International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 02 | May-2015 www.irjet.net p-ISSN: 2395-0072

AN AREA EFFICIENT MCM BASED DIGITAL FIR FILTER FOR SIGNAL PROCESSING SYSTEM

S. Sai Sreelekha [1], C. Vijaya Bhaskar [2]

[1] *PG Student, Department of ECE, Siddharth Institute of Engineering & Technology, A.P., India*
[2] *Assistant Professor, Department of ECE, Siddharth Institute of Engineering & Technology, A.P., India*

Abstract - *This paper proposes the computationally efficient, low power, high speed partial reconfigurable 9-tap FIR filter design using multiple constant multiplication technique. To reduce reconfiguration time dynamic partial reconfiguration is introduced. Module-based dynamic partial reconfiguration is proposed for the realization of both low pass filter and high pass filters. Here the filters are dynamically reconfigured by modifying the coefficients of the filter. The 9-tap fir filter is to design in XILINX ISE. The design is implemented using ALTERA FPGA kit. This dynamic partially reconfigurable FIR filter design shows improvement in reconfiguration time, flexibility and area utilization.*

Key Words: Dynamic Partial Reconfiguration, FIR filter, Distributed Arithmetic, Multiple constant multiplications (MCM), FPGA.

1. INTRODUCTION

Finite Impulse Response (FIR) filters are the most elementary elements in digital signal processing that are typically enforced on dedicated hardware instead of software for speed computation. Due to the need of low power high speed implementation of FIR filter in varied embedded applications, it's necessary to implement the reconfigurable filter architectures supported power or resources issues, or just to implement new practicality on the run.

FPGA is one such platform that permits adapting hardware resources to fulfill time-varying necessities in power, resources, or performance and at identical time maintaining an honest speed of operation. Many of the economical hardware architectures are developed to the exploitation of the reconfigurable and non reconfigurable architectures. one in every of the challenge in partial reconfigurable architectures is that the reconfigurable overhead, that is that the time spent for reconfiguration on the fly as a result of the machine quality and reconfiguration time of the FIR filter will increase with the rise in filter order and kind of arithmetic used.

As multipliers consume additional power in multiply and Accumulate (MAC) operation many multipliers less schemes are projected.

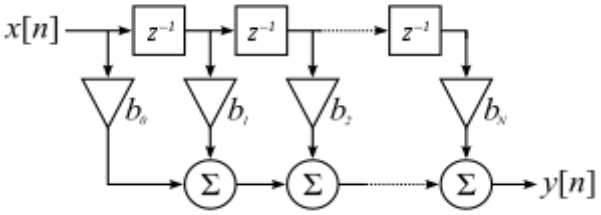

Fig.1: Structure of FIR Filter

MCM is concerned to supply constant multiplication in DSP systems, MIMO (Multiple Input Multiple Output) systems, and Frequency multiplication, Error correcting codes, Graphics and management applications. In such applications full fledge of multipliers aren't required. Since coefficient are constant to supply constant multiplication. Once the MCM design is made, it may be referred to as again and again it needed. Constant multiplication either may be done by digit parallel style or digit serial method. Digit parallel method of constant multiplier wants external wire for shifting. It needs a lot of space whereas implementation takes place in FPGA or the other ASIC. Thus digit serial method overcomes space constrain with acceptable delay temporal order.

Multiplication with constant is termed constant multiplication. This method is employed in filter operation. There are 2 forms of constant multiplication.
One is Single Constant Multiplication and another one is Multiple Constant Multiplication. Input is increased with single specific constant to supply output is termed SCM. Canonical Signed Digit (CSD) variety illustration is employed to implement SCM multipliers. Input is increased with multiple numbers of specific coefficients to supply multiple outputs is termed MCM. Multiplication may be a method of shifting and addition operation. Constant multiplier factor consists of variety of adder, substractor and shifter in keeping with the constant combine.

FIR filter output is obtained by multiplication of input and impulse response. The 2 types of the FIR filter implementations are direct form and transposed form. Instead of direct form, transposed form is best and realizable structure. The Multiplication operation takes place in multiplier block. These transposed structure multiplier block in FIR filter can replace by MCM design additionally referred to as shift and add design. FIR filter provides secure feed forward, stable and linear section response.

For FIR filter, impulse response is adequate the quantity of coefficients, however it's not the case in Infinite Impulse Response (IIR) filters. This paper is organized as follows. Section II describes the fundamentals of distributed arithmetic design for FIR filter design. Section III provides FIR filter by MCM technique with Partial reconfiguration module. Section IV describes the implementation results. Section V presents the conclusion.

2. EXISTING SYSTEM

Distributed Arithmetic technique is bit-serial in nature. It is actually a bit-level rearrangement of the multiply and accumulation operation. The basic DA technique is the computational algorithm that affords the efficient implementation of the weighted sum of the products or dot product.

DA is a bit-serial operation used to compute the inner (dot) product of a constant coefficient vector and a variable input vector in a single direct step and is given by

$$y = \sum_{k=1}^{K} A_k x_k$$

Where y - output response
A_k - constant filter coefficients
X_k - Input data
Let X_k be the N-bits and can be expressed in scaled two's

Complement number as

$$x_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n}$$

By substituting above equation

$$y = \sum_{k=1}^{K} A_k \left[-b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \right]$$

$$y = -\sum_{k=1}^{K} (b_{k0} \bullet A_k) + \sum_{k=1}^{K} \left[\sum_{n=1}^{N-1} (b_{kn} \bullet A_k) 2^{-n} \right]$$

Expanding the inner part,

$$\begin{aligned} y = & -[b_{10} \bullet A_1 + b_{20} \bullet A_2 + \cdots + b_{K0} \bullet A_K] \\ & + [(b_{11} \bullet A_1) 2^{-1} + (b_{12} \bullet A_1) 2^{-2} + \cdots + (b_{1,N-1} \bullet A_1) 2^{-(N-1)}] \\ & + [(b_{21} \bullet A_2) 2^{-1} + (b_{22} \bullet A_2) 2^{-2} + \cdots + (b_{2,N-1} \bullet A_2) 2^{-(N-1)}] \\ & \vdots \\ & + [(b_{K1} \bullet A_K) 2^{-1} + (b_{K2} \bullet A_K) 2^{-2} + \cdots + (b_{K,N-1} \bullet A_K) 2^{-(N-1)}] \end{aligned}$$

Rearranging the summation based on power terms and then grouping the sum of the products,

$$y = -\sum_{k=1}^{K} (b_{k0}) \bullet A_k + \sum_{n=1}^{N-1} [b_{1n} \bullet A_k + b_{2n} \bullet A_2 + \cdots + b_{Kn} \bullet A_K] 2^{-n}$$

The final formulation,

$$y = -\sum_{k=1}^{K} A_k \bullet (b_{k0}) + \sum_{n=1}^{N-1} \left[\sum_{k=1}^{K} A_k \bullet b_{kn} \right] 2^{-n}$$

2.1 Fir Realization Using DA:

The DA of FIR filter consists of Look Up Table (LUT), Shift registers and scaling accumulator.

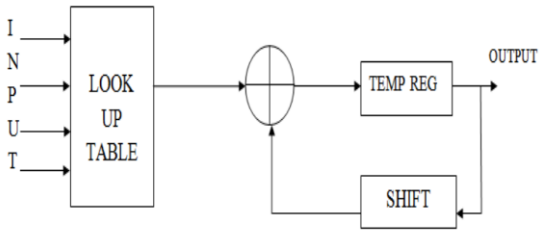

Fig. 2: FIR Filter Using Look up Table

By above, each term inside the brackets indicates a binary AND operation involving a bit of the input variable and all the bits of the constant. The plus signs denote arithmetic addition operations. The exponential factors denote the scaled parts of the bracketed pairs to the total sum. By using this, a look-up table can be constructed that can be addressed by the same type of scaled bit of all the input variables and can access the sum of the terms within each pair of brackets.

From above equation, we can deduce that has only 2^K possible values and it can be precalculated for all possible values of b_{1n} b_{2n} ...b_{Kn}. We can store these in a look-up table of 2^K words addressed by *K-bits*.

2.2 Drawbacks of DA:

1. The LUT in distributed arithmetic uses more memory.

2. Area of reconfigurable partition will be more by using entire LUT.
3. Reconfiguration time is more.

3. PROPOSED SYSTEM

Main objective is to eliminate the multiplier block and introducing the MCM design in digit serial FIR filter for the reducing the requirement of number of multiplications within the variety of shift and add operations

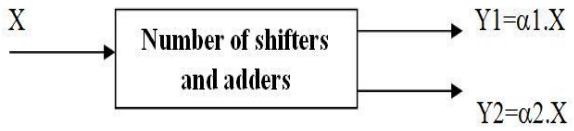

Fig. 3: MCM Operation

Where X denotes input

$\alpha 1$ and $\alpha 2$ are filter coefficients

Y1 and Y2 are the outputs.

3.1 Fir Filter with MCM Architecture:

The constant multiplications are implemented by addition/subtraction and shift operations. Firstly, the constants are represented in binary format. Then, for every 1 within the binary format of constant, based on its bit position, the variable is shifted and adds up the shifted variable to get result.

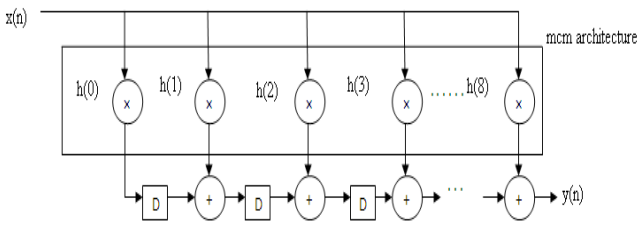

Fig. 4: FIR filters with MCM Architecture.

As an easy example, take into account the constant multiplications 29x and 43x. The decompositions of 29x and 43x in binary listed as follows

29x = (11101) bin x = x<<4 +x<<3 +x<<2 +x

43x = (101011) bin x = x<<5 +x<<3 +x<<1 +x

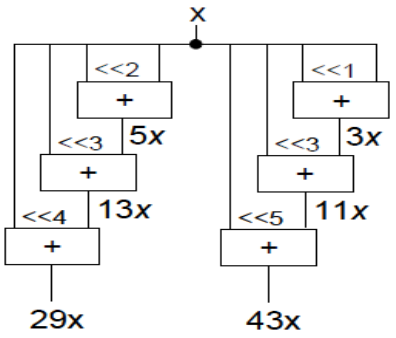

Fig.5: Shift Add Implementation Using MCM

However, implementation of constant multiplications in an exceedingly shift-adds design allows the sharing of common partial product among the constant multiplications that considerably reduces the realm and power dissipation of the MCM technique. Hence, the MCM technique is outlined as finding the minimum range of addition/subtraction operations that implement the constant multiplications, since shifts may be realized by wires in hardware.

3.2 Partial Reconfiguration:

Partial Reconfiguration is ability to reconfigure the chosen areas of FPGA at any time once after initial reconfiguration.

3.2.1 Two groups of PR:

Depending on design functionality, partial reconfiguration is often divided into 2 groups: Dynamic partial reconfiguration and Static partial reconfiguration. In static partial reconfiguration the device isn't active throughout the method of reconfiguration i.e. when the partial data is reconfiguring, on that time remaining action should be in ideal position till the configuration is completed.

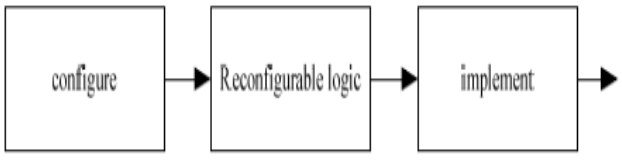

Fig. 6: Static partial reconfiguration

Dynamic partial reconfiguration additionally referred to as active partial reconfiguration permits to vary a section of the device whereas the remaining action of an FPGA is still running.

The Dynamic partial reconfiguration is dispensed to permit the FPGA to adapt to dynamical hardware algorithms, improve fault tolerance and resource utilization, to boost Performance or to scale back power consumption.

DPR is very valuable wherever devices operate in an exceedingly mission crucial setting that can't be discontinuous whereas some subsystems are being redefined. DPR isn't supported on all FPGAs. The Xilinx FPGAs are few devices in market permitting dynamic partial reconfiguration.

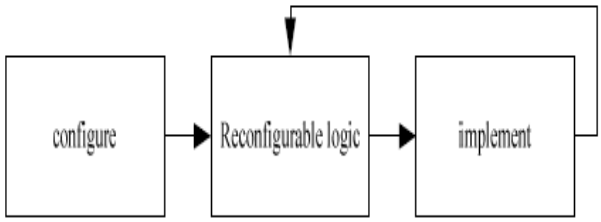

Fig.7: Dynamic partial reconfiguration

3.2.2 Several styles DPR:

The two basic varieties of the dynamic partial reconfiguration on FPGA are: (1) Difference-based partial reconfiguration. (2) Module-based partial reconfiguration. Difference-based partial reconfiguration uses once a little change is needed to the design. It's particularly helpful just in case of fixing Look-Up Table (LUT) equations or dedicated memory blocks content. The partial bit-stream contains information regarding the variations between the present style structure (that resides within the FPGA) and the new content of associate degree FPGA. Shifting the configuration of a module from one implementation to a different is incredibly fast, because the bit stream variations may be extraordinarily smaller than the whole device bit-stream.

Module-based partial reconfiguration uses standard design ideas to reconfigure massive blocks of logic. The distinct parts of the design to be reconfigured called as reconfigurable modules. As a result of specific properties and specific layout criteria should be met with relevancy a reconfigurable module, any FPGA design desiring to use partial reconfiguration should be planned and set out there upon in mind.

3.3. Advantages of MCM technique:

1. Area of reconfigurable partition will be minimized.
2. Reconfigurable time is less.
3. The LUT in MCM uses less memory.
4. The computational time is negligible compared to reconfigurable time.

4. SIMULATION RESULTS

The design and implementation of the partial reconfigurable 9 tap fir filter by an existing method and proposed method are done using the Verilog HDL coding and synthesized on ALTERA FPGA kit.

We followed the proposed reconfiguration scheme where only module containing coefficients is reconfigured. Two reconfigurable modules have been created. In one module we used the coefficients of a low pass filter and in the other module we have implemented a high pass filter.

4.1. Existing System:

The low pass and high pass filter simulation outputs for distributed arithmetic technique shown in fig. 8 and fig. 9.

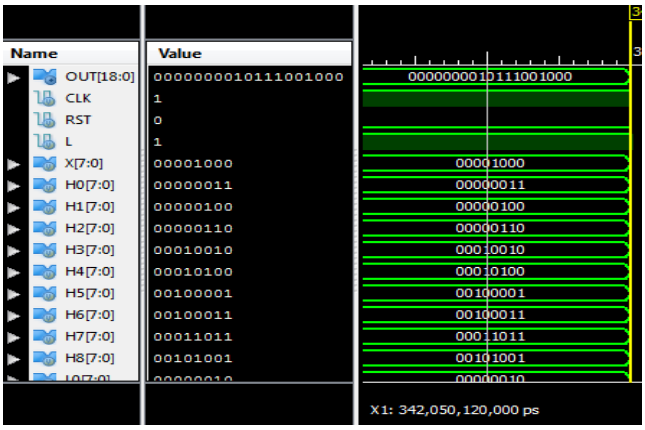

Fig. 8: Simulation Result for Low Pass Filter

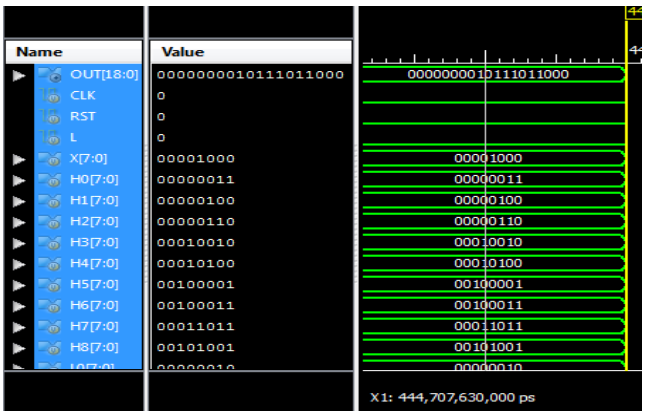

Fig. 9: Simulation Result for High Pass Filter

4.2 Proposed System

The low pass and high pass filter simulation outputs for multiple constant multiplication technique is shown in fig.10 and fig. 11.

The gate utilization area and delay is compared for existing and proposed method is tabulated in Table. 1

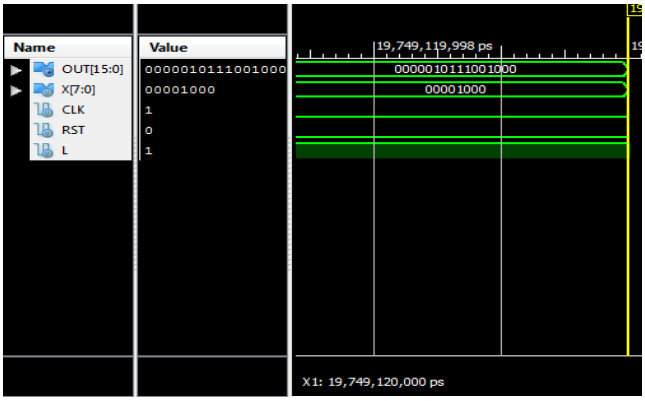

Fig.10: Simulation Result for Low Pass Filter

Fig. 11: Simulation Result for High Pass Filter

Table1. Comparison of Delay and Area

S.No	Name	Delay In (ns)	Device Utilization (LUTs,IOBs) (1408,108)
1.	Distributed Arithmetic Technique	44.644	1224,172
2.	MCM Technique	25.191	361,27

5. CONCLUSION

A partial reconfigurable FIR filter using multiple constant multiplication method is designed. It has been implemented using ALTERA FPGA. The design shows improvement in area utilization, speed and reconfiguration time.

REFERENCES

[1] S. A. White, "Applications of the distributed arithmetic to digital Signal processing: tutorial review," *IEEE ASSP Mag.*, Vol. 6, No. 3,pp. 5–19, Jul. 1989.

[2] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation.* New York: Wiley, 1999.

[3] S.-S. Jeng, H.C. Lin, and S.M. Chang, "FPGA implementation of FIR filter using M-bit parallel distributed arithmetic," in *Proc. IEEE Int.Symp. Circuits Systems (ISCAS)*, May 2006, p. 4.

[4] A. Pealed and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoustic. Speech, Signal Process.* Vol. 22, no. 6, pp. 456–462, Dec. 1974

[5] L. Aksoy, E. Costa, P. Flores, and J. Monteria. Exact and Approximate Algorithms for the Optimization of Area and Delay in Multiple Constant Multiplications. IEEE TCAD, 27(6):1013-1026, 2008.

[6] L. Aksoy, E. Gunes, and P. Flores. Search Algorithms for the Multiple Constant Multiplications Problem: Exact and Approximate. Elsevier Journal on Microprocessors and Microsystems, 34:151-162, 2010.

[7] L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteria. Optimization of Area in Digit-Serial Multiple Constant Multiplications at Gate-Level. In ISCAS, to appear, 2011.

[8] P. Cappello and K. Stieglitz. Some Complexity Issues in Digital Signal Processing. IEEE Tran. on Acoustics, Speech, and Signal Processing, 32(5):1037-1041, 1984..

BIOGRAPHIES

S. Sai Sreelekha pursuing Mtech VLSI Design in Siddharth Institute of Engineering and Technology, Puttur. She received Bachelor Degree in Department of Electronics and Communication Engineering in Siddharth Institute of Engineering and Technology, Puttur.

C. Vijaya Bhaskar working as Assistant professor in Siddharth Institute of Engineering and Technology, Puttur. He received his Bachelor Degree in Engineering from Jawaharlal Nehru Technological University, Hyderabad and Master Degree in Engineering from Jawaharlal Nehru Technological University, Anantapur.

