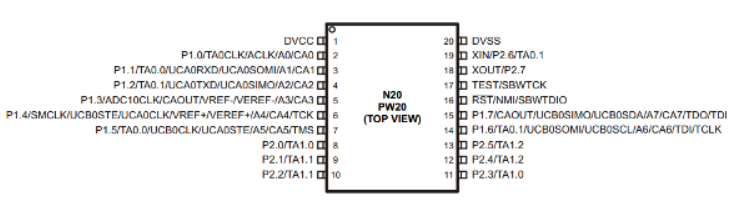
MSP430

Pinout:



PUC Power Up Clear

POR Power On Reset

Timer A

// Timer\_A is a 16-bit timer/counter with three capture/compare registers

//<https://www.embeddedrelated.com/showarticle/179.php>

<https://www.embeddedrelated.com/showcode/314.php>

3 timers in MSP430

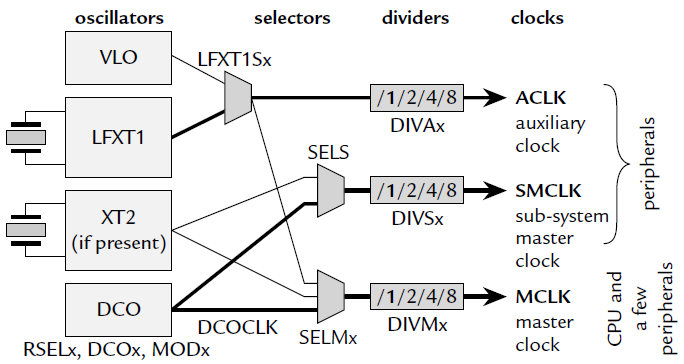
1. TimerA0,
2. TimerA1,
3. Watchdog Timer

There are 3 clocks in MSP430

1. MCLK : Master Clock. Used by CPU and some peripherals. Fast. Driven by DCO (Digitally Controlled Oscillator)
2. SMCLK: Sub-Master Clock. Used by Peripherals. Fast. Normally Driven by DCO.
3. ACLK: Auxiliary Clock. Used by peripherals. Slow. Driven by a low frequency Crystal Oscillator. Typically 32kHz.

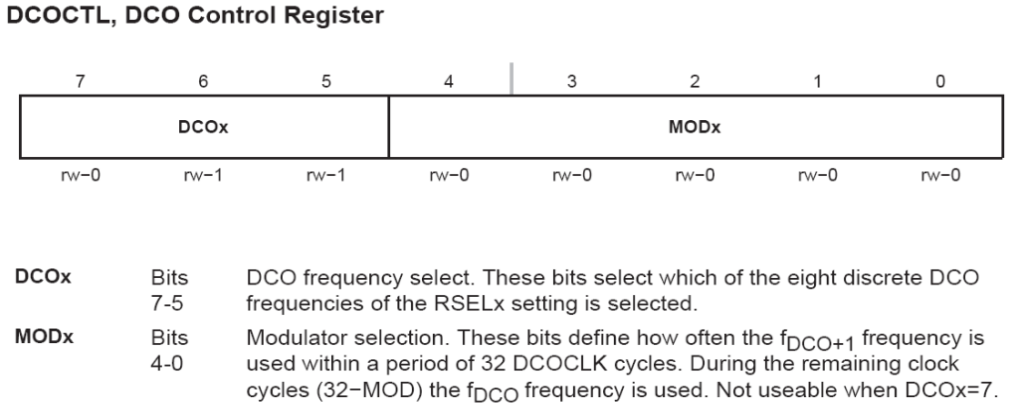
Clock Sources:

1. Low- or high-frequency crystal oscillator, LFXT1:
   1. External; used with a low- or high-frequency crystal; an external clock signal can also be used; connected to MSP430 through XIN and XOUT pins
2. High-frequency crystal oscillator, XT2:
   1. External; similar to LFXT1 but at high frequencies
3. Very low-power, low-frequency oscillator, VLO:
   1. Internal at 12 KHz; alternative to LFXT1 when accuracy of a crystal is not needed; may not available in all devices
4. Digitally controlled oscillator, DCO:
   1. Internal; a highly controllable RC oscillator that starts fast

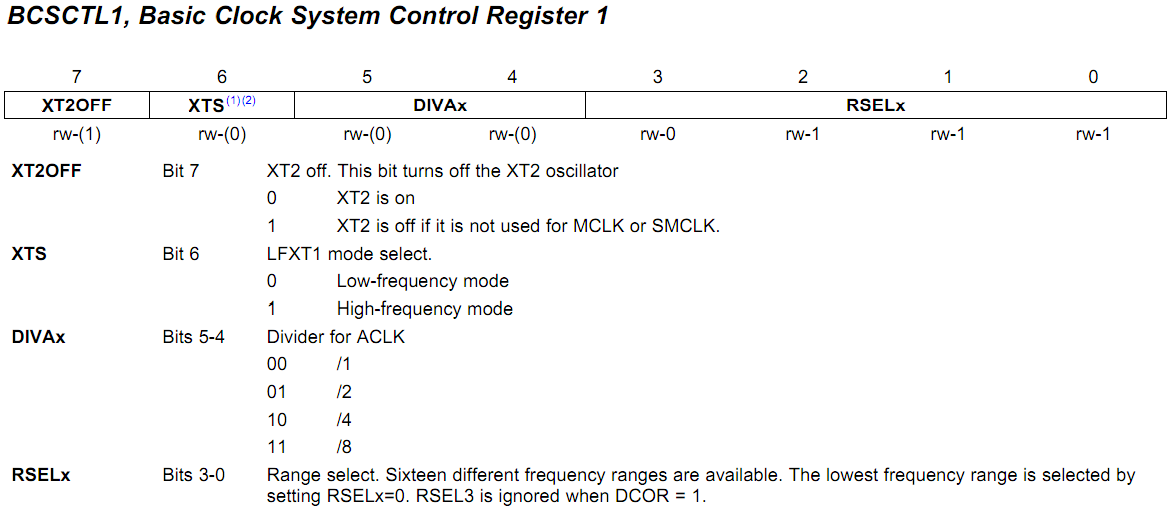


* In MSP430, the Basic Clock Module is also an IO peripheral
* Being an IO peripheral, it can be controlled by registers, DCOCTL and BCSCTL1–3
  + DCOCTL (056h): configure DCO
  + BCSCTL1 (basic clock system control 1, 057h): configure ACLK
  + BCSCTL2 (basic clock system control 2, 058h): configure MCLK, SMCLK
  + BCSCTL3 (basic clock system control 3, 053h): control LFXT1/VLO

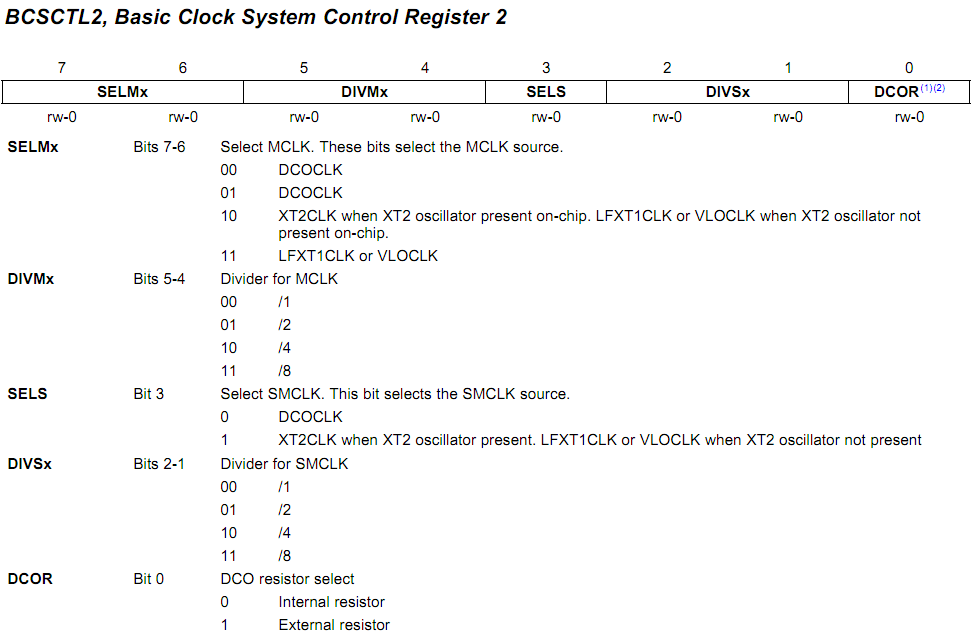
Configuring Clock:



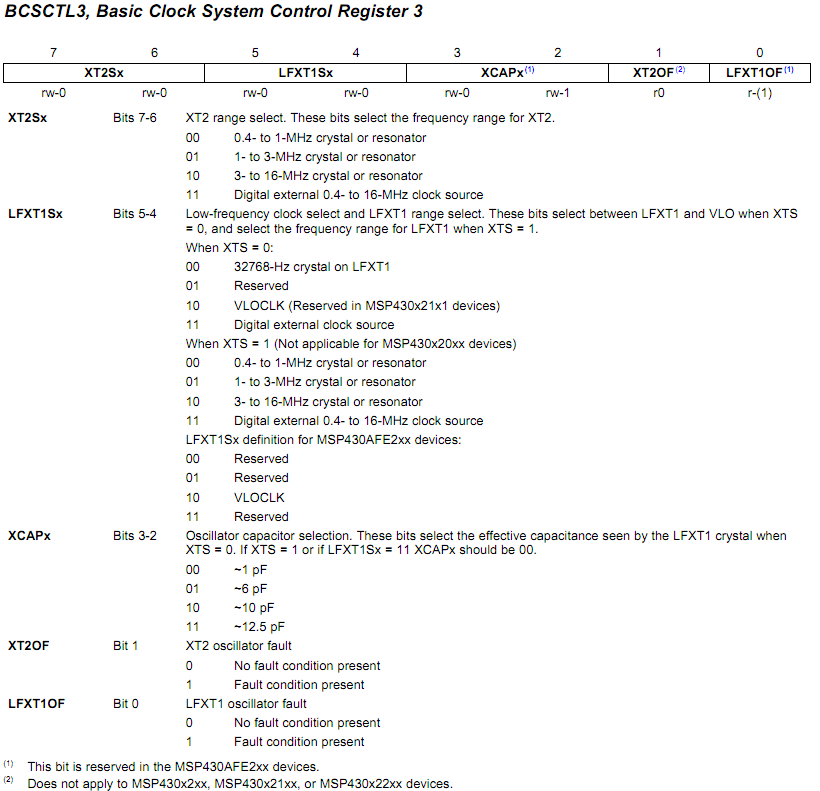
**DCOCTL = CALDCO\_1MHZ; // Set DCO step + modulation**



**BCSCTL1 = CALBC1\_1MHZ; // Set range**



**BCSCTL2 |= SELM\_3 + DIVM\_3; // MCLK = VLO/8**



**BCSCTL3 |= LFXT1S\_2; // Enable VLO as MCLK/ACLK src**

Steps for configuring a TimerA0:

1. Setup clock

DCOCTL = CALDCO\_1MHZ;

BCSCTL1 = CALBC1\_1MHZ;

1. Enable Interrupt for TimerA0 / TimerA1

TA0CCTL |= CCIE;

1. Configure TimerA0/ TimerA1

Select source in TA0CTL register: TACLK(TASSEL\_0)/ ACLK(TASSEL\_1) / SMCLK (TASSEL\_2)/ INCLK (TASSEL\_3)

1. Mode Select

Select mode in TA0CTL register: MC\_0 (stop timer) , MC\_1 (up to TACCR0), MC\_2 (Continuous mode), MC\_3 (up/down mode)

1. Use Input divider in TA0CTL register: ID\_0 (divider = 1) , ID\_1 (divider = 2), ID\_2 (divider = 4), ID\_3 (divider = 8)

NB: if you add TACLR, it will reset the divider to ID\_0.

1. Enable Global Interrupt

\_\_bis\_SR\_register (GIE)

1. Setup timerA interrupt service routine

**#pragma** vector = TIMER0\_A0\_VECTOR

**\_\_interrupt** **void** **Timer\_A**(**void**)

{

// things you wanna do

// no need to clear interrupt flag. It is automatically cleared

}

<http://learncontrollers.blogspot.com/>