
SpartanMC

Timer Watchdog Module (timer- wdt)

Table of Contents

1. Usage	1
2. Module Parameters	2
3. Interrupts	2
4. Peripheral Registers	2
4.1. Timer Watchdog Register Description	2
4.2. WDT_CTRL Register	3
4.3. WDT_DAT Register	3
4.4. WDT_CHK Register	3
4.5. WDT C-Header for Register Description	4

List of Figures

1 Watchdog timer block diagram 1

List of Tables

1 Timer watchdog module parameters	2
1 Timer watchdog registers	2
1 WDT_CTRL register layout	3
1 WDT maximum value register layout	3
1 WDT counter register layout	3

2. Module Parameters

Table 1: Timer watchdog module parameters

Parameter	Default Value	Description
BASE_ADR		Start address of the memory mapped peripheral registers. The value is taken as offset to the start address of the peripheral memory space. This parameter is set by jConfig automatically.
WDT_RESET_PIN	0x12345	Code word to clear the watchdog timer.

3. Interrupts

If the watchdog counter reaches its maximum value an interrupt can be generated. The interrupt can be cleared by writing the WTD_CTRL register.

4. Peripheral Registers

4.1. Timer Watchdog Register Description

The timer watchdog peripheral provides three 18 bit registers which are mapped to the SpartanMC address space e.g. $0x1A000 + \text{BASE_ADR} + \text{Offset}$.

Table 2: Timer watchdog registers

Offset	Name	Access	Description
0	WDT_CTRL	read/ write	Specify the operation mode of the watchdog timer. (Each write access clears the ALARM bit)
1	WDT_DAT	read/ write	Maximum value of watchdog timer.
2	WDT_CHK	read	If read it contains the current value of the watchdog timer.
2	WDT_CHK	write	Clears the watchdog timer if written with the configured code word.

4.2. WDT_CTRL Register

Table 3: WDT_CTRL register layout

Bit	Name	Access	Default	Description
0	WDT_EN	read/ write	0	If set to one the watchdog timer is enabled.
1	WDT_EN_PRE	read/ write	0	If set to one the prescaler is enabled.
2-4	WDT_PRE_VAL	read/ write	000	Specify the prescaler value : 000 = 2 ¹ 001 = 2 ² 010 = 2 ³ 011 = 2 ⁴ 100 = 2 ⁵ 101 = 2 ⁶ 110 = 2 ⁷ 111 = 2 ⁸
5	WDT_ALARM	read/ write	0	Determines a watchdog alert. Set to null on each write access to this register.
6-17	x	read	0	Not used.

Table 3: WDT_CTRL register layout

4.3. WDT_DAT Register

Table 4: WDT maximum value register layout

Bit	Name	Access	Default	Description
0-17	Max Counter	read/ write	x	Specify the maximum counter value.

4.4. WDT_CHK Register

Table 5: WDT counter register layout

Bit	Name	Access	Default	Description
0-17	Main Counter	read/ (write)	0	If read it contains the current watchdog counter value. If written with WDT_RESET_PIN it clears the watchdog counter value.

4.5. WDT C-Header for Register Description

```
#ifndef __WATCHDOG_H
#define __WATCHDOG_H

#ifdef __cplusplus
extern "C" {
#endif

#define WATCHDOG_EN          (1<<0)
#define WATCHDOG_PRE_EN     (1<<1)
#define WATCHDOG_PRE_VAL    (1<<2)          // *0 fuer 2^1 bis *7
fuer 2^8

#define WATCHDOG_PRE_2      (WATCHDOG_PRE_VAL * 0)
#define WATCHDOG_PRE_4      (WATCHDOG_PRE_VAL * 1)
#define WATCHDOG_PRE_8      (WATCHDOG_PRE_VAL * 2)
#define WATCHDOG_PRE_16     (WATCHDOG_PRE_VAL * 3)
#define WATCHDOG_PRE_32     (WATCHDOG_PRE_VAL * 4)
#define WATCHDOG_PRE_64     (WATCHDOG_PRE_VAL * 5)
#define WATCHDOG_PRE_128    (WATCHDOG_PRE_VAL * 6)
#define WATCHDOG_PRE_256    (WATCHDOG_PRE_VAL * 7)

#define WATCHDOG_ALARM      (1<<5)

typedef struct wdt {
    volatile unsigned int control;    // (r/w)
    volatile unsigned int limit;     // (r/w)
    volatile unsigned int val_rst;    // (r = val / w PIN = rst)
} watchdog_regs_t;
#ifdef __cplusplus
}
#endif

#endif
```