Indian Journal of Science and Technology

FPGA Implementation of RA-CORDIC Processor

Subha Sri Thirveedhi[1*] and Muthaiah Rajappa[2]

[1]PG Student, School of Computing (VLSI), SASTRA University, 613401, Thanjavur, Tamilnadu, India; luckysanju8@gmail.com[1]

[2]Professor, School of Computing (VLSI), SASTRA University, 613401, Thanjavur, Tamilnadu, India; sjamuthaiah@core.sastra.edu[2]

Abstract

In processing the real world data Digital Signal Processing algorithms provide unbeatable performance. One of the DSP algorithms is COordinate Rotation DIgital Computer (CORDIC). For real-time airborne computation, CORDIC act as a special purpose digital computer. Basically the CORDIC is categorized in two different styles such as sequential (folded) and combinational (unfolded). This paper presents a novel architecture of CORDIC using redundant arithmetic i.e., RA-CORDIC. The RA-CORDIC structure shows better latency and obtains maximum throughput. The structure has been coded in VERILOG, synthesis analysis are performed using Xilinx ISim tool and targeted on Xilinx FPGA synthesis tool.

Keywords: FPGA, CORDIC Algorithm, Folded & Unfolded Architectures, Redundant Arithmetic.

1. Introduction

The basic functions such as trigonometric, inverse trigonometric, logarithmic, exponential, multiplication and division functions are used in many of the DSP algorithms [1] some of the software solutions are the traditional approach to implement these functions. The exploitation of look-up tables, power series includes in software solutions, however they suffered from huge drawbacks.

Although Look-up tables are fast they require hefty amount of memory for high precision results. To achieve desired precision the use of power series was time consuming as it was too slow. One of the digital signal processing algorithms is CORDIC, it came into existence to present efficient hardware solutions [2]. The computation of CORDIC represents the compromise between power series and look-up tables, where the results of precision are saved without any small amount use of on chip memory. In real world the architectures of high speed VLSI becomes more reality for modern digital signal processing systems, with the advancement in VLSI technology. This provides the CORDIC algorithm to map easily into hardware structures according to the need of designers. Now-a-days these types of architectures are frequently used in DSP systems, which have a rapid growth of increase in their performance [1]. Various performance parameters such as speed, power and area are achieved by optimizing these structures. Whereas suitable hardware is chosen to implement these optimized structures.

A wide variety of implementation formats are provided by IC technology for system designers. The implementation format defines how the organization of switching elements takes place and how the system functionality will be analyzed. Moreover the FPGA platform is the most advanced implementation platform used in today's life time. To implement CORDIC architectures, FPGAs are often used as co-processors. Compared to ASIC, FPGAs have been slow, less energy efficient and generally achieves less functionality.

2. CORDIC Algorithm

By the utilization of only shift and add operations [3], the CORDIC offers iterative process which performs

** Corresponding author:*
Subha Sri Thirveedhi (luckysanju8@gmail.com)

vector rotations by denoting them as arbitrary angles. The Volder algorithm [4] results from the generalized rotation transform.

$$x \sin ø + y \cos ø = y' \quad (1)$$

$$x \cos ø - y \sin ø = x' \quad (2)$$

This Cartesian plane rotates by the angle ø, as shown in Figure 1.

The above equations can readjust as:

$$[y + x \tan ø] \cos ø = y' \quad (3)$$

$$[x - y \tan ø] \cos ø = x' \quad (4)$$

These rotation of angles constrained so $\tan ø = \pm 2^{-i}$. This will decreases tangent multiplication by simple shift operation. Another rotation of arbitrary angles can easily accessible by the execution of elementary rotations in simple way. CORDIC, however, performs rotations by a sequence of micro-rotations by elementary angles (α_i). These are obtained by decomposing ø into elementary rotations in sequence manner:

$$ø = \Sigma \alpha_i \quad (5)$$

By using these we can form simple iterative rotations

$$[y_i + x_i \tan \alpha_i] \cos \alpha_i = y_{i+1} \quad (6)$$

$$[x_i - y_i \tan \alpha_i] \cos \alpha_i = x_{i+1} \quad (7)$$

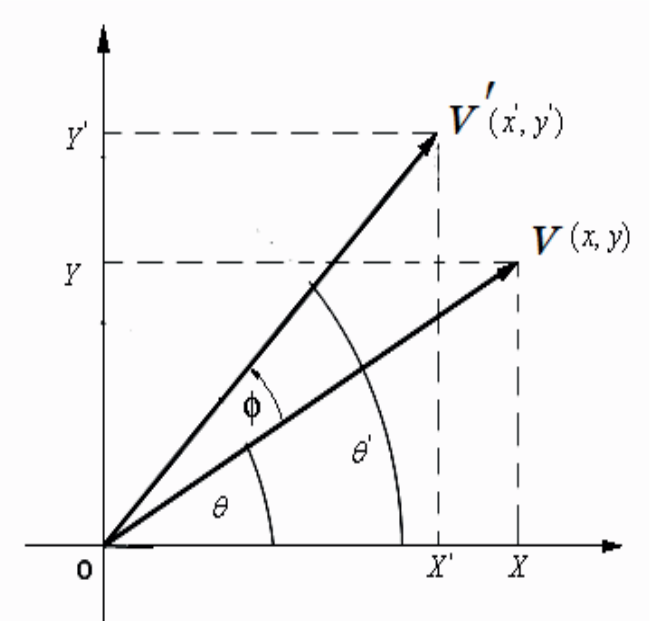

Figure 1. Vector rotation by **Ø** angle.

From trigonometric identities we have:

$$\cos \alpha_i = 1/ (1 + \tan^2 \alpha_i)^{1/2} \quad (8)$$

Substituting equation (8) in equation (6) and (7) we have:

$$[y_i + x_i \tan \alpha_i]/(1 + \tan^2 \alpha_i)^{1/2} = y_{i+1} \quad (9)$$

$$[x_i - y_i \tan \alpha_i]/(1 + \tan^2 \alpha_i)^{1/2} = x_{i+1} \quad (10)$$

To assure that the tangent multiplication reduced to small shifting operation, the rotation angles obtained from following relation:

$$\text{Tan}\alpha_i = \pm 2^{-i} \quad (11)$$

Here i represent total number of iteration. And tan α_i replaced in above equation Substituting equation (12) in (9) and (10), we have:

$$[y_i + x_i .d_i. 2^{-i}].K_i = y_{i+1} \quad (12)$$

$$[x_i - y_i .d_i. 2^{-i}].K_i = x_{i+1} \quad (13)$$

Here,

$1/(1+2^{-2i})^{1/2} = K_i$, denoted as scale of constant.

$d_i = \pm 1$, it's denoted as decision function.

Shift-add algorithm eliminates the constant scale value from yields of obtained equations of vector rotation. Product term K_i's is applied in the system. The product value reaches to 0.6073 by using infinite number of iterations. A_n denoted as rotation algorithm gain,

$$A_n = \pi [1 + 2^{-2i}]^{1/2} \quad (14)$$

For infinite iterations, the gain is approximately equal to 1.647. However gain can be calculated by the help of total number of iterations, and also it depends on composite rotation angle is distinct by sequence of elementary rotations in the directions manner. These sequences denoted as decision vector. And all these vectors used for angular measurement system, based on the values of binary arctangents. Conversions done by using a look-up table between angular system and any other systems. An improved conversion technique uses additional subtract or adder unit, for each iteration it mount up the elementary rotation angles. These angels are expressed by convenient angular unit. A small lookup table supplies the angular values, and also we can use hardwired, depending on suitable implementation.

Accumulator angle can add the 3rd value of difference equation to CORDIC:

$$z_i - \alpha_i = Z_{i+1}$$

$$Z_{i+1} = z_i - d_i \tan^{-1}(2^{-i})$$

Thus, only one CORDIC micro-rotation equations can be written as:

$$x_i - y_i . 2^{-I} . d_i = x_{i+1} \quad (15)$$

$$y_i + x_i . 2^{-I} . d_i = y_{i+1} \quad (16)$$

$$z_i - d_i \tan^{-1}(2^{-I}) = z_{i+1} \quad (17)$$

3. CORDIC Architectures

CORDIC architectures are commonly categorized into sequential (folded) and combinational (unfolded) depend on hardware realization of iterative equations [5]. The folded architecture is obtained by the direct duplication of equations 15, 16, and 17. In time domain the sequential architecture has to be multiplexed so that all iterations are approved in a particular functional unit. The signal processing architectures delivers a means for operating area for speed [6]. Using a word sequential design the folded architecture is implemented the unabridged CORDIC core.

3.1 Folded Word Sequential Design

Folded word sequential design [7] is duplicated by using three difference equations in each. This is also called as iterative bit-parallel design, the hardware as shown in Figure 2. Each block contains a shift unit, subtraction-adder unit block, and one register used for output buffering. Initial values are given to register by the multiplexer for first level calculation. Here z-branch of MSB stored value determines the operation mode for the adder-subtraction unit. x and y outlet Signals passes to block of shift unit and finally subtracted or added to un-shifted signal value in reverse path. The *z* branch mathematically mixing the registers values, lookup table passes these values, then each number of operations the address value is changed. After n operations output is passes again to register block before primary values are fed in again and these final value can be access as output. Normal FSM needs to control the addressing of the constant values and multiplexers. All the initial values are given hardwired in a word wide manner when it is implemented in FPGA. Both subtracted and adder component are passed out separately. Angle accumulator controls the multiplexer. Routing signals are required to find distinguish between subtraction and addition. For varying the shift

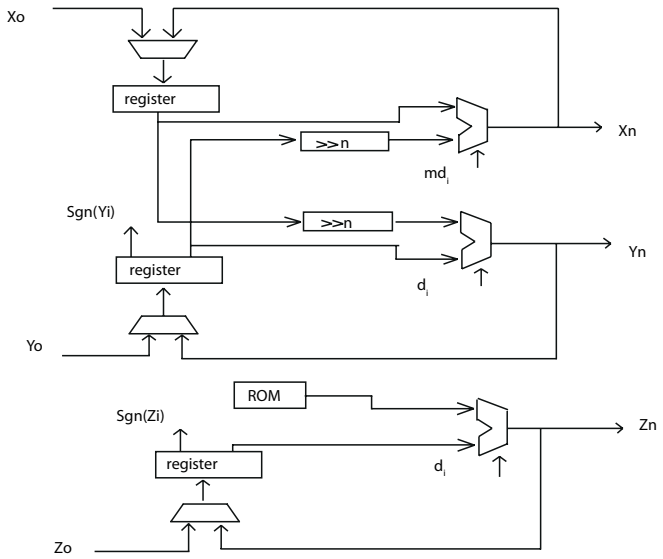

Figure 2. Folded word sequential design.

distance value, shift operations are used. Shifters are not suitable for FPGA architectures because it desires several layers of logic. Due to numerous layers of logic cells, the resultant design structure will become slow.

3.2 Unfolded Parallel Architecture

The CORDIC processor discussed above is iterative algorithm, it means processor need to perform n iterations at the given data rate. This is an unfolded operation [8], it means always performs the same iteration at n times processing elements. It shows in Figure 3. This will result the value in two simplifications. First one is fixed shifts at shifters; by using wiring we can implement it. Another one based on lookup table. Here angle accumulator distributed the LUT values, as constants to each adder; this is chain process in accumulator angle. Instead of using storage space we are moving to constants, which are hardwired. The whole CORDIC processor can be modified into grouping of subtraction-adder unit in interconnected manner. Now we do not require registers, building of this unrolled processor stringently combinatorial. Finally resulting circuit delay should be substantial; now processing time is decreased compare to iterative circuit. Mostly and particularly in FPGA, they do not make any sense of using combinatorial large circuit what we required. The design of unrolled processor can be easily pipelined [9] by inserting registers in-between the subtraction- adder unit block. Moreover FPGAs already contain registers in each logic cell, so this pipelined registers do not increase the hardware cost.

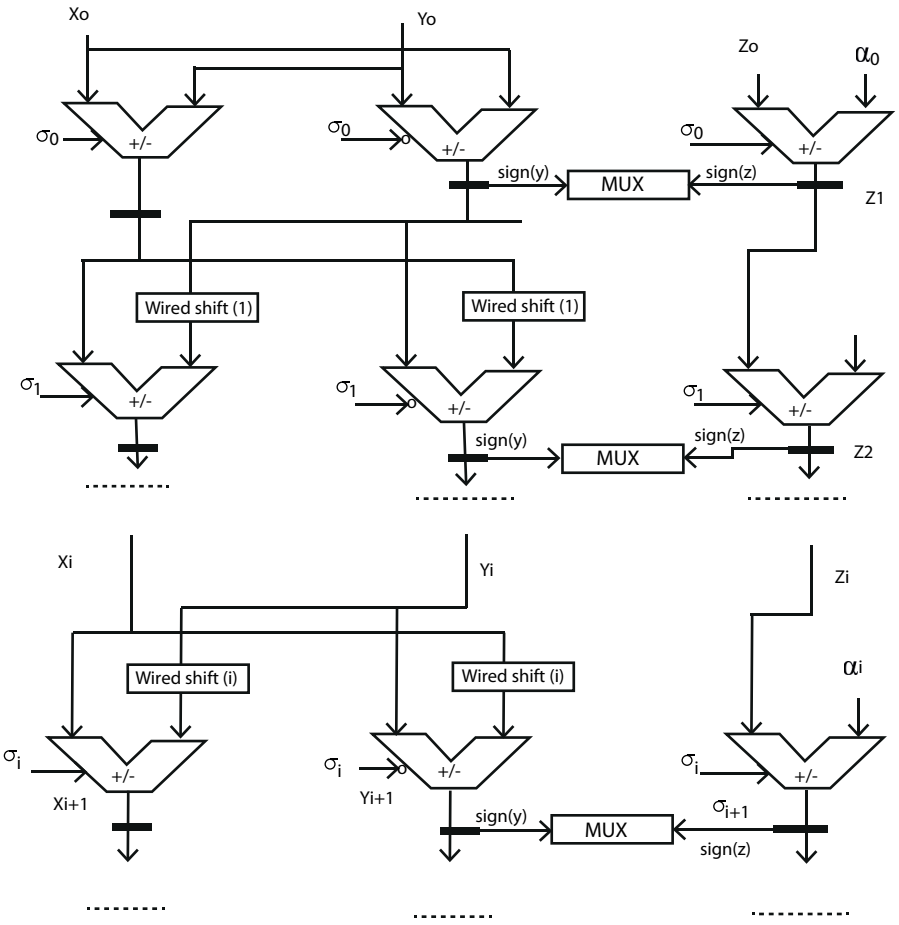

Figure 3. Unfolded parallel architecture.

4. Redundant Arithmetic

Adders plays a major role in CORDIC and due to carry propagation in adders the delay increases rapidly and slow down the speed of operation so that we move on to Redundant Arithmetic to decrease the delay and increase the speed of operation. The conservative tasks like subtraction, multiplication, and addition produces carry-propagation chains. Redundant number scheme announced to resolve this problem [10]. Redundant number scheme improves the arithmetic operations speed. This method used for sign processing and additional applications. When the reconversion and conversion circuitry shares the information among all the function units, this method also saves the area in VLSI and also power dissipation, due to these two reasons system will become more effective. Redundant number systems (RNS) suitable for numerically intensive applications. RNS can prevent or captures the carry propagation, by generating parallel adders with delay of constant value; it won't depend on the operand word-length. This will produced in an RNS format by using low latency results. RNS can improve the performance in mathematically intensive applications. However, these arithmetic circuits implementation is expensive because for each symbol multiple bits required. These circuits will eliminate carry propagation, by giving near constant addition delay, regardless of the operand width. The attractiveness of the redundant signed-digit number systems lies in their "carry-free" addition property. This feature makes them very useful in implementations where the computations start from the most significant bit (MSB) first.

4.1 Carry Free Radix-2 Addition

Redundant number representations limit the carry propagation to a few bit positions; this is usually independent to word length W. This carry free propagation feature enables fast addition.

The algorithm to carry out signed binary digit addition is not unique, and therefore, its logic implementation can be diverse. By using two binary unsigned numbers, it can perform radix-2 operation, one bit is negative and another bit is positive and it can be represented as [11]

$$X = X^{+} - X^{-} \tag{1a}$$

x_i^+ and x_i^- are both negative as well as positive numbers these bit values are 0 and 1, x_i should varied {1, 0, 1}, all these values are given in Table 1.

4.2 Hybrid Radix-2 Addition

In a hybrid operation, one input operand and the output operand are in redundant signed-digit representation, and the 2nd input operand is a conventional unsigned number. Consider the addition of signed-digit number the radix-2 operation $X_{<2.1>}$ came into picture, here 2 indicates radix-2 task, and 1 indicates largest digit value and an unsigned conventional number Y.

$$X_{<2.1>} + Y = S_{<2.1>} \tag{2a}$$

In 2 steps we can get addition value. Here 1st step all the bits are in parallel positions i ($0 \le i \le W\text{-}1$), W being the word length. The intermediary sum $p_i = x_i + y_i$ is calculated, it ranges between {1, 0, 1, 2}. This addition can be

$$x_i + y_i = p_i = 2t_i + u_i, \tag{3a}$$

Table 1. Redundant number system of radix-2

x^+	x^-	X	$X = x^+ x^-$
0	0	0	00
0	1	−1	01
1	1	1	10
1	1	0	11

Table 2. Summarizes the digit sets involved in hybrid radix-2 addition

Digit	Radix 2 Digit Set	Binary Code
x_i	{1, 0, 1}	$x_i^+ - x_i^-$
y_i	{0, 1}	y_i^+
$p_i = x_i + y_i$	{1, 0, 1, 2}	
u_i	{1, 0}	$-u_i^-$
t_i	{0, 1}	t_i^+
$s_i = u_i + t_{i-1}$	{1, 0, 1}	$s_i^+ - s_i^-$

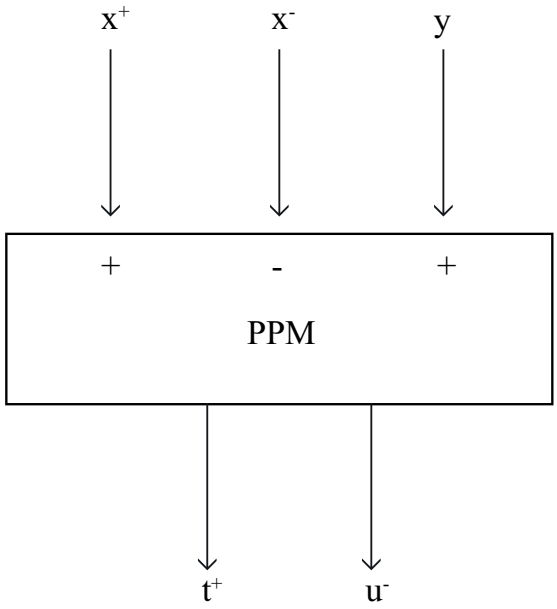

Figure 4. Hybrid radix 2 PPM adder.

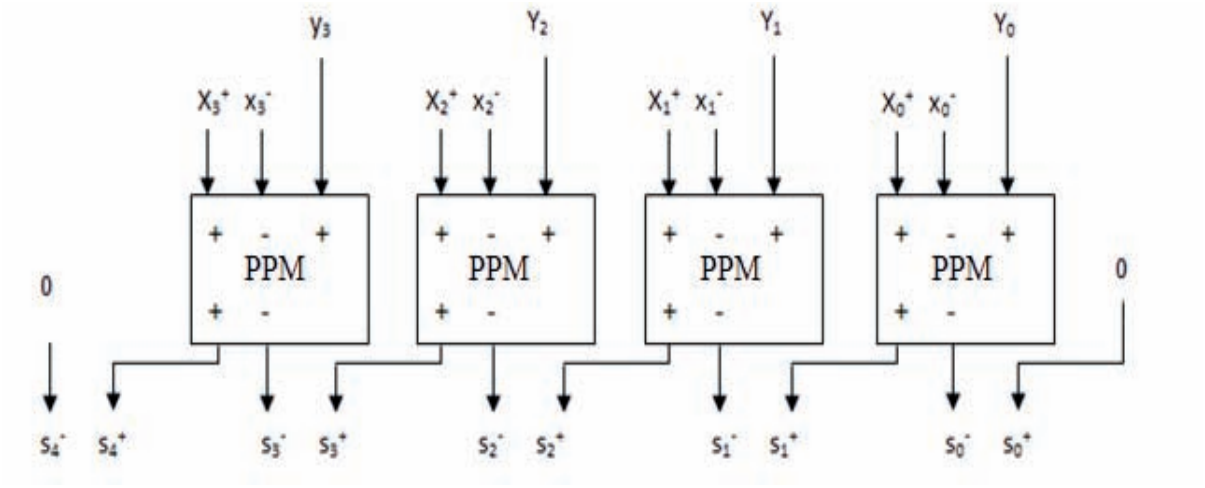

Figure 5. Four digit hybrid radix-2 adder.

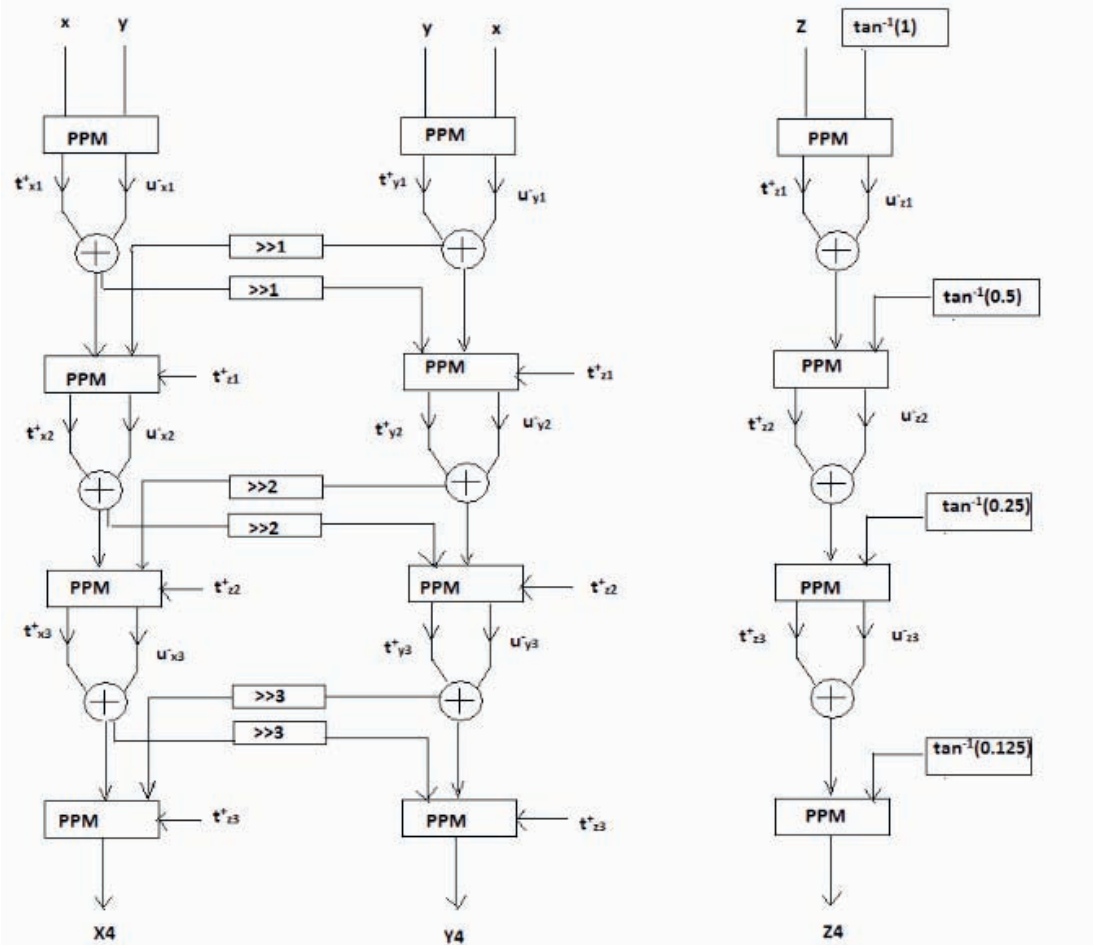

Figure 6. A novel architecture for CORDIC processor using redundant arithmetic.

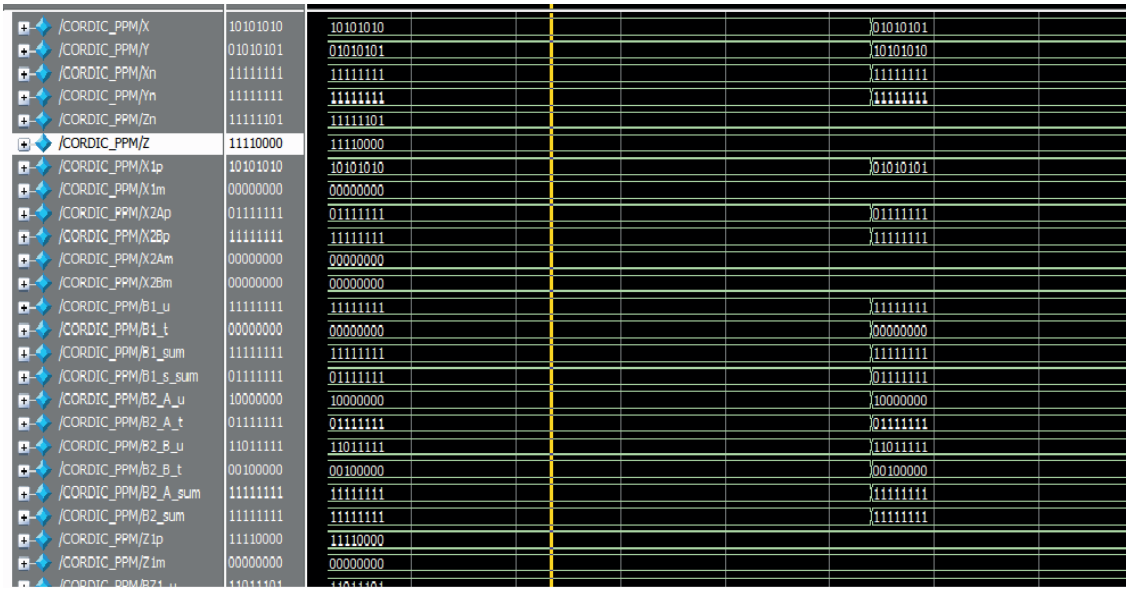

Figure 7. Simulation view of novel architecture (RA-CORDIC).

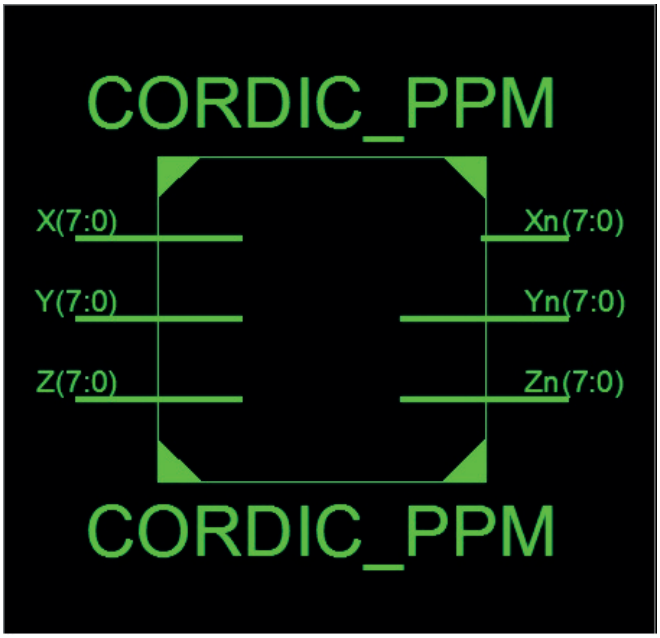

Figure 8. Block view of novel architecture (RA-CORDIC).

Table 2 summarizes hybrid radix-2 addition, in that table t_i denotes transfer digit and it varies value from 0 or 1, and it is also represented as t_i^+ and u_i denotes interim added sum and it varies the values either 1 or 0, and it is also represented as u_i^-. The least significant transfer digit t_{-1} is assigned the value zero, the same as the most significant interim sum digit u_w.

Second step contain digit sum s_i designed by linking t^+_{i-1} and also u_i^-, it is one of the single digit:

$$s_i = t^+_{i-1} - u_i^-. \tag{4a}$$

Replacing the corresponding binary codes from table 2 in (3a) we get:

$$x_i^+ - x_i^- + y_i^+ = 2t_i^+ + u_i^- \tag{5a}$$

These all operation are performed by using type-1 full adder [12], it is nothing but plus-plus-minus adder (PPM) [13] as shown in Figure 4. The four digit hybrid radix-2 adder is shown in Figure 5. The novel architecture for CORDIC algorithm using Redundant Arithmetic (RA-CORDIC) is shown in Figure 6.

5. Implementation Results

The proposed novel architecture of CORDIC processor using redundant arithmetic (RA- CORDIC) is implemented for four stages. The design is performed by using VERILOG HDL and simulation results are carried out by using

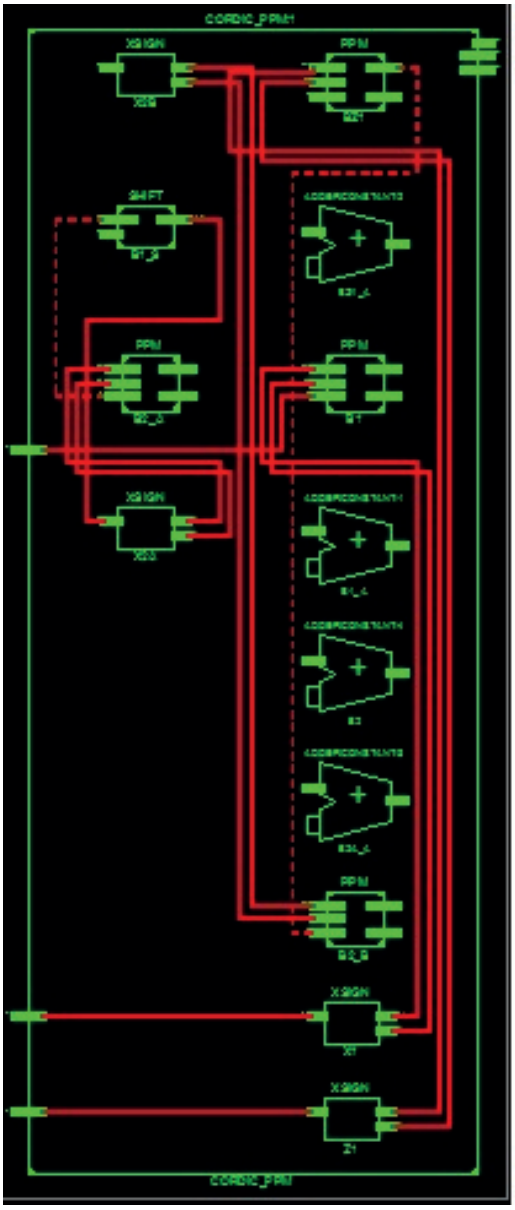

Figure 9. RTL schematic view of novel architecture (RA-CORDIC).

MODELSIM. Figure 7 shows the simulation report of novel architecture. The synthesis analysis of novel architecture is carried out by using Xilinx ISE 12.1 [14]. Figure 8 shows the block view of novel architecture. Figure 9 and Figure 10 evaluates the RTL and Technical view of novel architecture and they are specifically targeted on FPGA devices.

6. Conclusion

This paper presents a novel architecture for CORDIC by using redundant arithmetic (RA-CORDIC). Mainly in

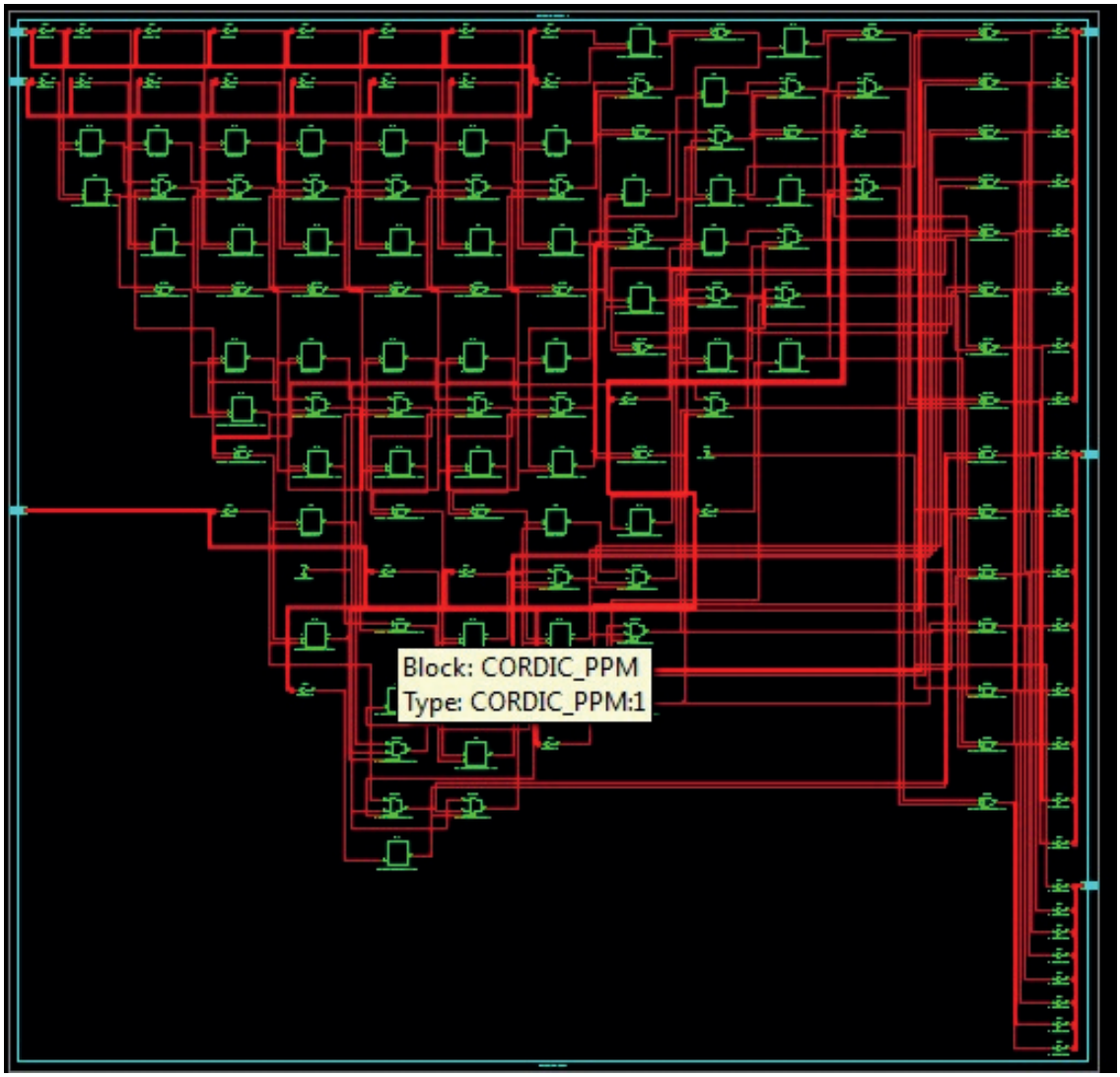

Figure 10. Technical view of novel architecture (RA-CORDIC).

hefty operand of DSP and high speed applications, the above novel architecture can be used. The novel architecture has been implemented and targeted for FPGA devices. This design offers a better latency and achieves maximum throughput rate.

7. References

1. Khurshid B, Rather G M et al. (2012). Performance comparison of non-redundant and redundant FPGA based unfolded CORDIC architectures, International Journal of Electronics and Communication Technology, vol 3(1), 85–89.
2. Khurshid B, Rather G M et al. (2012). Performance analysis of CORDIC architectures targeted for FPGA devices, International Journal of Advanced Research in Computer Science and Software Engineering, vol 2(2).
3. Deprettere E, Dewilde P et al. (1984). Pipelined CORDIC architecture for fast VLSI filtering and array processing, Proc. ICASSP'84, 1984, 250–253.
4. Volder J E (1959). The CORDIC trigonometric computing technique, IRE Transactions on Electronic Computing, vol EC-8(3), 330–334.
5. Meggitt E (1962). Pseudo division and pseudo multiplication processes, IBM Journal, vol. 6, no. 2, 210–226.
6. Lin C H, and Wu A Y (2005). Algorithm and architecture for high-performance vector rotational DSP applications, Regular IEEE Transactions: Circuits and Systems I, vol 52, 2385–2398.

7. Andraka R (1998). A survey of CORDIC algorithms for FPGA based computers, Proceeding FPGA '98 Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, 191–200.
8. Hu Y H (1985). Pipelined CORDIC architecture for the implementation of rotational based algorithm, Proceedings of the International Symposium on VLSI Technology, Systems and Applications, 259.
9. De.Lang A A, Van der Hoeven A J et al. (1988). An optimal floating point pipeline CMOS CORDIC processor, IEEE International Symposium on Circuits and Systems, vol 3, 2043–2047.
10. Avizienis A (1961). Signed digit number representation for fast parallel arithmetic, IRE Transactions on Electronic Computers, vol EC-10(3), 389–400.
11. Mudassir R, El-Razouk H (2005). New designs of 14-transistor PPM adder, Canadian Conference on Electrical and Computer Engineering, 1739–1742.
12. Hwang K (1999)., Computer Arithmetic: Principles, architectures and design, Wiley, 1979. Integrated System Laboratory, ETH Zurich.
13. Abid Z, and Wang W (2008), New designs of redundant-binary full adders and Its applications, IEEE International Symposium on Circuits and Systems, ISCAS, 3366–3369.
14. ISE Simulator (2011), Xilinx incorporation San Jose, U.S.A.

8. Authors Biography

SUBHA SRI THIRVEESDI received the B.Tech degree from the Department of Electronics and Communication Engg. Bhoj Reddy Engineering College for Women, Hyderabad (AP) - India in the year of 2007.

She is currently pursuing the M.Tech (VLSI DESIGN) in School of Computing, SASTRA UNIVERSITY, Thanjavur (TN) - India. Her research interest includes High Performance VLSI Architectures, Signal Processing, Communication and Electron Devices and Circuits.

MUTHAIAH RAJAPPA completed the B.Tech from Annamalai University – Chidambaram (TN), India in the year of 1989 and then the degree of M.E and PhD were completed in SASTRA UNIVERSITY- Thanjavur (TN), India in the year of 1997 and 2009. He has so far had 16 Years of Teaching Experience in Engineering Institution. He is currently working as Professor at SASTRA UNIVERSITY, Thanjavur (TN), India. Moreover he has 15 Reputed International Journal Publication, 2 International Conference Publication and 2 National Conference Publication. His area of interest includes VLSI, Embedded and Image Processing.