

FPGA Implementation of 8-bit Multiplier with Reduced Delay Time

Dhanabalan and Tamil Selvi

Abstract—This paper proposes a design method for an 8-bit multiplication with reduced delay time. Normally, two numeric data can be multiplied by repeated addition. In case of binary multiplication, combinational circuit can be designed using manual multiplication method which requires binary addition. Carry generated because of addition affects the speed of multiplication since the present addition depends on the value of previous carry. To overcome this problem, addition with the help of multiplexer is introduced and the result is an increased speed in multiplication. Even though the proposed design is mainly for FPGA implementation, it can also be implemented in ASIC as the logical delay is reduced when compared the result in Xilinx device.

Index Terms—Multiplexer, delay time, incrementer, FPGA.

I. INTRODUCTION

Multiplication is generally used in almost all applications or designs like digital signal processing [1], image processing, embedded system, design of Arithmetic Logic Unit etc., [2]. In this fast moving life, researchers are competing with the speed of technology. When a typical algorithm is implemented in FPGA, it provides better execution speed than the one which is implemented in a DSP or any other processor. It is because; processor utilizes its hardware architecture through sequential instructions. Whereas, FPGA activates its hardware at the same time to do a job. Hence, many of the applications are being replaced by FPGA nowadays. VLSI technology is categorized into ASIC or FPGA implementation. FPGA is chosen widely since it has the flexibility of implementing different types of applications in a single chip [3]. Type of an application can be changed by writing HDL program into it.

Adder is the basic circuit that is needed to do multiplication. Efficiency of multiplication automatically improves when the efficiency of adder is increased. In case of VLSI design, efficiency speaks in terms of area, speed & power. Since there is a trade-off among these parameters, only one of the parameter is considered for optimization and the other two parameters shall be brought up to the satisfactory level [4]. Delay time of multiplexer is reduced in this paper since many applications require high speed operation. Delay time of adder circuit mainly depends on processing carry generation when two bits are added. In ripple carry adder, the present addition waits for its previous

addition to generate carry. This increases delay time with increase in size of addition. Carry look-ahead adder is the best adder when speed is given more importance.

II. ARRAY MULTIPLIER

Based on performance comparison [5], [6], array multiplier is selected for the purpose of evaluation. Fig. 1 shows a 5×5 multiplication. In array multiplier, multiplication is achieved in three steps. In the first step, each multiplicand bit is logically AND with each multiplier bit. This logical AND operation can be done simultaneously. In the second step, partial product in each row is shifted by its row position minus one. In the third step, partial products are added to get the end result. Now, this addition can be performed depending on designer's choice. Effective design of adder circuit decides multiplier efficiency. A 5×5 multiplier requires, 20 AND gates, 15 shifters and 16 full adders. It is also possible to perform multiplication with repeated addition using a single adder circuit. Hardware requirement will be further increased if the designer chooses carry look-ahead adder. However, pipeline method of multiplication reduces delay time. Logical and in partial product is also responsible for increase in delay time. Delay time logarithmically varies in proportion to the bit size of multiplicand and multiplier.

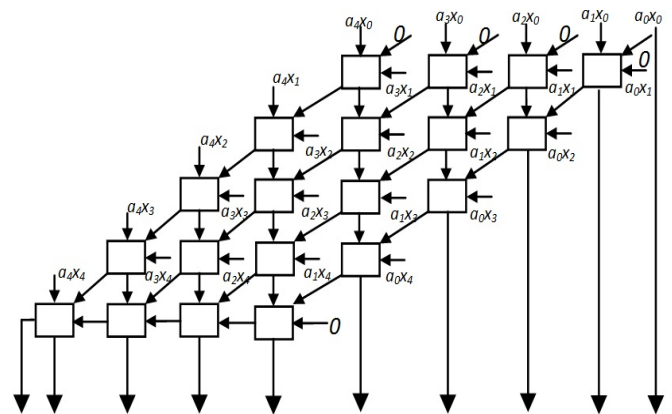


Fig. 1. Array Multiplier Block diagram

III. MULTIPLIER DESIGN

Multiplication of two binary numbers involve logical and OR operation. As in array multiplier, partial products are obtained by logically AND in each bit of multiplicand with each bit in multiplier [7]. After generating product terms, addition is done using suitable adder circuit. Now, in this design the product terms are grouped together so that all the

Manuscript received December 28, 2012; revised July 5, 2013.

G. Dhanabalan is with the Kamaraj College of Engineering and Technology, Virudhunagar, India (e-mail: dhanabalang@yahoo.com).

S. Tamil Selvi is with National Engineering College, Kovilpatti, India (e-mail: tamilgopal2004@yahoo.co.in).

addition operations are done [8] simultaneously as shown in Fig. 2(a). r_1, r_2, r_3 & r_4 are the resultant additions. These results are again grouped for addition to get x_1 and x_2 as shown in Fig. 2(b). The final result from Fig. 2(a) is x_1, x_2 .

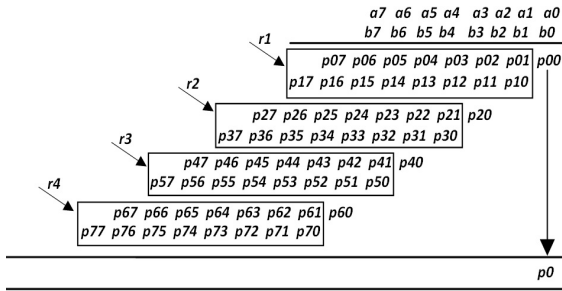


Fig. 2 (a) The final result is achieved after adding x_1 and x_2 .

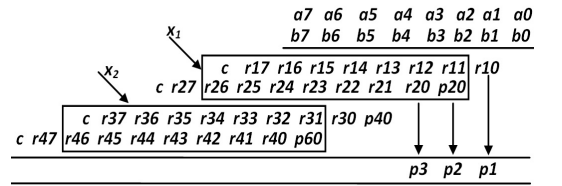


Fig. 2 (b)

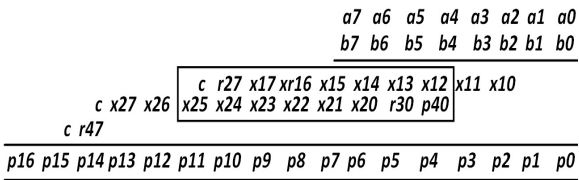


Fig. 2 (c)

Fig. 2. (a), (b) & (c) Different stages of multiplication

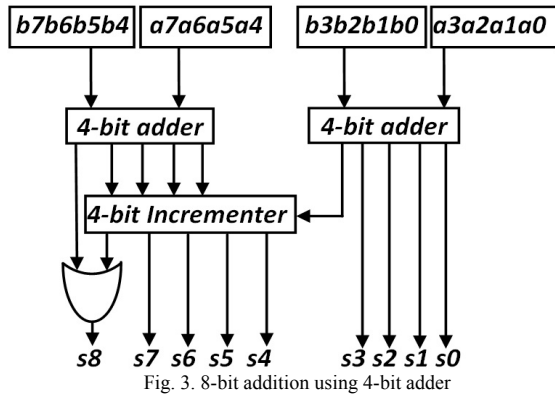


Fig. 3. 8-bit addition using 4-bit adder

Because of parallel addition, this design requires seven 8-bit additions for 8 bit multiplication.

A. Adder Using Multiplexer

Addition is achieved in three stages. In the first stage, 8 bit data is divided into two 4 bit data and added simultaneously using two 4-bit adder. In the second stage, Sum from the addition of a_0-a_3 & b_0-b_3 and carry from the addition of a_4-a_7 & b_4-b_7 are added using 4 bit incrementer. Carry from incremented & carry from the addition of a_4-a_7 & b_4-b_7 are logically ORed to get final carry in the third stage.

Adder circuit using multiplexer is designed from the adder truth table. For an 8-bit addition, truth table will have 65,536 rows. In this design, we have segregated 8-bit addition into two 4-bit addition and hence the truth table will have 256 rows only. Six inputs are used as selector lines. Hence 64:1

multiplexer is designed as a 4-bit adder. By having two 4 bit adders implemented into FPGA, 8-bit addition is done with lesser delay time as shown in Fig. 3.

Multiplexer design for four bit addition is derived from truth table as shown in Table I.

TABLE I: 4-BIT ADDER - TRUTH TABLE

Input a				Input b				carry	sum			
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	1	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	1	0	1	0	0	1	0	1
0	0	0	0	0	1	1	0	0	0	1	1	0
0	0	0	0	0	1	1	1	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0

IV. DESIGN IMPLEMENTATION AND VERIFICATION

The design is written in verilogHDL program and verified for correct output using ModelSim-Altera 6.6c simulation software. Figure 4 shows simulated result for sample data ($0d \times 15, 8d \times 55$ & $ff \times ff$). Using test bench, outputs for all the possible combination of inputs are verified.

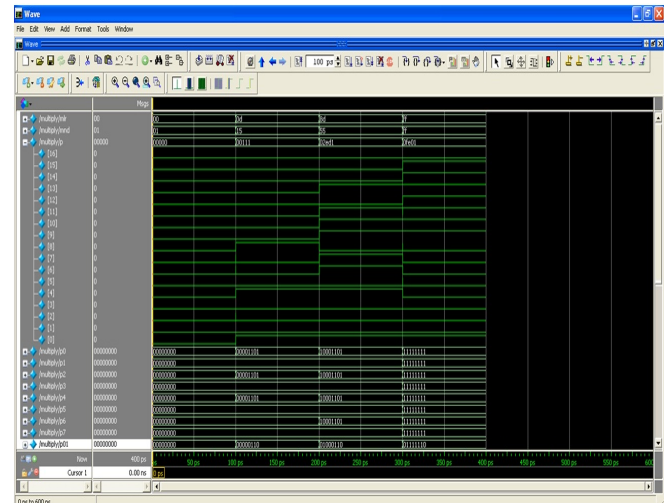


Fig. 4. Simulated output

For implementing multiplier into hardware, we used Spartan 3E FPGA board [9]. Synthesize and comparison is done using the target device xc5vlx30-3-ff324. The Cell usage is shown in Table II.

TABLE II: CELL USAGE

LUTs & IO Buffers	In numbers
LUT2	4
LUT3	13
LUT4	13
LUT5	41
LUT6	62
MUXF7	10
IBUF	16
OBUF	17

To have comparison, multiplication is done using * symbol in program for the same target device. Cell usage for

this design is shown in Table III.

TABLE III: CELL USAGE

LUTs/IO Buffers/Other Cells	In numbers
DSP48E	1
IBUF	16
OBUF	17

The RTL & Technological schematic of proposed multiplier is shown in Fig. 6, 7, 8 & 9.

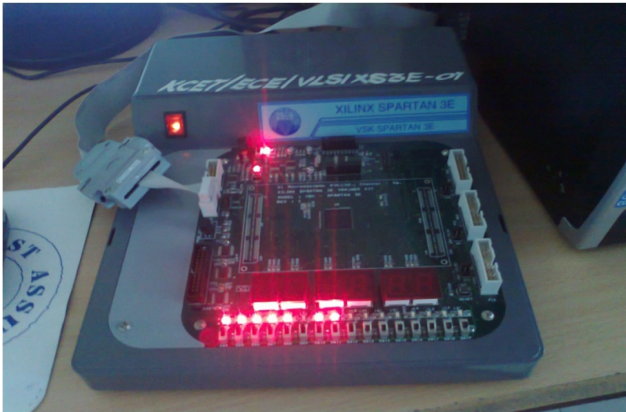


Fig. 5. Multiplier implemented in Spartan 3E board

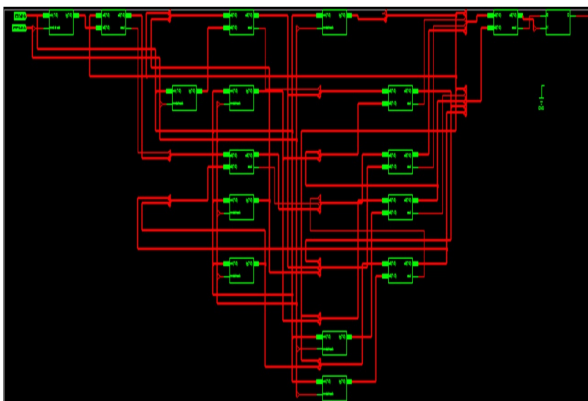


Fig. 6. RTL Schematic of multiplier

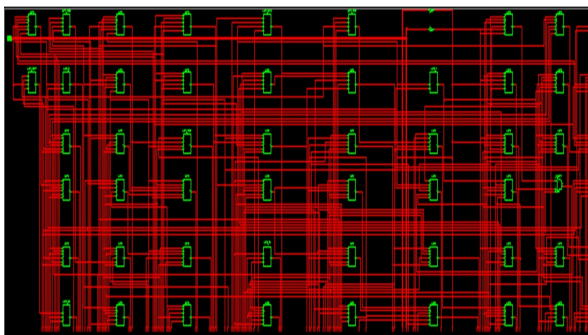


Fig. 7. Technological schematic of multiplier

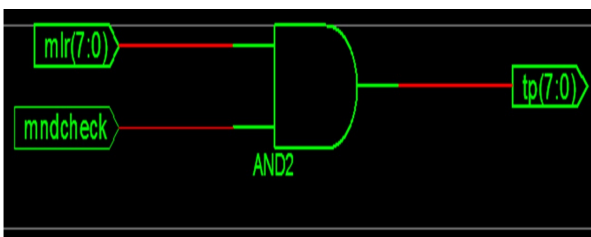


Fig. 8. RTL schematic of and gate

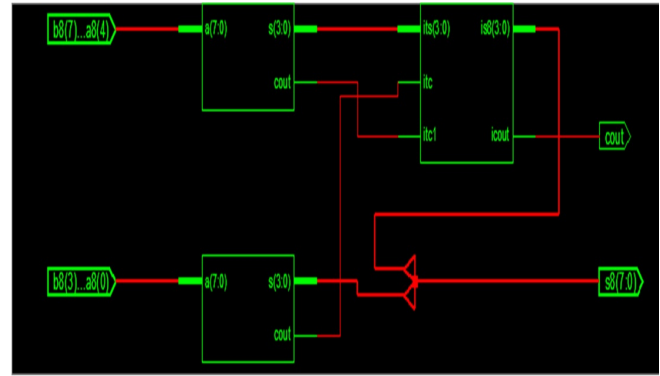


Fig. 9. RTL Schematic of adder

V. EVALUATION OF THE PROPOSED DESIGN

Array multiplier performs better with reference to other multipliers. Its delay time is 14.4ns. Delay time of proposed multiplier when synthesized using Virtex5 FPGA is 12.294ns. Hence the proposed design is able to execute multiplication faster than array multiplier.

The proposed design is also compared with the multiplier that is included as the standard cell by the vendor [10] – [11]. It is shown in Table IV. Delay time in the proposed design is higher than standard cell multiplier. Delay time is the addition of logic delay and route delay. Now, logic delay in proposed design is 1.525 ns less than standard cell multiplier. If logic delay of proposed design and route delay of standard cell multiplier is combined, then delay time would be than 5.622ns.

TABLE IV: DELAY TIME - COMPARISON

Device	Logic delay	Route delay	Delay time
Virtex 5 Proposed design	3.670ns	8.624ns	12.294ns
Virtex 5 Using DSP48E	5.195ns	0.427ns	5.622ns

VI. CONCLUSION

In this paper, we concentrated on how to increase the speed of multiplication. Logic delay in this design is reduced. Hence, the design can also be looked for ASIC implementation. In case of FPGA implementation, route delay is higher than the implementation using DSP48E. Route delay can be reduced with the help of an efficient algorithm. Defining a mechanism for routing algorithm in FPGA may pave way for reduced route delay and hence delay time.

REFERENCES

- [1] G. A. Ruiz and M. Granda “An area-efficient static CMOS carry-select adder based on a compact carry look-ahead unit,” *Microelectronics Journal*, pp. 939-944, 2004.
- [2] H. Boukadida, N. Hassen, Z. Gafsi, and K. Besbes, “A highly time-efficient digital multiplier based on the A2 binary representation,” *International Journal of Engineering and Technology*, vol. 3, pp. 4498–4509, May 2011.
- [3] B. Cope. Implementation of 2D convolution on FPGA, GPU and CPU. [Online]. Available: cas.ee.ic.ac.uk/people/btc00/index_files/convolution_filter.pdf,
- [4] M. H. Rais and M. H. Al Mijalli, “Virtex-5 FPGA Based Braun’s multipliers,” *International Journal of Computer Science and Network Security*, vol.11, no. 8., pp. 81–84, August 2011.

- [5] K. Mohammed and S. Agaian, "Efficient FPGA implementation of convolution," in *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pp. 3478–3483, October 2009.
- [6] S. R. Kuang and J. P. Wang, "Low-error configurable truncated multipliers for multiply-accumulate applications," *Electronics Letters*, vol. 42, no. 16, pp. 905-909, 2006.
- [7] Z. Huang and M. D. Ercegovic, "High-Performance low-power left-to-right array multiplier design," *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 272–283, Mar 2005.
- [8] C. N. Marimuthu and P.Thangaraj, "Low power high performance multiplier," *International Journal of Programmable Devices, Circuits and Systems (PDCS), International Congress for global Science and Technology (IGCST)*, vol. 8, pp. 31–38, December 2008.
- [9] M. H. Rais and M. H. A. Mijalli, "FPGA based fixed width 4x4, 6x6, 8x8 and 12x12 bit multipliers using Spartan-3AN," *International Journal of Computer Science and Network Security*, vol. 11, no. 2, pp. 61-68, June 2011.
- [10] G. L. Narayanan and B. Venkataramani, "Optimization Techniques for FPGA-Based Wave Pipelined DSP Blocks," *IEEE Trans. Very Large Integr. (VLSI) Syst.*, vol. 13. No. 7, pp. 783 – 792, July 2005.
- [11] B. H. Boukadida, Z. Gafsi, N. Hassen, and K. Besbes, "A 4-Bit CSA Adder using the arithmetic A2 redundant binary representation for mixed neural networks with On-Chip Learning," in *Proc. of 7th IEEE Computer Information Systems and Industrial Management Applications*, pp. 97–98, May 2008.



G. Dhanabalan received his B.E in instrumentation & control engineering from Madurai Kamarajar University, Madurai in 1994. He completed his M.E. in VLSI Design at Anna University, Chennai in 2007. Since 2010 he is doing Ph.D., in Anna University, Chennai.

Presently, he is working as an assistant professor / HOD, Department of EIE in Kamaraj College of Engineering and Technology, Virudhunagar. He has rich experience in installation & commission of instrumentation projects during the period 1994 - 2003.



S. Tamil Selvi received her B.E in electronics and communication engineering from Madurai Kamarajar University, Madurai in 1988. In 1997, She received her M.E. degree with specialization in optical communication from College of Engineering, Guindy, Anna University, Chennai. In 2009, she completed her Ph.D., in wireless communication at Manonmaniam Sundaranar University, Tirunelveli.

She is now working as a professor in National Engineering College, Kovilpatti. She has published more than 10 research papers, organized and chaired conference sessions, presented overview lectures. Her research interests include digital communications, signal/image processing, VLSI.

She is the member of IEEE, life member of ISTE, CSI, Fellow of IE (I) and IETE India.