Anatolij SERGYENKO[1], Piotr RATUSZNIAK[2]

National Technical University of Ukraine, Kiev, Computer Engineering Dep. (1)
Koszalin University of Technology, Dep. of Electronics and Computer Science (2)

Searching for Pythagorean triples in FPGA

Abstract. *In this paper a new method for the calculating the Pythagorean triple is proposed which provides the deriving such a triple without solving the complex combinatorial task. Pythagorean triples provide the simple method of deriving exact values of trigonometric functions. The designed module, which calculates Pythagorean triples, is configured in FPGA, and can calculate the triple for a single clock cycle. This module can be used in the application specific processors for the linear algebra problem solving and digital signal processing.*

Streszczenie. *W artykule przedstawiono metodę obliczania trójek pitagorejskich, która w porównaniu z innymi metodami, nie zawiera złożonych obliczeń kombinatorycznych. Obliczone trójki pitagorejskie mogą służyć do obliczania dokładnych wartości funkcji trygonometrycznych. Zaprojektowana architektura obliczeniowa, służąca do obliczania takich trójek, została zaimplementowana w układzie FPGA. Zaprojektowany moduł, w przeciwieństwie do innych powszechnie znanych (np. CORDIC), pozwala na obliczenie trójki dla zadanego kąta w jednym takcie zegarowym. Zaprojektowana architektura może być wykorzystywana do obliczania np. algorytmów algebry liniowej lub w szeroko pojętym cyfrowym przetwarzaniu sygnałów. (**Implementacja wyszukiwania trójek pitagorejskich w układach FPGA**).*

Keywords: FPGA, CORDIC, Pythagorean triples, trigonometric function.
Słowa kluczowe: FPGA, CORDIC, trójki pitagorejskie, obliczanie wartości funkcji trygonometrycznych.

Introduction

There are many methods for calculating of trigonometric functions in FPGA. One of the most used methods is based on the table function, and is limited by the volume of the inner ROM of the device [1]. The interpolating polynomial evaluation method, for example, the Tailor polynomial method needs a lot of additions and multiplications of the data with the large bit width to derive results with the proper precision [1,2]. Two decades the CORDIC method is widely used in FPGA to calculate the different trigonometric functions [1,3,4,13]. The only disadvantage of this method consists in that for n precise output data bits this algorithm takes n sequential iterations. As a result, the CORDIC algorithm has long latent delay of calculations.

In the article a new method for calculating the trigonometric functions in FPGA is proposed, which is based on the properties of the Pythagorean triples.

Pythagorean triples and their properties

The solution of the Pythagorean problem consists in finding all square triangles with the natural sides *a,b,c*, i.e. in solving the Diophantine equation [5] $a^2 + b^2 = c^2$

This equation can be rewritten as follows.

$$\frac{a^2}{c^2} + \frac{b^2}{c^2} = 1 \tag{1}$$

It is the equation of the circle with the radius one, for which the following relation is satisfied:

$$sin^2\varphi + cos^2\varphi = 1 \tag{2}$$

where φ = atan(a/b). Formulas (1) and (2) show that for some angles φ the sine, cosine, and tangent values are equal to the rational fractions:

$$\sin\varphi = a/c\,, \cos\varphi = b/c\,, \tan\varphi = a/b \tag{3}$$

Therefore, one can get exact values of trigonometric functions (3) deriving the proper Pythagorean triple (*a,b,c*) for the given angle φ'. The problem is how to find the triple (*a,b,c*), which provides the satisfactory error ε of the angle representation.

The Pythagorean problem can be solved by a set of methods [5,6,7]. All of them are combinatorial ones. For example, Euclid has provided a formula for finding Pythagorean triples from any two positive integers m and n, $m > n$, namely [5,14]:

$$a = m^2 - n^2; b = 2mn; c = m^2 + n^2. \tag{4}$$

Traversing m and n, we can find a couple m, n, which satisfy $\varepsilon \le |\varphi' - \varphi|$, where φ = *atan(a/b)*. The searching area is minimized using the formula [5]

$$\frac{n}{m} \approx \frac{\sqrt{r^2+1}-1}{r} \tag{5}$$

where r = tan φ'. Due to the formulas (4), (5) the calculation of trigonometric functions using the Pythagorean triple is more complex than the usual methods. But it is reasonable to form the table of triples which are found for different angles. For the small angles φ the formulas are known, which depend on a single argument:

$$a = 2n + 1;\ b = 2n^2 + 2n;\ c = 2n^2 + 2n + 1; \tag{6}$$

$$a = 4n;\ b = 4n^2 - 1;\ c = 4n^2 + 1, \tag{7}$$

so that $\varphi \approx tan\ \varphi \approx 1/n$.

If we consider not natural but integer values of *a, b* then we get the generalized Pythagorean triples [6,7]. For such triples the operation of the angle addition operation is defined, which mimics complex multiplication. Consider two triples (a_1,b_1,c_1), (a_2,b_2,c_2) with the angles φ_1, φ_2, respectively. Then [5]

$$(a,b,c) = (a_1,b_1,c_1) \oplus (a_2,b_2,c_2) = (a_1b_1 + b_1a_2\,, b_1b_2 - a_1a_2, c_1c_2). \tag{8}$$

The generalized Pythagorean triples with the angle addition operation $\oplus$ form the abelian group [6].

Using Pythagorean triples in computation

The algorithm, which is implemented in FPGA as a hardware unit, must be the determined algorithm. This means that the algorithm flow does not depend on the input data; if the algorithm has a set of stages, then the number of them has to be predefined. Therefore, the algorithms, based on the combinatorial search, usually do not fit FPGA implementation. The two staged schema of the Pythagorean triangle search is proposed. The schema is based on the operation of angle addition (8). This means that the given angle φ' is represented by the sum of angles $\varphi' = \varphi'_1 + \varphi'_2$. At the first stage the triple (a_1,b_1,c_1) is searched, for which the angle $\varphi_1 = \varphi'_1 + \delta_{\varphi 1}$ differs from the given value φ'_1 to the difference $\delta_{\varphi 1}$. At the second stage the triple (a_2,b_2,c_2) is searched for the angle $\varphi_2 = \varphi'_2 - \delta_{\varphi 1}$. The resulting Pythagorean triangle is calculated by the formula (8). It represents the angle

$$\varphi = \varphi_1 + \varphi_2 = \varphi'_1 + \varphi'_2 + \delta_{\varphi_2} \tag{9}$$

where $\delta_{\varphi 2}$ is the error of the angle φ' representation.

The angle φ'_1 can be represented by the higher bits of the code φ', and the angle φ'_2 can be represented by the lower bits of it. Then the first stage of the schema can be implemented in the ROM, for which the code φ'_1 is an address input. Consider that the least significant bit of the code φ'_2 is equal to $\pi/2^{15}$, i.e. $\delta_{\varphi 2} < \pi/2^{16}$, and we take into account, that $n \approx 1/\tan \varphi_2$. Then the second stage can be calculated by the formula (6), when $\varphi_2 < \pi/512$, and by the formula (7), when $\varphi_2 < \pi/1024$.

The structure of the module, which performs the search for the Pythagorean triple for the angles $0 < \varphi' < \pi/4$, is shown in the fig.1. The input phase code is stored in the register RGP. The most significant bits of the code select a Pythagorean triple (a_1,b_1,c_1), and the angle error value $\delta_{\varphi 1}$ in ROM1. The corrected value φ_2 of the angle φ'_2, which is given by the least significant bits of RGP, is formed by the adder SM1, and selects the second Pythagorean triple (a_2,b_2,c_2) in ROM2. The multipliers MPU, and adders SM1, SM2 calculate the formula (8).

For example, consider the phase $\varphi' = \pi/6$ is given. It is represented by the code $85 \cdot 2^7+43$, which is stored in RGP. The triple (120,209,241), and the error code –7 are read from ROM1 by the address 85. The code to select the second triple is $\varphi_2 = \varphi'_2 - \delta_{\varphi 1} = 43+7 = 50$. This code represents the angle 0.0024. Then the factor n is equal to $n =]1/\tan(0.0024)[= 417$. Due to the formulas (6), the second Pythagorean triple is equal to (835, 348612, 348613), and it is read from the ROM2 by the address $\varphi_2 = 50$.

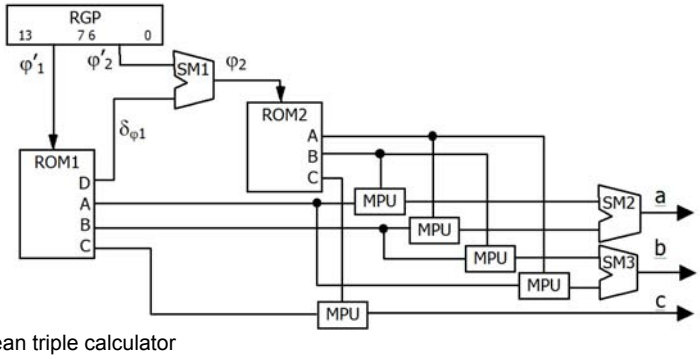

Fig.1. Structure of the Pythagorean triple calculator

The resulting Pythagorean triple, calculated due to the formula (8), is equal to (42007955, 72759708, 84015733). It represents the angle $\pi/6$ with the error $1.476 \cdot 10^{-5}$ radians, which is equal to 0.31 of the least significant bit in RGP.

The structure in the Fig.1 is implemented in the Xilinx Artix XC7A20SL, and occupies 235 LUT flip-flop pairs, and 8 multipliers DSP48E1. The maximum clock frequency achieves 135 MHz, when the usual synthesis constraints are used. This frequency can be much higher when the pipelining technique is used. If the fixed point results are needed then the division unit has to be attached, which hardware volume is no higher than four hundreds LUTs for the 16-bit result. The CORDIC module with the same precision, has, at least, in four times higher hardware volume, estimated in LUT flip-flop pairs, and the latent delay, which is higher in one degree of magnitude [13].

Conclusions

The Pythagorean triple provides the simple method of deriving the exact values of trigonometric functions. Such a triple can be found by a set of methods. But all of such known methods are combinatorial ones, and therefore, are ineffective ones for implementing in FPGA. A new method for the calculating the Pythagorean triples is proposed, which is based on the two step algorithm. The module which implements this algorithm is configured in FPGA with small hardware volume, and can calculate the triple for a single clock cycle. This module can be used in the application specific processors for the linear algebra problem solving, digital signal processing, and in many other fields.

REFERENCES

[1] Goldberg B.-G., Digital Frequency Synthesis Demystified, LLH Technology Publishing, USA, 1999

[2] Hariharan K., Hubert E.B., Divyalakshmi K.V.O., Shamalla K., Kumar A.V., Coherent Sinusoid Generation using Novel DDFS Architecture, *International Journal of Smart Home* Vol. 6, No. 1, January, pp. 17-28, 2012

[3] Andraka R., A survey of Cordic algorithms for FPGA based computers, ACM/SIGDA 6-th International Symposium on FPGAs, 1998, c. 1981-2000

[4] Sergyienko A., Maslennikov O., Implementation of Givens QR Decomposition in FPGA, Lecture Notes in Computer Science. – Berlin: Springer. –2002. –V. 2328. –P. 453–459

[5] Tan L., The Group of Rational Points on the Unit Circle, *Mathematics Magazine*, V69, N3, p.163-171, 1996

[6] McCullough D., Height and Success of Pythagorean Triples, *Mathematics Magazine*, V69, N1, p. 26-44, 2005

[7] Thibault Y., Kenmochi Y., Sugimoto A., Computing Admissible Angles from Rotated Digital Images, Combinatorial Image Analysis, LNCS, Springer, pp. 99-111, 2008

[8] Farouki R.T., Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable, Springer, Series: Geometry and Computing, Vol. 1, 2008

[9] Maslennikow O., Ratuszniak P., Sergiyenko A., Generator opisów VHDL bloków operacyjnych działających w arytmetyce ułamkowej, *PAK*, nr 8, 2008, 514-516

[10] Sergiyenko A., Maslennikow O., Ratuszniak P., Maslennikowa N., Tomas A., Application specific processors for the autoregressive signal analysis, 8-th Int. Conf. Parallel Processing and Applied Mathematic, PPAM'2009, Part I, LNCS, Springer, –, V. 6068, p. 80-86, 2010

[11] Maslennikow O., Lepekha V., Sergiyenko A., Tomas A., Wyrzykowski R., Parallel Implementation of Cholesky LLT Algorithm in FPGA–Based Processor, LNCS, Springer, V. 4967, p. 137-147, 2008

[12] Sergiyenko A., Melkovska V., A method of the signal demodulation with the DTMF modulation, *Bulletin of NTUU „KPI": Informatics, Control and Computer Engineering*, №48, p. 82-84, 2008. (In Ukrainian)

[13] *DS858, LogiCORE IP CORDIC v5.0, Product Specification,* October 19, 2011, available at http://www.xilinx.com/

[14] Joyce D.E., "Book X , Proposition XXIX", Euclid's Elements, Clark University, 1997

***Authors**: dr hab. inż. Anatolij Sergiyenko, Politechnika Kijowska, Wydział Inżynierii Komputerowej, Laboratorium KPI, ul. Peremogy,37, 03056 Kijów, Ukraina, E-mail: aser@ comsys.kpi.ua; dr inż. Piotr Ratuszniak, Politechnika Koszalińska, Wydział Elektroniki i Informatyki, Katedra Inżynierii Komputerowej, ul.Śniadeckich 2, 75-453 Koszalin, E-mail: ratusz@ie.tu.koszalin.pl.*