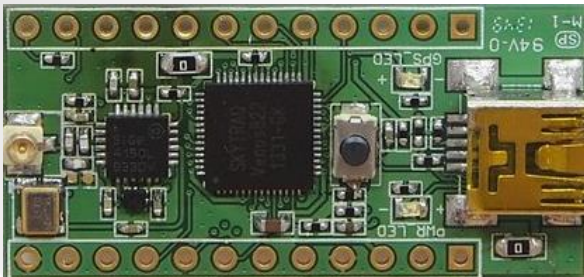


计算机组成原理

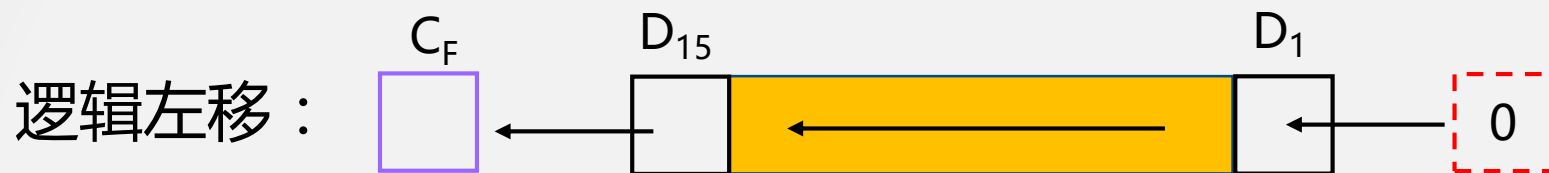
第三章 运算方法与运算器

3.3 原码一位乘法



1

移位操作及其意义



数据整体左移一位，最高位 D_{15} 被移出至 C_F ，最低位 D_1 补0

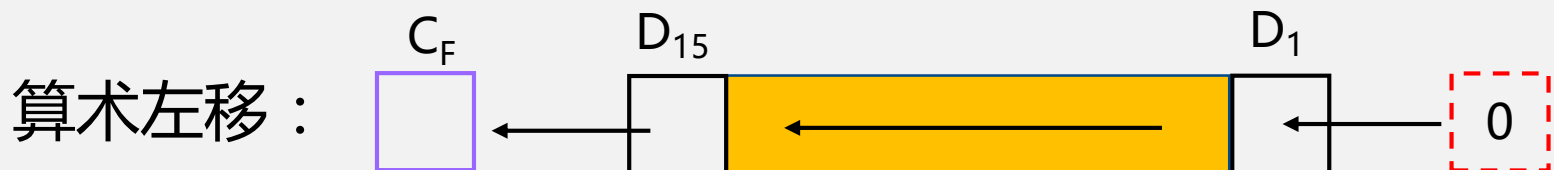
移位前

0 1 1 0 1 1 1 0

逻辑左移后

0

1 1 0 1 1 1 0 0



数据整体左移一位，最高位 D_{15} 被移至 C_F ，最低位 D_1 补0

移位前

0 1 1 0 1 1 1 0

逻辑左移后

0

1 1 0 1 1 1 0 0

相当于乘2

1

移位操作及其意义

逻辑右移：

数据整体右移一位，最高位 D_{15} 补0，最低位 D_1 被移出

移位前

1 1 1 0 1 1 1 0

逻辑右移后

0 1 1 1 0 1 1 1

算术右移：

数据整体右移一位，最高位 D_{15} 被复制填补 D_{15} ，最低位 D_1 被移出

移位前

1 1 1 0 1 1 1 0

逻辑左移后

1 1 1 1 0 1 1 1

相当于除2

2

二进制乘法的手工计算过程

$$\begin{array}{r} 0.010 \\ \times 0.101 \\ \hline 0010 \\ 0000 \\ 0010 \\ + 0000 \\ \hline 0001010 \end{array}$$

a. 说明乘法可由加法实现

b. 存在的问题：

- 需要多输入的全加器（最多为 $n+1$ ）；
- 需要长度为 $2n$ 的积寄存器；
- 对应乘数的不同位，部分积左移次数不同，且乘法过程中总移位次数多。

2

二进制乘法的手工计算过程

$$\begin{array}{r}
 0.010 \\
 \times 0.101 \\
 \hline
 0010 \\
 0000 \\
 0010 \\
 + 0000 \\
 \hline
 0001010
 \end{array}$$



$$\begin{array}{r}
 0.010 \\
 \times 0.101 \\
 \hline
 \rightarrow 0010 \\
 + 0000 \\
 \hline
 00010 \\
 \rightarrow 00010 \\
 + 0010 \\
 \hline
 001010 \\
 \rightarrow 001010 \\
 + 0000 \\
 \hline
 0001010
 \end{array}$$

如何解决上述问题（改进的方法）

- 需要多输入的全加器（最多为 $n+1$ ）
 ↳ 基于FA的循环累加0或被乘数
- 针对乘数不同位部分积左移次数不同的问题
 ↳ 右移部分积！乘数寄存器
- 需要长度为 $2n$ 的积寄存器
 ↳ 从部分积和乘数寄存器取结果

原码一位乘法算法

- 符号位单独参加运算，数据位取绝对值参加运算。

- 运算法则：

设： $[X]_{\text{原}} = X_0.X_1X_2\dots X_n$ $[Y]_{\text{原}} = Y_0.Y_1Y_2\dots Y_n$

则： $P_0 = X_0 \oplus Y_0$ $|P| = |X| \cdot |Y|$

- 运算过程采用改进的乘法运算方法。

3

原码一位乘法算法

$$\begin{array}{r}
 0.010 \\
 \times 0.101 \\
 \hline
 0010 \\
 \rightarrow 0010 \\
 + 0000 \\
 \hline
 00010 \\
 \rightarrow 00010 \\
 + 0010 \\
 \hline
 001010 \\
 \rightarrow 001010 \\
 + 0000 \\
 \hline
 0001010
 \end{array}$$

例1 已知 $X = 0.110$ $Y = -0.101$ 用原码一位乘法求 $X * Y$

解: $[X]_{\text{原}} = 0.110$

$[Y]_{\text{原}} = 1.101$

部分积	乘数 / 判断位	说明
00.000	$Y_0.101$	$Y_3 = 1$ 部分积 + $ X $
+ 00.110		
00.110		每次运算结果右移1位
→ 00.011	$0 Y_0.10$	$Y_3 = 0$ 部分积 + 0
+ 00.000		
00.011		
→ 00.001	$10 Y_0.1$	$Y_3 = 1$ 部分积 + $ X $
+ 00.110		
00.111		
→ 00.011	$110 Y_0$	

$$X * Y = (0 \oplus 1).011110 = 1.011110$$