DESIGN AND IMPLEMENTATION OF EFFICIENT VEDIC MULTIPLIER USING XILINX

T.ALMAZ Student, M.Tech (VLSI SYSTEM DESIGN ENGINEERING), NIMRA COLLEGE OF ENGINEERING AND TECHNOLOGY, A.P., India.

Dr. AKBER KHAN Professor, DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING, NIMRA COLLEGE OF ENGINEERING AND TECHNOLOGY, A.P., India

ABSTRACT:

Today most of the processor requires very high speed of operation. Usually DSP processors are based on mathematical approaches. In that multiply-accumulate operation plays vital role. Compared to addition, multiplication process takes large amount of time thus reduces the speed of the processor, consumes some amount of power and area. In this paper we proposed technique to improve processor speed based on Vedic mathematics. In Vedic mathematics among 16 sutras, 1 sutras is applicable for multiplication. First method URDHAVA TRIYAKBHYAM sutra which is similar to array multiplication .When number of bits increases, gate delay and area increases slowly compared to other multiplier. These sutras are meant for faster mental calculation. Though faster when implemented in hardware, it consumes more power than the conventional ones. In this project both the techniques are compared and found that URDHAVA TRIYAKBHYAM is best. This project presents a technique to modify the architecture of the Vedic multiplier by using some existing methods in order increase the processor speed.

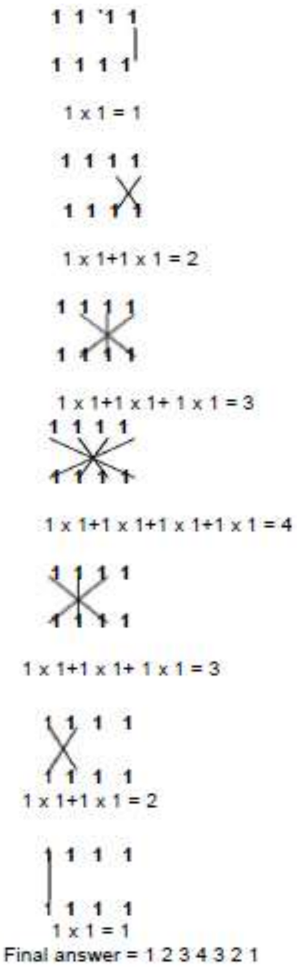
KEYWORDS: Vedic, dissipation, Latency, Complementary metal oxide semiconductor, Fabrication.

INTRODUCTION

Among the forceful investigation in the field of low power, high speed digital applications due to the growing demand of systems like phones, laptop, palmtop computers, cellular phones, wireless modems and portable multimedia applications etc has directed the VLSI technology to scale down to nano-regimes, allowing additional functionalityto be incorporated on a single chip. The designer‟s novel purpose in the field of multifaceted digital circuit design is minimization of power consumption. These investigations are responsible for special design techniques for digital circuits distant from conventional CMOS design style. A large body of investigate has been performed to expand and advance conventional Complementary Metal Oxide Semiconductor (CMOS) techniques for the fabrication ofULTRA low power integrated circuits (ICs). The purpose of this study is to expand a faster, lower power, and reduced area substitute to standard CMOS logic circuits. Mod-GDI technique is one such new technique for minimization ofpower consumption in the digital circuit design field[1,2,3]. The increasing market of portable, battery-powered electronic systems preferred microelectronic circuits design with ultra low power dissipation. Since the size, complexity and integration of the chips keep on to increase, thecomplexity in providing enough cooling might either include important cost or limit the operability of the computing systems which formulate utilization of those integrated circuits. At the same time as the technology node scalesdown to 65nm regime, not to a great extent increase in dynamic power dissipation is noticed, though the static or leakagepower is similar as or exceeds the dynamic power beyond 65nm technology node. Therefore the methods of powerdissipation reduction are not limited to dynamic power. In this paper we discuss circuit and logic level design approaches topower mini ization such that dynamic, leakage and short circuit power dissipation. Power optimization for low powerapplications can be achieved at different levels such that System level, Algorithm level , Architecture level, Circuit/Logic level and technology level. Logic level power optimization method have been considered here which have ahuge potential for power saving. As a result optimization at circuit / logic level is moreover very imperative for miniaturizationofIC[4]s. For the period of the last two decades investigation has resulted in the progress of numerous logic design techniques. An alternative approach to CMOS logic design is pass-transistor logic (PTL) design having advantagesover standard CMOS design, these are: high speed which results from the small node capacitances; low power dissipationwhich results from the reduced number of transistors used in fabrication; and lower interconnection effects which results from small area. But there were also some basic problems in PTL realizations; i) the threshold drop across thesinglechannel pass transistors which results in reduced current drive and therefore slower function at reduced supply voltages [1]. This threshold voltage drop is significant for low power design because it is advantageous to function at thelowest possible voltage level. ii) In view of the fact that the input voltage for a high logic level at the regenerative invertersis not VDD, the PMOS device in the inverter is not fully turned off, and for this reason direct-path static power dissipation can be important. After that many investigations have performed to resolve these problems which results in techniques like: Transmission Gate CMOS (TG), Complementary Pass-transistor Logic (CPL), and Double Pass-transistor Logic (DPL). The multiplier finds its usage in practically all kinds of processing systems ranging from application specific processors dealing with an infinitely large bit width or a small scale general processor dealing with a comparably smaller subset of data. It also happens to be one of the most time-consuming digital processes and offers a good scope for improvement in terms of area, delay as well as power efficiency.The conventional hierarchical array multiplication is a simplelooped method of binary multiplication in which a multiplicand is repetitively multiplied with individual bitsof second multiplicand and the partial product terms are then finally added to yield the product. The process, however simple in approach and implementable hierarchically, requires extensive hardware and is also marred by a heavy adder delay. The adder delays are often tackled with complex adder systems that in turn increase the hardware requirements of the system along with the inherent delay. We aim to tackle these hurdles by using the ancient concepts of Vedic Mathematics. Out of the hundreds oftheorems available in the arsenal of Vedic Mathematics, we aim to reduce the mathematical complexity of the binarymultiplication process by using the Urdhva-Tiryagbhyam (UT) theorem. It is basically a decimal multiplicationtheorem which effectively simplifies the multiplication process.

URDHVA TIRYAGBHYAM

*Urdhva – Triyakbhyam*is the general formula applicable to all cases of multiplication and also in the division of a large number by another large number. It means vertically and crosswise. We discuss multiplication of two, 4 digit numbers with this method [8-9]. Ex.1. the product of 1111 and 1111 using *Triyakbhyam*(vertically and crosswise) is given below. Methodology of Parallel Calculation



VEDIC MULTIPLIER ARCHITECTURE :

The hardware architecture of 2X2, 4x4 and 8x8 bit Vedic multiplier module are displayed in the below sections. Here, “Urdhva-Tiryagbhyam” (*Vertically and Crosswise*) sutra is used to propose such architecture for the multiplication of two binary numbers. The beauty of Vedic multiplier is that here partial product generation and additions are done concurrently. Hence, it is well adapted to parallel processing. The feature makes it more attractive for binary multiplications. This in turn reduces delay, which is the primary motivation behind this work

VEDIC MULTIPLIER FOR 2X2 BIT MODULE:

The method is explained below for two, 2 bit numbers A and B where $A = a1a0$ and $B = b1b0$ as shown in Fig. 2. Firstly, the least significant bits are multiplied which gives the least significant bit of the final product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with, the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the final product and the carry is added with the partial product obtained by multiplying the most significant bits to give the sum and carry. The sum is the third corresponding bit and carry becomes the fourth bit Of the finel product. The 2X2 Vedic multiplier module is implemented using four input AND gates & two half-adders which is displayed in its block diagram in Fig. 3. It is found that the hardware architecture of 2x2 bit Vedic multiplier is same as the hardware architecture of 2x2 bit conventional Array Multiplier [2]. Hence it is concluded that multiplication of 2 bit binary numbers by Vedic method does not made significant effect in improvement of the multiplier‟s efficiency. Very precisely we can state that the total delay is only 2-half adder delays, after final bit products are generated, which is very similar to Array multiplier. So we switch over to the implementation of 4x4 bit Vedic multiplier which uses the 2x2 bit multiplier as a basic building block. The same method can be extended for input bits 4 & 8. But for higher no. of bits in input, little modification is required.

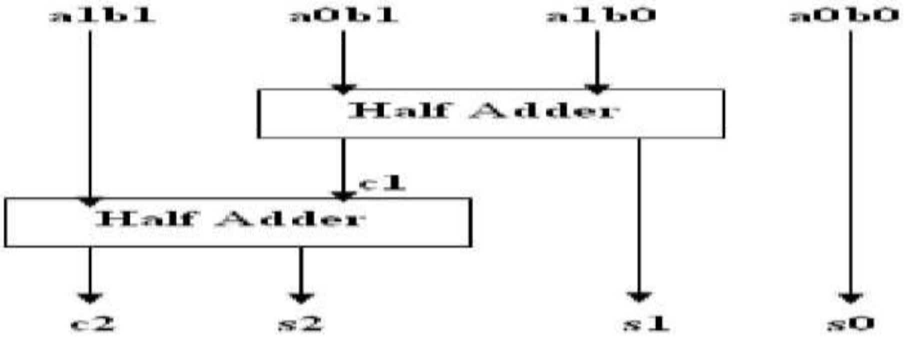


Fig. Block Diagram of 2x2 bit Vedic Multiplier

The 4x4 bit Vedic multiplier module is implemented using four 2x2 bit Vedic multiplier modules as discussed in Fig. 3. Let‟s analyze 4x4 multiplications, say A= A3 A2 A1 A0 and B= B3 B2 B1 B0. The output line for the multiplication result is – S7S6S5S4 S3 S2 S1 S0 .Let‟s divide A and B into two parts, say A3A2 & A1 A0 for A and B3 B2 & B1B0 for B. Using the fundamental of Vedic multiplication, taking two bit at a timeand using 2 bit multiplier block, we can have the following structure for multiplication as shown in Fig. 4.

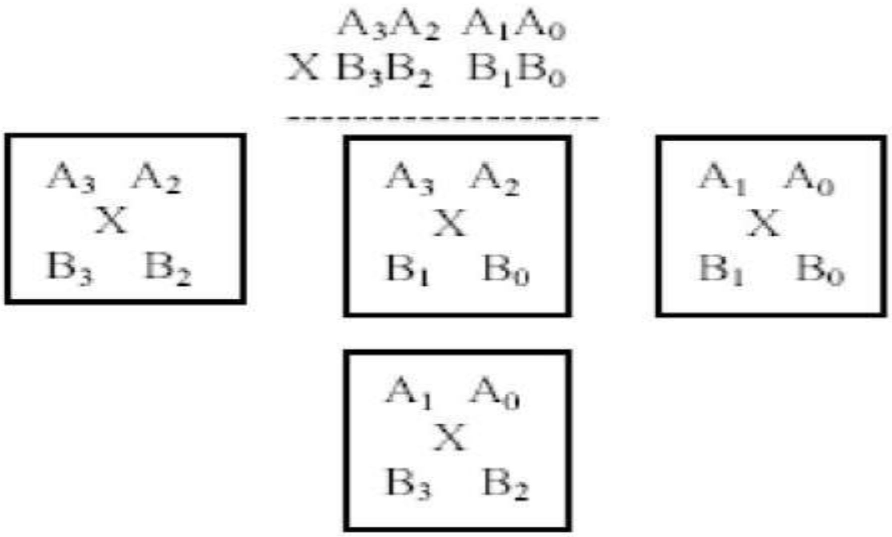


Fig. Sample Presentation for 4x4 bit Vedic Multiplication

Each block as shown above is 2x2 bit Vedic multiplier. First 2x2 bit multiplier inputs are A1A0 and B1B0. The last block is 2x2 bit multiplier with inputs A3 A2 and B3 B2. The middle one shows two 2x2 bit multiplier with inputs A3 A2 & B1B0 and A1A0 & B3 B2. So the final result of multiplication, which is of 8 bit, S7 S6S5S4 S3 S2 S1 S0. To understand the concept, the Block diagram of 4x4 bit Vedic multiplier is shown in Fig. 5. To get final product (S7 S6 S5 S4 S3 S2 S1 S0), four 2x2 bit Vedic multiplier (Fig. 3) and three 4-bit Ripple-Carry (RC) Adders are required. The proposed Vedic multiplier can be used to reduce delay. Early literature speaks about Vedic multipliers based on array multiplier structures. On the other hand, we proposed a new architecture, which is efficient in terms of speed. The arrangements of RC Adders shown in Fig. 5, helps us to reduce delay. Interestingly, 8x8 Vedic multiplier modules are implemented easily by using four 4x4 multiplier modules

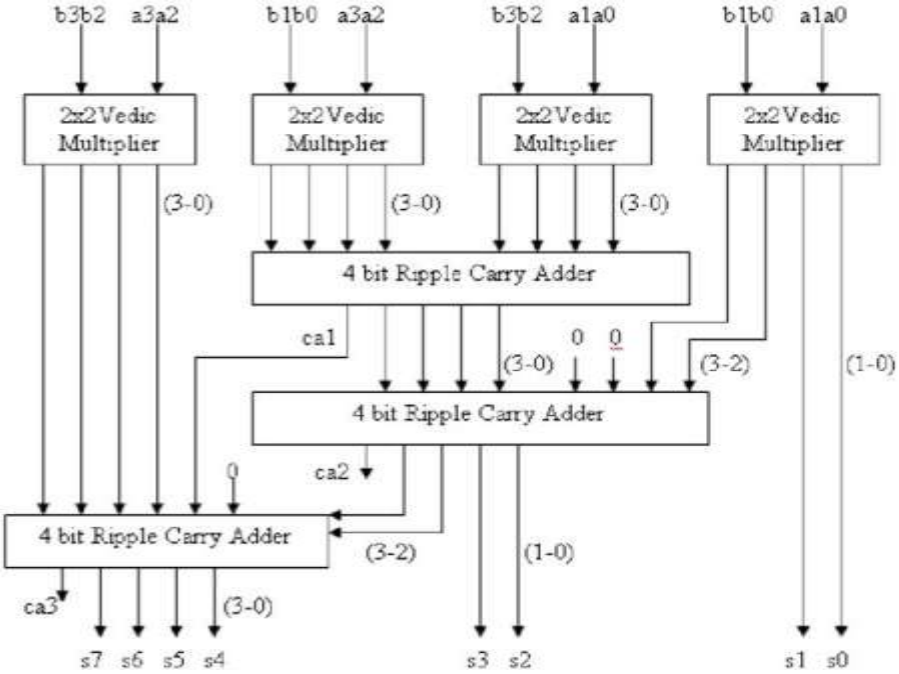
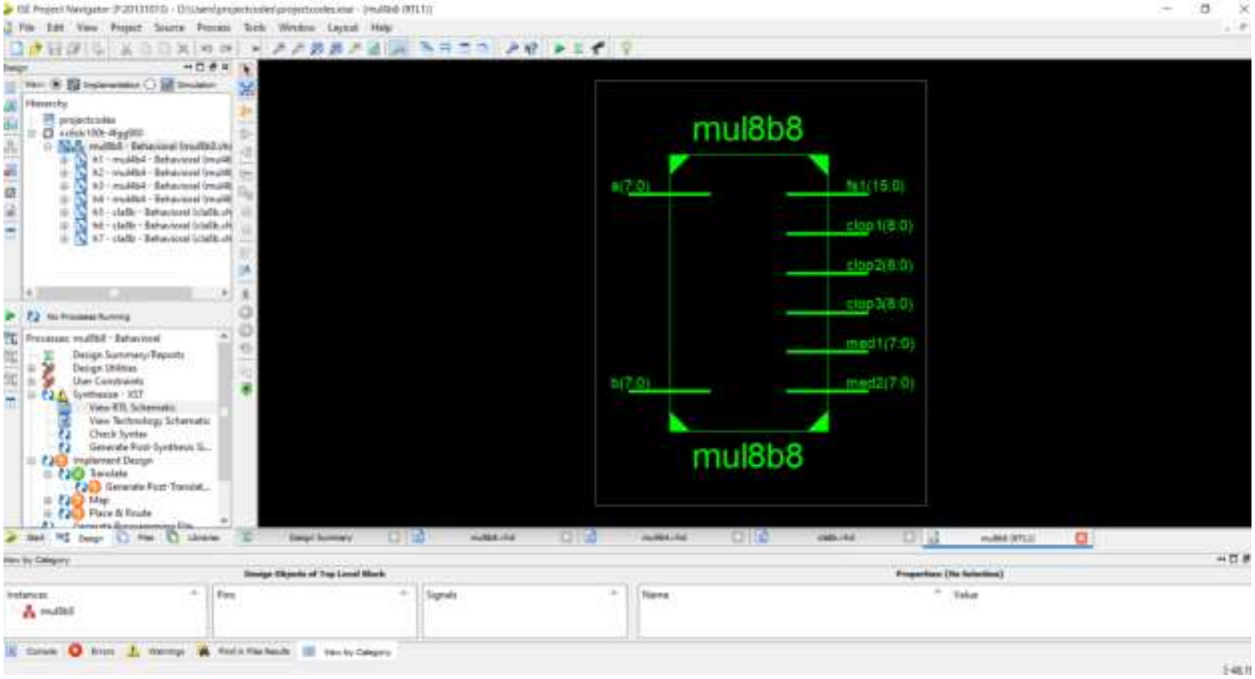
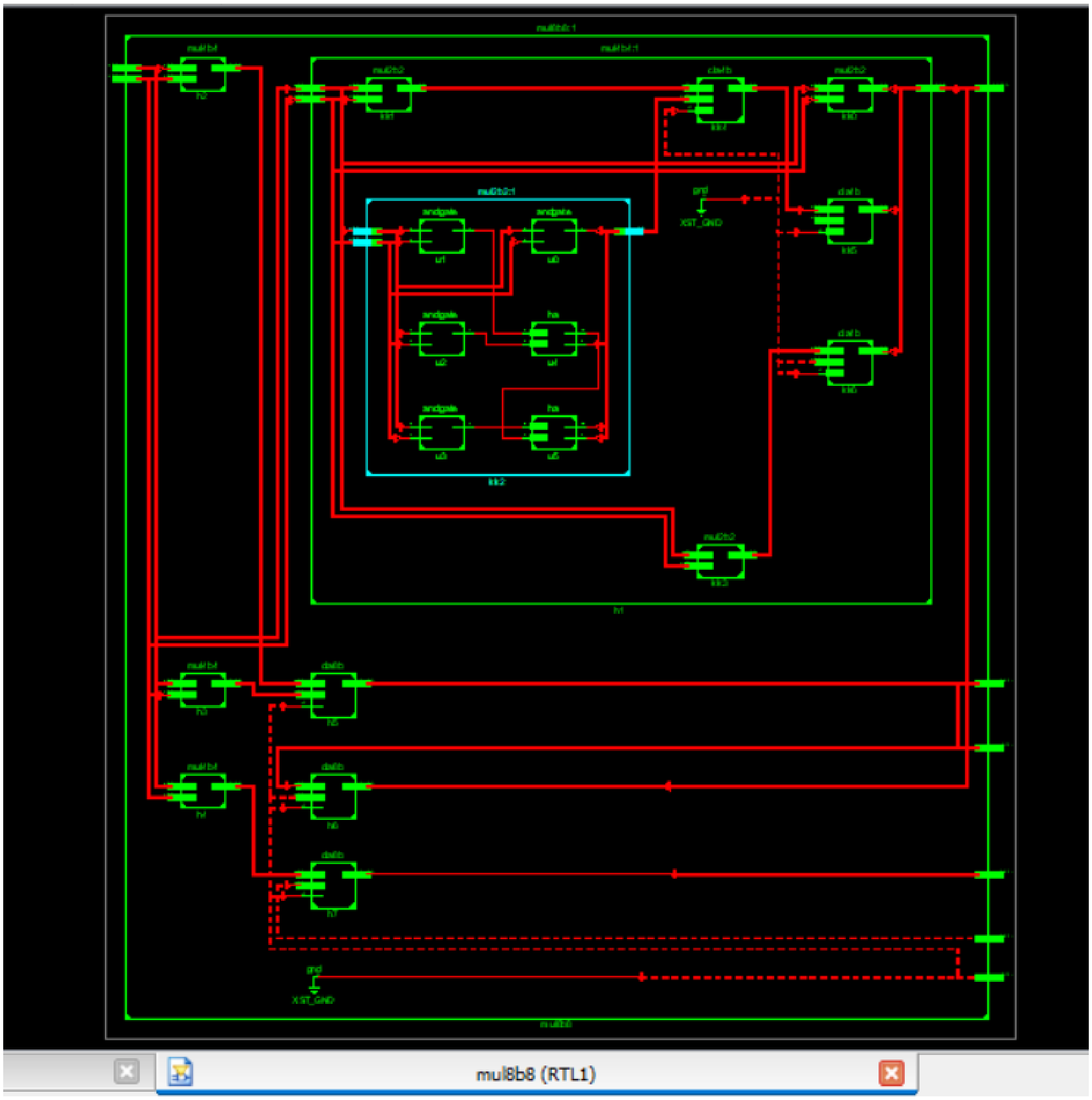
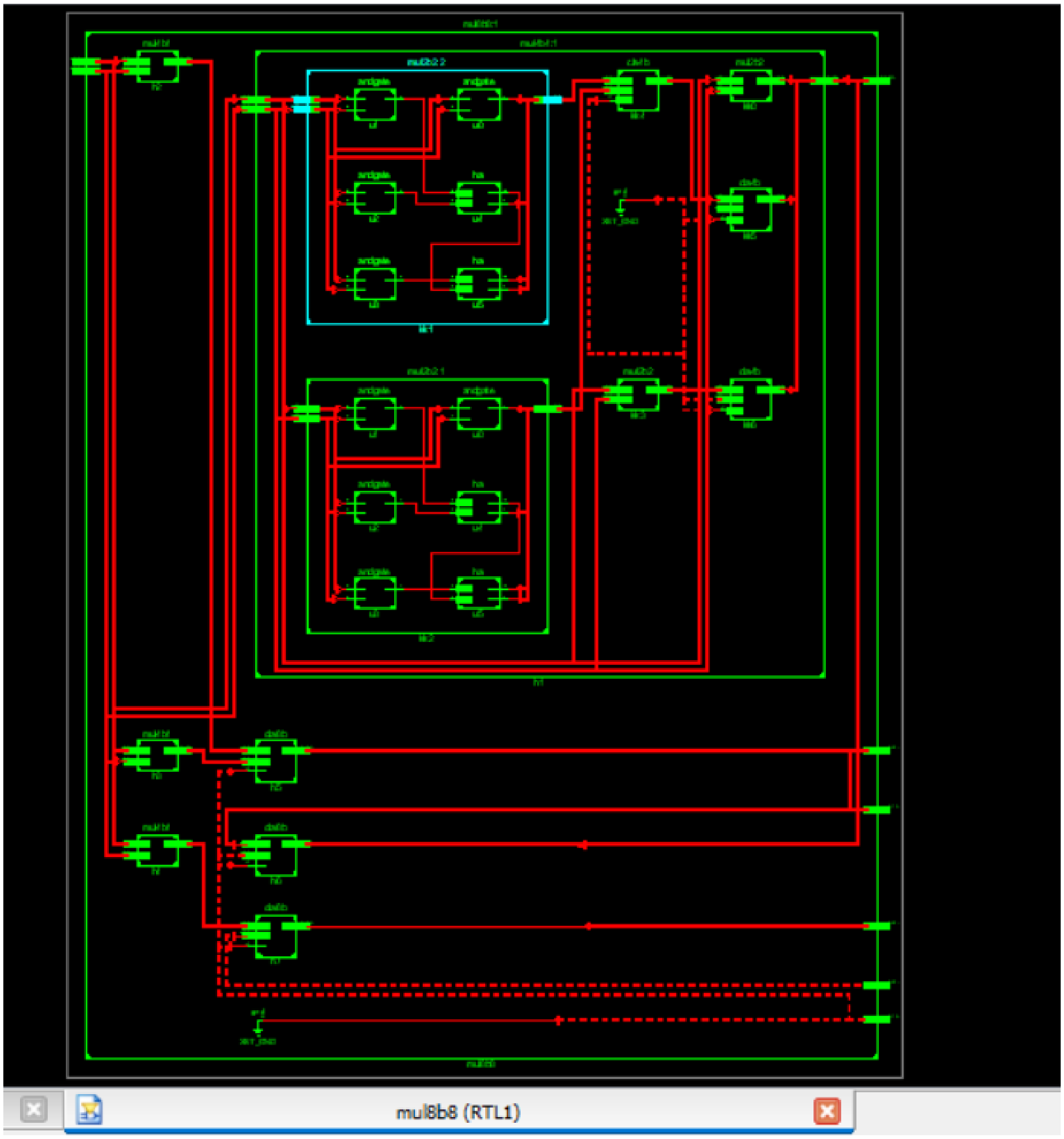


Fig. Block Diagram of 4x4 bit Vedic Multiplier

RESULT:




andgate
andgate
andgate
andgate
ha
ha
u1
u0
u2
u4
u3
u5
kk2
gnd
XST_GND
gnd
XST_GND
mul8b8 (RTL1)



CONCLUSION:

The performance of urdhavatriyagbhyam was investigated. The8X8-bit Vedic multiplier using urdhavatriyagbhyam sutra implementation uses less numbers of adders compared to 8X8 general multiplier using “digital electronics”. urdhavatriyagbhyam sutra reduces the multiplication of two large numbers into the multiplication of two small numbers and concatenation. Hence, there is significant reduction in the delay time which leads to improvement of speed and also the memory occupied is also reduced when using urdhavatriyagbhyam sutra.

REFERENCES :

[1] Jagadguru Swami, Sri BharatiKrisna, Tirthaji Maharaja, “Vedic Mathematics or Sixteen Simple Mathematical Formulae From the Veda, Delhi (1965)”, MotilalBanarsidas, Varanasi, India,

[2] M. Morris Mano, “Computer System Architecture”, 3rd edition, Prientice-Hall, New Jersey, USA, 1993, pp. 346-348.

[3] H. Thapliyal and H.R Arbania. “A Time-Area-Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics”, Proceedings of the 2004 International Conference on VLSI (VLSI‟04), Las Vegas, Nevada, June 2004, pp. 434-439.

[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] Harpreet Singh Dhillon and Abhijit Mitra, “A Reduced– Bit Multipliction Algorithm for Digital Arithmetics” International Journal of Computational and Mathematical Sciences

[7] Honey Durga Tiwari, GanzorigGankhuyag, Chan Mo Kim and Yong Beom Cho, “Multiplier design based on ancient Indian Vedic Mathematician”, International SoC Design Conference, pp. 65- 68, 2008.