Application Processor Lite *ApP Lite*

TZ1000 Series

Reference Manual

MCU Timer/Counter

Revision 1.3

2018-02

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Table of Contents

List of Figures

List of Tables

Preface

This document provides the specification for the MCU Timer/Counter designed for the TZ1000 Series.

Intended Audience

This document is intended for the following users.

Driver software developers.

System designers

Conventions in this document

- The following notational conventions apply to numbers:
  Hexadecimal number: 0xABC
  Decimal number: 123 or 0d123 - Only when it should be explicitly indicated that the number is decimal.
  Binary number: 0b111 - It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- Low active signals are indicated with a name suffixed with "_N."
- A signal is asserted when it goes to its active level while it is deasserted when it goes to its inactive level.
- A set of multiple signals may be referred to as [m:n].
  Example: S[3:0] indicates four signals, S3, S2, S1 and S0, collectively.
- In the text, register names are enclosed in brackets *[ ]*.
  Example:*[ABCD]*
- A set of multiple registers, fields or bits of the same type may be described collectively using "n."
  Example: *[XYZ1]*, *[XYZ2]*, and *[XYZ3]* to *[XYZn]*
- A range of register bits are referred to as [m:n].
  Example: [3:0] indicates a range from bit 3 to bit 0.
- Values set in registers are indicated using either a hexadecimal or binary number.
- Example: *[ABCD]*.EFG = 0x01 (hexadecimal), *[XYZn]*.VW = 1 (binary)
- Words and bytes are defined as follows:
  Byte: 8 bits
  Halfword: 16 bits
  Word: 32 bits
  Doubleword: 64 bits
- Register bit attributes are defined as follows:
  R: Read-only
  W: Write-only
  W1C: Clear by write of 1 - A write of "1" clears the corresponding bit to 0.
  W1S: Set by write of 1 - A write of "1" sets the corresponding bit to 1.
  R/W: Read/Write
  R/W0C: Read/Clear by write of 0
  R/W1C: Read/Clear by write of 1
  R/W1S: Read/Set by write of 1
  RS/WC: Set by read/Clear by write - Set after a read and cleared after a data write.
- Registers only support word access unless otherwise specified.
- Any registers defined as Reserved in the text must not be rewritten. Also, any values read from such registers should not be used.
- Any bits for which default values are defined as "—" would return undefined values if read.
- When a data is written to a register containing both writable and read-only (R) bit fields, its default values should be written to read-only (R) bit fields. For any bit fields with default values defined as "—," refer to the definitions of the relevant register.
- Default values should be written to any reserved bit fields in a write-only register. For any bit fields with default values defined as "—," refer to the definitions of the relevant register.

Abbreviation

These specifications introduce a part of the abbreviation which they used

APB	Arm® AMBA® Advanced Peripheral Bus
RIS	Raw Interrupt Status
MIS	Masked Interrupt Status
PMU	Power Management Unit

1. Overview

This module is a 32-bit timer.

The feature is as follows.

- 2 timer channels. (Each channel has identical specification.)
- Compliant to the AMBA® (2.0) protocol. The APB slave.
- 32-bit down counter
- Changeable between 32-bit and 16-bit
- Operating clock, TIMCLK, synchronizes the APB clock, PCLK.
- Each channel can be controlled independently by each timer enable.
- 3 operation modes
  - -One shot timer mode
  - -Constant period timer mode
  - -Free run timer mode
- Prescaler to divide the TIMCLK frequency (the period: ×1, ×16, and ×256)
- An interrupt per channel and an interrupt of ORed output of ch0 and ch1
- Each interrupt is controlled by each channel interrupt enable bit.
- Setting value (Load Value) can be changed while the timer is operating by a background register.
- The timer interrupt output is controlled by a test register.

2. Block Diagram

The internal blocks of this module is shown in the following figures.

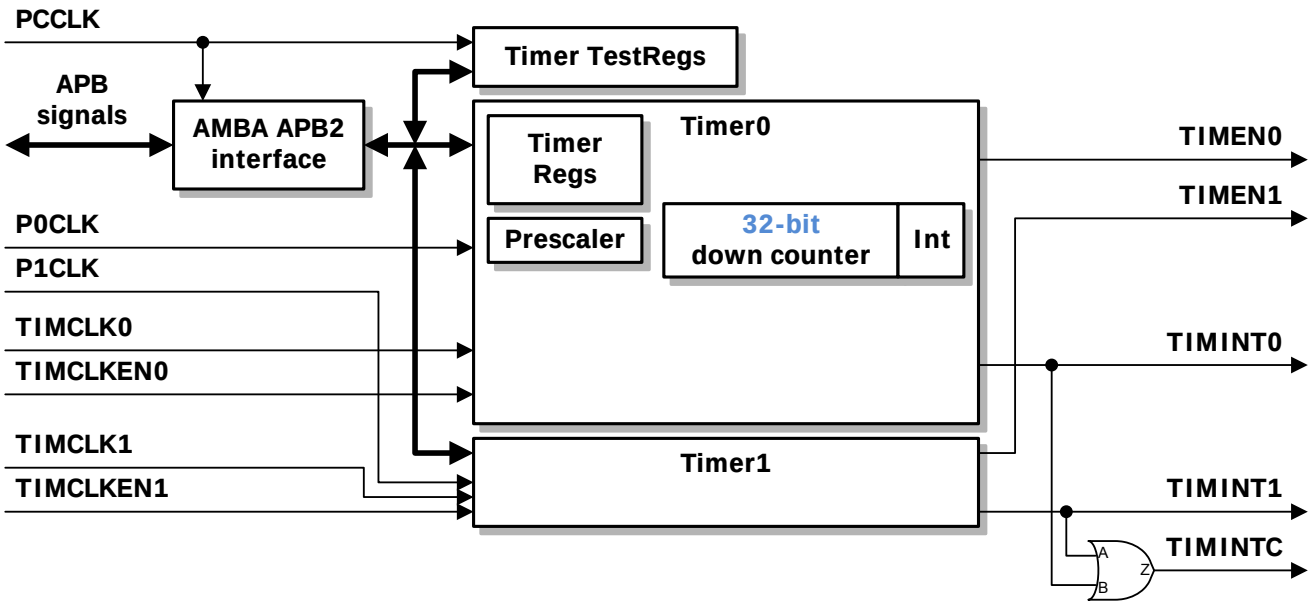

Figure 2.1 TMR internal block diagram

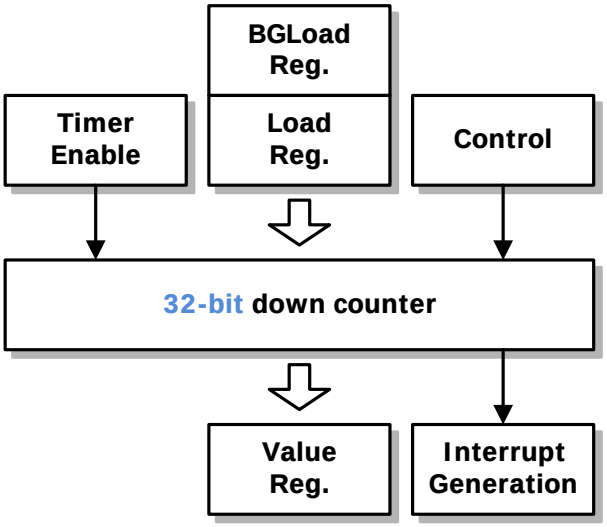

Figure 2.2 1 channel block diagram of TMR

The outlines of the internal blocks of the TMR are as follows.

-AMBA® APB2 interfaces
- This interface is the APB slave interface accessed by the CPU. It is used to access the registers in 4 KB address space.

-Test Regs
- The registers for the tests of the timer. These registers control the interrupt outputs.

-32-bit down counter
- The 32-bit counter which consists of the timer.

-Timer Regs
- The registers for the timer.

-Prescaler
- The prescaler which generates the clock to the 32-bit counter.

-Int
- Controls the interrupt signals.

3. Address Map

Table 3.1 MCU Timer/Counter Register Map

Register Name	Type	Width	Reset Value	Address Offset
TMR_TIMER0LOAD	RW	32	0x0000 0000	0x0000 0000
TMR_TIMER0VALUE	RO	32	0xFFFF FFFF	0x0000 0004
TMR_TIMER0CONTROL	RW	32	0x0000 0020	0x0000 0008
TMR_TIMER0INTCLR	RW	32	-	0x0000 000C
TMR_TIMER0RIS	RO	32	0x0000 0000	0x0000 0010
TMR_TIMER0MIS	RO	32	0x0000 0000	0x0000 0014
TMR_TIMER0BGLOAD	RW	32	0x0000 0000	0x0000 0018
TMR_TIMER1LOAD	RW	32	0x0000 0000	0x0000 0020
TMR_TIMER1VALUE	RO	32	0xFFFF FFFF	0x0000 0024
TMR_TIMER1CONTROL	RW	32	0x0000 0020	0x0000 0028
TMR_TIMER1INTCLR	RW	32	-	0x0000 002C
TMR_TIMER1RIS	RO	32	0x0000 0000	0x0000 0030
TMR_TIMER1MIS	RO	32	0x0000 0000	0x0000 0034
TMR_TIMER1BGLOAD	RW	32	0x0000 0000	0x0000 0038
TMR_TIMERITCR	RW	32	0x0000 0000	0x0000 0F00
TMR_TIMERITOP	RW	32	0x0000 0000	0x0000 0F04
TMR_TIMERPERIPHID0	RO	32	0x0000 0004	0x0000 0FE0
TMR_TIMERPERIPHID1	RO	32	0x0000 0018	0x0000 0FE4
TMR_TIMERPERIPHID2	RO	32	0x0000 0024	0x0000 0FE8
TMR_TIMERPERIPHID3	RO	32	0x0000 0000	0x0000 0FEC
TMR_TIMERPCELLID0	RO	32	0x0000 000D	0x0000 0FF0
TMR_TIMERPCELLID1	RO	32	0x0000 00F0	0x0000 0FF4
TMR_TIMERPCELLID2	RO	32	0x0000 0005	0x0000 0FF8
TMR_TIMERPCELLID3	RO	32	0x0000 00B1	0x0000 0FFC

4. Input and Output Signals

4.1. Function Signals and TMR signals

The function signals and the TMR signals are shown in the following table. The function signals are described in 5 Function.

Table 4.1 Signal comparison

function signal name	TMR signal name	Description
PCLK	PCCLK	Common bus clock
	P0CLK	ch0 bus clock
	P1CLK	ch1 bus clock
TIMCLK	TIMCLK0	ch0 counter clock
	TIMCLK1	ch1 counter clock
TIMCLKEN	TIMCLKEN0	ch0 clock enable
	TIMCLKEN1	ch1 clock enable
PRESETn	PCRESETn	Common bus reset
	P0RESETn	ch0 bus reset
	P1RESETn	ch1 bus reset
	TIM0RESETn	ch0 counter reset
	TIM1RESETn	ch1 counter reset
PADDR	PADDR	APB signals
PENABLE	PENABLE	
PSEL	PSEL	
PWDATA	PWDATA	
PWRITE	PWRITE	
PRDATA	PRDATA	
TIMINT	TIMINT0	ch0 interrupt
	TIMINT1	ch1 interrupt
	TIMINTC	Interrupt (combined)

5. Function

The function signal names are used in this chapter. For the function signal names, refer to Table 4.1.

5.1. Clock Operation

5.1.1. Clock and clock enable operation

The relation among the PCLK, the TIMCLK, and the TIMCLKEN is shown in the following figure (Each channel has its own clock names, but the following uses the function names to represent the channel signal names.) The count shows the count number of the base counter.

The TIMCLK and the TIMCLKEN are synchronized with the PCLK. The corresponding clocks of each channel are also synchronized as the same manner.

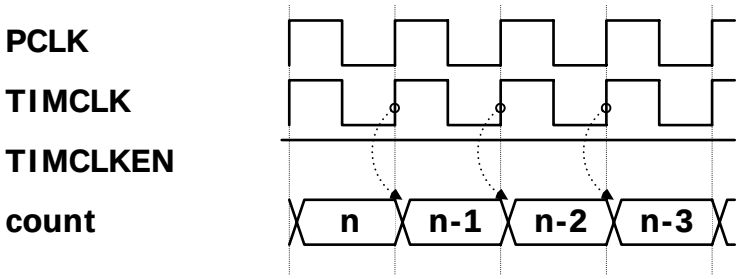

Figure 5.1 Case that PCLK and TIMCLK have the same frequency (TIMCLKEN = 1)

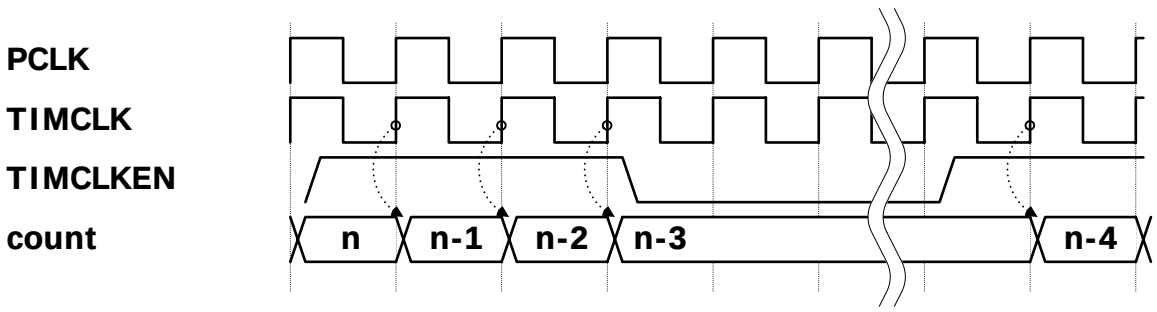

Figure 5.2 Case that PCLK and TIMCLK have the same frequency (TIMCLKEN is changing)

5.1.2. Prescaler operation

The block diagram of the Prescaler in the TMR is shown in the following figure. The prescaler enable output is used as the clock enable to the counter.

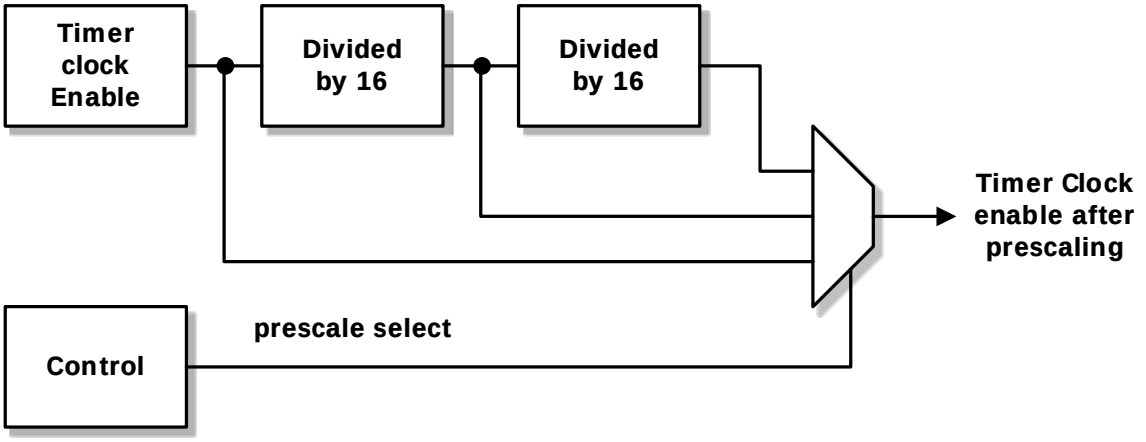

Figure 5.3 Block diagram of Prescaler

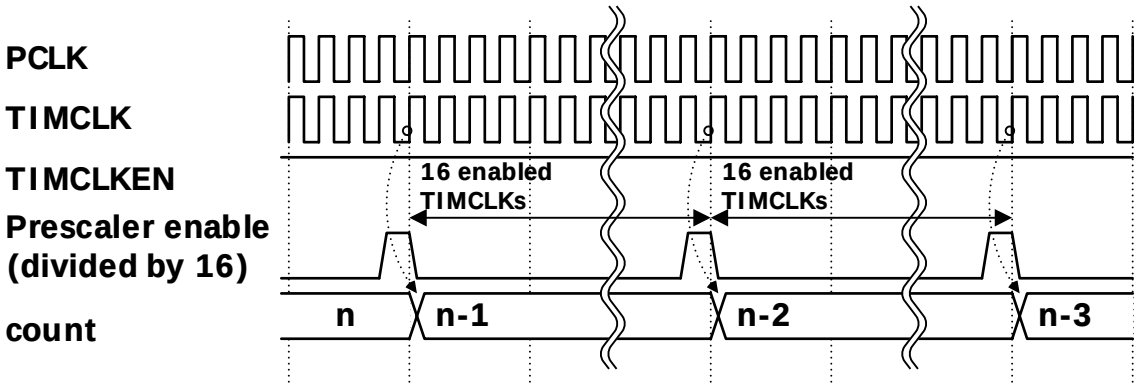

Figure 5.4 Example of base counter operation with dividing frequency by Prescaler

Interrupt generation interval is calculated with the following formula.

Interrupt generation interval = (**TIMERnLOAD** +1) × (**TIMCLKn** period: **TIMCLKENn**=1)
× (Prescaler setting value)
(**n** is a channel number: 0 or 1.)

Table 5.1 Interrupt generation interval setting (TIMCLK = 1 MHz, Prescaler ×1 and ×256)

Interrupt Interval	TIMERnLOAD Register (Prescaler ×1)		TIMERnLOAD Register (Prescaler ×256)	
	Hex	Decimal	Hex	Decimal
100 µs	0x00000063	99	—	—
500 µs	0x000001F3	499	—	—
1 ms	0x000003E7	999	0x00000003	3
5 ms	0x00001387	4999	0x00000013	19
100 ms	0x0001869F	99999	0x00000186	390
500 ms	0x0007A11F	499999	0x000007A1	1953
1 s	0x000F423F	999999	0x00000F42	3906
60 s	0x039386FF	59999999	0x00039385	234373
300 s	0x11E1A2FF	299999999	0x0011E1A2	1171874

Note: "—" shows that the setting is not available.
The following is the maximum value of the interrupt generation interval for each setting.
Prescaler ×1: 4294.967 s
Prescaler ×256: 1099511 s

Table 5.2 Interrupt generation interval setting (TIMCLK = 12 MHz, Prescaler x1 and x256)

Interrupt Interval	TIMERnLOAD Register (Prescaler x1)		TIMERnLOAD Register (Prescaler x256)	
	Hex	Decimal	Hex	Decimal
50 µs	0x00000257	599	—	—
100 µs	0x000004AF	1199	0x00000003	3
500 µs	0x0000176F	5999	0x00000017	23
1 ms	0x00002EDF	11999	0x0000002E	46
50 ms	0x000927BF	599999	0x00000927	2343
100 ms	0x00124F7F	1199999	0x0000124E	4686
500 ms	0x005B8D7F	5999999	0x00005B8D	23437
1 s	0x00B71AFF	11999999	0x0000B71A	46874
60 s	0x2AEA53FF	719999999	0x002AEA53	2812499
300 s	0xD693A3FF	3599999999	0x00D693A3	14062499

Note: "—" shows that the setting is not available.
The following is the maximum value of the interrupt generation interval for each setting.
Prescaler x1: 357.9139 s
Prescaler x256: 91625.969 s

5.2. Base Counter Operation

5.2.1. One-shot mode

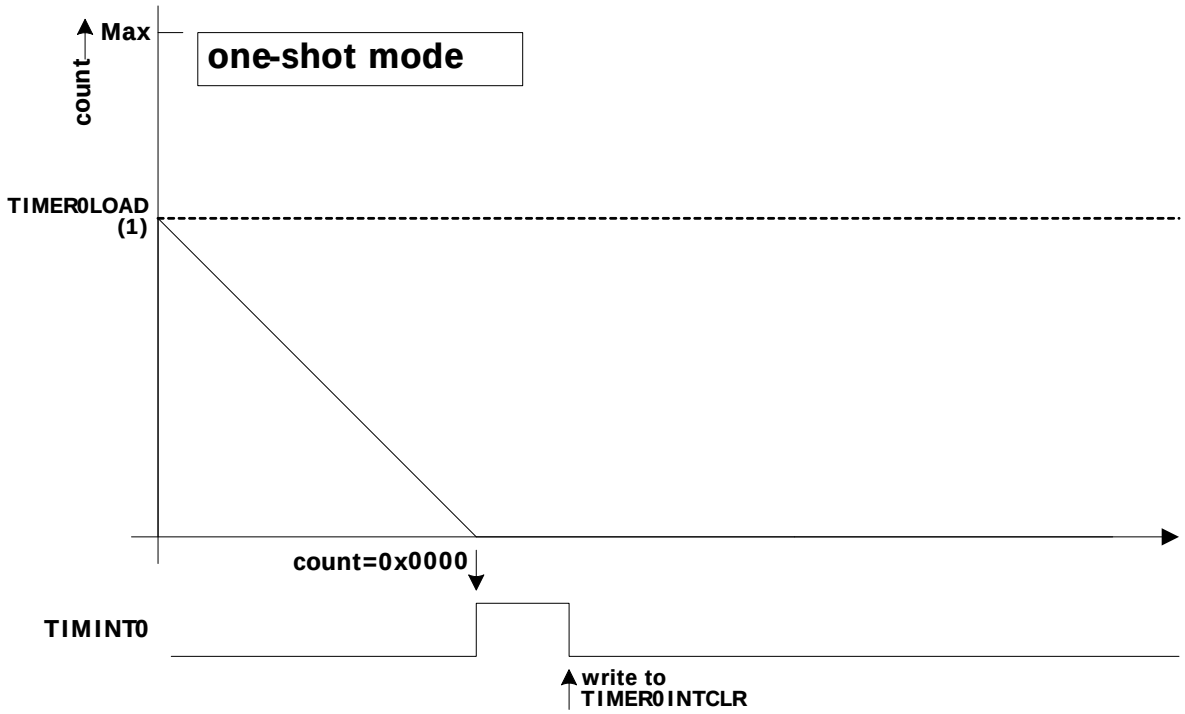

Figure 5.5 One-shot mode operation

In the one-shot mode, after the register is set and the timer enable is asserted, the counter starts to operate. The counter decrements from the value which is set to the load register *[TMR_TIMER0LOAD].* When the count value in the counter becomes 0x00000000 (the terminal count), the counter stops and the timer interrupt request is generated. After the interrupt process finishes in the CPU, the interrupt request is cleared by writing to the interrupt clear register *[TMR_TIMER0INTCLR]*.

In the following example, the register setting and the operation represent the channel 0 in the TMR. (The other examples later are also the same.)

Example

Register	Event	Reg. Value	Comments
TIMER0CONTROL		0x00000023	32-bit width, Timer disable, Interrupt disable ,Periodic, One-shot, x1
TIMER0LOAD		0x00001234	Set value :value(1)
TIMER0CONTROL		0x000000E3	Timer enable, Interrupt enable
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt (terminal count) occurs
TIMER0INTCLR		any value	Clear Timer Interrupt
	TIMINT0=0b0	—	

Note: When the one-shot operation finishes, the counter stops. The clock is, however, still supplied. Even though the dynamic clock gating is set (refer to 5.6), the clock does not stop. It is necessary to stop the clock that the timer should be disabled or the PMU register should be set to the clock stop.

5.2.2. Constant period timer wrapping mode

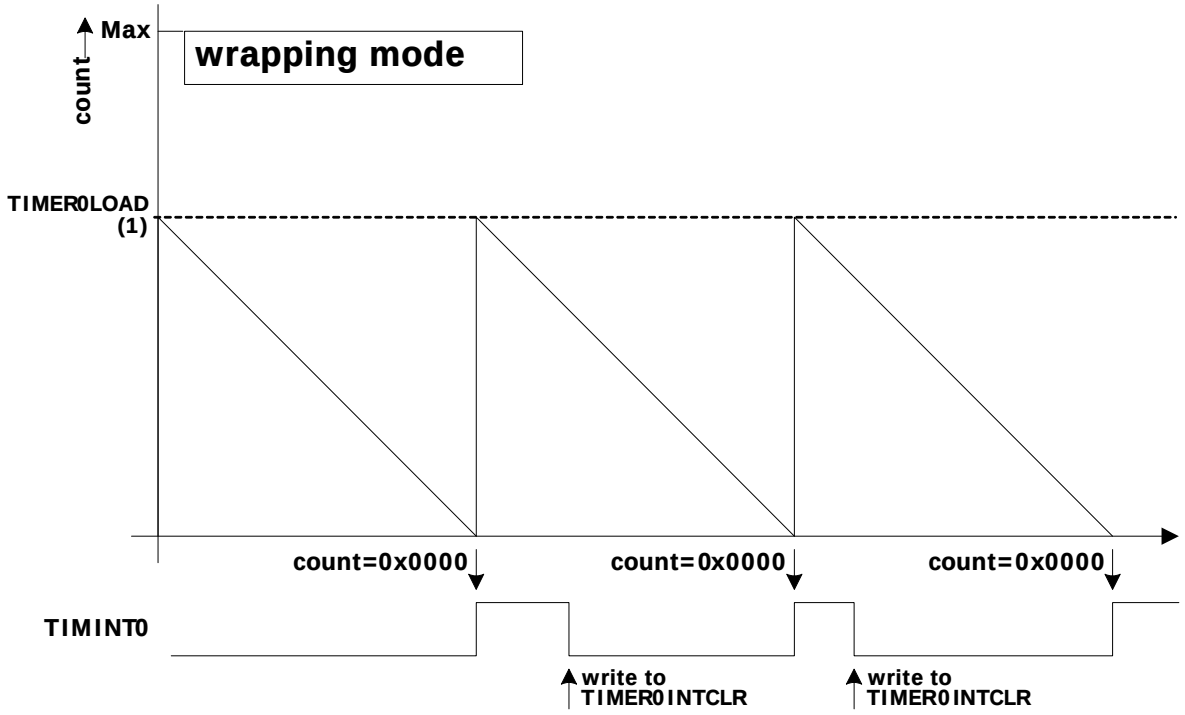

Figure 5.6 Constant period wrapping mode

In the constant period timer wrapping mode, after the register is set and the timer enable is asserted, the counter starts frequent counts. The counter decrements from the value which is set to the load register *[TMR_TIMER0LOAD].* When the count value in the counter becomes 0x00000000 (the terminal count), the counter stops and the timer interrupt request is generated. After the interrupt process finishes in the CPU, the interrupt request is cleared by writing to the interrupt clear register *[TMR_TIMER0INTCLR].*

The counter is set again to the value in the load register *[TMR_TIMER0LOAD]* and repeats the operation until the timer disable is asserted.

Example

Register	Event	Reg. Value	Comments
TIMER0CONTROL		0x00000020	32-bit width, Timer disable, Interrupt disable ,Periodic,Wrapping,x1
TIMER0LOAD		0x1111AAAA	Set value :value(1)
TIMER0CONTROL		0x000000E2	Timer enable, Interrupt enable
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt(terminal count) occurs
TIMER0INTCLR		any value	Clear Timer Interrupt
	TIMINT0=0b0	—	
:	:		
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt occurs
TIMER0INTCLR		any value	Clear Timer Interrupt
	TIMINT0=0b0	—	
:	:		
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt occurs

5.2.3. Free-run timer wrapping mode

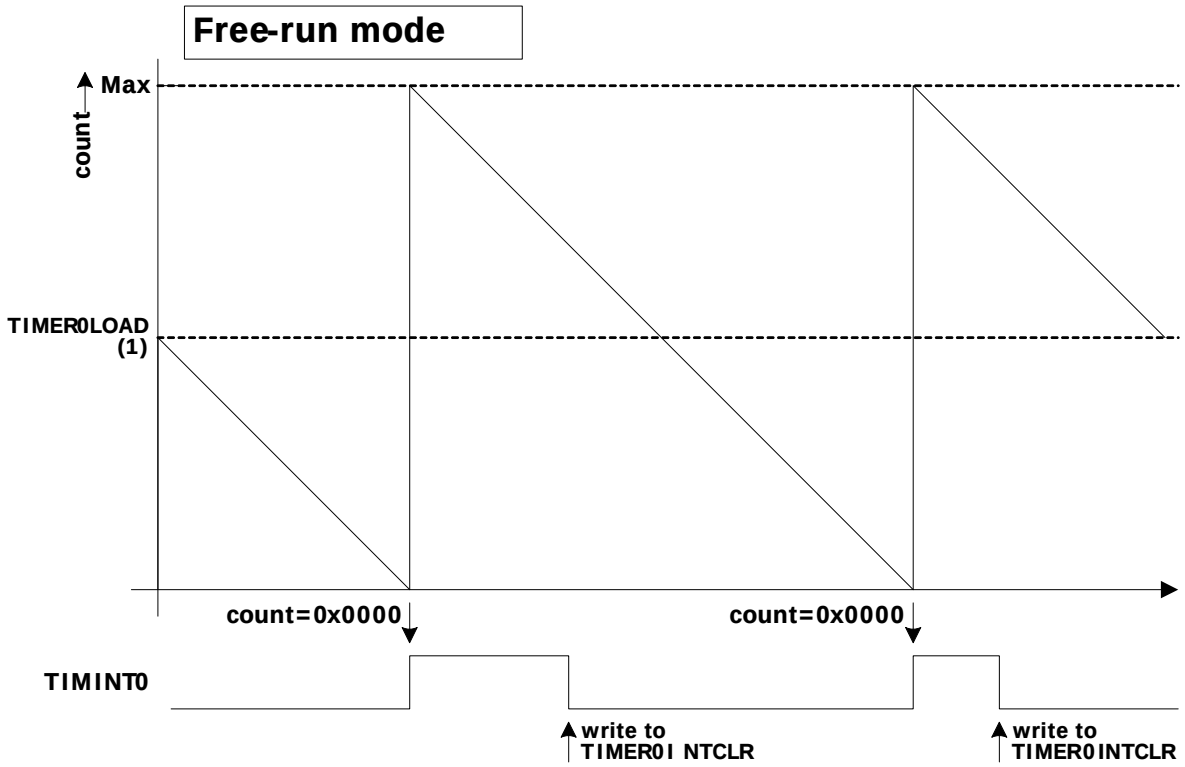

Figure 5.7 Free-run timer wrapping mode

In the free-run timer wrapping mode, after the register is set and the timer enable is asserted, the counter starts frequent counts. The counter decrements from the value which is set to the load register *[TMR_TIMER0LOAD].* When the count value in the counter becomes 0x00000000 (the terminal count), the counter stops and the timer interrupt request is generated. After the interrupt process finishes in the CPU, the interrupt request is cleared by writing to the interrupt clear register *[TMR_TIMER0INTCLR].*

The counter is set to the maximum count value regardless of the value in the load register *[TMR_TIMER0LOAD]* and repeats the operation until the timer disable is asserted.

Example

Register	Event	Reg. Value	Comments
TIMER0CONTROL		0x0002	32-bit width, Timer disable, Interrupt disable ,Free-run,Wrapping,x1
TIMER0LOAD		0x22224444	Set value :value(1)
TIMER0CONTROL		0x00A2	Timer enable, Interrupt enable
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt(terminal count) occurs
TIMER0INTCLR		any value	Clear Timer Interrupt
	TIMINT0=0b0	—	
:	:		
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt(terminal count) occurs
TIMER0INTCLR		any value	Clear Timer Interrupt
	TIMINT0=0b0	—	

5.2.4. Load register change operation

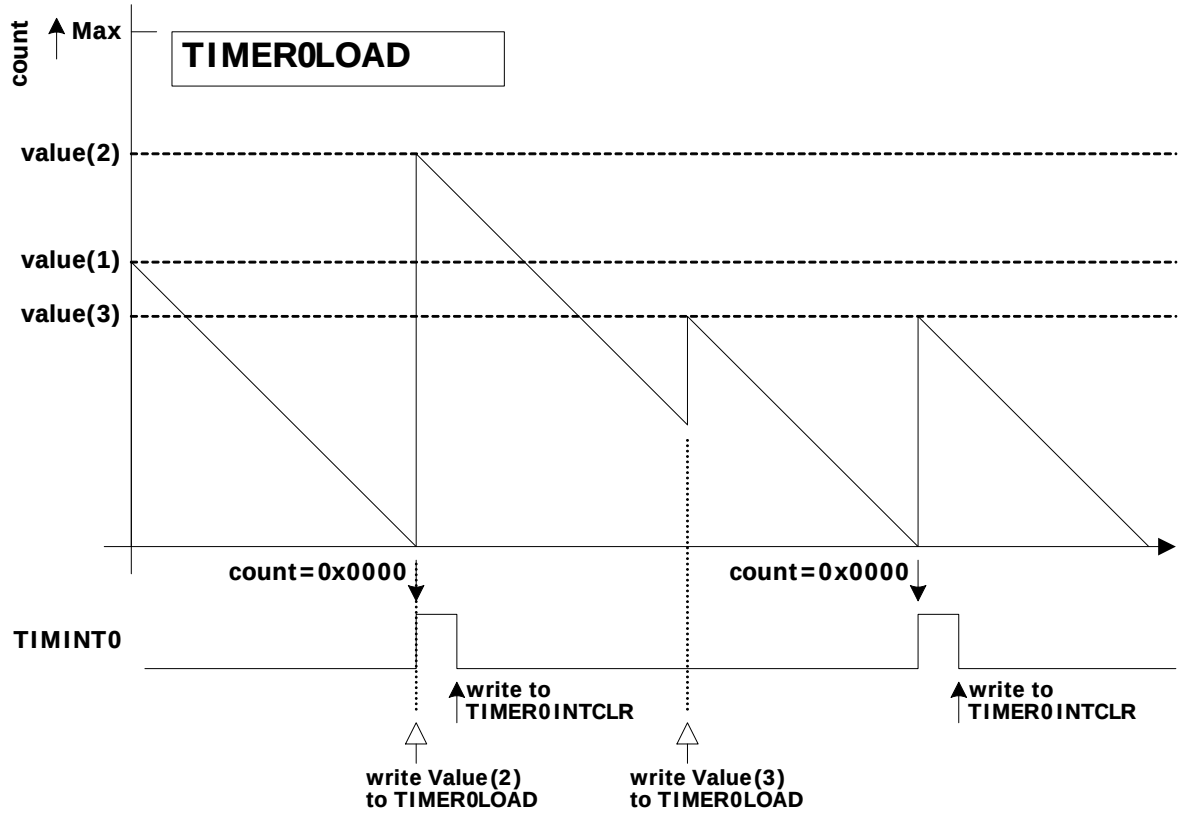

Figure 5.8 Load register change operation

In the constant period timer wrapping mode, when the counter is operating and the load register *[TMR_TIMER0LOAD]* is changed, the count value in the counter is set to the value immediately. This means that the counter value changes before the terminal count. To prevent this discontinuity, the setting of the load register *[TMR_TIMER0LOAD]* should be done in the following state.

- When the counter stops. Or,
- When the count value is 0x00000000 (the terminal count).

Example

Register	Event	Reg. Value	Comments
TIMER0CONTROL		0x0022	32-bit width, Timer disable, Interrupt disable ,Periodic,Wrapping,x1
TIMER0LOAD		0x11111111	Set value :value(1)
TIMER0CONTROL		0x00E2	Timer enable, Interrupt enable
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt(terminal count) occurs
TIMER0LOAD		0x11112222	Set value :value(2)
TIMER0INTCLR		any value	Clear Timer Interrupt
	TIMINT0=0b0	—	
TIMER0LOAD		0x11110000	Set value :value(3)
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt(terminal count) occurs
TIMER0INTCLR		any value	Clear Timer Interrupt
	TIMINT0=0b0	—	

5.2.5. Background load register change operation

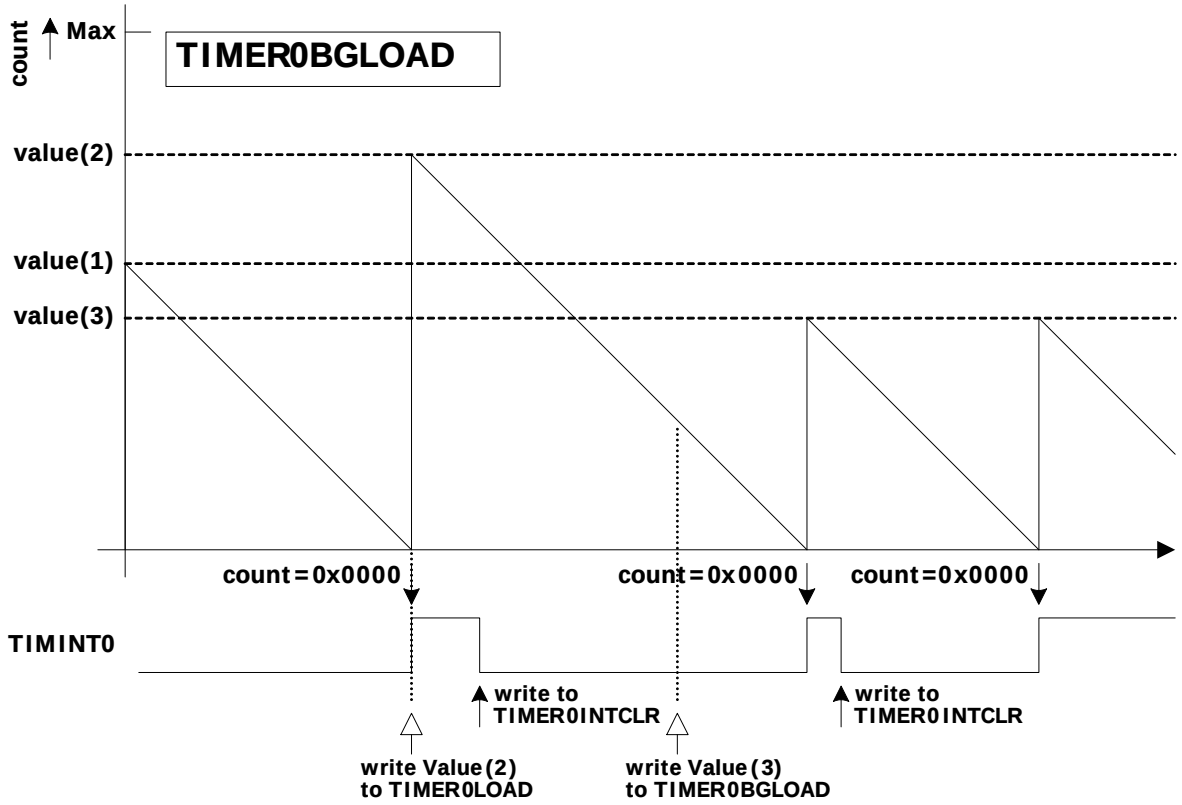

Figure 5.9 Background load register change operation

When the background load register *[TMR_TIMER0BGLOAD]* is used and the value is changed during the counter operation, the count value in the counter does not changed immediately. The *[TMR_TIMER0BGLOAD]* value is set to the counter when the count becomes 0x00000000 (the terminal count). So the discontinuity in the count does not occur.

Example

Register	Event	Reg. Value	Comments
TIMER0CONTROL		**0x0022**	32-bit width, Timer disable, Interrupt disable ,Periodic,Wrapping,x1
TIMER0LOAD		**0x11111111**	Set value **:**value(1)
TIMER0CONTROL		**0x00E2**	Timer enable, Interrupt enable
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt(terminal count) occurs
TIMER0LOAD		**0x11112222**	Set value **:**value(2)
TIMER0INTCLR		**any value**	Clear Timer Interrupt
	TIMINT0=0b0	—	
TIMER0BGLOAD		**0x11110000**	**Set value :value(3)**
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt(terminal count) occurs
TIMER0INTCLR		**any value**	Clear Timer Interrupt
	TIMINT0=0b0	—	
:	**:**		
	count=0x0000	—	terminal count occurs
	TIMINT0=0b1	—	Timer interrupt(terminal count) occurs

5.3. Interrupt Operation

The outline diagram of the interrupt generator is shown in Figure 5.11. The interrupt generation is controlled by a normal mode and a test mode. The mode selection is done by the Test_Mode_Enable bit in the test register *[TMR_TIMERITCR]*. The normal mode is selected when the Test_Mode_Enable bit is set to 0.

In the normal mode, the Inerrupt_Enable bit in the control register *[TMR_TIMERnCONTROL]* (**n** is a channel number: 0 or 1.) is set to 1, and when the down counter becomes 0x00000000, the interrupt request is generated. The interrupt generation and its clear timings are shown in Figure 5.10.

The interrupt request **TIMINT** is asserted at the rising edge of the **TIMCLK** during **TIMCLKEN** = 1. The interrupt request **TIMINT** is deasserted at the rising edge of the **PCLK** when the interrupt clear register *[TMR_TIMERnINTCLR]* is written, regardless of the **TIMCLKEN**. The internal interrupt cause is cleared at the rising edge of the **TIMCLK** during **TIMCLKEN**=1.

In the test mode, the TestTININT**n** in the test register *[TMR_TIMERITOP]* controls the interrupt generation and its clear (**n** is a channel number: 0 or 1).

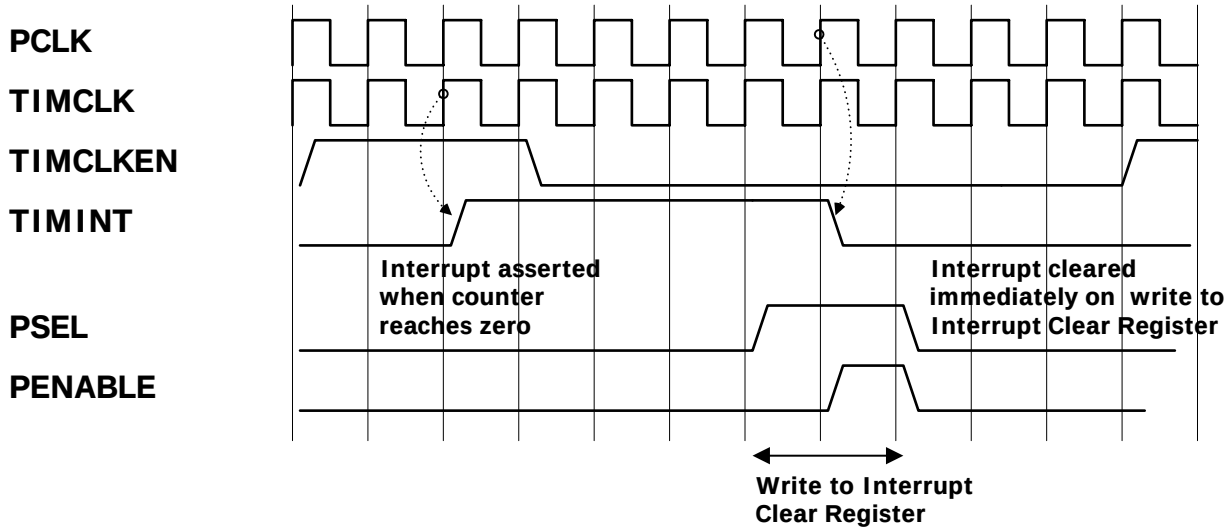

Figure 5.10 Example of interrupt timing chart

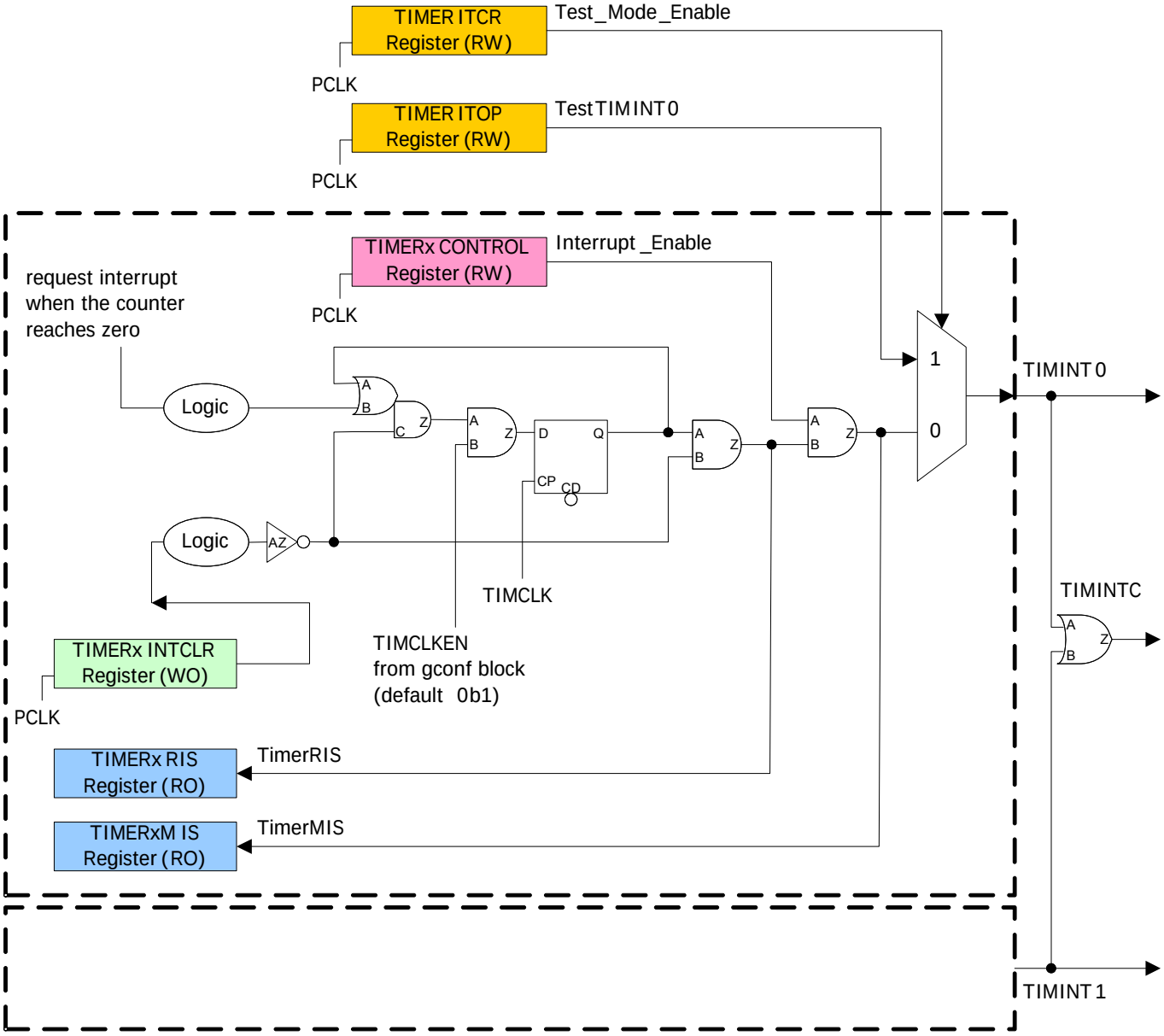

Figure 5.11 Interrupt generator

5.4. Power Management

The power modes of the TZ1000 Series are shown in the following table.

Table 5.3 Power mode and operation

Power mode	State of TMR
ACTIVE	Run (Note)
SLEEP0	Run (Note)
SLEEP1	Run (Note)
SLEEP2	Clock gating
WAIT	Clock gating
RETENTION	Retention
RTC	Power Down
STOP	Power Down

Note: The clock can be started or stopped by software.

- ACTIVE/SLEEP0/SLEEP1:
  Normal operation.
  When the clock stop is set by software, the communication operation stops and every signal operation. To prevent that, the clock should be stopped after the interrupt disable is set. It is inhibited that the clock stops while a data frame is being transferred, because the restart of the transfer may not be done successfully. The clock should stop after it is confirmed that the SPIM transfer is in the stop state.
  When the clock start is set, the transfer re-starts with the same state where the clock stopped.
- SLEEP2/WAIT/ RETENTION:
  The PCLK, TIMCLK0, and TIMCLK1 stop, so the counter also stops and every signal holds its own data. If the interrupt and others are asserted, they will not be able to be deasserted. It is necessary to set the disable to them before the clock stop. And, it is inhibited that those clocks stops while a data frame is being transferred.
  When returning from this mode, every signal is restored to the data in the previous mode.
- RTC/STOP:
  Before the transition to this mode, it should be checked that the appropriate settings are done to disable the corresponding function.
  When returning from this mode, the registers are initialized. So, the operation should re-start after the configuration of the registers completes.

5.5. Start-up and Stop Procedure

5.5.1. Start-up procedure

The start-up procedure after power-on is as follows.

For detail of the PMU registers, refer to the PMU section.

The following setting is supposed that the main bus (the bus connected to the CPU) is supplied with clocks.

Each channel of the timer will be supplied with a clock by this setting.

(In the following, ch0 and ch1 indicate channel 0 and channel 1, respectively.)

("*" shows the signal which also controls another function. The setting should be done together with another setting.)

The order of the setting procedure is a frequency setting at first, then clock supply, and reset deassertion at last.

- TMR clock frequency setting

PMU register	Bit name	Description
CSM_MAIN	CSMSEL_MAIN	0x0000000u u: 0x0: SiOSC4M, 0x1: OSC12M, 0x2: PLL, 0x3: ADPLL, 0x4: OSC32K/SIOSC32K 0x5 – 0x7: reserved
PRESCAL_MAIN	PSSEL_CD_PPIER0	0x****u*** u: 0x0: not generate clock 0x1: divided by 1, 0x2: divided by 2, 0x3: divided by 3, 0x4: divided by 4, 0x5: divided by 5, 0x6: divided by 6, 0x7: divided by 7, 0x8: divided by 8, 0x9: divided by 9, 0xA: divided by 10, 0xB: divided by 12, 0xC: divided by 18, 0xD: divided by 24, 0xE: divided by 36, 0xF: divided by 48

Note: CSM_MAIN sets the source of the clock. The PLL is set by another register and the frequency is changed by the setting. (For detail, refer to the PMU specification.)

Note: PRESCAL_MAIN has a setting to another power domain.

Note: PSSEL_CD_PPIER0 setting specifies all circuits in the PPIER0 power domain.

Note: The clock frequency setting can be changed even if the counter is operating. In this case, note that the period count is also changed.

The start-up sequence of the TMR is as follows.
Clock supply, and then, deassertion of the reset.

- TMR clock supply

PMU register	Bit name	value
CG_OFF_POWERDOMAIN	CG_PM	0x00000001
CG_OFF_PM_1	CG_ppier0clk_tmr_pclk	0x00080003
	CG_mpierclk_h2hp0_hclk	
	CG_ppier0clk_h2pp0_hclk	

Note: Bit 19 in the CG_OFF_PM_1 register corresponds to the TMR bus clock.

Each channel can be supplied separately with a clock by the following setting. To supply all channels with clocks, the following values are written to the CG_OFF_PM_1 register successively, or the ORed value of these values is written to the CG_OFF_PM_1 Register.

- TMR ch1 count clock supply

PMU register	Bit name	value
CG_OFF_PM_1	CG_ppier0clk_tmr_ch0_timclk	0x00200000

Note: Bit 21 in the CG_OFF_PM_1 register corresponds to the TMR ch0.

- TMR ch2 count clock supply

PMU register	Bit name	value
CG_OFF_PM_1	CG_ppier0clk_tmr_ch1_timclk	0x00800000

Note: Bit 23 in the CG_OFF_PM_1 register corresponds to the TMR ch1.

- Reset deassertion of the TMR

PMU register	Bit name	value
SRST_OFF_POWERDOMAIN	SRST_PM	0x00000001
SRST_OFF_PM_1	SRST_asyncrst_tmr_prstn	0x00080003
	SRST_asyncrst_h2pp0_hrstn	
	SRST_asyncrst_h2hp0_hrstn	

Note: Bit 19 in the SRST_OFF_PM_1 register corresponds to the TMR.

- Setting of the TMR and operation start
  - -When the reset is deasserted, the TMR is in the disable state.
  - -It starts to operate by writing to the following registers.

  - -Set the period count to the load register *[TMR_TIMERnLOAD]*.
  - -Set the timer operation mode (the one shot, the constant period, and others) to the control register *[TMR_TIMERnCONTROL]*.
  - -Set Enable = 1 in the control register *[TMR_TIMERnCONTROL]*.
    (**n** is a channel number: 0 or 1.)

5.5.2. Stop procedure

- In the case of that the TMR is stopped.
  The following two ways are used.
  -The TMR is disabled.
  : The Enable in the control register *[TMR_TIMERnCONTROL]* is set to 0.
  The operation of the counter stops.
  -The period count is set to 0.
  : The load register *[TMR_TIMERnLOAD]* is set to 0x00000000.
  The count operation stops.

- In the case that the TMR is not used (the whole block stops);
  The following two ways are used.
  -No reset assertion
  : Only the clock supply is stopped by the following PMU register setting.
  -Reset assertion
  : The reset assertion and the clock stop are set by the following PMU register.

- TMR reset

PMU register	Bit name	value
SRST_ON_PM_1	SRST_asyncrst_tmr_prstn	0x00080000

Note: Bit 19 in the SRST_ON_PM_1 register corresponds to the TMR.
Note: The reset is asserted to all channels in the TMR. Each channel cannot be reset separately.

Each channel clock can be stopped separately by the following setting. To stop all channel clocks, the following values are written to the CG_OFF_PM_1 register successively, or the ORed value of these values is written to the CG_OFF_PM_1 Register.

- TMR ch1 counter clock supply stop

PMU register	Bit name	value
CG_ON_PM_1	CG_ppier0clk_tmr_ch0_timclk	0x00200000

Note: Bit 21 in the CG_ON_PM_1 register corresponds to the TMR ch0.

- TMR ch2 counter clock supply stop

PMU register	Bit name	value
CG_ON_PM_1	CG_ppier0clk_tmr_ch1_timclk	0x00800000

Note: Bit 23 in the CG_ON_PM_1 register corresponds to the TMR ch1.

- TMR bus clock supply stop

PMU register	Bit name	Value
CG_ON_PM_1	CG_ppier0clk_tmr_pclk	0x00080000

Note: Bit 19 in the CG_ON_PM_1 register corresponds to the TMR bus clock.

5.6. Dynamic Clock Gating Setting Procedure

The TZ1000 Series can be set to stop the clock supply unless the clock is necessary. When it is set, the following operation reduces the power dissipation.

("*" shows the signal which also controls another function. The setting should be done together with another setting.)

- Clock supply only when the bus access to the TMR.
- Clock supply stop for the counter when the TMR is disabled (Enable = 0 in the control register *[TMR_TIMERnCONTROL]*).

- TMR dynamic clock gating setting

PMU register	Bit name	value
DCG_POWERDOMAIN	DCG_PM	0x*******1
DCG_PM_0	DCG_mpierclk_mpier_hclk	0x*******1
DCG_PM_1	DCG_ppier0clk_tmr_ch1_timclk	0x**A8***3
	DCG_ppier0clk_tmr_ch0_timclk	
	DCG_ppier0clk_tmr_pclk	
	DCG_ppier0clk_h2pp0_hclk	
	DCG_mpierclk_h2hp0_hclk	

The setting above is to set all channels in the TMR. For each channel setting, the assigned bit is set to 1 separately as follows.

DCG_PM_1 register: bit23 TMR ch1

DCG_PM_1 register: bit21 TMR ch0

DCG_PM_1 register: bit19 TMR bus clock

The setting of the dynamic clock gating can be cleared by writing 0 to the corresponding bit.

- TMR dynamic clock gating deassertion (TMR only)

PMU register	Bit name	value
DCG_PM_1	DCG_ppier0clk_tmr_ch1_timclk	0x**00***3
	DCG_ppier0clk_tmr_ch0_timclk	
	DCG_ppier0clk_tmr_pclk	

6. Precaution for Usage

6.1. Access Restriction Associated with Register Access

The registers in this module are assigned to a 4 KB space with 32-bit interval in the little endian format. The bit allocations are as follows.

This module is connected to the main data bus (the bus connected to the CPU) with 32-bit wide. When 8-bit or 16-bit data is accessed, the operation is in units of 32-bit only. This means that 8-bit or 16-bit access results in reading or writing the other bits than the content bits. So, 32-bit access is recommended. Otherwise, a read error or a write error may occur.

The write to a non-existing bit in a register is ignored. The read of the bit returns 0.

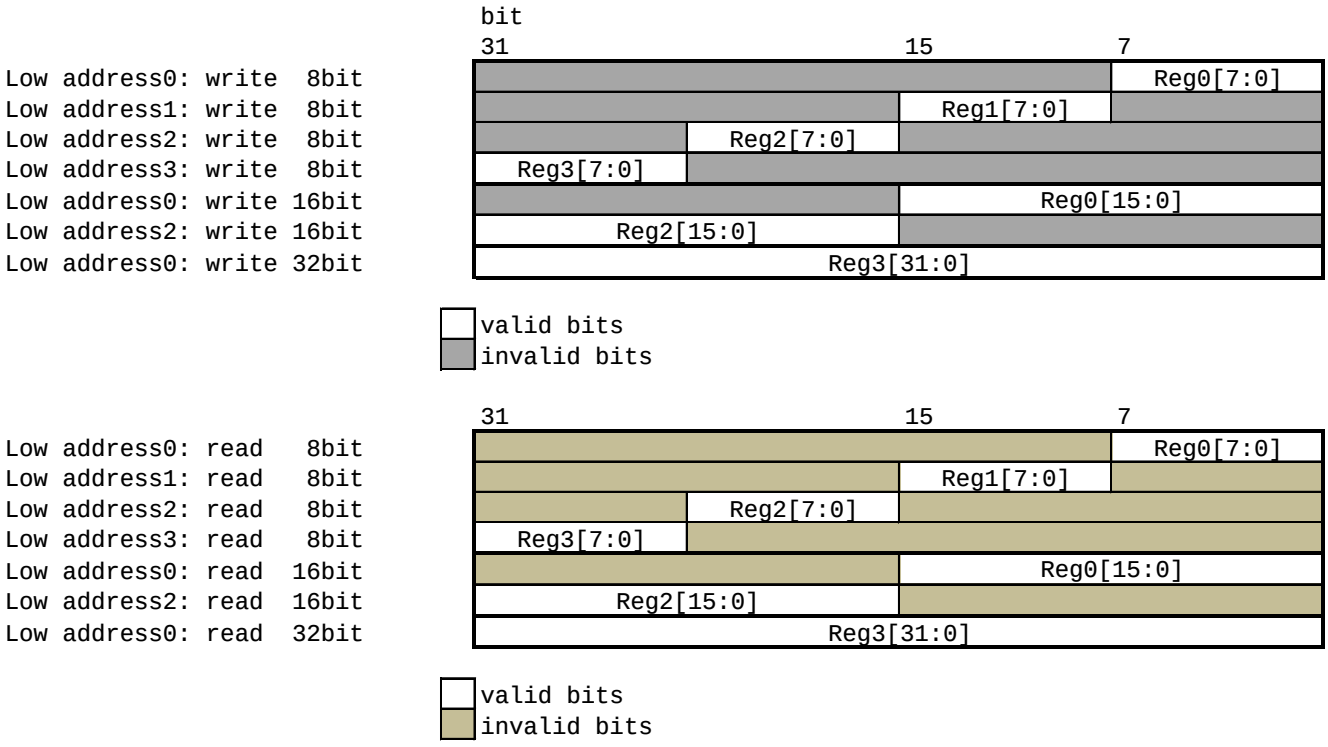

Figure 6.1 Bit allocation of register access

6.2. Precaution for Dynamic Clock Gating

When the dynamic clock gating is set, the following should be noted. When the interrupt generation and the interrupt cause are cleared, the TMR counter clock (**TIMCLK**) is necessary. The clear can be done only when the operation is enabled (Enable = 1 in the control register ***[TMR_TIMERnCONTROL]*** (**n** is a channel number: 0 or 1).

6.3. Using SIOSC4M as the source clock for timer clock

Although SIOSC4M can be selected as the source of TIMCLK, frequency of SIOSC4M may have variation of 4M Hz ± 3.5% due to temperature and/or voltage change. Note that the frequency variation is inherent in TIMCLK and affect timer operation such as interval time when SIOSC4M is used as the source of TIMCLK.

7. Details of Registers

7.1. TMR_TIMER0LOAD

TMR_TIMER0LOAD				
Description	Timer0 Load Register			
Address Region	tmr	**Type:**	RW	
Offset	0x0000 0000			
Physical address View0	0x4004 2000			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:0	Load	Initial or reload value of the counter (0x00000001 to 0xFFFFFFFF settable) The 0x00000000 setting does not operate the counter. Setting any numeric value other than that enables counting. (Note) Writing to this register writes the same value to Timer0BGLoad.	RW modify	0x0000 0000

7.2. TMR_TIMER0VALUE

TMR_TIMER0VALUE				
Description	Timer0 Current value Register			
Address Region	tmr	**Type:**	RO	
Offset	0x0000 0004			
Physical address View0	0x4004 2004			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:0	value	Count value of the counter at the current time	RO	0xFFFF FFFF

7.3. TMR_TIMER0CONTROL

TMR_TIMER0CONTROL				
Description	Timer0 Control Register			
Address Region	tmr	**Type:**	RW	
Offset	0x0000 0008			
Physical address View0	0x4004 2008			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:8	Reserved	-	-	-
7	Enable	Timer operation control 1: Enable 0: Disable	RW modify	0
6	Periodic_Mode	Timer operation mode 1: Periodic Timer (periodic timer) 0: Free-run Timer (free-run timer)	RW modify	0

5	Interrupt_Enable	Interrupt control 1: Interrupt enable 0: Interrupt disable	RW modify	1
4	Reserved	-	-	-
3:2	Clock_divid	Prescaler (clock frequency division) 0x3: Reserved 0x2: 1/256 frequency division 0x1: 1/16 frequency division 0x0: No frequency division (same magnification)	RW modify	0x0
1	Timer_size	Timer size selection 0x1: Operation as a 32-bit timer 0x0: Operation as a 16-bit timer (high-order bits not used)	RW modify	0
0	One_Shot_Count	Timer wrapping/one-shot selection 1: One-shot (only one time) 0: Wrapping (repetition)	RW modify	0

7.4. TMR_TIMER0INTCLR

TMR_TIMER0INTCLR			
Description	Timer0 Interrupt clear Register		
Address Region	tmr	**Type:**	RW
Offset	0x0000 000C		
Physical address View0	0x4004 200C		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:0	IntClr	Interrupt clear After the TIMINT0 interrupt occurs, it can be cleared by write access to this register.	RW clear	-

7.5. TMR_TIMER0RIS

TMR_TIMER0RIS			
Description	Timer0 Raw Interrupt Status Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0010		
Physical address View0	0x4004 2010		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:1	Reserved	-	-	-
0	RIS	A timer interrupt (TIMINT0) before masking is occurring. (The interrupt occurs when the counter becomes 0x00000000.) 1: Interrupt request available 0: No interrupt request	RO	0

7.6. TMR_TIMER0MIS

TMR_TIMER0MIS				
Description	Timer0 Masked Interrupt Status Register			
Address Region	tmr	**Type:**	RO	
Offset	0x0000 0014			
Physical address View0	0x4004 2014			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:1	Reserved	-	-	-
0	MIS	A timer interrupt (TIMINT0) after masking is occurring. 1: Interrupt request available 0: No interrupt request	RO	0

7.7. TMR_TIMER0BGLOAD

TMR_TIMER0BGLOAD				
Description	Timer0 BackGround Load Register			
Address Region	tmr	**Type:**	RW	
Offset	0x0000 0018			
Physical address View0	0x4004 2018			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:0	BGLoad	Background Load register A count value set in this register is reloaded after the counter becomes 0x00000000 (terminal count). The count value to be reloaded can be changed safely during counting. (Note) Writing to this register does not allow writing to Timer0Load.	RW modify	0x0000 0000

7.8. TMR_TIMER1LOAD

TMR_TIMER1LOAD				
Description	Timer1 Load Register			
Address Region	tmr	**Type:**	RW	
Offset	0x0000 0020			
Physical address View0	0x4004 2020			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:0	Load	Initial or reload value of the counter	RW modify	0x0000 0000

7.9. TMR_TIMER1VALUE

TMR_TIMER1VALUE			
Description	Timer1 Current value Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0024		
Physical address View0	0x4004 2024		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:0	value	Count value of the counter at the current time	RO	0xFFFF FFFF

7.10. TMR_TIMER1CONTROL

TMR_TIMER1CONTROL			
Description	Timer1 Control Register		
Address Region	tmr	**Type:**	RW
Offset	0x0000 0028		
Physical address View0	0x4004 2028		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:8	Reserved	-	-	-
7	Enable	Timer operation control	RW modify	0
6	Periodic_Mode	Timer operation mode	RW modify	0
5	Interrupt_Enable	Interrupt control	RW modify	1
4	Reserved	-	-	-
3:2	Clock_divid	Prescaler (clock frequency division)	RW modify	0x0
1	Timer_size	Timer size selection	RW modify	0
0	One_Shot_Count	Selection of timer wrapping and one-shot	RW modify	0

7.11. TMR_TIMER1INTCLR

TMR_TIMER1INTCLR			
Description	Timer1 Interrupt clear Register		
Address Region	tmr	**Type:**	RW
Offset	0x0000 002C		
Physical address View0	0x4004 202C		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:0	IntClr	Interrupt clear	RW clear	-

7.12. TMR_TIMER1RIS

TMR_TIMER1RIS			
Description	Timer1 Raw Interrupt Status Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0030		
Physical address View0	0x4004 2030		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:1	Reserved	-	-	-
0	RIS	A timer interrupt (TIMINT1) after masking is occurring. (The interrupt occurs when the counter becomes 0x00000000.)	RO	0

7.13. TMR_TIMER1MIS

TMR_TIMER1MIS			
Description	Timer1 Masked Interrupt Status Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0034		
Physical address View0	0x4004 2034		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:1	Reserved	-	-	-
0	MIS	A timer interrupt (TIMINT1) after masking is occurring.	RO	0

7.14. TMR_TIMER1BGLOAD

TMR_TIMER1BGLOAD			
Description	Timer1 BackGround Load Register		
Address Region	tmr	**Type:**	RW
Offset	0x0000 0038		
Physical address View0	0x4004 2038		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:0	BGLoad	Background load register	RW modify	0x0000 0000

7.15. TMR_TIMERITCR

TMR_TIMERITCR			
Description	Timer Integration test control Register		
Address Region	tmr	**Type:**	RW
Offset	0x0000 0F00		
Physical address View0	0x4004 2F00		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:1	Reserved	-	-	-
0	Test_Mode_Enable	Test mode control 1: Test mode status 0: Normal operation status	RW modify	0

7.16. TMR_TIMERITOP

TMR_TIMERITOP			
Description	Timer Integration test output set Register		
Address Region	tmr	**Type:**	RW
Offset	0x0000 0F04		
Physical address View0	0x4004 2F04		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:2	Reserved	-	-	-
1	TestTIMINT1	Outputs the TIMINT1 signal (only in test mode, write-only) 1: TIMINT1="1" 0: TIMINT1="0" If this register is read, the TMR_TIMERITCR value appears.	RW modify	0
0	TestTIMINT0	Outputs the TIMINT0 signal (only in test mode, write-only) 1: TIMINT0="1" 0: TIMINT0="0" If this register is read, the TMR_TIMERITCR value appears.	RW modify	0

7.17. TMR_TIMERPERIPHID0

TMR_TIMERPERIPHID0			
Description	Timer Peripheral ID0 Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0FE0		
Physical address View0	0x4004 2FE0		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:8	Reserved	-	-	-
7:0	Partnumber0	Peripheral ID (this register stores a	RO	0x04

		value from a hardware viewpoint.) Partnumber(Lower) 804=>04		

7.18. TMR_TIMERPERIPHID1

TMR_TIMERPERIPHID1				
Description	Timer Peripheral ID1 Register			
Address Region	tmr	**Type:**	RO	
Offset	0x0000 0FE4			
Physical address View0	0x4004 2FE4			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:8	Reserved	-	-	-
7:4	Designer0	Peripheral ID (this register stores a value from a hardware viewpoint.) Designer(Lower) 41=>1	RO	0x1
3:0	Partnumber1	Peripheral ID (this register stores a value from a hardware viewpoint.) Partnumber(Lower) 804=>8	RO	0x8

7.19. TMR_TIMERPERIPHID2

TMR_TIMERPERIPHID2				
Description	Timer Peripheral ID2 Register			
Address Region	tmr	**Type:**	RO	
Offset	0x0000 0FE8			
Physical address View0	0x4004 2FE8			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:8	Reserved	-	-	-
7:4	Revision_number	Peripheral ID (this register stores a value from a hardware viewpoint.) Revision: 1st=>1	RO	0x2
3:0	Designer1	Peripheral ID (this register stores a value from a hardware viewpoint.) Designer(Upper) 41=>4	RO	0x4

7.20. TMR_TIMERPERIPHID3

TMR_TIMERPERIPHID3			
Description	Timer Peripheral ID3 Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0FEC		
Physical address View0	0x4004 2FEC		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:8	Reserved	-	-	-
7:0	Configuration	Peripheral ID (this register stores a value from a hardware viewpoint.) Configuration: 00	RO	0x00

7.21. TMR_TIMERPCELLID0

TMR_TIMERPCELLID0			
Description	Timer PrimeCell ID0 Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0FF0		
Physical address View0	0x4004 2FF0		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:8	Reserved	-	-	-
7:0	TimerPCellID0	Peripheral ID (this register stores a value from a hardware viewpoint.)	RO	0x0D

7.22. TMR_TIMERPCELLID1

TMR_TIMERPCELLID1			
Description	Timer PrimeCell ID1 Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0FF4		
Physical address View0	0x4004 2FF4		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:8	Reserved	-	-	-
7:0	TimerPCellID1	Peripheral ID (this register stores a value from a hardware viewpoint.)	RO	0xF0

7.23. TMR_TIMERPCELLID2

TMR_TIMERPCELLID2			
Description	Timer PrimeCell ID2 Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0FF8		
Physical address View0	0x4004 2FF8		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:8	Reserved	-	-	-
7:0	TimerPCellID2	Peripheral ID (this register stores a value from a hardware viewpoint.)	RO	0x05

7.24. TMR_TIMERPCELLID3

TMR_TIMERPCELLID3			
Description	Timer PrimeCell ID3 Register		
Address Region	tmr	**Type:**	RO
Offset	0x0000 0FFC		
Physical address View0	0x4004 2FFC		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:8	Reserved	-	-	-
7:0	TimerPCellID3	Peripheral ID (this register stores a value from a hardware viewpoint.)	RO	0xB1

8. Revision History

Table 8.1 Revision History

Revision	Date	Description
0.1	2014-03-14	Newly released
0.2	2014-03-17	Modified copyright notation.
0.3	2014-10-20	Modified naming convention for timer channels
0.4	2014-11-17	Added the constraint about the clock frequency setting change.
1.0	2015-01-22	Official version
1.1	2015-02-18	Section 5**:** Revised Interrupt interval in Table 5.1 Interrupt generation interval setting.
1.2	2015-11-24	Added Section 6.3
1.3	2018-02-05	Changed header, footer and the last page. Changed corporate name and descriptions. Modified Arm logo and descriptions.

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, 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, devices related to electric power, and equipment used in finance-related fields. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative.
- 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.**