

ECE 448
Midterm Exam 1
Thursday, March 8, 2007

Problem 1 (2 points)

For a given below VHDL code, provide solutions to the following problems:

1. Which type of finite state machine, Moore or Mealy, does this code implement?
2. Draw a state diagram of this FSM.
3. Supplement timing waveforms given in the answer sheet with values of the state *s* and output *q*.

```
library IEEE;  
use IEEE.std_logic_1164.all;
```

```
entity PROBLEM1 is  
port (  
  CLK: in STD_LOGIC;  
  d: in STD_LOGIC;  
  q: out STD_LOGIC_VECTOR (1 downto 0));  
end;
```

```
architecture PROBLEM1_arch of PROBLEM1 is  
  type Sreg0_type is (S1, S2, S3, S4);  
  signal Sreg0: Sreg0_type;  
begin  
  Sreg0_machine: process (CLK)  
  begin  
    if CLK'event and CLK = '1' then  
      case Sreg0 is  
        when S1 =>  
          if d='0' then  
            Sreg0 <= S1;  
          else  
            Sreg0 <= S3;  
          end if;  
        when S2 =>  
          if d='1' then  
            Sreg0 <= S3;  
          else  
            Sreg0 <= S1;  
          end if;  
        when S3 =>  
          if d='1' then  
            Sreg0 <= S3;  
          else  
            Sreg0 <= S4;  
          end if;  
      end case;  
    end if;  
  end process;  
end PROBLEM1_arch;
```

```

        when S4 =>
            if d='1' then
                Sreg0 <= S1;
            else
                Sreg0 <= S2;
            end if;
        when others =>
            null;
        end case;
    end if;
end process;

```

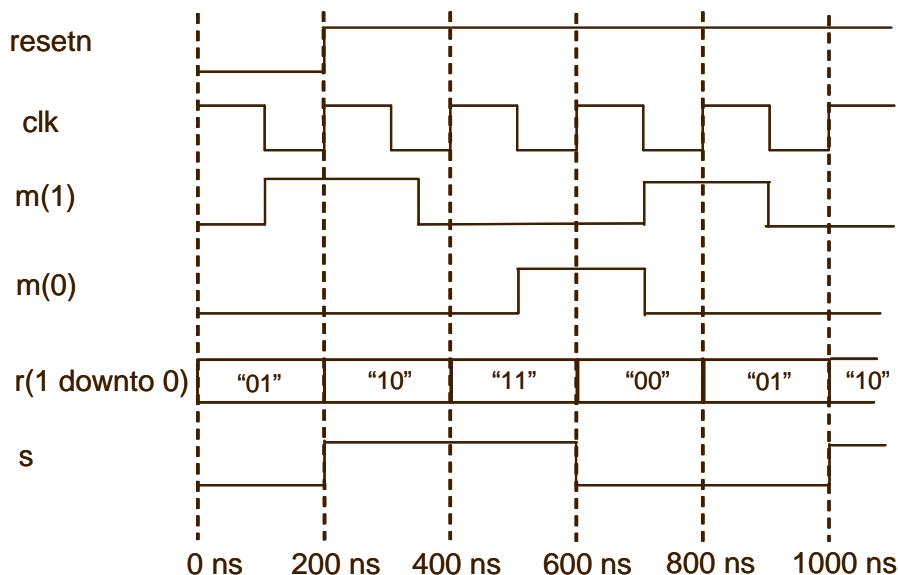
```

q_assignment:
    q <= "01" when (Sreg0 = S1 and d='0') else
        "11" when (Sreg0 = S1 and d='1') else
        "10" when (Sreg0 = S2 and d='1') else
        "00" when (Sreg0 = S2 and d='0') else
        "01" when (Sreg0 = S3 and d='1') else
        "10" when (Sreg0 = S3 and d='0') else
        "10" when (Sreg0 = S4 and d='1') else
        "00" when (Sreg0 = S4 and d='0') else
        "00";
end PROBLEM1_arch;

```

Problem 2 (2 points)

Write a testbench that generates the following input stimuli. Assume that the signals `clk`, `r` and `s` are periodical, and the signals `resetn` and `m` are non-periodical. The simulation should be able to continue indefinitely under this assumption. Do your best to minimize the amount of lines of VHDL code you write, by employing VHDL instructions such as `FOR LOOP`, etc.



Problem 3 (2 points)

Draw a full schematic of the internal structure of the component COMPAREN corresponding to the following instantiation of this component. The component COMPAREN is described below using entity declaration and entity architecture.

-- component instantiation in the top level entity

```
CMP8: COMPAREN
  generic map(N => 5)
  port map(A => A,
           B => B,
           CMP_IN => SIG_IN,
           CMP_OUT => SIG_OUT,
           G => G
  );
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
```

```
entity COMPAREN is
  generic(N: positive); -- N – width of operands
  port(
    A, B : in STD_LOGIC_VECTOR(N-1 downto 0);
    CMP_IN : in STD_LOGIC_VECTOR(1 downto 0);
    CMP_OUT : out STD_LOGIC_VECTOR(1 downto 0);
    G : out STD_LOGIC_VECTOR(N-1 downto 0));
end COMPAREN;
```

```
architecture STRUCTURE of COMPAREN is
  component BCOMP
    port(A, B, X_IN, Y_IN: in STD_LOGIC;
         X_OUT, Y_OUT: out STD_LOGIC);
  end component;
  signal INT_X, INT_Y: STD_LOGIC_VECTOR(N downto 0);
begin
  INT_X(N) <= CMP_IN(1);
  INT_Y(N) <= CMP_IN(0);

  CASCADE: for I in N-1 downto 0 generate
    C: BCOMP port map(A(I), B(I), INT_X(I+1), INT_Y(I+1),
                     INT_X(I), INT_Y(I));
    D: G(I) <= A(I) when INT_X(I) = '1' else
         B(I);
  end generate;

  CMP_OUT(1) <= INT_X(0);
  CMP_OUT(0) <= INT_Y(0);
end STRUCTURE;
```

