

MSP430 Architecture

Uday Gurnani

Senior, Electrical Engineering
Embedded & Adaptive Computing Group
The University of Texas at Dallas



Presentation Outline

MSP430 Architecture Block Diagram

MSP430 Architecture Basics

CPU Architecture

Memory

Resets, Interrupts, Operating Modes

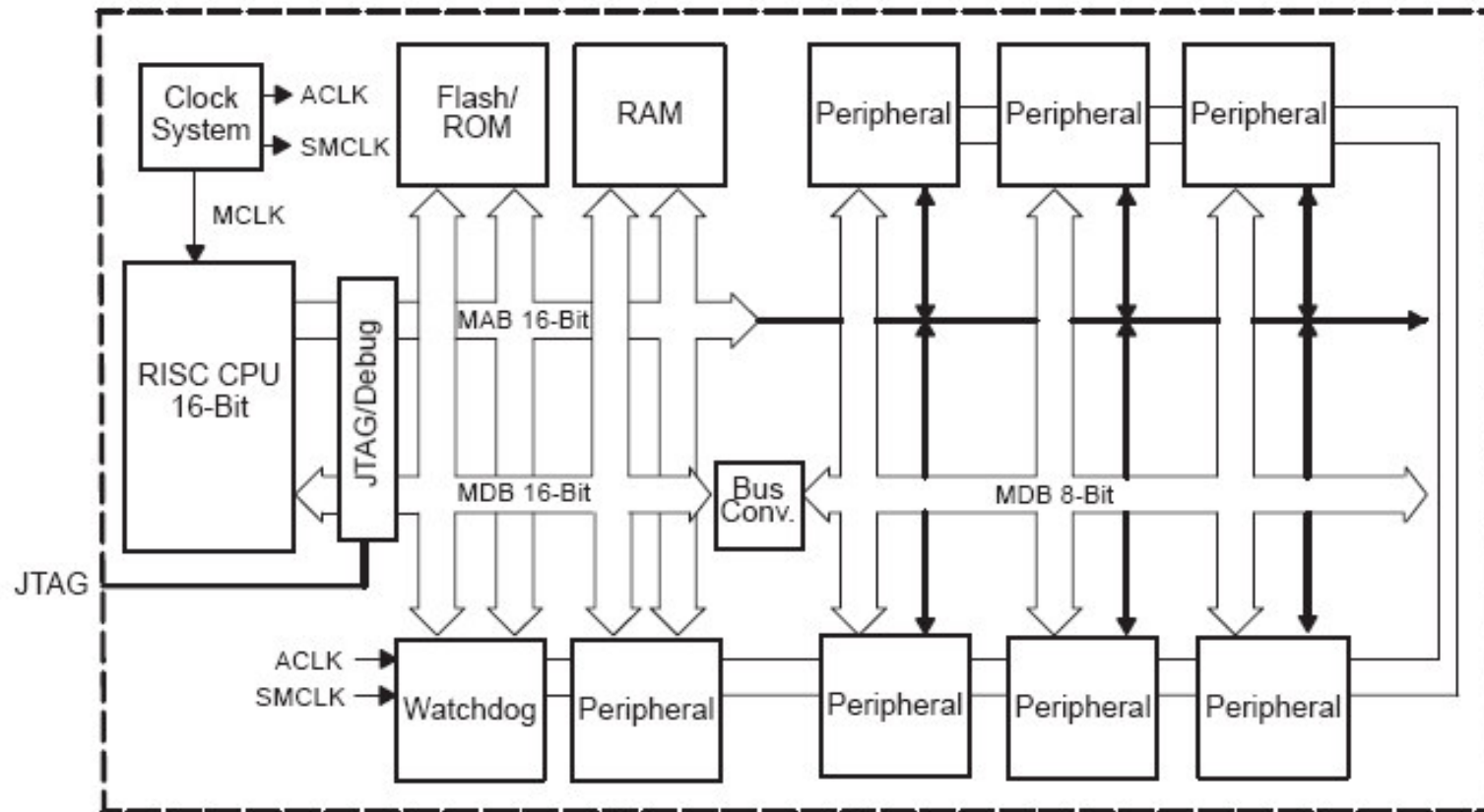
Digital I/O Ports

Clocks

Timers

Flash LED Example

Architecture Block Diagram



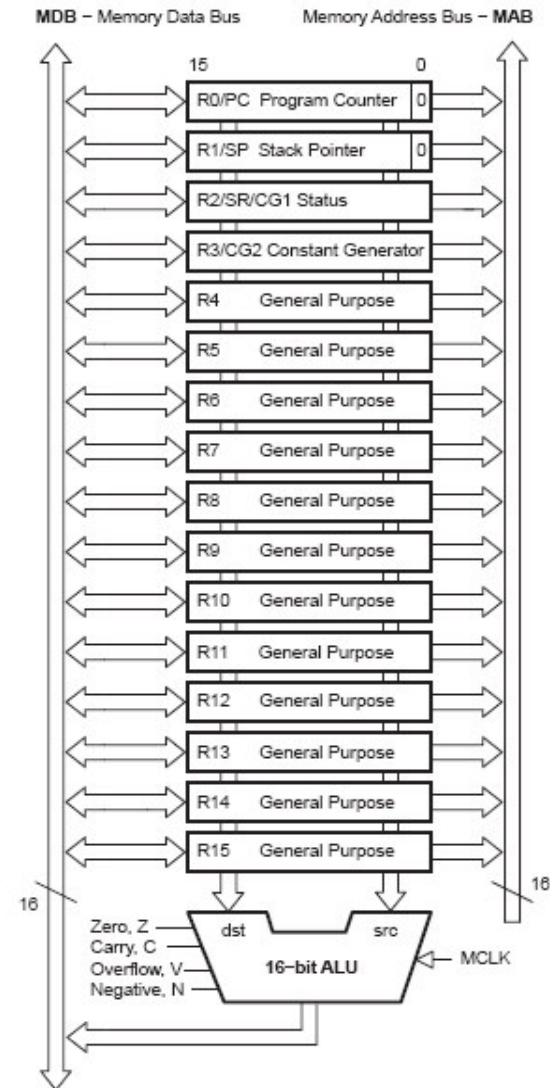
Architecture Basics

- Ultra-low power architecture
- 1.8 - 3.6 V operation
- 6 μ s wakeup from standby mode
- 27 core instructions, 24 emulated instructions
- 16-bit RISC CPU
- 16-bit ALU
- Common 16-bit MAB, MDB – von Neumann
- 16 registers
 - 4 Special Function Registers: PC, SP, SR, CG
- 7 addressing modes

CPU Architecture

CPU Registers

- **PC/R0: Program Counter**
 - Stores address of next instruction to be executed
- **SP/R1: Stack Pointer**
 - Stores return address of subroutine calls and interrupts
- **SR/R2: Status Register**
 - Flag bits indicate status of mathematical operations
- **CG1, CG2/ R2, R3: Constant Generator**
 - Generates 6 commonly used constants
 - Provides 24 additional emulated instructions
 - Example: CLR dst => MOV R3, dst
R3 with As=0, holds 00000h
- **General Purpose Registers**
 - Used to hold data, addresses, or index values
 - Can be accessed with byte or word instructions



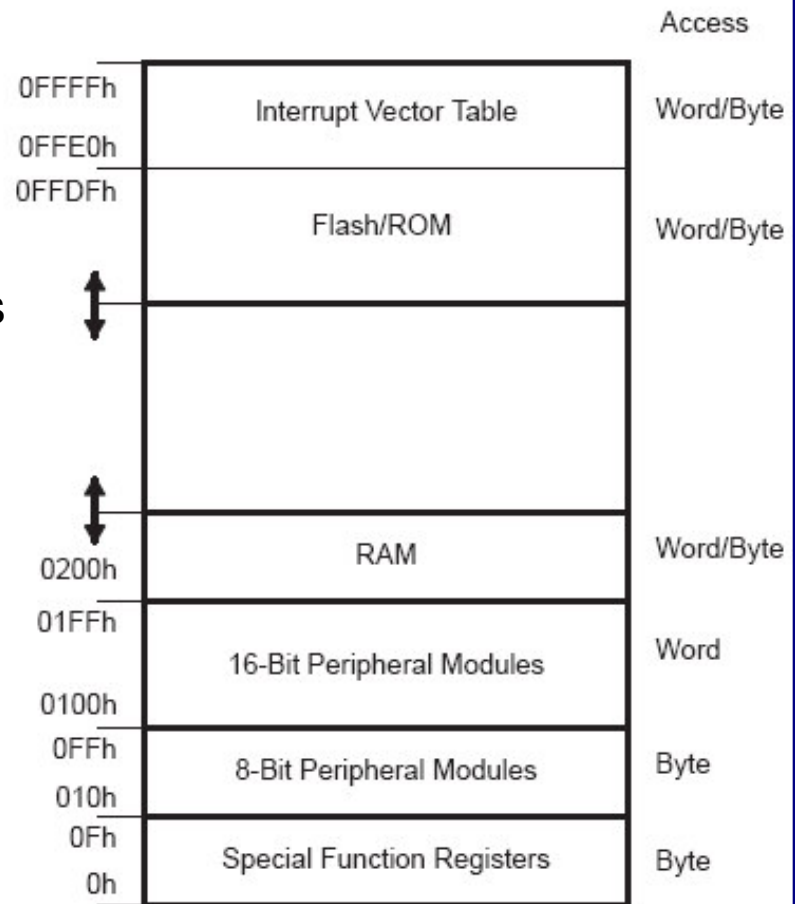
CPU Architecture

Seven Addressing Modes

- **Register Mode:** MOV R10, R11
- **Indexed Mode:** MOV 2(R5), 6(R6)
- **Symbolic Mode:** MOV add1, add2 => MOV X(PC), Y(PC)
where $X = \text{add1} - \text{PC}$; $Y = \text{add2} - \text{PC}$
- **Absolute Mode:** MOV &add1, &add2 => MOV X(0), Y(0)
where $X = \text{add1}$, $Y = \text{add2}$
- **Indirect Register Mode:** MOV @R10, 0(R11)
- **Indirect Auto-increment Mode:** MOV @R10+, 0(R11)
+ increments R10 by 1 for byte op.; 2 for word op.
- **Immediate Mode:** MOV #45h, add

Memory

- 64 KB of addressable memory spaces
- 0FFE0h to 0FFFFh = interrupt vector table
- 0200h to 0FFDFh = Data and programs
 - Low byte at even address
 - High byte at *next* odd address
- 0100h to 01FFh = 16-bit peripheral modules
- 0010h to 00FFh = 8-bit peripheral modules
- 0h to 0Fh = SP, PC, SR, CG



Resets, Interrupts, Operating Modes

System Resets

1. Power-on reset (POR)

- Triggered when powering up device
- I/O pins are switched to input mode
- Status Register (SR) is reset
- Watchdog timer = active in watchdog mode
- Peripheral modules, registers go to default states
- PC loaded with address at reset vector location OFFFEh
- CPU begins code execution from address in PC

2. Power-up clear (PUC)

- Triggered when:
- POR is encountered
 - Watchdog timer expiration, password violation
 - Flash memory security key violation

Resets, Interrupts, Operating Modes

Interrupts

1. Maskable

- Trigger event is not always important
- Programmer has control

2. Non-Maskable

- Highest priority and are never ignored
- Interrupt flag (IFG) set = triggers interrupt
- Example: Reset Interrupt - Jumps to beginning of memory & executes instructions there

Resets, Interrupts, Operating Modes

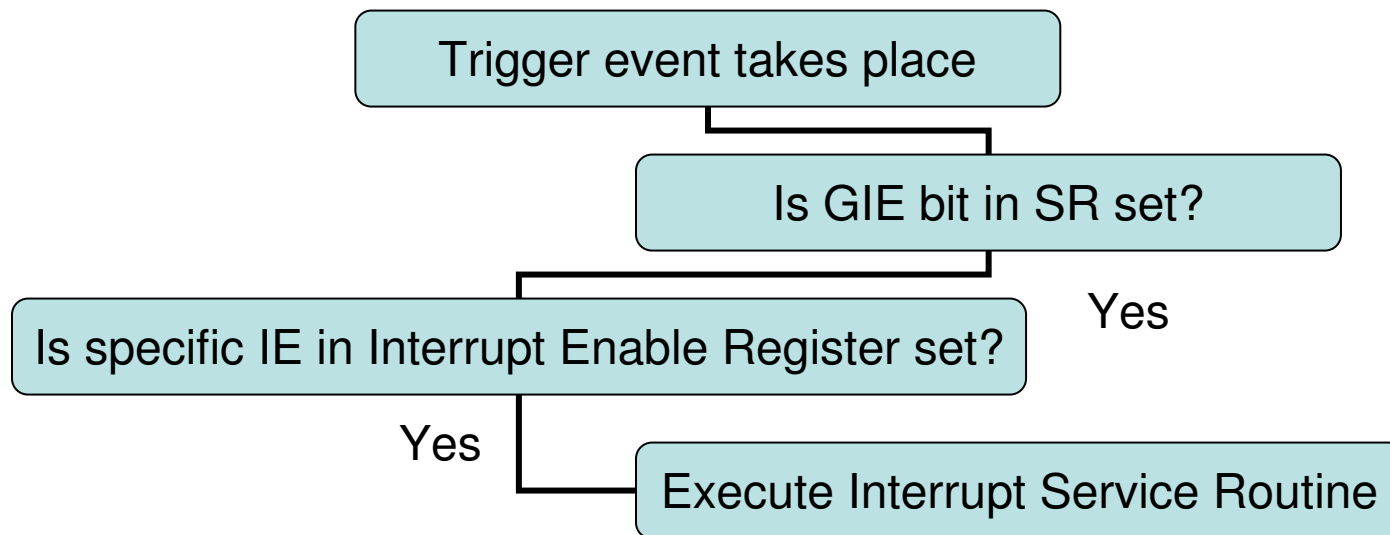
Interrupt Priority

- 8 - non-maskable : External Reset, Power-up, Watchdog Timer Reset, Flash Key Violation, NMI
- 7 - non-maskable : Oscillator Fault, Flash Memory Access Violation
- 6 - maskable : Watchdog Timer
- 5 - maskable : Timer A Capture Compare Register 0 (CCR0) Interrupt
- 4 - maskable : Timer A Capture Compare Register 1 (CCR1) Interrupt
- 3 - maskable : Sigma/Delta 16 bit (SD16) Converter Interrupt
- 2 - maskable : Universal Serial Interface (USI) Interrupts
- 1 - maskable : Port 2 I/O Interrupts
- 0 - maskable : Port 1 I/O Interrupts

Source:<http://cnx.org/content/m13650/latest/>

Resets, Interrupts, Operating Modes

Interrupt Process



Resets, Interrupts, Operating Modes

Operating Modes

1. Active Mode (AM)

- Fully-powered mode
- Processor executes instructions; peripherals, clocks are active
- 340 μ A with 1MHz at 3.3 V

2. Low Power Mode 0 (LPM0)

3. Low Power Mode 1 (LPM1)

4. Low Power Mode 2 (LPM2)

5. Low Power Mode 3 (LPM3)

6. Low Power Mode 4 (LPM4)

Resets, Interrupts, Operating Modes

Operating Modes

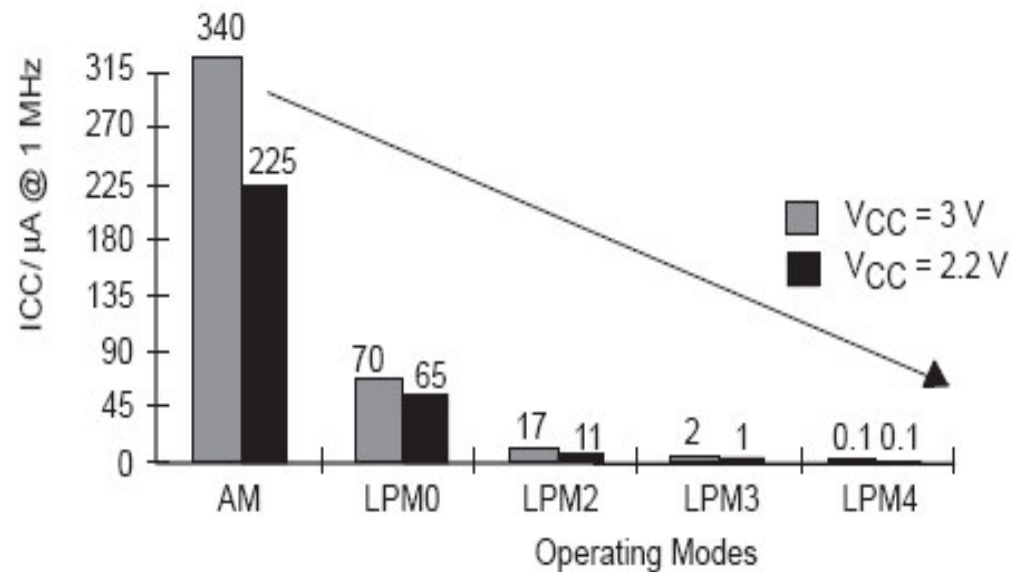
Mode	CPU	MCLK	SMCLK	ACLK	DCO	Current at 1MHz, 3.3 V
AM	ON	ON	ON	ON	ON	340 μ A
LPM0	OFF	OFF	ON	ON	ON	70 μ A
LPM1	OFF	OFF	ON	ON	OFF	-
LPM2*	OFF	OFF	OFF	ON	OFF	17 μ A
LPM3*	OFF	OFF	OFF	ON	OFF	2 μ A
LPM4	OFF	OFF	OFF	OFF	OFF	0.1 μ A

*For LPM2, SCG0=0 & SCG1=1 of the SR, For LPM3, SCG0=1 & SCG1=1

Resets, Interrupts, Operating Modes

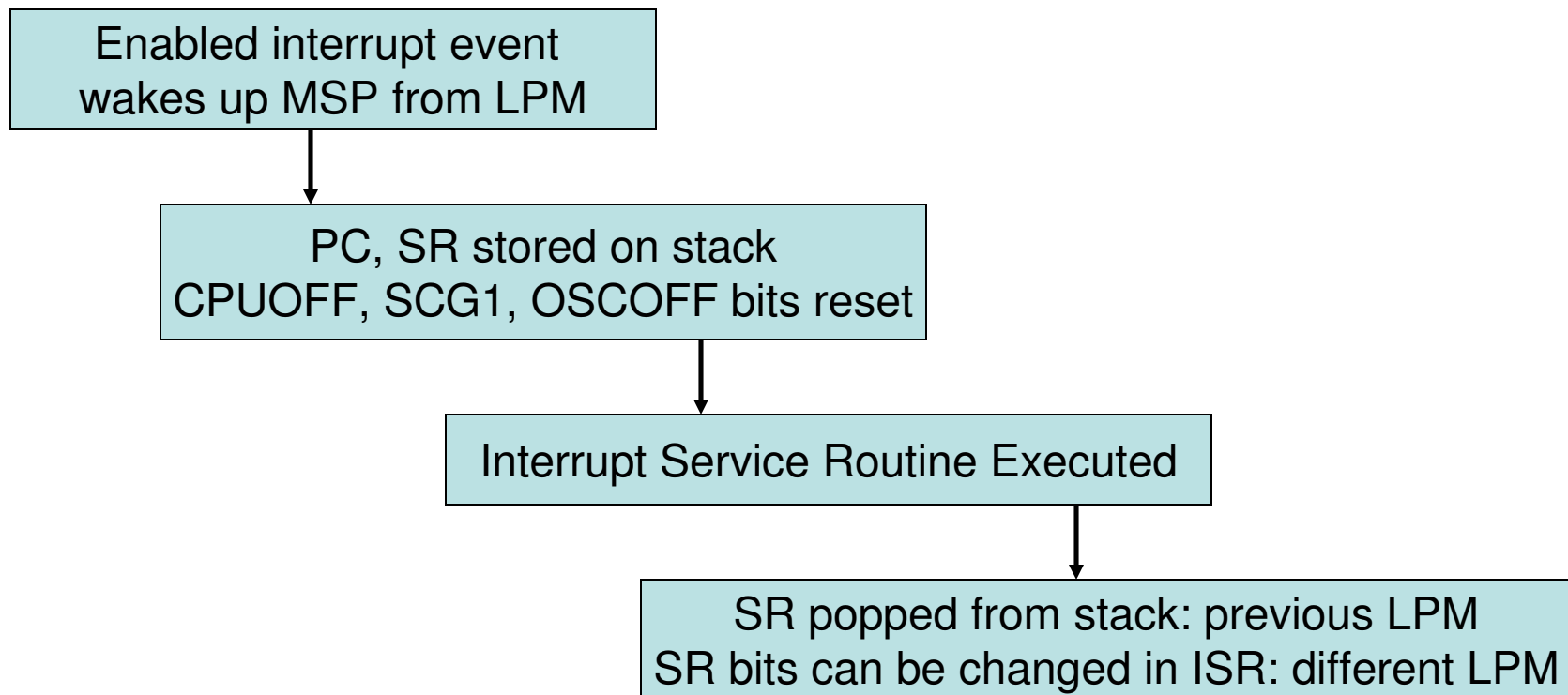
Power Considerations - Operating Modes

Figure 2-9. Typical Current Consumption of 13x and 14x Devices vs Operating Modes



Resets, Interrupts, Operating Modes

Operating Mode Switching



Resets, Interrupts, Operating Modes

Operating Mode Switching

1.LPM0 Example

```
BIS #GIE+CPUOFF,SR ; Enter LPM0  
; .....Perform ISR instructions  
BIC #CPUOFF,0(SP) ; Exit LPM0 on RETI  
RETI
```

2. LPM3 Example

```
BIS #GIE+CPUOFF+SCG1+SCG0,SR ; Enter LPM3  
; ... ; ISR instructions  
BIC #CPUOFF+SCG1+SCG0,0(SP) ; Exit LPM3 on RETI  
RETI
```

Digital I/O Ports

- 6 digital I/O ports, P1-P6, each with 8 I/O pins
- Pin: input or output configurable, read or written to
- Interrupt capability on ports P1, P2
- Digital I/O is configured with user software
- **PxIN register**
Each bit = value of I/P signal at corresponding pin
- **PxOUT register**
Each bit = value to be O/P on corresponding pin
- **PxDIR register**
Bit = 0: Input direction; Bit = 1: Output direction
- **PxSEL register**
Bit = 0: I/O function; Bit = 1: Peripheral module

Clocks

Clock Sources

1. Low Frequency Crystal Clock (LFXTCLK)

External Crystal, ~32kHz

2. Digitally Controlled Oscillator Clock (DCOCLK)

Internally generated, default at 1MHz, 2us startup

Lower frequency achievable using RSELx, MODx, DCOx bits

Clock Lines

1. Master Clock (MCLK)

Drives MSP CPU core, vital for processor

Source selection = SELMx bits, Divider control = DIVMx bits (of BCSCTL2)

2. Submaster Clock (SMCLK)

Drives most peripherals

Source = DCO/Crystal 2; controlled using SELS, SCG bits of BCSCTL2, SR

Divider control by DIVSx bits of BCSCTL2

3. Auxiliary Clock (ACLK)

Source = LFXTCLK; Typically used for wake-up from standby mode

Division control by DIVAx bits of BCSCTL1 register

Clocks

Clock Control Registers

1. DCOCTL: Digitally-Controlled Oscillator Clock-Frequency Control

	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

2. BCSCTL1: Basic Clock System Control 1

	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

3. BCSCTL2: Basic Clock System Control 2

	7							0
BCSCTL2	SELM.1	SELM.0	DIVM.1	DIVM.0	SELS	DIVS.1	DIVS.0	DCOR
058h								

Source: http://www.eng.uah.edu/~jovanov/CPE621/notes/msp430_clock.pdf

Timers

1. Timer_A

- TAR – 16-bit timer/counter: increments/decrements register with clock edge
- TACLRL = 1 clears TAR, clock divider, count direction
- Timer clock: Sourced from ACLK, SMCLK or externally using TASSELx bits
- IDx bits divide clock signal by 2, 4, or 8
- Has 4 operating modes (discussed on next slide)

2. Timer_B

- Similar to Timer_A with a few operating differences
- Length is programmable to 8, 10, 12 or 16 bits

3. Watchdog Timer

- 16-bit counter; Resets processor when code crashes (it rolls over to zero)
- Every program should reset Watchdog often to avoid processor reset
- Requires password (upper register part) = WDTPW in header file

Timers

Timer A Operating Modes

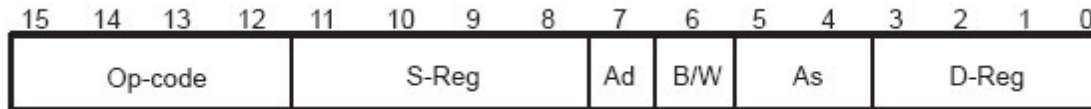
MCx	Mode	Description
00	Stop	The timer is halted.
01	Up	The timer repeatedly counts from zero to the value of TACCR0
10	Continuous	The timer repeatedly counts from zero to 0FFFFh.
11	Up/down	The timer repeatedly counts from zero up to the value of TACCR0 and back down to zero.

Timer A Registers

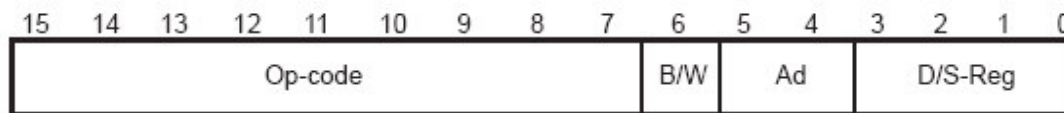
Register	Short Form	Register Type	Address	Initial State
Timer_A control	TACTL	Read/write	0160h	Reset with POR
Timer_A counter	TAR	Read/write	0170h	Reset with POR
Timer_A capture/compare control 0	TACCTL0	Read/write	0162h	Reset with POR
Timer_A capture/compare 0	TACCR0	Read/write	0172h	Reset with POR
Timer_A capture/compare control 1	TACCTL1	Read/write	0164h	Reset with POR
Timer_A capture/compare 1	TACCR1	Read/write	0174h	Reset with POR
Timer_A capture/compare control 2	TACCTL2	Read/write	0166h	Reset with POR
Timer_A capture/compare 2	TACCR2	Read/write	0176h	Reset with POR
Timer_A interrupt vector	TAIV	Read only	012Eh	Reset with POR

Instruction Set

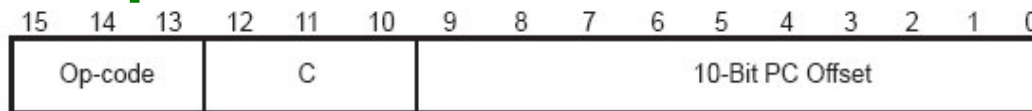
- **Dual-operand Instructions**



- **Single-operand Instructions**



- **Jump Instructions**



LED Flash Example

```
#include "msp430x20x1.h"
void main(void)
{ WDTCTL = WDTPW + WDTHOLD;          // Stop watchdog timer
  P1DIR |= 0x01;                     // Set P1.0 to output direction
  for (;;)
  {volatile unsigned int i;          // volatile to prevent optimization
   P1OUT ^= 0x01;                    // Toggle P1.0 using exclusive-OR
   i = 10000;                         // SW Delay
   do i--;
   while (i != 0);
  }
}
```

References

- Anand, Naren; *Introduction to Texas Instruments ez430*
Accessible at <http://cnx.org/content/col10354/1.6>
- Jovanov, Emil; *Advanced Embedded Systems CPE621 course slides*
Accessible at <http://www.eng.uah.edu/~jovanov/CPE621/CPE621.html>
- Luecke, Jerry; *Analog and Digital Circuits for Electronic Control System Applications*; Elsevier Inc., 2005
- Texas Instruments; *MSP430xxxx_Family_User's_Guide_(Rev._F).pdf*
Accessible at <http://focus.ti.com/mcu/docs/>