An Efficient Constant Multiplier Architecture for Reconfigurable Fir Filter Synthesis

Srinath R
Master of Engineering in Applied Electronics
Sri Venkateswara College of Engineering,
Sriperumbudur-602117

Ms. Kousalya R
Assistant Professor, Department of ECE,
Sri Venkateswara College of Engineering,
Sriperumbudur-602117

***Abstract*:- This paper proposes a reconfigurable finite impulse response (FIR) filter using constant multiplier algorithm with applying pipeline technique. To design a high performance Reconfigurable fir filter, according to the proposed constant multiplier algorithm with Retiming pipelining method has been applied in co-efficient generator block. This method capable of reducing the switching activity of constant multiplier process by 28% as compared to that of existing method. In this technique area and delay can be reduced approximately 24% and 62% respectively compare to previous method. This reconfigurable fir filter can be designed using Verilog hardware Description Language (HDL) and its behaviour can be tested using Modelsim 6.3c software and synthesized in Xilinx10.1.**

Keyword: Reconfigurable finite impulse response (FIR),Verilog, pipelining, Xilinx.

I. INTRODUCTION

Fir Filter is important component in any video processing, audio processing, Digital signal processing (DSP) and any bio medical application. Mostly these systems need some process from reconfigurable Fir Filter. This reconfigurable Fir filter commonly work with dynamically configurable filter coefficient and length and factor of interpolation which can change according to the given specification of different input standards in a computing platform.

In any Fir filter, the multiplier architecture is very important because to analysis of performance of any fir filter is mostly depends on that multiplier. In general direct form fir filter, the multiplication function between one input variable (Xin), and more constant variable like h0, h1, h2 coefficients. It is called Multiple Constant Method (MCM). In this paper we have to apply Retiming pipelining cell concept in multiplier block in reconfigurable fir filter.

II. RECONFIGURABLE FIR FILTER USING CONSTANT MULTIPLIER

Reconfigurable fir filter is evaluated from direct form fir filter. In this paper to design the reconfigurable fir filter using constant multiplier algorithm with retiming pipelining cell. Retiming pipelining cell is used to reduce usage of clock period. There are two ways to apply retiming concept in fir filter. First one is retiming pipeline cell placed in after the multiplier block in fir filter. The multiplier block performs multiplication operation between input X and H. Another one is cut set retiming cell placed in between each and every two tap block in fir filter. Here I am using retiming pipeline cell in reconfigurable fir filter. Here the fir filter contains sample data generator, co-efficient data generator, common branch filter, and accumulator [1][2].

A. SAMPLE DATA GENERATOR

The Sample Data generator block contains some parameter like main clock source, reset signal, RRCin signal, interpolation selector (Intp_sel).this data generator block used to generate sample data based on value of interpolation factor, master clock and reset signal. Output of the data generator block is passed to next block.

B. CO-EFFICIENT DATA GENERATOR

The two phase optimization technique is used to reduce the hardware usage of implementing reconfigurable fir filter with minimum time computation and design complexity also will be low. The coded co-efficient generator block is performing the multiplication operation between x and h. The h values are directly applied in look up tables. To find the size of input data is 16 bit and the co-efficient data is 17 bit and provide output data is 16 bit long[1]. The co-efficient data generator block consists of important modules such as sign conversion block, partial product generator, and eight multiplexer units , control logic generator, controlled addition layer two and three and final addition layer. The data flow diagram of co-efficient data generator block is shown in Fig.1.

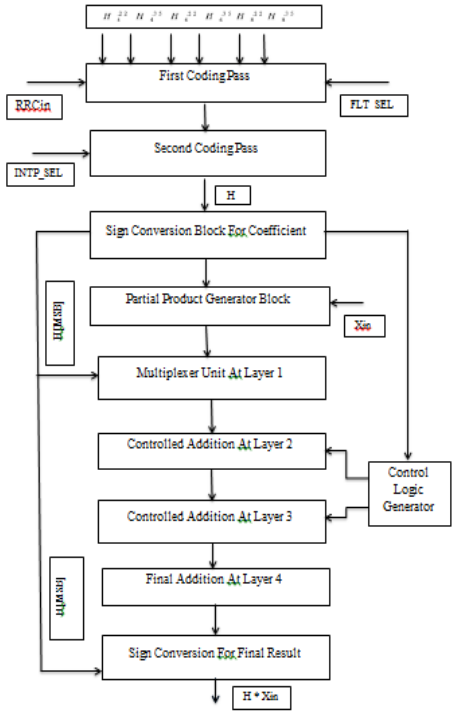

Fig.1. data flow diagram of co-efficient generator block

C. FIRST CODING PASS BLOCK

The First Coding Pass unit is providing initial process of co-efficient generator. The co-efficient value is passed to the first coding pass and it's performing in parallel for three different interpolation factors. The first coding pass block is designed by using 2:1 multiplexer circuit. Here vertically to matching operation between two different coded values in same size. Block diagram of Fir Coding Pass is shown in Fig.3.

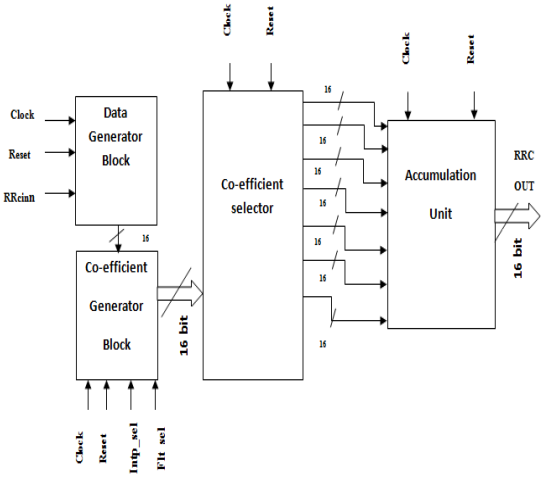

Fig.2. block diagram reconfigurable fir filter

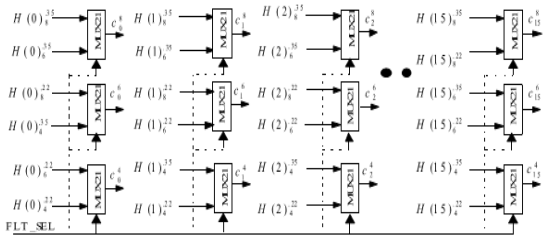

Fig.3. Block diagram of First Coding Pass

D. SECOND CODING PASS BLOCK

The second coding pass unit is get the input data from 3 set of coded value in first coding pass block. The block diagram of second coding pass is shown in Fig.4. In this block perform vertically in between these three set of coded co-efficient value.

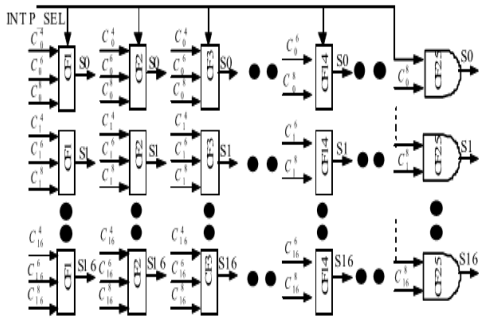

Fig.4. Block diagram of Second Coding Pass

E. SIGN CONVERSION BLOCK

The sign conversion block is used to analysis sign magnitude condition in given coefficient value. The sign conversion block diagram is shown in Fig.5. The sign conversion block gets 17 bit coded co-efficient value from second coding passing block.

The most significant bit of coded co-efficient value is equal to 0 means 16 bit coded coefficient values are passed to the output port else 16 bit inverted coded coefficient values are passed to the output port.

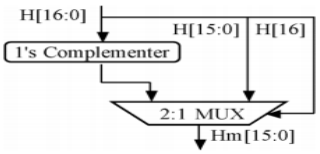

Fig.5.Architecture of the Sign Conversion Block

F. PARTIAL PRODUCT GENERATOR BLOCK

Constant Shift and addition method is used to get the partial product during the multiplication operation between the input data (Xin) and the filter coefficients. In BCSE technique, realizations of the common sub expression using constant shift and add method remove the common term present in a coefficient[1][3]. The block diagram of partial product generator shown in below Fig.6.

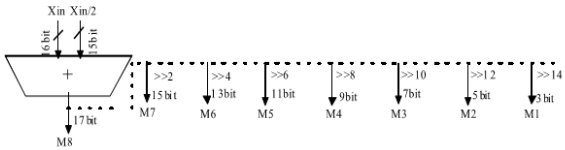

Fig.6. Block diagram of partial product generator

III. EXISTING BCSE BASED RECONFIGURABLE FIR FILTER

The multiplexer unit is used to select the appropriate data generated from the parial product generator block based on the coefficient's bit stream. At layer-1, eight 4:1 multiplexers are required to produce the partial products according to the two bit BCSE algorithm applied vertically on the MAT. The widths of these 8 multiplexers are 17, 15,

13, 11, 9, 7, 5, and 3-bit each instead of 16-bit for all, which would reduce the hardware and power consumption. The block diagram of the multiplexers unit is shown in Fig.7.

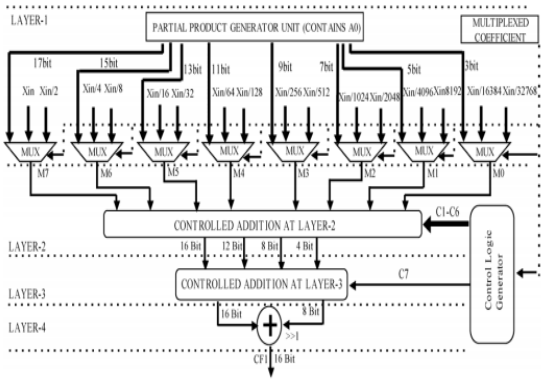

Fig.7. Block diagram of BCSE based reconfigurable fir filter

IV. PROPOSED BCSE BASED RECONFIGURABLE FIR FILTER WITH RETIMING PIPELINE CELL

The proposed method of binary common sub-expression based reconfigurable finite impulse response filter with retiming pipeline cell. The retiming pipeline cell is used reduce critical path delay and also either to increase virtual clock speed, sample time period, or to decrease power consumption at constant speed used on direct form fir filter. Here we have designed 16 bit retiming pipelining cell using data flip-flop[5][6].

Generally retiming pipeline cell method applied in between multiplier and adder block in fir filter. This 16 bit retiming pipelining cell applied in reconfigurable constant multiplier architecture. Finally we have to reduce area and also reduce the propagation delay of reconfigurable fir filter. The block diagram proposed reconfigurable constant multiplier architecture with retiming pipelining cell is shown in Fig.8.

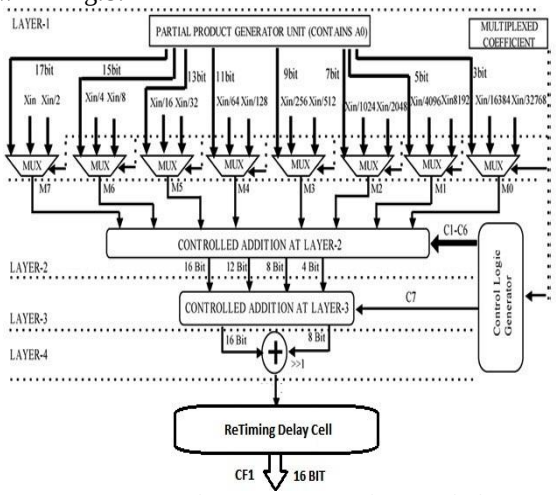

Fig.8. Block diagram of BCSE based reconfigurable fir filter with retiming pipelining cell

G. CONTROL LOGIC GENERATOR (CL) BLOCK

Control logic generator block takes the multiplexed coefficient (H [15:0]) as its input and groups into one of 4-bit each (H [15:12], H [11:8], H [7:4], and H [3:0]) and another of 8-bit each (H [15:8], H [7:0]).

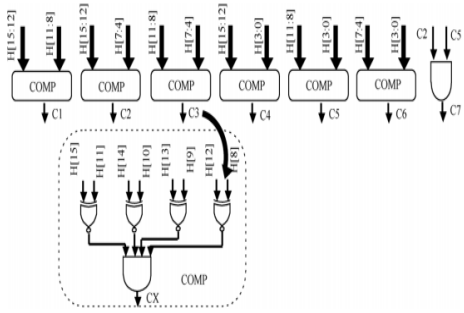

Fig.9. Block diagram of Control Logic (CL) Generator

H. MULTIPLEXERS UNIT

The Multiplexer unit will be used to select the proper data generated from the partial product generator unit which will depend upon the coefficients of the binary value. At layer-1, eight 4:1 multiplexers are needed to produce the partial products according to the 2-bit BCSE algorithm. The width of these 8 multiplexers are 17, 15, 13, 11, 9, 7, 5 and 3-bit each instead of 16 –bit for all partial products, which will reduce the power and hardware consumption[1][2].

I. CONTROLLED ADDITION AT LAYER -2

The architecture of controlled addition at layer-2 is shown in Fig.10. The partial product generator block this generates from eight set of 2-bit coded coefficients are added to the final result of multiplication operation in three different addition layers.

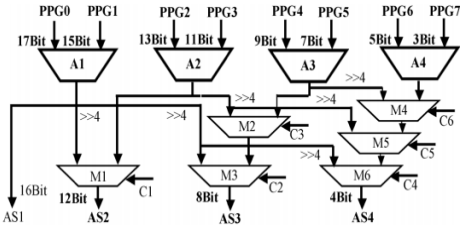

Fig.10. Architectural details of the controlled addition at layer-2 block.

J. CONTROLLED ADDITION AT LAYER-3

The architecture of controlled addition at layer-3 get the input data from output of four sum result in architecture of controlled addition at layer-2[1][4]. The block diagram controlled addition at layer-3 is shown in Fig.11.

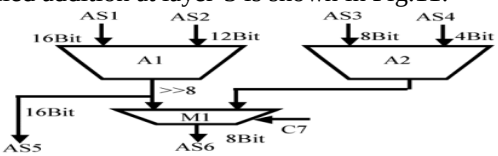

Fig.11. Architecture of the controlled addition at layer-3.

K. CONTROLLED ADDITION AT LAYER-4

The controlled addition at layer-4 which performs 16 bit addition operation between two 16 bit data from addition at layer-3.

L. RETIMING PIPELINING CELL BLOCK

This is final block of reconfigurable constant multiplier architecture. Here we have to design 16 bit retiming pipeline cell register using data flip flop. In this retiming

pipelining method used reduce critical path delay and also reduce the number of register in this design.

M. COMMON BRANCH FILTER

The CS block is used to steer proper data to the final accumulation block depending on the corresponding value of interpolation factor parameter. It takes the input from the CG block.

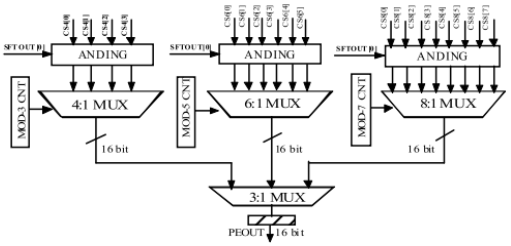

Fig.12. Block diagram of Coefficient Selector

N. ACCUMULATION UNIT BLOCK (FA)

The final accumulation block has a chain of six adders and six registers as there is seven sub filters.

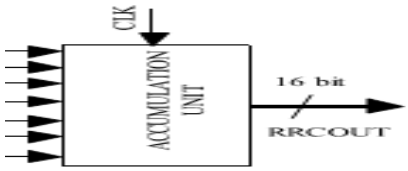

Fig.13. Block Diagram of Accumulation unit

V. SIMULATION RESULT

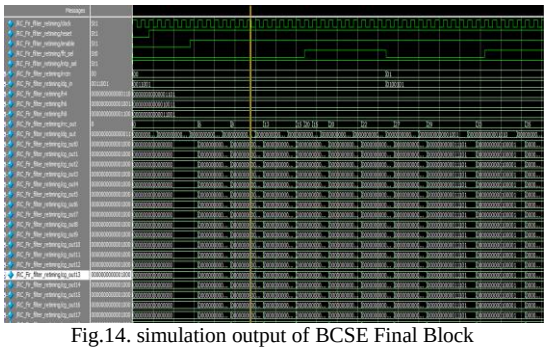

Fig.14. simulation output of BCSE Final Block

VI. TABLE -1
PERFORMANCE COMPARISON BETWEEN TWO DIFFERENT METHOD OF RECONFIGURABLE FIR FILTER

FPGA Spartan3e Xc3s250 PQ208	SLICE	LUT	DELAY
BCSE fir Filter	1367	2605	15.346(ns)
BCSE fir filter using Retiming	1269	2029	2.313(ns)

VII. CONCULSION

Finally we have design reconfigurable fir filter using constant multiplier algorithm with retiming pipelining register through Verilog hardware description language and simulated successfully in Modelsim 6.3c. After that synthesized in Xilinx software, here to analysis usage of hardware computation and propagation delay of fir filter. The proposed method area and delay has been reduced compared to existing method. Both design are synthesized in Xilinx FPGA (field programmable gate array) and verified successfully.

REFERENCES

[1] Hatai; I. Chakrabarti; S. Banerjee, "An Efficient Constant Multiplier Architecture Based on Vertical-HorizontalBinary Common Sub-expression Elimination Algorithm for ReconfigurableFIR Filter Synthesis"IEEE Transactions on Circuits and Systems I: Regular Papers ,Year: 2015, Volume: 62, Issue: 4,Pages: 1071 1080, DOI: 10.1109/TCSI.2015.2388838

[2] S. J. Darak, S. K. P. Gopi, V. A. Prasad, and E. Lai, "Low-complexity reconfigurable fast filter bank for multi-standard wireless receivers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 5, pp.1202–1206, May 2014.

[3] J. L. Nunez-Yanez, T. Spiteri, and G. Vafiadis, "Multi-standard reconfigurable motion estimation processor for hybrid video codecs," IET Comput. Digit. Tech., vol. 5, no. 2, pp. 73–85, Mar. 2011.

[4] H. Samueli, "An improved search algorithm for the design of multiplier less FIR filters with power-of-two coefficients," IEEE Trans. Circuits Syst., vol. 36, no. 7, pp. 1044–1047, Jul. 1989,

[5] G. Dempster and M. D. Macloed, "Use of minimum-adder multiplier blocks in FIR digital filters," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 42, no. 9, pp. 569–577, Sep. 1995.

[6] Y. Yao, H. H. Chen, T. F. Lin, C. J. Chien, and C. T. Hsu, "A novel common subexpression elimination method for synthesizing fixed-point FIR filters," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. , no. 11, pp. 2215–2221, Nov. 2004.

[7] M. Aktan, A. Yurdakul, and G. Dundar, "An algorithm for the design of low-power hardware-efficient FIR filters," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, pp. 1536–1545, Jul. 2008.

[8] C. Y. Yao, W. C. Hsia, and Y. H. Ho, "Designing hardware-efficient fixed-point FIR filters in an expanding subexpression space," IEEE Trans. Circuits and Systems I, Reg. Papers, vol. 61, no. 1, pp. 202–212, Jan. 2014.