Journal of Engineering Sciences

Vol 11, Issue 7,July/2020

ISSN NO:0377-9254

EFFECTIVE ERROR TOLERANT APPROXIMATE ADDER DESIGN FOR IMAGE PROCESSING APPLICATION FOR FIR FILTER

[1]AMUDALA.RUPA, [2]MANDAVILLI SRIHARI

[1]M.TECH VLSI&ES, DEPT OF E.C.E, KAKINADA INSTITUTE OF ENGINEERING AND TECHNOLOGY FOR WOMEN,KORANGI,ANDHRAPRADESH,INDIA,533461

[2]ASSISTANT PROFESSOR, KAKINADA INSTITUTE OF ENGINEERING AND TECHNOLOGY FOR WOMEN,KORANGI,ANDHRAPRADESH,INDIA,533461

ABSTRACT:

In the multiplier less realization of finite impulse response (FIR) filters utilizes a D-FF approx. adder to impart the design feature changes on the image application where region of interest and other segmentation algorithms are in use. In our design we propose a novel design of FIR filter structure. In the proposed structure of the FIR filter with 14 tap we have utilized a sequential logic to acknowledge the importance of the design structure and its delays as it provides an error tolerant scenario for the proposed approx. adder. As a result, half of the long word length STAs are replaced with adders which are named as pre-structural adders (PSTAs). PSTAs have relatively shorter word length. The HDL synthesis results shows that the delay to the output of the proposed 14-tap filter gains an improvement of 18.26% over the conventional method. The implementation is done using Verilog HDL. Simulation and synthesis are done with Xilinx ISE tool.

Keywords:finite impulse response (FIR), Xilinx, Digital signal processor (DSP),programmable logic devices (PLDs)

EXISTING DESIGN

The finite impulse response (FIR) filter is used in many digital signal processing (DSP) systems to perform signal preconditioning, antialiasing, band selection, decimation/interpolation, low-pass filtering, and video convolution functions. Only a limited selection of off-the-shelf FIR filter circuits is available; these circuits often limit system performance. Therefore, programmable logic devices (PLDs) are an ideal choice for implementing FIR filters. Altera FLEX devices, including the FLEX 10K and FLEX 8000 families, are flexible, high-performance devices that can easily implement FIR filters. For example, you can use a FLEX device for one or more critical filtering functions in a DSP microprocessor-based application, freeing the DSP processor to perform the lower-bit-rate, algorithmically complex operations. A DSP microprocessor can implement an 8-tap FIR filter at 5 million samples per second (MSPS), while an off-the-shelf FIR filter circuit can deliver 30 MSPS. In contrast, FLEX devices can implement the same filter at over 100 MSPS. This application note describes how to map the mathematical operations of the FIR filter into the FLEX architecture and compares this implementation to a hard-wired design. Implementation details— including performance/device resource tradeoffs through serialization, pipelining, and precision—are also discussed.

CONVENTIONAL FIR STRUCTURES:

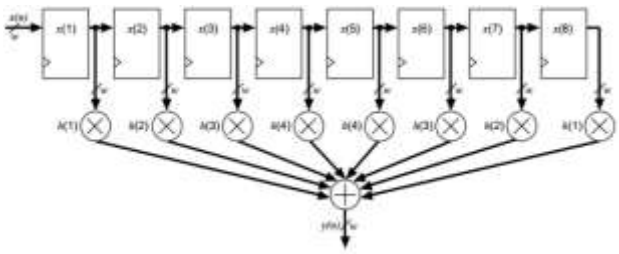


Figure1: Representing basic structure for FIR filters

The output of each register is called a tap and is represented by x(n), where n is the tap number. Each

tap is multiplied by a coefficient h(n) and then all the products are summed. The equation for this filter is:

$$y(n) = \sum_{n=1}^{8} x(n)h(n)$$

For a linear phase response FIR filter, the coefficients are symmetric around the center values. This symmetry allows the symmetric taps to be added together before they are multiplied by the coefficients. See Figure 2. Taking advantage of the symmetry lowers the number of multiplies from eight to four, which reduces the circuitry required to implement the filter.

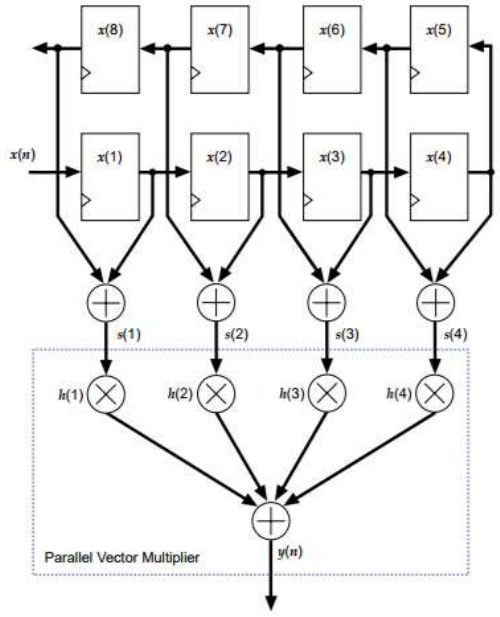


Figure 2. Block FIR filter for reconfigurable applications.

PROPOSED MODEL FOR FILTER DESIGN FOR DSP APPLICATIONS:

Structure Proposed for Fir Filter:

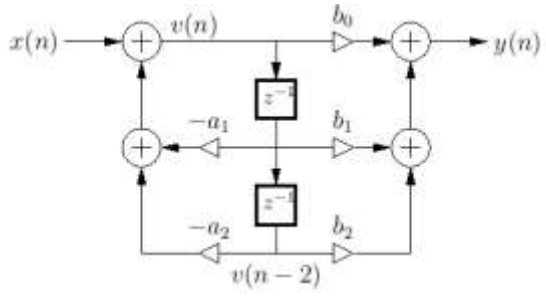


Figure 3. FIR filter structure form -II

The difference equation for the second-order DF-II structure can be written as

$$\begin{aligned} v(n) &= x(n) - a_1 v(n-1) - a_2 v(n-2) \\ y(n) &= b_0 v(n) + b_1 v(n-1) + b_2 v(n-2) \end{aligned}$$

Which can be interpreted as a two-pole filter followed in series by a two-zero filter. This contrasts with the DF-I structure of the previous section, in which the two-zero FIR section precedes the two-pole recursive section in series. Since LTI filters in series commute, we may reverse this ordering and implement an all-pole filter followed by an FIR filter in series. In other words, the zeros may come first, followed by the poles, without changing the transfer function. When this is done, it is easy to see that the delay elements in the two filter sections contain the same numbers. As a result, a single delay line can be *shared* between the all-pole and all-zero (FIR) sections. In summary, the DF-II structure has the following properties:

1. It can be regarded as a two-pole filter section followed by a two-zero filter section.
2. It is *canonical with respect to delay*. This happens because delay elements associated with the two-pole and two-zero sections are *shared*.
3. In fixed-point arithmetic, overflow can occur at the delay-line input (output of the leftmost summer in Fig), unlike in the DF-I implementation.
4. As with all direct-form filter structures, the poles and zeros are sensitive to round-off errors in the coefficients a_i and b_i, especially for high transfer-function orders. Lower sensitivity is obtained using series low-order sections (*e.g.*, second order), or by using ladder or lattice filter structures

PROPOSED OBJECTIVES:

1. Optimize Error tolerant algorithm for approx. adder
2. FIR filter design using Approx. adder
3. Improved scenario for image Thresholding algorithm for image segmentation.

ADDER DESIGN AND ITS CIRCUIT DIAGRAM:

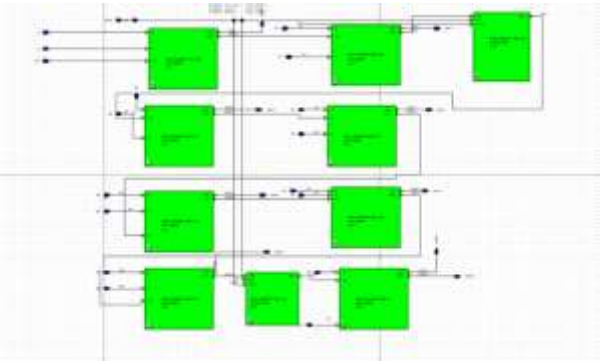


Figure 4: Block diagram for approx. adder.

The current design approaches with case of the D-FF with a scenario of the approx. carry additions and saving at one of the logics in 8 bit design. This incorporates a leaser view of the current design utilizing the different adder model and other application scenario. As per the abstract point of view we have acknowledged the design for the FIR filter using proposed approx. adder for the additions and input scenario. As the adder design we have utilized a formulation on the current design estimating the power area and delay at each filter stage using hybrid approach of the kogge stone tree and Wallace tree on carry propagation adder.

Block Diagram for the proposed Architecture:

The current design model for the FIR filter based on the approx. adder would suggest a filter with 16 bit design and 14 tap filter. Here the tap representing the stage of the filter and for each such filter stages we have the input bit of length 16 bits.

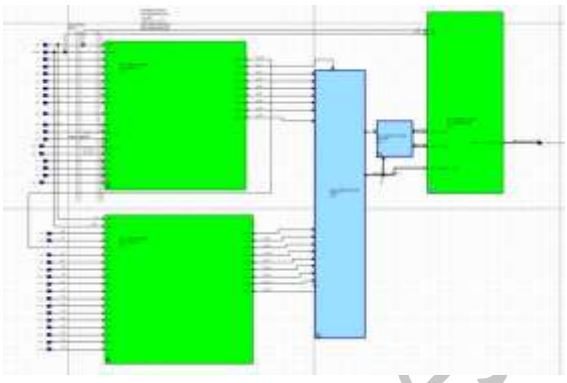


Figure 5: Block diagram for final FIR FILTER using approx. adder.

The error tolerance of the approx. adder is controlled using FF at different position of the carry at each stage of 8bit addition resulting carry save and approximation using conditional approach for output observed at each addition. Design and its analysis of the proposed circuit consists of 16 bits of data from a0-a15 and b0-b15 and after addition of these data which are stored with 16 bit data one and data 2 resulting in threshold values for each data input. These values are utilized for the FIR filter for coefficients of 14 tap improving the delay and other performance characteristics.

RESULTS AND DISCUSSION

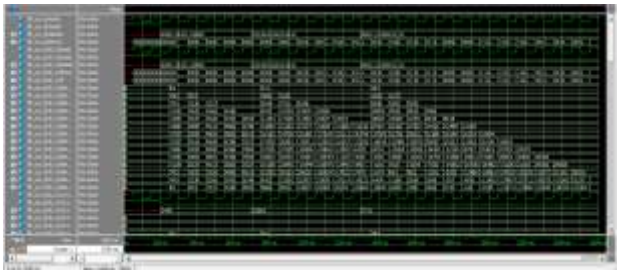
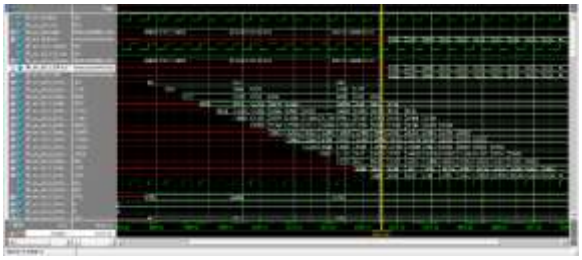


Figure 6: Correct Outputs for delay free condition:



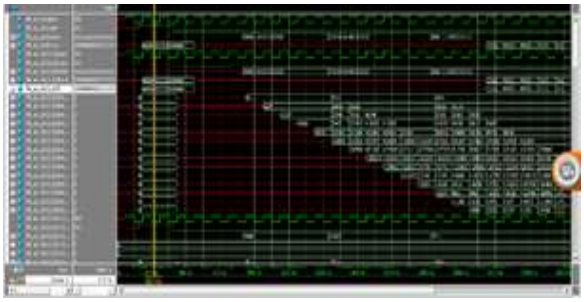


Figure 6a: Data1 consideration for the FIR filter design for 14 tap filter with 16 bit width

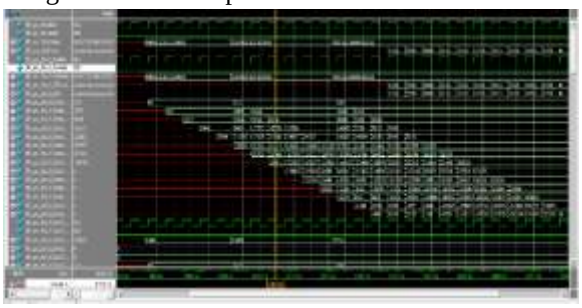


Figure 6b: Data2 consideration for the FIR filter design for 14 tap filter with 16 bit width

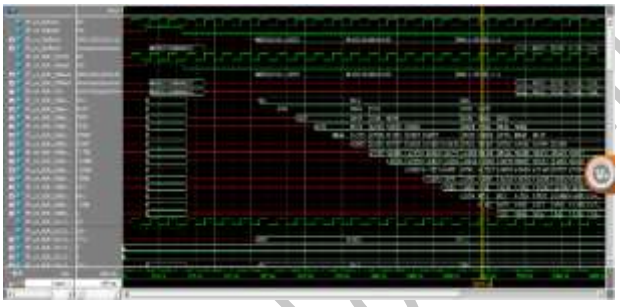


Figure 6c: Data 3 consideration for the FIR filter design for 14 tap filter with 16 bit width

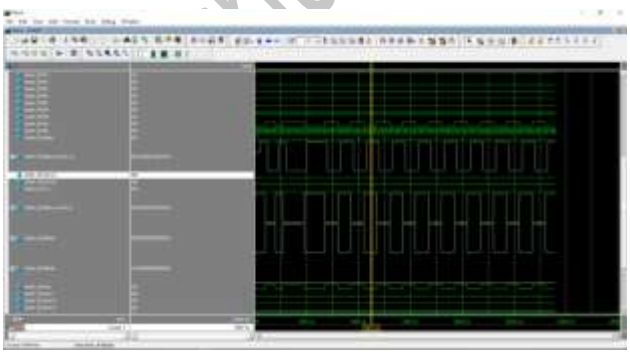


Figure 7: Representing the FIR FILTER ANALOG OUTPUT

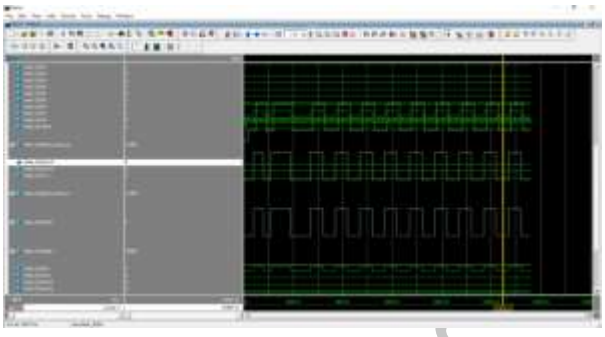


Figure 7a) Representing the threshold value at 45066.

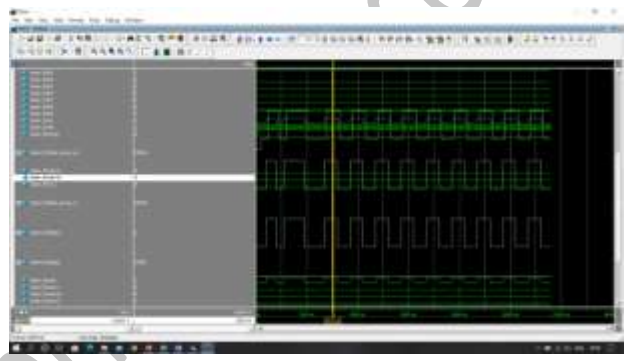


Figure 7b) Representing the threshold value at 12094.

Results discussion

1. From the above figure we have estimated and for different data streams with different simulated delay condition which would results in inappropriate data representation while operation.
2. Such delays can be overcome by representing the correct way of understanding rst condition and its impact with other stages so that each stage will be able to display the correct results as shown figure1.
3. Apart from the delays observed in FIR filter the proposed approx. adder with FIR filter design would improvise a constant square output with values ranging from 45066 to 12490. Hence resulting Thresholding approach for the FIR filter.

SYNTHESIS REPORT:

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	130	89088	
Number of Slice Flip Flops	240	178176	
Number of 4 input LUTs	223	178176	
Number of bonded IOBs	34	960	
Number of GCLKs	1	32	
Number of DSP48s	14	96	1

On-Chip Power Summary				
On-Chip	Power (mW)	Used	Available	Utilization (%)
Clocks	16.72	1	---	---
Logic	0.00	224	178176	0
Signals	0.00	606	---	---
IOs	0.00	34	960	4
DSPs	0.00	14	96	15
Quiescent	1344.95			
Total	1361.67			

Results Discussion:

Each such parameters such area power, and delay values are estimated according to the simulated results based on the correct analysis for the output verification where these parameter would play important role in shaping the design and its structures.

The below figure shows the tabulated results for existing and proposed design.

Note:

The current design is far superior from the existing design hence results in better performance as shown in the tabulated results.

CONCLUSION

FIR filters are extensively used in wired, wireless communications, video, audio processing and handheld devices are preferred because of their stability and linear phase properties. This paper presents a novel design methodology for an optimized FIR digital filters from software level to the hardware level.

The main goal is to encompass all the fields that are used in the efficient hardware realization of filters i.e. design method, selection of structure and the algorithm to reduce the arithmetic complexity of FIR filtering.

Theoretical and experimental result suggests that the power and area analysis for the current design would results in better and optimized latency for the structure implemented using direct-form structure approach is simpler, more robust to withstand the quantization errors, low cost and offers better performance than other common structures.

Proposed optimized filter implementation using an appropriate quantization scheme results in reducing arithmetic complexity, area and hardware resources. Comparison revealed that the optimized filter implementation is requiring 28% less hardware resources than the normal filter implementation.

REFERENCES

[1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications. Upper Saddle River, NJ, USA:Prentice-Hall, 1996.

[2] J. Mitola, Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering. New York, NY, USA: Wiley, 2000.

[3] P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," IEEE Trans. Signal Process., vol. 56, no. 7, pp. 3009–3017,Jul. 2008.

[4] P. K. Meher, "New approach to look-up-table design and memory based realization of FIR digital filter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 592–603, Mar. 2010.

[5] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable FIR filter for low- power and high-performance applications," IEEE J. Solid State Circuits, vol. 39, no. 2, pp. 348–357, Feb. 2004.

[6] K.-H. Chen and T.-D. Chiueh, "A low-power digit-based reconfigurable FIR filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 617–621, Aug. 2006.

[7] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," IEEE Trans. Comput.-Aided Design

Integr. Circuits Syst., vol. 29, no. 2, pp. 275–288, Feb. 2010.

[8] B. K. Mohanty and P. K. Meher, "A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm," IEEE Trans. Signal Process.,vol. 61, no. 4, pp. 921–932, Feb. 2013.

[9] B. K. Mohanty, P. K. Meher, S. Al-Maadeed, and A. Amira, "Memory footprint reduction for power-efficient realization of 2-D finiteimpulse response filters," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 1,pp. 120–133, Jan. 2014.

[10] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 61, no. 7, pp. 511–515, Jul. 2014.

[11] Ning Chi Huang,Szu Ying chen and kai Chaing Wu, "Sensor based Approximate Adder Design for Accelerating Error Tolerant and Deep learning applications"Design, Automation And Test in Europe (DATE 2019)978-3-9819263-2-3/DATE19/c 2019 EDAA