International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682 Vol.6, No.3, March 2017 DOI:10.15693/ijaist/2017.v6i3.43-48

ASCII IMPLEMENTATION OF AREA AND POWER MIMIMIZATION FOR DA FIR FILTER

M.Dhivyabharathi[1], K.Indhu[2], A.Kasthuri[3]

[1,2,3]UG Scholar Tejaa shakthi institute of technology for women, Coimbatore, India

E-Mail :divyam1196@gmail.com,indhuashika@gmail.com,kasthurimala@gmail.com

ABSTRACT

This paper discusses the implementation of parallel finite impulse response (FIR) filter using distributed arithmetic architecture (DAA). This architecture replace the multiply and accumulate (MAC) operation with a series Look-Up-Table(LUT) access. For high speed or low power application parallel FIR filter is used. The distributed arithmetic calculates the inner products of fixed-point data, based on the table lookups of pre calculated partial product to provide multiplication-free method. This implementation result in a high speed and low power proposed architecture based FIR filter. This filter is implemented in cadence to provide minimum power and area or delay. This proposed method provide a result of average reduction of 60% in the number of LUT, 40% reduction in occupied slices and 50% reduction in the number of gates for parallel FIR filter implementation.

Key words: Block processing, finite impulse response (FIR) filter, reconfigurable architecture, VLSI

1. INTRODUCTION

The Finite Impulse Response (FIR) filter is a digital filter widely used in Digital Signal Processing applications in various fields like imaging, instrumentation, communications, etc. Programmable digital processors signal (PDSPs) can be used in implementing the FIR filter. However, in realizing a large-order filter many complex computations are needed which affects the performance of the common digital signal processors in terms of speed, cost, flexibility, etc. In signal processing, a finite impulse response filter is a filter whose impulseresponse (or response to any finite length input) is of finite duration, because it settles to zero in finite time. Finite Impulse Response filters are important building blocks for various Digital Signal Processing (DSP) applications.

A finite impulse response filter is a filter structure that can be used to implement almost any sort of frequency response digitally. An FIR filter is usually implemented by using a series of delays, multipliers, and adders to create the filter's output.

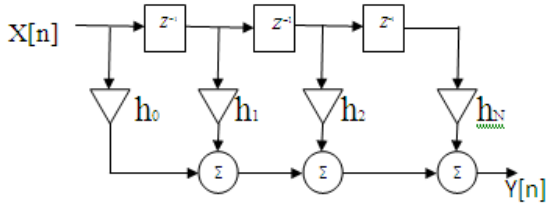

Figure 1. 1The logical structure of an FIR filter

Figure 1 shows the basic block diagram for an FIR filter of length N. The delays result in operating on prior input samples. The h_k 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. Most engineers utilize a program such as MATLAB to do their filter design. But whatever tool is used, the results of the design effort should be the same:

- A frequency response plot verifies that the filter meets the desired specifications, including ripple and transition bandwidth.
- The filter's length and coefficients.

The longer the filter (more taps), The more finely the response can be tuned.

1.1 FIR FILTER REPRESENTATION

A FIR filter can be mathematically expressed by the equation

$$y[n]= \sum_{i=0}^{N-1} h[i]x[n-i]$$

where,

- X represents the input signal,
- H the filter coefficients,
- Y the output signal,
- Y[n] is the current output sample, and
- N is the number of taps of the filter.

1.2PIPELINE AND PARALLEL PROCESSING

1.2.1 PIPELINING

Pipelining is an important technique used in several applications such as digital signal processing (DSP) systems, microprocessors, etc. It originates from the idea of a water pipe with continuous water sent in without waiting for the water in the pipe to come out. Accordingly, it results in speed enhancement for the critical path in most DSP systems. For example, it can either increase the clock speed or reduce the power consumption at the same speed in a DSP system.

CONCEPT

Pipelining allows different functional units of a system to run concurrently. Consider an informal example in the following figure. A system includes three sub-function units (F_0, F_1 and F_2). Assume that there are three independent tasks (T_0, T_1 and T_2) being performed by these three function units. The time for each function unit to complete a task is the same and will occupy a slot in the schedule.

If we put these three units and tasks in a sequential order, the required time to complete them is five slots.

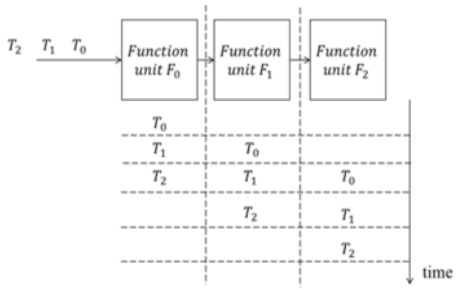

Figure 1.2 Sequential function unit

However, if we pipeline T_0 to T_2 concurrently, the aggregate time is reduced to three slots.

1.3 PARALLEL PROCESSING

Parallel processing and pipelining techniques are duals each other: if a computation can be pipelined, it can also be processed in parallel. Both of them exploit concurrency available in the computation in different ways. Parallel processing system is also called block processing, and the number of inputs processed in a clock cycle is referred to as the block size. In digital signal processing, parallelprocessing is a technique duplicating function units to operate differenttasks(signals)simultaneously. Accordingly, we can perform the same processing for different signals on the corresponding duplicated function units. Further, due to the features of parallel processing, the parallel DSP design often contains multiple outputs, resulting in higher throughput than not parallel.

Any system that can be pipelined can also be processed in parallel. In pipelining independent computations are executed in an interleaved manner, while parallel processing achieves the same using duplicate hardware. Parallel processing systems are also referred to as block processing systems. The block size indicates the number of inputs processed simultaneously

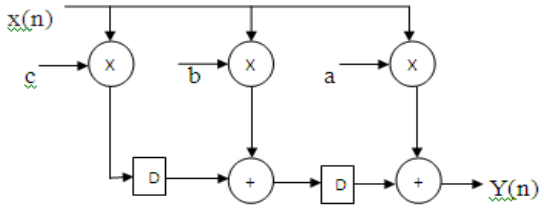

Figure 1. 3 Data broadcast implementation of 3-tap FIR filter

A complete parallel processing system shown in figure contains a serial to parallel converter (DMUX) the MIMO processing block and a parallel to serial converter (MUX). The data paths in the MIMO system either work with an offset of T_{clock}/M in a M-parallel system or the MUX and DMUX must be equipped with delay units allowing simultaneous processing. The throughput of a M-parallel system is M times the throughput of the sequential system,

$R_{T,M} = M.R_{T,1}$.

It should also be noted that for a parallel processing system $T_{clock} \neq T_{sample}$ whereas they are equal in pipeline system.

Consider a function unit (F_0) and three tasks (T_0, T_1 and T_2). The required time for the function unit F_0 to process those tasks is t_0,t_1 and t_2 respectively. Then, if we operate these three tasks in a sequential order, the required time to complete them is $t_0 + t_1 + t_2$.

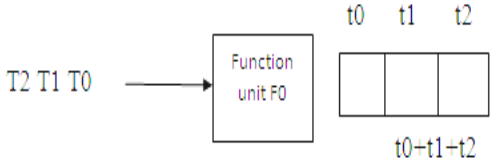

Figure 1. 4 Sequential function unit

However, if we duplicate the function unit to another two copies (F), the aggregate time is reduced to $\max(t_0,t_1,t_2)$, which is smaller than in a sequential order.

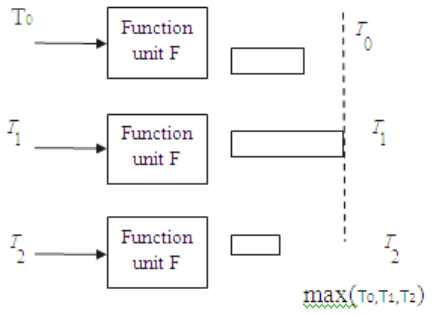

Figure 1.5 Pipelined function unit

Advantages of pipelining and parallel processing

- High speed
- Low power

When sample speed does not need to be increased, these techniques can be used for lowering the power consumption.

It has already been shown that pipelining and parallel processing can increase the sample speed. Now consider use of these techniques for lowering the power consumption where sample speed does not need to be increased. The propagation delay T_{pd} is associated with charging and discharging of the various gate and stray capacitances in the critical path

2. PROPOSED TECHNIQUE

2.1 PARALLEL FIR FILTER TECHNIQUE

The proposed DA based parallel filter architecture is shown in Fig. 4.1 for 256-tap parallel FIR filter. This architecture uses the concept of multiplexer based DA filtering scheme. The particular 2-input LUT is selected which represent all the possible sum combinations of filter coefficients. It implies about 50% reduction in the number of LUT used with increased speed. There are two main aspects to be considered when designing a parallel filter, namely the number of bits required for the signal and the required transfer function of the filter. The former one determines the word length of the entire data path. The later one is determined by two parameters, namely the number of taps, and the number of bits in each coefficient. The multipliers are the most expensive blocks in terms of area, delay, and power in a FIR filter when considering custom implementation. In fact, even for a dedicated implementation of constant-coefficient multipliers, the amount of hardware needed is very high as we have several multipliers in the entire filter. To evaluate the performance of the proposed scheme, 4-tap, 8-tap, 16-tap, 32-tap, 64-tap, 128-tap and 256-tap parallel FIR filters are implemented. The results are compared with the conventional parallel FIR filtering scheme and existing DA based implementation.

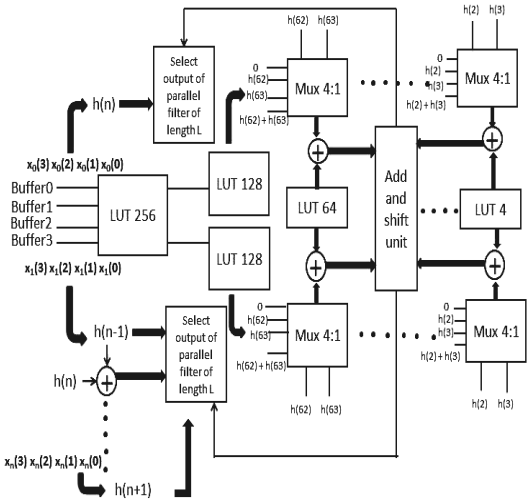

Figure 1.6 Proposed parallel digital FIR filter architecture.

2.2 DISTRIBUTED ARITHMETIC

Distributed Arithmetic was first brought up by Croisier and was extended to cover the signed data system. Then it was introduced to save MAC blocks with the development of FPGA technology. High performance FIR filter based on DA using LMS architecture. An efficient technique for calculation of sum of products or vector dot product or inner product or multiplies and accumulates (MAC). MAC operation is very common in all Digital Signal Processing Algorithms. The advantages of DA are best exploited in data-path circuit designing. Area savings from using DA can be up to 80% and seldom less than 50% in digital signal processing hardware designs. DA efficiently implements the MAC using basic building blocks (Look Up Tables). The "basic" DA technique is bit-serial in nature. DA is basically a bit-level rearrangement of the multiply and accumulate operation.

If h[n] is the filter coefficient and x [n] is the input sequence to be processed, the N -length FIR filter can be described as final form of distributed arithmetic as,

Table 1.1 device utilization

No of slices	No of slices flip flops	No of 4 input LUTs	No of bonded IOB s	No of GCLK s
2448	4896	4896	158	24

The basic LUT-DA scheme consists of three main components: the input registers, the 4-input LUT unit and the shifter/accumulator unit.

Additionally, it would require a control unit to manipulate the filter operation, and an adder tree unit to perform addition on partial filter results. Applying this approach the 4-input LUT unit will not be directly accessed instead 2-input LUT is used based on multiplexer select.

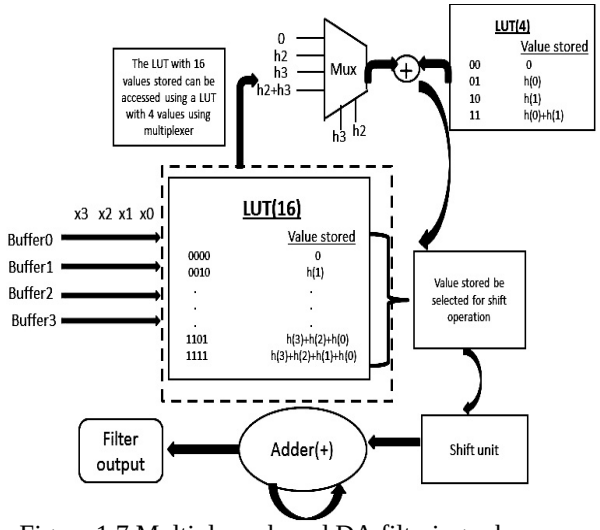

Figure 1.7 Multiplexer based DA filtering scheme.

RESULT AND DISCUSSION

Xilinx designs, develops and markets programmable logic products including integrated circuits (ICs), software design tools, predefined system functions delivered as intellectual property (IP) cores, design services, customer training, field engineering and technical support. Xilinx sells both FPGAs and CPLDs programmable logic devices for electronic equipment manufacturers in end markets such as communications, industrial, consumer, automotive and data processing.

DEVICE UTILIZATION

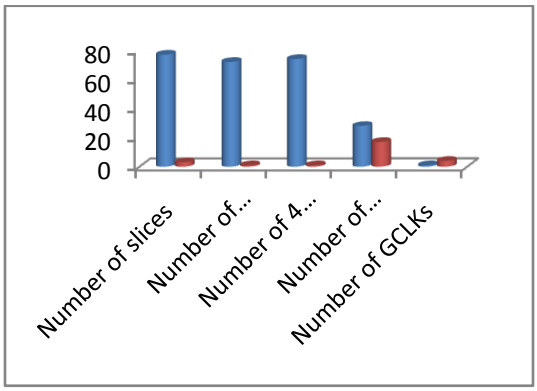

Figure 1.8 Device Utilization

AREA UTILIZATION

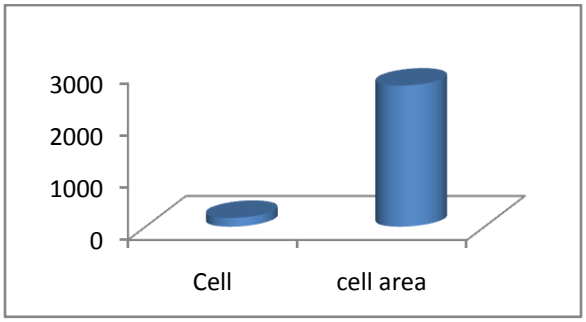

Fig 1.9Area utilization

Table 1.2 REPORT AREA

Cell	Cell Area
168	2701

POWER UTILIZATION

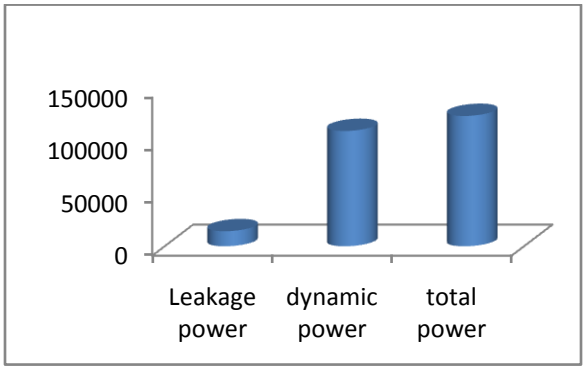

Fig 1.10 Power utilization

Table 1.3 POWER UTILIZATION

Leakage power (nW)	Dynamic power (nW)	Total power (nW)
14270.929	109563.341	123834.270

CONCLUSION

The proposed paper implements the parallel FIR filter by using an efficient multiplexer based Distributed Arithmetic scheme. The device utilization of the proposed architecture is less since the split Look-Up-Table technique is used. This proposed method is implemented for 4-tap to 256-tap parallel FIR filter and can also be extended for more number of taps. This achieves the high speed and less area implementation is achieved in cadence. The test result indicates that the designed filter using proposed DA can work stable with high speed and an area efficient filter. It can save almost 50 percent hardware resource. For digital signal processing applications the order and various parameters can be changed accordingly.

REFERENCES

[1] Chen, K. H., &Chiueh, T. D. (2006). A low-power digit-based reconfigurable FIR filter.IEEE Transactions on Circuits and Systems II: Express Briefs, 53(8), 617-621.

[2] Hentschel, T., &Fettweis, G. (1999). Software radio receivers.In CDMA techniques for third generation mobile systems (pp. 257-283).Springer US.

[3] Mahesh, R., & Vinod, A. P. (2008). A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 27(2), 217-229.

[4] Mahesh, R., & Vinod, A. P. (2010). New reconfigurable architectures for implementing FIR filters with low complexity.IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 29(2), 275-288.

[5] Meher, P. K., Chandrasekaran, S., & Amira, A. (2008). FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic. IEEE Transactions on signal processing, 56(7), 3009-3017.

[6] Meher, P. K. (2006). Hardware-efficient systolization of DA-based calculation of finite digital convolution.IEEE Transactions on Circuits and Systems II: Express Briefs, 53(8), 707-711.

[7] Meher, P. K. (2010). New approach to look-up-table design and memory-based realization of FIR digital filter.IEEE Transactions on Circuits and Systems I: Regular Papers, 57(3), 592-603.

[8] Mirchandani, G., Zinser, R. L., & Evans, J. B. (1992). A new adaptive noise cancellation scheme in the presence of crosstalk (speech signals).IEEE Transactions on circuits and systems II: Analog and Digital Signal Processing, 39(10), 681-694.

[9] Mitola III, J. (2000). Software Radio Architecture: Object-Oriented Approaches to Wireless System Engineering.

[10] Mohanty, B. K., &Meher, P. K. (2016). A high-performance FIR filter architecture for fixed and reconfigurable applications. IEEE transactions on very large scale integration (vlsi) systems, 24(2), 444-452.Mohanty,

[11] B. K., Meher, P. K., Al-Maadeed, S., & Amira, A. (2014). Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters. IEEE Transactions on Circuits and Systems I: Regular Papers, 61(1), 120-133.

[12] Parhi, K. K., &Nishitami, T. (Eds.). (1999). Digital Signal processing for multimedia systems.CRC press.

[13] Park, J., Jeong, W., Mahmoodi-Meimand, H., Wang, Y., Choo, H., & Roy, K. (2004). Computation sharing programmable FIR filter for low-power and high-performance applications.IEEE Journal of solid-state Circuits, 39(2), 348-357.

[14] DA-Based Reconfigurable, F. I. R. (2014). Efficient FPGA and ASIC Realizations of DA-Based Reconfigurable FIR Digital Filter.

[15] Proakis, J. G. (1996). Digital signal processing: principles, algorithms, and application-3/E.

[16] Vinod, A. P., & Lai, E. K. (2006). Low power and high-speed implementation of FIR filters for software defined radio receivers. IEEE Transactions on Wireless Communications, 5(7), 1669-1675.

[17] White, S. A. (1989). Applications of distributed arithmetic to digital signal processing: A tutorial review. IEEE ASSP Magazine, 6(3), 4-19.

[18] Xu, D., & Chiu, J. (1993, April). Design of a high-order FIR digital filtering and variable gain ranging seismic data acquisition system. In Southeastcon'93, Proceedings., IEEE (pp. 6-p). IEEE.

Author Profile

M.Dhivyabharathi is presently pursuing Bachelor's degree in Electronics and Communication Engineering at Tejaa Shakthi institute of Technology, Coimbatore. shehas interested in DSP Design using Cadence Encounter TMSC 90nm and Embedded system.

K.Indhu is presently pursuing Bachelor's degree in Electronics and Communication Engineering at Tejaa Shakthi institute of Technology, Coimbatore. She has interested in DSP Design using Cadence Encounter TMSC 90nm and Embedded system.

A.Kasthuri is presently pursuing Bachelor's degree in Electronics and Communication Engineering at Tejaa Shakthi institute of Technology, Coimbatore. she has interested in DSP Design using Cadence Encounter TMSC 90nm and Embedded system.