International Journal of Innovative Research in Electronics and Communications (IJIREC)
Volume 2, Issue 1, January 2015, PP 1-7
ISSN 2349-4042 (Print) & ISSN 2349-4050 (Online)
www.arcjournals.org

Realization of Fixed Angle Rotation for Co-Ordinate Rotation Digital Computer

G. Renuka[1], M. Sandhya Rani[2]

[1]PG Scholar, Electronics and Communication Engineering, Intell Engineering College, Anantapuramu. AP, India
[2]Associate Professor, Electronics and Communication Engineering, Intell Engineering College, Anantapuramu, AP, India

Abstract: *Rotation of vectors through fixed and known angles has wide applications in robotics, digital signal processing, graphics, games, and animation. But, we do not find any optimized coordinate rotation digital computer (CORDIC) design for vector-rotation through specific angles. Therefore, in this paper, we present optimization schemes and CORDIC circuits for fixed and known rotations with different levels of accuracy. For reducing the area- and time-complexities, we have proposed a hardwired pre-shifting scheme in barrel-shifters of the proposed circuits. Two dedicated CORDIC cells are proposed for the fixed-angle rotations. In one of those cells, micro-rotations and scaling are interleaved, and in the other they are implemented in two separate stages. Pipelined schemes are suggested further for cascading dedicated single-rotation CORDIC units for high-throughput and reduced latency implementations. We have the optimized set of micro-rotations for fixed and known angles. The derived optimized scale-factors from a set of micro rotations and dedicated shift-add circuits are used to implement the scaling. We have synthesized the proposed CORDIC cells using Xilinx field programmable gate array platform and shown that the proposed design offer higher throughput and less area-delay product than the reference CORDIC design for fixed and known angles of rotation.*

Keywords: *Coordinate rotation digital computer (CORDIC), numerically controlled oscillator (NCO).*

1. INTRODUCTION

CORDIC stands for COordinate Rotation DIgital Computer. The key concept of CORDIC arithmetic is based on the simple and ancient principles of 2-D geometry. But the iterative formulation of a computational algorithm for its implementation was first described in 1959 by Volder [1], [2] for the computation of trigonometric functions, multiplication, and division. Not only a wide variety of applications of CORDIC have been suggested over the time, but also a lot of progress has taken place in the area of algorithm design and development of architectures for high-performance and lowcost hardware solutions [3]–[12]. Rotation of vectors through a fixed and known angle has wide applications in robotics, graphics, games, and animation [4], [13], [14]. Locomotion of robots is very often performed by successive rotations through small fixed angles and translations of the links.

The translation operations are realized by simple additions of coordinate values while the new coordinates of a rotational step could be accomplished by suitable successive rotations through a small fixed angle which could be performed by a CORDIC circuit for fixed rotation [4]. Similarly, interpolation of orientations between key-frames in computer graphics and animation could be performed by fixed CORDIC rotations [14]. There are plenty of examples of uniform rotation starting from electrons inside an atom to the planets and satellites. A simple example of uniform rotations is the hands of an animated mechanical clock which perform one degree rotation each time. There are several cases where high-speed constant rotation is required in games, graphic, and animation.

The objects with constant rotations are very often used in simulation, modeling, games, and animation. Efficient implementation of rotation through a known small angle to be used in these areas could be implemented efficiently by simple and dedicated CORDIC circuits. Similarly, the multiplication of complex number with a known complex constant (which is the same as the rotation of vectors through a fixed and known angle) is often encountered in communication,

signal processing and many other scientific and engineering applications. In some early works, CORDIC circuits have been developed for the implementation of complex multiplications to be used for digital signal processing (DSP) applications [16]–[18], but we do not find any detailed study pertaining to efficient CORDIC realization of fixed and known angle rotations and constant complex multiplication.

Latency of computation is the major issue with the implementation of CORDIC algorithm due to its linear-rate convergence [19]. It requires (*n+1*) iterations to have –bit precision of the output. Overall latency of computation increases linearly with the product of the word-length and the CORDIC iteration period. The speed of CORDIC operations is, therefore, constrained either by the precision requirement (iteration count) or the duration of the clock period. The angle recoding (AR) schemes [5]–[9] could be applied for reducing the iteration count for CORDIC implementation of constant complex multiplications by encoding the angle of rotation as a linear combination of a set of selected elementary angles of micro-rotations. In the conventional CORDIC, any given rotation angle is expressed as a linear combination of *n* values of elementary angles that belong to the set {(arctan (2-r)) : {- 1, 1}, r {1, 2, . . . , *n*-1}} to obtain an –bit value of arctan(2-i)]. However, in AR methods, this constraint is relaxed by adding zero into the linear combination to obtain the desired angle using relatively fewer terms of the form (arctan 2-*r*) for {1, 0, -1}. The elementary-angle-set (EAS) used by AR scheme is given by SEAS = {(arctan2-*r*) : }. Hu and Naganathan [5] have proposed an AR method based on the greedy algorithm that tries to represent the remaining angle using the closest elementary angle.

2. Optimization of Elementary Angle Set

The rotation-mode CORDIC algorithm to rotate a vector U =[U_x U_y]^T through an angle $\emptyset$ to obtain a rotated vector V = [V_x V_y]

is given by [1], [2]

$$(U_x)_{i+1} = (U_x)_i - \sigma i \cdot (U_y)_I \cdot 2^{-I} \quad (1a)$$

$$(U_y)_{i+1} = (U_y)_I + \sigma i \cdot (U_x)_I \cdot 2^{-I} \quad (1b)$$

$$\emptyset_{i+1} = \emptyset_I - \sigma i \tan^{-1}(2^{-i}) \quad (1c)$$

Such that when is sufficiently large

$$\begin{bmatrix} Vx \\ Vy \\ \emptyset \end{bmatrix} \leftarrow T \begin{bmatrix} (Ux)n \\ (Uy)n \\ 0 \end{bmatrix} \quad (1d)$$

Where σi = -1 if $\emptyset i < 0$ and $\sigma i = 1$ otherwise, and T is the scale-factor of the CORDIC algorithm, given by

$$T = \pi_{i=0}^{n-1}[_{1+2} - 2_{ij}\text{-}1/2 \quad (2)$$

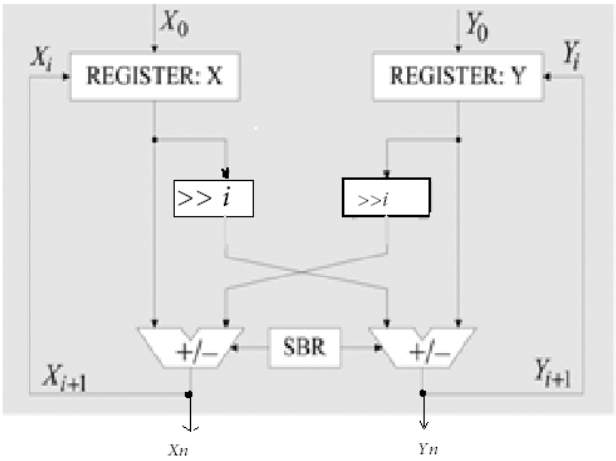

Fig1. *Reference CORDIC circuit for fixed rotations*

In case of fixed rotation, $\emptyset_i$ could be pre-computed and the sign-bits corresponding to σ_i could be stored in a sign-bit register (SBR) in CORDIC circuit. The CORDIC circuit therefore need not compute the remaining angle $\emptyset_I$ during the CORDIC iterations [3]. A reference CORDIC circuit for fixed rotations according to (1a) and (1b) is shown in Fig. 1. X_0 and Y_0 are fed as set/reset input to the pair of input registers and the successive feedback values X_i and Y_i at the ith iteration are fed in parallel to the input registers. Note that conventionally we feed the pair of input registers with the initial values X_0 and Y_0 as well as the feedback values X_i and Y_i through a pair of multiplexers.

3. IMPLEMENTATION OF MICRO ROTATIONS

Since the elementary angles and direction of micro-rotations are predetermined for the given angle of rotation, the angle estimation data-path is not required in the CORDIC circuit for fixed and known rotations. Moreover, because only a few elementary angles are involved in this case, the corresponding control- bits could be stored in a ROM of few words.

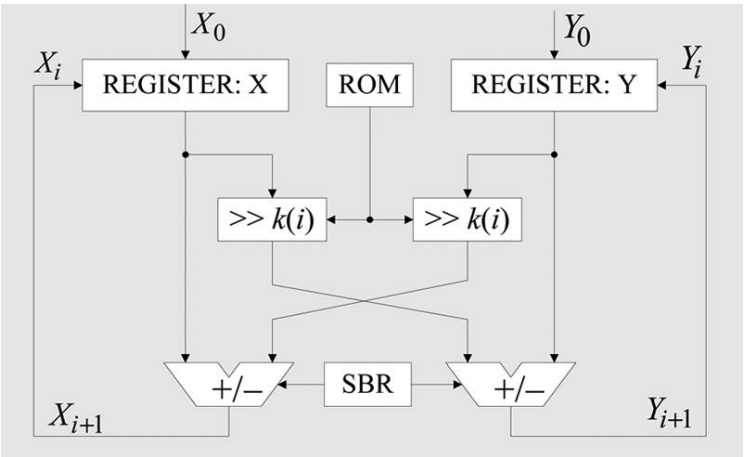

Fig2. *CORDIC cell for constant complex multiplications*

A CORDIC circuit for complex constant multiplications is shown in Fig. 2. The ROM contains the control-bits for the number of shifts corresponding the micro-rotations to be implemented by the barrel-shifter and the directions of micro rotations are stored in the sign-bit register (SBR). The major contributors to the hardware-complexity in the implementation of a CORDIC circuit are the barrel-shifters and the adders. There are several options for the implementation of adders [22], from which a designer can always choose depending on the constraints and requirements of the application. But, we have some scope to develop techniques for reducing the complexity of barrel-shifters over the conventional designs as discussed in the followings.

3.1. Minimization of Barrel-Shifter Complexity by Hardwired Pre-Shifting

A barrel-shifter for maximum of *S* shifts for word-length *L* is implemented by [log2(*S* +1)]-stages of de-multiplexors, where each stage requires number of 1:2 line MUXes. The hardware-complexity of barrel-shifter, therefore, increases linearly with the wordlength and logarithmically with the maximum number of shifts.

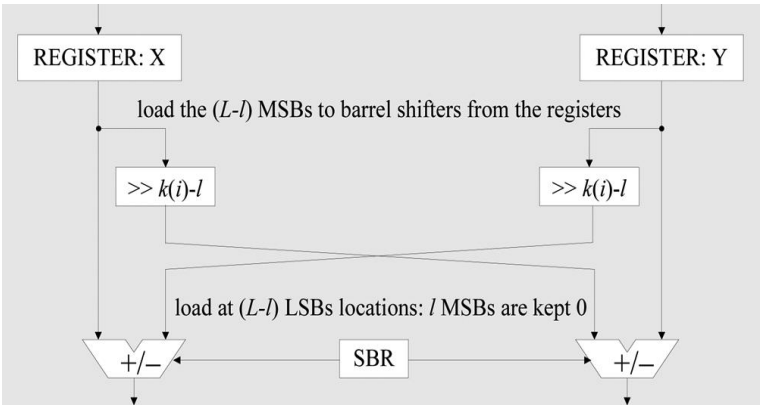

Fig3. *Hardwired pre-shifting in basic CORDIC module*

We can reduce the effective word-length in the MUXes of the barrelshifters, and so also the number of stages of MUXes by simple hardwired pre-shifting as shown in Fig. 3. If is the minimum number of shifts in the set of selected micro-rotations, we can load only the (L -l) more-significant bits (MSBs) of an input word from the registers to the barrel-shifters, since the less significant bits (LSBs) would get truncated during shifting. The barrel-shifter, therefore, needs to implement a maximum of (s - l) shifts only, where is the maximum number of shifts in the set of selected micro-rotations. The output of the barrel-shifters are loaded as the (L - l) LSBs to the add/subtract units, and the MSBs of the corresponding operand of add/subtract unit are hardwired to 0. Therefore, the hardware-complexity of a barrel-shifter could be reduced by the hardwired pre-shifting approach. The time involved in a barrel-shifter could also be reduced by hardwired preshifting, since the delay of the barrel-shifter is proportional to the number of stages of MUXes, and it also be possible to reduce the number stages by hardwired pre-shifting.

3.2. Bi-Rotation CORDIC Cell

We find that using only two micro-rotations, it is possible to get an accuracy up to 0.033 radian. Although the accuracy achieved by two micro-rotations is inadequate in many situations, but can be used for some applications where the outputs are quantized, e.g., in case of speech and image compression, etc., [23], [24]. Besides, the rotations with four and six micro-rotations can also be implemented successively by two and three pairs of micro-rotations, respectively. Therefore, we design an efficient CORDIC circuit to implement a pair of micro-rotations and named as "*bi-rotation* CORDIC". The proposed circuit for bi-rotation CORDIC is shown in Fig. 4. It consists of an adder module, two 2:1 multiplexers and a sign-bit register (SBR) of two-bit size. The adder module consists of a pair of adders/subtractors. The adders/subtractors perform additions or subtractions according to the sign-bit available from the SBR. The components of the input vector (real and the imaginary parts of the input complex operand) are loaded to the input-registers through set/reset input. The output of the registers are sent in two lines where the content of the register is fed to one of the adders/subtractors directly while that in the other line is loaded to the barrel shifter pre-shifted by $k(0)$ bit-locations to right by hardwired pre-shifting technique. The outputs of the adders are loaded back to the input registers for the second CORDIC iteration. The bi-rotation CORDIC involves only a pair of barrel-shifters consisting of only one stage of 2:1 MUXes. The control-bit for the barrel-shifters is 0 for the first micro rotation (no shift) and 1 for the second micro rotation (shift through $k(1) - k(0)$). The control bits are generated by a T flip-flop, since they are 1 and 0 in each alternate cycle.

3.3. Cascaded CORDIC with Single-Rotation Cells

A multi-stage-cascaded pipelined-CORDIC circuit consisting of single-rotation modules is shown in Fig. 5. Each stage of the cascaded design consists of a dedicated rotation-module that performs a specific micro-rotation. The structure and function of a rotation-module is depicted in Fig. 5(b). Each rotation-module consists of a pair of adders or subtractors (depending on the direction of micro-rotation which it is required to implement). Each of the adders/subtractors loads one of the pair of inputs directly and loads the other input in a pre-shifted format ($L - s(i)$) LSB locations, where $s(i)$ is the number of right-shifts required to be performed to implement the th micro-rotation. The $s(i)$ MSB locations are hardwired to be zero. The rotation-module in this case does not require input from SBR

Since each adder module always performs either addition or subtraction. It also does not require barrel-shifter since it has to implement only one fixed micro-rotation. The output of each stage is latched to the input of its succeeding stage as shown in the figure. The critical-path in this case amounts to only one addition/subtraction operation in the adder module. Total latency of -stage single-rotation cascade amounts to $n(TA + TFF)$, where TA and TFF, are the addition/subtraction time and D flip-flop delay, respectively. We find that in more than two-third of the rotation angles only three micro-rotations are adequate to have the maximum deviation of up to 0.04o. The complex multiplications involving three such micro-rotations could be implemented by three-stage-cascaded CORDIC circuit shown in Fig. 5 (for $n = 3$). The rotation using 4 and 6 micro-rotations, similarly, would require 4 and 6 stages of rotation module for pipelined implementation. This can also be implemented in no pipelined form using (n -1) carry-propagate adders with total latency of TA+ ((i)) TFA, where TA and TFA, are, respectively, the time required for L-bit

addition-time and full adder delay, L being word-length of implementation. $k(i)$ is the number of shifts of the ith stages.

3.4. Cascaded CORDIC with Bi-Rotation Cells

For reduction of adder complexity over the cascaded single-rotation CORDIC, the microrotations could be implemented by a cascaded bi-rotation CORDIC circuit. A two-stage cascaded bi-rotation CORDIC is shown in Fig 6. The first two of the micro-rotations as shown in Table I out of the four-optimized micro-rotations could be implemented by stage-1, while the rest two are performed by stage-2. The structure and function of the birotation CORDIC is shown in Fig. 4. For implementing six selected micro-rotations, we can use a three-stage-cascade of bi-rotation CORDIC cells. The three-stage bi-rotation cells could however be extended further when higher accuracy is required.

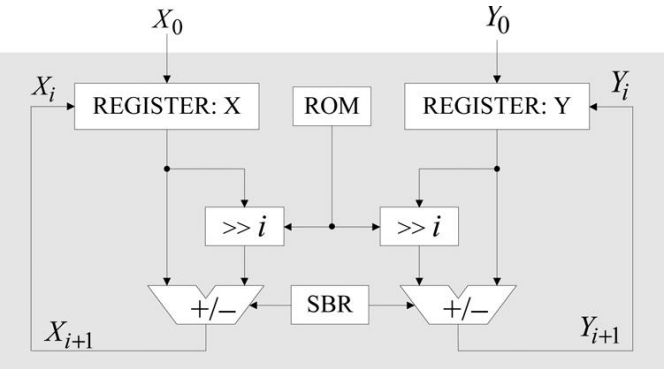

Fig4. *Shift-add scaling circuit using hardwired pre-shifted loading.*

4. SCALING OPTIMIZATION AND IMPLEMENTATION

We discuss here the optimization of scaling to match with the optimized set of elementary angles for the micro-rotations.

4.1. Calling Approximation for Fixed Rotation

The shift-add circuit for scaling according to (7) is shown in Fig. 4. The scaling circuit of Fig. 4 can use hardwired pre-shifting for minimizing barrel-shifter complexity and could be placed after the CORDIC cell of Fig. 2 to perform micro-rotation and scaling in two separate stages. The generalized CORDIC circuit for fixed rotation to perform the microrotation and the scaling in interleaved manner in alternate cycles is shown in Fig. 4. The circuit of Fig. 8 is similar to that of Fig. 2. It involves only an additional line-changer circuit to change the path of un shifted (direct) input. The structure and function of line changer is shown in Fig. 4. The line changer is placed on the un shifted input data line to keep the critical path the same as that of Fig. 2.

The generalized expression for the scale-factor given by (2) can be expressed explicitly for the selected set of m1micro-rotations as $K = 2\text{-}2k(i))]\text{-}1/2$ (4)]

Simulation results for the known fixed angle of +/- 90^0 for fixed rotations.

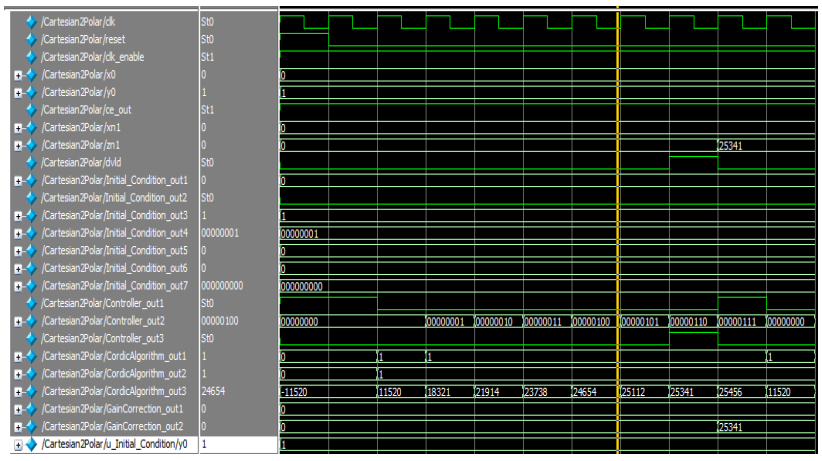

Fig5. *Simulation Results*

5. CONCLUSION

The number of micro-rotations for rotation of vectors through known and fixed angles is optimized and several possible dedicated circuits are explored for rotation mode CORDIC processing with different levels of accuracy. The proposed CORDIC cell with interleaved scaling involves 4% more area, but offers 43% more throughput and involves nearly less latency and less ADP, than the reference design for known and fixed rotations. The proposed single-rotation cascade and birotation cascade require, respectively, and times more area over the reference design, but offer nearly 16.3 and 7.0 times more throughput, and involve nearly 4.6 and 2.5 times less ADP with nearly half and two-thirds of the latency of the other. With progressing scaling trends, since the silicon area is getting continually cheaper, it appears to be a good idea to use the cascaded designs for their potential for high-throughput and low-latency implementation. It is found that higher accuracy could be achieved in case of smaller angles of rotation when the same number of micro-rotations is used. The small angle rotators could therefore be very much useful for shape design and curve tracing for animation and gaming devices. The fixedangle CORDIC rotation would have wide applications in signal processing, games, animation, graphics and robotics, as well.

FUTURE SCOPE

CORDIC Algorithm can be extended to cover 3D (Three Dimensional) Co-Ordinate Space as well as other Digital Signal Processing Applications**.**

REFERENCES

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

[2] J. S. Walther, "A unified algorithm for elementary functions," in Proc. 38th Spring Joint Comput. Conf., 1971, pp. 379–385.

[3] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing," IEEE Signal Process. Mag., vol. 9, no. 3, pp. 16–35, Jul. 1992.

[4] P. K. Meher, J. Valls, 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.

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

[6] Y. H. Hu andH.H.M.Chern, "Anovel implementation of CORDIC algorithmusing backward angle recoding (BAR)," IEEE Trans.Comput., vol. 45, no. 12, pp. 1370–1378, Dec. 1996

[7] C.-S.Wu, A.-Y.Wu, andC.-H. Lin, "A high-performance/lowlatency 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.

[8] T.-B. Juang, S.-F. Hsiao, and M.-Y. Tsai, "Para-CORDIC: Parallel cordic rotation algorithm," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 8, pp. 1515–1524, Aug. 2004.

[9] T. K. Rodrigues and E. E. Swartzlander, "Adaptive CORDIC: Using parallel angle recoding to accelerate CORDIC rotations," in Proc. 40th Asilomar Conf. Signals, Syst. Comput. (ACSSC), 2006, pp. 323–327.

[10] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modified virtually scaling free adaptive CORDIC rotator algorithm and architecture," IEEE Trans. Circuits Syst. For Video Technol., vol. 15, no. 11, pp. 1463–1474, Nov. 2005.

[11] C. Y. Kang and E. E. Swartzlander, Jr., "Digit-pipelined direct digital frequency synthesis based on differential CORDIC," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 5, pp. 1035–1044, May 2006.

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

AUTHORS' BIOGRAPHY

Ms. G. Renuka is PG Scholar in VLSI System Design at Intell Engineering College, Anantapuramu. She completed her Bachelor Degree of Technology in Intell Engineering College, Anantapuramu.

Mrs. M. Sandhya Rani, M.Tech.,(Ph.D) working as Associate Professor, ECE Department, in Intell Engineering College, Anantapuramu.