類組：電機類　科目：計算機系統(計算機組織)(300A)　　共 4 頁 第 1 頁

※請在答案卷內作答

一、[14%] Consider an implementation of an instruction set architecture. The instructions can be divided into three classes according to their CPI (class A, B, and C). The CPIs are 1, 2, and X for the three classes, respectively, and the clock rate is 4.4 GHz.

Give a program with a dynamic instruction count of 5×10^9 instructions divided into classes as flows: 20% class A, 60% class B, and 20 % class C. The program is compiled using compiler A.

(1) [3%] Find X (CPI for the class C) given that the global CPI is 2.2.
(2) [3%] Find the clock cycles required.
(3) [3%] What is the performance expressed in instructions per second (the global CPI is 2.2).
(4) [5%] A new compiler, compiler B, is developed that uses only 10^9 instructions and has an average CPI of 1.1. Assume the programs compiled using Compiler A and B run on Processor A and Processor B, respectively. If the execution time on the Processor B is a half of the execution time on the Processor A, how much faster is the clock of Processor A versus the clock of Processor B? (Find Clock A/Clock B)

二、[6%] Assume there is a 16-bit half precision format. The leftmost bit is still the sign bit, the exponent is 5 bits wide and has a bias of 15, and the mantissa is 10 bits long. A hidden 1 is assumed. Please write down the bit pattern to represent -2.1875×10^{-1} a ssuming the 16-bit half precision format. (Note: 0.21875 = 7/32)

三、[15%] Given the following C program segment, the converted MIPS assembly codes are shown in the right. Assume that arguments `n` and `s` locate in `$a0` and `$a1`.

```
int sum(int n, int s) {
    if (n < 1) return 0;
    else return (n + sum(n-s,s));
}
```

```
Sum: addi $sp,$sp,-12
          (a)
     sw   $a1,4($sp)
     sw   $a0,0($sp)
     slti $t0,$a0,1
     beq  $t0,$zero,L1
     add  $v0,$zero,$zero
          (b)
     jr   $ra
L1:  sub  $a0,$a0,$a1
     jal  Sum
     lw   $a0,0($sp)
     lw   $a1,4($sp)
          (c)
     addi $sp,$sp,12
     add  $v0,$a0,$v0
     jr   $ra
```

(1) [9%] Please fill in the blanks (a), (b), and (c) to complete this assembly codes.
(2) [6%] Let the initial values of `$a0`, `$a1`, `$t0`, `$sp` are $6_{(hex)}$, $2_{(hex)}$, $8_{(hex)}$, $6FFF808C_{(hex)}$ respectively. What are the final values of `$v0` and `$a0` after the completion of the program? What is the value of `$sp` when the first "`jr`" instruction is encountered?

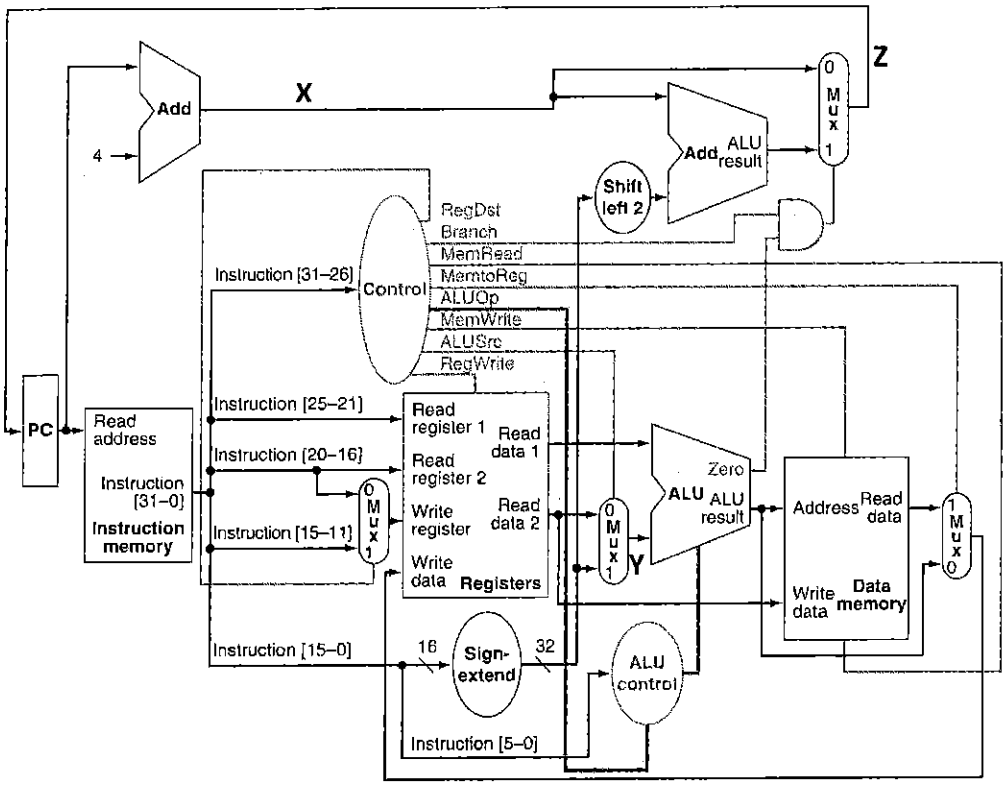
注意：背面有試題

參考用

類組：電機類　科目：計算機系統(計算機組織)(300A)　共 4 頁 第 2 頁

※請在答案卷內作答

四、[15%] Consider the following architecture. The latency of each block is given in the following. Assume that the control block has zero delay if not specified.



I-Mem	Add	Mux	ALU	Regs	D-Mem	AND	Sign-Extend	Sift-left-2
220ps	50ps	20ps	120ps	80ps	250ps	10ps	20ps	10ps

```
1020        sub $s4, $s3, $t3
1024        beq $t2, $s4, Else
1028        addi $t2,$t2,-4
1032        add $t0, $t1, $t2
1036        sw $s2,0($t0)
1040        j  Exit
1044  Else: addi $t2, $t2, 4
1048  Exit:…
```

考用

注意：背面有試題

(1) [5%] Which instruction is not supported by this architecture? How to revise the architecture to support that instruction. Please explain by drawing the related blocks and wires?

(2) [6%] Except the instruction that is not supported in (1), which instruction needs the longest clock cycle period? What would the cycle time be?

(3) [4%] Before this code segment, the register contents in decimal are given as follows. After execution, for some clock cycle, if x=1028, what are the values of signal Y (after the MUX controlled by ALUSrc) and signal Z (input of PC) ?

$t0	$t1	$t2	$t3	$s2	$s3	$s4
48	8	12	288	75	300	102

五、[17%] Consider the following code sequence running on 5-stage pipeline MIPS:

```
sw r16,12(r6)
lw r16,8(r6)
beq r5,r4,Label # Assume r5!=r4
add r5,r1,r4
slt r5,r15,r4
```

Assume that the individual pipeline stage of **IF**, **ID**, **Exe**, **Mem**, and **WB** has the latency of 200ps, 120ps, 200ps, 190ps, and 100ps, respectively.

(1) [3%] Assume that all branches are perfectly predicted and that no delay slots are used. If we only have one memory (for both instructions and data), there is a structural hazard every time we need to fetch an instruction in the same cycle in which another instruction accesses data. To guarantee forward progress, this hazard must always be resolved in favor of the instruction that accesses data. What is the total execution time of this instruction sequence in the 5-stage pipeline that only has one memory? We have seen that data hazards can be eliminated by adding `nops` to the code. Can you do the same with this structural hazard? Why?

(2) [3%] Assuming stall-on-branch and no delay slots, what speedup is achieved on this code if branch outcomes are determined in the **ID** stage, relative to the execution where branch outcomes are determined in the **Exe** stage?

(3) [5%] Assume that all branches are perfectly predicted and that no delay slots are used. If we change load/store instructions to use a register (without an offset) as the address, these instructions no longer need to use the ALU. As a result, **Mem** and **Exe** stages can be overlapped and the pipeline has only 4 stages. Change this code to accommodate this changed ISA. Assuming this change does not affect clock cycle time, what speedup is achieved in this instruction sequence?

(4) [3%] Given these pipeline stage latencies, repeat the speedup calculation from (3), but take into account the (possible) change in clock cycle time. When **Exe** and **Mem** are done in a single stage, most of their work can be done in parallel. As a result, the resulting **Exe/Mem** stage has a latency that is the larger of the original two, plus 20 ps needed for the work that could not be done in parallel.

(5) [3%] Given these pipeline stage latencies, repeat the speedup calculation from (3), taking into account the (possible) change in clock cycle time. Assume that the latency of the **ID** stage increases by 50% and the latency of the **Exe** stage decreases by 10ps when branch outcome resolution is moved from **Exe** to **ID**.

注意：背面有試題

參考用

類組：電機類　科目：計算機系統(計算機組織)(300A)　共 4 頁 第 4 頁

※請在答案卷內作答

六、[8%] Suppose you want to perform two sums: one is a sum of 10 scalar variables, and one is a matrix sum of a pair of two-dimensional arrays, which have dimensions 20 by 20. Suppose that only the matrix sum is parallelizable.

(1) [4%] Assume the load was perfectly balanced, what speed-up do you get with 40 multiple processors?

(2) [4%] If one processor's load is 2 times higher than all the rest, what speed-up do you get with 40 multiple processors?

七、[5%] Some disks are quoted to have a 1,000,000-hour mean time to failure (MTTF). For a data center, there might have 50,000 servers. Suppose each server has 4 disks. Use annual failure rate (ASF) to calculate how many disks we would expect to fail per year.

八、[20%]A new processor is advertised as having a total on-chip cache size of 168 Kbytes (KB) with a byte-addressed two-level cache. It integrates 8 KB of onboard L1 cache, split evenly between instruction and data caches, at 4 KB each. Meanwhile, the processor integrates 160 KB of unified L2 cache with the chip packaging. The two-level cache has the following characteristics:

	Address	Associative	Sector size	Block size	Caching method	Hit time	Average local miss rate
L1 cache	Physical byte-address	Direct mapped	2 blocks/sector	1 word/ block (8 bytes/word)	Write through, no write allocate	1 clock	0.15
L2 cache	Virtual byte-address	5-way set associative	2 blocks/sector	1 word/ block (8 bytes/word)	Write through, write allocate	5 clocks (after L1 miss)	0.05

The system has a 40-bit physical address space and a 52-bit virtual address space. L2 miss (transport time) takes 50 clock cycles.

(1) [3%] What is the total number of bits within each L1 cache block, including status bits?

(2) [3%] What is the total number of bits within each L1 cache sector, including status bits?

(3) [3%] What is the total number of bits within each L2 set, including status bits?

(4) [3%] Compute the average memory access time (AMAT, in clocks) for the given 2-level cache.

(5) [8%] Consider a case in which a task is interrupted, causing L1 and L2 caches described above to be completely flushed by other tasks, and then that original task begins execution again and runs to completion, completely refilling the cache as it runs. What is the approximate time penalty, in clocks, associated with refilling the caches when the original program resumes execution?

Note: "approximate" is that you should compute L1 and L2 penalties independently and then add them, rather than try to figure out coupling between them.

注意:背面有試題