数字电路与系统

Leo

2024 年 3 月 12 日

目录

绪论

上升时间：一个脉冲从低电平转换为高电平的时间称为上升时间，实际上，若设高电平幅度为 V_m，则一般把上升沿从 10% 到 90% 的时间所需要的时间定义为上升时间；类似地，可以定义下降时间。

脉冲宽度：从脉冲前沿到达 $50\%V_m$ 起，到脉冲后沿到达 $50\%V_m$ 为止的时间定义为脉冲宽度。

占空比：定义为脉冲宽度与脉冲周期的比值。

冯诺依曼体系结构计算机的五大功能：

1. 把需要的程序和数据送入计算机
2. 必须具有长期记忆的能力
3. 能完成各种算数与逻辑运算
4. 能够根据需要控制程序的走向，并能根据指令控制机器各部件的协调工作
5. 能够按照要求把计算结果输出给用户

1 数制与码制

1.1 数制

以十进制为例：

1. 权：10 的整数幂次方如 100，10，1，0.1 称为十进制的权
2. 基：表示数的数码的集合，如集合 {0,1,2,3,4,5,6,7,8,9}
3. 基数：数码集合的大小，也即基的大小，十进制的基数是 10

进制之间的转换：

1. 十进制转二进制，整数部分除以 2 取余数，逆序排列得到整数部分，小数部分通过乘以 2 取整数部分，正序得到二进制小数部分。小数转换，保留的位数与原数字的有效数字一样，比如十进制保留两位小数，则二进制应该保留到 1/100 以后，即 1/128，即 7 位小数
2. 二进制转十进制
3. 二进制转八进制：每三位一组转换，整数部分位数不够则高位补零，小数部分低位补零
4. 二进制转十六进制：每四位一组转换，整数部分位数不够则高位补零，小数部分低位补零
5. 八进制、十六进制转二进制：按位转换
6. 八进制与十六进制相互转换：先转成二进制

1.2 码制

计算机采用二进制码，并且用 0 代表正，1 代表负，这种数值化的二进制数称为机器数。机器数一般用原码、反码和补码表示。

1.2.1 原码

格式为：符号位＋数值位（尾数）。运算时，符号位不参与运算，单独处理。有正零和负零之分。

1.2.2 反码与补码

正数的反码与补码都与原码一样！反码直接各数值位取反，补码各数值位取反再加一。一般计算机用补码运算，数值位算出的溢出位不用管直接舍去。

1.3 计算机中的码

数值编码分为

1. 定点数：一般小数点固定在最高位之前或最低位之后，用于表示纯小数和纯整数
2. 浮点数：就是数的科学计数法，包含阶符、阶码、数符和尾数。更加常用。

字符编码：ASCII 和汉字编码

数字字符编码：

1. NBCD（8421）码：跟正常的二进制数转十进制数一样。注意：这里针对的是一个个的数字字符，比如十进制的 10，表示为二进制数为 1010，表示为 8421BCD 码就是 00010000.
2. 2421BCD 码：第一个码元权值为 2. 从十进制的 5 开始，最高位都为 1.
3. 余 3BCD 码：是一种无权码，由 8421 码加三构成
4. BCD 循环码：无权码，任意相邻两个代码之间只有一位不同。最特别的是 9，表示为 1000，而不是 1101. 不存在 10-15 的编码

5. 格雷码：与 BCD 循环码很像，但是 BCD 循环码只用于表示十进制数，0-8 的表示与 BCD 循环码无异。不限于 10 进制数，可以一直编码下去。

用 BCD 码表示的数进行运算时，每一位的大小不能超过 9，当相加的结果大于 9 时，需要再加 6，溢出来的高位视作进位。

检错码与纠错码

1. 奇偶校验码：由信息位与冗余位（校验位）组成，校验位的取值要看事先规定是奇校验还是偶校验。假设信息中 1 的个数是奇数，采用奇校验，校验位就取 0（以此保证了整个校验码中 1 的个数是奇数），偶校验取 1（以此保证了整个校验码中 1 的个数是偶数）. 总之要满足校验码中 1 的个数的奇偶性与奇校验还是偶校验一致。校验的基本操作是异或。

2. 海明码：属于纠错码

3. 5 位代码中包含两个 1 的十进制数编码

2 逻辑函数及其简化

2.1 复合逻辑运算

异或 $F = A \oplus B = A'B + AB'$!!! 各种符号
同或 $F = A \odot B = AB + A'B'$
与或 $F = AB + CD$
与或非 $F = (AB + CD)'$

2.2 逻辑代数的基本定律

常用基本定理：利用对偶规则，每一条都有两种形式

交换律	$AB = BA$	$A + B = B + A$
结合律	$A(BC) = (AB)C$	$A + (B + C) = (A + B) + C$
分配律	$A(B + C) = AB + AC$	$A + (BC) = (A + B)(A + C)$
吸收律	$A(A + B) = A$	$A + AB = A$
控制律	$A \cdot 0 = 0$	$A + 1 = 1$
自等律	$A \cdot 1 = A$	$A + 0 = A$
重叠律	$A \cdot A = A$	$A + A = A$
互补律	$A \cdot A' = 0$	$A + A' = 1$
双重否定律	$A'' = A$	$A'' = A$
反演律	$(AB)' = A' + B'$	$(A + B)' = A' \cdot B'$

完备集：一个代数系统，如果仅用它所定义的运算中的某一组运算就可以实现所有的运算，则这一组运算是完备的，称为完备集。比如，{与，或，非} 是一个完备集，{与，非} 和 {或，非} 也是完备集。非运算是不可替代的。

2.3 逻辑代数的基本规则

代换规则 任何出现变量 X 的逻辑等式，若将所有出现 X 的地方用另一个逻辑表示式代换，该等式仍然成立。

对偶规则 对任意一个逻辑函数 F，将表达式中所有的常量 1 与 0 对换，所有的逻辑符号与和或对换，且保持原函数变量之间的运算顺序不变，则得到原函数的对偶函数 F^*，原函数具有的一切性质，对偶函数都具备。如果两个函数相同，则它们的对偶函数也相同

反演规则 对任意一个逻辑函数 F，将表达式中所有的常量 1 与 0 对换，所有的逻辑符号与和或对换，所有的函数变量换成其反变量，且保持原函数变量之间的运算顺序不变，则得到原函数的反函数 F' 注意区分对偶和反演！只有后者要将函数变量换成其反变量！还要注意反变量与非运算的区别！

2.4 逻辑代数的常用公式

并项公式 $AB + A'B = B$，根据对偶规则，有 $(A + B)(A' + B) = B$

消冗余因子公式 $A + AB' = A + B, A(A' + B) = AB$

消冗余项公式 $AB + A'C + BC = AB + A'C, (A + B)(A' + C)(B + C) = (A + B)(A' + C)$

2.5 逻辑函数及其描述方法

逻辑函数：任何对 n 个逻辑变量进行有限次逻辑运算的逻辑表达式都称为逻辑函数表示为 $F = f(x_1, ..., x_n)$. 常用的表示方式有：逻辑函数表达式，逻辑图，真值表，卡诺图，标准表达式。

2.5.1 逻辑函数表达式

不具有唯一性

2.5.2 逻辑图

不具有唯一性

2.5.3 真值表

具有唯一性

2.5.4 卡诺图

具有唯一性。妙处在于：利用循环码做坐标，将逻辑相邻性转化位几何相邻性，更加直观。* 高于 4 维的卡诺图不再直观，因为关于循环码对称轴的两个对称的卡诺图小块也算“相邻”，化简时，只有圈出的大圈也是关于循环码对称轴对称的才有效

2.5.5 标准表达式

最小项：真值表中每一行或卡诺图中每一个小格对应的那个与项。用 m_i 表示

最大项：真值表中每一行或卡诺图中每一个小格对应的那个或项。用 M_i 表示。最大项可以通过对应位置的最小项取非得到。注意：用卡诺图写最大项时是反的，101 对应 $A' + B + C'$

标准与或表达式：任何一个逻辑函数都可以写成最小项与对应函数值乘积之和的形式。具有唯一性

$$F = \sum_{0}^{2^n-1} f_i \cdot m_i \tag{1}$$

一般会把 F 写成变量的与或形式，而不会写成 $\sum m(1,3,4...)$ 的形式

标准或与表达式

$$F = \prod_{0}^{2^n-1} (f_i + M_i) \tag{2}$$

2.6 逻辑函数的化简

意义：使用的门少，代表着可以使用更少的器件；可以减小芯片的面积

标准：（以与或为例）与项最少且每一项的变量最少。反映到电路中，就是与门少，且每个门的输入最少（引线因此减少）

方法：公式法与卡诺图法。同一个卡诺图，看 0 可以化成或与式，看 1 可以化为与或式

* 非完全定义的逻辑函数的化简：无关项与约束项，在卡诺图中统一用 × 表示，化简时可以看作 1 也可以看作 0. 约束项有些是自然满足的，有些要靠人为实现（但这并不需要在该逻辑函数化简中考虑）。

3 组合逻辑电路的分析与设计

全加器：输入 ABC(C 为低位向本位的进位)，输出本位 F1，进位 F2，$F_1 = A \oplus B \oplus C, F_2 = (A \oplus B)C + AB$

3.1 组合逻辑电路的竞争与险象

竞争：一个门的多个输入信号到达的时间有先后快慢之分，这种时差现象就是竞争

现象：当输入信号变化时，由于存在竞争，在输出端出现错误，出现瞬时的干扰脉冲现象，称为险象。

静态险象：输入信号在输入信号变化前后，稳定的输出值相同，而仅在转换瞬间有险象

功能险象：两个或多个输入发生变化，由于路径不同产生的险象

逻辑险象：只有一个输入发生变化，或者有多个信号变化，但是没有功能险象的可能，由于门的延迟不同却产生了静态险象

动态险象：输入信号在输入信号变化前后，稳定的输出值相同（应该从 0 到 1 或从 1 到 0）。这种情况下不会出现静态险象但是可能在最终输出稳定之前出现短暂的反复如 1-0-1-0 或 0-1-0-1. 只有在多级电路中才会出现。

只要消除了静态险象就不会出现动态险象。

静态险象的分类

$$\begin{cases} \text{按干扰信号分} \begin{cases} \text{静态 0 险象: 本来应该变化前后应一直为 1，但出现了 1-0-1 的干扰序列} \\ \text{静态 1 险象} \end{cases} \\ \text{按产生原因分} \begin{cases} \text{功能险象} \\ \text{逻辑险象} \end{cases} \end{cases} \tag{3}$$

3.1.1 功能险象的判断方法

从卡诺图看，如果信号变化的路径经过了不同的取值，则会发生功能险象。

从逻辑表达式看，如果在输入变量变化时，由不变化的输入变量组成的乘积项既不是函数包含的项，也不是冗余项（所以不能通过加冗余项的方法消除），则会发生功能险象。

功能险象不能通过修改逻辑设计消除，一般只能通过选通输出来解决。功能险象仅在两个及以上变量发生变化时才会发生

3.1.2 逻辑险象的判断方法

当用上述方法判断没有发生功能险象之后，就有可能发生逻辑险象（只有一个变量变化的时候，仅有可能发生逻辑险象）

当函数表达式里没有写冗余项的时候就可能发生逻辑险象。逻辑险象可以通过增加冗余项来消除，这时就不是最简逻辑表达式了。

4 常用的组合逻辑功能器件

4.1 编码器

将输入的信号转换为对应的二进制数码

4.1.1 4-2 编码器

可以设置一个额外的输出端，当其输出为 1 时表示编码器正常工作，这样可以避免“未工作状态”和“工作状态输出 00”两种情况的混淆。

缺点在于四个入口不允许任何两个入口同时有有效信号输入。

4.1.2 优先编码器

规定输入为 I_0, I_1, I_2, I_3 其中 I_3 的优先级最高。逻辑函数

$$Y_1 = I_3 + I_3' I_2 \tag{4}$$

$$Y_0 = I_3 + I_2' I_1 \tag{5}$$

$$F_{GS} = I_3 + I_2 + I_1 + I_0 \tag{6}$$

可以从功能表直接看出逻辑函数应该怎么写。看教材 p124 表 6-2，Y_1 对应出 1 的行是倒数两行，其中倒数第一行表示，只要 I_3 是 1，其余无所谓是什么，Y_1 都出 1，这一行对应的表达式为 I_3；倒数第二行表示当 $I_2 = 1, I_3 = 0$ 时无所谓其余是什么，Y_1 都出 1，这一行对应的表达式为 $I_3' I_2$.

4.1.3 集成编码器

74148　　8 线输入 3 线输出优先编码器，输出的是反码形式的 8421 码。

1. $I_0', \cdots I_7'$: 输入端，低电平有效，标号越大优先级越高，低电平为有效信号

2. $Y_0', \cdots Y_2'$: 输出端，标号越大权重越高

3. Y_S: 选通输出端，当为 0 时表示输出不代表编码，为 1 时表示输出的是有效编码

4. Y_{EX}': 扩展端，其值与 Y_S 相反，即 Y_{EX}' 输出为低电平时表示本片输出的编码有效

5. EN': 使能端；接低电平的时候才工作

两片 74148 构成 16-4 线优先编码器，从 $I_0', \cdots I_{15}'$ 编码至 $B_3', \cdots B_0'$：

1. 高位片的 Y_S 接低位片的 EN'，使得当低位片全出高电平（无效）

2. 高片的 Y_{EX}' 接 B_3'，这样其实就是表示：最高位输入在高片时，代表输入的十进制数大于等于 8，$B_3' = 0$ 这正好与高片工作时 $Y_{EX}' = 0$ 相契合

3. B_2', B_1', B_0' 分别是两片对应端的与，因为两片不会同时工作，不工作的一片出 1，代表放弃对这一个位的控制权。

74147 10-4 优先编码器，输出反码形式的 8421 码。输入端为 $I_1', \cdots I_9'$，当输入全为高电平时认为输入数值为 0. 输出 $Y_0', \cdots, Y_3'$.

4.2 译码器

将二进制信号还原为原来的信号

4.2.1 二进制译码器

2-4 译码器 输入为高电平有效，输出为低电平有效，由于将输出取与非就可以得到最小项的或（即与或式），所以可以做最小项发生器。74139 就是两个 2-4 译码器的集成块

74138 3-8 译码器

1. A_2, A_1, A_0：输入端，标号越大，权重越高

2. EN：使能端，分成三个子端口：E_3, E_2', E_1'

3. $Y_0', \ldots Y_7'$：输出端

两片 74138 扩展为 4-16 译码器：从 $B_3, ..., B_0$ 译码成 $Y_0', \ldots Y_{15}'$.B_3 接高片的 E_3 和低片的 E_2' 这样使得当输入的二进制数大于 7 时，高片工作而低片不工作，输出仅有可能在 $Y_8', ..., Y_{15}'$ 出现。$B_0, ...B_2$ 都同时接到两片的对应位。剩下的使能子端口都接到可以使该片工作的电位。

4.2.2 二-十进制译码器 7442

输入的是 8421 码，输出的是 10 个高低电平信号

1. $A_3, ..., A_0$：输入信号，标号大的权重大。

2. $Y_0', ..., Y_9'$：输出信号，用位置表示数码

4.2.3 显示译码器

7448 七段显示译码器　　专用于和七段显示器件合用。如共阴极连接的 BS201（接高电平发光，7448 可以直接驱动）或共阳极连接的 BS211

1. $A_3, ..., A_0$：输入信号，标号大的权重大。
2. $Y_a, ..., Y_g$：输出信号，与标号与显示器件 BS201 对应的引脚相连。
3. LT'：试灯信号，当为有效信号（0）且 BI'/RBO' 为无效时，七段灯全亮
4. RBI'：动态灭零信号，当试灯信号无效，本信号有效时，不显示零，但是不妨碍其他显示
5. BI'/RBO'：可以做输入也可以做输出。做输入时用于熄灭信号，即当它有效时，不管其他输入是什么一律不亮。做输出时，用于显示本译码器有没有灭零。如 $LT' = 1, RBI' = 0, A_3A_2A_1A_0 = 0000$，则 $RBO' = 0$，表示本片灭零了。当 $LT' = 0$(试灯) 或 $LT' = 1, RBI' = 1$（不试灯但是没有灭零输入），则 $RBO' = 1$，表示本片没有灭零。RBO' 主要用于方便多片显示器的灭零。一般将最高片的 RBI' 置零，代表最高位强制动态灭零。剩下位的 RBI' 接高一位的 RBO'，表示只有当上一片灭零了这一片才灭零。

4.3 数据选择器

从一组输入的数据中选出所需要的一个数据，将其输出到唯一的数据通道上，相当于一个从多路到一路的开关。常用作多路低速数据到一路高速数据的转换。可以作为函数发生器（这里地址端反而像是被数据端“选择”）。与编码器类似，只不过选择器相当于将编码器输出的量再揉到一起。

4.3.1 双四选一数据选择器 74153

集成两个相同的四选一数据选择器，有两个使能端。

1. A_1, A_0：地址端，作为控制信号。比如输入 01 时选择 D_1 抛弃其他数据。
2. $D_0, ..., D_3$：数据端，待选择的信号

4.3.2 八选一数据选择器 74151

1. S'：使能端
2. $D_0, ..., D_7$：数据端，待选择的信号
3. A_2, A_1, A_0：地址端，标号越大，权重越高
4. W, W'：输出端

作为函数发生器，若要实现 $F(A, B, C) = \sum m(3, 5, 6, 7)$，只需要将变量接到地址端，在 3，5，6，7 处将数据端接成 1 其余接 0 即可。注意看哪一个是高位！位数不够还可能用到卡诺图降维。

4.4 数据分配器

与数据选择器的作用相反，将一个数据送到多个输出通道中的一个。实际上与带有使能端的译码器相同。所以 74138 也是一个数据分配器。，当地址端给出从 000 到 111 的单个低脉冲节拍信号时，输出端会从 0 到 7 按节拍“点亮”，可用作分时数据传输系统。

4.5 数值比较器

单位二进制数比较大小，由真值表可以看出逻辑函数：

$$F_{A>B} = AB' \tag{7}$$

$$F_{A=B} = A'B' + AB = A \odot B \tag{8}$$

$$F_{A<B} = A'B \tag{9}$$

多位数值比较，可以分解为从高位开始的单位比较。

7485 中规模 4 位数值比较器　　用二进制数输入两个待比较的数 A, B

1. $A_0, B_0, ..., A_3, B_3$：输入两个待比较的数
2. $a < b, a = b, a > b$：级联端，当本片的八个输入端两两都相同时，本片的输出就取决于级联端.$A < B, A = B, A > B$ 与 $a < b, a = b, a > b$ 一一对应相关，互不影响。
3. $A < B, A = B, A > B$：输出端，高电平有效

两片 7485 构成 8 位二进制数值比较器。级联时低片的 $a < b, a = b, a > b$ 接 010，表示若一直比到低片比完，两个数每一位都相等，那说明这两个数其实就是相等的。高片的 $a < b, a = b, a > b$ 接低片对应的输出端

4.6 奇偶校验位产生与校验电路

取值原则：采用奇校验时，奇校验位保证整个代码中 1 的个数为奇数；偶校验反之。

奇偶校验分为校验位的产生与校验两部分，两种功能可以由一个电路来实现。奇偶校验的基本运算是异或。把输入位全部异或，若有奇数个 1，则全部异或的结果为 1。

1. $D_0, ..., D_7$：数据位
2. D_{in}：校验位，当用于产生校验位时，该端置零
3. F_{ODD}：奇校验输出，当 $D_0, ..., D_7$ 和 D_{in} 中有奇数个 1 时输出为 1。电路实现时就是所有输出位的异或结果。
4. F_{EVEN}：偶校验输出，当 $D_0, ..., D_7$ 和 D_{in} 中有奇数个 1 时输出为 0。在电路实现上，F_{EVEN} 就是 F_{ODD} 取非。

当电路用作校验电路时，若选择奇校验为校验规则，则只需要看 F_{ODD}，当输出为 1 时，说明输入有奇数个 1，当输出为 0 时，说明输入有偶数个 1；若选择偶校验为校验规则，则只需要看 F_{EVEN}，当输出为 1 时，说明输入有偶数个 1，当输出为 0 时，说明输入有奇数个 1。

当电路用作产生电路时，若取 $D_{in} = 0$，这样 F_{ODD} 取出来就可以作为偶校验位，加到数据位后面，使得整个代码始终有偶数个 1；F_{EVEN} 取出来就可以作为奇校验位，加到数据位后面，使得整个代码始终有奇数个 1。若取 $D_{in} = 1$，F_{ODD} 取出来就可以作为奇校验位，F_{EVEN} 取出来就可以作为偶校验位。

74280 中规模集成奇偶校验电路

1. $A, B, ..., I$：九位输入端
2. $EVEN, ODD$：输出端。$F_{ODD} = A \oplus B \oplus C \oplus D \oplus E \oplus F \oplus G \oplus H \oplus I = F'_{EVEN}$

4.7 算术运算电路

4.7.1 全加器

输入为 A, B, C，其中 C 表示低位向本位的进位；输出 S, CO,CO 表示向高位的进位。

$$S = A \oplus B \oplus C \tag{10}$$

$$CO = (A \oplus B)C + AB \tag{11}$$

逐位进位的级联全加器运算速度比较慢。

4.7.2 超前进位的 4 位二进制全加器

74283 超前进位的四位全加器

1. $P_0, Q_0, ..., P_3, Q_3$：各是 4 位输入端
2. C_{-1}：低位向本位的进位
3. CO：本位向高位的进位
4. $S_0, ..., S_3$：本位运算结果

4.7.3 减法运算

输入为 A, B, C，分别表示被减数，减数和低位向本位的借位；输出 S, CO,CO 表示向高位的借位。

$$S = A \oplus B \oplus C \tag{12}$$

$$CO = A'(B \oplus C) + BC \tag{13}$$

借助补码，可以将加法与减法统一起来。用全加器可以实现求补运算。加法和减法运算。

用全加器实现求补运算　　容易知道，一个变量与 0 异或会保持自身，与 1 异或相当于取反。对于一个四位二进制数 $D_3D_2D_1D_0$，将其按位跟 1 异或后灌入四位加法器的 $B_3B_2B_1B_0$，并将 $A_3A_2A_1A_0$ 与地相连，最后将 C_{-1} 置为 1，这样就实现了负数的求补运算。更巧的是可以用一位信号 V 来控制一个数是否求补。可以做到：当 $V = 0$ 时，不操作（即相当于把 $D_3D_2D_1D_0$ 这一正数“求补”）；当 $V = 1$ 时，将 $D_3D_2D_1D_0$ 当作负数求补。另外，也可以用 4 个双二选一数据选择器 74157 实现求反运算，将加一交给加法器做，也相当于把一个 4 位二进制数取补。

用全加器统一实现加减法　　74283+74157：其中 74157 通过外部输入 S 来决定要不要取反——作为加数时不取反，作为减数时要取反。S 同时控制要不要对全加器“加一”——加法时不加一，减法时要加一（相当于与 74157 共同实现了求补运算）。被加数（或者被减数）不变。可以看出，S 相当于决定了运算的类型，0 代表加法，1 代表减法。

用 74283 实现 8421 码到余 3BCD 码的转换　　原数加 3 即可

用 74283 和门电路实现：2 个（4 位）8421 码相加的结果用 8421 码表示　　一个 8421 码只能用于表示 0-9，如果像二进制数一样相加，得到的结果是二进制数而不是 8421 码。比如，当加法结果位 11 时，用二进制数表示应为 1011，进位为 0；用 8421 表示应为 0001，进位为 1. 所以，当计算结果的值超过 9 后，需要进行修正。修正的条件为 $CO + B_8B_4 + B_8B_2$(标号表示权值)，修正方法是：对满足修正条件的情况，在全加器运算结果的基础上再加 6. 一共用到两片全加器、两个双输入与门和一个三输入或门，整体的进位应取 $CO + B_8B_4 + B_8B_2$ 的结果。（p162）

4.8 算术逻辑单元 ALU

一位集成算数逻辑单元　　用 $S_2S_1S_0$ 表示 8 种基本算数或逻辑运算，一般用 S_2 区分是逻辑运算还是算数运算。输出为 f, 可能的进位为 CO。

ALU 可分为算数单元 AU 和逻辑单元 LU 和二选一数据选择器。逻辑单元可以用与非门构成，也可以用与、非、或、异或门和四选一数据选择器实现。算数逻辑单元可以用上文的统一加减器来实现，S_1S_0 控制最低位向本位的进位（借位）

74181 四位集成算数逻辑单元

1. M：高电平表示执行逻辑运算，低电平表示执行算术运算。
2. CI'_n：进位信号，仅在算数运算时有用

5 触发器

时序逻辑电路常用的存储单元是触发器与锁存器。锁存器 (latch) 是指在对输入电平敏感期间，可以根据输入随时改变输出的器件；触发器 (flip-flop) 是只在时钟信号变化的瞬间改变其状态的器件。

锁存器是一种对脉冲电平敏感的存储单元。锁存器对输入电平持续时间敏感，也就是在持续电平期间都会接收输入并在输出端发生变化。触发器对时钟边沿敏感，在边沿来临时状态发生改变。锁存器一般没有时钟控制，触发器受时钟控制，只有在触发时采样当前输入产生输出的改变。

存储单元必须具有如下特点：一是具有两个稳定的状态，能够存储一位的二进制信息；二是能够根据输入信号的变化，将存储单元清零或置一，完成状态转移。

描述锁存器和触发器的方法有状态转移方程（特征方程）、状态转移真值表、状态转换图、时序图和激励表。

不同类型的触发器转换的方法有两种：比较触发器状态转移方程的方法和根据状态转移表求激励表的方法。

5.1 锁存器

锁存器也叫基本触发器（但不是触发器）。可以分为没有控制端的锁存器和有门控端的锁存器。

5.1.1 R-S 锁存器

输入信号直接加在输出上，输入信号可以立刻表现出来，所以 S、R 也叫做直接置位端、复位端. **由两个与非门构成的 R-S 锁存器**

1. 输入端为 S', R'，输出端为 $Q.Q'$，一般把 Q 端的状态称为 R-S 锁存器的状态。
2. 逻辑表达式：$Q = (S'Q')', Q' = (R'Q)'$

3. 状态转移方程:$Q^{n+1} = S'' + R'Q^n$

4. 约束条件：$S' + R' = 1$

功能表

S'	R'	Q^{n+1}	功能
1	1	Q^n	保持
0	1	1	置一
1	0	0	清零
0	0	1	同态（同时撤销时状态不定）

由两个或非门构成的 R-S 锁存器

1. 输入端为 S, R，输出端为 Q, Q'，一般把 Q 端的状态称为 R-S 锁存器的状态。

2. 逻辑表达式：$Q = (R + Q')', Q' = (S + Q)'$

3. 状态转移方程:$Q^{n+1} = S + R'Q^n$

4. 约束条件：$SR = 0$

功能表

R	S	Q^{n+1}	功能
0	0	Q^n	保持
0	1	1	置一
1	0	0	清零
1	1	1	同态（不确定）

锁存器可以用来做机械消抖开关. 这利用了 RS 锁存器的这样一个特性: 只有第一次触发起作用, 后面的毛刺不再使锁存器翻转.

5.2 带有控制端的锁存器

5.2.1 门控 R-S 锁存器

在基本 R-S 锁存器的前面加了两个引导门和一个控制信号 CP

1. 状态方程:$Q^{n+1} = S + R'Q^n$

2. 约束条件:$RS = 0$(上述两条都是 CP=1 时的情况, 当 CP=0 时, 锁存器保持)

5.2.2 门控 D 锁存器

当需要将输入信号直接送到缓冲器暂存时, 门控 D 锁存器最方便. 它是将 RS 之间接一个反相器, 用一个输入信号同时控制两个输入端口. 一般令 $S = D, R = D'$.

1. 状态方程:$Q^{n+1} = D + D''Q^n = D$

2. 约束条件:$RS = DD' = 0$ 恒成立 (上述两条都是 CP=1 时的情况, 当 CP=0 时, 锁存器保持)

5.2.3 门控 J-K 锁存器

是在与非门构成的门控 R-S 锁存器基础上进行改造的.

$$R' = (CP \cdot K \cdot Q^n)' \tag{14}$$

$$S' = (CP \cdot J \cdot Q'^n)' \tag{15}$$

$$Q^{n+1} = CP \cdot J \cdot Q'^n + (CP \cdot K \cdot Q^n)' \cdot Q^n = JQ'^n + K'Q^n (CP = 1) \tag{16}$$

约束条件 $R' + S' = 1$ 天然满足. 但是当 $J = K = 1, CP = 1$ 时,$Q^{n+1} = Q'^n$, 锁存器飞速翻转.

5.2.4 门控 T 锁存器

基本功能是: 控制信号为 1 时, 状态翻转; 控制信号为 0 时, 状态保持不变. 用其构成异步时序逻辑电路的二进制计数器非常方便. 门控 T 锁存器就是这样一种锁存器, 它在门控 J-K 锁存器基础上完成, 把 J 和 K 直接接到一起. 状态方程:$Q^{n+1} = CP \cdot TQ'^n + (CP \cdot T \cdot Q^n)' \cdot Q^n$, 当 CP=1 时,$Q^{n+1} = TQ'^n + T'Q^n$. 在 T=1 时会快速翻转.

上述的电路都是电平触发式电路, 当时钟有效电平宽度较大时, 会引起锁存器连续翻转, 为了避免这个问题, 采用主从触发器或边沿触发器.

5.3 主从触发器

主从触发器由两级钟控触发器构成, 接收输入信号的钟控触发器称为主触发器, 提供输出的钟控触发器称为从触发器.

5.3.1 主从 R-S 触发器

由两级与非结构的门控 R-S 锁存器组成, 两级门控端由反相的时钟信号控制.

1. 特征方程:$Q^{n+1} = S + R'Q^n$

2. SR=0

与门控 R-S 触发器相同. 不同之处在于,RS 对 Q 的作用分两步进行. 门控信号 CP 为 1 时, 接收信号;CP 为 0 时, 输出信号.

5.3.2 主从 J-K 触发器

主锁存器的状态转移方程为 $Q^{n+1}_{master} = JQ'^n + (KQ^n)'Q^n_{master} (CP = 1)$, 这里的 Q_{master} 是主触发器的输出,Q 是总的输出. 从触发器的状态转移方程为 $Q^{n+1} = Q^{n+1}_{master} (CP = 0)$. 特征方程为 $Q^{n+1} = JQ'^n + K'Q^n$. 会出现" 一次翻转问题", 要避免在 CP=1 时 JK 信号发生变化.

5.4 边沿触发器

触发器的状态仅仅在某一时刻发生, 进一步提高触发器的抗干扰能力.

5.4.1 边沿 D 触发器

又叫维持阻塞触发器, 有三级与非结构. 有置零维持线/阻塞线和置一维持线/阻塞线, 还有直接置一/置零端或叫异步使能端.

5.4.2 传输延迟 J-K 触发器

由一级与非门和一级与或非门实现. 与或非门的传输时间小于与非门的传输时间, 是后边沿触发的触发器.

5.5 CMOS 触发器

6 时序逻辑电路的分析与设计

时序逻辑电路可以分为组合逻辑电路和逻辑单元两部分。外输出用 Z 表示，外输入用 X 表示，存储单元状态用 Y 表示。方程组

$$z_i = f_i[x_1, x_1, ..., x_n, y_1, y_2, ..., y_k] = f_i(X, Y), i = 1, ..., m \tag{17}$$

称为即刻输出方程。存储单元的激励信号也是组合逻辑电路的输出，称为内输出。存储单元的激励方程或驱动方程可以表示为

$$w_j = g_j[x_1, x_1, ..., x_n, y_1, y_2, ..., y_k] = g_i(X, Y), j = 1, ..., k \tag{18}$$

对于一个时序逻辑电路，即刻输出方程和存储单元的状态转移方程是最重要的。

6.1 时序逻辑电路的分析方法

6.1.1 同步时序逻辑电路的分析方法

1. 根据时序电路图写出激励方程和输出方程，并由激励方程代入触发器特征方程，求出电路的状态转移方程
2. 根据状态转移方程和输出方程，写出状态转移真值表
3. 画出状态图，必要时还要画出时序图
4. 根据状态表、状态图和时序图，说明电路的逻辑功能

因为同步时序电路的各触发器在同一时钟同一时刻完成状态转移，所以不强调触发时刻。

6.1.2 异步时序逻辑电路的分析方法

1. 根据时序电路图写出时钟方程（说明触发条件）、激励方程和输出方程，并由激励方程代入触发器特征方程，求出电路的状态转移方程
2. 根据状态转移方程和输出方程（标记触发条件），写出状态转移真值表，表要包括时钟脉冲栏
3. 画出状态图，必要时还要画出时序图
4. 根据状态表、状态图和时序图，说明电路的逻辑功能

6.2 时序逻辑电路的设计方法

只考察同步时序逻辑电路的设计。

6.2.1 同步时序逻辑电路的设计方法

1. 根据功能要求，设定初始状态，得出对应状态表或状态图，一般以大写字母来表示状态。其中，得出状态表这一步很关键，也是最困难的一步。状态表中要有现态（大写字母）、次态和输出
2. 状态化简：两个等价的状态可以合并为一个状态。化简可以用状态表和蕴含表的方法。最后要写出最大等价类和最简状态转移表
3. 状态分配，又叫状态编码。状态分配的一般原则是若两个电路的次态/前态/输出相同，则这两个状态的编码应相互邻近。
4. 选择触发器类型，一般用 D 触发器
5. 根据编码的状态表和所采用的触发器，求出电路输出方程和驱动方程
6. 根据这两个方程画出电路图
7. 检查自启动

两个状态是否等价：

1. 在所有输入条件下，两个状态的输出相同
2. 在所有输入条件下，两个状态的状态转移效果相同

有些时候，两个状态转移的下一个状态不同，比如 $S_1 \to S_2, S_3 \to S_4$，则 1、3 是否等价取决于 2、4 是否等价，$[S_2.S_4]$ 称为 $[S_1.S_3]$ 等价的隐含条件。有时候两者会互为隐含条件。

等价具有传递性。

7 常用的时序逻辑电路模块

7.1 寄存器与移位寄存器

7.1.1 寄存器

寄存器是能够存储二进制信息的时序电路，它具有接收和寄存二进制数码的功能。触发器就是可以存储一位二进制信息的寄存器。常用的有 D 触发器构成的 4 位集成寄存器 74175，有异步清零和数据寄存功能。

7.1.2 移位寄存器

移位寄存器不但可以寄存数据，还可以将寄存的数据按照指令进行移位。移位是指在移位脉冲的作用下，寄存器中的数据可以根据需要向左或向右移动一位。

单向移位寄存器都是将 D 触发器的输出端给到下一个 D 触发器的输入端，区别仅在于次序。双向移位寄存器使用一位控制端 S，驱动方程为

$$D_0 = (SD'_{SR} + S'Q'_1)' \tag{19}$$

$$D_1 = (SQ'_0 + S'Q'_2)' \tag{20}$$

$$D_2 = (SQ'_1 + S'Q'_3)' \tag{21}$$

$$D_3 = (SQ'_2 + S'D'_{SL})' \tag{22}$$

右移移位寄存器 74195　　串行输入、并行输入和串并行输出的四位右移移位寄存器，异步清零。**双向移位寄存器 74194**　　异步清零、左移、右移、同步并行置数。移位寄存器可以用于串并行数据的转换，也可以用于构造环形/扭环计数器，还可以用于算数运算（二进制乘除相当于移位）和数据延迟

7.1.3 环形计数器与扭环计数器

用 D 触发器和移位寄存器可以方便地构成环形计数器和扭环计数器。n 个计数器可以实现模 n 环形计数器和模 2n 扭环计数器。

为了实现自启动，n 级环形计数器的反馈函数可以写成

$$D_0 = (Q_0 + Q_1 + \cdots + Q_{n-2})' \tag{23}$$

扭环计数器的自启动反馈修正只能在第一级改，且没有定法。

7.2 计数器

分类

1. 按计数进制分：二进制计数器和非二进制计数器（以十进制为代表）
2. 按数字的增减的计数方式分：加法计数器、减法计数器和加减可逆计数器
3. 按计数器中触发器翻转是否与计数脉冲同步分：同步计数器和异步计数器

7.2.1 二进制计数器

同步计数器略。异步计数器可以用 D 触发器时钟端级联实现。需要注意的是，下降沿触发的 D 触发器组成的是加法计数器，上升沿触发的是减法计数器。异步计数器工作速度慢，可以用同步计数器。**4 位二进制同步加法计数器 74161**　　异步清零、同步置数、数据保持（进位端可清零可保持）、加法计数。（对比 74163 是同步清零）**4 位二进制同步可逆计数器 74191**　　异步置数、数据保持、加法计数、减法计数。还有最大/最小控制端和进位/借位输出端，两者的区别在于，前者只要计数到最大/最小就马上给信号，后者还要等 CP=0 才给信号。后者其实是前者与上 CP' 和 EN' 再取非。

7.2.2 十进制计数器

8421BCD 码同步十进制加法计数器　　用 D 触发器构成

$$Q_0^{n+1} = D_0 = Q_0'^n \tag{24}$$
$$Q_1^{n+1} = D_1 = Q_1^n \oplus Q_0^n Q_3'^n \tag{25}$$
$$Q_2^{n+1} = D_2 = Q_2^n \oplus Q_0^n Q_1^n \tag{26}$$
$$Q_3^{n+1} = D_3 = Q_3^n \oplus (Q_0^n Q_3^n + Q_0^n Q_1^n Q_2^n) \tag{27}$$
$$Y = Q_0^n Q_3^n \tag{28}$$

8421BCD 码同步加法计数器 74160　　异步清零、同步置数、数据保持和加法计数，有进位端 $RCO = ET \cdot Q_3 \cdot Q_0$

7.3 集成计数器的应用

7.3.1 计数器的级联

74161 同步级联：低片的进位端同时给到高片的 ET、EP 端，两片共用时钟；
74161 异步级联：低片进位端取非给到高片的 CP 端。高片 ET、EP 端接 1

7.3.2 使用单个计数器构成任意进制计数器（模值小于单个计数器的模值）

异步清零法：比如要产生模 6 计数器，则要将 0110（实际上是第 7 个状态）转换为 0 给到异步清零端，这样会出现一瞬间的 0110

同步送数法：将 0101（第六个状态）转换成有效信号送到同步置数端。对于有同步清零端的计数器（比如 74163）可以将反馈信号送到同步清零端实现。

利用进位信号反馈置数法：不从 0000 开始计，而从 1010 记到 1111，利用进位信号给到同步置数端重置为 1010.

利用送数控制端的多次送数实现计数法：不具有推广性。

7.3.3 构成分频器

假设要实现 $\frac{1}{2^n}$ 级分频，则把待分频信号作为时钟，在级联的 74161 的第 n 个输出取信号即可。

7.3.4 构成脉冲节拍

二进制计数器＋ 74138 可以得到脉冲信号。

7.4 序列信号发生器

串行的周期性数字信号称为序列信号。序列信号是在时钟脉冲下产生的一串周期性二进制信号。。序列信号的一个周期内，包含的二进制数据位数称为序列长度。序列信号发生器分为计数型和移存型两种。

计数型序列信号发生器的特点是：所产生的序列信号长度等于计数器的模值，并可以根据需要产生一个或多个序列信号。一般把计数器和数据选择器搭配使用。

移存型序列信号发生器：移位寄存器的级数 n 应该满足 2^n 大于等于序列长度。例如序列长度为 8 则移位寄存器至少有 3 位。输出是以移位寄存器的某个触发器作为序列信号输出的，所以关键在于求出要移进来的数据与各触

发器的关系。当一种状态对应两种次态时，不能实现，要考虑增加位数。在写状态转移表时，行数要比序列长度多一行（这意味着最后一行与第一行重复），表示移位能够闭环。

8 半导体存储器

8.1 存储器基本概念

8.1.1 存储器的地址与容量

数字系统使用的存储器按存储方式可以分为半导体存储器（如内存）、磁介质存储器（如硬盘）、光存储器（如光盘）。半导体存储器读写速度快，一般用作数字系统主存储器。计算机系统中的存储器一般由锁存器或电容组成的存储单元阵列构成。

在多数数字系统中，由 8 位二进制数码构成的存储单位称为字节，存储器中一个完整的信息存储单位称为字，一个字一般由一个或多个字节组成。

存储器的容量一般用其能存储的字的容量来衡量。比如 16K×8K，表明存储器的字长为 8bit，能够存储 16K 个 8bit 的字。一般个人计算机的存储器是按字节来组织的。三维存储矩阵可访问的最小数据单位为字节。

8.1.2 存储器基本操作

存储器的基本操作有读和写操作两种。写操作是把数据放入由地址指定的存储单元；读操作是将由地址指定的存储单元中的数据拷贝出来。其中，选择指定存储单元地址的过程称为寻址操作。

计算机系统中用于传输写入和读出数据的连线称为数据总线。数据总线是双向的，在按字节组织的存储器中，数据总线至少有 8 根连线，并行传输 8bit 的数据。一般个人计算机的数据总线是 32 位的，用来寻址 $2^{32} = 4G$ 的内存地址。

8.2 RAM

RAM 分为两大类：静态 RAM（SRAM）和动态 RAM（DRAM）。DRAM 一般用电容作为其存储单元。

RAM 的基本结构：存储矩阵、地址译码器、读写控制器、片选控制器和数据缓冲器。存储矩阵是 RAM 的主体。读写控制器、片选控制器和数据缓冲器称为读写电路。

8.2.1 SRAM

SRAM 通常用锁存器作为基本存储单元。一个基本单元有 6 个 CMOS 管子，其中 4 个构成两个 CMOS 非门，构成具有双稳态结构的锁存器，用来存储 1bit 二进制数据。另外两个管子构成传输门。行选择线为高电位时有效。

SRAM 的优点是读写速度快。但是成本高，所以一般用作 CPU 与 DRAM 之间的高速缓存（cache）。

SRAM 使用了三态缓冲器来实现双向的数据端。三态缓冲器（用倒三角表示）有三种状态：高电平（1 状态）、低电平（0 状态）和高阻。

SRAM 的工作过程：片选 → 行译码 → 列译码 → 读写操作。下面是典型的 SRAM 读周期的时序图中，一些关键量的定义

1. 读周期时间 r_{RC}: 有效地址码的持续时间

2. 地址存取时间 t_{AQ}: 从有效地址码出现在地址端上到有效数据读出的这段时间

3. 片选使能存取时间 t_{EQ}: 从片选信号变为低电平到数据读出的这段时间

4. 输出使能存取时间 t_{GQ}: 从输出使能端变为低电平到数据读出的这段时间

一般真正把有效数据读出的这段时间很短，门开的时间和数据出现的时间都留有一定裕量。下面是典型的 SRAM 写周期的时序图中，一些关键量的定义

1. 读周期时间 r_{WC}: 有效地址码的持续时间

2. 地址设定时间 t_{SA}: 从有效地址码出现在地址端上到写使能端变为低电平的这段时间

3. 写脉冲宽度：写使能端维持低电平的时间

4. t_{WD}: 将要写入的有效数据出现在数据端后，写使能端必须保持为低电平的这段时间

5. 数据保持时间 t_{HD}: 从写控制端变为高电平后有效数据必须保持在数据输入端上的这段时间

8.2.2 DRAM

DRAM 使用电容存储信息。DRAM 的优点是结构简单、集成度高、成本低廉。缺点在于不能将信息维持很长时间，必须使用额外的电路定时对存储单元进行刷新。一般每隔 8ms 或 16ms 就要刷新。在每次读操作后将自动对选中行的所有存储单元刷新。

DRAM 主要用作计算机的主存储器。

一个单管动态存储单元由一个 MOS 管和存储电容组成，MOS 管是一个传输门，作为开关使用，高电平导通。单管动态存储单元采用矩阵式结构，在每根位线上接有输出缓冲器和刷新缓冲器（也叫灵敏恢复/读出放大器）

DRAM 的读写过程采用了地址复用技术——时分复用，通过行地址选择信号和列地址选择信号来判断当前 8 位地址是行地址还是列地址。8 位地址线可以实现寻址 $2^{16} = 65536$ 的地址空间。

8.3 ROM

ROM 在工作时，只能读出不能写入。分为内容固定 ROM，可编程 ROM（PROM），可擦除的可编程 ROM（EPROM）。

ROM 由地址译码器、存储单元和输出电路组成。ROM 可以看成一个 n 输入 m 输出的门网络，可用 n 个输入变量通过 2^n 个 n 输入与门生成 2^n 个最小项，再根据 m 个输出函数的标准表达式用 m 个或门将一些最小项选择相加，得到 m 个输出。即 ROM 是一个与或门网络。

ROM 的特点：存储单元简单、集成度高、具有不易失性。一般 BIOS 就集成在主板上的一个 E2PROM 或 Flash 芯片上。

8.3.1 PROM

只能由用户进行一次编程，在出厂时所有存储单元均为 0，使用者添 1. 有熔丝和反熔丝型两种。

EPROM 的存储单元是由浮栅雪崩注入型 MOS 管构成的，有两个栅极，没有引线的栅极叫做浮栅。当浮栅有电子积累时，相当于存 0；当浮栅没有电子积累且栅极为高电平时，相当于存 1. 用紫外光照射浮栅可以释放浮栅上的电子回到衬底，所以擦除后全部单元存 1.

E2PROM：利用隧道效应，给浮栅注入电子存 1. 编程时，先抹成全 1，再按照需求写 0.

Flash：与 E2PROM 一样都是电擦除可编程的，与 EPROM 工艺相似的 MOS 管，但是擦除时类似 E2PROM（借助隧道效应存 1）. 写入过程类似 EPROM，将热电子注入浮栅存 0.Flash 快且容量大。

8.4 存储器容量扩展

位扩展：增加每个地址中存储的数据的比特数；字扩展：增加存储矩阵地址的数量。ROM 和 RAM 的容量扩展类似。

8.4.1 位扩展

为了增加字长，必须增加存储器数据线宽度。

8.4.2 字扩展

为了增加字的数量，必须增加存储矩阵地址的个数，也即增加存储器地址线的宽度。

9 数模与模数转换

9.1 数模转换的基本原理

数模转换器是由二进制数码转换为模拟信号的译码器。数模转换电路就是将二进制数字量转换成其与数字成正比的电流信号或电压信号的器件。数模转换的目的是使得输出电压的量值与数字量代表的数值成正比。转换特性曲线由若干离散的点组成，连接这些离散点的连线称为理想参考线，其斜率为 S。

数模转换比例系数 S 又称为分辨力，可以用来衡量数模转换的精细程度，定义为相邻两二进制代码对应的输出电压的差值，或者最小数字单位所对应的输出电压值，也叫最低有效位 LSB。

为了将台阶信号平滑化，可以用低通滤波电路。

9.2 常用的数模转换方案

9.2.1 开关树译码方案

n 位数模转换器，利用 2^n 个相同的电阻串联形成分压器，产生 2^n 个标准电压，再通过开关网络将对应的电压切换到输出端。最大输出电压为

$$v_{Omax} = \frac{2^n - 1}{2^n} V_{REF} \tag{29}$$

分辨力 S 为

$$S = \frac{v_{Omax}}{2^n - 1} = \frac{V_{REF}}{2^n} \tag{30}$$

开关树译码方案的优点是电阻种类单一、转换速度快、对开关的导通内阻要求不高；缺点是随着转换位数的增加，所用元件的数量急剧增加。所以一般用在位数较少的数模转换。

9.2.2 权电阻网络译码方案

对于 n 位数模转换，用到了 n+1 个电阻且电阻值各不相同，其中，最高位 D_{n-1} 对应电阻为基准电阻 R，最低位 D_0 对应电阻为 2^{n-1}; 反馈电阻为 $0.5R$。还用到了反向放大器。汇集到放大器反向输入端的电流为

$$I_{\sum} = \frac{V_{REF}}{2^{n-1}R}\sum_{i=0}^{n-1} D_i 2^i \tag{31}$$

输出电压为

$$v_{OA} = -R_f I_{\sum} = -\frac{V_{REF}}{2^n}\sum_{i=0}^{n-1} D_i 2^i = -\frac{V_{REF}}{2^n}(N)_D \tag{32}$$

分辨力 S 为

$$S = -\frac{V_{REF}}{2^n} \tag{33}$$

其中 $(N)_D$ 为数码代表的值。注意输出电压和分辨力都是负值！

这种方案的缺点是：电阻网络的电阻值相差太大，电阻精度的控制比较困难。

9.2.3 权电流译码方案

一般采用倒 T 型电阻网络译码方案，只有 R 和 2R 两种电阻，从每个节点往左看都是 R 的阻值。总电流 IV_{REF}/R，每经过一个节点，电流一分为二。流入放大器反向端的电流为

$$I_{\sum} = \frac{I}{2^n}\sum_{i=0}^{n-1} D_i 2^i \tag{34}$$

输出电压为

$$v_{OA} = -R_f I_{\sum} = -\frac{V_{REF}}{2^n}\sum_{i=0}^{n-1} D_i 2^i = -\frac{V_{REF}}{2^n}(N)_D \tag{35}$$

注意输出电压和分辨力都是负值！但电流不是负值！

这种方案的优点是速度快、电阻种类少所以便于控制精度、用恒流源提供电流，不受开关导通内阻的影响，对开关电路的要求低、精度较高

9.2.4 权电容译码方案

大规模集成电路芯片中常常使用权电容译码方案。以最低位对应的电容值为基准设置 C_0 和辅助电容 C_0'，最高位的电容值为 $C_{n-1} = 2^{n-1}C_0$。基本原理是电容的分压原理。

这种方案的优点是：可以保证精度，精度只与基准电压和电容分压网络有关、电容尺寸可以很小，从而减小芯片面积降低成本、电容容值小，充放电快，转换速度高。

9.3 数模转换的主要技术指标

9.3.1 分辨率

分辨率定义为输出电压的最小变化量 S 与输出电压满量程 V_{REF} 的比值。有关系

$$R_{es} = \frac{1}{2^n - 1} \approx \frac{1}{2^n} = \frac{S}{v_{Omax}} \tag{36}$$

9.3.2 转换精确度

转换精确度是 DAC 的主要技术指标，用来衡量数模转换电路的实际输出值是否精确地等于理论输出值，通常用电路实际输出模拟电压与理论输出差值地最大值 ϵ_{max} 来表示。必须满足下述关系

$$\epsilon_{max} < \frac{S}{2} \tag{37}$$

所以转换精确度还可以表示为“±LSB/2”. 另外，转换精确度还可以用最大误差与满量程输出电压之比的百分数来表示。

转换精度是一个综合指标，包含零点误差、增益误差和非线性误差。

1. 平移误差：可能是因为求和放大器的零点没有校准，又叫零点误差
2. 斜率误差：可能是基准电压不准确或是求和放大器的增益不准确，又叫增益误差
3. 非线性误差：实际参考线是一条曲线。一般只能通过选择高质量器件、提高组装质量来减小非线性误差

9.3.3 转换速度

通常用从最小数据输入（全零）变化到最大数据输入（全一）时，输出电压达到稳态值的 $\pm(1/2)LSB$ 所花的时间来衡量。

9.4 模数转换的基本原理

9.4.1 采样定理

采样脉冲的频率 f_s 至少大于模拟信号最高频率 f_{max} 的两倍。$f_s > 2f_{max}$。实际上采样频率一般远远大于信号最高频率，这样可以更好的保留信号的频率特性，将来可以通过低通滤波处理恢复出原始信号。

9.4.2 模数转换过程

模数转换一般分为 4 步：采样、保持、量化和编码。

采样、保持　　用一个开关、一个采样电容和一个运算放大器组成。运算放大器处在电压跟随器状态，保持高输入阻抗，防止采样电容放电。**量化、编码**　　按照舍零取整或四舍五入原则，将连续采值的采样信号变成离散化基准电压刻度值，这个过程叫做量化；采样信息量化后，幅度取值就变成了最小量化阶梯的整数倍，量化后的信号还要经过编码电路，转变成二进制数字信号，这个过程叫做编码。

9.4.3 量化误差

模数转换后的数字信号和取样保持得到的模拟信号的幅度是有差异的，这种误差叫做量化误差或数字化误差。这是一种固有误差，无法消除。对于舍零取整的量化方式，最大量化误差 $\epsilon_{max} = S$; 对于四舍五入的量化方式，最大量化误差 $|\epsilon_{max}| = 0.5S$

9.5 几种常见的模数转换方案

1. 直接法：通过一套基准电压与取样保持电压进行比较，从而直接转换成数字量。特点是工作速度快，转换精度容易保证，校准方便。比如并行比较型 ADC 和逐次逼近型 ADC

2. 间接法；将取样后的模拟信号先转换成时间 t 或频率 f，再将其转换成数字量。特点是工作速度慢，但转换精度可以很高，且抗干扰能力强。一般用在测试仪表中，比如双积分型 ADC。

9.5.1 并行比较型 ADC

n 位转换电路由 2^n 个 R 的基准电阻分压网络、电压比较器、数据缓冲寄存器和编码器组成。编码要处理的码是温度计码。

优点是速度快，整个转换过程只要一个时钟周期。又叫 Flash 型模数转换器。缺点是随着分辨率的提高，元件数和复杂程度几何式增长，成本昂贵。一般用在雷达和视频转换场景。

9.5.2 分级并行比较型 ADC

又叫半并行或 half-flash ADC. 先编码 $n/2$ 位高位信号，将高位数字信号对应的模拟量减去，放大 $n/2$ 倍，再编码低位剩下的数字信号。

9.5.3 逐次逼近型 ADC

是并行比较型 ADC 进一步改变，每次只实现一位数模转换。对 n 位模数转换，需要 (n+1) 个时钟周期（多的一个周期用于输出）

9.5.4 双积分型 ADC

基本思想：将待转换电压和参考电压分别进行两次积分，将电压幅度转换位与幅度正比的时间，然后利用脉冲计数器对时间进行测量，将计数器的结果作为模数转换的数字量输出。

该电路的组成成分有：积分器、过零比较器、计数器、输入选择开关 S_1、门控开关 S_2 和控制电路。积分器是双积分型模数转化的核心器件，可以采用集成运算放大器和电阻电容来构成。

工作原理如下：

1. 准备期：先将计数器清零，将积分器电容上的残余电荷释放掉。
2. 取样期：先接待转换信号 v_{IA}，积分器的输出电压与输入电压的关系为 $v_B = -\frac{v_{IA}}{RC}t$。理想情况下，一开始积分，过零比较器就跳变，使计数器开始计数，直到计数器满量程。所以这个过程又叫定时积分。
3. 转换期：计数器满量程后出现进位信号，用这个信号将输入信号切换到参考电压 $-V_{REF}$，此后积分器的输出电压为 $v_B = -\frac{v_{IA}}{RC} \times 2^n T_0 + \frac{V_{REF}}{RC}t$。当输出过零时，计数器被终止。此阶段用时 $t = \frac{v_{IA}}{V_{REF}} \times 2^n T_0$，得到的数据为 $N = \frac{t}{T_0} = \frac{v_{IA}}{V_{REF}} \times 2^n$,$N$ 就是模数转换的结果。

注意：参考电压要取负值，且其绝对值要大于待转换电压。

这种方案转换速度低，但是抗（周期性）干扰能力强，精度高且成本低，常用于万用表等低速率数字测量仪器。

9.6 模数转换的计数指标

9.6.1 分辨率

$$R_{es} = \frac{S}{V_{max}} = \frac{S}{2^n S} = \frac{1}{2^n} \tag{38}$$

9.6.2 转换精度

同 DAC

9.6.3 转换速度

同 DAC