ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2,Special Issue 6, March 2016

POWER OPTIMIZED MULTIPLIER DESIGN USING VEDIC ALGORITHM

Mary Grace Neela.M [#1],Kishore Kumar.K [*2]
*[#1]Assistant Professor, [*2]Student*
Department of Electronics and Communication Engineering
TJS Engineering College
Peruvoyal
[1]graceneela@gmail.com
[2] kishorekumar6305@gmail.com

***Abstract*—.**This work is developed to the design and FPGA implementation of a 16bit Arithmetic module, which uses Vedic Mathematics algorithms.For arithmetic multiplication various Vedic multiplication techniques like UrdhvaTiryakbhyam, Nikhilam and Anurupye has been thoroughly analyzed. Also the Karatsuba algorithm for multiplication has been discussed. It has been found that UrdhvaTiryakbhyam Sutra is most efficient Sutra (Algorithm), giving minimum delay for multiplication of all types of numbers.Using UrdhvaTiryakbhyam, a 16x16 bit Multiplier has been designed and using this Multiplier, a Multiply Accumulate (MAC) unit has been designed. Then, an Arithmetic module has been designed which employs these Vedic multiplier and MAC units for its operation. Logic verification of these modules has been done by using Modelsim 6.5. Further, the whole design of Arithmetic module has been released on Xilinx Spartan 3E FPGA kit and the output has been displayed on the LCD of the kit. The synthesis results show that the computation time for calculating the product of 16x16 bits is 10.148 ns, while for the MAC operation is 11.151 ns. The maximum combinational delay for the Arithmetic module is 15.749 ns.

***Keywords*— MAC, FPGA, LCD**

I.INTRODUCTION

Arithmetic is the oldest and most elementary branch of Mathematics. The name Arithmetic comes from the Greek word (arithmos). Arithmetic is used by almost everyone, for tasks ranging from simple day to day work like counting to advanced science and business calculations. As a result, the need for a faster and efficient Arithmetic Unit in computers has been a topic of interest over decades. The work presented in this thesis, makes use of Vedic Mathematics and goes step by step, by first designing a Vedic Multiplier, then a Multiply Accumulate Unit and then finally an Arithmetic module which uses this multiplier and MAC unit. The four multiplication based operations are some of the frequently used Functions, currently implemented in many Digital Signal Processing (DSP) applications such as Convolution, Fast Fourier Transform, filtering and in Arithmetic Logic Unit (ALU) of Microprocessors. Since multiplication is such a frequently used operation, it's necessary for a multiplier to be fast and power efficient and so, development of a fast and low power multiplier have been a subject of interest over decades.basic operations in elementary arithmetic are addition, subtraction, multiplication and division. Multiplication, basically, is the mathematical operation of scaling one number by another. Talking about today's engineering world,

Multiply Accumulate or MAC operation is also a commonly used operation in various Digital Signal Processing Applications. Now, not only Digital Signal Processors, but also general-purpose Microprocessors come with a dedicated Multiply Accumulate Unit or MAC unit. When talking about the MAC unit, the role of Multiplier is very significant because it lies in the data path of the MAC unit and its operation must be fast and efficient. A MAC unit consists of a multiplier implemented in combinational logic, along with a fast adder and accumulator register, which stores the result on the clock

The basic concept of multiplication, a historical and a simple algorithm for multiplication, to motivate creativity and innovation has been discussed first of all, then the focus has been brought to Vedic Mathematics Algorithms and their functionality. Then, Karatsuba-Ofman Algorithm and finally MAC unit architecture along with Arithmetic module architecture has been discussed

RELATED WORKS

Due to Modified PartialProduct Generator(Xiaoping Cui) for Redundant Binary Multipliers high modularity and carry-free addition, a redundant binary (RB) representation can be used when designing high performance multipliers. The conventional RB multiplier requires an additional RB partial product (RBPP) row, because an error-correcting word (ECW) is generated by both the radix-4Modified Booth encoding (MBE) and the RB

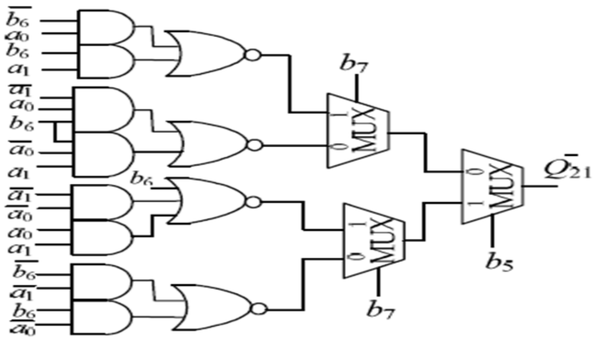
ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2,Special Issue 6, March 2016

encoding. This incurs an additional RBPP accumulation stage for the MBE multiplier. In this paper, a new RB modified partial product generator (RBMPPG) is proposed; it removes the extra ECW and hence, it saves one RBPP accumulation stage. Therefore, the proposed RBMPPG generates fewer partial product rows than a conventional RB MBE multiplier. Simulation results show that the proposed RBMPPG based designs significantly improve the area and power consumption when the word length of each operand in the multiplier is at least 32 bits, and next research tell about the an efficient reverse converter for transforming the redundant binary (RB) representation into two's complement form. The hierarchical expansion of the carry equation for the reverse conversion algorithm creates a regular multilevel structure, from which a high-speed hybrid carry- lookahead/carry-select (CLA/CSL) architecture is proposed to fully exploit the redundancy of RB encoding for VLSI efficient implementation. The optimally designed CSL sections interleaved evenly in the mixed-radix CLA network to boost the performance of the reverse converter well above those designed based on a homogeneous type of carry propagation adder. The logical effort characterization captures the effect of circuit's fan-in, fan-out and transistor sizing on performance, and the evaluation shows that our proposed architecture leads to the fastest design. A 64-bit transistor-level circuit implementation of our proposed reverse converter and that of its most competitive contender were simulated to validate the logical effort delay model. The pre- and post-layout HSPICE simulation results reveal that our new converter expends at least two times less energy (power-delay product) than the competitor circuit and is capable of completing a 64-bit conversion in 829 ps and dissipates merely 5.84 mW at a data rate of 1 GHz and a supply voltage of 1.8 V in TSMC 0.18-mum CMOS technology.

II.EXISTING SYSTEM

In the existing system Booth multiplier is used as a Arithmatic unit and Multiplier replace for efficient representation its Negative operands used as 2's complement,Error correcting , and Modified booth encoding and the normal binary to redundant binary to make simple to avoid hard multiples.this method is number of gates are reduced and partial products are reduced up to 32 for 128 bit processor.



$$X + Y = X - \overline{Y} - 1$$
$$= \left(-x_N 2^N + \sum_{i=0}^{N-1} x_i 2^i\right) - \left(-\overline{y_N} 2^N + \sum_{i=0}^{N-1} \overline{y_i} 2^i\right) - 1$$
$$= -(x_N - \overline{y_N}) 2^N + \sum_{i=0}^{N-1} (x_i - \overline{y_i})\, 2^i - 1$$
$$= (X, \overline{Y}) - 1 \qquad (2)$$

III PROPOSED MODEL

VEDIC MATHEMATICS

VEDIC MATHEMATICS is a mathematical elaboration of 'Sixteen Simple Mathematical formulae from theVedas' as brought out by Sri Bharati Krishna Tirthaji. In the text authored by the Swamiji, nowhere has the list of the Mathematical formulae (Sutras) been given. But the Editor of the text has compiled the list of the formulae from stray references in the text. The list so compiled contains Sixteen Sutras and Thirteen Sub - Sutras as stated hereunder.

Urdhva-tiryagbhyam– Vertically and crosswise.

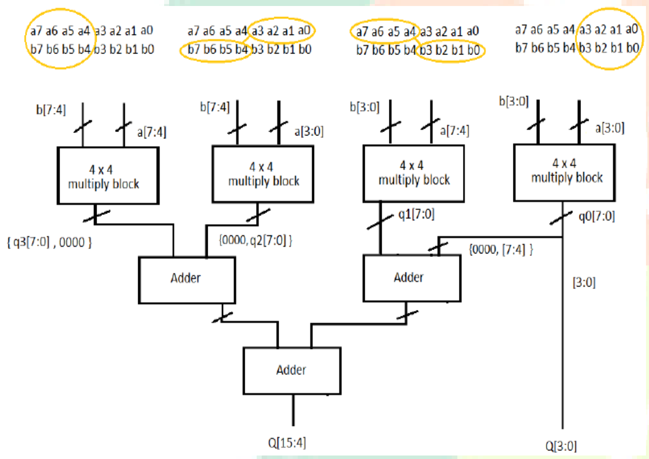
The early Indian mathematicians of the Indus Valley Civilization used a variety of intuitive tricks to perform multiplication. Most calculations were performed on small slate hand tablets, using chalk tables. One technique was of lattice multiplication. Here a table was drawn up with the rows and columns labeled by the multiplicands. Each box of the table is divided diagonally into two, as a triangular lattice. The entries of the table held the partial products, written as decimal numbers. The product could then be formed by summing down the diagonals of the lattice

VEDIC MULTIPLIER

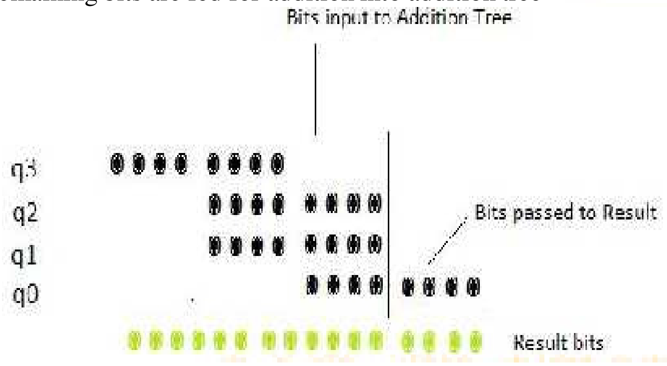
The design starts first with Multiplier design, that is 2x2 bit multiplier. Here, “UrdhvaTiryakbhyam Sutra” or “Vertically and Crosswise Algorithm” for multiplication has been effectively used to develop a digital multiplier architecture. This algorithm is quite different from the traditional method of multiplication, that is to add and shift the partial products. This Sutra shows how to handle multiplication of a larger number (N x N, of N bits each) by breaking it into smaller numbers of size (N/2 = n, say) and these smaller numbers can again be broken into smaller numbers (n/2 each) till we reach the multiplicand size of (2 x 2). Thus, simplifying the whole multiplication process.

8*8 bit Multiplier

The 8x8 Multiplier is made by using 4 , 4x4 multiplier blocks. Here, the multiplicands are of bit size (n=8) whereas the result is of 16 bit size. The input is broken into smaller chunk size of n/2 = 4, for both inputs, that is a and b, just like as in the case of 4x4 multiply block. These newly formed chunks of 4 bits are given as input to 4x4 multiplier block, where again, these new chunks are broken into even smaller chunks of size n/4 = 2 and fed to 2x2 multiply block. The result produced, from output of 4x4 bit multiply block which is about 8 bits, is sent in addition to an addition tree,



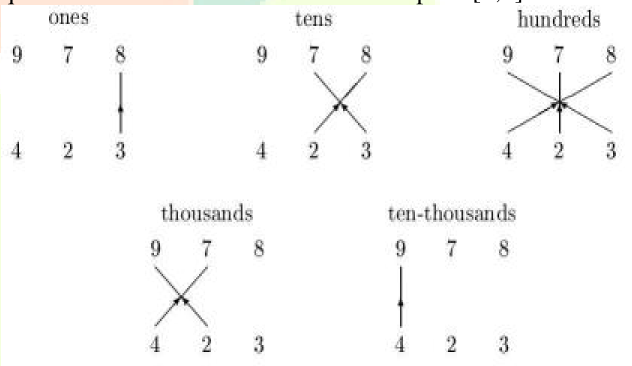
Here, one fact must be kept in mind that, each 4x4 multiply block works as illustrated in Fig 3.3. In 8x8 Multiply block, lower 4 bits of q0 are passed directly to output and the remaining bits are fed for addition into addition tree



Addition of Partial products in 8x8 block

URDHVATIRYAKBHYAM SUTRA

The multiplier is based on an algorithm UrdhvaTiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. UrdhvaTiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and crosswise". It is based on a novel concept through which the generation of all partial products can be done and then, the concurrent addition of these partial products can be done. This parallelism in generation of partial products and their summation is obtained using UrdhavaTiryakbhyam. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. By adopting the Vedic multiplier, microprocessor designers can easily circumvent these problems to avoid catastrophic device failures. The processing power of the multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular structure, it can be easily layout in a silicon chip. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore, it is time, space and power efficient. It is demonstrated that this architecture is quite efficient in terms of silicon area/speed [3,5].



SQUARE ALGORITHM

In order to calculate the square of a number, we have utilized "Duplex" D property of UrdhvaTriyakbhyam. In the Duplex, we take twice the product of the outermost pair and then add twice the product of the next outermost pair

D= 2 * (9 * 1) + 2 * (8 * 2) + 2 * (7 * 3) + 2 * (6 * 4) + 5 * 5 = 165. Further, the Duplex can be explained as follows

For a 1 bit number D is its square.

For a 2 bit number D is twice their product

PARALLEL COMPUTATION

1. D = X 0 * X0 = A
2. D = 2 * X1 * X0 = B
3. D = 2 * X2 * X0 + X1 * X1 = C
4. D = 2 * X3 * X0 + 2 * X2 * X1 = D

5 . D = 2 * X3 * X1 + X 2 * X2 = E
6. D = 2 * X3 * X2 = F
7. D = X3 * X3 = G

CUBE ALGORITHM

The cube of the number is based on the Anurupye Sutra of Vedic Mathematics which states "If you start with the cube of the first digit and take the next three numbers (in the top row) in a Geometrical Proportion (in the ratio of the original digits themselves) then you will find that the 4th figure (on the right end) is just the cube of the second digit

If a and b are two digits, then, according to Anurupye Sutra, a3 a2b ab2 b3

2a2b 2ab2

$a^3 + 3a^2b + 3ab^2 + b^3 = (a + b)^3$

KARATSUBA MULTIPLICATION

The Karatsuba Multiplication method offers a way to perform multiplication of large number of bits in fewer operations than the usual brute force technique of long multiplication. As discovered by Karatsuba and Ofman in 1962, multiplication of two n digit numbers can be done with a bit complexity of less than n^2 using the identities of the form

$$(a + b.10^n) (c + d.10^n) = a.c + [(a + b) (c + d) – a.c - b.d] 10^n + b.d.10^{2n}$$

By proceeding recursively the big complexity $O(n^{lg\ 3})$, where $lg^3 = 1.58$

Now let us take an example and consider multiplication of two numbers with just two digits (a1 a0) and (b1 b0) in base w

$$N1 = a0 + a1.w$$

$$N2 = b0 + b1w$$

Their product is:

$$P = N1.N2$$

$$= a0.b0 + (a0.b1 + a1.b0) w + a1.b1.w^2$$

$$= p0 + p1.w + p2.w^2$$

Instead of evaluating the individual digits, we can write

. q0 = a0. b0

. q1 = (a0 + a1) (b0 + b1)

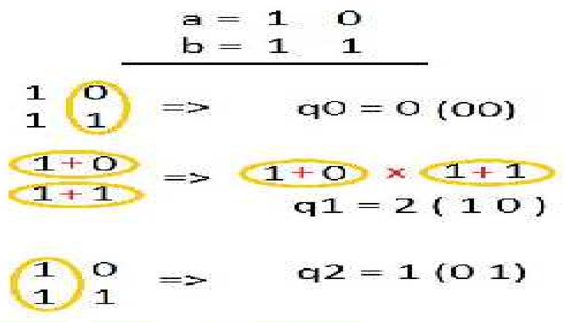
. q2 = a1. b1

Here q1 can be expanded, regrouped and can be written in the form of pus

. q1= p1+p0 +p2

Since, q0= p0,
p1= q1 – q0 – q2 p2=
q2

So the three digits of p have been evaluated using three multiplications rather than four. This technique can be used for binary numbers as well, with a tradeoff that more additions and subtractions are required.[8,9]

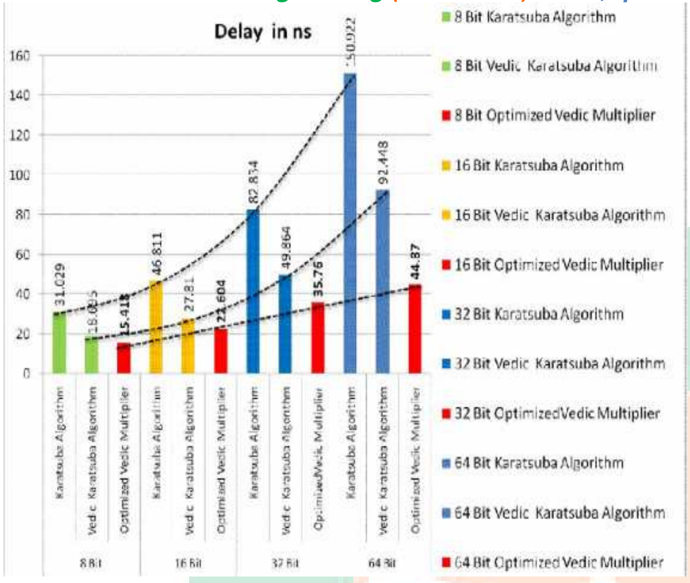


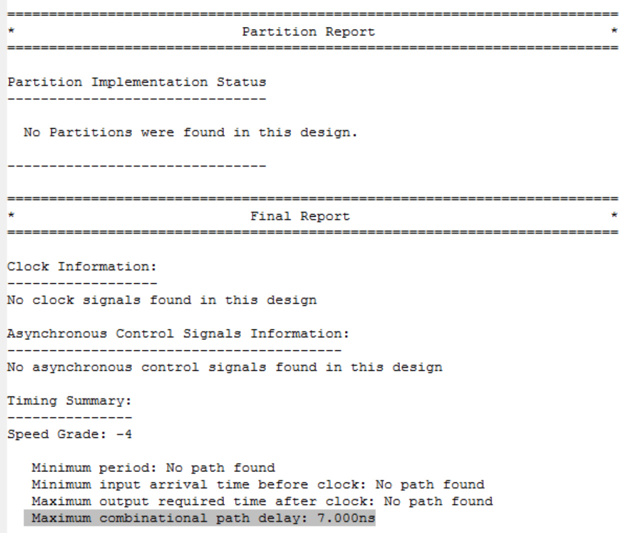
IV PERFORMANCE

Vedic multiplier is faster than array multiplier and Booth multiplier. As the number of bits increases from 8x8 bits to 16x16 bits, the time delay is greatly reduced for Vedic multiplier as compared to other multipliers. Vedic multiplier has the greatest advantage as compared to other multipliers over gate delays and the regularity of structures. Delay in Vedic multiplier for 16 x 16 bit number is 32 NS while the delay in Booth and Array multiplier are 37 NS and 43 NS respectively [10]. Thus, this multiplier shows the highest speed among conventional multipliers. It has this advantage than others to prefer a better multiplier

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2,Special Issue 6, March 2016

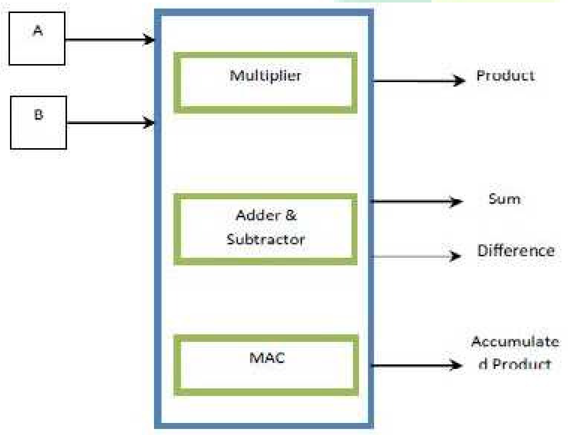



```
===========================================================================
*                          Partition Report                               *
===========================================================================

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

-------------------------------

===========================================================================
*                            Final Report                                 *
===========================================================================

Clock Information:
------------------
No clock signals found in this design

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -4

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 7.000ns
```

Executed Maximum combinational delay is 7.000ns

ARITHMETIC MODULE

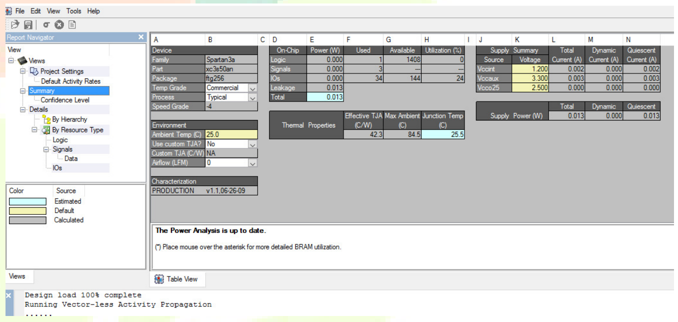
Arithmetic Logic Unit can be considered to be the heart of a CPU, as it handles all the mathematical and logical calculations that are needed to be carried out. Again, there may be different modules for handling Arithmetic and Logic functions. In this work, an arithmetic unit has been made using Vedic Mathematics algorithms and performs Multiplication, MAC operation as well as addition and subtraction. For performing addition and subtraction, conventional adder and subtractor have been used. The control signals which tell the arithmetic module, when to perform which operations are provided by the control unit, which is out of the scope of this thesis



V RESULT & DISCUSSION

OUTPUT DELAY

OUTPUT POWER



A	B
Device	
Family	Spartan3a
Part	xc3s50an
Package	ftg256
Temp Grade	Commercial
Process	Typical
Speed Grade	-4
Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0
Characterization	
PRODUCTION	v1.1,06-26-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	1	1408	0
Signals	0.000	3	---	---
IOs	0.000	34	144	24
Leakage	0.013			
Total	0.013			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	42.3	84.5	25.5

Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.002	0.000	0.002
Vccaux	3.300	0.003	0.000	0.003
Vcco25	2.500	0.000	0.000	0.000

	Total	Dynamic	Quiescent
Supply Power (W)	0.013	0.000	0.013

Color	Source
🟦	Estimated
🟨	Default
⬜	Calculated

The Power Analysis is up to date.

(*) Place mouse over the asterisk for more detailed BRAM utilization.

```
Design load 100% complete
Running Vector-less Activity Propagation
......
```

CONCLUSION

This project presented a high speed multiplier using Urdhvatiryakbhayam algorithm for multiplication based on Vedicmathematics and the partial product addition is done by the fault tolerant carry look ahead adder which offered fastcomputational speed. While architecture of multiplier is implemented using fault tolerant gate. The Proposed4x4 Fault tolerant Reversible Vedic Multiplier have minimum path delay and fault tolerant capability when compared toother multipliers.

FEATURE SCOPE

In future, adaptive LMS filter can be designed using fault tolerant reversible Vedic multiplier which provide fastercomputational speed, This multiplier is used in quantum computer ,as quantum computing perform computation inreversible manner

VI REFERENCES

[1] A. Avizienis, "Signed-digit number representations for fastparallel arithmetic,"IRE Trans. Electron. Computers, vol.

EC-10,pp. 389–400, 1961.
[2] N. Takagi, H. Yasuura, and S. Yajima, "High-speed VLSI multiplicationalgorithm with a redundant binary addition tree,"IEEE Trans. Computers, vol. C-34, pp. 789-796, 1985.
[3] Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa, and N.Takagi, "A high speed multiplier using a redundant binaryadder tree," IEEE J. Solid-State Circuits, vol. SC2, pp. 28-34,1987.

[4] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, "A33 MFLOPS floating point processor using redundant binaryrepresentation," in Proc. IEEE Int. Solid-State Circuits Conf.(ISSCC), pp. 152–153, 1988.
[5] H. Makino, Y. Nakase, and H. Shinohara, "A 8.8-ns 54x54-bitmultiplier using new redundant binary architecture," in Proc.Int. Conf. Comput. Design (ICCD), pp.202-205, 1993.
[6] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara,and K. Makino, "An 8.8-ns 54×54-bit multiplier with highspeed redundant binary architecture," IEEE J. Solid-State Circuits,vol. 31, pp. 773-783, 1996.
[7] Y. Kim, B. Song, J. Grosspietsch, and S. Gillig, "A carry-free54b×54b multiplier using equivalent bit conversion algorithm,"IEEE J. Solid-State Circuits, vol.36, pp.1538–1545, 2001.
[8] Y. He and C. Chang, "A power-delay efficient hybrid carrylookaheadcarry-select based redundant binary to two's complementconverter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol.55, pp. 336–346, 2008.
[9] G. Wang and M. Tull, "A new redundant binary number to 2'scomplementnumber converter," in Proc. Region 5 Conference:Annual Technical and Leadership Workshop, pp. 141-143, 2004.
[10] W. Yeh and C. Jen, "High-speed Booth encoded parallel multiplierdesign,"IEEE Trans. Computers, vol. 49, pp. 692-701,2000.
[11] S. Kuang, J. Wang, and C. Guo, "Modified Booth multiplierwith a regular partial product array," IEEE Trans. Circuits Syst.II, vol. 56, pp. 404-408, 2009.
[12] J. Kang and J. Gaudiot,"A simple high-speed multiplier design,"IEEE Trans.Computers, vol. 55, pp.1253-1258, 2006