1. After graduating, you are asked to become the lead computer designer at Speedo Computers, Inc. Your study of usage of high-level language constructs suggests that procedure calls are one of the most expensive operations. You have invented a scheme that reduces the loads and stores normally associated with procedure calls and returns. The first thing you do is run some experiments with and without this optimization. Your experiments use the same state-of-the-art optimizing compiler that will be used with either version of the computer. These experiments reveal the following information:

- The clock rate of the unoptimized version is 5% higher.
- 30% of the instructions in the unoptimized version are loads or stores.
- The optimized version executes 2/3 as many loads and stores as the unoptimized version. For all other instructions the dynamic counts are unchanged.
- All instructions (including load and store) take one clock cycle.

Which is faster? Justify your decision quantitatively.

2. You are building a computer with a hierarchical memory system that consists of separate instruction and data caches followed by main memory. You are using a RISC-V multicycle processor running at 5 GHz.

(i) The instruction cache is perfect (i.e., always hits) but the data cache has a 15% miss rate. On a cache miss, the processor stalls for 200ns to access main memory, then resumes normal operation. Taking cache misses into account, what is the average memory access time?

(ii) How many clock cycles per instruction (CPI) on average are required for loa5d and store word instructions considering the non-ideal memory system?

(iii) Consider a benchmark application that has 25% loads, 10% stores, 11% branches, 2% jumps, and 52% R-type instructions. Taking the nonideal memory system into account, what is the average CPI for this benchmark?

(iv) Suppose that the instruction cache is also nonideal and has a 10% miss rate. What is the average CPI for the benchmark in part (c)? Take into account both instruction and data cache misses

3. Assume we have a computer where the CPI is 1.0 when all memory accesses (including data and instruction accesses) hit in the cache. The cache is a unified (data + instruction) cache of size 256 KB, 4-way set associative, with a block size of 64 bytes. The data accesses (loads and stores) constitute 50% of the instructions. The unified cache has a miss penalty of 25 clock cycles and a miss rate of 2%. Assume 32-bit instruction and data addresses.

3.1 What are the Number of bits used for block offset?

3.2 What are the Number of sets in the cache?

3.3 What are the Number of bits for the cache index?

3.4 What are the Number of bits for the tag?

3.5 Calculate the number of Stall Cycles per instruction

3.6 How much faster would the computer be if all memory accesses were cache hits?

4. Assume that you have a computer with 1 clock cycle per instruction (CPI=1) when all accesses to memory are in cache. The only accesses to data come from load and store instructions. Those accesses account for 20 % of the total number of instructions. Miss penalty is 75 clock cycles and miss rate is 6 %. Determine the speedup obtained when there is no cache miss compared to the case when there are cache misses

5. Assume that we make an enhancement to a computer that improves some mode of execution by a factor of 13. Enhanced mode is used 30% of the time, measured as a percentage of the execution time when the enhanced mode is in use. Recall that Amdahl's law depends on the fraction of the original, unenhanced execution time that could make use of enhanced mode. Thus, we cannot directly use this 30% measurement to compute speedup with Amdahl's law
   (i) What is the speedup we have obtained from fast mode?
   (ii) What percentage of the original execution time has been converted to fast mode?
6. (i) A cache may be organized such that:
   - In one case, there are more data elements per block and fewer blocks
   - In another case, there are fewer elements per block but more blocks However, in both cases – i.e., larger blocks but fewer of them OR shorter blocks, but more of them – the cache's total capacity (amount of data storage) remains the same

   What are the pros and cons of each organization? Support your answer with a short example assuming that the cache is direct mapped

   (ii) Assume:

   - A processor has a direct mapped cache
   - Data words are 8 bits long (i.e., 1 byte)
   - Data addresses are to the word
   - A physical address is 20 bits long
   - The tag is 11 bits
   - Each block holds 16 bytes of data

   How many blocks are in this cache?

   (iii) Consider a 16-way set-associative cache:

   - Data words are 64 bits long

   - Words are addressed to the half-word

   - The cache holds 2 Mbytes of data

   - Each block holds 16 data words

   - Physical addresses are 64 bits long

   How many bits of tag, index, and offset are needed to support references to this cache?