1. **(9%)** Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (class A, B, C, and D). P1 with a clock rate of 2.5GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2.
   (a) (3%) Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D. What is the ratio of CPI for P1/P2?
   (b) (3%) The result of the benchmark running on the machine has an instruction count of 2.5E12, and execution time of 900 s, and a reference time of 9000 s. Find the CPI if the clock cycle time is 0.25 ns.
   (c) (3%) Find the increase in CPU time if the number of instructions of the benchmark is increased by 20% and the CPI is increased by 5%.

2. **(12%)** For the following calculation:
   (a) (4%) Assume 152 and 213 are signed 8-bit decimal integers store in two's complement format. Calculate 151+214. The result should be written in decimal.
   (b) (4%) Assume 152 and 213 are unsigned 8-bit integers. Calculate 151+214 using saturating arithmetic. The result should be written in decimal.
   (c) (4%) Given by a floating-point number $427D0000_{(hex)}$ that is represented by IEEE 754 standard. What decimal number does it?

3. **(12%)** For a direct-mapped cache design with 32-bit address [31—0], the following bits of the address are used to access the cache:

Tag field	Index field	Offset field
[31—14]	[13—6]	[5—0]

   (a) (3%) What is the block size (in words)?
   (b) (3%) How many entries does the cache have?
   (c) (3%) If 1 bit for the valid field is used, what is the total number of Kibits in a direct-mapped cache?
   (d) (3%) Assume the miss rare of an instruction cache is 3% and the miss rate of the data cache is 2%. The frequency of all loads and stores instruction is 36%. If a processor has a CPI of 2 without any memory stalls and the miss penalty is 100cycles for all misses, how much faster a processor would run with a perfect cache that never missed.

4. **(10%)** Caches are important to providing a high-performance memory hierarchy to

processors. Below is a list of 32-bit memory address-references, given as word addresses.

3, 180, 43, 2, 191, 88, 190, 14, 181, 44

(a) (3%) Assuming the cache is initially empty. For each of these references, given a direct-mapped cache with two-word blocks and a total size of 8 blocks. What is the miss rate?

(b) (3%) If the miss stall time is 25 cycles and 3 cycles for access time, what is the total cycles for this cache?

(c) (4%) For a three-way set associative cache with two-word blocks and a total size of 24 words. Use LRU replacement. What is the hit ratio?

5. **(7%)** Listed below are key page table parameters.

Virtual address size	Page size	Page table entry size
32 bits	8 KiB	4 bytes

(a) (3%) For a single-level page table, how much physical memory is needed for storing the page table?

(b) (4%] Calculate the total page table size for a system running 6 applications that utilize half of the memory available (it means that half of 32-virtual address for each running application).

6. **(10%)** Translate the following C codes into MIPS instructions. Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `$s0`, `$s1`, `$s2`, `$s3`, and `$s4`, respectively. Assume further that the base address of the arrays `A` and `B` are in registers `$s6` and `$s7`, respectively.

```
int leaf_example(int g, int h, int, i, int j)
{
    int f;
    f = ( A[B[g]+1] + h ) - ( i + j );
    return f;
}
```

Be sure to handle the stack pointer appropriately. Indicate the names of registers and variables stored on the stack and mark the location of `$sp`.

7. **(10%)** Consider the following MIPS loop:

```
LOOP:  slt $t2, $0, $t1
       beq $t2, $0, DONE
       subi $t1, $t1, 1
```

```
        addi $s2, $s2, 2
        j LOOP
DONE:
```

(a) (3%) Assume that the register `$t1` is initialized to the value 10. What is the value in register `$s2` assuming `$s2` is initially zero?

(b) (3%) Assume that the register `$t1` is initialized to the value *N*. How many MIPS instructions are executed?

(c) (4%) Assume that the registers `$s1`, `$s2`, `$t1`, and `$t2` are integers *A*, *B*, *i*, and *temp*, respectively, write the equivalent C code routine.

8. (25%) Consider the following fragment of MIPS code:

```
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, 150ps, 190ps, and 100ps, respectively.

(a) (4%) 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?

(b) (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?

(c) (4%) 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?

(d) (4%) Given these pipeline stage latencies, repeat the speedup calculation

from (b), 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.

(e) (4%) Given these pipeline stage latencies, repeat the speedup calculation from (c), taking into account the (possible) change in clock cycle time. Assume that the latency 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.

(f) (4%) Assuming stall-on-branch and no delay slots, what is the new clock cycle time and execution time of this instruction sequence if beq address computation is moved to the Mem stage? What is the speedup from this change?

9. (5%) Answer the following questions related a multiple processors system.

(a) (2%) Suppose you want to achieve a speed-up of 80 times faster with 100 processors. What fraction of the original computation can be sequential?

(b) (3%) One major problem in the multiple processors system is the remote access, because communications of data between separate processors are overheads. Suppose we have a program running on a 32-processor system, which has an average of 200 ns to handle a remote data access. And, the clock period of each processor is 0.3 ns. For the simplicity, in this program we assume that all the data accesses, except remote ones, hit in their local cache and the processor will be stalled on a remote request. If the base CPI is 0.5, how much faster is the multiprocessor if there is no remote access versus if 0.2% of the instructions involve a remote access?