

基于 Parallel_CORDIC 的高精度高速度 直接数字频率合成器的 FPGA 实现

祁艳杰, 刘章发

(北京交通大学电子信息工程学院, 北京 100044)

摘要: 本文提出了一种直接数字频率合成器(DDFS)的设计,以 Parallel-CORDIC(Coordinate Rotation Digital Computer)算法模块替代传统的查找表方式,实现了相位与幅度的一一对应,输出相位完全正交的正余弦波形;同时应用旋转角度预测及 4:2 的进位保存加法器(CSA)技术,将速度比传统 CORDIC 算法提高 41.7%,精度提高到 10^{-4} .最后以 Xilinx 的 FPGA 硬件实现整个设计.

关键词: 直接数字频率合成技术(DDFS); Parallel CORDIC; 进位保存加法器(CSA); FPGA

中图分类号: TN911.23 **文献标识码:** A **文章编号:** 0372-2112(2014)07-1392-06

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2014.07.023

FPGA Implementation of High Speed and High Precision Direct Digital Frequency Synthesizer Based on Parallel_CORDIC

QI Yan-jie, LIU Zhang-fa

(School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: The design of a direct digital frequency synthesizer (DDFS) is proposed in this paper. Parallel_CORDIC (Coordinate Rotation Digital Computer) algorithm module is used to replace the traditional look-up table method, the phase and amplitude of a one-to-one correspondence is realized, and outputs are sine and cosine waveforms that completely orthogonal in phase. The angle prediction and 4:2 Carry Save Adder (CSA) technologies are also applied in the design, the calculation speed is 41.7% faster than the traditional CORDIC algorithm and the accuracy is improved to 10^{-4} . Finally, the whole design is implemented based on Xilinx FPGA development board.

Key words: direct digital frequency synthesis (DDFS); Parallel CORDIC; Carry Save Adder (CSA); FPGA

1 引言

直接数字频率合成技术(DDFS)是频率合成技术的一场革命,最早由 Joseph Tierney 等三人于 1971 年提出^[1],它具有快速转换、高频率分辨率、低相位噪声、易于切换、相位连续等优点.随着计算机的快速发展,DDFS 已经广泛应用于通信、电子仪器仪表等领域.传统的 DDFS 主要由查找表实现,它的主要组成包括相位累加器、sin/cos ROM 表、DA 转换及低通滤波器模块,结构图如图 1^[2].

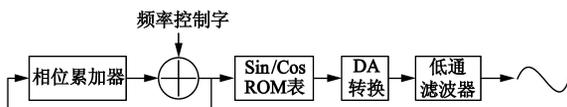


图1 传统DDFS的结构图

其中,频率控制字是每次相位累加的步长,控制输出频率大小.累加器的相位送入到 ROM 表,将相位与幅度一一对应后送给 DA 转换,最后滤波输出.传统的 DDFS 要实现高精度必然会占据很大的内存空间,在转换时,硬件开销过大,处理速度受到限制.近年来超大规模集成电路(VLSI)的发展使坐标旋转数字计算机(CORDIC)在数字信号处理中的优势得以展现出来.本文提出的以 Parallel CORDIC 算法为核心的 DDFS 成功避免了传统实现方式中的弊端.

2 CORDIC 算法基本原理

CORDIC 是由 Volder 在 1959 年提出的,这种算法可以应用在圆周坐标、线性坐标和双曲线坐标中.本文中主要应用的是圆周坐标下的 CORDIC,其基本思想是在

圆周坐标下, 向量旋转, 直到到达目标角度^[3]. CORDIC 的基本公式^[4]为:

$$\begin{cases} x_{i+1} = x_i - \delta_i 2^{-i} y_i \\ y_{i+1} = y_i + \delta_i 2^{-i} x_i \\ z_{i+1} = z_i - \delta_i \tan^{-1}(2^{-i}) \end{cases} \quad (1)$$

其中, $\delta_i = \text{sign}(z_i) \in \{1, -1\}$, 表示当前相位与期望值差值的正负, 用于指示下次迭代的旋转方向^[4], i 为迭代的次数. 当 $i \rightarrow \infty$ 时, $z_i \rightarrow 0$. 若初始值令 $x_0 = K$ (K 为伸缩因子, $K \approx 0.6073$), $y_0 = 0$, 则 $i \rightarrow \infty$ 时, $x_i \rightarrow \cos z_0$, $y_i \rightarrow \sin z_0$. 传统算法可以计算的角度 (z_0) 范围为 $[-99.88^\circ, 99.88^\circ]$.

从式(1)中可以发现, 每一次迭代的旋转方向都需要上一次迭代角度来确定. 而要实现精度越高, 则要求迭代的次数 N 越多^[10]. 这样就使得每一次角度值的计算都需要经过 N 个时钟, 限制了计算的速度. 因此, 利用传统 CORDIC 迭代, 需要在精度和速度之间做一个权衡.

3 Parallel CORDIC 算法优化及性能分析

在本文中, 利用三角函数的对称性, 以 $[0, \pi/4]$ 的正余弦值可以表示出 $[0, 2\pi]$ 范围内的所有值^[1]. 角度值利用的是弧度制, 以 26 位浮点数表示, 其中, 最高位为符号位, 次高位为整数位, 其余 24 位表示的是小数位. 这部分设计是本文的核心部分.

3.1 角度分解处理

考虑一个二进制角度值 $\theta = \sum_{k=1}^N b_k 2^{-k}$, 其中 $b_k \in \{0, 1\}$ ^[6]. 在文献[6]中提出, N 位的 CORDIC, 当 $i = 1 \sim [(N - \log_2 3)/3]$ 时, $\tan^{-1}(2^{-i}) \neq 2^{-i}$, 用 2^{-i} 替代 $\tan^{-1}(2^{-i})$ 的误差超出设计精度所能容忍的范围; 当 $i \geq [(N - \log_2 3)/3]$ 时, $\tan^{-1}(2^{-i}) \approx 2^{-i}$, 用 2^{-i} 替代 $\tan^{-1}(2^{-i})$ 的误差不影响设计精度. 在本设计中 $N = 24$, 高 7 位小数位, 不可以用 2^{-i} 替代 $\tan^{-1}(2^{-i})$, 必须要对角度进行转换, 分解为 \tan^{-1} 之和的表示形式, 转换公式^[3,5,6]如式(2)~(8)

$$2^{-1} = \tan^{-1}(2^{-1}) + \tan^{-1}(2^{-5}) + \tan^{-1}(2^{-8}) + \underbrace{0.000_000_000_000_111_100_001_110}_{e_1} \quad (2)$$

$$2^{-2} = \tan^{-1}(2^{-2}) + \tan^{-1}(2^{-8}) + \underbrace{0.000_000_000_000_100_100_100_010_100}_{e_2} \quad (3)$$

$$2^{-3} = \tan^{-1}(2^{-3}) + \underbrace{0.000_000_000_010_101_001_000_101}_{e_3} \quad (4)$$

$$2^{-4} = \tan^{-1}(2^{-4}) + \underbrace{0.000_000_000_000_010_101_010_010}_{e_4} \quad (5)$$

$$2^{-5} = \tan^{-1}(2^{-5}) + \underbrace{0.000_000_000_000_000_010_101_010}_{e_5} \quad (6)$$

$$2^{-6} = \tan^{-1}(2^{-6}) + \underbrace{0.000_000_000_000_000_000_010_101}_{e_6} \quad (7)$$

$$2^{-7} = \tan^{-1}(2^{-7}) + \underbrace{0.000_000_000_000_000_000_010}_{e_7} \quad (8)$$

$e_1 \sim e_7$ 与输入角度的低 16 位合并, 新的低 16 位角度为:

$$\hat{\theta}_L = \theta_L + \sum_{k=1}^7 r_k e_k = \sum_{k=1}^{24} \hat{\delta}_k 2^{-k} \quad (9)$$

因为 $\hat{\delta}_k 2^{-k} = 2^{-(k+1)} + \hat{r}_k 2^{-(k+1)}$, $\hat{r}_k = 2\hat{\delta}_k - 1$ ^[1,5,6], 则新的低 16 位角度可以重新表示为:

$$\hat{\theta}_L = (1 - 2\hat{\delta}_7)2^{-8} + \sum_{k=9}^{25} \hat{r}_k 2^{-k} - 2^{-25} \quad (10)$$

在式(3)(4)(10)中可以发现, $\tan^{-1}(2^{-5})$ 重复了一次, $\tan^{-1}(2^{-8})$ 重复三次, 则伸缩因子也必须进行修正, 公式^[5]为

$$K' = \prod_{i=1}^N (1 + 2^{-2i})^{-1/2} \times (1 + 2^{-10})^{-1/2} \times (1 + 2^{-16})^{-3/2} \quad (11)$$

在本文中, $K' \approx 0.858346641$.

CORDIC 算法的优势之一就是乘除运算以移位运算完成, 加快了运算速度. 在本文中, 角度浮点数的低 24 位为小数位, 而移位运算超过 25 位时, 所有的有效位则全部被移出, 运算将失去意义. 因此, 迭代的 13~25 级可以合并, 公式为^[1,5,9]式(12)

$$\begin{cases} x_{k+l} = x_k + y_k \sum_{i=k}^{k+l-1} \hat{r}_i 2^{-i} \\ y_{k+l} = y_k - x_k \sum_{i=k}^{k+l-1} \hat{r}_i 2^{-i} \end{cases} \quad k > \frac{N}{2} \quad (12)$$

3.2 旋转迭代的合并

在经过角度分解后, 每一次的旋转方向都可以根据输入的角度值提前预知. 但是本设计中的 Parallel CORDIC 算法是应用到 DDS 中的, 如果将前 8 级合并, 后 16 级合并, 这样的方法是可行的, 但是过深的逻辑运算容易造成整体时钟频率的下降. 虽然这样的合并经两个时钟的运算, 可能会缩短输入角度值到输出函数值的计算时间, 但这不利于 DDS 设计. 因此, 在本设计中, 充分考虑到 pipeline 的设计方法, 将 24 级迭代, 缩减为 7 级, 后 16 次的迭代分为 2 级, 前 8 次的迭代则根据运算数据的个数和逻辑深度, 均分为 5 级, 平均每级 8 个 26 位数参与加法运算.

因为本文的输入角度限定为 $[0, \pi/4]$, 所以第一次迭代的方向一定为正, 故可以将第一次迭代的结果提

前运算出来,作为后续迭代的常数参数:

```
'define x_init 26'h0d7d8ee
//Because the input angle must be bigger than
//zero,
'define y_init 26'h075947f
//so the value of the first stage rotation for
//2^-1, 2^-5, 2^-8 are calculated directly.
```

3.3 进位保留加法器的设计

在CORDIC算法中大量应用到了加法器,26位数据的累加运算,由于过长的进位链导致传统加法器在实现上具有很大的延时.因此必须对加法器进行优化.加法器的优化设计是本设计优化的重点.当加法器运算位数低于16位时,超前进位加法器、波纹加法器比进位保留加法器具有更快的运算速度,但当加数的位数超过16位时,进位保留加法器(CSA)则展现出明显的优势.根据每级运算的逻辑深度,本文的设计中采用的优化加法器结构为4:2压缩器,并利用4:2压缩器构成8:2压缩加法器,优化整个设计.

一位的3:2压缩器结构图^[12]如图2所示.

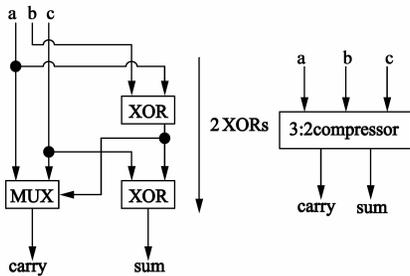


图2 一位3:2压缩器结构图

从图2中可知,一个3:2压缩器的延时为2个XOR门延时.传统的4:2压缩器是由两个3:2压缩器构成,结构图^[11~13]如图3所示.

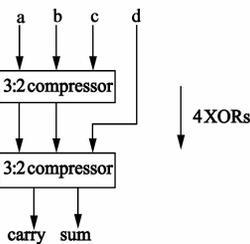


图3 传统4:2压缩器

传统的4:2压缩器的延时为4个XOR门延迟.本设计中应用的是优化4:2压缩器,一位的4:2压缩器延迟只为3个门延迟,相比传统的压缩器,缩减一个门延迟,其结构图^[11]如图4所示.

```
Cout = (in_1^in_2)? in_3:in_1;
sum = (in_1^in_2)^(in_3^in_4)^Cin;
carry = ((in_1^in_2)^(in_3^in_4))? Cin:in_4;
```

利用这种优化4:2压缩器,组成8:2压缩器,结构图如图5所示.

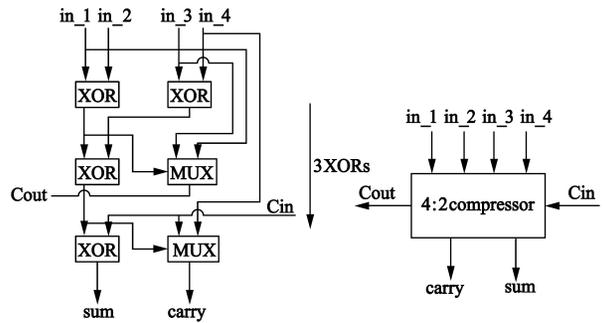


图4 优化4:2压缩器

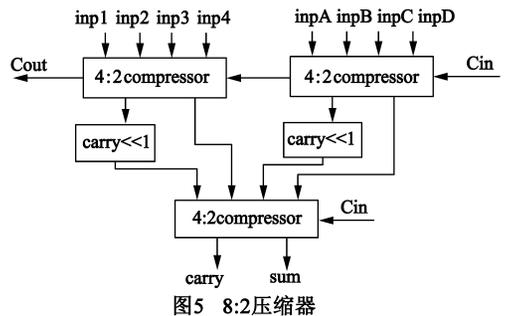


图5 8:2压缩器

在本文中,利用4:2压缩器组合成的8:2压缩器完成对8个26位数的并行压缩运算,是优化的CORDIC算法计算速度得到提高的重要原因.

3.4 性能分析

在精度上,24位小数的浮点数完全能实现 10^{-4} 的精度.然而,角度分解的方式和迭代次数的合并,仍然对计算引入了一些误差.表1和表2是本设计的精度数据输出结果.

表1和表2中分别列出了对CORDIC传统算法和Parallel CORDIC算法进行验证的结果.图6图7分别为sin, cos的传统算法(黑色曲线)及改进算法(灰色曲线)值与精确值的差值曲线图,横坐标为角度值,纵坐标为差值幅度.图中,以0.00024414rad为步长,曲线范围为0rad(0°)~0.78539816(45°).

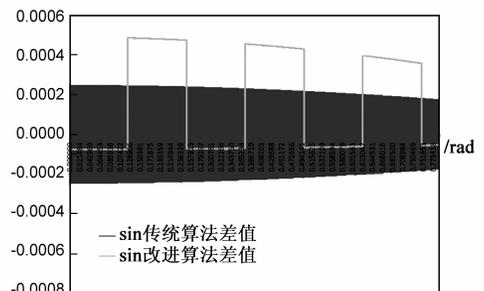


图6 传统算法及改进算法的sin值与精确值差值曲线图

表 1 传统 CORDIC 运算结果

角度	实际值		传统算法			
	sin	cos	sin		cos	
26'h0C90FDA(45°)	0.707106	0.707106	26'h0b4f9a1	0.706934	26'h0b51044	0.707279
26'h0B2B8C2(40°)	0.642787	0.766044	26'h0a48179	0.642600	26'h0c425ca	0.766201
26'h09C61AA(35°)	0.573576	0.819152	26'h092e303	0.573776	26'h0dlaac5	0.819011
26'h0860A91(30°)	0.5	0.866025	26'h07ff222	0.499788	26'h0ddbbd4	0.866147
26'h06FB379(25°)	0.422618	0.906307	26'h06c3f34	0.422839	26'h0e7fd09	0.906204
26'h0595C61(20°)	0.342020	0.939692	26'h0577f9a	0.341790	26'h0f09527	0.939775
26'h0430548(15°)	0.258819	0.965925	26'h042327e	0.258582	26'h0f74b11	0.965989
26'h02CAE30(10°)	0.173648	0.984807	26'h02c83f6	0.173888	26'h0fc1993	0.984765
26'h0165718(5°)	0.087155	0.996194	26'h0163fe9	0.086912	26'h0ff0802	0.996215
26'h000000(0°)	0	1	26'h001001	0.000244	26'h0fffffe	0.999999

表 2 Parallel_CORDIC 运算结果

改进合并算法(CSA)			
sin		cos	
26'h0b4eace	0.706707	26'h0b51f13	0.707505
26'h0a48db9	0.642787	26'h0c41b7a	0.766044
26'h092b79d	0.573114	26'h0d1c920	0.819475
26'h07fdffb	0.499511	26'h0ddc650	0.866307
26'h06c30b0	0.422617	26'h0e803d2	0.906308
26'h0576be7	0.341490	26'h0f09c4f	0.939885
26'h0421e4d	0.258274	26'h0f75074	0.966071
26'h02c7435	0.173648	26'h0fc1c5a	0.984807
26'h0162b07	0.086594	26'h0ff09d2	0.996243
26'h3ffdb11	-0.0005	26'h1000001	1.000000

改进算法 sin 的最大差值为 0.000485. 出现在 7°左右, cos 的最大差值为 0.000226 出现在 44°左右. 传统算法、Parallel CORDIC 算法与三角函数精确值的比对, 证明 24 位小数位的浮点数运算的误差达到 10^{-4} . 传统算法结果与精确值的比对, 印证了有限次的迭代必然引入计算误差^[9]. 通过 Parallel CORDIC 算法与传统算法结果的比对, 不难发现, 对输入角度的高 7 位小数位进行分解时, 由于校正角度位数上的限制, 引入了角度表示的误差, 这样二次误差传递, 使得计算结果更加偏离了实际值.

值得注意的一点是, 在运用 $\hat{b}_k 2^{-k} = 2^{-(k+1)} + \hat{r}_k 2^{-(k+1)}$, $\hat{r}_k = 2\hat{b}_k - 1$ 来对每一位进行重新表示时, 最低位 $\hat{b}_{24} 2^{-24} = 2^{-25} + \hat{r}_{24} 2^{-25}$. 右移 25 位的操作使得所有有效的小数位全部移出, 对计算结果不产生影响, 但是最低位的分解表示则失去了意义, 式(10)中的 -2^{-25} 也没有了实际意义, 这使得实际上的浮点数小数位数真正有意义的为高 23 位浮点数小数, 而不是最初设计

的 24 位.

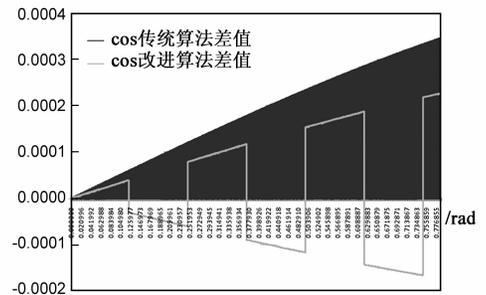


图7 传统算法及改进算法的cos值与精确值差值曲线图

虽然在改善精度上, 可以对特殊角度值的三角函数值进行预存储, 例如对 0°、30°、45°的函数值进行预存储, 当要计算的角度值落在设定的特殊角度中时, 输出的函数值直接采用查表的方式输出, 这样的修正对于单纯用于三角函数运算的 CORDIC 算法是有意义的, 但是对于应用到频率合成器中的 CORDIC 算法, 其意义并不大, 因为输入的控制字步长往往导致累加角度值很难恰好落到那些预设的特殊角度上.

可容忍的精度降低换取了运算速度的大幅提高. 本文在 Xilinx virtex5 XC5VLX30(ff676)开发板上进行了实测, 在只添加全局时钟约束的条件下, 从输入角度到输出函数值, 传统算法用时 72ns, 利用 Parallel CORDIC 算法及普通加法器时需要 63ns, 而运用 Parallel CORDIC 算法及 CSA 加法器时用时只需 41ns. 本设计在保证高精度的前提下, 将速度整整提高了 41.7%.

4 基于 Parallel CORDIC 的直接数字频率合成器的设计及性能分析

本文中, 直接数字频率合成器从相位量化的概念出发进行合成, 由四部分组成: 相位累加器、角度变换、Parallel CORDIC、输出选择. 结构图如图 8.

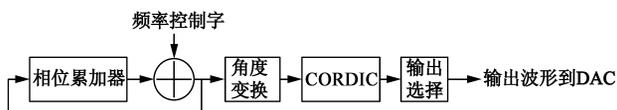


图8 基于Parallel CORDIC的DDFS结构图

4.1 相位累加器

相位累加器的设计,不是本文的重点和难点,但是它作为后续设计的输入,前一个周期波形与后一个周期波形的连接处必然要保持相位的平滑.在本设计中,当相位累加超过 2π 时,则将累加数减去 2π 后,再分别将角度送入后部逻辑和累加器,这样既保证了周期波形衔接的连贯性,同时还使累加相位一直在 $[0, 2\pi]$.

4.2 角度变换及输出选择

由于在设计时充分应用了三角函数的对称性,将Parallel CORDIC的角度输入局限在了 $[0, \pi/4]$,而累加器的累加相位值是在 $[0, 2\pi]$,所以必须要将累加器的计算结果转换到 $[0, \pi/4]$,也因此输出函数值必须经过相应的选择才能实现相位与幅度的一一对应.具体转换见表3.

表3 角度变换及输出选择

累加器 角度值(θ)	输入角 度变换(α)	函数值 输出 $\cos\theta$	函数值 输出 $\sin\theta$
$0 \sim \pi/4$	$\alpha = \theta$	$\cos\alpha$	$\sin\alpha$
$\pi/4 \sim \pi/2$	$\alpha = \pi/2 - \theta$	$\sin\alpha$	$\cos\alpha$
$\pi/2 \sim 3\pi/4$	$\alpha = \theta - \pi/2$	$-\sin\alpha$	$\cos\alpha$
$3\pi/4 \sim \pi$	$\alpha = \pi - \theta$	$-\cos\alpha$	$\sin\alpha$
$\pi \sim 5\pi/4$	$\alpha = \theta - \pi$	$-\cos\alpha$	$-\sin\alpha$
$5\pi/4 \sim 3\pi/2$	$\alpha = 3\pi/2 - \theta$	$-\sin\alpha$	$-\cos\alpha$
$3\pi/2 \sim 7\pi/4$	$\alpha = \theta - 3\pi/2$	$\sin\alpha$	$-\cos\alpha$
$7\pi/4 \sim 2\pi$	$\alpha = 2\pi - \theta$	$\cos\alpha$	$-\sin\alpha$

4.3 设计结果与性能分析

Parallel CORDIC 并行的计算方式实现每一个全局时钟都能正确输出一个函数值.同时以CORDIC替代传统的ROM查找表的方式来产生正弦波、余弦波,生成的波形具有完全正交的性质,不需要再进行相位的校正.

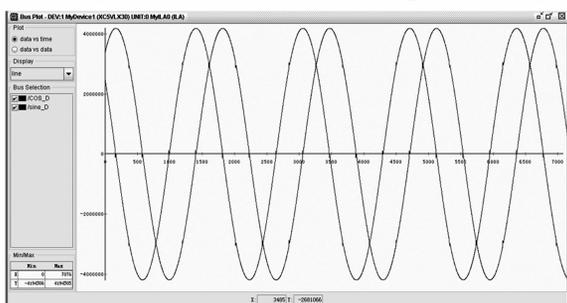


图9 chipscope在线采集波形图

在只添加全局时钟约束进行编译综合的条件下,本设计的时钟频率就可以达到179.49MHz.而整体设计的资源利用情况为:FF触发器使用量为1320,占片上可利用总量的6%,LUT的使用量为3850,占片上可利用总量的19%.

图9是将设计程序下载到xilinx-virtex5-XC5VXLX30(ff676)开发板,利用虚拟逻辑分析仪chipscope采集到的波形图.

5 总结

Parallel CORDIC 是传统CORDIC的优化改进,从输入的角度值中提前预测出每一次旋转的方向,从而能实现迭代次数的合并.在算法中,虽然以移位的方式替代了乘除法,但是26位数据的加法链造成了很大的延时.因此,加法器的优化对整个算法速度的提高做出了很大的贡献.4:2压缩器以及其组成的8:2压缩,应用到Parallel CORDIC算法中,将角度输入到函数值输出的延时相比传统算法减少了41.7%.以Parallel CORDIC为核心实现直接数字频率合成器,并行的方式直接实现每一个时钟完成函数值的计算,替代传统的查找表的方式,设计精度达 10^{-4} ,在只加全局时钟的约束下实现179.49MHz,实现了高精度高速度的设计.

参考文献

- [1] Avanindra Madiseti, Alan Y Kwentus, Alan N Willson. A 100-MHz, 16-b, direct digital frequency synthesizer with a 100-dBc spurious-free dynamic range[J]. IEEE Journal of Solid - State Circuits, 1999, 34(8): 1034 - 1043.
- [2] Chang Yongkang, Earl E Swartzlander. An analysis of the CORDIC algorithm for direct digital frequency synthesis[A]. Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures and Processors[C]. NY, USA: IEEE, 2002. 112 - 119.
- [3] TerenceK Rodrigues, Earl E Swartzlander. Adaptive CORDIC: using parallel angle recoding to accelerate rotations[J]. IEEE Computer Society, 2010, 59(4): 522 - 531.
- [4] 张晓彤, 辛茹, 王沁, 李涵. 基于改进混合式CORDIC算法的直接数字频率合成器设计[J]. 电子学报, 2008, 36(6): 1144 - 1148.
ZHANG Xiao-tong, XINRu, WANG Qin, LI Han. Design of direct digital frequency synthesizer based on improved hybrid CORDIC algorithm[J]. Acta Electronica Sinica, 2008, 36(6): 1144 - 1148. (in Chinese)
- [5] Shen-Fu Hsiao, Yu-Hen Hu, Tso-Bing Juang. A memory-efficient and high-speed sine/cosine generator based on parallel CORDIC rotations[J]. IEEE Signal Processing Letters 2004, 11(2): 152 - 155.

- [6] Tso-Pin Chuang, Chao-Chuan Huang, Shen-Fu Hsiao. Design of a CORDIC-based SIN/COS intellectual property (IP) using predictable sign bits[A]. Proceedings of the 27th European Solid-State Circuits Conference[C]. Villach, Austria, 2001. 277 – 280.
- [7] B Lakshmi, A S Dhar. FPGA implementation of a high speed VLSI architecture for CORDIC[A]. Proceedings of IEEE Region 10 Conference TENCN[C]. USA; IEEE, 2009. 1 – 5.
- [8] Tso-Bing Juang. Low latency angle recoding methods for the higher bit-width parallel CORDIC rotator implementations[J]. IEEE Circuits and Systems Society, 2008, 55 (11): 1139 – 1143.
- [9] Tso-Bing Juang, Shen-Fu Hsiao, Ming-Yu Tsai. Para-CORDIC: parallel CORDIC rotation algorithm[J]. IEEE Circuits and Systems Society, 2004, 51(8): 1515 – 1524.
- [10] Hu, Yu Hen. The quantization effects of the CORDIC algorithm[J]. IEEE Signal Processing Society, 1992, 40(4): 834 – 844.
- [11] Chip-Hong Chang, Jiangmin Gu, Mingyan Zhang. Ultra low-voltage low-power CMOS 4 – 2 and 5 – 2 compressors for fast arithmetic circuits[J]. IEEE Transactions on Circuits and Systems, 2004, 51(10): 1985 – 1997.
- [12] Karuna Prasad, Keshab K Parhi. Low-power 4 – 2 and 5 – 2 compressors[A]. Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers[C]. USA: IEEE, 2001, 1: 129 – 133.
- [13] Alvaro Vázquez, Elisardo Antelo. Multi-operand decimal addition by efficient reuse of a binary carry-save adder tree[A]. Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)[C]. USA: IEEE, 2010. 1685 – 1689.
- [14] Junhyung Um, Taewhan Kim. An optimal allocation of carry-save-adders in arithmetic circuits[J]. IEEE Transactions on Computers, 2001, 50(3): 215 – 232.
- [15] Anita Sharma, Dr R DDaruwala. Digital frequency (sinusoidal) synthesizer using CORDIC algorithm[A]. Proceedings of IEEE 3rd International Conference on Communication Software and Networks (ICCSN)[C]. USA; IEEE, 2011. 521 – 524.

作者简介



祁艳杰 女, 1987年12月出生于河北廊坊. 北京交通大学硕士研究生, 研究方向为 verilog 数字 VLSI 设计及 FPGA 嵌入式开发在射频通信方面的应用.
E-mail: qiy_77@126.com



刘章发 男, 汉族, 1963年11月出生于安徽省. 博士, 现为北京交通大学电子信息工程学院教授, 研究方向: 通信与导航集成电路设计.
E-mail: zhfliu@bjtu.edu.cn