第2章　单片机结构

第1节　单片机简介

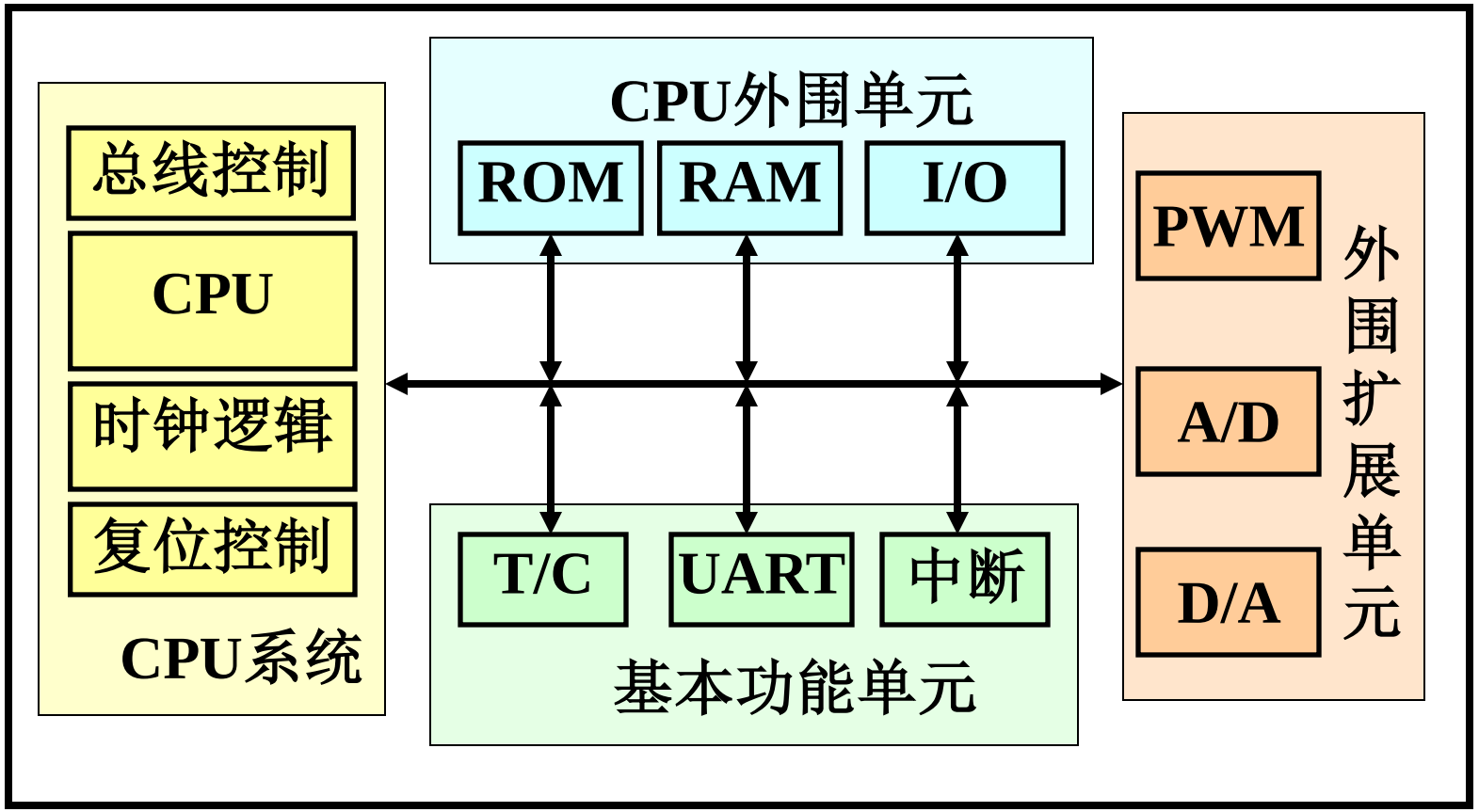
一、单片机与嵌入式系统

二、典型单片机的基本构成

一、单片机与嵌入式系统

- 将CPU、内存、I/O接口集成一块芯片上，

构成具备基本功能的计算机，称单片机。



典型单片机的基本构成

单片机 Single-chip Microcomputer

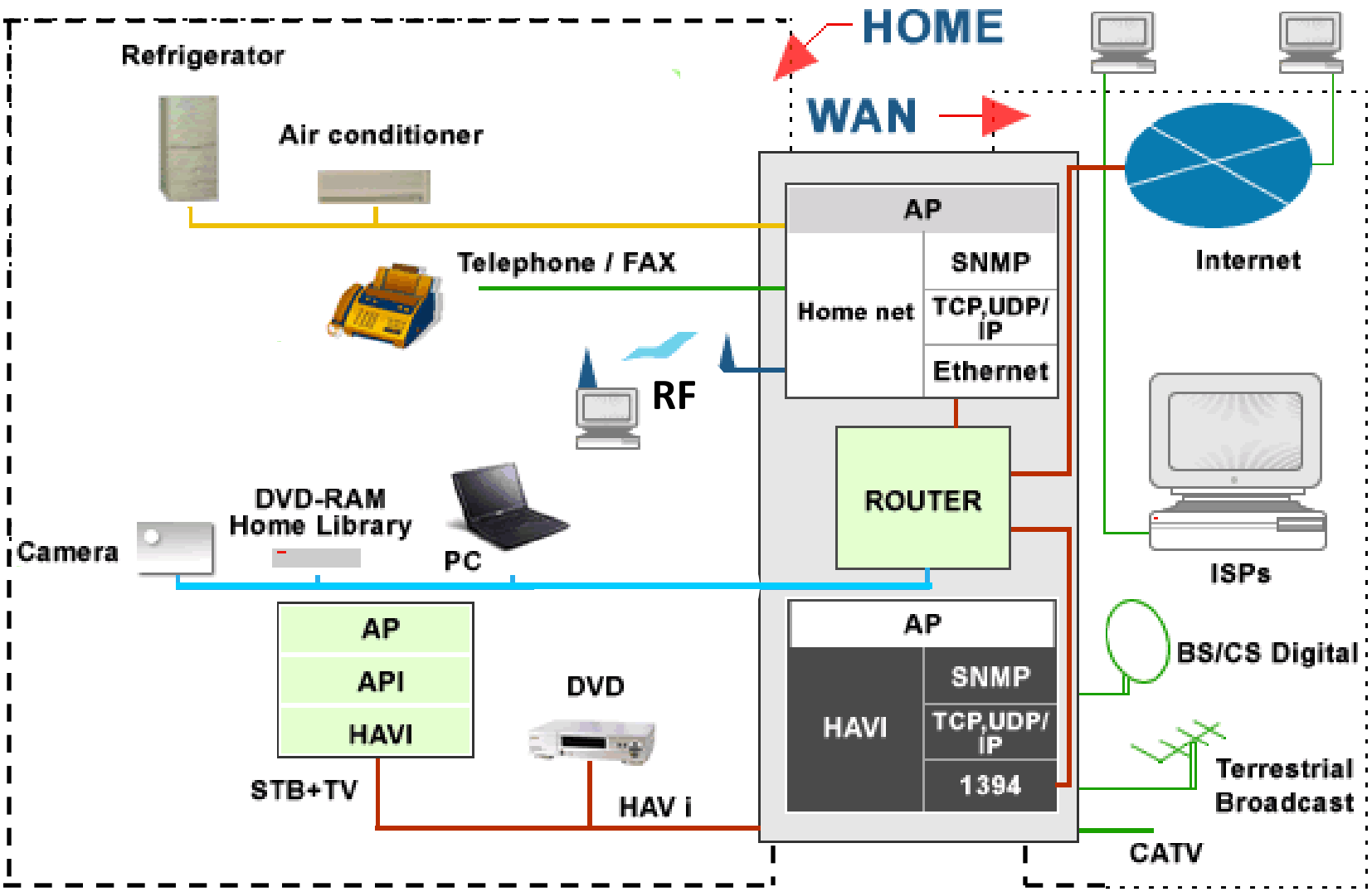
- 超小型、高可靠性、价廉
- 主要应用于控制领域，

  用以实现各种参数的测量和控制功能，

  为了强调其控制属性，称其为微控制器，

  MCU，MicroController Unit.

消费类电子产品



Home Audio/Video interoperability

平均每辆轿车内使用30~100个MCU

信息系统

- 信息处理
- GPS导航系统
- 娱乐
- 移动通信

传动控制系统

- 发动机控制
- 车速控制
- 节能控制
- 转向控制

行驶控制系统

- 仪表盘控制
- 空调控制
- 底盘控制

安全控制系统

- 安全气囊
- 防滑控制

车身控制系统

- 开关控制
- 车灯控制
- 门窗控制
- 防盗控制

汽车电子

一台通用计算机的外部设备中就包含了多个微控制器：

键盘、鼠标、硬盘、CD-ROM驱动、显卡、显示器、打印机、扫描仪等均含有MCU。

单片机的主要应用领域

- **工业控制**

  电机控制、物理量的检测与处理、机器人、过程控制、智能传感器、 数据传送等。

- **仪器仪表**

  智能仪器仪表、医疗器械、色谱仪、示波器等。

- **家用电子电器设备**

  电子字典、游戏机、录像机、电冰箱、洗衣机、照相机、空调、防盗控制等。

- **通信**

  调制解调器、程控电话交换机、遥控、手机等。

- **导航控制**

  鱼雷制导控制、智能武器装置、导弹控制、航天导航系统等。

- **数据处理**

  图形终端、复印机、硬盘驱动器、磁带机、打印机、打字机等。

- **汽车电子**

  点火控制、变速控制、防滑刹车、排气控制、导航、节能控制、冷气控制、报警等

单片机构成的应用系统特点

- 嵌入到具体的应用系统中，不以计算机的面貌出现；
- 面向控制对象；

  输入的是对象物理量经传感变换的信号；

  输出的是对象伺服驱动的控制量
- 突出控制功能；

  对外部信息及时捕捉；

  对控制对象能灵活地实时控制；
- 能在工业现场环境中可靠运行。

与满足海量高速数值计算的计算机有着明显不同

计算机按应用分类　　通用计算机系统、嵌入式计算机系统

通用计算机系统：　　**“非嵌入式应用”**

满足海量高速数值计算，主要用于信息处理，

能独立使用的计算机系统。

如：个人计算机，工作站等。

技术要求是高速、海量的数值计算；

技术发展方向是总线速度的提升，存储容量的扩大。

嵌入式计算机系统：　“嵌入式应用”

作为其它系统组成部分、

以嵌入的形式“隐藏”在各种装置、产品和系统中，

实现嵌入式应用的计算 机称为嵌入式计算机系统，

简称嵌入式系统(**Embedded System**) 。

嵌入式系统(Embedded System)的定义

- **IEEE的定义： They are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system.**

- 国内普遍认可的定义：以应用为中心，以计算机技术为基础，软件硬件可裁剪，符合应用系统对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。

在嵌入式系统的概念广泛使用后，

为了强调单片机的嵌入属性，也称其为嵌入式微控制器。

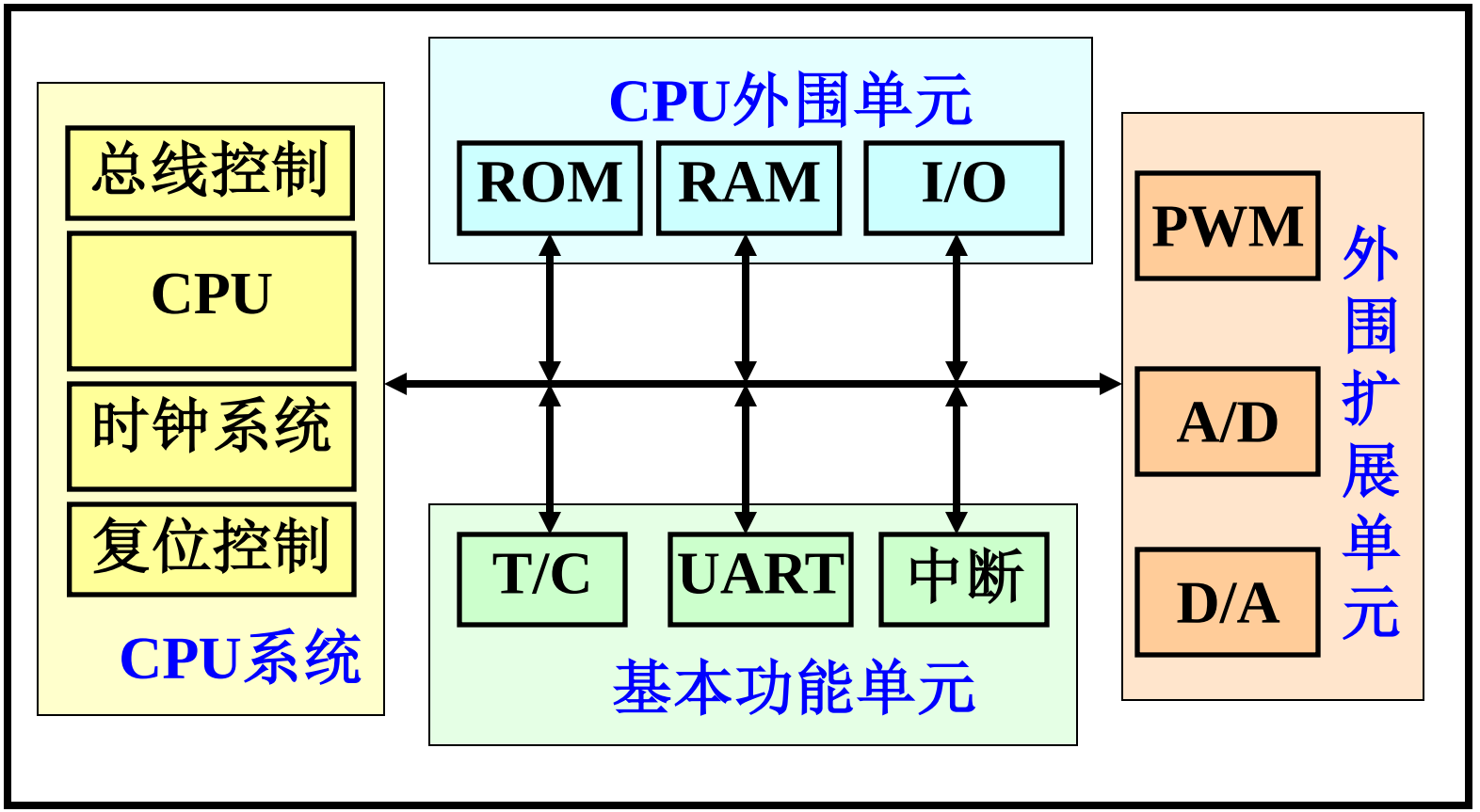
单片机是典型的嵌入式系统。

- 嵌人式系统开发面向具体应用，
  不同领域的应用市场需要不同款式和性能指标的单片机来开发.
  所以在嵌人式处理器市场中，
  高性能的32位MCU也有很多产品，
  中低端的4位、8位和16位MCU依然存在，
  在当前的嵌人式市场中不存在垄断的局面。

- PC机销售市场中，
  随着通用CPU技术的突破和工作频率的倍增，
  旧款低档CPU早已经不见踪迹。

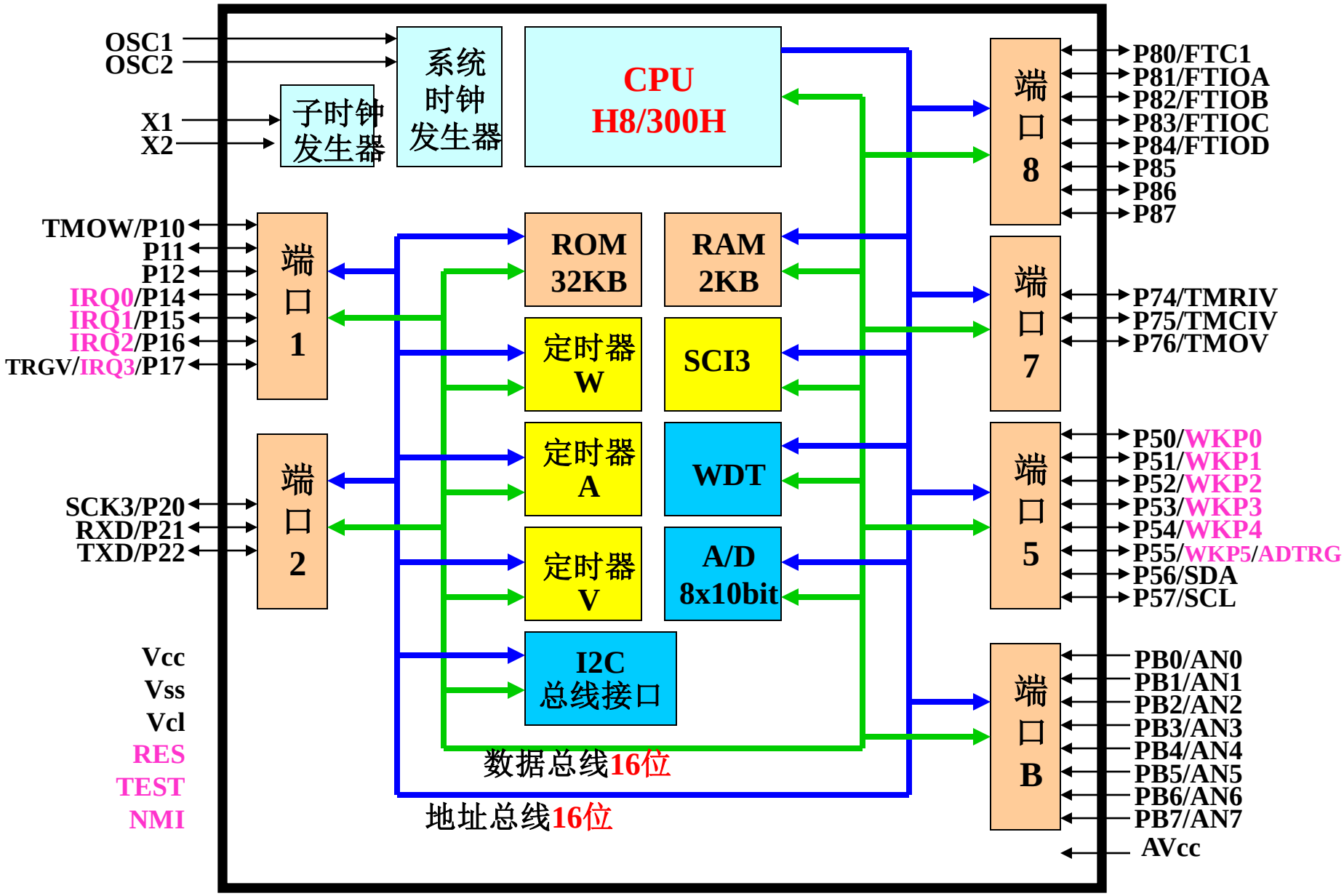
  在PC领域虽有AMD系列处理器和Linux操作系统的市场冲击，
  但是Win_ Tel体系架构(Windows+lntel) 仍占主导地位

二. 典型单片机的基本构成

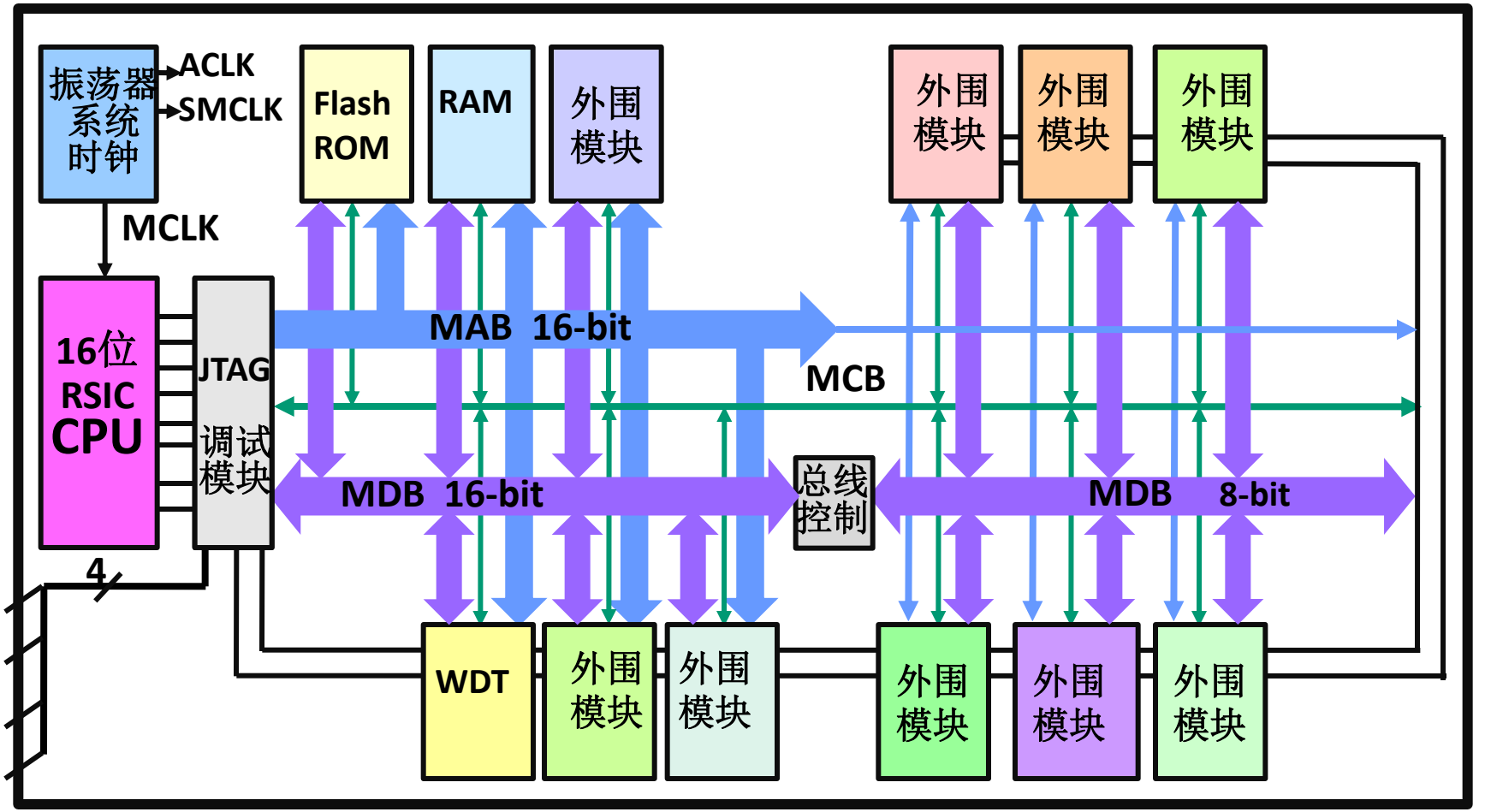


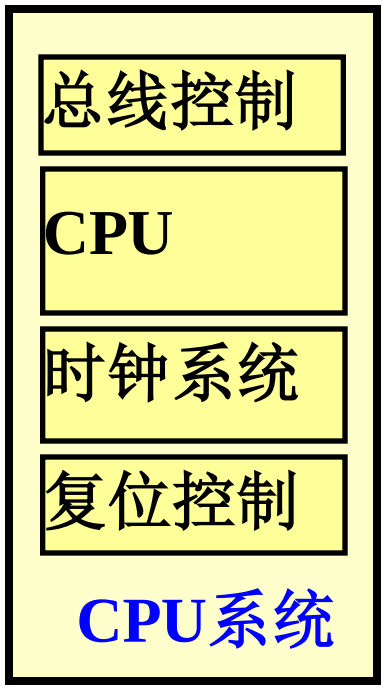
典型单片机的基本构成

瑞萨16位MCU H8/3664 结构框图



MSP430单片机结构图





- **CPU**

  内含寄存器,运算器，有突出的控制功能。

- **时钟系统**

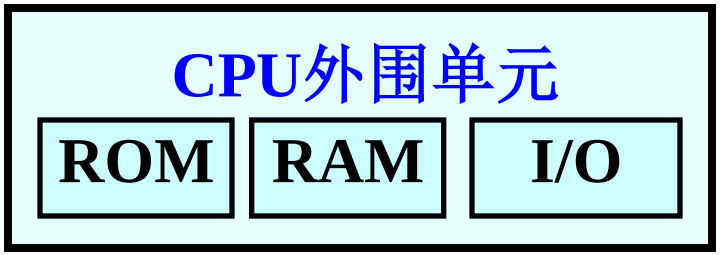
  提供各部分的时序同步

- **复位电路**

  提供CPU初始化所需的上电复位信号。

- **总线控制逻辑**

  实现MCU对内部总线和外部总线的控制。



与CPU运行直接相关的单元电路，与CPU构成单片机的最小系统。

- 程序存储器ROM （ReadOnlyMemory)

  用于固化单片机的应用程序和一些表格、常数。

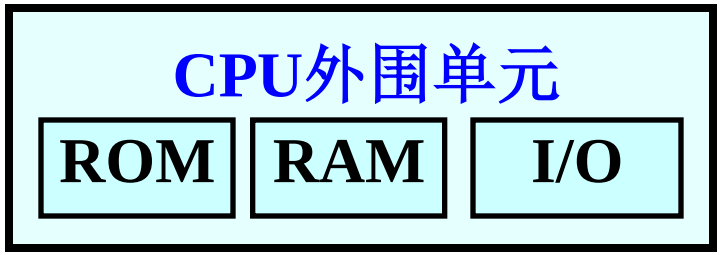
  ① ROMLess型 ： 无ROM

  ② MaskROM型： 掩模型ROM

  ③ OTPROM型 ： 可编程1次

  ④ EPROM型 ： 光擦除电编程

  ⑤ FlashROM型： 电擦除电编程



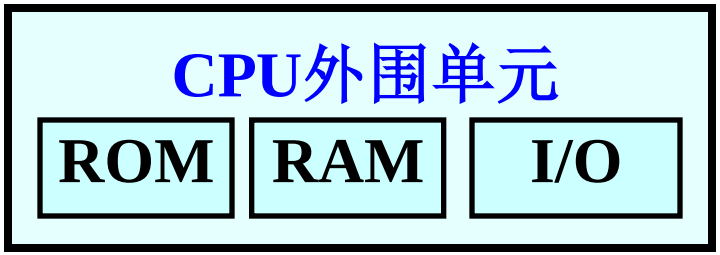
- **数据存储器RAM**

用于临时存放应用程序的处理数据、结果等。

由于面向测控系统，单片机中的数据存储器容量较小，

通常是256字节~2K字节，

使用静态随机存储器SRAM(Static Random Access Memory)。



- **I/O端口**

单片机与外部的输入输出接口，含芯片的输入/输出引脚。

按功能，I/O端口分为以下几种类型：

① 基本I/O端口

用于与外部电路信号的输入/输出控制。

②片内功能单元的I/O端口

例如，定时器/计数器的计数输入、外部中断源输入等。

③总线输入/输出端口

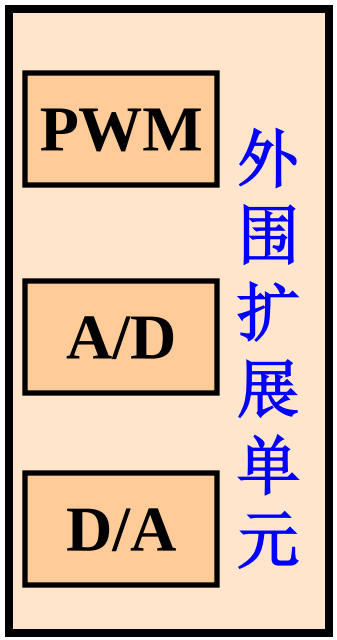
用于外部总线的扩展。

为了减少引脚数量，单片机的I／O端口一般采用复用方式。

- 满足单片机测控功能要求的基本外围电路，
  用来完善和扩大计算机功能。
  如定时器 / 计数器、中断系统、串行通信接口等。

- 每个功能单元含一个或多个寄存器，通过编程，
  实现这些功能单元的方式设置、启动运行、状态读取等，
  起到对功能单元的运行控制、管理功能。

  在有些单片机（51系列），称为特殊功能寄存器SFR
  (Special Function Regiser)



- 满足不同嵌入式应用要求的外围扩展功能电路

如 满足数据采集要求而扩展的**ADC**， **DAC**，

满足伺服驱动控制的**PWM**

满足程序可靠运行的监视定时器**WDT**

(WatchDogTimer)等。

每个系列单片机都有一个**CPU基核**，

在基核上扩展不同的外围单元电路，

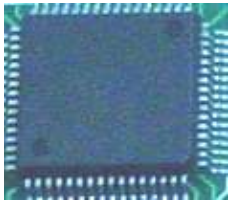
衍生出各种型号的单片机。

例如　　**TI的MSP430或MSP430x系列MCU**

Intel 的51系列MCU

Renesas的H8/300H系列MCU

TI几款MSP430 CPU的 MCU配置比较

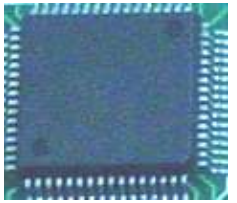


型号	MSP430F135	MSP430F149	MSP430F169
Flash ROM	16KB	60KB	60KB
SRAM	512B	2KB	2KB
I/O引脚数	48	48	48
TimerA	3	3	3
TimerB	3	7	7
USART	1	2	2
ADC12	有	有	有
硬件乘法器MPY	无	有	有
DAC12	无	无	有
I2C	无	无	有
DMA	无	无	有
电源电压检测SVS	无	无	有

TI几款MSP430x CPU(称CPUX)的 MCU配置比较

设备	闪存 (KB)	SRAM (KB) [1]	Timer_A[2]	Timer_B[3]	USCI		ADC12_A (Ch)	DAC12_A (Ch)	Comp_B (Ch)	I/O	封装类型
					通道A: UART/IrDA/ SPI	通道 B: SPI/I²C					
MSP430F6638	256	16 + 2	5, 3, 3	7	2	2	12 ext / 4 int	2	12	74	100 PZ, 113 ZQW
MSP430F6637	192	16 + 2	5, 3, 3	7	2	2	12 ext / 4 int	2	12	74	100 PZ, 113 ZQW
MSP430F6636	128	16 + 2	5, 3, 3	7	2	2	12 ext / 4 int	2	12	74	100 PZ, 113 ZQW
MSP430F6635	256	16 + 2	5, 3, 3	7	2	2	12 ext / 4 int	-	12	74	100 PZ, 113 ZQW
MSP430F6634	192	16 + 2	5, 3, 3	7	2	2	12 ext / 4 int	-	12	74	100 PZ, 113 ZQW
MSP430F6633	128	16 + 2	5, 3, 3	7	2	2	12 ext / 4 int	-	12	74	100 PZ, 113 ZQW
MSP430F6632	256	16 + 2	5, 3, 3	7	2	2	-	-	12	74	100 PZ, 113 ZQW
MSP430F6631	192	16 + 2	5, 3, 3	7	2	2	-	-	12	74	100 PZ, 113 ZQW
MSP430F6630	128	16 + 2	5, 3, 3	7	2	2	-	-	12	74	100 PZ, 113 ZQW

从MSP430的命名规则了解 MCU基本配置



MSP430 F 14 9 I PM

存储器 ROM类型	
C	ROM
P	OTP
F	FLASH
E	EPROM

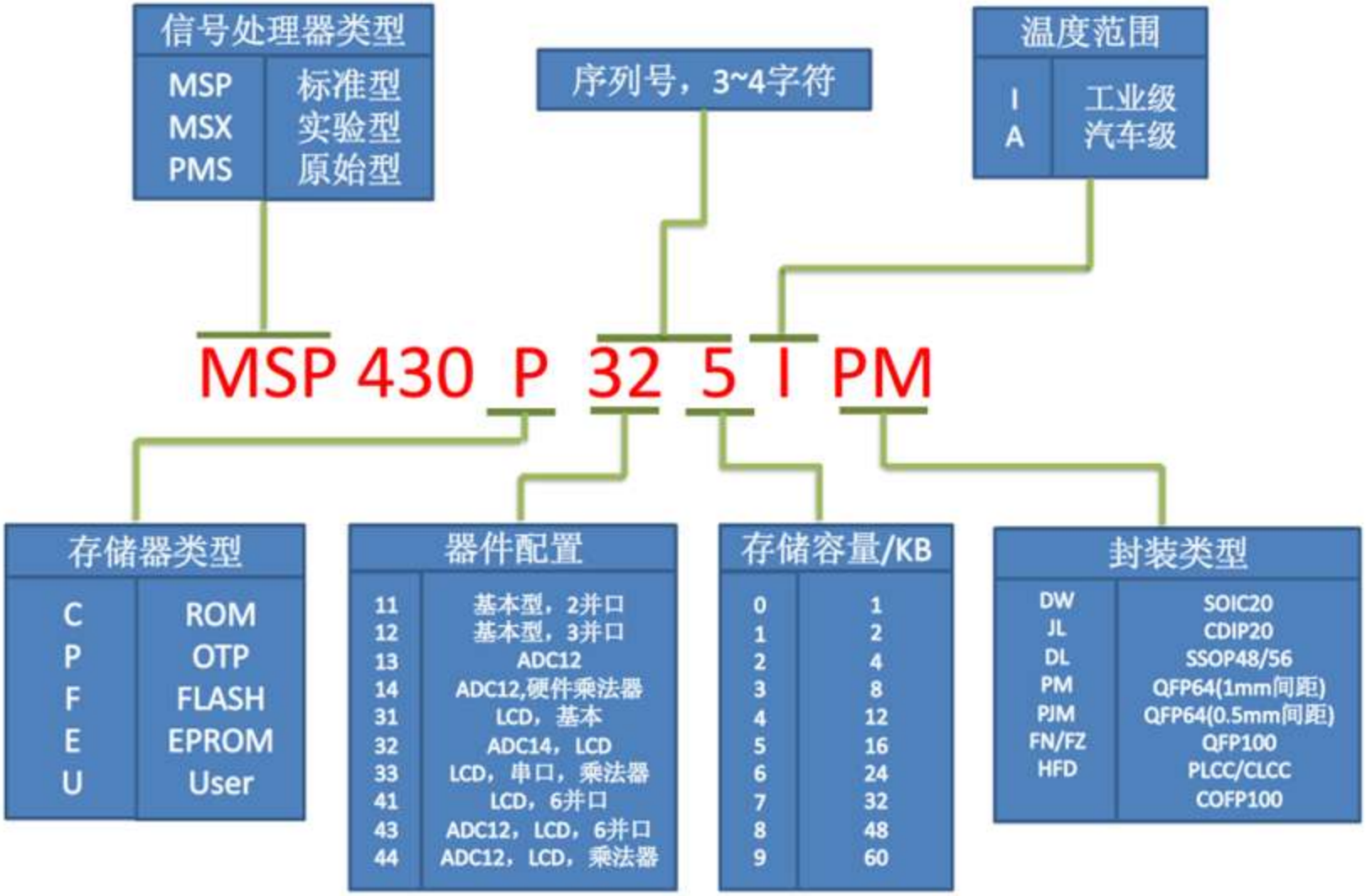
器件配置	
12	基本型, 3并口
13	ADC12, 1串口
14	ADC12, 2串口, 硬件乘法器
16	DAC12, 2串口,硬件乘法器
31	LCD, 1串口, 乘法器
41	ADC12, LCD, 6并口
44	ADC12, LCD, 乘法器

ROM容量	
2	4KB
5	16KB
7	32KB
8	48KB
9	60KB

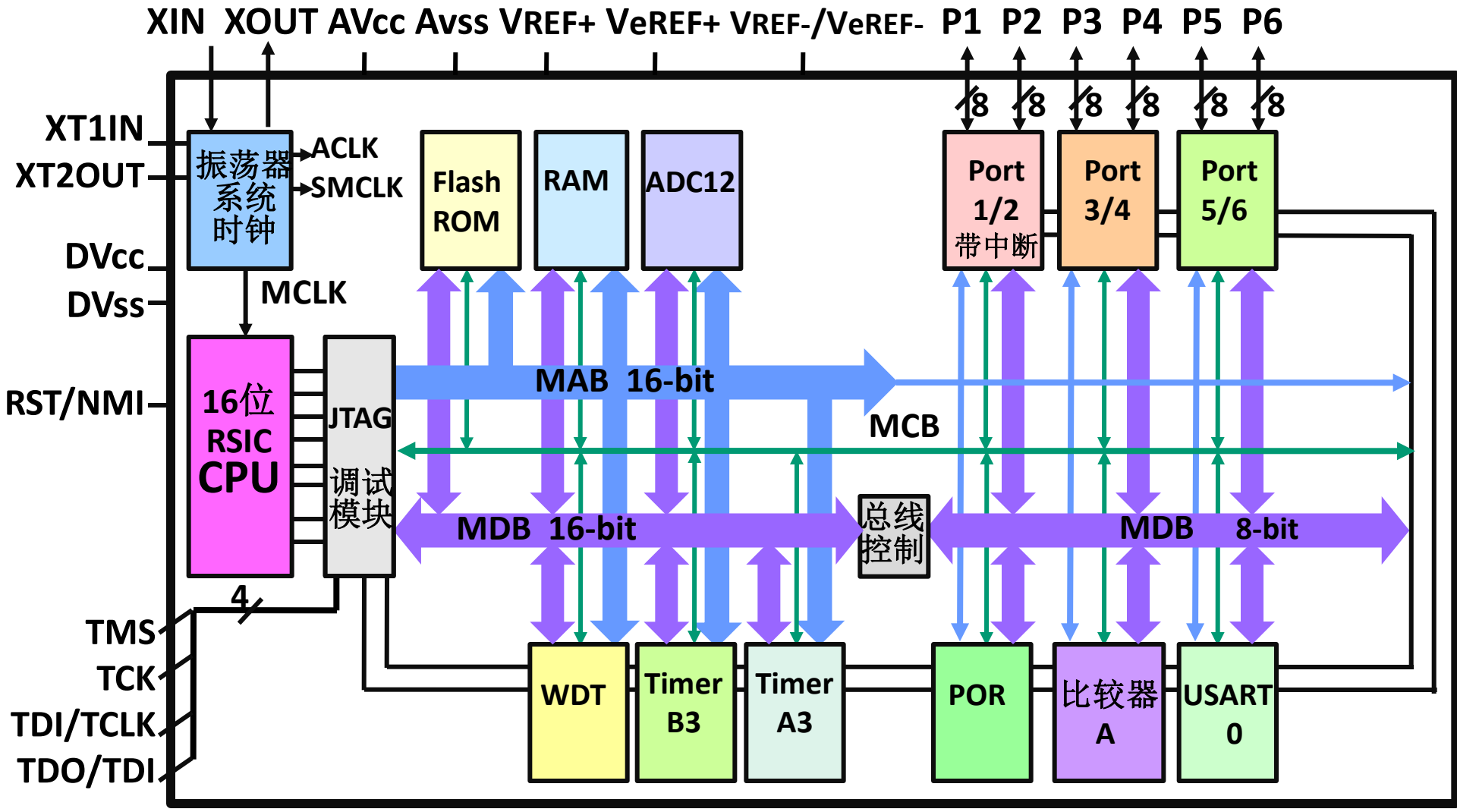
封装类型	
DW	SOIC20
PM	QFP64 (1mm间距)
PJM	QFP64 (0.5mm间距)

温度范围	
I	工业级: -40~+85℃
A	汽车级:-40~+125℃

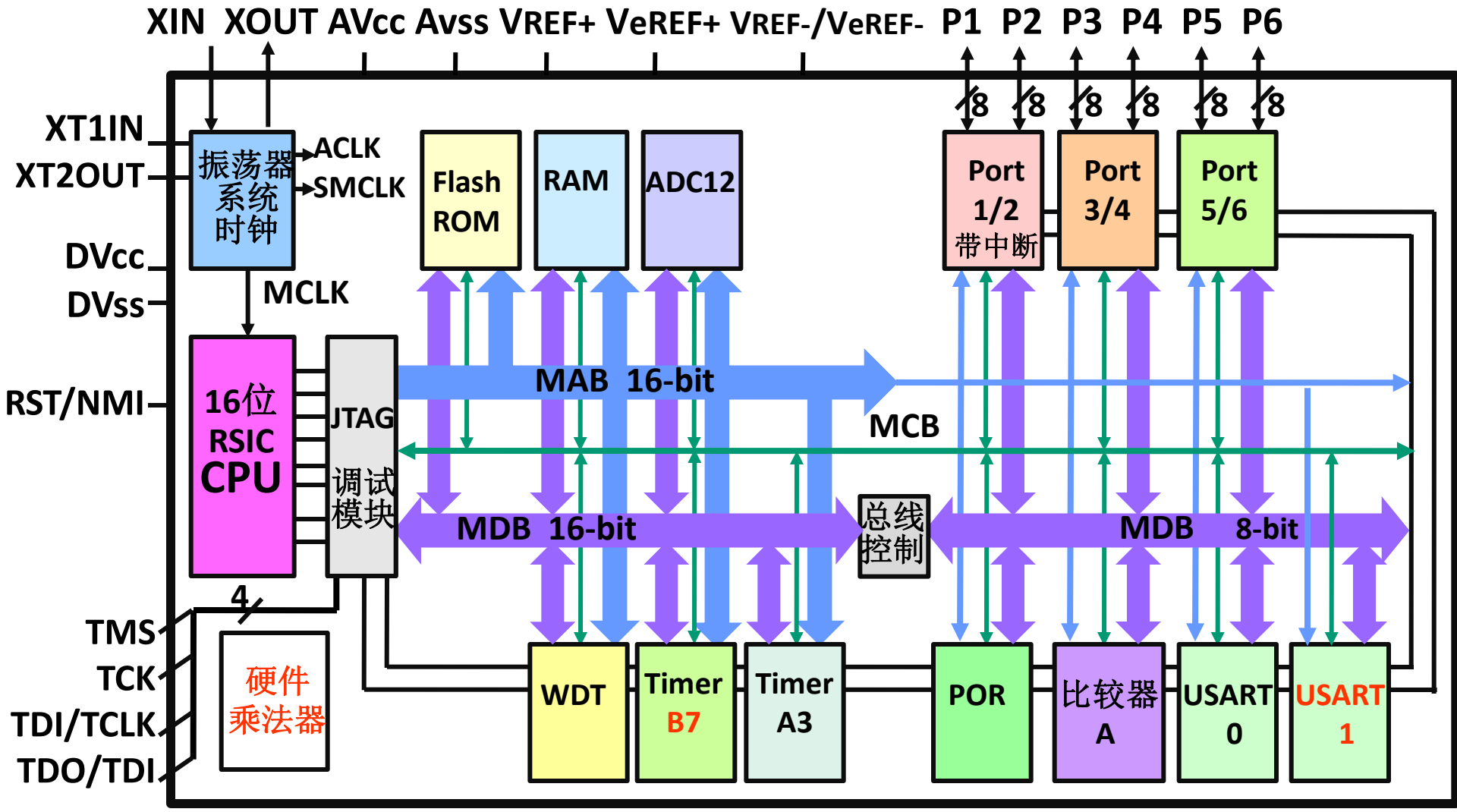
例 MSP430F135
MSP430F149
MSP430F169 均属于 MSP430x1xx系列


信号处理器类型
MSP 标准型
MSX 实验型
PMS 原始型
序列号，3~4字符
温度范围
I 工业级
A 汽车级
MSP 430 P 32 5 I PM
存储器类型
C ROM
P OTP
F FLASH
E EPROM
U User
器件配置
11 基本型，2并口
12 基本型，3并口
13 ADC12
14 ADC12,硬件乘法器
31 LCD，基本
32 ADC14，LCD
33 LCD，串口，乘法器
41 LCD，6并口
43 ADC12，LCD，6并口
44 ADC12，LCD，乘法器
存储容量/KB
0 1
1 2
2 4
3 8
4 12
5 16
6 24
7 32
8 48
9 60
封装类型
DW SOIC20
JL CDIP20
DL SSOP48/56
PM QFP64(1mm间距)
PJM QFP64(0.5mm间距)
FN/FZ QFP100
HFD PLCC/CLCC
COFP100

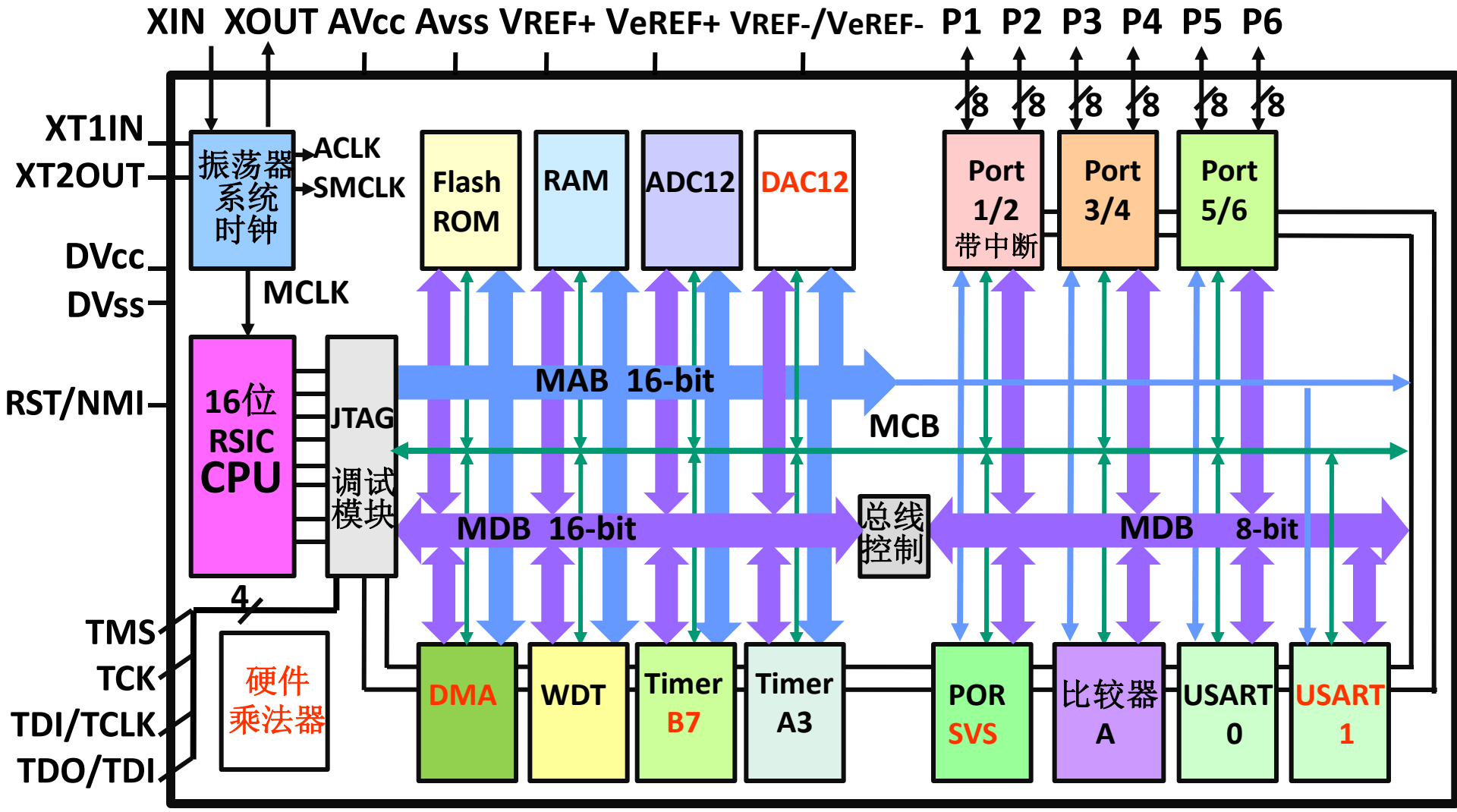
MSP430x13x功能结构图



MSP430x14x功能结构图

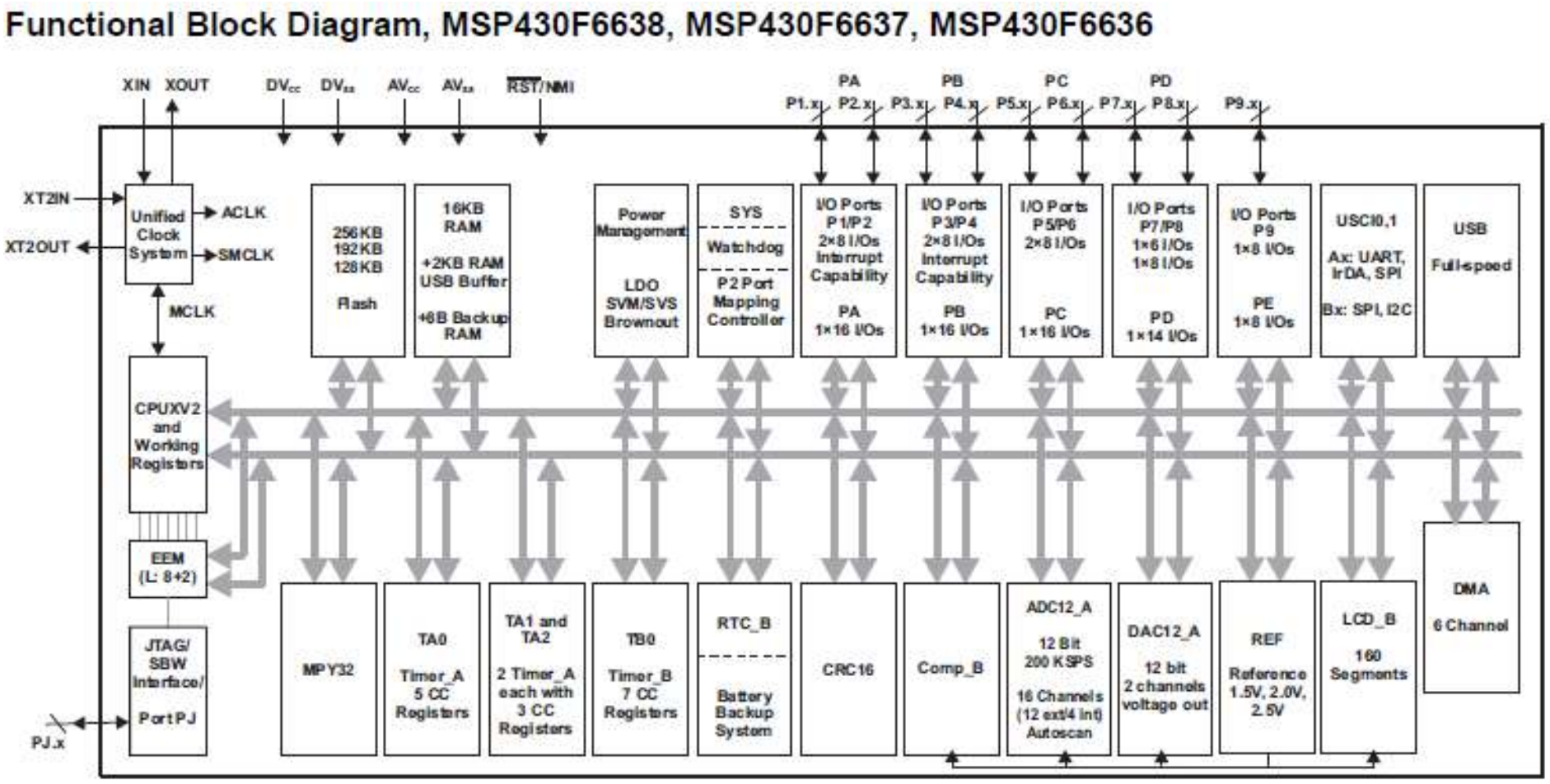


MSP430x16x功能结构图



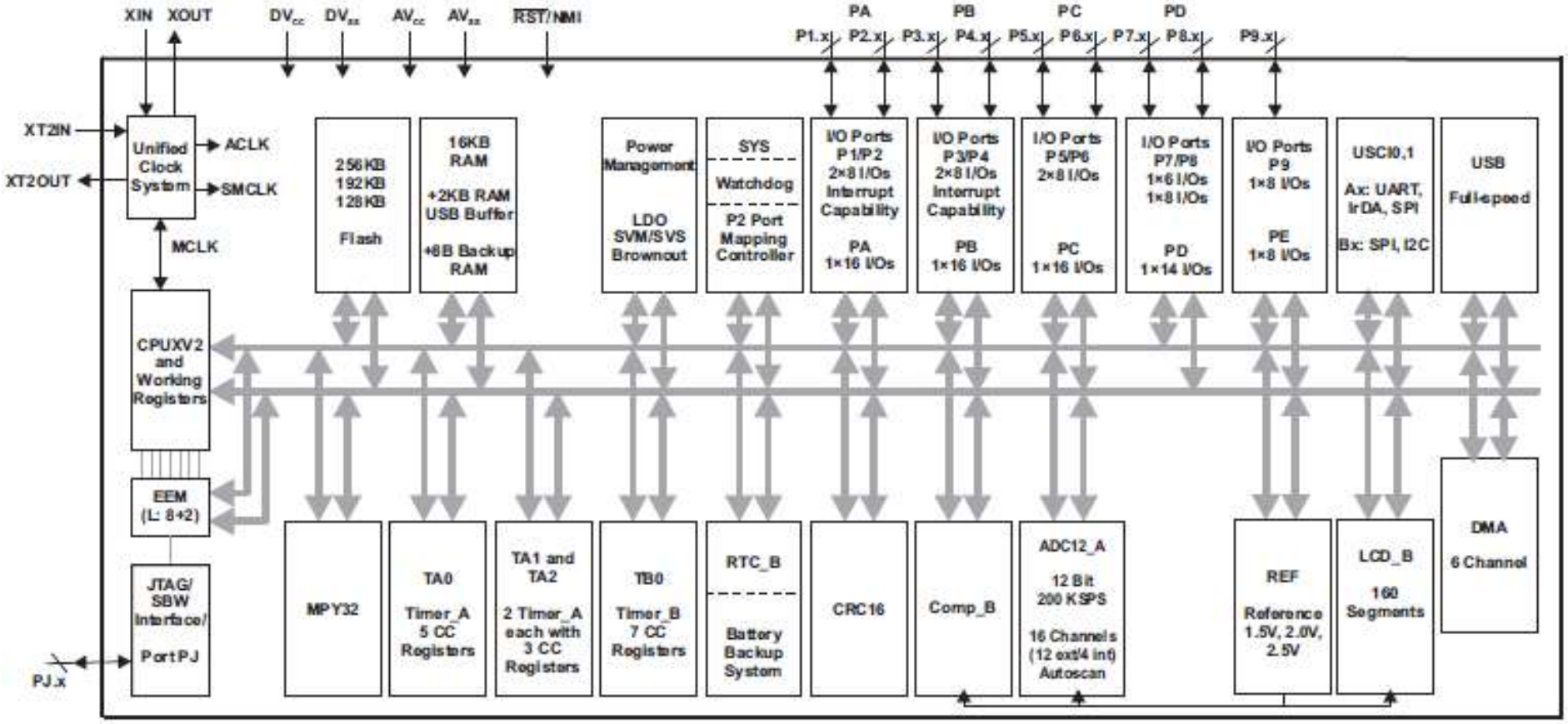
课后练习：阅读msp430x15x-16x.pdf 第5页，
比较msp430x15x和MSP430x16x 内部模块有何不同?

MSP430x6636/7/8功能结构图

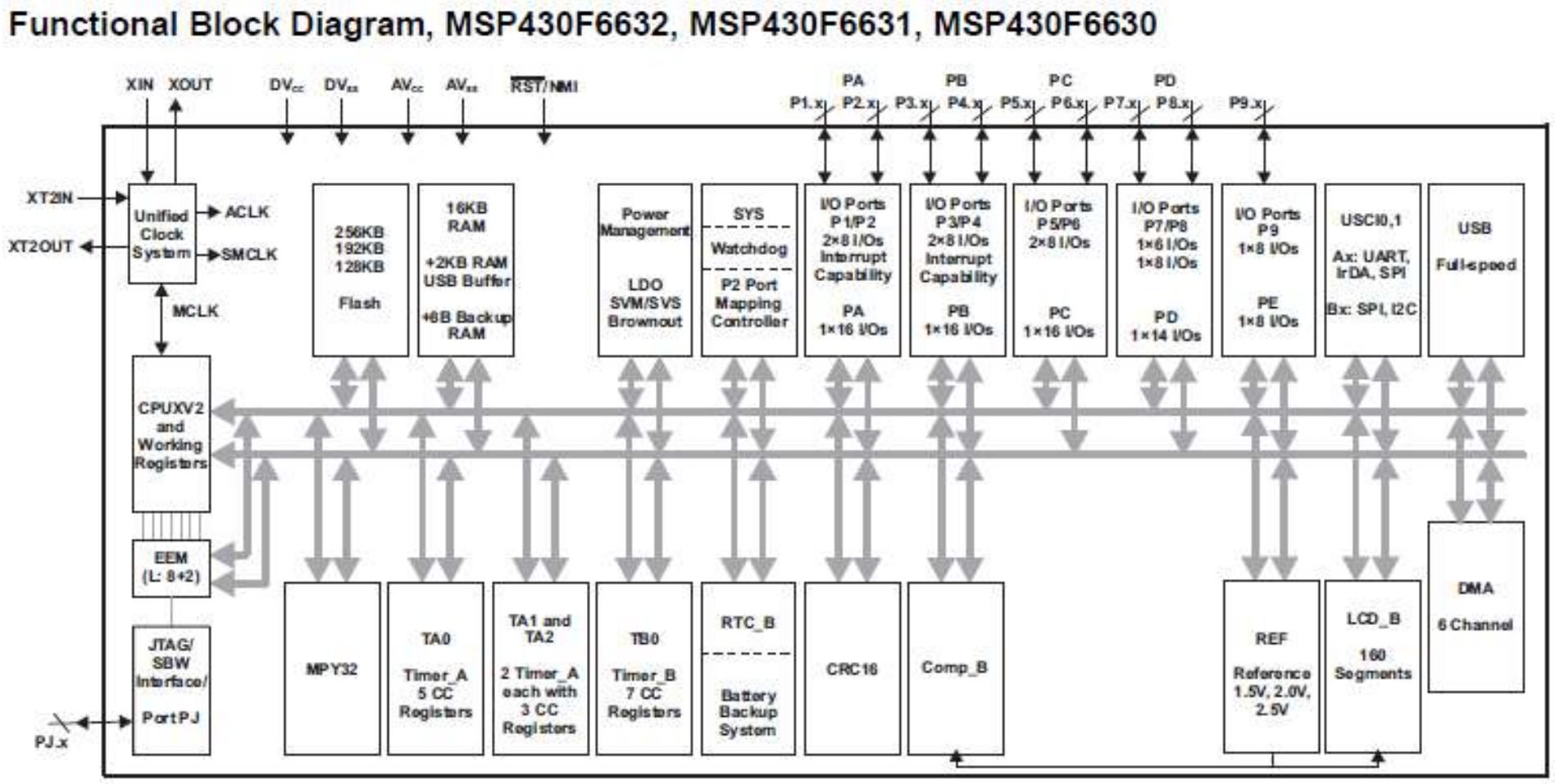


MSP430x6633/4/5功能结构图

Functional Block Diagram, MSP430F6635, MSP430F6634, MSP430F6633



MSP430x6630/1/2功能结构图



单片机技术发展趋势

- 低电压、低功耗、小封装、低价格

  采用CMOS工艺

- CPU内核性能、速度的提高

  4/8/16/32bit， CISC/RISC.

- ROM、RAM容量的增加
- 各种内置的通信接口电路

  SCI，SPI, I2C, CAN, USB，以太网通信接口.

- 支持在系统编程ISP(In System Programmable)

  PC机通过串行接口，实现对MCU的程序下载;

  以及通过仿真接口实现对MCU的仿真调试.

- 支持高级语言的编程
- 软件嵌入

  目前单片机上提供的主要是用户的程序空间，存放用户开发的应用软件

  随着单片机程序空间的扩大，嵌入 一些工具软件，提高开发效率.

第2节 MSP430单片机结构

一、 MSP430 单片机概述

二、 MSP430 的CPU编程结构

三、 MSP430x 的CPU（CPUX）编程结构

四、 MSP430的存储器组织

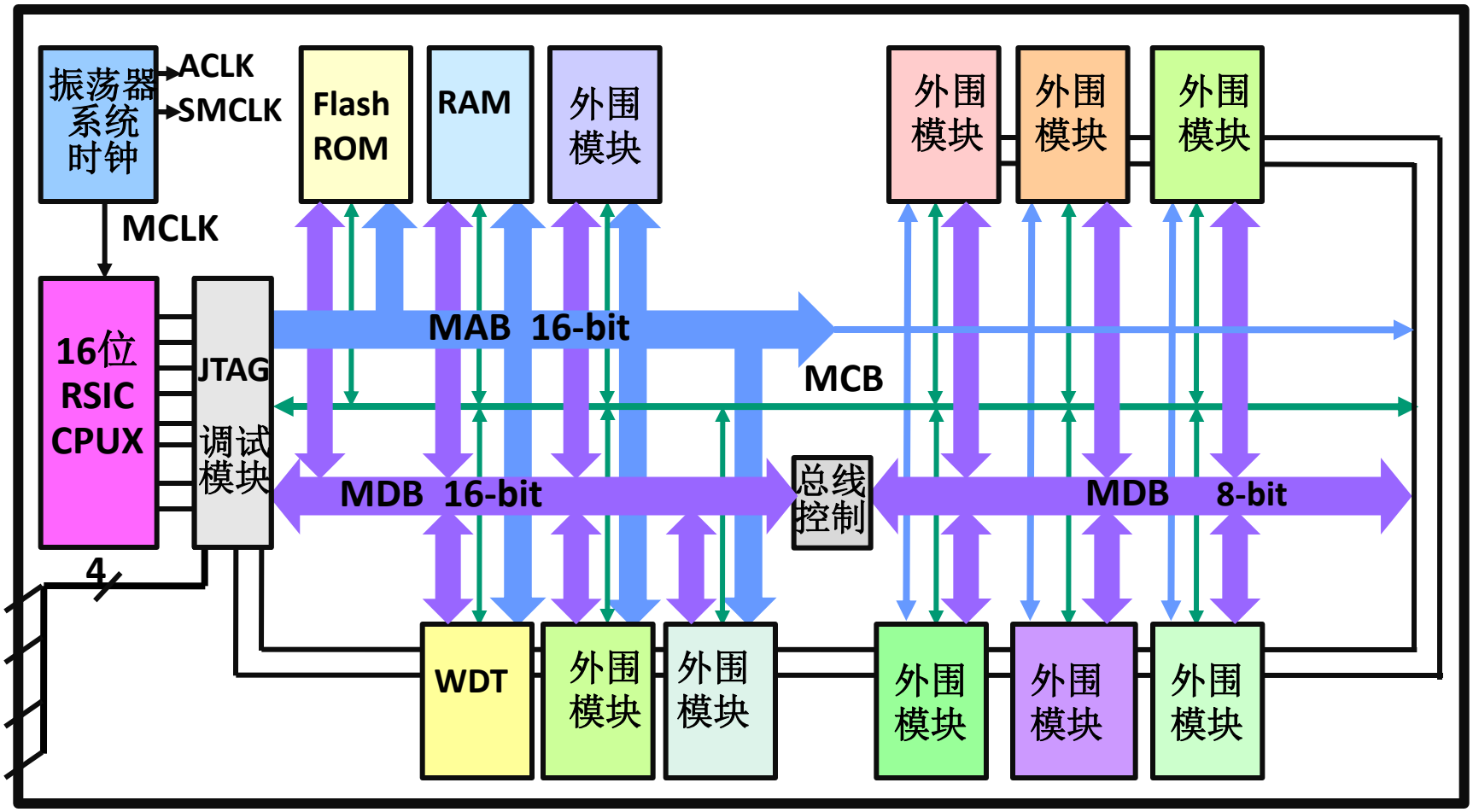
五、 MSP430端口的基本输入/输出

一、 MSP430 单片机概述

- 16-bit 精简指令集CPU (RISC CPU):
- 丰富的外围模块（peripherals模块, I/O接口）；

  外围模块I/O接口与存储器统一编址
- 灵活的时钟系统, 支持各种低功耗运行方式;
- 内嵌JTAG调试模块，方便开发调试

MSP430结构图



- 各模块通过地址(MAB)、数据(MDB)和控制(MCB)三大总线互连
- 外围模块I/O接口与存储器统一编址

二、MSP430 的CPU编程结构

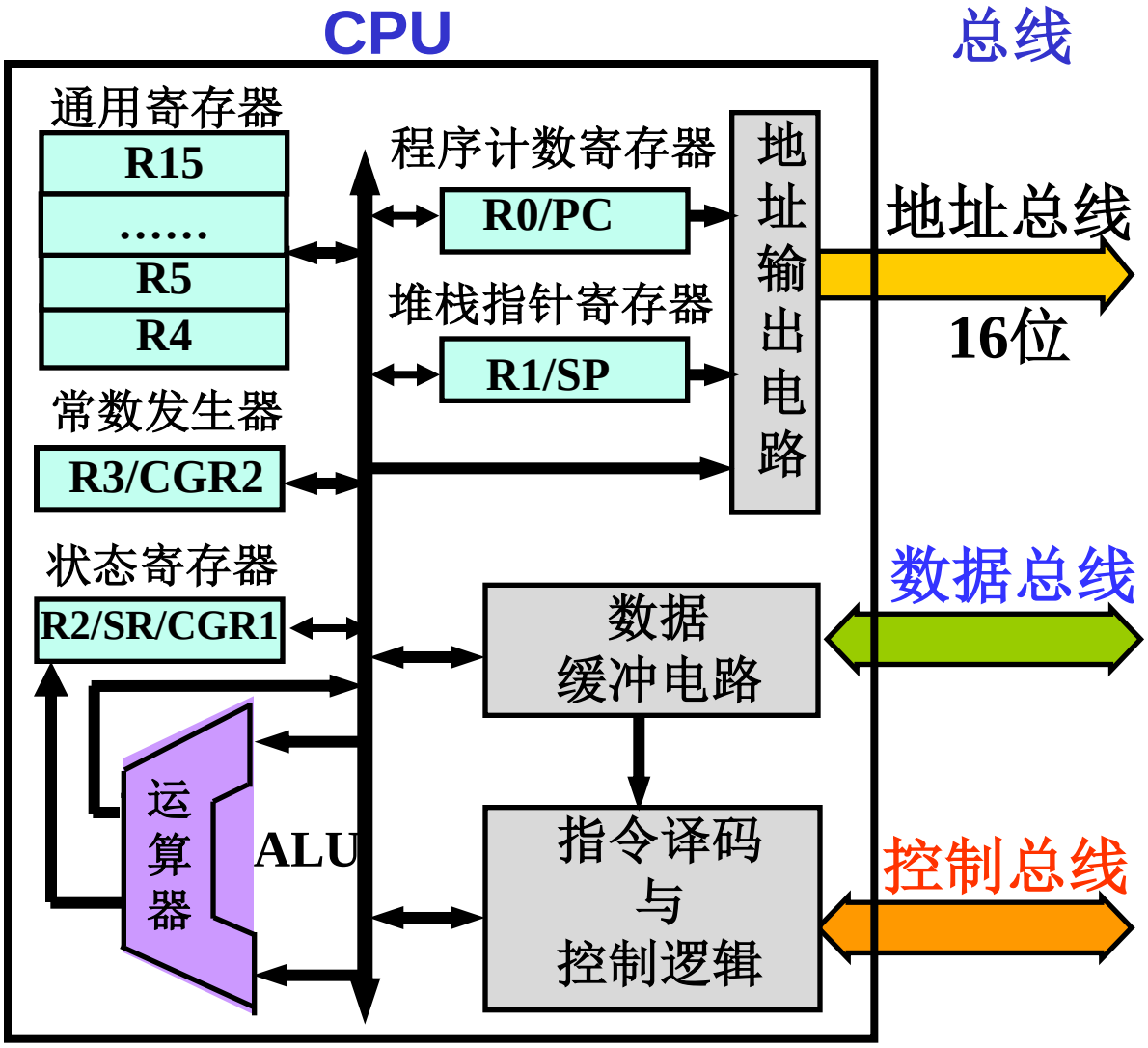
编程结构

- 指从程序员和使用者的角度看到的结构，
  与芯片内部的物理结构和实际布局有区别。

  编程结构便于程序员从软件角度了解微型计算机系统，
  而不必关心硬件实现的具体细节。

- **CPU编程结构主要掌握其内部寄存器组及其使用方法**

MSP430内CPU 原理示意图



16位RISC结构

精简指令集

7种寻址方式

27条核心指令

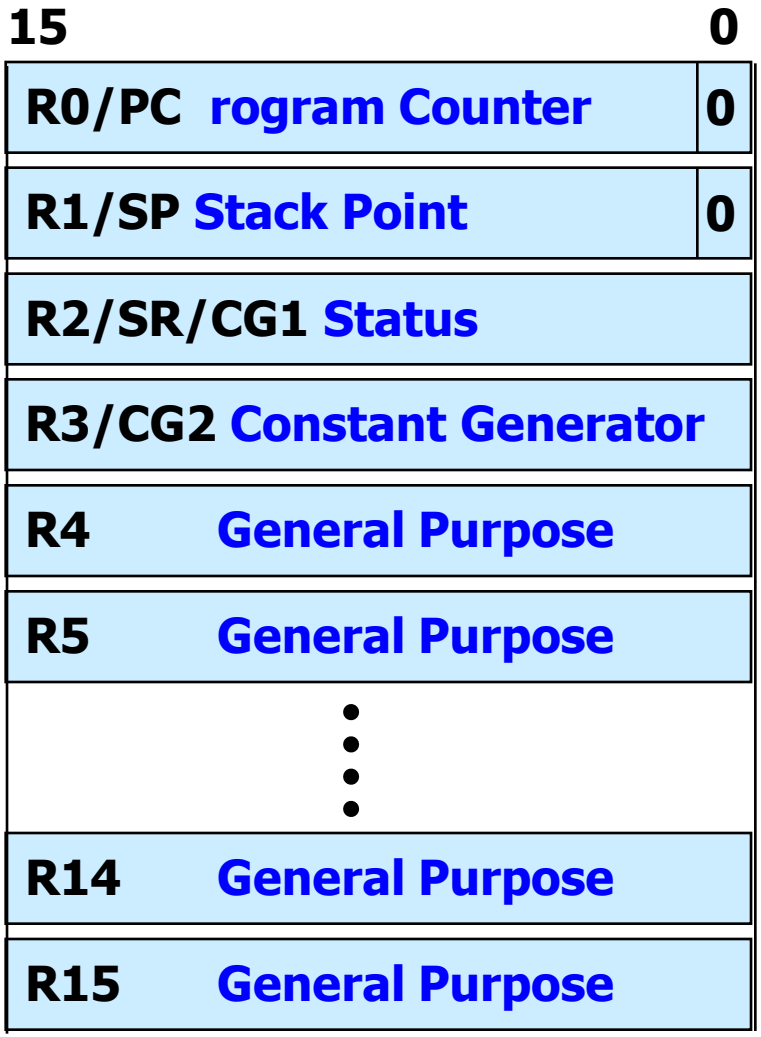
16个16位的寄存器

MSP430 CPU的编程结构

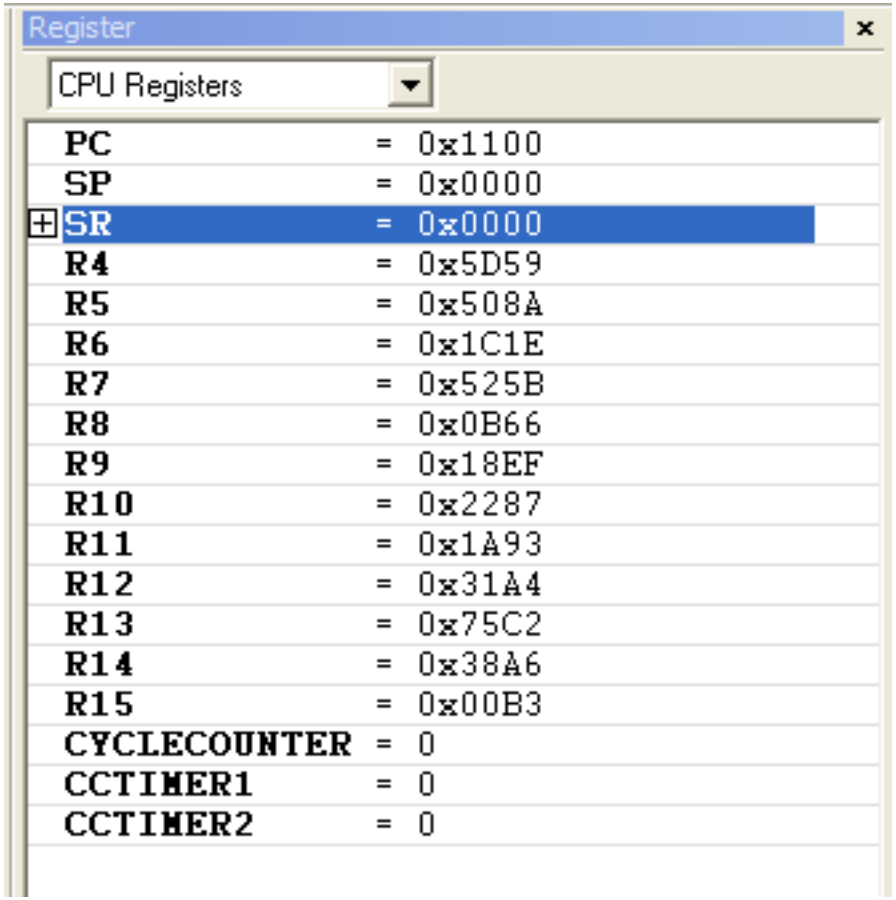
内含16个16位寄存器

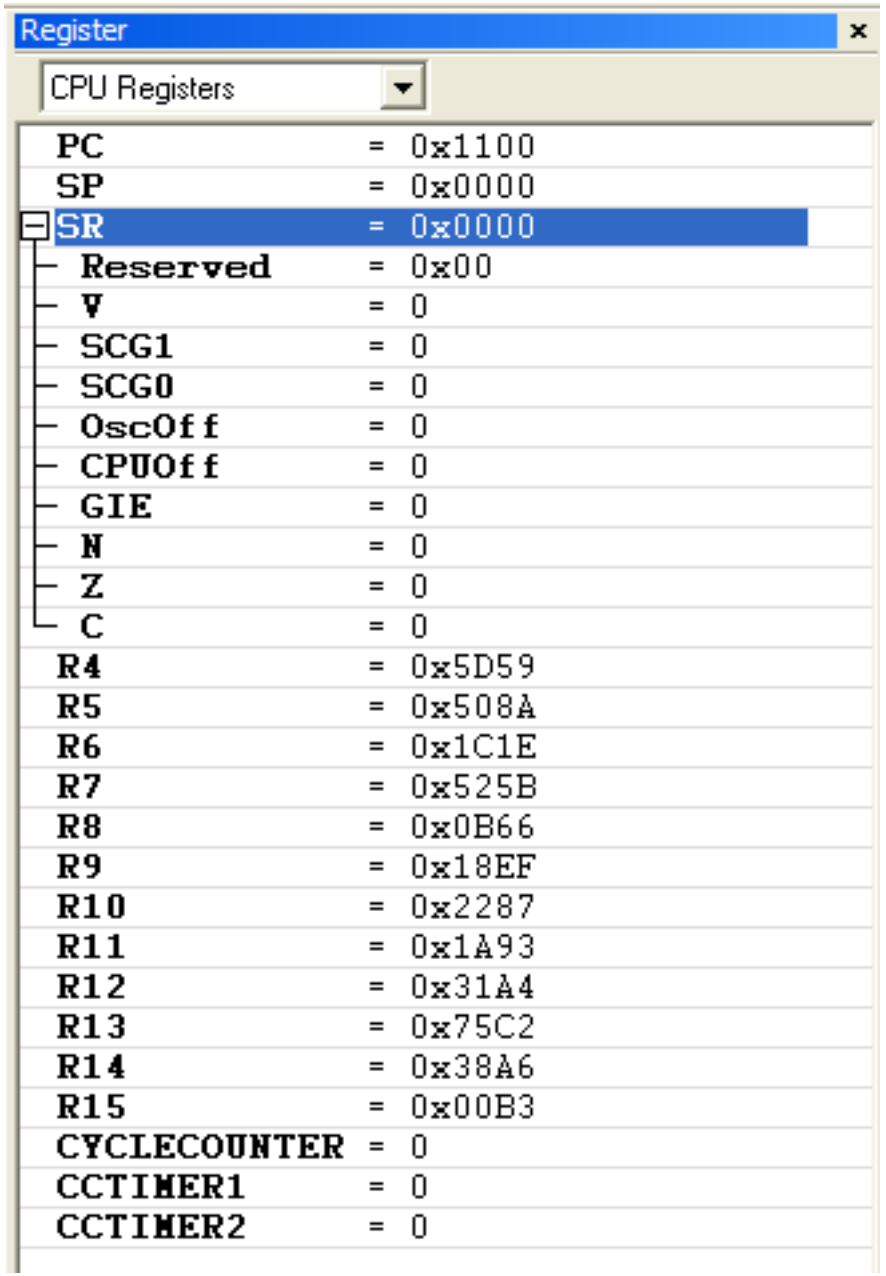
1. R0/PC 程序计数寄存器
2. R1/SP 堆栈指针寄存器
3. R2/SR/CGR1 状态寄存器
4. R3/CGR2 常数发生器
5. R4~R15 通用寄存器

MSP430 CPU 的寄存器组

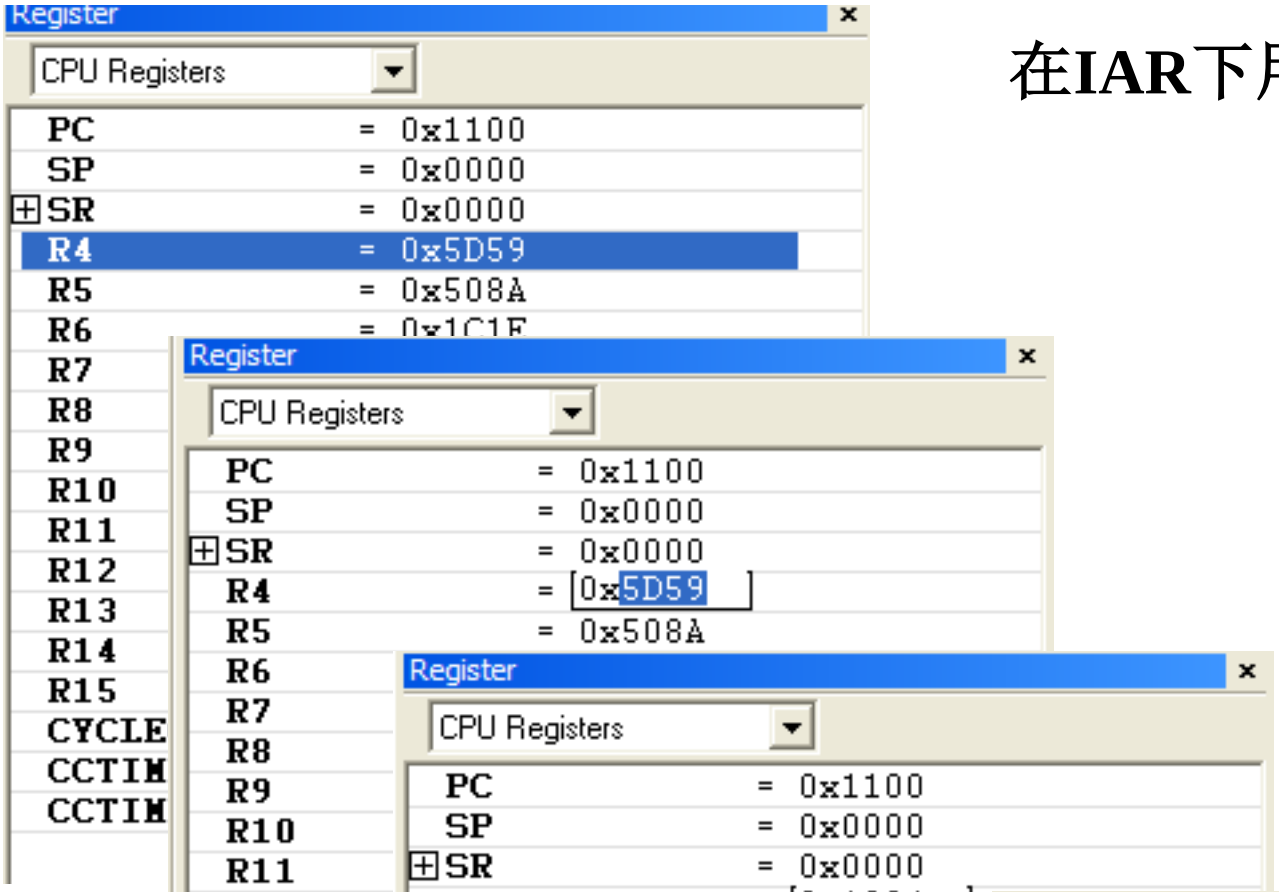


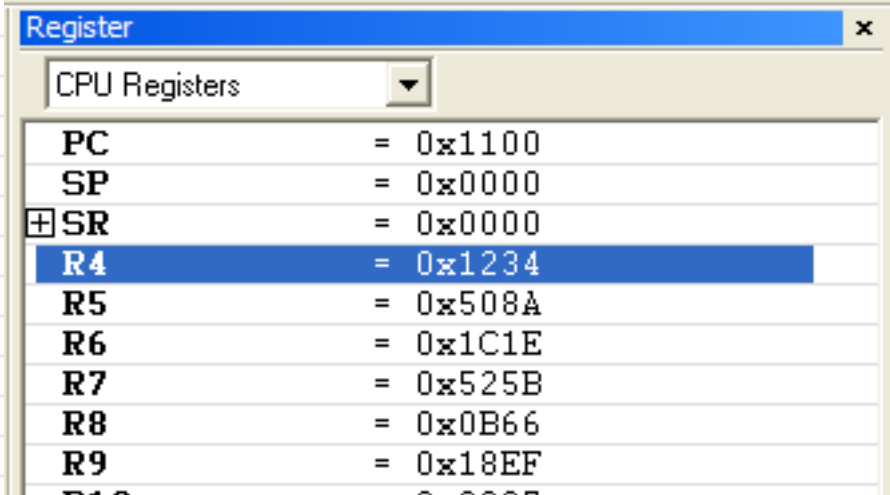
在IAR EW430下用View/ Registers 查看寄存器的内容





在IAR下用修改寄存器的内容





●状态寄存器SR (Status Register)

16位寄存器，目前用到其中的9位

存放指令执行过程的状态信息和对CPU工作的控制信息

15~9	8	7	6	5	4	3	2	1	0
保留	V	SCG1	SCG0	OSCoff	CPUoff	GIE	N	Z	C

C：进位标志 (Carry Bit)

反应加/减法运算过程的进位/借位信息

Z：零标志 (Zero Bit)

反应运算结果是否为0

N：负标志 (Negative Bit)

反应运算结果的符号位是否为1

V：溢出标志 (Overflow Bit)

反应加减算术运算结果是否超出带符号数范围

各状态标志的含义

15~9	8	7	6	5	4	3	2	1	0
保留	V	SCG1	SCG0	OSCoff	CPUoff	GIE	N	Z	C

C： 进位标志(Carry Bit)

在算术运算过程中，反应最高位是否产生进位/借位。

(最高位, 对字节操作指D7位 ,对字操作指D15位)

加法, 最高有效位有进位C＝1, 否则C＝0

减法, 最高有效位有借位C＝0, 否则C＝1

```
  1111 1111 B
+ 0000 0111 B
-------------
  0000 0110 B
```

C＝1

借位可理解为"非进位"：

	进位C
有借位	0
无借位	1

```
  0000 0000 B
- 0000 0010 B
-------------
  1111 1110 B
```

C＝0

各状态标志的含义(续)

15~9	8	7	6	5	4	3	2	1	0
保留	V	SCG1	SCG0	OSCoff	CPUoff	GIE	N	Z	C

Z：零标志(Zero Bit)

反应运算结果是否为0。

运算结果为 0 时，Z＝1，否则Z＝0

N：符号标志(Negative Bit)

反应运算结果的符号位。

对字节操作　　N＝D7

对字操作　　N＝D15

```
  1111 1111 B
+ 0000 0111 B
-------------
  0000 0110 B
```

Z = 0

N = 0

各状态标志的含义(续)

15~9	8	7	6	5	4	3	2	1	0
保留	V	SCG1	SCG0	OSCoff	CPUoff	GIE	N	Z	C

V： 溢出标志(Overflow Bit)

反应加/减运算结果是否超出带符号数表示的范围

超出，V＝1；否则V＝0

8位 -128 ~127

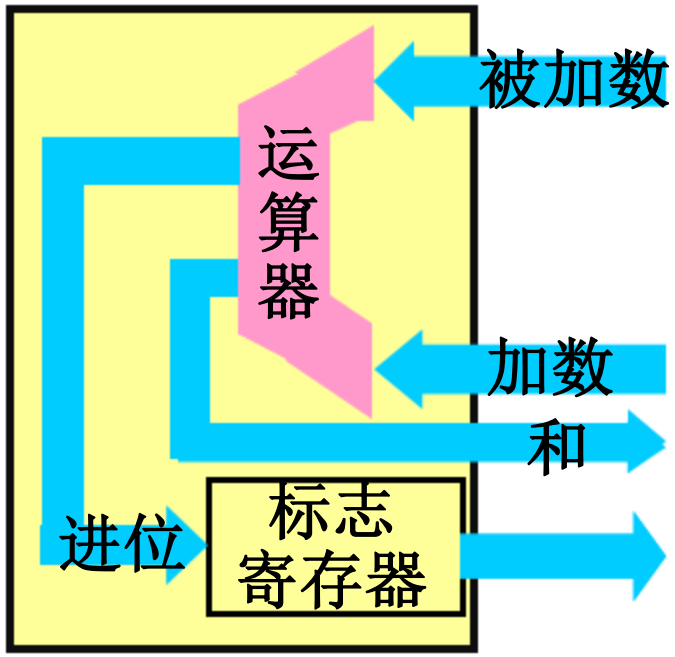
16位 -32768 ~ 32767

产生溢出的情况
正数+正数=负数
负数+负数=正数
正数-负数=负数
负数-正数=正数

溢出的判断方法参看讲义第1章第2节

例　8位二进制加法如下，给出各状态标志位的值

```
      1 0 1 1 0 1 0 1      被加数8位
  +   1 0 0 0 1 1 1 1      加数8位
进位  1   1 1 1 1 1 1
      ---------------
      0 1 0 0 0 1 0 0      和8位
```



最高位D7位产生进位:　C = _

相加的结果为44H, 不为0:　Z = _

结果的最高位为0:　N = _

负数+负数=正数：　V = _

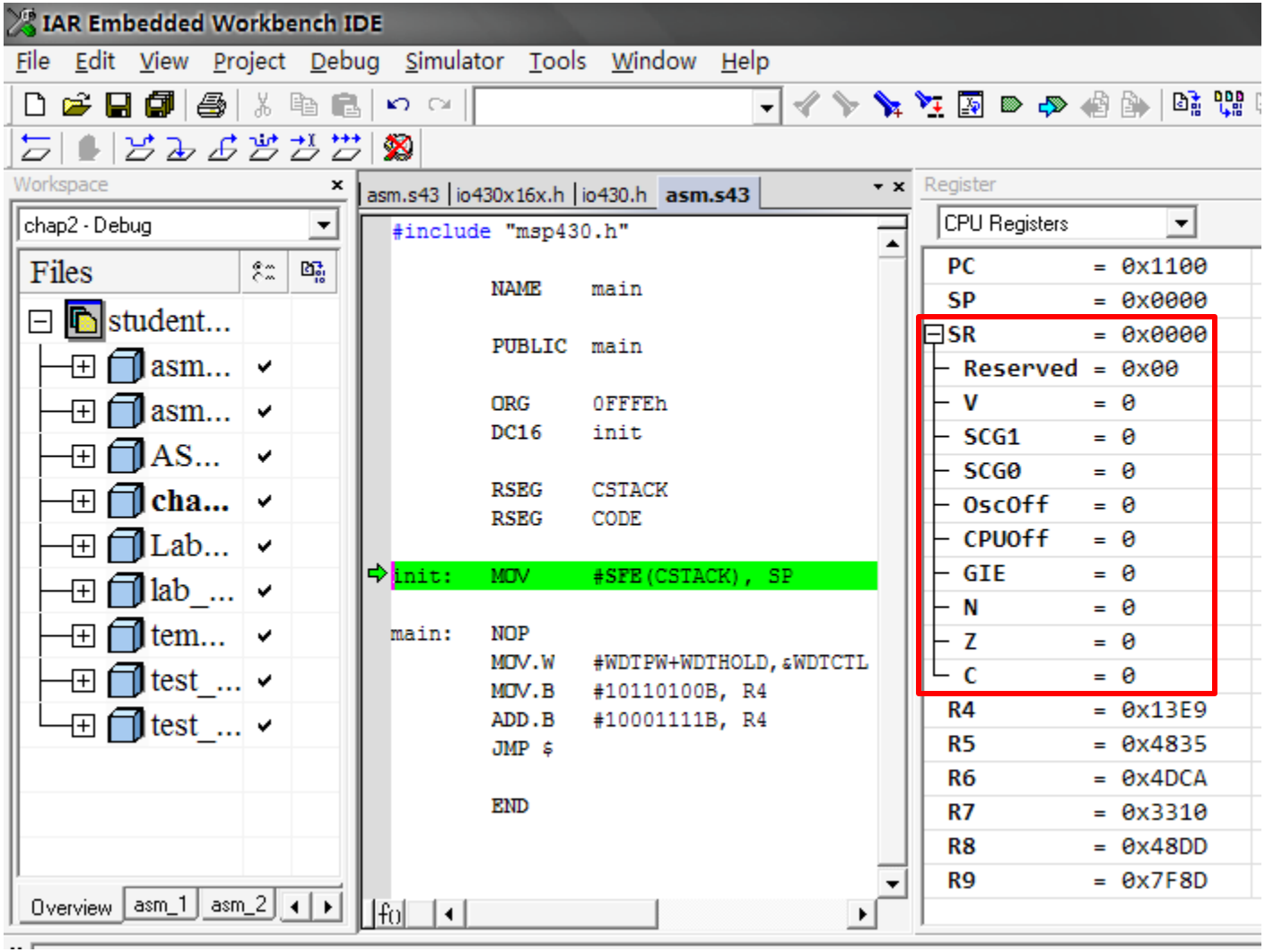
15~9	8	7	6	5	4	3	2	1	0
保留	V	SCG1	SCG0	OSCoff	CPUoff	GIE	N	Z	C

利用EW430查看标志位：View/registers/CPU Registers/SR

MOV.B #10110101B, R4

ADD.B #10001111B, R4

执行前的状态 C=0, Z=0, N=0, V=0

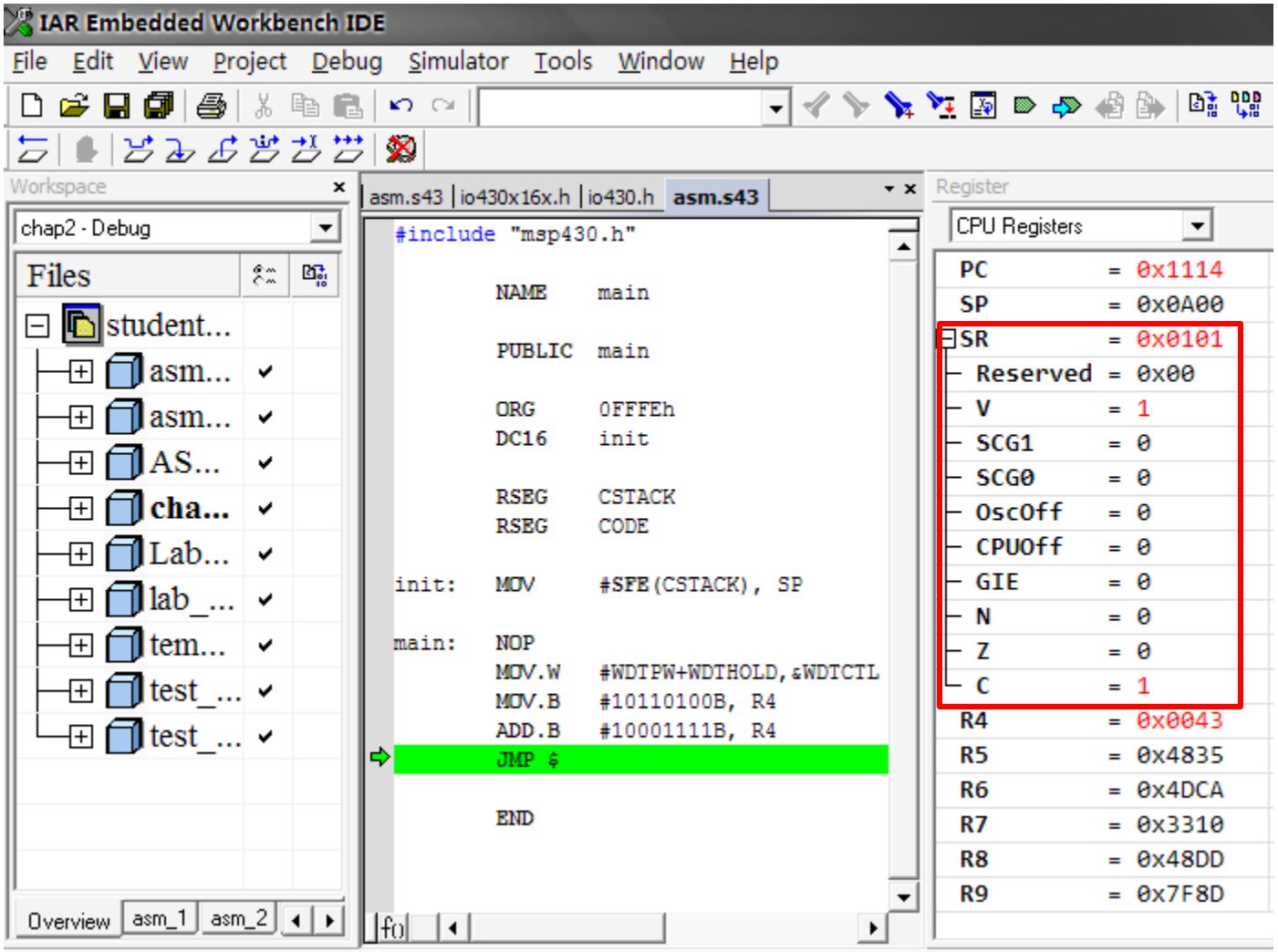


利用EW430查看标志位(续):View/registers/CPU Registers/SR

MOV.B #10110101B, R4

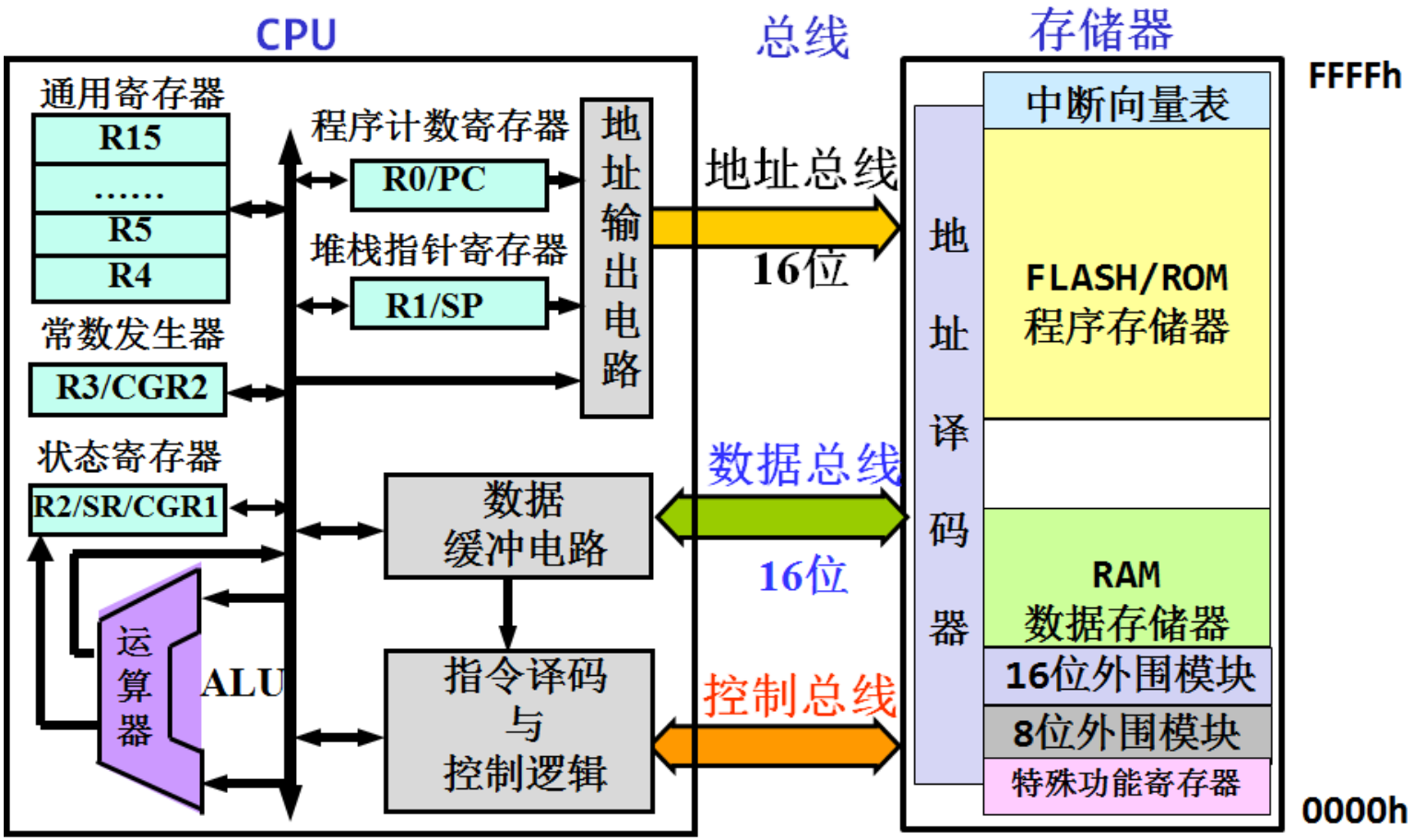
ADD.B #10001111B, R4

执行后的状态, C=1, Z=0, N=0, V=1



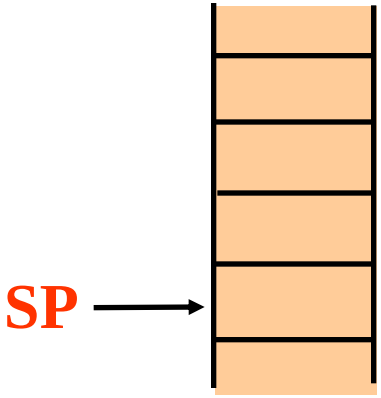
●指令计数寄存器PC（Program Counter)

- CPU到存储器取指令时，PC给出指令所在存储器单元的首址
  即: 取指令的地址由指令计数寄存器PC决定
- 取完一条指令，PC寄存器自动加上该指令长度，
  指向下一条指令



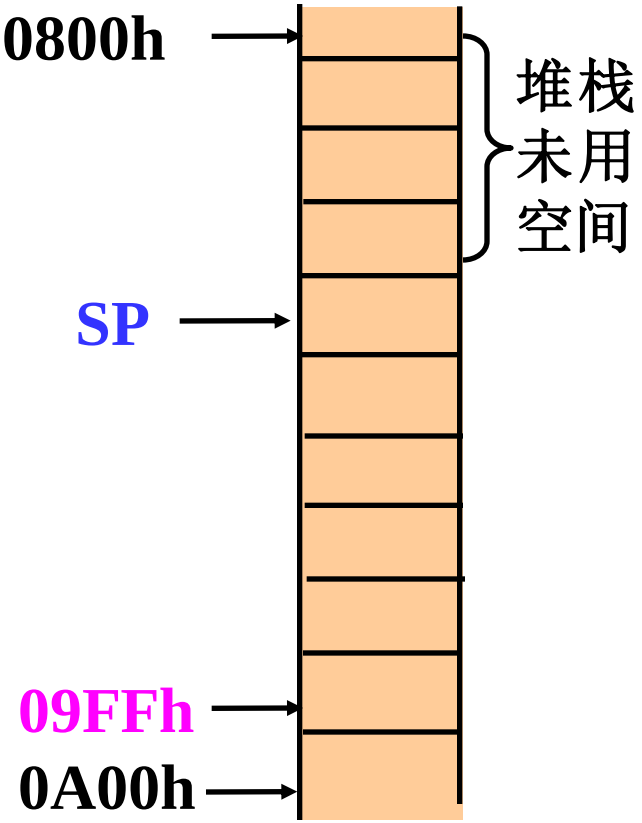
堆栈和堆栈指针寄存器SP

- 堆栈是一个存储区域。
  通常用于存放一些重要数据，
  如程序的地址、或是需要恢复的数据。



- 为方便数据的存放和恢复，
  设置专门的堆栈指针寄存器，指向堆栈中要操作的单元。
  SP → 堆栈指针寄存器 (stack pointer)

- 堆栈操作指令 PUSH 和 POP，
  对 SP 指向的存储器区域，
  以“后进先出”方式进行操作。



PUSH → 入栈操作　SP减小
POP → 出栈操作　SP增加

SP 指向栈顶，即堆栈的顶部，

- 编程员通过设置堆栈区域，
  利用堆栈操作，可方便的进行数据的存放和恢复。

MSP430程序上电第一条执行的指令通常是初始化SP：

```
……
MOV    #0x6400, SP
……
```

堆栈使用的场合

- 用堆栈保存恢复信息
- 子程序的调用、返回以及中断调用、返回
- 用堆栈传送数据
- 当局部变量较多时，利用堆栈存/取局部变量

三、MSP430X的CPU (CPUX) 编程结构

- MSP430X CPU地址总线 20位，

  数据总线16位，支持8位、16位和20位的内存访问；

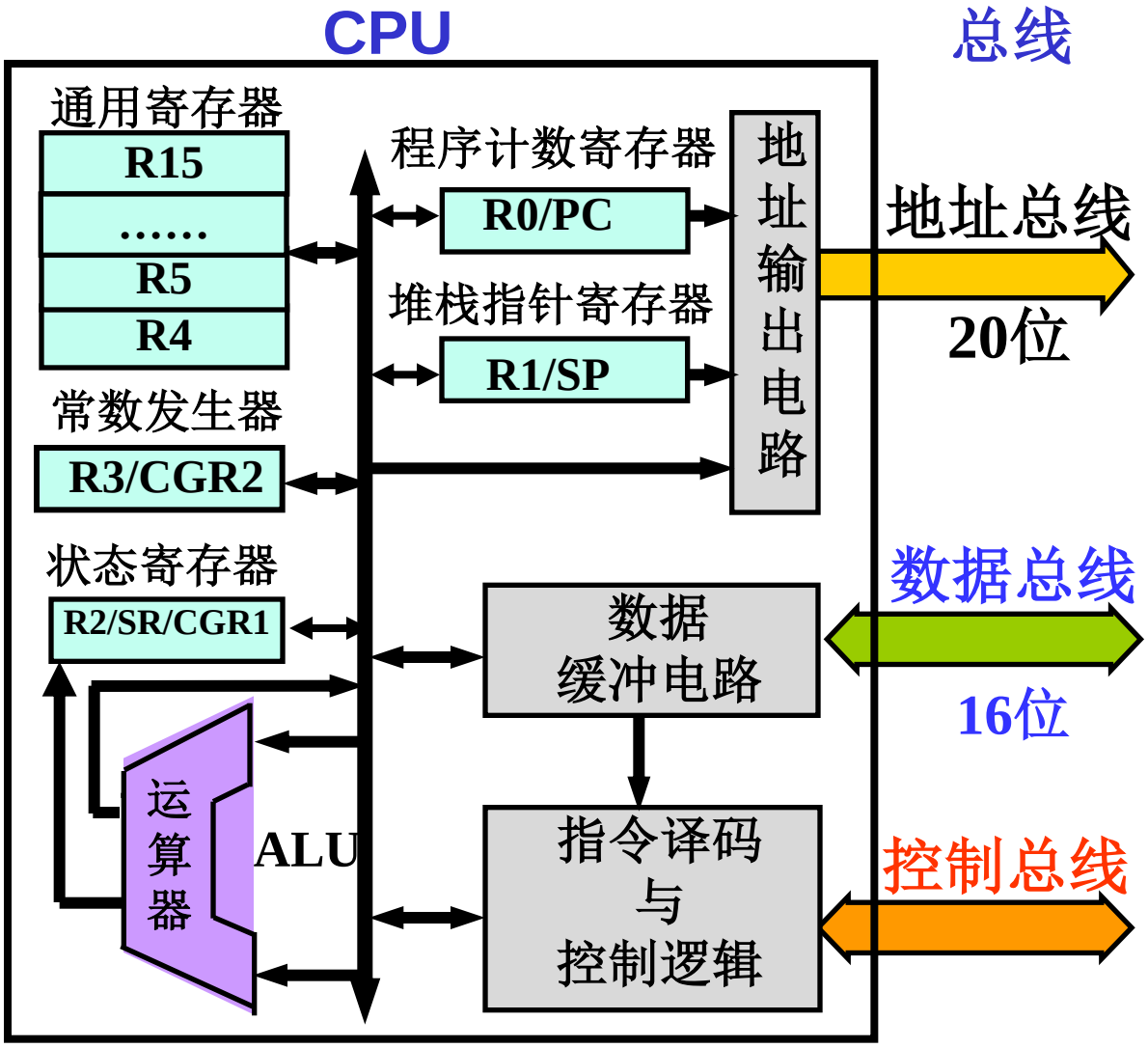
- 寻址范围从MSP430系列64kB扩展了到1MB;

- 7种寻址方式

- 指令系统与MSP430 CPU兼容

- 在MSP430 CPU基础上扩展了16条核心指令

MSP430内CPU 原理示意图



16位RISC结构

精简指令集

15个20位的寄存器

1个16位的寄存器

7种寻址方式

在MSP430 CPU基础上扩展了

16条核心指令

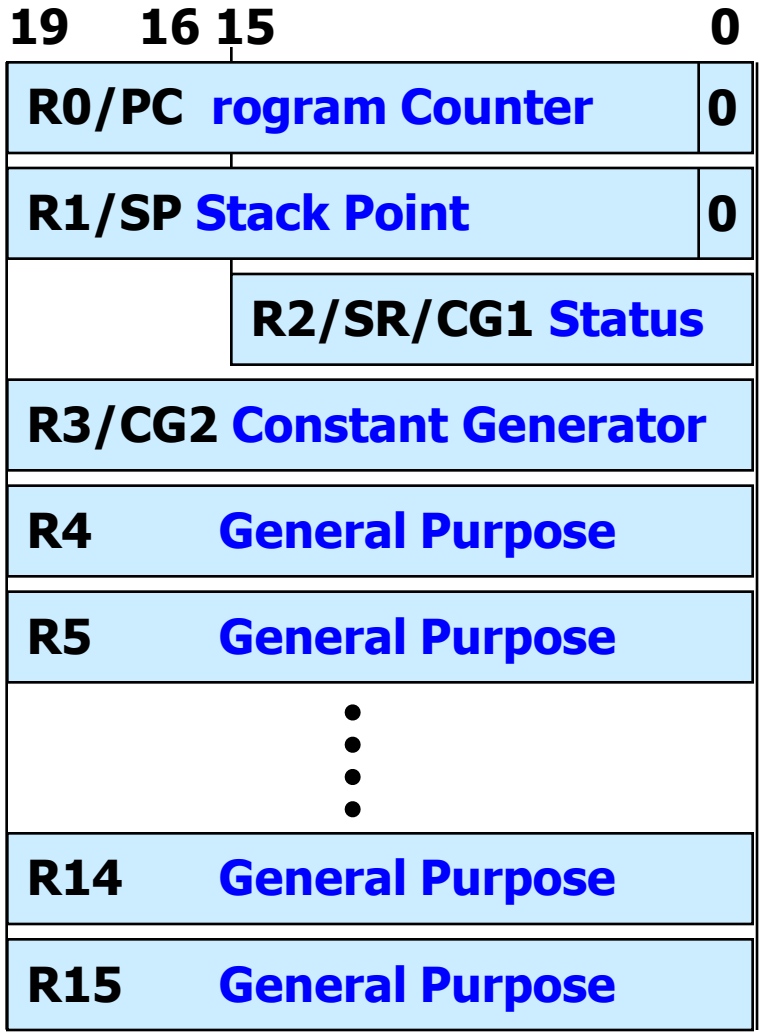
MSP430X CPU的编程结构

内含15个20位寄存器

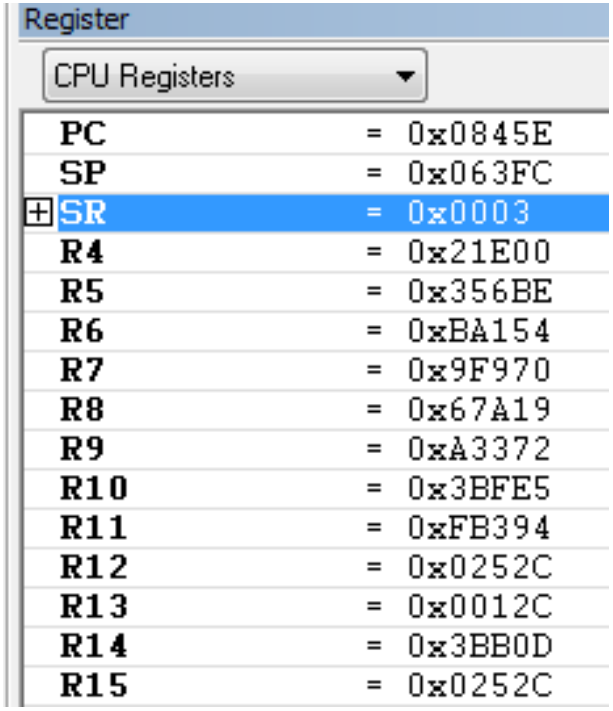
1个16位寄存器

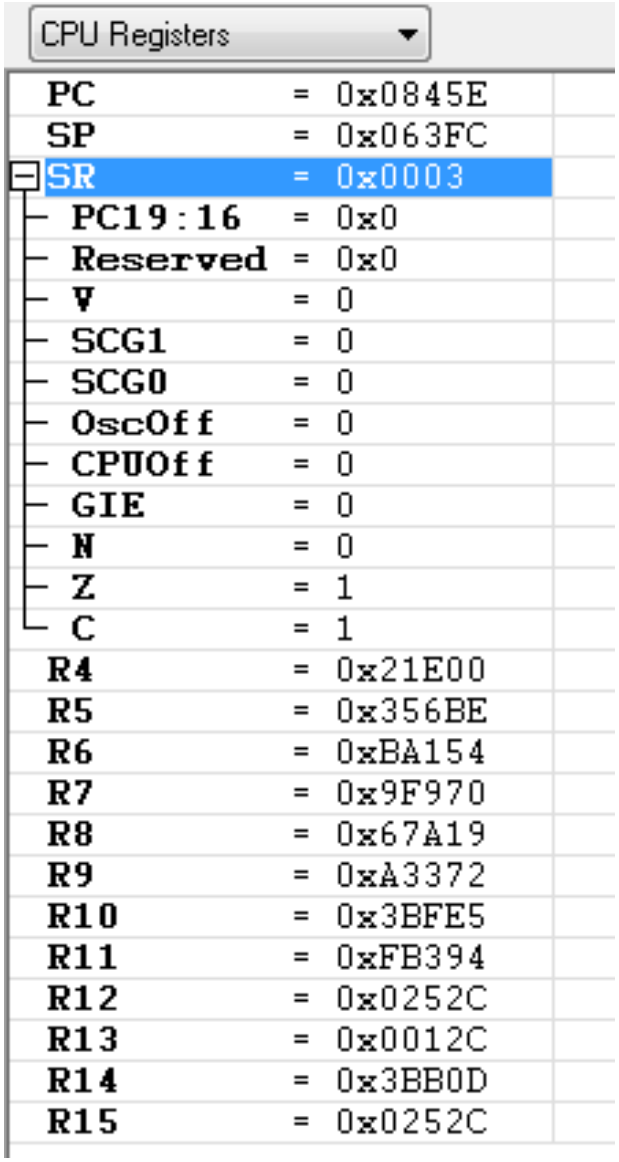
1. R0/PC 程序计数寄存器(20位)
2. R1/SP 堆栈指针寄存器(20位)
3. R2/SR/CGR1 状态寄存器(16位)
4. R3/CGR2 常数发生器(20位)
5. R4~R15 通用寄存器(20位)

寄存器组



在IAR EW430下用View/ Registers 查看寄存器的内容





●状态寄存器SR (Status Register)

与MSP 430 的CPU相同

16位寄存器，目前用到其中的9位

存放指令执行过程的状态信息和对CPU工作的控制信息

15~9	8	7	6	5	4	3	2	1	0
保留	V	SCG1	SCG0	OSCoff	CPUoff	GIE	N	Z	C

C：进位标志 (Carry Bit)

反应加/减法运算过程的进位/借位信息

Z：零标志 (Zero Bit)

反应运算结果是否为0

N：负标志 (Negative Bit)

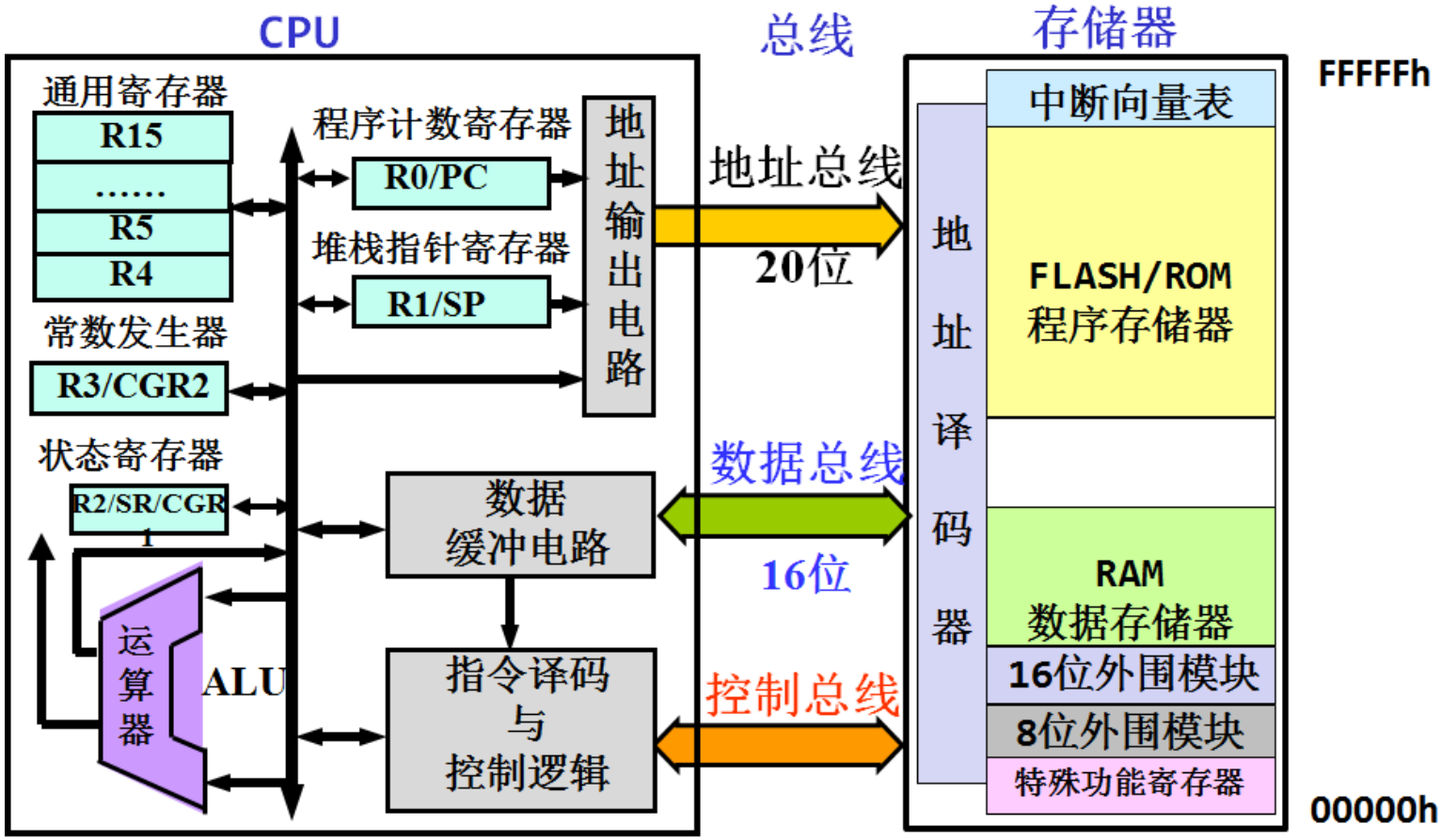
反应运算结果的符号位是否为1

V：溢出标志 (Overflow Bit)

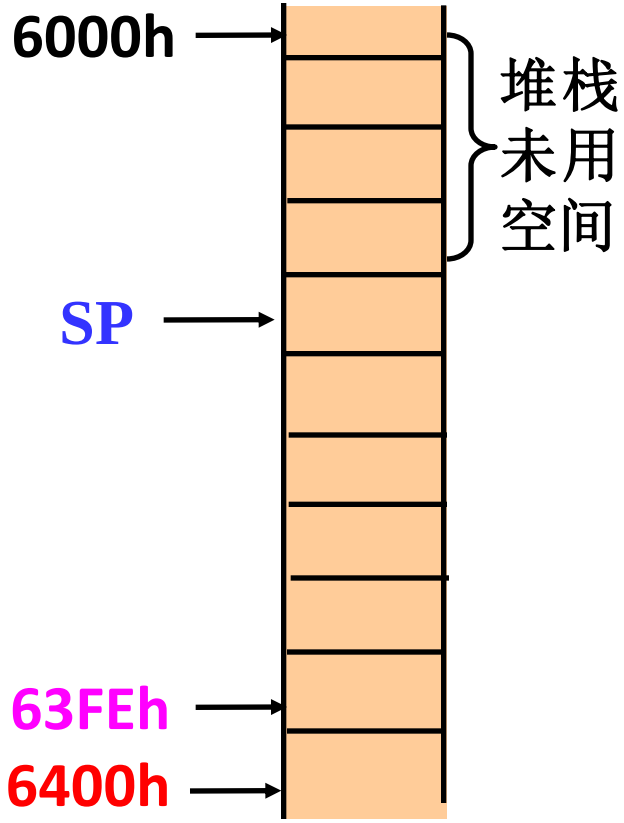
反应加减算术运算结果是否超出带符号数范围

●指令计数寄存器PC（Program Counter)

- CPU到存储器取指令时，PC给出指令所在存储器单元的首址
  即: 取指令的地址由指令计数寄存器PC决定
- 取完一条指令，PC寄存器自动加上该指令长度，
  指向下一条指令
- 寄存器PC 20位，CPU可寻址1MB空间的存储器



20位的堆栈指针寄存器SP



MSP430程序上电第一条执行的指令通常是初始化SP：

```
……
MOV    #0x6400, SP
……
```

四、MSP430的存储器组织结构

1. MSP430单片机的存储器结构

程序存储器（中断向量表、用户程序区)

数据存储器

外围模块寄存器（16位外围模块区、8位外围模块区）

2. MSP430单片机的存储器读写模式

1. MSP430的存储器结构

- 采用冯.诺依曼结构 (也称普林斯顿体系结构)

  程序(ROM)和数据(RAM)存储器使用一组地址和数据总线。

- 存储单元、I/O端口统一编址

  物理上完全分离的存储区域

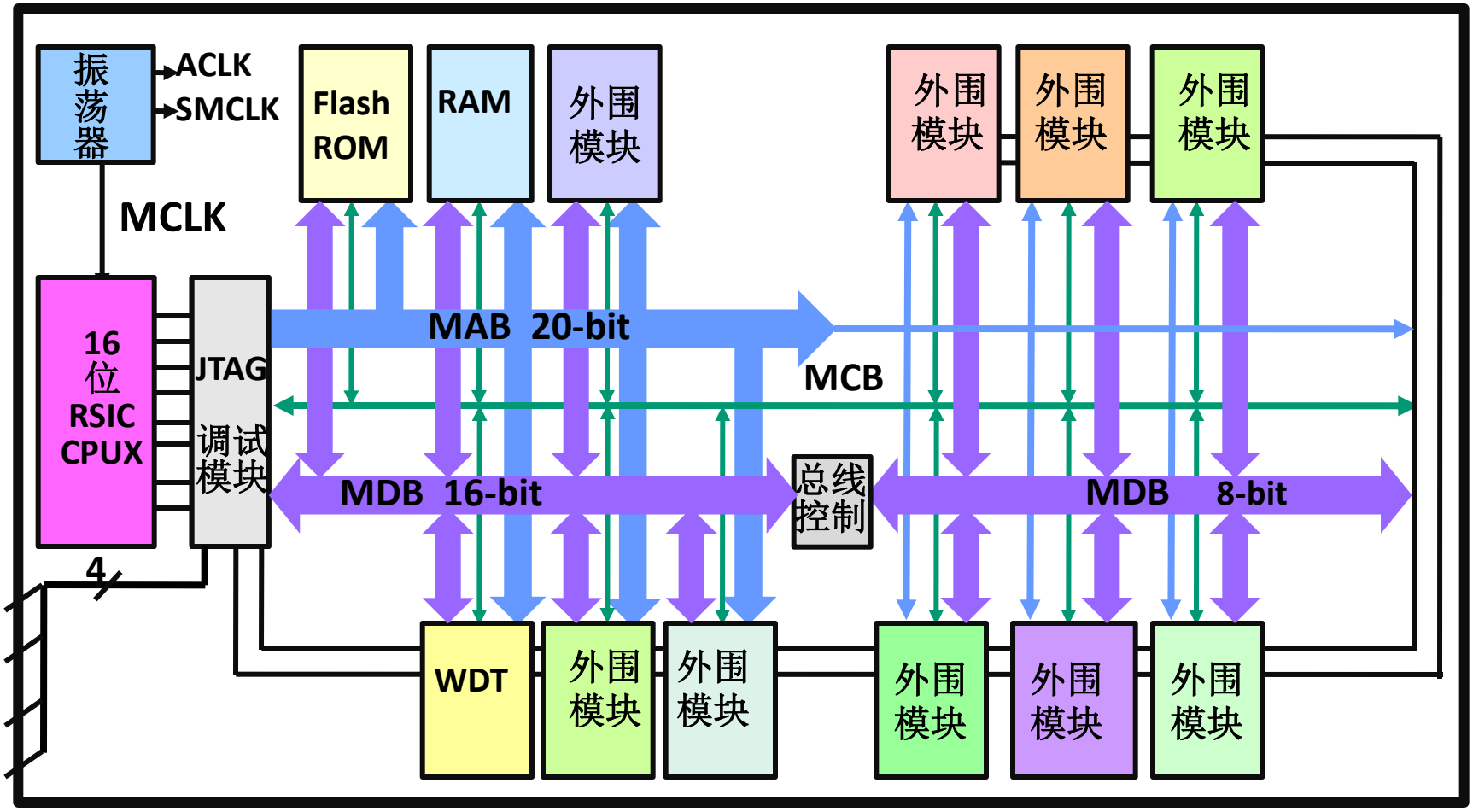
  如ROM/Flash、RAM、外围模块寄存器等

  被安排在0x00000~0xFFFFF范围的同一地址空间

- 使用一组地址、数据、控制总线以及相同的指令

  对这些空间进行访问/操作

1. MSP430的存储器结构



- 采用冯.诺依曼结构 (也称普林斯顿体系结构)
  程序(ROM)和数据(RAM)存储器使用一组地址和数据总线
- 外围模块I/O接口与存储器统一编址

MSP430F663x的存储器结构

	MSP430F6638-6	MSP430F6635-3	MSP430F6632-0
Flash ROM大小	256KB	192KB	128KB
中断向量地址	0FFFF~0FF80h		
代码存储器	47FFF-08000h	37FFF-08000h	27FFF-08000h
RAM	063FF-02400h		
USB RAM	023FF-01C00h		
Info mem(flash)	019FF-01800h		
BSL mem(Flash)	017FF-01000h		
外围模块	00FFF-00000h		

- 不同型号MCU的存储器ROM和RAM容量不一样，
- 代码ROM的低端地址都是08000H,
- RAM起始地址都是063FF-02400h,大小都是16KB

  RAM的末地址 = 该器件数据RAM 容量 + 0200H-1；
- 中断向量地址、外围模块地址地址范围相同

MSP430F663x 的存储器结构

		MSP430F6636 MSP430F6633 MSP430F6630	MSP430F6637 MSP430F6634 MSP430F6631	MSP430F6638 MSP430F6635 MSP430F6632
Memory (flash) Main: interrupt vector	Total Size	128KB 00FFFFh–00FF80h	192KB 00FFFFh–00FF80h	256KB 00FFFFh–00FF80h
Main: code memory	Bank 3	N/A	N/A	64 KB 047FFF-038000h
	Bank 2	N/A	64 KB 037FFF-028000h	64 KB 037FFF-028000h
	Bank 1	64 KB 027FFF-018000h	64 KB 027FFF-018000h	64 KB 027FFF-018000h
	Bank 0	64 KB 017FFF-008000h	64 KB 017FFF-008000h	64 KB 017FFF-008000h
RAM	Sector 3	4 KB 0063FFh–005400h	4 KB 0063FFh–005400h	4 KB 0063FFh–005400h
	Sector 2	4 KB 0053FFh–004400h	4 KB 0053FFh–004400h	4 KB 0053FFh–004400h
	Sector 1	4 KB 0043FFh–003400h	4 KB 0043FFh–003400h	4 KB 0043FFh–003400h
	Sector 0	4 KB 0033FFh–002400h	4 KB 0033FFh–002400h	4 KB 0033FFh–002400h
USB RAM(3)	Size RAM	2KB 0023FFh-001C00h	2KB 0023FFh-001C00h	2KB 0023FFh-001C00h
Information memory (flash)	Info A	128 B 0019FFh–001980h	128 B 0019FFh–001980h	128 B 0019FFh–001980h
	Info B	128 B 00197Fh–001900h	128 B 00197Fh–001900h	128 B 00197Fh–001900h
	Info C	128 B 0018FFh–001880h	128 B 0018FFh–001880h	128 B 0018FFh–001880h
	Info D	128 B 00187Fh–001800h	128 B 00187Fh–001800h	128 B 00187Fh–001800h
Bootstrap loader (BSL) memory (flash)	BSL 3	512 B 0017FFh–001600h	512 B 0017FFh–001600h	512 B 0017FFh–001600h
	BSL 2	512 B 0015FFh–001400h	512 B 0015FFh–001400h	512 B 0015FFh–001400h
	BSL 1	512 B 0013FFh–001200h	512 B 0013FFh–001200h	512 B 0013FFh–001200h
	BSL 0	512 B 0011FFh–001000h	512 B 0011FFh–001000h	512 B 0011FFh–001000h
Peripherals	Size	4KB 000FFFh–000000h	4KB 000FFFh–000000h	4KB 000FFFh–000000h

(1) N/A = Not available.
(2) Backup RAM is accessed via the control registers BAKMEM0, BAKMEM1, BAKMEM2, and BAKMEM3.
(3) USB RAM can be used as general purpose RAM when not used for USB operation.

2. MSP430的存储器读写模式

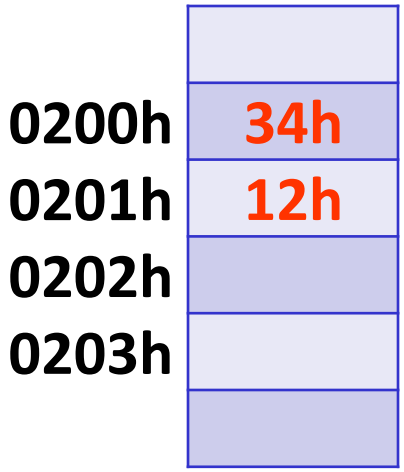
1个存储器单元可存放1个字节数据

当往存储器写入1个字的数据时，

有两种存储器读写模式

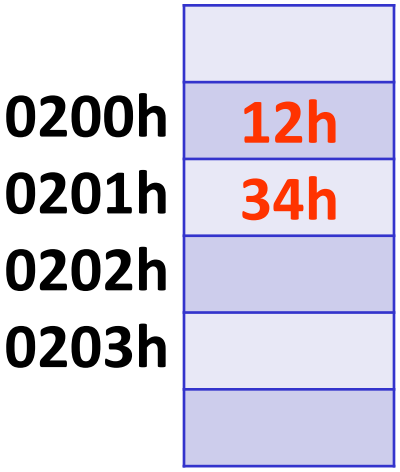
- 小端模式(Little Endian):
  数据的低字节存放在内存低地址中，
  高字节存放在高地址中，即"高高低低"原则

小端模式



- 大端模式(Big Endian):
  数据的低字节存放在内存高地址中，
  高字节存放在低地址中,即"高低低高"原则

大端模式



例 将#1234h写入0200h开始的存储单元中

MSP430的存储器读写模式

采用小端模式 (Little Endian):

小端模式

当往内存写一个字数据时，
写入规则是：
低字节到低地址单元(偶单元)
高字节到高地址单元(奇单元)

注意: 字操作必须从偶地址开始， 否则执行过程中报错"A word access on odd address"

例 ： MOV #1234h, &2401h

例 已知上电或手动复位后，

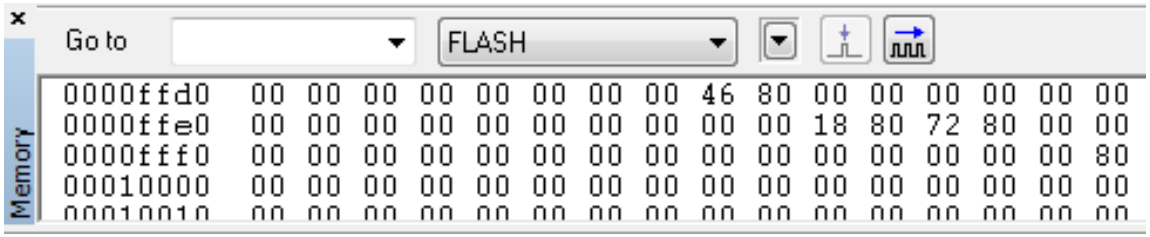
MSP430从中断向量表FFFFh~FFFEh位置，

获取1个字的内容来初始化程序计数器PC，

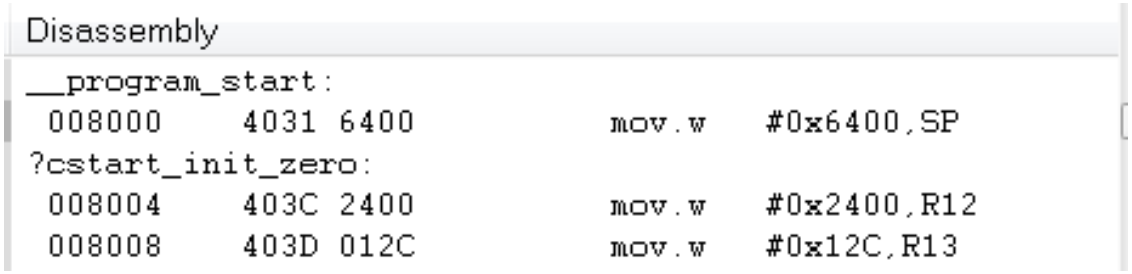
假设当前内存的内容如下图所示，问：

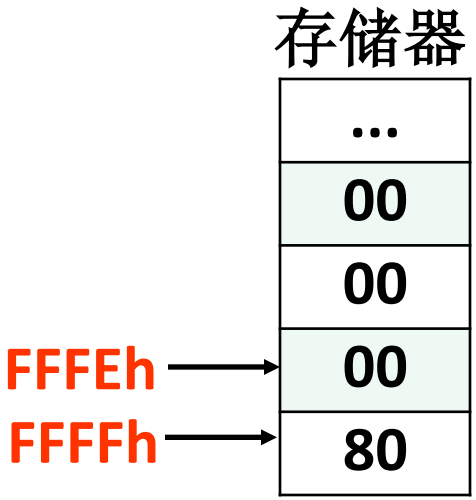
复位后，PC从什么地址开始取指令?

Reset CPU后用IAR的View/ Memory查看内存



用View/ disassemly查看程序反汇编代码





五、MSP430的基本输入/输出端口(I/O ports)

- MSP430有丰富的端口资源，
  各产品因型号不同，所含端口资源不同

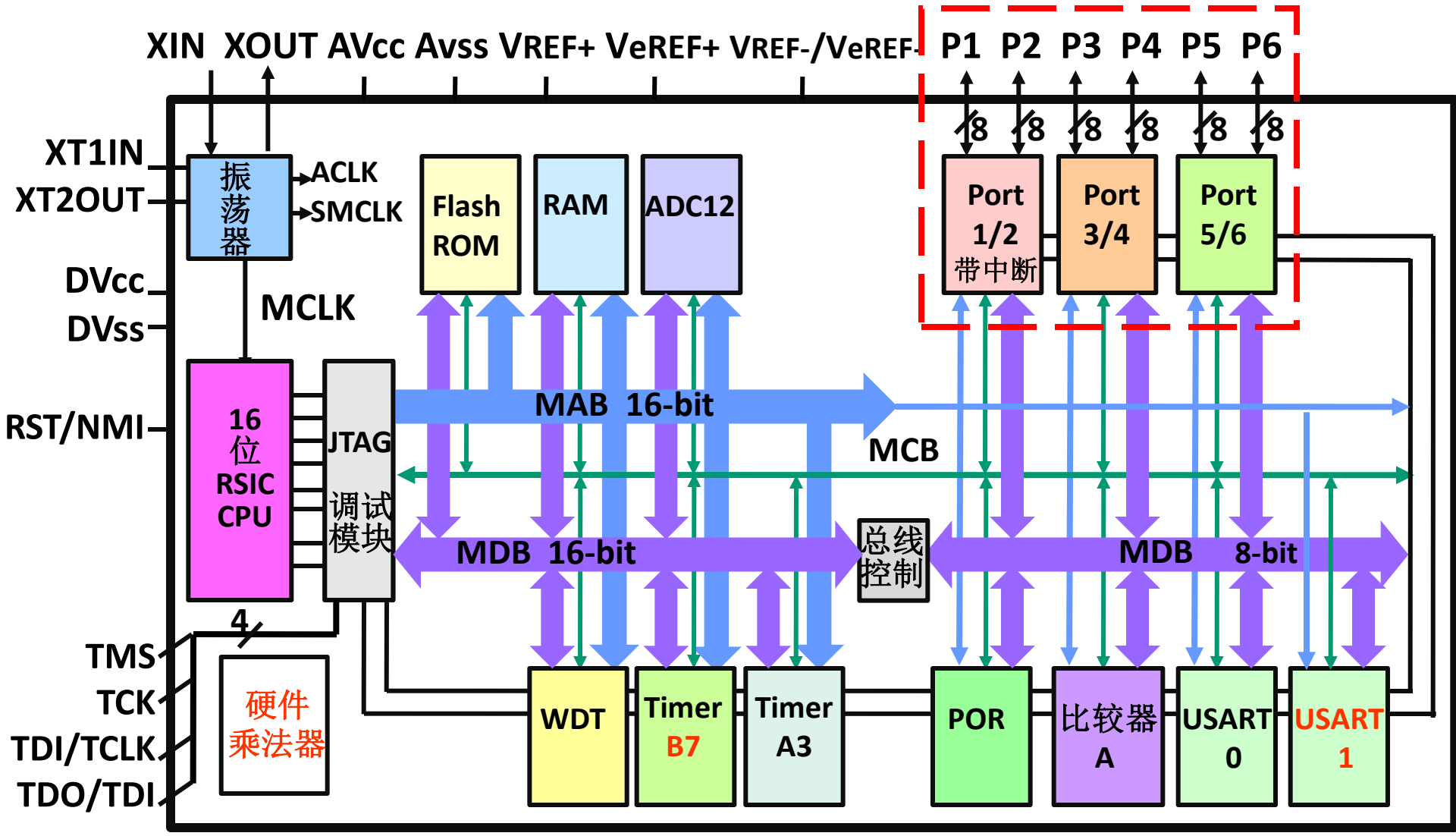
  MSP430x13x/14x/15x/16x有6个端口P1~P6，共48根I/O引脚
  MSP430x166 3x有10个端口P1~P9，PJ，共74根I/O引脚

- 目前MSP430系列单片机的总线不对外开放，
  I/O端口的引脚是MCU对外进行输入/输出的重要通道

- MSP430系统中操作I/O端口的方法与操作存储器单元相同，
  只是两者的地址不同

- 各基本输入/输出端口可通过控制寄存器设置方向、功能，
  且都可以位操作，即每位都可单独配置

MSP430F149有6个基本输入/输出端口，简称I/O端口，P1~P6



13x/14x/16x有6个I/O端口

每个端口对外有8根引脚

单片机

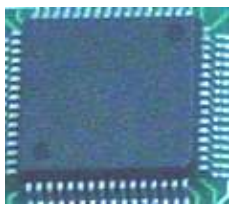
共有64个引脚

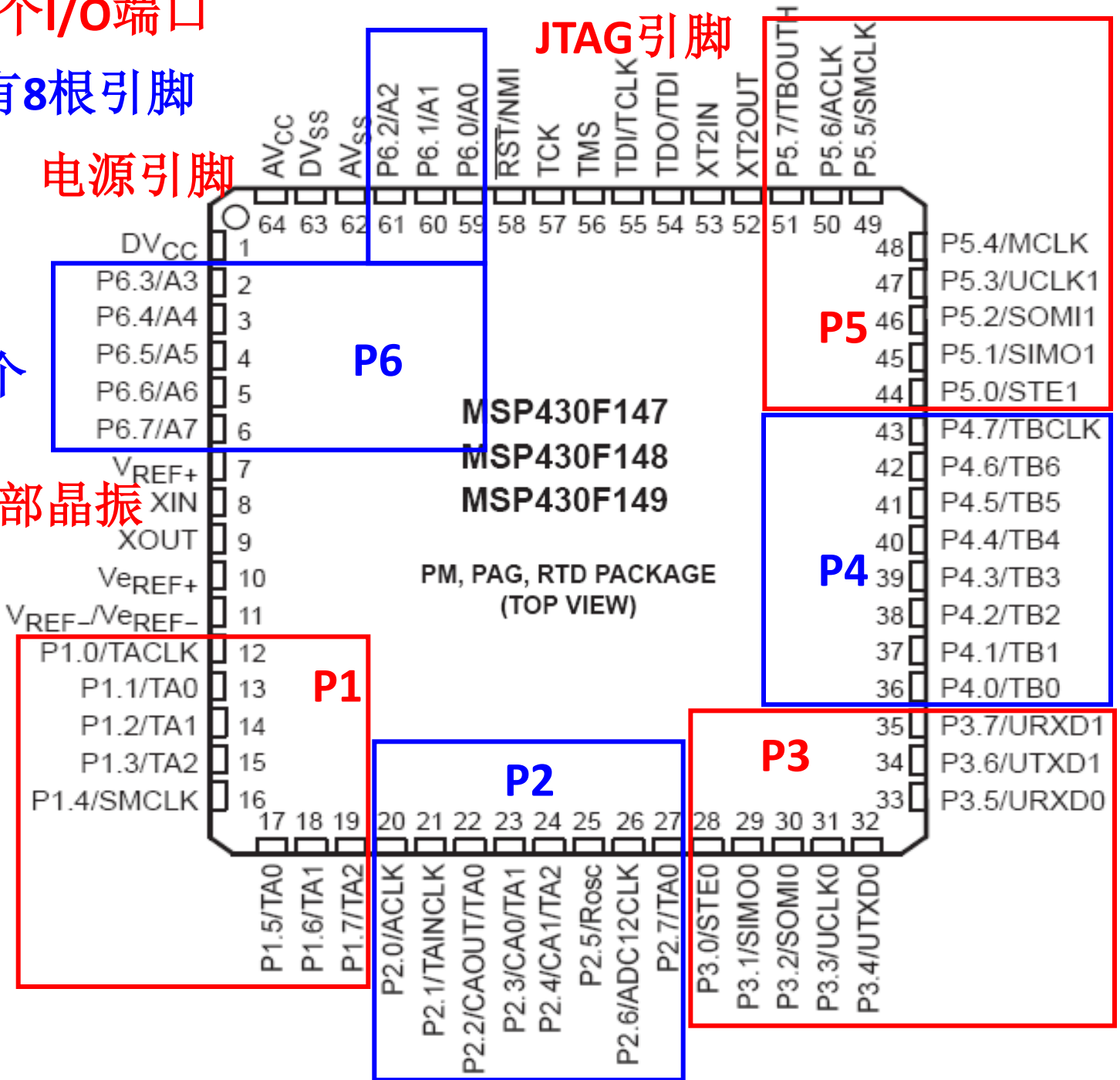
其中I/O引脚48个

与MCU内

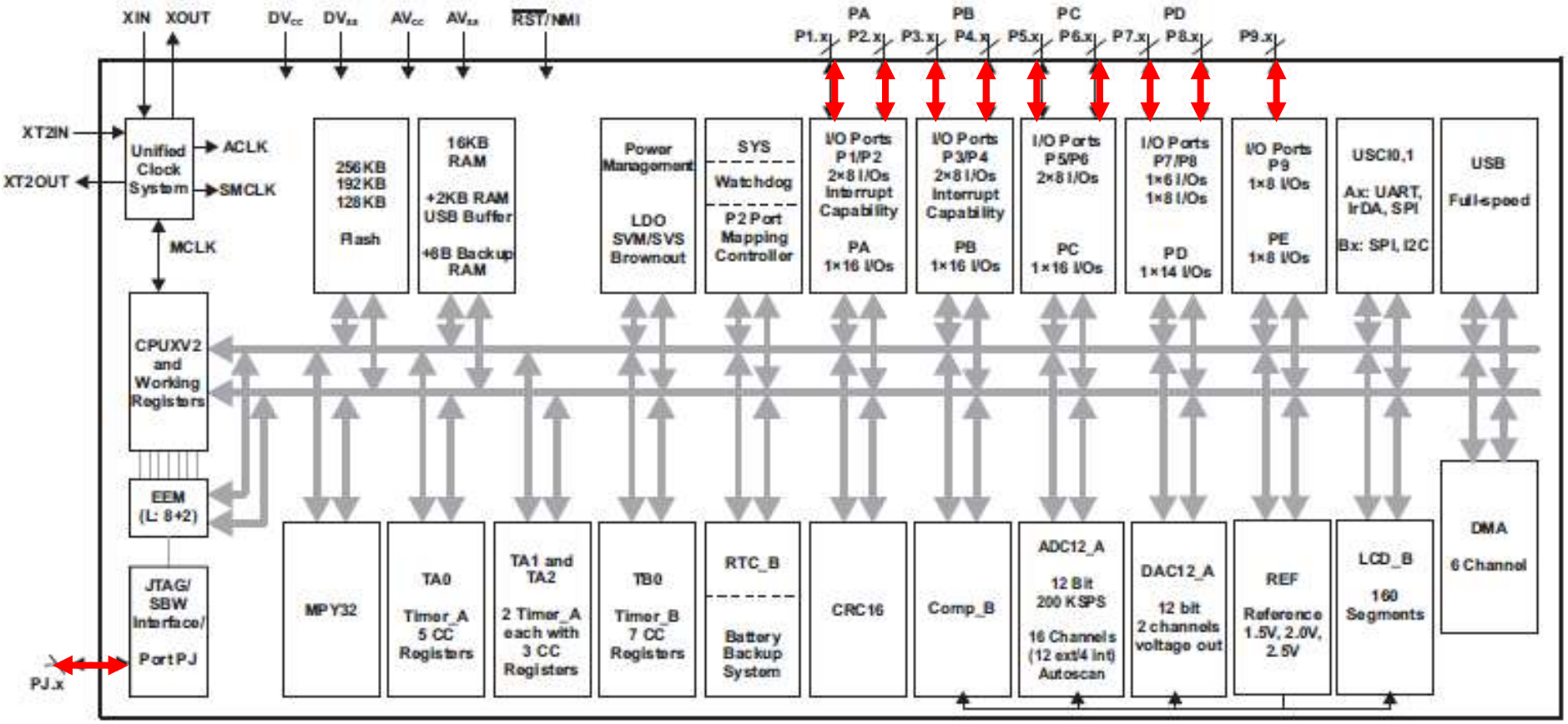
其他模块共用

(引脚复用)





Functional Block Diagram, MSP430F6638, MSP430F6637, MSP430F6636



MSP430F663x 有 P1~P9, PJ共10个I/O 端口， 共74根I/O引脚

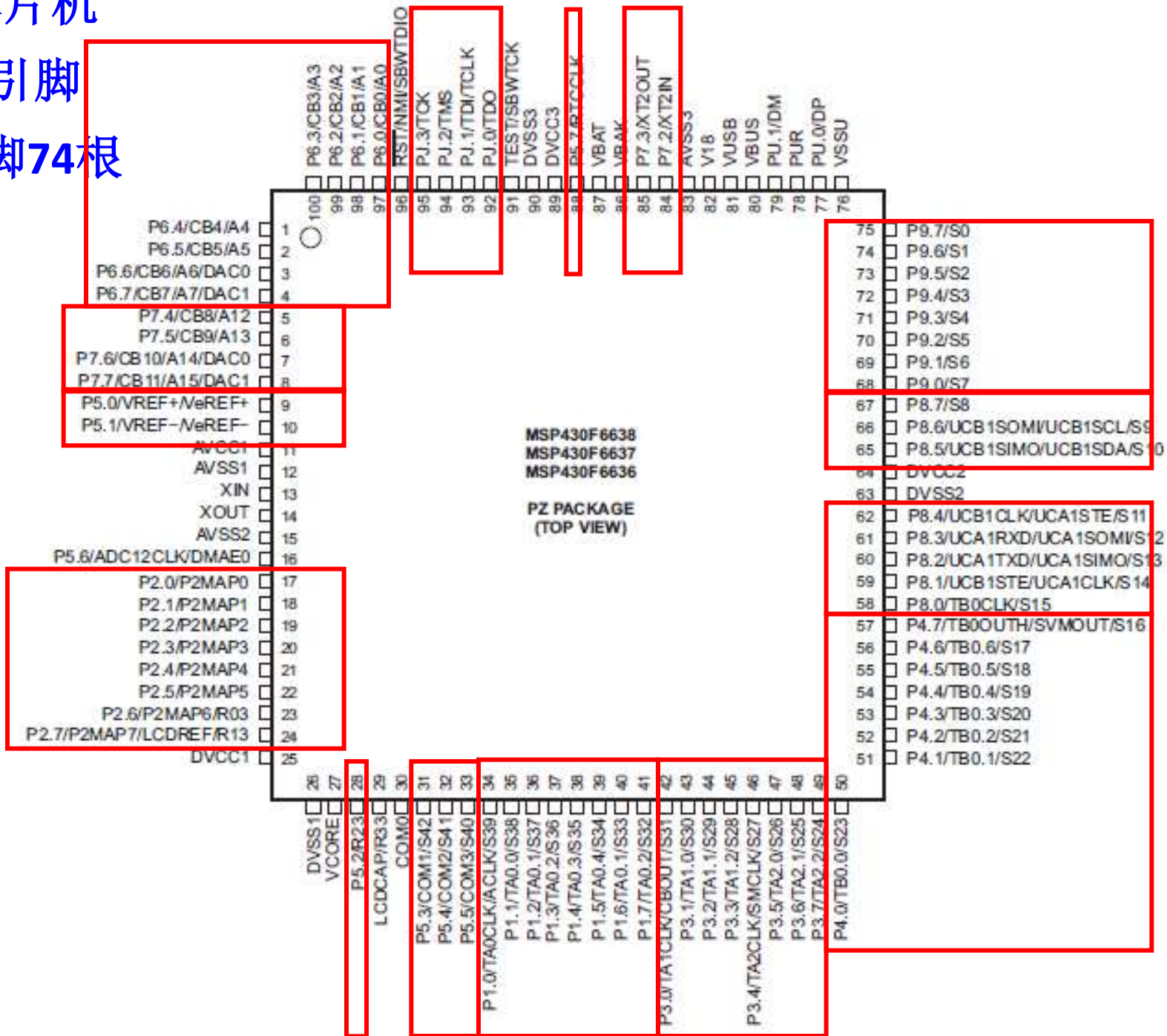
其中 P1~P6， P8， P9每个端口8根引脚

P7 有6根引脚， PJ有4根引脚

MSP6638单片机

共有100个引脚

其中I/O引脚74根



与每个I/O引脚输入/输出功能有关的I/O寄存器

- PxSEL功能选择寄存器 ：P1SEL， P2SEL，……， PJSEL
- PxDIR方向选择寄存器 ：P1DIR， P2DIR，……， PJDIR
- PxOUT输出数据寄存器： P1OUT, P2OUT, ……， PJOUT
- PxIN输入数据寄存器 ： P1IN， P2IN， ……， PJIN
- PxREN拉电阻使能寄存器： P1REN， P2REN， ……， PJREN
- PxDR 输出增强寄存器： P1DR, P2DR, ……, PJDR

▲x可为1~J, 由具体的单片机确定

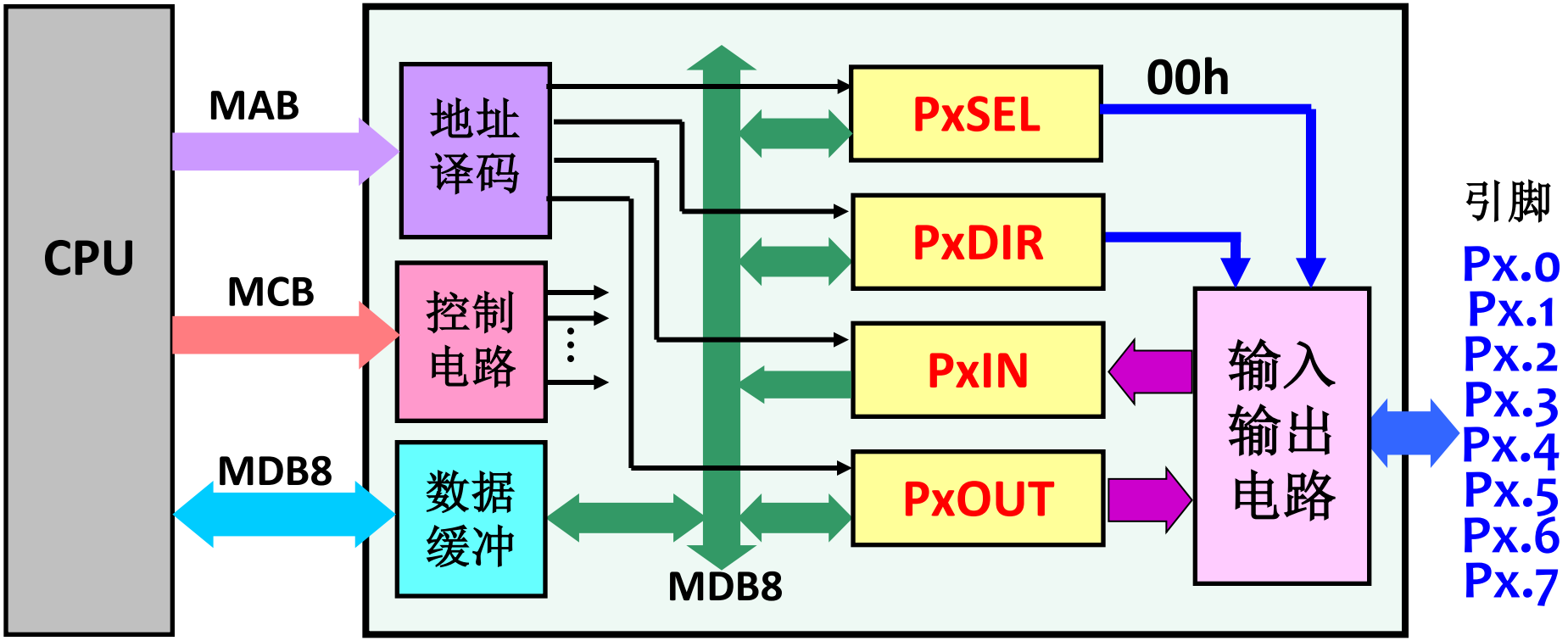
对于P1端口，有P1SEL，P1DIR， P1OUT，P1IN

▲均为8位寄存器

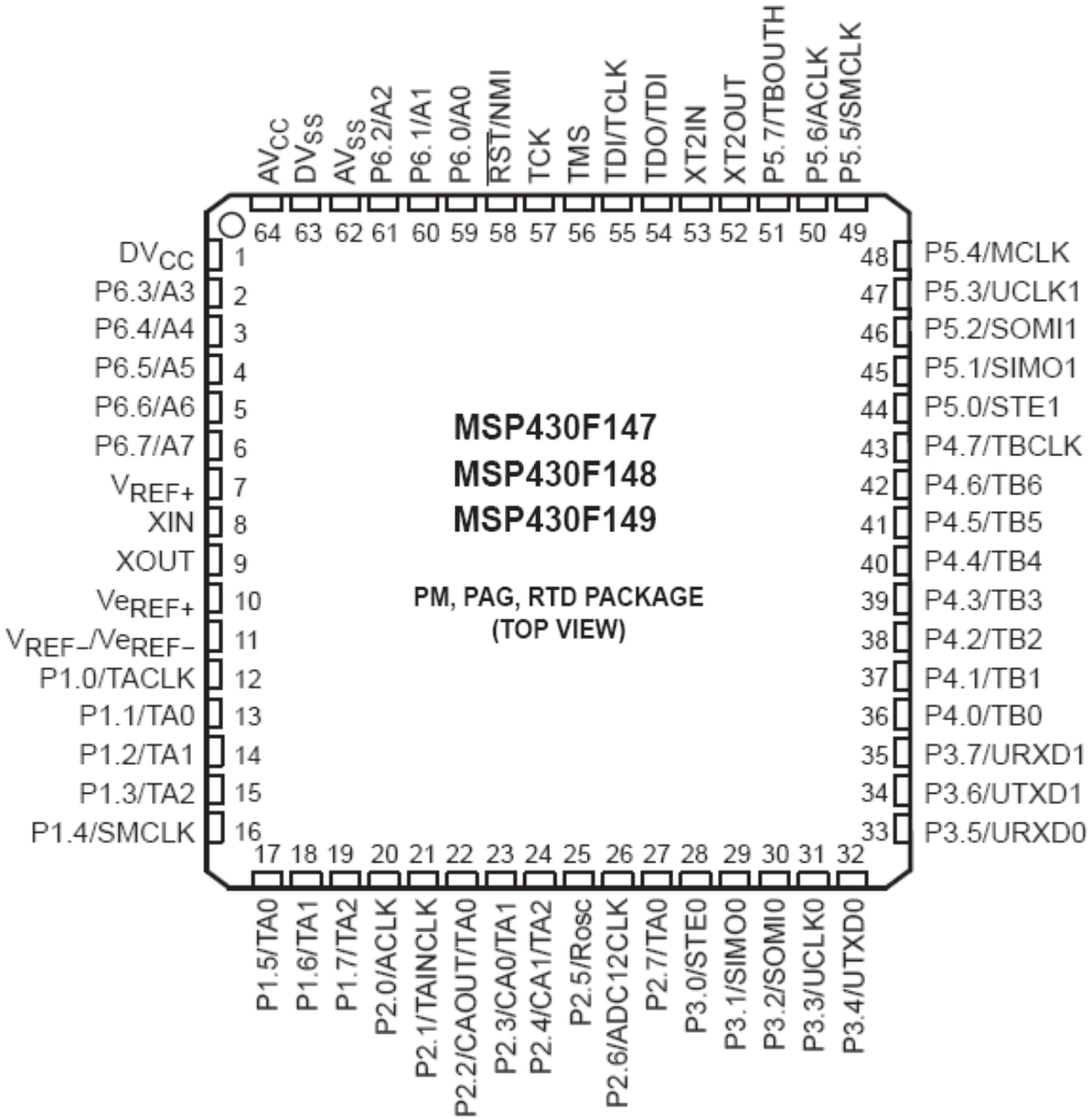
▲端口的每根引脚可单独配置，相互之间不影响

I/O端口x 基本输入/输出示意图(不考虑其他模块功能时)

General-purpose digital I/O（PxSEL=00H时）



PxSEL 功能选择寄存器：	0: I/O功能；	1: 外设模块功能
PxDIR 方向选择寄存器:	0: 输入；	1: 输出
PxIN 输入寄存器：	0: 输入低电平；	1: 输入高电平
PxOUT输出寄存器：	0: 输出低电平；	1: 输出高电平



I/O引脚与MCU内其他模块共用(引脚复用)

P4.0/TB0
P4.1/TB1
P4.2/TB2
P4.3/TB3
P4.4/TB4
P4.5/TB5
P4.6/TB6
P4.7/TBCLK

PxSEL功能选择寄存器

为减少MCU的对外引脚,
端口P1~P6的引脚有多种功能, 即被MCU内的多个模块复用,
功能选择寄存器就是用于选择引脚的功能

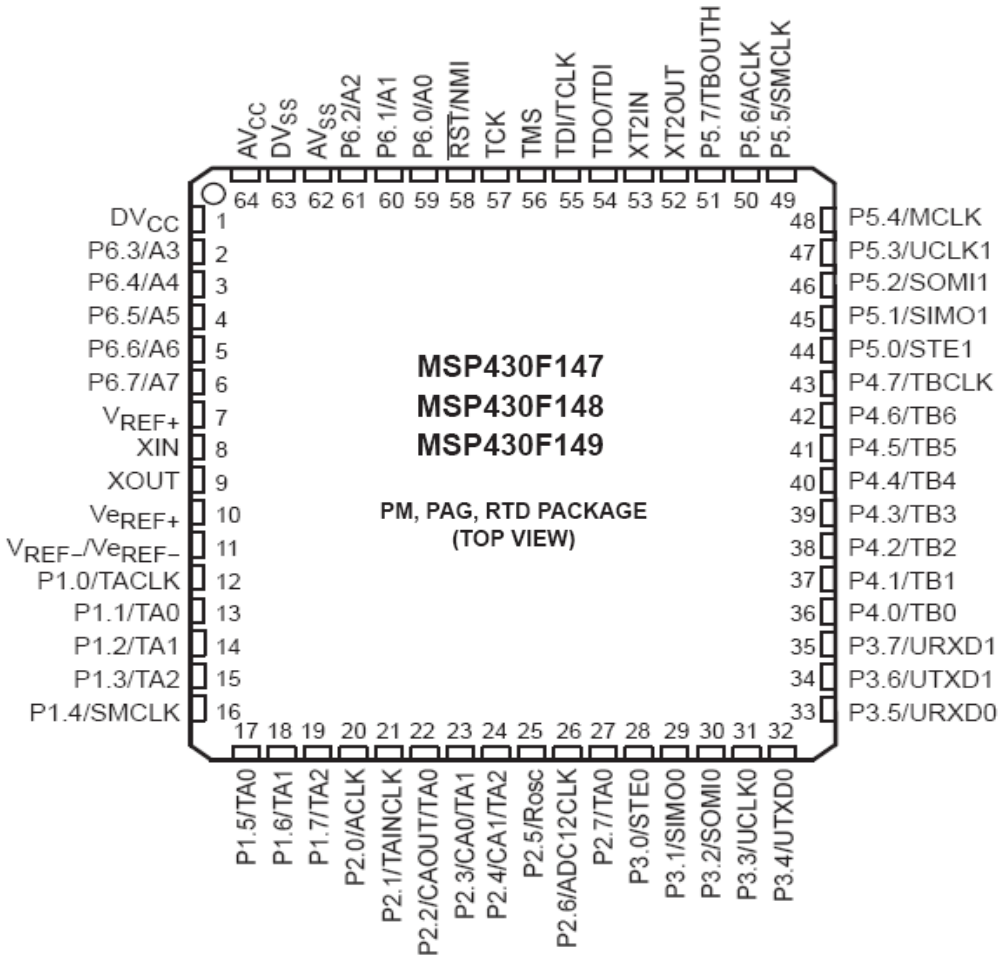
Port pins are often multiplexed with other peripheral module functions.
Each PxSEL bit is used to select the pin function – I/O port or peripheral module function.

位	7	6	5	4	3	2	1	0
	PxSEL.7	PxSEL.6	PxSEL.5	PxSEL.4	PxSEL.3	PxSEL.2	PxSEL.1	PxSEL.0
初始值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

PxSEL.y = 0 , 端口x的引脚y选择为基本I/O功能 **I/O Function is selected**

PxSEL.y = 1 , 端口x的引脚y选择为外围模块功能 **Peripheral module function …**

例 如果P1SEL的内容为00h,
表示端口P1的 8个引脚P1.7~P1.0 均用于基本I/O



每个引脚输入/输出的

方向是单向的，

要么是输入

要么是输出

PxDIR 方向选择寄存器

P1~P6端口为单向端口，
通过方向寄存器选择各端口引脚的输入/输出方向

Each bit in each PxDIR register selects the direction of the corresponding I/O pin, regardless of the selected function for the pin.

位	7	6	5	4	3	2	1	0
	PxDIR.7	PxDIR.6	PxDIR.5	PxDIR.4	PxDIR.3	PxDIR.2	PxDIR.1	PxDIR.0
复位值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

PxDIR.y = 0 ,端口x的引脚y选择为输入方向 The port pin is switched to input direction

PxDIR.y = 1 ,端口x的引脚y选择为输出方向 The port pin is switched to output direction

例 如果P1DIR的内容为00h, 表示端口P1的8个引脚均用于输入
如果P2DIR的内容为FFh, 表示端口P2的8个引脚均用于输出
如果P3DIR的内容为0Fh, 表示端口P3的P3.7~P3.4用于输入
端口P3的P3.3~P3.0用于输出

端口寄存器位属性和初值说明

标识	位属性
rw	可读写
r	只读
r0	读总为0
r1	读总为1
w	只写
w0	写0
w1	写1
(w)	不存在位，读出总为0
-0, -1	上电清零后的初值(简单理解为软复位)
-(0), -(1)	POR上电复位后的初值(简单理解为硬复位)
h0	由硬件清0
h1	由硬件置1

PxOUT输出寄存器

▲ 是端口的输出缓冲寄存器，

▲ 当端口某位的DIR设置为输出时，
该位端口引脚的状态对应为输出寄存器相应位的值。

▲ 编程改变输出寄存器该位的值就可改变该位引脚的输出状态

Each bit in each PxOUT register is the value to be output on the corresponding I/O pin when the pin is configured as I/O function and output direction.

位	7	6	5	4	3	2	1	0
	PxOUT.7	PxOUT.6	PxOUT.5	PxOUT.4	PxOUT.3	PxOUT.2	PxOUT.1	PxOUT.0
复位值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

PxOUT.y = 0 , 端口x的引脚y输出低 The output is low

PxOUT.y = 1 , 端口x的引脚y输出高 The output t is high

例 如果P2OUT的内容为00h,
表示向端口P2的8个引脚均输出0, 即低电平
如果P2OUT的内容为0Fh,
表示向端口P2的引脚P2.7~P2.4输出0, 即低电平
表示向端口P2的引脚P2.3~P2.0输出1, 即高电平

PxIN输入寄存器

▲是端口的输入缓冲寄存器;

▲当端口某位的DIR设置为输入时,
输入寄存器相应位的值对应为该引脚的输入状态;

▲通过编程读取该寄存器对应位的值,就可获得相应引脚的状态

Each bit in each PxIN register reflects the value of the input signal at the corresponding I/O pin when the pin is configured as I/O function.

位	7	6	5	4	3	2	1	0
r	PxIN.7	PxIN.6	PxIN.5	PxIN.4	PxIN.3	PxIN.2	PxIN.1	PxIN.0

PxIN.y = 0 , 端口x的引脚y输入为低 The input is low

PxIN.y = 1 , 端口x的引脚y输入为高 The input is high

例 如果P1IN的内容为00h,
表示端口P1的8个引脚的状态均为0, 即低电平
如果P1IN的内容为0Fh,
表示端口P1的引脚P1.7~P1.4的状态为0,即低电平
端口P1的引脚P1.3~P1.0的状态为1,即高电平

PxREN上拉或下拉电阻使能寄存器

▲使能或禁止相应I/O引脚的上拉或下拉电阻;

位	7	6	5	4	3	2	1	0
	PxREN.7	PxREN.6	PxREN.5	PxREN.4	PxREN.3	PxREN.2	PxREN.1	PxREN.0
复位值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

PxREN.y = 1 ,使能端口x的引脚y上拉或下拉电阻;

PxREN.y = 0 ,禁止端口x的引脚y上拉或下拉电阻;

▲ 使能引脚上拉或下拉功能后，

通过设置PxOUT.y相应位来选择上拉或下拉

PxOUT.y = 1 ,选择端口x的引脚y通过电阻接到Vcc电源上, 把电平拉高;

PxOUT.y = 0 ,选择端口x的引脚y通过电阻接到Vss地线上, 把电平拉低;

PxDR 输出驱动强度寄存器

▲设置相应I/O引脚的输出驱动强度;

位	7	6	5	4	3	2	1	0
	PxDR.7	PxDR.6	PxDR.5	PxDR.4	PxDR.3	PxDR.2	PxDR.1	PxDR.0
复位值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

PxDR.y = 1 , 置端口x的引脚y为高驱动强度;

PxDR.y = 0 , 置端口x的引脚y为低驱动强度;

与基本I/O功能有关的地址

Table 31. Port P1/P2 Registers (Base Address: 0200h)

REGISTER DESCRIPTION	REGISTER	OFFSET
Port P1 input	P1IN	00h
Port P1 output	P1OUT	02h
Port P1 direction	P1DIR	04h
Port P1 pullup/pulldown enable	P1REN	06h
Port P1 drive strength	P1DS	08h
Port P1 selection	P1SEL	0Ah
Port P1 interrupt vector word	P1IV	0Eh
Port P1 interrupt edge select	P1IES	18h
Port P1 interrupt enable	P1IE	1Ah
Port P1 interrupt flag	P1IFG	1Ch
Port P2 input	P2IN	01h
Port P2 output	P2OUT	03h
Port P2 direction	P2DIR	05h
Port P2 pullup/pulldown enable	P2REN	07h
Port P2 selection	P2SEL	0Bh
Port P2 interrupt vector word	P2IV	1Eh
Port P2 interrupt edge select	P2IES	19h
Port P2 interrupt enable	P2IE	1Bh
Port P2 interrupt flag	P2IFG	1Dh

Table 32. Port P3/P4 Registers (Base Address: 0220h)

REGISTER DESCRIPTION	REGISTER	OFFSET
Port P3 input	P3IN	00h
Port P3 output	P3OUT	02h
Port P3 direction	P3DIR	04h
Port P3 pullup/pulldown enable	P3REN	06h
Port P3 drive strength	P3DS	08h
Port P3 selection	P3SEL	0Ah
Port P3 interrupt vector word	P3IV	0Eh
Port P3 interrupt edge select	P3IES	18h
Port P3 interrupt enable	P3IE	1Ah
Port P3 interrupt flag	P3IFG	1Ch
Port P4 input	P4IN	01h
Port P4 output	P4OUT	03h
Port P4 direction	P4DIR	05h
Port P4 pullup/pulldown enable	P4REN	07h
Port P4 drive strength	P4DS	09h
Port P4 selection	P4SEL	0Bh
Port P4 interrupt vector word	P4IV	1Eh
Port P4 interrupt edge select	P4IES	19h
Port P4 interrupt enable	P4IE	1Bh
Port P4 interrupt flag	P4IFG	1Dh

Table 33. Port P5/P6 Registers (Base Address: 0240h)

REGISTER DESCRIPTION	REGISTER	OFFSET
Port P5 input	P5IN	00h
Port P5 output	P5OUT	02h
Port P5 direction	P5DIR	04h
Port P5 pullup/pulldown enable	P5REN	06h
Port P5 drive strength	P5DS	08h
Port P5 selection	P5SEL	0Ah
Port P6 input	P6IN	01h
Port P6 output	P6OUT	03h
Port P6 direction	P6DIR	05h
Port P6 pullup/pulldown enable	P6REN	07h
Port P6 drive strength	P6DS	09h
Port P6 selection	P6SEL	0Bh

Table 34. Port P7/P8 Registers (Base Address: 0260h)

REGISTER DESCRIPTION	REGISTER	OFFSET
Port P7 input	P7IN	00h
Port P7 output	P7OUT	02h
Port P7 direction	P7DIR	04h
Port P7 pullup/pulldown enable	P7REN	06h
Port P7 drive strength	P7DS	08h
Port P7 selection	P7SEL	0Ah
Port P8 input	P8IN	01h
Port P8 output	P8OUT	03h
Port P8 direction	P8DIR	05h
Port P8 pullup/pulldown enable	P8REN	07h
Port P8 drive strength	P8DS	09h
Port P8 selection	P8SEL	0Bh

Table 35. Port P9 Register (Base Address: 0280h)

REGISTER DESCRIPTION	REGISTER	OFFSET
Port P9 input	P9IN	00h
Port P9 output	P9OUT	02h
Port P9 direction	P9DIR	04h
Port P9 pullup/pulldown enable	P9REN	06h
Port P9 drive strength	P9DS	08h
Port P9 selection	P9SEL	0Ah

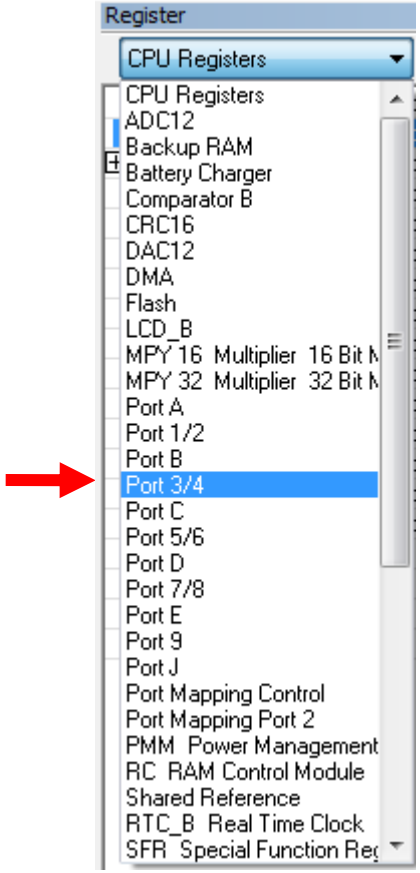
Table 36. Port J Registers (Base Address: 0320h)

REGISTER DESCRIPTION	REGISTER	OFFSET
Port PJ input	PJIN	00h
Port PJ output	PJOUT	02h
Port PJ direction	PJDIR	04h
Port PJ pullup/pulldown enable	PJREN	06h
Port PJ drive strength	PJDS	08h

在EW430下查看各P1~P6端口

View/Registers/

Port 1/2、 Port 3/4、 Port 5/6， Port 7/8, Port 9, Port J



例: 在EW430下查看P3DIR

View/Registers/ Port 3/4

点击P3DIR前的"+", 还可展开看P3DIR的各位