, 4
    bnez $s2, LOOP
```



4.25.2

```
LOOP:
    lw $s0, 0($s3)
    lw $s1, 4($s3)
    addi $s3, $s3, -8
    addi $t0, $s4, 0
    addi $s2, $s0, $s1
    addi $s4, $s4, 4
    NOP
    sw $s2, 0($t0)
    bnez $s2, LOOP
```



4.25.3

```
lw   $s0, 0($s3)
lw   $s1, 4($s3)
NOP
add  $s2, $s0, $s1
sw   $s2, 0($s4)
addi $s3, $s3, -8
addi $s4, $s4, 4
bnez $s2, LOOP
```

2nd iter

```
lw   $s0, 0($s3)
lw   $s1, 4($s3)
NOP
add  $s2, $s0, $s1
```

```
sw   $s2, 0($s4)
addi $s3, $s3, -8
addi $s4, $s4, 4
bnez $s2, LOOP
```