Name:
Turn in this exam with your answers, or your score will be zero (0).

CMSC 411 Computer Architecture
Midterm Exam
Summer, 1998

Remember, it is supposed to be HARD. Don't panic when you get stuck. Try something else and come back to it. If logic fails, try to think like a "dumb user". After that, thinking like a computer scientist will be a relief.

Problem 1: Bewitched, Bothered and Bewildered (24)

Write T or F next to the appropriate number on your answer sheet. If you are unsure, write U and *briefly* explain why you are unsure.

1. The instruction count per program and the total execution time of the program are independent.
2. Amdahl's law demonstrates that overall speedup is limited primarily by the fraction of the original process that is enhanceable.
3. The performance of a computer architecture is usually be improved by decreasing the clock rate.
4. Adding a new type of ALU instruction to a computer's Instruction Set Architecture (ISA) has no effect upon the total CPI.
5. All instructions in a General Purpose Register machine ISA of type (0,3) must have 3 operands.
6. Increasing the depth of a pipeline reduces the potential for RAW hazards.
7. MFLOPS is a reasonable measure of performance for a machine dedicated to numerically intensive calculations, such as data encryption and decryption.
8. An enhancement to machine A is desirable when the ratio of the average execution time on A' (the enhanced machine) to the average execution time on A exceeds one (1).
9. Compiler optimization can remove all control hazards.
10. The 90-10 law states that 90% of the hazards that must be handled during program execution arise in 10% of the code.
11. The arithmetic mean measures task execution time, while the harmonic mean measures task performance.
12. Quantitative design methods advocate the use of numerical estimates of key system parameters in making critical design decisions.

Problem 2: Life in the Fast Lane (24)

Suppose that you install new HyperOptomIZing Hardware, abbreviated below by H, which speeds up all floating point operations by a factor of K. No other operations are affected by this upgrade.
Let T_O denote the original execution time, and T_H denote the enhanced execution time. Assume that x is the amount of time spent performing non-floating point operations before H is used, and y is the time spent on floating-point operations before H is used.

2.1 Write an expression for the new execution time, T_H, in terms of K, x, and y.

2.2 Express Amdahl's law in terms of the variables x, y, K, T_0, and T_H. (You may not need all of these variables.)

2.3 If half of the original execution time is spent on floating point instructions, what must K be to achieve an effective (or overall) speedup of 5.

2.4 If the new floating point code runs 10 times faster than the old, the original code ran for 50 seconds, and the effective (overall) speedup is 5, how much time does the new code spend executing floating point instructions?

Problem 3: Quantitatively DeLuXe Torture (30)

3.1 (10) Suppose you change the ISA of a machine such that the CPI is increased by 20% and the IC decreases by 5%. What modification of the old clock rate would be needed to decrease the original CPU execution time by 15%.

3.2 (10) If ALU instructions take 1 clock cycle (cc), conditional branches take 3cc, and all other instructions take 2 cc's, compute the CPI if 55% of the instructions are ALU and 10% are conditional branches.

3.3 (10) Consider the equation of a line given by $A * X + B * Y + C = 0$. If $A = 0$, then the line is horizontal, and has no x-intercept. Similarly, if $B = 0$, the line is vertical and has no y-intercept. Otherwise, x-intercept and y-intercepts of this equation, are $X : ((-C/A),\ 0)$ and $Y : (0,\ (-C/B))$.

Write a DLX code fragment that will correctly compute the values of the x-intercept and y-intercept when they exist. A validity variable, V should be set to 1 if no x-intercept exists, -1 if no y-intercept exists, and zero otherwise. *Oh, and no comments, no points.*

Problem 4: A New Home for the DLX ISA (31)

Consider the following 8-stage variant of the basic 5-stage DLX pipeline. Separate instruction and data memories are used. However, no forwarding occurs and the writing/reading of registers is **NOT** split across a single clock cycle.

F1	Start Instruction Fetch
F2	Complete Instruction Fetch
ID	Decode instruction and fetch register values
EX1	Start Execution; branch condition tested, branch target computed
EX2	Execution completes; effective address or ALU result ready
MEM1/ALUWB	First half of memory cycle plus WB of ALU instructions
MEM2	Memory access completes
LWB	Load write back completes

4.1 (5) How many register read and write ports are needed to avoid structural hazards due to register contention?

4.2 (5) Give DLX instruction sequences that illustrate the possible data hazards. If no hazards of a given type exist, explain why not.

4.3 (5) Suppose the branch strategy assumes branch not taken. How far can a predicted (but not yet verified correct) instruction execute without making flushing impossible if the predicted direction is incorrect.

4.4 (16) Data hazards occur when the user or consumer of a value needs it before the value is available from the producer. Identify all possible combinations of producer/consumer stages, and show how many stalls are needed to ensure correct operation. For each producer/consumer pair, give an instruction sequence illustrating the stalls, and show the pipeline diagram executing the instruction sequence.