ISSN No :- 2456 - 2165

High Speed 4bit/8bit QSD Adder With Reversible Logic Gate

Richa Gupta
M.Tech Scholar
EC-VLSI
Jagannath University ,Jaipur(Raj),India
richagupta191@gmail.com

Nidhish tiwari
Assistant professor
Jagannath National Institute of Technology(JNIT),Jaipur(Raj),India
Nidhishtiwari@gmail.Com

Abstract :- The need for high speed digital circuits became more prominent as portable multimedia and communication applications incorporating information processing and computing. The drawback of modern computers lead to the deterioration in performance of arithmetic operations such as addition, subtraction, division, multiplication on the aspects of carry propagation time delay, high power consumption and large circuit complexity. This system explores the carry free n digits addition/subtraction as the carry propagation delay is most important factor regarding the speed of any digital system. In this paper, Quaternary signed digit (QSD) numbers whose radix is 4 are used in arithmetic operations to achieve the carry free arithmetic operations. The range of QSD number is from -3 to 3.In any n digit QSD number ,each digit can be represented by a number from the digit set [-3,-2,-1,0,1,2,3]. In this paper , we are improving the performance of the QSD addition by apply reversible logic gate . QSD addition is performing for the 4 bit,8 bit . In the normal condition QSD adder gives the high delay performance . We have to reduce the delay of the 4 bit,8 bit QSD adder . For reduce the delay we apply pipelining and Peres reversible logic gate . Pipelining and Peres reversible gate will be able to reduce the delay of the 4 bit , 8 bit QSD adder .

Keywords- *Carry free addition, Fast computing, FPGA, Quaternary Signed Digit, VHDL, VLSI.*

I. INTRODUCTION

Quaternary is the base-4 numeral system. It uses the digits 0, 1, 2 and 3 to represent any number. Four is the largest number within the subsidizing range and one of two numbers that is both a square and a highly composite number (the other being 36), making quaternary a convenient choice for a base at this scale. Despite being twice as large, its radix economy is equal to that of binary. However, it fares no better in the localization of prime numbers (the next best being the primordial base six, senary).

Quaternary shares with all fixed-radix numeral systems many properties, such as the ability to represent any real number with a canonical representation (almost unique) and the characteristics of the representations of rational numbers and irrational numbers. See decimal and binary for a discussion of these properties.

Arithmetic operations play an important role in various digital systems such as computers, process controllers, signal processors computer graphics and image processing. Recent advances in technologies for integrated circuits make large scale arithmetic circuits suitable for VLSI implementation [1]. However, arithmetic operations still suffer from known problems including limited number of bits, propagation time delay, and circuit complexity. Now, the flexibility of field programmable gate arrays (FPGAs) allows the rapid development of high performance custom hardware. By selecting arithmetic algorithms suited to the FPGA technology and subsequently applying optimal mapping strategies, high performance FPGA implementations can be developed [7].

High speed QSD arithmetic logic unit which is capable of carry free addition, borrow free subtraction, up-down count and multiply operations. The QSD addition/subtraction operation employs a fixed number of minters for any operand size. Signed digit number system offers the possibility of carry free addition. QSD Adder / QSD Multiplier circuits are logic circuits designed to perform high-speed arithmetic operations. In QSD number system carry propagation chain are eliminated which reduce the computation time substantially, thus enhancing the speed of the machine[2].

A. Quaternary Signed Digit Numbers

QSD numbers are represented using 3-bit 2's complement notation. Each number can be represented by:

$$D= \sum_i^n x_i 4^i \quad \text{.....(1)}$$

Where x_i can be any value from the set {3, 2, 1, 0, 1, 2, 3} for producing an appropriate decimal representation. A QSD negative number is the QSD complement of QSD positive number i.e.3 = -3, 2 = -2, 1 = -1. For digital implementation, large number of digits such as 64, 128, or more can be implemented with constant delay. A higher radix based signed digit number system, such as quaternary signed digit (QSD) number system, allows higher information storage density, less complexity, fewer system components and fewer cascaded gates and operations. A high speed and area effective adders and multipliers can be implemented using this technique.

For Example

$$(1\bar{2}\bar{3}\bar{3})_{QSD} = (23)_{10}$$
$$= 1*4^3 + \bar{2} * 4^2 + \bar{3} * 4^1 + 3 * 4^0$$
$$= 64-32-12+3 = 23$$
$$= (\bar{1}\ 2\ 3\ \bar{3})_{QSD} = (-23)_{10}$$

B. Adder Design

A fast Arithmetic unit allows the extension of the application domain of fast multiplier in digital signal processing, inversion of matrices, modular exponential, computation of Eigen values, digits filters etc. Computation of reciprocals, square roots, inverse square roots, and some elementary functions using small tables, small multipliers, and for some functions, a final (large) multiplication can be achieved.

Addition is the most important arithmetic operation in digital computation. A carry-free addition is highly desirable as the number of

digits becomes large. We can achieve carry free addition by exploiting the redundancy of QSD numbers and the QSD addition.

$$(6)_{10} = (12)_{QSD} = (2\overline{2}) \quad \text{.....(2)}$$

There are two steps involved in the carry-free addition. The first step generates an intermediate carry and sum from the addend and augends. The second step combines the intermediate sum of the current digit with the carry of the lower significant digit [3][6]. To prevent carry from further rippling, we define two rules. The first rule states that the magnitude of the intermediate sum must be less than or equal to 2. The second rule states that the magnitude of the carry must be less than or equal to 1.Consequently, the magnitude of the second step output cannot be greater than 3 which can be represented by a single digit QSD number; hence no further carry is required. In step 1,all possible input pairs of the addend and augends are considered.

Sum	QSD representation	QSD coded number
-6	$\overline{2}\,2\,,\overline{1}\overline{2}$	$\overline{1}\overline{2}$
-5	$\overline{2}3,\overline{1}\overline{1}$	$\overline{1}\overline{1}$
-4	$\overline{1}0$	$\overline{1}0$
-3	$\overline{1}1,0\overline{3}$	$\overline{1}1$
-2	$\overline{1}2,0\overline{2}$	$0\overline{2}$
-1	$\overline{1}3,0\overline{1}$	$0\overline{1}$
0	00	00
1	$01,1\overline{3}$	01
2	$02,1\overline{2}$	02
3	$03,1\overline{1}$	$1\overline{1}$
4	10	10
5	$11,2\overline{3}$	11
6	$12,2\overline{3}$	12

Table 1 :- The Intermediate Carry And Sum Between -6 To 6

Both inputs and outputs can be encoded in 3-bit 2'scomplement binary number. The mapping between the inputs, addend and augends, and the outputs, the intermediate carry and sum are shown in binary format in Table IV. Since the intermediate carry is always between numbers -1 and 1, it requires only a 2-bit binary representation. Finally, five 6-variable Boolean expressions can be extracted.

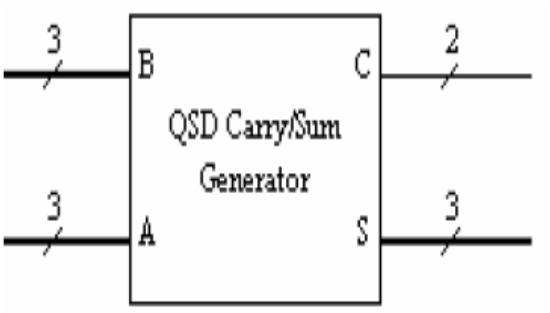

Figure.1. The intermediate carry and sum generator

The intermediate carry and sum generator In step 2, the intermediate carry from the lower significant digit is added to the sum of the current digit to produce the final result. The addition in this step produces no carry because the current digit can always absorb the carry-in from the lower digit. All possible combinations of the summation between the intermediate carry and the sum.

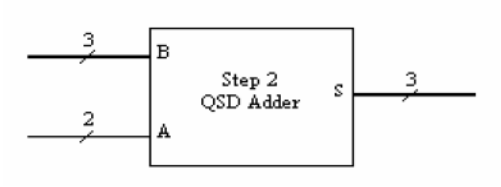

Figure 2. The second step QSD added.

The implementation of an n-digit QSD adder requires n QSD carry and sum generators and n-1 second step adders as shown in The result turns out to be an n+1-digit number. The result of addition in this step ranges from -3 to 3. Since carry is not allowed in this step, the result becomes a single digit QSD output. The inputs, the intermediate carry and sum, are 2-bit and 3-bit binary respectively. The output is a 3-bit binary represented QSD number [4] [5].

Boolean equations extracted are implemented using FPGA tools. Modelsim and Leonardo Spectrum. Design is compiled and simulated using Modelsim. Overall adders design requires two steps for addition. Figure 2 shows the diagram of the second step adder. The implementation of an n-digit QSD adder requires n QSD carry and sum generators and n-1 second step adders as shown in Figure 3. The result turns out to be an n+1-digit number.

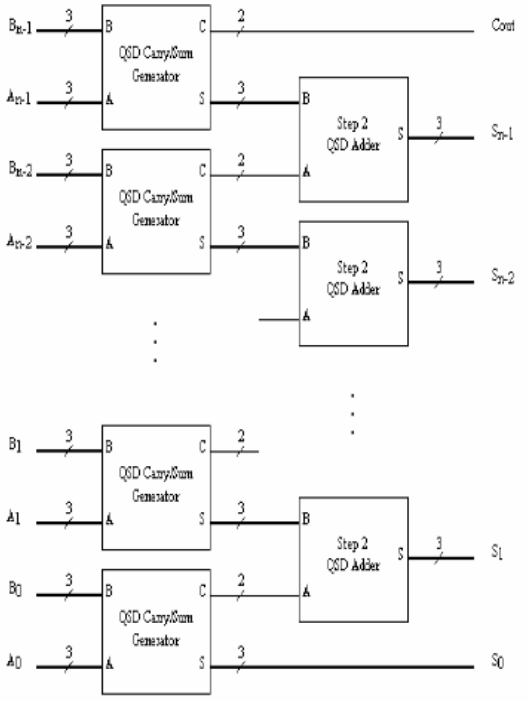

Figure.3 . n-digit QSD adder

II. PROBLEM STATEMENT

The limitation of speed of modern computers in performing the arithmetic operations such as addition, subtraction and multiplication suffer from carry propagation delay. Carry free arithmetic operations can be achieved using a higher radix number system such as Quaternary Signed Digit (QSD). In the previous paper fast adders based on Quaternary signed digit number system was presented . In QSD, each digit can be represented by a number from -3 to 3. Carry

ISSN No :- 2456 - 2165

free addition and other operations on a large number of digits such as 64, 128, or more can be implemented with constant delay and less complexity. The design of these circuits is carried out using FPGA tools. The designs are simulated using modelsim software and synthesized using Leonardo Spectrum. Delay is getting 6ns for the previous paper . The delay and area for the previous paper is high . So in this paper main problem which we find that is delay . We have reduce the delay from the previous paper work .

III. PROPOSED METHODLOGY

QSD is a carry free addition which is performing for increase the speed of the desire system . In the normal QSD for addition of carry and sum till to present time normal full adder is using for addition .

A:- Methodology 1

We will apply Peres gate based full adder by which we can further improve the performance of the delay .We design the full adder Peres reversible logic gate based .

The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P = A, Q = AB and R=AB C. Quantum cost of a Peres gate is 4. In the proposed design Peres gate is used because of its lowest quantum cost.

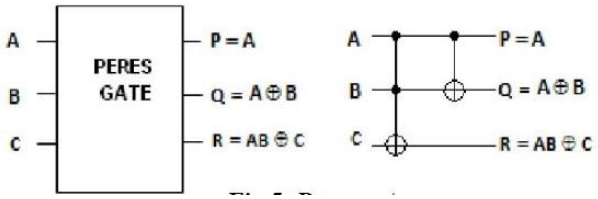

Figure 4 :- Peres Gate

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	1	1	0	0

Table 2 :- Peres gate Truth Table

A full- adder using two Peres gates is as shown in fig . The quantum realization of this shows that its quantum cost is 8 two Peres gates are used A single 4*4 reversible gate called PFAG gate with quantum cost of 8 is used to realize the multiplier.

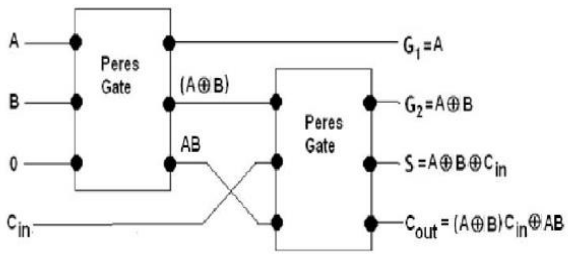

Figure 5 :- Full adder using two Peres gates

B. Methodlogy2

In this we apply the pipelining for further reduce the delay of the circuit . Firstly we apply pipelining in the in the QSD carry sum generator and then we apply the pipeline in the bit transfer values of carry and sum . In the pipeline we can transfer more than one clock by a single clock .

IV. RESULTS

A. 4 Bit QSD adder

Existing design is a normal design in which we use normal addition for add the bits . For normal addition we use simple addition sign . As shown in the figure the RTL schematic for the existing system .

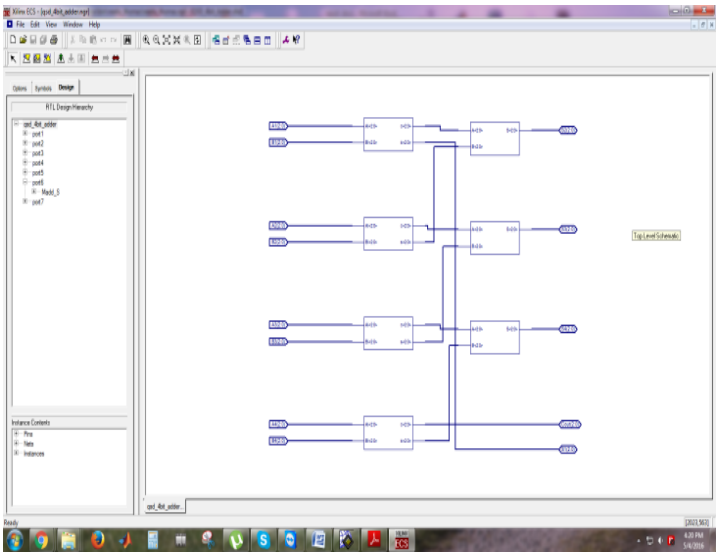
Figure 6:- RTL schematic for existing design

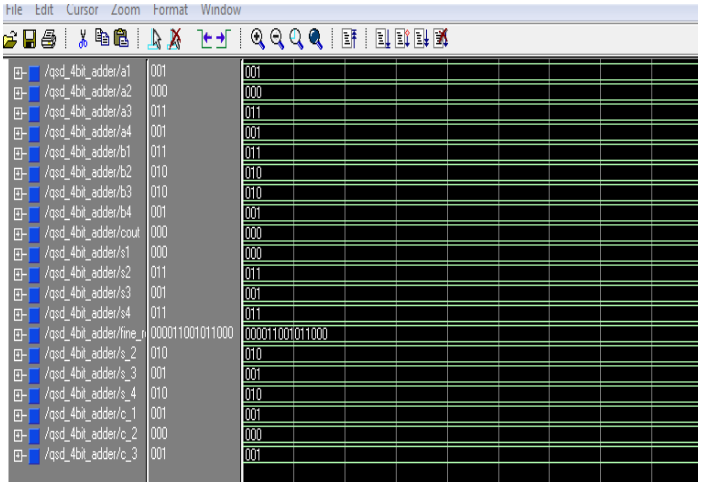

Figure 6 :- waveform generation for Existing design

In the existing design we give the input a1,a2,a3,a4 as a 1,0,3,1 and b1,b2,b3,b4 as a 3,2,2,1. As we can see from the waveform session the output comes in two parts sum and carry . For this addition sum is 3130 and carry is 0 . As we can see from the waveform session final_result is 000011001011000. At the input we give the input of the 4 bit and the output is also 4 bit bits with one carry . In the existing design delay is 15.194 ns. As we can see from the image delay is 15.194 ns for the existing design .

```
Timing constraint: Default path analysis
Delay:              15.194ns (Levels of Logic = 9)
  Source:           B3<0> (PAD)
  Destination:      Fine_Result<11> (PAD)

  Data Path: B3<0> to Fine_Result<11>
                              Gate     Net
    Cell:in->out      fanout  Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             13   1.492   0.895  B3_0_IBUF (B3_0_IBUF)
     LUT3:I1->O             1   0.720   0.240  port3_Ker261663 (CHOICE667)
     LUT3:I0->O             1   0.720   0.240  port3_Ker261666 (CHOICE668)
     LUT4:I0->O             2   0.720   0.465  port3_Ker261687 (CHOICE670)
     LUT4:I3->O             2   0.720   0.465  port3_Ker2616120 (N6846)
     LUT2:I0->O             2   0.720   0.465  port3__n0048<3>60 (C_3<0>)
     LUT4:I0->O             1   0.720   0.240  port7_Madd_S_Mxor_Result<2>_Xo<1>_SW1 (N8883)
     LUT4:I3->O             1   0.720   0.240  port7_Madd_S_Mxor_Result<2>_Xo<1> (Fine_Result_11_OBUF)
     OBUF:I->O                  5.412          Fine_Result_11_OBUF (Fine_Result<11>)
    ----------------------------------------
    Total                      15.194ns (11.944ns logic, 3.250ns route)
                                        (78.6% logic, 21.4% route)
```

Figure 7:- Delay report for the Existing design

B. 8 Bit QSD Adder

Existing design is a normal design in which we use normal addition for add the bits . For normal addition we use simple addition sign . As shown in the figure the RTL schematic for the existing system .

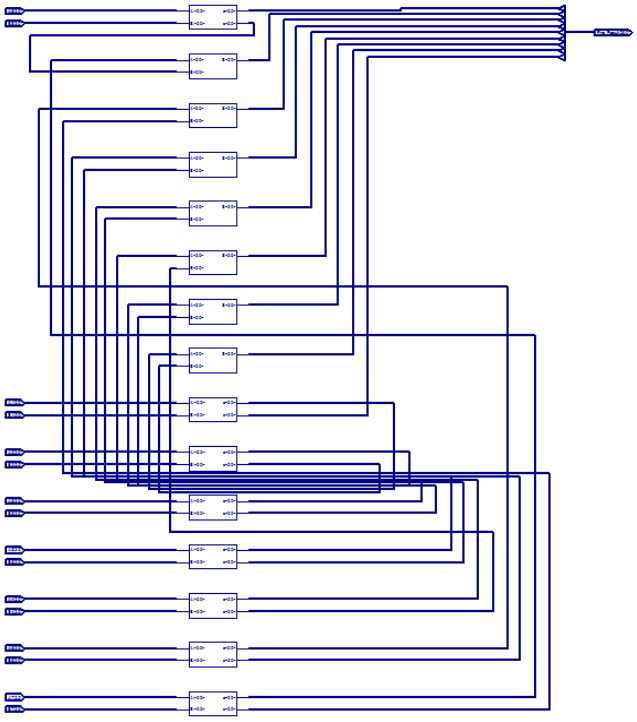

Figure 8:- RTL schematic for existing design

In the existing design we give the input a1,a2,a3,a4,a5,a6,a7,a8 and b1,b2,b3,b4,b5,b6,b7,b8. As we can see from the waveform session the output comes in two parts sum and carry .

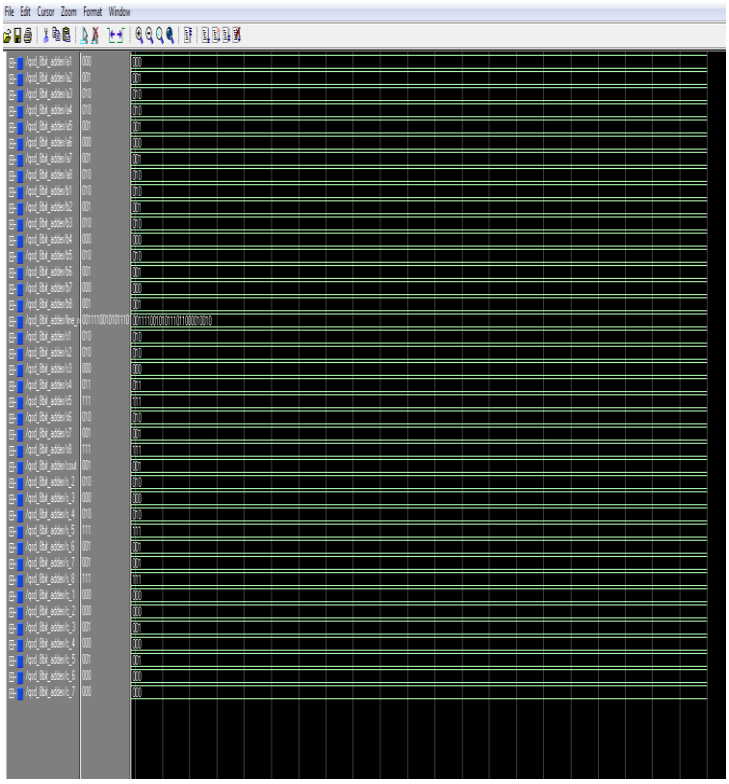

Figure 9 :- waveform generation for Existing design

As we can see from the waveform session final_result is 001111100101011011000010010. At the input we give the input of the 8 bit and the output is also 8 bit bits with one carry . In the existing design delay is 15.244 ns. As we can see from the image delay is 15.449 ns for the existing design .

```
Data Path: B5<0> to Fine_Result<17>
                              Gate     Net
    Cell:in->out      fanout  Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             14   1.492   0.925  B5_0_IBUF (B5_0_IBUF)
     LUT3:I1->O             1   0.720   0.240  port4_Ker482463 (CHOICE1173)
     LUT3:I0->O             1   0.720   0.240  port4_Ker482466 (CHOICE1174)
     LUT4:I0->O             2   0.720   0.465  port4_Ker482487 (CHOICE1176)
     LUT4:I3->O             2   0.720   0.465  port4_Ker4824120 (N12124)
     LUT2:I0->O             2   0.720   0.465  port4__n0048<3>60 (c_5<0>)
     LUT4:I0->O             1   0.720   0.240  port9_Madd_S_Mxor_Result<2>_Xo<1>_SW1 (N16197)
     LUT4:I3->O             1   0.720   0.240  port9_Madd_S_Mxor_Result<2>_Xo<1> (Fine_Result_17_OBUF)
     OBUF:I->O                  5.412          Fine_Result_17_OBUF (Fine_Result<17>)
    ----------------------------------------
    Total                      15.224ns (11.944ns logic, 3.280ns route)
                                        (78.5% logic, 21.5% route)
```

Figure 10:- Delay report for the Existing design

C. 4 bit QSD Adder with Reversible Logic Gate

In the proposed design we apply Peres based reversible gate for reduce the delay . As shown in the figure the RTL schematic for the existing system .

In the proposed design we give the input a1,a2,a3,a4 as a 1,0,3,1 and b1,b2,b3,b4 as a 3,2,2,1. As we can see from the waveform session the

output comes in two parts sum and carry . For this addition sum is 3130 and carry is 0 . As we can see from the waveform session final_result is 000011001011000. At the input we give the input of the 4 bit and the output is also 4 bit bits with one carry . In the proposed design delay is 1.361 ns. As we can see from the image delay is 1.361 ns for the proposed design .

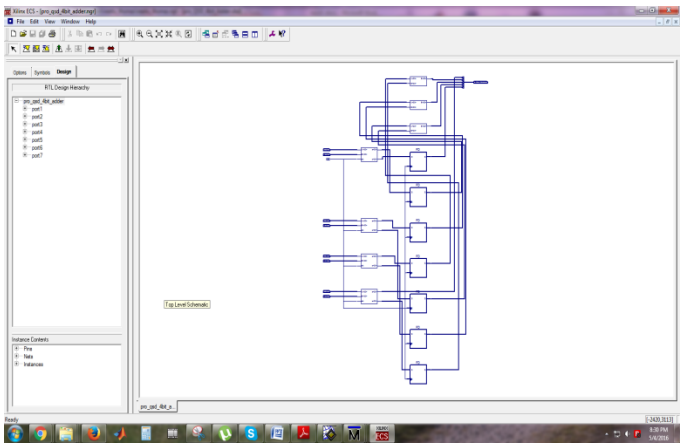

Figure 11 :- RTL schematic for Proposed design

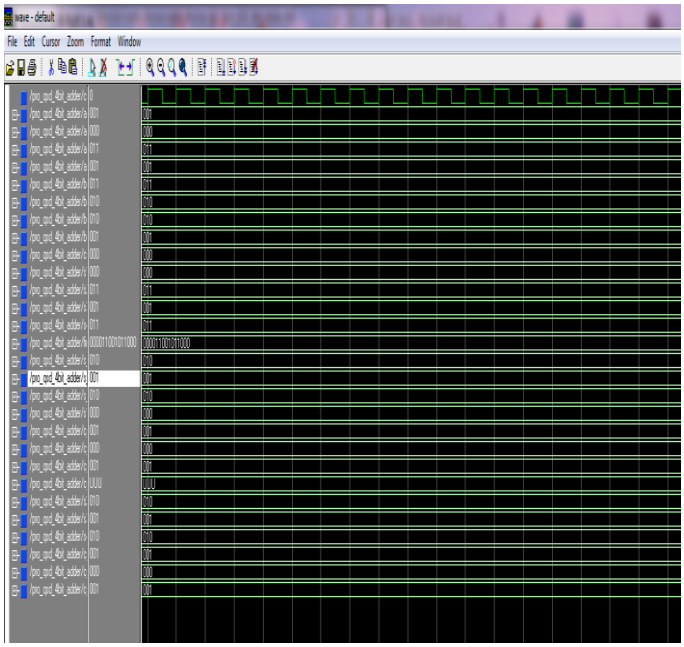

Figure 12:- waveform generation for Proposed design

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
FDS:C->Q	1	0.619	0.240	port4_s_2 (port4_s_2)
FD:D		0.502		S4_reg_2
Total		1.361ns	(1.121ns logic, 0.240ns route) (82.4% logic, 17.6% route)	

Figure 13 :- Delay report for the Proposed design

D. 8 bit QSD Adder with Reversible Logic Gate

In the proposed design we apply Peres based reversible gate for reduce the delay . As shown in the figure the RTL schematic for the existing system . In the proposed design we give the input a1,a2,a3,a4,a5,a6,a7,a8 and b1,b2,b3,b4,b5,b6,b7,b8. As we can see from the waveform session the output comes in two parts sum and carry . As we can see from the waveform session final_result is 001111100101011011000010010. At the input we give the input of the 8 bit and the output is also 8 bit bits with one carry . In the proposed design delay is 1.361 ns. As we can see from the image delay is 1.361 ns for the proposed design .

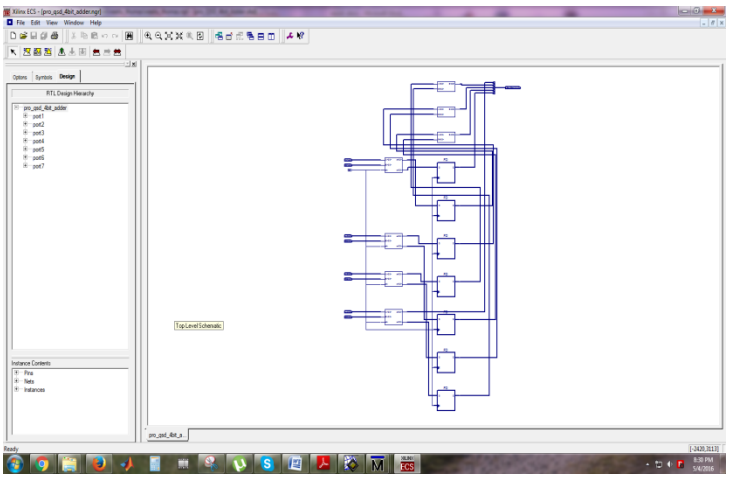

Figure 14:- RTL schematic for Proposed design

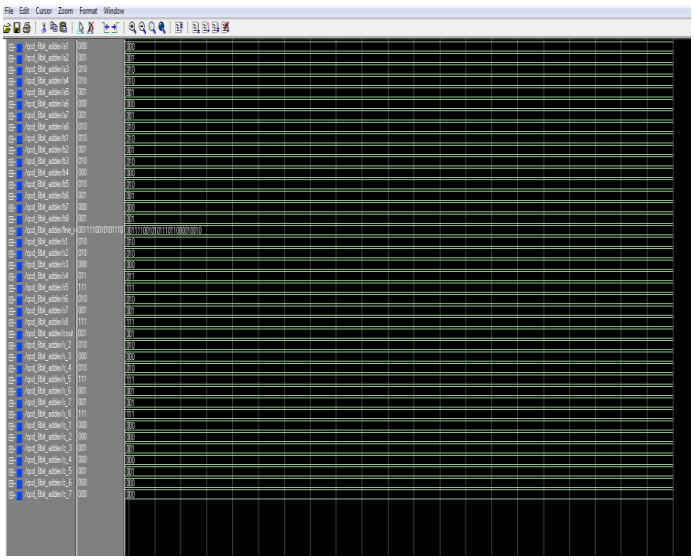

Figure 15:- waveform generation for Proposed design

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
FDS:C->Q	1	0.619	0.240	port4_s_2 (port4_s_2)
FD:D		0.502		S4_reg_2
Total		1.361ns	(1.121ns logic, 0.240ns route) (82.4% logic, 17.6% route)	

Figure 16:- Delay report for the Proposed design

ISSN No :- 2456 - 2165

	Delay(ns)
Existing Design(4 bit)	15.449 ns
Existing Design(8 bit)	15.224 ns
Proposed Design (4 Bit)	1.361 ns
Proposed Design (8 Bit)	1.361 ns

Table 3 :- Delay comparison

V. CONCLUSION & FUTURE SCOPE

A. Conclusion

The proposed QSD adder with Reversible logic gate is better than other binary adders in terms of low delay addition. Efficient design for adder block to perform addition or multiplication will increase operation speed. We proposed fast adders based on Quaternary signed digit number system with reversible gate . In QSD, each digit can be represented by a number from -3 to 3. Carry free addition and other operations on a large number of digits such as 64, 128, or more can be implemented with constant delay and less complexity. The design of these circuits is carried out using FPGA tools. The designs are simulated using modelsim software and synthesized using Xilinx.

Reversible logic has received great attention in the recent years due to their ability to reduce the power dissipation which is the main requirement in low power VLSI design. It has wide applications in low power CMOS and Optical information processing, DNA computing, quantum computation and nanotechnology. Irreversible hardware computation results in energy dissipation due to information loss.

Reversibility in computing implies that no information about the computational states can ever be lost, so we can recover any earlier stage by computing backwards or un-computing the results. This is termed as logical reversibility. The benefits of logical reversibility can be gained only after employing physical reversibility. Physical reversibility is a process that dissipates no energy to heat. Absolutely perfect physical reversibility is practically unachievable. Computing systems give off heat when voltage levels change from positive to negative: bits from zero to one. Most of the energy needed to make that change is given off in the form of heat. Rather than changing voltages to new levels, reversible circuit elements will gradually move charge from one node to the next. This way, one can only expect to lose a minute amount of energy on each transition. Reversible computing strongly affects digital logic designs. Reversible logic elements are needed to recover the state of inputs from the outputs. It will impact instruction sets and high-level programming languages as well. Eventually, these will also have to be reversible to provide optimal efficiency.

Reversible logic gates are using in VLSI circuit for improve the power performance of the circuit . In the QSD the power ,area and delay will be reduce after apply reversible logic gates . We use Peres gates for reduce the area and delay . In the normal QSD delay is getting 15.449ns and after apply proposed methodology , it is getting 1.361 ns .

B. Future Scope

In this project , we can further improve the performance of the adder towards the digital design development using reversible logic circuits which are helpful in quantum computing, low power CMOS, nanotechnology, cryptography, optical computing, DNA computing, digital signal processing (DSP), quantum dot cellular automata, communication, computer graphics.

REFERENCES

[1] Chao Cheng; Parhi and K.K. “High-Throughput VLSI Architecture for FFT Computation”, IEEE Transactions on Volume 54, Issue 10, Page 863 – 867, October 2007.

[2] I.M. Thoidis, D. Soudris, J.M. Fernandez , A. Thanailakis, “The circuit design of multiple-valued logic voltage-mode adders,” 2001 IEEE International Symposium on Circuits and Systems, pp 162-165,Vol. 4 , 2001.

[3] Hwang K. (1979) Computer Arithmetic Principles Architecture and Design. New York : Wiley.

[4] N. Takagi, H. Yasuura, and S. Yajima, “High Speed VLSI Multiplication Algorithm with a Redundant Binary Addition Tree, ” IEEE Trans. Comp., C-34, pp. 789-795, 1985.K. Elissa, “Title of paper if known,” unpublished.

[5] O. Ishizuka, A. Ohta, K. Tannno , Z. Tang, D. Handoko , “VLSI design of a quaternary multiplier with direct generation of partial products,” Proceedings of the 27th International Symposium on Multiple Valued Logic, pp. 169-174, 1997.

[6] J.U.Ahmed, A.A.S. Awwal, “Multiplier design using RBSD number system”, Proceedings of the 1993 National Aerospace and Electronics Conference, pp. 180-184, Vol. 1, 1993.

[7] A. Avizienis, “Signed-Digit Number Representation for Fast Parallel Arithmetic, ”IRE Transaction Electron. Comp., EC-10, pp. 389-400, 1961.

[8] M. E. Louie and M. D. Ercegovac,” On Digit -Recurrence Division Implementations for Field Programmable gate Arrays” In Proc. Of the 11th symposium on Computer Arithmetic, 2012.