

---

## Approximate Compressors based Inexact Vedic Dadda Multipliers

---

G.Suresh

Department of ECE, Sree Venkateswara Engineering College for Women, Tirupati, Andhra Pradesh, India  
[suresh.gurapu@gmail.com](mailto:suresh.gurapu@gmail.com)

Received: 06th October 2017 Accepted: 14th November 2017, Published: 31st December 2017

### Abstract

In correct registering is a growing world outlook for computation at nanoscale for error tolerant applications like in multimedia and image processing. PC number juggling offers critical operational focal points for estimated figuring, a broad writing exists on rough adders. However, this paper has concentrates on compression based multiplier design, which offer major advantages in terms of both circuit-level and error figures of merit. The approximate 4:2 compressors demonstrate a critical lessening in transistor tally, power consumption and delay contrasted and a correct outline. The Vedic mathematics based Urdhva-tiryagbhyam sutra is used to develop 8x8 Dadda multipliers using 4x4 Dadda multipliers. Four different approximate multiplier designs are proposed and are verified using XilinxISE14.5 Tool for the target device Spartan3E XC3S500E-5FG320. The simulation and synthesis results prove that the fourth design is 33.2% efficient in terms of power-delay-product and 96.7% efficient in terms of area-delay-product when compared with other three approximate multipliers.

**Keywords:** Inexact Computing, Majority Gate, Cost Metrics, Trade-off.

### INTRODUCTION

Most PC math applications are executed for computerized rationale circuits, in this working with a high level of unwavering quality and exactness. Nonetheless, numerous applications, for example, in mixed media and picture planning can persist botches [5] and imprecision in computation and still make huge and important results. Exact models and counts are not for the most part sensible or gainful for use in these applications. The perspective of ambiguous count relies upon loosening up totally correct and absolutely deterministic building modules when for example, arranging essentialness gainful frameworks. This permits loose calculation [4] to divert the current procedure of advanced circuits and frameworks by exploiting abatement in multifaceted nature and cost with a potential increment in execution and power proficiency. Surmised processing depends on property to configuration disentangled, so far inexact circuits working at further execution or potentially bring down power utilization contrasted and exact rationale circuits [1].

Addition and multiplication are often used in these applications. For addition, full adders have been analyzed in detail and various rough outlines have been proposed [1]. A few new measurements are proposed and a correlation is made among a portion of the adder designs. The mistake separate is characterized as the number juggle separation between a mistaken and the right yields for a given info. The mean error distance (MED) and normalized error distance (NED) are then proposed. The NED is most invariant with the evaluation of an execution and is in this manner helpful in the unwavering quality appraisal of a particular plan. The tradeoff amongst accuracy and power has been quantitatively surveyed in [1]. As of late, inexact multipliers have additionally picked up hugeness as a result of their significance in number juggling operations [2], a few estimated 4:2 compressors have been proposed in the lessening of the fractional results of a Dadda tree. In this paper, the surmised compressors are used to outline 8x8 piece multipliers [3] by a novel parcel of the halfway items.

Be that as it may, the plan of inexact multipliers has gotten less consideration. Increase thought the rehashed whole of halfway items, in any case, the clear use of surmised adders when outlining an estimated multiplier isn't reasonable, in glow of the reality that would be to a great degree inefficient in regards to exactness, hardware multifaceted design and other execution estimations. A couple of estimated multipliers are proposed in the written work. A vast segment of these plans utilize a truncated duplication technique, they gauge the minimum critical segments of the halfway items as a consistent.

A loose display multiplier is used for neural framework applications by blocking a bit of the slightest huge bits in the halfway items (and in this manner evacuating a couple of adders in the bunch). A truncated multiplier with a redress consistent is proposed in [6]. For a  $n \times n$  multiplier, this outline computes the entirety of the  $n+k$  most noteworthy sections of the halfway things and truncates the other  $n-k$  sections. The  $n+k$  bit result is then changed in accordance with  $n$  bits. The decrease blunder (i.e. the mistake produced by truncating the  $n-k$  slightest huge bits) and adjusting blunder (i.e. the mistake made by modifying the result to  $n$  bits) are create in the consequent stage.

The change reliable ( $n+k$  bits) is possible to the evaluated estimation of the total of these mistakes to lessen the blunder separate [9].

The section II discusses about the exact and approximate compressors. The compressor based Vedic Dadda multipliers are discussed in section III. The results are discussed in section IV. Finally the paper is concluded.

**I. COMPRESSORS**

To bring down the dormancy of the incomplete item collection organize, 4-2 and 5-2 compressors have been broadly utilized these days for fast multipliers. Attributable to its customary interconnection, the 4-2 compressor is perfect for the development of consistently organized Wallace tree with low many-sided quality. A few 4-2 compressor circuits have been proposed for low-control applications [2]. Some of these can work at low supply voltages however require unnecessary number of transistors because of their corresponding CMOS structures, others utilize more modest number of transistors yet neglect to work at ultra low voltages, or do not have the driving ability to drive the following level of sub circuits [8].

The inclusion of extra supports for each yield ports to give the yield drive builds the exchanging exercises and henceforth the power scattering. Higher information compressors have additionally been examined by scientists and quick 5:2 compressors have been progressively utilized in extensive word-measure multiplier and high accuracy increase collectors. The vast majority of the exploration on high-input compressors concentrates on the advancement of circuit structure for fast applications at standard input voltages. With patterns in VLSI toward profound sub micrometer innovation, circuits working dependably at sub-1 V will soon turn into a reality as the materials used to frame the transistors can't withstand an electric field of boundless quality, and as transistors get littler, the field quality increments if the input voltage is held constant. Since supply voltage has a quadratic commitment to the power dispersal, bringing down the supply voltage is likewise a clear method for decreasing force utilization [7]. Be that as it may, the significant issue with diminishing the given voltage is that the speed up the circuits is additionally debased. In this manner, there is a solid catalyst to recharge the full custom number juggling cells to accomplish high power proficiency for VLSI circuits working at ultra low supply voltages.

**A. Exact Compressors**

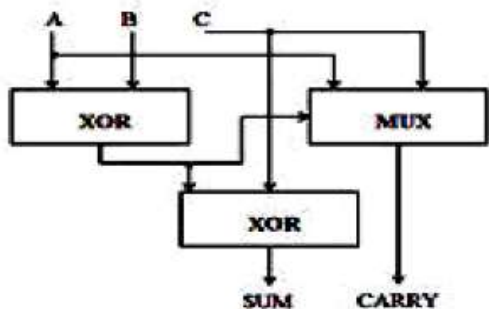


Fig.1 3:2 Compressor

Table 1: Truth Table of 3:2 Compressor

Inputs			Sum	Carry
A	B	C		
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The fig.1 and table 1 show the 3:2 exact compressor basic structure and truth table. The 3:2 compressors are similar to full adder except that this uses two XOR gates and one multiplexer. This has two XORs delay.

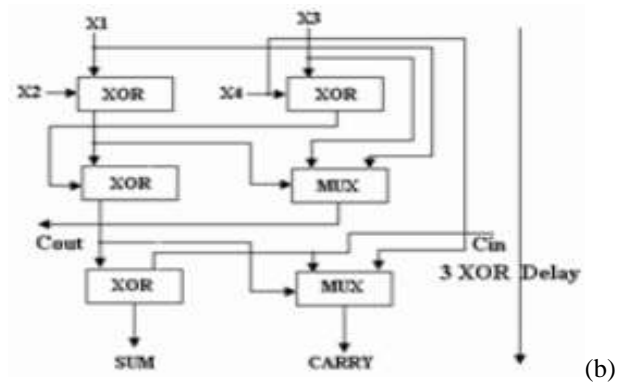
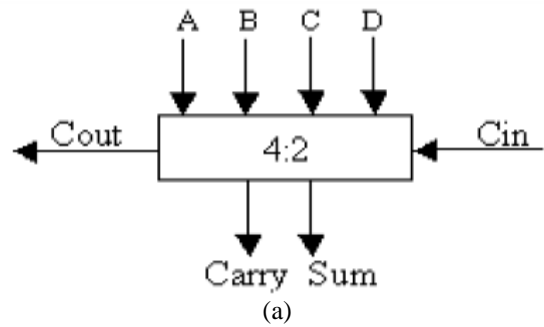


Fig.2 4:2 Compressor (a) Logic Symbol and (b) Internal Diagram

Table 2: Equivalent Table of 4:2 Compressors

n	Cin	Cout	Carry	Sum
0	0	0	0	0
1	0	0	0	1
2	0	+	+	0
3	0	1	0	1
4	0	1	1	0
0	1	0	0	1
1	1	0	1	0
2	1	+	+	1
3	1	1	1	0
4	1	1	1	1

The fig.2 shows the 4:2 compressors' logic symbol and its internal structure. The truth table is shown in table 2

indicates that  $n=X4X3X2X1$  i.e., a combination of four bits. The 4:2 compressor is similar to the output of given two full adders i.e., it uses 4 XOR gates and two multiplexers. This has three XORs delay.

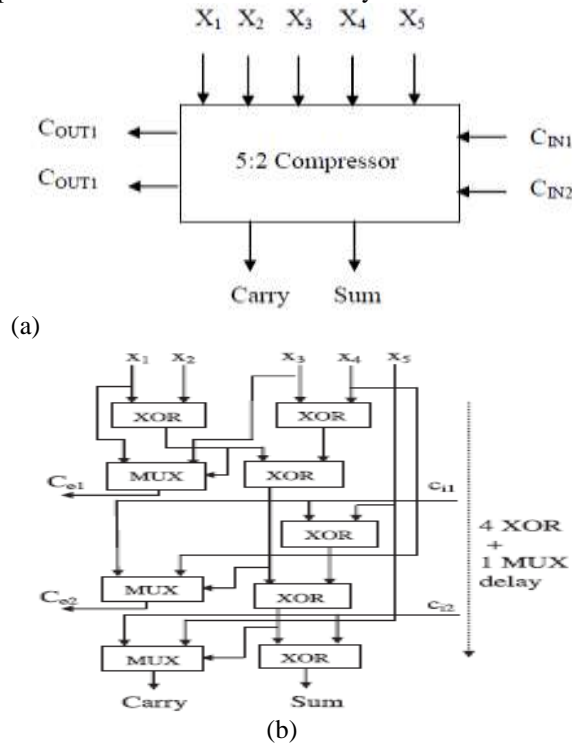


Fig.3 5:2 Compressor (a) Logic Symbol (b) Internal Diagram

Table 3: Truth Table of 5:2 Compressor

n	$C_{in1} \oplus C_{in2} = 0$					$C_{in1} \oplus C_{in2} = 1$				
	$a_4 \oplus a_5$	$C_{out1}$	$C_{out2}$	Sum	Carry	$a_4 \oplus a_5$	$C_{out1}$	$C_{out2}$	Sum	Carry
0	0	0	$a_4.a_5$	0	$C_{in1}.C_{in2}$	0	0	$a_4.a_5$	1	0
1	0	0	$a_4.a_5$	1	$C_{in1}.C_{in2}$	0	0	$a_4.a_5$	0	1
2	0	1	$a_4.a_5$	0	$C_{in1}.C_{in2}$	0	1	$a_4.a_5$	1	0
3	0	1	$a_4.a_5$	1	$C_{in1}.C_{in2}$	0	1	$a_4.a_5$	0	1
0	1	0	0	1	$C_{in1}.C_{in2}$	1	0	0	0	1
1	1	0	1	0	$C_{in1}.C_{in2}$	1	0	1	1	0
2	1	1	0	1	$C_{in1}.C_{in2}$	1	1	0	0	1
3	1	1	1	0	$C_{in1}.C_{in2}$	1	1	1	1	0

The fig.3 shows the 5:2 compressors' logic symbol and its internal structure. The truth table is shown in table 3 indicates that  $X5X4X3X2X1=a5a4a3a2a1a$  i.e., a combination of five bits. The 5:2 compressor is just like the utilization of three full adders i.e., it uses six XOR gates and three multiplexers. This has four XORs delay and in addition one MUX delay.

**B. Approximate Compressors**

The two plans of an inexact compressor are proposed. Normally to design a harsh 4-2 compressor, it is possible to replace the right full-snake cells in Figure3 by a construed full-wind cell. Be that as it may, this isn't extremely effective, in light of the way that it delivers no

less than 17 wrong outcomes out of 32 conceivable yields, i.e. the mistake rate of this estimated compressor is over 53% (where the blunder rate is given by the proportion of the quantity of mistaken yields over the aggregate number of yields). Two distinct plans are proposed alongside decrease the mistake rate, these outlines offer critical execution change contrasted with a correct compressor regarding delay, quantity of transistors and power utilization.

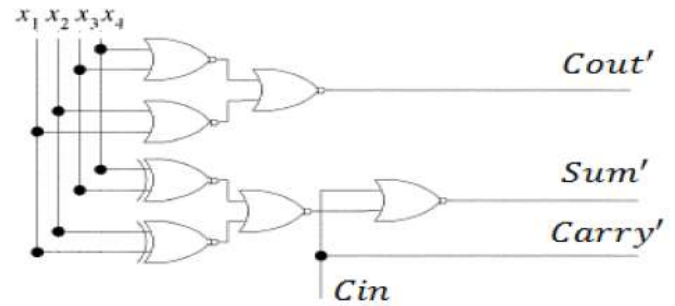


Fig.4 Gate level implementation of Design1

Table 4: Truth Table of Design1

$c_{in}$	$X_1$	$X_2$	$X_3$	$X_4$	$c_{out}'$	carry'	sum'	Difference
0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	1	-1
0	0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0	0
0	0	1	1	0	1	0	1	0
0	0	1	1	1	1	0	1	0
0	1	0	0	0	0	0	1	0
0	1	0	0	1	1	0	0	0
0	1	0	1	0	1	0	0	0
0	1	0	1	1	1	0	1	0
0	1	1	0	0	0	1	0	0
0	1	1	0	1	1	0	1	-1
0	1	1	1	0	1	0	1	0
0	1	1	1	1	1	0	1	0
0	1	1	1	1	1	1	1	-1
1	0	0	0	0	0	1	0	1
1	0	0	0	1	0	1	0	0
1	0	0	1	0	0	1	0	0
1	0	0	1	1	0	1	0	0
1	0	1	0	0	1	1	0	0
1	0	1	0	1	1	1	0	1
1	0	1	1	0	1	1	0	0
1	0	1	1	1	1	1	0	0
1	1	0	0	0	0	1	0	1
1	1	0	0	1	1	1	0	0
1	1	0	1	0	1	1	0	0
1	1	0	1	1	1	1	0	0
1	1	1	0	0	0	1	0	-1
1	1	1	0	1	1	1	0	0
1	1	1	1	0	1	1	0	0
1	1	1	1	1	1	1	0	-1

As appeared in Table 4, the convey yield in a correct compressor has a similar estimation of the info  $C_{in}$  is 24 out of 32 states. Subsequently, inexact outlines always consider this component. In Design 1, is suggest to improved the  $C_{in}$  by adjusting the estimation of the another 8 yields. The nearness of a couple of errors in the total banner will realizes an abatements of the deferral of delivering the estimated entirety and the general postponement of the plan. The distinction in the measure of  $c_{out}$  in a couple of states, may decrease the blunder remove gave by inexact convey and total and can be more improved in this outline. The design1 has 12 off base yields out of 32 yields.

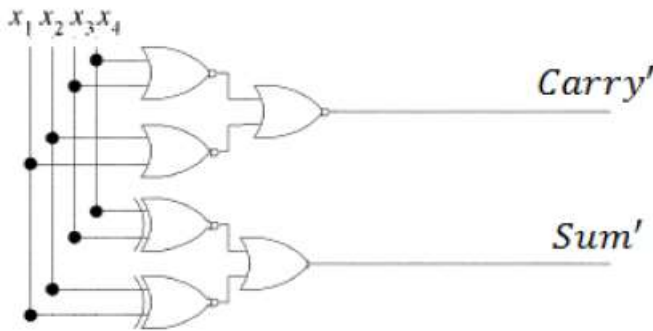


Fig.5 Gate level implementation of Design2

Table 5: Truth Table of Design2

$x_1$	$x_2$	$x_3$	$x_4$	$carry'$	$sum'$	$difference$
0	0	0	0	0	1	1
0	0	0	1	0	1	0
0	0	0	1	0	1	-1
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	-1
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	1	1	-1

The design2 of a surmised compressor additionally increment execution and additionally lessening the blunder rate. Since the convey and cout yields have a similar weight, the proposed conditions for the normal pass on and cout in the previous part can also be exchanged. In this new plan, convey utilizes the correct hand side of [8] and cout is also equal to cin, since cin is zero, cout and cin will be zero. Thusly, cin and cout can be unnoticed in the hardware outline. The deferral of the basic way of this estimated configuration is  $2\delta$ , so it is  $1\delta$  not as much as the past plans; in addition, a further diminishment in the quantity of doors is proficient. This outline has along these lines 4 mistaken yields out of 16 yields, so its blunder rate is presently lessened to 25%. This is a greatly positive part, since it shows that on a probabilistic introduce, the imprecision of the proposed arrangement is smaller than the other available plans.

## II. COMPRESSOR BASED VEDIC DADDA MULTIPLIER

The exact dada multiplication of 8x8 bits is as shown in fig.6. It is actually based on the tree structure where the eight rows of eight dots indicate the partial product matrix formed by AND gates. The four reduction levels are achieved for 6, 4, 3 and 2 levels by using the full adder and half adders for the three and two bits grouped together respectively. This method of

partial product reduction consumes less delay and remains the fastest multiplier.

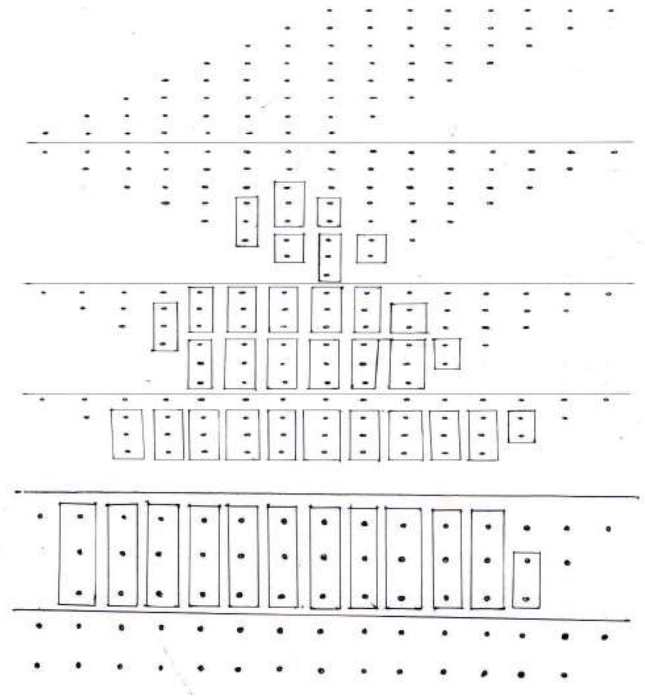


Fig.6 8x8 Dadda Multiplier for Exact Computation

As the vedic mathematics proves to be ever efficient in improving the speed of the designs, we have chosen Urdhva-tiryagbhyam sutra among the 16 sutras of vedic mathematics. The 16 sutras are Ekadhikena Purvena, Nikhilam Navatascaramam Dasatah, Urdhva-tiryagbhyam, Paravartya Yojayet, Sunyam Samyasamuccaye, (Anurupye) Sunyamanyat, Sankalana-vyavakalanabhyam, Puranapuranaabhyam, Calana-Kalanabhyam, Yavadunam, Vyastisamastih, Sesanyankena Caramena, Sopantadvayamantyam, Ekanyunena Purvena, Gunitasamuccyah and Gunakasamuccayah.

The 4x4 bit multipliers have been actualized and additionally used for the 8x8 piece increase. Every one of the plans are actualized utilizing the Dadda tree procedure by influencing utilization of various 4:2 compressors in the diminishment to organize. Utilizing the compressors, the  $4 \times 4$  bit item requires one decrease organize, making the item count speedier. For the main outline, design1\_dadda4x4mul, the plan 1 compressor appeared in Fig.4 is used for the incomplete item decrease arrange. The Dadda tree usage of design1\_dadda4x4mul is appeared in Fig.7 and its investigation is appeared in fig.8, just two compressors are required in the fractional item lessening stage.

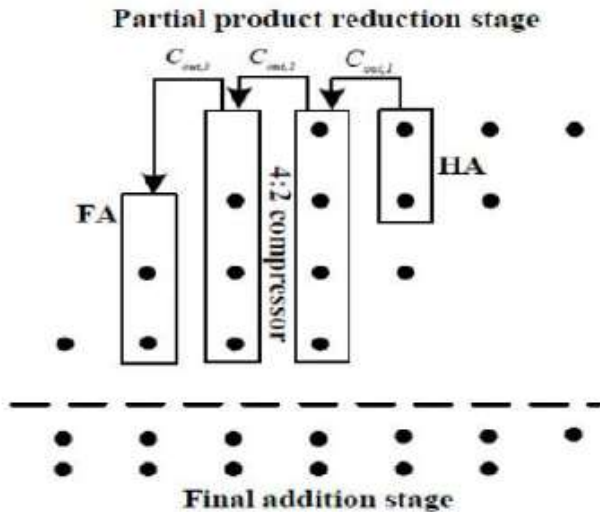


Fig.7 4x4 Dadda Multiplier for design 1

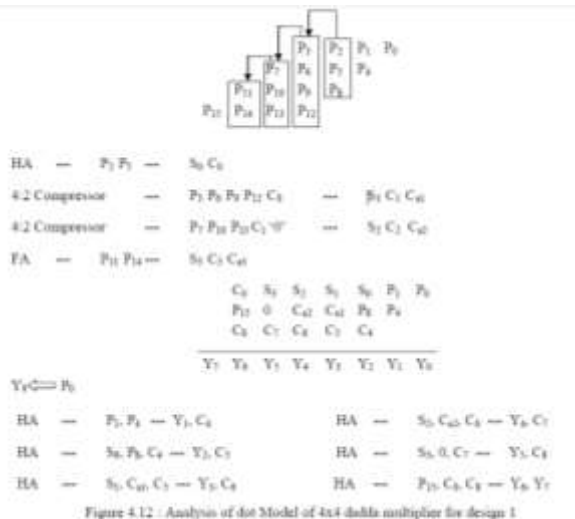


Fig.8 Analysis of dot model of 4x4 Dadda multiplier for design 1

Additionally, for the second plan, design2\_dadda4x4mul, outline 2 compressor appeared in Fig.5 is used for the decrease arrange and as outline 2 does not have a convey to the following stage, the plan is somewhat not the same as design1\_dadda4x4mul. The Dadda tree usage of design2\_dadda4x4mul is appeared in Fig.9 and its investigation is appeared in fig.10. Just a single compressor is required in the lessening stage, which fundamentally improves the plan. For the precise 4x4 multiplier, acc\_dadda4x4mul, the Dadda tree execution is the same as design1\_dadda4x4mul, on the grounds that the plan 1 compressor and the exact compressor utilize similar sorts of circuits. Henceforth, just exact compressor should be used for the place of the plan 1 compressor. In this multiplier, two precise compressors are required in the decrease arrange.

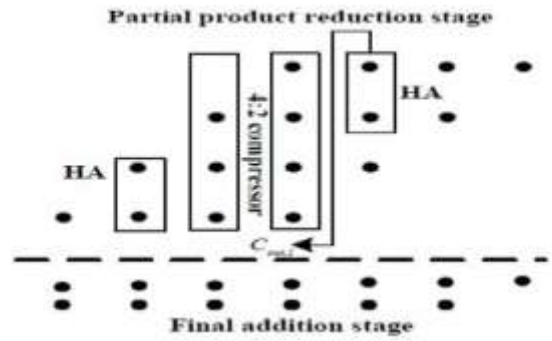


Fig.9 4x4 Dadda Multiplier for design 2

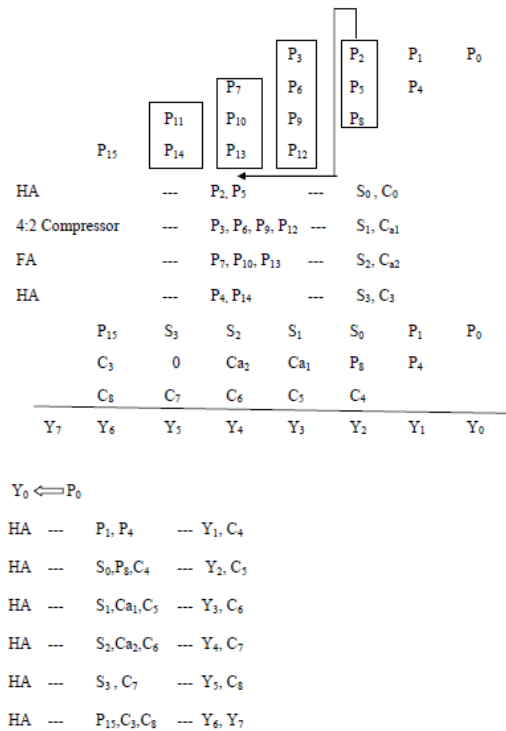


Fig.10 Analysis of dot model of 4x4 Dadda multiplier for design 2

These 4x4 multipliers composed are used as a part of the usage of 8x8 multipliers. The incomplete item tree of the 8x8 increase is separated to 4 results of 4x4 modules utilizing the strategy of recursive augmentation, as appeared in Fig.11. The benefit of breaking the items is to get littler duplication hinders that are performed in parallel and in this manner speedier. At that point, they only should be included, as indicated by Fig.12 to acquire the last item. The neighboring compressors inside a chain in the halfway item diminishment organize are appeared in fig.13. There are three unique decisions accessible to be utilized for the required four 4x4 items. For high precision plans, acc\_dadda4x4mul can be utilized for the three more noteworthy items i.e., AHBH, AHBL and ALBH, and any of the other two estimated outlines can be utilized for the minimum critical item, i.e., ALBL. The proposed multiplier design1\_dadda8x8

utilizes design1\_dadda4x4mul for the calculation of ALBL, the proposed multiplier design2\_dadda8x8 utilizes design2\_dadda4x4mul for a similar calculation. The two multipliers utilize acc\_dadda4x4mul for the calculation of the other three items.

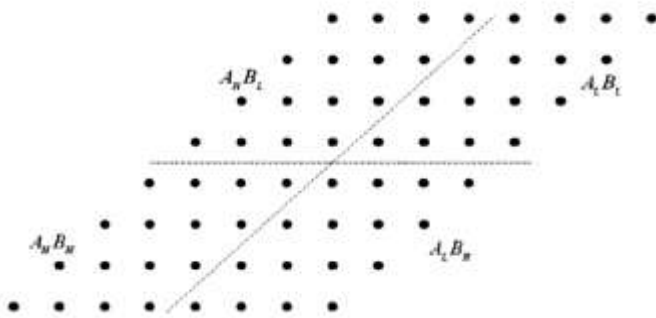


Fig.11 8x8 Dadda Multiplier using four 4x4 Dadda Multipliers

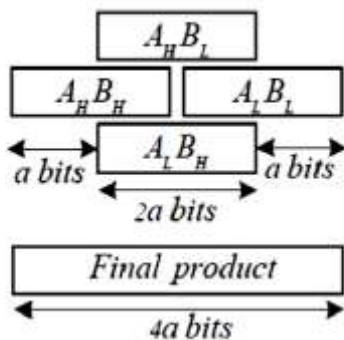


Fig.12 Arrangement of the inputs for recursive multiplication

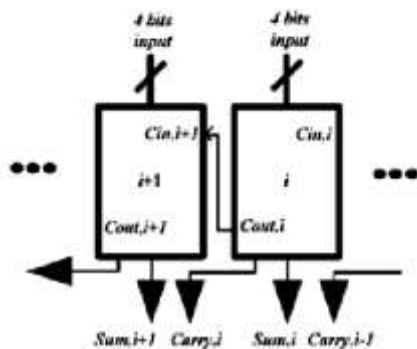


Fig.13 contiguous compressors within a chain in the partial product reduction stage

Many cases, exactness isn't a strict prerequisite and thus, it can be exchanged off with power, deferral and region. When outlining power proficient, low precision multipliers, acc\_dadda4x4mul is utilized just to calculate the most huge item and either design1\_dadda4x4mul or design2\_dadda4x4mul can be utilized for figuring the other three less huge items.

Opt\_dadda4x4mul is designed by replacing exact compressor with optimized compressor. Mul3\_dadda4x4mul is designed by replacing exact

compressors with design 1 and optimized compressor. The proposed multiplier design3\_dadda8x8 uses mul3\_dadda4x4mul for slightest huge item ie., ALBL and acc\_dadda4x4mul for the other three items, while design4\_dadda8x8 utilizes mul3\_dadda4x4mul for minimum critical item ie., ALBL and opt\_dadda4x4mul for the other three items. After the computation of the four items, the two items and are included utilizing a 9-bit viper (so not a 16-bit snake in this way sparing equipment), the outcome is included with the other two items and utilizing a 16-bit viper.

### III. RESULTS AND DISCUSSION

The design are coded in VHDL [10] and are synthesized for Spartan3E FPGA with the device XC3S500E-5FG320, using Xilinx ISE14.5 Tool. The functionality is verified using Xilinx ISIM simulator.

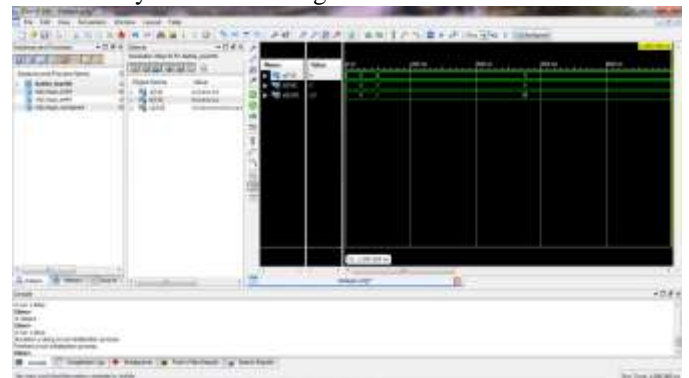


Fig.14 Simulation result of exact\_dadda8x8 Multiplier

All multipliers used in this project perform 8x8 multiplications. The simulation result for the exact\_dadda8x8 multiplier is shown in the Fig.14. The inputs are applied as a = 00000000 (0) and b = 00000000 (0), the output is obtained as X = 0000000000000000 (0). Similarly if the inputs are applied as a = 00000100 (4) and b = 00000101 (5), then the output is obtained as X = 00000000000010100 (20).

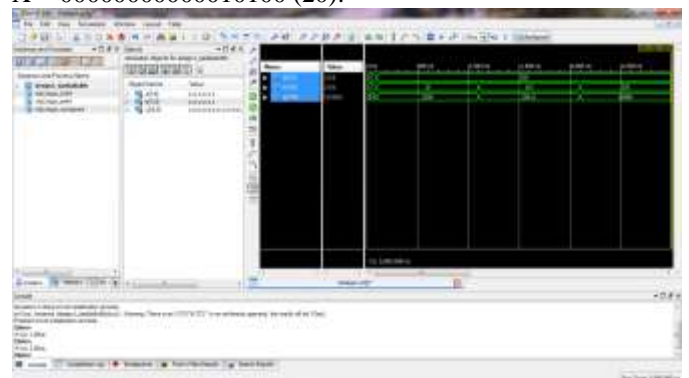


Fig.15 Simulation result of design1\_dadda8x8 multiplier

	Inputs		Output
1	a = 00000000 (0)	b = 00000000 (0)	x = 0000000000011000 (24)
2	a = 11111111 (255)	b = 00001010 (10)	x = 0000100111110110 (2550)
3	a = 11111111 (255)	b = 01100101 (101)	x = 0110010000001011 (25611)
4	a = 11111111 (255)	b = 11111111 (255)	x = 1101001011100001 (53985)

Fig.16 Inputs and Outputs of design1\_dadda8x8 multiplier

The simulation result for the design1\_dadda8x8 multiplier is shown in the Fig.15. The inputs applied to this design and the outputs obtained are as shown in fig.16. Hence a deviation of 24 results due to usage of approximate compressors.

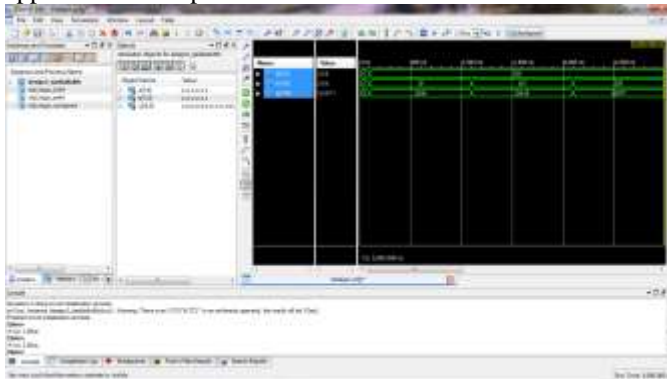


Fig.17 Simulation result of design2\_dadda8x8 multiplier

	Inputs		Output
1	a = 00000000 (0)	b = 00000000 (0)	x = 000000000001000 (8)
2	a = 11111111 (255)	b = 00001010 (10)	x = 0000100111011110 (2526)
3	a = 11111111 (255)	b = 01100101 (101)	x = 0110010000010011 (25619)
4	a = 11111111 (255)	b = 11111111 (255)	x = 1101001011011001 (53977)

Fig.18 Inputs and Outputs of design2\_dadda8x8 multiplier

The simulation result for the design2\_dadda8x8 multiplier is shown in the Fig.17. The inputs applied to this design and the outputs obtained are as shown in fig.18. Hence a deviation of 8 results due to usage of approximate compressors.

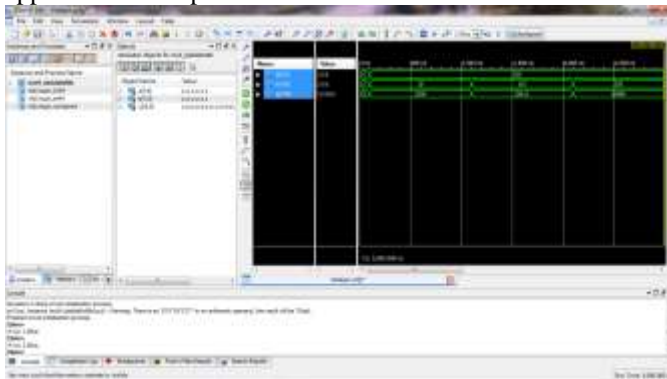


Fig.19 Simulation result of design3\_dadda8x8 multiplier

	Inputs		Output
1	a = 00000000 (0)	b = 00000000 (0)	x = 000000000001000 (8)
2	a = 11111111 (255)	b = 00001010 (10)	x = 0000100111110110 (2550)
3	a = 11111111 (255)	b = 01100101 (101)	x = 0110010000001011 (25611)
4	a = 11111111 (255)	b = 11111111 (255)	x = 1101001011100001 (53985)

Fig.20 Inputs and Outputs of design3\_dadda8x8 multiplier

The simulation result for the design3\_dadda8x8 multiplier is shown in the Fig.19. The inputs applied to this design and the outputs obtained are as shown in fig.20. Hence a deviation of 8 results due to usage of approximate compressors.

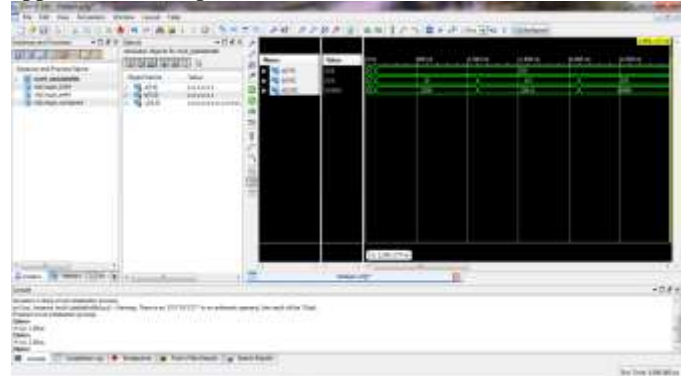


Fig.21 Simulation result of design4\_dadda8x8 multiplier

	Inputs		Output
1	a = 00000000 (0)	b = 00000000 (0)	x = 000000000001000 (8)
2	a = 11111111 (255)	b = 00001010 (10)	x = 0000100111110110 (2550)
3	a = 11111111 (255)	b = 01100101 (101)	x = 0110010000001011 (25611)
4	a = 11111111 (255)	b = 11111111 (255)	x = 1101001011100001 (53985)

Fig.22 Inputs and Outputs of design4\_dadda8x8 multiplier

The simulation result for the design4\_dadda8x8 multiplier is shown in the Fig.21. The inputs applied to this design and the outputs obtained are as shown in fig.22. Hence a deviation of 8 results due to usage of approximate compressors.

TABLE V I. COMPARISON BASED ON OBTAINED SYNTHESIS AND SIMULATION RESULTS

Designs	Delay (ns)	Power (mW)	No. of LUTs	No. of Slices	PDP (pJ)	ADP (µm²ns)	Deviation Error
Exact_dadda8x8	21.084	98	189	98	2.06623	390517.84	0
Design1_dadda8x8	20.328	95	156	81	1.93116	256864.60	24
Design2_dadda8x8	15.301	89	117	63	1.36178	112783.67	8
Design3_dadda8x8	15.231	89	119	64	1.34032	115999.29	8
Design4_dadda8x8	10.291	87	26	13	0.89531	3478.378	8

The Table.VI shows the comparison of all multipliers specifications like delay, power, number of LUTs, number of Slices, Power delay product(PDP), Area delay product(ADP) and percentage of error. All the specifications of exact multiplier are more than the designed multipliers. But among the designed approximate multipliers mul4\_dadda8x8 has less delay, power consumption when compared with other three approximate multipliers.

CONCLUSION

Inaccurate processing is a developing worldview for calculation at nanoscale. PC number juggling offers critical operational points of interest for estimated figuring; a broad writing exists on surmised adders. In this paper has at firstly centered around pressure as used as for multiplier. By utilizing fitting plan of a surmised compressor, multipliers can be intended for vague registering. These multipliers offer critical preferences

regarding both circuit-level and mistake figures of legitimacy. In this venture we utilized two plans of estimated 4-2 compressors. These rough compressors are used in the decrease module of four estimated multipliers. The estimated compressors demonstrate a huge decrease in transistor check, control utilization and deferral contrasted and a correct outline. Further vedic mathematics based Urdhva-tiryagbhyam sutra is used to develop 8x8 Dadda multipliers using 4x4 Dadda multipliers. Based on different combinations of approximate compressors, four approximate multiplier designs are developed named design1\_dadda8x8, design2\_dadda8x8, design3\_dadda8x8 and design4\_dadda8x8 multipliers. These designs are verified using XilinxISE14.5 Tool for the target device Spartan3E XC3S500E-5FG320. From the simulation and synthesis results, it is proved that mul4\_dadda8x8 has less delay, power consumption when compared with other three approximate multipliers. The design4\_dadda8x8 multiplier Power-Delay-Product is atleast 33.2% efficient than design3\_dadda8x8 multiplier and its Area-Delay-Product is atleast 96.7% efficient than the design2\_dadda8x8 multiplier. These inexact multipliers can be used in error tolerant applications and low energy computing systems like in image processing, network routers etc.

#### REFERENCES

- [1] Stefania Perri, Pasquale Corsonello, and Giuseppe Cocorullo "Area-Delay Efficient Binary Adders in QCA," Vol. 22, no.5, MAY 2014
- [2] A.Momeni, J.Han, Member, P.Montuschi, Senior Member and F.Lombardi, Fellow "Design and Analysis of Approximate Compressors for Multiplication", 25th FEB 2014, Volume:64 Issue:4.
- [3] J.Han and M.Orshansky, "Approximate Computing: An Emerging Paradigm for Energy - Efficient Design," in ETS'13, Avignon, France, May 27-31, 2013, pp. 1-6.
- [4] J. Liang, J.Han, F.Lombardi, "New Metrics for the Reliability of Approximate and Probabilistic Multipliers," IEEE Transactions on Computers, vol 63, no. 9, 1760 - 1771, 2013.
- [5] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, K. Roy, "IMPACT: IMPrecise multipliers for low power approximate computing," Low Power Electronics and Design (ISLPED) 2011 International Symposium on. 1-3 Aug. 2011.
- [6] Khaing Yin Kyaw, Wang Ling Goh, and Kiat Seng Yeo, "Low-power high-speed multiplier for error-tolerant application", IEEE Trans. Circuits syst. pp. 978-981, 2010.
- [7] M. J. Schutle and E. E. Swartzlander Jr, "Truncated multiplication with correction constant," VLSI Signal Processing, vol.6, pp.388-396, 1993.
- [8] S. S. Kidambi, F. El-Guibaly, Senior Member and A. Antoniou, Fellow IEEE, "Area-Efficient Multipliers for Digital Signal Processing Applications," IEEE Trans. On Circuits and Systems-IL vol. 43, no. 2, Feb 1996.
- [9] L. Dadda, "Some Schemes for Parallel Multipliers", Alta Frequenza, Vol.34, p.349-356, March 1965.
- [10] A. Weinberger, "4:2 Carry-Save Adder Module", IBM Technical Disclosure Bulletin., Vol.23, January 1981. [10]. IEEE Standard VHDL Language Reference Manual Amendment 1: Procedural Language Application Interface. 2007. doi:10.1109/IEEESTD.2007.4299594. ISBN 0-7381-5523-3.