《嵌入式系统》

2. 嵌入式硬件基础与系统设计

清華大學
Tsinghua University

提纲

- 嵌入式处理器介绍
- 体系结构
- 外围设备
- 51单片机介绍
- 嵌入式系统设计

清華大學
Tsinghua University

提纲

- 嵌入式处理器介绍
- 体系结构
- 外围设备
- 51单片机介绍
- 嵌入式系统设计

嵌入式处理器分类

微控制器(Microcontroller Unit, MCU)

微处理器(Microprocessor Unit, MPU)

DSP (Digital Signal Processor)

片上系统(System On Chip, SoC)

嵌入式微控制器（MCU）

- 微控制器（MCU）是目前嵌入式系统工业的主流。
- 其外设资源丰富，适合于控制，因此称为微控制器。内部常集成 Flash、EEPROM、ROM、EPROM、RAM、总线、总线逻辑、定时/计数器、看门狗、I/O、串口、A/D、D/A等各种必要功能和外设。
- 最大特点是单片化（又称单片机）：体积小，功耗和成本低、可靠性高。
- 代表性的通用系列包括MCS-51、MCS-96、AVR等。

单片机的特点

- **优异的性能价格比**
  - MSP430系列MCU的售价在$10上下
- **体积小、集成度高、可靠性高**
  - 单片机内集成了多种功能部件
  - 程序采取固化存储
- **控制功能强**
  - 强大的I/O逻辑操作及位处理功能（直接对RAM中的位操作，如开关量）

嵌入式微控制器（MCU）

- 1971年Intel开发出第一片具有4位总线结构的微控制器4004，主要用于玩具、家电，电子控制等。
- 1976年Intel推出功能相对完备的单片机8048；Motorola同时推出68HC05，Zilog推出Z80系列。
- 1980年，Intel在8048基础上研制成功8051，MCS-51单片机是小规模嵌入式系统中用得最多的微控制器。

常见MCS-51系列单片机

系列		片内存储器			定时器计数器	并行I/O口	串行I/O口	中断源	制造工艺
		片内ROM	片内EPROM	片内RAM					
MCS-51子系列	8031 80C31	0	0	128	2X16位	4X8位	1	5	HMOS
									CHMOS
	8051 80C51	4k	0	128					HMOS
									CHMOS
	8751 87C51	0	4k	128					HMOS
									CHMOS
MCS-52子系列	8032 80C32	0	0	256	3X16位	4X8位	1	6	HMOS
								7	CHMOS
	8052 80C52	8K	0	256				6	HMOS
								7	CHMOS
	8752 87C52	0	8K	256				6	HMOS
								7	CHMOS

嵌入式微处理器

- 嵌入式微处理器（MPU）：只保留和嵌入式应用紧密相关的硬件，去除其他功能部分，这样可以最低的功耗和资源实现嵌入式应用的要求。

- 目前主要的嵌入式处理器有 PowerPC、MIPS、68000、ARM系列等

典型的32位嵌入式微处理器

- ARM系列是应用较广泛的32位微处理器

MIPS

- MIPS： Microprocessor without Interlocked Pipeline Stages。于1984年成立。
- 1999年，MIPS公司发布了MIPS32和MIPS64体系结构标准。
- 龙芯采用MIPS指令架构。

MIPS TECHNOLOGIES

PowerPC体系结构

- PowerPC： Performance Optimization With Enhanced RISC – Performance Computing;
- AIM：Apple–IBM–Motorola 于1992年推出首款RISC处理器 PowerPC 601
- PowerPC 目前的管理者

  www.Power.org

MCU和MPU比较

- **MPU的优缺点**
  - 优点：可以根据需要灵活配置硬件。
  - 缺点：系统体积大、价格偏高。
- **MCU的优缺点**
  - 优点：体积小、价格便宜、功耗低。
  - 缺点：ROM、RAM和I/O端口等硬件的大小和数量配置有限，不易扩展，选型很重要。

数字信号处理器——DSP

- 70年代以来出现了许多DSP算法，但由于专门的DSP处理器还未出现，所以这些算法只能在通用CPU上实现。1982年诞生了首枚DSP芯片。
- DSP是专门用于信号处理方面的处理器，常见的应用场合为：数字滤波、FFT、谱分析，图像、音频和视频数据处理等。
- DSP对系统结构和指令进行了特殊设计，以适合执行DSP算法。例如：乘/累加操作 a = a + b*c，在卷积运算、点积运算、矩阵运算、数字滤波器运算中常用;

嵌入式DSP处理器

- 代表性产品：Texas Instruments的 TMS320系列和 Motorola的DSP56000系列。

嵌入式片上系统(SoC)

- SoC（System on Chip）是一种基于IP核的嵌入式系统设计技术。IP核，全称知识产权核（intellectual property core），是指某一方提供的、形式为逻辑单元、芯片设计的可重用模块。IP核通常已经通过了设计验证，设计人员以IP核为基础进行设计，可以缩短设计所需的周期。

- SoC设计思路
  - 将各种通用处理器内核和许多其它外设做成标准器库；
  - 用户用HDL语言定义和描述整个应用系统，在仿真通过后就可以将设计图交给半导体工厂制作样品；
  - 整个系统可集成到一块或几块芯片中。

对比

	嵌入式微处理器	嵌入式微控制器	嵌入式DSP处理器	嵌入式片上系统
硬件尺寸(包括外围)	大	小	小	最小
功耗	大	小	中	中
开发难度	小	大	大	大
软件移植性	好	**差**	**差**	**差**
成本	高	最低	低	中
性能	强	弱	较强	较强
应用领域	通用	较通用低端	专用	较通用高端
网络能力	强	弱	较弱	强
实时性	差	好	好	一般

对比

	嵌入式微处理器	嵌入式微控制器	嵌入式DSP处理器	嵌入式片上系统
硬件尺寸(包括外围)	大	小	小	最小
功耗				
开发难度				
软件移植性				
成本				
性能	强	弱	较强	较强
应用领域	通用	较通用低端	专用	较通用高端
网络能力	强	弱	较弱	强
实时性	差	好	好	一般

从计算机角度看

- 单片机功能太简单, 性能太差
- DSP太专用, 可以看成一个外设
- 嵌入式处理器与SoC是主要发展方向

清華大學
Tsinghua University

提纲

- 嵌入式处理器介绍
- 体系结构
- 外围设备
- 51单片机介绍
- 嵌入式系统设计

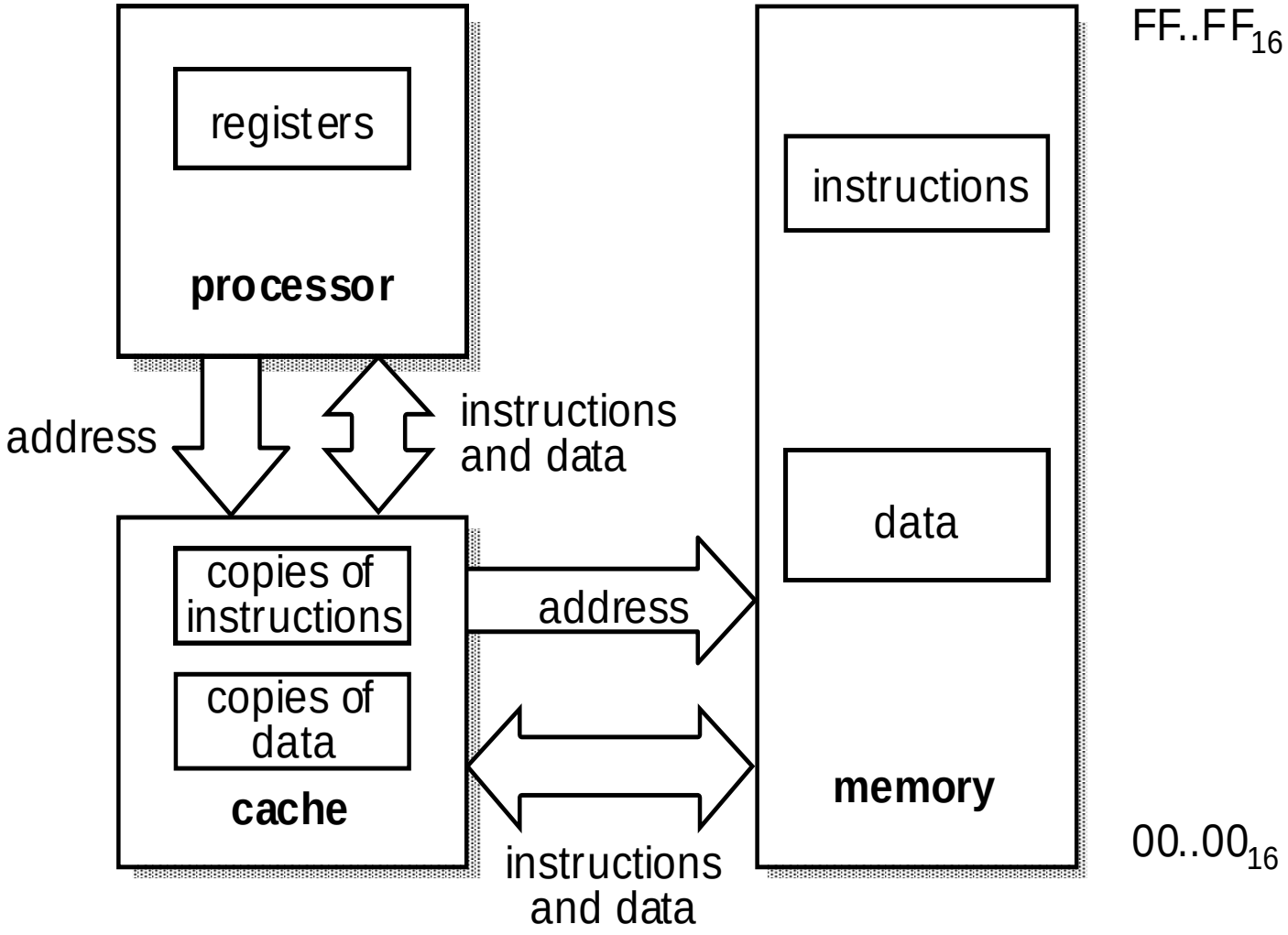
冯·诺依曼与奥本海默

冯·诺依曼(Von-Neumann)体系结构

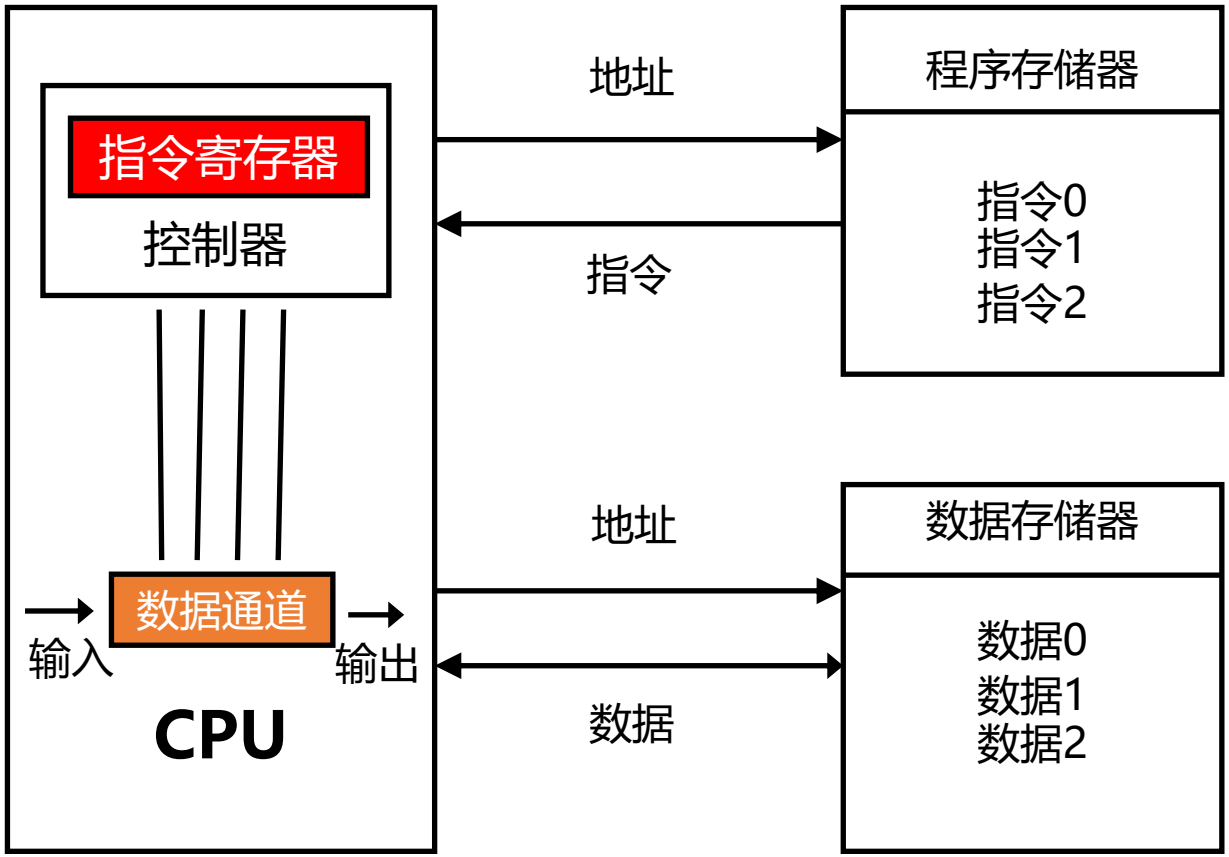
1）数据与指令存储在同一存储空间

2) 冯·诺依曼结构又称作普林斯顿体系结构

3）X86，ARM7TDMI是冯·诺依曼体系



哈佛体系结构


指令寄存器
控制器
数据通道
输入
输出
CPU
地址
指令
地址
数据
程序存储器
指令0
指令1
指令2
数据存储器
数据0
数据1
数据2

哈佛体系结构

1）程序存储器与数据存储器分开

2）大多数DSP都是哈佛结构，适合于数字信号处理

3）其他： ARM9+， 摩托罗拉MC68系列、Zilog的Z8系列

4）MCS-51是哈佛结构：

MOV和MOVX分别访问片内外数据存储器；

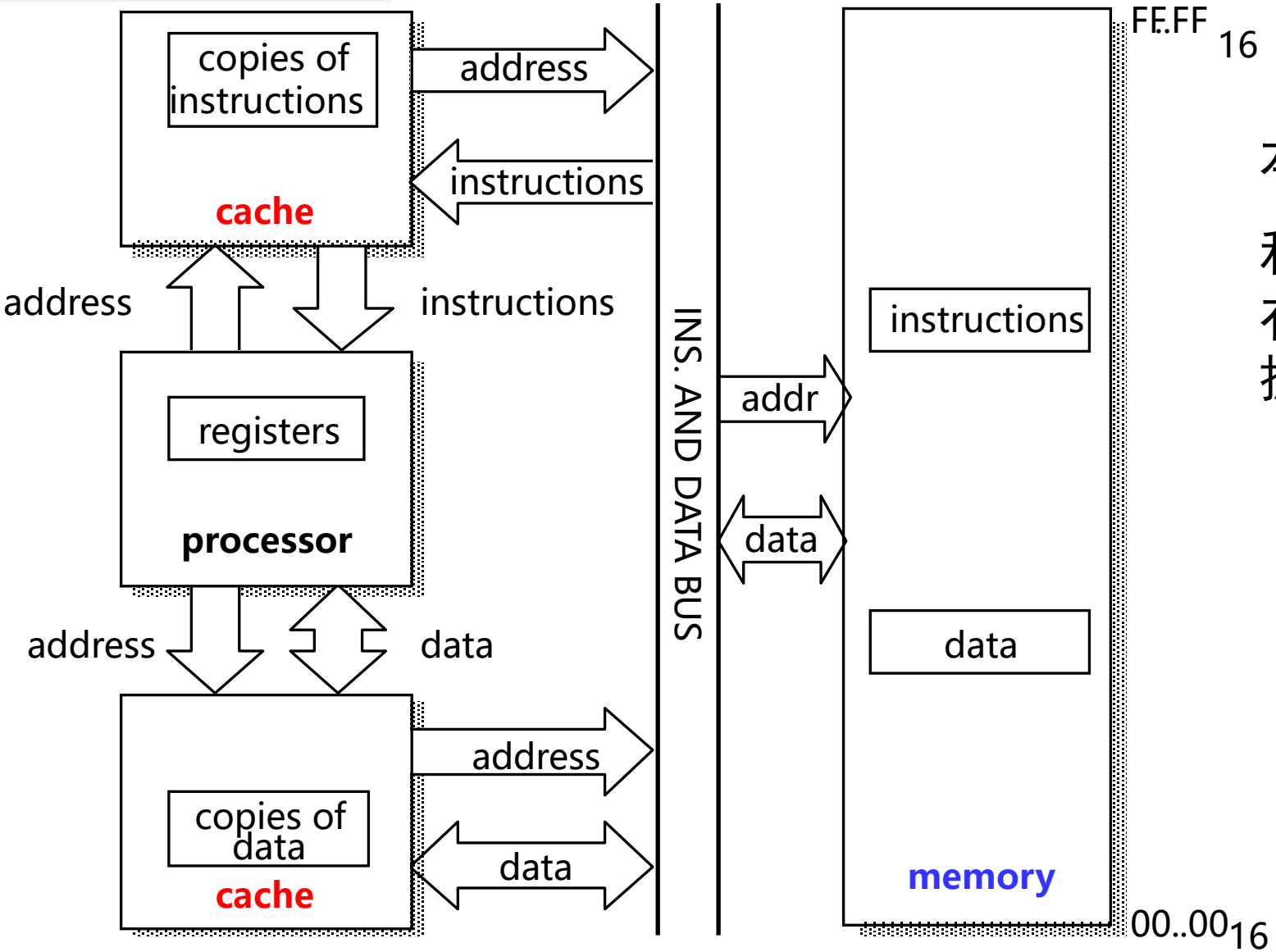
MOVC访问代码存储器。 但总线分时复用。

MCS-51也支持冯·诺依曼体系：通过改造控制电路（仅添加简单的门电路），程序和数据可被映射到同一个64K空间。这是早期51仿真器中的一项重要技术。

哈佛结构 v. 冯·诺依曼结构

- 通用计算机系统：操作系统对不同应用软件分配不同的数据和代码空间。冯·诺依曼结构因为统一编址可最大限度利用资源，而哈佛结构在此情形下则会对存储器资源产生浪费。
- 嵌入式系统：系统执行单一任务，程序固化在硬件里。代码不可修改，降低了被攻击的风险，提高了可靠性和安全性。所以，哈佛结构更适合于那些程序固化、任务相对简单的控制系统。
- 技术在相互融合,例如：

这个是什么结构?



本质区别：

和CPU之间有不同的数据通路。

缓存的两种写模式：直写（write-through）和回写（write-back）

- **基本定律：**在任意时刻，任意级别缓存中的缓存段的内容，等同于它对应的内存中的内容。

- **直写**：要更新缓存时，同时更新内存（或下一级缓存）中的内容，使得缓存与内存的内容保持一致。

- **回写**：缓存不会立即把写操作传递到下一级，而是仅修改本级缓存中的数据，并且把对应的缓存段标记为“脏”段。脏段会触发回写，也就是把里面的内容写到对应的内存（或下一级缓存）中。回写后，脏段又变“干净”了。当一个脏段被丢弃的时候，总是先要进行一次回写。回写所遵循的规律有点不同。

- **回写定律：**当所有的脏段被回写后，任意级别缓存中的缓存段的内容，等同于它对应的内存中的内容。

清華大學
Tsinghua University

提纲

- 嵌入式处理器介绍
- 体系结构
- 外围设备
- 51单片机介绍
- 嵌入式系统设计

外围硬件设备

- **片外总线：连接系统各部件，传输数据和控制信号。**
  - ESIA、PCI、PCIe、I2C …
  - 专用总线: 工业控制用的CompactPCI总线,
  - 汽车用CAN总线 …
- **存储器：**Flash、EPROM …
- **I/O设备：** A/D、D/A、中断控制器、LCD …
- **通讯设备**
  - 有线通讯：IEEE1394、USB
  - 无线通讯：IrDA、Bluetooth、802.11b/g

总线通信

总线

- 总线是把CPU与存储器、I/O设备相连接的信息通道，总线并不仅仅指的是一束信号线，其核心是相应的通信协议。

- 按照使用场合的不同，总线分成芯片级总线（CPU总线）、板卡级总线（内总线）和 系统级总线（外总线）。

- 按照工作模式不同可分为：串行和并行总线

总线

- **串行总线**
  - 数据在数据线上按位传输
  - 只需要一根数据线，线路的成本低，适合于低速外设、远距离的数据传输
  - 例如：键盘、鼠标、调制解调器
- **并行总线**
  - 并行总线的数据在数据线上同时有多位一起传送，每一位要有一根数据线，因此有多根数据线
  - 并行传输比串行传输速度要快得多，但需要更多的数据线

总线的若干概念

❑总线的主要参数有

带宽: 指单位时间内可传送的数据量，即每秒钟传送多少Mb的最大稳态数据传输率。

总线的带宽=总线的工作频率*总线的位宽

位宽: 指能同时传送的数据位数，即常说的32位、64位等总线宽度的概念。

工作时钟频率:总线的工作时钟频率以MHz为单位，工作频率越高，总线工作速度越快，总线带宽越宽。

波特率与比特率

- **波特：**波特(Baud)是计量单位，表示每秒时间内信号调制状态改变的次数
- **码元：**携带数据信息的信号单元，指在数字信道中传送数字信号的一个波形符号
- **波特率（码元传输速率）：**每秒钟通过信道传输的码元数，单位波特
- **比特率（Bit Rate）：**通信线路(单位时间内传输的信息量，即每秒能传输的二进制位数，单位为位/秒（bps）

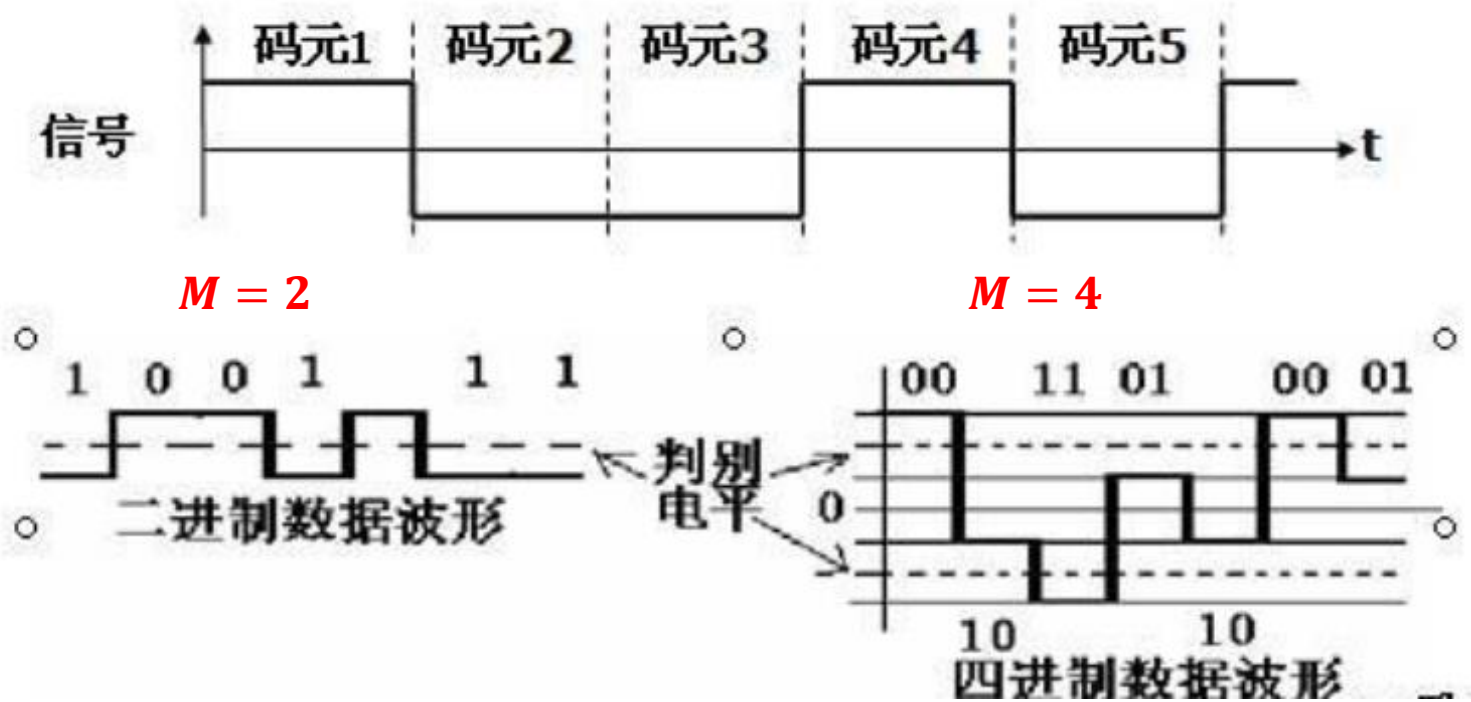
波特率与比特率

❑**波特率与比特率之间的关系：**

$$R_b = R_B \log_2 M$$

R_b**为比特率，** R_B**为波特率，** M**为1个码元代表的所有可能状态数**

一个码元可能携带多个二进制数据位数



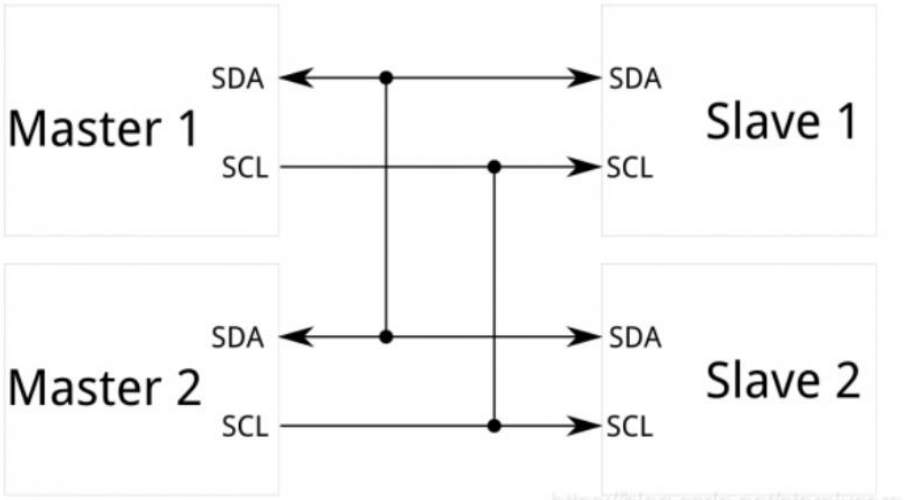
串行外设接口（SPI）

- SPI是一种高速的，全双工，同步的通信总线，它以主从方式工作，这种模式通常有一个主设备和一个或多个从设备，需要至少4根线：

1. MISO– Master Input Slave Output,主设备数据输入，从设备数据输出；
2. MOSI– Master Output Slave Input，主设备数据输出，从设备数据输入；
3. SCLK – Serial Clock，时钟信号，由主设备产生；
4. CS – Chip Select，从设备使能信号，由主设备控制。

I²C总线

- I²C（Inter-Integrated Circuit）总线是一种简单的双向二进制同步串行总线，它只需要两根线即可在连接于总线上的器件之间传送信息。
  - SDA：串行数据线
  - SCL：串行时钟线



USB总线

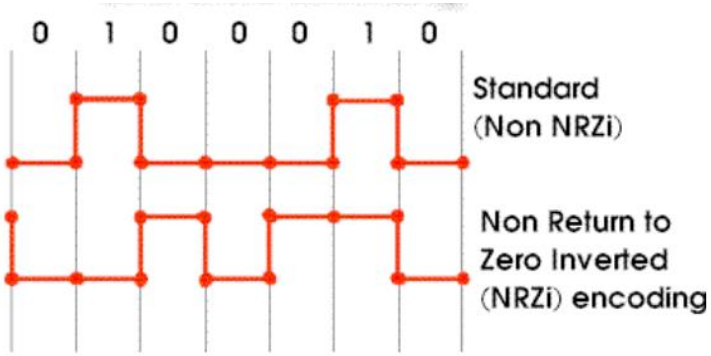
- 通用串行总线接口USB（Universal Serial Bus），是一种单极性、差分、倒转不归零NRZI（Non Return Zero Inverted）编码，半双工串行数据传输的，用于将USB外围设备连接到主机的外部总线结构。
- USB用一个4针（USB3.0标准为9针）插头作为标准插头，采用菊花链形式可以把所有的外设连接起来，最多可以连接127个外部设备，并且不会损失带宽。

NRZI编码

- 归零(RZ)编码

浪费带宽在“归零上”

- NRZI编码



电平翻转代表逻辑0，
电平保持不变代表逻辑1

CAN串行总线

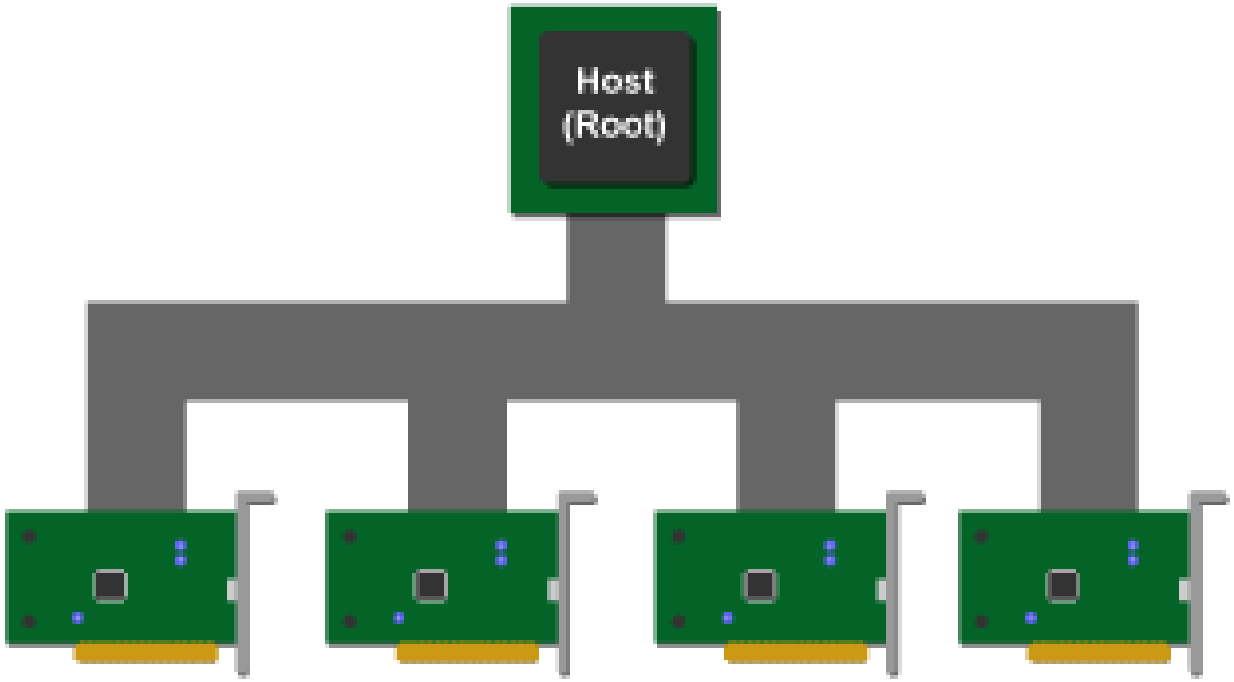
- CAN是控制器局域网络（Controller Area Network）的简称，由以研发和生产汽车电子品著称的德国BOSCH公司开发，提出CAN总线的最初动机就是为了解决现代汽车中庞大的电子控制装置之间的通讯，减少不断增加的信号线。
- CAN协议共分为二层：物理层和数据链路层。物理层定义了信号实际的发送方式、位时序、位的编码方式及同步的步骤。数据链路层将物理层收到的信号组织成有意义的消息，并提供传送错误控制等传输控制的流程。

PCI ——并行总线

- Peripheral Component Interconnect（外设部件互连标准）
- 1991 年下半年，Intel 首先提出了PCI 的概念。
- Intel联合IBM 、DEC 、Compaq、HP 等100 多家公司成立PCI-SIG　Peripheral Component Interconnect Special Interest Group (外设部件互连专业组) 。
- 93年：　PCI 2.0。
- 98年：PCI-X，位宽增至64，时钟增至133MHz。性能提高，成本增加：位宽和降噪。

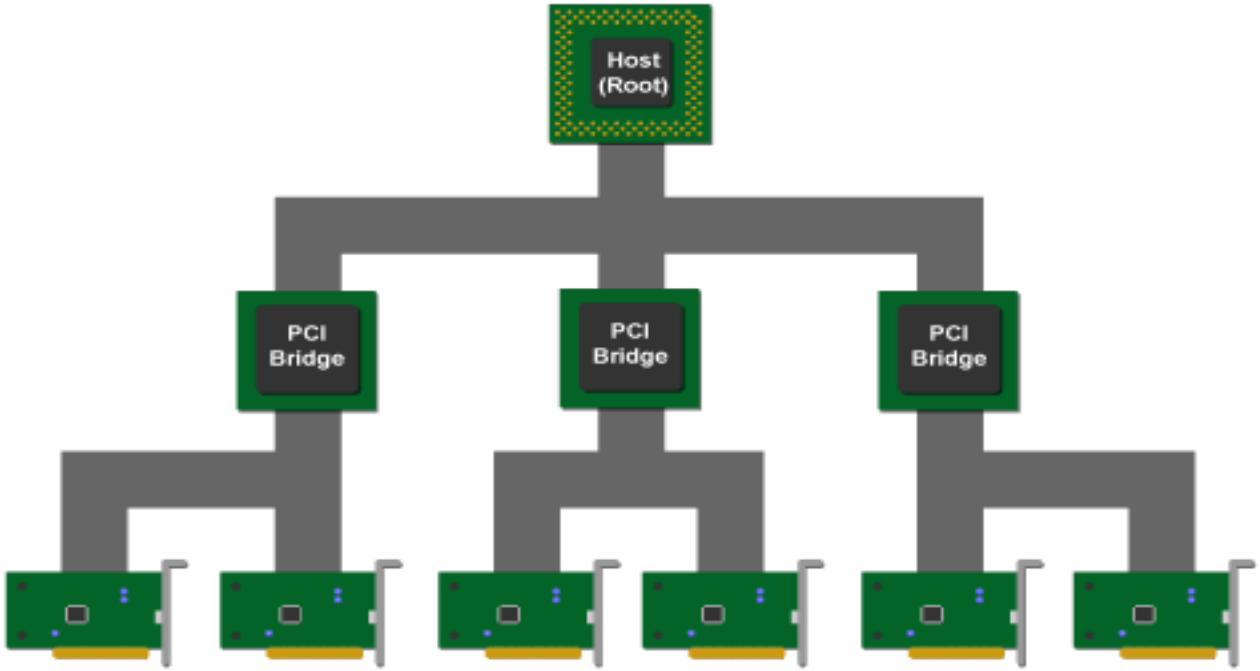
PCI采用共享并行总线

- 缺点1：每次仅一个PCI设备通信；



PCI采用共享并行总线

- 缺点2：同时监听总线上连接的PCI设备给总线带来噪声。结果使最多可连接的设备数量受限，这需要PCI桥来解决。



PCIe——串行总线

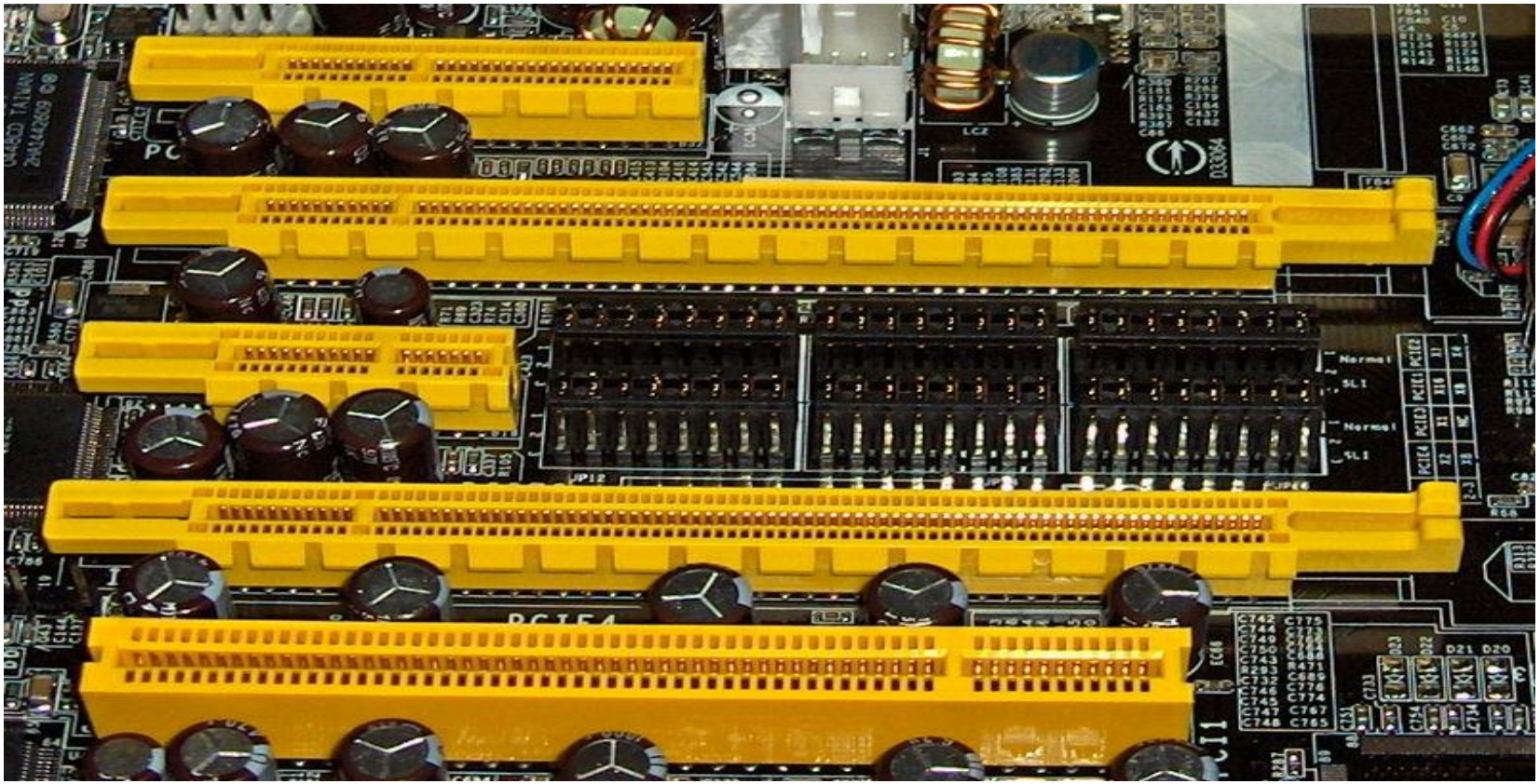
- 2002年，PCIe （或PCI-E）串行总线：

  Peripheral Component Interconnect Express 。

- 来自LAN设计的启发:
  - **Hub vs. Switch，**共享带宽vs专有通道，半双工 vs. 全双工
- PCIe使用交换机控制多个点到点的串行连接。每个设备都有专用连接，设备间不必共享带宽。
- PCIe采用与PCI相同的协议识别设备和连接， 所以PCIe与PCI软件兼容，但插槽本身并不兼容（见后图）。

PCIe——串行总线

由上而下：PCIe x4，x16，x1，与x16插槽，最下面为32-bit PCI插槽。



PCIe——串行总线

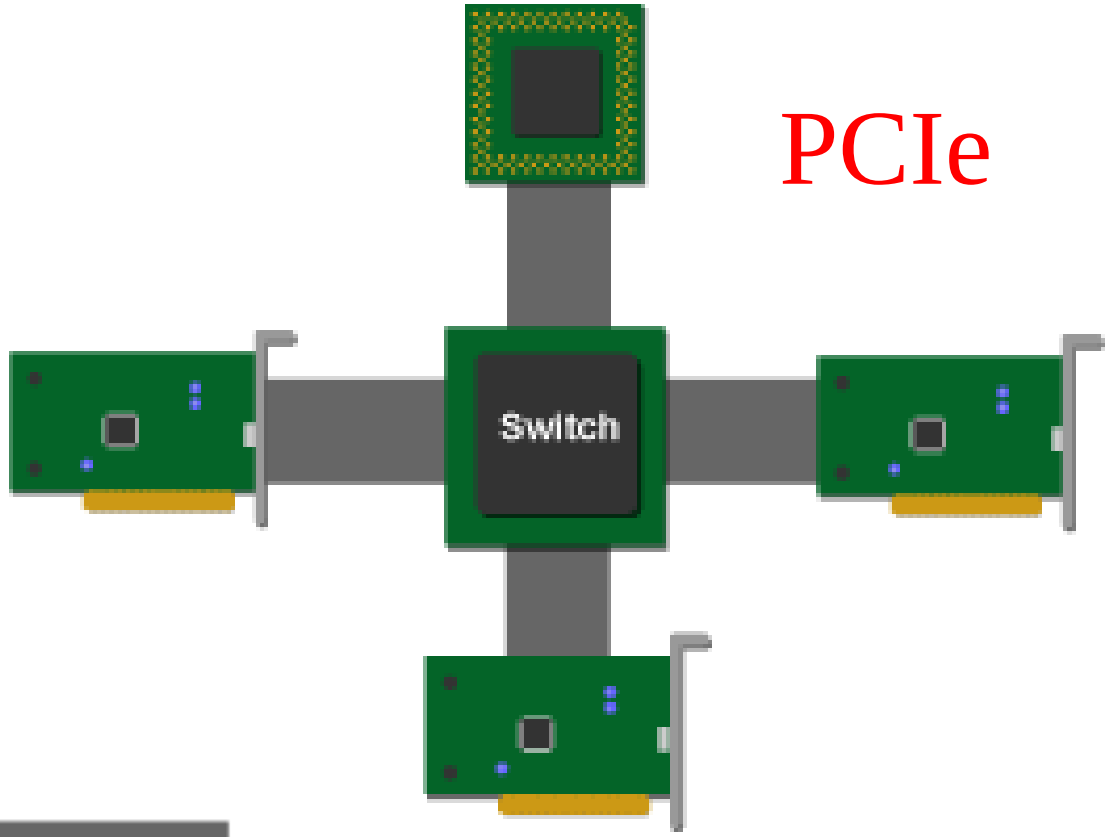
问：为什么现在各类并行总线反而被串行总线接替呢？

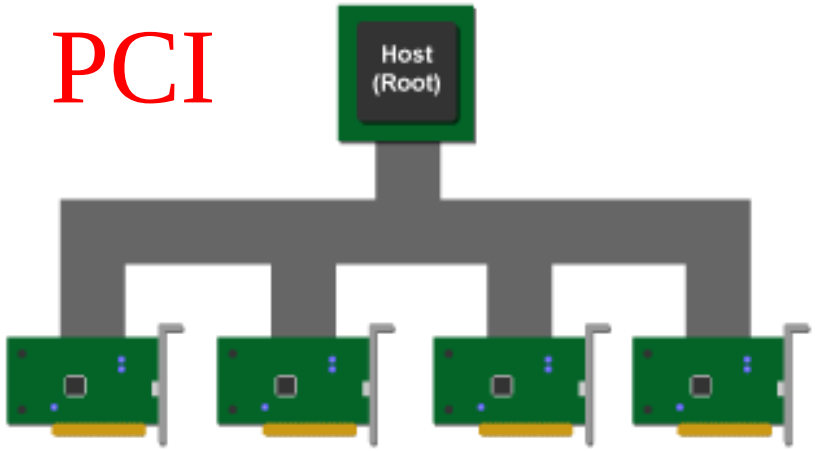
答：并行总线虽然一次可以传输多位数据，但信号间的同步和干扰问题不易解决，频率越高、位宽越大，问题越严重，这限制了并行总线带宽的提高。

串行总线不存在这些问题，总线频率可以大幅提高。

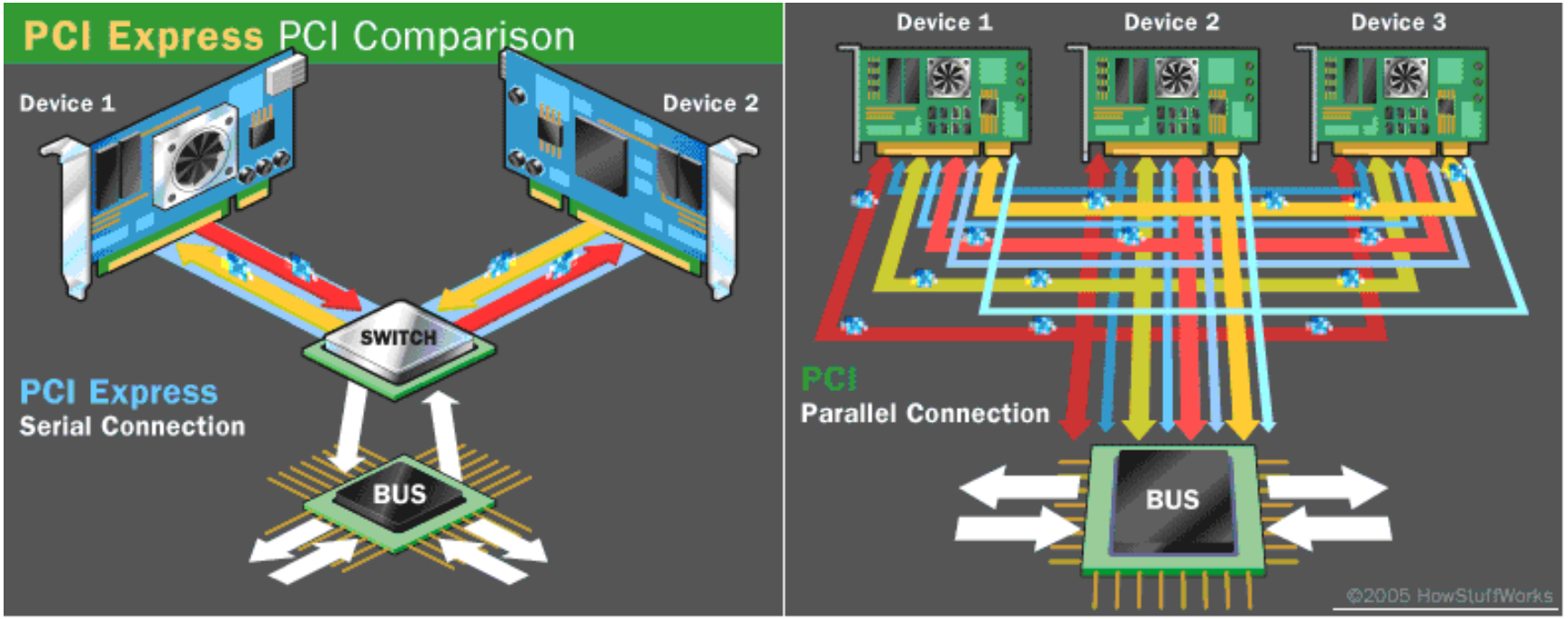
串行总线常采用多条各自独立的通道实现更高带宽，表面看它和并行总线类似，但在通道内部它是以串行原理工作的。

PCIe 点对点拓扑



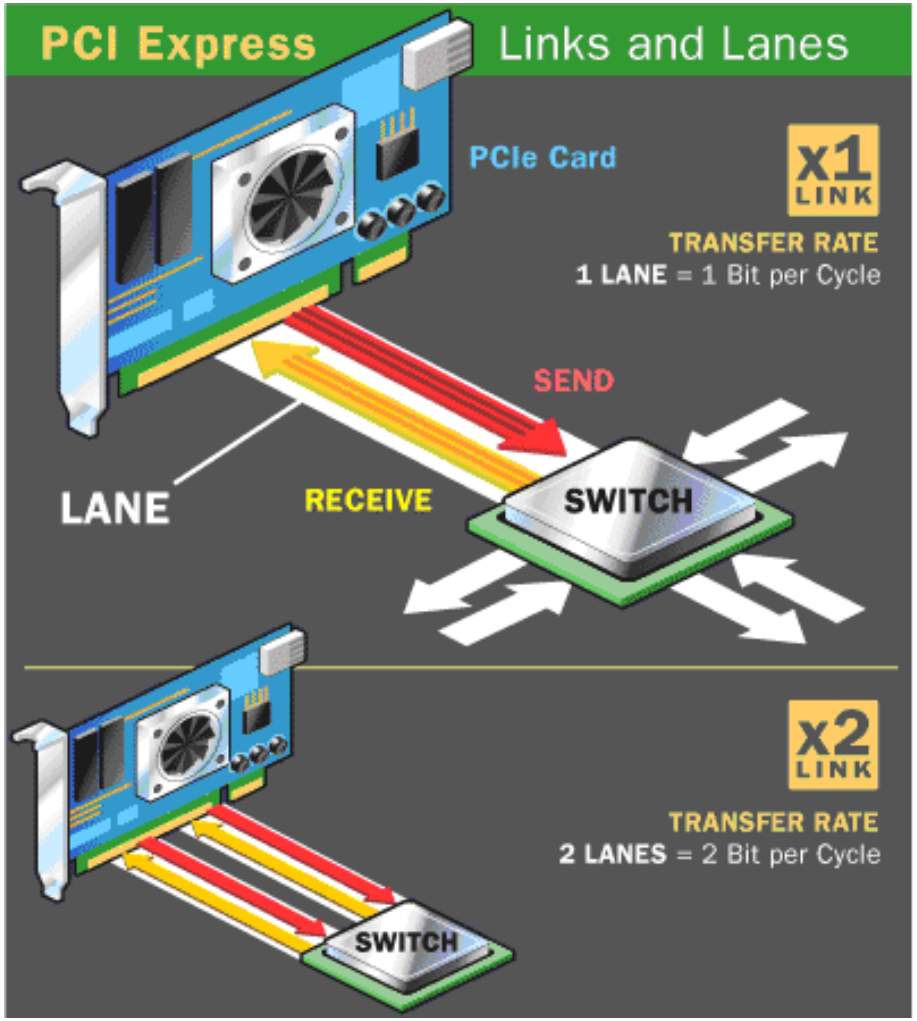


PCIe和PCI比较


PCI Express PCI Comparison
Device 1
Device 2
SWITCH
BUS
PCI Express
Serial Connection
Device 1
Device 2
Device 3
PCI
Parallel Connection
BUS
©2005 HowStuffWorks

PCIe的通道

PCIe连接建立在一个双向 1比特点对点连接基础上，此连接这称为一个“传输通道”。



PCI 和 PCIe比较

PCI总线的优点：结构、设计简单。

缺点：1) 并行总线无法连接太多设备，扩展性差，线间干扰大；

2) 连接多个设备时，总线有效带宽将大幅降低；

PCIe总线的优点：

1) 串行点对点传输，每个传输通道独享带宽。
2) 支持多通道连接，高带宽；
3) 新特征：电源管理、QoS（优先级高的包优先）等；
4) 软件兼容PCI总线；
5) 总线带宽提高同时，减少了硬件PIN的数量，降低了成本。

串口通信

UART/USART串行接口

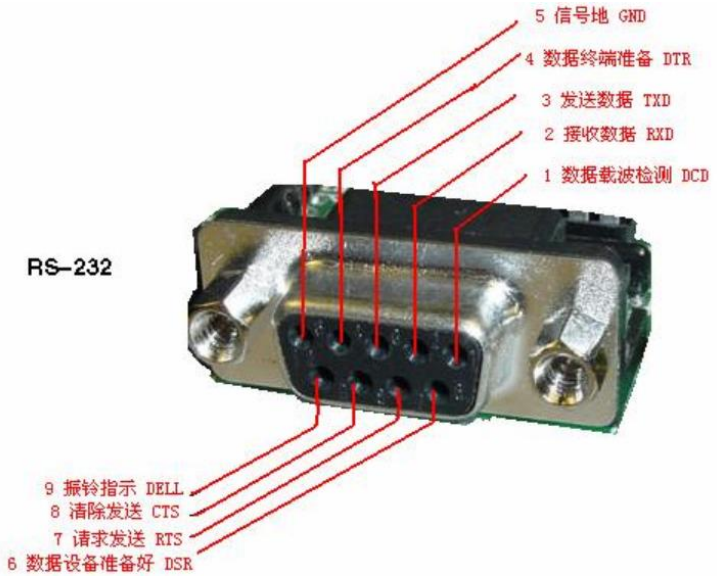
- UART（Universal Asynchronous Receiver Transmitter）即“通用异步收/发器”，是各种计算机与大多数微控制器中集成的全双工、单极性、串行通信接口模块，按字节传输数据，一条信号线TXD发送数据，以一条信号线RXD接收数据。
- USART:(Universal Synchronous/Asynchronous Receiver/Transmitter) 通用同步/异步 接收/发送器。全双工通用同步/异步串行收发模块，该接口是一个高度灵活的串行通信设备。
- 在UART上追加同步方式的序列信号变换电路，得到USART。
- USART串口可用于微控制器与PC机、微控制器与微控制器之间的通信。

USART与UART的区别

- **数据的计时方式**
  - UART在内部生成其数据时钟，并使用起始位转换将该时钟与数据流同步。没有与数据相关的传入时钟信号，因此为了正确接收数据流，接收器需要提前知道波特率。
  - 而USART可以以同步模式运行。在这种模式下，链路将使用完全独立的线路来传送时钟信号，使得USART的数据速率比标准的UART高得多，最高可达4Mbps。
- UART只支持简单的基本协议，而UASRT能支持多种标准协议。
- USART通常比标准UART具有更多的功能。

RS232规格的串口

RS-232： PC机使用的串口， COM1或者COM2



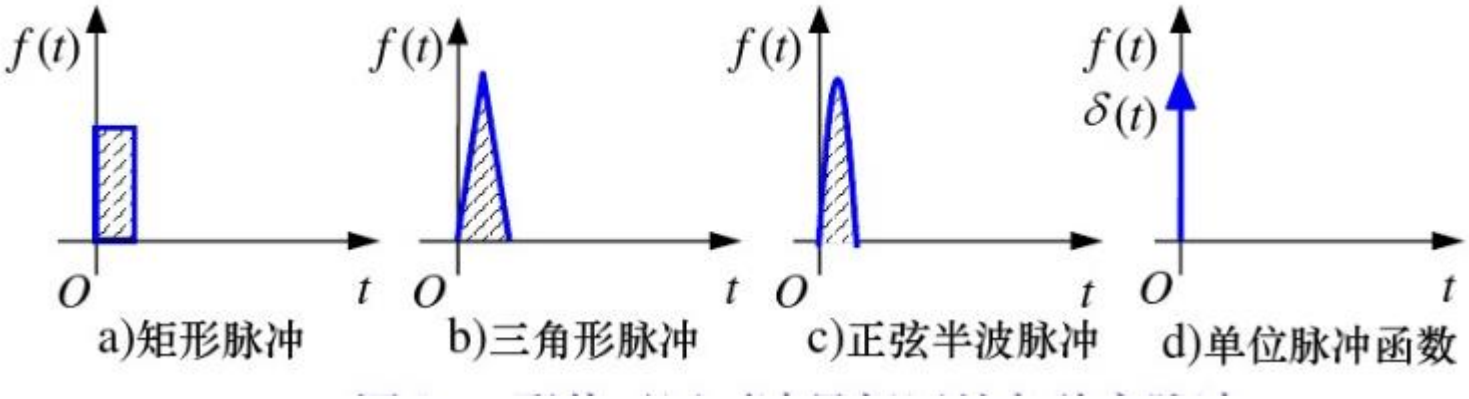
RS-232工业系统常用的有RS485。

定时器

- 定时器是微处理器内部专门设计的一部分电路，提供与定时相关的功能，它可用作：
  - 精准延时
  - 事件的时间安排: 设置预设值，到时产生中断，在中断服务程序中对相关事项进行处理
  - 测量事件之间的间隔
  - 实时时钟和看门狗时钟
  - 脉冲宽度调制（PWM）功能是一种对模拟信号电平进行数字编码的方法。利用定时器的计数功能实现对模拟输出的控制，可以产生周期和占空比可调的信号输出。注意PWM信号仍然是数字的。

脉冲宽度调制 (Pulse Width Modulation)

- **问题：** 因为在给定的任何时刻，满幅值的直流供电要么完全有(ON)，要么完全无(OFF)，如何实现“调速”？
- **原理：**
  - 面积等效原理--冲量相等而形状不同的窄脉冲加在具有惯性的环节上时，其效果基本相同。
  - 冲量：窄脉冲的面积



脉冲宽度调制 (Pulse Width Modulation)

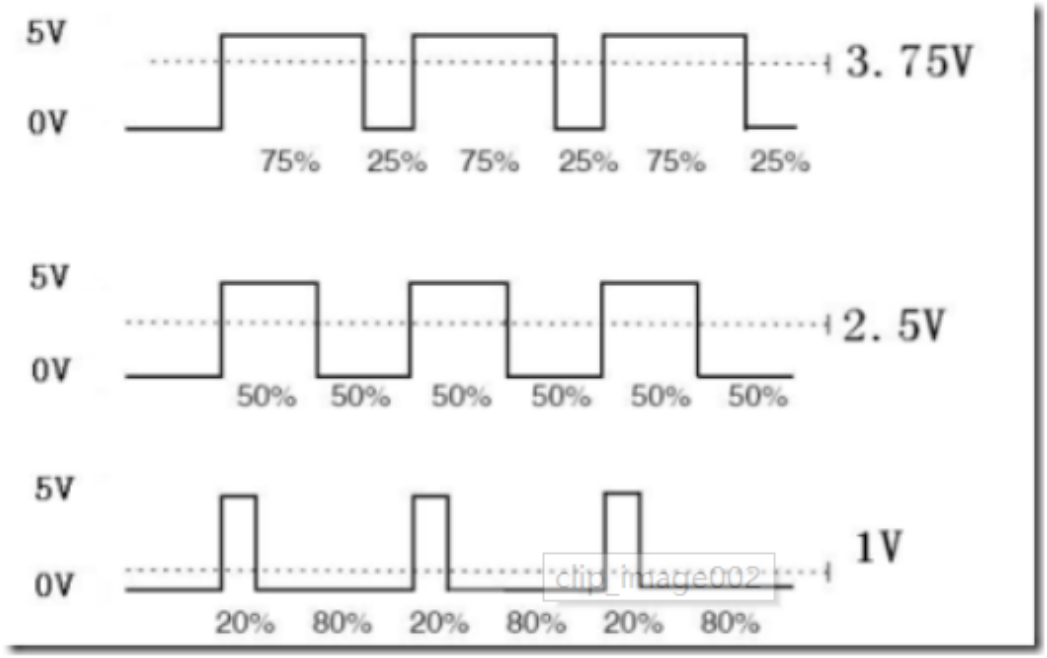
- 脉冲宽度：高电平或低电平持续的时间。
- 占空比：高电平在一个周期中所占的比例。
- PWM通过控制固定电压的直流电源的开关频率，从而改变负载两端的电压。
- 举例
  - 电机调速
  - LED呼吸灯

脉冲宽度调制 (Pulse Width Modulation)

- 只要带宽足够，任何模拟值都可以使用PWM 进行编码。

  输出电压=（接通时间/脉冲时间）*最大电压值

- 举例：满幅值的直流供电为5V，断电为0V



ADC模数转换器

- **模数转换器（Analog to Digital Converter ）**
  - 在采样系统中，首先通过传感器将不同类型的模拟信号变换为电信号，而外部电信号又分为电压信号与电流信号两种类型。最终都是转换为电压信号通过ADC量化。
- **ADC的主要技术指标**
  - 位数（转换结果的位数）、满量程电压、分辨率（使输出数字量变化一个相邻数码所需输入模拟电压的变化量 ）、转换速率、误差等等，还要根据实际测量的模拟量选择合适的ADC。

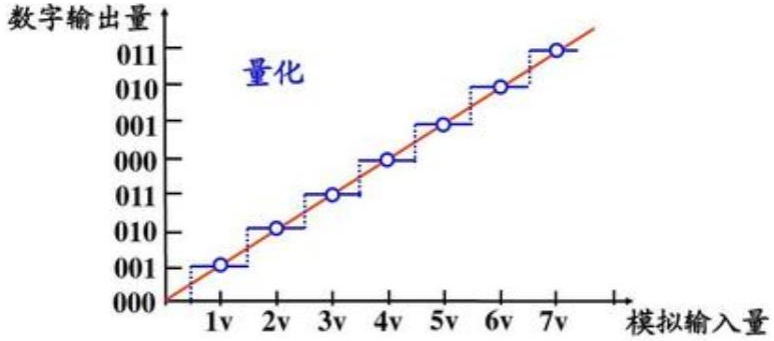
ADC转换过程

❑采样和保持

❑采样就是对连续变化的模拟信号进行定时测量，采样结束后再将该采样信号保持一段时间，使ADC有充分的时间进行AD转换。

❑量化和编码

❑量化就是把采样电压转化为某个最小单位电压的整数倍的过程，分成的等级称为量化级。编码就是用二进制编码表示量化后的电平。



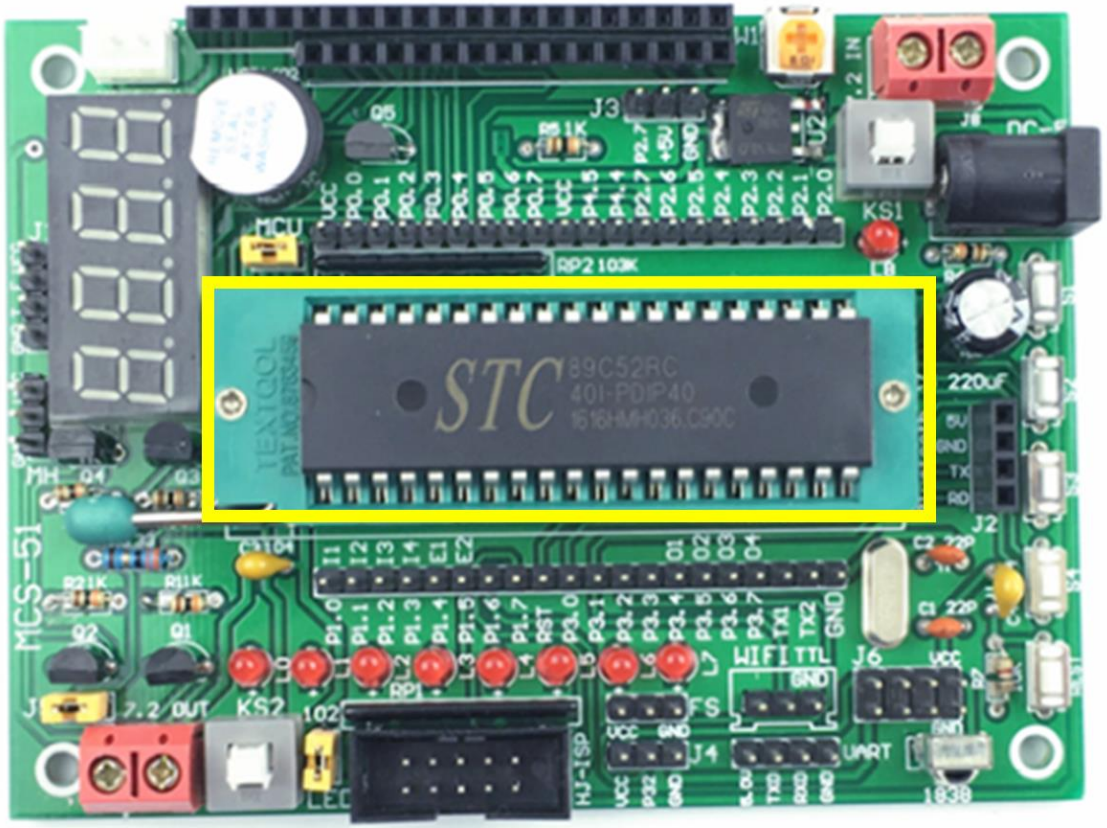
清華大學
Tsinghua University

提纲

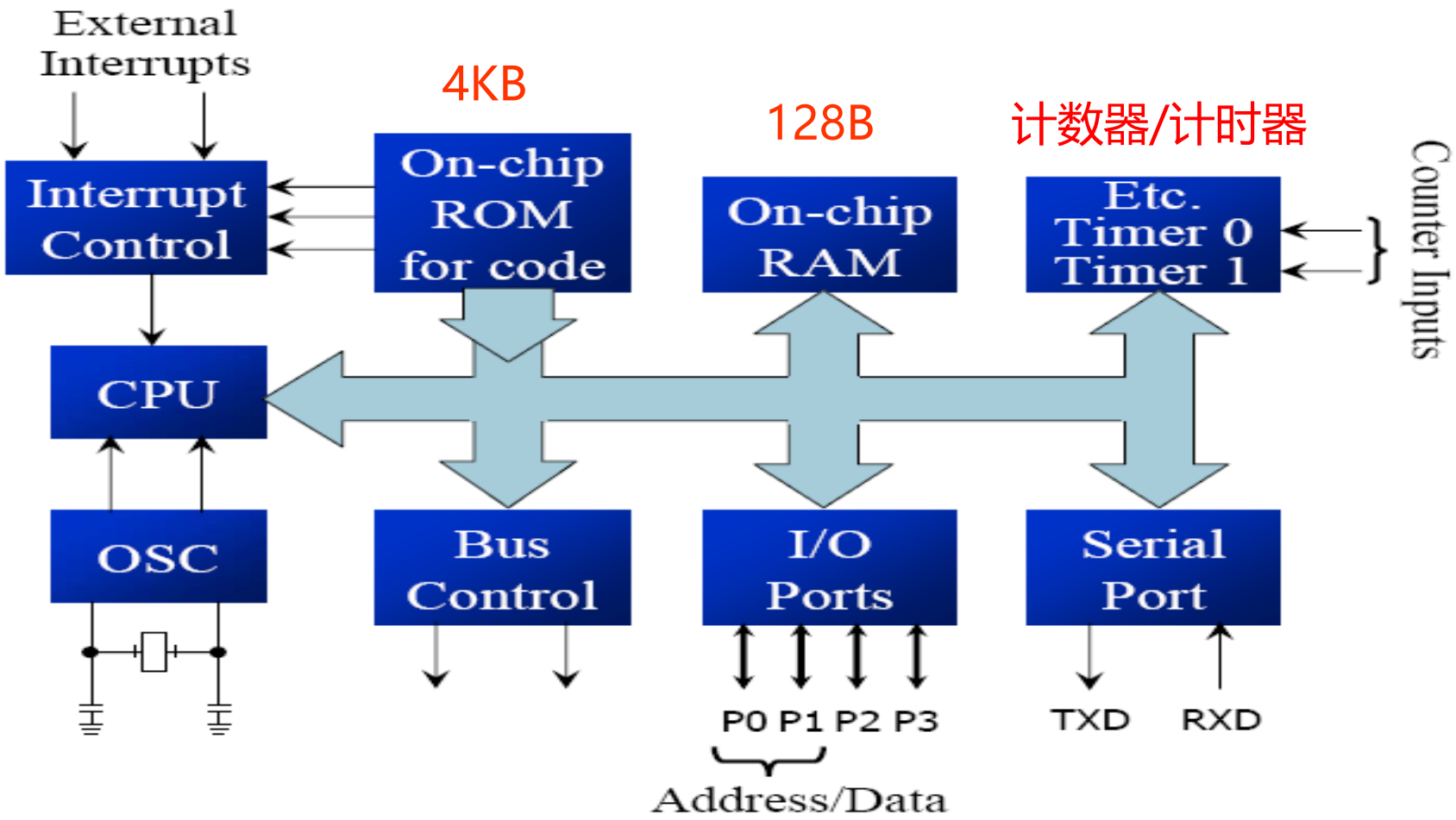
- 嵌入式处理器介绍
- 体系结构
- 外围设备
- 51单片机介绍
- 嵌入式系统设计

MCS-51实物图


P8051AH 0649
8736041
93X3903 Y15
©INTEL '80


STC
89C52RC
40I-PDIP40
MCS-51
MCU
KS1
KS2
WIFI
UART
J3
J4
J6
220uF

8051逻辑图

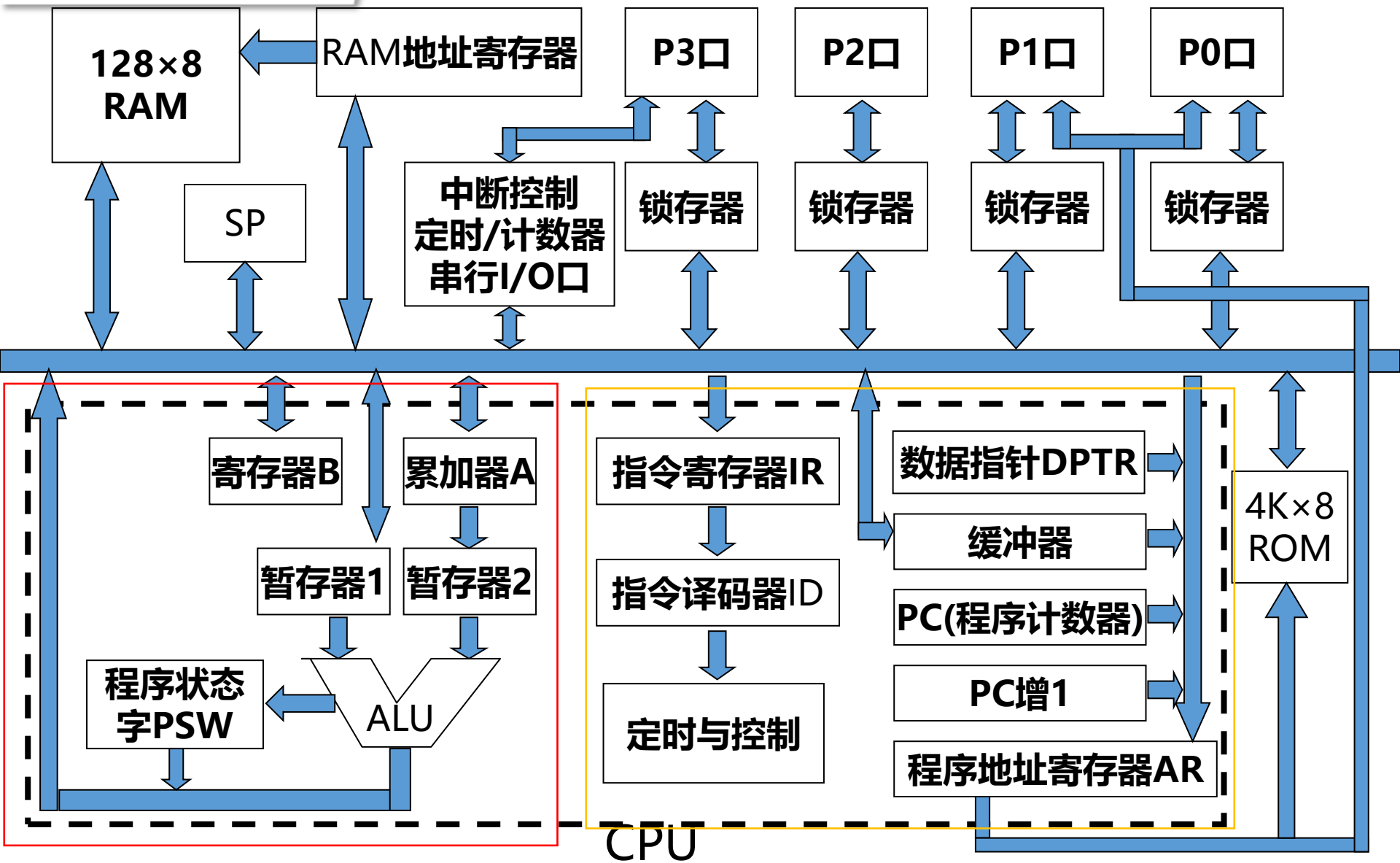


（石英晶振）振荡电路

MCS-51基本特性

- 面向控制的8位CPU和指令系统
- 4K字节的程序存储器ROM（或EPROM）
- 128字节的数据存储器RAM（实际上有256B空间）
- 可编程的并行I/O口：P0~P3,32位双向输入输出
- 1个全双工串行口I/O
- 两个16位定时器/计数器
- 5个中断源，2个中断优先级
- 1个片内时钟振荡器和时钟电路
- 可寻址64K字节程序存储器
- 可寻址64K字节外部数据存储器

8051内部结构图


128×8
RAM
RAM地址寄存器
P3口
P2口
P1口
P0口
SP
中断控制
定时/计数器
串行I/O口
锁存器
锁存器
锁存器
锁存器
寄存器B
累加器A
暂存器1
暂存器2
程序状态
字PSW
ALU
指令寄存器IR
指令译码器ID
定时与控制
数据指针DPTR
缓冲器
PC(程序计数器)
PC增1
程序地址寄存器AR
4K×8
ROM
CPU

MCS-51的CPU组成

- CPU
  - 运算器
    - 算术/逻辑部件ALU(Arithmetic Logic Unit)
    - 累加器**ACC** (Accumulator)
    - 程序状态字寄存器**PSW** (Program Status Word)
    - 暂存寄存器**1、2**
    - 寄存器**B**
  - 控制器
    - 指令寄存器IR (用户不可见)
    - 指令译码器ID (用户不可见)
    - 程序计数器**PC**
    - 定时控制与条件转移逻辑电路

运算器

- ALU：算术逻辑部件
  - 由加法器和其他逻辑电路组成
  - 算数运算：对8位数据进行算数加减乘除
  - 逻辑运算：逻辑与、或、异或、取反等
  - 循环移位，位操作等
- ACC：累加器（也简称A）
  - 一个8位寄存器，通过暂存器与ALU相连
  - 工作最频繁的寄存器
    - 存放算术运算和逻辑运算中的操作数
    - 存放ALU的运算结果
    - 变址寻址中的变址寄存器
  - 访问ACC的寻址方式：可字节寻址(E0H)，也可位寻址(E0H~E7H)

运算器中的其他寄存器

- B Register：暂存寄存器。
  - 暂存寄存器。在做乘、除法时放乘数或除数及结果。
- PSW（Program Status Word）
  - PSW是8位寄存器，存储程序运行状态及标志位。

PSW

PSW位地址	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H
	CY	AC	F0	RS1	RS0	OV	未定义	P

- CY (CarrY)：进位标志。
  - 加减运算时，保存最高位进位、借位状态。
- AC (Auxiliary Carry)：辅助进位（半进位）标志。
  - 低半字节向高半字节有进位或借位，AC置1，否则置0。

AC/CY在DA指令中的使用

❑二-十进制调整指令

DA　　A

这条指令对累加器A参与的BCD码加法运算所获得的 8 位结果进行十进制调整，使累加器A中的内容调整为压缩型BCD码(常用的是BCD8421码)的数。

8421 BCD码

- BCD (Binary-Coded Decimal)
  用二进制编码的十进制代码
- 用于简化对使用十进制数的设备（如计时器、电梯楼层显示）的处理。处理器不是把十进制数转换为二进制数以进行数学操作，然后再转换回十进制；而是可以按照BCD码保存数并且执行数学操作。
- BCD码有很多种，单片机常用的是8421 BCD码。
- 压缩型BCD码 / 非压缩BCD码 分别用 半个/1个字节表示一个十进制数字。例：56表示为0101 0110

AC/CY在DA指令中的使用

❑二-十进制调整指令

DA　　A

这条指令对累加器A参与的BCD码加法运算所获得的 8 位结果进行十进制调整, 使累加器A中的内容调整为压缩型BCD码(常用的是BCD8421码)的数。

执行该指令时, 判断 A中的低 4 位是否大于 9 或辅助进位标志 AC是否为 “1”, 若两者有一个条件满足, 则低 4 位加 6 操作; 同样, A中的高 4 位大于 9 或进位标志 Cy为 “1”两者有一个条件满足时, 高 4 位加 6 操作。

PSW

PSW位地址	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H
	CY	AC	F0	RS1	RS0	OV	未定义	P

- CY：进位标志。

  加减运算时，保存最高位进位、借位状态。

- AC：辅助进位（半进位）标志。低半字节向高半字节有进位或借位，AC置1，否则置0。

- 算一算：78H+97H，如何设置PSW的CY和AC?

```
  0111 1000
+ 1001 0111
1 0000 1111
```

有进位 CY=1

没有半进位 AC=0

PSW

PSW位地址	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H
	CY	AC	F0	RS1	RS0	OV	未定义	P

- RS1、RS0：工作寄存器组选择位：

0	0	选择工作寄存器0组
0	1	选择工作寄存器1组
1	0	选择工作寄存器2组
1	1	选择工作寄存器3组

- P：奇偶校验位，它用来表示累加器ACC中二进制数位“1”的个数的奇偶性。若为奇数，则P=1，否则为0。

  例：某运算结果是78H，P=0。

PSW

PSW位地址	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H
	CY	AC	F0	RS1	RS0	OV	未定义	P

F0：用户标志位。程序员自行使用。

OV：溢出标志位。有符号数运算时，若发生溢出，
OV置“1”，否则清“0”。

问题：何为溢出？8位二进制表示有符号数的范围？

控制器

- 程序计数器（PC）
  - 16位专用寄存器，下一条要执行的指令的16位地址
  - CPU将PC的内容送往地址总线，从存储单位取出指令
  - 具有自动加1功能，默认指向顺序执行的下一条指令
  - 如不按顺序执行，则须将转向的程序入口地址送往PC
  - 系统复位后，PC=0000H

控制器

- 堆栈寄存器（SP）
  - 8位寄存器，存放栈顶地址，后进先出（LIFO）
  - MCS-51的堆栈地址区间：07H起，这是SP初始化值
  - 建议用户编程时，SP改为1FH或更大
    - 片内RAM的08H~1FH单元常被用过工作寄存器组1~3
  - SP的常见用途
    - 暂存数据
    - 子程序调用或中断操作时，用于存放返回地址
- DPTR (Data Pointer)（由DPH和DPL组成）：
  - 除PC外的唯一16位寄存器，用来访问外部数据和程序存储器（变址寻址方式中的基址寄存器）
  - 可作通用寄存器使用，可分高低两个8位寄存器使用

变址存哪里？

控制器

1. 指令寄存器（用户不可见）

   ❑从程序存储器取出的指令，首先送到指令寄存器存放

2. 指令译码器（用户不可见）

   ❑指令随后进入指令译码器，转变成执行该指令所需要的电信号

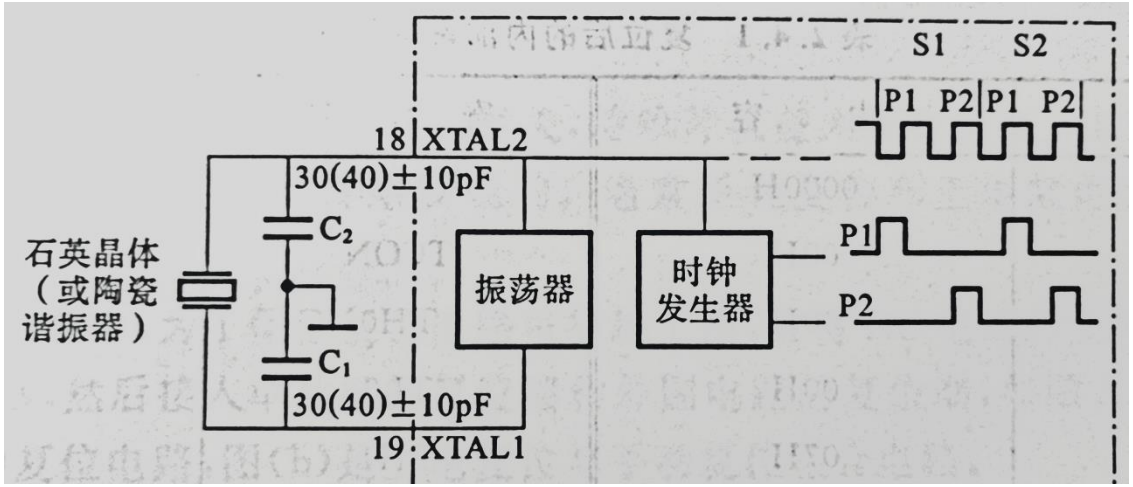
3. CPU定时控制

   ❑根据译码的输出，发出特定的时序信号，使计算机正确执行该指令所要求的各种操作

8051的I/O端口和外部总线

- 四个8位并行双向I/O口：P0、P1、P2、P3
- 一个串行口：TXD/RXD
- 地址总线AB
  - 宽度16位，所以可以直接寻址的外部存储器空间是?
  - 低8位地址A0~A7由P0口提供
  - 高8位地址A8~A15由P2口提供
- 数据总线DB
  - 宽度为8位，由P0口提供D0~D7

8051的内部时钟电路

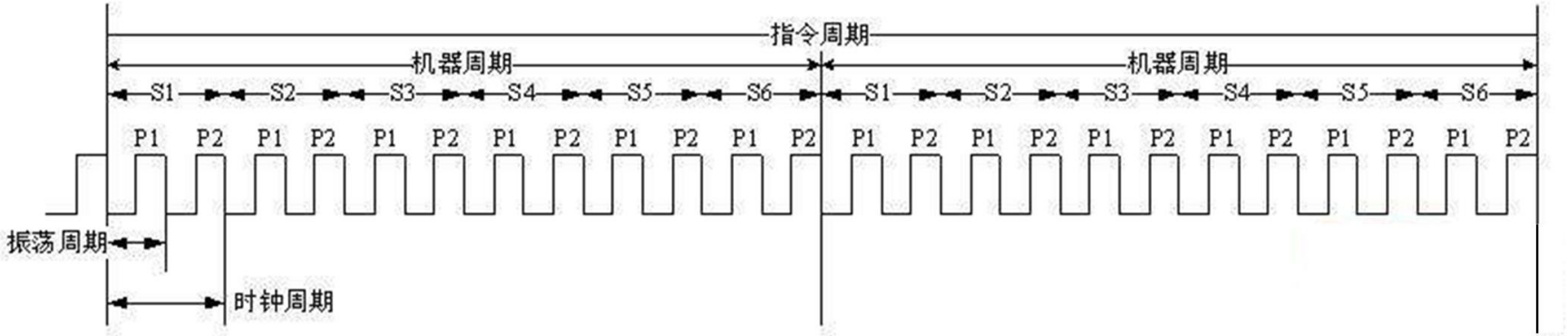


- 振荡频率由石英晶振的谐振频率决定，一般为1.2~12MHz

存在时钟同步的问题

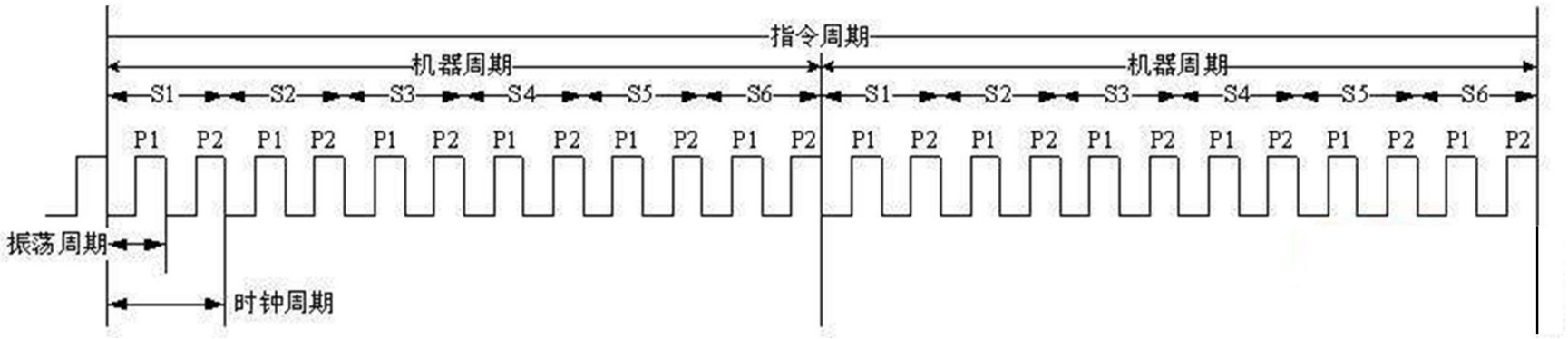
- 时钟电路是一个二分频触发电路,它将振荡器的信号进行二分频，向芯片内提供一个2节拍的信号
  - 在每个时钟周期的前半周期，节拍信号P1有效；
  - 在每个时钟周期的后半周期，节拍信号P2有效；

时序



- 振荡周期：即提供定时信号的振荡源的周期
- 时钟周期：又称状态周期
  - 前半周期(P1)通常做算术逻辑运算操作
  - 后半周期(P2)通常做寄存器传送操作
- 机器周期：一个基本操作执行需要6个时钟周期
- 指令周期（接下页）

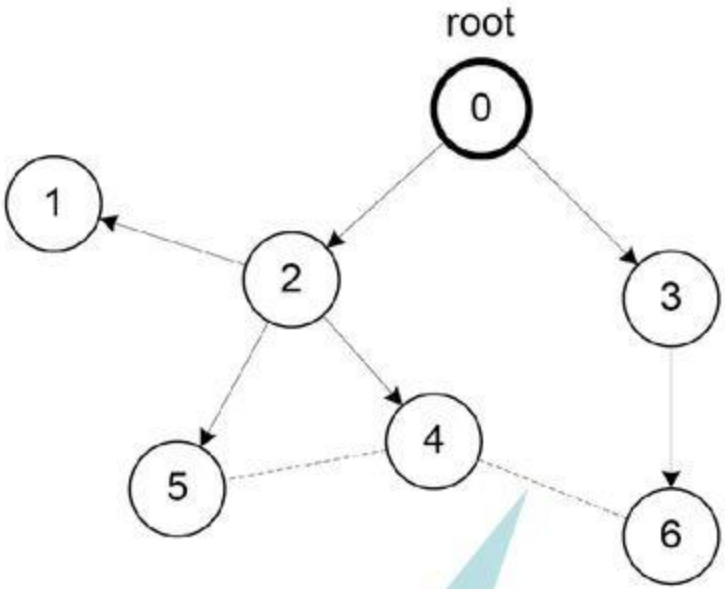
时序



- 指令周期：执行一条指令需要的时间
  - 一个指令周期由若干个机器周期组成
  - 不同指令的机器周期数不同
  - MCS-51的指令长度为1~3字节，单字节和双字节指令都可能是单周期或双周期的，三字节指令都是双周期的
  - 乘法和除法指令占用四个周期

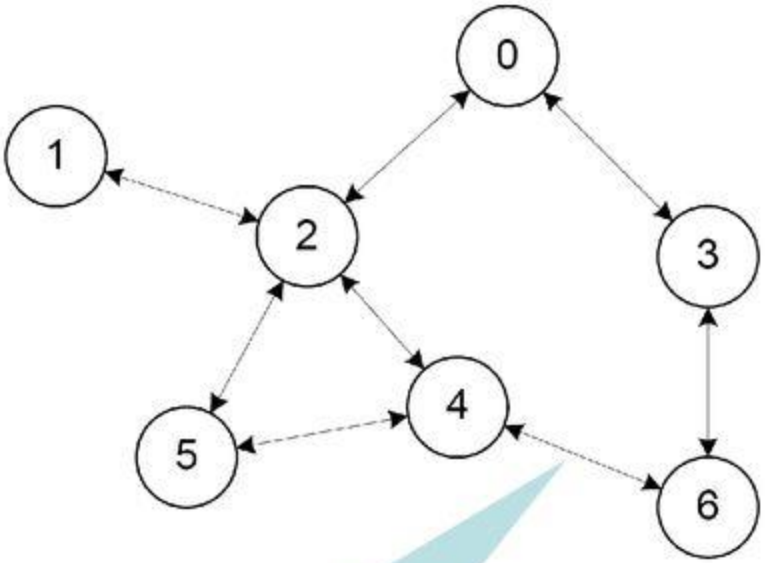
Variants of Clock Synchronization Algorithms

Tree-like Algorithms
e.g. FTSP



Bad local skew

Distributed Algorithms
e.g. GTSP



All nodes consistently average errors to *all* neigbhors

8051的存储器组织

8051与x86的存储器组织方式不同：x86只有一个逻辑地址空间，可在其中随意安排ROM或RAM。一个地址对应唯一的ROM或RAM单元，并用同类访问指令。

而MCS-51则是哈佛结构：
存储程序与数据的地址空间是分离的：

物理结构上，8051有四个存储空间：
片内程序存储器、片外程序存储器、
片内数据存储器、片外数据存储器。

逻辑上，用户角度看8051有三个存储空间

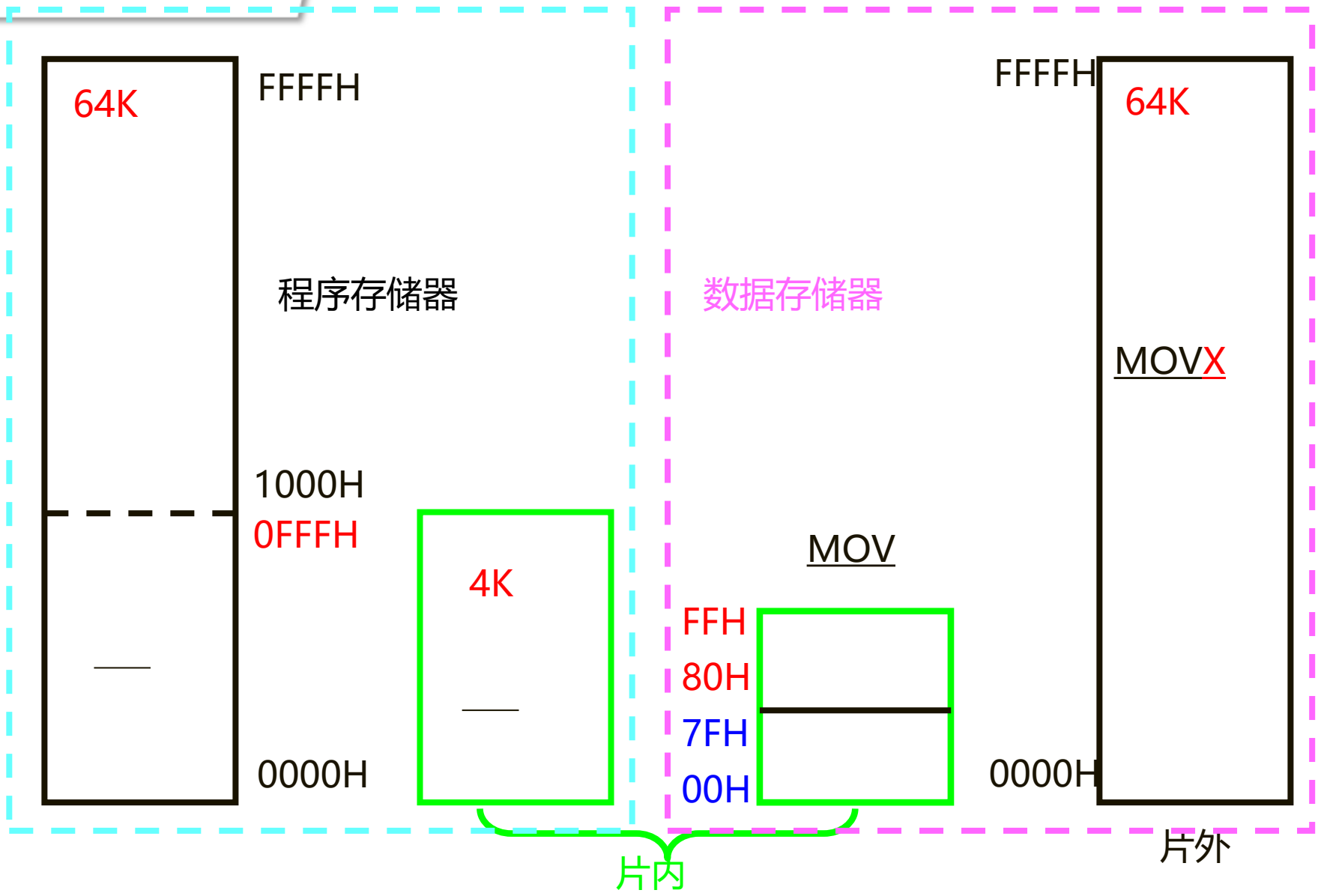
采用不同指令访问这几个不同的地址空间：

片内外程序存储器空间----MOVC

片内数据存储器空间和SFR----MOV

片外数据存储器地址空间----MOVX

三种访存指令


64K
FFFFH
程序存储器
1000H
0FFFH
0000H
4K
数据存储器
MOV
FFH
80H
7FH
00H
FFFFH
64K
MOVX
0000H
片内
片外

程序存储器及地址空间

- 用于存放程序和表格常数。
- 8051片内有4K字节ROM，片外16位地址线最多可扩展64K字节ROM，两者统一编址。通过EA（外部访问允许）引脚来选择
  - EA=1，程序地址<4KB，即在0000H～0FFFH区间时，从片内程序存储器中取指。程序地址>4KB，即在1000H～FFFFH区间时，从片外存储器取指。
  - EA=0，所有指令均从片外程序存储器中读取，这时片外存储器从0000H开始编址。

中断向量表

❑程序存储器中的6个特殊地址——中断向量表

0000H：8051复位后，PC = 0000H, SP=07H, …

0003H：外部中断0入口。

000BH：定时器T0溢出中断入口。

0013H：外部中断1入口。

001BH：定时器T1溢出中断入口。

0023H：串行口中断入口。

通常在这些入口地址处存放一条绝对跳转指令。为什么?

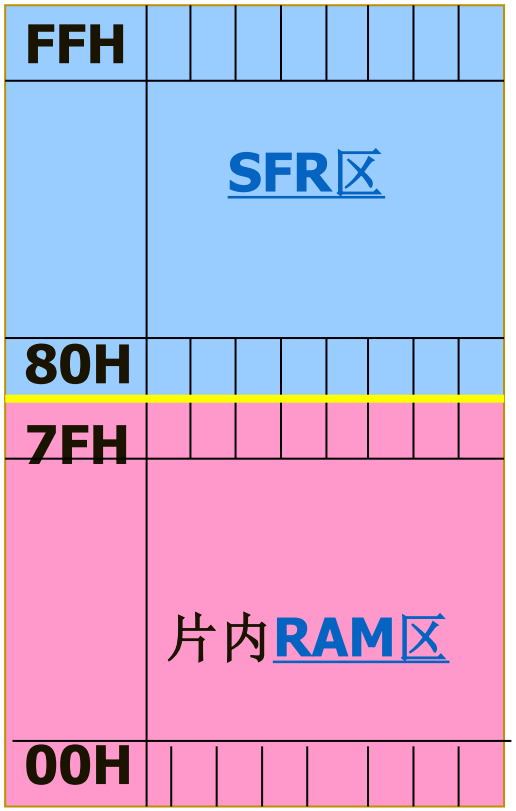
算一算这些地址间的间隔。

数据存储器及地址空间

数据存储器

- 片外RAM64KB，地址范围0000H~FFFFH
  用**MOVX**指令访问。
- 片内RAM128B，地址范围00H~7FH
  用**MOV**指令访问。

8051片内RAM的配置图

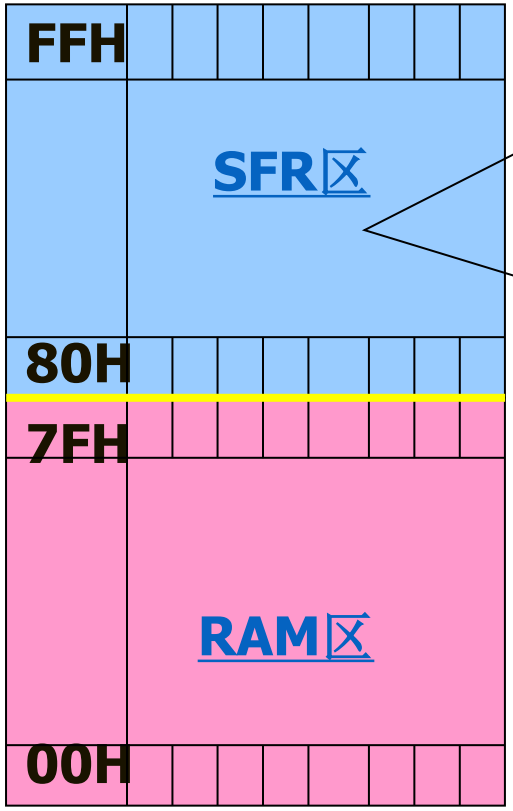


片内256B RAM分为两大部分:

高 128 字节（80H~FFH）：其中21个用于特殊功能寄存器区SFR。

低 128 字节（00H~7FH）
真正的RAM区.

特殊功能寄存器SFR(Special Functional Register)



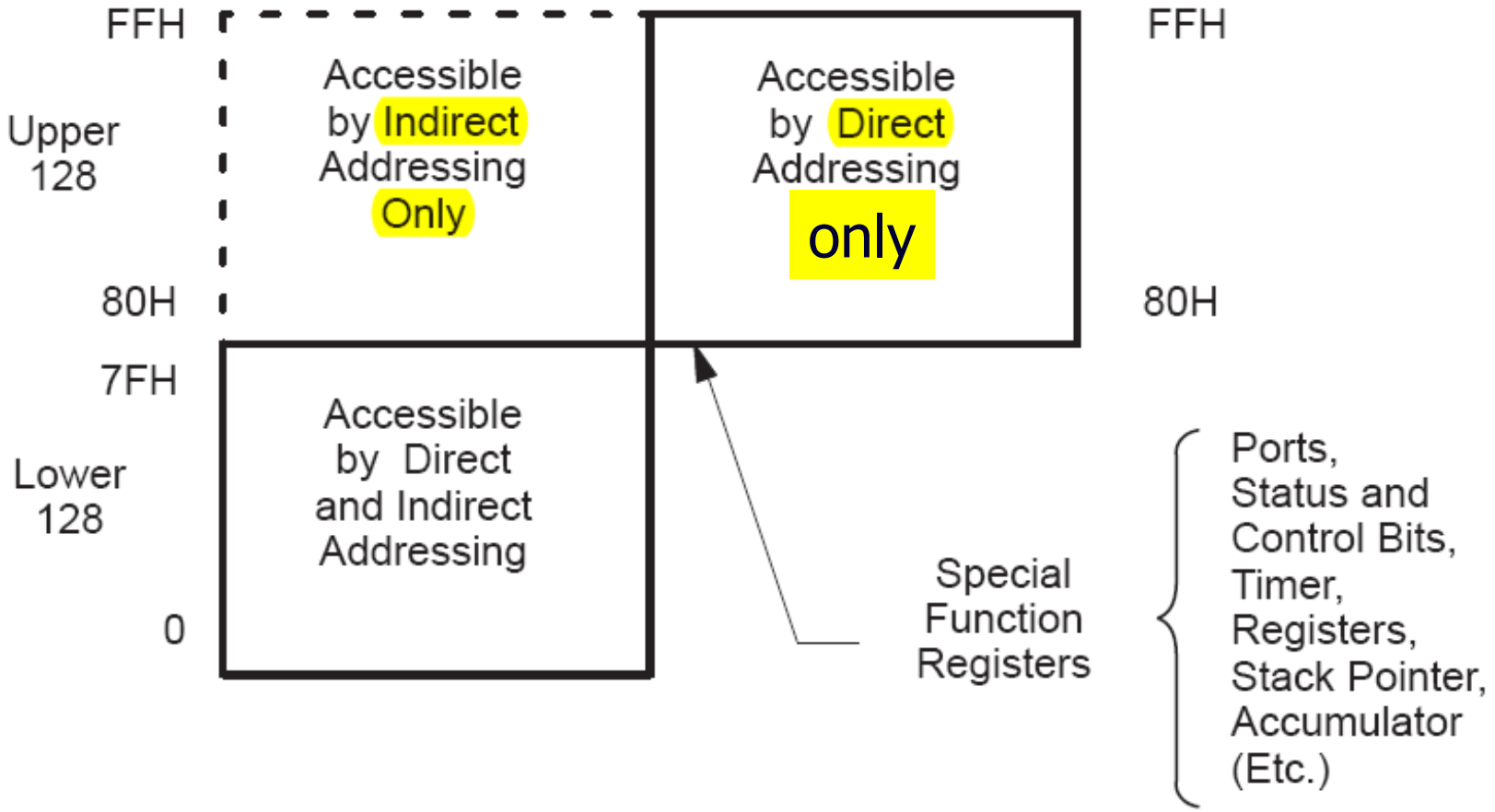
高128B的RAM单元中有21个单元可用，称为 SFR:
A，B，PSW，SP，DPH，DPL，P0，P1，P2，P3，IP，IE，TCON，TMOD，TH0，TL0，TH1，TL1，SCON，SBUF，PCON。

每个SFR寄存器均有自己唯一的8位物理地址，但是每个SFR只能采用直接寻址方式，不能采用寄存器间接寻址

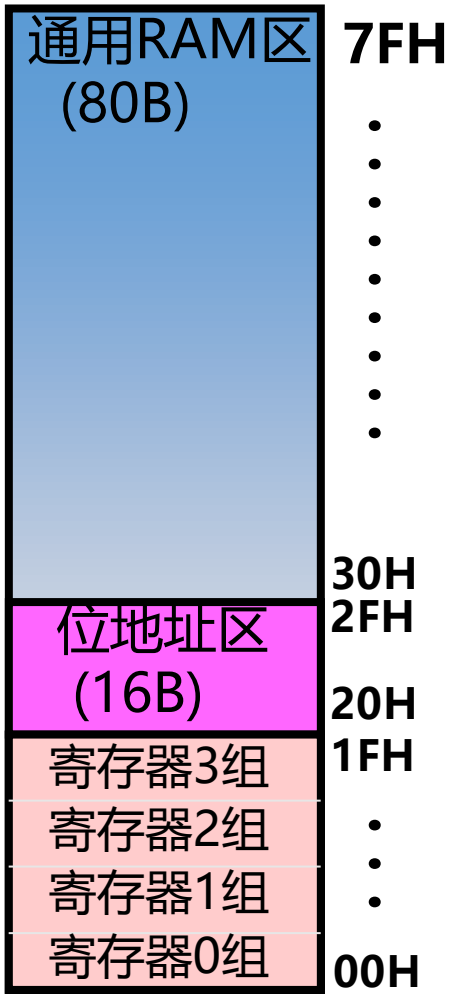
256字节片内SRAM的实现

当SRAM地址 >7FH 时，

直接地址与间接地址访问不同的物理SRAM。



RAM区 低128字节（00H~7FH）



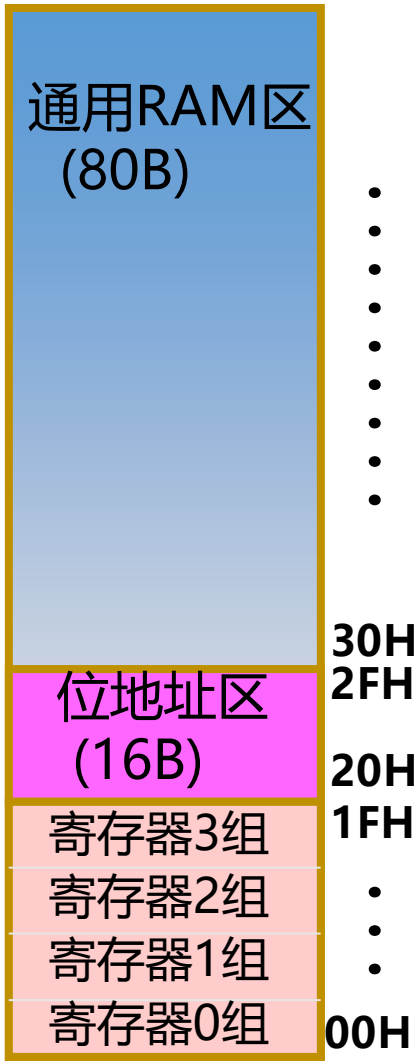
由PSW中的2位RS1、RS0决定选哪一组为当前工作寄存器：

RS1、RS0=00　选0组

RS1、RS0=01　选1组

RS1、RS0=10　选2组

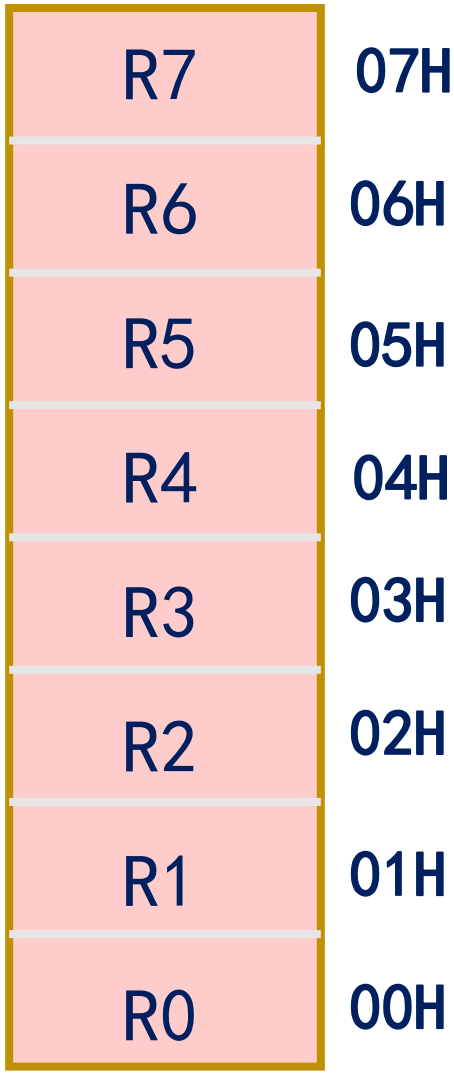
RS1、RS0=11　选3组

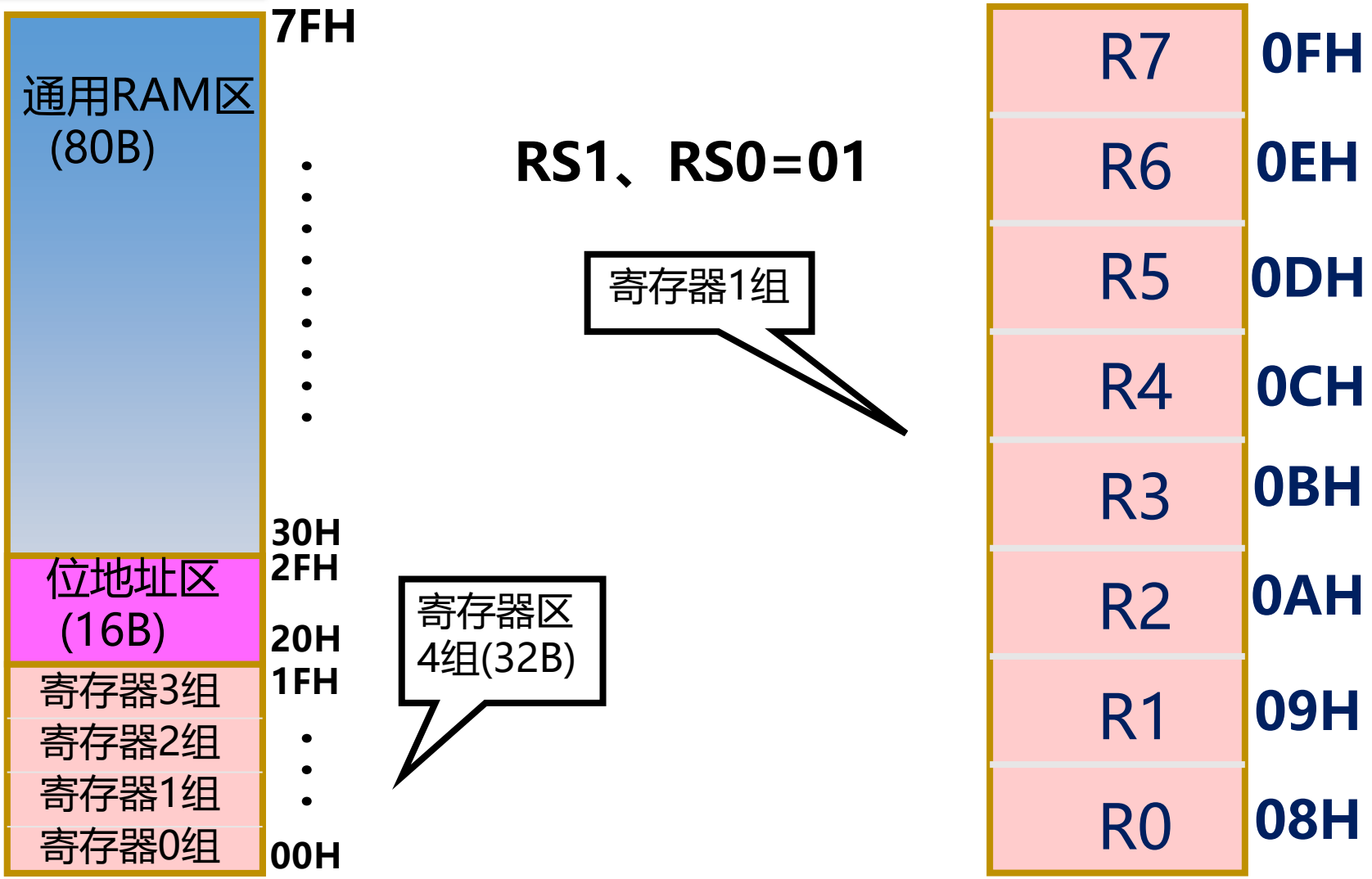


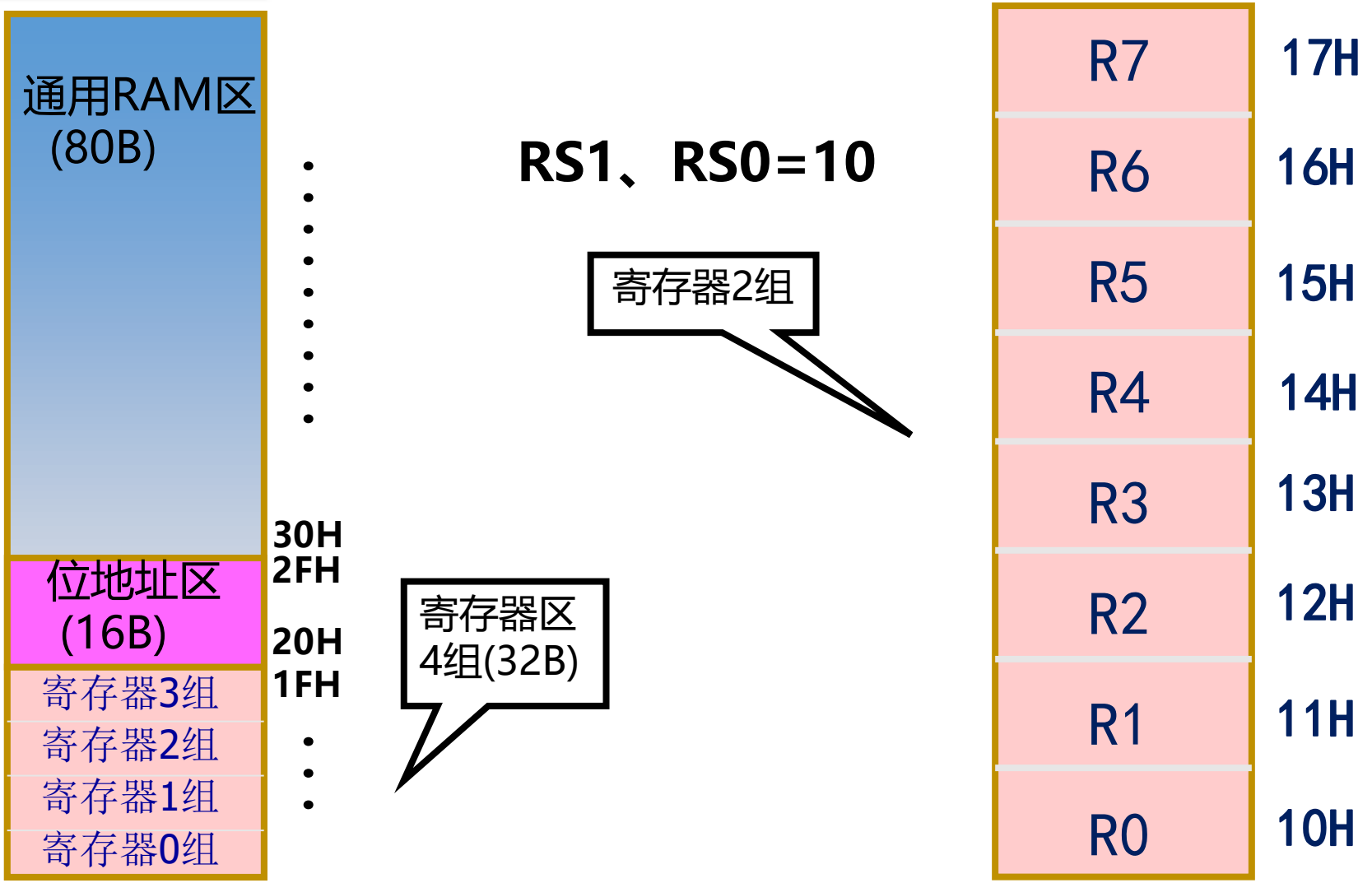
寄存器区
4组(32B)

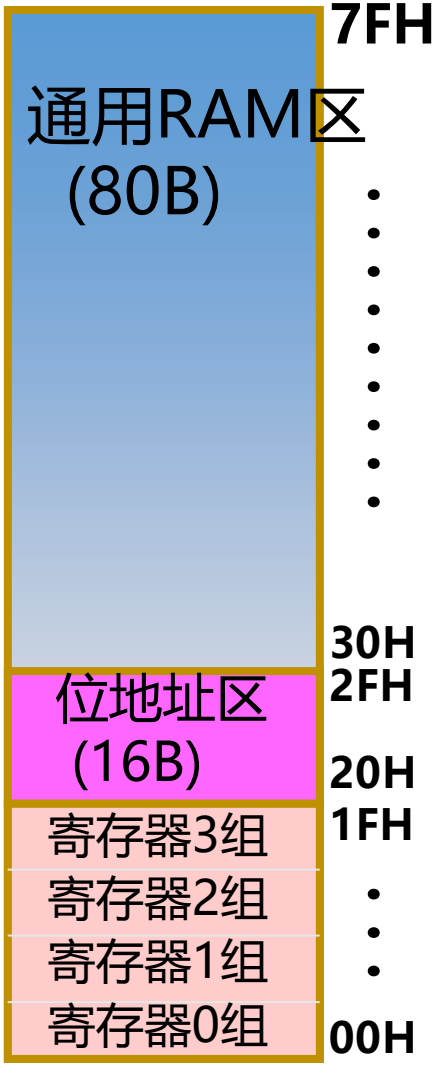
RS1、RS0=00

寄存器0组




7FH
通用RAM区
(80B)
30H
2FH
位地址区
(16B)
20H
1FH
寄存器3组
寄存器2组
寄存器1组
寄存器0组
00H
寄存器区
4组(32B)
RS1、RS0=01
寄存器1组
R7
R6
R5
R4
R3
R2
R1
R0
0FH
0EH
0DH
0CH
0BH
0AH
09H
08H


通用RAM区
(80B)
位地址区
(16B)
寄存器3组
寄存器2组
寄存器1组
寄存器0组
30H
2FH
20H
1FH
寄存器区
4组(32B)
RS1、RS0=10
寄存器2组
R7
R6
R5
R4
R3
R2
R1
R0
17H
16H
15H
14H
13H
12H
11H
10H



寄存器区
4组(32B)

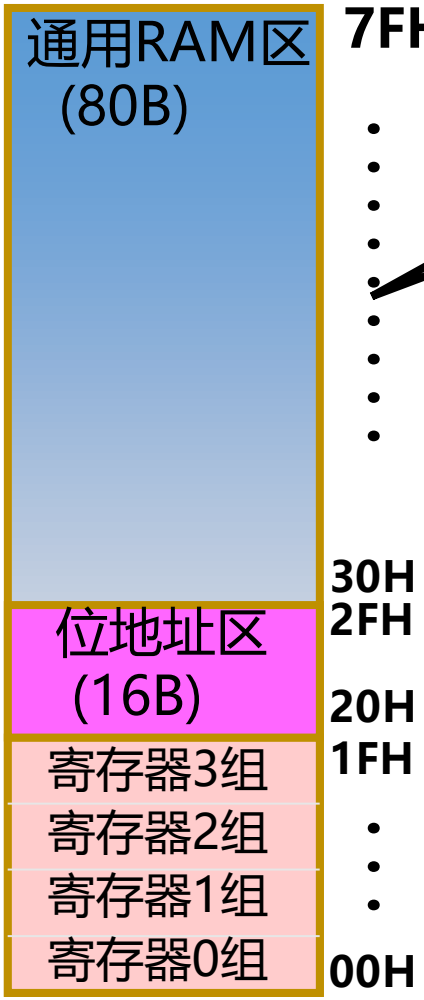
RS1、RS0=11

寄存器3组

R7	1FH
R6	1EH
R5	1DH
R4	1CH
R3	1BH
R2	1AH
R1	19H
R0	18H

思考，用PSW中的RS1和RS0控制选择工作寄存器组，好处是什么？

RAM区 低128字节（00H~7FH）



从30H到7FH，共80个字节，常作为一般的数据缓冲区，可设置为堆栈区

例子

内部RAM间可以直接传输数据

```
MOV A，0F0H      ;直接寻址，将片内物理地址为
                  0F0H的字节内容读入A
MOV P1，P0       ;P0地址为80H，P1地址为90H
MOV  A，@R0      ;间接寻址，操作数在片内RAM
MOVX A，@R0      ;间接寻址，操作数在片外RAM
MOVX A，@DPTR    ;间接寻址，操作数在片外RAM
MOVC A，@A+DPTR  程序存储器中 [A+DPTR] → A
```

只有累加器A可以与外部存储器间直接传输数据

位操作是MCS-51的重要特色

MCS-51的部份RAM（20H-2FH这16字节）和

11个SFR【A累加器，B寄存器、PSW、

IP（中断优先级控制寄存器）、

IE（中断允许控制寄存器）、

SCON（串行口控制寄存器）、

TCON（定时器/计数器控制寄存器）、

P0-P3】具有位寻址功能。

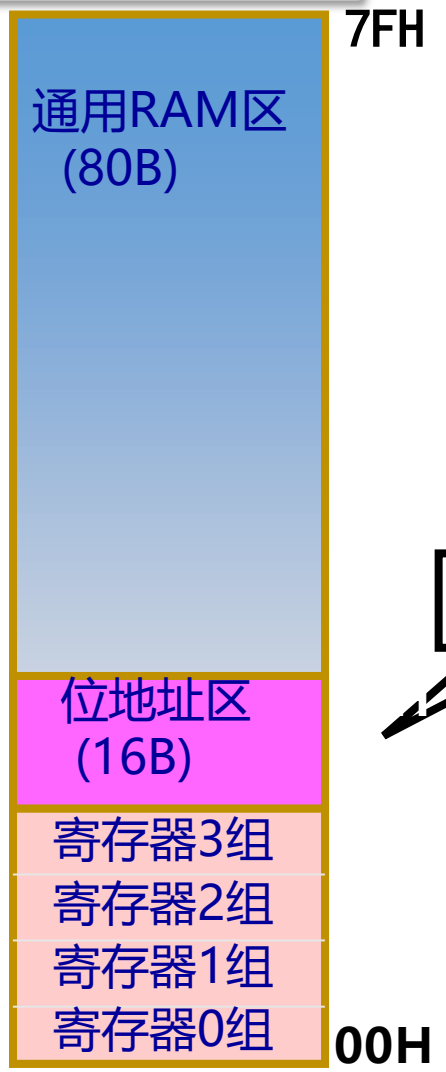
指令集中包括位变量传送、逻辑运算、控制程序转移等指令。

位操作是MCS-51的重要特色

位地址表达方式：以PSW中位4（RS1）为例，下面三种方式是等价的。

- 直接(位)地址方式：如 D4H；
- 点操作符号方式：如 PSW.4，D0H.4；
- 位名称方式：如 RS1；

```
例    MOV C，P1.3        ；C := P1.3
      CLR P1.0           ；P1.0 := 0
      JNB P3.2，L2       ；若 P3.2=0 则转到L2
      SETB P2.5          ；P2.5 := 1
L2:   ANL C，P1.1        ；C := C and P1.1
```



位地址区

D7	D6	D5	D4	D3	D2	D1	D0	
7FH		. .	. . . .	. . . .	. .		78H	2FH
77H							70H	2EH
6FH							68H	2DH
67H							60H	2CH
5FH							58H	2BH
57H							50H	2AH
4FH							48H	29H
47H							40H	28H
3FH		1					38H	27H
37H							30H	26H
2FH							28H	25H
27H							20H	24H
1FH							1FH	23H
17H							10H	22H
0FH		. .	. . . .	. . . .	. .		08H	21H
07H	06H	05H	04H	03H	02H	01H	00H	20H

指令“SETB 3DH”将27H.5位置1

寄存器	功能名称	位地址/位定义								地址
B	B寄存器	F7	F6	F5	F4	F3	F2	F1	F0	F0H
		B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	
ACC	累加器	E7	E6	E5	E4	E3	E2	E1	E0	E0H
		ACC.7							ACC.0	
PSW	程序状态寄存器	D7	D6	D5	D4	D3	D2	D1	D0	D0H
		CY	AC	F0	RS1	RS0	OV		P	
IP	中断优先级控制寄存器				BC	BB	BA	B9	B8	B8H
					PS	PT1	PX1	PT0	PX0	
P3	P3口数据寄存器	B7	B6	B5	B4	B3	B2	B1	B0	B0H
		P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	
IE	中断运行控制寄存器				AC	AB	AA	A9	A8	A8H
					ES	ET1	EX1	ET0	EX0	
P2	P2口数据寄存器	A7	A6	A5	A4	A3	A2	A1	A0	A0H
		P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	
SBUF	串口发送/接收缓冲器									99H
SCON	程序状态寄存器	9F	9E	9D	9C	9B	9A	99	98	98H
		SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
P1	P1口数据寄存器	97	96	95	94	93	92	91	90	90H
		P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
TH1	T1计数器高8位									8DH
TH0	T0计数器高8位									8CH
TL1	T1计数器低8位									8BH
TL0	T0计数器低8位									8AH
TMOD	定时器方式控制寄存器	GATE	C/T	M1	M0	GATE	C/T	M1	M0	89H
TCON	定时器控制寄存器	8F	8E	8D	8C	8B	8A	89	88	88H
		TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
PCON	电源控制寄存器	SMOD								87H
DPH	数据指针高8位									83H
DPL	数据指针低8位									82H
SP	堆栈指针寄存器									81H
P0	P0口数据寄存器	87	86	85	84	83	82	81	80	80H
		P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	

MCS-51的寻址方式

- 寻找指令操作数或操作数地址即是寻址
- 寻找操作数或操作数地址的方式就是寻址方式

一、立即寻址

二、寄存器寻址

三、直接寻址

四、寄存器间接寻址

五、基址 + 变址寻址

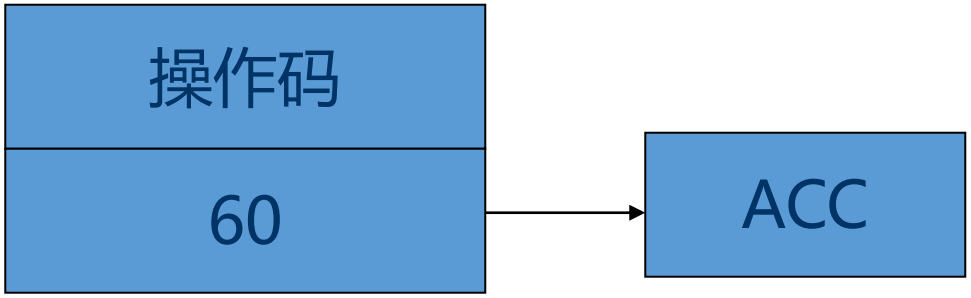
六、相对寻址

七、位寻址

一、立即寻址

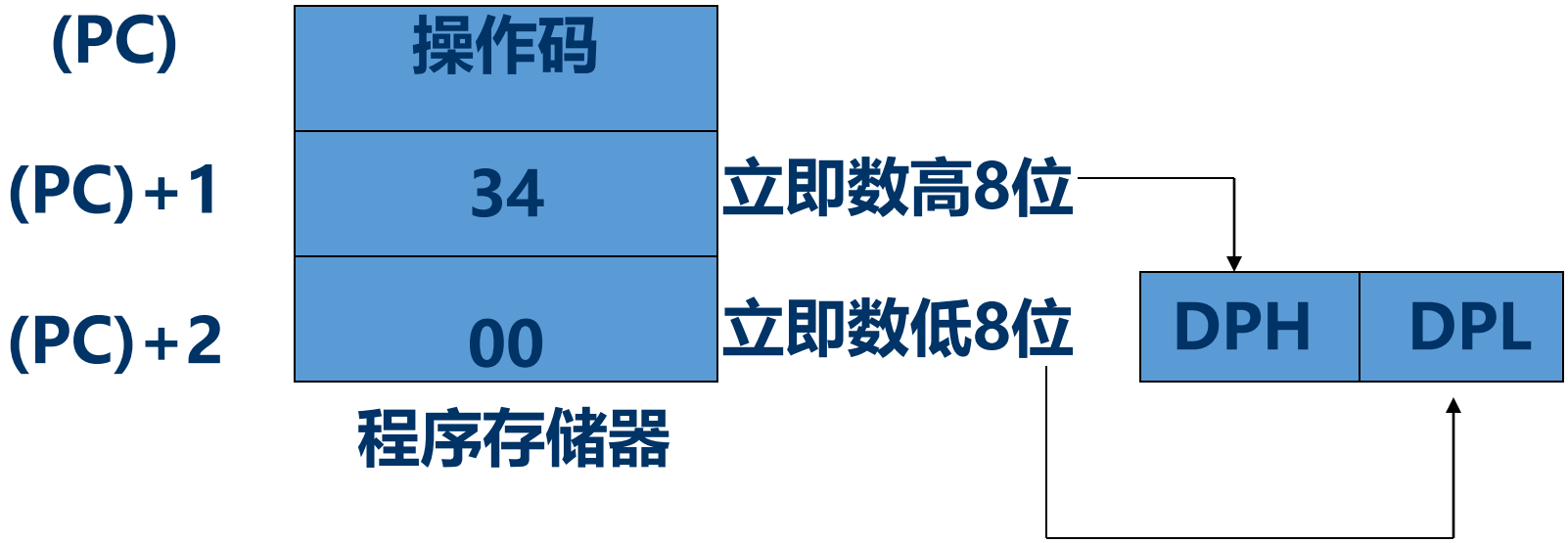
❑操作数放在指令中，紧跟在操作码后面， 用“#”号表示 。

例1：MOV　　A，# 60H ； A←#60H



例2：MOV　DPTR，# 3400H

功能：　DPTR←#3400H



结果（DPTR）＝3400H

二、寄存器寻址

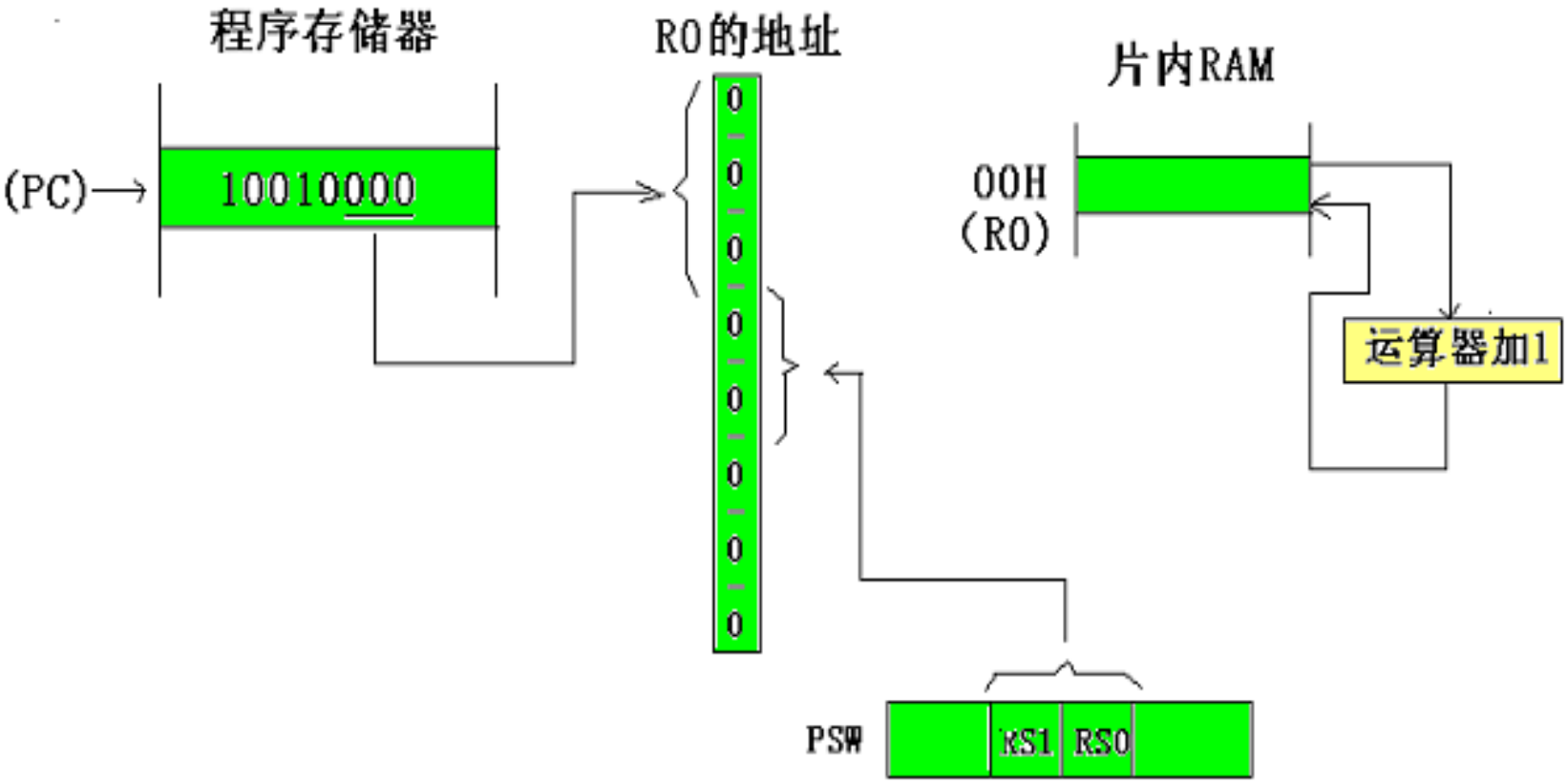
- 指出某一寄存器的内容作为操作数
- 寄存器指A、B 、DPTR、CY以及R0～R7
- R0～R7用操作码的低三位000～111选定
- A、B 、DPTR、CY隐含在操作码中

例1、MOV A，R0 ； (R0) →A

例2、ADD A，R1 ； (A) + (R1) →A

例3、INC R0 ； (R0) +1→ (R0)

例3、INC R0 ； (R0) +1→ (R0)

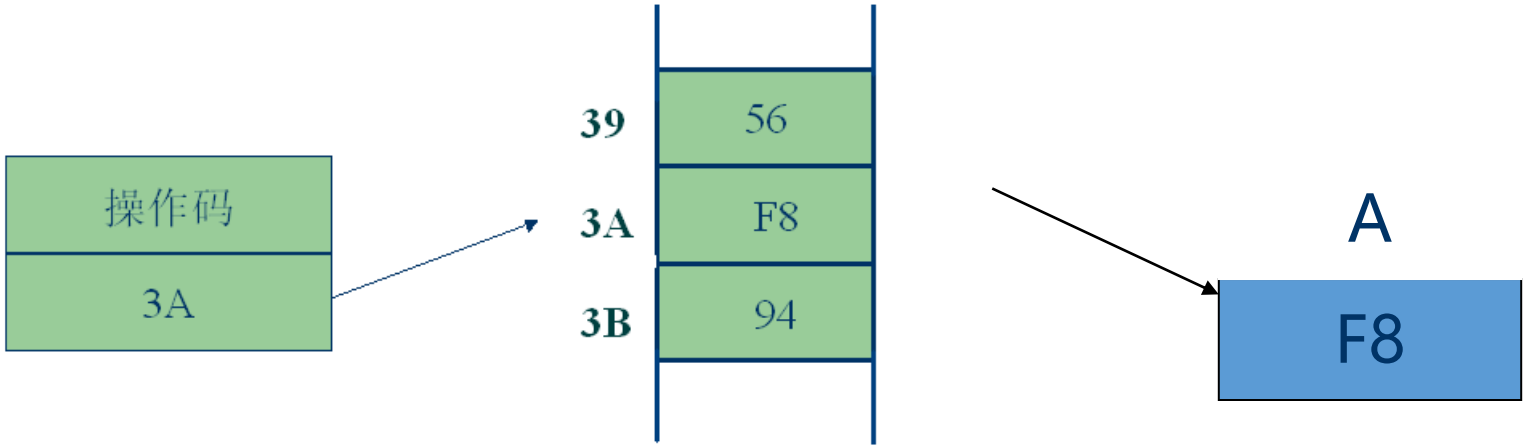


三、直接寻址

- 指令直接给出操作数的有效地址，不用“#”号表示。
- 能直接寻址的存储空间（只在内部）：
  1、RAM内部数据，地址00H～7FH
  2、SFR寄存器，地址80H～FFH
- 至少是双字节指令

例1：MOV　A，3AH　；　(3AH) →A

❑表示把片内RAM中3AH单元的内容送累加器A。

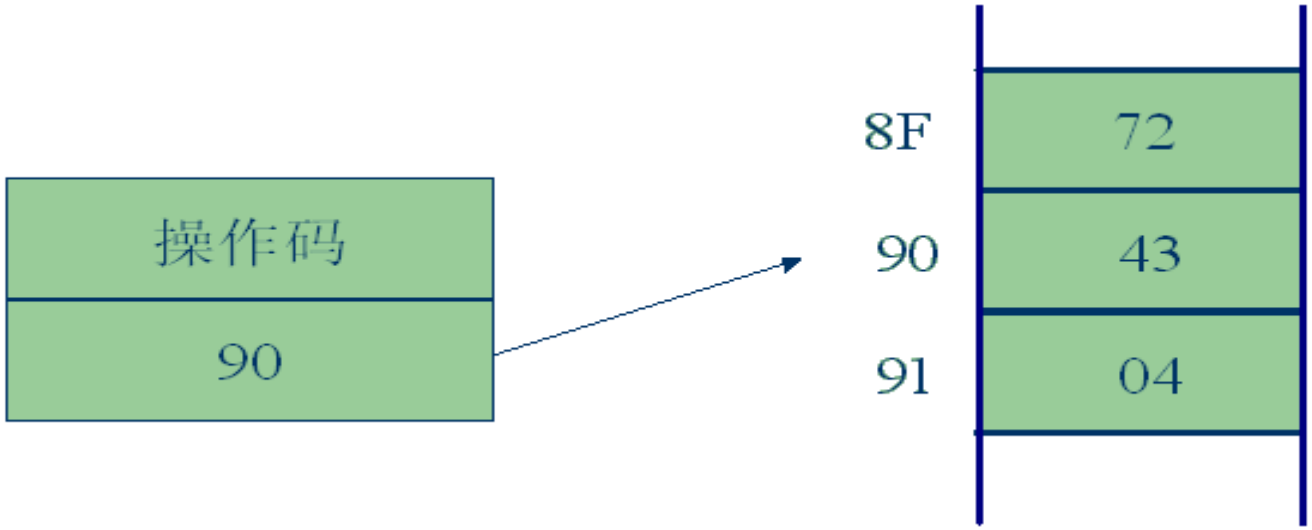


结果（A）＝F8H

例2：MOV A，90H (MOV A，P1)

90H可用寄存器符号P1表示

功能：表示把P1口的内容送到A； （P1）→A



结果（A）＝43H

四、寄存器间接寻址

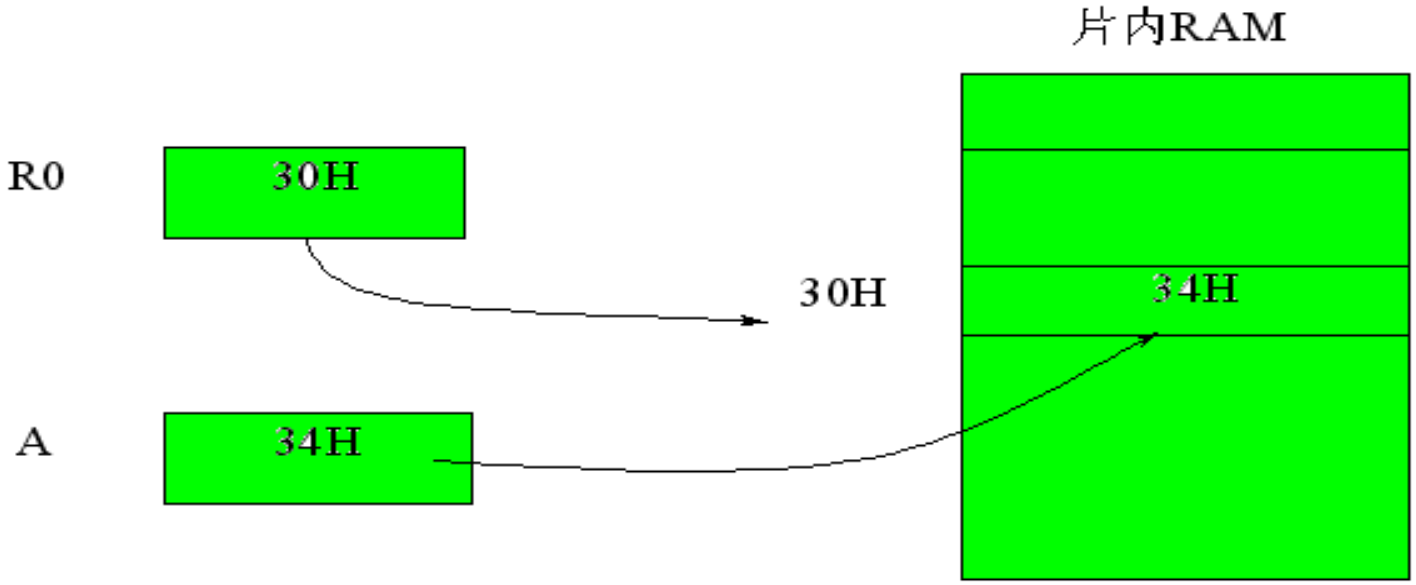
- 指令指出某一寄存器的内容作为操作数的地址，用符号@表示。
- 操作数的地址要事先放在指定的寄存器中。

  如：R0、R1、DPTR、SP为间址寻址寄存器
- 能进行寄存器间接寻址的存储空间有：

1. @R0、@R1寻址片内RAM的128字节单元
2. @DPTR、@R0、@R1寻址片外64kB存储器空间
3. SP寻址堆栈区的单元

例1：MOV @R0， A

R0 的内容为30H，A的内容为34H

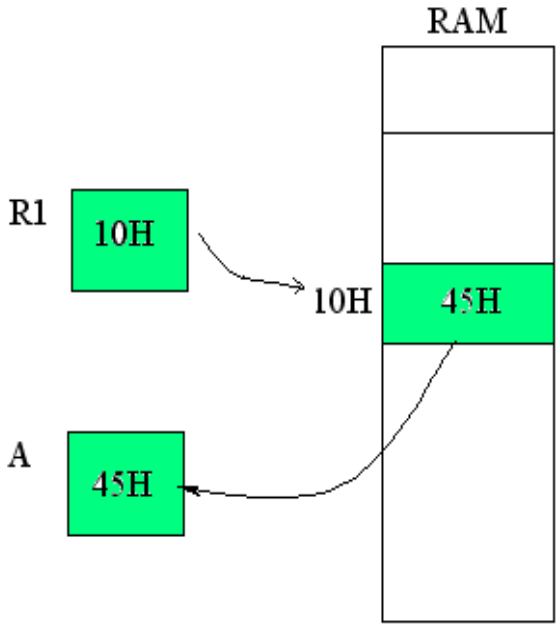


例2：MOV A，@R1

❑已知：

（R1）＝10H，

（10H）＝45H



结果（A）＝45H

MOVX @DPTR，A

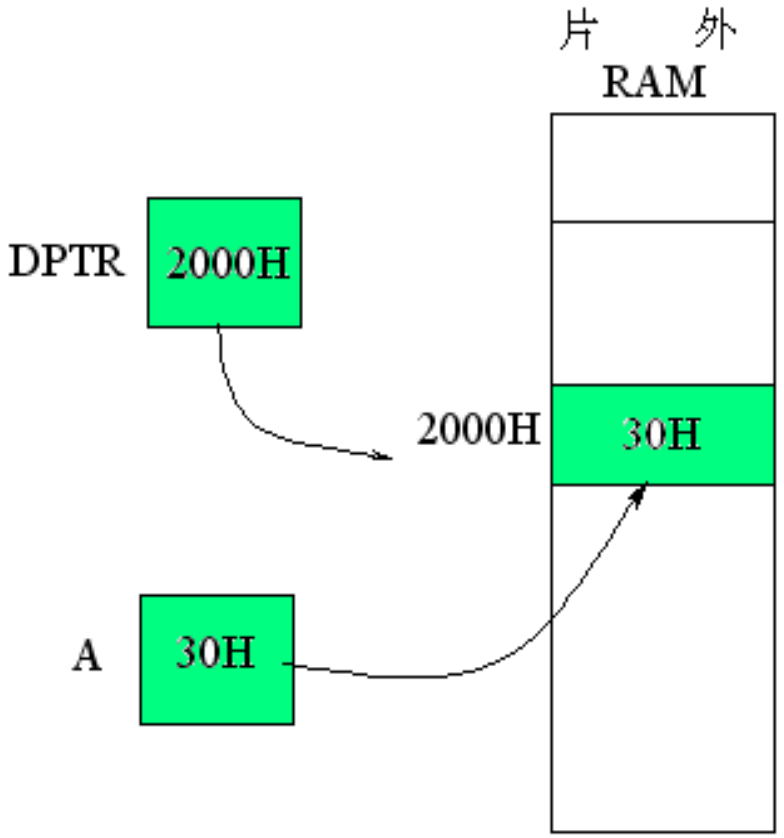
- 已知：

  (A) =30H，

  (DPTR) =2000H

结果：

(2000H) =30H



五、基址+变址寻址

- 是指基址寄存器内容加上变址寄存器内容之和作为操作数的有效地址。两者之和为16位的地址

  基址寄存器：PC或DPTR

  变址寄存器：？

- 能进行基址+变址寻址的存储空间只有：程序存储器中内容

五、基址+变址寻址

- 是指基址寄存器内容加上变址寄存器内容之和作为操作数的有效地址。两者之和为16位的地址

  基址寄存器：PC或DPTR

  变址寄存器：累加器A （A为无符号数）

- 能进行基址+变址寻址的存储空间只有：程序存储器中内容

- 由于程序存储器是只读的，因此基址＋变址寻址只有读操作而无写操作。

例如指令：

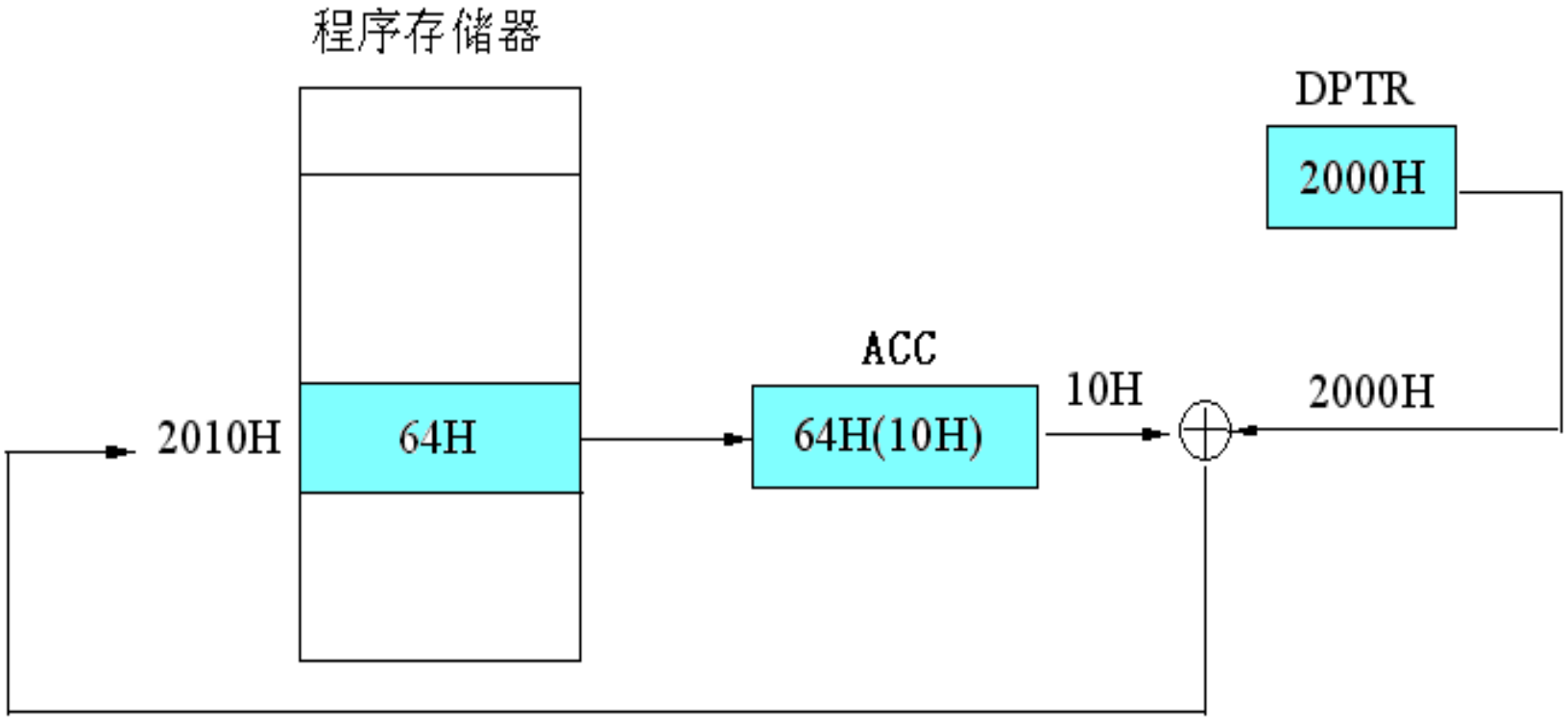
1、MOVC A，@A＋PC ；

（（A）＋（PC））→A

2、MOVC A，@A＋DPTR；

（（A）＋（DPTR））→A

指令MOVC A，@A＋DPTR；执行示意图



结果:（ACC）＝64H

六、相对寻址

❑以当前PC的内容为基准，加上指令给出的偏移量（rel）形成新的PC值(转移地址）的寻址方式。

转移地址 = 目的地址

= 当前（PC） + rel

目的地址 = PC当前值十rel

目的地址 = 转移指令的PC值 + 2（或3）十rel

目的地址 = 转移指令地址 + 转移指令字节数 + rel

- 仅出现在转移指令中，用于修改PC值，主要用于实现程序的分支转移。
- 相对偏移量（rel）放在操作码的后面，即指令第二个字节给出的数据 。
- rel是一带符号的8位二进制数，以补码形式出现。
- 程序转移范围：以PC的当前值为起始地址，相对转移在+127~ -128之间 。

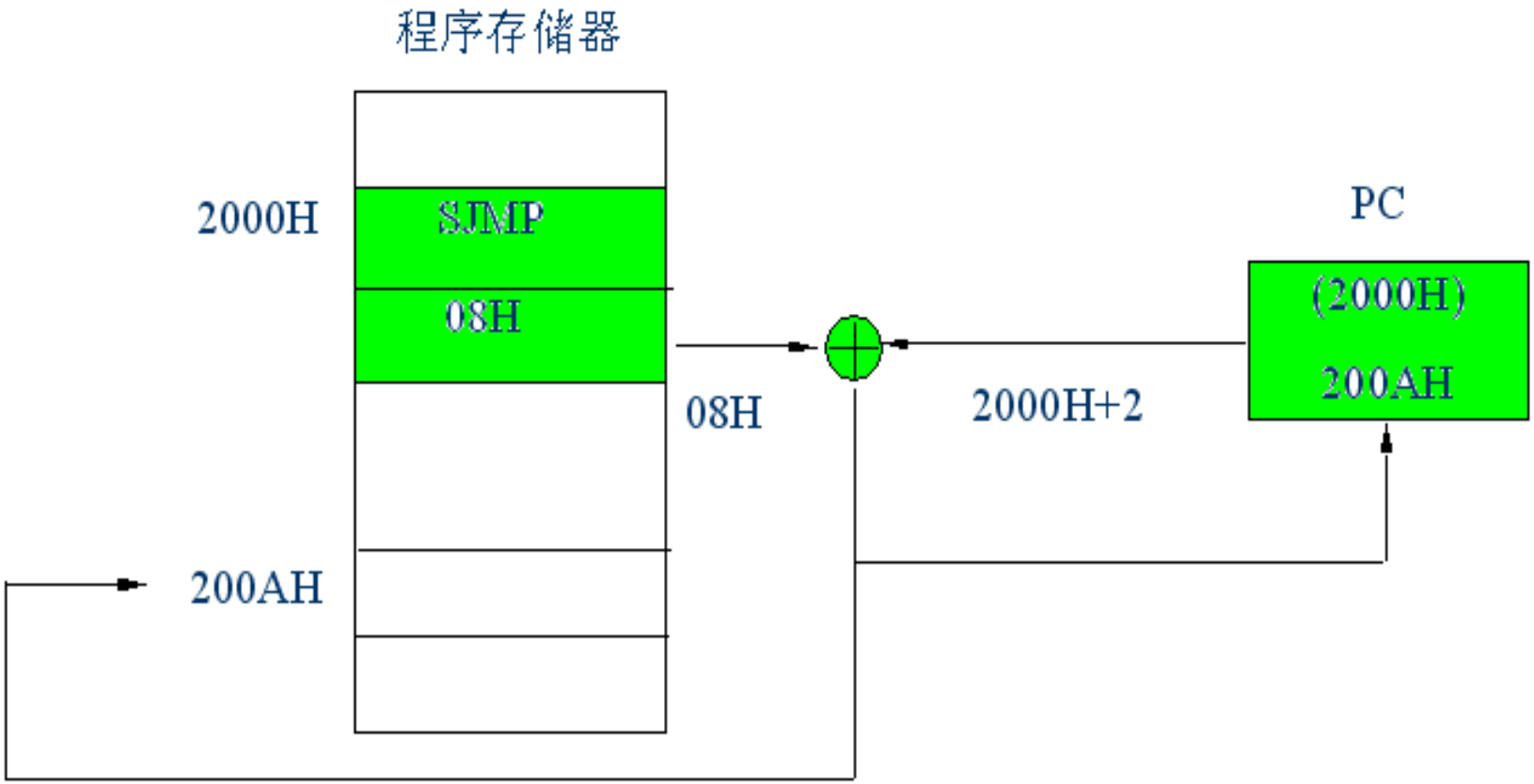
例如： SJMP rel

SJMP 08H

- 功能：PC←PC+2+08H
- 已知：(PC) = 2000H

  (rel) = 08H
- 转移地址：（PC） = 200AH

SJMP 08H 相对寻址示意图



七、位寻址

- 位寻址只能对有位地址的单元作位寻址操作。
- 位寻址其实是一种直接寻址方式，不过其地址是位地址。

例如： MOV 32H，C ； 32H←进位C

位地址

❑能进行位寻址的存储空间：

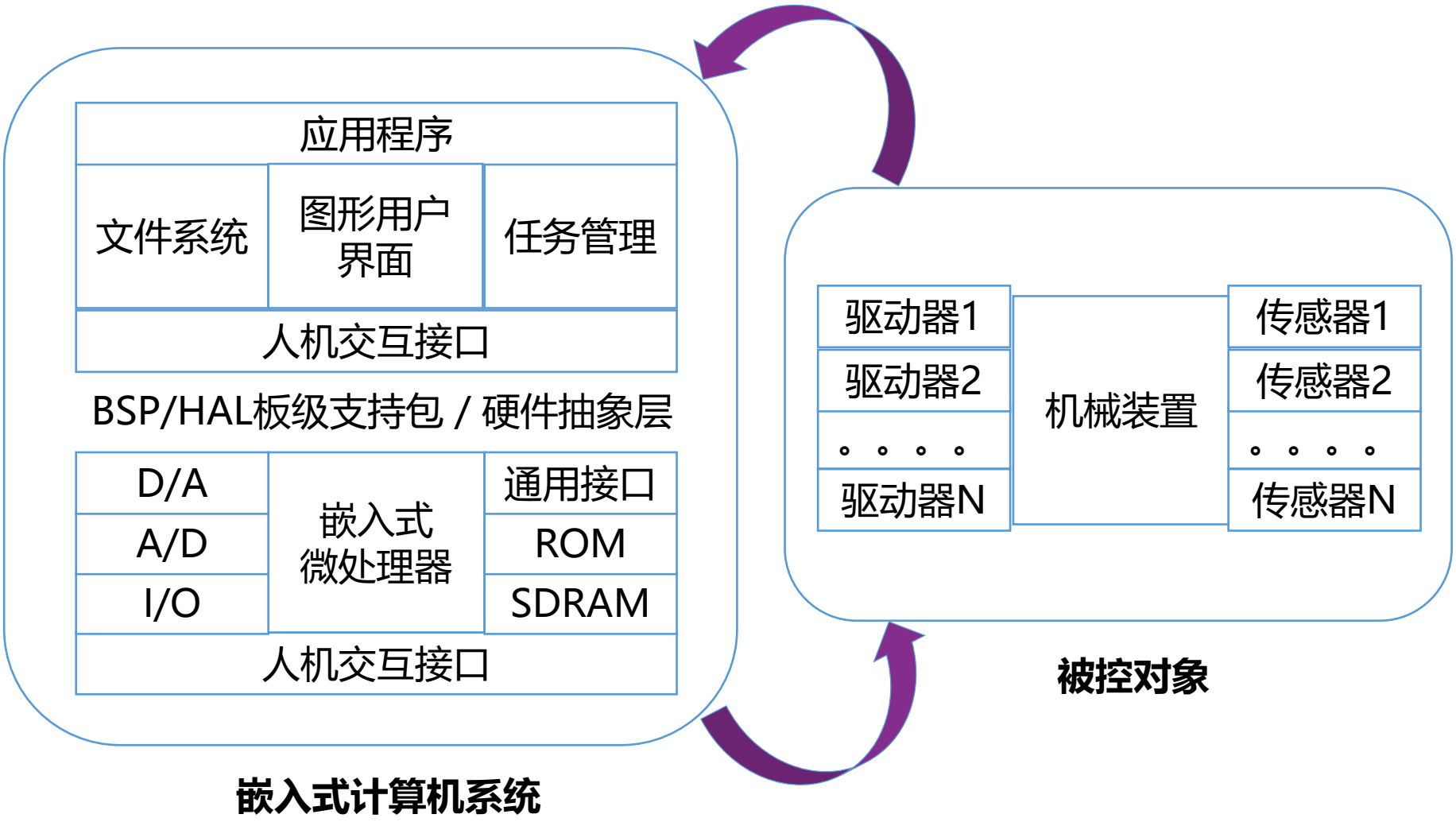
1. 内部RAM中的位寻址区。单元地址为20H～2FH，共16个单元128位，位地址是00H～7FH。
2. 专用寄存器的可寻址位。可供位寻址的专用寄存器共有11个，实际寻址位83位。

清華大學
Tsinghua University

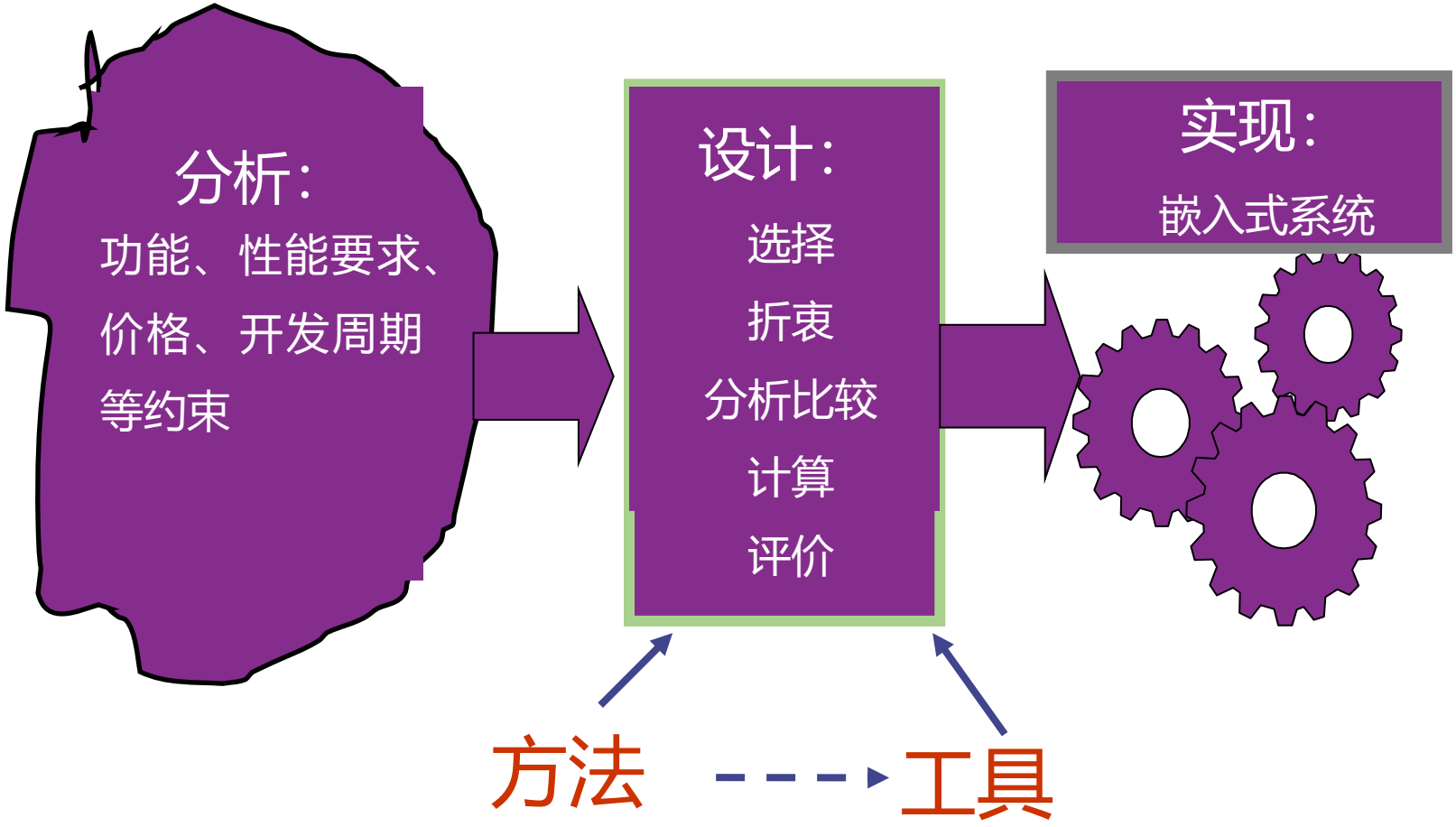
提纲

- 嵌入式处理器介绍
- 体系结构
- 外围设备
- 51单片机介绍
- 嵌入式系统设计

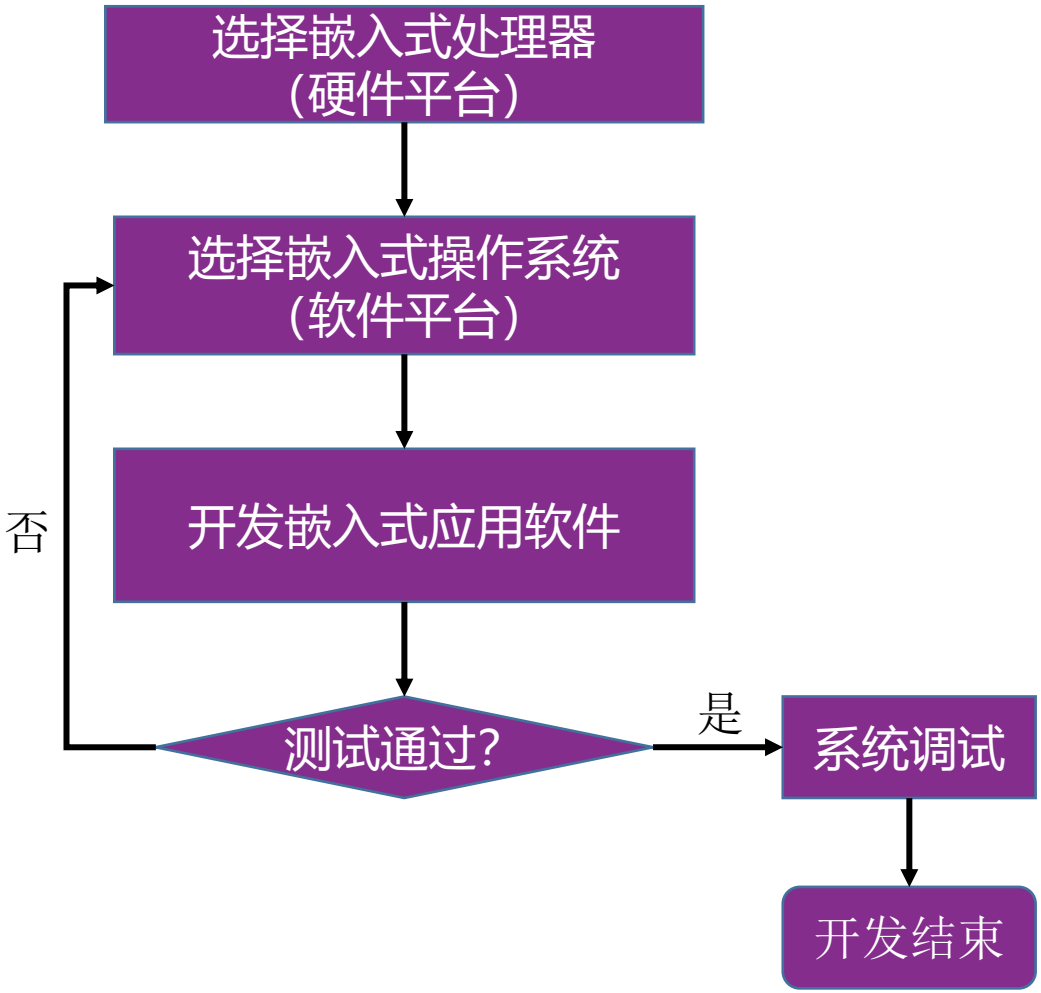
嵌入式系统的软硬件框架



嵌入式系统设计过程



嵌入式系统开发流程



处理器的选择

- 处理器作为嵌入式系统的核心部件，由于种类繁多，我们在选择时需要考虑的主要因素有
  - 处理器性能
    - GreenOrbs的经历：从都够用，到ROM不够，再到RAM不够
  - 技术指标
  - 功耗
  - 软件支持工具
  - 是否内置调试工具
- 其他因素
  - 自己设计和制备硬件，可以降低成本；选择相对成熟的硬件，可以加快开发进程。。。

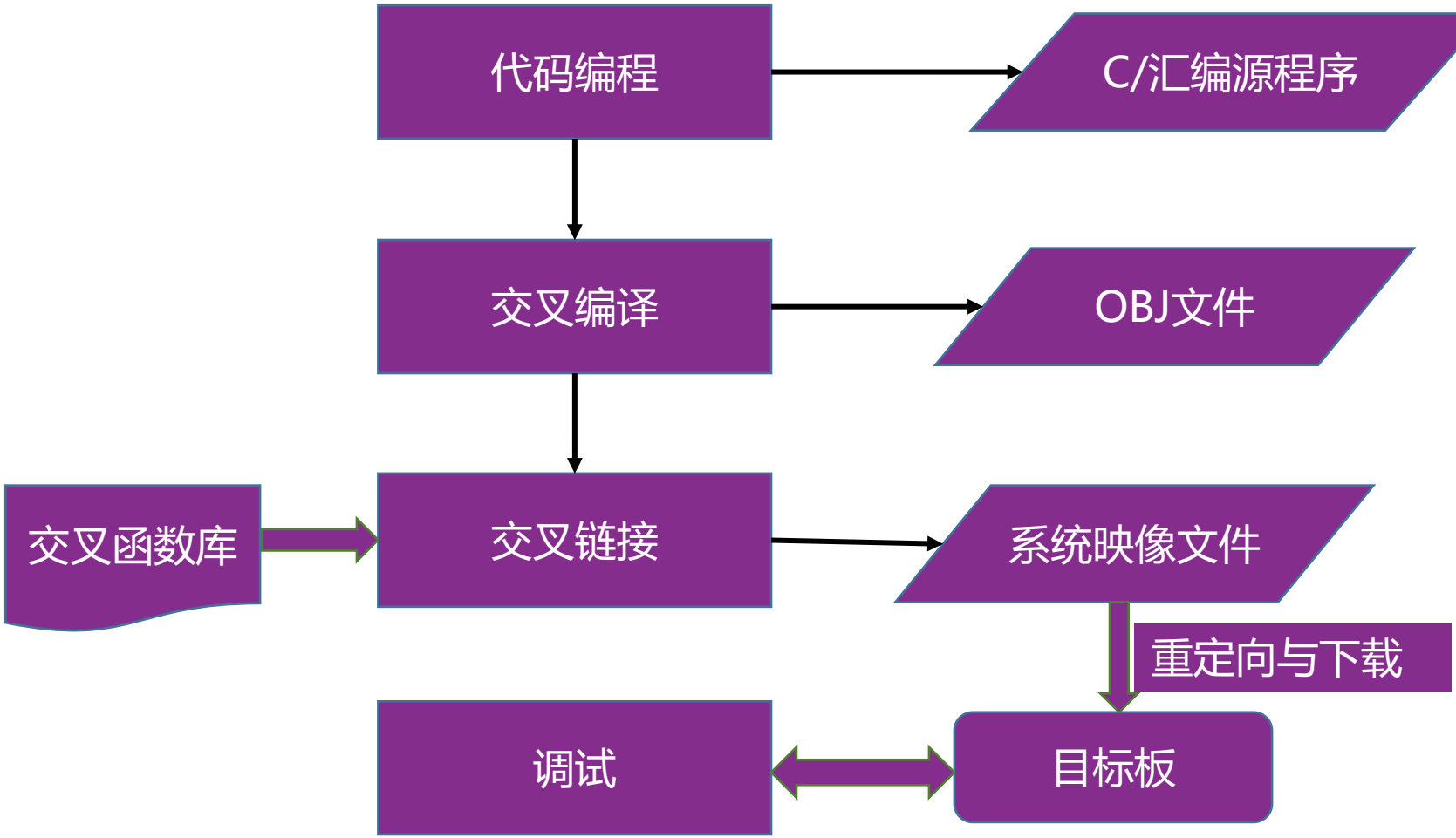
选择芯片的考虑因素

- 技术先进性是最后的考虑因素
- 成本是最主要的
  - 研发成本
    - 技术成熟度
    - 技术服务
  - 产品成本
    - 器件成本
    - 生产成本

示例

- 没有界面
  - 低端处理器：51、430、AVR、PIC、Cortex-M
- 以太网
  - 用专门以太网芯片：WizNet
  - 串口网络
- 人机界面
  - 果断拥抱Android
  - pcDuino、cubieboard：安志A10

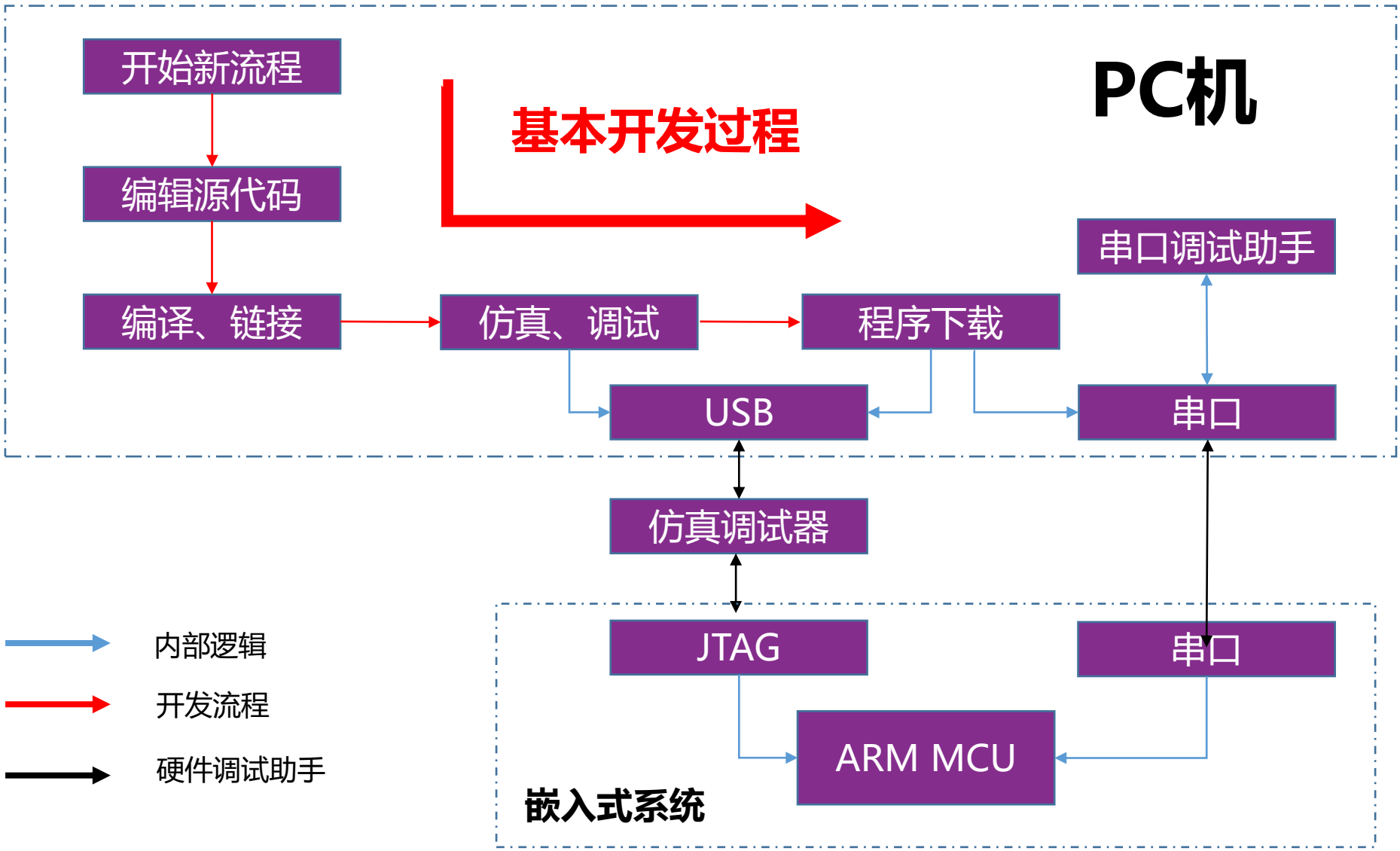
嵌入式软件设计流程



通常由软件实现的部分

- 操作系统功能
  - 任务调度
  - 资源管理
  - 设备驱动
- 协议栈
  - TCP / IP
  - 应用软件框架
  - 除基本系统、物理接口、基本逻辑电路，许多由硬件实现的功能都可以由软件实现。

基本开发流程


开始新流程
编辑源代码
编译、链接
基本开发过程
仿真、调试
程序下载
串口调试助手
USB
串口
PC机
仿真调试器
JTAG
串口
ARM MCU
嵌入式系统
内部逻辑
开发流程
硬件调试助手

软硬件技术发展对嵌入式系统设计的影响

- 硬软件设计的趋势——融合、渗透
  - 硬件设计的软件化
    - VHDL, Verilog
    - HANDL-C
  - 软件实现的硬件化
    - 各种算法的ASIC（专用集成电路）
- 对系统设计的影响——协同设计
  - 增加灵活性
  - 增加了风险

软件硬件协同设计的设计流程

- 用HDL语言和C语言进行系统描述并进行模拟仿真和系统功能验证；
- 对软硬件实现进行功能划分，分别用语言进行设计并将其综合起来进行功能验证和性能预测等仿真确认(协调模拟仿真)；
- 如无问题，则进行软件和硬件详细设计；
- 最后进行系统测试。

案例阅读

任务确定

- **医疗实时监护系统**
  - 单参数监护：单一生命体征监控
    - 心电（ECG）、血压（NIBP）、体温（TEMP）、血氧饱和度（SPO2）
  - 综合监护：社区服务、现场紧急救护
  - 重症监护系统：ICU（Intensive Care Unit）

需求分析

ICU（Intensive Care Unit）：重症监护

- 重症加强护理病房。把危重病人集中起来，在人力、物力和技术上给予最佳保障，以期得到良好的救治效果。
- 中小医院是一个病房，大医院是一个特别科室
- ICU的设备必须配有床边监护仪、中心监护仪、多功能呼吸治疗机、麻醉机、心电图机、除颤仪、起搏器、输液泵、微量注射器、气管插管及气管切开所需急救器材。
- 条件较好的医院，还配有血气分析仪、微型电子计算机、脑电图机、B超机、床旁X线机、血液透析器、动脉内气囊反搏器、血尿常规分析仪、血液生化分析仪等。
- 专科与综合ICU：
  - 烧伤ICU、心血管外科ICU、新生儿ICU等)
  - CCU（Coronary heart disease）是专科ICU中的一种，针对重症冠心病
  - 而设的。

需求分析-2

- **单参数监护**
  - 心电（ECG）
    - 心电参数采样存储
    - 心电参数监护报警（心脉生命监护报警，GPS定位，GPRS报警）
    - 心电图综合分析（图形显示）
      - 心律失常分析、起搏分析、ST段分析等，并可根据临床需求进行监测信息回顾，包括趋势图、表的信息存储功能，存储时间长，信息量大
  - 血压（NIBP）：电子血压计
  - 体温（TEMP）：体温计、全息体温图
  - 血氧饱和度（SPO2）
  - 一般用数字显示

需求分析-3

❑综合监护

- ❑监测ECG、NIBP、SPO2、TEMP等基本参数
- ❑同时，也可以连续监测有创血压、心输出量、特殊麻醉气体 （小型ICU）
- ❑数字和波形同屏显示

确定软/硬件界面

- **单参数监护：**
  - 血压（NIBP）、体温（TEMP）监控报警
    - 硬件提供参数通道，软件采样、计算、存储、数字显示; 无OS
  - 全息体温图（TEMP）：需 μClinux, WIN CE
  - 心电（ECG）、血氧饱和度（SPO2）监控报警
    - 硬件提供参数采样，软件计算、存储、数字显示；μC-OS
  - 心电图综合分析（图形显示）
    - 硬件提供参数采样，软件计算、存储、显示；μClinux, WIN CE
- **综合监护：**
  - 硬件提供参数采样，软件扫描、计算、存储、显示（数字或简图）；linux, WinCE
- **重症监护系统：**
  - 集成方案：硬件提供数据接口，软件扫描采样、转发、报警；μCOS / μClinux
  - 综合方案：硬件提供参数采样，软件扫描采样、计算、报警、存储、 综合显示；linux, WinCE

硬件裁剪—处理器

- **单参数监护：**
  - 血压（NIBP）、体温（TEMP）监控报警
    - 单片机：8051，z-8等
  - 全息体温图（TEMP）：ARM7： S3C44B0X， StrongARM
  - 心电（ECG）、血氧饱和度（SPO2）监控报警
    - ARM7： S3C44B0X， StrongARM
  - 心电图综合分析（图形显示）
    - ARM9，MIPS
- **综合监护：**
  - ARM9，MIPS
- **重症监护系统：**
  - 集成方案：ARM7，ARM9，MIPS
  - 综合方案：ARM9~ARM10，MIPS：XScale, 386EX

硬件裁剪—I/O接口与输入/输出设备

- **单参数监护：**
  - 血压（NIBP）、体温（TEMP）监控报警
    - 计数器/定时器（或PWM，ADC），PVC按键，LED（数字LCD）,蜂鸣器
  - 全息体温图（TEMP）：加显示接口，点阵LCD（小型CRT）
  - 心电（ECG）、血氧饱和度（SPO2）监控报警
    - ADC接口(I^2C)，UART，Watch dog，PVC按键，GPS， LED（数字LCD）,GPRS
  - 心电图综合分析（图形显示）
    - ADC接口(I^2C)，UART，Watch dog，PVC按键，点阵LCD（小型CRT）
- **综合监护：**
  - ADC接口(I2C)，UART，Watch dog，PVC按键，点阵LCD（小型CRT）
- **重症监护系统：**
  - 集成方案： 100M Ethernet （或USB，CAN），各种监护仪器接口，PVC按键， LED（数字LCD）,蜂鸣器
  - 综合方案： ADC接口(I2C)，UART，Watch dog，PVC按键，点阵LCD（小型CRT）

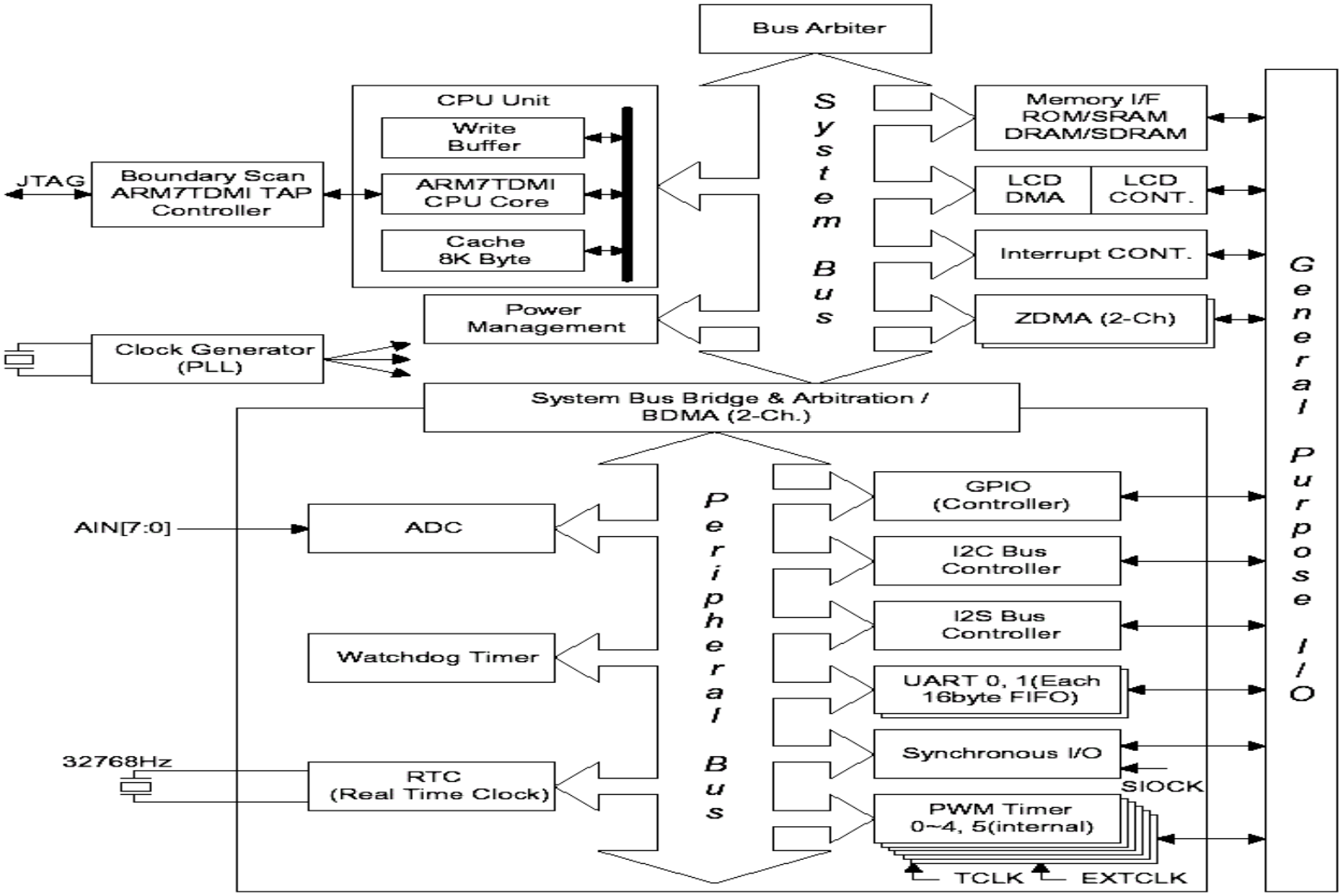
确定硬件结构框架—心电参数监护报警系统

- **需求分析**
  - 心电（脉）生命监护报警，GPS定位，GPRS报警
- **软/硬件界面**
  - 硬件提供参数采样，软件计算、存储、数字显示；μC-OS（linux）
- **处理器：** S3C44B0X（ARM7）
- **I/O接口与输入/输出设备**
  - ADC接口，UART，Watch dog，PVC按键，GPS，点阵LCD ,GPRS
  - 线性Flash(NOR)，SDRAM，非线性Flash(NAND)

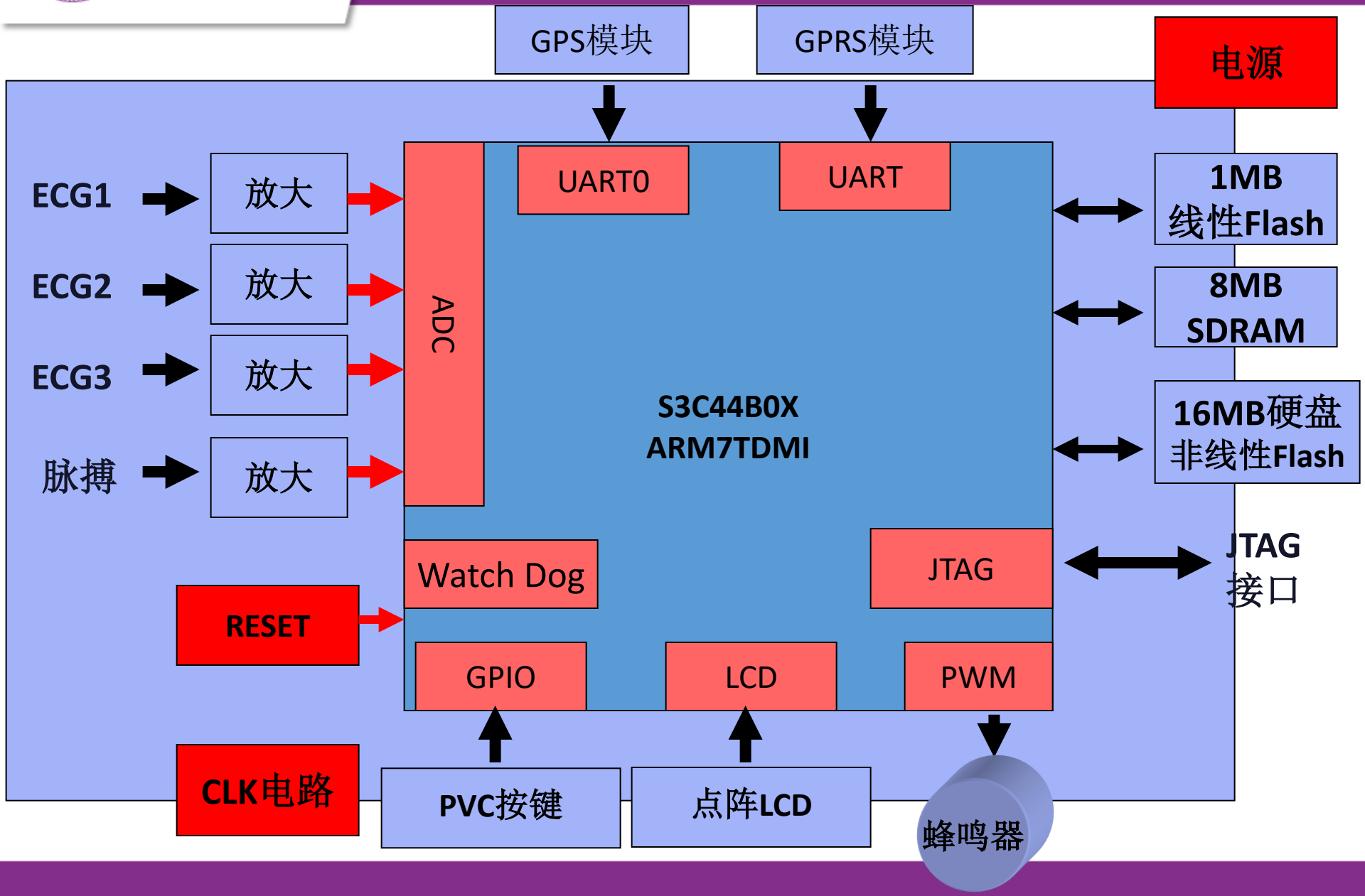
S3C44B0X功能

- Samsung S3C44B0X微处理器是三星公司专为手持设备和一般应用提供的高性价比和高性能的微控制器解决方案，它使用ARM7TDMI核，工作在66MHZ。为了降低系统总成本和减少外围器件，这款芯片中还集成了下列部件：
- 8KB Cache、外部存储器控制器、LCD控制器、4个DMA通道、2通道UART、1个多主I2C总线控制器、1个IIS总线控制器，5通道PWM定时器及一个内部定时器、71个通用I/O口、8个外部中断源、实时时钟、8通道10位ADC等。

S3C44B0X芯片体系结构



硬件结构框架-心脉生命监护报警




Thank you!