JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Optimization Analysis of Parallel 2×2 and 3×3 FIR Filter based on Partition Technique

Sangam Yadav[1], Prof. Satyarth Tiwari[2], Prof. Amrita Pahadia[3]
M. Tech. Scholar, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal[1]
Guide, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal[2]
Co-guide, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal[3]

Abstract: **In this paper Parallel FIR filter based on Brent Kung adder and partition multiplier using image system is present. Parallel FIR used in various application i.e. wireless communication, image processing and digital signal processing. We have design area efficient parallel FIR filter with the help of Brent kung adder and filter coefficient. 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).**

Index Terms – Brent Kung Adder, Vedic Multiplier, Parallel FIR Filter, 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. FIR FILTER

The term Finite Impulse Response (FIR) filter arises the filter output is computed as a weighted sum, finite term sum of past, present and perhaps future values of the filter input. The output of FIR filter is represented in Punskaya (2005) as,

$$y(n) = \sum_{k=-M_1}^{M_2} b_k \; x(n-k) \tag{1}$$

Where, both M_1 and M_2 are finite.

In one of the simplest FIR filters, we may consider 3-term moving average filter of the form,

$$y(n) = \frac{1}{3}(x(n-1) + x(n) + x(n+1)) \tag{2}$$

As, mentioned in early section, FIR filters consist only feedforward term. This is illustrated in Equation (2). Feedforward means, the present output does not depends on previous outputs and also there is no feedback of past or future outputs. It means that the output of FIR filter just contain only input related terms. 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. The structural architecture of direct form FIR filter is shown in Figure 1. This structure consists of delay unit, adder and multiplier units to perform filtering operations.

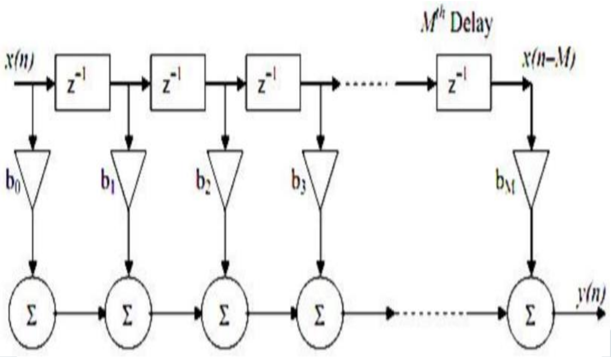


Figure 1: Direct Form FIR filter

III. METHODOLOGY

Partition Multiplier

We consider two N-bit operands x_{N-1} x_{N-2}…x_2 x1 x0 and y_{N-1} y_{N-2}…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 $M_0, M_1, M_2 ....M_{t-1}$ $N_0, N_1, N_2 ....N_{t-1}$ each consisting of r bits

Where

$$t = \frac{b}{r} \tag{3}$$

Consider the value of b = 8, r =2

Then

$$t = 4 \tag{4}$$

M and N can therefore be expressed as

$$M_0 = M(\frac{n}{2} - 1\ to\ 0) \tag{5}$$

$$M_1 = M(n - 1\ to\ \frac{n}{2}) \tag{6}$$

$$N_0 = N(\frac{n}{2} - 1\ to\ 0) \tag{7}$$

$$N_1 = N(n - 1\ to\ \frac{n}{2}) \tag{8}$$

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).

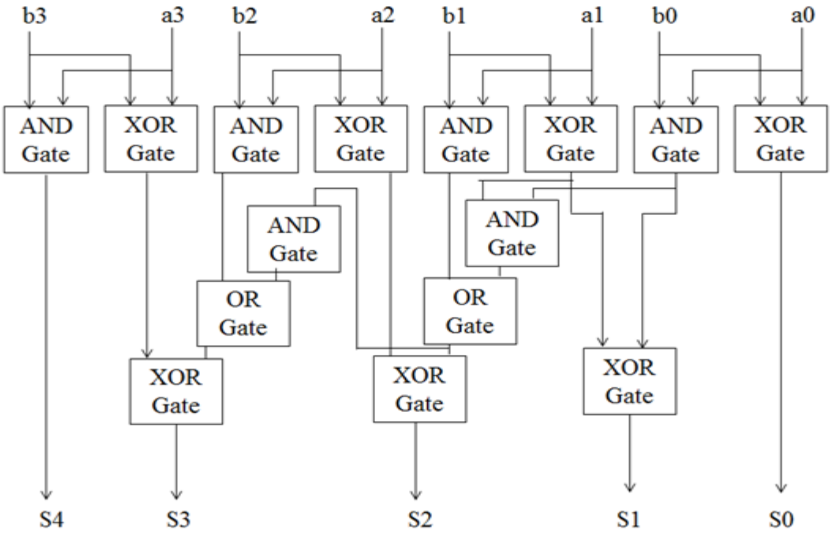


Figure 2: Block Diagram of Brent Kung Adder

IV. PROPOSED 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.

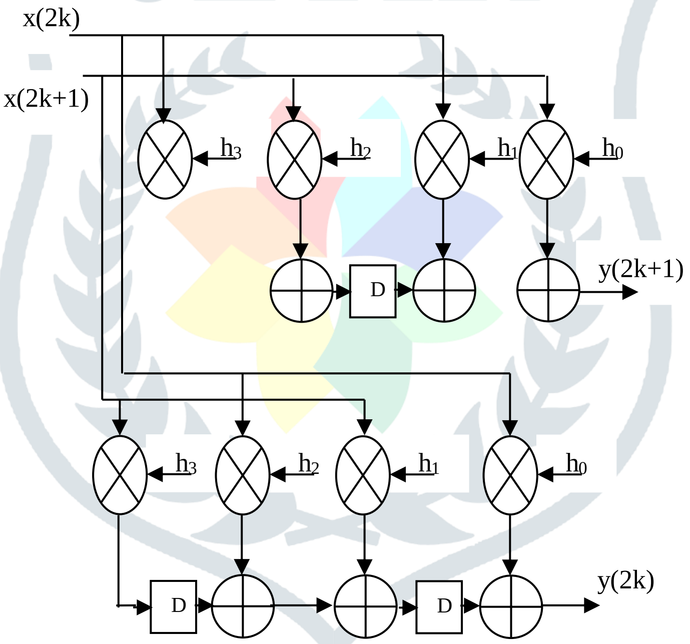


Figure 3: Block Diagram of 2×2 Parallel FIR Filter

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. 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.

From above equation we get that, for an N-tap three-parallel FIR filter, the proposed structure can save N/3 multipliers from the existing FFA structure with overhead of seven additional adders.

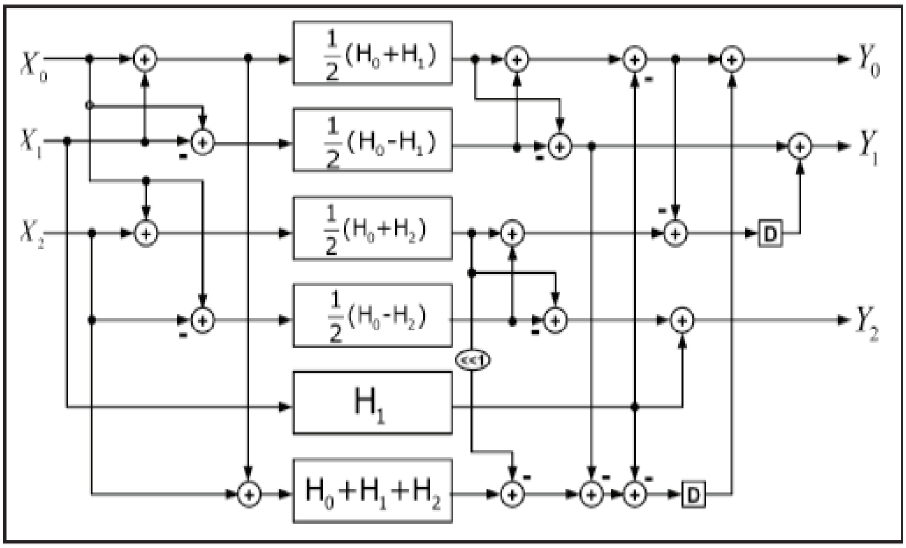


Figure 4: Proposed Three-Parallel FIR Filter Implementation

Multiplier-less Technique

On the off chance that the coefficients are little, it is exceptionally advantageous to acknowledge through the rich structure of FPGA LUT. While the coefficient is substantial, it will take parcel of capacity assets of FPGA and decrease the count speed. Then, the N-1 cycles likewise bring about too long LUT time and low registering speed. Shunwen Xiao, Yajun Chen, introduced a change and advancement of the DA calculation going for the issues of the arrangement in the coefficient of FIR channel, the capacity asset and the ascertaining speed, which make the memory size littler and the operation speed speedier to enhance the computational execution.

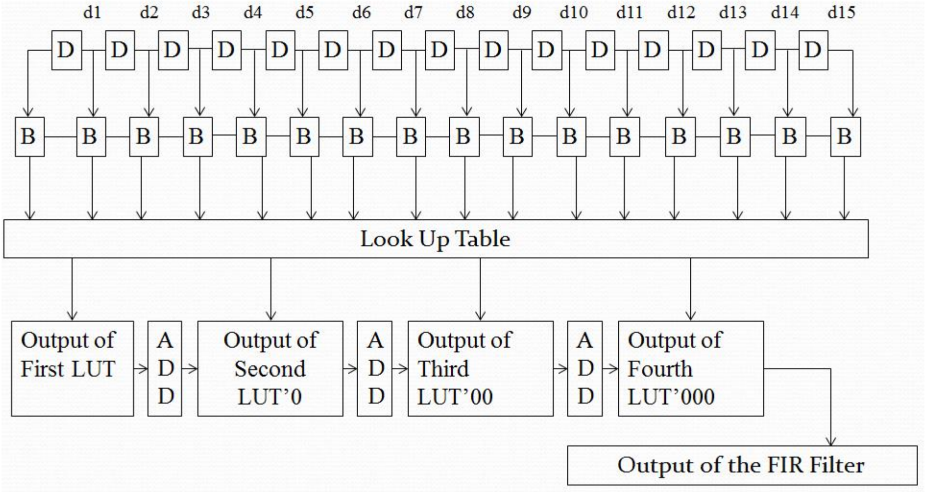


Figure 5: FIR Filter using Distributive Arithmetic Technique

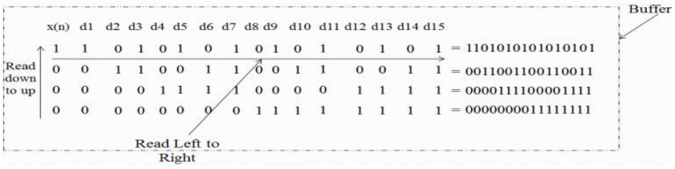
Example:-

Step 1:- x(n) = 0001, where x(n) is the input of the FIR Filter

Step 2:- x(n) is passing through all delay flip flop (D-FF),

d1= 0001, d2=0010, d3=0011, d4=0100, d5=0101, d6=0110, d7=0111, d8= 1000, d9=1001, d10=1010, d11=1011, d12=1100, d13=1101, d14=1110, d15=1111

Step 3:- Input of the FIR filter and output of the D-FF passing through Buffer



x(n)	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	= 1101010101010101
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	= 0011001100110011
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	= 0000111100001111
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	= 0000000011111111

Buffer
Read down to up
Read Left to Right

Step 4:- All buffer passing through LUT then

h0	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	h11	h12	h13	h14	h15
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Output of the LUT
P1 = h0+h1+h3+h5+h7+h9+h11+h13+h15
P2 = h2+h3+h6+h7+h10+h11+h114+h15
P3 = h4+h5+h6+h7+h12+h13+h14+h15
P4 = h8+h9+h10+h11+h12+h13+h14+h15

Suppose
h0=0000, h1=0001, h2=0010, h3=0011, h4=0100, h5=0101, h6=0110, h7=0111, h8=1000, h9=1001, h10=1010, h11=1011, h12=1100, h3=1101, h14=1110, h15=1111
So,

P1= 1000000
P2= 1000100
P3= 1001110
P4= 1011100

Step 5:- Output of the FIR Filter

Yn = P1 + P2’0 + P3’00 + p4’000

= 1000000 + 1000100’0 + 1001110’00 + 1011100’000
= 10011111000 (1240)

Parallel FIR Filter used in Image System

Field Programmable Gate Arrays (FPGAs) are pre-fabricated silicon devices that can be electrically programmed in the field to become almost any kind of digital circuit or system. For low and medium productions, FPGAs provide cheaper solution and faster time to market as compared to the ASIC circuits.

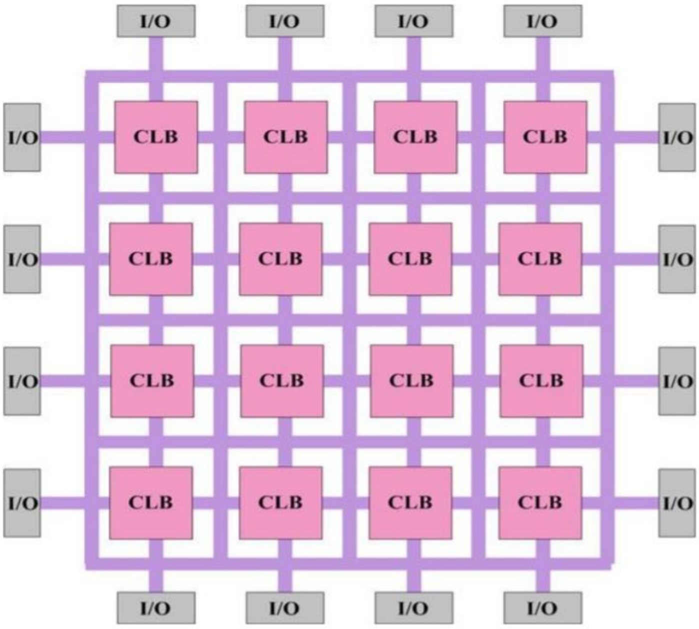


Figure 6: Architecture of FPGA

One alternative solution to SRAM based FPGA is the use of flash memory based FPGA. Flash memory or Electrically Erasable Programmable Read Only Memory (EEPROM) offers many advantages in FPGA. These memories are non-volatile in nature. Also flash memory based FPGAs are more area efficient than SRAM based FPGA. However, flash memory based FPGA has its own limitation of configurability and configurability. When compared to reconfigurable property, the SRAM based FPGA is considered as better than FPGA. In other hand, flash memory based FPGA uses non-standard CMOS process.

V. **SIMULATION RESULTS**

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 ISE^{TM} (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 Pro^{TM} 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.

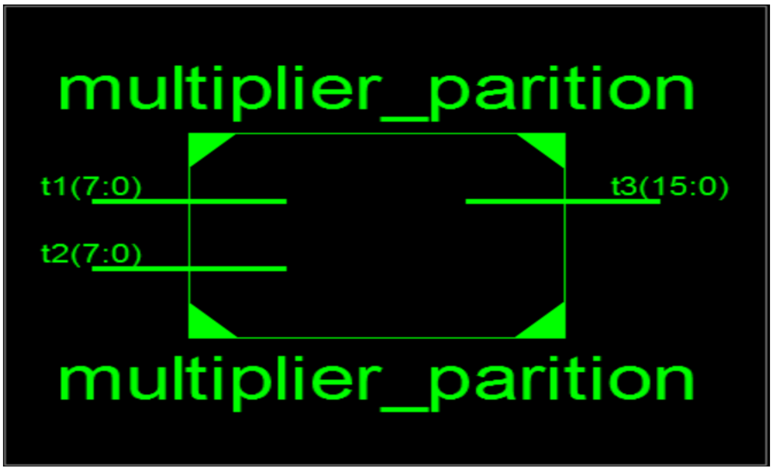


Figure 7: View Technology Schematic of Partition Multiplier

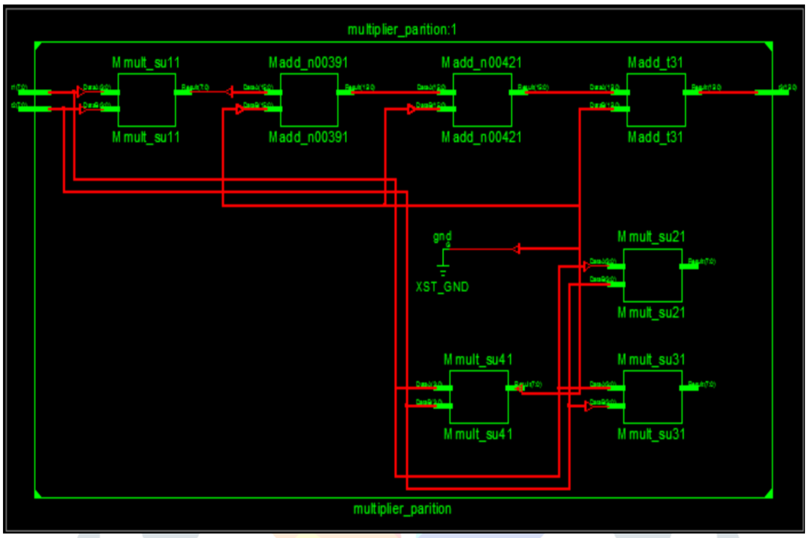
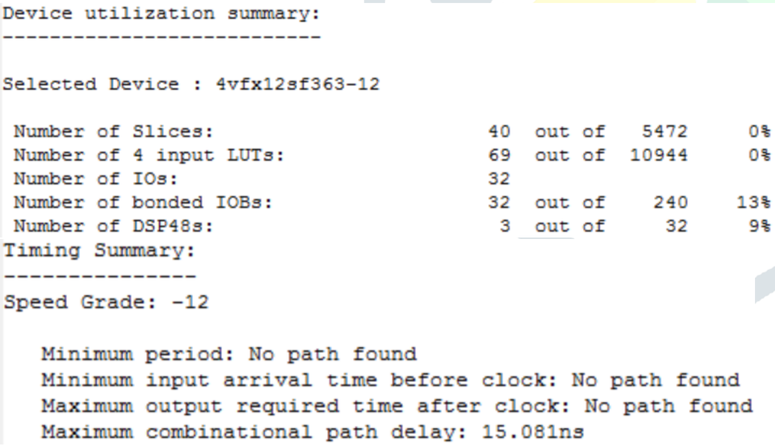


Figure 8: RTL View of Partition Multiplier



```
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
```

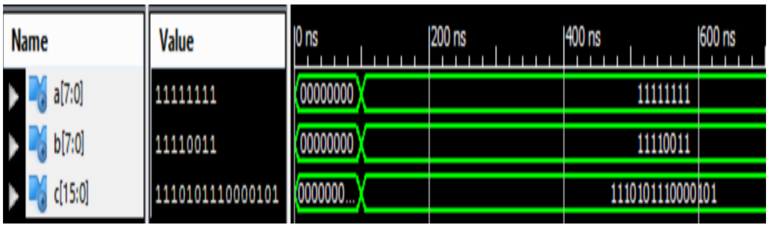


Figure 9: VHDL Test Bench of Partition Multiplier

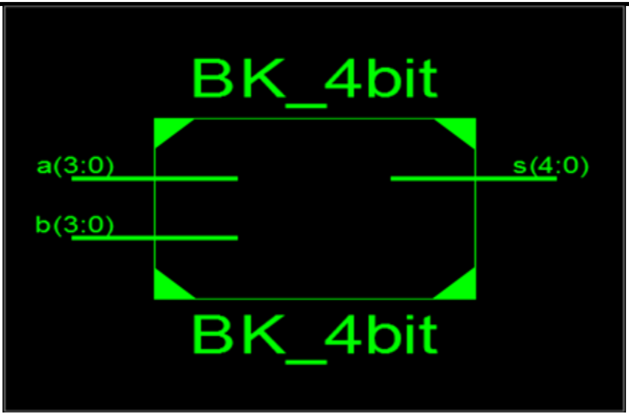


Figure 10: VTS for 4-bit BK Adder

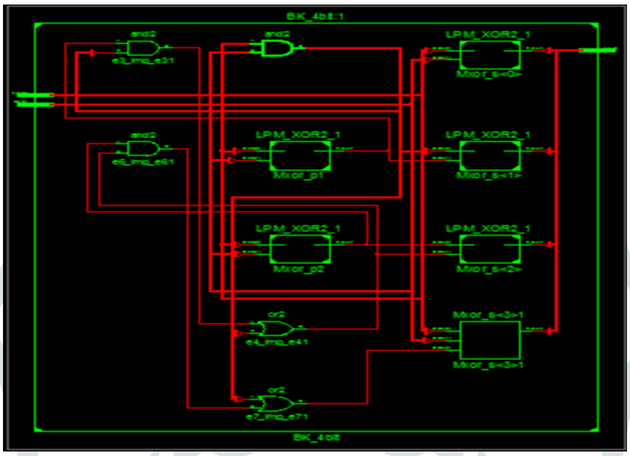


Figure 11: RTL View of 4-bit BK Adder

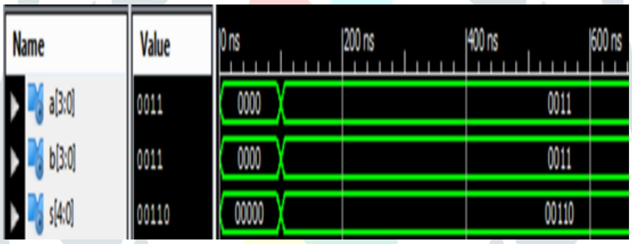


Figure 12: VHDL Test Bench of 4-bit BK Adder

Table 1: Comparison Result of Previous Multiplier and Proposed Multiplier

Bit	Previous Method [1]	Proposed Method
4-bit	10.43 ns	9.89 ns
8-bit	15.22 ns	15.081 ns
16-bit	19.58 ns	18.05 ns
32-bit	22.32 ns	21.55 ns

Table II: Theoretically Estimated Hardware of Proposed and Existing Structures for 4-bit

Structure	Filter Length(Tap)	FF	Adder	Multiplier
Previous Design (L=4) [2]	8	168	28	32
	16	360	60	64
	32	744	124	128
	64	1512	252	256
Proposed Structure (L=4)	8	60	58	0
	16	124	114	0
	32	252	226	0
	64	508	450	0

Table III: Comparisons Result for different types of proposed design and different types of device family

L	Length	Structure	No. of slice	No. of slice FF	No. of 4 i/p LUTs	MP
2	27-tap	FFA	4270	881	7998	-
		Previous Work	12875	950	23707	-
		Proposed	3465	721	6892	59.385ns
	32-tap	FFA	4530	910	8014	-
		Previous Work	11481	1021	24193	-
		Proposed	3821	742	7103	62.56ns
	81-tap	FFA	12713	2631	23875	-
		Previous Work	37900	2859	69559	-
		Proposed	10463	2319	20451	68.43 ns
3	27-tap	FFA	6277	1639	11441	-
		Previous Work	13837	1747	25764	-
		Proposed	5103	1295	9821	60.43 ns
	32-tap	FFA	8732	3219	13651	-
		Previous Work	15432	4388	27002	-
		Proposed	7821	2813	11395	64.63 ns
	81-tap	FFA	18583	4902	33895	-
		Previous Work	41244	5203	76770	-
		Proposed	15439	4021	28542	73.32 ns

VI. CONCLUSION

In previous design, carry look ahead adder was used for designing parallel FIR filter 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] S Narendran and B T Geetha, "Performance Analysis of Parallel FIR Digital Filter Based on Even Symmetric Fast FIR Algorithm using Different Adders", 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE 2021.

[2] K. Anjali Rao;Abhishek Kumar;Neetesh Purohit, "Efficient Implementation for 3-Parallel Linear-Phase FIR Digital Odd Length Filters", IEEE 4th Conference on Information & Communication Technology (CICT), IEEE 2020.

[3] Yi Zheng;Ping Zheng, "Case Teaching of Parallel FIR Digital Filter Design Combined Matlab with FPGAs", International Conference on Artificial Intelligence and Education (ICAIE), IEEE 2020.

[4] S. Sreekanth; Pratima B. Shinde; G. Vijaya Durga, "Performance Analysis of Higher Order FIR Polyphase Filter", Second International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE 2018.

[5] Qiaoyu Tian;Yinan Wang;Guiqing Liu;Xiangyu Liu;Jietao Diao, Hui Xu, "Hardware Efficient Parallel FIR Filter Structure Based on Modified Cook-Toom Algorithm", IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), IEEE 2018.

[6] Payal Paliwal;Janki Ballabh Sharma, "Efficient FPGA Implementation Architecture of Fast FIR Algorithm Using Han-Carlson Adder Based Vedic Multiplier", International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE 2018.

[7] Swetha Annangi;Ravisankar Puli, "ASIC implementation of efficient 16-parallel fast FIR algorithm filter structure", 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE 2017.

[8] Shalina Percy Delicia Figuli;Peter Figuli;Jürgen Becker, "A reconfigurable high-speed spiral FIR filter architecture", 40th International Conference on Telecommunications and Signal Processing (TSP), IEEE 2017.

[9] 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.

[10] 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).

[11] 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.

[12] 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.

[13] G.Vaithiyanathan, K.Venkatesan, S.Sivaramakrishnan, S.Sivaand, S.Jayakumar, "Simulation and implementation of Vedic multiplier usingVHDL code," International Journal of Scientific & Engineering Research, vol.4, 2013.

[14] Pushpalata Verma and K. K. Mehta, "Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool," International Journal of Engineering and Advanced Technolog(IJEAT), vol.1, June 2012.

[15] C. Cheng and K. K. Parhi, "Furthur complexity reduction of parallel FIR filters," in Proc. IEEE ISCAS, May 2005, vol. 2, pp. 1835–1838.

[16] 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.

[17] 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.

[18] K. K. Parhi, VLSI Digital Signal Processing systems: Design and Implementation. New York: Wiley, 1999.

[19] 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.

[20] Krishnaveni D. and Umarani.T.G, "Vlsi implementation of Vedic multiplier with reduced delay," International Journal of Scientific & Engineering Research, vol.2, May-2011.