# Chapter 5

# **Device Systems and Operating Modes**

This chapter describes the internal device systems of the MSP430. It includes a description of the internal system reset and clock sources.

The MSP430 supports interrupts, that is, events that occur sporadically and need to be processed immediately. The interrupt sources, both external and internal, cause the CPU to suspend execution of the current task, re-direct program flow to an interrupt service routine devoted to the specific event, then resume the former task. How the microcontroller responses to an interrupt may dictate its performance and how the software needs to be written.

This chapter covers all of the above points, extending them to the low power operating modes that are one of the main features of the MSP430. This chapter focuses the internal device systems supported by the MSP430 families that are used with the hardware development tools (2xx and 4xx families).

Торіс		Page
5.1 Sy	stem reset	5-3
5.2 Sy	stem clocks	5-4
5.2.1	Low/High frequency oscillator (LFXT1)	5-4
5.2.2	Second crystal oscillator (XT2)	5-5
5.2.3	Digitally-controlled oscillator (DCO)	5-5
5.2.4	Frequency Locked Loop (FLL)	5-8
5.2.5	Internal clock signals	5-11
5.3 In	terrupt management	5-12
5.3.1	Types of interrupts	5-13
5.4 W	atchdog timer (WDT and WDT+)	5-16
5.4.1	Watchdog Timer+ (WDT+)	5-18
5.5 Su	pervisory Voltage System (SVS)	5-18

5.6	Low power operating modes	. 5-20
5.6	0.1 Low power modes	. 5-20
5.7	Quiz	. 5-24
5.8	FAQs	. 5-26

#### 5.1 System reset

The MSP430 families make use of two independent reset signals:

- □ Hardware reset signal POR (Power On Reset);
- □ Software reset signal PUC (Power Up Clear).

Different events can generate each one of the reset signals. The following sources can generate a POR or a PUC:

- □ POR:
  - Initial device power up;
  - Low signal at the reset pin (RST/NMI), when this is configured in reset mode;
  - Low signal at the Supervisory Voltage System (SVS), when the register bit PORON is high. (This condition will be discussed in detail in section 5.5).
- □ PUC:
  - Active POR signal;
  - Expiry of watchdog timer, when it is configured in supervision mode (Further details in section 5.4);
  - Flash memory control registers access security key violation.

When the hardware reset signal (POR) is high, the Status Register is reset and the Program Counter is loaded with the address in program memory location 0FFFEh. Peripheral registers all enter their power-up state. When the reset signal is from software (PUC), the Status Register is reset, and the Program Counter is loaded with either the reset vector (0FFFEh), or the PUC source interrupt vector. Only some peripheral registers are reset by PUC. These conditions depend on the reset source and the specific MSP430 device.

All 2xx and 4xx MSP430 devices have a reset circuit to detect a power source disturbance, known as a Brown Out Reset (BOR). This circuitry is an elaborate POR system, which includes a hysteresis circuit to allow the device to stay in reset mode until the voltage is higher than the upper threshold ( $V_{B_{IT+}}$ ). When the voltage is higher than this value, the BOR takes 2 msec to become inactive and allow the program execution by CPU. Similarly, when the voltage decreases below the lower threshold ( $V_{B_{IT-}}$ ), either by power source interruption or battery discharge, the BOR circuit will generate a reset signal, which will remain active until the voltage rises above the lower threshold value.

## 5.2 System clocks

The MSP430 devices have a clock system that allows the CPU and the peripherals to operate from different clock sources. The system clocks depend on the particular device in the MSP430 family:

□ MSP430x2xx: The Basic Clock Module+ (BCM+) is composed of one or two oscillators (depending on the device) and is able to work with external crystals or resonators, in addition to the internal digitally controlled oscillator (DCO). It allows a working frequency up to 16 MHz, lower power consumption and lower internal oscillator start up time.

□ MSP430x4xx: The system clock is defined by the Frequency Locked Loop (FLL+). This system is composed of one or two oscillators (depending on the device), and is able to work with external crystals or resonators, as well as the internal Digitally Controlled Oscillator (DCO). The DCO is adjusted and controlled by hardware, providing multiple working frequencies from an external low frequency oscillator.

The clock sources from these oscillators can be selected to generate a range of different clock signals: Master clock (MCLK), Sub-system main clock (SMCLK) and auxiliary clock (ACLK). Each of these clock signals can be internally divided by 1, 2, 4 or 8, before being made available to the CPU and peripheral devices:

 $\Box$  MCLK: Can be generated by the DCO (but can also be fed by the crystal oscillator), which can be activated and reach stability in less than 6 µsec. It can be used by the CPU and high-speed peripherals;

□ SMCLK: Used as alternative clock source for peripherals;

□ ACLK: Background real-time clock with self wake-up function for low power modes (32.768 kHz watch crystal). It is always fed by the crystal oscillator.

#### 5.2.1 Low/High frequency oscillator (LFXT1)

The Low-frequency/high-frequency oscillator (LFXT1) is implemented in all MSP430 devices.

It can be used with low-frequency 32.768 kHz watch crystals, providing a Real Time Clock (RTC), or standard crystals, resonators, or external clock sources in the range 450 kHz to 8 MHz (16 MHz for the 2xx family). The operating mode selection is defined by a bit of a control register that is configured as a low signal (=0) to provide a low frequency clock, and otherwise (=1) to provide a high frequency clock:

XTS: located at the BCSCTL1 register (2xx families);

XTS\_FLL: located at the FLL\_CTL0 register (4xx families);

The main differences between the oscillator structure in the 2xx and 4xx families concern the ACLK nomenclature and the internal capacitors values. LFXT1 generates a clock source, that after being divided by 1, 2, 4 or 8, provides the ACLK clock signal (2xx families) or ACLK/n for 4xx family.

- □ In the 2xx and 4xx families:
  - The capacitor values are selectable (1, 6, 8 or 10 pF) by the XCAPxPF bits.

The LFXT1 oscillator can be powered down (bit OSCOFF=1) if it is not used to source MCLK and can use clock signal from an external oscillator (XIN pin).

#### **5.2.2** Second crystal oscillator (XT2)

Some devices in the 1xx, 2xx and 4xx families have a second crystal oscillator, XT2. XT2 sources XT2CLK and its characteristics are identical to those of LFXT1 in HF mode (450 kHz to 8 MHz, or 16 MHz for the 2xx family). For XT2, the required load capacitance for the high frequency crystal or resonator must be provided externally.

This oscillator can be disabled by the XT2OFF bit (BCSCTL1 register in 1xx and 2xx families, FLL\_CTL1 register in 4xx family), if XT2CLK is not used to source the MCLK and SMCLK clock signals.

#### 5.2.3 Digitally-controlled oscillator (DCO)

The DCO is an integrated ring oscillator with RC-type characteristics that is able to provide a wide, software-controllable frequency range. The DCO frequency is stabilized by the Frequency Locked Loop (FLL) for the 4xx devices.

Depending on device in the MSP430 family, the frequency modulation method is provided by FLL functionality or by using the following steps:

□ 2xx family: Does not have full FLL functionality. The DCO generates an internal signal identified as DCOCLK, which can be programmed either internally or externally via the DCOR bit state and controlled externally by means of a resistor connected to the  $R_{OSC}$  and  $V_{CC}$  pins. The DCO control bits are:

- RSELx: DCO's frequency range selection;
- DCOx: DCO's fundamental frequency, within the range defined by the RSEL bits. The step size is defined by the parameter S<sub>DCO</sub>;

• MODx: Modulation bits select how often  $f_{DCO(RSEL,DCO+1)}$  is used within the 32 DCOCLK cycles period. The frequency  $f_{DCO(RSEL,DCO)}$  is used for the remaining cycles.

Specific frequency ranges and values vary by device, and are described in the datasheet for the part you are using. In the case of the MSP430x20x3, from the datasheet, the average clock frequency (which depends also on voltage and temperature) can be calculated by:

$$f_{avg} = \frac{32 \times f_{DCO(RSEL,DCO)} \times f_{DCO(RSEL,DCO+1)}}{MOD \times f_{DCO(RSEL,DCO)} + (32 - MOD) \times f_{DCO(RSEL,DCO+1)}}$$

The DCOCTL, BCSCTL1, BCSCTL2, and BCSCTL3 registers that configure the Basic Clock Module+ (BCM+) are shown in the following tables. Their usage is given in some detail because the registers are used in the USB stick hardware development kits (eZ430-F2013 and eZ430-RF2500), and also in the experimenter's board, because one of the MSP430 devices belongs to 2xx family (F2013).

#### DCOCTL, DCO Control Register

7	6	5	4	3	2	1	0
DCOx					MODx		

Bit		Description
7-5	DCOx	Discrete DCO frequency selection step (depends on RSELx bits).
4-0	MODx	Modulator selection.

#### **BCSCTL1, BCS Control Register 1**

7	6	5	4	3	2	1	0	
XT2OFF	XTS	DI	DIVAx		RSELx			

Bit		Description		
7	XT2OFF	XT2 oscillator fault:		
		$XT2OFF = 0 \Rightarrow$	XT2 n	ormal operation
		$XT2OFF = 1 \Rightarrow$	XT2 fa	ault condition
6	XTS	LFXT1 oscillator operating	mode:	
		XTS = 0	$\Rightarrow$	LF mode (low frequency)
		XTS = 1	$\Rightarrow$	HF mode (high frequency)
5-4	DIVAx	ACLK frequency divider:		
		DIVA1 DIVA0 = 0 0	$\Rightarrow$	/1
		DIVA1 DIVA0 = 0 1	$\Rightarrow$	/2
		DIVA1 DIVA0 = 1 0	$\Rightarrow$	/4
		DIVA1 DIVA0 = 1 1	$\Rightarrow$	/8
3-0	RSELx	Range select. Sixteen diffe	rent fre	quency ranges are available.

## BCSCTL2, BCS Control Register 2

			-						
7	,	6	5	4	3		2	1	0
	SELMx		DIV	Мх	SELS	5	DIVSx		DCOR
Bit		De	escription			·			
7-6	SELMx	MC	CLK source: SELM1 SELM1 SELM1 SELM1	SELM0 = 0 SELM0 = 0 SELM0 = 1 SELM0 = 1	$\begin{array}{c} 0 \Rightarrow \\ 1 \Rightarrow \\ 0 \Rightarrow \\ 1 \Rightarrow \end{array}$	DCO DCO XT2 LFXT1			
5-4	DIVMx	MC	CLK frequent DIVM1 DIVM1 DIVM1 DIVM1 DIVM1	cy divider: DIVM0 = 0 DIVM0 = 0 DIVM0 = 1 DIVM0 = 1	$ \begin{array}{c} 0 \Rightarrow \\ 1 \Rightarrow \\ 0 \Rightarrow \\ 1 \Rightarrow \end{array} $	/1 /2 /4 /8			
3	SELS	SM	1CLK source SELS = SELS =	: 0 1	$\Rightarrow$	DCO XT2			
2-1	DIVSx	SM	1CLK frequer DIVS1 DIVS1 DIVS1 DIVS1 DIVS1	ncy divider DIVS0 = 0 DIVS0 = 0 DIVS0 = 1 DIVS0 = 1	$ \begin{array}{c} 0 \Rightarrow \\ 1 \Rightarrow \\ 0 \Rightarrow \\ 1 \Rightarrow \end{array} $	/1 /2 /4 /8			
0	DCOR	DC	C resistor s: DCOR = DCOR =	elector = 0 $\Rightarrow$ = 1 $\Rightarrow$	Intern Exterr	al resiston al resisto	r r		

## BCSCTL3, BCS Control Register 3

7 6	5	4	3	2	1	0
XT2Sx	L	FXT1Sx	XC	APx	XT2OF	LFXT1OF

Bit		Description				
7-6	XT2Sx	XT2 range select: XT2S1 XT2S0 = 0 0 XT2S1 XT2S0 = 0 1 XT2S1 XT2S0 = 1 0 XT2S1 XT2S0 = 1 1	↑ ↑ ↑ ↑	0.4 - 1 - 3 3 - 16 0.4 -	1 MHz MHz MHz 16-MHz (Dig	ital external)
5-4	LFXT1Sx	Low-frequency clock select XTS=1: LFXT1S1 LFXT1S0 = LFXT1S1 LFXT1S0 = LFXT1S1 LFXT1S0 = LFXT1S1 LFXT1S0 =	and LF. 0 0 0 1 1 0 1 1	XT1 ran $\Rightarrow$ $\Rightarrow$ $\Rightarrow$ $\Rightarrow$	nge select: XTS=0: 32768 Hz Reserved VLOCLK External	0.4 - 1-MHz 1 - 3-MHz 3 - 16-MHz 0.4 - 16-MHz
3-2	XCAPx	Oscillator capacitor selectio XCAP1 XCAP0 = 0 0 XCAP1 XCAP0 = 0 1 XCAP1 XCAP0 = 1 0 XCAP1 XCAP0 = 1 1	n: $\Rightarrow$ $\Rightarrow$ $\Rightarrow$ $\Rightarrow$ $\Rightarrow$	~1 pF ~6 pF ~10 p ~12.5	F pF	
1	XT2OF	XT2 oscillator fault: XT2OF = 0 $\Rightarrow$ XT2OF = 1 $\Rightarrow$	No fau Fault c	Ilt cond	ition n	
0	LFXT1OF	LFXT1OF oscillator fault: LFXT1OF = 0 $\Rightarrow$ LFXT1OF = 1 $\Rightarrow$	No fau Fault c	Ilt cond	ition n	

□ 4xx family: The DCO generates a signal identified as  $f_{DCOCLK}$ , which is equal to ACLK x D x (N+1).

The DCOPLUS bit sets the  $f_{DCOCLK}$  frequency to  $f_{DCO}$  or  $f_{DCO}/D$ . The FLLDx bits configure the divider, D, to 1, 2, 4 or 8. By default, DCOPLUS = 0 and D = 2 providing a clock frequency of  $f_{DCO}/2$  on  $f_{DCOCLK}$ . The multiplier (N+1) and D set the frequency of DCOCLK.

DCOPLUS = 0:  $f_{DCOCLK} = (N + 1) \times f_{ACLK}$ 

DCOPLUS = 1:  $f_{DCOCLK} = D \times (N + 1) \times f_{ACLK}$ 

The frequency range of  $f_{\text{DCO}}$  is selected with the FNx bits (register SCFI0).

#### 5.2.4 Frequency Locked Loop (FLL)

Devices in the 4xx family have a Frequency Locked Loop circuit, which automatically modulates the DCO frequency, providing greater precision and control. It operates by switching between the two closest neighbour frequencies to our frequency asked for to achieve the frequency requested as a time-weighted average of both frequencies.

The DCO signal is divided by D and afterwards divided by N+1. The signal obtained is continuously applied to the count down input of a 10-bit up/down counter (frequency integrator). Simultaneously, the ACLK signal from the LFXT1 oscillator (reference signal for FLL) is applied to the count up input of the same counter. The counter output feeds back to the DCO modulator, corrects and stabilizes the operating frequency. The output of the frequency integrator that drives the DCO, can be read from the SCFI1 and SCFI0 registers. The count is adjusted by +1 for each ACLK (xtal) period and by -1 for each period of the divided DCO signal.

Twenty-nine DCO frequency taps are set by five of the integrator bits, SCFI1 bits 7 to 3 (28, 29, 30, and 31 are equivalent). Each tap is approximately 10% higher than the previous. The modulator mixes two adjacent DCO frequencies to produce fractional taps.

SCFI1 register bits 2 to 0 and SCFI0 register bits 1 to 0 are used for the digital modulator. The SCFQCTL, SCFI0, SCFI1, FLL\_CTL0, and FLL\_CTL1 registers that configure the FLL+ clock module are shown in the following tables, because the experimenter's board contains one device from the 4xx family (FG4618).

5-8

7	6	5	4	3	2	1	0
SCFQ_M				Ν			

Bit		Description		
7	SCFQ_M	Modulation control:		
		$SCFQ_M = 0$	$\Rightarrow$	FLL modulation enable
		$SCFQ_M = 1$	$\Rightarrow$	FLL modulation disable
6-0	Ν	DCO frequency multiplier fa	actor:	
		DCOPLUS = 0	$\Rightarrow$	$f_{DCOCLK} = (N + 1) f_{crystal}$
		DCOPLUS = 1	$\Rightarrow$	$f_{DCOCLK} = D (N + 1) f_{crystal}$

## SCFI0, System Clock Frequency Integrator Register 0

7	6	5	4	3	2	1	0
FLL	Dx		FN	N_x		MODx	(LSBs)

Bit		Description			
7-6	FLLDx	FLL+ feedback loop f <sub>DCOCLK</sub>	divider:	1	
		FLLD1 FLLD0 = 0 0	$\Rightarrow$	/1	
		FLLD1 FLLD0 = 0 1	$\Rightarrow$	/2	
		FLLD1 FLLD0 = 1 0	$\Rightarrow$	/4	
		FLLD1 FLLD0 = 1 1	$\Rightarrow$	/8	
5-2	FN_x	f <sub>DCO</sub> operating range:			
		0000	$\Rightarrow$	0.65 – 6.1 MHz	
		0001	$\Rightarrow$	1.3 – 12.1 MHz	
		001x	$\Rightarrow$	2.0 – 17.9 MHz	
		01xx	$\Rightarrow$	2.8 – 26.6 MHz	
		1xxx	$\Rightarrow$	4.2 – 46.0 MHz	
1-0	MODx	LSB modulator bits modifie	d by th	e FLL+.	

## SCFI1, System Clock Frequency Integrator Register 1

7	6	5	4	3	2	1	0
		DCOx			ſ	MODx (MSBs	5)

Bit		Description
7-3	DCOx	DCO tap selection modified by the FLL+.
2-0	MODx	MSB modulator bits modified by the FLL+.

# FLL\_CTL0, FLL+ Control Register 0

7	7					0
DCOF	PLUS XTS_FLI	_ XCAPxPF	XT20	OF XT1OF	LFOF	DCOF
Bit		Description				
7	DCOPLUS	DCO output pre-divi selection: DCOPLUS = 0	der. Divider fa $0 \qquad \Rightarrow \\1 \qquad \rightarrow$	Divider enable	bits FLL_	DIV
6	XTS_FLL	LFXT1 oscillator ope XTS_FLL = 0 XTS_FLL = 1	rating mode: $\Rightarrow$ $\Rightarrow$	LF mode (low fr HF mode (high	equency) frequency	·)
5-4	XCAPxPF	LFXT1 oscillator load XCAP1PF XCA XCAP1PF XCA XCAP1PF XCA XCAP1PF XCA	d capacitance: POPF = 0 0 POPF = 0 1 POPF = 1 0 POPF = 1 1	$\begin{array}{ccc} \Rightarrow & \sim 1 \text{ pF} \\ \Rightarrow & \sim 6 \text{ pF} \\ \Rightarrow & \sim 8 \text{ pF} \\ \Rightarrow & \sim 10 \text{ pF} \end{array}$		
3	XT2OF	XT2 oscillator fault: XT2OF = 0 XT2OF = 1	$ \Rightarrow XT2 n  \Rightarrow XT2 fa$	ormal operation ault condition		
2	XT10F	HF mode LFXT1 osci XT1OF = 0 XT1OF = 1	$\begin{array}{llllllllllllllllllllllllllllllllllll$	normal operatio fault condition	n	
1	LFOF	LF mode LFXT1 oscil LFOF = 0 LFOF = 1	$\begin{array}{rllllllllllllllllllllllllllllllllllll$	normal operatio fault condition	n	
0	DCOF	DCO oscillator fault: DCOF = 0 DCOF = 1	$\begin{array}{l} \Rightarrow & DCO \ r \\ \Rightarrow & DCO \ f \end{array}$	normal operation ault condition		

# FLL\_CTL1, FLL+ Control Register 1

7	6	5	4	3	2	1	0
-	SMCLKOFF	XT2OFF	SELMx		SELS	FLL_	DIVx

Bit		Description			
6	SMCLKOFF	SMCLK disable: SMCLKOFF = 0 SMCLKOFF = 1		$\Rightarrow$ $\Rightarrow$	SMCLK enable SMCLK disable
5	XT2OFF	XT2 disable: XT2OFF = 0 XT2OFF = 1	$\Rightarrow$ $\Rightarrow$	XT2 eı XT2 di	nable sable
4-3	SELMx	MCLK source: SELM1 SELM0 = 0 0 SELM1 SELM0 = 0 1 SELM1 SELM0 = 1 0 SELM1 SELM0 = 1 1	$) \Rightarrow$ $\Rightarrow$ $) \Rightarrow$ $\Rightarrow$	DCO DCO XT2 LFXT1	
2	SELS	SMCLK source: SELS = 0 SELS = 1	$\Rightarrow$	DCO XT2	
1-0	FLL_DIVx	ACLK frequency divider: FLL_DIV_0 = 0 0 FLL_DIV_1 = 0 1 FLL_DIV_2 = 1 0 FLL_DIV_3 = 1 1	$\begin{array}{c} \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \end{array}$	/1 /2 /4 /8	

#### 5.2.5 Internal clock signals

The clock system of the 2xx family is supported by the basic clock module, that includes support for a 32.768 kHz watch crystal oscillator, an internal very-low-power low-frequency oscillator and an internal digitally-controlled oscillator (DCO). The basic clock module is designed to meet the requirements of both low system cost and low power consumption. The internal DCO provides a fast turn-on clock source that stabilizes in less than 1 µs. The basic clock module provides the following clock signals:

□ Auxiliary clock (ACLK), sourced either from a 32.768 kHz watch crystal or the internal oscillator LFXT1CLK in LF mode with an internal load capacitance of 6 pF.

□ Main clock (MCLK), the system clock used by the CPU.

□ Sub-Main clock (SMCLK), the sub-system clock used by the peripheral modules.

Both the MCLK and the SMCLK are sourced from DCOCLK at  ${\sim}1.1 \text{MHz}$  (see the device-specific data sheet for parameters), but can be sourced up to 16 MHz.

Table 5-1. 2xx DCO calibration data (in flash info memory segment A).

DCO frequency	Calibration register	Size	Address
1 MHz	CALBC1_1MHZ	Byte	010FFh
	CALBC0 1MHZ	Byte	010FEh
8 MHz	CALBC1_8MHZ	Byte	010FDh
	CALBC0_8MHZ	Byte	010FCh
12 MHz	CALBC1_12MHZ	Byte	010FBh
	CALBC0_12MHZ	Byte	010FAh
16 MHz	CALBC1_16MHZ	Byte	010F9h
	CALBC0_16MHZ	Byte	010F8h

The Status register control bits SCG0, SCG1, OSCOFF, and CPUOFF configure the MSP430 operating modes and enable or disable portions of the basic clock module+.

It is important to be aware of the electrical characteristics over the recommended ranges of supply voltage (see the device-specific data sheet for parameters). Higher DCO frequencies require higher supply voltages, ranging between 2.2 V and 3.6 V.

As an example, the following figure shows the graphs of the typical characteristics in active mode supply current of the 2xx family.



Figure 5-1. Typical characteristics - active mode supply current.

a) Active Mode Current vs.  $V_{CC}$ ,  $T_A = 25^{\circ}C$ 

b) Active Mode Current vs. DCO Freq.

#### 5.3 Interrupt management

An interrupt can be defined as an external event to the application program that forces a change of program flow to execute a CPU subprogram, typically known as an Interrupt Service Routine (ISR). When this routine ends, the program flow returns to the state before the interrupt occurred.

The interrupts are used to allow a CPU to respond quickly to an event, without losing its capability to execute other functions, while waiting for the interrupt to occur.

If both the peripheral interrupt enable bit and GIE are set, when an interrupt is requested, it calls the ISR.

The interrupt latency time is defined as the time interval between the start of the event and the beginning of the ISR execution. This time interval is fixed in the MSP430, requiring 6 clock cycles for the CPU recognize the interrupt and run the ISR call procedure.

During an interrupt event, the Program Counter (PC) of the next instruction and the Status Register (SR) are pushed to the stack. Afterwards, the SR is cleared with exception of SCG0 (4xx devices), along with the appropriate interrupt, disabling interrupts, through resetting the GIE flag. In consequence, other interrupts service routines (ISR) will not be called.

The reti instruction at the end of the ISR will return to program flow, automatically popping the SR and PC.

It is important to ensure that the ISR processing time is less than the interrupt request time interval, because if this condition is not met, the stack will overflow and the application program will crash.

#### 5.3.1 Types of interrupts

The MSP430 offers various interrupt sources, both internal and external. There are three types of interrupts:

- □ Reset;
- □ (Non)-maskable interrupts (NMI) by GIE;
- □ Maskable interrupts by GIE.

Each one of these interrupts has a priority, determining which interrupt is taken when more than one interrupt is pending at any one time. The nearer a module is to the CPU/NMIRS, the higher the priority.

The main difference between non-maskable interrupts and maskable interrupts is the fact that the non-maskable interrupt (NMI) cannot be disabled by the General Interrupt Enable (GIE) bit in the Status Register (SR). NMIs are used for high priority events such as emergency shutdown of a machine.

Because all maskable interrupts are recognized by the CPU's interrupt control, the GIE bit must be set.

The system reset interrupts (Oscillator/Flash and the Hard Reset) are treated as non-maskable interrupts, with highest priority possessing and their own interrupt vectors.

#### Non Maskable Interrupts

NMI is not masked by GIE, but is enabled by individual interrupt enable bits, depending on the event source:

□ NMIIE: Non-Maskable Interrupts Interrupt Enable. When this bit is set, the RST/NMI is configured in NMI mode. A signal edge selected by the WDTNMIES bit generates a NMI interrupt, if the NMIIE bit is set. The RST/NMI flag NMIIFG is also set.

□ ACCVIE: ACCess Violation to the flash memory Interrupt Enable. The flash ACCVIFG flag is set when a flash access violation occurs.

□ OFIE: Oscillator Fault Interrupt Enable. The oscillator fault signal warns of a possible error condition with the crystal oscillator. This kind of signal can be triggered by a PUC signal.

#### Maskable Interrupts

Peripherals with interrupt capability or the watchdog timer overflow in interval timer mode can cause maskable interrupts. Each maskable interrupt also has an individual enable/disable flag, located in peripheral registers or in the individual module. Additionally, all maskable interrupts can be disabled by the general interrupt enable (GIE) bit in the status register (SR).

Each individual peripheral interrupt will be discussed in the associated peripheral module chapter of these teaching materials.

For the devices used (eZ430: F2013; RF2500: F2274; and experimenter's board: FG4618/F2013), the interrupt vectors and addresses are shown in *Table 5-2*.

## Table 5-2. Interrupt vector addresses.

Priority		Interrupt source		Interrupt flag			System interrupt	Word address
	x20x3	x22x4	xG461x	x20x3	x22x4	xG461x		
31 (highest)		Power up External reset Watchdog timer+ Flash key violation			PORIFG RSTIFG WDTIFG KEYV		Reset	OFFFEh
30	Flas	NMI Oscillator fault h memory access viol	ation		NMIIFG OFIFG ACCIFG		Non-maskable Non-maskable Non-maskable	0FFFCh
29		Timer_B3	Timer_B7		TBCCR0 CCIFG	TBCCR0 CCIFG0	maskable	0FFFAh
28		Timer_B3	Timer_B7		TBCCR1 TBCCR2 CCIFGs TBIFG	TBCCR1 TBCCR6 CCIFGs TBIFG	maskable	0FFF8h
27			Comparator_A			CAIFG	maskable	0FFF6h
26	Watchdog timer+	Watchdog timer	Watchdog timer+		WDTIFG		maskable	0FFF4h
25	Timer_A2	Timer_A3	USCI_A0 Receive USCI_B0 Receive	TACCI	R0 CCIFG	UCA0RXIFG UCB0RXIFG	maskable	0FFF2h
24	Timer_A2	Timer_A3	USCI_A0 Transmit USCI_B0 Transmit	TACCR1 CCIFG TAIFG	TACCR1 CCIFG TACCR2 CCIFG TAIFG	UCA0TXIFG UCB0TXIFG	maskable	0FFF0h
23		USCI_A0 Receive USCI_B0 Receive	ADC12		UCA0RXIFG UCB0RXIFG	ADC12IFG	maskable	0FFEEh
22		USCI_A0 Transmit USCI_B0 Transmit	Timer_A3		UCA0TXIFG UCB0TXIFG	TACCR0 CCIFG0	maskable	0FFECh
21	SD16_A	ADC10	Timer_A3	SD16CCTL0 SD16OVIFG SD16CCTL0 SD16IFG	ADC10IFG	TACCR1 TACCR2 CCIFGs TAIFG	Maskable	0FFEAh
20	USI		I/O Port P1	USIIFG USISTTIFG		P1IFG.0 to P1IFG.7 (8 flags)	Maskable	0FFE8h
19	I/O Port P2		USART1 Receive	P2IFG.6 to P2IFG.7 (2 flags)	P2IFG.0 to P2IFG.7 (8 flags)	URXIFG1	Maskable	0FFE6h
18	I/O Port P1		USART1 Transmit	P1IFG.0 (8	to P1IFG.7 flags)	UTXIFG1	Maskable	0FFE4h
17			I/O Port P2			P2IFG.0 to P2IFG.7 (8 flags)	Maskable	0FFE2h
16			Basic Timer1/RTC			BTIFG	Maskable	0FFE0h
15		BSLSKEY	DMA		BSLSKEY	DMA0IFG DMA1IFG DMA2IFG	Maskable	0FFDEh
14			DAC12			DAC12.0IFG DAC12.1IFG	Maskable	0FFDCh
130 (lowest)							Maskable	0FFDAh 0FFC0h

The following figure shows the block diagram of the non-maskable interrupts sources. This block diagram provides all the information related to the interrupt processing of the 2xx family.

Example 1: Flash access violation event interrupt processing. ACCV=1  $\rightarrow$  ACCVIFG=1 ACCVIFG=1 and ACCVIE=1 (set by software)  $\rightarrow$  NMIRS=1

Figure 5-2. Block diagram of the non maskable interrupts and example 1 (2xx family).



## 5.4 Watchdog timer (WDT and WDT+)

The 16-bit watchdog timer (WDT) module can be used as a:

□ Processor supervisor: In supervision mode, the main function of the WDT is to supervise the correct operation of the application software. If a problem occurs with the software application that causes the software to hang or enter an infinite loop, the selected time interval in the watchdog timer is exceeded and the WDT performs a system reset: Power Up Clear (PUC). The procedure in this mode consists of performing an interrupt request on counter overflows. Under normal operating conditions, the watchdog timer would be reset by program code before its timer expires and would therefore inhibit the PUC operation.

□ Interval timer: This module can be configured as an independent interval timer, to perform a "standard" periodic interrupt on counter overflow, for example, to drive an event scheduler (a low-cost operating system). The 16-bit upper counter (WDTCNT) is not directly accessible by software. Its control and the interval time are selected through Watchdog Timer Control Register (WDTCTL). This counter can use the clock signal from ACLK or SMCLK, by defining the appropriate WDTSSEL bit.

The WDT mode is selected by the WDTTMSEL bit in the WDTCTL register. After a PUC condition, the WDT module is configured in supervision mode with approximately 32 msec initial time interval, using DCOCLK. The user should define, stop or clear the WDT before the time interval expires, to prevent a new PUC.

The WDT control is performed through the 16-bit Watchdog Timer Control Register, WDTCTL:

#### WDTCTL, Watchdog Timer Control Register

 $\Box$  Eight MSBs (WDTPW): Password function. These bits always read as 0x69h, and the value 0x5Ah should be written to it, unless the user wants to force a PUC from software.

15		8
	Read with the value 0x69h, WDTPW write with the value 0x5Ah	

□ Eight LSBs: Watchdog timer configuration as described below.

7	6	5	4	3	2	1	0
WDTHOLD	WDTNMIES	WDTNMI	WDTTMSEL	WDTCNTCL	WDTSSEL	WDTIS1	WDTIS0

Bit		Description	
7	WDTHOLD	WDT hold when WDTHOLD = 1. Useful for energy economy.	
6	WDTNMIES	Select the NMI interrupt edge when WDTNMI = 1 WDTNMIES = 0 $\Rightarrow$ NMI on rising edg	e
5	WDTNMI	Select the RST/NMI pin function WDTNMI = 0 $\Rightarrow$ Reset function WDTNMI = 1 $\Rightarrow$ NMI function	JC
4	WDTTMSEL	Select the WDT mode: WDTTMSEL = 0 WDTTMSEL = 1 $\Rightarrow$ Supervision mode Interval timer mode	e de
3	WDTCNTCL	WDT counter clear: WDTCNTCL = 0 $\Rightarrow$ No action Counter initializationWDTCNTCL = 1 $\Rightarrow$ Counter initialization0x0000h	ation at
2	WDTSSEL	Select the WDT clock signal:WDTSSEL = 0 $\Rightarrow$ WDTSSEL = 1 $\Rightarrow$ ACLK	
1-0	WDTISx	Select the WDT timer interval: $\bigcirc$ <t< td=""><td>'68 )2 <u>?</u></td></t<>	'68 )2 <u>?</u>

The WDT uses two bits in the Special Function Registers (SFRs) for interrupt control:

□ WDTIE: The WDT interrupt enable, located in IE1.0.

IE1, I	nterrupt	Enable	Register	1
--------	----------	--------	----------	---

7				0
				WDTIE

Bit		Description	
0	WDTIE	Enables the WDTIFG interrupt for interval timer mode when	
		WDTIE=1	

□ WDTIFG: WDT interrupt flag, located in IFG1.0:

Supervision mode: The WDTIFG flag sources a reset vector interrupt. If WDTIFG=1, the WDT initiates the reset condition, either by timing out or by a password violation. The user has control of the reset source.

Interval mode: The WDTIFG flag is set after the selected time interval and requests a WDT interval timer interrupt if the WDTIE and the GIE bits are set. When the interrupt is serviced, the WDTIFG flag is reset automatically. It also can be reset using software.

#### IFG1, Interrupt Flag Register 1

7				0
				WDTIFG

BitDescription0WDTIFGInterrupt flag set by WDT overflow (supervision mode)

#### 5.4.1 Watchdog Timer+ (WDT+)

The 2xx family devices and some devices from 4xx family include a watchdog timer with extended features. The main difference is an additional safety circuit, which monitors the watchdog timer clock signal. When the WDT is configured in supervision mode, the clock to the WDT+ cannot be disabled because of the fail-safe clocking feature.

This feature provides safety using low power modes. This is because it will disable the low power mode LPM4 (see next section for further details) if the clock signal is sourced by ACLK, as well as shifting the clock signal if an error occurs to the actual clock signal.

## 5.5 Supervisory Voltage System (SVS)

The supply voltage supervisor (SVS) module is used to monitor the  $AV_{CC}$  supply voltage or an external voltage located at the SVSIN input. The SVS can be configured to set a flag generating an interrupt or generate a system reset (POR) when the supply voltage or external voltage drops below a user-selectable threshold. The SVS is disabled after a brownout reset to minimise current consumption.

The SVS features include:

- $\Box$  AV<sub>CC</sub> monitoring;
- □ Selectable generation of POR;
- Output of SVS comparator accessible by software;
- □ Low-voltage condition latched and accessible by software;
- □ 14 selectable threshold levels;
- □ External channel to monitor external voltage.

The core of this module is an analogue comparator, with one input connected to an internal reference voltage of approximately 1.25 V and the other input is connected to the supply voltage (AV<sub>CC</sub> or SVSIN), by an analogue multiplexer. This allows the selection of the voltage level based on a percentage of the monitored voltage. This level,  $V_{(SVS\_IT-)}$ , (one of 14 levels), enables/disables SVS and is controlled by the configuration of the VLDx bits in SVS Control Register (SVSCTL).

The voltage level can be monitored by the SVSOP bit in the SVSCTL register. When the supply voltage  $AV_{CC}$  drops below the selected threshold or when the external voltage drops below the 1.25 V internal reference voltage level, the SVSFG bit is set.

If the PORON bit is set, a POR is generated when SVSFG is set. Otherwise, a low-voltage condition sets SVSFG, without generating a POR.

The SVSFG bit must be reset by user software.

#### SVSCTL, Supervisory Voltage System Control Register

7	6	5	4	3	2	1	0
	VL	Dx		PORON	SVSON	SVSOP	SVSFG

Bit		Description
7-4	VLDx	Voltage level detect.
		VLD3 VLD2 VLD1 VLD0 = 0000 $\Rightarrow$ SVS is off
		VLD3 VLD2 VLD1 VLD0 = 0001 $\Rightarrow$ 1.9 V
		VLD3 VLD2 VLD1 VLD0 = 0010 $\Rightarrow$ 2.1 V
		VLD3 VLD2 VLD1 VLD0 = 0011 $\Rightarrow$ 2.2 V
		VLD3 VLD2 VLD1 VLD0 = 0100 $\Rightarrow$ 2.3 V
		VLD3 VLD2 VLD1 VLD0 = 0101 $\Rightarrow$ 2.4 V
		VLD3 VLD2 VLD1 VLD0 = 0110 $\Rightarrow$ 2.5 V
		VLD3 VLD2 VLD1 VLD0 = 0111 $\Rightarrow$ 2.65 V
		VLD3 VLD2 VLD1 VLD0 = 1000 $\Rightarrow$ 2.8 V
		VLD3 VLD2 VLD1 VLD0 = 1001 $\Rightarrow$ 2.9 V
		VLD3 VLD2 VLD1 VLD0 = 1010 $\Rightarrow$ 3.05
		VLD3 VLD2 VLD1 VLD0 = 1011 $\Rightarrow$ 3.2 V
		VLD3 VLD2 VLD1 VLD0 = 1100 $\Rightarrow$ 3.35 V
		VLD3 VLD2 VLD1 VLD0 = 1101 $\Rightarrow$ 3.5 V
		VLD3 VLD2 VLD1 VLD0 = 1110 $\Rightarrow$ 3.7 V
		VLD3 VLD2 VLD1 VLD0 = 1111 $\Rightarrow$ Compares SVSIN to 1.25V
3	PORON	When PORON = 1 enables the SVSFG flag to cause a POR device reset
2	SVSON	This bit reflects the status of SVS operation, being set (SVSON=1) when the SVS is on
1	SVSOP	This bit reflects the output value of the SVS comparator: $SVSOP = 0 \implies SVS$ comparator output is low $SVSOP = 1 \implies SVS$ comparator output is high
0	SVSFG	When SVSFG=1 a low voltage condition occurred
		2

### 5.6 Low power operating modes

This section presents one of the main features of the MSP430 families, that is their low power consumption (around 1 mW/MIPS or less). This is increasingly important with the growth of battery operated embedded systems devices.

Although the MSP430 families are designed for low power consumption, it should borne in mind that this goal can only be accomplished using a design utilizing low power operating modes. The total power consumption depends on several factors: clock frequency, ambient temperature, supply voltage, peripheral selection, input/output usage and memory type.

#### 5.6.1 Low power modes

The MSP430 architecture allows six operating modes, five of these modes are suitable for low power consumption operation. These modes are configured by the Status Register bits: CPUOFF; OSCOFF; SCG1 and SCG0 as follows:

- □ Active mode (AM):
  - Configured disabling the SR bits described above;
  - CPU is active;
  - All enabled clocks are active;
  - Current consumption: 250  $\mu$ A.

It can be seen from Table 5-3, that because all modules are active, this mode presents the highest power consumption.

The MSP430 allows the software selection up to five low-power modes of operation. An interrupt event can wake up the device from any of the five low-power modes, service the interrupt request and then revert back to the low-power mode. These modes are shown in the following table, as well as SR bits configuration and their functionalities.

Mode	Current	SR b	Clock signals			Oscillator				
	~ [µA]	CPUOFF	OSCOFF	SCG1	SCG0	ACLK	SMCLK	MCLK	DCO	DC gen.
Low-power mode 0 (LPM0)	35	1	0	0	0	1	1	0	1	1
Low-power mode 1 (LPM1)	44	1	0	0	1	1	1	0	1	1*
Low-power mode 2 (LPM2)	19	1	0	1	0	1	0	0	0	1
Low-power mode 3 (LPM3)	0.8	1	0	1	1	1	0	0	0	0
Low-power mode 4 (LPM4)	0.1	1	1	1	1	0	0	0	0	0

Table 5-3. MSP430 low power operation modes.

\*The DCO's DC generator is enabled if it is used by peripherals.

- □ Characteristics of Low-power mode 0 (LPM0) to 3 (LPM3):
  - Suitable for periodic processing based on a timer interrupt.
  - LPM0 must be used when the DCO source signal is required as well as the DCO's DC generator.
  - The main difference between LPM0 and LPM1 lies in the ability of LPM0 to enable/disable the DCO's DC generator, depending on the peripherals used.
  - In LPM2, the DCO's DC generator remains active, even though the DCO is disabled.
  - In LPM3, only the ACLK is active. This condition provides the lower power consumption periodic interrupt capability (less than 2 µA typical with both a real-time clock function and with all interrupts active).

 $\square$  LPM4 is useful for systems that use externally generated interrupts only, as no clocks are active and available for peripherals. This mode uses the least current consumption, typically around 0.1  $\mu A.$ 

Figure 5-3. Typical current consumption vs Operating modes (21x1 devices).



The program flow using low power operating modes is as follows:

- □ Enter Low-power mode:
  - Enable/disable CPUOFF, OSCOFF, SCG0 and SCG1 bits in the status register;
  - The selected low power mode is active after writing to the SR;
  - The CPU will suspend execution of the program;
  - Peripherals operating from any disabled clock are disabled, until the clock becomes active again;

- The peripherals may also be disabled with their individual control register settings;
- All I/O port pins and RAM/registers are unchanged;
- Wake up is possible through any enabled interrupt;
- **Example:** LPM3 power consumption is less than 2  $\mu$ A.

□ An enabled interrupt event wakes the MSP430 from any of the low power operating modes, returning to active mode:

Example: LPM3 takes around 6 µs to wake up the DCO. It is important to mention that for extended low power mode periods where the DCO is disabled, the ambient temperature might change significantly. If the temperature variation changes the temperature coefficient of the DCO, its wake up frequency will be different. This effect can be minimized by setting the DCO to it lowest value, before entering the lowpower mode. Another related feature is the immediate stable clock start, to react to an interrupt event, for example, as shown in *Figure 5-4*.

Figure 5-4. DCO performance on demand.



- Enter ISR:
  - The present operating mode is saved on the stack during the interrupt service routine;
  - The PC and SR are stored on the stack;
  - The interrupt vector is moved to the PC;
  - The CPUOFF, SCG1, and OSCOFF bits are automatically reset, enabling normal operation of the CPU (SCG0 is also cleared in case of 2xx device);
  - The IFG flag is cleared.

- Returning from the ISR:
  - The original SR is popped from the stack, restoring the previous operating mode;
  - The SR bits stored on the stack are modified when the ISR returns to a different operating mode on execution of the RETI instruction.

The following table provides examples of applications development using the MSP430 with and without consideration for low power modes.

Table 5-4. Examples of applications development using the MSP430 with and without consideration for low power modes.

Example	Without low power mode	With low power mode
Toggling the bit 0 of port 1 (P1.0) periodically	Endless loop (100 % CPU load)	LPM0 Watchdog timer interrupt
UART to transmit the received message at a 9600 baud rate	Polling UART receive (100 % CPU load)	UART receive interrupt (0.1 % CPU load)
Set/reset during a time interval, periodically, of the peripheral connected to the bit 2 of port 1 (P1.2)	Endless loop (100 % CPU load)	Setup output unit (Zero CPU load)
Power manage external devices such as Op-Amps (OPAs)	Putting the OPA Quiescent (Average current: 1 μA)	Shutdown the OPA between data acquisition (Average current: 0.06 μA)
Power manage internal devices like Comparator A	Always active (Average typical current: 35 µA)	Disable Comparator A between data acquisition
Respond to button-press interrupt at P1.0 and toggle LED on P2.1	Endless loop (100 % CPU load)	Using LPMs while the LED is switch off: LPM3: 1.4 µA LPM4: 0.3 µA Configure unused ports in output direction P1 interrupt service routine

The following items give some rules-of-thumb for the correct configuration of low power applications:

□ Extended ultra-low power standby mode. Maximize the time in LPM3, if clocks are not needed;

- Minimise the active duty cycle;
- □ Give performance on-demand;
- □ Use interrupts to control program flow;
- □ Replace software with on-chip peripherals;

□ Manage the power of external devices;

□ Configure unused pins properly, setting them to outputs, to avoid floating gate current;

- □ Use low-power efficient coding techniques:
  - Optimize program flow;
  - Use CPU registers for calculations and dedicated variables;
  - Use the same code size for word or byte;
  - Use word operations whenever possible;
  - Use the Constant Generator for the values 0xFFFF, 0, 1, 2, 4 and 8 to reduce code size and cycles;
  - Use local variable (CPU registers) as much as possible, instead of global variables (RAM);
  - Use bit mask instead of bitfields;
  - Use unsigned data types where possible;
  - Use pointers to access structures and unions;
  - Use the "static const" class to avoid run-time copying of structures, unions, and arrays;
  - Avoid modulo;
  - Avoid floating-point operations;
  - Use count down "for" loops;
  - Use short ISR.

#### 5.7 Quiz

- 1. The operating mode of the MSP430 is:
- (a) Determined by the program counter (PC) register;
- (b) Determined by the state of the CPU;
- (c) Determined by four control bits in the status register (SR);
- (d) All of above.

**2.** The MSP430 clock system control registers of the 2xx family hardware development tools (eZ430-F2013 and eZ430-RF2500) are:

- (a) Registers R4 to R9;
- (b) Register BCSCTL1, BCSCTL2 and DCOCTL;
- (c) Registers SCFQCTL, SCFI0, SCFI1 and FLL\_CTL0;
- (d) Registers R13, R14 and R15.

3. If the XTS bit in the BCSCTL1 control register is enabled:

(a) The LFXT1 oscillator in the clock system can operate with a high-frequency crystal;

(b) The LFXT1 oscillator is OFF;

(c) The LFXT1 oscillator in the clock system can operate with a low-frequency crystal;

(d) The XT2 oscillator is enabled.

 ${\bf 4.}$  When the SELS bit in the FLL\_CTL1 control register of the MSP430FG4618 is reset:

- (a) The DCOCLK is OFF;
- (b) The SMCLK is divided by 8;
- (c) The source for the SMCLK clock is LFXT1 oscillator;
- (d) The source for the SMCLK clock is DCOCLK.

**5**. In the MSP430, when the watchdog timer control bit WDTTMSEL is set:

- (a) The watchdog timer is an interval timer (T);
- (b) The watchdog timer is inactive;
- (c) Clears the watchdog timer counter;
- (d) Restarts the watchdog timer.
- 6. The 16-bit WDTCTL control register must have:
- (a) All its high byte-bits at 0;

(b) A 0x069h value in the high-byte when WDTCTL is read and a 0x05Ah password must be written to the high-byte, to write to WDTCTL;

(c) A 0x05Ah password in the high-byte to read and write to WDTCL;

(d) All bit of its high-byte are 1.

**Solution:** 1. (c); 2. (b); 3. (a); 4. (d); 5. (a); 6. (b)

## 5.8 FAQs

**1**. Should a signal generating an NMI event hold the RST/NMI pin low when the device is configured in NMI mode?

No. If a PUC occurs from a different source while the NMI signal is low, the device will be held in the reset state.

**2.** Can a NMI be generated when in NMI mode and the WDTNMIES bit is changed?

It depends on the actual level at the RST/NMI pin. If the NMI edge select bit is changed before selecting the NMI mode, no NMI is generated.

3. Can I enable ACCVIE, NMIIE, and OFIE bits inside an NMI ISR?

This is not recommended. It can cause nested NMI interrupts that may result in inappropriate program flow

**4**. When developing a PCB for the MSP430 application system, is there any recommendation for the location of the external LFXT1?

The crystal should be placed as close as possible to the MSP430, with the crystal housing grounded and the crystal tracks guarded with ground tracks.

5. Is the DCO active during an oscillator fault?

Yes. The clock signal is available for the CPU to execute code and service an NMI during an oscillator fault.

 ${\bf 6.}$  What is the initial configuration of the Watchdog Timer after a PUC?

The WDT module is automatically configured in the watchdog mode with an initial 32 clock-cycle reset interval, using the DCOCLK. The user must setup or halt the WDT prior to the expiration of the initial reset interval.

**7.** What is the procedure to change the configuration of the Watchdog Timer?

The WDT interval should be changed together with WDTCNTCL = 1 in a single instruction, to avoid an unexpected immediate PUC or interrupt.

The WDT should be halted before changing the clock source to avoid a possible incorrect time interval.