32 bit RISC Microcontroller

TXZ Family

Reference Manual

Real Time Clock (RTC-A)

Revision 1.2

2019-05

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Contents

List of Figures

List of Tables

Preface

Related document

Document name
Exception
Clock Control and Operation Mode
Input/Output Ports
Product Information

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3:0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1], [XYZ2], [XYZ3] → [XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C …
  Example: ***[ADACR0], [ADBCR0], [ADCCR0] → [ADxCR0]***
  In case of channel, "x" means 0, 1, and 2…
  Example: ***[T32A0RUNA], [T32A1RUNA], [T32A2RUNA] → [T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3:0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> = 0x01 (hexadecimal), ***[XYZn]***<VW> = 1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value. In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

The Flash memory uses the Super Flash® technology under license from Silicon Storage Technology, Inc.
Super Flash® is registered trademark of Silicon Storage Technology, Inc.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Terms and Abbreviations

Some of abbreviations used in this document are as follows:

RTC	Real Time Clock

1. Outline

The real time clock (RTC) is a clock function to which even the calendar corresponding to a leap year can respond. At counting a low speed clock (fs), a 1Hz clock is generated and it operates. Clock correction function can adjust clock progress and delay due to low speed oscillator error. The alarm function can output a pulse from ALARM_N pin or output an interrupt request by detecting coincidence of the time and the clock which was set up beforehand.

Since the RTC operates with a low speed clock, it can operate during Low Power Consumption modes, such as IDLE, STOP1, and STOP2. Moreover, it can be made to return from a Low Power Consumption mode using the interrupt request of RTC.

The 1Hz clock which is generated by RTC can be outputted from a RTCOUT pin.

Table 1.1 Functional outline

Function classification	Function	A Functional Description or the range
Clock function	Clock source	Low speed clock (32.768 kHz)
	Clock	Respond to hour, minute and second. Selectable 12(am/ pm) and 24 hour display.
	Calendar	Respond to date, a day of the week, month, year and leap year.
Alarm function	Alarm	It can set up minute, hour, a day of the week, and a day.
	Interruption	An INTRTC interrupt request is output when the clock matches the date and time of the alarm register.
	Pulse output	Pulse is output from the ALARM_N terminal when the clock matches the date and time of the alarm register. (Note)
Periodic interruption function	Interruption	An INTRTC interrupt request is outputted to the frequency of 1Hz, 2Hz, 4Hz, 8Hz and 16Hz.
	Pulse output	The pulse is outputted to frequency 1Hz, 2Hz, 4Hz, 8Hz and 16Hz. from an ALARM_N pin. (Note)
Other functions	Clock correction	Correction-reference time: Select from 1, 10, 20, 30 seconds or 1 minute. Correction value: +255 to -256
	Protection of a clock correction functional register	A correction functional control register ***[RTCADJCTL]***, a correction value register ***[RTCADJDAT]***, and the correction value sign register ***[RTCADJSIGN]*** can be protected from writing by a protection register ***[RTCPROTECT]***.
	1Hz clock output	1Hz clock (duty 50%) is output from the RTCOUT pin.(Note)
	Reset register	A second counter and an alarm register are initialized.

Note: There are built-in products and a non-built-in product with a RTCOUT pin and an ALARM_N pin. Please refer to the reference manual "Product Information" for details.

2. Block Diagram

The Block Diagrams of RTC and a signal list are shown.

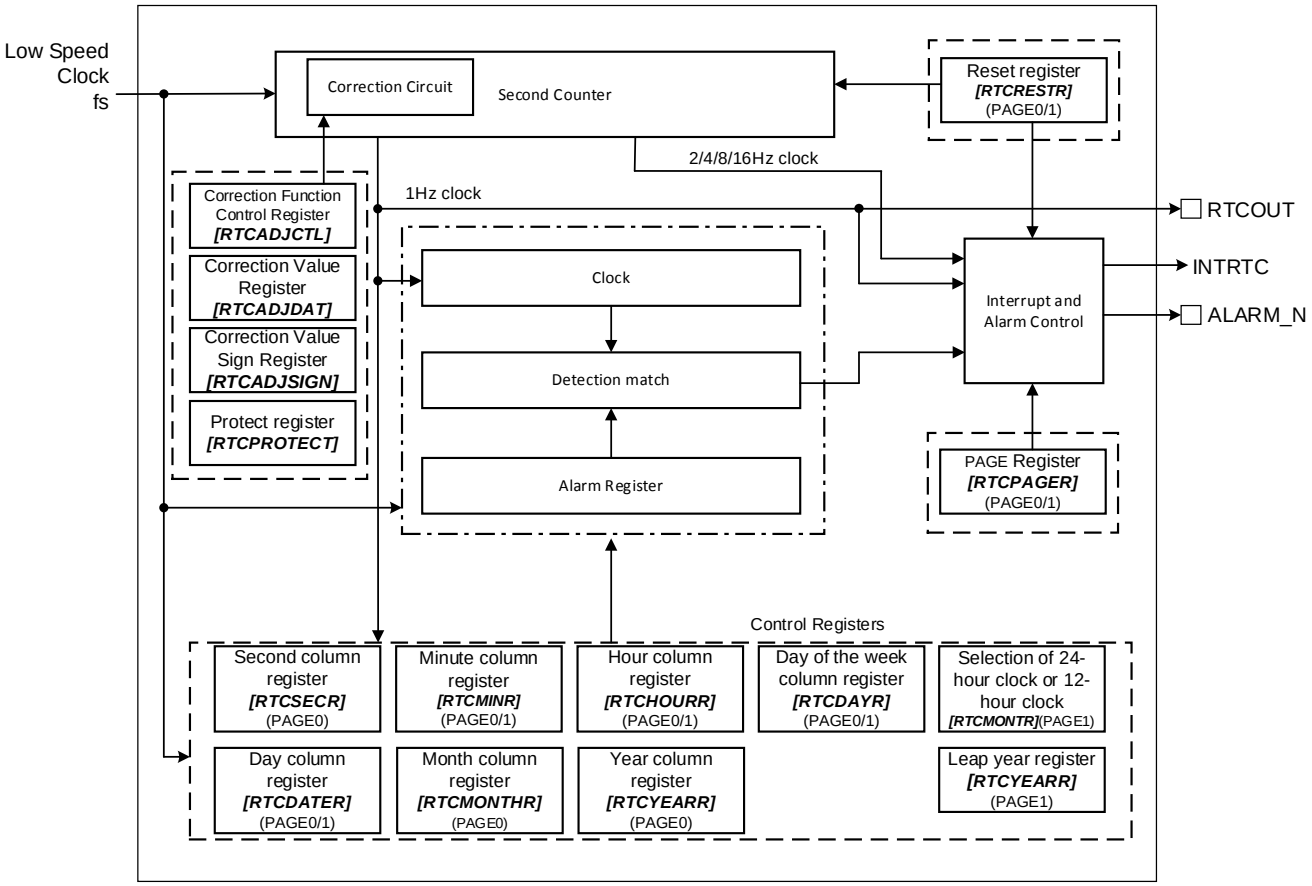

Figure 2.1 RTC Block Diagrams

Table 2.1 Signal list

No	Signal name	Signal name	I/O	Reference Manual
1	**fs**	Low speed clock	Input	Clock Control and Operation Mode
2	**ALARM_N**	Alarm signal	Output	Input/Output Ports Product Information
3	**INTRTC**	Interrupt signal	Output	Exception
4	**RTCOUT**	1Hz clock	Output	Input/Output Ports Product Information

3. Operational Description

It needs low speed clock (fs) for using RTC. Please refer to reference manual “Clock Control and Operation Mode” for clock control.

3.1. Clock supply

Even if asserted system reset after turn on power supply, RTC is counting the low speed clock fs. When change setting of RTC registers and use interruption and an Output (RTCOUT/ALARM_N), set a clock enabling bit applicable with the fsys supply on/off register A or B (***[CGFSYSENA], [CGFSYSENB]***) and fc supply on/off register (***[CGFCEN])*** as 1 (clock supply). The corresponding registers and the bit locations depend on a product. Some products do not have all registers. For the details, refer to “Clock Control and Operation Mode” in Reference manual.

3.2. Clock Function

For using clock function, after setting "0" to ***[RTCPAGER]***<PAGE> for specified clock functional register, set each register of second, minute, hour, the day of the week, day, month and year. And after set "1" to ***[RTCPAGER]***<PAGE> for specified alarm functional register, set data of leap year and 12hours/24hours. After that to set "1" to ***[RTCPAGER]***<ENATMR>.

To set "1" to ***[RTCRESTR]***<RSTTMR> for reset second counter.

- Christian era year digits

  This RTC only has the last two digits of the year digits. The year following 99 is 00 years. When dealing with the digits of the year in the Christian era, please manage the upper two digits on the system side
- Leap year:

  A leap year is divisible by 4 excluding a year divisible by 100; the year divisible by 100 is not considered to be a leap year. Any year divisible by 400 is a leap year. This product is considered the year divisible by 4 to be a leap year and does not take into account the above exceptions. It needs adjustments for the exceptions.

3.2.1. Writing clock data

The clock register has a temporary buffer. The data is written into a temporary buffer. And data is transmitted to a clock register synchronizing with the count-up timing of a second counter by counting the low speed clock. Since the CPU is operating asynchronously with the low speed clock, it is necessary to avoid and write in the timing of transfer. The correct data can't be written in the timing of the writing and transfer timing to a clock register overlap.

(1) Using 1Hz interrupt

Enable of 1Hz interrupt, the 1Hz interrupt is generated by being synchronized with counting up of the second counter. When data is written in the time between 1Hz interrupt and subsequent 1-second count, it completes correctly.

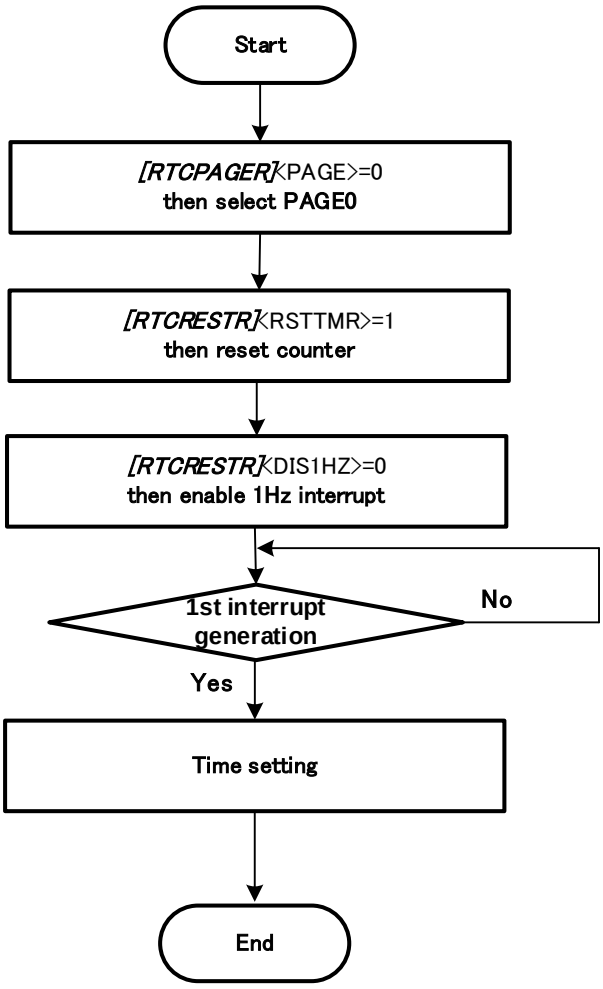

Figure 3.1 Flowchart of use 1Hz interruption

(2) Resetting the second counter

After writing "1" to ***[RTCRESTR]***<RSTTMR> for resetting the second counter, By writing the clock by the next second counter's count up, the expected value can be written

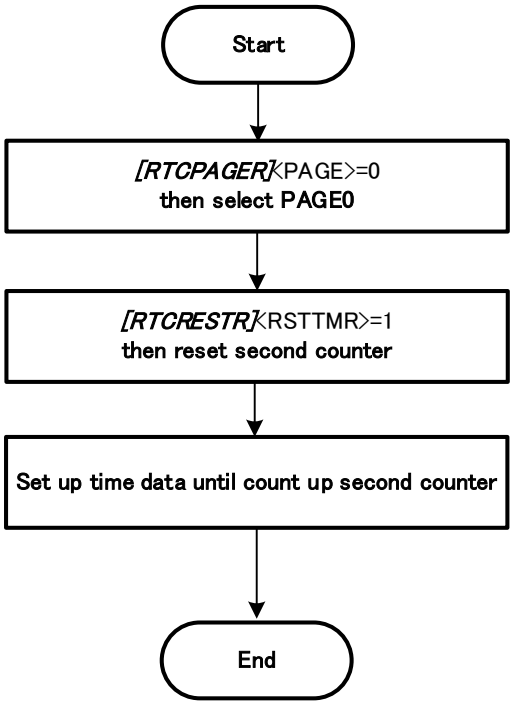

Figure 3.2 Flowchart of write data

(3) Disabling the clock

Writing "0" to ***[RTCPAGER]***<ENATMR> disables clock operation including a carry.
Stop the clock after the 1Hz interrupt. The second counter keeps counting. Set the clock again and enable the clock within one second before next 1Hz interrupt.

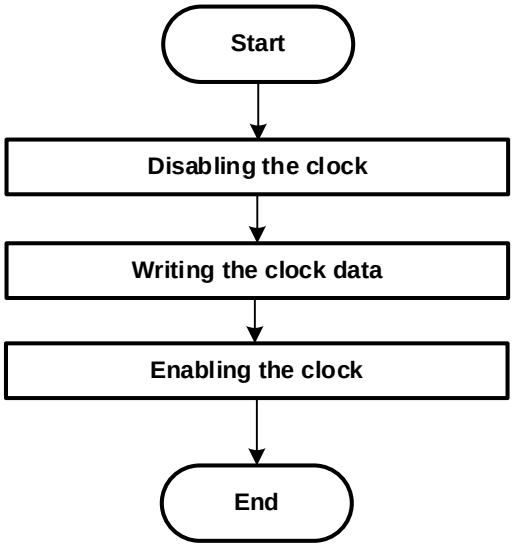

Figure 3.3 Flowchart of the disabling clock

3.2.2. Reading clock data

A clock register changes synchronizing with the low speed clock. Since the CPU is operating asynchronously with the low speed clock, in order to read a clock register correctly, it needs to avoid and read the timing from which a clock register changes.

(1) Using 1Hz interrupt

Enable of 1Hz interruption, the 1Hz interrupt is generated being synchronized with counting up of the second counter. Wait for a 1 Hz interrupt and read out the clock register before the next second counter counts up, so it can be read correctly

(2) Using pair reading

To ensure correct data reading, read the clock data twice as shown below. A pair of data read successively needs to match.

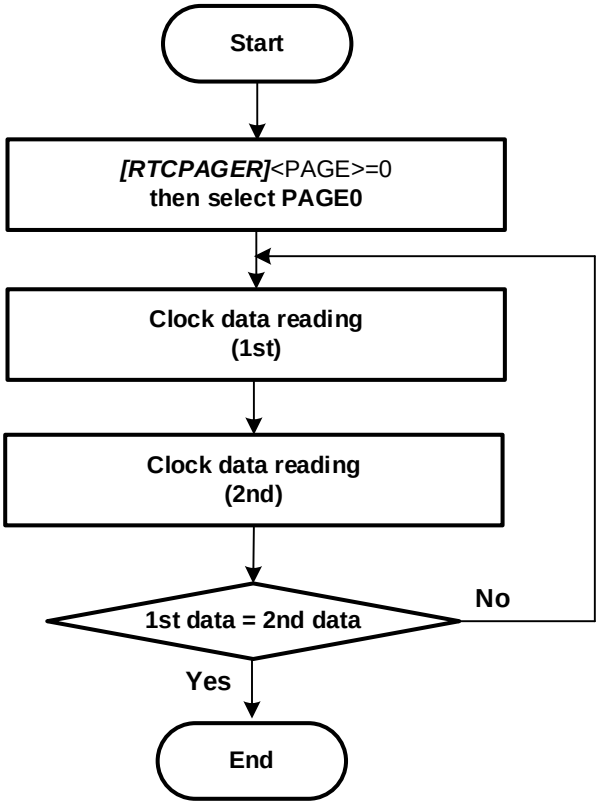

Figure 3.4 Flowchart of the clock data reading

3.2.3. Clock Correction Function

The clock correction function can precisely adjust the deviation of the clock.

3.2.3.1. Setting for Clock Correction Function

RTC counts the low speed clock by second counter. 1-second is generated by counting fs (32768Hz) 32768 times.
The clock correction function adjusts the number of counts of T2 that is a 1-second of the correction reference time (Tall). The correction reference time is selected from 1, 10, 20, 30 seconds or 1 minute with ***[RTCADJCTL]***<AJSEL>. A count value of T2 can be adjustable from 32768-255 to 32768+256 with ***[RTCADJDAT]***<ADJDAT>.

Table 3.1 Symbol and Description

Symbol	Item	Description
Tall	Correction reference time	Select from 1, 10, 20, 30 seconds or 1 minute with ***[RTCADJCTL]***<AJSEL>.
T1	1 second	Counts fs 32768 times
T2	Count correction	Adjust a count value with ***[RTCADJDAT]***<ADJDAT> , ***[RTCADJSIGN]*** by plus/minus 32768 counts

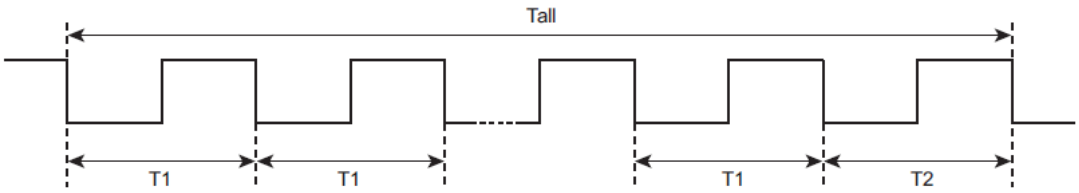

Figure 3.5 Clock Correction

3.2.3.2. Write protection for Clock Correction Function

The correction function related register, ***[RTCADJCTL]***, ***[RTCADJDAT]*** and ***[RTCADJSIGN]***, can be disabled with the ***[RTCPROTECT]*** register. In the initial state, ***[RTCPROTECT]*** is "0xC1" and write enable. If ***[RTCPROTECT]*** is set to a value other than "0xC1", ***[RTCADJCTL]***, ***[RTCADJDAT]*** and ***[RTCADJSIGN]*** will be write disable.

3.2.4. 1Hz Clock Output Function

RTCOUT pin outputs 1Hz clock. This clock is adjusted to operate on a 50% duty ratio. If the clock correction function is used, a duty ratio may be varied due to the error corrections.

In order to output 1Hz clock from a RTCOUT pin, it is necessary to select a RTCOUT output by port setup. Please refer to reference manual "Input/Output Ports" for details.

3.2.5. Entering the Low Power Consumption Mode

When making a transition to the low power consumption mode where the system clock stops after setting the clock register, correcting the clock, and resetting the clock, be sure to follow either of the following procedures.

(1) After changing the clock setting registers, or setting the ***[RTCPAGER]***<ADJUST> bit , ***[RTCRESTR]***<RSTTMR> bit, after waiting for 1-second interrupt, shift to low power consumption mode.

(2) After changing the clock setting registers, setting the ***[RTCPAGER]***<ADJUST> bit or setting the ***[RTCRESTR]***<RSTTMR> bit, after confirming that the read request has been executed by reading the values of <ADJUST> and <RSTTMR>, it shifts to the low power consumption mode.

3.3. Alarm function and Interrupt for alarm

3.3.1. Alarm setting

To use the alarm function, set ***[RTCPAGER]*** <PAGE> to "1" and specify the alarm register, then set the minute, hour, day of the week, day of the alarm, then set ***[RTCPAGER]*** <ENAALM> to "1".

By setting "1" to ***[RTCRESTR]*** <RSTALM>, initialize the minutes, hours, days of the week, and days of the alarm function register.

After initialization, it will be 00 minute, 00 hour, 01 day, Sunday.

When setting the alarm function register, if setting values of minute, hour, day of the week, and day are all set to "1", that item will not be compared with the clock

For example, if the alarm day digit is set to "0x111111" and the alarm day digit is set to "0x111" after initialization, a match of alarm and clock is detected every day at noon (12:00).

The above alarm works in synchronization with the low speed clock (fs). When the CPU is operating at high speed clock, a maximum of one clock delay at fs (about 30μs) may occur for the time register setting to become valid.

Note: ***[RTCPAGER]***<ENAALM> must be changed when ***[RTCPAGER]***<INTENA>=0. And it is prohibition that <ENAALM> and <INTENA> make a setting change same timing.

3.3.2. Alarm output

RTC outputs "Low" pulse from the ALARM_N pin at the same timing of INTRTC interrupt is generated.

Alarm output can be output only the product which has ALARM_N pin.

In order to output from ALARM_N pin, it is necessary to select ALARM output by port setup. Please refer to reference manual “Input/Output Ports” for details.

3.3.3. Interrupt setting

An INTRTC interrupt request output is enabled by setting "1" to ***[RTCPAGER]*** <INTENA>. If the time and the clock which were set as the alarm register matched in this state, an INTRTC interrupt request will be outputted.

A setup of INTRTC interruption is shown in Table 3.2.

An INTRTC interrupt request Output is controlled via INTIF. Refer to the "Exception" of a reference manual for the details of INTIF.

Table 3.2 Set up of INTRTC interruption

[RTCRESTR] <DIS1HZ>	*[RTCRESTR]* <DIS2HZ>	*[RTCRESTR]* <DIS4HZ>	*[RTCRESTR]* <DIS8HZ>	*[RTCRESTR]* <DIS16HZ>	*[RTCPAGER]* <ENAALM>	Interruption
1	1	1	1	1	1	ALARM
0	1	1	1	1	0	1Hz
1	0	1	1	1	0	2Hz
1	1	0	1	1	0	4Hz
1	1	1	0	1	0	8Hz
1	1	1	1	0	0	16Hz
Others						Interrupt not generated.

3.3.4. Periodic interrupt function

The periodic interruption function can generate INTRTC interruption with the cycle of 1 Hz, 2 Hz, 4 Hz, 8 Hz, or 16 Hz. Refer to “Table 3.2 Set up of INTRTC interruption“ for selection of an interruption cycle”.

3.3.5. Pulse output

The RTC outputs the "Low" pulse for low speed clock 1 cycle is outputted to the same timing of a periodic interrupt request. Refer to “Table 3.2 Set up of INTRTC interruption” for selection of an interruption cycle.

Alarm output can be output only the product which has ALARM_N pin.

In order to output from ALARM_N pin, it is necessary to select ALARM output by port setup. Please refer to reference manual “Input/Output Ports” for details.

4. Registers

4.1. Register List

The table below shows control registers and their addresses.
RTC has two functions, the clock(PAGE0) and alarm(PAGE1), which share some parts of registers.
The PAGE0/PAGE1 can be selected by setting ***[RTCPAGER]***<PAGE>.

Peripheral		Channel/Unit	Base address	
			TYPE 1	TYPE 2
Real Time Clock	RTC	-	0x400CC000	0x400E4800

Note: The base address type is different by products. Please refer to "Product Information" of the reference manual for the details.

Register name		Address(Base+)
Second column register (PAGE0)	***[RTCSECR]***	0x0000
Minute column register (PAGE0/1)	***[RTCMINR]***	0x0001
Hour column register (PAGE0/1)	***[RTCHOURR]***	0x0002
Reserved (Note1)	-	0x0003
Day of the week column register (PAGE0/1)	***[RTCDAYR]***	0x0004
Day column register (PAGE0/1)	***[RTCDATER]***	0x0005
Month column register (PAGE0)	***[RTCMONTHR]***	0x0006
Selection register of 24-hour,12-hour (PAGE1)		
Year column register (PAGE0)	***[RTCYEARR]***	0x0007
Leap year register (PAGE1)		
PAGE register (PAGE0/1)	***[RTCPAGER]***	0x0008
Reserved (Note1)	-	0x0009
Reserved (Note1)	-	0x000A
Reserved (Note1)	-	0x000B
Reset register (PAGE0/1)	***[RTCRESTR]***	0x000C
Reserved (Note1)	-	0x000D
Protect register	***[RTCPROTECT]***	0x000E
Correction Function Control Register	***[RTCADJCTL]***	0x000F
Correction Value Register	***[RTCADJDAT]***	0x0010
Correction Value sign Register	***[RTCADJSIGN]***	0x0011

Note1: "0x00" is read by reading the address. Writing is disregarded.
Note2: All the register of RTC must be accessed by 8-bit width.

Reset operation initializes the following registers.

-***[RTCPAGER]***<PAGE>, <ADJUST>, <INTENA>
-***[RTCRESTR]***
-***[RTCPROTECT]***

Other clock-related registers are not initialized by reset operation.
Before using the RTC, set the time, month, day, day of the week, year and leap year in the relevant registers.
Caution is required in setting clock data, adjusting seconds or resetting the clock.
Refer to "3.2.5 Entering the Low Power Consumption Mode" for more information.

Table 4.1 PAGE0 (clock function) registers

Symbol	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
[RTCSECR]	-	40s	20s	10s	8s	4s	2s	1s	Second column register
[RTCMINR]	-	40min	20min	10min	8min	4min	2min	1min	Minute column register
[RTCHOURR]	-	-	20hours PM/AM	10hours	8hours	4hours	2hours	1hour	Hour column register
[RTCDAYR]	-	-	-	-	-	Day of the week setting			Day of the week column register
[RTCDATER]	-	-	Day20	Day10	Day8	Day4	Day2	Day1	Day column register
[RTCMONTHR]	-	-	-	Oct.	Aug.	Apr.	Feb.	Jan.	Month column register
[RTCYEARR]	Year80	Year40	year20	Year10	Year8	Year4	Year2	Year1	Year column register (lower two columns)
[RTCPAGER]	Interrupt enable	-	-	Second adjust setting	Clock enable	Alarm enable	-	PAGE setting	PAGE register
[RTCRESTR]	1 Hz enable	16 Hz enable	Clock reset	Alarm reset	-	2 Hz enable	4 Hz enable	8 Hz enable	Reset register
[RTCPROTECT]	Protect code setting								Clock correction function protection register
[RTCADJCTL]	-	-	-	-	Correction reference time			Correction enable	Correction function Control register
[RTCADJDAT]	Correction value								Correction value register
[RTCADJSIGN]	-	-	-	-	-	-	-	+/-	Correction value sign register

Note1: Please cautions for ***[RTCSECR]***, ***[RTCMINR]***, ***[RTCHOURR]***, ***[RTCDAYR]***, ***[RTCDATER]***, ***[RTCMONTHR***], ***[RTCYEARR]*** of PAGE0 to write in. When you write in, please write in by the method explained in "3.2.1 Writing clock data".

Note2: Reading ***[RTCSECR]***, ***[RTCMINR]***, ***[RTCHOURR]***, ***[RTCDAYR]***, ***[RTCDATER]***, ***[RTCMONTHR]***, ***[RTCYEARR]*** of PAGE0 is updated asynchronously with CPU operation. When you read out, please read in by the method explained in "3.2.2Reading clock data".

Note3: It takes a maximum of 1s until the data written in ***[RTCSECR]***, ***[RTCMINR]***, ***[RTCHOURR]***, ***[RTCDAYR]***, ***[RTCDATER]***, ***[RTCMONTHR]***, ***[RTCYEARR]*** of PAGE0 is reflected in a register.

Table 4.2 PAGE1 (alarm function) registers

Symbol	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
[RTCSECR]	-	-	-	-	-	-	-	-	-
[RTCMINR]	-	40min	20min	10min	8min	4min	2min	1min	Minute column register
[RTCHOURR]	-	-	20hours PM/AM	10hours	8hours	4hours	2hours	1hour	Hour column register
[RTCDAYR]	-	-	-	-	-	Day of the week setting			Day of the week column register
[RTCDATER]	-	-	Day20	Day10	Day8	Day4	Day2	Day1	Day column register
[RTCMONTHR]	-	-	-	-	-	-	-	24/12	24-hour clock mode register
[RTCYEARR]	-	-	-	-	-	-	Leap-year setting		Leap-year mode register
[RTCPAGER]	Interrupt enable	-	-	Second adjust setting	Clock enable	Alarm enable	-	PAGE setting	PAGE register
[RTCRESTR]	1 Hz enable	16 Hz enable	Clock reset	Alarm reset	-	2 Hz enable	4 Hz enable	8 Hz enable	Reset register
[RTCPROTECT]	Protect code								Clock correction function register protection register
[RTCADJCTL]	-	-	-	-	Correction reference time			Correction enable	Correction function Control register
[RTCADJDAT]	Correction value								Correction value register
[RTCADJSIGN]	-	-	-	-	-	-	-	+/-	Correction value sign register

Note1: When write in ***[RTCMINR]***,***[RTCHOURR]***,***[RTCDAYR]***,***[RTCDATER]*** of PAGE1 must be written after write "0" to ***[RTCPAGER]***<ENAALM> and in the state of alarm disable.

Note2: Please cautions for ***[RTCMONTHR***], ***[RTCYEARR]*** of PAGE1 to write in. When you write in, please write in by the method explained in "3.2.1Writing clock data".

Note3: It takes a maximum of 1s until the data written in ***[RTCYEARR]*** of PAGE1 is reflected in a register.

Note4: Reading ***[RTCMINR]***, ***[RTCHOURR]***, ***[RTCDAYR]***, ***[RTCDATER]***, *[**RTCMONTHR]*, ***[RTCYEARR]*** of PAGE1 captures the current state.

Note5: ***[RTCYEARR]*** of PAGE1 is updated asynchronously with CPU operation. When you read out, please read in by the method explained in "3.2.2Reading clock data".

4.2. Detailed Description of Control Register

4.2.1. *[RTCSECR]* (Second column register (PAGE0)

Bit	Bit Symbol	After reset	Type	Function
7	-	0	R	Read as 0.
6:0	SE[6:0]	Undefined	R/W	Setting digit register of second 0000000: 00s 0010000: 10s 0100000: 20s 0000001: 01s 0010001: 11s 0000010: 02s 0010010: 12s 0110000: 30s 0000011: 03s 0010011: 13s 0000100: 04s 0010100: 14s 1000000: 40s 0000101: 05s 0010101: 15s 0000110: 06s 0010110: 16s 1010000: 50s 0000111: 07s 0010111: 17s 0001000: 08s 0011000: 18s 0001001: 09s 0011001: 19s 1011001: 59s

Note1: The setting other than listed above is prohibited.
Note2: Please cautions for ***[RTCSECR]*** to write in. When you write in, please write in by the method explained in "3.2.1Writing clock data".
Note3: ***[RTCSECR]*** is updated asynchronously with CPU operation. It must be read by the method explained in "3.2.2Reading clock data".
Note4: It takes a maximum of 1s until the data written in ***[RTCSECR]*** is reflected in a register.

4.2.2. *[RTCMINR]* (Minute column register (PAGE0/1))

Bit	Bit Symbol	After reset	Type	Function
7	-	0	R	Read as 0.
6:0	MI[6:0]	Undefined	R/W	Setting digit register of Minutes. 0000000: 00min 0010000: 10min 0100000: 20min 0000001: 01min 0010001: 11min | 0000010: 02min 0010010: 12min 0110000: 30min 0000011: 03min 0010011: 13min | 0000100: 04min 0010100: 14min 1000000: 40min 0000101: 05min 0010101: 15min | 0000110: 06min 0010110: 16min 1010000: 50min 0000111: 07min 0010111: 17min | 0001000: 08min 0011000: 18min | 0001001: 09min 0011001: 19min 1011001: 59min 1111111: Don't compare Minutes at alarm function.

Note1: The setting other than listed above is prohibited.
Note2: Please cautions for ***[RTCMINR]*** of PAGE0 to write in. When you write in, please write in by the method explained in "3.2.1 Writing clock data".
Note3: ***[RTCMINR]*** of PAGE0 is updated asynchronously with CPU operation. It must be read by the method explained in "3.2.2 Reading clock data".
Note4: It takes a maximum of 1s until the data written in ***[RTCMINR]*** of PAGE0 is reflected in a register.
Note5: Write ***[RTCMINR]*** of PAGE1 with ***[RTCPAGER]***<ENAALM> being "0".

4.2.3. *[RTCHOURR]* (Hour column register (PAGE0/1))

(1) 24-hour clock mode (***[RTCMONTHR]***<MO0>=1)

Bit	Bit Symbol	After reset	Type	Function
7:6	-	0	R	Read as 0.
5:0	HO[5:0]	Undefined	R/W	Setting digit register of Hour. 000000: 0 o'clock 010000: 10 o'clock 100000: 20 o'clock 000001: 1 o'clock 010001: 11 o'clock 100001: 21 o'clock 000010: 2 o'clock 010010: 12 o'clock 100010: 22 o'clock 000011: 3 o'clock 010011: 13 o'clock 100011: 23 o'clock 000100: 4 o'clock 010100: 14 o'clock 000101: 5 o'clock 010101: 15 o'clock 000110: 6 o'clock 010110: 16 o'clock 000111: 7 o'clock 010111: 17 o'clock 001000: 8 o'clock 011000: 18 o'clock 001001: 9 o'clock 011001: 19 o'clock 111111: Don't compare time digits in alarm function (PAGE1).

Note1: The setting other than listed above is prohibited.
Note2: Please cautions for ***[RTCHOURR]*** of PAGE0 to write in. When you write in, please write in by the method explained in "3.2.1 Writing clock data".
Note3: ***[RTCHOURR]*** of PAGE0 is updated asynchronously with CPU operation. It must be read by the method explained in "3.2.2 Reading clock data".
Note4: It takes a maximum of 1s until the data written in ***[RTCHOURR]***of PAGE0 is reflected in a register.
Note5: Write ***[RTCHOURR]*** of PAGE1 with ***[RTCPAGER]***<ENAALM> being "0".

(2) 12-hour clock mode (***[RTCMONTHR]***<MO0> =0)

Bit	Bit Symbol	After reset	Type	Function
7:6	-	0	R	Read as 0.
5:0	HO[5:0]	Undefined	R/W	Setting digit register of Hour. (AM) (PM) 000000: 0 o'clock 100000: 0 o'clock 000001: 1 o'clock 100001: 1 o'clock 000010: 2 o'clock 100010: 2 o'clock 000011: 3 o'clock 100011: 3 o'clock 000100: 4 o'clock 100100: 4 o'clock 000101: 5 o'clock 100101: 5 o'clock 000110: 6 o'clock 100110: 6 o'clock 000111: 7 o'clock 100111: 7 o'clock 001000: 8 o'clock 101000: 8 o'clock 001001: 9 o'clock 101001: 9 o'clock 010000: 10 o'clock 110000: 10 o'clock 010001: 11 o'clock 110001: 11 o'clock 111111: Don't compare hour at alarm function.

Note1: The setting other than listed above is prohibited.
Note2: Please cautions for ***[RTCHOURR]*** of PAGE0 to write in. When you write in, please write in by the method explained in "3.2.1Writing clock data".
Note3: ***[RTCHOURR]*** of PAGE0 is updated asynchronously with CPU operation. It must be read by the method explained in "3.2.2Reading clock data".
Note4: It takes a maximum of 1s until the data written in ***[RTCHOURR]***of PAGE0 is reflected in a register.
Note5: Write ***[RTCHOURR]*** of PAGE1 with ***[RTCPAGER]***<ENAALM> being "0".

4.2.4. *[RTCDAYR]* (Day of the week column register (PAGE0/1))

Bit	Bit Symbol	After reset	Type	Function
7:3	-	0	R	Read as 0.
2:0	WE[2:0]	Undefined	R/W	Setting digit register of the day of the week. 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Don't compare day of the week at alarm function.

Note1: The setting other than listed above is prohibited.
Note2: Please cautions for ***[RTCDAYR]*** of PAGE0 to write in. When you write in, please write in by the method explained in "3.2.1Writing clock data".
Note3: ***[RTCDAYR]*** of PAGE0 is updated asynchronously with CPU operation. It must be read by the method explained in "3.2.2Reading clock data".
Note4: It takes a maximum of 1s until the data written in ***[RTCDAYR]***of PAGE0 is reflected in a register.
Note5: Write ***[RTCDAYR]*** of PAGE1 with ***[RTCPAGER]***<ENAALM> being "0".

4.2.5. *[RTCDATER]* (Day column register (PAGE0/1))

Bit	Bit Symbol	After reset	Type	Function
7:6	-	0	R	Read as 0.
5:0	DA[5:0]	Undefined	R/W	Setting digit register of day. 000000: N/A 010000: 10th day 100000: 20th day 000001: 1st day 010001: 11th day 100001: 21th day 000010: 2nd day 010010: 12th day 100010: 22th day 000011: 3rd day 010011: 13th day 100011: 23th day 000100: 4th day 010100: 14th day 100100: 24th day 000101: 5th day 010101: 15th day 100101: 25th day 000110: 6th day 010110: 16th day 100110: 26th day 000111: 7th day 010111: 17th day 100111: 27th day 001000: 8th day 011000: 18th day 101000: 28th day 001001: 9th day 011001: 19th day 101001: 29th day 110000: 30th day 110001: 31th day 111111: Don't compare day digits with alarm function.

Note1: The setting other than listed above is prohibited.
Note2: Do not set for non-existent days (e.g. 30th Feb.).
Note3: Please cautions for ***[RTCDATER]*** of PAGE0 to write in. When you write in, please write in by the method explained in "3.2.1Writing clock data".
Note4: ***[RTCDATER]*** of PAGE0 is updated asynchronously with CPU operation. It must be read by the method explained in "3.2.2Reading clock data".
Note5: It takes a maximum of 1s until the data written in ***[RTCDATER]*** of PAGE0 is reflected in a register.
Note6: Write ***[RTCDATER]*** of PAGE1 with ***[RTCPAGER]***<ENAALM> being "0".

4.2.6. *[RTCMONTHR]* (Month column register (PAGE0))

Bit	Bit Symbol	After reset	Type	Function
7:5	-	0	R	Read as 0.
4:0	MO[4:0]	Undefined	R/W	Setting digit register of Month. 00001: January 00111: July 00010: February 01000: August 00011: March 01001: September 00100: April 10000: October 00101: May 10001: November 00110: June 10010: December

Note1: The setting other than listed above is prohibited.
Note2: Please cautions for ***[RTCMONTHR]*** of PAGE0 to write in. When you write in, please write in by the method explained in "3.2.1Writing clock data".
Note3: ***[RTCMONTHR]*** of PAGE0 is updated asynchronously with CPU operation. It must be read by the method explained in "3.2.2Reading clock data".
Note4: It takes a maximum of 1s until the data written in ***[RTCMONTHR]***of PAGE0 is reflected in a register.

4.2.7. *[RTCMONTHR]* (Selection register of 24-hour,12-hour (PAGE1))

Bit	Bit Symbol	After reset	Type	Function
7:1	-	0	R	Read as 0.
0	MO0	Undefined	R/W	24-hour/12-hour selection 0: 12-hour 1: 24-hour

Note: Please cautions for ***[RTCMONTHR]*** of PAGE1 to write in. When you write in, please write in by the method explained in "3.2.1Writing clock data".

4.2.8. *[RTCYEARR]* (Year column register (PAGE0))

Bit	Bit Symbol	After reset	Type	Function
7:0	YE[7:0]	Undefined	R/W	Setting digit register of Year. 00000000: 00year 0 0010000: 10years 01100000: 60years 00000001: 01year | | 00000010: 02years 00100000: 20years 01110000: 70years 00000011: 03years | | 00000100: 04years 00110000: 30years 10000000: 80years 00000101: 05years | | 00000110: 06years 01000000: 40years 10010000: 90years 00000111: 07years | | 00001000: 08years 01010000: 50years | 00001001: 09years | 10011001: 99years

Note1: The setting other than listed above is prohibited.
Note2: Please cautions for ***[RTCYEARR]*** of PAGE0 to write in. When you write in, please write in by the method explained in "3.2.1Writing clock data".
Note3: ***[RTCYEARR]*** of PAGE0 is updated asynchronously with CPU operation. It must be read by the method explained in "3.2.2Reading clock data".
Note4: It takes a maximum of 1s until the data written in ***[RTCYEARR]***of PAGE0 is reflected in a register.

4.2.9. *[RTCYEARR]* (Leap year register (PAGE1))

Bit	Bit Symbol	After reset	Type	Function
7:2	-	0	R	Read as 0.
1:0	LEAP[1:0]	Undefined	R/W	Leap year settings 00: A leap year 01: one year after a leap year 10: two years after a leap year 11: three years after a leap year

Note1: Please cautions for ***[RTCYEARR]*** of PAGE1 to write in. When you write in, please write in by the method explained in "3.2.1Writing clock data".
Note2: ***[RTCYEARR]*** of PAGE1 is updated asynchronously with CPU operation. It must be read by the method explained in "3.2.2Reading clock data".
Note3: It takes a maximum of 1s until the data written in ***[RTCYEARR]***of PAGE0 is reflected in a register.

4.2.10. *[RTCPAGER]* (PAGE register (PAGE0/1))

Bit	Bit Symbol	After reset	Type	Function
7	INTENA	0	R/W	INTRTC 0:Disable 1:Enable
6:5	-	0	R	Read as 0.
4	ADJUST	0	R	0: ADJUST no request 1: ADJUST requested If "1" is read, it indicates that ADJUST is being executed. If "0" is read, it indicates that the execution is finished.
			W	0: - 1: Sets ADJUST request Adjusts seconds. The request is sampled when the second counter counts up. If the time elapsed is between 0 and 29 seconds, the second counter is cleared to "0". If the time elapsed is between 30 and 59 seconds, the minute. the counter is carried and the second counter is cleared to "0".
3	ENATMR	Undefined	R/W	Clock 0: Disable 1: Enable
2	ENAALM	Undefined	R/W	ALARM 0: Disable 1: Enable
1	-	0	R	Read as 0.
0	PAGE	0	R/W	PAGE selection 0:Selects PAGE0 1:Selects PAGE1

Note1: A read-modify-write operation cannot be performed.
Note2: Please change the setting of <ENATMR>, <ENAALM> with <INTENA> being "0". Make sure not to set them at the same time (make sure that there is time lag between clock enable and interrupt enable).
Note3: It takes a maximum of fs 1 clock until the data written in <ADJUST>, <ENATMR>, <ENAALM> are reflected in a register.

4.2.11. *[RTCRESTR]* (Reset register (PAGE0/1)

Bit	Bit Symbol	After reset	Type	Function
7	DIS1HZ	1	R/W	1 Hz Interrupt 0: Enable 1: Disable
6	DIS16HZ	1	R/W	16 Hz Interrupt 0: Enable 1: Disable
5	RSTTMR	0	R	0: No reset request 1: RESET requested If "1" is read, it indicates that RESET is being executed. If "0" is read, it indicates that the execution is finished.
			W	0: - 1: Second counter reset Resets the second counter. The request is sampled using low-speed clock. Please wait until <RSTTMR>=0.
4	RSTALM	0	R/W	0: - 1: Alarm reset Initializes alarm registers (Minute column, hour column, day column and day of the week column) as follows. MInute:00, Hour:00, Day:01, Day of the week: Sunday
3	-	0	R	Read as 0.
2	DIS2HZ	1	R/W	2 Hz Interrupt 0: Enable 1: Disable
1	DIS4HZ	1	R/W	4 Hz Interrupt 0: Enable 1: Disable
0	DIS8HZ	1	R/W	8 Hz Interrupt 0: Enable 1: Disable

Note1: A read-modify-write operation cannot be performed.
Note2: <DIS1HZ>,<DIS2HZ>,<DIS4HZ>,<DIS8HZ>,<DIS16HZ> must be written in ***[RTCPAGER]*** <INTENA>=0 state.
Note3: Update <RSTALM> while ***[RTCPAGER]***<ENAALM> is "0".
Note4: It takes a maximum of fs 1clock until the data written in <RSTTMR> is reflected in a register.

4.2.12. *[RTCPROTECT]* (Protect register)

Bit	Bit Symbol	After reset	Type	Function
7:0	PROTECT[7:0]	0xC1	R/W	Clock correction function register protection 0xC1: Write enable. other than 0xC1: Write disable. In the initial state, ***[RTCPROTECT]*** is "0xC1" and write enable. If ***[RTCPROTECT]*** is set to a value other than "0xC1", ***[RTCADJCTL]*** , ***[RTCADJDAT]*** and ***[RTCADJSIGN]*** registers writing cannot be performed.

4.2.13. *[RTCADJCTL]* (Correction Function Control Register)

Bit	Bit Symbol	After reset	Type	Function
7:4	-	0	R	Read as 0.
3:1	AJSEL[2:0]	Undefined	R/W	Correction reference time setting 000: 1 second 001: 10 seconds 010: 20 seconds 011: 30 seconds 100: 1 minute 101 to 111: Reserved Set the reference time for correction.
0	AJEN	Undefined	R/W	Correction function control 0: Disabled 1: Enabled

4.2.14. *[RTCADJDAT]* (Correction Value Register)

Bit	Bit Symbol	After reset	Type	Function
7:0	ADJDAT[7:0]	Undefined	R/W	Correction value Sign of correction value is plus. 00000000: No correction 00000001: 32768 + 1 00000010: 32768 + 2 | 11111110: 32768 + 254 11111111: 32768 + 255 Sign of correction value is minus. 00000000: 32768 - 256 00000001: 32768 - 255 | 11111110: 32768 - 2 11111111: 32768 - 1

4.2.15. *[RTCADJSIGN]* (Correction Value Sign Register)

Bit	Bit Symbol	After reset	Type	Function
7:1	-	0	R	Read as 0.
0	ADJSIGN	Undefined	R/W	Sign of correction value 0: Plus(+) 1: Minus(-)

5. Example of setting

Example1**:** Clock setting/Alarm setting

		7	6	5	4	3	2	1	0	
[RTCPAGER]	←	0	0	0	0	1	1	0	0	Enables Clock and alarm
[RTCPAGER]	←	1	0	0	0	1	1	0	0	Enables interrupt

Example2**:**The following is an example program for outputting an alarm from the ALARM_N pin at noon (12:30) on Monday 5th.

		7	6	5	4	3	2	1	0	
[RTCPAGER]	←	0	0	0	0	1	1	0	0	Disables Interrupt
[RTCPAGER]	←	0	0	0	0	1	0	0	1	Disables alarm, sets PAGE1
[RTCRESTR]	←	1	1	0	1	0	1	1	1	Initializes alarm
[RTCDAYR]	←	0	0	0	0	0	0	0	1	Monday
[RTCDATER]	←	0	0	0	0	0	1	0	1	5th day
[RTCHOURR]	←	0	0	0	1	0	0	1	0	Sets 12 o'clock
[RTCMINR]	←	0	0	1	1	0	0	0	0	Sets 30 min
[RTCPAGER]	←	0	0	0	0	1	1	0	0	Enables alarm
[RTCPAGER]	←	1	0	0	0	1	1	0	0	Enables interrupts

Example3: Example of program for generating interrupt at 16 Hz cycle

		7	6	5	4	3	2	1	0	
[RTCPAGER]	←	0	0	0	0	1	1	0	0	Disables Interrupt
[RTCPAGER]	←	0	0	0	0	1	0	0	0	Disables alarm
[RTCRESTR]	←	1	0	0	0	0	1	1	1	Enable 16 Hz cycle Interrupt
[RTCPAGER]	←	1	0	0	0	1	0	0	0	Enables interrupts

6. Revision History

Table 6.1 Revision History

Revision	Date	Description
1.0	2017-09-08	First Release
1.1	2018-04-06	- Preface Modified “A related reference manual” to “Related document” - 4.1 Register List Deleted ch0 from previous Table4.1 Modified: previous Table4.2 to Table 4.1 Modified: previous Table4.3 to Table 4.2 - Deleted: previous 4.2 Control Register - 4.2.12 ***[RTCPROTECT]*** Modified Bit Symbol name “RTCPROTECT” to “PROTECT”
1.2	2019-05-13	- Table 1.1 modified about "Description of Calendar Function". - 4.1 revised about initialized operation for registers by Reset. (deleted ***[RTCADJCTL]*** and ***[RTCADJDAT]***) - 4.1.10 Type W of bit 4 modified “don’t care” to “-“ - 4.1.11 Type R/W of bit 4 modified “don’t care” to “-“ Type W of bit 5 modified “don’t care” to “-“ - Revised "RESTRICTIONS ON PRODUCT USE".

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**