ASIC Implementation of Redundant Arithmetic CORDIC Processor

Katakam Supraja Harsha
PG Student, M.Tech VLSI System Design, CVR College of Engineering Hyderabad, India.

T. Subha Sri Lakshmi
Assistant Professor, ECE Department, CVR College of Engineering Hyderabad, India

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). Actual time waving estimation, CORDIC act as a special purpose digital computer. The CORDIC is categorized in two different styles such as folded (sequential) and unfolded (combinational). 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 is performed using RTL compiler and Physical design is done using EDI tool.

Index Terms – ASIC, CORDIC Algorithm, Folded & Unfolded Architectures, Redundant Arithmetic.

1. INTRODUCTION

The essential functions such as trigonometric, inverse trigonometric, logarithmic, exponential, multiplication and division functions are used in many of the DSP functions [1] some of the shareware solutions are the conventional way to implement these functions. The using of Look-up tables, power series includes in program solutions, however they have many flaws.

Even though Look-up tables are fast but they need large amount of memory for accurate results. To get the desired accurate result the usage of power series is a bit time consuming as it is very slow. One of the DSP algorithms is CORDIC, this has been proposed to give efficient hardware solutions [2]. The estimation of CORDIC resembles the compromise between power series & look-up tables, in this the high precision results have been saved without usage of any on chip memory. In real time the architectures of high speed VLSI becomes more reality for modern DSP systems with the advancement in VLSI technology. This contributes the CORDIC algorithm to map easily into hardware designs suitable to the need of designers. Now-a-days the particular types of designs are usually used in DSP systems, which have a huge change of growth in their performance. Lot of performance parameters like speed, power and area are obtained by optimizing these structures. Whereas advisable hardware has to be chosen to implement this type of optimized structures.

System designers of IC technology have been provided a wide variety of implementation formats. The implementation format gives the brief about how the switching elements should be organized and how the functionality of a system will be analysed. Moreover the most advanced implementation platform used in today's life time is FPGA platform. For implementation of CORDIC architectures, these FPGAs are often used as co-processors. FPGAs have been slow, less energy efficient and generally achieve less functionality when compared to ASIC.

2. CORDIC ALGORITHM

CORDIC theory is an algorithm for vector rotation. In 1959 Jack. E. Volder proposed this CORDIC algorithm. This uses only shift and add operations, it is an iterative process. It performs vector rotation by calling them as arbitrary angles.

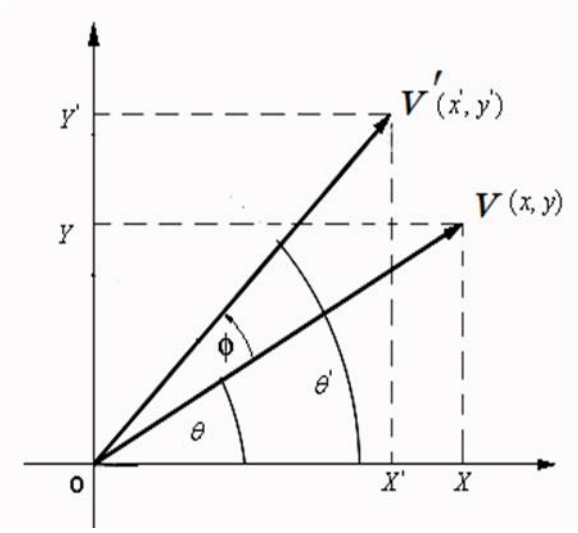

Fig 1 Vector rotation by Ø angle

Two vectors of same magnitude have been shown in fig 1. The second vector is obtained by rotating the first vector by angle

Ø. The first vector located at [*xy*] and new coordinates are [x' y']. The new coordinates [x' y'] can be calculated based on imposed vector and rotation angle Ø. The generalized equations were given by volder to calculate these [4].

$$x' = x\,cos\emptyset - y\,sin\emptyset \quad (1)$$

$$y' = y\,cos\emptyset + x\,sin\emptyset \quad (2)$$

The above can also be written as

$$x' = cos\emptyset\,(x - y\,tan\emptyset) \quad (3)$$

$$y' = cos\emptyset\,(y + x\,tan\emptyset) \quad (4)$$

Restrict the rotation of angles to $tan\emptyset = \pm 2^{-i}$. where i ranges from 0 to ∞.This will decrease the tangent multiplication by simple shift operation. But this applicable to only specific angles.

$$\cos\emptyset = \frac{1}{\sqrt{1+\tan(\emptyset)^2}} = \frac{1}{\sqrt{1+(\pm 2^{-i})^2}} = k_i \quad (5)$$

Where k_i is the scale of constant.

Slightly change the last set of equations (3) & (4) to reflect it will be rotated by a positive or negative angle of 2^{-i}. let us introduce d_i for multiplication whose value is +1 for positive angle and -1 for negative angle. d_i is known as Decision Function. Now the equations obtained are

$$x' = k_i[x - y\,d_i 2^{-i}] \quad (6)$$

$$y' = k_i[y + x\,d_i 2^{-i}] \quad (7)$$

Where i will be number of rotations and Z_i is the desired angle of rotation. In general the above equations (6) & (7) used for iterative process to rotate by any angle.

From yields of obtained equations of vector rotation shift-add algorithm eliminates the constant scale value. k_i is the product term applied in the system. By using infinite number of iterations the product value reaches to 0.6073. A_n is denoted as gain of rotation algorithm.

$$A_n = \pi[1 + 2^{-2i}]^{1/2} \quad (8)$$

The gain is approximately equal to 1.647, for infinite iterations. However by the help of total number of iterations gain can be calculated. Composite rotation angle is distinct by sequence of elementary rotations in the directions manner. All these sequences are denotes as decision vector.Based on the values of binary arctangents all these vectors used for angular measurement system. By using a look-up table between angular system and any other systems conversions are done. Convenient angular unit expressed in angles. The angular values are supplied by small look-up table, depending on suitable implementation we can also use hardwired.

3rd value of difference equation is added by accumulator angle to CORDIC :

$$z_i - \alpha_i = z_{i+1} \quad (9)$$

$$z_{i+1} = z_i - d_i tan^{-1}(2^{-i}) \quad (10)$$

CORDIC micro-rotation equations can be written as:

$$x_i - y_i 2^{-i} d_i = x_{i+1} \quad (11)$$

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

$$z_i - d_i tan^{-1}(2^{-i}) = z_{i+1} \quad (13)$$

3. CORDIC ARCHITECTURES

Depending on hardware realization of iterative process CORDIC architectures are mainly classified as Folded (sequential) and Unfolded (Combinational) [5]. By the direct duplication of equations 15,16 and 17 the folded architecture has been obtained. All the iterations should approve in a single functional unit for that in time domain the sequential architecture has to be multiplexed. In signal processing folding provides a measure for trading area for speed [6]. These folded architectures are classified as bit-serial and word-serial architectures

3.1 Folded Word Sequential Design

By duplicating each of the three difference equations in hardware the folded word sequential design [7] is obtained, this is also called as iterative bit parallel design. The hardware of this design is shown in figure 2. Each branch of this architecture consists of an adder-subtraction unit, a shift unit and a register, which is used for buffering the output. At the first stage of calculation initial values are given to register by using multiplexer. The mode of adder-subtraction unit will be determined by using the MSB value stored in the z-branch. Signals in x and y branches passes through the shift units and those will be added or subtracted from the un-shifted signal in the opposite path. Look up table addresses will be changed according to the number of iterations.

The z branch combines the registers values and values taken from look up table arithmetically. The output is mapped back to the registers for n iterations, before the initial values fed in again and at the output these final values can accessible. For controlling the multiplexers, shift distance and addressing of the constant values a simple finite-state machine is needed. The initial values for the vector coordinates as well as the constant values in the LUT can be hardwired in a word manner, when implemented in an ASIC. Adder and the subtractor component will be carried out separately. Multiplexer is controlled by the sign of the angle accumulator, which distinguishes between the addition and subtraction by routing the signals as required. The shift operations, the number of iterations changes the shift distance, but for this high fan in is required and it reduces the maximum speed for the application [4]. If implementation requires several layers of logic, these shifters do not map well in the FPGA architectures [8]. This results in slow design

which uses large number of logic cells. In addition to this, the output rate is limited by the operations performed iteratively. This results the maximum output rate equal to 1/n times of the clock rate.

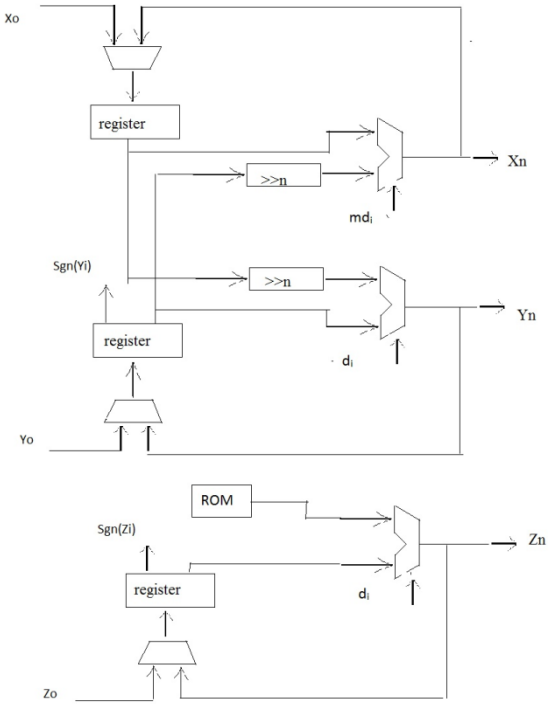

Figure 2. Folded Word Sequential Design

3.2 Unfolded Parallel Design

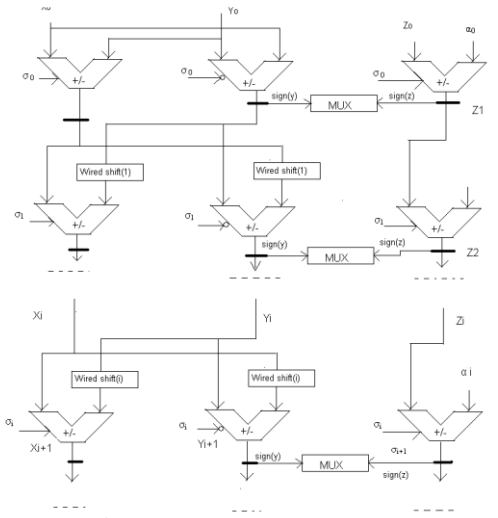

Figure 3. Unfolded Parallel Design

The above given CORDIC architecture is an iterative algorithm, that is the processor needs to perform iterations at n times the data rate. Each of n processing elements always performs the same iteration that is the iteration process can be unfolded. The unfolded parallel design architecture has been shown in figure 3. This architecture results in two significant simplifications. First, in these shifters performs fixed shifts, so these can be implemented in the wiring. Second, in the accumulator chain the look up values for the angle accumulator are distributed as constants to each adder. Instead of requiring storage space those constants can be hardwired. The complete CORDIC architecture has reduced to an array of interconnected adder-subtraction units. Need of registers has been eliminated, making this as unrolled processor strictly combinational. By the iterative circuit in the architecture the processing time has been reduced. The unfolded architecture can easily pipelined [9] by inserting registers between the adder-subtraction units.

4. REDUNDANT ARITHMETIC

In CORDIC processor adder plays the major role. In adders due to carry propagation the delay increases rapidly and the speed of operation will be slows down. So that we will go for redundant arithmetic, by this method the delay will be reduced and the speed of the operation increases. The tasks like addition, subtraction, multiplication produces carry propagation chains. Redundant number system is proposed to resolve this problem [10].

Arithmetic operations speed will be increased by redundant number system. Foe sign processing and additional applications this method is used. Area in VLSI and power dissipation will be decreased by this method. For numerically intensive applications redundant number system is suitable. By generating parallel adders with delay of constant value RNS can prevent or captures the carry propagation. This will not depend on the operand word length.

Thus RNS format produces low latency results. By using redundant number system, computational performance in numerically intensive applications can be improved. However, multiple bits are needed for each symbol (digit) so the implementation of their arithmetic circuits is usually expensive. To eliminate carry propagation these circuits use redundancy representation, irrespective of the operand width they provide near constant addition delay.

4.1 Carry Free Radix-2 Addition

Carry propagation is limited to a few bit positions in

the carry propagation of redundant number system[12], where it is generally independent of the word length W. the algorithm used for carry out signed binary digit addition is not a unique one, so the logic implemented for this can be diverse.

Using 2 unsigned binary numbers a radix-2 signed digit number is coded, as positive one and a negative one. As $X = X^+ - X^-$. hence, using 2 bits each signed digit is represented as $x_i = x_i^+ - x_i^-$, where, $x_i^+, x_i^- \epsilon \{0,1\}$, and $x_i \epsilon \{1,0,1\}$.

1.1 Hybrid Radix-2 Addition

First input and output operands are in redundant signed digit representation and the second input operand is conventional unsigned number in hybrid operation. Consider an unsigned conventional number Y and the addition of a radix-2 signed digit number $X_{<2.1>}$, where 2 indicates the radix-2 operation and 1 indicates the largest digit in the digit set is 1.

$$S_{<2.1>} = X_{<2.1>} + Y \qquad (4a)$$

Table 1. Redundant Number System of Radix-2

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

2 steps are required for this addition. First step will be carried out in parallel for all bit positions of i ($0 \le i \le W\text{-}1$), where W is the word length. An intermediate sum which is $P_i = x_i + y_i$ is calculated, the range of this intermediate sum lies in the range {1, 0, 1, 2}. This addition can be expressed as:

$$X_i + Y_i = P_i = 2t_i + u_i \qquad (4b)$$

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, 0, 1}	$S_i^+ - S_i^-$

Where t_i is transfer whose values are either 0 or 1 and it is denoted as t_i^+ ; and u_i is the interim sum whose values are either 1 or 0 and it is denoted as $-u_i^-$. The value assigned to the least significant transfer digit t-1 is zero, which is same as the MSB of the interim sum digit u_i.

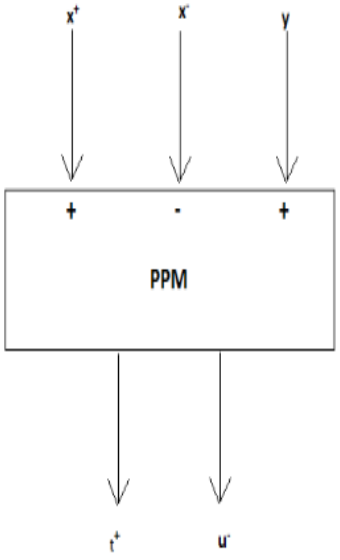

Figure 4. Hybrid Radix-2 PPM Adder

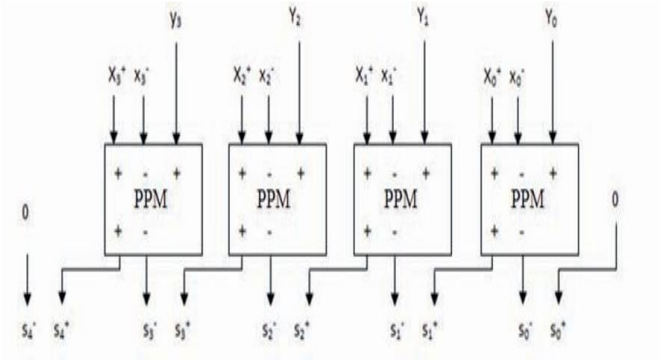

Figure 5. Four Digit Hybrid Radix-2 Adder

By combining t^{+i-1} and u_i^- as one digit the sum digit s_i is formed. This is the second step.

$$S_i = t^+_{i-1} - u_i^+ \qquad (4c)$$

Equation (4c) is replaced by corresponding binary codes from table 4.2

$$x_i^+ - x_i^- + y_i^+ = 2t_i^+ + u_i^- \qquad (4d)$$

PPM adder can perform this arithmetic operation of a generalized type-1 full adder, which is shown in figure 4. The structure of hybrid 4-digit radix-2 adder is shown in the fig 5. The novel architectural design of CORDIC processor using redundant arithmetic logic has shown in the figure 6.

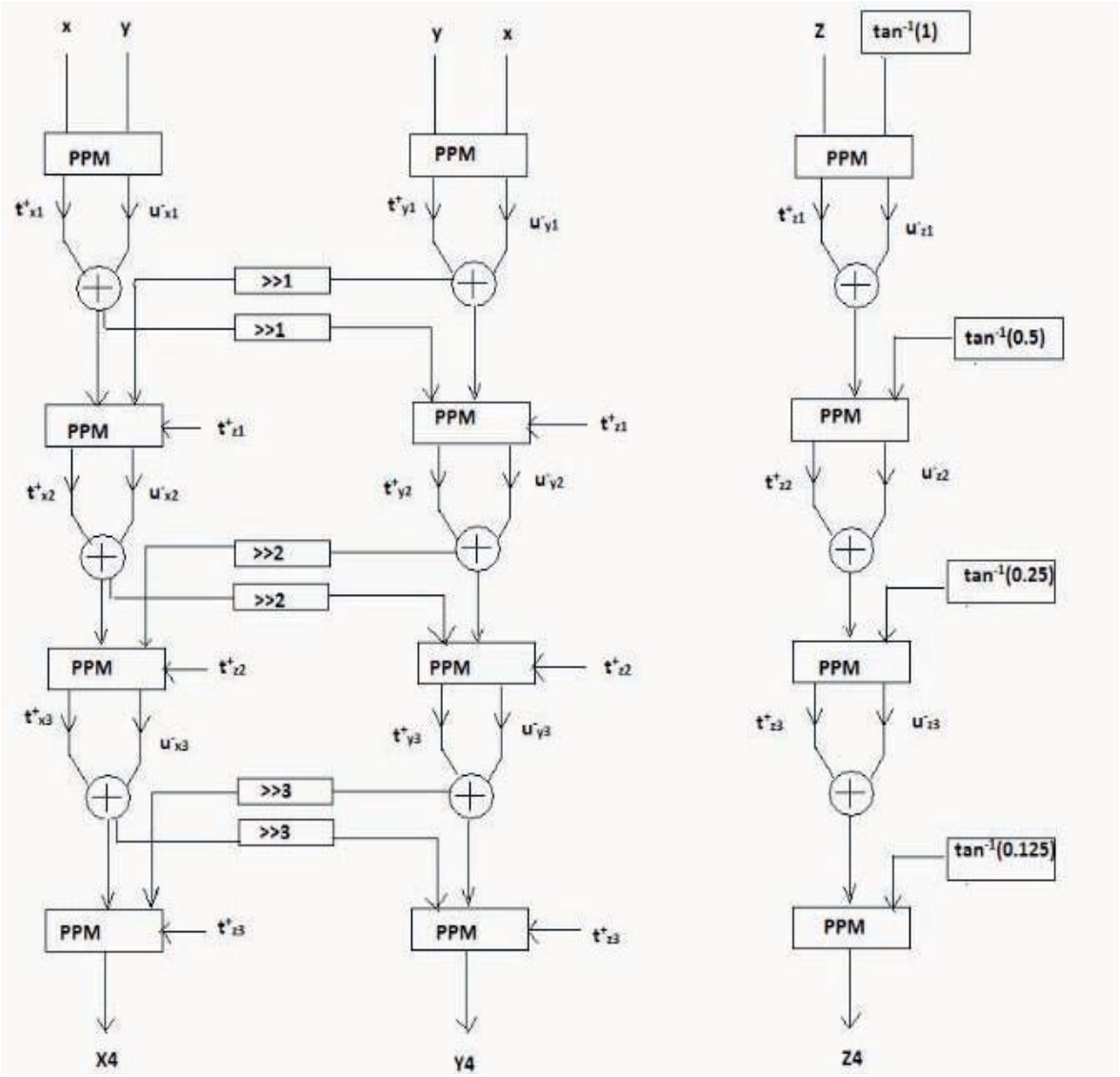

Figure 6. A Novel Architecture for CORDIC Processor Using Redundant Arithmetic

5. IMPLEMENTATION RESULTS

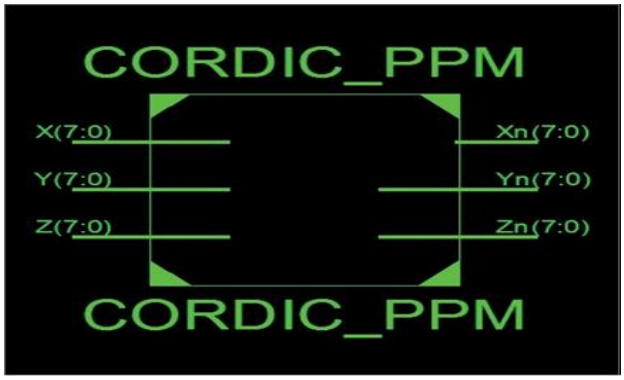

Figure 7. Block View of Novel Architecture (RA-CORDIC)

The proposed novel architecture of RA-CORDIC is implemented for four stages. This design is performed by using VERILOG and simulation results carried out by ISIM (ISE Simulator)[13]. Figure 7 shows the simulation result of novel architecture of RA-CORDIC processor. The synthesis analysis of novel architecture is carried out by RTL Compiler. Figure 8 shows the block view of novel architecture.

Figure 9 shows the RTL Schematic view of ASIC implementation and Figure 10 shows RTL schematic view of a PPM module of ASIC implementation. Figure 11 gives the RTL power analysis of Redundant Arithmetic-CORDIC. Figure 12& 13 gives Timing analysis and net power usage of a Redundant Arithmetic CORDIC.

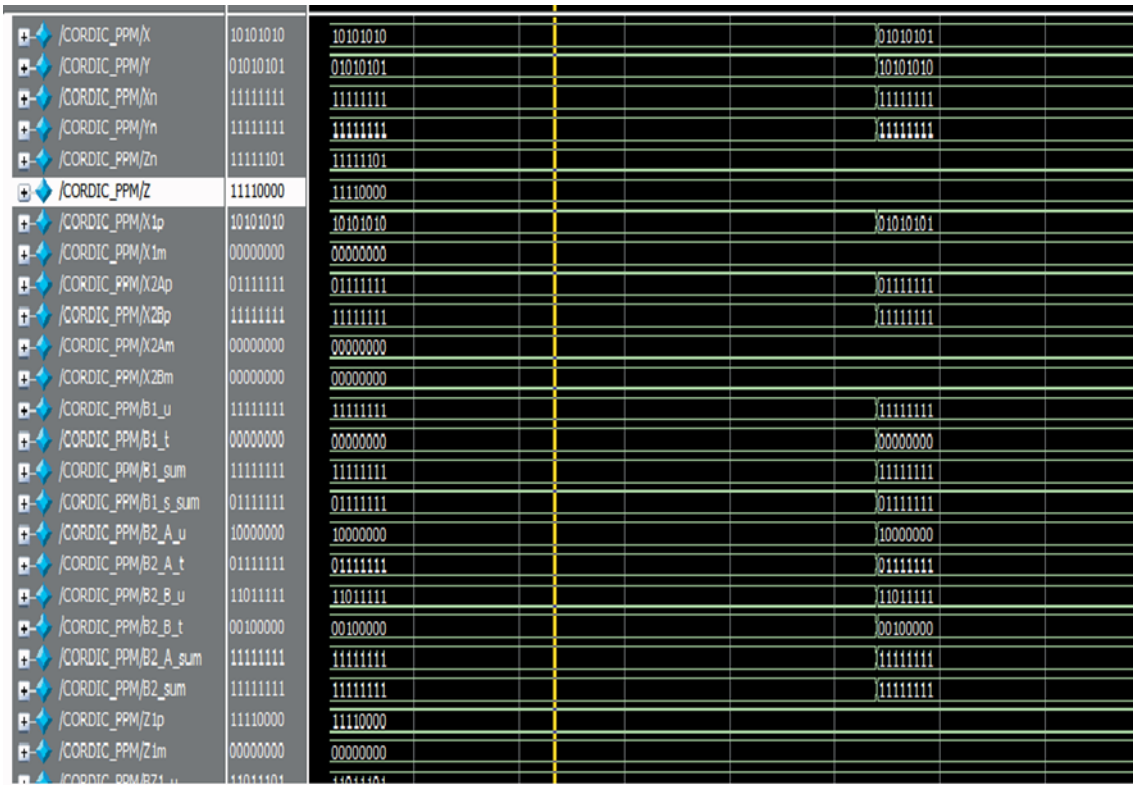

Figure 8. Simulation View of a Novel Architecture (Redundant Arithmetic-CORDIC)

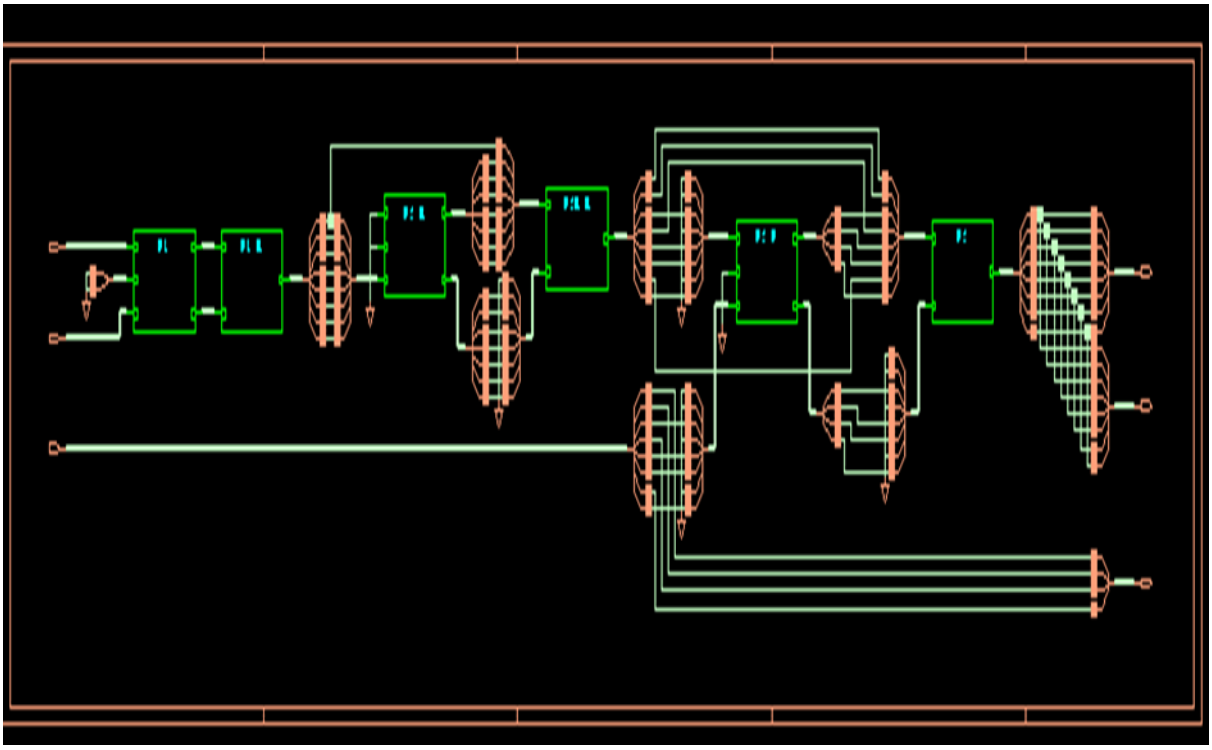

Figure 9. RTL Schematic View of RA-CORDIC

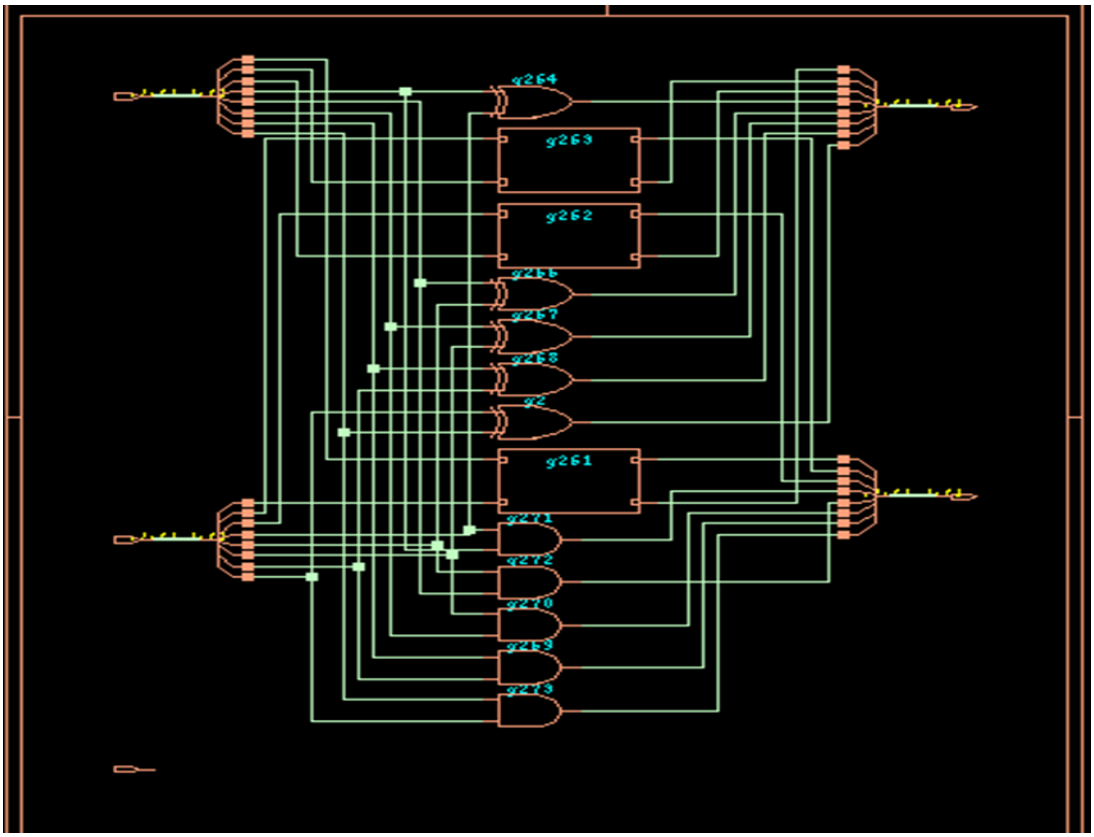

Figure 10. RTL Schematic View of PPM Module

File	Row	Leakage (nW)	Internal (nW)	Net (nW)
/export/home/cad09/CORDIC_PPM.v	18	1.05	252941.43	2529
/export/home/cad09/CORDIC_PPM.v	19	1.96	369460.77	3694
/export/home/cad09/CORDIC_PPM.v	27	0.86	381145.59	3811
/export/home/cad09/CORDIC_PPM.v	28	1.75	158656.97	1586
/export/home/cad09/CORDIC_PPM.v	30	0.49	107689.61	1076
/export/home/cad09/CORDIC_PPM.v	31	1.37	66674.45	666

Figure 11. RTL Power Analysis of RA-CORDIC

Pin	Type	Fanout	Load (fF)	Slew (ps)	Delay (ps)	Arrival (ps)
(clock clk)	launch					0.0
(in_del_1)	ext delay				0.0	0.0
X[1]	in port	2	69.9	0.0	0.0	0.0
B1/ap[1]						
g268/B					0.0	0.0
g268/Y	XOR2X1	3	62.9	239.3	225.3	225.3
B1/u[1]						
B1_A/x1[1]						
g8688_dup/B					0.0	225.3
g8688_dup/Y	NOR2X1	1	27.0	117.8	125.2	350.5

Figure 12. Timing Analysis of RA-CORDIC

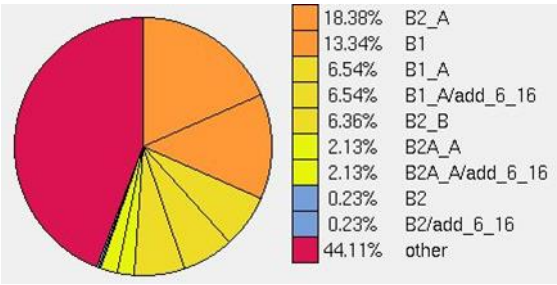

Figure 13. Net Power Usage of RA-CORDIC

6. CONCLUSION

A novel architecture for CORDIC by using redundant arithmetic (RA-CORDIC) is presented in this paper. Mainly inhefty operand of DSP and high speed applications, the above novel architecture can be used. The novel architecture has been implemented and targeted for ASIC devices. This design offers a better latency and achieves maximum throughput rate.

REFERENCES

[1] KhurshidB,RatherGMetal.(2012). "Performancecomparison 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, volEC-8(3),330–334.

[5] Meggitt E (1962). Pseudo division and pseudo multiplica- tion 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] AbidZ, and Wang W(2008), New designs of redundant-binary full adders and Its applications, IEEE International Symposium on Circuits and Systems, ISCAS, 3366–3369.

[13] ISE Simulator (2011), Xilinx incorporation San Jose, U.S.A.

Authors

Katakam Supraja Harsha received the B.Tech degree from the Department of Electronics and Communication Engg. Avanthi Institute of Engineering & technology, Rangareddy (TS)- India in the year of 2015. She is currently pursuing the M.Tech (VLSI-SD) in CVR College of Engineering, Rangareddy (TS), India. Her research interest includes Analog Communication & VLSI.

Subha Sri Lakshmi Thiruveedhi received the B.Tech degree from the Department of Electronics and Communication Engg. Bhoj Reddy Engineering College for women, Hyderabad (TS)- India in the year 2011. Received M.Tech Degree (VLSI Design) in School of Computing, SASTRA UNIVERSITY, Thanjavur (TN)- India, in the year 2013. She so far had 4 years of teaching experience in Engg. Institution. She is currently working as an Assistant Professor at CVR College of Engineering. Moreover she has 11 Reputed International journal publications, 2 International Conferences, 1 National Conference publication. Her research interests include High Performance VLSI Architectures, Signal Processing, Communication, Electron Devices and Circuits and VLSI Backend Design.