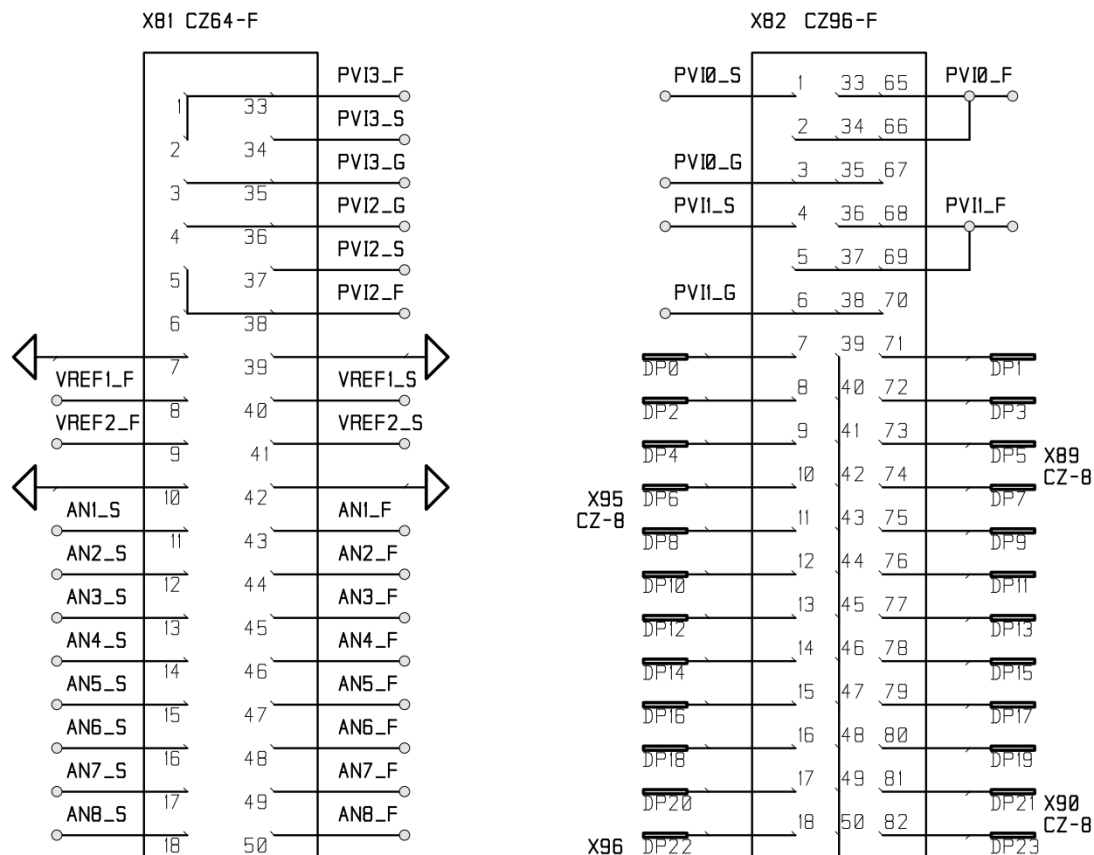


Make Testing More Valuable

AD/DA直流参数的测试

• 适配器资源:



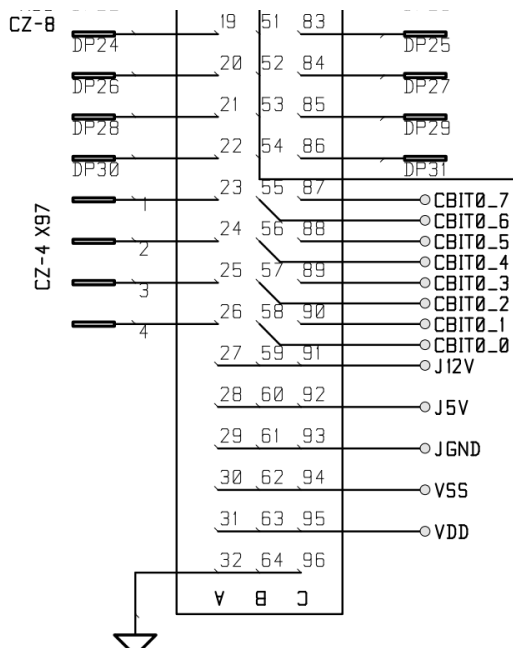
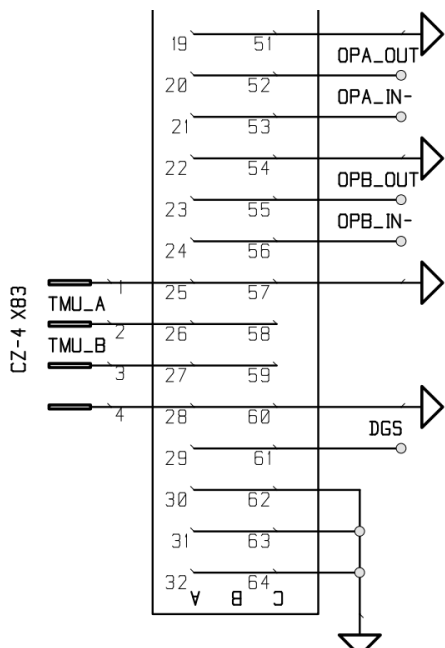
FPVI0-3 器件供电电源

VREF1-2 器件基准电压

AN1-8 AD的模拟输入
/DA的模拟输出及
基准电流的测试

DP0-31 器件数字管脚

• 适配器资源:



OP	电流型DA的电流 信号转换为电压
TMU	时间测量单元
CBIT0-7	继电器控制位
J5V	继电器电源

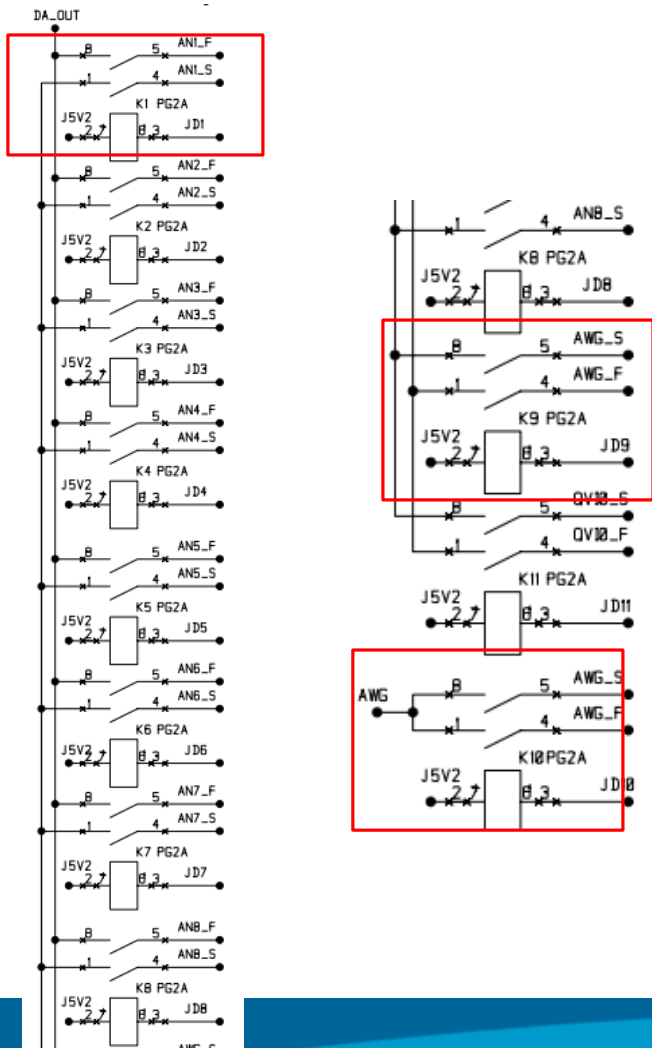
- 通用适配器



• 类别板提供的资源：

板卡	资源	用途	指标
FPVI (主机柜)	FPVI0-3	为被测 AD/DA 的供电电源， 测试芯片的电源电流等	$\pm 1V \sim \pm 40V$, $\pm 100\mu A \sim \pm 10A$
AWG (测试头)	VREF1-2	为被测器件提供外部基准	输出电压： $\pm 10V$ ， 负载能力： $\pm 20mA$ 。
	AWG	用作 AD 器件的模拟输入信号， 在 DA 器件的模拟输出信号测量中做差分比较的输入电压。	输出电压： $\pm 10V$ ， 负载能力： $\pm 20mA$ 。
DCM (测试头)	DCM	连接 AD/DA 的数字通道，提供被测器件数字管脚上的激励， 捕捉 AD 器件的响应。 测试所有数字管脚的 V_o 、 I_i 、 I_o 。	速率：10MHZ 。向量深度：65536 per pin。失效存储器深度：65536 per pin
QVM (主机柜)	QVM	DA 器件的模拟输出信号差分比较后的测量	电压测量范围： $\pm 10V$
FOVI (主机柜)	FOVI8	输入阻抗的测量等	$\pm 1V \sim \pm 40V$, $\pm 100\mu A \sim \pm 1A$

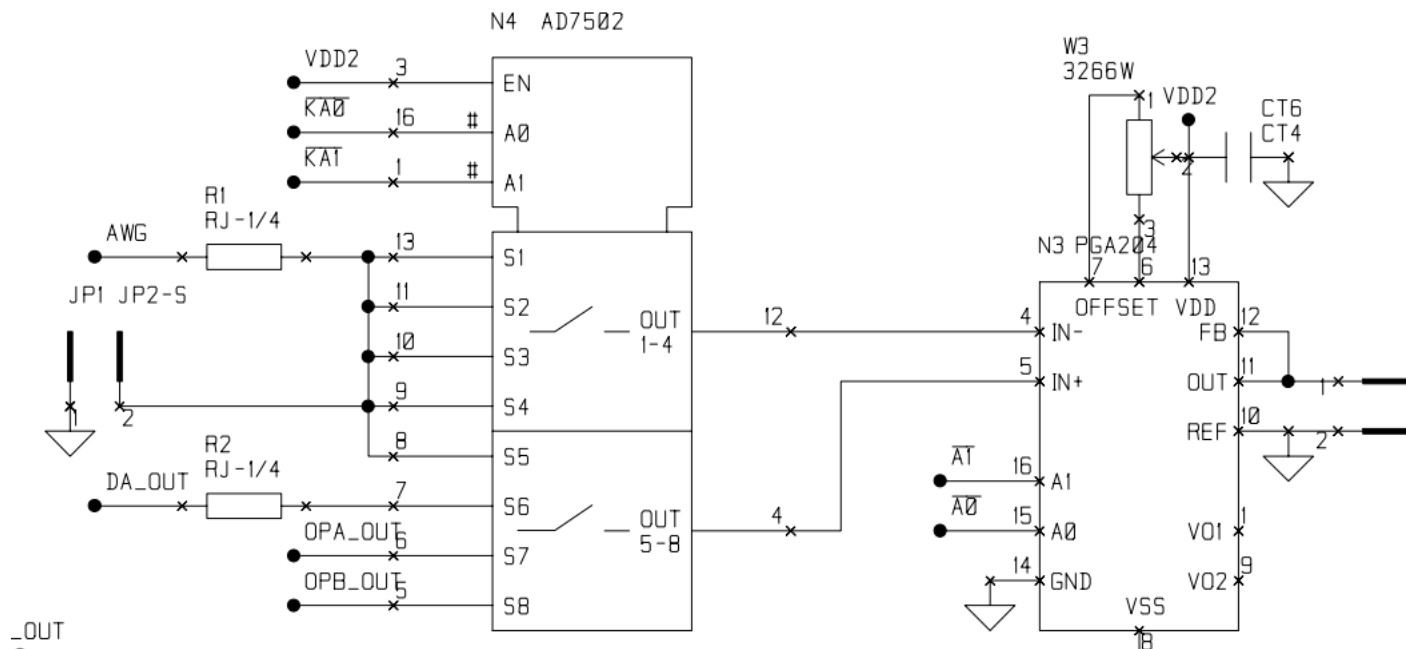
• 类别板:模拟资源的引入引出



如图：对**AD**器件测试，需要闭合**K9**和对应的**AN**继电器就可以把模拟信号接到适配器上。

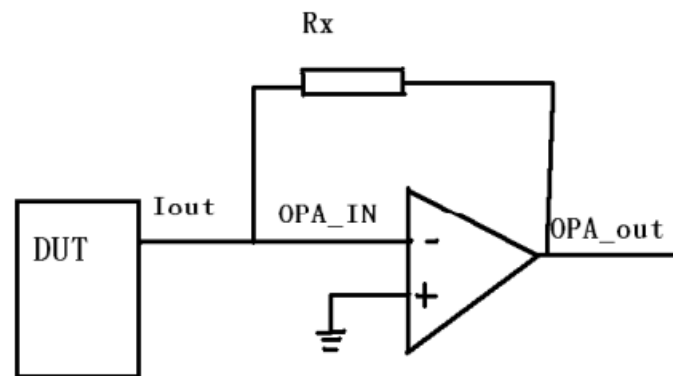
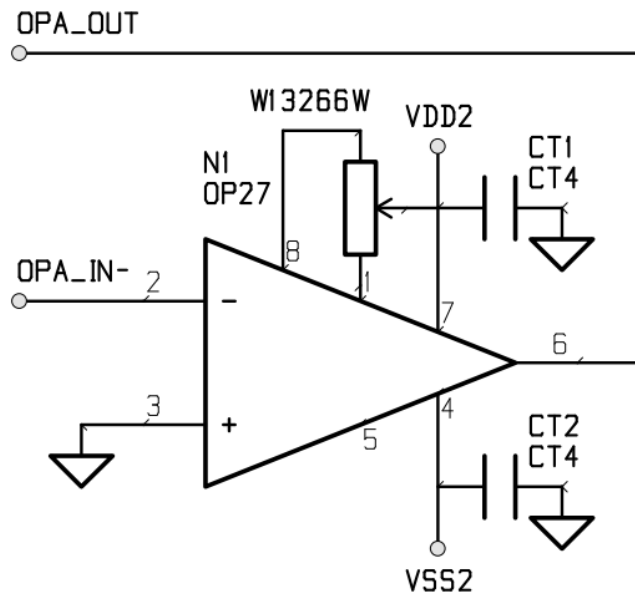
对于**DA**器件需要闭合对应的**AN**继电器将模拟信号接到**DA_OUT**，同时需要闭合**K10**，将**AWG**的输出信号接到差分信号的输入端。

• 类别板:DA模拟信号的测量



DA的模拟输出是通过差分比较的方式进行测量的，**AWG**为理想的模拟量，测量前会**N4**通过**KA0**和**KA1**选择要比较的信号，**DA_OUT**为电压型DA的输出，**OPA_OUT**和**OPB_OUT**为电流型DA的输出。信号会在**N3**出进行比较和放大，放大倍数通过**A1**和**A0**控制，放大倍数为**10**、**100**、**1000**。最后比较后的信号接到**QVM**进行测量。

- 类别板: 电流型DA信号的转换



类别板上提供两路放大器，用于将电流型**DA**的输出电流转换为电压值。有图为其典型应用图，**IN-**和**OUT**端都已经接到了适配器上，我们只需选择对应的电阻焊在两端即可，注意此处的**R_x**一定选用精密电阻。

• 类别板继电器控制表

ADDA 类别板 CBIT 信号控制位：

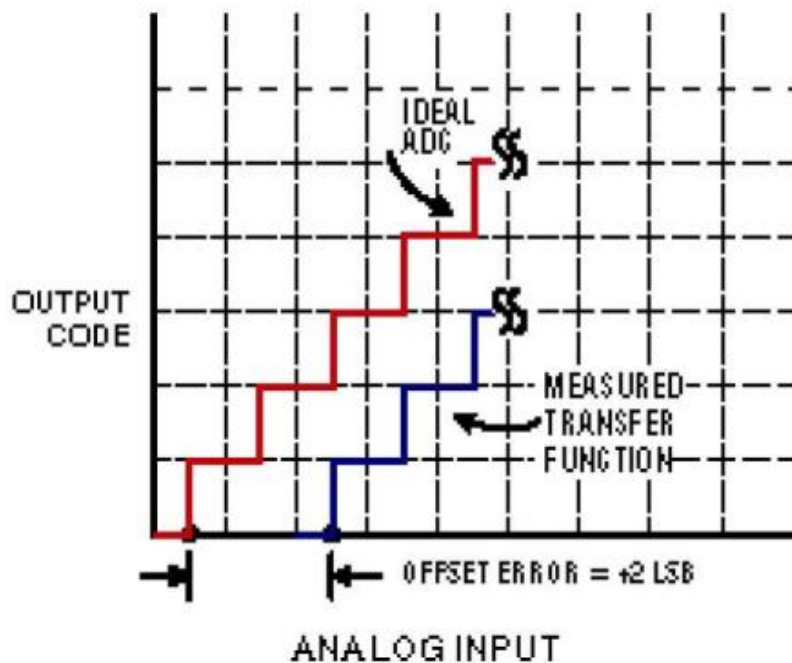
信号名称	CBIT 位	作用
JD1	56	闭合继电器 K1，使信号 AN1-F\S 连到信号 DA-OUT
JD2	57	闭合继电器 K2，使信号 AN2-F\S 连到信号 DA-OUT
JD3	58	闭合继电器 K3，使信号 AN3-F\S 连到信号 DA-OUT
JD4	59	闭合继电器 K4，使信号 AN4-F\S 连到信号 DA-OUT
JD5	60	闭合继电器 K5，使信号 AN5-F\S 连到信号 DA-OUT
JD6	61	闭合继电器 K6，使信号 AN6-F\S 连到信号 DA-OUT
JD7	62	闭合继电器 K7，使信号 AN7-F\S 连到信号 DA-OUT
JD8	63	闭合继电器 K8，使信号 AN8-F\S 连到信号 DA-OUT
JD9	43	闭合继电器 K9，使信号 AWG-F\S 连到信号 DA-OUT
JD10	42	闭合继电器 K10，使信号 AWG-F\S 连到信号 AWG
JD11	41	闭合继电器 K11，使信号 QVI0-F\S 连到信号 DA-OUT
A0	44	控制精密差分放大测量中的放大倍数
A1	45	
KA0	46	
KA1	47	控制精密差分放大测量中的工作模式

• AD器件的测试

A/D 转换器参数符号及名称

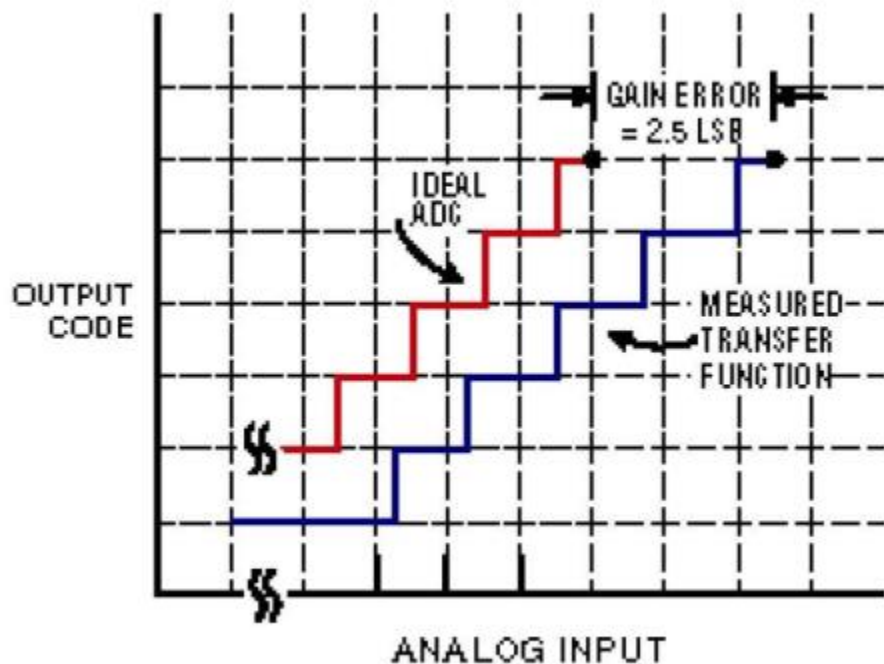
参数名称	参数符号	测试范围
基准电压	V_{ref}	-10V - +10V
输入阻抗	R_{IN}	
电源电流	I_{CC} 、 I_{EE} 、 I_{LOGIC}	0 - 1A
输入电流	I_{IH} 、 I_{IL}	0 – 100uA
高阻电流	I_{OZH} 、 I_{OZL}	0 – 100uA
输出电平	V_{OH} 、 V_{OL}	0 - 40V
零点误差	E_Z	12A/D、D/A
增益误差	E_G	12A/D、D/A
线性误差	DNL、INL	12A/D、D/A

- 误差参数



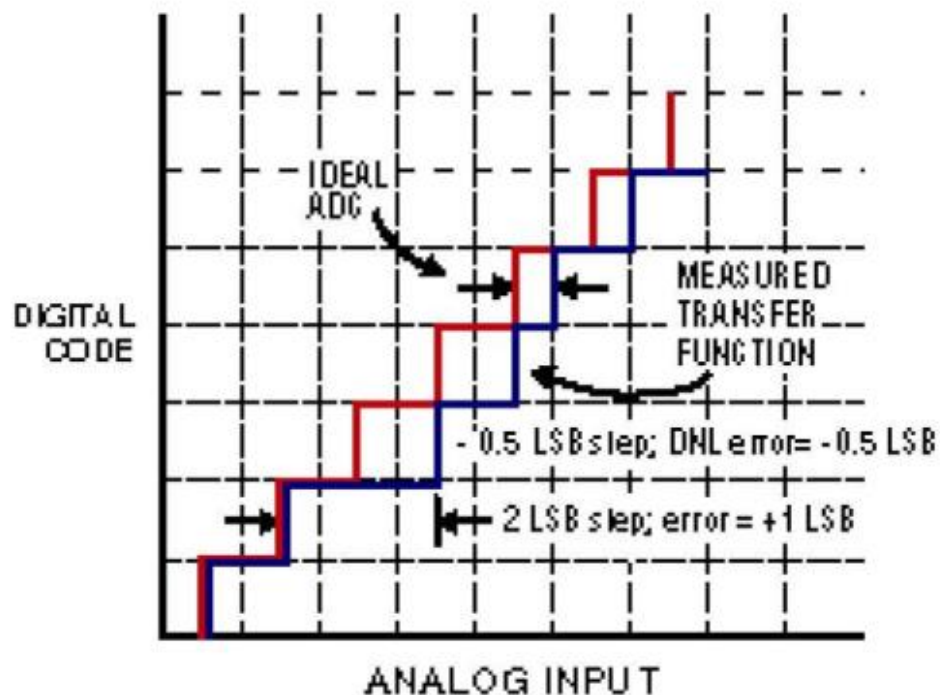
零点误差又称输入失调，是实际模数转换曲线中数字 **0** 的代码中点与理想模数转换曲线中数字 **0** 的代码中点的最大误差，记作 **EZ**。即当首个数字代码输出时的输入电压减去 $1/2\text{LSB}$ 再减去理想的 **0** 刻度值。

- 误差参数



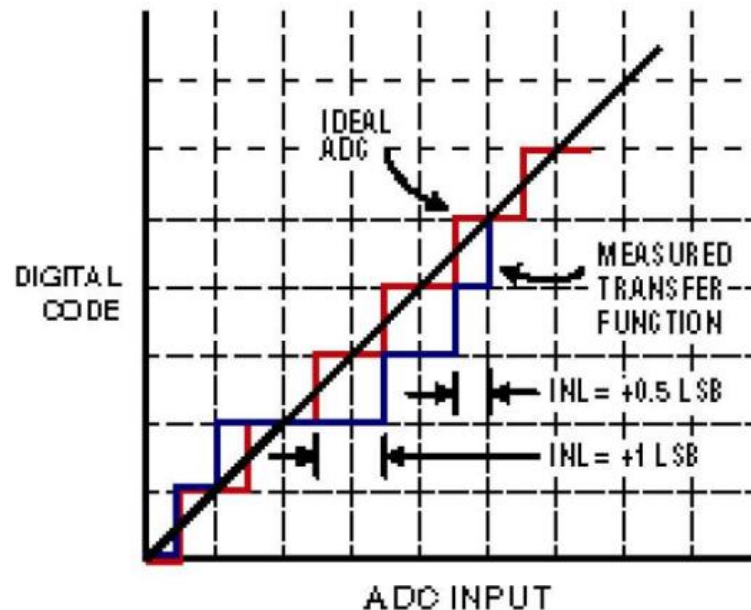
增益误差是指转换特性曲线的实际斜率与理想斜率之间的偏差，记作 **EG**。即当补偿失调偏置之后，输出为全 1 时对应的实际输入电压与理想输入电压之差。

- 误差参数



差分非线性误差是实际转换特性曲线的码宽与理想码宽之间的最大偏差，记作 **DNL**。

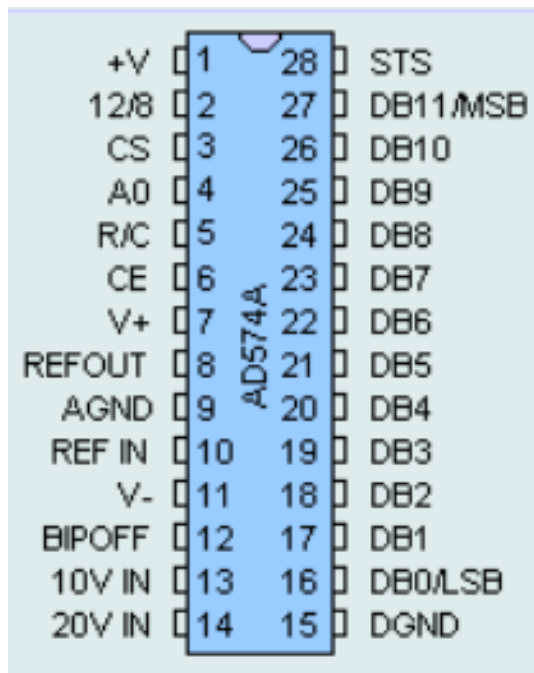
- 误差参数



积分非线性误差是指实际转换曲线与理想特性曲线之间的偏差，记作 **INL**。实际测量是测试第 j 码的代码中心值，将其与理想第 j 码的中心值比较。分别测出所有码值的误差，取其绝对值的最大值即为积分非线性误差。

• AD574的测试

AD574是一款12位A/D转换器，其管脚排列如图



PIN1	VL	数字电源+5V
PIN2	12/8	数字输出格式选择，高12位输出，低8位输出
PIN3	C/S	片选信号
PIN4	A0	字节选择控制
PIN5	R/C	读数据/转换控制信号，R/C=1，转换数据允许被读取；R/C=0允许启动A/D转换
PIN6	CE	启动转换信号，高有效。
PIN7/11	Vcc/Vee	模拟部分的正负电源
PIN8	REF OUT	10v 内部基准输出
PIN9	AGND	模拟地
PIN10	REF IN	基准电压输入端
PIN11		
PIN12	BIP OFF	补偿调整端
PIN13	10VIN	模拟量10V输入端
PIN14	20VIN	模拟量20V输入端
PIN15	DGND	数字地
PIN16-27	DB	数字量输出
PIN28	STS	输出状态信号引脚，转换时STS为高，转换完成后返回低

• AD574的测试

AD574的时序图

Table I. AD574A Truth Table

CE	\overline{CS}	R/\overline{C}	$12/\overline{8}$	A_0	Operation
0	X	X	X	X	None
X	1	X	X	X	None
1	0	0	X	0	Initiate 12-Bit Conversion
1	0	0	X	1	Initiate 8-Bit Conversion
1	0	1	Pin 1	X	Enable 12-Bit Parallel Output
1	0	1	Pin 15	0	Enable 8 Most Significant Bits
1	0	1	Pin 15	1	Enable 4 LSBs + 4 Trailing Zeroes

• AD574的测试

AD574适配器接线图：

*

FPVI2	1	VL	STS	28	DP0
FPVI2	2	12/8	DB11	27	DP1
DP14	3	\overline{CS}	DB10	26	DP2
DP15	4	A0	DB9	25	DP3
DP16	5	R/\overline{C}	DB8	24	DP4
DP17	6	CE	DB7	23	DP5
FPVI0	7	Vcc	DB6	22	DP6
AN2	8	REF_OUT	DB5	21	DP7
GND	9	AGND	DB4	20	DP8
VREF1	10	REF_IN	DB3	19	DP9
FPVI1	11	Vee	DB2	18	DP10
GND	12	BIP OFF	DB1	17	DP11
AN1	13	10Vin	DB0	16	DP12
NC	14	20Vin	DGND	15	GND

AD574

• AD574的误差测试方法

- 本系统 A/D 静态误差参数测试采用全码扫描方式，数据处理方法采用码密度直方图法。码密度直方图测试是基于数理统计论，ADC 对周期模拟输入信号进行随机采样，不同数字码输出的出现次数为码密度。以 ADC 的输出数字码及其出现次数为坐标画图，所得图像称为直方图。在直方图中，每个数字码称为码箱（Code Bin），每个数字码出现的次数称为码箱宽度。根据码密度数据，可以估算出 ADC 的静态特性参数。
- 为了得到被测器件准确的完整的模拟-数字的转换曲线，我们需要在被测器件的每个LSB电平区间内都取n个模拟电平点，n越大，得到的转换曲线约详细，测试速度越慢（当然n不能取很大，这受制与被测器件LSB和AWG源的精度）。对于12bit、0~10V输入的AD574，我们取 $n = 15$ ，代表在0-10v电平范围去平均取 15×2^{12} 个点。考虑到零点误差和增益误差的存在，为了得到完整的转换曲线，我们必须在0V以下和10V以上各多取10（自定义）LSB的点。因此对于AD574，我们一共要取 $15 \times 2^{12} + 300$ 个电平点，组成一根大约-0.026V~+10.024V的输入电平曲线。我们把这61740个点依次给进ad574内，得到61740组输出的数字量。再经过计算得到被测器件的误差值。

• AD574的向量文件

编写矢量表生成**vec**文件，矢量表的详细操作见《矢量表编程指南》

时间集定义

时间集名称: Conv T0: 35000 ns

管脚名	T1	T2	T3	管脚名
/CS	10	80	80	NRZ
A0	10	80	80	NRZ
R/C	500	1700	80	SBC
CE	10	80	80	NRZ
DB0	10	80	80	NRZ
DB1	10	80	80	NRZ
DB2	10	80	80	NRZ
DB3	10	80	80	NRZ
DB4	10	80	80	NRZ
DB5	10	80	80	NRZ
DB6	10	80	80	NRZ
DB7	10	80	80	NRZ
DB8	10	80	80	NRZ
DB9	10	80	80	NRZ
DB10	10	80	80	NRZ
DB11	10	80	80	NRZ

增加 删除 确定 退出

时间集

• AD574的向量文件

电平集设置

电平集 电平集名称

管脚名	Vih	Vil	Voh	Vol
/CS	2.5	0.5	2	0.8
A0	2.5	0.5	2	0.8
R/C	2.5	0.5	2	0.8
CE	2.5	0.5	2	0.8
DB0	2.5	0.5	2	0.8
DB1	2.5	0.5	2	0.8
DB2	2.5	0.5	2	0.8
DB3	2.5	0.5	2	0.8
DB4	2.5	0.5	2	0.8
DB5	2.5	0.5	2	0.8
DB6	2.5	0.5	2	0.8
DB7	2.5	0.5	2	0.8
DB8	2.5	0.5	2	0.8
DB9	2.5	0.5	2	0.8
DB10	2.5	0.5	2	0.8
DB11	2.5	0.5	2	0.8
STS	2.5	0.5	2	0.8

增加 删除 确定 退出

电平集

• AD574的测试

Line No	Label	Command	TSet	/CS	A0	R/C	CE	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8	DB9	DB10	DB11	STS	S4	S3	S2	S1	S0	Comment
0000		LOOP_START 61740	Conv	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	
0001			Conv	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	
0002		LOOP_END	Read	0	0	1	1	H	H	H	H	H	H	H	H	H	H	H	H	H	0	0	0	0	1	
0003			Conv	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	
0004			Conv	0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	
0005			Hi	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	
0006			Hi	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	
0007			Conv	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	Vout
0008			Hiz	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	Hiz

编写向量文件

对于AD器件在读取数据时，需要在读取数据行有一位失效位。例：AD574的向量文件中STS在装换完成后的理论电平为低，但在向量文件中写成了高，所以文件在执行这句时，会得到一个失效标志，方便程序中数据的读取。另外循环体结束时，需要在S0处写1，这是为在给AWG板卡外部触发信号，使AWG板卡按照顺序输出模拟量。

• AD574的关键代码：

DPin_t **inpin[4] = {3, 4, 5, 6};** //All input pins定义芯片所有的数字输入管脚

CVector **vec;** //定义向量表

AWG **awg0;** //定义AWG板卡

vec.SetVectorFile("AD574.vec"); //打开向量文件

vec.Load(); //装载向量文件

vec.Unload(); //卸载向量文件

awg0.SetREF(10,AWG_REF1);

awg0.REFConnect(AWG_REF1); //vref1输出设置：恒压为10v

vec.UseVoltageSet(0); //调用第一套管脚电平

vec.ClosePinRelay(allpin,SIZEOF_ARRAY(allpin)); //闭合所有数字管脚的继电器

vec.FVMI(inpin[i], QVI_VRNG_5V, 0.5, QVI_IRNG_10UA, 10e-6, -10e-6, val1, 5, 10);

• AD574的关键代码:

```
void create_awg_data(void) //计算需要测试的点数并计算每个点数需要施加的电压并存储
{
```

```
    int ucBits=12; //被测A/D的位数
    int NUM_PER_LSB=15; //每个台阶给15个点
    int EZ_ADD_POINT=150; //因为零点误差而加入的多扫描1500个点
    int EG_ADD_POINT=150; //因为零点误差而加入的多扫描150个点
    double VRang=10; //定义输出电压的范围
    int ScanPoint=0; //all scan point number
    double CodeNUM=0; //输出码值的数量
    double CodePoint=0; //均分0-10V的所有码
    CodeNUM = pow(2,ucBits); //4096个
    CodePoint=CodeNUM * NUM_PER_LSB; //4096*15=61440
    ScanPoint = (unsigned long)CodePoint + EZ_ADD_POINT + EG_ADD_POINT;
    //61440+150+150=61740, 这就是芯片测试误差参数需要的所有扫描点
    double * ScanPointVolt=new double[65536];
    assert(ScanPointVolt);
    // 生成每个扫描点电压值
    for(int i=0;i<ScanPoint;i++)
    {
        ScanPointVolt[i] = (VRang/CodePoint)*(i-150);
    }
    awg0.LoadData(ScanPointVolt, ScanPoint, AWG_VRNG_10V); //所有扫描点的电压都装载到awg0上
    awg0.LoopEnable(100, AWG_EXT, UP);
    awg0.CNTCLR();
    awg0.CNTEnable();
    awg0.Setup(AWG_VRNG_10V, AWG_Filter); //设置awg0的相应工作状态
    awg0.Connect(); //awg0输出继电器闭合
    delete[] ScanPointVolt; //清空数组
```

```
}
```

• AD574的关键代码:

```
vec.Run(0,4);//测试向量从第0拍跑到第4拍  
while(vec.Running());//等待跑完测试图形
```

```
DWORD ern = vec.GetResult();//返回测试图形的失效行数  
DWORD * pea = new DWORD[61740];//定义变量---失效行数  
BYTE * buf = new BYTE [802620];//定义变量----大于等于总的失效位数
```

```
vec.GetFailData(ern,pea,buf,outputpin,SIZEOF_ARRAY(outputpin));//加载所有的失效位数  
// ern          -----失效行数  
// pea          -----储存失效地址指针  
// buf          -----储存失效数据指针  
// outputpin    -----要读取数据的管脚  
// SIZEOF_ARRAY(outputpin) -----被读取数据管脚的总数目
```

```
for(i=0;i<(int)ern;i++)  
{  
    j=i*13;  
    for(k=0;k<12;k++)  
    {  
        mid_data[k]=(int)buf[j+k];//把总的失效位数据分离为61740（所有扫描点）组失效数据  
    }  
}
```

```
adc_data=(mid_data[11]<<11) | (mid_data[10]<<10) | (mid_data[9]<<9) | (mid_data[8]<<8) |  
(mid_data[7]<<7) | (mid_data[6]<<6) | (mid_data[5]<<5) | (mid_data[4]<<4) | (mid_data[3]<<3) |  
(mid_data[2]<<2) | (mid_data[1]<<1)|mid_data[0];    //把每组中总共13个数据中有用的12个数据提取出来并进  
行组合
```

```
data_num[adc_data]++;    //记录所有不同值的adc_data的个数
```

```
}
```

- AD574的关键代码:

```
for( i=1; i<4095; i++)  
{  
    all_data_num+=data_num[i];  
}  
double lsb=0.0;  
lsb=all_data_num/4094.0;  
  
double Ez_result;           //零点误差的计算  
Ez_result=(double)(data_num[0]-150-lsb/2.0)/lsb;  
  
double Eg_result;           //增益误差的计算  
Eg_result=(double)(300+lsb-data_num[4095]-data_num[0])/lsb;  
  
double DNL_result;  
for(i=1;i<4095;i++)        //微分误差的计算  
{  
    diff_err=data_num[i]-lsb;  
    if(fabs(diff_err)>fabs(max_err))  
        max_err=diff_err;  
}  
DNL_result=(double)max_err/lsb;
```

- AD574的关键代码:

```
double INL_result=0.0,tmp_inl=0.0; //积分误差的计算
for(i=1;i<4095;i++)
{
    tmp_inl=0.0;
    for(j=1;j<=i;j++)
    {
        tmp_inl+=data_num[j];
    }
    tmp_inl=tmp_inl-(((double)data_num[i]/2.0)+(double)lsb*(i-0.5));
    if(fabs(tmp_inl)>fabs(INL_result))
    {
        INL_result=tmp_inl;
    }
}
INL_result/=(double)lsb;
```


• AD调试过程:

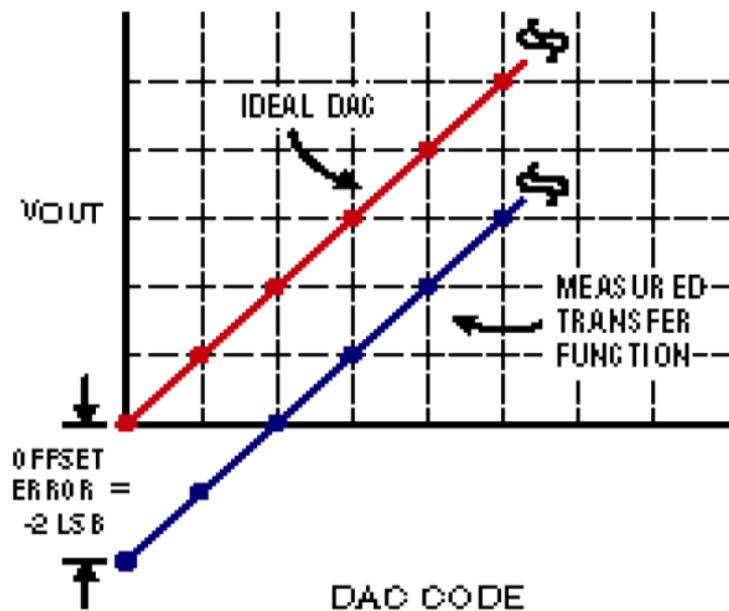
- 适配器上DGS与AGND是否短路;
- 转换时间与数字信号的输出时间是否应加长;
- 芯片供电端是否需要更改电容值;
- VREF端是否需要加电容;
- AWG输出端是否需要加电容或者串电阻;
- VREF信号的输出带载能力是否不够; 比如TLC876C, 需要加跟随器。

• DA参数的测试:

D/A 转换器参数符号及名称

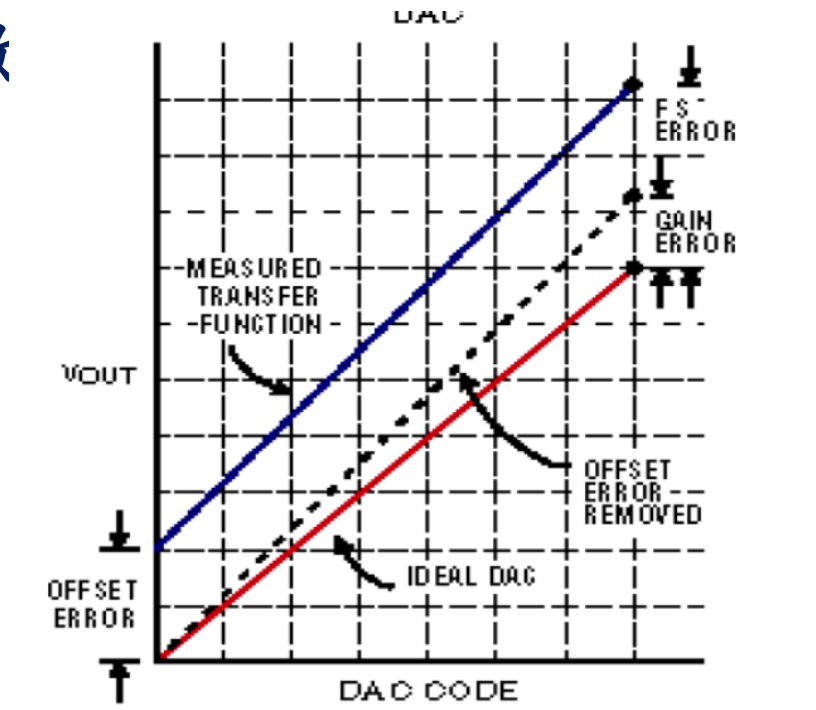
参数名称	参数符号	测试范围
输入阻抗	R_{REF}	
输入阻抗	$R_{REFMatch}$	
电源电流	I_{DD}	0 - 1A
输入电流	I_{IH} 、 I_{IL}	0 – 100uA
高阻电流	I_{OZH} 、 I_{OZL}	0 – 100uA
输出电平	V_{OH} 、 V_{OL}	0 - 40V
零点误差	E_Z	12AD、DA
增益误差	E_G	12AD、DA
线性误差	DNL、INL	12AD、DA

- 误差参数：



零点误差定义为模拟输出电压起始值与理想起始值之偏差，记作 **EZ**。当向被测器件输入数字量 **00..00** 或零电平数字信号时，理想输出电压应为零，但实际输出电压偏离 **0V**，测得实际输出与理想输出之间的差即为 **EZ**。

• 误差参数



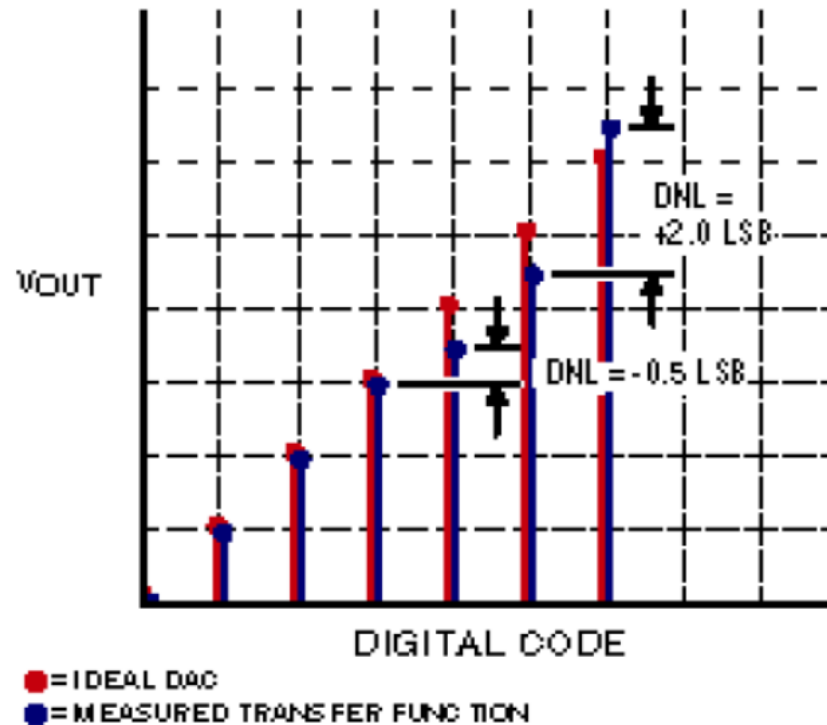
增益误差（**Gain Error**）定义为转换特性曲线的实际斜率与理想斜率之间的偏差，记作**EG**。测试时，在规定的电源电压和参考电压下，在数字输入端施加规定的逻辑电平（全**0**码）。将失调调整为零。然后在数字输入端施加规定的逻辑电平（全**1**码）在模拟输出端测得电压 **V'FSR**。则增益误差为：

$$EG = (V'FSR - VFSR) / VFSR \times 100\% (\%FSR)$$

$$EG = (V'FSR - VFSR) / VLSB \times 100\% (\%LSB)$$

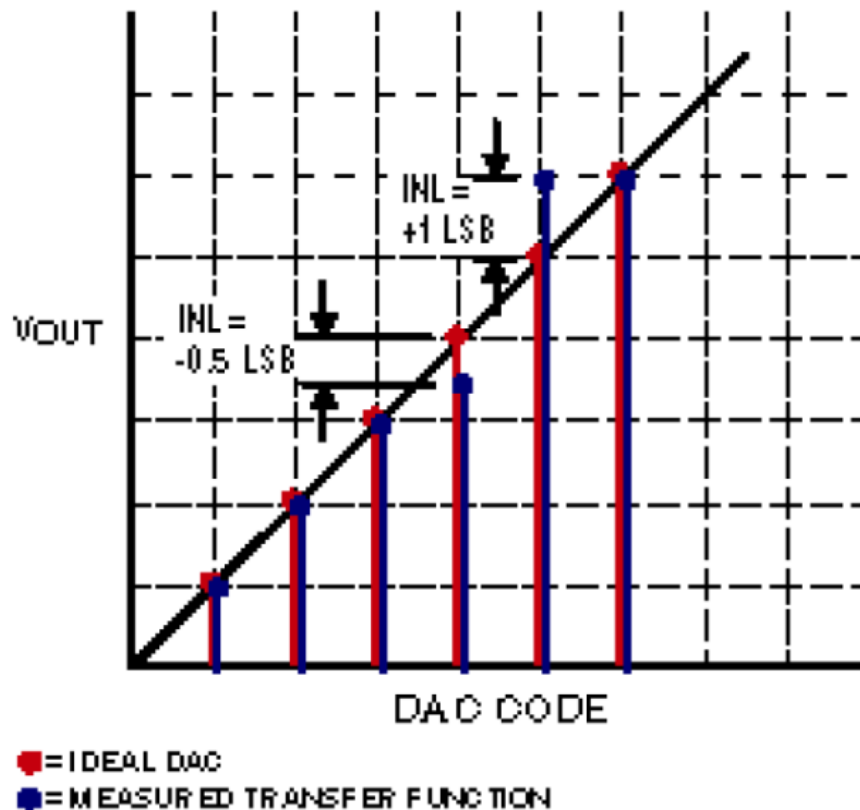
式中 **VFSR** 为理想输出满量程电压； **VLSB** 为单位码变化所对应的模拟量的变化。

- 误差参数：



差分非线性误差（**DNL, Differential Nonlinearity**）指在器件工作时，理想条件下两相邻输入数字量所对应的模拟量差值应相等，但实际上由于元器件参数不理想，使其相邻差值不同。差分非线性误差是一种测量小信号非线性误差的方法，为相邻两输入数码对应的模拟输出电压之差的实际值与理想 **1VLSB** 间的最大偏差

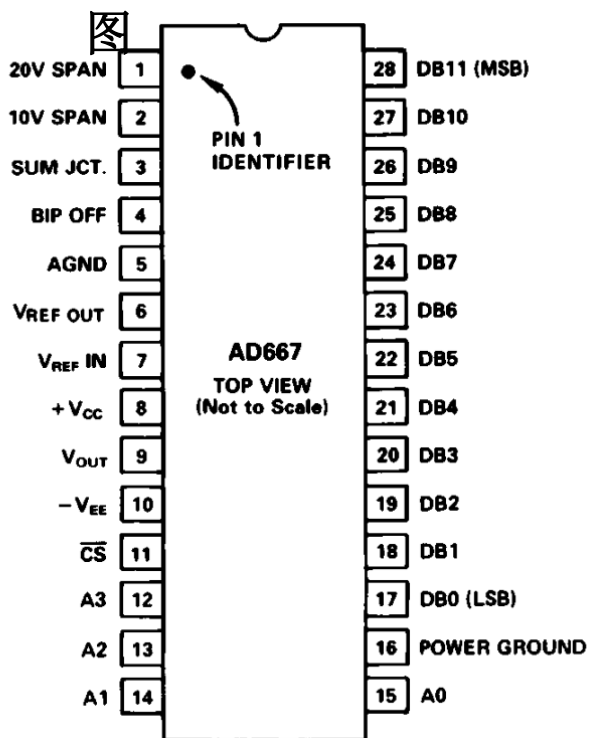
- 误差参数:



积分非线性误差（INL）与 DNL 对应，是实际输出的模拟电压与理想值的最大误差，是大信号输出非线性误差的测量方法，对任意给定输入和所有差分非线性，是一种累积误差。

• AD667的测试

AD667是一款12位D/A转换器，其管脚排列如



PIN1	20V SPAN	20V量程
PIN2	10V SPAN	20V量程
PIN3	SUM	求和端
PIN4	BIP OFF	补偿调整端
PIN5	AGND	模拟地
PIN6	REF OUT	内部基准输出
PIN7	REF IN	基准输入
PIN8	VCC	正电源
PIN9	VOUT	模拟输出
PIN10	VEE	负电源
PIN11	CS	片选
PIN12-15	A3-A1	缓冲器控制端
PIN16	PGND	电源地
PIN17-28	DB	数字量输入

• AD667的测试

AD667的输出选择及时序图

Table I. Output Voltage Range Connections

Output Range	Digital Input Codes	Connect Pin 9 to	Connect Pin 1 to	Connect Pin 2 to	Connect Pin 4 to
$\pm 10\text{ V}$	Offset Binary	1	9	NC	6 (Through 50 Ω Fixed or 100 Ω Trim Resistor)
$\pm 5\text{ V}$	Offset Binary	1 and 2	2 and 9	1 and 9	6 (Through 50 Ω Fixed or 100 Ω Trim Resistor)
$\pm 2.5\text{ V}$	Offset Binary	2	3	9	6 (Through 50 Ω Fixed or 100 Ω Trim Resistor)
0 V to +10 V	Straight Binary	1 and 2	2 and 9	1 and 9	5 (or Optional Trim—See Figure 2)
0 V to +5 V	Straight Binary	2	3	9	5 (or Optional Trim—See Figure 2)

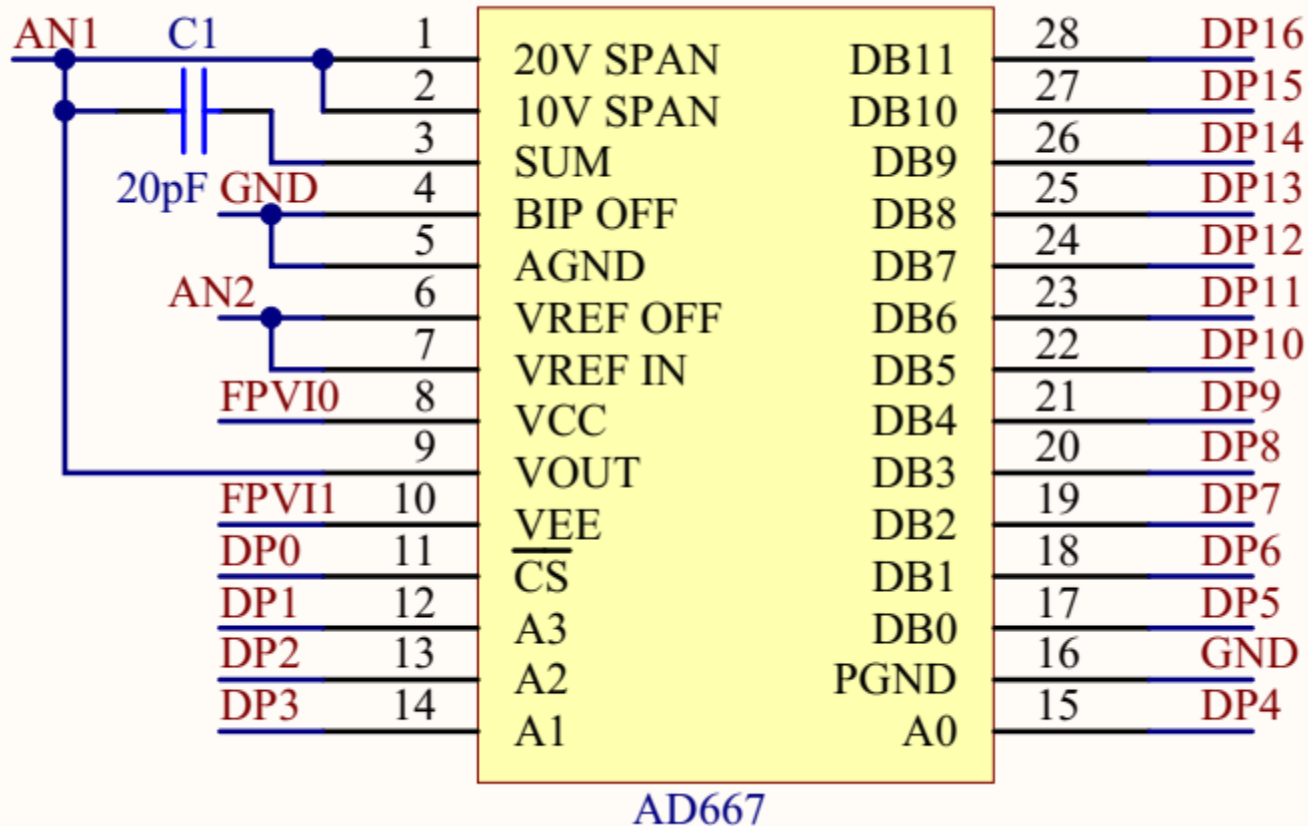
Table II. AD667 Truth Table

$\overline{\text{CS}}$	A3	A2	A1	A0	Operation
1	X	X	X	X	No Operation
X	1	1	1	1	No Operation
0	1	1	1	0	Enable 4 LSBs of First Rank
0	1	1	0	1	Enable 4 Middle Bits of First Rank
0	1	0	1	1	Enable 4 MSBs of First Rank
0	0	1	1	1	Loads Second Rank from First Rank
0	0	0	0	0	All Latches Transparent

“X” = Don’t Care.

• AD667的测试

AD667适配器接线图：



- **AD667的误差测试方法**

DA的误差参数测试跟**AD**的误差参数测试思路是相似的，对于**AD667**来说，位数是**12bit**，一共有**4096**个模拟输出电压值。我们需要的就是把所有的数字量对应的模拟输出电压值全部测回来，再进行公式计算。

DA的误差参数测试的测试向量较大。注意测试向量文件允许的深度只有**65536**。如果超过**65536**行，我们需要拆分为多个测试向量文件，并在**C**程序内分开调用。

0011		newset1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0012		newset1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0013		newset1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0014		newset1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0015		newset1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0016		newset1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0017		newset1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
0018		newset1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0019		newset1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
0020		newset1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
0021		newset1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0
0022		newset1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0023		newset1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0
0024		newset1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
0025		newset1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0

DA的向量文件相对较长，如AD667的全码输出至少需要4096行，所以在编写向量文件时，可以将所有的数据脚，编成一个数据组，利用向量编辑器自带的矢量表编辑计数器来实现自动编写，大大节省了人力。

- AD667的关键代码:

```
for(int i=0;i<4096;i++)  
{  
    double tmp;  
    tmp=10.0*i/4096.;  
    awg0.SetOutput(tmp);  
    vec.Run(i+10,i+10);  
    while(vec.Running());  
    delay_ms(1);  
    qvm0.MeasureLADC(10,10,QVM_LADC_10V,QVM_LADC_10KHz,MEAS_NORMAL);  
    val1[0]=qvm0.GetMeasResult(0,AVERAGE_RESULT);  
    output_v[i]=(float)(tmp+val1[0]/100);  
}
```

• AD667的关键代码:

```
VFSR=output_v[4095]-output_v[0];    //计算满量程电压
LSB=VFSR/4095;                        //计算LSB
EZ=output_v[0]/LSB;                   //计算EZ
GE=(10-VFSR)/VFSR; ;                  //计算EG
float DNL_DATA[4096];
float Max_DNL=0;
for(i=1;i<4096;i++)                    //计算DNL
{
    DNL_DATA[i]=output_v[i]-output_v[i-1]-LSB;
    if (fabs(DNL_DATA[i])>fabs(Max_DNL))
        {Max_DNL=DNL_DATA[i];}
}
Max_DNL/=-LSB;
float INL_DATA[4096];
float Max_INL=0;
for(i=1;i<4096;i++)                    //计算INL
{
    INL_DATA[i]= output_v[i]-output_v[0]-i*LSB; }
    if (fabs(INL_DATA[i])>fabs(Max_INL))
        {Max_INL=INL_DATA[i];}
}
Max_INL/=-LSB;
```

- 电流型D/A的测试:

电流输出型DA的测试跟电压输出型一致，只需要把输出的电流值经过放大器变为电压值就可以，具体方法在前面有介绍。注意的是：

1. 电阻 须选择精密电阻 必须选择精密电阻
2. 选择合适的阻值，使的输出的电压值必须小于±10V， 且最大幅值要接近±10V。

• AD7547的测试:

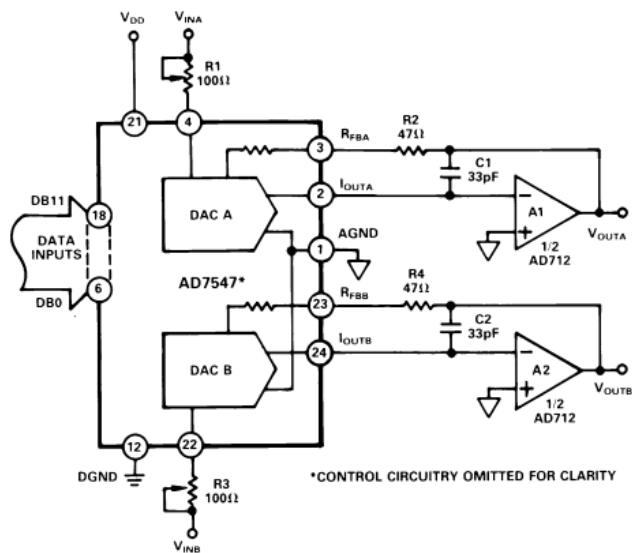


Figure 4. AD7547 Unipolar Binary Operation

Binary Number in DAC Register		Analog Output, V_{OUTA} or V_{OUTB}
MSB	LSB	
1111	1111 1111	$-V_{IN} \left(\frac{4095}{4096} \right)$
1000	0000 0000	$-V_{IN} \left(\frac{2048}{4096} \right) = -1/2 V_{IN}$
0000	0000 0001	$-V_{IN} \left(\frac{1}{4096} \right)$
0000	0000 0000	0V

Table II. Unipolar Binary Code Table for Circuit of Figure 4