CSEC 20th – Past Year Paper Solution 2016-2017 Sem2

CE/CZ1005 – Digital Logic

Question 1

1)

a)

i) 1011 1011

ii) 1100 0101

b)

i) 110 //gray code can only change one bit at a time

ii) 101 //without having to repeat the number, the only number left is 101

c) (a + b + c)'[(a + b c' + d')' (a' c + b' d')]'

= (a' b' c')[(a + b c' + d') + (a' c + b' d')']

= (a' b' c')[(a + d' + c' b + c' d)]

= (a' b' c')[(a + d' + c' b + c' d)]

= a' b' c' d' + a' b' c' d

= a' b' c'

d)

i) [(w' + x + y)(w' + x + z')(w' + y + z') (x + y + x')]'

ii) Refer to Fig1 below

iii) active low output is high (asserted) when logic = 0

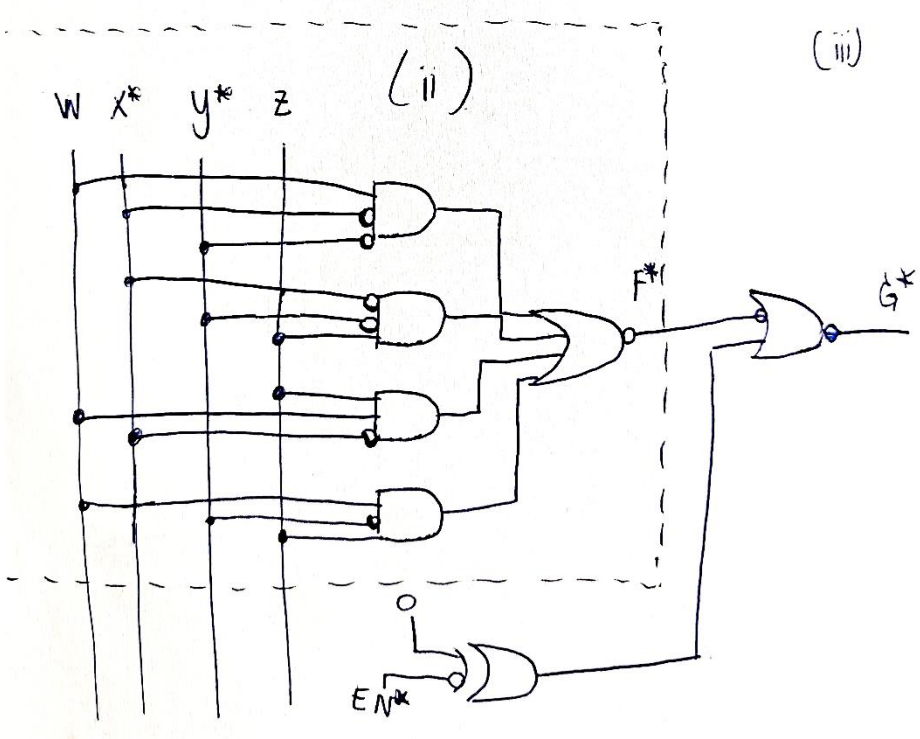


Fig1: Answers for (ii) and (iii)

Question 2

2)

a)

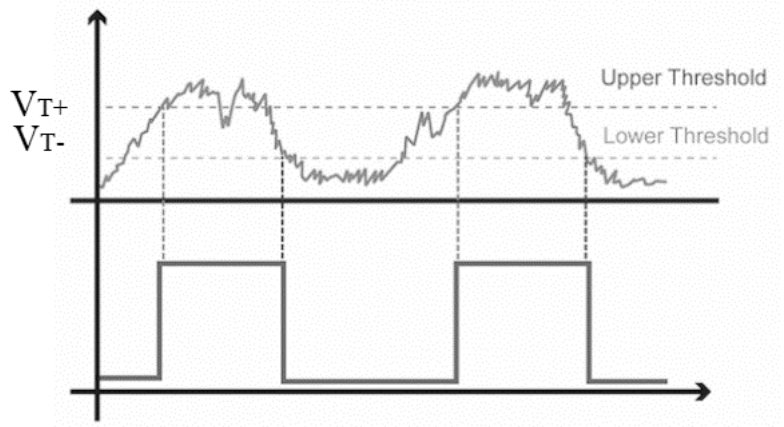


Fig2: Answers for (2a)

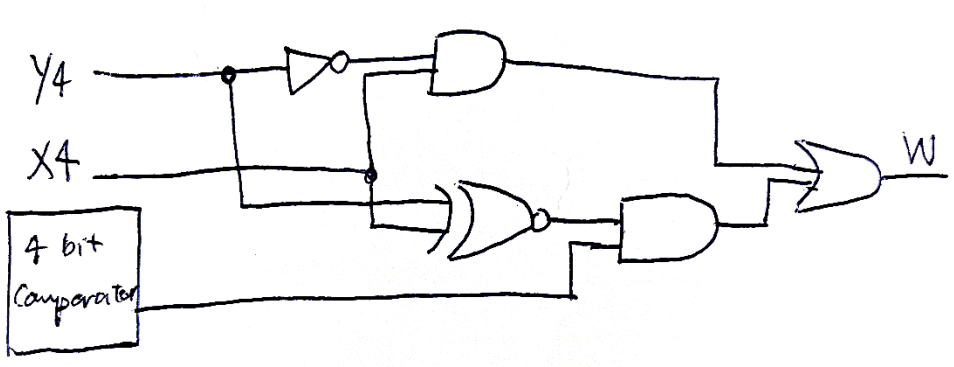
Source: https://www.allaboutcircuits.com/industry-articles/an-introduction-to-schmitt-triggers-otto-schmitt/

b) a' c' d' + b' c' d' + a' c d + a' b c

c,d \ a,b	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	1	1	0	0
10	0	1	0	0

c)

i)



ii) The output of W will be 1 when:

1. X4 =1 and Y4 = 0
2. X4 = Y4 and output from 4-bit comparator is 1

The XNOR gate compares if X4 = Y4

Question 3

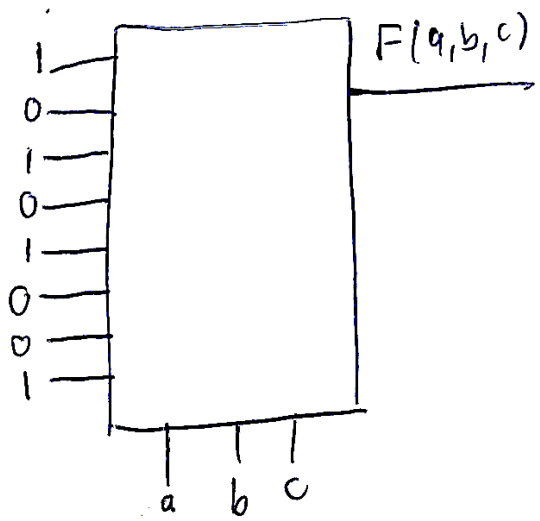
3)

a)
```
module finderrors(input[1:0] a, b, c,
                  input[1:0] sel, output[1:0] reg result);
always @ *
begin
   case (sel)
        2'b00: result = a;
        2'b01: result = b;
        2'b10: result = c;
        Default: result = 2'b0;
   endcase
end
endmodule
```

b)
```
module fixit(input p,q,r,z,
             output reg X,Y);
always @ *
begin
   if(p)
   begin
        X = q & r;
   end
   else
   begin
        X = z;
        Y = q^z;
   end
end
endmodule
```

c)

i)



ii) //assuming inp tp be the input a, b, c with a being the MSB and c being the LSB

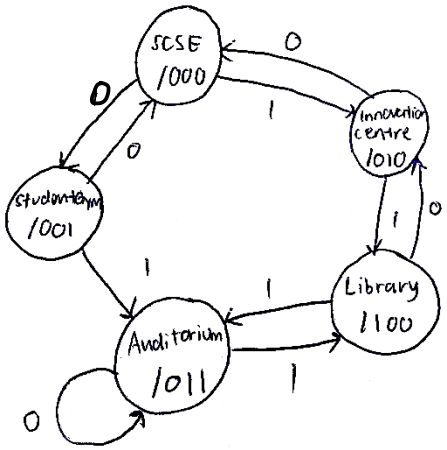
```
module (input[2:0] inp
        Output reg result);
always @ *
begin
case (inp)
   3'b000: result = 1;
   3'b001: result = 0;
   3'b010: result = 1;
   3'b011: result = 0;
   3'b100: result = 1;
   3'b101: result = 0;
   3'b110: result = 0;
   3'b111: result = 1;
end
endmodule
```

Question 4

4)

a)

i)



ii) SCSE -> 0 -> 1 -> 1 -> 0 -> 0

iii)

```
module autoveh (input MOVE,
                input reset, clk,
                 output[2:0]LOC);
parameter SCSE =3'b000, StudentGym = 3'b001,
Auditorium = 3'b011, Library = 3'b100,
InnovationCentre = 3'b010;

reg[2:0] nst, st;

assign LOC = st;
always @ (posedge clk)
begin
   if(reset)
        st <= 3'b000;
   else
        st <= nst;
end
always @ *
begin
   nst = st;
   case (st)
        SCSE:
             if(MOVE)
             begin
                  nst = InnovationCentre;
                  break;
             end
             else
             begin
                  nst = StudentGym;
                  break;
             end
        StudentGym:
             if(MOVE)
             begin
                  nst = Auditorium;
                  break;
             end
             else
             begin
                  nst = SCSE;
                  break;
             end
        Auditorium:
             if(MOVE) begin
                  nst = Library;
                  break;
```

```
                        end
                        else
                        begin
                              nst = Auditorium;
                              break;
                        end
                  Library:
                        if(MOVE) begin
                              nst = Auditorium;
                              break;
                        end
                        else begin
                              nst = InnnovationCentre;
                              break;
                        end
                  InnovationCentre:
                        if(MOVE)
                        begin
                              nst = Library;
                              break;
                        end
                        else begin
                              nst = SCSE;
                              break;
                        end
            endcase
         end
         endmodule
```

b)
```
module piso(input [3:0] IN, clk, LOAD,
                  output out);
      reg [3:0] q;
      always @ (posedge clk)begin
            if(LOAD)
                  q <= IN;
            else begin
                  q[3:0] <= { q[2:0],IN[0]};
            end

      end

      assign out = q[3];

   endmodule
```

== End of Answers ==

Solver: Lim Yan Jun

Email Address: ylim085@e.ntu.edu.sg