Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

*IJCSMC, Vol. 2, Issue. 4, April 2013, pg.52 – 57*

RESEARCH ARTICLE

# FPGA Design of Parallel Linear-Phase FIR Digital Filter Using Distributed Arithmetic Algorithm

**P. Karthikeyan[1], Dr. Saravanan.R[2]**

[1]Assistant Professor, Department of ECE, PSNA College of Engineering & Technology, Dindigul, India

[2]Professor, Department of CSE, PSNA College of Engg and Tech, Dindigul, India

*[1]karthickcnp@gmail.com; [2]hodcse@psnacet.edu.in*

***Abstract— Based on fast FIR algorithms (FFAs), we propose distributed arithmetic algorithm based new parallel FIR filter architectures, which are beneficial to symmetric convolutions in terms of the hardware cost. Multipliers are the major portions in hardware consumption for the parallel FIR filter implementation. The proposed new structures exploit the nature of symmetric coefficients of odd length and further reduce the amount of multipliers required at the expense of additional adders. Exchanging multipliers with adders is advantageous because adders weigh less than multipliers in terms of silicon area, and in addition, the overhead from the increase in adders in preprocessing and post processing blocks stay fixed, not increasing along with the length of the FIR filter, whereas the number of reduced multipliers increases along with the length of the FIR filter.***

***Key Terms: - FFA Algorithm; Distributed arithmetic algorithm; FIR Filter; Convolution; Digital Signal Processing***

## I. INTRODUCTION

Along the explosive growth of multimedia application, the demand for high-performance and low-power digital signal processing (DSP) is getting higher and higher. The FIR digital filter is one of the most widely used fundamental devices performed in DSP systems, ranging from wireless communications to video and image processing. Some applications need the FIR filter to operate at high frequencies such as video processing, whereas some other applications request high throughput with a low-power circuit such as multiple-input–multiple-output systems used in cellular wireless communication. Furthermore, when narrow transition band characteristics are required, the much higher order in the FIR filter is unavoidable.

In this brief, parallel processing in the digital FIR filter will be discussed. Due to its linear increase in the hardware implementation cost brought by the increase in the block size *L*, the parallel processing technique loses its advantage to be employed in practice. There have been a few papers proposing ways to reduce the complexity of the parallel FIR filter in the past [1]–[10]. In [1]–[4], poly phase decomposition is mainly manipulated, where the small-sized parallel FIR filter structures are derived first and then the larger block-sized ones can be constructed by cascading or by iterating small-sized parallel FIR filtering blocks. However, in both categories of methods, when it comes to symmetric convolutions, the symmetry of coefficients has not been taken into consideration yet, which can lead to a significant saving in hardware cost. Previously, we have investigated the design for symmetric convolutions based on even length [10]. In this brief, we will discuss symmetric convolutions based on odd length and provide new parallel FIR digital filter architectures consisting of advantageous poly phase decomposition, which can further reduce amounts of multipliers required in the sub filter section by exploiting the inherent nature of the symmetric coefficients, compared with the existing FFA fast parallel FIR filter structures.

 

## II. LITERATURE SURVEY

ISC-based linear convolution structure is transposed to obtain a new hardware efficient fast parallel finite-impulse response (FIR) filter structure, which saves a large amount of hardware cost, especially when the length of the FIR filter is large. A long convolution can be decomposed into several levels of short convolutions. After fast convolution algorithms for short convolutions are constructed, they can be iteratively used to implement the long convolution [4]. In this section, the mixed radix algorithm [7] is used to derive the generalized iterated short convolution algorithm using the Tensor Product operator in matrix form. In many signal processing applications, fast digital filtering is required. In fact, real-time signals may also be processed in this manner if the associated block-processing delay is acceptable [3]. Another potentially important application for backward filtering is the implementation of Mallat two-channel iterated filter banks based on power-complementary Butterworth filters (wavelets) [4]. The zero-phase case is often used to implement frequency-selective infinite-impulse response (IIR) filters corresponding to the squared-magnitude of the classical Butterworth, Chebyshev, and elliptic designs. However, other interesting and potentially important applications exist for non-causal IIR filters that are not zero-phase. Examples include equalizers for non-minimum- phase systems, non-causal speech models [15], half-sample interpolators, and 90-degree phase shifters such as Hilbert transformers and differentiators. On the other hand, many fast algorithms in the context of digital filtering have been obtained based on particular matrix structures [5], [16]. Many approaches to block digital filters (BDFs) design exist. Some approaches compel the BDF to be time-invariant so that conventional filter synthesis techniques can be used [13]. The best known and most widely used approach is Overlap-save [10, p. 558]. In some other approaches, no such constraint on the BDF is imposed so that the BDF can be time variant [14].

## III. FAST FIR FILTER ARCHITECTURE

Consider an *N*-tap FIR filter that can be expressed in the general form as

$$y(n) = \sum_{i=0}^{N-1} \mathbf{h(i)x(n-i)},\ \mathbf{n = 0, 1, 2, \ldots . \infty} \quad \textbf{(1)}$$

Where *x* (*n*) is an infinite length input sequence and *h* (*i*) represents the length-*N* FIR filter coefficients. Then, the traditional *L*-parallel FIR filter can be derived using poly phase decomposition as

$$\sum_{p=0}^{L-1} Y_p(z^L)z^{-p} = \sum_{q=0}^{L-1} X_q(z^L)z^{-q} \sum_{r=0}^{L-1} H_r(z^L)z^{-r} \quad (2)$$

Where

$$\boldsymbol{X_q(z) = \sum_{k=0}^{\infty} z^{-k}x(Lk+q)}$$

$$\boldsymbol{H_r(z) = \sum_{k=0}^{\frac{N}{L}-1} z^{-k}h(Lk+r)}$$ and

$$\boldsymbol{Y_p(z) = \sum_{k=0}^{\infty} z^{-k}y(Lk+p)}$$

For p, q, and r=0, 1, 2…., L-1.

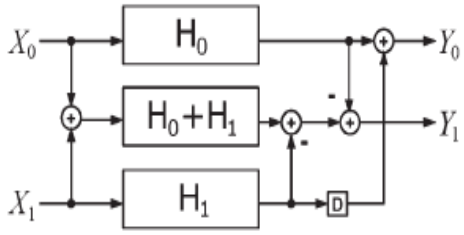


Fig. 1. Two-parallel FIR filter implementation using the FFA

 

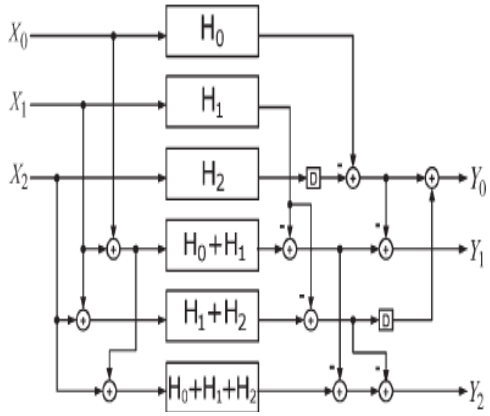


Fig. 2. Three-parallel FIR filter implementation using the FFA

2 × 2 FFA (L = 2)

According to (2), a two-parallel FIR filter can be expressed as

$$Y_0 = H_0X_0 + z^{-2}H_1X_1$$
$$Y_1 = H_0X_1 + z^{-2}H_1X_0 \quad (3)$$

However, (3) can be written as

$$Y_0 = H_0X_0 + z^{-2}H_1X_1$$
$$Y_1 = (H_0+H_1)(X_0 + X_1 - H_0X_0 - H_1X_1 \quad (4)$$

The two-parallel ($L$ = 2) FIR filter implementation using the FFA obtained from (4) is shown in Fig. 1. By the similar approach, a three-parallel FIR filter using the FFA can be expressed as

$$Y_0 = H_0X_0 - z^{-3}H_2X_2 + z^{-3}[(H_1+H_2)(X_1+X_2) - (H_1X_1)]$$
$$Y_1 = [(H_0+H_1)(X_0+X_1) - (H_1X_1)] - (H_0X_0 - z^{-3}H_2X_2)$$
$$Y_2 = [(H_0+H_1+H_2)(X_0+X_1+X_2)] - [(H_0+H_1)(X_0+X_1) - (H_1X_1)] - [(H_1+H_2)(X_1+X_2) - (H_1X_1)]$$

## IV. PROPOSED WORK

### *4.1 Architecture for 1 *N convolution and 1-bit input*

Most 2-D convolution implementations rely on the multiplying units embedded in modern FPGAs to carry out all the multiplications in parallel and to achieve great performance. However, as the kernel size increases, the number of embedded multipliers needed grows exponentially. This fact can constrain the kernel size or force to use a bigger FPGA device, which, in its turn, can yield a very high cost per operation ratio. On the other hand, a lot of work has been done on the design of multiplier less filters, mostly in the one-dimensional domain, and some authors have implemented 2-D convolution by replacing multiplications with shifting and adding operations or transforming the computation into the logarithmic domain.

The 1-bit design can be easily extended to n-bit input data through a range of approaches, from fully parallel to fully serial. For each row, it consists of n identical instances of the module. The 1-bit design can be easily extended to n-bit input data through a range of approaches, from fully parallel to fully serial.

 

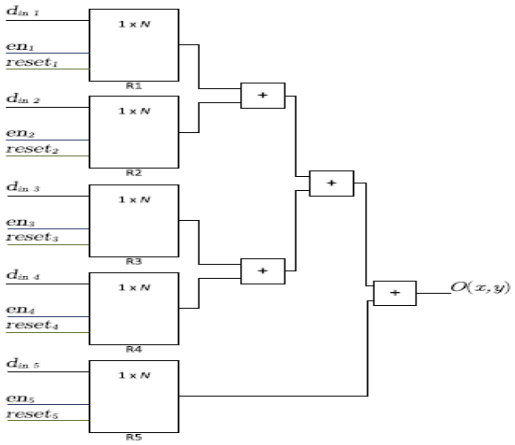


Fig. 3 Sequential Scheme for 1x N Kernel

For each row, it consists of n identical instances of the module. The outputs are all added according to the weights of their corresponding bit in the input word. Clearly, this solution will provide the highest speed at the expense of the highest resources consumption. Computations are carried out serially at bit-level in the input word using n-bit shift registers that replace flip-flops in the window buffer. The shift and add unit accumulates the partial results according to their corresponding weights in the input data item. With this configuration the increase of the resources occupancy is kept to a minimum, but the throughput is reduced by a factor n, since n clock cycles are required to generate each output result.

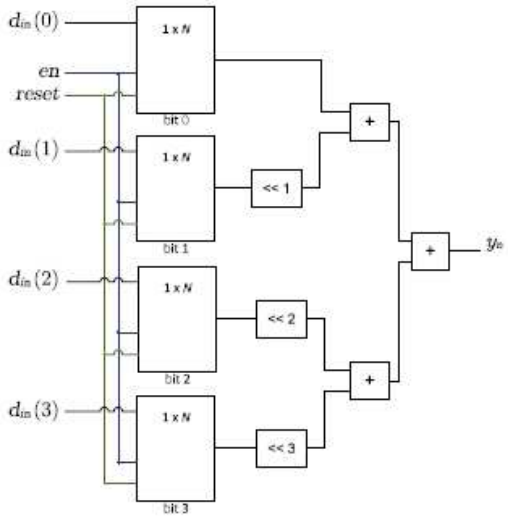


Fig. 4 Parallel Scheme for 1xN Kernel

## V. SIMULATION RESULTS

Simulation results are implemented by Modelsim software. It will give the number of reduced multipliers and adders. The below figures shows the experimental results of odd length sequence of kernel structure.

 

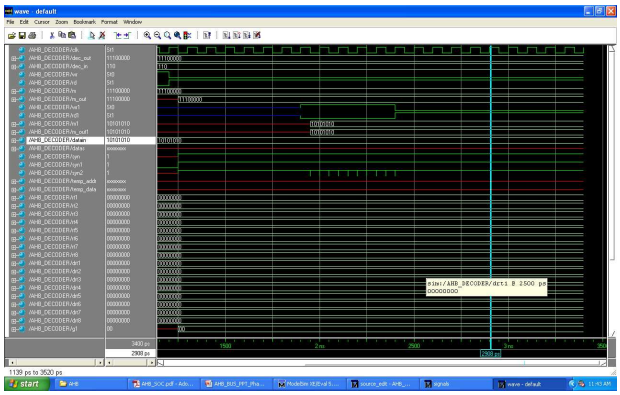

Fig. 5 Signals of kernel structure

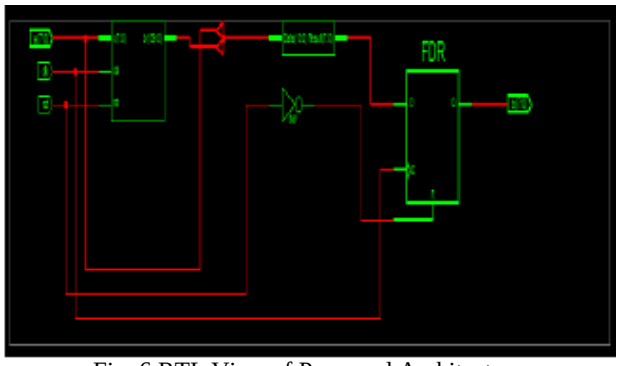


Fig. 6 RTL View of Proposed Architecture

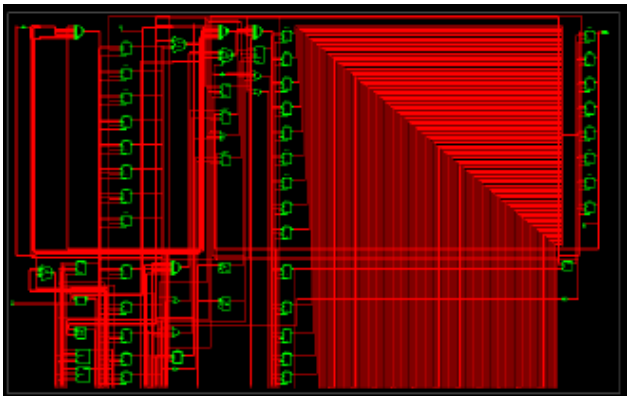

Fig. 7 Technology Schematic View

| Device Utilization Summary | | | |
|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** |
| Number of Slice Flip Flops | 682 | 9,312 | 7% |
| Number of 4 input LUTs | 1,587 | 9,312 | 21% |
| **Logic Distribution** | | | |
| Number of occupied Slices | 1,034 | 4,656 | 22% |
| Number of Slices containing only related logic | 1,034 | 1,034 | 100% |
| Number of Slices containing unrelated logic | 0 | 1,034 | 0% |
| **Total Number of 4 input LUTs** | 2,008 | 9,312 | 21% |
| Number used as logic | 1,587 | | |
| Number used as a route-thru | 21 | | |
| Number of bonded IOBs | 18 | 232 | 7% |
| IOB Flip Flops | 8 | | |
| Number of GCLKs | 1 | 24 | 4% |
| **Total equivalent gate count for design** | 22,689 | | |
| Additional JTAG gate count for IOBs | 864 | | |

Fig.8 Device Utilization Results

 

## VI. CONCLUSION

We have designed an FIR filter using distributed arithmetic Algorithm. In this work, we propose a Parallel Linear Phase FIR Filter to reduce the convolution multipliers in the FIR Filter Architecture. This can be done at 200MHz with core power of 175 mW and consumes 2008 LUTs and 682 FFs. The architecture is implemented using spartan3E family device XC3S500E using Modelsim 5.7 and Xilinx 9.2i.

## REFERENCES

[1] D. A. Parker and K. K. Parhi, "Low-area/power parallel FIR digital filter implementations," J. VLSI Signal Process. Syst., vol. 17, no. 1, pp. 75– 92, Sep. 1997.
[2] J. G. Chung and K. K. Parhi, "Frequency-spectrum-based low-area low power parallel FIR filter design," EURASIP J. Appl. Signal Process.,vol. 2002, no. 9, pp. 444–453, Jan. 2002.
[3] S. R. Powell and P. M. Chau, "Time reversed filtering in real- time," Proc. IEEE Int. Symp. Circuits Syst., vol. 2, pp. 1239–1243, May 1990.
[4] K. K. Parhi, VLSI Digital Signal Processing systems: Design and Implementation. New York: Wiley, 1999.
[5] Z.-J. Mou and P. Duhamel, "Short-length FIR filters and their use in fast nonrecursive filtering," IEEE Trans. Signal Process., vol. 39, no. 6, pp. 1322–1332, Jun. 1991.
[6] J. I. Acha, "Computational structures for fast implementation of L-path and L-block digital filters," IEEE Trans. Circuits Syst., vol. 36,
[7] J. Granata, M. Conner, and R. Tolimieri, "A tensor product factorizationof the linear convolution matrix," IEEE Trans. Circuits Syst., vol. 38, pp.1364–1366, Nov. 1991.no. 6, pp. 805–812, Jun. 1989.
[8] C. Cheng and K. K. Parhi, "Hardware efficient fast parallel FIR filter structures based on iterated short convolution," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 8, pp. 1492–1500, Aug. 2004.
[9] C. Cheng and K. K. Parhi, "Furthur complexity reduction of parallel FIR filters," in Proc. IEEE ISCAS, May 2005, vol. 2, pp. 1835–1838.
[10] C. Cheng and K. K. Parhi, "Low-cost parallel FIR structures with 2-stage parallelism," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 2, pp. 280–290, Feb. 2007.
[11] I.-S. Lin and S. K. Mitra, "Overlapped block digital filtering," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. vol. 43, no. 8, pp. 586–596, Aug. 1996.
[12] Y.-C. Tsao and K. Choi, "Area-efficient parallel FIR digital filter structures for symmetric convolutions based on fast FIR algorithm," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 2, pp. 366–371, Feb. 2010.
[13] I. S. Lin and S. K. Mitra, "Overlapped block digital filtering," IEEETrans. Circuits Syst. II, vol. 43, pp. 586–596, Aug. 1996.
[14] C. M. Loeffler and C. S. Burrus, "Optimal design of periodicallytime-varying and multirate digital filters," IEEE Trans. Acoust.,Speech, Signal Process., vol. ASSP-32, pp. 991–997, Oct. 1984.
[15] L. B. Jackson, "Noncausal ARMA modeling of voiced speech," IEEETrans. Acoust., Speech, Signal Process., vol. ASSP-37, no. 10, pp.1606–1608, Oct. 1989
[16] M. Teixeira and D. Rodriguez, "A class of fast cyclic convolution algorithmsbased on block pseudocirculants," IEEESignalProcess. Lett.,vol. 2, no. 5, pp. 92–94, May 1999

 