International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

Design of 6 Bit Vedic Multiplier using Vedic Sutra

Yashkumar .M . Warkari
PG Student, Electronics Engineering Department,
G. H. Raisoni College Of Engineering, Nagpur

Prof. L. P. Thakre
Asst. Professor, Electronics Engineering Department, G.H. Raisoni College of Engineering, Nagpur

Dr. A. Y. Deshmukh
Professor, Electronics Engineering Department, G.H. Raisoni College of Engineering, Nagpur

***Abstract*:- The tenets of mathematics is existing since from long time which is in time span of years together. Related to the several kinds of fundamental mathematical necessities during technical sorting &processing with various categories of data for processing . In order to heed & step out from such facing difficult situations, some expertise had explained the mathematical tenets during ancient times. Later on the absolutely important things drawn out by expertise arduously which collectively produce the helpful tool named as vedic mathematics. The main aim of the project is to improve the speed of the multiplier by using vedic mathematics . The 6*6 bit vedic multiplier is the proposed design of this paper.**

I. INTRODUCTION

There are several fundamental operations incorporated in a vedic mathematics such as addition, subtraction, multiplication, division, geometrical, logical etc. All alluded operations have their own significance & utilization based on the practical approach of work to be done.

Now multiplication operation is a operation which dartles Within almost all domain of work . Therefore it is quoth to be a sinequanon operation. It is used in several practical applications like DCT , FFT processor , DFT reckoning , Harr transform , linear convolution ,RADAR system etc. While heeding on digital domain , it also plays an vital role at most cusp of a digital system design like digital filter , digital filter bank , Wallace tree . The desideratum is to design multiplier for FFT processor. The multiplication can be performed on two numbers at a time , for outset of the set of equations or for some other moto. It is a day to day life quintessence which came from primary level mathematical approaches. Vedic Mathematics is one such division that involves thinking and best to best mind utilisation. In India maximum number of math learner studying the conventional mathematics can solve problems that are taught them at their school, but they are not able to solve the problems that are new and not taught to them.

Vedic mathematics is dealing rudimentarily with ancient time mathematical system with Vedic mathematical formulae and their application to various branches of mathematics. Some experts had constructed 16 sutras and 16 upa sutras after extensive research in Atharva Veda.

Author has proposed The design of the proposed 4x4 bit Vedic multiplier is implemented on Spartan xc3s50a-5-tq144 device. The computational path delay of the Vedic multiplier is found to be 13.102 ns In [1].

It proposed that the multiplication of two small numbers and addition. Hence, there is significant speed amendment for 16x16 Vedic multiplier implementation using Nikhilam Sutra In [2]. Due to various factors like timing efficiency, space, lesser area and PDP the traditional multipliers can be replaced by vedic multiplier In [3]. The device used in this proposed design is 7a30tcsg324-3.

II. IMPORTANCE OF VEDIC MATHEMATICS

High speed arithmetic operations are very important in many signal processing applications. Speed of the digital signal processor (DSP) is largely determined by the speed multiplying operation.The multiplication is very primarily important part of all digital signal processors; they are very important in realizing many important functions such as fast Fourier transforms and convolutions. Since a processor spends particular span of time in doing multiplication, an augmentationt in multiplication speed may greatly improve system working. Multiplication would be realised using many algorithms such as group models, carry save adder, design like Wallace tree.

The multiplier architecture is based on this tenet Urdhva tiryakbhyam sutra. The utilisation of this tenet is to do partial products and their sums are reckon of parallel. This parallelism do not makes the multiplier clock dependent.The other benefit is to perform operation in regularity in place of other designs. Because of such kind of modular nature the drawing of layout will become easy. The two numbers having 8 bit range each can be elucidated by the same algorithm easilyUrdhava Tiryakbhyam is a Sanskrit word which means vertically and cross pattern in English. General multiplication formula used by the method applicable to multiplication of all such cases. It is novel based on a concept through which

all partial products are generated concurrently. demonstrates a 4 x 4 binary multiplication using this method.

The method can be generalized for any N x N bit multiplication numbers. This Kind of multiplication operation is non dependent of the clock frequency of the processor because the partial products and their sums are reckon in parallel. The total advantage is that it reduces the need of microprocessors to operate at higher and by much frequencies of clock. As the operating frequency of a processor increases the number of switching instances also increases.

III. EFFICACY OF VEDIC OVER CONVENTIONAL METHOD

In general we are omniscient regarding two main methods of multiplication , that is conventional method as well as efficient method. Efficient method elucidates various computer algorithms, vedic sutras,booths multiplier,Wallace tree etc . In this paper our cynosure is vedic sutra.

When at a outset of any digital circuit design one has to follow several steps many times for large designs.

As per as conventional method is concerned the first step is to prepare truth table .In order to prepare truth table based on total bit length of input of any circuit, how many input combinations has to write is to be write it correctly.

Ex- for 2 bit input
2^2 =4 input combinations.

Ex-for 3 bit input
2^3 =8 input combinations.

In above two examples it is doddle to manually write reckoned number of combinations . After that the k-map can be prepared .Logical expression of circuit can be obtained and then the quoth circuit can be implemented.

When the same conventional methodology has to be applied for the 6-bit vedic multiplier the design of circuit come across predicament. The moto of implementation would reach towards the pessimism of successful design.

In order to construct 6 bit vedic multiplier design it ought to utilize two 6-bit numbers. It ought to consider total 12-bit input combining of both 6-bits of two input numbers.

Total 12-bit input

Table No-1. 2^{12} =4096 input combinations

Number of combinations	Input-1	Input-2	output
0 1 Upto 4096	Bits for first number	Bits for second number	Output bits

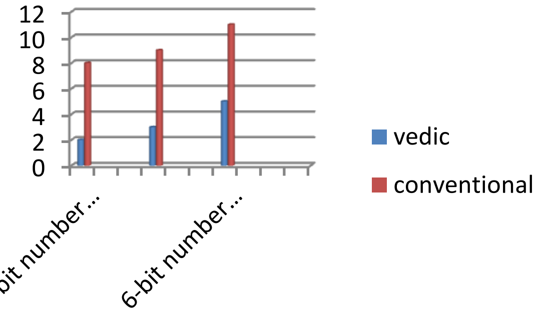


Fig1- Reckoning comparison graph

Above is the reckoning speed graph which represents time required on y-axis & length of input on x-axis

Practically it is not feasible to write 4096 number of input combinations in truth table. Let us tattle apropos of next step which is a k-map. It is understood that number of input bits is equal to that many variable k-map has to be used. But k-map has got lacuna over logical myriads. for 2,3,4,5 bit variable k-map it is feasible to draw k-map .But in this case that is six bit multiplier we ought to go for 12-variable k-map. But it is very cumbersome to draw 12-variable k-map. Then cogitate apropos to logical expression. The lengthy logical expression can be obtained out of alluded k-map for our quoth design.

There are again several methods that can be followed to reduce logical expression such as Boolean algebra , tabulation method etc. It is tedious task to make a use of basic Boolean reducing rules to apply over a huge logical terms in such a lengthy expression. Now let us cull another alluded approach which is nothing but tabulation method. At outset of tabulation we ought to write number of steps. let us tattle on one step ,which is a classification of groups of combinations containing same number of logic 1 bits. It is tedious to sort entire 4096 combinations. likewise remaining steps like prime implicant chart etc is cumbersome to be follow.

So after expatiating all such associating limitations for myriads of conventional method, the connotation is that the conventional method is not feasible for myriads & myriads can be incorporated in several realistic mathematical based applications. Hence for myriads the smart methods such as vedic tenets has to be incorporated. The vedic methods are capable of obliterating almost all the alluded lacunas alluded above for conventional method. Hence multiplications in design

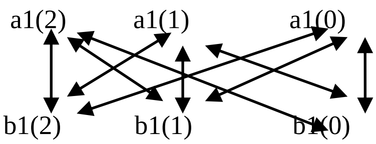
blocks can be performed at faster rate. This is a very interesting feature of vedic tenets of multiplication and presents some effective algorithms which would be applied to several distinct branches of engineering.

IV. METHODOLOGY OF 3-BIT VEDIC MULTIPLIER

Basically two popular vedic multiplication sutras are preferred. One is Nikhilam and another is urdhavya Triyakbhayam. We are imitating the urdhavya Triyakbhayam

Sutra , since it has got its own significance , versatility from small numbers to myriads .

Let number 'A' be of 3 bit i.e a2 a1 a0 and number 'B' be of 3 bit i.e b2 b1 b0



The governing equations are as follows

1) (a1(0) * b1(0))
2) (a1(1) * b1(0)) + (a1(0) * b1(1))
3) (a1(2) * b1(0)) +(a1(1) * b1(1))+(a1(0) * b1(2))
4) (a1(2) * b1(1)) + (a1(1) * b1(2))
5) (a1(2) * b1(2))

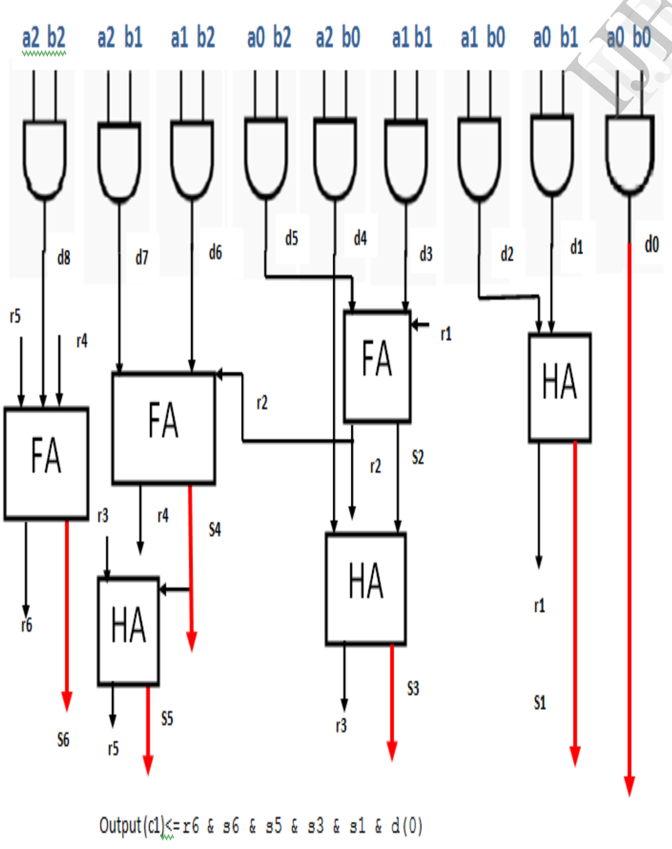


Fig2- circuit of 3 bit vedic multiplier

The 3 bit vedic multiplier has used as a primary unit with which the ensuing dependant multiplier designs can be designed. we cannot prefer any other bit range at the inception of our design . 2,4,6,8 will grow like (4,8,16) , (8 ,16,32) ,(12,24,36),(16,32,64). Only 3 is a value which elicits the 6 bit result as a progress stage.

Design has used urdhavya Triyakbhayam sutra which means vertical as well as crosswise multiplication. First step is to arrange the two 3 bit numbers in two individual rows one below other, then incept with formulating the equations. Multiply LSB of 'A' with LSB of 'B' .Then in second equation multiply second bit of 'A' with LSB of 'B' as well as LSB of 'A' with second bit of 'B'. In third equation multiply MSB of 'A' with LSB of 'B' and second bit of 'A' with second bit of 'B', also LSB of 'A' with MSB of 'B' . In fourth equation multiply MSB of 'A' with second bit of 'B' and MSB of 'B' with second bit of 'A'. Finally the fifth equation elicits product of MSB of both numbers.

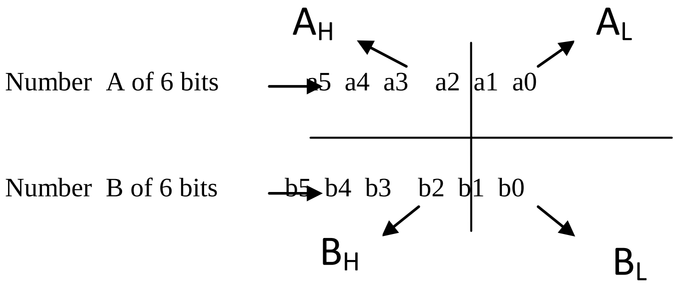
In above elucidation only corresponding bits multiplication as well as cross multiplication has been performed and therefore it is quoth to be framed under tenet of urdhavya Triyakbhayam. The urdhavya means vertical & Triyakbhayam means crosswise. since all the nine individual product terms are products of single bit and hence all product terms can be implemented by aid of nine and gate array delineated in fig2. The remaining addition operation can be implemented by the aid of coadjutor circuit i.e adder circuit.

The signal d1 and d2 can be reckon by half adder & the sum s1 can be drawn out. The signal d3 and d5 can be reckon by aid of full adder , the third input of this full adder is cull as a carry signal of half adder i .e r1 , it elicits sum s2 and carry r2. Signal d4 and sum s2 can be again reckon by half adder eliciting sum s3 and carry r3. Signal d7 & d8 as well as signal r2 can be reckoned by aid of full adder again eliciting sum s4 and carry r4. Then reckoned sum s4 and carry r3 of previous half adder can be reckon by aid of one more half adder eliciting the sum & carry signals s5 & r5 respectively. Eventually signals d8 ,r4,r5 can be reckon by aid of full adder eliciting sum as s6 and carry as r6.After implementing the addition hardware for product terms elicited by and gate array , we ought to conglomerate all output signals of adder components. The r6,s6,s5,s3,s1,d0 has to be assigned to output variable defined in entity in same order alluded above , r6 as MSB and d0 as LSB. Such circuits can be implemented for various tolerable bit range .This tenet has been used for larger applications like programming ,DSP processors etc where the multiplication is quoth to be a rudimentary as well as most frequently utilized mathematical operation for more than one process of other estimation of various parameters. In this work the algorithms are implemented in VHDL and a simulation can be obtained using ISIM simulator.

V. METHODOLOGY OF 6-BIT VEDIC MULTIPLIER

The 6-bit vedic multiplier design is a progressive block of 3-bit vedic multiplier design. The 3-bit multiplier design delineated & expatiated above is used as one of the

component in a structural style of modeling by aid of VHDL code.
Let number 'A' be of 6 bit i.e a5 a4 a3 a2 a1 a0 and number 'B' be of 6 bit i.e b5 b4 b3 b2 b1 b0.



Equation=>$(A_H * B_H) + (A_H * B_L) + (A_L * B_H) + (A_L * B_L)$

A) For the sake of well understanding it is necessary to refer some similar kind of circuit in Fig3 of previous publications which has a nexus with our paper.

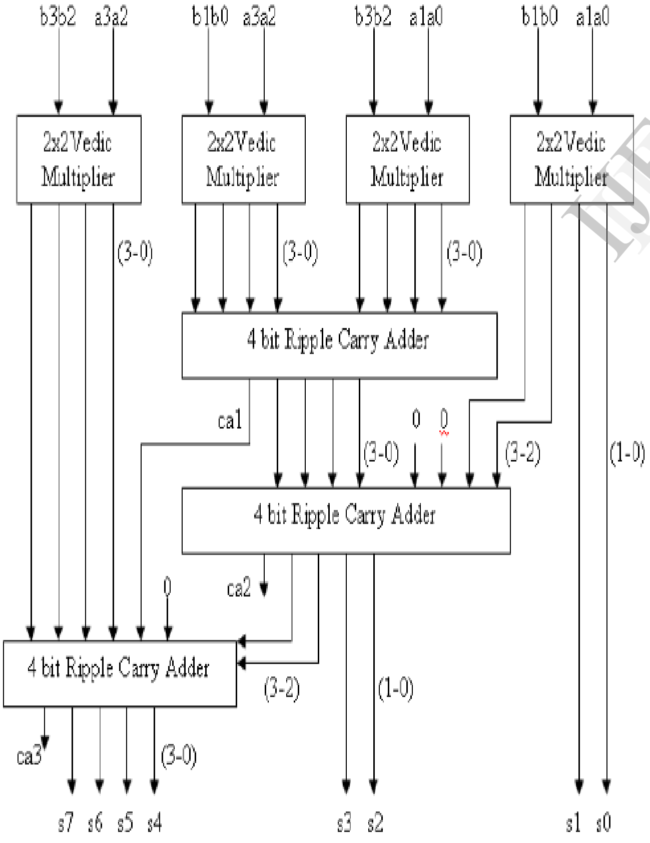


Fig3- The 4x4 bit Vedic Multiplier

B) The circuit delineated below is our modified block having nexus with circuit shown in Fig2 with slight variation.

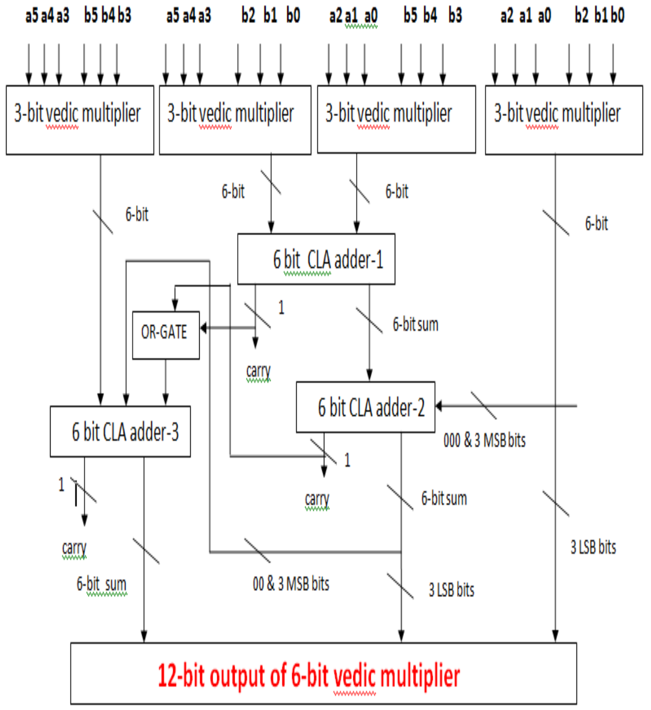


Fig4- The 6 * 6 bit vedic multiplier
CLA=Carry Look ahead Adder

The two numbers 6-bit each is ramified into four parts i.e A_H, A_L, B_H, B_L . Each subpart is of 3-bit which means four 3-bit vedic multiplier again. Simply map the above 3-bit vedic multiplier design as a component in a Fig4.
When the 3-bit vedic in Fig2 properly map then it elicit the 6-bit output from all four 3-bit vedic unit in Fig4.After that the role of adders comes into picture which exhibits through Equation elucidated above.
Then the output elicited from middle two 3-bit vedic units is endowed to 6bit CLA adder-1.The all output bits excluding carry has used as one input of 6bit CLA adder-2 and the second input of 6bit CLA adder-2 is framed by concatenating 3 MSB output bits of last 3-bit vedic multiplier unit with three leading zeroes. The output elicited from First 3-bit vedic unit is endowed to 6bit CLA adder-3 as one input and the second input of 6bit CLA adder-3 is framed by concatenating 3 MSB output bits of last 6-bit CLA adder-2 unit with two leading zeroes and output of OR gate. Output carries of 6bit CLA adder-1 & 6bit CLA adder-2 has used as an input of OR gate .Eventually the 12 bit output can be obtained by concatenating the bits shown in Fig4.

C) Significance of OR gate

Proposed design in paper shown in fig4 is more appropriate block for all numbers. The tenet used in Fig-3 will work for most numbers. But if the circuit in Fig-4 is designed by making use of same tenet imitated in Fig-3,then the output for some numbers will obtained incorrect.
Therefore the OR gate is adequate enough to obliterate the incorrectness. For reckoning in 6bit CLA adder-3 both the carries will be simultaneously considered.
Ex- 47 * 39=1833

Above is the quintessence to get the significance of OR gate. when 47 and 39 are put as an inputs in the circuit of Fig 4 then it elicit the output carry of 6bit CLA adder-1 as logic ‘0’ and output carry of 6bit CLA adder-2 as logic ‘1’ and here carry bit of 6bit CLA adder-2 will be considered. Whereas if the OR gate is connived from Fig 4 then whatever circuit is obtained is simply imitating of tenet in Fig3.If such circuit is used & 47 * 39 is reckon then carry bit of 6bit CLA adder-2 will be connived & entire 12 bit answer will obtained incorrect.

VI. CLA over Ripple carry adder in design

To understand the merit of CLA we ought to cogitate the connection type approach used in both adders. Basically in Ripple carry adder the reckoning of sum & carry bit of any stage excluding first stage cannot be performed till carry bit from previous stage peregrinates to its next stage. The connotation is that each successive stage is a function of a carry elicits by previous stage. The glimpse of above words is that the ripple carry adder imitate the serial connection

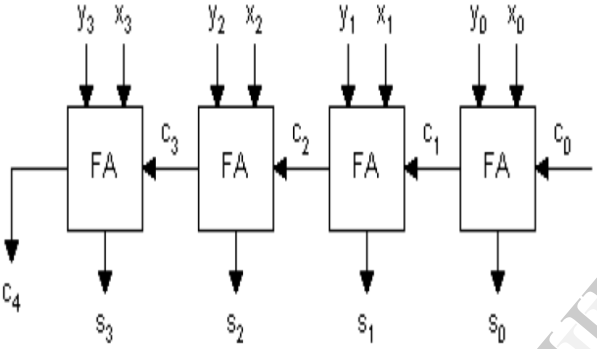


Fig5. Ripple carry adder

Whereas the CLA is independent of reckoning addition by making use of carry of previous stage. All the carries are reckoned in advance It can be possible by framing equations which are all independent & only dependent on initial carry i.e C_0 which is equal to ‘0’ always. Thus it saves the carry peregrination time & hence CLA is quoth to be a more faster adder than Ripple carry adder.

G_i =Ai .Bi Pi =Ai xor Bi

$C_1 = G_0 + P_0C_0$

$C_2 = G_1 + P_1C_1 = G_1 + P_1G_0 + P_1P_0C_0$

$C_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$

$C_4= G_3+ P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$ =G3 + p3c3

$S_i = P_i$ xor C_i

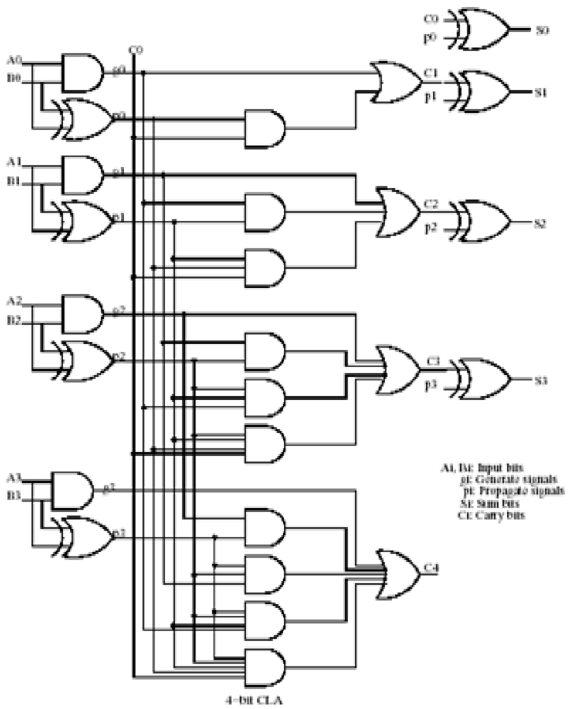


Fig6.Carry look ahead adder (CLA)

In proposed design our moto is to elicit output at much rapid rate. For that we ought to augment reckoning speed & this is possible by implementing CLA instead of Ripple carry adder. there are many applications like calculator , DSP processor , FFT reckoning where reckoning speed plays an important and sinequanon contribution.

Therefore the below table will doff the baffle regarding substantiation of CLA along with comparison with other types of adders.

Table No-2 Assumed values of power for logic gates

Logic GATES	Assumed values of power consumption
AND	2.4
OR	1.78
XOR	3.71
NAND	3.8
NOR	3.18
NOT	1.4

Table No-2 Assumed values of power for logic gate

Name of adder For(n=6)	No of Gates	Power utilization in mw	Delay required
1.Ripple carry adder	30	84	24
2.Carry look ahead adder	58	144.91	4
3.Squat carry select adder with RCA	57	142.52	6
4. Squat carry select adder with CLA	70	172.4	4.8

From the above table it is cleared that The carry look ahead adder is better for better reckoning speed. The delay governs the time required to elicit the output signal.

The table given below indicates the synthesis parameters

Table No-4 Synthesis report values

Parameters	Synthesis values
1.Input buffers	12
2.Output buffers	12
3.Number of slice LUT’S	56
4.Number of flip flops	56
5.Memory utilization	290524 KB

VII. SIMULATION RESULTS

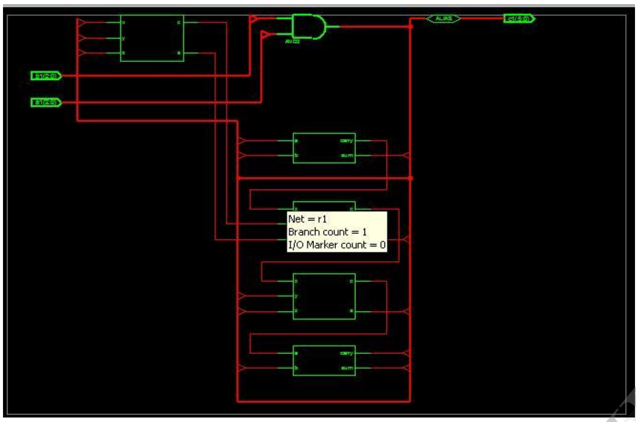


Fig-7 RTL of 3bit vedic multiplier

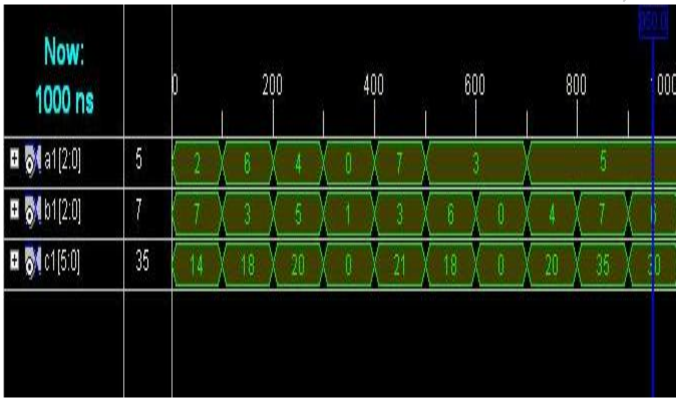


Fig8 Simulation result of 3 bit vedic multiplier

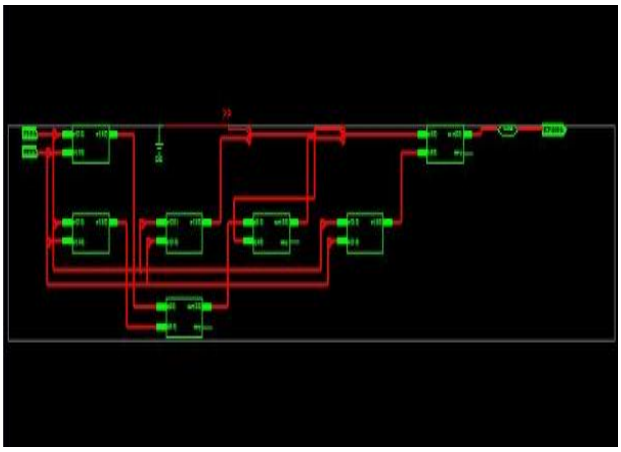
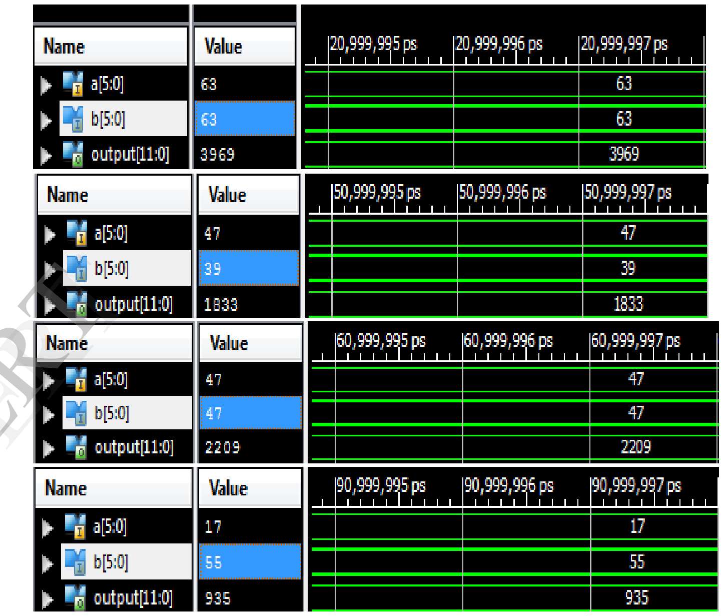


Fig-9 RTL of 6bit vedic multiplier



Name	Value	20,999,995 ps	20,999,996 ps	20,999,997 ps
a[5:0]	63			63
b[5:0]	63			63
output[11:0]	3969			3969

Name	Value	50,999,995 ps	50,999,996 ps	50,999,997 ps
a[5:0]	47			47
b[5:0]	39			39
output[11:0]	1833			1833

Name	Value	60,999,995 ps	60,999,996 ps	60,999,997 ps
a[5:0]	47			47
b[5:0]	47			47
output[11:0]	2209			2209

Name	Value	90,999,995 ps	90,999,996 ps	90,999,997 ps
a[5:0]	17			17
b[5:0]	55			55
output[11:0]	935			935

Fig-10 Simulation Result of 6bit vedic multiplier

VIII. CONCLUSION

Thus we can conclude that the above circuit in Fig4 elicits the more correct results for all numbers in 6 bit range .whereas the architecture in Fig3 elicits incorrect results for some numbers if it can be designed for 6 bit range .Therefore implementation of OR gate is sinequanon for more accuracy & hence our design is better in terms of correct reckoning. The carry look ahead adder also substantiated to be a coadjutor adder circuit to reduce the reckon delay and it can be proved from above table figures. The overall design is having complaisance for proviso to get more reckon accuracy as well as reckoning speed.

REFERENCES

1. Pushpalata Verma, Design of 4x4 bit Vedic Multiplier using EDA Tool , International Journal of Computer Applications (0975 – 888) Volume 48– No.20, June 2012
2. Manoranjan Pradhan, Rutuparna Panda, Sushanta Kumar Sahu Speed Comparison of 16x16 Vedic Multipliers, International Journal of Computer Applications (0975 – 8887) Volume 21– No.6, May 2011
3. Arushi Somani, Dheeraj Jain, Sanjay Jaiswal, Kumkum Verma and Swati Kasht,compare vedic multiplier with conventional hierarchical array of array multipliers, International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 6
4. P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu, Vol.8, No.2, 2004 UICEE Published in Australia.
5. Thapliyal H. and Srinivas M.B, "High Speed Efficient NxN Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics", Transactions on Engineering, Computing and Technology, 2004, Vol.2.
6. Shamim Akhter, "VHDL Implementation of Fast NXN Multiplier Based On Vedic Mathematics". Jaypee Institute of Information Technology
7. Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced– Bit Multipliction Algorithm for Digital Arithmatics", International Journal of Computational and Mathematical Sciences 2.2 @ www.waset.orgSpring2008.
8. Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho, "Multiplier design based on ancient Indian Vedic Mathematician", International SoC Design Conference, 2008.
9. Parth Mehta, Dhanashri Gawali, "Conventional versus Vedic mathematics method for Hardware implementation of a multiplier", International conference on Advance in Computing, Control, and Telecommunication Technologies, 2009.