

本节内容

微程序控制单元的设计

微程序控制单元的设计

设计步骤:

1. 分析每个阶段的微操作序列
2. 写出对应机器指令的微操作命令及节拍安排
3. 确定微指令格式
4. 编写微指令码点

取指周期-硬布线控制器的节拍安排

T_0 PC $\rightarrow$ MAR

T_0 1 $\rightarrow$ R

T_1 M (MAR) $\rightarrow$ MDR

T_1 (PC) + 1 $\rightarrow$ PC

T_2 MDR $\rightarrow$ IR

T_2 OP (IR) $\rightarrow$ ID

取指周期-微程序控制器的节拍安排

T_0 PC $\rightarrow$ MAR

T_0 1 $\rightarrow$ R

T_1 M (MAR) $\rightarrow$ MDR

T_1 (PC) + 1 $\rightarrow$ PC

T_2 MDR $\rightarrow$ IR

T_2 OP (IR) $\rightarrow$ 微地址形成部件

3 条微指令

微程序控制单元的设计

取指周期-硬布线控制器的节拍安排

T_0 $PC \rightarrow MAR$

T_0 $1 \rightarrow R$

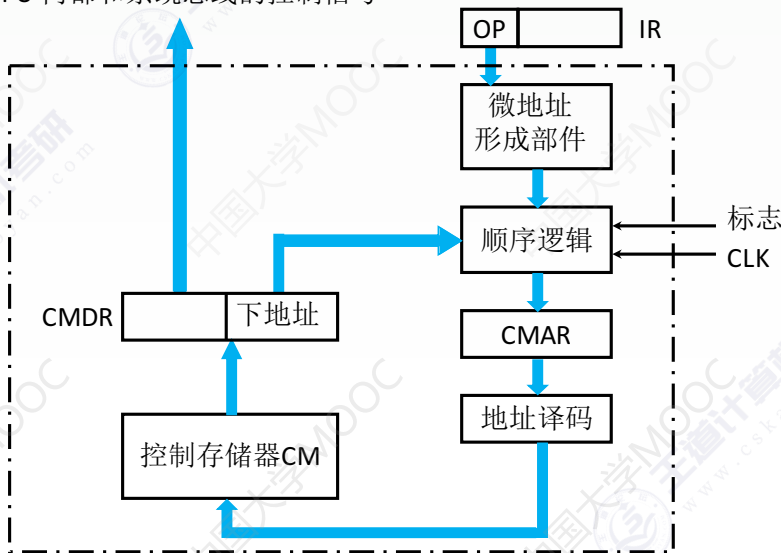
T_1 $M(MAR) \rightarrow MDR$

T_1 $(PC) + 1 \rightarrow PC$

T_2 $MDR \rightarrow IR$

T_2 $OP(IR) \rightarrow ID$

至 CPU 内部和系统总线的控制信号



取指周期-微程序控制器的节拍安排

T_0 $PC \rightarrow MAR$

T_0 $1 \rightarrow R$

T_1 $M(MAR) \rightarrow MDR$

T_1 $(PC) + 1 \rightarrow PC$

T_2 $MDR \rightarrow IR$

T_2 $OP(IR) \rightarrow$ 微地址形成部件

还需考虑 如何读出 这 3 条微指令，
以及如何转入下一个机器周期

$Ad(CMDR) \rightarrow CMAR$

$OP(IR) \rightarrow$ 微地址形成部件 $\rightarrow CMAR$

取指周期的
第一条微指
令地址由硬
件自动给出

微指令a

用微指令a的下地
址表示 b 的地址

微指令b

微指令c

用当前微指令的下地址
表示找到下一条微指令

微程序控制单元的设计

取指周期-硬布线控制器的节拍安排

T_0 $PC \rightarrow MAR$

T_0 $1 \rightarrow R$

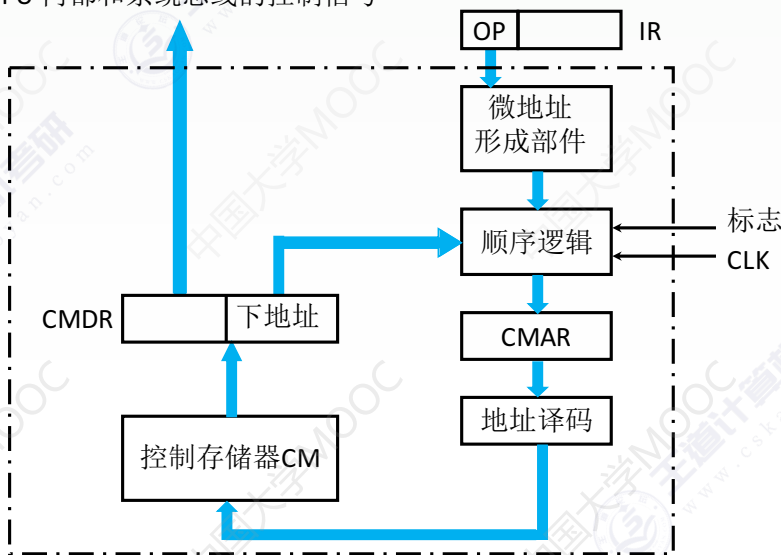
T_1 $M(MAR) \rightarrow MDR$

T_1 $(PC) + 1 \rightarrow PC$

T_2 $MDR \rightarrow IR$

T_2 $OP(IR) \rightarrow ID$

至 CPU 内部和系统总线的控制信号



取指周期-微程序控制器的节拍安排

T_0 $PC \rightarrow MAR$

T_0 $1 \rightarrow R$

微指令a

T_1 $Ad(CMDR) \rightarrow CMAR$

需要用 T_1 节拍确定下一条微指令的地址

T_2 $M(MAR) \rightarrow MDR$

T_2 $(PC) + 1 \rightarrow PC$

微指令b

T_3 $Ad(CMDR) \rightarrow CMAR$

需要用 T_3 节拍确定下一条微指令的地址

T_4 $MDR \rightarrow IR$

T_4 $OP(IR) \rightarrow$ 微地址形成部件

微指令c

T_5 微地址形成部件 $\rightarrow$ CMAR

根据指令操作码确定其执行周期微指令序列的首地址

显然，微程序控制器的速度比硬布线控制器更慢

微程序控制单元的设计

取指周期-硬布线控制器的节拍安排

T_0 $PC \rightarrow MAR$

T_0 $1 \rightarrow R$

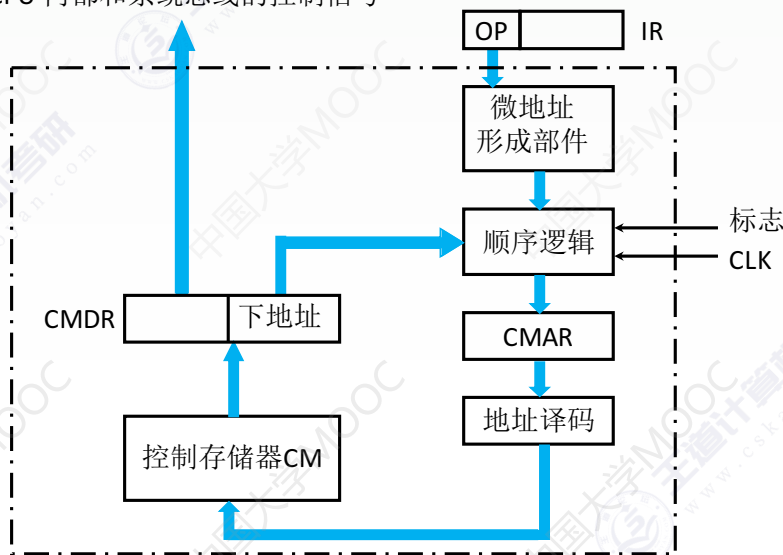
T_1 $M(MAR) \rightarrow MDR$

T_1 $(PC) + 1 \rightarrow PC$

T_2 $MDR \rightarrow IR$

T_2 $OP(IR) \rightarrow ID$

至 CPU 内部和系统总线的控制信号



取指周期-微程序控制器的节拍安排

T_0 $PC \rightarrow MAR$

T_0 $1 \rightarrow R$

微指令a

T_1 $Ad(CMDR) \rightarrow CMAR$

T_2 $M(MAR) \rightarrow MDR$

T_2 $(PC) + 1 \rightarrow PC$

微指令b

T_3 $Ad(CMDR) \rightarrow CMAR$

T_4 $MDR \rightarrow IR$

微指令c

T_5 $OP(IR) \rightarrow \text{微地址形成部件} \rightarrow CMAR$

取指周期最后一条微指令完成后，用一个特殊的微操作确定执行周期的微程序首地址

显然，微程序控制器的速度比硬布线控制器更慢

微程序控制单元的设计

设计步骤:

1. 分析每个阶段的微操作序列
2. 写出对应机器指令的微操作命令及节拍安排
 - (1) 写出每个周期所需要的微操作(参照硬布线)
 - (2) 补充微程序控制器特有的微操作:
 - a. 取指周期:
 $Ad(CMDR) \rightarrow CMAR$
 $OP(IR) \rightarrow \text{微地址形成部件} \rightarrow CMAR$
 - b. 执行周期:
 $Ad(CMDR) \rightarrow CMAR$
3. 确定微指令格式
4. 编写微指令码点

每条微指令结束
之后都需要进行

每条微指令结束
之后都需要进行

取指周期的最后一条微指令完成后, 要根据指令操作码确定其执行周期的微程序首地址

微程序控制单元的设计



设计步骤:

1. 分析每个阶段的微操作序列
2. 写出对应机器指令的微操作命令及节拍安排
 - (1) 写出每个周期所需要的微操作(参照硬布线)
 - (2) 补充微程序控制器特有的微操作:

a. 取指周期:

$Ad (CMDR) \rightarrow CMAR$

$OP (IR) \rightarrow CMAR$

b. 执行周期:

$Ad(CMDR) \rightarrow CMAR$

3. 确定微指令格式

根据微操作个数决定采用何种编码方式, 以确定微指令的操作控制字段的位数。

根据CM中存储的微指令总数, 确定微指令的顺序控制字段的位数。

最后按操作控制字段位数和顺序控制字段位数就可确定微指令字长。

4. 编写微指令码点

微程序控制单元的设计

设计步骤:

1. 分析每个阶段的微操作序列
2. 写出对应机器指令的微操作命令及节拍安排
 - (1) 写出每个周期所需要的微操作(参照硬布线)
 - (2) 补充微程序控制器特有的微操作:

a. 取指周期:

$Ad (CMDR) \rightarrow CMAR$

$OP (IR) \rightarrow CMAR$

b. 执行周期:

$Ad(CMDR) \rightarrow CMAR$

3. 确定微指令格式

根据微操作个数决定采用何种编码方式, 以确定微指令的操作控制字段的位数。

根据CM中存储的微指令总数, 确定微指令的顺序控制字段的位数。

最后按操作控制字段位数和顺序控制字段位数就可确定微指令字长。

4. 编写微指令码点

根据操作控制字段每一位代表的微操作命令, 编写每一条微指令的码点。

微程序设计分类



1. 静态微程序设计和动态微程序设计

静态 微程序无需改变，采用 **ROM**

动态 通过 **改变微指令** 和 **微程序** 改变机器指令
有利于仿真，采用 **EPROM**

2. 毫微程序设计

毫微程序设计的基本概念

微程序设计 用 **微程序解释机器指令**

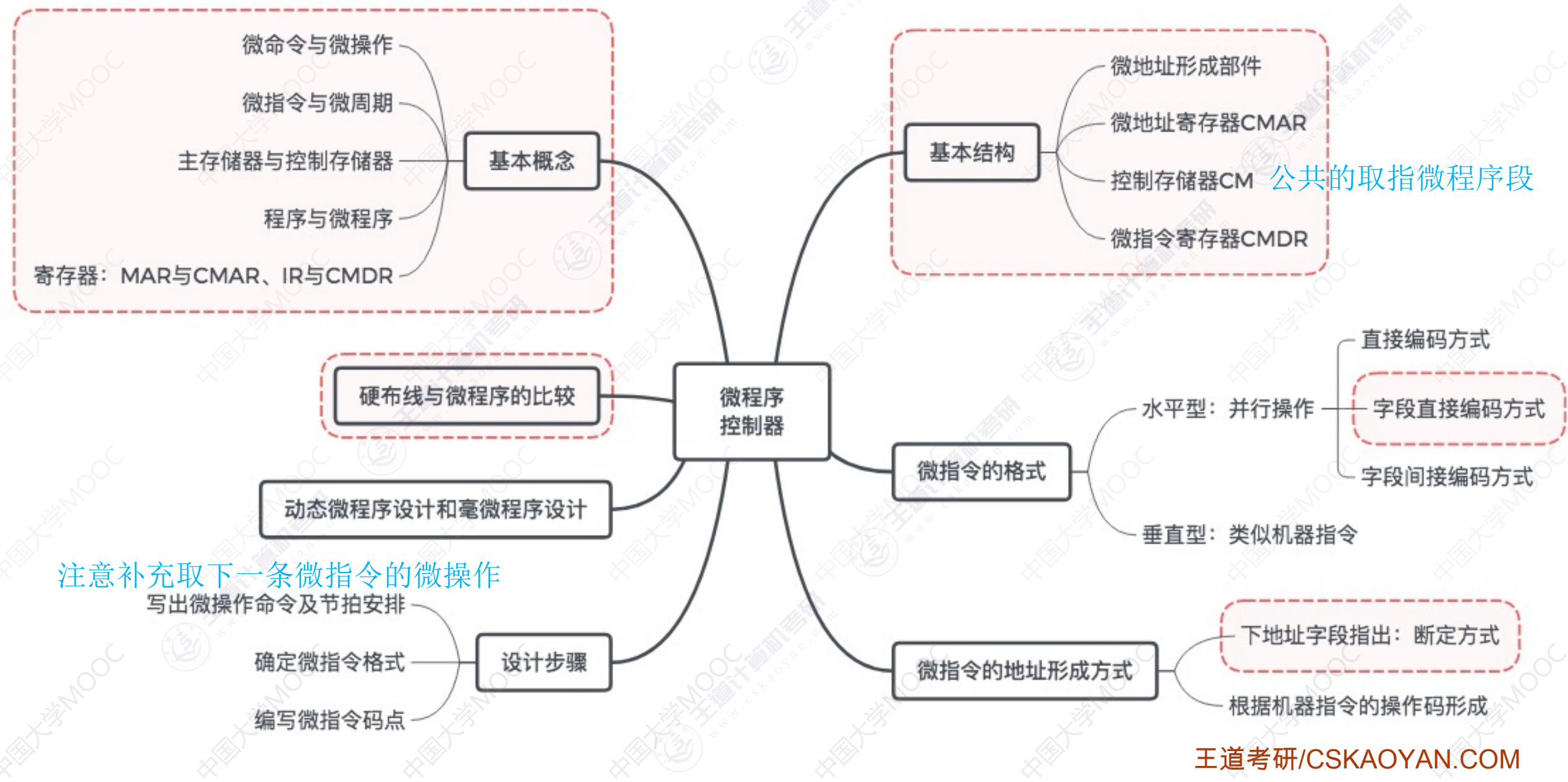
毫微程序设计 用 **毫微程序解释微程序**

毫微指令与微指令 的关系好比 **微指令与机器指令** 的关系

硬布线与微程序的比较

类 别 对 比 项 目	微程序控制器	硬布线控制器
工作原理	微操作控制信号以微程序的形式存放在控制存储器中，执行指令时读出即可	微操作控制信号由组合逻辑电路根据当前的指令码、状态和时序，即时产生
执行速度	慢	快
规整性	较规整	烦琐、不规整
应用场合	CISC CPU	RISC CPU
易扩充性	易扩充修改	困难

微程序控制器回顾





公众号：王道在线



b站：王道计算机教育



抖音：王道计算机考研