2.996/6.971 Biomedical Devices Design Laboratory

Lecture 5: Microprocessors I

Instructor: Dr. Hong Ma

Sept. 26, 2007

Analogy: A Complex Machine with Lots of Knobs

Courtesy of NASA.

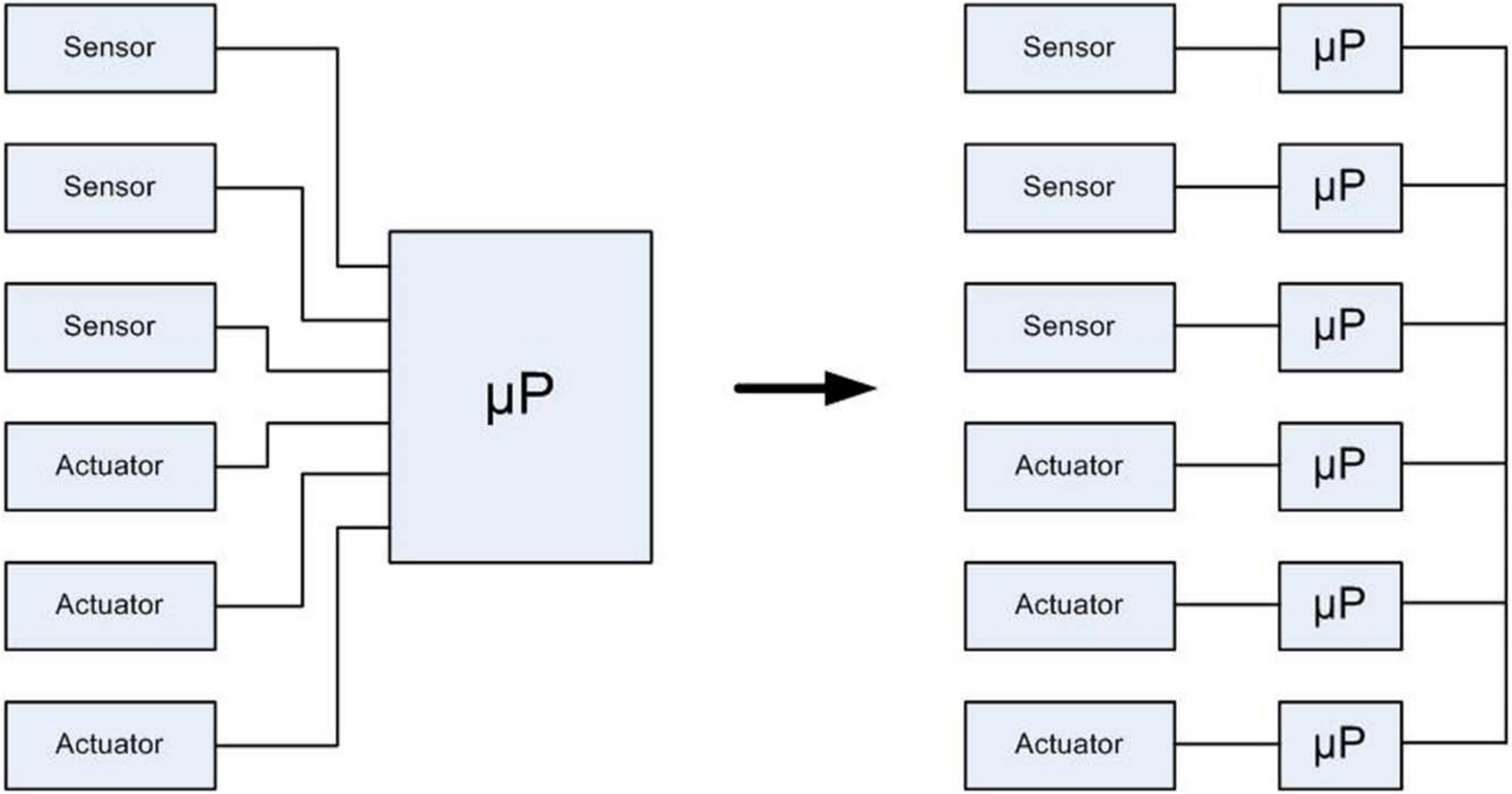
Microprocessor vs. PCs

- Microprocessors
  - Optimized to keep track of time
  - MSP430: 16MHz clock → 62.5ns timing
- PCs
  - Optimized to process large amounts of data
  - Windows: ~100Hz timing
  - Linux: ~1kHz timing

- **Timing accuracy can be leveraged for measurement functions**

Trends in Sensor Architectures

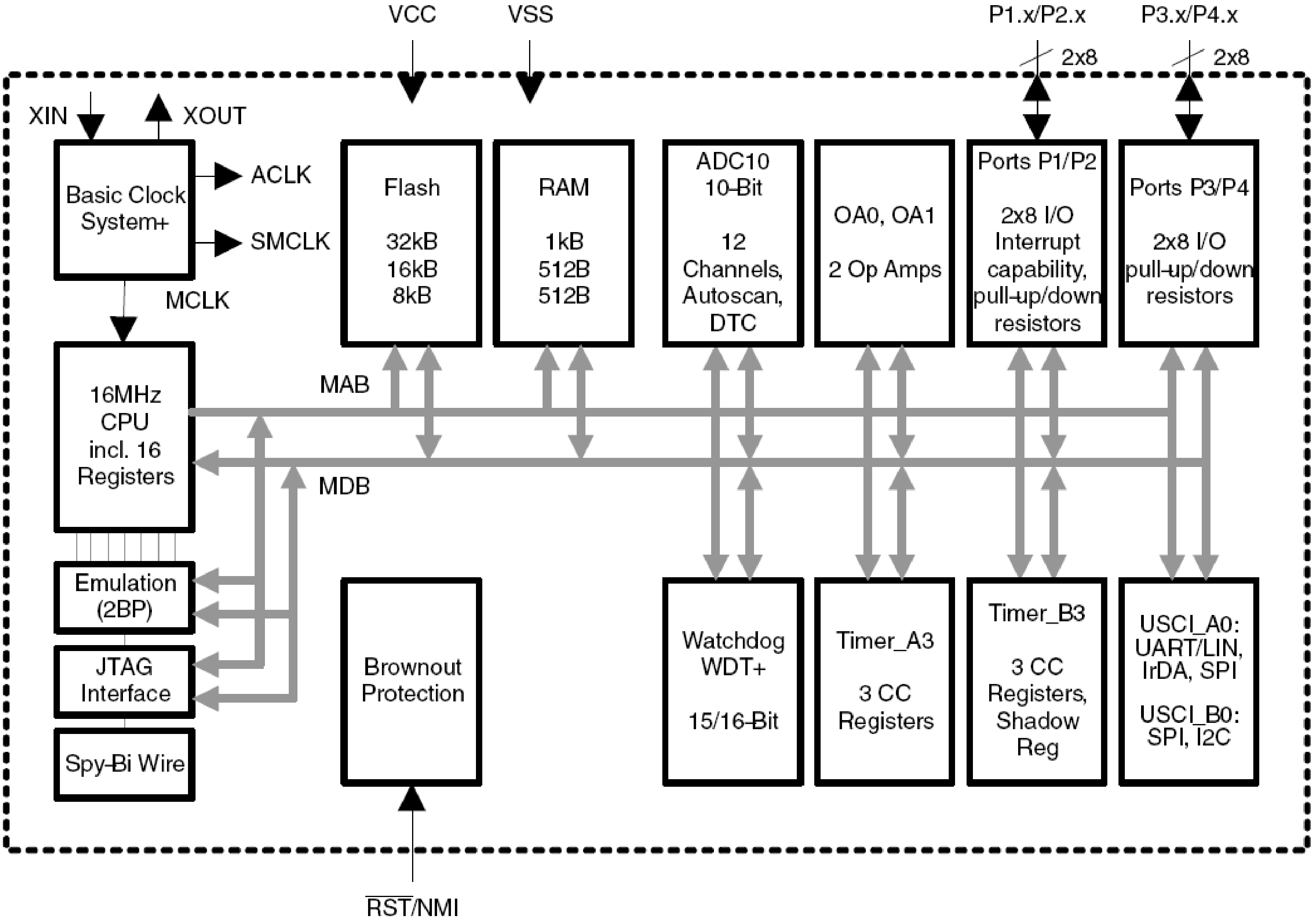
- Single processor → distributed processors



The MSP430F2xx Family

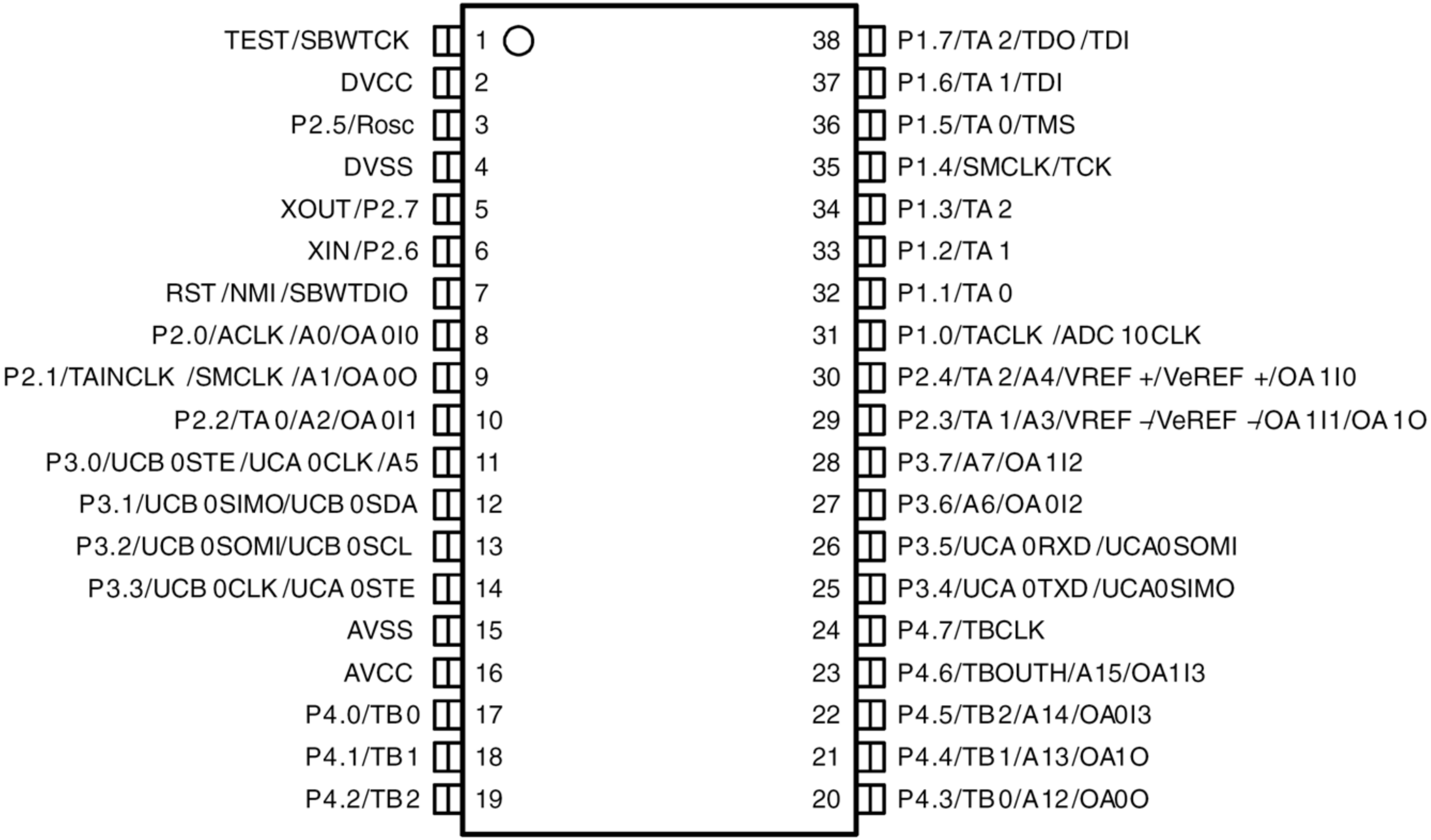
- Optimized for low-power and versatility
- Modern architecture, simple to program
- Many peripheral devices – designed to not require input from the CPU
- Unified address space, no paging
- Device emulates itself
- Inexpensive development tools
- Highly optimized code, designed for C compiler
- Low cost, price >$0.50

MSP430F2xx Architecture



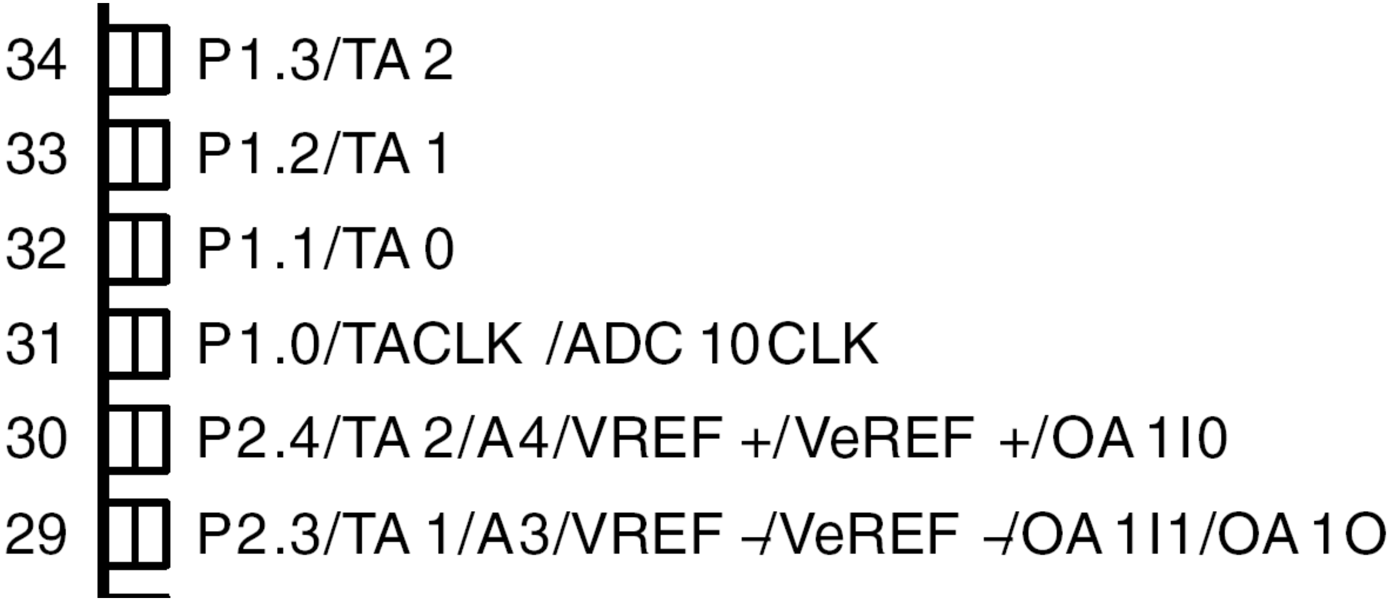
MSP430F2274 Pinout

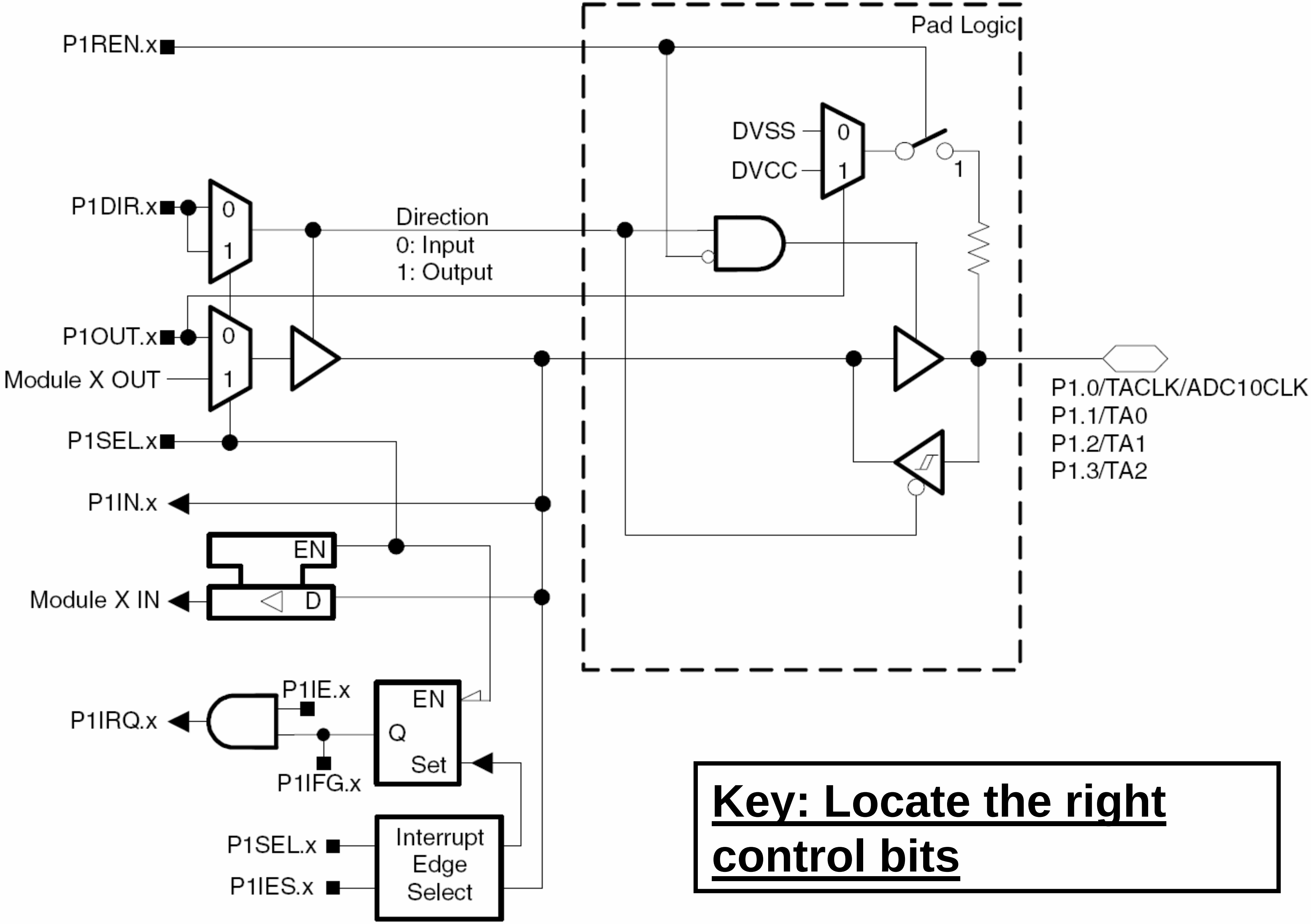
MSP430x22x4 device pinout, DA package



Port Functions

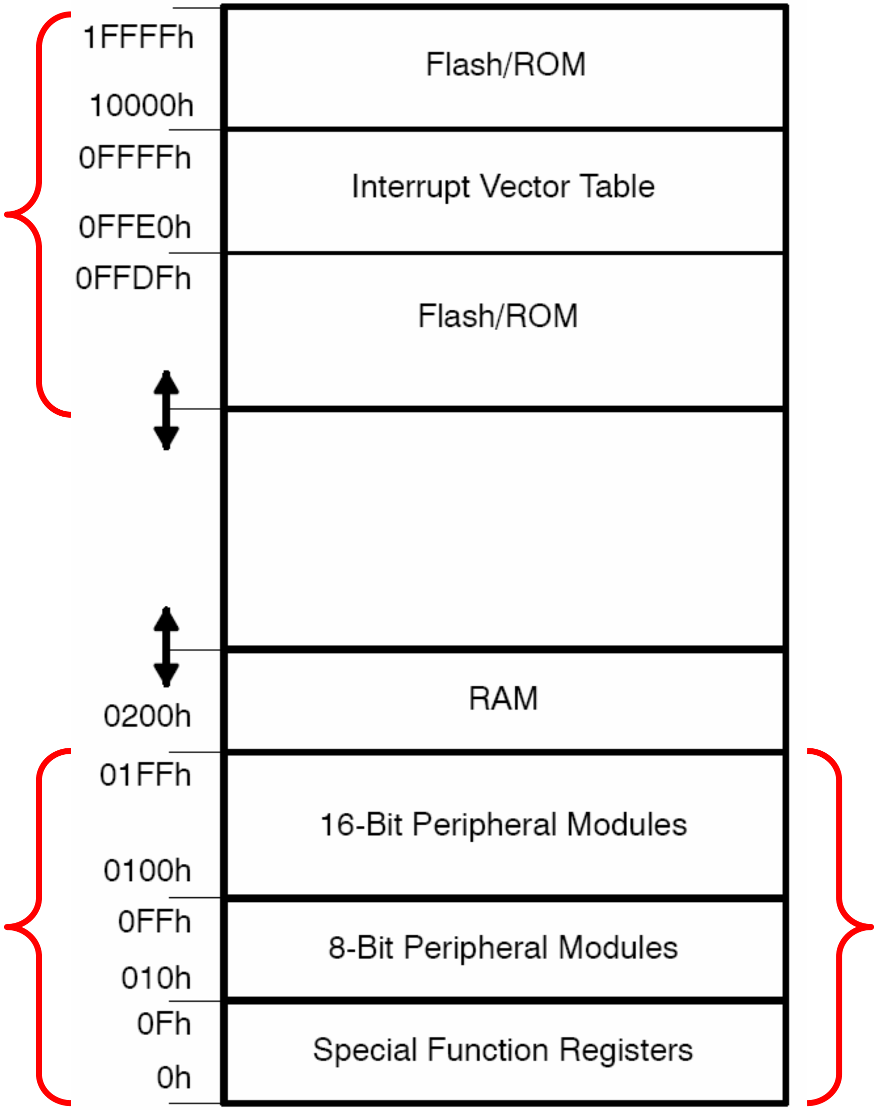
- Digital input
- Digital output
- Pulled-up / Pulled-down
- Peripheral input / output
- Interrupt on edge


34 P1.3/TA2
33 P1.2/TA1
32 P1.1/TA0
31 P1.0/TACLK/ADC10CLK
30 P2.4/TA2/A4/VREF+/VeREF+/OA1I0
29 P2.3/TA1/A3/VREF−/VeREF−/OA1I1/OA1O



Key: Locate the right control bits

Memory Map
Von Neuman Architecture



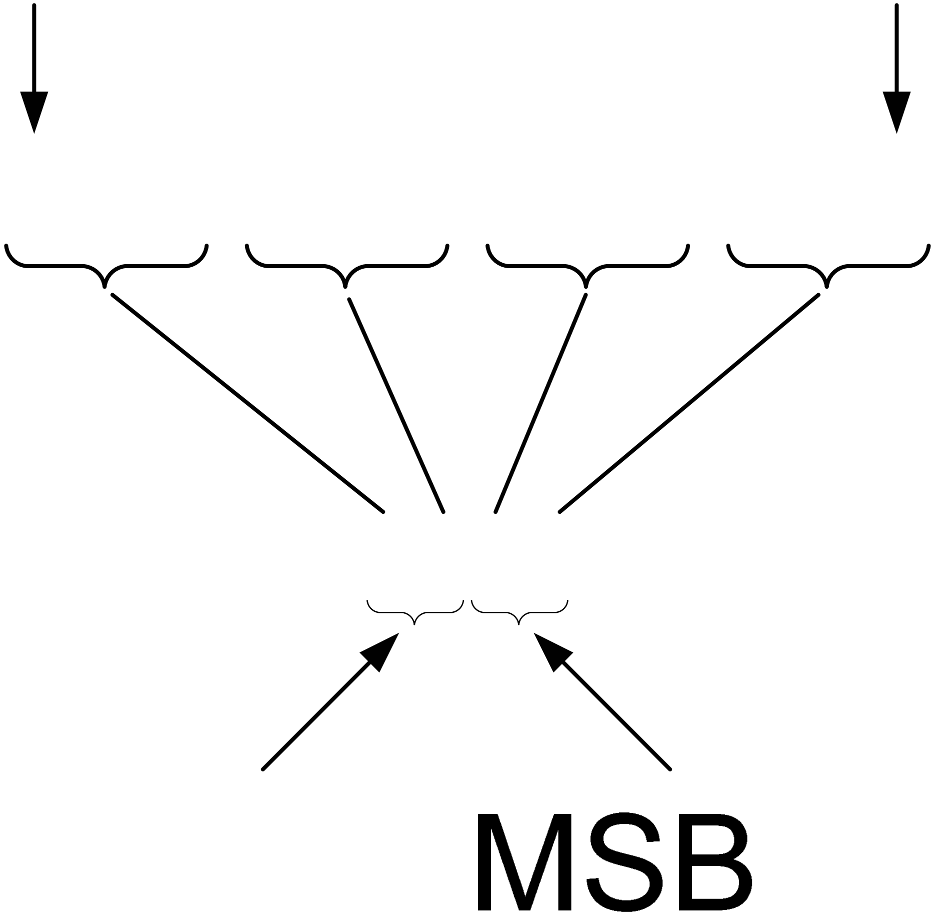
“Software” (Instructions for flipping the switches)

Hardware

(Switches and I/O)

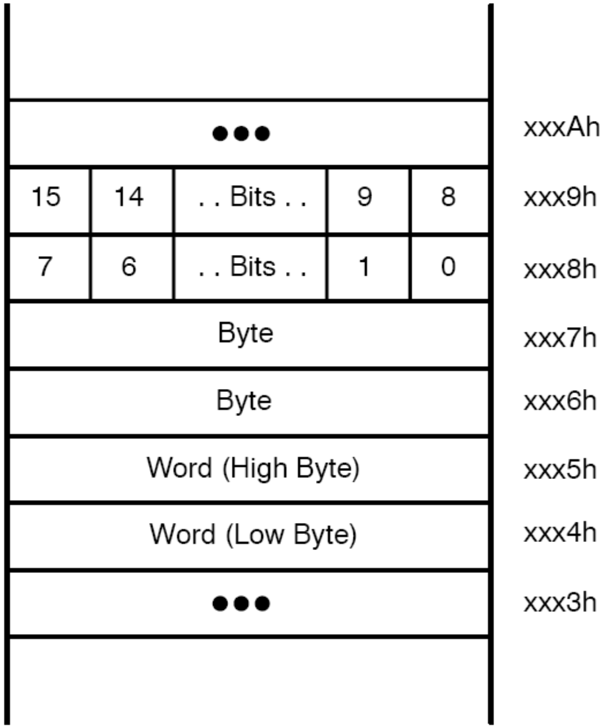
Registers

Hex Numbers and Memory



MSB

MSP430 Memory

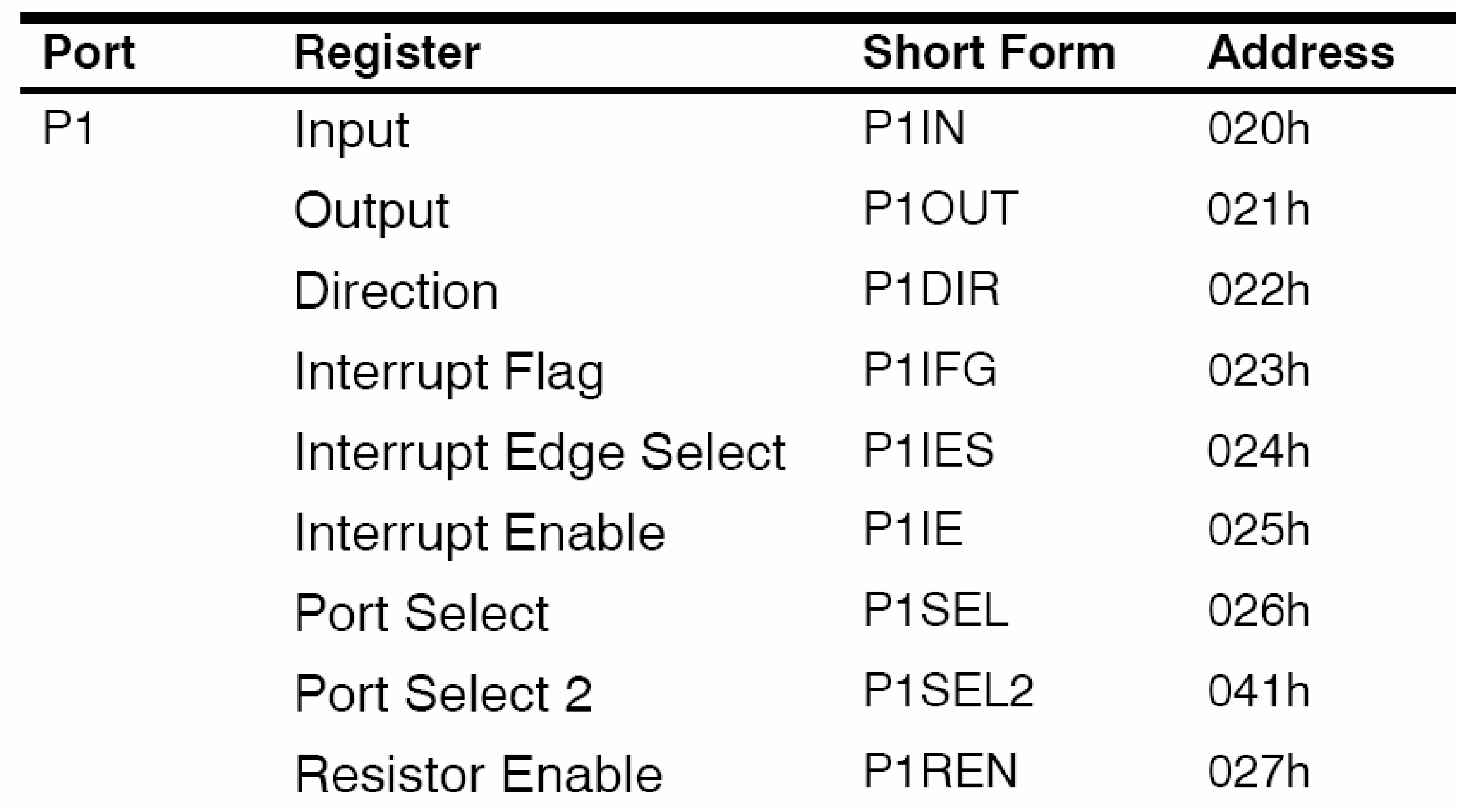


●●●					xxxAh
15	14	. . Bits . .	9	8	xxx9h
7	6	. . Bits . .	1	0	xxx8h
Byte					xxx7h
Byte					xxx6h
Word (High Byte)					xxx5h
Word (Low Byte)					xxx4h
●●●					xxx3h

- 8-bit addressing resolution

The Header File (msp430x22x4.h)

- Assigns aliases for registers
- Specific to each processor sub-group



Port	Register	Short Form	Address
P1	Input	P1IN	020h
	Output	P1OUT	021h
	Direction	P1DIR	022h
	Interrupt Flag	P1IFG	023h
	Interrupt Edge Select	P1IES	024h
	Interrupt Enable	P1IE	025h
	Port Select	P1SEL	026h
	Port Select 2	P1SEL2	041h
	Resistor Enable	P1REN	027h

Bit-wise Operators

- Bit-wise “OR”: |
  - 1000 | 0101 → 1101
- Bit-wise “AND”: &
  - 1001 & 0101 → 0001
- Bit-wise “NOT”: ~
  - ~1001 → 0110
- Bit-wise “XOR”: ^
  - 1001 ^ 0101 → 1100

Assigning Individual Bits

- Assigning all 8-bits at once
  - – P1OUT = 0xA7
- Assigning individual bits high
  - – P1OUT |= 0x81
- Assigning individual bits low
  - – P1OUT &= ~0x81
- Toggling individual bits
  - – P1OUT ^= 0x81

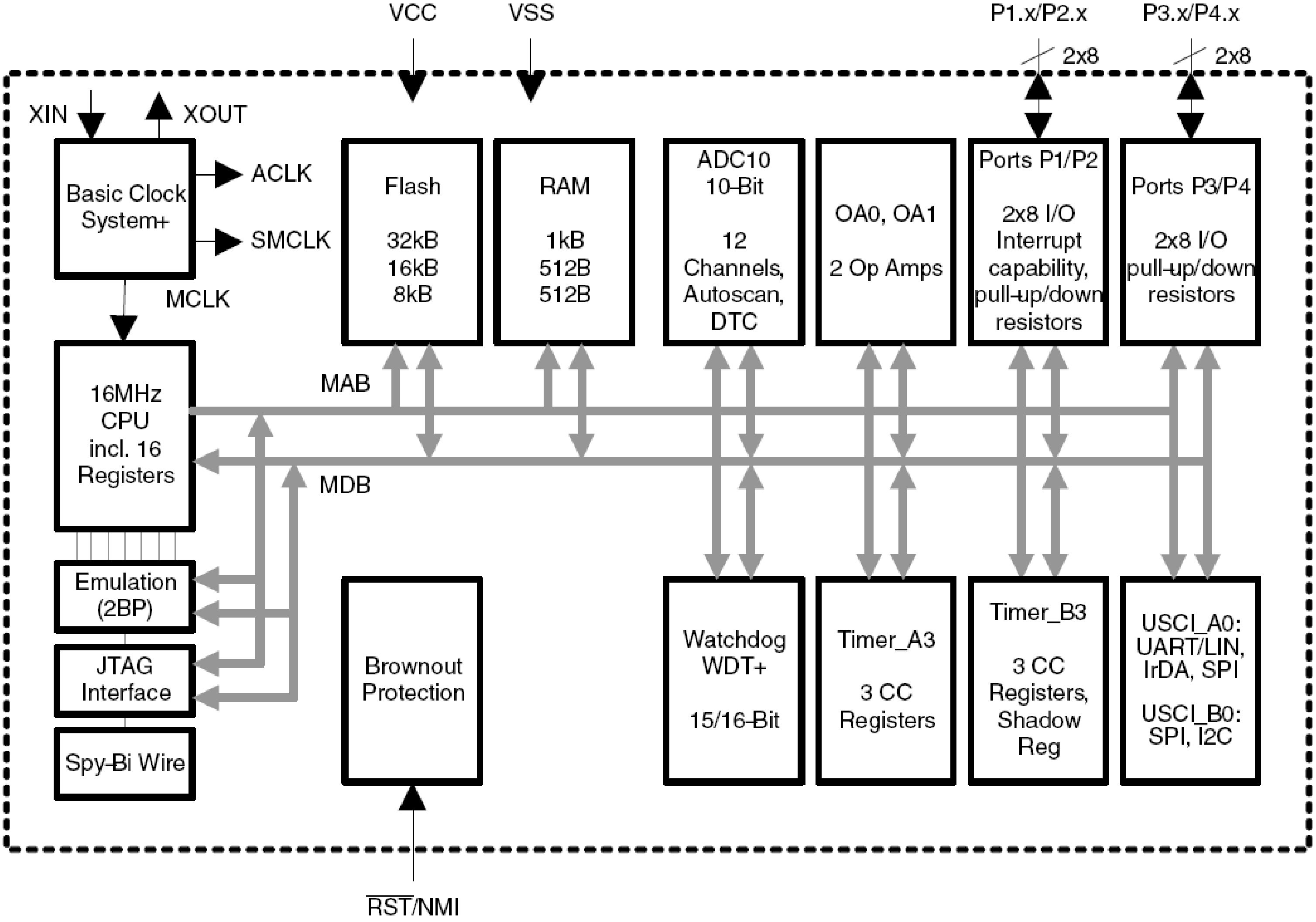
How to Assign Individual Bits (Better)

- Assign all 8-bits at once
  - P1OUT = BIT7 + BIT5 + BIT2 + BIT1 + BIT0
- Assign individual bits high
  - P1OUT |= BIT7 + BIT0
- Assign individual bits low
  - P1OUT &= ~(BIT7 + BIT0)
- Toggling individual bits
  - P1OUT ^= BIT7 + BIT0

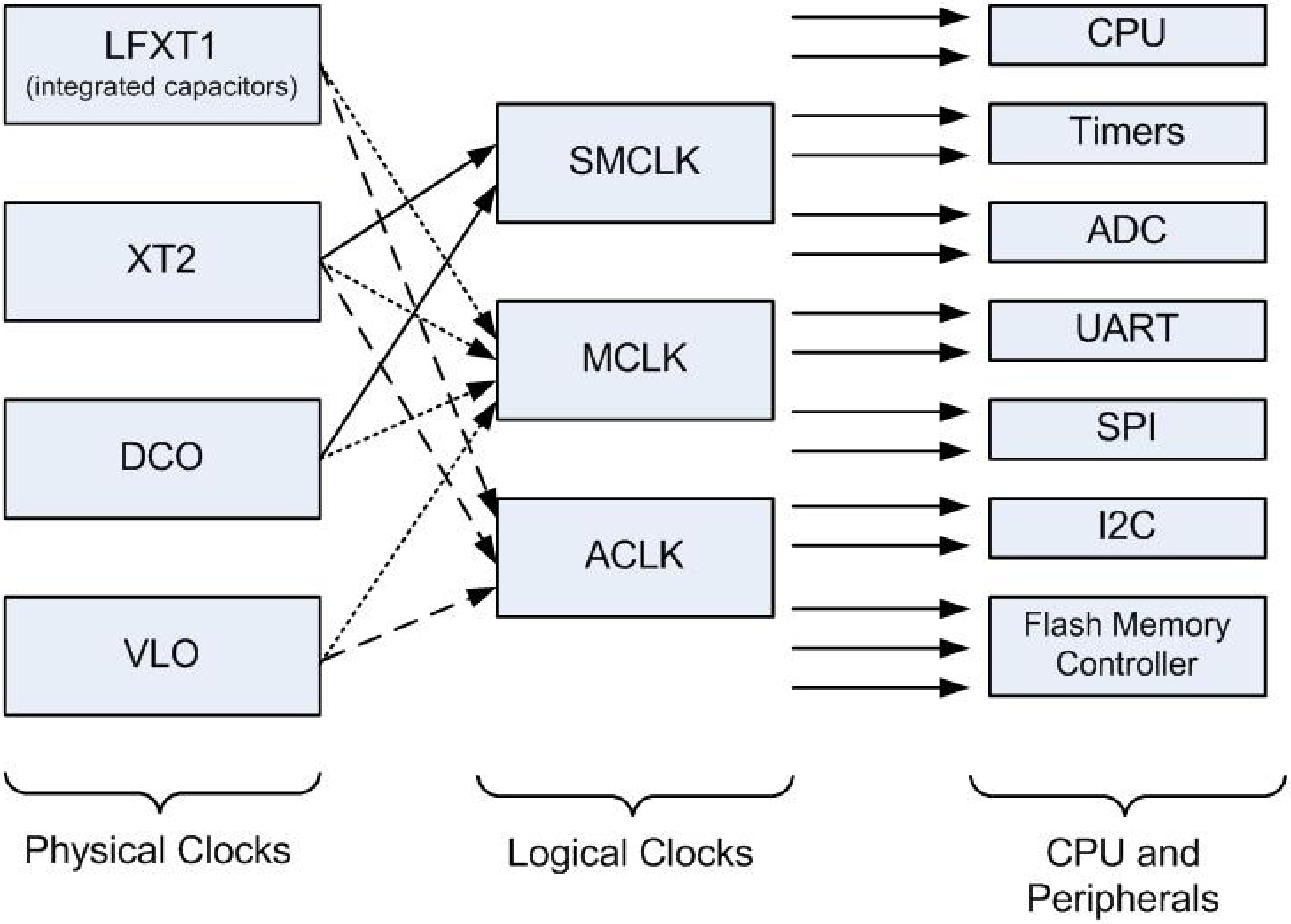
Configuring Ports

```
Main()
{
...
P1DIR |= BIT0 + BIT1 + BIT2 + BIT3 + BIT4 + BIT5;
//Set output mode
P1SEL |= BIT1 + BIT2;
//Output Timer_A1 and Timer_A2
P1REN |= BIT6 + BIT7;
//Enable pull-up/down resistors for BIT6 and BIT7
P1OUT |= BIT0 + BIT6
//Output high on BIT0; Pull-up BIT6
P1OUT &= BIT3 + BIT4 + BIT5 + BIT7
//Output low on BIT3, BIT4, and BIT5; Pull-down BIT7
...
}
```

Next Topic: Clocks



MSP430 Clocking Scheme



Crystal Oscillators

- Extremely accurate – standard frequency tolerance = 20ppm
- Many frequencies: 20kHz – GHz
- Real Time Clock: 32.768kHz
- Requires 2 external capacitors
- LFXT1 has integrated capacitors
- Ceramic resonator
  - Smaller, cheaper cousin
  - Frequency tolerance ~ 0.5%

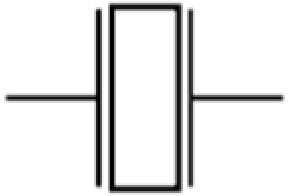


Photo removed due to copyright restrictions.

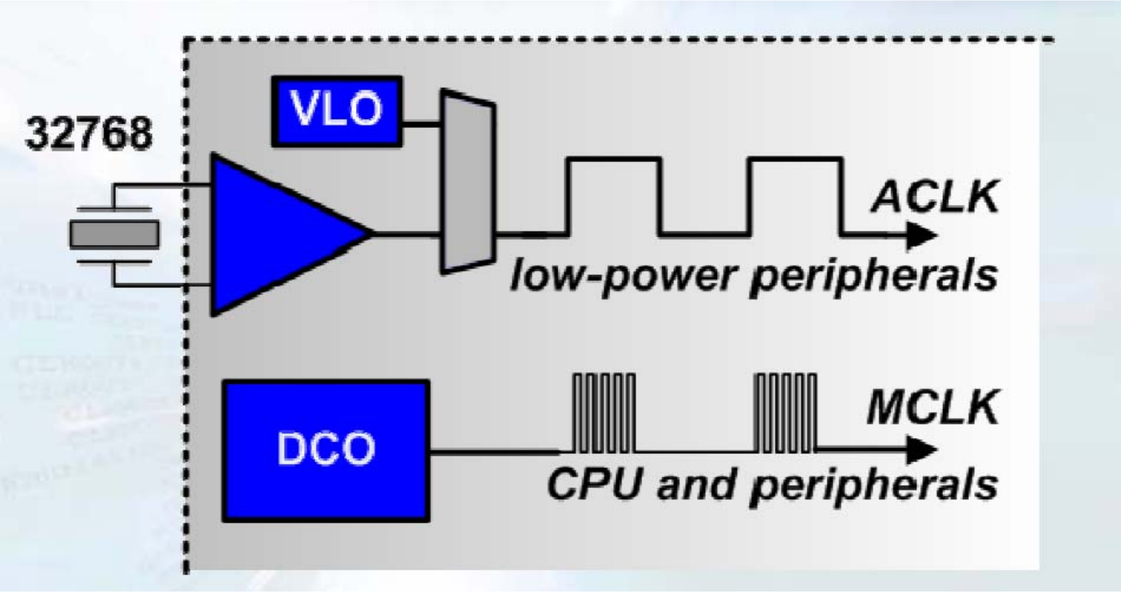
DCO (Digital Controlled Oscillator)

- 0 to 16 MHz
- Fast start-up <1uS
- ±3% tolerance
- ±6% tolerance over temperature
- Factory calibration in Flash
- Good enough for UART
- Application: watch

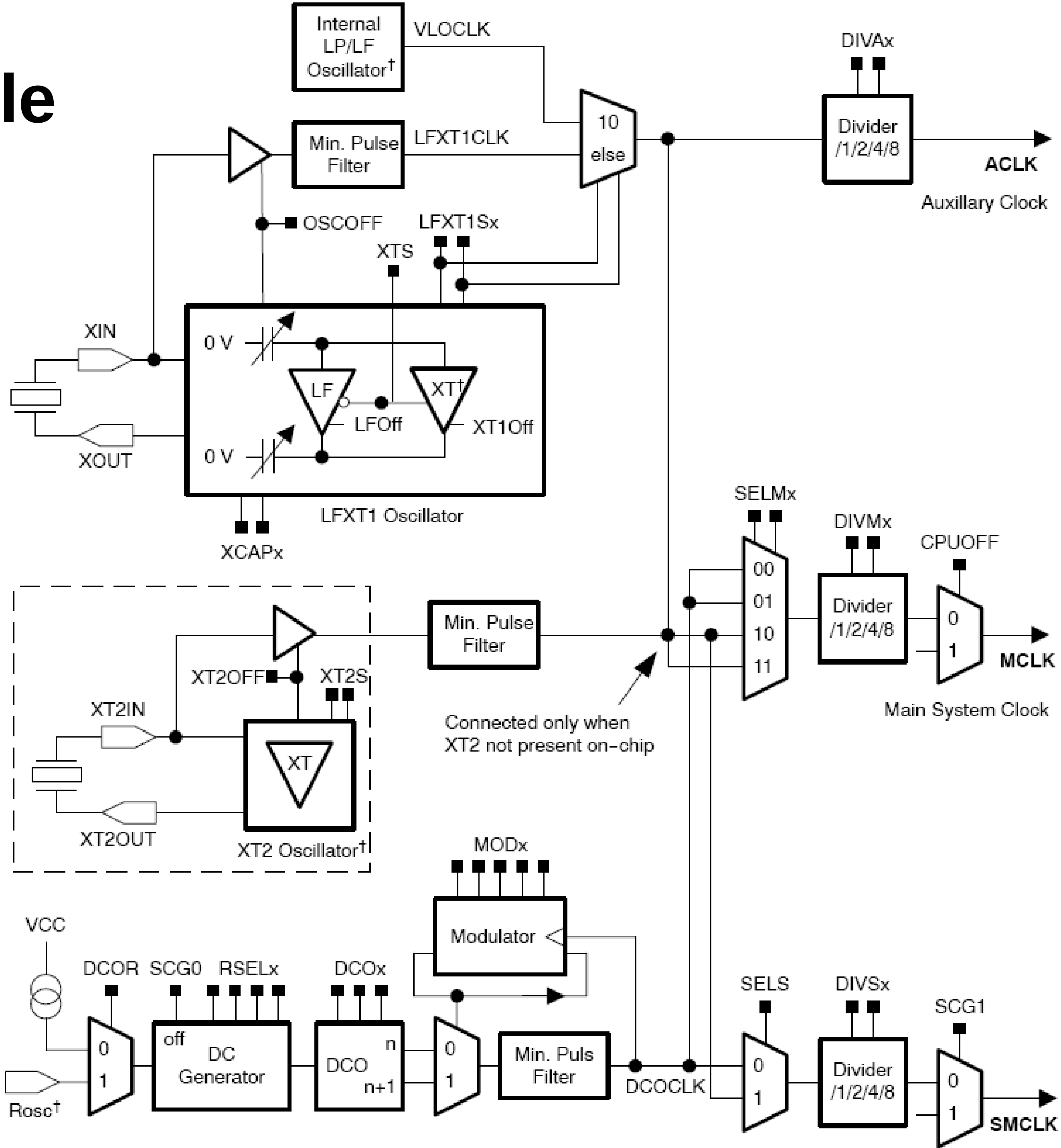
Images removed due to copyright restrictions.

VLO (Very Low-power Oscillator)

- 0.6µA typical at 25°C
- ~12 kHz (min 4kHz, max 20kHz)
- Can be calibrated using the DCO or XT
- Applications: temperature monitor, blood glucose sensor



Clock Module Diagram



Setting Up the Clock Module

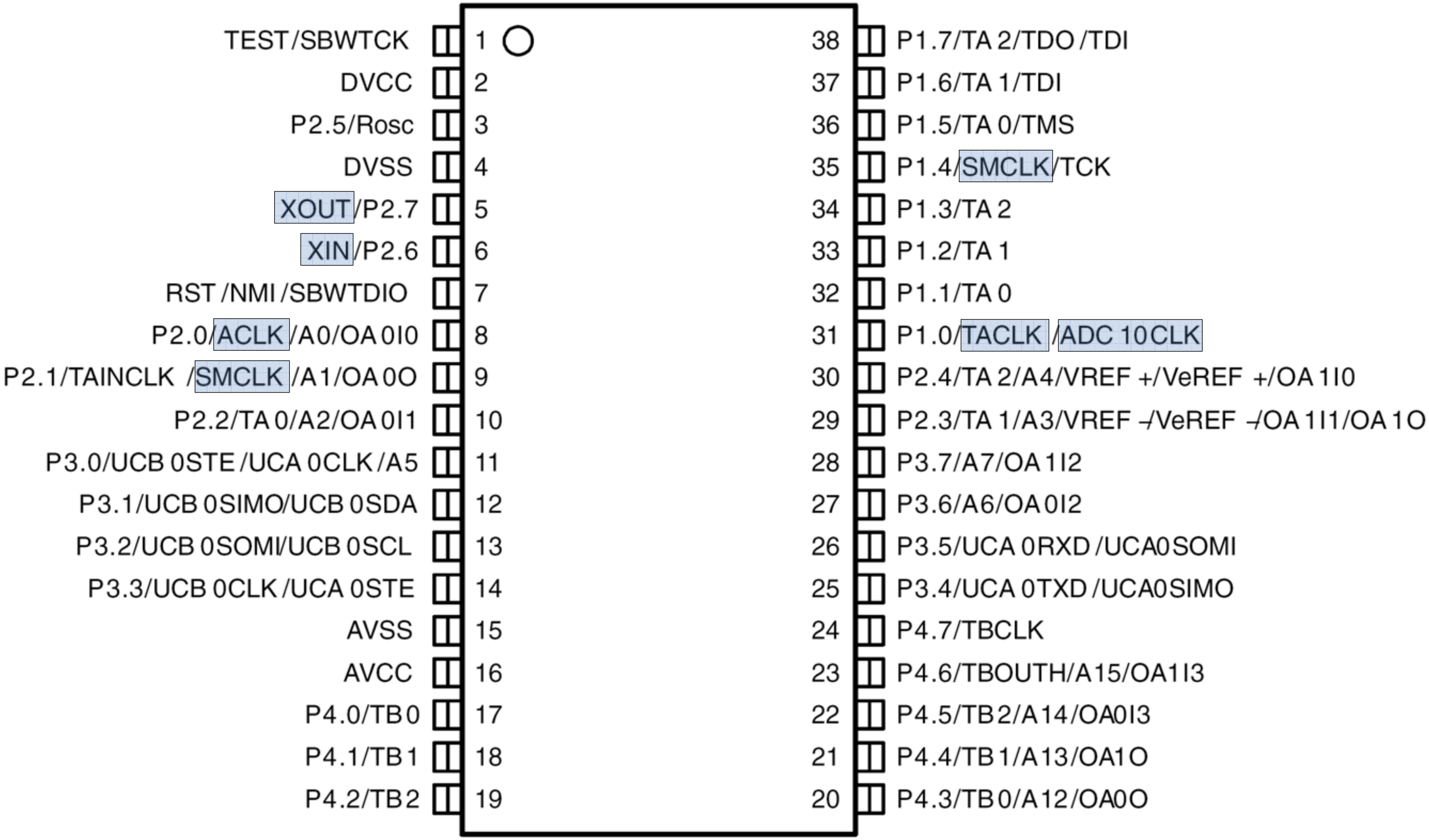
```
Main()
{
...
// 16MHz xtal clock setup: MCLK = 16MHz, SMCLK = 2MHz
BCSCTL1 = XT2OFF | XTS;
// No XT2, LFXT1 in high frequency mode

BCSCTL2 = SELM1 | SELM0 | SELS | DIVS1 | DIVS0;
// MCLK source is LFXT1;
// SMCLK source is LFXT1;
// SMCLK is divided by a factor of 8

BCSCTL3 = LFXT1S1;
// Select integrated capacitors for 3-16MHz resonator
...
}
```

Clock Ports

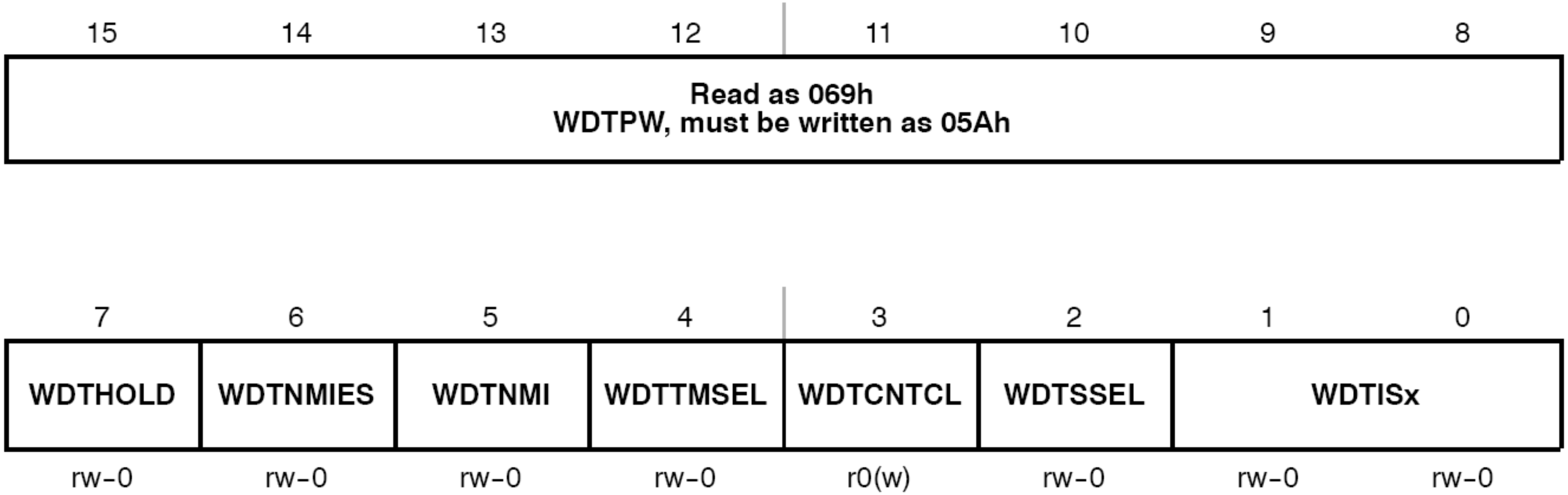
MSP430x22x4 device pinout, DA package



Watch-dog Timer (WDT)

- Designed to detect
  - Software halting
  - Oscillator fault
- Active after device reset
- “Kicking the dog” → Reset the WDT
- WDT runs down to 0 → Processor reset
- MSP430 WDT:
  - Automatically switch clocks after failure
  - Password protected
  - Can be used as an ordinary timer

MSP430 WDT



15	14	13	12	11	10	9	8
Read as 069h WDTPW, must be written as 05Ah							

7	6	5	4	3	2	1	0
WDTHOLD	WDTNMIES	WDTNMI	WDTTMSEL	WDTCNTCL	WDTSSEL	WDTISx	
rw-0	rw-0	rw-0	rw-0	r0(w)	rw-0	rw-0	rw-0

```
WDTCTL = WDTPW + WDTCNTCL;
// Clear WDT
```

```
WDTCTL = WDTPW + WDTHOLD;
// Stop WDT
```

Structure of MSP430 Program

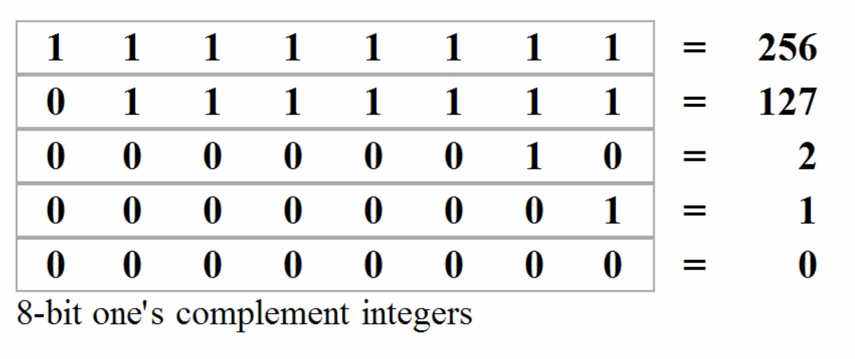
1. Declarations
2. `main()`
    1. Watch-dog timer servicing
    2. Setup clocking module
    3. Setup peripheral modules ←
    4. Enable interrupts
    5. Infinite loop
3. Subroutines
4. Interrupt Service Routines (ISR)

Variable Types

Type	Size	Single-cycle instruction
char	8-bits	Yes
int	16-bits	Yes
long	32-bits	No
float	64-bits	No

Number Representation

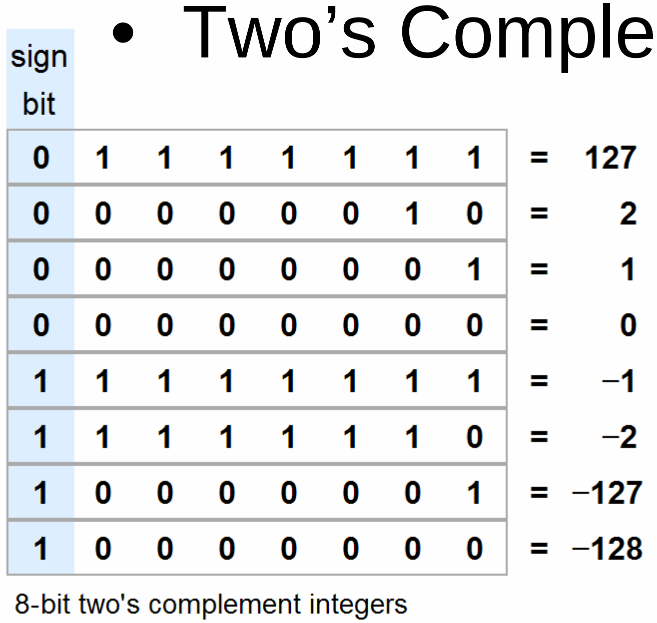
- One’s Complement



1	1	1	1	1	1	1	1	=	256
0	1	1	1	1	1	1	1	=	127
0	0	0	0	0	0	1	0	=	2
0	0	0	0	0	0	0	1	=	1
0	0	0	0	0	0	0	0	=	0

8-bit one's complement integers

- Two’s Complement



sign bit									
0	1	1	1	1	1	1	1	=	127
0	0	0	0	0	0	1	0	=	2
0	0	0	0	0	0	0	1	=	1
0	0	0	0	0	0	0	0	=	0
1	1	1	1	1	1	1	1	=	−1
1	1	1	1	1	1	1	0	=	−2
1	0	0	0	0	0	0	1	=	−127
1	0	0	0	0	0	0	0	=	−128

8-bit two's complement integers

```
//One’s comp. definition
unsigned char var1
unsigned int var2
```

```
//Two’s comp. definition
signed char var1
signed int var2
```

Always explicitly define signed / unsigned !!!

Global Variables

- Global variables not always updated due to compiler optimization

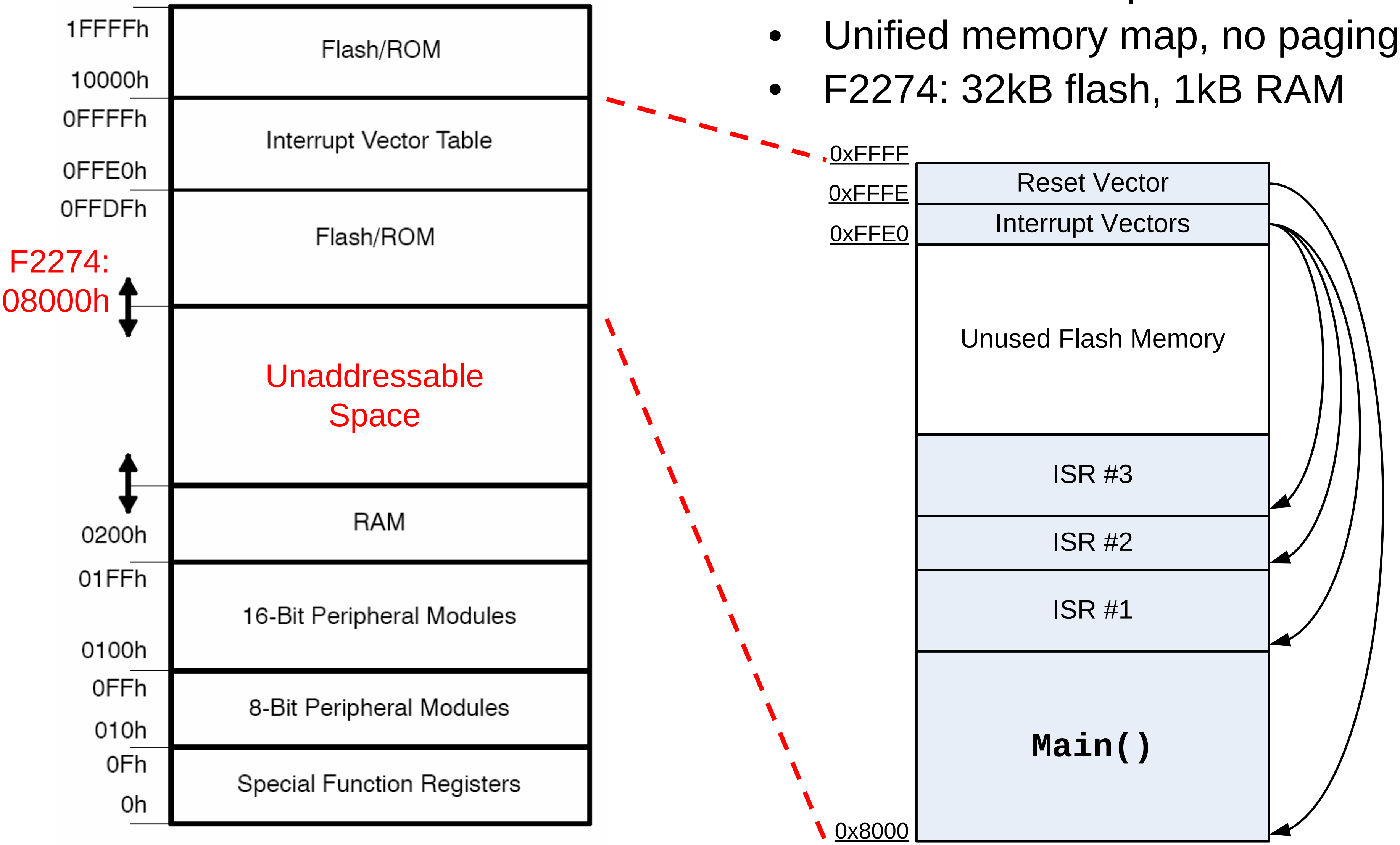
```
//Declarations
unsigned char var
volatile unsigned char gvar
...

Main()
{
...
gvar=1;
while(1);
}

#pragma vector=USCIAB0RX_VECTOR
__interrupt void UART_RX(void)
{
gvar=2;
...
}
```

MSP430F2xx Address Space

- 128kB address space
- Unified memory map, no paging
- F2274: 32kB flash, 1kB RAM



1FFFFh
Flash/ROM
10000h

0FFFFh
Interrupt Vector Table
0FFE0h

0FFDFh
Flash/ROM

F2274: 08000h

Unaddressable Space

RAM
0200h

01FFh
16-Bit Peripheral Modules
0100h

0FFh
8-Bit Peripheral Modules
010h

0Fh
Special Function Registers
0h

0xFFFF
Reset Vector
0xFFFE
Interrupt Vectors
0xFFE0
Unused Flash Memory
ISR #3
ISR #2
ISR #1
Main()
0x8000

Embedded Programming Styles

- Simple
  - – Poll for events in `main()`
- Interrupt-driven
  - – Code reside in the ISR
  - – Used for handling a single interrupt source
- Event-driven
  - – ISR sets flags for events
  - – `main()`poll for flags and services them
  - – Used for handling multiple interrupts sources

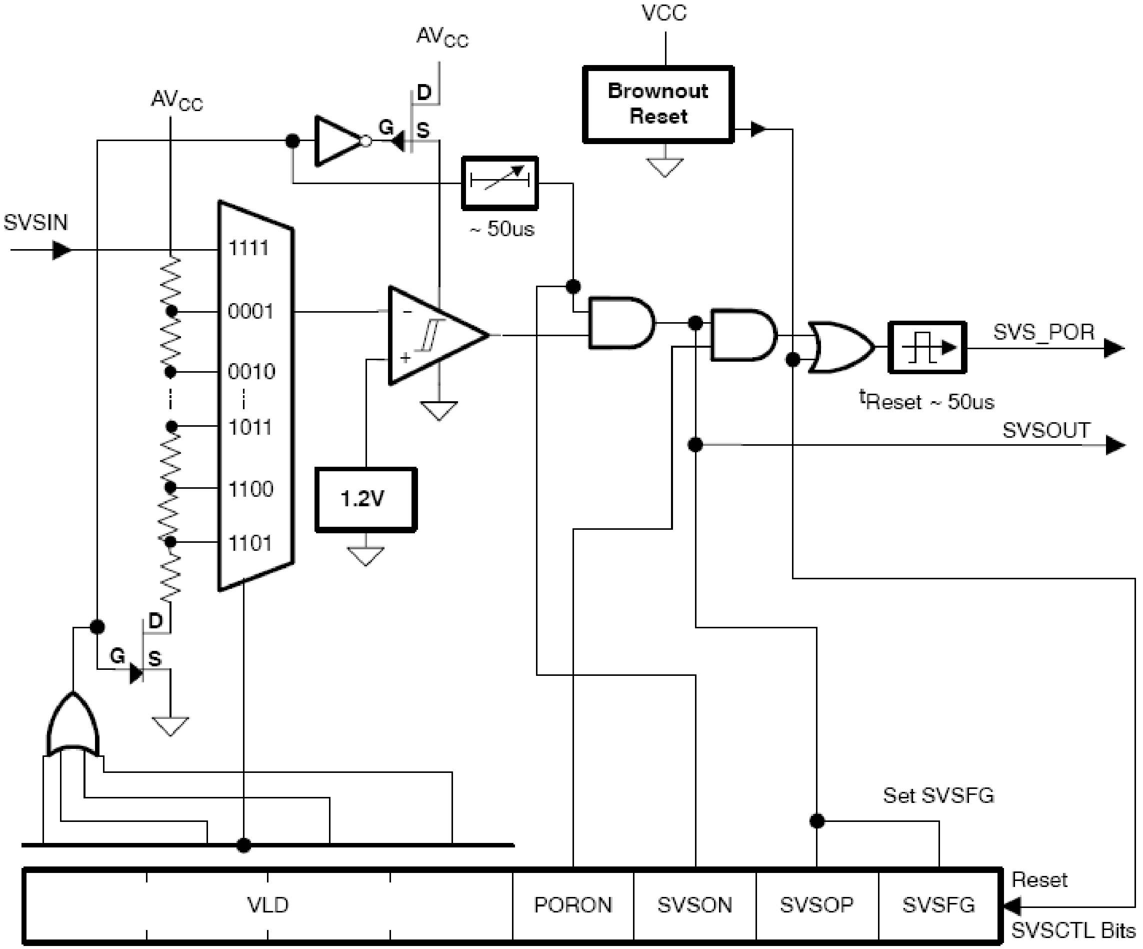
Components for Microprocessor Programming

- ICE – In-Circuit Emulator
  - Flash Emulation Tool (FET)
  - JTAG
  - Spy-Bi-Wire (2-wire JTAG)
- Bootloader
  - Rewrite flash via RS232
  - Password protected
- IDE – Integrated Development Environment
  - Editor, compiler, debugger
- Libraries for each microprocessor

Image removed due to copyright restrictions.

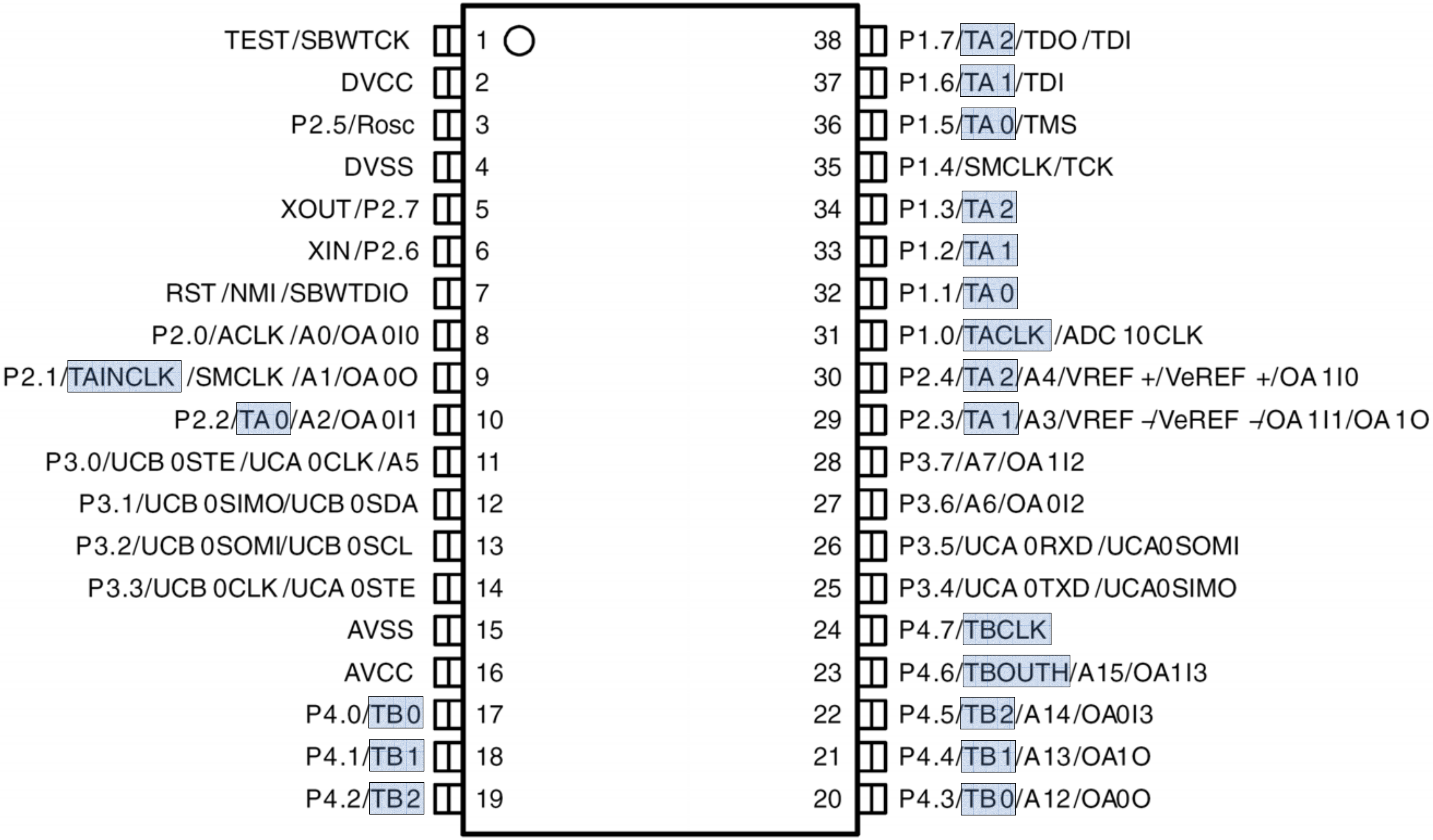
Brownout Detector and SVS

- Brownout detector triggers a POR when supply voltage drops below 1.8V
- SVS (Supply Voltage Supervisor) – Comparator-based (flash) ADC



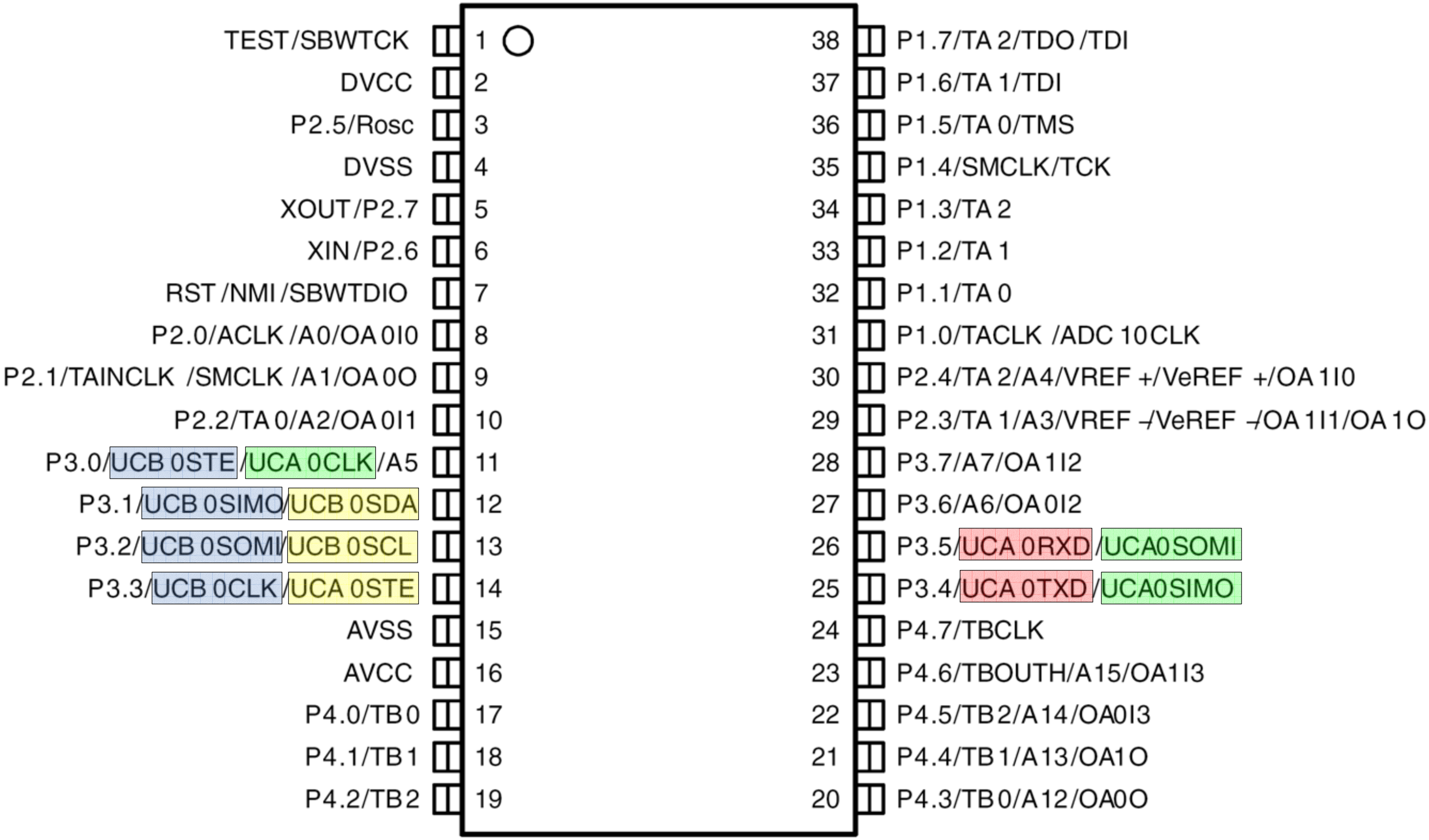
Timer Related I/O

MSP430x22x4 device pinout, DA package



Communications Ports

MSP430x22x4 device pinout, DA package



Start / Reset Sequence

- PUC (Power Up Clear)
- POR (Power On Reset)

Example C Code