isrjournals

International Journal of Advanced Research in Electronics, Communication & Instrumentation Engineering and Development

Volume: 3 Issue: 1 Jul,2016,ISSN_NO: 2347 -7210

Synthesis of Low-Power Area Efficient Constant Multiplier Architecture for Reconfigurable Fir Filter Using Hybrid Form

Nanthini B
PG Scholar, Department of ECE
APEC,
Melmaruvathur, Kancheepuram
nanthuece90@gmail.com

Ganesamoorthy B
Assistant Professor, Department of ECE
APEC,
Melmaruvathur, Kancheepuram
bganesamoorthy@gmail.com

***Abstract*— The design of low power and area efficient high performance DSP system. FIR digital filters are used in DSP by the stability, linear phase for fewer finite precision error and efficient implementation for different applications. Aim of getting reliable operation of these filters are protected using the eliminate common sub expression. Therefore FIR filter is largely dominated to the multiplication of input samples with filter coefficients using different algorithms. The reconfigurable FIR filter design increases in the delay are replicating the hardware. The multiple number of inputs gets processed parallel and same time generating multiple number of outputs and disadvantage of increased area in the design. To overcome this disadvantage for sense of retaining these such advantage of the hardware efficient reconfigurable filter structure is proposed by using VHBCSE algorithm. The least-mean- square (LMS) adaptive filter are deriving its Architectures for high-speed and low-complexity implementation. The direct-form LMS adaptive filter has nearly the same critical path are transpose-form counterpart but provides much faster convergence and lower register complexity. In this paper proposed in the Hybrid form of VHBCSE algorithm in the multiplier circuit. Further power, area and delay are reduced by replacing adder by square root select adder.**

Keywords*— *Reconfigurable FIR filter, VHBCSE algorithm, BCSE, Square root select adder, LMS, Hybrid form.

I. INTRODUCTION

The Digital Filters are plays a role in the analog and digital communication. The main purpose of using the filters eliminate the undesired signal components are providing the better quality signal at the output[1-2]. The digital filters are having the unique characteristics of generating the stabilized signal at the output while compared with analog filters. The digital filters are more preferable than the analog one[3]. There are two main kinds of digital filters are FIR and IIR filter. Moreover Software Defined Radio and multi-standard video codec [2] need a reconfigurable FIR filter are dynamically programmable filter coefficients and lengths which may vary according to different standards in a portable computing platform[4].

Finite Impulse Response (FIR) filters are the most elementary in digital signal processing t are enforced on dedicated hardware instead of software for speed computation. Due to the need for low power and high speed implementation of FIR filter for varied embedded applications, its necessary reconfigurable filter architectures supported power resources issues to implement[5-7].FPGA is one such platform that permits adapting hardware resources are time-varying necessities in power, resources, performance and at identical time maintaining honest speed of operation[8]. Many of the hardware architectures are developed exploitation of the reconfigurable and non-reconfigurable architectures [9].The partial reconfigurable architectures are reconfigurable overhead, that is that the time spent for reconfiguration on the fly machine and reconfiguration time for FIR filter will increase the rise in filter order for kind of arithmetic used.

Significant applicability of an efficient reconfigurable FIR filter motivates system designer to develop the chip is low cost, power, and area along with capability at very high speed. In any FIR filter, multiplier are the major constraint which performance are desired filter[10].Therefore, past three decades are efficient hardware architecture are fixed point FIR filter has been major research focus as reported in published literatures. In FIR filter multiplication operation is variable (the input) and many constants and known as the multiple constant multiplication. The algorithms proposed earlier to implement this MCM for an efficient FIR filter design is categorized by two main groups: 1) graph based algorithms and 2) common sub-expression elimination (CSE) algorithms [11]. Therefore FIR filter implementation employing effective MCM design algorithms are fixed set of coefficients is not suitable like SDR system because two reasons. They are coefficient of the filters in SDR system are dynamically based on requirement of different standards and highly computationally efficient platform needed for those algorithms is SDR system[11].

MCM is constant multiplication in DSP systems, MIMO systems and Frequency multiplication are Graphics and management applications. In such applications full fledge multipliers are required. Since coefficient are constant to supply constant multiplication [12]. The MCM design is made for referred to as again and again it needed. Constant multiplication either may be done by digit parallel and digit serial method. Digit parallel method of constant multiplier wants external wire for shifting. It needs a lot of space and implementation takes place in FPGA or the other ASIC. Thus digit serial method overcomes space constrain with delay temporal order. Multiplication with constant is termed constant multiplication. This method employed in filter operation[13].

The 2 forms of constant multiplication. One is Single Constant Multiplication and another one is Multiple Constant Multiplication. Therefore input is increased with single specific constant to supply output is termed SCM. Canonical Signed Digit is employed to implement SCM multipliers[14].

Input is increased with multiple numbers of specific coefficients for multiple outputs is termed MCM. Multiplication may be shifting and addition operation. Constant multiplier factor consists of variety of adder, subs tractor and shifter in keeping with the constant combine[15].

II. COMMON SUBEXPRESSION ELIMINATION

The CSE technique is to find the terms which are common between different constants and decreasing the number of repeated operations. Therefore some algorithms are in the literature which deal with CSE and in most of them classify three main steps:

- Identify the presence of multiple patterns in the input matrix of coefficients.
- Select one pattern for elimination.
- Eliminate all occurrences of selected pattern.

This process is iteratively repeated for more multiple patterns. The run time and quality solution are important metrics in these algorithms.

Multipliers usually have large area and multiplication is expensive in hardware. In MCM values are constants known before hand. Therefore multiplication implemented by sequence of additions and shifts.

This is because the number of nonzero bits in the CSD representation is fewer than that in corresponding binary representation. In it is shown that the number of nonzero digits is reduced by 50% for CSD representation compared to normal two’s complement form. As the number of nonzero bits in CSD is few, only fewer adders are needed to realize the coefficient multiplier compared to binary representation. However, it should also be noted that, as the number of nonzero bits are minimum, the potential of the CSD-based CSE technique to reduce the number of adders by forming CSs is less than that of binary.

The cost of CSE method mainly depends on three factors: the total number of nonzero bits in the coefficient set, the number of CSs that can be formed from the nonzero bits, and the number of unpaired bits (bits that do not form CSs).The CSE techniques have mainly concentrated on complexity reduction of FIR filters. The goal of CSE is to identify multiple bit patterns are present in the CSD representation of coefficients for eliminate these redundant multiplications.

One major problem in CSE techniques is the selection of representation of filter coefficients. When the multipliers in filters are implemented using shift and add operations, the number of adders (subtractors) is directly proportional to number of nonzero digits present in the filter coefficients. Thus, number systems with fewer numbers of nonzero digits are widely employed for the representation of filter coefficients compared to a conventional binary system. Binary common sub-expression elimination algorithm is the concept of eliminating the common sub-expression for designing an efficient constant multiplier applicable for reconfigurable FIR filters with low complexity. However, the length of the binary common sub-expressions in [13] makes the design of inefficient increasing adder step and hardware cost. The efficiency in terms of speed, power, and area for constant multiplier and it has been increased in the designing for reconfigurable FIR filter.

CSE algorithm is useful solution for achieving less hardware footprint for implementing higher order digital filters. A low-complexity architecture based on binary CSE algorithm has been proposed. This algorithm consumes less hardware for CSD-CSE method using a common constant/programmable shift-and-add block the complexity analyses of fixed bit BCSE is 2-bit and 3-bit BCSE algorithms proposed.

The coefficients in binary pattern, the fixed bit BCSE algorithms eliminate the redundant computation vertically considering 3-bit or 2-bit BCS across the adjacent coefficients. In common sub-expression elimination technique, multiplication operations are the constant coefficients and inputs are performed by shift and add operations. CSE algorithm is a solution for achieving less hardware footprint for implementing higher order digital filters. A reconfigurable constant multiplier and coefficient values can be dynamically programmable. Therefore reconfigurable multiplier is to consider the worst case all relatively better cases will also be taken.

The steps are involved in existing system is given by

- First get the input values X and coefficient values H
- Then to perform shift and add operations
- Coefficient value is 1 means to perform shift operations and added. Otherwise same value is stored in the output.

In existing system method no off adder and multiplier is expanded. So system performance can be reduced and area, power consumption also increased. To overcome this problem we introduced proposed method VHBCSE algorithm.

Partial Reconfiguration is reconfigure the chosen areas of FPGA at any time once after initial reconfiguration. The design functionality are partial reconfiguration is often divided into 2 groups: Dynamic partial reconfiguration and Static partial reconfiguration. Static partial reconfiguration the device isn't active throughout the method of reconfiguration of the partial data is reconfiguring, on that time remaining action should be in ideal position till the configuration is completed. Dynamic partial reconfiguration additionally referred to as active partial reconfiguration device whereas the remaining action of an FPGA is still running.

The Dynamic partial reconfiguration is dispensed to permit FPGA to adapt and improve fault tolerance and resource utilization of boost Performance for power consumption. DPR is very valuable devices operate in an exceedingly mission setting that can't discontinuous some subsystems are being redefined. DPR isn't supported all FPGAs. The Xilinx are few devices in market permitting dynamic partial reconfiguration.

The two basic varieties of dynamic partial reconfiguration are: Difference-based partial reconfiguration and Module-based partial reconfiguration. Therefore difference-based partial reconfiguration uses once a little change is needed in the design and helpful for just in case of fixing Look-Up Table (LUT) equations or dedicated memory blocks content. The partial bit contains information regarding the variations between the present style structure and new content of associate FPGA. Shifting configuration of a module from one implementation is incredibly fast, because bit stream variations are extraordinarily smaller than the whole device bit-stream. The module-based partial reconfiguration uses design ideas to reconfigure massive blocks of logic. As a result of specific properties and layout criteria should be relevancy a reconfigurable module, any FPGA design desiring to partial reconfiguration should be planned and set upon in mind.

III. VHBCSE ALGORITHM

VHBCSE algorithm is the vertical horizontal binary common sub expression elimination algorithm. Vertical and horizontal BCSEs are two types of BCSE which is used for eliminating the BCSs present across the adjacent coefficients and within coefficients respectively any BCSE method. Vertical BCSE produces more effective BCS elimination than the horizontal BCSE. This paper proposes one new BCSE algorithm which is a combination of vertical and horizontal BCSE for designing efficient reconfigurable FIR filter. This proposed algorithm, a 2-bit vertical BCSE has been applied adjacent coefficient is 4-bit and 8-bit horizontal BCSEs to detect and eliminate as many BCSs as possible which are present within each of the coefficient. The coefficients of the FIR filter form a 2-D matrix where each row represents single coefficient and the columns correspond to individual bits of coefficients. Application of 2-bit VCSE to these filter coefficients to generate the partial products requires one adder of 17 full adder cells. Application of 4-bit and 8-bit HCSE are H0 as one input used to designed multiplier and finds no match for the 4-bit and 8-bit BCS within the H0 coefficient.

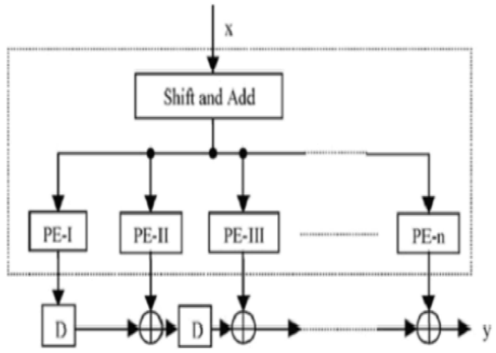


Fig. 1. Transposed direct form of an FIR filter

The blocks are involved in Proposed system are given by

- Sign Conversion
- Control Logic (CL) Generator
- Partial Product Generator
- Multiplexers Unit
- Final Addition

STEPS

The steps are involved in proposed system is given by

- Get the input of 16-bits (x[15:0]).
- Store coefficients of 17-bits (h[16:0]) in LUT.
- Take 1's complement of the 16-bits (except the MSB) coefficient(h'[15:0]).
- If the MSB, the 17th bit (h[16]) of the coefficient is 1, then choose the complemented version of coefficient
- Partition the multiplexed coefficient (MC) (hm[15:0]) into fixed groups of 2 bits each
- Partition the MC into groups of 4 bits each
- Compare the result
- Partition the MC into fixed group of 8-bit
- Compare the result again
- Obtain the final addition result by performing 1-bit right shift on the output
- Take 2's complement of the output
- If the MSB the 17th bit (h[16]) of the coefficient is 1, then choose the complemented version
- Multiplication is completed. Store this result, h*x, in the register.

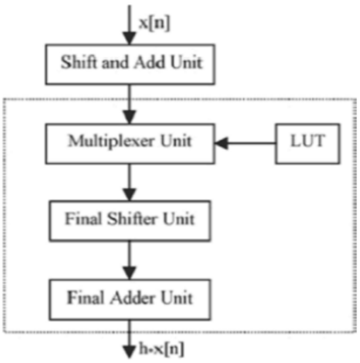


Fig. 2. Architecture of the proposed method

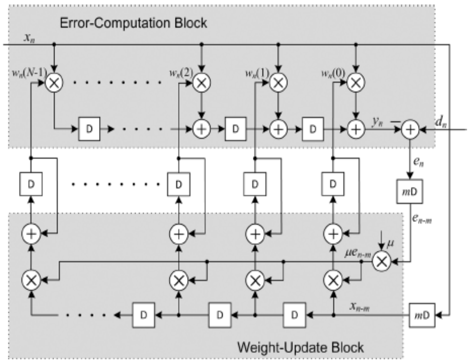


Fig. 3. The structure of the hybrid-form LMS algorithm

The transpose-form LMS is inherently a delayed LMS and consequently provides slower convergence performance. The convergence performance of LMS adaptive filters have simulated the direct-form LMS and transpose-form LMS for the same system. The system is defined by the same simulation configuration. The learning curves thus obtained for filter length and find that the direct from LMS adaptive filter provides faster convergence for transpose LMS adaptive filter in all cases. The direct-form LMS adaptive filter with delay provides faster convergence compared to the transpose-form LMS adaptive filter without delay. The residual mean-square error is found to be nearly the same in all cases. It can be further observed the transpose-form LMS involves significantly higher complexity over the direct-form implementation. Therefore additional signal-path delay line for weight updating, the adder-line to compute filter output are at least twice size for delay line of the direct-form LMS adaptive filter. The error-computation block of the transpose-form LMS adaptive filter. Therefore all multiplications are performed simultaneously, which involves time. Therefore combination of direct form and transpose form is called as hybrid form.

IV. SIMULATION RESULTS

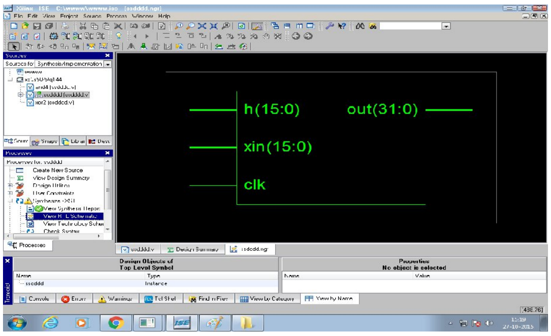


Fig. 4. RTL Schematic Diagram

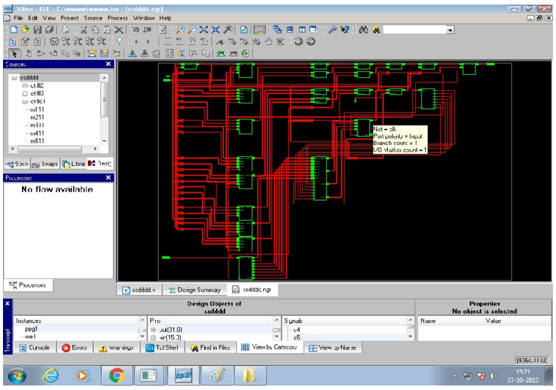
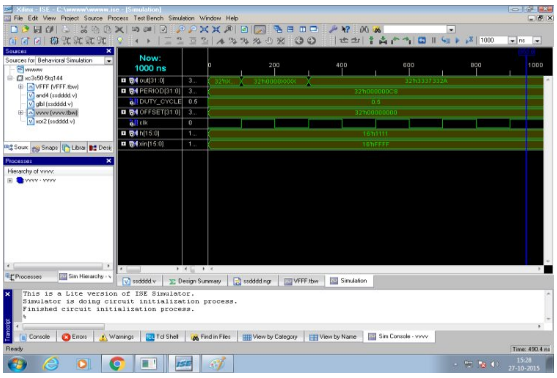


Fig.5. RTL Schematic Diagram



```
This is a Lite version of ISE Simulator.
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Fig. 6 Simulation Result of proposed system

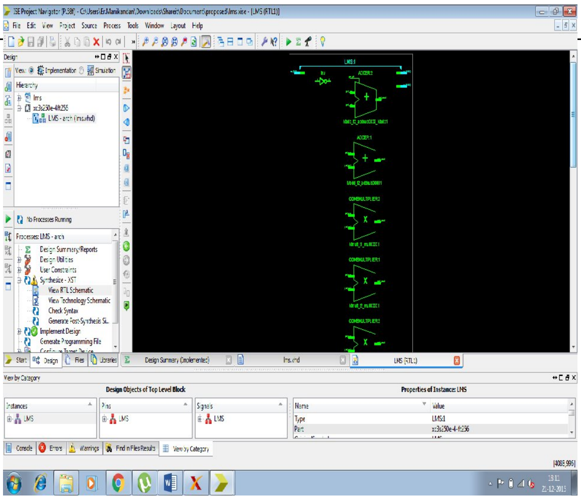


Fig. 7. Hybrid form of VHBCSE RTL Schematic

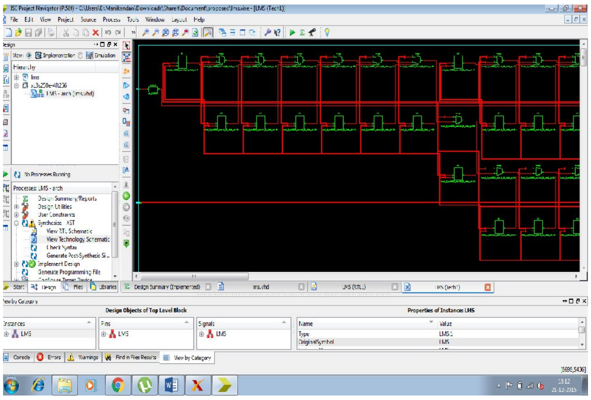
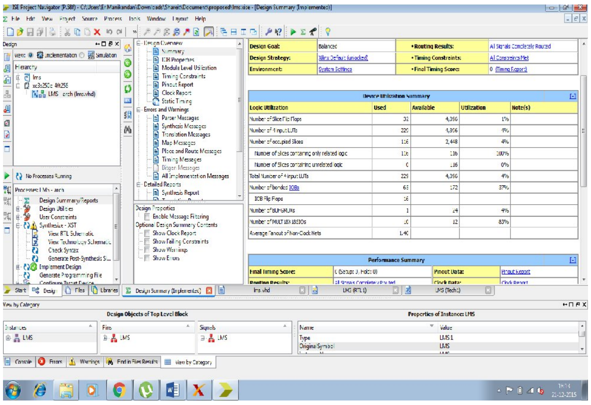


Fig. 8. Hybrid form of VHBCSE Technology Schematic



Design Goal:	Balanced	•Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	All Constraints Met
Environment:	System Settings	•Final Timing Score:	0 (Timing Report)

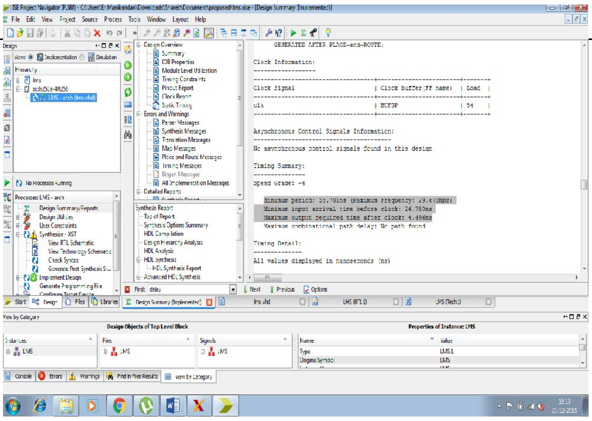
Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	32	4,096	1%	
Number of 4 input LUTs	220	4,096	4%	
Number of occupied Slices	116	2,448	4%	
Number of slices containing only related logic	116	116	100%	
Number of Slices containing unrelated logic	0	116	0%	
Total Number of 4 input LUTs	229	4,096	4%	
Number of bonded IOBs	63	172	37%	
IOB Flip Flops	16			
Number of BUFGMUXs	1	24	4%	
Number of MULT18X18SIOs	10	12	83%	
Average Fanout of Non-Clock Nets	1.40			

Performance Summary

Final Timing Score:	0 (Setup: 0, Hold: 0)	Pinout Data:	Pinout Report
Routing Results:	All Signals Completely Routed	Clock Data:	Clock Report

Fig. 9. Hybrid form of VHBCSE Area



```
     GENERATED AFTER PLACE-and-ROUTE.

Clock Information:
------------------
-----------------------------------+------------------------+-------+
Clock Signal                       | Clock buffer(FF name)  | Load  |
-----------------------------------+------------------------+-------+
clk                                | BUFGP                  | 54    |
-----------------------------------+------------------------+-------+

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 33.701ns (Maximum Frequency: 29.673MHz)
   Minimum input arrival time before clock: 26.780ns
   Maximum output required time after clock: 4.496ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)
```

Fig. 10. Hybrid form of VHBCSE Delay

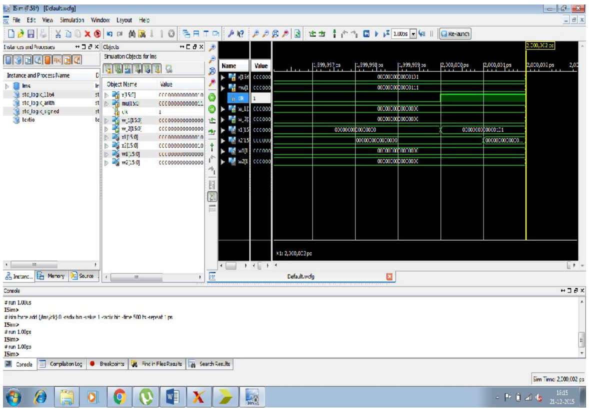
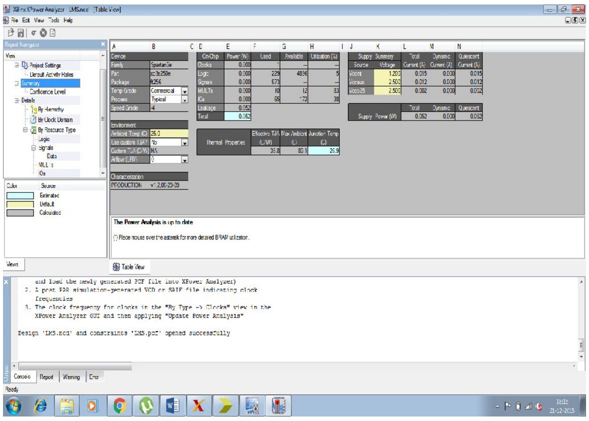


Fig. 11. Hybrid form of VHBCSE Simulation



The Power Analysis is up to date

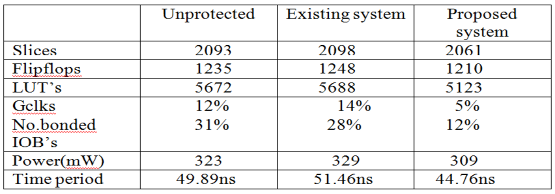
(*) Place mouse over the asterisk for more detailed BRAM utilization.

```
    and load the newly generated PCF file into XPower Analyzer)
 2. A post PAR simulation-generated VCD or SAIF file indicating clock
    frequencies
 3. The clock frequency for clocks in the "By Type -> Clocks" view in the
    XPower Analyzer GUI and then applying "Update Power Analysis"

Design 'LMS.ncd' and constraints 'LMS.pcf' opened successfully
```

Fig. 12. Hybrid form of VHBCSE Power

Table 1. Comparison table for Existing and Proposed



	Unprotected	Existing system	Proposed system
Slices	2093	2098	2061
Flipflops	1235	1248	1210
LUT's	5672	5688	5123
Gclks	12%	14%	5%
No.bonded IOB's	31%	28%	12%
Power(mW)	323	329	309
Time period	49.89ns	51.46ns	44.76ns

V. CONCLUSION

The Filter is designed using efficient adder implementation for getting the reduced power. The modified adder structure generates similar result for the existing module. The power consumption should be reduced from 315mW to 309mW.The utilization can be obtained by analyzing the slices, flip-flops, used gate clk's, and IOB's. The PSM also provides the flexibility of changing the filter coefficient values are dynamically. We have implemented on Spartan-III XC3S200-5PQ-208 FPGA and synthesized. The proposed hybrid form of VHBCSE multiplier reconfigurable architectures gave low Power and Area.

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my Supervisor as well as our principal who gave me the golden opportunity to do this wonderful project on the topic, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

REFERENCES

[1] 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, pp1202–1206, May 2014.

[2] 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.

[3] 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.

[4] A. 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.

[5] C. 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. 51, no. 11, pp. 2215–2221, Nov. 2004

[6] 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.

[7] 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.

[8] B. Rashidi, "High performance and low-power finite impulse response filter based on ring topology with modified retiming serial multiplier on FPGA," IET Signal Process., vol. 7, no. 8, pp. 743–753, Oct. 2013.

[9] H. Choo, K. Muhammad, and K. Roy, "Complexity reduction of digital filter using shift inclusive differential coefficients," IEEE Tans. Signal Process., vol. 52, no. 6, pp. 1760–1772, Jun. 2004.

[10] J. H. Choi, N. Banerjee, and K. Roy, "Variation aware low-power synthesis methodology for fixed point FIR filters," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 28, no. 1, pp. 87–97, Jan. 2009.

[11] P. K. Meher, "New approach to look-up-table design and memorybased realization of FIR digital filter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 592–603, Mar. 2010.

[12] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of DA-based reconfigurable FIR digital filter," IEEE Trans. Circuits Syst. II, Exp. Brief, vol. 61, no. 7, pp. 511–515, Jul. 2014.

[13] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 29, no. 2, pp. 275–288, Feb. 2010.

[14] I. Hatai, I. Chakrabarti, and S. Banerjee, "An efficient VLSI architecture of a reconfigurable pulse-shaping FIR interpolation filter for multi-standard DUC," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., May 2014 [Online]. Available: http://ieeexplore.ieee.org