CZ1005 Digital Logic AS1819 sem1

Name: Wu Sibing Email: WUSI0008@e.ntu.edu.sg

1.(a)

(i)

$$\begin{array}{r|l l}
16 & 98765 & 13 \\
16 & 6172 & 12 \\
16 & 385 & 1 \\
16 & 24 & 8 \\
 & 1 &
\end{array}$$

$\therefore 181CD_{16}$

(ii)

$$\begin{array}{r|l l}
16 & 98 & 2 \\
 & 6 &
\end{array}$$

$\therefore 98 = 62_{16}$

$0.765 \times 16 = 12.24$

$0.24 \times 16 = 3.84$

$\therefore C3_{16}$

$\therefore 62.C3_{16}$

(b) $3 = 11_2$ 2 digits

$2^{-m} < 0.141592$

$m > \log_{0.5} 0.141592 = 2.82$

$\therefore m \geq 3$ $\therefore$ need $2+3=5$ bits

(c)

{[(A'+C')(B'C'D)'][(A'+B)'+C']D'}'

$= (A'+C')(B'C'D)' + [(A'+B)'+C']' + D$

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

$= A'B + \underline{A'C} + A'D' + \underline{BC'} + C'D' + \underline{A'C} + \underline{BC} + D$

$= \underline{A'B} + A'C + A'D' + C'D' + \underline{B} + D$

$= A'C + A'D' + C'D' + B + D$

$= CA' + C'D' + A'D' + B + D$

$= CA' + \underline{C'D'} + B + \underline{D}$

$= \underline{CA'} + C' + B + D$ $\big\} (A + A'B = A + B)$

$= A' + B + C' + D$

(d)

$01010111 = 1\times2^6 + 1\times2^4 + 1\times2^2 + 1\times2^1 + 1\times2^0 = 87$

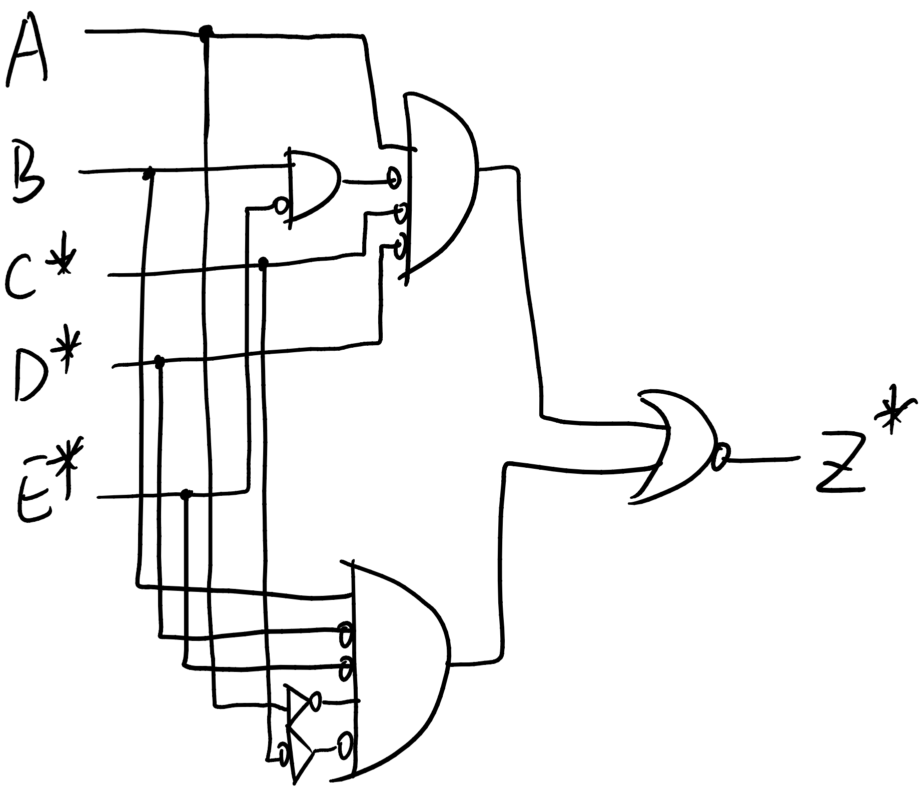
$11100110 = -(1\times2^6 + 1\times2^5 + 1\times2^2 + 1\times2^1) = -102$

$00110011 = 1\times2^5 + 1\times2^4 + 1\times2^1 + 1\times2^0 = 51$

$11101001 = -(00010111)$

$= -(1\times2^4 + 1\times2^2 + 1\times2^1 + 1\times2^0) = -23$

(e) $Z^* = [A' + C^* + D^* + B(E^*)'][B' + D^* + E^* + A + (C^*)']$



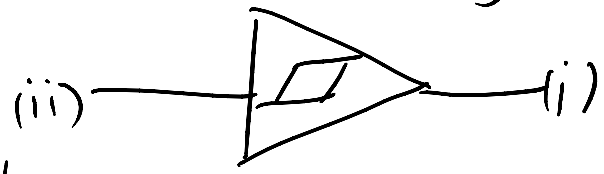
2.

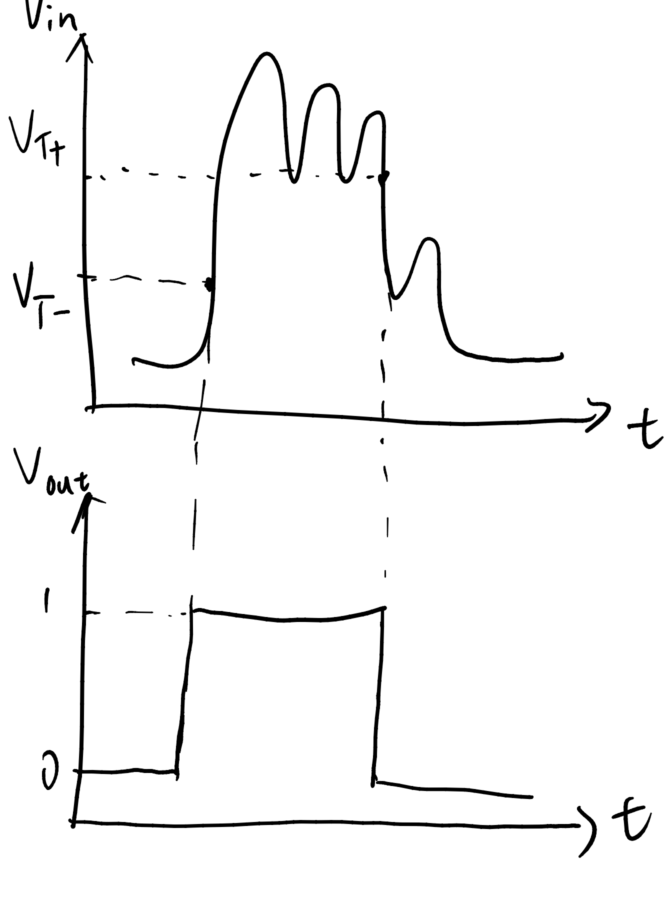
(a) -7: 0111 $\xrightarrow{\text{2's complement}}$ 1001

-6: 0110 $\xrightarrow{\text{2's complement}}$ 1010

```
      1001   → -7
    × 1010   → -6
 ─────────────
 00000000
 1111001
 000000
 0,0,1,1,1 1   → 2's complement
 ─────────────
 00101010   → 42
```

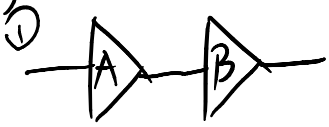
(b) use Schmitt-triger buffer





(c)

(i) ①

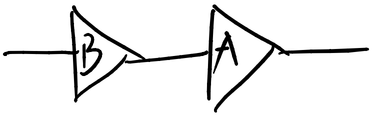


If VOH(min)(A) >= VIH(min)(B), VOL(max)(A) <= VIL(max)(B), IOH(max)(A)>= IIH(max)(B), IOL(max)(A) >= IIL(max)(B), then A drives B.

VOH(min)(A)=3.3V, VIH(min)(B)=3.5V, VOH(min)(A) < VIH(max)(B);

Hence A does not drive B.

②



If VOH(min)(B) >= VIH(min)(A), VOL(max)(B) <= VIL(max)(A), IOH(max)(B)>= IIH(max)(A), IOL(max)(B) >= IIL(max)(A), then B drives A.

VOH(min)(B)=4.5V, VIH(min)(A)=2.0V, VOH(min)(B) > VIH(max)(A);
VOL(max)(B)=0.6V, VIL(max)(A)=0.8V, VOL(max)(B) < VIL(max)(A);
IOH(max)(B)=19mA, IIH(max)(A)=0.6mA, IOH(max)(B)> IIH(max)(A);
IOL(max)(B)=21mA, IIL(max)(A)=0.8mA, IOL(max)(A) > IIL(max)(B);

Hence B drives A.

(ii)

$$\text{fan-out} = \min\left\{\frac{I_{OH}}{I_{IH}}, \frac{I_{OL}}{I_{IL}}\right\}$$

$$= \min\left\{\frac{19}{0.6}, \frac{21}{0.8}\right\}$$

$$= 26.25 = 26$$

Fan-out has to be an integer since it specifies the number of standard loads that the output gate can drive

(d) v = 0

c

ab	0	1
0 0	1	1
0 1	0	1
1 1	0	x
1 0	0	1

v = 1

c

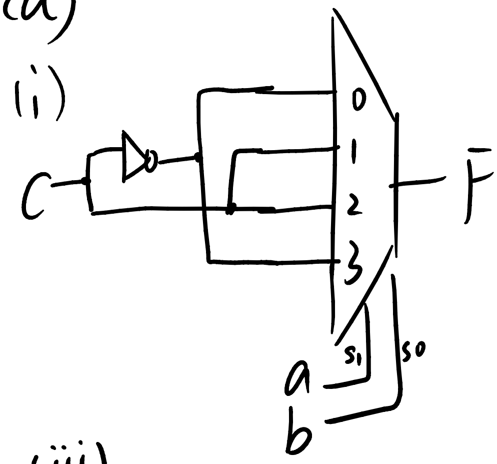
ab	0	1
0 0	1	0
0 1	0	0
1 1	0	X
1 0	1	1

$v'(a'b'+c)+v(b'c'+ac)$

$= v'a'b'+v'c+vb'c'+vac$

3(a)

(i)



(ii)

c

ab	0	1
0 0	1	0
0 1	0	1
1 1	1	0
1 0	0	1

$F = a'b'c'+ab'c+a'bc+abc'$

(iii)

```
module simplecircuit (input a,b,c, output F);
    assign F = (~a & ~b & ~c) | (a & ~b & c) | (~a & b & c) | (a & b & ~c);
endmodule
```

(b)

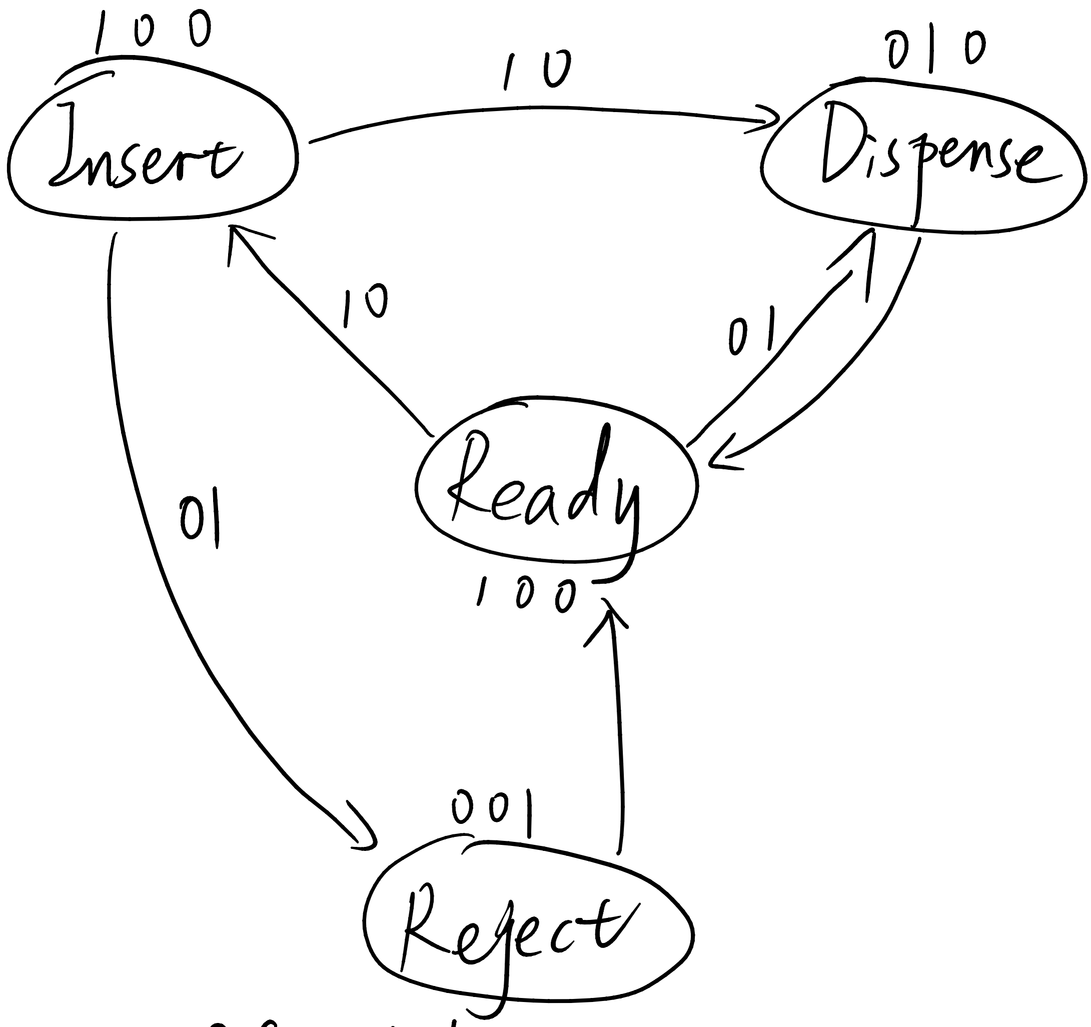
```
module decoder2_4 (input [1:0] Cin, output reg [3:0] Cout);
    always @ *
        case(Cin)
            2'b00 : Cout = 4'b0001;
            2'b01 : Cout = 4'b0010;
            2'b10 : Cout = 4'b0100;
            2'b11 : Cout = 4'b1000;
        endcase
endmodule
```

4. (a)

```
module simplecounter(#parameter WIDTH = 8)
(input clk, rst, up, output reg [WIDTH-1:0] q);
always@(posedge clk)
begin
    if(rst) q <= 8'b0000_0000;
    else
        if(up) q <= q +1'b1;
        else q <= q - 1'b1;
end
endmodule
```

(b)

(i)



input: fifty, dollar

output: insert_coin, dispense, reject_money

(ii)

```
module vendor (input clk, rst, fifty, dollar, output reg insert_coin, dispense,
reject_money);

parameter Ready = 2'b00, Insert = 1, Dispense = 2, Reject = 3;
reg [1:0] st, nst;

always@(posedge clk or negedge rst)
begin
    if(!rst)  st <= Ready;
    else  st <= nst;
end

always@* begin
nst = st;
{insert_coin, dispense, reject_money} = 3'b100;
case(st)
    Ready: begin
            if({fifty, dollar} == 2'b10)  nst = Insert;
            else if ({fifty, dollar} == 2'b01)  nst = Dispense;
            {insert_coin, dispense, reject_money} = 3'b100;
            end
    Insert: begin
            if({fifty, dollar} == 2'b01) nst = Reject;
            else if ({fifty, dollar} == 2'b10)  nst = Dispense;
            {insert_coin, dispense, reject_money} = 3'b100;
            end
    Dispense: begin
                nst = Ready;
                {insert_coin, dispense, reject_money} = 3'b010;
            end
    Reject: begin
                nst = Ready;
                {insert_coin, dispense, reject_money} = 3'b001;
            end
    Default: nst = Ready;
endcase
end
endmodule
```