

王道考研——组成原理

WWW.CSKAOYAN.COM

第四章 指令系统

本节内容

指令系统

指令格式

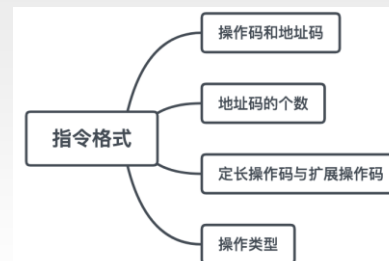
王道考研/CSKAOYAN.COM

本章总览



王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

指令的定义

指令（又称机器指令）：

是指示计算机执行某种操作的命令，是计算机运行的最小功能单位。

一台计算机的所有指令的集合构成该机的指令系统，也称为指令集。

注：一台计算机只能执行自己指令系统中的指令，不能执行其他系统的指令。



软件

1000100011111001001111110001010

指令



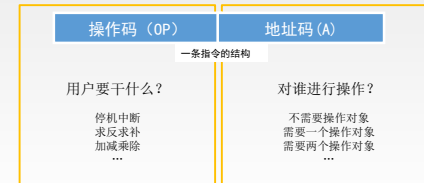
硬件

王道考研/CSKAOYAN.COM

指令格式

一条指令就是机器语言的一个语句，它是一组有意义的二进制代码。

一条指令通常要包括操作码字段和地址码字段两部分：



OP	A ₁	A ₂	A ₃ (结果)	A ₄ (下址)
----	----------------	----------------	---------------------	---------------------

指令含义：(A₁)OP(A₂)→A₃，A₄=下一条将要执行指令的地址

王道考研/CSKAOYAN.COM

指令格式-地址码

假设是+

四地址指令	OP	A ₁	A ₂	A ₃ (结果)	A ₄ (下址)
-------	----	----------------	----------------	---------------------	---------------------

指令含义：(A₁)OP(A₂)→A₃，A₄=下一条将要执行指令的地址

设指令字长为32位，操作码占8位，4个地址码字段各占6位
设存储字长为32位，即4B

- A₁可直接表示2⁶=64个不同的位置

- 一条指令的执行(假设每个地址都是主存地址)：

取指令 访存1次(假设指令字长=存储字长)

取两个操作数 访存2次

存回结果 访存1次

共访存4次

000000	000420C4H
000001	12344321H
000010	43211234H
000011	55555555H
000100	22343234H
000101	
...	
111101	
111110	
111111	

王道考研/CSKAOYAN.COM

指令格式-地址码

假设是+

四地址指令	OP	A ₁	A ₂	A ₃ (结果)	A ₄ (下址)
-------	----	----------------	----------------	---------------------	---------------------

指令含义：(A₁)OP(A₂)→A₃，A₄=下一条将要执行指令的地址

设指令字长为32位，操作码占8位，4个地址码字段各占6位
设存储字长为32位，即4B

- A₁可直接表示2⁶=64个不同的位置

- 一条指令的执行(假设每个地址都是主存地址)：

取指令 访存1次(假设指令字长=存储字长)

取两个操作数 访存2次

存回结果 访存1次

共访存4次

程序计数器PC 000000 取出00FFEF44H后 000001

三地址指令	OP	A ₁	A ₂	A ₃ (结果)
-------	----	----------------	----------------	---------------------

000000	00FFEF44H
000001	22BCE142H
000010	
000011	
000100	
000101	
...	
111101	55555555H
111110	43211234H
111111	12344321H

王道考研/CSKAOYAN.COM

指令格式-地址码

设指令字长及存储字长均为32位，操作码占8位

四地址指令	OP	A ₁	A ₂	A ₃ (结果)	A ₄ (下址)
-------	----	----------------	----------------	---------------------	---------------------

指令含义: (A₁)OP(A₂)→A₃, A₄=下一条将要执行指令的地址
4个地址码字段各占6位, 指令操作数直接寻址范围为2⁶=64; 完成一条指令需要访存4次

三地址指令	OP	A ₁	A ₂	A ₃ (结果)
-------	----	----------------	----------------	---------------------

指令含义: (A₁)OP(A₂)→A₃
3个地址码字段各占8位, 指令操作数直接寻址范围为2⁸=256; 完成一条指令需要访存4次

二地址指令	OP	A ₁ (目的操作数)	A ₂ (源操作数)
-------	----	------------------------	-----------------------

指令含义: (A₁)OP(A₂)→A₁
2个地址码字段各占12位, 指令操作数直接寻址范围为2¹²=4K; 完成一条指令需要访存4次

一地址指令	OP	A ₁
-------	----	----------------

指令含义: 1. OP(A₁)→A₁, 如加1、减1、取反、求补等 完成一条指令需要访存3次
2. (ACC)OP(A₁)→ACC, 隐含约定的目的地址为ACC 完成一条指令需要访存2次
1个地址码字段占24位, 指令操作数直接寻址范围为2²⁴=16M

王道考研/CSKAOYAN.COM

指令格式-地址码

设指令字长及存储字长均为32位，操作码占8位

四地址指令	OP	A ₁	A ₂	A ₃ (结果)	A ₄ (下址)
-------	----	----------------	----------------	---------------------	---------------------

指令含义: (A₁)OP(A₂)→A₃, A₄=下一条将要执行指令的地址
4个地址码字段各占6位, 指令操作数直接寻址范围为2⁶=64; 完成一条指令需要访存4次

三地址指令	OP	A ₁	A ₂	A ₃ (结果)
-------	----	----------------	----------------	---------------------

指令含义: (A₁)OP(A₂)→A₃
3个地址码字段各占8位, 指令操作数直接寻址范围为2⁸=256; 完成一条指令需要访存4次

二地址指令	OP	A ₁ (目的操作数)	A ₂ (源操作数)
-------	----	------------------------	-----------------------

指令含义: (A₁)OP(A₂)→A₁
2个地址码字段各占12位, 指令操作数直接寻址范围为2¹²=4K; 完成一条指令需要访存4次

一地址指令	OP	A ₁
-------	----	----------------

指令含义: 1. OP(A₁)→A₁, 如加1、减1、取反、求补等 完成一条指令需要访存3次
2. (ACC)OP(A₁)→ACC, 隐含约定的目的地址为ACC 完成一条指令需要访存2次
1个地址码字段占24位, 指令操作数直接寻址范围为2²⁴=16M

零地址指令	OP
-------	----

指令含义: 1. 不需要操作数, 如空操作、停机、关中断等指令
2. 堆栈计算机, 两个操作数隐含存放在栈顶和次栈顶, 计算结果压回栈顶

王道考研/CSKAOYAN.COM

定长指令字结构: 指令长度固定
变长指令字结构: 指令长度不等

指令格式-操作码

设指令字长及存储字长均为32位，操作码占8位

四地址指令	OP	A ₁	A ₂	A ₃ (结果)	A ₄ (下址)
-------	----	----------------	----------------	---------------------	---------------------

指令含义: (A₁)OP(A₂)→A₃, A₄=下一条将要执行指令的地址
4个地址码字段各占6位, 指令操作数直接寻址范围为2⁶=64; 完成一条指令需要访存4次

三地址指令	OP	A ₁	A ₂	A ₃ (结果)
-------	----	----------------	----------------	---------------------

指令含义: (A₁)OP(A₂)→A₃
3个地址码字段各占8位, 指令操作数直接寻址范围为2⁸=256; 完成一条指令需要访存4次

二地址指令	OP	A ₁ (目的操作数)	A ₂ (源操作数)
-------	----	------------------------	-----------------------

指令含义: (A₁)OP(A₂)→A₁
2个地址码字段各占12位, 指令操作数直接寻址范围为2¹²=4K; 完成一条指令需要访存4次

一地址指令	OP	A ₁
-------	----	----------------

指令含义: 1. OP(A₁)→A₁, 如加1、减1、取反、求补等 完成一条指令需要访存3次
2. (ACC)OP(A₁)→ACC, 隐含约定的目的地址为ACC 完成一条指令需要访存2次
1个地址码字段占24位, 指令操作数直接寻址范围为2²⁴=16M

零地址指令	OP
-------	----

指令含义: 1. 不需要操作数, 如空操作、停机、关中断等指令
2. 堆栈计算机, 两个操作数隐含存放在栈顶和次栈顶, 计算结果压回栈顶

王道考研/CSKAOYAN.COM

定长操作码: n位 → 2ⁿ条指令
扩展操作码: 操作码长度可变

扩展操作码

扩展操作码举例

在设计扩展操作码指令格式时, 必须注意以下两点:

- 1) 不允许短码是长码的前缀, 即短操作码不能与长操作码的前面部分的代码相同。
- 2) 各指令的操作码一定不能重复。

通常情况下, 对使用频率较高的指令, 分配较短的操作码; 对使用频率较低的指令, 分配较长的操作码, 从而尽可能减少指令译码和分析的时间。

还有其他扩展操作码设计方法。

OP	A ₁	A ₂	A ₃
----	----------------	----------------	----------------

4位操作码	0000 0001 1110 1111	A ₁ A ₁ A ₁ A ₁	A ₂ A ₂ A ₂ A ₂	A ₃ A ₃ A ₃ A ₃	15条三地址指令
-------	------------------------------	--	--	--	----------

8位操作码	1111 0000 1111 0001 1111 1110 1111 1111	A ₁ A ₁ A ₁ A ₁	A ₂ A ₂ A ₂ A ₂	A ₃ A ₃ A ₃ A ₃	15条二地址指令
-------	--	--	--	--	----------

12位操作码	1111 1111 0000 1111 1111 0001 1111 1111 1110 1111 1111 1111	A ₁ A ₁ A ₁ A ₁	A ₂ A ₂ A ₂ A ₂	A ₃ A ₃ A ₃ A ₃	15条一地址指令
--------	--	--	--	--	----------

16位操作码	1111 1111 1111 0000 1111 1111 1111 0001 1111 1111 1111 1110 1111 1111 1111 1111	A ₁ A ₁ A ₁ A ₁	A ₂ A ₂ A ₂ A ₂	A ₃ A ₃ A ₃ A ₃	16条零地址指令
--------	--	--	--	--	----------

王道考研/CSKAOYAN.COM

扩展操作码

扩展操作码举例

设指令字长固定为16位，试设计一套指令系统满足：

- 有15条三地址指令
共 $2^3=16$ 种状态
留出16-15=1种
- 有12条二地址指令
共 $1 \times 2^4=16$ 种
留出16-12=4种
- 有62条一地址指令
共 $4 \times 2^4=64$ 种
留出64-62=2种
- 有32条零地址指令
共 $2 \times 2^4=32$ 种

设地址长度为n，上一层留出m种状态，下一层可扩展出 $m \times 2^n$ 种状态

王道考研/CSKAOYAN.COM

指令操作码

操作码指出指令中该指令应该执行什么性质的操作和具有何种功能。

操作码是识别指令、了解指令功能与区分操作数地址内容的组成和使用方法等的关键信息。例如，指出是算术加运算，还是减运算；是程序转移，还是返回操作。

操作码分类：

定长操作码：在指令字的最高位部分分配固定的若干位（定长）表示操作码。

- 一般n位操作码字段的指令系统最大能够表示 2^n 条指令。
- 优：定长操作码对于简化计算机硬件设计，提高指令译码和识别速度很有利；
- 缺：指令数量增加时会占用更多固定位，留给表示操作数地址的位数受限。

扩展操作码(不定长操作码)：全部指令的操作码字段的位数不固定，且分散地放在指令字的不同位置上。

- 最常见的变长操作码方法是扩展操作码，使操作码的长度随地址码的减少而增加，不同地址数的指令可以具有不同长度的操作码，从而在满足需要的前提下，有效地缩短指令字长。
- 优：在指令字长有限的前提下仍保持比较丰富的指令种类；
- 缺：增加了指令译码和分析的难度，使控制器的设计复杂化。

王道考研/CSKAOYAN.COM

操作类型

1. 数据传送

LOAD 作用：把**存储器**中的数据放到**寄存器**中 数据传送类：进行主存与CPU之间的数据传送
STORE 作用：把**寄存器**中的数据放到**存储器**中

2. 算术逻辑操作

算术：加、减、乘、除、增1、减1、求补、浮点运算、十进制运算
逻辑：与、或、非、异或、位操作、位测试、位清除、位求反

运算类

3. 移位操作

算术移位、逻辑移位、循环移位(带进位和不带进位)

4. 转移操作

无条件转移 JMP
条件转移 JZ：结果为0；JO：结果溢出；JC：结果有进位
调用和返回 CALL和RETURN
陷阱(Trap)与陷阱指令

程序控制类：改变程序执行的顺序

5. 输入输出操作

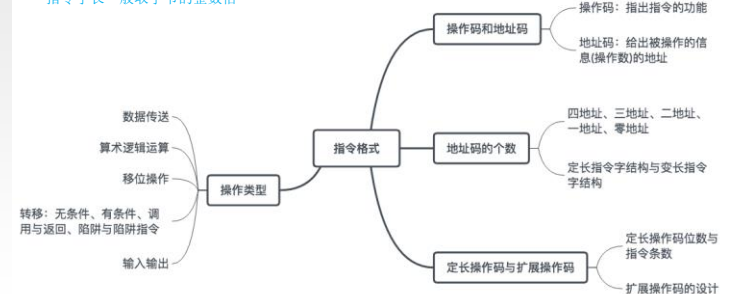
CPU寄存器与IO端口之间的数据传送(端口即IO接口中的寄存器)

输入输出类(I/O)：进行CPU和I/O设备之间的数据传送

王道考研/CSKAOYAN.COM

本节回顾

指令字长一般取字节的整数倍



王道考研/CSKAOYAN.COM

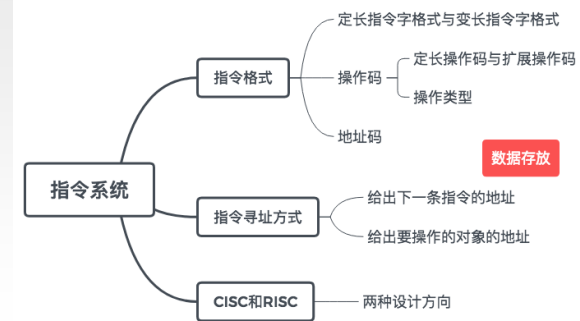
本节内容

指令系统

数据存放
指令寻址

王道考研/CSKAOYAN.COM

本章总览



王道考研/CSKAOYAN.COM

数据存放

按字节寻址：给出一个字节地址，可以取出长度为一个字节的数据

字地址：
0~3为一个字，4~7为一个字...
每个字中最小的字节地址作为字地址

假设某数据长度为4B，则需要给出1个字地址

字存储单元	字节存储单元
0	0 1 2 3
4	12 13 14 15
8	16 17 18 19
12	20 21 22 23
16	24 25 26 27
20	
24	

按字寻址：每个字存储单元对应一个地址编号
存放4B的数据 12345678H

按字节寻址：每个字节存储单元都有一个地址编号

按字节地址寻址：给出一个字节地址，可以取出长度为一个字节的数据

假设某数据长度为4B，则需要给出4个字节地址

三个字长：

机器字长：CPU一次能处理的二进制数据的位数。一般等于内部寄存器的位数。

指令字长：一个指令字中包含二进制代码的位数。单字长指令：指令长度=机器字长

存储字长：一个存储单元存储二进制代码的长度。半字长指令、双字长指令

注：这些长度都是字节的整数倍 通常一次并行取出一个存储字，按字节寻址时通过移位截取存储字中的一个字节

王道考研/CSKAOYAN.COM

边界对齐存储

按字节寻址：给出一个字节地址，可以取出长度为一个字节的数据

字地址：
0~3为一个字，4~7为一个字...
每个字中最小的字节地址作为字地址

假设某数据长度为4B，则需要给出1个字地址

字存储单元	字节存储单元
0	0 1 2 3
4	12 13 14 15
8	16 17 18 19
12	20 21 22 23
16	24 25 26 27
20	
24	

按字寻址：每个字存储单元都有一个地址编号
边界对齐：起始地址是字长 (按字节算) 的整数倍
存放4B的数据 12345678H

按字节寻址：每个字节存储单元都有一个地址编号

按字节地址寻址：给出一个字节地址，可以取出长度为一个字节的数据

假设某数据长度为4B，则需要给出4个字节地址

三个字长：

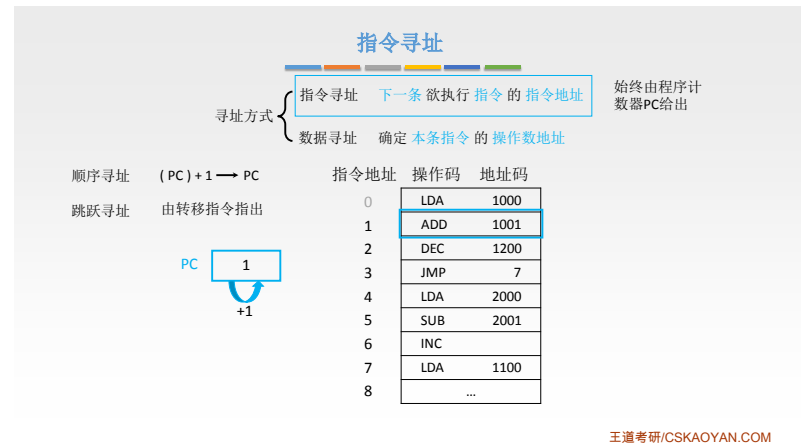
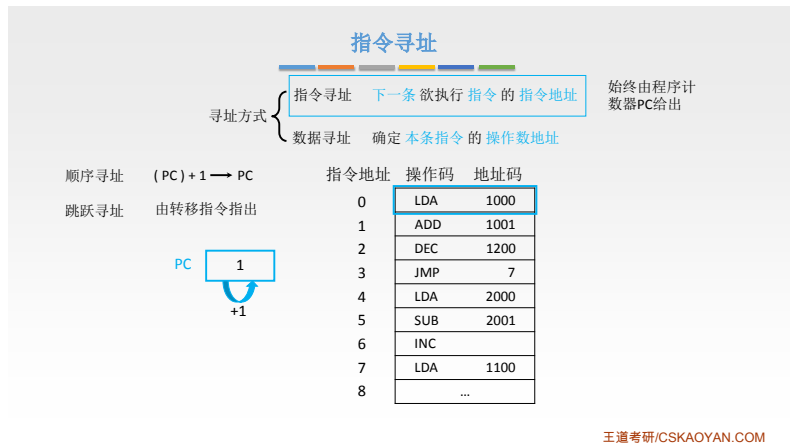
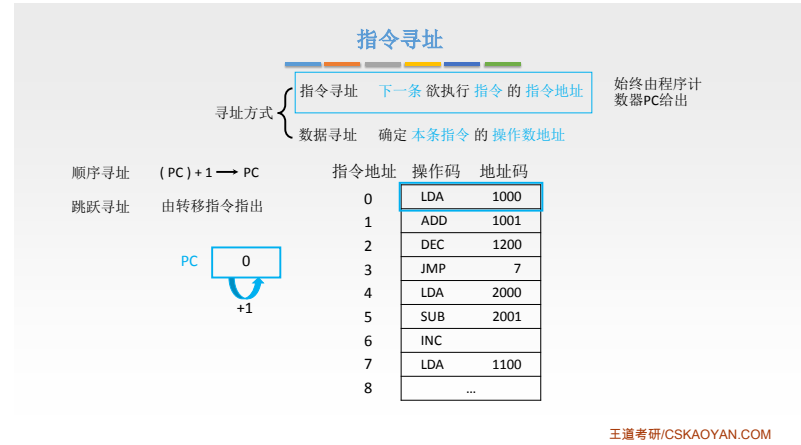
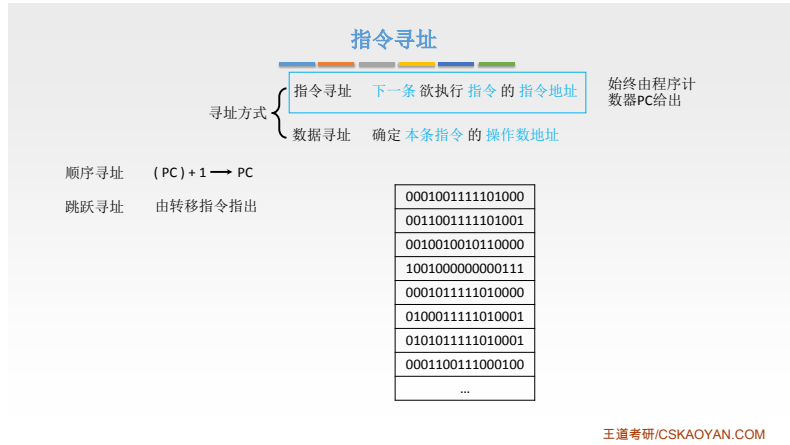
机器字长：CPU一次能处理的二进制数据的位数。一般等于内部寄存器的位数。

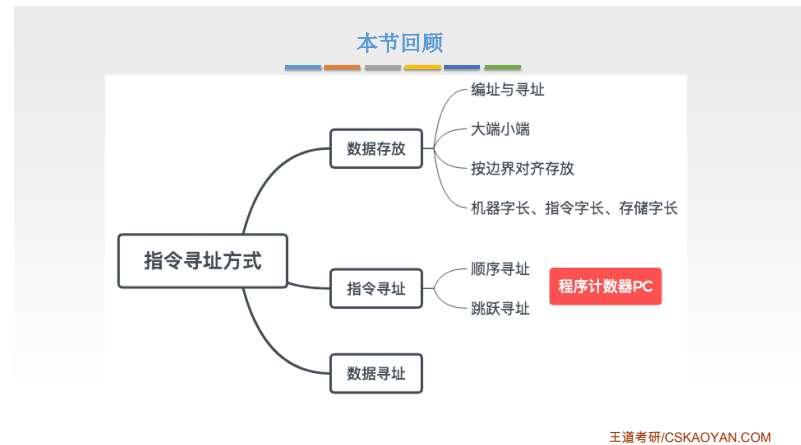
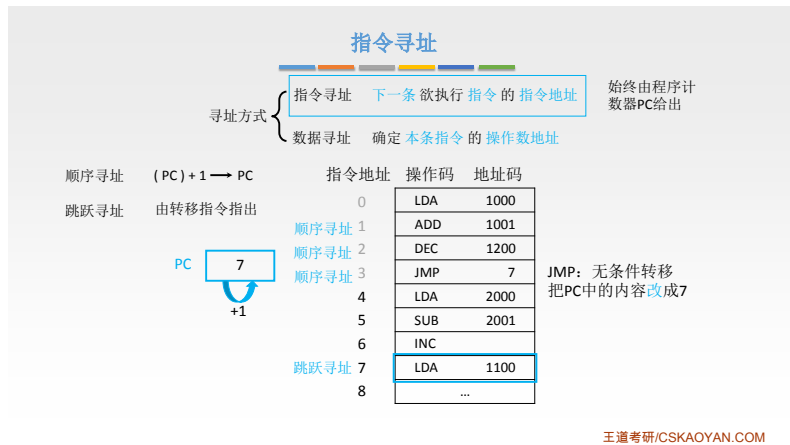
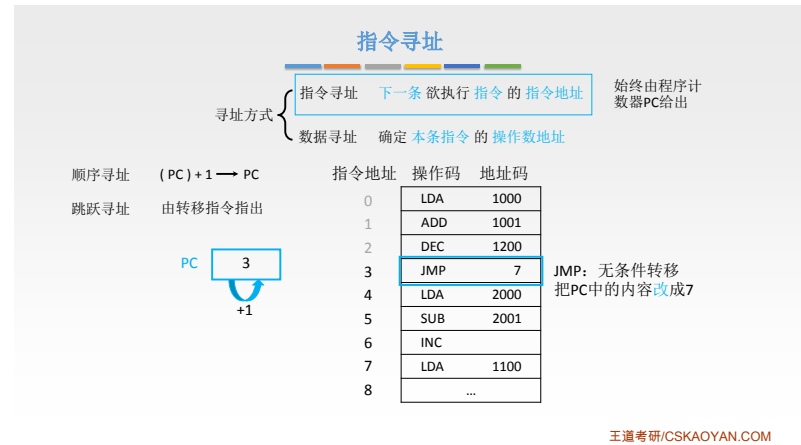
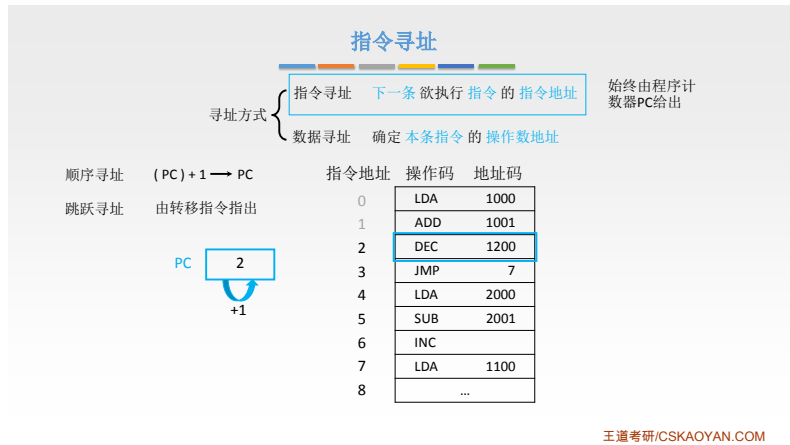
指令字长：一个指令字中包含二进制代码的位数。单字长指令：指令长度=机器字长

存储字长：一个存储单元存储二进制代码的长度。半字长指令、双字长指令

注：这些长度都是字节的整数倍 通常一次并行取出一个存储字，按字节寻址时通过移位截取存储字中的一个字节

王道考研/CSKAOYAN.COM





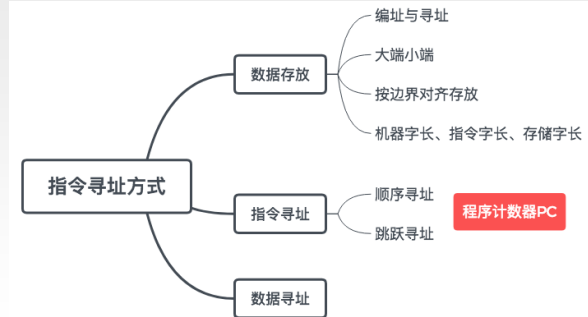
本节内容

指令系统

数据寻址-1

王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

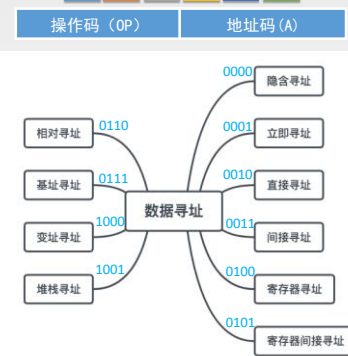
操作数类型



0	LDA	1000
1	ADD	1001
2	DEC	1200
3	JMP	7
4	LDA	2000
5	SUB	2001
6	INC	
7	LDA	1100
8	...	

王道考研/CSKAOYAN.COM

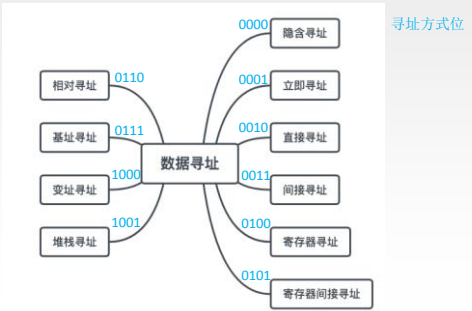
数据寻址



寻址方式位

王道考研/CSKAOYAN.COM

数据寻址



王道考研/CSKAOYAN.COM

数据寻址



假设指令字长=机器字长=存储字长
假设操作数为3

王道考研/CSKAOYAN.COM

数据寻址



王道考研/CSKAOYAN.COM

立即寻址



立即寻址：形式地址A就是操作数本身，又称为立即数，一般采用补码形式。
#表示立即寻址特征。

一条指令的执行：
取指令 访存1次
执行指令 访存0次
暂不考虑访存结果
共访存1次

优点：指令执行阶段不访问主存，指令执行时间最短

缺点：
A的位数限制了立即数的范围。
如A的位数为n，且立即数采用补码时，可表示的数据范围为 $-2^{n-1} \sim 2^{n-1} - 1$

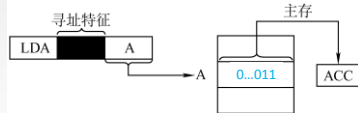
王道考研/CSKAOYAN.COM

直接寻址

假设指令字长=机器字长=存储字长, 操作数为3



直接寻址: 指令字中的形式地址A就是操作数的真实地址EA, 即EA=A。



一条指令的执行:
取指令 访问1次
执行指令 访问1次
 暂不考虑存储结果
 共访问2次

优点: 简单, 指令执行阶段仅访问一次主存, 不需专门计算操作数的地址。

缺点:
 A的位数决定了该指令操作数的寻址范围。
 操作数的地址不易修改。

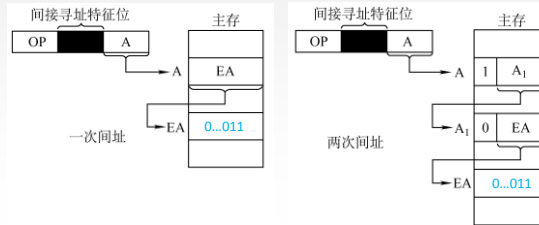
王道考研/CSKAOYAN.COM

间接寻址

假设指令字长=机器字长=存储字长, 操作数为3



间接寻址: 指令的地址字段给出的形式地址不是操作数的真正地址, 而是操作数有效地址所在的存储单元的地址, 也就是操作数地址的地址, 即EA=(A)。



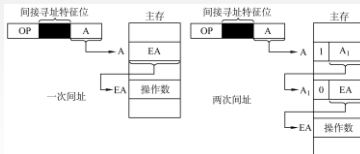
王道考研/CSKAOYAN.COM

间接寻址

假设指令字长=机器字长=存储字长, 操作数为3



间接寻址: 指令的地址字段给出的形式地址不是操作数的真正地址, 而是操作数有效地址所在的存储单元的地址, 也就是操作数地址的地址, 即EA=(A)。



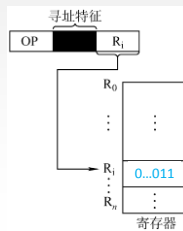
优点:
 可扩大寻址范围(有效地址EA的位数大于形式地址A的位数)。
 便于编制程序(用间接寻址可以方便地完成子程序返回)。

缺点:
 指令在执行阶段要多次访问(一次间接寻址需两次访问, 多次寻址需根据存储的最高位确定几次访问)。

王道考研/CSKAOYAN.COM

寄存器寻址

假设指令字长=机器字长=存储字长, 操作数为3

寄存器寻址: 在指令字中直接给出操作数所在的寄存器编号, 即EA=R_i, 其操作数在由R_i所指的寄存器内。

一条指令的执行:
取指令 访问1次
执行指令 访问0次
 暂不考虑存储结果
 共访问1次

优点:
 指令在执行阶段不访问主存, 只访问寄存器, 指令字短且执行速度快, 支持向量/矩阵运算。

缺点:
 寄存器价格昂贵, 计算机中寄存器个数有限。

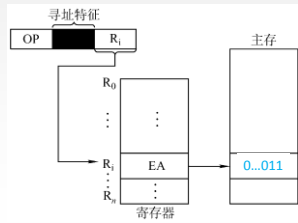
王道考研/CSKAOYAN.COM

寄存器间接寻址

假设指令字长=机器字长=存储字长，操作数为3



寄存器间接寻址：寄存器 R_i 中给出的不是一个操作数，而是操作数所在主存单元的地址，即 $EA=(R_i)$ 。



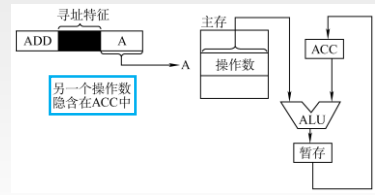
一条指令的执行：
取指令 访存1次
执行指令 访存1次
暂不考虑存结果
共访存2次

特点：
与一般间接寻址相比速度更快，但指令的执行阶段需要访问主存(因为操作数在主存中)。

王道考研/CSKAOYAN.COM

隐含寻址

隐含寻址：不是明显地给出操作数的地址，而是在指令中隐含着操作数的地址。



优点：有利于缩短指令字长。
缺点：需增加存储操作数或隐含地址的硬件。

王道考研/CSKAOYAN.COM

本节回顾

寻址方式	有效地址	访存次数(指令执行期间)
隐含寻址	程序指定	0
立即寻址	A即是操作数	0
直接寻址	$EA=A$	1
一次间接寻址	$EA=(A)$	2
寄存器寻址	$EA=R_i$	0
寄存器间接一次寻址	$EA=(R_i)$	1

王道考研/CSKAOYAN.COM

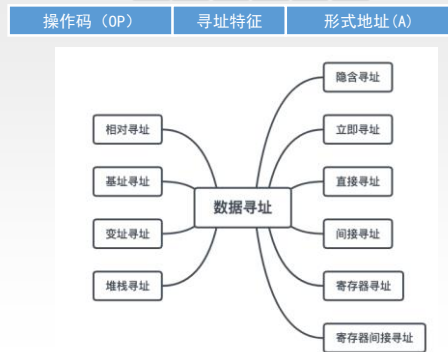
本节内容

指令系统

数据寻址-2
偏移寻址

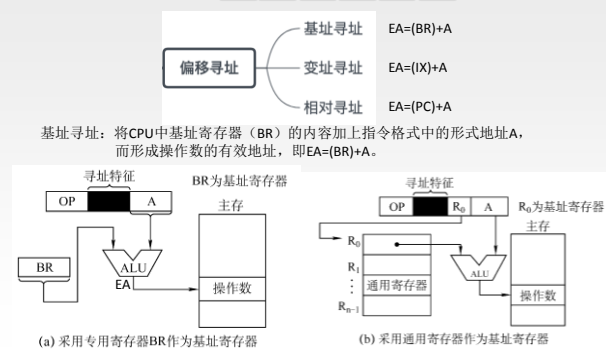
王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

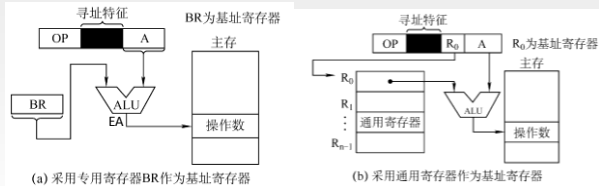
偏移寻址



王道考研/CSKAOYAN.COM

基址寻址

基址寻址：将CPU中基址寄存器（BR）的内容加上指令格式中的形式地址A，而形成操作数的有效地址，即 $EA=(BR)+A$ 。



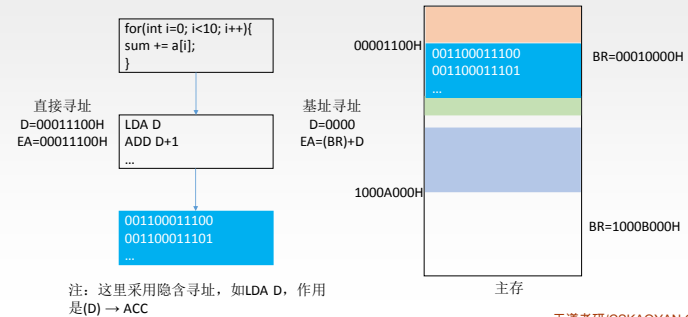
注：基址寄存器是面向操作系统的，其内容由操作系统或管理程序确定。在程序执行过程中，基址寄存器的内容不变（作为基地址），形式地址可变（作为偏移量）。当采用通用寄存器作为基址寄存器时，可由用户决定哪个寄存器作为基址寄存器，但其内容仍由操作系统确定。

优点：可扩大寻址范围（基址寄存器的位数大于形式地址A的位数）；用户不必考虑自己的程序存于主存的哪一空间区域，故有利于多道程序设计，以及可用于编制浮动程序。

王道考研/CSKAOYAN.COM

基址寻址

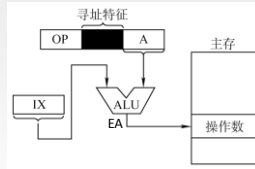
优点：可扩大寻址范围（基址寄存器的位数大于形式地址A的位数）；用户不必考虑自己的程序存于主存的哪一空间区域，故有利于多道程序设计，以及可用于编制浮动程序。



王道考研/CSKAOYAN.COM

变址寻址

变址寻址：有效地址EA等于指令字中的形式地址A与变址寄存器IX的内容相加之和，即 $EA = (IX) + A$ ，其中IX为变址寄存器（专用），也可用通用寄存器作为变址寄存器。



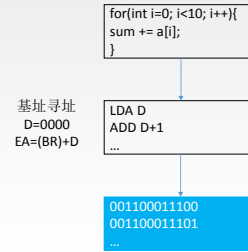
注：变址寄存器是面向用户的，在程序执行过程中，变址寄存器的内容可由用户改变（作为偏移量），形式地址A不变（作为基地址）。

优点：可**扩大寻址范围**（变址寄存器的位数大于形式地址A的位数）；在数组处理过程中，可设定A为数组的首地址，不断改变变址寄存器IX的内容，便可很容易形成数组中任一数据的地址，特别**适合编制循环程序**。

王道考研/CSKAOYAN.COM

基址寻址与变址寻址

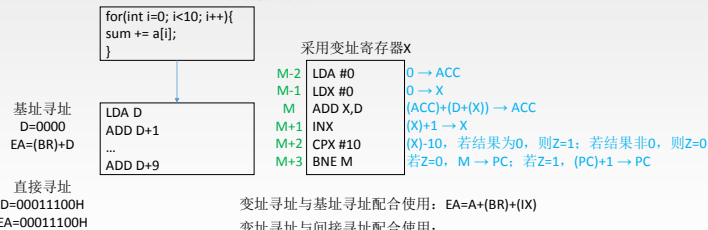
优点：可**扩大寻址范围**（变址寄存器的位数大于形式地址A的位数）；在数组处理过程中，可设定A为数组的首地址，不断改变变址寄存器IX的内容，便可很容易形成数组中任一数据的地址，特别**适合编制循环程序**。



王道考研/CSKAOYAN.COM

基址寻址与变址寻址

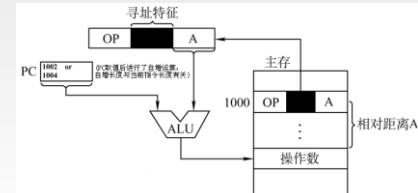
优点：可**扩大寻址范围**（变址寄存器的位数大于形式地址A的位数）；在数组处理过程中，可设定A为数组的首地址，不断改变变址寄存器IX的内容，便可很容易形成数组中任一数据的地址，特别**适合编制循环程序**。



王道考研/CSKAOYAN.COM

相对寻址

相对寻址：把程序计数器PC的内容加上指令格式中的形式地址A而形成操作数的有效地址，即 $EA = (PC) + A$ ，其中A是相对于当前指令地址的位移量，可正可负，补码表示。

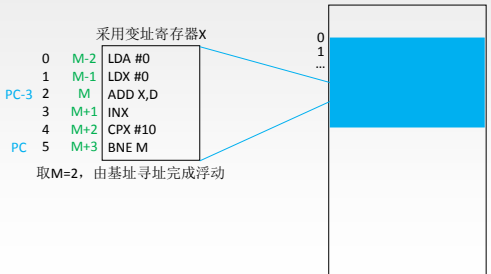


优点：操作数的地址不是固定的，它随着PC值的变化而变化，并且与指令地址之间总是相差一个固定值，因此便于程序浮动。
相对寻址广泛应用于**转移指令**。

王道考研/CSKAOYAN.COM

相对寻址

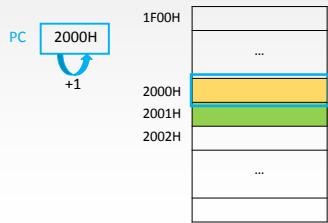
优点：操作数的地址不是固定的，它随着PC值的变化而变化，并且与指令地址之间总是相差一个固定值，因此便于程序浮动。
相对寻址广泛应用于转移指令。



王道考研/CSKAOYAN.COM

相对寻址举例

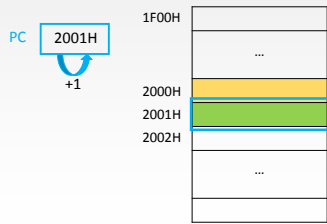
某机器指令字长为16位，主存按字节编址，取指令时，每取一个字节PC自动加1。当前指令地址为2000H，指令内容为相对寻址的无条件转移指令，指令中第一个字节为操作码，第二个字节为形式地址，当前形式地址为40H。
(1) 求取指令后及指令执行后PC内容。
(2) 若要求转移到1F00H，求形式地址的内容。



王道考研/CSKAOYAN.COM

相对寻址举例

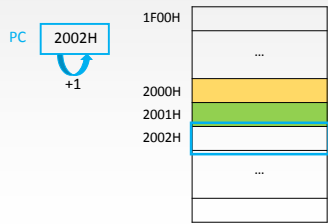
某机器指令字长为16位，主存按字节编址，取指令时，每取一个字节PC自动加1。当前指令地址为2000H，指令内容为相对寻址的无条件转移指令，指令中第一个字节为操作码，第二个字节为形式地址，当前形式地址为40H。
(1) 求取指令后及指令执行后PC内容。
(2) 若要求转移到1F00H，求形式地址的内容。



王道考研/CSKAOYAN.COM

相对寻址举例

某机器指令字长为16位，主存按字节编址，取指令时，每取一个字节PC自动加1。当前指令地址为2000H，指令内容为相对寻址的无条件转移指令，指令中第一个字节为操作码，第二个字节为形式地址，当前形式地址为40H。
(1) 求取指令后及指令执行后PC内容。
(2) 若要求转移到1F00H，求形式地址的内容。



CPU分析指令的结果：是一条无条件转移指令，
 $EA = (PC) + A$
 $= 2002H + 40H$
 $= 2042H$
故取指令后PC内容为2002H，指令执行后PC内容为2042H

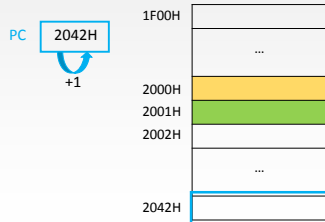
王道考研/CSKAOYAN.COM

相对寻址举例

某机器指令字长为16位，主存按字节编址，取指令时，每取一个字节PC自动加1。当前指令地址为2000H，指令内容为相对寻址的无条件转移指令，指令中第一个字节为操作码，第二个字节为形式地址，当前形式地址为40H。

(1) 求取指令后及指令执行后PC内容。

(2) 若要求转移到1F00H，求形式地址的内容。



CPU分析指令的结果：是一条无条件转移指令，
 $EA = (PC) + A$
 $= 2002H + 40H$
 $= 2042H$
 故取指令后PC内容为2002H，指令执行后PC内容为2042H

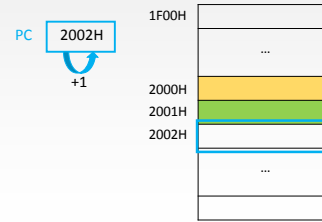
王道考研/CSKAOYAN.COM

相对寻址举例

某机器指令字长为16位，主存按字节编址，取指令时，每取一个字节PC自动加1。当前指令地址为2000H，指令内容为相对寻址的无条件转移指令，指令中第一个字节为操作码，第二个字节为形式地址，当前形式地址为40H。

(1) 求取指令后及指令执行后PC内容。

(2) 若要求转移到1F00H，求形式地址的内容。



即 $(PC) + A = 2002H + A = 1F00H$
 $A = 1F00H - 2002H = 1EFFH + 1H - 2002H$
 $= 1EFFH - 2002H + 1H = FEFDH + 1H = FEFEH$
 CPU分析指令的结果：是一条无条件转移指令，
 $EA = (PC) + A$
 $= 2002H + 40H$
 $= 2042H$
 故取指令后PC内容为2002H，指令执行后PC内容为2042H

王道考研/CSKAOYAN.COM

本节回顾

寻址方式	有效地址	访存次数(指令执行期间)
隐含寻址	程序指定	0
立即寻址	A即是操作数	0
直接寻址	$EA = A$	1
一次间接寻址	$EA = (A)$	2
寄存器寻址	$EA = R_i$	0
寄存器间接一次寻址	$EA = (R_i)$	1
转移指令 相对寻址	$EA = (PC) + A$	1
多道程序 基址寻址	$EA = (BR) + A$	1
循环程序 变址寻址 数组问题	$EA = (IX) + A$	1

偏移寻址

王道考研/CSKAOYAN.COM

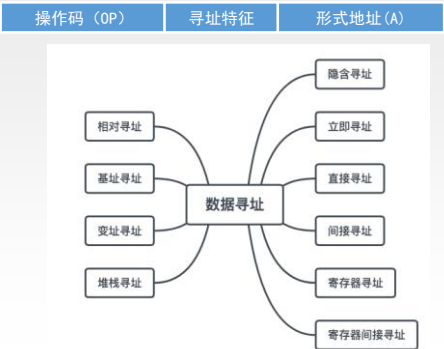
本节内容

指令系统

数据寻址-3
堆栈寻址

王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

堆栈寻址

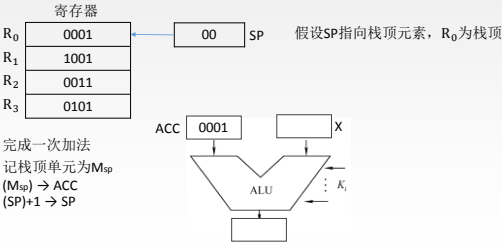
堆栈寻址：操作数存放在堆栈中，隐含使用堆栈指针(SP)作为操作数地址。
堆栈是存储器（或专用寄存器组）中一块特定的按“后进先出（LIFO）”原则管理的存储区，该存储区中被读/写单元的地址是用一个特定的寄存器给出的，该寄存器称为堆栈指针（SP）。



王道考研/CSKAOYAN.COM

堆栈寻址

堆栈寻址：操作数存放在堆栈中，隐含使用堆栈指针(SP)作为操作数地址。
堆栈是存储器（或专用寄存器组）中一块特定的按“后进先出（LIFO）”原则管理的存储区，该存储区中被读/写单元的地址是用一个特定的寄存器给出的，该寄存器称为堆栈指针（SP）。

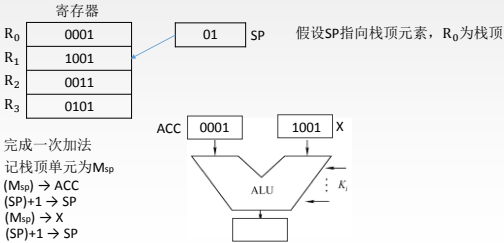


POP

王道考研/CSKAOYAN.COM

堆栈寻址

堆栈寻址：操作数存放在堆栈中，隐含使用堆栈指针(SP)作为操作数地址。
堆栈是存储器（或专用寄存器组）中一块特定的按“后进先出（LIFO）”原则管理的存储区，该存储区中被读/写单元的地址是用一个特定的寄存器给出的，该寄存器称为堆栈指针（SP）。



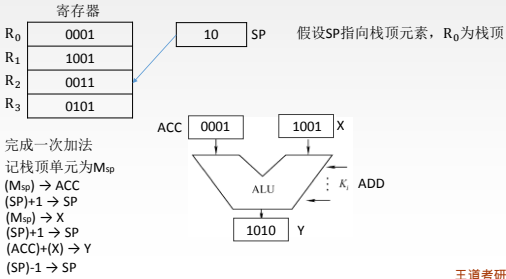
POP

POP

王道考研/CSKAOYAN.COM

堆栈寻址

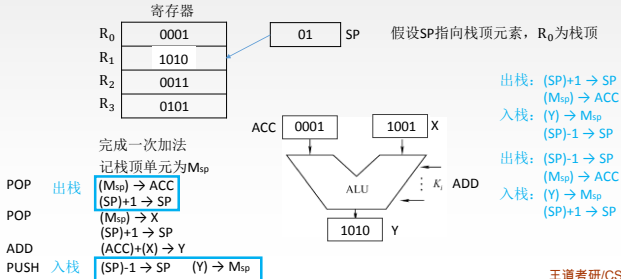
堆栈寻址：操作数存放在堆栈中，隐含使用堆栈指针(SP)作为操作数地址。
堆栈是存储器（或专用寄存器组）中一块特定的按“后进先出（LIFO）”原则管理的存储区，该存储区中被读/写单元的地址是用一个特定的寄存器给出的，该寄存器称为堆栈指针（SP）。



王道考研/CSKAOYAN.COM

堆栈寻址

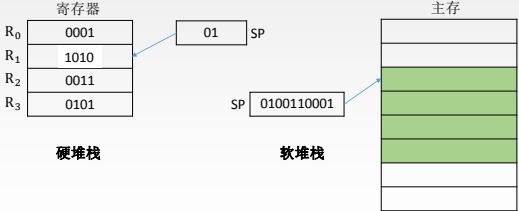
堆栈寻址：操作数存放在堆栈中，隐含使用堆栈指针(SP)作为操作数地址。
堆栈是存储器（或专用寄存器组）中一块特定的按“后进先出（LIFO）”原则管理的存储区，该存储区中被读/写单元的地址是用一个特定的寄存器给出的，该寄存器称为堆栈指针（SP）。



王道考研/CSKAOYAN.COM

堆栈寻址

堆栈寻址：操作数存放在堆栈中，隐含使用堆栈指针(SP)作为操作数地址。
堆栈是存储器（或专用寄存器组）中一块特定的按“后进先出（LIFO）”原则管理的存储区，该存储区中被读/写单元的地址是用一个特定的寄存器给出的，该寄存器称为堆栈指针（SP）。



王道考研/CSKAOYAN.COM

本节回顾

寻址方式	有效地址	访存次数(指令执行期间)
隐含寻址	程序指定	0
立即寻址	A即是操作数	0
直接寻址	EA=A	1
一次间接寻址	EA=(A)	2
寄存器寻址	EA=R _i	0
寄存器间接一次寻址	EA=(R _i)	1
转移指令 相对寻址	EA=(PC)+A	1
多道程序 基址寻址	EA=(BR)+A	1
循环程序 变址寻址 数组问题	EA=(IX)+A	1

偏移寻址

王道考研/CSKAOYAN.COM

本节内容

指令系统

CISC和RISC

王道考研/CSKAOYAN.COM

本章总览



王道考研/CSKAOYAN.COM

CISC和RISC

CISC: Complex Instruction Set Computer
设计思路：一条指令完成一个复杂的基本功能。

代表：x86架构，主要用于笔记本、台式机

80-20规律：典型程序中 80% 的语句仅仅使用处理器中 20% 的指令

比如设计一套能输出单词的指令集：

CISC的思路：每个单词的输出由一条指令完成
一条指令可以由一个专门的电路完成：
17万个单词=17万个电路
→ 采用“存储程序”的设计思想，由一个比较通用的电路配合存储部件完成一条指令

RISC: Reduced Instruction Set Computer
设计思路：一条指令完成一个基本“动作”；
多条指令组合完成一个复杂的基本功能。

代表：ARM架构，主要用于手机、平板等

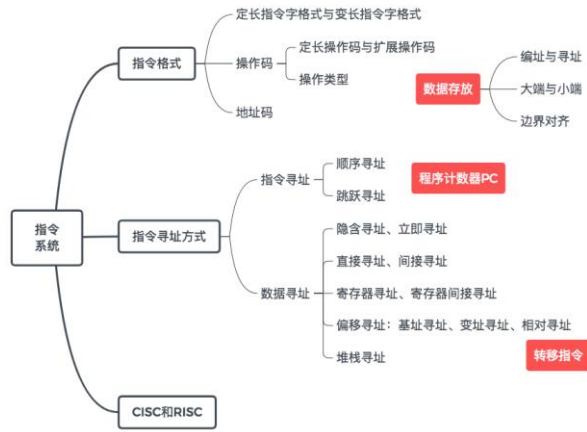
RISC的思路：每个字母的输出由一条指令完成，
多条指令组合完成一个单词
26个字母=26个电路
“并行”、“流水线”

王道考研/CSKAOYAN.COM

CISC和RISC

对比项目\类别	CISC	RISC
指令系统	复杂，庞大	简单，精简
指令数目	一般大于200条	一般小于100条
指令字长	不固定	定长
可访存指令	不加限制	只有Load/Store指令
各种指令执行时间	相差较大	绝大多数在一个周期内完成
各种指令使用频度	相差很大	都比较常用
通用寄存器数量	较少	多
目标代码	难以用优化编译生成高效的目标代码程序	采用优化的编译程序，生成代码较为高效
控制方式	绝大多数为微程序控制	绝大多数为组合逻辑控制
指令流水线	可以通过一定方式实现	必须实现

王道考研/CSKAOYAN.COM



J.COM