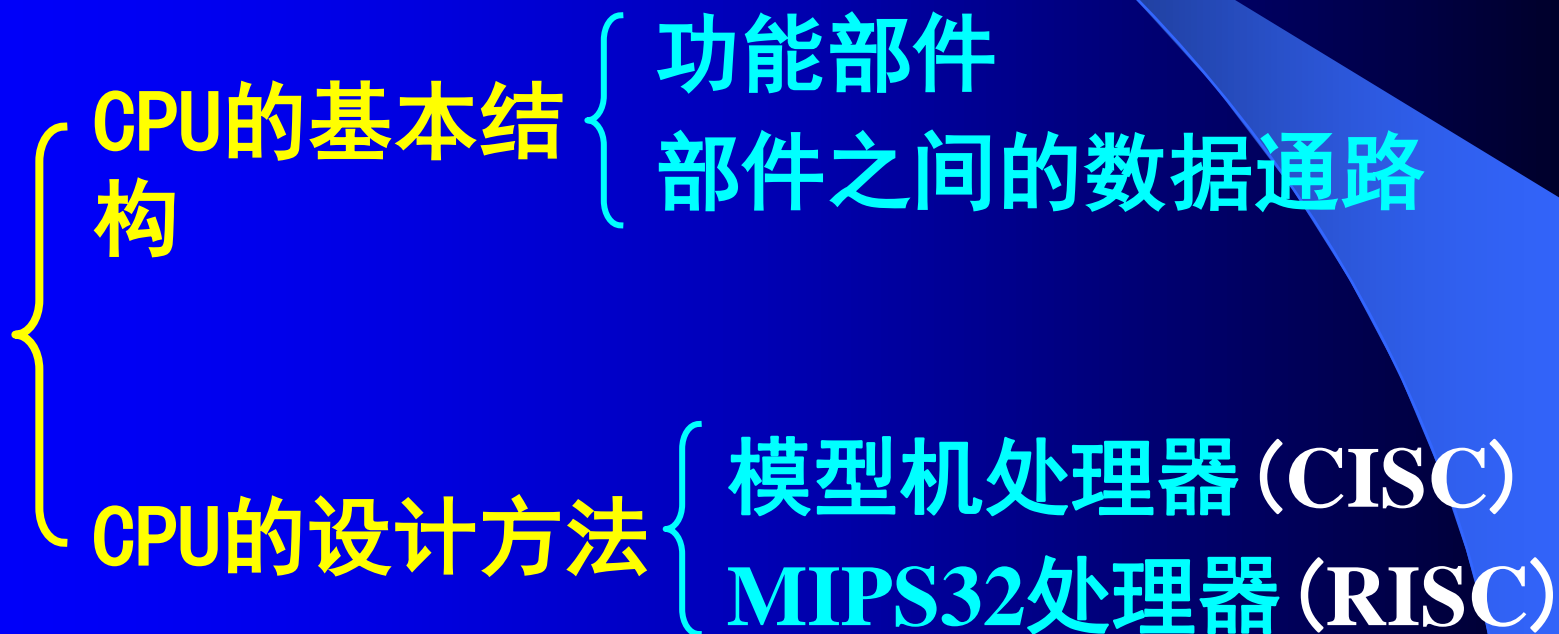


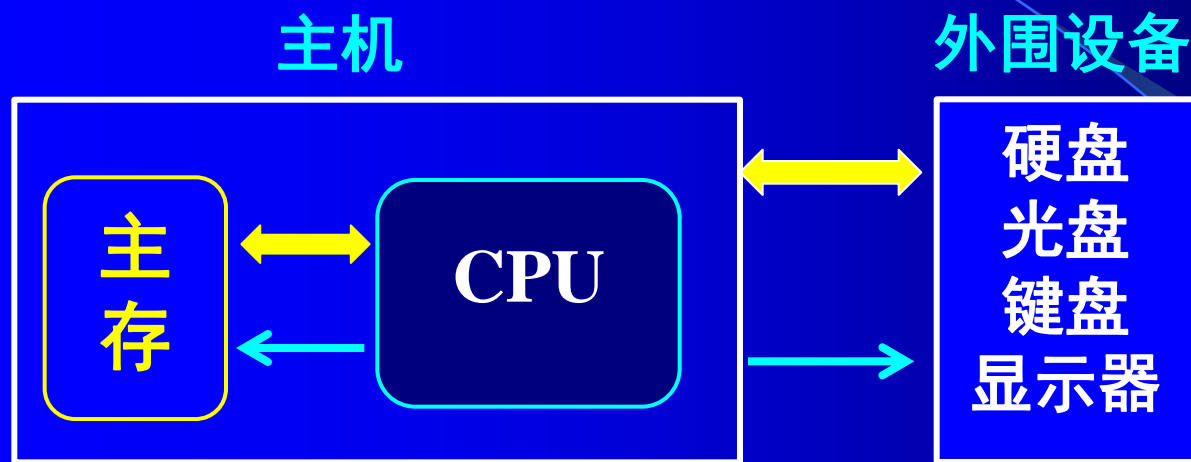
第3章 CPU子系统

※本章主要介绍:



3.1 CPU概述

※CPU在计算机中的角色

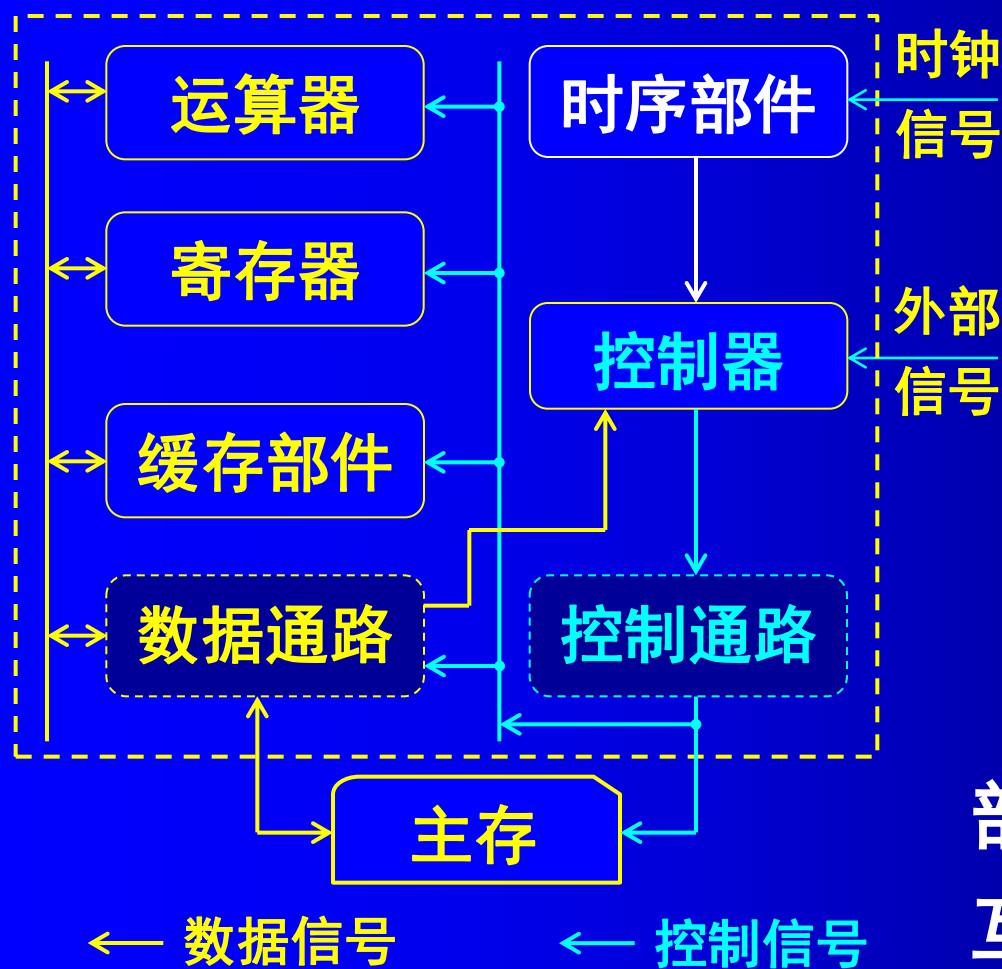


CPU是计算机中的核心部件
结构最复杂、
技术难度最高！

- ①数据运算功能
- ②系统控制功能

3.1.1 CPU的基本结构

1、总体结构模型



※主要部件:

- ①时序系统
- ②控制部件
- ③缓存部件
- ④寄存器(堆)
- ⑤运算部件



部件通过数据/控制通路互连，实现信息交互

2、CPU的主要部件

从硬件的角度，CPU内部的主要部件包括：

(1) 运算部件

(2) 缓存部件

(3) 寄存器

(4) 控制器

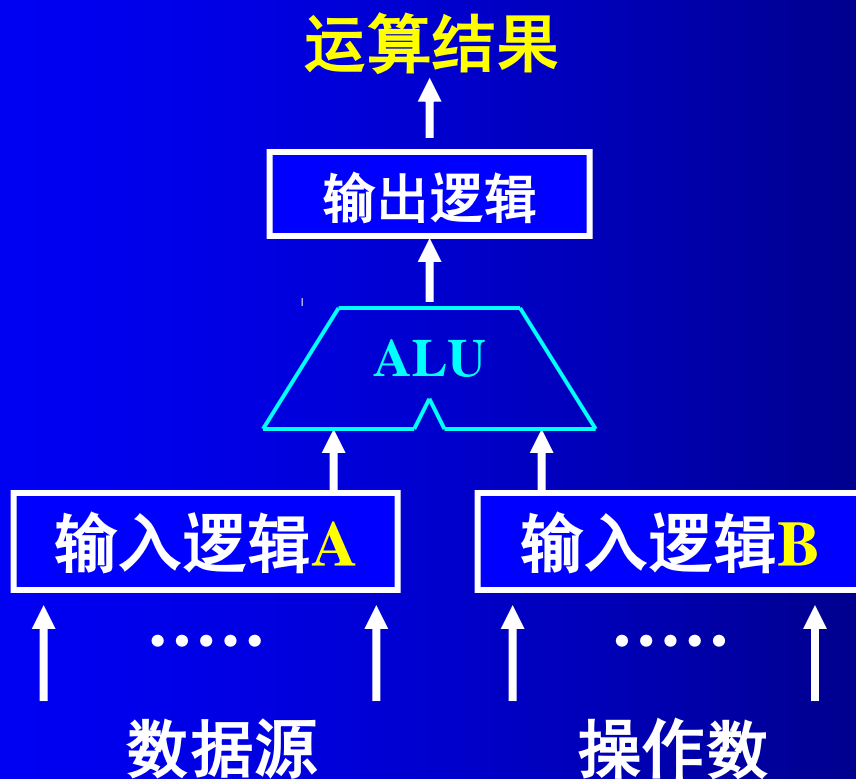
(5) 时序部件

通过数据通路/控制通路
互相连接

各种功能部件，分别通过数据通路/控制通路
互相连接形成CPU的硬件架构 → CPU的微架构

(1) 运算部件

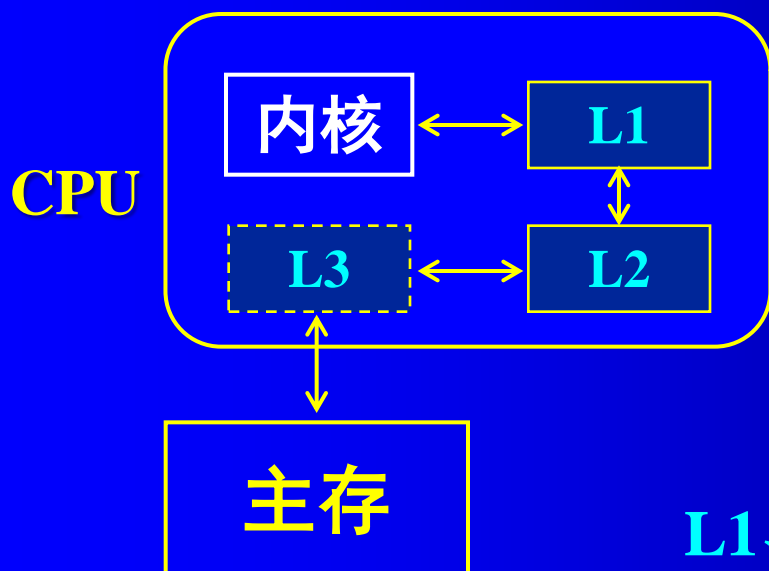
对操作数进行运算，主要是算术运算/逻辑运算
基本组成如下：



(2) 缓存部件

为提高CPU从主存中读取指令/数据的效率，在CPU内部集成了多级缓存部件。

[作用]缓存从主存中读取的部分指令/数据



程序的执行具有局部性
提前把需要的指令数据调入缓存

L1、L2和L3就是缓存部件

(3) 寄存器组(堆)

存储各种用途的数据信息。

[部件选用]一般用小容量的多端口存储器来构成寄存器组，其中1个存储单元作为1个寄存器。

基本类型如下：

①通用寄存器：多个

通用寄存器有全局唯一地址，可通过地址码访问，可在机器指令中直接使用。

[功能]提供操作数、地址码、存放运算结果等。

②暂存器

[特征] 多个，内部专用，无需分配地址码，不能在机器指令中使用。

[主要用途] 用来暂存产生的临时数据，以备在后续操作过程中使用。

③指令寄存器（IR）

[主要用途]

只有1个，用于存放**指令代码**。

从存储器(或者**指令缓存**)中读取到指令以后，就直接存入到指令寄存器中。

④程序计数器（PC）

[主要用途] 仅1个，用来指明指令在存储器中的存放位置，即存储单元的地址码。



[注意] 取指令结束后，PC保存的地址码自动修改，以指向下一条指令的存储单元，修改量取决于指令字长和存储器的编址单位。

指令存储器

⑤程序状态字寄存器（PSW）

[主要用途]

仅1个，记录现程序的运行状态和程序的工作模式。

❖ PSW-特征位

也叫标志位，反映CPU的当前状态。

指令执行时，根据情况**自动设置**这些特征位，作为后续操作的判断依据，通常有**5类**：

进位 C	溢出 V	零值 Z	负值 N	奇偶 P	...
---------	---------	---------	---------	---------	-----

自动设置（具备该特征，就设置该标志位=1）

❖ PSW-编程设定位

PSW中某些位或字段可通过程序来设定，以决定程序的调试、对中断的响应、程序的运行模式等。

跟踪位 T	允许中断 I	程序优 先级P	运行模式
----------	-----------	------------	------

⑥地址寄存器（MAR）

[主要用途]

只有1个，读写存储器时，先要定位存储单元，因此设置MAR来存放目标单元的地址码。

先将有效地址送入MAR，再启动后续的读写操作。

⑦数据缓冲寄存器（MBR）

[主要用途]

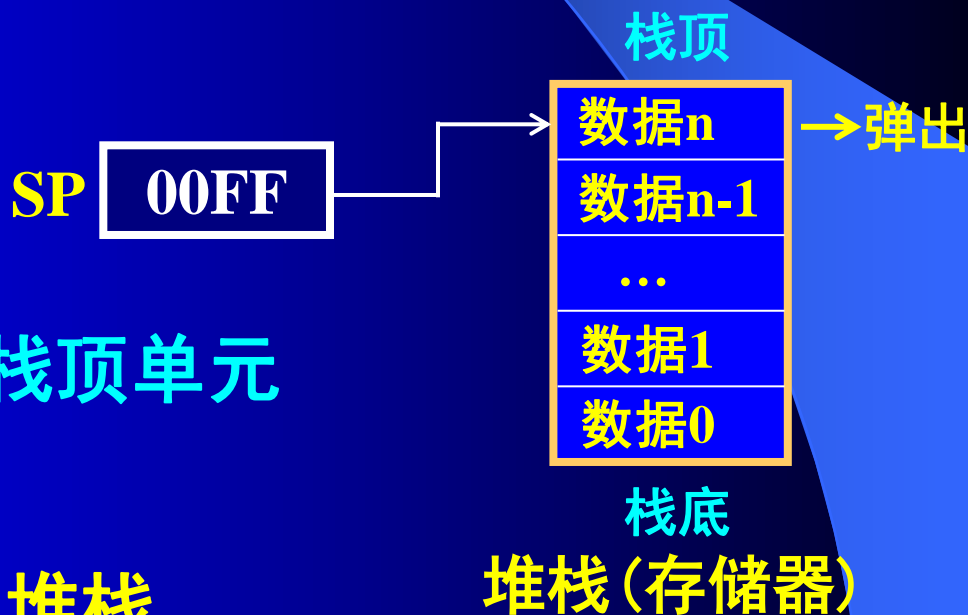
只有1个，过渡性地存放CPU与主存之间交换的数据。
无论是从主存读取的数据，还是写入到主存的数据，都要经过**MBR**。

⑧堆栈指针（SP）

[主要用途]

仅1个，固定存放堆栈的**栈顶单元**的地址码。

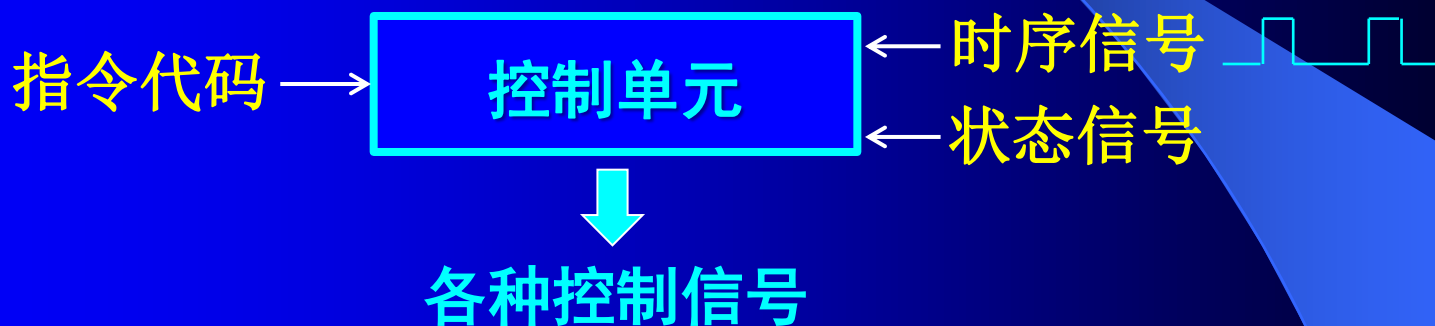
根据这个地址码，去读写**堆栈**。



4、控制器

[主要作用]

根据**指令**、**时钟信号**、**外部信号**等信息，产生各种**控制信号**（**微命令**），以便控制各种功能部件协同工作，完成指令的功能。



根据产生微命令的方式，有两类控制单元：

①**组合逻辑控制器**

组合逻辑硬件电路 → 控制信号

②**微程序控制器**

微程序译码 → 控制信号

5、时序部件

①时序信号

[定义]周期、节拍、脉冲等**频率型**信号序列。

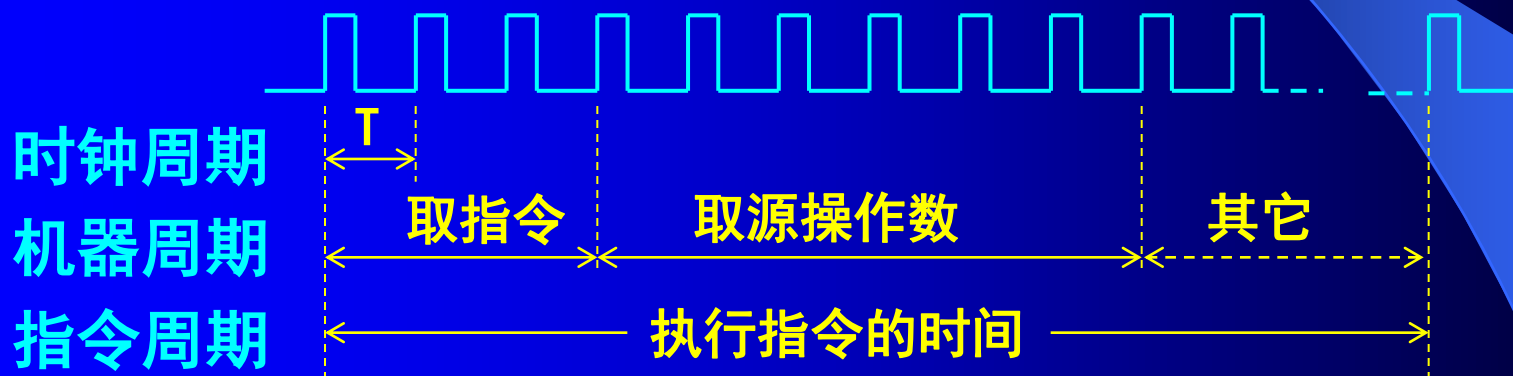
产生时序信号的部件称为**时序发生器**或**时序系统**，由**1个低频振荡器和倍频逻辑**组成。



低频信号振荡器：它是一个低频脉冲源，能输出固定频率的基准脉冲信号**(外频)**，作为**系统时钟信号**。

系统时钟信号经过倍频放大以后，产生执行指令所需要的各种时序信号：

- ①节拍信号，即CPU时钟周期信号；
- ②工作周期信号，即机器周期信号；
- ③指令周期信号。



指令周期包括若干 (≥ 2) 机器周期。

机器周期包括若干 (≥ 1) 时钟周期。

3.1.2 CPU的工作原理

1、主要功能

- ✓处理指令-控制指令的执行顺序；
- ✓执行操作-产生控制信号控制部件工作；
- ✓控制时间-控制各步操作的时序；
- ✓数据运算-算术和逻辑运算；

2、执行指令的流程

- ✓读取指令-从存储器中读取；
- ✓指令译码-通过控制器进行、产生控制信号；
- ✓指令执行-寻址、取数、运算；
- ✓后续工作-保存结果、响应外部请求等；

3、部件的控制方式

CPU是计算机内部技术最复杂的子系统：

- 部件众多
- 架构复杂
- 功能多样

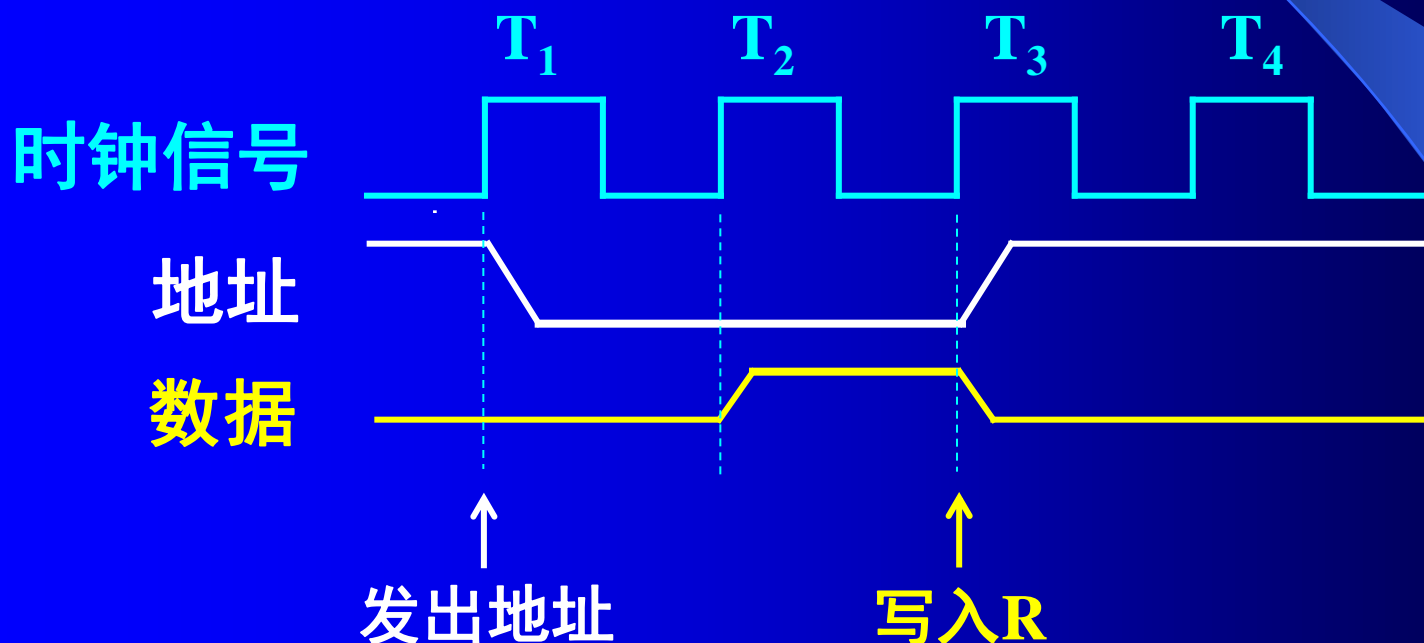
如何控制这些部件协同工作，完成指令功能？

- ① 同步控制方式；
- ② 异步控制方式；

※同步控制方式的特点

- ✓ 每步操作都向统一的外部时序信号对齐；
- ✓ 各步操作之间无交互。

[举例] 发出地址、送出数据与写入R的协同

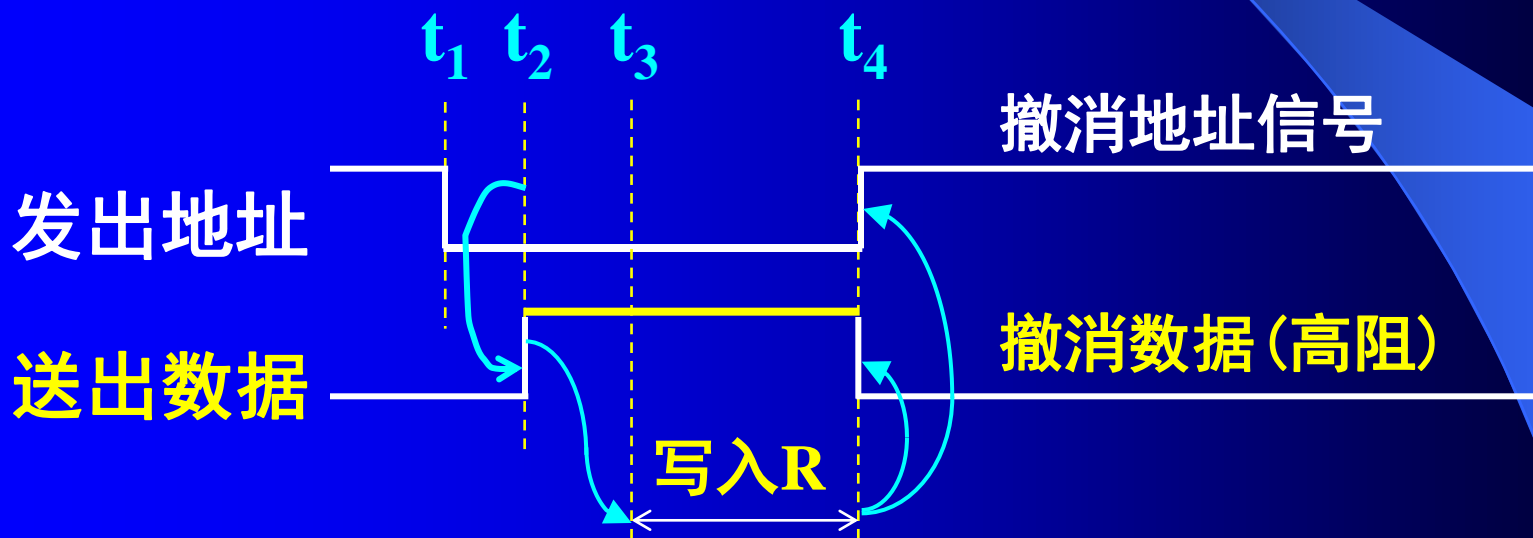


三个操作靠外部时钟信号的驱动来实现协同！

※异步控制方式的特点

- ✓ 每步操作都不需向统一的外部时序信号对齐；
- ✓ 各步操作之间通过交互应答来实现协同

[举例] 发地址信号、送出数据与写入R的协同



三个操作靠相互之间的控制信号来实现协同

3.1.3 外部连接与I/O控制任务

1、外部连接类型



- ✓ 单处理机系统：通过前端总线与北桥芯片组连接；
- ✓ 多处理机系统：高性能CPU中集成了主存、视频和PCI-E接口，CPU之间通过QPI、与芯片组之间通过DMI总线互连。

2、在I/O控制中的任务

主机←接口→外围设备

主机与外设之间进行数据输入/输出操作时，在不同的I/O控制模式下，CPU承担的任务各不相同：

- ①程序传送模式： **CPU直接执行I/O指令**
- ②中断模式： **CPU执行中断服务程序**
- ③DMA模式： **CPU管理DMA控制器、善后处理；**
- ④IOP和PPU模式： **CPU组织I/O程序，管理IOP与PPU，以及善后处理；**

3.1.4 CPU发展历程

1、1946-1970

[技术特点] 电子管和晶体管CPU阶段，体积大、功耗高、集成度低、运算速度慢。 ←非微处理器

2、1971-1973

→ CPU进入单核微处理器时代

[技术特点] 中小规模集成电路，4位或8位微处理器。

3、1974-1977

[技术特点] 8位中高档微处理器成为主流，1百万级IPS。

4、1978-1984

[技术特点] 16位微处理器，3um工艺，约2百万级IPS。

5、1985-1992

[技术特点] 32位微处理器，2um工艺，指令周期约0.16us，开始集成协处理器，采用流水线，千万级IPS。

[典型代表] Intel 80486；

6、1993-2002

[技术特点] 32位微处理器，集成度更高、工艺<1um，指令更丰富、功能更强，IPS突破1亿。

[典型代表] Intel 32位奔腾系列处理器；

7、2003-2004

[技术特点] 64位单核处理器，0.13um，晶体管的集成度上亿，主频高达2.0GHz。

[典型代表] Intel 64位奔腾4处理器；

8、2005-现在

[技术特点] 64位、多核、多线程处理器为主。

低于100nm级的工艺制程。

[典型代表] AMD Athlon 64X2、Intel Pentium EE;

Intel 酷睿i3/i5/i7/i8系列处理器;

※国内的情况:

2002, 龙芯1(32位/MIPSIII/7级流水/266MHz)

2005, 龙芯2(64位/0.18um/1GHz), ≈P4

2009, 龙芯3A(4核/65nm/1GHz), 飞腾1000(8核64线程)

2012, 龙芯3B-1500(8核/28nm/1.5GHz)

2013, 飞腾1500(16核 /SPARC V9/40nm/1.8GHz)

国内外差距很大、任重道远!