MSP430 Clock System and Timer

SWARM

College of Computer and Information Science,
Northeastern University

References:

Texas Instruments, "MSP430x1xx Family *User's Guide"*

Texas Instruments, "MSP430x15x, MSP430x16x, MSP430x161x MIXED SIGNAL MICROCONTROLLER",

Outline

- MSP430 basic clock module
- MSP430 Timer A
- Timer A examples

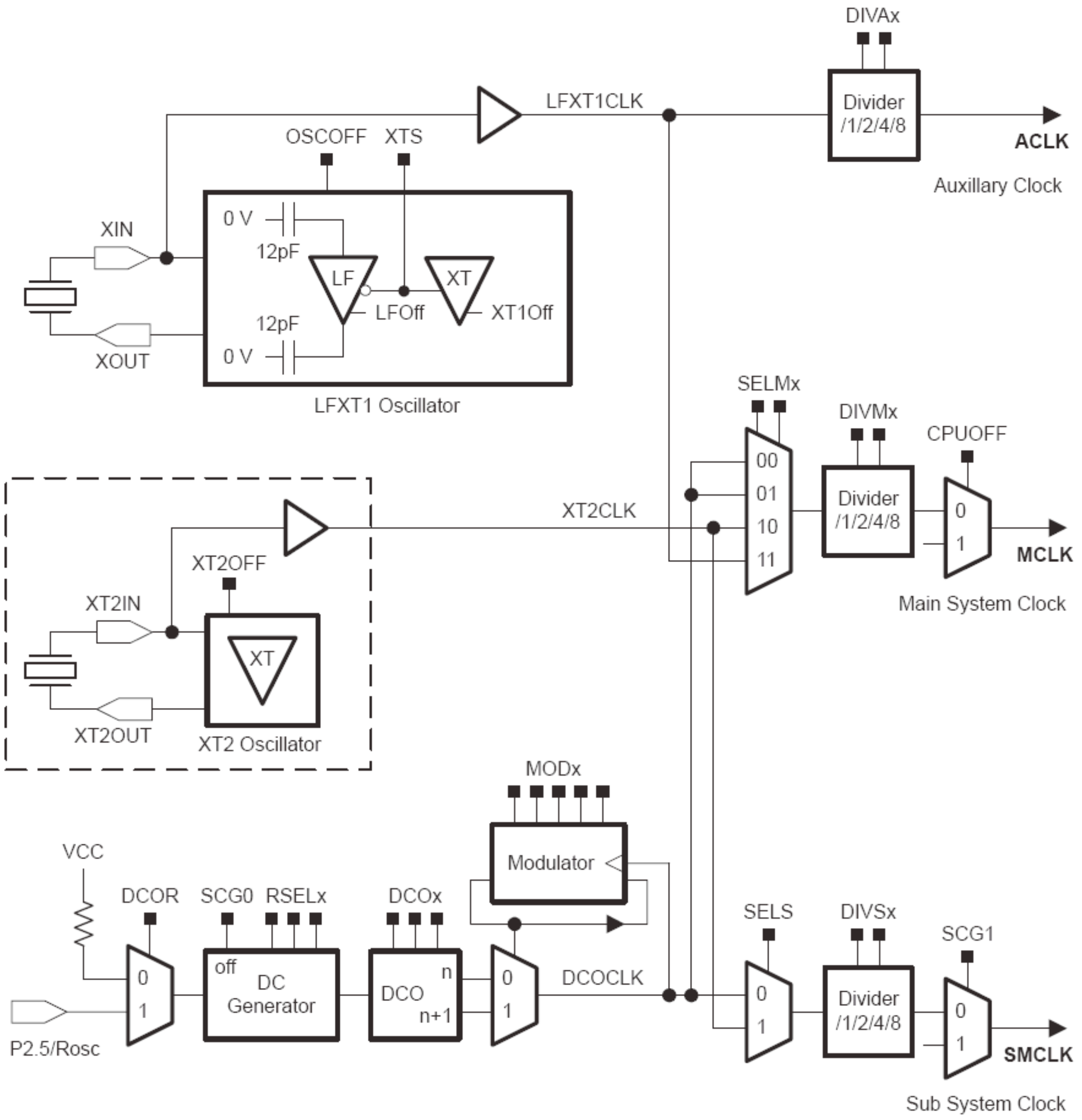
MSP430 Basic Clock Module

- Clock Sources:
  - **LFXT1CLK** : Low-frequency/high-frequency oscillator
  - **XT2CLK** : Optional high-frequency oscillator
  - **DCOCLK** : Internal digitally controlled oscillator (DCO)

- Tmote Sky Configuration:
  - **LFXT1CLK** : 32.768KHz crystal
  - **XT2CLK** : N/A
  - **DCOCLK** : Built-in DCO with configurable range from <100KHz to 4MHz

MSP430 Basic Clock Module

- Clock Signals:
  - **ACLK**: Auxiliary clock. The signal is sourced from LFXT1CLK with a divider of 1, 2, 4, or 8. (The calibration program for the serial link sets the divider to 4, but after the calibration it can be changed to any other values.) ACLK can be used as the clock signal for Timer A and Timer B.
  - **MCLK**: Master clock. The signal can be sourced from LFXT1CLK, XT2CLK (if available), or DCOCLK with a divider of 1, 2, 4, or 8. MCLK is used by the CPU and system.
  - **SMCLK**: Sub-main clock. The signal is sourced from either XT2CLK (if available), or DCOCLK with a divider of 1, 2, 4, or 8. SMCLK can be used as the clock signal for Timer A and Timer B.



DIVAx
LFXT1CLK
Divider /1/2/4/8
ACLK
Auxillary Clock
OSCOFF
XTS
XIN
0 V
12pF
LF
XT
LFOff
XT1Off
XOUT
LFXT1 Oscillator
SELMx
DIVMx
CPUOFF
00
01
10
11
MCLK
Main System Clock
XT2CLK
XT2OFF
XT2IN
XT2OUT
XT2 Oscillator
MODx
Modulator
VCC
DCOR
SCG0
RSELx
DCOx
off
DC Generator
DCO
n
n+1
DCOCLK
P2.5/Rosc
SELS
DIVSx
SCG1
SMCLK
Sub System Clock

Clock System Registers

DCOCTL, DCO Control Register

7	6	5	4	3	2	1	0
	DCOx				**MODx**		
rw−0	rw−1	rw−1	rw−0	rw−0	rw−0	rw−0	rw−0

DCOx	Bits 7-5	DCO frequency select. These bits select which of the eight discrete DCO frequencies of the RSELx setting is selected.
MODx	Bits 4-0	Modulator selection. These bits define how often the f_{DCO+1} frequency is used within a period of 32 DCOCLK cycles. During the remaining clock cycles (32−MOD) the f_{DCO} frequency is used. Not useable when DCOx=7.

BCSCTL1, Basic Clock System Control Register 1

7	6	5	4	3	2	1	0
XT2OFF	XTS	DIVAx		XT5V	RSELx		
rw–(1)	rw–(0)	rw–(0)	rw–(0)	rw–0	rw–1	rw–0	rw–0

XT2OFF	Bit 7	XT2 off. This bit turns off the XT2 oscillator 0 XT2 is on 1 XT2 is off if it is not used for MCLK or SMCLK.
XTS	Bit 6	LFXT1 mode select. 0 Low frequency mode 1 High frequency mode
DIVAx	Bits 5-4	Divider for ACLK 00 /1 01 /2 10 /4 11 /8
XT5V	Bit 3	Unused. XT5V should always be reset.
RSELx	Bits 2-0	Resistor Select. The internal resistor is selected in eight different steps. The value of the resistor defines the nominal frequency. The lowest nominal frequency is selected by setting RSELx=0.

BCSCTL2, Basic Clock System Control Register 2

7	6	5	4	3	2	1	0
SELMx		DIVMx		SELS	DIVSx		DCOR
rw−(0)	rw−(0)	rw−(0)	rw−(0)	rw−0	rw−0	rw−0	rw−0

SELMx Bits 7-6 Select MCLK. These bits select the MCLK source.

00	DCOCLK
01	DCOCLK
10	XT2CLK when XT2 oscillator present on-chip. LFXT1CLK when XT2 oscillator not present on-chip.
11	LFXT1CLK

DIVMx BitS 5-4 Divider for MCLK

00	/1
01	/2
10	/4
11	/8

SELS Bit 3 Select SMCLK. This bit selects the SMCLK source.

0	DCOCLK
1	XT2CLK when XT2 oscillator present on-chip. LFXT1CLK when XT2 oscillator not present on-chip.

DIVSx BitS 2-1 Divider for SMCLK

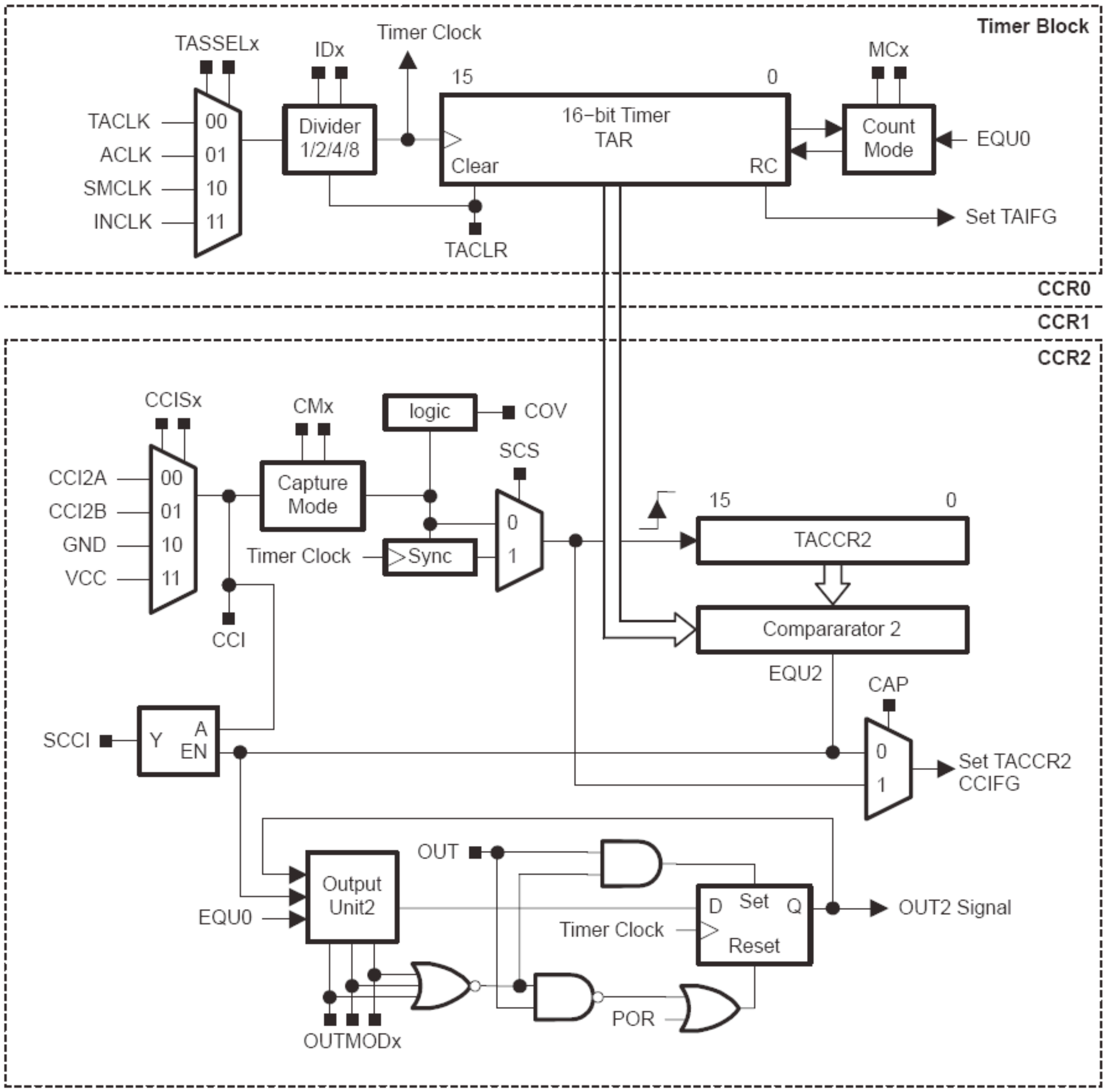
00	/1
01	/2
10	/4
11	/8

DCOR Bit 0 DCO resistor select

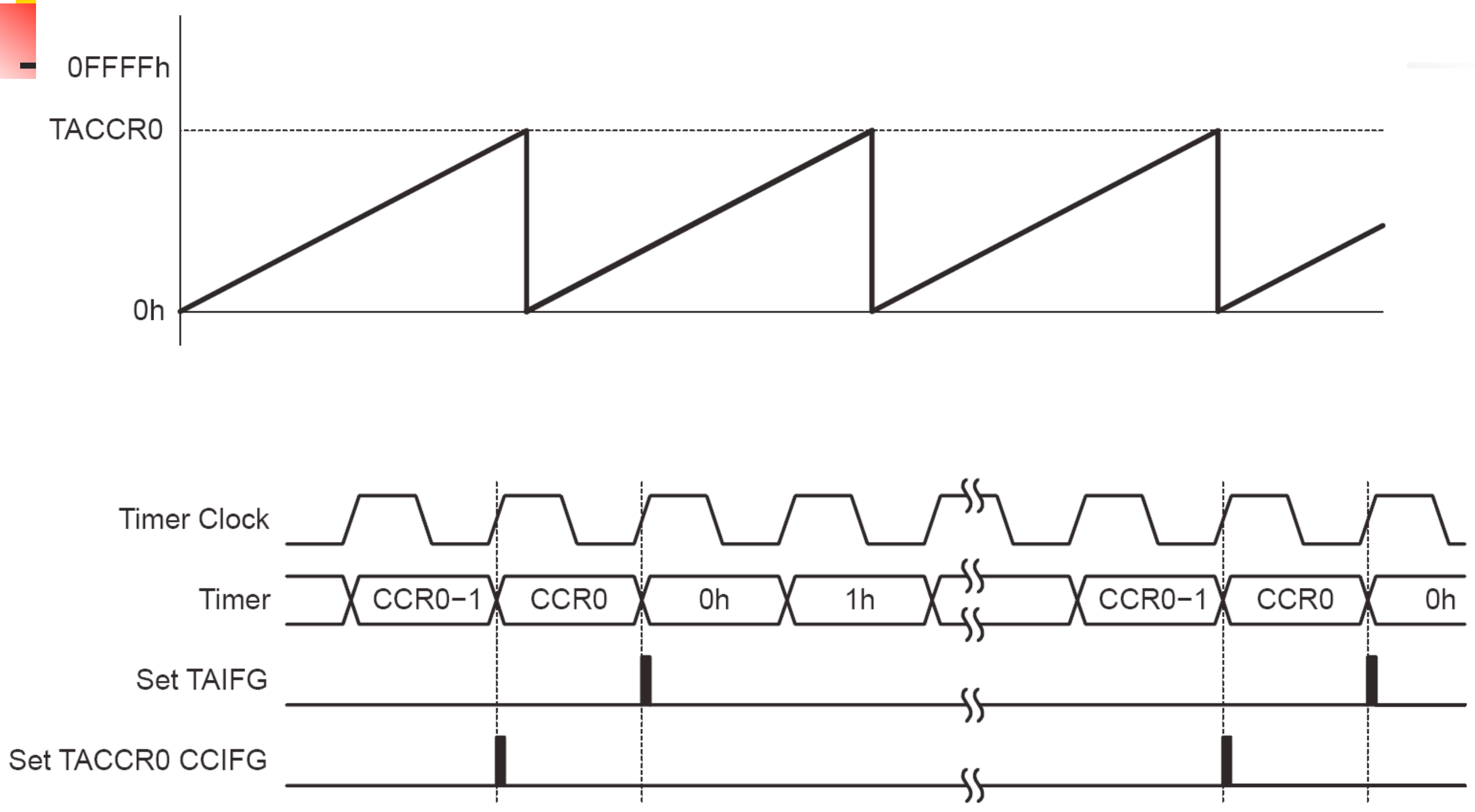
0	Internal resistor
1	External resistor

MSP430 Timer_A

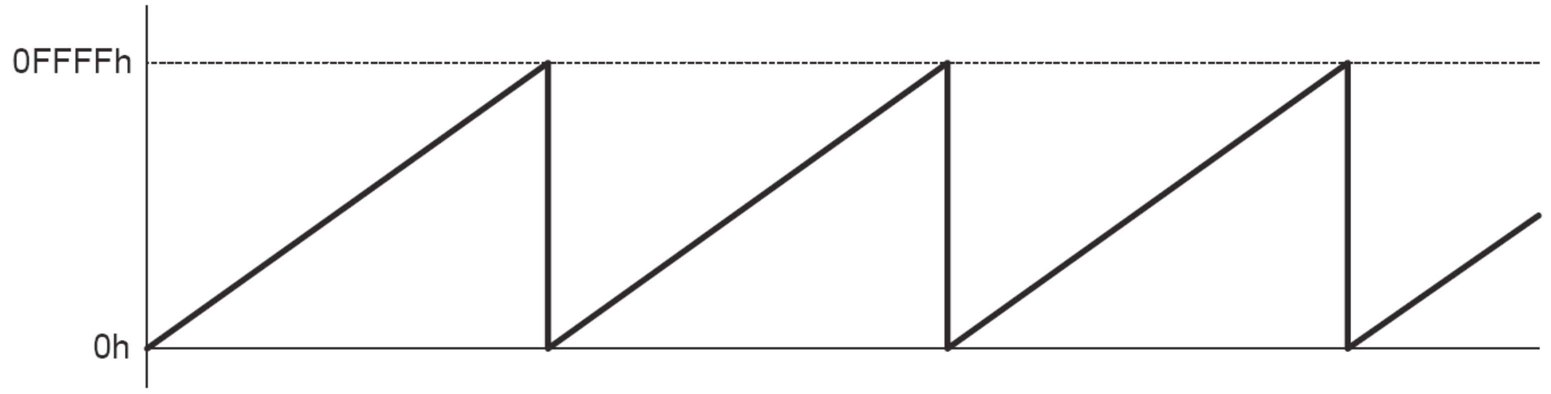
- A 16-bit counter
- 4 modes of operation – Stop, Up, Continuous, Up/Down
- 3 capture/compare registers (CCRx)
- 2 interrupt vectors – TACCR0 and TAIV

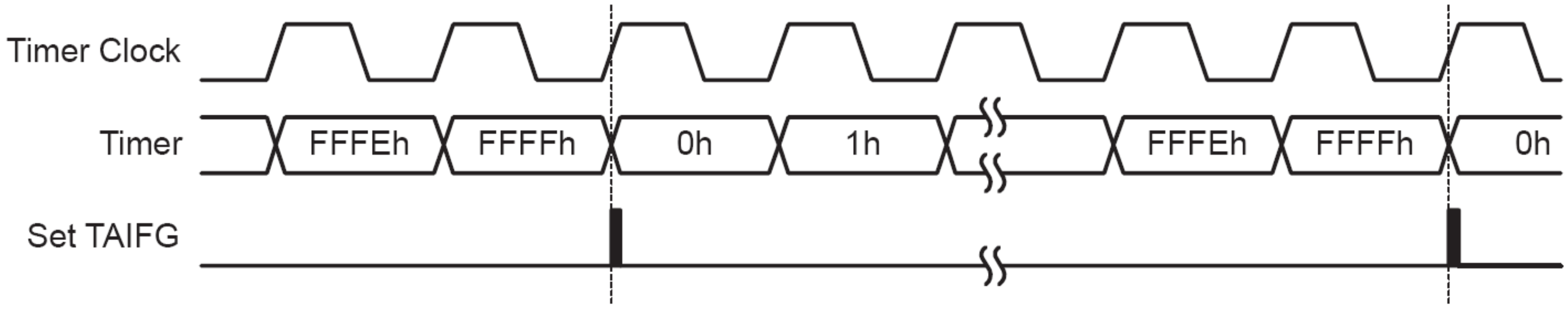

Timer Block
TASSELx
IDx
Timer Clock
15
0
MCx
TACLK
ACLK
SMCLK
INCLK
00
01
10
11
Divider 1/2/4/8
16−bit Timer TAR
Clear
RC
Count Mode
EQU0
Set TAIFG
TACLR
CCR0
CCR1
CCR2
CCISx
CMx
logic
COV
SCS
CCI2A
CCI2B
GND
VCC
Capture Mode
Timer Clock
Sync
0
1
15
0
TACCR2
Comparator 2
CCI
EQU2
CAP
SCCI
Y
A
EN
Set TACCR2 CCIFG
OUT
Output Unit2
EQU0
D
Set
Q
OUT2 Signal
Timer Clock
Reset
POR
OUTMODx

Modes of Operation: Up Mode

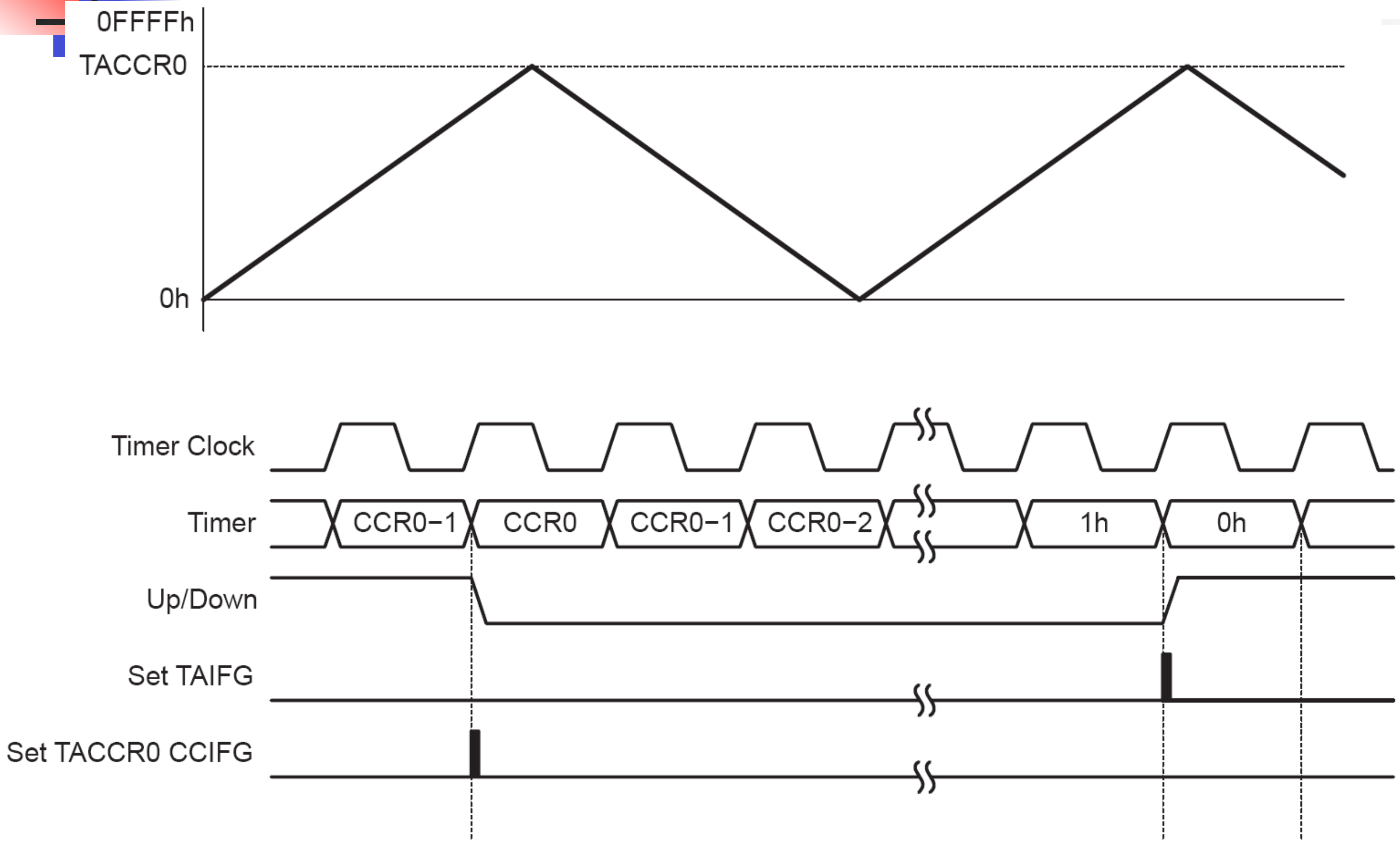


Modes of Operation: Continuous Mode



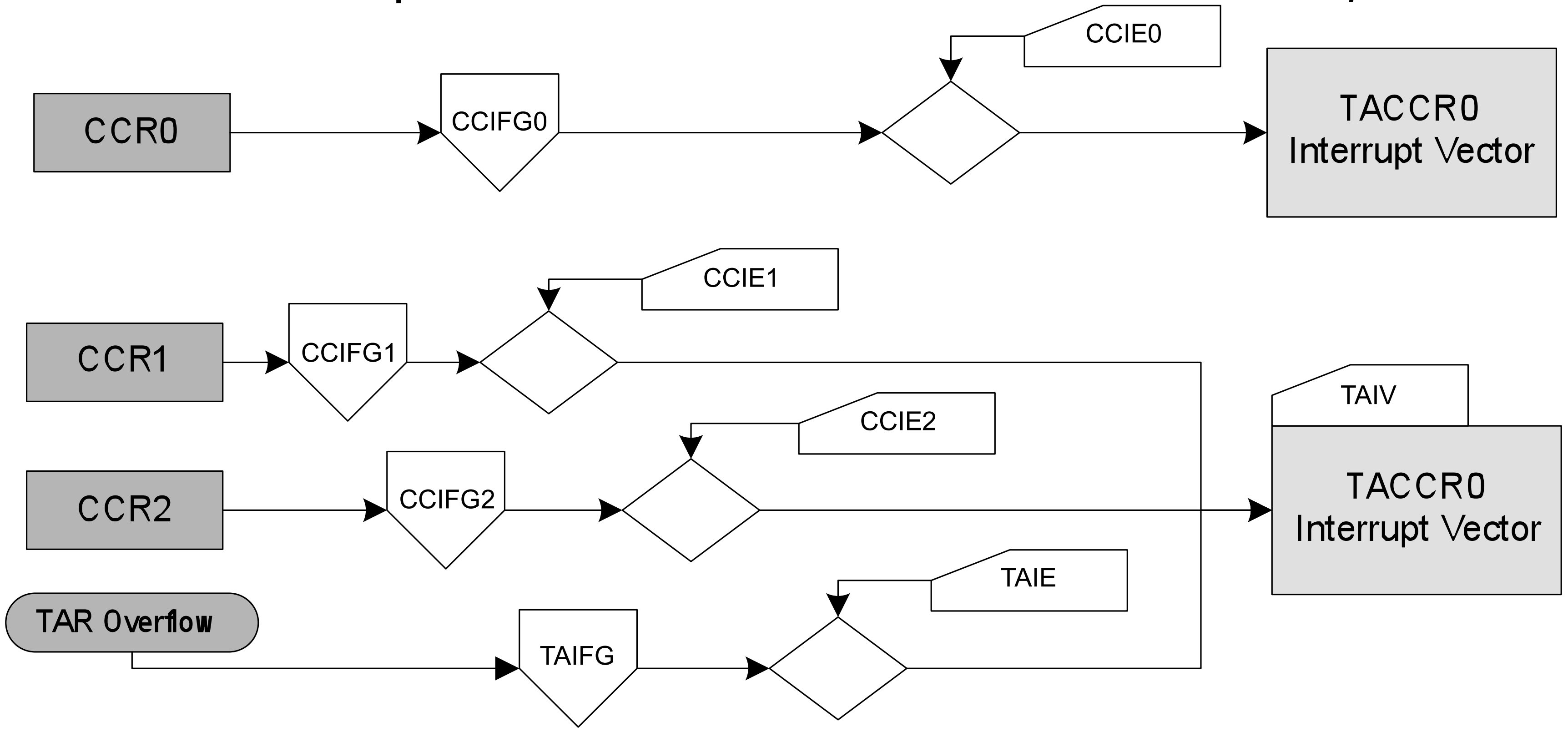


Modes of Operation: Up/Down Mode



Timer_A Interrupt Vectors

- TACCR0 interrupt vector for CCIFG of CCR0
- TAIV interrupt vector for TAIFG and CCIFGs of CCR1,CCR2



Timer_A Registers

- **TACTL, Timer_A Control Register (PART 1)**

15	14	13	12	11	10	9	8
Unused						**TASSELx**	
rw−(0)	rw−(0)	rw−(0)	rw−(0)	rw−(0)	rw−(0)	rw−(0)	rw−(0)

Unused	Bits 15-10	Unused
TASSELx	Bits 9-8	Timer_A clock source select 00 TACLK 01 ACLK 10 SMCLK 11 INCLK

TACTL, Timer A Control Register (PART

7	6	5	4	3	2	1	0
IDx		**MCx**		**Unused**	**TACLR**	**TAIE**	**TAIFG**
rw−(0)	rw−(0)	rw−(0)	rw−(0)	rw−(0)	w−(0)	rw−(0)	rw−(0)

IDx	Bits 7-6	Input divider. These bits select the divider for the input clock. 00 /1 01 /2 10 /4 11 /8
MCx	Bits 5-4	Mode control. Setting MCx = 00h when Timer_A is not in use conserves power. 00 Stop mode: the timer is halted 01 Up mode: the timer counts up to TACCR0 10 Continuous mode: the timer counts up to 0FFFFh 11 Up/down mode: the timer counts up to TACCR0 then down to 0000h
Unused	Bit 3	Unused
TACLR	Bit 2	Timer_A clear. Setting this bit resets TAR, the TACLK divider, and the count direction. The TACLR bit is automatically reset and is always read as zero.
TAIE	Bit 1	Timer_A interrupt enable. This bit enables the TAIFG interrupt request. 0 Interrupt disabled 1 Interrupt enabled
TAIFG	Bit 0	Timer_A interrupt flag 0 No interrupt pending 1 Interrupt pending

TACCTLx, Capture/Compare Control Register

15	14	13	12	11	10	9	8
CMx		CCISx		SCS	SCCI	Unused	CAP
rw−(0)	rw−(0)	rw−(0)	rw−(0)	rw−(0)	r−(0)	r−(0)	rw−(0)

7	6	5	4	3	2	1	0
OUTMODx			CCIE	CCI	OUT	COV	CCIFG
rw−(0)	rw−(0)	rw−(0)	rw−(0)	r	rw−(0)	rw−(0)	rw−(0)

CAP Bit 8 Capture mode
0 Compare mode
1 Capture mode

CCIE Bit 4 Capture/compare interrupt enable. This bit enables the interrupt request of the corresponding CCIFG flag.
0 Interrupt disabled
1 Interrupt enabled

CCIFG Bit 0 Capture/compare interrupt flag
0 No interrupt pending
1 Interrupt pending

TAIV, Timer_A Interrupt Vector Register

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
r0	r0	r0	r0	r0	r0	r0	r0

7	6	5	4	3	2	1	0
0	0	0	0		TAIVx		0
r0	r0	r0	r0	r−(0)	r−(0)	r−(0)	r0

TAIV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
00h	No interrupt pending	−	
02h	Capture/compare 1	TACCR1 CCIFG	Highest
04h	Capture/compare 2	TACCR2 CCIFG	
06h	Reserved	−	
08h	Reserved	−	
0Ah	Timer overflow	TAIFG	
0Ch	Reserved	−	
0Eh	Reserved	−	Lowest

Example 1

Continuous Mode

Output pin P5.4 (Red LED) with toggle rate = 32768/(32768) = 1 Hz

```
#include "include/include.h"
#include "include/hardware.h"

int main ( void )
{
      WDTCTL = WDTPW + WDTHOLD;          // Stop WDT
      P5DIR |= 0x10;                     // P5.4 output
      CCTL0 = CCIE;                      // CCR0 interrupt enabled
      CCR0 = 32768;
      TACTL = TASSEL_1 + MC_2;           // ACLK, continuous mode
      eint();                            // Enable the global interrupt }
      LPM0;                              // Enter low power mode      }
}

// Timer_A TACCR0 interrupt vector handler
interrupt (TIMERA0_VECTOR) TimerA_procedure( void ){
      P5OUT ^= 0x10;                     // Toggle P5.4
      CCR0 += 32768;                     // Add offset to CCR0
}
```

OR

```
_BIS_SR(LPM0_bits + GIE);
```

Example 2

Up Mode

Output pin P5.4 (Red LED) with toggle rate = 32768/(32768) = 1 Hz

```
#include "include/include.h"
#include "include/hardware.h"

int main ( void )
{
    WDTCTL = WDTPW + WDTHOLD;        // Stop WDT
    P5DIR |= 0x10;                   // P5.4 output
    CCTL0 = CCIE;                    // CCR0 interrupt enabled
    CCR0 = 32767;
    TACTL = TASSEL_1 + MC_1;         // ACLK, upmode
    _BIS_SR(LPM0_bits + GIE); // Enable the global interrupt and enter LPM0
}

// Timer_A TACCR0 interrupt vector handler
interrupt (TIMERA0_VECTOR) TimerA_procedure ( void ){
    P5OUT ^= 0x10;                   // Toggle P5.4
}
```

Example 3

Continuous Mode

Output pin P5.4 with toggle rate = 32768/(16384) = 0.5 Hz

Output pin P5.5 with toggle rate = 32768/(32768) = 1 Hz

Output pin P5.6 with toggle rate = 32768/(65536) = 0.5 Hz

```
#include "include/include.h"
#include "include/hardware.h"
int main ( void )
{
    WDTCTL = WDTPW + WDTHOLD;           // Stop WDT
    P5DIR |= 0x70;                      // P5.4, P5.5, P5.6 in output mode
    CCTL0 = CCIE;                       // CCR0 interrupt enabled
    CCTL1 = CCIE;                       // CCR1 interrupt enabled
    CCTL2 = CCIE;                       // CCR2 interrupt enabled
    CCR0 = 0;
    CCR1 = 0;
    CCR2 = 0;
    TACTL = TASSEL_1 + MC_2+ TAIE;      // ACLK, contmode, TAIE enabled
    _BIS_SR(LPM0_bits + GIE);           // Enable the global interrupt and enter LPM0
}
```

Example 3, continued

```
// Timer_A TACCR0 interrupt vector handler
interrupt (TIMERA0_VECTOR) TimerA0_procedure ( void ){
    P5OUT ^= 0x10;                          //on TACCR0  Toggle P5.4 (Red LED)
    CCR0 += 16384;                          // Add offset to CCR0
}

// Timer_A TAIV interrupt vector handler
interrupt (TIMERA1_VECTOR) TimerA1_procedure ( void ){
    switch( TAIV )
    {
         case  2: P5OUT ^= 0x20;            // on TACCR1 CCIFG Toggle P5.5 (Green LED)
                  CCR1 += 32768;            // Add offset to CCR1
                  break;

         case  10: P5OUT ^= 0x40;           // on Timer overflow TAIFG Toggle P5.6 (Blue LED)
                  break;
    }
}
```