

# 1.1 第 1 章习题解答

1.5

解：CPI1=2×0.4+3×0.2+4×0.15+5×0.25=3.25  
MIPS1= f/(CPI1×10<sup>6</sup>)=600×10<sup>6</sup>/(3.25×10<sup>6</sup>)=185  
  
CPI2=2×0.4+2×0.2+3×0.15+4×0.25=2.65  
MIPS2= f/(CPI2×10<sup>6</sup>)=800×10<sup>6</sup>/(2.65×10<sup>6</sup>)=302

1.6 解：1)优化前  $CPI = \sum_{i=1}^n (CPI_i \times \frac{IC_i}{IC}) = 1 \times 0.4 + 2 \times 0.2 + 2 \times 0.15 + 2 \times 0.25 = 1.6$

优化后：A、B、C、D 四类指令在程序中所占比例分别为 1/4、1/4、3/16、5/16，

$$CPI = \sum_{i=1}^n (CPI_i \times \frac{IC_i}{IC}) = 1 \times 1/4 + 2 \times 1/4 + 2 \times 3/16 + 2 \times 5/16 = 1.75$$

2) 优化前 MIPS = f/(CPI×10<sup>6</sup>) = (500×10<sup>6</sup>)/(1.6×10<sup>6</sup>)=312.5  
优化后 MIPS = f/(CPI×10<sup>6</sup>) = (500×10<sup>6</sup>)/(1.75×10<sup>6</sup>)=285.7

3) 优化后，程序中 A 类指令条数减少，其他指令不变，各指令 CPI 不变，所以程序执行时间变短，但程序的 CPI 却变大，MIPS 变小，所以不能简单的通过这些指标来评判计算机性能。

# 1.2 第 2 章习题解答

2.4

解：

#	真值	原码	反码	补码
1	0	0.00...0	0.00...0	0.00...0
2	-0	1.00...0	1.11...1	0.00...0
3	0.10101	0.10101	0.10101	0.10101
4	-0.10101	1.10101	1.01010	1.01011
5	0.11111	0.11111	0.11111	0.11111
6	-0.11111	1.11111	1.00000	1.00001
7	-0.10000	1.10000	1.01111	1.10000
8	0.10000	0.10000	0.10000	0.10000

## 2.5

解:

补码	真值	补码	真值
$[x]_{\text{补}} = 0.10010$	$x = 0.10010$	$[x]_{\text{补}} = 1.10010$	$x = -0.01110$
$[x]_{\text{补}} = 1.11111$	$x = -0.00001$	$[x]_{\text{补}} = 1.00000$	$x = -1.00000$
$[x]_{\text{补}} = 0.10001$	$x = 0.10001$	$[x]_{\text{补}} = 1.00001$	$x = -0.11111$

## 2.6

解:

输出结果如下:

$$x = 4294967295 = -1;$$

$$u = 2147483648 = -2147483648$$

- 1) %u 以无符号输出, %d 输出真值
- 2) 在计算机中整数以补码形式表示和存储。
- 3)  $x = -1$ , 先求  $-1$  的 32 位补码, 机器码是  $2^{32} - 1 = 4294967295$ 。所以第一行输出是分别是机器码和真值。
- 4)  $u = 2^{31}$  是一个无符号数, 无溢出, 由于首位为 1, %u 输出机器码就是 2147483648, %d 输出是真值, 将该机器码按补码转换成真值, 所以是  $-2147483648$ 。

## 2.9

解:

a) 首先分别将整数和分数部分转换成二进制数:

$$-6\frac{5}{8} = (-110.101)_2$$

移动小数点, 使其变成 1.M 的形式:  $-110.101 = -1.10101 \times 2^2$

于是得到:

$$S=1, \quad e = 2, \quad E = 2+127 = 1000\ 0001, \quad M = 101\ 0100\ 0000\ 0000\ 0000\ 0000$$

最后得到 32 位浮点数的二进制存储格式为:

$$\underline{1100\ 0000\ 1}101\ 0100\ 0000\ 0000\ 0000\ 0000 = (C0D40000)_{16}$$

2.10 求与单精度浮点数 43940000H 对应的十进制数。

$$\text{解: } 43940000H = (0\underline{100\ 0011\ 1}001\ 0100\ 0000\ 0000\ 0000\ 0000)_2$$

$$S=0, \quad E=(10000111)_2-127=8, \quad M=1.00101$$

所以表示数为  $1.00101 \times 2^8$ , 对应的十进制数为 296。

2.17

**解：**被检验位有 8 位，设检验位有  $r$  位，因为  $8+r \leq 2^r - 1$  所以  $r=4$

具体分组关系如下表：

海明码	H <sub>1</sub>	H <sub>2</sub>	H <sub>3</sub>	H <sub>4</sub>	H <sub>5</sub>	H <sub>6</sub>	H <sub>7</sub>	H <sub>8</sub>	H <sub>9</sub>	H <sub>10</sub>	H <sub>11</sub>	H <sub>12</sub>
检错码/位置	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100
映射关系	<b>P<sub>1</sub></b>	<b>P<sub>2</sub></b>	0/D <sub>1</sub>	<b>P<sub>3</sub></b>	1/D <sub>2</sub>	1/D <sub>3</sub>	0/D <sub>4</sub>	<b>P<sub>4</sub></b>	1/D <sub>5</sub>	1/D <sub>6</sub>	1/D <sub>7</sub>	0/D <sub>8</sub>
G <sub>1</sub> 校验组	√		√		√		√		√		√	
G <sub>2</sub> 校验组		√	√			√	√			√	√	
G <sub>3</sub> 校验组				√	√	√	√					√
G <sub>4</sub> 校验组								√	√	√	√	√

数据位	D <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>
P <sub>1</sub>	0	1	1	1	0	1	1	0
P <sub>2</sub>	0	1	1	1	0	1	1	0
P <sub>3</sub>	0	1	1	1	0	1	1	0
P <sub>4</sub>	0	1	1	1	0	1	1	0

$$P_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$P_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$P_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$P_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

海明码为：D<sub>8</sub>D<sub>7</sub>D<sub>6</sub>D<sub>5</sub>**P<sub>4</sub>**D<sub>4</sub>D<sub>3</sub>D<sub>2</sub>**P<sub>3</sub>**D<sub>1</sub>**P<sub>2</sub>**P<sub>1</sub>=0111**1**011**0****0****1****1**

接收方接收到的有效位只有 D<sub>1</sub> 位出错，则接收到的海明编码为 0111**1**011**0****1****1**

$$G_1 = P_1 \oplus \textcolor{red}{D}_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7 = 1$$

$$G_2 = P_2 \oplus \textcolor{red}{D}_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 = 1$$

$$G_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_8 = 0$$

$$G_4 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8 = 0$$

指错码  $G_4G_3G_2G_1 = 0011 = 3$ ，如果假设只有一位错，则是海明码 H<sub>3</sub> 出错，也就是 D<sub>1</sub> 出错，将对应位取反即可。

## 1.3 第 3 章习题解答

3.4 (3)  $[x]_{\text{补}} = 11.01001$   $[y]_{\text{补}} = 11.01000$   $[x+y]_{\text{补}} = 10.10001$ ，负溢出

3.5 (3)  $[x]_{\text{补}} = 11.00001$   $[-y]_{\text{补}} = 00.11001$   $[x-y]_{\text{补}} = 11.11010$ ，未溢出。

### 3.6

解: (1)  $x \times y = -0.1110000011$

(2)  $x \times y = 0.0100011110$

建议符号位采用双符号, 右移规则为逻辑右移, 中间运算可能产生溢出, 逻辑右移后正常。

### 3.7

解: (1)  $[x \times y]_{\text{补}} = 1.1110111110$

(2)  $[x \times y]_{\text{补}} = 0.001011110010$

建议符号位采用双符号, 右移规则为算术右移。

### 3.9

解: (1)  $[x]_{\text{补}} = \underline{000}11, \underline{00}100100 \quad [y]_{\text{补}} = \underline{000}10, \underline{11}100110$

(a) 对阶

阶差  $\Delta E = E_x - E_y = 00001$ , 阶差为1

将 $[y]_{\text{补}}$ 尾数右移一位, 同时阶码加1, 则

$[y]_{\text{补}} = \underline{000}11, \underline{111}10011$

(b) 尾数相加

$[x]_{\text{补}} + [y]_{\text{补}} = \underline{000}11, \underline{000}10111$ , 尾数为 $\underline{000}10111$

(c) 尾数规格化

由于尾数符号位跟最高有效位相同, 需要左规一位, 阶码减1

因此 $[x]_{\text{补}} + [y]_{\text{补}} = \underline{000}10, \underline{00}101110$

(d) 无需舍入, 阶码正常, 浮点运算无溢出

$[x]_{\text{补}} + [y]_{\text{补}} = \underline{000}10, \underline{00}101110$

### 3.10

解: (1)  $0.625 = (0.101)_2 = (0 \ 01111110 \ 010000000000000000000000)_2$

$-12.25 = (1100.01)_2 = (1 \ 10000010 \ 100010000000000000000000)_2$

(a) 对阶

阶差  $\Delta E = E_x - E_y = 01111110 - 10000010 = -00000100 = -4$

将0.625的尾数右移4位, 考虑隐藏位则尾数=0.000101

(b) 尾数计算

$0.000101 + (-1.100010) = -1.011101$

(c) 结果规格化

尾数符合1.X的形式, 无需规格化

(d) 舍入处理、溢出判断

尾数无需舍入，阶码正常，无溢出

$$\begin{aligned} \text{则 } 0.625 + (-12.25) &= (1\ 10000010\ 0111010000000000000000)_2 \\ &= (C13A0000)_{16} \end{aligned}$$

### 3.11

**解：**(1) 由于分配的都是 8 位寄存器，x 和 y 初始定义为无符号整数

$x=134=128+6=(1000\ 0110)_2$ ，因此 R1 中的内容为 x 的机器码=86H。

$y=246=255-9=(1111\ 0110)_2$ ，所以 y 的机器数为(1111 0110)<sub>2</sub>

$z1=x-y=1000\ 0110 - 1111\ 0110 = 1000\ 0110 + 0000\ 1010 = 1001\ 0000 = 90H$

因此 R5 中的内容为 90H。

$z2=x+y=1000\ 0110 + 1111\ 0110 = (1)\ 0111\ 1100 = 7CH$

括号中为加法器的进位，所以 R6 中的内容是 7CH

(2) m 的机器数与 x 的机器数相同，皆为 86H = (1000 0110)<sub>2</sub>

解释为有符号整数 m (用补码表示) 时，其值为  $(-111\ 1010)_2 = -122$

m-n 的机器数与 x-y 的机器数相同，皆为 90H = (1001 0000)<sub>2</sub>

解释为有符号整数 k1 (用补码表示) 时，其值为  $(-111\ 0000)_2 = -112$ 。

(3) 能。n 位加法器实现的是模  $2^n$  无符号整数加法运算。对于无符号整数 a 和 b，a+b 可以直接用加法器实现，而 a-b 可用 a 加 b 的补数实现， $a-b=a+[-b]_{\text{补}} \pmod{2^n}$ ，所以 n 位无符号整数加减运算都可在 n 位加法器中实现。有符号补码加减法运算中符号位直接参与运算，可直接复用无符号加减法电路。

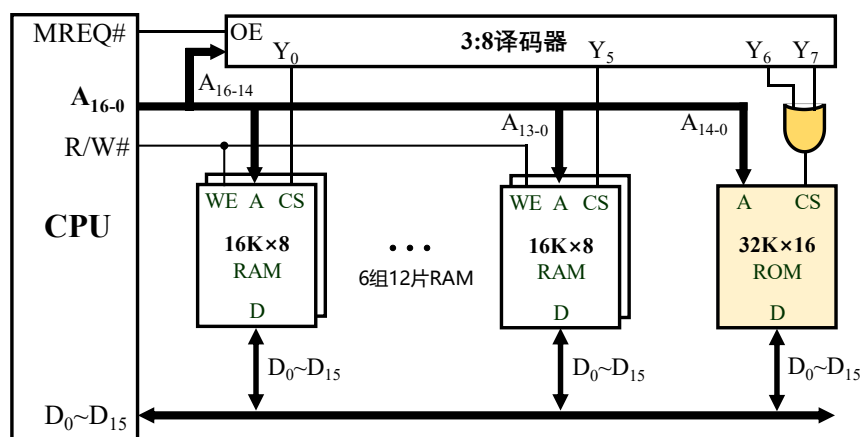
(4) 在计算机中，如果符号位进位  $C_f$  和最高数据位进位  $C_d$  不同，则结果溢出。

最后一条语句执行时会发生溢出。因为  $1000\ 0110 + 1111\ 0110 = (1)0111\ 1100$ ，括号中为符号位进位  $C_f=1$ ，最高数据位进位  $C_d=0$ ，根据上述溢出判断规则可知结果溢出。

## 1.4 第 4 章习题解答

### 4.7

**解：**



4. 8

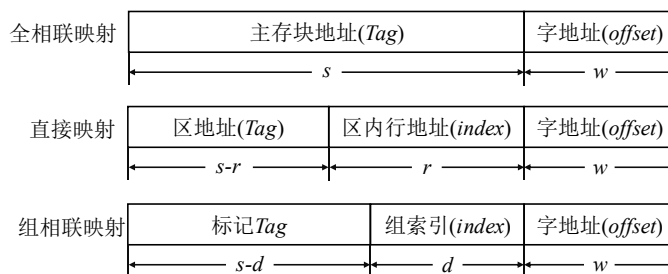
**解:** 64K×1 位的 DRAM 芯片的排列方式为 256 行×256 列, 已知每行刷新闻隔不超过 2ms, 可知最大刷新周期为 2ms, 如采用异步刷新, 需要将 2ms 分成 256 个时间段,  $2\text{ms}/256=7.8125\mu\text{s}$ , 每个时间段最后 0.5μs 用于刷新 DRAM 的一行, 因此产生刷新信号的时间间隔为 7.8125μs.

若采用集中刷新, 存储器刷新一遍至少需要 256 个读写周期, CPU 的死时间是  $256\times0.5\mu\text{s}=128\mu\text{s}$ .

4. 11

**解:**

不同映射方式下主存地址划分如下图, 这里  $s=22$ ,  $w=2$ ,  $r=14$ ,  $d=12$



(1)全相联映射: 主存数据块可映射到 cache 的任意行, 表中共有 5 个地址, 数据从主存映射到 cache 后占用 5 行, 假设就是 cache 的前 5 行, 则对应标志位就是主存块地址字段, 将主存地址逻辑右移两位就是标记字段, 具体分布如下表。

cache 行	标志	数据
0	000000	87568536
1	000002	87792301
2	004001	9ABEFCDO
3	007FFF	4FFFFC68
4	3FFFFE	01BF2460

(2) 直接相联映射

主存数据块只能映射到 cache 的特定行，对应行地址为 index 字段的值，index 字段长度为  $s-r=22-14=8$  位，也就是主存地址高 8 位，具体分布如下表所示。

cache 行	标志	数据
0000	00	87568536
0002	00	87792301
0001	01	9ABEFC0D
3FFF	01	4FFFFC68
3FFE	FF	01BF2460

(3) 组相联映射：

该方式下，主存的一个数据块只能射到 cache 的特定组中的任意行，则 cache 共分为 4K 组，组索引字段 12 位，标志字段为高 10 位，具体分布如下表。

cache 组	标志	数据
000	000	87568536
002	000	87792301
001	004	9ABEFC0D
0FFF	007	4FFFFC68
0FFE	3FF	01BF2460

4. 12

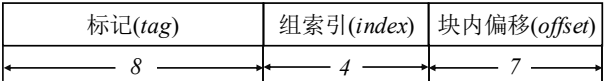
解：(1) 主存容量为： $4096 \times 128 = 2^{19}W$ ，故主存地址位数为 19 位

cache 容量为： $64 \times 128 = 2^{13}W$ ，故 cache 地址位数为 13 位

(2) 每个组中包含 4 个存储块，共包含  $64/4=16=2^4$  组，故组索引字段 index 位数为 4 位

由于每块由 128 个字组成，访问地址为字地址，故块内偏移地址位数为 7 位

故标记部分(Tag)的位数为： $19-4-7=8$  位，则主存地址划分情况如下：



4. 16

解：(1) 装入位为 0 的虚页为失效页，分别是 2、3、5、7

(2) 虚存空间大小为  $8 \times 1KB = 8KB$ ，故虚地址为 13 位，其中虚页号为 3 位

主存容量 4KB，则物理页号 2 位

主存地址 <sub>10</sub>	主存地址 <sub>2</sub>	VPN	PPN	实存地址 <sub>2</sub>	实存地址 <sub>16</sub>
0	<u>000</u> 00 0000 0000	0	3	<u>11</u> 00 0000 0000	3072
3028	<u>010</u> 11 1101 0100	2	缺页		
1023	<u>000</u> 11 1111 1111	0	3	<u>11</u> 11 1111 1111	4095
2048	<u>010</u> 00 0000 0000	2	缺页		
4096	<u>100</u> 00 0000 0000	4	3	<u>11</u> 00 0000 0000	3072
8000	<u>111</u> 11 0100 0000	7	缺页		

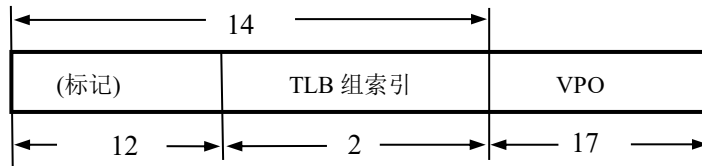
4.17

解：(1) 虚拟存储的页面数为： $2\text{GB}/128\text{KB}=2^{14}$ ，故虚页号的位数为 14 位

由于页大小为  $128\text{KB}=2^{17}\text{B}$ ，故页内偏移地址的位数为 17 位

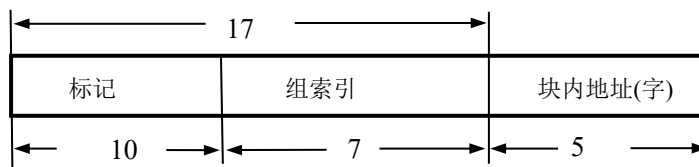
所以虚拟地址的高 14 位表示的是虚页号，低 17 位表示的是页内偏移地址

TLB 采用 4 路组相联方式，16 个页表项，每组 4 路，共 4 组，所以组索引字段 2 位，故虚页号的高 12 位为 TLB 标记，低 2 位为 TLB 组索引。



(2) 物理内存  $4\text{MB}=2^{22}\text{B}$ ，地址线 22 根，页内偏移地址位数为 17 位，高 5 位为物理页号，低 17 位表示页内偏移地址。

(3) cache 块大小为  $32\text{B}$ ，故块内字节偏移地址为 5 位，采用 4 路组相联，cache 共分成  $16\text{KB}/(8 \times 4 \times 4\text{B}) = 128 = 2^7$  组，故组索引字段需要 7 位，剩下的  $22 - 5 - 7 = 10$  位为标记，则主存地址划分如下：



4.18

解：(1) 页大小为  $8\text{KB}=2^{13}\text{B}$ ，则页内偏移地址 13 位， $D=13$ ；

A 字段为 VPN，会作为 TLB 标记存放在 B 中

$A=B=32-D=32-13=19$ ， $C=24-D=24-13=11$ 。

主存块大小为  $64\text{B}=2^6\text{B}$ ，故  $G=6$

cache 采用 2 路组相联，每组容量  $64\text{B} \times 2 = 128\text{B}$ ，共有  $64\text{KB}/128\text{B} = 512 = 2^9$  组，故组索引字段  $F=9$ ， $E=24-9-6=9$

因此 A~G 的位数分别为 19、19、11、13、9、9、6

TLB 中标记字段 B 的内容是虚页号，表示该 TLB 项对应哪个虚页的页表项。

(2) 块号  $4099 = \underline{00\ 0001\ 0000\ 0000\ 0011}\text{B}$ ，因此，所映射的 cache 组号为  $0\ 0000\ 0011\text{B} = 3$ ，对应的 H 字段内容为 **0 0000 1000B**。

(3) 因为缺页处理需要访问磁盘，而 cache 缺失只要访问主存，cache 缺失带来的开销小，而处理缺页的开销大。

(4) 因为采用写穿策略时需要同时写快速存储器和慢速存储器，而写磁盘比写主存慢得多，所以 cache 可以采用写穿策略，而虚存则应采用写回策略。



#### 4.19

解:

(1) 物理地址由实页号和页内地址拼接, 因此其位数为  $16+12=28$ ; 或从 cache 机制直接得  $20+3+5=28$ 。

(2) TLB 表标记字段 tag 为 20 位, 对应 VPN 字段 20 位, 所以采用全相联映射。TLB 应采用 SRAM, 读写速度快, 但成本高, 多用于容量较小的高速缓冲存储器。

(3) cache 中每组 2 行, 2 个比较器, 故采用 2 路组相联映射。2 路组相联采用 LRU 替换算法, 实现时只需要在 cache 行中增加 1 位 LRU 位用于替换计数; 采用回写策略, 所以每行还需要增加 1 位脏位, 如果脏位为 1 则需要写回内存。

28 位物理地址中 Tag 字段, 组索引字段, 块内偏移地址分别是 20、3、5 位, 故 cache 共有  $2^3=8$  组, 每组 2 行, 每行  $2^5 B=32B$ ; 故 Cache 总容量为  $8 \times 2 \times (20+1+1+1+32 \times 8)=4464$  位 = 558 字节。cache 中有效位用来指出所在 cache 行中的信息是否有效。

4) 虚拟地址分为两部分: 虚页号、页内地址; 物理地址分为两部分: 实页号、页内地址。利用虚拟地址的虚页号部分去查找 TLB 表 (缺失时从页表调入), 将实页号取出后和虚拟地址的页内地址拼接, 就形成了物理地址。

虚拟地址 0008C040H 的 VPO=040H, 虚页号 VPN=008CH, 与 TLB 表中第一行相符, 对应实页号 PPN=0040H 且有效位为 1, 将 PPN 与 VPO 组合, 得到物理地址是 0040040H。

物理地址 0040040H 分解成标志 20, 组索引, 块内偏移形式为 (00400H, 2, 0), 因此将 00400H 与 cache 中的第 2 组两行中的标志字段并发比较, 虽然有一个 cache 行中的标志字段相符, 但有效位为 0, 而另一 cache 行的标志字段与 00400H 不相等, 故 cache 不命中。

虚拟地址为 0007 C260H 的低 12 位为页偏移字段, 和物理地址相同, 具体为 0010 0110 0000<sub>2</sub>。根据物理地址的结构, 011 是组号, 因此该地址应该映射到 cache 第 3 组。

## 1.5 第 5 章习题解答

#### 5.6

解: 采用扩展操作码方式, 双操作数指令操作码字段为  $16-2 \times 6=4$  位。

则计算机最多有  $2^4=16$  条指令双操作数指令, 实际双操作数指令为 m 条, 因此还有  $(16-m)$  种状态用于扩展单操作数指令和无操作数指令, 最多表示  $(16-m) \times 2^6$  条单操作数指令, 假设单操作数指令条数为 y, 则  $((16-m) \times 2^6 - y) \times 2^6 = n$

因此单操作数指令条数  $y = (16-m) \times 64 - n/64$  条。

(2) 双操作数指令取最大数:  $2^4 - 1 = 15$  条 (只预留一个状态进行扩展)

单操作数指令取最大数:  $2^6 - 1 = 63$  条 (只预留一个状态进行扩展)

无操作数指令取最大数： $2^6=64$  条

5.7

**解：**根据相对寻址有效地址的计算公式  $EA=(PC)+D$  可知  $D=EA-(PC)$

对于相对寻址而言，关键是求出计算有效地址时 PC 的当前值。

(1) PC 值为 256, 由于指令字长为 3 个字节, 因此 PC 在取指令完成后修改为  $256+3=259$  , 所以  $D=290-259=(31)_{10}=001FH$ , 因此转移指令第 2 字节为 1FH, 第 3 字节为 00H。

(2) PC 值为 128, 则取指令完成后  $PC=128+3=131$ , 因此  $D=110-131=(-21)_{10}=FFEDH$ (补码), 因此转移指令第 2 字节为 EBH, 第 3 字节为 FFH。

5.8

**解：**

(1)  $4420H=010001\ 00\ 0010\ 0000_2$  直接寻址,  $EA=D=20H$

(2)  $2244H=001000\ 10\ 0100\ 0100_2$  X2 变址寻址,  $EA=(X2)+D=1122+44=1166H$

(3)  $1322H=000100\ 11\ 0010\ 0010_2$  相对寻址,  $EA=(PC)+2+D=1234+2+22=1258H$

(4)  $3521H=001101\ 01\ 0010\ 0001_2$  X1 变址寻址,  $EA=(X1)+D=0037+21=58H$

5.12

**解：**

(1) 操作码占 4 位, 则该指令系统最多可有  $2^4=16$  条指令; 操作数占 6 位, 寻址方式占 3 位, 于是寄存器编号占 3 位, 则该机最多有  $2^3=8$  个通用寄存器;

(2) 主存容量 128KB, 按字编址, 计算机字长为 16 位, 划分为  $128KB/2B=2^{16}$  个存储字, 故 MDR 和 MAR 至少各需 16 位。

(3) PC 和  $R_n$  可表示的地址范围均为  $0\sim 2^{16}-1$ , 而主存地址空间为  $2^{16}$ , 故转移指令的目标地址范围是  $0000H\sim FFFFH$  ( $0\sim 2^{16}-1$ )。

(4) 汇编语句 “add (R4), (R5) +”, 对应的机器码为  $0010\ 0011\ 0001\ 0101B=2315H$ 。该指令执行后, 寄存器 R5 和存储单元 5678H 的内容会改变。执行后 R5 的内容从 5678H 变成 5679H。存储单元 5678H 中的内容变成该加法指令计算的结果  $5678H+1234H=68ACH$ 。

## 1.6 第 6 章习题解答

6.24

**解：**(1) 通用寄存器( $R0\sim R3$ )和 PC 程序员可见。因为采用了单总线结构, 若无暂存器 T, 则 ALU 的 A、B 端口会同时获得两个相同的数据, 数据通路不能正常工作。增加暂存器后可以使 A、B 端口输入不同的数据。

(2)ALU 共有 7 种操作，故操作控制信号 ALUop 至少需要 3 位；移位寄存器有 3 种操作，其操作控制信号 SROp 至少需要 2 位。

(3)信号 SR<sub>out</sub>所控制的部件是三态门，用于控制移位器与总线之间数据通路的连接与断开。

(4)端口①、②、③、⑤、⑧都是控制信号，需要连接到控制部件输出端，其他都是数据通路。

(5)首先内总线数据需要送 MUX 的 1 号引脚，因此⑥连接⑨；另外多路选择器输出应该连接到运算器 B 端，所以④连接⑦。

(6)因为每条指令的字节长度是 2，顺序寻址时应该实现 (PC)+2 的逻辑，所以这里 MUX 的一个输入端为 2，便于执行(PC)+2 操作。

6.25

解：

(1)寻址方式 1 位，寄存器编号 2 位，操作码位数=16-3×(1+2)=7 位，最多可定义  $2^7=128$  条指令。

(2)各条指令的机器代码分别如下：

① “inc R1” 的机器码为：0000001 0 01 0 00 0 00，即 0240H。

② “shl R2, R1” 的机器码为：0000010 0 10 0 01 0 00，即 0488H。

③ “sub R3, (R1), R2” 的机器码为：0000011 0 11 1 01 0 10，即 06EAH。

(3)各标号处的控制信号或控制信号取值如下：

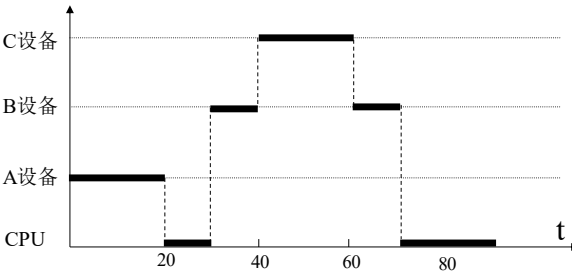
①0；②mov；③mova；④left；⑤read；⑥sub；⑦mov；⑧SR<sub>out</sub>。

(4)指令 “sub R1, R3, (R2)” 的执行阶段至少包含 4 个时钟周期；

指令 “inc R1” 的执行阶段至少包含 2 个时钟周期。

# 1.7 第 9 章习题解答

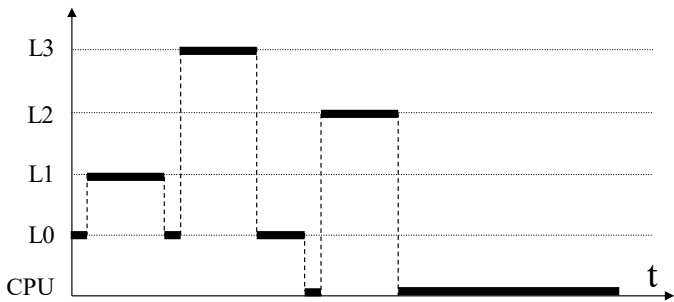
9.4 解：A 中断在 0μs 到达，首先执行 A 中断，20μs 后返回，30μs 时 B 中断到达开始执行 B 中断，40μs 时 C 到达，由于 B 的屏蔽码可知没有对 C 中断屏蔽，故 40μs 后去执行 C 中断，60μs 后 C 返回 B 中断，70μs 时返回主程序。CPU 运行轨迹如下：



9.5 解：（1）中断屏蔽字如下表所示：

设备名	中断			
	中断屏蔽字			
	L0	L1	L2	L3
L0	1	0	1	0
L1	1	1	1	1
L2	0	0	1	0
L3	1	0	1	1

（2）CPU 运行轨迹如图所示：



9.6

解：外设的最大数据传输率为 20KBps，缓冲区为 2Byte，每次中断传输 2Byte。

因此每秒钟产生的中断数为  $20\text{KB}/2\text{B}=10000$  次。

每次的执行为 500 个周期。则中断占 CPU 时间的比率为：

$500 \times 10000 / (500 \times 10^6) = 1\%$ ，对 CPU 的影响不大， 可以采用中断方式。

若最大数传率为 2MBps，则每秒钟产生的中断数为  $2\text{MB}/2\text{B}=10^6$

CPU 占用率为  $(500 \times 10000) / (500 \times 10^6) = 100\%$ ，故不能采用中断方式。

9.7 解：

(1)每传送一个 ASCII 字符，需要传输的位数有 1 位起始位、7 位 ASCII 数据位、1 位奇校验位和 1 位停止位，故总位数为  $1+7+1+1=10$ 。

I/O 端口每秒钟最多可接收  $1/0.5 \times 10^{-3} = 2000$  个字符。

(2)一个字符传送时间包括：设备 D 将字符送 I/O 端口的时间、中断响应时间和中断服务程序前 15 条指令的执行时间。时钟周期  $T=1/(50\text{ MHz})=20\text{ ns}$ ，设备 D 将字符送 I/O 端口的时间为  $0.5\text{ ms}/20\text{ ns}=2.5 \times 10^4$  个时钟周期。一个字符的传送时间大约为  $2.5 \times 10^4 + 10 + 15 \times 4 = 25070T$ 。完成 1000 个字符传送所需时间约为  $1000 \times 25070 = 2.507 \times 10^7 T$ 。

CPU 用于该任务的时间大约为  $1000 \times (10 + 20 \times 4) = 9 \times 10^4 T$ 。

在中断响应阶段，CPU 主要进行关中断、保护断点、中断识别操作。

9.8 解：

(1) 设备 A 每隔  $4B/2MB=2\mu s$  就会产生新数据, 为保证数据不丢失, 每隔  $2\mu s$  必须查询一次才能保证数据, 每秒的查询次数至少是  $1s/2\mu s=5\times 10^5$ , 每秒 CPU 用于设备 A 输入/输出的时间至少为  $5\times 10^5\times 10\times 4=2\times 10^7$  个时钟周期, 占整个 CPU 时间的百分比至少是  $2\times 10^7/500M=4\%$ 。

(2) 中断响应和中断处理的时间为  $400\times (1/500M)=0.8\mu s$ , 而 B 设备每隔  $4B/40MB=0.1\mu s<0.8\mu s$ , 设备 B 不适合采用中断 I/O 方式。

(3) 在 DMA 方式中, 只有预处理和后处理需要 CPU 处理, 设备 B 每秒 DMA 传输次数为  $40MB/1000B=40000$ , CPU 用于设备 B 输入输出时间为  $40000\times 500=2\times 10^7$  个时钟周期, 占 CPU 总时间的百分比最多为  $2\times 10^7/500M=4\%$ 。