Final exam
ECE 203
7 December 2004
Prepared by Robert Dick

You may not use books, notes, or calculators when completing this exam. Please show your work. Please look over all the problems now and ask questions if any of them are not clear. Manage your time. It's better to have good answers for all questions than a wonderful answer to only one question. Read the questions carefully.

Good luck!

1. **(50 pts.)** Implement an FSM that controls a line-following car.

   Hint: Don't panic. The problem is broken down into steps that you have already done many times.

   You have access to two inputs, *L* and *R*. These are produced by a pair of downward-facing light sensors near the front-left and front-right of the car.

LR	Meaning
00	Both sensors are on the line
01	Left sensor on the line, right sensor off the line
10	Left sensor off the line, right sensor on the line
11	Neither sensor on the line

   Your machine must produce two outputs, *A* and *D*, that will control the car's steering.

AD	Meaning
00	Turn right
01	Turn left
10	Go straight
11	Go straight

   Your machine should behave in a reasonable way even if both sensors leave the line. You may assume that only one sensor may leave the line at a time. You may also assume that the machine is clocked at a very high frequency, relative to the rate of changes to input resulting from the car's motion. Do not assume access to inverted inputs.

   Note: My (reasonably neat) solution to the entire problem will fit on two pages in the test book.

   (a) **(1 pts.)** Will you use a Mealy or Moore FSM?

   Hint: A Moore machine may require more states, making the later portions of the problem more difficult to solve.

   (b) **(7 pts.)** Draw the state diagram for the FSM.

   (c) **(6 pts.)** Show the state table.

   (d) **(6 pts.)** Do state assignment and update the state table.

   (e) **(6 pts.)** Use K-Maps or QM to Derive minimal next-state functions.

   (f) **(6 pts.)** Use K-Maps or QM to Derive minimal output functions.

   (g) **(6 pts.)** Prepare your functions for efficient implementation in CMOS.

   (h) **(6 pts.)** Show a circuit diagram of the FSM.

   (i) **(6 pts.)** For any gate (other than the D flip-flops) you used, show how to construct that type of gate from NMOS and PMOS transistors.

2. **(10 pts.)** Find a minimal SOP expression for the following function using the Quine-McCluskey method. Hint: Read carefully.

$$f(a,b,c) = \prod(1,5,7) + d(0,6)$$

3. **(35 pts.)** Consider the following PIC assembly language subroutine. The parameters `ain` and `bin` are passed in by the caller. Something is returned in `W`. You may assume that the left-most four bits of `ain` and `bin` are all 0s.

```
ain equ 0x20
bin equ 0x21
count equ 0x22

unknown_subroutine
  movlw d'8'
  movwf count
  clrw

unknown_subroutine_loop
  btfsc bin, 0
  addwf ain, W

  bcf STATUS, C
  rrf bin, F

  bcf STATUS, C
  rlf ain, F

  decfsz count, F
  goto unknown_subroutine_loop
  return
```

   (a) **(20 pts.)** When the subroutine returns, what values are left in `ain`, `bin`, `count`, and `W`?

   (b) **(5 pts.)** How many instruction cycles does it take to execute the subroutine?

   (c) **(5 pts.)** How can the subroutine be changed to have approximately twice the speed but still return the same value in *W*?

   (d) **(5 pts.)** After this change, what values are left in `ain`, `bin`, `count`, and `W`?

4. **(5 pts.)** Convert the following binary number to hex.

1 1 1 0 1 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 1

Have a good break!

Email me or stop by if you want to know where to learn more about computer engineering.