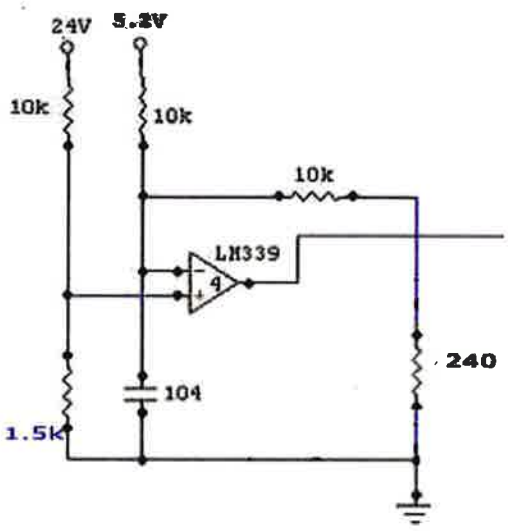
WEEK 7

Problem D, Solution



This is a low voltage detect circuit.

As long as v+>v- the output of the comparator will be high (an open circuit). As soon as v+<v-, the output goes low (short circuit).

The 5.2 volt source is regulated, so

$$v_{-} = 5.2\frac{10240}{20240} = 2.63.$$

Nominally (if the battery is at 24 Volts)

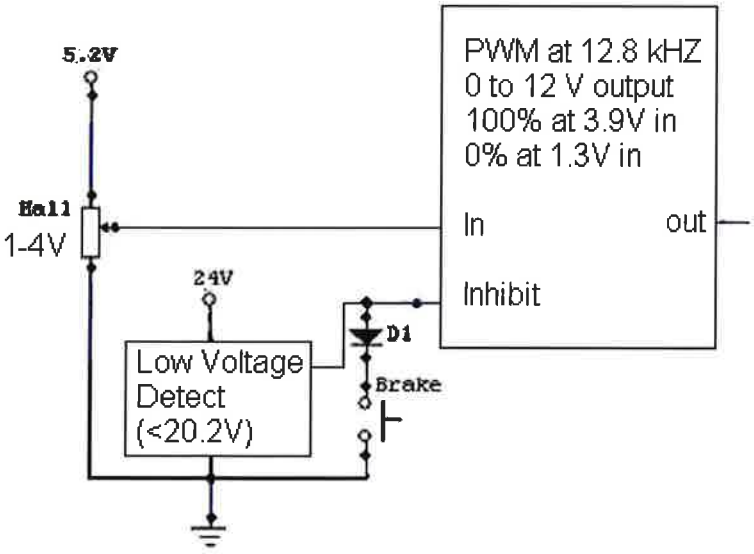
$$v_{+} = V_{batt}\frac{1500}{11500} = 3.13$$

The output of the comparator goes low when

$$v_{+} = V_{batt}\frac{1500}{11500} < v_{-} = 2.63$$

$$V_{batt} < 20.2$$

We can replace this part of the circuit with a Low Voltage Detect block that feeds into the inhibit of the PWM



Problem E, Solution

a) As the throttle voltage↑, duty cycle↑, % of time transistor is on and current flows↑, speed↑.

b) Since the motor is inductive, it clamps the voltage at the top (drain) of the transistors to 1 diode drop above the supply rail (24V) when the transistor shuts off (dI/dt is large, so there is a large voltage across the inductor (i.e., motor)).

c) The voltage across the resistor is $V_C=V_{dd}-V_R$. Power dissipated in the resistor is current times voltage.

$$P_r(t) = (i)\cdot V_R = \left(C\frac{dV_C}{dt}\right)\cdot V_R$$

$$= C\frac{d(V_{dd}-V_R)}{dt}\cdot V_R \quad \{V_{dd} \text{ is constant}\}$$

$$= -C\frac{dV_R}{dt}\cdot V_R$$

$$\text{Energy} = W = \int_0^\infty P_r(t)dt = \int_0^\infty \left(-C\frac{dV_R}{dt}\cdot V_R\right)dt$$

$$W = -\int_0^\infty C\frac{dV_R}{\cancel{dt}}\cdot V_R \cancel{dt} \quad \{\text{change variables}\}$$

$$= -\int_{V_{dd}}^0 CV_R dV_R = \int_0^{V_{dd}} CV_R dV_R$$

$$= \frac{1}{2}C\cdot V_{dd}^2$$

Note that the above shows that the energy lost charging a capacitor is not only independent of the resistor, but is true for any device (the current voltage relationship for a resistor was never used), even a non-linear one.

d) Power dissipated is split between resistor (33 Ohms) and the drive circuitry (unity gain buffer). This makes it easier to design the buffer. It also reduces the max current.

Clock Module

VLOCLK: typical 12kHz, min=40kHz, max=20kHz

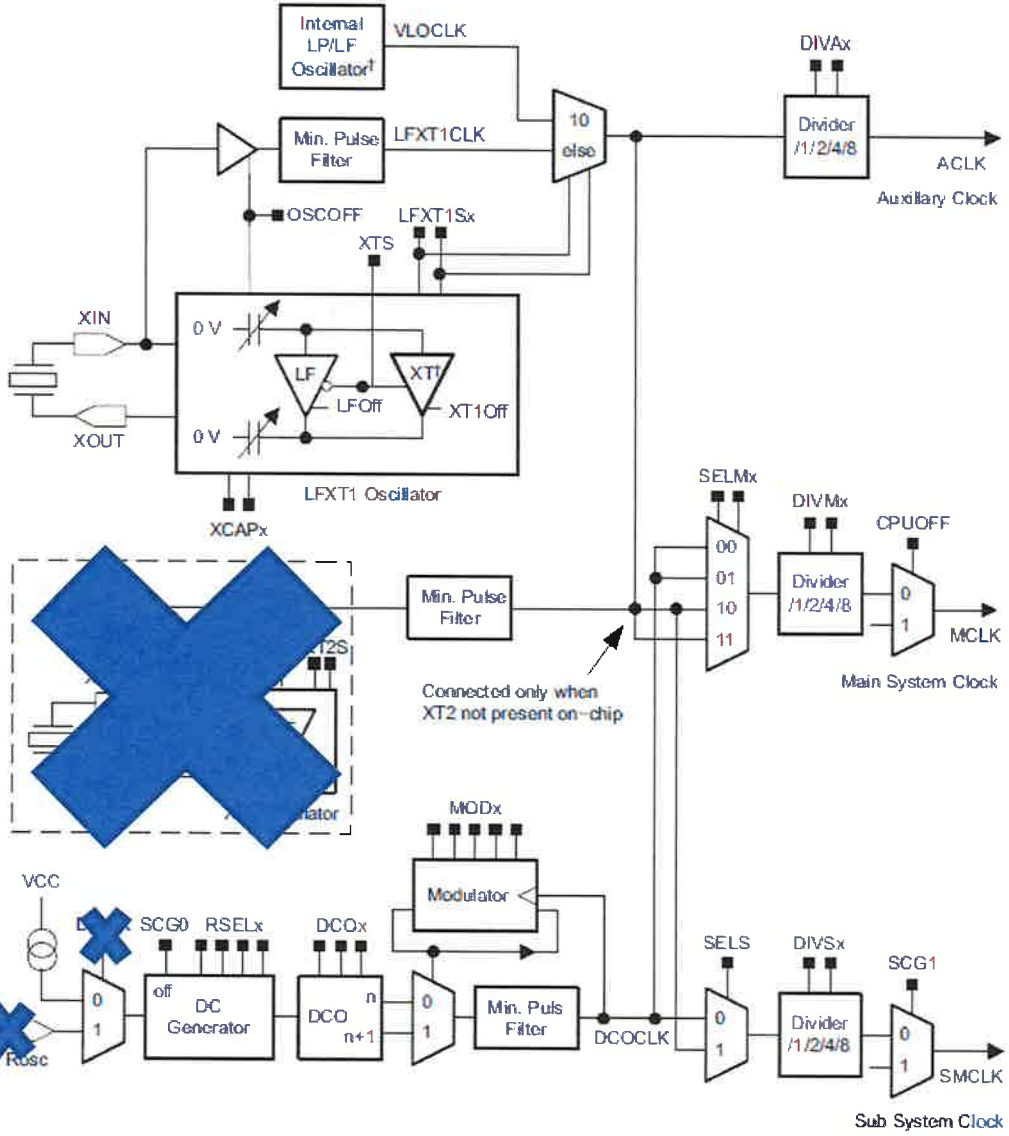
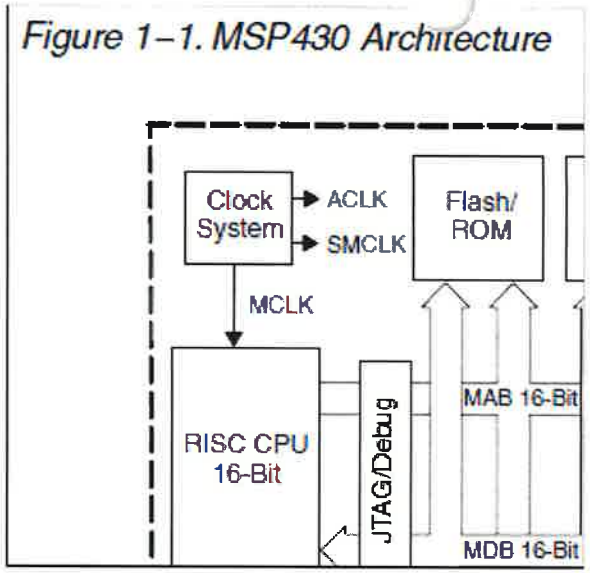


Figure 5-1. Basic Clock Module+ Block Diagram – MSP430F2xx



NOTE: † **Device-Specific Clock Variations**

Not all clock features are available on all MSP430x2xx devices:
MSP430G22x0: LFXT1 is not present, XT2 is not present, ROSC is not supported.

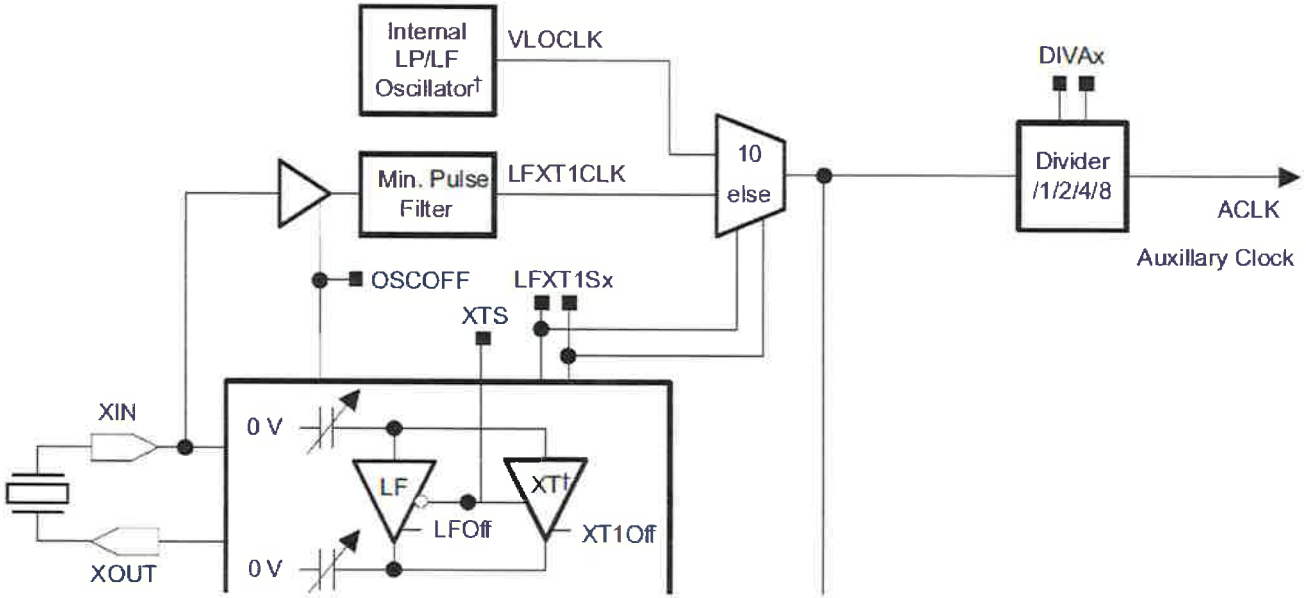
MSP430F20xx, MSP430G2xx1, MSP430G2xx2, MSP430G2xx3: LFXT1 does not support HF mode, XT2 is not present, ROSC is not supported.
MSP430x21x1: Internal LP/LF oscillator is not present, XT2 is not present, ROSC is not supported.
MSP430x21x2: XT2 is not present.
MSP430F22xx, MSP430x23x0: XT2 is not present.

What do control bits do?
Where do control bits reside?

The Fix

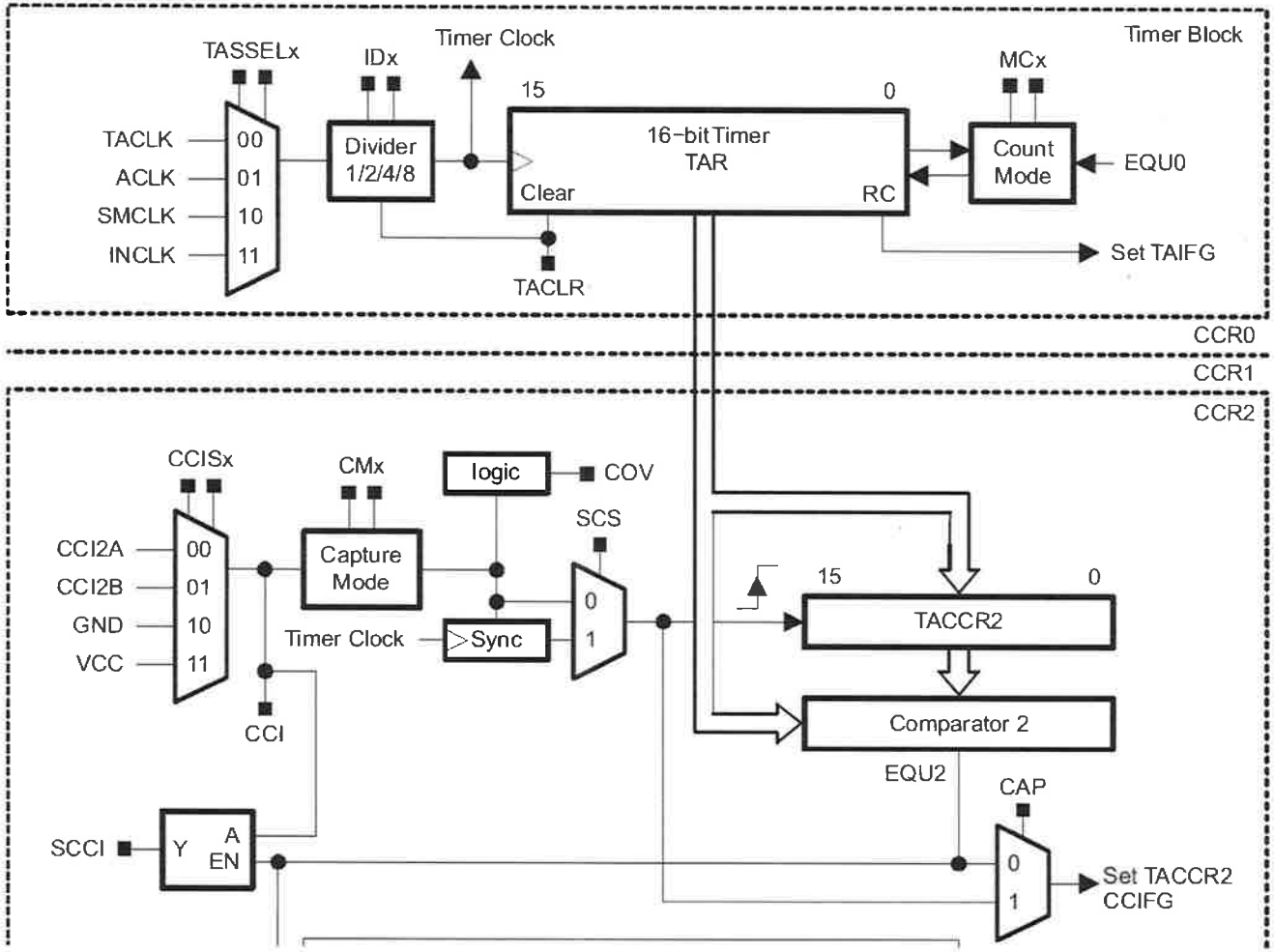
5.3.4 BCSCTL3, Basic Clock System Control Register 3

7	6	5	4	3	2	1	0
XT2Sx		LFXT1Sx(1)		XCAPx(2)		XT2OF(3)	LFXT1OF(2)
rw-0	rw-0	rw-0	rw-0	rw-0	rw-1	r0	r-(1)




```
void main(void){
    WDTCTL = WDTPW + WDTHOLD;          // Kill Watchdog
    P1DIR |= 0x01;                     // P1.0 output
    TACTL |= TASSEL_1 + ID_0 + MC_1;   // Select ACLK, select up mode, select divider to be 1
    TACCR0 = 6000;                     // Count up to 12000
    CCTL0 = CCIE;                      // CCR0 interrupt enabled
    BCSCTL3 |= BIT5;                   // Set ACLK to be VLO
    _BIS_SR(LPM3_bits + GIE);          // Go into lpm3 and enable interrupts
}
```

Timer A module




```
TACTL |= TASSEL_1 + ID_0 + MC_1;    // Select ACLK, select up mode, select divider to be 1
TACCR0 = 6000;                      // Count up to 6000
CCTL0 = CCIE;                       // CCR0 interrupt enabled
```

Note: Naming convention for registers can be confusing: CCR0=TACCR0=TA0CCR0, also TA1CCR0

```c
#include <msp430.h>

#define P1LED BIT0
#define P1SW  BIT3
/* This program sets up P1.0 as an output (connected
 * to an LED) and uses TimerA interrupts with 12kHz
 * ACLK as input.
 */
volatile int pbFlag=0;
#define LED BIT0

void main(void){
    WDTCTL = WDTPW + WDTHOLD;     // Kill Watchdog
    P1DIR |= LED;                 // P1.0 output
    TACTL |= TASSEL_1 + ID_0 + MC_1;   // Select ACLK, select up mode, select divider to be 1
    TACCR0 = 6000;                // Count up to 6000
    CCTL0 = CCIE;                 // CCR0 interrupt enabled
    BCSCTL3 |= BIT5;              // Set ACLK to be VLO
    _BIS_SR(LPM3_bits + GIE);     // Go into lpm3 and enable interrupts

}

// Timer A0 interrupt service routine
#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer_A (void)
{
    P1OUT ^= LED;              // Toggle P1.0
}


```

```c
#include <msp430.h>

#define P1LED BIT0
#define P1SW  BIT3
/* This program sets up P1.0 as an output (connected
 * to an LED) and uses TimerA interrupts with 12kHz
 * ACLK as input.
 */
volatile int pbFlag=0;
#define LED BIT0

void main(void){
    WDTCTL = WDTPW + WDTHOLD;     // Kill Watchdog

    //***********Start code, set to 1 MHz (from examples)*
    DCOCTL = 0x00;
    BCSCTL1 = CALBC1_1MHZ; /* Set DCO to 1MHz */
    DCOCTL = CALDCO_1MHZ;
    /* Basic Clock System Control 1
    * XT2OFF -- Disable XT2CLK
    * ~XTS -- Low Frequency
    * DIVA_0 -- Divide by 1 */
    BCSCTL1 |= XT2OFF + DIVA_0;
    /* Basic Clock System Control 3
    * XT2S_0 -- 0.4 - 1 MHz
    * LFXT1S_2 -- If XTS = 0, XT1 = VLOCLK ; If XTS = 1, XT1 = 3 - 16-MHz crystal or resonator
    * XCAP_1 -- ~6 pF */
    BCSCTL3 = XT2S_0 + LFXT1S_2 + XCAP_1;
    //***********End code, set to 1 MHz************

    BCSCTL3 |= BIT5;              // Set ACLK to be VLO
    BCSCTL2 |= BIT1|BIT2;         // SMCLK = MCLK/8 = 125000

    P1DIR |= LED;                 // P1.0 output
    TACTL |= TASSEL_2 + ID_2 + MC_1;   // Select SMCLK, select up mode, select divider to be 4 (31250)
    TACCR0 = 15625;                // Count up to 6000
    CCTL0 = CCIE;                 // CCR0 interrupt enabled
    _BIS_SR(LPM1_bits + GIE);     // Go into lpm3 and enable interrupts
}

// Timer A0 interrupt service routine
#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer_A (void)
{
    P1OUT ^= LED;              // Toggle P1.0
}


```

More DCO Registers

5.3.3 BCSCTL2, Basic Clock System Control Register 2

7	6	5	4	3	2	1	0
SELMx		DIVMx		SELS	DIVSx		DCOR(1)(2)
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 or VLOCLK when XT2 oscillator not present on-chip.
- 11 LFXT1CLK or VLOCLK

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. LFXT1CLK or VLOCLK when XT2 oscillator not present

DIVSx Bits 2-1 Divider for SMCLK
- 00 /1
- 01 /2
- 10 /4
- 11 /8

DCOR Bit 0 DCO resistor select. Not available in all devices. See the device-specific data sheet.
- 0 Internal resistor
- 1 External resistor

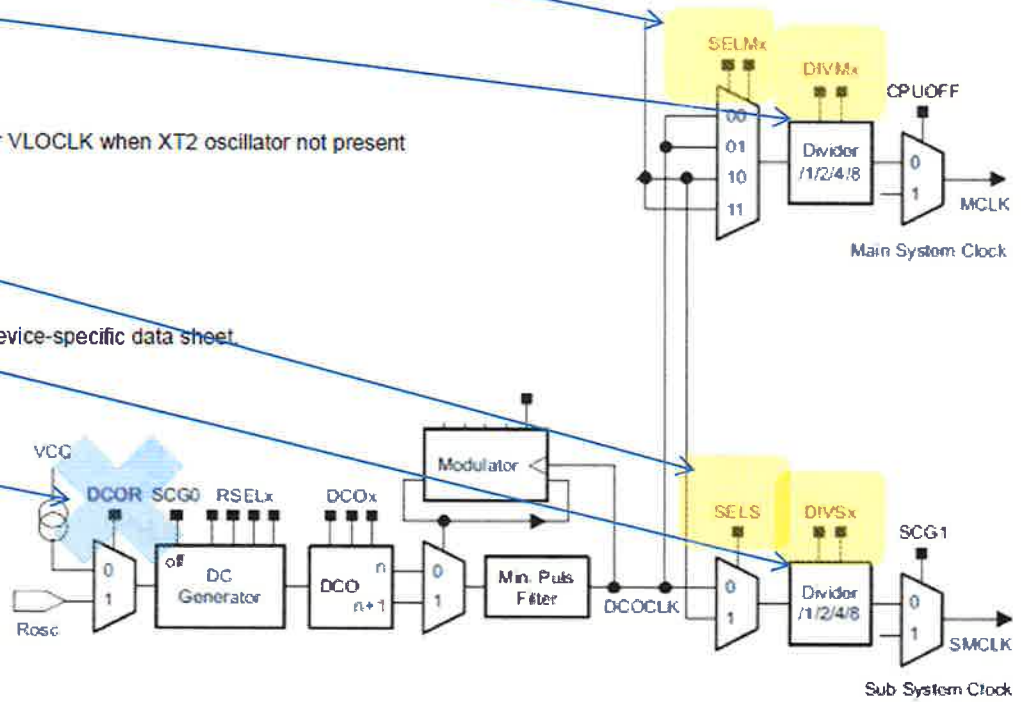


Figure 5-1. Basic Clock Module+ Block Diagram – MSP430F2xx

Yet again...

5.3.4 BCSCTL3, Basic Clock System Control Register 3

7	6	5	4	3	2	1	0
XT2Sx		LFXT1Sx[1]		XCAPx[2]		XT2OF[3]	LFXT1OF[2]
rw-0	rw-0	rw-0	rw-0	rw-0	rw-1	r0	r-(1)

XT2Sx Bits 7-6 XT2 range select. These bits select the frequency range for XT2.

00 0.4- to 1-MHz crystal or resonator

01 1- to 3-MHz crystal or resonator

10 3- to 16-MHz crystal or resonator

11 Digital external 0.4- to 16-MHz clock source

LFXT1Sx Bits 5-4 Low-frequency clock select and LFXT1 range select. These bits select between LFXT1 and VLO when XTS = 0, and select the frequency range for LFXT1 when XTS = 1.

When XTS = 0:

00 32768-Hz crystal on LFXT1

01 Reserved

10 VLOCLK (Reserved in MSP430F21x1 devices)

11 Digital external clock source

When XTS = 1 (Not applicable for MSP430x20xx devices, MSP430G2xx1/2/3)

00 0.4- to 1-MHz crystal or resonator

01 1- to 3-MHz crystal or resonator

10 3- to 16-MHz crystal or resonator

11 Digital external 0.4- to 16-MHz clock source

LFXT1Sx definition for MSP430AFE2xx devices:

00 Reserved

01 Reserved

10 VLOCLK

11 Reserved

XCAPx Bits 3-2 Oscillator capacitor selection. These bits select the effective capacitance seen by the LFXT1 crystal when XTS = 0. If XTS = 1 or if LFXT1Sx = 11 XCAPx should be 00.

00 ~1 pF

01 ~6 pF

10 ~10 pF

11 ~12.5 pF

XT2OF Bit 1 XT2 oscillator fault

0 No fault condition present

1 Fault condition present

LFXT1OF Bit 0 LFXT1 oscillator fault

0 No fault condition present

1 Fault condition present