

# 基于 GPU 的 NCO 相位累积误差消除方法\*

陈永强<sup>1</sup>, 万张云虹<sup>2</sup>

(1 航天工程大学电子与光学工程系 北京 101416

2 四川大学商学院 成都 610044)

**摘要:** 针对航天测控系统高精度数控振荡器 NCO 实现需求, 利用图形处理单元 GPU 的高灵活性和高效并行数据处理能力, 设计了一种基于 GPU 的高效高精度 NCO 实现方法。针对 NCO 计算中浮点数相位累加运算累积误差大的共性问题, 利用无误差变换技术设计了基于 Fast2Sum 算法和 2Sum 算法的单精度浮点数相位累积误差综合补偿算法, 并利用该算法为 NCO 系统设计了相位累积求和算子。最后, 基于 GPU 平台对该方法进行了验证。试验结果表明, 该方法能够将基于 GPU 的 NCO 系统浮点数相位累积误差控制在  $1 \times 10^{-5}$  rad 量级。

**关键词:** GPU; 数字下变频; 数控振荡器; 累积误差; 无误差变换

**中图分类号:** TN75      **文献标识码:** A      **文章编号:** CN11-1780(2021)06-0047-10

**DOI:** 10.12347/j.ycyk.20200924001

**引用格式:** 陈永强, 万张云虹. 基于 GPU 的 NCO 相位累积误差消除方法[J]. 遥测遥控, 2021, 42(6): 47-56.

## A new method for NCO phase accumulation error elimination using GPU

CHEN Yongqiang<sup>1</sup>, WAN-ZHANG Yunhong<sup>2</sup>

(1. Department of Electronic and Optical Engineering, Space Engineering University, Beijing 101416, China;

2. Business School, Sichuan University, Chengdu 610044, China)

**Abstract:** Aiming at the high precision NCO realization demand of aerospace TT&C system, a high precision NCO realization method based on GPU is designed by utilizing the high flexibility and high efficiency parallel data processing capability of GPU. Aiming at the common problem of large cumulative error in the phase accumulation operation of floating point numbers in NCO calculation, a single precision phase accumulation comprehensive compensation algorithm based on Fast2Sum algorithm and 2Sum algorithm is designed by using the Error-Free transformation technology, and a phase accumulation summation operator is designed for NCO system. Finally, the design method is verified based on GPU platform. The experimental results show that the design method can control the phase accumulation error of GPU-based NCO system at  $1 \times 10^{-5}$  rad.

**Key words:** GPU; Digital down converter; NCO; Accumulative error; Error-Free transformation

**DOI:** 10.12347/j.ycyk.20200924001

**Citation:** CHEN Yongqiang, WAN-ZHANG Yunhong. A new method for NCO phase accumulation error elimination using GPU[J]. Journal of Telemetry, Tracking and Command, 2021, 42(6): 47-56.

## 引 言

数字下变频 DDC (Digital Down Converter) 系统是现代航天测控系统的重要子系统, 而数字本振是 DDC 系统最复杂的核心器件。在传统的 DDC 系统中, 数字本振主要由数控振荡器 NCO (Numerical Control Oscillator) 实现<sup>[1]</sup>。受运算器件片上资源限制, 到目前为止, 高精度相位分辨率的获得依然需要在运算资源与存储空间之间进行折中。图形处理单元 GPU (Graphics Processing Unit) 高效的浮点运算能力和多层次存储体系为数字本振的实现提供了高效高精度实现方案<sup>[2,3]</sup>。计算统一设备架构 CUDA

\*基金项目: 北京市科技重大专项 (Z181100002918004)

收稿日期: 2020-09-24 收修改稿日期: 2021-09-27

(Compute Unified Device Architecture) 提供了高效的数学函数库, 可为 NCO 的直接计算提供基础, 同时, GPU 共享内存和纹理内存为查找表的实现提供了高效的解决途径<sup>[4,5]</sup>。文献[6]和文献[7]也为本文研究提供了可供借鉴的研究思路。

然而本文在研究中发现, 基于 GPU 的 NCO 实现方法虽然具有灵活高效的优势, 但浮点数在加法运算时采取的对阶和规格化操作会给运算结果带来舍入误差, 而 NCO 相位累加过程则会因为舍入误差的存在而出现误差累积<sup>[8]</sup>。尤其是当相位累加的点数较多时, 误差累积将变得显著。无误差变换 (Error-Free transformation) 技术<sup>[9]</sup>基于“总和补偿法 (Compensated sum method)”<sup>[8]</sup>的思想补偿浮点数运算误差, 可有效提高运算精度。浮点数加法的经典无误差变换算法主要为 Dekker 提出的 Fast2Sum 算法和 Knuth 提出的 2Sum 算法, 目前这两个算法依然在数值计算领域有着广泛的应用<sup>[10]</sup>。文献[8]对无误差变换技术做了系统的总结, 并给出了浮点数加、减、乘、除运算的无误差变换实现方法。文献[11]分析了 Fast2Sum 和 2Sum 算法的优劣, 并给出了以上算法的适用条件。

本文首先分析了 DDC 系统的数字 NCO 实现原理, 然后设计了基于 GPU 的 NCO 实现方法并进行了优化, 利用无误差变换技术, 设计了基于 CUDA 的浮点数相位累积误差综合补偿算法, 最后基于 GPU 平台对本文所提方法进行了实验验证。

## 1 数字 NCO 实现原理

在实时信号处理过程中, 为了实现输入数据的实时下变频, 需要产生与输入数据匹配的 NCO 信号。典型的 NCO 实现过程如图 1 所示。

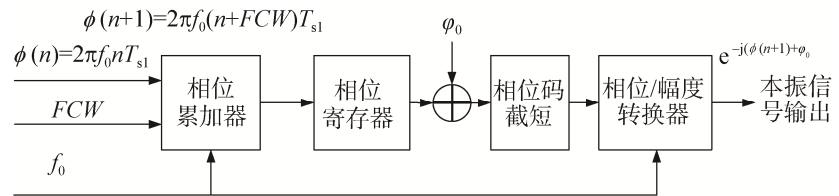


图 1 数字 NCO 生成过程流程图

Fig. 1 Flow chart of digital NCO generation process

图 1 中, 数字角频率  $\omega_0 = 2\pi f_0 / f_s$ ,  $f_s$  为时钟频率,  $f_0$  为 NCO 输出频率,  $wd$  为频率控制字,  $T_{s1}$  为输入信号的采样周期。相位寄存器内的相位值经过数据位数截短后输入相位幅度转换器  $Tr[\cdot]$ , 最终输出标准正弦波, 即

$$Tr[\phi_{\text{new}}(n+1)] = \sin(2\pi f_0(n+wd)T_{s1} + \phi_0) \quad (1)$$

当前, 在软件无线电系统中, 为了达到运算效率与运算精度的平衡, 实现 NCO 相位幅度转换的常用方法主要有两种, 一种是查找表法, 另一种是实时计算法<sup>[1]</sup>。查找表法优点是执行效率高、延迟低、占用运算资源少, 但缺点是存储位宽和深度受到存储空间的限制, 幅度值存储的有限字长效应将导致输出信号杂散增加。另外在计算出相位值后, 为了将相位值的位数与相位查找表的相位地址码位数匹配, 需要对相位输出值进行截断操作, 这将直接给输出信号带来截断杂散干扰。实时计算法是为了缓解运算器件片上存储空间不足而设计的替代方法, 该方法的突出优势是节省了紧张的存储资源, 但需要消耗额外的运算资源产生 NCO 信号, 这又增加了运算成本, 对运算效率提出了挑战。为了克服以上困难, 本文根据基带转换系统实际需求, 同时利用 GPU 高效的浮点数运算能力和充足的存储空间设计了基于 CUDA 的 NCO 并行实现方法。

## 2 基于 CUDA 的数字 NCO 并行实现方法

### 2.1 数字 NCO 实现方法设计

由数字本振的计算式 (1) 可知, 在 CUDA 架构下, 可以使用单精度或者双精度浮点数直接计算标准正弦波的输出。基于 GPU 的数字 NCO 实时生成过程如图 2 所示。

图 2 中, GPU 线程块取一维, Block 大小为 256, Grid 大小为 512。线程网格循环加载, 保证每个线程处理一个 NCO 数据的相位计算、相位累加和三角函数计算。最终得到了数字 NCO 的高效并行实现。

设需要处理的输入数据总长度为  $N$ , 考虑实时数据处理需求, 对输入数据做分段处理, 每段数据长度  $nL$ 。设本次处理的数据为第  $iN$  ( $i=0,1,2,\dots$ ) 段, 本段内每个数据点段内索引为  $iL$ , 则该点在整个数据序

列中的索引值为  $n=iN \times nL+iL$ 。设 NCO 初始相位为  $\varphi_0$ ，参与运算的线程块 (Block) 数量为  $nBPG$ ，每个 Block 内的线程 (Thread) 数量为  $nTPB$ 。数据处理时，设第  $n$  点数据对应的 NCO 相位为  $\phi(n)$ ，输入数据采样率为  $f_s$ ，本振频率为  $f_0$ ，由此可得第  $n$  点对应的相位为  $\phi(n) = 2\pi f_0 n / f_s$ ，初始相位加法器输出为  $\phi(n) = 2\pi f_0 n / f_s + \varphi_0$ ，根据此相位值，线程  $n$  可直接按照式 (1) 计算并输出 NCO 的 I、Q 两路数据值。

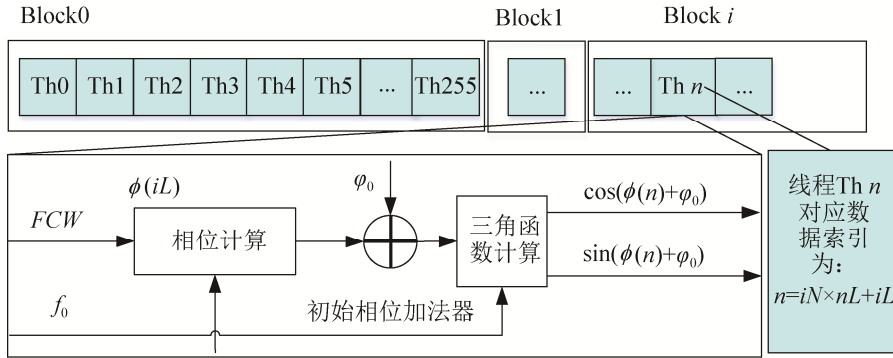


图 2 实时浮点运算 NCO 生成过程流程框图

Fig. 2 Flow chart of NCO generation process of real-time floating-point operation

### 2.2 算法优化与结果分析

图 2 所示的 NCO 实现方法简单高效，能够实现数字 NCO 的并行实时输出，但相位累加过程会带来舍入误差，特别是在相加的两个数值差距较大时，误差将更加显著。同时，在进行数据连续处理时，误差会随着相位累加运算而出现累积。在 1 024 MSPS 采样率下，当按照 10 ms 数据分段法用单精度浮点数处理 1 s 数据时，相位误差的累积将达到无法忍受的程度。为了说明此问题，采用如下参数对该过程进行仿真分析。输入信号采样率为 1 024 MSPS，NCO 频率为 249 MHz，分析数据长度为  $2^{20}$ ，分别采用单精度和双精度 NCO 数据对 NCO 相位、NCO 幅度和混频后输出数据幅度进行分析对比，结果如图 3 所示。

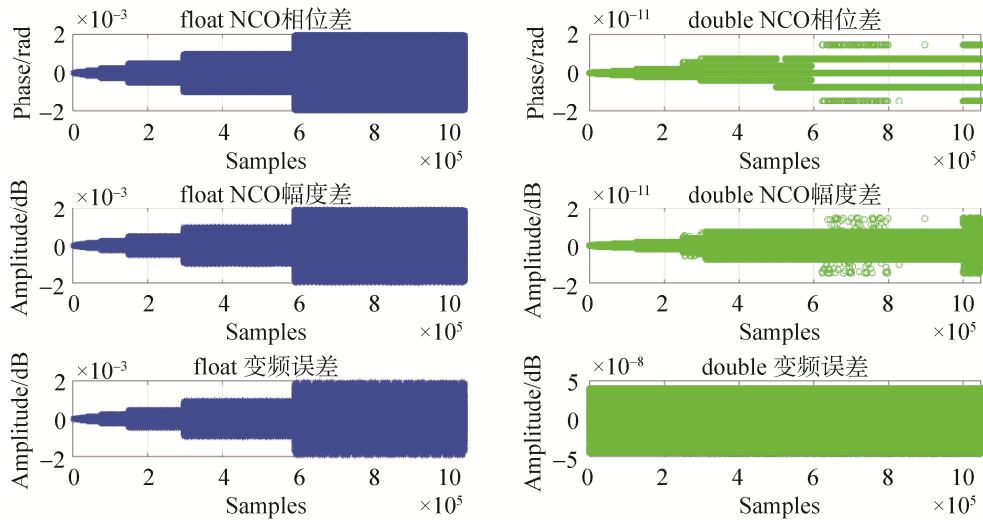


图 3 数字 NCO 变频过程误差累积仿真

Fig. 3 Error accumulation simulation of digital NCO frequency conversion process

图 3 中，第一列为采用单精度 NCO 计算结果，第二列为采用双精度 NCO 计算结果；第一行为 NCO 累积相位与理论相位偏差，第二行为 NCO 输出信号幅度与理论 NCO 幅度偏差，第三行为 NCO 变频结果与理论变频结果之间的偏差。从仿真结果可看出：

① 单精度模式误差累积效应较双精度模式明显差了好多个数量级，且在仿真数据范围内各项偏差较大；而双精度模式下 NCO 相位和幅度偏差优于  $1 \times 10^{-10}$ ，变频结果偏差也优于  $1 \times 10^{-8}$  且较为稳定，具有较高的精度。

② 纵向来看, NCO 幅度偏差和变频结果偏差均与累积相位偏差一致, 可见累积相位偏差是造成各环节误差的主要因素。

③ 采用逐点计算并扣除误差可降低累积误差, 但会大量消耗计算资源, 导致时延增加, 影响系统实时性。另一个有效的方法是采用双精度浮点数计算相位, 此法简单方便, 但会带来运算效率的损失。

为解决此问题, 本文设计了数据分段模式下相位整周模糊循环消除方法, 该方法核心思想是: 第一, 数据分段处理; 第二, 在处理过程中确保每个分段内单独的相位按照索引计算, 避免累加运算; 第三, 分段数据间 NCO 相位累加采用整周模糊循环消除方法, 将参与相加的数据限制在  $2\pi$  范围内, 确保每次相加的两个浮点数量级相当, 减小舍入误差。算法流程为:

① 对输入的实时数据做分段处理, 假设分为  $M$  段, 每一段数据长度为  $N$ , 为描述方便, 每一段数据段内索引取  $n=1, 2, \dots, N-1, N$ ;

② 逐点高精度计算第一段每个点的相位值得  $\varphi(n) = 2\pi f_0 n / f_s$ ;

③ 在第一段数据处理完后, 将末位本振相位做整周模糊消除操作, 即

$$\varphi(N) = 2\pi f_0 N / f_s - [2\pi f_0 N / f_s] \quad (2)$$

式中  $[\cdot]$  表示向下取整。

④ 计算下一段本振信号初始相位, 设每一采样点相位差值  $\Delta\varphi = 2\pi f_0 / f_s$ , 则第二段信号初始相位为

$$\varphi(N+1) = \varphi(N) + \Delta\varphi \quad (3)$$

由此计算第二段信号相位为

$$\varphi(N+n) = \varphi(N) + n\Delta\varphi = \varphi(N) + 2\pi f_0 n / f_s \quad (4)$$

⑤ 以此类推, 第  $i(i=1, 2, \dots, M)$  段信号相位为

$$\varphi((i-1)N+n) = \varphi((i-1)N) + n\Delta\varphi = \varphi((i-1)N) + 2\pi f_0 n / f_s \quad (5)$$

该方法在每一段数据结束时进行相位整周模糊消除操作, 避免了逐点判断对资源的过度消耗, 同时在每一段初始相位计算累加时保证相位值在一个整周内, 减小了误差累积。下面用仿真数据对算法进行测试, 信号参数设置与上文相同, 分析数据长度为  $2^{20}$ , 数据均匀分段, 每段长度为  $2^{16}$ , 分别采用单精度和双精度 NCO 数据对 NCO 相位、NCO 幅度和采用 NCO 混频后输出数据幅度进行分析对比, 计算结果如图 4 所示。

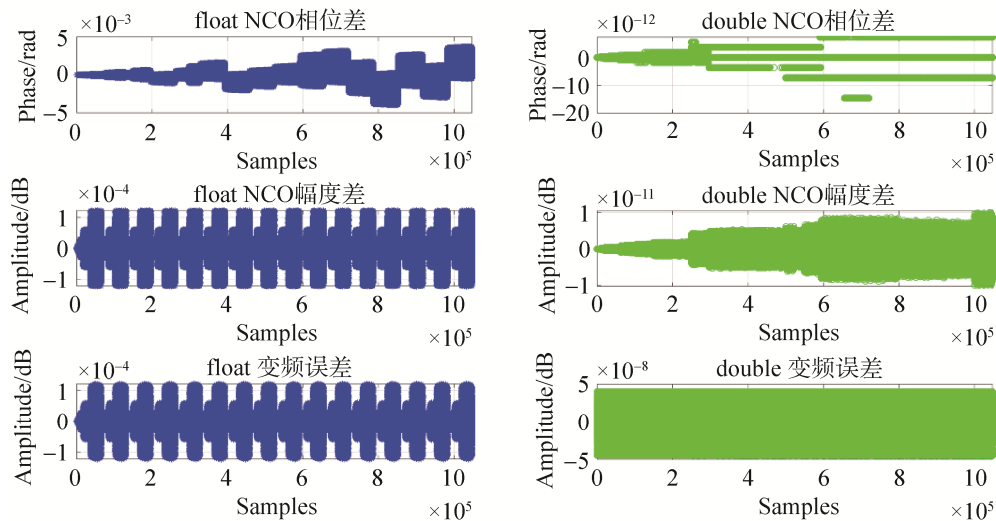


图 4 数字 NCO 变频过程优化算法误差累积仿真

Fig. 4 Error accumulation simulation of digital NCO frequency conversion process optimization algorithm

从仿真结果可以看出:

① 优化算法对单精度模式下运算偏差影响较大, 幅度偏差整体减小了 1 个数量级且误差被压缩在一个较小的范围内;



② 优化算法对双精度模式运算偏差影响较小，从优化前后的对比可看出，双精度模式优化结果略优于优化前偏差；

③ 从单精度仿真结果可以看出，通过分段整周模糊循环消除方法，一次处理较少的点数，有效压缩了浮点运算的误差累积。因此，可通过改变分段数据长度，进一步压缩累积误差。图 5 显示了分段长度为  $2^{10}$  时的运算结果，单精度浮点数 NCO 幅度误差和变频总误差均由  $1 \times 10^{-4}$  降低为  $1 \times 10^{-6}$ ，运算误差大大降低。

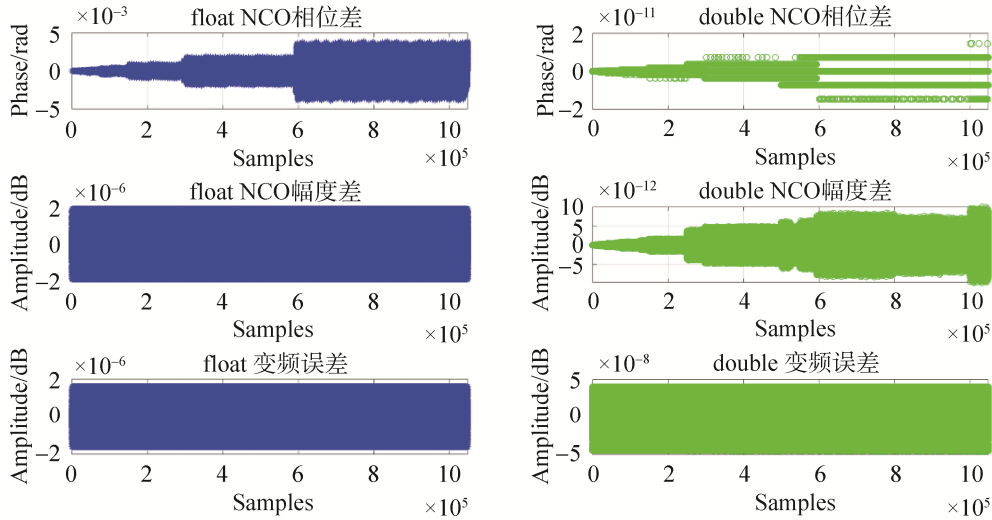


图 5 数字 NCO 变频过程优化算法仿真验证

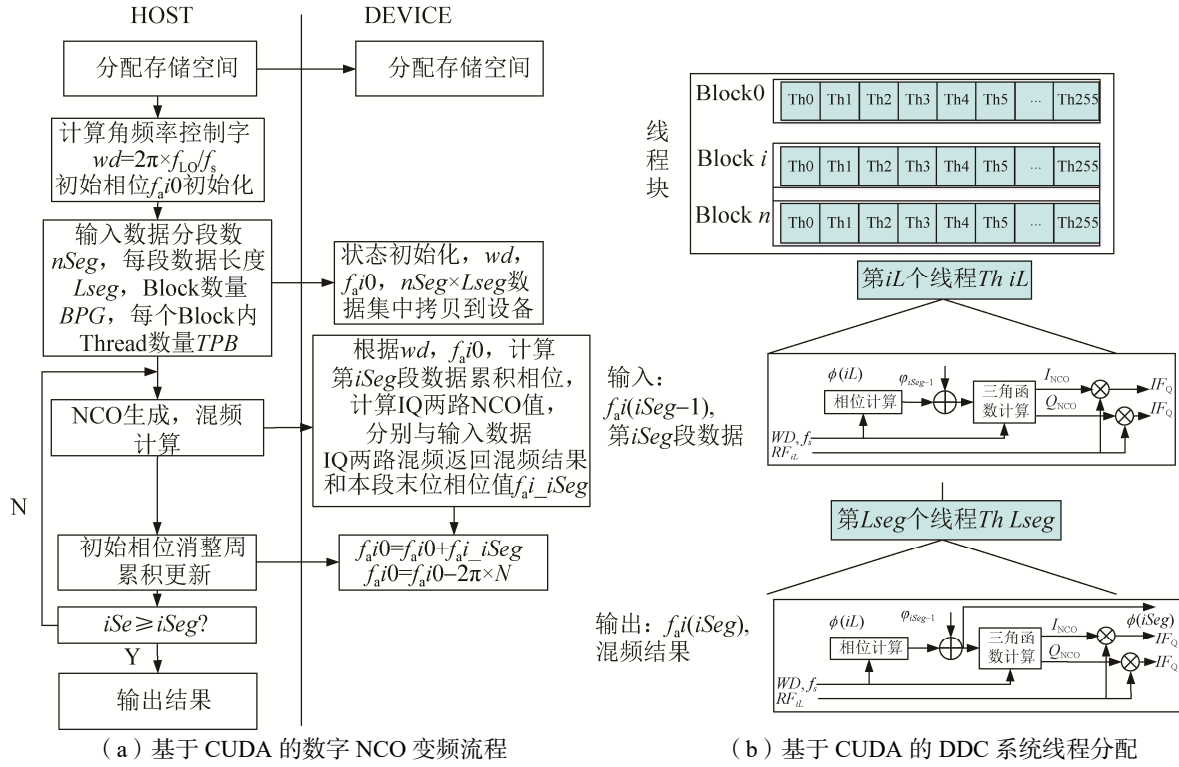
Fig. 5 Simulation verification of digital NCO frequency conversion process optimization algorithm

基于以上仿真分析结果，选择单精度模式进行 NCO 运算，在 CUDA 架构下，优化算法 NCO 的混频运算流程如图 6 (a) 所示。图中，输入数据被分成  $nSeg$  段，每段长度  $Lseg$ ，段内索引  $iL$ ，输入数据采样率为  $f_s$ ，本振频率为  $f_{LO}$ ，由此可得本振角频率控制字  $wd = 2\pi f_{LO} / f_s$ ，本振初相为  $f_{ai}0$ 。状态初始化阶段，数据分段、角频率控制字计算、数据拷贝过程全部完成。运算阶段，设备端在主机的控制下，调用核函数逐段数据计算累积相位、NCO 值、混频结果和末尾相位，完成正交混频过程，每一段数据处理完成后，利用末尾相位对下一段初相累积更新并循环消除整周模糊。参与运算的线程块设置为 1 维，Block 大小为  $TPB$ ，Grid 大小为  $BPG$ ，线程网格循环加载完成所有数据的处理。线程块的分配和每个线程完成的操作如图 6 (b) 所示。根据每段数据的长度开启足够的线程数，保证每一个线程用于处理一个输入数据点的 NCO 实现和正交混频。

需要注意的是，基于 GPU 的 NCO 浮点数直接计算实现方法最大的优势在于，其相位分辨率可以不受相位累加器位数的影响，输出频率的设置更加灵活可控，不用固定于特定的频点。然而，浮点数运算舍入误差将造成运算存在一定偏差，且长序列相位累加运算将导致误差累积，这些误差将给 NCO 的高精度实现造成不利影响。为克服浮点数误差累积需要尽可能短的数据分段，然而过多的数据分段将会影响运算效率。因此，分段的长度需要考虑基带转换器系统运算精度和运算效率约束。

### 3 基于无误差变换技术的相位累积误差补偿方法研究

在基于 GPU 的 NCO 实现中，以最近点舍入为基础的浮点数累加过程将导致相位误差的累积，且随着相位累加点数的增加，最终可能导致输出数据完全错误<sup>[12]</sup>。采用双精度浮点数虽然能够缓解累积误差，但双精度运算将给系统运算效率带来不利影响<sup>[13]</sup>。上一节采用整周模糊循环消除方法有效压缩了变频偏差，且将累积误差压缩在了一个固定范围内，但相位误差优化效果不明显。为了解决以上问题，本文基于无误差变换技术，提出一种单精度浮点数相位累积误差综合补偿方法。本节首先介绍两种经典的两数求和的无误差变换方法，然后在此基础上设计适用于 NCO 系统的相位累计误差补偿方法。



(a) Digital NCO frequency conversion process based on CUDA (b) Thread allocation of DDC system based on CUDA

图 6 基于 CUDA 的数字 NCO 变频流程图

Fig. 6 Digital NCO frequency conversion flow chart based on CUDA

### 3.1 无误差变换技术

无误差变换技术是高精度数值计算领域的重要技术。根据本文需求, 下面重点对浮点数和的无误差变换算法做简要介绍。

#### ① Fast2Sum 算法

Fast2Sum 算法是计算两个数之和的无误差变换经典算法, 其核心计算步骤为<sup>[14]</sup>:

$$\begin{aligned}
 & \text{Fast2Sum}(a, b): \\
 & s = \text{RN}(a + b) \\
 & z = \text{RN}(s - a) \quad , \text{ if } (|a| \geq |b|) \\
 & t = \text{RN}(b - z)
 \end{aligned} \tag{6}$$

式(6)中,  $a$ 、 $b$  为输入的两个加数, RN 表示按照浮点数所能表示的精度就近舍入。由此可得出浮点数相加的和  $s$  以及计算误差  $t$ 。在两个数大小确定时, Fast2Sum 算法只用三次加法运算就可完成误差的补偿, 简洁高效; 但在两个加数大小不确定时, 两次绝对值运算和分支判断将消耗大量的运算资源, 不利于运算效率的提高<sup>[8]</sup>。

#### ② 2Sum 算法<sup>[15]</sup>

为了克服 Fast2Sum 的不足, Knuth 提出了改进的 2Sum 算法, 该算法计算步骤为:

$$\begin{aligned}
 & f(s, t) = 2\text{Sum}(a, b): \\
 & s = \text{RN}(a + b) \\
 & b' = \text{RN}(s - a) \\
 & a' = \text{RN}(s - b') \\
 & da = \text{RN}(a - a') \\
 & db = \text{RN}(b - b') \\
 & t = \text{RN}(da - db)
 \end{aligned} \tag{7}$$

与 Fast2Sum 算法相比, 2Sum 算法的 6 次加法操作更加复杂。但在两个加数大小不明确时, 2Sum 算法的加法运算将比 Fast2Sum 算法的判断过程更加高效。

Boldo 在文献[11]中指出的, 作为数值计算算法的重要组成部分, Fast2Sum 算法和 2Sum 算法即使在舍入误差较大的条件下, 其返回结果也是有意义的, 而且几乎不会溢出, 也证明了这两个算法广泛的应用前景和更好的鲁棒性。同时文献[8]也指出, 2Sum 算法适用范围广, 是目前计算两个数和的无误差变换算法中的最优选项。综合以上分析, 本文将在两个算法的基础上, 设计提高 NCO 系统单精度浮点数相位累积运算精度的方法。

### 3.2 基于 Fast2Sum 算法的 NCO 相位累积误差补偿方法

#### 3.2.1 补偿方法推导

Fast2Sum 算法由于引入的新的运算量少, 在两个加数大小较为明确的累加运算中将具有较好的效率优势。因此, 本文首先考虑基于 Fast2Sum 算法的 NCO 相位浮点运算累积优化算法。

根据式(6)可得, 基于 Fast2Sum 算法的累加算法可表示为

$$\begin{aligned} & \text{for}(i = 0 : \text{length}(\text{input})) \\ & \quad y = \text{RN}(\text{input}[i] + \text{err}) \\ & \quad t = \text{RN}(\text{sum} + y) \\ & \quad \text{err} = \text{RN}(y - (t - \text{sum})) \\ & \quad \text{sum} = t \end{aligned} \quad (8)$$

式(8)中, 输入数据为  $\text{input}$ , 每一步累积偏差和下一步输入数据修正值为  $\text{err}$ , 实际得到的累加结果为  $t$ , 累加输出为  $\text{sum}$ 。通过以上运算, 可将每一步浮点数累加运算偏差在下一次运算时进行补偿。基于式(8)中的算法对 NCO 生成过程的相位累加运算进行修正, 可得 NCO 实现流程如图 7 所示。

从图 7 可知, 新的 NCO 生成算法是在数据分段的基础上, 将相位累积算子更新为新的 Fast2Sum 累积求和算子。在该算子中, 设本次数据分段索引为  $i\text{Seg}$ , 第  $i\text{Seg}-1$  段末位相位为  $\phi_{i\text{Seg}-1}$ , 求和偏差为  $\text{err}$ , 则第  $i\text{Seg}$  段索引为  $iL$  的数据修正后的输入相位为

$$\phi'(iL) = \phi(iL) + \text{err} \quad (9)$$

输出相位为

$$\phi^{\text{out}}(iL) = \phi'(iL) + \phi_{i\text{Seg}-1} \quad (10)$$

本次浮点数运算累积误差为

$$\text{err} = \phi'(iL) - (\phi^{\text{out}}(iL) - \phi_{i\text{Seg}-1}) \quad (11)$$

同时将第  $i\text{Seg}$  段数据末位相位更新为  $\phi_{i\text{Seg}}$  并进入下一段数据处理。

通过以上过程, 即完成了对相位累积误差的综合补偿。在 CUDA 平台计算时, 只需要在原来分段计算过程的基础上, 增加相位偏差补偿核函数, 即可完成相位累加过程的优化。

#### 3.2.2 算法性能仿真分析

利用以上算法对输出相位进行仿真分析, 仿真参数为 NCO 频率为 9 MHz, 采样率  $f_s$  为 64 MHz, 分

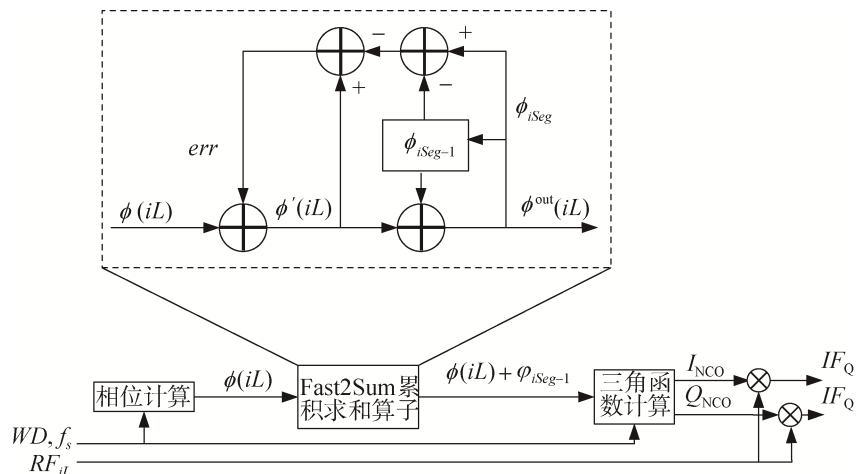


图 7 基于 Fast2Sum 算法的 NCO 实现流程图

Fig. 7 Flow chat of NCO implementation based on Fast2Sum algorithm

析数据长度为  $2^{15}$ , 共分为 32 段, 每段 1 024 点。分别采用单精度、补偿单精度和双精度 NCO 运算数据对 NCO 相位进行分析对比。仿真过程采用累积方法计算相位, 每 1 024 点对相位做循环复位消去整周部分, 累加计算采用 Fast2Sum 算法消除浮点数计算偏差, 最终得到双精度运算、单精度运算和单精度 Fast2Sum 优化算法运算结果与理论值偏差, 如图 8 所示。图中, float 代表单精度、Fast2Sum 代表 Fast2Sum 优化、double 代表双精度数据与理论数据偏差。

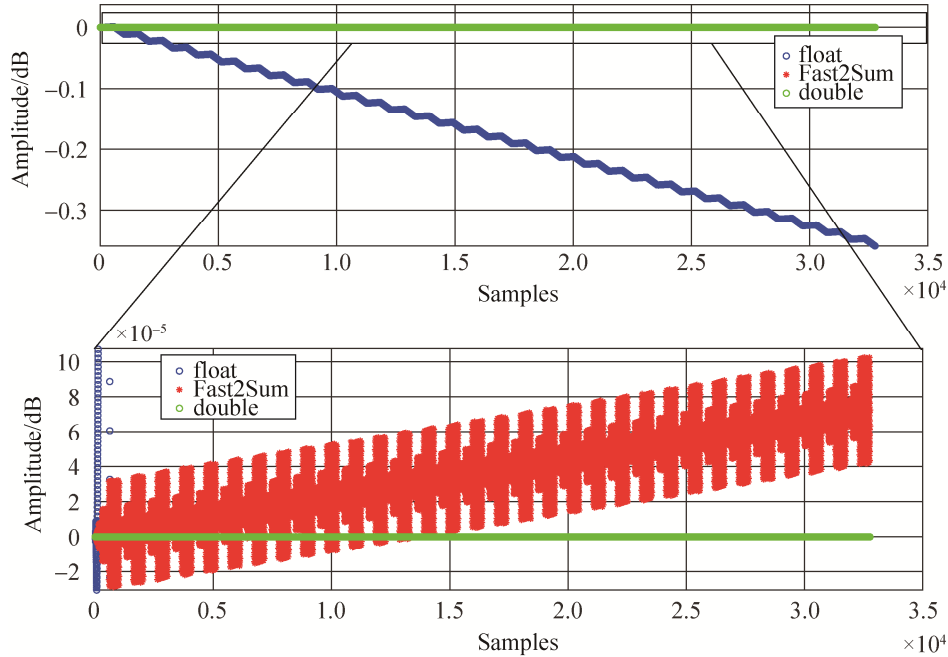


图 8 基于 Fast2Sum 算法的相位累积运算结果与理论值偏差

Fig. 8 The phase accumulation operation result of fast2sum algorithm deviates from the theoretical value

由图 8 可知, 双精度运算精度最高; 单精度运算随着累加样点的增加, 偏差持续增大, 测试数据末尾相位偏差已经超过 0.3 rad, 这将给输出 NCO 频率造成偏差; 在单精度运算基础上叠加 Fast2Sum 算法后, 运算偏差被大幅度压缩, 然而图 8 也显示了在单精度运算基础上叠加 Fast2Sum 算法后相位偏差依然残留一个明显的漂移趋势, 相位漂移率达到 0.141 58 rad/s, 在实验条件下这个趋势将给 NCO 输出频率引入约 0.022 5 Hz 偏差。扣除这一趋势项后, 相位偏差优于  $1 \times 10^{-5}$  rad。

### 3.3 基于 2Sum 算法的 NCO 相位累积误差补偿方法

#### 3.3.1 补偿方法推导

为了克服 Fast2Sum 算法分支判断和绝对值运算效率低的问题, 本文引入 2Sum 算法进行累积相位计算。根据式 (7) 可得, 基于 2Sum 算法的累加算法可表示为

$$\begin{aligned}
 & \text{for}(i = 0 : \text{length}(\text{input})) \\
 & \quad a = \text{input}[i] \\
 & \quad b = \text{sum} \\
 & \quad a = \text{RN}(a + da) \\
 & \quad b = \text{RN}(b + db) \\
 & \quad \text{sum} = \text{RN}(a + b) \\
 & \quad b1 = \text{RN}(\text{sum} - a) \\
 & \quad a1 = \text{RN}(\text{sum} - b1) \\
 & \quad db = \text{RN}(b - b1) \\
 & \quad da = \text{RN}(a - a1)
 \end{aligned} \tag{12}$$



式 (12) 中, 输入数据为  $input$ , 两个加数  $a$ 、 $b$  每一步求和偏差和下一步输入数据修正值分别为  $da$ 、 $db$ , 实际得到的累加输出为  $sum$ 。通过以上运算, 可将参与运算的两个数的每一步浮点数累加运算偏差在下次运算时进行补偿。基于式 (12) 对 NCO 系统相位累加运算误差进行修正, 可得 NCO 实现流程如图 9 所示。

从图 9 可知, 新的 NCO 生成算法是在数据分段的基础上, 将相位累加算子更新为新的 2Sum 累积和算子。在该算子中, 设本次数据分段索引为  $iSeg$ , 第  $iSeg-1$  段末位相位为  $\phi_{iSeg-1}$ , 求和偏差为  $da$ 、 $db$ , 则第  $iSeg$  段索引为  $iL$  的数据修正后的输入相位为

$$a = \phi_{iSeg-1}, a' = a + da \tag{13}$$

$$b = \phi(iL), b' = b + db$$

输出相位为

$$\phi^{out}(iL) = a' + b' \tag{14}$$

本次浮点数运算累积误差为

$$\begin{aligned} da &= a' - a1 = a' - (b' - b1) = a' - [b' - (\phi^{out}(iL) - a')] \\ db &= b' - b1 = b' - (\phi^{out}(iL) - a') \end{aligned} \tag{15}$$

同时, 将第  $iSeg$  段数据末位相位更新为  $\phi_{iSeg}$  并进入下一段数据处理。

通过以上过程, 即完成了对累积相位的综合补偿。在 CUDA 平台计算时, 只需要在原来分段计算过程的基础上, 增加偏差补偿核函数, 即可完成相位累加过程的优化。

### 3.3.2 算法性能仿真验证

利用以上算法对输出相位进行仿真分析, 仿真参数与 Fast2Sum 算法中取值相同。分别采用单精度、补偿单精度和双精度 NCO 运算数据对 NCO 相位进行分析对比。仿真过程采用累积方法计算相位, 每 1 024 点对相位做循环复位消去整周部分, 累加计算采用 2Sum 算法消除浮点数计算偏差, 最终得到双精度运算、单精度运算和单精度 2Sum 算法优化运算结果与理论值偏差, 如图 10 所示。图中 float、2Sum、double 分别代表单精度、2Sum 优化、双精度数据与理论数据偏差。由图 10 可知:

- ① 在单精度运算基础上叠加 2Sum 算法后, 每次累加运算的两个加数均得到补偿, 因此该算法无相位漂移;
- ② 采用 2Sum 算法后, 运算偏差被大幅度压缩, 相位偏差优于  $1 \times 10^{-5}$  rad;
- ③ 在相位整周模糊循环消除背景下, 参与相加的两个加数大小不定, 与 Fast2Sum 算法相比, 采用 2Sum 算法将避免绝对值运算和分支判断, 其运算效率更高。

## 4 结束语

本文首先推导了 DDC 系统实现原理, 重点分析了基于 GPU 的 DDC 系统中 NCO 的实现方法。针对 NCO 计算中浮点数相位累加运算累积误差大的问题, 利用无误差变换技术, 设计了基于相位整周模糊循

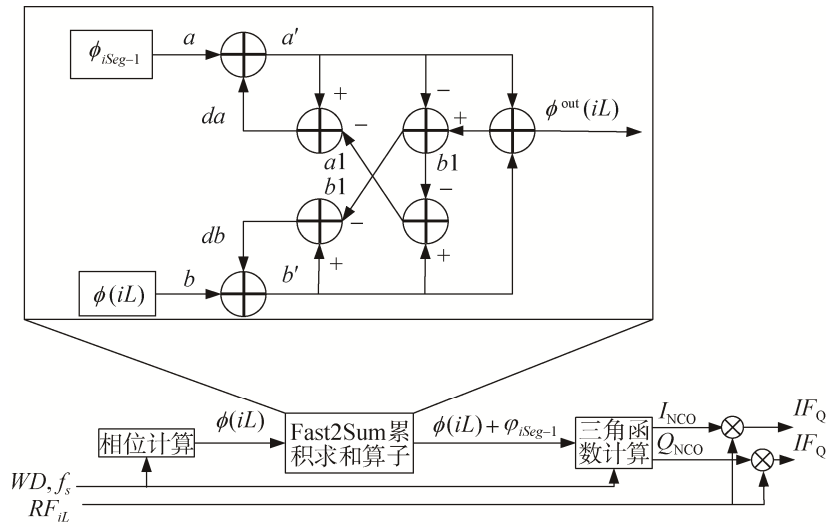


图 9 基于 2Sum 算法的 NCO 实现流程图

Fig. 9 Implementation flow chat of NCO based on 2sum algorithm

Fig. 9 Implementation flow chat of NCO based on 2sum algorithm

环消除+相位累积误差综合补偿算法, 利用该算法将浮点数相位累积误差控制在  $1 \times 10^{-5}$  量级, 有效提高了相位计算精度。同时, 利用 GPU 对本文设计的方法进行了仿真验证, 实验结果显示, 在基于 GPU 的 NCO 实现中, 采用相位循环消整周+2Sum 累积误差补偿方法, 将能够完成数字 NCO 的高效高精度实现。

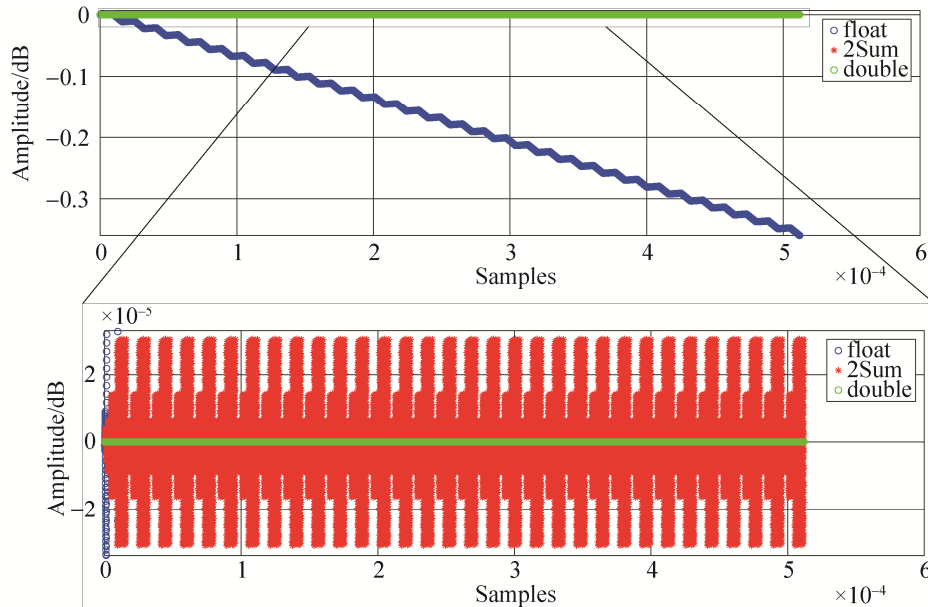


图 10 基于 2Sum 算法的相位累积运算结果与理论值偏差

Fig. 10 Deviation between phase accumulation operation result and theoretical value based on 2sum algorithm

### 参考文献

- [1] 楼才义, 徐建良, 杨小牛. 软件无线电原理与应用[M]. 电子工业出版社, 2014.
- [2] NVIDIA. CUDA11\_occupancy\_calculator[EB/OL]. <https://docs.nvidia.com/cuda/cuda-occupancy-calculator/index.html>. 2020-8-4
- [3] NVIDIA. CUDA C++ programming guide[EB/OL]. <https://docs.nvidia.com/cuda/pdf/CUDA-C-Programming-Guide.pdf>. 2020-8-4.
- [4] 苏华友. 面向应用的 GPU 并行计算关键技术研究[D]. 长沙: 国防科学技术大学, 2014.
- [5] 胡明. 基于 GPU 异构大规模超高精度浮点运算研究及其应用[D]. 重庆: 重庆邮电大学, 2013.
- [6] SHAO Guolin, CHEN Xingshu, YANG Lu. Research and implementation of a high performance parallel computing digital down converter on graphics processing unit [J]. Concurrency and Computation: Practice and Experience, 2017, 29(8): 40-42.
- [7] KIM S C, BHATTACHARYYA S S. A wideband front-end receiver implementation on GPUs[J]. IEEE Transactions on Signal Processing, 2016, 64(10): 2602-2612.
- [8] 姜浩. 高精度可靠浮点计算及舍入误差分析研究[D]. 长沙: 国防科学技术大学, 2013.
- [9] KAHAN W. Further remarks on reducing truncation errors [J]. Communications of the Acm, 1965, 8(1): 40.
- [10] KORNERUP P, LEFEVRE V, LOUVET N, et al. On the computation of correctly-rounded sums[C]// IEEE Symposium on Computer Arithmetic, 2009.
- [11] BOLDO S, GRAILLAT S, MULLER J. On the robustness of the 2Sum and Fast2Sum algorithms [J]. ACM Transactions on Mathematical Software (TOMS), 2017, 44(1): 1-14.
- [12] 张予器. 超高精度浮点运算的关键技术研究[D]. 长沙: 国防科学技术大学, 2005.
- [13] YOUNG E C. GPU-accelerated demodulation for a satellite ground station[Z]. ProQuest Dissertations Publishing, 2019.
- [14] DEKKER T J. A floating-point technique for extending the available precision [J]. Numerische Mathematik, 1971, 18(3): 224-242.
- [15] KNUTH D. The art of computer programming, volume 2 (3rd ed.): Seminumerical algorithms[M]. MA, USA: Addison-Wesley, 1997.

### [作者简介]

陈永强 1988 年生, 硕士, 工程师, 主要研究方向为航天测控通信系统。  
万张云虹 1998 年生, 大学在读。