ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 7, July 2013

Radix-2 CORDIC Method with Constant Scale Factor

J. M. Rudagi[1], Vinayak Dalavi[2]

Associate Professor, Dept of ECE, KLEDRMSSCET, Belgaum, Karnataka, India [1]

PG Student [VLSI], Dept of ECE, KLEDRMSSCET, Belgaum, Karnataka, India [2]

ABSTRACT*:* There are various simple hardware-efficient algorithms exist which can be used to increase speed while performing the desired signal processing tasks. One such simple and hardware-efficient algorithm is CORDIC which uses only Shift-and-Add arithmetic with table Look-Up to implement different functions. It can be used to efficiently implement Trigonometric and other functions. In this paper we present the conventional unrolled CORDIC architecture. The processor is designed using Verilog HDL using a structured coding method, simulated using ISIM simulator and implemented using Xilinx 14.2 FPGA synthesis Tool for 16 and 32 bit conventional radix-2 CORDIC architectures. The output of the CORDIC architectures are analyzed and verified, and compared with the actual values obtained from MATLAB.

Keywords: CORDIC, Cosine, Sine, Unrolled Architecture, Verilog

I.INTRODUCTION

Most of the engineers tasked with implementing a mathematical function such as sine, cosine or square root within an FPGA may initially think of doing so by means of a lookup table (LUT), possibly combined with linear interpolation or a power series if multipliers are available. LUTs are the fastest way to make the computation; but the precision of the result is directly related to size of the look-up table. The use of power series is slow to converge to a desired precision. In effect, the look-up table size is being traded off at the expense of computation time.

CORDIC [1], [2], [3] method for calculating these elementary functions, is a compromise between the two methods described above wherein the precision is preserved without any considerable memory requirement. The use of the architectures in modern DSP systems [4], [5] requires a rapid increase in performance accompanied by a decrease in cost and time-to market. Higher performance is achieved by optimizing these structures for improved timing behaviour and low power consumption. FPGA provides the hardware environment in which dedicated processors can be tested for their functionality. They perform various high-speed operations that cannot be realized by a simple microprocessor. The primary advantage that FPGA offers is On-site programmability. Thus, it forms the ideal platform to implement and test the functional of a dedicated processor designed using CORDIC algorithm.

The rest of the paper is organized in the following manner. CORDIC algorithm and its operating modes are discussed in section 2. Section 3 describes the unrolled architecture of the CORDIC algorithm. Section 4 describes the result and related comparison.

II. CORDIC ALGORITHM

The COordinate Rotation DIgital Computer (CORDIC) is known as an iterative algorithm using only shift-and-add operations to perform several mathematic functions for scientific and engineering fields. CORDIC was firstly described in 1959 by J.E. Volder [1] to evaluate trigonometric functions. In 1971, J. Walther [2] extended the CORDIC algorithm to hyperbolic functions and the algorithm is today used in many application areas such as matrix computation, digital signal processing, digital image processing, communication, robotics and graphics. The trigonometric and exponential functions that are evaluated via rotations in the circular, hyperbolic and linear coordinate systems. Their inverses can be implemented in a “vectoring” mode in the appropriate coordinate system.
Rotating a vector in a Cartesian plane by the angle θ this can be arranged so that

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 7, July 2013

$$x' = \cos\theta\,[\,x - y\tan\theta\,] \quad (1)$$
$$y' = \cos\theta\,[\,y + x\tan\theta\,] \quad (2)$$

If the rotation angles are restricted so that $\tan(\theta) = \pm 2^{-i}$, the multiplication by the tangent term is reduced to a simple shift operation. Arbitrary angles of rotation are obtainable by performing a series of successively smaller elementary rotations. If the decision at each iterations i, is which direction to relate rather than whether or not to rotate, then the cos(θ) term becomes a constant .The iterative rotation can now be expressed as:

$$x_{i+1} = K_i\,[\,x_i + d_i \,.\, 2^{-i} \,.\, y_i\,] \quad (3)$$
$$y_{i+1} = K_i\,[y_i - d_i \,.\, 2^{-i} \,.\, x_i] \quad (4)$$

Where,
$K_i = 1/(1+2^{-2i})^{1/2}$; known as scale constant.
$d_i = \pm 1$; known as decision function.

Removing the scaling constant from the iterative equations yields a shift-add algorithm for vector rotation. The product of the K can be applied elsewhere in the system or treated as part of a system processing gain or by initiating the rotating vector by the reciprocal of the gain of a certain number of iterations. The angle of a composite rotation is uniquely defined by the sequence of the directions of the elementary rotations. That sequence can be represented by a decision vector. The set of all possible decision vectors is an angular measurement system based on binary arctangents. A better conversion method uses an additional adder-subtractor that accumulates the elementary rotation angles at each single iteration. The elementary angles can be expressed in any convenient angular unit. Those angular values are supplied by a small lookup table or are hardwired, depending on the implementation. The angle accumulator adds a third difference equation to the CORDIC algorithm

$$z_{i+1} = z_i - d_i \,.\, \tan^{-1}(2^{-i}) \quad (5)$$

The CORDIC rotator is normally operated in one of two modes, the rotation mode and the vectoring mode [4]. In the rotation mode, a vector (x, y) is rotated by an angle θ. The angle accumulator is initialized with the desired rotation angle θ. The rotation decision per iteration is made to diminish the magnitude of the residual angle in the angle accumulator. The decision per is therefore based on the sign of the residual angle after each step. Naturally, if the input angle is already expressed in the binary arctangent base, the angle accumulator may be eliminated.

For rotation mode the CORDIC equations are

$$X_{i+1} = X_i - Y_i.d_i.2^{-i} \quad (6)$$
$$Y_{i+1} = Y_i + X_i.d_i.2^{-I} \quad (7)$$
$$Z_{i+1} = Z_i - d_i.\tan^{-1}(2^{-i}) \quad (8)$$

Where $d_i=-1$ if $Z_i<0$, else $d_i=+1$

The CORDIC rotator rotates the input vector through whatever angle is necessary to align the result vector with the x-axis. The result of the vectoring operation is a rotation angle and the scaled magnitude of the original vector. The vectoring function works by seeking to minimize the y component of the residual vector at each rotation. The sign of the residual y component is used to determine which direction to rotate next. If the angle accumulator is initialized with zero, it will contain the traversed angle at the end of the iterations.

The CORDIC equations are:

$$X_{i+1} = X_i - Y_i.d_i.2^{-i}$$
$$Y_{i+1} = Y_i + X_i.d_i.2^{-i}$$
$$Z_{i+1} = Z_i - d_i.\tan^{-1}(2^{-i})$$

Where $d_i=-1$ if $Z_i<0$, else $d_i=+1$

After n iterations we get the following results:

$$X_n = A_n[X_0\cos Z_0 - Y_0\sin Z_0] \quad (9)$$
$$Y_n = A_n[Y_0\cos Z_0 - + X_0\sin Z_0] \quad (10)$$
$$Z_n = 0 \quad (11)$$
$$An = \prod_{i=0}^{n} \sqrt{(1 + 2^{-2i})} \quad (12)$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 7, July 2013

III. UNROLLED CORDIC ARCHITECTURE

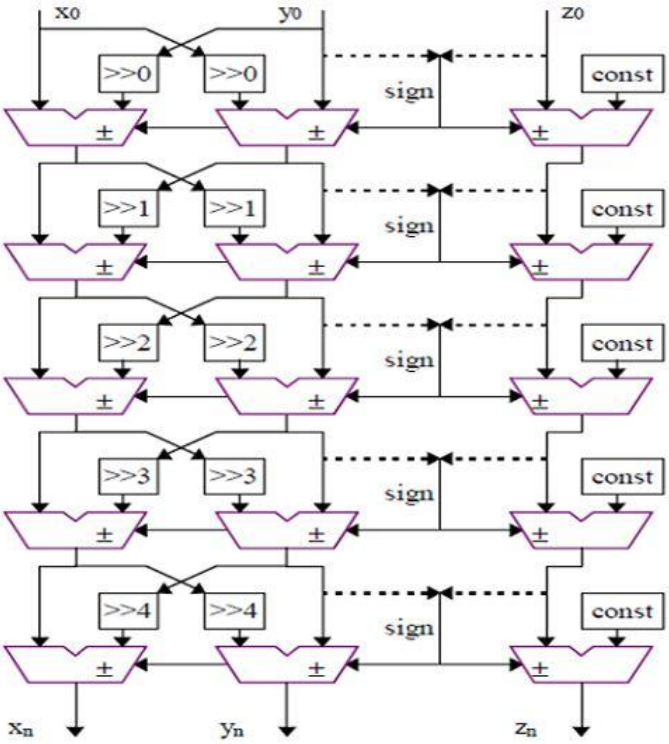

Fig. 1: Unrolled CORDIC Architecture

An unrolled architecture is shown in fig.1.Unrolled architecture has two advantages. First one is that the shifters are of fixed size and those can be implemented in the wiring. Second, Constants can be hardwired instead of requiring storage space that is the ROM that holds the arbitrary angle values need not to be updated after every iteration. The look up table (LUT) values for computing angle accumulator is distributed as constant to each adder in the angle accumulator chain so that the entire CORDIC processor is reduced to an array of interconnected adder-subtraction units. Unlike other architectures there is no need of registers which makes the unrolled architecture strictly combinational circuit. It has considerable delay, but processing time is reduced as compared to the iterative process. So the unrolled implementation provides the speed required for faster applications

.

The various components required for the radix-2 CORDIC processor implementation in unrolled fashion are the ROM required to store the angle values $\tan^{-1}(i)$ where i is varied from 0 to 16 and 32 for 16 bit and 32 bit processor respectively. There are barrel shifter required for shifting of the intermediate values of X_i and Y_i. The barrel shifters carry out a right shift which can be implemented using multiplexers. Adder/Subtraction unit is required in each iteration to calculate the next iteration values of X, Y and Z. The counter is required for the counting of the number of iteration of the CORDIC equations.

IV. IMPLEMENTATION AND RESULTS

The CORDIC processor is implemented with the following synthesis description:
Platform: FPGA
Family: Vertex6
Target device: XC6VCX75t
Package: FF484
Speed grade: -2

Fig. 2 and fig. 3, shows RTL schematics of the 16 & 32 bit CORDIC structure.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 7, July 2013

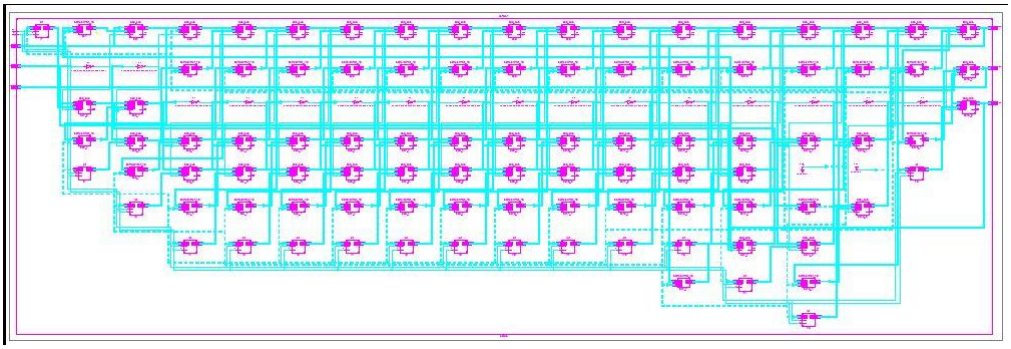

Fig.2 RTL Schematic of unrolled 16 bit CORDIC processor

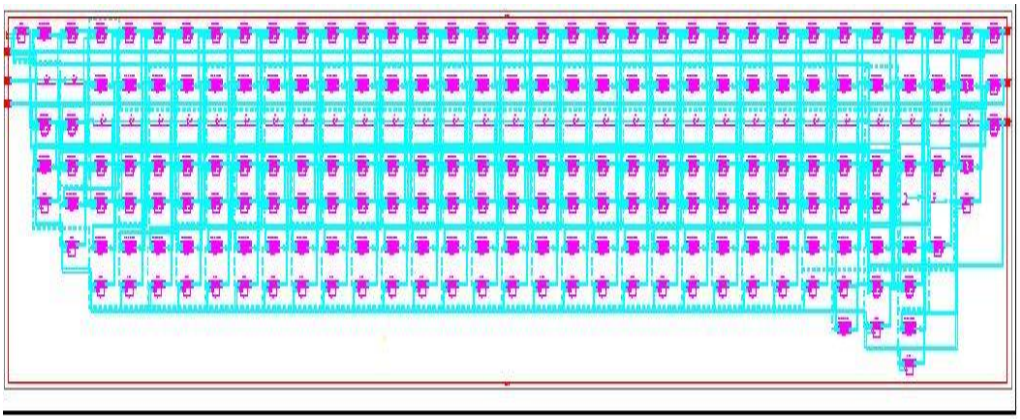

Fig.3 RTL Schematic of unrolled 32 bit CORDIC processor

Fig. 4 and fig.5, shows RTL simulation results of the 16 & 32 bit CORDIC structure.

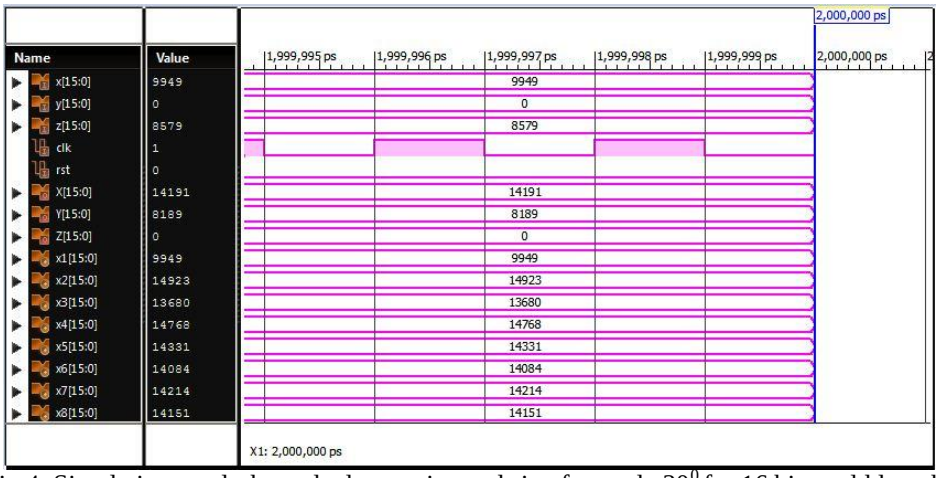

Fig.4: Simulation result that calculate cosine and sine for angle 30^0 for 16 bit world length.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 7, July 2013

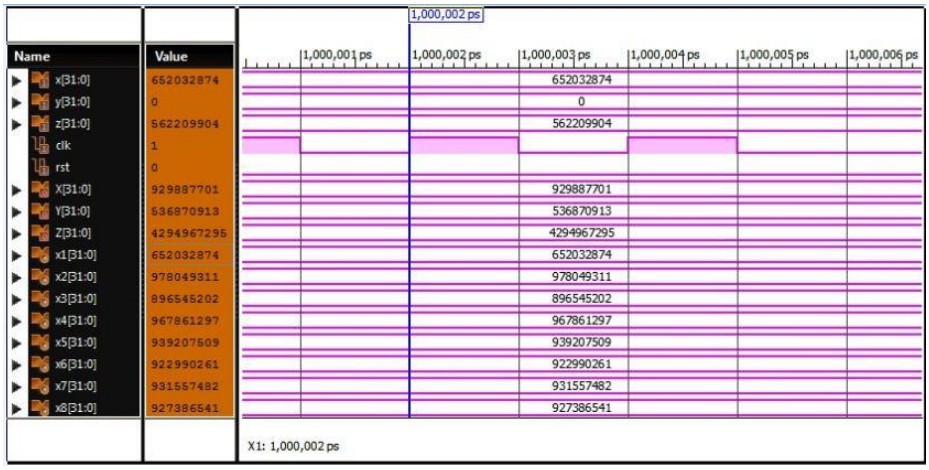

Fig.5: Simulation result that calculate cosine and sine for angle 30^0 for 32 bit world length.

TABLE I: device utilization summary for an input word length of 16 and 32bits.

Parameter	CORDIC structure	
	16 bit	32 bit
Number of Slice registers	1	1
Number of Slice LUTs	965	4104
Number of 4 input LUTs	1235	5441
Number of IOs	98	194
Number of bonded IOBs	98	194
No. of BUFG/BUFGCTRLs	1	1

TABLE II: Timing Behavior for 16 and 32 Bit Word Lengths

Parameter	CORDIC structure	
	16 bit	32 bit
Minimum input arrival time before clock	0.707ns	0.707ns
Maximum output required time after clock	45.278ns	165.979ns
Maximum combinational path delay	45.085ns	169.947ns

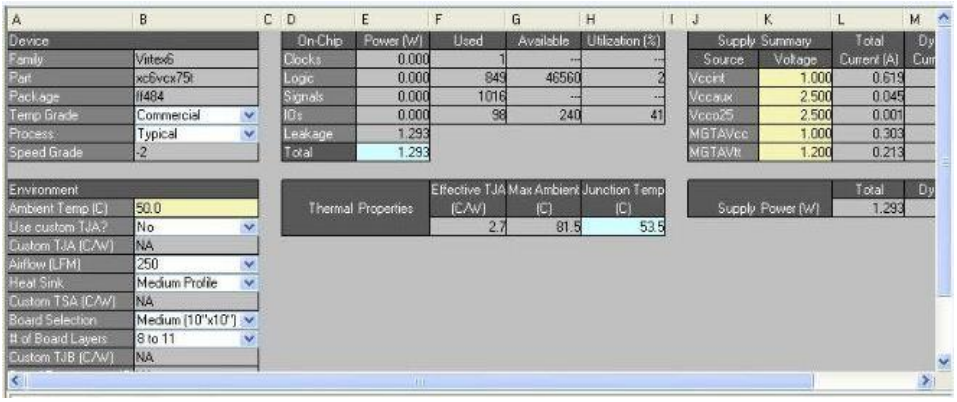

Fig.6: Power analysis report.

V. CONCLUSION

The CORDIC is a widely used algorithm in the field of DSP applications. This affects the cost, speed and flexibility of the DSP systems. Implementation of a CORDIC based processor on FPGAs can give enhanced speed at low cost with a lot of flexibility.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 7, July 2013

In this project 16 and 32 bit radix-2 CORDIC architectures are designed and simulated using Xilinx ISE using VERILOG as a synthesis tool. The output of the CORDIC architectures are analysed and verified, and compared with the actual values obtained from MATLAB. It is proved that by making use of CORDIC processor we can achieve high speed operation at reduced power and resource usage, which is essential in DSP applications. The analysis was carried for radix-2 CORDIC.

REFERENCES

[1] J.E. Volder, "The cordic trigonometric computing technique", IRE Trans Electronic Computers vol. 8, pp. 330–334, 1959.
[2] J.S. Walther, "A unified algorithm for elementary functions", in: Proceedings of spring. Joint Computer Conference, pp. 379–385, 1971.
[3] Pramod K. Meher et. Al "50 Years of CORDIC: Algorithms, Architectures, and Applications" IEEE transactions on circuits and systems—I: regular papers, vol. 56, no. 9, pp 1893-1907, September 2009.
[4] Andraka, R.,"A survey of CORDIC algorithms for FPGA based computers", FPGA '98, in ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp. 191- 200, 1998.
[5] Takafumi et al., "High Radix CORDIC algorithm for VLSI signal processing", in: IEEE Workshop on Signal Processing Systems (SIPS), pp. 183–192, November 1997.

BIOGRAPHY

J. M. Rudagi received the B.E. degree in Electrical Engineering from the Karnataka University ,Dharwad, India, in 1991 and M. Tech in Digital Electronics and Advanced Communication from Manipal University, Manipal,India. She is currently working as a Associate professor in KLE Dr.MSSCET,Belgaum. She has 22 years of teaching experience. Her research interest are in low power VLSI design.

Vinayak Dalavi received the B.E. degree in Electronics and Communications Engineering from the Visvesvaraya Technological University ,Belgaum, India, in 2008, He is currently pursuing M. Tech degree in VLSI design and Embedded Systems from Visvesvaraya Technological University. His research interests are VLSI design and embedded systems.