Assume we have the following values for a program with 20,000 instructions:

   Number of FP instructions: 2000  
   Average CPI for FP instructions: 8  
   Average CPI for other instructions: 2.5

Assume we have two design alternatives: Decrease the CPI of FP instructions to 6 or decrease the CPI for other instructions to 2.45. The clock frequency remains constant for either design. Compare these two design alternatives and show which one would be a better design decision. Calculate the speedup of the better solution with the original implementation using Amdahl's law.
Problem: CA2  
Area: Computers and Architecture  
Code #__________

Assume we have a computer where the CPI is 1.0 when all memory accesses hit in the cache. The only data accesses are load and store instructions which account for 50% of all instructions. If the miss penalty is 20 clock cycles and the miss rate is 3%, how much faster would the computer be if all instructions were cache hits?
Problem: CA3  Area: Computers and Architecture  Code #

Suppose the following MIPS instruction sequence gets executed on the basic five-stage pipeline MIPS processor.

lw $5, 0($5)
add $5, $5, $5
sw $5, 0($5)

a. Assume there is no forwarding in this pipelined processor. Indicate hazards and add nop instructions (if needed) to eliminate them. Clearly justify your answer.

b. Assume there is full forwarding (EXE forwarding and MEM forwarding). Indicate hazards and add nop instructions (if needed) to eliminate them. Clearly justify your answer.
Using the series of references given in the previous problem (a series of address references given as word addresses in 7-bit binary: 0000010, 0000011, 0001011, 0010000, 0010101, 0001101, 1000000, 0110000, 0010011, 0001011 (2, 3, 11, 16, 21, 13, 64, 48, 19, 11 in decimal)), show the hits and misses and final cache contents for a two-way set associative cache with 8 one-word blocks that is initially empty. Assume “first-in, first-out” replacement strategy. Show your work!
What is SIMD (Single Instruction Multiple Data) computer architecture? Extensively discuss advantages and disadvantages. Then, give three applications of SIMD architecture.
Assuming the true and the complemented inputs are available, implement the logic function Y using pass transistor logic with minimum number of transistors. $Y = ABD + A'B'D' + ACD + AC'D'$
Draw the transistor diagram of an AOI421 gate. Size the transistors so that it doubles the drive strength, in the worst case, of an inverter with 5/2 PMOS and 3/2 NMOS. Write down the size next to each transistor.
Design a binary mod-6 up/down counter. If the input $C = 0$, count up, and if $C = 1$, count down. Use "don't cares" for unused states and DFF's for the state memory. Calculate next-state equations for all flip-flops.
Design a Mealy state machine that will detect a 1010 sequence with overlap (e.g. the following stream of bits will cause the sequence to be detected twice: "101010"). When the sequence is detected, the output \( Y \) will be High then reset itself to Low. Use SR flip-flops for state memory. Calculate next-state equations for each input (S and R) of each flip-flop and an output equation for \( Y \).
Given the function $F(U,V,W,X,Y)$ defined by the following minterm expression. Answer the following questions.

$$F = \sum_{UVWXYZ} m(0,3,4,5,6,17,19,20,22,23,24,27,31) + \sum_{UVWXYZ} \text{dc}(1,26,30)$$

a) Determine the minimal SOP expression for $F$.

b) Determine the minimal POS expression for $F$. 

Page 1 of 2.
c) Implement F using an 8:1 MUX and other logic gates as needed.
Compare and contrast a popular Evolutionary Computation Method of your choice with Support Vector Machines. How would you expect them to perform in terms of run time and memory footprint and why?
Problem: CA21  Area: Networking  Code #

Answer both questions below.

1. Discuss the sliding window mechanism employed in TCP protocols. Give examples of the flow/congestion control algorithms used to select the window size. Briefly explain how they work for at least two algorithms.

Discuss the interaction between the TCP protocol and the random early detection (RED) queuing scheme. Is the RED scheme better than a simple drop-tail scheme? Explain your answer.
Assume you wish to transfer an $n$ B (byte) file along a path composed of the source, destination, seven point-to-point links, and five switches. Suppose each link has a propagation delay of 2 ms and a bandwidth of 4 Mbps, and that the switches support both circuit and packet switching. Thus, you can either break up the file into 1 KB packets or set up a circuit through the switches and send the file as one contiguous bit stream. Suppose that each packet has 24 B of header information and 1000 B of payload, store-and-forward packet processing at each switch incurs a 1 ms delay after the packet has been completely received, packets may be sent continuously without waiting for acknowledgements, and circuit setup requires a 1 KB message to make one round trip message on the path, incurring a 1 ms delay at each switch after the message has been completely received. Assume that the switches introduce no delay to data traversing a circuit. You may also assume that file size is a multiple of 1000 B. Answer all four questions below.

1. For what file size $n$ B is the total number of bytes sent across the network less for circuits than for packets? In other words, which file size is makes circuit switching more efficient than packet switching when the number of bytes transferred is the metric by which the two techniques are compared? Justify your answer.

2. For what file size $n$ B is the total latency incurred before the file arrives at the destination less for circuits than for packets? In other words, which file size is makes circuit switching more efficient than packet switching when the end-to-end latency is the metric by which the two techniques are compared? Justify your answer.

3. How sensitive are your answers to parts 1 and 2 to the number of switches along the path? To the data rate of the links? To the ratio of packet size to packet header size?

4. How accurate do you consider this model of the relative merits of circuit and packet switching? Does it ignore important considerations that discredit one or the other approach? If so, what are they?
With one-bit parity, we can detect all one-bit errors. Show that at least one generalization fails, as follows (answer both parts):

a. Show that if messages $m$ are eight bits long, then there is no error detection code $e = e(m)$ of size two bits that can detect all two-bit errors.

b. Find an $N$ (not necessarily minimal) such that no 32-bit error detection code applied to $N$-bit blocks can detect all errors altering up to eight bits.
A 1-km-long, 10-Mbps CSMA/CD LAN (not 802.3) has a propagation speed of 200 m/μsec. Repeaters are not allowed in this system. Data frames are 256 bits long, including 32 bits of header, checksum, and other overhead. The first bit slot after a successful transmission is reserved for the receiver to capture the channel in order to send a 32-bit acknowledgement frame. What is the effective data rate, excluding overhead, assuming that there are no collisions? Show every step of your work.