



南京凌鸥创芯电子有限公司

LKS32 MCU

运放差分 and 单端工作模式的区别

© 2020, 版权归凌鸥创芯所有
机密文件, 未经许可不得扩散





目录

1	概述.....	4
2	差分工作模式.....	6
3	单端工作模式.....	7
3.1	单端模式 V_{cm} 的校正.....	9
3.2	单端模式 V_{cm} 随温度的变化.....	11



1 概述

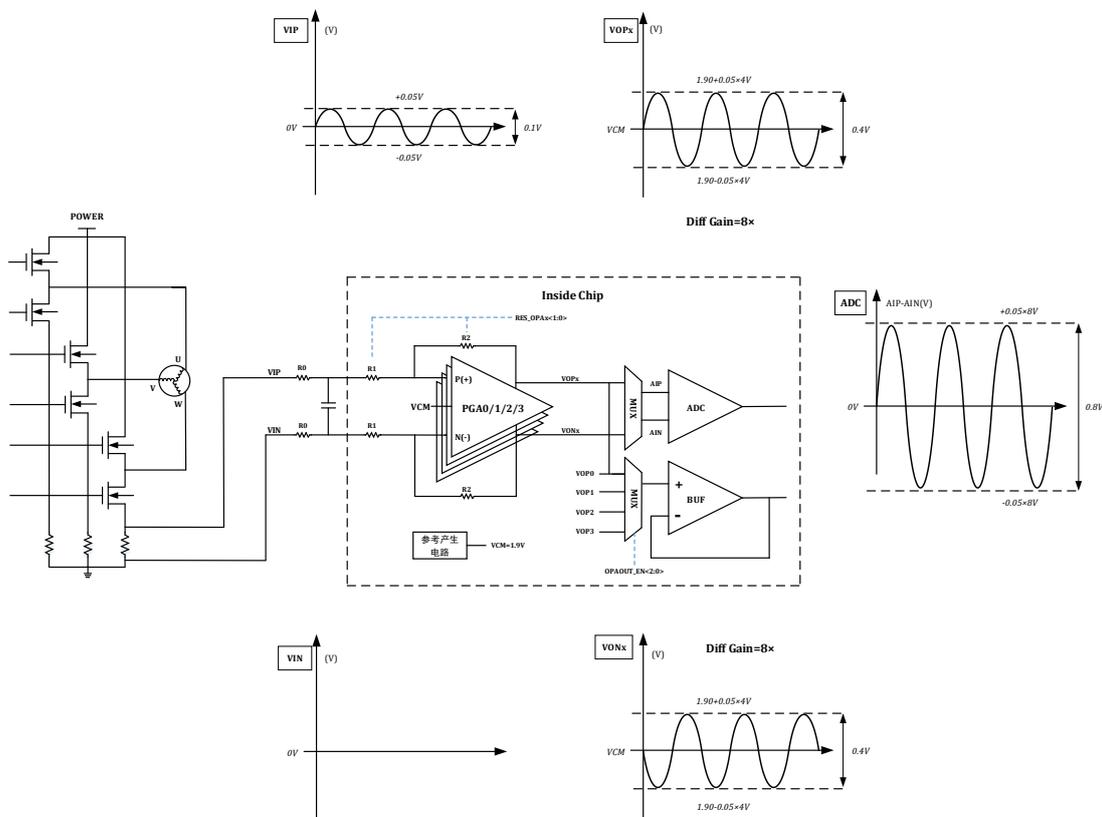


图 1

图 1 中两个 R_0 是片外需放置的电阻，阻值必须相等，最终的放大倍数为 $\text{Gain} = R_2 / (R_1 + R_0)$ ，反馈电阻 $R_2 : R_1$ 的阻值可通过寄存器 $\text{RES_OPAx}[1:0]$ 设置，以实现不同的放大倍数。ADC 的输入信号范围最大为 $\pm 3.6\text{V}$ ，因此运放增益设置的原则是使得最大输入信号时，经运放放大后的信号幅度接近 $\pm 3.6\text{V}$ ，留一定的冗余度即可。

对于 MOS 管电阻直接采样的应用，由于 MOS 下管关断、上管导通时信号会升高到数十 V 的电源电压，为减小此时往芯片引脚里流入的电流，一般建议接 $> 20\text{k}\Omega$ 的外部电阻 R_0 。

对于康铜丝等取样电阻采样的应用，建议接 $100 \sim 1\text{k}\Omega$ 的外部电阻。C0 为信号滤波电容，和 R_0 形成一阶 RC 滤波电路。 R_0 的具体阻值可根据 $R_0 * C_0$ 的滤波常数而定。如果信号上噪声较小不需要滤波、或者信号需要很大的带宽（较快的响应速度），则 C0 可以不加。

如果输入信号 VIP/VIN 之间的共模噪声特别大，则可将 1 个跨接电容 C0 改为 2 个电容到地的电容。如图 2 所示



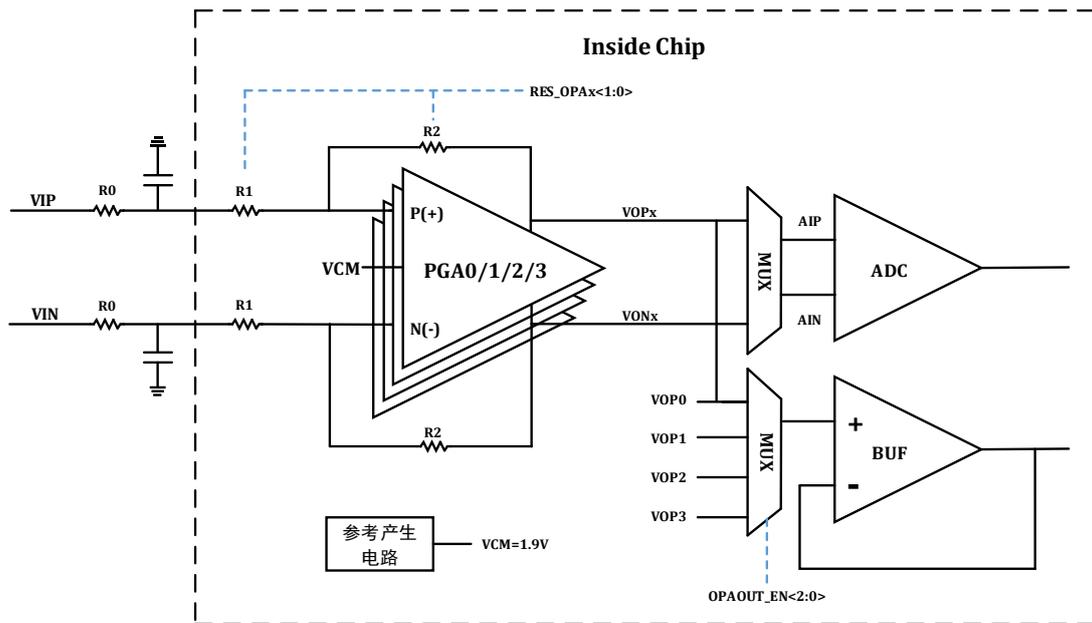


图 2

放大器可通过设置 OPAOUT_EN<2:0>选择将 4 路放大器中某一路的正端输出信号 VOP 通过 BUFFER 送至 P2.7 IO 口进行测量和应用（对应关系见 datasheet 芯片管脚说明）。因为有 BUFFER 存在，在运放正常工作模式下也可以选择将其 VOP 信号出来，此时该运放相当于同时作为差分运放和单端运放使用。

2 差分工作模式

运放输出的差分信号： $V_{sig}=V_{OPx}-V_{ONx}=Gain*V_{in}+Gain*V_{offset}$ 其中： $V_{in}=V_{IP}-V_{IN}$ ； $Gain=R2/(R1+R0)$ ； $Gain*V_{offset}$ 为运放经过放大后的零漂，对于应用来说 $Gain*V_{offset}$ 这项需要剔除。

处理方案：

在芯片上电之初，电机尚未工作时，运放输入信号为 $0V$ ，此时通过 ADC 对该路运放进行采样，采到的值就是 $Gain*V_{offset}$ ，软件上将其保存下来，之后读到的该路运放的采样值，都减掉保存的运放零漂。

使用到的每一路运放，都需要保存其相应的零漂值

运放输出信号送给 ADC 采样的方式，是属于差分工作模式。差分工作模式的优点是精度高、抗干扰能力强，差分模式是芯片的默认工作模式。



3 单端工作模式

除差分模式之外，运放还有两种单端应用模式：

1) 将某路运放的正输出端 VOP，通过设置 CMPx_SEL P<2:0>(x=0/1)送至比较器 0/1 的正输入端，比较器的负输入端通过设置 CMPx_SEL N<1:0>(x=0/1)连至 DAC 模块的输出，这种应用方式，可将放大后的信号做过流保护使用。

2) 通过设置 OPAOUT_EN<2:0>将 4 路放大器中某一路的正端输出信号(VOP)通过 BUFFER 送至 P2.7 IO 口，控制芯片外部的某些模块，或者将该信号在芯片外做滤波后得到信号平均值，重新送到芯片的另一个 ADC 输入通道管脚，由 ADC 去测量滤波后的信号。

这两种应用方式由于都只利用了运放输出正端信号 VOP，属于将差分运放作为单端运放应用。单个运放可以同时作为差分运放和单端运放来使用，即运放输出信号被 ADC 采样的同时，将正端输出 VOP 连至比较器，或送至 P2.7 IO 口使用。

VOP/VON 的信号公式为：

$$VOP = V_{cm} + V_{sig}/2$$

$$VON = V_{cm} - V_{sig}/2$$

其中, V_{cm} 是运放输出的共模电压，一般为 1.9V， V_{sig} 为运放输入信号经放大后的差分信号。假设某个输入信号下，放大后的 $V_{sig}=2V$ ，则 VOP 为 2.9V，VON 为 0.9V。

可以看出，对于差分应用模式， V_{cm} 的值没有影响。但是对于单端应用模式， V_{cm} 的作用则不可忽视。图 3 以系统 8 倍增益为例，说明各信号节点的增值大小。



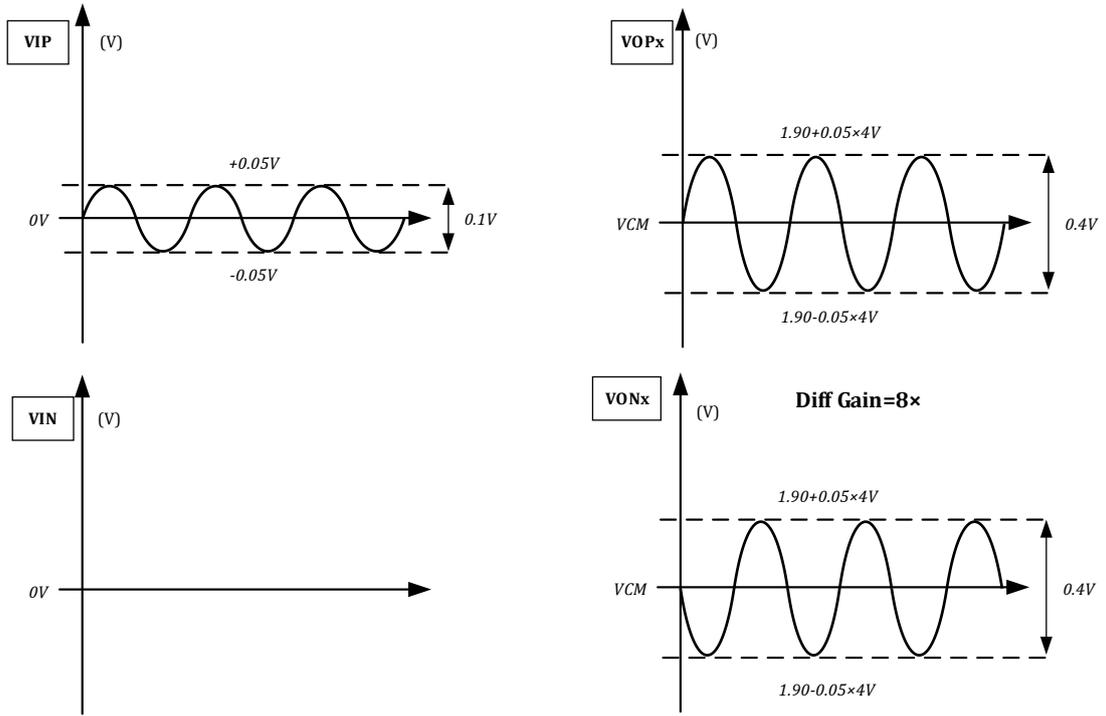


图 3

3.1 单端模式 V_{cm} 的校正

对于不同芯片运放的 V_{cm} ，有离散性，大批量时，相比于 1.9V 的平均值，最大偏差可达 250mV。因此有必要对 V_{cm} 做校正。校正方式：

在芯片上电之初，电机未运行时，运放输入信号为 0V，此时将作为单端模式应用的那路运放 VOP 通过配置 OPAOUT_EN<2:0>送至 P2.7 IO 口，P2.7 IO 口是 ADC_CH11，用 ADC 对 CH11 进行采样，采样得到的值即为 V_{cm} 对应的 ADC 值。实际的物理值(即多少 V)则可根据 ADC 值计算得到。

如果 P2.7 IO 口在 PCB 上有阻容滤波，如图 4 所示：

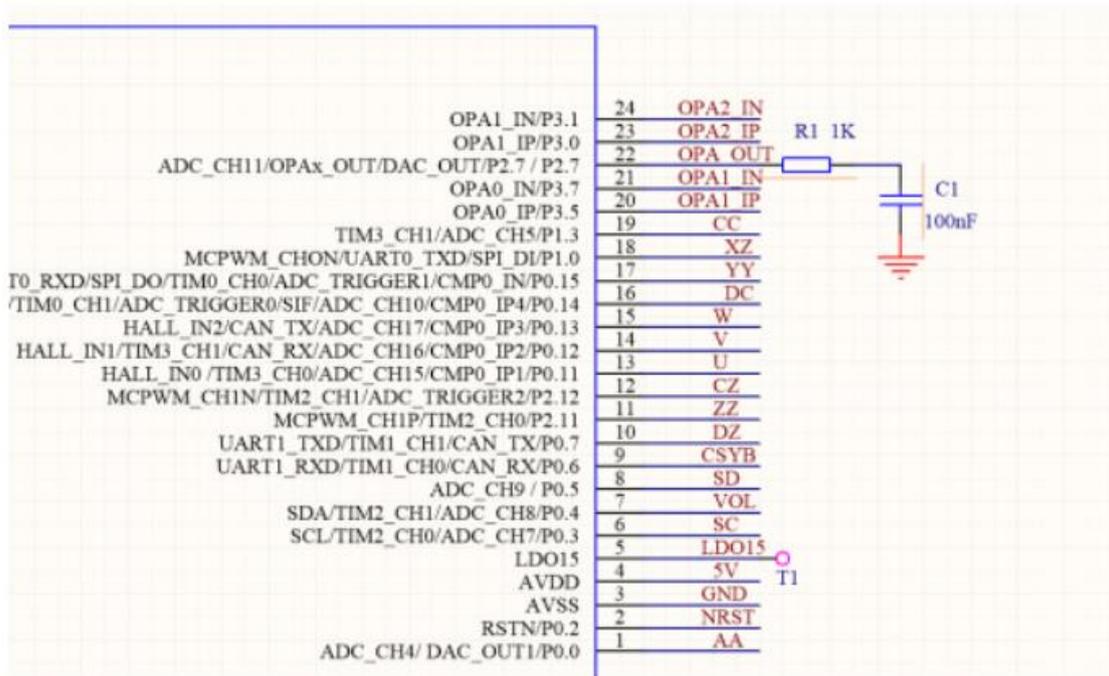


图 4

则将运放 VOP 信号送出后,P2.7 的信号要经过一段时间才能达到稳定,稳定时间的计算公式为： $T=6*(1000+R)*C$ 。

以图 4 中 $R1=1k\Omega$ ， $C1=100nF$ 来计算，需等待的时间为 1.2ms。如果片外没有电阻 $R1$ ，则取 $R=0$ 即可。需留意的是，VOP 信号送至 P2.7 IO 口的话，由于 VOP Buffer 的电流驱动能力有限，该 IO 口对地的阻抗需要大于 40k。图 3 中的电阻 $R1$ ，因为后面对地还有电容 $C1$ ，因此对地阻抗满足要求。

上电之初 ADC 采集到的 V_{cm} 在软件上保存下来，对于上述两种单端应用模式：

1) VOP 送至比较器的正输入端，DAC 信号送至比较器负输入端：



对于要做保护的阈值 V_{Hsig} （即运放放大后的信号 V_{sig} 超过 V_{Hsig} 后比较器应翻转），VOP 的信号是 $V_{cm}+V_{Hsig}/2$ ，因此连至比较器负端的 DAC 值应设置为 $V_{cm}+V_{Hsig}/2$ ，其中 V_{cm} 采用上面软件保存的值所对应的物理值，物理值和 ADC 值之间的换算参看 user manual 的 ADC 章节。

2) VOP 送至 P2.7IO 口，控制芯片外的模块，或者将该信号在芯片外做滤波后得到信号平均值，重新送到芯片的另一个 ADC 输入通道管脚，由 ADC 去测量滤波后的信号：ADC 采集到的做过滤波的 VOP 信号值，都需要减掉软件保存的 V_{cm} ，得到的是 $V_{sig}/2$ ，即运放输出信号的一半大小。



3.2 单端模式 V_{cm} 随温度的变化

上述部分校正了不同芯片之间运放共模电平 V_{cm} 的差异,但 V_{cm} 随温度还有约 $-0.5\text{mV}/^{\circ}\text{C}$ 的变化。即 125 度相比 25 度, V_{cm} 将下降 50mV。如果应用上要求精度较高,则有必要修正 V_{cm} 的温度变化量。LKS07x、LKS09x 设计上已做优化,不需要做温度补偿。

校正方式:

在芯片上电之初,将运放 VOP 送至 P2.7 IO 口进行 ADC 测量得到 V_{cm0} 的同时,测量芯片内部温度传感器的温度(详见 user manual 温度传感器章节) T_0 。后面电机运行过程中,每隔几秒或者 1 分钟,测量一次温度值 T_1 ,计算温度差 $VT=T_1-T_0$ 。此时的运放共模电压 $V_{cm}=V_{cm0}+VT*(-0.5\text{m})$ 。

对于上述两种单端应用模式,应采用随温度更新过的 V_{cm} :

1) VOP 送至比较器的正输入端, DAC 信号送至比较器负输入端:

对于要做保护的阈值 VH_{sig} (即运放放大后的信号 V_{sig} 超过 VH_{sig} 后比较器应翻转), VOP 的信号是 $V_{cm}+VH_{sig}/2$, 因此连至比较器负端的 DAC 值应设置为 $V_{cm}+VH_{sig}/2$, 其中 V_{cm} 采用上面软件保存的值所对应的物理值, 物理值和 ADC 值之间的换算参看 user manual 的 ADC 章节。

2) VOP 送至 P2.7 IO 口, 控制芯片外的模块, 或者将该信号在芯片外做滤波后得到信号平均值, 重新送到芯片的另一个 ADC 输入通道管脚, 由 ADC 去测量滤波后的信号: ADC 采集到的做过滤波的 VOP 信号值, 都需要减掉软件保存的 V_{cm} , 得到的是 $V_{sig}/2$, 即运放输出信号的一半大小。

但即使做过温度修正, 运放单端模式的精度仍然不如差分模式, 对于需要精确测量小电流信号场合, 建议还是使用单独一个运放的差分模式。

