

## 第七章 时钟与电源管理

译者：王建壮

### 概述

时钟电源控制块由三部分组成：时钟控制，USB 控制和电源控制。

S3C2440A 中的时钟控制逻辑能生成包括 CPU 内部时钟(FCLK)在内的所需时钟信号。主时钟脉冲(HCLK)用于高级高精度系统总线外围设备。FCLK 应用于 CPU 的时钟，PCLK 用于 APB bus 各种接口设备。S3C2410 有两个锁相环，一个用于 FCLK, HCLK 和 PCLK, 另一个用于 USB 块(48MHZ)。时钟控制逻辑能产生没有锁相环(PLL)的慢时钟和链接或不链接提供给外围设备块的软件时钟，可以降低功耗。

关于电源控制逻辑，S3C2410A 有多种电源配置方案来保证对假定任务的最佳功耗。电源控制块在 S3C2440 中能被激活成四种模式：通用模式，慢模式，空闲模式和睡觉模式。

**通用模式：**此块为 S3C2440A 中的 CPU 和所有外围设备提供时钟。这种模式下，在设备都运行时，电源功耗将最大。它允许用户用软件控制外设的运作。例如，一个定时器不需要啦，用户可以断开提供给这个定时器的时钟以降低功耗。

**慢模式：**无锁相环模式。与通用模式不同，慢模式用外部时

钟（XT1p11 或 EXTCLK）直接作为 S3C2440A 无锁相环的内部时钟。这种模式中，能耗只取决于外部时钟的频率。锁相环的功耗取消。

**空闲模式：**此块在为其他外设提供时钟时，停止对 CPU 内核提供时钟。空闲模式可降低 cpu 产生的功耗。任何对 CPU 的中断请求可以在空闲模式中被唤醒。

**掉电模式：**此块断开内部能源的提供。因此，不会产生 CPU 功耗，如果不在此模式中唤醒内部逻辑，也不会产生内部逻辑功耗。激活掉电模式需要两个独立电源。一个电源用于提供唤醒逻辑控制的能源。另一个用于支持包括 CPU 在内的其内部逻辑块，并且可用电源开关控制。在掉电模式，这第二个用于支持包括 CPU 在内的其内部逻辑块的电源被关闭。掉电模式的启动唤醒取决于中断使能位[15:0]或者实时时钟警告中断。

## 功能描述：

### 时钟体系机构

图 7-1 显示了时钟结构的框图。主时钟源来自外部晶振（XT1p11）或一个外部时钟（EXTCLK）。

时钟发生器包括一个与外部晶振连接的震荡器（震荡放大器），同时也有两个锁相环（相位锁定循环），用于产生 S3C2440A 需要的高频率时钟。

时钟源选择

表格 7-1 显示了方式控制引脚（OM3，OM2）不同组合的关系

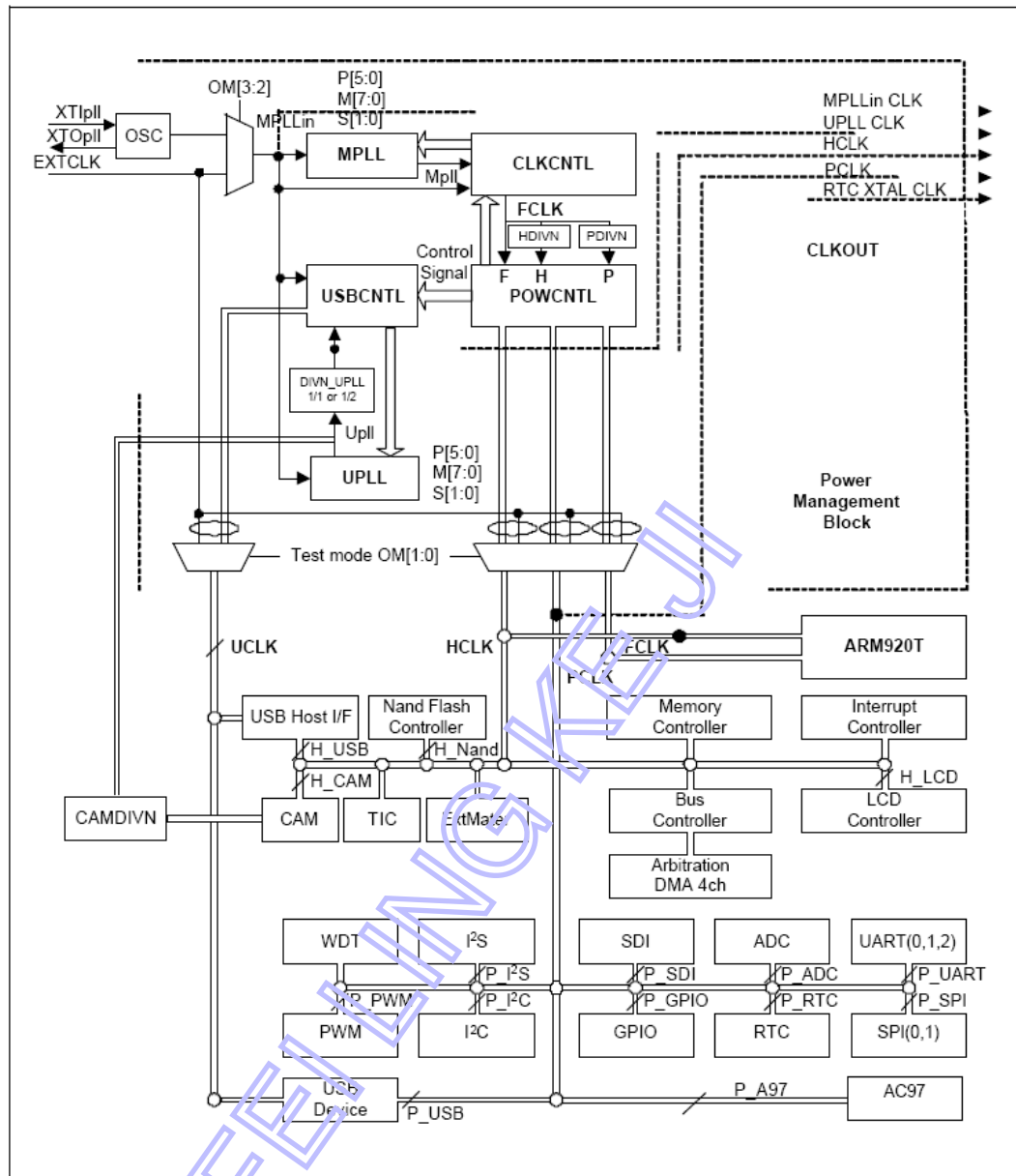
表格 7-1 启动时时钟脉冲源选择

模式引脚 [3:2]	主时钟锁 相环状态	USB时钟锁 相环状态	主时钟源	USB时钟源
00	开	开	晶振	晶振
01	开	开	晶振	外部时钟
10	开	开	外部时钟	晶振
11	开	开	外部时钟	外部时钟

及 S3C2410A 源时钟的选择。OM[3:2]状态决定于 OM3 和 OM2 引脚的复位信号的上升沿。

注意事项：1 虽然主锁相环在复位后就立即运作，但是它的输出不能用于系统时钟，除非软件方式向主锁相环寄存器输入有效的设置。在有效设置前，外部晶振时钟或外部晶振源会直接用做系统时钟。即使用户不想改变主锁相环寄存器的默认值，也要重新向寄存器写入同一个数值。

2. 当 OM[1: 0]是 11 时，OM[3:2]用于终止测试模式。



**Figure 7.4 Clock Generator Block Diagram**

## 图 7-1. 时钟发生器框图

## 锁相环

时钟发生器中的主锁相环，作为电路，在频率和相位上将一个输出信号与一个标准输入信号同步。在这个应用程序中，它包括以下图 7-2 所示基本模块：电压控制振荡器（VCO）生成一个与输入直流电频率成比例的输出。分频器 P 用 p 对

输入信号分频。分频器 M 用输入到相频检测器的 m 对 VCO 输出进行分频。分频器 S 用作为主锁相环(主锁相环输出频率)，相差检测器，进料泵，环路滤波器的 m 对 VCO 输出进行分频。输出频率主锁相环与标准输入时钟频率的关系如下方程式：

$$M_{pll} = (2^m * F_{in}) / (p * 2^s)$$
$$m = M \text{ (the value for divider M)} + 8, p = P \text{ (the value for divider P)} + 2$$

时钟发生器中的 USB 锁相环与主锁相环在各方面都很相似。下面部分描述锁相环的操作，包括相差检测器，电荷泵，电压控制振荡器和环路滤波器。

### 相频检测器 (PFD)

PFD 监视标准频率  $F_{ref}$  和 VCO 输出频率  $F_{vco}$ ，并当发现不同时产生控制信号（区间信号）。 $F_{ref}$  即如图 7-2 所示频率。

### 电荷泵 (PUMP)

进料泵将 PFD 控制信号按比例转化为电压通过过滤器来控制 VCO。

### 环路过滤器

PFD 对电荷泵产生的控制信号，在每次  $F_{vco}$  与  $F_{ref}$  进行比较时会产生大的偏移。为避免 VCO 过载，通过低通滤波器样本和滤波出控制信号的高频部分。这个滤波器是由一个电阻一个电容组成的标准单极 RC 滤波器。

### 电压控制振荡器 (VCO)

环路过滤器输出电压驱动 VCO，使其震荡频率作为基准电压

函数线性增减。当  $F_{vc}$  与  $F_{ref}$  在相频都匹配时，PFD 停止对电荷泵发控制信号，轮回稳定环路过滤器输入电压。VCO 频率保持恒值，PLL 保持与系统时钟一致。

PLL 通用条件和时钟发生器

PLL 与时钟发生器常规使用条件。

Loop filter capacitance	$C_{LF}$	MPLLAP: 1.3 nF ± 5%
		UPLLAP: 700 pF ± 5%
External X-tal frequency	–	12 – 20 MHz (note)
External capacitance used for X-tal	$C_{EXT}$	15 – 22 pF

注意：

- 1. 数值可以被改变。
- 2. 内部时钟输出必须大于 200MHz（这并不意味着 ARM 内核运行频率必须在 200MHz 以上）。

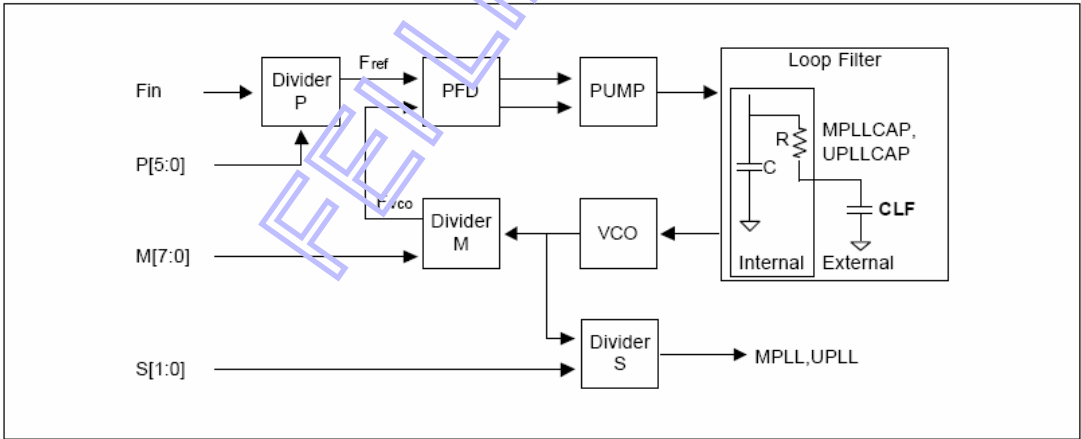


图 7-2 锁相环机构框图

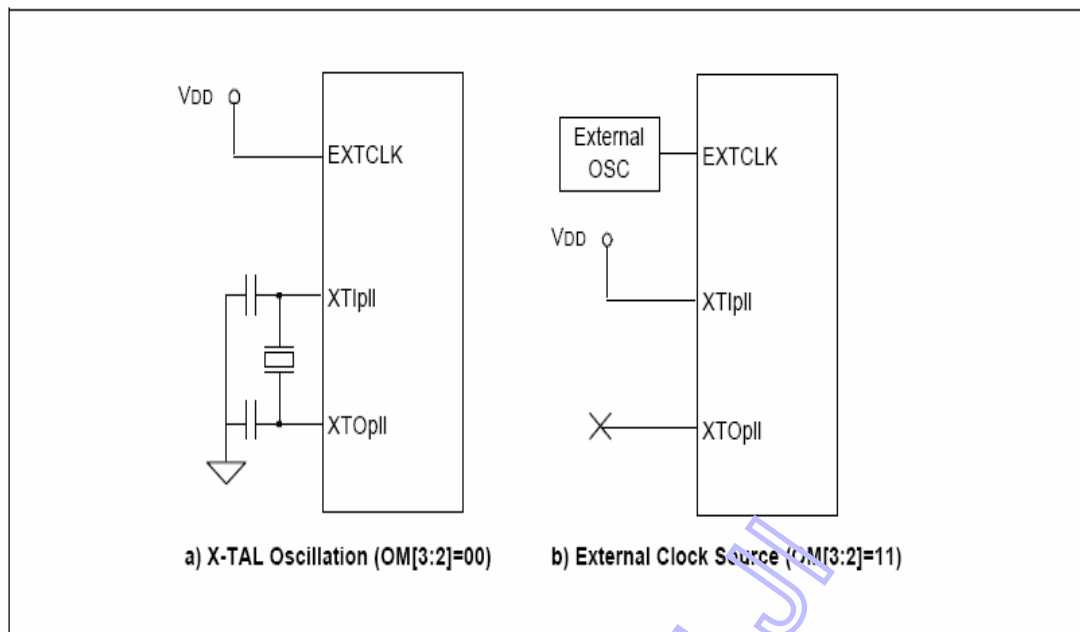


图 7-3 主振荡器电路实例

## 时钟控制逻辑

**时钟控制逻辑** 控制时钟的使用，i. e.，PLL 时钟或直接外部时钟 (Xt1p11 或 EXTCLK)。当 PLL 配置新频率值时，只有 PLL 输出稳定的使用 PLL 同步节拍时时钟控制逻辑才能使能 FCLK。时钟控制逻辑在通电复位和从掉电模式唤醒时被激活。

## 启动复位 (XTp11)

图 7-4 显示启动复位过程中时钟轨迹。晶振以数毫秒的频率开始震荡。当复位信号在稳定震荡量时钟后释放时，PLL 开始根据默认 PLL 配置运行。然而，普遍认为 PLL 在重启时不稳定，所以输入频率在软件重新设置 PLLCON 被 FCLK 直接设置来代替 Mp11 (P11 输出)。即使用户在复位后不想改变 PLLCON 寄存器的默认值，也要软件方式写入同样的数值。

PLL 只有在软件设置被设置新的频率后，才能以新的频率重启。

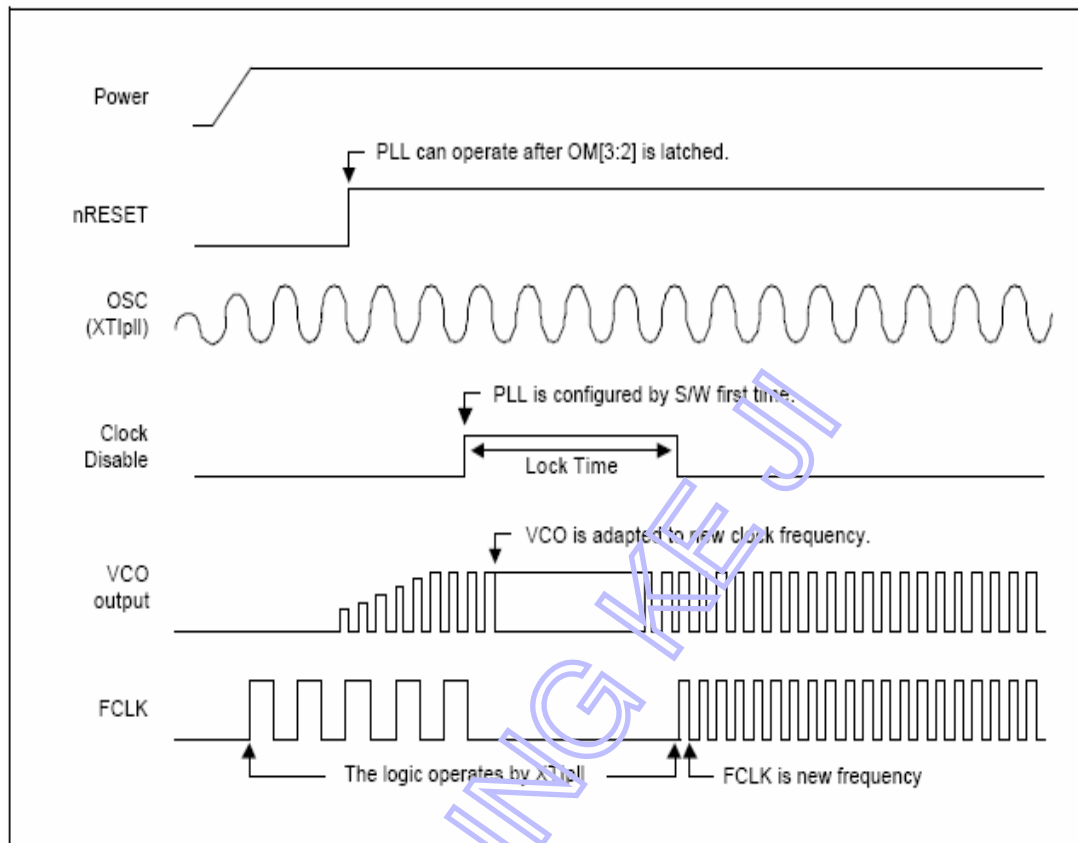


图 7-4 重启顺序（当外部时钟源是晶振）

### 常规操作模式中改变 PLL 设置

在 S3C2440A 在通用模式下运行时，用户可以通过写 PMS 值来改变频率，PLL 锁存时间会自动注入。在锁存时间内，时钟不支持 S 3 C 2 4 4 0 A。图 7-5 显示时间矢量图。



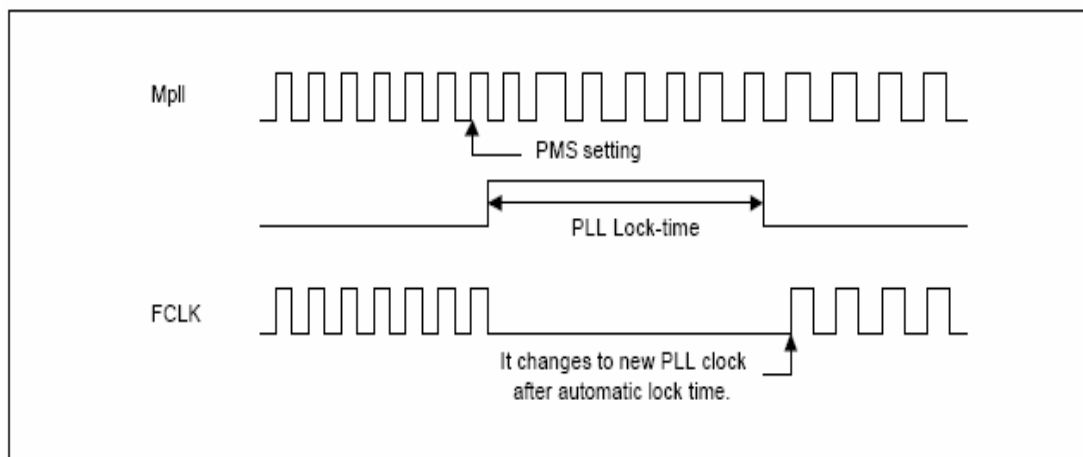


图 7-5 通过设置 PMS 值改变慢时钟

## USB 时钟控制

USB 主机接口和驱动接口需要 48Mhz 时钟。在 S3C2440A 中，USB 专用 PLL (UPLL) 产生为 USB 产生 48Mhz 频率。只有 UPLL 在设置后才能反馈。

Condition	UCLK State	UPLL State
After reset	XTipll or EXTCLK	On
After UPLL configuration	L: During PLL lock time 48MHz: After PLL lock time	On
UPLL is turned off by CLKSLW register	XTipll or EXTCLK	Off
UPLL is turned on by CLKSLW register	48MHz	On

FCLK, HCLK, PCLK

FCLK 由 ARM920T 使用。

HCLK 用于高精度系统总线，由 ARM920T 使用，，中断控制，存储器控制，LCD 控制，DMA 与 USB 主时钟。

PCLK 用于 APB 总线，被外设使用 (WDT, IIS, I2C, PWM timer, MMC 接口, ADC, UART, GPIO, RTC 和 SPI)。

S3C2410A 支持 FCLK, HLCK, PCLK 之间比例的选择。此比例由 CLKDIVN 控制寄存器的 HDIVN 和 PDIVH 决定。

HDIVN	PDIVN	HCLK3_HALF/ HCLK4_HALF	FCLK	HCLK	PCLK	Divide Ratio
0	0	—	FCLK	FCLK	FCLK	1 : 1 : 1 (Default)
0	1	—	FCLK	FCLK	FCLK / 2	1 : 1 : 2
1	0	—	FCLK	FCLK / 2	FCLK / 2	1 : 2 : 2
1	1	—	FCLK	FCLK / 2	FCLK / 4	1 : 2 : 4
3	0	0 / 0	FCLK	FCLK / 3	FCLK / 3	1 : 3 : 3
3	1	0 / 0	FCLK	FCLK / 3	FCLK / 6	1 : 3 : 6
3	0	1 / 0	FCLK	FCLK / 6	FCLK / 6	1 : 6 : 6
3	1	1 / 0	FCLK	FCLK / 6	FCLK / 12	1 : 6 : 12
2	0	0 / 0	FCLK	FCLK / 4	FCLK / 4	1 : 4 : 4
2	1	0 / 0	FCLK	FCLK / 4	FCLK / 8	1 : 4 : 8
2	0	0 / 1	FCLK	FCLK / 8	FCLK / 8	1 : 8 : 8
2	1	0 / 1	FCLK	FCLK / 8	FCLK / 16	1 : 8 : 16

设置完 PMS 值后，必须设置 CLKDIVN 寄存器。CLKDIVN 数值的设置将在 PLL 锁存时间后生效。这个数值还可用于复位和改变电源管理模式。

这个设置值在 1.5 倍主时钟后同样有效。1HCLK 能改变 CLKDIVN 寄存器的值，从默认 (1: 1: 1) 到其他分离比例 (1: 1: 2, 1: 2: 2, 1: 2: 4)。

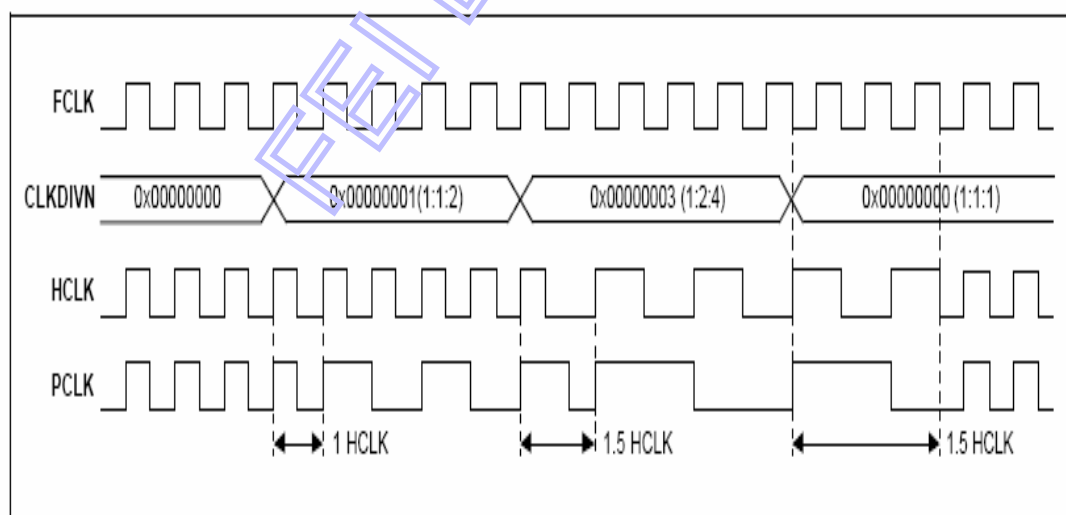


图 7-6，内部时钟改变实例

注意事项：

- 1 CLKDIVN 要谨慎设置以防止超过 HCLK 和 PCLK 极限。

2. 如果 HDIVN 不是 0，CPU 总线模式必须使用以下指令从快总线模式改为异步总线模式（S3C2440A 不支持同步总线模式）。

**MUU\_SetAsyncBusMode**

```
mrc p15, 0, r0, c1, c0, 0
```

```
orr ro, ro, #R1_nF:0R:R1_iA
```

```
mcr p15, 0, r0, c1, c0, 0
```

如果 HDIVH 不是 0 并且 CPU 总线模式是快总线模式，CPU 将由 HCLK 控制。这个特点能被用于在不改变 HCLK 和 PCLK 的情况下使 CPU 频率减半或更多。

## 电源控制

电源控制模块软件方式控制系统时钟来降低在 S3C2440A 中的功耗。这些设置涉及到 PLL，时钟控制逻辑（FCLK, HCLK, PCLK）和唤醒信号。

图 7-7 显示 S3C2440A 时钟分配。

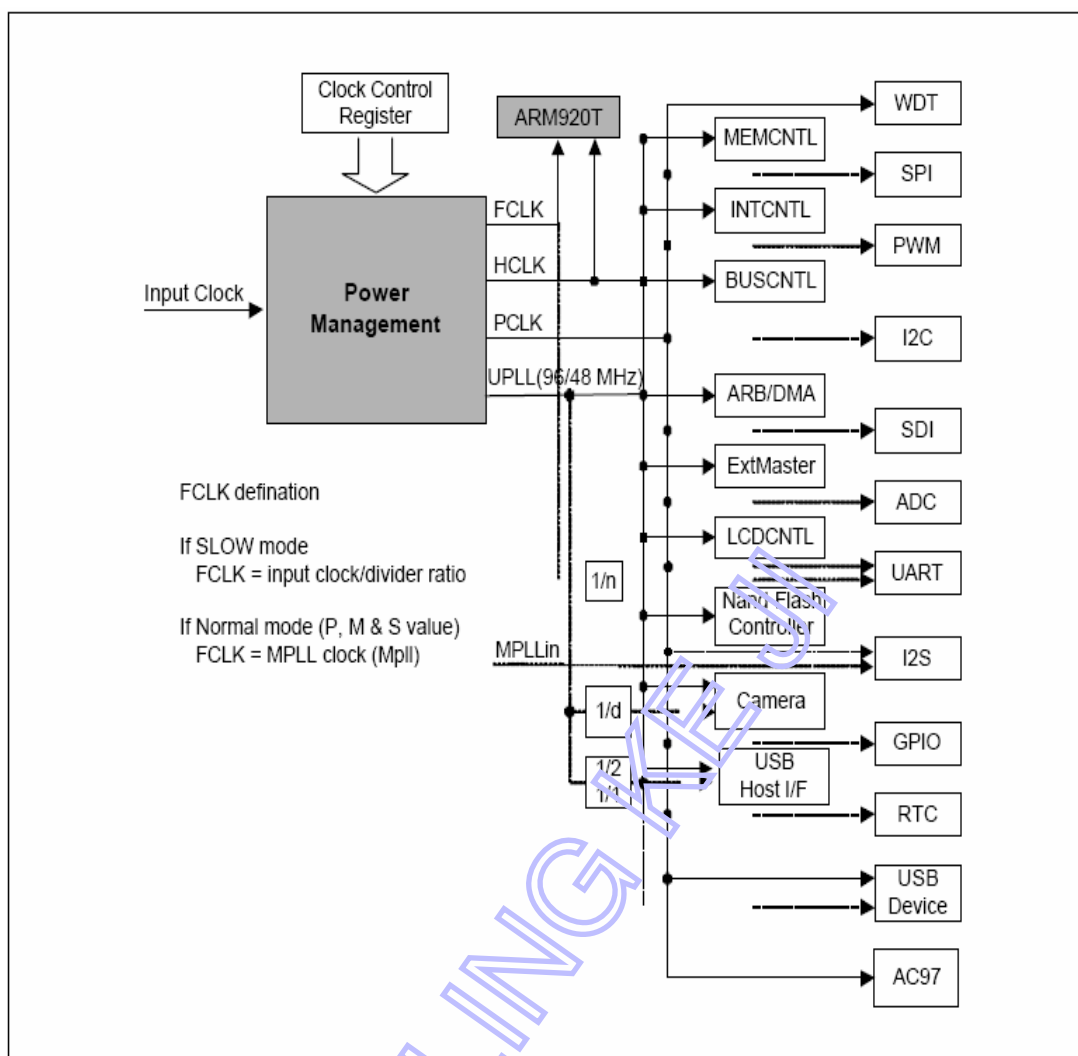


图 7-7 。时钟接线框图

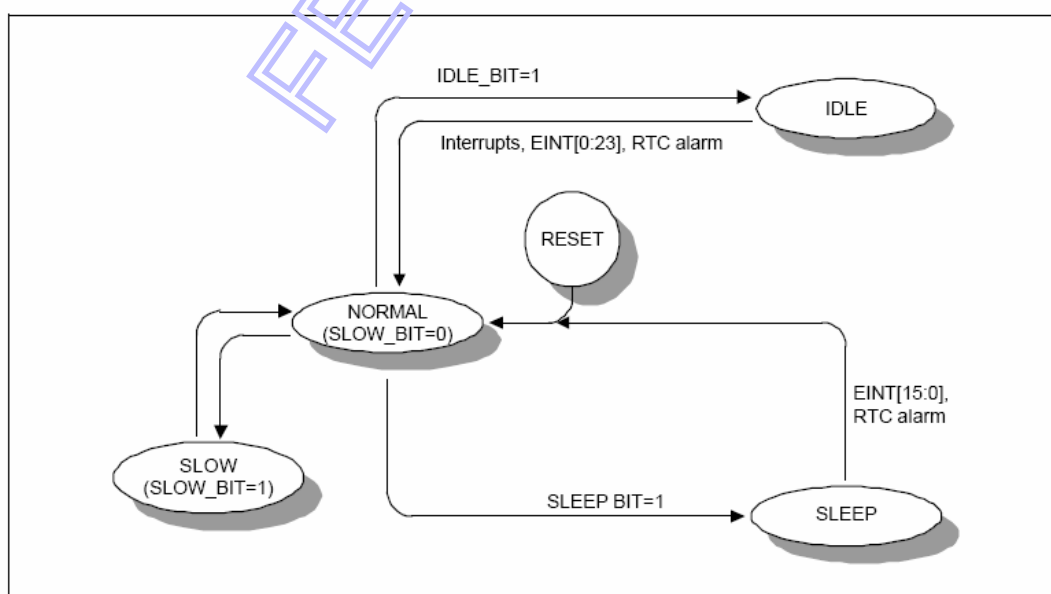


图 7-8 。电源控制状态图

表格 7-2. 不同电源模式中时钟和电源状态

Mode	ARM920T	AHB Modules <sup>(1)</sup> /WDT	Power Management	GPIO	32.768kHz RTC clock	APB Modules <sup>(2)</sup> & USBH/LCD/NAND
NORMAL	O	O	O	SEL	O	SEL
IDLE	X	O	O	SEL	O	SEL
SLOW	O	O	O	SEL	O	SEL
SLEEP	OFF	OFF	Wait for wake- up event	Previous state	O	OFF

**注意事项:** 1. USB 主机, LCD, flash 排除在外。  
2. WDT 排除在外, CPU 通道的 RTC 接口被包含。  
3. 选择: 精选 (O, X), O: 使能, X: 禁用 OFF:关掉电源

**通用模式**

标准模式中, 所有外设和基本块包括电源控制块, cpu 内核, 总线控器, 存储器控制器, 中断控制器, DMA 和外部控制都可完全操纵。

但是, 对于各个外设的时钟(基本块除外)可以用软件方式有选择的停用来降低能耗。

**空闲模式**

在空闲模式, 除了总线控制, 存储器控制, 中断控制, 电源控制块外, 对 CPU 内核的时钟被停用。退出空闲模式, EINT[23:0], 或实时时钟警告中断, 或其他中断被激活。(EINT 在 GPIO 块开启时才被激活)

**慢速模式 (无锁相环模式)**

在慢速模式中通过提供慢速时钟降低电源功耗, 包括 PLL 的电源功耗。FCLK 是没有 PLL 的输入时钟的的多分频。

表格 7-3 慢速时钟实例的 CLKSLOW 和 CLKDIVN 寄存器设置

SLOW_VAL	FCLK	HCLK		PCLK		UCLK
		1/1 Option (HDIVN=0)	1/2 Option (HDIVN=1)	1/1 Option (PDIVN=0)	1/2 Option (PDIVN=1)	
0 0 0	EXTCLK or XTIpll / 1	EXTCLK or XTIpll / 1	EXTCLK or XTIpll / 2	HCLK	HCLK / 2	48 MHz
0 0 1	EXTCLK or XTIpll / 2	EXTCLK or XTIpll / 2	EXTCLK or XTIpll / 4	HCLK	HCLK / 2	48 MHz
0 1 0	EXTCLK or XTIpll / 4	EXTCLK or XTIpll / 4	EXTCLK or XTIpll / 8	HCLK	HCLK / 2	48 MHz
0 1 1	EXTCLK or XTIpll / 6	EXTCLK or XTIpll / 6	EXTCLK or XTIpll / 12	HCLK	HCLK / 2	48 MHz
1 0 0	EXTCLK or XTIpll / 8	EXTCLK or XTIpll / 8	EXTCLK or XTIpll / 16	HCLK	HCLK / 2	48 MHz
1 0 1	EXTCLK or XTIpll / 10	EXTCLK or XTIpll / 10	EXTCLK or XTIpll / 20	HCLK	HCLK / 2	48 MHz
1 1 0	EXTCLK or XTIpll / 12	EXTCLK or XTIpll / 12	EXTCLK or XTIpll / 24	HCLK	HCLK / 2	48 MHz
1 1 1	EXTCLK or XTIpll / 14	EXTCLK or XTIpll / 14	EXTCLK or XTIpll / 28	HCLK	HCLK / 2	48 MHz

在慢速模式，PLL 会关闭来降低功耗。当 PLL 在慢速模式中关闭并且用户将电源模式从慢速模式改为通用模式时，PLL 需要时钟稳定时间（PLL 锁存时间）。PLL 稳定时间被内部逻辑用锁存时间计数寄存器自动嵌入。PLL 稳定时间将在 PLL 开启后占用 300us。在 PLL 锁存时段中，FCLK 变成慢时钟。用户可以在 PLL 导通状态下通过使能 CLKSLOW 寄存器的慢速模式位来改变频率。慢时钟在慢速模式中被发出。图 7-11（请检验图形正确性）显示时间矢量图。

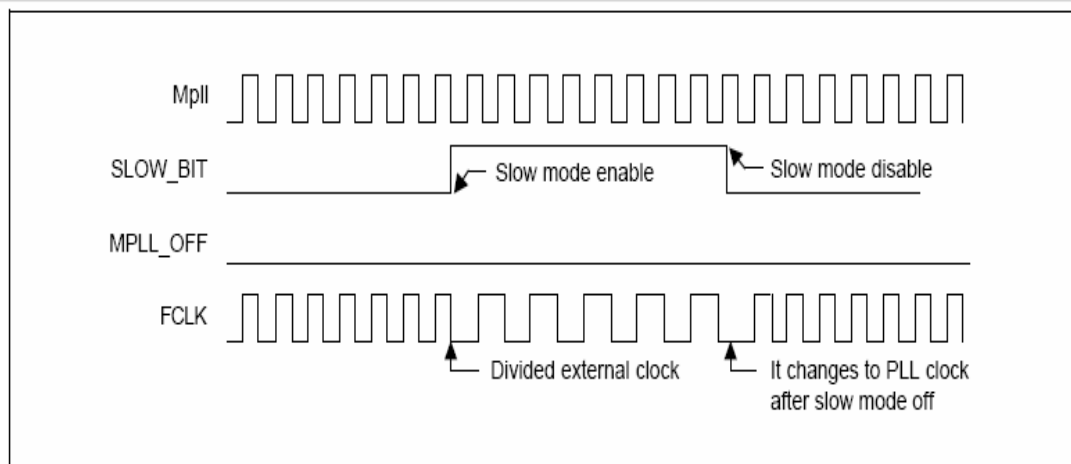


图 7-9 在 PLL 导通状态 Exit\_from\_mode 指令使用结果

如果用户在 PLL 锁存时间后通过使能 CLKSLOW 寄存器的 SLOW\_BIT 位使将慢模式转换为通用模式频率在慢速模式被禁止后，立即改变。图 7-12（请检查图形数字正确）显示时间适量图。

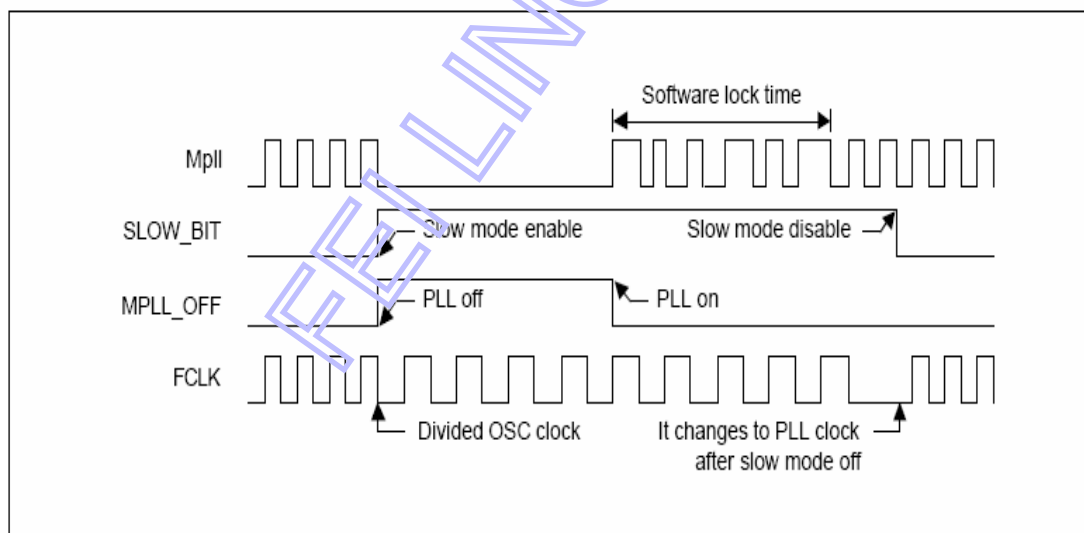


图 7-10. 锁存时间后使用 Exit\_from\_Slow\_mode 命令结果显示

如果用户通过同时禁用 CLKSLOW 中的 SLOW\_BIT 和 MPLL\_OFF 位来实现慢速模式到通用模式的切换，频率在 PLL 锁存时间

后立即改变。图 7—13（请核对图形数字，确保正确）显示时间矢量图。

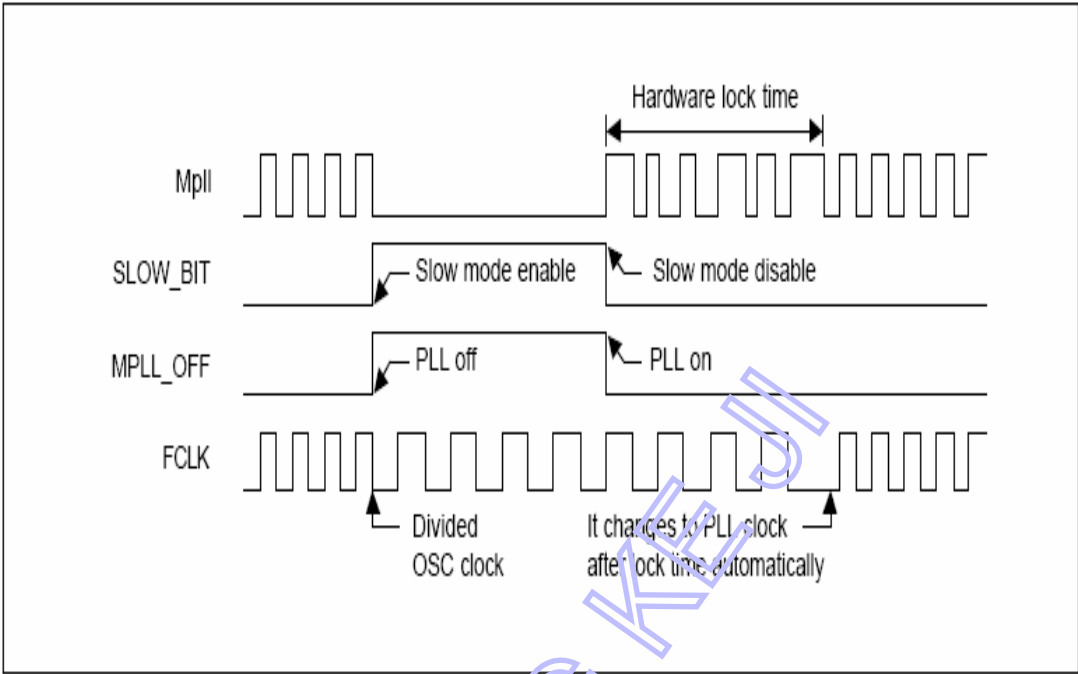


图 7-11. 同时使能 Exit\_from\_Slow\_mode 和 PLL\_on 指令结果显示

### 掉电模式

本块分开内部电源。所以，在这个模式中没有 CPU 功耗和除去唤醒逻辑外的所有内部逻辑功耗。激活掉电模式需要两个独立电源。一个电源为唤醒逻辑供电。另一个为 CPU 和其他所有的内部逻辑供电，所以，将应该控制电源的开关。在掉电模式，第二个为 CPU 和内部逻辑供电的电源被关闭。可通过 EINT[15:0]或者实时时钟警报中断来将状态从掉电模式唤醒。

### 按照以下流程进入睡眠模式

1. 为掉电模式合理的设置 GPIO。



- 2. 在中断屏蔽寄存器中屏蔽所有中断。
- 3. 合理配置包括实时时钟在内的唤醒源。(EINTMASK 寄存器中对应启动源的位没有被屏蔽目的是置位 SRCPND 或者 EINTEND 的相应位。虽然一个唤醒源启动，并且 EINTMASK 相应位被屏蔽， 唤醒仍然会发生，SRCPND 和 EINTPEND 的相应位不被设置)。
- 4. 设置 USB pads 作为挂起模式。(MISCCR[13:12]=11b)
- 5. 保存一些特殊含义值到 GSTATUS[4:3]寄存器中。寄存器在睡眠模式中被保护。
- 6. GSTATUS[3:4]的信息可被用于用户自己个人意图， 因为 GSTATUS[3:4]中的数值在睡眠模式中被保存过。
- 7. EINT[3:0]用于检测 SRCPND 寄存器。

EINT[15:4],确定 EINTPEND 代替 SRCPND(虽然 EINTPEND 的一些位被置位， SRCPND 将不被置位， )

表格 7-4 掉电模式下的引脚配置

引脚状态		引脚描述向导
GPIO 引脚	设置为输入	上拉使能
	设置为输出	上拉关使能， 输出低电平
输入引脚， 没有内部终止控制	假设外部驱动没有持续驱动引脚电平	用外部上拉寄存器上拉使能

输出引脚，链接外部驱动		假设驱动电源断开	输出低电平
		假设驱动电源打开	高或低（取决于外部驱动状态）
数据总线	假设存储器电源断开		输出低电平
	假设存储器电源启用	外部缓冲器存在	如果缓冲器能保持总线电平，上拉位使能
		没有外部缓存	输出低电平

注意事项：

1. ADC 应设置为备用模式。
2. USB pads 应设置为挂起模式。

\*此表仅作为使用信息，用户应考虑到自己板子条件和应用程序。

### VDDi 和 VDDiarm 的电源控制

在休眠状态下，被引脚 PWREN 控制的 VDDi, VDDiarm, VDDMPPL 和 VDDUPLL 将关闭。如果 PWREN 信号是高电平，VDDi 和 VDDiarm 由一个外部稳压器支持。如果 PWREN 引脚是低电平，VDDiarm 关闭。

注意事项：

虽然 VDDi, VDDiarm, VDDMPPL 和 VDDUPLL 可以被关闭，但其他的动力引脚必须被支持。

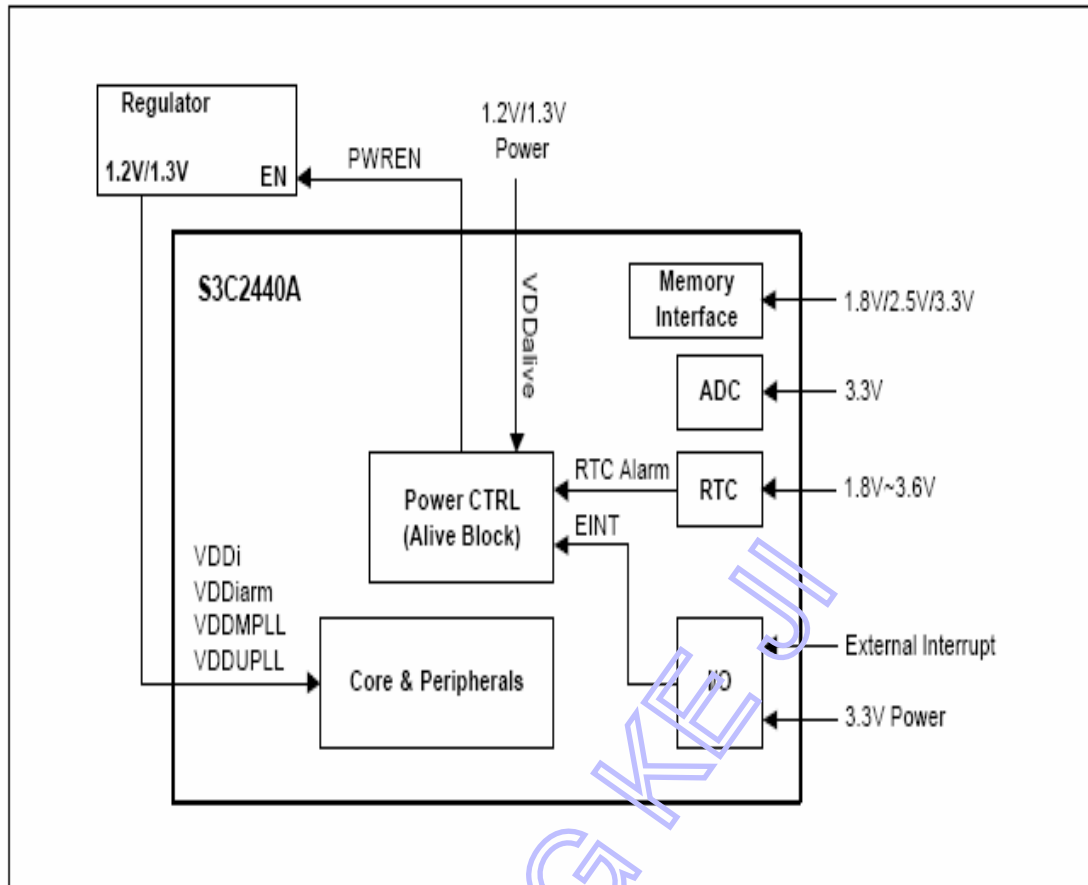


图 7-12. 掉电模式

掉电模式中，如果你不想使用触摸屏面板，触摸屏端口（XP, XM, YP 和 YM）必须拔掉。因为，触摸端口（XP, XM, YP 和 YM）不能接地。因为 XP, YP 在睡眠时间保持高电平。

### 发信号 EINT[15:0]用于唤醒

只要下面条件满足 S3C2440A 可以被从掉电模式唤醒

- 电平信号（H or L）或者沿信号（上升下降或都有）在 EINTn 输入引脚被激活。
- EINTn 引脚必须在 GPIO 控制器中设置为 EINT。
- nBATT\_FLT 引脚必须为高电平。考虑到上面的条件 a) 在 GPIO 控制寄存器中将 EINTn 设置为外部中断引脚变得十

分重要。

在唤醒后，相应的 EINTn 引脚将不再被用于唤醒。也就是说它可以再次被当作一个外部中断请求引脚使用。

### 进入空闲模式

如果 CLKCON[2] 被置 1 则进入空闲模式，S3C2440A 在少许延时后进入空闲模式（直到电源控制逻辑从 CPU 封装接收到正确应答信号）。

### PLL 开关

PLL 只能在慢模式中被关闭以达到降低功耗的目的。如果在其他任何模式下关闭，MCU 的操作将失去保证。

如果处理器在慢模式下并且想在 PLL 开启状态下切换其状态，SLOW\_BIT 必须被清零来在 PLL 稳定时移动到另一个状态。

### 数据总线和掉电状态的上拉电阻。

在休眠模式，数据总线（D[31:0] 或 D[15:0]）可被选择为 Hi-z 状态和输出低状态。

数据总线可以通过开启上拉电阻设置成 Hi-z 状态或通过关闭上拉电阻设置为低输出模式来在睡眠模式中降低功耗。

D[31: 0] 引脚上拉电阻可以用 GPIO 控制寄存器控制。如果有在数据总线上有外部总线占有者，如 74LVCH162245，用户可选择两种状态中的一种——其中一个上拉断开实现低电平输出，另一个是 Hi\_z with pull-up 0ff -，降低功耗。

### 输出端口状态和掉电模式

输出端口要在掉电模式下有一个适当的逻辑电平，使电流损耗最低。如果没有装载一个输出端口引脚，高电平是首选。如果输出低电平，电流流过内部寄生电阻；如果输出是高电平，不消耗电流。如果输出状态是高电平，输出端口的电流损耗可被降低。

建议在掉电模式下输出端口为高电平状态来降低电流损耗。

### 电池故障信号 (nBATT\_FLT)

nBATT\_FLT 引脚两种功能如下：

—— 当 CPU 不在掉电模式，通过设置 BATT\_FUNC(MISCCR[22:20])为 10'b 可使 nBATT\_FLT 引脚发出中断请求，中断请求为低电平触发。

—— 当 CPU 在睡眠模式下，通过设置 BATT\_FUNC(MISCCR[22:20])为 11x'b 完成 nBATT\_FLT 的激活，将阻止掉电模式的唤醒。所以，如果 nBATT\_FLT 激活，任何唤醒源将被屏蔽，来低电量的系统故障。

### 模数转换掉电

ADC 在 ADCCON 中有一个额外的掉电位，如果 S3C2410A 进入 SLEEP 模式，ADC 将进入自己地掉电模式。

### 时钟发生器和电源管理专用寄存器

锁 存 时 间 计 数 寄 存 器 ( locktime )

Register	Address	R/W	Description	Reset Value
LOCKTIME	0x4C000000	R/W	PLL lock time count register	0xFFFFFFFF

LOCKTIME	Bit	Description	Initial State
U_LTIME	[31:16]	UPLL lock time count value for UCLK. (U_LTIME > 300uS)	0xFFFF
M_LTIME	[15:0]	MPLL lock time count value for FCLK, HCLK, and PCLK (M_LTIME > 300uS)	0xFFFF

### MPLL 控制寄存器

$$M_{pll} = (2 * m * F_{in}) / (p * 2^s)$$

$$m = (MDIV + 8), p = (PDIV + 2), s = SDIV$$

### UPLL 控制寄存器

$$U_{pll} = (m * F_{in}) / (p * 2^s)$$

$$m = (MDIV + 8), p = (PDIV + 2), s = SDIV$$

### P11 值设置帮助 (MPLLCON)

$$1. F_{OUT} = 2 * m * F_{in} / (p * 2^s), F_{VCO} = 2 * m * F_{in} / p \text{ where: } m=MDIV+8, p=PDIV+2, s=SDIV$$

$$2. 600MHz \leq F_{VCO} \leq 1.2GHz$$

$$3. 200MHz \leq F_{CLK_{OUT}} \leq 600MHz$$

3. 不能把 P 或 M 值设置为 0，也就是说设置 P=00000000，M=00000000 会引起 PLL 故障。

5, P 和 M 的合理范围：  $1 \leq P \leq 62$ ,  $1 \leq M \leq 248$

### 注意事项：

虽然已经列出 PLL 的选择值，我们仍建议只使用建议表格中

的数值。如果您一定要使用其他的数值，请联系我们。

PLL 控制寄存器（MPLLCON 与 UPLLCON）

Register	Address	R/W	Description	Reset Value
MPLLCON	0x4C000004	R/W	MPLL 配置寄存器	0x00096030
UPLLCON	0x4C000008	R/W	UPLL 配置寄存器	0x0004d030

PLLCON	Bit	Description	Initial State
MDIV	[19:12]	主分割器控制	0x96 / 0x4d
PDIV	[9:4]	预分割器控制	0x03 / 0x03
SDIV	[1:0]	快速分割器控制	0x0 / 0x0

**注意：**当您设定 MPLL 和 UPLL 值时，必须先设置 Up11 在设置 MPLL 值。（需要大约七个延时单元）

### PLL 值选择表格

发现一个合适的 PLL 值不是很容易。所以我们推荐查阅如下推荐表的 PLL 值。

Input Frequency	Output Frequency	MDIV	PDIV	SDIV
12.0000MHz	48.00 MHz (Note)	<b>56(0x38)</b>	2	2
12.0000MHz	96.00 MHz (Note)	56(0x38)	2	1
12.0000MHz	271.50 MHz	173(0xad)	2	2
12.0000MHz	304.00 MHz	68(0x44)	1	1
12.0000MHz	405.00 MHz	127(0x7f)	2	1
12.0000MHz	532.00 MHz	125(0x7d)	1	1
16.9344MHz	47.98 MHz (Note)	60(0x3c)	4	2
16.9344MHz	95.96 MHz (Note)	60(0x3c)	4	1
16.9344MHz	<b>266.72 MHz</b>	118(0x76)	2	2
16.9344MHz	<b>296.35 MHz</b>	97(0x61)	1	2
16.9344MHz	399.65 MHz	110(0x6e)	3	1
16.9344MHz	530.61 MHz	83(0x56)	1	1
16.9344MHz	533.43 MHz	118(0x76)	1	1

注意：48.00MHz 和 96MHz 输出用于 UPLLCOMN 寄存器。

### 时钟控制寄存器（CLKCON）

Register	Address	R/W	Description	Reset Value
CLKCON	0x4C00000C	R/W	Clock generator control register	0xFFFFF0



CLKCON	Bit	Description	Initial State
AC97	[20]	Control PCLK into AC97 block. 0 = Disable, 1 = Enable	1
Camera	[19]	Control HCLK into Camera block. 0 = Disable, 1 = Enable	1
SPI	[18]	Control PCLK into SPI block. 0 = Disable, 1 = Enable	1
IIS	[17]	Control PCLK into IIS block. 0 = Disable, 1 = Enable	1
IIC	[16]	Control PCLK into IIC block. 0 = Disable, 1 = Enable	1
ADC(&Touch Screen)	[15]	Control PCLK into ADC block. 0 = Disable, 1 = Enable	1
RTC	[14]	Control PCLK into RTC control block. Even if this bit is cleared to 0, RTC timer is alive. 0 = Disable, 1 = Enable	1
GPIO	[13]	Control PCLK into GPIO block. 0 = Disable, 1 = Enable	1
UART2	[12]	Control PCLK into UART2 block. 0 = Disable, 1 = Enable	1
UART1	[11]	Control PCLK into UART1 block. 0 = Disable, 1 = Enable	1
UART0	[10]	Control PCLK into UART0 block. 0 = Disable, 1 = Enable	1
SDI	[9]	Control PCLK into SDI interface block. 0 = Disable, 1 = Enable	1
PWMTIMER	[8]	Control PCLK into PWMTIMER block. 0 = Disable, 1 = Enable	1
USB device	[7]	Control PCLK into USB device block. 0 = Disable, 1 = Enable	1
USB host	[6]	Control HCLK into USB host block. 0 = Disable, 1 = Enable	1

LCDC	[5]	Control HCLK into LCDC block. 0 = Disable, 1 = Enable	1
NAND Flash Controller	[4]	Control HCLK into NAND Flash Controller block. 0 = Disable, 1 = Enable	1
SLEEP	[3]	Control SLEEP mode of S3C2440A. 0 = Disable, 1 = Transition to SLEEP mode	0
IDLE BIT	[2]	Enter IDLE mode. This bit is not cleared automatically. 0 = Disable, 1 = Transition to IDLE mode	0
Reserved	[1:0]	Reserved	0

## 时钟慢速控制（CLKSLOW）寄存器

Register	Address	R/W	Description	Reset Value
CLKSLOW	0x4C000010	R/W	Slow clock control register	0x00000004

CLKSLOW	Bit	Description	Initial State
UCLK_ON	[7]	0: UCLK ON (UPLL is also turned on and the UPLL lock time is inserted automatically.) 1: UCLK OFF (UPLL is also turned off.)	0
Reserved	[6]	Reserved	–
MPLL_OFF	[5]	0: Turn on PLL. After PLL stabilization time (minimum 300us), SLOW_BIT can be cleared to 0. 1: Turn off PLL. PLL is turned off only when SLOW_BIT is 1.	0
SLOW_BIT	[4]	0 : FCLK = Mpll (MPLL output) 1: SLOW mode FCLK = input clock/(2xSLOW_VAL), when SLOW_VAL>0 FCLK = input clock, when SLOW_VAL=0. Input clock = XTlpll or EXTCLK	0
Reserved	[3]	–	–
SLOW_VAL	[2:0]	The divider value for the slow clock when SLOW_BIT is on.	0x4

## 时钟分配控制（CLKDIVN）控制器

Register	Address	R/W	Description	Reset Value
CLKDIVN	0x4C000014	R/W	Clock divider control register	0x00000000

CLKDIVN	Bit	Description	Initial State
DIVN_UPLL	[3]	UCLK select register(UCLK must be 48MHz for USB) 0: UCLK = UPLL clock 1: UCLK = UPLL clock / 2  Set to 0, when UPLL clock is set as 48MHz Set to 1. when UPLL clock is set as 96MHz.	0
HDIVN	[2:1]	00 : HCLK = FCLK/1. 01 : HCLK = FCLK/2.  10 : HCLK = FCLK/4 when CAMDIVN[9] = 0. HCLK= FCLK/8 when CAMDIVN[9] = 1.  11 : HCLK = FCLK/3 when CAMDIVN[8] = 0. HCLK = FCLK/6 when CAMDIVN[8] = 1.	00
PDIVN	[0]	0: PCLK has the clock same as the HCLK/1. 1: PCLK has the clock same as the HCLK/2.	0

## 照相机时钟分割（CAMDIVN）寄存器

Register	Address	R/W	Description	Reset Value
CAMDIVN	0x4C000018	R/W	Camera clock divider register	0x00000000

CAMDIVN	Bit	Description	Initial State
DVS_EN	[12]	0:DVS OFF ARM core will run normally with FCLK (MPLLout).  1:DVS ON ARM core will run at the same clock as system clock (HCLK).	0
Reserved	[11]	–	0
Reserved	[10]	–	0
HCLK4_HALF	[9]	HDIVN division rate change bit, when CLKDIVN[2:1]=10b. 0: HCLK = FCLK/4 1: HCLK = FCLK/8 Refer the CLKDIV register.	0
HCLK3_HALF	[8]	HDIVN division rate change bit, when CLKDIVN[2:1]=11b. 0: HCLK = FCLK/3 1: HCLK = FCLK/6 Refer the CLKDIV register.	0
CAMCLK_SEL	[4]	0: Use CAMCLK with UPLL output (CAMCLK=UPLL output). 1: CAMCLK is divided by CAMCLK_DIV value.	0
CAMCLK_DIV	[3:0]	CAMCLK divide factor setting register(0 – 15). Camera clock = UPLL / [(CAMCLK_DIV + 1)x2]. This bit is valid when CAMCLK_SEL=1.	0