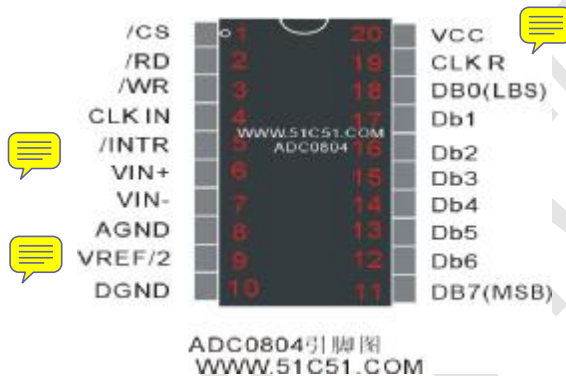


模拟/数字转换器（相关知识）



A/D 转换器的基本原理----ADC0804

- 1, 所谓 A/D 转换器就是模拟/数字转换器 (ADC), 是将输入的模拟信号转换成数字信号。信号输入端可以是传感器或转换器的输出, 而 ADC 的数字信号也可能提供给微处理器, 以便广泛地应用。
- 2, ADC0804 的规格及引脚图
8 位 COMS 依次逼近型的 A/D 转换器。
三态锁定输出
存取时间: 135US
分辨率: 8 位
转换时间: 100US
总误差: 正负 1LSB
工作温度: ADC0804LCN---0~70 度
引脚图及说明见图 1



ADC0804引脚图
WWW.51C51.COM

/CS 芯片选择信号。

/RD 外部读取转换结果的控制输出信号。/RD 为 HI 时, DB0-DB7 处理高阻抗; /RD 为 LO 时, 数字数据才会输出。

/WR: 用来启动转换的控制输入, 相当于 ADC 的转换开始 (/CS=0 时), 当 /WR 由 HI 变为 LO 时, 转换器被清除; 当 /WR 回到 HI 时, 转换正式开始。

CLK IN, CLK R: 时钟输入或接振荡无件 (R, C) 频率约限制在 100KHZ~1460KHZ, 如果使用 RC 电路则其振荡频率为 $1/(1.1RC)$

/INTR: 中断请求信号输出, 低电平动作。

VIN(+) VIN(-): 差动模拟电压输入。输入单端正电压时, VIN(-) 接地; 而差动输入时, 直接加入 VIN(+), VIN(-)。

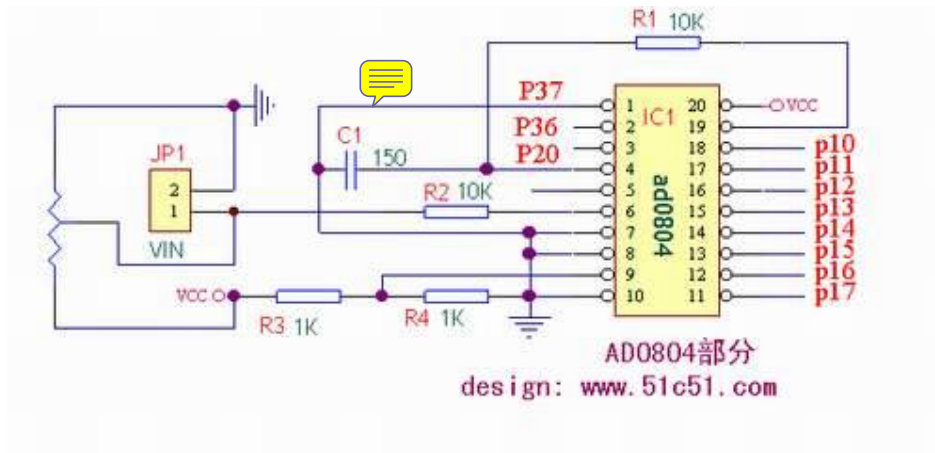
AGND, DGND: 模拟信号以及数字信号的接地。

VREF: 辅助参考电压。

DB0-DB7: 8 位的数字输出。

VCC: 电源供应以及作为电路的参考电压。

参考腾龙套件的 0804 原理图:



十六进制	二进制码	与满刻度的比率		相对电压值 VREF=2.560 伏	
		高四位字节	低四位字节	高四位电压	低四位电压
F	1111	15/16	15/256	4.800	0.300
E	1110	14/16	14/256	4.480	0.280
D	1101	13/16	13/256	4.160	0.260
C	1100	12/16	12/256	3.840	0.240.
B	1011	11/16	11/256	3.520	0.220.
A	1010	10/16	10/256	3.200	0.200
9	1001	9/16	9/256	2.880	0.180
8	1000	8/16	8/256	2.560	0.160
7	0111	7/16	7/256	2.240	0.140.
6	0110	6/16	6/256	1.920	0.120.
5	0101	5/16	5/256	1.600	0.100
4	0100	4/16	4/256	1.280	0.080
3	0011	3/16	3/256	0.960	0.060
2	0010	2/16	2/256	0.640	0.040.
1	0001	1/16	1/256	0.320	0.020.
0	0000			0	0

例:VIN=3V,由上表可知 $2.880 + 0.120 = 3V$ 为 10010110=96H

功能说明

1,ADC0804 将输入模拟值转换成数字值输出到 P0,使相对应的 LED 亮.如输入 3V,ADC0804 的输出应为

96H=10010110,此数字信号送入 8051 的 P1,再由 P1 存入 8051 的累加器,然后累加器再到 P0,使相应的 LED

亮。

2, 先将 ADC0804 的参考电压 VREF 调整为 2.56V。(在腾龙套件中主要演示原理, 未作此精确调整电压, 用 2 个 1K 电阻分压, 约 2.5V)

3 调整 ADC0804 的 VIN 可变电阻器. 由 0V 调到 5V 根据其关系观察 P1 的 LED 变化情形。

参考程序: ;0804 的基本应用, 转动电位器, P0 口显示取到的数

```
//定义 ADC 的连接端口
ad_cs equ P3.6
ad_wr equ P2.0
ad_rd equ P3.7
ad_input_port equ p1
```

```
org 0000h
ajmp main
org 0030h
main:
    lcall adc_demo
    ajmp main
```

```
//=====
// 启动 AD 转换
//=====
Adc_Start:
```

```
    clr ad_cs
    nop
    clr ad_wr
    nop
    setb ad_wr
    nop
    setb ad_cs
    nop
    ret
```

```
//=====
// 读 AD 转换
//=====
```

```
Adc_Read:
    mov ad_input_port,#0ffh
    clr ad_cs
    nop
    clr ad_rd
    nop
    nop
    mov a,AD_INPUT_PORT
```

```
    nop
    setb    ad_rd
    nop
    setb    ad_cs
    ret

;//=====================================================
;// AD 转换读取延时程序，显示读到的数值
;//=====================================================
Adc_Demo:
    lcall  Adc_Start
    lcall  delay1ms
    lcall  adc_read
    clr   p2.0
    clr   p2.3
    mov   p0,a
    ret

delay1ms:
    mov   r7,#10
tt1:
    mov   r6,#50
    djnz  r6,$      ;2us
    djnz  r7,tt1
    ret
    end
```