番外篇-定时器

一、OK6410定时器说明

Timers 0 and 1 include a PWM function (Pulse Width Modulation), which can drive an external I/O signal. The PWM for timer 0 and 1 have an optional dead-zone generator capability, which can be utilized to support a large current device. Timer 2, 3 and 4 are internal timers with no output pins.

To control the functionality of PWM, 16 special function registers are provided. The PWM is a programmable output. The 16 special function registers within PWM are accessed via APB transactions.

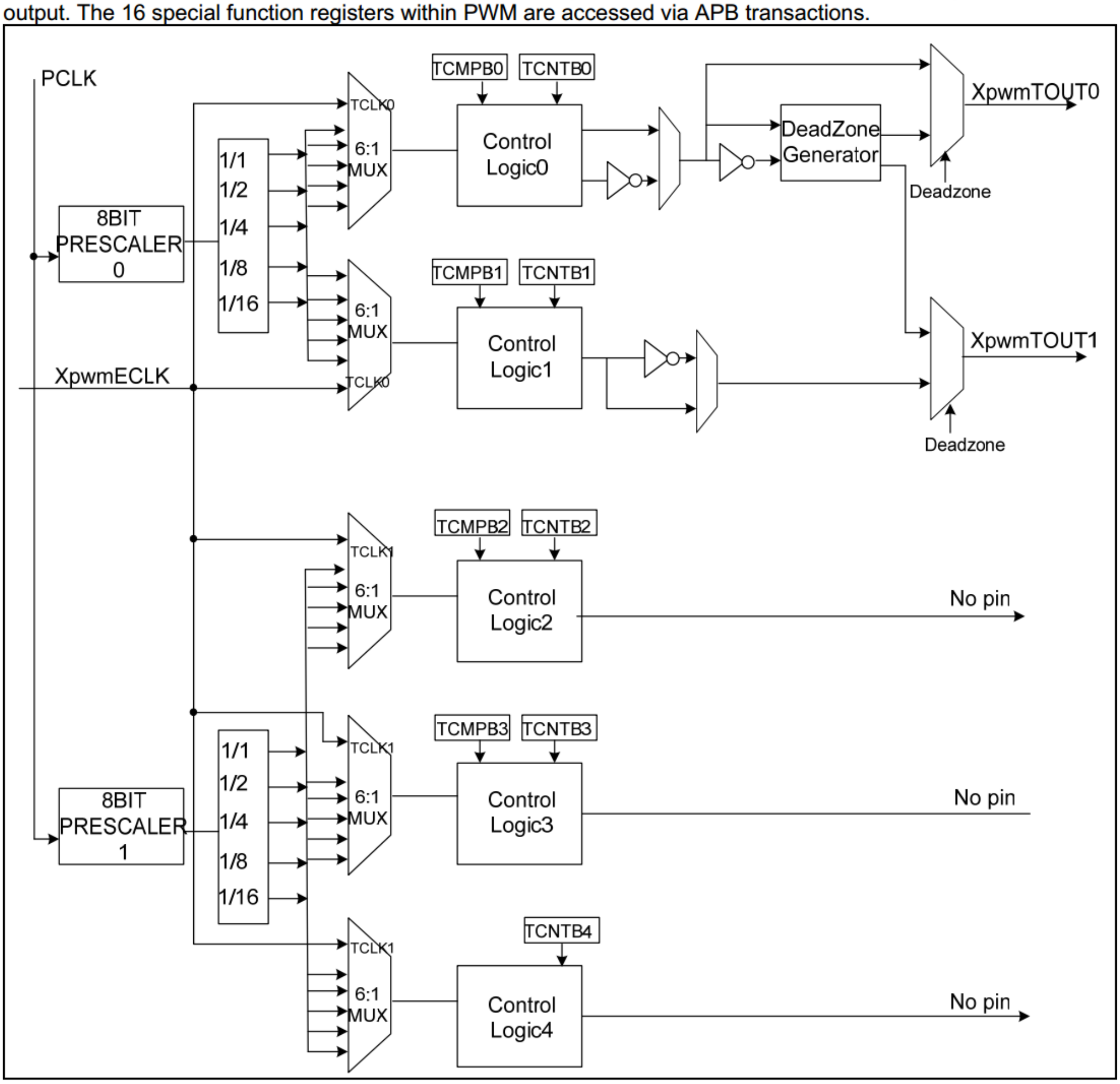


Figure 32-1. PWMTIMER Clock Tree Diagram

The Figure 32-1 depict the clock generation scheme for individual PWM Channels.

二、定时器时钟源

32.3.1 PRESCALER & DIVIDER

An 8-bit prescaler and 4-bit divider makes the following output frequencies:

Table 32-1. Min. and Max. Resolution based on Prescaler and Clock Divider Values

4-bit divider settings	Minimum resolution (prescaler=1)	Maximum resolution (prescaler=255)	Maximum interval (TCNTBn=4294967295)
1/1(PCLK=66 MHz)	0.030us (33.0 MHz)	3.87us(258 kHz)	16621.52s
1/2 (PCLK=66 MHz)	0.060us (16.5 MHz)	7.75us (129 kHz)	33243.05s
1/4 (PCLK=66 MHz)	0.121us (8.25 MHz)	15.5us (64.5 kHz)	66486.09s
1/8 (PCLK=66 MHz)	0.242us (4.13 MHz)	31.0us (32.2 kHz)	132972.19s
1/16 (PCLK=66 MHz)	0.484us (2.07 MHz)	62.1us (16.1 kHz)	265944.37s

三、定时器操作流程

32.3.2 BASIC TIMER OPERATION

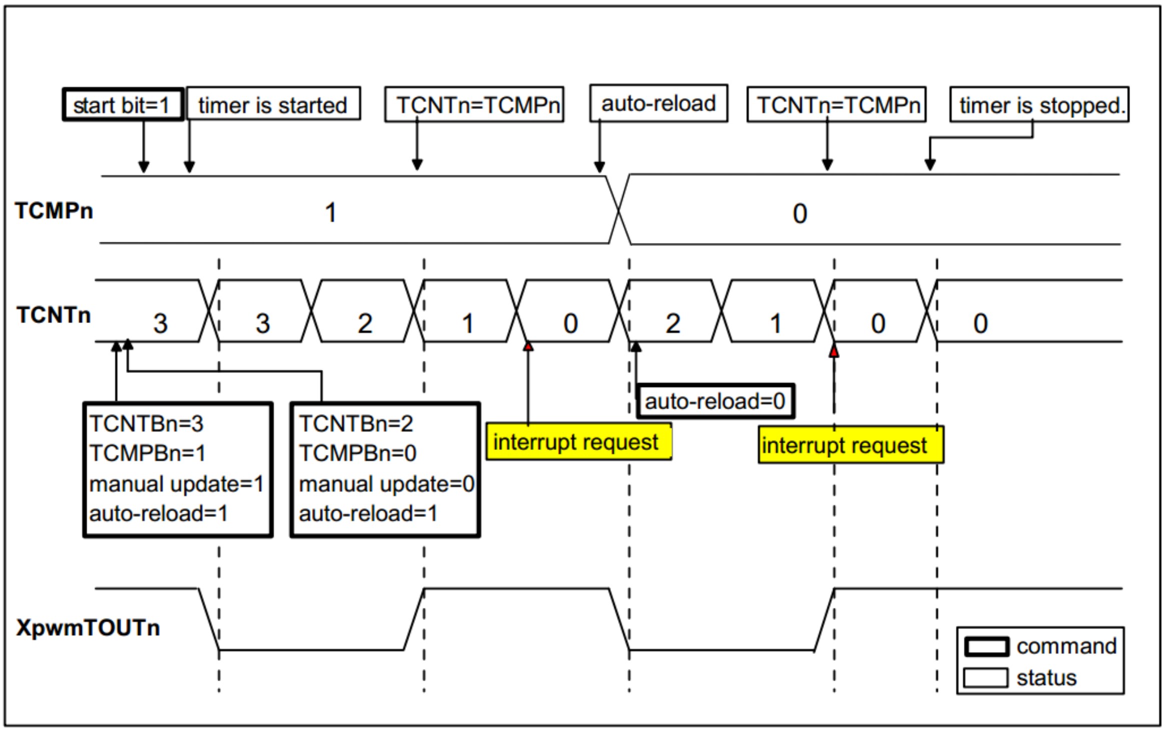


Figure 32-4. Basic Timer operations example

32.3.4 TIMER OPERATION EXAMPLE

The result of the following procedure is shown in Figure 32-6.

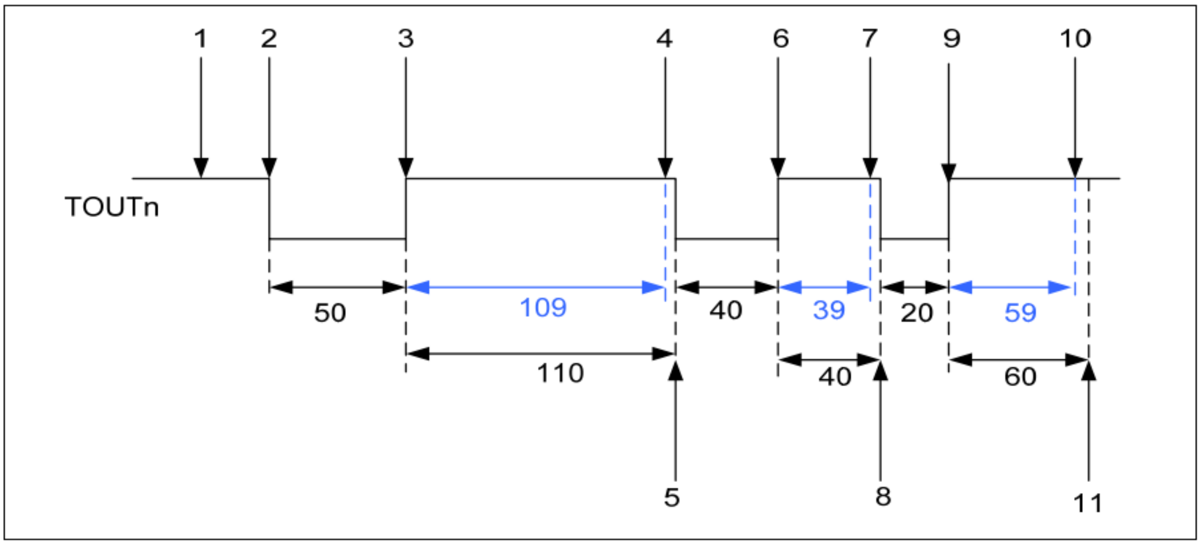


Figure 32-6. Example of a Timer Operation

1. Enable the auto-reload feature. Set the TCNTBn as 159(50+109) and the TCMPBn as 109. Set the manual

update bit and inverter bit(on/off). The manual update bit sets the TCNTn,TCMPn to the value of TCNTBn,TCMPBn.
And then, set TCNTBn,TCMPBn as 79(40+39) and 39.
2. Start Timer by setting the start bit and manual update bit off.
3. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high
4. As soon as TCNTn reaches to 0, the interrupt request is generated.
5. TCNTn and TCMPn are reloaded automatically with TCNTBn,TCMPBn as (79(40+39)) and 39. In the ISR(interrupt service routine), the TCNTBn and TCMPBn are set as 79(20+59) and 59.
6. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high
7. As soon as TCNTn reaches to 0, the interrupt request is generated.
8. TCNTn and TCMPn are reloaded automatically with TCNTBn,TCMPBn as (79(20+59)) and 59. In the ISR(interrupt service routine), auto-reload and interrupt request are disabled to stop the timer.
9. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high
10. Even when TCNTn reaches to 0, No interrupt request is generated.
11. TCNTn is not any more reloaded and the timer is stopped because auto-reload is disabled.

四、定时器寄存器

注意：

1、timer2~4的TCMPn在列表和描述中并没有地址值，但实际上地址有个4字节的预留。

2、(TCNTn and TCMPn are the names of the internal registers. The TCNTn register can be read from the TCNTOn register)

Register	Offset	R/W	Description	Reset Value
TCFG0	0x7F006000	R/W	Timer Configuration Register 0 that configures the two 8-bit Prescaler and DeadZone Length	0x0000_0101
TCFG1	0x7F006004	R/W	Timer Configuration Register 1 that controls 5 MUX and DMA Mode Select Bit	0x0000_0000
TCON	0x7F006008	R/W	Timer Control Register	0x0000_0000
TCNTB0	0x7F00600C	R/W	Timer 0 Count Buffer Register	0x0000_0000
TCMPB0	0x7F006010	R/W	Timer 0 Compare Buffer Register	0x0000_0000
TCNTO0	0x7F006014	R	Timer 0 Count Observation Register	0x0000_0000
TCNTB1	0x7F006018	R/W	Timer 1 Count Buffer Register	0x0000_0000
TCMPB1	0x7F00601c	R/W	Timer 1 Compare Buffer Register	0x0000_0000
TCNTO1	0x7F006020	R	Timer 1 Count Observation Register	0x0000_0000
TCNTB2	0x7F006024	R/W	Timer 2 Count Buffer Register	0x0000_0000
TCNTO2	0x7F00602c	R	Timer 2 Count Observation Register	0x0000_0000
TCNTB3	0x7F006030	R/W	Timer 3 Count Buffer Register	0x0000_0000
TCNTO3	0x7F006038	R	Timer 3 Count Observation Register	0x0000_0000
TCNTB4	0x7F00603c	R/W	Timer 4 Count Buffer Register	0x0000_0000
TCNTO4	0x7F006040	R	Timer 4 Count Observation Register	0x0000_0000
TINT_CSTAT	0x7F006044	R/W	Timer Interrupt Control and Status Register	0x0000_0000

五、实现定时器定时中断及延时程序设计

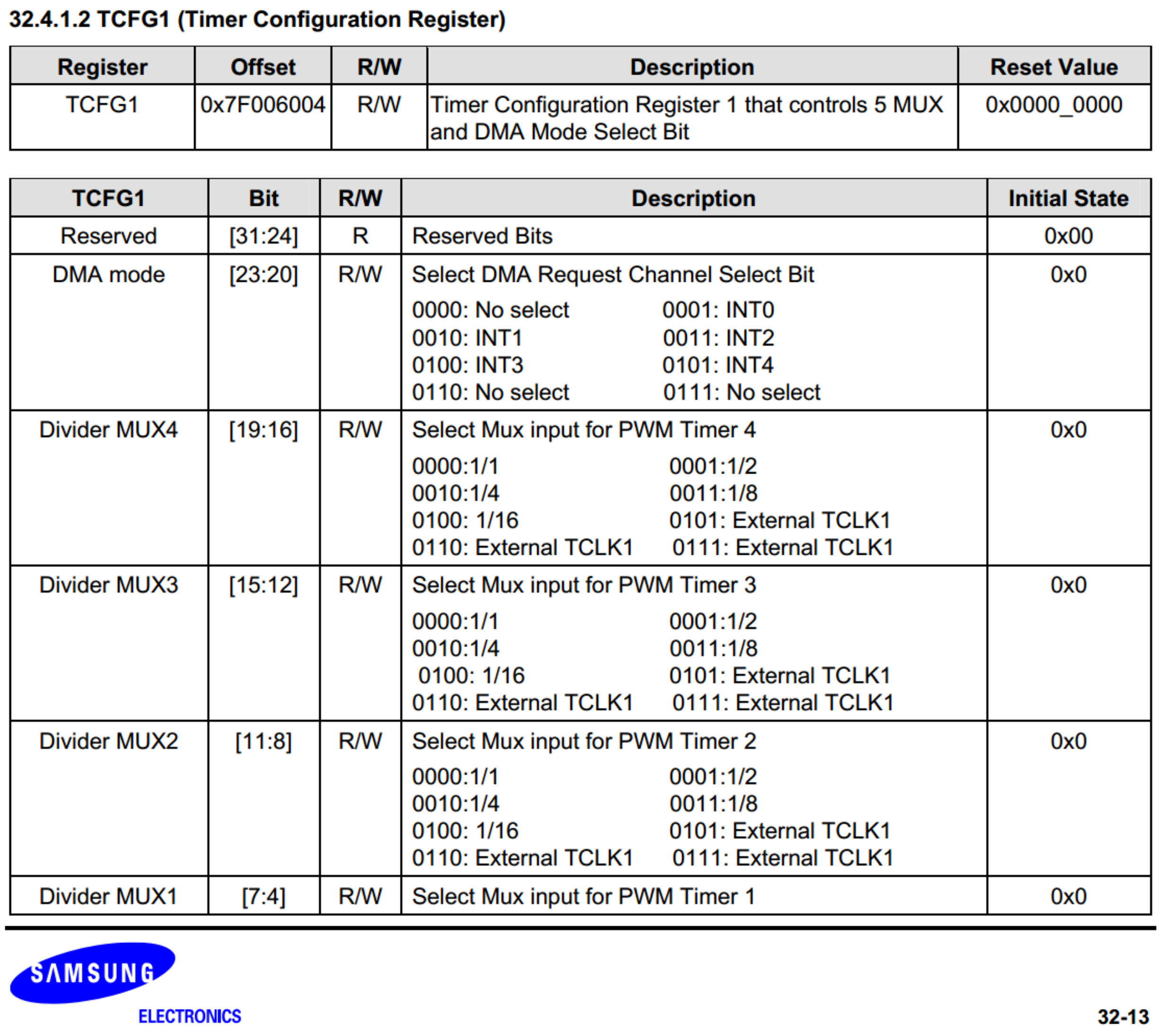
5.1、定时器初始化

32.4.1.1 TCFG0 (Timer Configuration Register)

Register	Offset	R/W	Description	Reset Value
TCFG0	0x7F006000	R/W	Timer Configuration Register 0 that configures the two 8-bit Prescaler and DeadZone Length	0x0000_0101

Timer input clock Frequency = PCLK / ({prescaler value + 1}) / {divider value}
{prescaler value} = 1~255
{divider value} = 1, 2, 4, 8, 16, TCLK

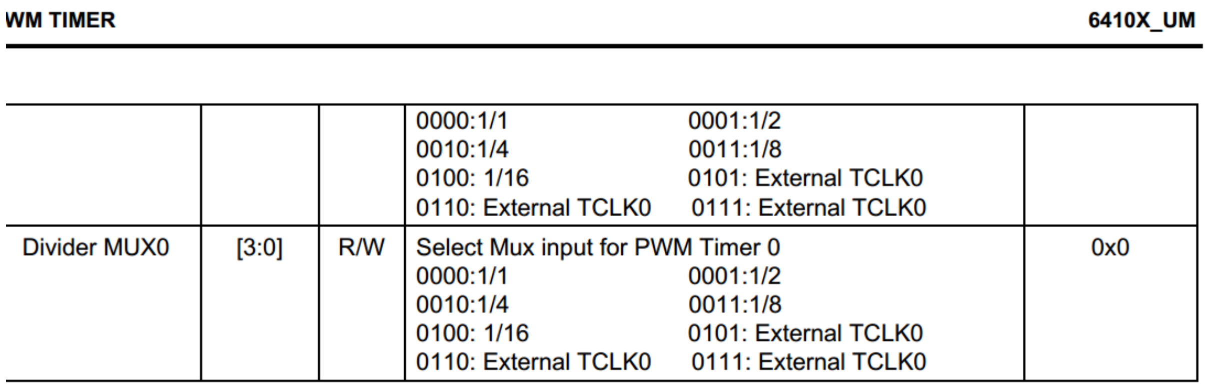
TCFG0	Bit	R/W	Description	Initial State
Reserved	[31:24]	R	Reserved Bits	0x00
Dead zone length	[23:16]	R/W	Dead zone length	0x00
Prescaler 1	[15:8]	R/W	Prescaler 1 value for Timer 2, 3 and 4	0x01
Prescaler 0	[7:0]	R/W	Prescaler 0 value for timer 0 & 1	0x01



32.4.1.2 TCFG1 (Timer Configuration Register)

Register	Offset	R/W	Description	Reset Value
TCFG1	0x7F006004	R/W	Timer Configuration Register 1 that controls 5 MUX and DMA Mode Select Bit	0x0000_0000

TCFG1	Bit	R/W	Description	Initial State
Reserved	[31:24]	R	Reserved Bits	0x00
DMA mode	[23:20]	R/W	Select DMA Request Channel Select Bit 0000: No select 0001: INT0 0010: INT1 0011: INT2 0100: INT3 0101: INT4 0110: No select 0111: No select	0x0
Divider MUX4	[19:16]	R/W	Select Mux input for PWM Timer 4 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	0x0
Divider MUX3	[15:12]	R/W	Select Mux input for PWM Timer 3 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	0x0
Divider MUX2	[11:8]	R/W	Select Mux input for PWM Timer 2 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	0x0
Divider MUX1	[7:4]	R/W	Select Mux input for PWM Timer 1	0x0



			0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK0 0110: External TCLK0 0111: External TCLK0	
Divider MUX0	[3:0]	R/W	Select Mux input for PWM Timer 0 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK0 0110: External TCLK0 0111: External TCLK0	0x0

```
#define TCFG0        (*((volatile unsigned long *)0x7F006000))
#define TCFG1        (*((volatile unsigned long *)0x7F006004))
#define TCON         (*((volatile unsigned long *)0x7F006008))
#define TCNTB2        (*((volatile unsigned long *)0x7F006024))
#define TCMPB2        (*((volatile unsigned long *)0x7F006028))
#define TCNTO2        (*((volatile unsigned long *)0x7F00602C))
#define TINT_CSTAT     (*((volatile unsigned long *)0x7F006044))

```

```
void timer_2_init(void)
{
	/*Timer input clock Frequency = 66MHz / ( {65 + 1} ) / 1 = 1us */

	/*set Prescaler 1 value for Timer 2, 3 and 4 is 65*/
	TCFG0 &= ~(0xff << 8);
	TCFG0 |= (65 << 8);

	/*Select Mux input for PWM Timer 2 : 1/1 */
	TCFG1 &= ~(0xf << 8);
}

```

5.2、定时器定时中断

注意：在开始定时时，

32.4.1.3 TCON (Timer Control Register)

Register	Offset	R/W	Description	Reset Value
TCON	0x7F006008	R/W	Timer Control Register	0x0000_0000

TCON	Bit	R/W	Description	Initial State
Reserved	[31:23]	R	Reserved Bits	0x000
Timer 4 Auto Reload on/off	[22]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Timer 4 Manual Update	[21]	R/W	0: No Operation 1: Update TCNTB4	0
Timer 4 Start/Stop	[20]	R/W	0: Stop 1: Start Timer 4	0
Timer 3 Auto Reload on/off	[19]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Reserved	[18]	R/W	Reserved Bits	0
Timer 3 Manual Update	[17]	R/W	0: No Operation 1:Update TCNTB3,TCMPB3	0
Timer 3 Start/Stop	[16]	R/W	0: Stop 1: Start Timer 3	0
Timer 2 Auto Reload on/off	[15]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Reserved	[14]	R/W	Reserved Bits	0
Timer 2 Manual Update	[13]	R/W	0: No Operation 1: Update TCNTB2,TCMPB2	0
Timer 2 Start/Stop	[12]	R/W	0: Stop 1: Start Timer 2	0
Timer 1 Auto Reload on/off	[11]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Timer 1 Output Inverter on/off	[10]	R/W	0: Inverter Off 1: TOUT1 Inverter-On	0
Timer 1 Manual Update	[9]	R/W	0: No Operation 1: Update TCNTB1,TCMPB1	0
Timer 1 Start/Stop	[8]	R/W	0: Stop 1: Start Timer 1	0
Reserved	[7:5]	R/W	Reserved Bits	000
Dead zone enable/disable	[4]	R/W	Deadzone Generator Enable/Disable	0
Timer 0 Auto Reload on/off	[3]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Timer 0 Output Inverter on/off	[2]	R/W	0: Inverter Off 1: TOUT0 Inverter-On	0
Timer 0 Manual Update	[1]	R/W	0: No Operation 1: Update TCNTB0,TCMPB0	0
Timer 0 Start/Stop	[0]	R/W	0: Stop 1: Start Timer 0	0

32.4.1.10 TCNTB2 (Timer2 Counter Register)

Register	Offset	R/W	Description	Reset Value
TCNTB2	0x7F006024	R/W	Timer 2 Count Buffer Register	0x0000_0000

TCNTB2	Bit	R/W	Description	Initial State
Timer 2 Count Buffer	[31:0]	R/W	Timer 2 Count Buffer Register	0x00000000

32.4.1.11 TCNTO2 (Timer2 Observation Register)

Register	Offset	R/W	Description	Reset Value
TCNTO2	0x7F00602C	R	Timer 2 Count Observation Register	0x0000_0000

TCNTO2	Bit	R/W	Description	Initial State
Timer 2 Count Observation	[31:0]	R	Timer 2 Count Observation Register	0x00000000

32.4.1.16 TINT_CSTAT (Interrupt Control And Status Register)

Register	Offset	R/W	Description	Reset Value
TINT_CSTAT	0x7F006044	R/W	Timer Interrupt Control and Status Register	0x0000_0000

TINT_CSTAT	Bit	R/W	Description	Initial State
Reserved	[31:10]	R	Reserved Bits	0x00000
Timer 4 Interrupt Status	[9]	R/W	Timer 4 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 3 Interrupt Status	[8]	R/W	Timer 3 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 2 Interrupt Status	[7]	R/W	Timer 2 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 1 Interrupt Status	[6]	R/W	Timer 1 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 0 Interrupt Status	[5]	R/W	Timer 0 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 4 interrupt Enable	[4]	R/W	Timer 4 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0
Timer 3 interrupt Enable	[3]	R/W	Timer 3 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0
Timer 2 interrupt Enable	[2]	R/W	Timer 2 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0
Timer 1 interrupt Enable	[1]	R/W	Timer 1 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0
Timer 0 interrupt Enable	[0]	R/W	Timer 0 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0

```
void timer_2_timing(unsigned long usec)
{
    int n,m,f;

    n = usec;

    /*Timer 2 Interrupt Enable. */
    TINT_CSTAT |= (1 << 2);

    //printf("\n\r Timer 2 Interrupt Enable\n\r");

    /*clear Timer Control Register */
    TCON &= ~(0xf << 12);

    //printf("\n\r clear Timer Control Register\n\r");

```

```
	/*clear timer2 interrupt statu */
	//TINT_CSTAT |= (0x1 << 7);

	/*f = TCNTB2;
	printf("\n\r TCNTB2 is : %d \n\r", f);
	m = TCNTO2;
	printf("\n\r TCNTO2 is : %d \n\r", m);*/

	/*load count value */
	TCNTB2 = n;

	/*printf("\n\r load count value\n\r");

	f = TCNTB2;
	printf("\n\r TCNTB2 is : %d \n\r", f);
	m = TCNTO2;
	printf("\n\r TCNTO2 is : %d \n\r", m);*/

	/*Timer 2 Auto Reload on */
	TCON |= (0x1 << 15);

	/*printf("\n\r Timer 2 Auto Reload on\n\r");

	f = TCNTB2;
	printf("\n\r TCNTB2 is : %d \n\r", f);
	m = TCNTO2;
	printf("\n\r TCNTO2 is : %d \n\r", m);*/

	/*Timer 2 Manual Update */
	TCON |= (0x1 << 13);

	/*printf("\n\r Timer 2 Manual Update\n\r");

	f = TCNTB2;
	printf("\n\r TCNTB2 is : %d \n\r", f);
	m = TCNTO2;
	printf("\n\r TCNTO2 is : %d \n\r", m);*/

	/*end Timer 2 Manual Update */
	TCON &= ~(0x1 << 13);

	/* Start Timer2 */
	TCON |= (1 << 12);

	/*printf("\n\r Start Timer2\n\r");

	f = TCNTB2;
	printf("\n\r TCNTB2 is : %d \n\r", f);
	m = TCNTO2;
	printf("\n\r TCNTO2 is : %d \n\r", m);*/

	/*Timer 2 Auto Reload off */
	TCON &= ~(0x1 << 15);

	/*printf("\n\r Timer 2 Auto Reload off\n\r");

	f = TCNTB2;
	printf("\n\r TCNTB2 is : %d \n\r", f);
	m = TCNTO2;
	printf("\n\r TCNTO2 is : %d \n\r", m);*/

}

void timer_2_ms(unsigned long msec)
{
	int n;
	n = msec * 1000;
	timer_2_timing(n);
```

```
}

/* sec < 4294 */
void timer_2_s(unsigned long sec)
{
        int n;
        n = sec * 1000000;
        timer_2_timing(n);
}

int timer_2_int_isr(void)
{
        led_xor();

        //input_time();

        /* end Timer 2 */
        TCON &= ~(1 << 12);

        /*Timer 2 Interrupt disable. */
        TINT_CSTAT &= ~(1 << 2);

        return 0;
}

```

5.2、定时器实现延时

```
void udelay(unsigned long usec)
{
        int n;

        n = usec;

        /*Timer 2 Interrupt disable. */
        TINT_CSTAT &= ~(1 << 2);

        /*clear Timer Control Register */
        TCON &= ~(0xf << 12);

        /*clear timer2 interrupt statu */
        TINT_CSTAT |= (0x1 << 7);

        /*load count value */
        TCNTB2 = n;

        /*Timer 2 Auto Reload on */
        TCON |= (0x1 << 15);

        /*Timer 2 Manual Update */
        TCON |= (0x1 << 13);

        /*end Timer 2 Manual Update */
        TCON &= ~(0x1 << 13);

        /* Start Timer 2 */
        TCON |= (1 << 12);

        /*Timer 2 Auto Reload off */
        TCON &= ~(0x1 << 15);

        while (!(TINT_CSTAT & (0x1 << 7))) ;

        /*clear timer2 interrupt statu */
        TINT_CSTAT |= (0x1 << 7);

        /* end Timer 2 */
        TCON &= ~(1 << 12);
}
```

```

void mdelay(unsigned long msec)
{
        int n;
        n = msec * 1000;
        udelay(n);
}

/* sec < 4294 */
void sdelay(unsigned long sec)
{
        int n;
        n = sec * 1000000;
        udelay(n);
}

void timer_2_TCNTO_print(void)
{
        int count;
        count = TCNTO2;
        printf("\n TCNTO2 is %d!\n\r", TCNTO2);
}
```