Using the Nonparametric Curve Generator Algorithm in H/W Acceleration Solutions

I.JIVET, B.DRAGOI
Applied Electronics Department
University "Politechnica" Timisoara
V Parvan 2, Timisoara
ROMANIA
http://www.etc.upt.ro

Abstract: - The paper presents the implementation and an analysis of the Jordan's nonparametric curve generator algorithm in H/W acceleration solutions like sine/cosine generation. It is shown that the x and y coordinate sequences generated by the algorithm as the curve representation are really approximated nonuniform sampled cosine/sine functions sequences of values with useful properties. The nonuniformity of the samples along the curve inherent to the algorithm is analyzed and acknowledged as a drawback in its use for signal generation purposes. An extension of the four points neighbor algorithm that keeps the x and y sequences separate is shown to have promising efficiency in sine/cosine digital signal generation applications. Possible solutions for the correction of the non uniformity problem for practical implementations are presented. A VHDL description of the algorithm was derived, simulated and synthesized in a FPGA with final target H/W acceleration applications. FPGA synthesis efficiency results are compared with classical methods if sine/cosine generation like the CORDIC algorithm.

Key-Words: - Jordan's nonparametric curve generator algorithm, h/w acceleration in signal generation, VHDL synthesis for FPGA.

1 Introduction

Generation of sequences of values of functions yield better results by efficient hardware implementation of the algorithm then as traditionally by software. Many architectures of digital systems in use today are incorporating often classical and new algorithms translated to hardware and use the result for speed performance enhancement – hardware acceleration solutions.

The CORDIC algorithm is the algorithm most frequently used for the generation of trigonometric functions sequences of values and was used as a point of reference for this work [1].

The main advantages of the CORDIC algorithm are its simple implementation using only shift and add operations and a small ROM. A considerable amount of work has been done over the last decade to improve the efficiency and reduce the resources needed for an efficient implementation of the algorithm.

The particular area of application of the CORDIC algorithm with notable resent results in hardware acceleration solutions is the sine/cosine signal generators and mixers [2] [3].

The FPGA implementation constitutes the target technology of choice in the exploration of the feasibility and performance of a hardware implementation of the algorithm [1][4].

The abundance of resources in modern FPGA led companies to offer IP cores for sine/cosine signal generation using more complex circuits like multipliers or DSP mega-functions [9] [10]. Efficient implementations for high precision continues to fuel searches for even better solutions[6] [7] [8].

This paper presents the results of a study exploring the efficiency of the use of Jordan's nonparametric curve generator algorithm in non-graphics areas of applications and particularly the generation of sine/cosine functions.

In its original form the algorithm generates coordinates for points that 'follow' closely a line or quadratic curve in the plane. The basic idea of the algorithm is the embedding of the curve to be generated in a predefined orthogonal grid model of the coordinates space. The successive coordinates for each incremental step are determined based on the minimum distance to the curve as basic constraint [5].

One drawback of the current algorithm in practice is that the tracking speed of the approximating point on the curve is not constant. This is one of the reasons for which the Jordan's algorithm was not used in recent applications to its full potential.

In applications where the stepping uniformity in approximating the function can be relaxed we show that the nonparametric generation algorithm with appropriate adjustments on sample delays can be successfully and efficiently used. For a circle the coordinate pairs values form a sequence of sine and cosine values that can be used successfully in signal synthesis applications.

Part 2 examines the nonuniformity of the Jordan's algorithm and suggests solutions for appropriate adjustments. Part 3 of the paper presents a VHDL implementation, simulation and synthesis of the algorithm. The efficiency in FPGA hardware implementation in comparison with other algorithms is determined. Part 4 of the paper has the conclusions and an outline of further research .

2 Analysis of Jordan's algorithm

The nonparametric characteristic of the Jordan's algorithm resides in the basic method of coordinate pairs generation by moving in space from one grid point to one another under the minimizing criteria of the distance to the curve. As can be seen from Fig. 1 the size of the steps project non uniformly on the arch length span by the rotating vector.

One can clearly see that the horizontal and vertical segments of the approximating path make successive angles with the tangent to the curve that vary along the curve .

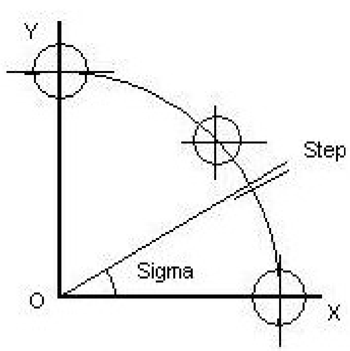

Fig.1 Source of projected stepping non uniformity along the curve as it is generated.

As a consequence the values of the x and y coordinates pairs generated do represent a sequence of approximating values of the curve characteristic sine and cosine functions but are non uniformly distributed with respect to the phase angle sigma..

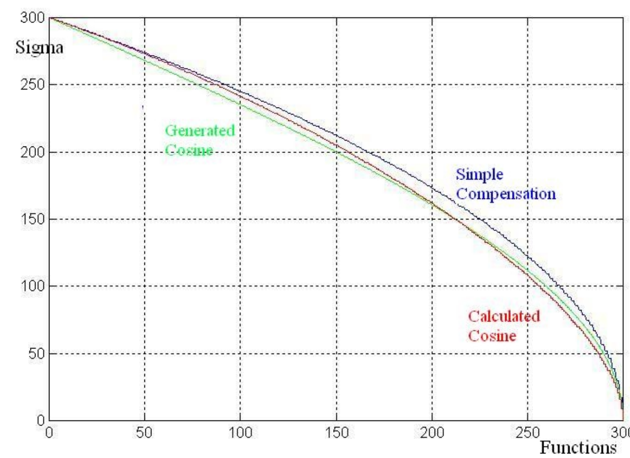

Fig 2. MathLab representation of the generated cosine function using the algorithm, a computed cosine and a simple phase correction result.

If these sequences of values are used to reconstruct the projections of the circle in a sine and a cosine function by presenting the values to a output at a constant frequency errors in the phase will occur resulting in generated function with less then ideal characteristics as it can be seen in Fig 2.

3 Phase non uniformity compensation

According to the algorithm the coordinates of the next approximating point to the curve are obtained by moving to the adjacent grid point in either x or y direction. The simultaneous move on both coordinates is a variant of the algorithm very useful in graphics applications but not appropriate for signal generation application since it further complicates the non uniformity in the phase.

The decision on the axis to take the step on corresponds to the minimum absolute value of the implicit function. The sign of the increment on one axis is the sign of the opposite axis partial variable derivative PDF of the implicit function. Zero values of the derivative (points exactly on circle), in the first quadrant trigger moves in positive directions on y and negative direction on the x axis for consistency.

The formulas below outline the calculus of the partial derivatives and implicit function values [5].

Deltax and deltay take the values {1,-1} if selected for a move along respective axis and zero if not selected. PDF variables stand for directional derivative values and Next and Last for implicit function variables values for each iteration.

PDFx = PDFx + 2 deltax (1)

PDFy = PDFy + 2 deltay (2)

NextFx = LastF + PDFx deltax + 1 (3)

NextFy = LastF + PDFy deltay + 1 (4)

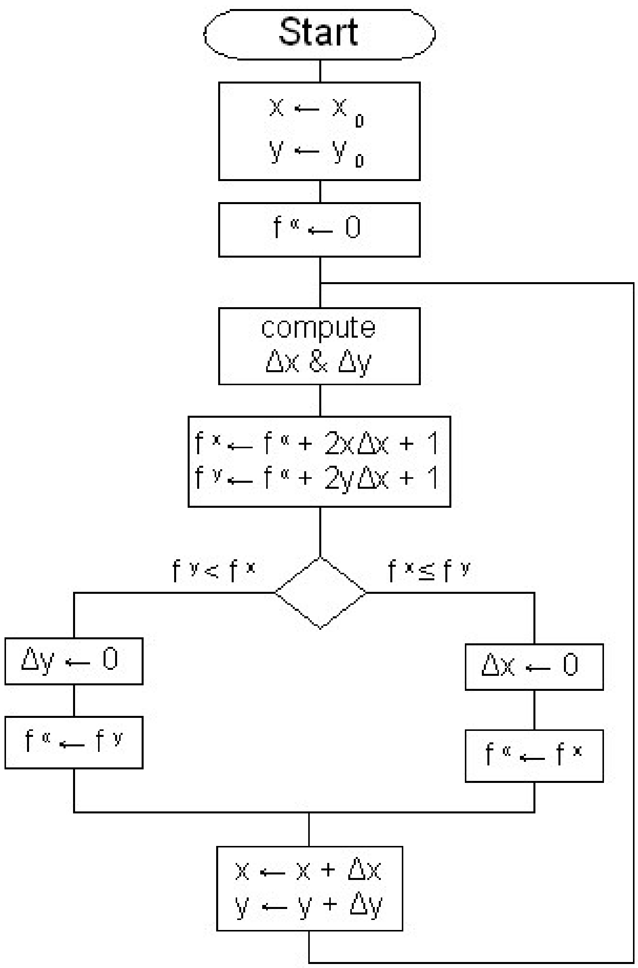

Fig.3 Original algorithm diagram for four point.

As it can be seen from the formulas and the algorithm in Fig. 3 only one addition and several increments are to be executed at each step.

As presented in the original paper the non uniformity is given by the following formula:

Ni/R = 1 + sin(sigma) – cos(sigma) (5)

where Ni is the number of steps taken up to the i-th node, R the radius and sigma the phase angle.

The proposed *extension* of the algorithm for sine and cosine function sequences generation with phase correction is :

Initialize variables with Start Coordinates;

Loop:

a)Select the x or y direction for the next step and the coordinate values according to the original algorithm;

b)Determine the phase angle advancement along the curve for the present step;

c) Take the generated coordinates for the new approximating point to the circle, as samples values for the sine and cosine functions ;

d)Successively present to the output the samples determined for each axis at a sample timing delay proportional to corresponding step phase advancement;

e)In the case of a interleaved step on the opposite axis add to sample timing a proportional delay corresponding to the phase advancement only and keep the sample value constant.

End Loop;

For ideal phase accuracy the choice would be to calculate the phase angle advancement at each step in advance and use stored values in the generation stage. The non uniform distribution of the values generated at output will still be there but the phase of the generated signal will not be distorted.

A more practical solution that minimize the hardware overhead is to use the just calculated coordinate value as a immediate approximation of the phase angle advancement. These values are the sine and cosine projection of the radius on the two axes.

A VHDL behavioral description of the algorithm was simulated and synthesized using ModelSim in a Xilinx ISE development environment for Spartan and Virtex families. We found that the synthesis yields far better results if the functional units are instantiated manually before the synthesis [4].

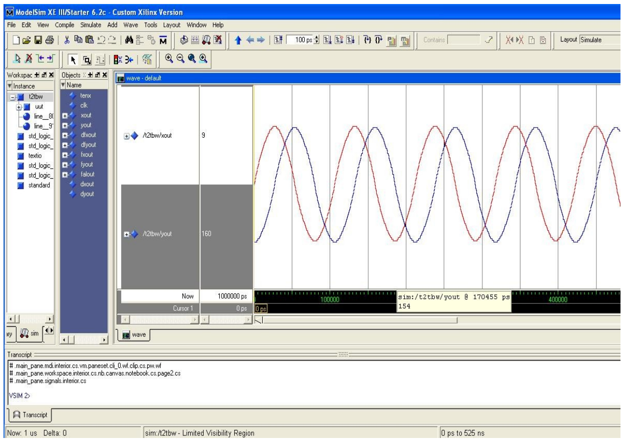

Fig.4 The VHDL behavioral simulation of the nonparametric curve generation algorithm.

As shown in Table 1 the FPGA resources needed for implementation confirm the practical technological efficiency for the use of the algorithm in sine/cosine generation applications [9] [10].

Table 1

Proposed Extended algorithm (8 - bit)

Target Device:	xc3s200-5ft256		xc2v500-6fg256	
Product Version:	ISE 9.1.01i		ISE 9.1.01i	
	VAR1 Partition		**VAR1 Partitio**	
No partition information was found.				
	Device Utilizatio		**Device Utilizati**	
Logic Utilization	**Used**		**Used**	
Number of Slice Flip Flops	27		26	
Number of 4 input LUTs	134		136	
Logic Distribution				
Number of occupied Slices	71		71	

OpenCore CORDIC Synthesis (16-bit)

Module Name:	r2p_cordic	• **Errors:**	No Errors
Target Device:	xc2v500-5fg256	• **Warnings:**	59 Warning
Product Version:	ISE 9.1i	• **Updated:**	Mon Jun 18
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	821	6,144	13%
Number of 4 input LUTs	755	6,144	12%
Logic Distribution			
Number of occupied Slices	441	3,072	14%

4 Conclusions

The paper presents a method to extend the Jordan's nonparametric curve generation algorithm to other applications. An extension of the algorithm is given such that the inherent non uniformity of the approximating points on the curve not to introduce phase errors.

The stepping intervals can be precomputed for a maximum accuracy of the generated sequence or timed proportional at each step from the generated coordinate values for the step end point. A trade-off on the loss of speed of generation versus the phase accuracy must be made.

The efficiency of the implementation in FPGA is synthesized from a VHDL description and compared with similar precision CORDIC algorithm implementation and IP CORDIC core parameters as reported in the literature. The efficiency of the proposed algorithm with respect to resources necessary is shown to be a good trade-off for speed.

References:

[1] R. Andraka, A survey of the CORDIC algorithms for the FPGA computers, *Proceedings of the ACM /SIGDA sixth international symposium on Field programmable gate array FPGA*, 3/1998, pp 191-200.

[2] T. C. Huang, H. T. Ho, Digital sine/cosine wave generator,*US Patent 6892213,* Issued May 10, 2005.

[3] Dengwei Fu, and A. N. Willson, Jr.A Two-Stage Angle-Rotation Architecture and Its Error Analysis for Efficient Digital Mixer Implementation, *IEEE Transactions on Circuits and Systems*,Vol.53,No. 3, March 2006.

[4]V. Boyer, Metaprogramming for the Generation of Nonparametric Curves, *EUROGRAPHICS*, 2001.

[5] B.W. Jordan, W.J. Lennon, and B.D Holm. An Improved Algorithm for the Generation of Nonparametric Curves. *IEEE Transactions on Computers*, Vol.C-22, No. 12, 1973, pp. 1052-1060.

[6] T. Vladimirova and H. Tiggeler,FPGA Implementation of Sine and Cosine Generators Using the CORDIC Algorithm, *Proceedings of 2006 MAPLD International Conference*, Washington, D.C.,September 26-28, 2006.

[7] D. Nowrouzezahrai, B. Decker and W. Bishop High-Performance Double-Precision Cosine Generation, *Proceedings of the 2005 International Conference on Computer Design*, Las Vegas, Nevada, USA, June 27-30, 2005 pp.42 – 48

[8] Madisetti, A.; Kwentus, A.; Willson, A.N., Jr. A Sine/Cosine Direct Digital Frequency Synthesizer using an Angle Rotation Algorithm, *1995 IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, Feb 1995 pp:262 – 263.

[9] Numerically Controlled Oscillators, *MegaCore Function Users Guide ,www.altera.com/literature/ ug/ug_nco.pdf* , May 07.

[10] CoreCORDIC CORDIC RTL Generator, *www.actel.com/ipdocs/CoreCORDIC_DS.pdf,* March 2006.