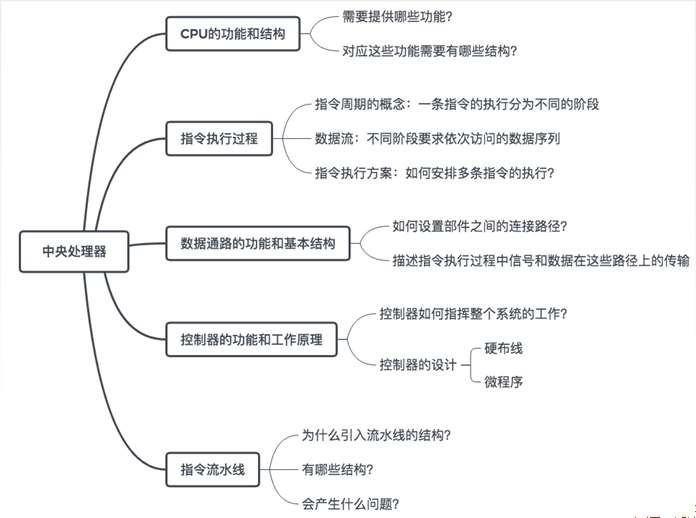
王道考研



# 

|  |
| --- |
| 王道考研——组成原理  [WWW.CSKAOYAN.COM](http://WWW.CSKAOYAN.COM/)  第五章 中央处理器 |

CPU的功能和

王道考研/CSKAOYAN.COM

基本结构

中央处理器

本节内容

|  |
| --- |
| 本章总览  /CSKAOYAN.COM |

CPU的功能

运算器

主存

输入设备

辅助

王道考研/CSKAOYAN.COM

主机 外设

存储器

输出设备

I/O设备

储器

控制器

CPU

控制器

CPU

从而控制这些部件按指令的要求进行动作。

1. 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。
2. 数据加工。对数据进行算术和逻辑运算。
3. 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/CSKAOYAN.COM

1. 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。
2. 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，

运算器

CPU的功能

王道考研/CSKAOYAN.COM

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)。

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。

CPU

协调并控制计算机各部件执行程序的指令序列，

控制器

对数据进行加工

运算器

运算器和控制器的功能



A

B

R□ R□ R□

R□

AH

AL

BH

BL

CH

CL

DH

DL

SP

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/CSKAOYAN.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。

ALU

**.**



A

B

R□ R□ R□

R□

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。

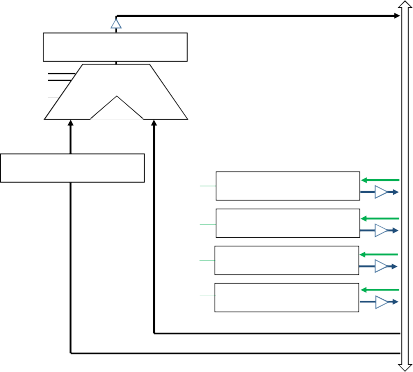
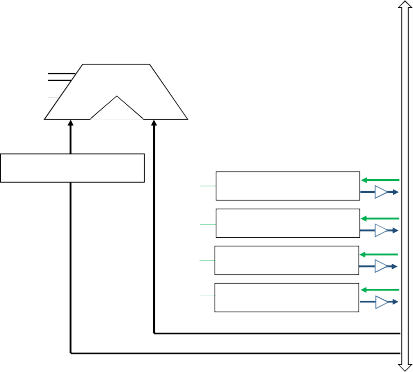
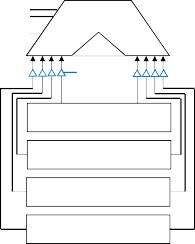
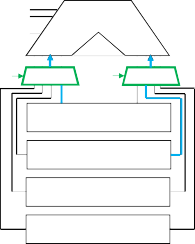
王道考研/CSKAOYAN.COM

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。

**.** ALU



|  |
| --- |
| 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  **.** ALU 地址。 |
| A B  00 01 如果直接用导线连接，相当于多个寄  C□ MUX C□ MUX 存器同时并且一直向ALU传输数据  R□ 解决方法1.使用多路选择器 |
| R 根据控制信号选择一路输出  □ 解决方法2.使用三态门  R□ 可以控制每一路是否输出  R□  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/CSKAOYAN.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。  CPU内部单总线方式：将所有寄存器的输入端 2.通用寄存器组：如AX、BX、CX、DX、SP等，用于  和输出端都连接到一条公共的通路上。 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  **.** ALU 地址。 |
| A B  R0out 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  R□ 解决方法1.使用多路选择器 |
| R 根据控制信号选择一路输出  □ 解决方法2.使用三态门  R□ 可以控制每一路是否输出  R 如：R0out为1时R□中的数据输出到A端，  □ R0out为0时R□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/CSKAOYAN.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 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  和输出端都连接到一条公共的通路上。 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  **.** ALU 地址。 |
| A B 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R□，最后结果存回R□，  暂存寄存器 那么从主存中取来的操作数直接放入暂存器，就不会  R0in R□ 破坏运算前R的内容。  R0out □  R1in R□  R1out  R2in R□  R2out |
| R3in R□  R3out 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/CSKAOYAN.COM |

|  |
| --- |
| 运算器的基本结构  CPU内部总线 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  暂存寄存器 增加一些功能 存放操作数（包括源操作数、目的操作数及中间结果）  . —>移位寄存器 和各种地址信息等。SP是堆栈指针，用于指示栈顶的  **.** ALU —>累加寄存器 地址。  A B 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R□，最后结果存回R□， |
| 暂存寄存器 那么从主存中取来的操作数直接放入暂存器，就不会  R0in R□ 破坏运算前R的内容。  R0out □  R1in R□  R1out  R2in R□  R2out |
| R3in R□  R3out 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/CSKAOYAN.COM |

暂存寄存器

暂存寄存器

移位寄存器

... ALU PSW

A B

ACCin ACC

ACCout

暂存寄存器

R0in R□

R0ou

R1in R□

R1ou

R2in R□

R2ou

R3in R□

R3ou

暂存寄存器

暂存寄存器

... ALU

A B

ACCin

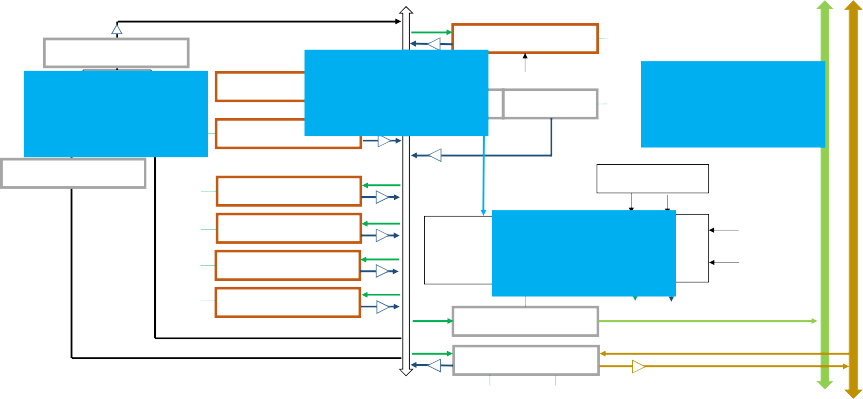
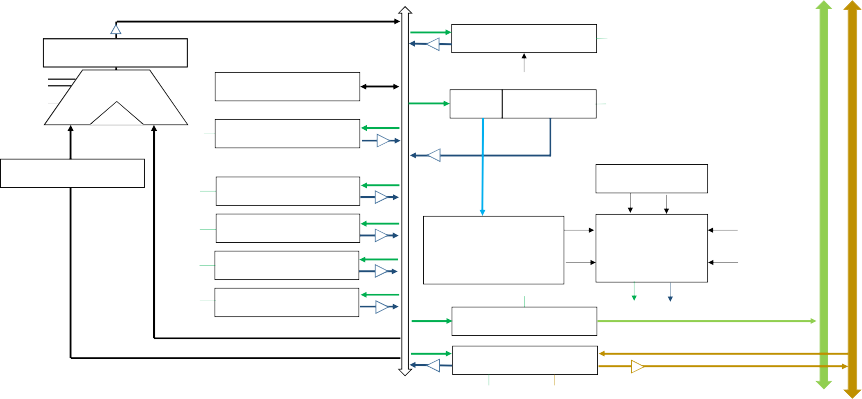
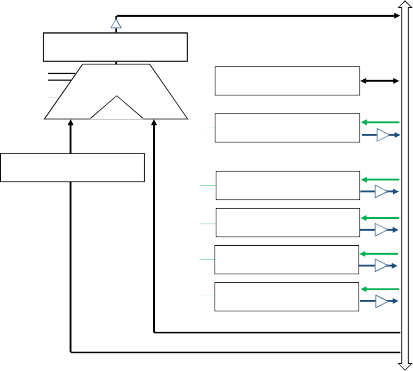
码器ID ... 微操作信

发生器

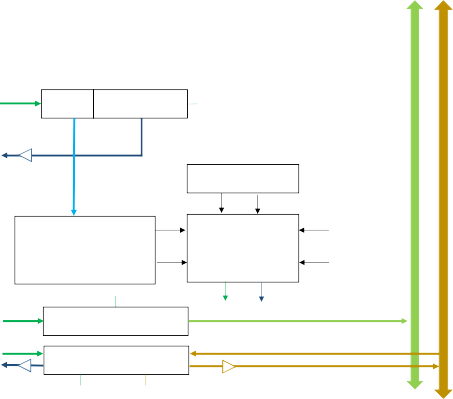
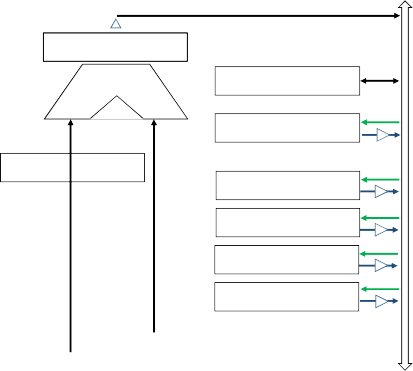
…

PCout

OP



|  |
| --- |
| 运算器的基本结构  CPU内部总线 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  移位寄存器 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  **.** ALU PSW 地 址 。  A B 3.暂存寄存器：用于暂存从主存读来的数据，这个数ACCin ACC 据不能存放在通用寄存器中，否则会破坏其原有内容。  ACCout 4.累加寄存器：它是一个通用寄存器，用于暂时存放  暂存寄存器 ALU运算的结果信息，用于实现加法运算。  R0in R□ 5.程序状态字寄存器：保留由算术逻辑运算指令或测R0out 试指令的结果而建立的各种状态信息，如溢出标志  R1in R□ （OP）、符号标志（SF）、零标志（ZF）、进位标志  R1out （CF）等。PSW中的这些位参与并决定微操作的形成。  R2in R□ 6.移位器：对运算结果进行移位运算。R2out  R3in R□ 7.计数器：控制乘除运算的操作步数。  R3out 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/CSKAOYAN.COM |



控制器的基本结构

CPU内部总线

1.程序计数器：用于指出下一条指令在主存中的存放

PCin

数据总线

地址总线

PCout

PC

+1

OP Ad

IR

IRin

AdIRout

时序系统

…

指令译码器ID

MARin MAR

**.**.

微操作信号

发生器 来自PSW

…

**.**.

标志

7.存储器数据寄存器：用于存放向主存写入的信息或从主存中读出的信息。

MDR

MDRout MDRin

MDRinE

MDRoutE

王道考研/CSKAOYAN.COM

t t

t

存储逻辑型两种。

5.时序系统：用于产生各种时序信号，它们都是由统 一时钟（CLOCK）分频得到。

6.存储器地址寄存器：用于存放所要访问的主存单元的地址。

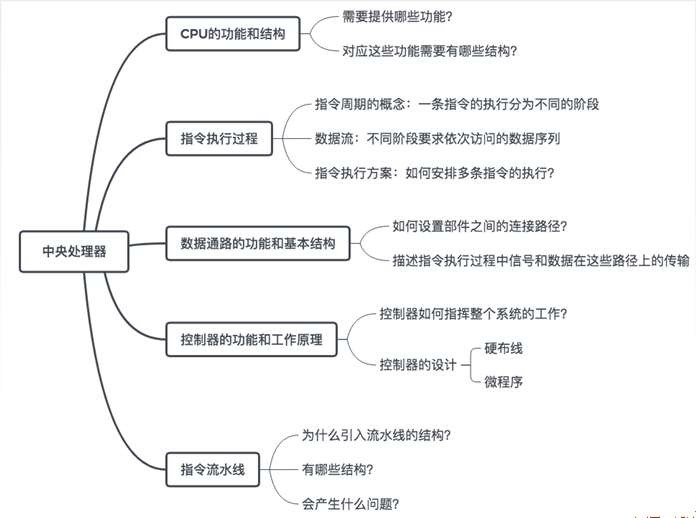
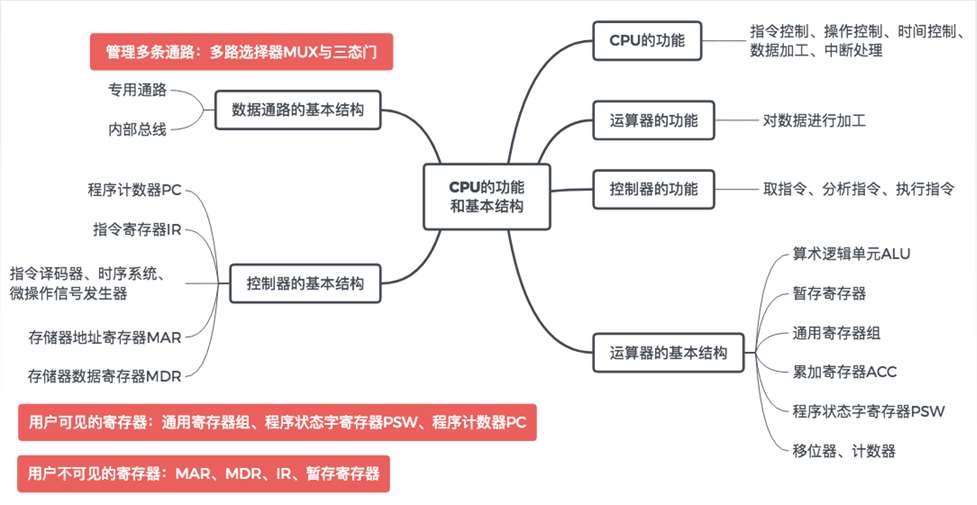
地址。CPU就是根据PC的内容去主存中取指令的。因程序中指令（通常）是顺序执行的，所以PC有自增功能。

1. 指令寄存器：用于保存当前正在执行的那条指令。
2. 指令译码器：仅对操作码字段进行译码，向控制器 提供特定的操作信号。
3. 微操作信号发生器：根据IR的内容（指令）、PSW 的内容（状态信息）及时序信号，产生控制整个计算机系统所需的各种控制信号，其结构有组合逻辑型和 t

|  |
| --- |
| CPU的基本结构  用户可见的寄存器(可编程) CPU内部总线 数据总线地址总线  PC PCin  移位寄存器 PCout +1  **.** ALU PSW  A B OP Ad IRin  ACCin ACC IR  ACCout  暂存寄存器 AdIRout 时序系统  R0in R□ …  R0out  R1in R□  R1out 指令译码器ID **.**. 微操作信号 **.**. 标志  R2in R□ 发生器 来自PSW  R2out …  R3in R□ MARin  R3out MAR  MDR  MDRout MDRin MDRinE MDRoutE  王道考研/CSKAOYAN.COM |

|  |
| --- |
| CPU的基本结构  用户可见的寄存器(可编程) CPU内部总线 数据总线地址总线  PC PCin  移位寄存器 +1  PSW 寄 存器 Ad IR 中断系统  ALU in  ACC IR  ACCout  暂存寄存器 AdIRout 时序系统  R0in R□ …  R0out  R1in R□ 号 **.** 标 志  R1out 指令译 CU . 来自PSW  R2in R□  R2out  R3in R□ MARin  R3out MAR  MDR  MDRout MDRin MDRinE MDRoutE  王道考研/CSKAOYAN.COM |

王道考研



CPU的基本结构

CPU

王道考研/CSKAOYAN.COM

CU

中断系统

寄存器

ALU

|  |
| --- |
| 本节回顾  王道考研/CSKAOYAN.COM |

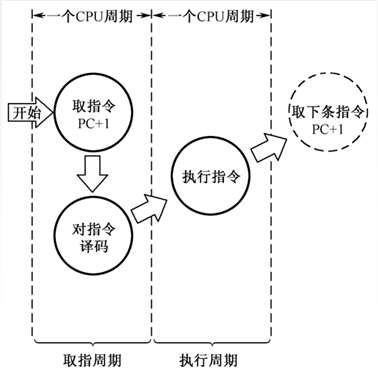
本节内容

中央处理器

指令执行过程

王道考研/CSKAOYAN.COM

|  |
| --- |
| 本章总览  /CSKAOYAN.COM |



指令周期流程

触发器，可以存放1个二进制位。

取指周期

FE

D

IND

D

EX

D

INT

D

有间址吗？

否

执行周期

是

CLK

间址周期

1 FE

1 IND

1 EX

1 INT

有中断吗？

否

是

中断周期

四个工作周期都有CPU访存操作，只是访存的目的不同。取指周期是为了取指令，间址周期是为了取有效地址，执行周期是为了取操作数，中断周期是为了保存程序断点。

王道考研/CSKAOYAN.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/CSKAOYAN.COM |



空指令NOP

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。 取指周期

加法指令ADD

指令周期

取指周期 执行周期指令周期

乘法指令MUL

取指周期

执行周期

指令周期

王道考研/CSKAOYAN.COM

指令周期

中断周期

执行周期

执行周期

间址周期指令周期

间址周期

取指周期

取指周期

具有间接寻址的指令

带有中断周期的指令

指令周期流程

取指周期

有间址吗？

否

是

间址周期

王道考研/CSKAOYAN.COM

中断周期

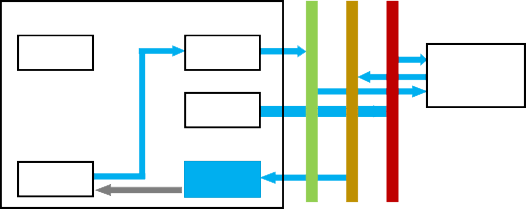
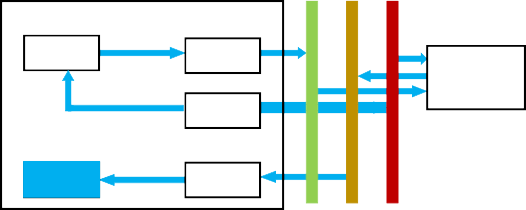
是

有中断吗？

否

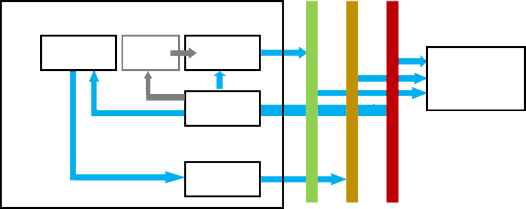
执行周期

|  |  |  |  |
| --- | --- | --- | --- |
| 取指周期： 1 | 0 | 0 | 0 |
| 间址周期： 0 | 1 | 0 | 0 |
| 执行周期： 0 | 0 | 1 | 0 |
| 中断周期： 0 | 0 | 0 | 1 |



IR

MDR

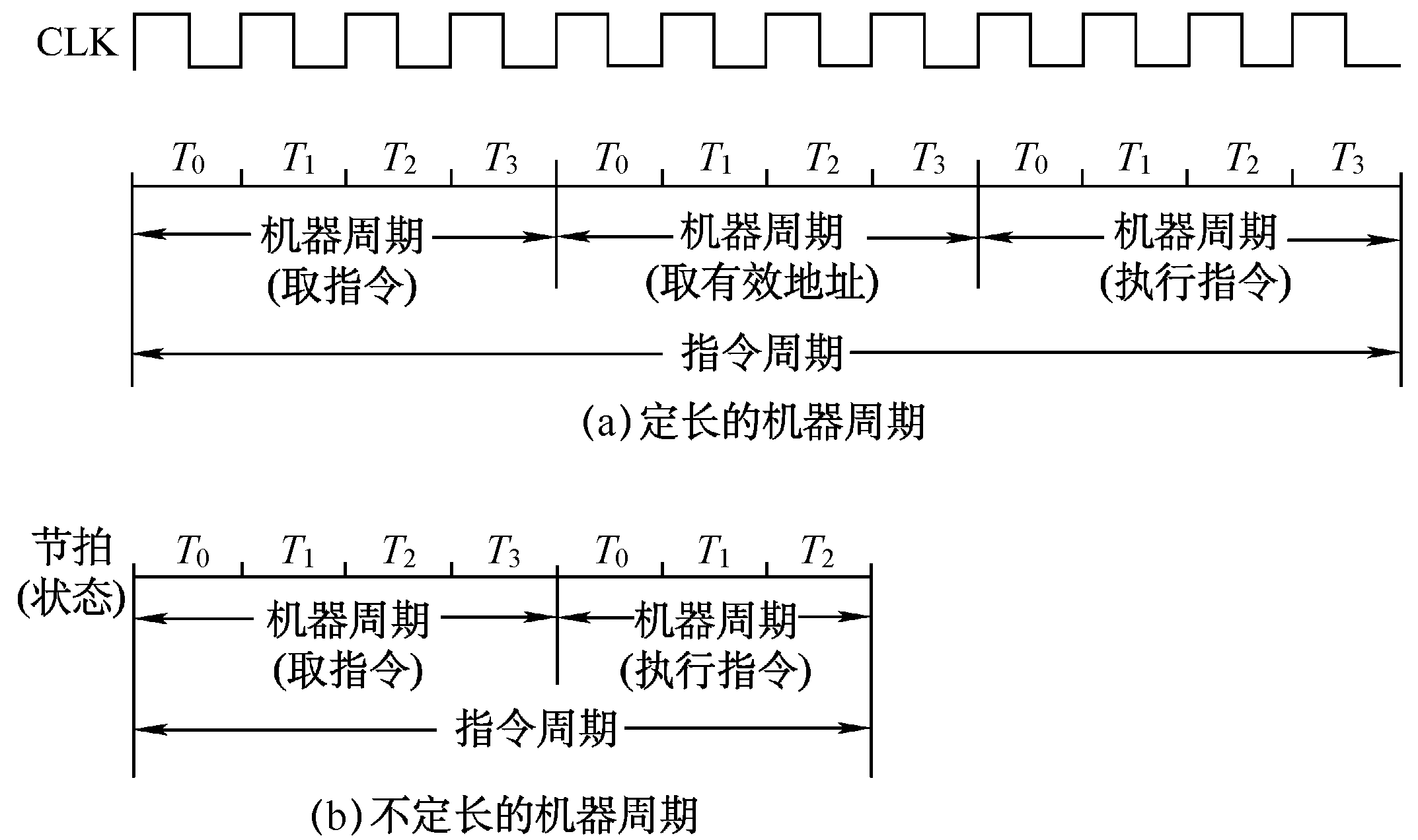
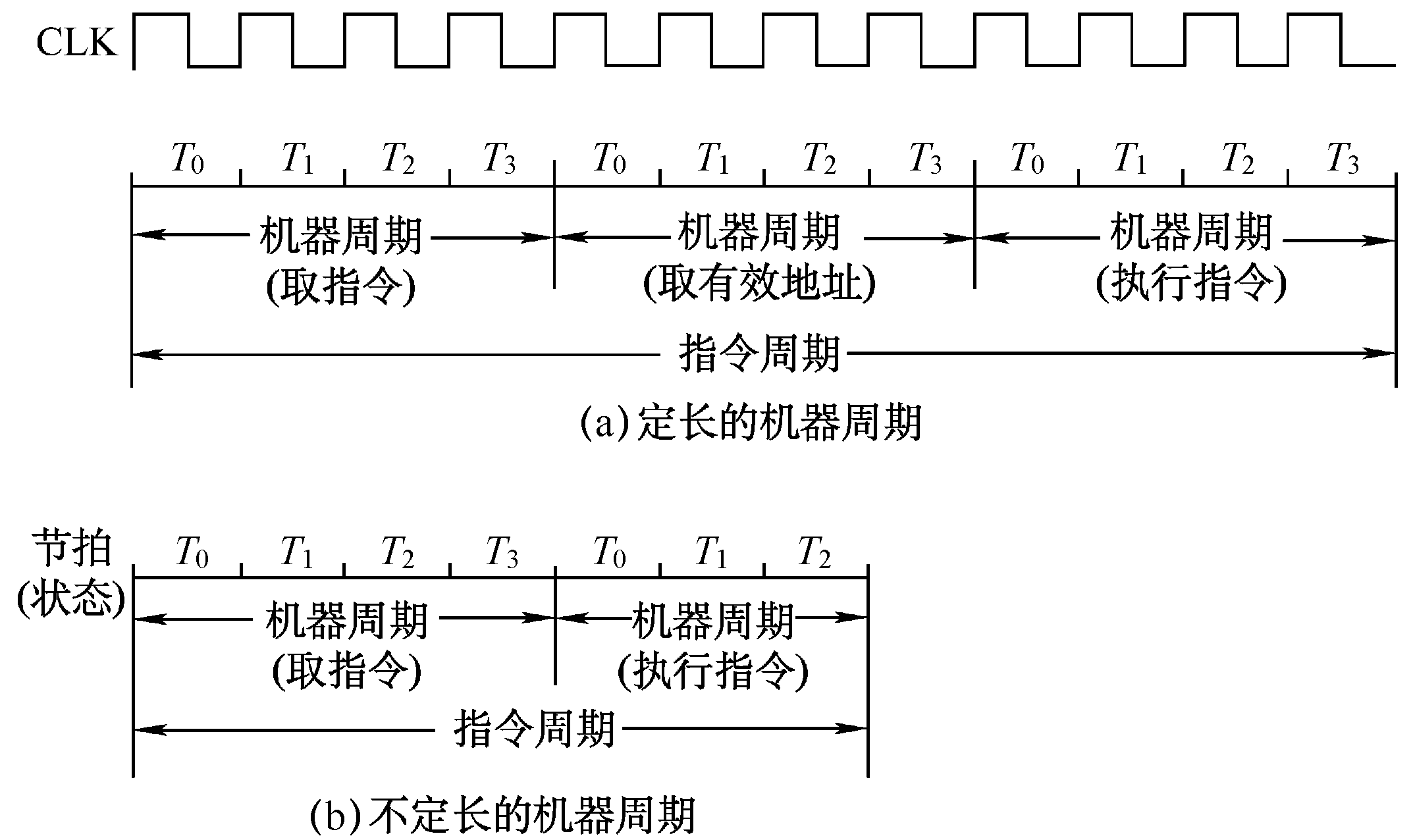


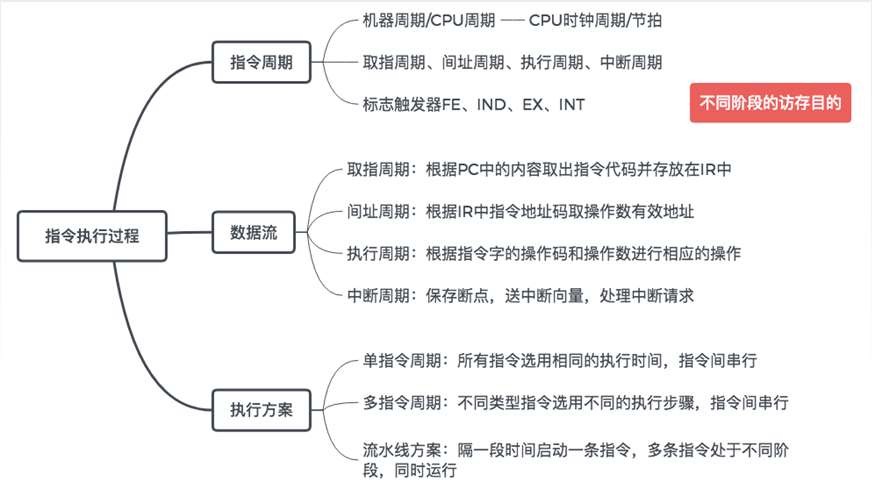
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令周期的数据流-取指周期  地 数 控 1.当前指令地址送至存储器地址寄存器，  址 据 制 记做：(PC) → MAR  CPU 总 总 总 2. CU发出控制信号，经控制总线传到主  线 线 线 存，这里是读信号，记做：1 → R | | | | | | |
| PC |  | MAR CU |  |  | 存储器 | 1. 将MAR所指主存中的内容经数据总线送入MDR，记做：M(MAR) → MDR 2. 将MDR中的内容(此时是指令)送入IR， 记做：(MDR) → IR |
| IR | +1 | MDR |  |  | 5. CU发出控制信号，形成下一条指令地  址，记做：(PC)+1 → PC | |
|  | | | | | | |
| 王道考研/CSKAOYAN.COM | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CPU |  | 指令周期的数据流-间址周期  地 数 控 1.将指令的地址码送入MAR，  址 据 制 记做：Ad(IR) → MAR  总 总 总 或Ad(MDR) → MAR  线 线 线 2. CU发出控制信号，启动主存做读操作，  记做：1 → R | | | |
| PC |  | MAR  CU |  |  |  | 3.将MAR所指主存中的内容经数据总线送入MDR，记做：M(MAR) → MDR  存储器 4.将有效地址送至指令的地址码字段， 记做：MDR→ Ad(IR) |
| IR |  | MDR |  |  |  |  |
|  | | | | | | |
| 王道考研/CSKAOYAN.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/CSKAOYAN.COM |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令周期的数据流-中断周期  中断：暂停当前任务去完成其他任务。  地 数 控 为了能够恢复当前任务，需要保存断点。  址 据 制 一般使用堆栈来保存断点，这里用SP表示  CPU 总 总 总 栈顶地址，假设SP指向栈顶元素，进栈操  线 线 线 作是先修改指针，后存入数据。 | | | | | | |
| PC | SP  -1 | MAR  CU |  |  | 存储器 | 1. CU控制将SP减1，修改后的地址送入MAR   记做： (SP)-1 → SP，(SP) → MAR  本质上是将断点存入某个存储单元，假设其地址为a，故可记做：a → MAR   1. CU发出控制信号，启动主存做写操作， |
| MDR | | |  |  | 记做：1 → W  3.将断点(PC内容)送入MDR， | |
| 记做：(PC) → MDR  4. CU控制将中断服务程序的入口地址  (由向量地址形成部件产生)送入PC， 记做：向量地址→ PC | | | | | | |
| 王道考研/CSKAOYAN.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/CSKAOYAN.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 。  方案3．流水线方案 需要更复杂的硬件设计。  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/CSKAOYAN.COM |

|  |
| --- |
| 本节回顾  王道考研/CSKAOYAN.COM |

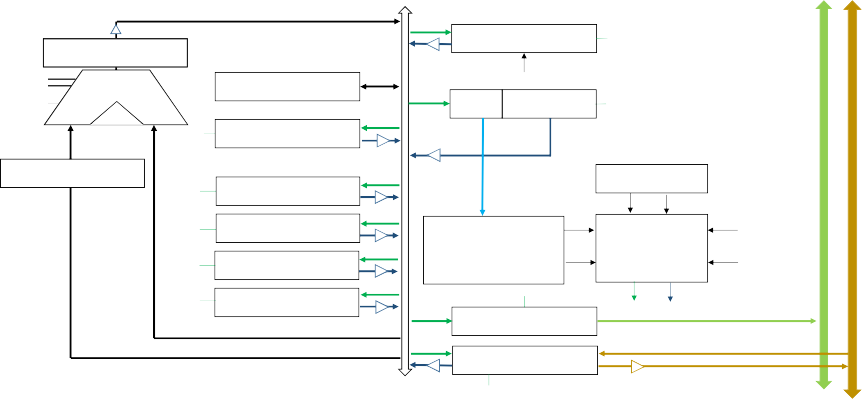
王道考研/CSKAOYAN.COM

数据通路的功

能和基本结构单总线

中央处理器

本节内容



王道考研

暂存寄存器

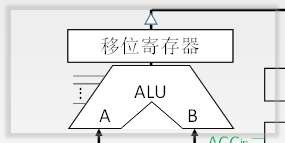
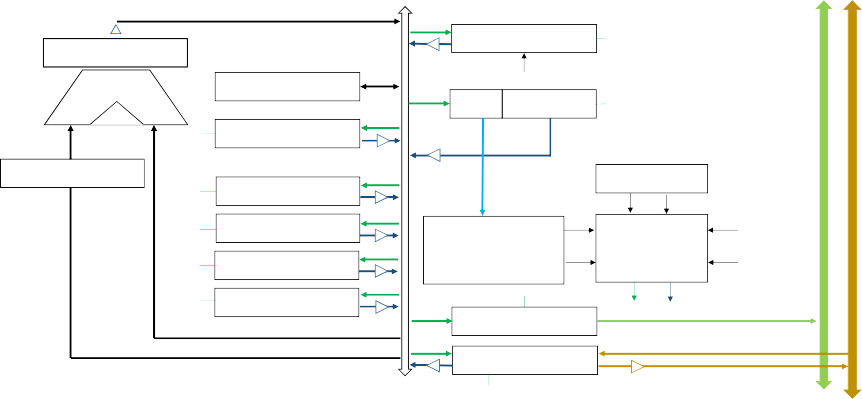
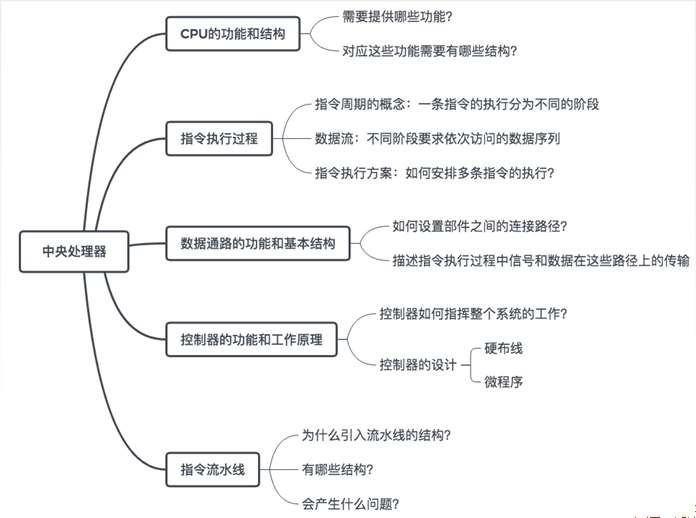
暂存寄存器

移位寄存器

... ALU

A B

ACCin



|  |
| --- |
| 本章总览  /CSKAOYAN.COM |

IR

MDR

王道考研/CSKAOYAN.COM

存储器

CU

MAR

PC

线 线 线

总 总 总

CPU

址 据 制

地 数 控

指令周期的数据流

|  |  |  |
| --- | --- | --- |
| 数据通路  数据通路：数据在功能部件之间传送的路径。 CPU内部总线  PC  移位寄存器 PCout +1  **.** ALU PSW  A B OP Ad  ACCin ACC IR  ACCout |  | 信息从哪里开始 |
| 中间经过哪些部件 数据总线 |
| 最后传到哪里 地址总线 |
| PCin |
| IRin |
| 由控制部件产生的控 |
| 制信号建立数据通路 |
| 暂存寄存器 AdIRout  R0in R□  R0out  R1in R□  R1out 指令译码器ID  R2in R□  R2out | **.** | 时序系统  …  微 操 作 信 号 **.** 标 志发生器 来自PSW  … |
| R3in R□ MARin  R3out MAR  MDR | | |
| MDRout MDRin |  | MDRoutE  王道考研/CSKAOYAN.COM |

|  |
| --- |
| 数据通路 信息从哪里开始  数据通路：数据在功能部件之间传送的路径。 CPU内部总线 中间经过哪些部件 数据总线  最后传到哪里 地址总线  数据通路的基本结构： PC PCin   1. CPU内部单总线方式。 PCout +1 2. CPU内部多总线方式。 PSW 3. 专用数据通路方式。 OP Ad IRin   ACC IR 由控制部件产生的控  ACCout 制信号建立数据通路  暂存寄存器 AdIRout 时序系统  R0in R□ …  R0out  R1in R□  R1out 指令译码器ID **.**. 微操作信号 **.**. 标志  R2in R□ 发生器 来自PSW  R2out …  R3in R□ MARin  R3out MAR  MDR  MDRout MDRin MDRoutE  王道考研/CSKAOYAN.COM |

DR

R

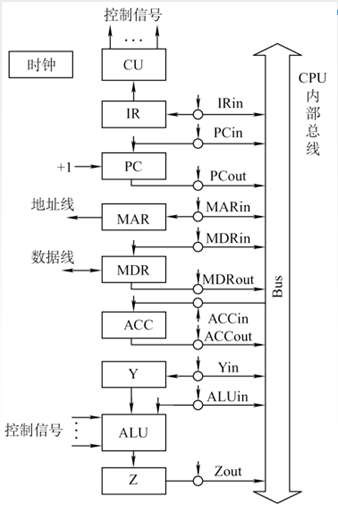
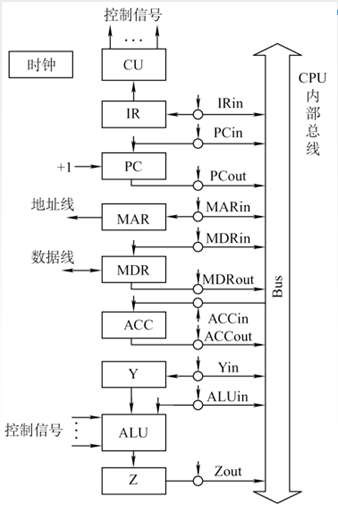
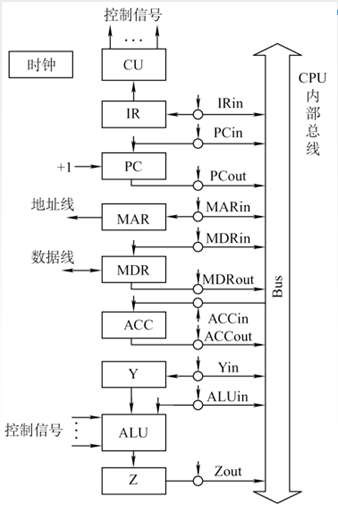
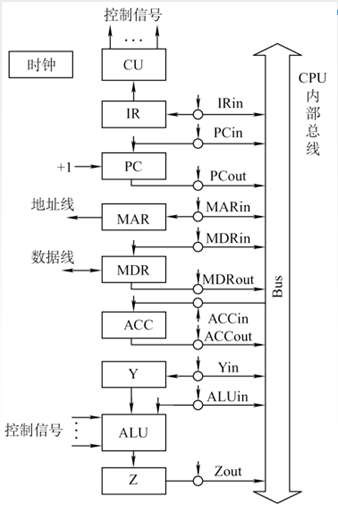
MMDR

DR

R

MMDR

ACC

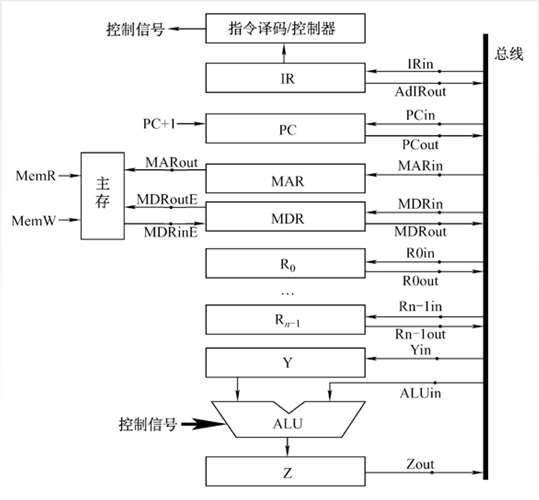
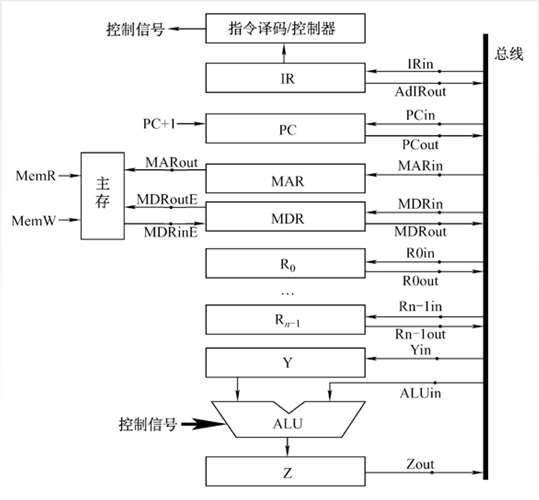
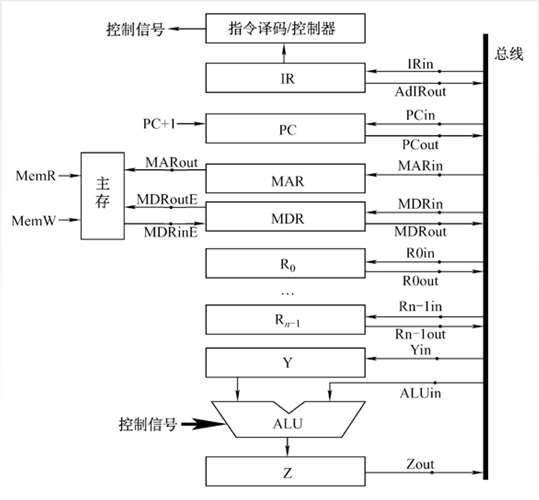
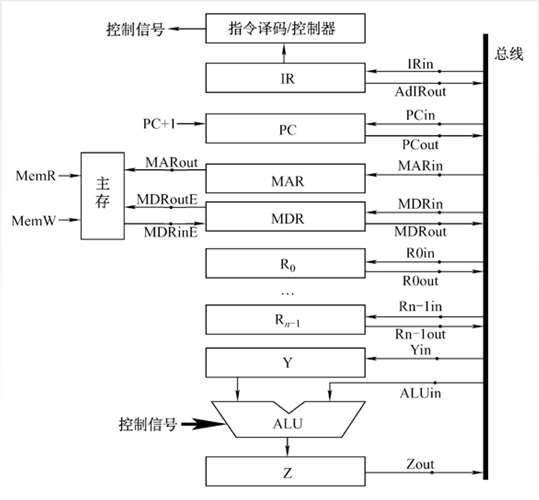


|  |
| --- |
| 数据通路-CPU内部单总线方式  内部总线是指同一部件，如CPU内部连接各寄存器及运算部 件之间的总线；  系统总线是指同一台计算机系统的各部件，如CPU、内存、 通道和各类I/O接口间互相连接的总线。  1．寄存器之间数据传送  PC 比如把PC内容送至MAR，实现传送操作的流程及控制信号为：  (PC)→Bus PCout有效，PC内容送总线  MAR Bus→MAR MARin有效，总线内容送MAR  关于是否加括号见王道p7最下方的讨论，写数据通路时，更关心数据流经的路径，可以省略括号，但从真题来看，考微操作序列居多，所以建议都加上，题目给出示例时跟题目保持一致。  王道考研/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、  IR 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  PC 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  MAR Bus→MAR MARin有效，总线内容送MAR  MDR 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/CSKAOYAN.COM |

|  |
| --- |
| 数据通路-CPU内部单总线方式  1．寄存器之间数据传送  AdIRout 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  **I**R 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  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 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  Y 1→R CU发读命令  MEM(MAR)→数据线→MDR MDRin有效MDR→Bus→Y MDRout和Yin有效，操作数→Y  王道考研/CSKAOYAN.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  **I**R 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  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 MDR→Bus→IR MDRout和IRin有效，现行指令→IR  3．执行算术或逻辑运算  ACC 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  Y 1→R CU发读命令  MEM(MAR)→数据线→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  Z  王道考研/CSKAOYAN.COM |



|  |  |  |
| --- | --- | --- |
| 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 | - |

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

|  |  |  |
| --- | --- | --- |
| 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 |

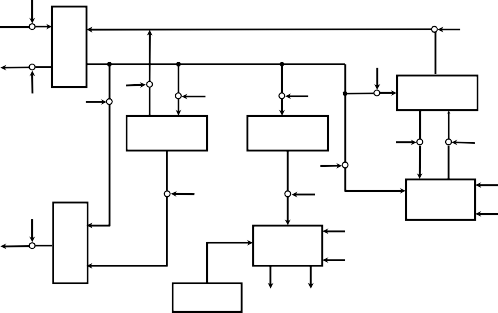
|  |  |  |
| --- | --- | --- |
| 时序 | 微操作 | 有效控制信号 |
| 1 | (R1)+(Y)→Z | R1out，ALUin，CU向  ALU发ADD控制信号 |
| 2 | (Z)→MDR | Zout，MDRin |
| 3 | (MDR)→M(MAR) | MemW，MDRoutE， MARout |

|  |
| --- |
| 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  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)→(R0)   取指周期、间址周期、执行周期   1. 写出各阶段的指令流程取指周期：公共操作   王道考研/CSKAOYAN.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  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)→(R0)   取指周期、间址周期、执行周期   1. 写出各阶段的指令流程取指周期：公共操作   王道考研/CSKAOYAN.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  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)→(R0)   取指周期、间址周期、执行周期   1. 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。  王道考研/CSKAOYAN.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  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)→(R0)   取指周期、间址周期、执行周期   1. 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。  王道考研/CSKAOYAN.COM |



王道考研/CSKAOYAN.COM

CU 标 志

…

控制信号

时钟

控制信号

M A R

ALU

C4

C0

1

C

C7

C8

C6

PC IR

AC

C3

C10

C13 C5

C9

12

C

C11

M D R

C2

专用数据通路方式

…

…

|  |
| --- |
| 本节回顾    王道考研/CSKAOYAN.COM |

王道考研/CSKAOYAN.COM

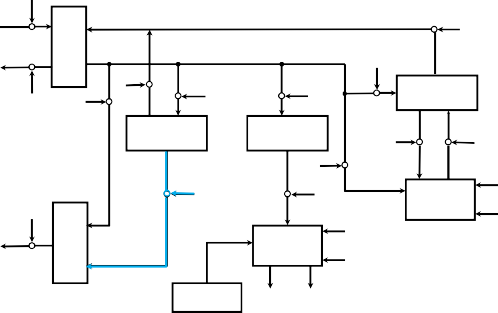
数据通路的功

能和基本结构专用通路

中央处理器

本节内容

|  |
| --- |
| 上节回顾    王道考研/CSKAOYAN.COM |



专用数据通路方式-取指周期

C2

(PC)→MAR C0有效

M D R

C11

C12

C9

C13

C5

C10

C3

AC

PC IR

C6

C8

C7

C

1

C0

C4

ALU

M A R

控制信号

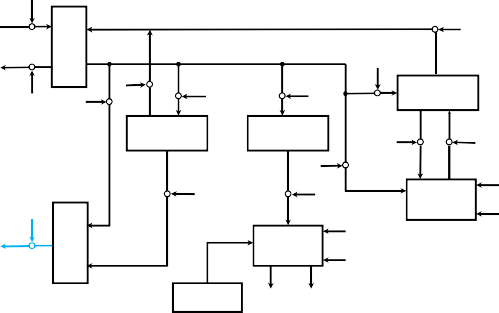
时钟

CU 标 志

…

控制信号

王道考研/CSKAOYAN.COM



专用数据通路方式-取指周期

C2

(PC)→MAR

M D R

C

11

(MAR)→主存

C0有效

C1有效

C12

C9

C13

C5

C10

C3

AC

PC IR

C6

C8

C7

C

1

C0

C4

ALU

M A R

控制信号

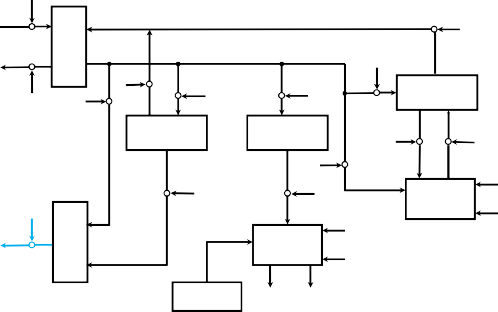
时钟

CU 标 志

…

控制信号

王道考研/CSKAOYAN.COM



王道考研/CSKAOYAN.COM

CU 标 志

…

控制信号

时钟

信号

M A R

1

控制

ALU

C4

C0

C

C7

C8

C6

PC IR

AC

C3

C10

5

C13 C

C9

1→R 控制单元向主存发送读命令

C12

C1有效

(MAR)→主存

11

C

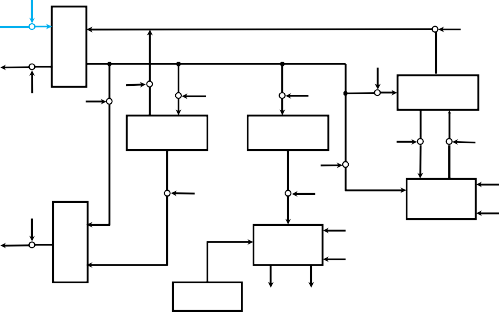
M D R

C0有效

(PC)→MAR

C2

专用数据通路方式-取指周期



王道考研/CSKAOYAN.COM

CU 标 志

…

控制信号

时钟

信号

M A R

1

控制

ALU

C4

C0

C

C7

C8

C6

PC IR

AC

C3

C10

5

C13 C

C9

1→R 控制单元向主存发送读命令

M(MAR)→MDR C2有效

C12

C1有效

(MAR)→主存

11

C

M D R

C0有效

(PC)→MAR

C2

专用数据通路方式-取指周期

…

…

…

…

…

…

…

…

的数据通路。

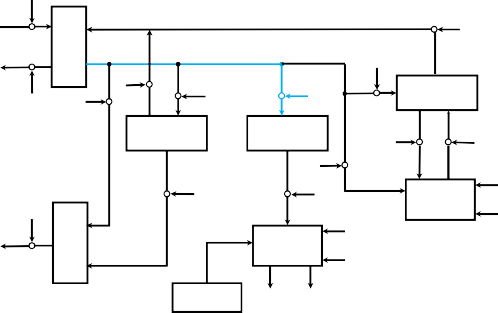
址 ， 的 功 能 为 → ） 。

址 ， 的 功 能为 → ）。址 ， 的 功 能为 → ） 。

的数据通路。

址 ， 的 功 能 为 → ） 。

址 ， 的 功 能为 → ）。址 ， 的 功 能为 → ） 。



专用数据通路方式-取指周期

C2

(PC)→MAR

C0有效

M D R

C

11

(MAR)→主存

C1有效

C12

C9

C13 C

5

C10

C3

AC

1→R 控制单元向主存发送读命令

M(MAR)→MDR C2 有 效(MDR)→IR C3有效(PC)+1→PC

PC IR

C6

C8

C7

C

1

C0

C4

ALU

M A R

控制信号

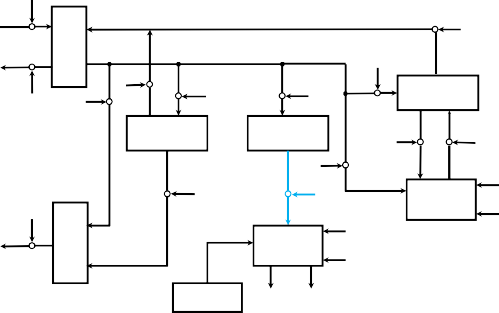
时钟

CU 标 志

…

控制信号

王道考研/CSKAOYAN.COM



专用数据通路方式-取指周期

C2

(PC)→MAR

C0有效

M D R

C

11

(MAR)→主存

C1有效

C12

1→R 控制单元向主存发送读命令

M(MAR)→MDR C2有效

C9

C

C3

AC

13

C

5

C10

PC

IR

(MDR)→IR (PC)+1→PC Op(IR)→CU

C

C3有效

C4有效

6

C8

C7

C

1

C0

C4

ALU

M A R

控制信号

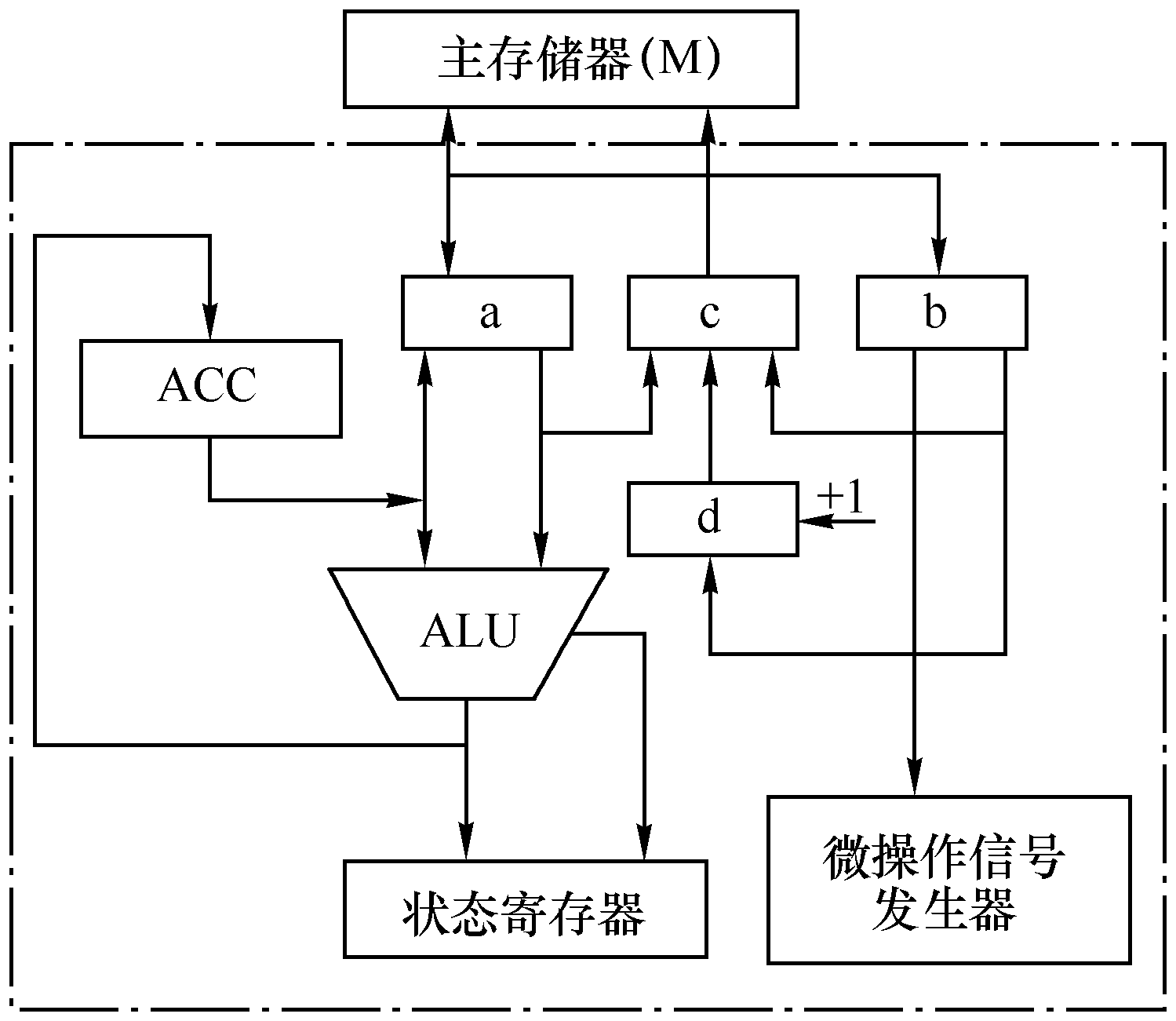
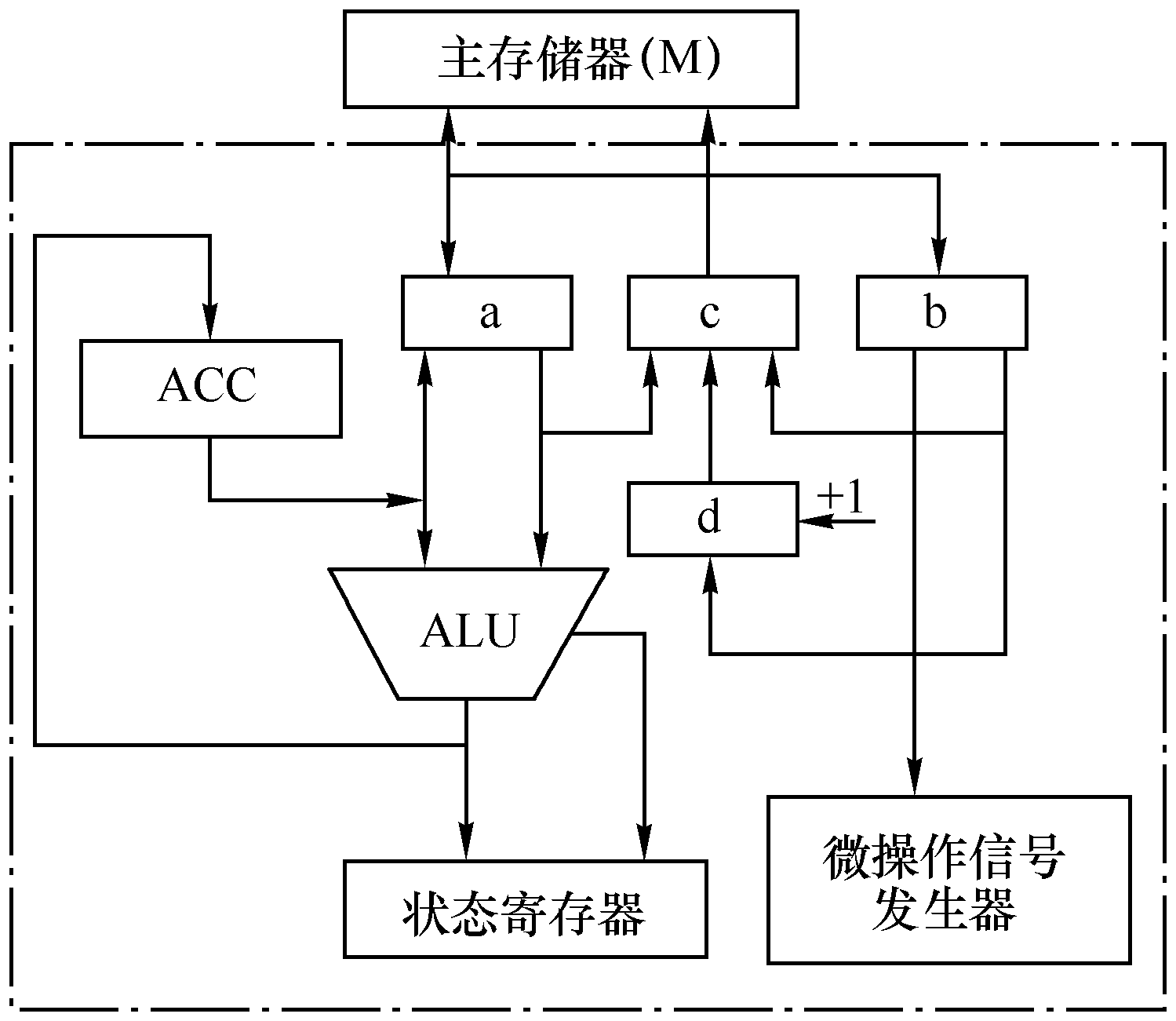
时钟

CU 标 志

…

控制信号

王道考研/CSKAOYAN.COM



…

…

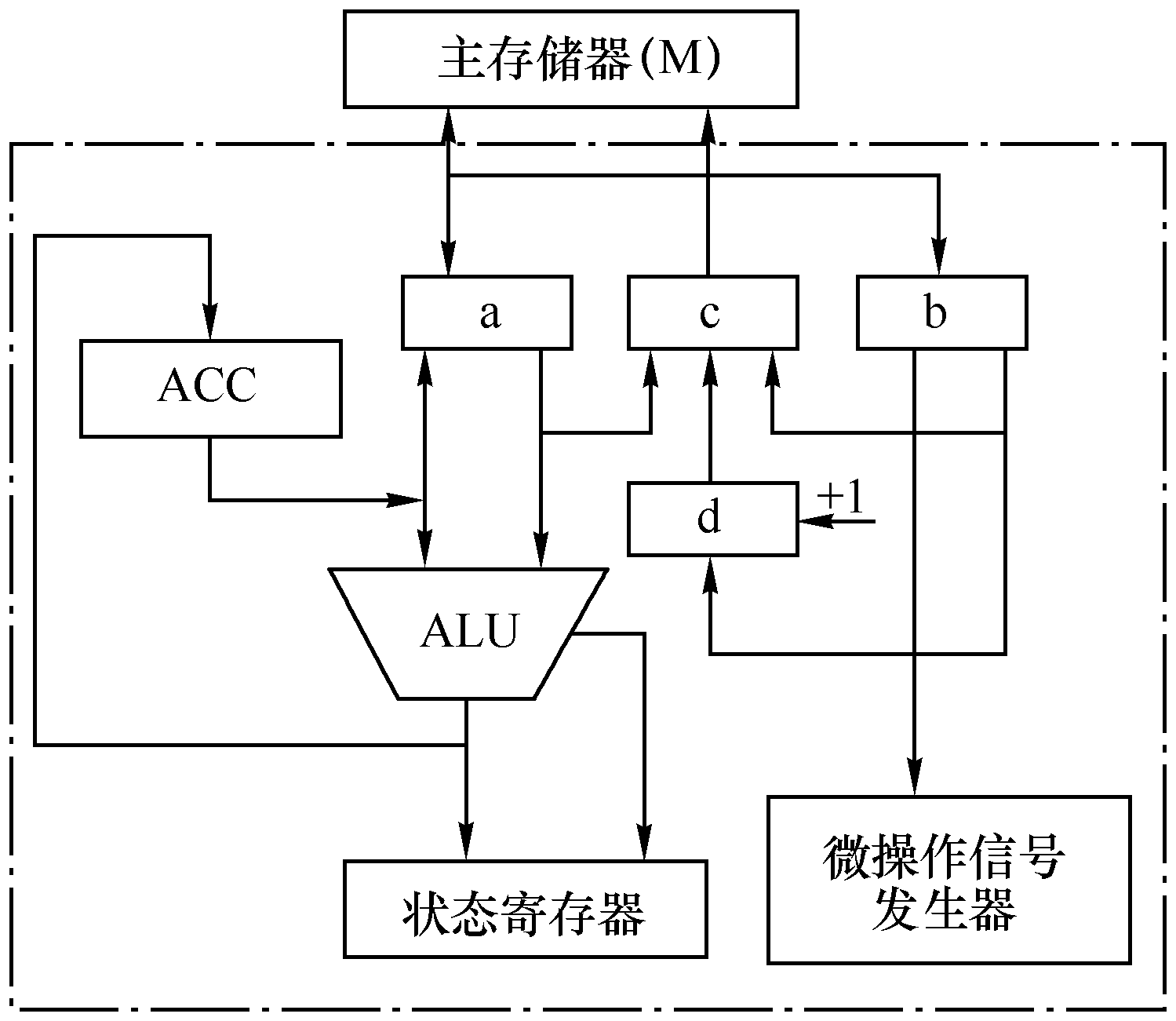
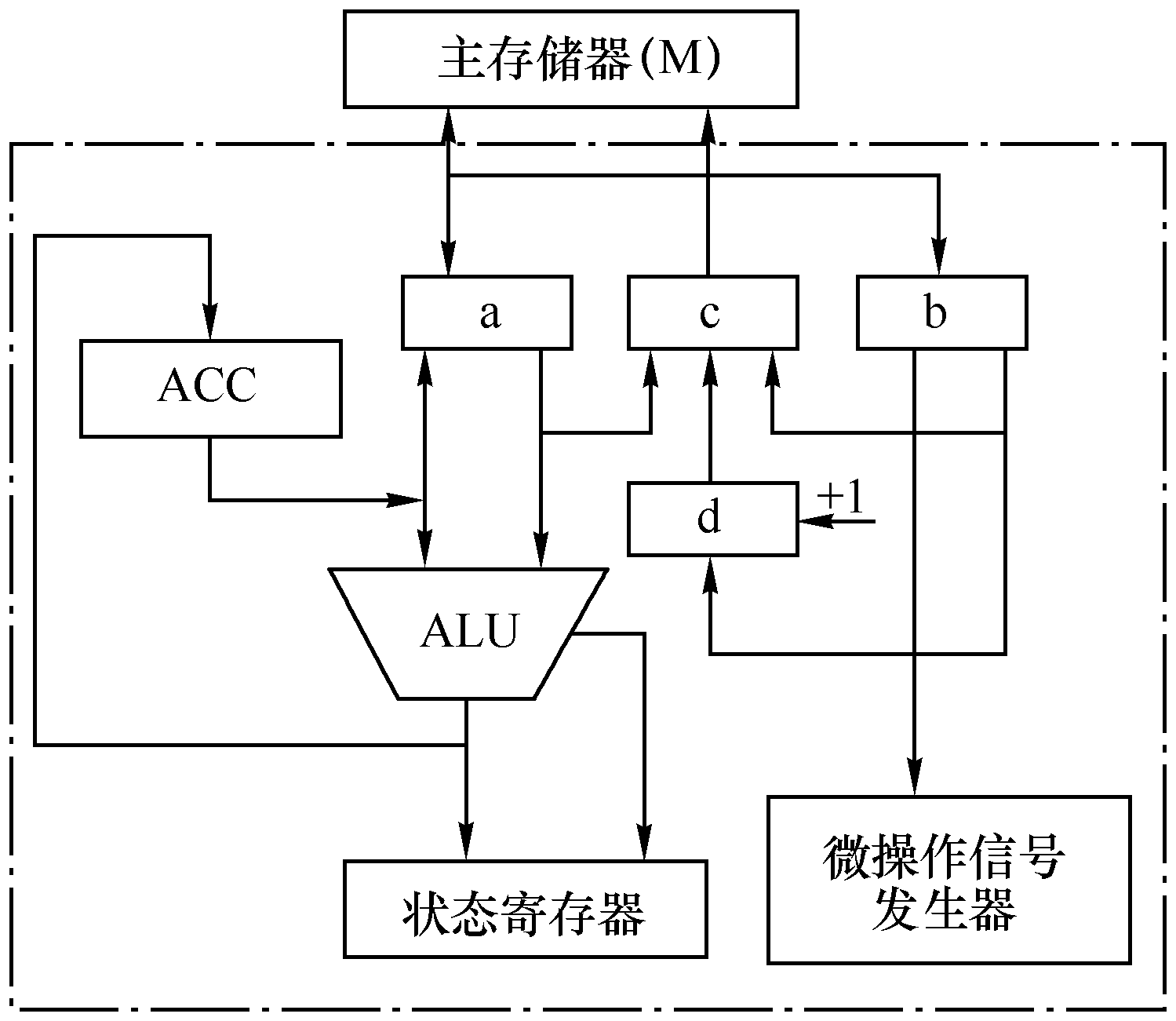
…

…

|  |
| --- |
| 专用数据通路方式-例题  下图是一个简化了的 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（ ）、一个状态数据寄存器和其他 个寄存器：主存地址寄存器（ ）、主存数据寄存器（ ）、程序寄存器（ ）和指令寄存器（ ），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：  （ ）请写出图中 、 、 、 个寄存器的名称。 |
| （ ）简述图中取指令的数据通路。  （ ）简述数据在运算器和主存之间进行存 取访问  （ ）简述完成指令 的数据通路（ 为主存地 MDR MAR IR  （ ）简述完成指令 的数据通路（ 为主存地  （ ）简述完成指令 的数据通路（ 为主存地  PC  (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/CSKAOYAN.COM |

|  |
| --- |
| 专用数据通路方式-例题  下图是一个简化了的 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（ ）、一个状态数据寄存器和其他 个寄存器：主存地址寄存器（ ）、主存数据寄存器（ ）、程序寄存器（ ）和指令寄存器（ ），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：  （ ）请写出图中 、 、 、 个寄存器的名称。 |
| （ ）简述图中取指令的数据通路。  （ ）简述数据在运算器和主存之间进行存 取访问  （ ）简述完成指令 的数据通路（ 为主存地 MDR MAR IR  （ ）简述完成指令 的数据通路（ 为主存地  （ ）简述完成指令 的数据通路（ 为主存地  PC  (2)  (PC) → MAR M(MAR) → MDR (MDR) → IR |
| 王道考研/CSKAOYAN.COM |

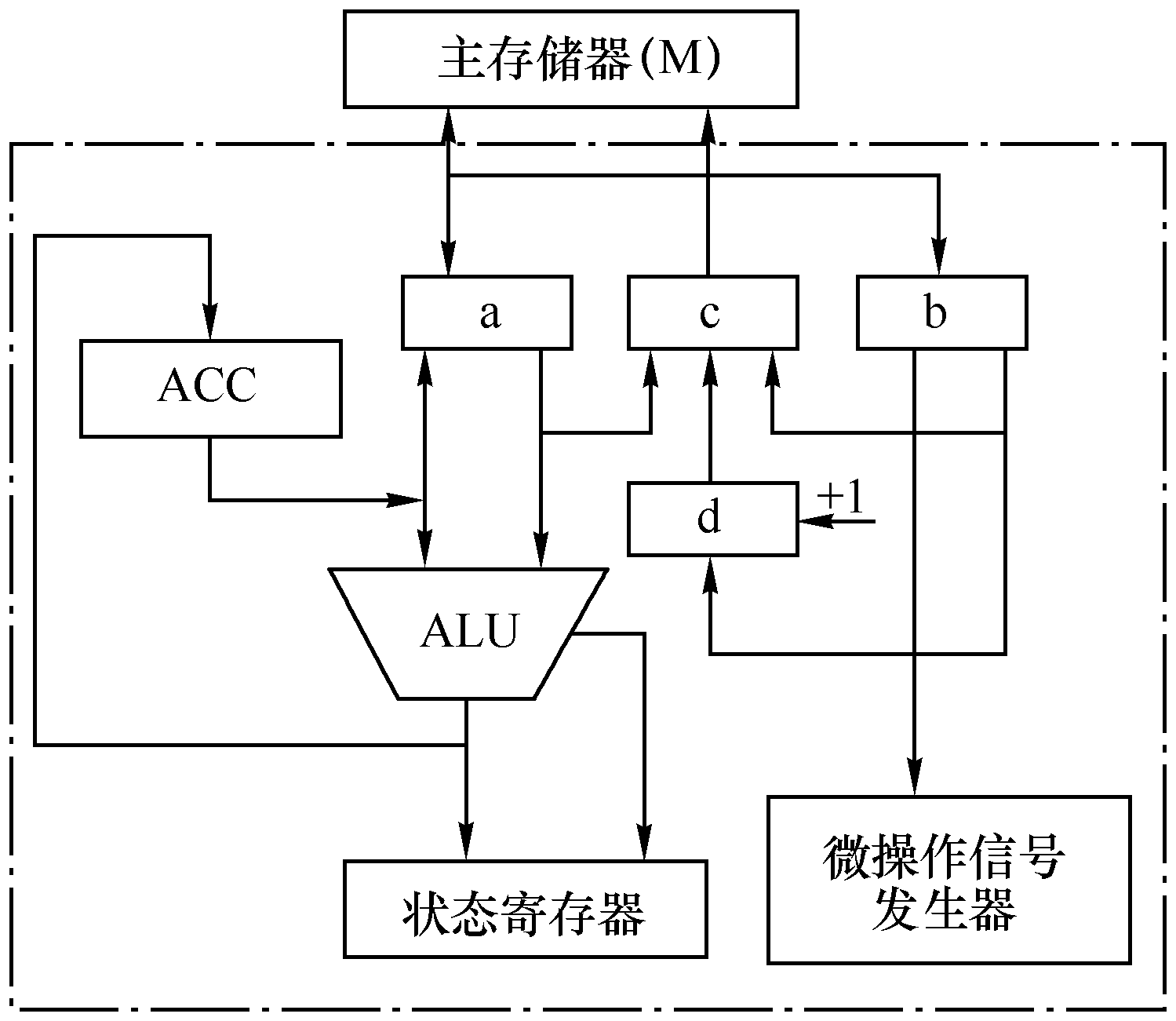
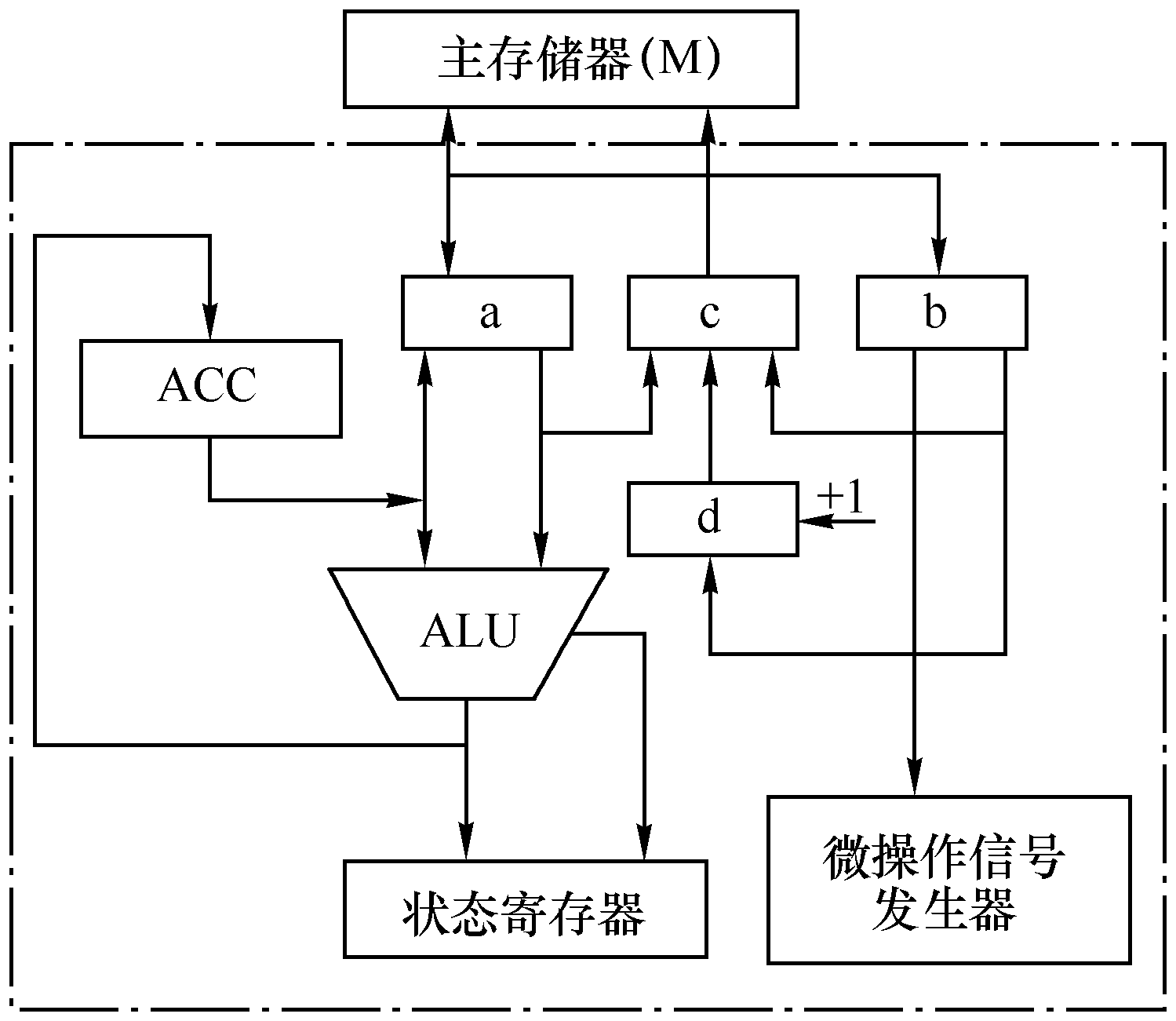
存/取的数据放到ACC中



|  |
| --- |
| 专用数据通路方式-例题  下图是一个简化了的 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（ ）、一个状态数据寄存器和其他 个寄存器：主存地址寄存器（ ）、主存数据寄存器（ ）、程序寄存器（ ）和指令寄存器（ ），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。  （ ）简述数据在运算器和主存之间进行存 取访问的数据通路。 |
| 设数据地址已放入MAR 取：  M(MAR) → MDR  (MDR) → ALU → ACC MDR MAR IR  存：  (ACC) → MDR PC  (MDR) → M(MAR) |
| 王道考研/CSKAOYAN.COM |

|  |
| --- |
| 专用数据通路方式-例题  下图是一个简化了的 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（ ）、一个状态数据寄存器和其他 个寄存器：主存地址寄存器（ ）、主存数据寄存器（ ）、程序寄存器（ ）和指令寄存器（ ），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。  （ ）简述完成指令 的数据通路（ 为主存地址， 的功能为 → ）。  X → MAR |
| M(MAR) → MDR (MDR) → ALU → ACC  MDR MAR IR  PC |
| 王道考研/CSKAOYAN.COM |

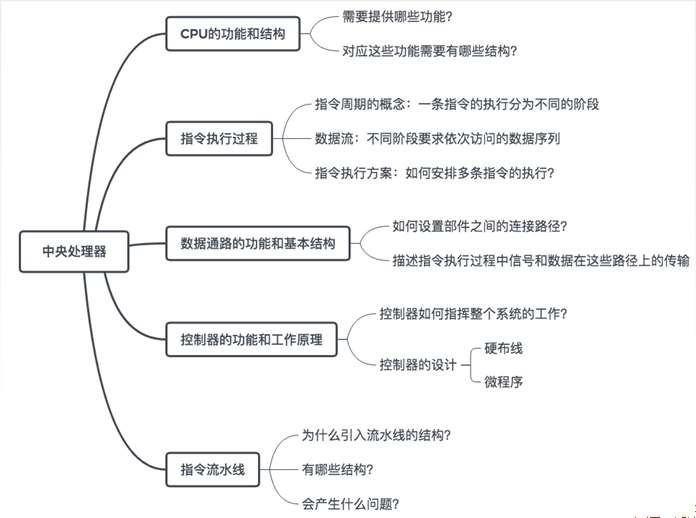
Y → MAR



|  |
| --- |
| 专用数据通路方式-例题  下图是一个简化了的 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（ ）、一个状态数据寄存器和其他 个寄存器：主存地址寄存器（ ）、主存数据寄存器（ ）、程序寄存器（ ）和指令寄存器（ ），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。  （ ）简述完成指令 的数据通路（ 为主存地址， 的功能为 → ） 。 |
| M(MAR) → MDR (MDR) → ALU, (ACC) → ALU ALU → ACC  MDR MAR IR  PC |
| 王道考研/CSKAOYAN.COM |

|  |
| --- |
| 专用数据通路方式-例题  下图是一个简化了的 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（ ）、一个状态数据寄存器和其他 个寄存器：主存地址寄存器（ ）、主存数据寄存器（ ）、程序寄存器（ ）和指令寄存器（ ），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。  （ ）简述完成指令 的数据通路（ 为主存地址， 的功能为 → ） 。  Z → MAR |
| (ACC) → MDR (MDR) → M(MAR)  MDR MAR IR  PC |
| 王道考研/CSKAOYAN.COM |

王道考研



|  |  |  |
| --- | --- | --- |
| 取指 | 分析 | 执行 |

# 

|  |
| --- |
| 本节回顾    涉及的主要操作类型： 基本思路：  寄存器之间的数据传送； 利用题目提供的数据通路进行数据传送； 主存与CPU之间的数据传送； 由CU发出的控制信号实现通路的建立。使用ALU进行算术逻辑运算。  王道考研/CSKAOYAN.COM |

王道考研/CSKAOYAN.COM

指令流水线

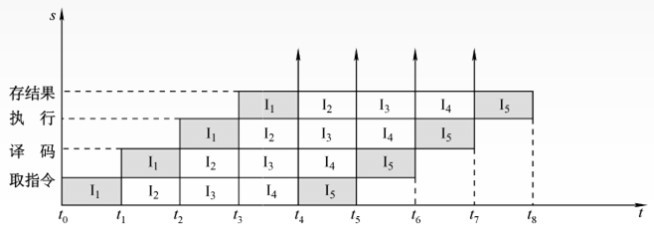
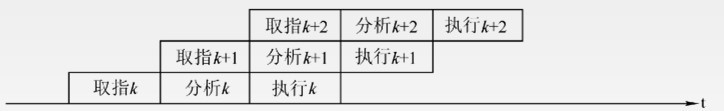
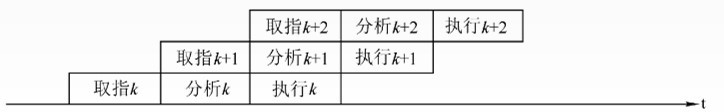
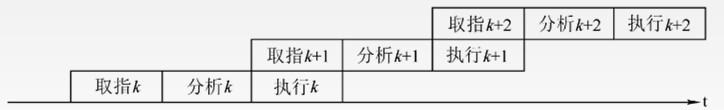
基本概念性能指标

中央处理器

本节内容

|  |
| --- |
| 本章总览  /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：根据PC内容访问主存储器，取出一条指令送到IR中。  特点：每个阶段用 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 到的硬件不一样。 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 个阶段的时间都相等，用 表示 ，按以下几种执行方式分析 条指令的执行时间：  1.顺序执行方式 总耗时T = n×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 |

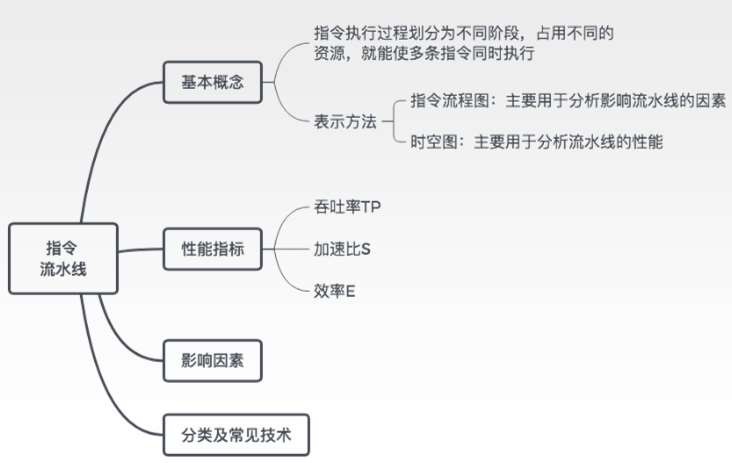
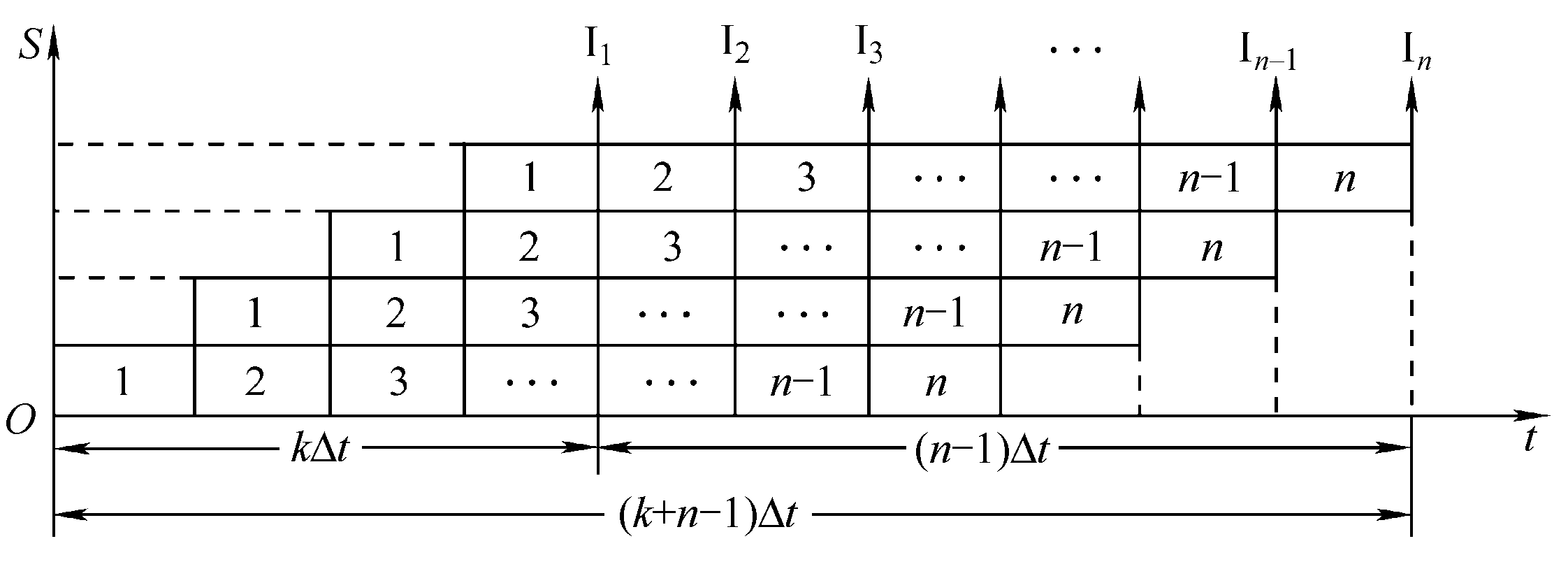
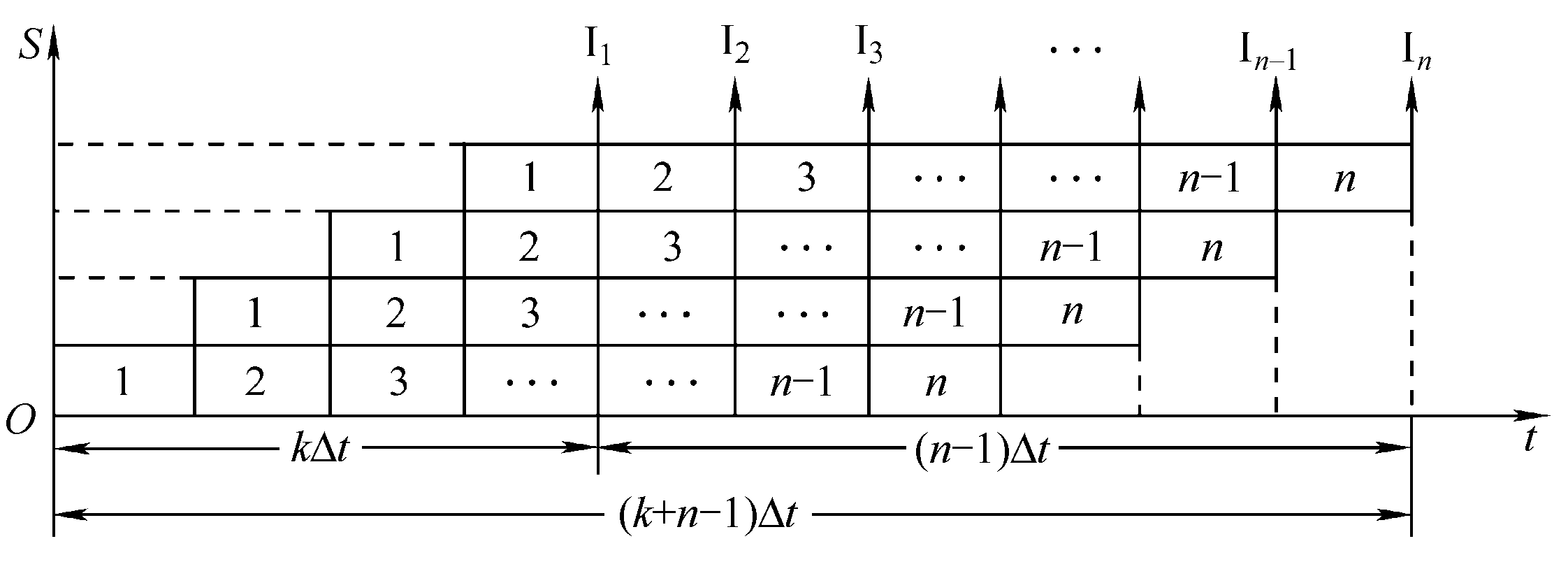


|  |
| --- |
| 指令流水的定义   1. 顺序执行方式 总耗时T = n×3t = 3nt     传统冯·诺依曼机采用顺序执行方式，又称串行执行方式。优点：控制简单，硬件代价小。  缺点：执行指令的速度较慢，在任何时刻，处理机中只有一条指令在执行，各功能部件的利用率很低。   1. 一次重叠执行方式 总耗时T = 3t + (n-1)×2t = (1+2n)t     优点：程序的执行时间缩短了1/3，各功能部件的利用率明显提高。  缺点：需要付出硬件上较大开销的代价，控制过程也比顺序执行复杂了。  王道考研/CSKAOYAN.COM |

|  |
| --- |
| 指令流水的定义   1. 顺序执行方式 总耗时T = n×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。   1. 一次重叠执行方式 总耗时T = 3t + (n-1)×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。   1. 二次重叠执行方式 总耗时T = 3t + (n-1)×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/CSKAOYAN.COM |

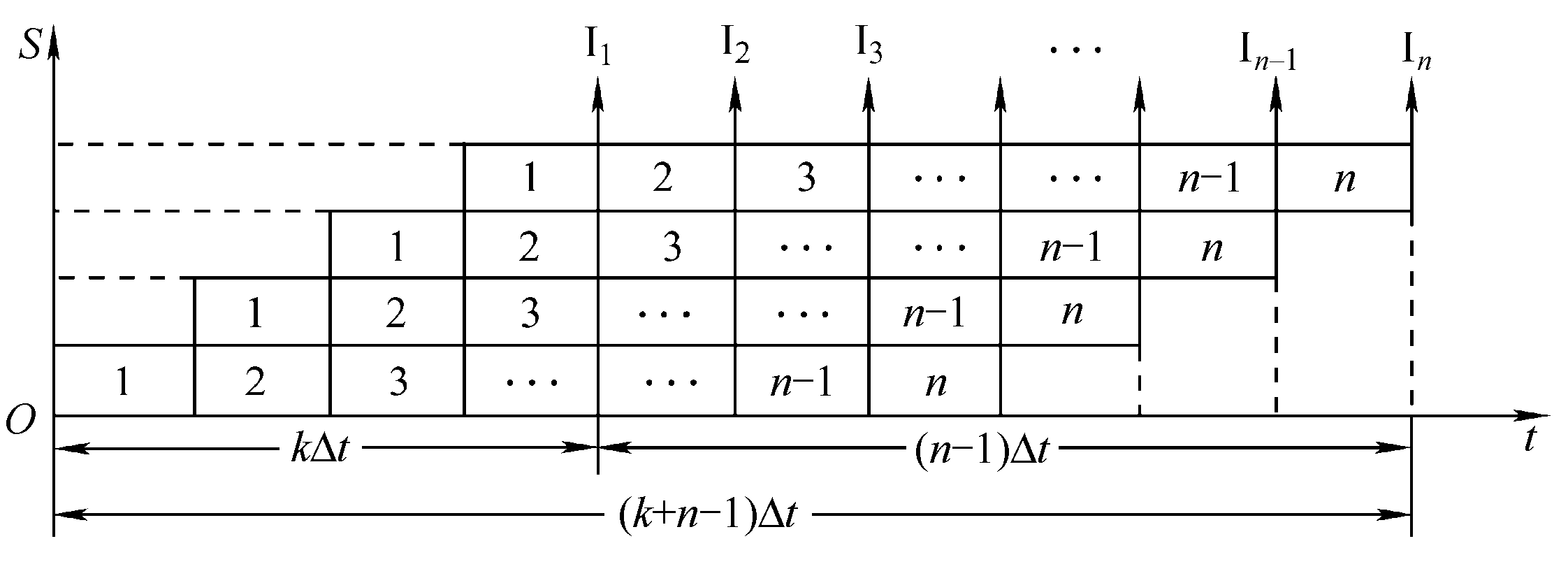
|  |
| --- |
| 流水线的表示方法   1. 指令执行过程图   指令序列  … …  第 条 指 令 I□□□  第 条 指 令 I□□□ 第 条指令I□  主要用于分析指令执行过程以及影响流水线的因素(见下一个视频)   1. 时空图   空间：不同的阶段所对应的不同的  硬件资源 主要用于分析流水线的性能  时间  王道考研/CSKAOYAN.COM |

|  |
| --- |
| 流水线的性能指标   1. 吞吐率 2. 加速比 3. 效率 |
|  |
|  |
| 王道考研/CSKAOYAN.COM |



|  |
| --- |
| 流水线的性能指标  1.吞吐率 吞吐率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Tk  则计算流水线吞吐率（TP）的最基本的公式为TP  n  Tk  理想情况下，流水线的时空图如下： 当连续输入的任务n→∞时，得最大吞吐率为TPmax=1/t。 装入时间 排空时间  = (k+n-1) t  流水线的实际吞吐率为  TP  n  (k  n 1)t  一条指令的执行分为k个阶段，每个阶段耗时t，一般取t =一个时钟周期  王道考研/CSKAOYAN.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T0表示不使用流水线时的执行时间，即顺序执行所用的时间；Tk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  T0 当连续输入的任务n→∞时，最大加速比为S =k。  T 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 t，则顺序完成n个任务耗时𝑇□= nk t  = (k+n-1) t  实际加速比为  S  knt  kn  (k  n 1)t k  n 1  一条指令的执行分为k个阶段，每个阶段耗时t，一般取t =一个时钟周期  王道考研/CSKAOYAN.COM |



理想情况下，流水线的时空图如下：

当连续输入的任务n→∞时， 最高效率为Emax=1。

一条指令的执行分为k个阶段，每个阶段耗时t，一般取t =一个时钟周期

王道考研/CSKAOYAN.COM

T0

k

n个任务占用k时空区有效面积

则流水线效率（E）的一般公式为E  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  kT

流水线的性能指标

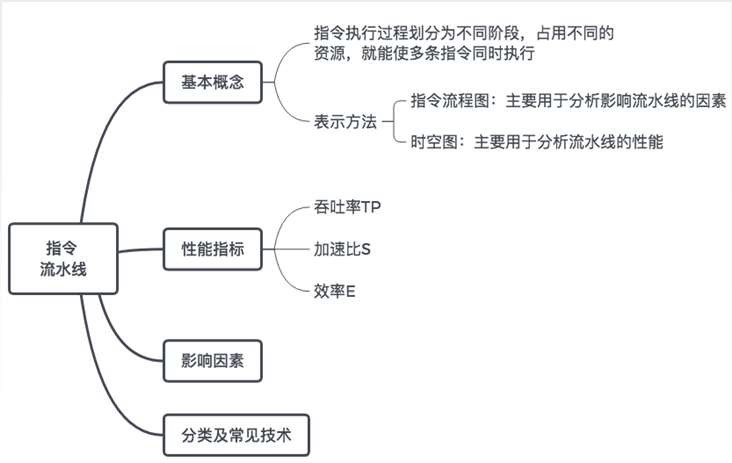
流水线的设备利用率称为流水线的效率。

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。

3.效率

|  |
| --- |
| 本节回顾  理想情况：  各阶段花费时间相同；  每个阶段结束后能立即进入下一阶段。  王道考研/CSKAOYAN.COM |



王道考研/CSKAOYAN.COM

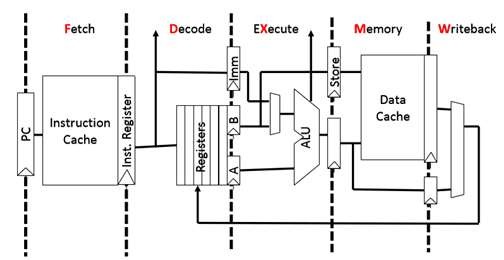
指令流水线

影响因素分类

中央处理器

本节内容

|  |
| --- |
| 本节总览  理想情况：  各阶段花费时间相同；  每个阶段结束后能立即进入下一阶段。  王道考研/CSKAOYAN.COM |



机器周期的设置

IF

ID

EX

M

WB

流水线每一个功能段部 件后面都要有一个缓冲寄存器，或称为锁存器，其作用是保存本流水段的 执行结果，提供给下一流水段使用。

王道考研/CSKAOYAN.COM

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。

50ns 50ns

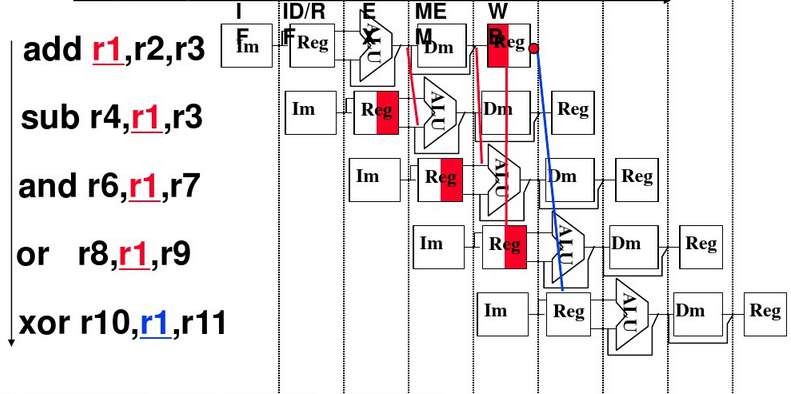
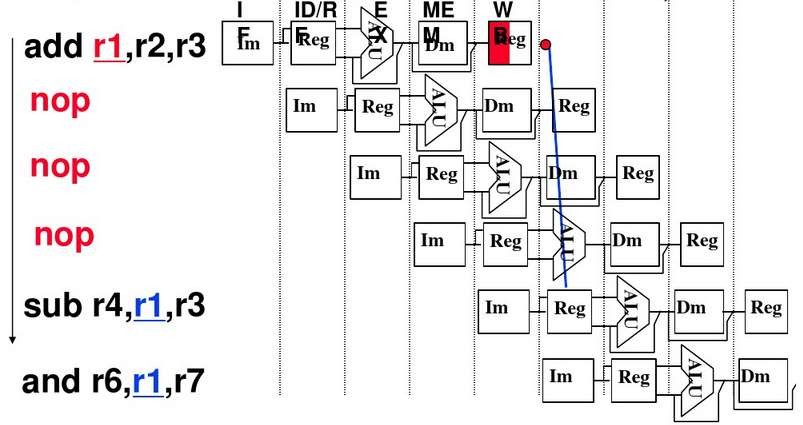
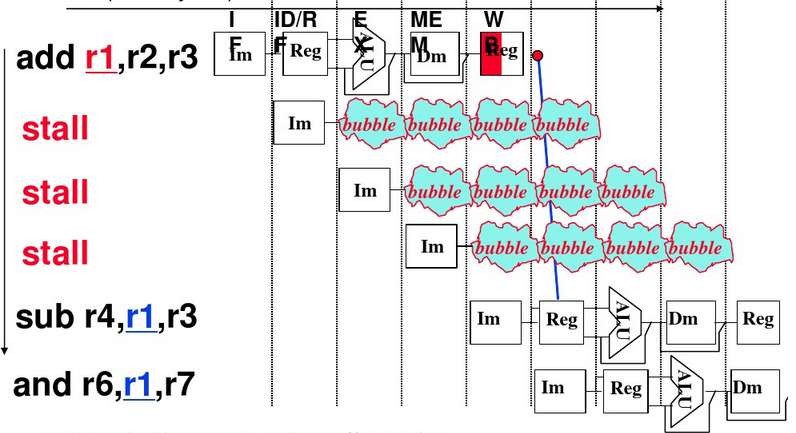
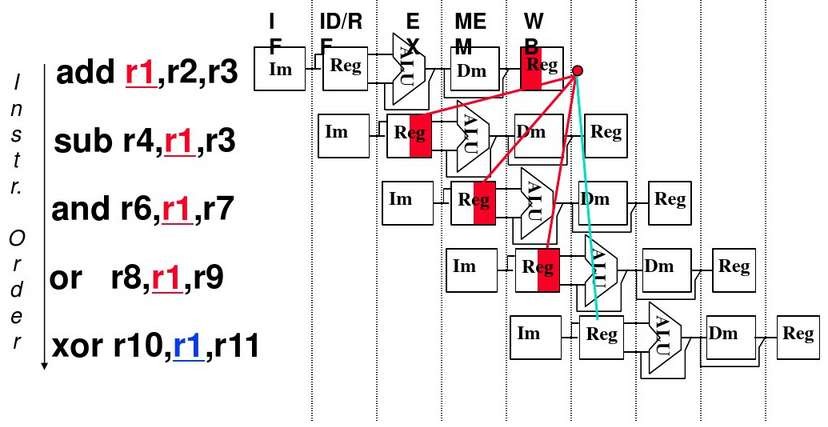
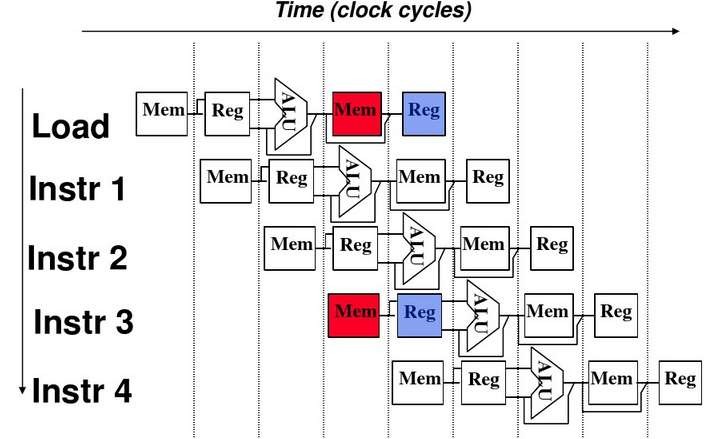
80ns 70ns

100ns

各部件实际耗时：

|  |
| --- |
| 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/CSKAOYAN.COM |



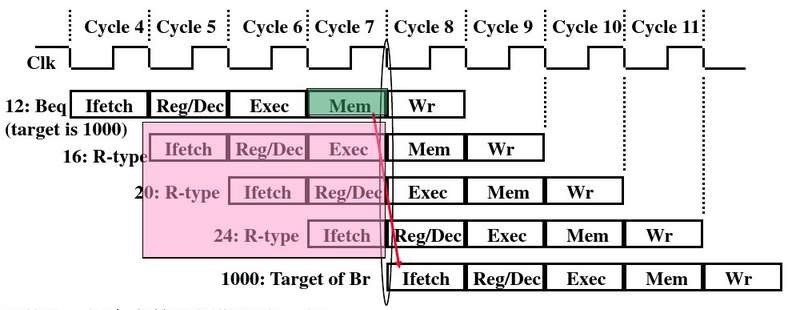


|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 影响流水线的因素  1.结构相关（资源冲突）  由于多条指令在同一时刻争用同一资源而形成的冲突称为结构相关。  解决办法： | | | | | | | | |
| **Im** | **Im** | **Im** | **Dm**  **Im** | **Dm**  **Im** | **Dm** | **Dm** | **Dm** | 后一相关指令暂停一周期  资源重复配置：  数据存储器 指令存储器 |
| 王道考研/CSKAOYAN.COM | | | | | | | | |

|  |
| --- |
| 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：  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。数据旁路技术。  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/CSKAOYAN.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  ， 、 和 均为 型变量，它们的存储单元地址分别表示为 、 和 。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。 | | | |
| ， | ( [a] ) —> R1 | 与和 |  |
| ， | ( [b] ) —> R2 |  |
| ，  ，  则这 条指令执行过程中 | ( R1 ) + ( R2 ) —> R2  ( R2 ) —> [x]  的 段和 的 段被阻塞的原因各是什么？ | 和 存在数据相关， 存在数据相关。 |
|  | | | |
| 王道考研/CSKAOYAN.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/CSKAOYAN.COM |





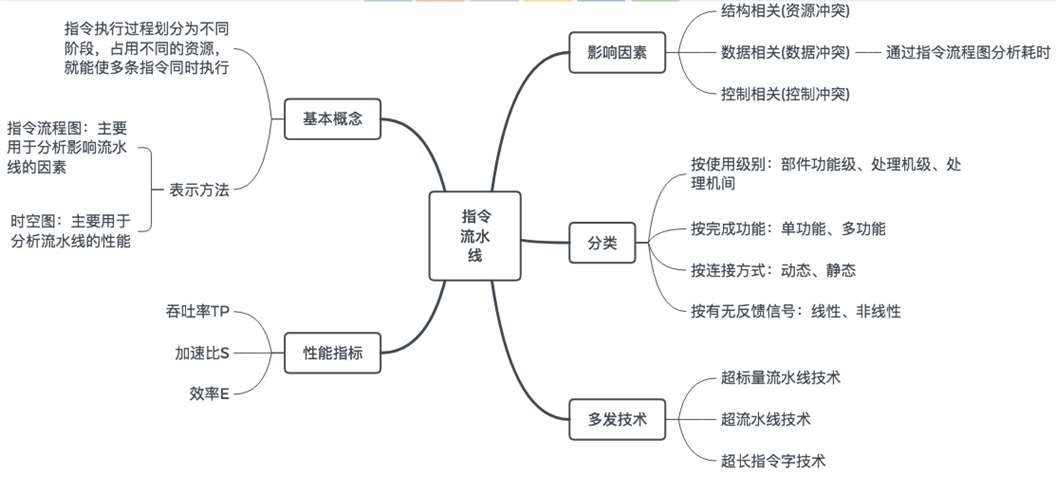
|  |
| --- |
| 影响流水线的因素  3.控制相关（控制冲突）  当流水线遇到转移指令和其他改变PC值的指令而造成断流时，会引起控制相关。  解决办法： |
| 尽早判别转移是否发生，尽早生成  转移目标地址  预取转移成功和不成功两个控制流  方向上的目标指令 |
| 加快和提前形成条件码  提高转移方向的猜准率 |
| 王道考研/CSKAOYAN.COM |

|  |
| --- |
| 影响流水线的因素    王道考研/CSKAOYAN.COM |

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。

|  |
| --- |
| 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/CSKAOYAN.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 |
| 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/CSKAOYAN.COM |



# 

|  |
| --- |
| 流水线的多发技术  1.超标量技术  IF ID EX WR  t  0 1 2 3 4 5 6 7 8 9 10 11 12 13  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/CSKAOYAN.COM |

|  |
| --- |
| 流水线的多发技术  2.超流水技术  IF ID EX WR  流水线速度是原来速度的 3倍  t  0 1 2 3 4 5 6 7 8 9 10 11 12 13  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/CSKAOYAN.COM |

|  |
| --- |
| 流水线的多发技术  3.超长指令字  IF ID EX WR  t  0 1 2 3 4 5 6 7 8 9 10 11 12 13  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/CSKAOYAN.COM |

|  |
| --- |
| 本节回顾  王道考研/CSKAOYAN.COM |