CENG/CSCI 3420 Computer Organization

Mid-term Examination Solution

1. (1) Advantages: Shorter Clock Cycle, higher through-put and higher resource usage.

Disadvantages: New hazards, complicated control unit, and higher hardware cost.

(2) 1624

2. One possible version:

```
1  ; If we consider the d is a byte type
2        add $t0, $zero, $zero      ; initialize i = 0
3        addi $t1, $zero, 100       ; set the bound to exit
4        add $a0, $zero, 0          ; set the constant to be stored
5        add $a1, $zero, 1
6  loop: beq $t0, $t1, exit  ; determine whether we should exit loop
7        andi $s0, $t0, 1    ; calculate i % 2
8        addi $s1, $s2, $t0  ; calculate the address to be stored
9        bne $s0, $zero, else  ; jump if i % 2 != 0
10       sb $a1, 0($s1)      ; store the byte to memory
11       j exif               ;
12 else: sb $a0, 0($s1)
13 exif: addi $t0, $t0, 1    ; i++
14       j loop              ; jump back to loop
15 exit:
```

```
1  ; if we consider the d is a word type
2        add $t0, $zero, $zero      ; initialize i = 0
3        addi $t1, $zero, 100       ; set the bound to exit
4        add $a0, $zero, 0          ; set the constant to be stored
5        add $a1, $zero, 1
6  loop: beq $t0, $t1, exit  ; determine whether we should exit loop
7        andi $s0, $t0, 1    ; calculate i % 2
8        slti $t2, $t0, 2    ; calculate $t0 * 4
9        addi $s1, $s2, $t2  ; calculate the address to be stored
10       bne $s0, $zero, else ; jump if i % 2 != 0
11       sw $a1, 0($s1)      ; store the word to memory
12       j exif
13 else: sw $a0, 0($s1)
14 exif: addi $t0, $t0, 1    ; i++
15       j loop              ; jump back to loop
16 exit:
```

3. (1) The average CPI for Processor 1 is $\frac{2720\times1+1440\times4+600\times2}{2720+1440+600} = 2.0336$

The average CPI for Processor 2 is $\frac{2720\times1+1440\times3+600\times3}{2720+1440+600} = 1.8571$

(2) $Execution\ Time = Cycle\ Time \times CPI \times Instruction\ Count$

Cycle time for Processor 1 is 1/2GHz = 0.5ns. The execution time for Processor 1 is $0.5ns \times 2.0336 \times 4760 = 4840ns = 4.84\mu s$

Cycle time for Processor 2 is 1/1.5GHz = 0.67ns. The execution time for Processor 1 is $0.67ns \times 1.8571 \times 4760 = 5923ns = 5.92\mu s$

4. (1)

Box 1: The missing part here response of two tasks. One is shitfing the 32 bit immediate left by 2 bits. Another one is to calculate the sum of PC+4 and the shifted immediate number.

Box 2: We need a PC register here.

Box 3: We need a multiplexer here to select the input source for Register Write data port.

Box 4: A sign extension unit here to signed-extend the 16-bit immediate to 32 bit.

(2)

This signal controls whether write the jump target address to PC register. So "j" instruction will set this signal to one.

5.

(1)

Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LW	IF	ID	EX	MEM	WB									
BEQ			IF	ID	EX	MEM	WB							
ADDI						IF	ID	EX	MEM	WB				
ADD							IF	ID	EX	MEM	WB			
BEQ								IF	ID	EX	MEM	WB		
BEQ									IF	ID	EX	MEM	WB	
SW										IF	ID	EX	MEM	WB

(2)

```
1         LW R2,0(R1)
2 Label1: SEQ R8, R2, R0
3         BNEZ R8, Label2
4         ADDI R2,R0,0
5         ADD R3,R2,R1
6         SEQ R8, R3, R0
7         BNEZ R8, Label1
8         ADD R1, R3, R1
9 Label2: SW R1, 0(R2)
```

(3) In the above shown code, there are two new hazards that may happen when we support branch execution in ID stage.

Line 1 and Line 2 show a kind of hazard: one load instruction loading values which is needed by a branch instruction just after it. The pipeline should be stalled 2 cycles.

Line 4 and Line 5 shows another kind of hazard: one R-type instruction is just before one branch instruction in need of its result. The pipeline should be stalled for one cycle.

Summarizing above two situations, we need a forwarding logic from EX/MEM to ID (for second case) and a forwarding logic from MEM/WB to ID (for first case). We should also have several multiplexers to select the source for branch comparators: from Register files, or the ALU output from EX/MEM registers or from MEM/WB pipeline registers.