Chapter 1

MSP430 Microcontroller Family

1.1 Introduction

The MSP430 is a 16-bit microcontroller that has a number of special features not commonly available with other microcontrollers:

- ❑ Complete system on-a-chip — includes LCD control, ADC, I/O ports, ROM, RAM, basic timer, watchdog timer, UART, etc.
- ❑ Extremely low power consumption — only 4.2 nW per instruction, typical
- ❑ High speed — 300 ns per instruction @ 3.3 MHz clock, in register and register addressing mode
- ❑ RISC structure — 27 core instructions
- ❑ Orthogonal architecture (any instruction with any addressing mode)
- ❑ Seven addressing modes for the source operand
- ❑ Four addressing modes for the destination operand
- ❑ Constant generator for the most often used constants (–1, 0, 1, 2, 4, 8)
- ❑ Only one external crystal required — a frequency locked loop (FLL) oscillator derives all internal clocks
- ❑ Full real-time capability — stable, nominal system clock frequency is available after only six clocks when the MSP430 is restored from low-power mode (LPM) 3; — no waiting for the main crystal to begin oscillation and stabilize

The 27 core instructions combined with these special features make it easy to program the MSP430 in assembler or in C, and provide exceptional flexibility and functionality. For example, even with a relatively low instruction count of 27, the MSP430 is capable of emulating almost the complete instruction set of the legendary DEC PDP-11.

Note:

The software examples provided in this document have been tested for functionality and may be used freely for system development.

1.2 Related Documents

The following documents are recommended for MSP430 reference:

- The *MSP430 Architecture User's Guide and Module Library* (TI literature number SLAUE10B) contains a detailed hardware description.
- The *MSP430 Software User's Guide* (TI literature number SLAUE11) contains further information regarding the instruction set, plus other more common software information.

1.3 Notation

The following abbreviations and special notations are used:

.and.	Logical AND function
.not.	Logical Inversion
.or.	Logical OR function
.xor.	Logical Exclusive-OR function
[ns]	Square brackets contain the unit for a value (here nanoseconds)
ACLK	Auxiliary clock (output of the 32-kHz oscillator)
ACTL.1	Bit 1 (value 2^1) of the register ACTL
ADC	Analog-to-digital converter
AGND	Ground connection for the ADC; Vss (MSP430x31x) or AVss (MSP430x32x)
Background	Normal program
BCD	Binary coded decimal (numbers 0 to 9 coded binary with 4 bits)
CPU	Central processing unit
DCO	Digitally controlled oscillator
(dst)	Destination (location receiving write data)
Foreground	Interrupt driven software parts (interrupt handlers)
I/O	Input and output Port
LCD	Liquid crystal display
LSB	Least significant bit (or byte)
MCLK	Master clock (output of the FLL oscillator) for the CPU
MSB	Most significant bit (or byte)
PC	Program counter (R0 of register set)
R1||R2	Resistor R1 is connected in parallel with resistor R2
R4|R3	32-bit number. MSBs in CPU register R4, LSBs in R3
RAM	Random access memory (data memory)

ROM	Read only memory (program memory)
SP	Stack pointer (R1 of register set)
(src)	Source (location supplying read data)
TOS	Top of stack (data word the Stack Pointer SP points to)

NOTES:If no units are defined for equations, the following standard units are used: Volt, Ampere, Farad, seconds and Ohm.

1.4 MSP430 Family

The MSP430 family currently consists of three subfamilies:

- MSP430C31x
- MSP430C32x
- MSP430C33x

All three are described in detail in the *MSP430 Family Architecture User's Guide and Module Library.* The hardware features of the different devices are shown in Table 1, Figure 1, Figure 2, and Figure 3.

Table 1–1. MSP430 Sub-Families Hardware Features

Hardware Item	MSP430C31x	MSP430C32x	MSP430C33x
14-bit ADC	No	Yes	No
16-bit timer_A	No	No	Yes
Basic timer	Yes	Yes	Yes
FLL oscillator	Yes	Yes	Yes
HW/SW UART	Yes	Yes	Yes
HW-multiplier	No	No	Yes
I/O ports with interrupt	8	8	24
I/O ports without interrupt	0	0	16
LCD segment lines	23	21	30
Package	56 SSOP	64 QFP	100 QFP
Universal timer/port module	Yes	Yes	Yes
USART (SCI or SPI)	No	No	Yes
Watchdog timer	Yes	Yes	Yes

NOTE: Examples and explanations in this document are applicable for all MSP430 devices, unless otherwise noted.

1.4.1 MSP430C31x

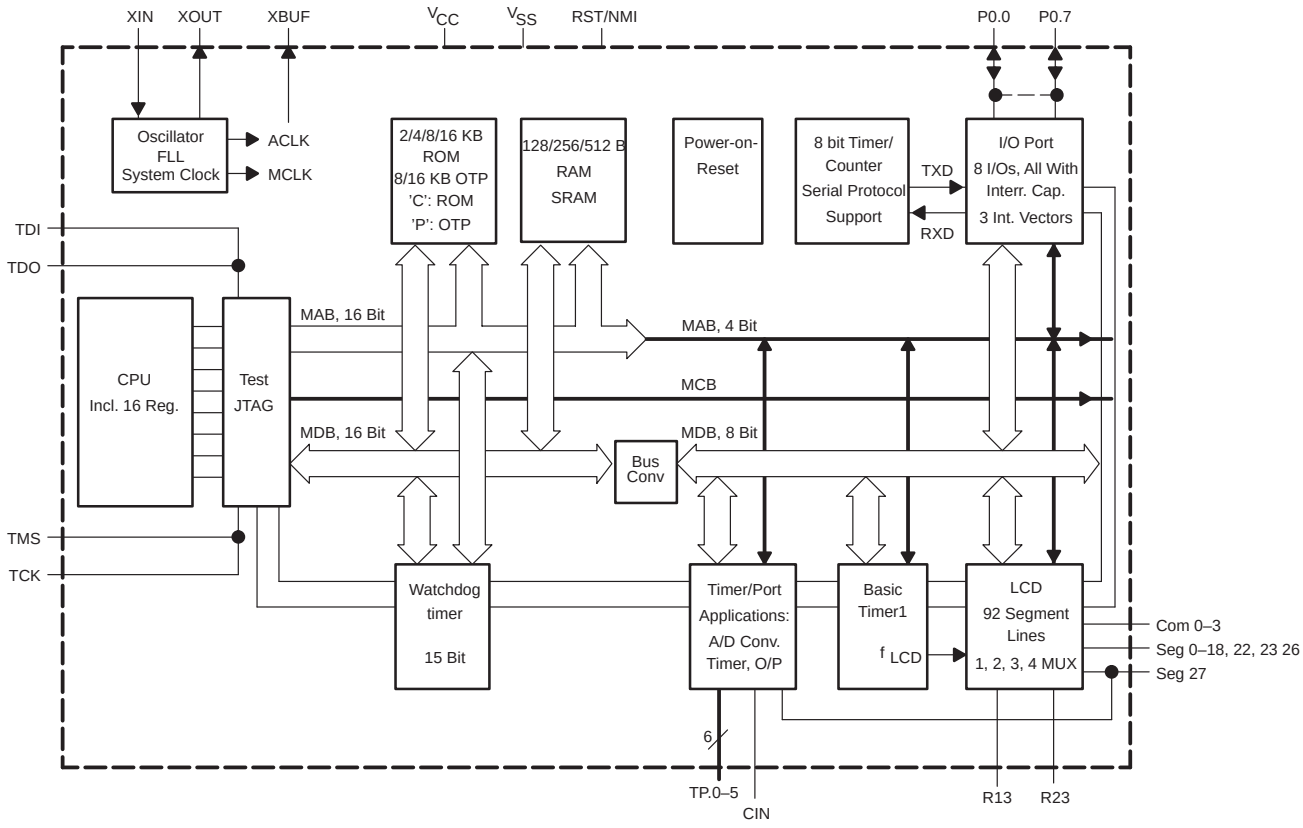

Figure 1–1. MSP430C31x Block Diagram

1.4.2 MSP430C32x

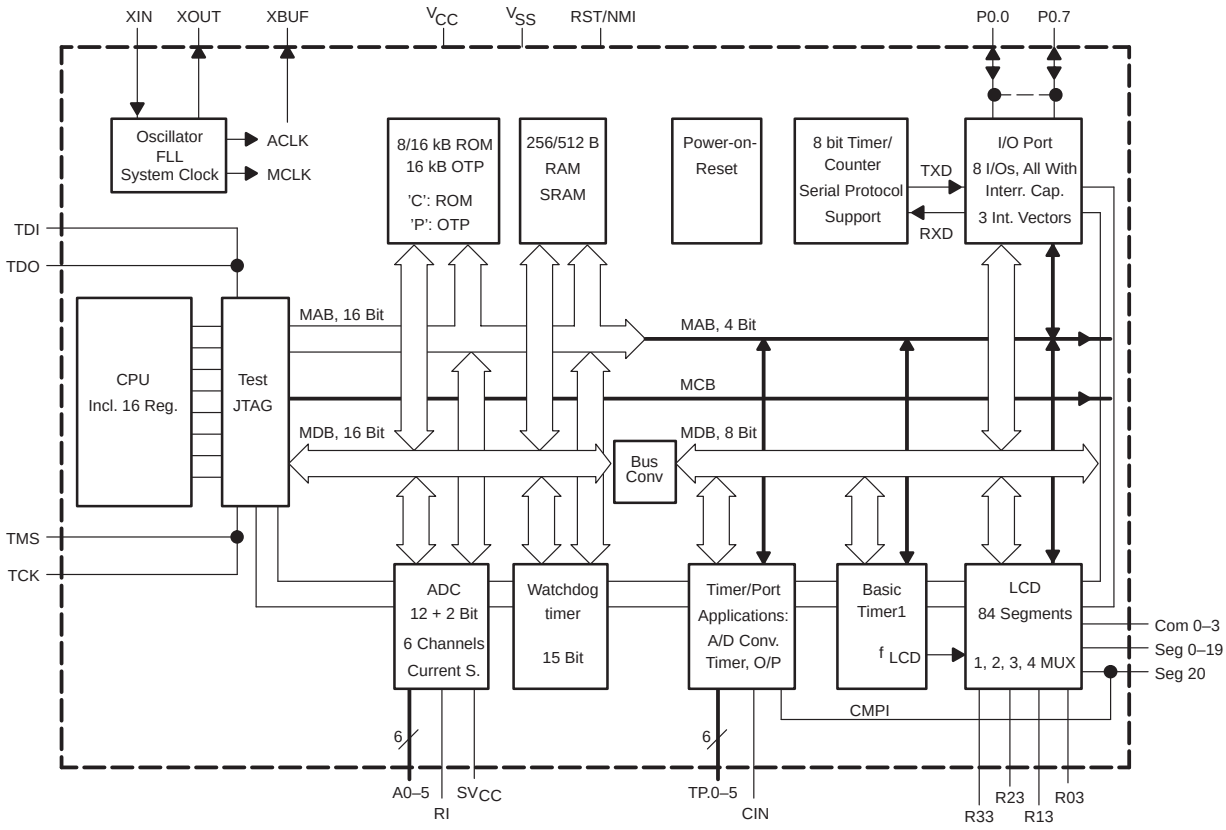

Figure 1–2. MSP430C32x Block Diagram

1.4.3 MSP430C33x

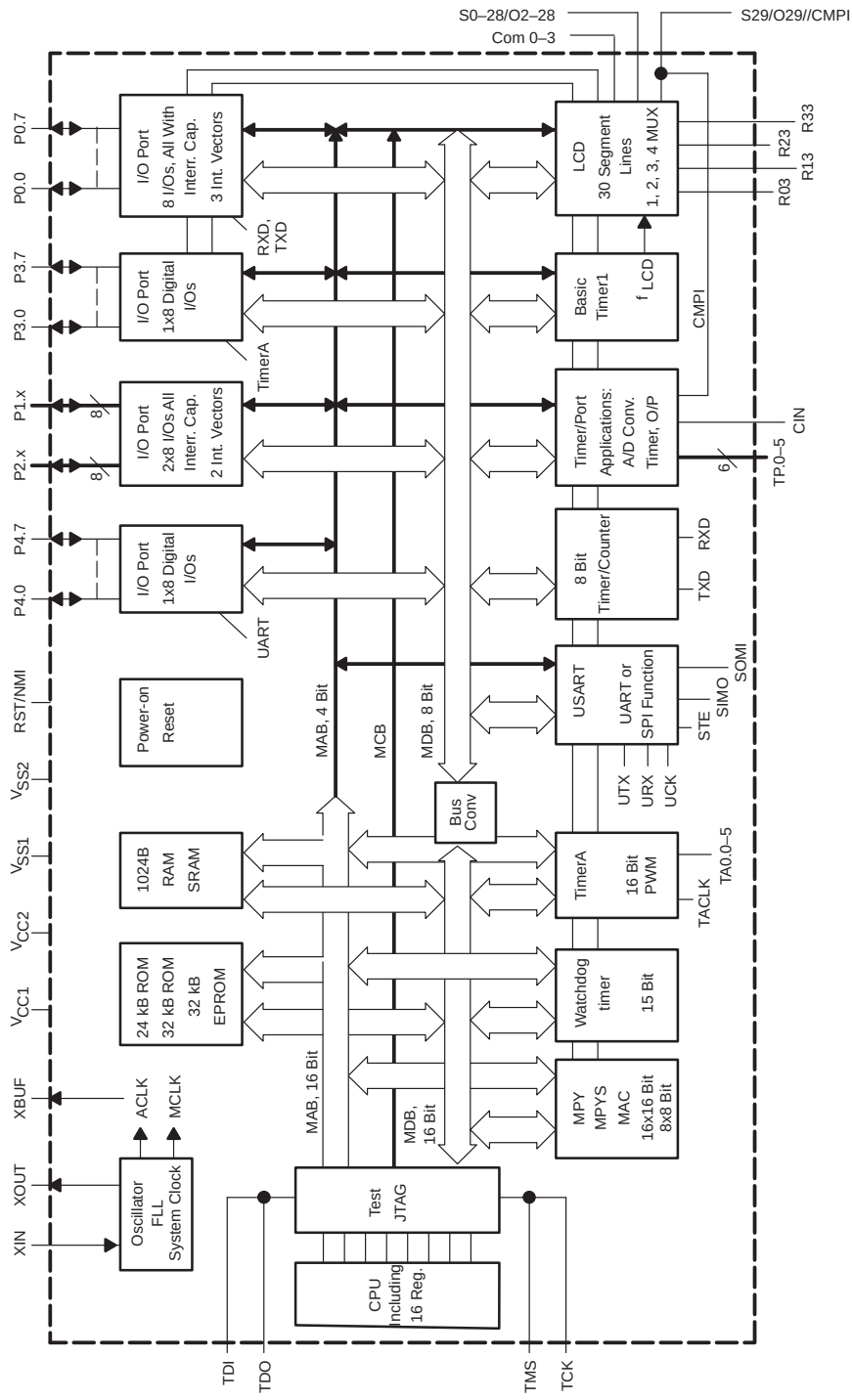

Figure 1–3. MSP430C33x Block Diagram

1.5 Advantages of the MSP430 Concept

The MSP430 concept differs considerably from other microcontrollers and offers some significant advantages over more traditional designs.

1.5.1 RISC Architecture Without RISC Disadvantages

Typical RISC architectures show their highest performance in calculation- intensive applications in which several registers are loaded with input data, all calculations are made within the registers, and the results are stored back into RAM. Memory accesses (using addressing modes) are necessary only for the LOAD instructions at the beginning and the STORE instructions at the end of the calculations. The MSP430 can be programmed for such operation, for example, performing a pure calculation task in the floating point without any I/O accesses.

Pure RISC architectures have some disadvantages when running real-time applications that require frequent I/O accesses, however. Time is lost whenever an operand is fetched and loaded from RAM, modified, and then stored back into RAM.

The MSP430 architecture was designed to include the best of both worlds, taking advantage of RISC features for fast and efficient calculations, and addressing modes for real-time requirements:

- ❏ The RISC architecture provides a limited number of powerful instructions, numerous registers, and single-cycle execution times.
- ❏ The more traditional microcomputer features provide addressing modes for *all* instructions. This functionality is further enhanced with 100% orthogonality, allowing any instruction to be used with any addressing mode.

1.5.2 Real-Time Capability With Ultra-Low Power Consumption

The design of the MSP430 was driven by the need to provide full real-time capability while still exhibiting extremely low power consumption. Average power consumption is reduced to the minimum by running the CPU and certain other functions of the MSP430 only when it is necessary. The rest of the time (the majority of the time), power is conserved by keeping only selected low-power peripheral functions active.

But to have a true real-time capability, the device must be able to shift from a low-power mode with the CPU off to a fully active mode with the CPU and all other device functions operating nominally in a very short time. This was accomplished primarily with the design of the system clock:

- No second high frequency crystal is used — inherent delays can range from 20 ms to 200 ms until oscillator stability is reached
- Instead, a sophisticated FLL system clock generator is used — generator output frequency (MCLK) reaches the nominal frequency within 8 cycles after activation from low power mode 3 (LPM3) or sleep mode

This design provides real-time capability almost immediately after the device comes out of a LPM — as if the CPU is always active. Only two additional MCLK cycles (2 µs @ f_C = 1 MHz) are necessary to get the device from LPM3 to the first instruction of the interrupt handler.

1.5.3 Digitally Controlled Oscillator Stability

The digitally controlled oscillator (DCO) is voltage and temperature dependent, which does not mean that its frequency is not stable. During the active mode, the integral error is corrected to approximately zero every 30.5 µs . This is accomplished by switching between two different DCO frequencies. One frequency is higher than the programmed MCLK frequency and the other is lower, causing the errors to essentially cancel-out. The two DCO frequencies are interlaced as much as possible to provide the smallest possible error at any given time. See *System Clock Generator* for more information.

1.5.4 Stack Processing Capability

The MSP430 is a true stack processor, with most of the seven addressing modes implemented for the stack pointer (SP) as well as the other CPU registers (PC and R4 through R15). The capabilities of the stack include:

- Free access to all items on the stack — not only to the top of the stack (TOS)
- Ability to modify subroutine and interrupt return addresses located on the stack
- Ability to modify the stored status register of interrupt returns located on the stack
- No special stack instructions — all of the implemented instructions may be used for the stack and the stack pointer
- Byte and word capability for the stack
- Free mix of subroutine and interrupt handling — as long as no stack modification (PUSH, POP, etc.) is made, no errors can occur

For more information concerning the stack, see *Appendix A*.

1.6 MSP430 Application Operating Modes

MSP430 applications fall into two main classes, depending on the power supply:

- AC power-driven applications such as electricity meters and AC-powered controllers. In these applications, the microcontroller needs to be active at all times. The low current consumption of the MSP430 when active (900 μA @ 5 V & f_C = 1 MHz) puts it well within the typical low-power category now (currently < 40 mA) and in the future as tolerable current consumption diminishes.
- Battery-powered applications such as gas meters, water flow meters, heat volume counters, data loggers, and other controller and remote metering tasks. For these applications, power consumption is the key issue since operation from a single battery for 10 years or longer is often required. The average current drawn by the MSP430 needs to be in the range of the self discharge current of the battery, approximately 1 μA to 3 μA.

MSP430 has six operating modes, each with different power requirements. Three of these modes are important for battery-powered applications:

- Active mode — CPU and other device functions run all the time
- Low power mode 3 (LPM3) — the normal mode for most applications during 99% to 99.9% of the time. This mode is also called done mode or sleep mode
- Low power mode 4 (LPM4) — the mode typically used during storage. This mode is also called off mode

1.6.1 Active Mode

Active mode is used for calculations, decision-making, I/O functions, and other activities that require the capabilities of an operating CPU. All of the peripheral functions may be used, provided that they are enabled. The examples shown in this document use the active mode.

1.6.2 Low Power Mode 3 (LPM3)

LPM3 is the most important mode for battery-powered applications. The CPU is disabled, but enabled peripherals stay active. The basic timer provides a precise time base. When enabled, interrupts restore the CPU, switch on MCLK, and start normal operation. Table 1–2 lists the status of the MSP430 system when in LPM3.

Table 1–2. System Status During LPM3

Active	Not Active
RAM	CPU
ACLK	MCLK
32768 Hz oscillator	Disabled peripherals
LCD driver (if enabled)	Disabled interrupts
Basic timer (if enabled)	FLL
I/O ports	
8-bit timer	
Enabled peripherals	
Universal timer/port	
RESET logic	

LPM3 is activated by the following code:

```
;
; Definitions for the Operating Modes
;
GIE       .EQU  008h  ; General Interrupt enable in SR
CPUOFF    .EQU  010h  ; CPU off bit in SR
OSCOFF    .EQU  020h  ; Oscillator off bit in SR
SCG0      .EQU  040h  ; System Clock Generator Bit 0
SCG1      .EQU  080h  ; System Clock Generator Bit 1
HOLD      .EQU  080h  ; 1: Hold Watchdog
CNTCL     .EQU  008h  ; Watchdog Reset Bit
;
; Enter LPM3, enable interrupts. The Watchdog
; must be held if the ACLK is used for timing
;
      MOV   #05A00h+HOLD+CNTCL,&WDTCTL  ; Define WD
      BIS   #CPUOFF+GIE+SCG1+SCG0,SR    ; Enter LPM3
;
```

After the completion of the interrupt routine the software returns to the instruction that set the CPUoff bit. The normal wake-up from LPM3 comes from the basic timer, programmed to wake the CPU at regular intervals (ranging from 0.5 Hz to 64 Hz, or more often) to maintain a software timer. This software timer controls all necessary system activities.

Example 1–1. Interrupt Handling I

The MSP430 system runs normally in LPM3. The enabled interrupt of the basic timer wakes the system once every second. After one minute, measurements are made and then the system returns to LPM3.

```
;
```

```
; Interrupt handler for Basic Timer: Wake–up with 1Hz
;
BT_HAN MOV    #05A00h+CNTCL,&WDTCTL   ; Reset watchdog
       INC.B  SECCNT                  ; Counter for seconds +1
       CMP.B  #60,SECCNT              ; 1 minute elapsed?
       JHS    MIN1                    ; Yes, do necessary tasks
       RETI                           ; No return to LPM3
;
; One minute elapsed: Return is removed from stack, a branch to
; the necessary tasks is made. There it is decided how to proceed
;
MIN1   INC    MINCNT                  ; Minute counter +1
       CLR    SECCNT                  ; 0 –> SECCNT
       ADD    #4,SP                   ; House keeping: SR, PC off Stack
       BR     #TASK                      ; Do tasks
       ...
;
TASK ...                                 ; Start of necessary tasks
;
; All measurements and calculations are made: Return to LPM3
;
       MOV    #05A00h+HOLD+CNTCL,&WDTCTL ; Hold WD
       BIS    #CPUOFF+GIE+SCG0+SCG1,SR   ; Enter LPM3
```

LPM3 is the lowest current consumption mode that still allows the use of a real-time clock. The basic timer can interrupt the LPM3 at relatively long time intervals (up to 2 seconds) and update the real-time clock. If the status register is not changed during the interrupt routines, the RETI instruction returns to the instruction that set the CPUoff bit (and placed the CPU in LPM3). The program counter points to the next instruction, which is not executed unless the interrupt routine resets the CPUoff bit during its run.

If the MSP430 is awakened from LPM3, two additional clock cycles are needed to load the PC with the interrupt vector address and start the interrupt handler (8 clocks compared to 6 when in the active mode).

Example 1–2. Interrupt Handling II

The MSP430 system runs normally in LPM3. The enabled interrupt of the basic timer wakes the system once every second. After one minute, measurements are made and then the system returns to LPM3. The branch to the task is made by resetting the CPUoff bit inside the interrupt routine.

```
; Interrupt handler for Basic Timer: Wake–up with 1 Hz
```

```
;
BT_HAN MOV    #05A00h+CNTCL,&WDTCTL      ; Reset watchdog
       INC.B  SECCNT                     ; Counter for seconds +1
       CMP.B  #60,SECCNT                 ; 1 minute over?
       JHS    MIN1                       ; Yes, do necessary tasks
       RETI                              ; No return to LPM3
;
; One minute elapsed: CPUoff is reset, the program continues
; after the instruction that set the CPUoff bit (label TASK)
;
MIN1   CLR    SECCNT                     ; 0 –> SECCNT
       INC    MINCNT                     ; Minute counter + 1
       BIC    #CPUOFF+SCG1+SCG0,0(SP)             ; Reset CPUoff–bit to
continue
       RETI                              ; at label TASK
;
; Background part: Return to LPM3
;
DONE   MOV    #05A00h+HOLD+CNTCL,&WDTCTL ; Hold WD
       BIS    #CPUOFF+GIE+SCG0+SCG1,SR   ; Enter LPM3
;
; Program continues here if CPUoff bit was reset inside of the
; Basic Timer Handler.
;
TASK   ...                               ; Tasks made every minute
       JMP    DONE                       ; Back to LPM3
```

Note:

The two 8-bit counters of the universal timer/port may also be used during LPM3. If a counter is incremented by an external signal (inputs CIN, CMPI, or TPIN.5) from 0FFh to 0h, then the appropriate RCxFG-flag is set. If interrupt is enabled, the CPU wakes up.

1.6.3 Low Power Mode 4 (LPM4)

Low power mode 4 (LPM4) is used if the absolute lowest supply current is necessary or if no timing is needed or desired (no change of the RAM content is allowed). This is normally the case for storage preceding or following the calibration process. Table 3 lists the status of the MSP430 system when in LPM4.

Table 1–3. System During LPM4

Active	Not Active
RAM	CPU
I/O ports	MCLK
Enabled interrupts	ACLK
Universal timer/port (external clock)	FLL
RESET logic	Disabled peripherals
	Disabled interrupts
	Watchdog
	Timers

Once the MSP430 is waked from LPM4, the software has to decide if it is necessary to either enter LPM4 again (if the wake-up was caused by EMI, for example), or to enter one of the other operating modes. To ensure the correct decision is made, a code can be placed on a port that can be checked by the MSP430 software. Then, the active mode is entered only if this code is present.

The start-up frequency of the DCO is approximately 500 kHz and may last up to 4 seconds until a stable MCLK frequency is reached. To enter the LPM4 the following code is necessary:

```
; Enter LPM4, enable GIE
;
      BIS    #CPUOFF+OSCOFF+GIE+SCG1+SCG0,SR
```

The exit from LPM4 is principally the same as described for LPM3. Interrupt handler software has to determine if the CPU stays active or if a return to a low-power mode is necessary.

When entering the LPM4 the information in control registers SCFI0 and SCFI1 of the system clock frequency integrator (SCFI) remains stored. If at this time the ambient temperature is high, SCFI1 contains a relatively high value to compensate the negative temperature coefficient of the DCO. If the LPM4 is later exited and the ambient temperature is very low, it is possible that the resulting DCO frequency, based on the value in SCFI1, will be outside of the oscillator range. It is therefore a good programming practice to set the SCFI control register to a low value before entering LPM4.

```
; Enter LPM4, enable GIE
;
   CLRC                                    ; Ensure that new MSB is 0
   RRC    &SCFI1                           ; Use halved tap number
   BIS    #CPUOFF+OSCOFF+GIE+SCG1+SCG0,SR  ; Enter LPM4
```

Note:

The two 8-bit counters of the universal timer/port may also be used during LPM4. If a counter is incremented by an external signal (inputs CIN, CMP, or TPIN.5) from 0FFh to 0h, then the appropriate RCxFG-flag is set. If interrupt is enabled, the CPU wakes up.