http://www.jctjournals.com

ISSN(ONLINE):2278 - 3814

Volume-11, Issue 04, April 2022

JOURNAL OF COMPUTING TECHNOLOGIES (JCT)

International Journal

Page Number: 01-09

Design & Implementation of VLSI Architecture of 2^5 bit Approximate Square Root for DSP-FPGA Applications

Smita Nagle[1], Prof. Nishi Pandey[2], Prof. Abhishek Agwekar[3]

[1]PG Scholar(M.Tech) , [2]Assistant Professor,[3]HOD, Department of EC

TRUBA College of Science and Technology, Bhopal

[1]smitanagle123456@gmail.com, [2]pandeynishi738@gmail.com, [3]abhiagwekar@gmail.com

***Abstract*— The terrifying sight of a square root symbol can create a challenging mathematical challenge, with square root problems not as difficult to solve as they may first appear. Simple square root problems can often be easily solved as basic multiplication and division problems. Complex square root problems, on the other hand, may require some work, but with the right approach, these too can be easy. This dissertation introduces the implementation of the VLSI Architecture of 32 bit almost square root for DSP-FPGA applications. We are proposing a new algorithm that does not return square root that does not require square roots or multiplexors. Compared to previous square root algorithms and our algorithm will work well in using VLSI. It will generate the correct amount of effect on each repetition and does not require additional rotation to adjust the output bit. Working on each repetition will be easy: adding or subtracting based on the bit of output generated in the previous repetition. Simulation and integration is done using Xilinx ISE 14.7 software with verilog code. Imitation results indicate that the proposed code is an extension of the existing function. The proposed square root is used for 32 bit square root and the previous one is for 16 bit. The total area is set for a 245 per share or 2.66% while the previous one is 536. The delay is 3.01ns while the previous one is 3.69ns. The upgraded frequency is 293 MHz as proposed and 107.50 MHz for the previous one. The proposed VLSI square root code therefore provides better performance within calculated parameters.**

***Keywords*— Adaptive active damper, grid connected inverters, harmonic current reference compensation technique, PI controller, DSMPI Controller.**

I. INTRODUCTION

Flexible floating point functions are widely used in many fields of computer engineering and IoT. The arithmetic functions of the floating point are included in most processing units. There are many functions of floating points that include addition, subtraction, multiplication, division, multiplication and square root. The Northeastern Reconfigurable Computing Lab has established its own VFLOAT, vender agnostic, easy-to-use and easy-to-trade interface between hardware resources, high clock frequency and delays. Field Programmable Gate Arrays (FPGAs), due to its flexibility, low power consumption and short development time compared to Application Specific ICs (ASICs), are selected as the VFloat library platform in which you will operate. High-speed integrated circuits Computer Definition Language (VHDL) is used to define these components. Xilinx and Altera are two of the largest providers of smart devices. Each company has its own Integrated Development Environment (IDE). Both IDE from Altera and Xilinx were used to implement this cross-platform project. There are two main ways to represent the numerical value of a computer. One is a fixed point format, and the other is a floating point format. The difference between the two methods is that the first one has a radix point with a fixed point separating the whole and part parts of the numerical value. The floating point format has three parts that represent a number of numbers: a signed beat, an expectation and a mantissa. The advantage of using a floating point format is that the floating point can represent a wider range of values than the fixed point format when using the same number of bits. The floating point format has been established and is widely used in scientific calculations. This reduces confusion and improves the quality of numbers. IEEE 754 is a floating point calculator technology developed by the Institute of Electrical and Electronics Engineers (IEEE) in 1985 and revised in 2008 [2]. At this level, the arithmetic format of the floating point number is defined as follows. We define b as a base that is probably 2 or 10, s as a symbol bit, e as an object and c as a mantissa. In our project, b = 2. Basic formats with floating binary points with one accuracy, double accuracy and four accuracy. The corresponding bit width of the symbol, object and mantissa of each format. Note that there are many combinations that are not included in IEEE 754. For example a bit of mark is 1, an element is 9 bits and mantissa has 30 bits. However, using this unconventional format can save some resources on flexible technologies

such as FPGA and still perform computer work. That is why a precarious floating space is also considered during the construction of our floating library. The number of floating point points is (- 1) s _ 1.c _ be − bias. First, if s is zero, the number is positive. If not, it is negative. Second, key digits are 1.c not c because the first digit of a significant digit of any normal floating point number is always '1 'and skipping it will save one bit of representation. Third, to make the spectrum range more attractive and easier to compare, bias is added to the actual exit representation representation in IEEE 754.

II. LITERATURE REVIEW

Z. Yan et al., [11] This function proposes a soft dual-mode dynamic multi-mode multi-mode information (MIMO) to help open the circle and close the circle in the advanced Ultra high throughput (EUHT) remote area (LAN). The proposed identifier uses a qualitative square error (MMSE) programmed QR decay (MMSE-SQRD) to deliver the channel's pre-processing results, approved by simultaneous systolic display engineering. Alternatively, the MMSE square-draw calculation in a closed circle re-uses MMSE-SQRD pre-processing to a large extent preserves the asset at the top. In addition, K-Best's enhanced recognition recognition is enhanced by an open circle, which creates an effect with irregular or parallel editing and results in a softer result with disposable methods (DPs). Flexible VLSI engineering is aimed at the proposed dual mode, holding 1 × 1 ~ 4 × 4 reception cables and BPSK ~ 64-QAM tweak settings. Made with the new SMIC 65 nm CMOS, the detector is equipped to operate at 550 MHz, with 2.64 Gb / s high performance for K-Best detection and 3.3 Gb / s for direct MMSE detection. The proposed identifier is critical to late-distributed activities and meets the EUHT level information requirements.

K. Chen et al., [12] Many current resource models (CSMs) have been proposed to investigate the entry point and time test of sub-90-nm CMOS systems over the past decade. However, most of them may have the negative consequences of major errors in long-term delayed logical interventions. This function introduces an extracted CSM that can provide maximum accuracy on both single-stage foundation doors and various integrating sections. The proposed CSM includes a voltage-controlled current source, information and parasitic power output, Mill operator power and input alignment power corresponding to defined information power. The input alignment power helps to display the information hub very accurately. In the tests, the proposed CSM beats the CSM benchmark with a common root-mean squared error (RMSE) and a common error to reverse the door from half to half. J. Lin et al., [13] In this brief case, a locked retrospective circuit (DLL) - an established explosive mode clock and a retrieval circuit (BMCDR) using a 4 × over-sampling process is approved by a nonprofit detection organization. With the help of DLL following the information section, the proposed region can recover explosive mode information in a short period of protection and achieve greater jitter resistance. In addition, a 2.5-GHz four-phase clock generator is installed on the chip. Made with 0.18-µm of new CMOS, try to show that the recovery time can be planted in an hour of 31 pieces. Approaches 2.5-Gb / s input information for 231-1 pseudorandom dual system, retrospective information with root-mean-square jitter 8.557 ps and high-resolution jitter 32.0 ps, and the error rate of the episode is intentionally less than 10 - 10. The total chip region is 1.4 × 1.4 mm 2, where the center of the BMCDR circuit includes 0.81 × 0.325 mm 2. The total power consumption is 130 mW from the inventory voltage of 1.8 V.

H. Amir-Aslanzadeh et al., [14] This work introduces a system, testing, and evaluating current 2.4-GHz self-guided automatic transmission for short-term remote control applications. Key commitments are the design / research design of the stamped power stamp (Father) / voltage-controlled oscillator (VCO), nonlinear duplication testing under Gilbert-based root-mean-square finder, and on-chip. LC - tank alignment circuit that does not require an advanced advanced converter (ADC) / computer signal processor. The sticky design reduces the number of controls required, uses an adequate supply headroom, and considers the alignment circle "ADC-less" which can dramatically intensify the repetition of my father's community by getting a transmitted signal. The real concept of direct control design blocks additional high-level non-standard signal generators, reduces complexity and allows web alignment. The frame is made of TSMC 0.18 µm CMOS, incorporating 0.7 mm 2 (TX) +0.1 mm 2 (self-adjusting), and is rated for QFN48 mass on the FR4 PCB. Naturally correcting the defects of the Baba / VCO tank in this case produced> 4 dB increase in yield capacity. With fixed tuning fluctuations, the transmitter transmits the power of the deliberate output> 0 dBm to a different load of 100-Ω, and the frame uses 22.9 Mama from the 1.8-V center circuit server.

K. Gupta et al., [15] This function introduces another real RMS-to-DC converter with a square view / square divider. The circuit uses MOSFETs that operate in a solid modified immersion system that generates a redesigned translinear circuit. The converter includes low cost creation, unaffected by physical impact, including the current two quadrant. RMS-to-DC Converter circuits are validated using the renamed TSMC 0.35µm CMOS.

B. Ramkumar et al., [16] Convey Select Adder (CSLA) is probably the fastest adder used in many information management processors to complete the tasks of combining quick numbers. Since the formation of the CSLA, it is clear that there are degrees of regional reduction and power consumption in the CSLA. This activity uses direct and productive transformation of the entry level to completely reduce the region and power of CSLA. In view of this change the CSLA (SQRT CSLA) 8, 16-, 32-, and 64-b square-root (SQRT CSLA) design has been created and compared to standard SQRT CSLA engineering. The proposed plan reduced the region and capacity as compared to the standard SQRT CSLA with only a slight extension of the reversal. This project examines the exhibition of proposed systems in relation to retrospect, region, power, and their components manually with intelligent effort

II.PROBLEM FORMULATION

A variety of computer arithmetic techniques can be used to implement a square root. Most techniques involve computing a set of partial products, and then summing the partial products together. In existing system computing the square root, there are three steps similar to the reciprocal computation. First step is reduction, second is evaluation, and the last step is post-processing. In the design of most commercial RISC processors, a square root is used for all iterations of div or sqrt instructions.There is still some of the limitation or challenges in square root detection code so the observed problem formulation is as followings

- The existing technique based circuit complexity is high for detection of the square root, it generate more number of carry adder.
- The square root requires a rather large number of gate counts.
- It is impractical to place as many square roots as required to realize fully pipelined operation for division and square root instructions.
- The existing circuits give more latency and consume more power during operation.

The main purpose of this study is to use the VLSI Architecture of 32 bit almost the square root of DSP-FPGA applications. We are proposing a new non-return square root algorithm that does not require square roots or multiplexors. Compared to previous square root algorithms and our algorithm will work well in using VLSI. It will generate the correct amount of effect on each repetition and does not require additional rotation to adjust the output bit. Working on each repetition will be easy: adding or subtracting based on the bit of output generated in the previous repetition.The main contribution of the proposed research work is as followings-

- To design of the 32 bit square root detection using non-restoring algorithm.
- To implement proposed algorithm using using Xilinx ISE 14.7 software with verilog language.
- To check result validation using Isim simulator in test bench window.
- To optimize the improvement in the performance parameters like total number of component required latency.

III.PROPOSED WORK METHODOLOGY

The proposed system proposes a new non-restoring square root algorithm that requires neither square roots nor multiplexors. Compared with previous square root algorithms, proposed algorithm is very efficient for VLSI implementation. This is feasible for the digital signal processing and field programmable gate array applications.

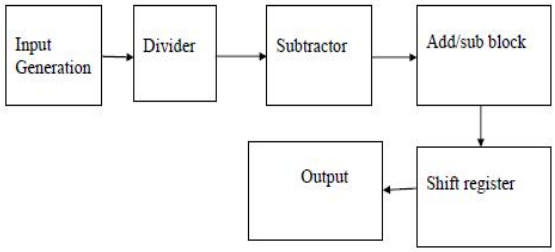


Figure 4.1: Flow Chart

Step-

Firstly assign the input numbers in form of binary, decimal or hexa form. Now perform the implementation process, it generates the register transfer level (RTL) and technological view.

Now simulate the results in the Xilinx test bench and run the simulation. The square root output will be generated.

If its fixed number then square root will be also fixed and accurate and of the number is not fixed or floating then square root output will be nearest fixed number.

4.1 ALGORITHM

New non-restoring square root algorithm that requires neither square roots nor multiplexors is presented. It generates the correct resulting value at each iteration and does not require extra circuitry for adjusting the result bit. The operation at each iteration is simple: addition or subtraction based on the result bit generated in previous iteration. The remainder of the addition or subtraction is fed via registers to the next iteration directly even it is negative. At the last iteration, if the remainder is non-negative, it is a precise remainder. Otherwise, we can obtain a precise remainder by an addition operation.

The focus of the new algorithm is on the partial remainder with each iteration. The algorithm generates a correct resulting bit in each iteration including the last iteration. The operation during each iteration is very simple: addition or subtraction based on the sign of the result of previous iteration. The partial remainder generated in each iteration is used in the next iteration even it is negative. Therefore it achieves high speed at minimum cost because neither a square root nor a multiplexor is required.

IV.RESULT AND ANALYSIS

The ISE Design Suite is the Xilinx® design environment, which allows you to take your design from design entry to Xilinx device programming. With specific editions for logic, embedded processor, or Digital Signal Processing (DSP) system designers, the ISE Design Suite provides an environment tailored to meet your specific design needs.

Xilinx ISE (Integrated Software Environment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

ISSN(ONLINE):2278 - 3814

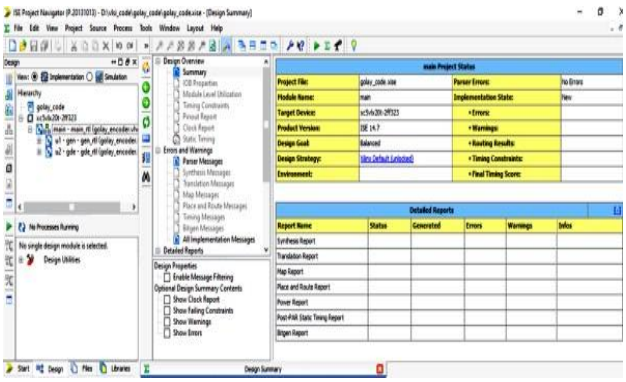


Figure 5.1: Snap shot of Xilinx

ISE Design Suite: Logic Edition

The ISE Design Suite: Logic Edition allows you to go from design entry, through implementation and verification, to device programming from within the unified environment of the ISE Project Navigator or from the command line. This edition includes exclusive tools and technologies to help achieve optimal design results, including the following: PlanAhea software - allows you to do advance FPGA floor planning. The PlanAhead software includes PinAhead, an environment designed to help you to import or create the initial I/O Port list, group the related ports into separate folders called “Interfaces” and assign them to package pins. PinAhead supports fully automatic pin placement or semi-automated interactive modes to allow controlled I/O Port assignment. With early, intelligent decisions in FPGA I/O assignments, you can more easily optimize the connectivity between the PCB and FGPA.

5.2SIMULATION RESULTS AND DISCUSSION

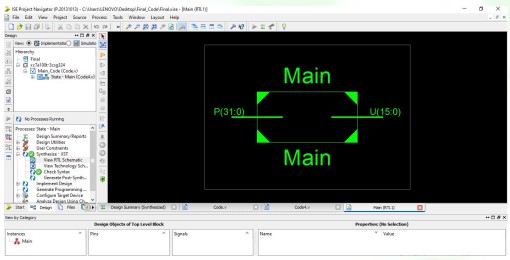


Figure 5.2: RTL View of Top module

Figure 5.2 presents the top level view of the proposed square root VLSI implementation. P stands for the input which is 32 bit and U stands for the output which is 16 bit.

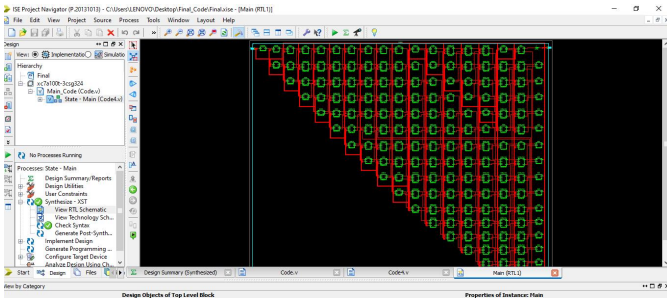


Figure 5.3: Complete RTL View

Figure 5.3 presents the complete RTL or register transfer level view of the proposed square root circuit.

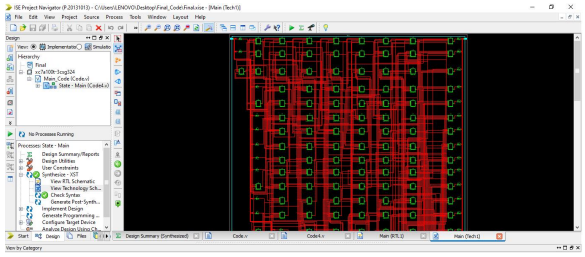


Figure 5.4: Technological View

Figure 5.4 shows the technological view of the 32 bit square root. Here red color shows the wires and green color shows the logic gates.

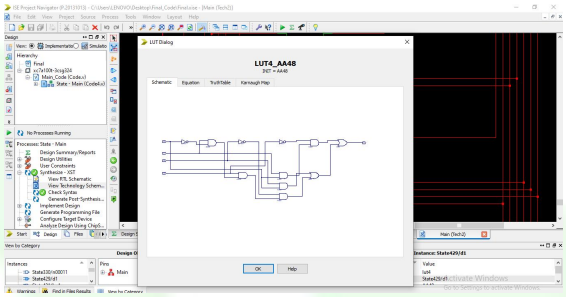


Figure 5.5: Look up table 4

Figure 5.5 is showing the look up table LUT4_AA48. The logical function in various combinations is carried out by the chip using the Lookup Table. Any combinatorial logic function can be implemented in a lookup table.

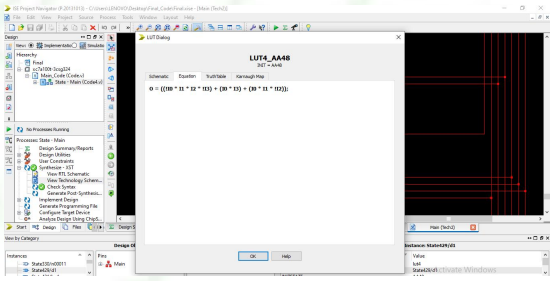
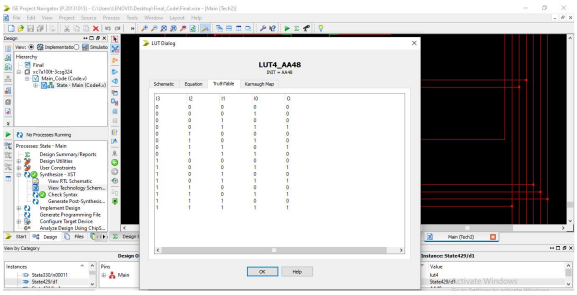


Figure 5.6: Equation of LUT

Figure 5.6 is showing the LUT equation by which the gate circuit’s view is generated.



ISSN(ONLINE):2278 - 3814

Figure 5.7: Truth table of LUT4

Figure 5.7 is showing the truth table of LUT4. Truth table shows the truth-value of one or more compound propositions for every possible combination of truth-values of the propositions making up the compound ones.

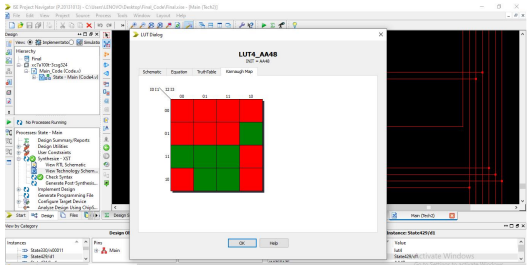


Figure 5.8: Karnaugh map

Figure 5.8 is showing the K- map (Karnaugh map) is a method of simplifying the Boolean expressions. Sometimes in practical problems, we need to find the expressions with minimum variables to minimize the cost of gates required. K-map is a tool that is used to minimize the output expressions of 3, 4, 5 or 6 variables

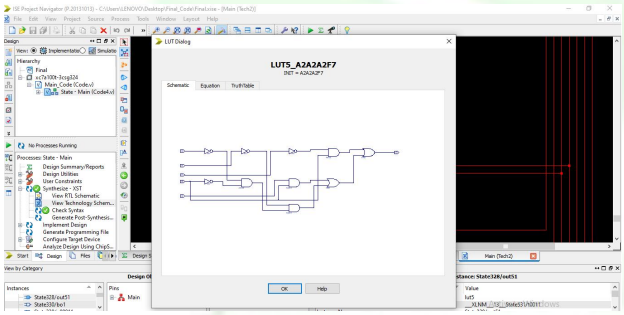


Figure 5.9: LUT5

Figure 5.9 is showing the LUT5, Lookup Tables in FPGAs help to significantly reduce the costs of computation and operation as it offers the computational range without requiring the time and massive gate count.

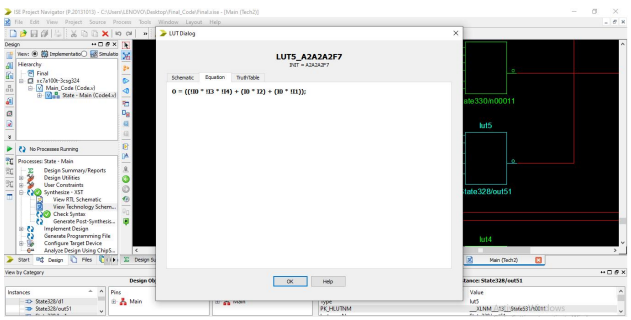


Figure 5.10: LUT5 equation

Figure 5.10 is showing the LUT equation by which the gate circuit's view is generated.

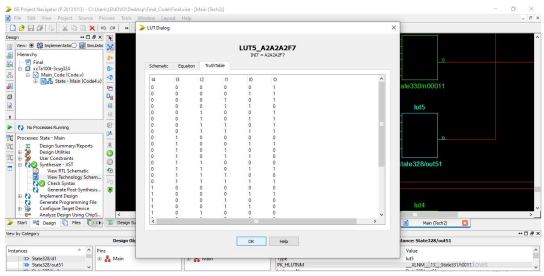


Figure 5.11: Truth table of LUT5

Figure 5.11 is showing the truth table of LUT5. Truth table shows the input and output combination of the gate circuit.

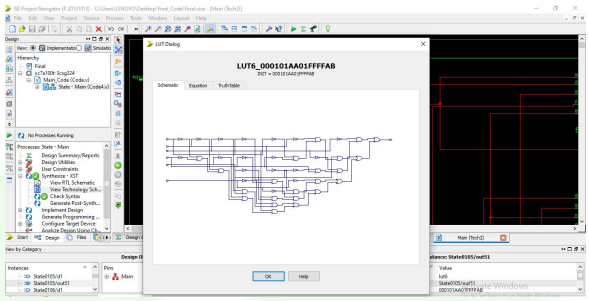


Figure 5.12: Look up table 6

Figure 5.12 is showing the look up table LUT6. The logical function in various combinations is carried out by the chip using the Lookup Table.

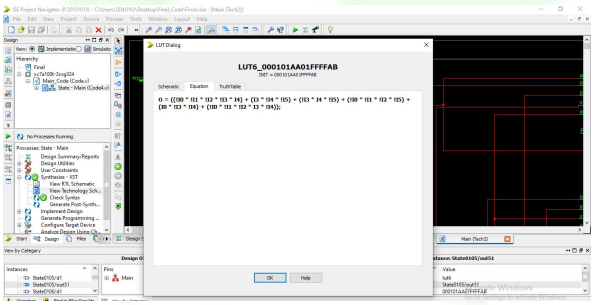


Figure 5.13: LUT6 equation

Figure 5.13 is showing the LUT equation by which the gate circuit's view is generated.

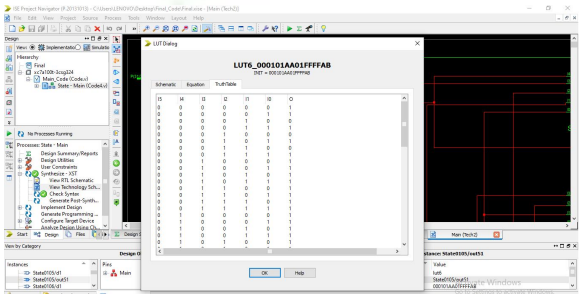


Figure 5.14: Truth table of LUT5

Figure 5.14 is showing the truth table of LUT5. Truth table shows the input and output combination of the gate circuit.

ISSN(ONLINE):2278 - 3814

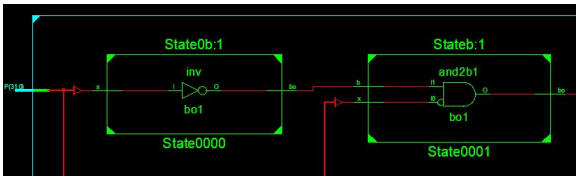


Figure 5.15: Invert state

Figure 5.15 is showing the internal circuit block, here invert logic gate is connected the AND gate. This contains the information of the initial state and final state.

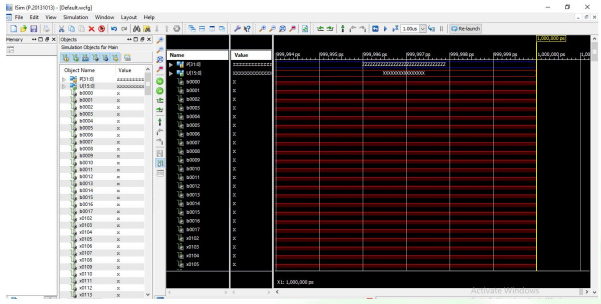


Figure 5.16: High Impedance test bench

Figure 5.16 presents the high impedance of the test bench, to represent the high impedance use symbol X or Z.

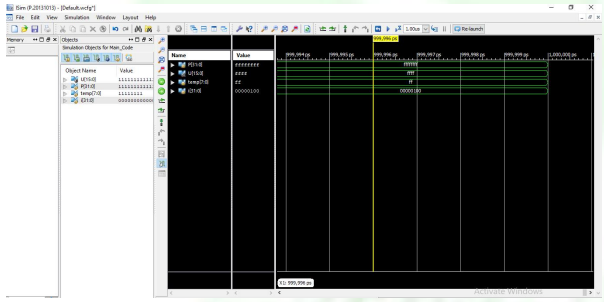


Figure 5.17: Test bench results -1

Figure 5.17 provides the test bench results. The bits are in hexa form. Input (P)= ffffffff and after the simulation the output (U) generate the ffff.

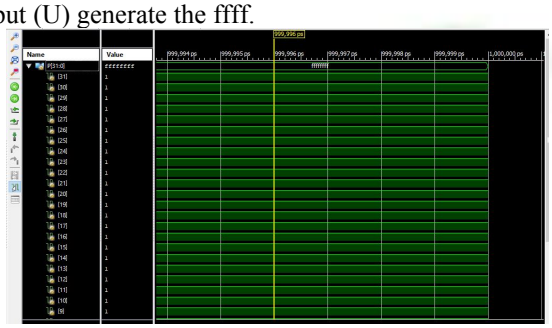


Figure 5.18: Signal of Input (P)

Figure 5.18 is presenting the signal status of the input (P), the logic 1 presents the high signal with green colour and logic o present the low signal with black colour.

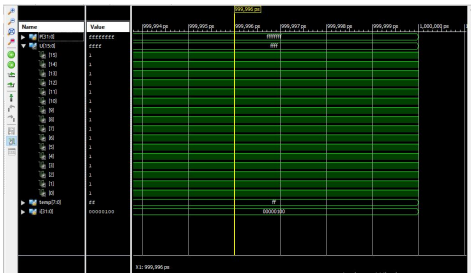


Figure 5.19: Signal of output (U)

Figure 5.19 is presenting the signal status of the output (U), the logic 1 presents the high signal with green colour and logic o present the low signal with black colour.

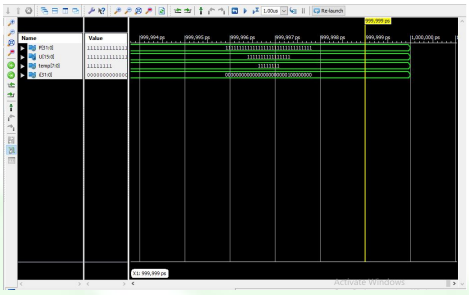


Figure 5.20: Test bench results-2

Figure 5.20 is showing bits are in binary form.
Input (P) = 11111111111111111111111111111111
Output (U) = 1111111111111111

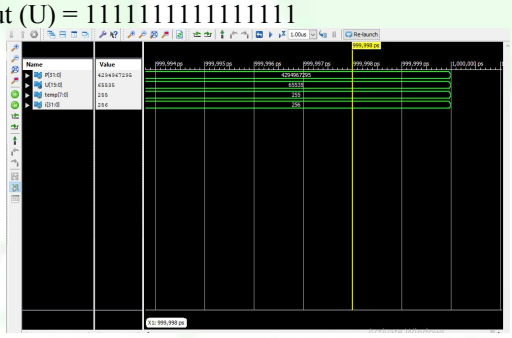


Figure 5.21: Test bench results-3

Figure 5.21 shows numbers in the decimal form for the square root calculation.
Input (P) = 4294967295 (not fixed number)
Output (U) = 65535 (nearest square root)

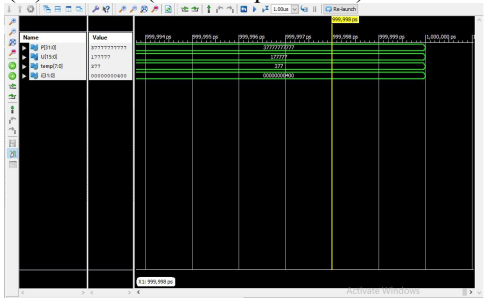


Figure 5.22: Test bench results-4

Figure 5.22 shows input number in octal form. Input = 37777777777
Output= 177777

ISSN(ONLINE):2278 - 3814

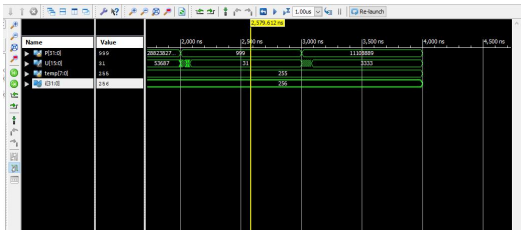


Figure 5.23: Test bench results-5

Figure 5.23 shows non fixed number. Input is 999 and nearest square root is 31.

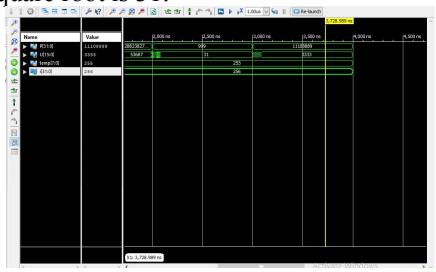


Figure 5.24: Test bench results-6

Figure 5.24 shows fixed number. Input is 11108889 and perfect square root is 3333.

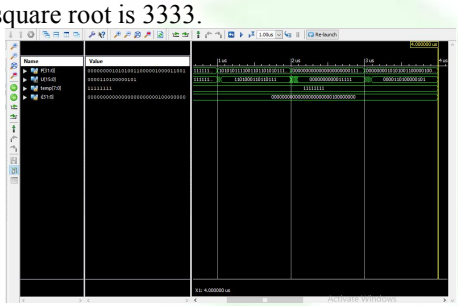


Figure 5.25: Test bench results signal

Figure 5.25 is showing the all previous and current state of the square root input and output signal or values.

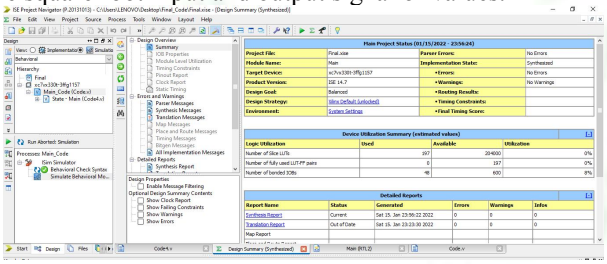


Figure 5.26: Summary in Xilinx window

Figure 5.26 shows the results summary in the Xilinx environment. The behavioral check syntax is successfully and simulated by using the ISim simulator.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	197	204000	0%
Number of fully used LUT-FF pairs	0	197	0%
Number of bonded IOBs	48	600	8%

Figure 5.27: Using component or area utilization

Figure 5.16 is showing the area utilization of the proposed square root. The slice look up table used 197 out of 204000 that is aprox 0%, the Look up table (LUT) flip-flop pair using 0%, bonded input output box using 48 out of 600 so it is 8%. Therefore the total required component is 245 or 2.66 %.

5.2.1 Synthesis Report

After successfully implementation and simulation of the proposed golay code, the synthesis report is generated to analyses the various performance parameters, which are followings-

Primitive and Black Box Usage:

BELS : 197
LUT2 : 6
LUT4 : 50
LUT5 : 46
LUT6 : 95
IO Buffers : 48
IBUF : 32
OBUF : 16

Device utilization summary:

Selected Device : 7vx330tffg1157-3
Slice Logic Utilization:
Number of Slice LUTs: 197 out of 204000 0%
Number used as Logic: 197 out of 204000 0%
Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 197
Number with an unused Flip Flop: 197 out of 197 100%
Number with an unused LUT: 0 out of 197 0%
Number of fully used LUT-FF pairs: 0 out of 197 0%
Number of unique control sets: 0
IO Utilization:
Number of IOs: 48
Number of bonded IOBs: 48 out of 600 8%

Timing Report

Minimum period: No path found
Minimum input arrival time before clock: No path found
Maximum output required time after clock: No path found
Maximum combinational path delay: 34.215ns
Timing constraint: Default path analysis
Total number of paths / destination ports: 699710430976924 / 16

-

Delay: 34.215ns (Levels of Logic = 72)
Source: P<30> (PAD)
Destination: U<0> (PAD)
Data Path: P<30> to U<0>
Gate Net
Cell:in->out fanout Delay Delay Logical Name (Net Name)
---------- ------------
IBUF:I->O 15 0.000 0.607 P_30_IBUF (P_30_IBUF)
LUT6:I1->O 35 0.043 0.451 State330/n00011 (State330/n0001)
LUT6:I4->O 4 0.043 0.564 State227/out51 (b1227)
LUT6:I0->O 35 0.043 0.564 State229/n00011 (State229/n0001)
LUT5:I1->O 2 0.043 0.294 State226/d1 (x1226)
LUT6:I5->O 4 0.043 0.564 State126/out51 (b1126)
LUT6:I0->O 32 0.043 0.535 State128/n00011 (State128/n0001)
LUT6:I3->O 2 0.043 0.461 State125/d1 (x1125)
LUT5:I1->O 4 0.043 0.564 State025/out51 (b1025)
LUT5:I4->O 4 0.043 0.564 State0419/out51 (b0419)

LUT6:I0->O 22 0.043 0.524 State0421/n00011 (State0421/n0001)
OBUF:I->O 0.000 U_0_OBUF (U<0>)

Total 34.215ns (3.010ns logic, 31.205ns route)
(8.8% logic, 91.2% route)
Total REAL time to Xst completion: 29.00 secs
Total CPU time to Xst completion: 29.23 secs

Table 5.1: Simulation Results

Sr No	Parameter	Value
1	Area	245 or 2.66 %
2	Total Delay	34.21 ns
3	Logic delay	3.01 ns
4	Power	0.45 mw
5	Power Delay Product	1539
6	Frequency	293 MHz
7	Throughput	9.3 GHz

In table 5.1, simulation parameters are showing which is taken during the execution of Xilinx script. The total utilization of the VLSI architecture or area is 245 components or 2.66%. The total delay or latency value is 34.21 ns. The frequency is 293 MHz and overall throughput is 9.3 GHz.

Table 5.2: Result Comparison

Sr No.	Parameters	Previous Result [1]	Proposed Result
1	Order	16 bit square root	32 bit square root
2	Area	536	245 or 2.66 %
3	Delay	3.69 ns	3.01 ns
4	Power	0.87 mw	0.45 mw
5	Frequency	107.50 MHz	293MHz

Therefore proposed square root VLSI code gives the better performance in terms of the calculated parameters. The proposed code is extension of the existing work. The proposed square root is implemented for the 32 bit square root while previous it is designed for the 16 bit. The total area is optimized 245 number of component or 2.66% while previous its is 536. The delay is 3.01ns while previous it is 3.69ns. The frequency optimized is 293 MHz by proposed and 107.50 MHz by the previous.

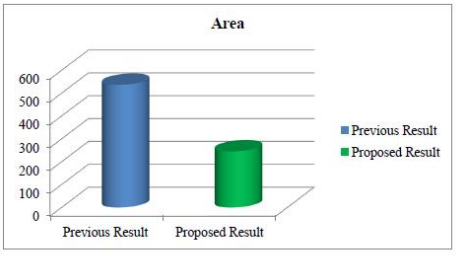


Figure 5.28: Area comparison

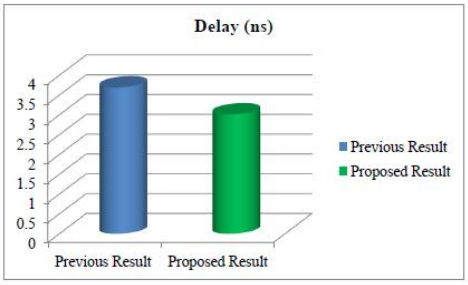


Figure 5.29: Delay comparison

Figure 5.28 and 5.29 is showing the area and delay comparison of the proposed and previous research work. It is clear from the graphical bar chart, proposed square root extension code gives reduced are and latency or delay.

V.CONCLUSION AND FUTURE WORK

Variable precision fixed and floating point operations have various fields of applications including scientific computing and signal processing.

Field Programmable Gate Arrays(FPGAs) are a good platform to accelerate such applications because of their flexibility, low development time and cost compared to Application Specific Integrated Circuits (ASICs) and low power consumption compared to Graphics Processing Units (GPUs). Among those operations, the square root can differ based on the algorithm implemented. They can highly affect the total performance of the application running them.This research proposes a new non-restoring square root algorithm that requires neither square roots nor multiplexors. Compared with previous square root algorithms, our algorithm is very efficient for VLSI implementation. It generates the correct resulting value at each iteration and does not require extra circuitry for adjusting the result bit. The operation at each iteration is simple: addition or subtraction based on the result bit generated in previous iteration. The remainder of the addition or subtraction is fed via registers to the next iteration directly even it is negative. At the last iteration, if the remainder is non-negative, it is a precise remainder. Otherwise, we can obtain a precise remainder by an addition operation

FUTURE SCOPE

Performance analysis through other new approaches.
More parameters can be calculated when use different approaches.
Experiential test in real time environment.
Implemented multi error correction and detection after square root can be used in real-time IOT based wireless sensor network applications..

REFERENCE

1. N. Arya, T. Soni, M. Pattanaik and G. K. Sharma, "Area and Energy Efficient Approximate Square Rooters for Error Resilient Applications," 2020 33rd International Conference on VLSI Design and 2020 19th International Conference on Embedded Systems (VLSID), 2020, pp. 90-95, doi: 10.1109/VLSID49098.2020.00033.

2. R. Nayar, P. Balasubramanian and D. L. Maskell, "Hardware Optimized Approximate Adder with Normal Error Distribution," 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2020, pp. 84-89, doi: 10.1109/ISVLSI49217.2020.00025.
3. Y. Fu, L. Li, Y. Liao, X. Wang, Y. Shi and D. Wang, "A 32-GHz Nested-PLL-Based FMCW Modulator With 2.16- GHz Bandwidth in a 65-nm CMOS Process," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 7, pp. 1600-1609, July 2020, doi: 10.1109/TVLSI.2020.2992123.
4. T. Fujibayashi and Y. Takeda, "A 76- to 81-GHz, 0.6° rms Phase Error Multi-channel Transmitter with a Novel Phase Detector and Compensation Technique," 2019 Symposium on VLSI Circuits, 2019, pp. C16-C17, doi: 10.23919/VLSIC.2019.8778158.
5. S. U. Rehman, M. M. Khafaji, C. Carta and F. Ellinger, "A 10-Gb/s 20-ps Delay-Range Digitally Controlled Differential Delay Element in 45-nm SOI CMOS," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 5, pp. 1233-1237, May 2019, doi:10.1109/TVLSI.2019.2894736.
6. S. Yang, J. Yin, P. Mak and R. P. Martins, "A 0.0056-mm2−249-dB-FoM All-Digital MDLL Using a Block-Sharing Offset-Free Frequency-Tracking Loop and Dual Multiplexed- Ring VCOs," in IEEE Journal of Solid-State Circuits, vol. 54, no. 1, pp. 88-98, Jan. 2019, doi: 10.1109/JSSC.2018.2870551.
7. J. -H. Hsieh, K. -C. Hung, Y. -L. Lin and M. -J. Shih, "A Speed- and Power-Efficient SPIHT Design for Wearable Quality-On-Demand ECG Applications," in IEEE Journal of Biomedical and Health Informatics, vol. 22, no. 5, pp. 1456- 1465, Sept. 2018, doi: 10.1109/JBHI.2017.2773097.
8. H. Fuketa, S. -i. O'uchi and T. Matsukawa, "A Closed- Form Expression for Minimum Operating Voltage of CMOS D Flip-Flop," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 7, pp. 2007-2016, July 2017, doi: 10.1109/TVLSI.2017.2677978.
9. J. J. Pimentel, B. Bohnenstiehl and B. M. Baas, "Hybrid Hardware/Software Floating-Point Implementations for Optimized Area and Throughput Tradeoffs," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 1, pp. 100-113, Jan. 2017, doi 10.1109/TVLSI.2016.2580142.
10. .T. Lee and P. A. Abshire, "Frequency-Boost Jitter Reduction for Voltage-Controlled Ring Oscillators," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 10, pp. 3156-3168, Oct. 2016, doi: 10.1109/TVLSI.2016.2541718.
11. Z. Yan, G. He, Y. Ren, W. He, J. Jiang and Z. Mao, "Design and Implementation of Flexible Dual-Mode Soft- Output MIMO Detector With Channel Preprocessing," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 11, pp. 2706-2717, Nov. 2015, doi: 10.1109/TCSI.2015.2479055.
12. K. Chen and Y. H. Kim, "Current source model of combinational logic gates for accurate gate-level circuit analysis and timing analysis," VLSI Design, Automation and Test(VLSI-DAT), 2015, pp. 1-4, doi: 10.1109/VLSI- DAT.2015.7114529.
13. J. Lin, C. Yang and H. Wu, "A 2.5-Gb/s DLL-Based Burst-Mode Clock and Data Recovery Circuit With $4\times$ Oversampling," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 4, pp. 791-795, April 2015, doi: 10.1109/TVLSI.2014.2316553.
14. H. Amir-Aslanzadeh, E. J. Pankratz, C. Mishra and E. Sanchez-Sinencio, "Current-Reused 2.4-GHz Direct-Modulation Transmitter With On-Chip Automatic Tuning," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 4, pp. 732-746, April 2013, doi: 10.1109/TVLSI.2012.2190538.
15. K. Gupta, M. Bhardwaj, B. P. Singh and R. Choudhary, "Design of Low Power Low Cost True RMS-to-DC Converter," 2012 Second International Conference on Advanced Computing & Communication Technologies, 2012, pp. 364-367, doi: 10.1109/ACCT.2012.44.

