ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

A HIGH SPEED AND EFFICIENT DESIGN FOR BINARY NUMBER SQUARING USING DWANDWA YOGA

B. Dilli kumar [1], M. Bharathi [2]

***Abstract*-The complexity in implementing complex logic functions in hardware circuitry is to be reduced in order to perform large calculations with minimum delay. This paper presents a most efficient and high speed design for doubling a binary number using Dwandwa Yoga logic, a squaring algorithm. The calculation is performed based on the "Duplex" D property. This method reduces the carry propagation delay when compared to the other vedic multiplication algorithms and conventional multiplication algorithms to a great extent. As the number of bits increases the size of the hardware circuitry decreases to a great extent by using the proposed logic. For the same number of bits, the dwandwa yoga requires less number of calculations compared to Urdhva tiryakbhyam. This design can be further used in apllications where low power and small area are main criteria.**

***Keywords*-Dwandwa yoga, duplex, high speed, minimum propagation delay, squaring algorithm, multiplier.**

I. INTRODUCTION

In many complex multiplications the square of a number can be calculated using multiplier unit. Perhaps it is one of the most time consuming operations in implementing large hardware circuitry. Squaring operation is a special case of multiplication unit. A dedicated square hardware may significantly improve the computation time and reduces the delay to a large extent.

The main core components of all the digital signal processors are digital multipliers and the performance and speed of the DSP depends on its multipliers. These digital multipliers are used in realizing many of the DSP computations like FFT(fast fourier transform), MAC(multiplier and accumulator) and DFT(discrete fourier transforms). The commonly used multiplication algorithms in digital hardware are array multiplication algorithm and Booth multiplication algorithm. In array multiplier as the partial products are calculated independently, the computation time is relatively less. The delay associated with the array multiplier is the time taken by the signal to propagate through the gates that form the multiplication array.

Another important multiplication algorithm is booth multiplication algorithm. Large booth arrays are required for high speed multiplication and exponential operation. But , in order to implement large booth arrays large partial sum and partial carry registers are required. Multiplication of two *n*-bit operands using a radix-4 booth recording multiplier requires approximately $n=(2m)$ clock cycles to generate the least significant half of the final product, where *m* is the number of Booth recoder adder stages. Thus, a large propagation delay is associated with this case.

The proposed square architecture is an improvement over partition multipliers in which the NXN bit multiplication can be performed by decomposing the multiplicand and multiplier bits into M partitions where M=N/K (here N is the width of multiplicand and multiplier(divisible by 4) and K is a multiple of 4 such as 4, 8 , 12 ,16…….. 4* n). The partition multipliers are the fastest multipliers implemented in the commercial processors and are much faster than conventional multipliers.

II. DWANDWA YOGA

The *Dwandwa Yoga* or 'duplex combination' can be used for general purpose squaring. The square of a number can be calculated by using the 'duplex' D property of dwandwa yoga. According to duplex property, for an even number of elements the result is taken as twice the product of the outermost pair and then twice the product of the next outermost pair and so on till no pairs are left. For an odd number of elements, there is one bit left itself in the middle and this enters as its square along with the product elements. It can be explained in the following example

$D(a) = a^2$

$D(ab) = 2ab$

$D(abc) = 2ac + b^2$

$D(abcd) = 2ad + 2bc$

$D(abcde) = 2ae + 2bd + c^2$

$D(abcdef) = 2af + 2be + 2cd$ and so on....

As we can see above, D of any number is the sum of square of the middle number and two times the product of the

other pairs.

Square of a number is given by

$(ab)^2 = D(a) | D(ab) | D(b)$

$(abc)^2 = D(a) | D(ab) | D(abc)|D(bc) | D(c)$

$(abcd)^2 = D(a) | D(ab) | D(abc) | D(abcd) | D(bcd) | D(bc) | D(c)$

Example :

$(25)^2 =$
$D(5) = 5^2 = 25 = 5 = A$
$D(25) = 2*2*5 = 20 = 20+2 = 22 = 2 = B$
$D(2) = 2^2 = 4 = 4+2 = 6 = C$
Now the required result after squaring 25 is CBA = 625.

Thus for a single bit number, the D is square of the number itself. For a 2 bit number, it is twice their product. For a 3 bit number, it is the sum of twice the product of the outermost pair and square of the middle number. For a 4 bit number, it is the sum of twice the product of the outermost pair and twice the product of the innermost pair.

III. Conventional squaring unit

The multiplication operation is one of the most important functions in many real time applications. In ordinary sequential multiplications the multiplicand is shifted bit by bit and added to a large 2n bit accumulator when the bit at the corresponding bit position of the multiplier is 1. To increase the multiplication speed the addition operation can be performed in parallel. In a straight forward parallel multiplication, the addition operations are carried out by an array of n(n-1) full adders. The squaring unit also performs the same kind of operation similar to the conventional multiplier except its inputs for both the multiplier and multiplicand is same. For the same number of elements the squaring unit requires less computations compared to the multiplication unit.

				A3	A2	A1	A0
				A3	A2	A1	A0
				A3A0	A2A0	A1A0	A0A0
			A1A3	A1A2	A1A1	A0A1	
		A2A3	A2A2	A2A1	A2A0		
	A3A3	A3A2	A3A1	A3A0			
P7	P6	P5	P4	P3	P2	P1	P0

The conventional squaring unit is a time consuming process as it requires large number of computations. So the delay associated with this type of algorithm is more which makes it not useful for handling large complex functions in designing most sophisticated hardware circuitry.

IV. Proposed squaring algorithm

In the proposed algorithm the square of a binary number can be calculated based on the duplex property of dwandwa yoga logic. This squaring algorithm has all the advantages as it is quite smaller than the array, booth and vedic multiplier.

This multiplication algorithm is also advantageous over urdhava tiryakbhyam multiplication algorithm as it requires less number of computations over urdhava tiryakbhyam for the same number of bits.

The algorithm is explained for 4 x 4 bit number.

a) Algorithm for 4 x 4 bit Square using Dwandwa Yoga

A3	A2	A1	A0	Multiplicand
A3	A2	A1	A0	Multiplier
H	G	F	E	D	C	B	A	Partial products
P7	P6	P5	P4	P3	P2	P1	P0	Final result

PARALLEL COMPUTATION

1. D(A0) = A 0 * A0 = A
2. D (A1A0) = 2 * A1 * A0 = B
3. D (A2A1A0) = 2 * A2 * A0 + A1 * A1 = C
4. D (A3A2A1A0) = 2 * A3 * A0 + 2 * A2 * A1 = D
5. D (A3A2A1) = 2 * A3 * A1 + A2 * A2 = E
6. D (A3A2) = 2 * A3 * A2 = F
7. D (A3) = A3 * A3 = G

The hardware architecture for 4 x 4 bit binary squaring is

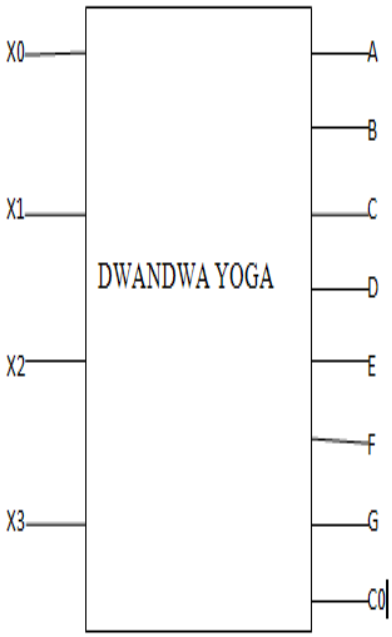

Figure 1: Block diagram for 4 x 4 bit squaring using dwandwa yoga

The logic can be best explained with the following example

Example :
$(1111)^2$ =
D(1) = 1 *1 = 1= A
D(11) = 2 * 1 * 1 = 10 = 0 = B
D(111) = 2 * 1 * 1 + 1 * 1 = 11 = 0 = C
D(1111)=2*1 * 1 + 2 * 1 * 1=100= 0 =D
D(111) = 2 * 1 * 1 + 1 * 1 = 0 = E
D(11) = 2 * 1 * 1 =10 = 1 = F
D(1) = 1 * 1 = 11 = C0 G

The final result that is obtained after squaring the binary number 1111 is C0GFEDCBA = 11100001.

This squaring algorithm has less number of gates required for given 8x8 bits Multiplier so its power dissipation is very small as compared to other multiplier architecture.This vedic multiplier is faster than array multiplier and Booth multiplier. As the number of bits increases from 8x8 bits to 16x16 bits, the timing 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 regularity of structures. The area needed for Vedic square multiplier is very small as compared to other multiplier architectures.

Thus the result shows that the Vedic square multiplier is smallest and the fastest of the reviewed architectures. The Vedic square and cube architecture proved to exhibit improved efficiency in terms of speed and area compared to Booth and Array Multiplier. Due to its parallel and regular structure, this architecture can be easily realized on silicon and can work at high speed without increasing the clock frequency.

V. Comparison of Dwandwa Yoga with Conventional Multiplier

When compared to the conventional multiplier the performance of the proposed squaring algorithm that uses dwandwa yoga logic has been increased to a great extent. The number of LUTs has also been decreased in the proposed logic compared to the conventional multiplier. As the number of bits increases the number of LUTs decreases to a great extent results in decreasing the size of the hardware circuitry. Similarly the delay of the proposed logic has also decreased to a great extent which makes the proposed algorithm to be used in the applications where high speed is required.

Table1: Comparison of dwandwa yoga with conventional multiplier

Number of bits	Delay(ns)		Number of LUTs	
	conventional multiplier	Proposed squaring unit	Conventional multiplier	Proposed squaring unit
4	8.964	6.422	32	6
8	15.421	14.256	186	35
16	36.425	32.324	880	294

With these advantages the proposed algorithm can be used to decrease the hardware complexity while implementing in most sophisticated systems.

VI. Simulation Result

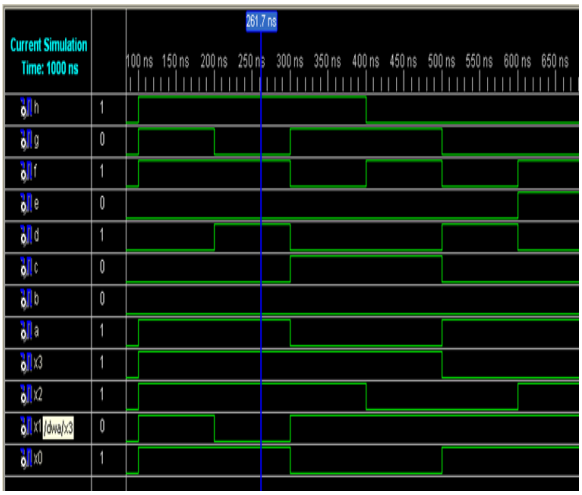

Figure 3: Output waveforms for 4 x 4 bit dwandwa yoga logic

The results are simulated in Xilinx 10.1 ISE of Spartan3E family and the package of the device is FG320 with a speed of -4.

The fig.3 shows the simulation results for a given four bit binary number. It has four inputs and a maximum of eight outputs. To perfom the required logic a four bit adder and a five bit adder has been designed.

VII. CONCLUSION

In this paper a new binary number squaring algorithm have been proposed. The duplex property is used to perform the squaring calculations. In addition a 4 x 4 bit binary squaring resuts have also been discussed. With the help of this algorithm large computations can be handled significantly with less delay. It is therefore seen that the Vedic multipliers are much more faster than the conventional multipliers. The algorithms of Vedic mathematics are much more efficient than that of conventional mathematics.

In future it is possible to implement cubing algorithm using the basic principles of proposed logic.

REFERENCES

[1] Purushottam D. Chidgupkar and Mangesh T. Karad, "T Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engng. Educ., Vol.8, No.2 © 2004 UICEE Published in Australia.

[2] Himanshu Thapliyal and Hamid R. Arabnia, "A Time-Area- Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics", Department of Computer Science, The University of Georgia, 415 Graduate Studies Research Center Athens, Georgia 30602-7404, U.S.A.

[3] E. Abu-Shama, M. B. Maaz, M. A. Bayoumi, "A Fast and Low Power Multiplier Architecture", The Center for Advanced Computer Studies, The University of South western Louisiana Lafayette, LA 70504.

[4] Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced- Bit Multiplication Algorithm for Digital Arithmetics", International Journal of Computational and Mathematical Sciences 2;2 © www.waset.org Spring 2008.

[5] Shamim Akhter, "VHDL Implementation of Fast NXN Multiplier Based on Vedic Mathematics", Jaypee Institute of Information Technology University, Noida, 201307 UP, INDIA, 2007 IEEE.

[6] Charles E. Stroud, "A Designer"s Guide to Built-In Self-Test", University of North Carolina at Charlotte, ©2002 Kluwer Academic Publishers New York, Boston, Dordrecht, London, Moscow.

[7] Douglas Densmore, "Built-In-Self Test (BIST) Implementations An overview of design tradeoffs", University of Michigan EECS 579 – Digital Systems Testing by Professor John P. Hayes 12/7/01.

[8] Shripad Kulkarni, "Discrete Fourier Transform (DFT) by using Vedic Mathematics", report, vedicmathsindia.blogspot.com, 2007.

[9] Jagadguru Swami Sri Bharati Krishna Tirthji Maharaja,"Vedic Mathematics", Moti Banarsidas, Varanasi, India, 1986.

[10] Himanshu Thapliyal, Saurabh Kotiyal and M. B Srinivas, "Design and Analysis of A Novel Parallel Square and Cube Architecture Based On Ancient Indian Vedic Mathematics", Centre for VLSI and Embedded System Technologies, International Institute of Information Technology, Hyderabad, 500019, India, 2005 IEEE.

[11] Himanshu Thapliyal and M.B Srinivas, "VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics", Center for VLSI and Embedded System Technologies, International Institute of Information Technology Hyderabad-500019, India.

[12] Abhijeet Kumar, Dilip Kumr, Siddhi, "Hardware Implementation of 16*16 bit Multiplier and Square using Vedic Mathematics", Design Engineer, CDAC, Moali.

[13] Himanshu Thapliyal and M.B Srinivas, "An Efficient Method of Elliptic Curve Encryption Using Ancient Indian Vedic Mathematics", IEEE, 2005.

[14] "Spartan-3E FPGA Starter Kit Board User Guide", UG230 (v1.1) June 20, 2008.

[15] Deming Chen, Jason Cong, and Peichan Pan, "FPGA Design Automation: A Survey", Foundations and Trends in Electronic Design Automation Volume 1 Issue 3, November 2006.

[16] Ken Chapman, "Initial Design for Spartan-3E Starter Kit (LCD Display Control)", Xilinx Ltd 16th February 2006.

[17] Goh Keng Hoo, "Verilog design of Input / Output Processor with Built-In-Self-Test", Universiti Teknologi Malaysia, April 2007.

[18] Michael L. Bushnell and Vishwani D. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", Kluwer Academic Publishers, 2002.

[19] R. Bencivenga, T. J. Chakraborty and S. Davidson, "The Architecture of the Gentest Sequential Test Generator", in Proc. of the Custom Integrated Circuits Conference, pp. 17.1.1–17.1.4, May 1991.

[20] G. K Ma, F. J. Taylor, "Multiplier policies for digital signal processing", IEEE ASSP Mag., vol.7,no.1,pp.6-20,Jan.1990.

[21] A. D. Booth, " A signed Binary multiplication technique", Qrt. J. Mech. App. Math.,, vol. 4, no. 2, pp.236-240,1951.

[22] D. Ait-Boudaoud, M. K. Ibraham and B. R. Hayes-Gill, "Novel pipelined serial/parallel multiplier", Electron. Lett., Vol. 26, no. 9, pp. 582-583,april,1990.

[23] B. K. Tirtha, Vedic Mathematics. Delhi: Motilal Banarasidass Publishers.

[24] 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, pp. 153-158, 2004.

[25] R. Gnanasekaran,"A Fast serial-parallel binary multiplier", IEEE Trans. Comput., vol. 34, no. 8, pp.741—744,Aug.1985.

B. Dilli Kumar, M.Tech(VLSI), Department of ECE, member IAENG, Sree Vidyanikethan Engineering College (Autonomous) , Tirupati, IndiaHe has completed B.Tech in Electronics and Communication Engineering, in Jawaharlal Nehru Technological University, Anantapur. His research areas are VLSI, Digital IC Design, and VLSI and Signal processing, Testing and Testability.
dillikumar406@gmail.com

M. Bharathi, Assistant Professor, Department of ECE, Sree VidyanikethanEngineering College (Autonomous), Tirupati, India. She has completed M.Tech in VLSI Design, in Satyabhama University. Her research areas are Digital System Design, VLSI Signal Processing
bharathi891@gmail.com