· 数字电路与系统 〈组合逻辑〉 $A(B+C)+\bar{A}\bar{B}\bar{C}$ 11.7

· (逻辑1 $U \geq 2.7V$ 逻辑0: $U \leq 0.5V$) 正逻辑

反之即为反逻辑

· 数制

· 最低有效位 LSB，最高有效位 MSB

· 数制下标：D〈十进制〉，B〈二进制〉，H〈十六进制〉

〈先说数制码制〉

· 组合/时序逻辑区别：组合逻辑无记忆功能，输出只取决于当前输入，而时序要看原来与当前状态

· 例1：$Y = A(B+C)+\bar{A}\bar{B}\bar{C}$ 用与非门或者或非门实现

$= AB+AC+\overline{\bar{A}+\bar{B}+\bar{C}}$ $AB+AC = \overline{\overline{AB+AC}} = \overline{\overline{AB}\cdot\overline{AC}}$ 〈与非〉

$= \overline{\overline{AB}\cdot\overline{AC}}+\overline{\bar{A}+\bar{B}+\bar{C}}$

· 例2：$Z = \sum m(0,3,5,6,7)$ 〈与非〉

R \ AG	00	01	11	10
0	1	0	1	0
1	0	1	1	1

$\Rightarrow Z = \bar{R}\bar{A}G+AG+RG+RA$

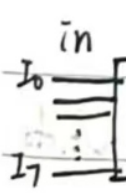
$= \overline{\bar{R}\bar{A}G+AG+RG+RA} = \overline{\overline{\bar{R}\bar{A}G}\cdot\overline{AG}\cdot\overline{RG}\cdot\overline{RA}}$

· 编码器（多位转低位） $M \to N$

N位二进制码可以表示 2^N 个信号，原则即为 $2^N \geq M$

如101位键盘，$2^7 \geq 101$，因此7位编码101位

·· 三位二进制编码器（八线-三线）（普通编码器）



注意C为高位，如 $I_1 \Rightarrow$ CBA: 001 $I_2 \Rightarrow$ CBA: 010

$C = I_4+I_5+I_6+I_7$ $A = I_1+I_3+I_5+I_7$

（任何时刻 I_i 只有一个有效） $B = I_2+I_3+I_6+I_7$

反演：$C = \overline{\bar{I}_4\cdot\bar{I}_5\cdot\bar{I}_6\cdot\bar{I}_7}$

·· 8-3线优先编码器

组合逻辑<编码器>

·8-3优先编码器<74148芯片>

verilog: input EIN, 7IN, 6IN ... 0IN ; //低电平有效

output GSN, EON, A2, A1, A0;

assign EON = EIN & (7IN到0IN不全1);

assign GSN = (工作且输入有效合规时为0);

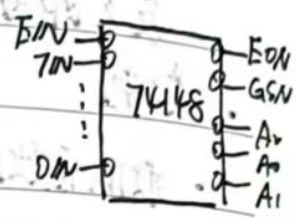
always@(*)

case({7IN, 6IN ... 0IN}) //低电平有效in/out, 反码输出

8'b0xxx_xxxx : {A2, A1, A0} = {0, 0, 0}; // 就是111反

8'b10xx_xxxx : {A2, A1, A0} = {0, 0, 1}; // 110反

...



·2-10优先编码器.

	D	C	B	A
I0	0	0	0	0
I1	0	0	0	1
I2	0	0	1	0
I3	0	0	1	1
I4	0	1	0	0
I5	0	1	0	1
I6	0	1	1	0
I7	0	1	1	1
I8	1	0	0	0
I9	1	0	0	1

D = I8 + I9

C = I4 + I5 + I6 + I7

B = I2 + I3 + I6 + I7

A = I1 + I3 + I5 + I7 + I9

·10-4线8421BCD编码器 (优先) <74147>

就是10输入, 转化4位BCD,

且均低电平有效.

如 I9 = 0, I8 ... I0 = X; DCBA = 0110 (1001反)

·二进制译码器.

输入n位, 情况2^n种, 即2^n位输出,

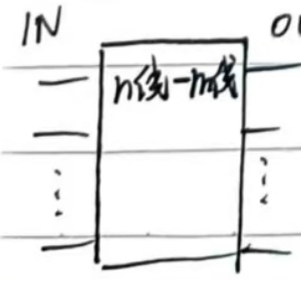
且2^n个输出位只有一位有效 ($2^n = m$)

·3位二进制译码器 (3→8)

$Y_0 = \bar{A_2}\bar{A_1}\bar{A_0}$ $Y_1 = \bar{A_2}\bar{A_1}A_0$ ... $Y_7 = A_2A_1A_0$

即 $Y_i = m_i$

··译码器. (少位数多位)



$m = 2^n$ 全译码器

$m < 2^n$ 部分译码器

期末复习 逻辑门逻辑芯片

4. 奇偶校验电路（大位，要求奇数个1时，Y=1，否则Y=0. <UESTC>

不适合卡诺图： $Y = A\oplus B\oplus C\oplus D\oplus E\oplus F$

如果 A,B,C=1，D,E,F=0， $A\oplus B=0$ $A\oplus B\oplus C=1$

$A\oplus B\oplus C\oplus D=1$ $A\oplus B\oplus C\oplus D\oplus E=1$ $A\oplus B\oplus C\oplus D\oplus E\oplus F=1$ 可得.

5. $F=\overline{AB+BC}+A\bar{C}$ 化为最简与或式： <TOUL>

$F = \overline{AB}\cdot\overline{BC}+A\bar{C} = (\bar{A}+\bar{B})(\bar{B}+\bar{C})+A\bar{C} = \bar{A}\bar{B}+\bar{A}\bar{C}+\bar{B}+\bar{B}\bar{C}+A\bar{C}$

$= \bar{B}+\bar{C}$

6. 芯片功能及引脚

①74595芯片（8位串行输入一并行输出移位寄存器）

功能描述：

Q_0~Q_7 并行输出引脚；SER 串行输入脚；SRCLK 移位寄存器时钟，RCLK 存储 reg 时钟，

OE 三态门控制，OE=1 输出 Z，OE=0 输出存储 reg 数据；MR 移位 reg 清空，

Q_{7S} 移位 reg 最高位；Vcc、GND

SRCLK↑，移位 reg 最低位接收 SER，即：

```
shift_reg <= {shift_reg[6:0], serial_in};
```

< shift_reg 在上升沿结束后得到更新?

RCLK↑，存储 reg 更新为 RCLK 上升沿触发时刻的数据，即：

```
q <= shift_reg;
```

<思考：当 RCLK、SRCLK 完全同步，q 在上升沿结束时更新为 shift_reg 上升沿前一瞬的寄存器值>.

时序更新细节. →

期末复习<FPGA>

9. Verilog HDL IEEE标准号，有哪两个版本？

IEEE std 1364，IEEE std 1364-1995和IEEE std 1364-2001.

10. Intel调试工具<都是FPGA和JTAG口通信>

① SignalTap II：占据大量存储单元作为数据缓存，单向收集FPGA信息，不能与FPGA双向对话。

② In-System Memory Content Editor：双向对话但是仅限存储器。

③ In-System Source and Probes Editor：双向对话，测试信号在内部引入测试系统，或由测试系统给为激励。

11. 速度优化方式<FPGA>

流水线设计、寄存器配平、乘法操作法、关键路径法 <时序驱动、分布配平...>

设计优化

12. 列举资源优化方法

资源共享、逻辑优化、串行化

13. 状态机FSM状态编码方式。

顺序编码，独热码，格雷码，约翰逊码

14. 以四个状态的FSM为例写具体编码：

顺序编码：S0=00，S1=01，S2=10，S3=11

一位热码：S0=0001，S1=0010，S2=0100，S3=1000

15. RV32I指令集指令类型：RISBUJ

① R（Reg-Reg） ① I（Imm立即数指令） ③ S（store reg→memory）

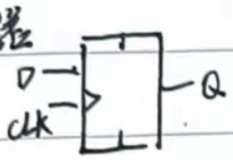
④ B（Branch分支） ⑤ U（Upper Imm） ⑥ J（Jump）

16. RISC-V处理器的寄存器x0的特殊性：等于0

期末复习（数电补充）

9. 单边沿触发器（cb）

① D触发器



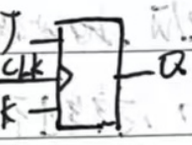
always @(posedge clk)
Q <= D;

② JK触发器

J=1, K=0, Qn+1=1
J=0, K=1, Qn+1=0
J=1, K=1, Qn+1=$\bar{Q}$
J=0, K=0, Qn+1=Q

$Q_{n+1} = J\bar{Q_n} + \bar{K}Q_n$

JK / Q_n	00	01	11	10
0	0	0	1	1
1	1	0	0	1



卡诺图得：$Q_{n+1} = Q_n\bar{K} + \bar{Q_n}J$

③ T触发器. $Q_{n+1} = T \oplus Q_n = T\bar{Q_n} + \bar{T}Q_n$.

T=0, $Q_{n+1}=Q_n$; T=1, $Q_{n+1}=\bar{Q_n}$

④ T'触发器：$Q_{n+1} = \bar{Q_n}$

⑤ JK→D （J=D, K=$\bar{D}$）$Q_{n+1} = D\bar{Q_n} + DQ_n = D$

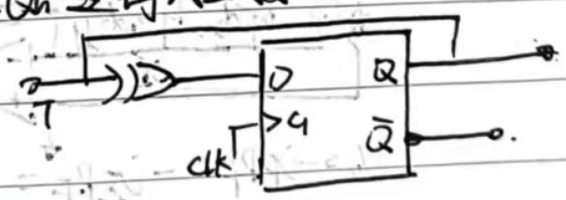
⑥ JK→T （J=T, K=T）$Q_{n+1} = \begin{cases} T=1 = \bar{Q_n} \\ T=0 = Q_n \end{cases}$

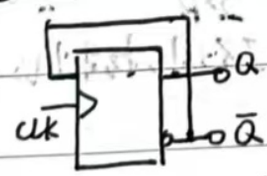
⑦ JK→T' （J=1, K=1）

⑧ D→JK 利用 $Q = J\bar{Q_n} + \bar{K}Q_n$ 逻辑式直接改造

⑨ D→T $D = T \oplus Q$

⑩ D→T' $D = \bar{Q}$





10. 存储器 cs

① 分类：ROM（Read-Only Memory），RAM（Random Access Memory）

<非易失性存储器>

例：ROM：PROM、EPROM、EEPROM、FLASH

RAM：SRAM（静态存储器）和动态存储器DRAM.

期末复习 <EDA verilog>

4. 设计异步复位、同步使能、同步清零、同步置位、同步装载 N=16位进制计数器。

```
module cnt16 #(parameter N=16) (
    input RST_A, EN, CLR, PRE, LOAD, CLK
    input [15:0] DATA,        // N-1=15
    output reg [15:0] Q);
    always @ (posedge CLK or negedge RST_A)
        if (!RST_A)
            Q<=0;
        else if (CLR) Q<=0;
        else if (PRE) Q<=16'hffff;
        else if (LOAD) Q<=DATA;
        else if (EN) { if (Q==16'hffff)
                         Q<=0;
                       else Q<=Q+1;
```

可以简化为 Q<=Q+1; ← Q自动复位。

```
end endmodule
```

5. Johnson计数器 (原理：最高位为0，补1左移；为1，补0左移)

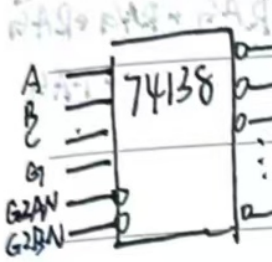
核心逻辑：

```
always @ (posedge clk or negedge rst)
    begin if (!rst) Q<=0
    else if (Q[N-1])
            Q<={Q[N-2:0],1'b0};
        else Q<={Q[N-2:0],1'b1};
    end
```

(N=4)

组合逻辑 <译码器> / <数据选择器>

· 3-8译码 <74LS138> LS低功耗肖特基



ABC三位二进制输入，C为最高位

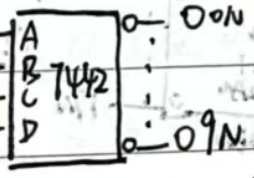
{Y0N ~ Y7N} 译码低电平有效 称为

G1=1, G2AN+G2BN=0 时译码，否则不译码

$\overline{Y_i} = m_i$ 即 $Y_i = \overline{m_i}$

· 二-十进制译码器

· 4线-10线译码器 <74LS42>

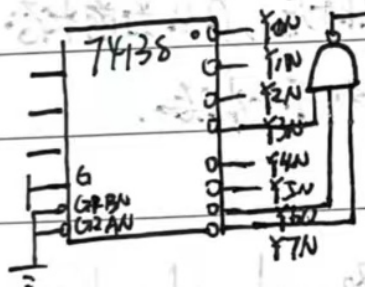


(低电平有效) $\overline{Y_i} = m_i$

· 例：F(A,B,C) = AB + BC 用74LS138和与非门实现

$F = AB\bar{C} + ABC + \bar{A}BC = \sum m(3,6,7) = m_3 + m_6 + m_7 = \overline{m_3 \overline{m_6 \overline{m_7}}}$

$= \overline{\overline{Y_3}\,\overline{Y_6}\,\overline{Y_7}}$



电路设计 ←

· 补充：7段数码管 a f g b e c dp 显示与即 a f g 时有效

<74LS47 共阳，74LS48 共阴>

·· 数据选择器

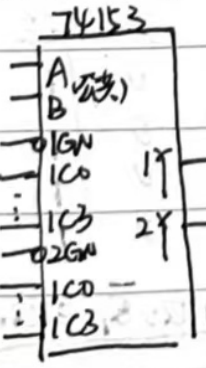
多至一数字开关 同选一真值表

IN			OUT
A_1	A_0	GN	Y
×	×	1	0
0	0	0	D_0
:	:	0	:
1	1	0	D_3

$Y = D_0\bar{A}_1\bar{A}_0 + D_1\bar{A}_1A_0 + D_2A_1\bar{A}_0 + D_3A_1A_0)\overline{GN}$

$= \overline{GN} \cdot \sum D_i m_i$

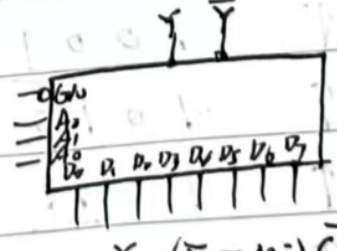
双四选一数据选择器 <74153>



AB选1C,2C

至1Y、2Y

八选一数据选择器 <74LS151>



{A2, A1, A0} 选 {D7 ~ D0} 至 Y

GN低电平使能

$Y = (\sum D_i M_i)\,\overline{GN}$

例：用74LS151实现 $F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB$

$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + AB\bar{C} = m_1 + m_2 + m_7 + m_6 \Rightarrow$ $\begin{cases} D_1, D_2, D_6, D_7 = 1 \\ D_0, D_4, D_5, D_3 = 0 \end{cases}$

组合逻辑 < 数据选择器 > / < 数据分配器 >

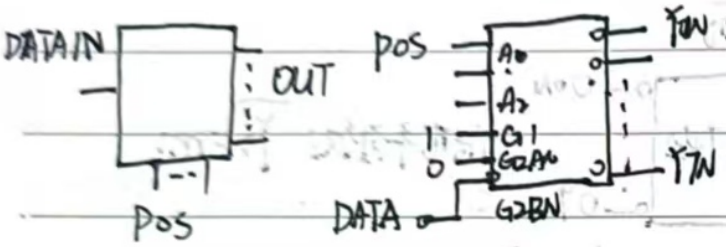
例2：用 74LS153（双四选一）实现 $Z=\bar{R}\bar{A}\bar{G}+\bar{R}A\bar{G}+R\bar{A}\bar{G}+RA\bar{G}+RAG$

$=\bar{R}\bar{A}\bar{G}+R\bar{A}\bar{G}+RA\bar{G}+1\cdot AG$

74153：$Y=D_0(\bar{A}_1\bar{A}_0)+D_1(\bar{A}_1A_0)+D_2(A_1\bar{A}_0)+D_3(A_1A_0)$

即 $D_0=\bar{R}$，$D_1=R$，$D_2=R$，$D_3=1$

· 数据分配器 74LS138译码器改数分配器。



$\{A_2,A_1,A_0\}$ 选择输出位，G2BN=0，被关位为0；G2BN=1 所有位均为1

· 4位二进制计数器 功能推导

当前状态 $Q_3\ Q_2\ Q_1\ Q_0$	CLK ↑	下一状态 $Q_3\ Q_2\ Q_1\ Q_0$
0 0 0 0	↑	0 0 0 1
0 0 0 1	↑	0 0 1 0
0 0 1 0	↑	0 0 1 1
0 0 1 1	↑	0 1 0 0
0 1 0 0	↑	0 1 0 1
0 1 0 1	↑	0 1 1 0
0 1 1 0	↑	0 1 1 1
0 1 1 1	↑	1 0 0 0
1 0 0 0	↑	1 0 0 1
1 0 0 1	↑	1 0 1 0
1 0 1 0	↑	1 0 1 1
1 0 1 1	↑	1 1 0 0
1 1 0 0	↑	1 1 0 1
1 1 0 1	↑	1 1 1 0
1 1 1 0	↑	1 1 1 1
1 1 1 1	↑	0 0 0 0

一个CLK个后，Q从当前状态更新到下一状态。

Q_0：

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$\therefore Q_0 \Leftarrow \bar{Q}_0$

Q_1：

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$\therefore Q_1 <= Q_1\bar{Q}_0+Q_0\bar{Q}_1=Q_1\oplus Q_0$

Q_2:

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	1	0	1
10	0	0	1	0

$Q_2 \Leftarrow Q_2\bar{Q}_1 + \bar{Q}_2Q_1Q_0 + Q_2Q_1\bar{Q}_0$

$= Q_2\overline{Q_1Q_0} + \bar{Q}_2Q_1Q_0$

$= Q_2 \oplus (Q_1 \cdot Q_0)$

Q_3

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	0	1
10	1	1	1	1

$Q_3 <= Q_3\overline{Q_1Q_0} + Q_3\bar{Q}_2Q_1Q_0 + \bar{Q}_3Q_2Q_1Q_0$

$= Q_3 \oplus (Q_2Q_1Q_0)$

Q_0:

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

$Q_0 <= \bar{Q}_0$

• BCD计数器功能推导

Q_3	Q_2	Q_1	Q_0	CLK	$Q_3Q_2Q_1Q_0$
0	0	0	0	↑	0 0 0 1
0	0	0	1	↑	0 0 1 0
0	0	1	0	↑	0 0 1 1
0	0	1	1	↑	0 1 0 0
0	1	0	0	↑	0 1 0 1
0	1	0	1	↑	0 1 1 0
0	1	1	0	↑	0 1 1 1
0	1	1	1	↑	1 0 0 0
1	0	0	0	↑	1 0 0 1
1	0	0	1	↑	0 0 0 0
1	0	1	0	↑	X X X X
1	0	1	1	↑	X X X X
1	1	0	0	↑	X X X X
1	1	0	1	↑	X X X X
1	1	1	0	↑	X X X X
1	1	1	1	↑	X X X X

Q_1:

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	X	X	X	X
10	0	0	X	X

$Q_1 <= Q_1\bar{Q}_0 + \bar{Q}_1Q_0\bar{Q}_3$

$= Q_1 \oplus (Q_0 \cdot \bar{Q}_3)$

Q_2:

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	X	X	X	X
10	0	0	X	X

$Q_2 \Leftarrow Q_2 \oplus (Q_1Q_0)$

Q_4:

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	1	0	X	X

$Q_3^{n+1} = \overline{Q_0}\,\overline{Q_1}\,Q_3 + Q_1Q_0Q_2$

$= (Q_2Q_1Q_0) \oplus \overline{Q_0}$

期末复习：

1. $(75.75)_D$ 转化二进制数：

$75 = 64 + 8 + 3 \Rightarrow (75)_D = (1001011)_B$. $(0.75)_H = \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^8} + \cdots$

$\therefore (75.75)_D \Rightarrow (1001011.11000001)_B$.

转化八进制数：$75 = 8^2 + 8 + 3$ $(75)_D = (113)_8$

$(0.75)_D = \frac{1}{8} \cdot 6 + \frac{1}{8^2} \cdot 3 = (0.603)_8$ $(75.75)_D \Rightarrow (113.603)_8$

转化十六进制数：$75 = 16^1 \cdot 4 + 9 \Rightarrow (4B)_H$ $(75.75)_D = (4B.C)_H$

$(0.75)_D = \frac{1}{16} \times 12 + \left(\frac{1}{16}\right)^2 \cdot 1 = (0.C1)$

8421BCD码：$(75.75)_D \Rightarrow 0111\_0101 . 0111\_0101\_0110$

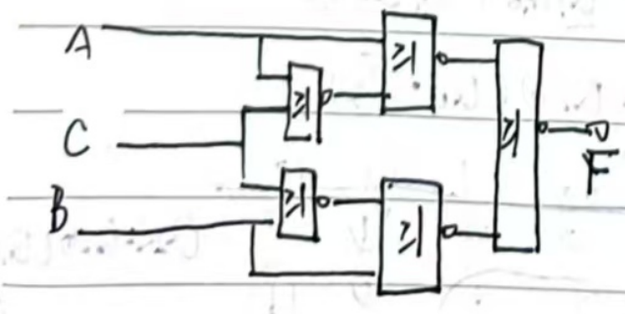
2. 真值为 $(-10001.00)_B$ 原码：110001 反码：101110

符号位 反数值不反符号

补码：101111（其正数取反加1）.

$(010001)_补 = 101110 + 1 = 101111$

3. 将或非门电路改为与非门 $\langle HHU \rangle$



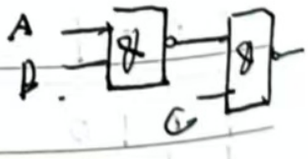
$F = \overline{A + (\overline{A+C}) + (B + \overline{(B+C)})}$

$= \overline{A + \overline{A}\overline{C} + B + \overline{B}\overline{C}}$

$= \overline{\overline{A} \cdot \overline{\overline{A}\overline{C}} + \overline{B} \cdot \overline{\overline{B}\overline{C}}}$

$= \overline{\overline{A}(A+C) + \overline{B}(B+C)}$

$= (\overline{A} + \overline{B}) \cdot C = \overline{AB} \cdot C$



期末复习 <数字例题>.

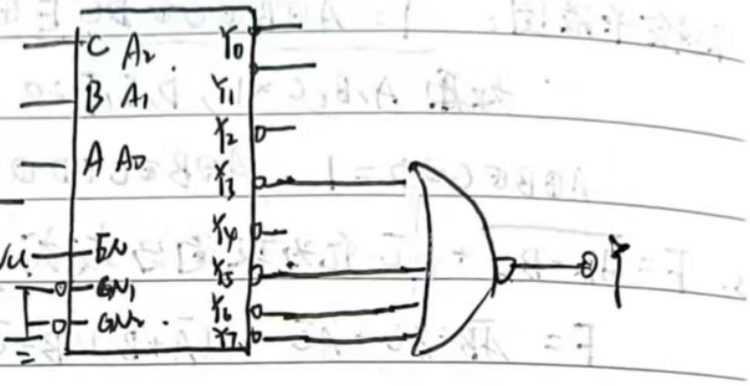
$AB = \overline{\overline{AB}} = \overline{\bar{A}+\bar{B}}$

三人表决电路（74138实现，两个同意即同意）.

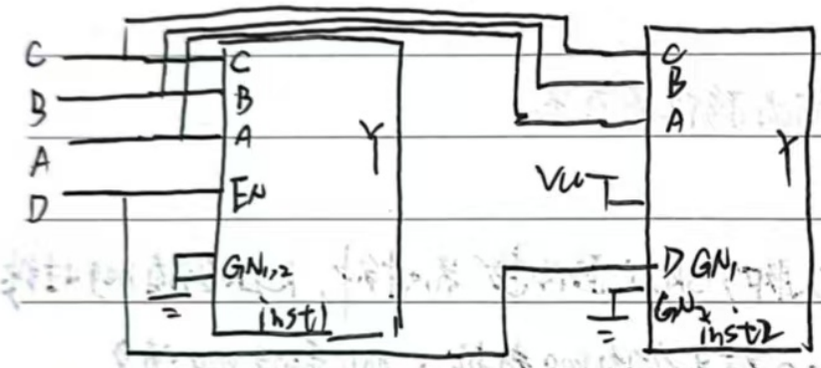
$Y = ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC$

$= \sum m_{(7,6,5,3)}$

$= Y_3 + Y_5 + Y_6 + Y_7 = \overline{\overline{Y_3 \cdot Y_5 \cdot Y_6 \cdot Y_7}}$



两个74138构成4-16译码器



当D=1，inst1工作，inst2不工作，inst1所有Y称为作为高8位；

D=0时，inst2作为低8位

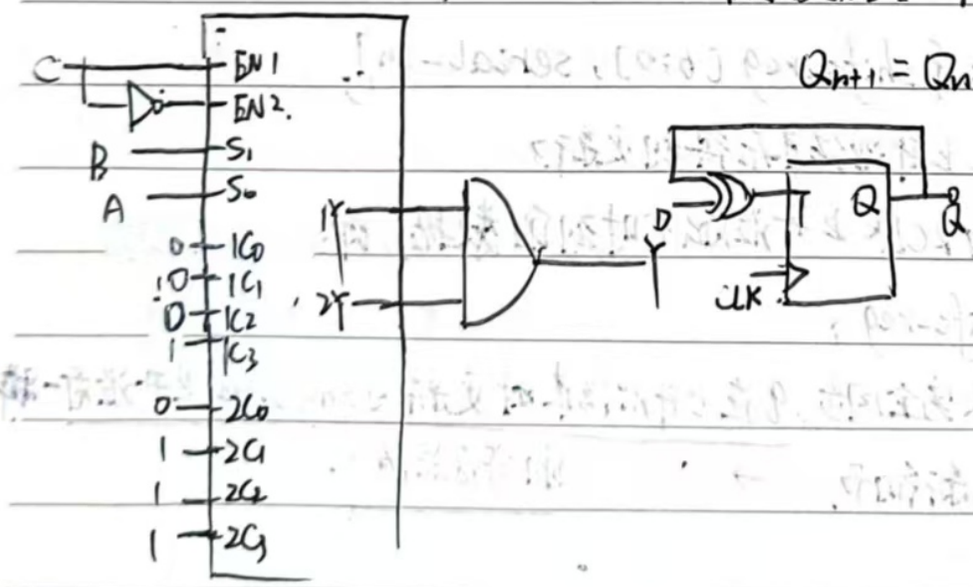
三人表决电路（74151双四选一实现）

· T触发器组成为DFF

T触发器：T=1，$Q^{n+1} = \bar{Q}^n$；T=0，$Q^{n+1} = Q^n$

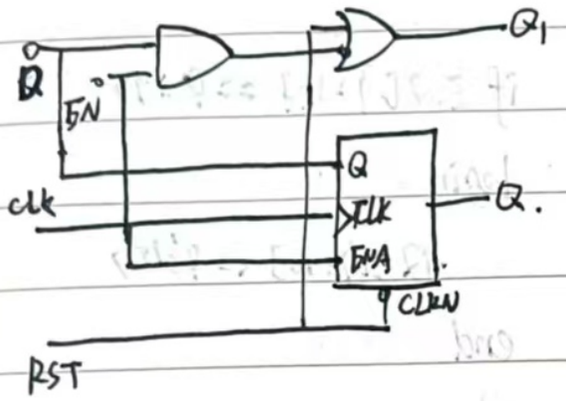
$Q^{n+1} = Q^n \oplus T$

Q为0，D=1，T=1即翻转；



RST == 0

RTL图为Verilog：




```
module top_module
(
    input D, EN, CLK, RST,
    output Q, Q1
);
reg Q;
assign Q = RST + ~(Q & EN);
always @ (negedge RST or posedge CLK)
begin
    if (!RST)
        Q <= 0;
    else
    begin
        if (ENA)
            Q <= D;
        else
            Q <= Q;
    end
end
endmodule
```

设计一个异步置位、同步使能、同步清零、同步置位、同步装载的N=16二进制计数器. parameter N=16 .

```
module top_module
(
    input [15:0] DATA,
    input CLK, EN, CLR_S, SET_S, LOAD_S,
    output Q,
    input RSTN_A
);
always @ (posedge CLK or negedge RSTN_A)
begin
    if(!RSTN_A)
        Q <= 0;
    else
        if (CLR_S)
            Q <= 0;
        else
            if (SET_S)
                Q <= 16'hffff;
            else
                if (LOAD_S)
                    Q <= DATA;
                else  Q <= Q + 1'b1;
    end
endmodule
```

// 开头应改为

```
module top_module
#( parameter N=16
)
(
    input ...
);
```

5位BCD计数器（异步复位，同步装载，同步使能）

```
module BCDcnt
(
    input RSTA, LOADS, ENS,
    input [19:0] DATA,
    output reg [19:0] Q
);
    always @(posedge CLK or negedge RST)
    begin
        if(!RSTA)
            Q<=0;
        else if(EN)
        begin
            if(LOAD)
            Q<=DATA;
            else
            begin
                if(Q[3:0]==4'd9)
                    begin
                    Q[3:0]<=4'd0;
                    if(Q[7:4]==4'd9)
                    begin
                    Q[7:4]<=4'd0;
                    if(Q[11:8]==4'd9)
                    begin
                    Q[11:8]<=4'd0;
                    if(Q[15:12]==4'd9)
                    begin
                    Q[15:12]<=4'd0;
                    if(Q[19:16]==4'd9)
                    begin
                    Q[19:16]<=4'd0;
                    end
                    else
                    Q[19:16]<=Q[19:16]+1;
                    end
                    else
                    Q[15:12]<=Q[15:12]+1;
                    end
                    else
                    Q[11:8]<=Q[11:8]+1;
                    end
                    else
                    Q[7:4]<=Q[7:4]+1;
                    end
                    else
                    Q[3:0]<=Q[3:0]+1;
                    end
                    end
                    end
                    endmodule
```

期末复习<EDA>

1. EDA：电子设计自动化　　<侧重逻辑正确性>

2. EDA设计流程：设计输入→功能仿真→综合→适配→时序仿真
→下载→硬件测试　　<侧重信号延迟、时序约束、竞争与冒险>

3. IP全称：知识产权核
IP分类：① 软IP：Verilog硬件描述能module块
② 固IP：完成了综合的功能块（门级网表）
Verilog netlist：提供基本单元间的连接关系/延迟信息
③ 硬IP：提供设计版图掩膜（版图）

4. PLD：可编程逻辑器件　　（固定AND+编程OR阵列，熔丝技术）
简单PLD：PROM（可编程只读存储器，只能烧写一次）
*EPROM（可擦除PROM）<电烧写，紫外光擦除>
*EEPROM（电可擦除PROM）<电烧写与擦除>
PLA（可编程逻辑阵列，可编程的与阵列+或阵列）
PAL（可编程阵列逻辑，可编程与阵列+固定或阵列）
升级↓
GAL（可重复编程PAL）
基本编程原理：乘积项逻辑（与或阵列）可编程结构
数电第四章

5. 复杂PLD：CPLD和FPGA
CPLD基于乘积项可编程结构，FPGA基于SRAM查找表逻辑可编程结构

6. Intel的嵌入式逻辑分析仪？由什么技术构建？
SignalTap II，由JTAG技术
<监控FPGA内信号>

7. Quartus中应用JTAG技术的工具：
SignalTap II，In-System Source and Probes，In-System Memory Content Editor.

8. JTAG标准：IEEE 1149　　(Test In)
JTAG信号：TMS，TCK，TDO，TDI，TRST　　<SPI相似>
(Mode Select)　(Data Out)

期末复习 <EDA: verilog>

1. 3>位ALU，加减、与、或运算.

补充：逻辑右移 >>

如：assign a=b>>2;

算术右移 >>>：

如：assign a=b>>>2;

```
module alu
#parameter(ADD=0,SUB=1,AND=2,OR=3)
(input [1:0] ALUOP,
 input [31:0] alua, alub,
 output reg [31:0] alur);
alway @(*)
begin
    case(aluop)
        ADD: alur = alua + alub;
        SUB: alur = alua - alub;
        AND: alur = alua & alub;
        OR : alur = alua | alub;
        defaule: alur = alua + alub;
    endcase.
end
endmodule.
```

2. 时钟24小时制. $2^7=128$ $2^6=64$

```
module clock60(
 input clk_50m, RSTN_A
output [5:0] hour, min, sec);
// 分频至一秒. 25,000,000 < 2^25 > 25M.
                50,000,000 -> 2^26
reg [25:0] cnt;
reg clk_1;
always @(posedge clk_50M or negedge
                                RSTN_A)
begin.
  if(!RSTN_A)
    cnt<=0;
  else if(cnt==26'd50_000_000)
    cnt<=0;
    clk_1<=~clk_1;
  else cnt<=cnt+1;
end
assign clk_1=(cnt>25000000)?1:0;
// 时钟复位
always @(posedge clk_1 or negedge RSTN_A)
begin if(!RSTN_A) cnt<=0;
  else if(sec==9)
begin①
    sec<=0;
```

```
if(min==59)
begin②
    min<=0;
if(hour==23)
    hour<=0;
else
    hour<=hour+1;
end②
else
    min<=min+1;
end①
else
    sec<=sec+1;
```

毛利率

期末复习〈数电补充〉

2. 芯片74161引脚及其功能，<4位二进制计数器>

in: MR = RSTN_A ; CP = CLK_posedge ; CET、CEP = EN ; D0~D3 : [3:0] DATA_IN

PE = LOADN_S ; TC = COUT 达到1111时置1

OUT: Q0~Q3 :

核心逻辑：

```
always @(posedge clk or negedge RSTN_A)
    begin if(!RSTN_A) begin
            q<=4'b0000;
        else if (!PE) begin
            q<=d;
        end else if (cet && cep) begin
            q<=q+1;
        end
    end
    assign tc = (q==4'b1111&& cet && cep)? 1:0;
```

3. 芯片74153引脚及其功能.〈两个四选一多路选择器〉

引脚 in: [3:0] I0（第一组多路选择器输入）; [3:0] I1

选择信号：S0; S1　　使能选择器 = E1, E2

out: 选择器输出 Y0, Y1

核心逻辑：// 第一组多路选择器逻辑

```
always @(*)
begin
    if (E1) begin
        Y0 = 1'b0;
    else begin
        case({S1,S0})
        2'b00: Y0 = I0[0];
        2'b01: Y0 = I0[1];
        2'b10: Y0 = I0[2];
        2'b11: Y0 = I0[3];
        default: Y0 = 1'b0;
        endcase
    end
end.
// 第二组同理，只是E1换成E2
```

6. 74151芯片引脚及功能.<8选1多路选择器>

引脚: I_n = ① I1~I7, 数据输入 ② 选择信号, S0, S1, S2

③ $\overline{E}$ 使能(低有效)

OUT: ① Y 数据输出(取反) ② W = $\overline{Y}$

核心逻辑:

```
always @(*) begin
    if(E) begin
        Y=1'b1;
        W=1'b0;
    end else begin
        case(S) //S2,S1,S0
        3'b000:{Y,W}={~I[0],I[0]};
        :
        3'b111:{Y,W}={~I[7],I[7]};
        endcase end
```

5. 74138芯片引脚及其功能. (高有效) (低有效)

引脚: I_n: ① 输入信号: C、B、A (高位→低位) ② 使能端: G2A、G2B、G1

OUT: Y7~Y0 八位输出(取反), 低位有效

核心逻辑:

```
always @(*) begin
    if(!G1 && G2A && G2B) begin
        case({C,B,A})
        3'b000: Y=8'b1111_1110;
        3'b111: Y=8'b0111_1111;
        default: Y=8'b1111_1111;
        endcase
    end else Y=8'b1111_1111;
end
```

期末复习<数电补充>

*6、码制补充：(CH)

① 奇偶校验码：偶校验：数据位尾加检验位(0、1)，

使得 数据中"1"的数量为偶数

② 格雷码：相邻两个数只有一位不同

③ ASCII码：A → 41H，65D　a → 97D，61H

0 → 48D，30H　空格 → 32D，20H

*7. 处理器 ISA(指令集架构)(CH)

组成：① 操作指令(ADD、SUB等)

② 寄存器集(定义处理器中可用的寄存器)

③ 数据类型(整数、浮点数)

④ 寻址模式(访问内存或寄存器数据)

⑤ 异常/中断机制

分类：① CISC(复杂指令集计算机) x86, VAX　② RISC(精简指令集计算机) ARM, RISC-V

机器码<ISA指令的二进制形式，处理器执行的二进制指令)

① 操作码：add、MOV等，具体操作

② 操作数：操作源数据或目的数据

例：add x1, x2, x3 (RISC-V)

8. 常输出高阻的特殊逻辑门 (C2)

① 传输门：控制信号为高电平时 → 传输门导通

低电平 → 输出高阻态

② 三态门：三种状态的逻辑门，EN=1，导通0/1输出；EN=0，输出高阻态

③ OD门(开漏门)：输出级仅包含一个下拉NMOS管的结构

NMOS导通：输出被拉至低电平

NMOS截止：输出高阻态，可由外部电阻拉至高电平

10. ① 存储容量：字数×字长。

存储指标、存储速度：存储时间、存储周期

③ 存储器的扩展

a. 位扩展（特点：地址线共用）如：两片 8KB×8位 → 8KB×16位

方法：地址线、读写控制线 R/W、片选线 $\overline{CS}$ 共用，数据线并行。

b. 字扩展（特点：地址线扩展）如：两片 8KB×8位 → 16KB×8位

方法：数据线、读写控制线 R/W、地址线并联，

高位地址译码后产生不同状态，控制不同的 $\overline{CS}$

c. 字位扩展。

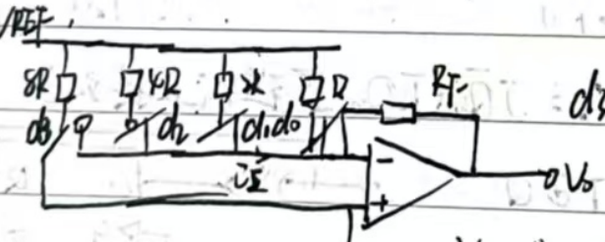
*11. AD/DA 转换器。

① D/A 转换工作原理。$D = d_{n-1}2^{n-1} + \cdots + d_1 2^1 + d_0 2^0$

（二进制D,位权展开）

转换量：$V_o = k_u \cdot D$（k_u 即为系数）。

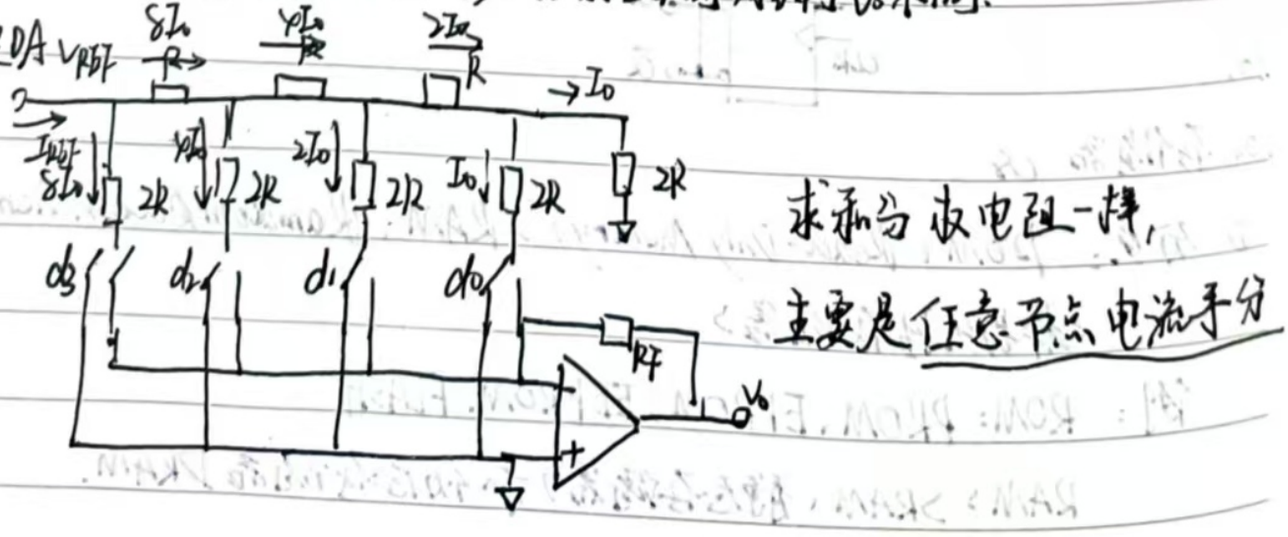
② 权电阻型 D/A：



$$V_o = -i_F R_F = -i_\Sigma R_F = \frac{-V_{REF} \cdot R_F}{2^3 R}(2^3 d_3 + 2^2 d_2 + 2^1 d_1 + 2^0 d_0)$$

根据 $d_3 \sim d_0$，$i_3 \sim i_0$ 的I不同，控制 V_o 不同。

③ 倒T型 D/A



求和为权电阻一样，

主要是任意节点电流守分

期末复习（数电补充）

④ DA技术指标：

分辨率 = $\frac{V_{LSB}}{V_{max}} = \frac{1}{2^n-1}$　V_{LSB}为最低有效位电压输出；V_{max}最大输出电压

转换误差：< $\frac{\pm V_{LSB}}{2}$；转换速度；温度系数。

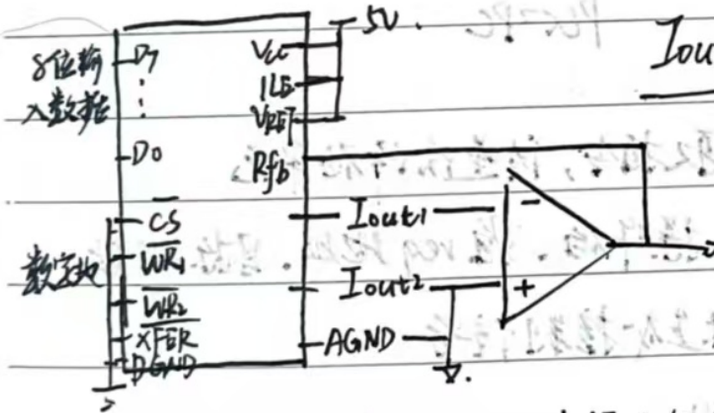
⑤ AD转换器

工作过程：采样，保持（互斥），量化，编码

技术指标：转换精度 → 分辨率 = $\frac{V_{REF}}{2^n}$

转换速度

⑥ DAC0832V应用电路 <输出电流DAC → 运放转化输出负电压>

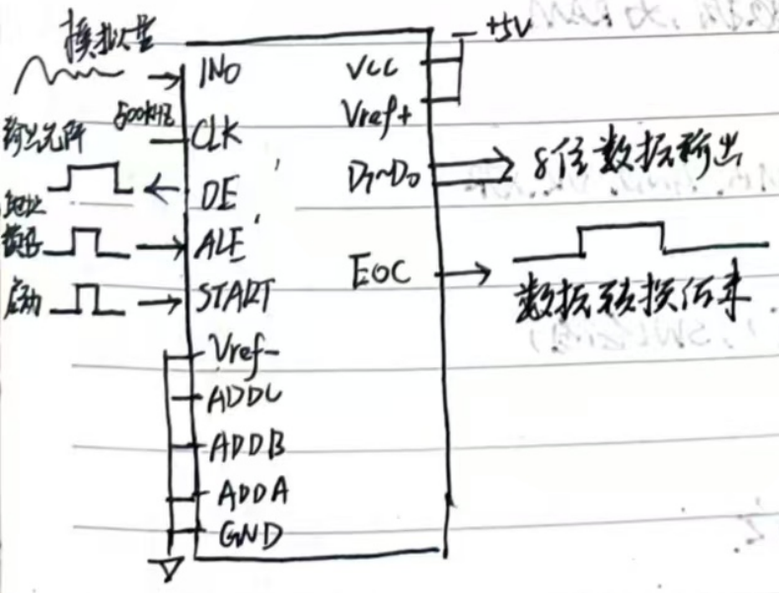


$I_{out1} = \frac{V_{REF}}{R} \cdot \frac{D_{(0)}}{256}$　$I_{out2} = \frac{V_{REF}}{R} \times \frac{256 - D_{(0)}}{256}$

$V_o = -(I_{out1} \times R_{fb})$

<反相放大器>

⑦ ADC0809应用电路 <选择IN0（通道1）>



步骤：① IN0模拟量输入，

ALE脉冲锁存ADDC~ADDA地址信号

② 启动START产生上升脉冲，

启动A/D转换

③ 转换完成后，EOC输出高电平

OE

*12. RISC处理器模型机

① 主要构成：

任务：
a. ALU：功能：ADD、SUB、AND、OR、XOR、MUL（算术右移 >>>，逻辑左移 <<，逻辑右移 >>）
b. 寄存器组、读取寄存器值/写入寄存器
b. 程序计数器 PC：（支持异步复位）

PC功能	控制信号 EN, LOAD, SEL			操作
指向下一条指令	1	0	X	PC<=PC+1
绝对地址加偏移量跳转	1	1	0	PC<=PC_base+PC_offset
相对地址跳转	1	1	1	PC<=PC+PC_base+PC_offset
等待 (default)	0	X	X	PC<=PC

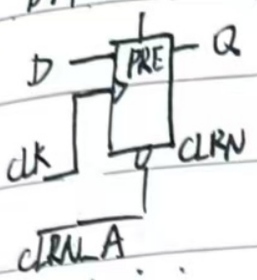
d. 取指单元：从指令寄存器读取指令，传递给译码单元
e. 译码单元：二进制指令拆成 操作码、源reg地址、目标reg址
f. 执行控制单元：由译码结果生成控制信号
g. 指令存储器：ROM或FLASH
h. 数据存储器：存储运行数据，为RAM.

② 指令集 ISA
a. 逻辑/运算指令：ADD、SUB、AND、OR、XOR
b. 分支指令：BEQ、BNE
c. 数据取存指令：LW（加载），SW（存储）
d. 无条件跳转：JUMP

RISC-I指令集：RISBUJ指令

期末复习 <EDA Verilog>

1. DFF注意 (RTL→Verilog)

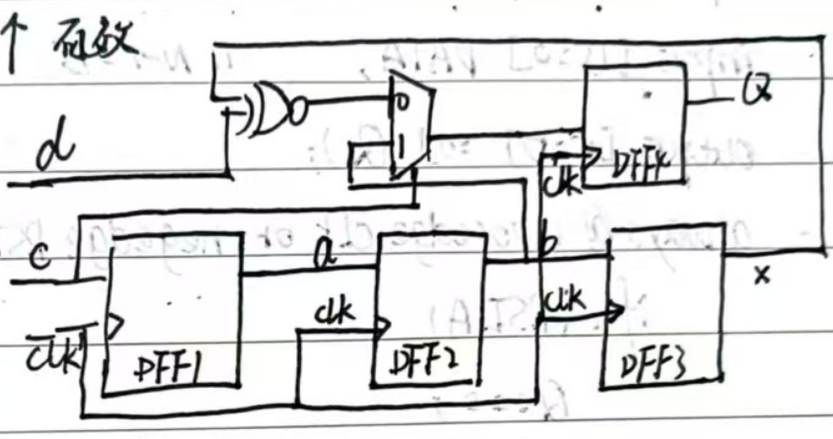


这里的 CLRN 通0位，无说明即为异步置0位

PRN 通1位，也为异步置1位

CLK↑有效

2. Verilog→RTL 例1

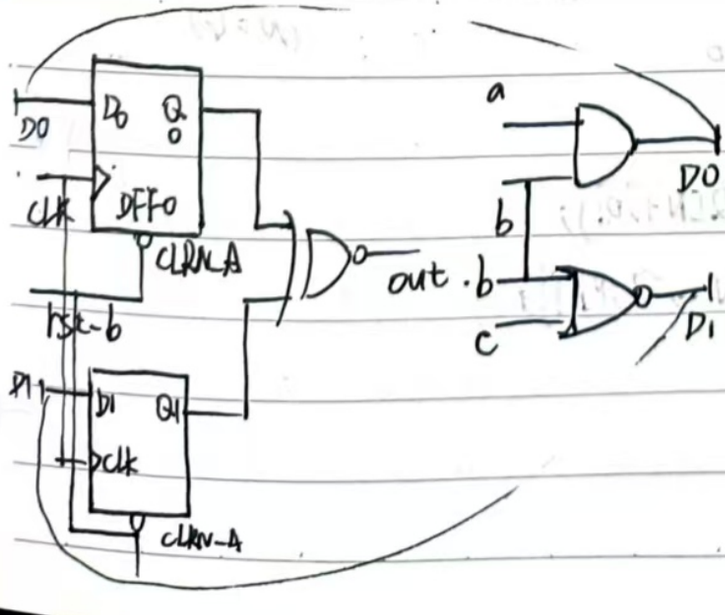



```
module test(clk,d,q,c);
  input clk,c,d;
  output q;
  reg a,b,x,q;
always @(posedge clk)
begin a<=d; b<=a; x<=b; end
alway @(c,b,x,d)
begin if(c) q<=b; //同或
      else q<=x~^d; end
endmodule
```

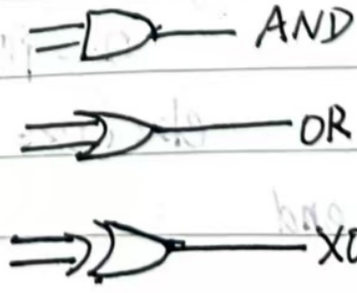
```
module Test2(
  input a,b,c,clk,rst_b,
  output out);
  reg [1:0] q;
  assign out = q[1] ~^ q[0];
always @(posedge clk, negedge rst_b)
    if(!rst_b) q<=0;
    else q<={a&b, ~(b|c)};
endmodule
```

3. Verilog→RTL 例2

RTL:



基本RTL注意：



期末复习 <EDA verilog>

1010 + 0110 = 10001

6. 乘法累加器：R = R + A * B

<P46>

模板：always @ (posedge clk or negedge rstn)

if (!rstn) R <= 0; else R <= R + A * B; // 打一拍无异

7. 8选一多路选择器：assign y = a[s]; // y输出，[7:0] a数据，[2:0] s选择

8. 1→8 数据分配器：assign y = a << s; // y [7:0] out，a待分配，[2:0] 分配位

9. 4-16 译码器：assign temp_y = 1'b1 << a; // 1为按位控制，

assign y = en ? temp_y : ~temp_y; 4位转16位独热码

10. BCD码 ① 规则：H → BCD { H<10 : BCD = H; H>=10 : BCD = H + 6; P59

② 两位BCD码加法器：<组合逻辑> // DTO 逻辑

```
module BCDadder2(
  input [7:0] A, B,
  output reg [8:0] D);
wire [4:0] DTO, DT1; reg S;
assign DTO = A[3:0] + B[3:0];
assign DT1 = A[7:4] + B[7:4] + S;
```

```
always @ (*) begin
  if (DTO >= 4'd10)
  begin D[3:0] = (DTO[3:0] + 4'd6);
    S = 1; end
  else begin D[3:0] = DTO[3:0]; S = 0; end
always @ (*) begin // DT1 逻辑
  if (DT1 >= 4'd10)
  begin D[7:4] = (DT1[3:0] + 4'd6);
    D[8] = 1; end
  else begin D[7:4] = DT1[7:4];
    D[8] = 0; end end
```

③ 4位BCD计数器

```
module BCDcnt4(
  input clk, reset_n, en, load,
  input [15:0] d,
  output reg [15:0] bcd);
always @ (posedge clk or negedge rst_n)
  if (!reset_n) bcd <= 0;
  else (load) bcd <= d;
```

后页

10. ④ BCD计数器(例化)

```
else if (en)
    if (bcd[3:0]<9) bcd[3:0] <= bcd[3:0]+1;
else if (bcd[7:4]<9) bcd[7:4] <= bcd[7:4]+1; bcd[3:0] <=0; end
else if (bcd[11:8]<9) bcd[11:8] <= bcd[11:8]+1; bcd[7:0]<=0; end
else if (bcd[15:12]<9) bcd[15:12] <= bcd[15:12]+1; bcd[11:0]<=0; end
else bcd<=0; end endmodule
```

⑧ 模60的 BCD加法计数器.

```
module BCDcnt60
(   input [7:0] data,
    input load, cin, clk, reset,
    output reg [7:0] qout,
    output cout );
always @ (posedge clk) begin
    if (reset) qout<=0;
    else if (load) qout<=data;
    else if (cin) begin
        if (qout[3:0]==9)
            begin qout[3:0]<=0;
                if (qout[7:4]==5)
                    qout[7:4]<=0;
                else qout[7:4] <= qout[7:4]+1;
            end else qout[3:0]<=q[3:0]+1;
    end
```

↓ 改能.

⑤ 5位 BCD计数器

```
module bcdcnt
(   input clk, rstn, load, en,
    input [19:0] data,
    output [19:0] q,
    output cout );
reg [3:0] q4, q3, q2, q1, q0;
wire c4, c3, c2, c1, c0;
assign q={q4, q3, q2, q1, q0};
always @ (posedge clk or negedge rstn)
    if (!rstn) q0<=0;
    else if (load) q0<=data[3:0];
    else if (en)
        if (q0<9) q0<=q0+1;
        else q0<=0;
assign c0=(q0==9);
..... <通过 en 和 c0 实现连述, 共5位>
assign cout=(q==20'h99999);
endmodule
```

mo10

log2运算，向上取整，计算位数。

timescale

$clog2(N)

⑥ BCD计数器 Test Bench

```
`timescale 1ns/100ps
module bcdcnt_tb;
reg clk, rstn, load, en;
reg [19:0] data;
wire [19:0] q;
wire cs;          // 实例化
bcdcnt u1
(   .clk(clk),
    .rstn(rstn),
    .load(load),
    .en(en),
    .q(q),
    .cout(cs));   // clk激励
initial begin
    clk=1'b0; forever #5 clk=~clk;
end
```

// 其余信号激励

```
initial begin
    rstn=0; load=0; en=0;
    data=20'h12345;
    #15 rstn=1'b1;
    #10 data=20'h23456;
    #10 load=1'b1;
    #10 load=1'b0;
    #10 en=1'b1;
    #1000000 en=1'b0;
    #10 data=20'h22222;
    #10 load=1;
    #10 load=0;
    #10 en=1'b1;
    #100000000 $stop;
end
endmodule
```

另延时5时间单位，再rstn置1。

load测试

计数测试

再load测试

计数测试