29 Periodic Interrupt Timer (PIT)

29.1 Introduction

The Periodic Interrupt Timer (PIT) block implements several timers that can be used for DMA triggering, general purpose interrupts and system wakeup.

Figure 622 shows the PIT block diagram.

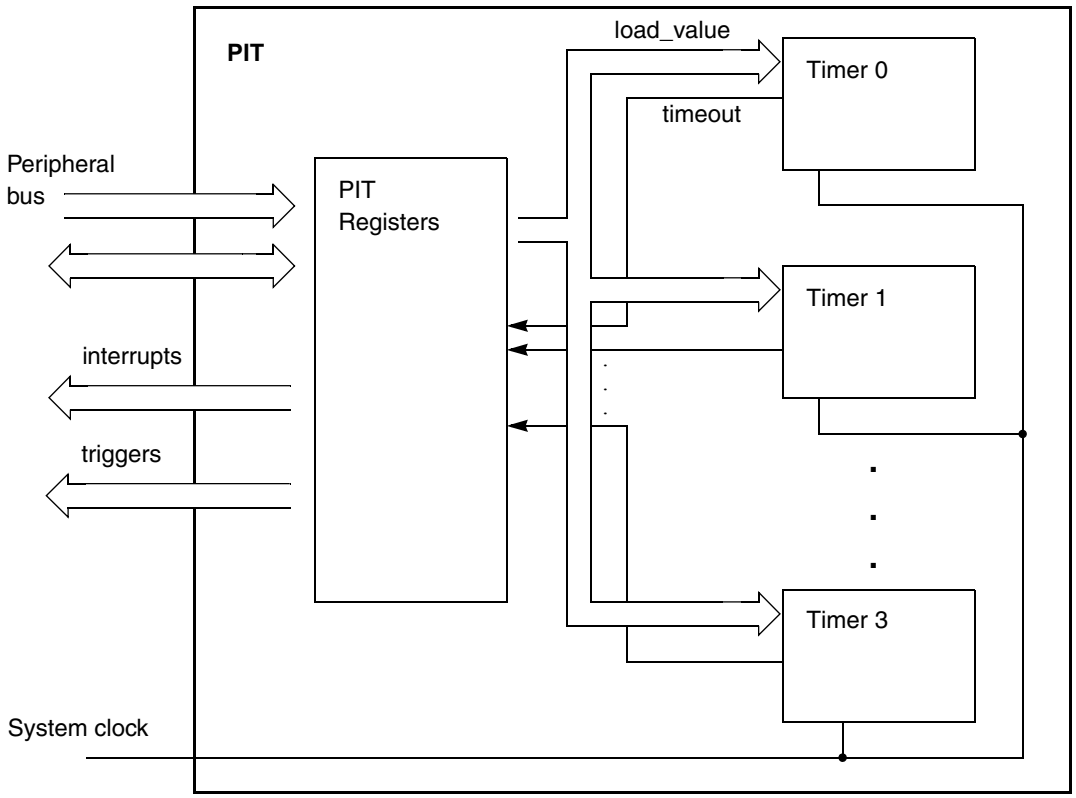

Figure 622. PIT block diagram

29.1.1 Overview

This chapter describes the function of the Periodic Interrupt Timer block (PIT). The PIT is an array of four timers that can be used to raise interrupts and trigger DMA channels.

29.1.2 Features

The main features of this block are:

- Timers can generate DMA trigger pulses to initiate DMA transfers with other peripherals (ex: initiate a SPI message transfer sequence)
- Timers can generate interrupts
- All interrupts are maskable
- Independent timeout periods for each timer

29.2 Signal description

The PIT module has no external pins.

29.3 Memory map and registers description

This section provides a detailed description of all registers accessible in the PIT module.

29.3.1 Memory map

Table 549 gives an overview on all PIT registers. The addresses presented here are the offsets relative to the controller base address 0xC3FF_0000.

Table 549. PIT memory map

Offset from PIT_BASE (0xC3FF_0000)	Register	Access	Reset value	Location
0x0000	PITMCR—PIT Module Control Register	R/W(1)	0x0000_0000	*on page 990*
0x0004–0x00FF	Reserved			
Timer Channel 0				
0x0100	LDVAL0—Timer 0 Load Value Register	R/W	0x0000_0000	*on page 991*
0x0104	CVAL0—Timer 0 Current Value Register	R	0x0000_0000	*on page 991*
0x0108	TCTRL0—Timer 0 Control Register	R/W1	0x0000_0000	*on page 992*
0x010C	TFLG0—Timer 0 Flag Register	R/W1	0x0000_0000	*on page 993*
Timer Channel 1				
0x0110	LDVAL1—Timer 1 Load Value Register	R/W	0x0000_0000	*on page 991*
0x0114	CVAL1—Timer 1 Current Value Register	R	0x0000_0000	*on page 991*
0x0118	TCTRL1—Timer 1 Control Register	R/W1	0x0000_0000	*on page 992*
0x011C	TFLG1—Timer 1 Flag Register	R/W1	0x0000_0000	*on page 993*
Timer Channel 2				
0x0120	LDVAL2—Timer 2 Load Value Register	R/W	0x0000_0000	*on page 991*
0x0124	CVAL2—Timer 2 Current Value Register	R	0x0000_0000	*on page 991*

Table 549. PIT memory map (continued)

Offset from PIT_BASE (0xC3FF_0000)	Register	Access	Reset value	Location
0x0128	TCTRL2—Timer 2 Control Register	R/W[1]	0x0000_0000	*on page 992*
0x012C	TFLG2—Timer 2 Flag Register	R/W[1]	0x0000_0000	*on page 993*
Timer Channel 3				
0x0130	LDVAL3—Timer 3 Load Value Register	R/W	0x0000_0000	*on page 991*
0x0134	CVAL3—Timer 3 Current Value Register	R	0x0000_0000	*on page 991*
0x0138	TCTRL3—Timer 3 Control Register	R/W[1]	0x0000_0000	*on page 992*
0x013C	TFLG3—Timer 3 Flag Register	R/W[1]	0x0000_0000	*on page 993*
0x0140–0x3FFF	Reserved			

1. Some bits are read-only.

Note: Reserved registers read as 0. Writes have no effect.

29.3.2 Registers description

This section describes in address order all the PIT registers and their individual bits. PIT registers are accessible only when the core is in supervisor mode (see *Section 13.4.3 ECSM_reg_protection*).

PIT Module Control Register (PITMCR)

This register controls whether the timer clocks are enabled and whether the timers run in debug mode.

Figure 623. PIT Module Control Register (PITMCR)

Address: Base + 0x0000 Access: User read/write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MDIS	FRZ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Table 550. PITMCR field descriptions

Field	Description
MDIS	Module Disable Used to disable the module clock. This bit should be enabled before any other setup is done. 0: Clock for PIT Timers is enabled 1: Clock for PIT Timers is disabled (default)
FRZ	Freeze Allows the timers to be stopped when the device enters debug mode. 0: Timers continue to run in debug mode. 1: Timers are stopped in debug mode.

Timer Load Value Register *n* (LDVAL*n*)

These registers select the timeout period for the timer interrupts.

Figure 624. Timer Load Value Register *n* (LDVAL*n*)

Channel Base + 0x0000
LDVAL0 = PIT_BASE + 0x0100
Address: LDVAL1 = PIT_BASE + 0x0110 Access: User read/write
LDVAL2 = PIT_BASE + 0x0120
LDVAL3 = PIT_BASE + 0x0130

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	TSV 31	TSV 30	TSV 29	TSV 28	TSV 27	TSV 26	TSV 25	TSV 24	TSV 23	TSV 22	TSV 21	TSV 20	TSV 19	TSV 18	TSV 17	TSV 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	TSV 15	TSV 14	TSV 13	TSV 12	TSV 11	TSV 10	TSV 9	TSV 8	TSV7	TSV6	TSV5	TSV4	TSV3	TSV2	TSV1	TSV0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 551. LDVAL*n* field descriptions

Field	Description
TSV*n*	Time Start Value Bits These bits set the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer, instead the value will be loaded once the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again (see *Figure 629*).

Current Timer Value Register *n* (CVAL*n*)

These registers indicate the current timer position.

Figure 625. Current Timer Value register *n* (CVAL*n*)

Channel Base + 0x0004
CVAL0 = PIT_BASE + 0x0104
Address: CVAL1 = PIT_BASE + 0x0114 Access: User read-only
CVAL2 = PIT_BASE + 0x0124
CVAL3 = PIT_BASE + 0x0134

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TVL31	TVL30	TVL29	TVL28	TVL27	TVL26	TVL25	TVL24	TVL23	TVL22	TVL21	TVL20	TVL19	TVL18	TVL17	TVL16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TVL15	TVL14	TVL13	TVL12	TVL11	TVL10	TVL9	TVL8	TVL7	TVL6	TVL5	TVL4	TVL3	TVL2	TVL1	TVL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 552. CVAL*n* field descriptions

Field	Description
TVL*n*	Current Timer Value These bits represent the current timer value. Note that the timer uses a downcounter. NOTE: The timer values will be frozen in Debug mode if the FRZ bit is set in the PIT Module Control Register (see *Figure 623*).

Timer Control Register *n* (TCTRL*n*)

The TCTRL register contains the control bits for each timer.

Figure 626. Timer Control register *n* (TCTRL*n*)

Channel Base + 0x0008
TCTRL0 = PIT_BASE + 0x0108
Address: TCTRL1 = PIT_BASE + 0x0118 Access: User read/write
TCTRL2 = PIT_BASE + 0x0128
TCTRL3 = PIT_BASE + 0x0138

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIE	TEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 553. TCTRL*n* field descriptions

Field	Description
TIE	Timer Interrupt Enable Bit 0: Interrupt requests from Timer x are disabled 1: Interrupt will be requested whenever TIF is set When an interrupt is pending (TIF set), enabling the interrupt will immediately cause an interrupt event. To avoid this, the associated TIF flag must be cleared first.
TEN	Timer Enable Bit 0: Timer will be disabled 1: Timer will be active

Timer Flag Register *n* (TFLG*n*)

These registers contain the PIT interrupt flags.

Figure 627. Timer Flag register *n* (TFLG*n*)

Channel Base + 0x000C
TFLG0 = PIT_BASE + 0x010C
Address: TFLG1 = PIT_BASE + 0x011C Access: User read/write
TFLG2 = PIT_BASE + 0x012C
TFLG3 = PIT_BASE + 0x013C

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIF
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 554. TFLG*n* field descriptions

Field	Description
TIF	Time Interrupt Flag TIF is set to 1 at the end of the timer period.This flag can be cleared only by writing it with a 1. Writing a 0 has no effect. If enabled (TIE = 1), TIF causes an interrupt request. 0: Time-out has not yet occurred 1: Time-out has occurred

29.4 Functional description

29.4.1 General

This section gives detailed information on the internal operation of the module. Each timer can be used to generate trigger pulses as well as to generate interrupts, each interrupt will be available on a separate interrupt line.

Timers

The timers generate triggers at periodic intervals, when enabled. They load their start values, as specified in their LDVAL registers, then count down until they reach 0. Then they load their respective start value again. Each time a timer reaches 0, it will generate a trigger pulse, and set the interrupt flag.

All interrupts can be enabled or masked (by setting the TIE bits in the TCTRL registers). A new interrupt can be generated only after the previous one is cleared.

If desired, the current counter value of the timer can be read via the CVAL registers.

The counter period can be restarted, by first disabling, then enabling the timer with the TEN bit (see *Figure 628*).

The counter period of a running timer can be modified, by first disabling the timer, setting a new load value and then enabling the timer again (see *Figure 629*).

It is also possible to change the counter period without restarting the timer by writing the LDVAL register with the new load value. This value will then be loaded after the next trigger event (see *Figure 630*).

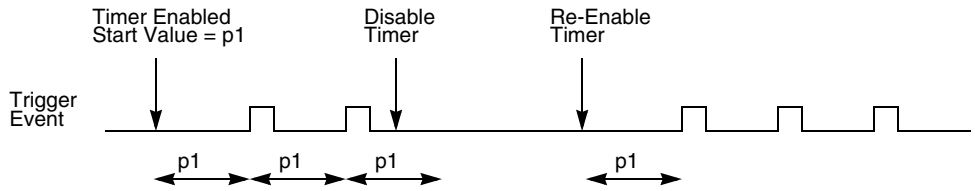

Figure 628. Stopping and starting a timer

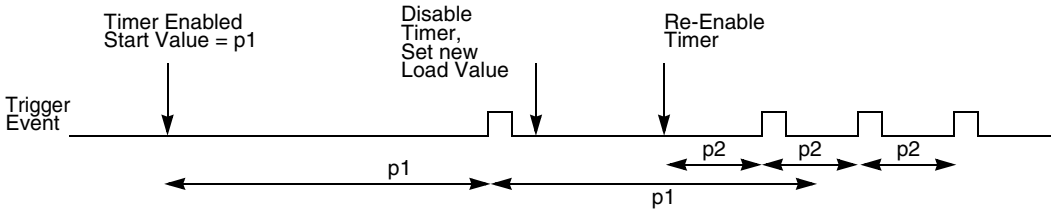

Figure 629. Modifying running timer period