科目：計算機系統(計算機組織)(500A)　　校系所組：中央大學電機工程學系（電子組）
交通大學電子研究所（乙組）
交通大學電控工程研究所（丙組）
清華大學電機工程學系（丙組）

一. [7%] Assume three 32-bit variables x, y, and z, are stored in memory with addresses A, B, and C, respectively.

(1) [2%] For the "load-store" instruction set architecture processor, write down the assembly codes to compute $x+y-z$ and then stores the result back to C.

(2) [5%] Suppose that all instruction operation codes are 6 bits, memory addresses are 24 bits (byte-addressable), and register addresses are 5 bits. What are the code sizes (in terms of bits) and the total memory accesses (in terms of bytes) for the codes in (1)? Note that the total memory accesses include both instructions and data moved to or from memory. And, the memory accesses must be aligned.

二. [8%] Consider the following binary data shown in the table.

case a.	$1010\ 1101\ 0001\ 0000\ 0000\ 0000\ 0000\ 0010_{two}$
case b.	$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_{two}$

(1) [4%] Write the MIPS code that creates the 32-bit constants listed above and stores that value to register `$t1`. Please use separate codes for the two cases.

(2) [4%] If the current value of the PC is `0x00000600`, can you use a single branch instruction to get to the PC address as shown in the table above? If the current value of the PC is `0x00400600`, can you use a single branch instruction to get to the PC address as shown in the table above? Please give your answers for the two cases in the table.

三. [10%] Given a finite word-length size, overflow occurs when a result is too large to be represented accurately; however, underflow occurs when a number is too small to be represented correctly. The right table shows pairs of decimal numbers.

	A	B
case a.	200	103
case b.	247	237

(1) [5%] Assume A and B are signed 8-bit decimal integers stored in two's-complement format. Calculate A+B and A–B using saturating arithmetic for the two cases. The result should be written in decimal. Show your work. Is there overflow, underflow, or neither?

(2) [5%] Assume A and B are unsigned 8-bit integers. Calculate A+B and A–B using saturating arithmetic for the two cases. The result should be written in decimal. Show your work. Is there overflow, underflow, or neither?

四. [8%] An instruction set is composed of L different instruction classes, denoted by $C(k)$, $k=1, 2, \ldots, L$. For every instruction in class $C(k)$, the execution time is given by $2+k$ ns. Consider a multiple-cycle implementation with a clock cycle 1 ns and assume class $C(k)$ instructions can then be executed in $2+k$ clock cycles, ignoring any overhead associated with control.

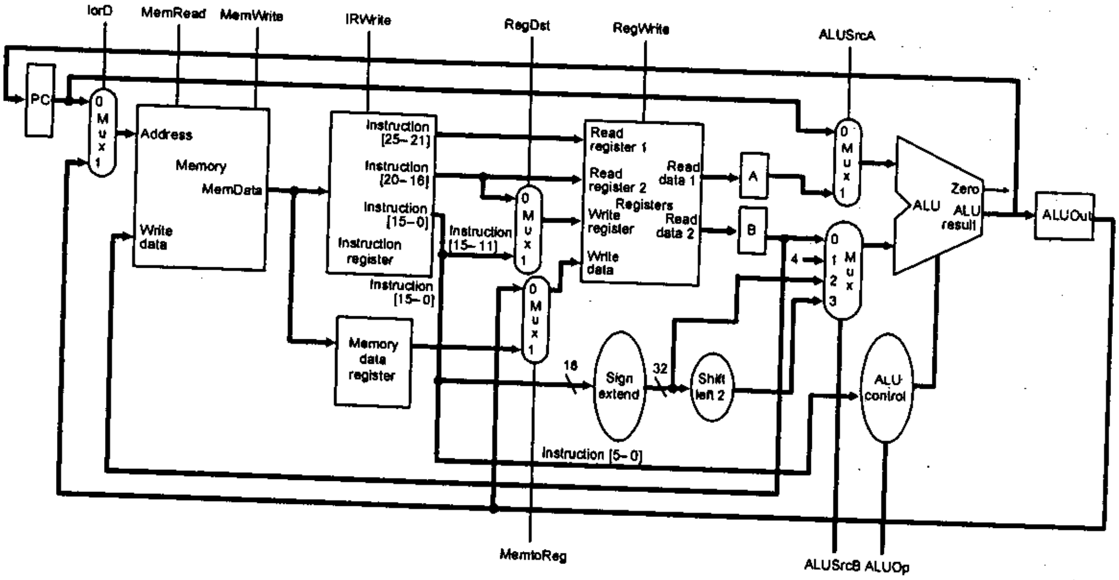
(1) [4%] Assume that all instruction classes are used with the same frequency. What is the performance advantage of multiple-cycle control relative to single-cycle control? Show the speedup factor.

(2) [4%] If the memory access latency is varied from 1 ns to 2 ns, this variation surely affects the multiple-cycle implementation. Should you increase the clock cycle or should you change the actions performed within each cycle? Justify your solution.

注意：背面有試題

五. [17%] Consider the following multiple-cycle implementation with control lines shown. The signals ALUOp and ALUSrcB are 2-bit control signals, while all the other control lines are 1-bit signals. Consider the following section of MISP code executing on the multiple-cycle processor. Instructions take from three to five execution steps. Note that `$16` denotes the register with index 16. The code starts with clock cycle 1. Assume that all register writes occur at the end of the clock cycle. Because we increment PC by 4 in clock cycle 1, PC is set to the value 1004 at the end of clock cycle 1.

```
    1000 addi $8,$16,4
    1004 bne $8,$16,L1
    1008 add $9,$17,$16
L1: 1012 sw $16,4($19)
    1016 lw $16,100($18)
    1020 sw $17,104($18)
```



(1) [4%] What are the values of RegDst, MemtoReg, ALUSrcA and ALUSrcB at the third step of `addi $8,$16,4`? For ALUSrcB, you can show its value with a decimal number.

(2) [4%] What are the values of PC at the end of clock cycles 4, 5, 6, and 7, respectively?

(3) [3%] Determine the average clock cycles per instruction (CPI) of the above MISP code.

(4) [3%] An engineer proposes to remove the memory data register (MDR) from the above multiple-cycle implementation. Her idea is to connect MemData (output port of the memory) directly to the multiplexor controlled by MemtoReg. With this modification, she claims that the `lw` instruction will take one less step. Does her modification effectively reduce the execution time? Justify your answer.

(5) [3%] Can the above multiple-cycle processor support the following register transfer language (RTL) in a step? Justify your answer.

```
if (A == B)  PC <= PC+(sign-extend(IR[15-0])<<2)
```

注:背面有試題

科目：計算機系統(計算機組織)(500A)　校系所組：中央大學電機工程學系（電子組）
交通大學電子研究所（乙組）
交通大學電控工程研究所（丙組）
清華大學電機工程學系（丙組）

六. [15%] Given a MIPS instruction sequence shown right. Assume this code is executed on a five-stage (IF, ID, EXE, MEM, WB) pipelined MIPS CPU with the capability to finish the register write in the first half cycle and the register reads in the second half cycle.

```
sll   $t1,$s1,2
add   $t1,$s2,$t1
lw    $s3,100($t1)
addi  $s3,$s3,1
add   $zero,$s4,$s3
slt   $t2,$s3,$zero
bne   $t2,$zero,L2
```

(1) [6%] If this MIPS CPU has no forwarding capability, please indicate all possible hazards and add NOP (no operation) instructions to eliminate them.

(2) [6%] Assuming this MIPS CPU has full forwarding capability to solve any hazards that can be solved, what is the speed-up achieved by adding full forwarding to the original CPU that had no forwarding?

(3) [3%] If we want to further reduce the hazards (if any) in (2), what can we do besides forwarding? Please make a brief discussion.

七. [10%] Given a MIPS instruction sequence shown right. Assume this code is executed on a pipelined MIPS CPU with a five-stage (IF, ID, EXE, MEM, WB) pipeline, full forwarding, and a predict-taken branch predictor. In addition, assume this CPU can finish the register write in the first half cycle and the register reads in the second half cycle.

```
L3: add  $s6,$s6,s5
    addi $s5,$s5,2
    slti $t0,$s5,10
    bne  $t0,$zero,L3
    sw   $s6,100($gp)
```

(1) [5%] Assuming the branch is taken, please draw the simple (traditional) multiple-clock-cycle pipeline execution diagram of the first 5 executed instructions, including all necessary stalls.

(2) [5%] If the branch is not taken, please redraw the simple (traditional) multiple-clock-cycle pipeline execution diagram of the first 5 executed instructions, including all necessary stalls.

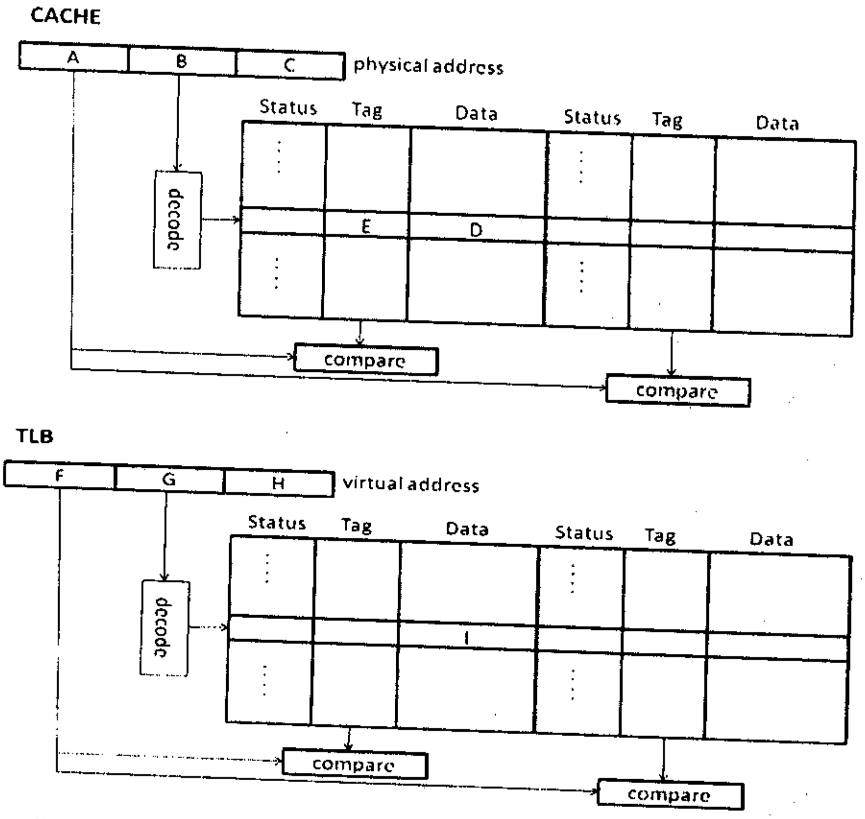
八. [7%] A computer system has 32 bits virtual addresses and 32 bits physical addresses. Assume the page size is 8KB and the cache block size is 8 Bytes. Only a first level physical cache (physical index and physical tag) is implemented in this system. One simple way to overlap TLB lookup with cache access is to use some of the page offset bits to index the cache since those bits are the same in both virtual and physical memory addresses. What is the maximum two-way set associative cache in terms of bytes you can design for this system? Briefly explain your answer.

九. [18%] Consider a memory system with the following parameters:

- A 2-way set associative TLB has 512 entries in total.
- 64Kbyte L1 data cache has 128 byte lines and is also 2-way set associative. Assume it is a write-back cache.
- Virtual addresses are 64 bits and physical addresses are 32 bits.
- 8KB page size.

注意:背面有試題

科目：計算機系統(計算機組織)(500A)　校系所組：中央大學電機工程學系（電子組）
交通大學電子研究所（乙組）
交通大學電控工程研究所（丙組）
清華大學電機工程學系（丙組）



Above are diagrams of the cache and the TLB. Copy the following table to your answer book and fill in with the appropriate information.

L1 Cache		TLB	
A=	bits	F=	bits
B=	bits	G=	bits
C=	bits	H=	bits
D=	bits	I=	bits
E=	bits		