June 2013. Vol. 3, No.2 ISSN 2305-1493

International Journal of Scientific Knowledge
Computing and Information Technology
© 2012 IJSK & K.A.J. All rights reserved
www.ijsk.org

FPGA IMPLEMENTATION OF CORDIC ALGORITHM ARCHITECTURE

Ramanpreet Kaur[1], Parminder Singh Jassal[2]

Department of Electronics And Communication Engineering, Punjabi University
Yadavindra College of Engineering, Talwandi Sabo
Punjab, India
[1]raman5432@gmail.com, [2]pammi_jassal@yahoo.co.in

ABSTRACT-

The Coordinate Rotation DIgital Computer (CORDIC) algorithm has been used for many years for efficient implementation of vector rotation operations in hardware. It is executed merely by table look-up, shift, and addition operations. Thus, the corresponding hardware can be implemented in very economic fashion. Subsequently, it has been applied for many performances.CORDIC has been implemented in pocket calculators .The key concept of CORDIC arithmetic is based on the simple and ancient principles of two-dimensional geometry. It could be used as a single algorithm for unified implementation .The key issue in successful implementation of CORDIC architecture to be able to characterize the resource utilization with pipeline and non-pipeline structure.

Keywords— CORDIC ; Architecture

1. INTRODUCTION

First described in 1959 COordinate Rotation DIgital Computer (CORDIC) algorithm is an iterative algorithm, which can be used for the computation of trigonometric functions, logarithmic, complex number multiplication, matrix inversion, solution of linear systems and general scientific computation. Last half century has witnessed a lot of progress in design and development of architectures of the algorithm for high-performance and low-cost hardware solutions. CORDIC algorithm got its popularity, when [John S. Walther, 1971] showed that, by varying a few simple parameters, it could be used as a single algorithm for unified implementation of a wide range of logarithms, exponentials, and square functions. The popularity of CORDIC was very much enhanced thereafter primarily due to its potential for efficient and low-cost implementation. With the advent of low cost, low power Field Programmable Gate Arrays (FPGAs), this algorithm has shown its potential for efficient and low-cost implementation. CORDIC algorithm can be widely used in as wireless communications, Software Defined Radio and medical imaging applications, which are heavily dependent on signal processing.

Keeping the requirements and constraints of different application environments in view, the development of CORDIC algorithm and architecture has taken place for achieving high throughput rate and reduction of hardware-complexity as well as the latency of implementation. Some of the typical approaches for reduced-complexity implementation are focused on minimization of the complexity of scaling operation and the complexity of barrel-shifter in the CORDIC engine. Latency of implementation is an inherent drawback of the conventional CORDIC algorithm. Parallel and pipelined CORDIC have been suggested for high-throughput computation and efficient CORDIC algorithm.

1.1 Applications of CORDIC Algorithm

Applications of CORDIC are find in Sin and Cos Function ,Tangent, Inverse Trigonometric functions, Phase, Polar to Rectangular transformation. As we know the CORDIC was developed by volder in 1959 to compute the rotation of a vector in the Cartesian coordinate system. The method has been extended for computation of hyperbolic functions, multiplication, division, exponentials and logarithms [John S. Walther, 1971]. Two types of CORDIC architectural configurations have been proposed named:

1. Serial configuration
2. Parallel configuration
   a. No Pipeline mode
   b. Pipeline mode

2. Related work

CORDIC is a special-purpose digital computer for real-time airborne computation. The key concept of CORDIC arithmetic is based on the simple and ancient principles of two-dimensional geometry. But the iterative formulation of a computational algorithm for its implementation was first described in 1959 by Jack E. Volder for the computation of trigonometric functions, multiplication and division. This year therefore marks the completion of 52 years of the CORDIC algorithm, Not only a wide variety of applications of CORDIC have emerged in the last 52 years, but also a lot of progress has been made in the area of algorithm design and development of architectures for high-performance and low-cost hardware solutions of those applications. CORDIC-based computing received increased attention in 1971, when John S. Walther showed that, by varying a few simple parameters, it could be used as a single algorithm for unified implementation of a wide range of elementary transcendental functions involving logarithms, exponentials, and square roots.

Dirk Timmermann et al.(1992) contributed several methods for increasing the speed of the CORDIC algorithm. First they developed an improved method which guarantees a constant scale factor when employing redundant addition schemes. Then architecture with increased parallelism will be described which considerably reduces the CORDIC latency time and the amount of hardware. Shaoyun Wang and Vincenzo Piuri, 1997 showed that the CORDIC processor for the circular coordinate system in the rotation mode, the rotation directions for approximately 2/3 of the iterations can be derived in parallel without introducing any error. Both Dirk Timmermann et al. and Shaoyun Wang and Vincenzo Piuri use the parallel implementation but they ignored fully parallelism, if constant scale factor is used then number of micro-rotation is twice the no. of conventional rotation

Bimal Gisuthan et al. (2002) concluded a unified flat CORDIC architecture for generating trigonometric and hyperbolic functions has been presented. The equations for the flat CORDIC had been defined and a generalized architecture has been presented. The salient features of the flat CORDIC architecture had been demonstrated with the help of a 16-bit flat CORDIC. Based on the area-time parameters and the performance comparisons with other CORDIC based sine/cosine generators, it is evident that flat CORDIC sine/cosine generators are on an average 30% faster with a 30% saving on silicon area. Both Shaoyun Wang and Earl E. Swartzlander and Bimal Gisuthan and T. Srikanthan all ignored the resource utilization. A novel method for inferring rotation directions directly from the input angle circumventing altogether the overhead of sign pre-computation within the flat CORDIC algorithm

Javier Valls et al. (2002) presented a study of the efficient mapping on FPGA of the operators required to implement redundant arithmetic-based CORDIC algorithms. It was shown that the redundant arithmetic operators require a 4 to 5 times larger area than the conventional ones The setup time for the design is reduced by the introduction of the CORDIC element. Low power and high speed could still be achieved by reducing the transistor count and efficient design

Chuen- Yau Chen et al. (2003) proposed architecture for implementing the CORDIC algorithm. By decomposing an arbitrary rotation angle into a sequence of coarse angles and a fine angle which was small enough, this architecture could be implemented by performing a sequence of shift-and-add operations in the radix-2 system without any ROM lookup table or real multiplication requirement. It was suitable to be designed in pipelined architecture for performing the high-speed operations

S Suchitra et al.(2005) The first-cut implementation of the proposed design using 0.35-micron technology showed the area and latency estimates to be less than 95000 nand gate equivalents and 35 ns respectively for an order of 16-bits was proposed, but it ignored the optimization of speed

F.J. Jaime et al.(2010) described scaling free CORDIC and its improved version using conventional CORDIC stages for a convergence range increase up to $\pi/8$ shown as very good alternatives which outperform the conventional CORDIC capabilities. However, these architectures were also susceptible of being improved by the addition of several modifications, like the elimination of the domain folding technique and the inclusion of the modified radix-4 booth recoding algorithm. The new enhancements had been implemented and tested, obtaining some new architecture which is able to reach a 35% lower latency and a 36% reduction in area and power consumption compared to the original scaling-free architecture flexible way, increasing the accuracy and resilience levels of anti-spam techniques. The experiment results obtained when considering five users of the Enron collection. The results of the two collaborative methods (available in the prototype) are compared with the default behavior of the classifier using a single local filter.

3. TABLE OF COMPARISON

Author	Topic	Strength	Weakness
Dirk Timmermann et al.(1992)	Low latency time CORDIC algorithms	With increase in parallel configuration, hardware complexity will reduce.	Greater computational complexity.
Yu Hen Hu and S. Naganathan, (1993)	An angle recoding method for CORDIC algorithm implementation	Used to reduce the no. of iteration (angle of rotation is known in advance).	For unknown angle of rotation hardware implementation is complex and reduction in overall efficiency.
Shaoyun Wang and Vincenzo Piuri, (1997)	Hybrid CORDIC algorithms	It introduces CORDIC processor without any error using parallel configuration.	It works only in Rotation mode.

Javier Valls et al. (2002)	Evaluation of CORDIC algorithms for FPGA design		Not optimized in terms of speed.
Chuen-Yau Chen and Wen-Chih Liu, (2003)	Architecture for CORDIC algorithm realization without ROM lookup tables	To control the internal precision and optimized with speed.	It doesn't control the no. of iteration and not optimized in terms of area.
F. Angarita et al. (2005)	Efficient FPGA implementation of cordic algorithm for circular and linear coordinates		Fully parallel configuration and resource utilization are ignored.
S. Suchitra et al.(2005)	Elimination of sign pre-computation in flat CORDIC	Fully optimized with area.	Not optimized with speed.
F.J. Jaime et al. (2010)	Enhanced scaling-free CORDIC algorithm		lower latency & a 36% reduction in area and power consumption

CONCLUSIONS

CORDIC algorithm is very simple and iterative process. From the literature review on the topic that relate to an efficient CORDIC architecture, the main factors that affects the performance of CORDIC algorithm are high throughput and fully optimized in parallel configuration. The key issue in successful implementation of CORDIC architecture to be able to characterize the resource utilization with pipeline and non-pipeline structure. Very few papers are related to fully parallel configuration and efficient implementation. Most of the literature lacks in calculation of resources utilized by a particular CORDIC architecture.

REFERENCES

[1] Bimal Gisuthan and T. Srikanthan, "Pipelining flat CORDIC based trigonometric function generators," Microelectronics Journal, volume 33, pp.77–89, 2002.

[2] Chuen-Yau Chen and Wen-Chih Liu, "Architecture for CORDIC algorithm realization without ROM lookup tables," in Proceedings 2003 International Symposium on Circuits and Systems, ISCAS'03, May 2003, volume 4, pp. 544–547.

[3] Dirk Timmermann, H. Hahn, B. J. Hosticka, "Low latency time CORDIC algorithms," IEEE Transactions on Computers, volume 41, no. 8, pp.1010–1015, August 1992

[4] F. Angarita, A. Perez-Pascual, T. Sansaloni, and J. Vails, "Efficient FPGA implementation of CORDIC algorithm for circular and linear coordinates,"in Inernational Conference on Field Programmable Logic and Applications, August 2005, pp. 535–538.

[5] F.J. Jaime, M.A Sanchez, J Hormigo; J Villalba, E.L Zapata, "Enhanced scaling-free CORDIC algorithm", IEEE Transactions on Circuits and Systems Part I: Regular Papers Volume 57 Issue 7, July 2010.

[6] Jack E. Volder, "The CORDIC trigonometric computing technique," IRE Transactions on Electronic Computers, volume EC-8, pp. 330–334, Sept. 1959.

[7] Javier Valls, M. Kuhlmann, K. K. Parhi, "Evaluation of CORDIC algorithms for FPGA design," Journal VLSI Signal Processing Systems, volume 32,pp. 207–222, Nov. 2002.

[8] Ray Andraka, "Building a high performance bit serial processor in an FPGA." In On-Chip System Design Conference, Jan. 1996.

[9] S. Suchitra, S. Sukthankar, T. Srikanthan, and C. T. Clarke, "Elimination of sign precomputation in flat CORDIC," in IEEE Inernational Symposium on Circuits and Systems, ISCAS'05, May 2005, volume 4, pp. 3319–3322.

[10] Satish Ravichandran and Vqayan Asari, "Implementation of unidirectional CORDIC algorithm using precomputed rotation bits," in 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS 2002, August 2002, volume 3, pp. 453–45