http://www.jctjournals.com

ISSN(ONLINE):2278 - 3814

Volume-12, Issue-11, November 2023

JOURNAL OF COMPUTING TECHNOLOGIES (JCT)

International Journal

Page Number: 01-07

Implementation of High Performance FIR Filter Architecture using Distributive Arithmetic Technique

[1]Rambhuwan Sahu, [2]Prof. Nishi Pandey, [3]Prof. Abhishek Agwekar
[1] M.Tech Scholar, [2]Assistant Professor ,[3]Head of department
[1, 2, 3] TIEIT Bhopal, M.P., INDIA
[1]Sahurambhuvan6@gmail.com, [2]nishi.pandey@trubainstitute.ac.in, [3]abhiagwekar@gmail.com

Abstract*— Digital filters are the essential units for digital signal processing systems. Traditionally, digital filters are achieved in Digital Signal Processor (DSP), but DSP-based solution cannot meet the high speed requirements in some applications for its sequential structure. Nowadays, Field Programmable Gate Array (FPGA) technology is widely used in digital signal processing area because FPGA-based solution can achieve high speed due to its parallel structure and configurable logic, which provides great flexibility and high reliability in the course of design and later maintenance. In general, Digital filters are divided into two categories, including Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). And FIR filters are widely applied to a variety of digital signal processing areas for the virtues of providing linear phase and system stability. The FPGA-based FIR filters using traditional direct arithmetic costs considerable multiply-and-accumulate (MAC) blocks with the augment of the filter order. A new design and implementation of FIR filters using Distributed Arithmetic is provided in this project to solve this problem. Distributed Arithmetic structure is used to increase the resource usage while pipeline structure is also used to increase the system speed. In addition, the divided LUT method is also used to decrease the required memory units. However, according to Distributed Arithmetic, we can make a Look-Up-Table (LUT) to conserve the MAC values and callout the values according to the input data if necessary. Therefore, LUT can be created to take the place of MAC units so as to save the hardware resources.

Keywords : Field Programmable Gate Array (FPGA), multiply-and-accumulate (MAC), Look-Up-Table (LUT), and Digital Signal Processor (DSP),

I. INTRODUCTION

In the recent years, there has been a growing trend to implement digital signal processing functions in Field Programmable Gate Array (FPGA). In this sense, we need to put great effort in designing efficient architectures for digital signal processing functions such as FIR filters, which are widely used in video and audio signal processing, telecommunications and etc. Traditionally, direct implementation of a K-tap FIR filter requires K multiply-and-accumulate (MAC) blocks, which are expensive to implement in FPGA due to logic complexity and resource usage. To resolve this issue, we first present DA, which is a multiplier-less architecture. Implementing multipliers using the logic fabric of the FPGA is costly due to logic complexity and area usage, especially when the filter size is large. Modern FPGAs have dedicated DSP blocks that alleviate this problem, however for very large filter sizes the challenge of reducing area and complexity still remains. An alternative to computing the multiplication is to decompose the MAC operations into a series of lookup table (LUT) accesses and summations. This approach is termed distributed arithmetic (DA), a bit serial method of computing the inner product of two vectors with a fixed number of cycles. The original DA architecture stores all the possible binary combinations of the coefficients w[k] of equation (1) in a memory or lookup table. It is evident that for large values of L, the size of the memory containing the pre computed terms grows exponentially too large to be practical. The memory size can be reduced by dividing the single large memory (2Lwords) into m multiple smaller sized memories each of size 2k where L = m × k. The memory size can be further reduced to 2L−1 and 2L−2 by applying offset binary coding and exploiting resultant symmetries found in the contents of the memories

II. LITERATURE REVIEW

It can be executed in a clear way. All in all, as flag recurrence builds, the uniqueness in productivity increments. Attributes of uses where advanced channels with more size and proficient than simple channels are: straight stage, high stop band weakening, low pass band swell; the channel's reaction must be programmable or versatile; the channel must control stage and, low shape considers (a computerized channel's shape variable is the proportion of the channel's pass data transfer capacity in addition to the channel's move transmission capacity to the channel's pass transmission capacity). The process of converting an analog signal into digital form is performed by sampling with a finite sampling frequency.

Deepak Kumar Patel et al. (2016, [2]), in this paper, the speed and range are presently the very beginnings of the principal configuration issues in advanced period. To build speed, while doing the duplication or expansion operations, has dependably been a fundamental prerequisite of planning of cutting edge framework and application. Convey Select Adder (CSA) is a quickest viper utilized as a part of numerous processors to fulfill quick number juggling capacity. A wide range of viper engineering outlines have been created to build the productivity of the snake. It is ordinarily realized that every second any processors performed a large number of work works in semiconductor industry. So when we do outlining of multipliers, one of the principle benchmarks is performing speed that ought to be taken in the brain. In this paper, we propose a method for outlining of FIR channel utilizing multiplier in light of compressor and convey select viper. Execution of all viper outlines is actualized for 16, 32 and 64 bit circuits.

K. Durga et al. (2016, [3]), in this paper, an effective engineering of FIR channel structure is exhibited. For accomplishing low power, reversible rationale method of operation is actualized in the plan. Territory overhead is the exchange off in the proposed outline. From the amalgamation comes about, the proposed low power FIR channel engineering offers 18.1 % of energy sparing when contrasted with the customary outline. The territory overhead is 2.6% for the proposed engineering.

Indranil Hatai et al. (2015, [4]), this brief proposes a two-step optimization technique for designing a reconfigurable VLSI architecture of an interpolation filter for multistandard digital up converter (DUC) to reduce the power and area consumption. The proposed technique initially reduces the number of multiplications per input sample and additions per input sample by 83% in comparison with individual implementation of each standard’s filter while designing a root-raised-cosine finite-impulse response filter for multistandard DUC for three different standards. In the next step, a 2-bit binary common subexpression (BCS)-based BCS elimination algorithm has been proposed to design an efficient constant multiplier, which is the basic element of any filter. This technique has succeeded in reducing the area and power usage by 41% and 38%, respectively, along with 36% improvement in operating frequency over a 3-bit BCS-based technique reported earlier, and can be considered more appropriate for designing the multi-standard DUC.

Shamim Akhter et all in 2018" Design and Analysis of Distributed Arithmetic based FIR Filter, In designing digital filters,Multiply-Accumulate (MAC) unit is used. MAC comprises of multiplier, adder and an accumulator. Faster adder and multiplier circuits are required for high speed MAC unit. But MAC based structures have disadvantages like high power dissipation, slow processing etc. The multiplication operation where input data is to be multiplied with the fixed coefficients considerably took large place to store their temporary data. So, memory based multiplication technique substitute multipliers to reduce area and latency of system. Distributed Arithmetic (DA) is one of the memory based technique. DA based technique substitute multipliers in FIR filters. In this paper, detailed analysis is presented for designing 16-Tap FIR filter using DA and Off-Set Binary Coding (OBC)-DA in VHDL. Synthesis is done using Xilinx ISE for Virtex-4 ML 402. Area, delay and power analysis is performed using Synopsys Design Compiler for 32/28 nm std_cell.

Narendiran S. in 2021, One of the essential components of a Digital Signal Processing (DSP) system is the Finite Impulse Response (FIR) filter. FIR filter uses the Multiply and Accumulate (MAC) operation for its computation. Conventional MAC units are slow and consume high power, making them unsuitable for energy-constrained devices. The MAC operations in FIR filter uses constant filter coefficients as one of its inputs. This situation is well suited for a bit-serial technique such as Distributed Arithmetic (DA). However, the traditional DA has the drawback of using huge memory resources as the filter order increases. An efficient LUT-less Modified Distributed Arithmetic architecture is proposed in this paper to solve the memory problem. This architecture removes the need for precipitation of weighted sums needed for the LUT in a DA using multiplexers and adders. Also, the architecture is designed to extend the range of input values. Further, a 16-Tap FIR filter is designed, synthesized with Xilinx ISE, and implemented for an XC4VSX35-FF668-10 based FPGA to measure the performance of this architecture. Our implementation results show that the design uses fewer resources and achieves faster filtering than the filter's previous implementations.

Balaji M,in 2020 This work presents a way to increase the throughput and energy efficiency of finite impulse response (FIR) filters through the efficient application of retiming and two-level pipelining. It is a challenge to increase the filter's throughput and energy efficiency while reducing latency and hardware complexity. The operations of addition and multiplication are divided using two-level pipelining. The break addition procedure is retimed. The architecture of m-tap filter (4-tap, 8-tap, 16-tap, 32-tap, and 64-tap) with n-bit input word length (4-bit, 8-bit, 16-bit, and 32-bit) Pipelined Retiming delay generation Filters (PRF), were designed. The proposed distributed arithmetic based FIR Filter with pipelining has produced the least delay of 2.564ns for 4-tap with 8-bit input, and the maximum delay of 56.040ns for 64-tap with 32-bit word length. The proposed distributed arithmetic-based FIR

Filter with retiming method has produced the least delay of 0.687ns for 4-tap with 8-bit input, and the maximum delay of 4.535ns for 64-tap with 32-bit word length. When compared with the pipelining method, the delay has been decreased by 73.20% for 4-tap with 8-bit input and 91.90% for 64-tap with 32-bit word length.

Ch. Pratyusha et all in 2019, In this paper the proposed efficient FIR filter architecture using a distributed arithmetic (DA) algorithm in which two issues are discussed in the conventional FIR filter. The FIR filter is well known to include delay elements, multipliers and adders. Due to the need for multipliers, this results in 2 demerits which are (i) increased in area and (ii) delayed increases that eventually lead to low efficiency (low speed). A notable feature of the proposed technique is to substitute a trivial amount of indexed LUT pages instead of conventional LUT based DA that it helps to maintain the access time lower. Also, significant idea connected with the proposed technique is required page can be thoroughly selected with the selection module without needing adders that result in reduced computation time. Furthermore, the proposed fast FIR filter is used for the powerful ECG noise elimination technique, which is prevalently used in biomedical and healthcare applications. The designs are simulated and synthesized by using Xilinx ISE. It can be seen from reports that our proposed DA consumes 30% less power for 11-tap FIR filters with a 40% shorter area, while the saving in power consumption for 8-tap FIR filters is 30% to 80% and 35% to 80% in the area. Especially in contrast with all the above-mentioned DA techniques, our enhanced quick FIR filters require less area and less power intake due to their lower memory requirements. All architectures are designed for FIR filters with 4 and 8 taps.

Manoj Srivastava et all in 2015, n this project use Distributed Arithmetic (DA) technique for FIR filter. In this technique consist of Look Up Table (LUT),shift register and accumulator. Based on this technique multipliers in FIR filter are removed. Multiplication is performed through shift and addition operations. The LUT can be subdivided into a number of LUT to reduce the size of the LUT forhigher order filter. Each LUT operates on a different set of filter taps. Analysis on the performance of various filter orders with different address length are done using Xilinx synthesis tool. The proposed architecture provides less latency and lessarea compared with existing structure of FIR.

D. Kalaiyarasi et all in 2019, Distributed Arithmetic (DA) based architecture is an efficient technique to attain high throughput without hardware multiplier and also it is essential for bit serial operation. The DA based Finite Impulse Response (FIR) adaptive filter is well suited for hardware implementation in Field Programmable Gate Array (FPGA) device. In conventional DA the partial products of the filter coefficients have been pre-calculated and stored in Look up Table (LUT) which in turn will increase the logic elements and power. To overcome this problem DA based Least Mean Square (LMS) adaptive filter using offset binary coding (OBC) without LUT is proposed. The proposed method will reduce the logic elements by half when compared to the conventional DA based OBC filter. The Carry Save Accumulator (CSA) is used to carry out the operation of shift and accumulation. The proposed architecture is implemented in Quartus II 9.1 with the device as Stratix-EP2S15F484C3 which offers 13.72% high throughput, 56.92% reduction in logic elements, 42.84% reduction in power, 57.74% reduction in logical registers for N = 16 and for N = 32 the number of logical element is reduced to 80.87%, 66.66% reduction in power and 24.12% high throughput

III. PROPOSED METHODOLOGY

A. Digital Fir Filter

Digital signal processing algorithms are increasingly employed in modern wireless communications and multimedia consumer electronics, such as cellular telephones and digital cameras. The new generation of telecommunication equipment often requires the use of high order high-speed low-power Finite Impulse Response (FIR) filters. The output of an N tap FIR filter, which is the convolution of the latest L input samples, is given in equation (3.1). L is the number of coefficients h (k) of the filter, and x (n) represents the input time series.

B. Distributed Arithmetic Technique

Distributed Arithmetic (DA) is a widely-used technique for implementing sum-of-products computations without the use of multipliers. Designers frequently use DDA to build efficient Multiply-Accumulate Circuitry (MAC) for filters and other DSP applications. The main advantage of DA is its high computational efficiency. DDA distributes multiply and accumulate operations across shifters, Look up Tables (LUTs) and adders in such a way that conventional multipliers are not required.

In order to reduce hardware cost, all the bits of an input data can be processed in a parallel manner using Look Up tables. But in Distributed Arithmetic structure, bitwise operation is performed. The input data is processed bit by bit, first processing the least significant bit and then rest of the bits. There is substantial reduction in hardware as all bits have to pass through same architecture. In other words, if there is a N-bit parallel design, the DA method requires only (1/N)th of the hardware resources. As a result only one clock cycle is required for execution while about N cycles are required in serial execution. However, for the serial design the time-hardware product is smaller than the parallel design because the propagation delays are generally smaller as compared to the parallel structure.

3.3 Proposed Architecture 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.

IV. DESIGN MODULES

A. Design

The design process involves conversion of requirements into a format that represents the desired digital function(s). Common design formats are schematic capture, hardware description language (HDL), or a combination of the two. Each method has its advantages and disadvantages but HDLs generally offer the greatest design flexibility. Schematic capture: Schematic capture is a graphical depiction of a digital design and shows the actual interconnection between each logic gate that produces the desired output function(s). Many of these logic gate symbols involve proprietary information which is available to the designer only through the specific vendor's component library. It makes the design unrecognizable by competitors' FPGA development tools and makes it vendor dependent. That means, the entire design process has to be repeated if a different vendor is used. View- Draw and EASE are examples of schematic capture tools by view logic and HDL respectively. The main advantage of schematic capture is that the graphical representation is easy to understand. But an increase in cost and time to reproduce a design for different vendors due to the design's proprietary nature are its major drawbacks. HDL method: Hardware Description Languages (HDLs) use code to represent digital functions. "Firmware" often refers to the resulting HDL code. Use of HDL codes is a common and popular approach to FPGA design. One can create the source code with any text editor. HDLs can be generic (supported by multiple simulation and synthesis tool sets) like Verilog or VHDL (Very High Speed Integrated Circuit HDL), or vendor specific like Altera's Hardware Description Language (AHDL), which is only recognizable by Altera's design tool set. There are two writing styles for HDL designs: structural or behavioral. Structural HDL firmware is the software equivalent of a schematic capture design. Like schematic capture, a structural design uses vendor specific components to construct the desired digital functions. It is again vendor dependent and has the same disadvantages. Behavioral HDL firmware describes digital functions in generic or abstract terms that are generally vendor independent. This provides enough flexibility for code reuse in different vendor's FPGAs with little or no code modification. Behavioral designs have advantages of its flexibility, time and cost-savings. Only those components are required to be changed for designs that require vendor specific resources, such as RAM. VHDL and Verilog are the most popular HDL languages.

B. About the Software Tool

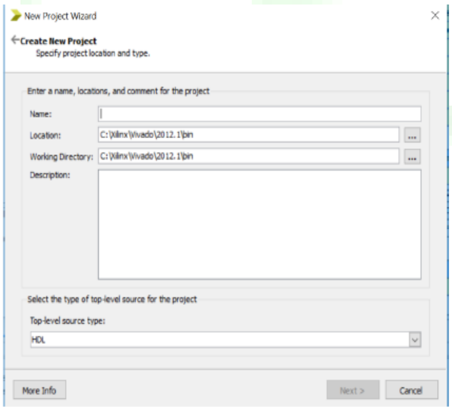
Condition setup is the workplace or apparatuses on which result examination has been done in Xilinx 14.1i. Xilinx is the exceptionally solid programming apparatus to investigation and reproduce the mind boggling circuits. There are such a large number of variants for Xilinx programming, for example, 6.1i, 9.1i, 10.2i, 13.1i and 14.2i. For the most part two programming dialect are utilizing VHDL and Verilog. VHDL is an acronym for VHSIC equipment depiction dialect (VHSIC is an acronym for fast incorporated circuits). It is an equipment depiction dialect that can be utilized to demonstrate a computerized framework at many levels of retention running from the calculation level to the door level [14]. VHDL permits clients or software engineers to utilize certain squares which contain certain arrangement of consecutive articulations. One such square is known as a procedure. The (<=) administrator, it is known as the task administrator and is utilized just to assign qualities to signals. For factors the administrator utilized is (:=). Some chief terms that are used at the basic level are:

i. Libraries
ii. Data types
iii. Signals
iv. Variables
v. Entity
vi. Architecture

Other important terms for the VHDL program such as process, component, function, procedures and state diagrams are used in programming.

C. VHDL 14.1i

Xilinx 14.1i is initial software for simulation result. Interfacing and work environment are different from other version. During the start the software click on new project and give the name according to file name and select the environment which in popup small screen as shows in Figure 4.3.



New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name:

Location: C:\Xilinx\Vivado\2012.1\bin

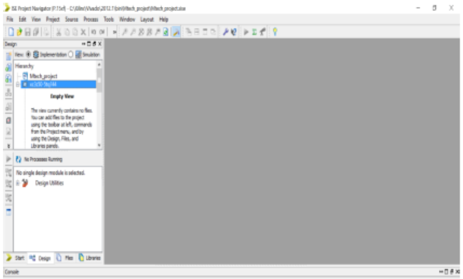
Working Directory: C:\Xilinx\Vivado\2012.1\bin

Description:

Select the type of top-level source for the project

Top-level source type: HDL

project name of create new project file than move the next button the IC according to the design properties is shown in Figure 4.4.



Empty View

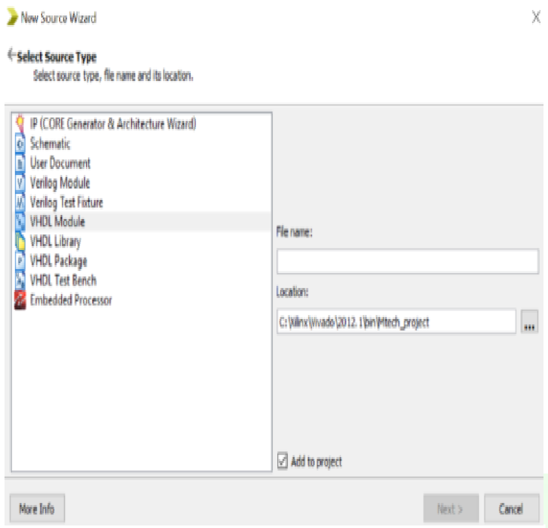
The view currently contains no files. You can add files to the project using the toolbar at left, commands from the Project menu, and by using the Design, Files, and Libraries panels.

No Processes Running

No single design module is selected.

Figure 4.4: Screenshot of 14.1i Software for Select IC

Right click of the IC and moving the new source create new project, select VHDL module and types the file name shows in Figure 4.4.



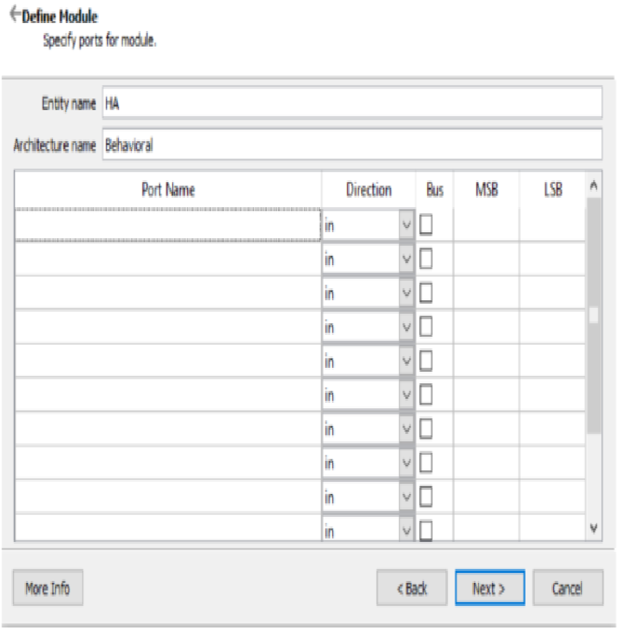
File name:

Location: C:\Xilinx\vivado\2012.1\bin\Mtech_project

☑ Add to project

Figure 4.5: Screenshot of 14.1i Software for VHDL Module

me and move the next butter and gives the input output of the project shows in Figure 4.6. Figure 4.6 clearly that the two types of buses are used i.e. MSB and LSB, MSB stand for most significant bit and LSB stand for lower significant bit. Figure 4.6:

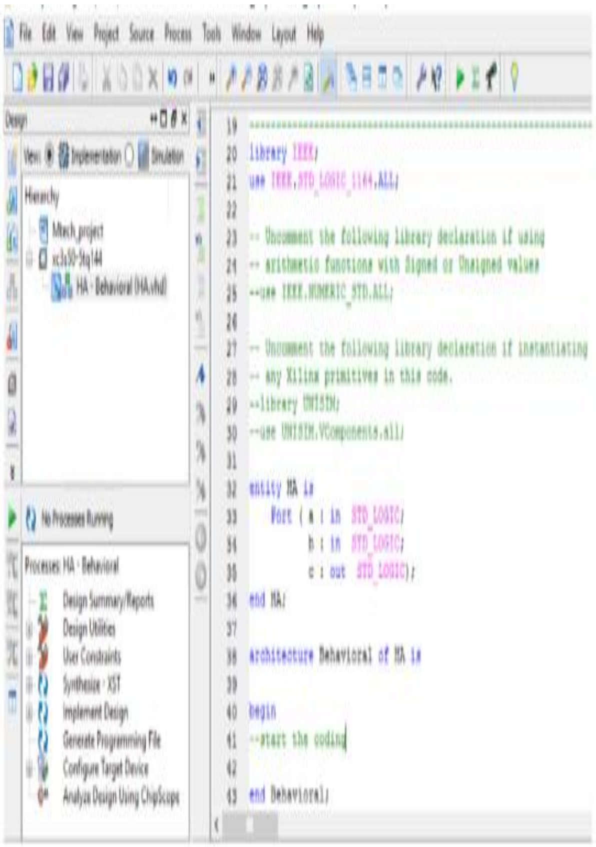


Entity name: HA

Architecture name: Behavioral

Figure 4.6: Screenshot of 14.1i Software for Input Output Port

Types the inputs output and move the next button, after the finish. Figure 4.5 clearly that the left hand site create file mane and right site create all the input output port. Last step create the coding in begin sector.



```
----------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity HA is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           c : out  STD_LOGIC);
end HA;

architecture Behavioral of HA is

begin
--start the coding

end Behavioral;
```

Figure 4.7: Screenshot of 14.1i Software for start the code.

V. SIMULATION RESULT

A. Simulation Parameter

All the experiment analysis is done by 14.1i in Vertex device family. Xilinx 14.1i tool provides less propagation delay than to 6.2i Xilinx tool. The most important advantage of this tool is less memory with high speed analysis any complex logical circuit. Simulation and synthesize of finite impulse response (FIR) logical circuit can be enhanced by Xilinx design suit 14.1i Vertex device family series and device. Result analysis is always being done according to some important parameters like slice, IOBs, propagation delay, memory and LUTs.

5.2 Simulation Result

We functionally verified each unit presented in this dissertation including FIR filter using distributed arithmetic algorithm. We have been found that number of slices, number of LUTs and maximum combinational path delay is less compared to conventional algorithm.

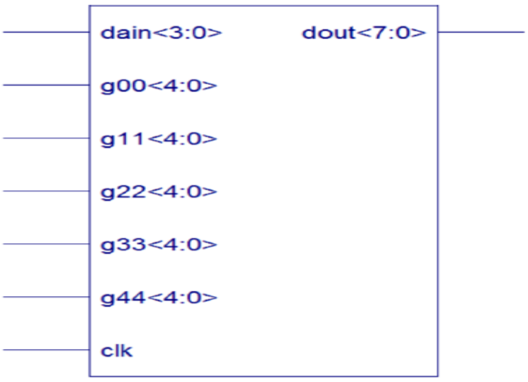


Figure 5.1: View Technology Schematic of 5-tap FIR Filter

Figure 5.1 shows the view technology schematic of 5-tap FIR filter using distributed arithmetic technique (D.A.T.). In this fig., 'dain' is the input of the 5-tap FIR filter using distributed arithmetic technique, 'g00, g11, g22, g33, g44' are the input of the 5-tap FIR filter using distributed arithmetic technique and 'dout' is the output of the 5-tap FIR filter using D.A.T..

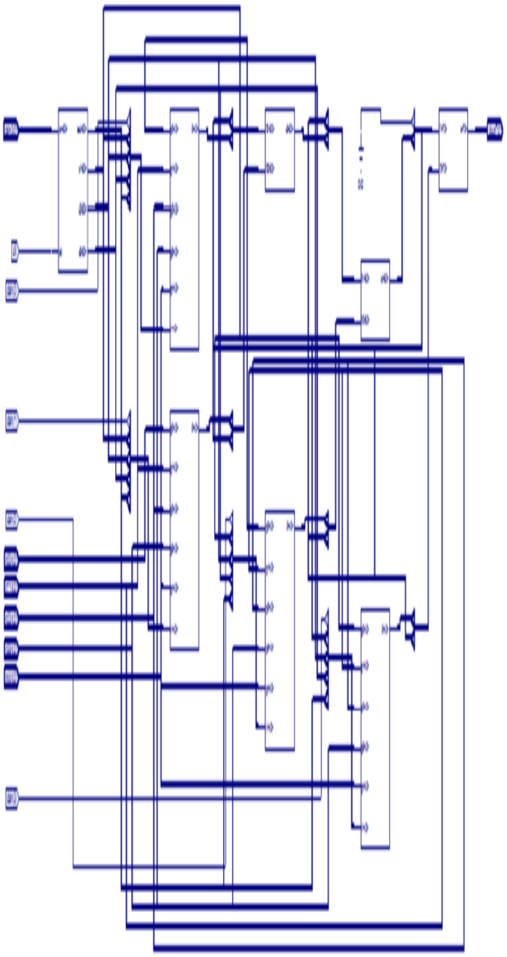


Figure 5.2: Resistor Transfer Level of 5-tap FIR Filter

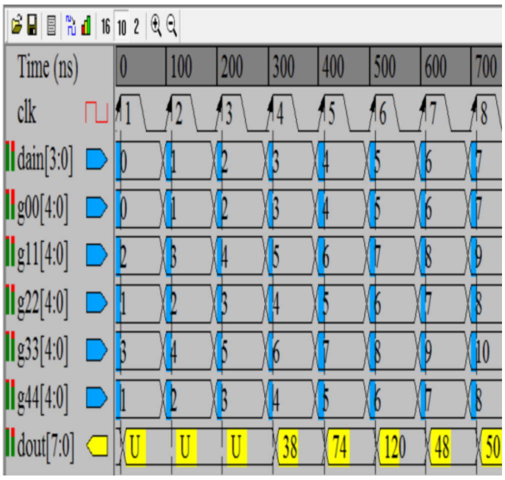
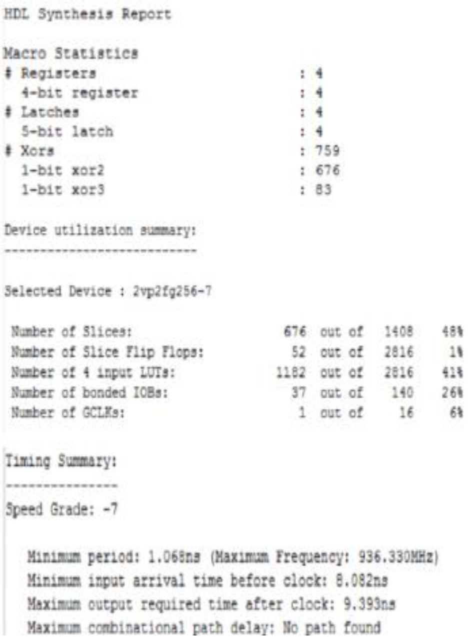


Figure 5.3: Output Waveform of the 5-tap FIR Filter



```
HDL Synthesis Report

Macro Statistics
# Registers                              : 4
  4-bit register                         : 4
# Latches                                : 4
  5-bit latch                            : 4
# Xors                                   : 759
  1-bit xor2                             : 676
  1-bit xor3                             : 83

Device utilization summary:
---------------------------

Selected Device : 2vp2fg256-7

 Number of Slices:                      676  out of   1408    48%
 Number of Slice Flip Flops:             52  out of   2816     1%
 Number of 4 input LUTs:               1182  out of   2816    41%
 Number of bonded IOBs:                  37  out of    140    26%
 Number of GCLKs:                         1  out of     16     6%

Timing Summary:
---------------
Speed Grade: -7

   Minimum period: 1.068ns (Maximum Frequency: 936.330MHz)
   Minimum input arrival time before clock: 8.082ns
   Maximum output required time after clock: 9.393ns
   Maximum combinational path delay: No path found
```

Figure 5.4 shows the view technology schematic of 8-tap FIR filter using distributed arithmetic technique. In this figure, 'dain' is the input of the 8-tap FIR filter using distributed arithmetic technique, 'h0, h1, h2, h3, h4, h5, h6, h7' are the input of the 8-tap

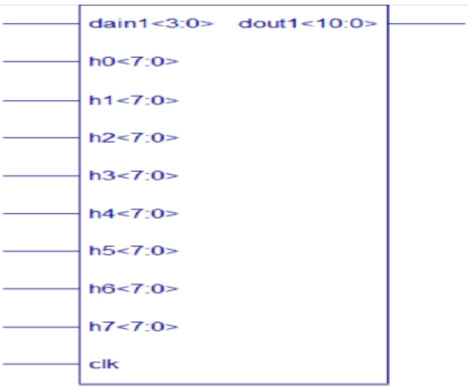


Figure 5.4: View Technology Schematic of 8-tap FIR Filter

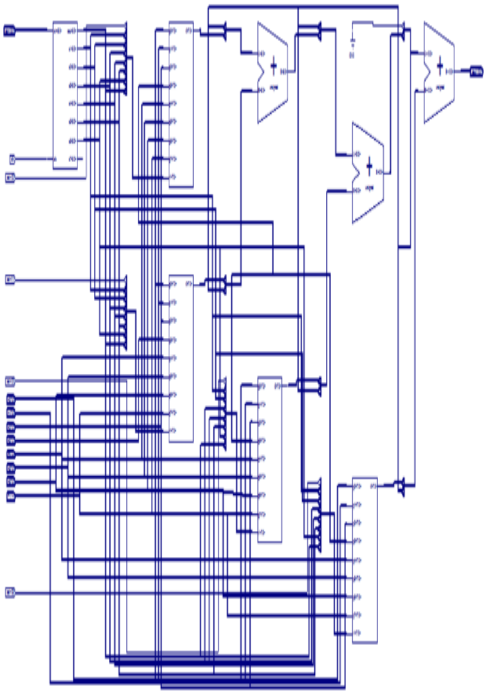
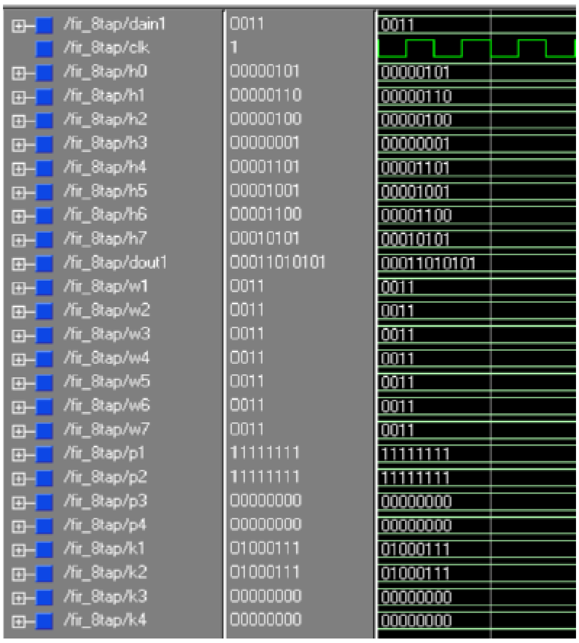


Figure 5.5: Resistor Transfer Level of 8-tap FIR Filter



```
/fir_8tap/dain1   0011          0011
/fir_8tap/clk     1
/fir_8tap/h0      00000101      00000101
/fir_8tap/h1      00000110      00000110
/fir_8tap/h2      00000100      00000100
/fir_8tap/h3      00000001      00000001
/fir_8tap/h4      00001101      00001101
/fir_8tap/h5      00001001      00001001
/fir_8tap/h6      00001100      00001100
/fir_8tap/h7      00010101      00010101
/fir_8tap/dout1   00011010101   00011010101
/fir_8tap/w1      0011          0011
/fir_8tap/w2      0011          0011
/fir_8tap/w3      0011          0011
/fir_8tap/w4      0011          0011
/fir_8tap/w5      0011          0011
/fir_8tap/w6      0011          0011
/fir_8tap/w7      0011          0011
/fir_8tap/p1      11111111      11111111
/fir_8tap/p2      11111111      11111111
/fir_8tap/p3      00000000      00000000
/fir_8tap/p4      00000000      00000000
/fir_8tap/k1      01000111      01000111
/fir_8tap/k2      01000111      01000111
/fir_8tap/k3      00000000      00000000
/fir_8tap/k4      00000000      00000000
```

Figure 5.6: Output Waveform of the 8-tap FIR Filter

B. Comparison Result

As shown in table 5.1 the number of slice, number of slice flip flop, number of LUTs, minimum period and maximum frequency are obtained from the proposed FIR filter using distributed arithmetic algorithm.

A. Simulation

The simulation was performed using XILINX and Model Sim simulator.

B. Hardware Utilization

The VHDL languages were used to design the VLSI architecture modules and are synthesized Virtex-2p, Vertex-E and Spartan-3 (XC2VP27FG, XCV50E8CS, and XC3s504PQ). Hardware description language (HDL) synthesis report for 16-tap, 32-tap and 64-tao FIR filter using multiplier-less DA technique are shown in table I, table II and table III respectively. It is observed from the table I, shows that the B(N-1) + B'(N-1) flip flop, L(N-1) adders, NL multiplier and TM + TA + 3TFA cyclic period for previous design but (N-1)L + NL flip flop, L(N-1) + (L-1)(N+2) adder, 0 multiplier and TA + (N-1) TFA cyclic period for proposed multiplier-less FIR filter implemented. It is observed from the table II, the flip flop, adder, multiplier and cyclic period for the proposed FIR filter using DA technique and previous design. From the analysis of the result, it is found that the proposed FIR filter using DA technique gives a superior performance as compared with previous algorithm.

C. Synthesis Utilization

Device utilization summary for FIR filter using DA technique are shown in table IV respectively. It is observed from the table that the processing unit for FIR filter using DA technique uses 5736 number of slice, 154 number of slice flip flop, 10566 numbers of LUTs, 2.187 ns minimum period and 457.247 MHz maximum frequency for Spartan-3 device family. It is similarly that the processing unit for FIR filter using DA technique uses 5736 number of slice, 154 number of slice flip flop, 10566 number of LUTs, 1.298 ns minimum period and 770.416 MHz maximum frequency for vertex-2p device family.

Table 5.1: General Comparison of Hardware and Time Complexities for B = 8 and B' = 16

Structure	FF	Adder	Multiplier	Cycle Period
Previous Design	B (N-1) + B'(N-1)	L(N-1)	NL	$T_M + T_A + 3T_{FA}$
Proposed Structure	(N-1)L + NL	L(N-1)+ (L-1) (N+2)	0	$T_A + (N-1)T_{FA}$

.

VI. CONCLUSION AND FUTURE WORK

FIR filter designed using high level system tools, shows that Verilog based filter design is better for low cost FPGA applications. System generator based FIR filter design works for high speed FPGA applications. Hence for low area designs, VHDL based design is better than System generator design. From the analysis of the results, it is found that the proposed FIR filter using DA algorithm gives a superior performance as compared with previous algorithm. It is observed from the result, shows that the B(N-1) + B'(N-1) flip flop, L(N-1) adders, NL multiplier and TM + TA + 3TFA cyclic period for previous design but (N- 1)L + NL flip flop, L(N-1) + (L-1)(N+2) adder, 0 multiplier and TA + (N-1) TFA cyclic period for proposed multiplier-less FIR filter implemented. From the analysis of the result, it is found that the proposed FIR filter using DA technique gives a superior performance as compared with previous algorithm.

FIR filters with proposed Vedic multiplier for the design and implementation of the different adder. As an extension to the present work, algorithms for composite radix could also be incorporated in the General radix algorithm to make it faster by reducing the operational complexity when the value of N is not a power of 2 or 4, but it can be expressed as a multiplication of various radices. In this thesis, two efficient designs of Adders are proposed which provides much optimized results in terms all important parameters considered. It would be interesting to implement higher bits modified Kogge-stone adder. This proposed architecture of Adders will be very efficient in various applications like Digital Electronics, Signal Processing and Robotics as Floating Point Arithmetic. Designed high speed convolution technique can use for designing the filter to filter the blurred signal. This can be used for image 49 processing and digital signal processing technique. By using high speed multiplier ALU, GPU and CPU can be designed. The algorithms presented may be extended to multidimensional data

REFERENCES

[1] 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.04, pp. 01-09, April 2016.

[2] Indranil Hatai, Indrajit Chakrabarti, and Swapna Banerjee, "An Efficient VLSI Architecture of a Reconfigurable Pulse-Shaping FIR Interpolation Filter for Multi-standard DUC", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 23, No. 6, pp. 345-353, June 2015.

[3] Ravi H Bailmare, S. J. Honale and Pravin V Kinge, "Design And Implementation of Adaptive FIR Filter using Systolic Architecture", in International Journal of Current Engineering and Technology , Vol.4, No.6,pp. 143-152, June 2014.

[4] M. Usha, R. Ramadoss, "An Efficient Adaptive Fir Filter Based On Distributed Arithmetic", International Journal of Engineering Science Invention, Vol. 3, Issue. 4, pp. 15-20, April 2014.

[5] B. K. Mohanty, P. K. Meher, S. Al-Maadeed, and A. Amira, "Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters," IEEE Transaction Circuits System. I, Reg. Papers, Vol. 61, No. 1, pp. 120–133, Jan. 2014.

[6] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of a DAbased reconfigurable FIR digital filter," IEEE Transaction Circuits System II, Exp. Briefs, Vol. 61, No. 7, pp. 511–515, July 2014.

[7] Sang Yoon Park and Pramod Kumar Meher, "Low power, High-throughput And Low- Area Adaptive FIR Filter Based on Distributed Arithmetic", in IEEE Transactions On Circuits And Systems-ii, Vol. 60, No. 6, pp. 346- 350, 2013.

[8] Basant K. Mohanty, And Pramod Kumar Meher, "A High-Performance Energy- efficient Architecture For FIR Adaptive Filter Based on New Distributed Arithmetic Formulation Of Block LMS Algorithm", In IEEE Transactions On Signal Processing, Vol. 61, No. 4, pp. 01-08, February, 2013.

[9] Pallavi Saxena, Urvashi Purohit, Priyanka Joshi, "Analysis of Low Power, Area Efficient and High Speed Fast Adder", International Journal Of Advanced Research In Computer And Communication Engineering, Vol. 2, Issue 9, pp. 01-09, September 2013.

[10] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," IEEE Transcation Computer- Aided Design Integral Circuits System, Vol. 29, No. 2, pp. 275–288, Feb. 2010.

[11] P. K. Meher, "New approach to look-up-table design and memory based realization of FIR digital filter," IEEE Transaction Circuits System I, Regular Papers, Vol. 57, No. 3, pp. 592–603, Mar. 2010.

[12] K. Meher, S. Chandrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," IEEE Transaction Signal Processing, Vol. 56, No. 7, pp. 3009–3017, July 2008.

[13] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters," IEEE Transaction Computer-Aided Design Integral Circuits System, Vol. 27, No. 2, pp. 217–219, Feb. 2008.

[14] A. P. Vinod and E. M. Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers," IEEE Trans. Wireless Communication, Vol. 7, No. 5, pp. 1669–1675, July 2006.

[15] K.-H. Chen and T.-D. Chiueh, "A low-power digit-based reconfigurable FIR filter," IEEE Transaction Circuits System II, Exp. Briefs, Vol. 53, No. 8, pp. 617–621, Aug. 2006.