Timer 32

Last updated 6/17/19

Timer 32

- Basic Timer Function

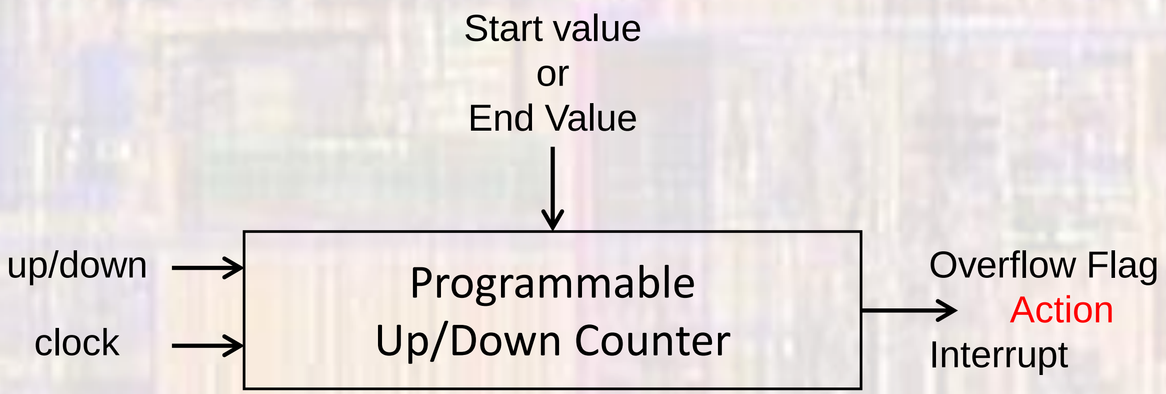
  - Delay Counter
    - Load a value into a counter register
    - The counter counts
      - Down to zero (count down timer)
      - Up from zero (count up timer)
    - An action is triggered when complete

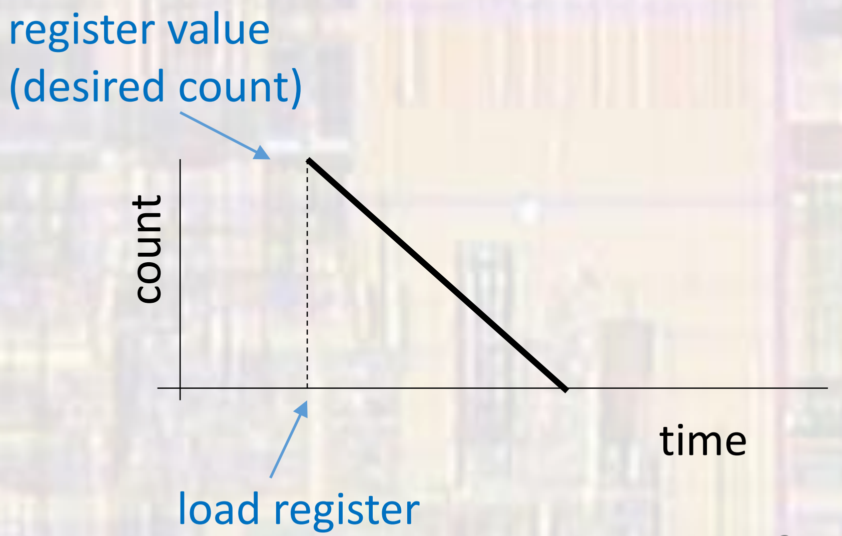
  - Delay is a function of
    - Clock frequency
    - Count value

  - 1ms delay with a 12Mhz clock
    - 1ms * 12M cycles/s = 12000 cycles → set count value to 12,000

Timer 32

- Basic Timer Function



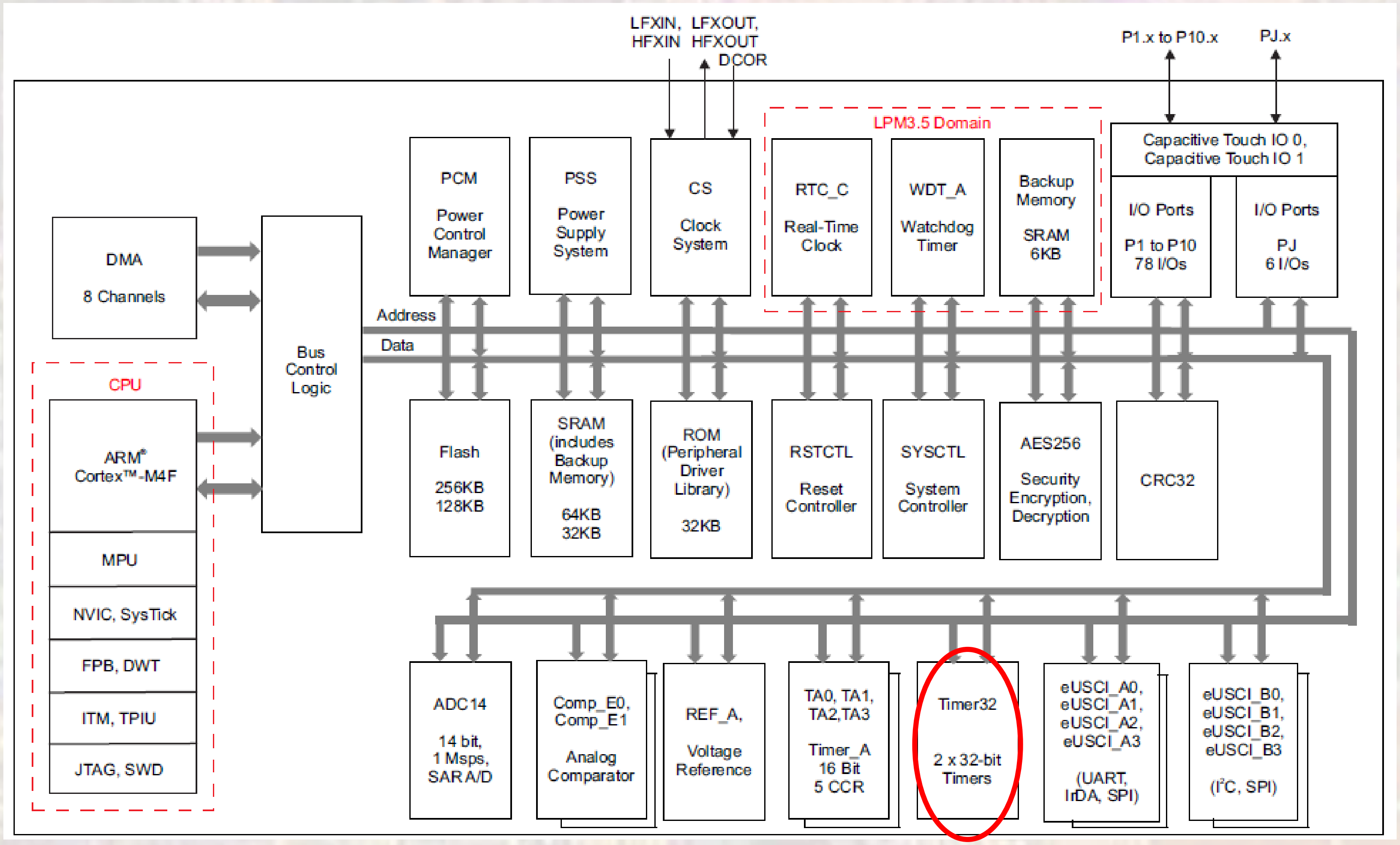


Counter value

9
8
7
6
5
4
3
2
1
0 → Action

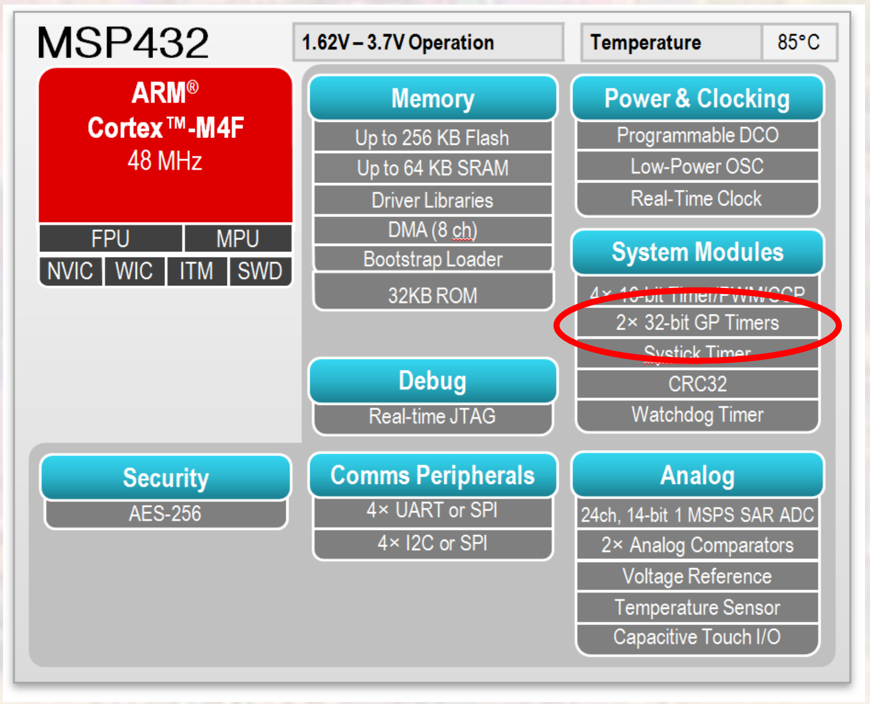
Timer 32

- MSP432 Timer 32


LFXIN, LFXOUT,
HFXIN HFXOUT
DCOR
P1.x to P10.x
PJ.x
LPM3.5 Domain
PCM Power Control Manager
PSS Power Supply System
CS Clock System
RTC_C Real-Time Clock
WDT_A Watchdog Timer
Backup Memory SRAM 6KB
Capacitive Touch IO 0, Capacitive Touch IO 1
I/O Ports P1 to P10 78 I/Os
I/O Ports PJ 6 I/Os
DMA 8 Channels
Bus Control Logic
Address
Data
CPU
ARM® Cortex™-M4F
MPU
NVIC, SysTick
FPB, DWT
ITM, TPIU
JTAG, SWD
Flash 256KB 128KB
SRAM (includes Backup Memory) 64KB 32KB
ROM (Peripheral Driver Library) 32KB
RSTCTL Reset Controller
SYSCTL System Controller
AES256 Security Encryption, Decryption
CRC32
ADC14 14 bit, 1 Msps, SAR A/D
Comp_E0, Comp_E1 Analog Comparator
REF_A, Voltage Reference
TA0, TA1, TA2,TA3 Timer_A 16 Bit 5 CCR
Timer32 2 x 32-bit Timers
eUSCI_A0, eUSCI_A1, eUSCI_A2, eUSCI_A3 (UART, IrDA, SPI)
eUSCI_B0, eUSCI_B1, eUSCI_B2, eUSCI_B3 (I²C, SPI)

Timer 32

- MSP432 Timer 32
  - ARM (AMBA Compliant) timer
  - 2 – 32 bit timers
  - Configurable as 16 bit timers
  - 3 Modes
    - Free Running
    - Periodic
    - One-Shot
  - Interrupt generation capable
  - COUNT DOWN TIMER



Timer 32

- MSP432 Timer 32

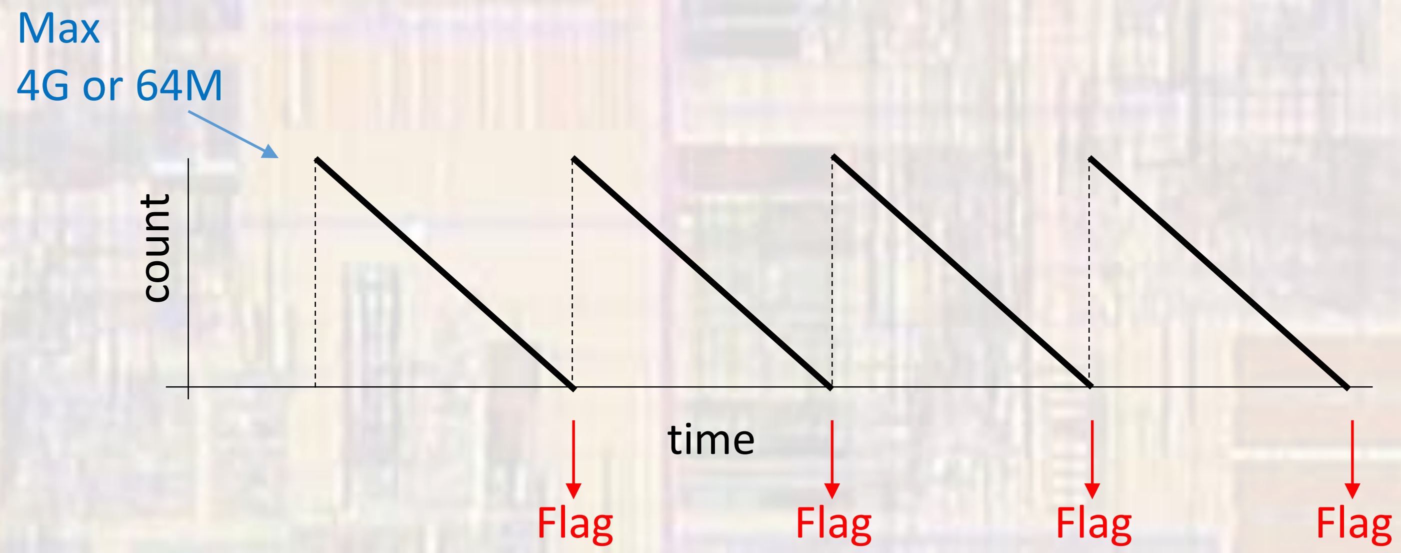
  - 32 bit mode
    - Max count value is 2^{32}
    - 4G counts
    - 4,294,967,296 counts

  - 16 bit mode
    - Max count value is 2^{16}
    - 64M counts
    - 65,536 counts

  - 48MHz clock and max 32 bit count
    - 4G counts / 48M counts/sec = 89.48 sec

Timer 32

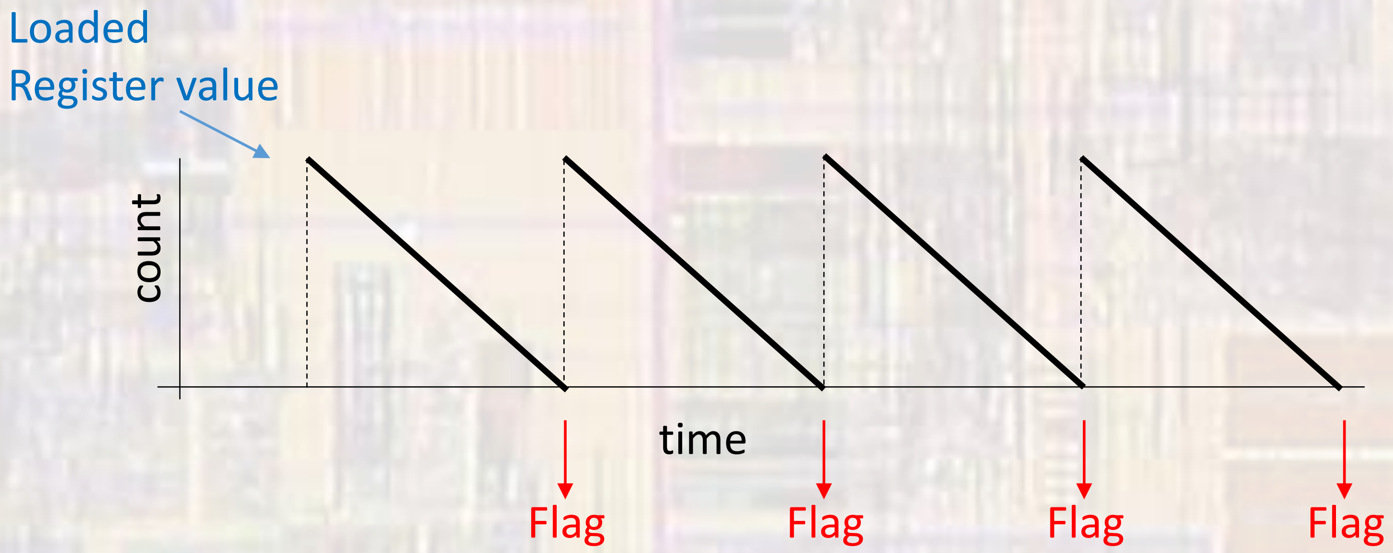
- MSP432 Timer 32
  - Free running mode
    - Counter wraps around to the maximum value after counting down to 0
    - Flag / Interrupt set when the counter wraps around
      - 1st clock edge after 0



Timer 32

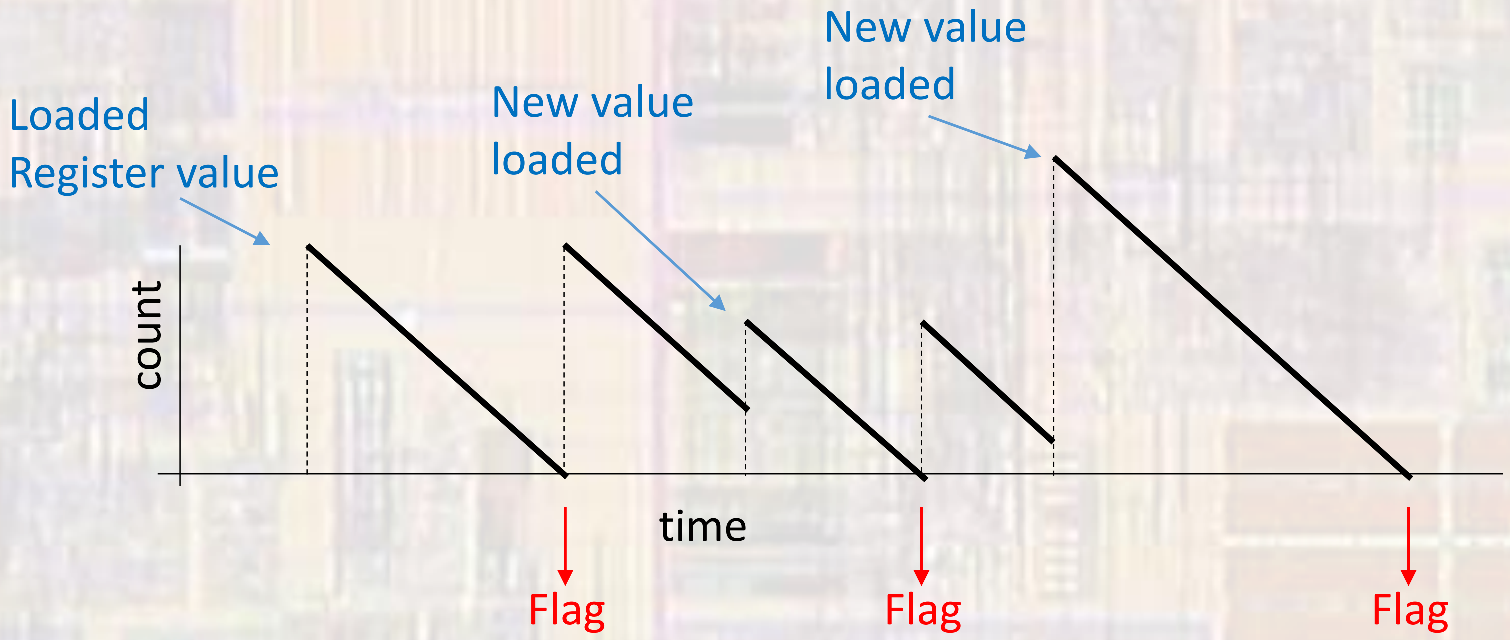
- MSP432 Timer 32

  - Periodic timer mode
    - Counter wraps around to the preset (loaded) register value after counting down to 0
    - Flag / Interrupt set when the counter wraps around
      - 1st clock edge after 0



Timer 32

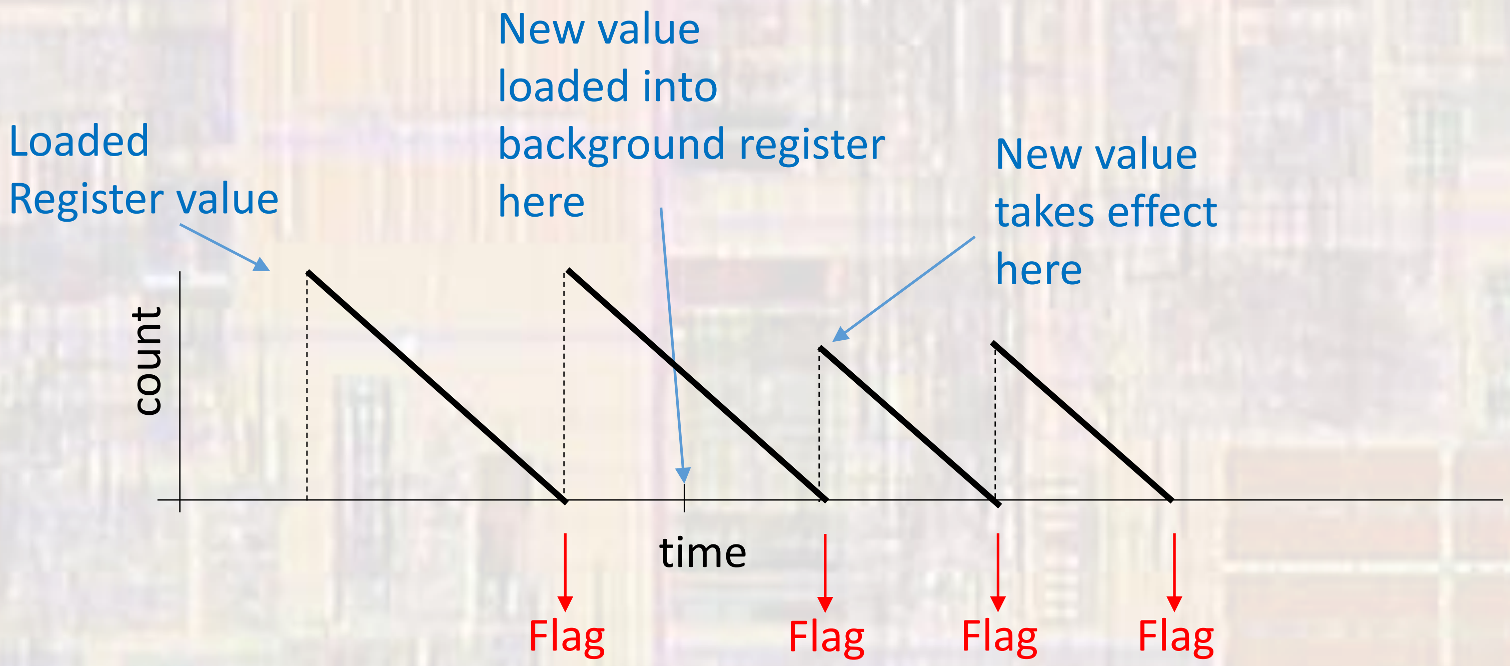
- MSP432 Timer 32
  - Periodic timer mode
    - Load Register
      - Counter immediately restarts the count at this value when loaded



Timer 32

- MSP432 Timer 32

  - Periodic timer mode
    - Background Load Register
      - Counter restarts the count at this value the next time the counter wraps around

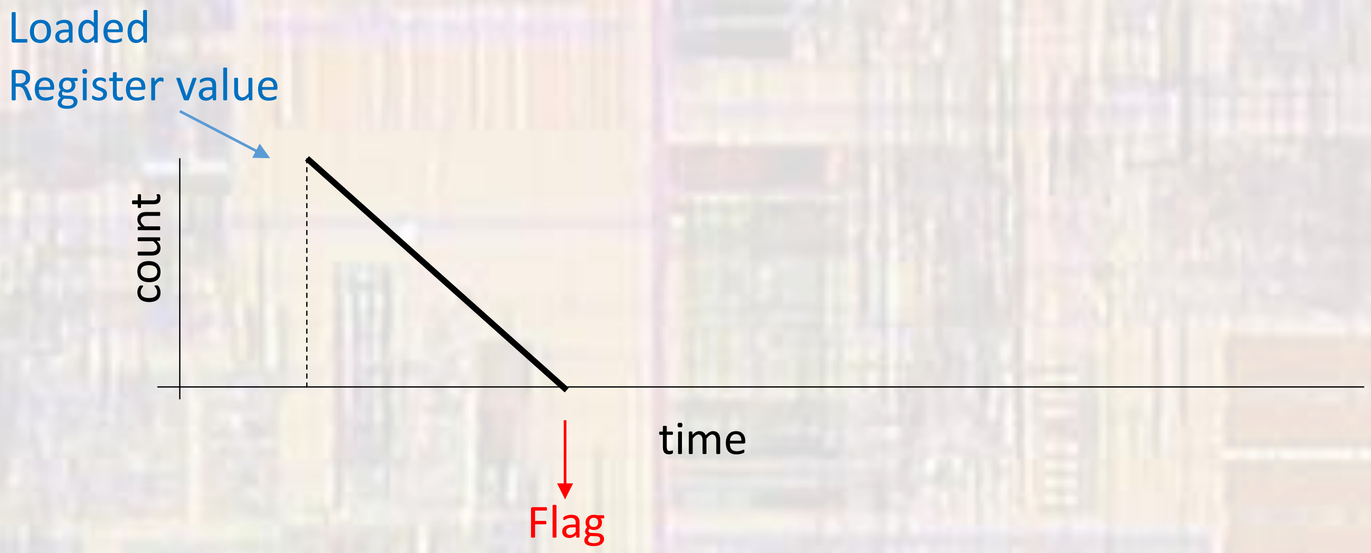


Loaded Register value
New value loaded into background register here
New value takes effect here
count
time
Flag
Flag
Flag
Flag

Timer 32

- MSP432 Timer 32

  - One-shot mode
    - Counter counts down to 0 and stops
    - Flag / Interrupt set when the counter reaches 0
      - 1st clock edge after 0 ??



Timer 32

- MSP432 Timer 32
  - Clock configuration
    - Mclk
    - Mclk/16
    - Mclk/256
    - Enable

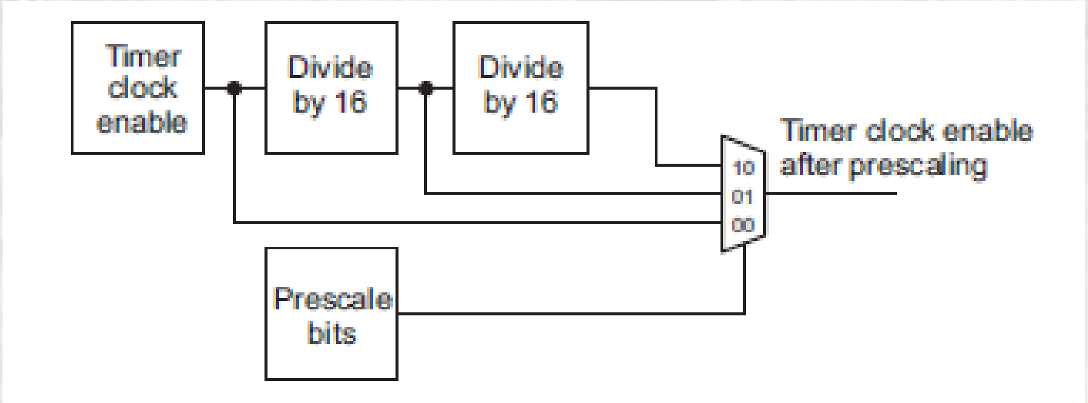


Figure 16-1. Prescale Clock Enable Generation

Timer 32

- MSP432 Timer 32

  - Interrupt configuration
    - Each timer creates an independent interrupt
      - TIMINT1 and TIMINT2
    - A third interrupt can be generated
      - TIMINTC which is the OR of TIMINT1 and TIMINT2

    - All interrupts are maskable

    - Interrupts must be cleared by software

Timer 32

- MSP432 Timer 32

TIMER32_1->LOAD, …
TIMER32_2-> BGLOAD

  - Register configuration

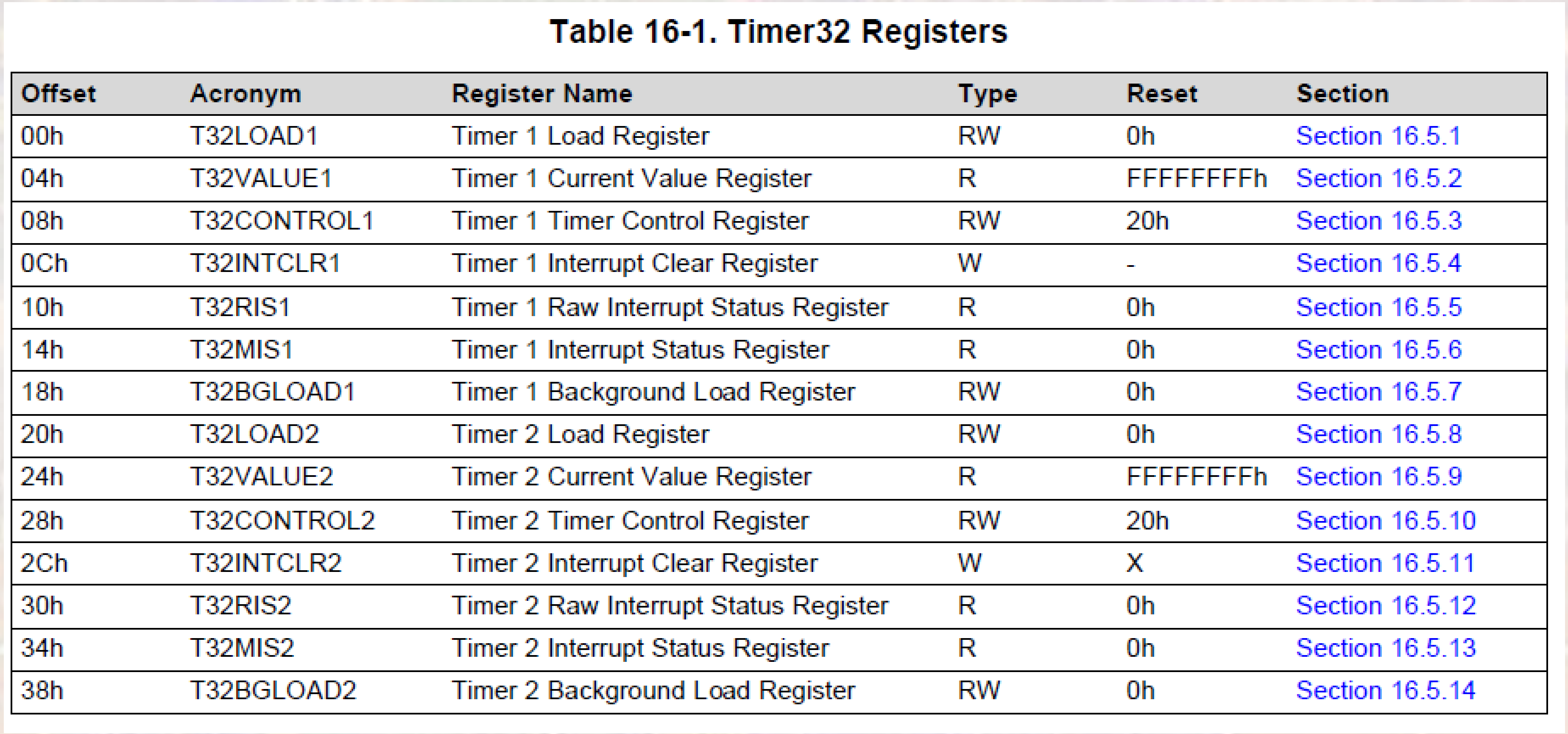


Table 16-1. Timer32 Registers

Offset	Acronym	Register Name	Type	Reset	Section
00h	T32LOAD1	Timer 1 Load Register	RW	0h	Section 16.5.1
04h	T32VALUE1	Timer 1 Current Value Register	R	FFFFFFFFh	Section 16.5.2
08h	T32CONTROL1	Timer 1 Timer Control Register	RW	20h	Section 16.5.3
0Ch	T32INTCLR1	Timer 1 Interrupt Clear Register	W	-	Section 16.5.4
10h	T32RIS1	Timer 1 Raw Interrupt Status Register	R	0h	Section 16.5.5
14h	T32MIS1	Timer 1 Interrupt Status Register	R	0h	Section 16.5.6
18h	T32BGLOAD1	Timer 1 Background Load Register	RW	0h	Section 16.5.7
20h	T32LOAD2	Timer 2 Load Register	RW	0h	Section 16.5.8
24h	T32VALUE2	Timer 2 Current Value Register	R	FFFFFFFFh	Section 16.5.9
28h	T32CONTROL2	Timer 2 Timer Control Register	RW	20h	Section 16.5.10
2Ch	T32INTCLR2	Timer 2 Interrupt Clear Register	W	X	Section 16.5.11
30h	T32RIS2	Timer 2 Raw Interrupt Status Register	R	0h	Section 16.5.12
34h	T32MIS2	Timer 2 Interrupt Status Register	R	0h	Section 16.5.13
38h	T32BGLOAD2	Timer 2 Background Load Register	RW	0h	Section 16.5.14

Timer 32

- MSP432 Timer 32

  - Load register for Periodic and One-shot operation
    - Counter is reset immediately when writing to this register

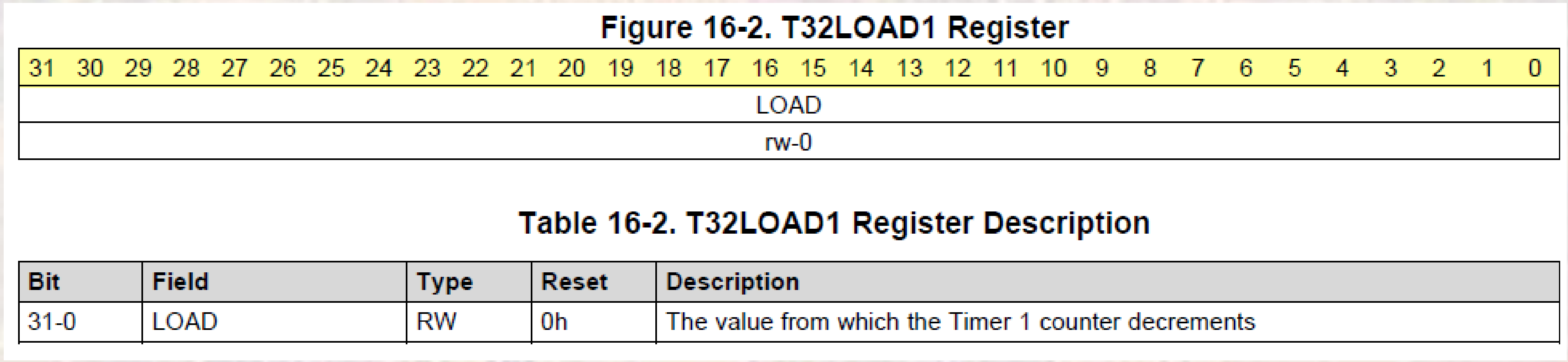


Figure 16-2. T32LOAD1 Register

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

Table 16-2. T32LOAD1 Register Description

Bit	Field	Type	Reset	Description
31-0	LOAD	RW	0h	The value from which the Timer 1 counter decrements

Timer 32

- MSP432 Timer 32
  - Current counter value

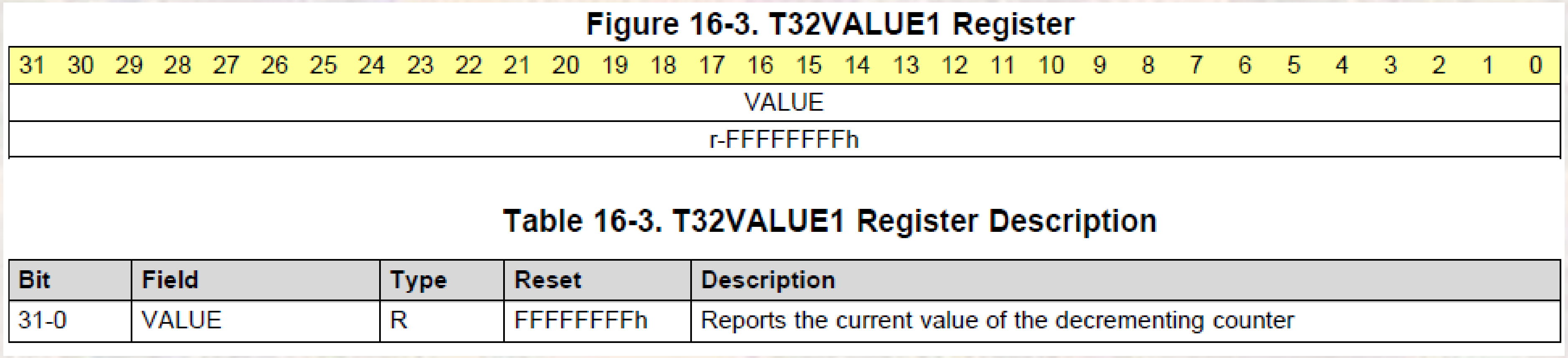


Figure 16-3. T32VALUE1 Register

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

Table 16-3. T32VALUE1 Register Description

Bit	Field	Type	Reset	Description
31-0	VALUE	R	FFFFFFFFh	Reports the current value of the decrementing counter

Timer 32

- MSP432 Timer 32
  - Control Register

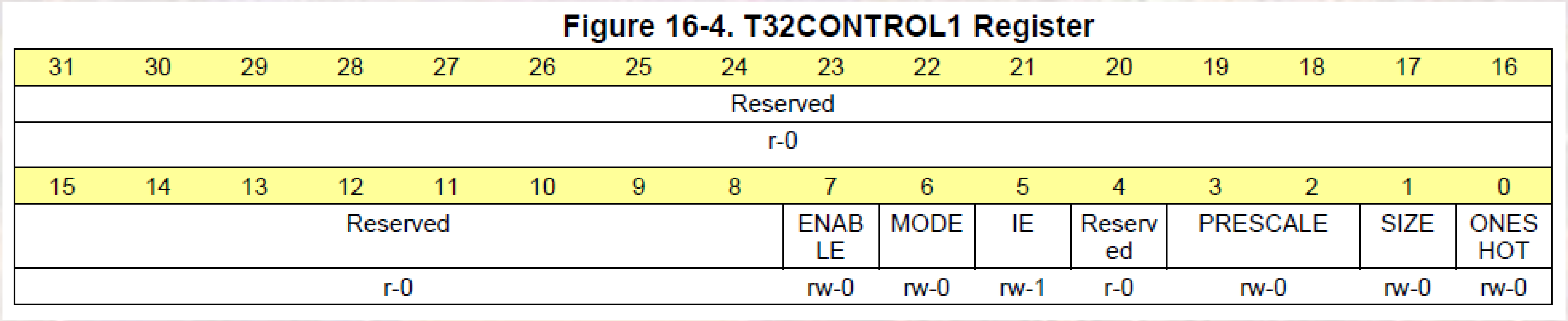


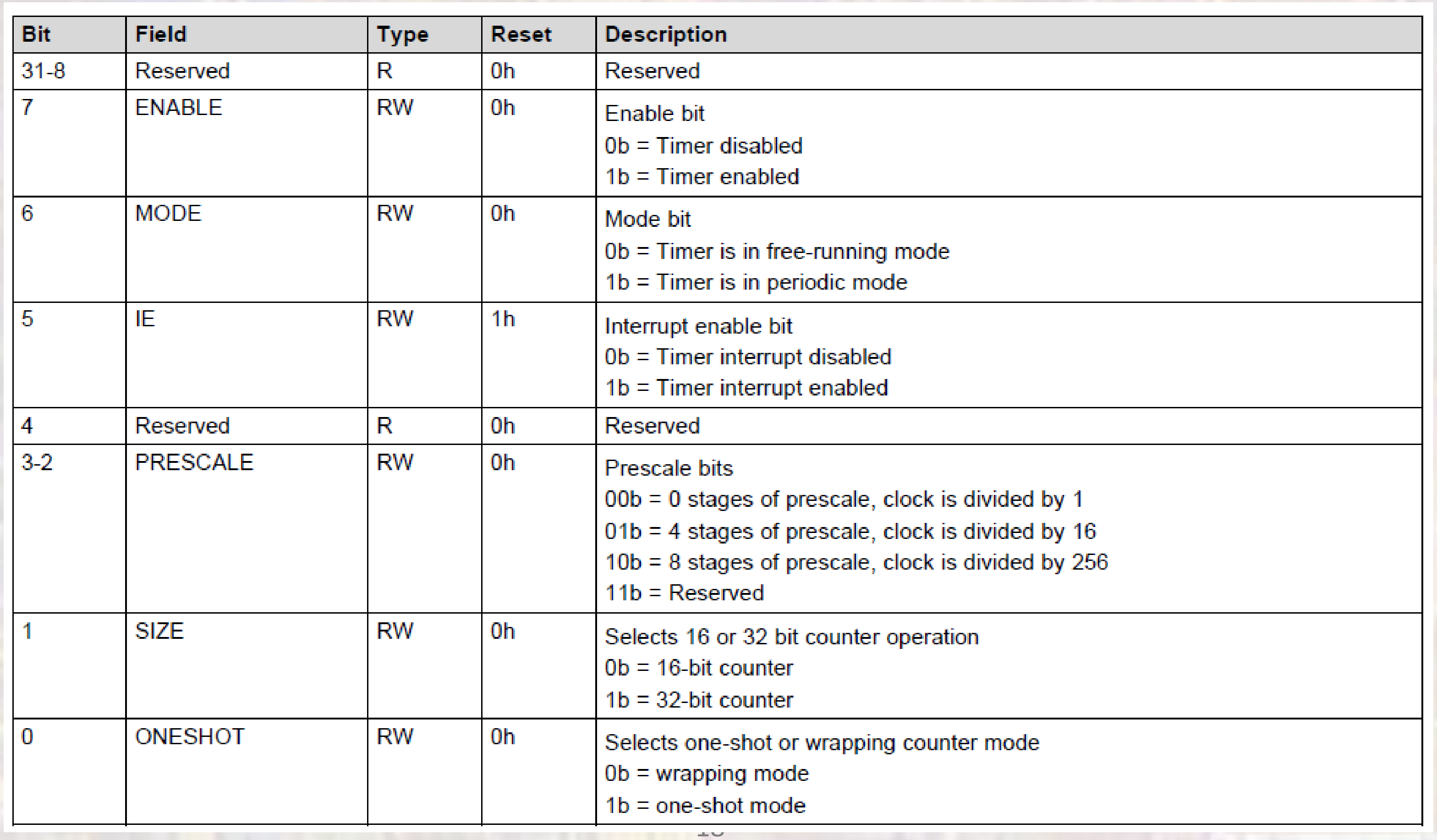
Figure 16-4. T32CONTROL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r-0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved								ENABLE	MODE	IE	Reserved	PRESCALE		SIZE	ONESHOT
r-0								rw-0	rw-0	rw-1	r-0	rw-0		rw-0	rw-0

Timer 32

- MSP432 Timer 32

  - Control Register



Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	Reserved
7	ENABLE	RW	0h	Enable bit 0b = Timer disabled 1b = Timer enabled
6	MODE	RW	0h	Mode bit 0b = Timer is in free-running mode 1b = Timer is in periodic mode
5	IE	RW	1h	Interrupt enable bit 0b = Timer interrupt disabled 1b = Timer interrupt enabled
4	Reserved	R	0h	Reserved
3-2	PRESCALE	RW	0h	Prescale bits 00b = 0 stages of prescale, clock is divided by 1 01b = 4 stages of prescale, clock is divided by 16 10b = 8 stages of prescale, clock is divided by 256 11b = Reserved
1	SIZE	RW	0h	Selects 16 or 32 bit counter operation 0b = 16-bit counter 1b = 32-bit counter
0	ONESHOT	RW	0h	Selects one-shot or wrapping counter mode 0b = wrapping mode 1b = one-shot mode

Timer 32

- MSP432 Timer 32
  - Interrupt Clear Register

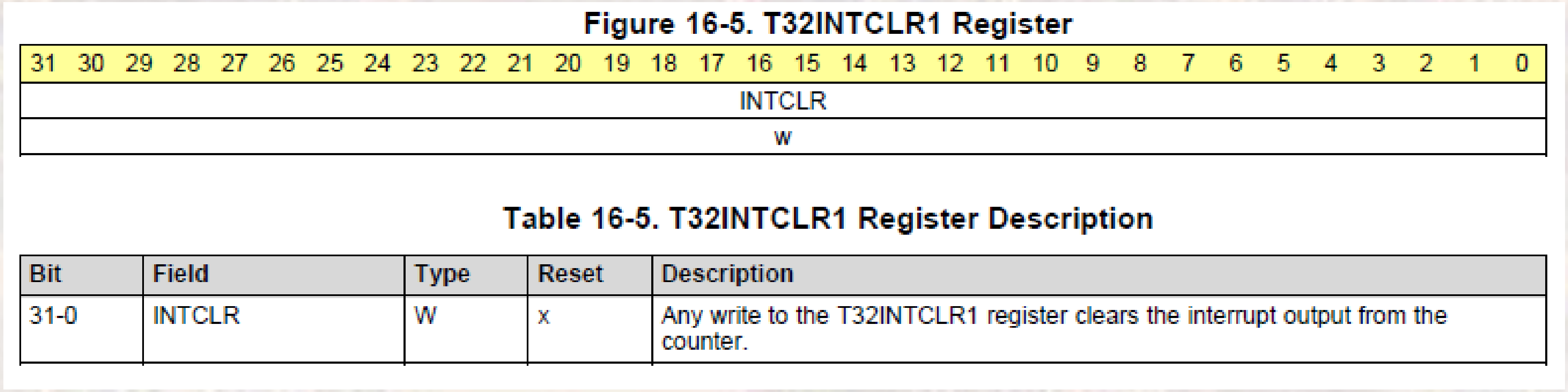


Figure 16-5. T32INTCLR1 Register

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

Table 16-5. T32INTCLR1 Register Description

Bit	Field	Type	Reset	Description
31-0	INTCLR	W	x	Any write to the T32INTCLR1 register clears the interrupt output from the counter.

Timer 32

- MSP432 Timer 32
  - RAW Interrupt Register
    - Holds the current interrupt value – independent of masking

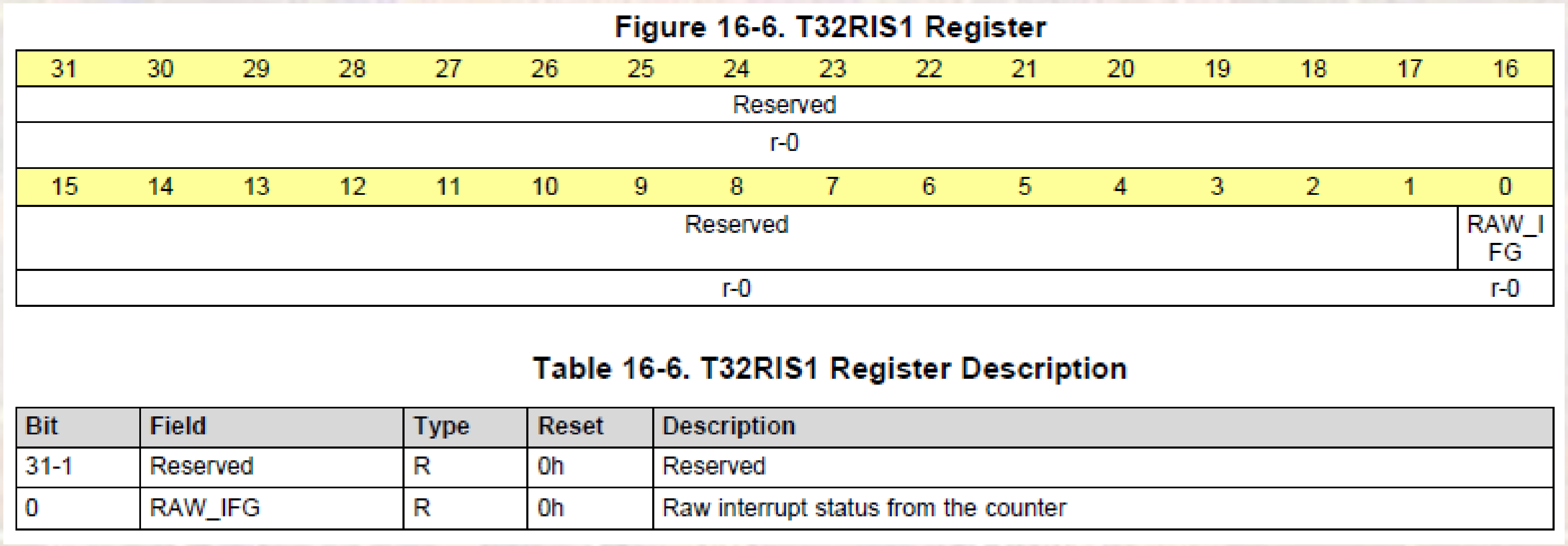


Figure 16-6. T32RIS1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r-0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved															RAW_IFG
r-0															r-0

Table 16-6. T32RIS1 Register Description

Bit	Field	Type	Reset	Description
31-1	Reserved	R	0h	Reserved
0	RAW_IFG	R	0h	Raw interrupt status from the counter

Timer 32

- MSP432 Timer 32
  - Masked Interrupt Register
    - Holds the current masked interrupt value

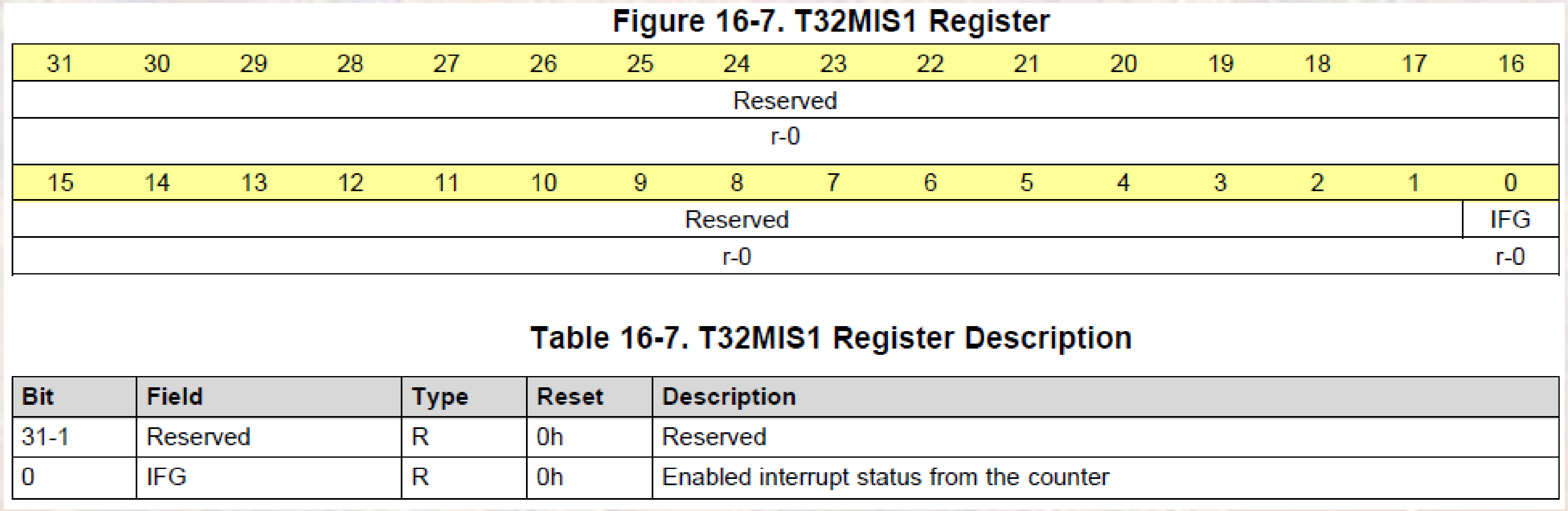


Figure 16-7. T32MIS1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r-0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved															IFG
r-0															r-0

Table 16-7. T32MIS1 Register Description

Bit	Field	Type	Reset	Description
31-1	Reserved	R	0h	Reserved
0	IFG	R	0h	Enabled interrupt status from the counter

Timer 32

- MSP432 Timer 32

  - Background Load Register
    - Updates the count value only when the counter wraps around

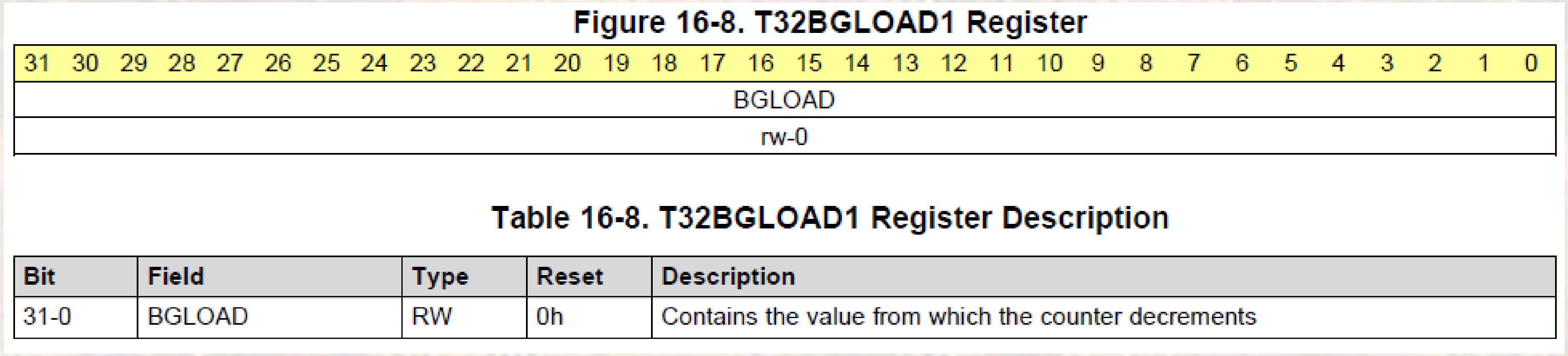


Figure 16-8. T32BGLOAD1 Register

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

Table 16-8. T32BGLOAD1 Register Description

Bit	Field	Type	Reset	Description
31-0	BGLOAD	RW	0h	Contains the value from which the counter decrements

Timer 32

- MSP432 Timer 32
  - Similar set of registers for timer 2
  - WHAT ABOUT TIMINTC – not in the document

Timer 32

- MSP432 Timer 32

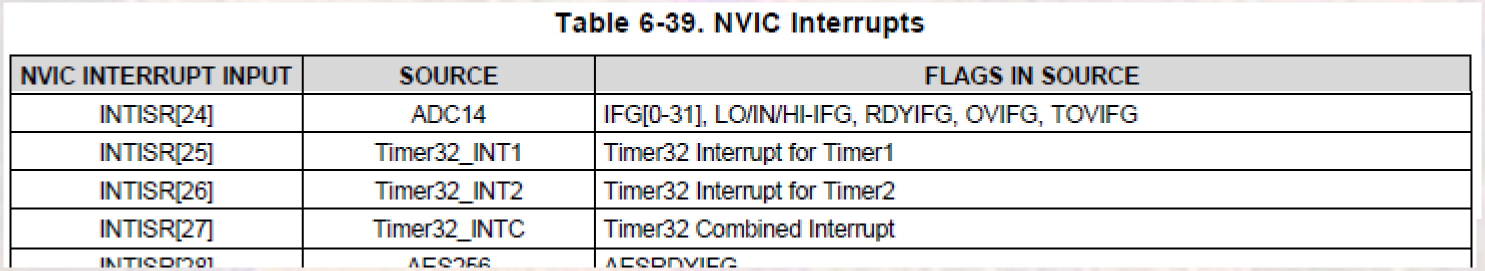


Table 6-39. NVIC Interrupts

NVIC INTERRUPT INPUT	SOURCE	FLAGS IN SOURCE
INTISR[24]	ADC14	IFG[0-31], LO/IN/HI-IFG, RDYIFG, OVIFG, TOVIFG
INTISR[25]	Timer32_INT1	Timer32 Interrupt for Timer1
INTISR[26]	Timer32_INT2	Timer32 Interrupt for Timer2
INTISR[27]	Timer32_INTC	Timer32 Combined Interrupt
INTISR[28]	AES256	AESRDYIFG

Timer 1

```
NVIC->IP[25] |= 0x?0;               // set priority
NVIC->ISER[0] |= 0x02000000;            // enable interrupt
```

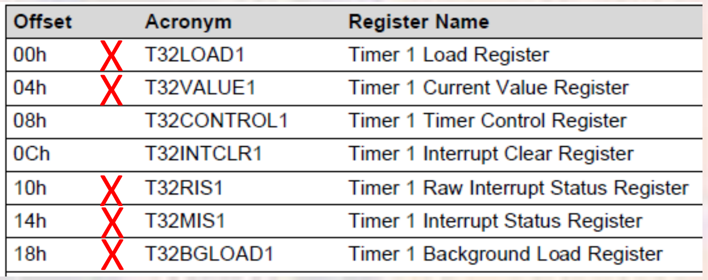
Timer 2

```
NVIC->IP[26] |= 0x?0;               // set priority
NVIC->ISER[0] |= 0x04000000;            // enable interrupt
```

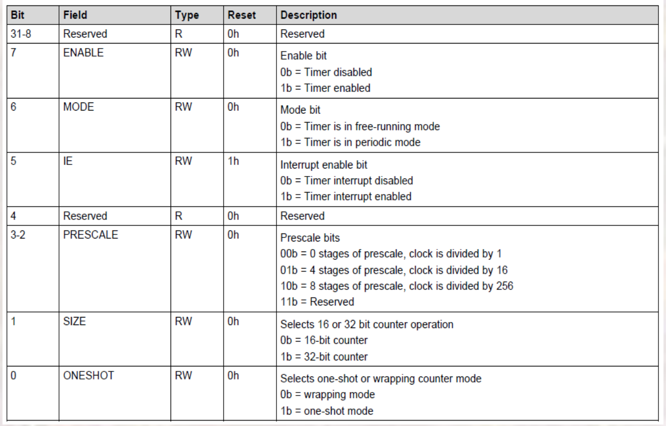
```
T32_INT1_IRQHandler,          /* T32_INT1 ISR         */
T32_INT2_IRQHandler,          /* T32_INT2 ISR         */
T32_INTC_IRQHandler,          /* T32_INTC ISR         */
```

Timer 32

- MSP432 Timer 32
  - Example 1
    - Setup the Timer in 32 bit periodic mode with interrupts enabled and running at MCLK/16
    - Write an interrupt handler to print the global variable "val"



Offset		Acronym	Register Name
00h	X	T32LOAD1	Timer 1 Load Register
04h	X	T32VALUE1	Timer 1 Current Value Register
08h		T32CONTROL1	Timer 1 Timer Control Register
0Ch		T32INTCLR1	Timer 1 Interrupt Clear Register
10h	X	T32RIS1	Timer 1 Raw Interrupt Status Register
14h	X	T32MIS1	Timer 1 Interrupt Status Register
18h	X	T32BGLOAD1	Timer 1 Background Load Register



Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	Reserved
7	ENABLE	RW	0h	Enable bit 0b = Timer disabled 1b = Timer enabled
6	MODE	RW	0h	Mode bit 0b = Timer is in free-running mode 1b = Timer is in periodic mode
5	IE	RW	1h	Interrupt enable bit 0b = Timer interrupt disabled 1b = Timer interrupt enabled
4	Reserved	R	0h	Reserved
3-2	PRESCALE	RW	0h	Prescale bits 00b = 0 stages of prescale, clock is divided by 1 01b = 4 stages of prescale, clock is divided by 16 10b = 8 stages of prescale, clock is divided by 256 11b = Reserved
1	SIZE	RW	0h	Selects 16 or 32 bit counter operation 0b = 16-bit counter 1b = 32-bit counter
0	ONESHOT	RW	0h	Selects one-shot or wrapping counter mode 0b = wrapping mode 1b = one-shot mode

```
// ctrl
// xxxx xxxx xxxx xxxx xxxx xxxx    en  per  int  x  /16  32b  wr
// 0000 0000 0000 0000 0000 0000    1     1    1  0   01    1   0
TIMER32_1->CONTROL = 0x000000E6;


NVIC->IP[25] |= 0x20; // Set a priority
NVIC->ISER[0] |= 0x02000000; // ISER0 starts at 0
```

Enable Periodic Interrupt / 16 32 bit Wrapping

1 0 1 x 01 1 0

Timer 32

- MSP432 Timer 32
  - Example 1

```c
/*
 * timer32_example.c
 *
 *  Created on: Aug 4, 2018
 *      Author: johnsontimoj
 */
//////////////////////////////////
//
// use Timer 32 in interrupt mode
//
// input None
//
// output - LCD
//
///////////////////////////////////

// includes
#include "msp.h"
#include <stdio.h>
#include "msoe_lib_all.h"

// prototypes
void T32_setup(void);
void NVIC_setup(void);
void lcd_setup(void);

int val;    // global variable
val = 0;
```

```c
int main(void){
    // setup routines
    lcd_setup();
    T32_setup();
    NVIC_setup();

    _enable_interrupts();

    // hang out
    while(1){
        __delay_cycles(1500000);
    } // end while

   return 0;
}

void T32_setup(void){
    // Using MCLK/16 for timing
    // Free running
    // Interrupts
    //
    // ctrl
    // xxxx xxxx xxxx xxxx xxxx xxxx    en  per  int  x  /16  32b  wr
    // 0000 0000 0000 0000 0000 0000    1     1    1  0   01   1    0
    TIMER32_1->CONTROL = 0x000000E6;
    // load
    TIMER32_1->LOAD = 187500;   // 3,000,000 / 16 --> 1 sec
    //
    TIMER32_1->INTCLR = 0x0;        // clear flag

    return;
}
```

```c
void lcd_setup(void){
    // setup LCD
    LCD_Config();
    LCD_clear();
    LCD_contrast(9);

    return;
}

void T32_INT1_IRQHandler(void){
    val++;
    TIMER32_1->INTCLR = 0x0;        // clear flag
    // need to put print in ISR otherwise it never
    // gets a chance to run
    LCD_goto_xy(0,0);
    LCD_print_udec10(val);

    return;
}

void NVIC_setup(void){
    // setup NVIC
    // Enable Timer 32
    // Note: Timer 32 - 1  is INTISR(25)
    NVIC->IP[25] |= 0x20; // Set a priority
    NVIC->ISER[0] |= 0x02000000; // ISER0 starts at 0

    return;
}
```

Timer 32

- MSP432 Timer 32
  - Example 2
    - Use Timer32 to toggle Pin 2 at 1Hz with a 70% duty cycle

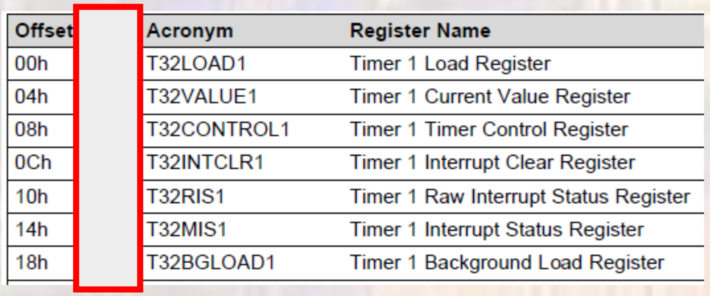
3MHz → 333.33ns/clk
1Hz → 1s/cycle
70% duty cycle → 700ms high, 300ms low
700ms → 2,100,000 clks
300ms → 900,000 clks

→ 32 bit Periodic mode

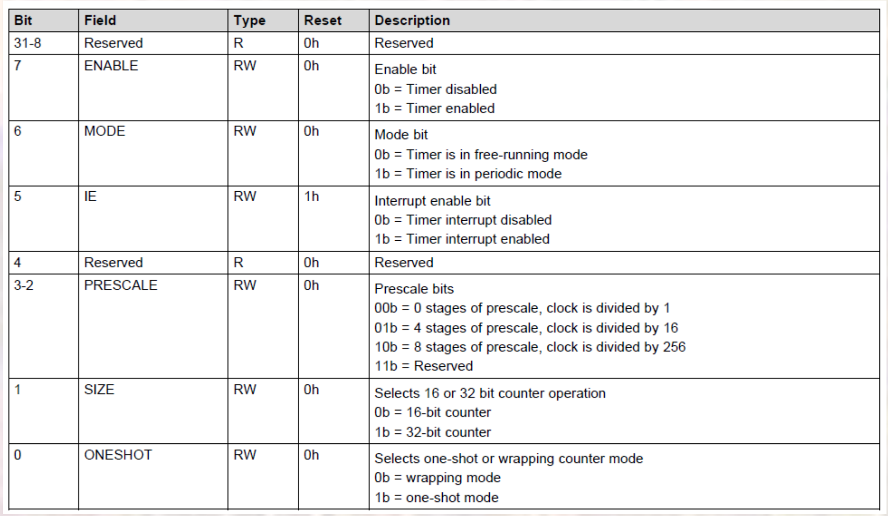
Use the interrupt to toggle the pin and reset the BG load value

Timer 32

- MSP432 Timer 32
  - Example 2



Offset		Acronym	Register Name
00h		T32LOAD1	Timer 1 Load Register
04h		T32VALUE1	Timer 1 Current Value Register
08h		T32CONTROL1	Timer 1 Timer Control Register
0Ch		T32INTCLR1	Timer 1 Interrupt Clear Register
10h		T32RIS1	Timer 1 Raw Interrupt Status Register
14h		T32MIS1	Timer 1 Interrupt Status Register
18h		T32BGLOAD1	Timer 1 Background Load Register

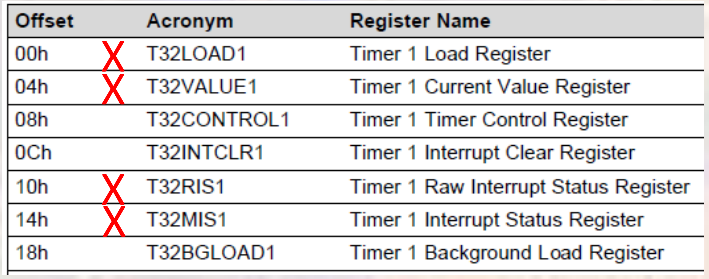


Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	Reserved
7	ENABLE	RW	0h	Enable bit 0b = Timer disabled 1b = Timer enabled
6	MODE	RW	0h	Mode bit 0b = Timer is in free-running mode 1b = Timer is in periodic mode
5	IE	RW	1h	Interrupt enable bit 0b = Timer interrupt disabled 1b = Timer interrupt enabled
4	Reserved	R	0h	Reserved
3-2	PRESCALE	RW	0h	Prescale bits 00b = 0 stages of prescale, clock is divided by 1 01b = 4 stages of prescale, clock is divided by 16 10b = 8 stages of prescale, clock is divided by 256 11b = Reserved
1	SIZE	RW	0h	Selects 16 or 32 bit counter operation 0b = 16-bit counter 1b = 32-bit counter
0	ONESHOT	RW	0h	Selects one-shot or wrapping counter mode 0b = wrapping mode 1b = one-shot mode

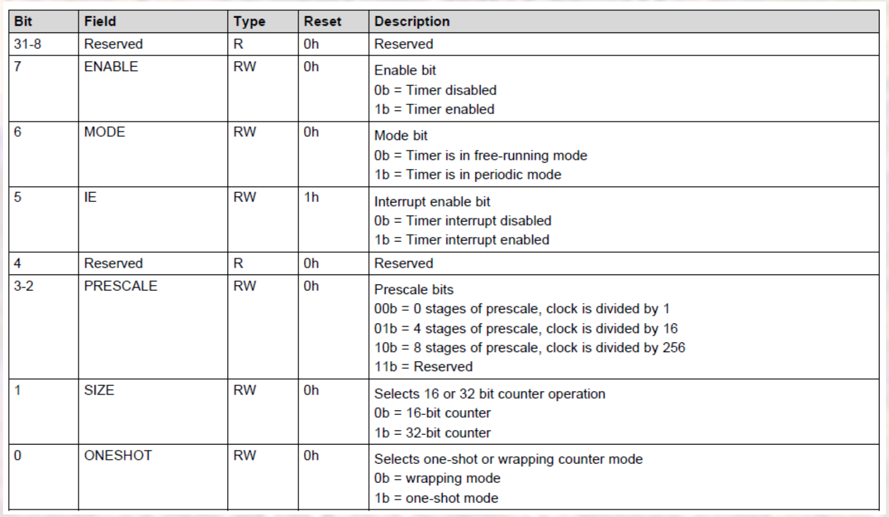
This must be first
why?

Timer 32

- MSP432 Timer 32
  - Example 2



Offset		Acronym	Register Name
00h	X	T32LOAD1	Timer 1 Load Register
04h	X	T32VALUE1	Timer 1 Current Value Register
08h		T32CONTROL1	Timer 1 Timer Control Register
0Ch		T32INTCLR1	Timer 1 Interrupt Clear Register
10h	X	T32RIS1	Timer 1 Raw Interrupt Status Register
14h	X	T32MIS1	Timer 1 Interrupt Status Register
18h		T32BGLOAD1	Timer 1 Background Load Register



Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	Reserved
7	ENABLE	RW	0h	Enable bit 0b = Timer disabled 1b = Timer enabled
6	MODE	RW	0h	Mode bit 0b = Timer is in free-running mode 1b = Timer is in periodic mode
5	IE	RW	1h	Interrupt enable bit 0b = Timer interrupt disabled 1b = Timer interrupt enabled
4	Reserved	R	0h	Reserved
3-2	PRESCALE	RW	0h	Prescale bits 00b = 0 stages of prescale, clock is divided by 1 01b = 4 stages of prescale, clock is divided by 16 10b = 8 stages of prescale, clock is divided by 256 11b = Reserved
1	SIZE	RW	0h	Selects 16 or 32 bit counter operation 0b = 16-bit counter 1b = 32-bit counter
0	ONESHOT	RW	0h	Selects one-shot or wrapping counter mode 0b = wrapping mode 1b = one-shot mode

```
 // xxxx xxxx xxxx xxxx xxxx xxxx    en  per  int  x   /1  32b  wr
    // 0000 0000 0000 0000 0000 0000    1     1    1  0   00   1    0
    TIMER32_1->CONTROL = 0x000000E2;

void T32_INT1_IRQHandler(void){
    TIMER32_1->INTCLR = 0x0;            // clear flag
    if (flg == 1){                      // output goes low
        P6->OUT &= ~0x01;
        TIMER32_1->BGLOAD = 2100000;// preset for high time
        flg = 0;
    } else {
        P6->OUT |= 0x01;                // output goes high
        TIMER32_1->BGLOAD = 900000;// preset for low time
        flg = 1;
    }
}
```

Enable Periodic Interrupt / 1 32 bit Wrapping

1 1 1 x 00 1 0

This must be first
why?

Timer 32

- MSP432 Timer 32
  - Example 2

```c
/*
 * timer32_example2.c
 *
 *  Created on: Aug 7, 2018
 *      Author: johnsontimoj
 */
//////////////////////////////////
//
// use Timer 32 in interrupt mode
//
// input None
//
// output - LED running at 1Hz, 70% duty
//
///////////////////////////////////

// includes
#include "msp.h"
#include <stdio.h>

// prototypes
void led_setup(void);
void T32_setup(void);
void NVIC_setup(void);

int flg;    // global variable
flg = 0;
```

```c
int main(void){
    // setup routines
    led_setup();
    T32_setup();
    NVIC_setup();

    _enable_interrupts();

    // hang out
    while(1){
        __delay_cycles(1500000);
    } // end while

   return 0;
}

void T32_setup(void){
    // Using MCLK/1 for timing
    // Free running
    // Interrupts
    //
    // ctrl
    // xxxx xxxx xxxx xxxx xxxx xxxx    en  per  int  x   /1  32b  wr
    // 0000 0000 0000 0000 0000 0000    1    1    1   0   00   1    0
    TIMER32_1->CONTROL = 0x000000E2;
    // load
    TIMER32_1->LOAD = 2100000;    // 3,000,000 / 1 --> 0.7 sec initial cnt
    TIMER32_1->BGLOAD = 2100000;    // 3,000,000 / 1 --> 70% duty cycle
    //
    TIMER32_1->INTCLR = 0x0;         // clear flag

    return;
}
```

Very important – why?

```c
void led_setup(void){
    // setup LED
    // Pin 2 is P6.0
    P6->DIR |= 0x01;
    P6->OUT &= ~0x01;   // initialize to off

    return;
}

void T32_INT1_IRQHandler(void){
    TIMER32_1->INTCLR = 0x0;          // clear flag
    if (flg == 1){                    // output goes low
        P6->OUT &= ~0x01;
        TIMER32_1->BGLOAD = 2100000;// preset for high time
        flg = 0;
    } else {
        P6->OUT |= 0x01;              // output goes high
        TIMER32_1->BGLOAD = 900000;// preset for low time
        flg = 1;
    }
}

void NVIC_setup(void){
    // setup NVIC
    // Enable Timer 32
    // Note: Timer 32 - 1  is INTISR(25)
    NVIC->IP[25] |= 0x20; // Set a priority
    NVIC->ISER[0] |= 0x02000000; // ISER0 starts at 0

    return;
}
```