

# 计组第四次作业

姓名：李雅帆

学号：2213041

## 3. 13

62×12

迭代次数	步骤	被乘数	乘积/乘数
0	初始值	110010	000000001010
1	1sb:0⇒无操作	110010	000000001010
	乘数右移	110010	000000000101
2	乘积=乘积+被乘数	110010	110010000101
	被乘数左移	110010	011001000010
3	1sb:0⇒无操作	110010	011001000010
	被乘数左移	110010	001100100001
4	乘积=乘积+被乘数	110010	111110100001
	被乘数左移	110010	011111010000
5	1sb:0⇒无操作	110010	011111010000
	被乘数左移	110010	001111101000
6	1sb:0⇒无操作	110010	001111101000
	被乘数左移	110010	000111110100

## 3. 17

0x33=32+16+2+1

先将 55 分别左移 5 位，4 位和 1 位，再将这三个结果与 55 相加得到最终结果

## 3. 19

74/21=3……11

迭代次数	步骤	除数	余数/商
0	初始值	010001	000000111100
1	余数左移	010001	000001111000
	余数=余数-除数	010001	111000111000
	余数<0⇒+除数	010001	000001111000
2	余数左移	010001	000011110000
	余数=余数-除数	010001	110010110000
	余数<0⇒+除数	010001	000011110000
3	余数左移	010001	000111100000
	余数=余数-除数	010001	110110110000
	余数<0⇒+除数	010001	000111100000
4	余数左移	010001	001111000000
	余数=余数-除数	010001	111110000000
	余数<0⇒+除数	010001	001111000000

### 3. 28

$$-1.5625 \times 10^{-1} = -0.5625 \times 10^0 = -0.00101 \times 2^0$$

二进制小数点右移2位得到  $-0.101 \times 2^{-2}$

指数: -2

小数部分:  $-0.10100000000000000000000000000000$

101100000000000000000000000000101

### 3. 39

$$1.666015625 \times 10^0 \times (1.9760 \times 10^4 - 1.9744 \times 10^4)$$

$$A. 1.666015625 \times 10^0 = 1.1010101010 \times 2^0$$

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

$$C. -1.9744 \times 10^4 = -1.0011010010 \times 2^{14}$$

指数相加:  $0 + 14 = 14$

$$A \quad 1.1010101010$$

$$B \quad \begin{array}{r} \times 1.0011010011 \\ \hline \end{array}$$

$$11010101010$$

$$11010101010$$

$$11010101010$$

$$11010101010$$

$$11010101010$$

$$11010101010$$

$$10.000001001100001111$$

$$A \times B \quad 1.000000101 \times 2^{15}$$

$$0 + 14 = 14$$

$$\begin{array}{r}
A \quad 1.10101010 \\
C \quad 1.00110100 \\
\hline
110101010 \\
110101010 \\
110101010 \\
110101010 \\
110101010 \\
\hline
10.0000001111011010 \\
A \times C \quad -1.000000100 \times 2^{15} \\
A \times B \quad 1.000000101 \times 2^{15} \\
A \times C \quad -1.000000100 \times 2^{15} \\
\hline
A \times B + A \times C \quad 0.000000101 \times 2^{15} \\
A \times B + A \times C \quad 1.000000000 \times 2^5
\end{array}$$

### 3.47

在面向具有 SIMD 指令集和 128 位寄存器的处理器的优化中，可以利用子字并行操作来最大限度地提高性能。在这种情况下，可以假设处理器支持 128 位宽度的 SIMD 指令集，比如 AVX2 或者 NEON。

首先可以将输入信号 sig\_in 加载到寄存器中，然后利用寄存器宽度执行并行操作。通过使用 128 位寄存器，我们可以一次性处理多个数组元素，并且可以在一个时钟周期内完成多个乘法和加法操作。

在优化后的代码中，将尽可能多的计算合并成一个指令，以减少存储器和寄存器间的数据传输。可以使用 128 位宽度的加载和存储指令来最小化数据传输量。此外还可以利用乘法和加法指令的并行性来同时处理多个乘法和加法操作。

基于以上假设和优化策略，可以使用适当的 128 位宽度的 SIMD 指令集，结合加载、存储、乘法和加法指令，对给定的 C 代码进行优化。在实际进行优化时需要了解目标处理器的具体指令集细节，以确保代码可以正确地利用 SIMD 指令集的性能优势。