DESIGN AND IMPLEMENTATION OF REVERSIBLE LOGIC-BASED BINARY SQUARE ROOTER

B. Manish Choudary Post Graduate scholor**,**
Dr.Pradeep Kumari Associate Professor,
Department of Electronics and Communication Engineering, Malla Reddy EngineeringCollege,
Kompally, Secunderabad -500100, Telangana, India

Abstract

Calculating square root is an important mathematical operation which has wide applications. The design of square rooter in hardware needs to achieve low power, low area and high speed. Often there can be a trade-off among the three metrics. As the current technology aims for low power, designs require major architectural modification. This paper presents a low power binary square rooter using reversible logic. It uses reversible logic to achieve low power. The binary square rooter is designed and implemented using RCSM (Reversible Controlled Subtract Multiplexer).For further development such as number of quantum cost, garbage outputs and the constant inputs , binary square rooter is implemented using SRG (Samiur Rahman Gate).Binary square rooter using non-restoring algorithm is designed using both SRG and conventional approach. Simulations are carried out using Xilinx ISE software.

Keywords— Square rooter, low power, reversible logic, conventional logic, non-restoring algorithm

I. INTRODUCTION

Square root is a vital mathematical operation which has a lot of applications. Square rooters are used in computer graphics, global positioning system (GPS), and digital signal processing (DSP) computations, mathematical calculations and data processing. In the present world of developments in various fields, the main novelty in the field of VLSI is Low Power. In order to attain low power Reversible logic is used in this design. In Irreversible logics, the amount of output port and input ports are not uniform and therefore the will be a loss of information bits. According to Launders Theorem [1], kTln2 Joules of energy is dissipated for a single bit information. In order to avoid the energy and information loss, reversible computation has to be reversible [2].In reversible logic, the amount of output ports and input ports are uniform and so the loss of heat can be prevented. It reduces the consumption of more power. Authors in [3] developed a low power reversible nonlinear feedback shift register. Square root methods such as Newton Raphson method, Goldschmidt methods [4] are complicated as it involves more steps and hardware resources when compared to the digit-by-digit method. The square root calculation using digit-by-digit method can be performed either by restoring method or non-restoring method. In restoring method more hardware resources are required whereas in non-restoring method it consumes less hardware compared to restoring method. Hence non-restoring algorithm [5] is preferred. Many hardware architectures for digit-by-digit square root calculation using irreversible logic has been proposed. There are many reversible logic circuits that are used for the computation. Power reduction can also be done using array based arithmetic computation [6]. Reversible logic is used to design hardware realization of non-restoring algorithm for computation of square root in [7]. Power obtained was high.

III. METHODOLOGY

To find the square root of a binary number is performed by digit by digit procedure. The algorithm is further classified into non-restoring and restoring algorithms. The restoring algorithm involves more number of hardware resources which in turn increases the total power of the system. Hence the non-restoring algorithm is used for the computation. The non-restoring algorithm simply uses subtraction and appending 01. In the non-restoring algorithm, the total number of bits N is divided into groups of two digits. In general, the length of the quotient is N/2. The algorithm is as follows:

- **Step 1:** Split the number of bits N into groups of two digits.
- **Step 2:** Subtract 1 from the leftmost significant group of digits. Quotient is 1 if the difference is positive and quotient is 0 of the difference is negative
- **Step 3:** Bring the next group of two digits. Append 01 and the previous quotient and then subtract.
- **Step 4:** Proceed to step 2 until the end of groups of two digits.

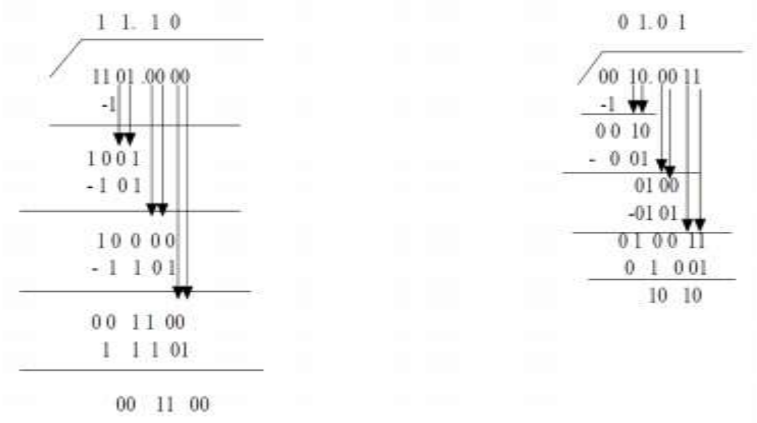


Fig. 1 Non-restoring algorithm example

The radicand may be a whole number (13) or may be a decimal number (2.2).If the radicand is a whole number (13) it can be represented as 1101 and if it is a decimal number 2.2 it can be represented as0010.0011… Here 0010 corresponds to 6 and 0011...corresponds to 0.4 in binary. The number of bits after the decimal can be extended to N number of bits depending upon the application and accuracy that is required. For example, consider a binary square rooter (N7N6N5N4.N3N2N1N0). The quotient for the square rooter is U3U2.U1U0.The count of bits before and after the decimal point can be decided upon the user. For example the square root of 13(1101) and 2.2(0010.0011) is given in figure 1. From Fig.1, the square root of 13 and 2.2 are found. The square root of 13 is approximately equal to 3.6.From the quotient 11.10(U3U2.U1U0) ,11(U3U2) corresponds to 3 and 10 (U1U0) corresponds to .6.The binary representation of .6 is 1001..The square root of 2.2(0010.0011) is approximately equal to 1.4..From the quotient 01.01, 01(U3U2) corresponds to 1 and 01 (U1U0) corresponds to .4.The binary representation of .4 is 0110..As earlier mentioned the number of bits before and after the decimal point can be decided upon the user and application. If more number of bits in the quotient is necessary the bit size after the decimal point shall be increased.

IV. PROPOSED METHOD

The non-restoring algorithm used for the computation requires a reversible full subtractor. The square rooter is designed using Saimur Rahman gate and Feynman Gate.The functionality of the Saimur Rahman gate is a full subtractor. It is shown in Fig.2.

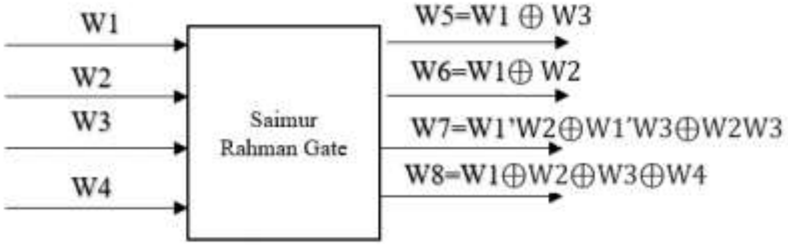


Fig. 2 Saimur Rahman Gate

Here the output W7 represents the borrow (BO) and W8 represents the difference (DI).

Table.1 Truth Table for Samiur Rahman Gate

W1	W2	W3	W4	W7	W8
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	0	0	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

The SRG gate acts as a full subtractor only if W4=0, if W4 is ≠ 0 then the gate won't act as a full subtractor. W5 and W6 acts as a garbage output. The truth table of the SR gate is given in Table.1 Let the radicand be N7N6N5N4.N3N2N1N0 with a total length of 8 bits. Since it is an 8-bit square rooter the no of bits for the quotient is 8/2= 4(U3U2.U1U0). The implementation of 8-bit square rooter using SR and FG gates is shown in Fig.3.

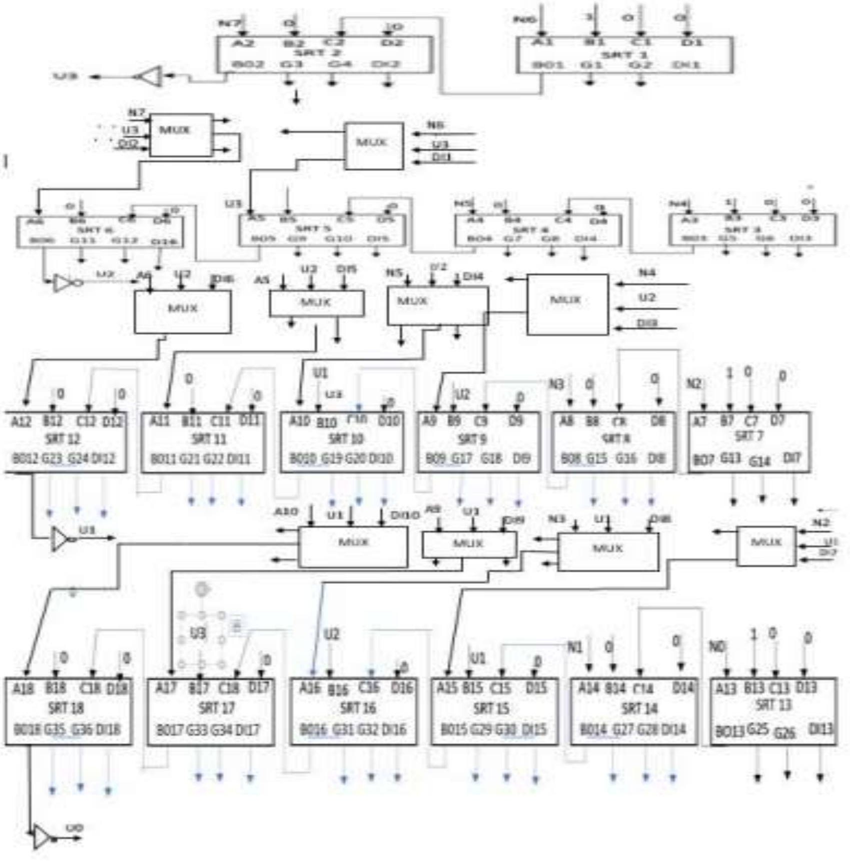


Fig. 3 Design of an 8 Bit Binary square rooter

According to the non-restoring computation if the remainder is positive then quotient (U=1) and the difference should be carried over for the next process. If the remainder is negative then quotient is (U=0) and the previous inputs should be carried over for the next process. So a control unit is designed using RT reversible gate [13] to switch between the difference (W8) and the inputs (W1).A reversible multiplex is used for all the gates to switch between the difference and the inputs. The input (W1), the difference (W8) and the quotient is given as input to the gates. The configuration is shown in Fig.3.So depending upon the quotient (U) the input and the difference will get switched automatically.

Thus with these reversible gates the design is proposed for the non-restoring algorithm. The conventional design is carried upon using the irreversible gates. The basic xor, and, or, not gates are used for conventional approach. The power results of reversible and conventional design is obtained.

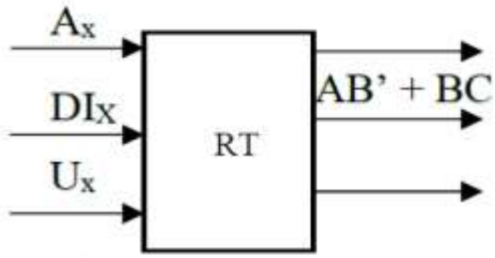
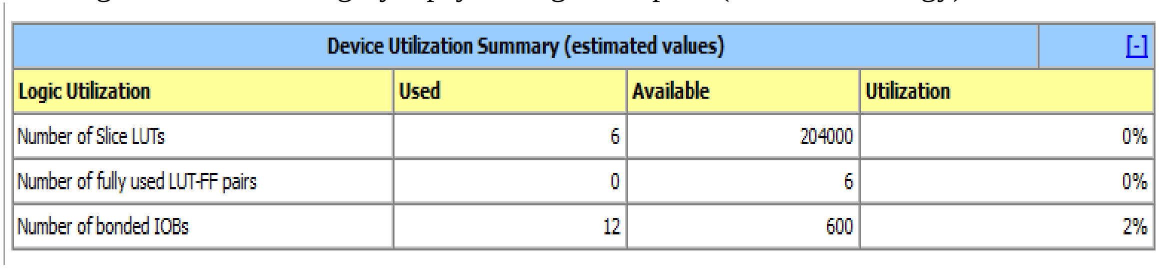


Fig. 4: Reversible mux

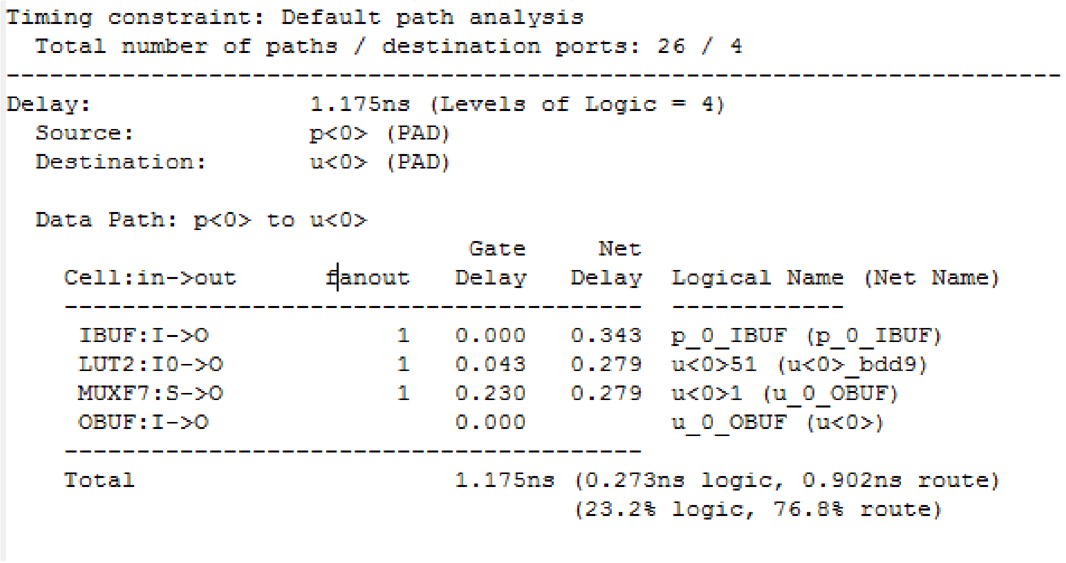
V. SIMULATION RESULTS

The proposed design has been coded in verilog language in Xilinx ISE. The corresponding outputs for the input radicand are verified. The proposed designed has been compiled and power result for the reversible design is obtained using Synopsys Design Compiler (25nm technology).



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	6	204000	0%
Number of fully used LUT-FF pairs	0	6	0%
Number of bonded IOBs	12	600	2%

Figure 5: Design summary



```
Timing constraint: Default path analysis
  Total number of paths / destination ports: 26 / 4
-------------------------------------------------------------------------
Delay:                  1.175ns (Levels of Logic = 4)
  Source:               p<0> (PAD)
  Destination:          u<0> (PAD)

  Data Path: p<0> to u<0>
                                     Gate     Net
    Cell:in->out       fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O              1   0.000   0.343  p_0_IBUF (p_0_IBUF)
     LUT2:I0->O             1   0.043   0.279  u<0>51 (u<0>_bdd9)
     MUXF7:S->O             1   0.230   0.279  u<0>1 (u_0_OBUF)
     OBUF:I->O                  0.000          u_0_OBUF (u<0>)
    ----------------------------------------
    Total                       1.175ns (0.273ns logic, 0.902ns route)
                                        (23.2% logic, 76.8% route)
```

Figure 6: Time summary

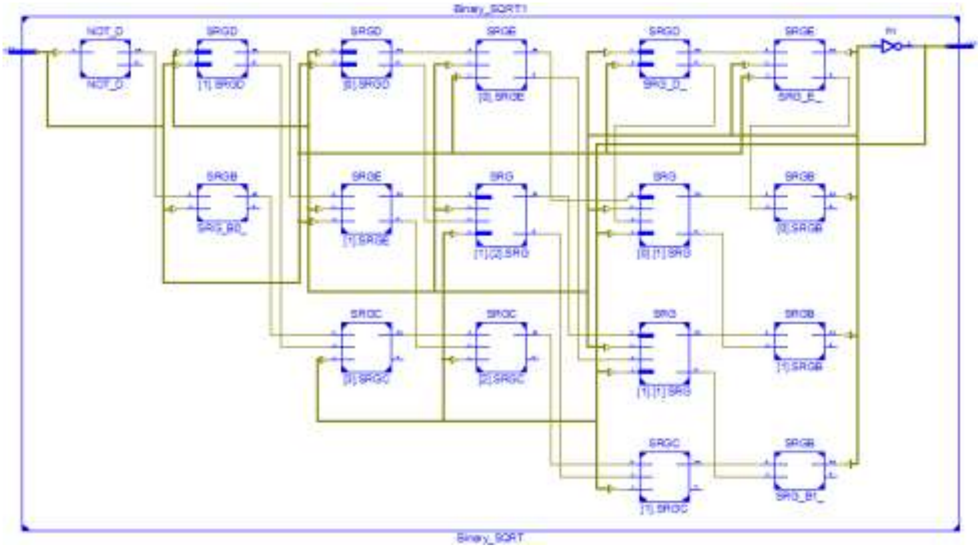


Figure 7: RTL Schematic

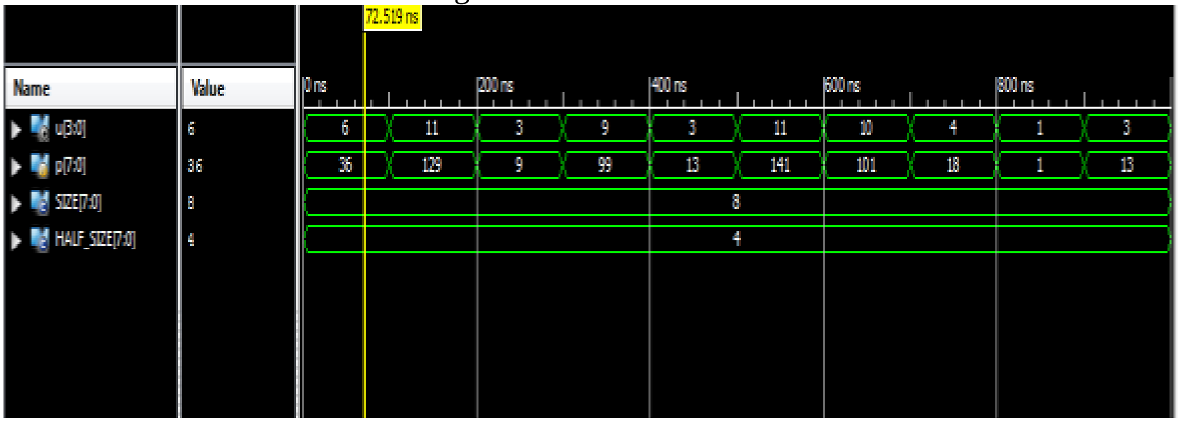


Figure 8: Simulation output

VI. CONCLUSION

In this paper, it can be seen that the performance of square rooter circuits can be enhanced using reversible gates and in terms of speed and power; thereby concluding that reversible designs are faster and power efficient. A low power reversible binary square rooter has been designed. The square rooter can be used in ALUs especially in DSP processors for reversible computing. Conventional approach consumes more power than the reversible computing. Also a decrease in gate count is observed.

REFERENCES

[1]. R. Landauer, Irreversibility and heat generation in the computing process, IBM J. Res. Dev. 5 (1967) 183–191.

[2]. C.H. Bennett, Logical reversibility of computation, IBM J. Res. Dev. 17 (1973), 525–532. [3] N. Krishna, V. Murugappan, R. Harish, M. Midhun and E. Prabhu,

[3]. "Design of a novel reversible nlfsr," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017, pp. 2279-2283.

[4]. L. Yamin, C. Wanming, Implementation of single precision floating point square root on FPGAs, in: Proceedings of the IEEE Symposium on FPGA for Custom Computing Machines,1997,pp 226–232.

[5]. T. Sutikno, An efficient implementation of the non-restoring square root algorithm in gate level, Int. J. Comput. Theory Eng.3 (1) (Feb. 2011) 1793–8201.

[6]. H. Haritha and S. R. Ramesh, "Design of an Enhanced Array Based Approximate Arithmetic Computing Model for Multipliers and Squarers," 2017 14th IEEE India Council International Conference (INDICON), Roorkee, 2017, pp. 1-5.

[7]. L. Yamin, C. Wanming, A new non-restoring square root algorithm and its VLSI Implementation ,in :Proceedings of the IEEE International Conference, 1996, pp. 538–544.

[8]. L. Yamin, C. Wanming, Parallel-array implementations of a non-restoring square root algorithm, in: Proceedings of the IEEE International Conference on ComputerDesign: VLSI in Computers and Processors, 1997, pp. 690–695.

[9]. R. Balakumaran and E. Prabhu, "Design of high speed multiplier using modified booth algorithm with hybrid carry look-ahead adder," 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, 2016, pp. 1-7.

[10]. T. Sutikno, An efficient implementation of the non-restoring square root algorithm in gate level, Int. J. Comput. Theory Eng. 3 (1) (Feb. 2011) 1793–8201.

[11]. A.V. AnanthaLakshmi ,Gnanou Florence Sudha, “ Design of a reversible floating-point square root using modified non-restoring algorithm”, in Microprocessors and Microsystem, Vol.50, pp.39-53, May 2017.

[12]. Md. SamiurRahman,SajjadWaheed, Ali NewazBahar, “Optimized Design of Full-Subtractor Using New SRG Reversible Logic Gates and VHDL Simulation”, in International Conference on Electrical & Electronic Engineering,Rajshahi,2015,pp.69-72

[13]. L. Gopal, N Raj, A. A. Gopalai and A. K. Singh, "Design of reversible multiplexer/de-multiplexer," 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014), Batu Ferringhi, 2014, pp. 416-420.