硬件篇

目录：

第 0 章 AT91SAM7S32 芯片介绍

AT91SAM7S 系列 ARM 控制器是 ATMEL 在 2004 年推出的内置 FLASH，内置 RAM 的小 ARM7 器件，非常适用于成本敏感型应用，可以以 8bit MCU 的价格，提供 32bit MCU 的性能。AT91SAM7S32 具备 32K 的 FLASH，8K 的 RAM，极高的性价比适合其在低成本，大产量的消费类产品中应用。AT91SAM7S32 主要有如下优点：

>ARM7TDMI 内核，标准的 JTAG 接口是片内调试电路（ICE）；与 LPC2000 系列 ARM 不同，AT91SAM 系列 ARM7 采用的是 ARM7TDMI，而 LPC2000 系列是 ARM7TDMI-S，-S 内核是的 JTAG 调试速度不能太高，最高只能达到 1/6 系统时钟，实际测试情况是最高上到 4.8M；而 AT91SAM7S 的 JTAG 时钟可以和系统时钟一样高，实际测试情况是可以轻松上到 12M JTAG 时钟

>32K Flash，共 256 页，每页 128 字节。AT91SAM7S 系列的 FLASH 页（page）大小和 LPC2000 的 FLASH 的扇区（sector）大小也不一样。LPC2000 的各个扇区大小不一样，编号为 0-7 的 Sector 的大小是 4KB，编号为 7-21 的 Sector 的大小是 32KB，编号为 22-26 的 Sector 的大小是 4KB。在对 FLASH 编程的时候，LPC2000 会稍微快一点，但是在应用的时候，由于 ATMEL 的 page 比较小，使用起来就比较方便。

>FLASH 有 10,000 次写寿命，具备 FLASH 安全锁定位，可以防止非法读取，和 AVR 一样，可以通过整片擦除来取消该锁定位，也可以用 AT91SAM7S 的 ERASE 引脚来擦除 FLASH 内容和锁定位。既可以保护芯片 FLASH 内容，又可以保证 FLASH 的再次利用

>具备快速 FLASH 编程接口（FFI），适合量产（需 FFI 编程器支持）

>复位控制器（RSTC），提供上电复位和掉电检测。该控制器可以提供复位源信息，以告诉用户程序复位是何原因造成的，同时可以输出复位信号，用于控制外部设备。注意：AT91SAM7S 的复位输入默认是禁用的！即如果不对复位控制器进行设置，施加于复位引脚的复位信号是不被响应的

>时钟发生器，AT91SAM7S 的时钟发生器可能不是很好理解，在后文中，有详细的计算实例，建议参考阅读

>低功耗 RC 振荡器，3-20MHz 的片上振荡器和一个 PLL。注意 AT91SAM7S 的集成的标称 32K 的 RC 振荡器的误差较大，如果将其用来计时，请参考 ATMEL 的 AN，不然误差比较大。该 RC 振荡器并不象 AVR 那样有工厂标定过

>电源管理控制器（PMC），可以通过该控制器来优化电源，以降低功耗

>先进的中断控制器（AIC），可以单独屏蔽的、具有 8 个优先级的的向量式中断源；两个外部中断和一个快速中断

>调试单元（DBGU），其实就是用于调试的 2 线 UART，并可以通过程序来禁用 ICE，对于具备 USB 接口的 AT91SAM7S64/128/256，不光可以通过 DBGU 接口进行 ISP 操作，也可以通过 USB 口进行 ISP。注意，AT91SAM7S 的 ISP 功能相对 LPC2000 而言比较麻烦，需要等待 10 秒钟进行 boot 操作，而 LPC2000 只需要一个 IO 电平控制和一个复位信号即可进入 ISP 状态。不过在 AT91SAM7X 系列，ISP 功能有所改进，也可以通过 IO 电平和复位信号来控制进入 ISP 状态

>周期性间隔定时器（PIT），20 位可编程计数器，加上 12 位的间隔计数器，

>看门狗（WDT），12 位可编程计数器

>实时定时器（RTT），32 位计数器，具有报警功能，时钟来源是片内的 32K RC 时钟，RTT 通过软件可以用来设计实时时钟，但是精度有限，请参考 ATMEL 的 AN 进行优化设计

>并行 IO 控制器（PIOA），21 个可编程复用 IO，每个 IO 的电平变化，都可以引起中断，IO 可以独立编程为开漏输出，上拉电阻使能，或者同步输出

>9 个外设数据控制器通过（PDC）

>一个同步串行控制器（SSC），每个收发器都有独立的时钟信号和帧信号，支持 IIS 接口

>一个通用同步/异步收发器（USART），独立的波特率发生器，支持 IrDA

>主/从串行外设（SPI），8 位和 16 位数据宽度可选，具备 4 个片选，虽然 AT91SAM7S32/64/128/256 只有一个 SPI 接口，但是具备 4 个片选，可以连接的外设并不比具备两个 SPI 接口的 LPC2000 少。不过需要注意的是该 4 个片选使用的时候需要进行正确配置

>3 通道 16 位定时器/计数器（TC），可以用于定时，计数和 PWM 发生

>4 通道 16 位 PWM 控制器（PWMC）

>一个双线接口（TWI），仅支持主机模式，ATMEL 的 TWI 和 LPC2000 的 IIC 接口兼容

>一个 8 通道 10 位模数转换器，其中 4 通道和 IO 复用，另 4 通道为独立 ADC 通道

>IO 口兼容 5V 电压，并具备 4 个大电流 IO，单个 IO 可以达到 16mA 驱动能力，注意，所有 IO 的总电流不能超过 100mA，不然有可能损坏器件

>电源：IO 口线 3.3V 单独供电；FLASH 部分 3.3V 单独供电；内核 1.8V 供电，可以通过片内的 1.8V LDO 供电，也可以使用外部 1.8V 电源，如果使用片内 LDO 来产生 1.8V 内核电压，则 VDDIN 需要连接到 3.3V，如果 1.8V 内核电压有外部电源提供，则 VDDIN 需要连接到 GND。一般的设计是 VDDIN,VDDIO,VDDFLASH 一起连接到 3.3V ；VDDPLL,VDDCORE,VDDOUT 连接到一起，即 1.8V 有 VDDOUT 来提供，VDDOUT 的输出能力是 100mA。

第一章 AT91SAM7S 供电

电源系统为系统提供能源。随着微控制器片上系统的丰富和复杂，电源系统也变得复杂起来，一般的，器件的电源引脚也开始增多。必须正确理解各引脚的含义和作用，进行正确的电路连接，以保证器件工作在最合适的状态。

对AT91SAM7S32而言，有6种类型的电源输入引脚和一个内置的1.8V 稳压器：

1，VDDIN：内置稳压器的电源输入端。输入的电压范围为3.0~3.6V，标称为3.3V。如果不使用内置的稳压器，应将此引脚接地。

2， VDDOUT:稳压器输出，1.8V

3， VDDIO:I/O 和USB 部分的电源。电压范围为3.0~3.6V，标称为3.3V。

4， VDDFLASH:为片上flash 存储器提供电源，这是flash 正确工作的先决条件。电压范围为3.0~3.6V，标称为3.3V。

5， VDDCORE:芯片逻辑部分的电源。电压范围从1.65~1.95V,典型值为1.8V。可以连接到VDDOUT引脚，并连接退耦电容。该电压是器件内核与flash正常工作的前提。

6， VDDPLL:PLL部分的电源，可以直接连接到VDDOUT。需要注意的是，以上信号并没有独立的GND，因此应保证GND 引脚与系统地平面的引线尽量短。

典型的供电模型如下：

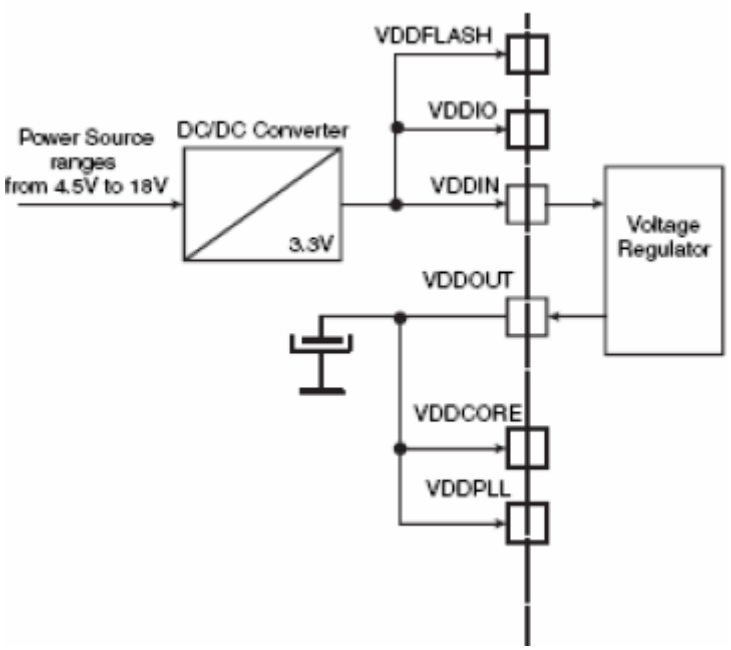

图 1-1

实际 AT91SAM7S32 的典型供电电路如下：

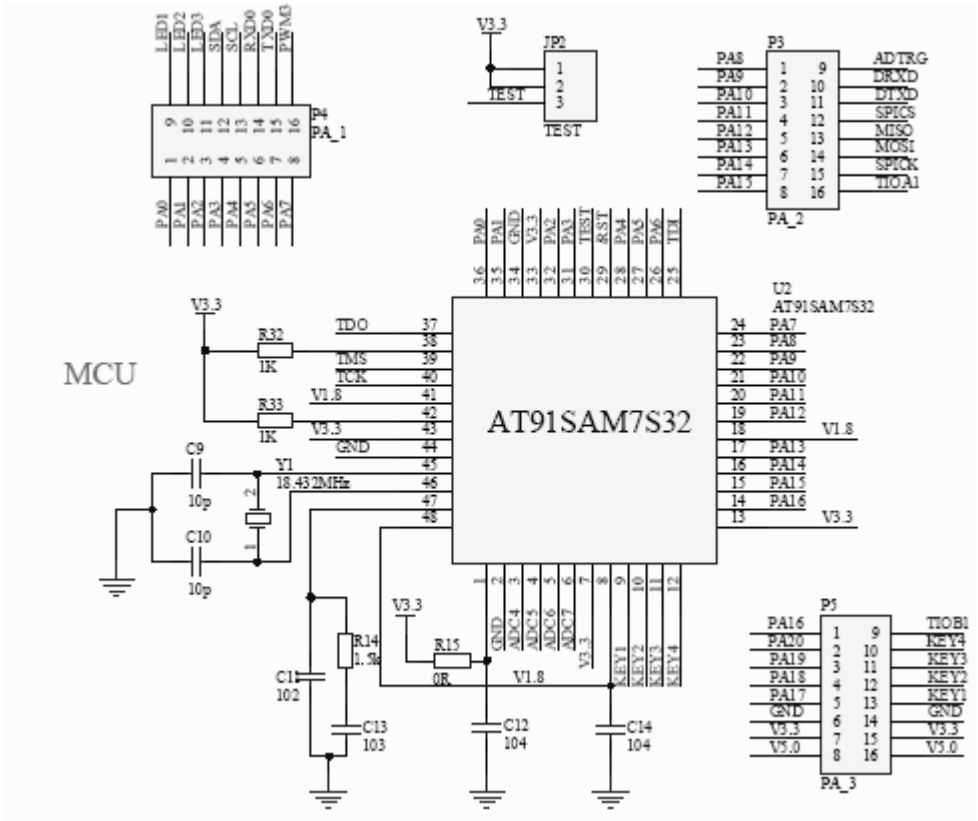

图 1-2

第二章 时钟

时钟是系统工作的节拍。S32 的时钟系统由下列组件构成：

1．SCLK，慢速时钟

2．MAINCLK，主时钟输出

3．PLLCK，分频器和PLL输出

其组成框图如下：

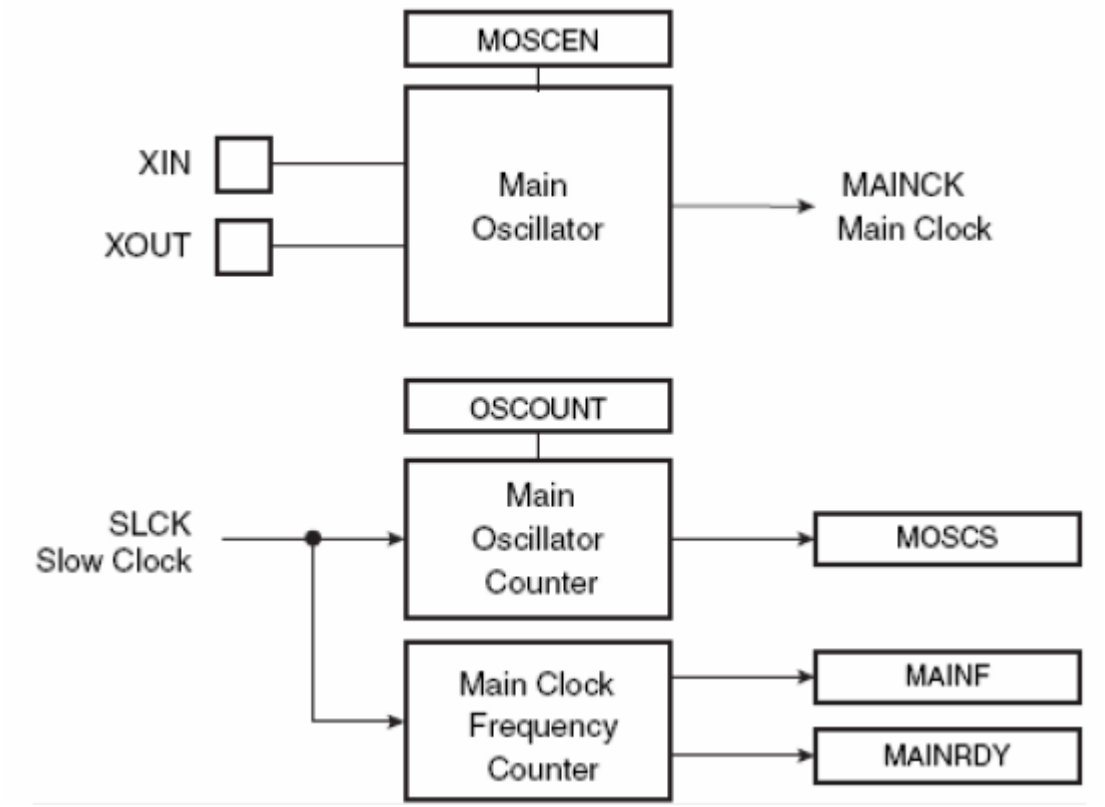

图 2-1

慢速时钟来源于片上的RC振荡器，其标称频率为32.758KHz，但是其精度比较差，具体的偏差可以参看器件手册。该时钟是器件上唯一恒定的时钟。

主时钟输出来源于外部晶体，其频率范围为3～20MHz，主振荡器的启动时间由晶体频率决定，频率高时间短。

为降低系统启动对电源需求，复位后主振荡器禁用而使用慢速时钟。通过清除主振荡器寄存器 (CKGR_MOR) 的 MOSCEN 位也可禁用主振荡器以减低功耗。当通过清除CKGR_MOR 中的MOSCEN 位将主振荡器禁用时，PMC_SR 中的 MOSCS 位自动清除，表示主时钟关闭。当使能主振荡器时，用户必须用对应于振荡器启动时间的值初始化主振荡器计数器。启动时间由与主振荡器连接的晶体频率确定。当 MOSCEN 位及OSCOUNT 写入 CKGR_MOR 后使能主振荡器，PMC_SR(状态寄存器) 中的 MOSCS
位清零且计数器按照慢时钟 8 分频的速度向下对 OSCOUNT 值开始计数。由于OSCOUNT 值为 8 位，因此最大启动时间为 62 ms。当计数器达到 0， MOSCS 位置位表示主时钟有效。因此在程序中使能了主时钟后，应等待该位置位。设置 PMC_IMR 中的 MOSCS 位可触发处理器中断。

主时钟频率计数器可以测量与主振荡器连接的石英晶体频率。通常，该值由系统设计者确定 ；但是它对引导程序正确配置时钟速度非常有用。 当主振荡器稳定后，即 MOSCS置位后，在慢时钟下一个上升沿时，主时钟频率计数器以主时 钟速度开始向上计数。然后在慢时钟第 16 个下降沿时，CKGR_MCFR(主时钟频率寄存器) 中 的MAINRDY 位置位且计数器停止计数。其值可从CKGR_MCFR 中的MAINF 域读出且给出16 个慢时钟中主时钟周期数，这样可通过计算得到与主振荡器连接的晶体频率。当不使用片上的振荡器而是使用外部输入时钟时，外部时钟必须由XIN 引脚输入，必须将主

OSC 寄存器 (CKGR_MOR) 的 OSCBYPASS位设置为 1 而 MOSCEN 位设置为 0，以保证外部时钟正常工作。

PLL 可以利用一个片外的低频时钟产生一个高频率的时钟源提供给内核及片上设备使用。PLL 内置一个输入分频器以增加结果时钟信号的精度。但是，当对分频器编程时，用户必须考虑 PLL 最小输入频率。

PLL的组成框图如下：

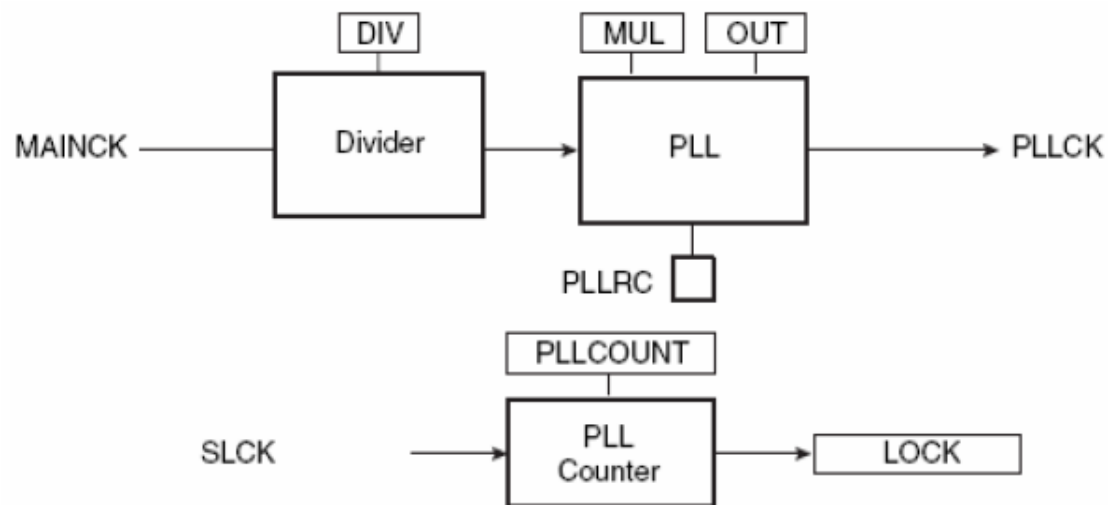

图 2-2

PLL需要通过PLLRC引脚链接到一个外部的二阶滤波器：

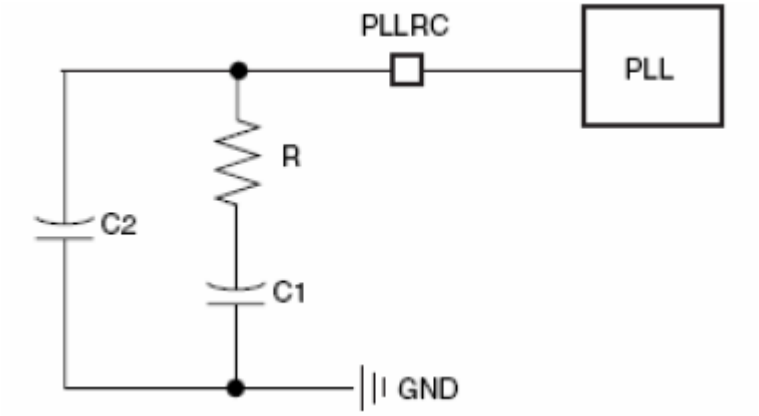

图 2-3

与 PLLRC 连接的 R、C1 与 C2 值由 PLL 输入频率、PLL 输出频率及相位容限确定。 必须在输出信号过量与启动时间间找到平衡。

PLL 的分频器可在 1 到 255 间设置，步长为 1。当分频器域(DIV) 设置为 0，相应分频器及 PLL 输出为连续的 0 信号。复位时，每个 DIV 域置为 0，因此相应的 PLL 输入时钟置为 0。 PLL 允许分频器输出相乘。PLL 时钟信号频率由各自源信号频率及参数 DIV与MUL 确定。源信号频率系数为 (MUL + 1)/DIV。当 MUL 写入 0，相应的 PLL 禁用，可节省其功耗。在 MUL 域写入大于 0 的值将重新使能 PLL。当 PLL 重新使能或它的某个参数改变，PMC_SR 中的 LOCK 位自动清零。 CKGR_PLLR 中PLLCOUNT 域值载入PLL 计数器。PLL 计数器开始以慢时钟速率开始递减直到其值为0。此时， LOCK位置位并能触发处理器中断。用户须在 PLLCOUNT 域载入所需慢时钟周期数来覆盖PLL过渡时间。过渡时间由 PLL滤波器确定。 PLL 初始状态及其目标频率可使用 Atmel 提供的专用工具进行计算。

向处理器提供不同频率的时钟，向片上外设提供不同时钟，可以控制处理器的功耗，这些都是由电源管理控制器（PMC）来完成的。电源管理控制器提供下列时钟：

• MCK，主机时钟，可编程，其频率由几百 Hz 到器件最高工作频率。可用在恒定运行的模块中，如 AIC 或存储控制器。

• 处理器时钟 (PCK)，提供给ARM 处理器的时钟，当处理器进入空闲模式时关闭。

• 外设时钟，提供给内置外设 (USART、SSC、SPI、TWI、TC、MCI 等) 并可独立控制。为减少产品中时钟名称数目，产品手册中将外设时钟称为 MCK。

• 可编程时钟输出可从时钟发生器提供的时钟中选择并在 PCKx 引脚上输出，以驱动其它的外围设备。

主机时钟控制器提供主机时钟 (MCK) 的选择与分频。MCK 为所有外设及存储控制器时钟。主机时钟从时钟发生器提供的时钟中选择。选择慢时钟则向整个器件提供一个慢时钟信号。选择主时钟会节省 PLL功耗。 主机时钟控制器由时钟选择器及预分频器组成。通过写PMC_MCKR (主机时钟寄存器)的CSS 域(时钟源选择)对主机时钟进行选择。预分频器分频因子为在 1 到 64 间 2 的幂次。PMC_MCKR 中的 PRES 域对预分频器编程。 每次 PMC_MCKR 写入定义一个新的主机时钟，PMC_SR 中的 MCKRDY 位清除。在主机时钟建立前它的值为 0。然后，MCKRDY位被置位并能触发处理器中断。该特性在当由高速时钟向低速时钟切换，用来通知处理器何时改变的频率变得稳定。

主机时钟控制器的组成框图如下：

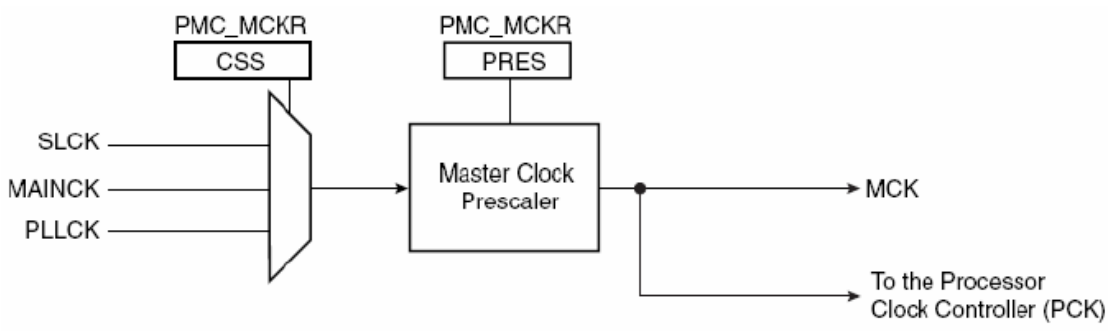

图 2-4

PMC 可使处理器时钟控制器 (PCK) 执行处理器空闲模式。通过写系统时钟使能寄存器(PMC_SCER)及系统时钟禁用寄存器(PMC_SCDR)可使能或禁用处理器时钟。该时钟状态可在系统时钟状态寄存器 (PMC_SCSR) 中读取。 处理器时钟 PCK 在复位后使能并可通过任意使能中断的中断触发而自动重新使能。通过禁用处理器时钟使处理器进入空闲模式，而处理器时钟可由任意使能的快速或普通中断重新使能，或由复位来使能。当处理器时钟禁用，当前指令在时钟停止前结束，但这不能防止数据由其它主机通过系统总线的传输（比如外设PDC）。

电源管理控制器通过外设时钟控制器来控制每个片上外设时钟。用户可通过对外设时钟使能 (PMC_PCER)及外设时钟禁用(PMC_PCDR)寄存器来独立使能或禁用片上外设的主机时钟。在外设时钟状态寄存器 (PMC_PCSR) 中可读取外设时钟状态。 当外设时钟禁用时，该时钟立即停止。外设时钟在复位后自动禁用以关闭不用的片上外设，降低功耗。 停止外设时，建议在禁用时钟前执行完最后一条指令，以避免数据出错或系统出错。外设时钟控制寄存器位

序号(PMC_PCER、PMC_PCDR 及 PMC_PCSR)为定义在产品级的外设标识符。通常，该序号对应于分配给外设的中断源序号。

PMC 控制外部引脚 PCKx 的 3 个输出信号。每个信号均可通过 PMC_PCKx 寄存器独立编程。 通过写 PMC_PCKx 中的 CSS 域，PCKx 可在慢时钟、 PLL 输出及主时钟间选择。每个输出信号可由 1 到 64 间 2 的幂次进行分频，具体系数在 PMC_PCKx 中的 PRES (预分频) 域写入。通过在 PCKx 中 PMC_SCER 或 PMC_SCDR 位写入 1 来使能或禁用输出信号。工作的可编程时钟状态由 PMC_SCSR(系统时钟状态寄存器) 中的 PCKx 位给出。此外，与 PCK 类似，PMC_SR 中的状态位表示可编程时钟是否已经稳定。由于可编程时钟控制器不会管理当切换时钟时出现的脉冲，强烈建议在配置变化前将可编程时钟禁用并在变化后重新使能。

下面由一段实用的程序来分析时钟系统的编程步骤：

```
void SAMInit(void)
{
AT91PS_PMC pPMC = AT91C_BASE_PMC;
// 设置flash的等待状态，这个在上面的说明中并没有提到。可以参考下数据手册。
// 由于ARM核心处理器速度较快，而flash本身的速度不会那么快
// 7S系列的flash能保证在ARM状态时可以30MHz的速度访问（单周期访问）
```

```
// 所以为了防止超速访问flash造成处理器取指异常，必须限制对flash的访问速度
//* Set Flash Waite sate
// Single Cycle Access at Up to 30 MHz, or 40
// if MCK = 47923200 I have 50 Cycle for 1 usecond ( flied
MC_FMR->FMCN
AT91C_BASE_MC->MC_FMR = ((AT91C_MC_FMCN)&(48<<16)) |
AT91C_MC_FWS_1FWS ;
//* Watchdog Disable
AT91C_BASE_WDTC->WDTC_WDMR= AT91C_WDTC_WDDIS;
//* Set MCK at 47 923 200
// 将处理器速度设置为47 923 200Hz，这个速度是USB 设备所需要的（48MHz）
// 1 Enabling the Main Oscillator:
// 使能主时钟
// SCK = 1/32768 = 30.51 uSecond
// 等待时间，为慢速时钟的倍数
// 注意其中的＋1，因为计数从6到0
// Start up time = 8 * (6＋1) / SCK = 56 * 30.51 = 1,46484375 ms
pPMC->PMC_MOR = ( (AT91C_CKGR_OSCOUNT) & (0x06 <<8)) |
AT91C_CKGR_MOSCEN ;
// Wait the startup time
```

```
// 等待主振荡器稳定
while(!(pPMC->PMC_SR & AT91C_PMC_MOSCS));
// 2 Checking the Main Oscillator Frequency (Optional)
// 3 Setting PLL and divider:
// 设置PLL与分频器
// PLL 输出频率18,432 MHz/ 5 * 26 = 95,8464MHz
// - div by 5 Fin = 3,6864 =(18,432 / 5)
// - Mul 25+1: Fout = 95,8464 =(3,6864 *26)
// for 96 MHz the erroe is 0.16%
// Field out NOT USED = 0
// PLLCOUNT pll startup time estimate at : 0.844 ms
// PLLCOUNT 28 = 0.000844 /(1/32768)
pPMC->PMC_PLLR = ((AT91C_CKGR_DIV & 0x05) |
(AT91C_CKGR_PLLCOUNT & (28<<8)) |
(AT91C_CKGR_MUL & (25<<16)));
// Wait the startup time
// 等待PLL 锁定
while(!(pPMC->PMC_SR & AT91C_PMC_LOCK));
while(!(pPMC->PMC_SR & AT91C_PMC_MCKRDY));
// 4. Selection of Master Clock and Processor Clock
// 将PLL输出2 分频作为主时钟与处理器时钟
// select the PLL clock divided by 2
```

```
pPMC->PMC_MCKR = AT91C_PMC_PRES_CLK_2 ;
while(!(pPMC->PMC_SR & AT91C_PMC_MCKRDY));
pPMC->PMC_MCKR |= AT91C_PMC_CSS_PLL_CLK ;
while(!(pPMC->PMC_SR & AT91C_PMC_MCKRDY));
// 5. ReMap
//AT91C_BASE_MC->MC_RCR=1;
F_CPU=XTAL/CKGR_DIV*CKGR_MUL/PCK_DIV;
}
```

这样初始化后，处理器的时钟已经建立，系统可以开始高速运行。但是为了功耗的考虑，启动后几乎所有外设的时钟都是关闭的，也就是说外设都不能工作。如果要启用某外设，在对外设初始化的时候，首先就要打开外设的时钟。这就需要在电源管理控制器（PMC）中进行设置。可以通过对其中管理外设时钟的几个寄存器(PMC_PCER, PMC_PCDR, PMC_PCSR)中当前外设ID 对应的位操作来实现。外设的ID 如下表：

外设 ID	外设助记符	外设名称	外部中断
0	AIC	先进的中断控制器	FIQ
1	SYSIRQ(1)	系统中断	
2	PIOA	并行 I/O 控制器 A	
3	Reserved		
4	ADC(1)	模数转换器	
5	SPI	串行外设接口	
6	US	USART	
7	Reserved		
8	SSC	同步串行控制器	
9	TWI	两线接口	
10	PWMC	PWM 控制器	
11	Reserved		
12	TC0	定时器 / 计数器 0	
13	TC1	定时器 / 计数器 1	
14	TC2	定时器 / 计数器 2	
15 - 29	Reserved		
30	AIC	先进的中断控制器	IRQ0
31	保留		

图 2-5

由于功耗和处理器速度，外设启用的多少都有关系，因此，应在系统的性能和速度之间取得均衡，且不需要的设备就不要启用。

第三章 复位

系统的复位由复位控制器（RSTC）控制。基于上电复位单元的复位控制器 (RSTC) 处理系统的所有复位，而无需其它器件。它可以给出上一次复位源的信息。复位控制器可以独立地、或同时驱动外部复位和外设及处理器复位。掉电检测可以防止处理器进入不可预测的状态。RSTC 的组成如下图：

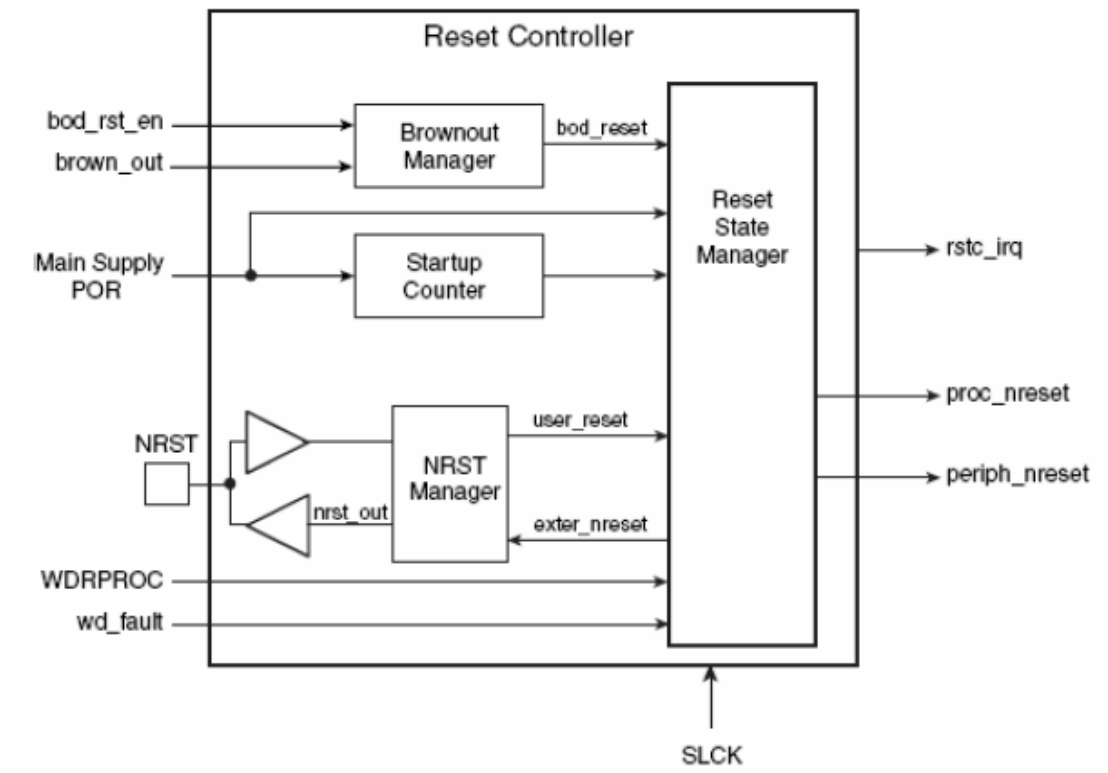

图3-1

复位控制器由 NRST 管理器、掉电检测管理器、启动计数器和复位状态管理器组成。它运行于慢速时钟，可以产生如下复位信号：

• proc_nreset：处理器复位。它同时也复位看门狗定时器。

• periph_nreset：作用于所有的外设。

• nrst_out：驱动 NRST 引脚。这些复位信号由复位控制器或者基于

外部事件，或者基于软件行为产生。复位状态管理器控制 着这些信号的产生，并在需要激活 NRST 引脚的时候为 NRST管理器提供信号。 NRST 管理器控制 NRST 信号保持预先编好程的一段时间，从而控制外部器件的复位。

NRST 管理器对NRST 引脚的输入进行采样，并在复位状态管理器需要的时候将引脚电平拉低。 NRST 管理器的方框图如下：

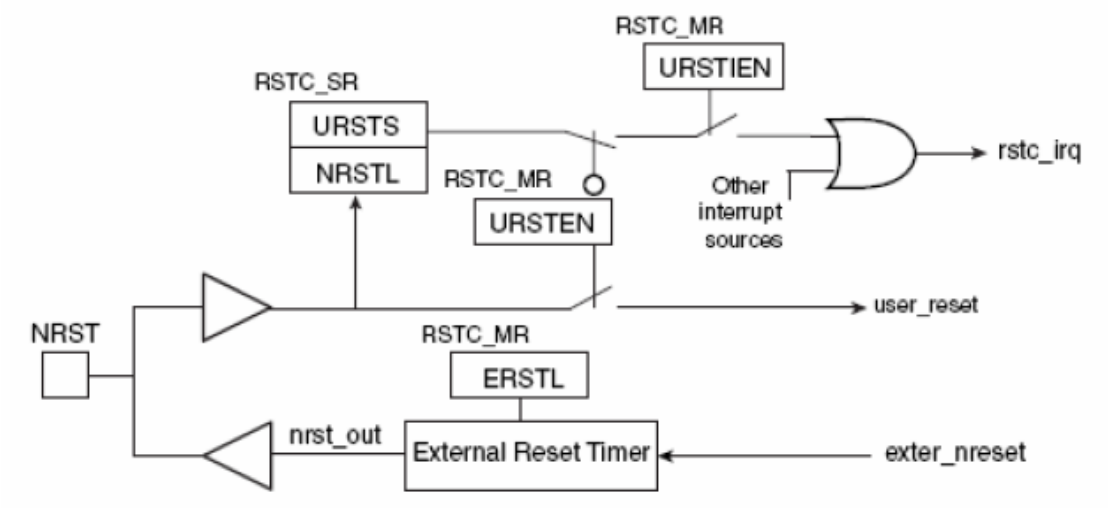

图3-2

NRST 管理器以低速时钟对NRST 引脚信号进行采样。当检测到信号为低时，用户复位的信号将报告给复位状态管理器。此外，通过编程还可以使NRST 管理器在NRST 为低时并不触发复位。这可以通过将RSTC_MR 寄存器的 URSTEN 位清零来实现。引脚 NRST 的电平可以在任何时候通过读取寄存器 RSTC_SR 的 NRSTL 来了解。一旦 NRST 上施加了有效信号，寄存器 RSTC_SR 的位 URSTS 置位。只有读取 RSTC_SR 之后这一位才清零。通过编程还可以使复位控制器产生中断，而不是复位。方法就是置位 RSTC_MR 寄存器的 URSTIEN。

复位状态管理器产生 ext_nreset信号来拉低 NRST 引脚。同时，“nrst_out” 信号被 NRST管理器拉低，并持续由 RSTC_MR 寄存器的 ERSTL 域控制的一段时间。这段名为 EXTERNAL_RESET_LENGTH 的时间持续2 的(ERSTL+1)次方个慢速时钟周期。所以其时间范围为60 μs 到 2 秒。 ERSTL 为 0 时 NRST 脉冲持续两个时钟周期。 这个特性使得复位控制器可以塑造 NRST 引脚的电平，从而保证 NRST 低电平持续时间满足连接到系统复位的外部器件的复位要求。

掉电检测可以防止处理器在电源下降到某个特定电平之后进入不可预测的状态。当VDDCORE 低于掉电检测门限时，掉电检测管理器通过激活 bod_reset 信号来请求掉电检测复位。程序员可以通过拉低 bod_rst_en 信号的方法来禁止掉电检测复位，即锁定 Flash 中对应的通用 NVM 位。然后掉电检测复位就不会再发生了。此时可以通过 寄存器RSTC_SR 的位BODSTS来 查看是否发生了掉电检测。 BODSTS 置位后只能通过读取 RSTC_SR 来清除。若寄存器 RSTC_MR 的 BODIEN 置位，则 BODSTS 可以触发中断。 芯片出厂时掉电检测复位是禁止的。

复位状态管理器处理不同的复位源，并产生内部复位信号。它通过状态寄存器 RSTC_SR的 RSTTYP 域来报告复位状态。处理器复位释放后 RSTTYP 即得以更新。RSTTYP 有如下类别：

1． VDDCORE 上电后，主电源 POR单元输出被运行于慢速时钟的启动计数器所过滤。这个计数器的目的是保证启动芯片之前慢速

时钟振荡器先稳定下来。经过启动时间并稳定下来之后，复位信号被释放， RSTC_SR 寄存器的 RSTTYP 得以更新，指明发生了上电复位。

2． 当 NRST 引脚电平为低，且寄存器 RSTC_MR 的 URSTEN 为 1 时即进入用户复位。NRST 输 入信号被同步到 SLCK 以保证系统的正确运行。 一旦检测到NRST 为低电平，系统即进入用户复位。同时还引发处理器复位和外设复位。 NRST电平拉高后，经过两个周期的重新同步时间和三个周期的处理器启动时间，处理器即退出 用户复位。一旦 NRST 为高，处理器时钟即重新使能。 处理器复位信号释放之后，状态寄存器的 RSTTYP 域即更新为 0x4，表示发生了用户复位。 NRST管理器保证 NRST 信号持续 EXTERNAL_RESET_LENGTH 个慢速时钟周期，正如域 ERSTL 编程的那样。然而，如果由于被外部电路拉低而使得 NRST 并没有在预定时间变高，内 部复位将保持有效直到 NRST 变高。

3． brown_out/bod_reset 信号有效时，复位状态管理器立即进入掉电检测复位。处理器复位、外设 复位和外部复位信号同时生效。brown_out/bod_reset 变高后，经过两个周期的重新同步时间和额外的 3 个慢速时钟周期，处理 器退出掉电检测复位。同时外部复位被触发。 处理器复位释放之后， RSTC_SR 寄存器的 RSTTYP 更新为 0x5，表明发生了掉电检测复位。

4． 复位控制器提供了几个命令来激活不同的复位信号。执行这些命令只需要将控制寄存器 RSTC_CR 的如下控制位写 1：

• PROCRST：置 1 将复位处理器和看门狗

• PERRST：置 1 将复位所有的外设，包括存储器系统，特别是重映像 (Remap)命令。外设复 位一般用于调试的目的。 • EXTRST：置1 将拉低NRST 引脚，并保持由模式寄存器RSTC_MR的ERSTL域定义的一段 时间。 一旦软件设置了上述的某一位或某几位，处理器即进入软件复位。所有这些命令可以单独执行， 也可以同时执行。软件复位持续 3 个慢速时钟周期。 一旦执行了控制寄存器写操作，内部各个复位信号立即生产。这可以通过主时钟 (MCK) 检测出 来：处理器退出软件复位时这些复位信号即被释放，亦即它们都同步于SLCK。 如果 EXTRST 置1， nrst_out 信号是否有效还取决于ERSTL 的设置。然而 NRST 的下降沿并 不会导致用户复位。 若仅有 PROCRST 置位，复位控制器将通过状态寄存器RSTC_SR 的 RSTTYP 域报告软件复 位。 RSTTYP 不报告其它的软件复位。

一旦发生了软件复位，状态寄存器 RSTC_SR 的位 SRCMP (Software ResetCommand in Progress-正在执行软件复位命令)即置位。处理器退出软件复位后它即被清零。SRCMP置位后 无法执行其它软件复位，对 RSTC_CR 的写操作也没有任何效果。

5． 看门狗错误发生时引发看门狗复位。这个状态将持续 3 个慢速时钟周期。 看门狗复位时，内部各个复位信号的产生取决于WDT_MR 寄存器的位 WDRPROC：

•若 WDRPROC 为 0，处理器复位和外设复位信号有效。NRST 引

脚也被拉低，持续时间与 ERSTL 有关。但是 NRST 引脚的低电平并不会引起用户复位。

• 若WDRPROC = 1，则只有处理器复位有效。 看门狗定时器由 proc_nreset 信号复位。如果 WDRSTEN 置位，看门狗溢出总是会引起处理器 复位，因此看门狗定时器总是在看门狗复位之后复位。在缺省条件下看门狗是使能的，而且溢 出时间设置为最长。 寄存器 WDT_MR 的WDRSTEN 清零时，看门狗错误(定时器溢出)对复位控制器没有任何影响。

软件可以根据上述的复位类别来进行相应的复位处理。

当多个复位源同时发生，复位源将按照如下优先级被处理（依次递减）：

- 上电复位
- 掉电检测复位
- 看门狗复位
- 软件复位
- 用户复位

特例则排列如下：

- 用户复位状态：

– 看门狗事件永远不会发生，因为看门狗定时器被 proc_nreset 信号复位了

– 软件复位不会发生，因为处理器复位已经被激活了

- 软件复位状态：

– 看门狗事件优先级高于当前状态

– NRST 没有效用

· 看门狗复位状态：

– 处理器复位已经激活，所以不可能进行软件复位的编程

– 不能进入用户复位

为了保证处理器的正确运行，复位信号必须被正确处理。需要注意的Tf10.0是默认的情况下，用户复位是禁止的，如果不重新使能，则通过手动的方式将不能使处理器复位。所以如果需要外部按键复位，需要在程序中使能用户复位，可以通过设置RSTC的相关寄存器来实现，可参考如下语句：

AT91C_BASE_RSTC->RSTC_RMR=0xA5000001;

第四章 DBGU/ISP/BOOT

DBGU 接口是 Debug Unit 的缩写，即调试单元，该调试单元由两线 UART 构成，该两线 UART 可以用于多种调试和跟踪目的。

调试单元（DBGU）的结构框图如下：

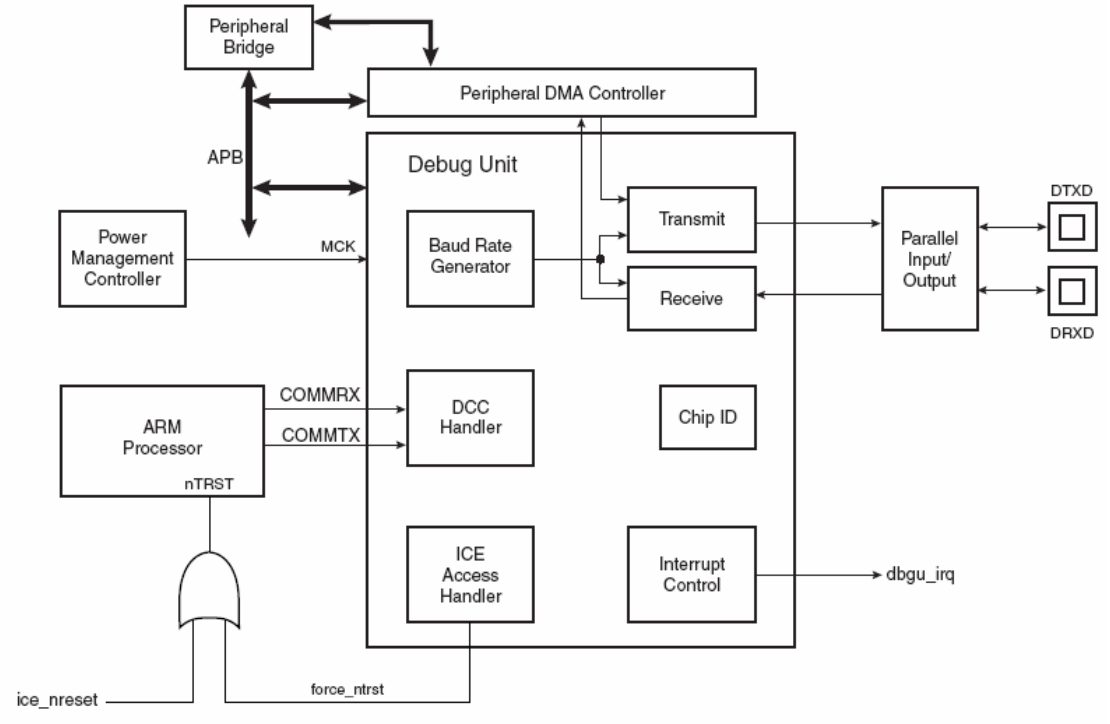

图 4-1

DRXD 是一个输入引脚，用于接收调试数据；DTXD 是一个输出引脚，用于输出调试数据。

调试单元（DBGU）的主要应用如下图：

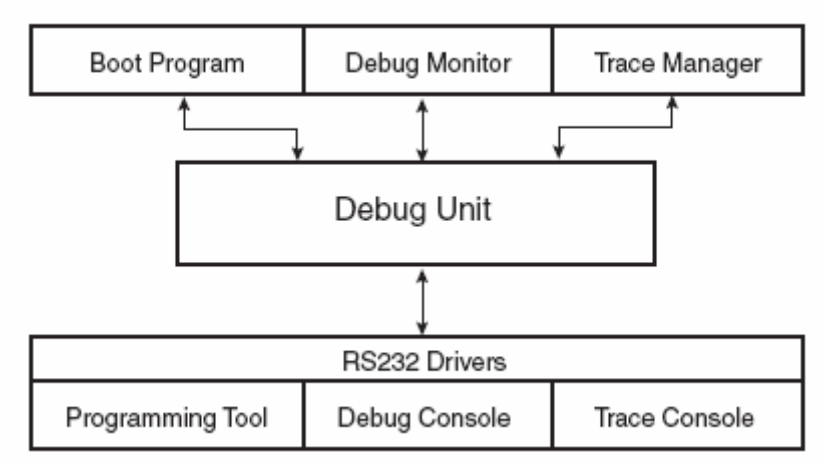

图 4-2

可以看到，DBGU 的主要应用有三：1，是用于进行 Boot 操作，即复制 Boot 代码并运行，这个时候就可以进行 ISP 操作，来编程用户 FLASH 文件；2，是用于调试，通过 DBGU 输入和输出调试信息；3，是用来跟踪片内信息。以上几个应用均需要 PC 端软件的配合。

一般情况下，我们直接通过 JTAG 进行调试，DBGU 接口用于调试的机会并不多见，当然这并不说明 DBGU 调试不如 JTAG 方便，相反，通过 DBGU 还有禁用 JTAG ICE 的使用。在没有 JTAG ICE 的情况下，或者 JTAG 调试器功能所限，比如需要烧写一个 Bin 文件的时候，又没有一个高档 JTAG ICE 来进行 FLASH 编程，这个时候就可以用到 DBGU 了的第一个应用了，即“ Boot Program ”，启动 Boot 代码后，通过 DBGU 接口来编程 FLASH，并运行用户代码。

ATMEL 的 ISP 操作稍微有点烦琐，首先需要进行 Boot 操作，就是通过控制 TST 的电平和复位信号，来进行 boot 操作。具体的步骤

是：

系统恢复程序（**System Recovery Procedure**）

1，目标板下电；

2，重新对目标板上电之前请确保**TST**，**PA0/PGMEN0**，**PA1/PGMEN1**，

PA2/PGMEN2 信号已经置位（参见上面的表格），由于**PA0/PGMEN0**，

PA1/PGMEN1，**PA2/PGMEN2** 内置上拉电阻，且复位后上拉电阻使能，而**TST**复位后为下拉电阻使能，故只要将**TST** 信号上拉即可，对于**AT91SAM7S32 EVB**，短接**JP1**的**1-2** 即可；

3，目标板上电，并请等待**10** 秒钟；

4，下电，将**TST** 信号悬空，对于**AT91SAM7S32 EVB**拔掉**JP1** 短路帽即可；

5，上电，此时目标板上的**SAM-BA Boot** 应用程序已经运行于**FLASH** 并等待来自于**DBGU** 的主连接（**PC**）；

6，打开SAM-BA软件，选择AT91SAM7S32 EK，选择DBGU方式连接，如果连接成功，将出现操作界面。

更详细的信息，可以参见本站翻译的 SAM-BA中文用户手册，该手册可以在我们的网站下载到。

第五章 JTAG 电路

ATMEL 的 JTAG 电路比较简单，并不需要 TRST 和 RTCK，所以仅 4 线即可：TMS,TDI,TDO,TCK，其中 TMS 和 TCK 和 5V 兼容，TDI 不兼容 5V。且 TMS,TDI,TCK 需要上拉。

典型的 ATMEL 的 JTAG 电路如下：

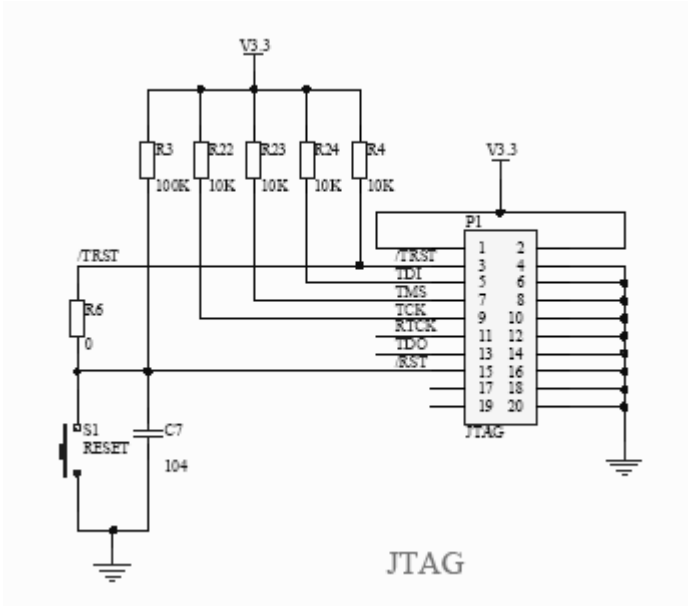

图 5-1

和 JTAG 电路相关的还有两个引脚，一个是测试引脚，即 TST，该引脚用来测试和控制进入快速编程模式，该引脚有一个片内下拉电阻，正常情况下可以悬空，如果将该 TST 引脚上拉到 VCC，则不能进行 JTAG 调试。

另外还有一个和 JTAG 相关的引脚，JTAGSEL 引脚，该引脚也有一个片内下拉电阻，如果拉高，进入 JTAG 边界扫描状态，也不能进行 JTAG 调试。

第六章 ADC 电路

AT91SAM7S 的片内 ADC 是基于连续寄存器（SAR）模型，片内通过一个 8 到 1 的模拟复用器来实现 8 通道的模数转换。ADC 输入范围是 0 到 ADVREF。

模数转换器框图：

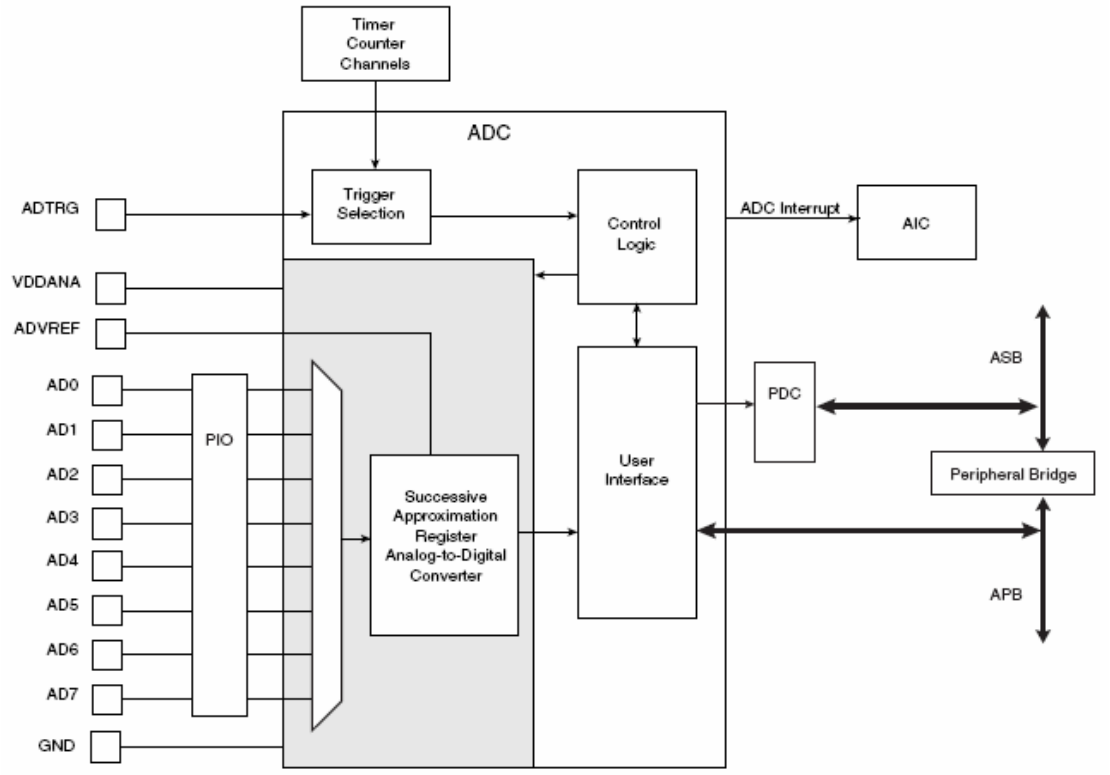

图 6-1

ADC 支持 8 和 10 位两种分辨率，可以通过软件出发、外部 ADTRG 触发引脚、内部触发定时器来启动 ADC。

可以通过配置 ADC 时钟，启动时间，采样保持时间来提高 ADC 的精度。

ADC 不受电源管理器管理。

ADC 有一个中断源，如果用使用到 ADC 中断信号，则需要事前配置中断控制器（AIC）。

注意：ADC 的 ADVref 的电压范围是 2.6V-VDDIN，范围比较窄，如果参考电压太低，ADC 工作将不正常。

ADC 的具体操作方法会在软件篇进行实例分析。

第七章 IIC/TWI

ATMEL 的 TWI 仅支持主机模式，由一根时钟线和一根传输速率达到 400Kb/s 的数据线组成，以直接位单位进行传输，理想用于连接 ATMEL 的所有 24 系列 EEPROM。

结构框图如下：

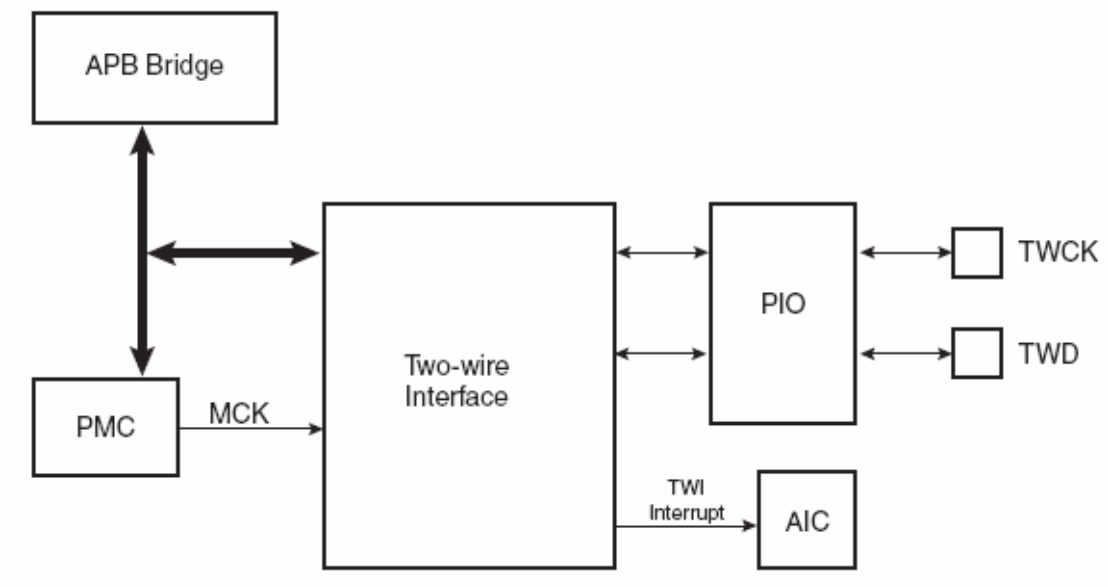

图 7-1

从框图可以看到，使用 TWI 需要配置一些寄存器。首先是配置 PMC，以是的 TWI 获得时钟支持，然后是配置 AIC，以使能 TWI 中

断，接下来还要配置 PIO，以启用 PIOA 作为 TWI 的第二功能。

具体的 TWI 应用，将会在软件篇以实例方式进行分析。

第八章 SPI

串行外设接口（SPI），是同步串行数据链路，可以工作于主机模式或者从机模式。若与别的控制连接，还可以实现控制器之间的高速通信。

SPI 的框图如下：

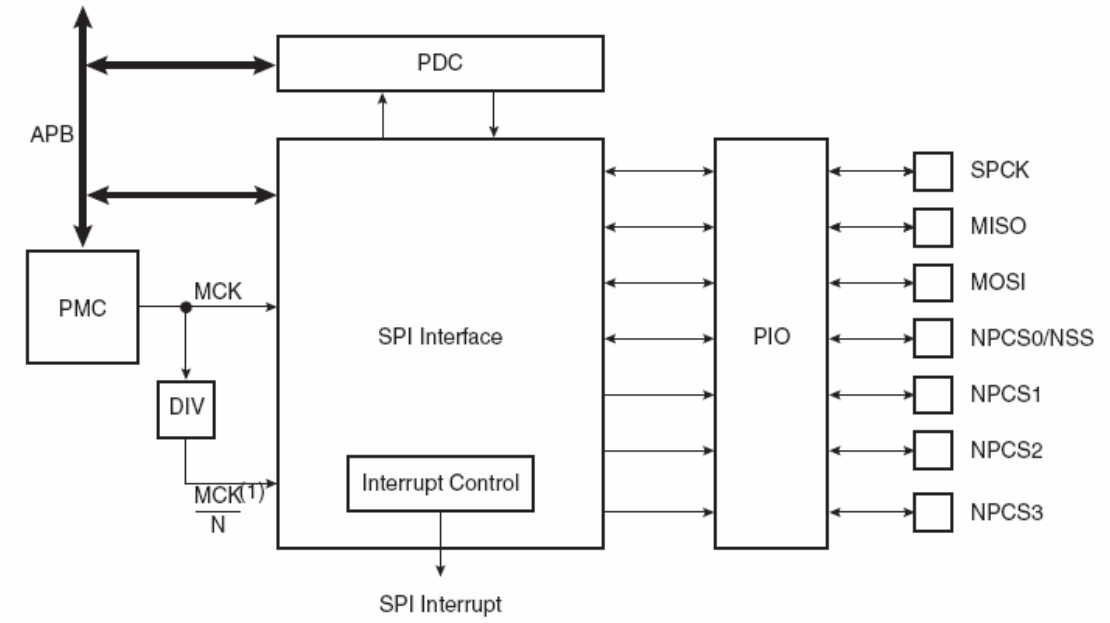

图 8-1

从图中可以看到，要使用 SPI，需要配置 PMC，AIC，PIO，使

得 SPI 获得时钟源，开启中断源，启用第二引脚功能。

SPI 的几个引脚的信号署名如下表：

引脚名称	引脚说明	类型	
		主机	从机
MISO	主入从出	输入	输出
MOSI	主出从入	输出	输入
SPCK	串行时钟	输出	输入
NPCS1-NPCS3	外设片选	输出	未用
NPCS0/NSS	外设片选 / 从机选择	输出	输入

表 8-1

NPCS0-NPCS3 使用起来需要注意，具体内容将通过软件篇的实例来分析。

第九章 USART

通用同步异步收发器（USART）提供一个全双工通用同步异步串行连接。

USART 的框图如下：

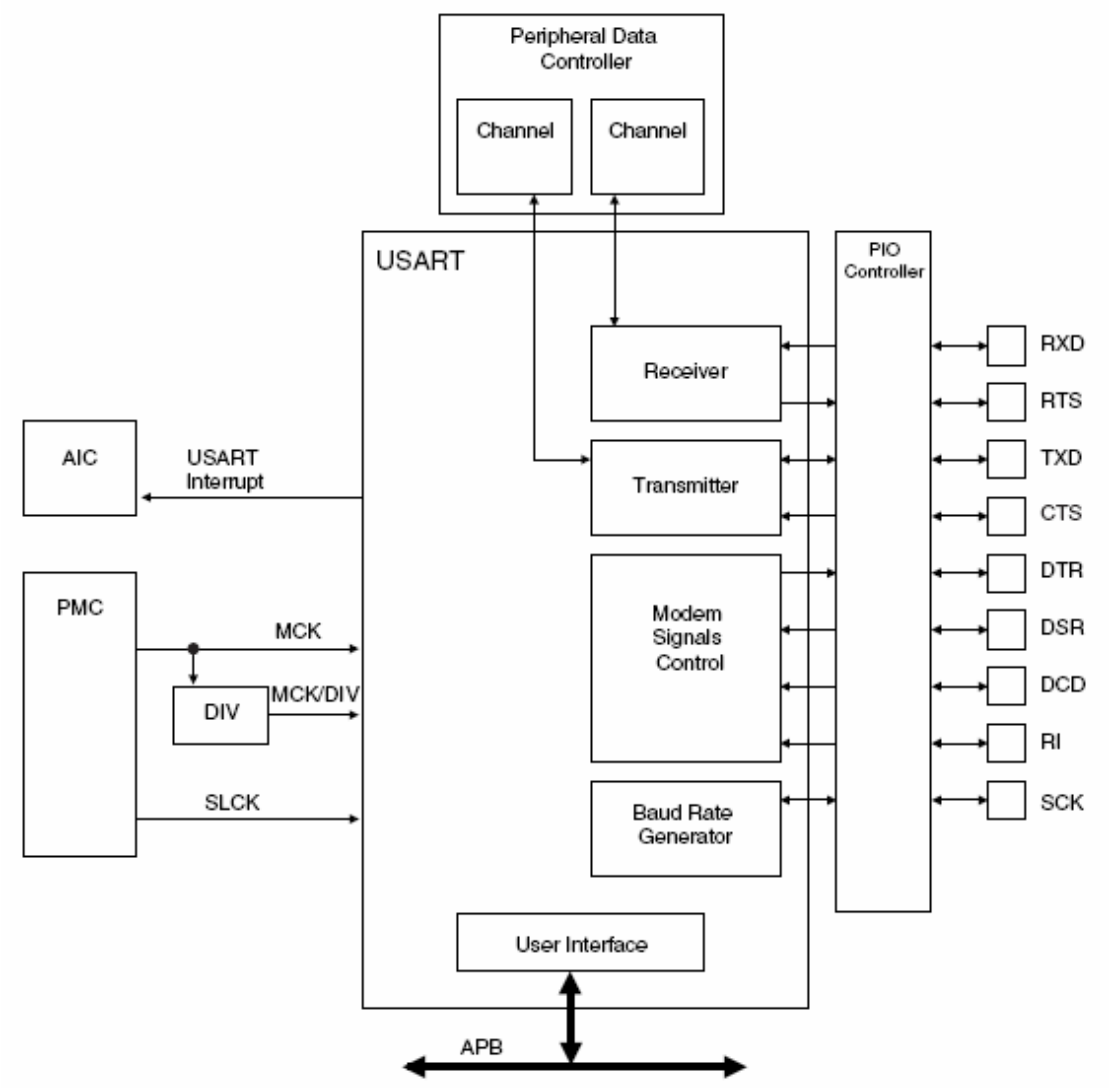

图 9-1

是一个完整的 Modem。同样，使用 USART 的话需要先设置 AIC 和 PMC。

USART 的应用框图如下：

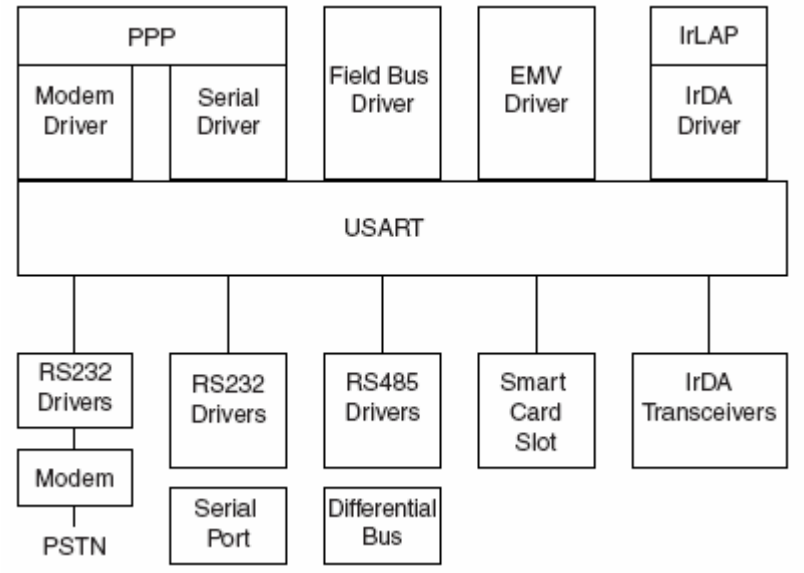

图 9-2

可以看到 AT91SAM7S 的 USART 可以用于 Modem，普通 2 线串口，RS485，智能卡，IrDA 红外传输，现场总线，PPP 通讯，EMV 驱动等等用途。

引脚说明如下：

名称	说明	类型	有效电平
SCK	串行时钟	I/O	
TXD	发送串行数据	I/O	
RXD	接收串行数据	输入	
RI	环指示器	输入	低
DSR	数据设置就绪	输入	低
DCD	数据载波检测	输入	低
DTR	数据中止就绪	输出	低
CTS	发送清除	输入	低
RTS	发送请求	输出	低

图 9-3

AT91SAM7S 的 USART 可以管理多类型的串行同步或异步通讯。

USART 主要支持如下通讯模式：

—5 到 9 位全双工异步串行通讯：

－高位或者低位在先

－1、1.5 或 2 位停止位

－奇校验、偶校验、标志、间隔或无

－接收器屏幕 8 或 16 倍重采样

－可选硬件握手

－可选调制解调器信号管理

－可选间断管理

－可选多点串行通信

—含驱动器控制信号的 RS485

—ISO7816，与智能卡连接的 T0 或 T1 协议

－NACK 处理，有复制与反复限制的错误计数器

—红外 IrDA 调制解调

—测试模式

－远程回环、本地回环、自动回应

对于 USART，还有很重要的一个部分是波特率发生器，波特率发生器的框图如下：

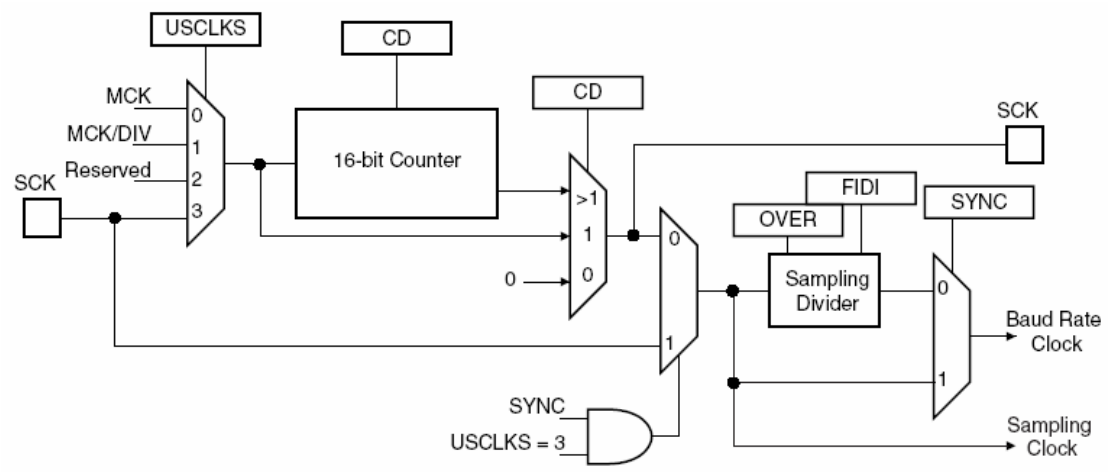

图 9-4

异步模式下的计算公式如下：

$$Baudrate = \frac{SelectedClock}{(8(2-Over)CD)}$$

误差计算如下：

$$Error = 1 - \left(\frac{ExpectedBaudRate}{ActualBaudRate}\right)$$

误差越大，误码率也就越高，建议误差不要超过 5%。

同步模式下的计算公式如下：

$$BaudRate = \frac{SelectedClock}{CD}$$

ISO7816 模式下的计算公式如下：

$$B = \frac{Di}{Fi} \times f$$

其中：

- B 为比特率
- Di 为比特率调整因子
- Fi 为时钟频率分频因子
- f 为 ISO7816 时钟频率 (Hz)

Di 是一个 4 位二进制值，称为 DI，见 Table 59。

具体各种模式下的收发程序将在软件篇里面讲到。

第十章　PWM

PWM 是脉宽调制控制器的缩写，PWM 宏单元独立控制几个通道。每个通道控制一个输出方波。用户可以设置输出波形的频率，占空比以及极性。

PWM 的框图下：

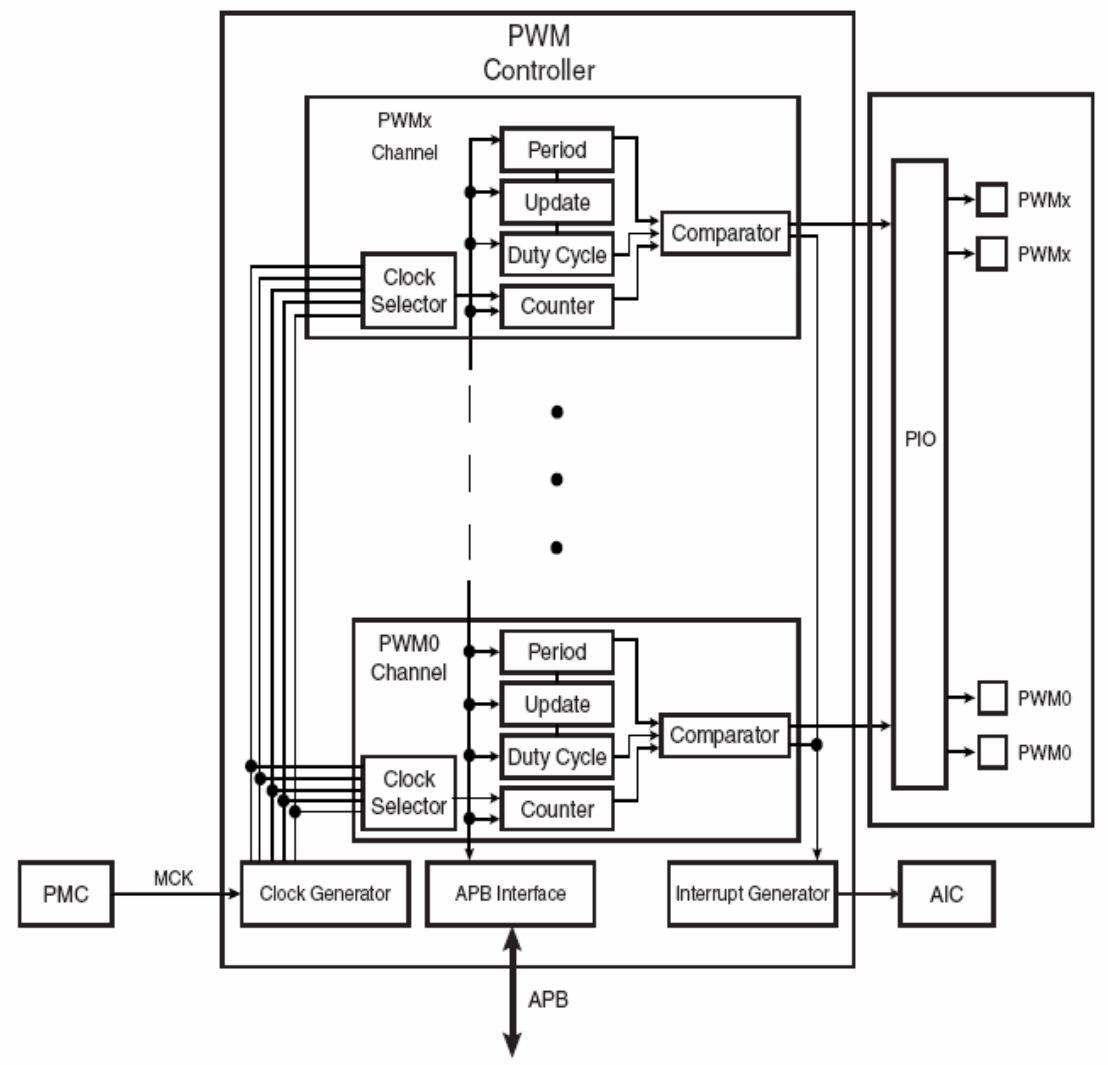

图 10-1

每个PWM发生器都可以独立选择PWM时钟源。同样，使用PWM需要设置 PIO，PMC 和 AIC。

PWM 的时钟发生器的框图如下：

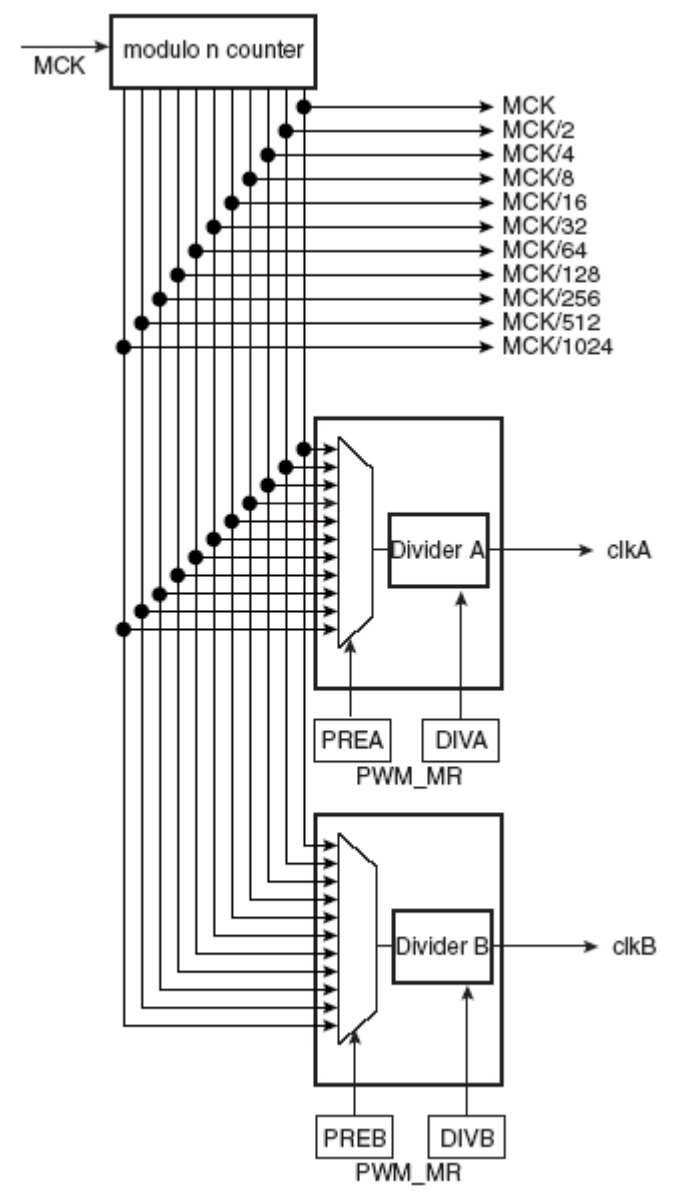

图 10-2

可以看到，PWM 时钟可以由三个模块产生，首先是模 n 计数器，提供 11 个时钟信号，从 MCK 到 MCK/1024；另外两个是相互独立的分频器 A 和分频器 B，都可以实现线形分频，即在 MCK 到 MCK/1024

的基础上再进行 1，1/2，…1/255 的分频。

PWM 通道的框图如下：

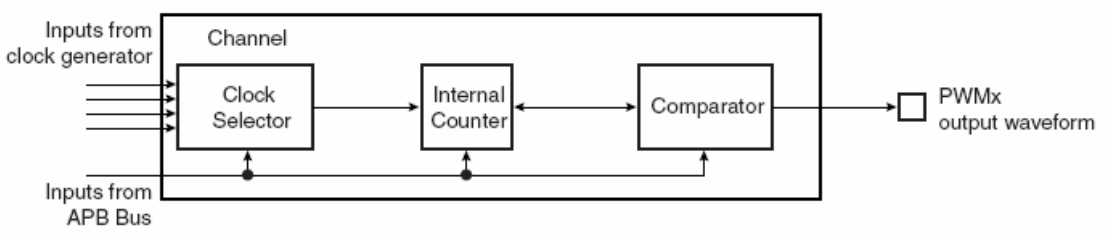

图 10-3

关于 PWM 的具体设置和初始化代码，会在软件篇详细介绍。

第 11 章 AIC

高级中断控制器（AIC）是具有 8 个优先级，独立可屏蔽的向量中断控制器，最多可以处理 32 个中断源。

AIC 的框图如下所示：

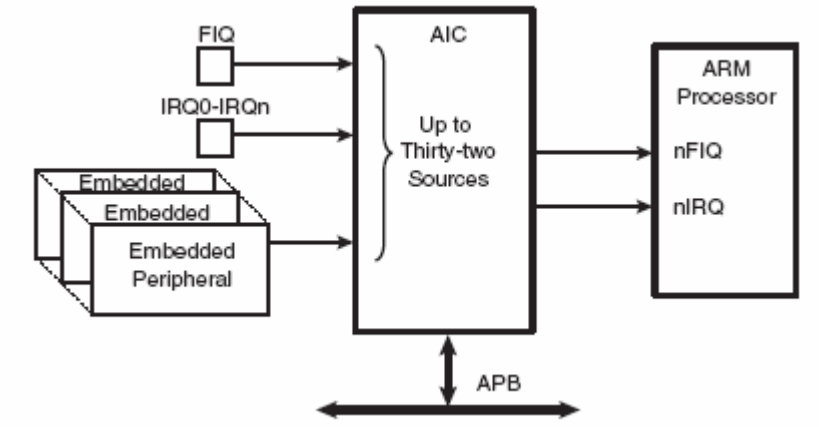

图 11-1

可以看到，AIC 的来源可以是 FIQ 快速中断请求和 IRQ 标准中断请求，FIQ 是 ATMEL 独有的，其他公司的小 ARM7 并没有引出该引脚。

AIC 应用框图如下：

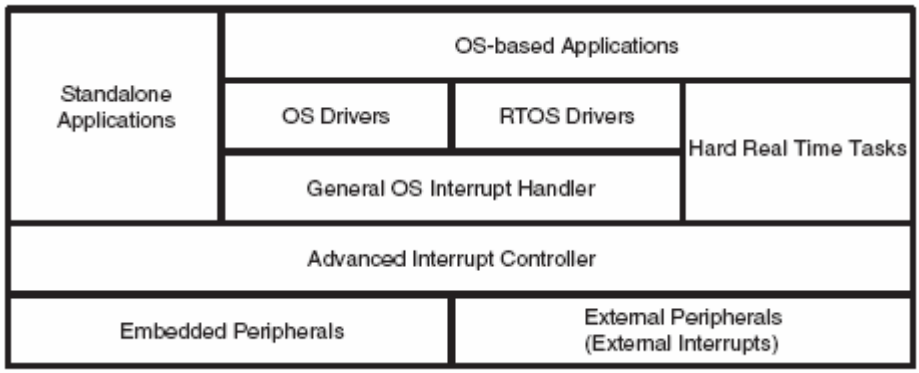

图 11-2

AIC 详细框图如下：

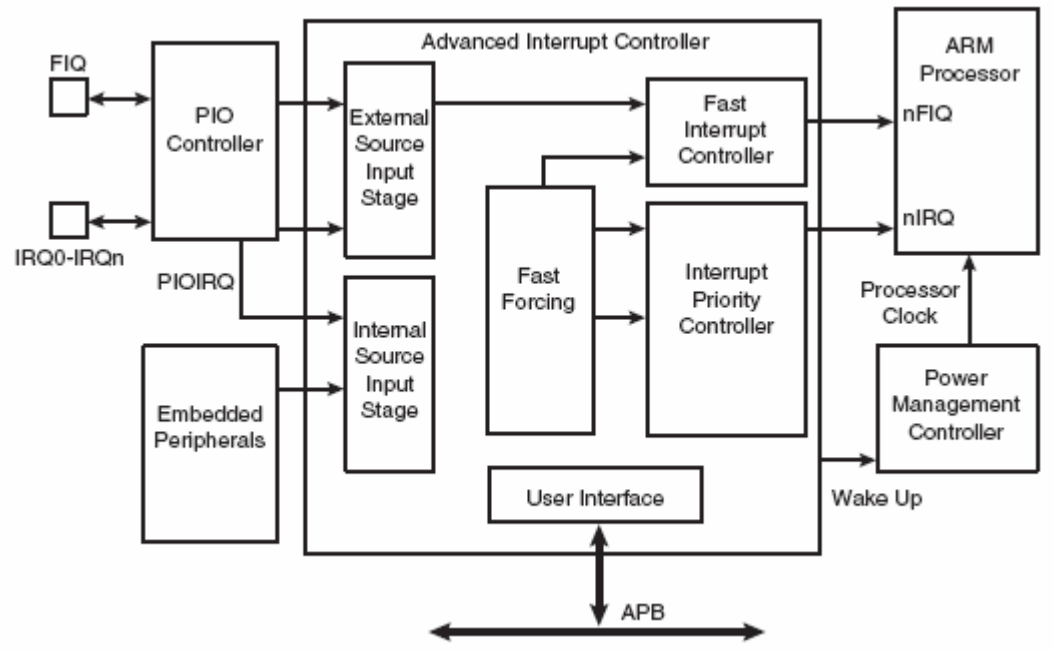

图 11-3

内部中断源输入流程：

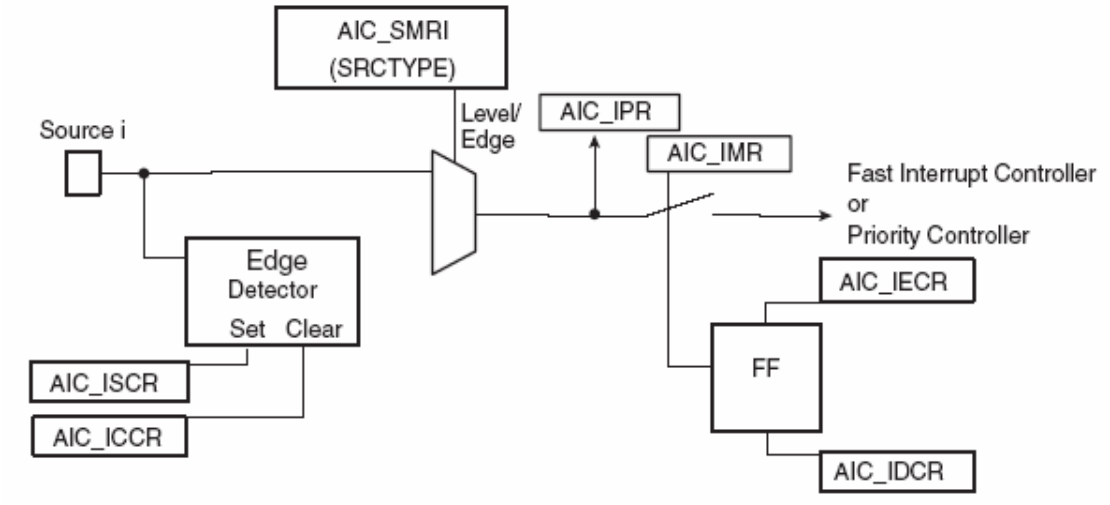

图 11-4

外部中断源输入流程：

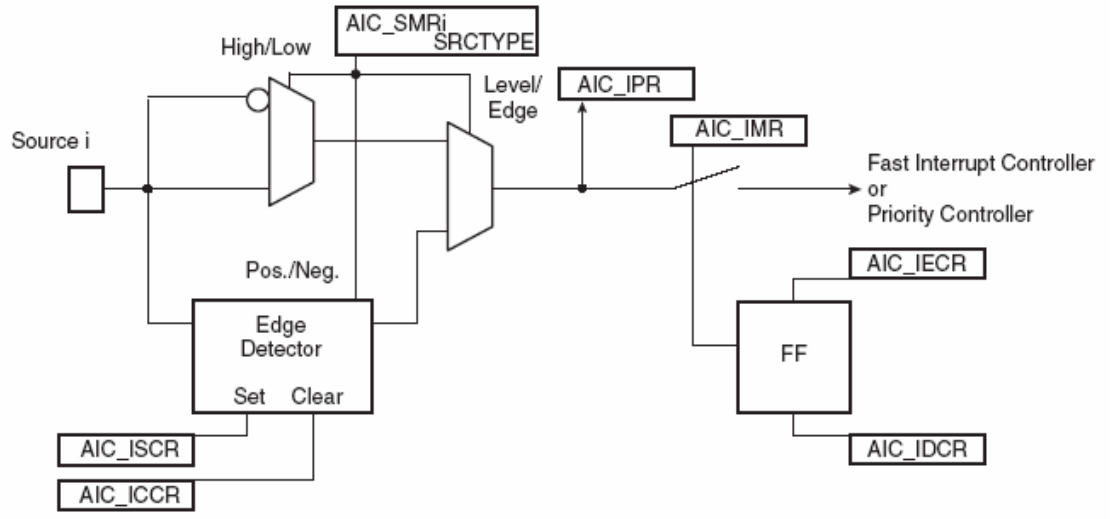

图 11-5

关于中断，会穿插在各个范例中进行分析和演示，具体内容参见软件篇。

第 12 章 PMC

电源管理控制器（PMC）通过控制系统及用户外设时钟来优化功耗，通过 PMC 可以启用和禁用大部分外设，以及 ARM 内核的处理器时钟输入。

电源管理控制器提供如下时钟：

— MCK，主机时钟，可编程，频率范围是从几十 Hz 到器件最高工作频率

— 处理器时钟（PCK）当处理器进入空闲模式时关闭

— 外设时钟，提供给片内外设时钟，比如 USART 模块，SPI 模块，TWI 模块等，并口独立控制

— 可编程时钟输出可从时钟发生器提供的时钟中选择，并在 PCKx 引脚上输出

主机时钟控制器：

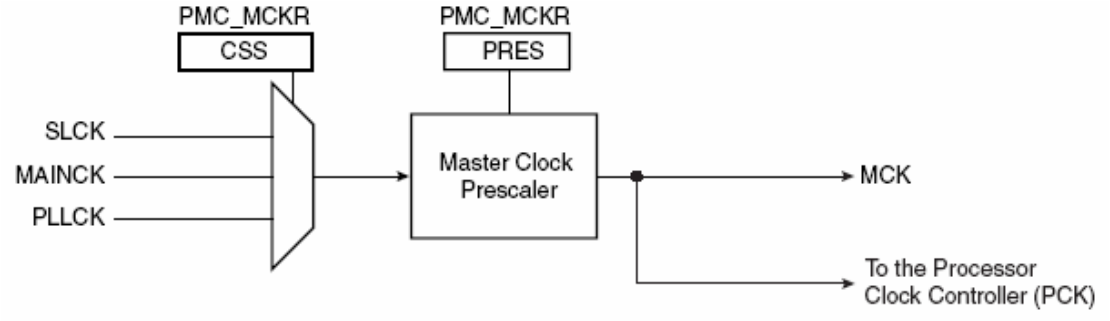

图 12-1

具体的编程序列和时钟切换时间可以详细参照数据手册 PMC 章节。

第 13 章 RTT

实时定时器 RTT，基于一个 32 位的计数器。实时定时器的时钟来源是片内 32K 的 RC 振荡器，RTT 可以用来作为定时器，也可以用来作为秒计数器。

实时定时器的框图如下：

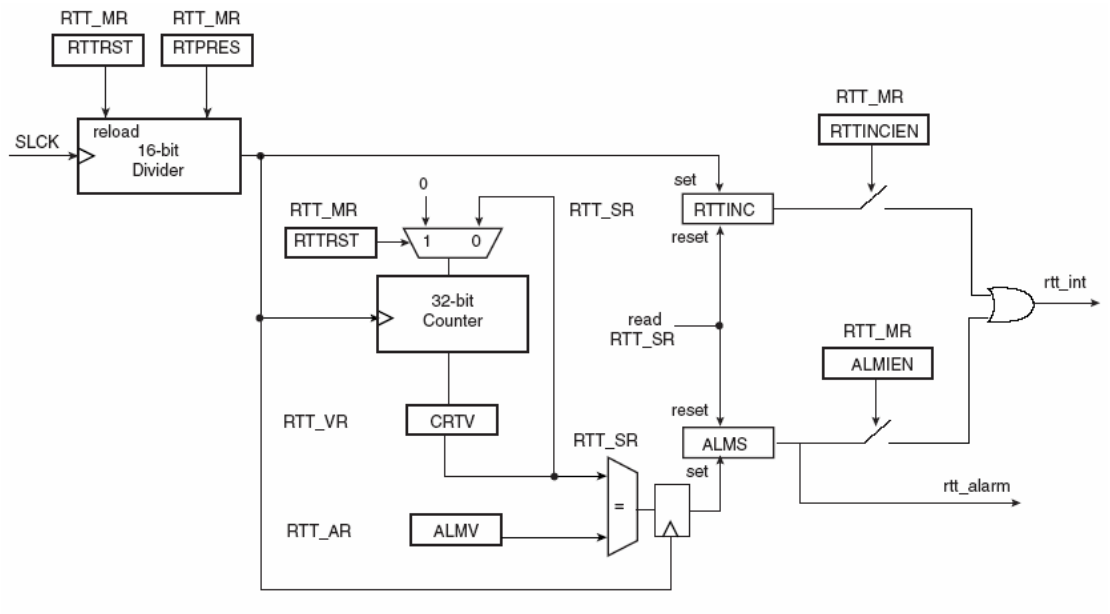

图 13-1

和 RTT 相关的寄存器并不多，只有 RTT_MR，RTT_AR，RTT_VR，RTT_SR 共四个寄存器。由于 RTT 基于 32K 片内 RC 振荡器，所以其比较适合于较长时间的定时，如果将 RTT_MR 域的 RTPRES 设置为 1，定时器最长可以计时 131072 秒，约 36 天；如果将 RTT_MR 域的 RTPRES 设置为 0x00008000，则定时器最长可以记录 2 的 32 次方，约 136 年。由于片内 32K RC 振荡器的精度有限，所以定时器的精度也及其有限。不推荐应用于需要较高时间精度的应用。

第 14 章 USB 转 UART

在本站提供的 AT91SAM7S32 EVB 上提供的是基于 ARK3116T 的 USB 转 UART 解决方案，该芯片是深圳艾科创新微电子有限公司出品，ARK3116T USB-UART 接口转换控制器提供了一个低成本高性能的全双工异步串口和 USB 接口相互转换的解决方案。该芯片包含一个 USB 2.0 全速功能控制器、USB 收发器和带有全部调制解调器控制信号的异步串行数据总线(UART)。利用 USB 接口的优势，用户可以使 UART 接口的外围设备易于使用，例如即插即用和热插拔。在大多数操作系统中，ARK3116T 驱动提供并能模仿传统的串行端口，允许基于串口的应用转为 USB 接口，所以现存的系统固件无需改动。ARK3116T 可以作为红外通讯控制器，支持从 9600 bps 到 115200 bps 的多种波特率。

主要特点

支持 UART 异步串行通信接口，可设置多种波特率（最高 3M bps）

支持 230400，460800，921600 bps 扩展 UART 速率

内置 32 字节 OTP ROM，可定制 VID/PID、厂商名/产品名等信息

驱动虚拟一个普通的串口，现有应用软件无需改动

符合 2.0 版本的 USB 全速技术规范,支持即插即用和热插拔

集成 LDO 电路

内置 DMA 控制器

支持串口 modem 和手机 modem 上网（包括 GPRS 和 CDMA 手

机）

内置硬件流控功能

两个可编程通用 IO 管脚

可作为红外线通信控制器，支持 SIR 协议

提供 Win XP，Win2K，Windows ME, Win98, Linux 的驱动软件

SSOP28 封装

ARK3116T 的典型应用电路如下：

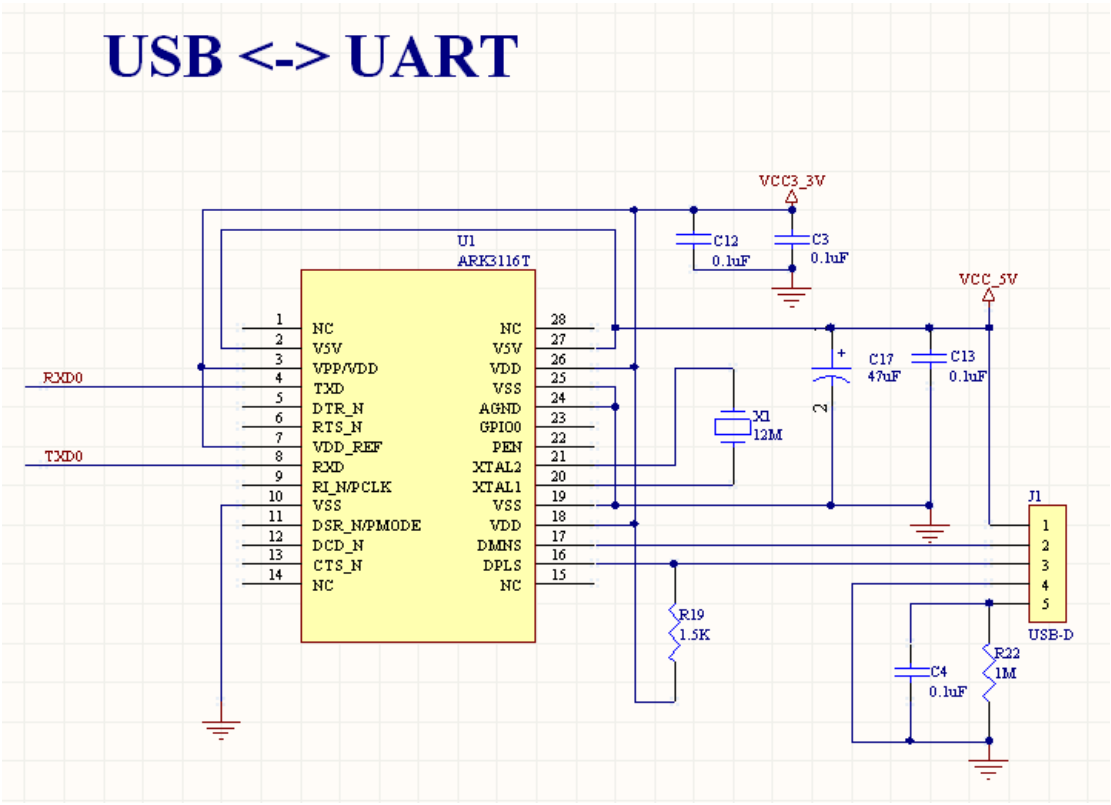

图 14-1

如果对该芯片感兴趣，本站可以提供 ARK3116T 零售服务，同时备有简易评估板：

图 14-2

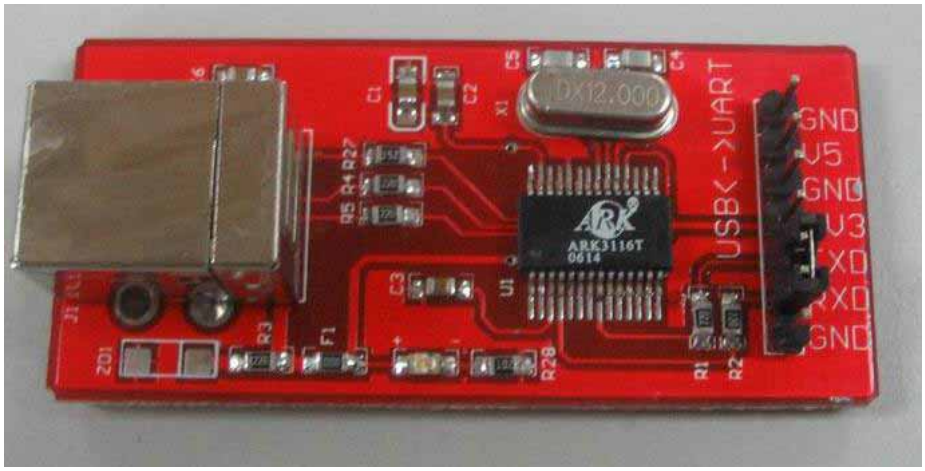

图 14-3

如有兴趣，可以直接和本站联系。

Team Mcuzone

2006-11