Power notes:

1. GPIO
2. PMIC
3. Consumption measure points
4. DVFS
5. Dou
6. Suspend current fine tuning
7. Idle current fine tuning
   1. GPIO basic knowledge

做驱动最基本的就是要了解GPIO，不能做了多年的驱动对GPIO的了解仅仅停留在输出输入状态，这是很忌讳的。

1. 上拉下拉

节选了一段网上文章

那么在什么时候使用上、下拉电阻呢？

1、当TTL电路驱动CMOS电路时，如果TTL电路输出的高电平低于CMOS电路的

最低高电平（一般为3.5V），这时就需要在TTL的输出端接上拉电阻，以提高输出高电平的值。

2、OC门电路必须加上拉电阻，以提高输出的高电平值。

3、为加大输出引脚的驱动能力，有的单片机管脚上也常使用上拉电阻。

4、在CMOS芯片上，为了防止静电造成损坏，不用的管脚不能悬空，一般接上拉电阻降低输入阻抗，提供泄荷通路。

5、芯片的管脚加上拉电阻来提高输出电平，从而提高芯片输入信号的噪声容限，增强抗干扰能力。

6、提高总线的抗电磁干扰能力。管脚悬空就比较容易接受外界的电磁干扰。

7、长线传输中电阻不匹配容易引起反射波干扰，加上下拉电阻是电阻匹配，有效的抑制反射波干扰。

另外，上拉电阻阻值的选择原则包括:

1、从节约功耗及芯片的灌电流能力考虑应当足够大；电阻大，电流小。

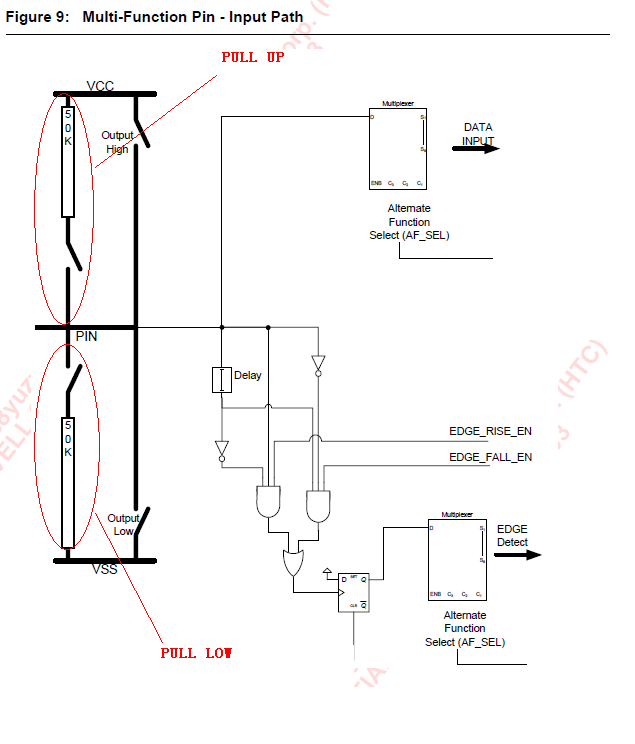
2、从确保足够的驱动电流考虑应当足够小；电阻小，电流大。

3、对于高速电路，过大的上拉电阻可能边沿变平缓。

综合考虑以上三点,通常在1k到10k之间选取。对下拉电阻也有类似道理

-----------引用结束！

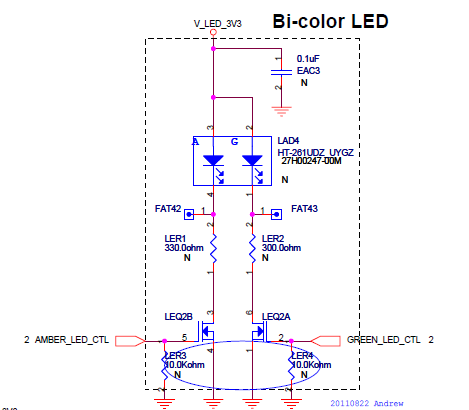
红色字体是常规做法。Marvell 用了50K来做上下拉电阻



1. 驱动能力

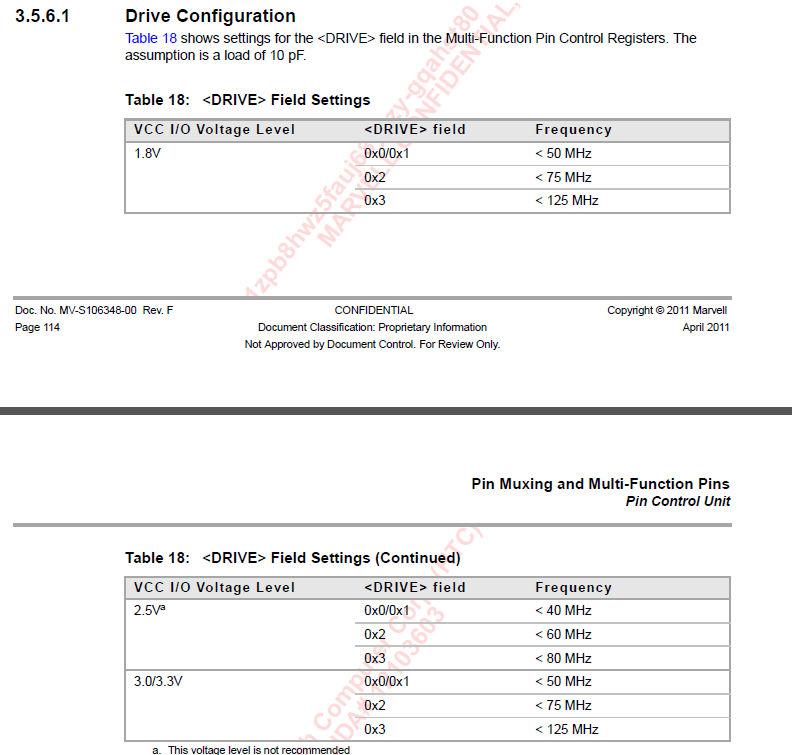
驱动能力分为输出能力 和输出响应频率

**驱动能力：**一般来说一个GPIO输出的能力是有限的，就比如你用一个GPIO输出为高电平（假设为5V），另外一端你接一个5V灯泡（或者手电筒灯泡）,发现这时候是无法将这个灯泡点亮的。因为GPIO的输出电流非常小，一般极限是10mA.为了提高驱动能力会加三极管或MOS管。常见的就是驱动LED时候（如下图所示）



**输出响应频率：**假设你想通过一个GPIO输出一定频率的PWM，发现这个PWM波形很差，甚至连一个三角波都算不上。这是因为 GPIO的输出频率也是有限制的，每个平台都不同，原则上主频越高，输出响应频率越大，但是不尽然。

Marvell平台的输出响应频率是可以通过寄存器调整（如下图所示）。



**GPIO Controller：** 现在由于芯片集成度越来越高，GPIO是以一个Controller单独存在于芯片中的，既然是单独的Controller，那么就需要给这个Controller 提供CLK 和Power。

所以有个平台的GPIO controller 在使用之前或者不用是都要开启/关闭 function CLK 和Power domain。STE 的8500就是这样的设计，mavell PXA920也是这样的设计

下面即是Marvell的代码

static void mfp\_gpio2\_power\_up(void)

{

\_\_raw\_writel(FIRST\_SECURITY\_VALUE,APBC\_PXA910\_ASFAR);

\_\_raw\_writel(SECOND\_SECURITY\_VALUE,APBC\_PXA910\_ASSAR);

\_\_raw\_writel(AIB\_POWER\_TURNON,AIB\_GPIO2\_IO);

}

static void mfp\_gpio2\_power\_down(void)

{

\_\_raw\_writel(FIRST\_SECURITY\_VALUE,APBC\_PXA910\_ASFAR);

\_\_raw\_writel(SECOND\_SECURITY\_VALUE,APBC\_PXA910\_ASSAR);

\_\_raw\_writel(AIB\_POWER\_SHUTDOWN,AIB\_GPIO2\_IO);

}

1. 输入输出

这个大家都了解，不过注意可以使用bitwise 的register 方式位操作GPIO，这样写可以不用考虑同组（eg:0~31,32~63,…）其他GPIO状态。

操作A组

#define GPSR(x) GPIO\_REG(BANK\_OFF(gpio\_to\_bank(x)) + 0x18)

B

A

#define GPCR(x) GPIO\_REG(BANK\_OFF(gpio\_to\_bank(x)) + 0x24

和操作B组是不一样的

//bit wise set gpio direction 1:output 0:input

#define GSDR(x) GPIO\_REG(BANK\_OFF(gpio\_to\_bank(x)) + 0x54)

//bit wise clear gpio direction 1:input 0:noeffect

#define GCDR(x) GPIO\_REG(BANK\_OFF(gpio\_to\_bank(x)) + 0x60)

1. 中断和唤醒

GPIO 具有中断功能，这个需要对芯片的Datasheet的仔细阅读，看看这个GPIO中断是否是独享中断号还是共享中断号，另外很多芯片不支持所有GPIO Wakeup功能，仅支持部分GPIO唤醒功能，这个需要注意。

1. Pin altfunction and pin function switch.

每个Pin 本身可以复用其他其他功能，比如做SD卡的pin能做GPIO或者其他，这个很常见也很容易理解，但是目前又出现了一种新的状态。就是内部SWITCH，比如GPIO24，marvell PXA920 有3个地方可以配置成GPIO24.

1

2.



3.



这样的一种现象是CPU内部的SWITCH造成的。

GPIO24

MMC1\_DAT7

ND\_IO11

ALT3

ALT2

ALT1

ALT3

ALT2

ALT1

* 1. GPIO table review

每个项目logic设计完之后，会召集相关人员开个concall,探讨有疑问的地方，或者你觉得不太合适的地方。这个要关注2个地方：一个是DATASHEET中这个pin是否有限制，第二他的逻辑电路是否行得通。