

第七章 时钟及电源管理模块

时钟电源管理模块包含了 3 部分: Clock 控制、USB 控制、POWER 控制。

时钟控制逻辑单元能够产生 s3c2440 需要的时钟信号, 包括 CPU 使用的主频 FCLK, AHB 总线设备使用的 HCLK, 以及 APB 总线设备使用的 PCLK. 2440 内部有 2 个 PLL (锁相环): 一个对应 FCLK、HCLK、PCLK, 另外一个对应的是 USB 使用 (48MHz)。时钟控制逻辑单元可以在不使用 PLL 情况下降低时钟 CLOCK 的频率, 并且可以通过软件来驱使时钟和各个模块的连接/断开, 以减少电源消耗。

对于电源控制逻辑单元, 2440 有许多钟电源管理方案来针对所给的任务保持最优的电源消耗。S3c2440 中的电源管理模块对应 4 种模式: NORMAL 模式、SLOW 模式, IDLE 模式, SLEEP 模式。

NORMAL 模式: 这个模块给 CPU 时钟以及 2440 相应的外围设备提供时钟。这个模式下, 当所有的外围设备都被打开, 电源消耗被最大化。它允许用户通过软件来控制外部设备的操作。例如, 如果一个定时器不需要时, 那么用户可以通过 CLKCON 寄存器来关闭时钟和定时器的连接, 来降低电源消耗。

SLOW 模式: NON-PLL 模式, 不同于 Normal 模式, 这个模式使用的一个外部时钟 (XT1p11 或 EXTCLK) 来直接作为 2440 的主频 FCLK, 而没有使用 PLL。在这个模式下, 电源的消耗仅依赖于外部时钟频率, 电源同 PLL 有关的消耗可以被排除。

IDLE 模式: 这个模式下 CPU 的时钟 FCLK 被断开, 而还继续提供其他外围设备的时钟。因此空闲模式导致减少了 CPU 核相应的电源消耗。任何中断请求都能够将 CPU 唤醒。

Sleep 模式: 这个模式断开了内部电源。因此在这个模式下 CPU & 内部的逻辑单元都没有电源消耗, 除了一个 wake-up 逻辑单元。激活 sleep 模式需要 2 个独立的电源。一个给 wake-up 逻辑模块提供电源, 另外一个给内部逻辑包括 CPU 提供电源, 并且其是对于 power on/off 可控的。在 Sleep 模式, 提供给内部逻辑 & CPU 的电源模块将被关闭, 而从 Sleep 模式唤醒可以通过 EINT[15:0] & RTC 中断来引发。

7.1 功能描述

7.1.1 时钟结构

如图 7-1 所示时钟结构模块图。主时钟源来自外部晶振 (XTIPLL) 或者是外部时钟 (EXTCLK)。时钟生成器包含了一个振荡器 (振荡放大器), 其连接外部晶振, 并且还有 2 个 PLL, 可以产生满足 s3c2440 所需的高频时钟。

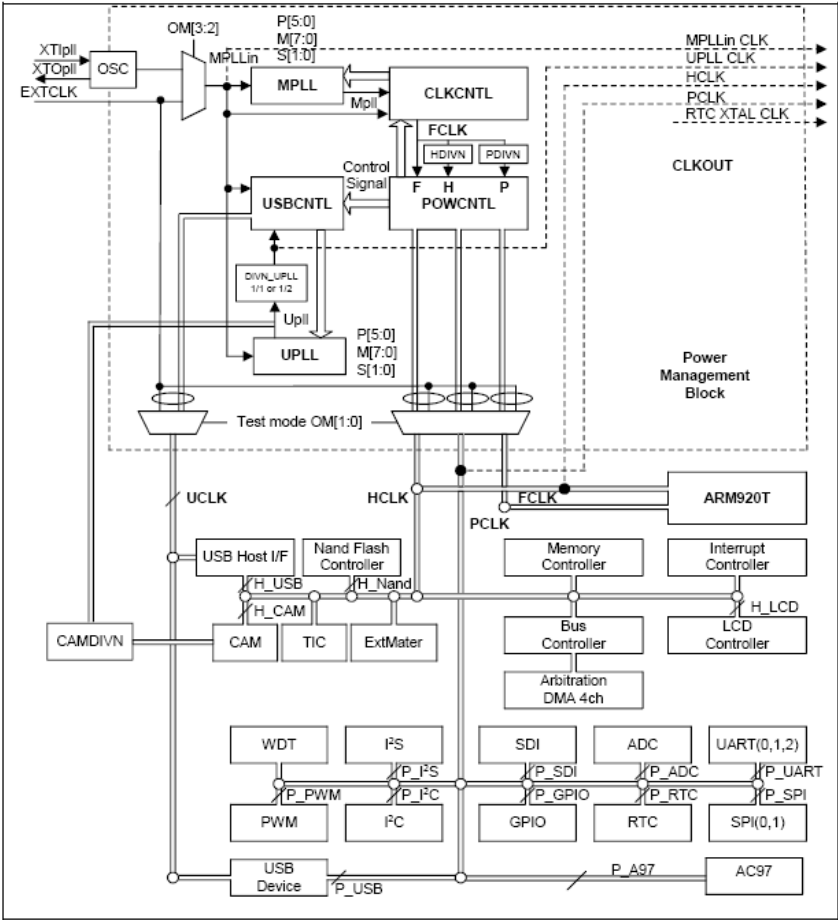


Figure 7-1. Clock Generator Block Diagram

7.1.2 时钟源的选择

如表 7-1 所示模式选择引脚(OM2 & OM3)组合之间的关系以及为 2440 选择时钟源，在 nRESET 上升沿，通过参考引脚 OM3 和 OM2，OM[3:2]的状态被锁存。

Mode OM[3:2]	MPLL State	UPLL State	Main Clock source	USB Clock Source
00	On	On	Crystal	Crystal
01	On	On	Crystal	EXTCLK
10	On	On	EXTCLK	Crystal
11	On	On	EXTCLK	EXTCLK

注意

- 1、虽然 MPLL 在 reset 以后就开始工作，但是 MPLL 输出（Mpll）没有作为系统时钟来使用，直到软件对 MPLLCON 寄存器写入有效设置。有效设置前，来自外部晶振或外部时钟源的时钟将直接被当作系统时钟来使用。就算是用户不想改变 MPLLCON 寄存器的默认值，用户应该再重新写入 MPLLCON 寄存器一次相同的值。
- 2、当 OM[1:0]=11 时，OM[3:2]被用于决定 test 模式。

7.1.3 锁相环

Phase Locked Loop (PLL)

位于时钟发生器中的 MPLL，作为一个集成电路，使得输出信号在频率和相位上同步于一个参考输入信号。应用上其包括如图 7-2 所示以下基本模块：其产生于输入直流电压成比例的输出频率 VCO（电压控制振荡器）；通过 P 值来对输入频率（Fin）来进行分频的分频器 P；通过 m 值来对 VCO 的输出频率进行分频的分频器 M，其输出频率被输入到一个相位频率侦测器（PFD）；通过 s 值来对 VCO 的输出频率进行分频的分频器 S，其分频器 S 的输出是 Mpll(从 Mpll 模块的输出频率)；鉴相器；电荷泵；环路滤波器。

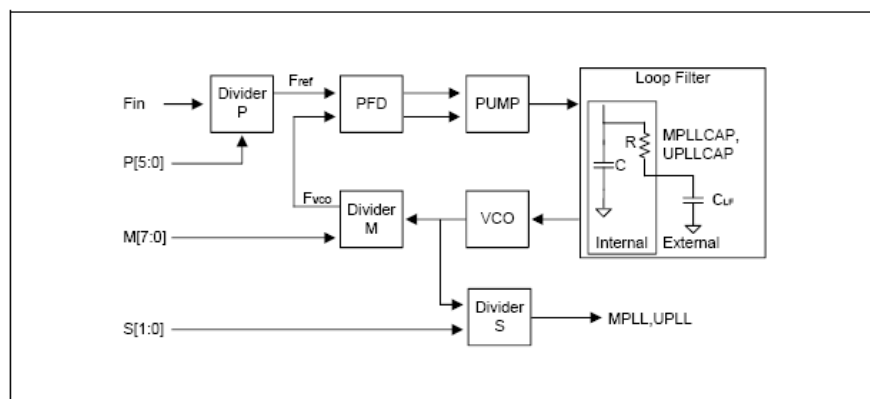


Figure 7-2. PLL (Phase-Locked Loop) Block Diagram

输出时钟频率 Mpll 相对于参考输入时钟频率 Fin 如以下公式所示：

$$M_{pll} = (2 * m * F_{in}) / (p * 2^8)$$

$$m = M (\text{分频器 M 的值}) + 8, \quad p = P (\text{分频器 P 的值}) + 2$$

Upll 时钟发生器的内部也和 Mpll 类似

7.1.3.1 鉴相器

Phase Frequency Detector (PFD)

PFD 的作用是监测 Fref 和 Fvco 的相位差，然后当检测到相位差时产生一个控制信号（跟踪型号）。Fref 意为参考频率，如图 7-2 所示。

7.1.3.2 电荷泵

Charge Pump (PUMP)

电荷泵通过一个由 VCO 驱动的外部滤波器将 PFD 的控制信号转变成一定电压的比例变化。

7.1.3.3 环路滤波器

Loop filter

PFD 为电荷泵生成的控制信号，通过每次 Fvco 与 Fref 比较，可能产生很大的偏移（纹波）。因此为了避免 VCO 过载，低通滤波采样以及滤除控制信号的高频分量。滤波器采用由一个电阻和电容组成的典型单极 RC 滤波器。

7.1.3.4 电压控制振荡器

Voltage Controlled Oscillator (VCO)

VCO 驱动的环路滤波器的输出电压,引起其振荡频率按均值变化线性的增加或减少。当 Fref 和 Fvco 在频率和相位上匹配，PFD 停止发送控制信号到电荷泵，其依次稳定给环路滤波器

的输入电压。VCO 频率保持稳定, PLL 保持固定为系统时钟。

7.1.4 通用条件和时钟发生器

PLL&时钟发生器需要如下硬件满足:

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: 1.值可变

2.FCLK 必须大于 200MHz

7.1.5 时钟控制逻辑

时钟控制逻辑决定哪个时钟源被使用, mppll 还是直接使用外部时钟(XTIpll or EXTCLK)。当 PLL 配置新频率值时, 时钟控制逻辑使 Fclk 无效直到 PLL 输出使用 PLL 锁定时间达到稳定。时钟控制逻辑单元在上电 reset 和 power-down 模式被唤醒时被激活。

7.1.5.1 上电重启 Power-on Reset

如图 7-4 所示在电源上电 reset 顺序中的时钟行为。晶振在几毫秒内开始振荡, 当在晶振时钟稳定后 nReset 被释放, PLL 开始参照默认的 PLL 配置运行。但是, PLL 通常在电源启动 reset 后都不稳定, 因此在软件没有配置 PLLCON 之前, Fin 直接代替 MPLL 作为 FCLK。注意, 用户应该通过软件给 PLLCON 寄存器重写一遍相同的值。

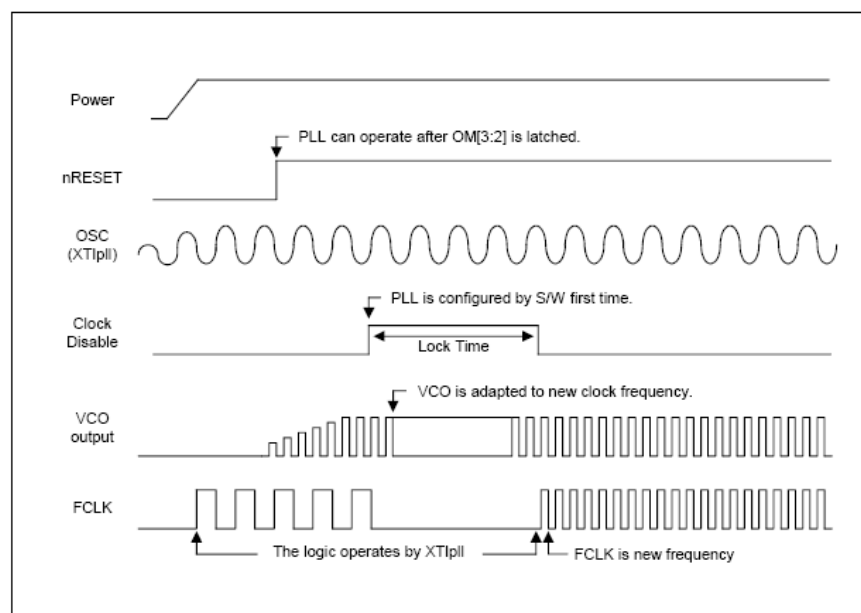


Figure 7-4. Power-On Reset Sequence (when the external clock source is a crystal oscillator)

仅在软件用新频率配置 PLL, PLL 重启一个锁定序列逼近一个新频率。FCLK 能在锁定时间后迅速被配置为 PLL 输出 (MPLL)。

7.1.5.2 正常模式下改变 PLL 设置 Change PLL Settings In Normal Operation Mode

在 Normal 模式下的 s3c2440 操作中, 用户可以改变频率通过写 P M S 三个分频器的值, 在 PLL 锁存时间会被自动的插入。在锁定时间, 时钟没有被提供给 s3c2440A 的内部时钟。图 7-5 所示的时序图。

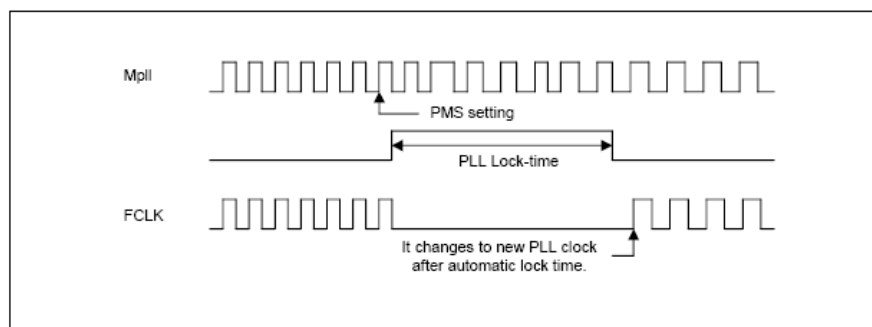


Figure 7-5. Changing Slow Clock by Setting PMS Value

7.1.6 USB 时钟控制

USB 主接口和 USB 从接口都需要 48Mhz 的时钟。在 S3C2440A 内，USB 知明需要 PLL(upll)产生 48Mhz 的时钟给 USB，UCLK 直到 PLL(UPLL)被配置才可以提供。

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

7.1.7 FCLK HCLK 和 PCLK

- FCLK 用于 ARM920T
- HCLK 用于 AHB 总线，例如用于 ARM920T,内存控制，中断控制，LCD 控制，DMA 以及 USB 主模块。
- PCLK 用于 APB 总线，用于外围设备如看门狗，IIS,I2C,PWM,MMC 接口，ADC UART,GPIO,RTC 以及 SPI。

2440 支持 FCLK,HCLK,PCLK 分频比的选择。比例由 CLKDIVN 控制寄存器中的 HDIVN,PDIVN 位确定。

HDIVN	PDIVN	HCLK3_HALF/ HCLK4_HALF	FLCK	HCLK	PCLK	分频比
0	0	-	FLCK	FLCK	FLCK	1:1:1
0	1	-	FLCK	FLCK	FLCK/2	1:1:2
1	0	-	FLCK	FLCK/2	FLCK/2	1:2:2
1	1	-	FLCK	FLCK/2	FLCK/4	1:2:4
3	0	0/0	FLCK	FLCK/3	FLCK/3	1:3:3
3	1	0/0	FLCK	FLCK/3	FLCK/6	1:3:6
3	0	1/0	FLCK	FLCK/6	FLCK/6	1:6:6
3	1	1/0	FLCK	FLCK/6	FLCK/12	1:6:12
2	0	0/0	FLCK	FLCK/4	FLCK/4	1:4:4
2	1	0/0	FLCK	FLCK/4	FLCK/8	1:4:8
2	0	0/1	FLCK	FLCK/8	FLCK/8	1:8:8
2	1	0/1	FLCK	FLCK/8	FLCK/16	1:8:16

在设置 PMS 3 个值后，需要设置 CLKDIVN 寄存器的值。CLKDIVN 的值设置在 PLL 锁定期后有效。这个值也对复位和改变电源管理模式也是可用的。

设置值在 1.5HCLK 后有效，1 个 HCLK 可以 CLKDIVN 寄存器的值从默认（1：1：1）到其他分频比（1：1：2，1：2：2，1：2：4）改变有效。

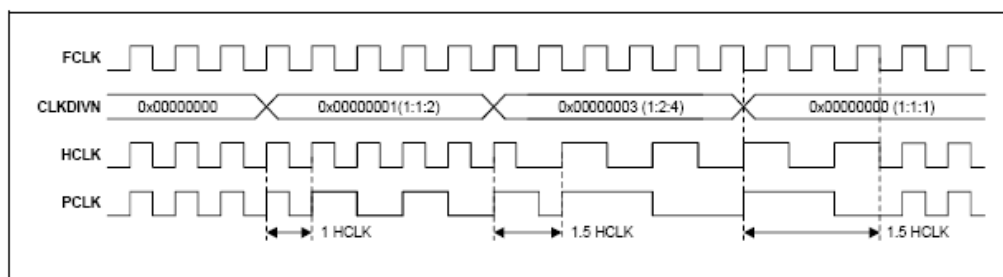


Figure 7-6. Example of internal clock change

注意

- 1、CLKDIVN 的值需要小心设置，不能超过 HCLK,PCLK 的极限值。
- 2、如果 HDIVN 不为 0，根据如下指令，CPU 总线模式从 Fast Bus Mode 变为 Asynchronous（异步总线模式）

MMU_setAsyncBusMode

mrc p15,0,r0,c0,0 ; 协处理器指令

orr r0, r0, #R1_nF : OR: R1_iA

mcr P15, 0, r0, c1, c0, 0

如果 HDIVN 不为 0，且 CPU 总线模式为 Fast Bus mode,CPU 的时钟为 HCLK.,这种方式可以用在将 CPU 频率降低，但是却又不改变 HCLK 和 PCLK.

7.2 电源管理

2440 的电源管理模块通过软件控制系统时钟来降低各个模块的电源消耗。这些方案与 PLL, 时钟控制逻辑(FCLK,HCLK,PCLK),以及唤醒信号相关。如图 7-7 所示系统时钟的结构图。

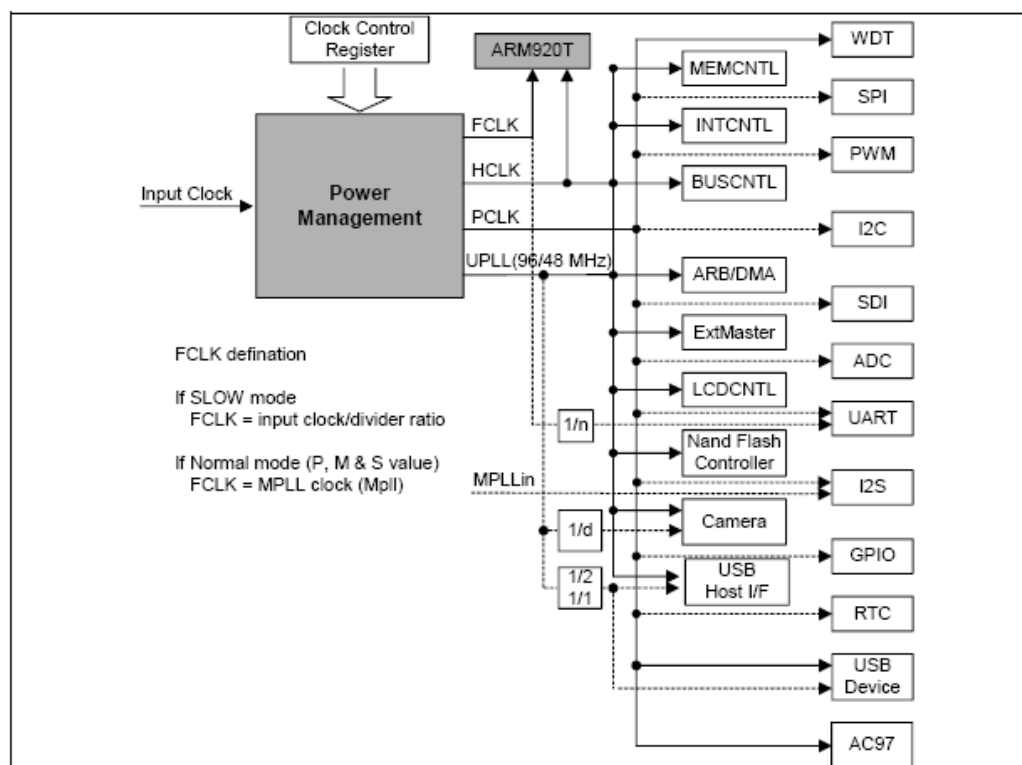


Figure 7-7. The Clock Distribution Block Diagram

表 7-3 CLKSLOW 和 CLKDIVN 寄存器对于减慢时钟的例子

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)	
000	EXTCLK or XTIpll /1	EXTCLK or XTIpll /1	EXTCLK or XTIpll /2	HCLK	HCLK/2	48MHz
001	EXTCLK or XTIpll /2	EXTCLK or XTIpll /2	EXTCLK or XTIpll /4	HCLK	HCLK/2	48MHz
010	EXTCLK or XTIpll /4	EXTCLK or XTIpll /4	EXTCLK or XTIpll /8	HCLK	HCLK/2	48MHz
011	EXTCLK or XTIpll /6	EXTCLK or XTIpll /6	EXTCLK or XTIpll /12	HCLK	HCLK/2	48MHz
100	EXTCLK or XTIpll /8	EXTCLK or XTIpll /8	EXTCLK or XTIpll /16	HCLK	HCLK/2	48MHz
101	EXTCLK or XTIpll /10	EXTCLK or XTIpll /10	EXTCLK or XTIpll /20	HCLK	HCLK/2	48MHz
110	EXTCLK or XTIpll /12	EXTCLK or XTIpll /12	EXTCLK or XTIpll /24	HCLK	HCLK/2	48MHz
111	EXTCLK or XTIpll /14	EXTCLK or XTIpll /14	EXTCLK or XTIpll /28	HCLK	HCLK/2	48MHz

在 SLOW 模式，PLL 将被关闭降低 PLL 的电源消耗。在 SLOW 模式下，PLL 被关闭，用户要从 SLOW 模式变成 NORMAL 模式，PLL 需要时钟稳定时间（PLL Lock time）。PLL 稳定时间可以通过内部逻辑借助锁定时间计数寄存器自动被插入。PLL 稳定时间在 PLL 开启后花费大概 300us 左右。在 PLL 锁定期间内，FCLK 成了 SLOW 时钟。

在 PLL on 态，用户可以使能 CLKSLOW 寄存器中的 SLOW 模式位来改变频率。SLOW 时钟在 SLOW 模式期间被生成，如图 7-9 所示时序图：

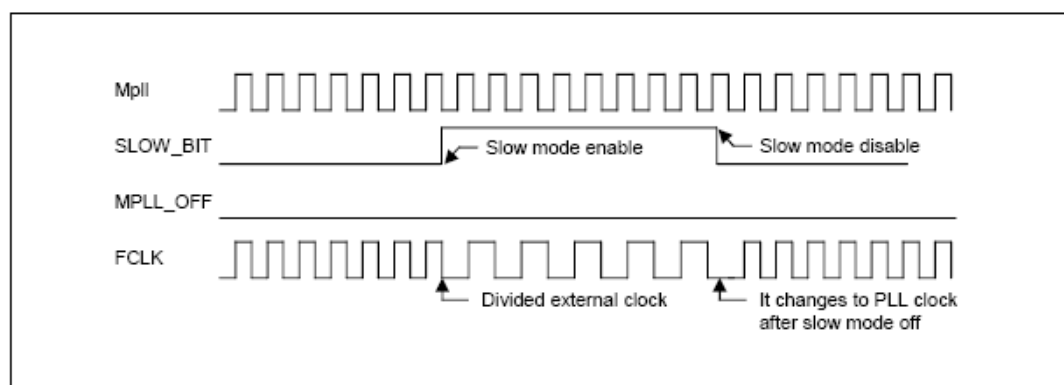


Figure 7-9. Issuing Exit_from_Slow_mode Command in PLL on State

如果用户在 PLL 锁定后通过使 CLKSLOW 寄存器中的 SLOW_BIT 位无效，从 SLOW 模式切换至 NORMAL 模式，频率在 SLOW 模式无效后立即改变，如图 7-10 所示时序图：

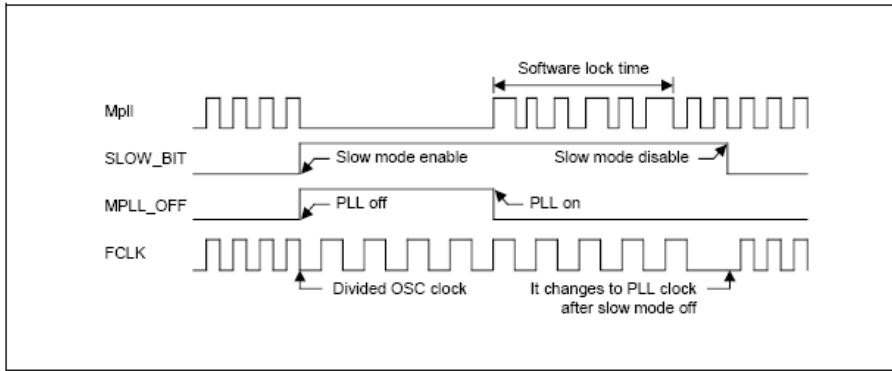


Figure 7-10. Issuing Exit_from_Slow_mode Command After Lock Time

如果用户通过使 CLKSLOW 寄存器中的 SLOW_BIT 和 MPLL_OFF 位同时无效, 从 SLOW 模式切换到 NORMAL 模式, 频率在 PLL 锁定时间后立即改变, 如图 7—13 所示时序图:

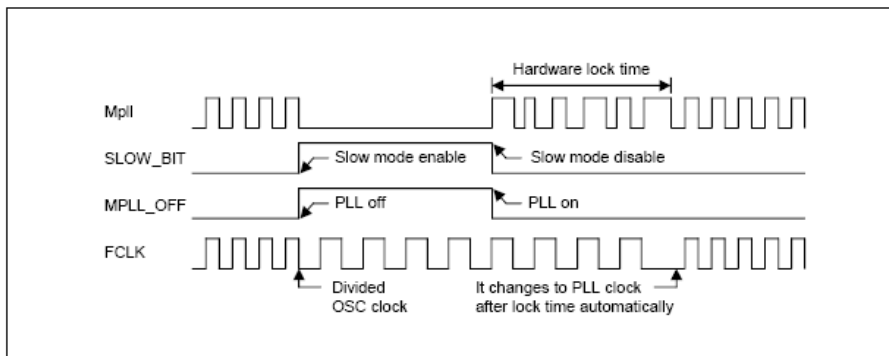


Figure 7-11. Issuing Exit_from_Slow_mode Command and the Instant PLL_on Command Simultaneously

7.2.4 SLEEP 模式

电源管理模块关闭内部电源, 因此, CPU 和内部逻辑模块都没有电源消耗, 除了此模式下的唤醒模块。激活 SLEEP 模式需要 2 个独立的电源, 其中一个给唤醒模块提供电源, 另一个给包括 CPU 的其他逻辑模块提供电源, 并且应该可以由 power on/off 控制。在 SLEEP 模式下, 给 CPU 和内部逻辑单元提供电源的第二个电源被关闭。只有唤醒模块是工作的, 睡眠模式被唤醒可以由 EINT[15:0]和定时器中 RTC(real time control)中断引起。

7.2.4.1 依据以下步骤进入到睡眠模式:

- 1、设置 GPIO 配置满足 SLEEP 模式。ConfigSleepGPIO();
- 2、在 INTMSK 寄存器设置屏蔽所有中断。rINTMSK=BIT_ALLMSK; 屏蔽所有中断;
- 3、正确设置唤醒源包括 RTC alarm (唤醒源的在 EINTMASK 中对应位不必被屏蔽,目的是使得 SRCPND 和 EINTPEND 的相应位 置 1。尽管一个唤醒源引发, 而 EINTMASK 相应位被屏蔽, 唤醒也将出现但 SRCPND 和 EINTPEND 的相应位不会被置 1)
`rEINTMASK = rEINTMASK & ~(1<<11); //SRCPND:EINT8_23 will be set by EINT11 after wake-up.`
- 4、设置 USB 挂起模式(MISCCR[13:12]=11b)
`rMISCCR |= (1<<12); //USB port0 = suspend`
`rMISCCR |= (1<<13); //USB port1 = suspend`
- 5、保存有意义的值到 GSTATUS[4:3]寄存器。这些寄存器在 SLEEP 期间被保留。
`rGSTATUS3=(U32)StartPointAfterSleepWakeUp; //复位后起始点的地址`
- 6、为数据总线上的上拉电阻配置 MISCCR[1:0]。如果有外部总线保持器, 如 74LVCH162245, 关闭上拉电阻; 如果没有, 开启上拉电阻。另外内存相关的引脚设置成

两态，一个是高阻，另一个是其他非活动状态。

```
*****rMISCCR=rMISCCR| (3<<0);
*****rMSLCON
(1<<11)|(0<<10)|(0<<9)|(1<<8)|(1<<7)|(1<<6)|(1<<5)|(1<<4)|(1<<3)|(1<<2)|(1<<1)|(1<<0);
查看 P284 寄存器可以看此设置
```

7、清除 LCDCON1.ENVID 位来使 LCD 停止工作。

```
*****rLCDCON1 =rLCDCON1 & 0x3ffe;
```

- 8、读取 rREFRESH 和 rCLKCON 寄存器以填充 TLB 位
- 9、设置 REFRESH[22]=1b，使能 SDRAM 进入自刷新模式。
- 10、等待直到 SDRAM 自刷新有效。

11、设置 MISCCR[19:17]=111b 使得 SDRAM 的信号(SCLK0,SCLK1,SCLKE)在睡眠模式期间被保护。

12、设置 CLKCON 寄存器的 SLEEP 位。

注意：当系统运行在 NAND boot 模式启动时，硬件引脚配置 EINT[23:21]必须，为 Sleep 模式唤醒的启动，设置为输入。

7.2.4.2 根据下面过程唤醒 SLEEP 模式

- 1、如果有唤醒源被触发，内部的复位信号就会动作。这和外部的 nReset 引脚触发非常相似。reset 复位持续时间由内部的 16 位计数器逻辑决定，reset 复位决断时间可以计算 $t_{RST}=(65535/XTAL_frequency)$
- 2、检测 GSTATUS2[2]来判断是否是由 sleep 模式唤醒引起的电源开启。
- 3、释放 SDRAM 的信号保护，通过设置 MISCCR[19:17]=000b
- 4、配置 SDRAM 内存控制器
- 5、等待直到 SDRAM 自刷新释放。大部分 SDRAM 需要所有 SDRAM 行的自刷新周期。
- 6、GSTATUS[3:4]的信息可用于用户自己目的，因为在 GSTATUS[3:4]中的值在睡眠模式下被保留。
- 7、-对 EINT[3:0]，检查 SRCPND 寄存器
-对 EINT[15:4]，查看 EINTPEND 寄存器而不是 SRCPND 寄存器。（尽管 EINTPEND 寄存器的一些位被置位，SRCPND 寄存器不会被置位）

表 7-4 在睡眠模式下引脚配置表

引脚条件		引脚配置指引	
GPIO 引脚	配置为输入	上拉有效	
	配置为输出	上拉无效输出低电平	
输入引脚，无内部上拉控制	外部设备不能一直驱动引脚电平	上拉有效，通过外部上拉电阻	
输出引脚，连接外部设备	外部设备电源关闭	输出低电平	
	外部设备电源开启	高或低（根据外部设备状况）	
数据总线	内存电源关闭		输出低电平
	内存电源开启	且外部缓存存在	上拉有效，控制信号先前值
		且无外部缓存	上拉有效，控制信号先前值

7.2.4.3 关于 VDDi&VDDiarm 的电源控制

在睡眠模式下，VDDi,VDDiarm,VDDMPLL 以及 VDDUPLL 会被关闭，其由 PWREN 引脚来控制，如 PWREN 信号被置位，VDDi 和 VDDiarm 由一个外部变压器供电。

当 $PWREN=0$ 时, $VDDi$ 和 $VDDiarm$ 被关闭

注意:

即使 $VDDi$, $VDDiarm$, $VDDMPLL$ 以及 $VDDUPLL$ 可能被关闭, 部分其他电源脚也可供电。
如图 7-12 所示

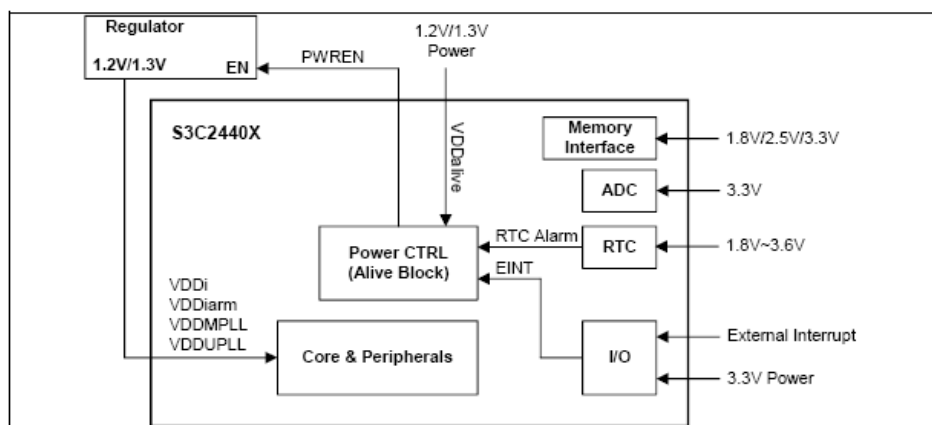


Figure 7-12. SLEEP Mode

在睡眠模式, 如果你不使用触摸屏, 4 个触摸接口必须悬空, 不能接地, 在睡眠模式下, XP, YP 都是高电平。

7.2.4.4 唤醒中断信号 EINT[15:0]

S3c2440 只有满足下面条件, 才能从 Sleep 模式中被唤醒

- a) $EINTn$ 引脚上的电平信号或跳变信号。
- b) GPIO 控制寄存器中配置 $EINTn$ 引脚为 EINT 外部中断功能。
- c) $nBATT_FLT$ 引脚必须为高电平, 这个在设置 GPIO 控制寄存器设置外部中断时候是非常重要的。

当唤醒后, 这些中断引脚 $EINTn$ 就不用作唤醒功能, 也就是说这些引脚又可以当作外部中断请求引脚。

7.2.4.5 Entering IDLE Mode

如果 $CLKCON[2]$ 设置为 1 进入空闲模式, 经过一段时间延时 2440 就进入空闲模式 (直到电源控制逻辑模块从 CPU 处收到一个 ACK 应答信号)。

7.2.4.6 PLL On/OFF

PLL 可以在 SLOW 模式下为降低电源功耗被关闭。如果 PLL 在其他任何模式下关闭, MCU 操作不能被保证。

当处理器在 SLOW 模式下且试图改变它的状态到其他态, PLL 为开启, 在 PLL 稳定后, $SLOW_BIT$ 应该被清除以转换到其他态。

7.2.4.7 PULL-up Resistors on the Data Bus and SLEEP Mode

在睡眠模式下, 数据总线($D[31:0]$ 或 $D[15:0]$)可以选择高阻或者输出低电平。

通过打开 pull-up 寄存器, 数据总线可以设置为高阻状态; 或通过关闭 pull-up 寄存器, 数据总线可以设置为输出低电平, 目的是在 SLEEP 模式下降低电源消。 $D[31:0]$ 引脚上拉电阻

能够由 GPIO 控制寄存器 (MISCCR) 控制。但是, 如果在数据总线上有一个外部总线保持器, 例如 74LVCH162245, 用户选择两态中的一个, 一个是关闭上拉, 输出低电平, 另一个是高阻, 上拉关闭, 这样可以消耗最少的电源。

7.2.4.8 Output Port State and SLEEP mode

输出引脚在 power off 模式下应该有一个适合逻辑电平, 使得电流消耗最小。如果输出引脚没有负载, 高电平比较好。如果输出是低电平, 电流会通过内部的寄生电阻被消耗。如果输出是高电平, 电流就不会被消耗。对于一个输出引脚, 如果输出态是高电平, 电流消耗可以被减少。

推荐把所有输出引脚都设为高电平态, 以减少在 SLEEP 模式下的电流消耗。

但是如果输出要低, 那么需要通过设置 DSC0,DSC1 来设置驱动电流, 举例如下:

```
*****rDSC0 = 0x7ffffff;  
*****rDSC0 &= ~(1<<31); //enable    调节内部的电阻阻值  
*****rDSC1 = (3<<28)|(3<<26)|(3<<24)|(3<<22)|(3<<20)|(3<<18);
```

7.2.4.9 Battery Fault Signal (nBATT_FLT)

nBATT_FLT 脚有 2 个功能, 如下

1、当 CPU 不是在 SLEEP 模式时 (大多数工作时候), nBATT_FLT 此脚将引起中断请求, 通过设置 BATT_FUNC(MISCCR[22:20])=10xb。其中断为低电平触发。

2、当 CPU 为 SLEEP 模式时, 设置此脚可以禁止从 SLEEP 模式的唤醒, 通过设置 BATT_FUNC(MISCCR[22:20])=11xb。因此, 任何一个唤醒源将被屏蔽如果 nBATT_FLT 被设置, 保护在低电池能力下系统故障。

7.3 时钟发生器及电源管理特殊寄存器

(Clock Generator & Power Management Special Register)

1. 锁定时间计数寄存器
LOCK TIME COUNT REGISTER (LOCKTIME): PLL lock time count register
2. 锁相环控制寄存器
PLL CONTROL REGISTER (MPLLCON & UPLLCON)
MPLLCON: MPLL configuration register
UPLLCON: UPLL configuration register
3. 时钟控制寄存器
CLOCK CONTROL REGISTER (CLKCON): Clock generator control register
4. 时钟减慢控制寄存器
CLOCK SLOW CONTROL REGISTER (CLKSLOW): Slow clock control register
5. 时钟分频控制寄存器
CLOCK DIVIDER CONTROL REGISTER (CLKDIVN): Clock divider control register
6. 摄像头时钟分频寄存器
CAMERA CLOCK DIVIDER REGISTER (CAMDIVN): Camera clock divider register

7.3.1 锁定时间计数寄存器

LOCK TIME COUNT REGISTER (LOCKTIME)

寄存器	地址	读写	描述	复位值
LOCKTIME PLL	0x4C000000	R/W	锁定时间计数寄存器	0xFFFFFFFF

LOCKTIME	位	描述	初始值
U_LTIME	[31:16]	UPLL 对于 UCLK 的锁定时间计数值. (U_LTIME: 300uS)	0xFFFF
M_LTIME	[15:0]	MPLL对于FCLK、HCLK、PCLK的锁定时间计数值 (M_LTIME: 300uS)	0xFFFF

7.3.2 锁相环控制寄存器

PLL CONTROL REGISTER (MPLLCON & UPLLCON)

MPLL Control Register

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

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

UPLL Control Register

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

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

锁相环值选择指引 (MPLLCON)

1. $F_{out} = 2 * m * F_{in} / (p * 2^s)$, $F_{vco} = m * F_{in} / p$, $m = MDIV + 8$, $p = PDIV + 2$, $s = SDIV$
2. $600MHz \leq F_{VCO} \leq 1.2GHz$
3. $200MHz \leq F_{CLKOUT} \leq 600MHz$
4. 不能设置P和M值为零, 因为P=000000, M=00000000 会引起PLL故障。
5. 合适的P、M值范围是: $1 \leq P \leq 62$, $1 \leq M \leq 248$

寄存器	地址	读写	描述	复位值
MPLLCON	0x4C000004	R/W	MPLL 配置寄存器	0x00096030
UPLLCON	0x4C000008	R/W	UPLL 配置寄存器	0x0004d030

PLLCON	位	描述	初始值
MDIV	[19:12]	主分频器控制	0x96 / 0x4d
PDIV	[9:4]	预分频器控制	0x03 / 0x03
SDIV	[1:0]	后分频器控制	0x0 / 0x0

注: 当你设置MPLL和UPLL值的时候, 需要先设置UPLL再设置MPLL的值

PLL值选择表 PLL VALUE SELECTION TABLE

一般较难找到一个合适的PLL值, 因此推荐参考以下PLL推荐值表:

输入频率	输出频率	MDIV	PDIV	SDIV
12.0000MHz	48.00 MHz 注	56 (0x38)	2	2
12.0000MHz	96.00 MHz 注	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
12.0000MHz	47.98 MHz 注	60 (0x3c)	4	2
16.9344MHz	95.96 MHz 注	60 (0x3c)	4	1
16.9344MHz	266.72 MHz	118 (0x76)	2	2
16.9344MHz	296.35 MHz	97 (0x61)	1	2
16.9344MHz	399.65 MHz	110 (0x6e)	3	1
16.9344MHz	530.61 MHz	86 (0x56)	1	1
16.9344MHz	533.43 MHz	118 (0x76)	1	1

注: 48MHz和 96MHz输出是用于UPLL。

7.3.3 时钟控制寄存器

CLOCK CONTROL REGISTER (CLKCON)

寄存器	地址	读写	描述	复位值
CLKCON	0x4C00000C	R/W	时钟发生器控制寄存器	0xFFFFF0

CLKCON	位	描述	初始值
AC97	[20]	控制进入 AC97 模块的 PCLK 0: 无效, 1: 有效	1
Camera	[19]	控制进入 Camera 模块的 HCLK 0: 无效, 1: 有效	1
SPI	[18]	控制进入 SPI 模块的 PCLK 0: 无效, 1: 有效	1
IIS	[17]	控制进入 IIS 模块的 PCLK 0: 无效, 1: 有效	1
IIC	[16]	控制进入 IIC 模块的 PCLK 0: 无效, 1: 有效	1
ADC	[15]	控制进入 AC97 模块的 PCLK 0: 无效, 1: 有效	1
RTC	[14]	控制进入 AC97 模块的 PCLK 0: 无效, 1: 有效	1
GPIO	[13]	控制进入 AC97 模块的 PCLK 0: 无效, 1: 有效	1
UART2	[12]	控制进入 UART2 模块的 PCLK 0: 无效, 1: 有效	1
UART1	[11]	控制进入 UART1 模块的 PCLK 0: 无效, 1: 有效	1
UART0	[10]	控制进入 UART0 模块的 PCLK 0: 无效, 1: 有效	1
SDI	[9]	控制进入 SDI 模块的 PCLK 0: 无效, 1: 有效	1
PWMTIMER	[8]	控制进入 PWMTIMER 模块的 PCLK 0: 无效, 1: 有效	1
USB Device	[7]	控制进入 USB Device 模块的 PCLK 0: 无效, 1: 有效	1
USB Host	[6]	控制进入 USB Host 模块的 HCLK 0: 无效, 1: 有效	1
LCDC	[5]	控制进入 LCDC 模块的 HCLK 0: 无效, 1: 有效	1
NAND Flash Controller	[4]	控制进入 NAND Flash Controller 模块的 HCLK 0: 无效, 1: 有效	1
SLEEP	[3]	控制s3c2440 的睡眠模式 0: 无效, 1: 进入睡眠模式	0
IDLE BIT	[2]	进入空闲模式 0: 无效, 1: 进入空闲模式	0
Reserved	[1:0]	保留	0

7.3.4 时钟减慢控制寄存器

CLOCK SLOW CONTROL REGISTER (CLKSLOW)

寄存器	地址	读写	描述	复位值
CLKSLOW	0x4C000010	R/W	减慢时钟控制寄存器	0x00000004

CLKSLOW	位	描述	初始值
UCLK_ON	[7]	0: UCLK ON (UPLL 被打开并且自动插入UPLL锁定时间.) 1: UCLK OFF (UPLL 也被关闭.)	0
保留	[6]	保留	-
MPLL_OFF	[5]	0: Turn on PLL. 在PLL稳定时间（至少 300us）后，SLOW_BIT位可以被清零 1: Turn off PLL. 仅当SLOW_BIT是 1，PLL才被关闭	0
SLOW_BIT	[4]	0 : FCLK = Mpll (MPLL output) 1: SLOW 模式 -FCLK = input clock/(2xSLOW_VAL), 当SLOW_VAL>0 -FCLK = input clock, 当 SLOW_VAL=0. -Input clock = XTIpil or EXTCLK	0
保留	[3]	保留	-
SLOW_VAL	[2:0]	当SLOW_BIT位开启，该位表示减慢时钟的分频器值	0x4

7.3.5 时钟分频器控制寄存器

CLOCK DIVIDER CONTROL REGISTER (CLKDIVN)

寄存器	地址	读写	描述	复位值
CLKDIVN	0x4C000014	R/W	时钟分频器控制寄存器	0x00000000

CLKDIVN	位	描述	初始值
DIVN_UPLL	[3]	UCLK 选择寄存器(UCLK必须对USB提供 48MHz) 0: UCLK = UPLL clock 1: UCLK = UPLL clock / 2 设置为 0，当UPLL时钟被设置为 48Mhz。 设置为 1，当 UPLL 时钟被设置为 96Mhz。	0
HDIVN	[2:1]	00: HCLK = FCLK/1. 01: HCLK = FCLK/2. 10: HCLK = FCLK/4, 当 CAMDIVN[9] = 0. HCLK = FCLK/8, 当 CAMDIVN[9] = 1. 11: HCLK = FCLK/3, 当 CAMDIVN[8] = 0. HCLK = FCLK/6, 当 CAMDIVN[8] = 1.	00
PDIVN	[0]	0: PCLK 是和HCLK/1 相同的时钟。 1: PCLK 是和HCLK/2 相同的时钟。	0

7.3.6 摄像头时钟分频器寄存器

CAMERA CLOCK DIVIDER REGISTER (CAMDIVN)

寄存器	地址	读写	描述	复位值
CAMDIVN	0x4C000018	R/W	摄像头时钟分频器寄存器	0x00000000

CAMDIVN	位	描述	初始值
DIVN_UPLL	[12]	0:DVS OFF (FCLK = MPLL clock) 1:DVS ON (FCLK = HCLK)	0
保留	[11]	保留	0
保留	[10]	保留	0
HCLK4_HALF	[9]	HDIVN分频比率改变位, 当CLKDIVN[2:1]=10b. 0: HCLK = FCLK/4 1: HCLK= FCLK/8 参考CLKDIV寄存器	0
HCLK3_HALF	[8]	HDIVN分频比率改变位, 当CLKDIVN[2:1]=11b. 0: HCLK = FCLK/3 1: HCLK= FCLK/6 参考CLKDIV寄存器	0
CAMCLK_SEL	[4]	0:用UPLL output作为CAMCLK (CAMCLK=UPLL output). 1:CAMCLK 用CAMCLK_DIV的值分频.	0
CAMCLK_DIV	[3:0]	CAMCLK 分频参数设置寄存器(0-15). Camera clock = UPLL / [(CAMCLK_DIV + 1)x2]. 这个位有效, 当CAMCLK_SEL=1.	0