

Divider MUX 4	[19:16]	读/写	选择定时器 4 的 MUX 输入： 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: 外部 TCLK1	0x00
Divider MUX 3	[15:12]	读/写	选择定时器 3 的 MUX 输入： 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: 外部 TCLK1	0x00
Divider MUX 2	[11:8]	读/写	选择定时器 2 的 MUX 输入： 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: 外部 TCLK1	0x00
Divider MUX 1	[7:4]	读/写	选择定时器 1 的 MUX 输入： 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: 外部 TCLK1	0x00
Divider MUX 0	[3:0]	读/写	选择定时器 0 的 MUX 输入： 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: 外部 TCLK1	0x00

32. 4. 4. TCON (定时控制寄存器)

寄存器	地址	读/写	描述	复位值
TCON	0x7F006008	读/写	定时器控制寄存器	0x0000_0000

TCON	位	读/写	描述	初始值
Reserved	[31:23]	读	保留。	0x000d

Timer 4 Auto Reload on/off	[22]	读/写	确定定时器 4 的自动加载开/关。 0 = One-shot 1 = 间隔模式(自动重载)	0x0
Timer 4 Manual Update (note)	[21]	读/写	确定定时器 4 的手动更新。 0 = 无操作 1 = 更新 TCNTB4	0x0
Timer 4 Start/Stop	[20]	读/写	确定定时器 4 的启动/停止。 0 = 停止 1 = 开始定时器 4	0x0
Timer 3 Auto Reload on/off	[19]	读/写	确定定时器 3 的自动加载开/关。 0 = One-shot 1 =间隔模式(自动重载)	0x0
Timer 3 output inverter on/off	[18]	读/写	确定定时器 3 的输出反转器开/关。 0 = 逆变器关闭 1 = 逆变器开, 用于 TOUT3	0x0
Timer 3 Manual Update (note)	[17]	读/写	确定定时器 3 的手动更新。 0 = 无操作 1=更新 TCNTB3 或 CMPB3	0x0
Timer 3 Start/Stop	[16]	读/写	确定定时器 3 的启动/停止。 0 = 停止 1 = 开始定时器 3	0x0
Timer 2 Auto Reload on/off	[15]	读/写	确定定时器 2 的自动加载开/关。 0 = One-shot 1 =间隔模式(自动重载)	0x0
Timer 2 Output Inverter on/off	[14]	读/写	确定定时器 2 的输出反转器开/关。 0 =逆变器关闭 1 =逆变器开, 用于 TOUT2	0x0
Timer 2 Manual Update (note)	[13]	读/写	确定定时器 2 的手动更新。 0 =无操作 1 = 更新 TCNTB2 或 TCMPB2	0x0
Timer 2 Start/Stop	[12]	读/写	确定定时器 2 的启动/停止。 0 = 停止 1 =开始定时器 2	0x0
Timer 1 Auto Reload on/off	[11]	读/写	确定定时器 1 的自动加载开/关。 0 = One-shot 1 =间隔模式(自动重载)	0x0
Timer 1 Output Inverter on/off	[10]	读/写	确定定时器 1 的输出反转器开/关。 0 =逆变器关闭 1 = 逆变器开, 用于 TOUT1	0x0
Timer 1 Manual Update (note)	[9]	读/写	确定定时器 1 的手动更新。 0 =无操作 1 = 更新 TCNTB1 或 TCMPB1	0x0

Timer 1 Start/Stop	[8]	读/写	确定定时器 1 的启动/停止。 0 = 停止 1 = 开始定时器 1	0x0
Reserved	[7:5]	读/写	保留。	0x0
Dead Zone Enable	[4]	读/写	确定死区的操作。 0 = 禁用 1 = 使能	0x0
Timer 0 Auto Reload on/off	[3]	读/写	确定定时器 0 的自动加载开/关。 0 = One-shot 1 = 间隔模式(自动重载)	0x0
Timer 0 output inverter on/off	[2]	读/写	确定定时器 0 的输出反转器开/关。 0 = 逆变器关闭 1 = 逆变器开, 用于 TOUT0	0x0
Timer 0 Manual Update (note)	[1]	读/写	确定定时器 0 的手动更新。 0 = 无操作 1 = 更新 TCNTB0 或 TCMPB0	0x0
Timer 0 Start/Stop	[0]	读/写	确定定时器 0 的启动/停止。 0 = 停止 1 = 开始定时器 0	0x0

32.4.5. TCNTB0 (定时器 0 计数寄存器)

寄存器	地址	读/写	描述	复位值
TCNTB0	0x7F00600C	读/写	定时器 0 计数缓冲器	0x0000_0000

TCNTB0	位	读/写	描述	初始状态
Timer 0 Count Buffer	[31:0]	读/写	设置定时器 0 的计数缓冲器的值。	0x00000000

32.4.6. TCMPB0 (定时器 0 比较寄存器)

寄存器	地址	读/写	描述	复位值
TCMPB0	0x7F006010	读/写	定时器 0 比较缓冲寄存器	0x0000_0000

TCMPB0	位	读/写	描述	初始状态
Timer 0 Compare Buffer	[31:0]	读/写	设置定时器 0 的比较缓冲器的值	0x00000000

32.4.7. TCNT00 (定时器 0 计数观察寄存器)

寄存器	地址	读/写	描述	复位值
TCNT00	0x7F006014	读	定时器 0 计数观察寄存器	0x0000_0000

TCNT00	位	读/写	描述	初始状态
Timer 0 Count Observation	[31:0]	读	设置定时器 0 计数观察寄存器的值	0x00000000

32.4.8. TCNTB1 (定时器 1 计数寄存器)

寄存器	地址	读/写	描述	复位值
TCNTB1	0x7F006018	读/写	定时器 1 计数缓冲器	0x0000_0000

TCNTB1	位	读/写	描述	初始状态
Timer 1 Count Buffer	[31:0]	读/写	设置定时器 1 的计数缓冲器的值	0x00000000

32.4.9. TCMPB1 (定时器 1 比较寄存器)

寄存器	地址	读/写	描述	复位值
TCMPB1	0x7F00601c	读/写	定时器 1 比较缓冲寄存器	0x0000_0000

TCMPB1	位	读/写	描述	初始状态
Timer 1 Compare Buffer	[31:0]	读/写	设置定时器 1 的比较缓冲器的值	0x00000000

32.4.10. TCNT0

1 (定时器 1 计数观察寄存器)

寄存器	地址	读/写	描述	复位值
TCNT01	0x7F006020	读	定时器 1 计数观察寄存器	0x0000_0000

TCNT01	位	读/写	描述	初始状态
Timer 1 Count Observation	[31:0]	读	设置定时器 1 计数观察寄存器的值	0x00000000

32.4.11. TCNTB2 (定时器 2 计数寄存器)

寄存器	地址	读/写	描述	复位值
TCNTB2	0x7F006024	读/写	定时器 2 计数缓冲器	0x0000_0000

TCNTB2	位	读/写	描述	初始状态
Timer 2 Count Buffer	[31:0]	读/写	设置定时器 2 的计数缓冲器的值	0x00000000

32.4.12. TCMPB2 (定时器 2 比较寄存器)

寄存器	地址	读/写	描述	复位值
TCMPB2	0x7F006028	读/写	定时器 2 比较缓冲寄存器	0x0000_0000

TCMPB2	位	读/写	描述	初始状态
Timer 2 Compare Buffer	[31:0]	读/写	设置定时器 2 的比较缓冲器的值	0x00000000

32.4.13. TCNT02 (定时器 2 计数观察寄存器)

寄存器	地址	读/写	描述	复位值
TCNT02	0x7F00602c	读	定时器 2 计数观察寄存器	0x0000_0000

TCNT02	位	读/写	描述	初始状态
Timer 2 Count Observation	[31:0]	读	设置定时器 2 计数观察寄存器的值	0x00000000

32.4.14. TCNTB3 (定时器 3 计数寄存器)

寄存器	地址	读/写	描述	复位值
TCNTB3	0x7F006030	读/写	定时器 3 计数缓冲器	0x0000_0000

TCNTB3	位	读/写	描述	初始状态
Timer 3 Count Buffer	[31:0]	读/写	设置定时器 3 的计数缓冲器的值	0x00000000

32.4.15. TCMPB3 (定时器 3 比较寄存器)

寄存器	地址	读/写	描述	复位值
TCMPB3	0x7F006034	读/写	定时器 3 比较缓冲寄存器	0x0000_0000

TCMPB3	位	读/写	描述	初始状态
Timer 3 Compare Buffer	[31:0]	读/写	设置定时器 3 的比较缓冲器的值	0x00000000

32.4.16. TCNT03 (定时器 3 计数观察寄存器)

寄存器	地址	读/写	描述	复位值
TCNT03	0x7F006038	读	定时器 3 计数观察寄存器	0x0000_0000

TCNT03	位	读/写	描述	初始状态
Timer 3 Count Observation	[31:0]	读	设置定时器 3 计数观察寄存器的值	0x00000000

32. 4. 17. TCNTB4 (定时器 4 计数寄存器)

寄存器	地址	读/写	描述	复位值
TCNTB4	0x7F00603c	读/写	定时器 4 计数缓冲器	0x0000_0000

TCNTB4	位	读/写	描述	初始状态
Timer 4 Count Buffer	[31:0]	读/写	设置定时器 4 的计数缓冲器的值	0x00000000

32. 4. 18. TCNT04 (定时器 4 计数观察寄存器)

寄存器	地址	读/写	描述	复位值
TCNT04	0x7F006040	读	定时器 4 计数观察寄存器	0x0000_0000

TCNT04	位	读/写	描述	初始状态
Timer 4 Count Observation	[31:0]	读	设置定时器 4 计数观察寄存器的值	0x00000000

32. 4. 19. TINT_CSTAT (定时器中断控制和状态寄存器)

寄存器	位	读/写	描述	复位值
TINT_CSTAT	0x7F006044	读/写	定时器中断控制和状态寄存器	0x0000_0000

TINT_CSTAT	位	读/写	描述	初始状态
Reserved	[31:10]	读	保留位	0x00000
Timer 4 Interrupt Status	[9]	读/写	定时器 4 中断状态位。通过写 ‘1’ 清除该位	0x0

Timer 3 Interrupt Status	[8]	读/写	定时器 3 中断状态位。通过写 ‘1’ 清除该位	0x0
Timer 2 Interrupt Status	[7]	读/写	定时器 2 中断状态位。通过写 ‘1’ 清除该位	0x0
Timer 1 Interrupt Status	[6]	读/写	定时器 1 中断状态位。通过写 ‘1’ 清除该位	0x0
Timer 0 Interrupt Status	[5]	读/写	定时器 0 中断状态位。通过写 ‘1’ 清除该位	0x0
Timer 4 interrupt Enable	[4]	读/写	定时器 4 中断启动。 1: 启动 0: 禁止	0x0
Timer 3 interrupt Enable	[3]	读/写	定时器 3 中断启动。 1: 启动 0: 禁止	0x0
Timer 2 interrupt Enable	[2]	读/写	定时器 2 中断启动。 1: 启动 0: 禁止	0x0
Timer 1 interrupt Enable	[1]	读/写	定时器 1 中断启动。 1: 启动 0: 禁止	0x0
Timer 0 interrupt Enable	[0]	读/写	定时器 0 中断启动。 1: 启动 0: 禁止	0x0

33 RTC 实时时钟

本节介绍了实时时钟（RTC）在 S3C6410 RISC 微处理器上的功能及其使用。当系统电源关闭时，通过备用电源可以运行实时时钟（RTC）单元。数据包含的时间，即秒，分钟，小时，日期，日，月和年。RTC 单元操作一个外部 32.768kHz 的晶体，并可以执行报警功能。

33.1 RTC 实时时钟的特性

实时时钟包括以下功能：

- 二进制编码数据：秒，分钟，小时，日期，日，月和年。
- 闰年发生器。
- 报警功能：报警中断或从断电模式中唤醒。
- 时钟计数功能：时钟节拍中断或从断电模式中唤醒。
- 不存在千年虫问题。
- 独立地电源引脚（RTCVDD）。
- 支持毫秒标记的时间中断信号，用于 RTOS 内核时间标记。

33.2 实时时钟的操作说明

实时时钟的结构框图，如图 33-1 所示。

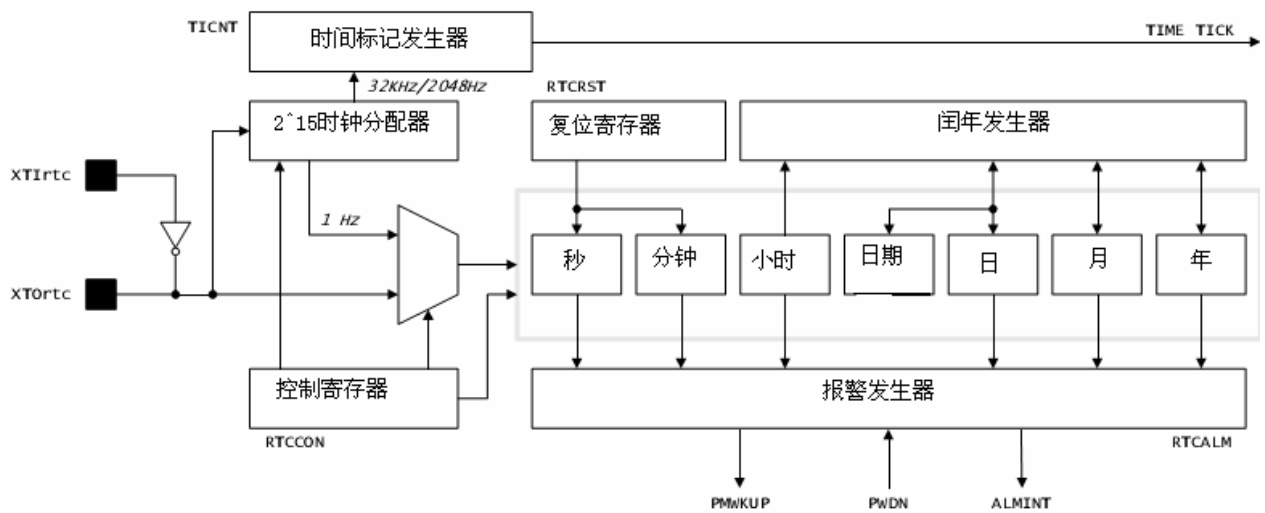


图 33-1 实时时钟的结构框图

1. 闰年发生器

闰年发生器通过 BCDDAY, BCDMON 和 BCDYEAR 的数据来决定每个月的最后一天是 28, 29, 30 还是 31。这个模块是通过决定最后的日期来判断闰年的。一个 8 位的计数器只能代表两个 BCD 数字, 因此它不能决定 ‘00’ 年 (年的最后两个数字为 ‘00’) 是不是闰年。举例来说, 它不能区分 1900 年和 2000 年。要解决这个问题, S3C6410 中的实时时钟模块, 在 2000 年中, 硬连接逻辑支持闰年。注意 1900 年不是闰年, 而 2000 年是闰年。因此在 S3C6410 中的 ‘00’ 的两个数字表示 2000 而不是 1900。

2. 读/写寄存器

RTCCON 寄存器的位 0 必须被设置为高位, 原因是可以正常写入实时时钟模块中的 BCD 寄存器, 以显示秒, 分钟, 小时, 日期, 日, 月和年。CPU 必须分别在 RTC 模块的 BCDSEC, BCDMIN, BCDHOUR, BCDDATE, BCDDAY, BCDMON 和 BCDYEAR 寄存器中读取数据。但是, 因为多个寄存器被读取, 所以可能有一秒的偏差存在。例如, 当用户从 BCDYEAR 到 BCDMIN 读取寄存器时, 结果假设为 2059 (年), 12 (月), 31 (日期), 23 (小时) 和 59 (分钟)。当用户读取 BCDSEC 寄存器及值范围从 1 到 59 (秒) 时, 没有问题, 但值为 0 秒, 年, 月, 日, 小时和分钟将被改变为 2060 (年), 1 (月), 1 (日期), 0 (小时) 和 0 (分钟), 就是因为这一秒的变差。在这种情况下, 如果 BCDSEC 置 0, 用户必须从 BCDYEAR 到 BCDSEC 重新读取。

3. 备份电源操作

通过备用电池可以驱动实时时钟逻辑, 它是通过 RTCVDD 引脚进入实时时钟模块来提供电源的, 即使