

※請在答案卷內作答

一、[15%]

- (a) [4%] For a 32-bit address space, calculate the total number of bits required for the cache listed below: 32 KiB direct-mapped, write-back data cache, 2 words per cache block, and 2 control bits per cache block.
- (b) [4%] Given that cache size in problem (a), please find the total size of the closet direct-mapped, write-back data cache with 16-word blocks of equal size or greater. Explain why this data cache, despite its larger size, might provide slower performance than that in (a).
- (c) [7%] A generic memory hierarchy would consist of a TLB and a cache. A memory reference can encounter three different types of misses: a TLB miss, a page fault, and a cache miss. Consider all the combinations of these three events with one or more occurrences. There are seven possibilities. For each possibility, please state whether this event can actually occur and under what circumstances.

二、[8%] Regarding computer arithmetic, is the following statement “True” or “False”? If True, please give a brief explanation. If False, please give a counter example.

- (a) [4%] Associativity holds for a sequence of two’s complement integer additions, even if the computation overflows.
- (b) [4%] Associativity holds for a sequence of floating-point additions.

三、[12%] When making changes to optimize part of a computer, it is often the case that speeding up one type of instructions comes at the cost of slowing down something else.

- (a) [3%] Suppose that floating-point operations take 20% of the original program’s execution time and the new fast floating-point unit speeds up floating-point operation by, on average, 2 times. Ignoring the penalty to any other instructions, what is the overall speedup?
- (b) [6%] Suppose that speeding up the floating-point unit would slow down data cache accesses, resulting in a 1.5 times slowdown. Suppose that data cache accesses consume 10% of the execution time. What is the overall speedup?
- (c) [3%] After implementing the new floating-point unit, what percentage of execution time is spent on floating-point operations? What percentage is spent on data cache accesses?

注：背面有試題

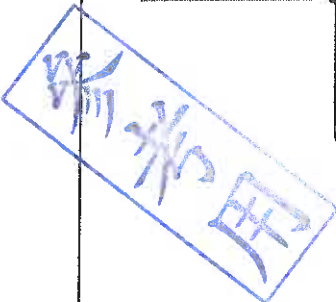
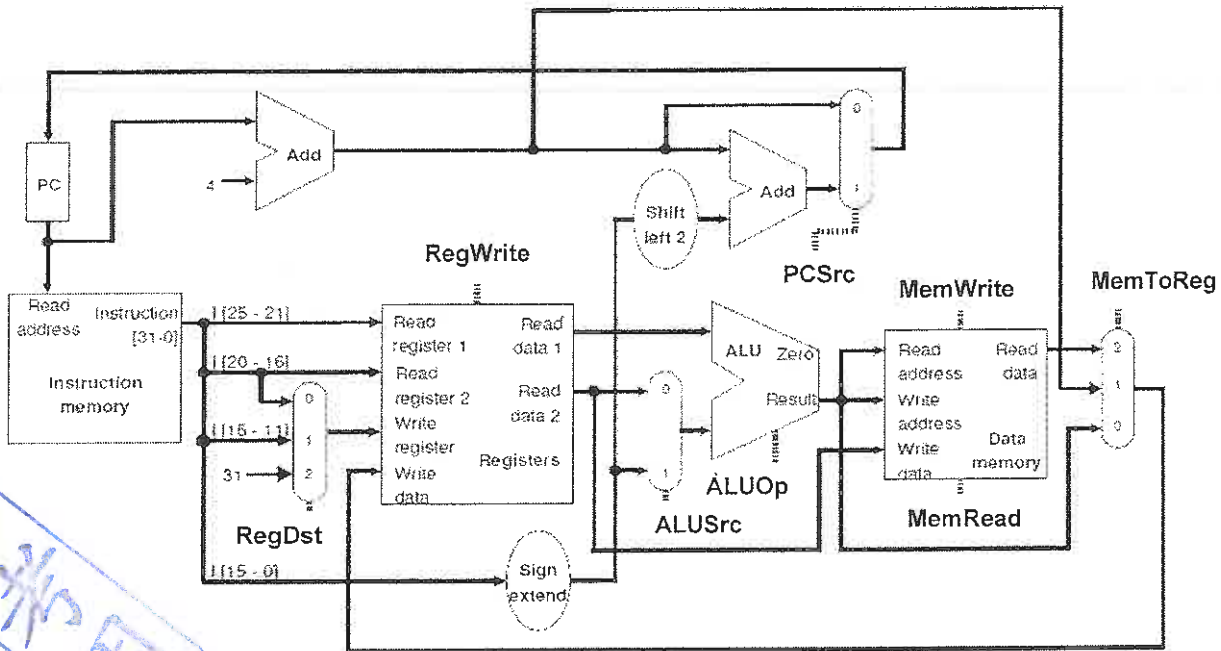
修書用

※請在答案卷內作答

四、[25%] A new I-type format instruction **swu** has been added to the MIPS instruction set. Its format is **swu rt, l(rs)**. It takes arguments register **rt**, register **rs**, and immediate **l**, and it stores the contents of **R[rt]** at the memory address **(R[rs] + l)** and then increments **R[rs]** by **l**.

(a) [8%] Given the single-cycle datapath in the following (control signals are marked with dashed lines), fill in the blanks in the table below for this new instruction. You must give the control signals (0, 1, 2, X) for the MIPS instruction. Each control signal must be specified as 0, 1, 2 or X (don't care). Writing a 0 or 1 when an X is more accurate is *not* correct.

Opcode	RegDst	RegWrite	ALUSrc	ALUOp	MemWrite	MemRead	MemToReg	PCSrc
swu								



This new I-type format instruction **jin imm16(rs)** is developed for the single-cycle processor, which is a Jump Indirect instruction and will cause the processor to jump to the address stored in the word at memory location **imm16 + R[rs]** (the same address computed by **lw** and **sw**).

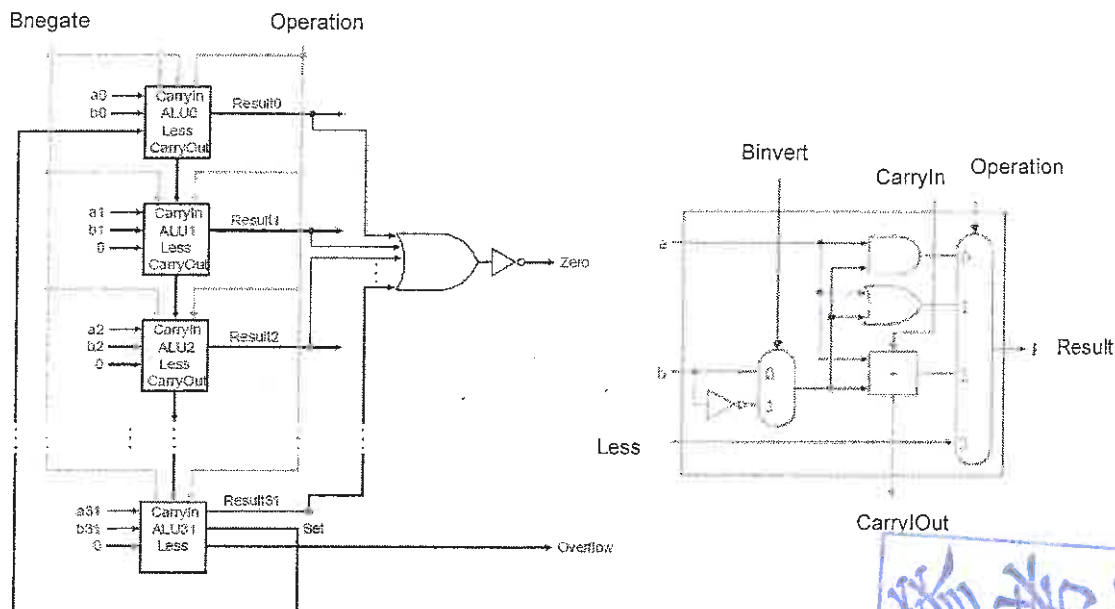
- (b) [5%] Draw the necessary modifications to implement the **jin** instruction on your sheet according to the figure of the single-cycle datapath provided above.
- (c) [4%] What is/are the new control signal(s) required to implement **jin**? Why is/are the control signal(s) necessary?

Consider the 32-bit ALU design shown in the next page. Now suppose that we wish to add hardware support for **xor** – exclusive-OR. For example, **xor \$t0, \$t1, \$t2**.

注意：背面有試題

※請在答案卷內作答

- (d) [8%] Please clearly describe (1) the necessary changes to the ALU hardware showing the changes to the i^{th} ALU bit position by modifying the figure below, (2) the corresponding values of the ALU control signals, and (3) describe briefly how the single-cycle datapath would operate with these changes.



參考用

五、[10%] Answer the following questions with respect to the MIPS program shown below. Note that this simple program does not use the register saving conventions followed in class. Assume that each instruction is a native instruction and can be stored in one word! Further, assume that the data segment starts at 0x10001000 and that the text segment starts are 0x00400000.

- (a) [2%] What does the program do?
- (b) [2%] State the values of the labels loop and main?
- (c) [4%] Please finish the hexadecimal encodings of the following instructions?

bne \$8, \$0, loop

Op-Code (6-bit)	1 st source register (5-bit)	2 nd source register (5-bit)	OFFSET (16-bit)
5			

add \$9, \$9, \$22

Op-Code (6-bit)	1 st source register (5-bit)	2 nd source register (5-bit)	Destination register (5-bit)	Shift amount (5-bit)	Function code (6-bit)
0					0x20

- (d) [2%] Identify the local and global labels in the program?

```

.data
label: .word 8,16,32,64
       .byte 64, 32
.text
.globl main
main:  la $4, label
       li $5, 16
       jal func
done
func:  move $2, $4
       move $3, $5
       add $3, $3, $2
       move $9, $0
loop: lw $22, 0($2)
       add $9, $9, $22
       addi $2, $2, 4
       slt $8, $2, $3
       bne $8, $0, loop
       move $2, $9
       jr $31
    
```

注意：背面有試題

※請在答案卷內作答

六、[12%] Given a 5-stage pipelined MIPS ISA design, the individual pipeline stages are named as IF, ID, EX, MEM, and WB, respectively. The latencies of five stages are given as 350 ps, 250 ps, 280 ps, 400 ps, 280 ps, respectively.

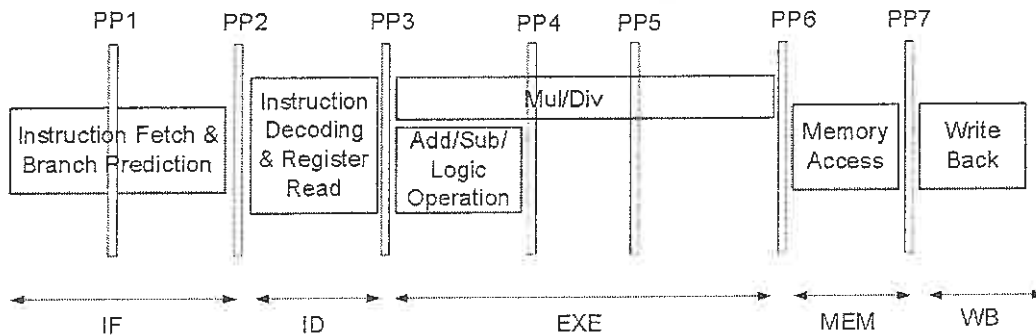
- (a) [3%] What is the maximum operating frequency for this processor?
- (b) [6%] The following fragment of MIPS codes is being executed. If there is no forwarding or hazard detection, please insert **nops** and rewrite the assembly to ensure correct execution.

```

add    $t4, $s2, $t1
or     $t2, $t1, $t2
lw     $s4, 20($t4)
sw     $t1, 16($t4)
sub    $s3, $s4, $t2
    
```

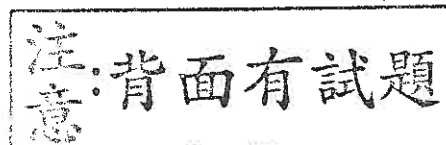
- (c) [3%] How many clock cycles are required to complete the execution of these instructions?

七、[12%] Assume that a pipelined processor has 8 pipelined stages as shown in the following. Due to the branch prediction, the IF stage needs two clock cycles, called IF1 and IF2. The EXE stage needs three clock cycles to support multiplication and division. However, the addition, subtraction, and logic operations are still completed in one clock cycle. The three EXE stages are named EXE1, EXE2, and EXE3. The latencies for 8 stages are 150 ps, 150 ps, 100 ps, 200 ps, 200 ps, 180 ps, 200 ps, and 100 ps, respectively. The pipeline registers are named from PP1 to PP7 sequentially.



Consider the following MIPS code.

1	lw	\$s2, 0(\$t4)
2	add	\$t0, \$s2, \$t1
3	mul	\$t2, \$s3, \$s5
4	sub	\$t0, \$t0, \$t2
5	sw	\$t0, -4(\$t4)



※請在答案卷內作答

- (a) [3%] If there is no forwarding or hazard detection, how many **nops** instructions should be inserted between the first instruction (**lw**) and the second instruction (**add**)? Please also give some simple explanation.
- (b) [3%] If a forwarding path from the memory output result in pipeline registers PP7 to the input of ALU at EXE1 stage is added to reduce the delay, how many **nops** instructions should be inserted between the first instruction (**lw**) and the second instruction (**add**)? Please also give some simple explanation.
- (c) [3%] How to add the forwarding path to solve the data hazard with the minimum delay between instruction 3 (**mul**) and instruction 4(**sub**)? Please also give some simple explanation.
- (d) [3%] How to add the forwarding path to solve the data hazard with the minimum delay between instruction 4 (**sub**) and instruction 5(**sw**)? Please also give some simple explanation.

八、[6%] Assume an 8-core computer system can process database queries at a steady state rate of requests per second and each transaction averagely takes a fixed amount time to process. Some results regarding the transaction latency and processing rate are given in the following table. How many requests are being processed at any given instant per core for the following 2 cases? Please write the values for (a) and (b), respectively.

Case	Average Transaction Latency	Maximum Transaction Processing Rate	No. of Requests per Core
1	1 ms	10,000/sec	(a)
2	2 ms	24,000/sec	(b)

參考用