

基于非线性逼近法的 QDDFS 新构架

赵占锋, 周志权, 乔晓林
(哈尔滨工业大学, 黑龙江哈尔滨 150001)

摘要: 本文将三角近似法和非线性逼近法相结合, 提出了一种高压缩比的设计方案, 并对该设计方案进行了详细的理论分析和参数优化, 在 16 bit 的 QDDFS 系统中, 压缩比达到了 655.36, 并且不失真动态范围(SFDR) 优于 96dBc, 最后给出了本方案的详细结构.

关键词: DDFS; 三角近似法; 非线性近似

中图分类号: TN919.3 文献标识码: A 文章编号: 0372-2112(2007)01-0001-03

A New QDDFS Architecture Base on Non-Linear Approach

ZHAO Zhan-feng, ZHOU Zhi-quan, QIAO Xiao-lin
(Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

Abstract: Based on the principle of trigonometric approximation and non-linear approach, a new QDDFS Architecture with high compression ratio is presented. Its design principle has been analyzed theoretically and all the parameters have also been optimized in Matlab. A high compression ratio of 655.36 is achieved at 96 dBc in a 16-bit system. Finally the whole architecture for proposed scheme is given.

Key words: direct digital frequency synthesizer(DDFS); trigonometric approximation; non-linear approach

1 引言

DDFS 因其具有连续相位、频率变化快、频率分辨率高和信噪比高的诸多特点, 在航天、军事、民用等众多领域发挥了越来越大的作用. 在传统的正交调制和解调技术中, 采用移相器来获得正交的两路正弦余弦信号, 但往往存在不完全正交的问题. QDDFS 却能同时提供完全正交的正弦和余弦信号两路信号. 随着存储数据压缩比的增大、功耗的降低和输出频率的提高, DDFS 芯片在数字通信、无线通信和频率控制系统中应用越来越广.

DDFS 理论是由 Tierney, Rader 和 Gold 在 1971 年提出的^[1], 主要思想是将 $0 \sim 2\pi$ 的正弦值(或余弦值)量化后存储在 ROM 中, 利用相位累加器产生的数字序列作为地址, 对 ROM 查表获得正弦(或余弦)波形, 地址和相位、振幅三者之间完全对应的关系. 图 1 给出了 DDFS 的原理简图, 图中的输出 $\text{Sin}(2\pi n/2^L)$ 是离散正弦信号, 经过 D/A 转换器和低通滤波器之后得到模拟正弦信号.

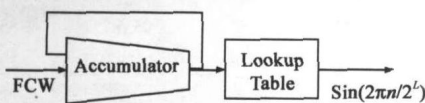


图 1 DDFS 原理简图

令 f_{clk} 表示系统时钟频率, f_{out} 表示输出的合成信号频率, L 表示相位累加器字长, 单位 bit, FCW(Frequency Control Word) 频率控制字, 则输出频率的表达式为:

$$f_{\text{out}} = \frac{f_{\text{clk}} \text{FCW}}{2^L} \quad 0 \leq \text{FCW} \leq 2^{L-1} \quad (1)$$

系统最小输出频率为:

$$\Delta f = \frac{f_{\text{clk}}}{2^L} \quad (2)$$

对于一个 L bit 的数字系统, 如果量化误差为 $\pm 2^{-1}$ bit, 则系统的量化信噪比为 $20\log(2^{L+1})$ dB, 它是该系统的极限信噪比, 实际系统的信噪比要比它低. 所以为了提高系统的信噪比系统量化的比特数越来越高从 16 bits、24 bits 增加到了 32 bits. 对于 16 bits, 24 bits 和 32 bits 的 DDFS 系统直接存储的数据就是 128 KBytes, 和 48 MBytes, 16 GBytes ($4 \text{ Gwords} \times 4 \text{ Bytes/word} \times 8 \text{ bits/Byte} = 128 \text{ Gbits}$). 对于这么大的数据表, 如果简单的存储到芯片中, 芯片功耗非常大、资源利用率也不高.

鉴于数据表的庞大, 众多的科学家开始研究数据表的压缩技术.

1984 年 David 在他的论文中提出了三角逼近算法^[2], 其压缩比率为 11.7:1.

1988 年 Henry T. Nicholas, Henry Samuelli 和 Bruce Kim 等人提出了 sine phase difference technique^[3], 将压缩比提高到了

13.5.1.

1997年中国台湾的学者 Shyuan Liao 和 Liang Gee Chen^[4], 对文献[3]进行了改进, 将压缩比提高到了 236.1.

2001年 Curticepan 等人^[5]将三角逼近算法进行了改进, 将压缩比提高到了 455.1.

2005年林茂六等人^[6]在 Shyuan Liao 和 Curticepan 的工作基础上, 将三角逼近算法和 Sine Phase Difference Algorithm 两者相结合, 利用式(3)、(4)来进行数据表的压缩, 每个数据点只需存储相应的差值, $\max\{f(x)\} = 0.0417$ 故每个正弦值可以节省 4 bit; $\max\{i(x)\} = 0.0803$ 每个余弦数据点可以节省 3 bit, 设 l 表示存储位数(用于量化三角函数的幅度 $[-1, 1]$), q 用于量化区间 $[0, 2\pi]$, p 是 q 的低位, m 和 n 表示参与乘法计算的位数, ϵ_c 和 ϵ_m 分别表示余弦函数近似为 1 时系统产生的误差和有限字长产生的误差, 在 $l = 16, q = 16, p = 7, m = 5, n = 6$ 条件下^[6], 压缩后数据量计算公式为 $(2l - 9)2^{q-3-p} + n2^{p-1}$ bits, 压缩比为 565.1.

$$f(x) = \sin(\pi x/2) - 1.375x \quad (3)$$

$$i(x) = \cos(\pi x/2) - (1 - x + 0.375x) \quad (4)$$

2 高压压缩比原理和参数优化

本文首次将非线性逼近法和三角逼近法相结合, 用于提高 QDDFS 数据表的压缩. 该方法主要用到了三种技术: 正弦弦对称性, 三角逼近法和非线性逼近法.

2.1 正弦余弦对称性

众所周知, 正弦函数和余弦函数关于 $\pi, \pi/2, \pi/4$ 存在对称性. 利用该对称性, 正弦和余弦函数仅仅需要存储 $0 \sim \pi/4$ 的数据, 从而达到压缩 LUT(Look Up Table)的目的.

2.2 三角逼近法

从数学理论, 可以得到等式(5)和(6), 如果 B 很小, 以至于 $\cos B \approx 1$, 结合实际情况, 那么该表达式可以写为式(7)和(8)^[5]:

$$\sin(A+B) = \sin A \cos B + \cos A \sin B \quad (5)$$

$$\cos(A+B) = \cos A \cos B - \sin A \sin B \quad (6)$$

$$\sin(\alpha + \beta) = [\sin \alpha]_l + [[\cos \alpha]_m [\sin \beta]_n]_l + \epsilon_c + \epsilon_m \quad (7)$$

$$\cos(\alpha + \beta) = [\cos \alpha]_l - [[\sin \alpha]_m [\sin \beta]_n]_l + \epsilon_c + \epsilon_m \quad (8)$$

$[]_l$ 表示 l bits 量化, 和 $l-1$ bits 截断; $[]_m$ 和 $[]_n$ 表示截断位数为 m 和 n .

应用三角逼近法后, 数据计算公式为 $(2l - 2)2^{q-3-p} + n2^{p-1}$ bits.

2.3 非线性折线逼近原理

折线逼近法最早由中国台湾学者提出来, 该文献[4]提出了采用折线逼近余弦曲线, 但是没有给出具体的实现过程, 该方法的压缩效果仅仅为 236.1.

本文首次对正弦函数采用了折线逼近法, 并且对余弦折线法的方程进行了修正; 经过仿真和参数优化后其折线方程为式(9)、(10)、(11)和(12), 这时的逼近效果最理想, 如图 2 和 3 为逼近效果图.

$$k(x) = \begin{cases} 0.5x & \text{when } 0 < x < 0.25 \\ 0.140625 & \text{when } 0.25 < x < 0.5 \end{cases} \quad (9)$$

$$h(x) = \cos(\pi x/2) - (1 - x + 0.125x + k(x)) \quad (10)$$

$$o(x) = \begin{cases} 0.25x & \text{when } 0 < x < 0.3125 \\ 0.078125 & \text{when } 0.3125 < x < 0.5 \end{cases} \quad (11)$$

$$r(x) = \sin(\pi x/2) - (1.25x + o(x)) \quad (12)$$

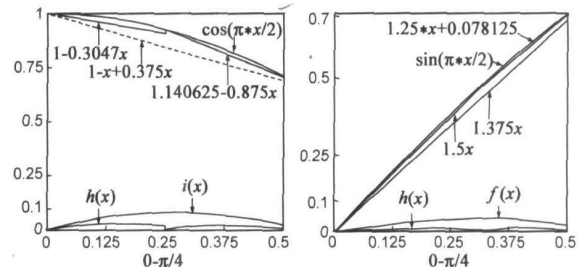


图 2 余弦逼近效果图

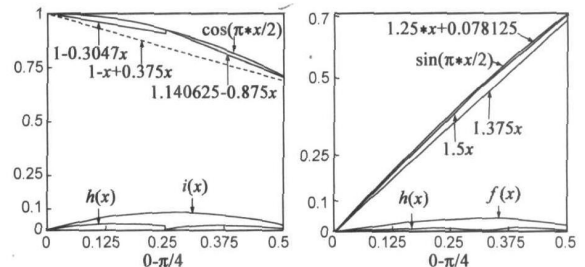


图 3 正弦逼近效果图

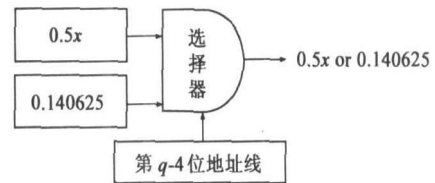


图 4 数据选择器

在 16 根地址寻址的系统中, 如果设小数点在 15 和 14 位之间, 16 位地址数据就可以表示为 $[0, 4)$ 之间的数据, 例如 0000 0000 0000 0000 表示 00.00 0000 0000 0000 即数字 0; 1111 0000 0000 0000 表示 11.11 0000 0000 0000 即 $3.75 - x$ 可以用文献[6]中介绍的方法实现, x 右移三位轻松实现 $0.125x$, 只要通过第 $q-4$ 条地址线来选择 $0.5x$ 和 0.140625 就可以轻松的实现, 见图 4. 关于(11)和(12), 只需要图 4 中的 $0.5x$ 和 0.140625 ($0.001001B$) 替换为 $0.25x$ 和 0.078125 ($0.000101B$), 地址线的 $q-4$ 和 $q-6$ 两条线取逻辑与来进行数据选择.

从表 1 中可以看出折线逼近法的逼近效果. 可以在文献[6]的基础上, 正弦和余弦数据表每个存储单元分别减少 2 bits, 达到 6 bits 和 5 bits. 压缩后数据量计算公式为:

$$ROM \text{ Size} = (2l - 13)2^{q-3-p} + n2^{p-1} \text{ bits} \quad (13)$$

表 1 压缩效果比较

方法	最大差值	0~0.25/0~0.3125	0.25~0.5/0.3125~0.5	节省效果
文献	$\{f(x)\}_{\sin}$		0.0417	4 bits
[6]	$\{i(x)\}_{\cos}$		0.0803	3 bits
本文	$\{h(x)\}_{\cos}$	0.0286	0.01897	5 bits
	$\{r(x)\}_{\sin}$	0.00904	0.00981	6 bits

表 2 参数优化选择

m, n	l, q, p	ROM	SFDR	Ratio
$m = 5$	(16, 16, 9)	1840	83.8035	569.88
	(16, 16, 8)	1376	93.5364	762.05
	(16, 16, 7)	1600	96.8757	655.36
$m = 6$	(16, 16, 6)	2624	101.3179	399.61
	(16, 16, 9)	1840	83.7024	569.88
	(16, 16, 8)	1376	94.2972	762.05
$n = 7$	(16, 16, 7)	1600	100.1591	655.36

表 2 为 m, n 不同情况下结果, 当 $p = 8$ 时, 系统最大的压缩比为 762.05.1; 当 SFDR 大于 96 dBc 时, 压缩为 655.36.1. 仔

