

8.2

- 1). 单元 0 包含 $0000\ 0000_2$, 单元 4 包含 $0100\ 0010_2$.
- 2).
 - ①. 单元 0: 0, 单元 1: -2
 - ②. 单元 2: 128, 单元 3: 127
 - ③. 单元 4: B
 - ④. $0\ 1000\ 0101\ 101\ 0101\ 1000\ 0000\ 0000\ 0000_{IEEE\ 754} = 106.75$
- 3). 将 R3 和 R4 里的内容相加, 结果存回 R5 里.
- 4). 单元 11 指向单元 1, 其中的数为 $1111\ 1110_2$.

8.3 可知补码整数有 $16 - \lceil \log_2 12 \rceil - 2 \times \lceil \log_2 8 \rceil = 6$ 位, 范围为 -2^5 到 $2^5 - 1$, 即 -32 到 31.

8.4 可知无符号整数有 $32 - \lceil \log_2 200 \rceil - 3 \times \lceil \log_2 60 \rceil = 6$ 位, 最大数为 $2^6 - 1$, 即 63.

9.1

- 1). 立即数有 16 位, 范围为 -2^{15} 到 $2^{15} - 1$, 即 -32768 到 32767.
 - 2). 范围为 0 到 $2^{16} - 1$, 即 0 到 65535.
 - 3). 所能加载的最大地址为 $0x4000\ 0000 + 0xFFFFC + 0x3 = 0x4000\ FFFF$.
 - 4). 表示寄存器所需的位数减 1, 故立即数位数增加 2, 可表示的最大值为 $2^{17} - 1$, 即 133071.
- 9.2 R - 类型指令有 5 个未用位, 增加寄存器需要多用 $3 \times (\log_2 128 - \log_2 32) = 6$ 位, 故不可以.

9.3

- 1). 需要 $\log_2 65536 = 16$ 位表示地址.
- 2). 偏移值为 $20 - (10 + 1) = 9$.

9.4 $001101\ 01000\ 01001\ 0000\ 0000\ 0000\ 0011$ (SLLI R9, R8, #3)

9.5

- 1). $000001\ 00001\ 00010\ 0000\ 0000\ 0000\ 0000$ (ADDI R2, R1, #0)
- 2). $001011\ 00001\ 00001\ 1111\ 1111\ 1111\ 1111$ (XORI R1, R1, #0xFFFF)
- 3). $001001\ 00001\ 00001\ 0000\ 0000\ 0000\ 0011$ (ANDI R1, R1, #3)

9.9

$R_1 = 0x4000\ 0000$ $R_2 = 0x4321\ 0000$ $R_3 = 0x0000\ 0043$
 $R_4 = 0x0000\ 8000$ $R_5 = 0x0000\ 0021$ $R_6 = 0x0000\ 4000$

9.10 前三次计算后, $R_3 = \overline{\overline{R_1} + R_2}$, 跳转说明 $R_3 = 0$, 即 $\overline{\overline{R_1} + R_2} = -1$, 所以 $R_1 = R_2$.

9.11 此程序等价于

```

1  R1 = 0;
2  R3 = 16;
3  R4 = 1;
4  do {
5      R5 = R2 & R4;
6      if (R5 != 0)
7          R1 = R1 + 1;
8      R4 = R4 << 1;
9      R3 = R3 - 1;
10 } while (R3 != 0);

```

即计算 R_2 的低 16 位中 1 的个数, 因此可知 R_2 的低 16 位中有 7 个 1.