Performance of Improved FIR filer using FPGA

Rajendra Prasad.K

Abstract**:*****A crucial part of the digital system is the FIR filter where its framework is robust and simple to connect. In this paper, a regular FIR filter and an optimized FIR filter were modeled using window functions. The FIR filter was designed and contrived by FPGA for digital signal filtering. Also in this paper, a regular FIR filter and an optimized FIR filter were built with window usability. The main characteristic of the application is the Xillinx ISE development suite program to build the FPGA data filter accelerator. The paper also simulates the hardware and software co-designed FIR filter and provides simulation findings about hardware assets and variations in quality compared to the standard and improved FIR filter.***

Keywords:FIR filter, DSP applications, FPGA synthesis

I. INTRODUCTION

For multi-rate systems, finite impulse response (FIR) filters are of great importance. To multi-rate signal processing, the main advantage of FIR systems is inherent system stability and phase linearity. The performance of a multirate FIR filter improved considerably in contrast to the single-rate configurations using multirate techniques. FIR filter was designed and built on FPGA for digital signal filtering. This approach can be used for all FPGAs. Signal processing is an important area in which FPGAs in recent years have found a number of applications. FIR filters, for example software and hardware, can be done in two ways. The code approach is simple, as filter parameters can be adjusted to modify the filter efficiency. Nonetheless, the computer method's real-time performance is low. Alternatively, the technology is applied more efficiently and securely in real time. However, the cycle of development is longer and more complex. The article focuses on a FIR filter design approach which incorporates the advantages of software and hardware implementation through both HW and SW. It not only has good performance in real time and high reliability but also has a short development cycle, versatility and easy design.

The processing of a linear filter is focused on a multiplication method. The FIR digital filter's simple arrangement consists of several delay modules. The output of every delay unit builds up to one weight. Then we get the output of the filter. FIR filters are finite in size, M-order FIR filters can be represented as:

$$y(n)=\sum_{i=0}^{M-1}h(i)x(n-i) \qquad (1)$$

$$H(z)=\sum_{i=0}^{M-1}h(i)z^{-i} \qquad (2)$$

Manuscript published on November 30, 2019.
* Correspondence Author

Dr. Rajendra Prasad. K*, Assistant Professor, Department of ECE,MallaReddy Engineering College (A),Telangana India.

The system function is

The development of a so called FIR filter is designed to find constant impulsive response coefficients that meet the requirements.

Many approaches, including window control, frequency sampling and suitable approximation equipment are usable. The move/add operational blocks introduced in an iterative multiplier usually change the operand by a fixed number of bits and a fixed number of iterations is required for the complete multiplication operation. This simple approach guarantees a consistent power and overhead energy per device phase throughout the whole multiplication process. The new iterative multiplier structure suggested in this paper includes the use of a variable number of iterations to convert an operand from 2 to a Canonical Signed Digit (CSD) representation in real time using a new CSD recoding method where the number of iterations depends on the data. The new real-time CSD tracker is very easy to implement and only needs a few practical windows. Actually, only the control signals needed to accumulate the partial product based on the value of the multiplicands can be produced by the actual CSD representation of the multiplier operand.

The main contributions and organization of this paper are summarized as follows: In section 2 we describe background details of different FIR filtering methods adopted by the authors. Section 3 discusses the proposed work. Section 4 deliberates results and discussions. Finally, in section 5, we concluded the paper.

II. BACKGROUND WORKS

Partial products are generated and sequentially added in an iterative multiplier. For the binary-number system complement of 2 it is possible to generate and insert partial products in hardware using a basic shift/add technique as shown in Fig.1.To multiply *n*-bit by *n*-bit, right-shifting requires only an n-bit adder as opposed to the 2*n*bit adder required in a left-shifting structure. Be aware that, as shown in the fig.1 the multiplier and the lower half of the partial product can share a common register, since multiplier bits are transferred as the partial product raises.

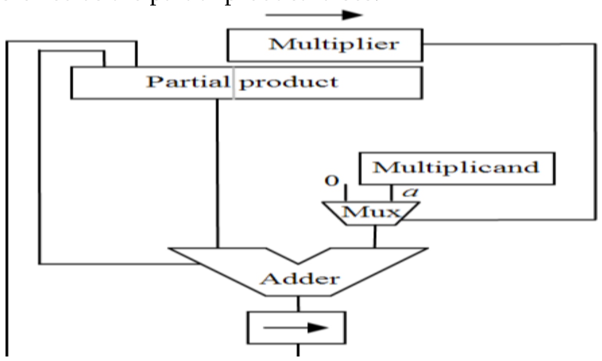

Fig.1. Schematic depiction of a right-shifting 2's complement iterative shift/add multiplier 5

In [1], the authors suggested the FIR filter design in the beamformers synthesis. As the frequency of the signal increases, the beam width of the signal also decreases and decreases. The researcher suggested that the large beam of a signal still be a component of the frequency and signal range, and a skewed beam is produced as the beam is fed to the low-pass filter for the frequency variations. The FIR is used as a solution to concentrate more. Numerous solutions to broadband beamformers are given through the application of FIR filters with an invariant frequency system. The second order cone programming approach is used to model the filter utilizing frequency invariant patterns.

In [2], the researchers addressed the different designs of FIR filters, and they concentrated on rate and power consumption. The focus is on the different applications of FIR filters. Due to its features such as linear phase response, absolute stability, etc., the FIR filter has found the broad application. FIR found usage for pulsing, channel equalization and matched filtering in each sector, but most commonly used in communication. For most of the execution of the FIR filter, it uses the preferred FPGA method, now widely used for the VLSI development.

In [3], the researchers recommended the design of multipliers using the SORIGA method and addressed the difficulty of multiplier implementations in FIR filters, thereby increasing the cost of equipment. By using successive adders instead of multipliers, this hardware difficulty can be omitted or reduced and effects are calculated. Different algorithms were built for eliminating such constraint and SORIGA uses an advanced optimization as the most essential tool in this article. SORIGA is a clever optimisation tool like some other genetic algorithms, differential algorithms and so much more. An FIR filter with power-of-two is designed in this algorithm.

In [4], the researchers provided a brief summary of electronic filters, their methods of enhancing their use over analog filters and their forms (FIR and IIR) and their use of FPGA. The numerous potential benefits of FIR filters are stressed, as the design of the FIR filter is easy and its frequency reaction is faster than the IIR filter and can be applied to any type of frequency. But it also suggested how a filter can use adders and shifters rather than multipliers to reduce the size of the process.

III. PROPOSED WORK

Standard FIR filter:

The window feature designs a generic FIR filter that fulfills the specifications. Rectangular screen, Hanning window, Hamming window, Blackman window and Kaiser window are widely used for window features. For the performance of hardware design, first of all, the use of MATLAB technology to obtain the FIR filter correlation coefficients we need.

Improved FIR filter

This introduces a new algorithm for static or programmable digital filters. The fundamental idea is tosubstitute multipliers for shifters and additives in the development of electronic multiplierless filters. A commonly used approach to replace a multiplier with shifters and adders is focused on the canonically signed numbers (CSD) multiplicand form. This allows further executing prices, though, because it avoids intermediate outcomes of multiplying. The minimum number of shift-add operations and the accompanying term to carry out the multiplication with a specified multiplicand must occur when using shifters to enforce multipliers.The signal flow chart for Real-time CSD multiplication is shown in Fig.2. It is evident from the flow chart that this encoder produces the control signals required to perform a CSD-based multiplier directly on the hardware.

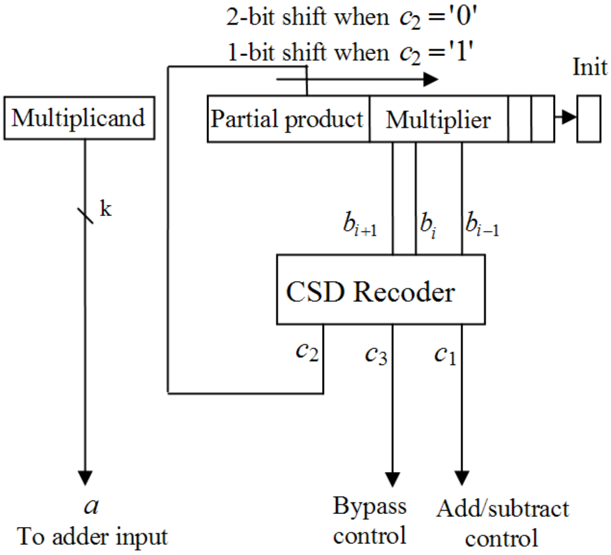

Fig.2. Real-time CSD multiplication depends on CSD recoder

The new strategy regarded here efficiently decreases the overhead implementation by avoiding the need to specifically represent the hardware CSD number; only the control signals derived from the concept of DFS numbers are explicitly displayed in the hardware.

From Fig.3 it is evident thatCSD recoderiterations can be seen that multiplier is dependent on data and uses the variable number of bits to shift. Shifting by a variation in bits usually means that a register-based shift is needed that adds time and energy consumption for each iteration; instead, shifting by a continuous number of bits by direct wire connections can be easily implemented and requires very low energy and chip area costs. Nevertheless, the model that we suggest here simply does not involve random changes, but only one or two bits shifts. It ensures that the model can be applied with a pair of hardwired switches, when the command signal chooses switching by one bit or two. This approach helps us to acquire the gains of dynamic transformation at the expense of continuous transformation. Notice that it is possible to further increase the computing rate of the current multiplier model by using specialized adders and asynchronous loop techniques.

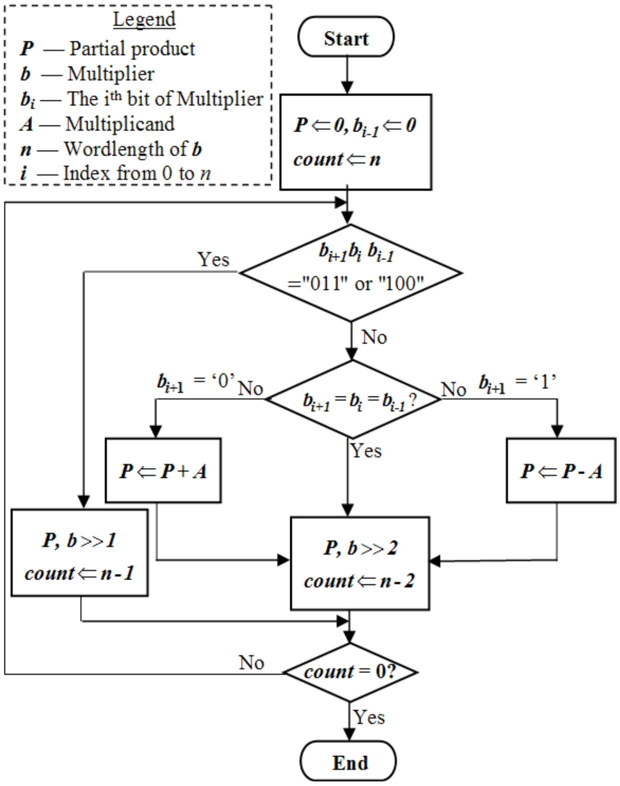

Fig. 3. Overview of CSD recoder

IV. RESULTS AND DISCUSSION

Modelsim simulation enables us to obtain the result of the simulation of hardware code, such as the clock cycle, time running, clock cycle number and errors, to determine whether the design is right. In addition, the result can be verified by simulation waveform. The efficiency of each design method is shown in Table 1.

Table 1. Exchange coefficients into 2^k mode

Method	Standard FIR	Improved FIR
Demanded clock	30M	25M
Delay estimated	15.340ns(65M)	24.562ns(40M)
DP area	65	127
clock cycle	106110	20264
Errors	**0**	0

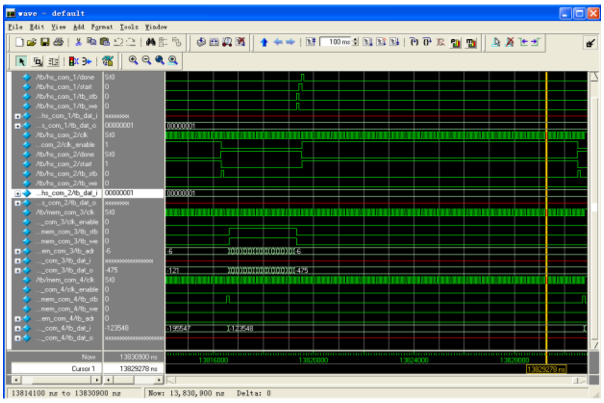

Fig. 4. Simulation output of Standard design with Isim

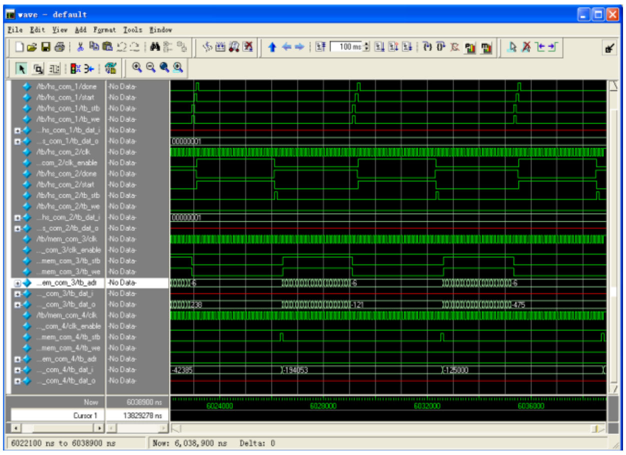

Fig. 5. Simulation output of improved design with Isim

V. CONCLUSION

This paper discusses the development methods of the FIR filter. The function of a filter in signal processing applications is to eliminate the undesirable part of the signal. It is often used to remove the random noise and remove a useful part of the input signal, for example the specific frequency component. FPGAs are high-performance tools for data processing. We investigated the development and synthesis of the FIR filter using various techniques.

REFERENCES

1. Yan S (2006) optimal design of FIR Beamformer with frequency invariant method. Appl Acoust 67:511–528.
2. Bhattacharjeea S, Sil S, Chakrabartic A (2013) Evaluation of power efficient FIR filter for FPGA based DSP applications. Proc Technol 10:856–865.
3. Chandra A, Chattopadhayay S, Ghosh B (2014) Design and implementation of SORIGA-optimized powers-of two FIR filters on FPGA. AASRI Proc 9:51–56.
4. Thakur R, Khare J (2013) High speed FPGA implementation of FIR filter for DSP applications. Int J Model Optim 3(1):92–94.
5. B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, Oxford Press, London, 1999.

AUTHOR PROFILE:

Dr. Rajendra Prasad.K is presently working as Assistant Professor in the Department of Electronics and communication Engineering, MallaReddy Engineering College (A)**,**Telangana India. His areas of interests include VLSI system Design, Low power VLSI.