

Clock Reset Generator

Viktor Kämpe

Clock Reset Generator Diagram

Kvittera:

1. Kvittera

Initiera:

1. Avbrottsintervall
2. Aktivera CRG
3. Avbrottsvektor
4. I-flaggan

Clock Reset Generator (CRG) @ Base Address 0x34											
Offset		7	6	5	4	3	2	1	0	Mnemonic	Namn
\$00	R	0	0	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0	SYNR	Synthesizer Register
	W										
\$01	R	0	0	0	0	REFDV3	REFDV2	REFDV1	REFDV0	REFDV	Reference Divide Register
	W										
\$02	R	0	0	0	0	0	0	0	0	CTFLG	*)Test Flags Register
	W										
\$03	R	RTIF	PORF	LVRF	LOCKIF	LOCK	SCMIE	SCMIF	SCM	CRGFLG	Flags Register
	W										
\$04	R	RTIE	0	0	LOCKIE	0	0	SCMIE	0	CRGINT	Interrupt Enable Register
	W										
\$05	R	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI	CLKSEL	Clock Select Register
	W										
\$06	R	0	PLLON	AUTO	AOQ	0	PRE	PCE	SCME	PLLCTL	PLL Control Register
	W										
\$07	R	0	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0	RTICTL	RTI Control Register
	W										
\$08	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0	COPCTL	COP Control Register
	W										
\$09	R	0	0	0	0	0	0	0	0	FORBYP	*)Force and Bypass Test Register
	W										
\$0A	R	0	0	0	0	0	0	0	0	CTCTL	*)Test Control Register
	W										
\$0B	R	0	0	0	0	0	0	0	0	ARMCOP	COP Arm/Timer Reset
	W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		

RTIF = real time interrupt flag

RTIE = real time interrupt enable

RTR = real time rate

Initiering (steg 1-3)

```
#include "defsCRG.H"
#ifdef SIMTEST
    #define TIMEBASE 0x10
#else
    #define TIMEBASE 0x49
#endif

#define SET_IRQ_VECTOR( interrupt_handler, address ) \
    *((unsigned int *) address = &( interrupt_handler )

void timer_interrupt_routine( void );

void timer_init( void )
{
    // Skriv tidbas för avbrottsintervall till RTICTL
    CRG->rtictl = TIMEBASE;

    // Aktivera avbrott från CRG-modul
    CRG->crgint = RTIE;

    // måste också initiera avbrottsvektor
    SET_IRQ_VECTOR(timer_interrupt_routine, 0x3FF0 );
}
```

```
// i fil "defsCRG.H"
struct sCRG {
    volatile unsigned char synr;      // 0x00
    volatile unsigned char refdv;    // 0x01
    volatile unsigned char ctflg;    // 0x02
    volatile unsigned char crgflg;   // 0x03
    volatile unsigned char crgint;   // 0x04
    volatile unsigned char clkssel;  // 0x05
    volatile unsigned char pllctl;   // 0x06
    volatile unsigned char rtictl;   // 0x07
    volatile unsigned char copctl;   // 0x08
    volatile unsigned char forbyp;   // 0x09
    volatile unsigned char ctctl;    // 0x0A
    volatile unsigned char armcop;   // 0x0B
};

#define CRG_BASE 0x34
#define CRG ((struct sCRG*) (CRG_BASE))

#define PLLSEL      0x80
#define LOCK        8
#define RTIE        0x80
#define RTIF        0x80

#define REFDVVal    1
#define SYNRVAl     5
```

Errata

I LabBM del2, uppgift 4.3 står det:

"init_cLock. Nollställer klockan och initierar CRG-kretsen så att den genererar ett avbrott ungefär var 10:e ms."

Det borde stå:

•*"init_cLock. Initierar CRG-kretsen så att den genererar ett avbrott ungefär var 10:e ms."*

Nollställning ej nödvändig. Initiering sätter CRG i det önskade läget.