

RX23E-A 产品群

有效使用 AFE 和 DSAD

概述

本文档描述了在 RX23E-A 上执行各种用于传感器处理的方法，以及外围电路和电路板设计的要点，以发挥 RX23E-A 的模拟性能。

目标器件

RX23E-A

目录

1. 模拟前端(AFE)和 24 位 Δ - Σ A/D 转换器(DSAD).....	2
1.1 Δ - Σ A/D 转换器.....	2
1.2 使用通道功能对多个信号进行采样.....	3
1.3 去除商用电源频率噪声.....	5
1.3.1 在高于必要带宽的情况下.....	5
1.3.2 在必要带宽内或更低的情况下.....	5
1.4 斩波.....	8
1.5 温度传感器.....	10
1.6 断路检测.....	12
2. 硬件设计注意事项.....	13
2.1 输入至 DSAD · AFE.....	13
2.1.1 输入电压范围.....	13
2.1.2 输入方法的选择.....	14
2.2 抗混叠滤波器.....	16
2.3 电路设计和电路板布局.....	17
2.3.1 信号布线.....	17
2.3.2 电源噪声.....	18
2.3.3 元器件布局.....	20
修订历史.....	21

1. 模拟前端(AFE)和 24 位 Δ - Σ A/D 转换器(DSAD)

1.1 Δ - Σ A/D 转换器

RX23E-A 包含两个 24 位 Δ - Σ A/D 转换器 (DSAD) 单元。

Δ - Σ A/D 转换器的原理图如图 1.1 所示：

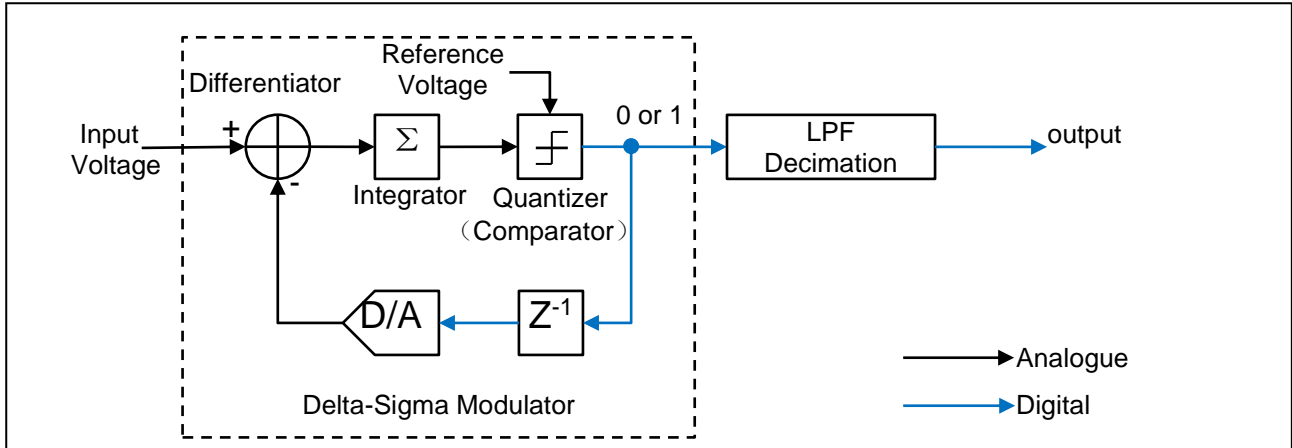


图 1.1 Δ - Σ A/D 转换器的原理图

Δ - Σ A/D 转换器由 Δ - Σ 调制器和低通滤波器/抽取处理组成。

Δ - Σ 调制器将输入电压与前次 A/D 转换值对应电压的差值的累加结果与基准电压进行比较，输出 0 或者 1 这 2 个值。

Δ - Σ 调制器的采样率对于输入信号的频率分量通常足够高，因此可以简化或省略为抗混叠滤波器。另外，对于 DC 电压输入，它不是固定值，表现为在 0 和 1 之间摆动。这是对于输入信号的频率足够高的频率分量。

Δ - Σ 调制器的 A/D 转换结果在低通滤波后按过采样率进行抽取。由于 Δ - Σ 调制器具有噪声整形特性，将量化噪声转移到高频范围。此外，通过低通滤波器抑制被吸引到高频范围的噪声分量，可以在抽取期间提高分辨率。

安装在 RX23E-A 上的 DSAD 以 0.5MHz（正常模式）或 0.125MHz（低功耗模式）的频率将模拟信号转换为数字信号，并且通过采用 SINC4 滤波器作为低通滤波器，来消除和抽取高频分量，其频率特性由过采样率决定。另外，它能够通过对 SINC4 滤波器输出的 8、16、32、64 个样本进行平均来执行抽取。

1.2 使用通道功能对多个信号进行采样

RX23E-A 使用通道功能为多个信号中的每一个设置 A/D 转换条件，并能够逐一进行 A/D 转换。一个 DSAD 最多可以转换六个信号。

通道功能主要可以设置以下条件作为每个输入信号的 A/D 转换条件。

- 输入引脚
- A/D 转换的参考电压
- PGA 的增益设置
- SINC4 的过采样率 (OSR)
- 偏移/增益的校正因数
- 均值功能

本节描述通过使用 DSAD0 按通道 0、1、2 的顺序为每个通道分配三个输入信号，在每种条件下每次将一个样本的信号从模拟转换为数字的示例。框图如图 1-2 所示。

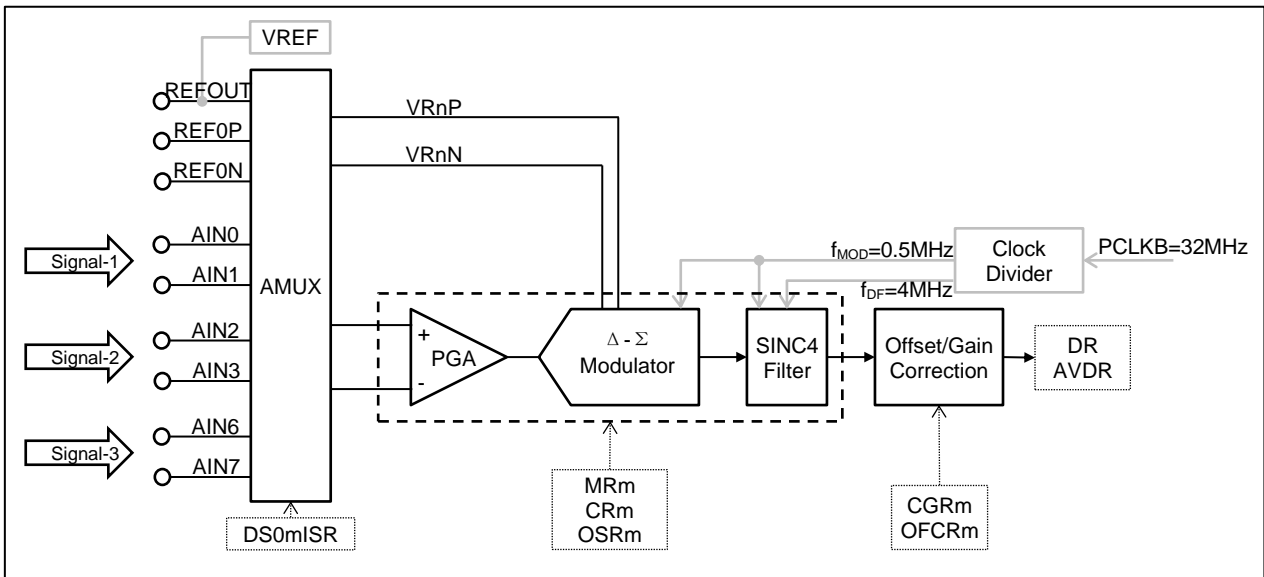


图 1-2 DSAD0 框图

A/D 转换的顺序如图 1-3 所示，每个输入信号的 DSAD0 转换条件如表 1-1 所示。

第一次测量的通道切换和模拟块稳定的时间在图 1-3 中描述为“a”，在第二次及以后的测量中描述为“a'”。

数字滤波器处理所需的时间被描述为 4T，包括数字滤波器稳定的时间，因为它是每个通道中的第一次转换。

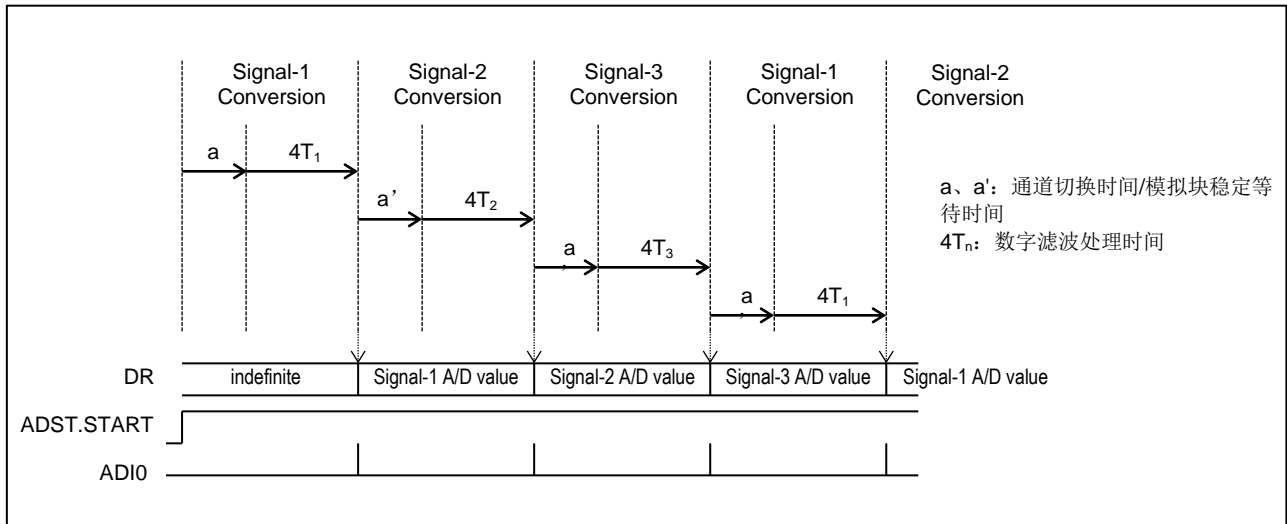


图 1-3 A/D 转换顺序

表 1-1 输入信号的 DSAD0 转换条件示例

正常模式: $f_{MOD} = 0.5\text{MHz}$

通道		0	1	2	备注
输入信号		1	2	3	
设置	输入引脚	+: AIN0 -: AIN1	+: AIN2 -: AIN3	+: AIN6 -: AIN7	
	PGA 增益	x32	x16	x128	
	参考电压	REF0P - REF0N	AVDD0 - AVSS0	REFOUT - AVSS0	
	OSR	50000	10000	8336	过采样率设置
A/D 转换时间		400.256[msec]	80.256[msec]	66.944[msec]	$a' + 4T_n$
		a 0.259[msec]			用于通道切换和稳定的时间
		a' 0.256[msec]			
		T_n 100.0[msec]	20.0[msec]	16.672[msec]	用于数字滤波器处理的时间 $T_n = \text{OSR}_n / f_{MOD}$
数据速率		1.8266[sps]			1 / 总的 A/D 转换时间= $1 / ((a'+4T_1) + (a'+4T_2) + (a'+4T_3))$

1.3 去除商用电源频率噪声

去除 50Hz/60Hz 商用电源频率噪声的实例根据噪声频率和必要带宽之间的关系分别进行说明。

1.3.1 在高于必要带宽的情况下

如果噪声频率高于必要带宽，则用低通滤波器将其去除。

利用 SINC4 滤波器的特性设置过采样率。设置该过采样率，使 SINC4 的陷波达到要去除的频率。详见《RX23E-A 组用户手册硬件版本（R01UH0801）》。

与巴特沃斯低通滤波器相比，逆切比雪夫低通滤波器可以缩小过渡带宽。此外，将衰减带纹波设置在 50Hz/60Hz，可以提高去除商用频率噪声的效果。

1.3.2 在必要带宽内或更低的情况下

如果商用电源频率噪声在必要带宽内或低于必要带宽，可以使用软件通过数字滤波器处理去除。噪声在必要带宽内的情况下，带阻滤波器有效，而噪声在必要带宽之外的情况下，带通滤波器有效。DSP 库可用于滤波操作。本节介绍带阻滤波器通过 IIR 双二阶滤波操作去除存在于 50Hz 至 60Hz 之间分量的示例。在此示例中，使用《RX DSP 库 V5.0（R01AN4359）》”的双二阶 IIR 来执行浮点运算。

图 1-4 中显示了用以下等式描述的 IIR 双二阶滤波器的传递函数。

$$H(z) = \prod_{m=1}^M \frac{b_0^m + b_1^m * Z^{-1} + b_2^m * Z^{-2}}{1 + a_1^m * Z^{-1} + a_2^m * Z^{-2}}$$

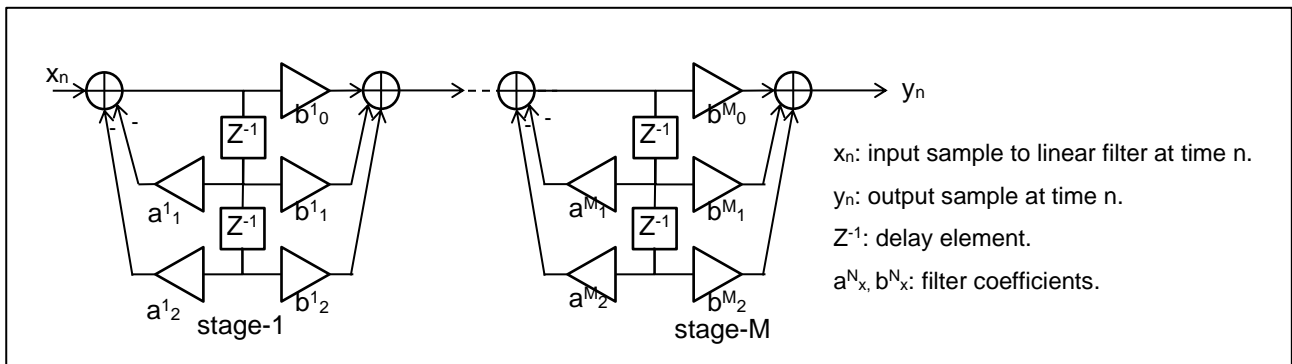


图 1-4 IIR 双二阶滤波器（两级级联）

由于商用电源的频率误差为±0.3Hz，在以下滤波器条件下设计的带阻滤波器的系数如表 1-2 所示。频率特性如图 1-5 所示。IIR 双二阶滤波器级数可实现 10 级。

采样频率:	1000[Hz]: 由 DSAD 的输出数据速率决定
较低的通带边缘频率:	42[Hz]
较低的阻带边缘频率:	48[Hz]
较高的阻带边缘频率:	62[Hz]
较高的通带边缘频率:	70[Hz]
通带衰减:	0.5[dB]
阻带衰减:	48[dB]

表 1-2 带阻滤波器系数

频带	系数	值	频带	系数	值
1	b0	-1.759983012	6	b0	-1.728177942
	b1	0.859464848		b1	0.878290095
	b2	0.859754821		b2	1.0
	a1	-1.620427124		a1	-1.884754914
	a2	0.859754821		a2	1.0
2	b0	-1.734986523	7	b0	-1.869770676
	b1	0.849769017		b1	0.943952367
	b2	1.0		b2	0.927312973
	a1	-1.884754914		a1	-1.747757683
	a2	1.0		a2	0.927312973
3	b0	-1.79337287	8	b0	-1.751726936
	b1	0.880852509		b1	0.917752298
	b2	0.871253429		b2	1.0
	a1	-1.642099183		a1	-1.884754914
	a2	0.871253429		a2	1.0
4	b0	-1.723132138	9	b0	-1.908330129
	b1	0.855408533		b1	0.980811771
	b2	1.0		b2	0.969800235
	a1	-1.884754914		a1	-1.827835759
	a2	1.0		a2	0.969800235
5	b0	-1.830921396	10	b0	-1.792740878
	b1	0.910019237		b1	0.97044565
	b2	0.893996162		b2	1.0
	a1	-1.684963659		a1	-1.884754914
	a2	0.893996162		a2	1.0

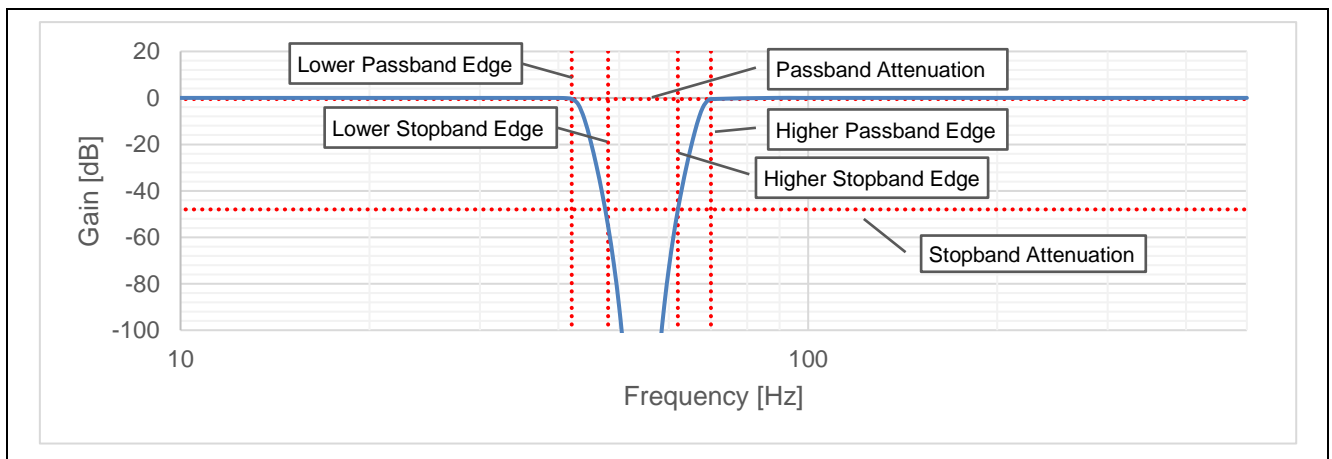


图 1-5 带阻滤波器频率特性

DSP 库的 IIR 双二阶执行周期如表 1-3 所示。此外，带阻滤波器计算的示例程序请见下一页。

表 1-3 执行周期数

ICLK=32MHz			
运算函数	执行周期数 (执行时间)	处理负载[%]	条件
R_DSP_IIRBiquad_f32f32	306cycle (9.5625 μ sec)	0.014	stages: 10, 1sample I/O

```

#include <r_dsp_filters.h>
#include <iodef.h>
#define NUM_COEFS_PER_BIQUAD (5)
#define NUM_DELAYS_PER_BIQUAD (2)
#define NUM_BIQUAD_STAGES (10)

/* Coefficients should be stored in reserved order as follows,
   b10, b11, b12, a11, a12, b20, b21, b22, a21, a22, b30, ... */
float myCoeffs[NUM_COEFS_PER_BIQUAD * NUM_BIQUAD_STAGES] = // b0,b1,b2,a1,a2
{-1.759983012, 0.859464848, 0.859754821, -1.620427124, 0.859754821, // stage-1
-1.734986523, 0.849769017, 1.0, -1.884754914, 1.0, // stage-2
-1.79337287, 0.880852509, 0.871253429, -1.642099183, 0.871253429, // stage-3
-1.723132138, 0.855408533, 1.0, -1.884754914, 1.0, // stage-4
-1.830921396, 0.910019237, 0.893996162, -1.684963659, 0.893996162, // stage-5
-1.728177942, 0.878290095, 1.0, -1.884754914, 1.0, // stage-6
-1.869770676, 0.943952367, 0.927312973, -1.747757683, 0.927312973, // stage-7
-1.751726936, 0.917752298, 1.0, -1.884754914, 1.0, // stage-8
-1.908330129, 0.980811771, 0.969800235, -1.827835759, 0.969800235, // stage-9
-1.792740878, 0.97044565, 1.0, -1.884754914, 1.0 // stage-10
};

float myDLine[NUM_DELAYS_PER_BIQUAD * NUM_BIQUAD_STAGES];

r_dsp_iirbiquad_t myHandle = { // instantiate a handle for IIR Biquad filter api
    NUM_BIQUAD_STAGES, // stages
    (void*)myCoeffs, // Pointer to coefficients array
    (void*)myDLine, // Pointer to delay line
    1.0, // scaling factor
    0, // qint, not referred on floating points api
    0, // option, not referred on floating point api
    R_DSP_BIQUAD_FORM_II // Biquad form
};

float InData, OutData;
vector_t myInput = {1, (void*)&InData}; // input vector
vector_t myOutput = {1, (void*)&OutData}; // output vector
r_dsp_status_t myRetVal = R_DSP_STATUS_OK; // place to store return status

/* Initialize the coefficients and internal state */
myRetVal = R_DSP_IIRBiquad_Init_f32f32(&myHandle);

while(1)
{
    while (1U == IR(DSAD0, ADI0)) // waiting for A/D conversion to end
    {
        /*----- Get DSAD0 data -----*/
        IR(DSAD0, ADI0) = 0; // Clear IR flag
        InData = (float)((int32_t)((DSAD0.DR.LONG) << 8) >> 8); // sign extension

        /*----- DSP Library function call -----*/
        myRetVal = R_DSP_IIRBiquad_f32f32(&myHandle, &myInput, &myOutput);

        /*--- filtered output is stored in OutData as float type ---*/
    }
}

```

1.4 斩波

A/D 转换器的偏移电压可以通过 DSAD 的偏移校正功能进行校正。然而，它不适合校正由温度漂移引起的偏移变化，温度漂移是由于环境温度随时间变化而引起的变化。

如果 A/D 输入采用差分输入，则可以通过交替输入正接和反接将信号由模拟转换为数字的结果进行平均，则可以交替设置正常连接输入和反向连接输入，并将模拟信号转换为数字信号后，对转换结果进行平均来补偿温度漂移引起的偏移。图 1-6 显示了去除偏移漂移的图像。

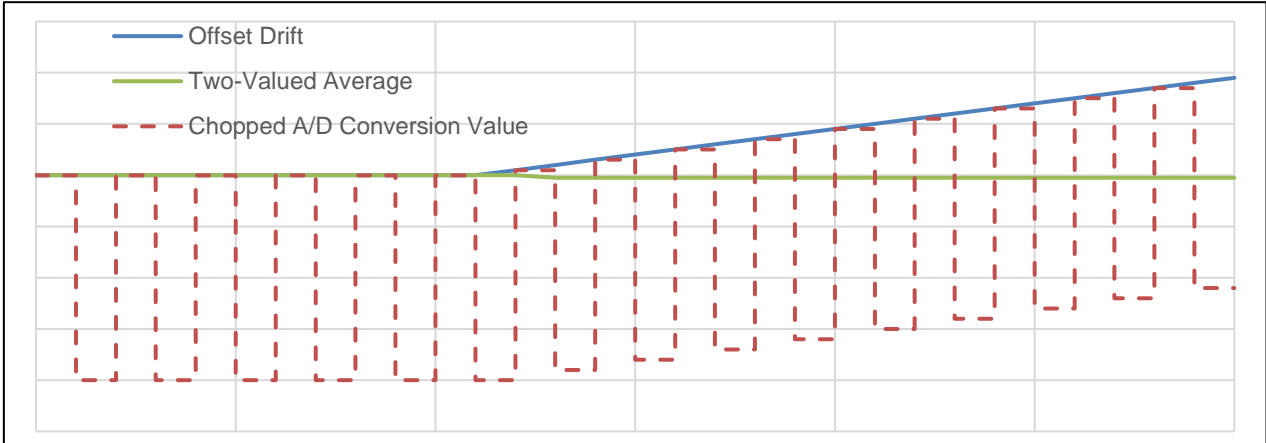


图 1-6 通过斩波去除偏移漂移

例如，使用通道功能进行采样，通道 0 正常连接输入、通道 1 反向连接，然后继续交替进行一次采样。框图如图 1-7 所示，A/D 转换条件如表 1-4 所示。

除了输入引脚的设置外，通道 0 和 1 的设置相同。通道 0 为正常连接，通道 1 为反向连接。

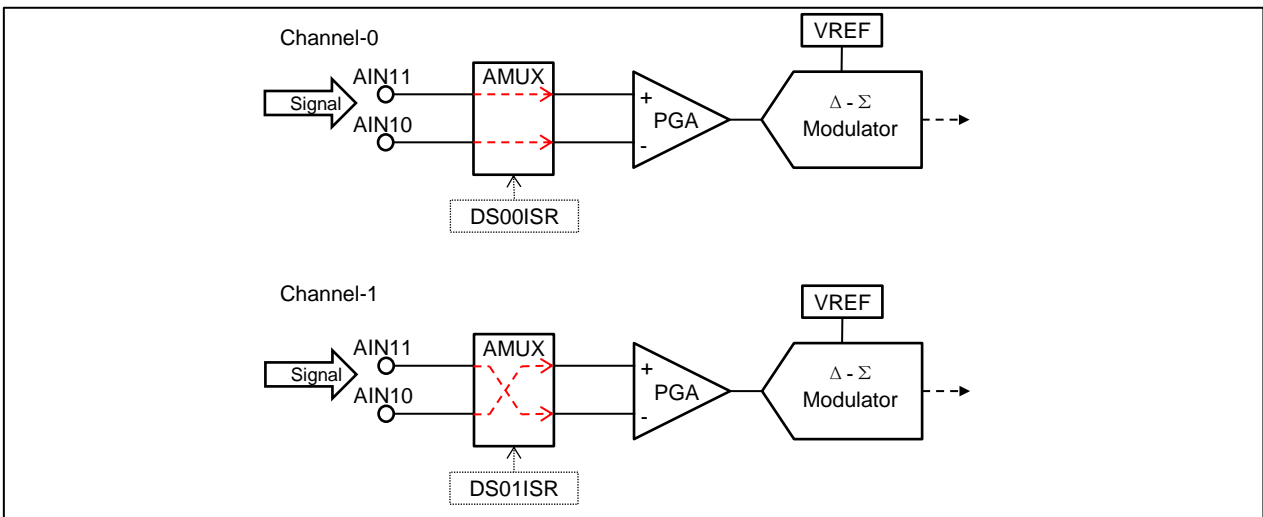


图 1-7 DSAD0 斩波框图

表 1-4 用于斩波的 DSAD0 转换条件示例

正常模式: $f_{MOD} = 0.5\text{MHz}$

项目	设置		备注
通道	0	1	
输入引脚	+ : AIN10 - : AIN11	+ : AIN11 - : AIN10	
PGA 增益	x128		
参考电压	REFOUT - AVSS0		
OSR	50000		设置过采样率
A/D 转换时间	400.256[msec]	400.256[msec]	$a' + 4T_n$
	a'	0.256[msec]	通道切换时间 · 模拟块稳定等待时间
	T _n	100.0[msec]	数字滤波器处理时间 $T_n = \text{OSR}_n / f_{MOD}$
数据速率	1.2492[sps]		$1 / \text{总的 A/D 转换时间} = 1 / ((a'+4T_1) + (a'+4T_2))$

下面是对正/反连接获取的 A/D 转换值进行平均处理的代码示例。

```
#include <iodef.h>
#include <stdint.h>

uint32_t ad_data;
int32_t ad_value = 0;

while(1)
{
    while (1U == IR(DSAD0, ADI0))    // waiting for A/D conversion to end
    {
        IR(DSAD0, ADI0) = 0;          // Clear IR flag
        ad_data = DSAD0.DR.LONG;      // Get DSAD0 data
        if ( 1 == (ad_data >> 29) )
        { // channel-0: positive connection data
            ad_value = ((int32_t)(ad_data << 8)) >> 8; // sign extension
        }
        else
        { // channel-1: negative connection data
            ad_value -= ((int32_t)(ad_data << 8)) >> 8; // sign extension and substitute
            ad_value >>= 1; // averaging
        }
    }
}
```

1.5 温度传感器

RX23E-A 具有内置温度传感器，用于测量器件内部的温度。温度传感器的输出通过 DSAD0 从模拟转换为数字，然后使用器件中集成的浮点格式系数转换为温度。

温度传感器输出可以与其他外部传感器信号按顺序转换，使用“1.2 使用通道功能对多个信号进行采样”中所述的通道功能。

此示例说明如何对温度传感器的输出进行采样，并将其转换为温度。表 1-5 显示了 DSAD0 转换条件。下一页显示了一个根据所获取的 A/D 转换值来计算温度的代码示例。

数字滤波器的增益随着过采样率(OSR)的设置而变化，因此，A/D 转换值需要进行校正。使用下面的等式求校正系数 $DFGainCoef$ ，将所获得的系数乘以 A/D 转换值进行校正。

$$DFGainCoef = 2^{ceil(4 \log_2 OSR) - 4 \log_2 OSR}$$

通过以下等式，使用单精度浮点计算，用 $ADValue$ 来计算温度， $ADValue$ 是通过数字滤波器对 A/D 转换值进行增益校正后确定的值。

$$Temperature = (AFE.TC2R) \cdot ADValue^2 + (AFE.TC1R) \cdot ADValue + (AFE.TC0R) [^{\circ}C]$$

表 1-5 用于温度传感器的 DSAD0 转换条件的示例

正常模式: $f_{MOD} = 0.5MHz$

项目	设置	备注
通道	0	
输入引脚	温度传感器	DS00ISR.TSEN = 1
PGA 增益	旁路	DSAD0.CR0.GAIN 设置被禁用
参考电压	温度传感器专用	RSEL 设置被禁用
OSR	2048	过采样率设置
数据速率	10.0[sps]	

```
#include <iodef.h>
#include <stdint.h>
#include <math.h>

#define LOG10_2 0.301029996 //log10f(2.0)

/* Temperature coefficients */
float tc0r = *(float*)&AFE.TC0R;
float tc1r = *(float*)&AFE.TC1R;
float tc2r = *(float*)&AFE.TC2R;

uint32_t osr = DSAD0.MR0.BIT.OSR; // OSR selection
float DFGainCoef; // Digital Filter Gain Correction Coefficient
float temp; // temperature

/*----- Calculate Digital Filter Gain Correction Coefficient -----*/
osr = (7 == osr) ? ((DSAD0.OSR0 + 1) << 4) : 64 << osr; // OSR0 should be over 0
DFGainCoef = (4.0/LOG10_2)*log10f((float)osr);
DFGainCoef = powf(2.0,ceilf(DFGainCoef) - DFGainCoef);

while(1)
{
    while (1U == IR(DSAD0, ADI0)) // waiting for A/D conversion to end
    {
        /*----- Get DSAD0 data -----*/
        IR(DSAD0, ADI0) = 0; // Clear IR flag
        temp = (float)(((int32_t)((DSAD0.DR.LONG) << 8)) >> 8); // sign extension
        temp *= DFGainCoef; // Digital Filter Gain correction

        /*----- Conversion to temperature -----*/
        temp = temp * temp * tc2r + temp * tc1r + tc0r;
    }
}
```

1.6 断路检测

RX23E-A 具备断路检测辅助功能，可检测测量电路的断线。断路检测辅助功能由恒定电流源和给测量电路提供电流的开关组成。

本节介绍检测热电偶测量电路断线的示例，如图 1-8 所示。图 1-9 显示了在表 1-6 所示的 DSAD0 转换条件下进行断路检测时 A/D 转换值的变化。

如图 1-9 所示，如果在发生断线后使能断路检测辅助电流，则外部电路的电容将被充电。结果，A/D 转换值将变为最大值。因此，如果通过使能断开检测辅助电流，使 A/D 转换值变为最大值，则判断为发生断路。A/D 转换值保持在最大值的时间取决于断路检测辅助电流外部电路和 PGA 增益的时间常数。

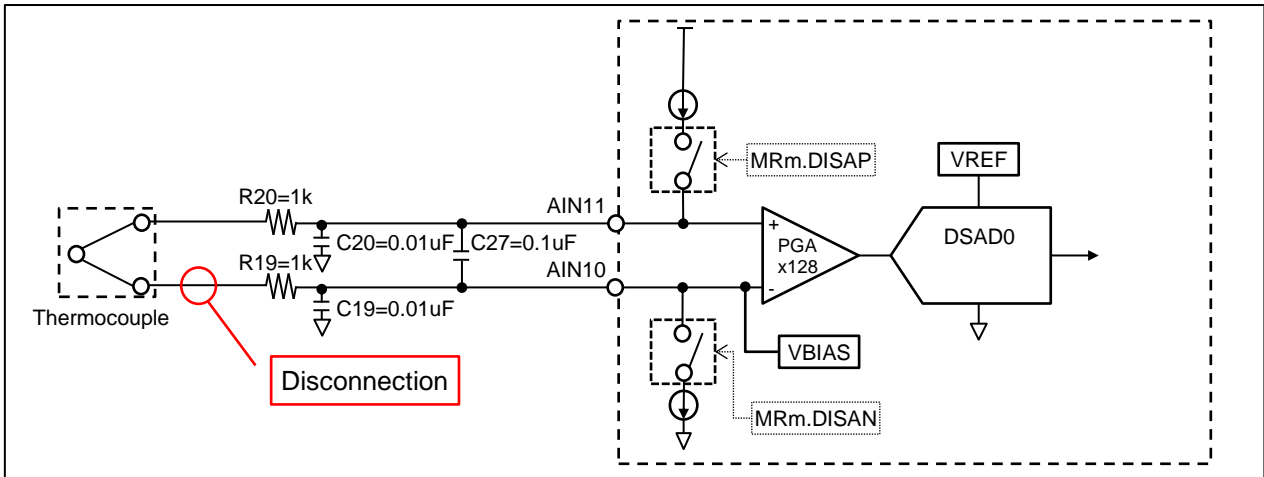


图 1-8 热电偶测量电路示例

表 1-6 DSAD0 转换条件示例

正常模式: $f_{MOD} = 0.5\text{MHz}$

项目	设置		备注
	传感器测量	断路检测	
输入引脚	+ : AIN10 - : AIN11		
PGA 增益	x128		
参考电压	REFOUT - AVSS0		
断路检测辅助电流	4 μ A		MR0.DISC = 2
正输入信号断路检测辅助	OFF(0)	ON(1)	MR0.DISAP
负输入信号断路检测辅助	OFF(0)	ON(1)	MR0.DISAN

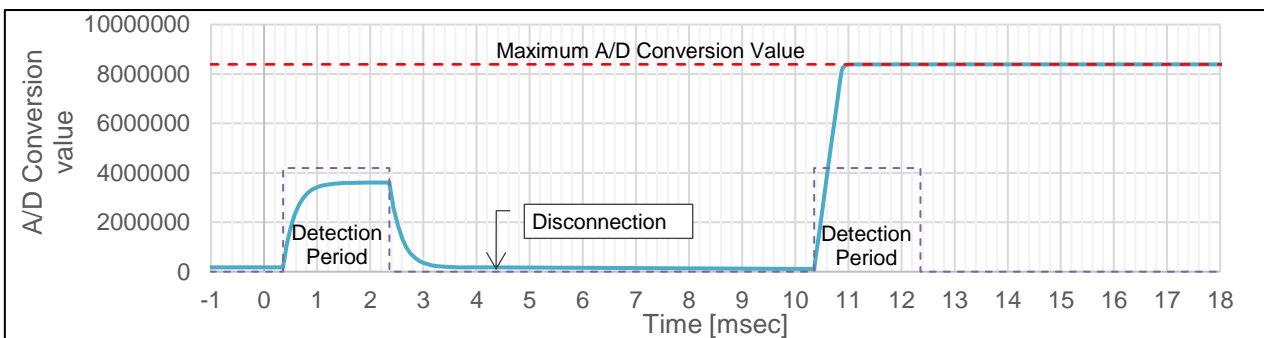


图 1-9 检测出断线时的 A/D 转换值变化示例

2. 硬件设计注意事项

2.1 输入至 DSAD · AFE

RX23E-A 上的 DSAD 配备 12 个输入引脚，其中 6ch 可用作差分输入/伪差分输入或 11ch 可用作单端输入。使用 DSAD 时，在 12 个输入引脚中选择 AIMP (+侧的输入选择引脚) 或 AINM (-侧的输入选择引脚)。

2.1.1 输入电压范围

输入到 DSAD 的输入电压必须在绝对输入电压范围和差分输入电压范围内。如果输入电压超出输入电压范围，则无法正确测量。

将 AINP 输入电压 V_{AINP} 和 AINM 输入电压 V_{AINM} 设置于绝对输入电压范围 $AVCC+V_{HRP}$ 以下、 $AVSS+V_{HRN}$ 以上。 V_{HRP} 和 V_{HRN} 指净空电压。当不使用模拟输入缓冲区 BUF 时，用以下等式描述： $V_{HRP} = +50mV$ ， $V_{HRN} = -50mV$ 。另一方面，当使用 BUF 时，则 $V_{HRP} = -100mV$ ， $V_{HRN} = +100mV$ 。请注意，使用 BUF 时绝对输入电压范围变窄。

将差分输入电压 $V_{IN} = V_{AINP} - V_{AINM}$ 设置在差分输入电压范围 $\pm VREF/GAIN$ 内。但是，当使用 PGA 且 GAIN 为 1x 时，差分输入电压范围将为 $\pm VREF$ 或 $\pm (AVCC0 - AVSS0 - 0.5V)$ 中的较窄者。如果输入了高于差分输入电压范围的电压，AD 值将饱和。

例如：当 $\pm VREF = \pm (AVCC0 - AVSS0) = \pm 5V$ 时，差分输入电压范围将为 $\pm 4.5V$ 。

根据以上描述，图 2.1 显示 $AVCC0=5V$ 、 $AVSS0=0V$ 、 $VREF=2.5V$ 、PGA 使能 (1、2、4、128x) 情况下的输入电压范围。在 AD 值不饱和的情况下可以测量的输入电压范围被限制在如图 2.1 所示的六边形内部。

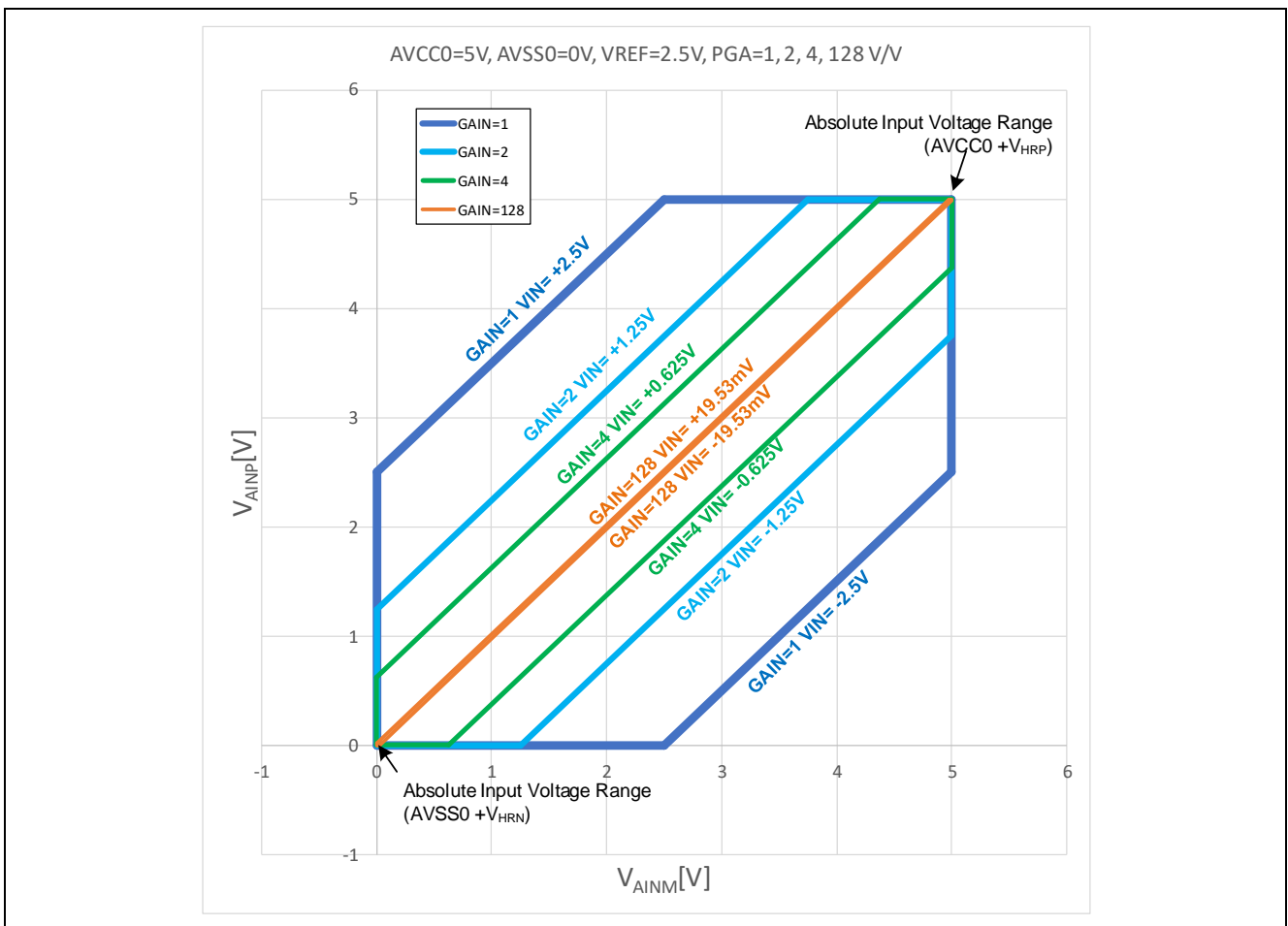


图 2.1 DSAD 的输入电压范围

2.1.2 输入方法的选择

(1) 单端模式

图 2.2 显示了单端模式下的电压测量示例。

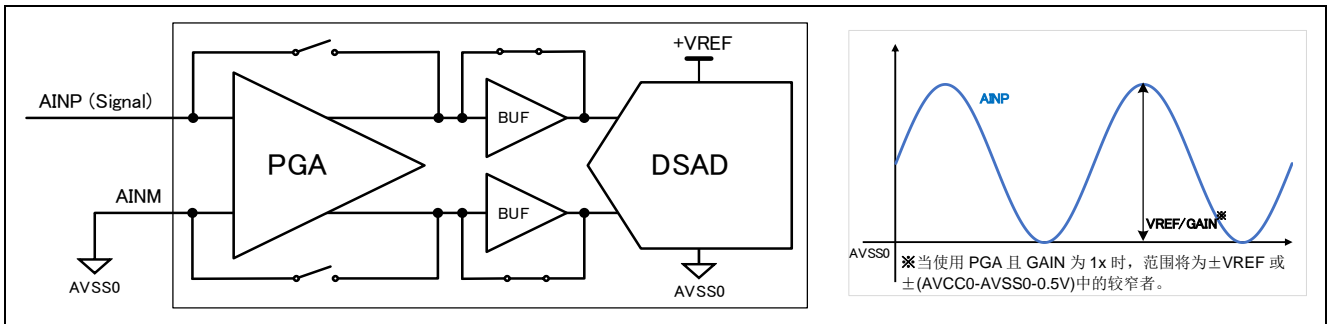


图 2.2 单端模式下的电压测量示例

在单端模式下使用 DSAD 时，从 12 个输入引脚中选择一个作为 AINM。如果所选引脚连接到 AVSS0，则可以通过从其余 11 个引脚中选择 AINP 来测量 AINP 和 AVSS0 之间的电压。虽然单端模式有一些影响，例如共模抑制比(CMRR)特性恶化，但它能够处理大多数输入信号。

(2) 伪差分输入模式

图 2.3 阐明了伪差分输入模式下的电压测量示例。

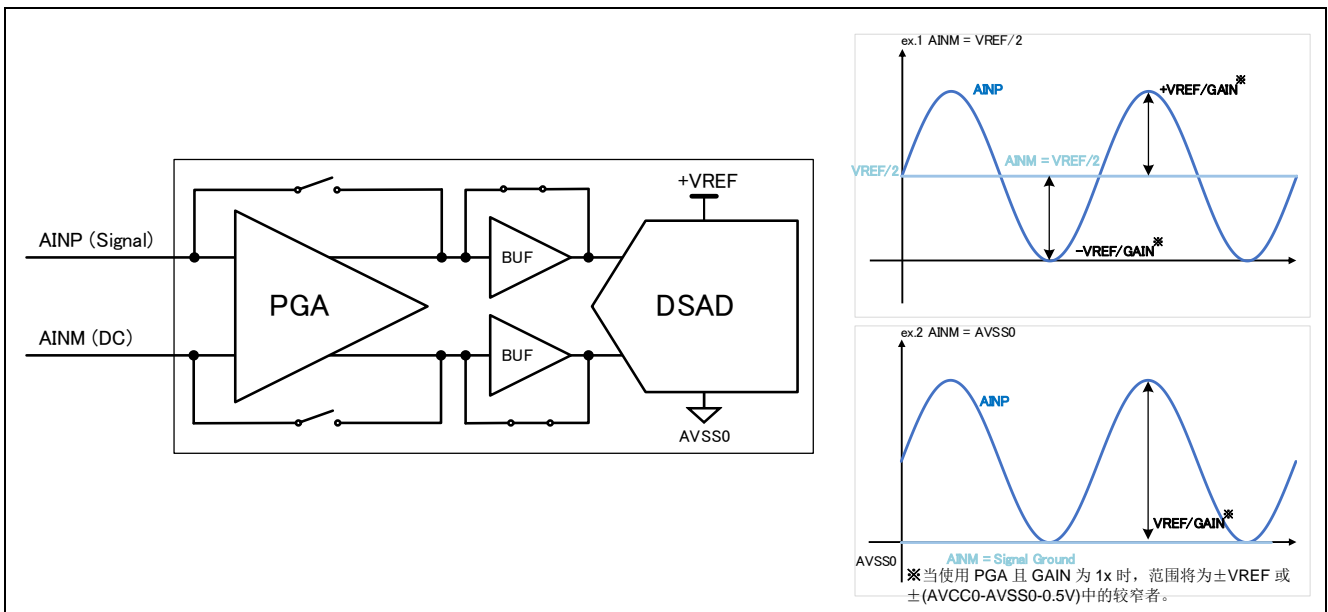


图 2.3 伪差分输入模式下的电压测量示例

在伪差分输入模式下，在一个引脚上保持 DC 电压的同时，向另一个引脚输入信号。在 DSAD 中，将信号与 DC 电压之差进行 A/D 转换，称为伪差分。如果 AINM 与 AVSS0 相连，则此模式的工作方式与单端模式相同。与单端模式的区别在于，AINM 与地回路分开，这可以去除共模噪声。然而，一般情况下，由于诸如信号源阻抗等各种因素的影响，在伪差分输入模式下，从 AINP 和 AINM 所看到的共模噪声源的阻抗是不同的，CMRR 特性不如在（真）差分输入模式下运行时好。

(3) 差分输入模式

图 2.4 阐明了（真）差分输入模式下的电压测量示例。

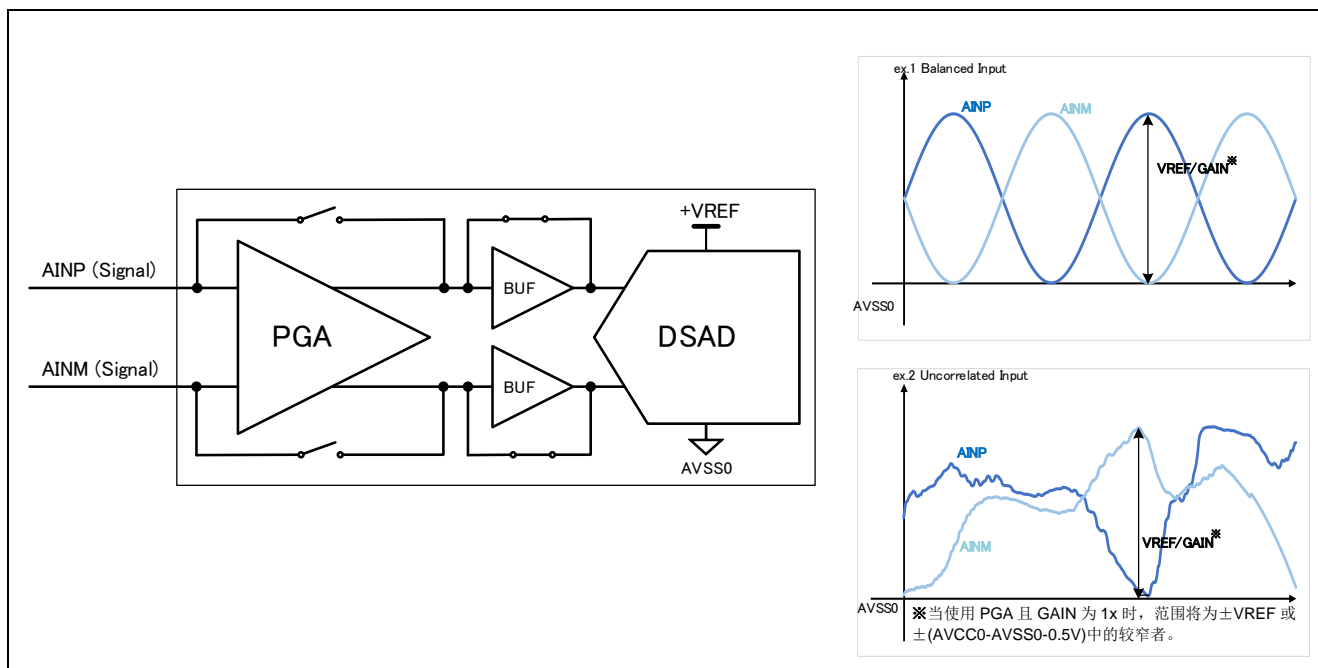


图 2.4 差分输入模式下的电压测量示例

在差分输入模式下，可以测量独立电压的平衡信号和差分电压。此模式展示了最高的 CMRR 特性。在差分输入和伪差分输入模式下，使用两个相邻的引脚。

在差分输入模式和伪差分输入模式下，建议成对使用 AINP 和 AINM 旁边的两个引脚。使用两个相邻的引脚可以加强两个引脚之间的连接。此外，通过匹配布线阻抗，可以减少 CMRR 特性的劣化。而且，在信号与其他信号之间敷设屏蔽线可以减少与其他信号的串扰。

2.2 抗混叠滤波器

图 2.5 说明为 DSAD 输入配置滤波器电路的示例。

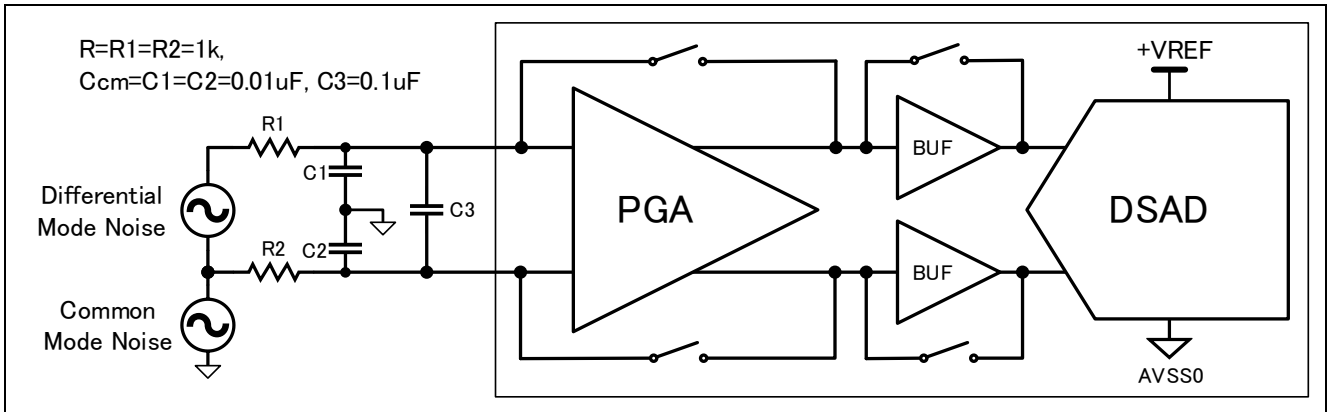


图 2.5 为 DSAD 输入配置滤波器电路示例

如果有可能进入带有通带附近频率分量的噪声，则需要添加抗混叠滤波器。如果过采样率很大，则输出数据速率和调制器（采样）频率之间会有很大的差异。因此，可以将抗混叠滤波器的截止频率设置得足够低，低于尼奎斯特频率 $f_{MOD} / 2$ 。这样，可以使用诸如 RC 滤波器的初级滤波器进行简单配置。

在图 2.5 中，用于共模噪声的低通滤波器的截止频率 f_{cm} 和用于差模噪声的低通滤波器的截止频率 f_{diff} 可以用下面的等式来描述。

$$f_{cm} = \frac{1}{2\pi RC_{cm}} = 15.9 \text{ [kHz]}$$

$$f_{diff} = \frac{1}{2\pi R(C_{cm} + 2C_3)} = 757.9 \text{ [Hz]}$$

图 2.6 说明了当 $f_{diff} = 757.9\text{Hz}$ 时，低通滤波器的增益频率特性，以及当输出数据速率为 956.5625SPS（过采样率 512）时，抽取滤波器的增益频率特性。从图中可以看出，即使差分低通滤波器的截止频率(757.9 Hz)配置在数据速率(956.5625 SPS)附近，也可以为 f_{MOD} 附近的通带确保足够的衰减量。

对于共模噪声的滤波器，要考虑由部件差异引起的 CMRR 劣化。当 CMRR 特性很重要时，请考虑使用公差小的部件。否则，由于电容的变化，高频带的 CMRR 特性会恶化。

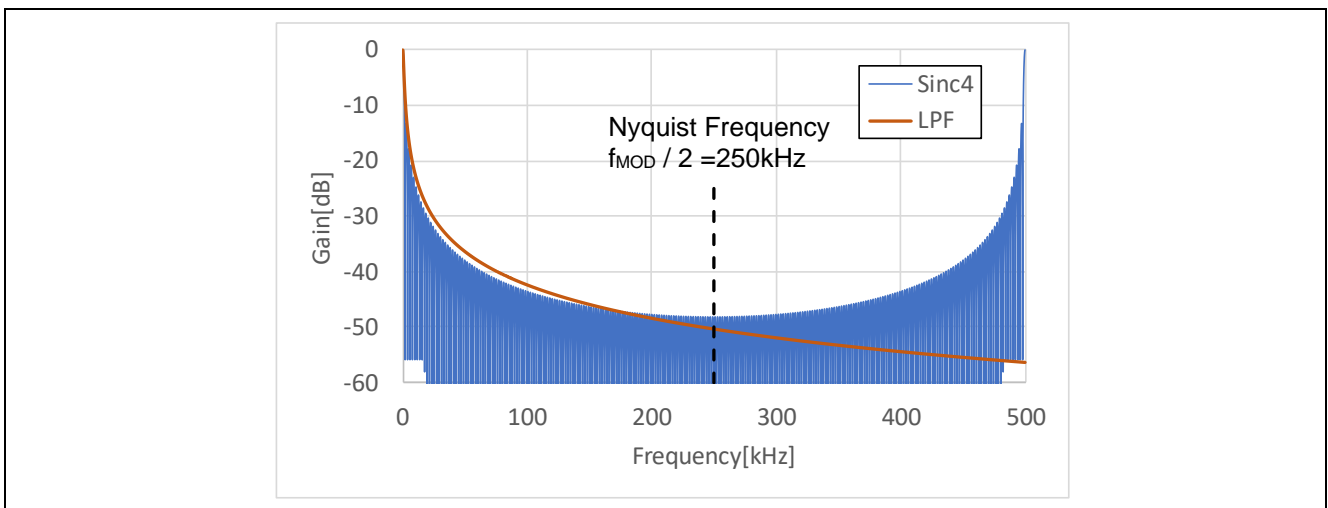


图 2.6 低通滤波器和抽取滤波器的增益频率特性

2.3 电路设计和电路板布局

为了充分发挥 DSAD 的性能，必须合理设置电源、接地布局和信号布线。本节描述了有关电路设计和电路板设计的注意事项。

2.3.1 信号布线

(1) 串扰

为防止串扰，不要在诸如数字信号、开关电源输出等变化较大并消耗电流较大的信号附近布置微小的模拟信号线。此外，确保微小信号的布线与成为噪声源的布线之间的空间。同时缩短电线彼此之间的接近的范围。布置地电位图案屏蔽层也是有效的。

需要保持屏蔽图案的端点闭合，并使用接地过孔以适当的间隔与接地平面连接。当端点保持开路状态时，不能起到屏蔽高频信号的作用。

当在差分输入模式或伪差分输入模式下使用 DSAD 时，保持成对并且靠近的方式进行布线。加强差分信号之间的耦合，可以提高对共模噪声的抗扰度。

图 2.7 显示屏蔽线和差分信号布线的示例。

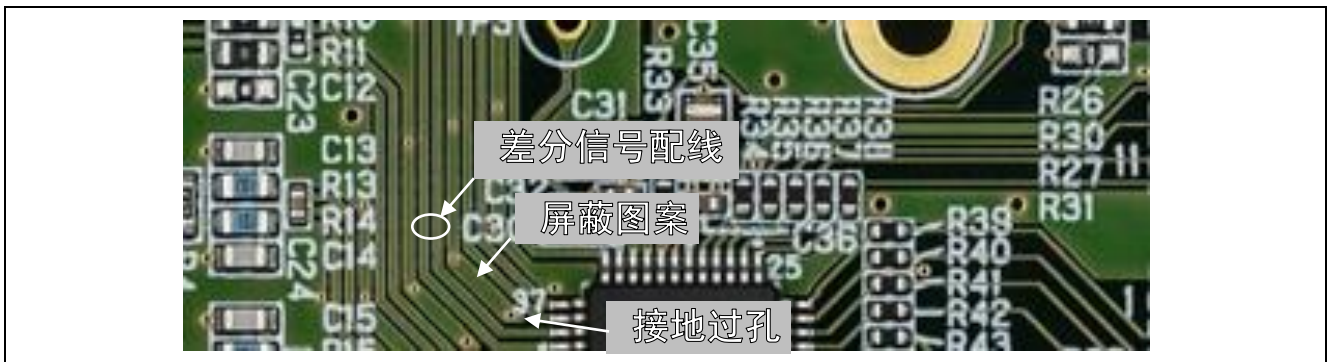


图 2.7 屏蔽线和差分信号布线的示例

(2) 温度梯度

当需要测量 μV 量级的微小信号时，例如高精度的热电偶测量，请注意电路板上的温度梯度。由于电路板式样和元件的焊接部分连接不同类型的金属，就像热电偶一样，它们会产生与温度差相对应的热电动势。因此，请确保作为高精度测量对象的信号线没有温度梯度。作为对策，通常有以下方法。

- 减少发热

若选择低能耗部件，则可以减少部件自身产生的热量。根据情况降低电源电压也很有效。

- 合理布置散热部件

可以通过在热源上安装诸如散热器等散热部件，将热量散发到周围的空气中。如此，热量被散发，并且可以避免局部温度梯度。电源 IC 通过在电路板平面上布置散热孔，使电路板平面具有散热器的功能。

- 隔热

将弱信号线与热源隔离也很有效。板上没有实心部分的区域具有很高的隔离效果。通过分割平面以创建没有实心部分的区域，可以减少暴露于弱信号线的热源产生的热量。除了这种隔离方法，将热源远离弱信号线的布置也是有效的。由于热源不仅存在于电路板上，还存在于电路板外部，因此使弱信号线远离这些热源很重要。

2.3.2 电源噪声

请考虑噪声造成的影响。虽然 RX23E-A 上的 DSAD 具有约 80dB 的电源抑制比(PSRR)特性，但在高精度应用中，需使电源变化最小。通过选择诸如合适的电源、合适的电源去耦和合适的电源布局等方法可以降低电源噪声。

(1) 选择电源

在选择 LDO、开关稳压器或其它 DC 电源时，请注意每个产品的性能特性，特别是输出电压纹波。请避免在高精度应用中使用开关稳压器。这样做的原因是尽管开关电源体积小、价格低廉且效率高，但它们会产生很大的噪声。但是，在使用开关电源时，请在使用前彻底评估噪声造成的影响。

使用电源设备时，请务必遵守各设备制造商的设计指南。使用开关电源作为电源时，请注意开关频率。避免使用开关频率接近 DSAD 的调制器时钟频率 f_{MOD} 的器件。

(2) 电源去耦

图 2.8 和图 2.9 分别展示了 LDO 和 RX23E-A 的电源去耦布局图。

在每个电源的输出附近放置一个具有足够容量的电容。合适的去耦电容有助于降低由电源输出的纹波噪声和负载波动的影响。将电容放置在 RX23E-A 上的 VCC、VSS、VCL、AVCC0 和 AVSS0 的输入引脚附近。使用 REFOUT 时，将电容放置在 REFOUT 引脚附近。

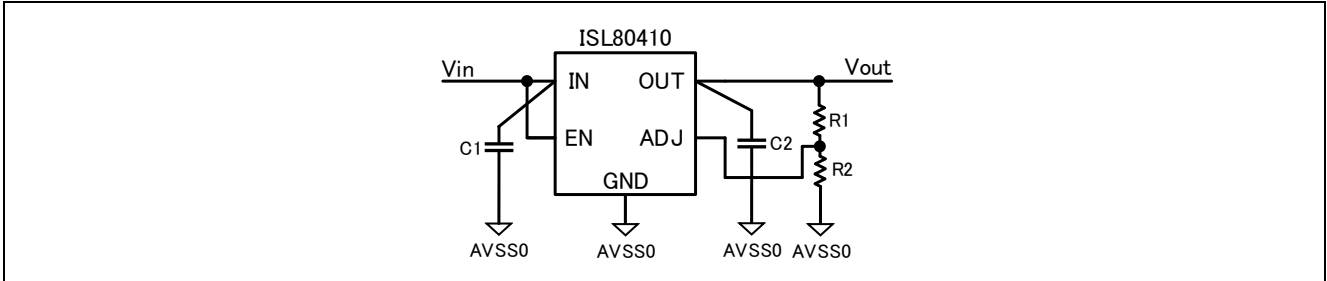


图 2.8 LDO 的电源去耦示例

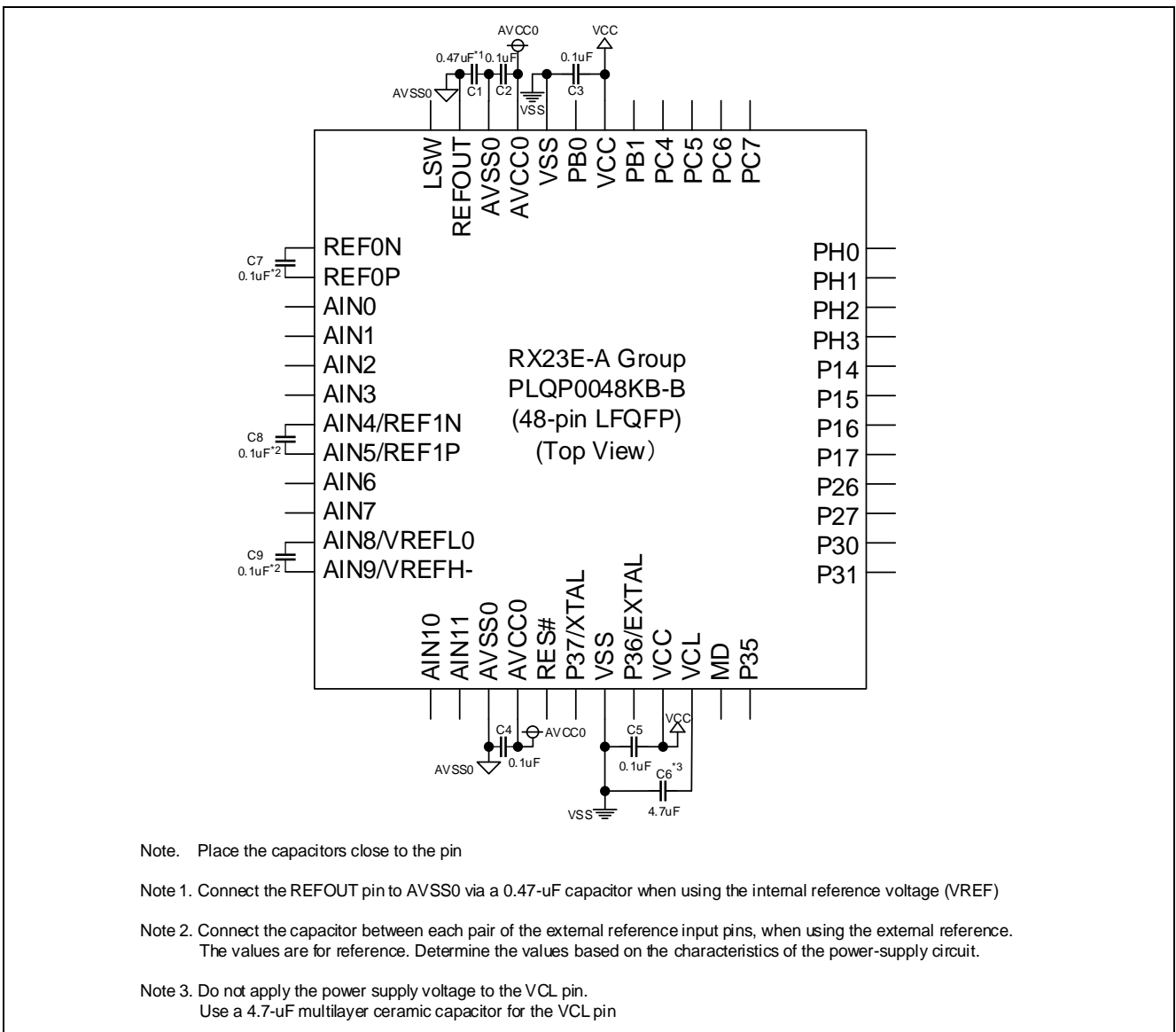


图 2.9 RX23E-A 的电源去耦

2.3.3 元器件布局

图 2.10 展示了使用 RX23E-A 的电路板布局示例。

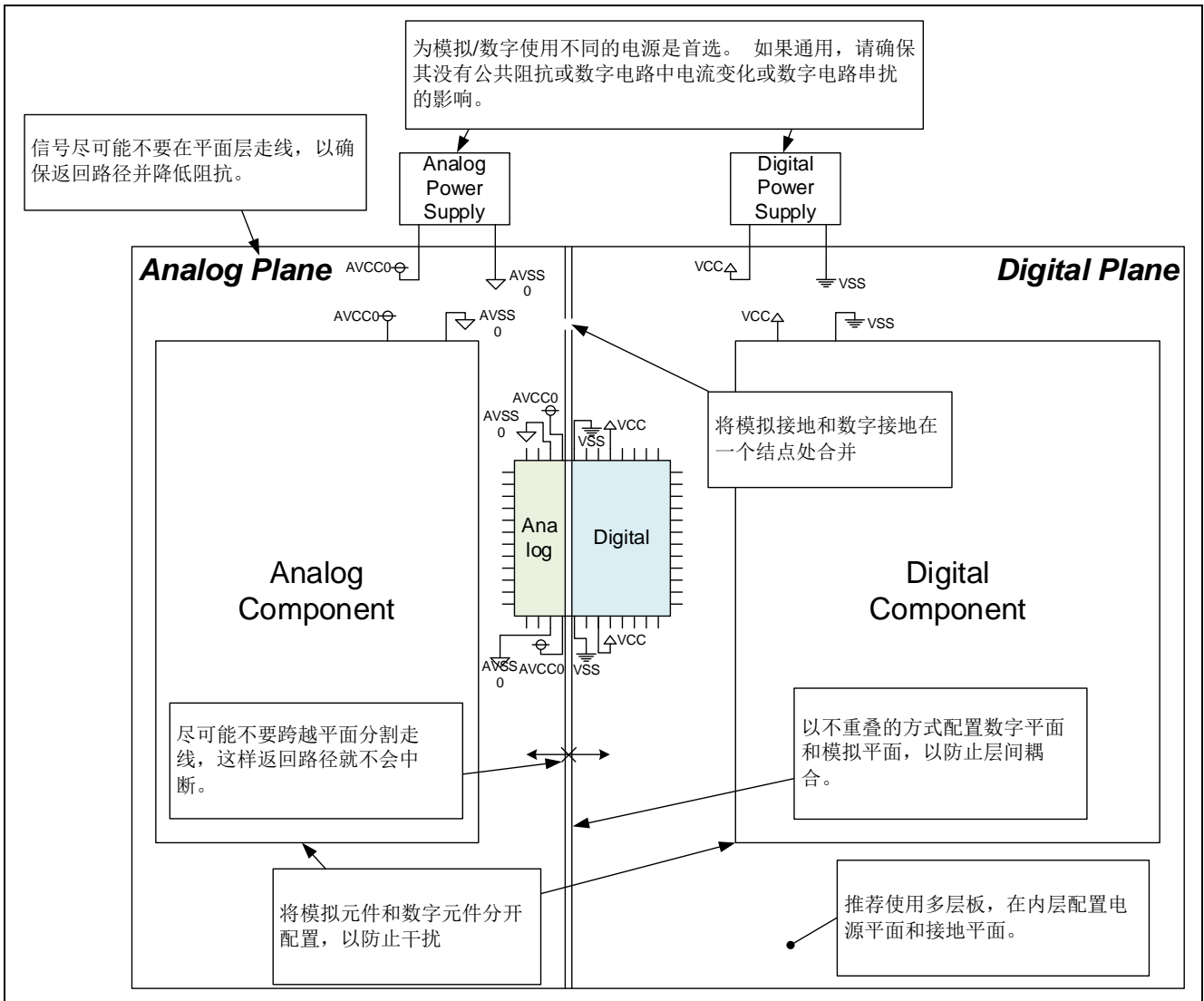


图 2.10 使用 RX23E-A 的电路板布局示例

请确保模拟电源不受数字电源和数字信号的影响。选择多层板并设置电源和接地平面可以保持电位稳定。以 4 层板为例，如果配置信号层、GND 层、电源层和信号层，从上往下看，层间电容起到电容器的作用，这样可以降低噪声。尽可能不要在平面层上连接其它走线。平面会被走线和过孔分开，然后增加的阻抗可以降低电压，如果返回路径不连续，噪声可能会以迂回的方式进入。配置数字信号和数字平面使其不会与模拟信号和模拟平面重叠。即使它们被配置在不同的层中，层与层之间也可能发生串扰。

在图 2.10 中，模拟和数字电源由电路板外部提供，模拟和数字接地在一个结点处合并。在这种情况下，该结点是参考电位，电源返回电流的模拟数字和直流分量之间的信号通过该结点。一般情况下，该结点靠近电源或 RX23E-A。如果它们在电源附近结合，模拟和数字之间的电源路由回路将变得最小。如果它们在 RX23E-A 附近结合，RX23E-A 上的 AVSS0 和 VSS 引脚之间的电位差可能最小，因此通过 RX23E-A 内的杂散电容传输到模拟块的数字噪声耦合可能最小。

修订历史

版本号	修订日期	描述	
		页码	概要
1.00	Jan. 16, 23	-	初版发行

有关微处理器和微控制器产品处理的一般预防措施

下方的使用说明适用于 **Renesas** 的所有微处理器和微控制器产品。有关本文档所述产品的详细使用说明，请参见本文档的相关章节以及针对相应产品发布的各项技术更新。

1. 防静电 (ESD) 措施

当暴露于 CMOS 器件时，强电场会导致栅极氧化物损坏，并最终使器件的工作性能下降。必须采取措施来尽可能地阻止静电的产生，并在发生静电时将其迅速消除。必须充分进行环境控制。如果环境干燥，应使用加湿器。建议避免使用容易产生静电的绝缘体。半导体器件必须在防静电容器、静电屏蔽袋或导电材料中存储和运输。所有测试和测量工具（包括工作台和地板）都必须接地。操作员还必须使用腕带接地。禁止徒手触摸半导体器件。对于装有半导体器件的印刷电路板，必须采取类似的预防措施。

2. 上电时的处理

产品在上电时仍处于未定义状态。在上电时，LSI 内部电路状态未定，寄存器设置和引脚的状态都未定义。对于复位信号已施加到外部复位引脚的成品，从上电开始到复位过程完成之前，引脚的状态并不能确定。同样地，对于以片上上电复位功能复位的产品，从上电开始到达到指定的复位电位之前，都无法确保引脚的状态。

3. 断电状态下的信号输入

器件断电时，请勿输入信号或 I/O 上拉电源。输入此类信号或 I/O 上拉电源导致的电流注入可能导致故障，此时流经器件的异常电流可能导致内部元件性能下降。请遵循产品文档中所述的电源关闭状态下输入信号的准则。

4. 未用引脚的处理

未使用的引脚应根据本手册“未用引脚的处理”部分给出的说明处理。CMOS 产品的输入引脚通常为高阻抗状态。在与开路状态下的未用引脚配合使用时，将在 LSI 周围产生额外的电磁噪声，内部将产生相关的直通电流，从而可能因为错误地将引脚状态识别为输入信号而产生故障。

5. 时钟信号

在应用复位后，只有在作业时时钟信号稳定后再释放复位线。当在程序执行过程中切换时钟信号时，等待目标时钟信号稳定。如在复位时外部谐振器（或外部振荡器）生成时钟信号，确保在时钟信号完全稳定后再释放复位线。此外，如在程序执行过程中切换到外部谐振器（或外部振荡器）生成的时钟信号，也需要等待目标时钟信号稳定。

6. 输入引脚上的电压施加波形

输入噪声或反射波引起的波形失真可能会导致故障。例如，如果由于噪声而使 CMOS 器件的输入处于 V_{IL} （最大值）和 V_{IH} （最小值）之间，则器件可能会发生故障。当输入电平固定时，以及在输入电平通过 V_{IL} （最大值）和 V_{IH} （最小值）之间时的过渡期间，请注意防止颤动噪声进入器件。

7. 禁止访问保留地址

禁止访问保留地址。保留地址用于未来可能的功能扩展。请勿访问此类地址，否则将无法保证 LSI 能正常运行。

8. 产品差异

在从一个产品切换到另一个前（例如，切换到具有不同零件号的产品），确认切换不会引起问题。对于同在一个系列但零件号不同的微处理器单元或单片机产品，其内部存储能力、布局图案和其他特性因素可能有所差异，进而可能会影响特性值、运行裕量、抗干扰性和噪声辐射量等电气特征的范围。在更换为部件编号不同的产品时，应针对给定产品执行系统评估测试。

注意

1. 本文件中电路、软件和其他相关信息的描述仅用于说明半导体产品的操作和应用示例。用户应对产品或系统设计中电路、软件和信息纳入或任何其他用途承担全部责任。对于您或第三方因使用这些电路、软件或信息而引起的任何损失和损害，Renesas Electronics 不承担任何责任。
2. Renesas Electronics 特此声明，对于因使用本文件中所述的 Renesas Electronics 产品或技术信息（包括但不限于产品数据、图纸、图表、程序、算法和应用示例）而引起的侵权或与第三方有关的专利、版权或其他知识产权的任何其他索赔，概不承担任何责任和赔偿。
3. 对 Renesas Electronics 或其他公司的任何专利、版权或其他知识产权均不授予任何明示、暗示或其他形式的许可。
4. 您应负责确定需要从任何第三方获得哪些许可，并在需要时为合法进口、出口、制造、销售、使用、分销或以其他方式处置包含 Renesas Electronics 产品的任何产品获得此类许可。
5. 不得对 Renesas Electronics 产品的全部或部分进行更改、修改、复制或逆向工程。对于因更改、修改、复制或逆向工程而导致您或第三方蒙受的任何损失或损害，Renesas Electronics 不承担任何责任。
6. Renesas Electronics 产品根据以下两个质量等级进行分类：“标准”和“优质”。Renesas Electronics 每种产品的预期应用取决于产品的质量等级，具体如下所示。
“标准”：计算机、办公设备、通信设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备、工业机器人等
“优质”：运输设备（汽车、火车、轮船等）；交通管制（交通信号灯）；大型通信设备；关键金融终端系统；安全控制设备等
除非在 Renesas Electronics 数据手册或 Renesas Electronics 其他文档中明确指定为高可靠性产品或用于恶劣环境的产品，否则 Renesas Electronics 产品不适合或不授权用于可能对人类生命构成直接威胁或造成人身伤害（人造生命支持设备或系统；手术植入物等），或者可能造成严重的财产损失（空间系统、海底中继器、核动力控制系统、飞机控制系统、关键设备系统、军事装备等）的产品或系统。对于因使用任何与 Renesas Electronics 数据手册、用户手册或其他 Renesas Electronics 文档不一致的 Renesas Electronics 产品而引起的您或任何第三方所造成的任何损坏或损失，Renesas Electronics 不承担任何责任。
7. 没有任何半导体产品是绝对安全的。尽管 Renesas Electronics 的硬件或软件产品中可能实施了任何安全措施或功能，Renesas Electronics 对因任何漏洞或入侵（包括但不限于以任何未经授权的方式访问或使用 Renesas Electronics 产品或使用 Renesas Electronics 产品的系统）而产生的任何后果概不负责。RENESAS ELECTRONICS 不担保或保证 RENESAS ELECTRONICS 产品或使用 RENESAS ELECTRONICS 产品创建的任何系统不会被破坏，或者可免于数据损坏、攻击、病毒、干扰、黑客攻击、数据丢失或失窃或其他安全入侵（“漏洞问题”）。RENESAS ELECTRONICS 不承担任何由任何漏洞问题引起的或与之相关的任何和所有责任或义务。此外，在适用法律允许的范围内，RENESAS ELECTRONICS 不对本文件和任何相关或附带的软件或硬件提供任何和所有明示或暗示的保证，包括但不限于对适销性或特定用途的适用性的暗示保证。
8. 使用 Renesas Electronics 产品时，请参见最新的产品信息（数据手册、用户手册、应用笔记、可靠性手册中的“处理和使用半导体器件的一般说明”等），并确保使用条件符合 Renesas Electronics 在最大额定值、工作电源电压范围、散热特性和安装等方面的规定。对于因在超出上述规定范围的情况下使用 Renesas Electronics 产品而引起的任何失常、故障或事故，Renesas Electronics 不承担任何责任。
9. 尽管 Renesas Electronics 致力于提高 Renesas Electronics 产品的质量和可靠性，但半导体产品具有特定的特性，例如在特定速率下发生故障以及在某些使用条件下出现故障。除非在 Renesas Electronics 数据手册或 Renesas Electronics 其他文档中指定为高可靠性产品或用于恶劣环境的产品，否则 Renesas Electronics 的产品将不受抗辐射设计的约束。用户应负责采取安全措施，以防止人身伤害、火灾造成的伤害，和/或因 Renesas Electronics 产品发生故障或失常而对公众造成的危险，例如硬件和设备的安全设计，包括但不限于冗余、火控和故障预防、针对老化退化的适当处理或任何其他适当的措施。由于对微型计算机软件进行评估非常困难且无实操性，因此用户有责任评估自己生产的最终产品或系统的安全性。
10. 请联系 Renesas Electronics 销售办事处，以获取有关环境事宜的详细信息，例如每个 Renesas Electronics 产品的环境相容性。用户有责任认真、充分地研究有关纳入或使用受控物质的适用法律和法规（包括但不限于欧盟 RoHS 指令），并按照所有适用法律和法规使用 Renesas Electronics 产品。对于因您未遵守适用的法律和法规而造成的损坏或损失，Renesas Electronics 不承担任何责任。
11. Renesas Electronics 产品和技术不得被用于或纳入为任何适用的本国或外国法律、法规所禁止制造、使用或销售的产品或系统范围内。用户应遵守由对当事方或交易拥有管辖权的任何国家/地区的政府颁布和管理的任何可适用的出口控制法律和法规。
12. 应由 Renesas Electronics 产品的购买方或分销商，或者对产品进行分发、处置或以其他方式出售或转让给第三方的任何其他当事方，负责将本文件中阐明的内容和条件提前通知前述第三方。
13. 未经 Renesas Electronics 事先书面同意，不得以任何形式全部或部分重印、再现或复制本文件。
14. 如果对本文件中包含的信息或 Renesas Electronics 产品有任何疑问，请联系 Renesas Electronics 销售办事处。

（注 1）本文件中的“Renesas Electronics”是指 Renesas Electronics Corporation，也包括其直接或间接控制的子公司。

（注 2）“Renesas Electronics 产品”是指 Renesas Electronics 开发或制造的任意产品。

（版本 5.0 - 2020 年 10 月 1 日）

公司总部

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

商标

Renesas 和 Renesas 徽标是 Renesas Electronics Corporation 的商标。所有商标和注册商标都是各自所有者的财产。

联系信息

有关产品、技术、文档最新版本或离您最近的销售办事处的更多信息，请访问：www.renesas.com/contact。