

A Comparative Study on LUT and Accumulator Radix-4 Based Multichannel RNS FIR Filter Architectures

Britto Pari. J^{#1}, Joy Vasantha Rani S.P^{*2}

[#]Research Scholar, Department of Electronics Engineering, MIT campus, Anna University, Chennai, Tamil nadu, India (Mobile: 9791072673)

¹brittopari@yahoo.co.in

^{*}Assistant Professor, Department of Electronics Engineering, MIT campus, Anna University, Chennai, Tamil nadu, India (Mobile: 9444167996)

²joy_mit@annauniv.edu

Abstract - In this paper, a comparative study of two architectures proposed for multichannel reconfigurable FIR filter are performed in terms of complexity and speed. The proposed architectures, viz, dual port memory based LUT multiplier and accumulator based radix-4 multiplier architectures, are designed to reduce the complexity and to improve the speed of operation of multiplier used in multichannel reconfigurable FIR filter. Both the architectures accepts residues of given binary input in which the 3n-bit binary input is converted into three residues using binary to Residue Number System (RNS) converter, and then processed in three FIR sub filters constructed in direct form. The reconfigurable structure is achieved by combining Power of Two (PoT) FIR sub modules and altering the filter taps based on select signals. The proposed designs can be realized up to 20-taps and has been tested for 4, 8, 16 and 20 taps. The architectures have been realized in Verilog HDL and synthesized using Altera FPGA device Stratix II EP2S15F672C5. The performance comparison of two architectures shows that dual port memory based LUT multiplier architecture significantly reduces the area by 20% and accumulator based Radix-4 multiplier increases the speed by 90% regardless of the number of taps.

Keywords: Multichannel FIR filter, Residue number system, Look-up Table, Reconfigurable Architecture, Power of Two.

I. INTRODUCTION

Large sampling array size is feasible with the advancement of recent DSP technology, and can be used in variety of applications such as communication and multimedia in which the information from single channel may be erroneous. So multichannel signal processing is essential as far as reliability and efficient processing of those signals are concerned. The sampled multichannel data are processed in FIR filter through time multiplexed mechanism to achieve resource optimization [1].

With the advent of software defined radio (SDR), the research has been concentrated on reconfigurable realization of FIR filters [2][3] mainly due to the need of high flexibility and low complexity[4][5][6][7]. The digit-based reconfigurable architecture presented in [3] provides a flexible and low power solution with a wide range of precision and tap length of FIR filters. Conventionally, the reconfigurable FIR filters are designed based on programmable multiply-accumulate (MAC) architecture [6], systolic architecture [7] and Programmable Shift Method (PSM) [5]. The performances of the designs are analyzed in terms of hardware complexity, power consumption and throughput. The programmable MAC architectures [6] consume low power with reduced supply voltage and it requires large area. Even though systolic based architecture reduces the complexity, it increases the latency when the order of the filter increases. [7] The PSM based reconfigurable architecture achieves less complexity and significant speed due to the presence of programmable shifters [5]. Although the researchers addressed the problem of reducing the hardware complexity and low power, speed of operation is vital while designing the channel filters in SDR. Since the filter has to be operated at high sampling frequency in front end of SDR to meet the speed constraint, the PSM based reconfigurable architecture is not a significant method in the design of channel filters in SDR.

The speed of operation and parallelism of digital filters is improved when RNS number system is used [8][9]. RNS number system requires conversion from binary number to residue numbers using a set of moduli. This leads to efficient implementation of arithmetic operations in which the carry propagation is minimized by decomposing those operations into smaller ones. Several researchers [10][11][12] have attempted to implement the RNS based digital filters. The n-bit or 2n-bit adder based implementations [13] for residue converters changes the moduli sets used in ROM based approach [14]. The adder based implementation helps in achieving improvement both in area and speed.

The direct implementation of N-tap FIR filter requires N MAC operations, which are expensive to implement in hardware due to its logic complexity and area requirement. Memory based structures are well-suited for the implementation of many digital signal processing (DSP) algorithms, which involve multiplication with a fixed set of coefficients. There are two basic variants of memory based techniques. One of them is based on distributed arithmetic (DA) for inner-product computation and the other is based on the computation of multiplication by look-up-table (LUT). In the LUT-multiplier-based approach [4], multiplications of input values with a fixed co-efficients are performed by a LUT consisting of all possible pre-computed product values corresponding to all possible values of input while in the DA based approach, LUT is used to store all possible values of inner-products of a fixed N –point bit-vector. If the inner-products are implemented in a straight-forward way, the memory-size of LUT multiplier based implementation increases exponentially with the word-length of input values, while that of the DA based approach increases exponentially with the inner-product-length. Attempts have been made to reduce the memory space in DA-based architectures using offset binary coding (OBC) and group distributed technique [17]. A decomposition scheme [16] is one of the techniques used for reducing the memory size of DA based implementation of FIR filter. But, it is observed that the reduction of memory size achieved by such decompositions is accompanied by the increase in latency as well as the increase in the number of adders and latches. In this paper proposed two new multichannel FIR filter architectures wherein the speed of operation increases or complexity of hardware reduces. Hardware complexity is reduced by manipulating the odd multiples of the fixed coefficient in the LUT design[4], whereas the speed of operation is increased by reducing the partial products required for accumulator based radix-4 multiplier[18].

In the proposed architectures, a multichannel FIR filter is implemented through time division multiplexing mechanism with the help of two different architectures such as accumulator based radix-4 multiplier and dual port memory based LUT multiplier and their performances are analyzed. The use of time division multiplexing mechanism enables to optimize the resources utilized. The inputs are in residue format and coefficients are in fixed point binary representation. The three residues are processed in the FIR sub filters which are implemented using proposed architectures, where the number of taps can be programmable. The performances of the proposed architectures are analyzed with reconfiguration in terms of area and speed by varying the number of taps.

The rest of the paper is organized as follows. In Section II, preliminaries about RNS number system, radix multiplier and LUT multiplier are described. Section III describes the details of architecture design and tells how reconfiguration based multichannel FIR filter is implemented. The performance of the designs is analyzed and discussed in Section IV. Finally, Section V concludes the paper in brief.

II. PRELIMINARIES

This section provides brief technical backgrounds on RNS number system, accumulator based radix-4 multiplier and LUT based multiplier which is background for the proposed architectures.

A. Residue Number System

The process of carry propagation is eliminated by decomposition of an integer into smaller parts (residues) thereby performing parallel independent operations. This method of decomposition is known as Residue Number System.(RNS) The residue number system [18] always has a set of relative prime numbers {m₁, m₂, ... m_r}.

Let X be an ordered set of residues {x₁, x₂,... x_r},where x_i= X mod m_i. The integer should be in the dynamic range {0, M}, where M is the product of relative prime numbers in moduli set. In residue arithmetic [19][20][8][9], the choice of moduli sets and the conversion of residue to binary numbers are important issues. The residue number system used in our work is based on the set of moduli (2ⁿ -1, 2ⁿ, 2ⁿ +1)[9]. This allows the process of residue addition using binary adder. By splitting 3n-bit binary input integer into three equal sized parts and then performing modulo addition, the conversion of binary to RNS is obtained. The three residues, (2ⁿ -1).(2ⁿ).(2ⁿ +1) are n-bit binary integers except for the residue corresponding to modulo 2ⁿ +1 which is a (n+1)-bit integer. The 3n-bit binary representation of X is

$$X = \left| \begin{array}{c} X_{3n-1} \dots X_{2n} \\ \leftarrow k_2 \\ X_{2n-1} \dots X_n \\ \leftarrow k_1 \\ X_{n-1} \dots X_0 \\ \leftarrow k_0 \end{array} \right|$$

The RNS representation of X can be calculated through modulo arithmetic operations. Hence the three residues can be written as

$$x_1 = | p_1 + k_1 |_{2^n - 1}$$

$$x_2 = k_0$$

$$x_3 = |p_2 - k_1|_{2^{n+1}} \tag{1}$$

where $p_1 = |k_2 + k_0|_{2^{n-1}}$ and $p_2 = |k_2 + k_0|_{2^{n+1}}$

The RNS to binary converter generates the 3n-bit binary integer output from the FIR sub-filters output residues $y_1, y_2,$ and y_3 that can be implemented using ROM-less adder based converter as described in [13]. The equations needed for RNS to binary conversion are given below

$$Z = y_2 + 2^n \cdot Y \tag{2}$$

where $Y = |A + 2^n \cdot B|_{2^{2n-1}}$

$$A = ((y_1 + (y_{1_0} \oplus y_{3_0}) \cdot 2^n) + (2^n - 1 - y_3) + (2^n - 1)) / 2$$

$$B = ((y_1 + (y_{1_0} \oplus y_{3_0}) \cdot 2^n) + y_3 + 2(2^n - 1 - y_2)) / 2$$

where y_{1_0} and y_{3_0} are the least significant bits of y_1 and y_3 respectively.

B. Accumulator Based Radix-4 Multiplier

Parallelism is enhanced using high speed multipliers which in turn reduce the number of subsequent stages. The speed of the operation is increased by using Booth's bit pair recoding algorithm by reducing the number of partial products. The original version of Booth algorithm (Radix-2) for n-bit numbers has 'n' number of partial products, whereas in Booth's bit pair recoding algorithm (radix-4) partial products are reduced as n/2 for n-bit numbers. This algorithm recodes the pair of three bits of the multiplier and generates the partial products simultaneously. Thus there is a reduction in the total number of partial products. In this work, accumulator based radix-4 multiplier is used and it generates the number of partial products as n/2 for (nxn)-bit multiplication. The partial products are shifted and accumulated using carry lookahead adder (CLA) that accumulates partial products. Hence, it is concluded that by reducing the partial products with the help of accumulator based radix-4 multiplier the speed of the operation is increased. The multiply and accumulate (MAC) architecture executes the multiplication operation and accumulates the result for every clock cycle. The inputs of MAC are one of the three residues obtained from binary to RNS converter and the filter coefficient represented in fixed point binary. The architecture of accumulator based radix-4 multiplier is shown in Fig.1.

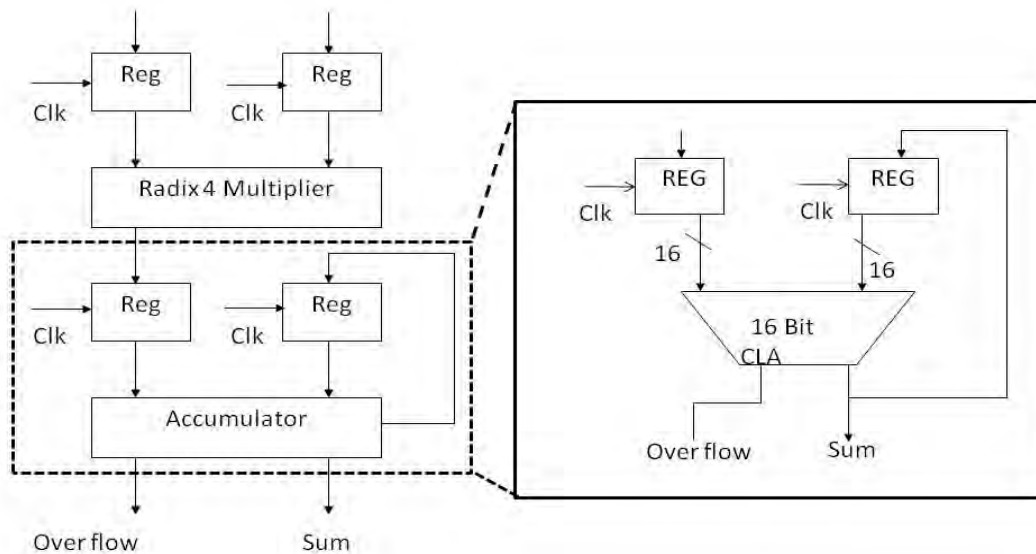


Fig.1 Block diagram of accumulator based radix-4 multiplier

C. Dual port Memory based LUT Multiplier

Let X be an input which is multiplied with fixed coefficient H. If the length of X is assumed to be an unsigned binary number L, 2^L values of X is possible, and hence, the product $P = (H \cdot X)$ contains 2^L possible values. Therefore, the LUT consisting of pre-computed product values corresponding to all possible values of X requires a memory unit of 2^L words for the conventional implementation of memory-based multiplication. The dual port memory based LUT multiplier proposed in [4] shows that it is enough to store only $(2^{L/2-1})$ words corresponding to the odd multiples of H in the LUT. One of the possible product words is zero, while all the rest $(2^{L/2-1} - 1)$ are even multiples of H which could be derived by left-shift operations from of the odd multiples of H. By resetting the LUT output, address corresponding to (0000) can be obtained. The concept behind memory based multiplication [4] is given in Table I.

This multiplier consists of memory with the size of eight words of (w+4) bit width and a 3-to-8 line address decoder, a NOR-cell, a shifter, a 4-to-3 bit encoder to map the 4-bit input operand to 3-bit LUT-address and a control circuit for generating the control word and RESET signal for the shifter and the NOR-cell respectively.

The 8-bit input binary number $[x_7, x_6, x_5, \dots, x_0]$ is split into two 4-bit numbers and these are given to two separate 4 to 3-bit the encoder which produces three address bits $[d_2 d_1 d_0]$ for dual port memory per the relation

$$d0 = \overline{(x0.x1).(x1.x2).(x0 + x2 + x3)} \tag{3a}$$

$$d1 = \overline{(x0.x2).(x0 + (x1.x3))} \tag{3b}$$

$$d2 = x0.x3 \tag{3c}$$

Similarly address bits $[d_5 d_4 d_3]$ is generated from x_4, x_5, x_6 and x_7 . The address bits are given to the two different decoders that convert them into eight word-select signals $\{w_i, 0 \leq i \leq 7\}$. Hence the dual port memory can be accessed through two ports (decoders) of eight word select signals each. They are used to select the corresponding value from memory which is a multiple of the co-efficient in bit-inverted form. Then, control circuit performs the number of shifts on the memory output to obtain the even multiples. Three signals s_0, s_1 and RESET are generated according to the relation

$$s0 = \overline{x0 + (x1 + x2)} \tag{4a}$$

$$s1 = \overline{(x0 + x1)} \tag{4b}$$

$$RESET = \overline{(x0 + x1).(x2 + x3)} \tag{4c}$$

If the input combinations are (0010), (0110), (1010) and (1110), the output of LUT is shifted once. The inputs equals to (0100) and (1100) requires two shifts and the input (1000) requires three shifts. When the input word is (0000), then the reset operation is performed by setting the RESET bit as given in the logical relation (4c). The detailed diagram of dual port memory based LUT multiplier is shown in Fig.2. It is obtained by two 4-bit multipliers along with a shift-adder. The shift-adder left-shifts the output of the first shifter corresponds to most significant half of input by four bits and adds the result to the output of the other shifter.

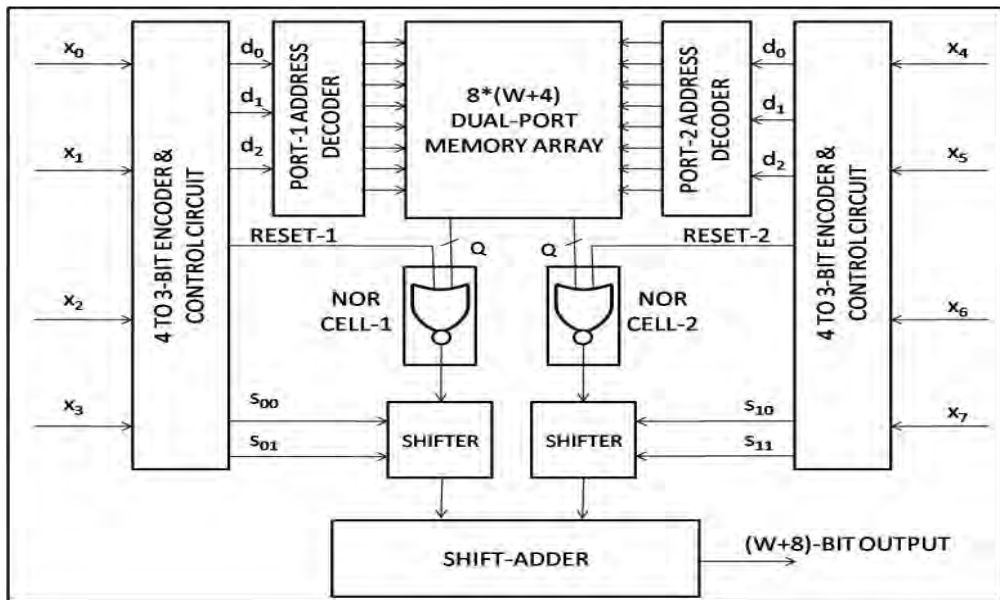


Fig.2. Dual port Memory based LUT Multiplier

TABLE I
LUT Input and Product Values for Word Length L=4

address d ₂ d ₁ d ₀	stored value	input x ₃ x ₂ x ₁ x ₀	product value	no. of shifts	control s ₁ s ₀
000	H	0001	H	0	00
		0010	2 ¹ x H	1	01
		0100	2 ² x H	2	10
		1000	2 ³ x H	3	11
001	3H	0011	3H	0	00
		0110	2 ¹ x 3H	1	01
		1100	2 ² x 3H	2	10
010	5H	0101	5H	0	00
		1010	2 ¹ x 5H	1	01
011	7H	0111	7H	0	00
		1110	2 ¹ x 7H	1	01
100	9H	1001	9H	0	00
101	11H	1011	11H	0	00
110	13H	1101	13H	0	00
111	15H	1111	15H	0	00

III. MULTICHANNEL RNS BASED RECONFIGURABLE FIR FILTER

A. Proposed Single Channel RNS based Reconfigurable FIR filter

As we know that the structure of the FIR filter has the multipliers in the form of MAC structure and delay blocks as the main building blocks. The performance of the DSP algorithms entirely depends upon multipliers in terms of critical path. Both the accumulator based radix-4 multiplier architecture which increases the speed of the operation and dual port memory based LUT multiplier architecture which reduces the complexity are proposed for Single channel Reconfigurable FIR filter. With the help of accumulator based radix-4 multiplier, the number of partial products is reduced to $n/2$. The 3-bit pair recoding generates the partial product set of 0, $\pm 1M$, $\pm 2M$, where M is the multiplicand. By means of dual port memory based LUT multiplier, the number of memory locations needed to store partial products is reduced from 2^L to $2^{(L/2)-1}$. Hence the memory size is greatly reduced as compared to conventional LUT based multiplication, which leads to reduction in complexity.

Let $X(n)$ and $Y(n)$ be the input and output sequences of the FIR filter respectively. Consider an N-tap FIR filter that can be formulated as

$$Y(n) = \sum_{k=0}^{N-1} h_k X(n-k) \tag{5}$$

where h_k is the k^{th} coefficient of the filter impulse response.

Subsequently, the FIR filter is partitioned into r sub filters, each corresponds to one residue x_i for a given moduli set $\{m_1, m_2, \dots, m_r\}$. Hence, the sub filter input is in its residue format as given in equation (1) and is denoted as,

$$x_i = |X_k|_{m_i} \quad i = 1, 2, \dots, r \tag{6}$$

The output of the sub filter is calculated by the equation (7) as,

$$y_i(n) = \sum_{k=0}^{N-1} h_k x_i(n-k) \tag{7}$$

The 3n-bit data X is decomposed into set of three residues and as in equation(1), it can be noted that the decomposed set of residues are processed in three sub filters and all the three filters use the same set of filter coefficients represented in fixed point binary.

The results of FIR sub filters in RNS form are converted back into binary form. The ROMless n-bit adder based converter, derived from new CRT algorithm [13], is used for the conversion of RNS back to binary. It improves the hardware complexity and speed since it can be implemented using fast parallel adders and multiplexers.

B. Proposed Multichannel RNS based Reconfigurable FIR filter

In the conventional structure of multichannel FIR filter [21], shown in Fig.3, dedicated filters are used for each channel. This results an increase in the speed of processing, at the expense of increased hardware complexity with reduced throughput. For efficient utilization of hardware resources, proposed multichannel reconfigurable FIR filter architecture shares the logic resources between multiple sample streams through time division mechanism. So even though the numbers of channels increase the logic area remains approximately constant. In this work, the input data from multichannel is transferred based on time-multiplexed mechanism and shares the same single channel reconfigurable FIR filter.

In multichannel reconfigurable FIR filter, the sampling frequency of the FIR filter is the ratio of output of clock frequency by the number of clock cycles required for processing the sampled input data at the output. For example, if the sampling frequency for a single channel FIR filter is f_s , a M-channel filter, process M sample streams, each with a sampling frequency of f_s/M .

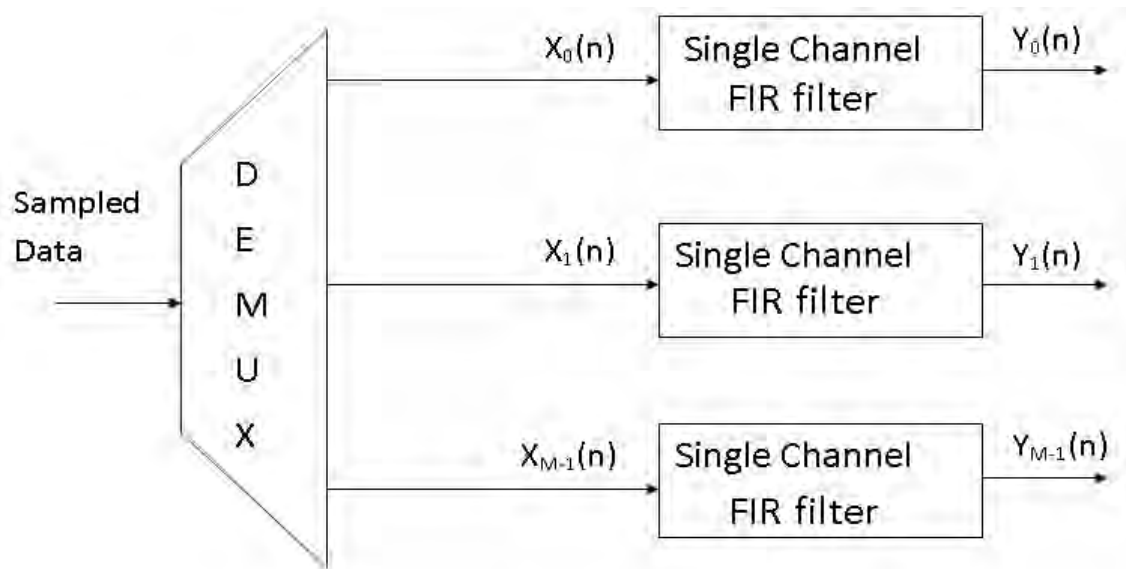


Fig.3 Conventional Structure of Reconfigurable FIR filter

Consider the general transfer function of FIR filter,

$$Y(z) = X(z) \sum_{k=0}^{n-1} h_k z^{-k} \tag{8}$$

where $h_0, h_1, h_2, \dots, h_{n-1}$ are the filter coefficients and the z^{-1} represents the delay elements.

Now the equation (8) can be decomposed into Power of two sub modules in which the number of coefficients is the increasing powers of two as given in equation (9).

$$H(z) = h_0 + z^{-1}(h_1 + h_2 z^{-1}) + z^{-3}(h_3 + h_4 z^{-1} + h_5 z^{-2} + h_6 z^{-3}) + \dots + z^{-(n-1/2)}(h_{n-1/2} + h_{n+1/2} z^{-1} + \dots + h_{n-1} z^{-(n-1/2)}) \tag{9}$$

The sub modules can be described as,

$$H(2^0) = h_0, H(2^1) = h_1 + h_2 z^{-1}, H(2^2) = h_3 + h_4 z^{-1} + h_5 z^{-2} + h_6 z^{-3} \text{ and so on.}$$

For N-tap FIR filter with ‘n’ number of Coefficient the equation (9) can be written as,

$$H(z) = H(2^0) + H(2^1)z^{-1} + H(2^2)z^{-3} + H(2^3)z^{-7} + H(2^4)z^{-15} + \dots + H(2^m)z^{-(2^m-1)} \quad (10)$$

where $N=2^{m+1}-1$ taps, for $m=0,1,2,3,\dots$

It can be concluded from equation (10) that for any N-tap filter, the sub modules can be combined with respective delay elements. The reconfigurable FIR filter structure is implemented in the proposed multichannel FIR filter structure as shown in Fig.4. Here the selection of filter tap is based on the encoder output from 2^m outputs by the 'm' select lines. This system provides flexibility by way of selecting the number of taps for the required application.

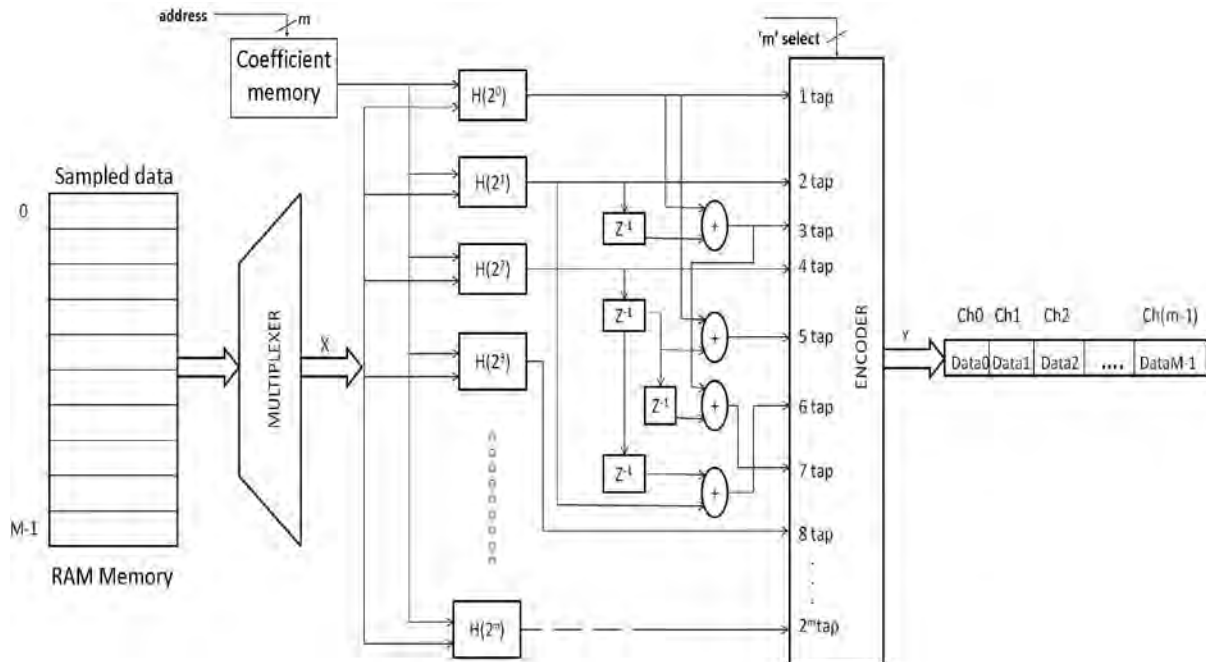


Fig.4 Multichannel Reconfigurable FIR sub-filter

IV. RESULTS AND DISCUSSION

The proposed architectures are designed by parameterizable Verilog cores. A key advantage of hardware description languages (HDLs) is that all the statements are executed in concurrent manner. The performances of both single channel and multichannel reconfigurable architectures were synthesized on the FPGA device Altera Stratix II EP2S15F672C5.

A. Single channel RNS based Reconfigurable FIR filter

The proposed Reconfigurable RNS based FIR Filter architectures are implemented using two methods: (i) accumulator based radix-4 multiplier and (ii) Dual port Memory based LUT Multiplier, and synthesized using Altera FPGA device Stratix II EP2S15F672C5. The performance results of RNS FIR filter architectures are analyzed in Table II. From RNS based FIR filter structures, it can be observed that the increase in the number of taps linearly increases delay and reduces the frequency.

While comparing the performance results, it is found that the speed of the operation is increased in accumulator based Radix-4 multiplier by reducing the partial products and parallelism of RNS structure, but low complexity was achieved using New LUT multiplier based implementation, because the memory size is reduced to nearly half of the conventional LUT based multiplication. For a 4-tap reconfigurable FIR filter, $H(z)$ is split into $H(2^0)$, $H(2^1)$ and $H(2^2)$ according to equation(11) as a sub module for variable tap implementation from tap 1 to 4. Similarly 8-tap can be split into $H(2^0)$, $H(2^1)$, $H(2^2)$ and $H(2^3)$. In the same way 16-tap and 20-tap reconfigurable FIR filter are implemented.

TABLE II
Performance results of RNS based single channel FIR filter using Altera Stratix II EP2S15F672C5

Performance measures	Dual port Memory based LUT				ACC-Radix4			
	4-tap	8-tap	16-tap	20-tap	4-tap	8-tap	16-tap	20-tap
No of taps	4-tap	8-tap	16-tap	20-tap	4-tap	8-tap	16-tap	20-tap
No of input bits	24	24	24	24	24	24	24	24
No of Logic Elements	967	1639	3976	4317	1013	2020	4902	5042
Delay(ns)	5.558	8.154	10.273	10.402	2.222	2.500	2.695	2.815
Frequency(MHz)	179.22	122.64	97.34	96.14	450.05	400.00	371.06	355.24

TABLE III
Comparison of proposed reconfigurable RNS filter with other architectures

No. of taps	Pramodkumar [7] Frequency[MHz]*	Yoo et al [16] Frequency[MHz] *	Dual port Memory based LUT Multiplier Frequency[MHz] *	ACC-RADIX-4 Frequency[MHz] *
8	74.025	70.552	75.721	214.316
16	67.222	62.755	69.247	181.120

* Note: Device used: Xilinx Virtex-E XCV2000E

Table III compares the synthesis results of the 8 tap and 16 tap reconfigurable RNS based architectures with existing architectures proposed in [16] and [7] synthesized using the FPGA device Xilinx Virtex-E XCV2000E. Both the proposed architectures provide highest frequency of operation compared to the existing architectures. Hence it can be noted that the proposed reconfigurable RNS based architectures achieves high speed due to the parallelism of RNS.

A. Multichannel RNS based Reconfigurable FIR Filter

Time division multiplexed multichannel reconfigurable FIR filter is synthesized using the FPGA device, Altera Stratix II EP2S15f672C5 and the performances are analyzed in Table IV. Due to Time Division Multiplexing (TDM), logic resources are optimized. The proposed multichannel RNS based reconfigurable FIR filter results are also compared with the proposed single channel RNS based reconfigurable FIR filters, using Cadence RC compiler with 0.18 μ m CMOS technology as given in Table V. From Table IV and Table V, it is seen that TDM multichannel RNS FIR filter implementation is highly efficient in the utilization of logic resources, further it can be concluded that area is almost independent of the number of channels.

TABLE IV
Performance analysis of Multichannel Reconfigurable FIR filter

Performance measures	Dual port Memory based LUT		ACC-Radix4	
	16-tap	20-tap	16-tap	20-tap
No of taps	16-tap	20-tap	16-tap	20-tap
No. of Channels	24	24	24	24
Total Logic Elements	4168	4192	5032	5046
Delay(ns)	11.406	10.498	5.384	5.426
Power(mw)	324.69	324.89	325.07	325.78
Freq(MHz)	97.12	95.26	185.74	184.30

TABLE V
Synthesis results of Cadence RC compiler for RNS based multichannel Reconfigurable 20-tap FIR filter

Parameter	Proposed design with single channel		Proposed design with multichannel			
	RNS-Acc-radix-4	RNS-LUT	RNS-Acc-radix-4		RNS-LUT	
No of channels	1	1	3	24	3	24
Area(mm ²)	0.464	0.448	0.421	0.502	0.400	0.448
Power(mw)	21.68	142.51	56.21	67.12	136.23	142.65

V. CONCLUSION

In this paper an efficient high speed time division multiplexed multichannel reconfigurable RNS based FIR filter architectures have been discussed which can be effectively used for implementing any N-tap filters. The N-tap Reconfigurable FIR filter thus implemented combines PoT (2^m) sub modules thereby increasing the flexibility. The results of RNS based reconfigurable FIR filter architectures are analyzed and compared with respect to area and speed, and show better performance in terms of area with dual port memory based LUT multiplier and improvement in speed of operation with accumulator based Radix-4 multiplier. Comparing the results, the area gets reduced in LUT based RNS FIR Filter since half the memory was reduced by manipulating the odd multiples of the fixed coefficient, and the speed of operation was increased in RNS FIR Filter using Accumulator based radix-4 multiplier due to the reduction in the partial products. These architectures were synthesized and compared in various platforms like Altera, Xilinx and Cadence. The results of the proposed reconfigurable RNS architectures were also compared with existing architectures and shows improved performance in terms of frequency. Thus the proposed reconfigurable architectures achieve high speed and low complexity and the reprogrammability of FPGA technology make the architectures a viable alternative to the development of reconfigurable hardware for real time signal processing applications.

REFERENCES

- [1] Digital Signal Processing solution: "Designing for Optimal Results High-Performance DSP using Virtex-FPGAs," Xilinx corporation, pp. 99-103, 2005
- [2] David V. Anderson, and Erhan Özalevli, "A Reconfigurable Mixed-Signal VLSI implementation of Distributed Arithmetic Used for Finite Impulse Response Filtering," IEEE Transactions on Circuits and Systems-I: Regular Papers, Vol. 55, No. 2, March 2008.
- [3] Kuan-hung, and Tzi-Dar, "A low power digit based Reconfigurable FIR Filter," IEEE Transactions on circuits and systems, Vol. 53, Aug 2006.
- [4] Pramod Kumar Meher, "New Approach to Look Up Table Design and Memory-Based Realization of FIR Digital Filter," IEEE Transactions on circuits and systems irregular papers, Vol. 57, No. 3, March 2010.
- [5] R.Mahesh, and A.P Vinod, "New Reconfigurable Architectures for implementing FIR Filter with low complexity," IEEE Transactions on computer aided design of integrated circuits and systems, Vol. 29, Feb 2010.
- [6] T. Solla, and O. Vainio, "Comparison of programmable FIR filter architectures for low power," in Proc. of 28th European Solid State Circuits Conference, pp. 759-762, September 24 – 26, 2002
- [7] Pramod Kumar Meher, "FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic", IEEE Transactions on signal processing, Vol. 56, No. 7, July 2008.
- [8] Y.wang, "Residue to binary converters based on new Chinese remainder theorems," IEEE Transactions on Circuits and Systems II, pp. 197-206, Mar 2000.
- [9] B.Vinnakota and V.V Bapeswara Rao, "Fast conversion techniques for Binary –Residue number system," IEEE Transactions on Circuits and Systems, Vol. 41, No. 12, Dec 1994.
- [10] Naresh R. Shanbhag and Raymond .E.Sifred, "A single chip pipelined 2-D FIR filter using Residue Arithmetic," IEEE Journal of solid state circuits, vol.26, No.5, May 1991
- [11] Andreas Lindahl, and Lars Bengtsson, "Low Power FIR filter using combined residue and radix-2 signed digit representation," DSD'05, May 2005
- [12] W. J. Jenkins, "Techniques for residue-to-analog conversion for residue encoded digital filters," IEEE Transactions on Circuits and Systems Vol. CAS-25, pp.555–562, July 1978.
- [13] Yuke Wang, Xiaoyu Song, Mostapha Aboulhamid, and Hong Shen, "Adder Based Residue to binary Number Converters for (2^n-1 , 2^n , 2^n+1)," IEEE Transactions On Signal Processing, Vol. 50, No. 7, July 2002.
- [14] D. Gallaher, F. Petry, and P. Srinivasan, "The digital parallel method for fast RNS to weighted number system conversion for specific moduli($2^n - 1$; 2^n ; $2^n + 1$)," IEEE Transactions on Circuits and Systems II, vol. 44, pp. 53–57, Jan. 1997.
- [15] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital filter for PCM encoded signals," U.S. Patent 3 777 130, Dec. 4, 1973.
- [16] 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), Mar. 2005, vol. 5, pp. v/125–v/128.
- [17] Xilinx Incorporation, "The Role of Distributed Arithmetic in FPGA-based signal Processing," Xilinx application notes, San Jose, CA.
- [18] Young-Ho Seo and Dong-Wook Kim, "A New VLSI Architecture of parallel multiplier –Accumulator Based on Radix-2 Modified Booth Algorithm," IEEE Transactions on VLSI Systems, Vol.18, No.2, Feb 2010.

- [19] Chip-Hong Chang, "Radix-8 Booth Encoded Modulo Multipliers with Adaptive Delay for High Dynamic Range Residue Number System," IEEE Transactions on Circuits and Systems—I: Regular Papers, 2010
- [20] Shuangching Chen and Shugang Wei "Performance Evaluation of Signed-Digit Architecture for Weighted-To-Residue and Residue-to-Weighted Number Converters with Moduli Set $(2^n - 1, 2^n, 2^{n+1})$," IPSJ Digital Courier, Vol. 2, June 2006.
- [21] Liu Ming, Yan Chao," The Multiplexed Structure of Multi-channel FIR Filter and its Resources Evaluation," International Conference on Computer Distributed Control and Intelligent Enviromental Monitoring, IEEE 2012.