LAB 8
LOW POWER MODE

Engr. Shahzada Fahim Jan

Operating Modes

The MSP430 family is designed for low-power applications and uses the different operating modes

The operating modes take into account three different needs:

- Low power
- Speed and data throughput
- Minimizing current consumption of individual peripherals

Low-power modes LPM0 through LPM4 are configured with the CPUOFF, OSCOFF, SCG0, and SCG1 bits in the SR.

The advantage of including the CPUOFF, OSCOFF, SCG0, and SCG1 mode-control bits in the SR is that the present operating mode is saved onto the stack during an interrupt service routine.

Program flow returns to the previous operating mode if the saved SR value is not altered during the interrupt service routine.

Program flow can be returned to a different operating mode by manipulating the saved SR value on the stack inside of the interrupt service routine.

When setting any of the mode-control bits, the selected operating mode takes effect immediately.

Peripherals operating with any disabled clock are disabled until the clock becomes active.

Peripherals may also be disabled with their individual control register settings.

All I/O port pins, RAM, and registers are unchanged.

Wake-up from LPM0 through LPM4 is possible through all enabled interrupts.

When LPMx.5 (LPM3.5 or LPM4.5) is entered, the voltage regulator of the Power Management Module (PMM) is disabled.
All RAM and register contents are lost. Although the I/O register contents are lost, the I/O pin states are locked upon LPMx.5 entry.
Wake-up from LPM4.5 is possible from a power sequence, a RST event, or from specific I/O.
Wake-up from LPM3.5 is possible from a power sequence, a RST event, an RTC event, an LF crystal fault, or from specific I/O.

Table 1-2. Operation Modes

SCG1[1]	SCG0	OSCOFF[1]	CPUOFF[1]	Mode	CPU and Clocks Status[2]
0	0	0	0	Active	CPU, MCLK are active.
					ACLK is active. SMCLK optionally active (SMCLKOFF = 0).
					DCO is enabled if sources ACLK, MCLK, or SMCLK (SMCLKOFF = 0).
					DCO bias is enabled if DCO is enabled or DCO sources MCLK or SMCLK (SMCLKOFF = 0).
					FLL is enabled if DCO is enabled.
0	0	0	1	LPM0	CPU, MCLK are disabled.
					ACLK is active. SMCLK optionally active (SMCLKOFF = 0).
					DCO is enabled if sources ACLK or SMCLK (SMCLKOFF = 0).
					DCO bias is enabled if DCO is enabled or DCO sources MCLK or SMCLK (SMCLKOFF = 0).
					FLL is enabled if DCO is enabled.
1	0	0	1	LPM2 (device specific)	CPU, MCLK, and FLL are disabled.
					ACLK is active. SMCLK is disabled.
					FLL is disabled.
1	1	0	1	LPM3	CPU, MCLK, and FLL are disabled.
					ACLK is active. SMCLK is disabled.
					FLL is disabled.
1	1	1	1	LPM4	CPU and all clocks are disabled.
1	1	1	1	LPM3.5	When PMMREGOFF = 1, regulator is disabled. RAM retention in backup memory. In this mode, RTC and LCD operation is possible when configured properly. See the RTC and LCD modules for further details.
1	1	1	1	LPM4.5	When PMMREGOFF = 1, regulator is disabled. No memory retention. In this mode, all clock sources are disabled; that is, no RTC operation is possible.

(1) LPMx.5 modes are entered by following the correct entry sequence as defined in Section 1.4.2.

(2) The system clocks and the low-power modes can be affected by the clock request system. See the CS chapter for details.

Low-Power Modes and Clock Requests

A peripheral module request its clock sources automatically from the clock system (CS) module if it is required for its proper operation, regardless of the current power mode of operation.

Because of the clock request mechanism the system might not reach the low-power modes requested by

the bits set in the CPU status register, SR.

Table 1-3. Requested vs Actual LPM

Requested (SR Bits According to Table 1-2	Actual LPM ...		
	If No Clock Requested	If Only ACLK Requested	If SMCLK Requested
LPM0	LPM0	LPM0	LPM0
LPM2 (device specific)	LPM2	LPM2	LPM0
LPM3	LPM3	LPM3	LPM0
LPM4	LPM4	LPM3	LPM0

Entering and Exiting Low-Power Modes LPM0 Through LPM4

An enabled interrupt event wakes the device from low-power operating modes LPM0 through LPM4.

The program flow for exiting LPM0 through LPM4 is:

• Enter interrupt service routine

– The PC and SR are stored on the stack.

– The CPUOFF, SCG1, and OSCOFF bits are automatically reset.

• Options for returning from the interrupt service routine

– The original SR is popped from the stack, restoring the previous operating mode.

– The SR bits stored on the stack can be modified within the interrupt service routine to return to a different operating mode when the RETI instruction is executed.

Low-Power Modes LPM3.5 and LPM4.5 (LPMx.5)

The low-power modes LPM3.5 and LPM4.5 (LPMx.5) give the lowest power consumption on a device.

In LPMx.5, the core LDO of the device is switched off. This has the following effects:

• Most of the modules are powered down.

– In LPM3.5, only modules powered by the RTC LDO continue to operate. At least an RTC module is connected to the RTC LDO.

– In LPM4.5 the RTC LDO and the connected modules are switched off.

• The register content of all modules and the CPU is lost.

• The SRAM content is lost.

Enter LPMx.5

1. Store any information that must be available after wakeup from LPMx.5 in FRAM.
2. For LPM4.5 set all ports to general-purpose I/Os (PxSEL0 = 00h and PxSEL1 = 00h).

For LPM3.5 if the LF crystal oscillator is used do not change the settings for the I/Os shared with the LF-crystal-oscillator. These pins must be configured as LFXIN and LFXOUT. Set all other port pins to general-purpose I/Os with PxSEL0 and PxSEL1 bits equal to 0.

3. Set the port pin direction and output bits as necessary for the application.

5. For LPM3.5, the modules that stay active must be enabled. For example, the RTC must be enabled if necessary. Only modules connected to the RTC LDO can stay active.
6 Disable the watchdog timer WDT if it is enabled and in watchdog mode. If the WDT is enabled and in
watchdog mode, the device does not enter LPMx.5.
7. Clear the GIE bit:
BIC #GIE, SR

Exit From LPMx.5

The following conditions cause an exit from LPMx.5:

• A wake-up event on an I/O, if configured and enabled. The interrupt flag of the corresponding port pin is set (PxIFG).

• A wake-up event from the RTC, if enabled. The corresponding interrupt flag in the RTC is set

• A wake-up signal from the RST pin.

• A power cycle.

Wake up From LPM3.5

Do the following steps after a wakeup from LPM3.5:

1. Initialize the registers of the modules connected to the RTC LDO exactly the same way as they were configured before the device entered LPM3.5 but do not enable the interrupts.
2. Initialize the port registers exactly the same way as they were configured before the device entered LPM3.5 but do not enable port interrupts.
3. If the LF-crystal-oscillator was used in LPM3.5 the corresponding I/Os must be configured as LFXIN and LFXOUT. The LF-crystal-oscillator must be enabled in the clock system
4. Clear the LOCKLPM5 bit in the PM5CTL0 register.
5. Enable port interrupts as necessary.
6. Enable module interrupts.
7. After enabling the port and module interrupts, the wake-up interrupt is serviced as a normal interrupt.

Wake up from LPM4.5

Do the following steps after a wakeup from LPM4.5:

1. Initialize the port registers exactly the same way as they were configured before the device entered LPM4.5 but do not enable port interrupts.
2. Clear the LOCKLPM5 bit in the PM5CTL0 register.
3. Enable port interrupts as necessary.
4. After enabling the port interrupts, the wake-up interrupt is serviced as a normal interrupt.

If a crystal oscillator is needed after a wakeup from LPM4.5 then configure the corresponding pins and start the oscillator after you cleared the LOCKLPM5 bit.

```c
#include <msp430.h>

int main(void)
{
    WDTCTL = WDT_ADLY_1000;                   // WDT 1000ms, ACLK, interval timer

    SFRIE1 |= WDTIE;                          // Enable WDT interrupt

    P4SEL0 |= BIT2 + BIT1;                    // P4.2: XOUT; P4.1: XI1

    CSCTL4 = SELMS__DCOCLKDIV + SELA__XT1CLK;  // MCLK=SMCLK=DCO; ACLK=XT1

    // Port Configuration all un-used pins to output low
    P1OUT = 0x00;
    P2OUT = 0x00;
    P3OUT = 0x00;
    P4OUT = 0x00;
    P5OUT = 0x00;
    P6OUT = 0x00;
    P7OUT = 0x00;
    P8OUT = 0x00;

    P1DIR = 0xff;
    P2DIR = 0xff;
    P3DIR = 0xff;
    P4DIR = 0xff;
    P5DIR = 0xff;
```

```
P6OUT = 0x00;
P7OUT = 0x00;
P8OUT = 0x00;

P1DIR = 0xff;
P2DIR = 0xff;
P3DIR = 0xff;
P4DIR = 0xff;
P5DIR = 0xff;
P6DIR = 0xff;
P7DIR = 0xff;
P8DIR = 0x0f;

// Disable the GPIO power-on default high-impedance mode
// to activate previously configured port settings
PM5CTL0 &= ~LOCKLPM5;

do
{
    CSCTL7 &= ~(XT1OFFG | DCOFFG);      // Clear XT1 and DCO fault flag
    SFRIFG1 &= ~OFIFG;
}while (SFRIFG1 & OFIFG);               // Test oscillator fault flag
```

```
    __bis_SR_register(LPM3_bits | GIE);     // Enter LPM3
    __no_operation();                       // For debugger
}

// Watchdog Timer interrupt service routine
#if defined(__TI_COMPILER_VERSION__) || defined(__IAR_SYSTEMS_ICC__)
#pragma vector=WDT_VECTOR
__interrupt void WDT_ISR(void)
#elif defined(__GNUC__)
void __attribute__ ((interrupt(WDT_VECTOR))) WDT_ISR (void)
#else
#error Compiler not supported!
#endif
{
    P1OUT ^= 0x01;                          // Toggle P1.0 (LED) every 1s
}
```

TASK

Task 1

Configure Watchdog Timer (WDT) for 1-second interval.
Enter LPM3.

Task 2

Use Timer_A (instead of WDT) to generate an interrupt every 500 ms.
Enter LPM3.
Toggle P1.1 LED every 500 ms when Timer_A interrupt fires.

Task 3

- **Configure a push-button on P1.2 as wake-up pin.**
- **Enter LPM4.5.**
- **When the button is pressed, MCU resets and blinks P1.0 continuously.**

(You must configure the wake-up interrupt before entering LPM4.5.)

Task 4

Setup a timer or RTC (Real Time Clock) to wake device after 10 seconds.
Enter LPM4.5.
When it wakes up, blink P1.0 LED three times.

Task 5

- **Configure a Timer_A to overflow every 2 seconds.**
- **Enter LPM3.5.**
- **On wakeup, toggle an LED and go back to sleep automatically.**