New Approach to Memory Less Design and Look-Up-Table Realization for Low-Complexity Reconfigurable Digital FIR Filter Architectures

J.L.MAZHER IQBAL
Department of Electronics and Communication Engineering
Rajalakshmi Engineering College,
Chennai-602 105, India
mazheriq@gmail.com

S.VARADARAJAN
Department of Electronics and Communication Engineering
Sri Venkateswara University College of Engineering
Tirupati-517 502, India
varadasouri@gmail.com

Abstract: - Low-complexity and high-speed digital finite impulse response (FIR) filter is widely used in various signal processing and image processing applications because of less area, low cost, low power and high speed of operation. This article presents optimum low-complexity, reconfigurable digital FIR filter architectures based on memory less design and look up table (LUT) realization. The memory less design uses computation sharing multipliers (CSHM) and binary based common sub-expression elimination (BCSE) method for different word length filter coefficients. The memory based LUT multiplier approach uses memory elements to store the sub set of products of the filter coefficients. The LUT based multiplier removes the need of decoders in the FIR filter design. Thus reduce hardware complexity of the proposed reconfigurable digital FIR filter architectures. In this article, we show that the proposed memory based LUT multiplier approach could be an area-efficient alternative to distributed arithmetic (DA) based design of FIR filter with the same throughput of implementation. Also the proposed novel reconfigurable FIR filter architecture using CSHM involves less area and lower latency of implementation compared to the existing reconfigurable FIR filter implementations.

Key-Words: - Memory-based computing, Common sub-expression elimination (CSE), Low-complexity, Reconfigurable architectures, VLSI.

1 Introduction

In many digital signal processing (DSP) and image processing applications finite impulse response (FIR) digital filter is widely used as a basic tool because of their absolute stability and linear phase property. Digital FIR filters find extensively used in mobile communication systems and multimedia applications such as channelization, channel equalization, matched filtering, and pulse shaping, video convolution functions, signal preconditioning and various communication applications. The disadvantage of using digital FIR filters is that it involves a lot of computations to process a signal. The implementation cost and power consumption are also high because of computational complexity. Real-time implementation of large orders filters is a challenging task, as the filter order increases the number of multiply-accumulate (MAC) operations required per filter output also increases therefore the complexity of FIR filter implementation also increases. Steady advances in VLSI technology and design tools have extensively expanded the application domain for DSP over the past decade. While fixed logic application specific integrated circuits (ASICs) and programmable digital signal processors (PDSPs) remain the implementation choice for many DSP applications, increasingly new system implementations based on reconfigurable computing is being considered to develop dedicated and reconfigurable architectures for realization of FIR filters. The core processing element of most reconfigurable computers is field programmable gate array (FPGA). Many characteristics of FPGA such as flexibility, parallelism, reconfigurable time, reduce power consumption, hardware reuse make them attractive in digital FIR filter design. The proposed reconfigurable FIR filter architectures are implemented on an FPGA. The complexity of FIR filter is dictated by the complexity of coefficient multipliers. The multiplication consumes more area, high power, and high latency. The width of the transition band of an FIR filter depends on filter order. The higher the filter order, the sharper is the transition between a pass-band and adjacent stop-band.

Systolic decomposition of distributed arithmetic (DA) based inner product computation (dependent adder graph) exploits high-level of concurrency using pipelining or parallel processing [1]. It is an attractive technique for efficient hardware implementation of DSP applications [2-5]. The systolic decomposition scheme is found to offer a flexible choice of the address length of the look-up-table (LUT) for DA-based computation to decide on suitable area time trade-off. It is observed that by using smaller address lengths for DA based computing unit it is possible to reduce memory size, but leads to increase in adder complexity and latency. In our earlier article [22], we have proposed the reconfigurable FPGA based hardware accelerator, a balance shared memory architecture for FIR filtering and time variant discrete Fourier transform (TVDFT) which provides reconfigurability but not low-complexity. Thus the paper [1, 2-5, 22] does not address the low-complexity and low-latency of multipliers and hence not suitable for higher order filter.

Memories (SRAM) are much faster, high reliable with high packing density, and wide temperature range, hence the computing logic and memory elements are at close proximity to each other to achieve minimum delay, minimum bandwidth requirement, and power dissipation in reconfigurable computing systems [6]. The memory based implementations are computation of multiplication by LUT [7–11] and DA for inner product computation [12–21]. A DA based approach has been suggested in [26], where memory space is reduced at the cost of additional adders. The multiplier less DA based technique computation of inner product by accessing the sequence of LUT followed by shift-accumulation operations. The LUT and shift-add operations, can be efficiently mapped in DA-based computation and well suited for FPGA based realization. The limitation of DA-based implementation for FIR filters is that as the filter order increases the memory requirement increases exponentially. In [31], common sub-expression elimination algorithm for implementing low-complexity FIR filters in software defined radio receivers (SDR) is presented but it does not consider reconfigurability. The memory element has been used as a part or complete arithmetic circuit in various DSP algorithms such as digital FIR filters for a set of fixed coefficients. The advantages of memory based design are reduced latency, dynamic power consumption because of less switching activities for memory-read operations and high-throughput compare to conventional multipliers.

In this article, we have proposed a novel approach to memory based design using optimized LUT multiplier in order to reduce the hardware complexity. The proposed LUT based multiplier removes the need of using decoders in the FIR filter design. Also the complexity of the multiplier in a FIR filter is reduced using the memory less computation. The multiplier block in memory based computation multiplier avoids the computation to obtain the possible product value. All the possible values of odd products of the filter coefficients are pre computed and are stored in LUT. As discussed earlier the memory based design is an area efficient alternative to DA based approach of FIR filter with the same throughput. The product value of the filter coefficient is retrieved using array indexing operation. The memory based approach of storing and retrieving the product value is much faster than performing a computation at runtime.

The proposed memory less reconfigurable FIR filter architecture, (processing element architecture is a modification to what we have proposed in our earlier article titled ‘High Performance Reconfigurable Balanced Shared Memory Architecture for Embedded DSP’ [22]. This architecture is useful for any CSE method with appropriate modifications. The adder shifter unit computes values such as 0x, 2x, 4x, 6x, 8x, 10x, 12x, 14x, where x is the input signal, then reuse these pre-computations efficiently. The multiplier blocks using memory less computation technique avoid redundancy in computation and therefore reduce hardware complexity.

In Section 2, the proposed memory less computation multiplier design for FIR filter is presented. Section 3, presents the low-complexity memory based LUT multiplier. Sections 4, presents FIR filter architecture for the proposed memory less multiplier. In Section 5, we have presented the memory-based structures for FIR filter using LUT multipliers. The area and time-complexity of the computation sharing multiplier (CSHM) and LUT multiplier-based designs of FIR filter are evaluated and compared in Section 6. Conclusions are drawn in Section 7.

2 Proposed Memory less Multiplication

The low-complexity memory less computation multiplier for 4-bit input is shown in Fig.1. The multiplier block of the proposed memory less FIR filter consists of four-to-three line address encoder, adder shifter unit, a shifter, and a control circuit. The control circuit generating the control word (s_1, s_0) for the shifter.

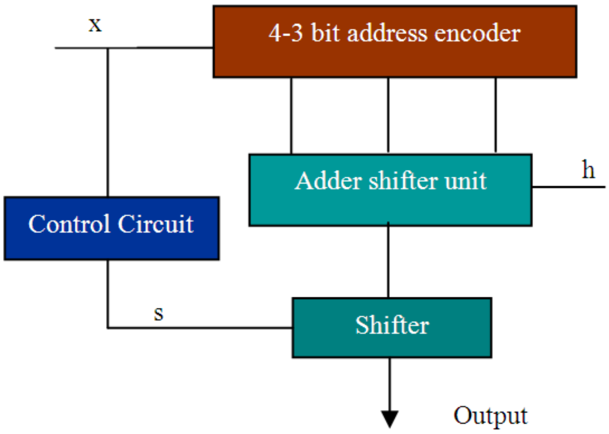

Fig.1 Proposed computation sharing based multiplication for 4-bit input

The function of different blocks of the memory less computation based multiplication is explained below.

2.1 Four to three line address encoder

The four to three bit encoder is a digital circuit that performs logical operation according to equation (1)-(3). It receives a four-bit input word (x_3 x_2 x_1 x_0) and generates three outputs d_0, d_1, and d_2.

$$d_0 = \overline{\overline{(x_0.x_1)}.\overline{(x_1.x_2)}.(x_0 + \overline{(x_2.x_3)})} \qquad (1)$$

$$d_1 = \overline{\overline{(x_0.x_2)}.(x_0 + \overline{(x_1.x_3)})} \qquad (2)$$

$$d_2 = (x_0.x_3) \qquad (3)$$

2.2 Adder and shifter unit

The adder and shifter unit is realized using of shift and add unit, multiplexer and adder. The complexity of multiplier block in FIR filter is reduced, if implemented as shift-adders and sharing common sub-expressions. The shift and add unit uses binary common sub expression elimination (BCSE) method. The shift and add unit is used to realize the 4-bit even BCSs ranging from 0h, 2h, 4h, 6h, 8h, 10h, 12h, 14h. In Fig. 2, “h<<k” represents the filter coefficient h shifted left by k units. The even BCSs such as 0h, 2h, 4h, and 8h do not require any adders for implementation but the BCSs 6h, 10h, 12h, and 14h requires five adders in a straight forward realization of shit and add unit. These even BCSs are expressed as:

$$[0\ 1\ 1\ 0] = h_3 = 2^1h + 2^2h \qquad (4)$$

$$[1\ 0\ 1\ 0] = h_6 = 2^1h + 2^3h \qquad (5)$$

$$[1\ 1\ 0\ 0] = h_8 = 2^2h + 2^3h \qquad (6)$$

$$[1\ 1\ 1\ 0] = h_{10} = 2^1h + 2^2h + 2^3h \qquad (7)$$

Where h is the filter coefficient.

However, h_8 can be obtained from h_3 without using any adders as follows:

$$h_8 = 2^2h + 2^3h = 2^1(2^1h + 2^2h) = 2^1h_3 \qquad (8)$$

Also h_{10} can be obtained from h_8 as follows:

$$h_{10} = 2^1h + 2^2h + 2^3h = 2^1h + h_8 \qquad (9)$$

Thus 4-bit even BCSs 0h, 2h, 4h, 6h, 8h, 10h, 12h, 14h of a 4-bit number are generated using only three adders. All these eight BCSs (0h, 2h, 4h, 6h, 8h, 10h, 12h, and 14h) are then fed to the 8:1 multiplexer. The select signal of the multiplexer are the output of four to three bit encoder d_0, d_1, and d_2. In contrast to conventional shift and add units, the proposed architecture require less number of adders/subtractors. The output of the multiplexer is added with x (n) to find the correct alphabet. The correct alphabet is then fed to the shifter unit.

2.3 Shifter

The inputs to the shifter unit are the output of adder and shifter unit and control circuit. The control circuit generated control-bits. The shifter performs the shift operation on the output of the adder and shifter unit by amount of control bits generated in the control circuit.

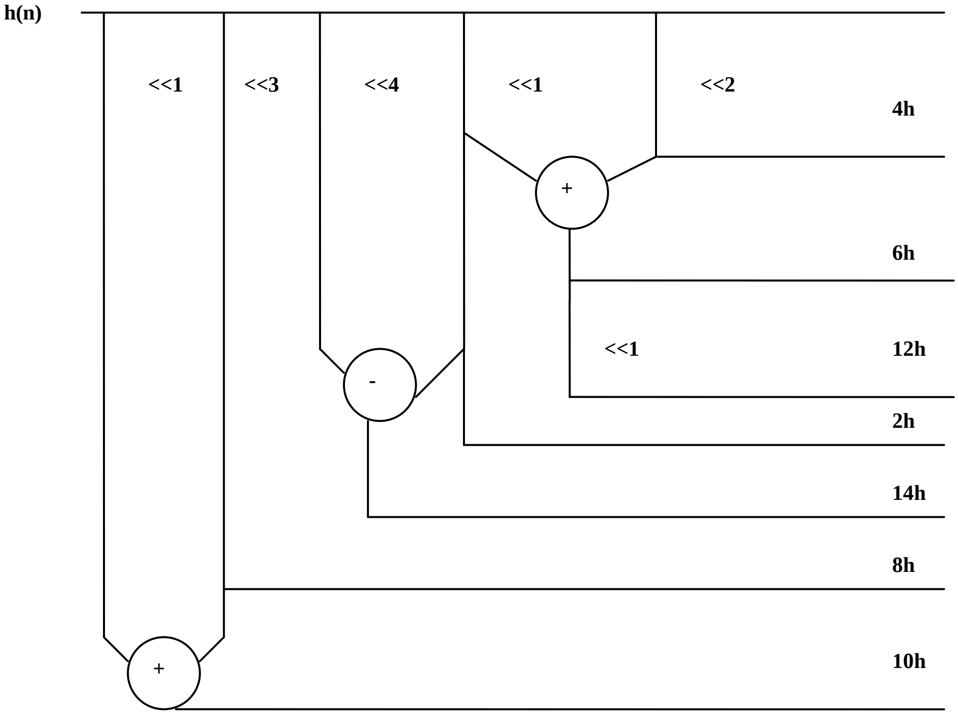

Fig. 2 Shift and add unit for the proposed computation sharing based multiplication

The shifter performs one left shift operation on the output of the adder and shifter unit, if the input x is one of the values {(0 0 1 0), (0 1 1 0), (1 0 1 0), (1 1 1 0)}. The adder and shifter unit output is required to be shifted through two location to left when the input x either (0 1 0 0) or (1 1 0 0). Three left shifts are required if the input x = 1 0 0 0. For all other possible input x, no shifts are required.

2.4 Control Unit

The amount of shifts required performing on the output of the adder and shifter unit is generated by the control circuit. The control circuit generates the control bits using the following expressions given below

$$s_0 = \overline{x_0 + \overline{(x_1 + x_2)}} \quad (10)$$

$$s_1 = \overline{(x_0 + x_1)} \quad (11)$$

Where s_0, s_1 are control bits and x_0, x_1, x_2, x_3 are four-bit input word.

3 Memory based Multiplication using LUT approach

Fig. 3 show the LUT based multiplier for 4-bit input. The low-complexity memory based multiplication using LUT FIR filter consists of a four-to-three line address encoder, memory array of eight words of (W+4) bit width, a shifter, and a control circuit for generating the control word (s_1, s_0) for the shifter. This section addresses the problem of efficiently computing a set of products A_N* x, for N = 1. . . n, for a variable x with known set of fixed coefficients A_N using memory based array. The four to three line address encoder, shifter and the control circuit is same as discussed earlier in section 2.1, 2.3 and 2.4. Let x is an input word and A is an unsigned binary fixed coefficient then there can be 2L possible values of product A*x, where L is the width of input x. The odd multiples of pre computed product value A*x is stored in the memory unit (LUT). Table 1 show the possible product values for 4 bit input word length, amount of shifts generated by the control circuit, odd multiples of product value to be stored in the memory, and 3-bit address generated by the encoder. The binary value of x is encoded using four to three bit encoder. The encoded value is used as select line to access the referenced word from the memory array. If the output of the encoder is ‘000’, A is retrieved from the memory and 2A, 4A and 8A are obtained by performing amount of left shift operation generated by the control circuitry. If the output of the encoder is ‘001’, 3A is retrieved from LUT and 6A, and 12A are derived by left shift operation on 3A. Similarly if the encoder output is ‘010’ and ‘011’, 5A and 7A are retrieved from the memory and 10A, 14A are derived by left shift operation on 5A and 7A respectively. 9A, 11A, 13A, and 15A are already stored in the memory. Thus all the possible product value A, 2A, 3A … 15A are obtained from odd multiples of product value stored in memory unit. Fig. 4 show the LUT based multiplier for 8-bit input and Fig. 5 show the low-complexity memory less computation multiplier for 4-bit input.

The proposed memory based FIR filter using LUT multiplier is analysed for the following set of fixed coefficient {3, 13, 21, and 37}, {3, 13, 219, and 221}. The odd multiples of product value to be stored in the memory for a set of fixed coefficient {3, 13, 21, and 37} are {0000000011, 0000001001, 0000001111, 0000010101, 0000011011, 0000100001, 0000100111, and 0000101101}, {0000001101, 0000100111, 0010000001, 0001011011, 0001110101, 0010001111, 0010101001, and 0011000011}, {0000010101, 0000111111, 0001101001, 0010010011, 0010111101, 0011100111, 0100010001, and 0100111011}, {0000100101, 0001101111, 0010111001, 0100000011, 0101001101, 0110010111, 0101010101, 1000101011}.

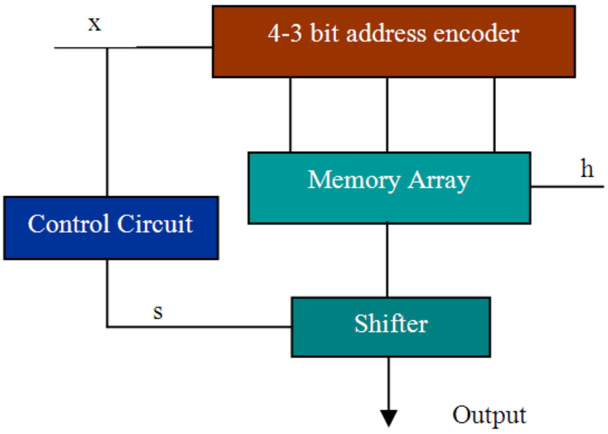

Fig. 3 Proposed memory based multiplication for a set of fixed coefficients of 4-bit input

4 FIR Filter architecture for the Proposed Memory less Multiplier

A discrete time filter produces a discrete time output sequence y(n) for the discrete time input x(n). An FIR filter of length M is described by the difference equation:

$$y(n) = h_0x(n) + h_1x(n-1) + h_2x(n-2) + h_3x(n-3) + ... + h_{M-1}x(n-M+1) \quad (12)$$

Where h(n) is the set of filter coefficient while x(n-i) , for i= 0,1,2….N-1 , represent N recent input samples, and y(n) represents the current output sample.

The proposed FIR filter architecture is a reconfigurable processing element architecture (FPGA based hardware accelerator) shown in Fig. 6. Each processing element performs memory less mapping of the input values to a single output values. The requirements on the PE are that it completes its operation within the specified time limit. The proposed FIR filter architecture can obtain a high area efficiency and high performance. The filter architecture consists of two PEs (PE_1 & PE_2), memory element, and interconnection network. The PE performs multiplication using the proposed low-complexity memory less computation multiplier and this is analysed for 4-bit and 8-bit input and is shown in Fig. 1 and Fig. 5. Control signals s_1 and s_2 enable multiplication and addition respectively. If s_1=0 and s_2=1, the PE_1 behaves as accumulation block and PE_2 behaves as multiplier block [32]. The low-complexity PE_2 for 8-bit input consists of two four-to-three line address encoder, one adder shifter unit, two shifters, and two control circuits for generating the control word (s_1, s_0) for the shifter. The 8-bit input x is partitioned into two 4-bit sub words and fed into two pair of four to three bit encoder and control circuits. The four to three bit encoder generates two set of select signals {d_{00}, d_{01}, and d_{02}} and {d_{10}, d_{11}, and d_{12}} to the adder and shifter unit. Control circuit generates two set of control signals {S_{00}, S_{01}}, and {S_{10}, S_{11}} and is fed to the shifter unit. The outputs of both the shifter are concatenated to get the multiplied output of PE_2.

5 Memory-based structures for FIR Filter using LUT Multiplier

Fig. 7 show the memory based structure of FIR filter using LUT multiplier for 8-bit input word length. The proposed structure of FIR filter consist of two four to three bit encoders, a single memory-module, control circuits, an array of shift-add (SA) cells, AS cells and a delay register. Two four to three bit encoders are used to generate word select signals from the input word x for the dual-port memory core and control circuits to generate the necessary control signals for the dual-port memory core. If all the multiplications are implemented by a single memory module, the hardware complexity of the decoder circuits is eliminated. The 8-bit input x is partitioned into two 4-bits sub words and fed into two pair of four to three bit encoder and control circuits. The four to three bit encoders provide two set of word select signals {d_{00}, d_{01}, and d_{02}} and {d_{10}, d_{11}, and d_{12}}. Control circuit provide two set of control signals {S_{00}, S_{01}}, and {S_{10}, S_{11}}. The dual port memory module receives all these four signals set as shown in Fig.7.

The single memory module is a dual port memory consists of eight rows of [(W + 4) N]-bit width. Each segment in the row of memory is (W + 4) bit wide and the size of the segment is 8 × (W + 4). The i^{th} segment in the memory stores the product value of x*h. The output of the dual port memory are fed to the 2N NOR cells and finally to the pair of barrel shifters. When the filter coefficient is positive or negative, the add/sub cell adds or subtracts its input from top with or from that of its input from the left.

Memory is used to perform DSP functions. It enhances overall performance of computing systems minimizing the bandwidth requirement, access-delay and power dissipation. The computing logic and memory elements are at closest proximity to each other. In addition to that, memory elements have also been used either as a complete arithmetic circuit or a part of that in various application specific platforms. Memory based design have many advantage over MAC structures such as greater potential for high-throughput, reduced latency and less dynamic power consumption because of less switching activities for memory-read operations compared to the conventional multipliers.

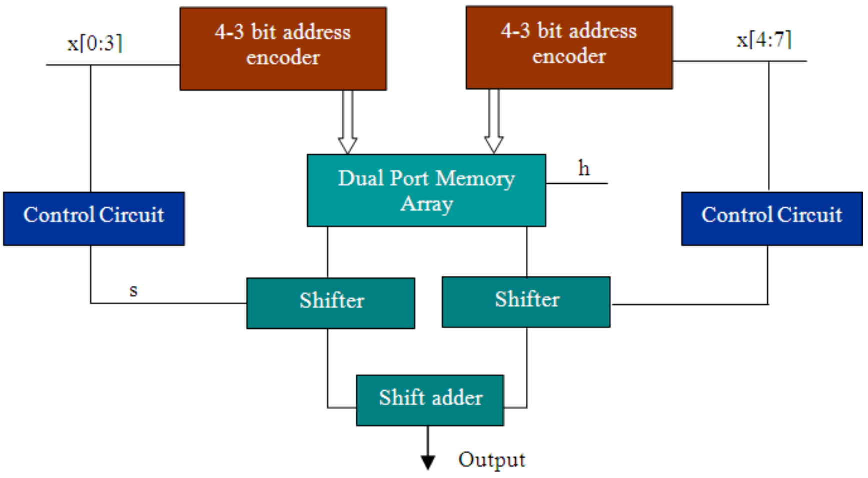

Fig. 4 Proposed memory based multiplication for a set of fixed coefficients for 8 bit input

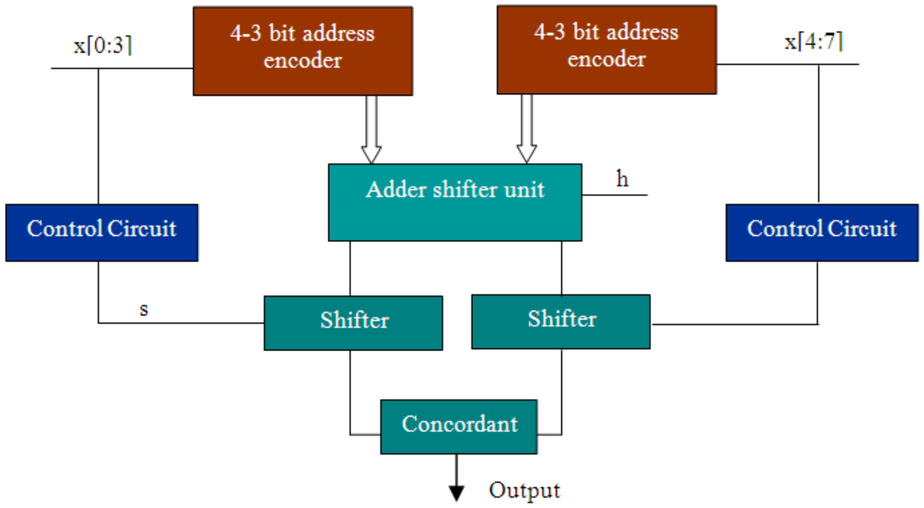

Fig. 5 Proposed memory less multiplication processing element for 8 bit input

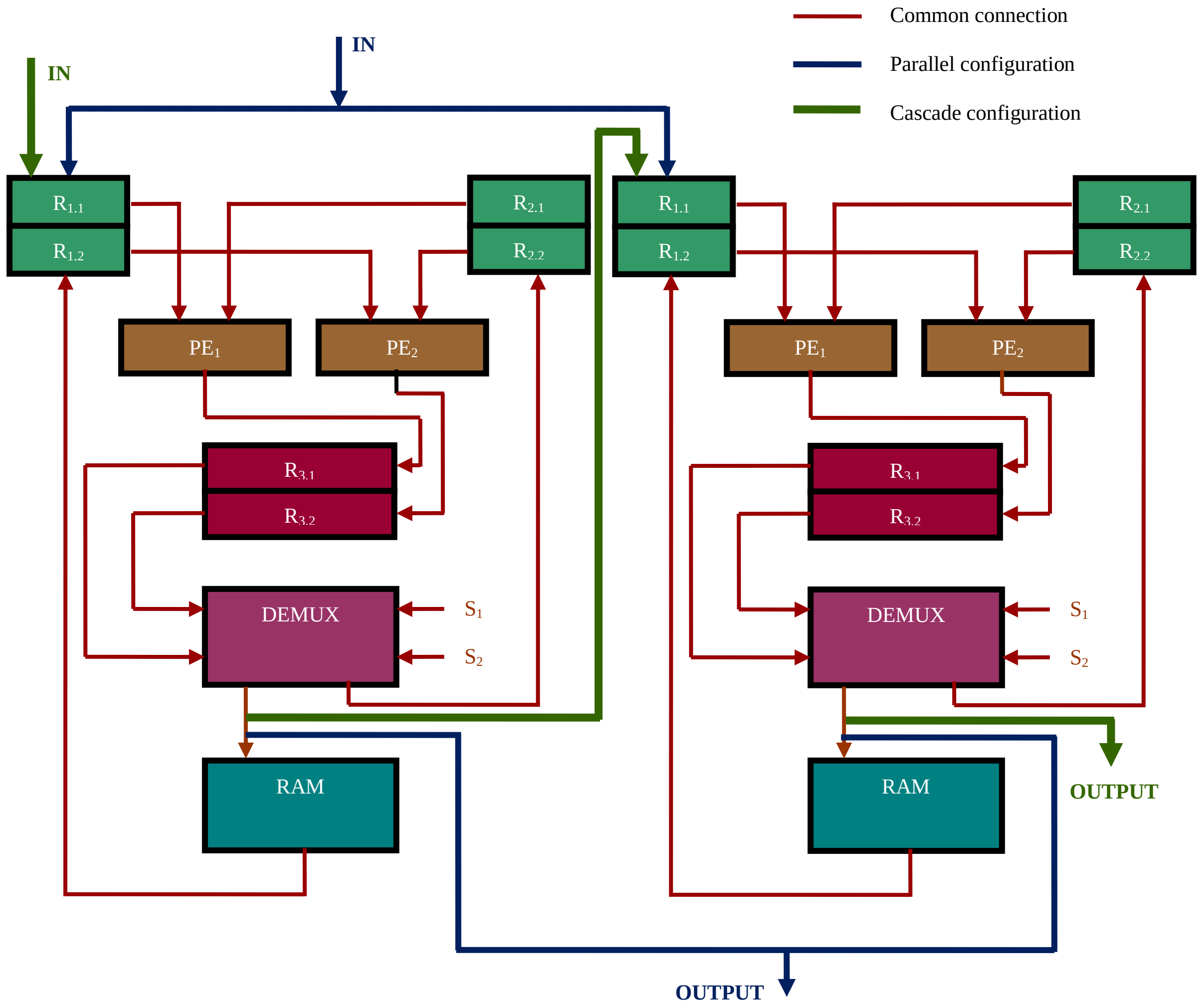

Fig. 6 FIR filter architecture for the proposed computation sharing multiplier [22]

Table 1 LUT words and product values for input word length L = 4

Input $x_3x_2x_1x_0$	Product value	Number of shifts	Control s_0 s_1	Symbol	Stored Value	Address $d_2d_1d_0$
0 0 0 1	A	0	0 0	P0	A	0 0 0
0 0 1 0	2^1*A	1	0 1			
0 1 0 0	2^2*A	2	1 0			
1 0 0 0	2^3*A	3	1 1			
0 0 1 1	3A	0	0 0	P1	3A	0 0 1
0 1 1 0	2^1*3A	1	0 1			
1 1 0 0	2^2*3A	2	1 0			
0 1 0 1	5A	0	0 0	P2	5A	0 1 0
1 0 1 0	2^1*5A	1	0 1			
0 1 1 1	7A	0	0 0			
1 1 1 0	2^1*7A	1	0 1	P3	7A	0 1 1
1 0 0 1	9A	0	0 0	P4	9A	1 0 0
1 0 1 1	11A	0	0 0	P5	11A	1 0 1
1 1 0 1	13A	0	0 0	P6	13A	1 1 0
1 1 1 1	15A	0	0 0	P7	15A	1 1 1

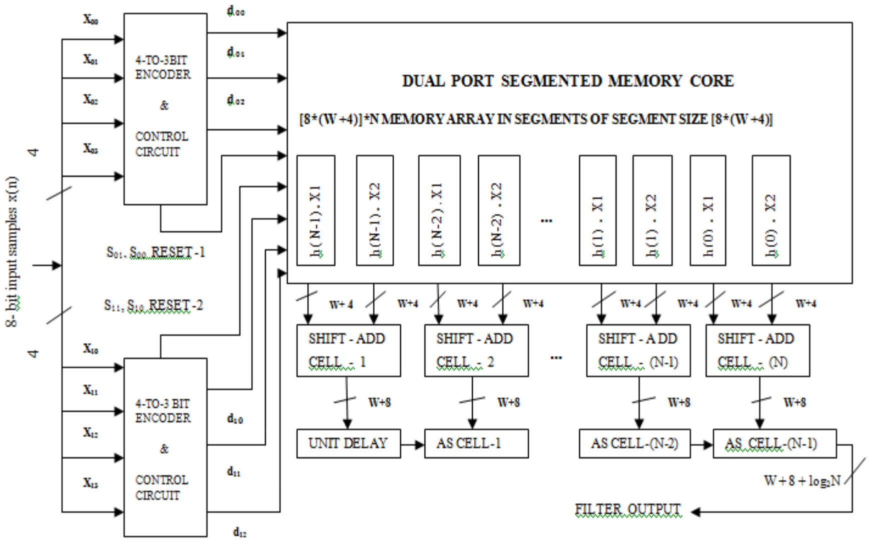

Fig. 7 Structure of Nth order FIR filter using proposed LUT-multiplier

6 Experimental Results

In this section, the synthesis and design results of proposed FIR filter architecture for a set of fixed coefficients using the memory based LUT multiplier and FIR filter architecture using CSHM are presented. We have used Xilinx 9.1i ISE for synthesizing purposes. The synthesis has been done on Xilinx's Virtex-II family 2vp2fg256-6. Table 2 show the synthesis results of the proposed reconfigurable MB CSM architecture and LUT based multiplier block. The simulation result of the proposed MB CSM architecture for programmable coefficient and LUT based multiplier block for a set of fixed coefficient are shown in Fig. 9.

Table 2 Performance for the proposed reconfigurable FIR filter architectures

Device Utilization(8 bit)	CSHM MB	Memory based LUT MB
Number of Slices	29	20
Number of Slice Flip Flops	8	17
Number of 4 input LUTs	55	30
Number of bonded IOBs	18	20
Number of GCLKs	1	1

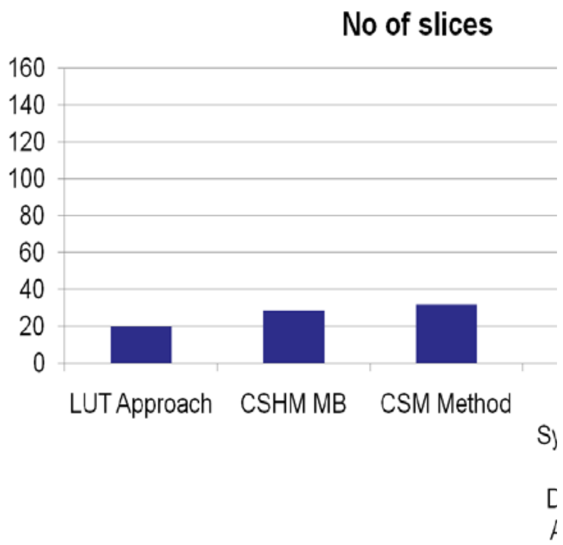

Fig. 8 Comparison of performance for the proposed memory based LUT approach, computation sharing multiplier and the existing reconfigurable implementation

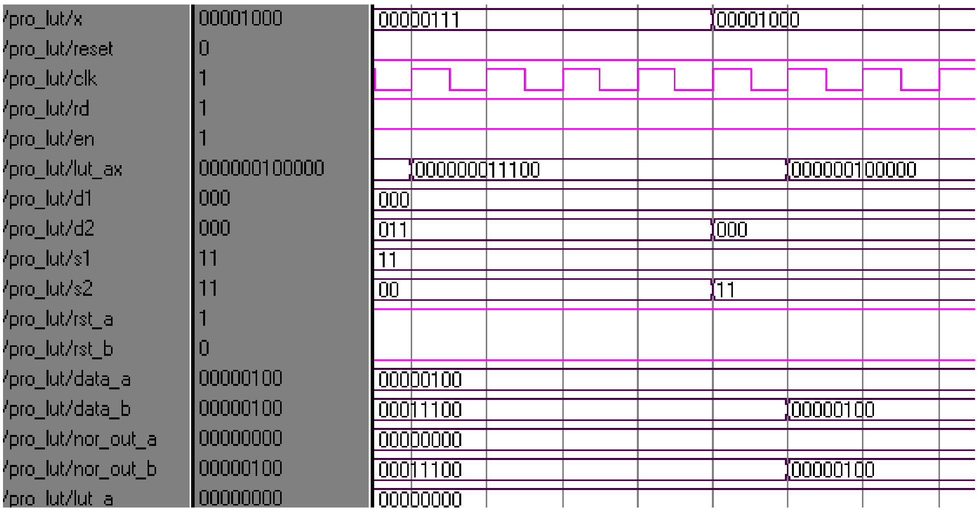
```
/pro_lut/x          00001000
/pro_lut/reset      0
/pro_lut/clk        1
/pro_lut/rd         1
/pro_lut/en         1
/pro_lut/lut_ax     000001100000
/pro_lut/d1         000
/pro_lut/d2         000
/pro_lut/s1         11
/pro_lut/s2         11
/pro_lut/rst_a      1
/pro_lut/rst_b      0
/pro_lut/data_a     00000100
/pro_lut/data_b     00000100
/pro_lut/nor_out_a  00000000
/pro_lut/nor_out_b  00000100
/pro_lut/lut_a      00000000
```

Fig. 9 Simulation Result for the memory based FIR filter using LUT multiplier and CSHM

7 Discussions

The proposed FIR filter architecture using the memory based LUT multiplier block for the set of fixed coefficient and CSHM block is compared with the existing FIR filter architectures [22-24, 19, 25-18] in terms of the basic design metrics. The proposed low-complexity memory based LUT multiplier design for a set of fixed coefficient and memory less computation multiplier has been synthesis and observed that it consumes less area than the existing techniques. Fig.8 shows the results of performance of the proposed memory based LUT approach and CSHM with existing reconfigurable implementations. The paper entitled “FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic” (Pramod Kumar Meher et al 2008), appearing in the proceedings of IEEE Trans, Signal Processing, Systolic decomposition of distributed arithmetic (DA) based inner product computation (dependent adder graph) have been used. In the aforesaid work, address length is reduced to reduce the memory size but it also leads to increase of adder complexity and latency. The memory requirement of DA based implementation for FIR filters, however, increases exponentially with the filter order. An LUT less adder based DA approach has been suggested (Yoo and Anderson 2005), where memory space is reduced at the cost of additional adders. Another paper (Jae-Jin Lee et al 2004) presents a bit level super systolic FIR filter with an FPGA based bit-serial semi-systolic multiplier which twists on the shift and add multiplier by positioning upper and lower half of the serial multiplier side by side physically in floor planning, instead of linearly. The super systolic FIR filter with an FPGA based bit serial semi systolic multiplier requires more area compared to the proposed architecture. The paper entitled "Computation sharing programmable FIR filter for low-power and high-performance applications" (J. Park et al 2004), appearing in the proceedings of IEEE J. Solid State Circuits, deals with programmable digital FIR filter using CSHM efficiently. The architecture has pre-computers using nine adders employing two programmable shifters. These programmable shifters reduce the overall speed of operation of the resulting filters especially for higher order channel filter applications in wireless communication receivers. The proposed reconfigurable FIR filter use CSHM and BCSE method to reduce the area required for implementation and it exploits pipelining and parallel processing to increase the speed of operation. The proposed LUT based multiplier removes the need of decoders in the FIR filter design. Thus reduce hardware complexity of the proposed architecture.

7 Conclusions

In this paper we have proposed two new architectures for FIR digital filter synthesis for a set of fixed coefficients and programmable coefficients for digital signal processing, image processing, mobile communication systems and multimedia applications. The two novel architectures are memory based FIR filter architecture using low-complexity LUT multiplier for a set of fixed coefficients and FIR filter based on low-complexity CSHM for programmable coefficients. In the first method, a low-complexity memory based FIR filter using optimized LUT multiplier is suggested. In the proposed LUT based multiplier the complexity is reduced by eliminating the need of decoder compare to DA based design. The memory based FIR filter could be more efficient in terms of area and latency. The second architecture is a reconfigurable FIR filter architecture using the proposed low-complexity CSHM. The redundant computations in the proposed CSHM based FIR filter is reduced based on BCSE algorithm. The novel architectures ensure low-complexity as well as reconfigurability for FIR filters in the channelizers of SDRs. The architectures have been implemented on Virtex 2 xc2vp2-6fg256 FPGA and compared to numerous reconfigurable FIR filter architectures. Experimental results show that the proposed architectures out performs the previous architectures such as the systolization using distributed arithmetic and DA based design.

References:

[1] H. T. Kung,"Why systolic architectures?," IEEE Computer, vol. 15, no. 1, Jan. 1982, pp. 37–45.

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

[3] R.Wyrzykowski and S. Ovramenko, Flexible systolic architecture for VLSI FIR filters, Proc. Inst. Elect. Eng.—Comput. Digit Techniques, vol. 139, no. 2, Mar. 1992, pp. 170–172.

[4] B. K. Mohanty and P. K. Meher, Cost-effective novel flexible cell level systolic architecture for high throughput implementation of 2-D FIR filters, Proc. Inst. Elect. Eng.—Comput. Digit. Techniques, vol. 143, no. 5, Nov. 1996, pp. 436–439.

[5] B. K. Mohanty and P. K. Meher, Novel flexible systolic mesh architecture for parallel VLSI implementation of finite digital convolution, IETE J. Res., vol. 44, no. 6, Nov. 1988, pp. 261–266.

[6] D. G. Elliott, M. Stumm, W. M. Snelgrove, C. Cojocaru, and R.Mckenzie, Computational RAM: Implementing processors in memory, IEEE Trans. Design Test Comput., vol. 16, no. 1, Jan. 1999, pp. 32–41.

[7] J.-I. Guo, C.-M. Liu, and C.-W. Jen, The efficient memory-based VLSI array design for DFT and DCT, IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process, vol. 39, no. 10, Oct. 1992, pp. 723–733.

[8] D. F. Chiper, A systolic array algorithm for an efficient unified memory-based implementation of the inverse discrete cosine transform, in IEEE Conf. Image Process, Oct. 1999, pp. 764–768.

[9] D. F. Chiper, M. N. S. Swamy, M. O. Ahmad, and T. Stouraitis, Systolic algorithms and a memory-based design approach for a unified architecture for the computation of DCT/DST/IDCT/IDST, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 6, Jun. 2005, pp. 1125–1137.

[10] P. K. Meher and M. N. S. Swamy, New systolic algorithm and array architecture for prime-length

discrete sine transform, IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 3, Mar. 2007, pp. 262–266.

[11] P. K. Meher, J. C. Patra, and M. N. S. Swamy, High-throughput memory-based architecture for DHT using a new convolutional formulation, IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 7, Jul. 2007, pp. 606–610.

[12] S. A. White, Applications of the distributed arithmetic to digital signal processing:Atutorial review, IEEE ASSP Mag., vol. 6, no. 3, Jul. 1989, pp. 5–19.

[13] M. Mehendale, S. D. Sherlekar, and G.Venkatesh, Area-delay trade off in distributed arithmetic based implementation of FIR filters, in Proc. 10th Int. Conf. VLSI Design, Jan. 1997, pp. 124–129.

[14] H. Yoo, D. V. Anderson, Hardware-efficient distributed arithmetic architecture for high-order digital filters, in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, (ICASSP'05), Mar. 2005, vol. 5, pp. v/125–v/128.

[15] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, LMS adaptive filters using distributed arithmetic for high throughput, IEEE Trans. Circuits Systems I, Reg. Papers, vol. 52, no. 7, Jul. 2005, pp. 1327–1337.

[16] S.-S. Jeng, H.-C. Lin, S.-M. Chang, FPGA implementation of FIR filter using M-bit parallel distributed arithmetic, in Proc. 2006 IEEE Intl. Symp. Circuits Syst. ISCAS 2006, May 2006, pp. 4.

[17] Y. H. Chan and W. C. Siu, On the realization of discrete cosine transform using the distributed arithmetic, IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 39, no. 9, Sep. 1992, pp. 705–712.

[18] H.-C. Chen, J.-I. Guo, T.-S. Chang, and C.-W. Jen, "A memory-efficient realization of cyclic convolution and its application to discrete cosine transform," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 3, Mar. 2005, pp. 445–453.

[19] P. K. Meher, S. Chandrasekaran, and A. Amira, FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic, IEEE Trans, Signal Processing, vol. 56, no. 7, Jul. 2008, pp. 3009–3017.

[20] P. K. Meher, Unified systolic-like architecture for DCT and DST using distributed arithmetic, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 5, Dec. 2006, pp. 2656–2663.

[21] J.-I. Guo, C.-M. Liu, and C.-W. Jen, The efficient memory- Based VLSI array design for DFT and DCT, IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 39, no. 10, Oct. 1992, pp. 723–733.

[22] J.L Mazher Iqbal, S.Varadarajan, High Performance Reconfigurable Balanced Shared Memory Architecture For Embedded DSP, in International Journal of Computer Science & Information Security, vol.8, no.4, 2010, pp198-206.

[23] R.Mahesh, A.P.Vinod, New Reconfigurable Architectures for Implementing FIR Filters with Low Complexity, IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, vol. 29, no. .2, pp.275-288, 2010.

[24] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, Computation sharing programmable FIR filter for low-power and high-performance applications, IEEE J. Solid State Circuits, vol. 39, no. 2, 2004, pp. 348–3.

[25] Jae-Jin Lee and Gi-Yong Song, Implementation of a Bit-level Super-Systolic FIR Filter, IEEE Asia-Pacific Conference on Advanced System Integrated Circuits(AP-ASlC2004)I Aug. 4-5, 2004, pp.206-209.

[26] H. Yoo and D. V. Anderson, Hardware-efficient distributed arithmetic architecture for high-order digital filters, in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP) , vol. 5, 2005, pp. v/125–v/128.

[27] J.L Mazher Iqbal, S. Varadarajan A New Algorithm for FIR Digital Filter Synthesis for a Set of Fixed Coefficients, European Journal of Scientific Research**,** EJSR, Vol.59 No.1, July 2011, pp.104-114.

[28] G. Rubin, M. Omieljanowicz, A. Petrovsky, Reconfigurable FPGA–based Hardware Accelerator For Embedded DSP, MIXDES 2007, Bialystok Technical University, Poland, 21 – 23 June 2007, pp 147-151**.**

[29]. Jeong-Ho Han,In-Cheol Park, FIR Filter Synthesis Considering Multiple Adder Graphs for a Coefficient. In: IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems, vol.27, no.5, 2008, pp.958-962.

[30] J.L Mazher Iqbal, S. Varadarajan, "Performance Comparison of Reconfigurable Low Complexity FIR Filter Architectures", International Conference on Computational Intelligence and Information Technology, CIIT, LNCS-CCIS, Vol. 250, Nov 2011, pp.637-642.

[31] Mahesh, R., Vinod, A. P, A New Common Subexpression Elimination Algorithm for Implementing Low Complexity FIR Filters in Software Defined Radio Receivers. In: Proc.

IEEE Int Symp. on Circuits and Systems, 2006, pp. 4515 – 4518.

[32] J.L Mazher Iqbal, S. Varadarajan, High Performance Reconfigurable FIR Filter Architecture Using Optimized Multipliers, Springer Journal-Circuits System and Signal Processing, Journal No.34, 2012, DOI 10.1007/s00034-012-9473-3.

[33] F. Neri, Agent based modeling under partial and full knowledge learning settings to simulate financial markets, AI Communications, IOS Press, printing, 2012.

[34] Masakazu Higuchi, Shuji Kawasaki, Kazuki Katagishi, A Design Method of Narrow Band FIR Filters Based on Fluency Sampling Function of Quadratic Piecewise Polynomial, WSEAS Transactions on systems, Issue 6, Volume 8, June 2009, PP 733-742.

[35] NIKOS E. MASTORAKIS, New General Transformations for 2-D FIR and IIR Filters Design, WSEAS Transactions on circuits and systems, Issue 1, Volume 10, January 2011, PP 1-9.

[36] NIKOS E. MASTORAKIS, New Method for Designing 2-D (Two-Dimensional) IIR Comb Filters, WSEAS Transactions on systems, Issue 1, Volume 10, January 2011, PP 1-6.