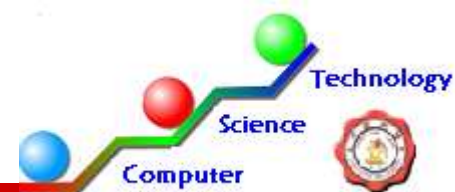


# 计算机组成与设计

## 第二章习题解

## 第二章 2.4



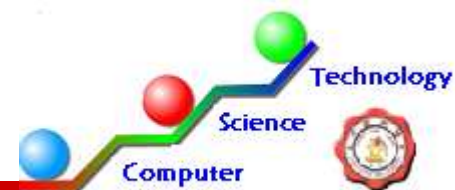
□ 2.4一般来说，CISC比RISC的指令复杂，因此可以用较少的指令完成相同的任务。然而，由于指令的复杂，一条CISC指令需要花费比RISC更多的时间来完成。假设一个任务需要P条CISC指令或者2P条RISC指令，完成每条CISC指令花费8Tns，每条RISC指令花费2Tns。在此假设下，哪一种指令系统性能更好？

□ 题解：

○  $P \times 8 > 2P \times 2$

○ RISC性能更好

## 第二章 2.5

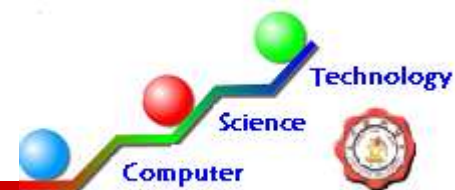


❑ 2.5 ASCII码是7位，如果设计主存单元字长为31位，指令字长为12位，是否合理？为什么？

❑ 题解

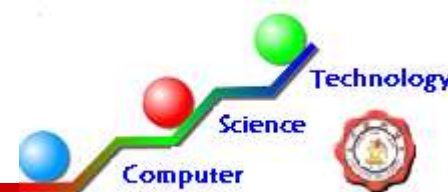
○ 此设计方案不合理。其原因是：① **ASCII码是7位**，通常加一位校验位为8位，以字节为单位进行处理比较方便。故主存应设计成按字节编址，这种编址方式下一般主存单元字长应取字节长度的2、4、8倍。若按8位标准字节设计，主存字长取32位比较合适，取31位显然不合理。② 一般指令字长应与机器字长或字节长度间有整数倍关系，若主存设计成按字节编址方式，则指令字长取单字节、双字节等较合适，取12位显然不合理。

## 第二章 2.6



- 2.6 在某些计算机中，子程序调用是以下述方法实现的：  
转子指令将返回地址（即主程序中该指令的下一条指令地址）存入子程序的第一单元，然后转到第二个单元开始执行子程序。
- （1）设计一条相应的从子程序返回主程序的指令；
  - （2）在这种情况下，你怎样在主、子程序间进行参数的传递？
  - （3）上述调用方法是否可用于子程序嵌套？
  - （4）上述调用方法是否可用于子程序多重嵌套时的递归调用（即某个子程序调用它本身）？如果改用堆栈链接方法，是否可实现此问题？

## 第二章 2.6



### □ 题解:

- (1) 返回指令是一地址指令，其格式如下：

JMP	I	K
-----	---	---

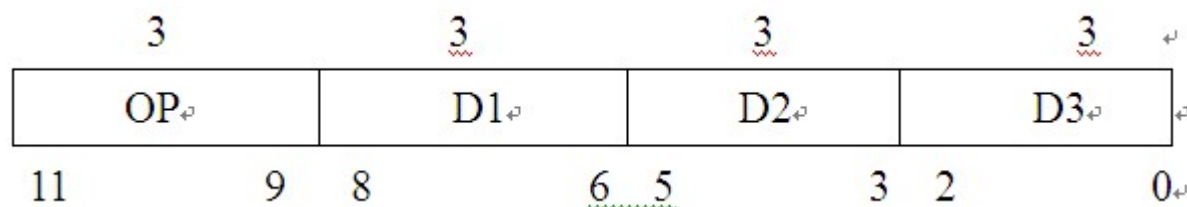
这是一条间接寻址的无条件转移指令。其中，I为间接寻址标志，K为子程序在主存第一单元的地址。

- (2) 在这种情况下，可利用寄存器或主存单元进行主、子程序之间的参数传递。
- (3) 可以用于子程序的嵌套(多重转子程序)，因为每个返回地址都存放在被调用的子程序的第一个单元中。
- (4) 不可以用于子程序的递归，因为当某个子程序自己调用自己时，子程序的第一个单元的内容将被破坏。  
如果改为堆栈方法，可以实现子程序的递归，因为堆栈具有后进先出的功能。

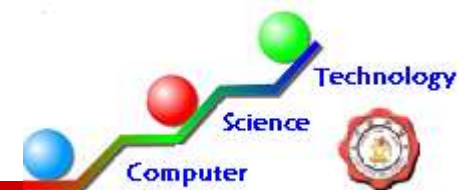
## 第二章 2.7



- 2.7 设某指令系统基本指令格式如下图所示。图中，指令总字长12位，其中OP表示操作码字段，占3位； $D_i$  ( $i=1, 2, 3$ ) 表示地址码字段，每个分别占3位。请利用扩展操作码法，试提出一种编码方案使该指令系统有5条三地址指令，8条二地址指令，120条单地址指令，60条零地址指令。要求具体分配每条指令的操作码编码。



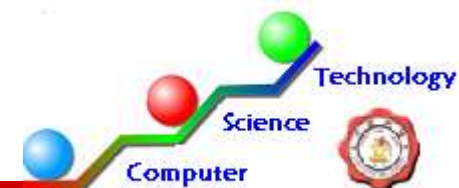
## 第二章 2.7



□ 解：该指令系统操作码编码分配方案如下：

000	XXX	YYY	ZZZ	}	5条三地址指令
↓	↓	↓	↓		
100	XXX	YYY	ZZZ	}	8条二地址指令
101	000	YYY	ZZZ		
↓	↓	↓	↓	}	120条单地址指令
101	111	YYY	ZZZ		
110	000	000	ZZZ	}	
↓	↓	↓	↓		
110	111	111	ZZZ	}	
111	000	000	ZZZ		
↓	↓	↓	↓	}	
111	110	111	ZZZ		

## 第二章 2.7

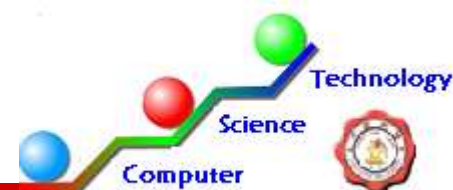


### □ 操作码编码分配方案 (续)

111	111	000	000	}	60条零地址指令
↓	↓	↓	↓		
111	111	110	111		
111	111	111	000		
↓	↓	↓	↓	}	4条冗余编码备用
111	111	111	011		
111	111	111	100		
↓	↓	↓	↓		
111	111	111	111		



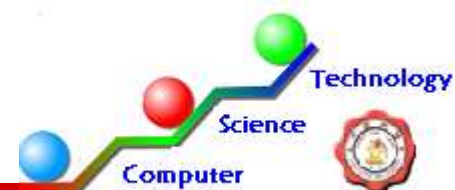
## 第二章 2.8



2.8 某32位计算机，CPU中有32个通用寄存器，主存容量为4GB。指令字长等于机器字长，若该机指令系统可完成138种操作，操作码位数固定，且具有立即寻址、直接寻址、间接寻址、寄存器间接寻址、变址寻址、基址寻址和相对寻址7种方式，试回答：  
(要求：答案中数据分别用2的幂形式表示)

- (1) 画出一地址指令格式，并指出各字段的作用；
- (2) 该指令立即数的最大范围；
- (3) 直接寻址的最大范围；
- (4) 一次间接寻址和多次间接寻址的寻址范围；
- (5) 寄存器间接寻址的范围；
- (6) 分别采用专用寄存器和通用寄存器作为变址寄存器时，变址寻址的位移量范围；
- (7) 分别采用专用寄存器和通用寄存器作为基址寄存器时，基址寻址的位移量范围；
- (8) 相对寻址的位移量。

## 第二章 2.8



### □ 题解

(1)            8            3            5            16

OP	M	R	A
----	---	---	---

(2) 立即数最多可以用21位，所以最大取值范围：

$-2^{20} \sim 2^{20}-1$  (有符号数) 或者  $0 \sim 2^{21}-1$  (无符号数)

(3) 直接寻址时，形式地址位数最多21位，所以寻址最大分范围：  
 $2^{21}\text{B} = 2\text{MB}$

(4) 一次间接寻址的寻址范围：4GB

多次间接寻址的寻址范围：2GB

(5) 寄存器间接寻址的范围：4GB

(6) 采用专用寄存器作为变址寄存器时，变址寻址的位移量范围：4GB  
采用通用寄存器作为变址寄存器时，变址寻址的位移量范围：4GB

## 第二章 2.8



(7) 采用专用寄存器作为基址寄存器时，基址寻址的位移量范围：

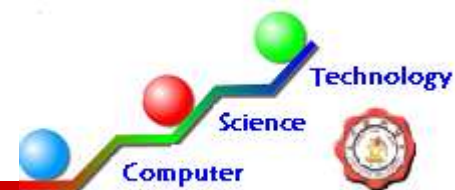
$-2^{20} \sim 2^{20}-1\text{B}$ ，即 $-1\text{M} \sim 1\text{M}-1\text{B}$

采用通用寄存器作为基址寄存器时，基址寻址的位移量范围：

$-2^{15} \sim 2^{15}-1\text{B}$ ，即 $-32\text{K} \sim 32\text{K}-1\text{B}$

(8) 相对寻址的位移量：  $-2^{20} \sim 2^{20}-1\text{B}$ ，即 $-1\text{M} \sim 1\text{M}-1\text{B}$

## 第二章 2.9



2.9 某16位机，采用单字长单地址指令格式，其中形式地址码字段占7位。若基址寄存器的内容为2000H，变址寄存器的内容为23A0H，指令的形式地址码部分是3FH，当前正在执行的指令所在地址为2B00H。请回答下列问题：

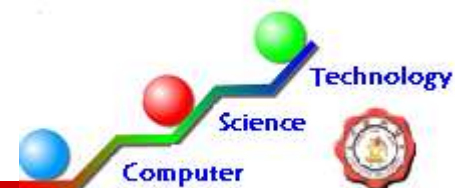
- (1) 变址寻址、基址寻址和相对寻址三种情况下的访存有效地址；
- (2) 设变址寻址用于取数指令，相对寻址用于转移指令，存储器内存放的相关内容如下：

地 址	内 容
003FH	2300H
2000H	2400H
203FH	2500H
2B3FH	2600H
23A0H	2700H
23DFH	2800H
2B00H	063FH

请写出从存储器中所取的数据以及转移地址。

- (3) 若采用直接寻址，请写出从存储器中取出的数据。

## 第二章 2.9



### □ 题解

○ (1) 相对寻址:  $EA = (PC) + A$   
 $= 2B01H + 003FH = 2B40H$

变址寻址:  $EA = (R_x) + A$   
 $= 23A0H + 003FH = 23DFH$

基址寻址:  $EA = (R_b) + A$   
 $= 2000H + 003FH = 203FH$

○ (2) 变址寻址时  $EA = 23DFH$   
因此从  $23DFH$  单元取出的数据为  $2800H$ ;  
相对寻址时转移地址 =  $2B40H$

○ (3) 直接寻址时  $EA = 003FH$ ,  
 $S(EA) = 2300$ , 因此取出的数据为  $2300H$

## 第二章 2.10



- 2.10 某计算机字长16位，主存按字编址，采用单字长单地址指令格式，指令的一般格式如下所示：

OP Code	I	X	D
---------	---	---	---

操作码      间址位      基址寄存器号      形式地址

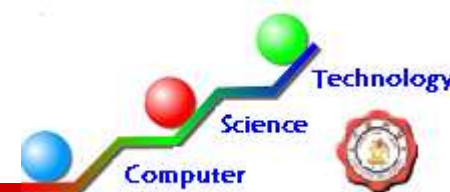
用户程序中某条指令K格式如下：

0	1	3	401
---	---	---	-----

主存某几个单元的内容如下：（参数均为十进制表示）

地 址	内 容
⋮	⋮
4016	3528
⋮	⋮
4300	2053
⋮	⋮
4416	1764
4417	4300
⋮	⋮

## 第二章 2.10



若3号基址寄存器内容是4016，试用先基址后间址（一次）的复合寻址方式，求指令K的操作数P。

□ 题解：

○ 先基址寻址： $EA' = (Rb) + A$   
 $= 4016 + 401 = 4417$

○ 后间接寻址： $EA = (EA') = (4417) = 4300$   
 $P = (4300) = 2053$

## 第二章 2.11



**2.11** 某计算机字长16位，主存按字编址，采用单字长单地址指令格式，其格式如下所示：

OP-Code	X	D
---------	---	---

**OP-Code**：操作码。**D**：形式地址。

**X**：寻址方式码，**X=00**：直接寻址；

**X=01**：用变址寄存器**X1**变址；

**X=10**：用变址寄存器**X2**变址；

**X=11**：相对寻址；

若执行指令时，机器状态如下：

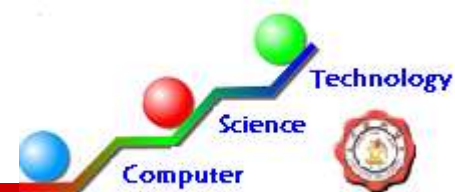
**(PC) = 1548H, (X1) = 036AH, (X2) = 46B2H**

请分别确定下列指令的有效地址**EA**。

① **3056H**    ② **42A0H**    ③ **1347H**    ④ **4598H**    ⑤ **67CEH**



## 第二章 2.11



### □ 题解:

○①指令码=0011 00 0 0 0101 0110, 直接寻址

$$EA=D=0101\ 0110B = 0056H$$

○②指令码=0100 00 1 0 1010 0000, 用变址寄存器X2变址

$$EA=(X2)+D = 46B2H + A0H = 4752H$$

○③指令码=0001 00 1 1 0100 0111, 相对寻址

$$EA=(PC)+D = 1548H + 47H = 158FH$$

○④指令码=0100 01 0 1 1001 1000, 用变址寄存器X1变址

$$EA=(X1)+D = 036AH + 98H = 0402H$$

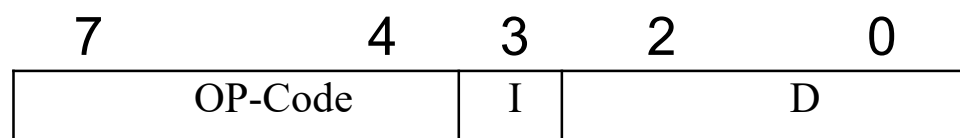
○⑤指令码=0110 01 1 1 1100 1110, 相对寻址

$$EA=(PC)+D = 1548H + FFCEH = 1516H$$

## 第二章 2.12



2.12 某8位计算机，其指令格式如下图所示：



其中，**OP-Code**为操作码；**I**为间址特征位，只允许一次间址；**D**为形式地址。假设主存储器部分单元内容如下：

地 址	内 容
00H	9DH
01H	04H
02H	A4H
03H	5EH
04H	15H
05H	76H
06H	B8H
07H	23H

指出下列指令的有效地址：

- ① A7H    ② DFH    ③ B2H    ④ CEH

## 第二章 2.12



□ 题解:

○①指令码=1010 0 111, 直接寻址

$EA=D=07H$

○②指令码=1101 1 111, 间接寻址

$EA=(D)=(07H)=23H$

○③指令码=1011 0 010, 直接寻址

$EA=D=02H$

○④指令码=1100 1 110, 间接寻址

$EA=(D)=(06H)=B8H$

## 第二章 2.13



- 2.13 某计算机字长16位，主存按字编址，采用单字长单地址指令格式，指令各字段定义如下：

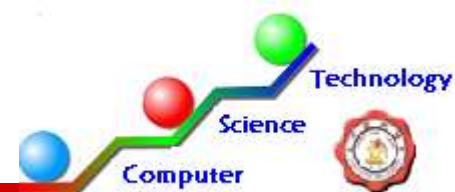
15	12	11	9	8	6	5	0
OP-Code		M		Rn		A	

其中，OP-Code为操作码，M为寻址方式码，Rn为通用寄存器编号，A为形式地址。寻址方式码定义如下：

M	寻址方式	有效地址表达式
000B	一次间接	$EA = (A)$
001B	寄存器间接	$EA = (Rn)$
010B	变址	$EA = (Rn) + A, Rn \leftarrow (Rn) + 1$
011B	相对	$EA = (PC) + A$

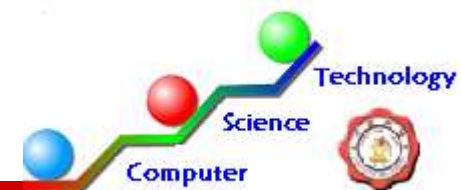
注：有效地址表达式中(X)表示存储器地址X或寄存器X的内容；指令中Rn字段和A字段是否使用视寻址方式而定；位移量用补码表示。

## 第二章 2.13



- ❑ 请回答下列问题：
- ❑ (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？
- ❑ (2)上表中各种寻址方式的寻址范围多大（不包括相对寻址）？相对寻址的浮动范围多大？
- ❑ (3)设开始取指令时，对应寄存器和主存相关单元的内容如下图，图中的数字均为十六进制表示，请写出指令0627H和3559H的操作数各为多少？分别单独执行这两条指令后相关寄存器的内容各是多少？

## 第二章 2.13



		地址	主 存
PC	2000H	19H	0100H
R0	0627H	27H	4000H
		400H	1000H
R5	0400H	401H	3559H
		419H	0123H
R7	3559H	41AH	0627H
		1FE7H	1234H
		1FE8H	5678H

## 第二章 2.13



□ 题解:

○ 1、该指令系统最多可有 $2^4=16$ 条指令，该计算机最多有 $2^3=8$ 个通用寄存器

○ 2、一次间接寻址范围： $2^{16}=64\text{K}$ 字  
寄存器间接寻址范围： $2^{16}=64\text{K}$ 字

变址寻址范围： $-2^{15}\sim +2^{15}=-32\text{K}\sim +32\text{K}$ 字（Rx内容为偏移量，带符号整数）

相对寻址的浮动范围= $-32\sim +31$ 字

○ 3、a、指令0627H展开：0000 011 0 00 10 0111B  
OP=0000B，M=011B=相对寻址，  
Rn=000B（无用），A=10 0111B（负数补码）  
EA=(PC)+A=2001H+FFE7H=1FE8H  
（取指后(PC)+1，且A符号扩展）  
操作数a=(EA)=(1FE8H)=5678H  
指令执行后：(PC)=2001H

## 第二章 2.13



- b、指令3559H展开：0011 010 1 01 01 1001B  
OP=0011B, M=010B=变址寻址,  
Rn=101B (R5), A=01 1001B (正数补码)  
EA=(R5)+A=0400H+0019H=0419H (A符号扩展)  
操作数b= (EA) = (0419H) =0123H  
(R5) = (R5) +1=0401H  
指令执行后：(PC) =2001H, (R5) =0401H



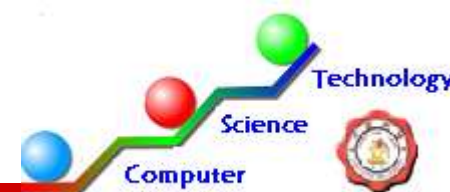
## 第二章 2.14



- 2.14 某机字长16位，主存容量为1M字，采用单字长指令格式<sup>h2</sup>，共有50条指令，采用立即寻址、直接寻址、间接等寻址方式。CPU中有PC，IR，MAR，MDR等专用寄存器，和4个通用寄存器。问：
- (1) 指令格式如何安排？
  - (2) 立即寻址的数据范围是多大？
  - (3) 为使指令能寻址到主存的任一单元，可采取什么措施？
  - (4) 能否增加其它寻址方式？



## 第二章 2.14 (解法1)



### □ 题解:

- (1) 据题意, 该机指令应能表示出50种操作码, 3种以上寻址方式, 该指令格式为单字长单地址, 如下图示:



其中, 寻址方式码M分配如下:

- M = 00, 直接寻址, EA = D
- = 01, 间接寻址, EA = (D)
- = 10, 立即寻址,
- = 11, 备用。

## 第二章 2.14 (解法1)



- (2)  $-2^7 \sim 2^7 - 1$  (有符号数) 或者  $0 \sim 2^8 - 1$  (无符号数) ;
- (3) 由于机器字长限制, 上述格式求出的有效地址EA 为8~16位长, 但题意所给主存容量为1M, 需20位地址。要将EA扩展成20位主存实际地址, 还需使用段寻址方式。为简化设计, 在此设段寻址方式为默认的, 既无需指令格式给出, 由硬件隐含完成。设硬件配置有段寄存器DS, 其长度 = 字长 = 16位, 其内容为段地址, 则:

$$\text{物理地址} = (\text{DS}) \times 2^4 + \text{EA}$$

由此式可得20位主存物理地址。

- (4) 由于剩一种寻址方式码未用, 故在寻址方式码M位数不增加的前提下, 还可增加一种寻址方式。

例如:  $M=11$ , 相对寻址,  $\text{EA} = (\text{PC}) + D$

## 第二章 2.14 (解法2)



### □ 题解:

- (1) 据题意, 该机指令应能表示出50种操作码, 3种以上寻址方式, 该指令格式为单字长单地址, 如下图示:

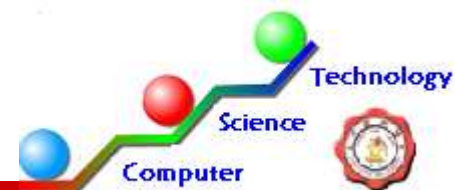
6位	3位	2位	5位
OP_Code	Mod	R	D
操作码	寻址码	通用寄存器号	形式地址

其中, 寻址方式码M分配如下:

M = 000, 直接寻址, EA = D  
= 001, 间接寻址, EA = (D)  
= 010, 立即寻址,  
= 其它, 备用。

上述格式中未给PC, IR, AR, DR四个寄存器分配地址码, 这是因为这四个寄存器是专用寄存器, 硬件会自动访问, 不需编程指定。

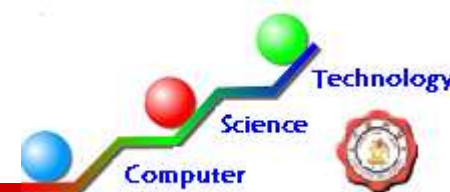
## 第二章 2.14 (解法2)



- (2)  $-2^4 \sim 2^4 - 1$  (有符号数) 或者  $0 \sim 2^5 - 1$  (无符号数) ;
- (3) 由于机器字长限制, 上述格式求出的有效地址EA 为5~16位长, 但题意所给主存容量为1M, 需20位地址。要将EA扩展成20位主存实际地址, 还需使用段寻址方式。为简化设计, 在此设段寻址方式为默认的, 既无需指令格式给出, 由硬件隐含完成。设硬件配置有段寄存器DS, 其长度 = 字长 = 16位, 其内容为段地址, 则:  
$$\text{物理地址} = (\text{DS}) \times 2^4 + \text{EA}$$

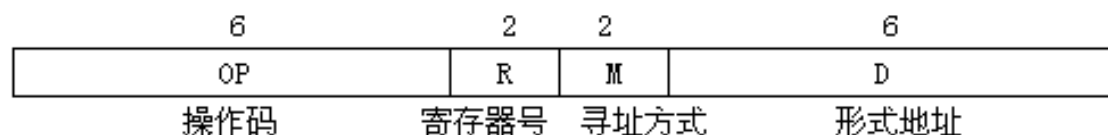
由此式可得20位主存物理地址。
- (4) 由于剩5种寻址方式码未用, 故在寻址方式码M位数不增加的前提下, 还可增加5种寻址方式。例如, 寄存器, 寄存器间接寻址和偏移寻址。

## 第二章 2.14 (解法3)



### □ 题解:

- (1) 据题意, 该机指令应能表示出50种操作码, 3种以上寻址方式, 可寻址4个通用寄存器和主存, 因此, 该指令格式可以设计为RS型单字长二地址指令。如下图示:



其中, 寻址方式码M分配如下:

M = 00, 直接寻址, EA = D  
= 01, 间接寻址, EA = (D)  
= 10, 立即寻址,  
= 11, 备用。

上述格式中未给PC, IR, AR, DR四个寄存器分配地址码, 这是因为这四个寄存器是专用寄存器, 硬件会自动访问, 不需编程指定。

## 第二章 2.14 (解法3)



- (2)  $-2^5 \sim 2^5 - 1$  (有符号数) 或者  $0 \sim 2^6 - 1$  (无符号数) ;
- (3) 由于机器字长限制, 上述格式求出的有效地址EA 为6~16位长, 但题意所给主存容量为1M, 需20位地址。要将EA扩展成20位主存实际地址, 还需使用段寻址方式。为简化设计, 在此设段寻址方式为默认的, 既无需指令格式给出, 由硬件隐含完成。设硬件配置有段寄存器DS, 其长度 = 字长 = 16位, 其内容为段地址, 则:

$$\text{物理地址} = (\text{DS}) \times 2^4 + \text{EA}$$

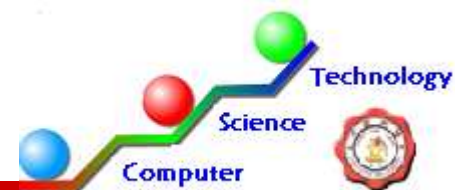
由此式可得20位主存物理地址。

- (4) 由于剩一种寻址方式码未用, 故在寻址方式码M位数不增加的前提下, 还可增加一种寻址方式。

例如:  $M=11$ , 相对寻址,  $\text{EA} = (\text{PC}) + D$



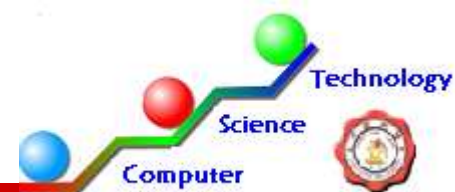
## 第二章 2.15



2.15 设某机字长32位，CPU中有16个32位的通用寄存器，主存按字编址，欲设计一种能容纳64种操作的指令系统，存储器寻址可提供8种方式，采用通用寄存器作变址寄存器，若取指令字长与机器字长相等，请安排RS型指令的格式，并回答下述问题：

- (1) 如果采用直接寻址方式，指令可寻址的最大存储空间是多少？
- (2) 如果采用一次间接寻址方式，指令可寻址的最大存储空间是多少？
- (3) 如果采用变址寻址，指令可寻址的最大存储空间又是多少？

## 第二章 2.15



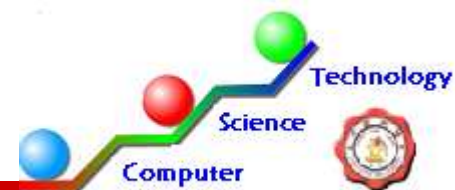
### □ 题解:

据题意，指令格式可安排如下：

6位	4位	3位	4位	15位
OP_Code	Ri	Mod	Rx	D
操作码	通用寄存器	寻址码	通用寄存器	形式地址

- (1) 直接寻址时，不需要指出变址寄存器，所以形式地址可以扩展为19位， $EA = D$ ，则指令可寻址的最大存储空间 $2^{19} = 512K$ 字。
- (2) 间接寻址时，也不需要指出变址寄存器，所以形式地址也是19位， $EA = (D)$ ，则指令可寻址的最大存储空间是 $2^{32} = 4G$ 字。  
注意： $EA$ 的位数与存储字长有关，与形式地址 $D$ 的长度无关。
- (3) 变址寻址时， $EA = (Rx) + D$ ，则该RS型指令的最大存储空间是- $2^{31} \sim 2^{31}$ 字（4G字）。注意： $EA$ 的位数仅与 $Rx$ 的位数有关，与形式地址 $D$ 的长度无关。 $Rx$ 内容为偏移量（有符号整数）

## 第二章 2.16



□ 2.16 对于一个按字节编址的存储器，存储字长32位。

请问：

- (1) 第42个字的字节地址是什么？
- (2) 单字长数据0xFF223344按照大端或小端方式存储在第42个字中，画出数据在主存中放置的示意图，并标出与每个字节数据对应的字节地址。

□ 题解：

- (1). 通常说的编号是从0开始连续编号的，所以第42个字的字节地址是 $41 * 4 = 164$ 。
- (2). 大端

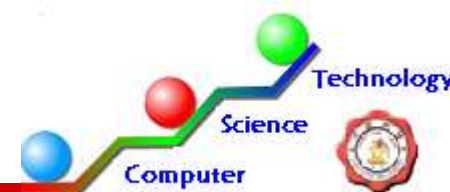
小端

FF	22	33	44
164	165	166	167

FF	22	33	44
167	166	165	164

## 第二章 2.17



2.17 在某32位计算机中，存储器按字节编址，采用小端方式存放数据。假设C语言编译器规定int型和short型长度分别为32位和16位，并且数据按边界对齐存储。某C语言程序段如下：

```
struct{  
    char x;  
    short y;  
    int z;  
} data;  
data.x='0';  
data.y=1026;  
data.z=258;
```

若程序加载时，将data分配在以0x0A000012为首地址的主存区域内，请画图示意该主存区域中存放的数据值及对应的地址编码，要求用16进制表示。

## 字符 ‘0’ 的ASCII码是：30H

## 短整型十进制数1026的十六进制值是：0402H

**长整型十进制数 258 的十六进制值是：0000 0102H**

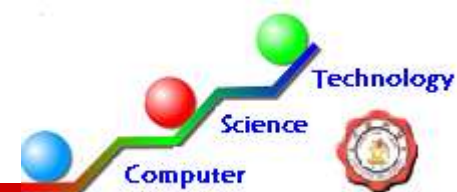
所以，主存区域中存放的数据值及对应的地址编码如下：

(用十六进制表示)

字节地址	3	2	1	0	字地址
	...				⋮
		30H			0A000010H
			04H	02H	0A000014H
	00H	00H	01H	02H	0A000018H
					0A00001CH
	...				⋮

高位 ←————— 低位

## 第二章 2.20



□ 2.20 就你对PC相对寻址的理解，解释为何汇编器在下面代码序列中直接实现分支指令时可能会有问题：

here: beq \$s0, \$s2, there

...

there: add \$s0, \$s0, \$s0

说明汇编器可能如何重写该代码序列来解决这些问题

## 第二章 2.20



### □ 题解:

在PC相对寻址中 $EA = (PC) + A$ ，实际的跳转位置为Here+There，在Here不为0的情况下，程序无法正确跳转到期望的There地址。

要实现正确跳转，需要汇编器计算Here到There之间的地址差，即代码为：

```
here: beq $s0, $s2, there-here
```

```
...
```

```
there: add $s0, $s0, $s0
```