

A New Approach for High Performance and Efficient Design of CORDIC Processor

Rohit Kumar Jain, V.K. Sharma and K. K. Mahapatra
Dept. of Electronics & Communication Engineering
National Institute of Technology, Rourkela, India-769008
rohitjain.nitrkl@gmail.com, vijay4247@gmail.com, kmaha2@gmail.com

Abstract—This paper presents a new approach for the high performance and hardware efficient design of coordinate rotation digital computer (CORDIC) processor structure. The proposed design approach completely eliminates the ROM requirement of constant arctangent values. Furthermore, efficient designs of carry look ahead adders (CLAs), exploiting one input as constant, in the angle adder/subtractor datapath speeds-up the computation while maintaining regularity. The proposed architecture is implemented in FPGA as well as in 180nm standard cell library. The proposed implementation has about 39% delay improvement in FPGA and about 34% delay improvement in standard cell technology as compared to basic structure. About 47% power savings has been achieved in the proposed structure.

Keywords— Coordinate rotation digital computer (CORDIC), vector rotation, FPGA, carry look ahead adder (CLA), carry save adder (CSA).

I. INTRODUCTION

Coordinate rotation digital computer (CORDIC) processor computes a number of functions including sine and cosine of angles [1]–[3]. Since, CORDIC architecture is multiplication free (adder/subtractor as the main computational block), it has been used as a basic unit of computation in the implementation of different well known algorithms like discrete cosine transform (DCT), discrete Hartley transform (DHT), fast Fourier transform (FFT), etc. [4]–[8]. The basic CORDIC computations are expressed by the iterative equations at i th step as [9],

$$\begin{aligned} X_{i+1} &= k_i (X_i - m\sigma_i 2^{-S(m,i)} Y_i) \\ Y_{i+1} &= k_i (Y_i + \sigma_i 2^{-S(m,i)} X_i) \\ Z_{i+1} &= Z_i - \sigma_i \alpha_{m,i} \end{aligned} \quad (1)$$

where $i = 0, 1, \dots, N-1$. The number of iterations N will decide the fractional bit accuracy of the final result obtained [10], m parameter stands for one of the three coordinate systems namely linear, circular and hyperbolic (for $m = 0, 1$ and -1 respectively) and $S(m, i)$ is shift sequence having values $0, 1, \dots, N-1$ [11]. Scale factor k_i remains constant for a particular computer if all rotations from 0 to $N-1$ are

completed (i.e., iterations are not bypassed to achieve faster convergence) [12]. Parameter α_i is angle by which a vector is rotated in i th step and is given by,

$$\alpha_i = \tan^{-1} 2^{-i}$$

The parameter σ_i takes two values, -1 and 1 . If Z_i , the input angle, is left positive in a particular iteration step then its value is 1 otherwise -1 .

Figure 1 depicts the basic architecture of the CORDIC processor. Arctangent values are constant and they are stored in ROM. For the iteration to go from i th stage to $(i+1)$ th, the sign of Z_i has to be predetermined. Adder/subtractor is the only computational unit in Z -datapath, latency of this architecture is determined by the latency of the adder/subtractor module. As the delay time of the adder is proportional to its size in bit-width, redundant number system with sign-digit (SD) representation has been used in [13] and [14]. The delay time of the redundant adder is independent of the size of word and is approximately equal to delay time of two full adders. However, use of redundant number increases the hardware overhead (redundant to non-redundant converter is required) and also it makes scale factor variable (complex design) which has to be computed in each iteration [15]. Carry save adder (CSA) has been used in [16] to improve the speed as there is no carry propagation in CSA and the propagation delay is equivalent of only one full-adder [17]. Carry look-ahead adder (CLA) is very fast adder but it has the disadvantage of large silicon area.

In regular CORDIC VLSI structure, ROM is used to store the precomputed values of arctangents [18]. However, ROM based design is not preferred because ROM has slow speed (ROM access time) and more power consumption [19], [20].

In this paper we propose a ROM free, high speed regular architecture of CORDIC processor based on of CSA

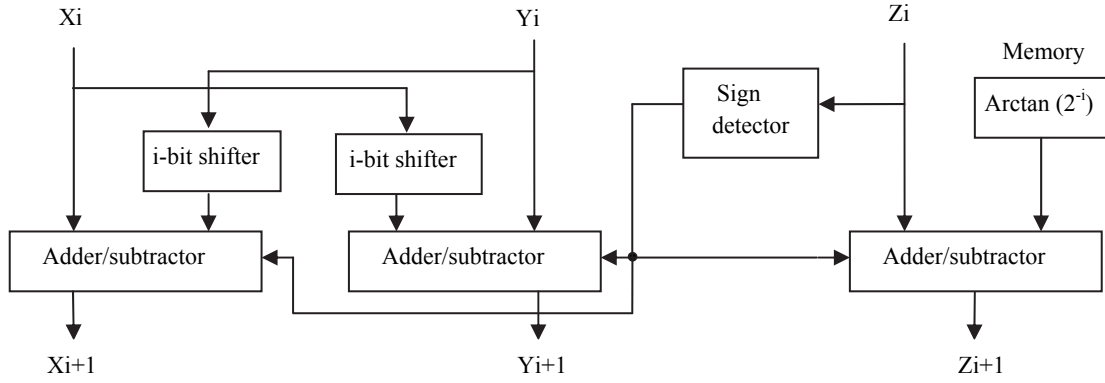


Fig. 1. Basic CORDIC processor structure

and CLA. CSA has been used in X/Y- datapath whereas, Z- datapath uses CLA. The constant arctangent values has been exploited in the design of CLA to reduce area of CLA along with increasing the speed in each iteration. The proposed architecture is implemented in Xilinx FPGA as well as 180 nm standard cell based technology library in Synopsys DC. It has 34% delay improvement and 7% area improvement (excluding ROM) in standard cell based implementation and 39% delay improvement and 20% area improvement (in terms of slices) in FPGA implementation as compared to basic CORDIC design.

The remainder of the paper is as follows. Optimised implementation of CLA exploiting constant arctangent value is described in section II. Section III contains the proposed architecture of CORDIC using optimised CLA in Z-datapath and CSA in X/Y datapath along with implementations. Section IV concludes the paper.

II. EFFICIENT IMPLEMENTATION OF CLA FOR Z-DATAPATH IN CORDIC

Adder/subtractor is the computational unit in Z-datapath circuit. The next iteration can begin only when sign of Z has been determined (Figure 1) and hence in order to have

high performance, adder/subtractor should be faster. Figure 2 shows the 2'S complement adder/subtractor circuit with n-bit inputs A and B. C_i is the initial carry input and C_o is the final carry output. To perform the subtraction using adder circuit, 2'S complement of subtrahend is taken that results in optimum design of adder/subtractor. Delay in multi-bit circuit arises because of carry propagation from one stage of full adder to the next stage. Hence, CLA is used to optimise the delay. In case of CLA, carry required to compute the full addition in successive stages are available in parallel. The negative part of CLA is that it requires extra hardware to compute the carry. Here, we have optimised the area (reduced number of gates) by exploiting the constant arctangent value as another input in the CLA.

For the n-bit inputs, carry of the $(i+1)$ th stage is given by [21],

$$C_{i+1} = A_i \cdot B_i + (A_i \oplus B_i) \cdot C_i \quad (3)$$

By using this, carry sum bit at i th position can be computed as,

$$S_i = (A_i \oplus B_i) \oplus C_i \quad (4)$$

Let,

$$G_i = A_i \cdot B_i \quad (5)$$

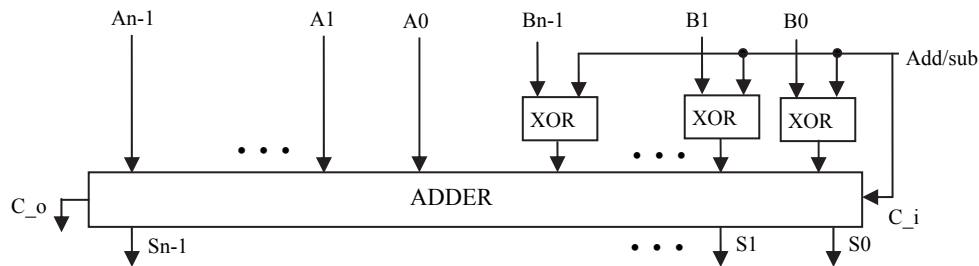


Fig. 2. 2'S complement adder/subtractor

and

$$P_i = A_i \oplus B_i \quad (6)$$

then carry for the each successive stages are expressed by,

$$\begin{aligned} C_1 &= G_0 + P_0 C_0 \\ C_2 &= G_1 + P_1 G_0 + P_1 P_0 C_0 \\ C_3 &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \\ C_4 &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0 \\ &\dots \\ C_{n-1} &= G_{n-2} + \dots + P_{n-2} P_{n-3} \dots P_1 G_0 + P_{n-2} P_{n-3} \dots P_1 P_0 C_0 \end{aligned} \quad (7)$$

For the CLA adder, suppose the one n-bit input z (the initial angle for vectoring mode), expressed in 2'S complement binary representation) is

$$z = (-z_0) + \sum_{i=1}^{n-1} z_i 2^{-i} \quad (8)$$

Since, the second input is a constant and known in advance, AND operation according to (5) can be precomputed and will be 0 or z_i depending on arctangent. Let's take $\tan^{-1} 2^{-i}$ in binary 2'S complement representation. From Table I for $i=0$,

$$B = \tan^{-1}(-1) = 0110010010000111 \quad (9)$$

So from (5), (8) and (9), $G_i = A_i \cdot B_i = z \cdot B$ becomes,

$$\begin{aligned} G_{15} &= 0, G_{14} = z_{14}, G_{13} = z_{13}, G_{12} = 0, G_{11} = 0 \\ G_{10} &= z_{10}, G_9 = 0, G_8 = 0, G_7 = z_7, G_6 = 0 \\ G_5 &= 0, G_4 = 0, G_3 = 0, G_2 = z_2, G_1 = z_1, G_0 = z_0 \end{aligned} \quad (10)$$

From (7) and (10), it can be seen that the terms which have AND operation with 0 will be removed resulting in area savings in CLA. From this, there will not be timing improvement as all partial products are in parallel and the critical-path will be decided by one of those.

From (6) and (9), $P = z \oplus B$ becomes,

$$\begin{aligned} P_{15} &= z_{15}, P_{14} = \overline{z_{14}}, P_{13} = \overline{z_{13}}, P_{12} = z_{12}, P_{11} = z_{11} \\ P_{10} &= z_{10}, P_9 = z_9, P_8 = z_8, P_7 = z_7, P_6 = z_6 \\ P_5 &= z_5, P_4 = z_4, P_3 = z_3, P_2 = z_2, P_1 = z_1, P_0 = z_0 \end{aligned} \quad (11)$$

From (11), it is evident that XOR gate is completely removed from all the bits, i.e., bits of P becomes either invert of Z or same as Z . In the Z -datapath there are two levels of XOR gates in adder/subtractor (one level before the

ADDER architecture and another inside it) and consequently both of them gets eliminated by the precomputation. Hence, delay as well as area improvement.

TABLE I
VALUES OF ARCTANGENT IN BINARY 2'S COMPLEMENT 16-BIT FRACTIONAL REPRESENTATION

| Arctangent | Binary 2's complement fraction |
|----------------------|--------------------------------|
| $\tan^{-1}(2^0)$ | 0110010010000111 |
| $\tan^{-1}(2^{-1})$ | 0011101101011000 |
| $\tan^{-1}(2^{-2})$ | 0001111101011011 |
| $\tan^{-1}(2^{-3})$ | 0000111111101010 |
| $\tan^{-1}(2^{-4})$ | 0000011111111101 |
| $\tan^{-1}(2^{-5})$ | 0000001111111111 |
| $\tan^{-1}(2^{-6})$ | 0000000111111111 |
| $\tan^{-1}(2^{-7})$ | 0000000011111111 |
| $\tan^{-1}(2^{-8})$ | 0000000011111111 |
| $\tan^{-1}(2^{-9})$ | 0000000011111111 |
| $\tan^{-1}(2^{-10})$ | 0000000001111111 |
| $\tan^{-1}(2^{-11})$ | 0000000000111111 |

We have implemented conventional CLA as well as our proposed CLA design for CORDIC computation in Xilinx FPGA and Synopsys DC using TSMC 180nm standard cell technology library (ASIC). Table II shows the comparison results in FPGA whereas, Table III shows the comparisons in standard cell library. While FPGA implementation has large area savings, it has small amount of delay improvement whereas, standard cell based technology library has large delay improvement (about 61%) but small area savings.

TABLE II
COMPARISON RESULTS FOR CONVENTIONAL CLA AND PROPOSED CLA DESIGN IMPLEMENTED IN XILINX FPGA

| FPGA Kit Xilinx XC2VP30 | | |
|----------------------------|------------------|-------------------------|
| | Conventional CLA | Proposed CLA for CORDIC |
| # of slices | 67 | 26 |
| # of 4 i/p LUTs | 118 | 48 |
| Max Delay | 13.16 ns | 10.64 ns |

TABLE III
COMPARISON RESULTS FOR CONVENTIONAL CLA AND PROPOSED CLA DESIGN IMPLEMENTED IN SYNOPSYS DC

| Standard Cell Library TSMC 180nm | | |
|-------------------------------------|------------------------|-------------------------|
| | Conventional CLA | Proposed CLA for CORDIC |
| Area | 2182.1 μm^2 | 1776.2 μm^2 |
| Max Delay | 3.02 ns | 1.16 ns |
| Power | 0.982 μW | 0.368 μW |

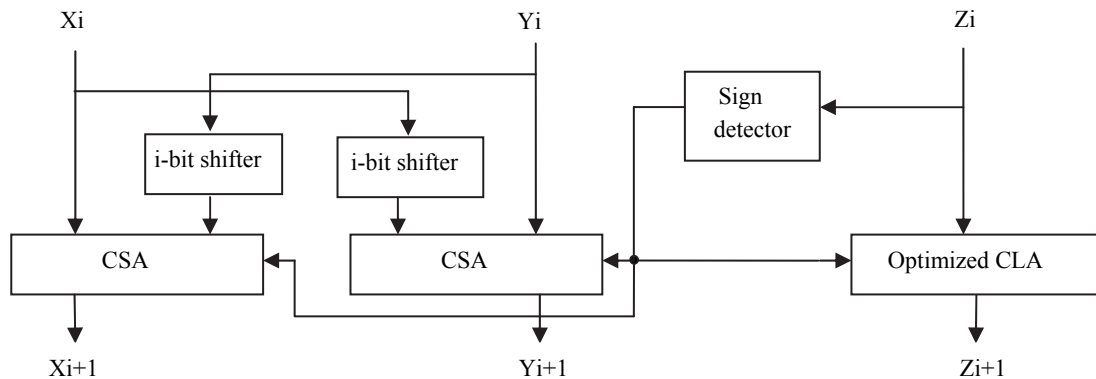


Fig. 3. ROM free regular CORDIC processor structure using proposed CLA and CSA

III. CORDIC ARCHITECTURE USING OPTIMISED CLA IN ANGLE DATAPATH

Since, CORDIC is an iterative structure, it takes longer time to compute an algorithm. High throughput design for CORDIC is done by inserting pipeline registers between iterative structures. Pipeline structure of CORDIC is applicable in places where different angles have to be computed in a row (e.g., in sine and cosine wave generations). For the applications requiring one input to be processed at a time (e.g., in calculator) pipeline registers has no use and final results can be obtained in a single clock cycle using parallel structure. Overall delay of the parallel/pipeline structure depends on the adder/subtractor module in their datapath. Hence fast adder design is important to improve the speed. As explained in the previous section, use of CLA in z-datapath can improve the speed as well as reduce the area along with complete removal of ROM from the z-datapath. Once angle z-datapath is implemented in CLA, CSA can be used in X/Y-datapath because CSA has no carry propagation from one full adder to next and the overall delay depends upon the iterative stages used for the desired accuracy (i.e., each iterative stage has one full adder delay). Figure 3 shows the ROM free regular CORDIC architecture using proposed adders.

ROM free CORDIC architecture using proposed CLA and CSA is implemented in FPGA and 180 nm ASIC standard cell based library. For the comparison, conventional CORDIC is also implemented using CLA and CSA. In ASIC implementation, ROM is assumed to be connected externally whereas, for FPGA, internal ROM has

been used. Table IV shows the FPGA implementation comparison and Table V shows the comparisons in ASIC.

Total 13 parallel iterative structures have been used with 18-bit fractional precisions in adders inputs. 39% delay improvement in FPGA and 34% improvement in ASIC has been obtained. Also, the proposed design has about 47% power savings in ASIC apart from power consumption in ROM. Power savings in logic is achieved by removal of switching activities with pre-computation. Since, the amount of hardware requirements depends on the iteration steps as well as bit-width of datapath, basic design has been used for the comparison purpose.

TABLE IV
COMPARISON RESULTS FOR CONVENTIONAL CORDIC AND PROPOSED USING CLA AND CSA IMPLEMENTED IN XILINX FPGA

| FPGA Kit Xilinx XC2VP30 | | | |
|----------------------------|--------------|----------|-------------|
| | Conventional | Proposed | improvement |
| # of slices | 1305 | 1044 | 20% |
| # of 4 i/p LUTs | 2290 | 1834 | - |
| Max Delay | 123.4 ns | 74.9 ns | 39% |

TABLE V
COMPARISON RESULTS FOR CONVENTIONAL CORDIC AND PROPOSED USING CLA AND CSA IMPLEMENTED IN SYNOPSIS DC

| Standard Cell Library TSMC 180nm (ROM is assumed to be external) | | | |
|---|-----------------------|-------------------------|----------------------|
| | Conventional | Proposed | improvement |
| Area | 95401 μm^2 | 88578.7 μm^2 | 7% + ROM(18x13-bits) |
| Max Delay | 62.6 ns | 40.82 ns | 34% |
| Power | 165.0 mW | 86.7 mW | 47.4% |

IV. CONCLUSIONS

We have proposed a ROM free based structure of CORDIC using CSA and CLA. By exploiting the arctangent constants, optimized CLA has been designed to improve the speed as well as area and power savings. The proposed design approach maintains the regularity of basic CORDIC structure. The CORDIC structure is implemented in Xilinx FPGA as well as 180 nm standard cell technology using proposed CLA in angle adder/subtractor datapath and CSA in coordinates adder/subtractor datapath. The proposed implementation has about 39% delay improvement in FPGA and about 34% delay improvement in standard cell technology as compared to conventional CORDIC structure. About 47% power savings is achieved in proposed structure.

ACKNOWLEDGMENT

The authors acknowledge to DIT (Ministry of Information & Communication Technology) for the financial support for carrying out this research work.

REFERENCES

- [1] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol. EC-8, pp. 330–334, Sept. 1959.
- [2] J. S. Walther, "A unified algorithm for elementary functions," *Joint Spring Computer Conference Proceedings*, vol. 38, pp. 379–385, Jul. 1971.
- [3] P. K. Meher, J. Valls, T. B. Juang, K. Sridharan and K. Maharatna, "50 Years of CORDIC: Algorithms, Architectures, and Applications," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, Vol. 56(9), pp.1893-1907, Sept. 2009.
- [4] H. Jeong, J. Kim, and W. K. Cho, "Low-power multiplierless DCT architecture using image data correlation," *IEEE Transactions on Consumer Electronics*, 50 (1), pp. 262–26, Feb. 2004.
- [5] Cheng-Ying Yu, Sau-Gee Chen, and Jen-Chuan Chih, "Efficient CORDIC Designs for Multi-Mode OFDM FFT," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 1036-1039, May 2006.
- [6] B. Das and S. Banerjee, "Unified CORDIC-based chip to realise DFT/DHT/DCT/DST," *IEE Computers and Digital Techniques, Proceedings*, Vol. 149(4), pp.121-127, Jul 2002.
- [7] C.-C. Sun, S.-J. Ruan, B. Heyne and J. Goetze, "Low-power and high-quality Cordic-based Loeffler DCT for signal processing," *IET Circuits, Devices & Systems*, Vol.1(6), pp.453-461, Dec.2007.
- [8] Jue-Hsuan Hsiao, Liang-Gee Chen, Tzi-Dar Chiueh, and Chun-Te Chen "High Throughput CORDIC-Based Systolic Array Design for the Discrete Cosine Transform," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 5(3), June 1995.
- [9] S. Wang, V. Piuri, Wartzlander, Jr. E. E., "Hybrid CORDIC algorithms," *IEEE Transactions on Computers*, Vol. 46(11), pp.1202-1207, Nov. 1997.
- [10] T. B. Juang, S. F. Hsiao and M. Y. Tsai, "Para-CORDIC:Parallel CORDIC Rotation Algorithm," *IEEE Transactions on Circuits and System I*, vol. 51(8), pp. 1515–1524, Aug. 2004.
- [11] Y. H. Hu, "CORDIC-based VLSI Architectures for Digital Signal Processing," *IEEE Signal Processing Magazine*, pp.16-35, Jul 1992.
- [12] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modified virtually scaling free adaptive CORDIC rotator algorithm and architecture," *IEEE Transactions on Circuits and Systems for Video Technoogy*, vol. 15(11), pp. 1463–1474, Nov. 2005.
- [13] N. Takagi, T. Asada and S. Yajima, "Redundant CORDIC Methods with a Constant Scale Factor for Sine and Cosine Computation," *IEEE Transactions on Computers*, Vol. 40(9), pp.989-995, Sept. 1991.
- [14] D. Timmermann, H. Hahn and B. J. Hosticka, "Low Latency Time CORDIC Algorithms," *IEEE Transactions on Computers*, Vol. 41(8), Aug. 1992.
- [15] B. Lakshmi, and A. S. Dhar, "FPGA Implementation of a High Speed VLSI Architecture for CORDIC," *IEEE TENCON 2009*.
- [16] R. Kunemund, H. Soldner, S. Wohlleben, T. Noll, "CORDIC Processor with Carry-Save Architecture," *Sixteenth European Solid-State Circuits Conference, 1990. ESSCIRC '90*, pp.193-196, 19-21 Sept. 1990.
- [17] K. S. Yeo and K. Roy, *Low-Voltage, Low-Power VLSI Subsystem*, Tata Mcgraw-Hill, New Delhi, 2009.
- [18] E. O. Garcia, R. Cumplido and Miguel Arias, "Pipelined CORDIC Design on FPGA for a Digital Sine and Cosine Waves Generator," *2006 3rd International Conference on Electrical and Electronics Engineering(ICEEE)*, pp.104-107, Sept.2006.
- [19] Chua-Chin Wang, Chia-Hao Hsu, Tuo-Yu Yao and Jian-Ming Huang, "A ROM-less DDFS Using A Nonlinear DAC With An Error Compensation Current Array," *IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2008*, pp.1632-1635, Nov. 30 2008- Dec. 3 2008.
- [20] Chua-Chin Wang, Jian-Ming Huang, Y.-L. Tseng, Wun-Ji Lin and Ron Hu, "Phase-Adjustable Pipelining ROM-Less Direct Digital Frequency Synthesizer With a 41.66-MHz Output Frequency," *IEEE Transactions on Circuits and Systems—II: Vol.53(10)*, pp.1143-1147, Oct. 2006.
- [21] C. H. Roth Jr and L. K. John, *Principles of Digital Systems Design using VHDL*, Cengage Learning, New Delhi, 2008.