宁波大学
NINGBO UNIVERSITY

《单片机原理及应用》第十三讲：数模转换系统

- **1. 转换原理**
  - ➤ D/A转换器的原理很简单，可以总结为“按权展开，然后相加”几个字。
  - ➤ D/A转换器内部必须有一个解码网络，以实现按权值分别进行D/A转换。
  - ➤ 解码网络通常有两种：

    二进制加权电阻网络

    T型电阻网络

转换原理

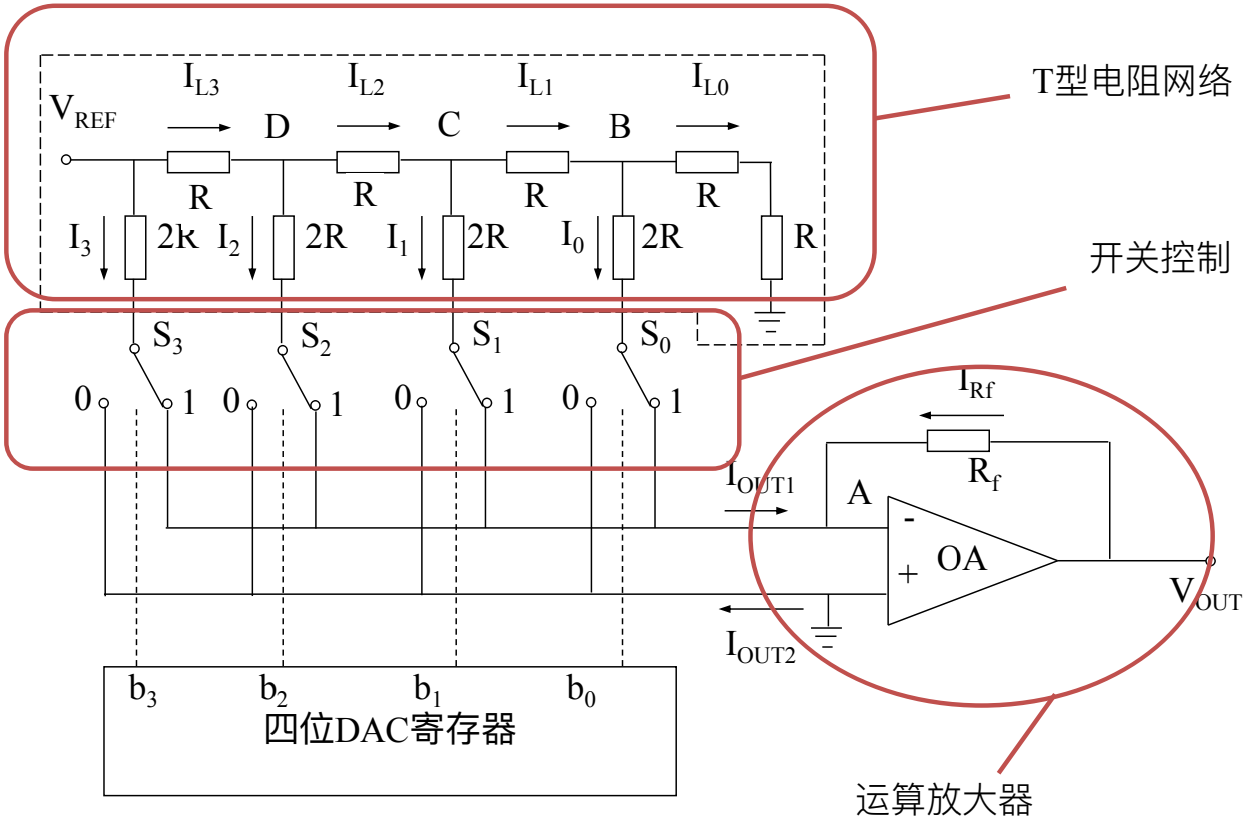
数字量 → 按权相加 → 模拟量

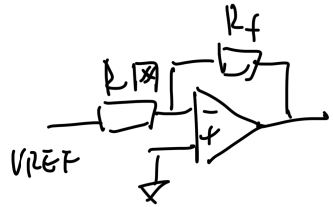
1101B

$= 1\times2^3+1\times2^2+0\times2^1+1\times2^0$

$= 13$

T型电阻网络D/A转换原理框图





$$\therefore \frac{V_{REF}-0}{R}=\frac{0-V_{out}}{R_f}$$

$$\Rightarrow V_{out}=-\frac{R_f}{R}\cdot V_{REF}$$

- 由图可得$I_0 = I_{L0}$，根据基尔霍夫电流定律：$I_{L1} = I_{L0} + I_0$ ，
- 分析C点到A点的电阻及到地线的电阻可得
- $I_1 = I_{L1}$，则$I_0 = 1/2\ I_1$，
- 同理可推得$I_1 = 1/2\ I_2$、$I_2 = 1/2\ I_3$，所以可得如下关系：

$$I_3 = \frac{V_{REF}}{2R} = 2^3 \times \frac{V_{REF}}{2^4 \times R}$$

$$I_2 = \frac{I_3}{2} = 2^2 \times \frac{V_{REF}}{2^4 \times R}$$

$$I_1 = \frac{I_2}{2} = 2^1 \times \frac{V_{REF}}{2^4 \times R}$$

$$I_0 = \frac{I_1}{2} = 2^0 \times \frac{V_{REF}}{2^4 \times R}$$

- 事实上，S_3～S_0的状态是受$b_3b_2b_1b_0$控制的，并不一定是全“1”，所以流入A点的电流应该是：

$$I_{OUT1} = b_3 I_3 + b_2 I_2 + b_1 I_1 + b_0 I_0 = (b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0)\frac{V_{REF}}{2^4 R}$$

选取Rf＝R，并考虑A点为虚地，则有 $I_{Rf} = -I_{out1}$

可以得到：

$$V_{OUT} = I_{Rf} R_f = -(b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0)\frac{V_{REF}}{2^4 R} R_f = -B\frac{V_{REF}}{16}$$

对于n位T型电阻网络

$$V_{OUT} = -(b_{n-1} 2^{n-1} + b_{n-2} 2^{n-2} + \ldots + b_1 2^1 + b_0 2^0)\frac{V_{REF}}{2^n R} R_f = -B\frac{V_{REF}}{2^n}$$

B为一个二进制数，T型电阻网络的D/A转换输出电压量绝对值与该二进制数的大小成正比

性能指标

- 主要的性能指标有4条：

（1） 分辨率（resolution）

指D/A转换器能分辨的最小输出模拟增量，为满量程值的2^{-n}倍。例如，满量程为10V的8位D/A芯片的分辨率为$10V\times2^{-8}$＝39mV；而16位的D/A是$10V\times2^{-16}$＝153μV。

（2）转换精度（conversion accuracy）

转换精度是指满量程时D/A的实际模拟输出值和理论值的接近程度。例如，满量程时理论输出值为10V，实际输出值是在9.99～10.01之间，则其转换精度为±10mV。通常为LSB/2。LSB (Least Significant Bit)是分辨率，指最低1位数字变化引起输出电压幅度的变化量。

（3）偏移量误差（offset error）

偏移量误差是指输入数字量为零时，输出模拟量对零的偏移值。这种误差通常可以通过D/A转换器的外接VREF和电位器加以调整。

（4）线性度（linearity）

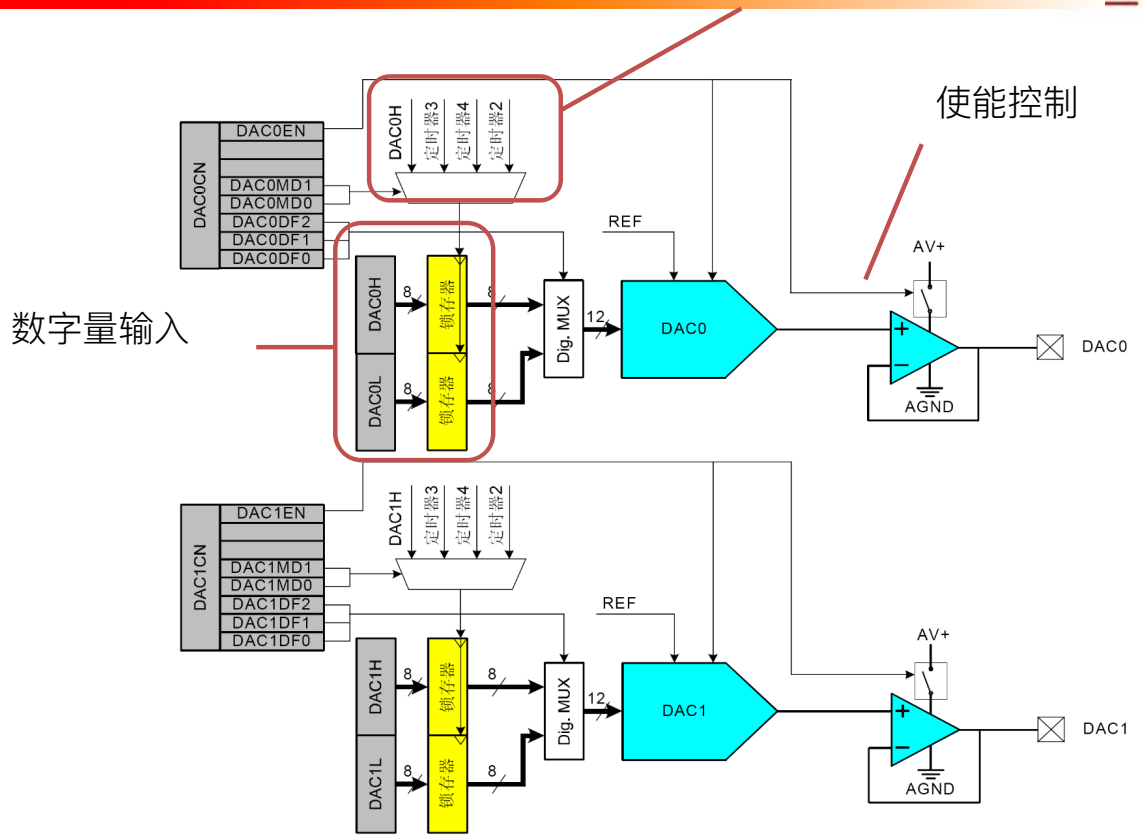
线性度是指D/A转换器的实际转换特性曲线和理想直线之间的最大偏差。通常线性度不应超出±1/2LSB。

- 除此以外，指标还有转换速度、温度灵敏度等，通常这些参数都很小，一般不予考虑。

12.2 C8051F020的DAC功能

- C8051F020 单片机有两个片内12 位电压方式数/模转换器（DAC）。
- 每个DAC的输出摆幅均为0V 到（VREF-1LSB），对应的输入码范围是0x000 到0xFFF。
- 控制寄存器DAC0CN 和DAC1CN 使能/禁止DAC0 和DAC1。
- 在被禁止时，DAC 的输出保持在高阻状态，DAC 的供电电流降到1μA 或更小。
- 每个DAC 的电压基准在VREFD引脚提供。如果使用内部电压基准，为了使DAC 输出有效，该基准必须被使能。

DAC 功能框图


输出更新方式
使能控制
数字量输入
DAC0CN
DAC0EN
DAC0MD1
DAC0MD0
DAC0DF2
DAC0DF1
DAC0DF0
DAC0H
定时器3
定时器4
定时器2
REF
AV+
DAC0H
DAC0L
8
锁存器
Dig. MUX
12
DAC0
AGND
DAC0
DAC1CN
DAC1EN
DAC1MD1
DAC1MD0
DAC1DF2
DAC1DF1
DAC1DF0
DAC1H
DAC1L
DAC1
AGND
DAC1

控制寄存器DAC0CN

- 控制DAC工作的主要是控制寄存器DAC0CN和DAC1CN，两个SFR分别控制DAC0和DAC1，以

位7	位6	位5	位4	位3	位2	位1	位0	复位值
DAC0EN	-	-	DAC0MD1	DAC0MD0	DAC0DF2	DAC0DF1	DAC0DF0	00000000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	地址：0xD4

位7： DAC0EN：DAC0 使能位

位4–3： DAC0MD1–0：DAC0 方式位。

00：DAC 输出更新发生在写DAC0H 时。

01：DAC 输出更新发生在定时器3 溢出时。

10：DAC 输出更新发生在定时器4 溢出时。

11：DAC 输出更新发生在定时器2 溢出时。

位2–0: DAC0DF2–0：DAC0 数据格式位。

000：DAC1 数据字的高 4 位在 DAC1H[3:0]，低字节在 DAC1L 中。

DAC1H								DAC1L							
				MSB											LSB

001：DAC1 数据字的高 5 位在 DAC1H[4:0]，低 7 位在 DAC1L[7:1]。

DAC1H								DAC1L							
			MSB											LSB	

010：DAC1 数据字的高 6 位在 DAC1H[5:0]，低 6 位在 DAC1L[7:2]。

DAC1H								DAC1L							
		MSB											LSB		

011：DAC1 数据字的高 7 位在 DAC1H[6:0]，低 5 位在 DAC1L[7:3]。

DAC1H								DAC1L							
	MSB											LSB			

1xx：DAC1 数据字的高字节在 DAC1H 中，低 4 位在 DAC1L[7:4]。

DAC1H								DAC1L							
MSB											LSB				

12.3 DAC 输出更新

- 每个DAC 都具有灵活的输出更新机制，允许全量程内平滑变化并支持无抖动输出更新，适合于波形发生器应用。
- **1．根据软件命令更新输出**

  在缺省方式下（DAC0CN.[4:3] =‘00’），DAC0 的输出在写DAC0 数据寄存器高字节（DAC0H）时更新。

  注意：写DAC0L 时数据被保持，对DAC0 输出没有影响，直到对DAC0H的写操作发生。
- **2．基于定时器溢出的输出更新**

  当DAC0MD位（DAC0CN.[4:3]）被设置为‘01’、‘10’或‘11’时（分别为定时器3、定时器4 或定时器2），对DAC 数据寄存器的写操作被保持，直到相应的定时器溢出事件发生时DAC0H:DAC0L 的内容才被复制到DAC 输入锁存器，允许DAC 数据改变为新值。

12.4. DAC 输出定标/调整

- 对DAC0 进行写入操作之前应对输入数据移位，以正确调整DAC 输入寄存器中的数据。
- 这种操作一般需要一个或多个装入和移位指令，因而增加软件开销和降低DAC的数据通过率。
- 数据格式化功能为用户提供了一种能对数据寄存器DAC0H 和DAC0L 中的数据格式编程的手段。
- 三个DAC0DF 位（DAC0CN.[2:0]）允许用户在5 种数据字格式指定一种，见DAC0CN 寄存器定义。

12.5 数模转换举例

- D/A转换器的编程相对A/D转换器要简单。
- 按照要求设置好输出更新的条件，将要转换的数值量送到DAC数据寄存器就行。

- 下面是产生（1）阶梯波和（2）锯齿波的示例。
- （1）阶梯波：将DAC0设置成输出更新发生在写DAC0H 时，即直接更新。
- （2）锯齿波：将DAC1设置成输出更新发生在定时器2 溢出时。

例（1）产生阶梯波

宁波大学 NINGBO UNIVERSITY

```
// DAC0用程序更新输出,产生一个阶梯波形。
sfr16 DAC0 = 0xd2;
#define UP 0x010
#define T=1000
void main(void)
{
   int i;
   WATCHDOG_init(); //禁止看门狗
   SYSCLK_init();  //外部晶振初始化
   DAC0_init ();  //DAC0初始化
   for(i=0;i<=4095;i+UP)     //形成阶梯波形
   {
      DAC0=i;                          //送数字量到DAC0直接更新输出
      d1ms(T);                          //软件延迟
   }
}
```

看门狗、时钟、延迟函数

宁波大学 NINGBO UNIVERSITY

```
void WATCHDOG_init(); // 禁止看门狗
{
   EA = 0;
   WDTCN = 0xDE;
   WDTCN = 0xAD;
   EA = 1;
}
void SYSCLK_init(void)                    //外部晶振初始化
{
   OSCXCN=0x67;
   for (n=0; n<255;n++)
   while(!(OSCXCN & 0x80));
   OSCICN=0x88;
}
void d1ms(int count)                      //软件延迟
{   int j;
   while(count--!=0)
   { for (j=0;j<100;j++); }
```

DAC0初始化程序

宁波大学 NINGBO UNIVERSITY

```
void DAC0_init (void)                    //DAC0初始化
{
   REF0CN = 0x03;                        // 内部偏压发生器工作
   DAC0CN = 0x80;                        // 允许DAC0，程序直接更新输出
                                         // 数据右对齐
    DAC0L = 0x00;                        // DAC0数据寄存器初值
    DAC0H = 0x00;
}
```

(2) 产生锯齿波


```
sfr16 T2 = 0xcc;
Sfr16 DAC1=0xd5;

void main(void)
{
   WATCHDOG_init();  //禁止看门狗
   SYSCLK_init();       //外部晶振初始化
   DAC1_init ();          //DAC1初始化
   Timer2_init();
   while(1);
}
```

看门狗、时钟


```
void WATCHDOG_init(); // 禁止看门狗
{
   EA = 0;
   WDTCN = 0xDE;
   WDTCN = 0xAD;
   EA = 1;
}
void SYSCLK_init(void)                    //外部晶振初始化
{
   OSCXCN=0x67;
   for (n=0; n<255;n++)
   while(!(OSCXCN & 0x80));
   OSCICN=0x88;
}
```

DAC1初始化

宁波大学 NINGBO UNIVERSITY

```
void DAC0_init (void)                    //DAC0初始化
{
   REF0CN = 0x03;                        // 内部偏压发生器工作
   DAC1CN = 0x98;                        // 允许DAC1，T2溢出中中断更新
                                         // 数据右对齐
    DAC1L = 0x00;                        // DAC0数据寄存器初值
    DAC1H = 0x00;
}
```

T2初始化


```
void Timer2_init(void)
{
   RCAP2H = 0x05;                  // 重新装入的时间常数
   RCAP2L = 0x00;
   TH2 = 0x05;                     //初始值
   TL2 = 0x00;
   T2CON = 0x04;                   // 启动T2
   IE = 0x20;                      // T2中断允许
}
```

$\frac{1.024}{2.048} = \frac{x}{4096} \Rightarrow x = 2047$

$\frac{1}{2} = \frac{x}{4095}$

T2中断程序


```
void T2_ISR() interrupt 5
{
        TF2=0;                  //清中断标志
        DAC1++;                 //因为是T2溢出更新DAC1输出，
                                //所以可以对SFR16操作，
                                //此时并不立即更新
        if(DAC1>=0x1000)        //形成锯齿波
                DAC1=0;
}
```

0x0FFF

if[DAC1>=0x0D2B] → 2V需求.

3371

$\frac{2^{13}}{2.4}$

∴定时器要定

$\frac{1}{3371} \simeq 30\mu s$ → 确定初值.

思考题

- 例（2）锯齿波：将DAC1设置成输出更新发生在定时器2 溢出时。

- 增加下列约束条件：
- （1）锯齿波的周期为100ms；→定时器计时周、采样频率。
- （2）锯齿波的最大电压为2V；$\frac{2}{V_{REF}} = \frac{N}{2^{12}-1}$
- 假设时钟为24MHz，使用内部参考电压（2.4V）
- 请问上述程序怎么修改?