CS 14 Spring 2001 — Mid-term #1

Name: _____

1. (15 points) Use a Karnaugh map to simplify the boolean function described by the truth table below. Draw your rectangles clearly and express your result as a boolean algebraic equation – **do *not* draw a circuit.**

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

2. (15 points) Consider the the logic function described by the truth table below. Write the boolean algebraic equation **in the sum-of-products form**, and then **draw a two-level circuit** that implements the function. (**Do *not* bother with simplifying the function.**)

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

3. (20 points) Design and draw a clocked circuit that repeatedly emits the sequence `00, 01, 11, 10`, where the output is a pair of lines that should be labeled Y_1 and Y_0. ***Hint:*** *This circuit is a two-bit counter, but the sequence is different from the normal counting sequence.* You may assume that the clock and clear lines are provided – you need not express how they are controlled.

4. Consider the addressable memory, both registers and main memory, available when programming in MIPS assembly for the follow questions:

   (a) (5 points) Explain the difference between the following two instructions:

   ```
   lw   $t0, 0($s2)
   move $t0, $s2
   ```

   (b) (5 points) For the following instruction, three different memory addresses are specified. Identify those three addresses, and explain how they are used by the instruction.

   ```
   sw   $t5, 0($t6)
   ```

   (c) (5 points) If we wanted to build a MIPS chip with 64 registers instead of 32, what would we have to change about the instruction set?

5. (15 points) When a procedure is called, it is responsible for preserving the values of a number of registers: `$s0 to $s7`, `$a0 to $a3`, `$sp`, and `$ra`, to name a few.. How does it perform that preservation? How and why is the preservation of `$sp` different from the others?

6. Consider an instruction set that looks very much like the MIPS instruction set, *except* that the *jump* instructions have been removed. Instead, the instruction set allows for direct manipulation of the program counter by using the symbol `$pc` in assembly programming. (Note that this new instruction set still allows, as MIPS does, direct manipulation of `$ra`). How would you replace the following MIPS *jump* instructions with direct manipulation of `$pc` and `$ra` so that your program will work with this new instruction set?
(Note: You may assume that the assembler allows the use of labels as part of any instruction where an immediate value is allowed.)

   (a) (5 points) `j L1`

   (b) (5 points) `jr $s0`

   (c) (10 points) `jal MyProc`