

Optimization Design of Partition Multiplier based on Brent Kung Adder

Shambhavi Shukla, Dr. Vibha Tiwari

M. Tech. Scholar, Professor

Department of Electronics and Communication Engineering

TIT, Bhopal

Shambhavi.shukla266@gmail.com

ABSTRACT- In this paper a partition multiplier based on Brent Kung adder is presented. Partition multiplier is used in various applications i.e. wireless communication, image processing and digital signal processing. We have design area efficient partition multiplier with the help of Brent kung adder. All design is implemented VHDL platform and Xilinx 14.2 software. The Brent-Kung adder is a parallel prefix adder (PPA) and is modified form of carry-look ahead adder (CLA). It acquainted higher normality with the structure of the adder and has less wiring issues, reduces complexities, provides better execution and less chip region which is not the case with the Kogge-Stone adder (KSA). It is also very faster than ripple-carry adders (RCA).

Keywords: - Brent Kung Adder, Vedic Multiplier, Delay, Slice Flip Flop, Carry Look Ahead Adder

I. INTRODUCTION

Image processing is one of the most prominent medium of processing an image and uses it on various technologies for utilizing it in the best possible way. Since a long time, image processing has been a very influential topic and it is developing drastically day by day. In different genre, this image processing technology is used for more advancement and effectiveness. That's why we try to make this process more efficient with the help of multi-threading which is a famous process of multi-tasking and make the image processing as fast as possible on the factor of time. We have also included the occurrence of parallel processing which help the multi-threading work more efficiently. Because in image processing time is a very important factor as delay of time and delay in arrival of frame may cause a lot of problem in the final result of the process.

The FIR filter is also termed as recursive filter. The literal meaning for recursive is running back and technically refers, the previously calculated outputs are going back to the latest output. The every stage of FIR filter uses previous output values in addition to input values for processing filtering operation. Hence, it is called as recursive filter. Therefore, in FIR filter, previous input values are stored in the processors memory. From above consecution, it is clear that more calculation is required to perform the process of FIR filter, since the filter expression consist of previous output terms as well as present input

terms. There are two types of FIR filters widely used. One is direct form FIR filter and other is transposed FIR filter. Parallel processing DSP is a technique of duplication function units to operate different tasks simultaneously. Each task can be operated in different function units. Parallel processing is completely different from pipelining process. Pipelining process leads to reduction in critical path, which can increase the sample speed or reduce power consumption at the same speed. The parallel processing technique requires multiple fan-outs, which are computed in parallel and received outputs of every task within a single clock cycle. Therefore, effective sampling period is increased by the level of parallelism.

II. MULTIPLIER

Multipliers play an important role in today's digital signal processing and various other applications. Essential design targets of multiplier include high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier are required thereby making them suitable for various VLSI implementations. The straightforward way to implement a multiplication is based on an iterative adder-accumulator for the generated partial products as shown in fig. 1. This multiplier is called a serial multiplier. However, this solution is quite slow as the final result is only available after 'n' clock cycles, 'n' is the size of the operands. Serial multipliers are used where area and power are of the most importance and increased delay can be tolerated. A faster version of the iterative multiplier should add partial products at once. This could be achieved by unfolding the iterative multiplier and yielding a combinational circuit that consists of several partial product generators together with several adders that operate in parallel.

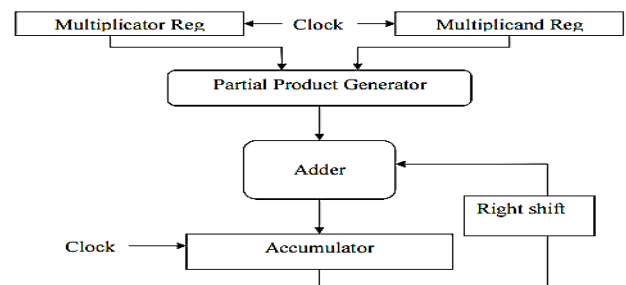


Fig. 1: Iterative multiplier

This multiplier is called Parallel Multiplier and is shown in fig. 2. Consider two unsigned binary numbers X and Y that are M and N bits wide, respectively. To introduce the multiplication operation, it is useful to express X and Y in the binary representation

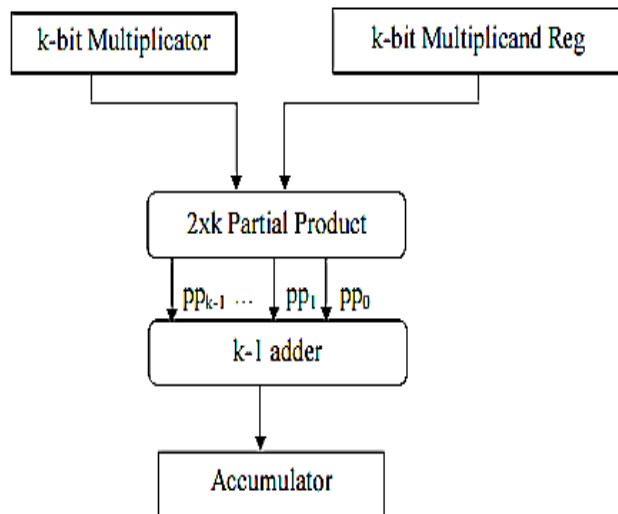


Fig. 2: Parallel multiplier

$$X = \sum X_i 2^i \quad i = 0 \text{ to } M \quad (1)$$

$$Y = \sum Y_j 2^j \quad j = 0 \text{ to } N \quad (2)$$

The multiplication operation is then defined as follows:

$$\begin{aligned} Z &= X \times Y \\ &= \sum Z_k 2^k \quad k = 0 \text{ to } M + N - 1 \\ &= (\sum X_i 2^i) (\sum Y_j 2^j) \\ &= \sum (\sum X_i Y_j 2^{i+j}) \end{aligned} \quad (3)$$

The simplest way to perform a multiplication is to use a single two input adder. For inputs that are M and N bits wide, the multiplication tasks M cycles, using an N-bit adder. This shift –and–add algorithm for multiplication adds together M partial products. Each partial product is generated by multiplying the multiplicand with a bit of the multiplier – which, essentially, is an AND operation – and by shifting the result in the basis of the multiplier bit's position. Similar to the familiar long hand decimal multiplication, binary multiplication involves the addition of shifted versions of the multiplicand based on the value and position of each of the multiplier bits. As a matter of fact, it's much simpler to perform binary multiplication than decimal multiplication. The value of each digit of a binary number can only be 0 or 1, thus, depending on the value of the multiplier bit, the partial products can only be

a copy of the multiplicand, or 0. In digital logic, this is simply an AND function.

III. METHODOLOGY

a. Partition Multiplier

We consider two N-bit operands $x_{N-1}, x_{N-2}, \dots, x_2, x_1, x_0$ and $y_{N-1}, y_{N-2}, \dots, y_2, y_1, y_0$ for N by N multiplier, the partial products of two N-bit numbers are $x_i y_j$ where i, j go from 0, 1, ..., N-1. The longest two columns in the middle of the partial products contribute to the maximum delay. We then proceed to sum up each column of the two parts in parallel. Partition Multiplier Method: - Let M and N be numbers of b-bits such that multiplication of the inputs M and N gives the 2b bits product. Here M and N are split into t numbers as $M_0, M_1, M_2, \dots, M_{t-1}$ and

$N_0, N_1, N_2, \dots, N_{t-1}$ each consisting of r bits

Where

$$t = \frac{b}{r} \quad (4)$$

Consider the value of $b = 8, r = 2$

Then

$$t = 4 \quad (5)$$

M and N can therefore be expressed as

$$M_0 = M \left(\frac{n}{2} - 1 \text{ to } 0 \right) \quad (6)$$

$$M_1 = M \left(n - 1 \text{ to } \frac{n}{2} \right) \quad (7)$$

$$N_0 = N \left(\frac{n}{2} - 1 \text{ to } 0 \right) \quad (8)$$

$$N_1 = N \left(n - 1 \text{ to } \frac{n}{2} \right) \quad (9)$$

b. Brent Kung Adder

The Brent–Kung adder is a parallel prefix adder (PPA) and is modified form of carry-look ahead adder (CLA). Proposed by Richard Peirce Brent and Hsiang Te Kung in 1982 it acquainted higher normality with the structure of the adder and has less wiring issues, reduces complexities, provides better execution and less chip region which is not the case with the Kogge–Stone adder (KSA). It is also very faster than ripple-carry adders (RCA).

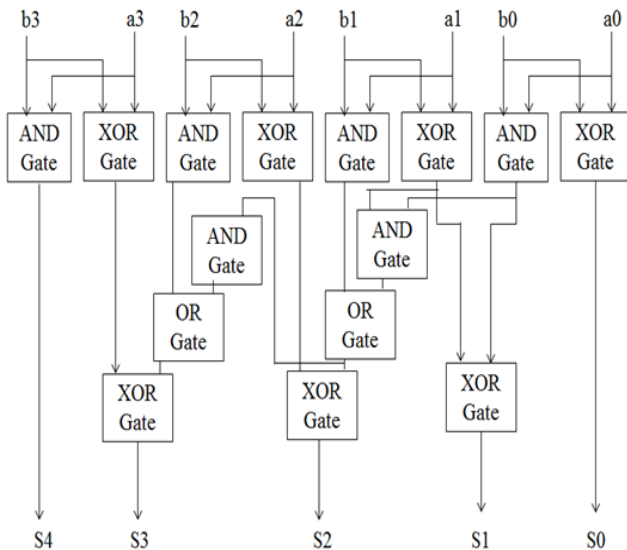


Fig. 3: Block Diagram of Brent Kung Adder

IV. IMPLEMENTATION METHODOLOGY

Following are the equations used to design the two parallel FIR filter with two inputs $x(2k)$, $x(2k+1)$ and two outputs $y(2k)$, $y(2k+1)$. For implementing this filter three FIR sub-filter blocks has been used as compare to traditional two FIRs sub-block filter, having length $N/3$. Two of three sub-filters $2k$ and $2k+1$ are having symmetric coefficient which reduces the number of multiplier and adders. Here two preprocessing and four post-processing adders have been used along with delay equipment. The symmetric sub-filter block has been implemented at the cost of two additional adders among those one is pre-processing and other one is post-processing for $L=2$.

Digital Signal Processing (DSP) operations are increasingly being implemented on Field Programmable Gate Array (FPGA) platform.

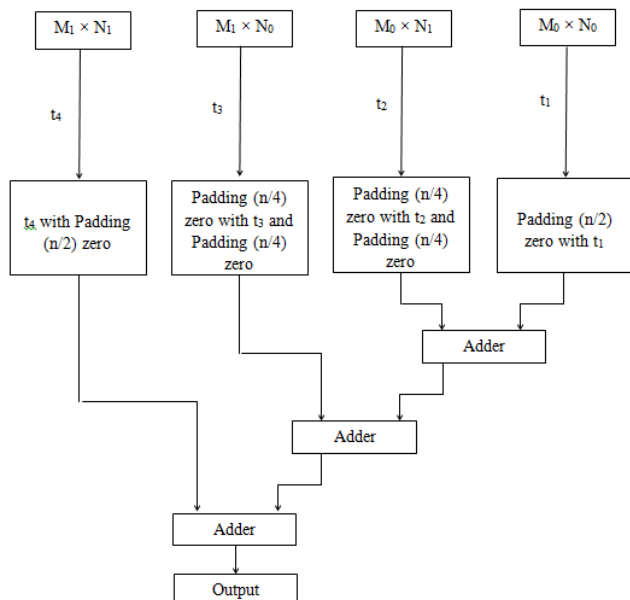


Fig. 4: Flow chart of proposed partition multiplier method

FPGA supports all signal processing operations like audio processing, video processing, filtering, frequency transformations and so on. On the other hand, Application Specific Integrated Circuits (ASICs) are also used to implement the signal processing operations. ASIC level of design is used to implement only specific application. Hence, flexibility and configurability is not possible in ASIC level of design. Therefore, FPGA is better than ASIC by all means. The analog operations can be implemented in FPGA by compiling Verilog Hardware Description Language (HDL) code for corresponding operation into register transfer logic (RTL) net list. In addition, synthesis process is used to produce bits which control logic gates and fills the registers and memories in an FPGA.

V. SIMULATION RESULT AND DISCUSSION

All the designing and experiment regarding algorithm that we have mentioned in this paper is being developed on Xilinx 14.2i updated version. Xilinx 9.2i has couple of the striking features such as low memory requirement, fast debugging, and low cost. The latest release of ISETM (Integrated Software Environment) design tool provides the low memory requirement approximate 27 percentage low. ISE 14.2i that provides advanced tools like smart compile technology with better usage of their computing hardware provides faster timing closure and higher quality of results for a better time to designing solution. ISE 14.1i Xilinx tools permits greater flexibility for designs which leverage embedded processors. The ISE 14.2i Design suite is accompanied by the release of chip scope ProTM 14.2i debug and verification software. By the aid of that software we debug the program easily. Also included is the newest release of the chip scope Pro Serial IO Tool kit, providing simplified debugging of high-speed serial IO designs for Virtex-4 FX and Virtex-5 LXT and SXT FPGAs. With the help of this tool we can develop in the area of communication as well as in the area of signal processing and VLSI low power designing.

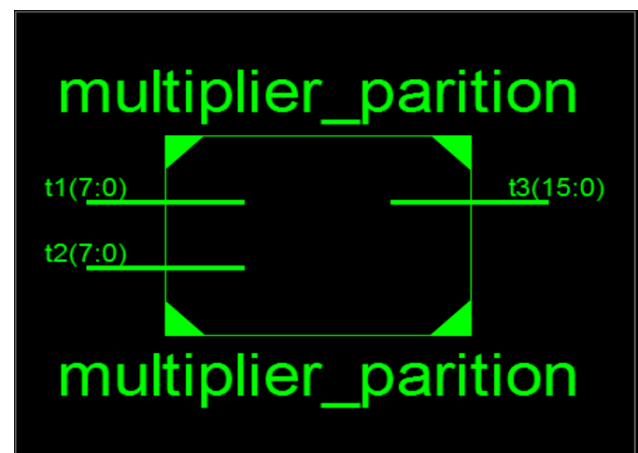


Fig. 5: View Technology Schematic of Partition Multiplier

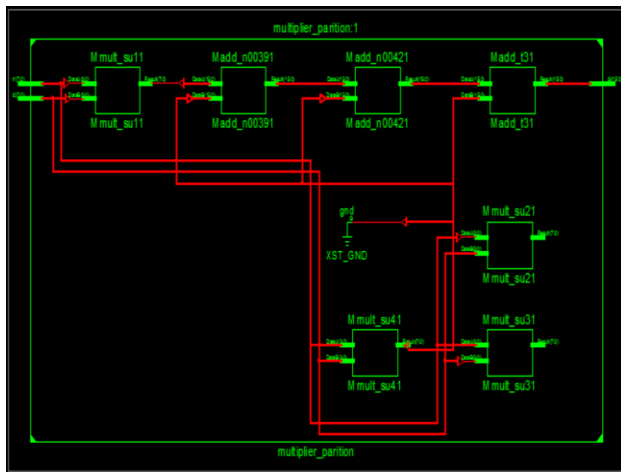


Fig. 6: RTL View of Partition Multiplier

Device utilization summary:

Selected Device : 4vfx12sf363-12

Number of Slices: 40 out of 5472 0%
 Number of 4 input LUTs: 69 out of 10944 0%
 Number of IOs: 32
 Number of bonded IOBs: 32 out of 240 13%
 Number of DSP48s: 3 out of 32 9%

Timing Summary:

Speed Grade: -12

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: 15.081ns

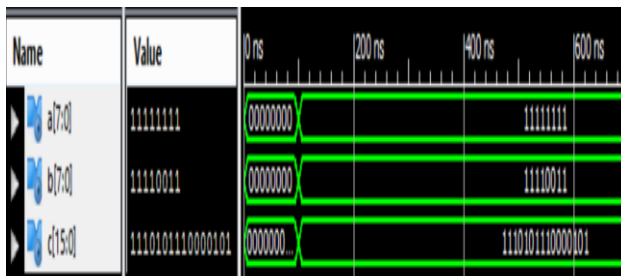


Fig. 7: VHDL Test Bench of Partition Multiplier

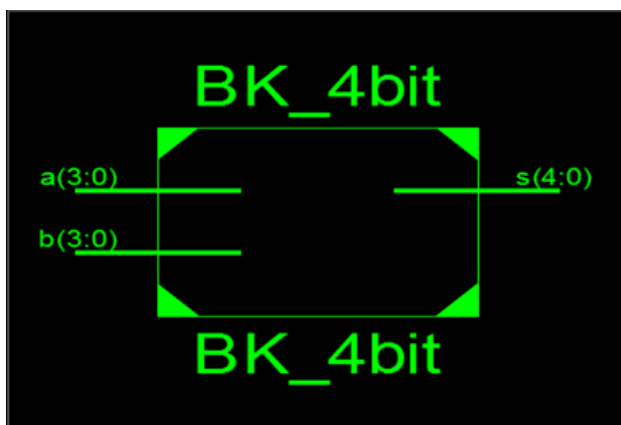


Fig. 8: VTS for 4-bit BK Adder

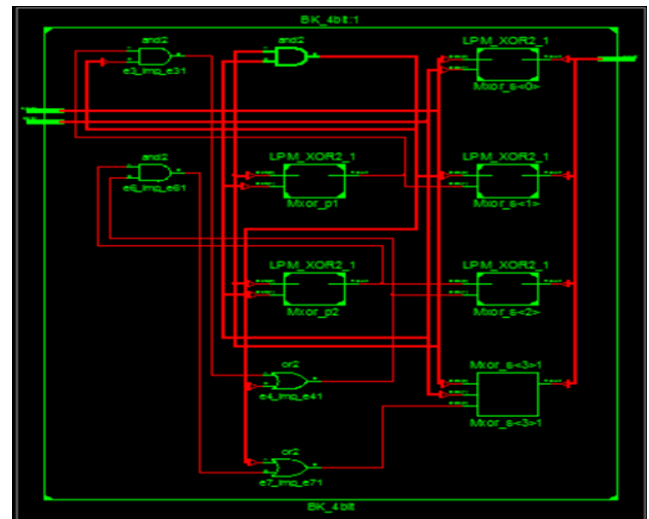


Fig. 9: RTL View of 4-bit BK Adder



Fig. 10: VHDL Test Bench of 4-bit BK Adder

Table I: Comparisons result for implemented design and existing algorithm

Design	Width	Area Gate court
Ripple Carry Adder	4	52
	8	104
	16	208
	32	416
CSLA Adder	4	64
	8	128
	16	256
	32	512
Compressor based Adder	4	45
	8	90
	16	180
	32	360
Implemented Brent Kung Adder	4	43
	8	86
	16	172
	32	344

Table II: Comparison Result of Previous Multiplier and Proposed Multiplier

Bit	Previous Vedic Method [1]	Proposed Partition Method
4-bit	10.43 ns	9.134 ns
8-bit	15.22 ns	11.631 ns
16-bit	19.58 ns	13.816 ns
32-bit	22.32 ns	16.234 ns

VI. CONCLUSION

In previous design, carry look ahead adder was used for designing partition multiplier but it was having the drawback that it was not working for every bit and large circuit complexity. So to remove this drawback, we have used Brent Kung adder which is an advanced binary adder. Its advantage is that it reduces the cost and the complexities of wire and is much quicker than Ripple Carry adder and carry look ahead adder. So it provides better performance and less area to implement in comparison to Kogge Stone adder. The main advantage of Brent Kung adder is that it works on every bit and consumes less space.

REFERENCES

- [1] Ranjan Kumar Barik, Manoranjan Pradhan, Rutuparna Panda, "Time efficient signed Vedic multiplier using redundant binary representation", Journal Engineering 2017, Vol. 2017, Iss. 3, pp. 60–68.
- [2] Basant Kumar Mohanty, and Pramod Kumar Meher, "High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 78, No.06, April 2016.
- [3] Omid Akbari, Mehdi Kamal, Ali Afzali-Kusha, and Massoud Pedram, "Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 34, Issue 7, 2017.
- [4] L kholee phimu and Manoj Kumar, "Design and Implementation of Area Efficient 2-Parallel Filters on FPGA using Image System", International Conference on Energy, Communication, Data Analytics and Soft Computing, IEEE 2017
- [5] Shahnam Mirzaei, Anup Hosangadi, Ryan Kastner, "FPGA Implementation of High Speed FIR Filters Using Add and Shift Method", 1-4244-9707-X/06/\$20.00@2006 IEEE.
- [6] Amina Naaz.S, Mr.Pradeep M.N, Satish Bhairannawar and Srinivas halvi, "FPGA Implementation Of High Speed Vedic Multiplier using CSLA For Parallel Fir Architecture", 2014 2nd International Conference on Devices, Circuits and Systems (ICDCS).
- [7] Laxman P.Thakre, Suresh Balpande, Umesh Akare, Sudhir Lande, "Performance Evaluation and Synthesis of Multiplier used in FFT operation using Conventional and Vedic algorithms," Third international conference on emerging trends in Engineering and Technology , IEEE, 2010.
- [8] S. S. Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V. A., "Implementation of Vedic Multiplier for Digital Signal Processing," International

Conference on VLSI ,Communication & Instrumentation (ICVCI), 2011.

- [9] G. Vaithiyathan, K. Venkatesan, S.Sivaramakrishnan, S.Sivaand, S. Jayakumar, "Simulation and implementation of Vedic multiplier using VHDL code," International Journal of Scientific & Engineering Research, vol.4, 2013.
- [10] Pushpalata Verma and K. K. Mehta, "Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool," International Journal of Engineering and Advanced Technology(IJEAT), vol.1, June 2012.
- [11] C. Cheng and K. K. Parhi, "Further complexity reduction of parallel FIR filters," in Proc. IEEE ISCAS, May 2005, vol. 2, pp. 1835–1838.
- [12] C. Cheng and K. K. Parhi, "Low-cost parallel FIR structures with 2-stage parallelism," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 2, pp. 280–290, Feb. 2007.
- [13] J. G. Chung and K. K. Parhi, "Frequency-spectrum-based low-area low-power parallel FIR filter design," EURASIP J. Appl. Signal Process., vol. 2002, no. 9, pp. 444–453, Jan. 2002.
- [14] K. K. Parhi, VLSI Digital Signal Processing systems: Design and Implementation. New York: Wiley, 1999.
- [15] Nivedita A. Pande, Vaishali Niranjane, Anagha V. Choudhari, "Vedic Mathematics for Fast Multiplication in DSP," International Journal of Engineering and Innovative Technology (IJEIT) ,vol.2, 2013.
- [16] Krishnaveni D. and Umarani.T.G, "Vlsi implementation of Vedic multiplier with reduced delay," International Journal of Scientific & Engineering Research, vol.2, May-2011.