

第4章 输入/输出接口技术

第1节 接口技术概述

第2节 CPU与外设间的数据传送方式

第1节 接口技术的概述

- 一、接口的概念和功能
- 二、I/O接口电路的典型结构
- 三、端口的编址

一、接口的概念和功能

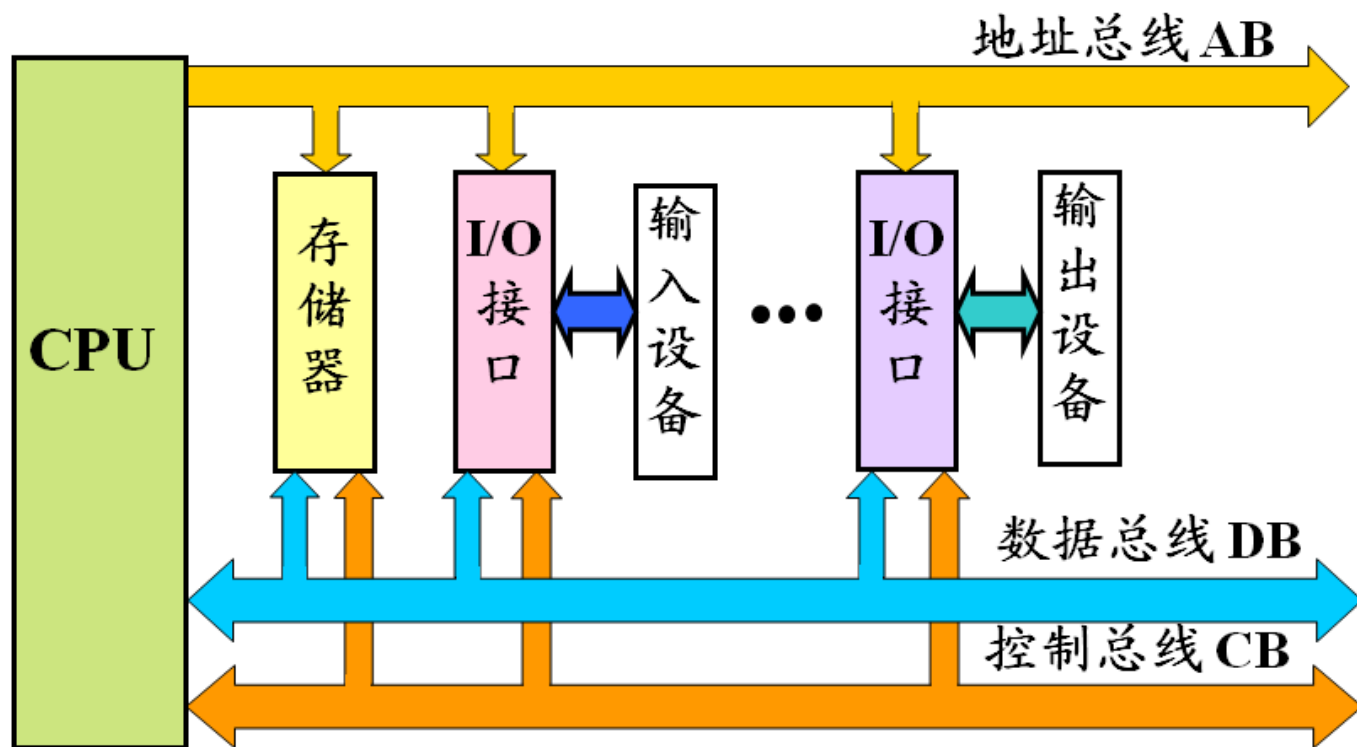
1. 接口和接口技术

2. 常见外设的信号类型、特点和接口电路的作用

1. 接口和接口技术

接口

指CPU、存储器、外设之间通过总线进行连接的电路，是CPU与外界进行信息交换的桥梁，用于实现CPU与外设间速度、电平、信息类型等的正确匹配。



接口技术

是研究CPU如何与外部世界在信息类型、通信方式、信号电平、操作时序上进行最佳匹配，实现双方高效、可靠地交换信息的一门技术，是软件、硬件结合的体现，是计算机应用的关键。

外部世界：除CPU以外的所有设备或电路，包括存储器、输入/输出设备等

2、常见外设的信号类型、特点和接口电路的作用

外设是用来实现人机交互的一些机电设备。

外设的信息类型、速度、通信方式与CPU不匹配，不能直接挂在总线上，必须通过接口和系统相连

	CPU	接口作用	外设
信息类型	数字量	模/数转换 (A/D) ←→ 数/模转换 (D/A)	模拟量
		三态缓冲、锁存	数字量
工作速度	快	解决传送方式	慢
通信方式	并行	串/并转换 ←→ 并/串转换	串行
		三态缓冲、锁存	并行

二、I/O接口电路的典型结构

- 各种接口电路

并行接口

定时器

串行接口

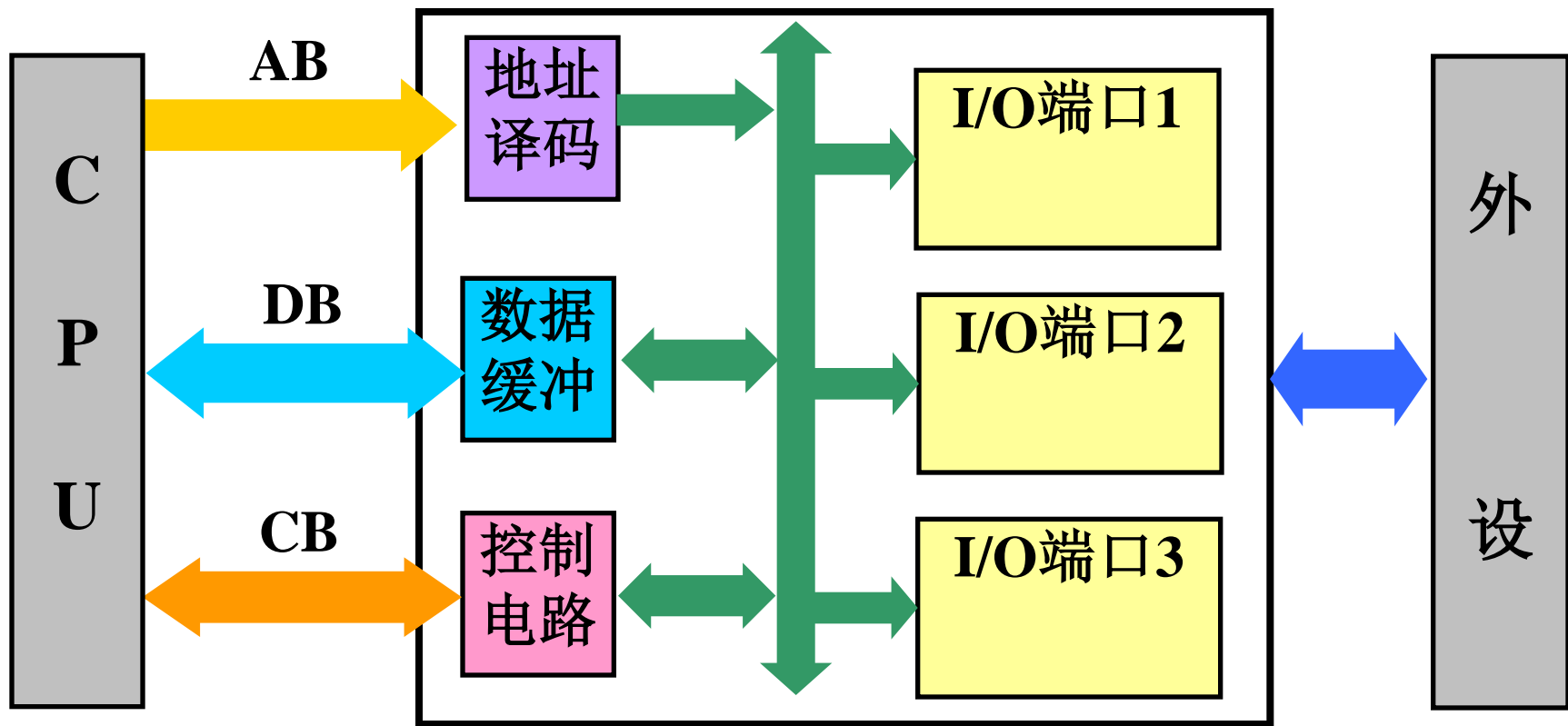
中断接口

A/D转换接口

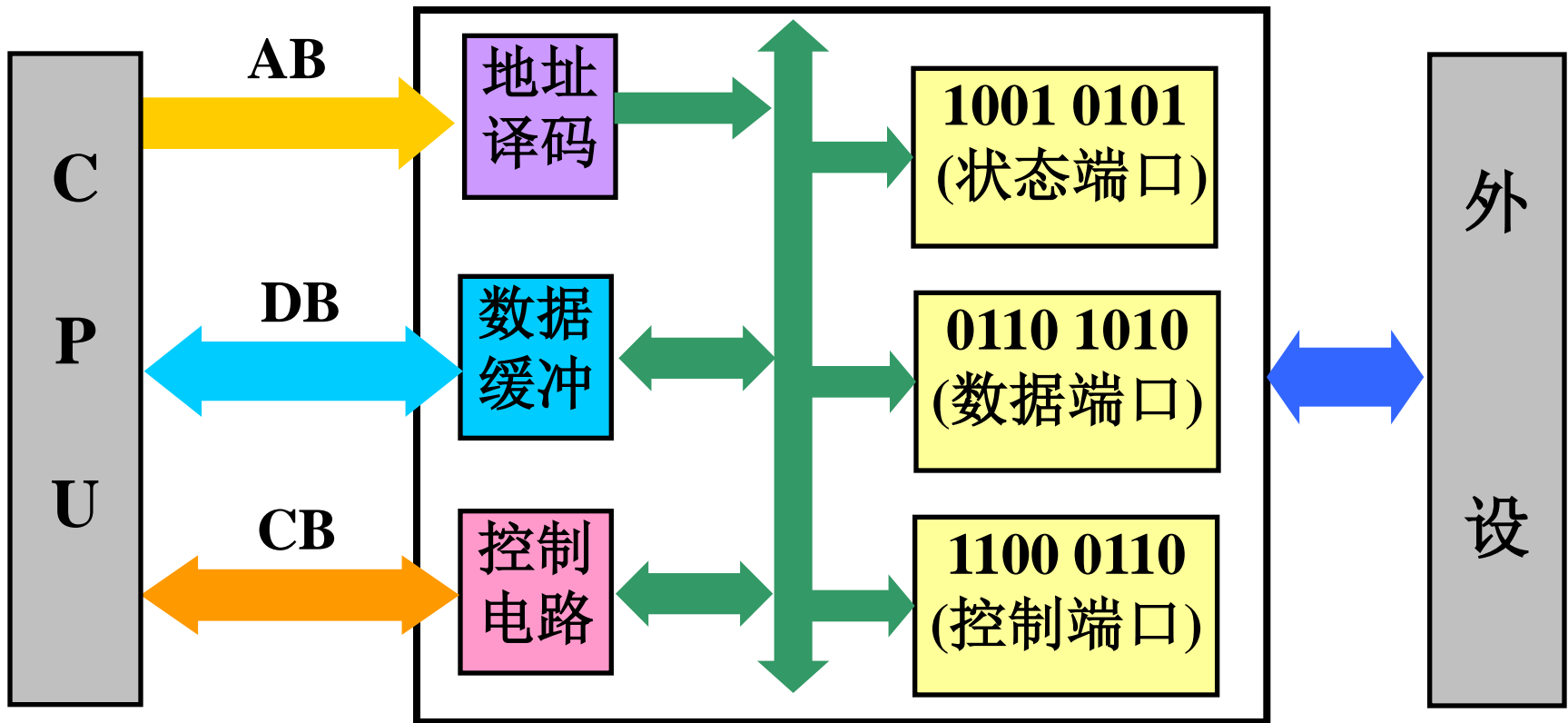
D/A转换接口

.....

I/O接口电路的典型结构



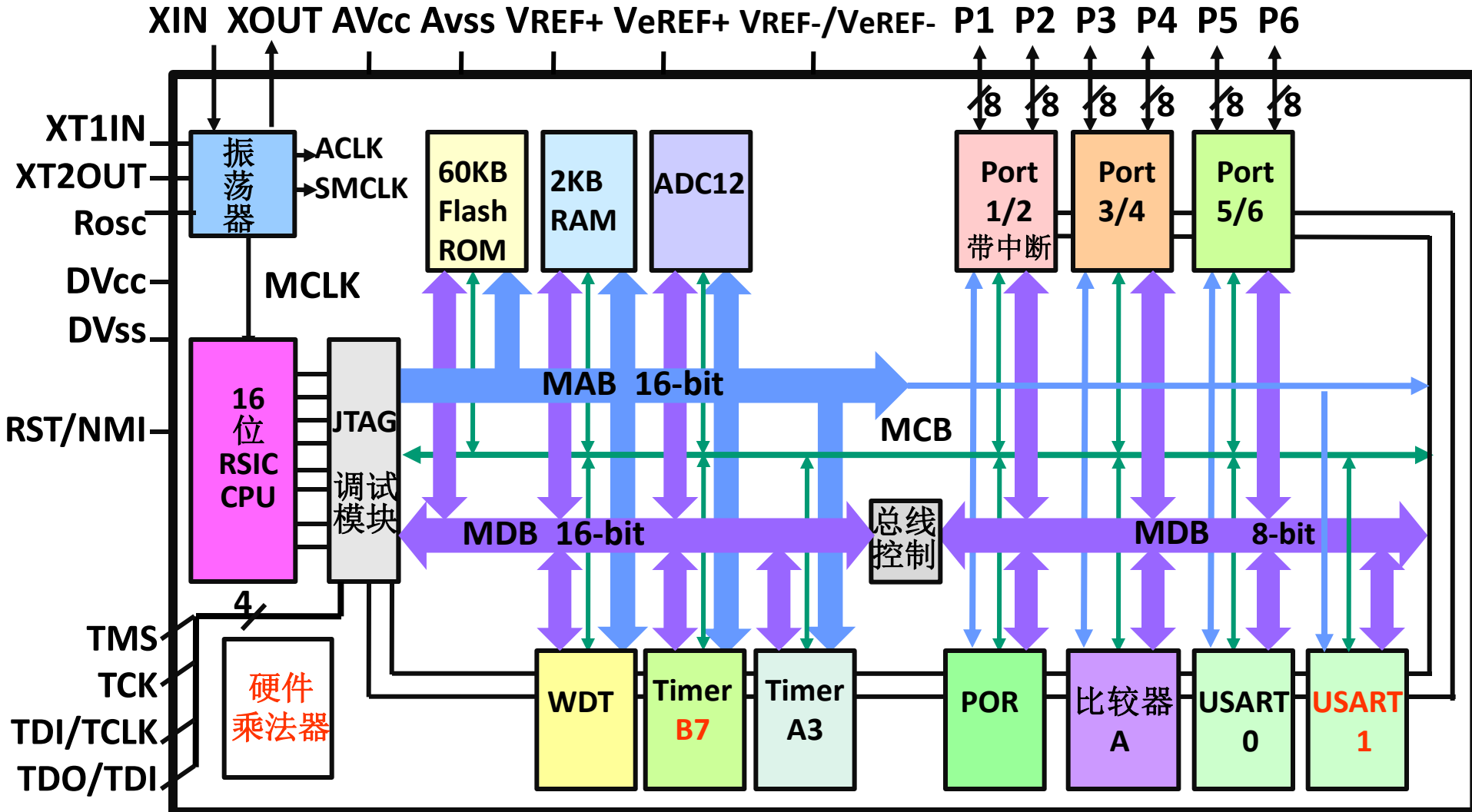
- 从编程角度看，接口内部主要包括一个或多个CPU可以进行读/写操作的寄存器，又称为I/O端口。
- 各I/O端口由端口地址区分。



- 按存放信息的不同，I/O端口可分为三种类型
 - 数据端口：用于存放CPU与外设间传送的数据信息
 - 状态端口：用于暂存外设的状态信息
 - 控制端口：用于存放CPU对外设或接口的控制信息，控制外设或接口的工作方式。

MSP430F149内部各模块端口

参看MSP430F13x-14x.pdf的P20~23



MSP430F149内部各模块端口(续)

模块	端口名称	端口地址
特殊功能	IE1	000h
	IE2	001h
	IFG1	002h
	IFG2	003h
	ME1	004h
	ME2	005h
P1	P1IN	020h
	P1OUT	021h
	P1DIR	022h
	P1IFG	023h
	P1IES	024h
	P1IE	025h
	P1SEL	026h

模块	端口名称	端口地址
P2	P2IN	028h
	P2OUT	029h
	P2DIR	02Ah
	P2IFG	02Bh
	P2IES	02Ch
	P2IE	02Dh
	P2SEL	02Eh
P3	P3IN	018h
	P3OUT	019h
	P3DIR	01Ah
	P3SEL	01Bh
P4	P4IN	01Ch
	P4OUT	01Dh
	P4DIR	01Eh
	P4SEL	01Fh

模块	端口名称	端口地址
P5	P5IN	030h
	P5OUT	031h
	P5DIR	032h
	P5SEL	033h
P6	P6IN	034h
	P6OUT	035h
	P6DIR	036h
	P6SEL	037h
基本时钟	DCOCTL	056H
	BCSCTL1	057H
	BCSCTL2	058H
比较器A	CACTL1	059H
	CACTL2	05AH
	CAPD	05BH

MSP430F149内部各模块端口(续)

模块	端口名称	端口地址
串口0	U0CTL	070h
	U0TCTL	071h
	U0RCTL	072h
	U0MCTL	073h
	U0BR0	074h
	U0BT1	075h
	U0RXBUF	076h
	U0TXBUF	077h
串口1	U1CTL	078h
	U1TCTL	079h
	U1RCTL	07Ah
	U1MCTL	07Bh
	U1BR0	07Ch
	U1BT1	07Dh
	U1RXBUF	07Eh
	U1TXBUF	07Fh

模块	端口名称	端口地址
ADC12	ADC12MCTL0	080h
	~	~
	ADC12MCTL15	08Fh
	ADC12MEM0	0140h
	ADC12MEM1	0142h
	~	~
	ADC12MEM14	015Ch
	ADC12MEM15	015Eh
	ADC12CTL0	01A0h
	ADC12CTL1	01A2h
	ADC12IFG	01A4h
	ADC12IE	01A6h
ADC12IV	01A8h	
FLASH	FCTL1	0128h
	FCTL2	012Ah
	FCTL3	012Ch
WDT	WDTCTL	0120h

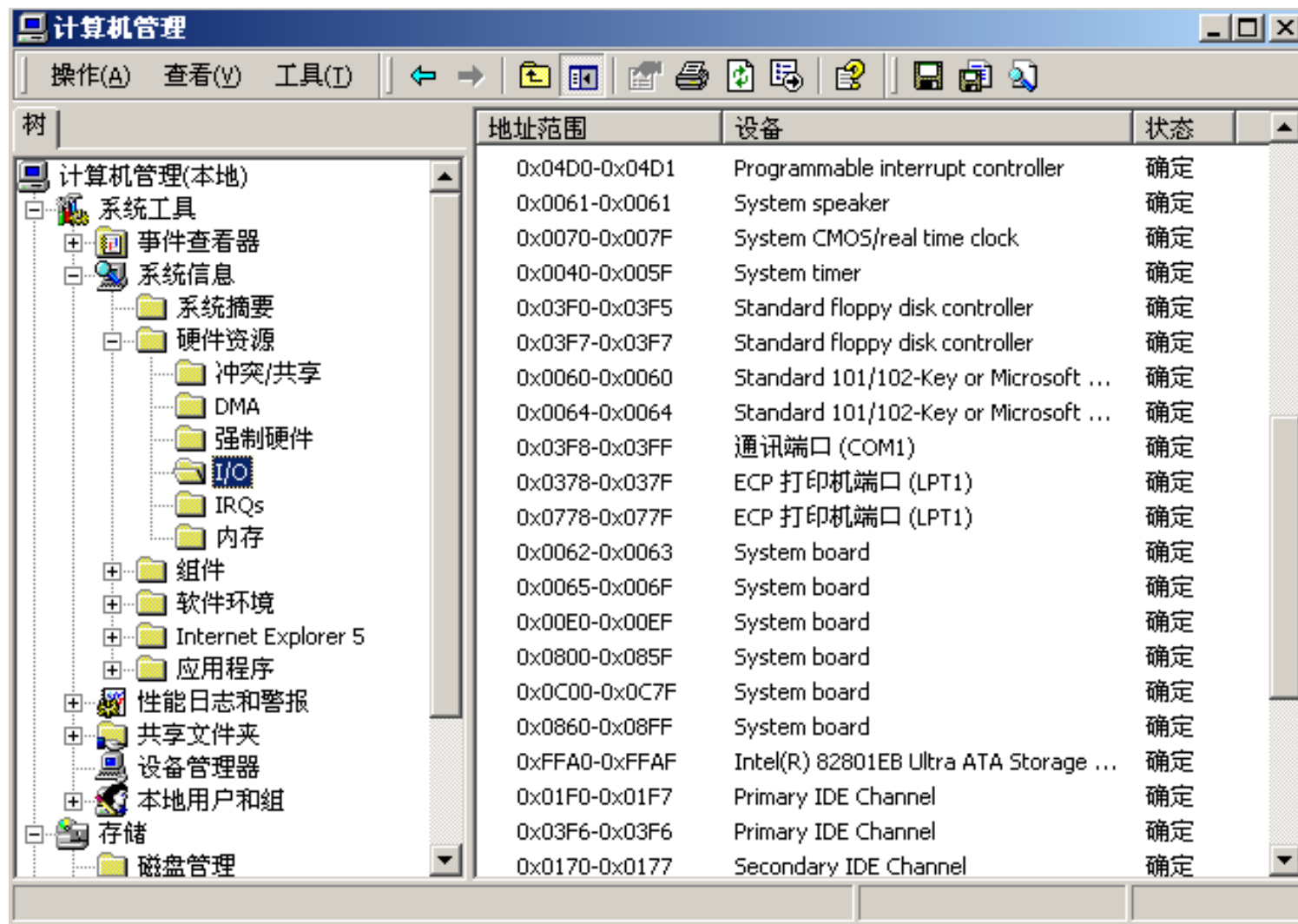
MSP430F149内部各模块端口(续)

模块	端口名称	端口地址
乘法器	SUMEXT	013Eh
	RESHI	013Ch
	RESH0	013Ah
	OP2	0138h
	MACS	0136h
	MAC	0134h
	MPYS	0132h
	MPY	0130h
定时器 A	TAIV	012Eh
	TACTL	0160h
	TACCTL0	0162h
	TACCTL1	0164h
	TACCTL2	0166h
	TAR	0170h
	TACCR0	0172h
	TACCR1	0174h
	TACCR2	0176h

模块	端口名称	端口地址
定时器 B	TBIV	012Eh
	TBCTL	0160h
	TBCCTL0	0182h
	TBCCTL1	0184h
	TBCCTL2	0186h
	TBCCTL3	0188h
	TBCCTL4	018Ah
	TBCCTL5	018Ch
	TBCCR6	018Eh
	TBR	0190h
	TBCCR0	0192h
	TBCCR1	0194h
	TBCCR2	0196h
	TBCCR3	0198h
	TBCCR4	019Ah
	TBCCR5	019Ch
TBCCR6	019Eh	

I/O端口是计算机硬件系统的一种重要资源

在Windows下查看PC机硬件IO端口资源的分配情况

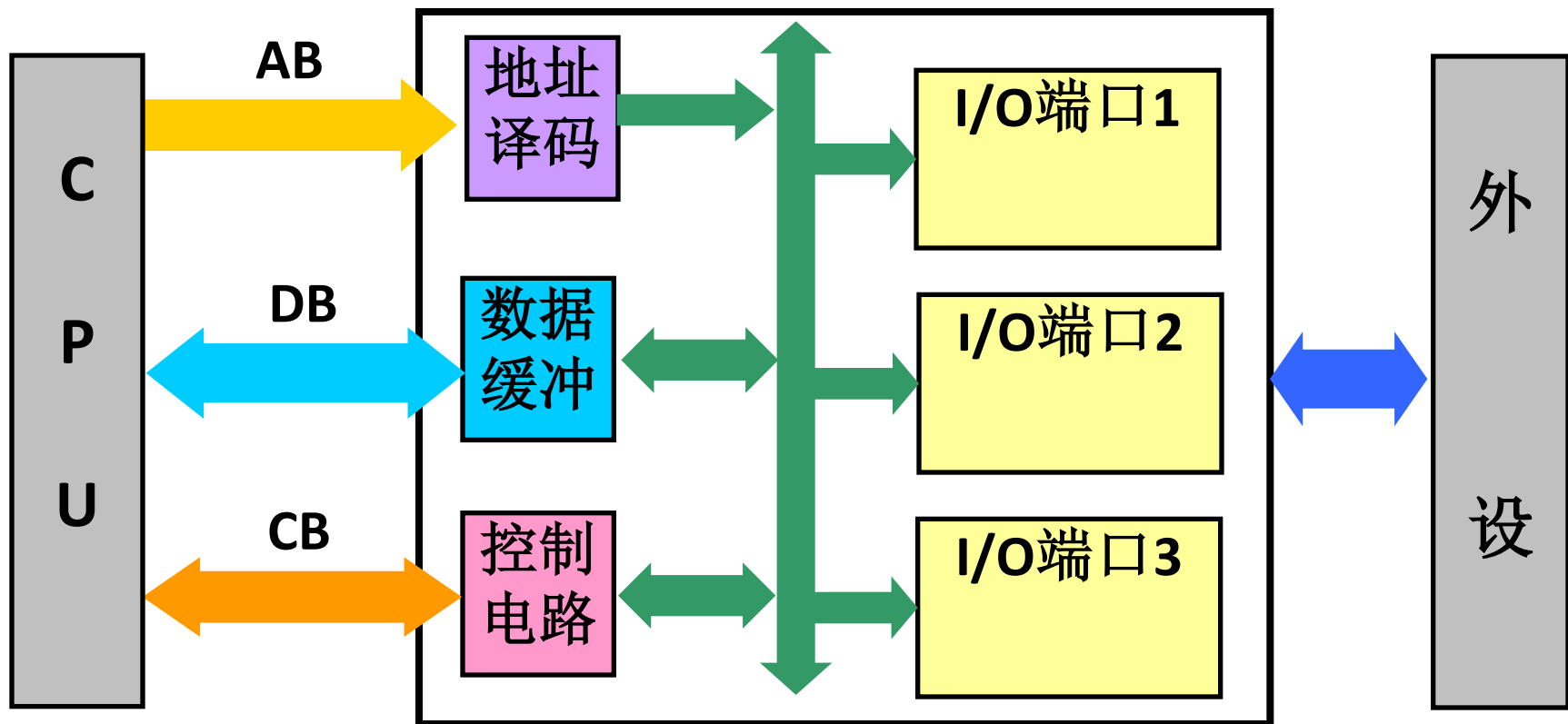


The screenshot shows the Windows '计算机管理' (Computer Management) console. The left pane shows the tree view expanded to '系统信息' > '硬件资源' > 'I/O'. The right pane displays a table of I/O port resources.

地址范围	设备	状态
0x04D0-0x04D1	Programmable interrupt controller	确定
0x0061-0x0061	System speaker	确定
0x0070-0x007F	System CMOS/real time clock	确定
0x0040-0x005F	System timer	确定
0x03F0-0x03F5	Standard floppy disk controller	确定
0x03F7-0x03F7	Standard floppy disk controller	确定
0x0060-0x0060	Standard 101/102-Key or Microsoft ...	确定
0x0064-0x0064	Standard 101/102-Key or Microsoft ...	确定
0x03F8-0x03FF	通讯端口 (COM1)	确定
0x0378-0x037F	ECP 打印机端口 (LPT1)	确定
0x0778-0x077F	ECP 打印机端口 (LPT1)	确定
0x0062-0x0063	System board	确定
0x0065-0x006F	System board	确定
0x00E0-0x00EF	System board	确定
0x0800-0x085F	System board	确定
0x0C00-0x0C7F	System board	确定
0x0860-0x08FF	System board	确定
0xFFA0-0xFFAF	Intel(R) 82801EB Ultra ATA Storage ...	确定
0x01F0-0x01F7	Primary IDE Channel	确定
0x03F6-0x03F6	Primary IDE Channel	确定
0x0170-0x0177	Secondary IDE Channel	确定

与Intel 80x86 PC 微机兼容的 I/O端口地址分配

I/O地址	I/O设备
0020~003F	中断控制器8259A主片
0040~005F	系统时钟8253
0060~006F	键盘
0070~007F	系统CMOS/实时钟
00A0~00BF	中断控制器8259A从片
00C0~00DF	DMA控制器
0170~0177	从IDE通道(硬盘)
01F0~01F7	主IDE通道(硬盘)
02F8~02FF	串行通信端口2(COM2)
0378~03FF	并行打印机端口(LPT1)
03B0~03DF	显示适配器(显卡)
03F0~03F7	软盘控制器
03F8~03FF	串行通信端口1(COM1)



- CPU对外设输入/输出的控制，
是通过对接口中各I/O端口的读/写操作完成。

这些端口如何编址，如何操作？

三、I/O端口的编址

1. 端口与存储器分别独立编址
2. 端口与存储器统一编址

1. 端口与存储器分别独立编址 (I/O映射方式)

例 Intel的80x86系列、Z80系列

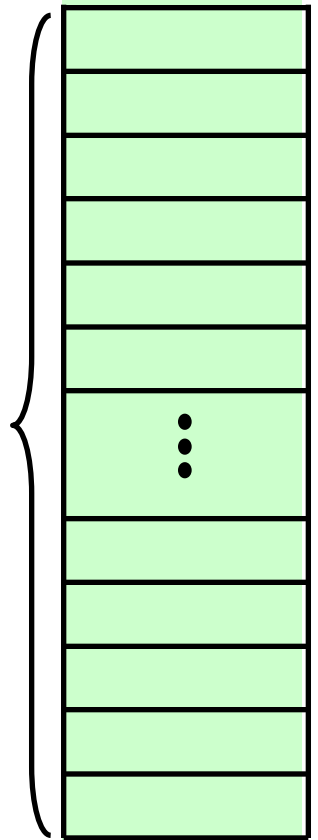
特点:

- 端口与存储器分别独立编址
端口不占用存储器空间
- 设有专门的 I/O指令对端口进行读写,
对存储器操作的指令不能用于I/O端口

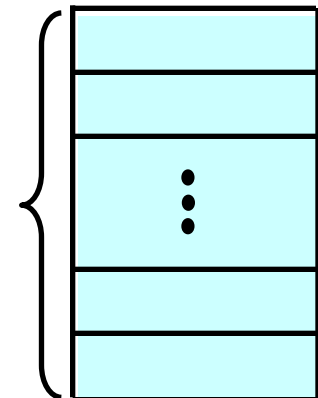
例 MOV [10H], AL 对存储器操作

 OUT 10H, AL 对端口操作

存储器空间



I/O空间

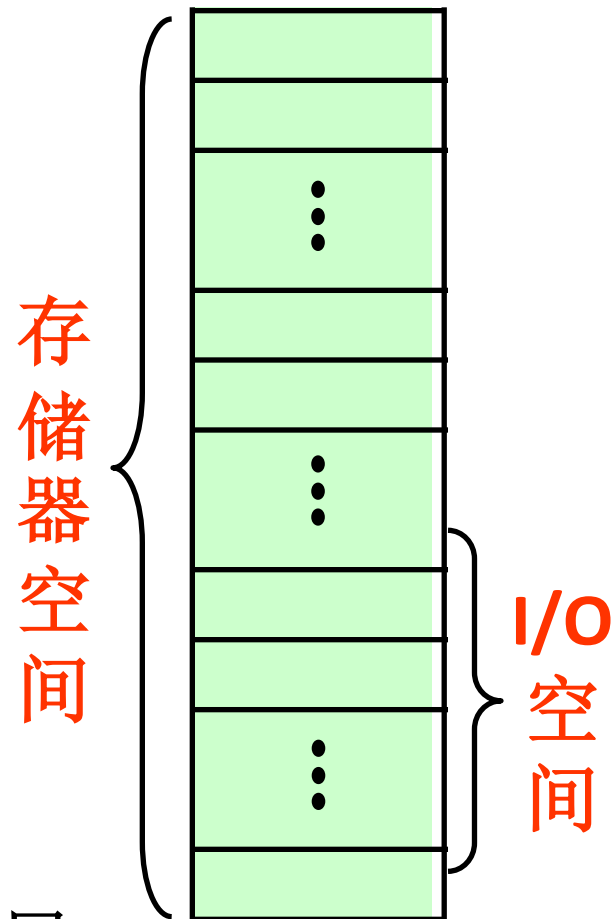


2. 端口与存储器统一编址 (存储器映射方式)

例 MSP430单片机
motorola的M6800系列
瑞萨单片机系列
8051单片机系列

特点:

- I/O端口相当于存储器的一部分,使存储器容量减小
- 对I/O端口的读/写与对存储器的读/写相同,所有可对存储器操作的指令对I/O端口均可使用
- 指令系统中不专设I/O指令



第2节 CPU与外设间的数据传送方式

CPU与外设的工作速度不一致，
如何使两者高效、可靠地进行数据传送，
是本节讨论的问题。

有以下几种传送方式:

- 一、无条件传送方式
- 二、条件传送方式 (查询方式)
- 三、中断传送方式
- 四、DMA传送方式
(**D**irect **M**emory **A**ccess)

一、无条件传送方式 (同步传送方式)

● 实现方法

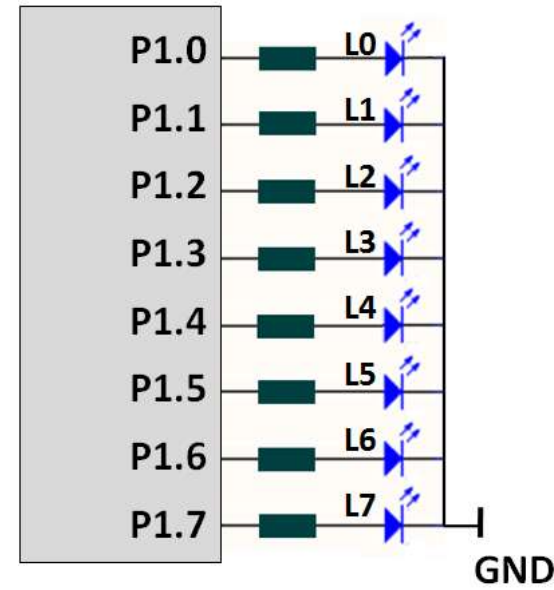
CPU不查询外设工作状态，
与外设速度的匹配通过在软件上延时完成，
在程序中直接用I/O指令，完成与外设的数据传送

● 特点

1. 适用于外设动作时间已知，
在CPU与外设进行数据传送时，外设保证已准备好的情况
2. 软硬件十分简单。

第3章 例1中对LED灯的控制采用无条件传送方式

```
#include "msp430F6638.h"
unsigned char LED_0=0x01, LED_temp;
unsigned int i, j;
int main( void )
{ WDTCTL = WDTPW + WDTHOLD;
  P1SEL=0; //设置P1为基本I/O
  P1DIR=0xFF; //设置P1为输出端口
  P1OUT=0; //使8个LED全灭
  while(1) //无限循环
  { LED_temp=LED_0; //点亮L0的值
    for ( i=0; i<8; i++) //8个LED依次点亮
    { P1OUT=LED_temp; //
      for (j=0;j<0xffff;j++); //延时
      LED_temp=LED_temp<<1; //改变点亮值
    }
  }
};
}
```



二、条件传送方式(查询传送方式)

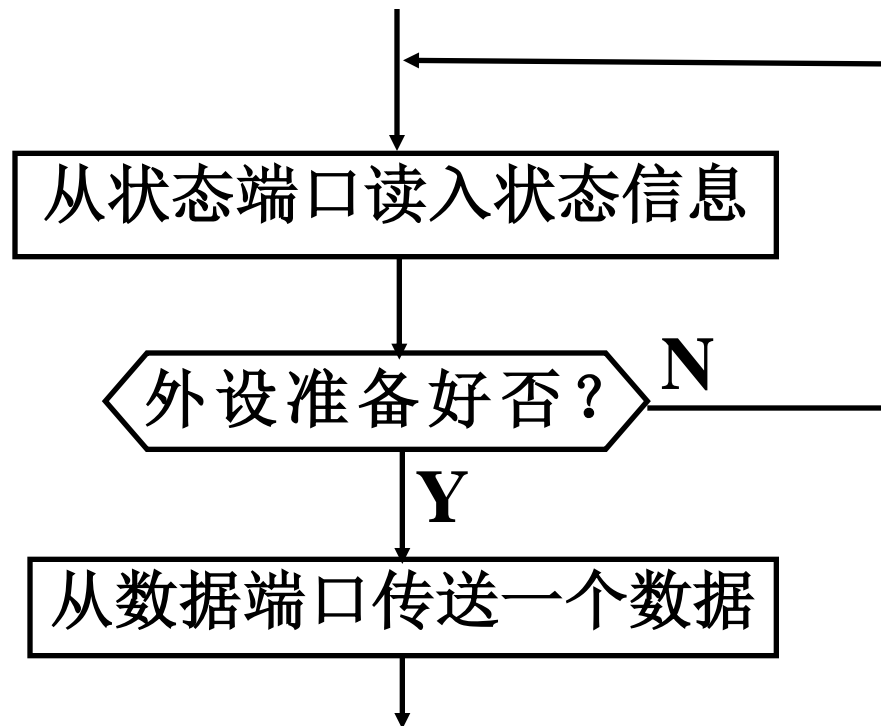
- **实现方法:**

在与外设进行数据传送前，**CPU**先查询外设状态，当外设准备好后，才执行**I/O**指令，实现数据传送

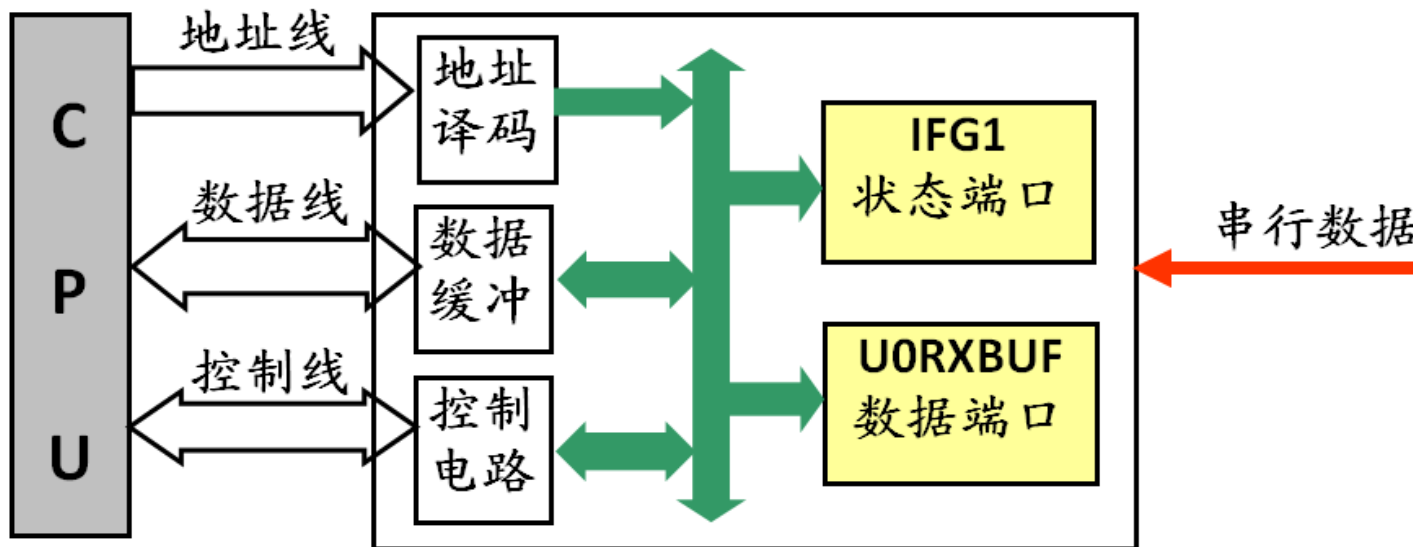
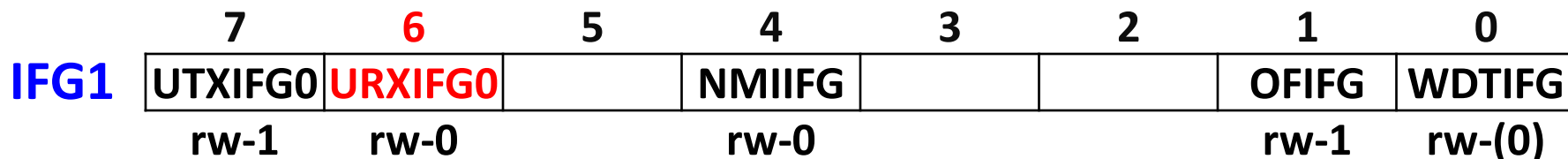
- **特点:**

1. **CPU**通过不断查询外设状态，实现与外设的速度匹配
2. **CPU**的工作效率低

- 查询传送方式，编程流程：



在MSP430的串行接口中，
寄存器IFG1的D6位为1,表示接收到新的数据，
CPU可读取寄存器UORXBUF中的数据，
CPU读取新数据后，将自动清除置寄存器IFG1的D6位为0



MSP430的串行接收示意图

编程实现采用查询方式从串行接口接收20个字节数据，
存放到到首地址为buffer的RAM中

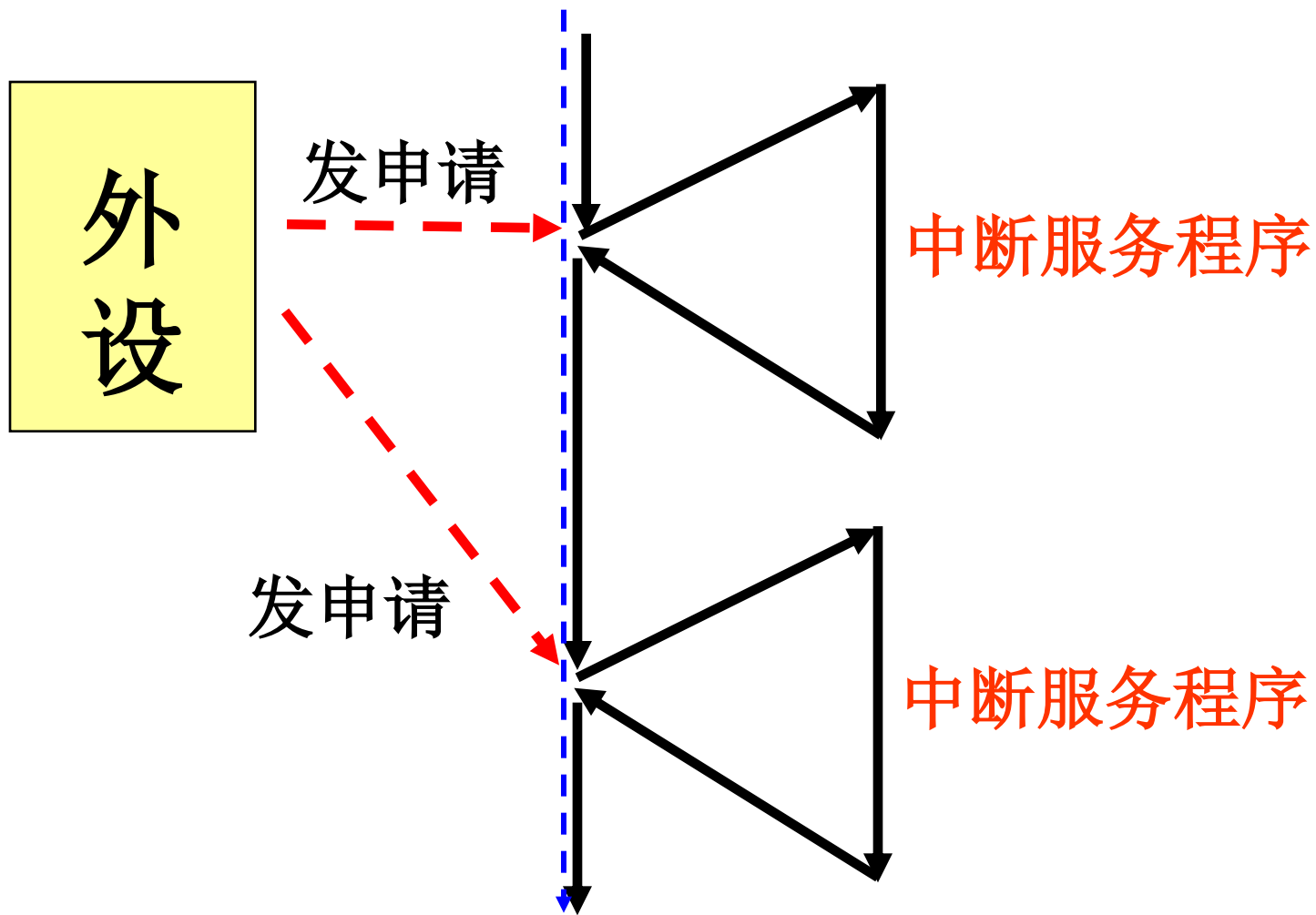
```
....  
char  buffer[32];  
unsigned int  j;  
...  
for (j=0; j<32; j++)  
{ while((IFG1&URXIFG0)==0); //检测接收缓冲器是否满  
  buffer[j]= U0RXBUF; //接收一个字符并保存  
};  
....
```

三、中断传送方式

● 实现方法:

1. 当外设准备好，向CPU发出中断请求
2. CPU在满足响应中断的条件下，发出中断响应信号；
3. CPU暂停当前的程序，转去执行中断服务程序，
完成与外设的数据传送；
4. CPU从中断服务程序返回，继续执行被中断的程序

中断方式下 CPU执行程序流程



- **中断传送方式的特点：**

1. **CPU和外设大部分时间处在并行工作状态，**

只在**CPU**响应外设的中断申请后，

进入数据传送的过程

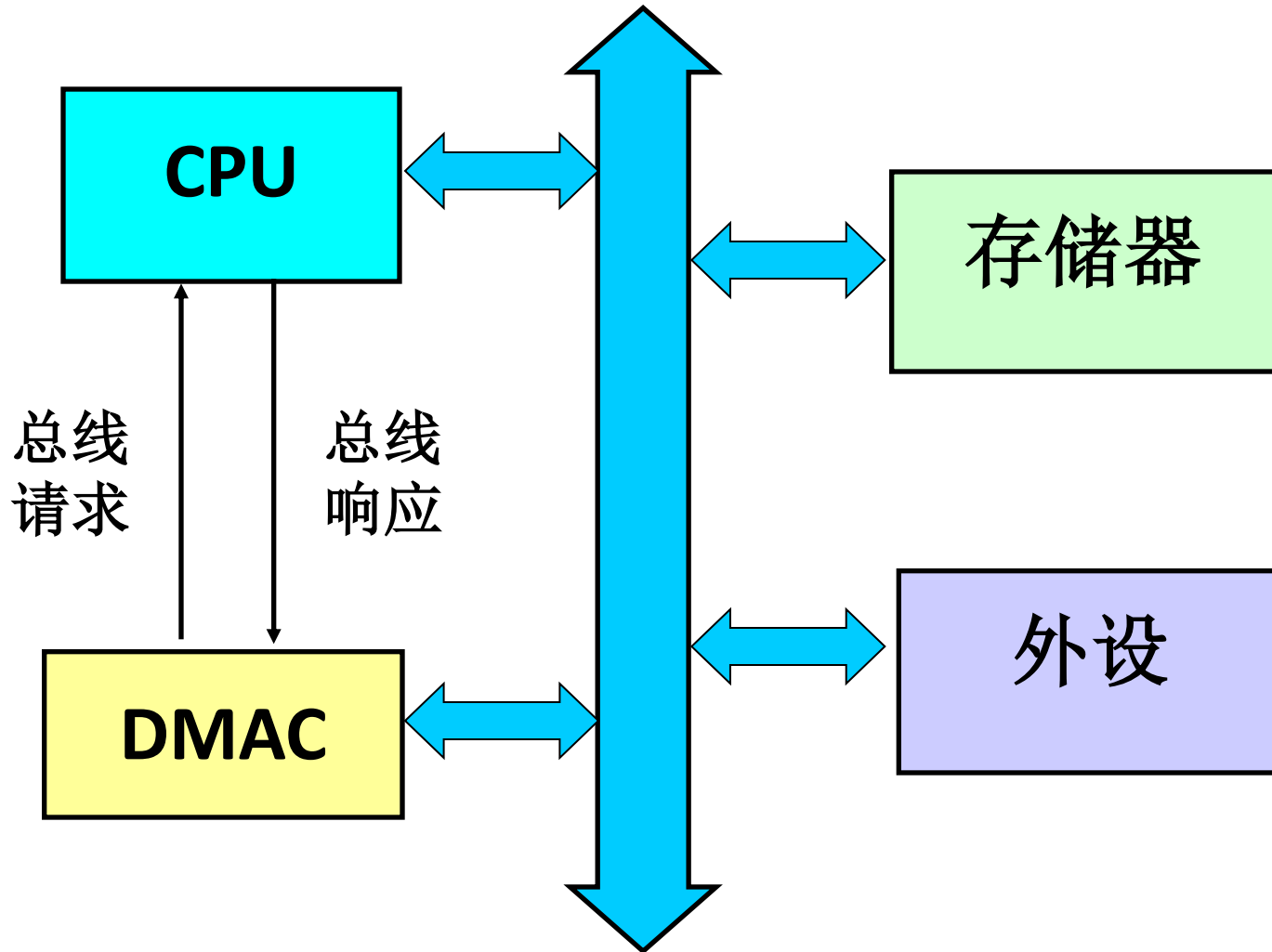
2. **中断传送方式提高了CPU的效率**

四、DMA 传送方式(直接存储器存取方式)

● 实现方法:

1. 由专用接口芯片**DMA控制器 (称DMAC)** 控制传送过程,
2. 当外设需传送数据时, 通过 **DMAC**向**CPU**发出总线请求;
3. **CPU**发出总线响应信号, 释放总线;
4. **DMAC**接管总线, 控制外设、存储器之间直接数据传送

DMA 传送方式过程



● DMA传送方式的特点

1. 外设和存储器之间，直接进行数据传送，不通过CPU, 传送效率高。

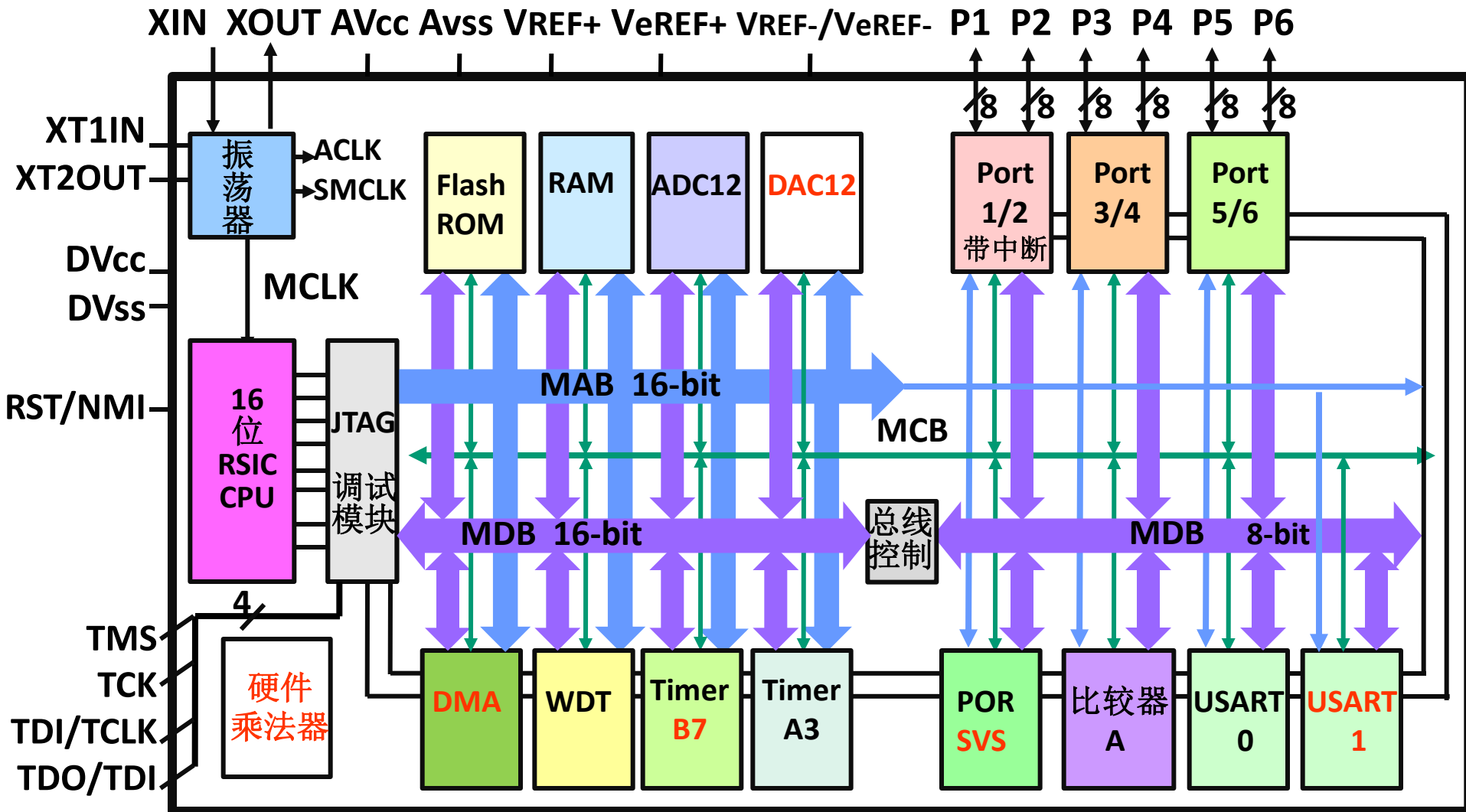
适用于在存储器与高速外设、

或两个高速外设之间进行大批量数据传送。

2. 电路结构复杂，硬件开销较大。



MSP430F16x内含DMA控制器



MSP430F6638内含DMA控制器

Functional Block Diagram, MSP430F6638, MSP430F6637, MSP430F6636

