浮点三角函数计算：完整推导流程

> 基于论文 *"Floating-Point Trigonometric Functions for FPGAs"* (Detrey & de Dinechin, FPL 2007)

---

1. 问题定义

输入： 任意浮点数 x
输出： $\sin(x)$ 和 $\cos(x)$，以浮点数格式输出，保证 faithful rounding（忠实舍入）

1.1 浮点数格式

$$x = (-1)^{S_x} \times 1.F_x \times 2^{E_x - E_0}$$

字段	位宽	含义
exn_x	2	异常编码（零、无穷、NaN）
S_x	1	符号位
E_x	w_E	指数（存储值）
F_x	w_F	尾数小数部分

其中 $E_0 = 2^{w_E - 1} - 1$ 为指数偏移。

$1.F_x$ 表示在 F_x 前面加一个**隐含的 1**，构成 $(w_F + 1)$ 位的完整尾数，值域为 $[1, 2)$。

单精度参数：$w_E = 8, w_F = 23, E_0 = 127$。

---

2. 总体架构

```
输入 x (浮点)
      │
      ▼
┌──────────────────┐
│   参数归约          │  → 输出: k(3位), Y(定点), (Ey, My)(浮点)
│ (Argument Reduction)│
└──────────────────┘
        │
        │
   ┌─────┴─────┐
   ▼          ▼
┌──────┐ ┌──────┐
│ sin  │ │ cos  │  → 分别计算 sin(πy/4) 和 cos(πy/4)
│ 通路  │ │ 通路  │
└──────┘ └──────┘
    │        │
    │        │
    ▼        ▼
┌──────────────────┐
│   象限重构          │  → 根据 k mod 4 和 Sx 组合最终结果
│ (Reconstruction)  │
└──────────────────┘
        │
        │
        ▼
┌──────────────────┐
│   异常处理          │  → 处理 0, ±∞, NaN 等特殊情况
│ (Exception Handling)│
└──────────────────┘
        │
        │
        ▼
   sin(x), cos(x) (浮点输出)
```

3. 参数归约：原理推导

3.1 传统方法

$\sin$ 和 $\cos$ 以 2π 为周期，可将 x 归约到一个小区间：

$$\alpha = x - k \cdot \frac{\pi}{2}, \quad \alpha \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$$

其中 $k = \text{round}\left(\frac{2x}{\pi}\right)$。

传统方法的计算步骤：

步骤	运算	说明
1	$x \times \frac{2}{\pi}$	乘以无理数常量
2	$k = \text{round}(\cdots)$	取整
3	$k \times \frac{\pi}{2}$	**再乘一次无理数常量**
4	$\alpha = x - k \cdot \frac{\pi}{2}$	**做一次减法**
5	求 $\sin(\alpha), \cos(\alpha)$	函数求值

步骤 3 和 4 是额外的代价。

3.2 Markstein 变体

计算 $x \times \frac{4}{\pi}$ 并拆分：

$$x \cdot \frac{4}{\pi} = k + y, \quad k = \text{round}\left(\frac{4x}{\pi}\right), \quad y \in \left[-\frac{1}{2}, \frac{1}{2}\right]$$

等价性推导：

由 $x \cdot \frac{4}{\pi} = k + y$，两边乘以 $\frac{\pi}{4}$：

$$x = (k + y) \cdot \frac{\pi}{4} = k \cdot \frac{\pi}{4} + y \cdot \frac{\pi}{4}$$

代入三角函数：

$$\sin(x) = \sin\left(k \cdot \frac{\pi}{4} + y \cdot \frac{\pi}{4}\right)$$

$$\cos(x) = \cos\left(k \cdot \frac{\pi}{4} + y \cdot \frac{\pi}{4}\right)$$

由于 sin/cos 的周期为 $2\pi = 8 \cdot \frac{\pi}{4}$，只需 $k \bmod 8$（3 位）即可确定恒等式关系。

与传统 α 的关系：

$$\alpha = y \cdot \frac{\pi}{4}$$

由于 $y \in \left[-\frac{1}{2}, \frac{1}{2}\right]$，因此

$$\alpha \in \left[-\frac{\pi}{8}, \frac{\pi}{8}\right]$$

所以 $\sin(\alpha) = \sin\left(\frac{\pi}{4}y\right)$，$\cos(\alpha) = \cos\left(\frac{\pi}{4}y\right)$。两种方法在数学上完全等价。

Markstein 方法的计算步骤（硬件视角）：

步骤	运算	说明
1	$x \times \frac{4}{\pi}$	乘以无理数常量（仅此一次）
2	$k = \text{round}(\cdot)$	取目标窗口的整数低位，并结合舍入位实现"就近取整"（取位 + 条件加 1）
3	$y = \left(x \cdot \frac{4}{\pi}\right) - k$	由同一窗口得到（必要时做补码/借位调整），$y \in \left[-\frac{1}{2}, \frac{1}{2}\right]$
4	求 $\sin\left(\frac{\pi}{4}y\right)$, $\cos\left(\frac{\pi}{4}y\right)$	函数求值

节省： 一次乘法 $\left(k \times \frac{\pi}{2}\right)$ 和一次减法 $\left(x - k \cdot \frac{\pi}{2}\right)$。

3.3 为什么选择 $\frac{4}{\pi}$ 而不是 $\frac{2}{\pi}$

$\frac{4}{\pi}$ 对应以 $\frac{\pi}{4}$ **为单位**度量 x，实现**八分区（octant）归约**。

$\frac{2}{\pi}$ 对应以 $\frac{\pi}{2}$ 为单位度量，实现四分区（quadrant）归约。

选择 $\frac{\pi}{4}$ 的关键原因——$\cos\left(\frac{\pi}{4}y\right)$ **的值域特性**（此处 $y \in \left[-\frac{1}{2}, \frac{1}{2}\right]$，对应角度区间 $\left[-\frac{\pi}{8}, \frac{\pi}{8}\right]$）：

归约区间	$\cos$ 的值域	$\cos$ 的指数
$\left[0,\ \frac{\pi}{2}\right]$	$[0,\ 1]$	不确定（可接近 0）
$\left[0,\ \frac{\pi}{8}\right]$	$\left[\cos\left(\frac{\pi}{8}\right),\ 1\right] \approx [0.924,\ 1]$	**确定！恒为 0**

$\frac{\pi}{8}$（对应 $y \in \left[-\frac{1}{2}, \frac{1}{2}\right]$）已经足以保证 $\cos$ 的浮点指数确定。在此区间上：

- $\cos$ 永远 $\geq \cos\left(\frac{\pi}{8}\right) > \frac{1}{2}$，首位恒为 1 → **可用纯定点计算**，硬件便宜
- $\sin$ 可以任意接近 0，指数不定 → **必须用浮点**，但无法避免

如果选 $\frac{\pi}{2}$，$\cos$ 也可能接近 0，两条通路都需要浮点归一化，硬件成本翻倍。

3.4 $k \bmod 8$ 为什么折叠为 $k \bmod 4$

8 个八分区中，相距 4 的两个（即相距 π）具有对称性：

$$\sin(x+\pi) = -\sin(x), \quad \cos(x+\pi) = -\cos(x)$$

所以八分区 j 和八分区 $j+4$ 的恒等式模式完全相同，只差一个全局取反。

展开全部 8 种情况验证：

$k \bmod 8$	$\sin(x)$	$\cos(x)$	等价于
0	$+\sin\left(\frac{\pi}{4}y\right)$	$+\cos\left(\frac{\pi}{4}y\right)$	$k \bmod 4 = 0$
1	$+\cos\left(\frac{\pi}{4}y\right)$	$-\sin\left(\frac{\pi}{4}y\right)$	$k \bmod 4 = 1$
2	$-\sin\left(\frac{\pi}{4}y\right)$	$-\cos\left(\frac{\pi}{4}y\right)$	$k \bmod 4 = 2$
3	$-\cos\left(\frac{\pi}{4}y\right)$	$+\sin\left(\frac{\pi}{4}y\right)$	$k \bmod 4 = 3$
4	$-\sin\left(\frac{\pi}{4}y\right)$	$-\cos\left(\frac{\pi}{4}y\right)$	$= k \bmod 4 = 2$
5	$-\cos\left(\frac{\pi}{4}y\right)$	$+\sin\left(\frac{\pi}{4}y\right)$	$= k \bmod 4 = 3$
6	$+\sin\left(\frac{\pi}{4}y\right)$	$+\cos\left(\frac{\pi}{4}y\right)$	$= k \bmod 4 = 0$
7	$+\cos\left(\frac{\pi}{4}y\right)$	$-\sin\left(\frac{\pi}{4}y\right)$	$= k \bmod 4 = 1$

8 种只有 4 种“交换/符号模式”由 $k \bmod 4$ 决定；另外还需要 1 个比特决定是否进行全局取反。

实现上最方便的是保留 $k \bmod 8$ 的 3 位:

- 用 $k \bmod 4$（低 2 位）决定 sin / cos 的交换与局部符号;
- 用 $k[2]$（等价于 $\lfloor k/4 \rfloor \bmod 2$，也就是 $k \bmod 8$ 的高位）决定是否对 sin 与 cos 两者同时取反。

---

4. 参数归约：实现细节

4.1 核心难题：大 x 的精度问题

x 可以非常大（无偏置指数 $e = E_x - E_0$ 的动态范围很宽），但 $x \times \frac{4}{\pi}$ 的有用信息只在低几位整数和小数部分。

如果用普通浮点乘法（$\frac{4}{\pi}$ 只存 w_F 位）：

```
x = 2^50

x × (4/π) ≈ 2^50 × 1.273...

乘积 ≈ 1.273 × 2^50

整数部分有 51 位，但 4/π 只有 23 位精度
→ 小数部分完全是噪声，结果不可用
```

4.2 Payne-Hanek 算法：只提取有用窗口

4.2.1 部分积位置公式

将 $\frac{4}{\pi}$ 表示为逐位序列，$1.F_x$ 也逐位展开:

$$\frac{4}{\pi} = \sum_{i=-\infty}^{0} c_i \cdot 2^i, \qquad 1.F_x = \sum_{j=0}^{w_F} f_j \cdot 2^{-j}$$

其中 $f_0 = 1$（隐含位），f_1 到 f_{w_F} 是 F_x 的各位。

乘法展开：

$$x \times \frac{4}{\pi} = \sum_{i,j} (f_j \cdot c_i) \cdot 2^{i+E-j}$$

关键公式：c_i 和 f_j 的部分积落在结果的第 $(i+E-j)$ 位。

对于某个固定的 c_i，它与所有 f_j 相乘后，影响结果的位置范围：

$$c_i \text{ 影响的区间} = [i+E-w_F, \quad i+E]$$

4.2.2 确定可丢弃的高位 c_i

如果 c_i 的所有部分积都落在结果第 3 位以上（不需要的高位整数区域），则可丢弃。

条件：c_i 的**最低**部分积 $>$ 位置 2

$$i+E-w_F > 2 \quad \Longrightarrow \quad i > w_F - E + 2$$

所有 $i > w_F - E + 2$ 的 c_i 可以左截断丢弃。

4.2.3 确定可丢弃的低位 c_i

如果 c_i 的所有部分积都落在精度需求以下，也可丢弃。

我们需要的最低结果位是位置 $-(w_F + g + g_K)$。

条件：c_i 的**最高**部分积 $<$ 位置 $-(w_F + g + g_K)$

$$i + E < -(w_F + g + g_K) \quad \Longrightarrow \quad i < -(E + w_F + g + g_K)$$

4.2.4 窗口大小

需要保留的 c_i 范围：

$$-(E + w_F + g + g_K) \le i \le (w_F - E + 2)$$

窗口宽度：

$$W = (w_F - E + 2) - (-(E + w_F + g + g_K)) + 1 = 2w_F + g + g_K + 3$$

E 被消掉了！ 窗口宽度恒为 $2w_F + g + g_K + 3 \approx 3w_F + 5$，与输入的指数无关。

E 只决定窗口的**起始位置**（从 $\frac{4}{\pi}$ 的哪里开始截取）。

4.2.5 截断不引入误差的证明

左截断安全性： 被丢弃的最低一个高位是 $i = w_F - E + 3$，它与 f_{w_F} 相乘后落在：

$$(w_F - E + 3) + E - w_F = 3$$

恰好在位置 3 —— 我们不需要的区域。更高的 c_i 产生的部分积更高。多个部分积相加的进位只会向上传播（位置 4, 5, ...），不会向下进入位置 2 及以下。

右截断安全性： 被丢弃的低位 c_i 的最高部分积低于精度阈值 $-(w_F + g + g_K)$，对可见结果无影响。

4.2.6 $\frac{4}{\pi}$ 的总存储量

不同的 x（不同的指数 E）需要从 $\frac{4}{\pi}$ 的不同位置截取窗口。预存的 $\frac{4}{\pi}$ 需要覆盖所有可能的指数，总位宽约：

$$\text{总存储} \approx 2^{w_E - 1} + 3 \cdot w_F \text{ 位}$$

4.3 保护位 g 和 gK

参数	含义	确定方式	典型值
g	舍入保护位：补偿后续 HOTBM、乘法、减法的累积舍入误差	逐级误差分析	$g = 2$
g_K	Kahan 保护位：补偿 y 接近 0 时的前导零	Kahan-Douglas 算法（数论搜索）	$g_K \approx w_F$

g 的作用： 中间计算多保留 g 位精度，使累积舍入误差落在"不会被输出"的低位：

```
                wF 位（输出）          g 位（丢弃）
              ┌───────────────────┬───────┐
计算精度：    │ 有效输出位        │ 吸收误差 │
              └───────────────────┴───────┘
```

g_K 的作用： y 可能非常接近 0，前导零"吃掉"有效位，g_K 预留空间保证归一化后仍有 w_F 位

精度：

```
y = 0.00000001 10110100
      ├─ gK 个零 ─┤├ wF 位有效 ┤
归一化后：1.0110100 × 2^{-8}，仍有 wF 位精度
```

g_K 的具体值由 Kahan-Douglas 算法确定：在所有合法浮点输入中搜索使 $\text{frac}\left(x \times \frac{4}{\pi}\right)$ 最小的 x，该最小值的前导零数量即为 g_K。这是一个有理逼近无理数的数论问题，可通过 $\frac{4}{\pi}$ 的连分数展开求解。

4.4 乘法与 k、y 的提取

4.4.1 乘法过程

$$\text{步骤 1：计算偏移} \quad E = E_x - E_0 + 1$$

$$\text{步骤 2：从预存的 } \frac{4}{\pi} \text{ 中，以 } E \text{ 为偏移，提取宽度为 } W = 2w_F + g + g_K + 3 \text{ 的窗口 } C$$

$$\text{步骤 3：计算定点乘积} \quad P = 1.F_x \times C$$

乘法器规模：$(w_F + 1)$ 位 $\times\ W$ 位，乘积位宽约 $w_F + 1 + W$ 位。

4.4.2 乘积的结构

```
乘积 P = 1.Fx × C:


┌──────────────┬─────────┬──────────────────────────┬──────────┐
│ 污染区 (wF位) │ k (3位)  │    y (wF + g + gK 位)      │ 超精度     │
│ 高位无效       │ 整数低位  │     小数部分                 │ 可丢弃     │
└──────────────┴─────────┴──────────────────────────┴──────────┘
```

污染区的成因： 被左截断的高位 c_i 本来会与 $1.F_x$ 的低位相乘，产生进位传入乘积的高位区域。这些进位缺失，导致乘积的最高约 w_F 位不可信。但这些位对应结果的第 3 位及以上（不需要的高位整数），所以不影响 k 和 y 的提取。

4.4.3 k 和 y 的提取

k： 直接从乘积的可信区域中取 3 位整数，即乘积中紧接污染区之后的 3 位。$k \bmod 4$ 用于查

重构恒等式表。

y： 取 k 之后的 $(w_F + g + g_K)$ 位作为定点小数，$y \in \left[-\frac{1}{2}, \frac{1}{2}\right]$。

4.5 双路径架构（Dual-Path）

y 需要同时以两种格式输出：

- **定点 Y（$w_F + g$ 位）**：给 cos 通路使用
- **浮点 (E_y, M_y)**：给 sin 通路使用

为什么 cos 用定点： 由于 $y \in [-\frac{1}{2}, \frac{1}{2}]$，

$$\cos\left(\frac{\pi}{4}y\right) \in \left[\cos\left(\frac{\pi}{8}\right),\ 1\right] \approx [0.924,\ 1]$$

值不会接近 0，浮点指数恒定，定点即可。

为什么 sin 用浮点： $\sin\left(\frac{\pi}{4}y\right) \approx \frac{\pi}{4} \cdot y$，$y$ 可以任意接近 0，指数不定，必须用浮点。

从定点 y 到浮点 (E_y, M_y) 需要**前导零计数（LZC）+ 桶形移位器**，延迟很高。为此采用双路径优化：

路径	条件	逻辑	优势
Close path	x 接近 0（$E_x < E_0 - 1$）	$y \approx \frac{4}{\pi}x$，指数从 x 的指数直接推出 → 快速得到 (E_y, M_y)；再通过变量移位得到定点 Y	跳过 LZC
Far path	x 远离 0（$E_x \geq E_0 - 1$）	先通过大乘法器得到定点 Y；再用 LZC + 移位得到浮点 (E_y, M_y)	跳过变量移位

每条路径省略一个昂贵操作，通过 MUX 选择输出。

5. $\sin\left(\frac{\pi}{4}y\right)$ 和 $\cos\left(\frac{\pi}{4}y\right)$ 的求值

5.1 HOTBM 方法

使用 **HOTBM（Hardware-Oriented Table-Based Method）**：

1. 用 minimax 多项式逼近目标函数
2. 构建由**查找表 + 幂运算单元 + 小乘法器**组成的优化并行架构
3. 输出 faithful rounding 精度的定点结果

5.2 cos 通路（纯定点）

$\cos\left(\frac{\pi}{4}y\right)$ 的首位恒为 1（因为 $y \in [-\frac{1}{2}, \frac{1}{2}]$ 时 $\cos \geq \cos(\frac{\pi}{8}) > 0.5$），无需计算首位。
HOTBM 实际求值：

$$f_{\cos}(y) = 1 - \cos\left(\frac{\pi}{4}y\right)$$

输入：定点 Y（$w_F + g$ 位）
输出：定点 $f_{\cos}$（$w_F + g$ 位），最后用 $1 - f_{\cos}$ 恢复 $\cos$ 值。

5.3 sin 通路（定点 $\times$ 浮点）

$\sin\left(\frac{\pi}{4}y\right)$ 可以任意接近 0，不能直接用定点。巧妙分解：

$$\sin\left(\frac{\pi}{4}y\right) = y \times \frac{\sin\left(\frac{\pi}{4}y\right)}{y}$$

右侧 $\frac{\sin\left(\frac{\pi}{4}y\right)}{y}$ 的 Taylor 展开：

$$\frac{\sin\left(\frac{\pi}{4}y\right)}{y} = \frac{\pi}{4} - \frac{1}{6}\left(\frac{\pi}{4}\right)^3 y^2 + \cdots \approx \frac{\pi}{4} + O(y^2)$$

这是一个在 $y = 0$ 处约为 $\frac{\pi}{4} \approx 0.785$ 的**平滑函数**，值域有限，可用定点计算。

HOTBM 实际求值：

$$f_{\sin}(y) = \frac{\pi}{4} - \frac{\sin\left(\frac{\pi}{4}y\right)}{y}$$

得到 $f_{\sin}$ 后：

$$\sin\left(\frac{\pi}{4}y\right) = y \times \left(\frac{\pi}{4} - f_{\sin}(y)\right) = M_y \times 2^{E_y} \times (\text{定点值})$$

这是一个浮点数 $\times$ 定点数的乘法，比完整浮点乘法更简单。结果的指数就是 E_y。

---

6. 象限重构

根据 $k \bmod 4$ 和输入符号 S_x，从 $\sin\left(\frac{\pi}{4}y\right)$ 和 $\cos\left(\frac{\pi}{4}y\right)$ 组合出最终结果：

$k \bmod 4$	$\sin(x)$	$\cos(x)$
0	$+\sin\left(\frac{\pi}{4}y\right)$	$+\cos\left(\frac{\pi}{4}y\right)$
1	$+\cos\left(\frac{\pi}{4}y\right)$	$-\sin\left(\frac{\pi}{4}y\right)$
2	$-\sin\left(\frac{\pi}{4}y\right)$	$-\cos\left(\frac{\pi}{4}y\right)$
3	$-\cos\left(\frac{\pi}{4}y\right)$	$+\sin\left(\frac{\pi}{4}y\right)$

输入符号处理：

- $\sin(-x) = -\sin(x) \rightarrow$ 若 $S_x = 1$，对 sin 结果取反
- $\cos(-x) = \cos(x) \rightarrow$ cos 结果不变

异常处理：

- $x = 0 \rightarrow \sin = 0, \cos = 1$
- $x = \pm\infty \rightarrow \sin = \text{NaN}, \cos = \text{NaN}$
- $x = \text{NaN} \rightarrow \sin = \text{NaN}, \cos = \text{NaN}$

---

7. 误差分析

目标：faithful rounding（忠实舍入），即结果的相对误差 $< 2^{-w_F}$。

误差来源与控制：

来源	误差量级	控制方式
参数归约中的截断	由 $g_K + g$ 位吸收	窗口宽度 $2w_F + g + g_K + 3$
HOTBM 多项式逼近	≤ 1 ulp	minimax 逼近保证
中间运算舍入	每级 ≤ 1 ulp	$g = 2$ 位保护位
最终归一化舍入	≤ 1 ulp	标准舍入逻辑

累积误差在 $g = 2$ 位保护位的保护下，不会影响最终输出的 w_F 位精度。

---

8. 完整数值例子

输入

假设简化参数：$w_F = 4, g = 2, g_K = 4$

输入浮点数：

$$x = 1.1001_2 \times 2^3 = 1.5625 \times 8 = 12.5$$

即 $E_x - E_0 = 3, E = 3$, 尾数 $1.F_x = 1.1001_2 = 1.5625_{10}$

第一步：预存 $\frac{4}{\pi}$

$\frac{4}{\pi} \approx 1.27324\ldots$ 的二进制展开：

```
4/π = 1.0100010111100110001100111100...

按位编号（小数点 = 位0与位-1之间）：

位号:  1   0  -1  -2  -3  -4  -5  -6  -7  -8  -9 -10 -11 -12 -13 -14 -15 ...
值:    1   0   1   0   0   0   1   0   1   1   1   1   0   0   1   1   0  ...

即: 4/π = 1·2^0 + 0·2^(-1) + 1·2^(-2) + 0·2^(-3) + ...
```

第二步：确定窗口范围

由推导公式，需要保留的 c_i 范围：

$$-(E + w_F + g + g_K) \leq i \leq w_F - E + 2$$

代入 $E = 3$, $w_F = 4$, $g = 2$, $g_K = 4$:

$$-(3 + 4 + 2 + 4) \leq i \leq 4 - 3 + 2$$

$$-13 \leq i \leq 3$$

从 $\frac{4}{\pi}$ 中截取第 3 位到第 -13 位，共 17 位：

```
位号:  3   2   1   0  -1  -2  -3  -4  -5  -6  -7  -8  -9 -10 -11 -12 -13
值:    0   0   1   0   1   0   0   0   1   0   1   1   1   1   0   0   1
```

注：位号 3 和 2 对应 $\frac{4}{\pi}$ 中 2^3 和 2^2 的系数。由于 $\frac{4}{\pi} \approx 1.27 < 4$，所以 2^2 位和 2^3 位为 0。

实际上窗口的高位就是 $\frac{4}{\pi}$ 本身左移 E 位后的对应段。等效地，这是计算 $2^E \times \frac{4}{\pi}$ 后保留从整数第 3 位到小数第 $(w_F + g + g_K)$ 位的片段。

可验证：$2^3 \times \frac{4}{\pi} = 8 \times 1.27324\ldots = 10.186\ldots$

```
10.186... 的二进制 ≈ 1010.001011110011...

                       位3 位2 位1 位0 . 位-1 位-2 ...
                        1   0   1   0    0    0   1 ...

→ 与窗口的高位 "0 0 1 0 1 0 0 ..." 一致
```

等等，让我重新对齐。$2^3 \times \frac{4}{\pi} = 10.186$:

```
10.186 的二进制：
10 = 1010, 0.186... ≈ 0.001011111...
完整：1010.001011110011...

位号:  3  2  1  0 . -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 ...
值:    1  0  1  0    0  0  1  0  1  1  1  1  0   0  ...
```

所以窗口 C 应修正为:

```
位号:  3   2   1   0  -1  -2  -3  -4  -5  -6  -7  -8  -9 -10 -11 -12 -13
值:    1   0   1   0   0   0   1   0   1   1   1   1   0   0   1   1   0
```

第三步：乘法

$$\text{被乘数：} 1.F_x = 1.1001_2 \quad (5\text{ 位定点数，值} = 1.5625)$$

$$\text{乘数：窗口 } C \quad (17\text{ 位})$$

将两者都视为整数相乘（小数点位置最后调整）：

```
1.Fx 作为整数 M = 11001 (二进制) = 25 (十进制)

窗口 C 作为整数 = 10100010111100110 (二进制)
```

但为了清晰说明，我们直接用十进制计算真实值:

$$x \times \frac{4}{\pi} = 12.5 \times 1.27324\ldots = 15.9155\ldots$$

第四步：从乘积中提取 k 和 y

$$x \times \frac{4}{\pi} = 15.9155\ldots$$

取整（round to nearest）：$15.9155 \to k = 16$（因为 $0.9155 > 0.5$，向上取整）

$$y = 15.9155 - 16 = -0.0845$$

$$k = 16 = 10000_2, \quad k \bmod 4 = 0$$

从二进制乘积中提取的过程：

$15.9155\ldots$ 的二进制：$15 = 1111_2$，$0.9155 \approx 0.111010\ldots_2$

```
二进制: 1111.111010...

位号:  3   2   1   0 .  -1  -2  -3  -4  -5  -6
值:    1   1   1   1     1   1   1   0   1   0 ...
```

乘积总长约 22 位 ($5 + 17$)。最高 $w_F = 4$ 位是污染区，丢弃：

```
乘积位排列：

┌─────────────┬─────────┬─────────────────────────┐
│ 污染区 (4位)  │ k (3位)  │ y (wF+g+gK = 10位)       │
│ [丢弃]       │ 位2,1,0  │ 位-1 到 位-10            │
└─────────────┴─────────┴─────────────────────────┘
```

但由于 15.9155 取整到 16 后 y 为负数，实际硬件中 k 和 y 的提取方式是：

取乘积的**位 2, 1, 0** 作为 k 的低 3 位：$k[2:0] = 111_2$，但注意小数部分 $0.9155 > 0.5$，所以需要 $+1$ 调整，最终 $k = 1111_2 + 1 = 10000_2$，$k \bmod 8 = 0$。

$$y = -(1 - 0.9155) = -0.0845$$

y 的二进制：$y = -0.0845 \approx -0.000101011\ldots_2$

有 2 个前导零，归一化时需要左移 3 位：

$$E_y = -3, \quad M_y = 1.01011\ldots_2$$

第五步：生成两种输出格式

定点 Y（给 cos 用）：

取 y 的定点表示，保留 $w_F + g = 6$ 位：

$$|y| = 0.0845 \approx 0.000101_2 \quad (6\text{位定点})$$

$$Y = 0.000101_2$$

浮点 (E_y, M_y)（给 sin 用）：

$$|y| = 0.000101011\ldots_2$$

前导零计数 (LZC)：小数点后有 3 个零（位$-1, -2, -3$ 为 0），左移 3 位：

$$E_y = -3 \quad (\text{在指数域中意味着 } E_0 - 3)$$

$$M_y = 1.0101_2 \quad (\text{取 } w_F + 1 = 5 \text{ 位})$$

第六步：计算 $\sin\left(\frac{\pi}{4}y\right)$ 和 $\cos\left(\frac{\pi}{4}y\right)$

cos 通路（定点）：

输入 $Y = 0.000101_2$，HOTBM 计算 $f_{\cos}(Y) = 1 - \cos\left(\frac{\pi}{4}Y\right)$：

$$\cos\left(\frac{\pi \times 0.0845}{4}\right) = \cos(0.0663) \approx 0.99780$$

$$f_{\cos} = 1 - 0.99780 = 0.00220 \quad (6\text{位定点中非常小})$$

> 注：这里的 $f_{\cos} \approx 0.00220$ 采用实数近似用于说明流程；若定点只有 $w_F + g = 6$ 个小

> 数位，则最小量级为 $2^{-6} \approx 0.0156$，该值会被量化为 0。实际设计中需使用更高位宽/保护位以覆盖这类小量。

$$\cos\left(\frac{\pi}{4}y\right) = 1 - f_{\cos} \approx 0.99780$$

sin 通路（浮点 × 定点）：

HOTBM 计算 $f_{\sin}(Y) = \frac{\pi}{4} - \frac{\sin\left(\frac{\pi}{4}Y\right)}{Y}$:

$$\frac{\sin(0.0663)}{0.0845} \approx \frac{0.0662}{0.0845} \approx 0.7834$$

$$f_{\sin} = \frac{\pi}{4} - 0.7834 = 0.7854 - 0.7834 = 0.0020$$

> 注：同理，$f_{\sin} \approx 0.0020$ 为实数近似展示；在极低位宽示例下它无法被 6 位小数定点精确表达。

$$\sin\left(\frac{\pi}{4}y\right) = M_y \times 2^{E_y} \times \left(\frac{\pi}{4} - f_{\sin}\right) = 1.0101_2 \times 2^{-3} \times 0.7834 \approx 0.0845 \times 0.7834 \approx 0.0662$$

验证：$\sin(0.0663) \approx 0.0662$ ✓

第七步：象限重构

$k \bmod 4 = 0$，$S_x = 0$（x 为正）

查表（$k \bmod 4 = 0$）：

$$\sin(x) = +\sin\left(\frac{\pi}{4}y\right), \quad \cos(x) = +\cos\left(\frac{\pi}{4}y\right)$$

但注意 y 为负数，利用 $\sin(-|y|) = -\sin(|y|)$，$\cos(-|y|) = \cos(|y|)$:

$$\sin\left(\frac{\pi}{4}y\right) = -0.0662, \quad \cos\left(\frac{\pi}{4}y\right) = 0.9978$$

最终结果:

$$\sin(12.5) \approx -0.0662, \quad \cos(12.5) \approx 0.9978$$

验证:

$$\sin(12.5) \approx -0.0663 \quad \checkmark$$

$$\cos(12.5) \approx 0.99780 \quad \checkmark$$

全流程一图总结

```
x = 1.1001 × 2^3 = 12.5
          │
          ├── E = 3
          │      │
          │      ▼
          │   4/π 预存常量 ──→ 移位器 ──→ 窗口 C (17位)
          │                                │
          ├── 1.Fx = 1.1001 (5位)          │
          │          │                     │
          │          ▼                     ▼
          │        ┌──────────────────────────┐
          │        │    定点乘法 (5×17位)      │
          │        └────────────┬─────────────┘
          │                     │
          │               乘积 P (22位)
          │                     │
          │          ┌──────────┼───────────┐
          │          │          │           │
          │       污染区       k=0      y=-0.0845
          │       (丢弃)      (3位)    (10位定点)
          │                   │             │
          │                   │      ┌──────┴──────┐
          │                   │      │             │
          │                   │    定点 Y      浮点(Ey,My)
          │                   │  =0.000101   =1.0101×2^{-3}
          │                   │      │             │
          │                   │      ▼             ▼
          │                   │   ┌─────┐    ┌──────────┐
          │                   │   │HOTBM│    │  HOTBM   │
          │                   │   │f_cos│    │  f_sin   │
          │                   │   └──┬──┘    └────┬─────┘
          │                   │      │            │
          │                   │  cos=0.9978   sin=0.0662
          │                   │      │            │
          │                   ▼      ▼            ▼
          │              ┌─────────────────────────────┐
          ├── Sx=0 ─────→│          象限重构            │
          │              │  k mod 4 = 0                │
          │              │  sin(12.5) = -sin = -0.0662 │
          │              │  cos(12.5) = +cos = 0.9978  │
```

```
         │              └──────────────┬──────────────┘
         │                             │
         ▼                             ▼
    sin(12.5) ≈ -0.0663         cos(12.5) ≈ 0.9978  ✓
```