DESIGN AND ANALYSIS OF SQUARING CIRCUIT USING VARIOUS ADDERS

Dr, Nagaraju N[1], Shobana S[2], Siva Ranjani G[3], Sivagami S[4]

Assistant Professor[1], UG Scholars[2,3,4,5], Department of Electronics and Communication Engineering, Adhiyamaan College of Engineering, Hosur, Krishnagiri district, Tamilnadu, India.
[1]nnagaece@gmail.com, [2]shobanasambath15@gmail.com,
[3]sivaranjanig2000@gmail.com, [4]shansivagami29@gmail.com

Abstract: In VLSI technology, many numbers of components are incorporated in a single chip to reduce the size of the device. Each arithmetic operation has certain algorithm which connected deeply with technology used for its implementation. Nowadays VLSI technology acts as a platform for those arithmetic operations. In this operations addition and multiplication plays a significant role. Repeated addition of number gives rise to multiplication which is a peculiar case for squaring process. Multiplication based algorithm is used to reduce the time taken for squaring the binary numbers. Therefore, squaring circuits were implemented and takes part in almost all fields of signal processing, animation and image processing. In this project, the efficiency of squaring circuit will be analysed to achieve better performance in terms of power, speed and area. This circuit will be designed by using VHDL and synthesis will be processed through Xilinx ISE 14.2i.

Keywords*:* Adder, Squarer, Vedic multiplier, VHDL.

I.INTRODUCTION

As a rule of very-large-scale integration (VLSI) technology, there is a rapid growth in electronics industries over the last few decagons. Multipliers and adders are often used to get advanced order of squarer mostly in the field of engineering and technology. Due to the degree of usage in real time applications, multiplier is important for optimization and performance enhancement of squarer circuit.squarer circuit use very fast multiplier, implementing a lesser complex circuit would certainly result in better performance.

II.RELATED WORK

Adder is a very basic component in a central processing unit. The speed of compute becomes the most considerable condition for a designer. The carry lookahead adder is the highest speed adder nowadays. [7]

A Highly efficient arithmetic operations are necessary to achieve the desired performance in many real-time systems and digital image processing applications. In all these applications, one of the important arithmetic operations frequently performed is to multiply and accumulate with a small computational time. As in all the arithmetic operations, it is the squaring which is most important in finding the transforms or the inverse transforms in signal processing. [5]

The performance and efficiency of conventional carry skip adder structure is improved by employing increment and concatenation scheme. A hybrid variable latency structure of carry skip adder consist of a modified parallel structure with a prefix adder consist of a modified parallel structure with a prefix adder to reduce the power consumption and slack time. [9]

A new approach that uses high speed squaring circuit with the Ancient Vedic Mathematics technique like Urdhva Triyagbhyam Duplex-D algorithm. An improvement of multiplier efficiency is obtained by reducing the complexity of the squaring circuit. Squaring is the important mathematical operation in recent DSP application. [4]

III.EXISTING SYSTEM

2-bit binary multiplier and squarer

Figure 1.Block diagram of 2-bit multiplier *[1]*, which consist of two half adder blocks. Here the two inputs are X (X0 X1), Y (Y0 Y1) and the outputs are M (M0 M1 M2 M3).

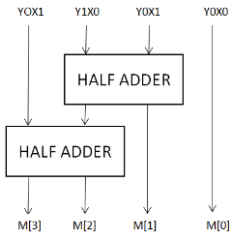


Figure 1. 2-bit Multiplier

By providing the same number at both the inputs of 2-bit multiplier i,e X=Y, we can design the 2-bit squarer circuit. The outputs are
M0 = X0.X0 = X0

M1 (sum bit of first half adder) = X1.X0 XOR X0.X1 = 0
The carry bit of first half adder (X0.X1) is redirected as input to second half adder.
M2 (sum bit of second half adder) = X1.X1 XOR X0.X1 (output carry of first half adder) = X1 XOR X0.X1.
M3 (carry bit of second half adder) = X1.X1 AND X0.X1 (output carry of first half adder) = X1 AND X0.X1.

Therefore, 2-bit squarer circuit is demonstrated in figure 2.

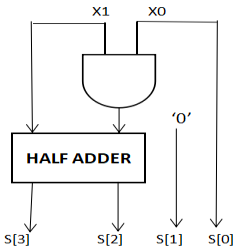


Figure 2. 2-bit Squarer circuit

The 2-bit squarer circuit is further modified. The half adder of figure 2 adds X1 and X0.X1. The generated outputs are
Sum(s[2]) = X1 XOR (X1.X0) = X1.(NOT X0)
Carry(s[3]) = X1.X0
From the above expression, the half adder can be replaced with NOT and AND gates. The modified squarer is shown in figure 3.

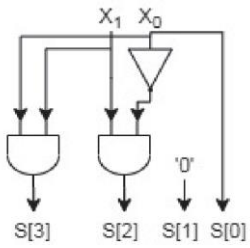


Figure 3. Modified squarer

Vedic based 4-bit multiplier

The block diagram of Vedic based 4-bit multiplier *[2]* is shown in the Figure 4,to which the a(a0 a1 a2 a3) and b(b0 b1 b2 b3) are given as inputs and produces 8-bit outputs m(m0 m1 m2 m3 m4 m5 m6 m7).

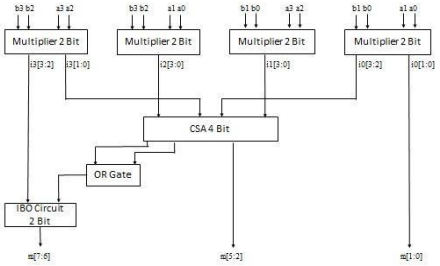


Figure 4. 4-bit Vedic multiplier

For designing the efficient squaring circuit which is proposed in this paper, the above Vedic based multiplier has been used.

IV.PROPOSED SYSTEM

Parallel adders [6] are used to find the arithmetic sum of two numbers which is more than one bit in length and corresponding pairs of bits are processed in parallel form. The efficiency of a squaring circuit can be improved by using parallel adders. A carry and sum are generated by the parallel adder at last the MSB to be added.An 'n' bit parallel adder needs'n' full adders to perform the operation. Few parallel adders which we are used in our paper is discussed below,

Ripple Carry Adder

Generally, in ripple carry adder [11], the carry bit used to ripple through all the stages of the adder. Addition operation of two bit input is performed by one full adder, similarly for multiple bits RCA can be used.

Carry Look Ahead Adder

Carry look ahead adders [7] are composed with 2 process namely carry generation (G) and carry propagation (P).
P = A XOR B
G = A AND B
The outputs (sum and carry) can be generated,
S = P XOR G
C = G OR (P AND C)

Carry Skip Adder

Carry Skip Adder (CSKA) [9] uses skip logic in the propagation of carry. Carry Skip Adder is a fast adder compared to ripple carry adder. Carry Skip Adder provides minimum delay.The expressions for the output of carry skip adder is,

P = A XOR B
S = P XOR C
C = (A AND B) OR (P AND C)

Carry Select Adder

It consist of two parallel RCA's and multiplexers to produce sum and carry as output. The inputs given to the first and second set of RCA remains same but the carry input is '0' for first set and '1' for the second set. Carry Select Adder [8] consume low power consumption and it provide less complexity.

Carry Increment Adder

It has half adder blocks which perform the increment operations.Carry Increment Adder [10] occupies less area and it improves the delay performance of the circuits.

By applying the above concepts in related work, a 4-bit squarer circuit has been developed. The enumeration steps for the proposed 4-bit squarer are given below with the input 'a',

$$\begin{array}{r} a3a2a1a0 \\ \times\ a3a2a1a0 \\ \hline (a3a2) \times (a1a0) \quad (a1a0) \times (a1a0) \\ (a3a2) \times (a3a2) \quad (a3a2) \times (a1a0) \end{array}$$

The terms (a1a0) × (a1a0) and (a3a2) × (a3a2) are generated by using 2-bit squarer and the term (a3a2) × (a1a0) is produced by using 2-bit multiplier. The outputs are shown below

(a1a0) x (a1a0): i0
(a3a2) x (a1a0): i1
a3a2) x (a3a2): i2

It can be spotted that i1 is added 2 times. Therefore, the output from the 2-bit multiplier can be represented by shifting i1 one bit to left side. Due to this reduction, the CSA is replaced by normal 4-bit full adder in the middle stage. The output of OR gate behaves as a control signal for 2-bit IBO module. Figure 5 shows the block diagram of proposed 4-bit squarer.

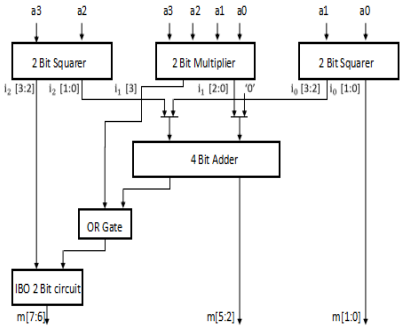


Figure 5. Block diagram for proposed 4-bit squarer circuit

The proposed system can be used for the implementation of 8-bit squarer which can be further used to design 16-bit squarer. Figure 6 and figure 7 shows the block diagram of proposed 8-bit squarer circuit and 16-bit squarer circuit.

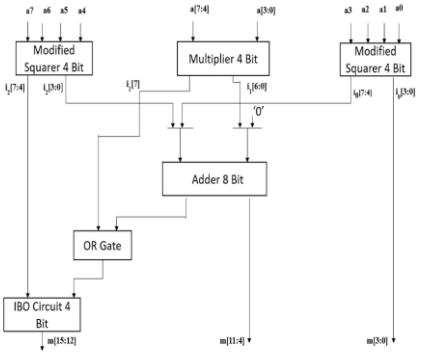


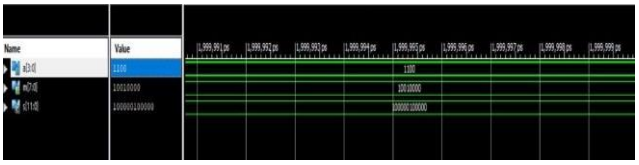
Figure 6. Block diagram for proposed 8-bit squarer circuit

V.EXPERIMENTAL RESULTS

This discussion is about the simulated results of the proposed 4-bit, 8-bit squarer circuits,

Simulated waveform for 4-bit squarer circuit

For 4-bit squarer, input is denoted as a[3:0] which represents a0, a1, a2, a3 and the simulated output is denoted as m[7:0] which represents m0, m1, m2, m3, m4, m5, m6, m7.

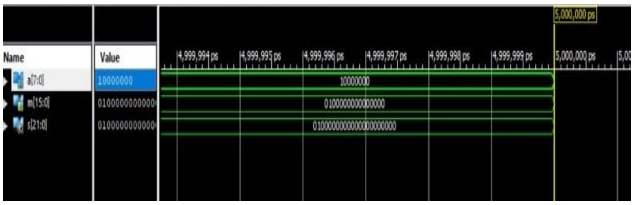


The simulated waveform for proposed 4-bit squarer circuit using Xilinx platform.

Input a[3:0] = 1100 (12),
Output m[7:0] = 10010000 (144)

Simulated waveform for 8-bit squarer circuit

For 8-bit squarer, input is denoted as a[7:0] which represents a0, a1, a2, a3, a4, a5, a6, a7 and the simulated output is denoted as m[15:0], which represents m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14, m15.



The simulated waveform for proposed 8-bit squarer circuit using Xilinx platform

Input a[7:0] = 10000000(128),
Output m[15:0] = 0100000000000000(16,384)

Synthesis was done on different FPGA families like spartan-3, virtex-4 and virtex-5 using various adders as mentioned above. The performance comparison is tabulated.

Table 1. Performance comparison for 4-bit squaring circuit using different adders on three various FPGA families.

DEVICES	SPARTAN-3			VIRTEX-4			VIRTEX-5		
ADDERS	DELAY (ns)	SLICES	LUTs	DELAY (ns)	SLICES	LUTs	DELAY (ns)	SLICES	LUTs
RIPPLE CARRY	7.931	4	6	4.993	3	6	3.890	6	6
CARRY LOOK AHEAD	7.862	3	6	4.993	3	6	3.890	6	6
CARRY INCREMENT	7.862	3	6	4.993	3	6	3.890	6	6
CARRY SKIP	7.862	3	6	4.993	3	6	3.890	6	6
CARRY SELECT	7.862	3	6	4.993	3	6	3.890	6	6

Table 1 is the performance measure of 4-bit squarer. Various adders were also examined closely and it can be seen that parallel adders are much faster compared to the Ripple carry adder. Using parallel adders in the proposed design shows 0.87% improvement in delay whereas LUT count remains the same irrespective of the adder used.

Table 2. Performance comparison for 8-bit squaring circuit using different adders on three various FPGA families

DEVICES	SPARTAN-3			VIRTEX-4			VIRTEX-5		
ADDERS	DELAY (ns)	SLICES	LUTs	DELAY (ns)	SLICES	LUTs	DELAY (ns)	SLICES	LUT s
RIPPLE CARRY	18.090	24	42	9.809	24	42	7.203	30	30
CARRY LOOK AHEAD	17.769	25	44	9.665	25	44	6.952	30	30
CARRY INCREMENT	20.353	28	48	10.701	28	48	8.717	34	34
CARRY SKIP	17.858	25	46	9.539	25	44	6.952	30	30
CARRY SELECT	17.840	25	44	9.631	25	44	7.286	29	29

Table 2 is the performance comparison for 8-bit circuit gives minimum delay while using carry look ahead adder on spartan-3(17.769ns), carry skip adder produce less delay on virtex-4(9.539ns) and virtex-5(6.952ns). The proposed design in this case showed 9.69% improvement in delay using SPARTAN3 and also the LUT count were also greatly reduced. Various adders were also examined and it was found that using carry look-ahead adder in the proposed squarer was more efficient.

VI.CONCLUSION

A multiplication and the squaring algorithm with a small look-up table, which are based on the classical squaring process. This project presents with a whole range of numbers converted into binary digits to identify the squarer output for corresponding binary inputs. The efficiency of the classical process increases the speed in public key cryptography techniques which is more beneficial in data encryption. This project describes the sequential process of digital squaring. This approach made a way to achieve a better performance than the prevail circuit. Based on the simulated result, it can be concluded that there is a improvement in terms of delay, Slices and LUTs, This design can also be expanded for higher bits like 32,64 and so on. The higher bit architecture circuit will be flexible and has a very vast scope in future in higher security required system which comes under the field of data encryption technology.

REFERENCES

[1] D. Kumar and M. Bharathi, *"A high speed and efficient design for binary number squaring using Dwandwa yoga,"* International Journal of Advanced Research in Computer Engineering & Technology, vol. 1, no. 4, pp. 476-479, Jun. 2012.

[2] S. Akhter and S. Chaturvedi, *"Modified binary multiplier circuit based on Vedic mathematics,"* in Proc. IEEE 2019 International Conference on Signal Processing and Integrated Networks, 2019, pp. 234-237.

[3] S. Akhter, *"VHDL implementation of fast N×N multiplier based on Vedic mathematics,"* in Proc. 18th European

Conference on Circuit Theory and Design, 2007, pp. 472-475.

[4] R. Jaikumar, M. Karpagam, and L. Raju, *"A novel approach to implement high speed squaring circuit using ancient Vedic mathematics techniques,"* International Journal of Applied Engineering Research, vol. 10, no. 67, pp. 1-6, 2015.

[5] P. S. Kasliwal, B. P. Patil, and D. K. Gautam, *"Performance evaluation of squaring operation by Vedic mathematics,"* IETE Journal of Research, vol. 57, no. 1, pp. 39-41, Jan.-Feb. 2011.

[6] NagendraC., Irwin, M. J., & Owens, R. M. (1996). "*Area-time-power tradeoffs in parallel adders*". IEEE Transactions on circuits and systems.

[7] Yu-Ting Pai, & Yu-Kung Chen.(n.d.) "*The Fastest Carry Look Ahead Adder*" Second IEEE International Workshop on Electronic Design, Test and Applications.

[8] Katreepalli, R., & Haniotakis, T.(2017) "*High Speed Power Efficient Carry Select Adder Design".* 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI).

[9] Karthik D., & Jayamani S.(2016).*"High speed energy efficient carry skip adder operating at different voltage supply".* 2016 International Conference on Wireless Communications, Signal Processing & Networking (WiSPNET).

[10] Grad, J., & Stine, J.E.(n.d.). *"Low Power Binary Addition Using Carry Increment Adders".* 2006 IEEE International Symposium on Circuits and Systems.

[11] Archana, S., & Durga, G.(2014). *"Design of low power and high speed ripple carry adder".* 2014 International Conference on Communication and Signal processing.