

第5章 中央处理器



中华人民共和国成立70周年



大连理工大学 建校70周年

主要内容:

- (一) CPU 的功能和基本结构
- (二) 指令执行过程
- (三) 数据通路的功能和基本结构
- (四) 控制器的功能和工作原理

- 1. 硬布线控制器
- 2. 微程序控制器

微程序、微指令和微命令；微指令的编码方式；微地址的形成方式。

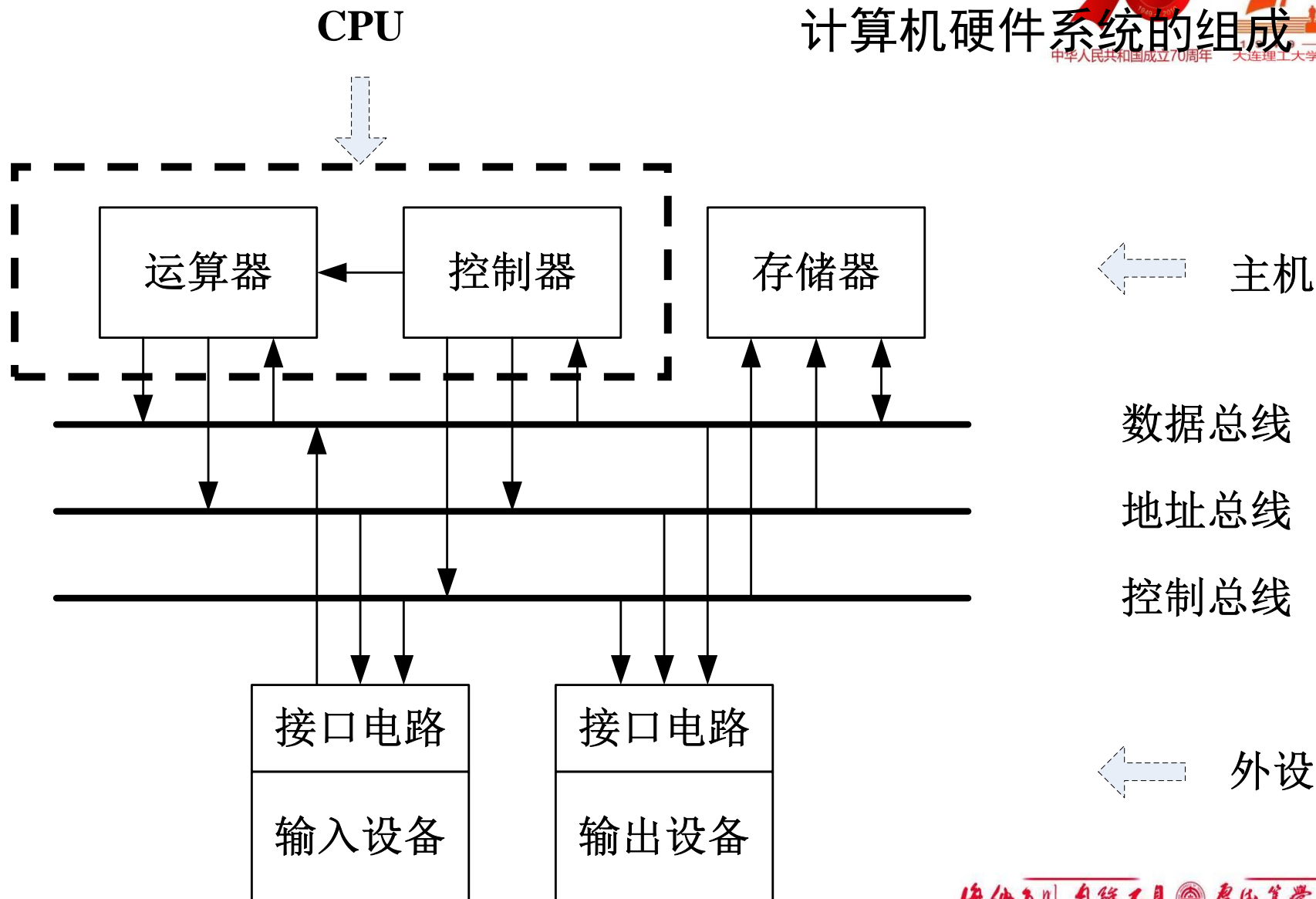
- (五) 指令流水线

- 1. 指令流水线的基本概念
- 2. 超标量和动态流水线的概念

海阔凭鱼跃 天高任鸟飞 自强不息 厚德笃学 知行合一

5.1 CPU的功能和基本结构

- 中央处理器简称CPU，是计算机系统的核心。
- 主要功能：程序的执行(指令控制)，信息的处理(数据加工)，操作控制，时间控制，I/O设备的控制。
- 组成：运算器，控制器，寄存器，时钟电路，(某些CPU中还包括一定容量的ROM、RAM存储器等)。



计算机硬件系统的组成



- ALU： 见2.4节
- 存储器： 见第3章
- 输入输出： 见第7章

控制器的功能



冯·诺依曼结构的计算机

“存储程序”计算机，设置内存，存放程序和数据
在程序运行之前将程序调入内存，然后执行程序

计算机的功能是执行程序

程序是依次排列起来的指令序列

计算机执行程序的基本过程

从程序首地址开始执行第一条指令

分步执行每一条指令，并形成下一条待执行指令地址
自动地连续执行指令，直到程序的最后一条指令

5.1.1 CPU 的功能



1. 控制器的功能

取指令
分析指令
执行指令，发出各种操作命令
控制程序输入及结果的输出
总线管理
处理异常情况和特殊请求

指令控制

操作控制

时间控制

2. 运算器的功能

实现算术运算和逻辑运算

处理异常 (中断)

数据加工

海纳百川 自强不息 厚德笃学 知行合一

控制器的功能

控制器是指指挥与控制整台计算机各功能部件协同工作、自动执行计算机程序的部件。其作用是控制**指令的执行过程**。

能够正确并且自动地连续执行指令

按程序中设定的指令次序执行

正确地分步完成每一条指令规定的功能

读取指令 → 分析指令 → 执行指令

进一步讲，就是向计算机**各功能部件**提供协调运行**每一个步骤**所需要的**控制信号**

海纳百川 自强不息 厚德笃学 知行合一

5.1.2 CPU 的基本结构



0 1 9
交70周年

1. CPU 与系统总线

指令控制

PC IR

操作控制

控制单元CU

时间控制

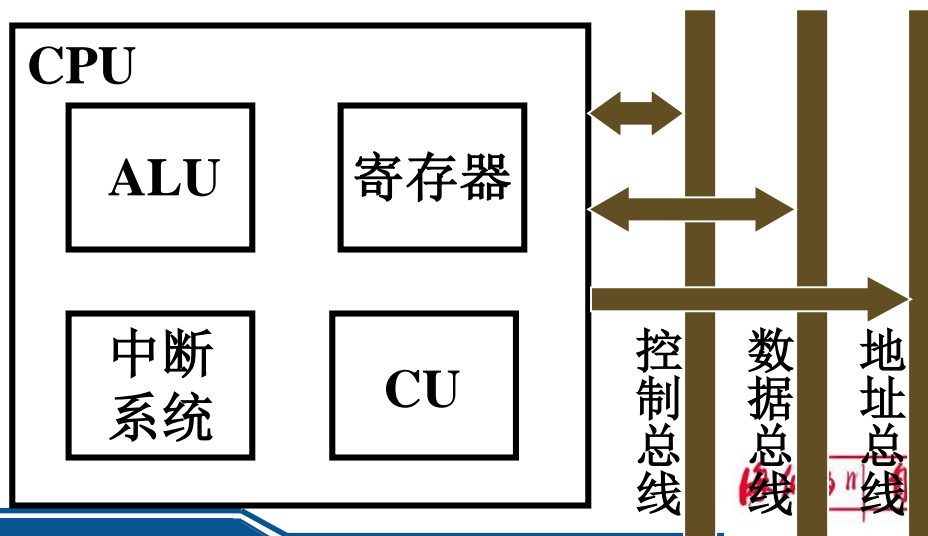
时序电路

数据加工

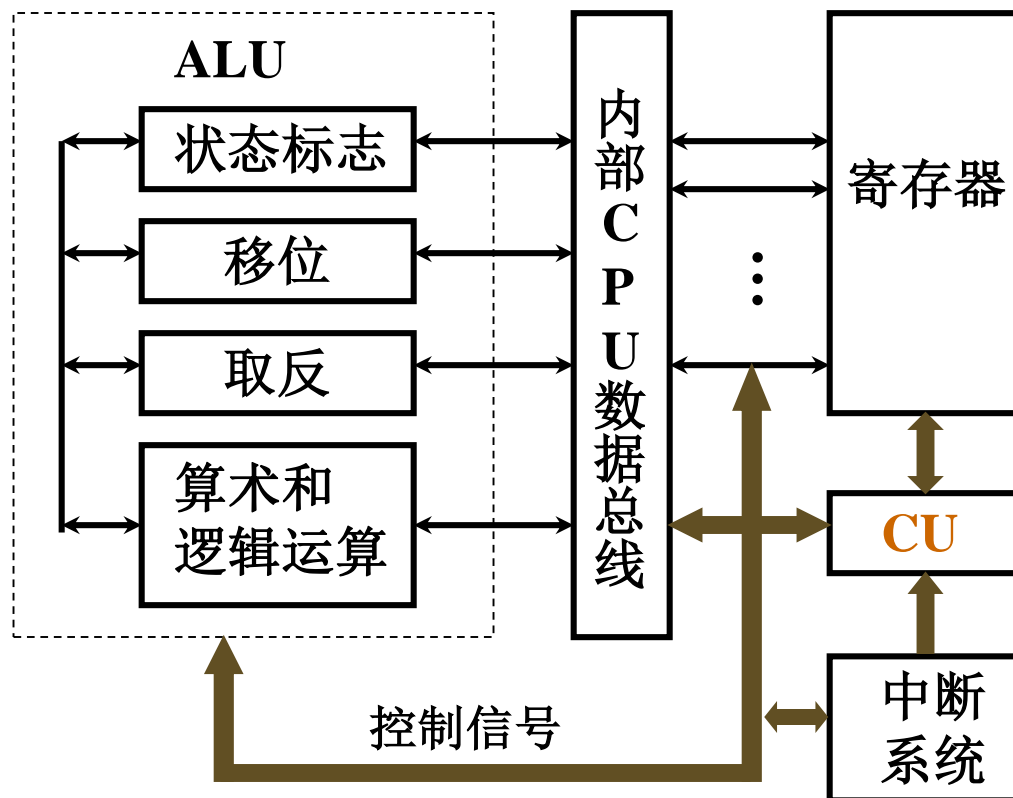
ALU 寄存器

处理中断

中断系统



2. CPU 的内部结构



寄存器 (Register)

- 寄存器R是CPU中的重要组成部分，是CPU内部的临时存储单元。
- 寄存器数量的增加可以提高CPU运行速度。
- 存放内容：数据、地址、控制信息、CPU的工作状态信息。
- 分类：数据寄存器、地址寄存器、标志状态寄存器、控制寄存器和一些其它作用的寄存器。按功能基本可分为三类：通用寄存器组、控制寄存器和主存接口寄存器。。



通用寄存器组

- 通用寄存器组是一组可编程可访问、具有多功能的寄存器统称。对于用户来说是“可见的”。
- 可用于保存操作数、计算结果、或用作地址指针、基址寄存器、变址寄存器和计数器。

控制寄存器

控制寄存器主要包括：

- 程序计数器PC；
- 指令寄存器IR
- 程序状态标志寄存器PSW

附赠：计算机组成原理与系统结构

控制寄存器： 程序计数器（Program Count，简称PC）

- 为了保证程序能自动连续执行，CPU必须具有在当前指令执行结束时能自动确定下一条指令的地址功能。PC正是为此而设置的，也称为指令计数器。
- 在程序开始执行前，将程序的第一条指令的地址送入PC，以便从程序的第一条指令开始执行。执行程序过程中，**CPU将自动修改PC的内容**，使其保持的总是将要执行的下一条指令的地址。
- 当遇到转移指令时，需要改变程序的执行顺序，则一般由转移指令形成转移地址后送往PC，作为下一条指令的地址。因此，**PC应具有计数和接收代码的功能**。

控制寄存器： 指令寄存器 IR

- 指令寄存器用以存放现行指令。
当执行一条指令时，首先从内存将指令取出送到指令寄存器IR，然后送至指令译码器ID，由ID来产生各种微操作命令，控制其他的组成部件工作，从而完成所需的功能。
- 为了提高指令的执行速度，现在大多数计算机都将指令寄存器IR扩充为指令队列或称指令栈，允许预取若干条指令。

控制寄存器：

程序状态标志寄存器 PSW

- 记录算术和逻辑运算指令运行或测试的结果而建立的各种条件码信息，如进位标志、零标志、符号标志、溢出标志等。
- 可被后面的条件转移指令所测试，作为决定程序流向的因素之一。
- 程序状态字反映指令执行结果或控制指令执行形式，

格式如下：

15 12 11 10 9 8 7 6 5 4 3 2 1 0

OF

DF

IF

TF

SF

ZF

AF

PF

CF

状态标志：记录程序运行结果，指令执行结束后自动设置。

控制标志：根据程序需要用指令设置，用于控制指令的执行方式。

程序状态字（续）

- OF：溢出标志，运算过程中如操作数超出了机器能表示的范围，称为溢出。
- SF：符号标志，记录运算结果的符号，结果为负时置1，为正时置0。
- ZF：零标志，运算结果为0时置1，否则为0。
- CF：进位标志，记录运算时从最高有效位产生的进位，有进位时置1，否则为0。

程序状态字（续）

- AF：辅助进位标志，记录运算时第3位产生的进位值。
- PF：奇偶进位标志，当结果操作数中1的个数为偶数时置1，否则为0。
- IF：中断屏蔽位，为1时可屏蔽中断。
- DF：方向标志，在串处理指令中控制处理信息的方向。当DF为1时，每次操作后使变址寄存器SI和DI减小，当DF为0时增加。

海纳百川 自强不息 厚德载物 学无止境

主存接口寄存器

- 主存接口寄存器是与存储器进行数据交换使用寄存器。主要包括：
 - ✓ 主存地址寄存器MAR：用于存放将被访问的存储单元的地址
 - ✓ 主存数据寄存器MDR：用于存放欲存入存储器中的数据或最近从存储器中读出的数据。

控制 CPU 操作

$PC \rightarrow MAR \rightarrow M \rightarrow MDR \rightarrow IR$

其中 **MAR MDR IR**

用户不可见

PC

用户可见

海纳百川 自强不息 厚德笃学 知行合一

附录：8086的寄存器

8086共有14个16位寄存器

| | |
|----|----|
| AH | AL |
| BH | BL |
| CH | CL |
| DH | DL |

AX累加器

BX基址寄存器

CX计数器

DX数据寄存器

| |
|----|
| SI |
| DI |
| BP |
| SP |

源变址寄存器

目的变址寄存器

基址指针

堆栈指针

| |
|-------|
| FLAGS |
| IP |

程序状态字

程序计数器

| |
|----|
| CS |
| SS |
| DS |
| ES |

代码段寄存器

堆栈段寄存器

数据段寄存器

附加段寄存器

海纳百川 自强不息 厚德笃学 知行合一

5.1.3 时序控制方式



- 一条指令的读取与执行过程一般需要分成读取指令、译码、执行、存放结果等几个步骤。这样就需要一种时间划分的信号标志，如周期、节拍等。
- CPU的每次操作都需要有严格的时序控制。
- **时序控制部件**一般由脉冲源、锁相环倍频电路、周期状态触发器、节拍发生器、起停线路等组成。
- **脉冲源**产生固定频率的脉冲可以直接作为机器的主频时钟信号，也可以经由锁相环倍频电路后来产生主频时钟。
- 主频时钟的频率高低与机器的性能和选用的器件有关。一般情况下主频时钟越高，计算机速度越快。

1. 时序相关的基本概念

石英晶体振荡器：产生一定频率的时钟脉冲信号，作为整个机器的时间基准源。也称为主振荡频率。最常用的脉冲源。

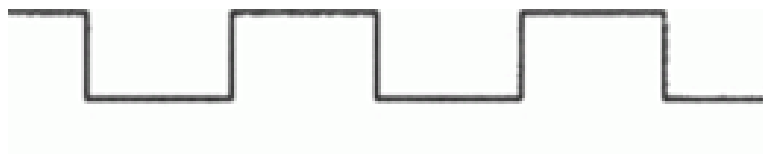


时钟控制电路：为每条指令按时间顺序执行提供基准信号。由时钟脉冲发生器(石英晶体振荡器)和启停控制电路组成。

时序系统



时序系统：在计算机的控制系统中，常将其时序关系划分为**机器周期(CPU周期)**、**时钟周期(节拍)**、**工作脉冲**三个层次，也称为**计算机系统三级时序关系**。



海纳百川 自强不息 厚德笃学 知行合一

计算机系统三级时序



➤ 机器周期

- ◆ 在CPU执行程序的过程中，将指令周期划分成几个时间段，每个阶段称为一个机器周期，也称为CPU周期或工作周期。

- ◆ ep:

如一条指令可分为取指令、取源操作数、取目的操作数、执行处理等阶段。对应取指周期、取操作数周期、执行周期、中断周期等。

- ◆ 每个机器周期通常又包括几个时钟周期。

海纳百川 自强不息 厚德笃学 知行合一

计算机系统三级时序



➤ 时钟周期

- 在机器周期的工作往往还需要细分成几个步骤并按一定顺序来完成。为此，需要将一个机器周期又分为若干个相等的时间段。
- 每一个时间段内完成一步操作，这个时间段被称为时钟周期（节拍）。
例如，按变址方式读取操作数时，则先要进行变址计算，然后才能访存取数。
- 时钟周期是时序系统中**最基本的时间分段**。每个时钟周期完成一步操作，如完成一次传送、或加减等操作。

海纳百川 自强不息 厚德笃学 知行合一

计算机系统三级时序



➤ 工作脉冲

- 在一个时钟周期（节拍）内可设置几个工作脉冲，用于寄存器的清除、接收数据等工作。
- 具体机器设置的工作脉冲数量也是不同的。

指令周期

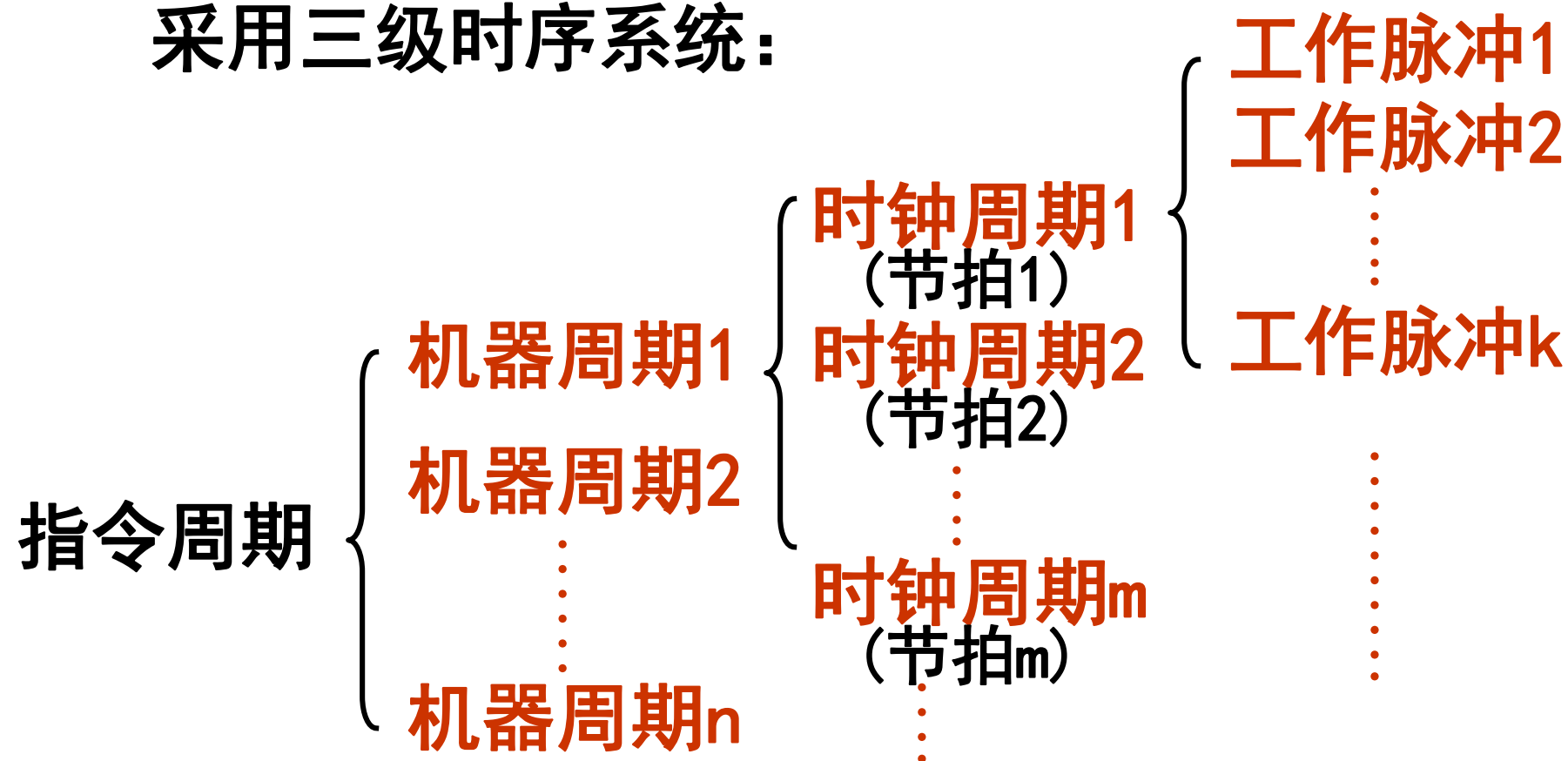
- **指令周期**是指计算机从读取指令、分析指令到执行完该指令所需要的全部时间。
- 各种指令的功能不同，不同指令的指令周期是不相同的。在计算机的时序系统中**没有指令周期的时间标志信号**。
- 因此也**不将**指令周期列为具体时序系统的划分的三个等级之中。

指令周期 > 机器周期 > 时钟周期 > 工作脉冲信号

海纳百川 自强不息 厚德笃学 知行合一

2. 组合逻辑控制器的时序划分

采用三级时序系统：



● 时序关系：

晶振输出

工作脉冲P——对微操作定时

时钟 控制分步操作时间

时钟T1 取出指令

时钟T2

修改PC

机器周期1——控制不同阶段操作时间

取指

机器周期2

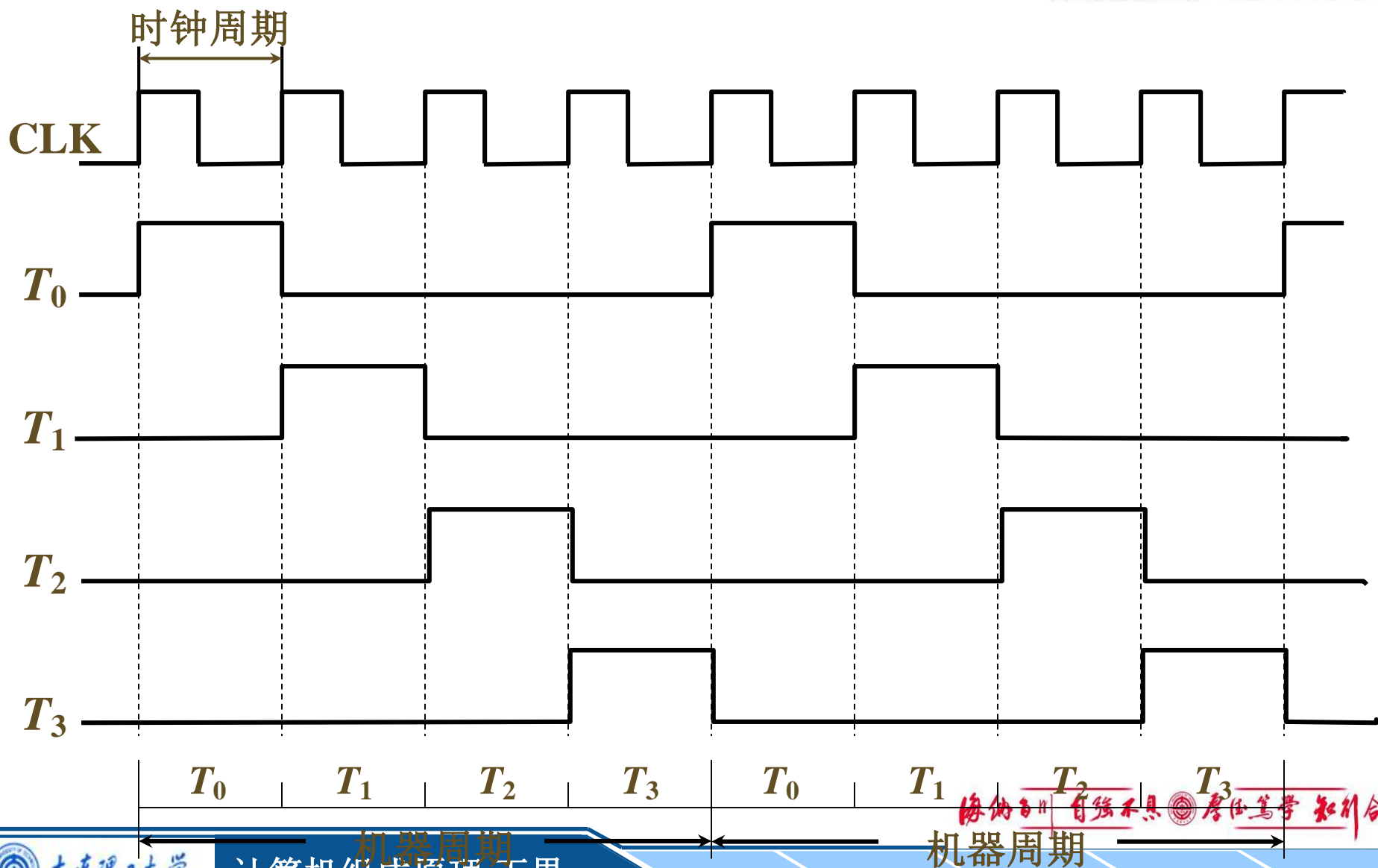
取数

机器周期3

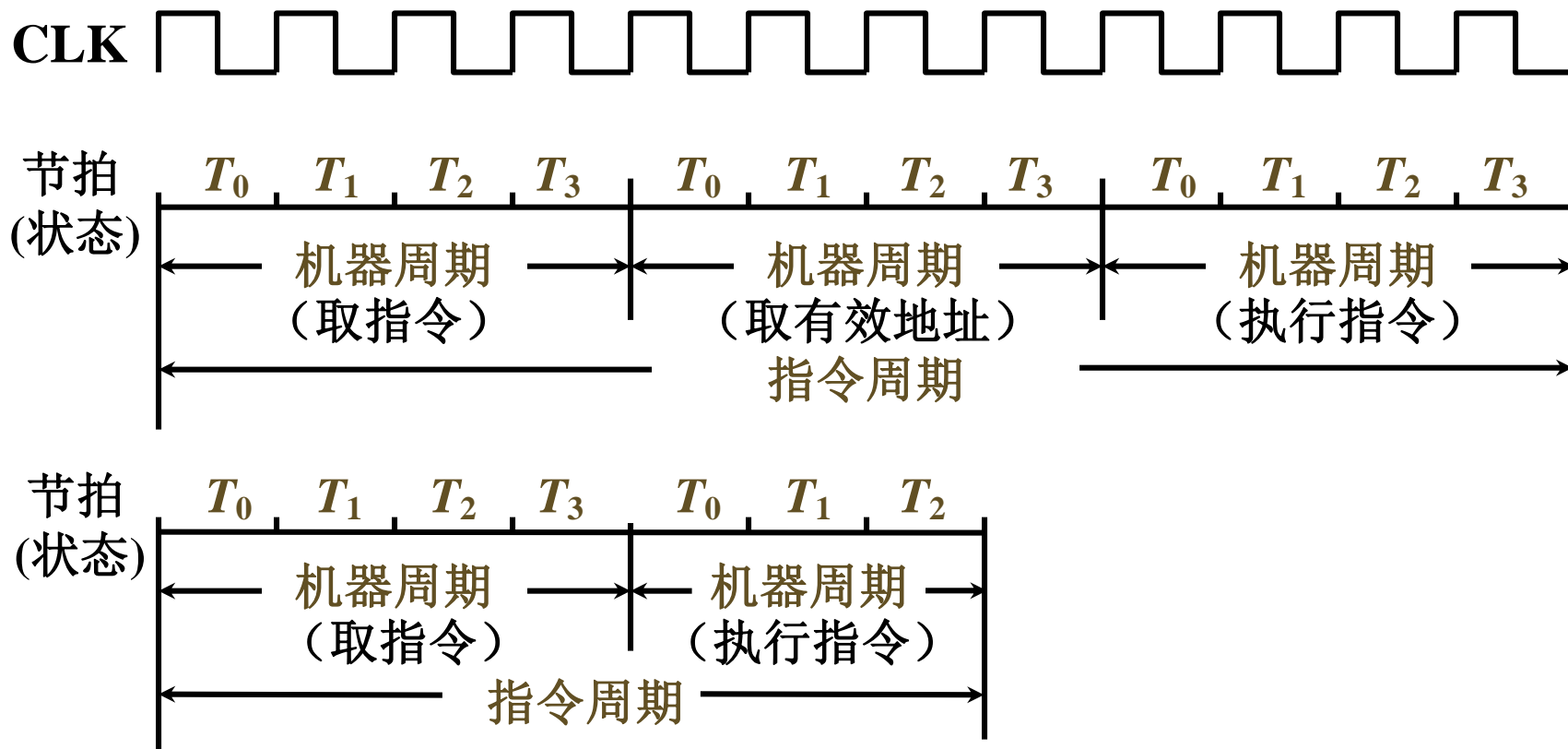
执行

指令周期

时钟周期（节拍、状态）



多级时序系统





2. 时序控制方式

- 执行一条指令的过程是通过执行一个确定的具体操作序列来实现的。(同时执行 or 顺序执行)
- 形成控制不同操作序列的时序控制信号有多种方法，称为控制器的控制方式。常用的有**同步控制、异步控制和联合控制等方式。**

海纳百川 自强不息 厚德笃学 知行合一

(1) 同步控制方式

- ① **定义**：指任何指令的执行或指令中每个操作的执行都要受到事先确定的时序信号控制，这样每个时序信号的结束就意味着一个操作已经完成。指令在执行时所需的机器周期和时钟周期都是固定不变的。
- 不同操作的执行时间可能不同，通常需要选择最长操作的执行时间作为计算标准。

同步控制方式

中华人民共和国成立70周年 大连理工大学 建校70周年

② **特点**：有明显时序时间划分，时钟周期时间固定，各步操作的衔接、各部件之间的数据传送受严格同步定时控制。

③ **优点**：时序关系简单，时序划分规整，控制不复杂；控制逻辑易于集中，便于管理。

缺点：执行时间短的操作需要等待，时间安排不合理。

④ **应用场合**：用于CPU内部、设备内部、系统总线操作（各挂接部件速度相近，传送时间确定，传送距离较近）。

适合于硬布线逻辑控制器和微程序控制器。

(2) 异步控制方式

① **定义**：是指各项具体操作按其需要来选择不同的时间，不受统一的时钟周期(节拍)的约束。

② **特点**：无统一时钟周期划分，各操作间的衔接和各部件之间的信息交换采用异步**应答方式**。

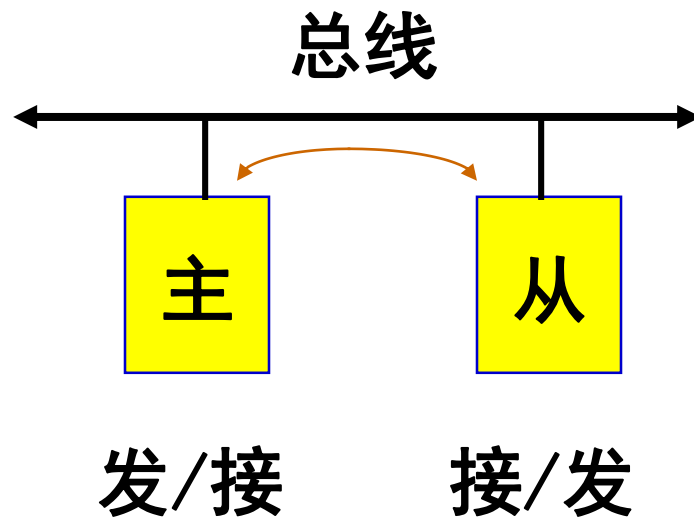
例：异步传送操作

主设备：

申请并掌握总线权的设备。

从设备：

响应主设备请求的设备。



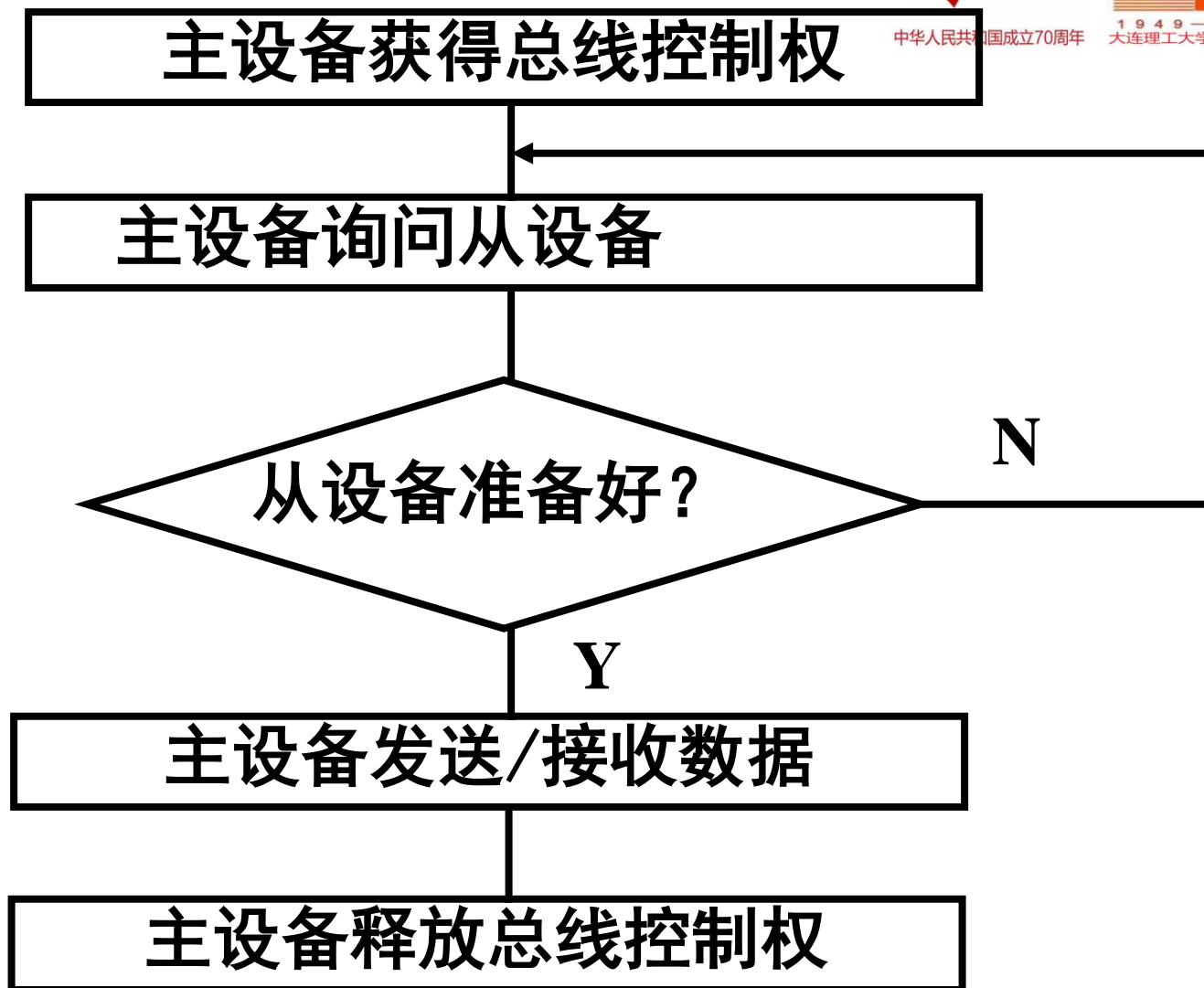
海纳百川 自强不息 厚德载物 知行合一

操作流程:

主设备输出端与总线连接

中华人民共和国成立70周年

1949—2019
大连理工大学 建校70周年



主设备输出端与总线断开

(2) 异步控制方式

③ **优缺点：** 时间安排紧凑、合理；控制复杂。

④ **应用场合：** 用于异步总线操作（各挂接部件速度差异大，传送时间不确定，传送距离较远）。

(3) 联合控制方式

以固定时钟周期作为时序基础，引入应答思想。

1949—2019
大连理工大学 建校70周年

- 同步控制和异步控制相结合的方式。对不同指令的各个微操作实行大部分统一，小部分区别对待的方式。
- 即大部分微操作安排在一个固定机器周期中，并在同步时序信号控制下进行；而对那些时间难以确定的微操作则以执行部件送回的“回答”信号作为本次微操作的结束。
- 在CPU内部或设备内部大部分都采用同步控制，而对系统总线所连接的各种设备，由于其工作速度差异较大，通常采用异步方式较为合适。

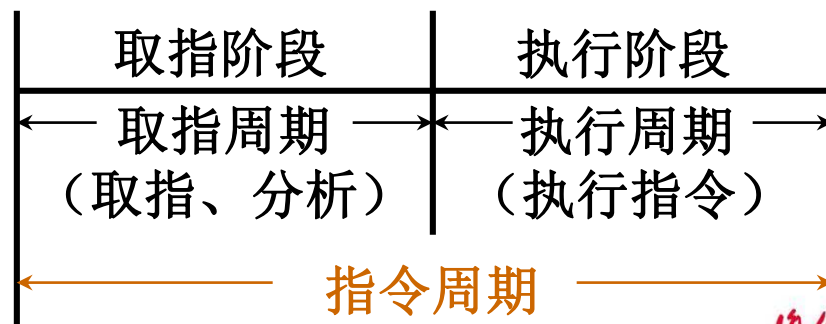
5.2 指令周期

5.2.1 指令周期的基本概念

1. 指令周期

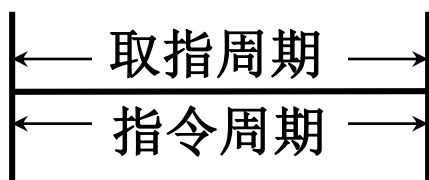
取出并执行一条指令所需的全部时间

完成一条指令 { 取指、分析 取指周期
 执行 执行周期

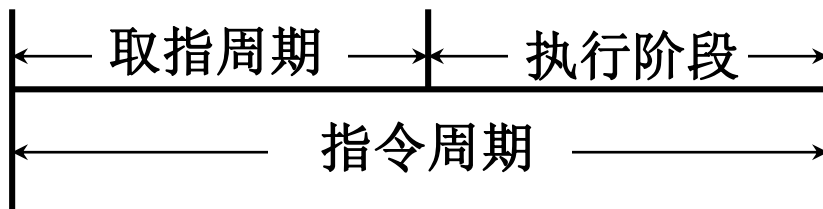


海纳百川 自强不息 厚德笃学 知行合一

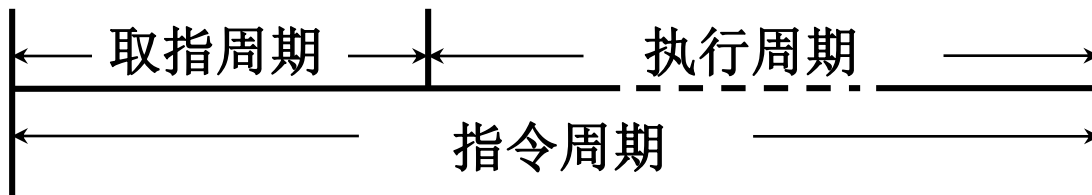
2. 每条指令的指令周期不同



NOP



ADD mem

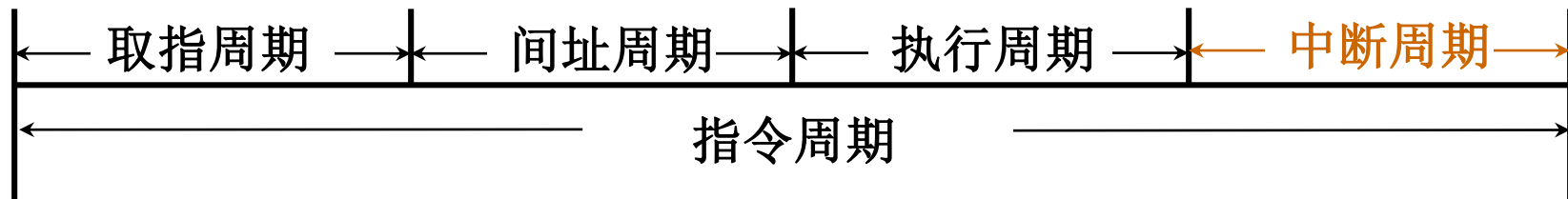


MUL mem

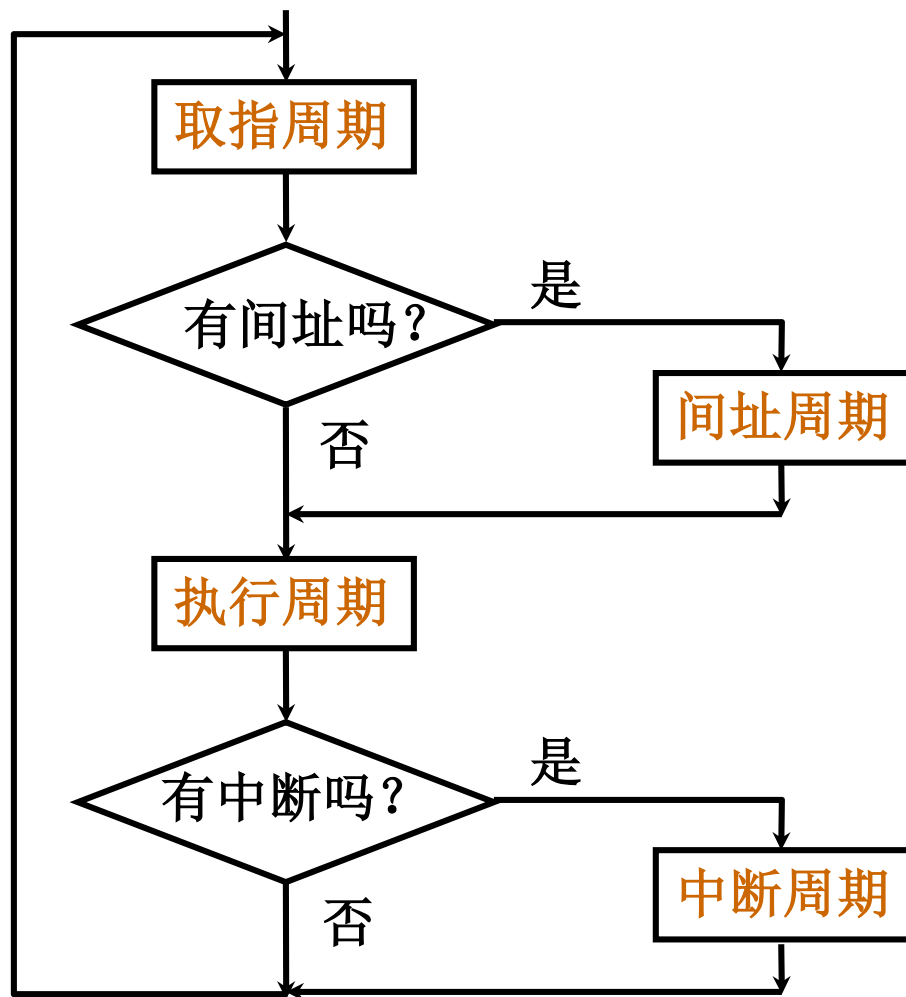
3. 具有间接寻址的指令周期



4. 带有中断周期的指令周期



5. 指令周期流程



6. CPU 工作周期的标志

CPU 访存有 4 种性质

取 指令

取指周期

取 地址

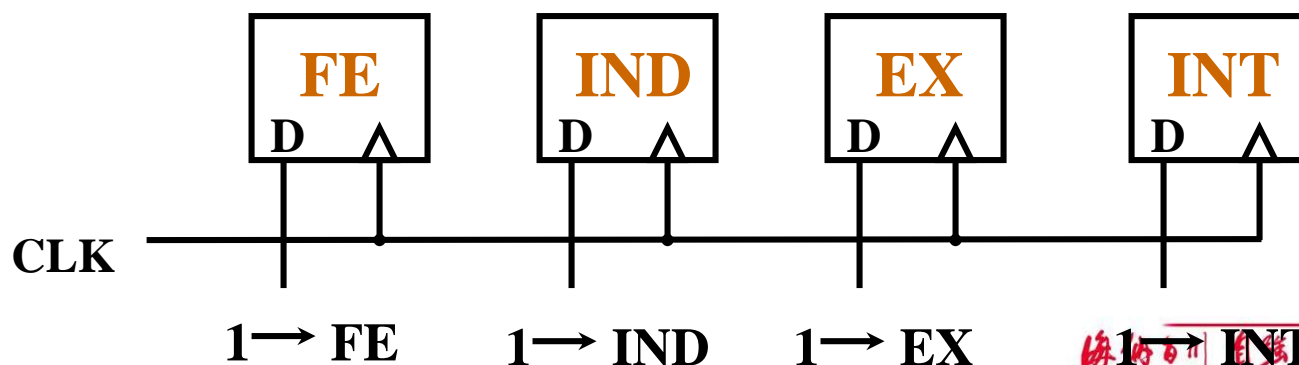
间址周期

取 操作数

执行周期（当指令为访存指令）

存 程序断点

中断周期



5.2.2 具体执行过程

1. 指令执行中的数据传输通道

为了便于分析指令周期中的数据流，设定CPU中包含有：

存储器地址寄存器MAR

存储器数据寄存器MDR

程序计数器PC

指令寄存器IR。



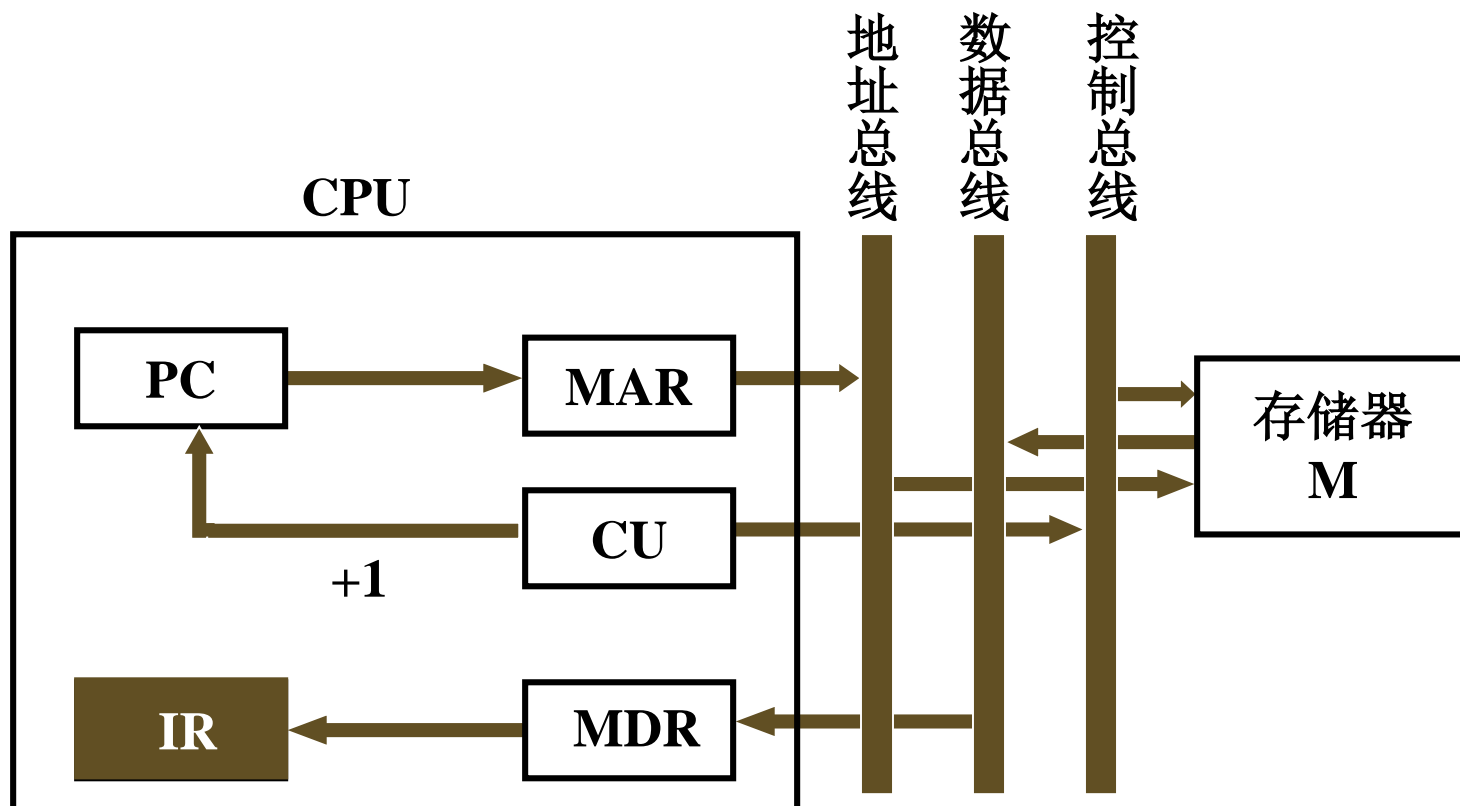
中华人民共和国成立70周年



1949—2019
大连理工大学 建校70周年

5.2.2 具体执行过程

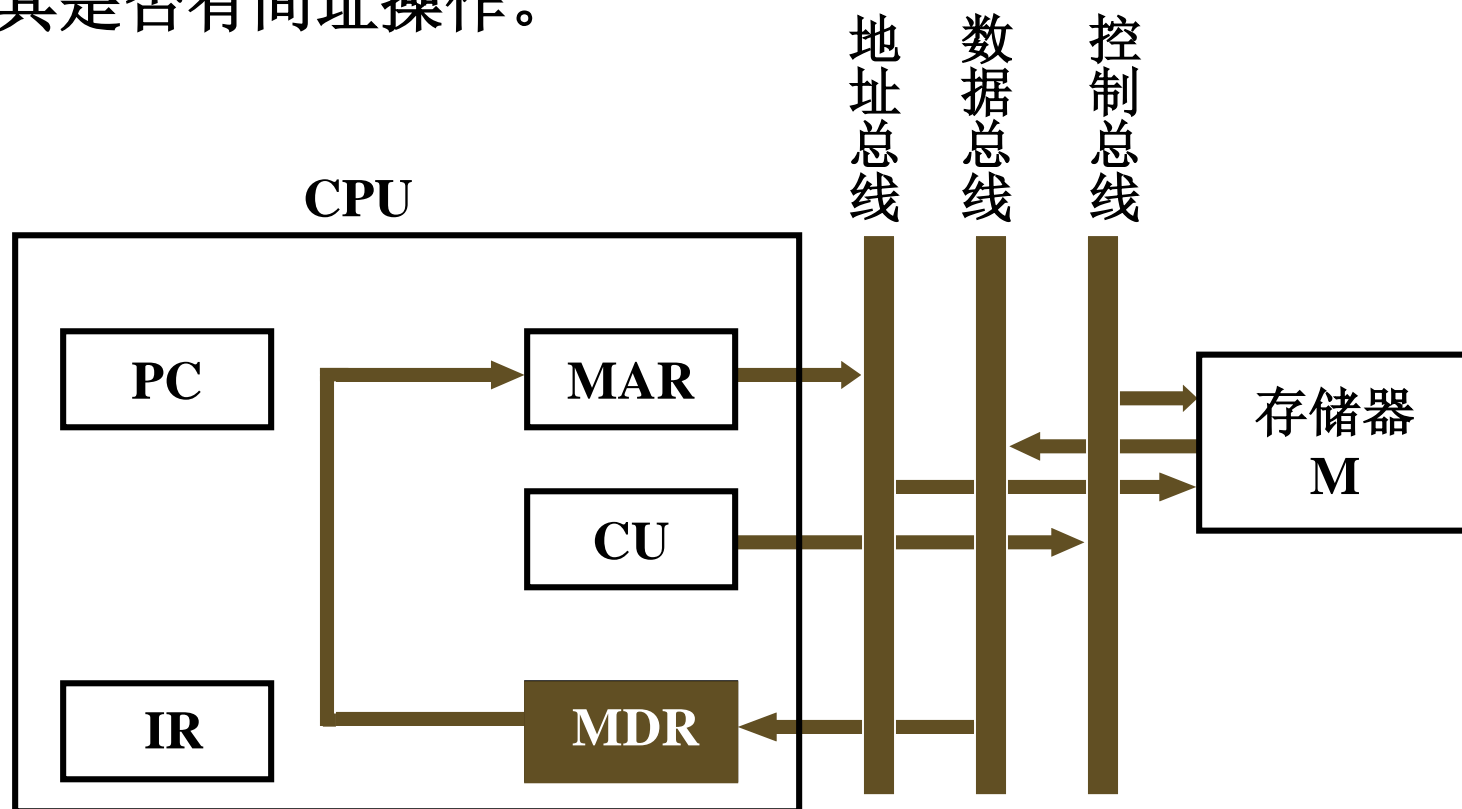
(1) 取指周期数据流



PC → MAR → BUS → M → BUS(M) → MDR → IR → PC + 1

(2) 间址周期数据流

一旦取指周期结束，执行单元CU便检查IR中的内容，以确定其是否有间址操作。



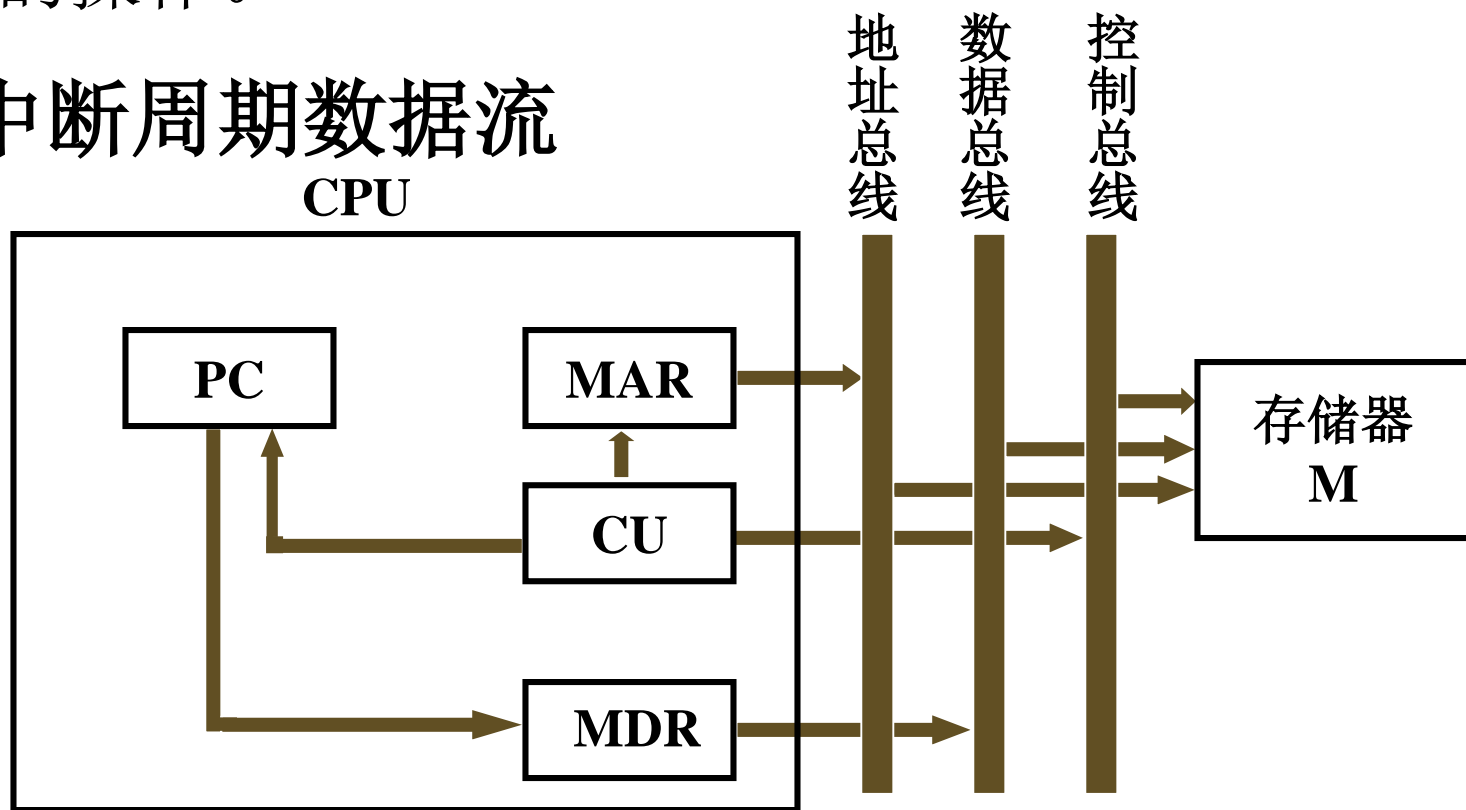
$Ad(MDR) \rightarrow MAR \rightarrow BUS(M) \rightarrow MDR$

海纳百川 自强不息 厚德载物 知行合一

(3) 执行周期数据流

不同指令的执行周期数据流不同。可能涉及到CPU内部寄存器间的数据传送，或对存储器(或I/O)进行读 / 写操作，或对ALU的操作。

(4) 中断周期数据流



2 指令执行过程

● 计算机启动过程:

(1) 在机器内部设置有存放固定程序的只读存储器(ROM), 利用上电时硬件产生的复位(Reset)信号使计算机处于初始状态。

(2) 这是通过将固定程序的入口地址(即开机后执行的第1条指令的地址)装入程序计数器PC来实现的, 也可以直接在指令寄存器IR中置入一条无条件转移指令, 转移到固定程序入口, 然后开始执行程序。

(3) 从程序入口地址开始执行该程序的指令序列, 每执行一条指令后, 即将准备好下条指令的地址并送入PC中。

2 指令执行过程

● 机器指令的执行

(1) 一条最简单的机器指令中，至少需要取指令机器周期和执行机器指令周期这两个机器周期。

(2) 由控制器产生一系列微操作控制信号的过程。将这些微操作控制信号合理地分配在各个机器周期的各个节拍中，便可构成各条机器指令的操作流程图。

指令执行过程举例

(1) 一条加法指令的执行过程举例

加法指令: **ADD rd,rs,[rs1+disp]**

rs, rd, rs1为通用寄存器地址（编号）；imm（disp）为立即数或位移量。

假定运算器由一组通用寄存器GR和一个ALU组成，并有4个记忆运算结果的状态标志N、Z、V、C（触发器）。

N（负数），运算结果为负数时，置1，否则为0。

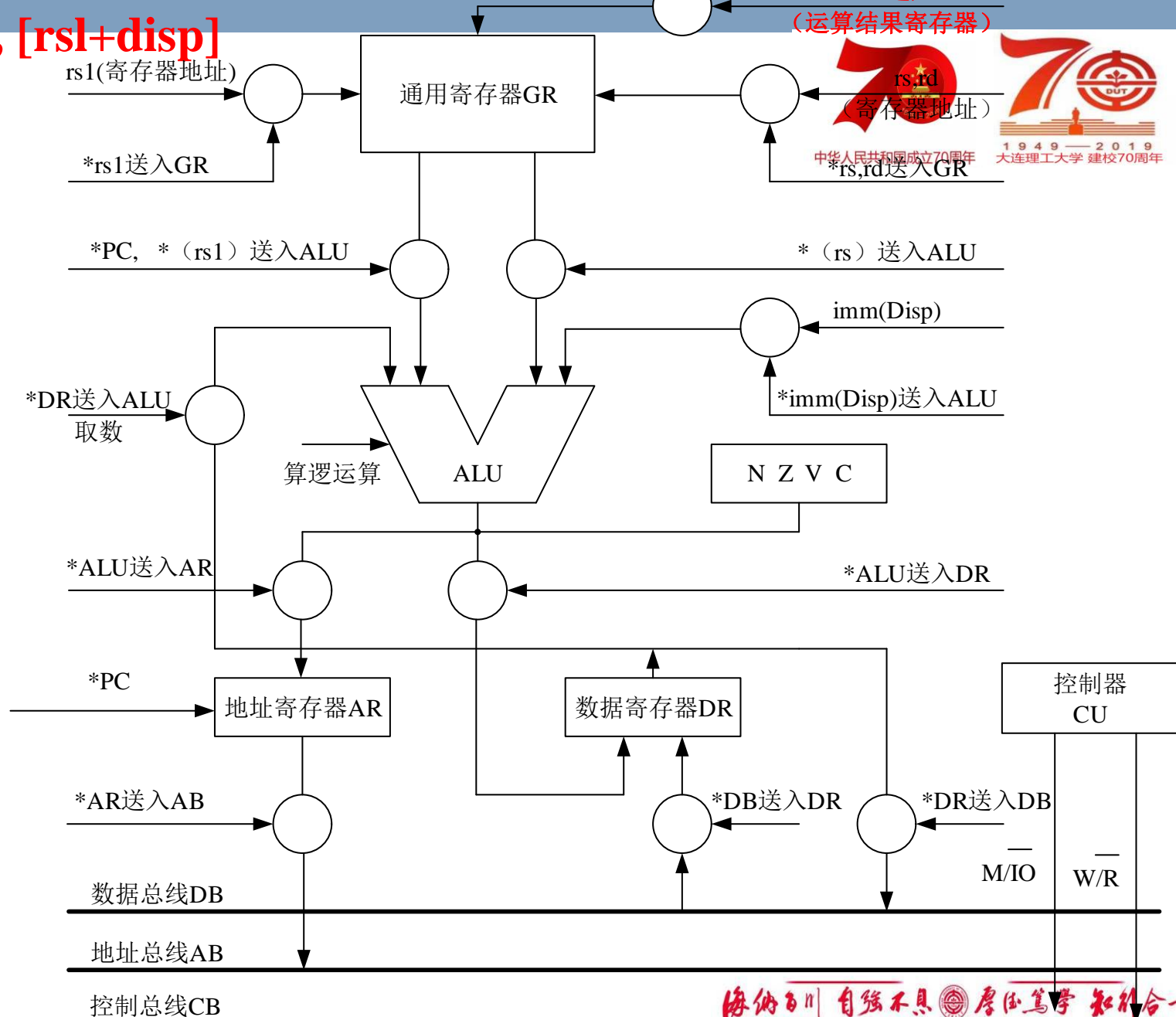
Z（零），运算结果为零，Z=1，否则为0。

V（溢出），运算结果溢出，V=1，否则为0。

C（进位），运算结果进位或借位是，C=1，否则为0

ADD rd,rs,[rs1+disp]

(运算结果寄存器)



海纳百川 自强不息 厚德笃学 知行合一

- 该加法指令完成的操作, 需要4个机器周期。



① 从存储器取指令, 送入指令寄存器, 并进行操作码译码。程序计数器加1, 为取下一条指令作准备。

控制器发出的控制信号: $W/\overline{R}=0$, $M/\overline{IO}=1$;

数据通路: $PC \rightarrow AR \rightarrow AB$; $DB \rightarrow DR \rightarrow IR$; $PC+1$

② 计算数据地址, 将计算得到的有效地址送地址寄存器AR。

控制器发出的控制信号: “+” (加法命令送ALU) ;

数据通路:

$rs1 \rightarrow GR \rightarrow ALU$, $disp \rightarrow ALU$;

$ALU \rightarrow AR$ (有效地址送地址寄存器)。

海纳百川 自强不息 厚德笃学 知行合一

③ 到存储器取数。

控制器发出控制信号： $W/\overline{R}=0, M/\overline{IO}=1$ ；

数据通路： $AR \rightarrow AB, DB \rightarrow DR$

即将地址寄存器内容送地址总线，同时发访存读命令，存储器读出数据送数据总线后，打入数据寄存器。

④ 进行加法运算，结果送目的寄存器，并根据运算结果置状态位N, Z, V, C。

控制器送出的控制信号：“+”

数据通路：

$rs \rightarrow GR \rightarrow ALU, DR \rightarrow ALU; ALU \rightarrow rd$ 。

其中rs表示源操作数地址，rd表示目的操作数地址。

海纳百川 自强不息 厚德笃学 知行合一

(2) 条件转移指令的执行过程举例

- 指令功能根据N, Z, V, C的状态, 决定是否转移。如转移条件成立则转移到本条指令所指定的地址, 否则顺序执行下一条指令。
- 指令完成以下操作:
 - ① **从存储器取指令**, 送入指令寄存器并进行操作码译码。
程序计数器加1, 如不转移, 即为下一条要执行的指令地址。
本操作对所有指令都是相同的。
 - ② **如转移条件成立**, 根据指令规定的寻址方式计算有效地址, 转移指令经常采用相对寻址方式。此时转移地址=PC+disp。
此处PC通常是指本条指令的下一条指令的地址 (在上一机器周期已执行PC+1操作)。

(2) 条件转移指令的执行过程举例

- 该条件转移指令只需要两个机器周期：
 - 取值周期，计算周期
- 如转移条件成立，在第2个机器周期增加一个 $ALU \rightarrow PC$ 信号；另外如为相对转移，则用 $PC \rightarrow ALU$ 信号取代加法指令第2周期中的 $(rs1) \rightarrow ALU$ 信号，其余操作与加法指令的操作相同。

CPU 与系统总线

指令控制

PC IR

操作控制

控制单元CU

时间控制

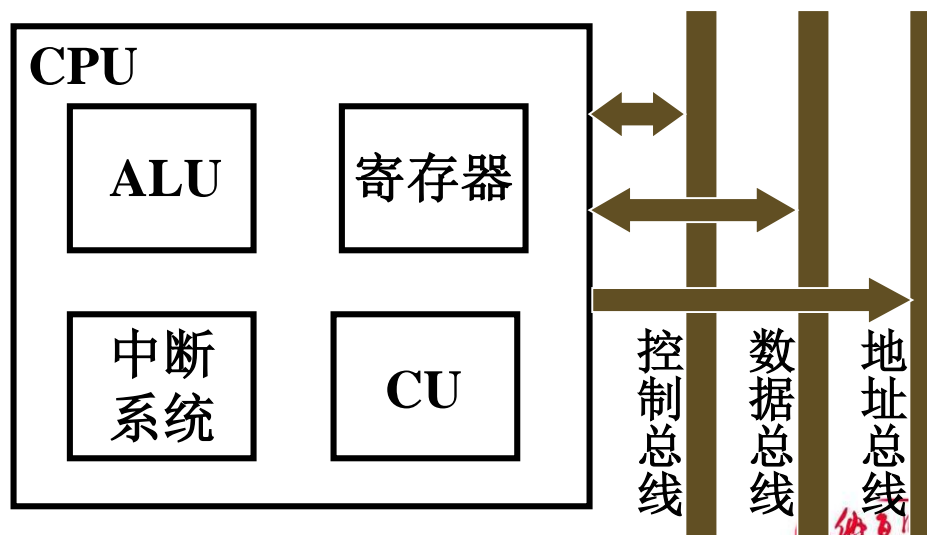
时序电路

数据加工

ALU 寄存器

处理中断

中断系统



5.3 控制器的组成和工作原理



控制器设计需要解决的问题：

- 为什么需要控制信号？
- 需要什么样的控制信号？
- 如何产生控制信号？

海纳百川 自强不息 厚德笃学 知行合一

控制器的主要组成

1949—2019

① 程序计数器 PC

存放指令地址，有 增量 或 接收新值功能

② 指令寄存器 IR

存放指令内容：操作码与操作数地址

③ 指令译码器ID（或操作码译码器）

对指令寄存器中的**操作码**进行分析解释，产生相应的控制信号。

④ 时序控制部件

用来产生一系列的时序信号，用以保证各个微操作的执行顺序。

⑤微操作控制信号形成部件

微操作控制信号形成部件，又称为控制单元(CU)。它根据指令译码器产生的操作控制信号、时序控制部件产生的时序信号以及其他控制条件，产生整个机器所需要的全部微操作控制信号。

微操作：

即计算机中最简单（不能再分解）的操作。

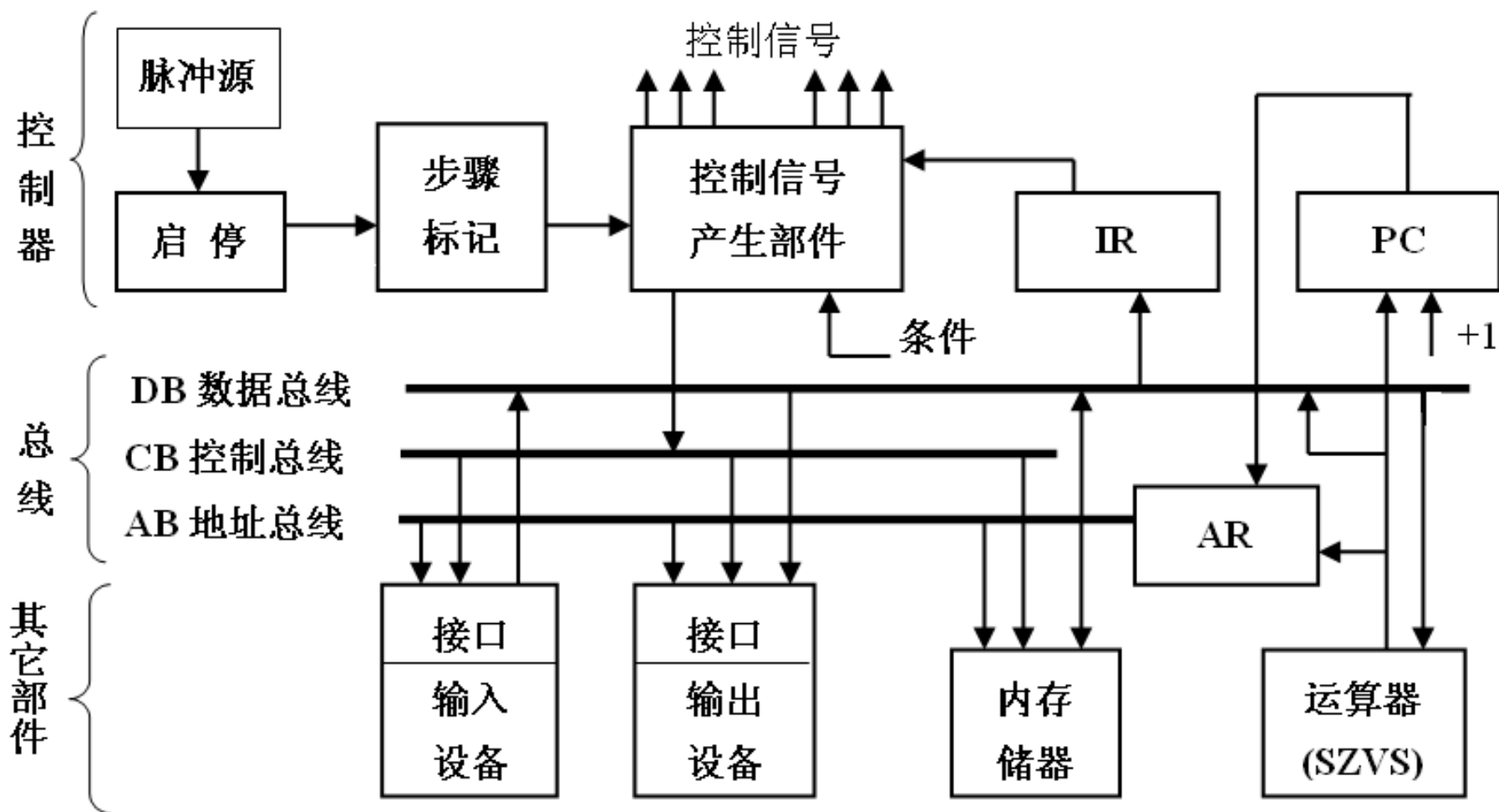
如打开某一个控制门，寄存器的清除脉冲等。

复杂操作是通过执行一系列微操作来实现的。

控制器组成和在整机中的地位



在计算机硬件系统中，控制器处于指挥控制地位



两种不同类型的控制器

根据时序控制部件和微操作控制信号形成部件不同的组成和不同的运行原理，有两种不同类型的控制器：

硬连线控制器(组合逻辑控制器)：

用组合逻辑电路实现各种控制功能，又称为组合逻辑型控制器。采用组合逻辑线路、依据指令及其执行步骤直接产生控制信号。

微程序控制器：

用一个ROM做为控制信号产生的载体，ROM中存储着一系列的微程序，组成微程序的微指令代码产生相应的操作控制信号，这是一种存储逻辑型的控制器



周年 1949—2019
大连理工大学 建校70周年

组合逻辑设计

硬连线逻辑

微程序设计

存储逻辑

两类控制器中的PC和IR是相同的，但确定和表示指令执行步骤的办法以及给出控制各部件运行所需要的控制信号的方案是不同的。

海纳百川 自强不息 厚德笃学 知行合一

5.3.2 微程序控制器



1.几个概念

➤**微操作**：在计算机中，一条机器指令的功能是通过按一定次序执行一系列基本操作完成的，这些最基本的控制命令称为微操作。

➤**微指令**：在微程序控制的计算机中，将由同时发出的控制信号所执行的一组微操作称为微指令。组成微指令的微操作，又称**微命令**。

➤一条机器指令分成若干条微指令，按次序执行这些微指令，就可以实现指令的功能。

➤ **微程序**：计算机每条指令的功能均由微指令序列解释完成，这些微指令序列的集合就叫做微程序。

➤ **控制存储器**：微程序事先存放在专用的存储器中。由于该存储器主要存放控制命令（信号）与下一条执行的微指令地址（简称为**下址**），所以被叫做控制存储器。

- 一般计算机指令系统是固定的，所以实现指令系统的微程序也是固定的，控制存储器可用只读存储器实现。
- 机器内控制信号数量多，再加上决定下址的地址码有一定宽度，所以控制存储器的字长比机器字长要长得多。
- 控制存储器的容量取决于实现指令系统所需要的微程序长度。

2. 微程序控制的基本思想

中华人民共和国成立70周年 大连理工大学 建校70周年

➤ 若干微命令编制成**一条微指令**，控制实现**一步操作**。

➤ 若干微指令组成**一段微程序**，解释执行**一条机器指令**。

➤ 微程序事先存放在控制存储器中，执行机器指令时再取出。执行一条指令实际上就是执行一段存放在控制存储器中的微程序。

CPU的构成 { 引入了程序技术，使设计规整；
引入了存储逻辑，使功能易于扩展。

海纳百川 自强不息 厚德笃学 知行合一

3、微指令的基本结构

- 微指令基本结构包括微命令字段（操作控制字段）和顺序控制字段组成。
- **微命令字段**（也称操作控制码）是为一次微操作所需要的全部控制信号的编码，用以发出管理和指挥整个机器工作的控制信号。
- **后继微地址控制字段**（也称下地址字段）用以决定产生下一条微指令的地址。该字段通常包含转移控制字段（转移条件）和转移地址字段（指定下一条微指令的地址）两部分。

微命令字段—编码方式



- 直接表示法
- 编码表示法

(1) 直接表示法（直接控制法）

微命令按位给出：在微指令的控制字段中，每一位代表一个微命令，在设计微指令时，是否发出某个微命令，只要将控制字段中相应位置成“1”或“0”，这样就可打开或关闭某个控制门。

- 操作码的位数代表支持的微命令数量。
- 无需译码，产生微命令的速度快。
- 微命令太多，信息的表示效率低。
- 微命令中通常只有个别位采用直接控制法。

(2) 编码表示法（字段编译法）

1949—2019
中华人民共和国成立70周年 大连理工大学建校70周年

- 将操作码分成若干字段，将一组互斥的微命令分在同一字段中进行编码形成较短的代码。
- 互斥：一个微周期内，不能同时出现。比如寄存器的读和写操作。
- 微周期：执行一条微指令所需的执行时间。
- 选出互斥的微命令，将它们编成一组，作为微指令字的一个字段，用二进制编码。

同类操作中互斥的微命令放同一字段。

微命令分组原则：需要并行发出的微命令可分在不同的字段中。

海纳百川 自强不息 厚德笃学 知行合一

(3) 混合表示法

- 把直接表示法与编码表示法结合使用，采用部分直接表示、部分编码表示的方法。
- 具体做法是将一些要求速度高的，或者一些相容的控制信号用直接表示法，将其他相斥信号用编码表示法。
- 直接表示法、编码表示法和混合表示法的三种编码方式的微指令都属于水平型微指令。

后继微地址控制字段

- 计数器方式（又称增量方式、顺序执行方式）
- 断定方式
 - 需要增加一个微程序计数器uPC，通过uPC加1来产生下一条微指令的微地址。

(1) 计数器方法（增量方式）

以顺序执行为主，
辅以各种常规转移方式。
增加一个微程序计数器（uPC）

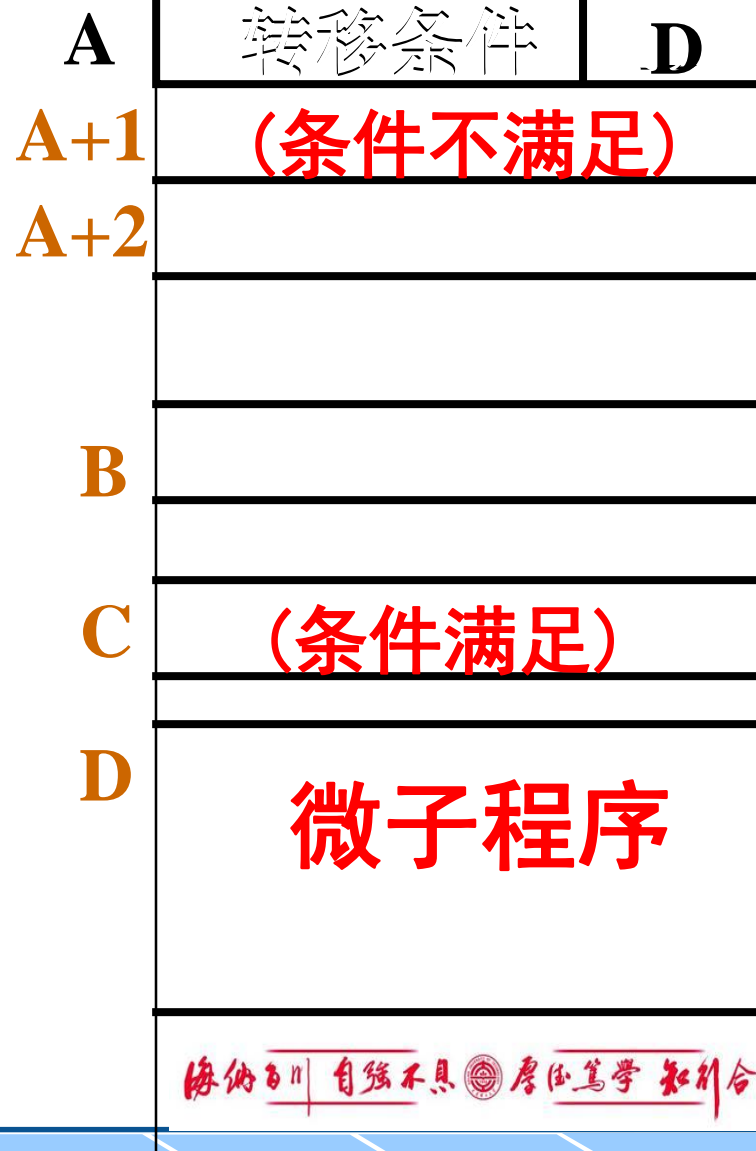
顺序：uPC+1。

跳步：uPC+2。

无条件转移：现行微指令
给出转移微地址。

条件转移：现行微指令给
出转移微地址和转移条件。

转微子程序：现行微指令
给出微子程序入口。



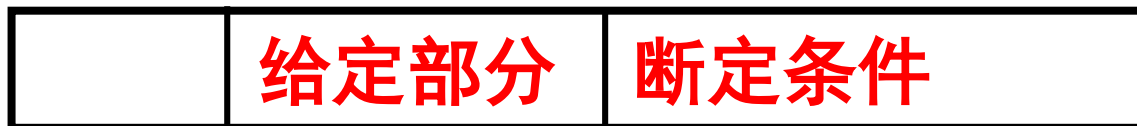
(2) 断定方式

在微指令中增加一个下地址字段，用来指明下一条要执行的微指令地址。

该字段通常包含转移控制字段（转移条件）和转移地址字段（指定下一条微指令的地址）两部分。

通过微指令顺序控制字段由设计者指定或指定的判别字段控制产生后继微指令地址——由直接给定和测试断定相结合。

微指令



海纳百川 自强不息 厚德笃学 知行合一

4、微指令编码格式

- 微指令的格式大体上可分成两类：

| |
|--------|
| 水平型微指令 |
| 垂直型微指令 |

- 微指令的编译法是决定微指令格式的主要因素。

(1) 垂直型微指令

| | | |
|----------------|-------|-------|
| μOP | R_d | R_s |
|----------------|-------|-------|

微操作码 目的地址 源地址

垂直型微指令位数少（如几十位），每条微指令中只有1~2个微操作命令（控制1~2种微操作）。

一条微指令定义并执行一种基本操作。

优点：微指令短、简单、规整，便于编写微程序。

缺点：微程序长，执行速度慢；**不强调实现微指令的并行控制功能**，工作效率低。

4、微指令编码格式

中华人民共和国成立70周年 1949—2019
大连理工大学 建校70周年

(2) 水平型微指令

| | | | | | | | |
|-------|-------|-------|-------|-------|-----------|-------|-----|
| T_1 | T_2 | T_3 | T_4 | | T_{n-1} | T_n | SCF |
|-------|-------|-------|-------|-------|-----------|-------|-----|

操作控制字段 DOCF

后继地址字段 SCF

指一次能定义并执行多个并行操作控制信号的微指令。水平型微指令的位数多(可达百位以上)，它所追求的是对各部件并行控制的能力和更快的执行速度。**一条微指令定义并执行几种并行的基本操作。**

相关编译法：直接控制法、字段编译法是水平型微指令。

优点：微程序短，执行速度快。

缺点：微指令长，编写微程序较麻烦。

海纳百川 自强不息 厚德笃学 知行合一



中华人民共和国成立70周年



1949—2019
大连理工大学 建校70周年

水平型微指令与垂直型微指令的比较

| 水平型微指令 | 垂直型微指令 |
|------------|----------|
| 并行能力强，效率高 | 并行能力差 |
| 灵活性差 | 灵活性强 |
| 执行一条指令的时间短 | 微指令执行时间长 |
| 微指令字比较长 | 微指令字比较短 |
| 微程序短 | 微程序长 |
| 难以掌握 | 容易掌握 |

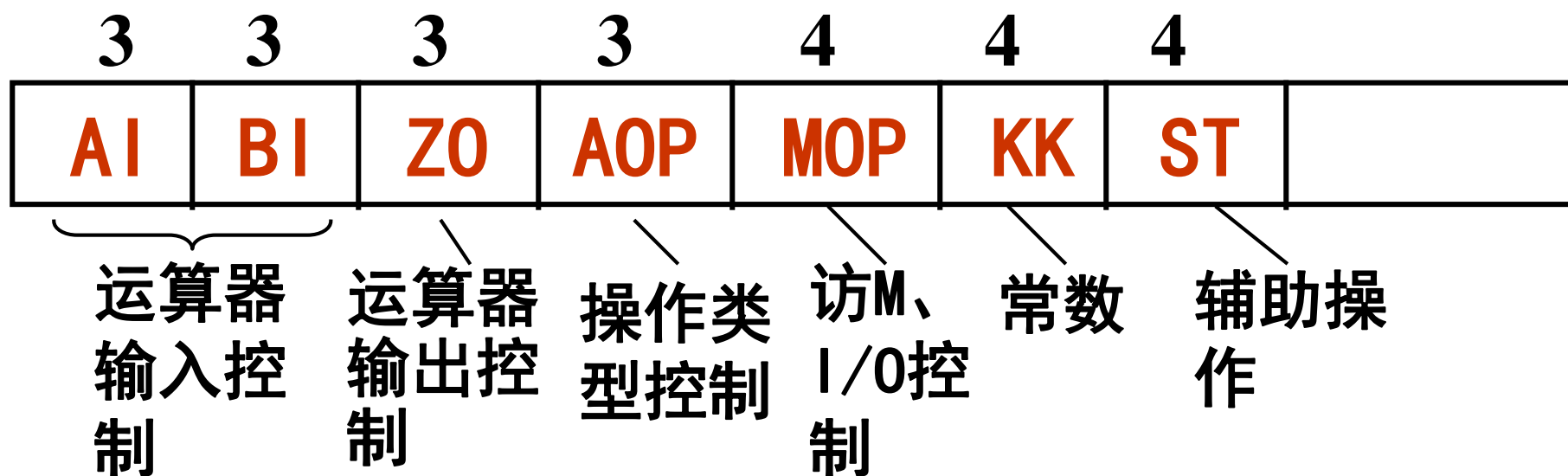
海纳百川 自强不息 厚德笃学 知行合一

(3) 混合型微指令

在垂直型的基础上增加一些不太复杂的并行操作。

微指令不长，便于编写；微程序不长，执行速度加快。

例：长城203微指令



书例题

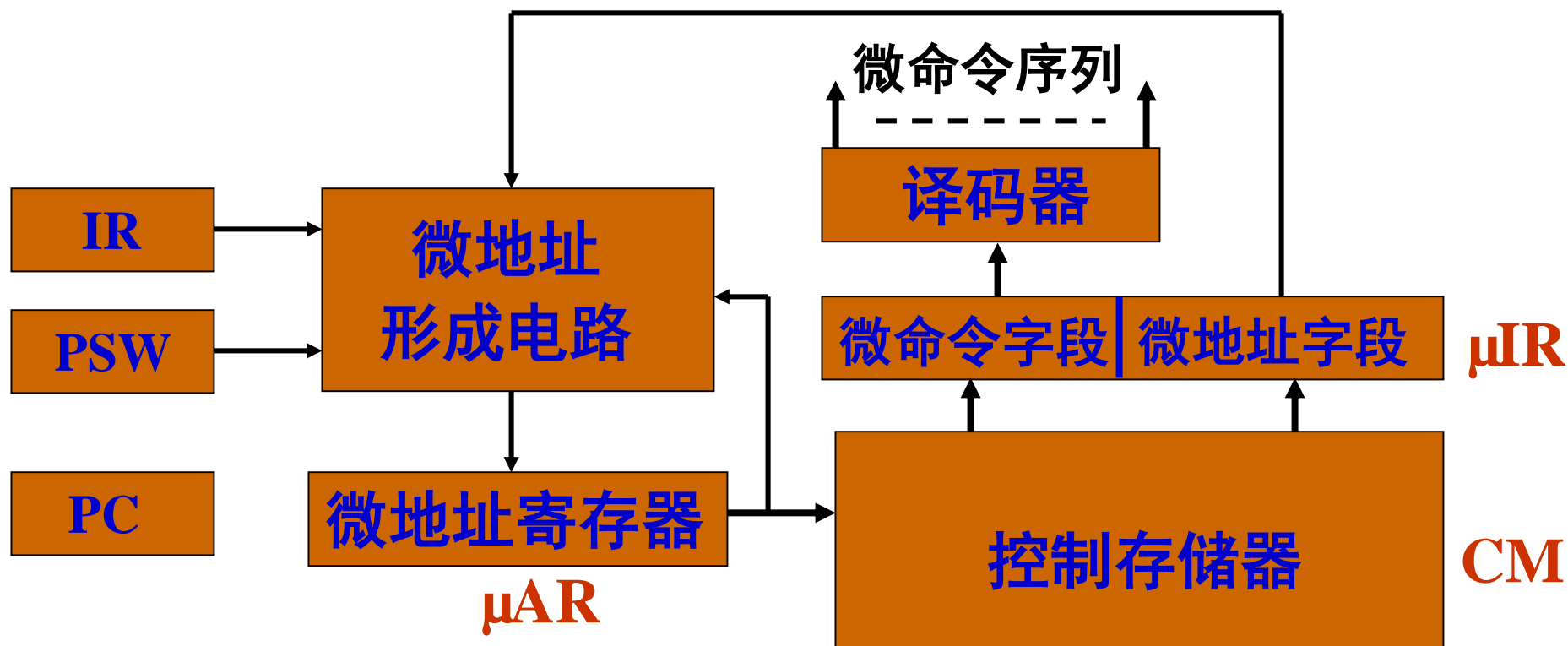
P125

习题:

P141-13、14题



5、微程序控制器组成原理



- 组成：IR、PC、PSW、时序系统等部件，控制存储器（**CM**）、微地址形成电路、微地址寄存器（ μAR ）、微指令寄存器（ μIR ）及译码电路等。

海纳百川 自强不息 厚德笃学 知行合一

主要部件

(1) 控制存储器CM

功能： 存放微程序。

CM属于CPU，不属于主存储器。

(2) 微指令寄存器 μIR

功能： 存放 现行微指令——当前正执行的微指令。

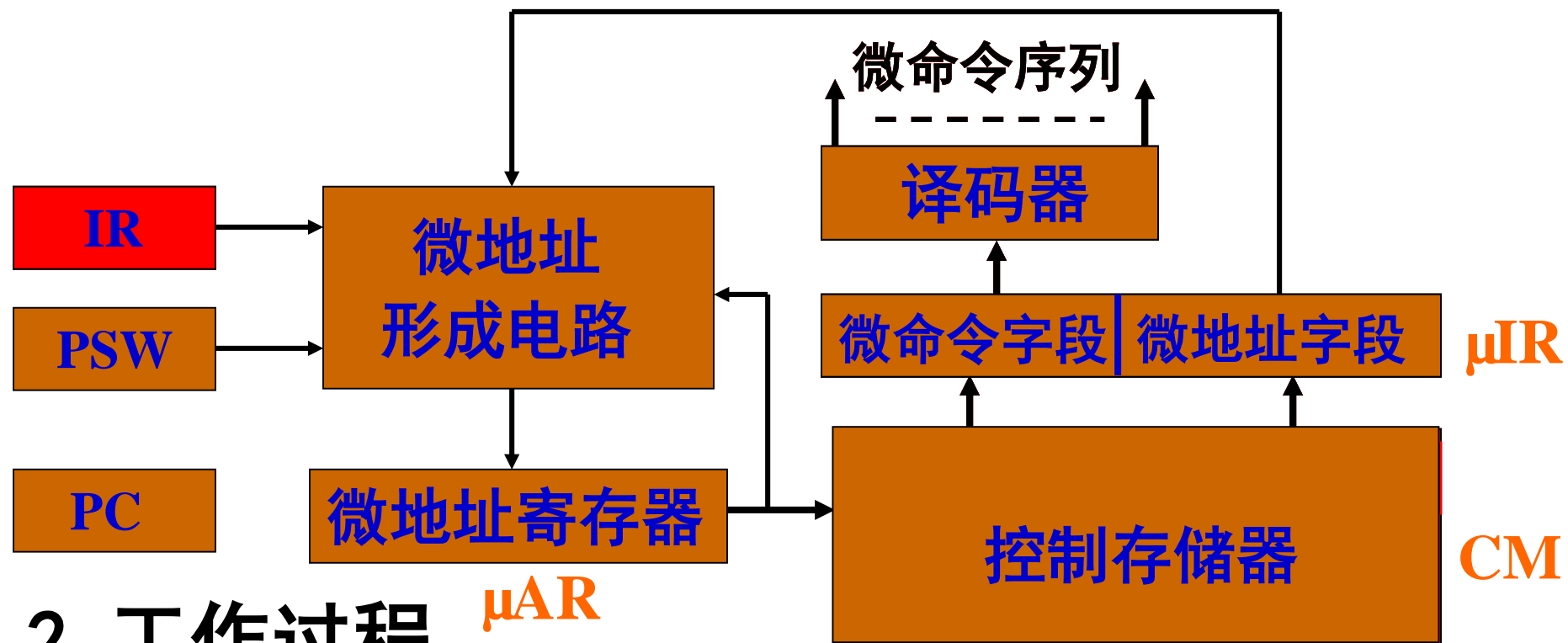
微命令字段： 提供一步操作所需的微命令。
(微操作控制字段)
微地址字段： { 指明后续微地址的形成方式。
(顺序控制字段) { 提供微地址的给定部分。

海纳百川 自强不息 厚德笃学 知行合一

(3) 微地址形成电路

功能：**提供两类微地址。**

- 微程序入口地址：**由机器指令操作码形成。**
- 后续微地址：由微地址字段、现行微地址、运行状态等形成。

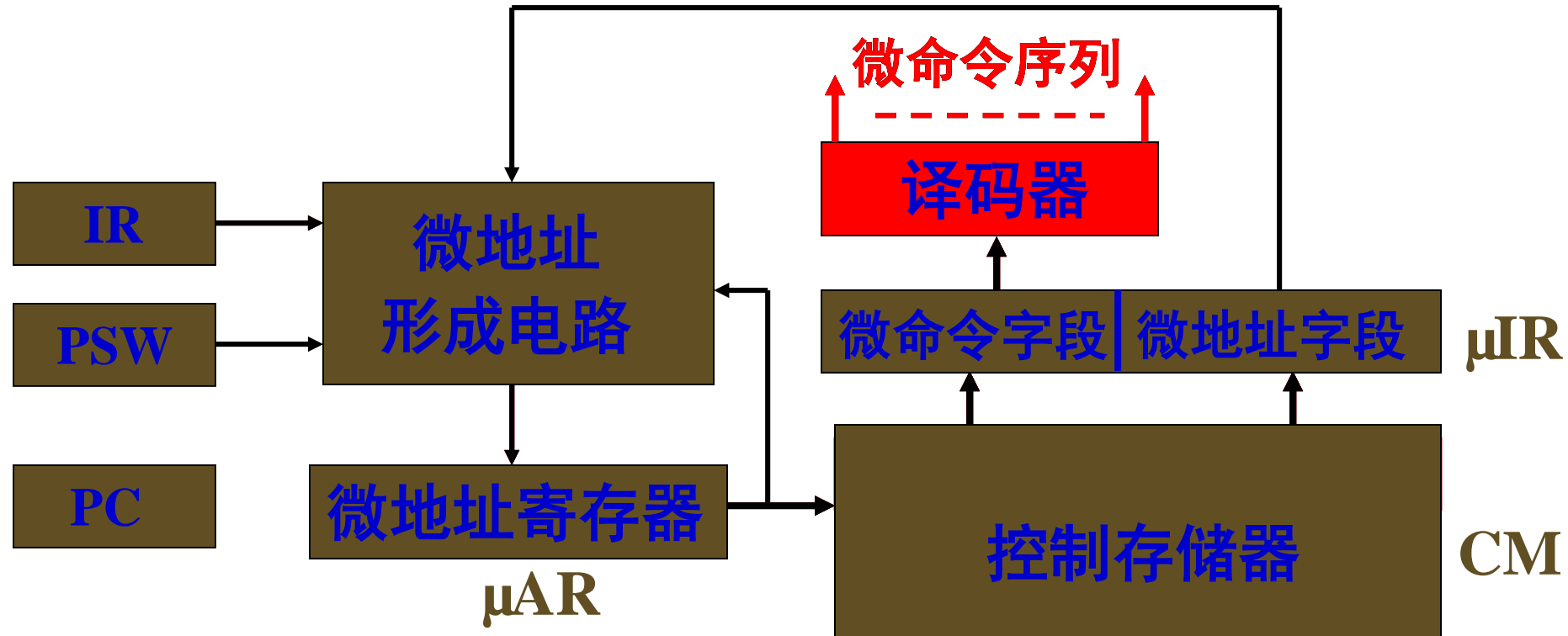


2. 工作过程

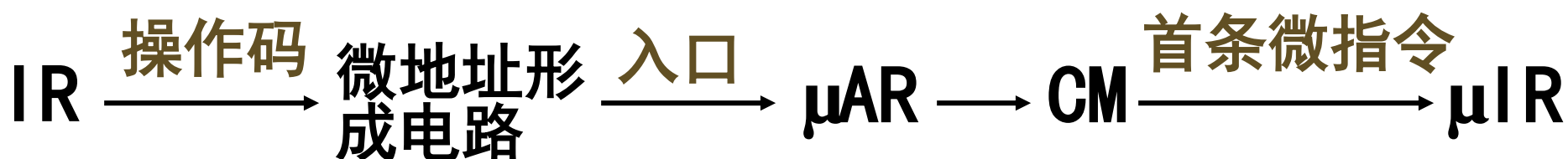
每条机器指令都对应一段微程序

(1) 取机器指令



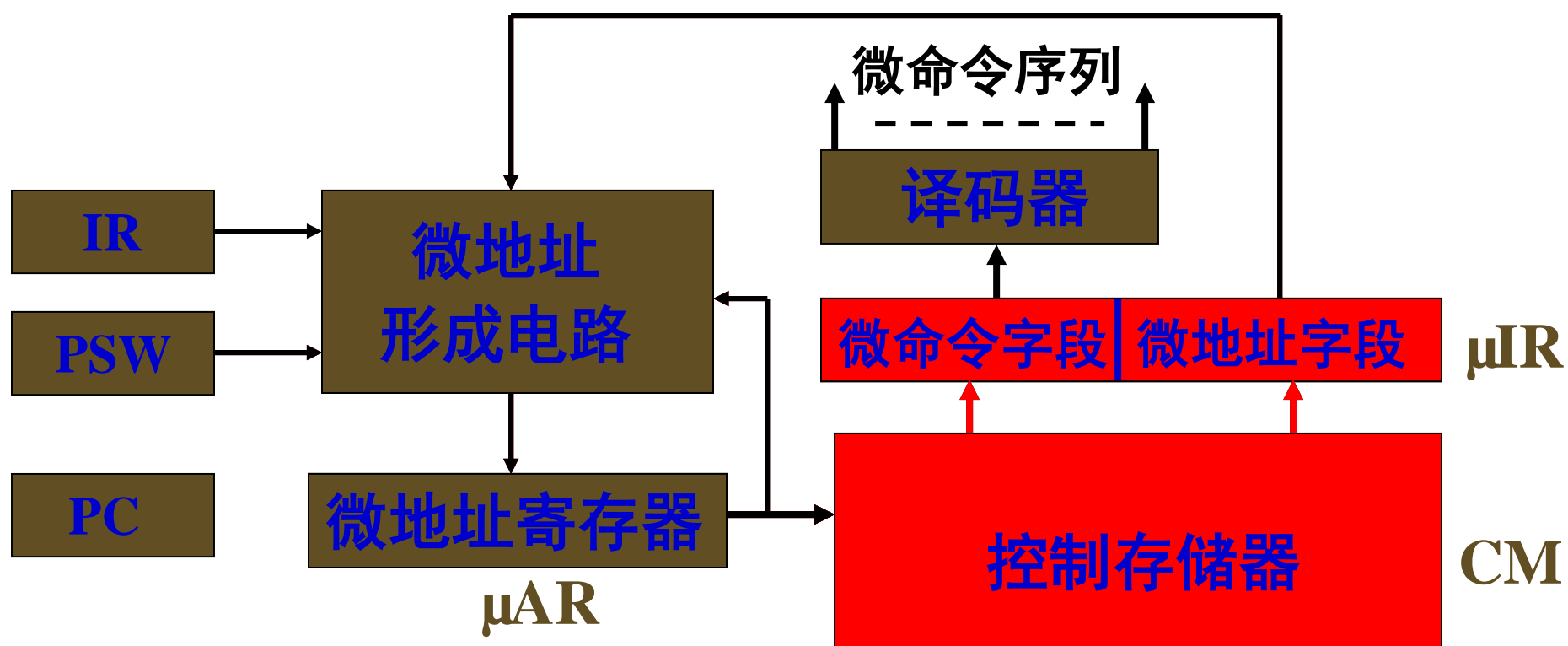


(2) 转微程序入口

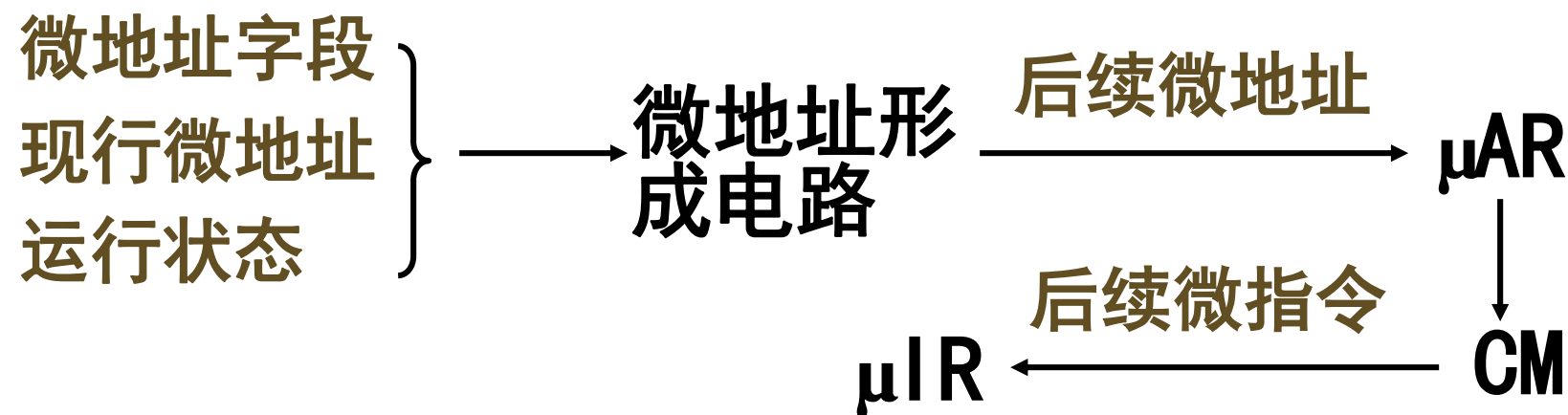


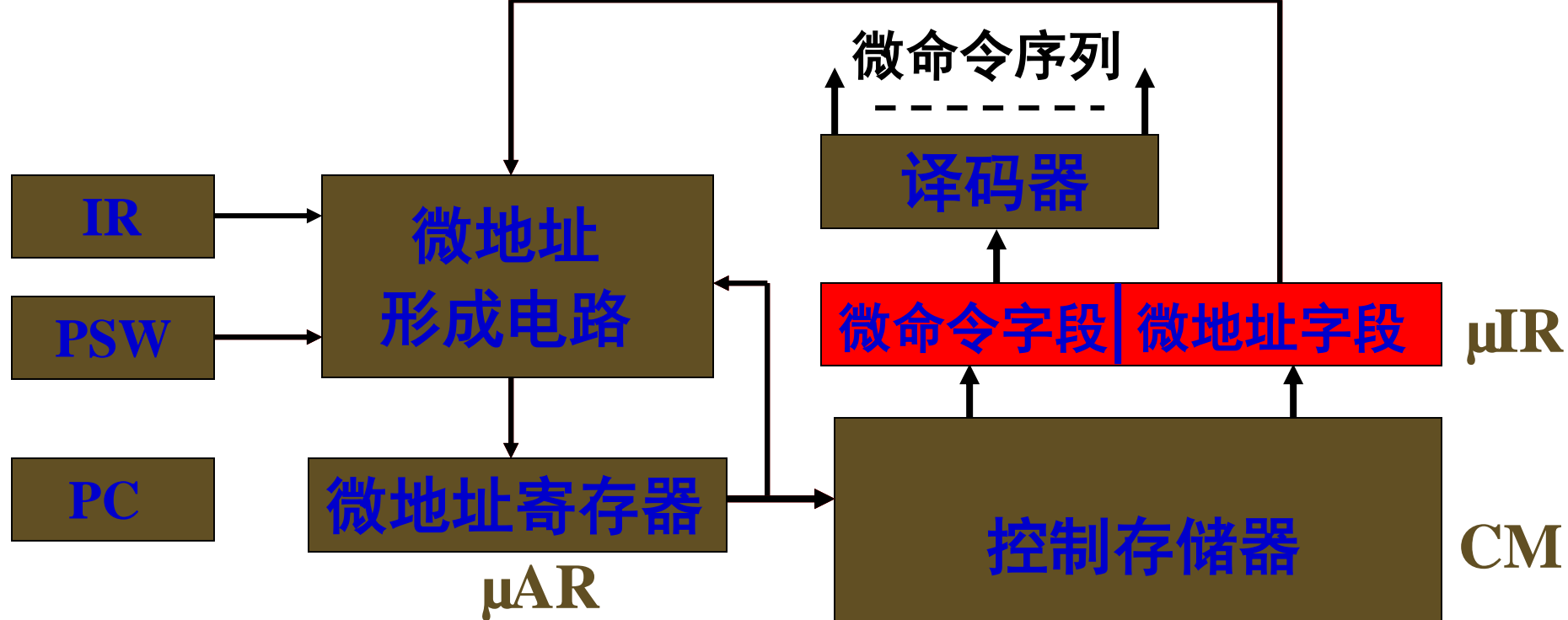
(3) 执行首条微指令





(4) 取后续微指令





(5) 执行后续微指令
同 (3)

(6) 返回，机器指令自行完毕

微程序执行完，返回CM（存放**取指微指令**的固定单元）。

控制信号举例

ADD rd ,rs, [rsl+disp]

仍以执行一条加法指令为例，它由4条微指令解释执行，一条微指令中的所有控制信号都是同时发出的。每条微指令所需的控制信号如下。

(1) **取指微指令**（具体格式也可参照书P129 表5-1）

① 指令地址送入地址总线：PC→AB

② 发访存控制命令：ADS(地址有效)，

$$M/\overline{IO}=1 \quad W/\overline{R}=0$$

从存储器取指令送入数据总线。

③ 指令送入指令寄存器：DB→IR

④ 程序计数器+1：PC+1

(2) 计算地址微指令

ADD rd,rs,[rs1+disp]

- ① 取两个源操作数（计算地址用）： $rs1 \rightarrow GR(rs1) \rightarrow ALU$, $disp \rightarrow ALU$ 。
- ② 加法运算：“+”
- ③ 有效地址送地址寄存器： **$ALU \rightarrow AR$** 。

(3) 取操作数微指令

- ① 数据地址送地址总线： $AR \rightarrow AB$ 。
- ② 发访存控制命令： $ADS, M / \overline{IO}, W / \overline{R}$ 。
由存储器将数据送数据总线DB。
- ③ 数据送入数据寄存器： $DB \rightarrow DR$

(4) 执行加法运算和传送结果微指令



- ① 两源操作数送ALU: $rs \rightarrow GR, (rs) \rightarrow ALU; DR \rightarrow ALU$
- ② 加法运算: “+”
- ③ 送结果: $ALU \rightarrow GR$

如何组织微指令产生上述信号？

微指令最简单的组成形式是将每个控制信号用一个控制位来表示，当该位为‘1’时定义为有控制信号。当该位为‘0’时，没有控制信号。 M / \overline{IO} 、 W / \overline{R} 则根据是访问存储器还是I/O设备，是写还是读而设置成1或0。

海纳百川 自强不息 厚德笃学 知行合一

- 当指令取入IR中以后，根据操作码进行译码，得到相应指令的第一条微指令的地址（**微程序的入口地址**）。
- 指令译码部件可用只读存储器组成，将操作码作为只读存储器的输入地址，该单元的内容即为相应的微指令在控制存储器中的地址，根据此地址从控制存储器取出微指令并将它存放在微指令寄存器中。
- 微指令分成两部分，产生控制信号的部分一般称为**控制（微命令）字段**，产生下址的部分称为下址字段。控制字段各位的输出通过连接线直接与受控制的门相连，于是就提供了相应的控制信号。

6、微程序设计技术

在实际进行微程序设计时还应关心下面三个问题：

- (1) 如何缩短微指令字长；
- (2) 如何减少微程序长度；
- (3) 如何提高微程序的执行速度。

微指令的编译法（编码译码方法）

- (1) 直接表示法（直接控制法）
- (2) 编码表示法（字段编译法）

(3) 混合表示法

中华人民共和国成立70周年 大连理工大学 建校70周年

- 把直接表示法与编码表示法结合使用，采用部分直接表示、部分编码表示的方法。
- 具体做法是将一些要求速度高的，或者一些相容的控制信号用直接表示法，将其他相斥信号用编码表示法。
- 直接表示法、编码表示法和混合表示法的三种编码方式的微指令都属于水平型微指令。

海纳百川 自强不息 厚德笃学 知行合一

7、微程序流的控制

中华人民共和国成立70周年 大连理工大学 建校70周年

- 当前正在执行的微指令，称为**现行微指令**。
- 现行微指令所在的控制存储器单元的地址，称为**现行微地址**。
- 现行微指令执行完毕后，下一条要执行的微指令称为**后继微指令**，后继微指令所在的控存单元地址称为**后继微地址**。
- 微程序流的控制，是指当前微指令执行完毕后，怎样控制产生后继微指令的微地址，即后继微地址的形成方式。

(1) 计数器方法（增量方式）

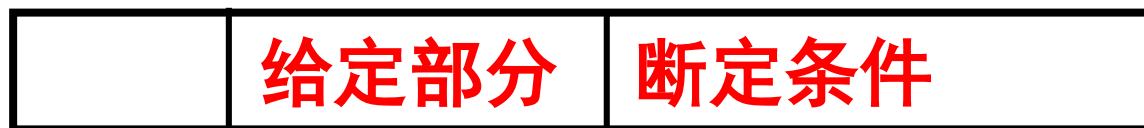
无条件转移：现行微指令给出转移微地址。

条件转移：现行微指令给出转移微地址和转移条件。

转微子程序：现行微指令给出微子程序入口。

(2) 断定方式

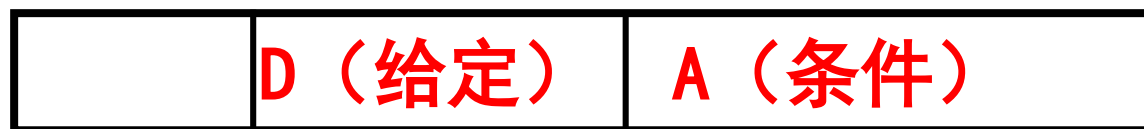
微指令



给定后续微地址
高位部分

指明后续微地址低
位部分的形成方式

例： 微指令



微地址10位，

约定：

位数可变

2位

16路
分支

- A= {
- 01 微地址低4位为操作码，D给定高6位；
 - 10 微地址低3位为机器指令源寻址方式编码，D给定高7位；—— 8路分支
 - 11 微地址低3位为机器指令目的寻址方式编码，D给定高7位。—— 8路分支

5.3.5 微程序控制方式优缺点及应用

1. 优点

- (1) 设计规整，设计效率高；
- (2) 易于修改、扩展指令系统功能；
- (3) 结构规整、简洁，可靠性高；
- (4) 性价比高。

2. 缺点

(1) 速度慢

(2) 执行效率不高

访存频繁
转移较多

未充分发挥数据
通路本身具有的
并行能力

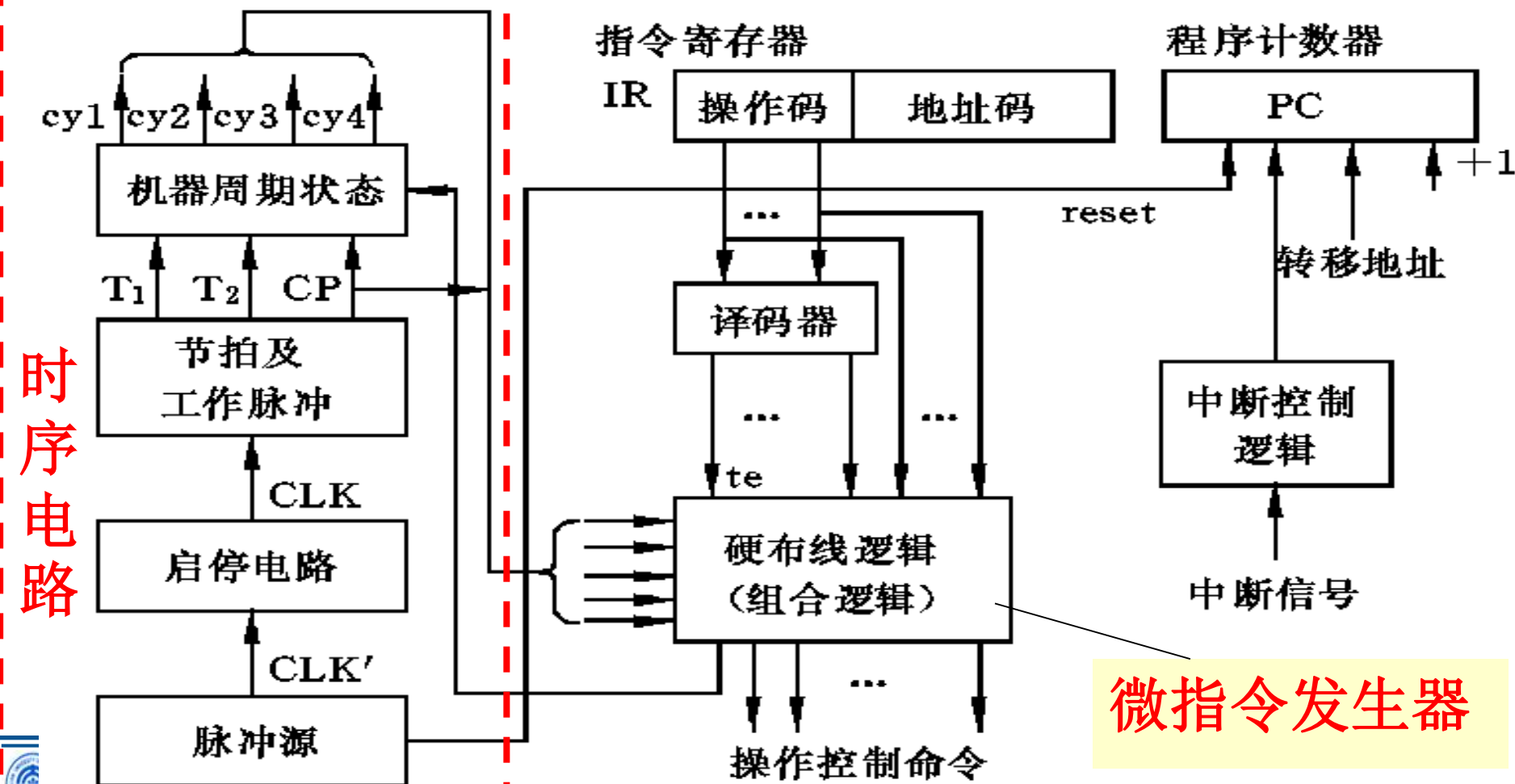
3. 应用范围

特别适用于系列机

用于速度要求不高、功能较复杂的机器中。

5.3 硬布线控制器原理

- **硬布线控制器**的控制信号是通过逻辑电路直接连线而产生的，所以又称为**组合逻辑控制器**。



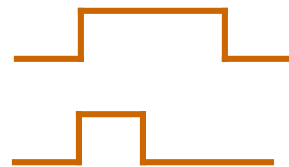
硬布线控制器组成



1. 微命令发生器

功能：产生所需的各种微命令

电位型
脉冲型



控制最基本的操作(微操作)的命令

构成微命令发生器：将产生微命令的条件综合化简，形成逻辑式，用组合逻辑电路实现。

2. 时序线路

功能：控制操作时间和操作时刻。



1) 主要特点

操作控制部件中操作控制命令是基本逻辑电路生成的。操作控制部件根据指令的要求和指令执行流程，按照一定顺序发出各种操作控制信号，控制有关部件，完成指定的操作。



硬布线方式主要应用在RISC机器中。

2) 设计操作控制器步骤

(1) 列出微操作命令的操作时间表

写出每条指令、每个周期、每个节拍的全部操作命令信号

(2) 进行微操作信号综合

写出各微操作控制信号的逻辑表达式。

(3) 画出微操作命令的逻辑图

根据逻辑表达式可画出对应每一个微操作信号的逻辑电路图，并用逻辑门电路实现之。

(4) 硬布线逻辑的实现途径

采用可编程器件实现。

3) 硬布线控制方式的优缺点及应用



中华人民共和国成立70周年



1949—2019
大连理工大学 建校70周年

硬布线控制方式

综合化简产生微命令的条件，形成逻辑式，用组合逻辑电路实现；

执行指令时，由组合逻辑电路（微命令发生器）在相应时间发出所需微命令，控制有关操作。

优缺点

产生微命令的速度较快。

设计不规整，设计效率较低； 控制器核心结构零乱，
不便于检查和调试。

不易修改、扩展指令系统功能。

海纳百川 自强不息 厚德笃学 知行合一

硬布线控制与微程序控制的比较



中华人民共和国成立70周年

1949—2019
大连理工大学 建校70周年

- 差别：操作控制信号的形成方法和原理不同。

(1) 实现方法上，

- **微程序控制器**的控制功能体现在存放微程序的控制存储器；其电路比较规整，增加或修改指令，只要增加或修改控存内容即可，控存可用EPROM芯片。
- **硬布线控制**则由逻辑门组合实现；其控制信号先用逻辑式列出，经化简后用电路实现，因而显得零乱且复杂，当需修改指令或增加指令时很麻烦。

指令系统复杂的计算机一般都采用微程序来实现控制功能。

海纳百川 自强不息 厚德笃学 知行合一

硬布线控制与微程序控制的比较



- 差别：操作控制信号的形成方法和原理不同。

(2) 在性能上

- 微程序控制的速度比硬布线控制的速度低，
- 而硬布线逻辑主要取决于电路延迟，因而在超高速机器中，对影响速度的关键部分如CPU，往往采用硬布线逻辑。
- 如在RISC中，一般选用硬布线逻辑。