

第4章 指令系统



(一) 指令格式

1. 指令的基本格式
2. 定长操作码指令格式
3. 扩展操作码指令格式

(二) 寻址方式

1. 有效地址的概念
2. 数据寻址和指令寻址
3. 常见寻址方式

(三) CISC 和 RISC 的基本概念

海纳百川 自强不息 厚德笃学 知行合一



4.1 指令格式



➤ **指令**：指示计算机执行某项基本操作（运算或处理）功能的命令（二进制代码串），是计算机运行的最小功能单位。

➤ **指令系统**：一台计算机能直接理解与执行的全部指令的集合称为该计算机的指令系统，也叫机器语言。

- 指令系统是进行计算机逻辑设计和编制程序的基本依据。它直接说明了这台计算机的功能。
- 不同类型CPU的指令系统是不能混用与兼容的，但同一系列的CPU一般升级后指令都有扩充，并可兼容。
- 计算机硬件的核心作用是实现每条指令的功能，执行用机器指令组成的程序。

海纳百川 自强不息 厚德笃学 知行合一



设计指令系统考虑的因素

- 完备性，常用指令齐全，编程方便。
- 高效性，程序占内存空间少，运行速度快。
- 规整性，指令和数据使用规则简单，易学易记。
- 兼容性，同一系列的低档计算机的程序能在新的高档机上直接运行。

核心问题：选定指令的功能和格式。

海纳百川 自强不息 厚德笃学 知行合一





9
周年

4.1.1 指令的结构

- 指令字：代表指令的一组二进制代码信息。
- 指令字长：指令字中二进制代码的位数。

{ 定长指令格式
变长指令格式

便于控制

合理利用存储空间

典型的指令格式

操作码OP

地址码A

操作码 OP——指明操作性质的命令码，
提供指令的操作控制信息。

操作对象A——说明操作数存放的地址，
有时则就是操作数本身

自强不息 厚德笃学 知行合一



4.1.2 操作数地址字段的表示与编码



不同指令使用不同数目、不同来源去向、不同用法的操作数，要尽量把它们统一并安排在指令字的操作数地址字段。

指令中提供的地址 { 寄存器编号
内存单元地址码
I/O端口地址码

(1) 指令提供地址的方式 直接或间^{直接}接给出

显地址方式：指令中明显指明地址。

隐地址方式：地址隐含约定，不出现在指令中。

使用隐地址可以减少指令中的地址数，简化地址结构。

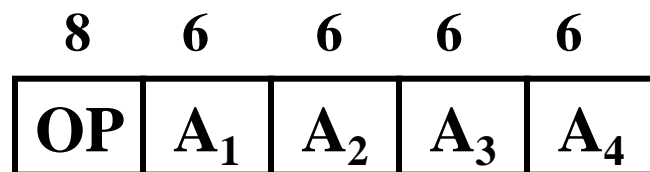
海纳百川 自强不息 厚德笃学 知行合一



(2) 操作数个数的类型

(a) 四地址指令

设指令字长为 32 位，操作码固定为 8 位



A₁ 第一操作数地址

A₂ 第二操作数地址

程序中大部分指令是顺序执行的，可以用程序计数器PC来指明下一条指令地址，因此可以去掉A₄字段。



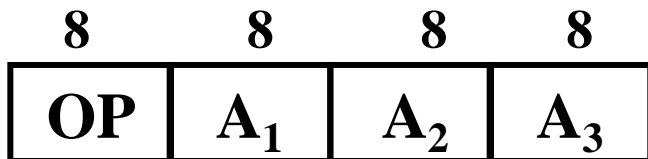
中华人民共和国成立70周年



1949—2019
大连理工大学 建校70周年

(b) 三地址指令

设指令字长仍为 32 位，操作码和地址码均为 8 位



$(A_1) \text{ OP } (A_2) \longrightarrow A_3$

寻址范围 $2^8 = 256$

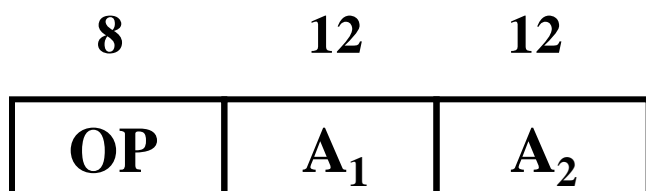
例：[A1]+[A2]→[A3]，其中[A1]、[A2]为源地址，[A3]为目的地址，操作后源地址内容不变，仅被拷贝。

如果A3用A1或者A2代替，那么A3可以省略。



(c) 二地址指令

设指令字长仍为 32 位，操作码8位，地址码12 位



4 次访存

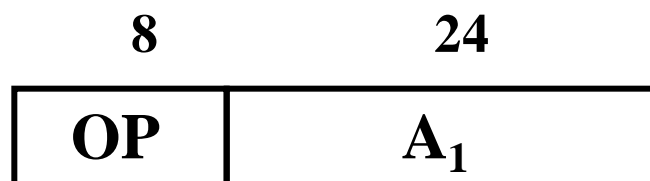
如果一个操作数可以隐含为ACC，则指令中可以只给出一个地址码。

寻址范围 $2^{12} = 4 \text{ K}$

海纳百川 自强不息 厚德笃学 知行合一

(d) 一地址指令

设指令字长仍为 32 位，操作码8位，地址码24 位



$(ACC) OP (A_1) \longrightarrow ACC$

2 次访存

寻址范围 $2^{24} = 16 M$

例如：递增，移位，取反，
INC AX,

地址数目



(4) 零地址指令 也叫无操作数指令

设指令字长仍为 32 位。

OP

例如：空操作NOP、停机HLT、关中断CLI

海纳百川 自强不息 厚德笃学 知行合一



4.1.3 操作码字段的格式



(1) 定长操作码

各指令OP的位置、位数固定相同。

优点：有利于简化硬件设计，减少译码时间。

应用：在字长较长的大中型及超级小型机上使用。

(2) 变长操作码

各指令OP的位置、位数不固定，根据需要变化。

关键在设置扩展标志。

特点：在不增加指令字长度的情况下可表示更多的指令，但增加了译码和分析难度，需更多硬件支持。

应用：微机中常使用此方式。

海纳百川 自强不息 厚德笃学 知行合一



例：指令字长16位，定长格式操作码可含有3、2、1或0个地址，每个地址占4位。

操作码		地址码		
15~12	11~8	7~4	3~0	
0000	X	Y	Z	
⋮	⋮	⋮	⋮	
1110	X	Y	Z	
1111	0000	Y	Z	
⋮	⋮	⋮	⋮	
1111	1110	Y	Z	
1111	1111	0000	Z	
⋮	⋮	⋮	⋮	
1111	1111	1110	Z	
1111	1111	1111	0000	
⋮	⋮	⋮	⋮	
1111	1111	1111	1111	

三地址指令 15条

二地址指令 15条

一地址指令 15条

零地址指令 16条

1111——扩展操作码标志

例： 指令字长16位，用可变格式操作码分别形成三地址指令、二地址指令、一地址指令和零地址指令15、14、31、16条。结果之一如下：

15~ 12 11~ 8 7 ~ 4 3 ~ 0

0000	X	Y	Z
⋮	⋮	⋮	⋮
1110	X	Y	Z
1111	0000	Y	Z
⋮	⋮	⋮	⋮
1111	1101	Y	Z
1111	1110	0000	Z
⋮	⋮	⋮	⋮
1111	1111	1110	Z
1111	1111	1111	0000
⋮	⋮	⋮	⋮
1111	1111	1111	1111

三地址指令 15条

二地址指令 14条

一地址指令 31条

零地址指令 16条

1111——扩展操作码标志

4.2 寻址方式



- 寻址方式是指确定本条指令的数据地址以及下一条将要执行的指令的地址的方式，与硬件结构密切相关。分为**指令寻址**和**数据寻址**两大类。
 - 指令寻址分为顺序寻址和跳跃寻址两种。
 - ✓ 顺序寻址可以通过程序计数器PC加1自动形成下一条指令的地址。
 - ✓ 跳跃寻址则通过转移类指令实现，是通过对PC的运算得到新的下一条指令的地址。
 - 数据寻址方式：是指**寻找操作数地址或操作数的方式**。即规定如何对地址字段作出解释以找到操作数。
 - **有效地址**：操作数的真实地址，记做EA，它是寻址方式和形式地址共同来决定的。

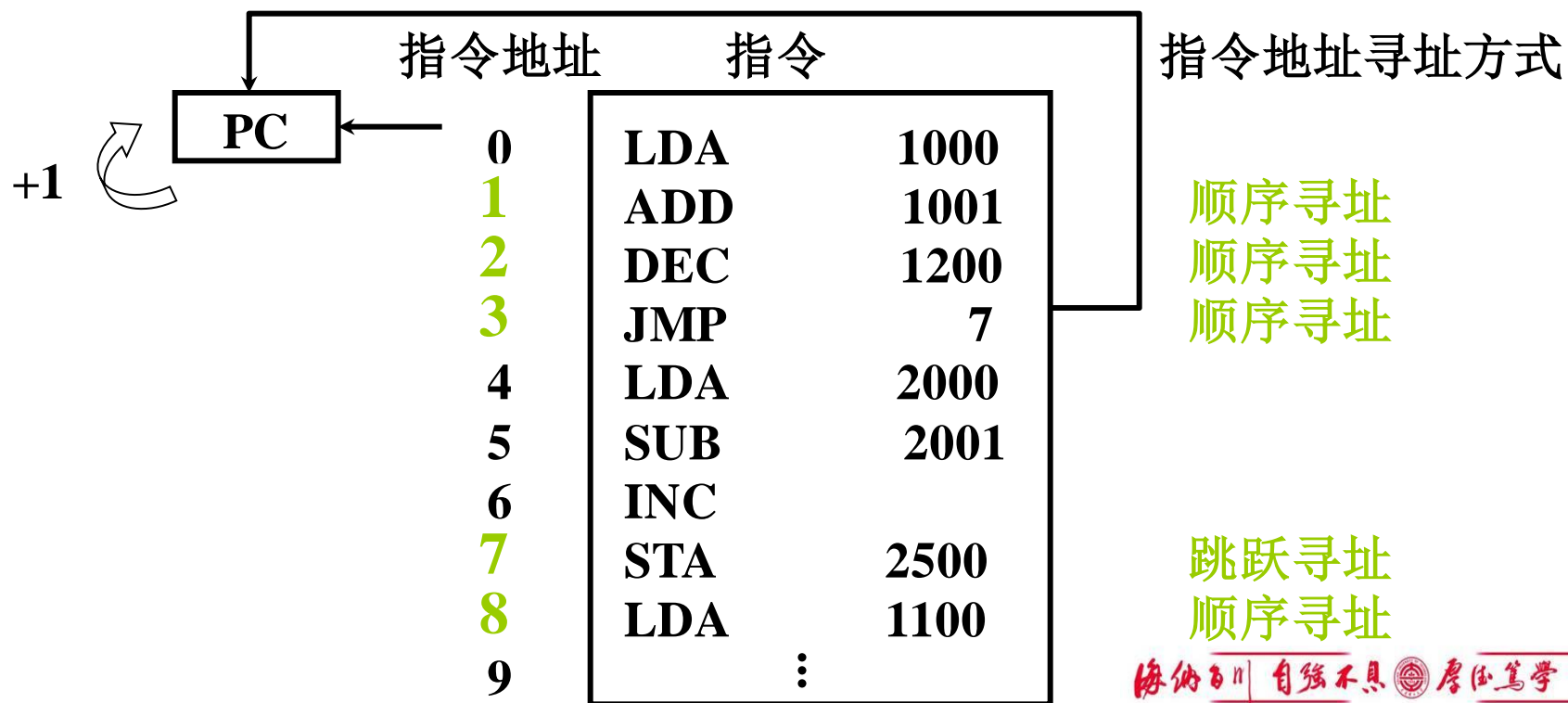
海纳百川 自强不息 厚德笃学 知行合一



指令寻址

顺序 $(PC) + 1 \longrightarrow PC$

跳跃 由转移指令指出



海纳百川 自强不息 厚德笃学 知行合一

数据寻址

数据寻址有多种，需要在指令中明确指出采用哪一种寻址方式，可以专门设置一个寻址方式特征字段，或纳入操作码中。

操作码	寻址特征	形式地址 A
-----	------	--------

形式地址 指令字中的地址

有效地址 操作数的真实地址

有效地址由形式地址根据寻址方式来确定。

约定 指令字长 = 存储字长 = 机器字长

海纳百川 自强不息 厚德笃学 知行合一



4.2.1 常见寻址方式

1. 立即寻址

指令直接给出操作数。

操作码OP	立即数A
-------	------

用来提供常数、设置初值等。

例： MOV AX, #20H

- 指令执行阶段不访存
- A 的位数限制了立即

数的范围

2. 直接寻址



指令的操作数地址字段直接给出操作数在内存中的地址。
(数在内存中)

操作码OP	有效地址A
-------	-------

$$S = (A)$$

例：MOV AX, [20H]

- 执行阶段访问一次存储器，用于访问固定的存储单元。
- A 的位数限制了该指令操作数的寻址范围

海纳百川 自强不息 厚德笃学 知行合一



3. 寄存器寻址

格式

操作码OP	寄存器号R
-------	-------

$$S = (R)$$

给出操作数所在的寄存器的编号。

执行阶段不访存，只访问寄存器，执行速度快
寄存器个数有限，可缩短指令字长

4. 间接寻址

指令给出操作数的间接地址。

＜ 存储单元号(数在存储器中)
寄存器号 (数在寄存器中)

指令中的形式地址
不是操作数的地址
，而是“操作数地
址的地址”。

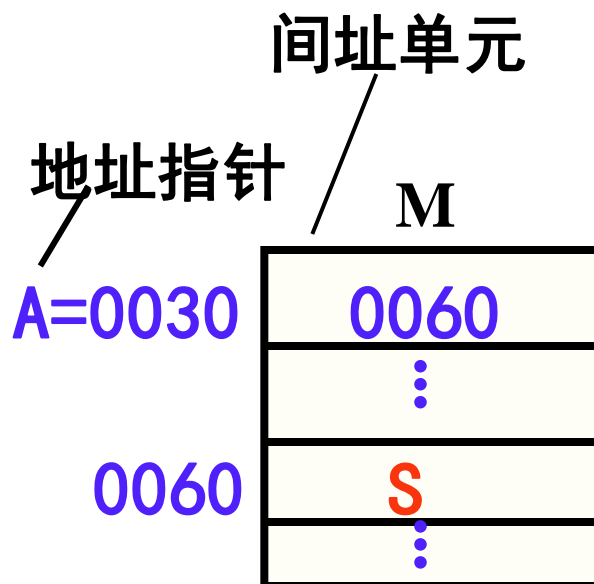
(1) 存储器间址

格式

操作码OP

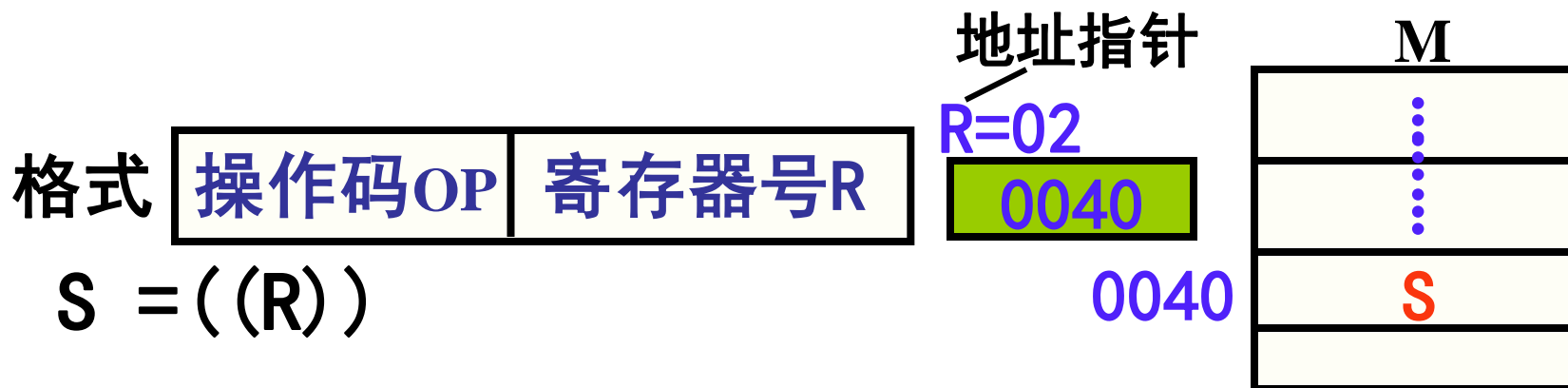
间接地址A

$$S = ((A))$$



海纳百川 自强不息 厚德笃学 知行合一

(2) 寄存器间址



指令中的形式地址为寄存器的**编号**，所占位数少；
R可提供全字长地址码；修改R内容比修改M内容快。

指针不变(由指令指定)，指针内容可变，使同一指令
可指向不同存储单元。

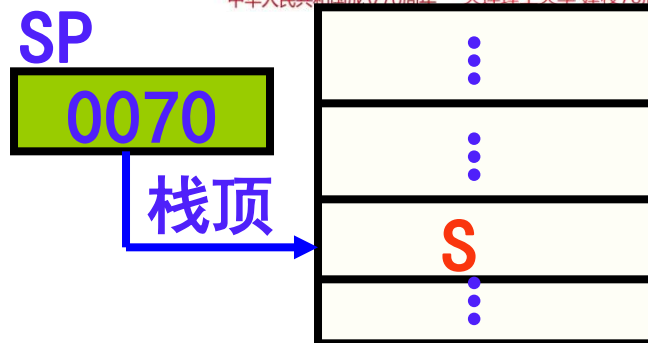
便于实现程序的循环、共享，并提供转移地址

(3) 堆栈寻址

格式

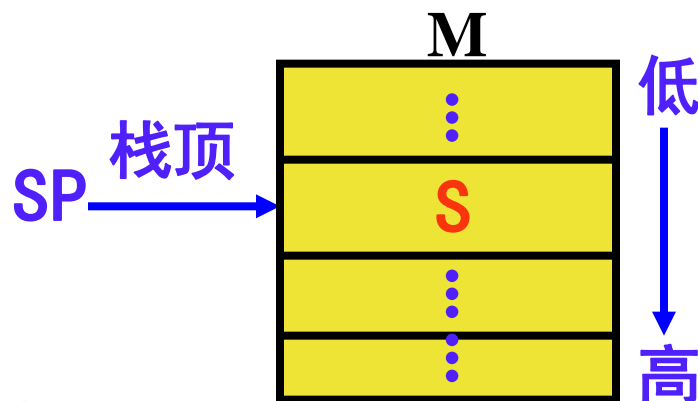
操作码OP	堆栈指针SP
-------	--------

$$S = ((SP))$$



堆栈向上生成

- 压栈: SP自动减1, 再存数。
 $-(SP)$, 自减型间址。
- 出栈: 先取数, SP再自动加1。
 $(SP)+$, 自增型间址。



SP既可出现在指令中, 也可隐含约定。

海纳百川 自强不息 厚德笃学 知行合一

5. 变址、基址寻址及其变化

(1) 变址寻址

指令给出一个寄存器号和一个基准地址，寄存器内容与基准地址之和为有效地址。



变址寄存器号

形式地址

$$S = (RX + A)$$

修改量

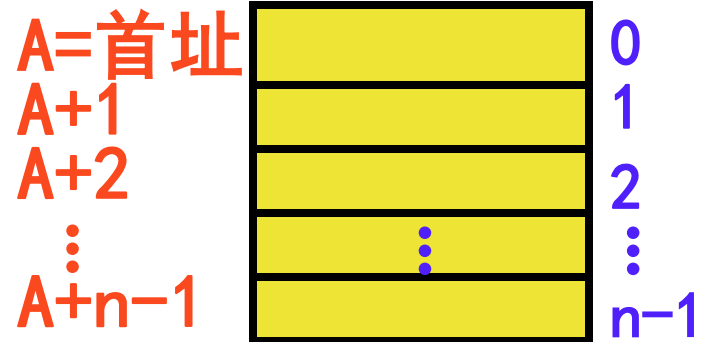
基准地址

例：用变址方式访问一组连续区间内的数组元素。

A为存储区首址；

(RX) 为所访单元距离
首址的长度；

RX初值为0，每访问一
个单元， $(RX)+1$ 。



A的位数有限，若不能提供全字长地址码，会使
访存空间受到限制。

(2) 基址寻址

指令给出一个寄存器号和一个地址量，寄存器内容与地址量之和为有效地址。



基址寄存器号

位移量

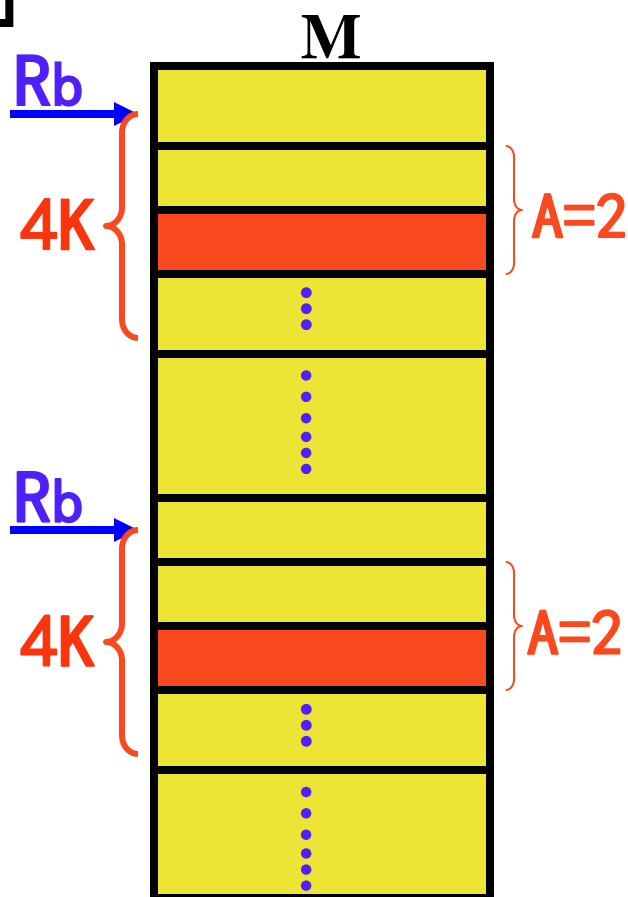
$$S = (Rb) + A$$

基准地址

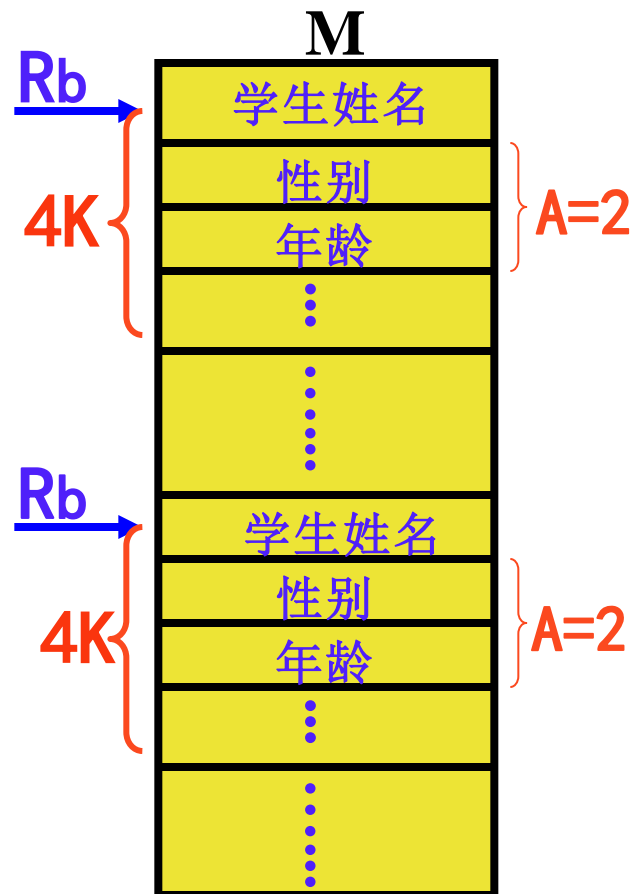
相对于基址的位移

改变Rb的内容，程序能访问存储空间中任何一个定长区间(如：4K)。

改变Rb的内容，程序能访问存储空间中任何一个与基址保持相同距离的单元。



便于访问二维数组中某类指定的元素。



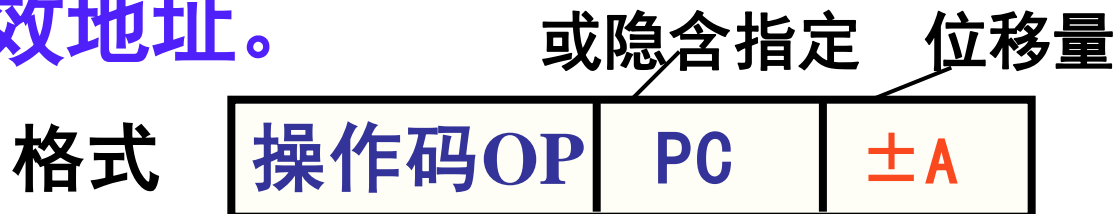
变址与基址的区别：

变址：指令提供**基准量**（不变）， R 提供**修改量**（可变）；适于处理一维数组。

基址：指令提供**位移量**（不变）， R 提供**基准量**（可变）；用于扩大有限字长指令的访存空间和程序装配。
解决程序的存储定位问题

(3) 相对寻址

指令给出位移量，PC内容与位移量之和为有效地址。



有效地址相对PC上下浮动, 应用于相对转移指令。适用于浮动程序的装配与运行。

$$S = (PC) \pm A$$

补码

例：某机器字长16位，主存按字节编码，转移指令采用相对寻址，由两个字节组成，第一字节为操作码字段，第二字节为相对位移量字段。假定取指令时，每取一个字节PC自动加1。某转移指令所在主存地址为2000H，相对位移量字段的内容为06H（或FCH），求该转移指令成功转移后的目标地址。

解：取出该指令后，源地址=PC+2=2002H，相对位移量A=06H，目标地址=源地址+A=2002H+0006H=2008H。

若A=FCH，目标地址=源地址+A（符号扩展）
=2002H+FFFCH=1FFE H

4.2.2 对寻址方式的说明

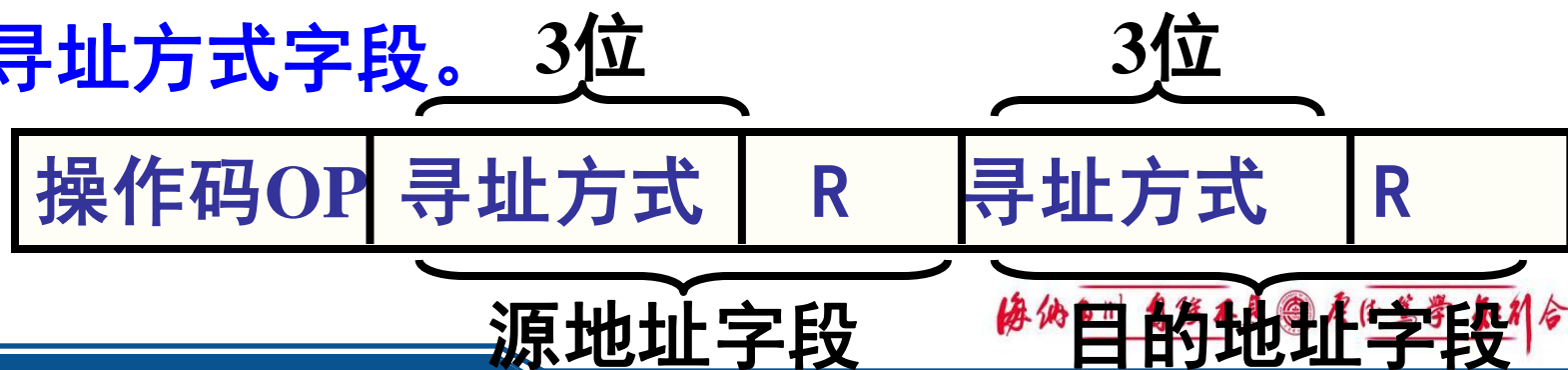
1. 操作码隐含说明不同寻址方式

例：某机指令操作码最高两位

- 00: RR型指令，寄存器-寄存器寻址
- 01: RX型指令，寄存器-变址寻址
- 10: SI型指令，基址-立即寻址
- 11: SS型指令，基址-基址寻址

2. 指令中设置专门字段说明寻址方式

例：某机指令的每个地址字段中各设置一个3位的寻址方式字段。



4.3 指令类型



- 1、数据传送类指令
- 2、运算类指令
- 3、移位指令
- 4、转移指令
- 5、输入输出指令
- 6、其他指令

海纳百川 自强不息 厚德笃学 知行合一



4.3 操作类型

1、数据传送类指令

源 目的

寄存器 寄存器

MOV AX, BX

寄存器 存储器

MOV [20], AX

STORE指令

存储器 寄存器

MOV AX, [20]

LOAD指令

存储器 存储器

MOV [20], [30]

堆栈操作: **PUSH AX**

POP AX

清零、置1: **MOV AX, #0**

MOV AX, #1



2、运算类指令



算术运算：加、减、乘、除、求补、浮点、十进制运算

ADD AX, #20

DIV AX, #3

逻辑运算：与、或、非、异或

AND AX, #30

XOR AX, #30

其他：位测试、位清除、位求反

海纳百川 自强不息 厚德笃学 知行合一



3、移位指令



一般来说，有**8种**移位指令

算术左移、算术右移、逻辑左移、逻辑右移

小循环左移、小循环右移

大循环左移、大循环右移

海纳百川 自强不息 厚德笃学 知行合一



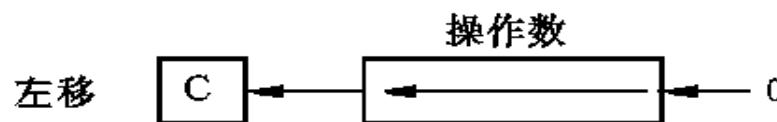


新中国成立70周年

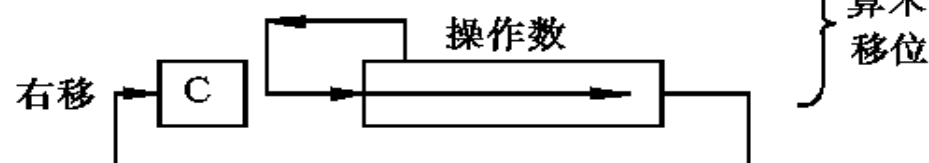


1949—2019
大连理工大学 建校70周年

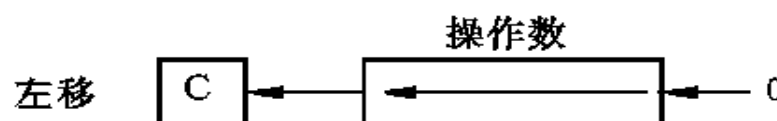
SAL



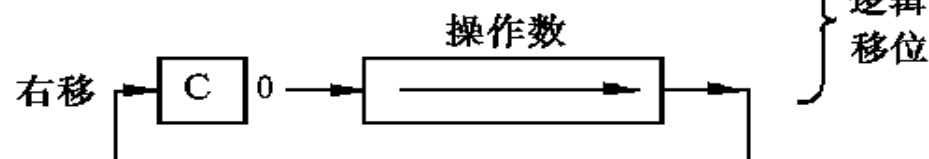
SAR



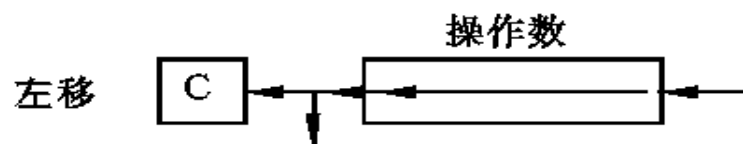
SHL



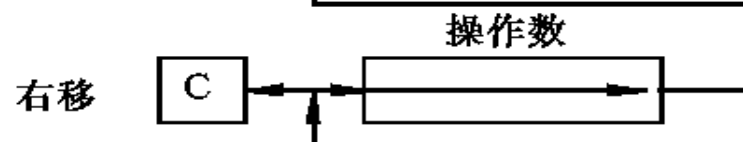
SHR



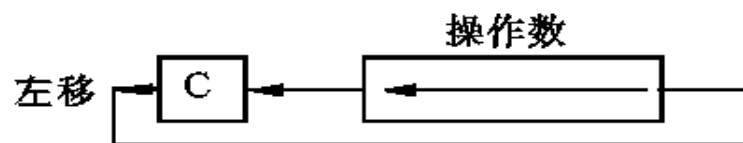
ROL



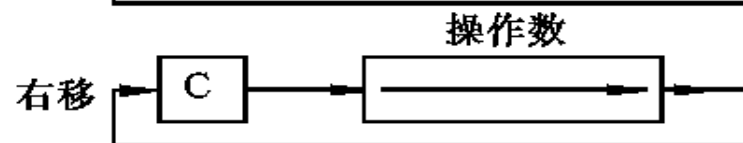
ROR



RCL



RCR



移位操作

循环移位

4、转移指令

A、无条件转移指令

直接跳转到某处，不取决于任何条件。类似C中的goto语句。例如：**JMP LOOP**

B、条件转移指令

根据机器当前的程序状态字中的某位来决定是否执行转移。例如：**JZ LOOP**

I	T										F	Z	S	O	C
---	---	--	--	--	--	--	--	--	--	--	---	---	---	---	---

PSW



4、转移指令



程序状态字PSW (FLAG) 是CPU内部的一个寄存器，用来存放两类信息：

- 1、体现当前指令执行结果的各种状态信息，如有无进位（CF位），有无溢出（OF位），结果正负（SF位），结果是否为零（ZF位），奇偶标志位（PF位）等；状态位全部自动设置。
- 2、存放控制信息，如允许中断（IF位），跟踪标志（TF位）等。

I	T										F	Z	S	O	C
---	---	--	--	--	--	--	--	--	--	--	---	---	---	---	---



4、转移指令

C、调用与返回指令：类比C程序中的函数调用，以及函数返回。例如：CALL PRO1、RET

子程序调用需要**注意**以下几点：

1. 子程序可以在多处被调用
2. 子程序调用可以嵌套
3. **CALL**与**RET**指令配对使用
4. 要妥善保存子程序的返回地址、专用寄存器、堆栈





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

子程序调用过程

地址

2000

2100

2101

主程序

⋮
CALL SUB1
⋮

子程序SUB1

2400

2500

2501

2560

2561

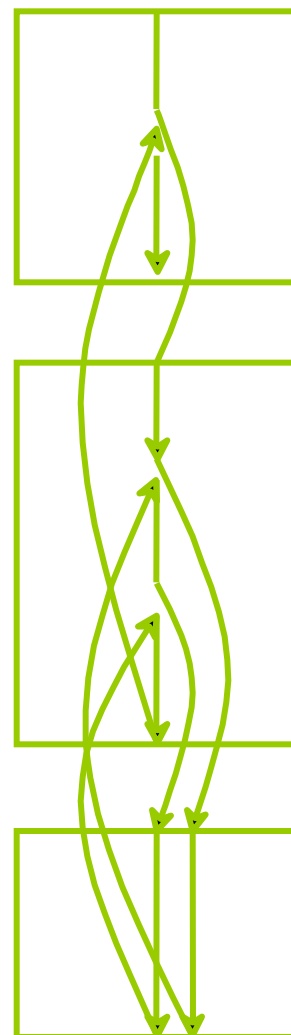
⋮
CALL SUB2
⋮
CALL SUB2
⋮
RETURN

子程序SUB2

2700

⋮
RETURN

主存空间分配



程序执行流程

海纳百川 自强不息 厚德笃学 知行合一



4、转移指令



D、陷阱（TRAP）指令

一旦机器运行出现意外故障（未定义指令、除0、设备故障、电压不稳），计算机发出陷阱信号（陷阱隐指令），暂停当前指令的执行，转入故障处理程序。陷阱指令不提供给用户使用，由机器自动执行。

也有某些机器提供陷阱指令，例如IBM—PC提供的INT × ×软中断指令，用来完成系统调用。

海纳百川 自强不息 厚德笃学 知行合一



5、输入输出指令

对I/O单独编址的计算机，设置有专门的输入输出指令，用来操纵外设。例如：

IN AX, [20]

OUT DX, AX

端口地址 \longrightarrow CPU 的寄存器

CPU 的寄存器 \longrightarrow 端口地址

独立编址 vs 统一编址

6、其他指令



等待指令、停机指令、空操作指令、开中断指令、关中断指令、置条件码指令。

字符串传送、字符串比较、字符串查询

特权指令（操作系统用）

向量指令

多处理机指令

海纳百川 自强不息 厚德笃学 知行合一



```

01 DATAS SEGMENT
02     CHAR DB 'Please enter the capital letters:',0AH,0DH,'$';此处输入数据段代码
03 DATAS ENDS
04
05 STACKS SEGMENT
06     ;此处输入堆栈段代码
07 STACKS ENDS
08
09 CODES SEGMENT
10     ASSUME CS:CODES,DS:DATAS,SS:STACKS
11 START:
12     MOV AX,DATAS
13     MOV DS,AX
14
15     MOV DX,OFFSET CHAR ;输出Please enter the capital letters:语句
16     MOV AH,09H
17     INT 21H
18
19 NEXT:
20     MOV AH,08H ;键盘输入
21     INT 21H
22     MOV DL,AL
23     MOV AH,02H
24     INT 21H
25
26     CMP AL,0DH ;如果是回车键退出程序
27     JZ S3
28
29     CMP AL,'A' ;判断输入的字母是否为大写字母
30     JB STOP
31     CMP AL,'Z'
32     JA STOP

```





1949—2019
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

```
33
34  ADD AL,20H ;如果为大写字母通过加20H, 转化为小写字母
35  MOV DL,AL
36  MOV AH,02H
37  INT 21H
38  STOP:
39  JMP NEXT
40
41  ;此处输入代码段代码
42  S3:
43  MOV AH,4CH ;返回结束指令
44  INT 21H
45  CODES ENDS
46  END START
47
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TEMPFILE

Please enter the capital letters:

▲

海纳百川 自强不息 厚德笃学 知行合一



● **INT 21**是计算机中断的一种,不同的**AH**值表示不同的中断功能。

AH	功能	调用参数
00	程序终止(同INT 20H)	CS=程序段前缀
01	键盘输入并回显	AL=输入字符
02	显示输出	DL=输出字符
03	异步通讯输入	AL=输入数据
04	异步通讯输出	DL=输出数据
05	打印机输出	DL=输出字符
06	直接控制台I/O	DL=FF(输入) DL=字符(输出)
07	键盘输入(无回显)	AL=输入字符
08	键盘输入(无回显) 检测Ctrl-Break	AL=输入字符
09	显示字符串	DS:DX=串地址 '\$'结束字符串
0A	键盘输入到缓冲区	DS:DX=缓冲区首地址 (DS:DX)=缓冲区最大字符数
0B	检验键盘状态	AL=00 有输入 AL=FF 无输入

4.4 CISC和RISC简介



CISC和RISC的基本概念

- **CISC: 复杂指令系统计算机 (Complex Instruction Set Computer)**，从增强指令系统功能出发，指令功能复杂。
- **RISC: 精简指令系统计算机 (Reduced Instruction Set Computer)**，从提高指令执行效率出发，指令功能简单。

海纳百川 自强不息 厚德笃学 知行合一



CISC的主要特征



- 指令系统庞大复杂，各种指令使用频度差别大
 - 指令长度不固定、指令格式种类多，寻址方式多
 - 访存指令不受限制
 - CPU中设有专用寄存器
 - 大多数指令需要多个时钟周期执行完毕
 - 采用微程序控制器
-
- 英特尔生产的x86系列（也就是IA-32架构）CPU及其兼容CPU，如AMD、VIA的，即使是现在新起的X86-64（也被称为AMD64）都是属于CISC的范畴

海纳百川 自强不息 厚德笃学 知行合一



RISC的由来



➤ 计算机的不断升级扩充，同时又兼容过去产品使指令系统日趋复杂，形成了“复杂指令系统计算机(CISC)”。如VAXII / 780有303条指令，18种寻址方式。Pentium机有191条，9种寻址方式。

➤ 复杂指令系统增加硬件复杂性，降低机器运行速度。经实际分析发现：

- 各种指令使用频率相差悬殊。80%指令使用较少。
- 指令系统的复杂性带来系统结构的复杂性，增加了设计时间和售价，也增加了VLSI设计负担，不利于微机向高档机器发展。
- 复杂指令操作复杂、运行速度慢。

由此提出“精简指令系统计算机(RISC)”的概念。

海纳百川 自强不息 厚德笃学 知行合一



RISC的特点



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

RISC不是简单地简化指令系统，而是通过简化指令使计算机的结构更加简单合理，从而提高机器性能。

- 优先选取使用频率最高的一些简单指令；
- 指令长度固定；
- 只有取数/存数指令（load/store）访问内存；
- CPU中的寄存器数量很多；
- 大部分指令在一个或小于一个机器周期完成；
- 硬布线控制逻辑为主，不用或少用微码控制；
- 一般用高级语言编程，特别重视编译优化，以减少程序执行时间。

海纳百川 自强不息 厚德笃学 知行合一



RISC和CISC的比较



中华人民共和国成立70周年



大连理工大学 建校70周年

- RISC更能充分利用VLSI芯片的面积
- RISC更能提高计算机运算速度
 - 指令数目、指令格式、寻址方式少
 - 通用寄存器多，采用组合逻辑
 - 便于流水线操作
- RISC便于设计、成本低、可靠性高
- RISC有利于编译程序代码优化
- RISC不易实现指令系统兼容

海纳百川 自强不息 厚德笃学 知行合一



CISC机与RISC机的主要特征对比



	CISC	RISC
指令系统 指令数 指令格式 指令字长 寻址方式 可访问指令 各种指令使用频率 各种指令执行时间	复杂、庞大 一般大于200 一般大于4 一般大于4 不固定 不加限制 相差很大 相差很大	简单、精简 一般小于100 一般小于4 一般小于4 固定 只有LOAD/STORE指令 相差不大 绝大多数在一个机器周期完成
优化编译实现	很难	较容易
程序源代码长度	较短	较长
控制逻辑实现方式	绝大多数为微程序控制	绝大多数为硬连线控制

海纳百川 自强不息 厚德笃学 知行合一



CISC机与RISC机



- 简单指令集的CPU在处理一些特定的运算时速度远高于复杂指令集，所以它常被用在工业领域，比如专用的仪器仪表、某些软件的专用服务器，流水线操作等方面。
- 复杂指令集则有更广泛的功能，适用性更广。现在的所有个人多媒体计算机都是用的复杂指令集CPU。

海纳百川 自强不息 厚德笃学 知行合一



RISC计算机举例



中华人民共和国成立70周年 大连理工大学建校70周年

- **PowerPC:** IBM/Apple/Motorola, 可伸缩性好、方便灵活

Performance Optimized With Enhanced RISC

PowerPC 处理器有非常强的嵌入式表现, (任天堂 **Gamecube** 使用了 **PowerPC**) 因为它具有优异的性能、较低的能量损耗以及较低的散热量。

- **MIPS:** MIPS的意思是“无内部互锁流水级的微处理器”(Microprocessor without interlocked piped stages), 其机制是尽量利用软件办法避免流水线中的数据相关问题。它最早是在80年代初期由斯坦福(**Stanford**)大学**Hennessy**教授领导的研究小组研制出来的。
- **SPARC:** **SUN Microsystems**,可扩充处理器架构。**SPARC**微处理器最突出的特点就是它的可扩展性,这是业界出现的第一款有可扩展性功能的微处理器。**SPARC**的推出为**SUN**赢得了高端微处理器市场的领先地位。

海纳百川 自强不息 厚德笃学 知行合一



ARM

(Advanced RISC Machine)

- 授权ARM内核给生产和销售半导体的合作伙伴
 - ARM 公司不生产芯片
 - IP(Intelligence Property)
- 另外也提供基于ARM架构的开发设计技术
 - 软件工具, 评估板, 调试工具, 应用软件,
 - 总线架构, 外围设备单元, 等等



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

海纳百川 自强不息 厚德笃学 知行合一



ARM公司特点



公司的业务是一个纯粹的知识产权的贩卖者，没有硬件，没有软件，只有图纸上的知识产权。

- 全球领先的原始设备制造商 (OEM) 都在广泛使用 **ARM** 技术，应用领域涉及手机、数字机顶盒以及汽车制动系统和网络路由器。

- 当今，全球 **95%** 以上的手机以及超过四

ARM The Architecture for the Digital World® **ARM 技术。**

海纳百川 自强不息 厚德笃学 知行合一





龙芯2F

600MHz~1GHz
64位高性能处理器

News 龙芯新闻

- 龙芯3A开始量产 龙芯2G流片成
- NEW 龙芯梦兰在江苏完成六万套龙
- “发展中国家信息化与可持续
- NEW 龙芯抢占国产数字电视芯片先
- NEW 升腾携手龙芯 助力保险行业信
- NEW 大灾有大爱 龙芯助舟曲

更多新闻>>

Product 产 品



芯 片



IP



解决方案

Design 设计服务

田 芯片物理设计

田 IP与SOC定制

田 软件开发

田 硬件设计

田 成功案例

Technology 龙芯技术

田 架构

田 平台

田 标准

田 软件

田 应用

田 评测

Q&A
龙芯直通车

Download
资料下载

About Us

北京龙芯中科技术
服务中心有限公司

龙芯3A

- 龙芯3A的工作频率为**900MHz~1GHz**，频率为**1GHz**时双精度浮点运算速度峰值达到每秒**160**亿次，单精度浮点运算速度峰值每秒**320**亿次。龙芯3A采用意法半导体公司（**STMicro**）**65**纳米**CMOS**工艺生产，晶体管数目达**4.25**亿个，芯片采用**BGA**封装，引脚的数目为**1121**个，功耗小于**15**瓦。

- 龙芯3A集成了四个64位超标量处理器核、4MB的二级Cache、两个DDR2/3内存控制器、两个高性能HyperTransport控制器、一个PCI / PCIX控制器以及LPC、SPI、UART、GPIO等低速I/O控制器。龙芯3A的指令系统与MIPS64兼容并通过指令扩展支持X86二
进制翻译。



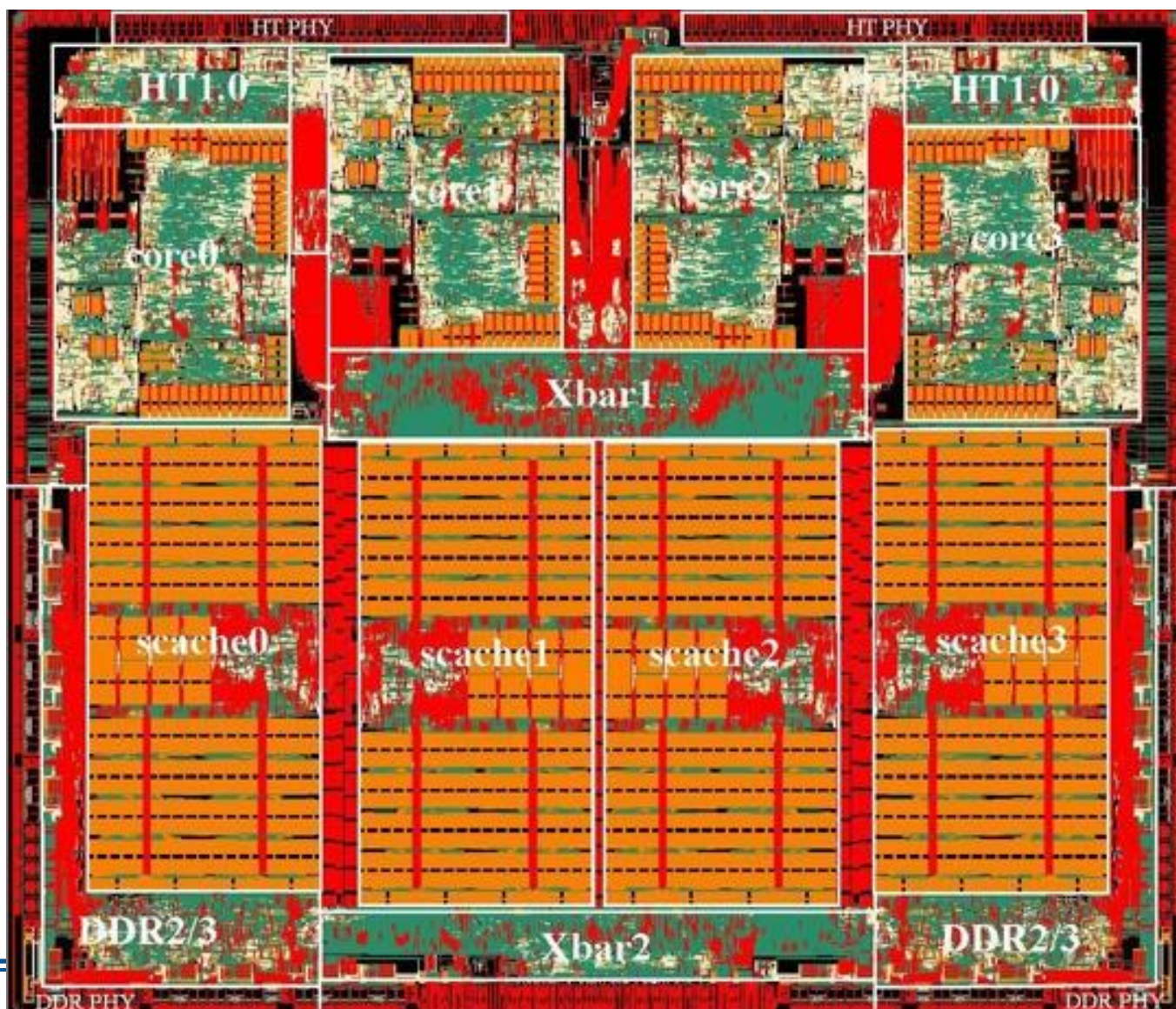
龙芯3A



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



博学 知行合一

同步练习



一、单选题

1、采用直接寻址，操作数在（ ）中。

A、主存 B、寄存器 C、硬盘 D、光盘

答案：A

2、为了缩短指令中某个地址段的位数，有效的方法是采取（ ）。

A、立即寻址 B、变址寻址 C、间接寻址 D、寄存器寻址

答案：D

3、指令系统采用不同寻址方式的目的是（ ）。

A、实现存储程序和程序控制

B、缩短指令长度，扩大寻址空间，提高编程灵活性

C、可以直接访问外存

D、提供扩展操作码的可能，并降低指令译码难度

答案：B

海纳百川 自强不息 厚德笃学 知行合一



4、假设寄存器R中的数值为200，主存地址为200和300的地址单元中存放的内容分别是300和400，则（ ）方式下访问到的操作数为200。

- A、直接寻址200
- B、寄存器间接寻址（R）
- C、存储器间接寻址（200）
- D、寄存器寻址R

答案：D

5、下列关于RISC的描述中，不正确的是（ ）。

- A、只选用简单的，使用频率高的指令，所以采用RISC的CPU的性能较差，不能运行复杂的指令
- B、指令长度固定，指令格式和寻址方式少
- C、采用硬布线逻辑，提高指令执行的速度
- D、采用指令流水线技术，大部分指令都能在一个机器周期中完成。

答案：A

海纳百川 自强不息 厚德笃学 知行合一



二、综合应用题



某台字长和地址都为16位的计算机，程序计数器为PC，内存以字编址。地址为2003H的内存中，有一个无条件相对转移指令，其机器码为41FCH，其中的操作码为8位，**请计算相对转移的具体地址。**

解：根据题意，机器码**41FCH**为一个字长，取指后，程序计数器为“**2004H**”；

相对转移指令中的第2个字节为偏移量**FCH**，为8位，则符号位扩展后的**16位**偏移量是**FFFCH**；

转移的目标地址计算：目的地址**PC=源地址PC+16位位移量=2004H+FFFCH=2000H**，进到高位的1自动丢失。

所以转移的具体地址为**2000H**。

思考：若内存以字节编址呢？

海纳百川 自强不息 厚德笃学 知行合一



真题解析

一、单选题：

- 1、（2013）某机器字长 16 位，主存按字节编址，转移指令采用相对寻址，由两个字节组成，第一字节为操作码字段，第二字节为相对位移量字段。假定取指令时，每取一个字节 PC 自动加 1。若转移指令所在主存地址为 2000H，相对位移量字段的内容为 06H，则该转移指令成功转移后的目标地址是（ ）。
- A. 2006H B. 2007H C. 2008H D. 2009H

答案：C

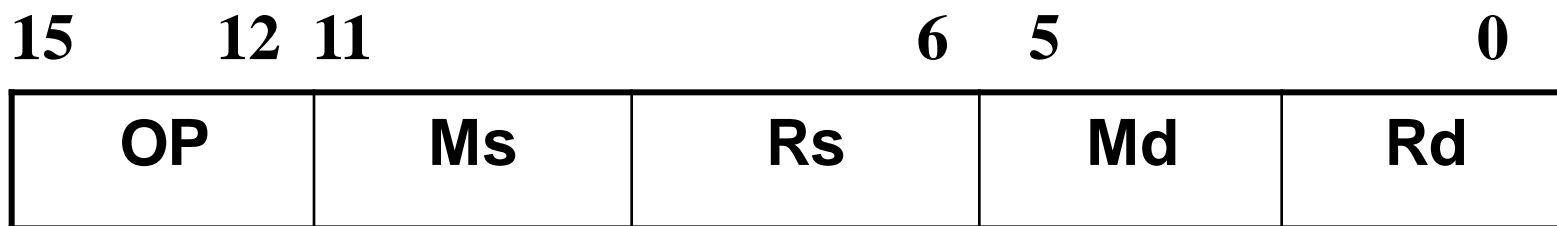
- 2、（2012）下列关于 RISC 的叙述中错误的是（ ）。
- A. RISC 普遍采用微程序控制器
- B. RISC 大多数指令在一个时钟周期内完成
- C. RISC 的内部通用寄存器数量相对 CISC 多
- D. RISC 的指令数、寻址方式和指令格式种类相对 CISC 少

答案：A



二、综合应用题:

(11分) (2010) 某计算机字长为16位, 主存地址空间大小为128KB, 按字编址, 采用定长指令格式, 指令名字段定义如下:



源操作数

目的操作数



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

转移指令采用相对寻址方式，相对偏移是用补码表示，寻址方式定义如下：

Ms/Md	寻址方式	助记符	含义
000B	寄存器直接	Rn	操作数= (Rn)
001B	寄存器间接	(Rn)	操作数= ((Rn))
010B	寄存器间接、自增	(Rn) +	操作数= ((Rn)) , (Rn) +1 → Rn
011B	相对	D (Rn)	转移目标地址= (PC) + (Rn)

注：(X) 表示存储地址X或寄存器X的内容，请回答下列问题：

海纳百川 自强不息 厚德笃学 知行合一



- (1) 该指令系统最多可有多少条指令？该计算机最多有多少个通用寄存器？存储器地址寄存器（MAR）和存储器数据寄存器（MDR）至少各需多少位？
- (2) 转移指令的目标地址范围是多少？
- (3) 若操作码0010B表示加法操作（助记符为add），寄存器R4和R5的编号分别为100B和101B，R4的内容为1 2 3 4 H，R5的内容为5 6 7 8 H，地址1 2 3 4 H中的内容为5 6 7 8 H，5 6 7 8 H中的内容为1 2 3 4 H，则汇编语言为add（R4），（R5）+（逗号前为源操作数，逗号后为目的操作数）对应的机器码是什么（用十六进制表示）？该指令执行后，哪些寄存器和存储单元的内容会改变？改变后的内容是什么？

解：该题的考点是指令系统设计，注意操作码位数与指令条数的关系，地址码与寄存器数的关系，指令字长与MDR的关系，存储容量与MAR的关系。

注意寻址方式的定义，指令格式各字段的编码、含义。

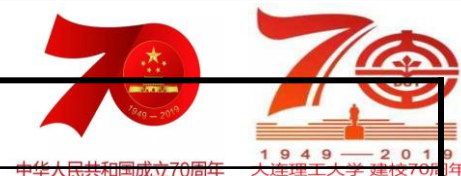
(1) 该指令系统最多可有多少条指令？

- 指令格式中操作码占15~12位共4位， $2^4=16$ ，可知该指令系统最多可有16条指令。
- 通用寄存器字段中源和目的寄存器Rs和Rd的编码都是3位， $2^3=8$ ，所以，该计算机最多有8个通用寄存器。 (寄存器寻址)
- 计算机字长为16位，且采用字长指令格式，所以，存储器地址寄存器 (MAR) 和存储器数据寄存器 (MDR) 至少都需16位。

(2) 主存地址空间大小为128KB，按字编址为64K，转移目标地址= (PC) + (Rn)，PC为16位，16位相对偏移d16用补码表示，d16: -32768—+32767，转移指令的目标地址范围是：
0 — $2^{16}-1$ 即0 — 65535 的64K空间。

(3)

Ms/Md	寻址方式	助记符	含义
000B	寄存器直接	Rn	操作数= (Rn)
001B	寄存器间接	(Rn)	操作数= ((Rn))
010B	寄存器间接、自增	(Rn) +	操作数= ((Rn)) , (Rn) +1 → Rn
011B	相对	D (Rn)	转移目标地址= (PC) + (Rn)



add (R4) , (R5) + ; ((R4)) + ((R5)) → ((R5)) ,
; (R5) +1 → R5, 对应的各位指令代码为:

OP	Ms	Rs	Md	Rd
0010	001	100	010	101

即该指令对应的机器码是: 2315H

该指令执行后, R5寄存器和存储单元 5 6 7 8 H的内容会改变。

改变后, R5自增1, (R5) = 5 6 7 9H,

(1234H) + (5678H) = 5678H+1234H=68ACH → (5678H)

存储单元 5 6 7 8 H的内容变为: 6 8ACH

海纳百川 自强不息 厚德笃学 知行合一



常见寻址方式



- 1. 立即寻址
- 2. 直接寻址
- 3. 寄存器寻址
- 4. 间接寻址 (存储器间址、寄存器间址)
- 5. 变址寻址
- 6. 基址寻址
- 7. 相对寻址

操作码OP

地址码A

海纳百川 自强不息 厚德笃学 知行合一



4.2.1 常见寻址方式

1. 立即寻址

操作码OP	立即数A
-------	------

指令直接给出操作数。

- 指令执行阶段不访存
- A 的位数限制了立即数的范围

2. 直接寻址

指令的操作数地址字段直接给出操作数在存储器中的地址。(数在存储器M中)

操作码OP	有效地址A
-------	-------

- A 的位数限制了该指令操作数的寻址范围

- 执行阶段访问一次存储器，内存。

海纳百川 自强不息 厚德笃学 知行合一

3. 寄存器寻址

格式

操作码OP

寄存器号R



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

$$S = (R)$$

给出操作数所在的寄存器的编号。

执行阶段不访存，只访问寄存器，执行速度快
寄存器个数有限，可缩短指令字长

4. 间接寻址

指令给出操作数的间接地址。

＜ 存储单元号(数在存储器中)
寄存器号 (数在存储器中)

指令中的形式地址
不是操作数的地址
，而是“操作数地
址的地址”。

(1) 存储器间址

格式

操作码OP

间接地址A

$$S = ((A))$$

• 执行阶段访问两次内存



(2) 寄存器间址

格式

操作码OP	寄存器号R
-------	-------


$$S = ((R))$$

指令中的形式地址为寄存器的**编号**，所占位数少；R可提供全字长地址码。

4堆栈寻址

格式

操作码OP	堆栈指针SP
-------	--------

$$S = ((SP))$$

海纳百川 自强不息 厚德笃学 知行合一



5 变址寻址

变址寄存器号

格式

操作码OP

RX

变址偏移量A

$$S = ((RX) + A)$$

修改量

基准地址

A的位数有限，若不能提供全字长地址码，会使访存空间受到限制。

6 基址寻址

格式

操作码OP

Rb

A

基址寄存器号

位移量

$$S = ((Rb) + A)$$

基准地址

相对于基址的位移

改变Rb的内容，程序能访问存储空间中任何一个定长区间(如：4K)。



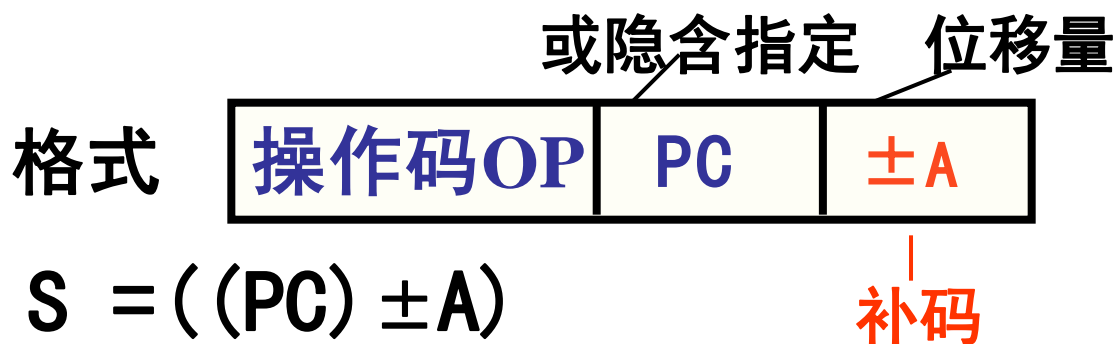
中华人民共和国成立70周年



1949—2019
大连理工大学 建校70周年

7相对寻址

指令给出位移量，PC内容与位移量之和为有效地址。



有效地址相对PC上下浮动，应用于相对转移指令。适用于浮动程序的装配与运行。

海纳百川 自强不息 厚德笃学 知行合一

