International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

LOW-LATENCY ASIC ARCHITECTURE FOR RECURSIVE KARATSUBA MULTIPLICATION EMPLOYING AN EFFICIENT SQRT CARRY SELECT ADDER

Masroor Rashid[1], Ms Nilofer[2]
[1]PG Scholar, Department of ECE, Shadan Women's College of Engineering and Technology, Hyderabad, masroorrashid247@gmail.com
[2]Asst Professor, Department of ECE, Shadan Women's College of Engineering and Technology nilofer.eng@gmail.com

ABSTRACT

Computation intensive applications such as DSP, image processing, floating point processors and communication technologies today require efficient binary multiplication which usually is the most power and time-consuming block. This paper proposes an efficient design for unsigned binary multiplication to reduce delay. A 16×16-bit multiplier has been designed which is based on Vedic Karatsuba algorithm using reversible logic. It is optimized using adaptive and recursive approach combined with square-root- carry-select-adder. The designs have been coded in Verilog, synthesized in Xilinx ISE.

Index Terms—Square Root Carry Select Adder, reversible logic, Karatsuba Multiplier, Recursive Adaptive Karatsuba Algorithm

INTRODUCTION

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.

The common multiplication method is "add and shift" algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, Vedic multiplier is one of the most popular method. In this lecture we introduce the multiplication algorithms and architecture and compare them in terms of speed, area, power and combination of these metrics.

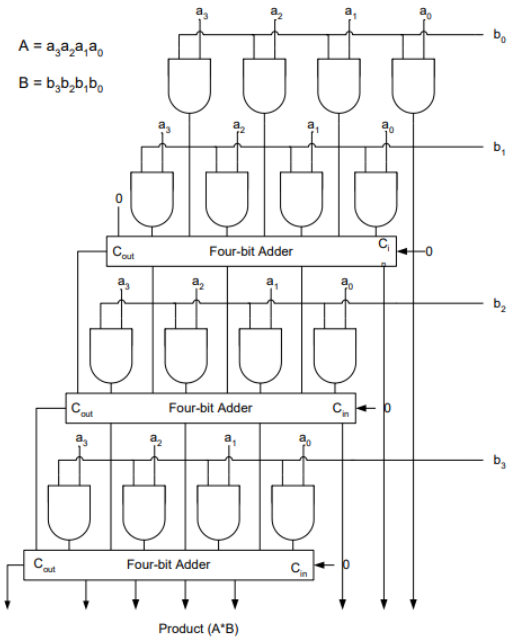
The basic method of multiplier explains below

```
   123
 x 456
 =====
   738  (this is 123 x 6)
  615   (this is 123 x 5, shifted one position to the left)
+ 492   (this is 123 x 4, shifted two positions to the left)
 =====
 56088
```

The binary multiplication also happens in same way of digit multiplication as shown in below example here by getting partial products and gates are used and we are using adder (half adder ,full adder)adding the columns .

```
  1 0 1 0 X 1 0 1
  ---------------
    1 0 1 0
  0 0 0 0
1 0 1 0
---------------
1 1 0 0 1 0
```

An example of 4-bit multiplication method is shown below:



Although the method is simple as it can be seen from this example, the addition is done serially as well as in parallel. To improve on the delay and area the CRAs are replaced with Carry Save Adders, in which every carry and sum signal is passed to the adders of the next stage. Final product is obtained in a final adder by any fast adder (usually carry ripple adder). In array multiplication we need to add, as many partial products

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

as there are multiplier bits. These arrangements are shown in the figure below

In applications like multimedia signal processing and data mining which can tolerate error, exact computing units are not always necessary. They can be replaced with their approximate counterparts. Research on approximate computing for error tolerant applications is on the rise. Adders and multipliers form the key components in these applications. In, approximate full adders are proposed at transistor level and they are utilized in digital signal processing applications.

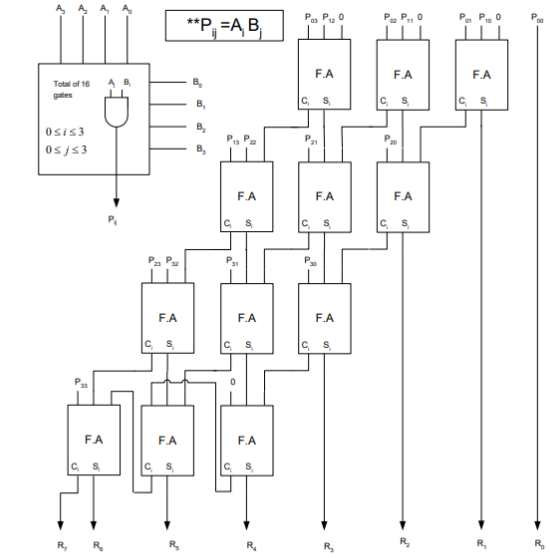


Fig : array multiplier

LITERATURE VIEW

Vijay Kumar Reddy **Modified High Speed Vedic Multiplier Design and Implementation** The proposed research work specifies the modified version of binary Vedic multiplier using Vedic sutras of ancient Vedic mathematics. It provides modification in preliminarily implemented Vedic multiplier. The modified binary Vedic multiplier is preferable has shown improvement in the terms of the time delay and also device utilization. The proposed technique was designed and implemented in Verilog HDL. For HDL simulation, modelsim tool is used and for circuit synthesis, Xilinx is used. The simulation has been done for 4-bit, 8-bit,16-bit, multiplication operation. Only for 16-bit binary Vedic multiplier technique the simulation results are shown. This modified multiplication technique is extended for larger sizes. The outcomes of this multiplication technique are compared with existing Vedic multiplier techniques.

A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "**Design and Analysis of Approximate Compressors for Multiplication**", Inexact (or approximate) computing is an attractive paradigm for digital processing at nanometric scales. Inexact computing is particularly interesting for computer arithmetic designs. This paper deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. These designs rely on different features of compression, such that imprecision in computation (as measured by the error rate and the so-called normalized error distance) can meet with respect to circuit-based figures of merit of a design (number of transistors, delay and power consumption). Four different schemes for utilizing the proposed approximate compressors are proposed and analyzed for a Dadda multiplier. Extensive simulation results are provided and an application of the multipliers to image processing is presented. The results show that the proposed designs accomplish significant reductions in power dissipation, delay and transistor count compared to an exact design; moreover, two of the proposed multiplier designs provide excellent capabilities for image multiplication with respect to average normalized error distance and peak signal-to noise ratio (more than 50 dB for the considered image examples).

C. Liu, J. Han, and F. Lombardi, "**A Low-Power, High-Performance Multiplier with Configurable Partial Error Recovery**", Proc. of IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), [Approximate circuits have been considered for error-tolerant applications that can tolerate some loss of accuracy with improved performance and energy efficiency. Multipliers are key arithmetic circuits in many such applications such as digital signal processing (DSP). In this paper, a novel multiplier with a lower power consumption and a shorter critical path than traditional multipliers are proposed for high-performance DSP applications. This multiplier leverages a newly-designed approximate adder that limits its carry propagation to the nearest neighbors for fast partial product accumulation. Different levels of accuracy can be achieved through a configurable error recovery by using different numbers of most significant bits (MSBs) for error reduction. The multiplier has a low mean error distance, i.e., most of the errors are not significant in magnitude. Compared to the Wallace multiplier, a 16-bit multiplier implemented in a 28nm CMOS process shows a reduction in delay and power of 20% and up to 69%, respectively. It is shown that by utilizing an appropriate error recovery, the proposed multiplier achieves similar processing accuracy as traditional exact multipliers but with significant improvements in power and performance.

G. Zervakis, et al., "**Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation**" Approximate computing has received significant attention as a promising strategy to decrease power consumption of inherently error tolerant applications. In this paper, we focus on hardware-level approximation by introducing the

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

partial product perforation technique for designing approximate multiplication circuits. We prove in a mathematically rigorous manner that in partial product perforation, the imposed errors are bounded and predictable, depending only on the input distribution. Through extensive experimental evaluation, we apply the partial product perforation method on different multiplier architectures and expose the optimal architecture-perforation configuration pairs for different error constraints. We show that, compared with the respective exact design, the partial product perforation delivers reductions of up to 50% in power consumption, 45% in area, and 35% in critical delay. In addition, the product perforation method is compared with the state-of-the-art approximation techniques, i.e., truncation, voltage over scaling, and logic approximation, showing that it outperforms them in terms of power dissipation and error.

T. Yang, T. Ukezono, and T. Sato "**A Low-Power High-Speed Accuracy-Controllable Multiplier Design**", Multiplication is a key fundamental function for many error-tolerant applications. Approximate multiplication is considered to be an efficient technique for trading off energy against performance and accuracy. This paper proposes an accuracy-controllable multiplier whose final product is generated by a carry-maskable adder. The proposed scheme can dynamically select the length of the carry propagation to satisfy the accuracy requirements flexibly. The partial product tree of the multiplier is approximated by the proposed tree compressor. An 8 × 8multiplier design is implemented by employing the carry maskable adder and the compressor. Compared with a conventional Wallace tree multiplier, the proposed multiplier reduced power consumption by between 47.3% and 56.2% and critical path delay by between 29.9% and 60.5%, depending on the required accuracy. Its silicon area was also 44.6% smaller. In addition, results from an image processing application demonstrate that the quality of the processed images can be controlled by the proposed multiplier design.

VEDIC MULTIPLIER USING KARASTUBA ALGORITHM

Vedic mathematics:

Vedic Mathematics is one of the most ancient methodologies used by the Aryans in order to perform mathematical calculations. This consists of algorithms that can boil down large arithmetic operations to simple mind calculations. The above said advantage stems from the fact that Vedic mathematics approach is totally different and considered very close to the way a human mind works. The efforts put by Jagadguru Swami Sri Bharati Krishna Tirtha Maharaja to introduce Vedic Mathematics to the commoners as well as streamline Vedic Algorithms into 16 categories or Sutras needs to be acknowledged and appreciated. The Vedic algorithm is one such multiplication algorithm which is well known for its efficiency in reducing the calculations involved.

With the advancement in the VLSI technology, there is an ever-increasing quench for portable and embedded Digital Signal Processing (DSP) systems. DSP is omnipresent in almost every engineering discipline. Faster additions and multiplications are the order of the day. Multiplication is the most basic and frequently used operations in a CPU. Multiplication is an operation of scaling one number by another. Multiplication operations also form the basis for other complex operations such as convolution, Discrete Fourier Transform, Fast Fourier Transforms, etc. With ever increasing need for faster clock frequency, it becomes imperative to have faster arithmetic unit. Therefore, DSP engineers are constantly looking for new algorithms and hardware to implement them. Vedic mathematics can be aptly employed here to perform multiplication.

RECURSIVE KARATSUBA MULTIPLICATION

Recursive Karatsuba is based on incorporating Karatsuba algorithm repeatedly at every stage to improve speed when bit size is high. The algorithm works on separating the bits (N) into groups of half-the-number-of-bits (N/2) and then following the same Karatsuba procedure with the segmented bits recursively. For a 16-bit multiplication, e.g., it breaks the number down to 8-bit multiplication, which is further divided into 4-bit and finally reduced to 2-bit which is the last stage for normal multiplication to be performed. At every stage we have implemented adaptive Karatsuba for the 3rd product term .

Standard Karatsuba Multiplier

Let X and Y are inputs of 'n' bits. Assuming decomposition of X and Y into 2 equal parts; XH, YH represent the higher order bits and XL, YL the lower order bits. Their product can be computed as:

$$XY = \left(2^{\frac{n}{2}}X_H + X_L\right)\left(2^{\frac{n}{2}}Y_H + Y_L\right)$$
$$= 2^n(X_HY_H) + 2^{\frac{n}{2}}(X_HY_L + X_LY_H) + (X_LY_L) \quad (1)$$

In Karatsuba algorithm the computation is rewritten as:

$$X_HY_L + X_LY_H = (X_H + X_L)(Y_H + Y_L) - X_HY_H - X_LY_L \quad (2)$$

So, 4×n/2-bit multiplications can be reduced to 3×n/2-bit multiplications: $(X_H + Y_H)(X_L + Y_L)$, X_HY_H and X_LY_L. Fig. 1 shows the standard Karatsuba multiplier at a stage when inputs are n-bits.

Time complexity of conventional multiplication algorithm requires:

$$O(n) = n^2 \quad (3)$$

whereas Karatsuba multiplication algorithm requires :

$$O(n) = n^{1.58} \quad (4)$$

where n is the number of bits and O is order of complexity, considering O(1) = 1. This shows

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

analytically that Karatsuba algorithm with a complexity of $n^{1.58}$ (due to logarithmic dependency of n) is faster than the standard multiplication due to the logarithmic power of n.

We modify the Karatsuba algorithm such that the computation of the third product is performed efficiently. Assuming inputs X and Y be of 'n' bits, we have the argument
of the third product of $(^n/2 + 1)$ bits. Let Z and U be these
arguments left padded with (n/2 -1) bits. ZH, UH represent the higher order bits and ZL, UL represent the lower order bits where ZH and UH can be either 0 or 1, because they are the carry-outs of the third product arguments $(X_H + Y_H)$ and $(X_L + Y_L)$. Thus

$$\text{Product}_3 = ZU$$
$$= \left(2^{\frac{n}{2}}Z_H + Z_L\right)\left(2^{\frac{n}{2}}U_H + U_L\right)$$
$$= 2^n(Z_HU_H) + 2^{\frac{n}{2}}(Z_HU_L + Z_LU_H) + (Z_LU_L) \quad (5)$$

Depending on ZH, UH the above expression can be evaluated as shown in Table I.

THIRD PRODUCT COMPUTATION

Z_H	U_H	Product_3
0	0	Z_LU_L
0	1	$2^{\frac{n}{2}}Z_L + Z_LU_L$
1	0	$2^{\frac{n}{2}}U_L + Z_LU_L$
1	1	$2^n + 2^{\frac{n}{2}}(U_L + Z_L) + Z_LU_L$

Table 1

It can be observed from Table I that third product-computation of (n/2 + 1) bits requires one multiplication of (n/2 -1) bits and additional shifting, adding and multiplexing operations, instead of a (n/2 + 1) bit multiplier at every stage. This makes Karatsuba implementation recursive, without additional hardware. Fig. 2 shows the adaptive concept for(n/2 + 1) bit computation at a stage when inputs are n bits and where the 'Shift and Add' block applies in appropriate cases as mentioned in Table I.

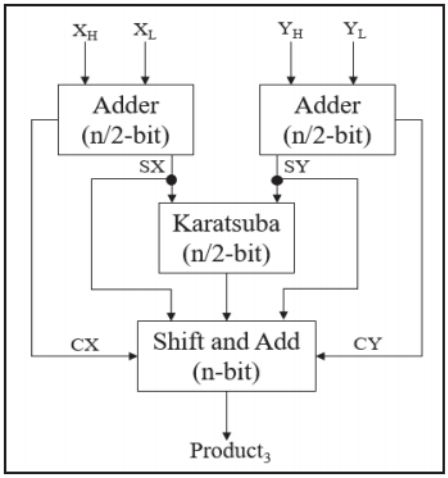


Fig. 2. Adaptive Concept for 3rd Product Computation

Conventional Wallace and Dadda multiplier use 3:2 compressors for carrying out addition of the generated partial products. The analyses depict that the proposed multiplier based on adaptive, recursive Karatsuba approach has lesser delay compared to the conventional Wallace and Dadda multiplier.

CARRY SELECT ADDERS

The performance of the Karatsuba multiplier has been further improved by the use of high-speed parallel adders at different stages. The optimizations that have been carried out are as follows:

- Additions involving 16 bit and above are carried out using the proposed fast adders.
- Final addition in the third-product-computation is carried out using carry save adders with the final stage involving the fast adders.

Carry Select Adder

In electronics, a **carry-select adder** is a particular way to implement an adder, which is a reduced logic delay element that computes the carry-select adder generally consists of ripple-carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple-carry adders), in order to perform the calculation twice, one time with the assumption of the carry-in being zero and the other assuming it will be one. After the two results are calculated, the correct sum, as well as the correct carry-out, is then selected with the multiplexer once the correct carry-in is known.

Ripple carry adder generates carry out bit by rippling effect of the incoming carry generating from the previous stage after receiving the carry-in (C_{in}) bit. Hence, the speed of RCA is less, as C_{out} of a stage is dependent on C_{out} of the previous stage or the C_{in} of the current stage. This linear dependency problem of C_{out} on C_{in} is overcome by assuming the possible values of C_{in} to be a '0' and a '1'. By using these possible values of C_{in}, the partial Sum (S) and C_{out} are generated in parallel. Then, a multiplexer is employed to choose the final sum and carry based on the actual carry input received. Though this feature helps in reducing the computation delay, the area efficiency suffers due the use of the redundant adder circuit. Thus, Carry Select Adder consists of two RCA blocks.in this project to reduce the optimal delay we proposed square root carry save adder.

Square Root Carry Select Adder

In Square Root Carry Select Adder (SRCSA), [9]-[10] the block size can be variable. The complete analysis is omitted here for brevity, but here, e.g., a 16-bit adder can be created using block sizes of 2-2-3-4-5 instead of using uniform block size of four (as done before) [8]. This break-up is ideal when the Full-Adder delay is equal to the MUX delay. Fig. 3 shows the block diagram of proposed SRCSA adder for 16 bits

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

where the inputs are A and B, Carry-in is denoted as C_{in} and outputs are denoted by sum (S) and Carry-out (C_{out}).

Vedic algorithms have been used to construct function circuits with high performance and a simple architecture. However, because many of these methods are built on decimal number systems, banalization frequently resulted in a trade-off between speed and simplicity in architecture due to conversion-hardware overhead. However, Vedic algorithms have recently reawakened interest, owing to innovative circuit realization — mainly multipliers. The current project is significant since these authors have updated a well-known algorithm (Karatsuba) to incorporate an adaptive aspect that allows recursive operations to reduce the order of complexity from square to logarithmic value of power of bit-length.

To demonstrate the concept, a 1616-bit multiplier has been conceived and designed with the primary goal of decreasing the delay so that it may be used in DSP, Image Processing, and computation-intensive ASIPs. It is based on the Vedic Karatsuba algorithm, which produces fewer partial product terms. To improve speed, the technique is further refined by employing an adaptive notion for the computation of the third product term. Furthermore, integrating the carry save adders with the proposed Square Root Carry Select Adder (SRCSA) adder, as presented in this paper, improves the compression speed of partial product terms.

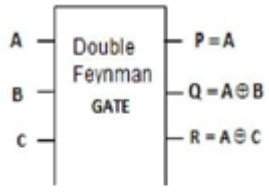
The proposed multiplier with high-speed compression provides a delay decrease of 18.19 percent compared to Dadda and 24.08 percent compared to Wallace Tree, according to this benchmarking exercise with other candidate designs. In addition, the latency is less than when the suggested design employs CSA (with other parameters being comparable). As a result, we believe it is an excellent choice for fast multiplier blocks in ASIP and general-purpose processors.

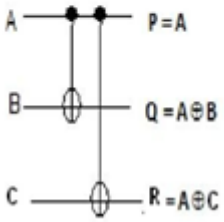
REVERSIBLE GATES

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. According to Landauer's research, the amount of energy dissipated for every irreversible bit operation is at least KTln2 joules, where $K=1.3806505*10^{-23}m^2kg^{-2}K^{-1}$ (joule/Kelvin^{-1}) is the Boltzmann's constant and T is the temperature at which operation is performed . The heat generated due to the loss of one bit of information is very small at room temperature but when the number of bits is more as in the case of high speed computational works the heat dissipated by them will be so large that it affects the performance and results in the reduction of lifetime of the components In 1973, Bennett showed that KTln2 energy would not dissipate from a system as long as the system allows the reproduction of the inputs from observed outputs . Reversible logic supports the process of running the system both forward and backward. This means that reversible computations can generate inputs from outputs and can stop and go back to any point in the computation history. A circuit is said to be reversible if the input vector can be uniquely recovered from the output vector and there is a one-to-one correspondence between its input and output assignments, i.e. not only the outputs can be uniquely determined from the inputs, but also the inputs can be recovered from the outputs Energy dissipation can be reduced or even eliminated if computation becomes Information-lossless

REVERSIBLE LOGIC GATES

A reversible logic gate is an n-input n-output logic device with one-to-one mapping. This helps to determine the outputs from the inputs and also the inputs can be uniquely recovered from the outputs. Also, in the synthesis of reversible circuits direct fan-Out is not allowed as one–to-many concept is not reversible. However, fan-out in reversible circuits is achieved using additional gates. A reversible circuit should be designed using minimum number of reversible logic gates. From the point of view of reversible circuit design, there are many parameters for determining the complexity and performance of circuits.





- The number of Reversible gates (N): The number ofreversible gates used in circuit.
- The number of constant inputs (CI): This refers to the number of inputs that are to be maintained constant at either 0 or 1 in order to synthesize the given logical function.
- The number of garbage outputs (GO): This refers to the number of unused outputs present in a reversible logic

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

- circuit. One cannot avoid the garbage outputs as these arevery essential to achieve reversibility.
- Quantum cost (QC): This refers to the cost of the circuit in terms of the cost of a primitive gate. It is calculated knowing the number of primitive reversible logic gates (1*1 or 2*2) required to realize the circuit.

BASIC REVERSIBLE LOGIC GATES

Feynman Gate

Feynman gate is a 2*2 one through reversible gate as shownin figure 1. The input vector is I(A, B) and the output vector isO(P, Q). The outputs are defined by P=A, Q=A xor B. Quantum cost of a Feynman gate is 1. Feynman Gate (FG)can be used as a copying gate. Since a fan-out is not allowedin reversible logic, this gate is useful for duplication of the required outputs.

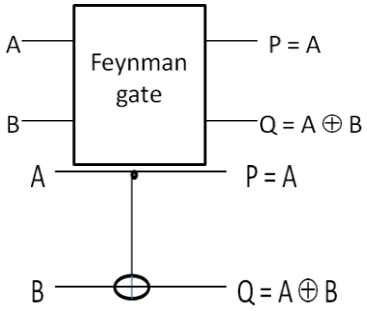


Figure 1: Feynman Gate

A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Table 1: Truth table of Feynman gates

Double Feynman Gate (F2G)

Fig.2 shows a 3*3 Double Feynman gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The outputs are defined by P = A, Q=A xor B, R=A xor C. Quantum cost of double Feynman gate is 2.

Fig 2: Double Feynman gate

Toffoli Gate:

Fig 3 shows a 3*3 Toffoli gate. The input vector is I (A, B, C) and the output vector is O(P,Q,R). The outputs are defined by P=A, Q=B, R=AB xor C. Quantum cost of a Toffoligate is 5.

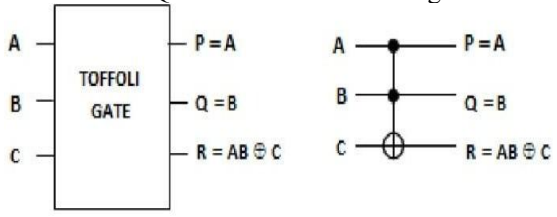


Fig 3: Toffoli gate

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	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Table 3: Truth table of Toffoli gate

Fredkin Gate

Fig 4 shows a 3*3 Fredkin gate. The input vector isI (A, B, C) and the output vector is O (P, Q, R). The output is defined by P=A, Q=A′B xor AC and R=A′Cxor AB. Quantum cost of a Fredkin gate is 5.

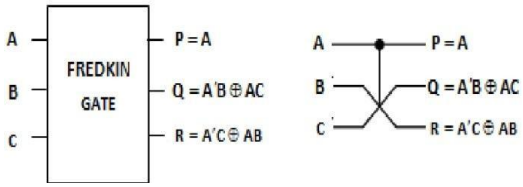


Fig 4(a): Fredkin gate

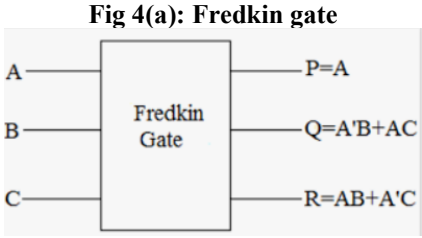


Fig 4(b): fault tolerant Fredkin gate

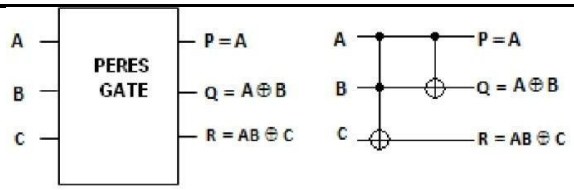
A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Table 4: Truth table of Fredkin gate

Peres Gate

Fig 5 shows a 3*3 Peres gate. 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 = Axor B and R=AB xor C. Quantum cost of a Peres gate is 4. In the proposed design Peres gate is used because of its lowest quantum cost.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper



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	0	1	0	1
1	1	1	1	0	0

Table 5: Truth table of Peres gate

TSG gate

Fig 7 shows a 4*4 TSG gate. The input vector is I (A, B, C, D) and the output vector is O (P, Q, R, S). The output is defined by P = A, Q = A'C' xor B', R = (A'C' xor B') xor D and S = (A'C' xor B').D xor (AB xor C) Quantum cost of a Peres gate is 4. In the proposed design Peres gate is used because of its lowest quantum cost. It can be verified that the input pattern corresponding to a particular output pattern can be uniquely determined. The proposed TSG gate is capable of implementing all Boolean functions and can also work singly as a reversible Full adder

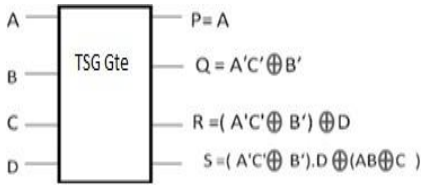


Fig 7: TSG gate

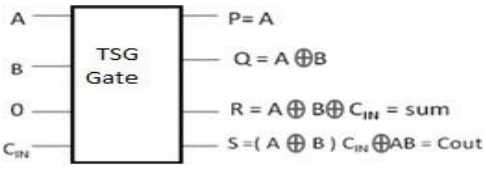


Fig 8: TSG Gate Working as Reversible Full Adder

Sayem gate

SG is a 1 trough 4x4 reversible gate. The input andoutput vector of this gate are, Iv = (A, B, C, D) and Ov = (A,A'B xor AC, A'B xor AC xor D, AB xor A'C xor D). The block diagram of this gate is shown in Fig 9

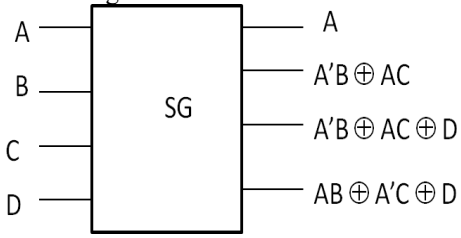


Fig 9: Sayem gate

RESULTS

RTL SCHEMATIC:

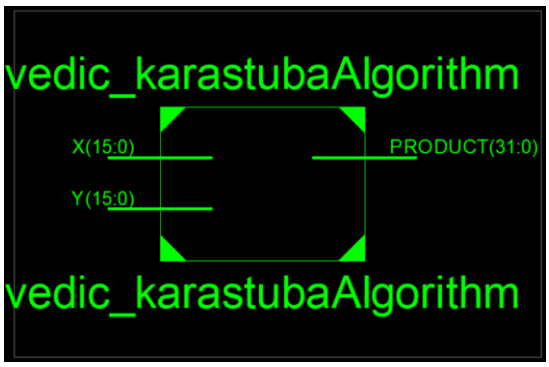


Fig1: RTL Schematic of existed Vedic multiplier

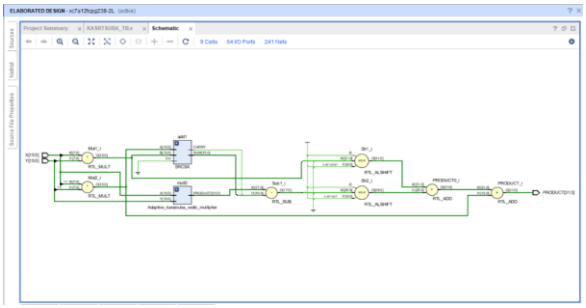
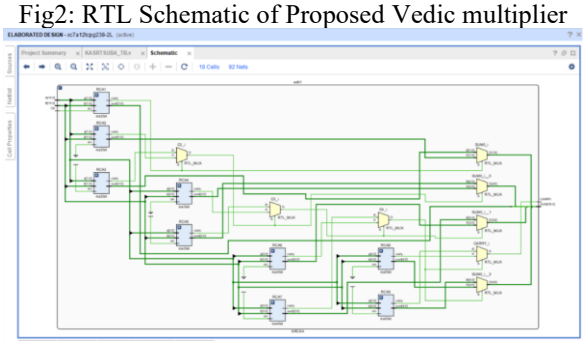


Fig2: RTL Schematic of Proposed Vedic multiplier



F: RTL adder

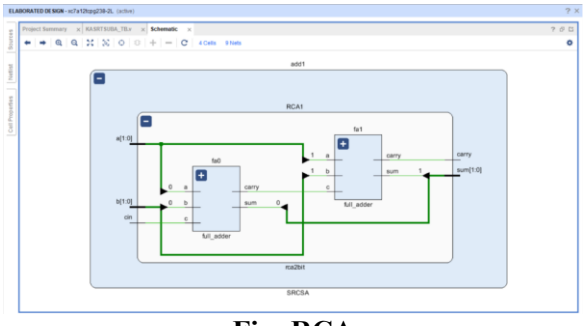


Fig: RCA

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

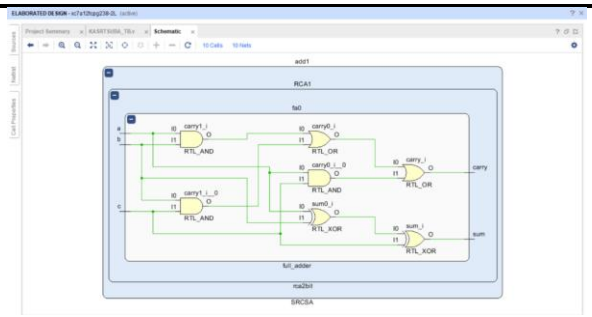


Fig: full adder

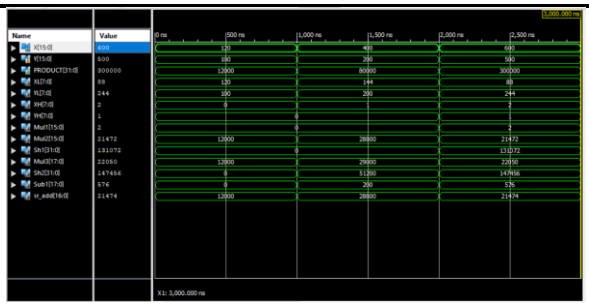


Fig :Simulated Waveforms of existed Vedic multiplier

TECHNOLOGY SCHEMATIC

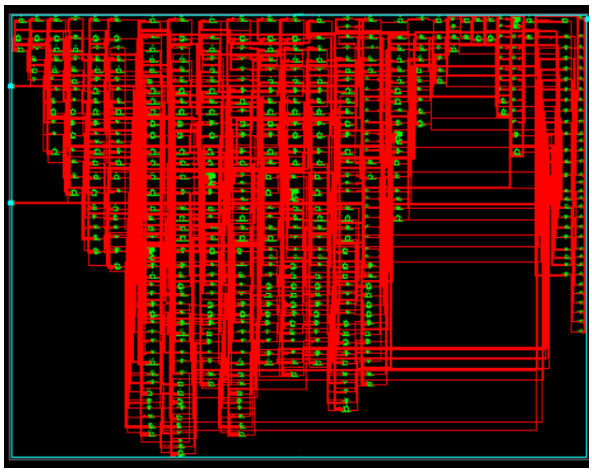


Fig :View Technology Schematic of existed Vedic multiplier

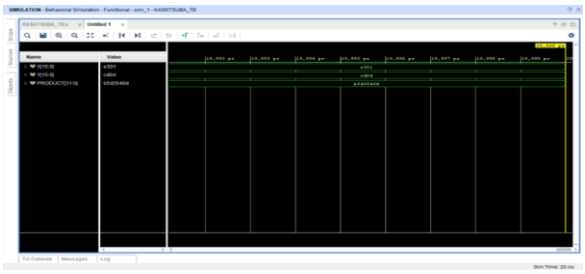


Fig :Simulated Waveforms of proposed Vedic multiplier

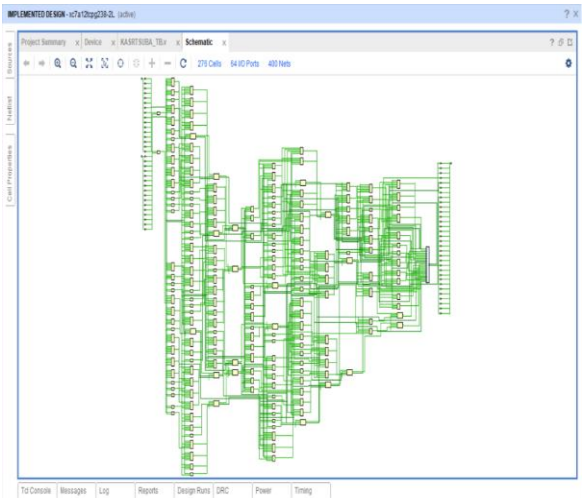


Fig: View Technology Schematic of proposed Vedic multiplier

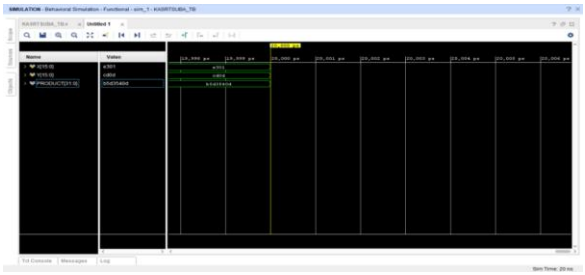


Fig :Simulated Waveforms of proposed Vedic multiplier

PARAMETERS: -

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	378	1,408	26%	
Number of occupied Slices	206	704	29%	
Number of Slices containing only related logic	206	206	100%	
Number of Slices containing unrelated logic	0	206	0%	
Total Number of 4 input LUTs	397	1,408	28%	
Number used as logic	378			
Number used as a route-thru	19			
Number of bonded IOBs	64	108	59%	
Average Fanout of Non-Clock Nets	2.31			

Table: Device Utilization Summery of Existed Design

SIMULATION: -

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Utilization Post-Synthesis | Post-Implementation

Graph | Table

Resource	Estimation	Available	Utilization %
LUT	348	8000	4.35
IO	64	112	57.14

Table :Device Utilization Summery of Proposed Design

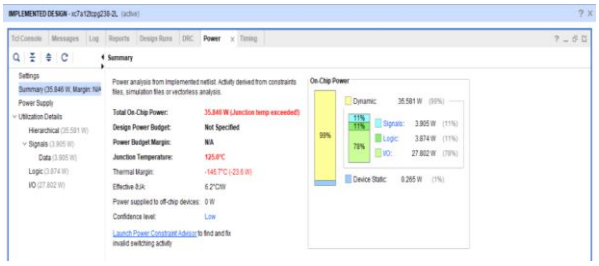


Table : Power Analysis

Parameter	Karatsuba Multiplier	Karatsuba Multiplier Using Reversible Logic
Power (Mwatt)	2.819	1.661
Number Of 4 Input LUT's	397	234

Table : Parameter Comparison Observed In Xilinx

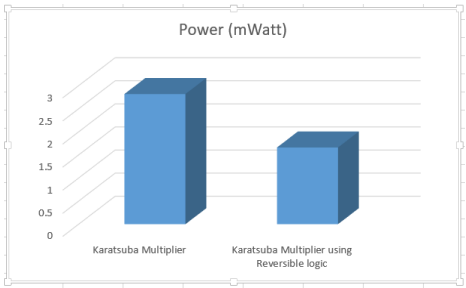


Fig : Power Comparison Bar Graph

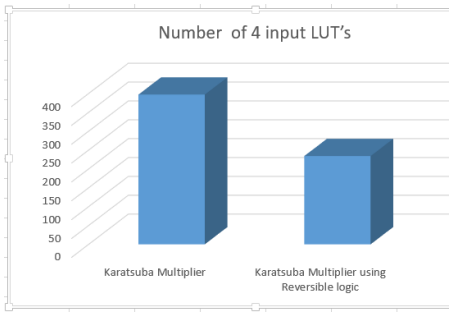


Fig : LUT Comparison Bar Graph

CONCLUSION AND FUTURE SCOPE

Vedic algorithms have been useful in designing function circuits for achieving high speed and simplified architecture. However, the challenge had been that all these algorithms are based on decimal number systems and as such binarization often led to trade-off of the speed-advantage and simplified-architecture due to conversion-hardware-overhead. Yet renewed interest has been observed recently in Vedic algorithms particularly due to clever circuit realization – primarily multipliers. The present endeavour assumes importance in that context where a known algorithm (Karatsuba) has been modified by these authors to include adaptive aspect allowing recursive operation to reduce the order of complexity from square to logarithmic value of power of bit-length.

A 16×16-bit multiplier using reversible logic has been proposed and designed to showcase the technique with the primary objective of minimizing the delay so that it can find application in DSP, Image Processing and computation intensive ASIPs. It is based on the Vedic Karatsuba algorithm that generates lesser number of partial product terms. The algorithm is further optimized using adaptive concept for computation of the third product term to yield faster speed. Moreover, the compression speed of the partial product terms is also enhanced by combining the carry save adders with the proposed Square Root Carry Select Adder (SRCSA) adder with reversible logic as discussed in this

The implementation, synthesis and simulation are performed in XILINX-ISE tool in Verilog HDL language. In future the implementation of this multiplier employed which eliminates gate delays and adding the approximation to the architecture can enhance the performance in dsp applications, image processing, filters and cryptographic applications. Area and speed-based applications, it will be used in future.

REFERENCES

[1] Swami Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics", Motilal Banarsidass Publishers, 1965.

[2] Rakshith T R and Rakshith Saligram, "Design of High-Speed Low Power Multiplier using Reversible logic: a Vedic Mathematical Approach", *International Conference on Circuits, Power and Computing Technologies* (ICCPCT-2013), ISBN: 978-1-4673-4922-2/13, pp.775-781.

[3] M.E. Paramasivam and Dr. R.S. Sabeenian, "An Efficient Bit Reduction Binary Multiplication Algorithm using Vedic Methods", *IEEE 2nd International Advance Computing Conference, 2010,* ISBN: 978-1-4244-4791-6/10, pp. 25-28.

[4] Sushma R. Huddar, Sudhir Rao Rupanagudi, Kalpana M and Surabhi Mohan, "Novel High Speed Vedic Mathematics Multiplier using Compressors", *International Multi conference on Automation, Computing, Communication, Control and Compressed Sensing*(iMac4s), *22-23 March 2013, Kottayam,* ISBN: 978-1-4673-5090-7/13, pp.465-469.

[5] L. Sriraman and T. N. Prabakar, "Design and Implementation of Two Variables Multiplier Using

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

KCM and Vedic Mathematics", *1st International Conference on Recent Advances in Information Technology* (RAIT -2012), ISBN: 978-1-4577-0697-4/12.

[6] Prabir Saha, Arindam Banerjee, Partha Bhattacharyya and Anup Dandapat, "High Speed ASIC Design of Complex Multiplier Using Vedic Mathematics", *Proceeding of the 2011 IEEE Students' Technology Symposium 14-16 January,2011, IIT Kharagpur*, ISBN: 978-1-4244-8943-5/11, pp.237-241.

[7] Soma BhanuTej, "Vedic Algorithms to develop green chips for future", *International Journal of Systems, Algorithms & Applications*, Volume 2, Issue ICAEM12, February 2012, ISSN Online: 2277-2677.

[8] Gaurav Sharma, Arjun Singh Chauhan, Himanshu Joshi and Satish Kumar Alaria, "Delay Comparison of 4 by 4 Vedic Multiplier based on Different Adder Architectures using VHDL", *International Journal of IT, Engineering and Applied Sciences Research (IJIEASR)*, ISSN: 2319-4413, Volume 2, No. 6, June 2013, pp. 28-32.

[9] Aniruddha Kanhe, Shishir Kumar Dasand Ankit Kumar Singh, "Design and Implementation of Low Power Multiplier using Vedic Multiplication Technique", *International Journal of Computer Science and Communication*, Vol. 3, No. 1, June 2012, pp. 131-132.

[10] Anju and V.K. Agrawal, "FPGA Implementation of Low Power and High-Speed Vedic Multiplier using Vedic Mathematics", *IOSR Journal of VLSI and Signal Processing* (IOSR-JVSP) Volume 2, Issue 5 Jun. 2013, ISSN: 2319 – 4200, pp. 51-57.

[11] Animul islam, M.W. Akram, S.D. pable ,Mohd. Hasan, "Design and Analysis of Robust Dual Threshold CMOS Full Adder Circuit in 32 nm Technology", International Conference on Advances in Recent Technologies in Communication and Computing,2010.

[12] Deepa Sinha, Tripti Sharma, k. G. Sharma, Prof.B.P.Singh, "Design and Analysis of low Power 1-bit Full Adder Cell", IEEE, 2011.

[13] Nabihah Ahmad, Rezaul Hasan, "A new Design of XOR-XNOR gates for Low Power application", International Conference on Electronic Devices, Systems and Applications(ICEDSA) ,2011.

[14] R. Uma, "4-Bit Fast Adder Design: Topology and Layout with Self-Resetting Logic for Low Power1VLSI Circuits", International Journal of Advanced Engineering Sciences and Technology, Vol No. 7,1Issue No. 2, 197 – 205.

[15] David J. Willingham and izzet Kale, "A Ternary Adiabatic Logic (TAL) Implementation of a Four-Trit Full-Adder, IEEE, 2011.

[16] Padma Devi, Ashima Girdher and Balwinder Singh, "Improved Carry Select Adder with Reduced1Area and Low Power Consumption", International Journal of Computer Application, Vol 3.No.4, June1 2010 .

[17] B. Ramkumar, Harish M Kittur, P. Mahesh Kannan, "ASIC Implementation of Modified Faster Carry Save Adder", European Journal of Scientific Research ISSN 1450-216X Vol.42 No.1, pp.53-58,2010.

[18] Y. Sunil Gavaskar Reddy and V. V. G. S. Rajendra Prasad, "Power Comparison of CMOS and Adiabatic Full Adder Circuits", International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.3, September 2011

[19] Mariano Aguirre-Hernandez and Monico Linares-Aranda, "CMOS Full-Adders for Energy-Efficient Arithmetic Applications", IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, Vol.19, No. 4, April 2011.

[20] Ning Zhu, Wang Ling Goh, Weija Zhang, Kiat Seng Yeo, and Zhi Hui Kong, "Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing", IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, Vol. 18, No. 8, August 2010.

[21] Sreehari Veeramachaneni, M.B. Srinivas, "New Improved 1-Bit Full Adder Cells", IEEE, 2008.International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1, February 2012164

[22] Tripti Sharma, k.G.Sharma, Prof.B.P.Singh, Neha Arora, "High Speed, Low Power 8T Full Adder Cell with 45% Improvement in Threshold Loss Problem", Recent Advances in Networking, VLSI and Signal Processing.

[23] G. Shyam Kishore, "A Novel Full Adder with High-Speed Low Area", 2nd National Conference on Information and Communication Technology (NCICT) 2011 Proceedings published in International Journal of Computer Applications® (IJCA).

[24] Shubhajit Roy Chowdhury, Aritra Banerjee, Aniruddha Roy, Hiranmay Saha, "A high Speed 8 Transistor Full Adder Design using Novel 3 Transistor XOR Gates", International Journal of Electrical and Computer Engineering 3:12 2008.

[25] Romana Yousuf and Najeebuddin, "Synthesis of Carry Select Adder in 65 nm FPGA", IEEE.

[26] Shubin. V. V, "Analysis and Comparison of Ripple Carry Full Adders by Speed", Micro/Nano Technologies and Electron Devices(EDM),2010, International Conference and Seminar on, pp.132-135,2010.

[27] Pudi. V, Sridhara., K, "Low Complexity Design of Ripple Carry and Brent Kung Adders in QCA", Nano technology, IEEE transactions on,Vol.11,Issue.1,pp.105-119,2012.

[28] Jian-Fei Jiang; Zhi-Gang Mao; Wei-Feng He; Qin Wang, "A New Full Adder Design for Tree Structured Arithmetic Circuits", Computer Engineering and

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Technology(ICCET),2010,2nd International Conference on,Vol.4,pp.V4-246-V4- 249,2010.