※ 注意：全部題目均請作答於試卷內之「非選擇題作答區」，請標明題號依序作答。

1. (10 分，複選題) In a pipeline computer, which issues in the following are reasons that the pipeline cannot always be full ?
   (A) data register dependencies.
   (B) conditional branch statements.
   (C) functional unit contention.
   (D) out of order execution.
   (E) expression strength reduction.

2. (10 分，複選題) We want to execute the following loop in parallel.

```
for (i = 0; i < 1024768; i++) {
  if (A[i] > 0) {
    A[i] = A[i]*B[i];
    B[i] = A[i]+B[i];
  }
}
```

   Assume that the if statement, addition, and multiplication can all be done in one cycle of a core. Which of the following statements are true?
   (A) For a processor with more than 64 cores, the cache coherency issue may prevent the program from being directly parallelized.
   (B) For a 16-core processor, control flow conflicts may prevent the program from directly being parallelized.
   (C) For an 8-core processor, data conflicts may prevent the program from directly being parallelized.
   (D) If we use a SIMD parallel computer with 256 processors, in the worst case the processor utilization factor is 259/1024. Assume that all threads can be fully parallelized and all data items have been arranged in the vector registers to the processors.
   (E) If we use a SIMD parallel computer with 16 processors, in the worst case the processor utilization factor is 3/8. Assume that all threads can be fully parallelized and all are data items have been arranged in the vector registers to the processors.

見背面

3. (10 分，複選題) Consider two different implementations, *M1* and *M2*, of the same instruction set. *M1* has a clock rate of 8 GHz and *M2* has a clock rate of 5 GHz. There are three classes of instructions (*A*, *B*, and *C*) in the instruction set. The average number of cycles for each instruction class on *M1* and *M2* is given in the following table.

Instruction classes	*M1* (8GHz)	*M2* (5GHz)
A	2	1
B	3	3
C	6	3

Now there are two compilers, *C1* and *C2*, for C++ programs to this instruction set. In average, for the same program, *C1* generates 20% more instructions than *C2*. The following table summarizes the average proportion of instruction classes in machine programs generated by *C1* and *C2*.

Instruction class	*C1*	*C2*
A	40%	50%
B	30%	30%
C	30%	20%

Please answer which of the following statements are correct. All percentage numbers are round off to 0.1%. For example, 2.833% is round off to 2.8% while 2.875% is round off to 2.9%.

(A) Using *C1*, in average, *M1* is 2.8% faster than *M2* running the same benchmark.

(B) Using *C2*, in average, *M1* is 1.1% slower than *M2* running the same benchmark.

(C) Using *M1*, the same machine program generated by *C1* runs 15.3% faster than that generated by *C2* in average.

(D) Using *M2*, the same machine program generated by *C1* runs 24.2% slower than that generated by *C2* in average.

(E) In average, for the same C++ source program, running *C1* on *M1* is 3.9% slower than running *C2* on *M2*.

4. (10 分，複選題) The effective memory access time for a microprocessor with one level of cache is 3.0 clock cycles. Read/write to the cache costs one clock cycle while a cache miss costs 100 clock cycles. Which statements in the following are correct ?

(A) The cache hit ratio is more than 98%.

(B) If we want to improve the effective memory access time to 2.5 clock cycles, the hit ratio must be improved to more than 99.4%.

(C) If we insert a 2nd level cache with read/write time = 3 clock cycles, the hit ratio to the 2nd level cache for an effective memory access time of 2.5 clock cycles must be greater than 70%.

(D) If we insert a 2nd level cache with hit ratio 80%, for an effective memory access time of 1.5 clock cycles, the read/write time of the 2nd level cache must be no greater than 8 clock cycles.

(E) If we insert a 2nd level cache with hit ratio 90% and read/write time = 4 clock cycles, the effective memory access time is lower than 1.3 clock cycles.

接次頁

5. (10分，問答題) Following is the MIPS assembly language.

MIPS assembly language

Category	Instruction	Example	Meaning	Comments
Arithmetic	add	add $s1,$s2,$s3	$s1 = $s2 + $s3	Three register operands
	subtract	sub $s1,$s2,$s3	$s1 = $s2 – $s3	Three register operands
	add immediate	addi $s1,$s2,100	$s1 = $s2 + 100	Used to add constants
Data transfer	load word	lw $s1,100($s2)	$s1 = Memory[$s2 + 100]	Word from memory to register
	store word	sw $s1,100($s2)	Memory[$s2 + 100] = $s1	Word from register to memory
	load half	lh $s1,100($s2)	$s1 = Memory[$s2 + 100]	Halfword memory to register
	store half	sh $s1,100($s2)	Memory[$s2 + 100] = $s1	Halfword register to memory
	load byte	lb $s1,100($s2)	$s1 = Memory[$s2 + 100]	Byte from memory to register
	store byte	sb $s1,100($s2)	Memory[$s2 + 100] = $s1	Byte from register to memory
	load upper immed.	lui $s1,100	$s1 = 100 * 2^{16}	Loads constant in upper 16 bits
Logical	and	and $s1,$s2,$s3	$s1 = $s2 & $s3	Three reg. operands; bit-by-bit AND
	or	or $s1,$s2,$s3	$s1 = $s2 | $s3	Three reg. operands; bit-by-bit OR
	nor	nor $s1,$s2,$s3	$s1 = ~ ($s2 |$s3)	Three reg. operands; bit-by-bit NOR
	and immediate	andi $s1,$s2,100	$s1 = $s2 & 100	Bit-by-bit AND reg with constant
	or immediate	ori $s1,$s2,100	$s1 = $s2 | 100	Bit-by-bit OR reg with constant
	shift left logical	sll $s1,$s2,10	$s1 = $s2 << 10	Shift left by constant
	shift right logical	srl $s1,$s2,10	$s1 = $s2 >> 10	Shift right by constant
Conditional branch	branch on equal	beq $s1,$s2,25	if ($s1 == $s2) go to PC + 4 + 100	Equal test; PC-relative branch
	branch on not equal	bne $s1,$s2,25	if ($s1 != $s2) go to PC + 4 + 100	Not equal test; PC-relative
	set on less than	slt $s1,$s2,$s3	if ($s2 < $s3) $s1 = 1; else $s1 = 0	Compare less than; for beq, bne
	set less than immediate	slti $s1,$s2,100	if ($s2 < 100) $s1 = 1; else $s1 = 0	Compare less than constant
Uncondi-tional jump	jump	j 2500	go to 10000	Jump to target address
	jump register	jr $ra	go to $ra	For switch, procedure return
	jump and link	jal 2500	$ra = PC + 4; go to 10000	For procedure call

Please implement the following program with the MIPS assembly language. You should use *load* word and *store* word instructions to minimize the number of instructions of the generated machine program.

```
void arrayProc(int A[], int B[], int n) {
  for (int i = 0; i < n; i++) {
    A[i] = A[i] + B[i];
    B[i] = A[i] /2;
} }
```

6. (10分, 複選題) For memory management in OS, which of the following statements are correct ?
   (A) A page fault happens when the CPU detects a hardware read/write error to a page.
   (B) When a segment fault happens in the execution of a thread, OS must first bring the segment to the memory and then give CPU back to the thread.
   (C) When a page fault happens, the OS may use a spare page before the hardware error is removed.
   (D) When a page fault is caused by a stack push operation, one page swap-out can usually be saved.
   (E) When page thrashing occurs, the OS must find out the page that is illegally writing to all pages of other processes.

見背面

7. (10 分，複選題) Suppose that we have opened a file with the following statement in a program running on UNIX.

```
fd = open(FILEPATH, O_RDWR | O_CREAT | O_TRUNC,  0600);
```

Which of the following statements about memory-mapped files in UNIX are correct ?

(A) The following statement is incorrect since the address of the memory mapping is NULL.

```
mmap(NULL, 380, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0)
```

(B) The following statement maps a memory buffer starting at logical address 360 to file fd.

```
mmap(360, 380, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0)
```

(C) The following statement maps a memory buffer of length 380 bytes to file fd.

```
mmap(360, 380, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 10)
```

(D) After executing the following statement, two or more processes sharing the mapped buffer can thus read and write to the shared mapped buffer.

```
mmap(1024, 380, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0).
```

(E) Many operating systems support memory-mapped files. This feature allows a process to read and write the mapped pages in its address space using ordinary file operations instead of special load and store instructions.

8. (10 分，複選題) In a 32-bit ARM processor, which of the following statements are correct ?

(A) There are six different sizes of pages.
(B) 4-KB pages must be used together with 4-MB pages.
(C) 1-MB pages must be used together with 16-MB pages.
(D) Two-level paging is used with 4-KB pages.
(E) Two TLBs are used for the outer level page table.

9. (10 分，複選題) Which of the following statements are correct about disk arrays ?

(A) Rebuilding is the most efficient for RAID level 1 since data can be copied from another disk.
(B) RAID level 0 is used in high-performance applications where data loss is not critical.
(C) RAID 0+1 is used where both performance and reliability are important.
(D) RAID level 5 is often preferred for storing large volumes of data.
(E) Solaris ZFS file system uses checksums to provide fault-tolerance in case file pointers are wrong.

10. (10 分，問答題) Please write down the pseudo code that solves the reader-writer problem in process synchronization. Your solution should prevent the starvation of both the readers and the writers. You can only use OS service via semaphore procedures: wait() and signal().