# AM17x/AM18x ARM Microprocessor Real-Time Clock (RTC)

# User's Guide

Literature Number: SPRUFV9A
August 2010

# List of Figures

# List of Tables

# Read This First

## About This Manual

This document describes the real-time clock (RTC). The RTC provides a time reference and the capability to generate time-based alarms to interrupt the CPU.

## Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.

## Related Documentation From Texas Instruments

Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

The current documentation that describes the DSP, related peripherals, and other technical collateral, is available in the C6000 DSP product folder at: www.ti.com/c6000.

**SPRUGU3 — *AM1705 ARM Microprocessor System Reference Guide.*** Describes the ARM subsystem, system memory, memory protection unit (MPU), device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, ARM interrupt controller (AINTC), and system configuration module.

**SPRUGR6 — *AM1707 ARM Microprocessor System Reference Guide.*** Describes the ARM subsystem, system memory, memory protection unit (MPU), device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, ARM interrupt controller (AINTC), and system configuration module.

**SPRUGU4 — *AM1806 ARM Microprocessor System Reference Guide.*** Describes the ARM subsystem, system memory, memory protection unit (MPU), device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, ARM interrupt controller (AINTC), and system configuration module.

**SPRUGM9 — *AM1808 ARM Microprocessor System Reference Guide.*** Describes the ARM subsystem, system memory, memory protection unit (MPU), device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, ARM interrupt controller (AINTC), and system configuration module.

**SPRUFU0 — *AM17x/AM18x ARM Microprocessor Peripherals Overview Reference Guide.*** Provides an overview and briefly describes the peripherals available on the AM17x/AM18x ARM Microprocessors.

User's Guide
SPRUFV9A–August 2010

# Real-Time Clock (RTC)

## 1 Introduction

### 1.1 Purpose of the Peripheral

The RTC provides a time reference to an application running on the device. The current date and time is tracked in a set of counter registers that update once per second. The time can be represented in 12-hour or 24-hour mode. The calendar and time registers are buffered during reads and writes so that updates do not interfere with the accuracy of the time and date.

Alarms are available to interrupt the CPU at a particular time, or at periodic time intervals, such as once per minute or once per day. In addition, the RTC can interrupt the CPU every time the calendar and time registers are updated, or at programmable periodic intervals.

### 1.2 Features

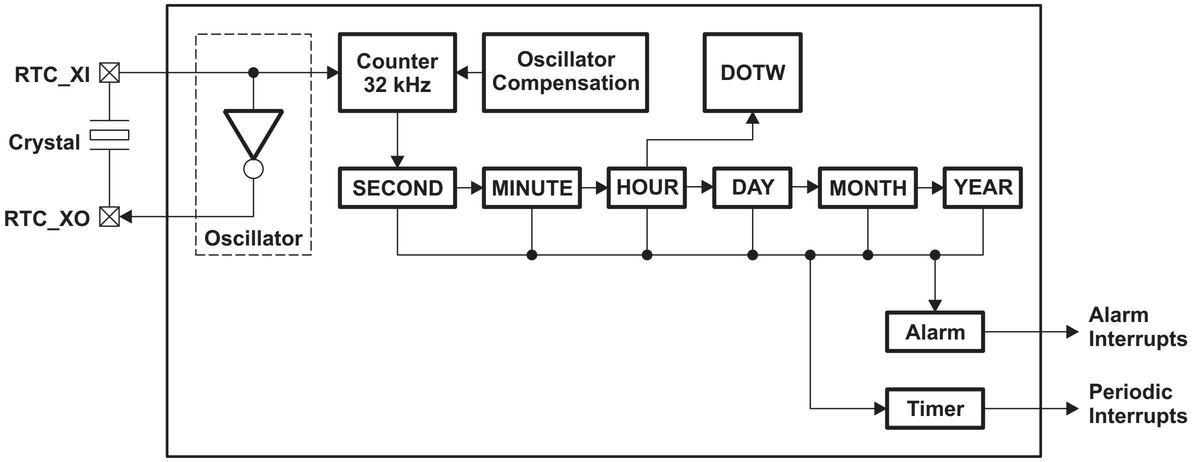
The real-time clock (RTC) provides the following features:

- 100-year calendar (xx00 to xx99)
- Counts seconds, minutes, hours, day of the week, date, month, and year with leap year compensation
- Binary-coded-decimal (BCD) representation of time, calendar, and alarm
- 12-hour clock mode (with AM and PM) or 24-hour clock mode
- Alarm interrupt
- Periodic interrupt
- Single interrupt to the CPU
- Supports external 32.768-kHz crystal or external clock source of the same frequency
- Isolated power supply

### 1.3 Block Diagram

Figure 1 shows a block diagram of the RTC.

**Figure 1. Real-Time Clock Block Diagram**



# 2 Architecture

## 2.1 Clock Source

The clock reference for the RTC is an external 32.768-kHz crystal or an external clock source of the same frequency. The RTC also has a separate power supply that is isolated from the rest of the system. When the CPU and other peripherals are without power, the RTC can remain powered to preserve the current time and calendar information.

The source for the RTC reference clock may be provided by a crystal or by an external clock source. The RTC has an internal oscillator buffer to support direct operation with a crystal. The crystal is connected between pins RTC_XI and RTC_XO. RTC_XI is the input to the on-chip oscillator and RTC_XO is the output from the oscillator back to the crystal. For more information about the RTC crystal connection, see your device-specific data manual.

An external 32.768-kHz clock source may be used instead of a crystal. In such a case, the clock source is connected to RTC_XI, and RTC_XO is left unconnected.

If the RTC is not used, the RTC_XI pin should be held low and RTC_XO should be left unconnected. The RTCDISABLE bit in the control register (CTRL) can be set to save power; however, the RTCDISABLE bit should not be cleared once it has been set. If the application requires the RTC module to stop and continue, the RUN bit in CTRL should be used instead.

## 2.2 Signal Descriptions

The RTC signals are listed in Table 1.

**Table 1. Real-Time Clock Signals**

| Signal | I/O | Description |
|---|---|---|
| RTC_XI | I | RTC time base input signal. RTC_XI can either be driven with a 32.768-kHz reference clock, or RTC_XI and RTC_XO can be connected to an external crystal. This signal is the input to the RTC internal oscillator. |
| RTC_XO | O | RTC time base output signal. RTC_XO is the output from the RTC internal oscillator. If a crystal is not used as the time base for RTC_XI, RTC_XO should be left unconnected. |

## 2.3 Isolated Power Supply

The RTC has a power supply that is isolated from the rest of the system. This allows the RTC to continue to run while the rest of the system is not powered. In this state, the RTC time and calendar counters continue to run, but the powered down CPU is not able to receive RTC interrupts. Separate power supply pins for the RTC are provided on the device package.

### 2.3.1 Split-Power Circuitry

To decrease power consumption, RTC includes leakage-isolation circuitry that is activated by setting the SPLITPOWER bit in the control register (CTRL). Because of its isolated power supply, RTC does not have a power-on hardware reset signal. Therefore, upon initial device power-on, the RTC is in an unknown state until it has been properly configured. After the RTC module has been configured once, it functions as programmed as long as its power supply and clock source are provided.

### 2.3.2 Power Considerations

The RTC leakage-isolation circuitry requires that the CPU supply be powered down to VSS when the RTC is powered on while the rest of the device is powered off. A floating CPU supply creates undesired RTC leakage current. Also, the RTC power consumption is higher when the CPU is powered on versus the RTC power consumption when the CPU is powered off. Therefore, if the RTC module is expected to run from a small-capacity power supply (ex. watch battery) while the rest of the device is powered off, a power system should be implemented such that the RTC is powered from a high-capacity power supply when the CPU is powered on.

## 2.4 Operation

### 2.4.1 Using the Real-Time Clock Time and Calendar Registers

The current time and date are maintained in the RTC time and calendar registers.

#### 2.4.1.1 Time/Calendar Data Format

The time and calendar data in the RTC is stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. Although most of the time/calendar registers have 4 bits assigned to each BCD digit, some of the register fields are shorter since the range of valid numbers may be limited. For example, only 3 bits are required to represent the day since only BCD numbers 1 though 7 are required.

The following time and calendar registers are supported (BCD Format):

- SECOND - Second Count (00-59)
- MINUTE - Minute Count (00-59)
- HOUR - Hour Count (12HR: 01-12; 24HR: 00-23)
- DAY - Day of the Month Count (01-31)
- MONTH - Month Count (01-12; JAN = 1)
- YEAR - Year Count (00-99)
- DOTW - Day of the Week Count (0-6; SUN = 0)

Note that the ALARM registers which share the names above also share the same BCD formatting.

#### 2.4.1.2 12-Hour and 24-Hour Modes

The current time can be represented in 12-hour or 24-hour mode by configuring the HOURMODE bit in the control register (CTRL):

- When HOURMODE = 0, 24-hour mode is selected. The hours are represented as 00 through 23. The MERIDIEM bit in the HOURS register has no function and should be cleared.
- When HOURMODE = 1, 12-hour mode is selected. The hours are represented as 00 through 12. MERIDIEM = 0 indicates ante meridiem (AM), and MERIDIEM = 1 indicates post meridiem (PM).

#### 2.4.1.3 Reading from Time/Calendar Registers

The time/calendar registers are updated every second as the time changes. During a read of the SECOND register, the RTC copies the current values of the time/date registers into shadow read registers. This isolation assures that the CPU can capture all the time/date values at the moment of the SECOND read request and not be subject to changing register values from time updates.

If desired, the RTC also provides a one-time-triggered minute-rounding feature to round the MINUTE:SECOND registers to the nearest minute (with zero seconds). This feature is enabled by setting the ROUNDMIN bit in the control register (CTRL); the RTC automatically rounds the time values to the nearest minute upon the next read of the SECOND register.

#### 2.4.1.4 Writing to Time/Calendar Registers

When setting the RTC time and date, values are written directly to the time/calendar registers. Therefore, care must be taken to avoid writing to the time/calendar registers while the time is updating to the next second. This can be accomplished in one of two ways:

1. The RTC can be stopped by clearing the RUN bit in the control register (CTRL). When stopped, there is no danger of contention during writes because the registers do not auto-update.
2. The BUSY bit in the status register (STATUS) is low when a time update does not take place for at least 15 μs. By checking for a low BUSY bit before writing to registers, the CPU is assured of a 15 μs window of time during which multiple accesses to the time/calendar registers can be performed.

After writing to a time/calendar register, the RTC requires four peripheral clock cycles to update the register value. Any reads that take place within four peripheral clock cycles of a write returns old data.

Note that all registers in the RTC except for KICK*n*R have write-protection. See Section 2.6 for information on unlocking registers.

### 2.4.2 Real-Time Clock Update Cycle

The RTC executes an update cycle once per second to update the current time in the time/calendar registers. The update cycle also compares each alarm register with the corresponding time register. These comparisons are done to determine when to trigger an alarm. The BUSY bit in the status register (STATUS) provides a mechanism to indicate when the time/calendar registers are updated. When the BUSY bit is high, an update takes place within 15 μs. When BUSY returns low again, the update has been completed.

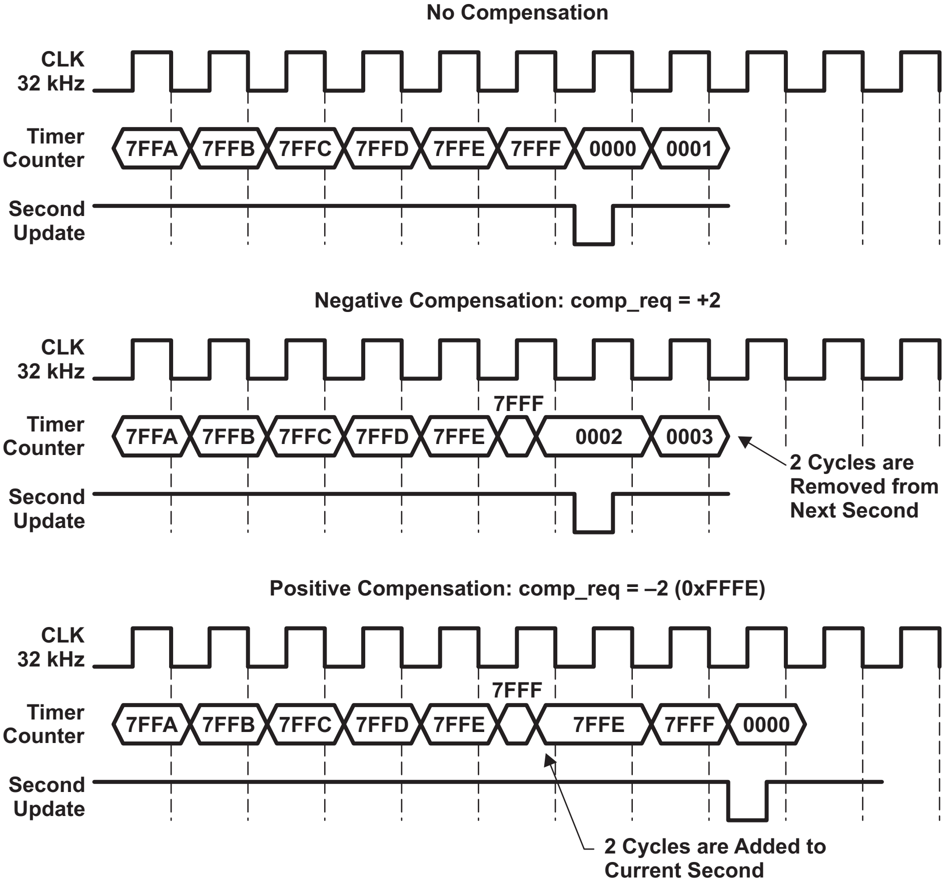
The BUSY bit should be checked when writing to any of the following registers while RTC is running:

- SECOND
- MINUTE
- HOUR
- DAY
- MONTH
- YEAR
- DOTW
- ALARMSECOND (when ALARM interrupt is enabled)
- ALARMMINUTE (when ALARM interrupt is enabled)
- ALARMHOUR (when ALARM interrupt is enabled)
- ALARMDAY (when ALARM interrupt is enabled)
- ALARMMONTH (when ALARM interrupt is enabled)
- ALARMYEAR (when ALARM interrupt is enabled)
- CTRL (SET32COUNTER field only -- the other fields in CTRL do not require BUSY to be low)
- INTERRUPT
- COMPLSB (when oscillator drift compensation is enabled)
- COMPMSB (when oscillator drift compensation is enabled)

### 2.4.3 Oscillator Drift Compensation

If the RTC 32.768-kHz reference clock is susceptible to oscillator drift, the RTC provides the ability to compensate the update cycle by subtracting oscillator periods. The COMPMSB and COMPLSB registers hold the number of two's complement reference periods to subtract from the update cycle every hour. For example, Figure 2 shows how programming the value of 2h into the compensation registers shortens the update cycle by two 32.786-kHz reference periods every hour. Figure 2 also shows how programming the value of FFFEh (decimal negative 2) into the compensation register lengthens the update cycle by two reference periods every hour. To enable the oscillator compensation, the AUTOCOMP bit in the control register (CTRL) must be set.

Figure 2. 32-kHz Oscillator Counter Compensation



## 2.5 Interrupt Requests

The RTC provides the ability to interrupt the CPU based on two events: a periodic interrupt and an alarm interrupt. Although two interrupt sources are available, the RTC makes a single interrupt request to the CPU.

When the device is initially powered on, the RTC may issue spurious interrupt signals to the CPU. To avoid issues, a software reset should be performed on the RTC module before the CPU interrupt controller is initialized. See Section 2.10 for more information on reset considerations.

### 2.5.1 Alarm Interrupt Enable and Status Bits

The ALARM bit in the interrupt register (INTERRUPT) enables the alarm interrupt. When the current time and date match the ALARMSECOND, ALARMMINUTE, ALARMHOUR, ALARMDAY, ALARMMONTH, and ALARMYEAR registers, the RTC issues an interrupt to the CPU and sets the ALARM bit in the status register (STATUS). Once set, the ALARM status bit stays high until cleared by a write of 1 to the ALARM bit.

As with writing to time and calendar registers (Section 2.4.1.4), writes to the INTERRUPT and STATUS registers should only be done when the RTC is stopped or when the BUSY bit is low.

Note that all registers in the RTC except for KICK*n*R have write-protection. See Section 2.6 for information on unlocking registers.

### 2.5.2 Periodic Interrupt Enable and Status Bits

The TIMER bit and EVERY field in the interrupt register (INTERRUPT) work together to enable periodic interrupts. When the TIMER bit is enabled, interrupts are issued at a time period indicated by the EVERY field (0 = Second, 1h = Minute, 2h = Hour, 3h = Day). Regardless of the period selected in the EVERY field, the periodic timer status bits (DAYEVT, HREVT, MINEVT, SECEVT) are set in the status register (STATUS) whenever they are valid. Note that the appropriate status bits are set when the TIME bit is enabled, not when the desired interrupt is generated. Active periodic status bits remain high as long as the TIMER bit is enabled.

For example, if daily periodic interrupts are enabled and the time (in HH:MM:SS format) transitions from 23:59:59 to 00:00:00, the STATUS register sets all four periodic status bits (DAYEVT, HREVT, MINEVT, and SECEVT) because all four time periods were incremented. These bits all remain high until:

1. The TIME bit is cleared and all four status bits clear to zero until TIME is set again **OR**
2. The current time reaches 00:00:01. At that point, the SECEVT remains set while the DAYEVT, HREVT, and MINEVT bits are cleared. The next interrupt is not generated until the next day transition.

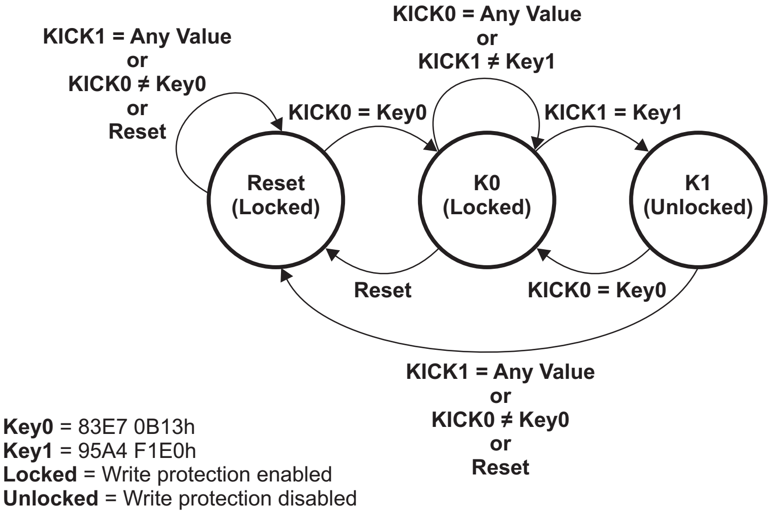
As with writing to time and calendar registers (Section 2.4.1.4), writes to the INTERRUPT and STATUS registers should only be done when the RTC is stopped or when the BUSY bit is low.

Note that all registers in the RTC except for KICK*n*R have write-protection. See Section 2.6 for information on unlocking registers.

## *2.6 Register Protection Against Spurious Writes*

All registers in the RTC except for the KICK*n*R registers are protected from spurious writes. Out of reset, writes to protected registers are disabled until the registers are unlocked using the KICK*n*R registers. To unlock the registers, a key of 83E7 0B13h needs to be written to KICK0R, followed by a write of 95A4 F1E0h to KICK1R. Registers remain unlocked until write protection is enabled again by writing any value to KICK0R or KICK1R. The write protection state machine is shown in Figure 3.

**Figure 3. Kick State Machine**



## 2.7 General-Purpose Scratch Registers

The RTC provides three general-purpose registers (SCRATCH*n*) that can be used to store 32-bit words -- these registers have no functional purpose for the RTC. Software using the RTC may find the SCRATCH*n* registers to be useful in indicating RTC states. For example, the SCRATCH*n* registers may be used to indicate write-protection lock status or unintentional power downs.

To indicate write-protection, the software should write a unique value to one of the SCRATCH*n* registers when write-protection is disabled and another unique value when write-protection is enabled again. In this way, the lock-status of the registers can be determined quickly by reading the SCRATCH register.

To indicate unintentional power downs, the software should write a unique value to one of the SCRATCH*n* registers when RTC is configured and enabled. If the RTC is unintentionally powered down, the value written to the SCRATCH register is cleared.

## 2.8 Real-Time Clock Response to Low Power Modes (Idle Configurations)

The device is divided into idle domains that can be programmed to be idle or active. The state of all domains is called the idle configuration. The RTC runs on its own external clock source and is not affected by any of the other device idle domains.

## 2.9 Emulation Modes of the Real-Time Clock

The RTC always continues to run regardless of the state (running/halted) of the emulation debugger software.

## 2.10 Reset Considerations

When the device is initially powered on, the RTC may issue spurious interrupt signals to the CPU. To avoid issues, a software reset should be performed on the RTC module before the CPU interrupt controller is initialized.

As the RTC is configured, the SPLITPOWER bit in the control register (CTRL) should be set.

A software reset is performed on the RTC by setting the SWRESET bit in the oscillator register (OSC). The software reset applies to all registers except the oscillator (OSC) and kick (KICK*n*R) registers. The RTC requires three 32.768-kHz reference clocks to pass before RTC registers can be accessed.

## 3 Registers

Table 2 lists the memory-mapped registers for the RTC. See your device-specific data manual for the memory address of these registers.

**Table 2. Real-Time Clock (RTC) Registers**

| Address Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 0h | SECOND | Seconds Register | Section 3.1 |
| 4h | MINUTE | Minutes Register | Section 3.2 |
| 8h | HOUR | Hours Register | Section 3.3 |
| Ch | DAY | Day of the Month Register | Section 3.4 |
| 10h | MONTH | Month Register | Section 3.5 |
| 14h | YEAR | Year Register | Section 3.6 |
| 18h | DOTW | Day of the Week Register | Section 3.7 |
| 20h | ALARMSECOND | Alarm Seconds Register | Section 3.8 |
| 24h | ALARMMINUTE | Alarm Minutes Register | Section 3.9 |
| 28h | ALARMHOUR | Alarm Hours Register | Section 3.10 |
| 2Ch | ALARMDAY | Alarm Days Register | Section 3.11 |
| 30h | ALARMMONTH | Alarm Months Register | Section 3.12 |
| 34h | ALARMYEAR | Alarm Years Register | Section 3.13 |
| 40h | CTRL | Control Register | Section 3.14 |
| 44h | STATUS | Status Register | Section 3.15 |
| 48h | INTERRUPT | Interrupt Enable Register | Section 3.16 |
| 4Ch | COMPLSB | Compensation (LSB) Register | Section 3.17 |
| 50h | COMPMSB | Compensation (MSB) Register | Section 3.18 |
| 54h | OSC | Oscillator Register | Section 3.19 |
| 60h | SCRATCH0 | Scratch 0 Register (General-Purpose) | Section 3.20 |
| 64h | SCRATCH1 | Scratch 1 Register (General-Purpose) | Section 3.20 |
| 68h | SCRATCH2 | Scratch 2 Register (General-Purpose) | Section 3.20 |
| 6Ch | KICK0R | Kick 0 Register (Write Protect) | Section 3.21 |
| 70h | KICK1R | Kick 1 Register (Write Protect) | Section 3.21 |

## *3.1 Second Register (SECOND)*

**NOTE:** Out of reset, the second register (SECOND) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The second register (SECOND) sets the second value of the current time. Seconds are stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The SECOND register is shown in Figure 4 and described in Table 3.

**Figure 4. Second Register (SECOND)**

| 31 16 |
|---|
| Reserved |
| R-0 |

| 15 7 | 6 4 | 3 0 |
|---|---|---|
| Reserved | SEC1 | SEC0 |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 3. Second Register (SECOND) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved. |
| 6-4 | SEC1 | 0-5h | Most significant digit of second value. Range for SEC1:SEC0 is 00-59. |
| 3-0 | SEC0 | 0-9h | Least significant digit of second value. Range for SEC1:SEC0 is 00-59. |

## *3.2 Minute Register (MINUTE)*

**NOTE:** Out of reset, the minute register (MINUTE) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The minute register (MINUTE) sets the minute value of the current time. Minutes are stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The MINUTE register is shown in Figure 5 and described in Table 4.

**Figure 5. Minute Register (MINUTE)**

| 31 16 |
|---|
| Reserved |
| R-0 |

| 15 7 | 6 4 | 3 0 |
|---|---|---|
| Reserved | MIN1 | MIN0 |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 4. Minute Register (MINUTE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved. |
| 6-4 | MIN1 | 0-5h | Most significant digit of minute value. Range for MIN1:MIN0 is 00-59. |
| 3-0 | MIN0 | 0-9h | Least significant digit of minute value. Range for MIN1:MIN0 is 00-59. |

## 3.3 Hour Register (HOUR)

**NOTE:** Out of reset, the hour register (HOUR) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The hour register (HOUR) sets the hour value of the current time. Hours are stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The HOUR register is shown in Figure 6 and described in Table 5.

**Figure 6. Hour Register (HOUR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | MERIDIEM | Rsvd | HOUR1 | | HOUR0 | |
| R-0 | | R/W-0 | R-0 | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 5. Hour Register (HOUR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7 | MERIDIEM | | Determines whether the hour is ante meridiem (AM) or post meridiem (PM) when 12-hour mode is enabled. |
| | | 0 | Hour is AM |
| | | 1 | Hour is PM |
| 6 | Reserved | 0 | Reserved. |
| 5-4 | HOUR1 | 0-2h | Most significant digit of hours value. Range for HOUR1:HOUR0 is 00-24. |
| 3-0 | HOUR0 | 0-9h | Least significant digit of hours value. Range for HOUR1:HOUR0 is 00-24. |

### *3.4 Day of the Month Register (DAY)*

**NOTE:** Out of reset, the day of the month register (DAY) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The day of the month register (DAY) sets the day of the month value of the current date. Days are stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The DAY register is shown in Figure 7 and described in Table 6.

**Figure 7. Days Register (DAY)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | DAY1 | | DAY0 | |
| R-0 | | R/W-0 | | R/W-1 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6. Day Register (DAY) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved. |
| 5-4 | DAY1 | 0-3h | Most significant digit of day of the month value. Range for DAY1:DAY0 is 01-31. |
| 3-0 | DAY0 | 0-9h | Least significant digit of day of the month value. Range for DAY1:DAY0 is 01-31. |

### *3.5 Month Register (MONTH)*

**NOTE:** Out of reset, the month register (MONTH) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The month register (MONTH) sets the month in the year value of the current date. The month is stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The MONTH register is shown in Figure 8 and described in Table 7.

**Figure 8. Month Register (MONTH)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|
| Reserved | | MONTH1 | MONTH0 | |
| R-0 | | R/W-0 | R/W-1 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7. Month Register (MONTH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Reserved. |
| 4 | MONTH1 | 0-1h | Most significant digit of months value. For MONTH1:MONTH0, JAN = 01 and DEC = 12. |
| 3-0 | MONTH0 | 0-9h | Least significant digit of months value. For MONTH1:MONTH0, JAN = 01 and DEC = 12. |

## *3.6 Year Register (YEAR)*

**NOTE:** Out of reset, the year register (YEAR) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The year register (YEAR) sets the year value of the current date. The year is stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The YEAR register is shown in Figure 9 and described in Table 8.

**Figure 9. Year Register (YEAR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 8 | 7 4 | 3 0 |
|---|---|---|
| Reserved | YEAR1 | YEAR0 |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8. Year Register (YEAR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7-4 | YEAR1 | 0-9h | Most significant digit of years value. Range for YEAR1:YEAR0 is 00-99. |
| 3-0 | YEAR0 | 0-9h | Least significant digit of years value. Range for YEAR1:YEAR0 is 00-99. |

## *3.7 Day of the Week Register (DOTW)*

**NOTE:** Out of reset, the day of the week register (DOTW) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The day of the week register (DOTW) sets the day of the week value of the current date. The day of the week is stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The DOTW register is shown in Figure 10 and described in Table 9.

**Figure 10. Day of the Week Register (DOTW)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 3 | 2 0 |
|---|---|
| Reserved | DOTW |
| R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9. Day of the Week Register (DOTW) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved. |
| 2-0 | DOTW | 0-6h | Day of the week. Sunday = 0, Saturday = 6h. |

## 3.8 Alarm Second Register (ALARMSECOND)

**NOTE:** Out of reset, the alarm second register (ALARMSECOND) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The alarm second register (ALARMSECOND) sets the second value for the alarm interrupt. Seconds are stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The ALARMSECOND register is shown in Figure 11 and described in Table 10.

**Figure 11. Alarm Second Register (ALARMSECOND)**

| 31-16 |
|---|
| Reserved |
| R-0 |

| 15-7 | 6-4 | 3-0 |
|---|---|---|
| Reserved | SEC1 | SEC0 |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 10. Alarm Second Register (ALARMSECOND) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved. |
| 6-4 | SEC1 | 0-5h | Most significant digit of alarm seconds value. Range for SEC1:SEC0 is 00-59. |
| 3-0 | SEC0 | 0-9h | Least significant digit of alarm seconds value. Range for SEC1:SEC0 is 00-59. |

## 3.9 Alarm Minute Register (ALARMMINUTE)

**NOTE:** Out of reset, the alarm minute register (ALARMMINUTE) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The alarm minute register (ALARMMINUTE) sets the minute value for the alarm interrupt. Minutes are stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The ALARMMINUTE register is shown in Figure 12 and described in Table 11.

**Figure 12. Alarm Minute Register (ALARMMINUTE)**

| 31-16 |
|---|
| Reserved |
| R-0 |

| 15-7 | 6-4 | 3-0 |
|---|---|---|
| Reserved | MIN1 | MIN0 |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 11. Alarm Minute Register (ALARMMINUTE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved. |
| 6-4 | MIN1 | 0-5h | Most significant digit of alarm minutes value. Range for MIN1:MIN0 is 00-59. |
| 3-0 | MIN0 | 0-9h | Least significant digit of alarm minutes value. Range for MIN1:MIN0 is 00-59. |

### 3.10 Alarm Hour Register (ALARMHOUR)

**NOTE:** Out of reset, the alarm hour register (ALARMHOUR) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The alarm hour register (ALARMHOUR) sets the hour value for the alarm interrupt. Hours are stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The ALARMHOUR register is shown in Figure 13 and described in Table 12.

**Figure 13. Alarm Hour Register (ALARMHOUR)**

| 31-16 |
|---|
| Reserved |
| R-0 |

| 15-8 | 7 | 6 | 5-4 | 3-0 |
|---|---|---|---|---|
| Reserved | MERIDIEM | Rsvd | HOUR1 | HOUR0 |
| R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 12. Alarm Hour Register (ALARMHOUR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7 | MERIDIEM | | Determines whether the hour is ante meridiem (AM) or post meridiem (PM) when 12-hour mode is enabled. |
| | | 0 | Hour is AM |
| | | 1 | Hour is PM |
| 6 | Reserved | 0 | Reserved. |
| 5-4 | HOUR1 | 0-2h | Most significant digit of hours value. Range for HOUR1:HOUR0 is 00-24. |
| 3-0 | HOUR0 | 0-9h | Least significant digit of hours value. Range for HOUR1:HOUR0 is 00-24. |

### 3.11 Alarm Day of the Month Register (ALARMDAY)

**NOTE:** Out of reset, the alarm day of the month register (ALARMDAY) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The alarm day of the month register (ALARMDAY) sets the day of the month value for the alarm interrupt. Days are stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The ALARMDAYS register is shown in Figure 14 and described in Table 13.

**Figure 14. Alarm Day Register (ALARMDAY)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | DAY1 | | DAY0 | |
| R-0 | | R/W-0 | | R/W-1 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13. Alarm Day Register (ALARMDAY) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved. |
| 5-4 | DAY1 | 0-3h | Most significant digit of day of the month value. Range for DAY1:DAY0 is 01-31. |
| 3-0 | DAY0 | 0-9h | Least significant digit of day of the month value. Range for DAY1:DAY0 is 01-31. |

## *3.12 Alarm Month Register (ALARMMONTH)*

**NOTE:** Out of reset, the alarm month register (ALARMMONTH) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The alarm month register (ALARMMONTH) sets the month in the year value for the alarm interrupt. The month is stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The ALARMMONTH register is shown in Figure 15 and described in Table 14.

**Figure 15. Alarm Month Register (ALARMMONTH)**

| 31 16 |
|---|
| Reserved |
| R-0 |

| 15 5 | 4 | 3 0 |
|---|---|---|
| Reserved | MONTH1 | MONTH0 |
| R-0 | R/W-0 | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14. Alarm Month Register (ALARMMONTH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Reserved. |
| 4 | MONTH1 | 0-1h | Most significant digit of months value. For MONTH1:MONTH0, JAN = 01 and DEC = 12. |
| 3-0 | MONTH0 | 0-9h | Least significant digit of months value. For MONTH1:MONTH0, JAN = 01 and DEC = 12. |

## *3.13 Alarm Year Register (ALARMYEAR)*

**NOTE:** Out of reset, the alarm year register (ALARMYEAR) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The alarm year register (ALARMYEAR) sets the year for the alarm interrupt. The year is stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The ALARMYEAR register is shown in Figure 16 and described in Table 15.

**Figure 16. Alarm Year Register (ALARMYEAR)**

| 31 16 |
|---|
| Reserved |
| R-0 |

| 15 8 | 7 4 | 3 0 |
|---|---|---|
| Reserved | YEAR1 | YEAR0 |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15. Alarm Years Register (ALARMYEARS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7-4 | YEAR1 | 0-9h | Most significant digit of years value. Range for YEAR1:YEAR0 is 00-99. |
| 3-0 | YEAR0 | 0-9h | Least significant digit of years value. Range for YEAR1:YEAR0 is 00-99. |

### 3.14 Control Register (CTRL)

**NOTE:** Out of reset, the control register (CTRL) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The control register (CTRL) is shown in Figure 17 and described in Table 16.

**Figure 17. Control Register (CTRL)**

| 31 – 8 |
|---|
| Reserved |
| R-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPLITPOWER | RTCDISABLE | SET32COUNTER | Reserved | HOURMODE | AUTOCOMP | ROUNDMIN | RUN |
| W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 16. Control Register (CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7 | SPLITPOWER | | Enable leakage-isolation circuitry used for isolated power schemes. **Write-only bit. Read-modify-write updates to the control register may unintentionally clear the SPLITPOWER bit because the bit always reads back 0.** |
| | | 0 | Disable split power. |
| | | 1 | Enable split power. |
| 6 | RTCDISABLE | | Disable RTC module and gate 32-kHz reference clock. RTC should only be disabled using this bit if the module will never be used and saving power is desired. |
| | | 0 | RTC is functional. |
| | | 1 | RTC is disabled and 32-kHz reference clock is gated. |
| 5 | SET32COUNTER | | Set the 32-kHz counter with the value stored in the compensation registers when the SET32COUNTER bit is set. RTC does not run normally when the SET32COUNTER bit is high so this bit should be toggled low-high-low when used. |
| | | 0 | No action. |
| | | 1 | Set 32-kHz counter with compensation register value. |
| 4 | Reserved | 0 | Reserved. |
| 3 | HOURMODE | | Enable 12-hour mode for HOURS and ALARMHOURS registers. |
| | | 0 | 24 Hour Mode (Valid hours 00-24). |
| | | 1 | 12 Hour Mode (Valid hours 00-12; MERIDIEM bit in HOURS and ALARMHOURS must be used to denote AM or PM). |
| 2 | AUTOCOMP | | Enable oscillator compensation mode. Compensation takes place once every hour. |
| | | 0 | Auto compensation is disabled. |
| | | 1 | Auto compensation is enabled. |
| 1 | ROUNDMIN | | Enable one-time rounding to nearest minute on next time register read. |
| | | 0 | Minute rounding disabled. |
| | | 1 | Rounding to nearest minute enabled. |
| 0 | RUN | | Stop the RTC 32-kHz counter. RTC should be stopped using this bit for stopping and resuming the counter. |
| | | 0 | Stop RTC counter. |
| | | 1 | Run RTC counter. |

## 3.15 Status Register (STATUS)

The STATUS register is shown in Figure 18 and described in Table 17.

**NOTE:** Out of reset, the STATUS register is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

**Figure 18. Status Register (STATUS)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | ALARM | DAYEVT | HREVT | MINEVT | SECEVT | RUN | BUSY |
| R-1 | | R/W1C-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; W1C = Write 1 to clear bit; -*n* = value after reset

**Table 17. Status Register (STATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 1 | Reserved |
| 6 | ALARM | | Indicates if an alarm interrupt has been generated. Write a 1 to clear the ALARM status. |
| | | 0 | No new alarm interrupt was generated. |
| | | 1 | New alarm interrupt was generated. |
| 5 | DAYEVT | | When the (TIMER = 1) in the INTERRUPTS register, DAYEVT indicates if the DAYS register incremented during the most recent time update. |
| | | 0 | DAYS register did not increment during the last time update. |
| | | 1 | DAYS register incremented during the last time update. |
| 4 | HREVT | | When the (TIMER = 1) in the INTERRUPTS register, HREVT indicates if the HOURS register incremented during the most recent time update. |
| | | 0 | HOURS register did not increment during the last time update. |
| | | 1 | HOURS register incremented during the last time update. |
| 3 | MINEVT | | When the (TIMER = 1) in the INTERRUPTS register, MINEVT indicates if the MINUTES register incremented during the most recent time update. |
| | | 0 | MINUTES register did not increment during the last time update. |
| | | 1 | MINUTES register incremented during the last time update. |
| 2 | SECEVT | | When the (TIMER = 1) in the INTERRUPTS register, SECEVT indicates if the SECONDS register incremented during the most recent time update. |
| | | 0 | SECONDS register did not increment during the last time update. |
| | | 1 | SECONDS register incremented during the last time update. |
| 1 | RUN | | Indicates if RTC is running or stopped. |
| | | 0 | RTC is stopped. |
| | | 1 | RTC is running. |
| 0 | BUSY | | Indicates if RTC is busy updating or is within 15 µs of updating the time and calendar registers. |
| | | 0 | RTC is free. The time, calendar, and control registers can be written to without contention. |
| | | 1 | RTC is or will soon be busy updating the time and calendar registers. |

### 3.16 Interrupt Register (INTERRUPT)

The INTERRUPT register is shown in Figure 19 and described in Table 18.

**NOTE:** Out of reset, the INTERRUPT register is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

**Figure 19. Interrupt Register (INTERRUPT)**

| 31 16 |
|---|
| Reserved |
| R-0 |

| 15 4 | 3 | 2 | 1 0 |
|---|---|---|---|
| Reserved | ALARM | TIMER | EVERY |
| R-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18. Interrupt Register (INTERRUPT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved. |
| 3 | ALARM | | Enable interrupt generation for when the alarm time and date match the current time and date |
| | | 0 | Alarm interrupt is disabled. |
| | | 1 | Alarm interrupt is enabled. |
| 2 | TIMER | | Enable periodic timer interrupt generation. Period is determined by the EVERY field. |
| | | 0 | Periodic timer interrupt is disabled. |
| | | 1 | Periodic timer interrupt is enabled. |
| 1-0 | EVERY | 0-3h | Selects the time period desired when periodic timer interrupts are enabled by the TIMER bit. |
| | | 0 | Second |
| | | 1h | Minute |
| | | 2h | Hour |
| | | 3h | Day |

### 3.17 Compensation (LSB) Register (COMPLSB)

**NOTE:** Out of reset, the compensation (LSB) register (COMPLSB) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The compensation (LSB) register (COMPLSB) works together with the COMPMSB register to set the hourly oscillator compensation value. The AUTOCOMP bit in the control register (CTRL) must be enabled for compensation to take place. The COMPLSB register is shown in Figure 20 and described in Table 19.

**Figure 20. Compensation (LSB) Register (COMPLSB)**

|  |  |  |  |
|---|---|---|---|
| 31 |  |  | 16 |
| Reserved |  |  |  |
| R-0 |  |  |  |
| 15 | 8 | 7 | 0 |
| Reserved |  | COMPLSB |  |
| R-0 |  | R/W-0 |  |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19. Compensations Register (COMPLSB) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7-0 | COMPLSB | 0-FFh | Lower bits of the 16-bit compensation value. The COMPMSB:COMPLSB register value is subtracted from the 32-kHz period. Compensation values are two's complement. The COMPMSB:COMPLSB value of 7F:FFh is not allowed. |

### 3.18 Compensation (MSB) Register (COMPMSB)

**NOTE:** Out of reset, the compensation (MSB) register (COMPMSB) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The compensation (MSB) register (COMPMSB) works together with the COMPLSB register to set the hourly oscillator compensation value. The AUTOCOMP bit in the control register (CTRL) must be enabled for compensation to take place. The COMPMSB register is shown in Figure 21 and described in Table 20.

**Figure 21. Compensation (MSB) Register (COMPMSB)**

|  |  |  |  |
|---|---|---|---|
| 31 |  |  | 16 |
| Reserved |  |  |  |
| R-0 |  |  |  |
| 15 | 8 | 7 | 0 |
| Reserved |  | COMPMSB |  |
| R-0 |  | R/W-0 |  |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20. Compensations Register (COMPMSB) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7-0 | COMPMSB | 0-FFh | Lower bits of the 16-bit compensation value. The COMPMSB:COMPLSB register value is subtracted from the 32-kHz period. Compensation values are two's complement. The COMPMSB:COMPLSB value of 7F:FFh is not allowed. |

### 3.19 Oscillator Register (OSC)

**NOTE:** Out of reset, the oscillator register (OSC) is write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The oscillator register (OSC) is shown in Figure 22 and described in Table 21.

**Figure 22. Oscillator Register (OSC)**

| 31 16 |
|---|
| Reserved |
| R-0 |

| 15 6 | 5 | 4 0 |
|---|---|---|
| Reserved | SWRESET | Reserved |
| R-0 | W-0 | R/W-7h |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 21. Oscillator Register (OSC) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved. |
| 5 | SWRESET | | Software reset. Always reads back 0. |
| | | 0 | No action. |
| | | 1 | Reset RTC module and registers (except for OSC and KICK*n*R registers). Registers must not be accessed for three 32-kHz reference periods after reset is asserted. |
| 4-0 | Reserved | 7h | Reserved. This field is writeable, but should only be programmed to the value of 7h. |

### *3.20 Scratch Registers (SCRATCH0-SCRATCH2)*

**NOTE:** Out of reset, the scratch registers (SCRATCH*n*) are write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6).

The scratch registers (SCRATCH*n*) are 32-bit general-purpose registers that have no effect on RTC functionality. They can be used by the user arbitrarily. The SCRATCH*n* register is shown in Figure 23 and described in Table 22.

**Figure 23. Scratch Registers (SCRATCH*n*)**

| 31 | 0 |
|---|---|
| SCRATCH*n* | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 22. Scratch Registers (SCRATCH*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SCRATCH*n* | 0-FFFF FFFFh | General-purpose 32-bit registers that have no effect on RTC functionality. |

### *3.21 Kick Registers (KICK0R, KICK1R)*

The kick registers (KICK*n*R) are used to enable and disable write protection on the RTC registers. Out of reset, the RTC registers are write-protected. To disable write protection, correct keys must be written to the KICK*n*R registers (see Section 2.6). The KICK*n*R register is shown in Figure 24 and described in Table 23.

**Figure 24. Kick Registers (KICK*n*R)**

| 31 | 0 |
|---|---|
| KICK*n* | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 23. Kick Registers (KICK*n*R) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | KICK*n* | 0-FFFF FFFFh | To disable RTC register write protection, the value of 83E7 0B13h must be written to KICK0R, followed by the value of 95A4 F1E0h written to KICK1R. RTC register write protection is enabled when any value is written to KICK0R. |

# Appendix A Revision History

Table 24 lists the changes made since the previous version of this document.

Table 24. Document Revision History

| Reference | Additions/Modifications/Deletions |
|---|---|
| Section 2.6 | Changed second sentence. |
| Table 2 | Deleted footnote. |
| Section 3.21 | Changed first sentence. |

# IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

| **Products** | | **Applications** | |
|---|---|---|---|
| Amplifiers | amplifier.ti.com | Audio | www.ti.com/audio |
| Data Converters | dataconverter.ti.com | Automotive | www.ti.com/automotive |
| DLP® Products | www.dlp.com | Communications and Telecom | www.ti.com/communications |
| DSP | dsp.ti.com | Computers and Peripherals | www.ti.com/computers |
| Clocks and Timers | www.ti.com/clocks | Consumer Electronics | www.ti.com/consumer-apps |
| Interface | interface.ti.com | Energy | www.ti.com/energy |
| Logic | logic.ti.com | Industrial | www.ti.com/industrial |
| Power Mgmt | power.ti.com | Medical | www.ti.com/medical |
| Microcontrollers | microcontroller.ti.com | Security | www.ti.com/security |
| RFID | www.ti-rfid.com | Space, Avionics & Defense | www.ti.com/space-avionics-defense |
| RF/IF and ZigBee® Solutions | www.ti.com/lprf | Video and Imaging | www.ti.com/video |
| | | Wireless | www.ti.com/wireless-apps |

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265