JES
Journal of Engineering Sciences

Vol 11, Issue 1, Jan /2020
ISSN NO:0377-9254

A NOVEL FIR FILTER DESIGN FOR DSP APPLICATIONS

[1]Dr. M Nagamani, [2]Dr. D SubbaRao, [3]M Pushpa Latha
[1]PG Scholar, MTech, Dept of ECE, Siddhartha Institute of Engineering & Technology, Ibrahimpatnam, T.S.
nagamani.maddipatla@gmail.com
[2]Professor & HOD, Dept of ECE, Siddhartha Institute of Engineering & Technology, Ibrahimpatnam, T.S.
[3]Asst Professor, Dept of ECE, Siddhartha Institute of Engineering & Technology, Ibrahimpatnam, T.S.

ABSTRACT - The finite impulse response filter (FIR) is the most primary aspects in digital signal processing systems and is widely used in communications, image processing, and pattern recognition. FIR filter design using FPGA not only improves the process speed and overcomes integration difficulties but also achieves fixed function DSP-specific chip din real time. It also achieves DSP processor flexibility by building it with FPGA. The mixture of FPGA and DSP science can similarly enhance integration, expand work speed and make bigger gadget capabilities. Though 32-bit FIR filter design is considered primarily, it is extended to 64-bit to merit the process in a better way.
Keywords: FIR, Xilinx, FPGA, Synthesize, Implementation, Simulation.

I. INTRODUCTION

With the extensive use of the PLDs, high-speed digital signal processing (DSP) technology has additionally achieved accelerated speed and integration, which affords new methods for DSP modules implementation. In practice, the signal processing ought to be timely and accurate otherwise it will lose its meaning. Hence, rapidity and reliability became important indicators of real-time digital signal processing devices and processes.
Design of FIR digital filter implementation on FPGA is also one of such. It solely ensures the correct strict linear section characteristic with simple structure and stable. It can be easily implemented using high end software languages like C, C++ etc in digital form. FPGA is one form of typical PLD, its logical block association is normal and the connection useful resource is rich, concurrently its architecture of LUT is be applicable to implement real-time, high-speed and dependable FIR filter. Therefore, FIR filter module based on FPGA chip format has extremely good advantages.

One of the key technologies of EDA is to layout digital hardware gadget with hardware description language (HDL). Currently Verilog HDL is the most extensively used hardware description language. Verilog HDL is one of the most broadly used hardware description languages. Its hardware description capacity is very effective from the good judgment gate level, circuit degree to gadget degree and different ranges can be described and modelled, such as a counter, a storage system, a microprocessor, becoming the enterprise preferred hardware description language. In addition, it can elevate on the simulation, the synthesis and the debugging are high quality to the circuit function and the enhancement of the structure, can shorten the sketch cycle greatly, reduces the expense. Most importantly, the format of the Verilog HDL language has nothing to do with the specific hardware, thereby lowering the situation of designing the hardware circuit, permitting impartial sketch and bendy description. Therefore, under the guide of EDA technology, reconfiguring the internal hardware structure and working mode of the FPGA chip is extra timesaving, cost-saving, excellent flexibility, and exact transplant ability. Using Verilog HDL as a description method, a technique of imposing a FIR filter with an FPGA is studied, which outcomes in higher performance, decrease scale and lower cost.

The primary task of FIR filter section in the LTE- based downstream transmitter for FPGAs. In present day electronic circuit design, occupy a fantastically small amount of resources and sources in this rather fast with the quickest velocity is the sketch direction. In the existing gadget FIR is applied in DSP filter.

Design of FIR Filter Based on FPGA Chip, the FPGA algorithms can increase speed and minimize the execution time. The FPGA gadgets use the Spartan-6, which function with extraordinarily high overall performance and density and are optimized for complete system power. Altera's special redundancy science notably improves throughput and reduces element costs. And at the equal time get higher performance and better signal integrity.

Once the implementation of the FIOR filter on FPGA is successful, single FPGA can be sufficient for entire system design. As FIR filter is only a small part of DSP system and as it can be designed on FPGA along with DSP circuit integration of both can be easily accomplished without any leakage issues. As both are designed on the same chip overall performance of the DSP system will be greatly improved and space requirement also will be less.

Chapter 1 gives introduction to project details. It briefly explains about FIR filter and its fundamentals. data integration and its evolution, about uncertainty/error and its propagation over various factors, heterogeneity and its forms, chosen study area, reasons for choosing it and objectives planned to achieve, how much achieved in the research time and what can be done further. Chapter 2 is the literature review part. This explains various concepts studied during project execution process and their details. Reference to the previous works was made wherever relevant. Chapter 3 deals with the design methodology of FIR filter. Chapter 4 summarizes the circuit outputs and reports of Verilog programs execution. Chapter 5 concludes the report by providing further research scope of the project.

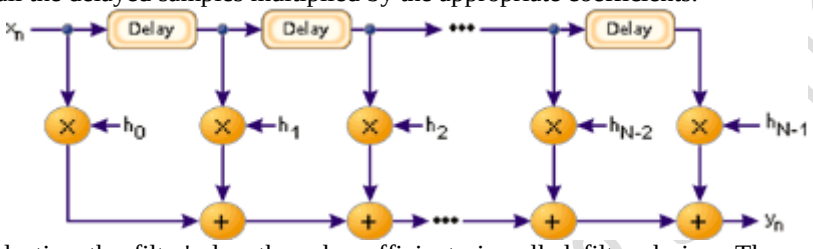
II. LITERATURE REVIEW

Filters are signal conditioners. Each filter functions by accepting an input signal, blocking pre-specified frequency components, and passing the original signal minus those components to the output. A digital filter takes a digital input, gives a digital output, and consists of digital components. In a typical digital filtering application, software running on a digital signal processor (DSP) reads input samples from an A/D converter, performs the mathematical manipulations dictated by theory for the required filter type, and outputs the result via a D/A converter.

Finite impulse response (FIR) filters are the most popular type of filters implemented in software. It is a non-recursive type of filter and the output depends only on past and present inputs. It does not have feedback. Hence FIR filter is inherently more stable. Another main advantage of FIR filter is it provides linear phase response. In order to achieve linear phase FIR filter, we must maintain symmetry in time domain.

$$b(n) = \pm b[M-1-n]$$

Figure given below shows the basic block diagram for an FIR filter of length N. The delays result in operating on prior input samples. The hk values are the coefficients used for multiplication, so that the output at time n is the summation of all the delayed samples multiplied by the appropriate coefficients.



The process of selecting the filter's length and coefficients is called filter design. The goal is to set those parameters such that certain desired stop-band and pass-band parameters will result from running the filter.

Frequency response plot of a typical FIR filter is given below which verifies that the filter meets the desired specifications, including ripple and transition bandwidth. As per the filter's length and coefficients, the longer the filter (more taps), the more finely the response can be tuned.

Various FIR realization structures are

- Direct form structure
- Linear phase structure
- Cascade form structure
- Complementary structure
- Miscellaneous structure

A digital filter can be implemented using only three elements. They are

- Addition
- Multiplication by a constant (in case of filter it is coefficient
- Delay blocks

FIR filter implementation is generally done in two ways

(i) The Window Method

The basic idea behind windowing method is to choose a proper ideal frequency selective filter which always has a non-casual infinite duration impulse response and then to truncate (window) it's impulse response to obtain linear phase and casual FIR filter. Hence the emphasis in this method is on selecting an appropriate windowing function and appropriate ideal filter.

(ii) Frequency sampling technique

The approximation error would then be exactly zero at the sampling frequencies and would be finite in frequencies between them. The smoother the frequency response being approximated; the smaller will be the error of interpolation between the pattern points.

One way to decrease the error is to enlarge the quantity of frequency samples [Rab75]. The other way to enhance the exceptional of approximation is to make a number of frequency samples specified as unconstrained variables. The values of these unconstrained variables are normally optimized via laptop to decrease some simple characteristic of the approximation error e.g. one would possibly pick out as unconstrained variables the frequency samples that lie in a transition band between two frequency bands in which the frequency response is precise e.g. in the band between the passband and the stopband of a low bypass filter.

Thus, a given band aspect frequency might also be nearer to type-II frequency sampling factor that to type-I in which case a type-II sketch would be used in optimization procedure.

III. FILTER DESIGN METHODOLOGY

The major task proposed is FIR filter design using high end FPGA technology. As explained above this implementation improves the speed of the operation and avoids circuit redundancy in FPGA. Challenge is the FIR filter phase of the LTE- based totally downstream transmitter for FPGAs. In modern electronic circuit design, occupy a enormously small amount of resources and assets in this especially quickly with the fastest speed is the sketch direction. Design FIR Filter Based on FPGA Chip, the FPGA algorithms can increase speed and reduce the delay time. The FPGA devices use the Stratix II household of EP2S60F1020I4 devices, which feature extraordinarily excessive performance and density and are optimized for complete machine power. Altera's special redundancy technology radically improves throughput and reduces thing costs. And at the identical time get higher overall performance and higher sign integrity.

Following is the flow of the design process

- Construction of algorithm for designing FIR filter module based on FPGA chip
- Writing independent Verilog HDL code and simulating it with Xilinx simulation software
- Conducting overall performance evaluation.

Digital filter design involves four steps. They are

- Determining specification: Specification of the FIR filter to be designed are
  - To be implemented on a FPGA.
  - Verilog HDL scripting is to be used
  - Minimum 32-bit filter is to be designed.
- Finding a transfer function
- Choosing a realization structure: Direct form of realization structure is chosen for simplicity.
- Implementing the filter: Spartan 6 series FPGA chip (XC6SLX100) is chosen for filter implementation.

The process is divided in to 4 sub modules. It is extended to 64-bit filter as an experiment.

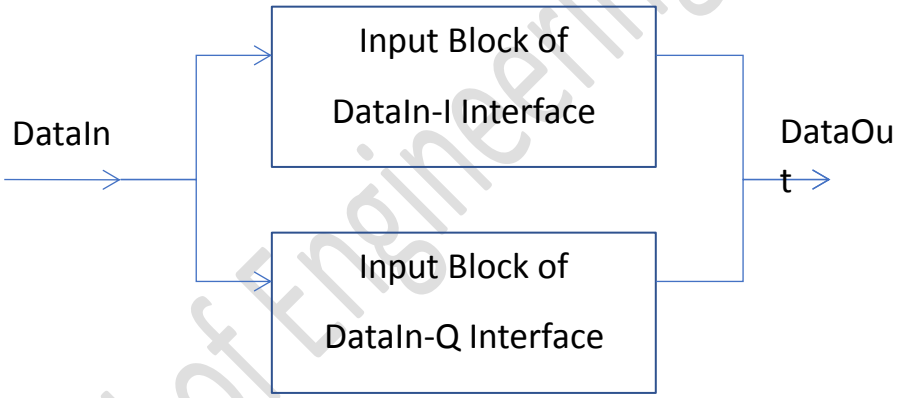


Fig. FIRonFPGA main module flowchart

As shown in the above figure The FIR filter design basically consists of two following modules. DataIn is of 32 bit which will be handled by 2 similar input blocks

- Input block for DataIn-I interface
- Input block for DataIn-Q interface.

Functionality of both the above modules is same except the bit sequence processed. Only the bit sequence of the data varies between the two blocks. Lower 16 bits will be processed by DataIn-I block and the upper 16 bits will be processed by DataIn-Q block.

In addition, it also contains a multiplexer which combines both data I and Q channels. Following is the detailed block diagram of the input block of FIRonFPGA main design implementation.

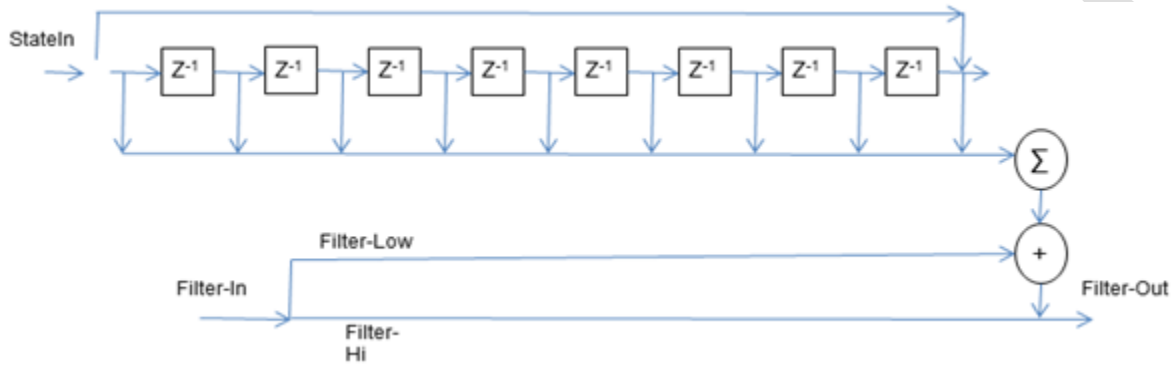
Fig. Input block of DataIN-I/DataIn-Q part

Following are the various modules of FIR filter designed using FPGA (FIRonFPGA)

- FIR8FirstI
- FIR8FirstQ
- FIR8
- FIR8Last
- InCompMux

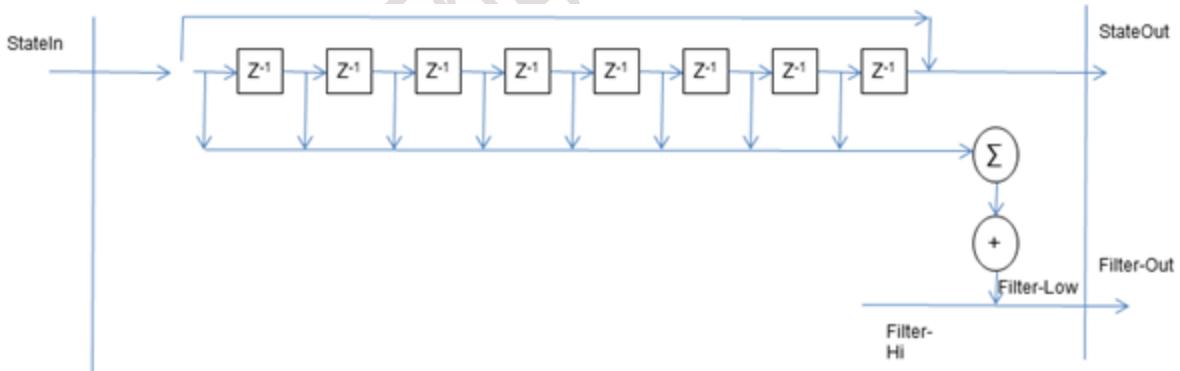
FIR8FirstI, FIR8FirstQ, FIR8, FIR8Last and InCompMux these five modules of the algorithm code are essentially the same. They all use the easiest direct network structure, with the difference being that the details of the parameters and the management of the variables are different. Out of all the above modules FIR8 is the most frequently used module and algorithm understanding of this module will provide an overall understanding of the design with minor variations in the implementation of other algorithms.

FIR8



Input from stateIn is a 32-bit data, divided into extreme 16- bit as DataHi, and low 16-bit as DataLo. DataLo is shifted to the right by way of a unit delay, x0 will become x1, x0 is multiplied by Cfs (0), x1 is accelerated through Cfs (1), and at the end of dataLo is the initial value X0 is shifted respectively with CfsCi3 by cumulative operation. Filter-In is also a 32-bit input data, is also divided into Filter-Hi extreme 16 and filter-Low is lower 16 bits data. Add filter-Low and temp sign that it passes by way of multiply- accumulate in the extreme 16 bit, and then Filter-Hi by splicing, and eventually get 32-bit Filter-Out.

FIR8FirstI



FIR8First module is similar to the FIR8 module in most of the functionality. Following are the variations from FIR8 module.

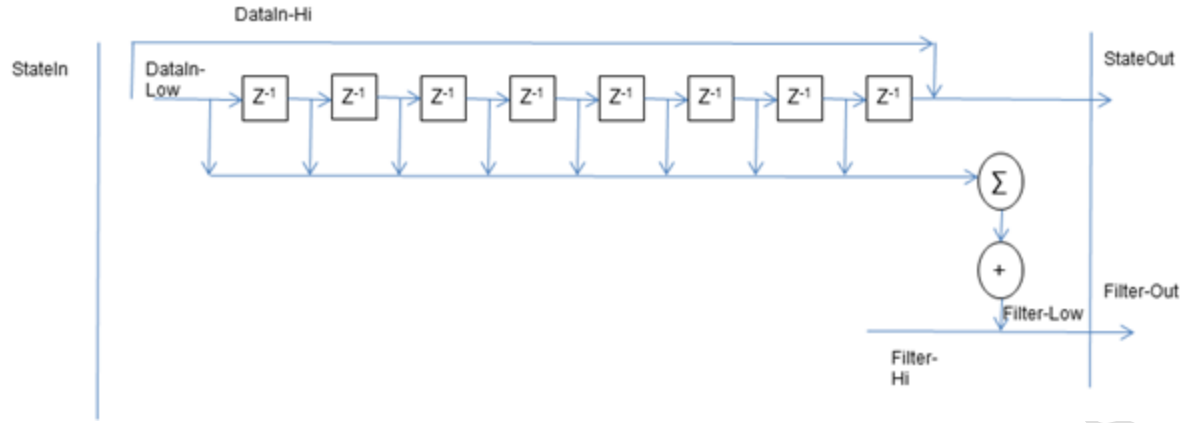
- The higher sixteen bits of the multiply-accumulate tmp sign in this module are used without delay used as the lower 16 bits of filterOut.ss
- Lower 16 bits processing is similar to FIR8.
- Both State-Out and Filter-Out components are used as inputs to the next proce in the chain.

FIRFisrt module used with extreme 16 bits is called FIR8FirstI and if used for lower 16 bits it is called FIR8FirstQ module.

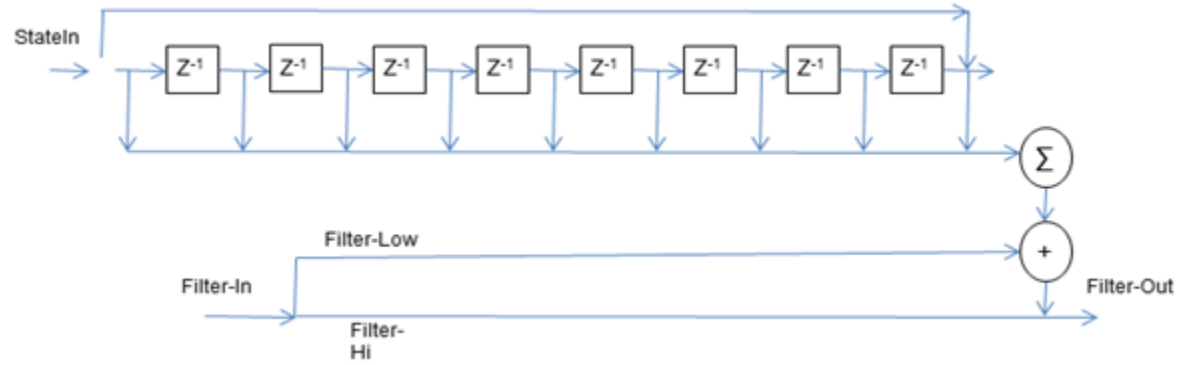
FIR8FirstQ

This module is basically the equal as the FIR8FirstI module except that it processes lower 16 bits of the data.

Journal of Engineering Sciences
Vol 11, Issue 1, Jan /2020
ISSN NO:0377-9254



FIR8Last



The algorithm of FIR8Last module is similar to that of the FIR8 module. The only variation is that the Data-Hi statistics and the Data-Low records acquired after completing 8 shifts from the input 32-bit State-In statistics are discarded, so there is no State-Out sign in the FIR8 module.

InCompMux

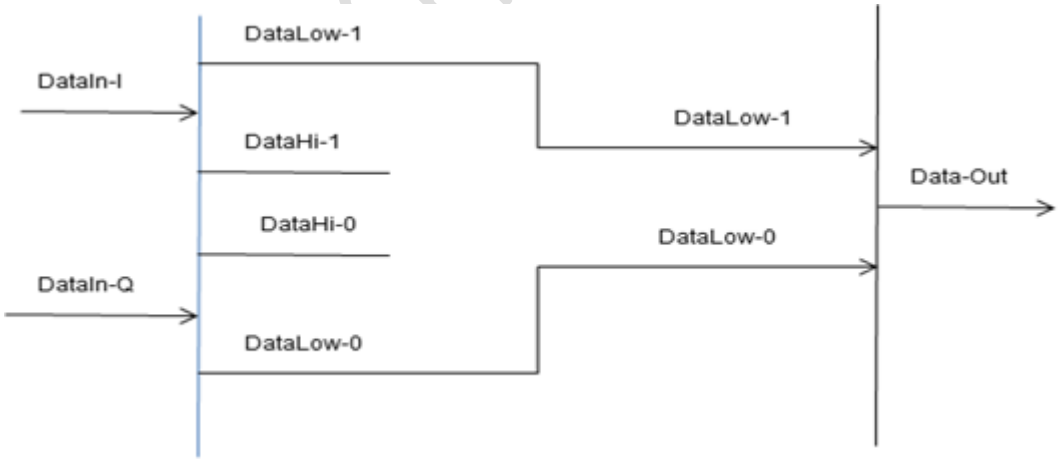


Fig: InCompMux module flowchart

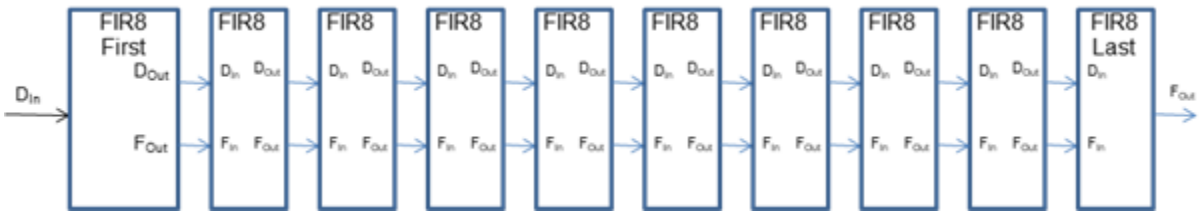


Fig: DataIn-I/DataIn-Q module flowchart

InCompMux module perform signal-splitting feature and splice functions. DataIn-I and DataIn-Q are respectively connected through Filter-Out output port of FIR8Last. First, DataIn-Q divides the DataHi-1 of the top 16 bits and the DataLow-1 of the lower sixteen bits, and then the DataInI divides the DataHi-0 of the higher sixteen bits and the DataLow-0 of the lower 16 bits. Finally, put DataLow-1 to the upper 16 bits, DataLow-0 to the lower sixteen bits, and concatenate them into the last 32-bit output signal.
DataIn-I and dataIn-Q are linked by means of filter-Out output port of FIR8Last. As shown in Figure 8, the parameters of upper and lower corresponding positions in these two strings of modules are the same.

IV. RESULTS AND DISCUSSION

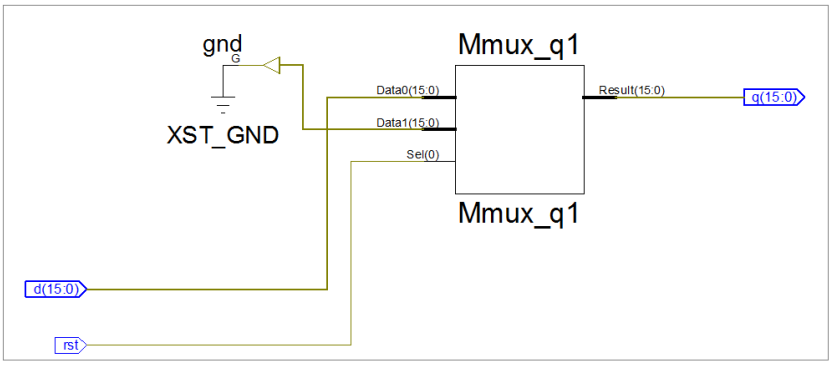


Fig. RTL schematic of dff module

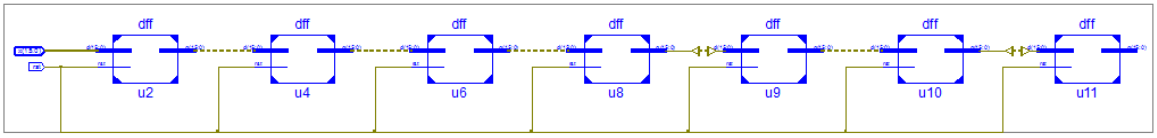


Fig. RTL schematic of bfir module

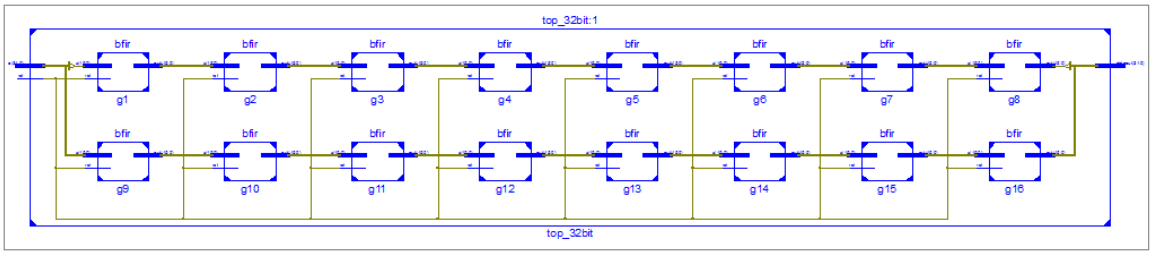


Fig. RTL schematic of 32bit_module

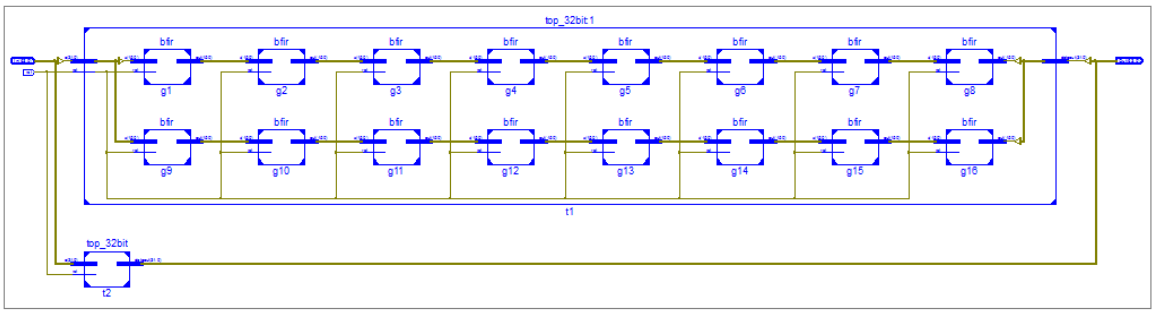


Fig. RTL schematic of 64bit_module

Following are the summary reports of individual modules. This summary provides device utilization summary, detailed and secondary reports and performance summary

top_64bit Project Status (01/06/2020 - 16:20:19)			
Project File:	work.xise	**Parser Errors:**	No Errors
Module Name:	top_64bit	**Implementation State:**	Synthesized
Target Device:	xc6slx100-3fgg484	• **Errors:**	No Errors
Product Version:	ISE 14.7	• **Warnings:**	1 Warning (0 new)
Design Goal:	Balanced	• **Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	
Environment:	System Settings	• **Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	1568	63288	2%
Number of fully used LUT-FF pairs	0	1568	0%
Number of bonded IOBs	129	326	39%
Number of DSP48A1s	128	180	71%

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Mon 6. Jan 18:14:51 2020	0	1 Warning (0 new)	0
Translation Report	Out of Date	Mon 6. Jan 16:16:51 2020	0	0	0
Map Report	Out of Date	Mon 6. Jan 16:17:29 2020			
Place and Route Report	Out of Date	Mon 6. Jan 16:17:57 2020	0	0	2 Infos (0 new)
Power Report					
Post-PAR Static Timing Report	Out of Date	Mon 6. Jan 16:18:45 2020	0	0	4 Infos (0 new)
Bitgen Report					

Secondary Reports			[-]
Report Name	**Status**	**Generated**	
ISIM Simulator Log	Out of Date	Sat 4. Jan 16:39:06 2020	

Fig. 64bit_module summary report

V. CONCLUSION

The language for designing FIR filter modules is the Verilog HDL language. Write the code in accordance to the waft chart of the module, then join all the submodules to the top-level module, and finally write testbench to furnish enter stimuli. The MODELSIM software is used to simulate the waveform, and all the verilog files of the FIR filter module are added to the new assignment thru Create a new mission in the Quartus II, select the Stratix II series chip EP2S60F1020I4. By compiling the top module FIRonFPGA, get the maximum clock frequency of the whole module is 94.04Mhz. Due to the massive range of multipliers in the module, write code via yourself will be certain to considerably expand the delay, limit the filter speed. Therefore, the FIR filter is evaluated using the mechanically generated IP core in the Quartus II software. Therefore, countless new intermediate register variables are described in the IntCompMux module of the emulation code, enabling the output allow to amplify the extend of a number of CLKs, finally obtain the motive that the High level of output allow is aligned with the output, to meet the timing concept requirements.

VI. REFERENCES

Bibliography

1. https://www.includehelp.com/dictionary/vhdl-full-form.aspx
2. https://bulletin.zu.edu.ly/issue_n15_3/Contents/E_04.pdf
3. https://engineering.eckovation.com/forum/t/whats-the-difference-between-verilog-and-vhdl/65/2
4. http://hardwarebee.com/top-10-fpga-advantages/
5. https://www.xilinx.com/products/design-tools/ise-design-suite.html
6. https://en.wikipedia.org/wiki/Finite_impulse_response
7. https://en.wikipedia.org/wiki/Verilog
8. https://www.researchgate.net/profile/Kris_Gaj/publication/228387355_Comparison_of_FPGA-targeted_hardware_implementations_of_eSTREAM_stream_cipher_candidates/links/02e7e53c6c873e20ed000000.pdf

Citations

1. Zhou Xianchum, Shi Lanfang Digital Signal Processing[M] Beijing Tsinghua University Press 2015
2. David Hwang, Mark Chaney, Shashi Karanam, Nick Ton, and Kris Gaj, 'Comparison of FPGA-Targeted Hardware Implementations of eSTREAM Stream Cipher Candidates'

Books

1. 'Understanding Digital Signal Processing' by Richard G. Lyons
2. Discrete Time Signal Processing' by Alan V Oppenheim & Ronal W Schaffer, PHI
3. Multi rate systems & Filter banks by P. P. Vaidyanathan, Pearson Education

AUTHOR PROFILE'S

M. NAGAMANI doing post-graduation in VLSI and Embedded systems from Siddhartha Institute of Engineering and Technology in Department of Electronics and Communication Engineering. Communications, New Delhi, Post-graduation in RS&GIS from JNTU, Hyderabad and Ph.D in Spatial Information Technology from JNTU, Hyderabad. She worked as Senior Scientist in NRSC for 23 years and as Project Manager in Cyient (Infotech) for 3 years.

DR. D SUBBA RAO, is a proficient Ph.D person in the research area of wireless communications from Rayalseema University, Kurnool along with initial degrees of Bachelor of Technology in Electronics and Communication Engineering (ECE) from Dr. SGIET, Markapur and Master of Technology in Embedded Systems from SRM University, Chennai. He has 16 years of teaching experience and has published 98 Papers in International Journals, 2 Papers in National Journals and has been noted under 4 International Conferences. He has a fellowship of The Institution of Electronics and Telecommunication Engineers (IETE) along with a Life time membership of Indian Society for Technical Education (ISTE). He is currently bounded as an Associate Professor and is being chaired as Head of the Department for Electronics and Communication Engineering discipline at Siddhartha Institute of Engineering and Technology, Ibrahimpatnam, Hyderabad.

M PUSHPALATHA working as associate professor in ECE branch, Siddhartha Institute of Engineering and Technology, Hyderabad, TS, India. She has completed her Master's degree with specialization in Digital Electronics and communication systems from Mahaveer Institute of Science & Technology affiliated to JNTU Hyderabad in the year 2008. Prior to this has completed bachelor's degree in Electronics and communication engineering from G. Narayanamma Institute of Technology and Science affiliated to JNTU Hyderabad. Pursuing Ph.d in the area of wireless communication, Sri Satya Sai University of technology and medical Science, Bhopal, Madhya Pradesh Her carrier started as a lecturer and has total 7 years of experience in teaching field. Out of which 3 years worked as Assistant Professor and 4 years as Senior Assistant Professor in a single organization named Abhinav Hi-Tech College of Engineering. Presently working as an Associate Professor for Siddhartha Institute of Engineering and Technology in Department of Electronics and Communication Engineering. She attended and also conducted many workshops and conferences. She is very much interest to do research on DECS, wireless technology and signals.