

► 冯诺依曼计算机的特点。

- λ 计算机由运算器、控制器、存储器、输入设备、输出设备五大部件组成；
- λ 指令和数据以同同等地位存放于存储器内，并可以按地址访问；
- λ 指令和数据均用二进制表示；
- λ 指令由操作码、地址码两大部分组成，操作码用来表示操作的性质，地址码用来表示操作数在存储器中的位置；
- λ 指令在存储器中顺序存放，通常自动顺序取出执行；
- λ 机器以运算器为中心（原始冯·诺依曼机）。

► 名词解释：MIPS, CPI, FLOPS, 机器字长, 存储字长, 指令字长, CACHE, EPROM 等等。

MIPS: Million Instruction Per Second, 每秒执行百万条指令数，为计算机运算速度指标的一种计量单位。

CPI: Cycle Per Instruction, CPU 每执行一条指令所需的时钟周期数。

FLOPS: floating-point operations per second, 每秒所执行的浮点运算次数

机器字长: 指 CPU 一次能处理的二进制数据的位数，通常与 CPU 的寄存器位数有关。

存储字长: 一个存储单元所存储的二进制代码的总位数。

指令字长: 机器指令中二进制代码的总位数。

EPROM: 可擦除可编程只读存储器

► 什么是计算机系统，说明计算机系统的层次结构。

从计算机系统的层次结构来看，它通常可以有 5 个以上的层次，在每一个层次上都能进行程序设计。从下到上依次为

- 1.微程序机器级，微指令由硬件直接执行；
- 2.传统机器（机器语言机器），用微程序解释机器指令；
- 3.操作系统级，一般用机器语言程序解释作业控制语句；
- 4.汇编语言级，由汇编程序支持和执行；
- 5.高级语言级，采用高级语言，有各种高级语言编译程序支持和执行。还可以有
- 6.应用语言机器级，采用各种面向问题的应用语言。

► 计算机体系结构、组成与实现之间的关系。

计算机体系结构是指那些**能够被程序员所见到的计算机系统的属性**，如指令系统、数据类型、寻址技术组成及 I/O 机理等。计算机组成是指**如何实现计算机体系结构所体现的属性**，包含对程序员透明的硬件细节，如指令的实现 (...), 组成计算机系统的各个功能部件的结构和功能，及相互连接方法等。

► 比较链式查询方式、计数器定时查询方式和独立请求方式各自的特点。

总线判优控制解决多个部件同时申请总线时的使用权分配问题；

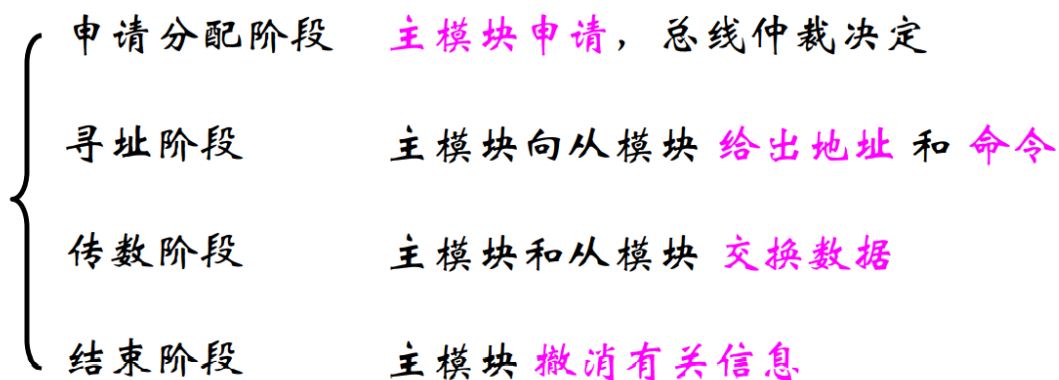
链式查询方式连线简单，易于扩充，对电路故障最敏感；

计数器定时查询方式优先级设置较灵活，对故障不敏感，连线及控制过程较复杂；

独立请求方式速度最快，但硬件器件用量大，连线多，成本较高。

► 总线周期的四个阶段。

## 2. 总线传输周期



### ▶ 试比较同步通信和异步通信。

同步通信：指由**统一时钟**控制的通信，控制方式简单，灵活性差，当系统中各部件工作速度差异较大时，总线工作效率明显下降。适合于**速度差别不大**的场合。

异步通信：指**没有统一时钟**控制的通信，部件间采用**应答方式**进行联系，控制方式较同步复杂，灵活性高，当系统中各部件工作**速度差异较大**时，有利于提高总线工作效率。

### ▶ 试比较缓存-主存和主存-辅存两个层次的相同点和不同点。

cache-主存层次主要是为了解决 **CPU 与主存速度不匹配**的问题...

主存-辅存层次主要是解决**存储系统容量**问题...

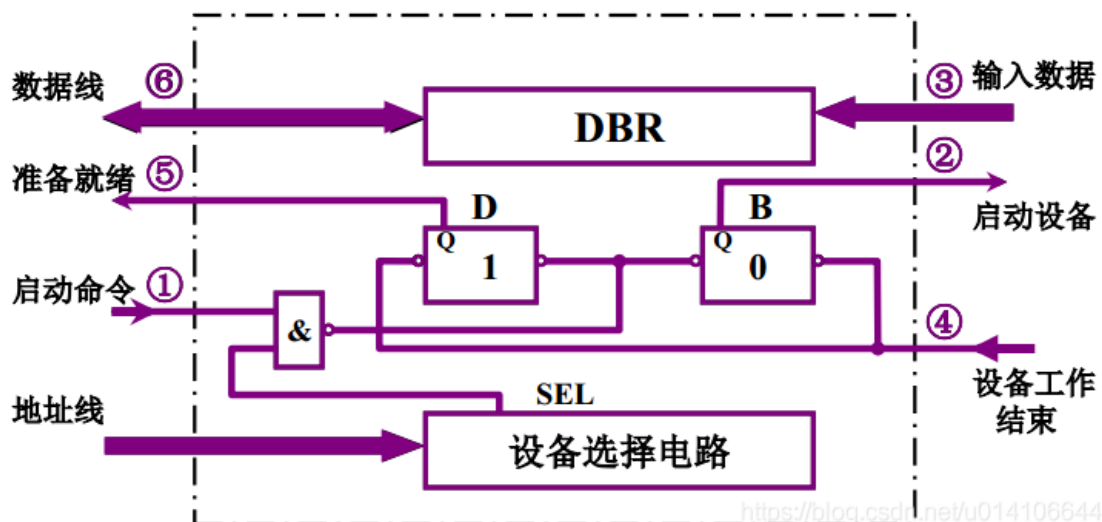
cache-主存之间的数据调动是由**硬件自动完成**，对程序员是透明的。

主存-辅存之间的数据调动是由**硬件和操作系统共同完成的**。

### ▶ 提高访存速度可以采取哪些措施？

单体多字、多体并行、高性能存储芯片 (SDRAM 同步 DRAM ,RDRAM ,CDRAM 带 Cache 的 DRAM)

### ▶ 程序查询方式接口电路的基本组成及程序流程



解：程序查询接口工作过程如下（以输入为例）：

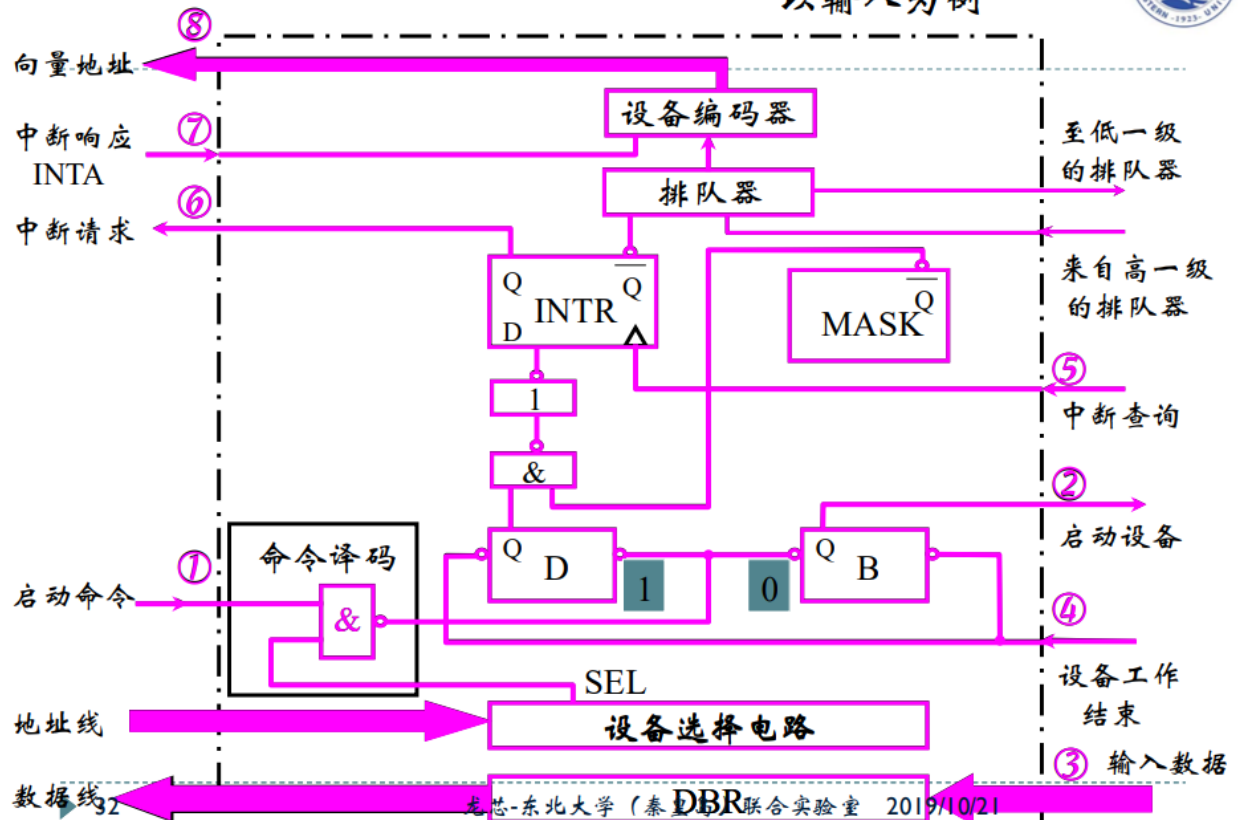
- 1) CPU发I/O地址→地址总线→接口→设备选择器译码→选中，发SEL信号→开命令接收门；
- 2) CPU发启动命令→D置0，B置1→接口向设备发启动命令→设备开始工作；
- 3) CPU等待，输入设备读出数据→DBR；
- 4) 外设工作完成，完成信号→接口→B置0，D置1；
- 5) 准备就绪信号→控制总线→CPU；
- 6) 输入：CPU通过输入指令（IN）将DBR中的数据取走；

► 中断的处理过程，程序中断接口电路



## 2. I/O 中断处理过程

以输入为例



### ► 中断服务程序的流程

- (1) 保护现场
- (2) 中断服务
- (3) 恢复现场
- (4) 中断返回

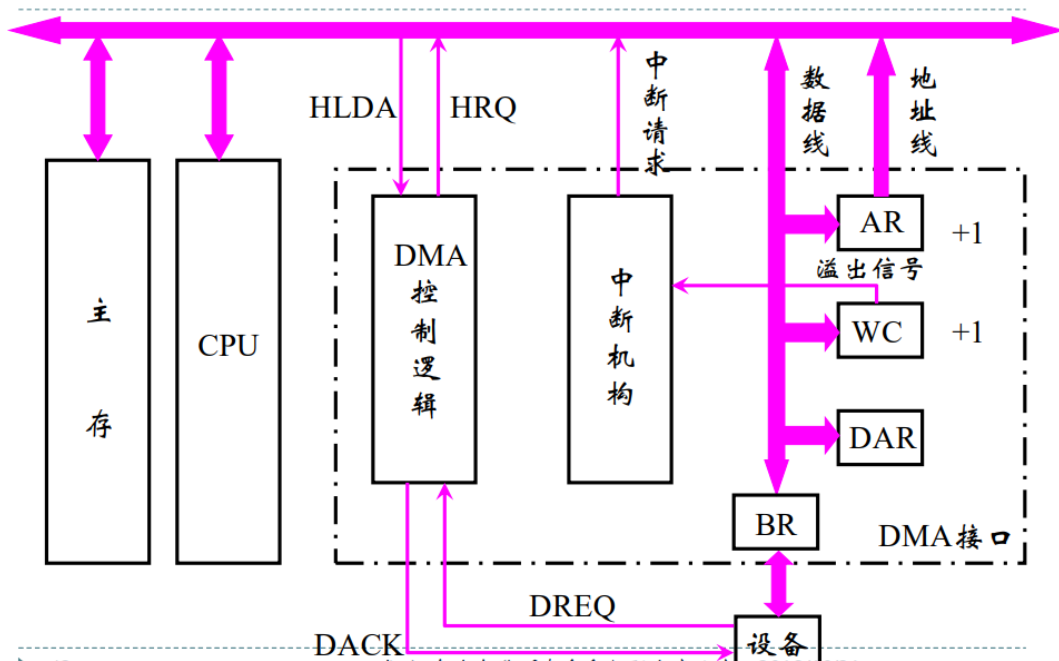
### ► DMA 接口的功能是什么

- (1) 向 CPU 申请 DMA 传送
- (2) 处理总线 控制权的转交
- (3) 管理 系统总线、控制 数据传送
- (4) 确定 数据传送的 首地址和长度
- (5) DMA 传送结束时, 给出操作完成信号

► DMA 接口的基本组成有哪些



## 2. DMA 接口组成



► 简述 DMA 的工作过程

- (1) 当设备准备好一个字时，发出选通信号，将该字读写到 DMA 的数据缓冲寄存器 (BR) 中，表示数据缓冲寄存器为满。
- (2) 与此同时设备向 DMA 接口发请求 (DREQ)。
- (3) DMA 接口向 CPU 申请总线控制权 (HRQ)
- (4) CPU 发出 HLDA 信号，表示允许将总线控制权交给 DMA 接口。
- (5) 将 DMA 主存地址寄存器中的主存地址送地址总线，并命令存储器写。
- (6) 通知设备已被授予一个 DMA 周期 (DACK)，并为交换下一个字做准备。
- (7) 将 DMA 数据缓冲寄存器的内容送数据总线。
- (8) 主存将数据总线上的信息写至地址总线指定的存储单元中。
- (9) 修改主存地址和字计数值。
- (10) 判断数据块是否传送结束，若未结束，则继续传送，若已结束，则向 CPU

发送中断请求，标志数据块传送结束。

► 试比较程序查询、程序中断和 DMA 三种方式的综合性能。

(1) 数据传送依赖软件还是硬件。

(2) 传送数据的基本单位。

(3) 并行性。

(4) 主动性。

(5) 传输速度。

(6) 经济性。

(7) 应用对象。

解：比较如下：

(1) 程序查询、程序中断方式的数据传送主要依赖软件，DMA 主要依赖硬件。（注意：这里指主要的趋势）

(2) 程序查询、程序中断传送数据的基本单位为字或字节，DMA 为数据块。

(3) 程序查询方式传送时，CPU 与 I/O 设备串行工作；程序中断方式时，CPU 与 I/O 设备并行工作，现行程序与 I/O 传送串行进行；DMA 方式时，CPU 与 I/O 设备并行工作，现行程序与 I/O 传送并行进行。

(4) 程序查询方式时，CPU 主动查询 I/O 设备状态；程序中断及 DMA 方式时，CPU 被动接受 I/O 中断请求或 DMA 请求。

(5) 程序中断方式由于软件额外开销时间比较大，因此传输速度最慢；程序查询方式软件额外开销时间基本没有，因此传输速度比中断快；DMA 方式基本由硬件实现传送，因此速度最快；

注意：程序中断方式虽然 CPU 运行效率比程序查询高，但传输速度却比程序查

询慢。

(6) 程序查询接口硬件结构最简单，因此最经济；程序中断接口硬件结构稍微复杂一些，因此较经济；DMA 控制器硬件结构最复杂，因此成本最高；

(7) 程序中断方式适用于中、低速设备的 I/O 交换；程序查询方式适用于中、低速实时处理过程；DMA 方式适用于高速设备的 I/O 交换；

### ► 指令的寻址方式

#### 1)立即寻址

所需的操作数由指令的地址码部分直接给出，就称为立即数(或直接数)寻址方式。

#### 2)直接寻址

指令的地址码部分给出操作数在存储器中的地址。

#### 3)隐含寻址

操作数的地址隐含在操作码或者某个寄存器中。

#### 4)间接寻址 **【子程序编制】**

在寻址时，有时根据指令的地址码所取出的内容既不是操作数，也不是下一条要执行的指令，而是操作数的地址或指令的地址，这种方式称为间接寻址或间址。

#### 5)寄存器寻址

计算机的中央处理器一般设置有一定数量的通用寄存器，用以存放操作数、操作数的地址或中间结果。假如指令地址码部分给出某一通用寄存器地址，而且所需的操作数就在这一寄存器中，则称为寄存器寻址。通用寄存器的数量一般在几个至几十个之间，比存储单元少很多，因此地址码短，而且从寄存器中存取数据比从存储器中存取快得多，所以这种方式可以缩短指令长度、节省存储空间，提高指令的执行速度，在计算机中得到广泛应用。

#### 6)寄存器间接寻址



寄存器中给出的是操作数的地址，因此还需要访问一次存储器才能得到操作数。

#### 7)基址寻址

在计算机中设置一个专用的基址寄存器，或由指令指定一个通用寄存器为基址寄存器。操作数的地址由基址寄存器的内容和指令的地址码 A 相加得到。

#### 8)变址寻址 【循环程序】

指令地址码部分给出的地址 A 和指定的变址寄存器 X 的内容通过加法器相加，所得的和作为地址从存储器中读出所需的操作数。这是几乎所有计算机都采用的一种寻址方式。

#### 9)相对寻址 【浮动程序】

把程序计数器 PC 的内容(即当前执行指令的地址)与指令的地址码部分给出的位移量 (disp)之和作为操作数的地址或转移地址，称为相对寻址。

#### 10)堆栈寻址

### ► RISC 的特点，CISC 的特点，RISC 和 CISC 的比较

RISC:

选用使用频度较高的一些简单指令，

复杂指令的功能由简单指令来组合

指令长度固定、指令格式种类少、寻址方式少

只有 LOAD/STORE 指令访存

CPU 中有多个通用寄存器

采用流水技术一个时钟周期内完成一条指令

采用组合逻辑实现控制器

采用优化的编译程序

CISC:

系统指令复杂庞大，各种指令使用频度相差大

指令长度不固定、指令格式种类多、寻址方式多

访存指令不受限制

CPU 中设有专用寄存器

大多数指令需要多个时钟周期执行完毕

采用微程序控制器

难以用优化编译生成高效的目的代码

比较：

1.RISC 更能充分利用 VLSI 芯片的面积

2.RISC 更能提高计算机运算速度

指令数、指令格式、寻址方式少，

通用寄存器多，采用组合逻辑，

便于实现指令流水

3.RISC 便于设计，可降低成本，提高可靠性

4.RISC 有利于编译程序代码优化

5.RISC 不易实现指令系统兼容

► 指令流水、指令流水受阻一般有哪些情况，请举例说明。

流水线受阻一般有三种情况：

(1) 结构相关：在指令重叠执行过程中，硬件资源不足，满足不了指令重叠执行的要求，发生了**硬件资源冲突**，引起结构相关。例如，同一时间，几条重叠执行的指令分别要取指令、取操作数、存结果，都需要访存，就会引发访存冲突。

(2) 数据相关：在程序的相邻指令之间出现了某种关联，如当一条指令需要用到前面指

令的执行结果,而这些指令均在流水线中重叠执行,可能**改变读写访问顺序**,引起数据相关。

(3) 控制相关:当流水线遇到**分支指令**,如一条指令要等前一条指令做出转移方向的决定后,才能进入流水线,便发生了控制相关。

► 简述 CPU 的功能,画出其内部组成框图,并说明各部件作用。

## 一、CPU 的功能

### 1. 控制器的功能

取指令

指令控制

分析指令

操作控制

执行指令,发出各种操作命令

时间控制

控制程序输入及结果的输出

处理中断

总线管理

处理异常情况和特殊请求

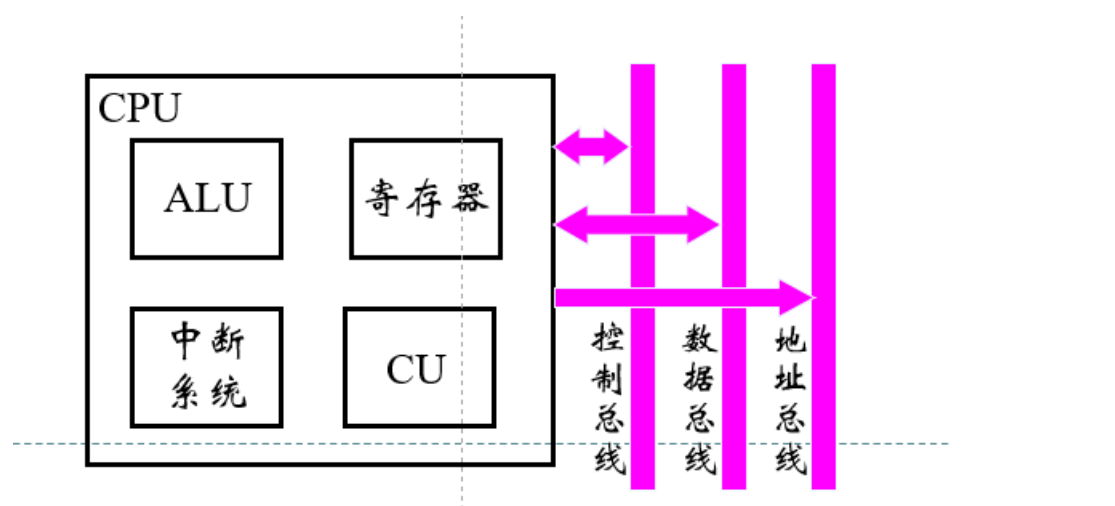
数据加工

### 2. 运算器的功能

实现算术运算和逻辑运算

► 2

龙芯-东北大学(秦皇岛)联合实验室



ALU: 实现算术和逻辑运算

寄存器: 存放操作数

CU: 发出各种操作命令序列的控制部件

中断系统：处理异常情况和特殊请求

- ▶ 一个完整的指令周期包含哪些 CPU 工作周期？中断周期前和中断周期后各是 CPU 的什么工作周期？DMA 周期前和后各是 CPU 的什么周期？

取指周期、间址周期、执行周期、中断周期。

中断周期前是执行周期，中断周期后是取指周期。

DMA 周期前可以是取指周期、执行（存取数）周期或者中断周期，DMA 周期后也可以是取指周期、执行（存取数）周期或者中断周期，总之，DMA 周期前后都是存取周期。

- ▶ 指令周期、时钟周期、机器周期，及其相互关系

指令周期：指取出并执行完一条指令所需的时间。

机器周期：所有指令执行过程中的一个基准时间，以 **访问一次存储器** 的时间为基准，或以完成 **最复杂** 指令功能的时间为准。

时钟周期：时钟周期是控制计算机操作的最小单位时间，用时钟周期控制产生一个或几个微操作命令。

*若指令字长 = 存储字长 取指周期 = 机器周期*

- ▶ 试比较同步控制、异步控制和联合控制的区别

同步控制是指任何一条指令或指令中任何一个微操作的执行都是**事先确定的**，并且都受同一基准时标的**时序信号所控制**的方式。

异步控制无基准时标信号，微操作的时序是由专门的**应答线路**控制，即控制单元发出执行某一微操作的控制信号后，等待执行部件完成了该操作后发回“回答”或“结束”信号，再开始新的微操作。

联合控制是同步控制和异步控制相结合的方式，即大多数操作（如 CPU 内部各操作）在

同步时序信号的控制下进行，少数时间难以确定的微操作（如涉及 I/O 操作）采用异步控制。

- ▶ 解释概念：组合逻辑控制单元和微程序控制单元，机器语言程序和微程序，机器指令和微指令，微指令和毫微指令，微操作命令和微操作，主存储器和控制存储器，MAR 和 CMAR，串行微程序设计和并行微程序设计，水平型微指令和垂直型微指令，静态微程序设计和动态微程序设计。

(1) 机器指令由“0/1”代码组成，能被机器直接识别。机器指令可由**有序微指令组成的微程序来解释**，微程序也由“0/1”代码组成，能被机器直接识别。

(2) 组合逻辑控制单元通过**组合逻辑电路的硬布线**的方式设计的控制单元；微程序控制单元采用微程序设计方法，**使用存储逻辑，在控存中存储微指令**。

(3) 水平型微指令一次能定义并执行多个**并行**操作，分为操作控制字段和下址字段，操作控制字段编码操作控制信号。垂直型微指令采用**类似机器指令**操作码的方式。在微指令中设置**微操作码**字段，由微操作码字段规定为操作等功能。

(4) 主设备是能够获得总线控制权的设备，从设备指被总线主设备访问的设备，只能响应从总线主设备发过来的各种总线命令。

静态：微程序无须改变，采用 **ROM**，动态采用微指令和微程序改变机器指令，有利于仿真，

采用 **EPROM**