

MULTIPLEXER BASED CORDIC ALGORITHM BY USING FPGA

¹Neeraja Bandaru,²Mr. S. Ali Asgar,³Dr.P.Krishna Murthy

¹M.Tech PG Scholar,²Associate Professor, ³Associate Professor & Head

^{1,2,3}Department of ECE

^{1,2,3}ChadalawadaRamanamma Engineering College, Tirupati, Andhra Pradesh, India

Abstract:- This project presents the high speed and area achieved using the Coordinate Rotation Digital Computer (CORDIC) algorithm for digital signal processing applications. There are many efficient algorithms for CORDIC, these algorithms there are shifters, adders, and subtractors for sine/cosine wave generation. In this project we proposed multiplexers based CORDIC algorithm. Multiplexers based Coordinate Rotation Digital Computer algorithm used to achieve the fast and efficient hardware on FPGA. A six-stage Coordinate Rotation Digital Computer is achieved by three arrangements proceeding by unrolled CORDIC and MUXes based CORDIC upto three stages, MUXes based CORDIC upto the fourth stage with and without pipelining. The proposed architecture for CORDIC adders, subtractors and shifters, all are replaced by multiplexers upto the third stage and fourth stage. An 8 bit and 16-bit Coordinate Rotation Digital Computer to achieving the sine functions and cosine functions perform all methods on Xilinx Spartan 3E (XC3S250E) and Xilinx Virtex 6 FPGA(XC6VLX240). Compared with the unrolled CORDIC MUXes based CORDIC achieves the high operating frequency and less area requires for hardware implementation.

Keywords: CORDIC algorithm, ration mode, Field-programmable gate arrays (FPGAs), multiplexer, pipelining

I INTRODUCTION

Digital signal processing techniques are designed on microprocessors and microcontrollers. These processors are low cost but the problem is high and speed is not achieved, to overcome this problem FPGA introduces, which work at high speed at many digital signals processing applications. Based on the microrotation calculate the trigonometric function, this algorithm is called Coordinate Rotation Digital Computer. The advantage of the coordinate rotation digital computer algorithm is that the micro rotations are calculated by simple shift-and-add operations, which is very efficient in the hardware implementation to perform the sine and cosine function [1].

The coordinate rotation digital computer (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 such as singular value decomposition, eigenvalue estimations, QR decomposition, phase and frequency estimations, and synchronization in digital receivers, 3-D graphics processor, and interpolators require the CORDIC to operate in both rotation and vectoring-

modes. The 3-D 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, 3-D 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 brief, 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 this brief is structured as follows. Section II.

This MUXes based shows the high speed and hardware efficient Coordinate Rotation Digital Computer algorithm using for the digital signal processing applications. A reconfigurable CORDIC can be operated in rotation mode and vectoring operation mode. In this paper discuss only rotation mode. Nevertheless, the number of microrotation is a serious drawback of this critical path delay, the microrotation increases mean the propagation delay increases. Many authors have initiate schemes to decrease the micro iterations. In this project we proposed coordinate rotational digital computer (CORDIC) architecture adders and subtractors and shifters, all are replaced by multiplexers. CORDIC is designed by three followed by unrolled CORDIC and MUXes based CORDIC up to three stages and MUXes based CORDIC up to the four stages with and without pipelining. Replacing the multiplexer in place of adders and shifters reduces area utilization on FPGA and increases the speed of the operation.

I. The Original Unrolled Cordic

The CORDIC algorithm is based on vector rotations to find an approximation of a non-linear function. The main advantage with the algorithm is that it is multiplier less. It uses additions and subtractions only. The CORDIC is built with a number of stages. By increasing the number of stages the accuracy is improved. Fig. 1 shows an example of three B vector rotations, corresponding to a three stage CORDIC. The dotted vector is the vector with an angle α of which, in this case, the approximate cosine and sine values are searched for. There are many ways to choose where to start the rotations. In this project, the starting point is chosen to be at the coordinates $(x = 1/R, y = 0)$,

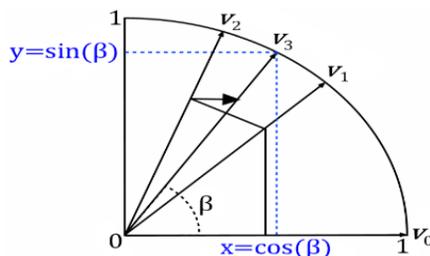


Figure 1: Three CORDIC Rotations

where R is the last vector length if we had started at the unit circle. A starting point at $1/R$ will thus result in that we come as close to the unit circle as possible when the last stage is computed. R is a fixed coefficient for the chosen number of CORDIC stages. It can thus be hardware wired in the implementation. When the last rotation is done, the coordinates give the approximate cosine and sine values for the input angle, i.e. (x_3, y_3) in the figure. (x, y) .

The figure shows three rotations. The first is positive and detected to be too large. The second will then do a negative rotation and finally the third will do a positive again. The rotations are fixed and done in the order 45, 27, and 14 degrees. The reason is that $\arctan(45) = 1$, $\arctan(27) = 1/2$, $\arctan(14) = 1/4$ (approximately). That is, they correspond to binary shifts.

A six stage CORDIC is chosen to demonstrate the methodology. However, the methodology is valid for any number of stages. Fig. shows the architecture of the six stage CORDIC, see at the end of the project. In the six adders at the top, the remaining angle is computed for each stage.

The input variable α is the angle for the cosine and sine that is searched for. At the top, the fixed coefficient angle values for each rotation are provided. These are added or subtracted from α in each stage. The coefficients are in this demonstration, chosen to be 8-bit wide. The coefficients can

be chosen with other bit widths, but it is not important in this particular case when the complexity reduction methodology is to be demonstrated. They are fixed coefficients but they can be stored in a ROM as well. In the middle and the lower adder rows the approximation of the cosine and sine value is computed and provided to the right.

II. HARDWARE IMPLEMENTATION PROPOSED CORDIC ARCHITECTURE

MUX BASED CORDIC

A six-stage multiplexer based on the coordinate rotational digital computer algorithm shown in Fig.. Let suppose the number of micro-iteration increases means accuracy also increases but increasing the iteration steps delay of the algorithm increase more and more.

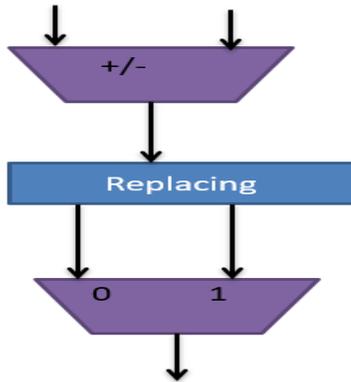


Figure 2: Adder/sub replacing with MUX

Using the multiplexer based CORDIC algorithm propagation delay reduced more effectively without compromising the accuracy. Using the multiplexer based CORDIC algorithm not only decrease the delay, but hardware utilization also reduces more effectively.

The number of stages removes by archives area utilization on FPGA implementation. A six-stage Coordinate Rotation Digital Computer is designed by three arrangements followed by unrolled CORDIC and MUXes based CORDIC up to three stages and MUXes based CORDIC up to the fourth stages with and without pipelining. The proposed architecture adders and subtractors and shifters, all are replaced by multiplexers up to fourth stages. In this paper, multiplexers are proposed for CORDIC algorithm, multiplexer based CORDIC algorithm used to achieve the fast and efficient hardware implementation on FPGA.

The first stage input is $x_{in} = x_{in}$, and $y_{in}=0$, so the output of the initial stage.

$$\begin{aligned}
 X_{i+1} &= (X_i - d_i \times 2^{(-i)} Y_i) \\
 X_{0+1} &= (X_0 - d_0 \times 2^{(-0)} Y_0) \\
 X_1 &= (x_i - d_0 \times 0) \\
 X_1 &= (x_{in})
 \end{aligned}
 \tag{1}$$

$$\begin{aligned}
 Y_{i+1} &= (Y_i + d_i \times 2^{(-i)} X_i) \\
 Y_{0+1} &= (Y_0 + d_0 \times 2^{(-0)} X_0) \\
 Y_{0+1} &= (0 + d_0 \times 2^{(-0)} x_{in}) \\
 Y_1 &= (x_{in})
 \end{aligned}
 \tag{2}$$

So remove the initial stage, directly this output of the first stage values give to the 2nd stage of the input. Addition/subtraction operation depends on the MSB of Z_i .

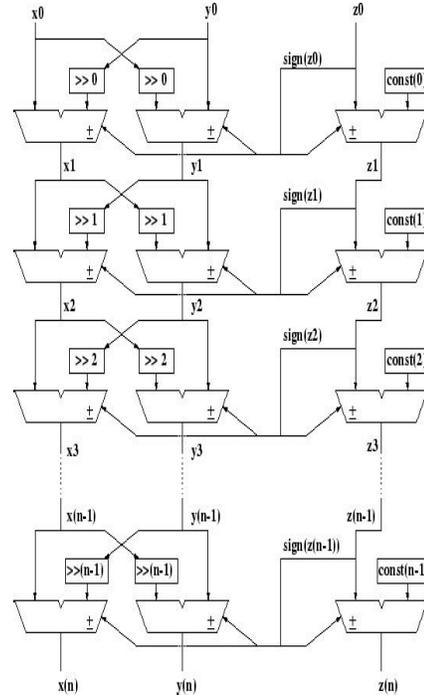


Figure 3: Architecture of Unrolled CORDIC

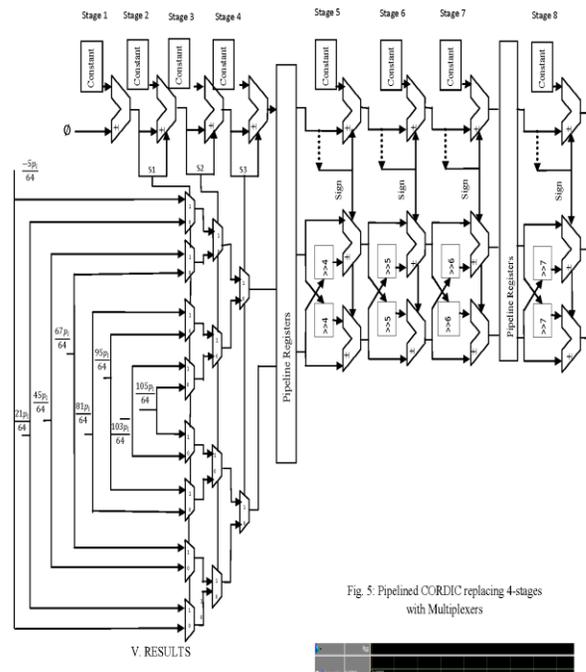


Fig. 5: Pipelined CORDIC replacing 4-stages with Multiplexers

Figure 4: The multiplexer is replaced up to four stages CORDIC algorithm.

Table 1: Implementation Results of Xilinx Spartan 3E eight bit CORDIC FPGA

Scheme	Number of slice Flip-flop	Number of Occupied slices	Critical path Delay (ns)
Pipelined unrolled CORDIC	187	166	6.3
Three stage multiplexers based CORDIC Pipelined	30	51	3.25
Four stage multiplexers based CORDIC Pipelined	32	58	3.85

IV Simulation Results

The architecture of the MUXes based four-stage pipelined CORDIC algorithm. Pipelined multiplexer based CORDIC consists of the only multiplexer and registers (excluding the last three stages). Pipeline concept mainly used to reduces the propagation delay and producing something that acceptable for the high-speed applications. But when use pipelining such as hardware (area) increases more and more. The register depends upon the how many stages present in the architecture. A six-stage CORDIC is implemented by three arrangements achieved by unrolled CORDIC, MUXes based CORDIC up to three stages and multiplexer based CORDIC up to the fourth stage with and without pipelining. Fig. shows the Multiplexer are replaced up to three stages CORDIC algorithm. First, three stages replaced with the multiplexers and third, fourth and fifth stages are used pipelining registers shown in above Fig. 3.8. Due to pipelining, the propagation delay should be reduced more but the area also increased. Four fixed values that should be given to the input of the multiplexers for the three-stage MUX based CORDIC shown in Fig. 3.8. Pipelined multiplexer based CORDIC operating frequency is very high compared to the pipeline unrolled CORDIC architectures. Eight fixed values that should be given to the input of the multiplexers need for the fourth stage MUX based CORDIC shown in above Fig. The Fig. shows the MUXes are replaced up to four stages. Pipelined registers placed in fourth and fifth stages shown in Fig. Multiplexer based CORDIC operating frequency is very high compared to the unrolled CORDIC architectures.

FPGA Simulation Results:

Fig. 5. Shows the RTL schematic for the proposed CORDIC architecture for the top module name (final module). In the final module we have the five inputs and three outputs (See the Proposed architecture). In the top module we are instant ion the mux and adder, subtractor and hardshifters. The proposed cordic architecture we are implemented the with pipelined and without the piplined. Both results mentioned in the below page. The proposed and existing architectures we are implemented in the eight bit and sixteen bits in the xilinx tool. Fig shows the proposed architecture simulation results in xilinx tools. Xin yin and ein represents the inputs of the cordic architecture and clk and rst represents the clock and reset and xx1 and yy1 represents the output of the right shifter and sin cos and eout represents the output of the cordic architecture.

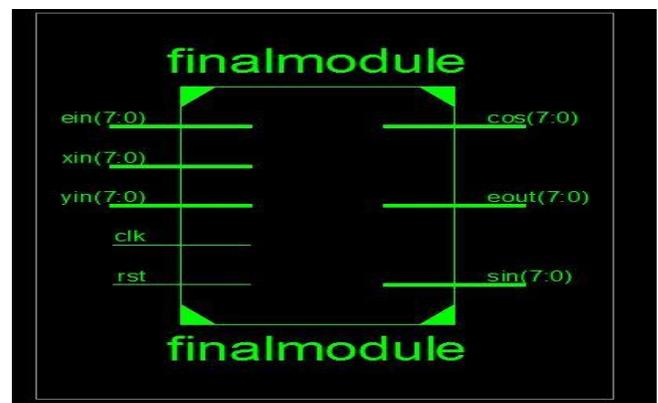


Figure 5: RTL schematic for the proposed CORDIC architecture for the top module

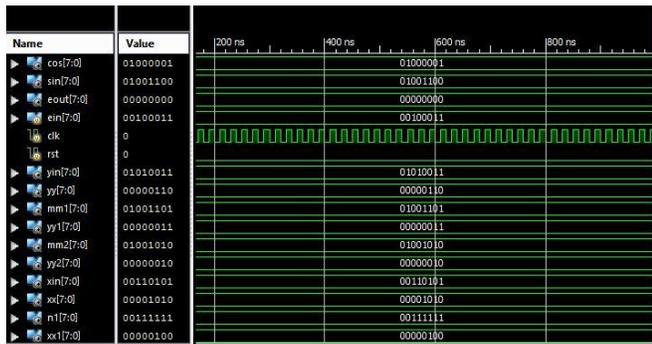


Figure 6: The proposed architecture for the simulation results

An eight-bit and sixteen-bit CORDIC for producing the sine function and cosine function without and with pipeline

implementation based on the unrolled and multiplexer based CORDIC. The proposed architectures were modelled in Verilog and synthesized using FPGA, this device named as Spartan 3E (XC3S250E) and Xilinx Virtex6 FPGA (XC6VLX240). In this project proposed multiplexer based CORDIC algorithm, this proposed algorithm achieves the high speed and less hardware implementation on FPGA. Table II shows the comparison of the pipelined unrolled and multiplexer based CORDIC. Implementation Results on Xilinx Spartan 3E eight-bit CORDIC on FPGA. Shown in above Table II unrolled pipelined CORDIC requires the more area and propagation delay also very high compared to the multiplexer based CORDIC. In the unrolled CORDIC, all adders and subtractors replace with the multiplexers up to the third and fourth stages.

Table 2: Implementation Results of Xilinx Spartan 3E eight bit CORDIC FPGA

Scheme	Number of slice Flip-flop	Number of Occupied slices	Critical path Delay (ns)
Pipelined unrolled CORDIC	187	166	6.3
Three stage multiplexer based CORDIC Pipelined	30	51	3.25
Four stage multiplexer based CORDIC Pipelined	32	58	3.85

The total of six multiplexers needs in pipelined multiplexer based CORDIC up to the third stage. Because of this area and propagation delay, both are reduced compared to the unrolled CORDIC. Same procedure repeats the fourth stage, in the fourth stage need total 14 multiplexers in the place of adders and sub tractors. In the fourth stage eight fixed input values shown in the above Fig. Now the fifth stage also replaces with the multiplexers, but this is not efficient shown in above Table III. In fifth stage total need 30 multiplexers are required for in place of adders and subtractors. When replacing the multiplexers in the fourth and fifth stage area and propagation delay not more efficient.

Table 3: Proposed Results on Xilinx devices name is Virtex 6 FPGA (XC6VLX240) 16 bit CORDIC for unrolled and Pipelined up to three stage multiplexer based CORDIC.

Scheme	Unrolled two level Pipeline [6]	Pipelined up to three stage multiplexer based CORDIC
Slice LUTs	959	487
Critical Path Delay(ns)	29	1.262
Frequency (MHz)	34	792.45

III. CONCLUSION

In this paper, 8 bit CORDIC and 16 bit CORDIC using the unrolled and multiplexers based for generating the sine and cosine. A six-stage CORDIC is implemented by three schemes followed by unrolled CORDIC and multiplexer based CORDIC up to three stages and multiplexer based CORDIC up to the fourth stage with and without pipelining. The implementation results on Xilinx Spartan 3E(XC3S250) devices and Xilinx Virtex6 FPGA(XC6VLX240), shown in above TABLE I and TABLE II. From these tables, it is found that the pipelined multiplexer-based approach (up to the third stage) requires fewer area and propagation delay also very less compared with the unrolled pipelined CORDIC. Pipelined multiplexer-based approach for the three-stage CORDIC has very good efficiency compared with the fourth stage and fifth stage CORDIC because fixed values and MUXes should be increased more and more. The TABLE I, TABLE II show the eliminating four stages is almost as good as three stages. Replacing the multiplexer in place of adders and shifters reduces area utilization on FPGA and increases the speed of the operation.

REFERENCES

- [1] Smith, S.M., Brady, J.M., "SUSAN – a new approach to low-level image processing," *International Journal of Computer Vision*, pp. 45–78, 1997.
- [2] Gonzales and Woods, "Digital Image Processing", Pearson Education, India, Third Edition.
- [3] Schlick, "An adaptive sampling technique for multidimensional ray tracing," in *Photorealistic Rendering in Computer Graphics*, pp. 21–29, Springer, Berlin, Germany, 1994.
- [4] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D.P. Greenberg, "Multiscale model of adaptation and spatial vision for realistic image display," in *Proceedings of the Annual Conference on Computer Graphics (ACM SIGGRAPH '98)*, pp.287–298, July 1998.
- [5] Tomasi, C., Manduchi, R., "Bilateral filtering for gray and Color images," in *IEEE Proc. Int. Conf. Computer Vision*, pp. 839–846, 1998.
- [6] Z. Rahman, D. J. Jobson, and G. A. Woodell, "Multiscale Retinex for color rendition and dynamic range compression," in *Signal and Image Processing*, vol. 2847 of *Proceedings of the SPIE*, pp. 183–191, 1996.