EE224: End-semester Test

24 April 2014, 1730-2030 hrs

All questions have 5 marks. Make suitable assumptions if needed, and mention them clearly.

1. Suppose $f(x_1, x_2, x_3, x_4)$ is a function of four variables. Derive a procedure to check if it is possible to factorize

$$f(x_1, x_2, x_3, x_4) = g(x_1, x_2).h(x_3, x_4)$$

where g and h are functions of two variables (5 marks). Apply your procedure to find suitable factors g, h (or show that no such factors exist) if

$$f(x_1, x_2, x_3, x_4) = x_1.(x_4.x_2.\overline{x}_3 + x_3.\overline{x}_4.x_2) + \overline{x}_1.(\overline{x}_3.\overline{x}_2.\overline{x}_4 + \overline{x}_2.\overline{x}_3.x_4).$$

2. Consider the two state-machines shown in Figure 1. The left-machine has states $\{S0, S1\}$, input symbols $\{rL, f, s\}$ and output symbols $\{rR, u, v\}$. The right-machine has states $\{T0, T1\}$, input symbols $\{rR, u, v\}$ and output symbols $\{d, n\}$. The output of the left-machine is the input to the right-machine. Taken together, the pair of state machines is equivalent to a single FSM with four states, input symbols $\{rL, u, v\}$ and output symbols $\{d, n\}$. Draw the state-transition graph of this equivalent FSM.

3. Using two input NAND gates, inverters and D-flipflops, implement the state machine shown in Figure 2. The input symbols are $\{S, C, T, N\}$, the output symbols are $\{H, L\}$ and the states are $\{Q0, Q1\}$. Clearly show your input, output and state encodings and any intermediate work you do in getting to your implementation. Assume that all gates have unit delay, and the D-flipflop has delay, setup and hold times which are all equal to 1 unit. For your solution, find the

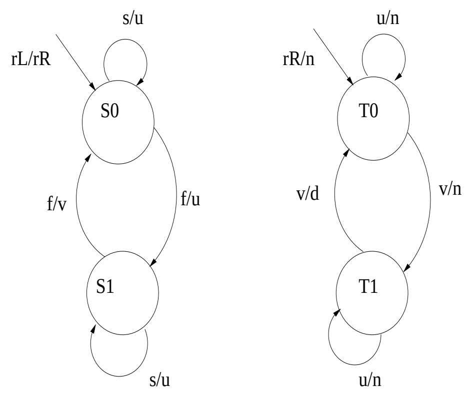

Figure 1: Interacting FSMs

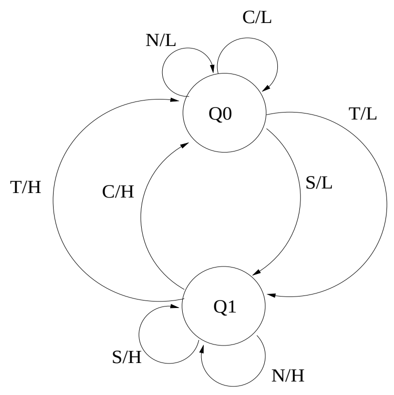

Figure 2: FSM specification

- input setup times relative to clock (for each input).
- input hold times relative to clock (for each input).
- delay from clock to output.

4. Consider the circuit shown in Figure 3. The capacitances shown correspond to wire capacitances. The variables x, y, z are input capacitances of the three gates. For the circuit, it is required that $C_{in} = 1$ and $C_{out} = 4$ units. Write the total delay of the circuit as a function of x, y, z (in τ units). For what values of x, y, z is the minimum delay achieved (assume that x, y, z are positive real numbers)? What is the minimum delay?

5. The pseudo-code in Figure 4 describes an 8-bit divider. Simulate the

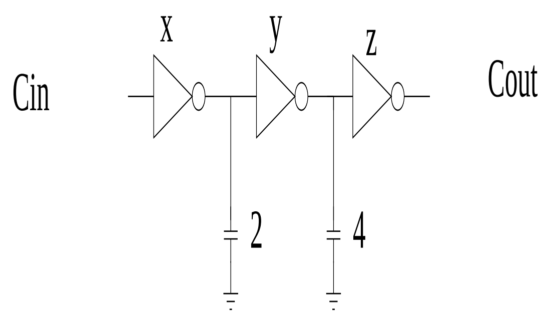

Figure 3: Inverter Chain

pseudo code (calculate the state and register values at the different k instants) until the div_done output signal becomes 1. The system starts in div_reset at $k = 0$ and at $k = 1$, div_start is 1. The divisor input is maintained at 00011100 and the dividend input is maintained at 10101110 throughout the period of interest. Does it work?

6. Consider the VHDL description of a piece of logic.

```
-- a, b, c, d are bits.
process(a,b,c,d, clk)
  variable x,y: bit;
begin
  x := (a or b);
  if(b = '1') then
    x := (not x);
 else
    x := (x or c);
 end if;
 if(c = '1') then
   x := (x and d);
 end if;
 e <= x;
 y := (not (x and (a or b)));
 if(clk'event and clk = '1') then
   q <= y;
 end if;
end process;
```

Draw a logic network which is equivalent to this process statement

```
// divisor, dividend are 8-bit inputs.
// remainder, dividend_by_2 shifted_divisor,
// quotient are all 8-bits wide registers.
reset:
 if div_start then
    goto div_outerloop/
          {remainder = dividend || quotient = 0}
 else
    goto div_reset
 endif

div_outerloop:
 if (divisor <= remainder) then
    goto div_innerloop/
        { curr_quotient = "0000000000000001" ||
            dividend_by_2 = (remainder >> 1) ||
            shifted_divisor = divisor}
 else
    goto div_completed
 endif

div_innerloop:
 if (shifted_divisor < dividend_by_2) then
    goto div_innerloop/
      { shifted_divisor = (shifted_divisor << 1) ||
        curr_quotient = (curr_quotient << 1) }
 else
    goto div_update
 endif

div_update:
 goto div_outerloop/
      {quotient = (quotient OR curr_quotient) ||
       remainder = (remainder - shifted_divisor)}

div_completed:
 goto div_reset/{emit div_done}
```

Figure 4: Divider

(you may use AND, OR, NOT gates and D-flip-flops).

7. You are asked to design a system which receives a sequence of bits $\{x(k)\}$ and produces an output sequence of bits $\{y(k)\}$ such that $y(k) = 1$ if and only if the bits $x(0), x(1), \ldots x(k)$ form a palindrome (that is a sequence which reads the same from left to right as it does from right to left, for example $1, 1, 0, 1, 1$). Can you implement such a system using a Mealy finite state machine? If yes, give the state transition graph of the FSM. If not, why not? (5 marks).

8. Suppose you are given a collection of combinational gates each of which is a four-input programmable logic block which can implement any four-input Boolean function. Using these gates, implement a majority voter function which has eight input bits and outputs a 1 if and only if least 5 of its input bits are 1 (5 marks).

9. Design a CMOS gate to implement the Boolean function $\overline{(A+B).(C+D)}$ with a single size parameter W and size it so that the pull-up and pull-down networks are equivalent to a $2W$ PMOS transistor and W NMOS transistor respectively. What is the logical effort of the gate relative to that of a standard inverter?

10. Do you agree or disagree with the following statements? In each case, give a short justification for your opinion.

    - In a well designed CMOS circuit, all nodes should have approximately the same rise/fall times.
    - In a well designed CMOS circuit, all paths should have approximately the same delays.
    - In a well designed CMOS circuit, all gates should be minimum sized.
    - A digital system with many clocks is easier to implement and verify than one with a single clock.
    - When implementing a digital system, most of the time should be spent in VHDL/Verilog coding.