



$$3.13 \quad (62)_{16} = (0110\ 0010)_2 \quad (12)_{16} = (0001\ 0010)_2$$

迭代次数	步 骤	乘 数	被 乘 数	乘 积
0	初始值	0110 0010	0001 0010	0000 0000 0000 0000
1	0 \Rightarrow 无操作 左移被乘数, 右移乘数	0110 0010	0001 0010	0000 0000 0000 0000
2	1 \Rightarrow 乘积 = 乘积 + 被乘数 左移被乘数, 右移乘数	0011 0001	0010 0100	0000 0000 0010 0100
3	0 \Rightarrow 无 左移被乘数, 右移乘数	0001 1000	0100 1000	0000 0000 0010 0100
4	0 \Rightarrow 无 左移被乘数, 右移乘数	0000 1100	1001 0000	0000 0000 0010 0100
5	0 \Rightarrow 无 左移被乘数, 右移乘数	0000 0110	1001 0000	0000 0000 0010 0100
6	1 \Rightarrow 乘积 = 乘积 + 被乘数 左移被乘数, 右移乘数	0000 0011	1001 0000	0000 0000 0010 0100
7	1 \Rightarrow 乘积 = 乘积 + 被乘数 左移被乘数, 右移乘数	0000 0001	1001 0000	0000 0000 0010 0100
8	0 \Rightarrow 不变	0000 0000	1001 0000	0000 0000 0010 0100

故 结 果 为 $(0000\ 0110\ 1110\ 0100)_2 = (06E4)_{16}$

$$3.17 \quad (33)_{16} = (2^5 + 2^4 + 2 + 1)_{10} \quad (55)_{16} = (2^6 + 2^4 + 2^2 + 1)_{10}$$

从而 $(33)_{16} \times (55)_{16} = (2^5 + 2^4 + 2 + 1) \times (55)_{16} \quad (55)_{16} = (0101\ 0101)_2$

- 1) 左移 5 位: 101 0101 00000
 2) 左移 4 位: 101 0101 0000
 3) 左移 1 位: 1010 1010
 4) 不变: 101 0101
- 相加得 $(0001\ 0000\ 1110\ 1111)_2 = (10EF)_{16}$



3.19 $(74)_8 = (111\ 100)_2$ $(21)_8 = (010\ 001)_2$

迭代次数	步骤	除数	余数/商
0	初始值	010 001	000 000 111 100
1	左移1位	010 001	000 001 111 000
	余数 = 原数	010 001	110 000 111 000
	$r < 0, r + = d$	010 001	000 001 111 000
	左移1位	010 001	000 011 110 000
2	$r - = d$	恒定	110 010 110 000
	$r < 0, r + = d$		000 011 110 000
	左移1位		000 111 100 000
3	$r - = d$		110 110 100 000
	$r < 0, r + = d$		000 111 100 000
	左移1位		001 111 000 000
4	$r - = d$		111 110 100 000
	$r < 0, r + = d$		001 111 000 000
	左移		011 110 000 000
5	$r - = d$		001 101 000 000
	$r \geq 0$, 左移1位		011 010 000 000
	尾数置1		011 010 000 001
6	$r - = d$		001 001 000 001
	$r \geq 0$, 左移		010 010 000 010
	尾数置1		010 010 000 011
结束	余数左移1位		001 001 000 011

故商为 $(000\ 011)_2 = (3)_8$, 余数为 $(001\ 001)_2 = (11)_8$



$$3.28 \quad (-1.5625 \times 10^{-1})_{10} = (-0.00101)_2 = (0.101 \times 2^{-2})_2$$

① 尾数: 0101 ~~补码~~ 补码为 1011 0000 0000 0000 0000

② 指数: -2 为 0000 0101

故为 1011 0000 0000 0000 0000 0000 0101

比较: 1) 精确度: IEEE 754 尾数 23 位, 而该模式 24 位, 略高.

2) 范围: 均为 8 位指数, 差不多.

$$3.39 \quad 1.666015625 \times 10^0 = 1.101010101 \times 2^0$$

$$1.9760 \times 10^4 = 100110100110000 = 1.0011010011 \times 2^{14}$$

$$-1.9744 \times 10^4 = -100110100100000 = -1.001101001 \times 2^{14}$$

$$\textcircled{1} \quad 0+14=14 \quad 1.101010101 \times 1.0011010011 = 10.000000100110001111 = 1.000000010110001111 \times 2^1$$

$$= 1.0000000101 \times 2^1$$

$$\text{为 } 1.0000000101 \times 2^{15}$$

$$\textcircled{2} \quad 0+14=14 \quad 1.101010101 \times 1.001101001 = 10.0000001111011101 = 1.0000001111011101 \times 2^1$$

$$= 1.000000100 \times 2^1$$

$$\text{为 } -1.000000100 \times 2^{15}$$

$$\text{故为 } 0.000000001 \times 2^{15} = 1 \times 2^5 = (32)_{10}$$

3.47 假设: ① 存在能一次按加载并存储 128 位数据的指令

② 存在执行 64 位乘法和加法的指令 mul-64, sum-r

③ 存在可以对数据进行位移操作的指令 lshift-16, rshift-16

④ 算法的循环部分可以充分利用指令集的并行操作

代码: 1) 加载数组到寄存器: load fr, f

2) 个循环开始

load_start: load sig_in-r, sig_in[0:7] 加载 sig_in 一部分到 sig_in-r

mul-64 temp_mul, sig_in-r, fr 乘法

sum-r temp, temp_mul 加法

lshift-16 temp

rshift-16 sig_in-r

位移操作



南开大学

作业纸

系别_____ 班级_____ 姓名_____ 第 4 页

循环乘法、求和、位移3次

之后加载 sig-in 另一部分到 sig-in-r load sig-in-r, sig-in[8:12]

乘法、求和、位移3次

将计算结果存储到 sig-out 中 store sig-out, temp

重复以上步骤至循环结束。