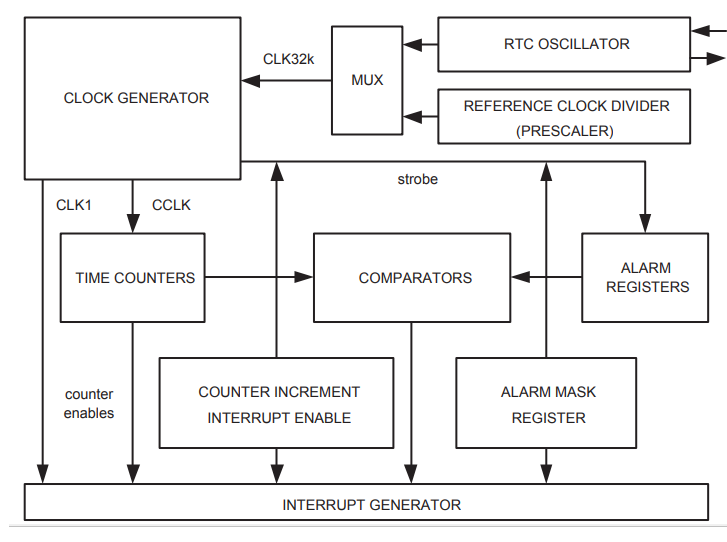
**Features**

* Measures the passage of time to maintain a calendar and clock
* Ultra Low Power design to support battery powered system
* Provides Seconds, Minutes, Hours, Day of Math, Month, Year, Day of Week and Day of Year.
* Dedicated 32kHz oscillator
* Dedicated power supply pin to a battery
* set of counters for measuring time when system power is on and off
* can be clocked by separate 32.768kHz oscillator or programmable prescale divider based on APB clock
* separate supply pin, VBAT
* Leap year works for 1901 to 2099



**Register Description**

**Interrupt Location Register (ILR - 0xE002 4000)**

* Specifies which blocks are generating interrupt.
* writing 1 clears the interrupt

|  |  |  |
| --- | --- | --- |
| **7:2 – Reserved** | **1 – RTCALF** | **0 - RTCCIF** |

* **RTCALF –** the alarm registers generated an interrupt
* **RTCCIF –** the Counter Increment Interrupt block generated an interrupt

**Clock Tick Counter Register (CTCR - 0xE002 4004)**

* contains the bits of clock divider counter.
* read only and cleared by Clock Control Register

|  |  |  |
| --- | --- | --- |
| **15 – Reserved** | **14:1 – Clock Tick Counter** | **1 - Reserved** |

* CTC counts 32768 clocks per second
* implemented as a 15-bit ripple counter
* If RTC driven by external 32.786 kHz, then the consecutive read of CTCR may be wrong because of ripple counting
* If RTC driven by PCLK, then the consecutive read of CTCR is correct.

**Clock Control Register (CCR - 0xE002 4008)**

* controls the operation of clock divide circuit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **7:5** | **4 – CLKSRC** | **3:2 – Test Enable** | **1 - CTCRST** | **0 – CLKEN** |
| Reserved | 1 – Clock Tick Counter takes its clock from Prescalar  0 – Clock Tick Counter takes its clock from external 32.768kHz oscillator. | Should be zero during normal operation | Clock Tick Counter is reset. | Clock Enable to enable the clock tick counter |

**Counter Increment Interrupt Register (CIIR - 0xE002 400C)**

* generate interrupt, every time counter is increment
* writing 1 into Interrupt Location Register(ILR[0]) clears

|  |  |  |
| --- | --- | --- |
| **Bits** | **Symbol** | **Description** |
| **0** | **IMSEC** | **1 –** increment of Second value generates an interrupt |
| **1** | **IMMIN** | **1 –** increment of Minute value generates an interrupt |
| **2** | **IMHOUR** | **1 –** increment of Hour value generates an interrupt |
| **3** | **IMDOM** | **1 –** increment of Day of Month value generates an interrupt |
| **4** | **IMDOW** | **1 –** increment of Day of Week value generates an interrupt |
| **5** | **IMDOY** | **1 –** increment of Day of Year value generates an interrupt |
| **6** | **IMMON** | **1 –** increment of Month value generates an interrupt |
| **7** | **IMYEAR** | **1 –** increment of Year value generates an interrupt |

**Alarm Mask Register (AMR - 0xE002 4010)**

* allows to mask any of the alarm registers
* **For alarm function, every non-masked alarm registers must match the corresponding time counter for an interrupt to be generated.**
* If all the mask bits are set, then the alarm is disabled.

|  |  |  |
| --- | --- | --- |
| **Bits** | **Symbol** | **Description** |
| **0** | **AMRSEC** | **1 –** Second value is not compared for alarm |
| **1** | **AMRMIN** | **1 –** Minute value is not compared for alarm |
| **2** | **AMRHOUR** | **1 –** Hour value is not compared for alarm |
| **3** | **AMRDOM** | **1 –** Day of Month value is not compared for alarm |
| **4** | **AMRDOW** | **1 –** Day of Week value is not compared for alarm |
| **5** | **AMRDOY** | **1 –** Day of Year value is not compared for alarm |
| **6** | **AMRMON** | **1 –** Month value is not compared for alarm |
| **7** | **AMRYEAR** | **1 –** Year value is not compared for alarm |

**Alarm register group**

* are used for comparison if unmasked and interrupt is generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Size** | **Description** | **Address** |
| **ALSEC** | 6 | Alarm value of seconds | 0xE0024020 |
| **ALMIN** | 6 | Alarm value of minutes | 0xE0024024 |
| **ALHOUR** | 5 | Alarm value of hours | 0xE0024028 |
| **ALDOM** | 5 | Alarm value of day of month | 0xE002402C |
| **ALDOW** | 3 | Alarm value of day of week | 0xE0024030 |
| **ALDOY** | 9 | Alarm value of day of year | 0xE0024034 |
| **ALMON** | 4 | Alarm value of months | 0xE0024038 |
| **ALYEAR** | 12 | Alarm value of years | 0xE002403C |

**Consolidated Time registers**

* to read time and data
* read only
* update can be done using Timer Counter Addresses

**Consolidated Time register 0 (CTIME0 - 0xE002 4014)**

|  |  |  |
| --- | --- | --- |
| **Bits** | **Symbol** | **Description** |
| **5:0** | **Seconds** | Second value in range of 0 to 59 |
| **13:8** | **Minutes** | Minute value in range of 0 to 59 |
| **20:16** | **Hours** | Hours value in range of 0 to 23 |
| **26:24** | **Day of Week** | Day of week value in range of 0 to 6 |

**Consolidated Time register 1 (CTIME1 - 0xE002 4018)**

|  |  |  |
| --- | --- | --- |
| **Bits** | **Symbol** | **Description** |
| **4:0** | **Day of Month** | Day of month value in range of 1 to 28,29,30, 31 |
| **11:8** | **Month** | Month value in range of 1 to 12 |
| **27:16** | **Year** | Year value in range of 0 to 4095 |

**Consolidated Time register 2 (CTIME2 - 0xE002 401C)**

|  |  |  |
| --- | --- | --- |
| **Bits** | **Symbol** | **Description** |
| **11:0** | **Day of Year** | Day of Year value in range of 1 to 365,366 |

**Time counter group**

* **The actual Counters can be read and written**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Counter** | **Size** | **Enabled by** | **Range** | **Address** |
| Second(SEC) | 6 | Clk1 | 0 to 59 | 0xE0024020 |
| Minute(MIN) | 6 | Second | 0 to 59 | 0xE0024024 |
| Hour(HOUR) | 5 | Minute | 0 to 23 | 0xE0024028 |
| Day of Month(DOM) | 5 | Hour | 1 to 28/29/30/31 | 0xE002402C |
| Day of Week(DOW) | 3 | Hour | 0 to 6 | 0xE0024030 |
| Day of Year(DOY) | 9 | Hour | 1 to 365/366 | 0xE0024034 |
| Month(MONTH) | 4 | Day of Month | 1 to 12 | 0xE0024038 |
| Year(YEAR) | 12 | Month or day of Year | 0 to 4095 | 0xE002403C |

**Prescaler Integer register (PREINT - 0xE002 4080)**

* integer portion of RTC prescale value
* PREINT = int(PCLK/32768) – 1
* PREINT >=0
* **12:0 bits**

**Prescaler Fraction register (PREFRAC - 0xE002 4084)**

* fractional portion of RTC prescale value
* PREFRAC =PCLK – ((PREINT + 1) x 32768)
* **14:0 bits**

**RTC usage:**

* ***VBAT*** pin must be at least 1.8V all times(may be connected to VDD)
* if RTC not used,
  + RTCX1 pin is connected to either VSS or VDD
  + RTCX2 pin is left floating
* Can used either 32.768 kHz oscillator @RTCX1-2 pins or APB clock (PCLK).
* A reference clock divider (prescalar), can generate 32.768 kHz from any PCLK value greater than or equal to 65.536 kHz.
* The reference clock divider consist of 13-bit integer counter (**PREINT**) and 15-bit fractional counter (**PREFRAC**).
  + **example calculation**
    - PCLK = 60MHz
    - PREINT = int(PCLK / 32768) -1 = 1830
    - PREFRAC = PCLK – ([PREINT + 1] x 32768) = 1792

**RTC External 32.765kHz Oscillator**

* Cap of around 22 pF

