

CPE/EE 421/521

## The MSP430 Clock Module

Dr. Rhonda Kay Gaede

**UAH**

**UAH**

**CPE/EE 421/521**

## The MSP430 Clock Module – Basic Clock Systems

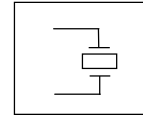
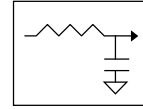
---

### **MSP430 Clock System**

- **Low System Cost**
- **Low Power**
  
- Variety of operating modes driven by \_\_\_\_\_, \_\_\_\_\_ selectable
- Support for the *Burst Mode* - when activated system starts and reacts rapidly
- Stability over \_\_\_\_\_ and \_\_\_\_\_

## The MSP430 Clock Module – MSP430x1xx

- **One DCO, digital controlled oscillator**  
Generated \_\_\_\_\_  
\_\_\_\_\_ frequency controlled by SW + HW
- **One LF/XT oscillator**  
LF: \_\_\_\_\_  
XT: \_\_\_\_\_
- **Second LF/XT2 oscillator**
- **Clocks:**  
ACLK \_\_\_\_\_ clock ACLK  
MCLK \_\_\_\_\_ clock MCLK  
SMCLK \_\_\_\_\_ system clock



Fall 2004 Page 3 of 44

## The MSP430 Clock Module – Clock Sources

- **LFXT1CLK:** Low-frequency/high-frequency oscillator that can be used either with low-frequency 32,768-Hz watch crystals, or standard crystals, resonators, or external clock sources in the 450-kHz to 8-MHz range.
- **XT2CLK:** Optional high-frequency oscillator that can be used with standard crystals, resonators, or external clock sources in the 450-kHz to 8-MHz range.
- **DCOCLK:** Internal digitally controlled oscillator (DCO) with RC-type characteristics.

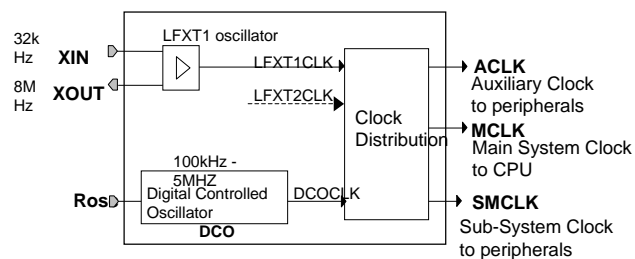
Fall 2004 Page 4 of 44

## The MSP430 Clock Module - Available Clocks

- **ACLK**: Auxiliary clock. The ACLK is the buffered LFXT1CLK clock source divided by 1, 2, 4, or 8. ACLK is software selectable for \_\_\_\_\_.
- **MCLK**: Master clock. MCLK is software selectable as LFXT1CLK, XT2CLK (if available), or DCOCLK. MCLK is divided by 1, 2, 4, or 8. MCLK is used by the \_\_\_\_\_.
- **SMCLK**: Sub-main clock. SMCLK is software selectable as LFXT1CLK, XT2CLK (if available), or DCOCLK. SMCLK is divided by 1, 2, 4, or 8. SMCLK is software selectable for individual peripheral modules.

Fall 2004 Page 5 of 44

## The MSP430 Clock Module - Basic Clock Systems: Block Diagram



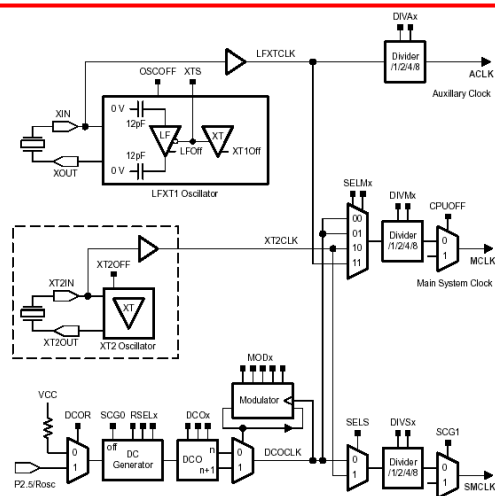
Fall 2004 Page 6 of 44

## The MSP430 Clock Module – Basic Clock Systems: Characteristics

- DCOCLK Generated on-chip with 6μs start-up
- 32.768 KHz Watch Crystal - or - High Speed Crystal / Resonator to 8MHz
  - (our system is 4MHz/8MHz high Speed Crystal)
- Flexible clock distribution tree for CPU and peripherals
- Programmable open-loop DCO Clock with internal and external current source

Fall 2004 Page 7 of 44

## The MSP430 Clock Module – Basic Clock Systems: Detail



Fall 2004 Page 8 of 44

## The MSP430 Clock Module – Basic Operation

---

- After PUC (\_\_\_\_\_) MCLK and SCLK are sourced by DCOCLK (approx. 800KHz) and ACLK is sourced by LFXT1 in LF mode
- Status register control bits **SCG0**, **SCG1**, **OSCOFF**, and **CPUOFF** configure the MSP430 \_\_\_\_\_ and enable or disable portions of the basic clock module. The **DCOCTL**, **BCSCTL1**, and **BCSCTL2** registers configure the \_\_\_\_\_
- The basic clock can be configured or reconfigured by software \_\_\_\_\_ during program execution

Fall 2004 Page 9 of 44

## The MSP430 Clock Module – Low-Power Operation: An Example

---

- \_\_\_\_\_ can be configured to oscillate with a low-power 32,786-Hz watch crystal
- \_\_\_\_\_ can be configured to operate from the on-chip DCO that can be only activated when requested by interrupt-driven events.
- \_\_\_\_\_ can be configured to operate from either the watch crystal or the DCO, depending on peripheral requirements.

Fall 2004 Page 10 of 44

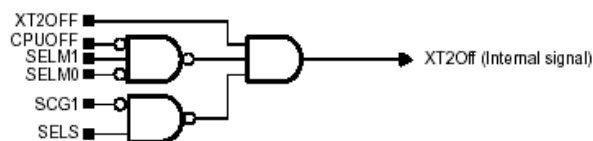
## The MSP430 Clock Module – LFXT1 Oscillator

- **LF mode:  $XTS = 0$** 
  - 32,768-Hz watch crystal in LF mode. A watch crystal connects to XIN and XOUT without any \_\_\_\_\_
- **HF mode:  $XTS = 1$** 
  - The high-speed crystal or resonator connects to XIN and XOUT and requires \_\_\_\_\_ on both terminals. These capacitors should be sized according to the crystal or resonator specifications.
- Software can \_\_\_\_\_ LFXT1 by setting **OSCOFF**, if this signal does not source SMCLK or MCLK

Fall 2004 Page 11 of 44

## The MSP430 Clock Module – XT2 Oscillator

- Similar to LFXT1 in HF mode
- **XT2OFF** bit disables the XT2 oscillator if XT2CLK is not used for MCLK or SMCLK



Fall 2004 Page 12 of 44

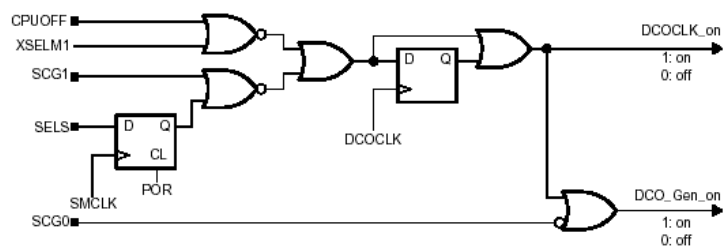
## The MSP430 Clock Module – Digitally-Controlled Oscillator DCO

- Integrated ring oscillator with RC-type characteristics.
  - frequency varies with \_\_\_\_\_, \_\_\_\_\_, and \_\_\_\_\_.
- DCO frequency can be adjusted by \_\_\_\_\_ using the **DCOx**, **MODx**, and **RSELx** bits.
- The digital control of the oscillator allows frequency stabilization despite its RC-type characteristics.

Fall 2004 Page 13 of 44

## The MSP430 Clock Module – Disabling DCO

- Software can disable DCO if not used for MCLK and SMCLK



Fall 2004 Page 14 of 44

## The MSP430 Clock Module – Adjusting Frequency

---

- After a PUC, an internal resistor is selected for the DC generator **RSEL<sub>x</sub>=4**, and **DCO<sub>x</sub>=3**, allowing the DCO to start at a \_\_\_\_\_ frequency.
- MCLK and SMCLK are sourced from DCOCLK. Because the CPU executes code from MCLK, which is sourced from the fast-starting DCO, code execution begins from PUC in less than 6 $\mu$ s.

## The MSP430 Clock Module – Adjusting Frequency

---

- DCO frequency is determined by:
  - The current injected into the DC generator by either the internal or external resistor. **DCOR** bit selects the internal or external resistor.
  - The three **RSEL<sub>x</sub>** bits select one of eight nominal frequency ranges for the DCO.
  - The three **DCO<sub>x</sub>** bits divide the DCO range into 8 frequency steps, separated by approx. 10%.
  - The five **MOD<sub>x</sub>** bits, switch between the frequency selected by the **DCO<sub>x</sub>** bits and the next higher frequency set by **DCO+1**.

## The MSP430 Clock Module – Basic Clock Control Registers: Overview

- Direct SW Control
- DCOCLK can be Set - Stabilized
- Stable DCOCLK over Temp/Vcc.

### BCSCTL1

057h

XT2Off	XTS	DIVA.1	DIVA.0	XT5V	Rsel.2	Rsel.1	Rsel.0
rw-(1)	rw-(0)	rw-(0)	rw-(0)	rw-0	rw-1	rw-0	rw-0

Selection of  
DCO nominal  
frequency

**RSEL.x** Select DCO nominal frequency

**DCO.x** and **MOD.x** set exact DCOCLK  
... select other clock tree options

### BCSCTL2

058h

SELM.1	SELM.0	DIVM.1	DIVM.0	SELS	DIVS.1	DIVS.0	DCOR
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

### DCOCTL

056h

DCO.2	DCO.1	DCO.0	MOD.4	MOD.3	MOD.2	MOD.1	MOD.0
rw-0	rw-1	rw-1	rw-0	rw-0	rw-0	rw-0	rw-0

Which of eight  
discrete DCO  
frequencies is  
selected

Define how often frequency  
 $f_{DCO+1}$  within the period of  
32 DCOCLK cycles is  
used. Remaining clock  
cycles (32-MOD) the  
frequency  $f_{DCO}$  is mixed

Fall 2004 Page 17 of 44

## The MSP430 Clock Module – Basic Clock Control Registers: Detail

The Basic Clock Module is configured using control registers DCOCTL, BCSCTL1, and BCSCTL2, and four bits from the CPU status register: SCG1, SCG0, OscOff, and CPUOFF.

User software can modify these control registers from their default condition at any time. The Basic Clock Module control registers are located in the \_\_\_\_\_ peripheral map and should be accessed with \_\_\_\_\_ instructions.

Register State	Short Form	Register Type	Address	Initial State
DCO control register	DCOCTL	Read/write	056h	060h
Basic clock system control 1	BCSCTL1	Read/write	057h	084h
Basic clock system control 2	BCSCTL2	Read/write	058h	reset

Fall 2004 Page 18 of 44

## The MSP430 Clock Module – Basic Clock Control Registers: Detail

- **Digitally-Controlled Oscillator (DCO) Clock-Frequency Control**

DCOCTL is loaded with a value of 060h with a valid PUC condition

	7							0
DCOCTL	DCO.2	DCO.1	DCO.0	MOD.4	MOD.3	MOD.2	MOD.1	MOD.0
056H	0	1	1	0	0	0	0	0

**MOD.0 .. MOD.4:** The MOD constant defines how often the discrete frequency  $f_{DCO+1}$  is used within a period of 32 DCOCLK cycles. During the remaining clock cycles (32-MOD) the discrete frequency  $f_{DCO}$  is used.

**DCO.0 .. DCO.2:** The DCO constant defines which one of the \_\_\_\_\_ discrete frequencies is selected. The frequency is defined by the current injected into the dc generator. When the DCO constant is set to \_\_\_\_\_, no modulation is possible since the \_\_\_\_\_ feasible frequency has then been selected.

## The MSP430 Clock Module – Basic Clock Control Registers: Detail

BCSCTL1 is affected by a valid PUC or POR condition.

	7							0
BCSCTL1	XT2Off	XTS	DIVA.1	DIVA.0	XT5V	Rsel.0	Rsel.1	Rsel.2
057h	1	0	0	0	0	1	0	0

**Rsel.0 to Rsel.2** 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 Rsel=0.

**XT5V:** XT5V should always be reset.

**DIVA.1 – DIVA.0:** The selected source for ACLK is divided by:

- DIVA = 0: 1
- DIVA = 1: 2
- DIVA = 2: 4
- DIVA = 3: 8



## The MSP430 Clock Module – Basic Clock Control Registers: Detail

### Bit3, SELS: Selects the source for generating SMCLK:

SELS = 0: Use the DCOCLK

SELS = 1: Use the XT2CLK signal (in three-oscillator systems)

or

LFXT1CLK signal (in two-oscillator systems)

### Bit4, Bit5: The selected source for MCLK is divided by DIVM.0 .. DIVM.1

DIVM = 0: 1, DIVM = 1: 2, DIVM = 2: 4, DIVM = 3: 8

### Bit6, Bit7: Selects the source for generating MCLK: SELM.0 .. SELM.1

SELM = 0: Use the DCOCLK

SELM = 1: Use the DCOCLK

SELM = 2: Use the XT2CLK (x13x and x14x devices)

or

Use the LFXT1CLK (x11x(1) devices)

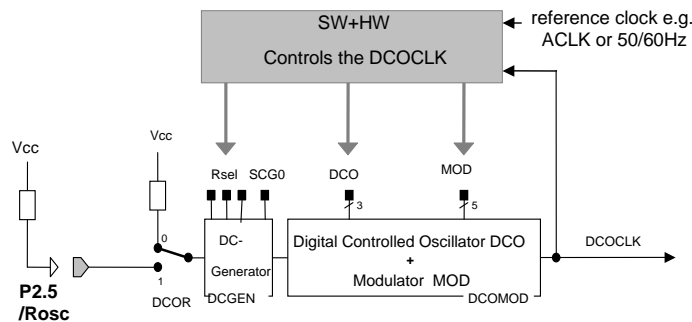
SELM = 3: Use the LFXT1CLK

Fall 2004 Page 23 of 44

## The MSP430 Clock Module – Basic Clock Systems: Software FLL Idea

### ➤ Basic Clock DCO is an open loop - close with SW+HW

- ❖ A reference frequency e.g. ACLK or 50/60Hz can be used to measure DCOCLK's
- ❖ **Initialization or Periodic** software set and stabilizes DCOCLK over reference clock
- ❖ DCOCLK is programmable 100kHz - 5Mhz and stable over voltage and temperature



Fall 2004 Page 24 of 44

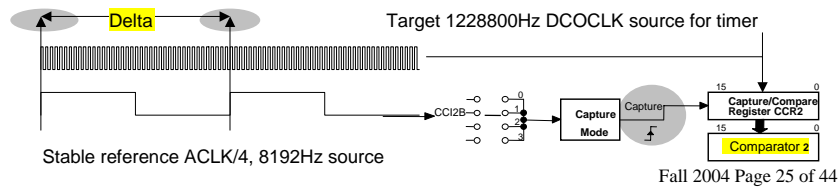
## The MSP430 Clock Module – Basic Clock Systems: Software FLL Implementation

➤ **Example: Set DCOCLK= 1228800, ACLK= 32768**

- ❖ ACLK/4 captured on CCI2B, DCOCLK is clock source for Timer\_A
- ❖ Comparator2 HW captures SMCLK (1228800Hz) in one ACLK/4 (8192Hz) period
- ❖ Target Delta = 1228800/8192= 150

```

CCI2BInt ...           ; Compute Delta
    cmp    #150,Delta   ; Delta= 1228800/8192
    jlo    IncDCO       ; JMP to IncDCO
DecDCO    dec    &DCOCTL ; Decrease DCOCLK
    reti
IncDCO    inc    &DCOCTL ; Increase DCOCLK
    reti
    
```

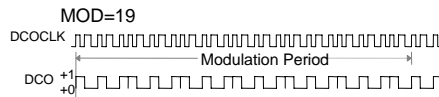


## The MSP430 Clock Module – Basic Clock Systems: DCO TAPS

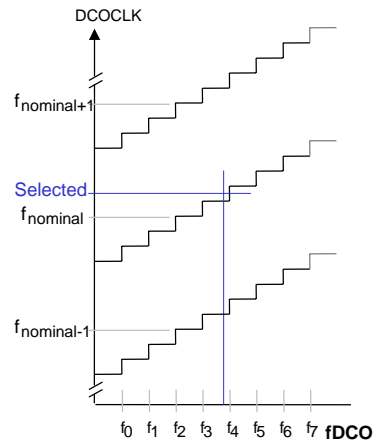
➤ **DCOCLK frequency control**

- ❖ nominal - injected current into DC generator
  - 1) internal resistors Rsel2, Rsel1 and Rsel0
  - 2) an external resistor at Rosc (P2.5/11x)
- ❖ Control bits DCO0 to DCO2 set  $f_{DCO}$  tap
- ❖ Modulation bits MOD0 to MOD4 allow mixing of  $f_{DCO}$  and  $f_{DCO+1}$  for precise frequency generation

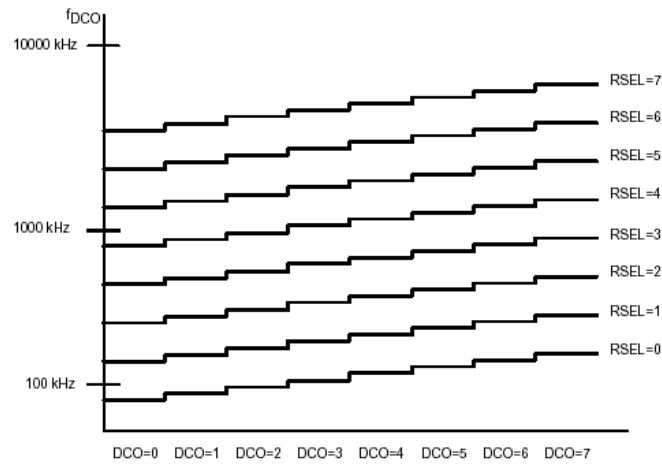
Example	Frequency	Cycle time
Selected:	1000kHz	1000 nsec
f3:	943kHz	1060 nsec
f4:	1042kHz	960 nsec



Cycle\_time = ((32-MOD)\*t1+MOD\*t2)/32 = 1000.625 ns,  
 selected frequency ≈ 1 MHz.



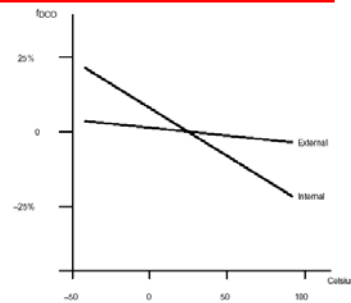
## The MSP430 Clock Module – Range (RSELx) and Steps (DCOx)



Fall 2004 Page 27 of 44

## The MSP430 Clock Module – External Resistor

- The DCO temperature coefficient can be reduced by using an external resistor ROSC to source the current for the DC generator.
- ROSC also allows the DCO to operate at higher frequencies.
  - Internal resistor nominal value is approximately 200 kOhm => DCO to operate up to 5 MHz.
  - External ROSC of approximately 100 kOhm => the DCO can operate up to approximately 10 MHz.



Fall 2004 Page 28 of 44

## The MSP430 Clock Module – DCO Modulator

---

- To produce an intermediate effective frequency between  $f_{\text{DCO}}$  and  $f_{\text{DCO}+1}$

The modulator mixing formula is:

$$t = (32 - \text{MODx}) \times t_{\text{DCO}} + \text{MODx} \times t_{\text{DCO}+1}$$

## The MSP430 Clock Module – Fail Safe Operation

---

- Basic module incorporates an \_\_\_\_\_ detection fail-safe feature.
- The oscillator fault detector is an analog circuit that monitors the LFXT1CLK (in HF mode) and the XT2CLK.
- An oscillator fault is detected when either clock signal is not present for approximately \_\_\_\_\_.
  - When an oscillator fault is detected, and when MCLK is sourced from either LFXT1 in HF mode or XT2, MCLK is automatically switched to the DCO for its clock source.

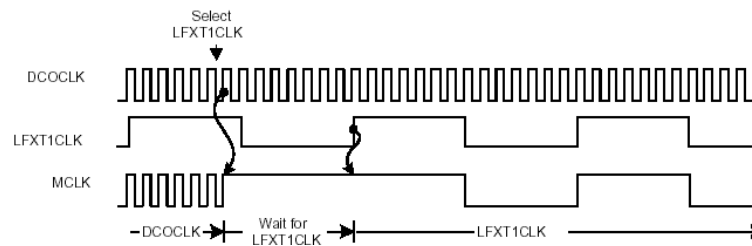
## The MSP430 Clock Module – Fail Safe Operation

- When OFIFG is set and OFIE is set, an NMI interrupt is requested. The NMI interrupt service routine can test the OFIFG flag to determine if an oscillator fault occurred. The OFIFG flag must be cleared by software.

Fall 2004 Page 31 of 44

## The MSP430 Clock Module – Synchronization of Clock Signals

- To avoid race conditions
  - The current clock cycle continues until the next rising edge.
  - The clock remains high until the next rising edge of the new clock.
  - The new clock source is selected and continues with a full high period.



Fall 2004 Page 32 of 44

## The MSP430 Clock Module – Basic Clock Systems: Examples

### ➤ How to select the Crystal Clock

```
void selectclock(void)
{
    IFG2=0;          /* reset interrupt flag register 1 */
    IFG1=0;          /* reset interrupt flag register 2 */
    BCSCCTL1|=XTS;  /*attach HF crystal (4MHz) to XIN/XOUT */
    do {
        /*wait in loop until crystal is stable*/
        IFG1&=~OFIFG;
    }while(OFIFG&IFG1);
    Delay();
    IFG1&=~OFIFG;    /*Reset osc. fault flag again*/
}

```

### ➤ How to select a clock for MCLK

```
BCSCCTL2=SELM0+SELM1; /*Then set MCLK same as LFXT1CLK*/
TACTL=TASSELO+TACLR+ID1; /*USE ACLK/4 AS TIMER_A INPUT CLOCK
                           (1MHz) */

```

## The MSP430 Clock Module – Adjusting the Basic Clock

### ➤ Adjusting the Basic Clock

The control registers of the Basic Clock are under full software control. If clock requirements other than those of the default from PUC are necessary, the Basic Clock can be configured or reconfigured by software at any time during program execution.

- ❑ ACLKGEN from LFXT1 crystal, resonator, or external-clock source and divided by 1, 2, 4, or 8. If no LFXTCLK clock signal is needed in the application, the OscOff bit should be set in the status register.
- ❑ SCLKGEN from LFXTCLK, DCOCLK, or XT2CLK (x13x and x14x only) and divided by 1, 2, 4, or 8. The SCG1 bit in the status register enables or disables SMCLK.
- ❑ MCLKGEN from LFXTCLK, DCOCLK, or XT2CLK (x13x and x14x only) and divided by 1, 2, 4, or 8. When set, the CPUOff bit in the status register enables or disables MCLK.
- ❑ DCOCLK frequency is adjusted using the RSEL, DCO, and MOD bits. The DCOCLK clock source is stopped when not used, and the dc generator can be disabled by the SCG0 bit in the status register (when set).
- ❑ The XT2 oscillator sources XT2CLK (x13x and x14x only) by clearing the XT2Off bit.

## The MSP430 Clock Module – Interrupt Service Routine Declaration

```
// Func. declaration
Interrupt[int_vector] void myISR (Void);

Interrupt[int_vector] void myISR (Void)
{
// ISR code
}
```

### EXAMPLE

```
Interrupt[TIMERA0_VECTOR] void myISR (Void);

Interrupt[TIMERA0_VECTOR] void myISR (Void)
{
// ISR code
}
```

Fall 2004 Page 35 of 44

## The MSP430 Clock Module – Interrupt Vectors

```
/*
 * Interrupt Vectors (offset from 0xFFE0)
 */
#define PORT2_VECTOR      1 * 2 /* 0xFFE2 Port 2 */
#define UART1TX_VECTOR   2 * 2 /* 0xFFE4 UART 1 Transmit */
#define UART1RX_VECTOR   3 * 2 /* 0xFFE6 UART 1 Receive */
#define PORT1_VECTOR     4 * 2 /* 0xFFE8 Port 1 */
#define TIMERA1_VECTOR   5 * 2 /* 0xFFEA Timer A CC1-2, TA */
#define TIMERA0_VECTOR   6 * 2 /* 0xFFEC Timer A CC0 */
#define ADC_VECTOR       7 * 2 /* 0xFFEE ADC */
#define UART0TX_VECTOR   8 * 2 /* 0xFFF0 UART 0 Transmit */
#define UART0RX_VECTOR   9 * 2 /* 0xFFF2 UART 0 Receive */
#define WDT_VECTOR       10 * 2 /* 0xFFF4 Watchdog Timer */
#define COMPARATORA_VECTOR 11 * 2 /* 0xFFF6 Comparator A */
#define TIMERB1_VECTOR   12 * 2 /* 0xFFF8 Timer B 1-7 */
#define TIMERB0_VECTOR   13 * 2 /* 0xFFFA Timer B 0 */
#define NMI_VECTOR       14 * 2 /* 0xFFFC Non-maskable */
#define RESET_VECTOR     15 * 2 /* 0xFFFE Reset [Highest Pr.] */
```

Fall 2004 Page 36 of 44

## The MSP430 Clock Module – Watchdog Timer: Overview

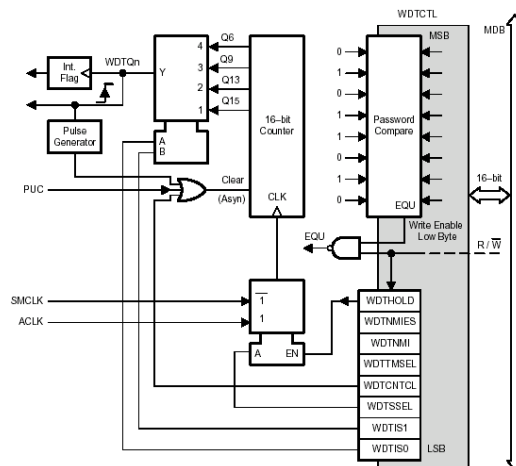
### Watchdog/Timer two functions:

- SW Watchdog Mode
- Interval Timer Mode

### Features of the Watchdog Timer include:

- Eight software-selectable time intervals
- Two operating modes: as watchdog or interval timer
- Expiration of the time interval in watchdog mode, which generates a system reset; or in timer mode, which generates an interrupt request
- Safeguards which ensure that writing to the WDT control register is only possible using a password
- Support of ultralow-power using the hold mode

## The MSP430 Clock Module – Watchdog Timer: Block Diagram

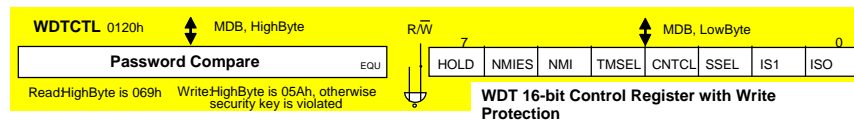


## The MSP430 Clock Module – Watchdog Timer: Registers

### ❑ Watchdog Timer Counter

The watchdog-timer counter (WDTCNT) is a 16-bit up-counter that is not directly accessible by software. The WDTCNT is controlled through the watchdog-timer control register (WDTCTL), which is a 16-bit read/write register located at the low byte of word address 0120h. Any read or write access must be done using word instructions with no suffix or .w suffix. In both operating modes (watchdog or timer), it is only possible to write to WDTCTL using the correct password.

### ❑ Watchdog Timer Control Register



Bits 0, 1: Bits ISO0 and IS1 select one of four taps from the WDTCNT, as described in following table. Assuming  $f_{\text{crystal}} = 32,768 \text{ Hz}$  and  $f_{\text{System}} = 1 \text{ MHz}$ , the following intervals are possible:

Fall 2004 Page 39 of 44

Electrical and Computer Engineering

## The MSP430 Clock Module – Watchdog Timer: Registers

SSEL	IS1	ISO	Interval [ms]
0	1	1	0.064 $t_{\text{SMCLK}} \times 2^6$
0	1	0	0.5 $t_{\text{SMCLK}} \times 2^9$
1	1	1	1.9 $t_{\text{ACLK}} \times 2^6$
0	0	1	8 $t_{\text{SMCLK}} \times 2^{13}$
1	1	0	16.0 $t_{\text{ACLK}} \times 2^9$
<b>0</b>	<b>0</b>	<b>0</b>	<b>32</b> $t_{\text{SMCLK}} \times 2^{15}$ ← Value after PUC (reset)
1	0	1	250 $t_{\text{ACLK}} \times 2^{13}$
1	0	0	1000 $t_{\text{ACLK}} \times 2^{15}$

**Bit 2:** The SSEL bit selects the clock source for WDTCNT. SSEL = 0: SMCLK, SSEL = 1: ACLK.

**Bit 3:** Counter clear bit. In both operating modes, writing a 1 to this bit restarts the WDTCNT at 00000h. The value read is not defined.

**Bit 4:** The TMSEL bit selects the operating mode: watchdog or timer. TMSEL = 0: Watchdog, TMSEL = 1: Interval-timer

Fall 2004 Page 40 of 44

Electrical and Computer Engineering

## The MSP430 Clock Module – Watchdog Timer: Registers

**Bit 5:** The NMI bit selects the function of the RST/NMI input pin. It is cleared by the PUC signal.

**NMI = 0:** The RST/NMI input works as reset input.

As long as the RST/NMI pin is held low, the internal signal is active (level sensitive).

**NMI = 1:** The RST/NMI input works as an edge-sensitive non-maskable interrupt input.

**Bit 6:** If the NMI function is selected, this bit selects the activating edge of the RST/NMI input. It is cleared by the PUC signal.

**NMIES = 0:** A rising edge triggers an NMI interrupt.

**NMIES = 1:** A falling edge triggers an NMI interrupt.

CAUTION: Changing the NMIES bit with software can generate an NMI interrupt.

**Bit 7:** This bit stops the operation of the watchdog counter. The clock multiplexer is disabled and the counter stops incrementing. It holds the last value until the hold bit is reset and the operation continues. It is cleared by the PUC signal.

**HOLD = 0:** The WDT is fully active.

**HOLD = 1:** The clock multiplexer and counter are stopped.

Fall 2004 Page 41 of 44

## The MSP430 Clock Module – Watchdog Timer: Interrupt Function

❑ **The Watchdog Timer (WDT) uses two bits in the SFRs for interrupt control.**

The WDT interrupt flag (WDTIFG) (located in IFG1.0, initial state is reset)

The WDT interrupt enable (WDTIE) (located in IE1.0, initial state is reset)

- ❖ When using the watchdog mode, the WDTIFG flag is used by the reset interrupt service routine to determine if the watchdog caused the device to reset. If the flag is set, then the Watchdog Timer initiated the reset condition (either by timing out or by a security key violation). If the flag is cleared, then the PUC was caused by a different source. See chapter 3 for more details on the PUC and POR signals.
- ❖ When using the Watchdog Timer in interval-timer mode, the WDTIFG flag is set after the selected time interval and a watchdog interval-timer interrupt is requested. The interrupt vector address in interval-timer mode is different from that in watchdog mode. In interval-timer mode, the WDTIFG flag is reset automatically when the interrupt is serviced.
- ❖ The WDTIE bit is used to enable or disable the interrupt from the Watchdog Timer when it is being used in interval-timer mode. Also, the GIE bit enables or disables the interrupt from the Watchdog Timer when it is being used in interval-timer mode.

Fall 2004 Page 42 of 44

## The MSP430 Clock Module - Watchdog Timer: Timer Mode

---

- ❖ Setting WDTCTL register bit TMSEL to 1 selects the timer mode. This mode provides periodic interrupts at the selected time interval. A time interval can also be initiated by writing a 1 to bit CNTCL in the WDTCTL register.
- ❖ When the WDT is configured to operate in timer mode, the WDTIFG flag is set after the selected time interval, and it requests a standard interrupt service. The WDT interrupt flag is a single-source interrupt flag and is automatically reset when it is serviced. The enable bit remains unchanged. In interval-timer mode, the WDT interrupt-enable bit and the GIE bit must be set to allow the WDT to request an interrupt. The interrupt vector address in timer mode is different from that in watchdog mode.

Fall 2004 Page 43 of 44

## The MSP430 Clock Module - Watchdog Timer: Examples

---

### ❑ How to select timer mode

```
/* WDT is clocked by fACLK (assumed 32Khz) */
WDTCL=WDT_ADLY_250; // WDT 250MS/4 INTERVAL TIMER
IE1 |=WDTIE; // ENABLE WDT INTERRUPT
```

### ❑ How to stop watchdog timer

```
WDTCTL=WDTPW + WDT_HOLD ; // stop watchdog timer
```

### ❑ Assembly programming

```
WDT_key      .equ      05A00h           ; Key to access WDT
WDTStop      mov       #(WDT_Key+80h),&WDTCTL ; Hold Watchdog
WDT250       mov       #(WDT_Key+1Dh),&WDTCTL ; WDT, 250ms Interval
```

Fall 2004 Page 44 of 44