2017 Paper E2.1: Digital Electronics II

Answer ALL questions.

There are THREE questions on the paper.

Question ONE counts for 40% of the marks, other questions 30% each

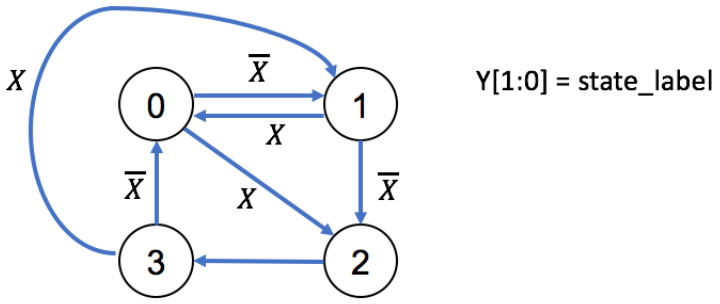
Time allowed: 2 hours

SOLUTIONS

(Not to be removed from the Examination Room)

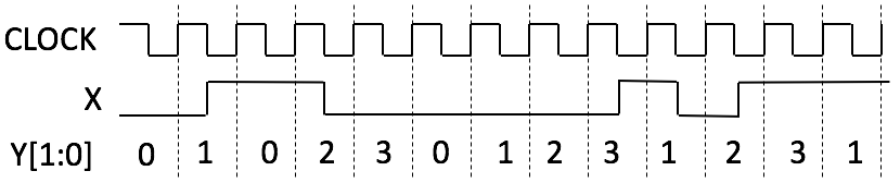
1. (a) This question examine student's basic understand of a FSM and how to specify a FSM in Verilog.

(i)



[3]

(ii)



[2]

(iii)

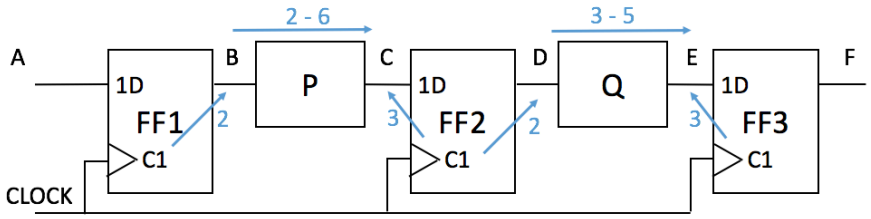
```
module FSM (CLOCK, X, Y);
   input X;
   input CLOCK;
   output reg [1:0] Y;

   // declare and define states
   reg [3:0]   state;
   parameter   S0 = 4'b0001, S1 = 4'b0010,S2 = 4'b0100, S3 = 4'b1000;

   initial state = S0;

   // state transitions
   always @ (posedge CLOCK)
      case (state)
         S0:   if (X==1'b0) state <= S1;
               else  state <= S2;
         S1:   if (X==1'b0) state <= S2;
               else  state <= S0;
         S2:   state <= S3;
         S3:   if (X==1'b0) state <= S0;
               else  state <= S1;
         default: state <= S0;
      endcase

   // FSM outputs
   always @ (*)
      case (state)
         S0:   Y = 2'b00;
         S1:   Y = 2'b01;
         S2:   Y = 2'b10;
         S3:   Y = 2'b11;
         default: Y = 2'b00;
      endcase
endmodule
```

[3]

(Almost all student did well on this question. It tests basics on FSM and Verilog. A few students obviously did not do the Lab or understood the fundamentals of describing digital hardware in Verilog, and missed out the "always @ …".)

(b) This question tests student's understanding of timing constraints in digital circuits.

(i)



Worst case CLOCK -> C delay = 2 + 6 = 8,

and CLOCK -> E delay = 2 + 5 = 7. Therefore the first stage dictates maximum operating frequency.

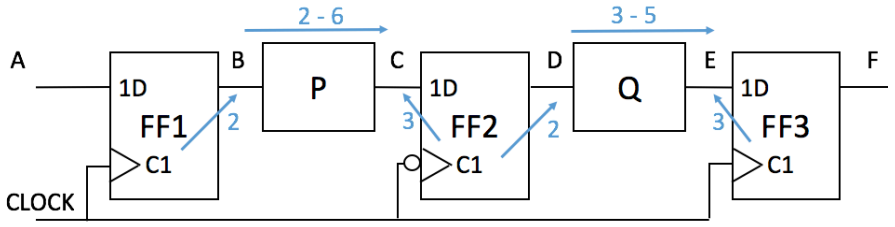
fmax = 1/(8+3) = 90.91MHz.

[3]

(ii)

Assume that CLOCK is symmetrical, the clock frequency will be halved at 45.46MHz.

This is NOT the fastest possible clock frequency. If the worst case high period of the clock is 11ns, and the low period of the clock is 10ns. Therefore if the CLOCK has a mark-space ratio of 11:10, then the clock period is 21ns. Therefore under this assumption, fmax = 1000/21 = 47.62MHz.



[5]

(Most student did well on this question. However, only a small number of student got right the assumption that the mark-space ratio not being equal in order to maximize the operating clock frequency.)

(c) This question tests student's knowledge in designing a combinatorial circuit in Verilog, and show some knowledge in the FPGA internal architecture

(i)

```
module decoder (X, Y);
    input   [3:0]   X;
    output      [3:0] Y;

    reg [3:0]   Y;

    always @ (X)
        case (X)
        //****** input <5, pass to output unchanged ******
            4'b0000: Y <= 4'b0000;
            4'b0001: Y <= 4'b0001;
            4'b0010: Y <= 4'b0010;
            4'b0011: Y <= 4'b0011;
            4'b0100: Y <= 4'b0100;

        //****** input >=5 and < 12, output = input + 3 *****
            4'b0101: Y <= 4'b1000;
            4'b0110: Y <= 4'b1001;
            4'b0111: Y <= 4'b1010;
            4'b1000: Y <= 4'b1011;
            4'b1001: Y <= 4'b1100;
            4'b1010: Y <= 4'b1101;
            4'b1011: Y <= 4'b1110;

        //****** input >= 12, output = input - 2    ******
            4'b1100: Y <= 4'b1010;
            4'b1101: Y <= 4'b1011;
            4'b1110: Y <= 4'b1100;
            4'b1111: Y <= 4'b1101;
        default: Y <= 4'b0000;
        endcase
endmodule
```

[6]

(ii)

Since each ALM has 8 inputs and two independent outputs, this would take 2 ALMs to implement.

[2]

(This is a straight forward question that most students got right.)

(d) This question tests student's understanding of basic DAC ideas.

(i) Resolution is 3.3V/4096 = 0.806mV per bit.

[2]

(ii) The reference voltage source "sees" a resistance of R (basic concept of the R-2R network is that every node downstream appears as a resistor R).

$I_{in} = 16\ I_o = Vref/R$, therefore $I_o = 3.3V/(16 \times R) = 0.206/R$ A.

With switch set at 9, the current flowing into the virtual earth terminal is $9*I_o$.

Therefore $V_{out} = - 9*0.206 = -1.854V$.

[6]

[2]

(EEE students on the whole did better in this question than EIE students.)

(e) This question tests student's basic understanding of address decoding.

(i)

Signal	Address range
RAM_CS	0000 – 7FFF
ROM_CS	A000 - BFFF
IO1_CS	FE00 – FE03
IO1_CS	FF00 – FF03

[4]

(ii)

RAM_CS = A15

ROM_CS = /(A15*/A14*A13)

IO1_CS = /(A15*A14*A13*A12 * A11*A10*A9*/A8 */A7*/A6*/A5*/A4*/A3*/A2)

IO2_CS = /(A15*A14*A13*A12 * A11*A10*A9*A8 */A7*/A6*/A5*/A4*/A3*/A2)

[4]

(Most students scored well in this question.)

2.

(a)

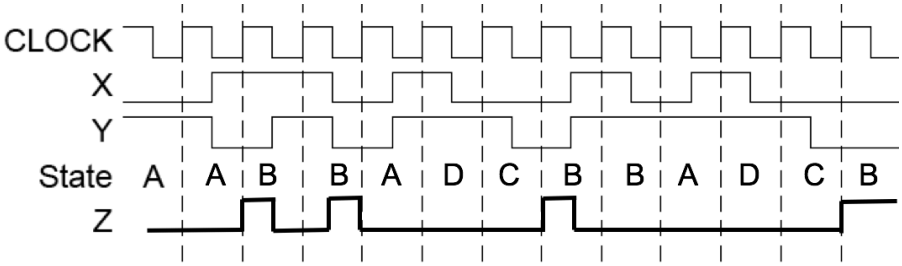
State / Z	X, Y			
	00	01	10	11
A	A	A	B	D
B	A/1	A	B	B
C	B	D	B	D
D	A	C	B	D

Input/Output: X,Y/Z

Default: Z=0

[8]

(b)



[6]

(c)

A3:0				D2:0		
S1	S0	Y	X	Z	NS1	NS0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	0	0
0	1	1	1	0	0	1

[10]

(d) Maximum frequency = 1/ (5+2+1) = 125MHz.

[2]

(e) If one-hot encoding (instead of binary encoding as shown here) were to be used, the address will need to have 6 bits. Therefore, the ROM would need to be 64 x 5-bit.

One-hot encoding is good when FSM is implemented using FPGA logic cells, which are register rich. Since state decoding is not needed, it generally requires less logic to implement. However, ROM size increases exponentially with number of address bits, and ROM can implement arbitrary complex logic (since it is only a lookup table), it is resource inefficient to use one-hot encoding for states.

[4]

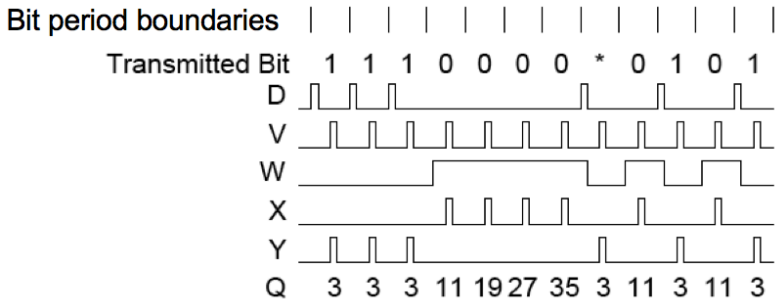
(Most students scored well in this question.)

3. This question tests student's ability to use counter circuit to perform serial communication with digital circuits.

(a) Due to the AND operation, pulses on X or Y can only occur when Q[2:0] = 3'b011 which means that Q[5:0] = 3+8k for some integer k.

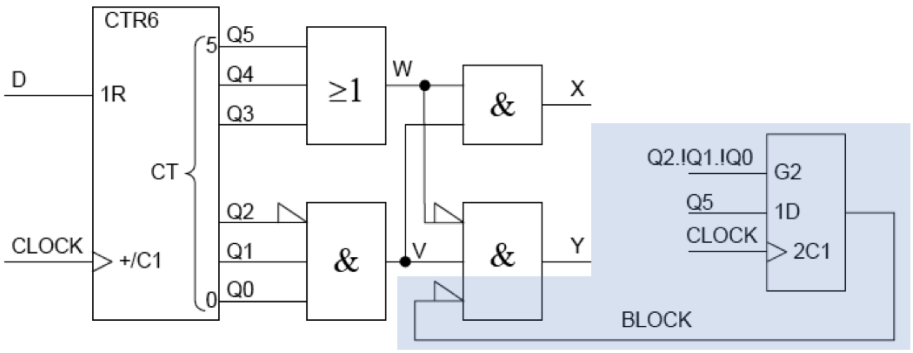
The falling edges of W occur when the counter resets to zero which is immediately at the end of the input pulses.

Pulses on V occur midway between the input pulses. In the timing diagram below, Q is the value of Q[5:0].



[15]

(b) We need to remember that we have had four consecutive zeros and then suppress the next pulse. The easiest way to do this is to use a flipflop with a clock enable input to remember the value of Q5 at the last pulse:



.

[15]

[2]

(This is clearly the hardest question in the paper. Most student could only attempt a) and very few got b) right.)