2.85/

Bias=2^(k-1)-1; V=2^E\*M

A.

E=2, M=0b1.11, f=0b0.11, V=7.0

Bits: 0 10…01 110…

B.

E=n, M=0b1.111…, f=0b0.111… (n bits 1), V=0b11111… (n+1 bits 1) =2^(n+1)-1

Bits: 0 bias+n 111…

C.

E=bias-1, M=0b1.000…, f=0b0.000…, V=2^(bias-1)

Bits: 0 11…101 000…

2.87/

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 描述 | Hex | M | E | V | D |
| -0 | 0x8000 | 0 | -14 | -0 | -0.0 |
| 最小的>2的值 | 0x4001 | 1025/1024 | 1 | 1025/512 | 2.00195312 |
| 512 | 0x6000 | 1 | 9 | 512 | 512.0 |
| 最大的非规格化数 | 0x03FF | 1023/1024 | -14 | 1023/(2^24) | 6.09755516e-5 |
| -∞ | 0XFC00 | — | — | -∞ | -∞ |
| 十六进制表示为3BB0的数 | 0x3BB0 | 123/64 | -1 | 123/128 | 0.9609375 |

2.88/

|  |  |  |  |
| --- | --- | --- | --- |
| Format A | | Format B | |
| Bits | Value | Bits | Value |
| 1 01111 001 | -9/8 | 1 0111 0010 | -9/8 |
| 0 10110 011 | 11\*(2^4) | 0 1110 0110 | 11\*(2^4) |
| 1 00111 010 | -5/(2^10） | 1 0000 0101 | -5/(2^10) |
| 0 00000 111 | 7/(2^17) | 1 0000 0000 | -0 |
| 1 11100 000 | -1\*(2^13) | 1 1111 0000 | -1\*(2^8) |
| 0 10111 100 | 3\*(2^7) | 0 1111 1000 | 3\*(2^7) |

2.95/

/\* Compute 0.5\*f. If f is NaN, then return f. \*/

float\_bits float\_half(float\_bits f)

{

unsigned sign = f >> 31;

unsigned noSign = f & 0x7FFFFFFF;

unsigned exp = f >> 23 & 0xFF;

unsigned frac = f & 0x7FFFFF;

if (exp == 0xFF)

{

return f;

}

int add = ((frac & 0x3) == 0x3);

if (exp == 0)

{

frac >>= 1;

frac += add;

}

else if (exp == 1)

{

noSign >>= 1;

noSign += add;

exp = noSign >> 23 & 0xFF;

frac = noSign & 0x7FFFFF;

}

else

{

exp -= 1;

}

return sign << 31 | exp << 23 | frac;

}

3.58/

Long decode2(long x, long y, long z)

{

long temp = y - z;

return (temp \* x) ^ (temp << 63 >> 63);

}

3.59/

/\* dest in %rdi, x in %rsi, y in %rdx \*/

1 store\_prod:

2 movq %rdx, %rax # %rax = y(b0)

3 cqto # sign-extend y, store high byte in %rdx (b1)

4 movq %rsi, %rcx # %rcx = x(a0)

5 sarq $63, %rcx # sign-extend x, store high byte in %rcx(a1)

6 imulq %rax, %rcx # c1 = b0 \* a1

7 imulq %rsi, %rdx # c2 = a0 \* b1

8 addq %rdx, %rcx # c1 + c2

9 mulq %rsi # %rsi \* %rax, high in %rdx, low in %rax

10 addq %rcx, %rdx # c1 + c2 +%rdx

11 movq %rax, (%rdi) # store low

12 movq %rdx, 8(%rdi) # store high

13 ret

3.60/

A. x in %rdi, n in %esi, result in %rax, mask in %rdx

B. result=0, mask=1

C. mask!=0

D. mask=mask<<n

E. result=result | (x & mask)

F.

long loop(long x, long n)

{

long result = 0;

long mask;

for (mask = 1; mask != 0; mask <<= n )

{

result | = (x & mask);

}

return result;

}