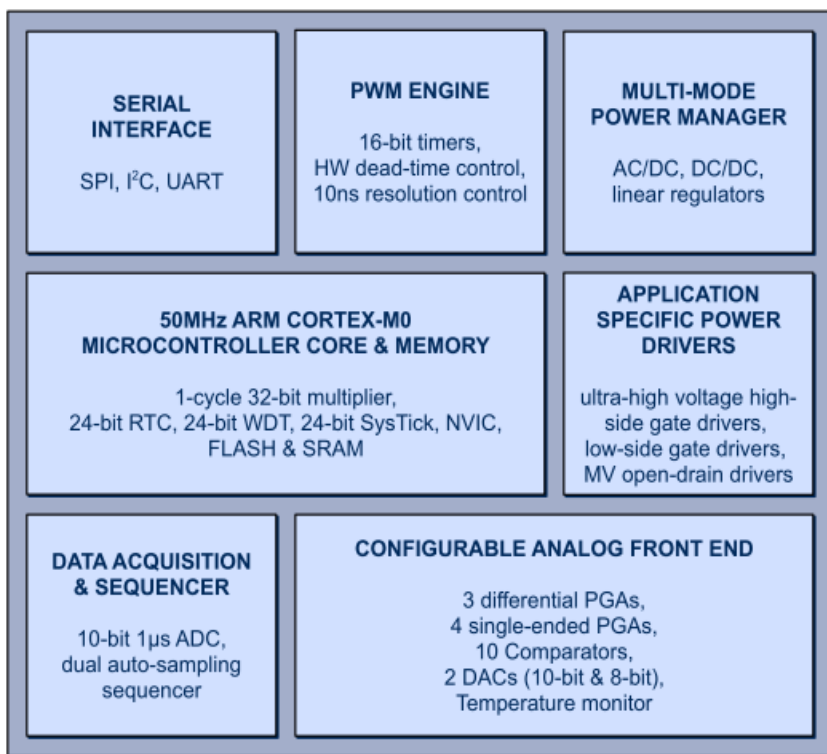


# PAC52xx 系列产品介绍及使用Tips

模拟电路工程师  
夏德刚

2017年3月

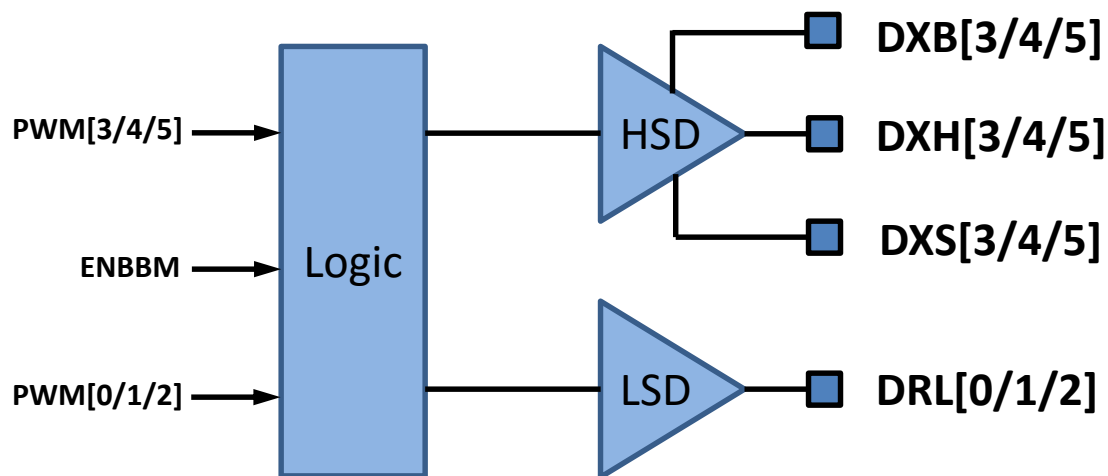
集 **MCU**，串口通信，**PWM引擎**，自动数据采集，多模式电源管理，栅极驱动，可配置模拟前端及保护 于一体



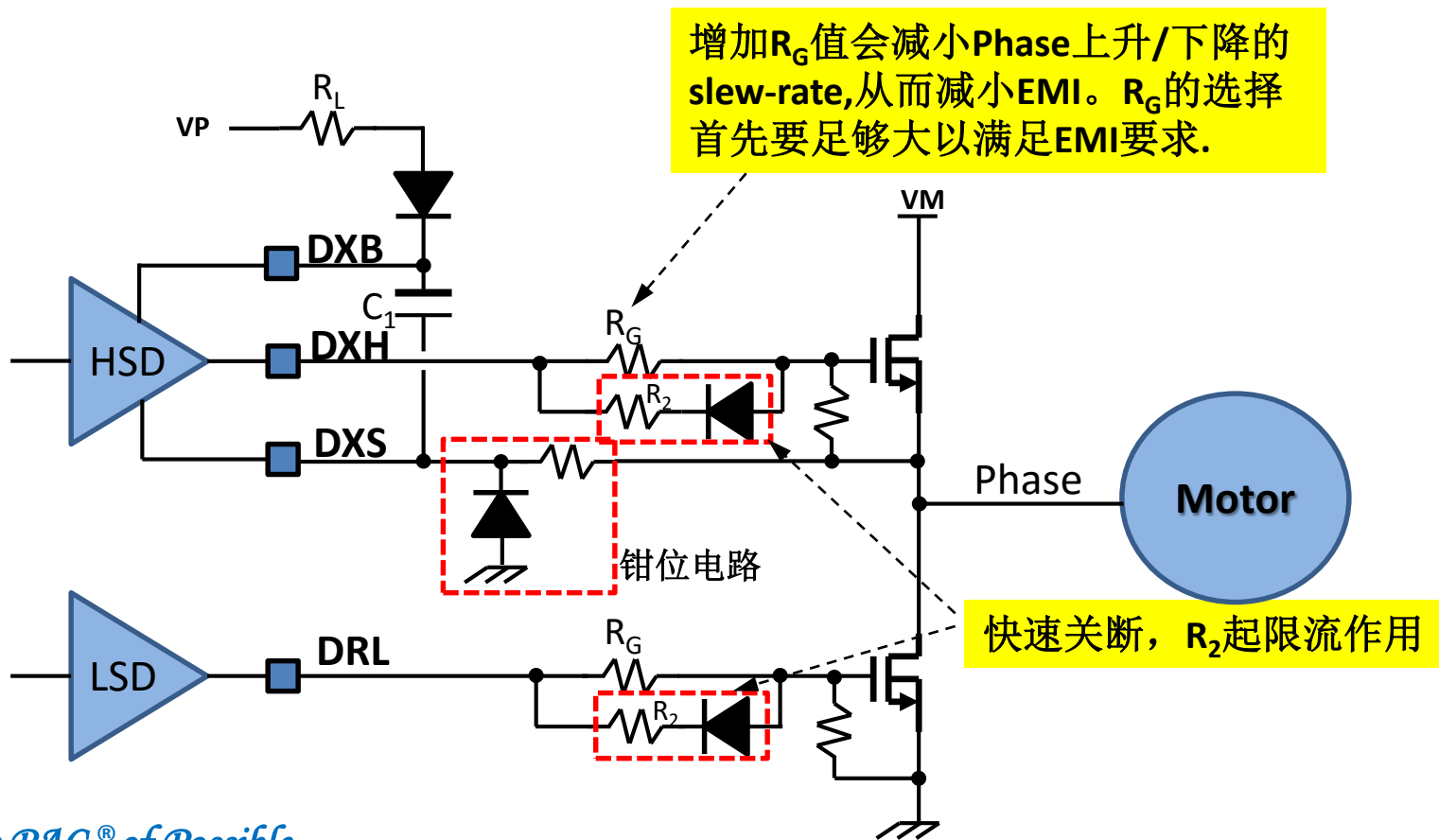
## 优点:

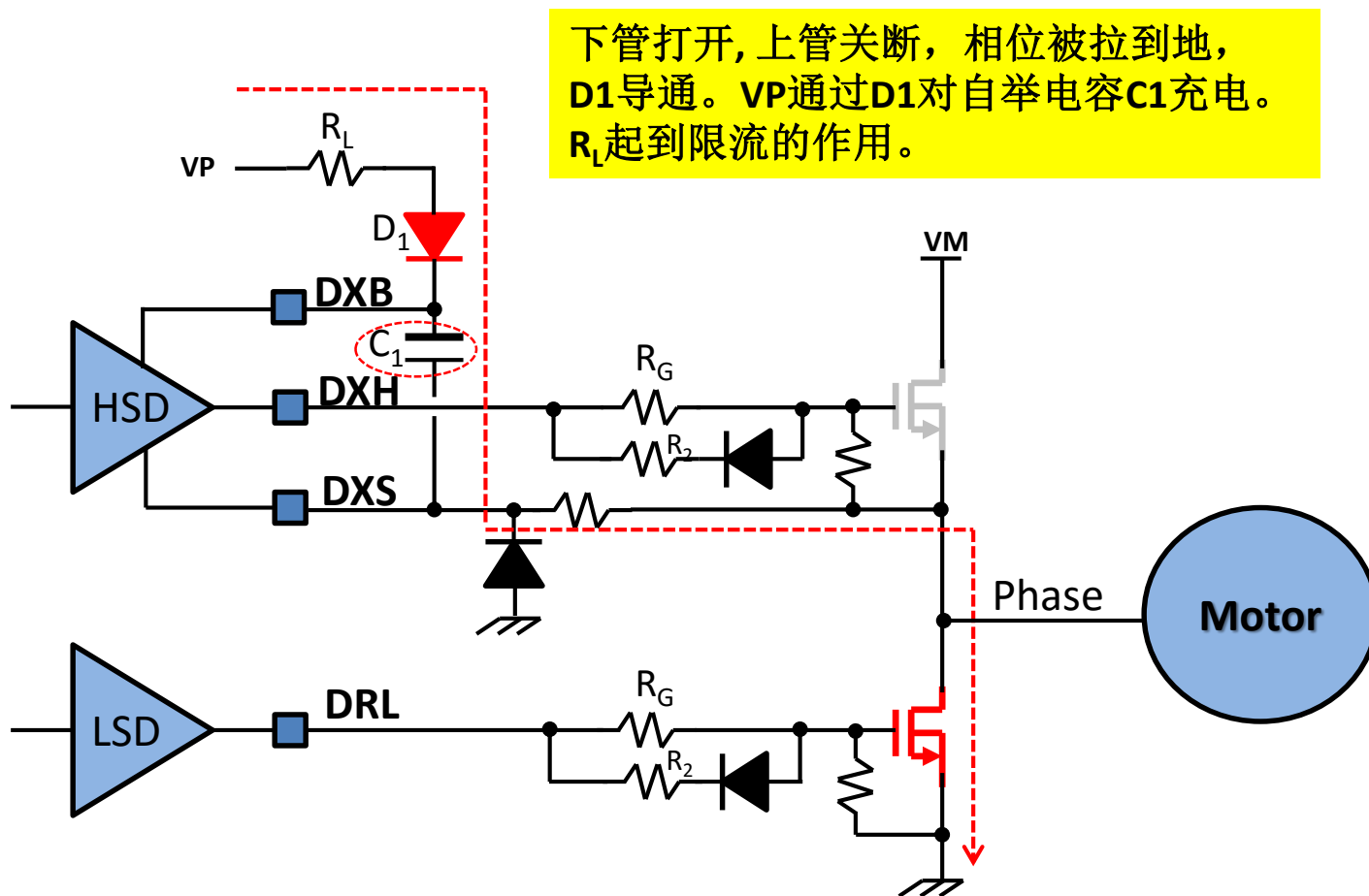
- ❑ 宽广的马达电压范围，高度集成的SOC芯片，为客户提供一个平台
- ❑ 实现快速自动数据采集，把Cortex-M0性能发挥到极至，为相对复杂的驱动算法提供高性价比方案
- ❑ 多模式电源管理提供一站式服务
- ❑ 简约灵活的可配置模拟前端提供许多种可能性
- ❑ 集成了多种保护功能
- ❑ MCU可实时监测芯片内部工作情况

- ❑ PAC提供三个半桥驱动
- ❑ PWM引擎产生6路PWM信号, PWM3与PWM0配对控制半桥的上下管, 4与1, 5与2配对
- ❑ 用16-bit来控制配对的PWM信号之间的死区时间, 分辨率为10纳秒.
- ❑ **ENBBM(BBM=Break Before Make)**从逻辑上保证半桥的上管和下管不能同时打开以防止贯穿电流
- ❑ 为保证产品的可靠性, 所选芯片的耐压应为马达电源电压的2-3倍

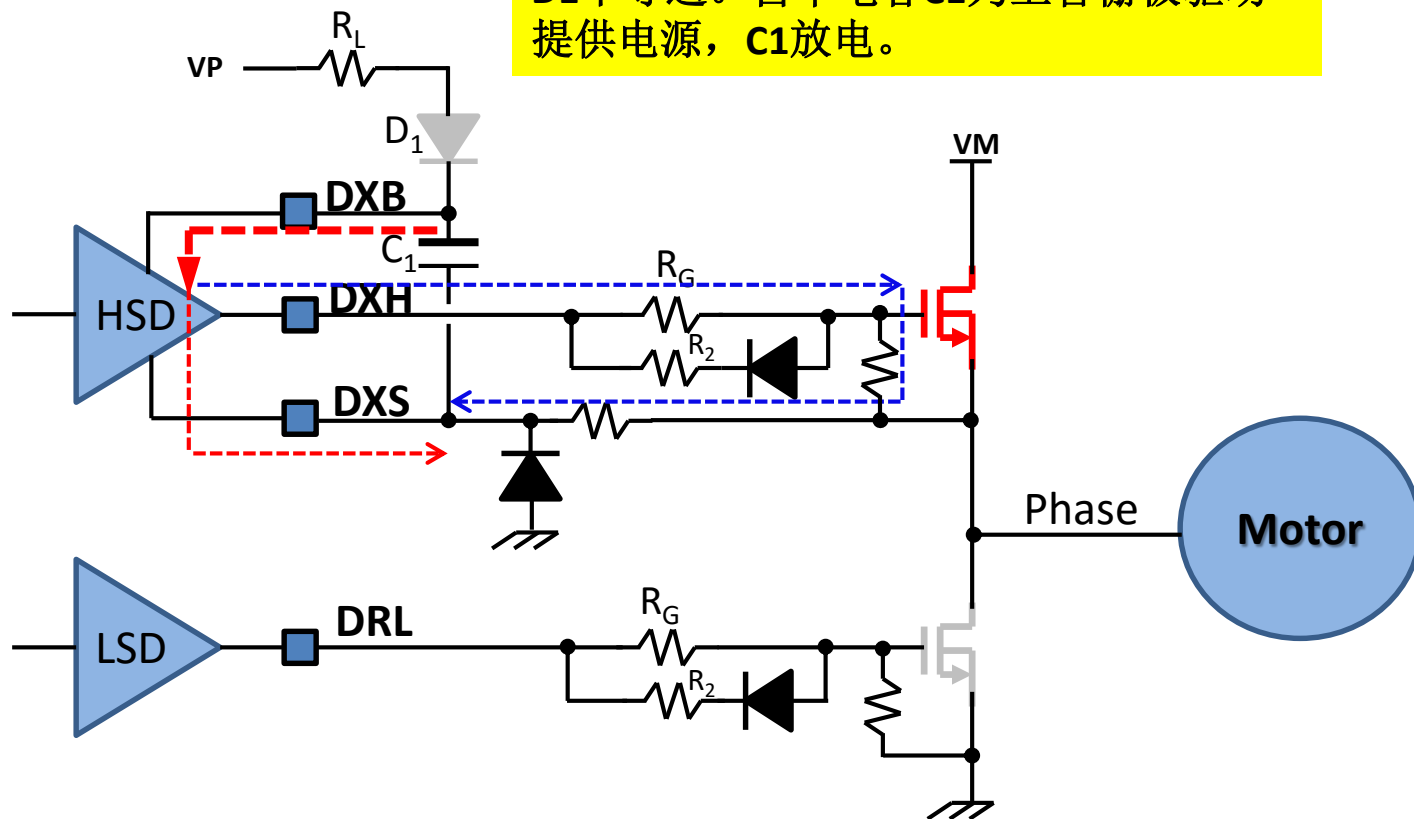


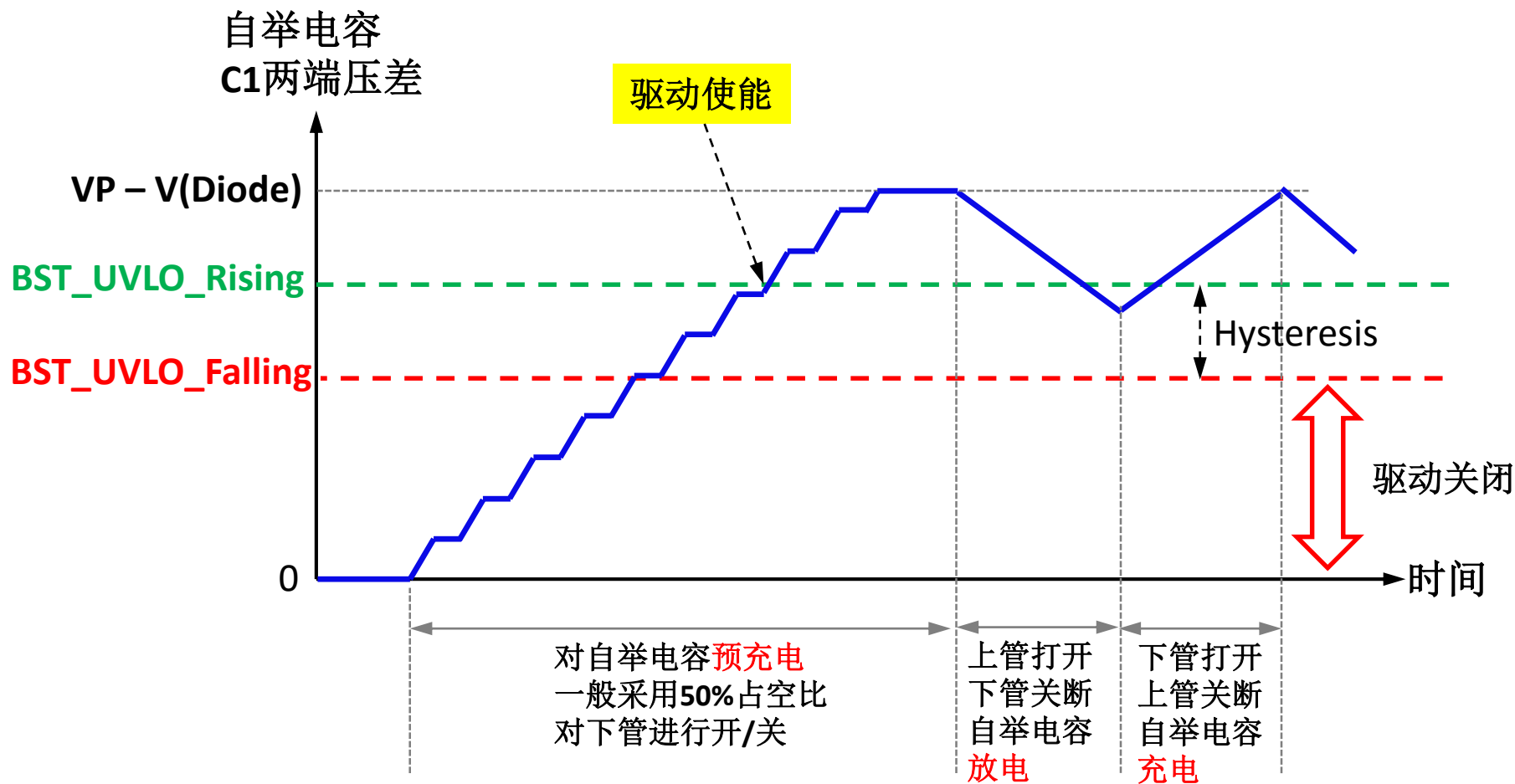
- ❑ 上下管驱动输出端加栅极电阻 $R_G$ 来调整功率管打开速度以减小EMI, 同时防止高频振荡
- ❑ 并联一个二极管来提高关闭速度
- ❑ 版图上栅极电阻尽量靠近功率管
- ❑ 根据应用需要加嵌位电路防止DXS有太大的负压, 版图上靠近芯片





下管关断，上管打开，相位被拉高到VM。  
D1不导通。自举电容C1为上管栅极驱动  
提供电源，C1放电。

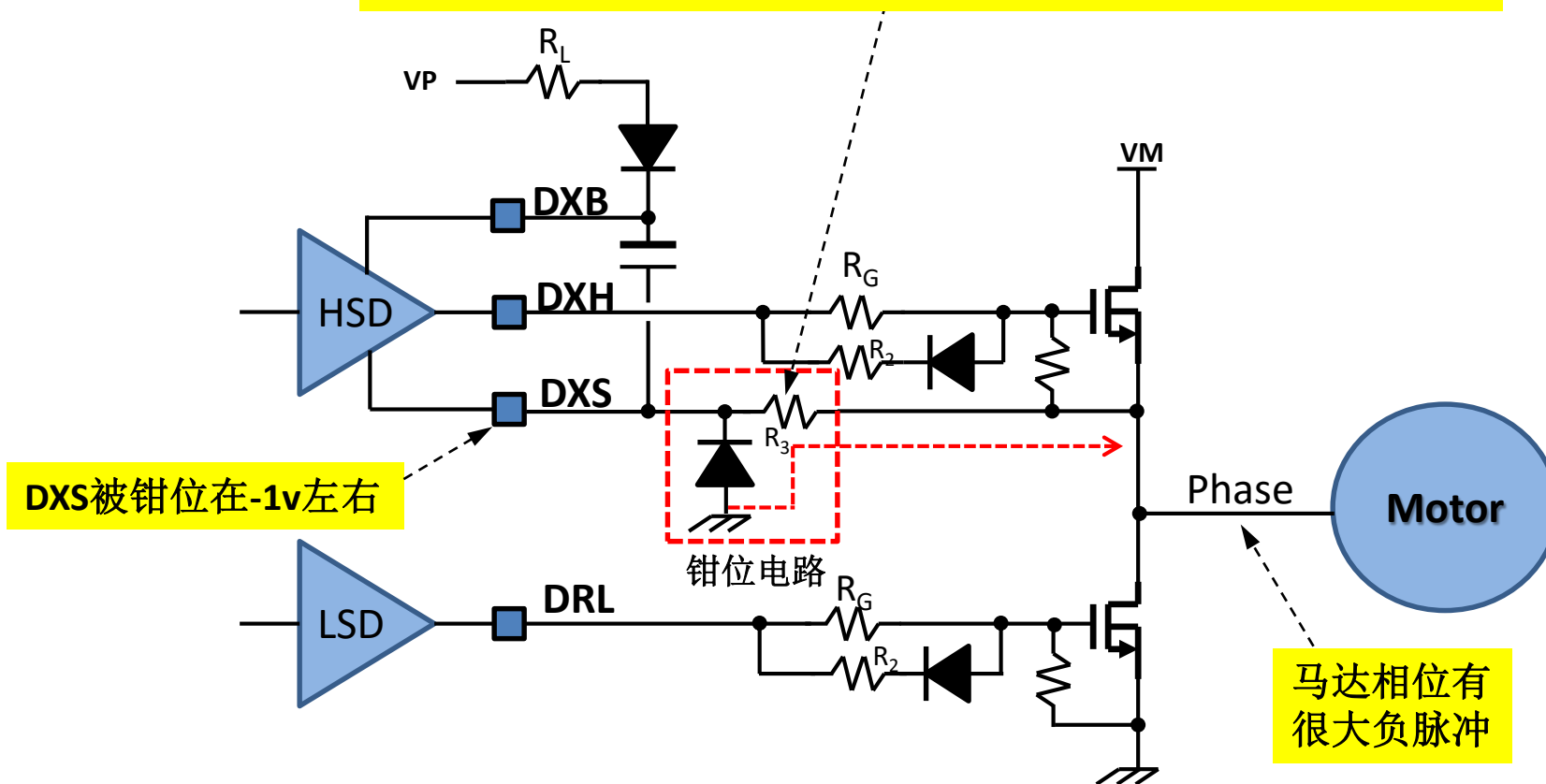


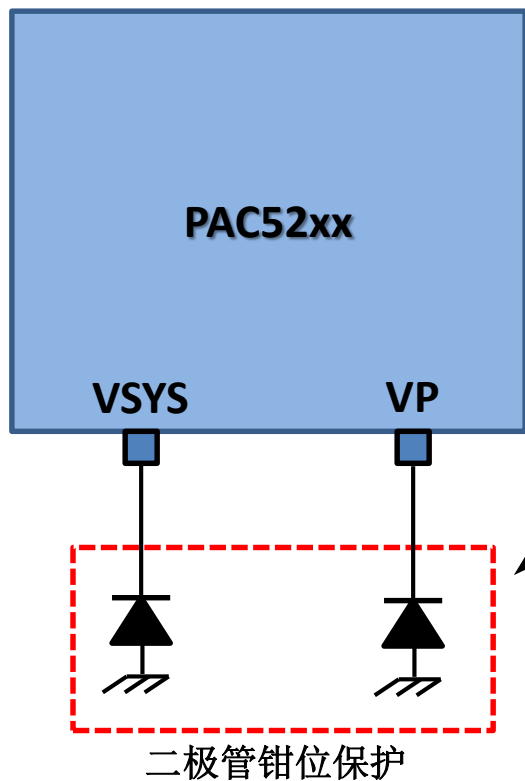


- ❑ 自举电容的值至少是功率管栅极电容(  $C_{iss}$  )的10倍
- ❑ 若PWM频率较小，自举电容放电时间相应增加，自举电容的值也应增加
- ❑ 在正常情况下，选择适当的值使得自举电容两端压差谷底值大于“BST\_UVLO\_Falling”，并有足够余量
- ❑ 在最坏情况下，自举电容两端的压差不小与 “BST\_UVLO\_Falling”



**R3电阻不能大，否则会影响正常工作。一般情况下不大于 $5\Omega$ 。假如负脉冲有 $-5v$ ， $R3=5\Omega$ 。 $R3$ 电流为 $[(-1) - (-5)]/5=0.8A$ ， $R3$ 瞬间功率达到 $(0.8^2 \times 5=3.2w)$ 。若钳位周期性发生， $R3$ 产生许多焦耳热。所以 $R3$ 电阻的封装必需具有较大的散热能力，否则很容易被损坏。**



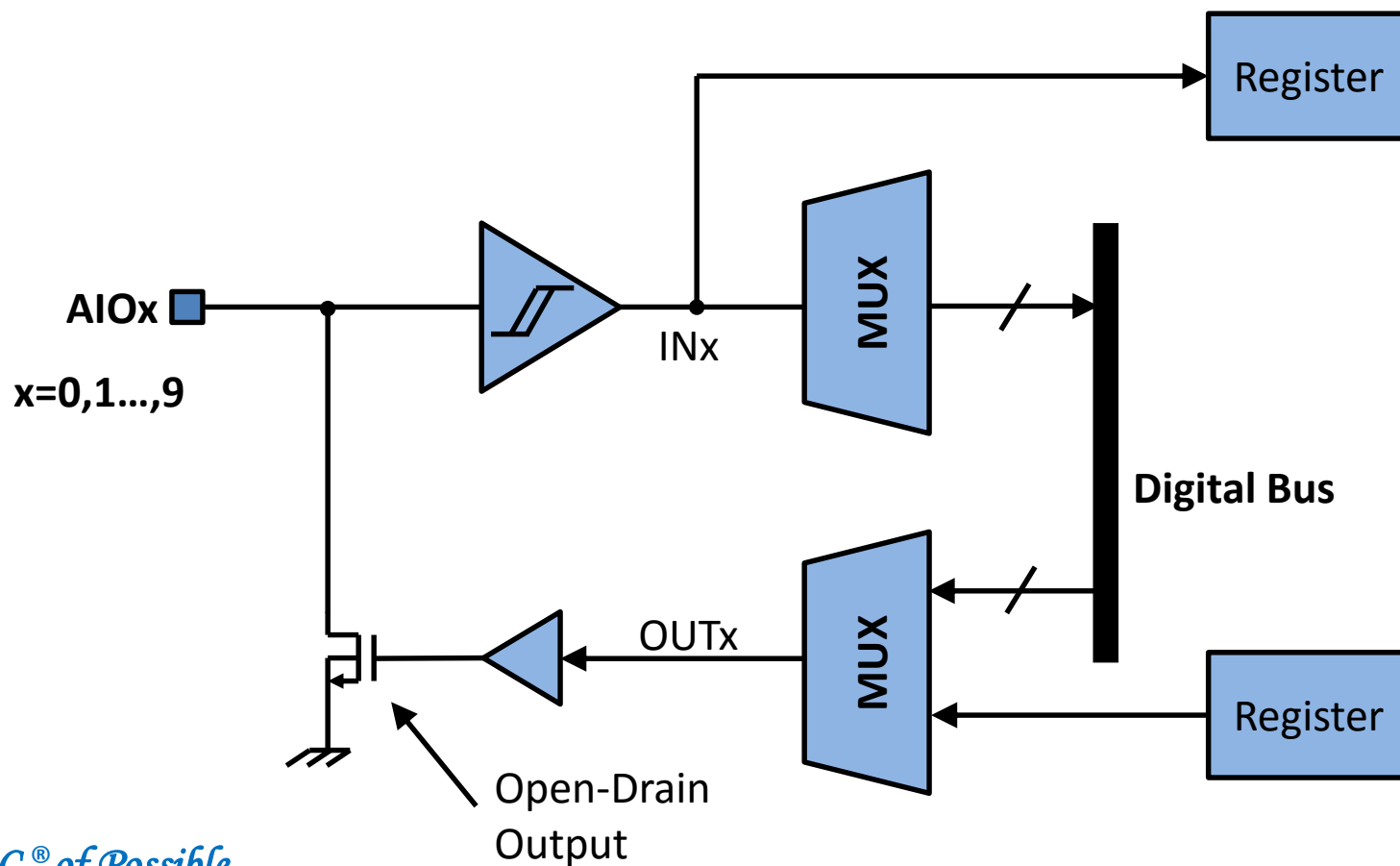


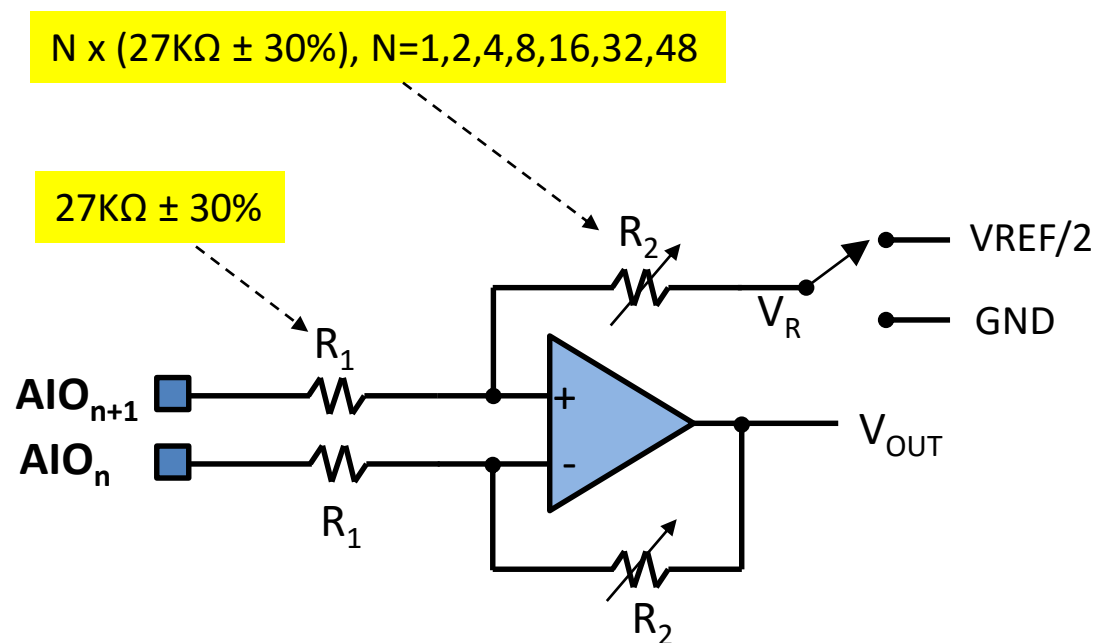
在做极端条件下的可靠性测试时发现当用1米长的导线使VP或VSYS对地短路，VP或VSYS会产生很大负压从而损坏芯片。在线路板上加两个二极管使得VP/VSYS被钳位在-1v左右从而保护芯片。

这个保护是个选项，不是必需的。

端口	数字 输出/输入	增益	比较器	特殊模式
AIO[5:0]	GPIO	DIFF AMP		
AIO6	GPIO	GAIN	COMP	Unit Gain Buffer Output
AIO[9:7]	GPIO	GAIN	COMP	Sensor-less

- ❑ AIOx(x=0,1,...9) 在GPIO模式下可配制成“输入”或“输出”
- ❑ 在输入模式下, 可把输入信号通过MUX配置到任意一条数字总线
- ❑ 在输出模式下, 可任意一条数字总线信号通过MUX配置到输出口
- ❑ 各个端口之间通过数据总线相互连通

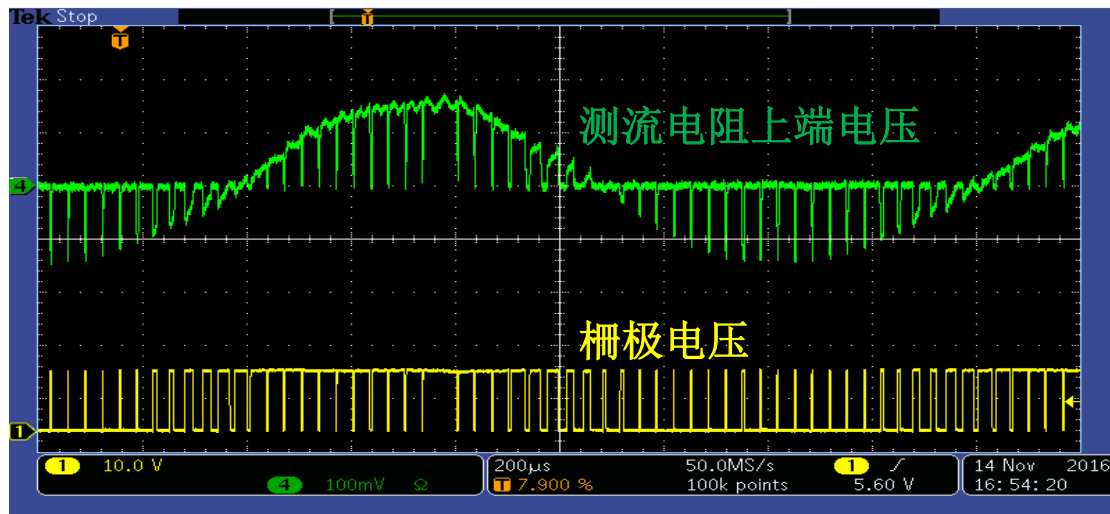
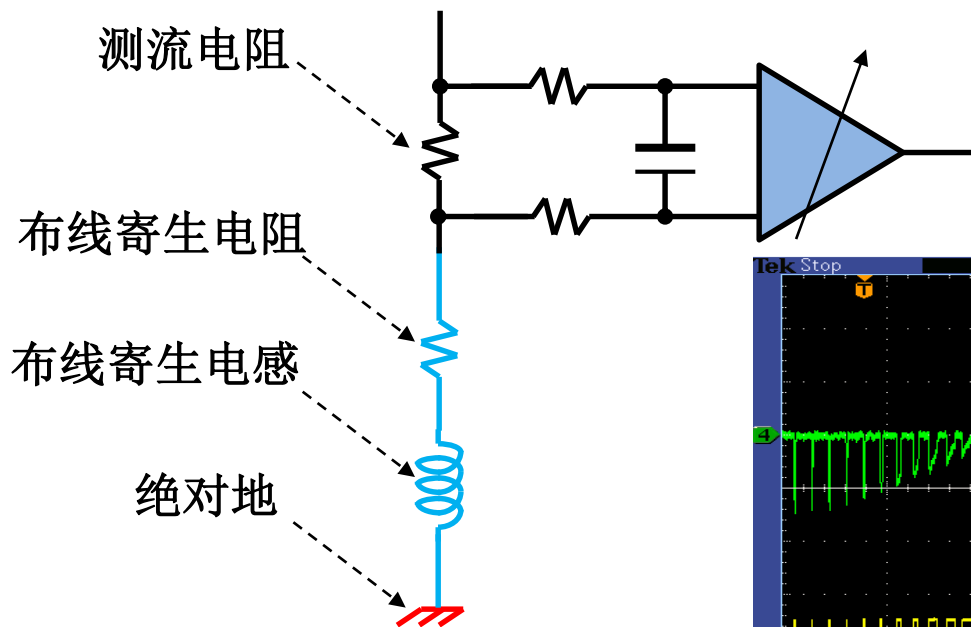


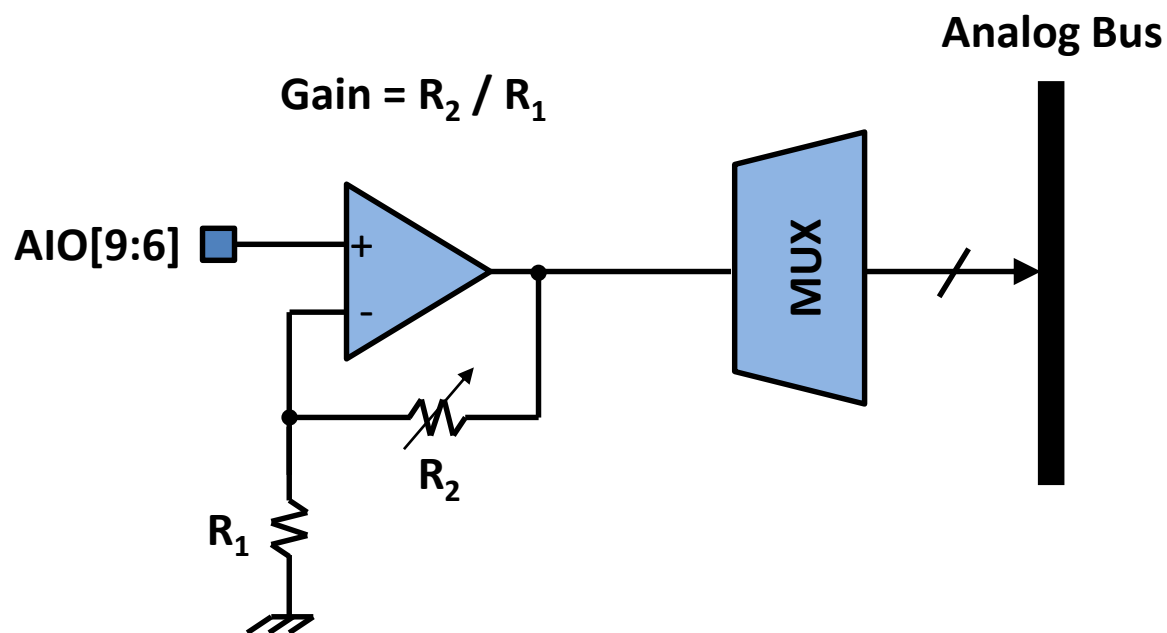


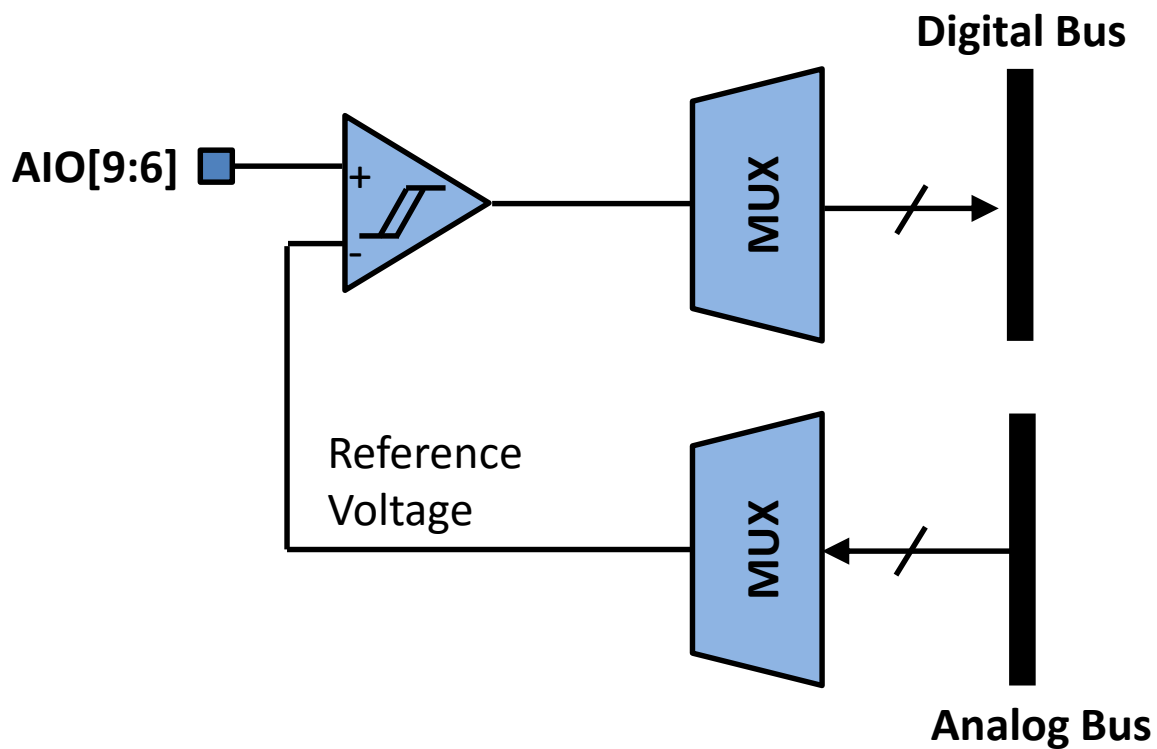
$V_R = "V_{REF}/2" \text{ or } "0"$   
大多数的应用选择 " $V_{REF}/2$ "

$$V_{OUT} = V_R + (AIO_{n+1} - AIO_n) \cdot (R2/R1)$$

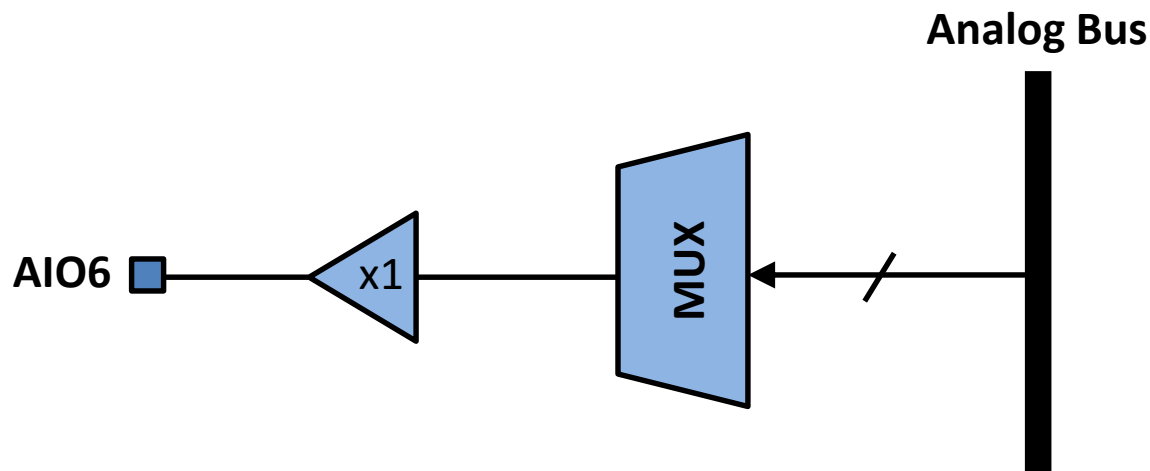
- ❑ 马达驱动是一种开关电路，测流电阻两端有相当大的由开关电路产生的共模噪音。
- ❑ 差分放大器具有较高的共模抑制比，从而最大限度消除共模噪音。
- ❑ 所测电流与PCB板寄生参数无关。
- ❑ 差分放大器提供高精度增益，为变频算法提供高精度的电流测量，同时提供高精度的过流保护。
- ❑ 任何依赖于高精度电流测量的算法都将受益于它。
- ❑ 可调增益的功能使得客户能灵活选择测流电阻。

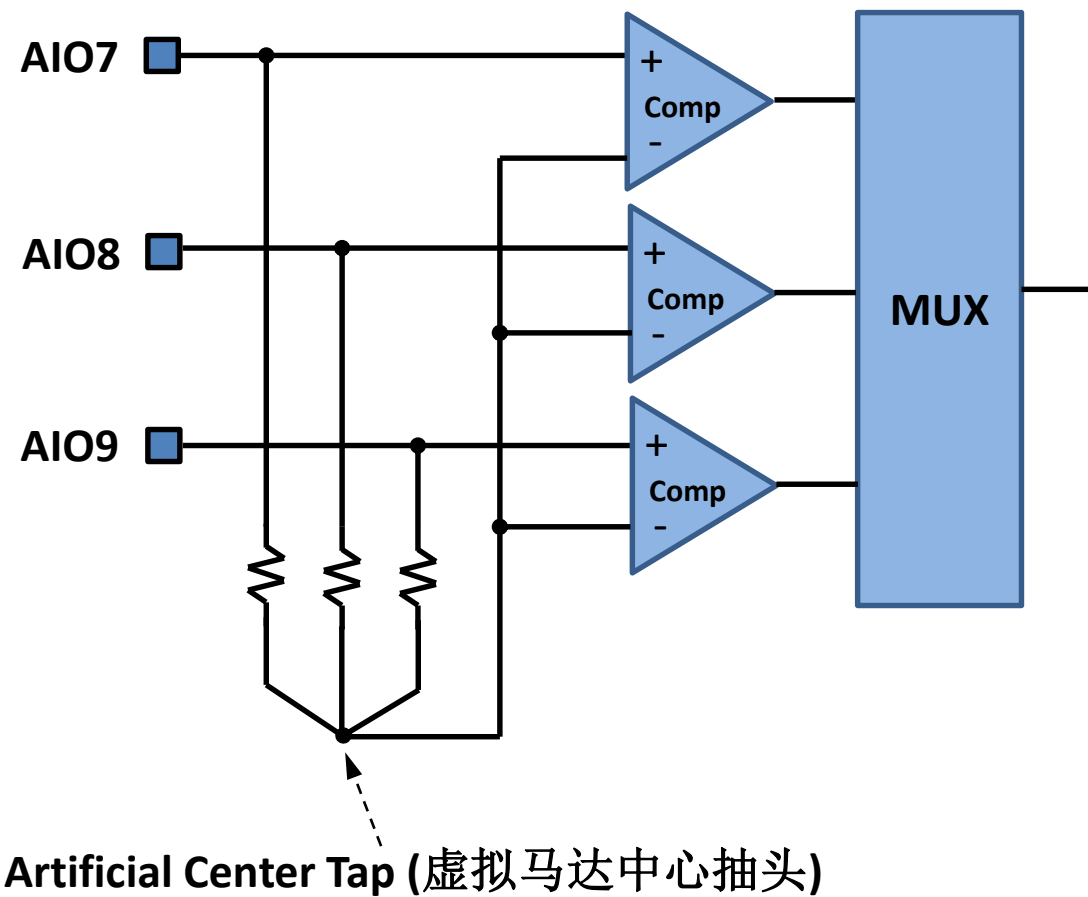




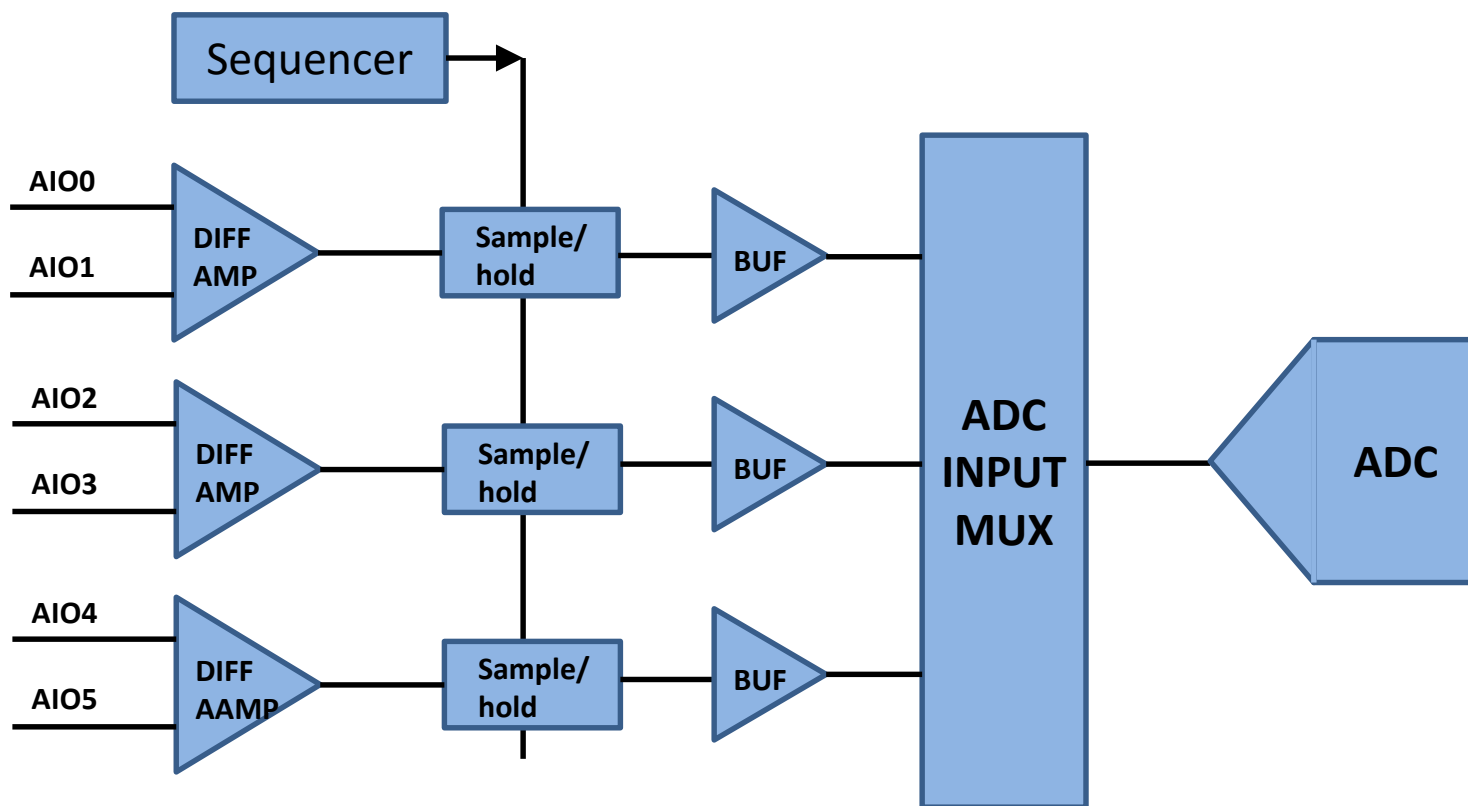




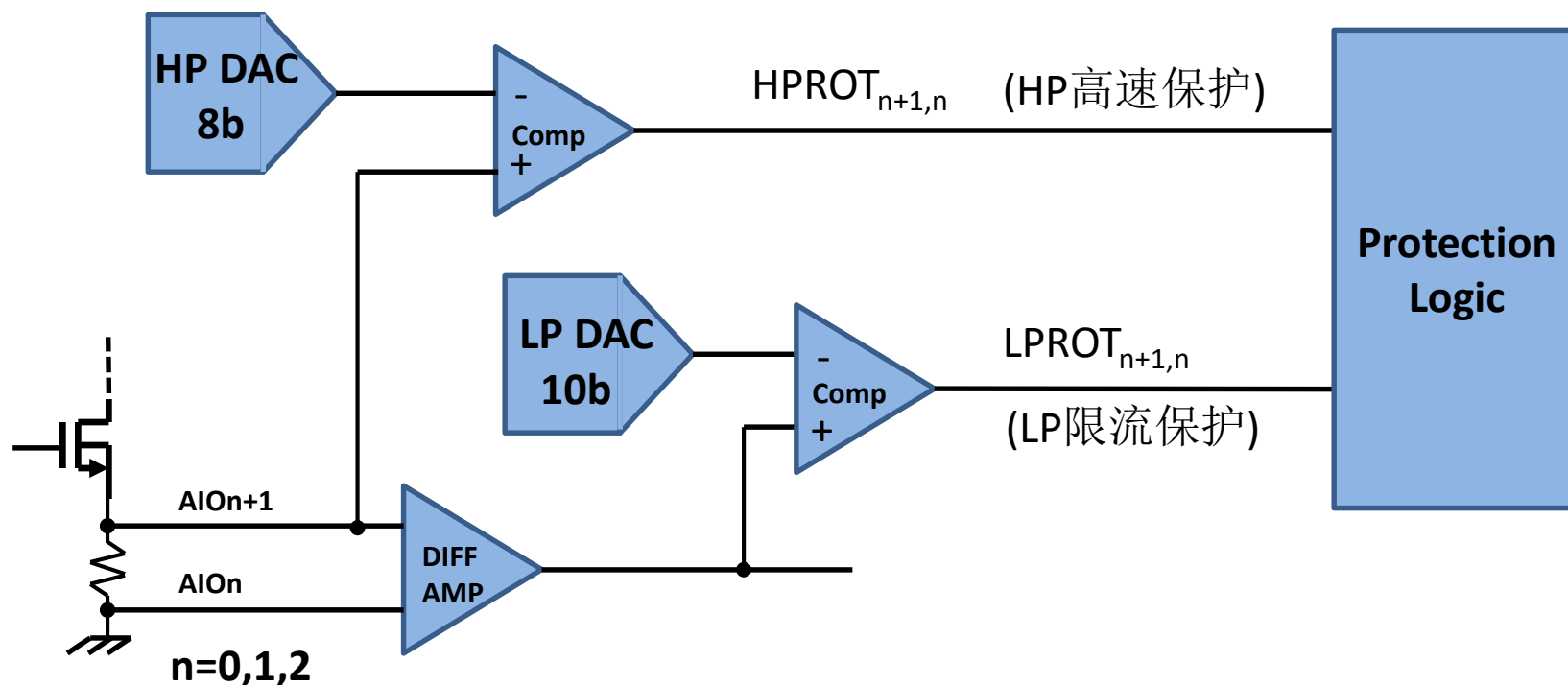




PAC提供三路精确的,可调增益的电流测量差分放大器,与之配套的还有三路采样/保持电路,可同时获取马达三相电流,为变频(FOC)控制提供所需的电流信息。采样/保持电路的时间由Sequencer灵活快速控制。



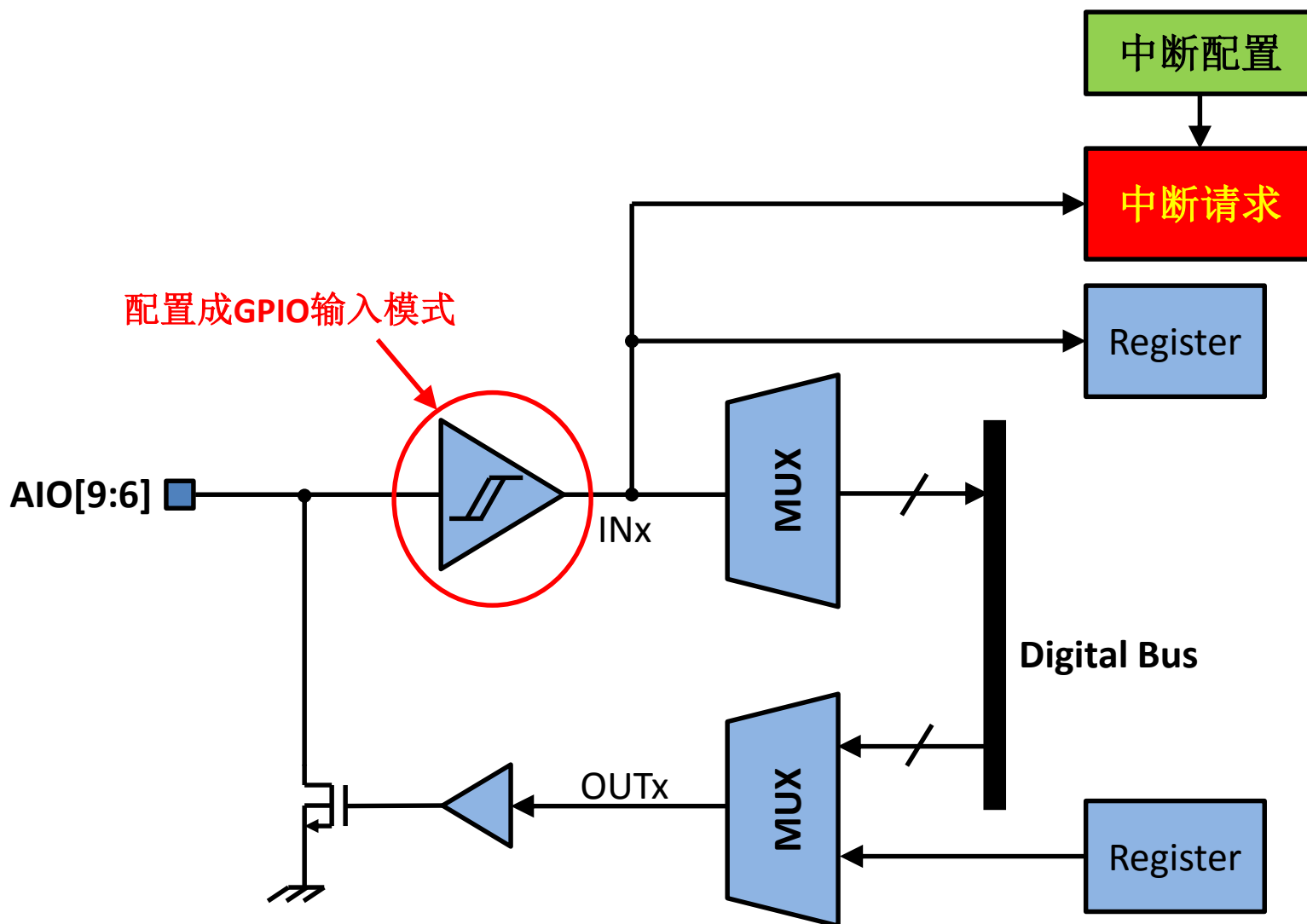
对于每对电流测量管脚，有二个特定比较器来做过流保护。比较器阈值分别由一个人8位及10位的DAC来设定。LP 保护精度高，被广泛应用。



- ❑ **PAC52xx**产品系列有个**4MHz**的时钟, **Class-B**安规要求另外一个独立时钟以便于二个独立时钟相互监督。
- ❑ **PAC5255**可提供一个额外的独立低频时钟
- ❑ 除了**PAC5255**, 别的**PAC52xx**产品可用外挂晶振产生独立时钟。
- ❑ 若交流电为电源可利用其固有频率来产生**50Hz**或**60Hz**的独立时钟

- ❑ MCU写一个特定的寄存器位, 便使芯片进入最省电的深度睡眠模式
- ❑ 深度睡眠模式中, 开关电源 (Buck/Fly-back) 电路停止工作, MCU和所有电源 (LDOs) 都停止工作。
- ❑ 芯片只需20uA的电流来维持深度睡眠状态
- ❑ 芯片具有自动唤醒功能, 时间可预先设定
- ❑ AIO6 Push Button功能可唤醒芯片( 该功能被广泛地应用于电冰箱中 )。当AIO6被拉到地并维持大于30ms时间, 芯片即被唤醒。唤醒后芯片经历和初次上电一样的过程。

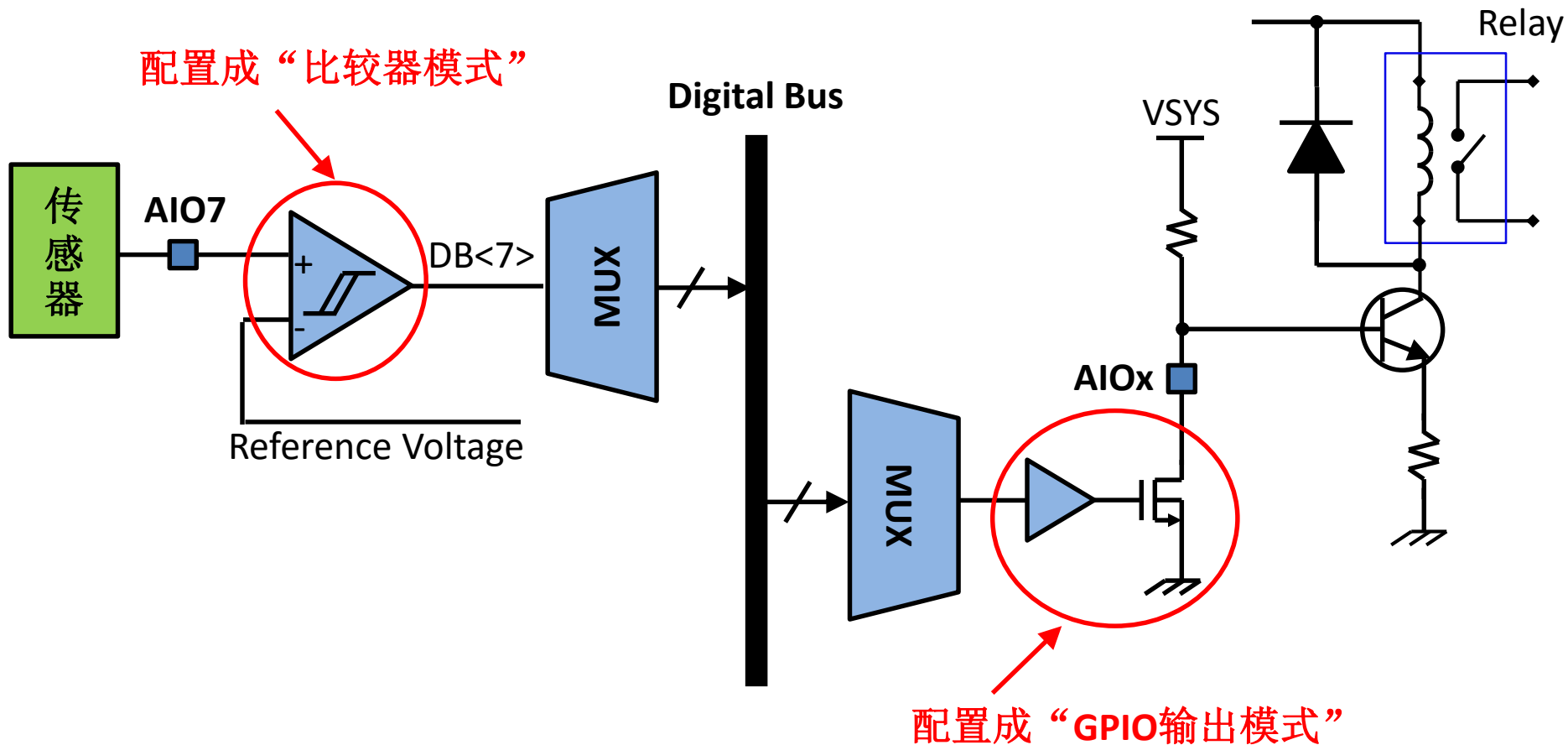
- ☐ **GPIO**
- ☐ **Push Button**产生中短请求
- ☐ **Push Button**唤醒芯片, 若芯片处于深度睡眠模式
- ☐ **Push Button**( $\geq 8$ 秒) 初始化芯片
- ☐ 比较器的输入端
- ☐ 可输出恒定参考电压, 并有一定的电流驱动能力



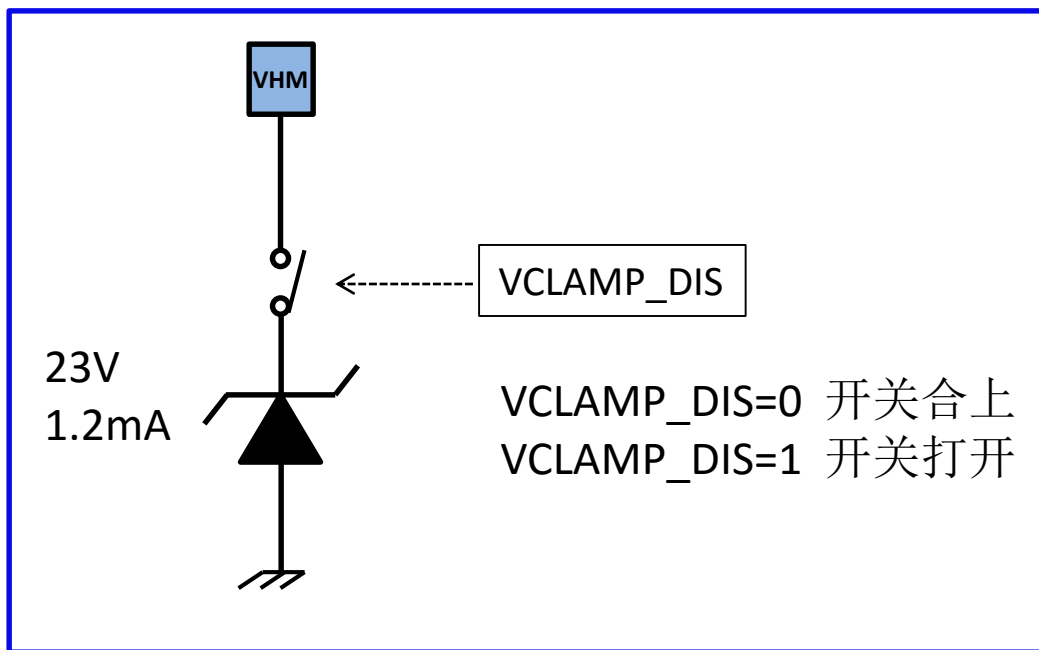


AIO7联接一个传感器的输出，通过比较器监视其电压，当该电压大于特定阈值时，AIOx(x≠7)端口发出一个信号，该信号可用来控制一个外围器件，如Relay。

配置成“比较器模式”



VHM是芯片的电源输入端, 根据输入电压的不同, 可以对它进行不同的配置。PAC5250, PAC5253, PAC5255的VHM耐压是50V, PAC5223是70V。如下图所示, 芯片内部与VHM相联的有一个可以使能的大约23V的钳位电路, 当它工作时能拉约1.2毫安左右的电流。上电及芯片初始化后, 使能位VCLAMP\_DIS=0, 开关合上也就是说钳位电路处于工作状态。



电源通过启动电阻对启动电容充电，当VHM电压超过14.5V时，芯片启动开关电源。  
 在开关电源启动之前，整个芯片有约20uA的静态电流消耗，也就是从Cs电容拉走20uA电流  
 当VHM接近14.5v时，流入Cs电容的电流为  $(V_{IN} - 14.5) / R_s$

系统启动的必要条件是：

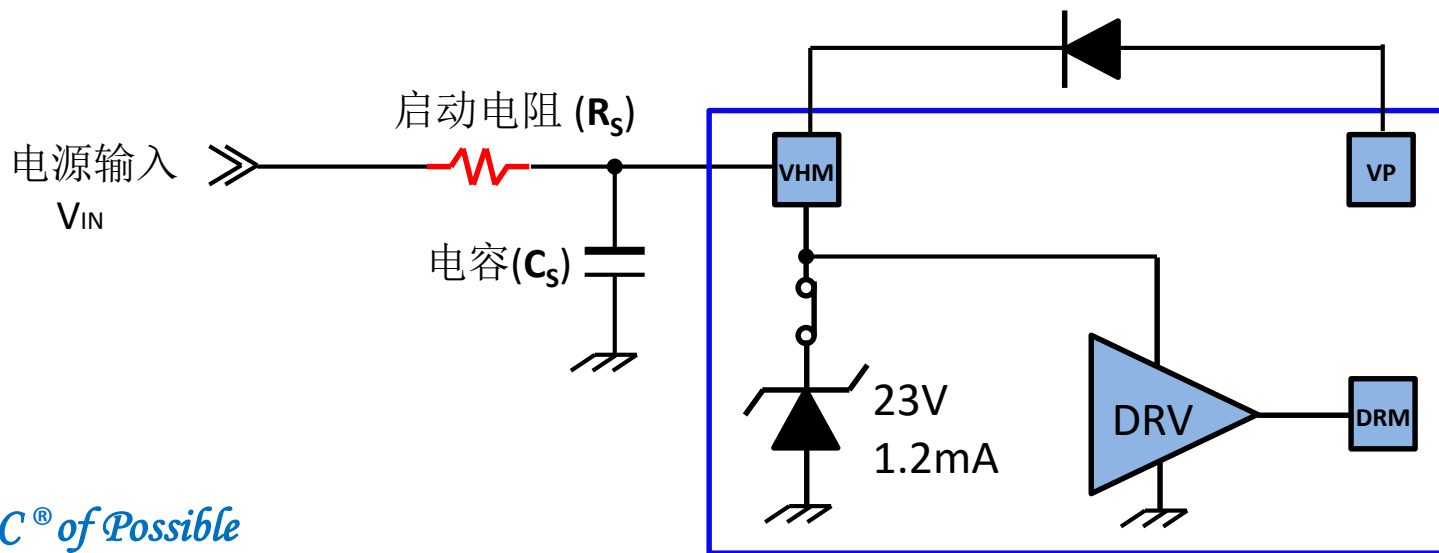
当VHM=14.5v时，流入Cs电容的净电流大于零

也就是：

$$(V_{IN,MIN} - 14.5) / R_s - 20\mu A > 0$$

这个条件设定了启动电阻的上限： $R_{S,MAX} < (V_{IN,MIN} - 14.5) / 20\mu A$

我们同时也看到，Rs 越小，流入Cs电容的电流越大，净电流越大，系统能越快启动



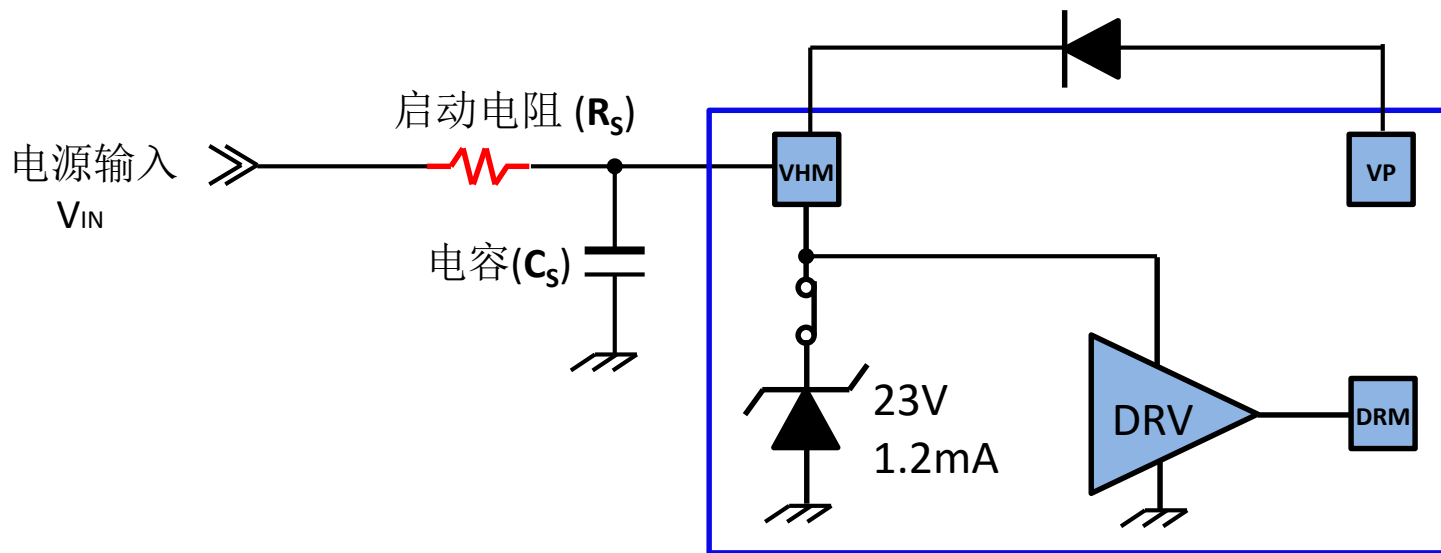
芯片内置的钳位电路只有**1.2mA**的拉电流能力。启动电阻要选得足够大使得流过电阻的电流小于**1.2mA**。

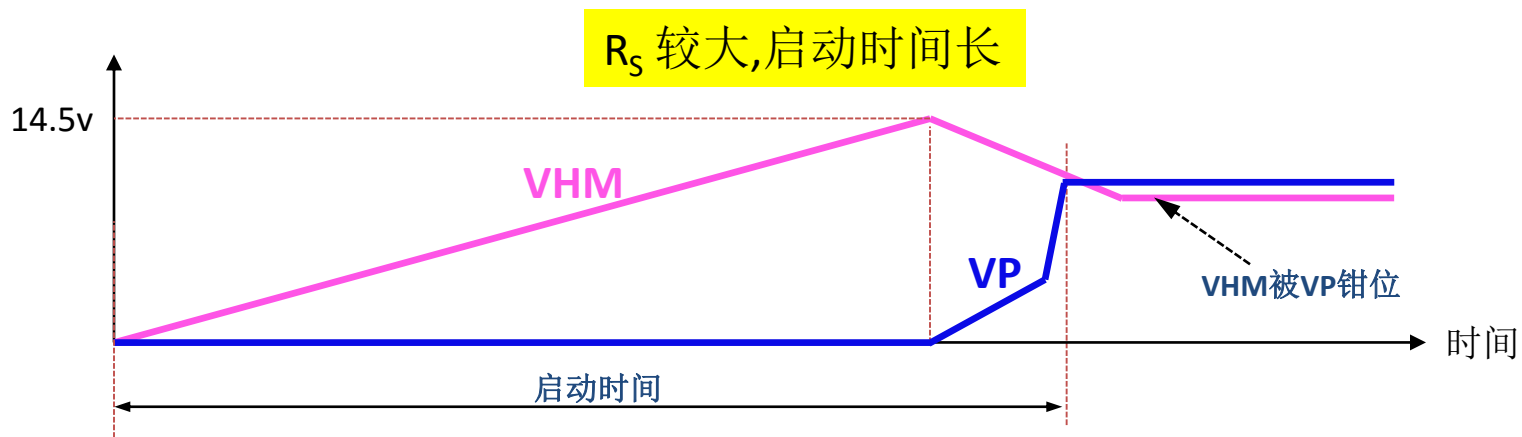
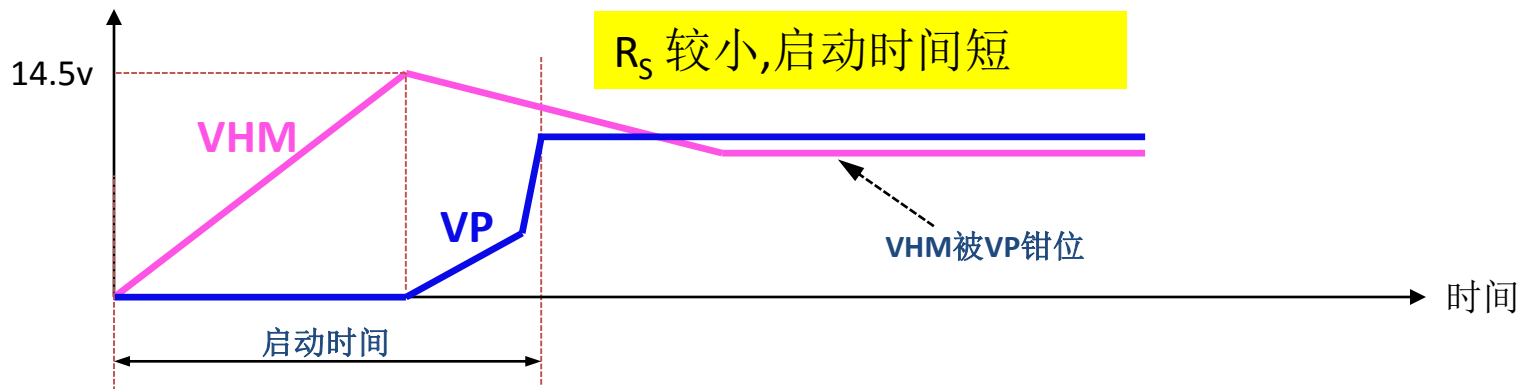
也就是：

$$(V_{IN,MAX} - 23)/R_{S,MIN} < 1.2mA$$

这个条件设定了启动电阻的下限：

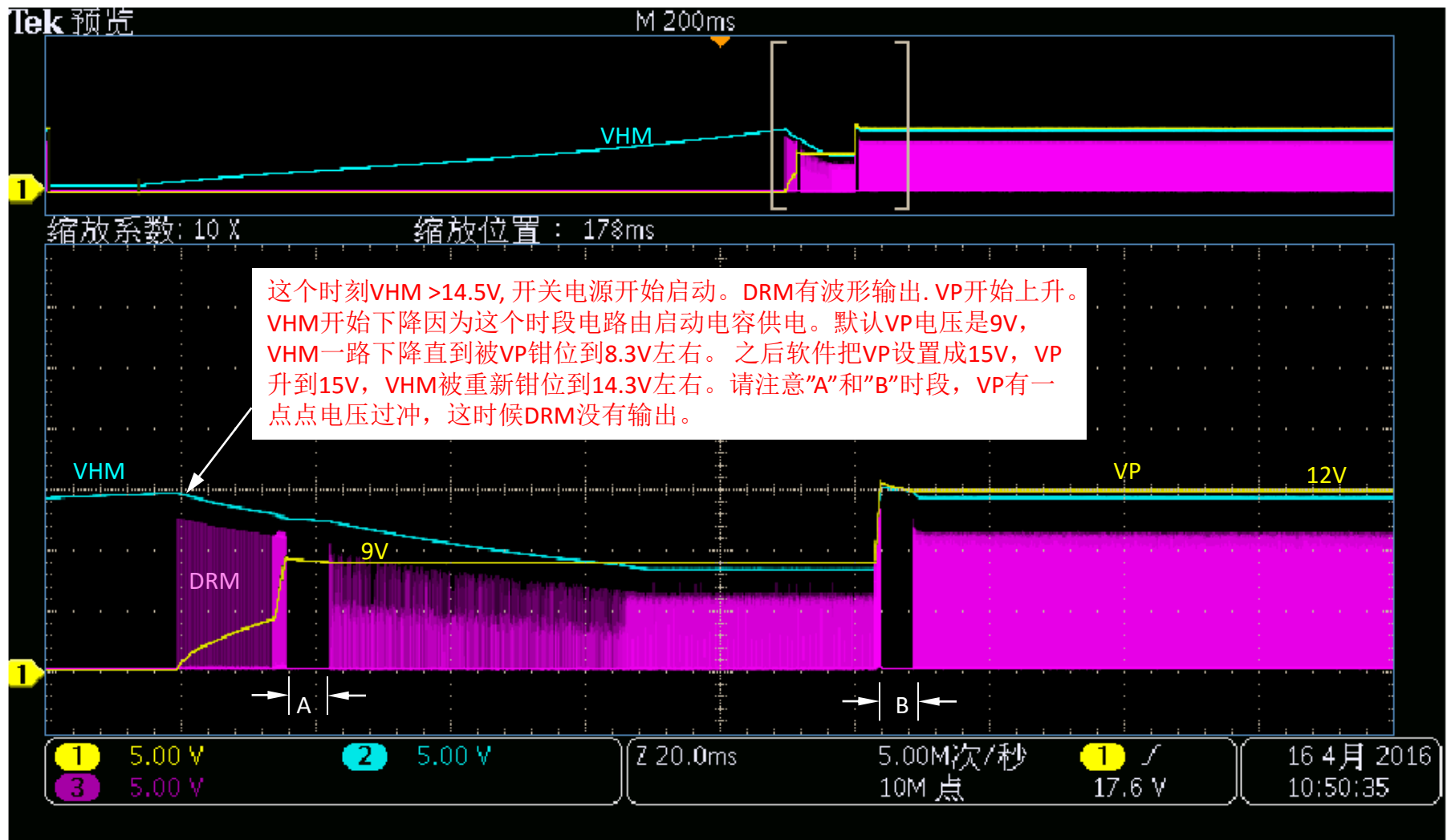
$$R_{S,MIN} > (V_{IN,MAX} - 23)/1.2mA$$





The PAC® of Possible

PAC5250启动时的各路信号观察



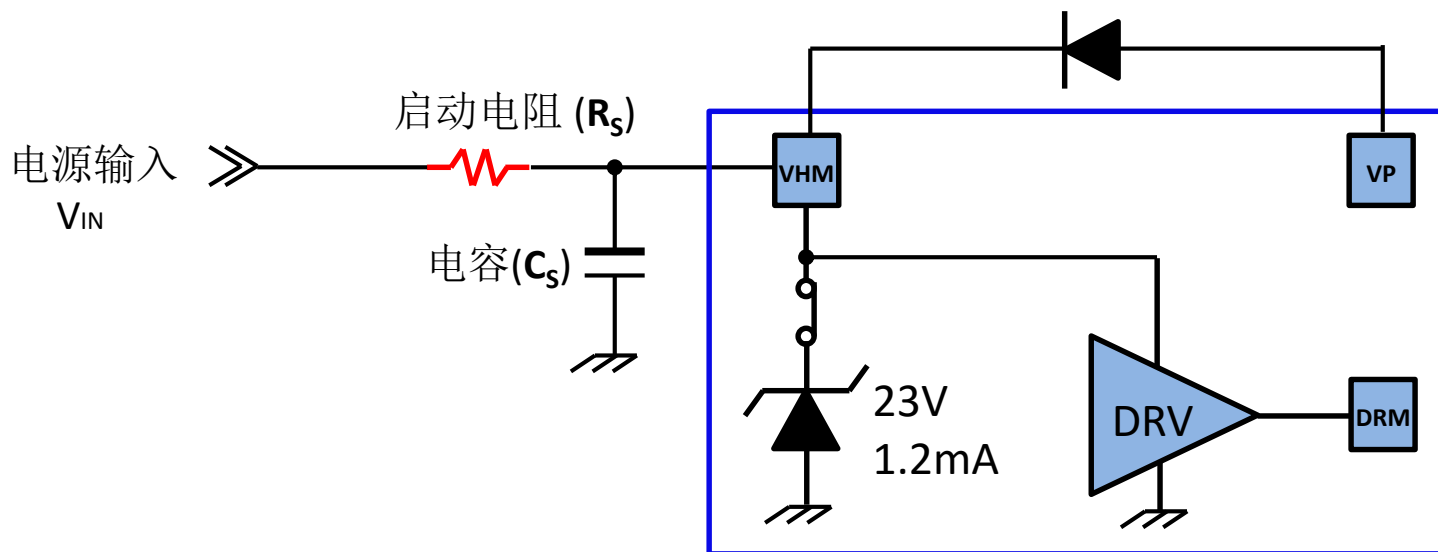
当用户使用交流电作为电源时, 整流后有几百伏的电压。在这种情况下在电源与VHM之间需要加一个阻值较大的启动电阻, 该电阻配合芯片内部的钳位电路使得VHM不超过23V以保护芯片。PAC芯片有深度睡眠功能, 进入深度睡眠时, 芯片除钳位电路之外只需20uA的电流, 也就是说如果钳位电路被关闭, 总耗电只有20uA。然而交流电或高压应用中, 钳位电路必须永远被打开, 所以大部分电流从钳位电路流走。流过VHM的总电流是:

$$I(VHM) = (V_{IN} - 23)/R_S$$

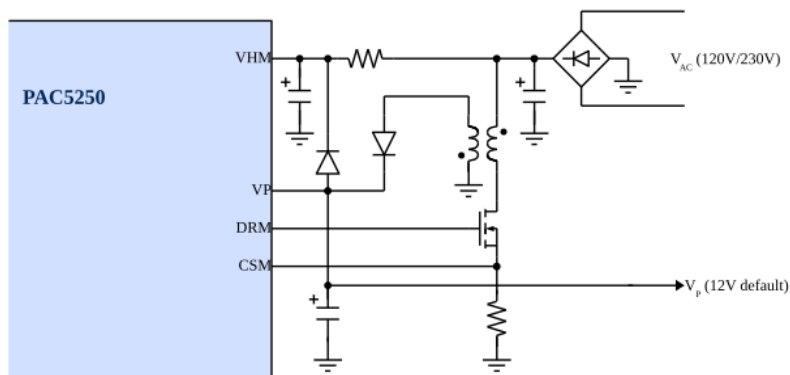
总体待机功耗是:

$$\text{待机功耗} = V_{IN} * I(VHM) = V_{IN} * (V_{IN} - 23)/R_S$$

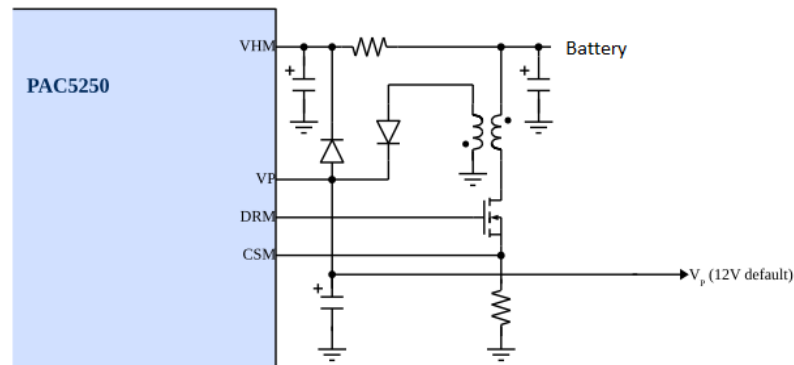
可以看到, 启动电阻( $R_S$ )越大, 待机功耗越小。但若启动电阻太大, 会增加启动时间, 并且芯片可能无法正常启动。用户要掌握好启动电阻的选择在保证芯片满足启动同时要求的同时达到最小待机功耗。



PAC525X系列产品的规格书详细介绍如何配置电源。  
下图所示的反激式主要应用在交流电源或高压电池电源。

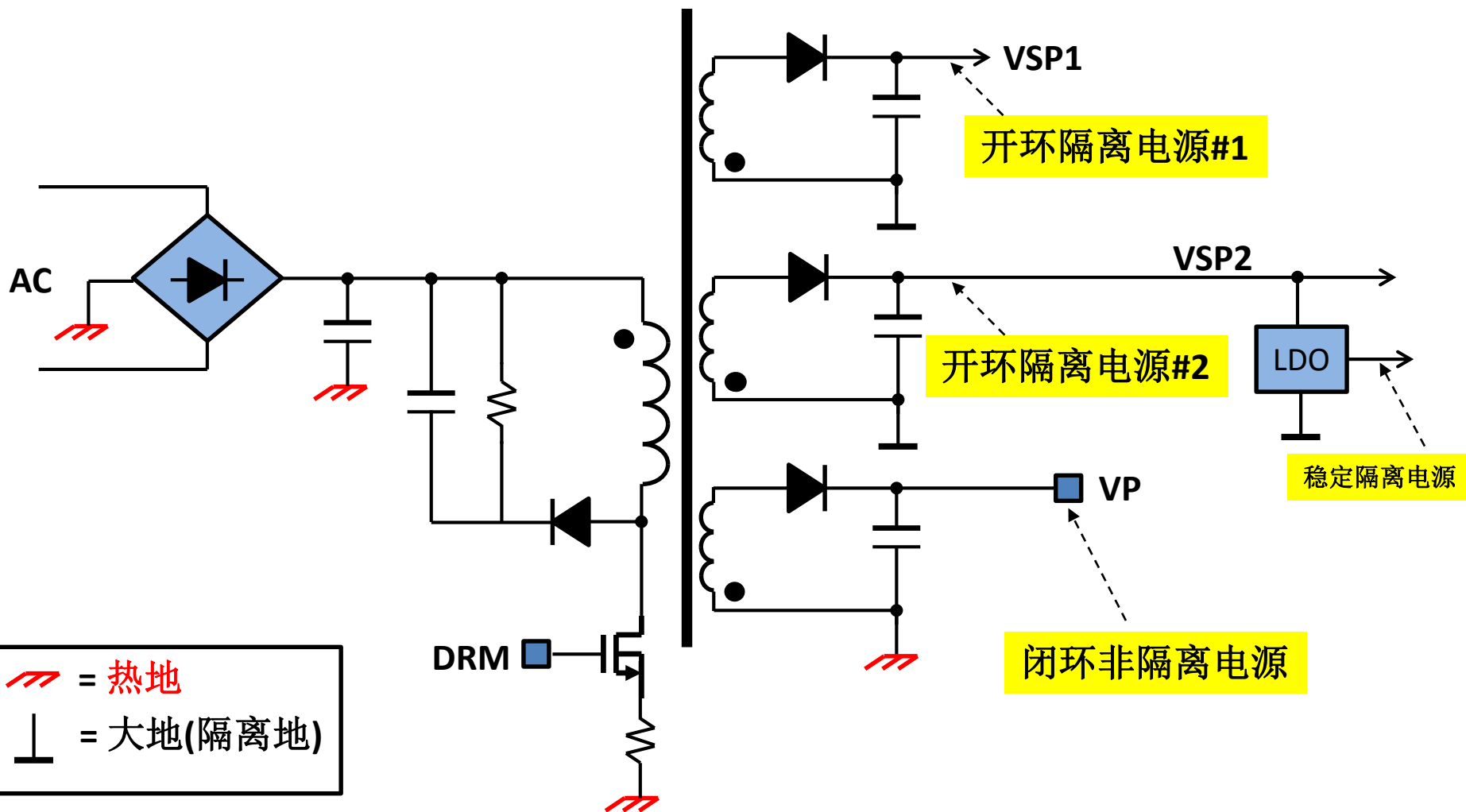


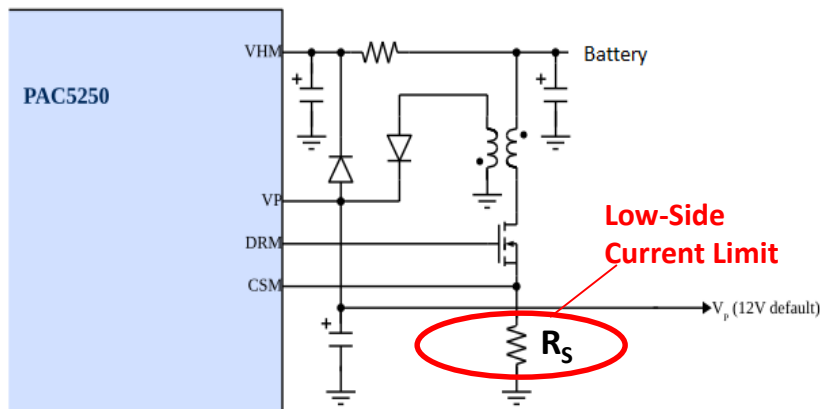
交流电源下的反激式



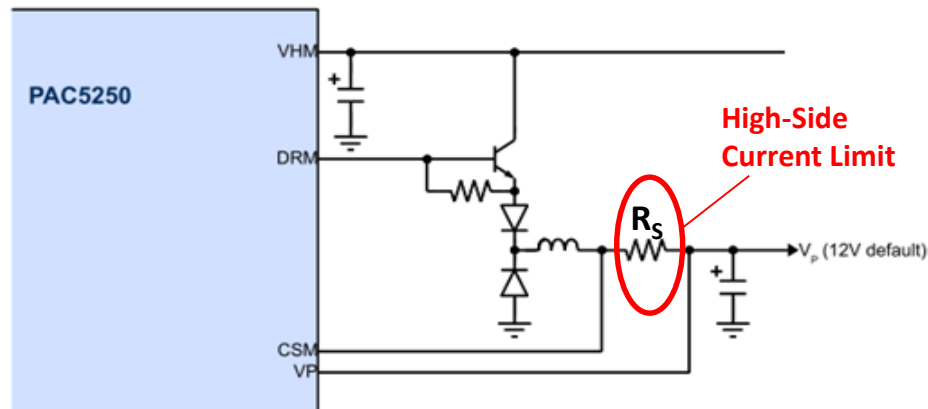
高压电源下的反激式







反激式开关电源



降压开关电源

开关电源利用限流电阻来做过流保护。当限流电阻两端压差超过阈值时，功率管被强制关断直到下一个PWM周期开始。PAC芯片根据开关电源的结构，有二种情况：

- (1) Low-Side Current Limit: 阈值 = 1v (Typical)
- (2) High-Side Current Limit: 阈值 = 0.26v (Typical)

限流电阻的值由下面表达式得到：

$$R_S = \text{阈值} / I_{LIM}$$

$I_{LIM}$  由用户或具体应用决定。以降压开关电源为例，通常情况下，VP的负载( $I_{LOAD}$ )较小，开关电源工作在非连续导通模式下，也就是说每个周期结束电感电流回零。在临界导通状态下， $I_{PEAK} = 2 \times I_{LOAD}$

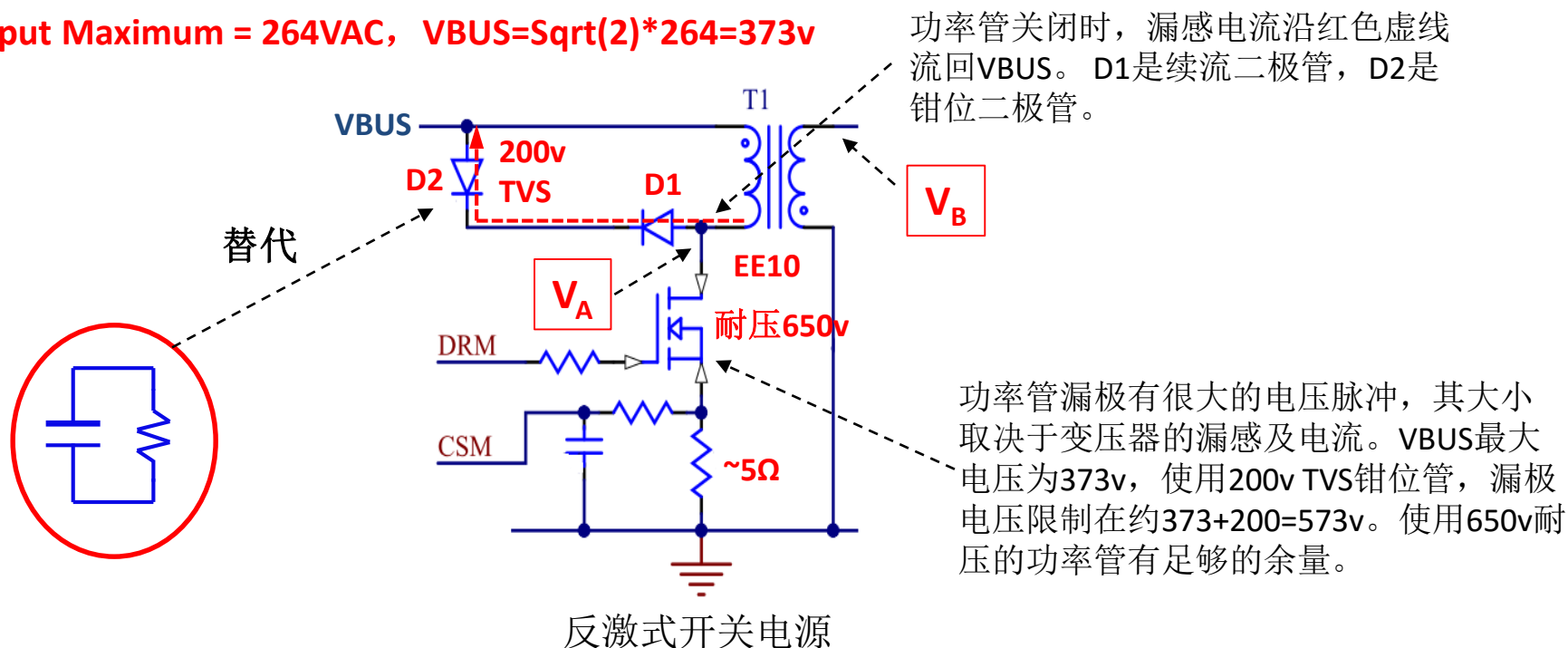
假定  $I_{LOAD,MAX} = 200\text{mA}$ ， $I_{PEAK} = 400\text{mA}$ 。加50%余量，限制电流  $I_{LIM} = (1+50\%) \times I_{PEAK} = 700\text{mA}$ 。

电感的饱和电流的选择应大于700mA。  $R_S = 0.26\text{v} / 700\text{mA} = 0.37\Omega$

*The PAC<sup>®</sup> of Possible*

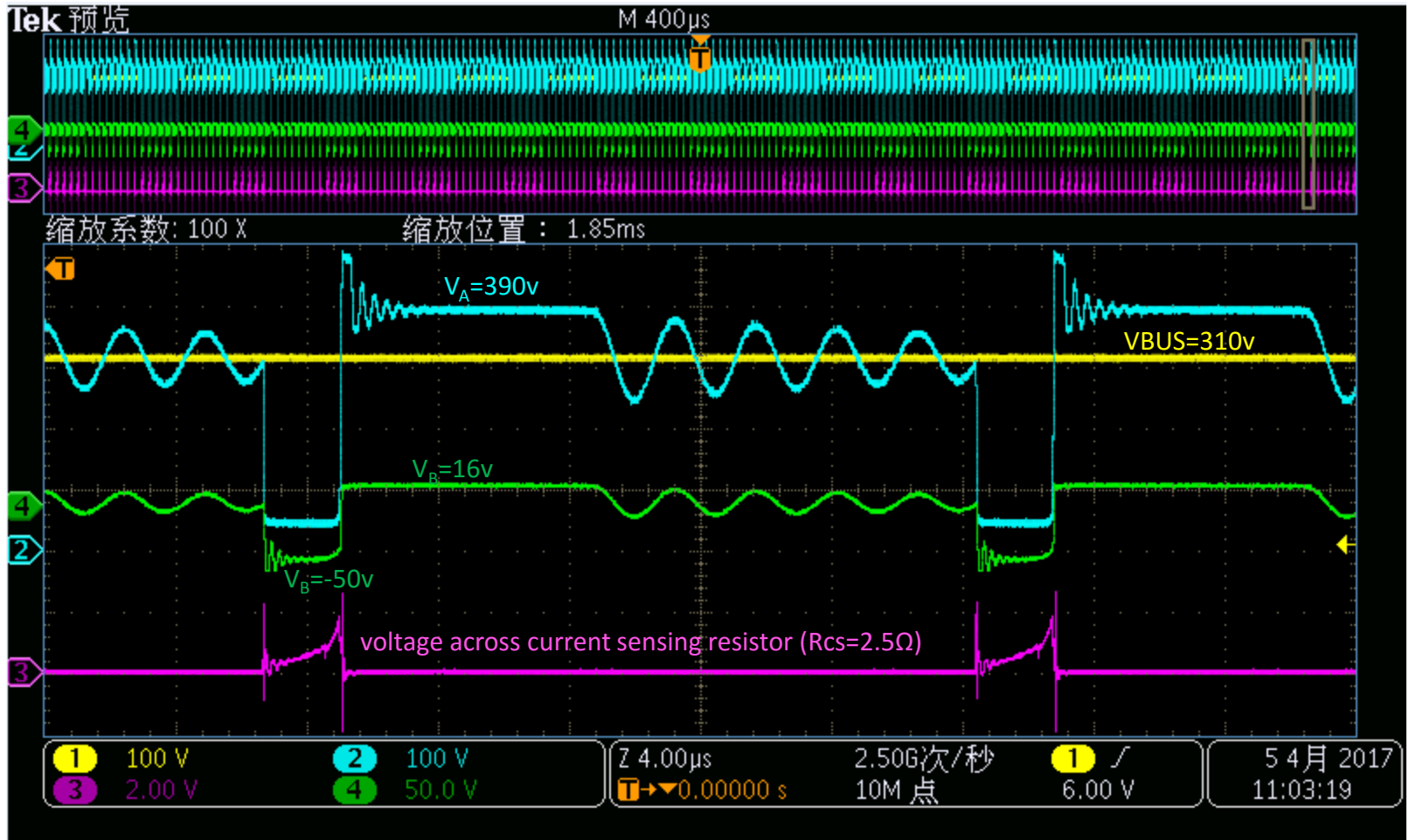
# 反激式应用元件的选择

**AC Input Maximum = 264VAC,  $V_{BUS} = \sqrt{2} * 264 = 373v$**



以反激式开关电源为例，通常情况下，VP的负载( $I_{LOAD}$ )较小。考虑到变压器的匝比效应，流过初级的电流比次级更小，所以我们能够采用比较大的限流电阻，通常约5Ω。功率管关闭时，漏感电流引起漏极很大的电压脉冲，可选择钳位二极管或并联电阻电容来限制漏极电压。所选功率管的耐压要比漏极电压脉冲大，并有一定的余量。

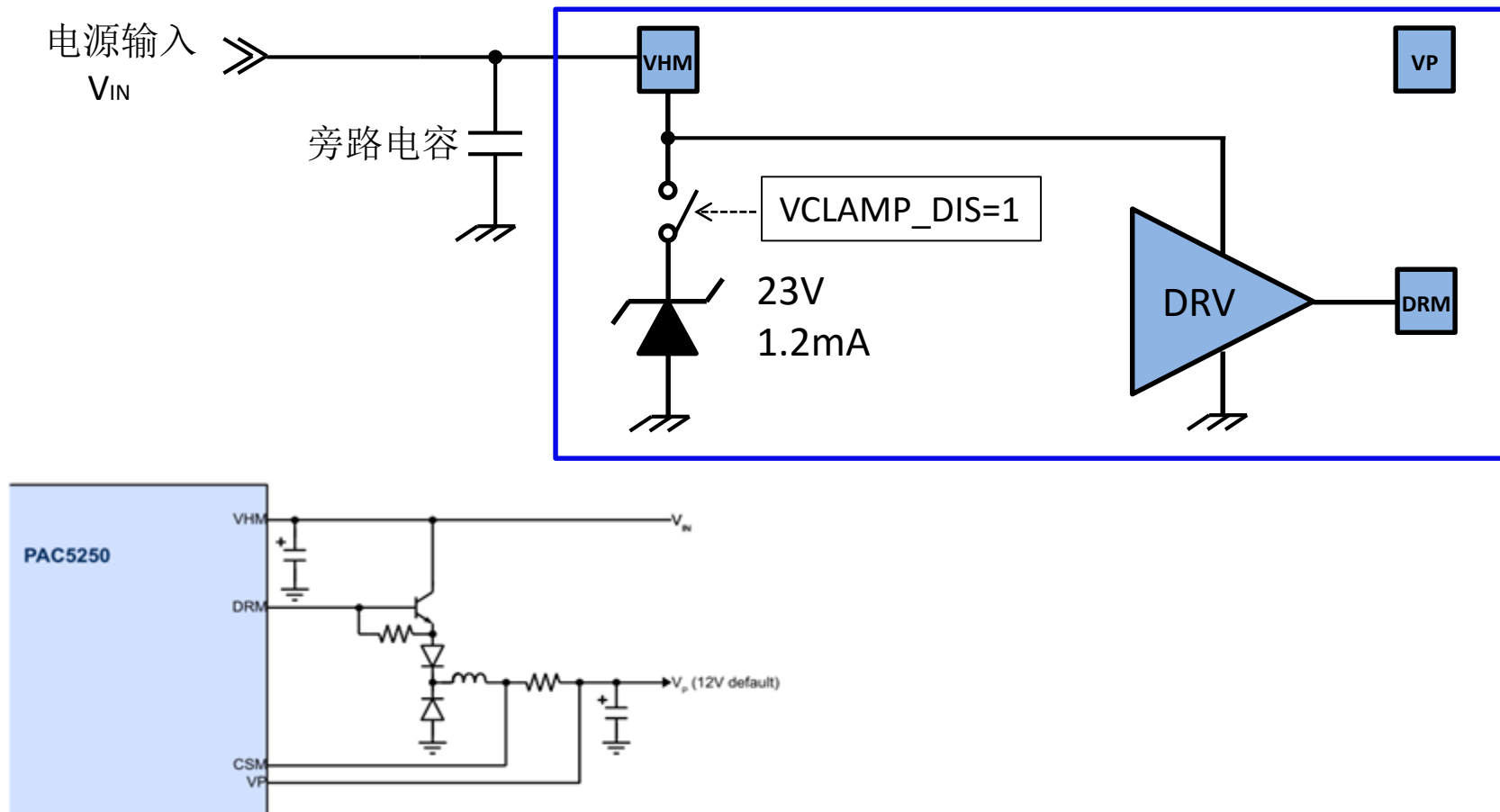
NEW



$$V_B = -(VBUS/N) = -(310/6) = -50\text{v}, \text{ where } N \text{ is transformer coil ratio}$$

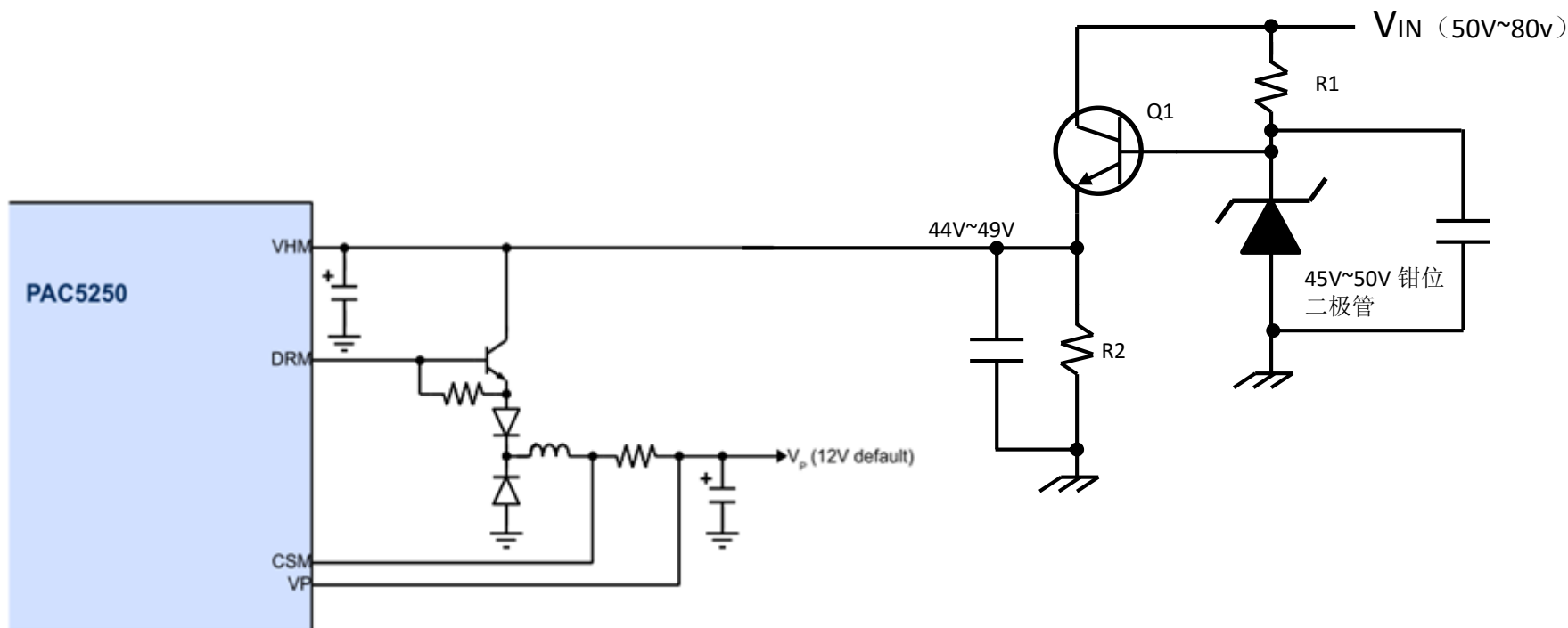
$$V_A = VBUS + N \cdot [V_P + V(\text{diode})] = 310 + 6 \cdot (12 + 1) = 388\text{v}$$

当电源电压大于18V且小于VHM最大耐压时,可以把电源直接联到VHM。PAC525X系列VHM最大耐压是**50V**, PAC5223是**70V**。系统起动后默认状态下钳位电路处于工作状态,可能浪费掉1.2mA电流。需要软件把 VCLAMP\_DIS设成逻辑“1”来去除钳位功能。**芯片处于深度睡眠模式时耗电约20uA。**



中压电源下的降压开关电源示意图

当电源电压略大于VHM最大耐压时，例如对于PAC5250芯片，电源电压介于50V和80V。这时可以采用如下图所示的折中方案，在牺牲掉一点效率的情况下，继续采用中压开关电源模式，以达到最高性价比。R1与钳位二极管配合在Q1管的基极产生较恒定钳位电压，Q1发射极电压小于基极1V左右，是一个较恒定的电压并具有很小的输出电阻。加电阻R2的目的是给Q1一定的偏置电流使它一直处于道通状态。根据Q1集电极与发射极压差的不同以及系统总电流的大小，三极管（Q1）可能会有较大的功耗。

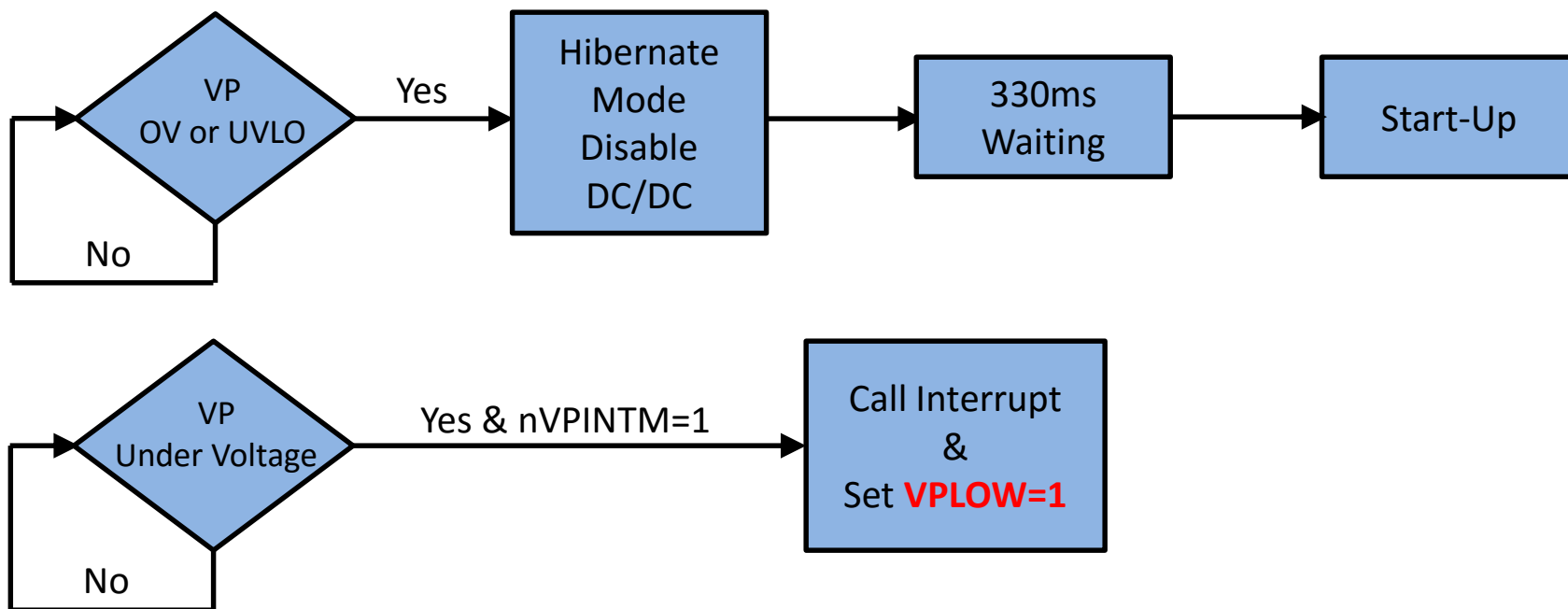


## 短接VHM和VP



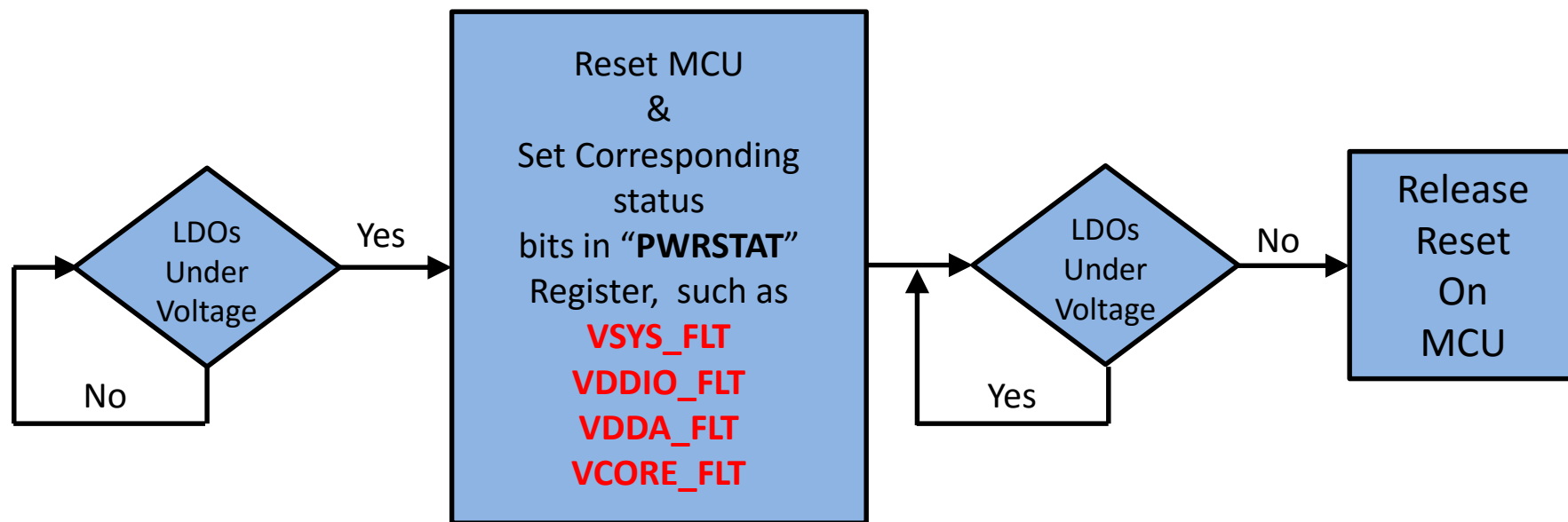
**CSM=NC**

- ❑ VP有过压保护功能，通常情况下当VP电压大于设定值36%时触发过压保护。即使VP由外部电源提供，芯片仍然保持过压保护功能。
- ❑ VP有欠压锁定的保护功能
- ❑ 过压：OV = 36%
- ❑ 欠压锁定：VP\_UVLO=4.3v (Typ)
- ❑ 欠压：VP\_PG= 92% rising / 85% falling (Typ)

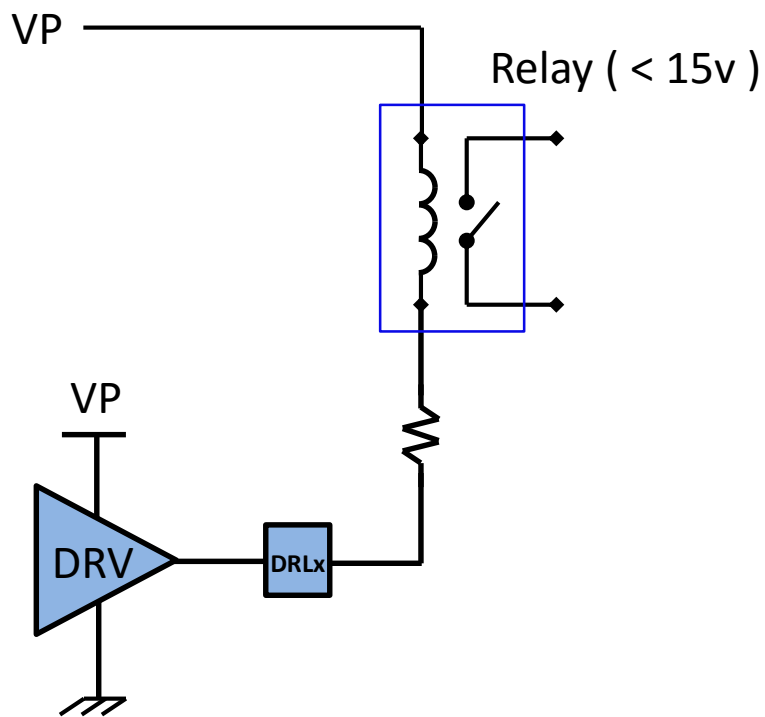




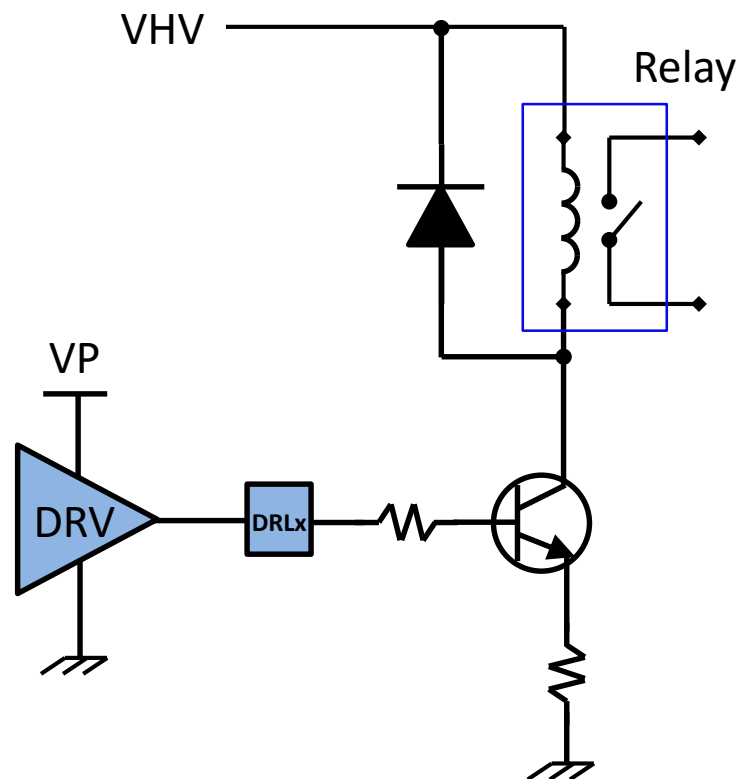
- ❑ V<sub>SY</sub>S欠压: 4v rising / 3.8v falling (Typ),
- ❑ V<sub>CCIO</sub>欠压: 82% rising / 72% falling (Typ)
- ❑ V<sub>CC33</sub>欠压: 78% rising / 68% falling (Typ)
- ❑ V<sub>CORE</sub>欠压: 66% rising / 56% falling (Typ)



要注意的是当马达从高速转动状态突然快速减速时，马达成了发电机，会把低压电源电位抬高。当抬高的电压高于VP绝对最高耐压(20V)时，有可能损坏芯片。当电源为电池时，这个现象基本不发生因为电池本身是个巨大的电容。但在PCB板组装完成后做测试时，通常用只有灌(source)电流能力而没有拉(sink)电流能力的普通电源，并且PCB板与电源之间有较长电源线。为节约测试时间，马达短时间内被加速到最高值，然后短时间内被减速到零。VP被抬高大于其最高耐压从而损坏芯片，这是测试过程中要注意的。

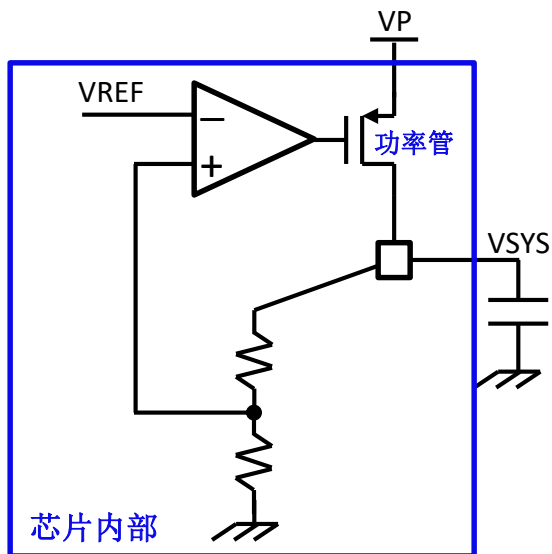


低压继电器 控制



高压继电器 控制

# 为何加REGO 电阻？ 如何选择它的值？



左上图所示的是通常用的线性稳压电源。若输出口VSYS上的负载电流是 $I_L$ ，则损耗在芯片内部的热量 $= (VP - VSYS) * I_L$ 。通常情况下， $VP=15v$ ， $VSYS=5v$ ， $I_L \approx 40mA$ ，热量损耗是400mW。若芯片外部没有很好的散热装置，400mW在芯片内部可产生相当可观的温升。为了解决这个问题，如右下图所示，在功率管的漏极与输出口VSYS之间加一个电阻，这里叫它 $R_{REGO}$ 电阻。有了这一电阻，部分的热量损耗被转移至 $R_{REGO}$ 电阻上，从而减少芯片温升。 $R_{REGO}$ 电阻上的热量损耗是 $R_{REGO} * I_L^2$ ，芯片的热量损耗减小至： $(VP - VSYS - R_{REGO} * I_L) * I_L$ 。对于同样的负载电流， $R_{REGO}$ 电阻越大，越多的热量损耗在 $R_{REGO}$ 电阻上，芯片温升越小。但 $R_{REGO}$ 电阻不能选的过大，它有上限值。一般情况下，为了保证线性稳压电源有足够的开环增益，功率管必须处于饱和区，或者说功率管源极与漏极之间之少有约400mV的压差，于是有下列方程：

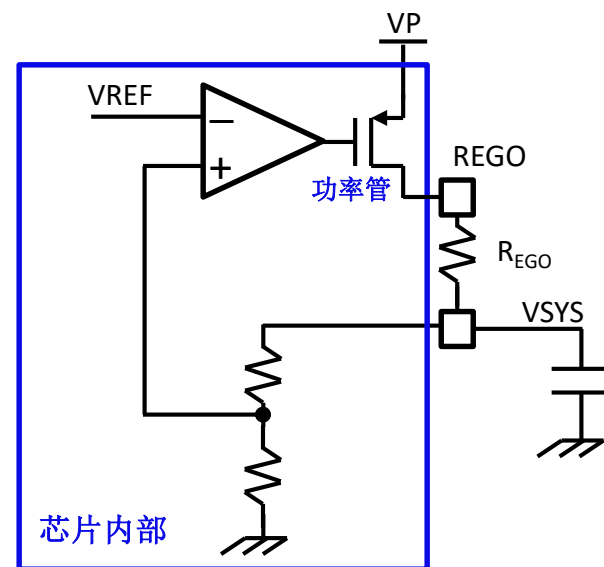
$$VP = 0.4 + R_{REGO} * I_L(max) + VSYS$$

或者，

$$R_{REGO} = (VP - VSYS - 0.4) / I_L(max) \quad , \quad \text{其中 } I_L(max) \text{ 是最大负载电流}$$

举例来说， $VP=15v$ ， $VSYS=5v$ ， $I_L(max) = 40mA$ ，可得  $R_{REGO} = 240 \Omega$ 。

通常情况下选 $R_{REGO}$  小于手算值以保有余量。



## 栅极驱动端口，功率管易被损坏，谁是元凶？

Answer:

- (1) 过流，加过流保护
- (2) 过热，选适当功率管 + 散热装置
- (3) 过压，Why?

## 马达转动时正/负脉冲是如何产生的？

## 布线时的注意事项

许多新手在马达控制板的研发过程中遇到许多麻烦. 线路板改了好几版还是有许多这样那样的问题. 一个常见的现象是功率管或控制芯片被损坏, 原因之一有可能是太强的负电压脉冲导致芯片损坏. 为了更好地理解负电压产生机制, 我们首先回顾一下物理或电学基本定律:

## 流过电感的电流不可能发生突变

定义:  $L$  电感的感量,  $V$  是电感两端的压差,  $I$  是流过电感的电流

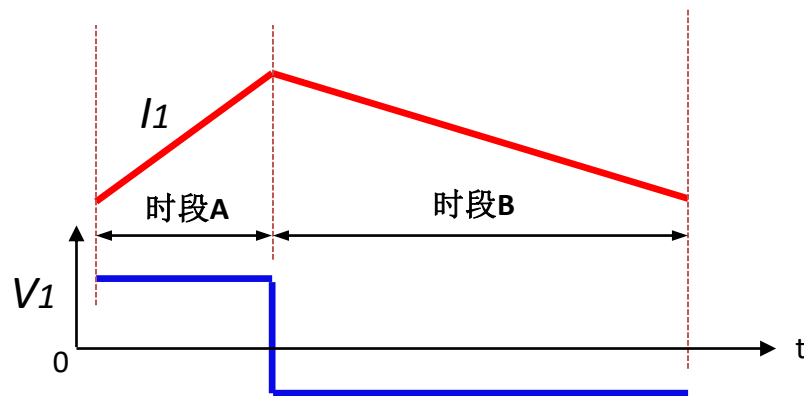
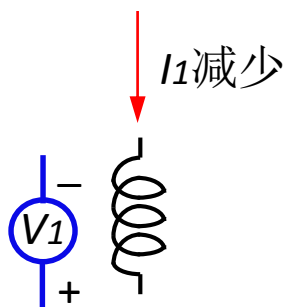
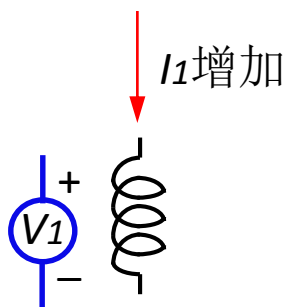
$$V = L \cdot dI/dt$$

如果流过电感的电流有突变,  $dI/dt = \infty$ , 导致  $V = \infty$ , 就是说电感两端的压差无穷大, 显然这是不可能的

还有一点要牢记的是因电流变化而引起的电感电压总是会阻碍电感电流的变化, 如下图所示

**时段A:**  $I_1$  电流增加, 产生正的  $V_1$  电压,  $V_1$  阻碍电感电流增加

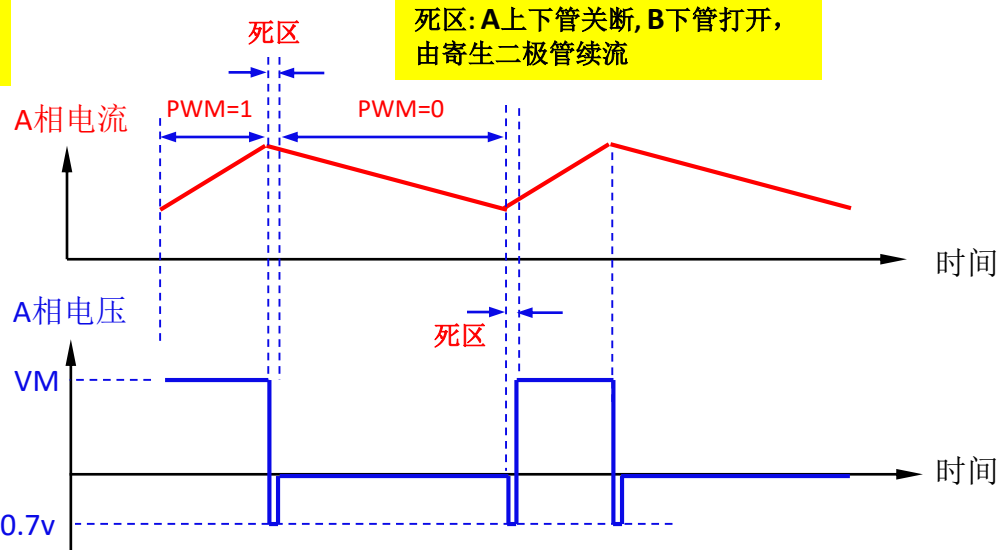
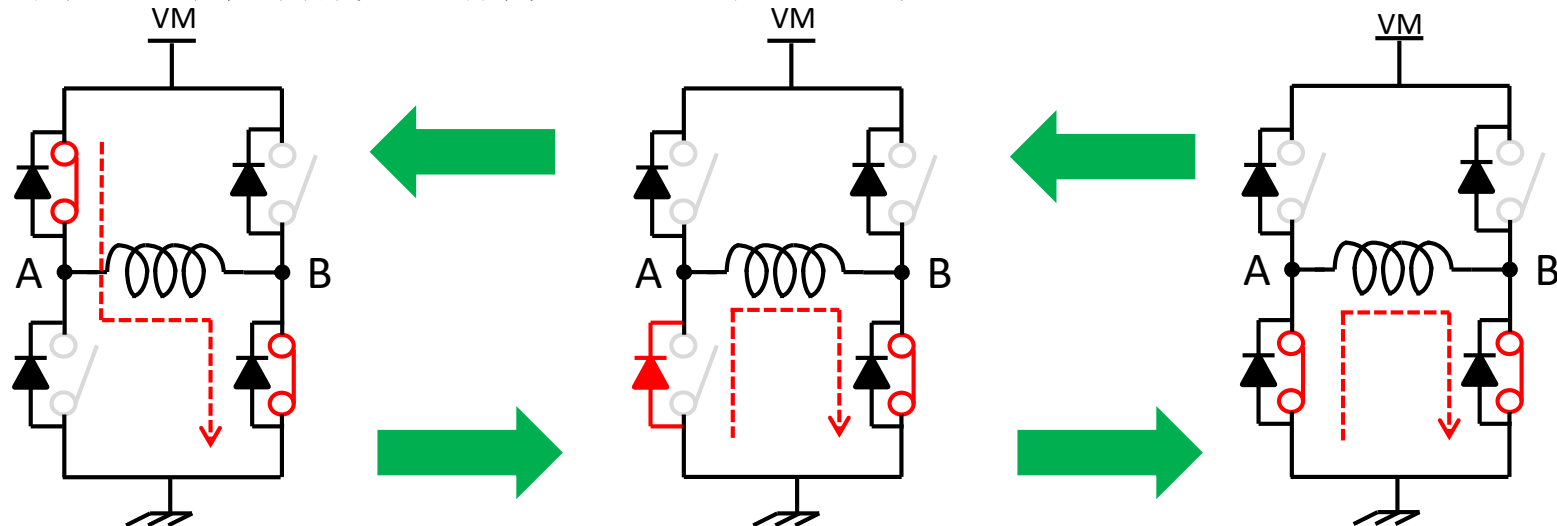
**时段B:**  $I_1$  电流减少, 产生负的  $V_1$  电压,  $V_1$  阻碍电感电流减少



# 马达转动时负脉冲是如何产生的？

用双相马达解释工作原理：

这里用“理想开关”加“二极管”来替代“NMOS功率管”，没有加入任何寄生参数，电流从相A到相B。



没有寄生参数时相A  
电压只在负0.7v左右

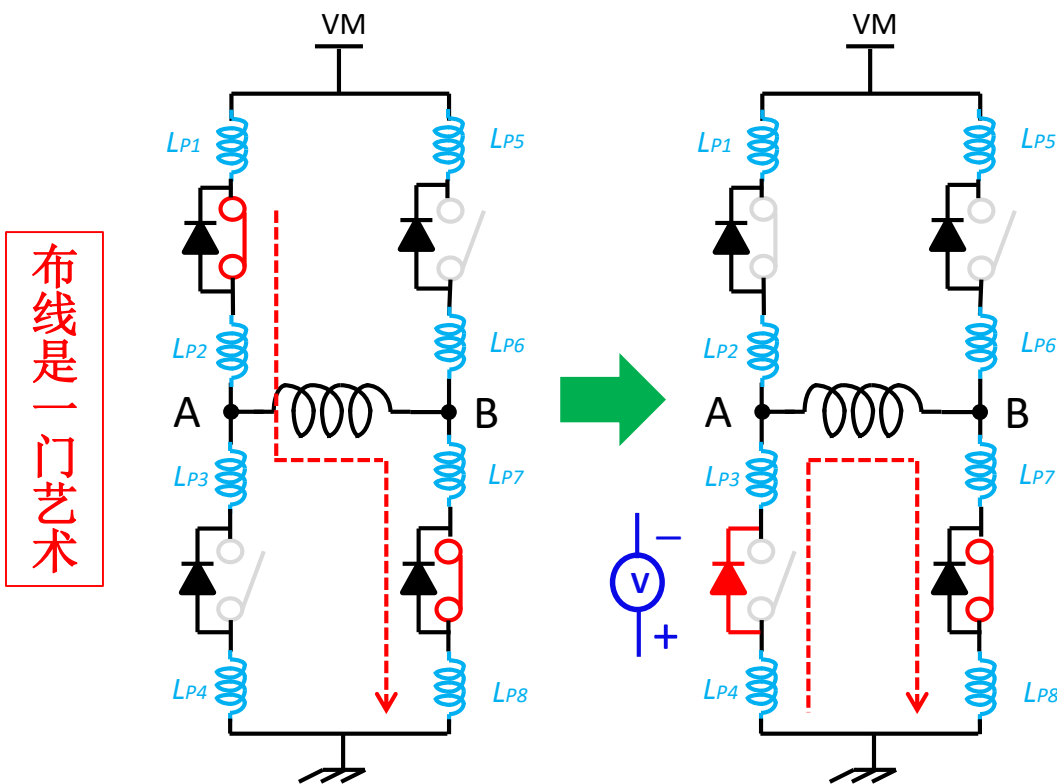
*The PAC® of Possible*



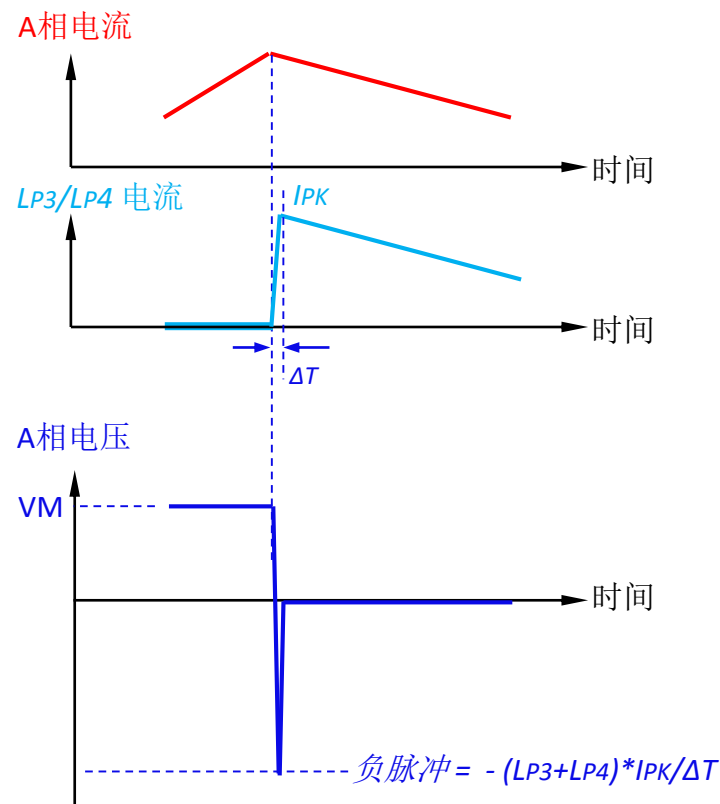
# 马达转动时负脉冲是如何产生的？

用双相马达解释工作原理: 这里用“理想开关”加“二极管”来替代“NMOS功率管”，但加入线路板布线所产生的寄生电感。

如下图所示，从功率管到电源正极，负极，以及马达相位均有寄生电感，而寄生电感是产生巨大负脉冲的元凶。当马达从PWM=1进入死区时，寄生电感 $LP3$ 和 $LP4$ 的电流在短时间( $\Delta T$ )内从0变到 $IPK$ ，从而产生负脉冲。负载越大，马达电流越大， $IPK$ 越大，产生负脉冲越大。当马达堵转时，反电动势为零，马达电流达到最高值，产生巨大负脉冲并导入芯片继而损坏芯片，这就是常见的大负载或堵转时芯片容易损坏的原因。一般来说布线越短，越粗，越厚，寄生电感值越小。所以解决方法是在线路板的设计过程中把减小如图所示的寄生电感作为最高优先级考虑。半桥上下功率管之间要尽量靠近，连线加粗加厚。功率管与电源正负极的连线也要尽量靠近并加粗加厚。



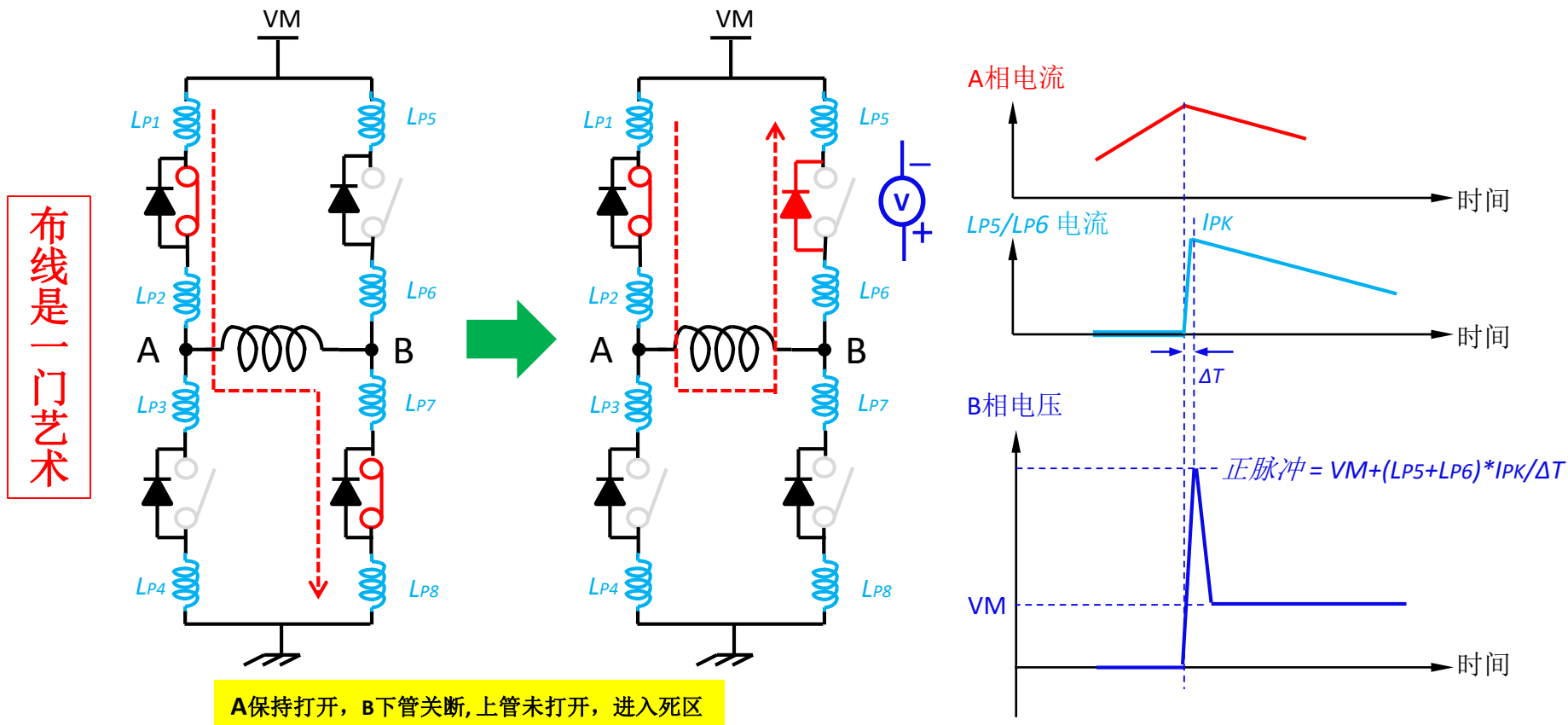
A上管关断, B下管一直打开, 死区时间段由寄生二极管续流



# 马达转动时正脉冲是如何产生的？

用双相马达解释工作原理: 这里用“理想开关”加“二极管”来替代“NMOS功率管”，但加入线路板布线所产生的寄生电感。

如下图所示，从功率管到电源正极，负极，以及马达相位均有寄生电感，而寄生电感是产生巨大正脉冲的元凶。当马达换相时，寄生电感  $LP5$  和  $LP6$  的电流在短时间 ( $\Delta T$ ) 内从 0 变到  $IPK$ ，从而产生正脉冲。负载越大，马达电流越大， $IPK$  越大，产生正脉冲越大。当马达堵转时，反电动势为零，马达电流达到最高值，产生巨大正脉冲并导入芯片继而损坏芯片，这就是常见的大负载或堵转时芯片容易损坏的原因。一般来说布线越短，越粗，越厚，寄生电感值越小。所以解决方法是在线路板的设计过程中把减小如图所示的寄生电感作为最高优先级考虑。半桥上下功率管之间要尽量靠近，连线加粗加厚。功率管与电源正负极的连线也要尽量靠近并加粗加厚。



请参考[www.active-semi.com](http://www.active-semi.com)网页提供的应用笔记，里面有PCB布线规则

## 3. 硬件指南

- AN1201: 关于开关电源设计 
- AN1202: PCB布线规则 - PCB Layout Rules v1.1 
- AN1203: VP直接供电线路 
- AN1204: 功率管开关导致的电压电流振荡整改 
- AN1205: PAC 525x高压控制板测试流程 

## 4. IDE和编码指导

- AN1301: IDE (Coocox) 开发环境安装与项目创建 
- AN1302: Coocox使用Jlink仿真配置 
- AN1303: BootLoader (Coocox版) FW简要说明与应用 

## 5. 模拟PAC指南

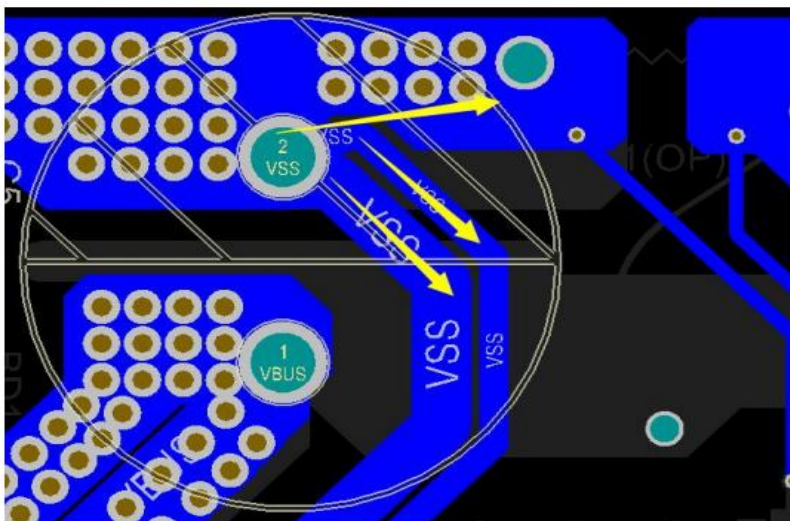
- AN1401: PAC模拟核使用手册 

## 6. 数字PAC指南

- AN1501: PAC 数字核使用手册 

# 线路板布线的一些注意事项

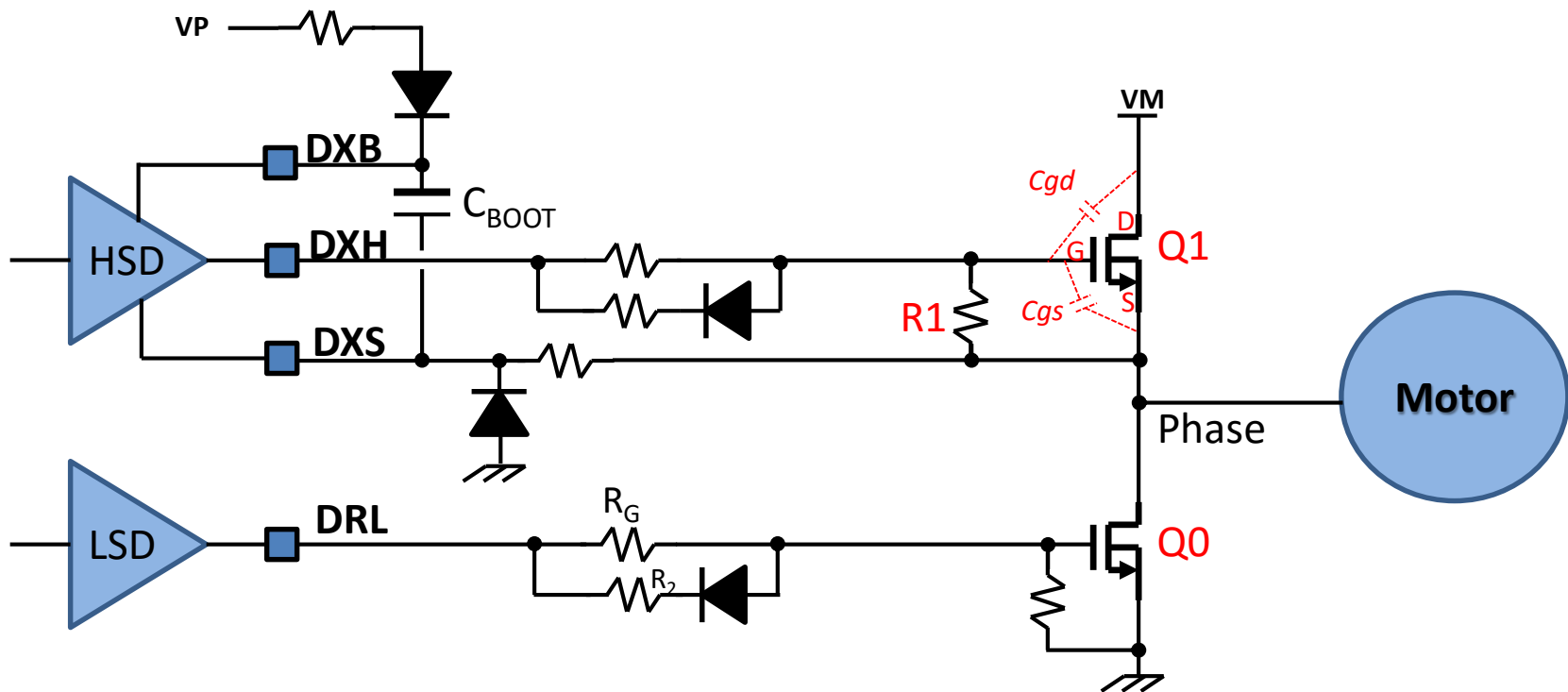
1. 针对 220VAC 输入整流后或直接输入的直流电压源，需将功率地与信号地在滤波大电容处做分离，做到单点接地。



2. 开关电源处产生开关信号的元件应远离各信号采集线。如：开关电源部分的功率管、二极管、电感等应远离 AIOx 和 PCx 等信号线。
3. VP, REGO, VSYS, VCCIO, VCC33, VCC18 等电源端口，请将对应电阻电容器件尽量靠近此引脚。
4. AIO10, AIO32, AIO54 等差分信号检测口，请走等长平行线，连接端口通过电阻到采样电阻焊盘处，滤波电阻电容尽量靠近芯片引脚。
5. QFN 封装背后的 pad 需接地，并用通孔将上下层连接，做到良好散热。
6. 上臂驱动自举线路元件需尽量靠近芯片引脚。
7. 半桥驱动门极电阻需尽量靠近功率管。
8. 各桥臂驱动线长度差异不宜过大，尽量保证等长。

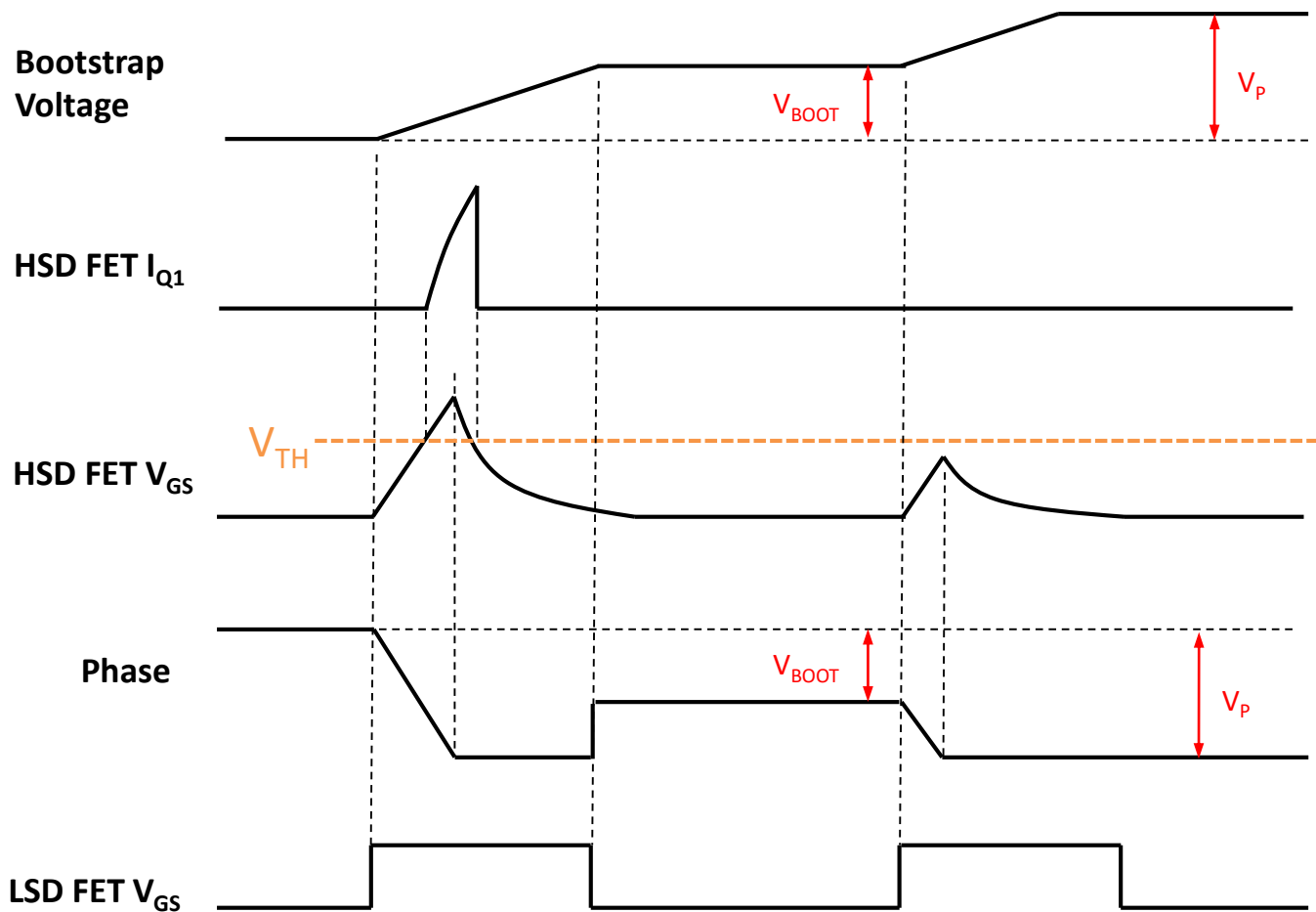
9. 高压各线间距应保证 1mm/100V 的线距标准。
10. 大功率线路需保证 1mm 线宽、1 盎司铜厚，可以流过 1A 电流。若要在不增加线宽情况下增大流过电流，请在画板时附加 solder 层，用以加锡扩流。

- ❑ 每个LDO输出引脚加一个0.01 - 0.1uF对地去耦电容，抑制MCU主频、倍频辐射传导。
- ❑ 上下管驱动输出端加足够大的栅极电阻 $R_G$ 以减少辐射。
- ❑ 表贴元件比通孔元件有较少的辐射性。
- ❑ 减少环路的面积。
- ❑ 有大电流流过的线要短而粗以减少压降及辐射，如电源线，地线，功率管的连接线。
- ❑ 加Ferrite-Bead，靠近辐射源。
- ❑ 加吸收电磁辐射的贴膜。

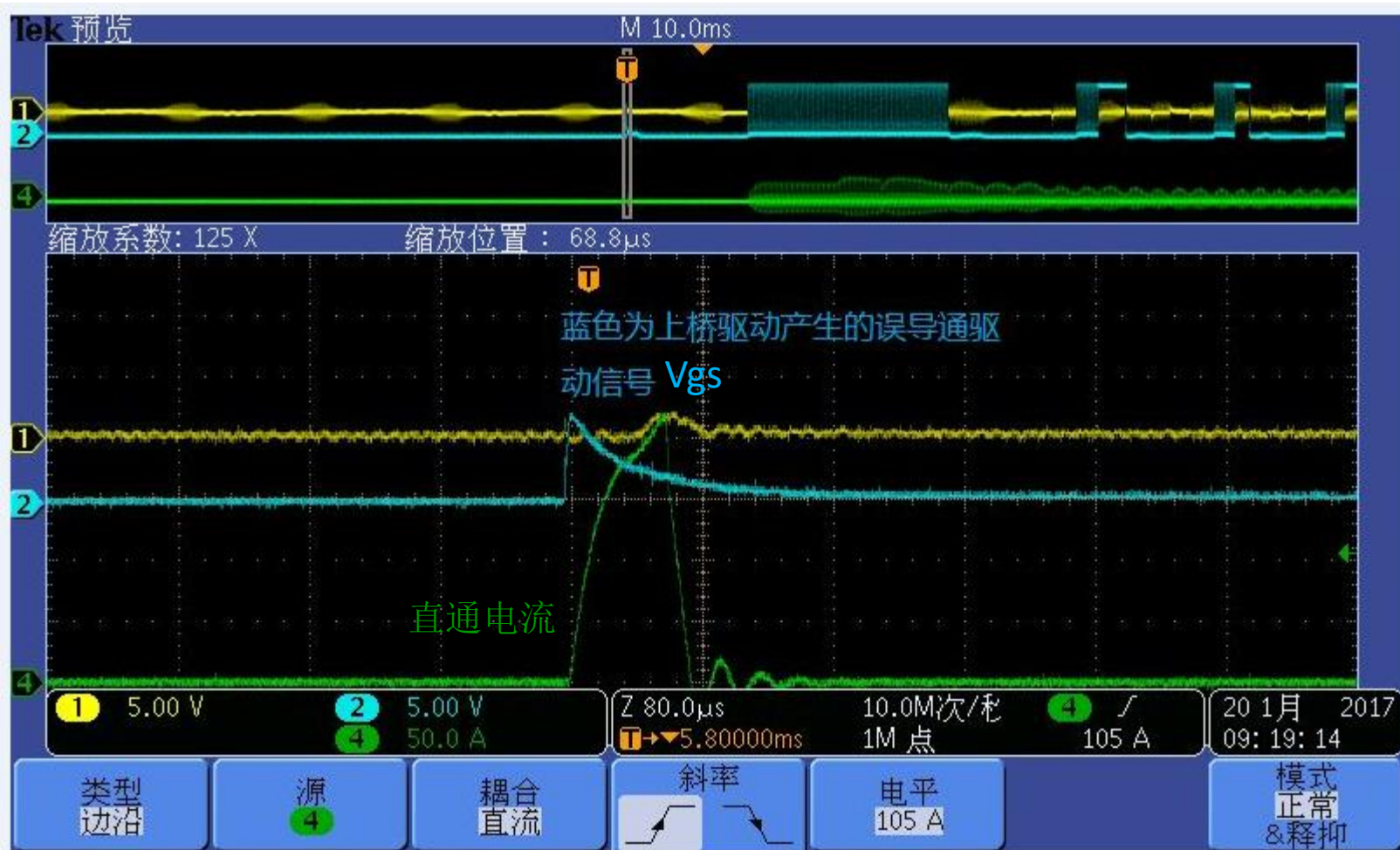


初始启动时，自举电压为零，上管栅极驱动不工作，R1短路G和S，上管关闭。因为R1(~10K)有高阻值，上管是弱关闭。当下管(Q0)初始打开对自举电容( $C_{BOOT}$ )充电时，Q1的源极(Phase)被下拉到0v，因为Q1寄生电容Cgd和Cgs的存在，有电流从VM流过Cgd,Cgs,Q0到地，Cgs被充电，在Q1的栅极和源极之间产生压差： $V_{gs}(Q1) = VM * C_{gd} / (C_{gd} + C_{gs})$

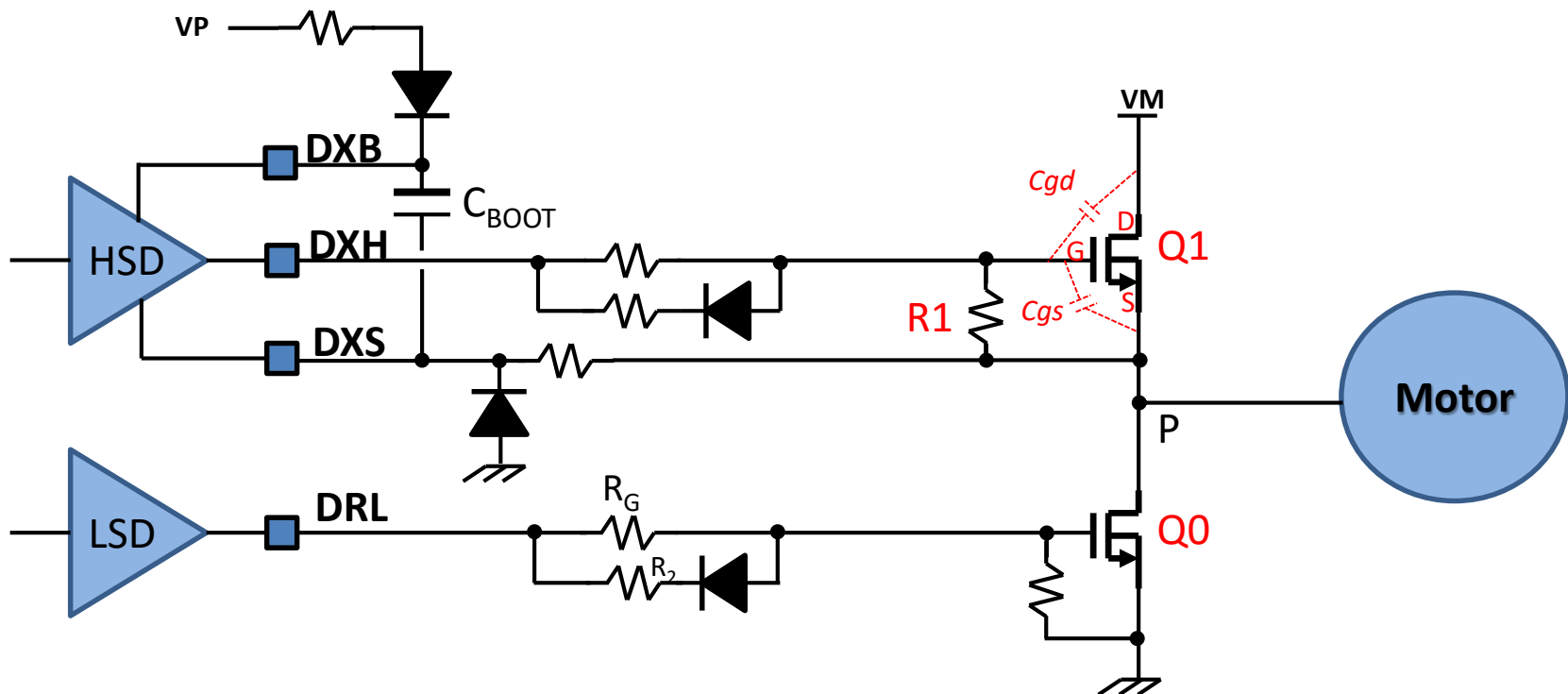
若 $V_{gs}$ 大于上管(Q1)的阈值，Q1瞬间被误打开，从而产生直通电流。之后R1对寄生电容Cgs放电， $V_{gs}$ 以指数形式降为零。当 $V_{gs}$ 降到小于Q1阈值时，直通电流消失。









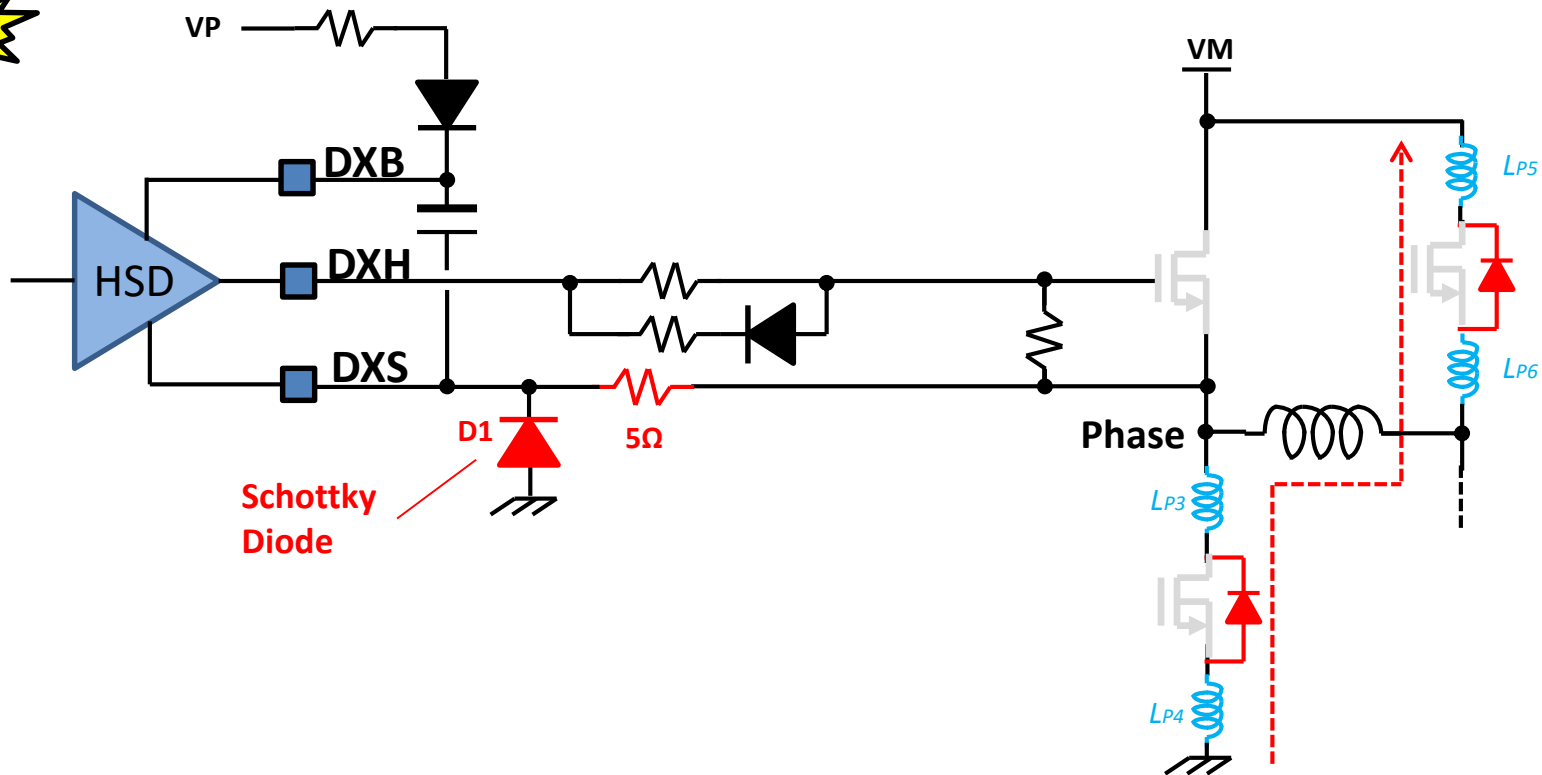


解决方案 #1:

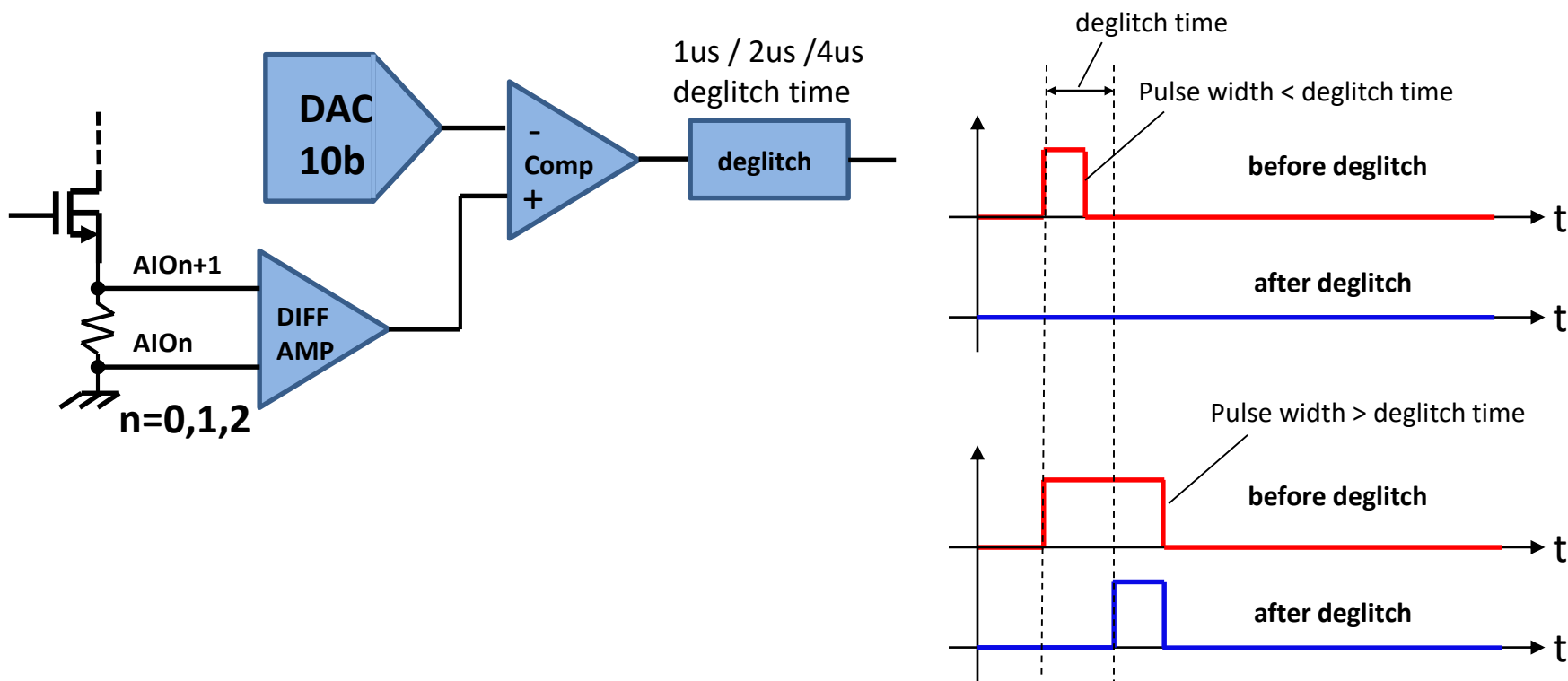
逐步减少启动阶段下管的占空比直到直通电流小到可接受的程度。

解决方案 #2:

选择阈值高的功率管，使得对上管寄生电容充电引起的V<sub>gs</sub>一直小于它的阈值。



过流保护发生时，所有上下管同时关闭。如上图所示，马达电流通过某个下管和另一个上管的体位寄生二极管续流。与该下管漏极相连的相位电压瞬间被具大的马达电流拉到负压。该负压正比于 $(L_p \cdot di/dt)$ ，也就是正比于保护电流， $L_p = L_{p3} + L_{p4}$  是功率管两端连线的寄生电感和。保护电流值设置得越大，负压越大。当保护电流大到一定程度时，所产生的负压超过规格书 (datasheet) 给定值，芯片易受损。当普通二极管不能起到保护作用时，可选用反向恢复时间短的肖特基二极管来做钳位保护。这与版图的好坏有极大的联系，关键是减小寄生电感。



为了防止过流保护误触发, 在过流比较器的输出端加一个去毛刺时延电路。时延可选1/2/4微秒。因为时延带来的保护延迟, 马达在关闭时的电流往往大于预先设定值。时延越大, 两者的差别越大。用户要选择适当的时延值使得既能防止误触发, 由能起到过流保护作用。

```
pac5xxx_tile_register_write(ADDR_PROTINTM, 0x00);           // Clear PROTINTM to enable real-time status on PROTSTAT
while (pac5xxx_tile_register_read(ADDR_PROTSTAT));          // Read PROTSTAT to wait for OC event to clear
pac5xxx_tile_register_write(ADDR_PROTINTM, 0x07);           // Set PROTINTM for OC protection
// Disable driver
pac5xxx_tile_register_write(ADDR_ENDRV, 0);
// Enable driver
pac5xxx_tile_register_write(ADDR_ENDRV, 1);
```

application dependent

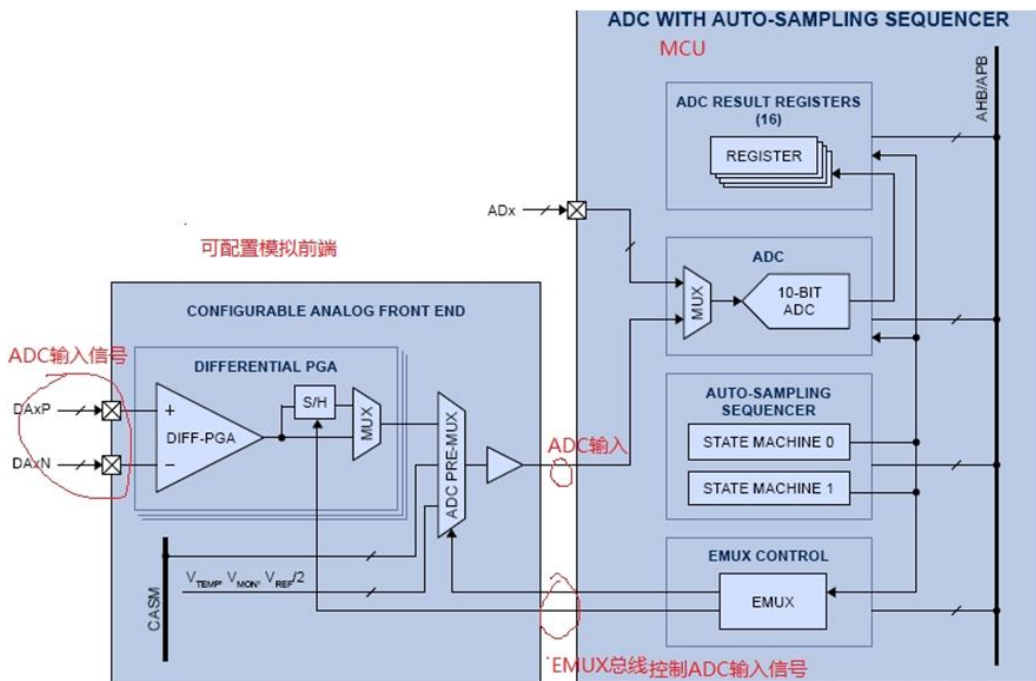
过流事件发生时，用户需要置ENDRV=0，再置ENDRV=1，从而重新使能栅极驱动。但用户往往发现要来回做许多次才能最终达到目的。这给用户带来许多疑惑，倒底如何做才是正确的。过流事件发生后，往往需要一定的时间才会消失。正确的做法是要等到过流事件完全消失后，才重新使能栅极驱动。否则的话即使栅极驱动被使能，也会被尚未消失的过流事件打断。了解这一道理，用户就不难理解上面的代码。

上述代码是处理过流事件的标准代码，请严格参照：

第一行代码设置掩码为零。只有当掩码为零时，状态寄存器`PROTSTAT`的值才会实时反应过流状况。  
第二行代码反复读状态寄存器`PROTSTAT`，直到它的值为零。  
第三行代码的写入掩码寄存器`PROTINTM`的值应按照具体应用来定。

栅极驱动使能位**ENDRV**是控制栅极驱动的最后一把锁。这把锁一打开，就会有巨大的能量输出。若打开这把锁的时机不对，可能造成硬件损坏。用户一定要遵循如下的开锁/关锁条例，才能开发出可靠的产品。这是从实践中总结的经验教训，请严格参照。

- (1) 芯片初使化时置ENDRV=0 (**推而广之，应该初使化所有的控制寄存器**)
- (2) 只有当芯片收到栅极驱动使能命令时，才置ENDRV=1，并遵循如下顺序：
  - ☐ 初使化六路PWM载波
  - ☐ 设置相应的GPIO端口为输出口
  - ☐ . . .
  - ☐ 置ENDRV=1 **← 当所有的步骤都准备完毕，最后一步才开锁**
- (3) 当芯片收到关闭栅极驱动命令时，并遵循如下顺序：
  - ☐ 置ENDRV=0 **← 第一步先把锁关闭**
  - ☐ . . .
  - ☐ 处理六路PWM载波，用户决定如何处理
  - ☐ 处理相应的GPIO端口，用户决定如何处理
  - ☐ . . .



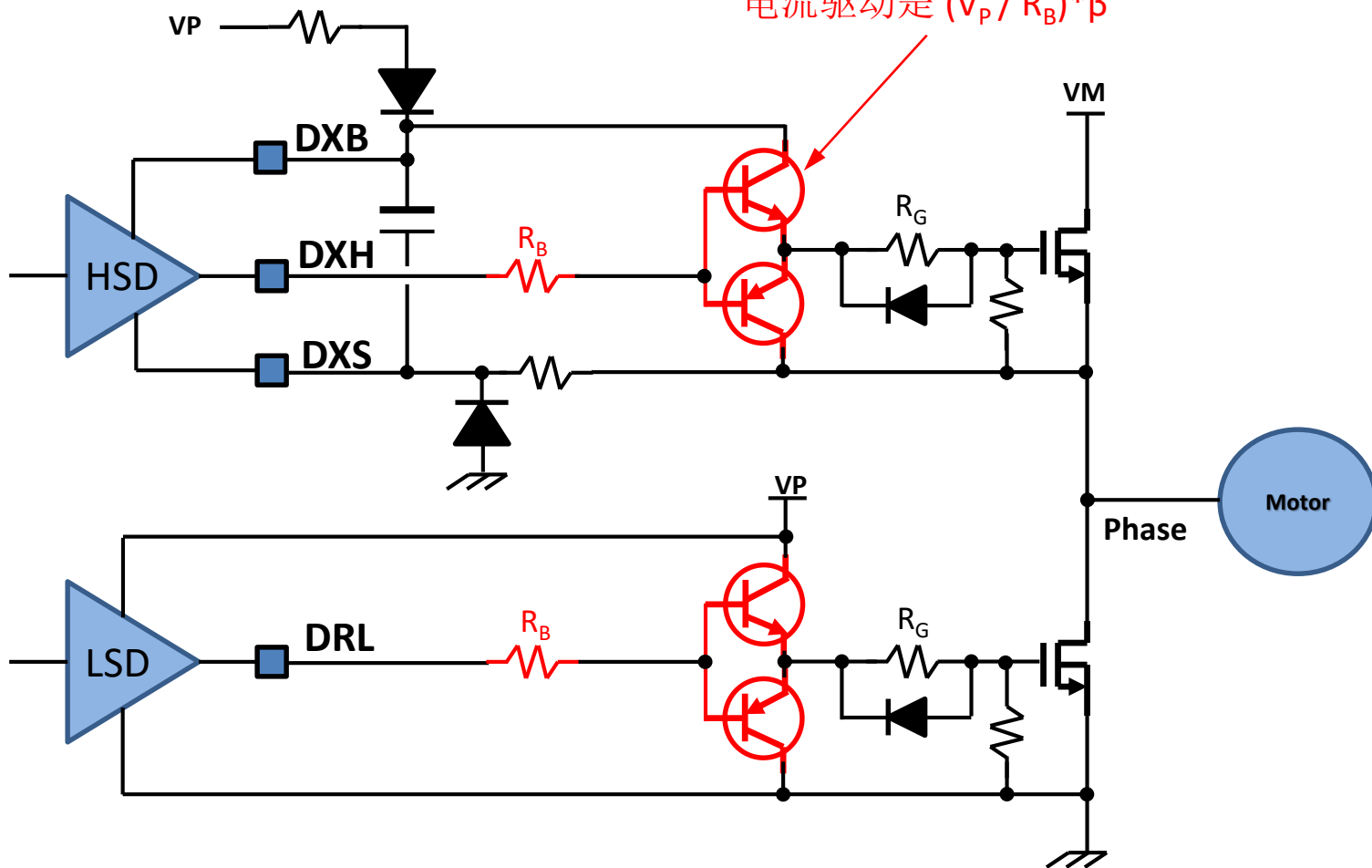
如左图所示，PAC内部分为两个部分，MCU和可模拟配置前端。可配置模拟前端控制ADC输入信号的选择，MCU控制软件的执行。可配置的模拟前端是MCU的一个外设，他们之间通过SPI总线及EMUX总线相互连接通信。EMUX是一个二位单向的快速通信口。可配置模拟前端和MCU之间需要通过EMUX来进行数据通信以保证系统功能的快速运行。

EMUX的通信指令由用户设定，不需要CPU介入。通常情况下EMUX通信用于ADC输入信号的选择。

**软复位导致的异常：**当MCU执行软件复位指令时，MCU被复位但并没有同时复位可配置的模拟前端。事实上存在一个小概率事件，也就是当可配置模拟前端和MCU正在通过EMUX通信时，此时MCU执行了软复位指令进行复位，正常进行的EMUX通信被强行中断，可配置模拟前端一直在等待完成一次完整的EMUX通信。MCU复位后，EMUX继续接收指令，但此时输入的指令已经和复位前未完成的指令混叠，导致EMUX指令错误，结果当然是ADC采样的数据也不是当前软件执行代码期望得到的数据。

**解决方案：**在 `cafe_init()` 子程序中先把EMUX使能位置“0”，这样EMUX被复位。之后再当EMUX使能位置“1”。

加NPN和PNP三极管组成乙类推挽增强栅极驱动能力。新的电流驱动是  $(V_p / R_B) * \beta$



PAC系列有二类 "soft reset":

SoftReset-1: 数字核看门狗或别的原因引发的MCU复位。这是MCU自生复位，对模拟核无影响。

SoftReset-2: 写模拟核寄存器(地址=14h, 7<sup>th</sup> Bit) "SRST"位，通过模拟核对MCU复位。

用户通过读二个标志位进行判断:

- 1) MCUALIVE                    // Address = 10h, 6<sup>th</sup> Bit
- 2) nSU                         // Address = 12h, 6<sup>th</sup> Bit

说明: nSU 是一个表征VHM电压状态的标志位。上电后用户把该位置"1"，只要VHM没有欠压，该标志位保持为"1"。同样上电后用户把MCUALIVE位置"1"。

不同状态的启动后有如下区别:

冷起动:            MCUALIVE=0 nSU=0

热起动:

SoftReset-1: MCUALIVE=1, nSU=1

SoftReset-2: MCUALIVE=0, nSU=1