# Reconfigurable Low Power FIR Filter based on Partitioned Multipliers

Farhat Abbas Shah
National Engineering and Scientific Commission, Islamabad, Pakistan
Email: fas77pk@yahoo.com

Habibullah Jamal
Vice Chancellor and Professor in Electrical Engineering, University of Engineering and Technology Taxila, Pakistan
Email: drhjamal@uettaxila.edu.pk

Muhammad Akhtar Khan
National Engineering and Scientific Commission, Islamabad, Pakistan
Email: Muhammad.Khan@sli-institute.ac.uk

***Abstract* — This paper presents a low power programmable FIR filter based on partitioned multipliers. Architecture chosen for implementation is conventional direct form. Power efficient techniques like unsigned multiplication and reduction of switching activity are used. Paper presents power, area and speed analysis of the proposed design. FIR Filter is fully parameterized, dynamically programmable and technology independent. Results are presented for 20-tap FIR filter implemented on Xilinx Vertex-II FPGA 2s200fg256-6. Maximum power saving of 48.2% is achieved with an area overhead of 2.08 % only**

*Index Terms*— **Partitioned data dependent multiplier (PDDM), partitioned multiplier (PM), System on chip (SoC), System-on-a-reprogrammable-chip (SoRC).**

## I. Introduction

Present era of mobile computing and multimedia technology demands high-performance and low-power VLSI digital signal processing (DSP) systems. The availability of larger FPGA devices has started a shift of SoC designs towards using Reprogrammable FPGAs, thereby starting a new era of System-on-a-Reprogrammable-Chip (SoRC). Parameterized IP cores remain a standard way to utilize the improvement in FPGA technology and contend with time to market pressure through reuse. [1]

One of the most widely used operations in DSP is finite-impulse response (FIR) filtering which performs the weighted summations of input sequences. There are two main types of FIR Filter implementations namely sequential and parallel [2]. Former is selected for its low complexity and area over head as compared with the later. Sequential implementation requires a single multiplier as compared to multiple adders and multipliers required in parallel implementation. Due to high-speed requirements and increasing complexity of DSP systems, filtering operations at times become computationally intensive and power expensive. This makes low power design an important area of research in the field of digital design.

Most of the previous work has been limited to the design of FIR filters with fixed coefficients [3]. FIR filters with programmable coefficients are used in many applications like adaptive pulse shaping and signal equalization on the received data in real time. Modification of filter coefficients is difficult to be accomplished in real time.

Dynamic power dissipation is the dominant factor contributing over 80% of the total system power [4] and is given by (1).

$$P_{dynamic} = \frac{1}{2}.C_L.V_{dd}^{\ 2}.N.f \qquad (1)$$

Where $C_L$ is the total capacitance, $V_{dd}$ is the supply voltage, N is the weighted mean number of transitions per node per clock cycle and f is the clock frequency. Power can be improved decreasing any one or all of above mentioned parameters. $C_L$ & $V_{dd}$ are constant for a specific FPGA based design while decreasing $f$ also reduces speed. Consequently, dynamic power can be reduced by decreasing N, and N can be decreased by minimizing or switching activity. This is a popular technique for low power algorithmic/architectural level implementation.

In recent years a number of techniques have been proposed for low power implementation of FIR filters. These include the use of differential coefficients, word-length optimization, multi-rate architectures, and dynamic adjustment of filter order, coefficient ordering, and coefficient segmentation [5]. These techniques stress on decreasing switching activity at the input of multiplier and ignoring multiplier itself. MAC unit consumes about 46% of the total power of an FIR Filter while multiplier consumes 66% of the MAC power [6]. Therefore both techniques are used in this work to achieve maximum power saving.

In past lots of work has been done on low power multipliers. [4] These include: serial multipliers, sequential multipliers, array multipliers and tree multipliers. Serial multipliers and sequential multipliers are rarely used because

Manuscript received May 28, 2006.

F. A. Author is with National Engineering & Scientific Commission, Islamabad, PAKISTAN (ph: +92-51-90142667; Mob: +92-304-5180499; fax: +92-51-90142777; e-mail: fas77pk@ yahoo.com).

S. B. Author is with University of Engineering and Technology Taxila, Pakistan (ph: +92-51-9047401; Mob: +92-300-5008337; fax: +92-51-9047420; email: drhjamal@uettaxila.edu.pk).

T. C. Author is with National Engineering and Scientific Commission, Islamabad, Pakistan

of low throughput. Array multipliers and tree multipliers are based on gate level or even switch level efforts suited for VLSI not for FPGAs. Therefore, decision was taken to design new multiplier architecture for power efficient FPGA implementation.

Main objective of the work is to present a low power FIR Filter design using low power multipliers. This however adds small area overhead. Run-time programmability of coefficients is also included in the design which makes the filter versatile. Therefore, power efficient design having less area overhead is aimed.

Switching activity is reduced using sign changing algorithm, unsigned multiplication, and partitioned multiplier architecture. Basic architecture chosen for implementation is direct form and the proposed multipliers are partitioned multipliers (PM). Idea of partitioned multiplier evolved from the fact that larger multipliers can be constructed using smaller multipliers [7]. Low power can be achieved by exploiting the internal architecture of FPGA's and similarity in the design [8]. Partitioned multipliers has been designed and tested for correct functionality and favorable results of power are achieved.

Organization of the paper is as follows: Section-II describes the implementation of reference core and proposed partitioned multipliers. Results of the reference core using these multipliers are presented in Section-III. Finally, the research work is concluded in Section-IV.

## II. IMPLEMENTATION

### *A. Reference FIR Filter* Core *Implementation*

Finite impulse response (FIR) filtering is one of the most widely used operations in Digital Signal Processing (DSP) devices. The basic equation of the FIR filter is given as

$$y_n = \sum_{m=0}^{M-1} b_m x_{n-m} \qquad (2)$$

$b_m$'s are the filter coefficients and $x_{n-m}$'s are the filter input sample values and $y_n$ is the output. Equation (2) can also be written in the form shown below.

$$Y_n = b_0.X_n + b_1X_{n-1} + .... + b_{(M-1)/2}.X_{n-(M-1)/2} + .. \\ ..... + b_{M-1}.X_{n-(M-1)} \qquad (3)$$

Implementation of (3) is called direct form FIR Filter as shown in Fig.1.

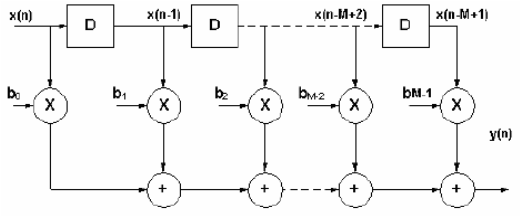


Fig.1. Direct form FIR filter

Specification of the given filter core are: 64 taps, 16-bit data and coefficient width and single MAC implementation. Basic block diagram of the direct form programmable FIR Filter is shown in the Fig.2.

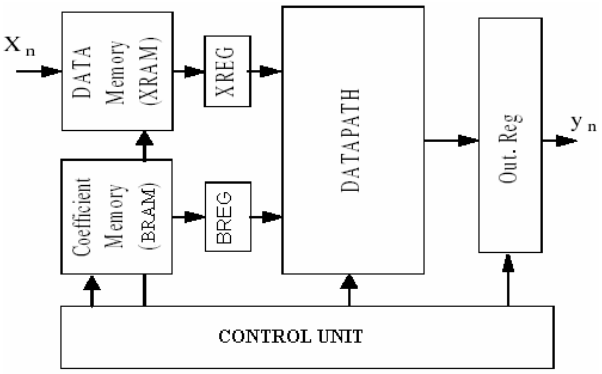


Fig.2.Basic Block diagram of FIR Filter

Data width of all components is 16-bit. XRAM and BRAM both have size 16x64. XRAM stores the input data while coefficients are already stored in the BRAM. Controller is responsible for sequencing and control of each logic function. It generates addresses, read and write signals for both memories. Data path is responsible for performing data manipulations. As far as operation is concerned, the present and N-1 previous data samples of input $x_n$ comes to the data path through XREG & are multiplied by corresponding N-tap coefficients one by one at each clock through BREG. Summation is also done at each clock by adding current and previous results of multiplications to form the filter output $y_n$ after N cycles. Data path consists of a single 16-bit MAC unit and a round off module to get a 16-bit output of the filter output after rounding the 32-bit output of MAC unit.

Direct form FIR filter implementation is run time programmable and uses generic multiplier. Upper limit for the number of taps is 64, data and coefficient width is 16-bit. Power, area and speed results of this core are considered as a reference to study the effect of partitioned multipliers on the performance of Direct Form FIR Filter.

### *B. Low power multipliers*

Most power expensive part of FIR Filter is multiplier and is made power efficient by adopting two techniques:

*First:* using unsigned multiplier: because unsigned multiplication reduces switching activity during sign changing and sign extension as compared to signed multiplication. A sign changing algorithm is used to take care of sign. *Second*: Use of a couple of half data-width low power multipliers to form a single double data width multiplier to further improve the data path power.

The basic principle that a 2n bit (number of bits in the input data; multiplier and multiplicand) multipliers can be constructed using two n bit multipliers [8] is mathematically shown by (4).

$$A.B = (A_H.2^n + A_L) . (B_H.2^n + B_L)$$

$$= A_H.B_H.2^{2n} + (A_H.B_L + A_L.B_H). 2^n + A_L.B_L \qquad (4)$$

$A_H$, $A_L$, $B_H$, $B_L$, A and B are the High significance half number of bits and Low significance half number of bits in the multiplier (A) and multiplicand (B) respectively. This scheme breaks the multiplier and multiplicand bits into two halves and processes them separately. This process of partitioning can be continued to smaller and smaller multipliers. The intermediate products of halved input data is arranged to get faster addition as shown in Fig.3.

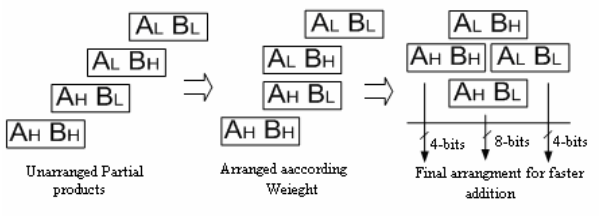


Fig.3 Basic Anatomy of partitioned multiplier

One of the schemes presented here uses four 8x8 constituent multipliers to build single main 16x16 multiplier and is named 2Nx2N_PM partitioned multiplier. Block diagram of such a multiplier is shown in Fig.4.

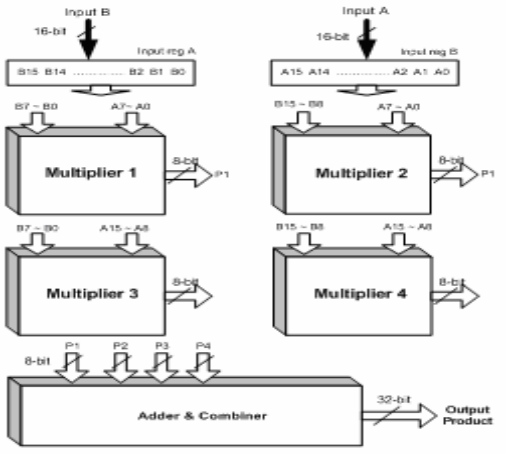


Fig.4 2Nx2N partitioned multiplier

Registered inputs of the main multiplier are applied to the constituent multipliers 1, 2, 3 and 4. The outputs of these constituent multipliers are selectively input to the adder & combiner block. Adder & combiner block rearranges the intermediate products; P's for faster addition. The addition is speeded up due to the reduced number of levels in the addition.

Second multiplier designed and implemented is a 16x16 multiplier using sixteen 4x4 multipliers. Final product is achieved by adder and combiner block. This multiplier is named 4Nx4N_PM partitioned multiplier. Basic anatomy remains same for further partitioning the multiplier; difference is number of constituent multipliers required and the adder & combiner block. As a general rule, to construct an m-level partitioned multiplier of size mNxmN, a total of $2^m$ constituent multipliers of size NxN will be required.

To avoid redundancy, data dependency was introduced in the above multipliers. The idea is to enable selective blocks to avoid zero redundancy depending upon the input data. This architecture is named partitioned data dependent multiplier (PDDM). Partial product selector was added to the partitioned multipliers introducing data dependency, which consist of multiplexers to select the redundancy free P's those are the inputs to adder & combiner block as shown in Fig.5.

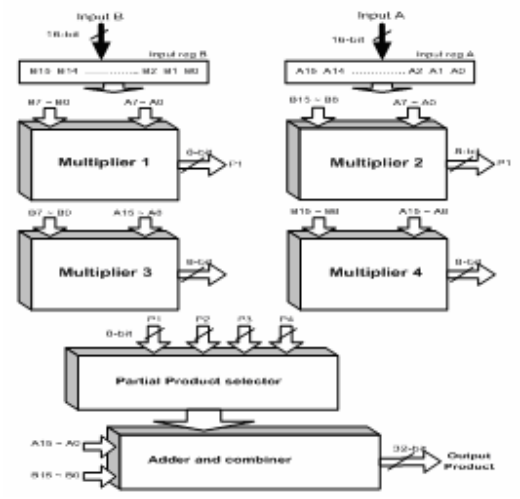


Fig.5 4Nx4N Partitioned Data Dependant multiplier

Third and fourth multipliers implemented are data dependent partitioned multipliers and are named 2Nx2N_PDDM & 4NxN4_PDDM. A booth redix-4 multiplier is also implemented.

## III. Simulation and Results:

Effects of using different kinds of multipliers in Direct Form FIR Filter (df_gen) on area, speed and power triangle were analyzed. Six multipliers are designed and implemented and are all fully parameterized except booth redix-4 multiplier. Partitioned multipliers implemented are: 2Nx2N_PM & 4NX4N_PM and data dependant partitioned multipliers implemented are 2Nx2N_PDDM & 4NX4N_PDDM.

Results were taken for 20-tap, 16 bit direct form FIR filter with generic multiplier, booth multiplier, partitioned multipliers and data dependent partitioned multipliers. Cores were designed in Verilog HDL and implemented using Xilinx ISE 7.1i tool. Simulations were performed using ModelSim SE 5.7g. Xilinx XPower tool was used for power evaluation.

### *A. Power results*

Power results are shown in the table-1. Results show a maximum power improvement of 48.2% with partitioned architecture (df_2Nx2N_PM). This is because single larger multiplier needs more CLBs and complex inter-connect while partitioned architecture exploits internal architecture of FPGA's and because of similarity in the design utilizes most of the available inter-connect. Power efficiency decreases after a certain level of partitioning as in df_4Nx4N_PM. This is because length of interconnects and complexity of the circuit increases with increase in levels of partitioning. Data



Top

dependent version fails to save power because of complexity of the logic, longer select lines and interconnects between P's selector, multipliers & adders. It is observed that df_2Nx2N_PM is the best as far as power is concerned.

*Table-1: Power Analysis*

| Filter cores | Toggle rate | Total power | power improvement |
|---|---|---|---|
| df_gen | 88 % | 27mW | -- |
| df_2Nx2N_PM | 87 % | 14 mW | 48.2 % |
| df_4nx4n_PM | 89 % | 16 mW | 41.8 % |
| df_2nx2n_PMDD | 84 % | 28 mW | -3 % |
| df_4nx4n_PMDD | 86 % | 28 mW | -3 % |
| df_booth | 69 % | 26 mW | 3.8 % |

*B. Area and timing results*

Area and timing analysis are shown in Table-2. Area efficient core is df_2Nx2N_PM with 2% area overhead. Area expense increases with the increase in partitioning level as area expense of 6.5% for df_4Nx4N_PM. Data dependent versions increase area overhead to 20.5 % making this technique inefficient.

*Table-2 Area & timing Analysis*

| Filter cores | df_gen | df_2Nx2N_PM | df_4Nx4N_PM | df_2Nx2N_PDDM | df_4Nx4N_PDDM |
|---|---|---|---|---|---|
| Slices | 31% | 29% | 31% | 30% | 36 % |
| slice Flip-Flop | 16% | 16% | 16% | 16 % | 16 % |
| 4 input LUT's | 22% | 23% | 24% | 25% | 29 % |
| Bonded IOB's | 32% | 32% | 32% | 32% | 32 % |
| No. of GCLKs | 25% | 25% | 25% | 25% | 25 % |
| Gate Count | 13428 | 13707 | 14302 | 14381 | 16186 |
| **Area expense** | **--** | **2.08 %** | **6.5 %** | **7.1 %** | **20. 5%** |
| Rout Delay | 58.1 % | 61.3% | 61 % | 61.4 % | 61 % |
| Logic delay | 41.9 % | 38.7% | 39 % | 38.6% | 39 % |
| **Max. Freq.** | **36.7 MHz.** | **26..4 MHz.** | **22. 3 MHz.** | **26.1 MHz** | **21. 3 MHz.** |

Timing results show a decrease in speed because of the latency increased due to additional hardware and longer interconnects. For high speed requirements pipelining can be introduced in the design. Significant area can be saved by making serial synchronous implementation.

## IV. Conclusion

A number of power efficient multipliers to examine their effect on the performance of a programmable FIR filter are presented. Area, speed and power performance triangle shows favorable results. Results are taken for 20-tap FIR filter using proposed low power multipliers. Device used is Xilinx Vertex-II FPGA 2s200fg256-6. Maximum power saving of 48.2% is achieved with an area overhead of only 2.08 %.

The work can be extended to a pipelined implementation which can improve the throughput of the design. Serial implementation of the partitioned multipliers can save great area. Vendor provided components e.g. multipliers, RAM etc. can be instantiated to get an optimized design for a specific application.

## REFERENCES


[1] Xilinx Design Reuse Methodology for ASIC and FPGA Designers, 2004

[2] C.H.Wang, A.T.Erdoganand T.Arslan, "Algorithmic Implementation of Low-Power High Performance FIR Filtering IP Cores"18th International Conference on VLSI Design(VLSID'05) , 1063-9667/05©2005 IEEE

[3] Jongsun Park, "Computation Sharing Programmable FIR Filter for Low-Power and High-Performance Applications", IEEE Journal of Solid-State Circuits, vol. 39, no. 2, February 2004.

[4] Yijun Liu, Steve Furber, "The Design of a Low Power Asynchronous Multiplier", *ISLPED'04,* August 9–11, 2004, Newport Beach, California, USA.

[5] *A.T. Erdogan and T. Arslan*, "Low Power Block Based FIR Filtering Cores", ISCAS-2003

[6] Muhammad Akhtar Khan, A. T. Erdogan, "Parameterized and Programmable Low Power Soft FIR Filtering IP Cores", WSEAS 2005

[7] Israel Koren, "Computer Arithmetic Algorithms", 2nd Edition, A K Peters, Ltd. 63 South Avenue Natick, MA 01760

[8] Mehra R., Rabaey J., "Exploiting regularity for low-power design", proceedings of the international Conference on computer-aided design, 1996.