

1 基本门电路

笔试/面试考点：逻辑门，完备集，最小项的概念和性质，逻辑化简

1.1 二值逻辑运算与布尔逻辑运算

二值逻辑是最简单的逻辑，具有两个值：真和假，多个二值逻辑之间可以通过“ \wedge ”、“ \vee ”、“ \neg ”，即“与”、“或”、“非”进行运算。

将二值逻辑的真、假用逻辑常量1、0来表示，将二值逻辑运算符用“ \cdot ”、“ $+$ ”、“ $-$ ”来表示，就构成了布尔逻辑运算。将“与”、“或”、“非”三种基本运算进行结合，可以构造出复合运算：“与非”、“或非”、“与或非”、“异或”、“同或”，其中“异或”和“同或”运算可用“ \oplus ”和“ \odot ”符号表示。这几种逻辑运算的真值表如下。

A	B	$A \cdot B$	$A + B$	$A \oplus B$	$A \odot B$
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	1	0	1

同时可得，这几种逻辑运算之间的关系为： $A \oplus B = \overline{A}B + A\overline{B}$ ， $A \odot B = \overline{\overline{A}B + A\overline{B}} = \overline{A}\overline{B} + AB$ ， $A \oplus B = \sim(A \odot B)$ 。

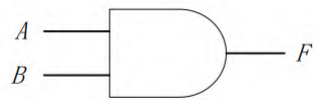
1.2 基本逻辑门

数字电路中的数据处理可以采用布尔逻辑运算来实现，任何逻辑函数都可以由基本逻辑运算组合实现。实现布尔逻辑运算的基本电路称为逻辑门，逻辑门可采用代数式（逻辑表达式）、真值表、逻辑图（逻辑电路）或波形图等方式表示，在逻辑门表示中，采用电平来表示逻辑变量，即用高电平来表示逻辑1、用低电平来表示逻辑0。

1. 与门AND：

实现与运算的电路称为与门。与运算的特点是，只有所有输入都为高电平时，输出才为高电平；只要有输入是低电平，输出就是低电平。与门的代数式为 $F = AB$ ，真值表与逻辑图如下：

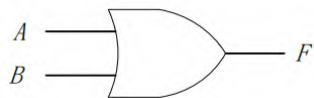
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1



2. 或门OR:

实现或运算的电路称为或门。或运算的特点是，只要有输入为高电平时，输出就为高电平；只有所有输入都为低电平，输出才为低电平。或门的代数式为 $F = A + B$ ，真值表与逻辑图如下：

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



3. 非门NOT:

实现非运算的电路称为非门。非运算的特点是，输出电平与输入电平相反，也称为反相器。或门的代数式为 $F = \overline{A}$ ，真值表与逻辑图如下：

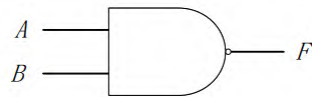
A	F
0	1
1	0



4. 与非门NAND:

实现与非运算的电路称为与非门。与非运算的特点是，只有所有输入都为高电平时，输出才为低电平；只要有输入是低电平，输出就是高电平。与非门的代数式为 $F = \overline{AB}$ ，真值表与逻辑图如下：

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0



5. 或非门NOR:

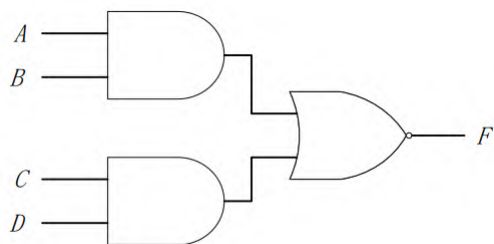
实现或非运算的电路称为或非门。或非运算的特点是，只要有输入为高电平时，输出就为低电平；只有所有输入都为低电平，输出才为高电平。或门的代数式为 $F = \overline{A+B}$ ，真值表与逻辑图如下：

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



6. 与或非门:

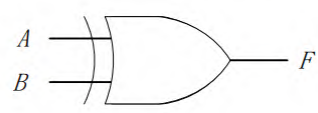
实现与或非运算的电路称为与或非门。与或门的代数式为 $F = \overline{AB+CD}$ ，逻辑图如下：



7. 异或门XOR:

实现异或运算的电路称为异或门。异或运算的特点是，如果有奇数个输入为高电平时，则输出为高电平。异或门的代数式为 $F = A \oplus B$ ，真值表与逻辑图如下：

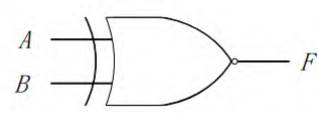
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0



8. 同或门XNOR：

实现同或算的电路称为同或门。同或运算的特点是，如果有奇数个输入为高电平时，则输出为低电平。同或门的代数式为 $F = A \odot B$ ，真值表与逻辑图如下：

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1



9. Verilog中的逻辑门：

逻辑门	Verilog表达式
与门	<pre>assign F = A[N:0] && B[N:0]; assign F[N:0] = A[N:0] & B[N:0]; assign F = &A[N:0];</pre>
或门	<pre>assign F = A[N:0] B[N:0]; assign F[N:0] = A[N:0] B[N:0]; assign F = A[N:0];</pre>
非门	<pre>assign F = !A[N:0]; assign F[N:0] = ~A[N:0];</pre>
异或门	<pre>assign F[N:0] = A[N:0] ^ B[N:0]; assign F = ^A[N:0];</pre>
同或门	<pre>assign F[N:0] = ~(A[N:0] ^ B[N:0]); assign F = ~(^A[N:0]);</pre>

1.3 逻辑运算的基本定律（自学）

名称	公式1	公式2
0-1律	$A + 1 = 1$	$A \cdot 0 = 0$
自等律	$A + 0 = A$	$A \cdot 1 = A$
重叠律	$A + A = A$	$A \cdot A = A$
互补律	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
交换律	$A + B = B + A$	$A \cdot B = B \cdot A$
结合律	$(A + B) + C = A + (B + C)$	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
* 分配律	$(A + B)(A + C) = A + BC$	$A \cdot (B + C) = AB + AC$
* 反演律	$\overline{A+B} = \overline{A} \cdot \overline{B}$	$\overline{AB} = \overline{A} + \overline{B}$

1.4 完备集、最小项与逻辑化简（自学）

1. 完备集：

对代数系统而言，若仅用它所定义的一组运算符号就能解决（表示）所有运算问题，则称这一组符号是一个完备集。在逻辑代数中，“与、或、非”逻辑运算符是一组完备集，它们能表示所有逻辑运算。同理，“与非”、“或非”、“与或非”分别是三组完备集。

2. 逻辑式的常用表达形式：

利用反演律，可以将任意逻辑代数式转化为以下形式：与或式、与非-与非式、或与非式、与或非式、或与式、或非-或非式。由于与非、或非都是完备集，因此对于与非-与非式和或非-或非式，采用单一种类的逻辑门就可以实现所有逻辑电路。

3. 最小项：

最小项的定义： n 个变量的最小项是 n 个变量的与项，其中每个变量都以原变量或反变量的形式出现一次。例如逻辑变量 A 、 B 构成的最小项为： $\overline{A}\overline{B}$ 、 $\overline{A}B$ 、 $A\overline{B}$ 、 AB ，它们分别与00、01、10、11对应。

最小项的性质包括：

- (1) n 变量可以构成 2^n 个最小项；
- (2) n 变量的所有最小项的逻辑和（或）为1，例如 $\overline{A}\overline{B} + \overline{A}B + A\overline{B} + AB = 1$ ；
- (3) 任意两个不同的最小的逻辑乘（与）为0；
- (4) n 变量的每一个最小项有 n 个相邻项（即有一个变量不同、剩余 $n-1$ 个变量均相同的最小项）；
- (5) 对于任何一个最小项，只有一组变量值使它的值为1，变量的其余组合值均使它为0，例如只有 $\{A=0, B=1\}$ 使 $\overline{A}B=1$ 。

如果在与或式中，所有与项均为最小项，则称该式为最小项表达式/标准与或式。任何逻辑函数都可以转化为最小项表达式，只需要列出真值表后，将真值表中函数值为1的最小项相或即可。

4. 逻辑化简：

常用卡诺图（K图）对逻辑函数进行化简。

$$F = \overline{A}\overline{C}D + AB\overline{C} + ACD + \overline{A}BC$$

$CD \backslash AB$	00	01	11	10
00		1		
01		1	1	1
11	1	1	1	
10			1	

2 组合逻辑电路

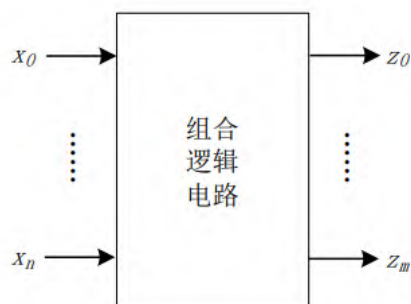
笔试/面试考点：根据真值表写出逻辑函数或根据逻辑函数列出真值表、并描述电路功能，竞争和冒险的定义，消除冒险的方法

项目应用：组合逻辑Verilog写法

2.1 逻辑电路的分类

逻辑电路是一个因果系统，只有当输入发生变化时，输出才有可能发生变化。逻辑电路通常可分为两类：组合逻辑电路和时序逻辑电路。

如果一个逻辑电路的输入相同时，输出也相同，即电路中任意时刻的输出仅取决于当前时刻的输入，则称为组合逻辑电路（Combinational Logic Circuit）。组合逻辑电路的输出 Z 是输入 X 的逻辑函数， $Z = F(X)$ 。



如果一个逻辑电路的输入相同时，输出不相同，则它一定不是组合逻辑电路。如果电路的输出不仅取决于当前时刻的输入，而且与决定电路当前时刻状态的过去输入有关，则称为时序逻辑电路（Sequential Logic Circuit）。

2.2 组合电路的分析与设计方法

在分析组合电路时，若已知信息是逻辑电路图，则应从输入端开始，逐级推导输出端的逻辑函数，再根据逻辑函数列出真值表，根据逻辑函数或真值表来描述电路的功能。若已知信息是电路的真值表，则需要根据真值表列出逻辑函数，期间可能会需要进行逻辑化简，再根据化简结果还原出逻辑电路的门级实现结构。

在设计组合电路时，首先应对要实现的功能进行逻辑抽象，确定输入、输出的逻辑变量所对应的实际含义，将功能转化为输入、输出+真值表的表达形式；再根据真值表，进行必要的化简，选用对应的逻辑门进行实现。

2.3 常用组合电路

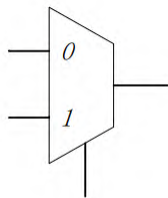
1. 数据选择器/数据分配器：

数据选择器也称为多路选择器MUX，具有 n 个地址输入/片选输入、 2^n 个数据输入、1个数据输出，其功能是从多路数据输入信号中，依据地址输入信号，选择其中一路输出。

数据选择器的功能表如下所示：

A_{n-1}	A_{n-2}	A_1	A_0	Y
0	0	0	0	D_0
0	0	0	1	D_1
.....
1	1	1	1	D_{2^n-1}

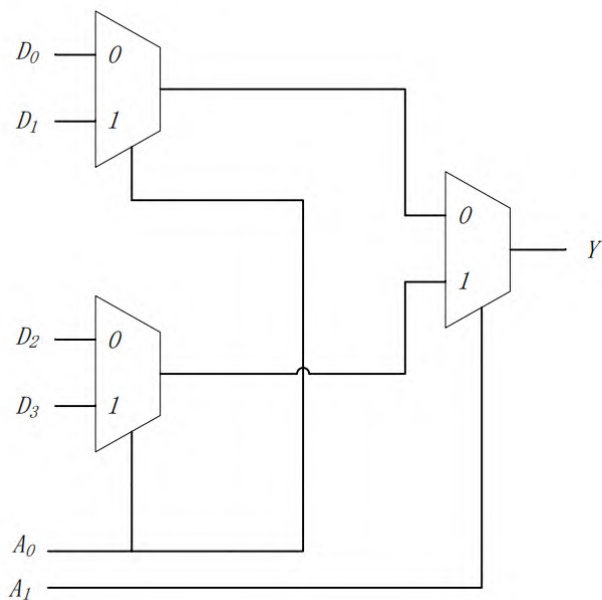
2选1MUX的逻辑符号如下所示：



2选1MUX的Verilog描述如下所示:

```
always @ (*) begin
    case (A)
        1'b0: Y = D[0];
        1'b1: Y = D[1];
        default: Y = D[0];
    endcase
end
```

可以用多个MUX实现更多输入的MUX，例如用3个2选1MUX可以实现1个4选1MUX:



与数据选择器功能相反的电路是数据分配器。数据分配器也称为多路分配器、解复用器DEMUX，具有 n 个地址输入/片选输入、1个数据输入、 2^n 个数据输出，其功能是依据地址输入信号，将输入数据信号传输到对应输出通路，其余输出通路则保持无效输出。

数据分配器的功能表如下所示:

A_{n-1}	A_{n-2}	A_1	A_0	Y_{2^n-1}	Y_1	Y_0
0	0	0	0	x	x	D
0	0	0	1	x	D	x
.....
1	1	1	1	D	x	x	x

2. 编码器/译码器:

编码器具有 2^n 个输入、 n 个输出，当某一路输入有效时，输出为该路的编码结果。通常而言，为了防止多路输入同时有效，会对各路信号的优先级进行规定，因此通常所说的编码器也称为优先编码器。

编码器的功能表如下所示（1表示有效）：

I_{2^n-1}	I_{2^n-2}	I_1	I_0	Y_{n-1}	Y_{n-2}	Y_1	Y_0	Y_{vld}
0	0	0	0	0	0	0	0	0
1	x	x	x	1	1	1	1	1
0	1	x	x	1	1	1	0	1
.....
0	0	1	x	0	0	0	1	1
0	0	0	1	0	0	0	0	1

以8-3编码器为例，其Verilog描述如下所示：

```
always @ (*) begin
    if (I[7] == 1'b1) begin
        {Y, Y_vld} = {3'b111, 1'b1}; // {3'd7, 1'b1}
    end
    else if (I[6] == 1'b1) begin
        {Y, Y_vld} = {3'b110, 1'b1}; // {3'd6, 1'b1}
    end
    .....
    else if (I[0] == 1'b1) begin
        {Y, Y_vld} = {3'b000, 1'b1}; // {3'd0, 1'b1}
    end
    else begin
        {Y, Y_vld} = {3'b000, 1'b0};
    end
end
end
```

与编码器功能相反的电路是译码器。译码器具有 n 个输入、 2^n 个输出，其功能是根据输入信号，在对应的输出数据线上输出有效电平，其余数据线上输出无效电平。

3. 加法器:

加法器，通常也称为全加器，用于实现加法运算。全加器的基础是一位二进制全加器，具有2个输入比特、1个输入低位进位比特、1个输出比特和1个输出高位进位比特，其真值表如下：

C_I	A	B	C_O	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

由真值表可得，全加器的逻辑表达式为 $S = A \oplus B \oplus C_I$ ， $C_O = AB + AC_I + BC_I$ 。

一位全加器的Verilog实现如下：

```
assign {C_o, S} = A + B + C_i;
```

对多位全加器而言，可以用多个一位全加器级联构成，因此也称为行波进位加法器。但当加数位数较多、加法器级数较多时，各级之间的电路延迟会累加到一个较大值，因此行波进位加法器的延迟比较大、效率比较低。

对此，出现了改良后的超前进位加法器，通过对进位部分增加了一部分逻辑电路，使得各级之间可以先并行计算，减少了电路整体的延迟。

2.4 组合电路中的竞争与冒险

在逻辑电路中，逻辑门、信号走线均存在延迟，且各个信号的传输路径不同。因此，“组合逻辑的各个输入到达逻辑门的时间不一致”这一现象称为竞争。由于竞争的存在，当输入信号的电平发生瞬间变化时，电路可能短暂地产生与预期结果不一致的输出电平，这样的电平称为毛刺，由于竞争产生毛刺的现象称为冒险。

冒险的判别方法：（1）代数法，当存在 $F = X + \bar{X}$ 或 $F = X \cdot \bar{X}$ 时，便可能存在冒险现象；（2）K图法：若存在两个相切的K圈、且相切处未被其他K圈包围，则可能存在冒险现象。

冒险的消除方法：（1）增加冗余项，即利用K图，在K圈相切处增加冗余项；（2）加滤波电容；（3）加选通信号。

3 时序逻辑电路

笔试/面试考点：触发器的工作原理、内部结构等（重点在D锁存器和D触发器），锁存器与触发器的异同，常用时序电路代码掌握

项目应用：常用时序电路代码掌握，避免锁存器写法，状态机设计

3.1 时序电路的结构

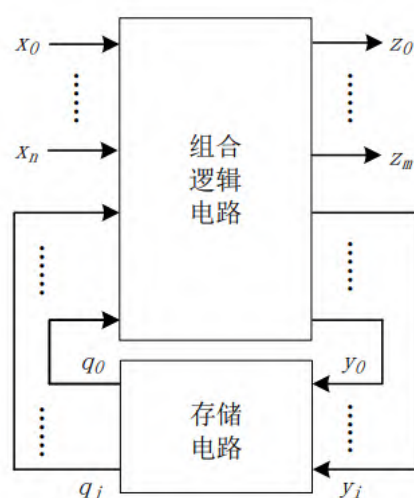
时序逻辑电路可以分为两部分，组合逻辑电路和存储电路，其中存储电路具有记忆功能，能够存储电路过去输入的相关信息，通常称之为电路状态。对整个时序电路而言，电路外部输入为 X ，输出为 Z （外部输出）；从电路内部结构来看，存储电路的输入是组合逻辑电路的输出 Y （内部输出/存储激励），存储电路的输出是 Q （内部输入/状态）；而整个电路的外部输出 Z 和内部输出 Y 则由外部输入 X 和内部输入 Q 共同决定。因此，时序逻辑电路可以通过三个函数表示：

（1）状态转移函数 $Q^{n+1} = H(Y, Q^n)$ ，其中 Q^n 称为现态， Q^{n+1} 称为次态，描述了存储电路存储状态的更新；

（2）输出函数 $Z = F(X, Q)$ ，描述了时序电路的输出，由输入与现态共同决定；

（3）激励函数 $Y = G(X, Q)$ ，描述了控制存储电路的信号变化。

若时序逻辑电路的输出与现态和输入都有关，则称为米里（Mealy）型时序电路；若输出仅与现态有关，则称为摩尔（Moore）型时序电路。



3.2 时序电路的分析与设计方法

时序逻辑电路的工作过程，实质上就是在不同的输入条件下内部状态不断更新的过程。通常而言，无论是分析电路还是设计电路，都需要关注电路的状态图/状态转移图，也可以采用时序图来辅助理解或设计。绘制时序图的软件推荐：TimeGen，[时序图绘制工具 --- TimeGen使用方法总结元直数字电路验证的博客-CSDN博客timegen](#)



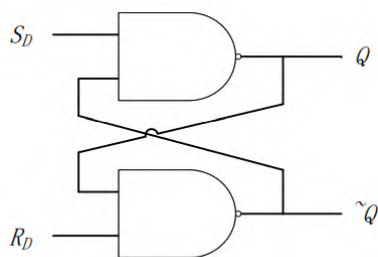
3.3 时序电路的基本单元

触发器具有记忆功能，是时序电路中存储电路的主要单元。触发器的本质是一种输出反馈电路，即在基本逻辑门（与门、或门、异或门等）中，将逻辑门的输出端连接到该逻辑门的其中一个输入端，这样的电路能够产生“保持”的概念，即记忆功能。常用的触发器可分为基本RS触发器和钟控触发器，钟控触发器又可细分为电平触发和边沿触发。

【注意，本段话中所说的“触发器”一词指广义上的触发器，即具备记忆功能的电路；而我们常说的触发器是指钟控触发器中边沿触发的电路，锁存器则是指钟控触发器中电平触发的电路；本部分重点关注D锁存器和D触发器即可，其余电路仅作辅助理解】

1. 基本RS触发器：

基本RS触发器由两个与非门构成，具有2个输入端和2个互补的输出端，其逻辑电路组成如下：



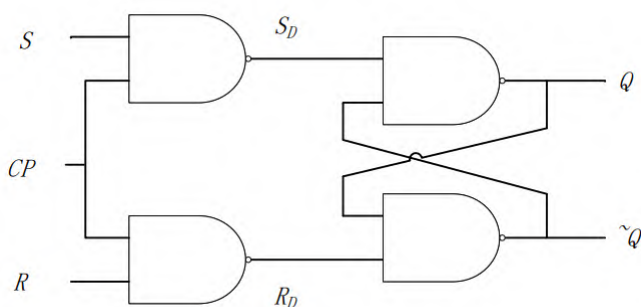
基本RS触发器的状态转移逻辑与输出如下：

- (1) 当 $R_D = 0, S_D = 1$ 时，由 $R_D = 0$ 可得，无论现态为何， $\sim Q^{n+1}$ 一定为1，继而次态 Q^{n+1} 一定为0，因此这一状态称为“复位”， R_D 为低电平有效的复位端；
- (2) 当 $R_D = 1, S_D = 0$ 时，同理，无论现态为何，次态 Q^{n+1} 一定为1，因此这一状态称为“置位”， S_D 为低电平有效的置位端；
- (3) 当 $R_D = 1, S_D = 1$ 时，次态 $Q^{n+1} = \sim(\sim Q^{n+1}) = Q^{n+1}$ ，即“保持”状态；
- (4) 当 $R_D = 0, S_D = 0$ 时，两个与非门输出均为1，不再满足互补关系，因此基本RS触发器不允许输入同时为0，输入需满足约束 $R_D + S_D = 1$ 。

因此可得，基本RS触发器的特征方程（状态转移函数+约束条件）为： $Q^{n+1} = \sim(S_D) + R_D Q$ ， $R_D + S_D = 1$ 。

2. 钟控RS触发器（RS触发器）：

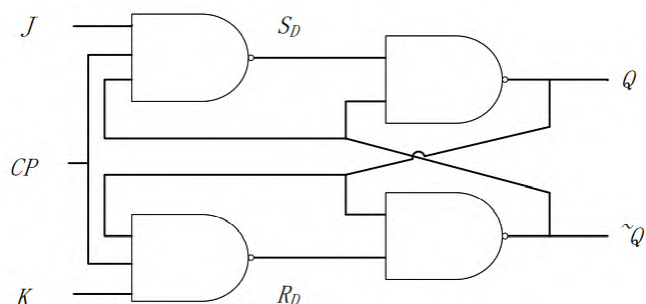
在基本RS触发器的基础上，加入时钟信号，就可以使电路按一定的时钟节拍进行状态切换，这样构成的电路就是钟控触发器。钟控RS触发器在基本RS触发器的基础上，增加了两个与非门，其逻辑电路组成如下：



当 $CP = 0$ 时，无论 R, S 为何值，必有 $R_D = 1, S_D = 1$ ，故时钟低电平期间电路处于“保持”状态；当 $CP = 1$ 时， $R = \sim R_D, S = \sim S_D$ ，故时钟高电平期间电路的状态才有可能改变，同时复位端和置位端均为高电平有效。

3. 钟控JK触发器（JK触发器）：

在钟控RS触发器的基础上进行改动，可以得到钟控JK触发器，其逻辑电路组成如下：



当 $CP = 0$ 时，钟控JK触发器与钟控RS触发器相同，电路处于“保持”状态；当 $CP = 1$ 时，分四种情况：

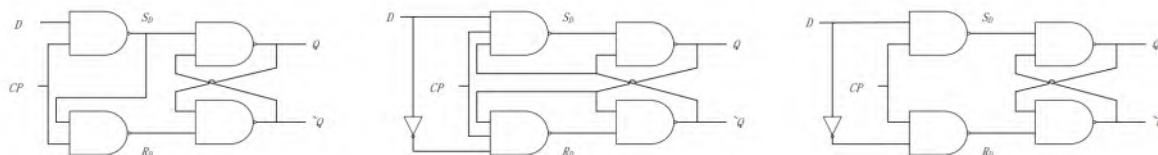
- (1) $J = 0, K = 0$ ，则次态 $Q^{n+1} = Q$ ，即“保持”状态；
- (2) $J = 0, K = 1$ ，则次态 $Q^{n+1} = 0$ ，即“复位”状态；
- (3) $J = 1, K = 0$ ，则次态 $Q^{n+1} = 1$ ，即“置位”状态；
- (4) $J = 1, K = 1$ ，则次态 $Q^{n+1} = \sim Q$ ，即“翻转”状态。

4. 钟控T触发器（T触发器）：

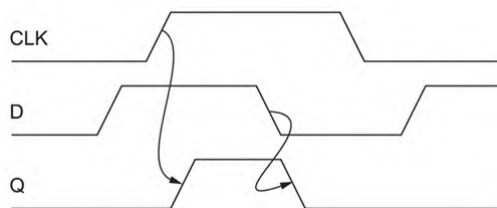
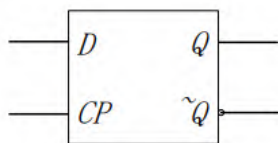
将钟控JK触发器的两个输入端连接到一起，可以得到钟控T触发器。因此，当 $CP = 1$ 时，若 $T = 0$ ，则电路处于“保持”状态；若 $T = 1$ ，则电路处于“翻转”状态。

5. D锁存器（钟控D触发器/数据锁存器/锁存器）：

在钟控RS触发器或钟控JK触发器的基础上进行改动（改动的本质是在S端输入D，R端输入 $\sim D$ ），可以得到钟控D触发器。当 $CP = 0$ 时，输出保持不变；当 $CP = 1$ 时，输出等于输入，可以实现信号保存、锁存功能，因此也称为D锁存器、数据锁存器和锁存器（Latch），其逻辑电路组成如下图：



D锁存器的特征方程为： $Q^{n+1} = D$ ，其逻辑符号可简记为下图：

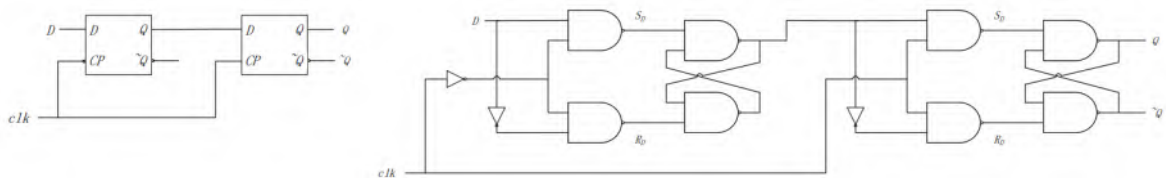


在Verilog中使用以下描述可以综合出D锁存器：

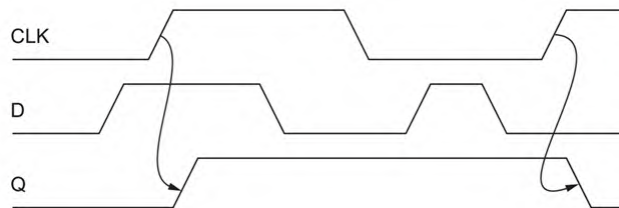
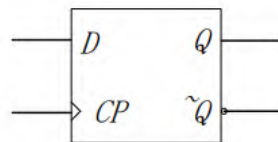
```
always @ (clk) begin
    Q <= D;
end
```

6. D触发器（触发器）：

D锁存器在时钟电平有效时导通输出，无效时保持数据。利用两个有效电平相反的D锁存器，可以得到一个D触发器（Flip-Flop），D触发器是边沿触发的器件，只在时钟变化的有效沿（上升沿或下降沿）时，才将输出导通。上升沿D触发器的逻辑电路组成如下图：



D触发器的逻辑符号可简记为下图：



在Verilog中使用以下描述可以综合出D触发器：

```
always @ (posedge clk) begin
    Q <= D;
end
```

3.4 锁存器与触发器的比较

锁存器和触发器的不同点：

- （1）锁存器是电平敏感的电路，属于非同步控制，而触发器是边沿敏感的电路，属于同步控制；
- （2）锁存器由于在电平有效时持续导通，受布线延迟影响较大，容易产生毛刺，不利于时序收敛和时序分析，而触发器不容易产生毛刺；
- （3）锁存器消耗的资源（逻辑门）比触发器少。

通常在电路设计中，会避免产生锁存器，因为锁存器容易产生意想不到的结果。电路设计时，可能会综合出预想之外的锁存器的情况有（这些情况原本预想的是综合出组合逻辑）：

- （1）在组合逻辑中，if或if-else if缺少else分支；
- （2）在组合逻辑中，case情况未列全时缺少default分支；
- （3）在组合逻辑中，always的敏感列表没有列全。

```

/* 缺少else分支，误综合出锁存器 */
always @ (*) begin
    if (a == 'd5) begin
        b = 'd3;
    end
end

/* 缺少default分支，误综合出锁存器 */
always @ (*) begin
    case (curr_state)
        IDLE: begin
            output_data = 'd0;
        end
        WORK_STATE_1: begin
            output_data = 'd1;
        end
    endcase
end

/* 敏感列表未列全，误综合出锁存器 */
always @ (a, b) begin
    if (c == 'b1) begin
        output = a + b;
    end
    else begin
        output = a - b;
    end
end

```

3.5 常用时序电路（自学）

1. 计数器：

代码掌握：模2的次幂计数器、模N计数器

2. 分频器：

代码掌握：利用D触发器的行波分频、偶数分频、奇数分频（50%占空比）

了解（没遇到考过）：N.5小数分频、平均意义上的任意小数分频

3. 移位寄存器

4. 序列检测器：

代码掌握：状态机写法、移位寄存器写法；非重叠检测、重叠检测、及其他变种

5. 线性反馈移位寄存器：

掌握原理：两种LFSR结构，生成多项式

代码掌握：CRC检测

6. 序列发生器：

掌握原理：本原多项式、M序列

代码掌握

7. 时钟门控电路：

原理掌握、代码掌握

8. 时钟切换电路:

原理掌握、代码掌握

通用解决方案: 结构冒险还可使用改变寄存器/端口方式解决

7. 时钟相关处理

(时钟切换, 门控时钟)

时钟

7. 时钟切换

(1) 普通切换

$$\text{或: } = (\text{select} \& \text{clk-a}) \mid (\sim \text{select} \& \text{clk-b})$$

assign clk_out = select ? clk-a : clk-b;

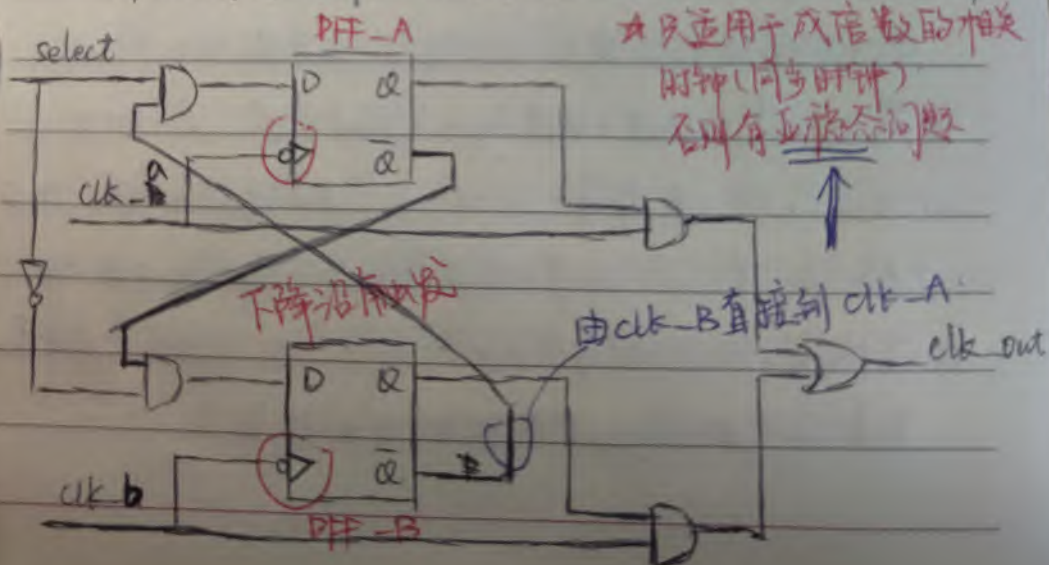
⇒ 电路成本小、简单, 但切换时存在毛刺。

⇒ 可在电路非工作, 或有复位时使用

(2) 无缝切换 (相关时钟)

(毛刺)

⇒ 即使得电路中没有超出 $\max\{\text{clk-a}, \text{clk-b}\}$ 的脉冲



想用下
将两个
X clock
元, 这样
k-out 可以
切换对上升沿

```
always @ (negedge clk-a or posedge rst) begin
    if (rst) begin
        Q-A <= 'b0; [Q-B <= 'b0;]
    end
    else begin
        [Q-B <= ~select & ~Q-A;]
        Q-A <= select & ~Q-B;
    end
end
```

作用: 在下降沿寄存 select, 使得高电平期间输出不变, 不会被破坏。 (因为 TA 电平的, 无论哪个时钟输出都为 0。 只要保证当前时钟完整输出高电平之后再换到另一个时钟的 TA 电平处切换即可)

时序分析:

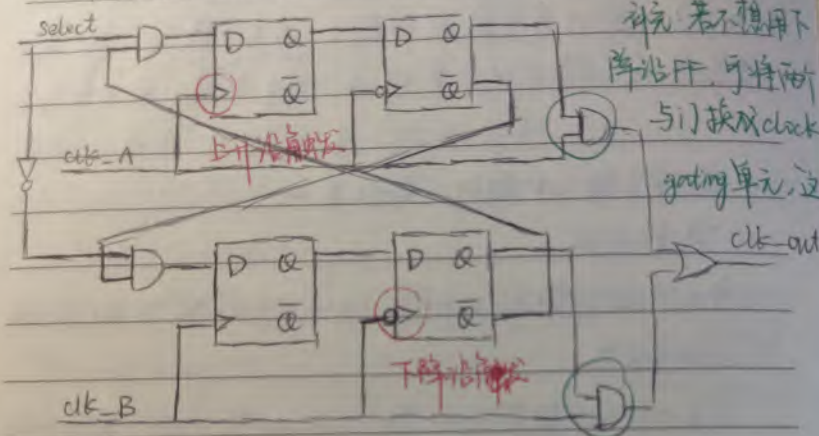
- 假设现在正在输出 A 支路, B 支路为 0。
- (1) 某时刻 t_1 , select 切换到 B 支路。
 - (2) t_1 之后 clk-A 的第一个下降沿 t_2 , select 信号寄存, 此后 A 支路输出 0。 由于 B 支路仍为 0, 故 clk-out = 0。
 - (3) t_2 之后 clk-B 的第一个下降沿 t_3 , 此后 B 支路开始输出 (从 TA 电平开始输出)。

DATE / /

S M T W T F S
O O O O O O O

13) 无缝切换 (异步时钟)

在(12)的基础上增加一个或多个上升沿同步器即可。



补充: 若不想用下降沿FF, 可将两个与门换成clock

gating单元, 这样就可以把下降沿FF换成上升沿

8. 门控时钟 clock gating

1) 简单门控

⇒ 将门控信号和时钟信号通过与门后输出

⇒ 问题: 会有毛刺

2) 锁存器门控

⇒ 将门控信号经过负时钟沿的锁存器后, 再和时钟信号相与, 使得门控只在时钟由平期间变化

DATE / /

always @ (p

if (rst)

Q = A

end

else be

gating单元, 这样就可以把下降沿FF换成上升沿

end

作用: 在

会被破坏

及保证

TA电平处

时序分析

假设

1) 某时

2) t_{12}

A支

3) t_{23}

B支

9. 去抖动

debounce 电路

⇒ 循环移位寄存器接收输入，当全0或全1时认为输出0或1，其余认为是抖动。和具体应用场景有关

10. 时钟分频 《时钟分频技术》

(1) 偶数分频

方式一：计数器 $0 \sim N/2 - 1$ ，计到 $N/2 - 1$ 时将输出时钟翻转

方式二：计数器 $0 \sim N/2 - 1$ ，计到0时（即时间结束就翻转一次）将输出时钟翻转 ⇒ 可保证各种分频时钟起始沿同步

(2) 奇数分频

方式一：计数器 $0 \sim N - 1$ ，期间翻转一次（翻转时刻决定了占空比），计到 $N - 1$ 时翻转一次 ⇒ 非50%占空比

方式二：计数器 $0 \sim N - 1$ ，一个上升沿触发器，一个下降沿触发器，分别在计到 $\frac{N-1}{2}$ 和 $N - 1$ 时翻转一次，最终输出 = clk_pos / clk_neg ⇒ 50%占空比

DATE

(3) $N.5$ 小数分频

即相当于每 $N.5$ 个周期 / 每 $(2N+1)$ 个半周期, 为输出时钟的一个周期。

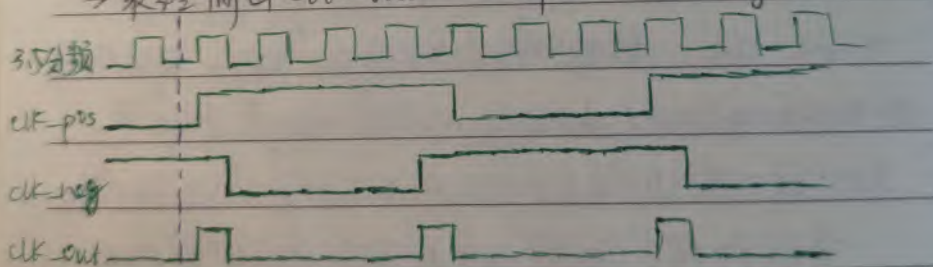
方式一: 采用上升沿 / 下降沿 2 计数器, 在复位结束后的第一个对应沿开始计数

上升沿计数器 $0 \sim 2N$, 计到 0 和 $N+1$ 时翻转, 初始输出 $clk_pos = 0$ 。

下降沿计数器 $0 \sim 2N$, 计到 0 和 N 时翻转, 初始

输出 $clk_neg = 1$

⇒ 最终输出 $clk_out = clk_pos \oplus clk_neg$ ⇒ 占空比 $\frac{1}{2N+1}$

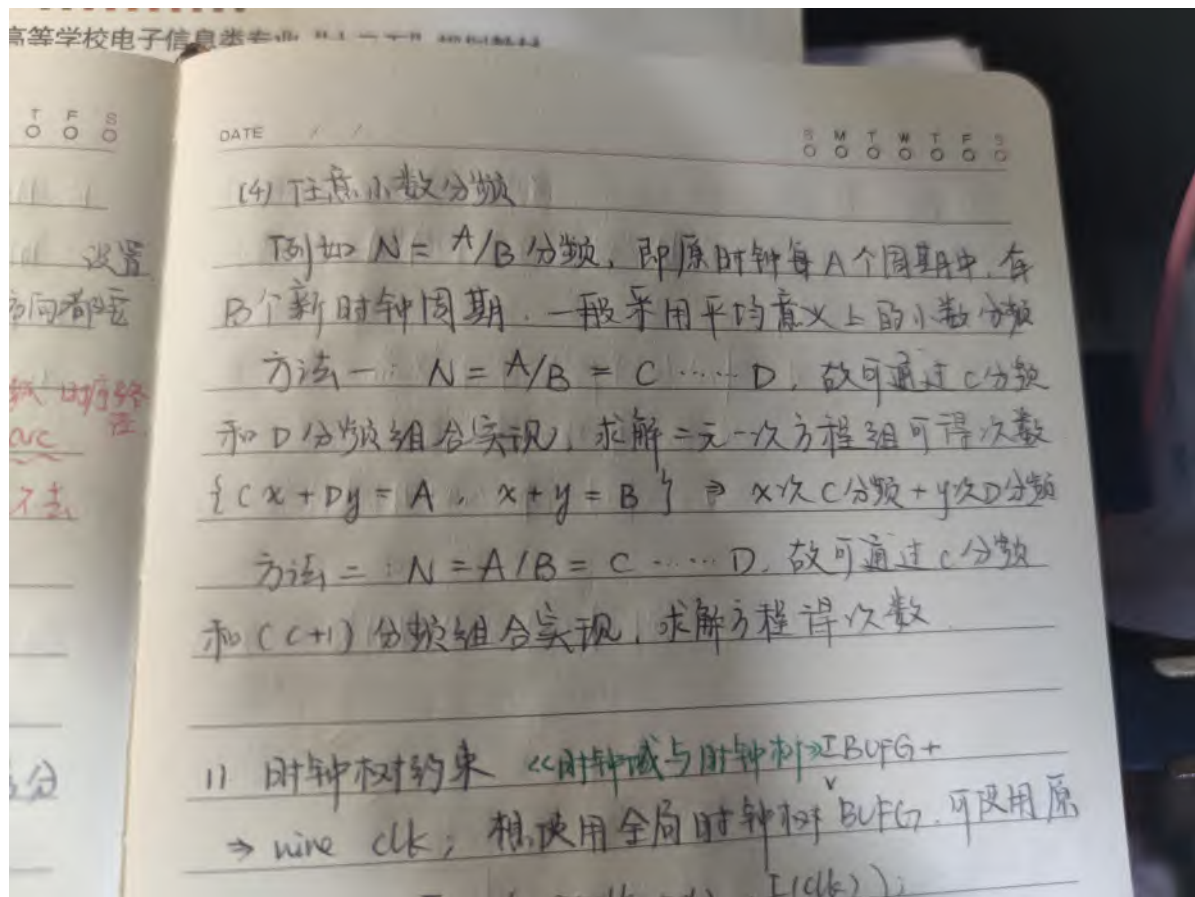


复位新放

向下取整

方式二: 上升沿计数器改为计到 0 和 $N+1 + \frac{N}{2}$ 时翻转, 下降沿计数器改为计到 $\frac{N}{2}$ 和 N 时翻转

⇒ 占空比 $(N: 2N+1)$, 接近 50%



4 存储器

笔试/面试考点: 存储器的分类、各自的特点和不同点比较

项目应用: SDRAM/DDR数据带宽计算, SDRAM/DDR如何使用 ([Xilinx MIG 控制器使用详解51CTO博客](#)
[opendaylight控制器](#))

4.1 存储器的基本原理

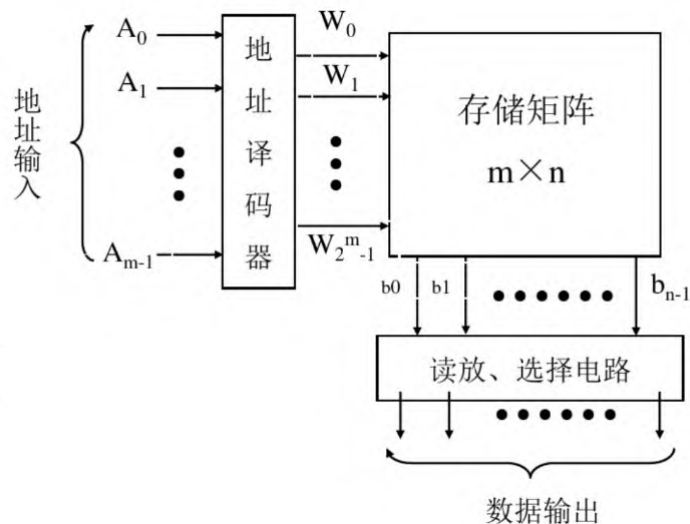
存储器是时序逻辑电路的一种, 是用来存储程序和各种数据信息的记忆电路, 存储器的内部由地址译码、存储矩阵、读写控制和输入输出控制四个基本单元构成. 存储矩阵由多个存储单元构成, 用于存储数据信息; 输入的地址信息通过地址译码单元指向存储矩阵中的对应存储单元; 读写控制单元和输入输出控制单元则负责控制数据流向, 实现存储和读出功能。

从读写功能的角度, 可以将存储器分为只读存储器 (ROM) 和随机存取存储器 (RAM)。一般而言, ROM在正常工作时只能读、不能写, 数据是通过专用方式写入的, 可以长期存储, 断电后也不会丢失, 属于非易失性存储器; RAM在正常工作时可以读也可以写, 但数据会在断电后消失, 属于易失性存储器。

4.2 常用的存储器

1. ROM、PROM、EPROM、EEPROM:

ROM (Read Only Memory), 即只读存储器, 是通过掩模工艺制造、将数据永久固化保存的半导体电路, 在使用时只能读出数据, 不能对数据进行修改或写入. 掩模ROM由于不可修改, 也称为固定ROM. ROM的内部由地址译码器、存储矩阵和输出缓冲器三部分构成。



PROM (Programmable ROM)，即可编程ROM，其在出厂时将存储内容固定为全0或全1，根据用户需求，可采用熔丝或PN结击穿的方式进行一次性编程。由于熔丝或PN结击穿都无法恢复，因此PROM只能编程/改写一次。

EPROM (Erasable PROM)，即可擦除可编程ROM，可以进行多次擦除和改写。其利用浮栅技术生产，编程过程相当于向MOS管浮栅注入电荷，编程结束后浮栅上的电荷会被绝缘层包裹，不会泄露，故电荷分布不变、数据不会丢失。当使用紫外线照射时，EPROM内部的电荷分布会被破坏，使得电路恢复到初始状态，完成了擦除过程。EPROM由于使用紫外线进行擦除，也称为UVEEPROM。

EEPROM (Electrically EPROM)，即电可擦除可编程ROM，与EPROM相比，它的特点是通过电信号进行擦除和写入，且擦除速度比EPROM快得多。

2. SRAM、DRAM：

RAM (Random Access Memory)，即随机存取存储器，可以随时读写，而且速度很快，通常作为OS或其他正在运行中的程序的临时资料存储介质。RAM可进一步分为SRAM和DRAM。

SRAM (Static RAM)，即静态随机存取存储器，是在静态触发器的基础上附加门控管而构成的，依靠触发器存储数据，对SRAM的读写操作，可类似于对寄存器进行读写操作，在读数据时相当于查看寄存器的状态，在写数据时相当于改变寄存器的状态。SRAM每个存储元（存储1bit数据）需要花费6个晶体管才能存储1 bit，所以SRAM的集成度低、占用面积大、成本较高，一般用于CPU中的cache（高速缓冲存储器）。SRAM在寻址时，同时发送行地址和列地址，把行地址和列地址拼接起来作为一次读写访问的地址，然后放到地址线上。

DRAM (Dynamic RAM)，即动态随机存取存储器，依靠电容存储数据。在DRAM进行读操作时，需要将电容与外界形成回路，通过检测电流变化判断存储元存储的是1还是0，因此读操作会造成电容中电荷流失（数据丢失）；同时电容上的电荷也只能保存2ms左右，综合以上两点，DRAM需要不断刷新来保持数据，DRAM刷新以行为单位，每次刷新一行存储单元。DRAM每个存储元只需要花1个电容和1个晶体管，因此集成度比较高、成本比较低，但是相对SRAM来说访问速度比较慢，一般被大量的采用作为系统的主存。DRAM在寻址时，把行地址和列地址分为长度相等的两段，然后分两次发送。

3. SDRAM、DDR：

SDRAM (Synchronous DRAM)，即同步动态随机存取存储器，是DRAM的一种，主要突出的“同步”，利用单一的一个的系统时钟同步所有地址数据和控制信号。与DRAM相比，SDRAM只有一个异步接口，因此SDRAM的传输速率要高于DRAM。

DDR SDRAM (Double Data Rate SDRAM)，简称DDR，即双倍速率同步动态随机存取存储器，可以在同一个时钟脉冲传输两次数据，上升沿和下降沿均可以进行数据传输，故其等效频率等于核心频率的2倍。例如16bit位宽的DDR，核心频率为400MHz，则其数据传输速率为 $400M * 16bit * 2 = 12.8Gbps$ 。目前已有DDR、DDR2、DDR3、DDR4等几代DDR芯片，各代DDR芯片之间的主要差别在于存储容量、传输速

率、电压、功耗等，代数越高的DDR，时钟频率越高→传输速度越快，容量越大，电压越小→功耗越低。

DDR SRAM Standard	DDR2	DDR3	DDR4
Internal Rate (MHz)	133-200	133-200	133-200
Bus Clock (MHz)	266-400	533-800	1066-1600
Prefetch	4n	8n	8n
Data Rate (MT/s)	533-800	1066-1600	2133-4266
Transfer Rate (GB/s)	4.2-6.4	8.5-14.9	17-21.3
BVoltage (V)	1.8	1.35/1.5	1.2

4. Flash:

Flash，也称Flash ROM、闪存，也属于非易失性存储器，是EEPROM的改进产品，结合了ROM和RAM的优势，既可以快速读写数据，又可以实现断掉不丢失数据，且可以用电信号对数据进行擦除和改写。Flash的特点是必须按照Block进行擦除，而EEPROM则可以按照Byte进行擦除。U盘中使用的通常是FLASH。

Flash主要有NOR Flash和NAND Flash两种。NOR Flash的读取方式与SDRAM相同，NAND Flash则是以一次一块的形式读取数据，块大小通常是512Byte。通常而言，NOR Flash用于存储少量、快速读取的数据，NAND Flash则用于存储大容量数据。两者的比较：

- (1) NOR Flash可按照SDRM方式读取，NAND Flash必须一次读取一块大小（512Byte）；
- (2) NOR Flash读速度略快于NAND Flash；
- (3) NOR Flash写速度慢于NAND Flash；
- (4) NOR Flash擦除速度慢于NAND Flash，因为其擦除块更大（NOR 64-128KB，NAND 8-32KB）。

5 集成门电路（自学）

笔试考点：三极管工作状态相关，根据CMOS电路写出逻辑函数

5.1 半导体逻辑电路

1. 半导体、PN结：

可参考[半导体PN结以及逻辑门电路以及软件思考 - 简书 \(jianshu.com\)](https://jianshu.com/p/1e1e1e1e)或网上其他资料

关注点：大概了解半导体和PN结的结构、工作原理

2. 二极管、三极管、场效应管（MOS管）：

可参考[二极管三极管电路原理和理解寒听雪落的博客-CSDN博客二极管放大电路基本原理](#)、[2、通过mos管构成的逻辑门电路你板子冒烟了的博客-CSDN博客mos管组成的逻辑电路](#)或网上其他资料

关注点：了解二极管和三极管的结构、工作原理，三极管的三种工作状态（截止、放大、饱和），MOS管的结构，PMOS和NMOS管构造的逻辑门电路

5.2 TTL集成逻辑电路

可参考[第13讲 TTL集成逻辑门电路 - 百度文库 \(baidu.com\)](#)或网上其他资料

5.3 CMOS集成逻辑电路

可参考[CMOS 集成逻辑门电路 - 百度文库 \(baidu.com\)](#)或网上其他资料