International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

The Efficiency of Algorithms and the Cordic Architecture to Implement an Area Oflow and High Productivity

.G.Harish & .B.Shiva Kumar

1.Pg Scholar, Department of ECE, Vaagdevi College of Engineering, Bollikunta Warangal, Telangana

2.Assistant Professor, Department of ECE, Vaagdevi College of Engineering, Bollikunta Warangal, Telangana

ABSTRACT: *This paper presents an area-time efficient coordinate rotation digital computer (CORDIC) algorithm that completely eliminates the scale-factor. A generalized micro-rotation selection technique based on high speed most-significant-1-detection obviates the complex search algorithms for identifying the micro-rotations. This algorithm is redefined as the elementary angles for reducing the number of CORDIC iterations. Compared to the existing re-cursive architectures the proposed one has 17% lower slice-delay product on Xilinx Spartan XC2S200E device. The CORDIC processor pro-vides the flexibility to manipulate the number of iterations depending on the accuracy, area and latency requirement*

I. INTRODUCTION

The CORDIC algorithm involves a simple shift-add iterative procedure to perform several computing tasks by operating in either rotation-mode or vectoring-mode following any one among linear, hyperbolic, and circular trajectories [1]. Applications like singular value decomposition (SVD), Eigenvalue estimations, QR decomposition, phase and frequency estimations, synchronization in digital receivers, 3D graphics processor, and interpolators require the CORDIC to operate in both rotation and vectoring-modes. The 3D structures such as hyperboloids, paraboloids and ellipsoids require the CORDIC to be operated in both circular and hyperbolic trajectories. The hardware implementation of these applications requires more than one CORDIC processor operating in different modes and different trajectories. A reconfigurable CORDIC which can operate in rotation and vectoring-modes for both circular and hyperbolic trajectories can replace multiple CORDIC processors, and would be highly useful for such applications. A reconfigurable CORDIC can be utilized for a variety of applications in communication systems, signal processing, 3D graphics, robotics apart from general scientific calculations and waveform generations. In the last five decades, several algorithms have been proposed for area-delay-efficient and power-

efficient implementation of CORDIC algorithms, either for circular trajectory [2]–[7], or for hyperbolic trajectory [8]–[10]. But, we do not find any systematic study on design and implementation of reconfigurable CORDIC in the existing literature. A basic design of reconfigurable CORDIC based on a unified CORDIC algorithm [11] has been proposed recently [12]. The reconfigurable design of [12] is found to involve high reconfiguration overhead and results in low hardware utilization efficiency. Therefore, in this paper, we present a methodology for the design of reconfigurable CORDIC to be used for rotation-mode and vectoring-mode in circular and hyperbolic trajectories. The rest of the paper is structured as follows: Section II deals with a brief overview of CORDIC algorithm. In Section III we explore the possibility and difficulties in the design and implementation of reconfigurable CORDIC.

2. Pipelined Architecture of CORDIC

There are so many CORDIC architectures available in the literature. The pipelined architecture has an edge over others in terms of delay and throughput. Convergence is also quite good in this architecture. In this CORDIC architecture, a number of rotational modules have been incorporated, and each module is responsible for one elementary rotation. The modules are cascaded through intermediate latches (Fig. 1). During every stage within the pipelined CORDIC architecture, only adders/subtractors are used. The shift operations are hardwired permanently to perform multiplications by − i 2 reducing a large silicon area as required by barrel shifters. The precomputed values, as given in Table I, of i - th iteration angle α i required at each module can be Fig 1. Pipelined CORDIC Architecture The delay can be adjusted by using proper bit-length in the shift register. Since there is no need of sign detection for the convergence to get final outcome zero, the carry save adders are well-suited in this architecture. The use of these adders reduces the stage delay significantly. With the pipelining architecture, the propagation delay of the multiplier is the total delay of a single adder. So, ultimately, the throughput of the architecture is increased manyfold . It is obvious that if we increase the number of iterations, the latency of the design also will increase significantly. If an iterative implementation of the CORDIC were used, the processor would take several clock cycles to give output for a given input. But in the pipelined architecture, it converts iterations into pipeline phases. Therefore, an output is obtained at every clock cycle after pipeline stage propagation. Each pipeline stage takes exactly

one clock cycle to pass one output. The simulated output for digital Sine/Cosine

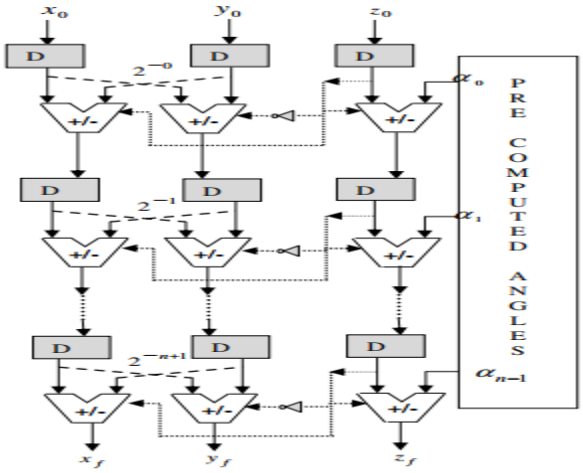


Fig 1. Pipelined CORDIC Architecture

3 Proposed Method:

The coordinate calculationmatrices for circular and hyperbolic CORDICs differ by the sign of operands, and to realize that additions are to be replacedby subtractions and vice-versa. This can be easily realized by areconfigurable add/subtract circuit. In both cases, the basic-shift could

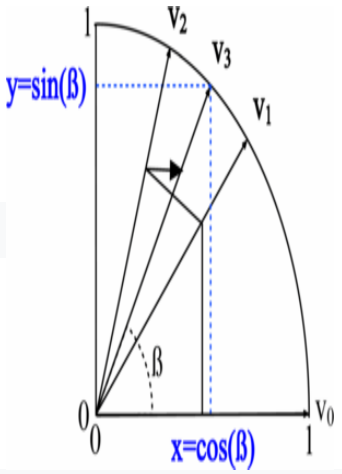
be either 2 or 3, but the number of microrotations vary with the mode of operation. Besides, each case will have its own circuit to enablethe extension of RoC. Based on these observations, we designthree reconfigurable CORDIC architectures: 1) rotation-mode

reconfigurable CORDIC; 2) vectoring-mode reconfigurable CORDIC;and 3) generalized reconfigurable CORDIC.

Rotation mode[edit]

CORDIC can be used to calculate a number of different functions. This explanation shows how to use CORDIC in *rotation mode* to calculate the sine and cosine of an angle, and assumes the desired angle is given in radians and represented in a fixed-point format. To determine the sine or

cosine for an angle , the *y* or *x* coordinate of a point on the unit circlecorresponding to the desired angle must be found. Using CORDIC, one would start with the vector :



An illustration of the CORDIC algorithm in progress.

In the first iteration, this vector is rotated 45°

counterclockwise to get the vector .

Successive iterations rotate the vector in one or the other direction by size-decreasing steps, until the desired angle has been

Vectoring mode

The rotation-mode algorithm described above can rotate any vector (not only a unit vector aligned along the *x* axis) by an angle between –90° and +90°. Decisions on the direction of the rotation depend on being positive or negative.The vectoring-mode of operation requires a slight modification of the algorithm. It starts with a vector the *x* coordinate of which is positive and the *y* coordinate is arbitrary. Successive rotations have the goal of rotating the vector to the *x* axis (and therefore reducing the *y* coordinate to zero). At each step, the value of *y* determines the direction of the rotation. The final value of contains the total angle of rotation. The final value of *x* will be the magnitude of the original vector scaled by *K*. So, an obvious use of the vectoring mode is the transformation from rectangular to polar coordinates.

Generalized Reconfigurable CORDIC.denotes the Reconfigurable Computing capability of a system, so that its behavior can be changed by reconfiguration, i. e. by loading different configware code. This static RESULTS:

reconfigurability distinguishes between reconfiguration time and run time. Dynamic reconfigurability denotes the capability of a dynamically reconfigurable system that can dynamically change its behavior during run time, usually in response to dynamic changes in its environment.In the context of wireless communication dynamic reconfigurability tackles the changeable behavior of wireless networks and associated equipment, specifically in the fields of radio spectrum, radio access technologies, protocol stacks, and application services.Research regarding the (dynamic) reconfigurability of wireless communication systems is ongoing for example in working group 6 of the Wireless World Research Forum (WWRF), in the Wireless Innovation Forum (WINNF) (formerly Software Defined Radio Forum), and in the European FP6 project End-to-End Reconfigurability (E^2R)..In the context of Control reconfiguration, a field of fault-tolerant control within control engineering, reconfigurability is a property of faulty systems meaning that the original control goals specified for the fault-free system can be reached after suitable control reconfiguration

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue14
November 2017

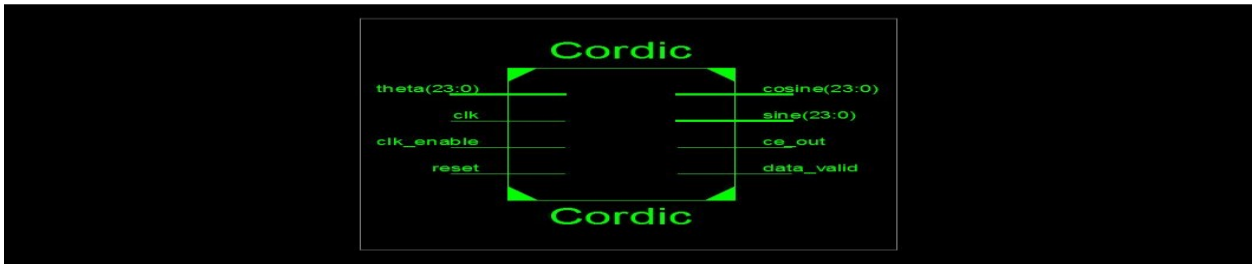


FIG: RTL LAY OUT

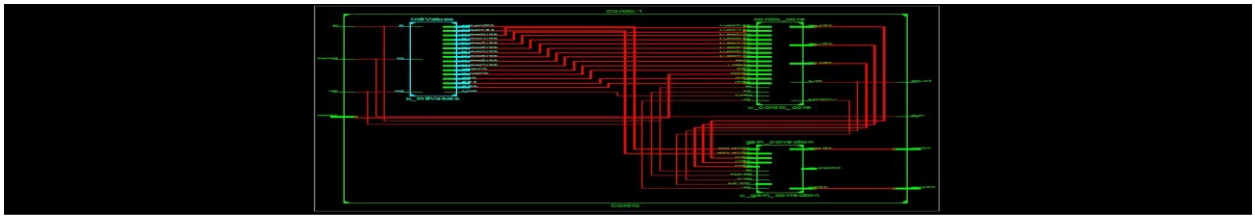


FIG: INTERNAL LAYOUT

CONCLUSION The proposed algorithm provides a scale-free solution for realizing vector-rotations using CORDIC. The order of Taylor series approximation is decided appropriately by the proposed algorithm, not only to meet the accuracy requirement but also to attain adequate range of convergence. The generalized micro-rotation selection technique is suggested to reduce the number of iterations for low latency implementation. Moreover, a high

speed most-significant-1 detection scheme obviates the complex search algorithms for identifying the micro-rotations. The proposed CORDIC processor has 17% lower slice-delay product with a penalty of about 13% increased slice consumption on Xilinx Spartan 2E device.

REFERENCES

[1] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput., vol. EC-8, pp. 330–334, Sep. 1959.

[2] K. Maharatna, A. S. Dhar, and S. Banerjee, "A VLSI array architecture for realization of DFT, DHT, DCT and DST," Signal Process., vol. 81, pp. 1813–1822, 2001.

[3] P. K. Meher, J.Walls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures and applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 9, pp. 1893–1907, Sep. 2009.

[4] C. S. Wu and A. Y. Wu, "Modified vector rotational CORDIC (MVRCORDIC) algorithm and architecture," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 48, no. 6, pp. 548–561, Jun. 2001.

[5] C.-S.Wu, A.-Y.Wu, and C.-H. Lin, "A high-performance/low-latency vector rotational CORDIC architecture based on extended elementary angle set and trellis-based searching schemes," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process, vol. 50, no. 9, pp. 589 601, Sep.2003.

[6] Y. H. Hu and S. Naganathan, "An angle recoding method for CORDIC algorithm implementation," IEEE Trans. Compute., vol. 42, no. 1, pp. 99–102, Jan. 1993.

[7] M. G. B. Sumanasena, "A scale factor correction scheme for the CORDIC algorithm," IEEE Trans. Compute., vol. 57, no. 8, pp. 1148–1152, Aug. 2008.

[8] J. Villalba, T. Lang, and E. L. Zapata, "Parallel compensation of scale factor for the CORDIC algorithm," J. VLSI Signal Process. Syst., vol. 19, no. 3, pp. 227–241, Aug. 1998.

[9] L. Vachhani, K. Sridharan, and P. K. Meher, "Efficient CORDIC algorithms and architectures for low area and high throughput implementation," IEEE Trans. Circuit Syst. II, Exp. Briefs, vol. 56, no. 1, pp. 61–65,Jan. 2009.

[10] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modifiedvirtually scaling-free adaptive CORDIC rotator algorithm and architecture,"IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 11,pp. 1463–1474, Nov. 2005.

[11] F. J. Jaime, M. A. Sanchez, J. Hormigo, J. Villalba, and E. L. Zapata, "Enhanced scaling-free CORDIC," IEEE Trans. Circuits