

Real Time Clock and Interrupt



Reference Manual

CRG

Block User Guide

(Clock & Reset Generator)

CRG Registers

CRG Related 12 Registers Base Address = \$34

Table 3-1 CRG Memory Map

Address Offset	Use	Access
\$_00	CRG Synthesizer Register (SYNR)	R/W
\$_01	CRG Reference Divider Register (REFDV)	R/W
\$_02	CRG Test Flags Register (CTFLG) ¹	R/W
\$_03	CRG Flags Register (CRGFLG)	R/W
\$_04	CRG Interrupt Enable Register (CRGINT)	R/W
\$_05	CRG Clock Select Register (CLKSEL)	R/W
\$_06	CRG PLL Control Register (PLLCTL)	R/W
\$_07	CRG RTI Control Register (RTICTL)	R/W
\$_08	CRG COP Control Register (COPCTL)	R/W
\$_09	CRG Force and Bypass Test Register (FORBYP) ²	R/W
\$_0A	CRG Test Control Register (CTCTL) ³	R/W
\$_0B	CRG COP Arm/Timer Reset (ARMCOP)	R/W

Real Time Interrupt Control Register

RTICTL = \$3B

B7	6	5	4	3	2	1	B0
0	n	n	n	m	m	m	m

Reset : 0 0 0 0 0 0 0 0

Use External Clock to generate Real time Clock Interrupt:

RTI Frequency Divide Rate = $(m+1) \times 2^{n+9}$

m = 0~15

n = 1~7

RTI Frequency Divide Rate

2^{10}

RTR[3:0]		RTR[6:4] =							
m	n	000 (OFF)	001 (2 ¹⁰)	010 (2 ¹¹)	011 (2 ¹²)	100 (2 ¹³)	101 (2 ¹⁴)	110 (2 ¹⁵)	111 (2 ¹⁶)
		0000 (+1)	0001 (+2)	0010 (+3)	0011 (+4)	0100 (+5)	0101 (+6)	0110 (+7)	0111 (+8)
		OFF*	2 ¹⁰	2 ¹¹	2 ¹²	2 ¹³	2 ¹⁴	2 ¹⁵	2 ¹⁶
		OFF*	2x2 ¹⁰	2x2 ¹¹	2x2 ¹²	2x2 ¹³	2x2 ¹⁴	2x2 ¹⁵	2x2 ¹⁶
		OFF*	3x2 ¹⁰	3x2 ¹¹	3x2 ¹²	3x2 ¹³	3x2 ¹⁴	3x2 ¹⁵	3x2 ¹⁶
		OFF*	4x2 ¹⁰	4x2 ¹¹	4x2 ¹²	4x2 ¹³	4x2 ¹⁴	4x2 ¹⁵	4x2 ¹⁶
		OFF*	5x2 ¹⁰	5x2 ¹¹	5x2 ¹²	5x2 ¹³	5x2 ¹⁴	5x2 ¹⁵	5x2 ¹⁶
		OFF*	6x2 ¹⁰	6x2 ¹¹	6x2 ¹²	6x2 ¹³	6x2 ¹⁴	6x2 ¹⁵	6x2 ¹⁶
		OFF*	7x2 ¹⁰	7x2 ¹¹	7x2 ¹²	7x2 ¹³	7x2 ¹⁴	7x2 ¹⁵	7x2 ¹⁶
		OFF*	8x2 ¹⁰	8x2 ¹¹	8x2 ¹²	8x2 ¹³	8x2 ¹⁴	8x2 ¹⁵	8x2 ¹⁶
		OFF*	9x2 ¹⁰	9x2 ¹¹	9x2 ¹²	9x2 ¹³	9x2 ¹⁴	9x2 ¹⁵	9x2 ¹⁶
		OFF*	10x2 ¹⁰	10x2 ¹¹	10x2 ¹²	10x2 ¹³	10x2 ¹⁴	10x2 ¹⁵	10x2 ¹⁶
		OFF*	11x2 ¹⁰	11x2 ¹¹	11x2 ¹²	11x2 ¹³	11x2 ¹⁴	11x2 ¹⁵	11x2 ¹⁶
		OFF*	12x2 ¹⁰	12x2 ¹¹	12x2 ¹²	12x2 ¹³	12x2 ¹⁴	12x2 ¹⁵	12x2 ¹⁶
		OFF*	13x2 ¹⁰	13x2 ¹¹	13x2 ¹²	13x2 ¹³	13x2 ¹⁴	13x2 ¹⁵	13x2 ¹⁶
		OFF*	14x2 ¹⁰	14x2 ¹¹	14x2 ¹²	14x2 ¹³	14x2 ¹⁴	14x2 ¹⁵	14x2 ¹⁶
		OFF*	15x2 ¹⁰	15x2 ¹¹	15x2 ¹²	15x2 ¹³	15x2 ¹⁴	15x2 ¹⁵	15x2 ¹⁶
		OFF*	16x2 ¹⁰	16x2 ¹¹	16x2 ¹²	16x2 ¹³	16x2 ¹⁴	16x2 ¹⁵	16x2 ¹⁶

16×2^{16}

Real Time Clock Divider Example

RTICTL

EQU

\$3B

LDAA

#\$74

;48.828125/Sec.

STAA

RTICTL

```
#define RTICTL  (*((unsigned char*)(0x003B)))
```

```
RTICTL = 0x74;
```

```
/* ( 4 + 1 ) x ( 2 (7+9) ) */
```

```
16MHz/5x216 = 48.828125/Sec.
```

Clock & Reset Generator Interrupt Register

RGINT = \$38

	B7	6	5	4	3	2	1	B0
38	RTIE	0	0	LOCKIE	0	0	SCMIE	0
Reset :	0	0	0	0	0	0	0	0

RTIE=1: Real Time Interrupt Enable

LOCKIE: PLL Lock Interrupt Enable

SCMIE: Self Clock Mode Interrupt Enable

Enable Real Time Interrupt

CRGINT EQU \$38

RTI_ENABL LDAA #\$80

ORA CRGINT

#define CRGINT (*((unsigned char*)(0x0038)))

CRGINT |=0x80; /* Enable Interrupt */

Interrupt Service Routine & Initial INT Vector

Vector Address	Interrupt Source	CCR Mask	Local Enable	HPRIO Value to Elevate
FFFE, \$FFFF	Reset	None	None	—
FFFC, \$FFFD	Clock Monitor fail reset	None	COPCTL (CME, FCME)	—
FFFA, \$FFFB	COP failure reset	None	COP rate select	—
FFF8, \$FFF9	Unimplemented instruction trap	None	None	—
FFF6, \$FFF7	SWI	None	None	—
FFF4, \$FFF5	XIRQ	X-Bit	None	—
FFF2, \$FFF3	IRQ	I-Bit	INTCR (IRQEN)	\$F2
FFF0, \$FFF1	Real Time Interrupt	I-Bit	CRGINT (RTIE)	\$F0
FFEE, \$FFEF	Timer channel 0	I-Bit	TMSK1 (C0I)	\$EE
FFEC, \$FFED	Timer channel 1	I-Bit	TMSK1 (C1I)	\$EC
FFEA, \$FFEB	Timer channel 2	I-Bit	TMSK1 (C2I)	\$EA
FFE8, \$FFE9	Timer channel 3	I-Bit	TMSK1 (C3I)	\$E8
FFE6, \$FFE7	Timer channel 4	I-Bit	TMSK1 (C4I)	\$E6

Notice: Real Time IRQ has moved to \$EFF0!

Real Time Clock Interrupt Example

CRGINT **EQU** **\$38**
RTICTL **EQU** **\$3B**

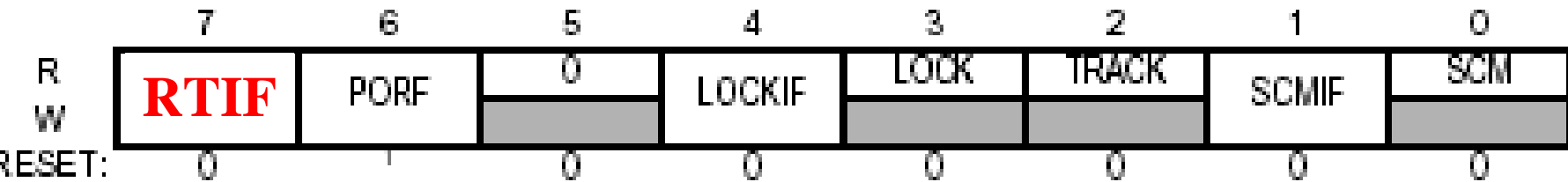
LDAA **#\$74** **;48.828125/Sec.**
 STAA **RTICTL**
 LDAA **#\$80**
 ORA **CRGINT** **;Enable IRQ**

```
#define CRGINT  (*((unsigned char*)(0x0038)))  
#define RTICTL  (*((unsigned char*)(0x003B)))
```

```
RTICTL = 0x74; /* ( 4 + 1 ) x ( 2 ( 7 + 9 ) ) */  
CRGINT |= 0x80; /* Enable Interrupt */
```

Clear Interrupt in ISR

CRG Flags Register (CRGFLG) Address = \$37



If RTIE=1 in CRGINT, RTIF in CRGINT causes an IRQ, when Time Out

Writing 1 to RTIF in CRGFLG clear IRQ

```
CRGFLG EQU $37
CLRRTIRQ LDAA #$80
STAA CRGFLG
```

RTIF is set to 1 at the end of the RTI period. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (RTIE=1), RTIF causes an interrupt request.

```
#define CRGFLG (*((unsigned char*)(0x0037)))
```

```
CRGFLG &= 0x7F; ???
```

Interrupt Service Routine Example

```
int j;  
LED_ON_OFF()  
{  
    if (j == 0)  
    {  
        PORTB &= 0x7f; /* LED7 On */  
        j = 1;}  
    else  
    {  
        PORTB |= 0x80; /* LED7 Off */  
        j = 0;}  
};
```

XREF RTIISR

```
RTIISR JSR     LED_ON_OFF ;  
      LDAA    #$80        ; Clear IRQ  
      STAA    CRGFLG  
      RTI
```

```
ORG    $EFF0  
DC.W   RTIISR
```

Question: How to Get Exact 50mS Time Tick?