第二章课后习题

- **第七题解：**

  由题意知，首先机器是32位机器，所以
  $R_1 = 0000\ 108BH = 0000\ 0000\ 0000\ 0000\ 0001\ 0000\ 1000\ 1011b$，
  $R_2 = 8080\ 108BH = 1000\ 0000\ 1000\ 0000\ 0001\ 0000\ 1000\ 1011b$

  1. 对于**无符号整数**加法而言：R_1和R_2的真值就为$0000\ 108BH$和$8080\ 108BH$。
  2. 对于**带符号整数**乘法而言：由R_1和R_2的32位二进制表示可知，R_1的符号为正，R_2的符号为负。因为正数的原码和补码均一致，所以R_1**的真值为**$+0000\ 108BH$。对负数的补码先减一，然后再按位取反即可得到其原码，所以R_2**的真值为**
     $-0111\ 1111\ 0111\ 1111\ 1110\ 1111\ 0111\ 0101b = -7F7F\ EF75H$。
  3. 对于**单精度浮点数**减法而言：R_1**的真值为**
     $(+1.000\ 0000\ 0001\ 0000\ 1000\ 1011)_2 \times 2^{-127}$；$R_2$的真值为
     $(-1.000\ 0000\ 0001\ 0000\ 1000\ 1011)_2 \times 2^{-126}$。

- **第八题解：**

关系表达式	运算类型	结果	说明
$0 == 0U$	无符号整数	1	$00\cdots0B(0) = 00\cdots0B(0)$
$-1 < 0$	带符号整数	1	$11\cdots1B(-1) < 00\cdots0B(0)$
$-1 < 0U$	无符号整数	0	$11\cdots1B(2^{32}-1) > 00\cdots0B(0)$
$2147483647 > -2147483647 - 1$	带符号整数	1	$011\cdots1B(2^{31}-1) > 100\cdots0B(-2^{31})$
$2147483647U > -2147483647 - 1$	无符号整数	0	$011\cdots1B(2^{31}-1) < 100\cdots0B(2^{31})$
$2147483647 > (int)2147483648U$	带符号整数	1	$011\cdots1B(2^{31}-1) > 100\cdots0B(-2^{31})$
$-1 > -2$	带符号整数	1	$11\cdots1B(-1) > 11\cdots0B(-2)$
$(unsigned) - 1 > -2$	无符号整数	1	$11\cdots1B(2^{32}-1) > 11\cdots0B(2^{32}-2)$

- **第十题解：**

  1. 已知x的机器数为$FFFF\ 0006H = 1111\ 1111\ 1111\ 1111\ 0000\ 0000\ 0000\ 0110$，同时$x$的数据类型为**32位有符号整数**，所以真值为$x = -6$
  2. 已知y的机器数为$DFFCH$，同时y的数据类型为**16位有符号整数**，所以$y = -12$
  3. 已知z的机器数为$FFFF\ FFFAH$，同时z的数据类型为**32位无符号整数**，所以
     $z = 4294967290$
  4. 已知c的机器数为$2AH$，同时c的数据类型为**8位有符号整数**，所以$c = 42$
  5. 已知a的机器数为$C448\ 0000H$，同时a的数据类型为**单精度浮点数**，所以$a = -36.0$

6. 已知b的机器数为$C024\ 8000\ 0000\ 0000H$，同时b的数据类型为**双精度浮点数**，所以$b=-36.5$

- **第十八题解：**

  已知$x=-0.125, y=7.5, i=100$，所以三者的**二进制表示为**$x=(-0.001)_2=-1.0\times 2^{-3}, y=(111.1)_2, i=(1100100)_2$。从而可知，三者的**机器数表示为**$x=(BE00\ 0000)_{16}, y=(40F0\ 0000)_{16}, i=(0064)_{16}$。

  在大端方式和小端方式机器上的存放位置如下表格**（采用十六进制表示）**：

地址	大端机内容	小端机内容
100	BE	00
101	00	00
102	00	00
103	00	BE
108	40	00
109	F0	00
110	00	F0
111	00	40
112	00	64
113	64	00

第三章课后习题

- **第三题解：**

> **Note**
>
> 表格中机器数采用**十六进制表示**

w**的机器数**	w**的值**	$func1(w)$**的机器数**	$func1(w)$**的值**	$func2(w)$**的机器数**	$func2(w)$**的值**
$0000\ 007F$	127	$0000\ 007F$	127	$0000\ 007F$	127
$0000\ 0080$	128	$0000\ 0080$	128	$FFFF\ FF80$	-128
$0000\ 00FF$	255	$0000\ 00FF$	255	$FFFF\ FFFF$	-1
$0000\ 0100$	256	$0000\ 0000$	0	$0000\ 0000$	0

- **第九题解：**

表示														
无符号	0xB0	176	0x8C	140	0x6C	60	无意义	无意义	1	0x24	36	无意义	无意义	0

表示	X	x	Y	y	$X+Y$	$x+y$	OF	SF	CF	$X-Y$	$x-y$	OF	SF	CF
带符号	0xB0	-80	0x8C	-116	0x6C	60	1	0	无意义	0x24	36	0	0	无意义

- **第十题解：**

  已知$x=10,y=-6$，则可知$x=(001010)_2,y=(-000110)_2$，所以$x$和$y$的补码表示为$x=001010,y=111010$

  1. $[x+y]_{补}=[x]_{补}+[y]_{补}=000100=4$，$[x-y]_{补}=[x]_{补}+[-y]_{补}=010000=16$
  2. 首先符号位异或$1\oplus 0=1$，我们可以确定最后结果应该为**负数**，运算过程如下表格。其中A为部分积，B为乘数，C为被乘数。

A	C	对应的操作
000000	000110	$C_0=0$，将A和C联合右移一位，得000000 000011
000000	000011	$C_0=1$，$A=A+B=001010$，将A和C联合右移一位，得000101 000001
000101	000001	$C_0=1$，$A=A+B=001111$，将A和C联合右移一位，得000111 100000
000111	100000	$C_0=0$，将A和C联合右移一位，得000011 110000
000011	110000	$C_0=0$，将A和C联合右移一位，得000001 111000
000001	111000	$C_0=0$，将A和C联合右移一位，得000000 111100，即为最终结果

     综上所述，$[x\times y]_{原}=-111100=-60$
  3. 运算过程如下表格，其中A为累加寄存器，B为乘数，C为被乘数。

A	C	C_{-1}	操作
000000	111010	0	$(C_0,C_{-1})=(0,0)$，将$A\|\|C$算术右移一位，得000000 011101，$C_{-1}=0$
000000	011101	0	$(C_0,C_{-1})=(1,0)$，$A=A-B=110110$，将$A\|\|C$算术右移一位，得111011 001110，$C_{-1}=1$
111011	001110	1	$(C_0,C_{-1})=(0,1)$，$A=A+B=000101$，将$A\|\|C$算术右移一位，得000010 100111，$C_{-1}=0$
000010	100111	0	$(C_0,C_{-1})=(1,0)$，$A=A-B=111000$，将$A\|\|C$算术右移一位，得111100 010011，$C_{-1}=1$
111100	010011	1	$(C_0,C_{-1})=(1,1)$，将$A\|\|C$算术右移一位，得111110 001001，$C_{-1}=1$
111110	001001	1	$(C_0,C_{-1})=(1,1)$，将$A\|\|C$算术右移一位，得111111 000100，$C_{-1}=1$

     综上所述，$[x\times y]_{补}=000100=-60$

4. 计算过程如下表格：

步骤	处理的被除数位（下次加入的 bit）	移位前的 R（十进制）	左移并加入该位后的 R（十进制）	R 的二进制（6 位补码）	根据上一步符号进行运算	运算结果 $R = R \pm D$（十进制）	运算后 R 的二进制	生成商位 q_i
初始	—	0	—	`000000`	—	—	`000000`	—
1	1（最高位）	$R = 0$	$(0 \ll 1) + 1 = 1$	`000001`	上一步 $R \geq 0 \Rightarrow R - D$	$1 - 6 = -5$	`111011`	$q_1 = 0$
2	0	-5	$(-5 \ll 1) + 0 = -10$	`110110`	上一步 $R < 0 \Rightarrow R + D$	$-10 + 6 = -4$	`111100`	$q_2 = 0$
3	1	-4	$(-4 \ll 1) + 1 = -7$	`111001`	上一步 $R < 0 \Rightarrow R + D$	$-7 + 6 = -1$	`111111`	$q_3 = 0$
4	0（最低位）	-1	$(-1 \ll 1) + 0 = -2$	`111110`	上一步 $R < 0 \Rightarrow R + D$	$-2 + 6 = 4$	`000100`	$q_4 = 1$
结束调整	—	—	—	—	最终 $R \geq 0$，无需恢复	最终余数 $= 4$	`000100`	商二进制 $Q = 0001$

综上所述，商为-1，余数为4.

- **第十七题解：**

  采用**减法与移位**结合可以使计算所耗费得时钟周期最少。首先$55 = 64 - 8 - 1 = 2^6 - 2^3 - 2^0$，所以我们采用以下算法。

  $$55 \times x = (x << 6) - (x << 3) - x$$

  易得，该种算法只需**4个时钟周期。**

- **第二十题解：**

  1. **结论：永真**
     证明如下：
     IEEE 754 浮点数的平方运算不会引入负号，且 `+∞·+∞=+∞`，`+0·+0=+0`，`NaN` 输入在本题不可能出现（`dx` 由 `int` 转换而来）。
     因此 `dx*dx` 必为 `+0` 或正数，恒 `>= 0`。

  2. **结论：不永真**
     反例如下：
     取 `x = 0x1fffffff`（即 `2^29 - 1`）。

     - `dx = (double)x` 精确等于该整数。
     - 先转 `float` 时需舍入到 24 位有效位，结果变为 `2^29`，再转回 `double` 后得到 `2^29`，与 `dx` 不等。

  3. **结论：不永真**
     反例如下：
     令

     - `x = 0x7fffffff`（`INT_MAX`）
     - `y = 1`

则 `x + y` 发生**有符号整数溢出**，按 C 规则为未定义行为；在常见 2 补码实现下实际结果为 `INT_MIN`（`-2^31`）。
而

- `dx = 2^31 - 1`
- `dy = 1`
- `dx + dy = 2^31`（精确可表示为 `double`）

显然 `2^31 ≠ -2^31`，等式不成立。

4. **结论：永真**
   证明如下：
   `double` 的精度（53 位）足以完整容纳任意 `int`（31 位有效值）的精确值，故 `dx, dy, dz` 均为**精确整数。**
   整数加法在 `double` 中**无舍入误差**，且整数加法本身满足结合律，因此浮点运算亦精确相等。
5. **结论：永真**
   证明如下：
   同上，`dx, dy, dz` 都是精确整数，且乘积结果远小于 `2^53`，故三次乘法均**无舍入误差。**
   整数乘法满足交换律，因而等式恒成立。
6. **结论：不永真**
   反例如下：
   取
   - `x = 0`
   - `y = 1`

   则
   - `dx / dx = 0.0 / 0.0 → NaN`
   - `dy / dy = 1.0 / 1.0 → 1.0`

   `NaN ≠ 1.0`，等式不成立。

- **第二十四题解：**

题号	无附加位结果（编码）	有 2 位附加位结果（编码）	真值
(1)	0 1000 001000	0 1000 001001	128 / 144
(2)	0 1111 001111	0 1111 001111	120
(3)	0 1111 000110	0 1111 000110	48
(4)	0 1111 000010	0 1111 000010	16

- **第二十五题解：**

1. 计算如下：

$$0.75 + (-65.25) = -64.5 = (-1.0000001)_2 \times 2^6 = 1\ 10000101\ 00000010000000000000000$$

2. 计算如下：

$$0.75 - (-65.25) = 66 = (1.000010)_2 \times 2^6 = 0\ 10000101\ 00001000000000000000000$$