

王道考研——组成原理

WWW.CSKAOYAN.COM

第五章 中央处理器

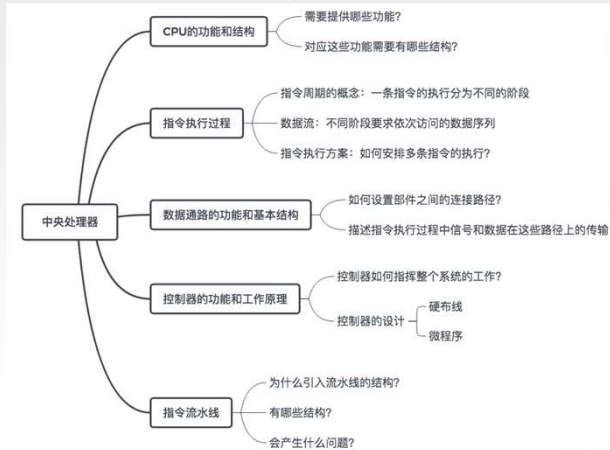
本节内容

中央处理器

CPU的功能和
基本结构

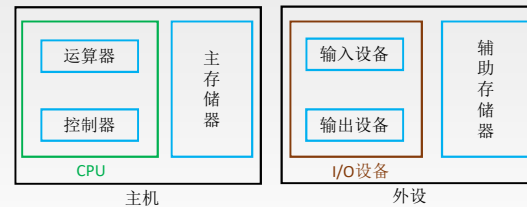
王道考研/CSKAOYAN.COM

本章总览



王道考研/CSKAOYAN.COM

CPU的功能

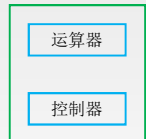


主机

外设

王道考研/CSKAOYAN.COM

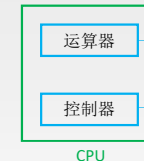
CPU的功能



- 指令控制。**完成取指令、分析指令和执行指令的操作，即程序的顺序控制。
- 操作控制。**一条指令的功能往往是由若干操作信号的组合来实现的。CPU管理并产生由内存取出的每条指令的操作信号，把各种操作信号送往相应的部件，从而控制这些部件按指令的要求进行动作。
- 时间控制。**对各种操作加以时间上的控制。时间控制要为每条指令按时间顺序提供应有的控制信号。
- 数据加工。**对数据进行算术和逻辑运算。
- 中断处理。**对计算机运行过程中出现的异常情况和特殊请求进行处理。

王道考研/CSKAQYAN.COM

运算器和控制器的功能



对数据进行加工

协调并控制计算机各部件执行程序的指令序列，基本功能包括取指令、分析指令、执行指令

取指令：自动形成指令地址；自动发出取指令的命令。

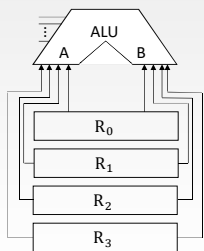
分析指令：操作码译码(分析本条指令要完成什么操作)；产生操作数的有效地址。

执行指令：根据分析指令得到的“操作命令”和“操作数地址”，形成操作信号控制序列，控制运算器、存储器以及I/O设备完成相应的操作。

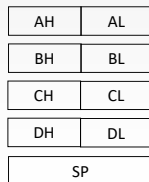
中断处理：管理总线及输入输出；处理异常情况(如掉电)和特殊请求(如打印机请求打印一行字符)。

王道考研/CSKAQYAN.COM

运算器的基本结构



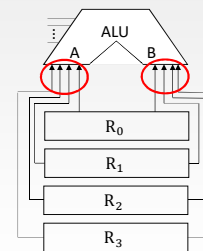
- 算术逻辑单元：主要功能是进行算术/逻辑运算。
- 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。



专用数据通路方式：根据指令执行过程中的数据和地址的流动方向安排连接线路。

王道考研/CSKAQYAN.COM

运算器的基本结构

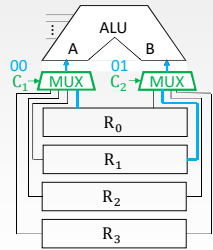


如果直接用导线连接，相当于多个寄存器同时并且一直向ALU传输数据
解决方法1. 使用多路选择器

专用数据通路方式：根据指令执行过程中的数据和地址的流动方向安排连接线路。

王道考研/CSKAQYAN.COM

运算器的基本结构



如果直接用导线连接，相当于多个寄存器同时并且一直向ALU传输数据

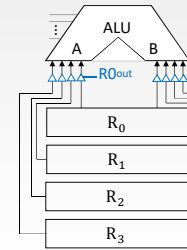
解决方法1. 使用多路选择器
根据控制信号选择一路输出
解决方法2. 使用三态门
可以控制每一路是否输出

专用数据通路方式：根据指令执行过程中的数据和地址的流动方向安排连接线路。

王道考研/CSKAQYAN.COM

运算器的基本结构

CPU内部单总线方式：将所有寄存器的输入端和输出端都连接到一条公共的通道上。



如果直接用导线连接，相当于多个寄存器同时并且一直向ALU传输数据

解决方法1. 使用多路选择器
根据控制信号选择一路输出
解决方法2. 使用三态门
可以控制每一路是否输出
如：R0out为1时R0中的数据输出到A端，
R0out为0时R0中的数据无法输出到B端

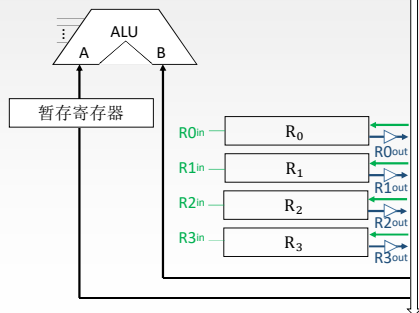
性能较高，基本不存在数据冲突现象，但结构复杂，硬件量大，不易实现。

专用数据通路方式：根据指令执行过程中的数据和地址的流动方向安排连接线路。

王道考研/CSKAQYAN.COM

运算器的基本结构

CPU内部单总线方式：将所有寄存器的输入端和输出端都连接到一条公共的通道上。

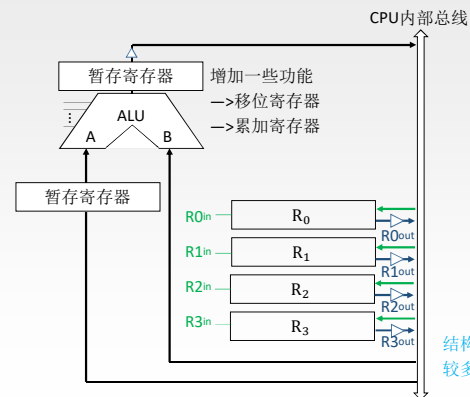


1. 算术逻辑单元：主要功能是进行算术/逻辑运算。
2. 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。
3. 暂存寄存器：用于暂存从主存读来的数据，这个数据不能存放在通用寄存器中，否则会破坏其原有内容。
如：两个操作数分别来自主存和R0，最后结果存回R0，那么从主存中取来的操作数直接放入暂存器，就不会破坏运算前R0的内容。

结构简单，容易实现，但数据传输存在较多冲突的现象，性能较低。

王道考研/CSKAQYAN.COM

运算器的基本结构

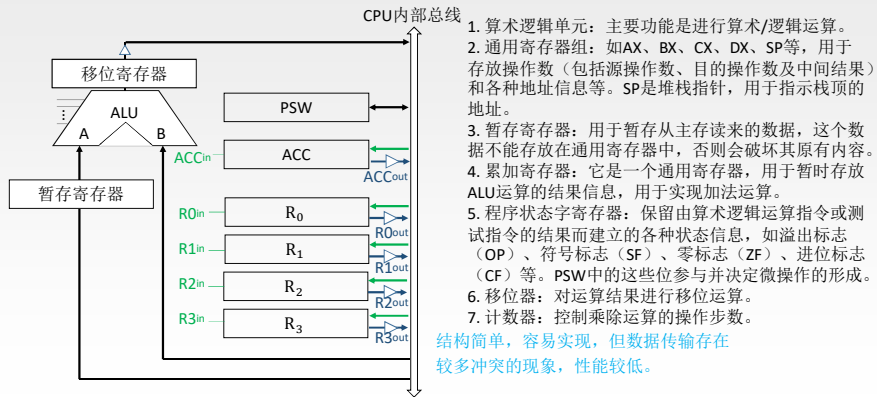


1. 算术逻辑单元：主要功能是进行算术/逻辑运算。
2. 通用寄存器组：如AX、BX、CX、DX、SP等，用于存放操作数（包括源操作数、目的操作数及中间结果）和各种地址信息等。SP是堆栈指针，用于指示栈顶的地址。
3. 暂存寄存器：用于暂存从主存读来的数据，这个数据不能存放在通用寄存器中，否则会破坏其原有内容。
如：两个操作数分别来自主存和R0，最后结果存回R0，那么从主存中取来的操作数直接放入暂存器，就不会破坏运算前R0的内容。

结构简单，容易实现，但数据传输存在较多冲突的现象，性能较低。

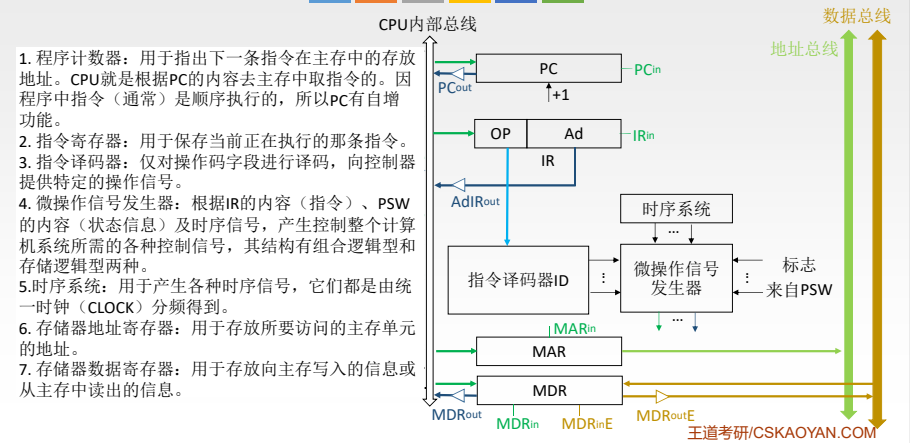
王道考研/CSKAQYAN.COM

运算器的基本结构



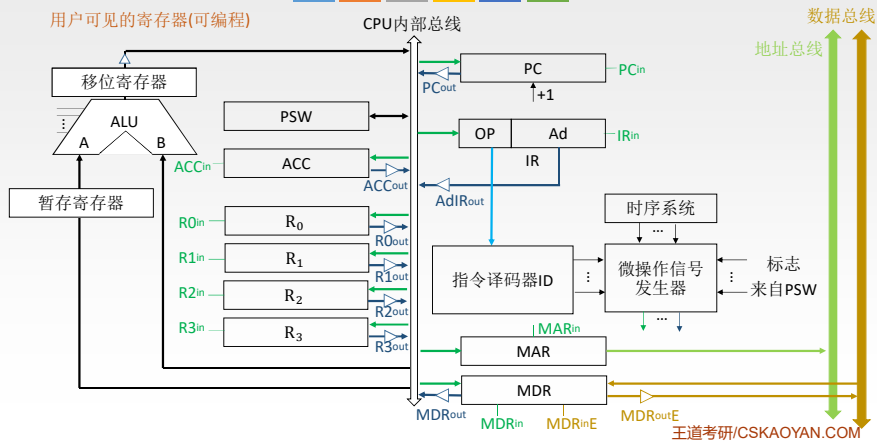
王道考研/CSKAOYAN.COM

控制器的基本结构

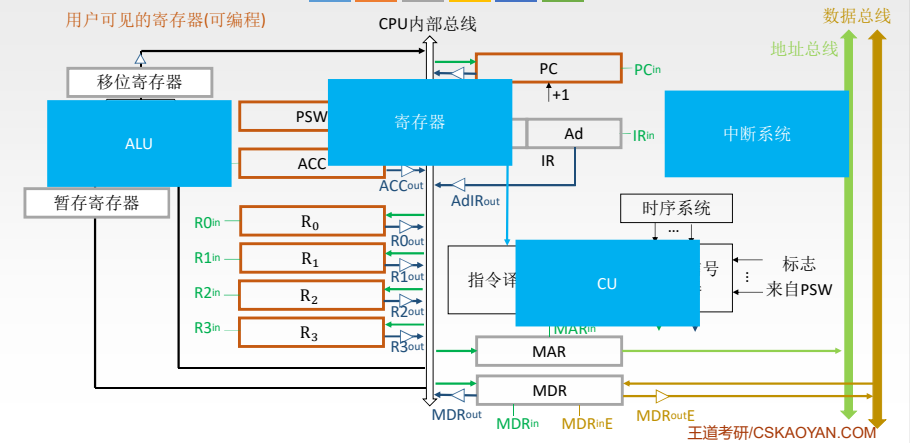


王道考研/CSKAOYAN.COM

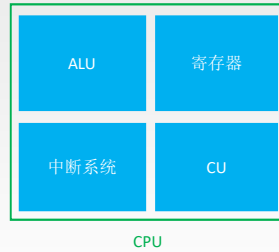
CPU的基本结构



CPU的基本结构

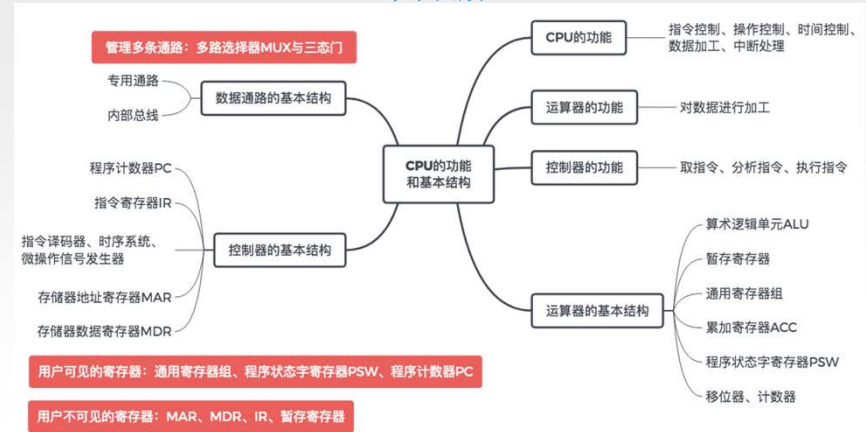


CPU的基本结构



王道考研/CSKAQYAN.COM

本节回顾



王道考研/CSKAQYAN.COM

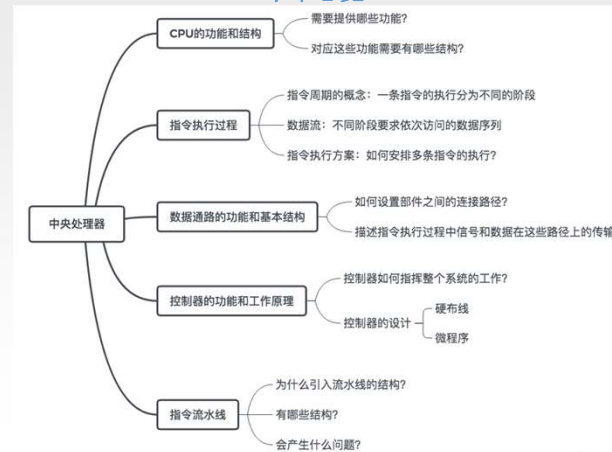
本节内容

中央处理器

指令执行过程

王道考研/CSKAQYAN.COM

本章总览



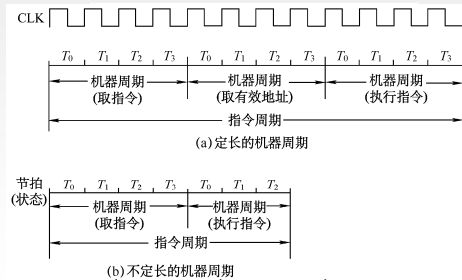
王道考研/CSKAQYAN.COM

指令周期

指令周期：CPU从主存中每取出并执行一条指令所需的全部时间。

指令周期常常用若干**机器周期**来表示，机器周期又叫**CPU周期**。

一个**机器周期**又包含若干**时钟周期**（也称为**节拍**、**T周期**或**CPU时钟周期**，它是CPU操作的最基本单位）。

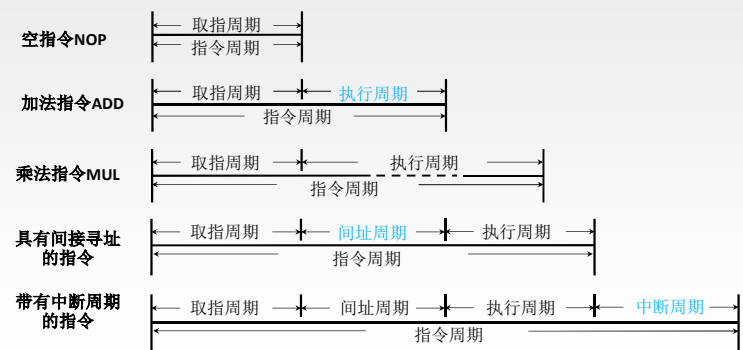


每个指令周期内机器周期数可以不等，每个机器周期内的节拍数也可以不等。

王道考研/CSKAQYAN.COM

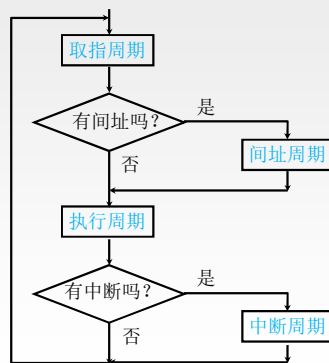
指令周期

每个指令周期内机器周期数可以不等，每个机器周期内的节拍数也可以不等。



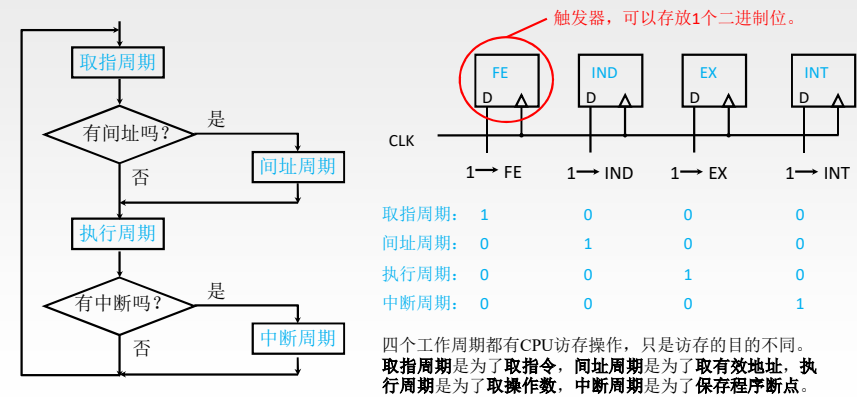
王道考研/CSKAQYAN.COM

指令周期流程



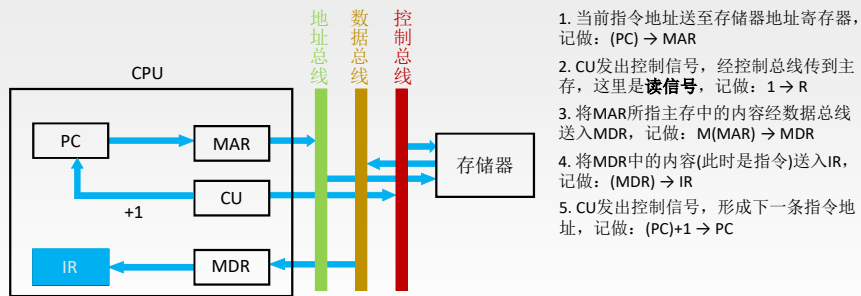
王道考研/CSKAQYAN.COM

指令周期流程



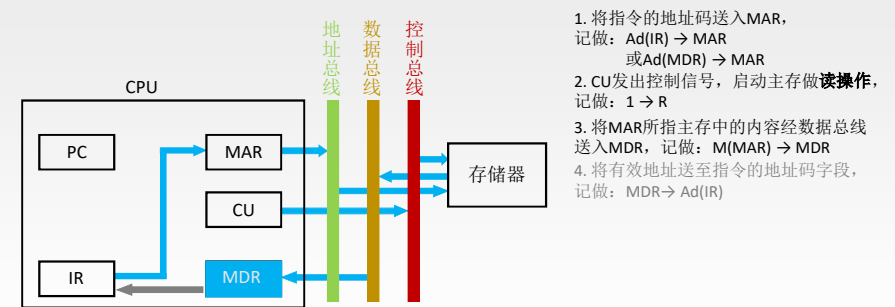
王道考研/CSKAQYAN.COM

指令周期的数据流-取指周期



王道考研/CSKAQYAN.COM

指令周期的数据流-间址周期



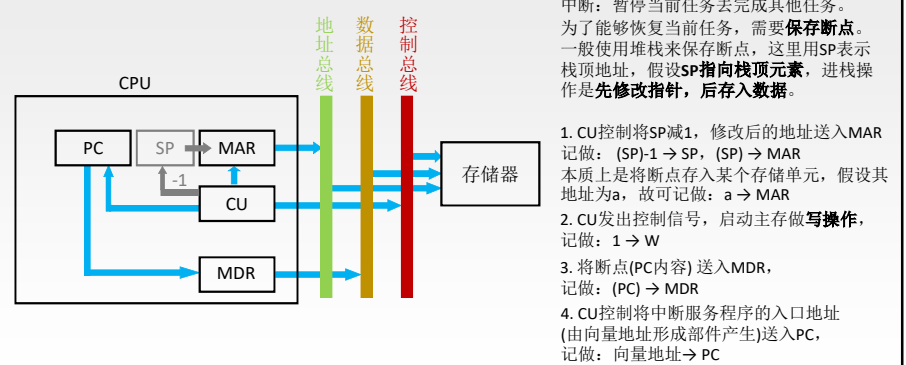
王道考研/CSKAQYAN.COM

指令周期的数据流-执行周期

执行周期的任务是根据IR中的指令字的操作码和操作数通过ALU操作产生执行结果。不同指令的执行周期操作不同，因此没有统一的数据流向。

王道考研/CSKAQYAN.COM

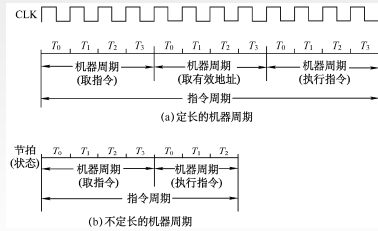
指令周期的数据流-中断周期



王道考研/CSKAQYAN.COM

指令执行方案

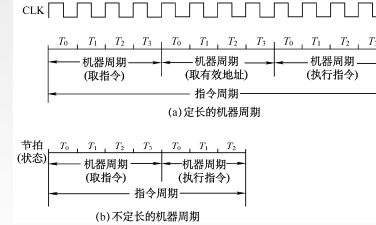
一个指令周期通常要包括几个时间段（执行步骤），每个步骤完成指令的一部分功能，几个依次执行的步骤完成这条指令的全部功能。



王道考研/CSKAQYAN.COM

指令执行方案

一个指令周期通常要包括几个时间段（执行步骤），每个步骤完成指令的一部分功能，几个依次执行的步骤完成这条指令的全部功能。



方案1. 单指令周期

对所有指令都选用相同的执行时间来完成。指令之间串行执行；指令周期取决于执行时间最长的指令的执行时间。

对于那些本来可以在更短时间内完成的指令，要使用这个较长的周期来完成，会降低整个系统的运行速度。

方案2. 多指令周期

对不同类型的指令选用不同的执行步骤来完成。指令之间串行执行；可选用不同个数的时钟周期来完成不同指令的执行过程。

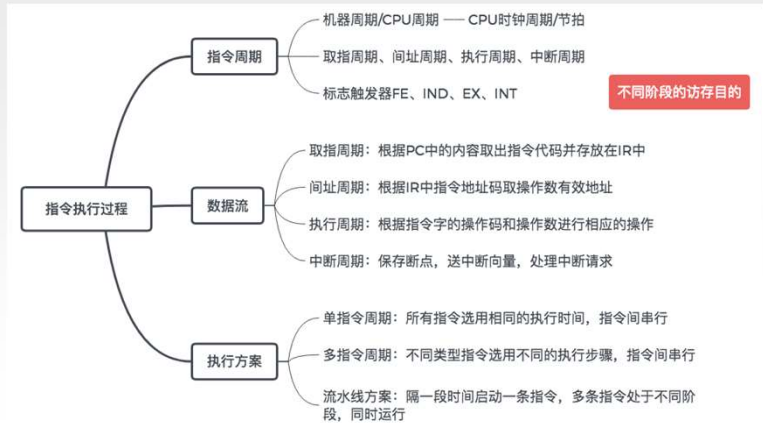
需要更复杂的硬件设计。

方案3. 流水线方案

在每一个时钟周期启动一条指令，尽量让多条指令同时运行，但各自处在不同的执行步骤中。指令之间并行执行。

王道考研/CSKAQYAN.COM

本节回顾



王道考研/CSKAQYAN.COM

本节内容

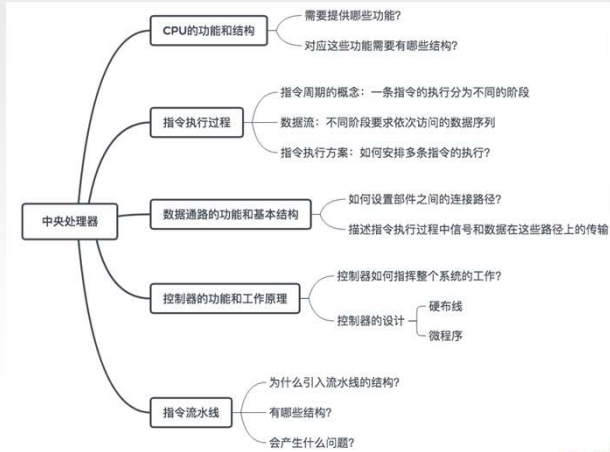
中央处理器

数据通路的功能和基本结构

单总线

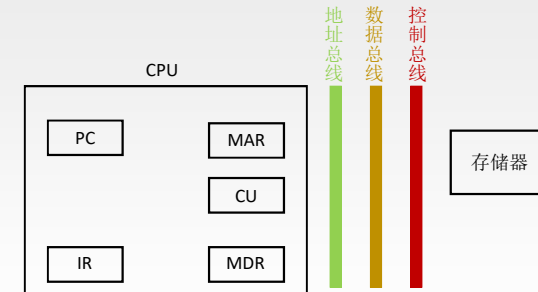
王道考研/CSKAQYAN.COM

本章总览



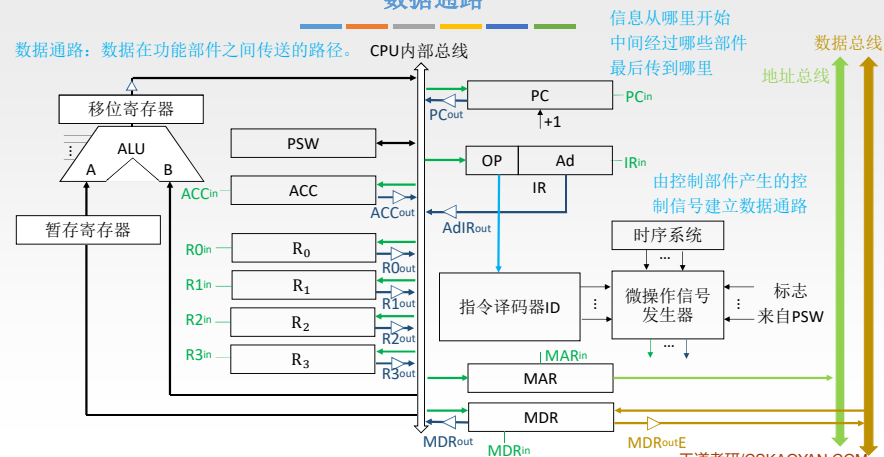
王道考研/CSKAOYAN.COM

指令周期的数据流



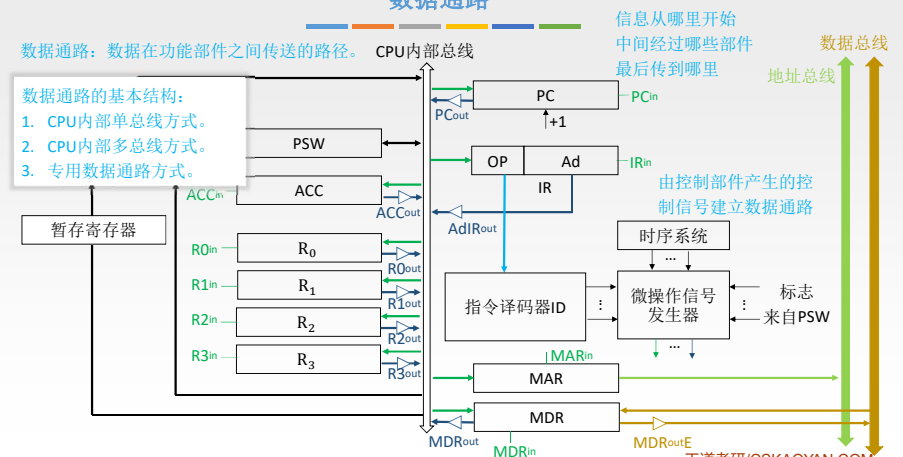
王道考研/CSKAOYAN.COM

数据通路



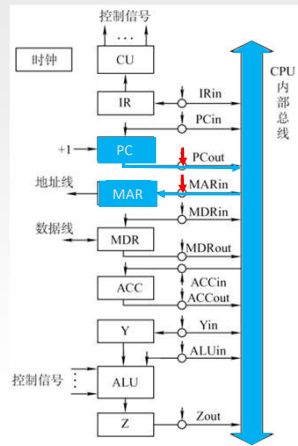
王道考研/CSKAOYAN.COM

数据通路



王道考研/CSKAOYAN.COM

数据通路-CPU内部单总线方式



内部总线是指同一部件，如CPU内部连接各寄存器及运算部件之间的总线；
系统总线是指同一台计算机系统的各部件，如CPU、内存、通道和各类I/O接口间互相连接的总线。

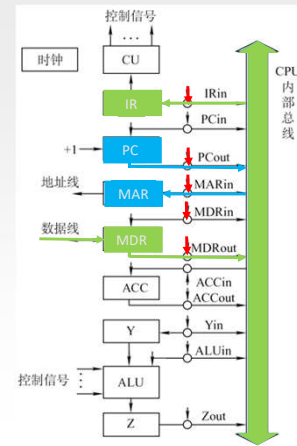
1. 寄存器之间数据传送

比如把PC内容送至MAR，实现传送操作的流程及控制信号为：
 (PC)→Bus PCout有效，PC内容送总线
 Bus→MAR MARin有效，总线内容送MAR

关于是否加括号见王道p7最下方的讨论，写数据通路时，更关心数据流经的路径，可以省略括号，但从真题来看，考微操作序列居多，所以建议都加上，题目给出示例时跟题目保持一致。

王道考研/CSKAQYAN.COM

数据通路-CPU内部单总线方式



内部总线是指同一部件，如CPU内部连接各寄存器及运算部件之间的总线；
系统总线是指同一台计算机系统的各部件，如CPU、内存、通道和各类I/O接口间互相连接的总线。

1. 寄存器之间数据传送

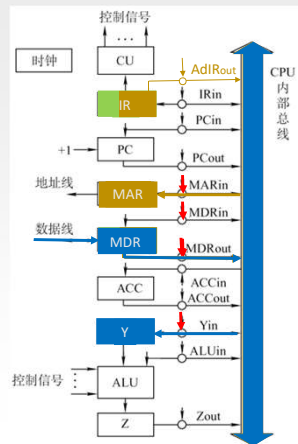
比如把PC内容送至MAR，实现传送操作的流程及控制信号为：
 (PC)→Bus PCout有效，PC内容送总线
 Bus→MAR MARin有效，总线内容送MAR

2. 主存与CPU之间的数据传送

比如CPU从主存读取指令，实现传送操作的流程及控制信号为：
 (PC)→Bus→MAR PCout和MARin有效，现行指令地址→MAR
 1→R CU发读命令(通过控制总线发出，图中未画出)
 MEM(MAR)→MDR MDRin有效
 MDR→Bus→IR MDRout和IRin有效，现行指令→IR

王道考研/CSKAQYAN.COM

数据通路-CPU内部单总线方式



1. 寄存器之间数据传送

比如把PC内容送至MAR，实现传送操作的流程及控制信号为：
 (PC)→Bus PCout有效，PC内容送总线
 Bus→MAR MARin有效，总线内容送MAR

2. 主存与CPU之间的数据传送

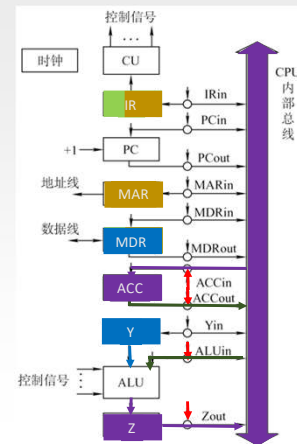
比如CPU从主存读取指令，实现传送操作的流程及控制信号为：
 (PC)→Bus→MAR PCout和MARin有效，现行指令地址→MAR
 1→R CU发读命令(通过控制总线发出，图中未画出)
 MEM(MAR)→MDR MDRin有效
 MDR→Bus→IR MDRout和IRin有效，现行指令→IR

3. 执行算术或逻辑运算

比如一条加法指令，微操作序列及控制信号为：
 Ad(IR)→Bus→MAR MDRout和MARin有效 或 AdIRout和MARin有效
 1→R CU发读命令
 MEM(MAR)→数据总线→MDR MDRin有效
 MDR→Bus→Y MDRout和Yin有效，操作数→Y

王道考研/CSKAQYAN.COM

数据通路-CPU内部单总线方式



1. 寄存器之间数据传送

比如把PC内容送至MAR，实现传送操作的流程及控制信号为：
 (PC)→Bus PCout有效，PC内容送总线
 Bus→MAR MARin有效，总线内容送MAR

2. 主存与CPU之间的数据传送

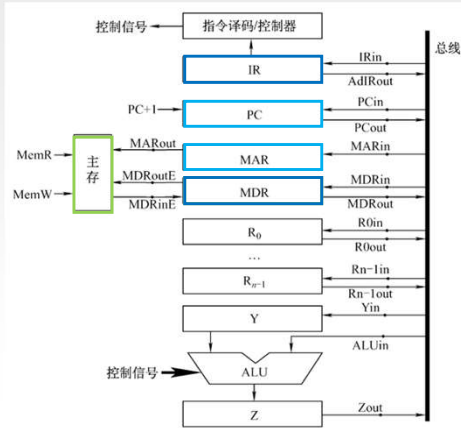
比如CPU从主存读取指令，实现传送操作的流程及控制信号为：
 (PC)→Bus→MAR PCout和MARin有效，现行指令地址→MAR
 1→R CU发读命令(通过控制总线发出，图中未画出)
 MEM(MAR)→MDR MDRin有效
 MDR→Bus→IR MDRout和IRin有效，现行指令→IR

3. 执行算术或逻辑运算

比如一条加法指令，微操作序列及控制信号为：
 Ad(IR)→Bus→MAR MDRout和MARin有效
 1→R CU发读命令
 MEM(MAR)→数据总线→MDR MDRin有效
 MDR→Bus→Y MDRout和Yin有效，操作数→Y
 (ACC)+(Y)→Z ACCout和ALUin有效，CU向ALU发送加命令
 Z→ACC Zout和ACCin有效，结果→ACC

王道考研/CSKAQYAN.COM

CPU内部单总线方式-例题



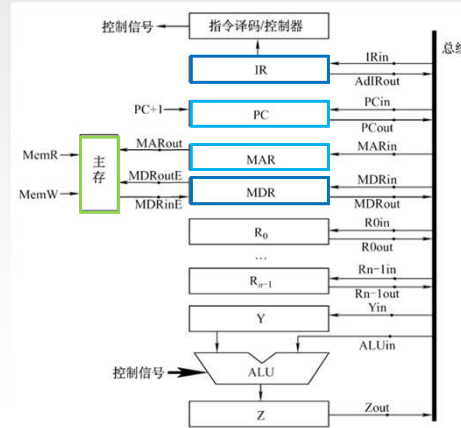
设有如图所示的单总线结构，分析指令
ADD (R0), R1 的指令流程和控制信号。

1. 分析指令功能和指令周期
功能: $((R0))+(R1) \rightarrow (R0)$
取指周期、间址周期、执行周期
2. 写出各阶段的指令流程
取指周期: 公共操作

时序	微操作	有效控制信号
1	(PC)→MAR	PCout, MARin
2	M(MAR)→MDR	MemR, MARout, MDRinE
3	(MDR)→IR	MDRout, IRin
4	指令译码	-
5	(PC)+1→PC	-

王道考研/CSKAQYAN.COM

CPU内部单总线方式-例题



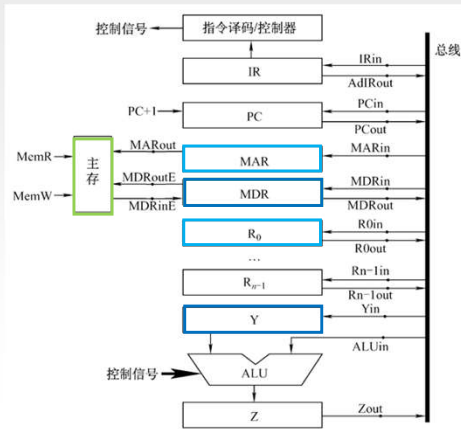
设有如图所示的单总线结构，分析指令
ADD (R0), R1 的指令流程和控制信号。

1. 分析指令功能和指令周期
功能: $((R0))+(R1) \rightarrow (R0)$
取指周期、间址周期、执行周期
2. 写出各阶段的指令流程
取指周期: 公共操作

时序	微操作	有效控制信号
1	(PC)→MAR	PCout, MARin
2	M(MAR)→MDR	MemR, MARout, MDRinE
3	(MDR)→IR	MDRout, IRin
4	指令译码	-

王道考研/CSKAQYAN.COM

CPU内部单总线方式-例题



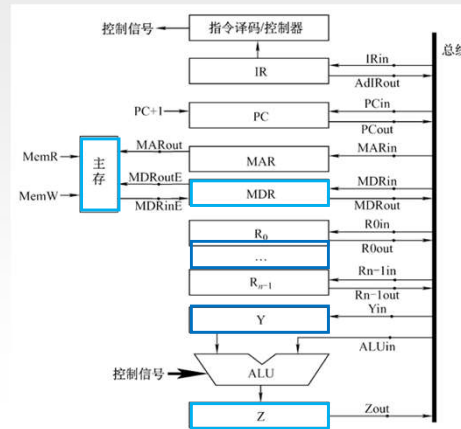
设有如图所示的单总线结构，分析指令
ADD (R0), R1 的指令流程和控制信号。

1. 分析指令功能和指令周期
功能: $((R0))+(R1) \rightarrow (R0)$
取指周期、间址周期、执行周期
2. 写出各阶段的指令流程
间址周期: 完成取数操作, 被加数在主存中, 加数已经放在寄存器R1中。

时序	微操作	有效控制信号
1	(R0)→MAR	R0out, MARin
2	M(MAR)→MDR	MemR, MARout, MDRinE
3	(MDR)→Y	MDRout, Yin

王道考研/CSKAQYAN.COM

CPU内部单总线方式-例题



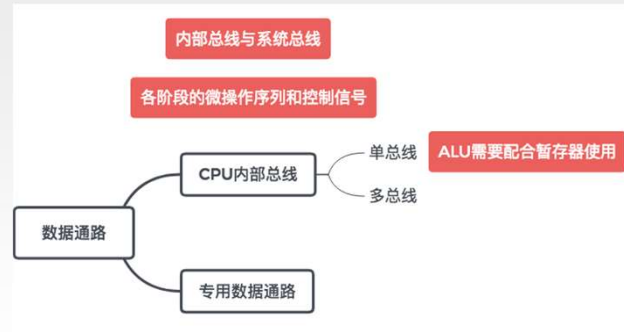
设有如图所示的单总线结构，分析指令
ADD (R0), R1 的指令流程和控制信号。

1. 分析指令功能和指令周期
功能: $((R0))+(R1) \rightarrow (R0)$
取指周期、间址周期、执行周期
2. 写出各阶段的指令流程
执行周期: 完成取数操作, 被加数在主存中, 加数已经放在寄存器R1中。

时序	微操作	有效控制信号
1	(R1)+(Y)→Z	R1out, ALUin, CU向ALU发ADD控制信号
2	(Z)→MDR	Zout, MDRin
3	(MDR)→M(MAR)	MemW, MDRoutE, MARout

王道考研/CSKAQYAN.COM

本节回顾



王道考研/CSKAOYAN.COM

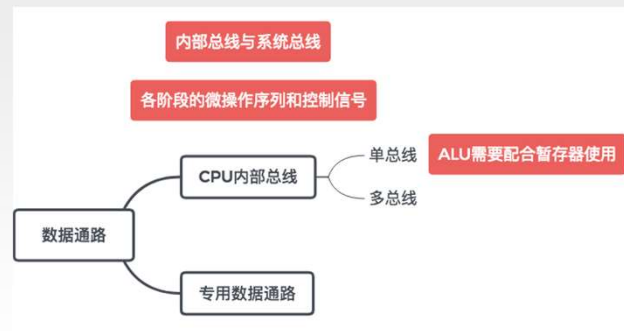
本节内容

中央处理器

数据通路的功能和基本结构
专用通路

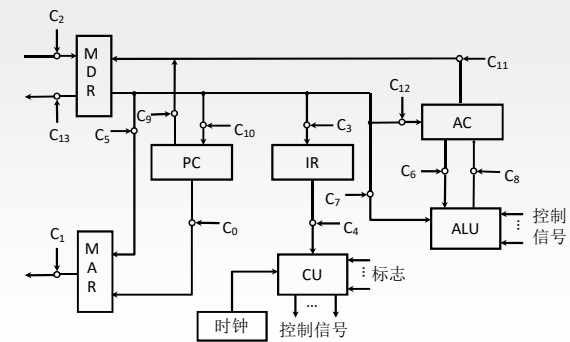
王道考研/CSKAOYAN.COM

上节回顾



王道考研/CSKAOYAN.COM

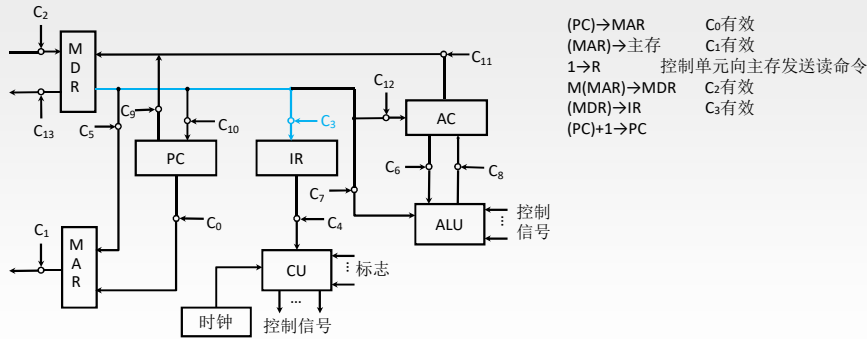
专用数据通路方式



王道考研/CSKAOYAN.COM

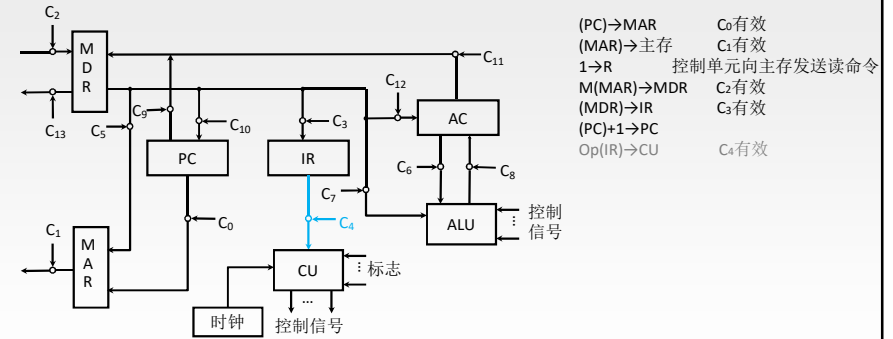


专用数据通路方式-取指周期



王道考研/CSKAQYAN.COM

专用数据通路方式-取指周期



王道考研/CSKAQYAN.COM

专用数据通路方式-例题

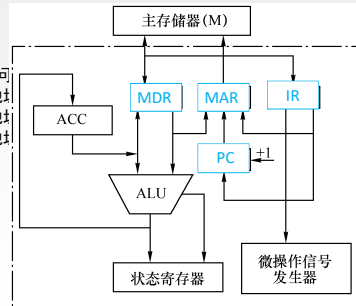
下图是一个简化的CPU与主存连接结构示意图（图中省略了所有的多路选择器）。其中有一个累加寄存器（ACC）、一个状态数据寄存器和其他4个寄存器：主存地址寄存器（MAR）、主存数据寄存器（MDR）、程序寄存器（PC）和指令寄存器（IR），各部件及其之间的连线表示数据通路，箭头表示信息传递方向。

要求：

- (1) 请写出图中a、b、c、d 4个寄存器的名称。
- (2) 简述图中取指令的数据通路。
- (3) 简述数据在运算器和主存之间进行存/取访问。
- (4) 简述完成指令LDA X的数据通路（X为主存地址）。
- (5) 简述完成指令ADD Y的数据通路（Y为主存地址）。
- (6) 简述完成指令STA Z的数据通路（Z为主存地址）。

(1)

d能自动“+1”，是PC
PC内容是地址，送MAR，故c是MAR
b与微操作信号发生器相连，是IR
与主存相连的寄存器是MAR和MDR，c是MAR，
则a是MDR



王道考研/CSKAQYAN.COM

专用数据通路方式-例题

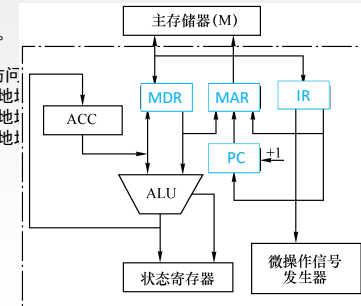
下图是一个简化的CPU与主存连接结构示意图（图中省略了所有的多路选择器）。其中有一个累加寄存器（ACC）、一个状态数据寄存器和其他4个寄存器：主存地址寄存器（MAR）、主存数据寄存器（MDR）、程序寄存器（PC）和指令寄存器（IR），各部件及其之间的连线表示数据通路，箭头表示信息传递方向。

要求：

- (1) 请写出图中a、b、c、d 4个寄存器的名称。
- (2) 简述图中取指令的数据通路。
- (3) 简述数据在运算器和主存之间进行存/取访问。
- (4) 简述完成指令LDA X的数据通路（X为主存地址）。
- (5) 简述完成指令ADD Y的数据通路（Y为主存地址）。
- (6) 简述完成指令STA Z的数据通路（Z为主存地址）。

(2)

(PC) → MAR
M(MAR) → MDR
(MDR) → IR



王道考研/CSKAQYAN.COM

专用数据通路方式-例题

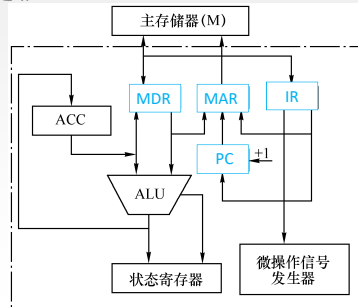
下图是一个简化的CPU与主存连接结构示意图（图中省略了所有的多路选择器）。其中有一个累加寄存器（ACC）、一个状态数据寄存器和其他4个寄存器：主存地址寄存器（MAR）、主存数据寄存器（MDR）、程序寄存器（PC）和指令寄存器（IR），各部件及其之间的连线表示数据通路，箭头表示信息传递方向。

（3）简述数据在运算器和主存之间进行存/取访问的数据通路。

存/取的数据放到ACC中
设数据地址已放入MAR

取：
M(MAR) → MDR
(MDR) → ALU → ACC

存：
(ACC) → MDR
(MDR) → M(MAR)



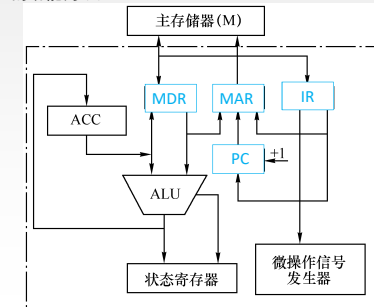
王道考研/CSKAQYAN.COM

专用数据通路方式-例题

下图是一个简化的CPU与主存连接结构示意图（图中省略了所有的多路选择器）。其中有一个累加寄存器（ACC）、一个状态数据寄存器和其他4个寄存器：主存地址寄存器（MAR）、主存数据寄存器（MDR）、程序寄存器（PC）和指令寄存器（IR），各部件及其之间的连线表示数据通路，箭头表示信息传递方向。

（4）简述完成指令LDA X的数据通路（X为主存地址，LDA的功能为 $(X) \rightarrow ACC$ ）。

X → MAR
M(MAR) → MDR
(MDR) → ALU → ACC



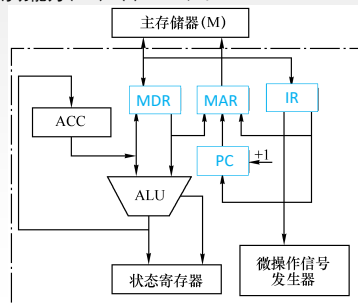
王道考研/CSKAQYAN.COM

专用数据通路方式-例题

下图是一个简化的CPU与主存连接结构示意图（图中省略了所有的多路选择器）。其中有一个累加寄存器（ACC）、一个状态数据寄存器和其他4个寄存器：主存地址寄存器（MAR）、主存数据寄存器（MDR）、程序寄存器（PC）和指令寄存器（IR），各部件及其之间的连线表示数据通路，箭头表示信息传递方向。

（5）简述完成指令ADD Y的数据通路（Y为主存地址，ADD的功能为 $(ACC) + (Y) \rightarrow ACC$ ）。

Y → MAR
M(MAR) → MDR
(MDR) → ALU, (ACC) → ALU
ALU → ACC



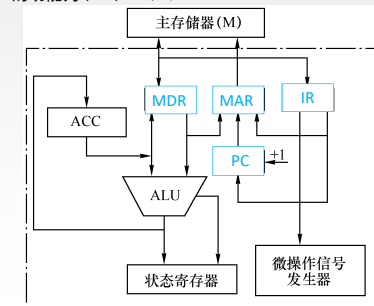
王道考研/CSKAQYAN.COM

专用数据通路方式-例题

下图是一个简化的CPU与主存连接结构示意图（图中省略了所有的多路选择器）。其中有一个累加寄存器（ACC）、一个状态数据寄存器和其他4个寄存器：主存地址寄存器（MAR）、主存数据寄存器（MDR）、程序寄存器（PC）和指令寄存器（IR），各部件及其之间的连线表示数据通路，箭头表示信息传递方向。

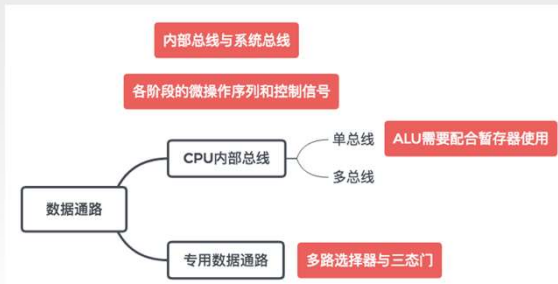
（6）简述完成指令STA Z的数据通路（Z为主存地址，STA的功能为 $(ACC) \rightarrow Z$ ）。

Z → MAR
(ACC) → MDR
(MDR) → M(MAR)



王道考研/CSKAQYAN.COM

本节回顾



涉及的主要操作类型：
寄存器之间的数据传送；
主存与CPU之间的数据传送；
使用ALU进行算术逻辑运算。

基本思路：
利用题目提供的数据通路进行数据传送；
由CU发出的控制信号实现通路的建立。

王道考研/CSKAOYAN.COM

本节内容

中央处理器

指令流水线 基本概念 性能指标

王道考研/CSKAOYAN.COM

本章总览



王道考研/CSKAOYAN.COM

指令流水的定义

一条指令的执行过程可以分成多个阶段（或过程）。
根据计算机的不同，具体的分法也不同。

取指	分析	执行
----	----	----

取指：根据PC内容访问主存储器，取出一条指令送到IR中。

分析：对指令操作码进行译码，按照给定的寻址方式和地址字段中的内容形成操作数的有效地址EA，并从有效地址EA中取出操作数。

执行：根据操作码字段，完成指令规定的功能，即把运算结果写到通用寄存器或主存中。

设取指、分析、执行3个阶段的时间都相等，用 t 表示，按以下几种执行方式分析 n 条指令的执行时间：

1. 顺序执行方式 总耗时 $T = n \times 3t = 3nt$



传统冯·诺依曼机采用顺序执行方式，又称串行执行方式。

优点：控制简单，硬件代价小。

缺点：执行指令的速度较慢，在任何时刻，处理机中只有一条指令在执行，各功能部件的利用率很低。

王道考研/CSKAOYAN.COM

指令流水的定义

1. 顺序执行方式 总耗时 $T = n \times 3t = 3nt$

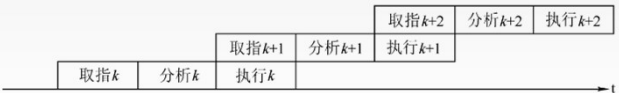


传统冯·诺依曼机采用顺序执行方式，又称串行执行方式。

优点：控制简单，硬件代价小。

缺点：执行指令的速度较慢，在任何时刻，处理机中只有一条指令在执行，各功能部件的利用率很低。

2. 一次重叠执行方式 总耗时 $T = 3t + (n-1) \times 2t = (1+2n)t$



优点：程序的执行时间缩短了1/3，各功能部件的利用率明显提高。

缺点：需要付出硬件上较大开销的代价，控制过程也比顺序执行复杂了。

王道考研/CSKAQYAN.COM

指令流水的定义

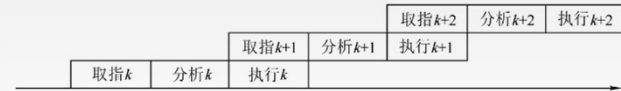
1. 顺序执行方式 总耗时 $T = n \times 3t = 3nt$



传统冯·诺依曼机采用顺序执行方式，又称串行执行方式。优点：控制简单，硬件代价小。

缺点：执行指令的速度较慢，在任何时刻，处理机中只有一条指令在执行，各功能部件的利用率很低。

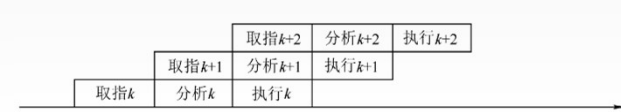
2. 一次重叠执行方式 总耗时 $T = 3t + (n-1) \times 2t = (1+2n)t$



优点：程序的执行时间缩短了1/3，各功能部件的利用率明显提高。

缺点：需要付出硬件上较大开销的代价，控制过程也比顺序执行复杂了。

3. 二次重叠执行方式 总耗时 $T = 3t + (n-1) \times t = (2+n)t$



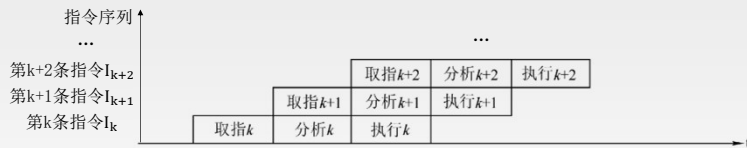
与顺序执行方式相比，指令的执行时间缩短近2/3。这是一种理想的指令执行方式，在正常情况下，处理机中同时有3条指令在执行。

注：也可以把每条指令的执行过程分成4个或5个阶段，分成5个阶段是比较常见的做法。

王道考研/CSKAQYAN.COM

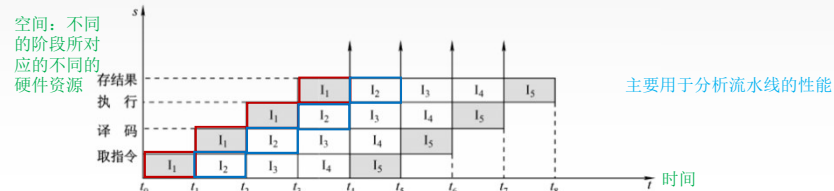
流水线的表示方法

1. 指令执行过程图



主要用于分析指令执行过程以及影响流水线的因素(见下一个视频)

2. 时空图



主要用于分析流水线的性能

王道考研/CSKAQYAN.COM

流水线的性能指标

1. 吞吐率
2. 加速比
3. 效率

王道考研/CSKAQYAN.COM

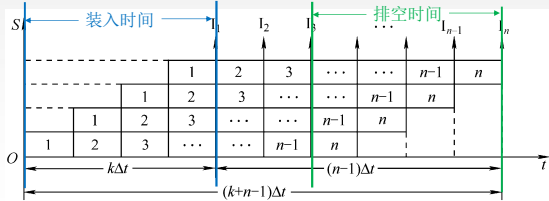
流水线的性能指标

1. 吞吐量 吞吐量是指在单位时间内流水线所完成的任务数量，或是输出结果的数量。

设任务数为 n ；处理完成 n 个任务所用的时间为 T_k

则计算流水线吞吐量（TP）的最基本的公式为 $TP = \frac{n}{T_k}$

理想情况下，流水线的时空图如下：当连续输入的任务 $n \rightarrow \infty$ 时，得最大吞吐率为 $TP_{\max} = 1/\Delta t$ 。



$$T_k = (k+n-1)\Delta t$$

流水线的实际吞吐率为

$$TP = \frac{n}{(k+n-1)\Delta t}$$

一条指令的执行分为 k 个阶段，每个阶段耗时 Δt ，一般取 Δt = 一个时钟周期

王道考研/CSKAQYAN.COM

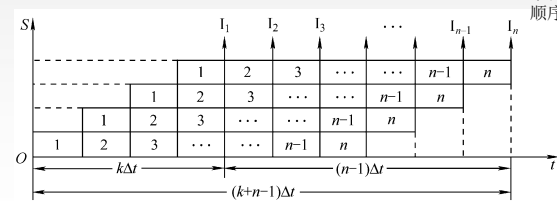
流水线的性能指标

2. 加速比 完成同样一批任务，不使用流水线所用的时间与使用流水线所用的时间之比。

设 T_0 表示不使用流水线时的执行时间，即顺序执行所用的时间； T_k 表示使用流水线时的执行时间

则计算流水线加速比（S）的基本公式为 $S = \frac{T_0}{T_k}$ 当连续输入的任务 $n \rightarrow \infty$ 时，最大加速比为 $S_{\max} = k$ 。

理想情况下，流水线的时空图如下：



单独完成一个任务耗时为 $k\Delta t$ ，则顺序完成 n 个任务耗时 $T_0 = nk\Delta t$

$$T_k = (k+n-1)\Delta t$$

实际加速比为

$$S = \frac{kn\Delta t}{(k+n-1)\Delta t} = \frac{kn}{k+n-1}$$

一条指令的执行分为 k 个阶段，每个阶段耗时 Δt ，一般取 Δt = 一个时钟周期

王道考研/CSKAQYAN.COM

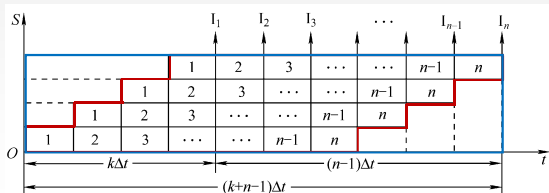
流水线的性能指标

3. 效率 流水线的设备利用率称为流水线的效率。

在时空图上，流水线的效率定义为完成 n 个任务占用的时空区有效面积与 n 个任务所用的时间与 k 个流水段所围成的时空区总面积之比。

则流水线效率（E）的一般公式为 $E = \frac{n \text{个任务占用} k \text{个流水段所围成的时空区有效面积}}{n \text{个任务所用的时间与} k \text{个流水段所围成的时空区总面积}} = \frac{T_0}{kT_k}$

理想情况下，流水线的时空图如下：

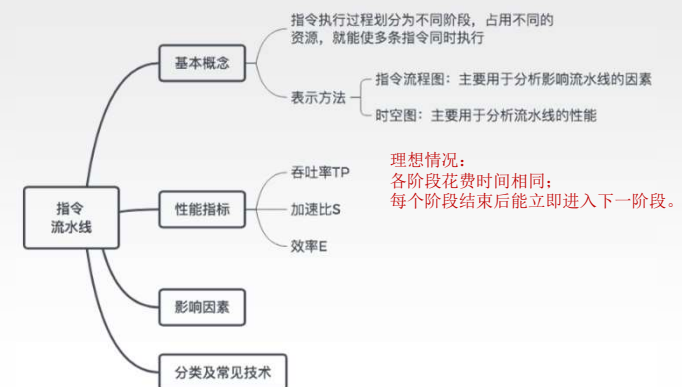


当连续输入的任务 $n \rightarrow \infty$ 时，最高效率为 $E_{\max} = 1$ 。

一条指令的执行分为 k 个阶段，每个阶段耗时 Δt ，一般取 Δt = 一个时钟周期

王道考研/CSKAQYAN.COM

本节回顾



王道考研/CSKAQYAN.COM

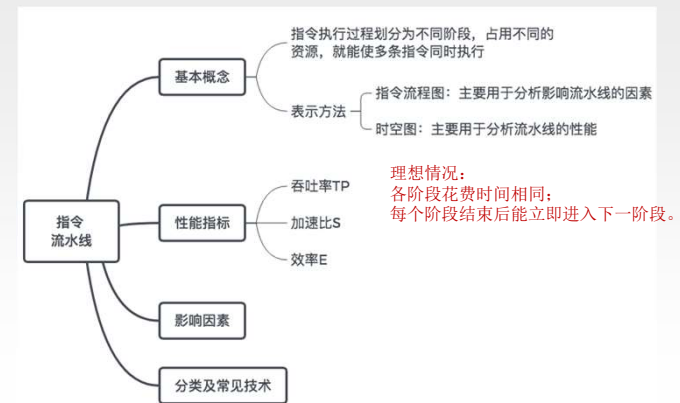
本节内容

中央处理器

指令流水线
影响因素
分类

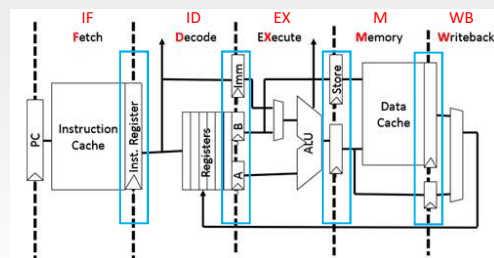
王道考研/CSKAQYAN.COM

本节总览



王道考研/CSKAQYAN.COM

机器周期的设置



各部件实际耗时： IF 100ns, ID 80ns, EX 70ns, M 50ns, WB 50ns

为方便流水线的设计，将每个阶段的耗时取成一样，以最长耗时为准。即此处应将机器周期设置为100ns。

王道考研/CSKAQYAN.COM

影响流水线的因素

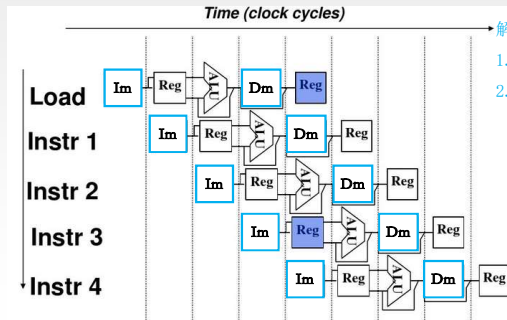
1. 结构相关（资源冲突）
2. 数据相关（数据冲突）
3. 控制相关（控制冲突）

王道考研/CSKAQYAN.COM

影响流水线的因素

1. 结构相关（资源冲突）

由于多条指令在同一时刻争用同一资源而形成的冲突称为结构相关。



解决办法:

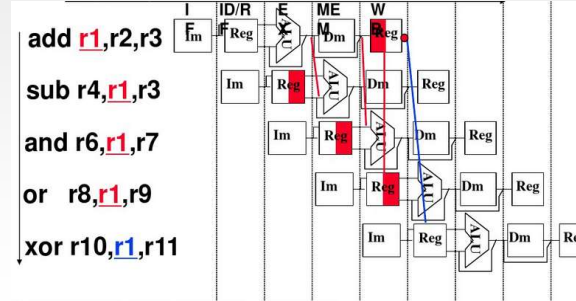
1. 后一相关指令暂停一周期
2. 资源重复配置:
数据存储器+指令存储器

王道考研/CSKAQYAN.COM

影响流水线的因素

2. 数据相关（数据冲突）

数据相关指在一个程序中，存在必须等前一条指令执行完才能执行后一条指令的情况，则这两条指令即为数据相关。



解决办法:

1. 把遇到数据相关的指令及其后续指令都暂停一至几个时钟周期，直到数据相关问题消失后再继续执行。可分为硬件阻塞(stall)和软件插入“NOP”两种方法。
2. 数据旁路技术。
3. 编译优化：通过编译器调整指令顺序来解决数据相关。

王道考研/CSKAQYAN.COM

影响流水线的因素

2. 数据相关（数据冲突）

例题. 假设某指令流水线采用“按序发射，按序完成”方式，没有采用转发技术处理数据相关，并且同一寄存器的读和写操作不能在同一个时钟周期内进行。若高级语言程序中某赋值语句为 $x=a+b$ ， x 、 a 和 b 均为int型变量，它们的存储单元地址分别表示为 $[x]$ 、 $[a]$ 和 $[b]$ 。该语句对应的指令序列及其在指令流中的执行过程如下图所示。

I1 LOAD R1, [a] ([a]) → R1
 I2 LOAD R2, [b] ([b]) → R2
 I3 ADD R1, R2 (R1) + (R2) → R2
 I4 STORE R2, [x] (R2) → [x]

I3与I1和I2存在数据相关，

则这4条指令执行过程中I3的ID段和I4的IF段被阻塞的原因各是什么？I4和I3存在数据相关。

指令	1	2	3	4	5	6	7	8	9	10	11	12	13	14
I ₁	IF	ID	EX	M	WB									
I ₂		IF	ID	EX	M	WB								
I ₃			IF				ID	EX	M	WB				
I ₄							IF				ID	EX	M	WB

王道考研/CSKAQYAN.COM

影响流水线的因素

2. 数据相关（数据冲突）

数据的基本操作：读（R）、写（W）

冲突的基本类型：RAW、WAR、WAW

RAW

注：“按序发射，按序完成”时，只可能出现RAW相关。

I1: ADD R5, R2, R4; (R2)+(R4) → R5

I2: ADD R4, R5, R3; (R5)+(R3) → R4

WAR

I1: STA M, R2; (R2) → M, M为主存单元

乱序发射，编写程序的时候希望I1在I2前完成，

I2: ADD R2, R4, R5; (R4)+(R5) → R2

但优化手段导致I2在I1前发射。

WRW

I1: MUL R3, R2, R1; (R2)*(R1) → R3

存在多个功能部件时，后一条指

I2: SUB R3, R4, R5; (R4)-(R5) → R3

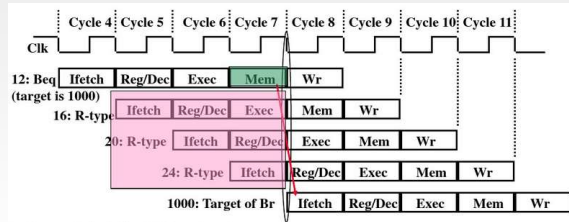
令可能比前一条指令先完成。

王道考研/CSKAQYAN.COM

影响流水线的因素

3. 控制相关（控制冲突）

当流水线遇到转移指令和其他改变PC值的指令而造成断流时，会引起控制相关。

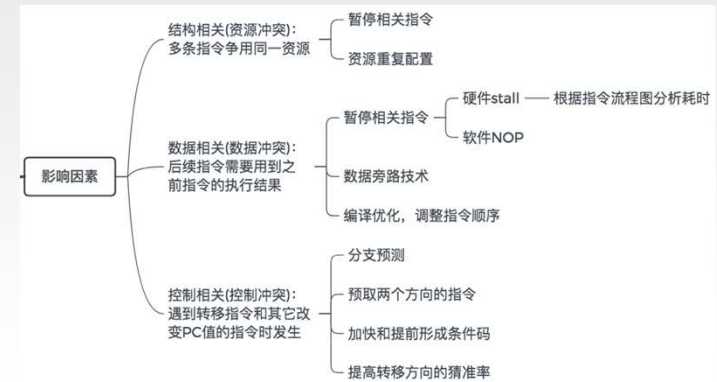


解决办法：

1. 尽早判别转移是否发生，尽早生成转移目标地址
2. 预取转移成功和不成功两个控制流方向上的目标指令
3. 加快和提前形成条件码
4. 提高转移方向的猜准确率

王道考研/CSKAQYAN.COM

影响流水线的因素



王道考研/CSKAQYAN.COM

流水线的分类

1. 部件功能级、处理机级和处理机间级流水线

根据流水线使用的级别的不同，流水线可分为部件功能级流水线、处理机级流水线和处理机间流水线。
部件功能级流水就是将复杂的算术逻辑运算组成流水线工作方式。例如，可将浮点加法操作分成求阶差、对阶、尾数相加以及结果规格化等4个子过程。

处理机级流水是把一条指令解释过程分成多个子过程，如前面提到的取指、译码、执行、访存及写回5个子过程。

处理机间流水是一种宏流水，其中每一个处理机完成某一专门任务，各个处理机所得到的结果需存放在与下一个处理机所共享的存储器中。

2. 单功能流水线和多功能流水线

按流水线可以完成的功能，流水线可分为单功能流水线和多功能流水线。

单功能流水线指只能实现一种固定的专门功能的流水线；

多功能流水线指通过各段间的不同连接方式可以同时或不同时地实现多种功能的流水线。

王道考研/CSKAQYAN.COM

流水线的分类

3. 动态流水线和静态流水线

按同一时间内各段之间的连接方式，流水线可分为静态流水线和动态流水线。

静态流水线指在同一时间内，流水线的各段只能按同一种功能的连接方式工作。

动态流水线指在同一时间内，当某些段正在实现某种运算时，另一些段却正在进行另一种运算。这样对提高流水线的效率很有好处，但会使流水线控制变得很复杂。

4. 线性流水线和非线性流水线

按流水线的各个功能段之间是否有反馈信号，流水线可分为线性流水线与非线性流水线。

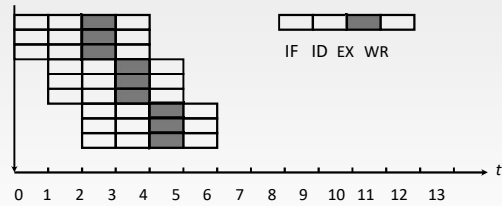
线性流水线中，从输入到输出，每个功能段只允许经过一次，不存在反馈回路。

非线性流水线存在反馈回路，从输入到输出过程中，某些功能段将数次通过流水线，这种流水线适合进行线性递归的运算。

王道考研/CSKAQYAN.COM

流水线的多发技术

1. 超标量技术



每个时钟周期内可 并发多条独立指令

要配置多个功能部件

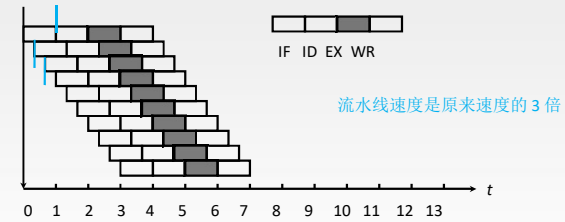
不能调整 指令的 执行顺序

通过编译优化技术，把可并行执行的指令搭配起来

王道考研/CSKAQYAN.COM

流水线的多发技术

2. 超流水技术



流水线速度是原来速度的 3 倍

在一个时钟周期内 再分段 (3 段)

在一个时钟周期内 一个功能部件使用多次 (3 次)

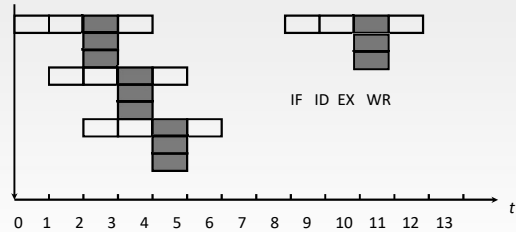
不能调整 指令的 执行顺序

靠编译程序解决优化问题

王道考研/CSKAQYAN.COM

流水线的多发技术

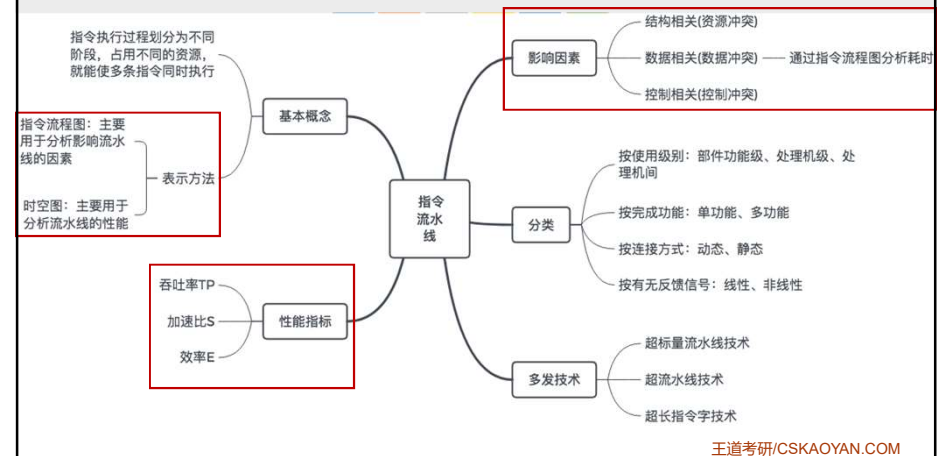
3. 超长指令字



由 编译程序挖掘 出指令间 潜在的 并行性，
将 多条 能 并行操作 的指令组合成 一条
具有 多个操作码字段 的 超长指令字 (可达几百位)
采用 多个处理部件

王道考研/CSKAQYAN.COM

本节回顾



王道考研/CSKAQYAN.COM