Solver: Cai Lingzhi

Email Address: LCAI004@e.ntu.edu.sg

1 (a) (i).

For Hex value "CD",

Hex	Binary
C	1100
D	1101

The 8-bit data should be expressed as the following, while the parity bit being the leftmost bit.

	1	1	0	0	1	1	0	1
parity	C				D			

Since there are 5 "1"s in the 8-bit data, the value of the parity bit should be "0".

0	1	1	0	0	1	1	0	1
parity	C				D			

ANS:

0	1	1	0	0	1	1	0	1

(ii).

ANS:

The receiver has received the data "111001001", which, obviously, is incorrect comparing to the answer we have in the first part of the question.

Data Received:

1	1	1	0	0	1	0	0	1
parity	C				9			

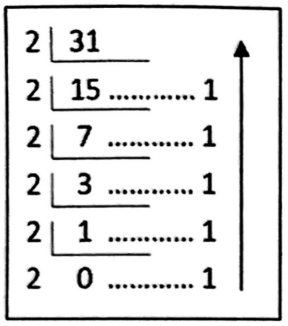
However, the receiver will NOT be able to detect the error using the parity method. Since there are 4 "1"s in the data, the parity bit should be "1", and the received data is apparently correct.

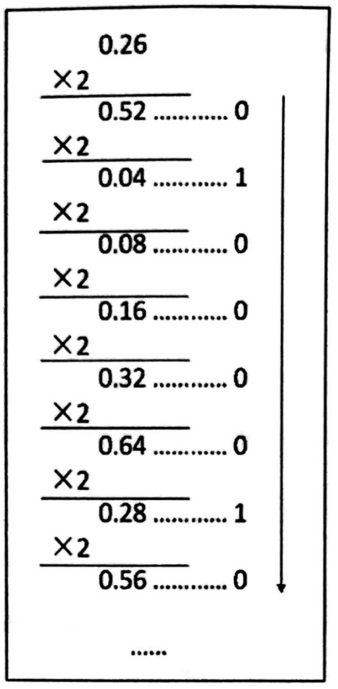
1 (b) (i).

ANS:

The unsigned decimal value 31.26 = 31 + 0.26.

∵





$\therefore 31_{(10)} = 11111_{(2)}, 0.26_{(10)} = 0.01000010\ldots\ldots_{(2)}$

$\therefore 31.26_{(10)} \approx 11111.0100001_{(2)}$ (12 significant bits)

(ii).

ANS:

A 18-digit decimal integer can represent a number with the maximum value of 10^{18}.

$\because \lceil \log_2 10^{18} \rceil = 60$,

i.e., the minimum integer that is greater or equal than $(\log_2 10^{18})$ is 60,

$\therefore$ The minimum number of bits required to represent an unsigned 18-digit decimal number is 60.

1 (c).

F = [(x'y') + (wyz)']' + (wz+x) (w'+y'+xz)'

= **[(x'y')' (wyz)'']** + (wz+x) **[w''y''(xz)']**

= [**(x+y)** (wyz)] + (wz+x)[wy**(x'+z')**]

= **(wxyz + wyz)** + (wz+x)**[wx'y + wyz']**

= (wxyz + wyz) + **(wx'yz + <u>wyzz'</u> + <u>wxx'y</u> + wxyz')** // wyzz' and wxx'y equals to 0

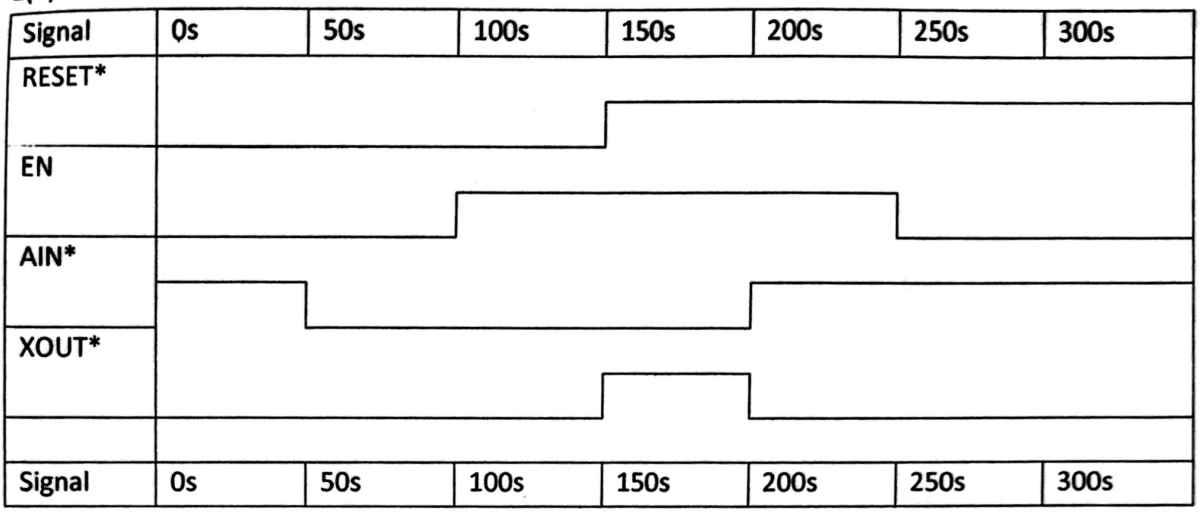
= wxyz + wyz + wx'yz + wxyz'

= (wxyz + wx'yz) + (**wxyz** + wxyz') + wyz // A = A + A

= wyz + wxy + wyz

= wxy + wyz

1(d).



2 (a).

/*As the invigilator illustrated, the destination "Local cities" is NOT considered as "export destination in Asia".*/

Let the BCD-input be a_3, a_2, a_1, a_0. We can draw the truth table.

MSB	BCD a_3, a_2, a_1, a_0	Output
0	0000	0
1	0001	0
2	0010	1
3	0011	1
4	0100	1
5	0101	0
6	0110	0
7	0111	1
8	1000	1
9	1001	0

(i).

ANS: (SOm) $Z = m2 + m3 + m4 + m7 + m8$

$= a_3'a_2'a_1a_0' + a_3'a_2'a_1a_0 + a_3'a_2a_1'a_0' + a_3'a_2a_1a_0 + a_3a_2'a_1'a_0'$

(ii).

ANS: (POM) $Z = M0 \cdot M1 \cdot M5 \cdot M6 \cdot M9$

$= (a_3+a_2+a_1+a_0) \cdot (a_3+a_2+a_1+a_0') \cdot (a_3+a_2'+a_1+a_0') \cdot (a_3+a_2'+a_1'+a_0) \cdot (a_3'+a_2+a_1+a_0')$

(iii).

ANS:

The K-map is drawn as below:

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	1	1
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

// X: Don't Care

which can be expressed as:

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	1	1
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

a_1a_0

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	1	1
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

a_3a_0'

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	1	1
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

$a_3'a_2'a_1$

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	1	1
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

$a_2a_1'a_0'$

// It's okay to draw all the loops in one diagram.

$\therefore$ (SOP) $Z = a_1a_0 + a_3a_0' + a_3'a_2'a_1 + a_2a_1'a_0'$

2 (b) (i).

ANS: Component U is a tristate inverter and component V is a tristate buffer.

(ii).

ANS: The input E1 and E2 are the enable input of the transistors, which can control whether the current will be able to flow through U and V.

(iii).

ANS: If both E1 and E2 are logic 1, i.e. both U and V are enabled, signal A and B will be both connected to the output, and the outcome will be uncertain. The excessive voltage from two input may also cause damage to the output.

(iv).

ANS:

E1	E2	F
0	0	Z
0	1	B
1	0	A

Z denotes to high-impedance state.

2 (c) (i).

$\because X = 0x63$
$= 0110\ 0011_2$
$= 2^6 + 2^5 + 2^1 + 2^0$
$= 99_{10}$

$\because Y = 0xB5$
$= 1011\ 0101_2$
$= -(0100\ 1011_2)$
$= -(2^6 + 2^3 + 2^1 + 2^0)$
$= -75_{10}$

ANS: X = 99; Y = -75.

(ii).

ANS: X+Y:

	0	1	1	0	0	0	1	1
+	1	0	1	1	0	1	0	1
(1)	0	0	0	1	1	0	0	0

There is NO overflow here.

(iii).

ANS:

$\because -Y = 0100\ 1011_2$

$\therefore X - Y = X + (-Y)$:

	0	1	1	0	0	0	1	1
+	0	1	0	0	1	0	1	1
	1	0	1	0	1	1	1	0

There is overflow here.

3 (a) (i)

ANS: The circuit can be used as an oscillator, and generates a signal for every 7ns when enabled.

(ii)

$T = 5 \times 1ns + 2ns = 7ns = 7 \times 10^{-9}s$

$f = 1/T$
$= 1/7 \times 10^{-9}s$
$= 1.429 \times 10^{8}\ Hz$

ANS: The frequency of the circuit is $1.429 \times 10^{8}\ Hz$.

(iii)

ANS (1):

```
module Q3a_1 (input enable,
              output Op);

and (w1, Op, enable);
not (w2, w1);
not (w3, w2);
not (w4, w3);
not (w5, w4);
not (Op, w5);

endmodule
```

ANS (2):

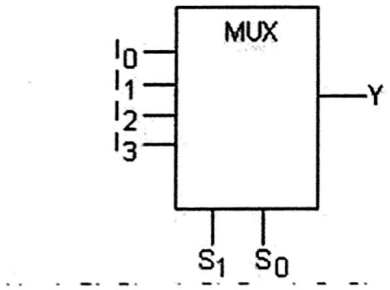
```
module Q3a_2 (input enable,
              output Op);

assign w1 = Op & enable;
assign w2 = ~ w1;
assign w3 = ~ w2;
assign w4 = ~ w3;
assign w5 = ~ w4;
assign Op = ~ w5;

endmodule
```

3 (b) (i).
/*This question is a bit tricky haha.*/

A 4x1 multiplexer looks like this. There are six inputs in total, and one output.



//Source:

http://4.bp.blogspot.com/-UvZjpzIaIrM/Up67_lgX2YI/AAAAAAAAA2c/WAmBsNsT5pk/s1600/BLOCK+DIAGRAM+OF+4X1+MUX+OR+MULTIPLEXER.PNG

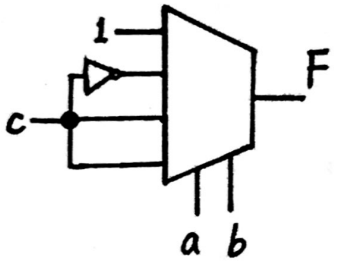
ANS:
Firstly we draw the truth table of F:

a	b	c	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Which can be simplified as:

a	b	c	F
0	0	0	1
		1	1
0	1	0	1
		1	0
1	0	0	0
		1	1
1	1	0	0
		1	1

From the truth table, we can find out that if we use a and b as select inputs, the mux should be connected as following:



(ii). **ANS:**

```
module Q3b (input a,b,c,
            output reg F);

reg [1:0] sel;
assign sel = {a,b};     // make sel a 2-bit register with MSB being a, and LSB being
b.

always @*
begin
      case(sel)
            2'b00 : F = 1;
            2'b01 : F = ~c;
            2'b10 : F = c;
            2'b11 : F = c;
            default : F = 0;     // Not required in the question though…
      endcase
end

endmodule
```

4 (a).

ANS:

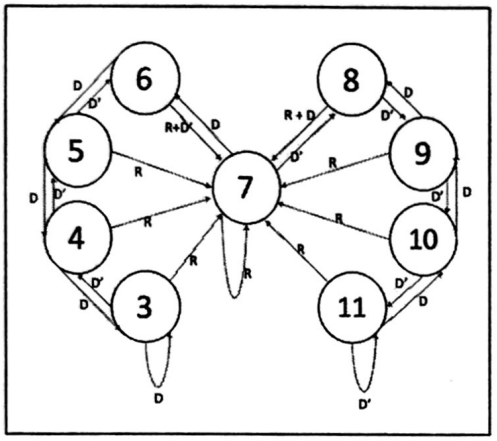
"For combinational always blocks, we always use a blocking assignment (=), and *order matters.* For synchronous always blocks, we always use non-blocking assignments (<=), and *order does not* matter."

The blocking assignment will create and connect logic gates and wires, and the assignment *is* permanent; while "the non-blocking assignment results in a register."

//Quote from the lecture notes.

(b) (i).

ANS:



(ii).

ANS:

```
module Q4 (input D,
           input clk, R,
           output reg [3:0] st);

parameter s3=2'b0011, s4=2'b0100, s5=2'b0101, s6=2'b0110, s7=2'b0111,
      s8=2'b1000, s9=2'b1001, s10=2'b1010, s11=2'b1011;
reg [2:0] nst;

always @(posedge clk)
begin
      if (R)
            st <= s7;
      else
            st <= nst;

end

always @*
```

```
begin
      nst = st;   // make default setting
      case (st)
            s3:
                  if (D) nst = s3;
                  else if (D') nst = s4;
            s4:
                  if (D) nst = s3;
                  else if (D') nst = s5;
            s5:
                  if (D) nst = s4;
                  else if (D') nst = s6;
            s6:
                  if (D) nst = s5;
                  else if (D') nst = s7;
            s7:
                  if (D) nst = s6;
                  else if (D') nst = s8;
            s8:
                  if (D) nst = s7;
                  else if (D') nst = s9;
            s9:
                  if (D) nst = s8;
                  else if (D') nst = s10;
            s10:
                  if (D) nst = s9;
                  else if (D') nst = s11;
            s11:
                  if (D) nst = s10;
                  else if (D') nst = s11;
      endcase
end
endmodule
```

/*This is the most straightforward coding. Nevertheless, since it is the very last question in the exam, most people would probably just come up with solutions like this due to time limit haha.*/

4 (c).

The FSM starts at state A.

Inputs	1	1	0	0	0	1	0	0	0	1	1	
State		A	A	B	C	D	A	B	C	D	A	A
Output		0	0	0	0	1	0	0	0	1	0	0

ANS:

The output sequence should be 0,0,0,0,1,0,0,0,1,0,0.

All the best for your final exam~