文章编号:1004-4213(2011)03-0407-6

一种光栅信号细分算法的 FPGA 实现

胡晓东[1],彭琅[1,2],雷明[3],雷宇生[4],熊梦飞[5],熊慕雪[6]

(1 中国科学院西安光学精密机械研究所,西安 710119)

(2 中国科学院研究生院,北京 100049)

(3 湖北航天飞行器研究所,武汉 430023)

(4 航天科工九院 国营江北机械厂,湖北 孝感 432000)

(5 北京师范大学 翰林实验学校,广东 东莞 523063)

(6 沙市中学,湖北 荆州 434000)

摘 要:提出了一种基于 FPGA 的光栅位移传感器信号细分处理技术,采用 CORDIC(坐标旋转数字计算机)解算相位,同时结合单准确度浮点运算,将辨向、粗码计数、相位解算、粗码精码浮点运算集成于单片 FPGA 实现,保证了细分的准确度、细分数据的有效计算准确度及信号处理的速度.设计中采用 Altera 公司的 Cyclone II_EP2C20F484C8 型芯片进行验证,对 20 μm 栅距的光栅尺信号进行细分处理,实验结果证实了该方法的正确性和可行性.

关键词:光栅位移传感器;FPGA;细分;CORDIC;浮点运算

中图分类号:TP212 **文献标识码**:A **doi**:10.3788/gzxb20114003.0407

0 引言

光栅位移传感器作为精密测量位移(或角度)的一种重要工具,已经普遍应用在雷达、光电经纬仪、机器人、数控机床等诸多领域[1].其基本原理是将直线或角度位移量的变化用莫尔条纹表现出来,同时对莫尔条纹信号进行处理得到位移量.

在许多对光栅信号电子学细分的处理方法中[2-4],都讨论了细分的原理以及实现的方法.但是可以发现,对高倍数细分、细分数据的有效计算准确度以及细分的处理速度三者结合起来考虑得很少.单从原理上,细分倍数是很高,但是硬件实现出来要打很大的折扣,因为输入、输出以及中间的运算过程都采用的是定点数,极大地约束了精码的分辨率和运算的有效计算准确度,许多方法只能用线性拟合或固化精码查找表来简化运算过程,这样就从细分原理到定点精码数据就损失了准确度,在随后的定点运算中继续损失准确度.在位移的解算中,处理速度是很重要的.在光栅读头允许的条件下,如果响应太慢,就会约束传感器的信号频率,从而限制了标尺光栅和指示光栅的相对移动速度.精码和粗码应该是相关的,二者的处理应该是并行进行的.如果先后进行必然降低处理速度,在这方面,采用单片 FPGA 结构比采用单片机、DSP 更有优势.FPGA 的并行处理的特点可以充分用在各模块内部以及模块之间.这样既能提高位移解算的正确性和系统的响应速度.本文从这个角度出发,提出了利用坐标旋转数字计算机(Coordinate Rotation Digital Computer,CORDIC)算法解算相位保证细分的高准确度,同时用单准确度浮点运算保证细分后的有效计算准确度,整个系统在 FPGA 中大量采用流水线结构实现保证了处理的速度.

1 细分电路组成

细分电路由移相电路、AD 转换电路、坐标平移模块、CORDIC 相位解算模块、浮点转换模块、粗码计数及辨向模块、浮点运算模块等构成,如图 1.由光栅尺产生的两路正交的正弦信号经过移相电路,得到 4 路相位相差 $\pi/4$ 的正弦信号 a、b、c、d,分别是:$A\sin\theta$、$A\sin(\theta+\pi/4)$、$A\cos\theta$、$A\sin(\theta+3\pi/4)$.其中 a、c 两路信号并行经过 16 位 AD 转换后,进行平移、翻转,再进行 CORDIC 相位解算及浮点转换,得到 32 位浮点精码.同时,a、b、c、d 四路信号通过比较器得到 4 路方波,送入 FPGA 中用状态机进行

第一作者:胡晓东(1964—),男,研究员,主要研究方向为光电信息与信号处理、计算机控制.Email:hxd@opt.ac.cn

通讯作者:彭琅(1985—),男,硕士研究生,主要研究方向为高速信号实时处理.Email:penglang2004@163.com

收稿日期:2010-11-25;**修回日期**:2010-12-21

区间判定和辩向计数，计数值经过浮点转换，得到32位浮点粗码. 然后粗码、精码经过加、减、乘浮点运算，得到32位的浮点位移结果，最后经过多字节BCD译码，将结果显示输出.

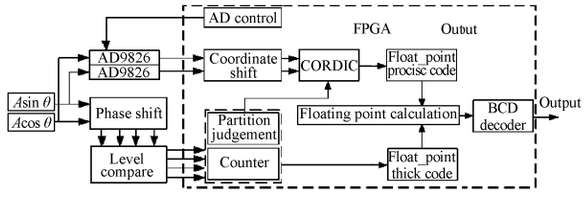

图1 细分电路组成框图

Fig. 1 Working frame of the subdividing circuit

2 细分原理

本文将一个正弦信号周期均匀的划分为8个区间，对整1/8周期的正弦信号计数得到位移粗码，对区间里不满1/8周期的正弦信号用CORDIC算法解算相位，得到区间里对应的位移精码. 将AD转换后的信号进行坐标平移翻转，得到$|A\sin\theta|$、$|A\cos\theta|$，区间划分如图2.

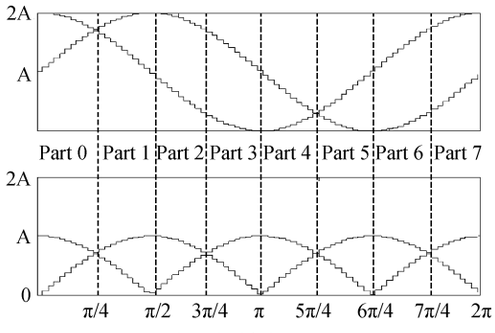

图2 细分区间划分图

Fig. 2 Subdividing zone partition

每个区间的弧度的范围是0到$\pi/4$，在坐标平移后，CORDIC算法模块解算的弧度范围是0到$\pi/2$，因此结合区间信息就能准确得出各个区间里的弧度位移，设由CORDIC算法解算出的弧度为φ，区间弧度位移为θ，则

在正向0、4，反向3、7区间，弧度位移为

$$\theta=\varphi \tag{1}$$

在正向1、5，反向2、6区间，弧度位移为

$$\theta=\varphi-\pi/4 \tag{2}$$

在正向2、6，反向1、5区间，弧度位移为

$$\theta=(\pi/2)-\varphi \tag{3}$$

在正向3、7，反向0、4区间，弧度位移为

$$\theta=(\pi/4)-\varphi \tag{4}$$

在本文中，设l是位移，d是栅距，θ是区间弧度位移，区间粗计数用N来表示，则位移可以表示为

$$l=\left(N+\frac{4\theta}{\pi}\right)\frac{d}{8} \tag{5}$$

a，b两点间的位移为

$$l_{ab}=\left(N_b-N_a+\frac{4(\theta_b-\theta_a)}{\pi}\right)\frac{d}{8} \tag{6}$$

2.1 区间判定、辩向、粗计数模块

在高速时钟下，对一路信号进行二级缓存处理，可以提取出该信号的跳边沿，对多路信号，也可以用二级缓存处理，判断出信号变化的方向. 本文采用VHDL语言描述的状态机来处理粗计数和辩向问题. 文中a、b、c、d四路信号通过电平比较得到4路方波. 1、0分别表示高、低电平，一个信号周期内的8个状态如图3.

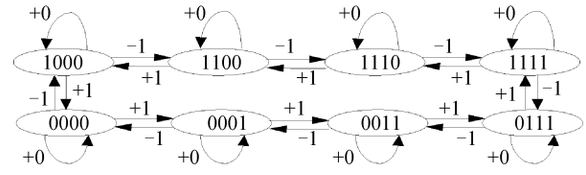

图3 状态变化图

Fig. 3 State movement

二级缓存辨向处理中，正向变化一个状态，计数加1，反向则计数减1，8个状态变化前后相互关联，状态的不连续跳变视为无效. 辩向与计数同时进行，根据这8个状态和变化方向就可以判断精码所在的区间.

2.2 CORDIC相位解算模块

在对诸如指数、三角函数和对数的硬件实现中，常用的方法是Talyor和Mactaurin级数展开，但是CORDIC算法对于三角函数的运算应该优先于Taylor展开[5]. CORDIC基本思想是通过对一系列固定的、与运算基数有关的角度的不断偏摆、迭代逼近所需要的旋转角度，这样就使得矢量的旋转和定向运算不需要进行乘法、开方、反三角函数等复杂的数学运算. CORDIC算法用于现代数字信号处理中求解复杂的函数[6]. 它可以在复杂角度旋转的情况下，把旋转运算转变为加、减和移位运算[7].

CORDIC算法有两种模式：旋转模式和向量化模式. 旋转模式是对向量旋转指定的角度，可以用来对信号进行相位调制. 而向量化模式是将所给向量旋转到X轴，得到所旋转的角度. 因此，向量化模式可用来解算相位. 设一向量(x_a,y_a)旋转∂角得到向量(x_b,y_b)，坐标旋转如式(7)

$$\begin{cases}x_b=x_a\cos\partial-y_a\sin\partial\\ y_b=y_a\cos\partial+x_a\sin\partial\end{cases} \tag{7}$$

矩阵形式为

$$\begin{bmatrix}x_b\\ y_b\end{bmatrix}=\begin{bmatrix}\cos\partial & -\sin\partial\\ \sin\partial & \cos\partial\end{bmatrix}\begin{bmatrix}x_a\\ y_a\end{bmatrix}=\cos\partial\begin{bmatrix}1 & -\tan\partial\\ \tan\partial & 1\end{bmatrix}\begin{bmatrix}x_a\\ y_a\end{bmatrix} \tag{8}$$

假设∂是由n个∂_i角度经过n步旋转而成，可

表示为 $\partial=\sum_{i=0}^{n}S_i\partial_i$，其中 s_i 表示旋转的方向，顺时针时为−1，逆时针时为 1. 每一步旋转的角度为 $\partial_i=\arctan\left(\frac{1}{2^i}\right)$，可得到：$\tan\partial_i=S_i2^{-i}$，每一次旋转可用式(9)表示

$$\begin{bmatrix}X_{i+1}\\Y_{i+1}\end{bmatrix}=\cos\partial_i\begin{bmatrix}1 & -S_i2^{-i}\\S_n2^{-i} & 1\end{bmatrix}\begin{bmatrix}X_i\\Y_i\end{bmatrix} \tag{9}$$

因为 $\cos\partial_i=\cos\left[\arctan\left(\frac{1}{2^i}\right)\right]$，经过足够多次数旋转后 $K=\frac{1}{P}=\prod_{i=0}^{n}\cos\left(\arctan\left(\frac{1}{2^i}\right)\right)\approx0.607\ 253$，该乘法因子在结果中是常量，只对结果向量的幅度有用，因此可以不考虑它对每次旋转的影响. 用 Z 来累计所转过的角度，那么每次旋转可表示为式(10). 经过足够多次的旋转，向量与 X 轴正向重合，此时得到式(11)

$$\begin{cases}X_{i+1}=X_i-S_i2^{-i}Y_i\\Y_{i+1}=Y_i+S_i2^{-i}X_i\\Z_{i+1}=Z_i-S_i\theta_i\end{cases}\text{，其中 }S_i=\begin{cases}+1,Y_i<0\\-1,Y_i>0\end{cases} \tag{10}$$

$$[X_n,Y_n,Z_n]=[P\sqrt{X_0^2+Y_0^2},0,Z_0+\arctan(Y_0/X_0)] \tag{11}$$

若令 $Z_0=0$，则得到所累计的角度 $\arctan(Y_0/X_0)$，在用 FPGA 实现时，用弧度来表示.

许多方法中，通常将三角函数线性化或线性拟合处理，虽然方便了计算，但是损失了准确度. 与传统的对反正切函数的线性拟合函数处理相比，用 CORDIC 来解算，具有更高的准确度. 利用 CORDIC 算法，可以以惊人的速度逼近反正切函数，如图 4.

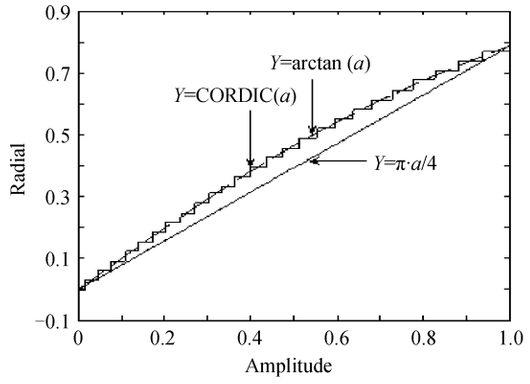

图 4　CORDIC 解算结果及对比

Fig. 4　Result of CORDIC solving and comparing

图 4 中 CORDIC 解算只用了前 8 级旋转因子，已经能很好地反映反正切函数的曲线轮廓，远比线性拟合函数 $y=\pi*x/4$ 精确. 如果再进行后几级更高准确度的旋转，用 CORDIC 算法解算出的弧度曲线几乎和反正切函数曲线重合，具有很高的准确度，而且旋转级数越高，解算出弧度的准确度就越高.

本文设计中拆开迭代环路，用 18 级流水线结构实现，第一级通过 X、Y 值的比较、交换将向量放在 0～π/4 范围内，中间 16 级进行 16 次旋转，最后一级根据第一级的比较结果和 16 次旋转累计的弧度解算出最终弧度的量化值. 在 CORDIC 算法的 FPGA 实现中，幅度输入值、每一级的弧度旋转因子、弧度输出值的量化关系一定要准确对应，否则将得到错误的弧度. 设定弧度 π 的量化值用十六进制表示为：80 000(HEX)，可以得到各级旋转的弧度的 20 位的量化值，例如 $\arctan(2^0)$、$\arctan(2^{-1})$、$\arctan(2^{-2})$依次为 20 000、12E40、09FB3. 仿真结果如图 5.

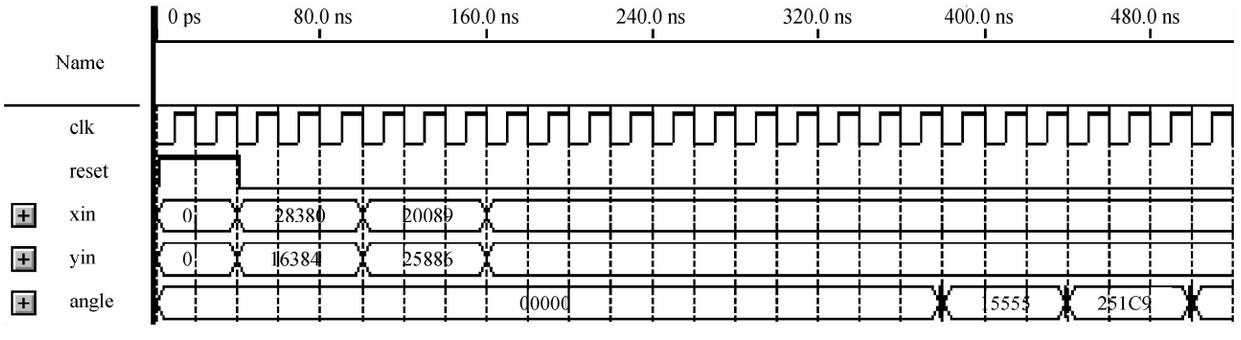

图 5　弧度解算

Fig. 5　Radian solving

$\arctan\left(\frac{16\ 384}{28\ 380}\right)=0.523\ 567$，解算出来的弧度量化值 15 555(HEX)，其量化值 15 555 的十进制值乘以 $\pi/2^{19}$，弧度值为：$0.523\ 596.\arctan\left(\frac{25\ 886}{20\ 089}\right)=0.910\ 826$，解算出来的弧度值为：0. 910 855. 这个模块的最后工作就是按式(1)、(2)、(3)、(4)用所得到的弧度∂结合区间信息解算出该区间的弧度位移θ.

文中 CORDIC 相位解算模块得到的是弧度 θ 的 20bit 的量化值，并不能直接用于粗精码结合运算，弧度的 20bit 的量化值要乘以 $\pi/2^{19}$ 才是最终弧度值. 因为定点运算很难解决动态范围大或者浮点数的基本运算. 但是如果采用浮点运算，就不存在这

些问题. 最终弧度结果，代入式(6)中，可知小数部分为：$(\theta_a-\theta_b)/2^{17}$. 本文将通过浮点转换模块将整数 N、小数 $\theta/2^{17}$ 转装换成 32 位单准确度浮点数，既方便了位移的解算，又不损失准确度. 浮点数比定点数的表示范围宽，有效准确度高，更适合于科学与工程计算的需要[8].

2.3 浮点运算单元(FPU)

单准确度浮点数(32bit)格式如下：$(-1)^s\times(1.0+\text{Mantissa})\times2^{(\text{Exponent}-b)}$. 其中：S 是浮点数的符号(1 代表负数，0 代表正数)；Exponent 是指数(单准确度：8 bit)；Mantissa 是有效数字(单准确度：23bit)；b 是偏移值(单准确度 127). 例如：$(7.625)_{10}\rightarrow(111.101)_2\rightarrow1.1101\times2^2$，按照 IEEE-754 标准[9]、用十六进制可以表示为：40F40000.

基本的浮点数加、减法步骤[10]如下：1)指数相减，求阶差；2)尾数移位，对阶；3)尾数加、减操作；4)指数调整，阶数再确定；5)结果规格化. 由于运算过程比较复杂，为了提高数据吞吐量，本设计采用 5 级流水线，能提高运算速度，代价是消耗 FPGA 中大量的寄存器.

在流水线设计中，获取第一个计算结果所用的时间是最多的，这个时间称为首次延迟. 本设计中，获得第一个结果要用 5 个时钟周期，之后只需一个时钟周期来获取随后的计算结果. 功能仿真数据如图 6(mode 为 1 的时候加，为 0 的时候减).

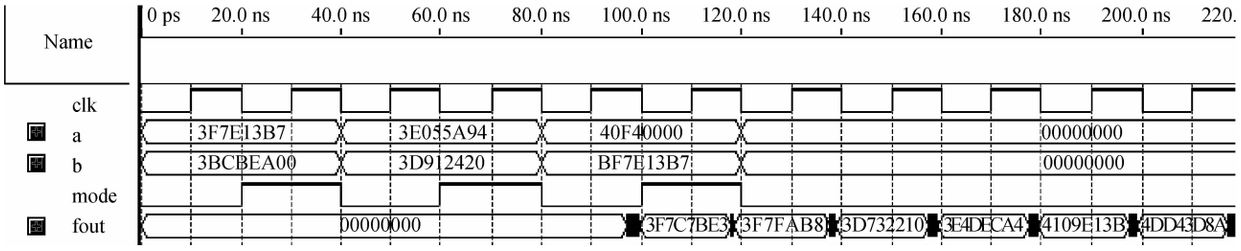

图 6 浮点加、减法仿真

Fig. 6 Simulation result of floating-point addition and subtraction

由式(6)知，粗码加精码的结果还要乘以 $d/8$，才能得到最终的位移结果. 当选用一个光栅尺后，栅距就确定了，本设计采用 QH-400 光栅位移传感器，栅距 0.02 mm，所以粗码加精码的结果还要乘以 2.5 μm. 2.5 对应 IEEE-754 浮点格式为：(40200000)(hex).

浮点乘法比浮点加减法简单，但计算量较大，这里采用十级流水线来实现，主要操作为：符号位处理(符号位异或)；指数位相加；尾数相乘及舍入；结果规格化输出.

在 FPGA 中，通过对莫尔条纹信号解算，得到位移信息. 仿真结果如图 7. 输入信号中，reset 为复位信号，clk 时钟频率为 50 MHz，a、b、c、d 信号频率为 100 KHz，sin[15..0]、cos[15..0]是经 AD 转换后的 16 bits 幅度值(图中用十进制表示)，zero 是相对起始位置确定信号；输出信号中，Int 是位移值的整数部分，Frac 是位移值的小数部分，sign 是位移的符号，1 为负，0 为正. 信号输出的首次延时 1.58 μs，随后获得结果只需 20 ns，数据处理的速度较快.

2.4 数据处理部分时序仿真

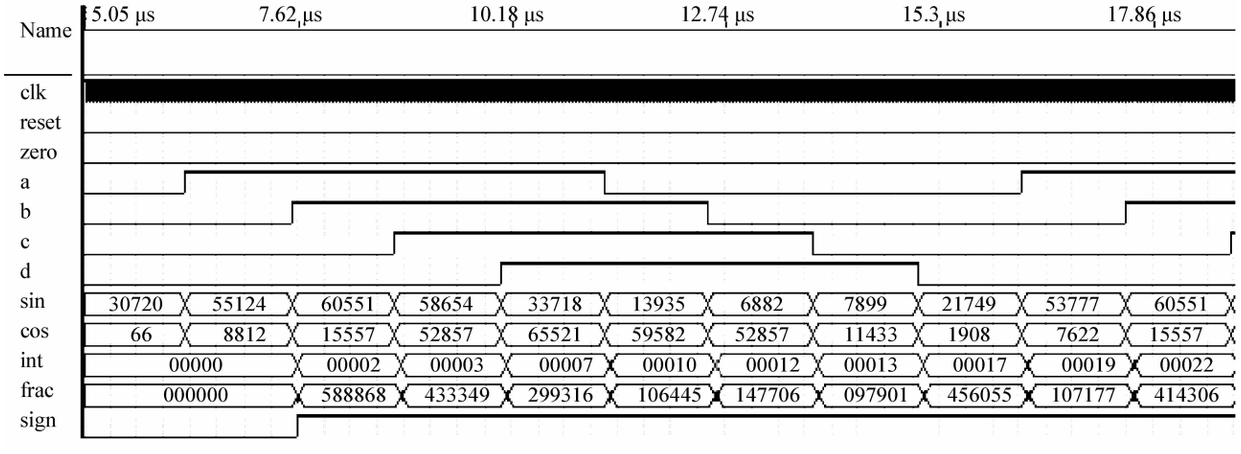

图 7 系统时序仿真结果

Fig. 7 Simulation result of the system

2.5　实验结果

上述细分算法和位移解算，已在试验中进行了测试. 位移测量结果用激光干涉仪 ML10 Gold 进行标定和对比，部分结果如表 1. L_g 是光栅尺测量结果，L_s 是干涉仪的测量结果.

表 1　测量结果

Table 1　Measurement results

Num	$L_s/\mu m$	$L_g/\mu m$	L_g-L_s
1	1 224.361	1 224.348	−0.013
2	1 218.535	1 218.521	−0.014
3	1 235.699	1 235.685	−0.014
4	3 137.185	3 173.171	−0.014
5	3 186.562	3 186.546	−0.016
6	3 187.268	3 187.254	−0.014
7	5 159.707	5 159.691	−0.016
8	5 170.966	5 170.953	−0.013
9	5 189.198	5 189.183	−0.015
10	7 202.504	7 202.491	−0.013
11	7 204.239	7 204.225	−0.014
12	7 215.701	7 215.685	−0.016

在小数最终结果的多字节 BCD 译码显示中，只用到了浮点数中 24 位有效尾数的高 12 位，所以实验值比理论值小.

3　结论

本文采用 FPGA 实现的、流水线结构的 CORDIC 算法解算相位得到高倍细分的精码，避免了对反正切函数的传统线性拟合所带来的误差，省掉了除法和查表过程，具有较高的准确度和速度，同时采用 32 位单准确度浮点运算，处理粗码、精码的结合相关运算，提高了精码分辨率，保证了位移的有效计算准确度. 另外，该方法将传统的“FPGA＋单片机”结构中单片机所做的数据处理部分也集成到 FPGA 里实现，浮点运算可以胜任对精码粗码、相对位移的处理，提高了系统的集成度.

参考文献

[1] DONG L L, XIONG J W, WAN Q H. Development of photoelectric rotary encoders [J]. *Optics and Precision Engineering*, 2008, **8**(2): 198-202.
董莉莉，熊经武，万秋华. 光电轴角编码器的发展动态[J]. 光学精密工程，2000，**8**(2)：198-202.

[2] HU Chao, FANG Gang, WANG Yao. The measurement method for incremental displacement transducer by using a computer[J]. *Acta Metrologica Sinica*, 1997, **18**(4): 50-54.
胡超，方刚，王耀. 增量式位移传感器信号的计算机检测[J]. 计量学报，1997，**18**(4)：50-54.

[3] LIU Zhong-li. Research on dividing and sensing about optical grating signal in angle measure unit [D]. Changchun: Changchun University of Science and technology, 2007.
刘中力. 光栅度盘测角仪中信号的辨向细分技术的研究[D]. 长春：长春理工大学，2007.

[4] YU Wen-xin, ZOU Zi-qiang, HU Xiao-tang. A new grating-based metrology method to achieve nanometer-order resolution [J]. *Aviation Precision Manufacturing Technology*, 2001, **37**(4): 35-37.
余文新，邹自强，胡小唐. 一种计量光栅实现 2 nm 测量分辨率的新方法[J]. 航空精密制造技术. 2001，**37**(4)：35-37.

[5] KITS S. 高级 FPGA 设计结构实现和优化[M]. 孟宪元，译. 北京：机械工业出版社，2009：94.

[6] ANDRAKA R A. A survey of CORDIC algorithms for FPGA based computers [C]. Proc 1998 ACM/SIGDA Sixth International Symposium on FPGA, 1998, 191-200.

[7] YU Hen-hu, HOMER H. M Chern, A novel implementation of cordic algorithms using backward angle rcoding(BAR)[J]. *IEEE Transactions on Computers*, 1996, **45**(12): 1370-1378.

[8] CHEN Fang-yuan. Research and implementation of key techniques of high performance floating point unit designs[D]. Changsha: National University of Defense Technology, 2007.
陈芳园. 浮点处理单元设计关键技术研究与实现[D]. 长沙：国防科学技术大学，2007.

[9] FARNUM C. Compiler support for floating-point computation [J]. *Software Practices and Experience*, 1988, **7**(18): 21.

[10] HENNESSY J L, PATTERSON D A. Computer architecture: a quantitative approach [M]. 2nd ed. San Fransisco: Morgan Kaufmann Publisher, 1996.

Realization of a Subdividing Method for Grating Signal Based on FPGA

HU Xiao-dong[1], PENG Lang[1,2], LEI Ming[3], LEI Yu-sheng[4], XIONG Meng-fei[5], XIONG Mu-xue[6]

(1 *Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an* 710119, *China*)

(2 *Graduate University of Chinese Academy of Sciences, Beijing* 100049, *China*)

(3 *Hubei Institute of Aerospacecraft, Wuhan* 430023, *China*)

(4 *SANJIANG Aerspace Group Jiangbei Machine Plant, Xiaogan, Hubei* 432000, *China*)

(5 *Beijing Normal University HanLin Experimental school, Dongguan, Guangdong* 523063, *China*)

(6 *ShaShi Middle School, Jingzhou, Hubei* 434000, *China*)

Abstract: A subdividing method for grating signal based on FPGA was proposed. The method uses CORDIC algorithms and single-precision floating-point calculation, which can implement the function of direction-judgement, counting, phase decoding by CORDIC algorithm, floating-point calculation between the two codes on a single FPGA. It can assure the subdivision accuracy, guarantee the effective calculation accuracy of subdivision data, and ensure the computational speed. The simulation was based on chip CycloneII-EP2C20F484C8 of Altera company, by subdividing the signal of grating with 2 μm length period. The result proves the feasibility and validity of this method.

Key words: Grating; FPGA; Subdividing; CORDIC; Floating-point calculation