CSEC 20th – Past Year Paper Solution 2018-2019 Sem2

CE/CZ1005 – Digital Logic

Question 1

1)

a) Answer: D6.4E97

Convert to Binary number first: Integer Part :

214 = 128 + 64 + 16 + 4 + 2

214 = 11010110_2

Fractional Part :

0.307 x 2 =	0.614
0.614 x 2 =	1.228
0.228 x 2 =	0.456
0.456 x 2 =	0.912
0.912 x 2 =	1.824
0.824 x 2 =	1.648
0.648 x 2 =	1.296
0.296 x 2 =	0.592
0.592 x 2 =	1.184
0.184 x 2 =	0.368
0.368 x 2 =	0.736
0.736 x 2 =	1.472
0.472 x 2 =	0.944
0.944 x 2 =	1.888
0.888 x 2 =	1.776
0.776 x 2 =	1.552

0100 1110 1001 0111

214 = $11010110.\ 0100\ 1110\ 1001\ 0111_2$

= D6.4E97

b)

Decimal	Binary	Grey
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Grey code change one bit when traversing from one point to another. In a not non-synchronized process, for a normal binary digit system, incrementing or decrementing can leave the number in a variety of intermediary states.

c) Answer: ac' + ad + b'cd + e'

F(a, b, c, d, e) = [(c' + b e) (a' e + b' c)]' [(c d') + e']

(a' c' e + b' c c' + a' b e + b b' c e)' (c' + d + e')

= (a' c' e + a' b e)' (c' + d + e')

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

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

= (a + e' + c b') (c' + d + e')

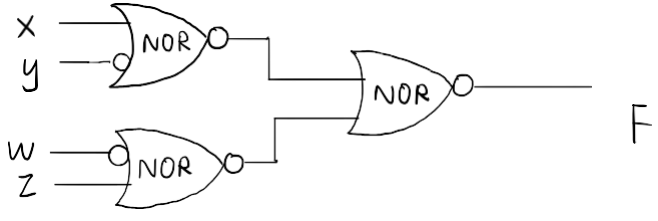
= a c' + a d + a e' + e' c' + e' d + e' + b' c c' + c b' d + c b' e'

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

d)

NOR Gate (A+B)'

F = ((x + y')' + (w' + z)')



e)

AB/CD	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	1	0	1

Circle 0

Answer: F = (B'+ D)(A+B+D')(B+C'+D')

Question 2

2)

a) 10001100
+10011010
100100110

There is overflow, the 2 same sign bit added and got an answer with the opposite sign. Add two negative number and get a positive answer indicates that there is overflow.

01001010
+00111111
10001001

No overflow

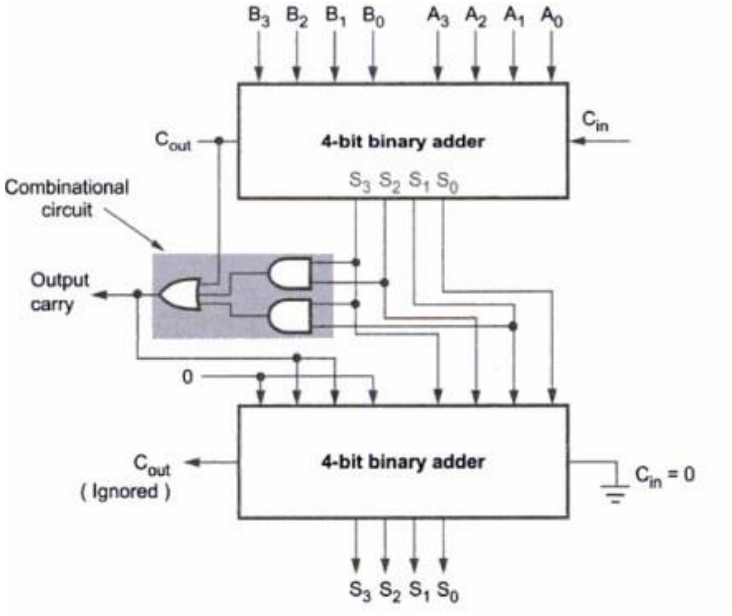
b)

i) X=1 when sum >= 9

Sum[3:0]	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

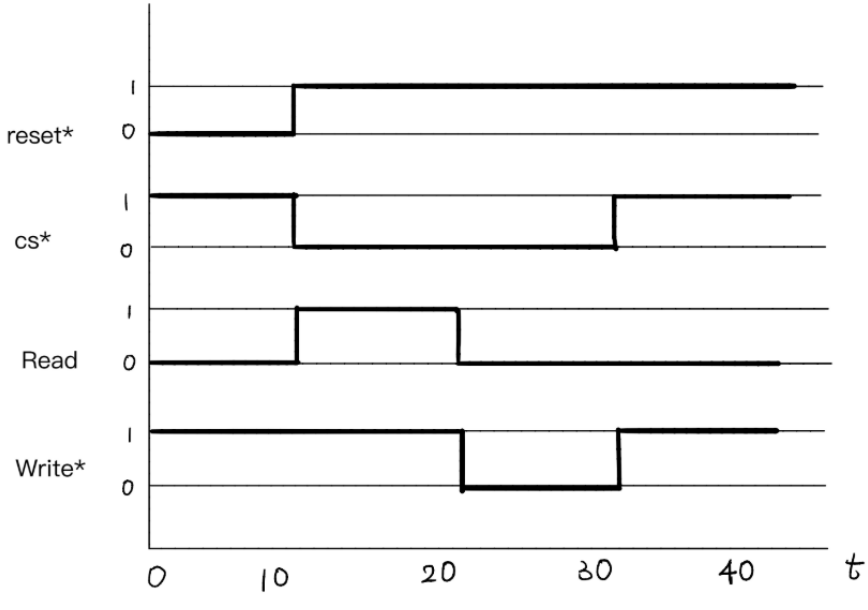
X = sum[3]sum[2] + sum[3]sum[1]

ii)



Source: https://www.ques10.com/p/15365/implement-single-digit-bcd-adder- using-4-bit-bin-2/

c)



d)

i) Z = 1

Boolean Expression: A'(B' + C') + D'

B = 0 or C = 0 and A =0 or D = 0, Z produces logic 1.

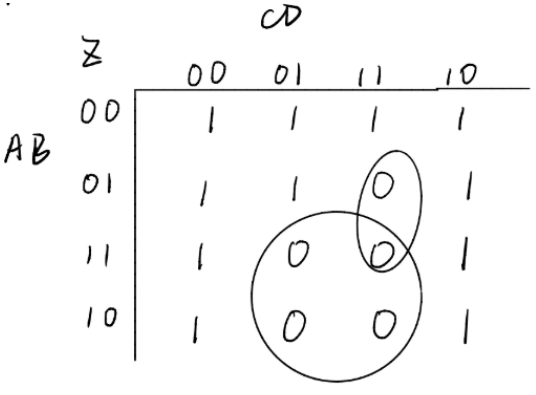
Z = 0

Boolean Expression: (A + BC)D

B = 1 and C = 1 or A = 1 and D = 1, Z produce logic 0

ii)

A	B	C	D	Z
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0



Answer:

Z = (A' + D') (B' + C' + D')

Question 3

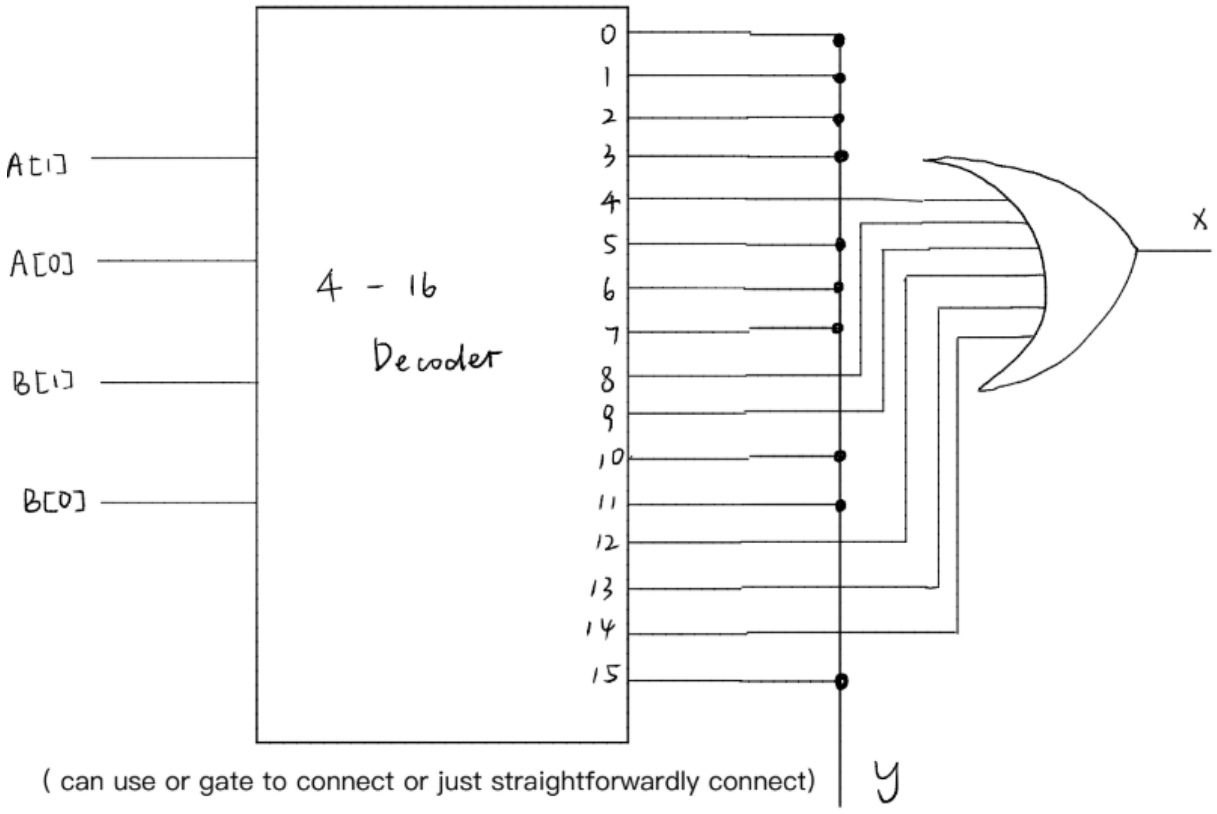
3)

a)

i)

A[1]	A[0]	B[1]	B[0]	X	Y
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1 4	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1 8	0
1	0	0	1	1 9	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	1 12	0
1	1	0	1	1 13	0
1	1	1	0	1 14	0
1	1	1	1	0	1

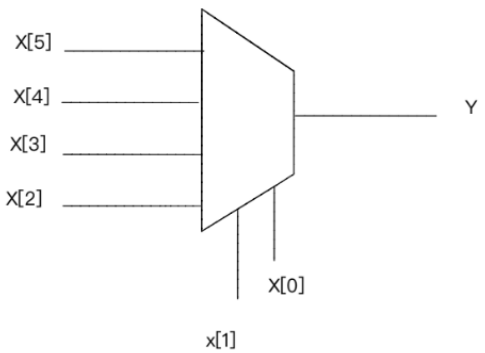
ii)



b)

```
//write the module for the 2-4 decoder

module decoder2_4 (output reg [3:0] d_out,
                   input [1:0] ival)
always @*
    case(ival)
        2'b00 : d_out = 4'b0001;
        2'b01 : d_out = 4'b0010;
        2'b10 : d_out = 4'b0100;
        2'b11 : d_out = 4'b1000;
    endcase
endmodule

//combinational circuit, instantiate decoder module

module decoder_app (input [5:0] X,

                    output Y)
    wire [3:0] d;
    wire xd1, xd2, xd3, xd4;

    decoder2_4 dd (.output(d[3:0]),
                   .input(x[1:0]));

    assign xd1 = x[2]&d[0];
    assign xd2 = x[3]&d[1];
    assign xd3 = x[4]&d[2];
    assign xd4 = x[5]&d[3];

    assign Y = xd1|xd2|xd3|xd4;
endmodule
```

c) It is a 4 input multiplexer with 2 selection pin.



d)

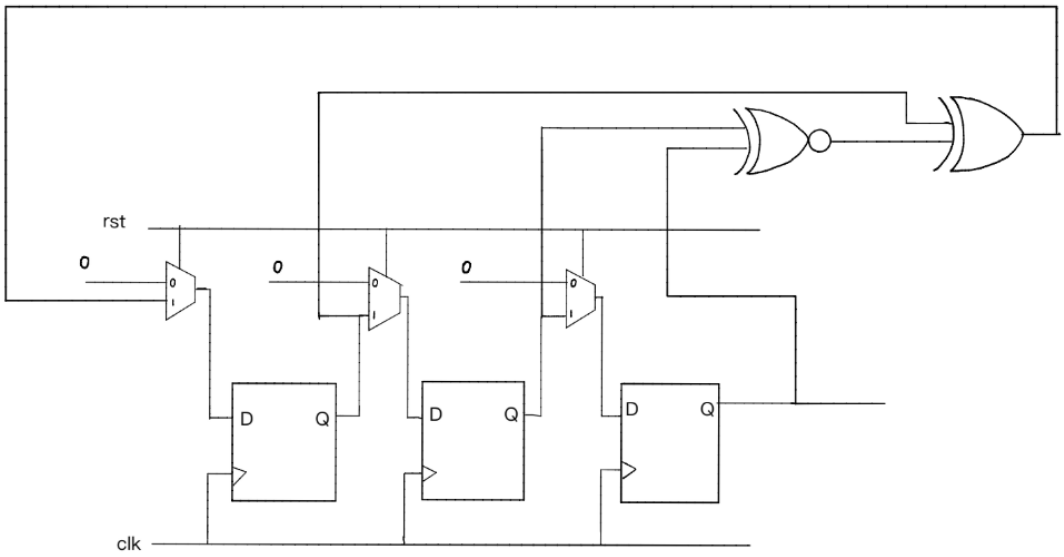
```
  module mux_4to1 ( input [5:0]  X, output reg Y);
  always @ * begin
     case(X [1:0])
        2'b00: out <= X[2];
        2'b01: out <= X[3];
        2'b10: out <= X[4];
        2'b11: out <= X[5];
      endcase
    end

   endmodule
```

Question 4

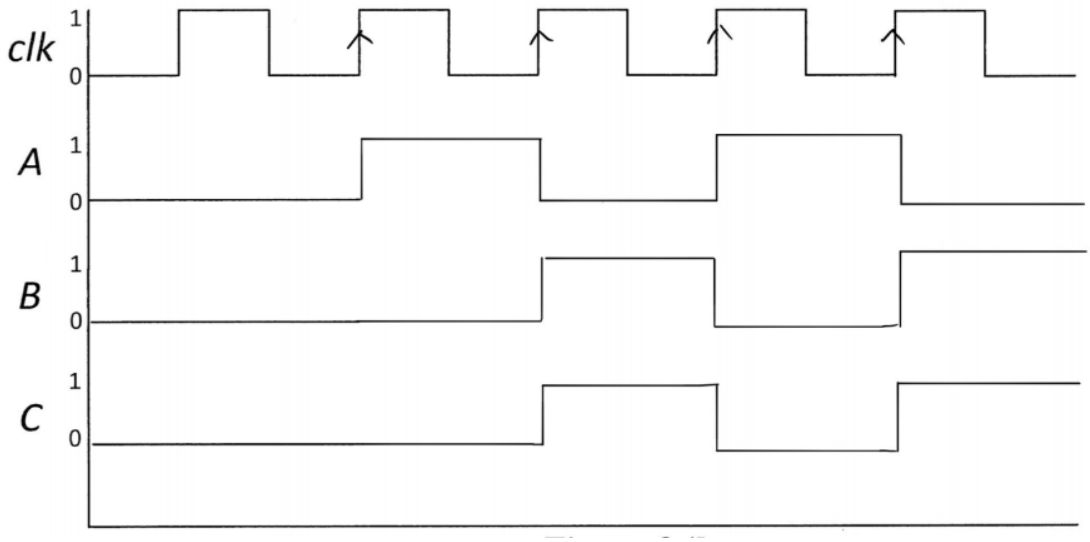
4)

a)

i)



ii)



b)

i)

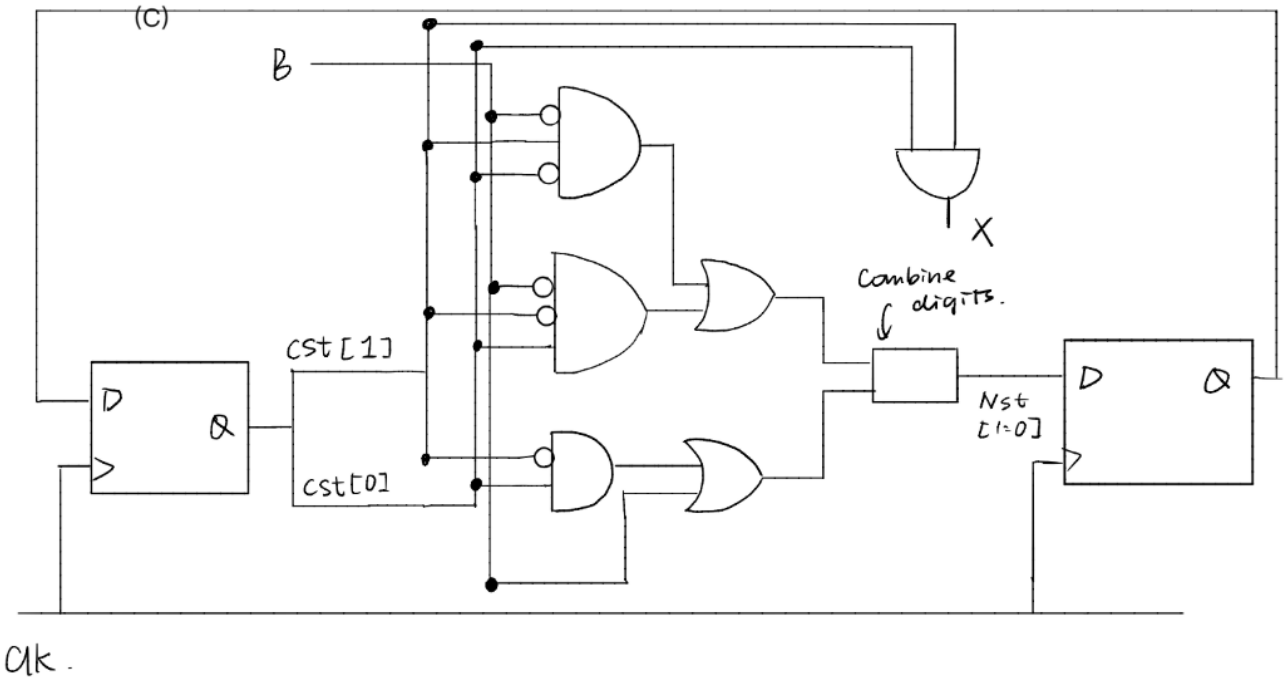
B	Current_State	Next_State [1:0]	X
0	00	00	0
1	00	10	0
0	01	11	0
1	01	10	0
0	10	01	0
1	10	10	0
0	11	00	1
1	11	10	1

ii)

cst[1] cst[0] B	Next_State[1]	Next_State[0]
000	0	0
001	1	0
010	1	1
011	1	0
100	0	1
101	1	0
110	0	0
111	1	0

Next_State[1] = B + cst[1]'cst[0]
Next_State[0] = cst[1]'cst[0]B' + cst[1]cst[0]' B'
X = cst[1]cst[0]

c)



== End of Answers ==

Solver: Yang Yubei
Email: C180052@e.ntu.edu.sg