electronics

MDPI

Article

Reconfigurable Frequency Response Masking Multi-MAC Filters for Software Defined Radio Channelization

Subahar Arivalagan [1,*], Britto Pari James [1] and Man-Fai Leung [2]

[1] Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai 600062, India; brittopariece@veltech.edu.in

[2] School of Computing and Information Science, Faculty of Science and Engineering, Anglia Ruskin University, Cambridge CB1 1PT, UK; man-fai.leung@aru.ac.uk

* Correspondence: vtd1007@veltech.edu.in

Abstract: Mobile technology is currently trending toward supporting multiple communication standards on a single device. This means that some reconfigurable techniques must be the foundation of their design. The two essential requirements of channel filters are minimized complexity and reconfigurability. In this research, a novel extension of Frequency Response Masking (FRM) was investigated by employing Time Division Multiplexing (TDM)-based single Multiply and Accumulate (MAC) architecture using the principle of resource sharing to realize multiple sharp filter responses from a single prototype constant group delay low pass filter. This paper uses a single multiply and add units regardless of the quantity of channels and taps. The suggested reconfigurable filter was synthesized on technology based on 0.18-µm CMOS and put into practice. Further trials were carried out on Virtex-II 2v3000ff1152-4 FPGA device. The outcomes revealed that the suggested channel filter, which was synthesized using FPGA, provides 21.36% of the area curtail and 14.88% of power scaling down on average and put into practice using ASIC provides 5.18% of the area reduction and 9.08% of power scaling down on average.

Keywords: Time Division Multiplexing (TDM); Frequency Response Masking (FRM); Multiply and Accumulate (MAC); Finite Impulse Response (FIR); Mega Samples per Second (MSPS); Lookup Table (LUT)

check for updates

Citation: Arivalagan, S.; James, B.P.; Leung, M.-F. Reconfigurable Frequency Response Masking Multi-MAC Filters for Software Defined Radio Channelization. *Electronics* **2024**, *13*, 4211. https://doi.org/10.3390/electronics13214211

Academic Editor: George Angelos Papadopoulos

Received: 31 August 2024
Revised: 23 October 2024
Accepted: 24 October 2024
Published: 27 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The channelizer, which functions with the biggest sampling rate in the software radio (SR) section, is the component with the highest computational load [1–3]. Sample rate conversion (SRC), channel filtering, and digital down conversion (DDC) are all part of channelization [3]. Essentially, the channelizer uses a bank of channel filters to separate numerous low bandwidth channels from a baseband input signal. To minimize the sampling rate to baud rate multiplied by an integer, and to change the desired band's frequency to baseband which is chosen by employing the channel filters to baseband, the channelizer needs to include DDC and SRC stages [2,3]. Since they have guaranteed stability and a linear phase, FIR filters are used as channel filters. To achieve the strict specifications for wireless communication, the channelizer must have FIR filters with sharp, fast-moving, low-power transition bands [2]. The quantity of adders prescribed to perform the coefficient multiplication operation is primarily responsible for the complexity of the filtering operation. Several strategies have been put forth to make FIR filters less complex. A method known as Frequency Response Masking (FRM) was presented in [4] to create FIR filters with sharp transition bands. What makes the FRM methodology beneficial is that the filter's bandwidth remains unchanged, and the developed filter has a large number of sparse coefficients, making it less complex overall. The common subexpression elimination (CSE) method was used in [5,6] to eliminate redundant additions from FIR filters, thereby

reducing their complexity. The approach in [7] addresses the minimally complex application of traditional FIR channel filters by enhancing the CSE method. But the approach in [7] has a flaw with respect to attaining the distinct transition band, and as a result a very high order is needed. Additionally, the techniques in [5–7] represented coefficients for the filter using the canonic sign digit (CSD) approach. A CSE approach oriented binary depiction of coefficients was introduced in [8], and it produced a superior reduction of adders than [5–7].

Apart from simplicity, another essential need is reconfigurability for SDR channel filters since filter descriptions vary depending on the communication mode that is chosen. A repositionable digit FIR filter design was proposed in reference [9]. Due to the arbitrary assignment of digits that have other than zero in every tap and the number of taps, the proposed architecture was independent of the tap count. However, the architecture in [9] required a significant number of hardware resources, making the method unfeasible for SDR applications with limited resources. A fast and customizable FIR filter according to CSD was proposed in [10]. This architecture included a Wallace addition, a carry look-ahead addition at the end, and a programmable Booth encoding scheme based on CSD. The technique in [10] consistently produced high speeds, but power consumption was not taken into account. Two potent reconfigurable reduced complexity designs for FIR filters, based on BSE algorithm from [8], were presented in [11]. The FIR filters in [5–11] were primarily created using the Parks–Mc–Clellan algorithm, a traditional filter design technique that calls for greater order filters to fulfill the channel filters sharp transition-band requirements. Adaptability to every subfilter was added to the FRM filter configuration in [12], which was first proposed in [4] for the realization of lower-order fixed coefficient filters. With a moderate increase in speed, this architecture provides average area and power savings. But, for SDR applications, the architecture's percentage of area and power reduction is undesirable. A two-stage FRM filter structure is used in [13] to implement conventional FIR filters with low complexity. Time-multiplexed FRM (TMFRM) filter framework are covered in [14]. The technique demonstrates that the filters come through a decreased multiplier utilization, which led to a minimal complexity application of traditional FIR channel filters. But the technique in [13–16] has the flaw that the quantity of multipliers in the FIR filter is not decreased to the extent necessary for SDR application. Using the DA approach, an effective scheme for the potent sharable-LUT design of reconfiguration utilized filtering was presented, in which the DA units share LUTs for bit slices with varying weights. Furthermore, a very short reconfiguration latency can be attained by progressively modifying the filter accessories during runtime. In [17], a unique matrix grouped CSE formulary was introduced, demonstrating improvements over prevailing the CSE approach with regard to the terms reducing logic operators, larger route delay, effective latency, area and delay product (ADP), power and delay product(PDP). However, this method also results in an increased number of multipliers, which adds to the complexity. In [18], a method for natively designing fixed-point filters was presented, specifically targeting FPGA-utilized redesignable filters for SR applications. This approach produces high-quality filters in terms of stopband attenuation and transition bandwidths. However, when implemented for higher orders, the filters become more complex, leading to increased area and power consumption. In [19], a FRM scheme was employed, allowing the maskable filters to be adjusted to specific frequencies using an adjustable bandwidth filter configured in a Farrow framework. However, the Farrow structure is primarily intended for interpolation and fractional delay filters, making it less efficient or unsuitable for general-purpose FIR filter design that requires specific frequency response characteristics. Additionally, the polynomial-based approximation in the Farrow structure may experience accuracy issues at higher frequencies, with errors increasing as the signal frequency approaches the Nyquist limit, rendering it less effective for high-frequency signal processing.

In this paper, to realize multiple sharp filter responses from an individual prototype FIR filter that is low-pass as well as with linear-phase, an innovative dilatation of the FRM was propounded by employing single MAC architecture based on Time Division

Multiplexing (TDM) and the resource sharing principle. Regardless of the quantity of taps and channels, this paper only used one multiplier and one adder. Using 0.18-µm CMOS technology, the proposed reconfigurable filter was created, tested, and implemented on a Virtex-II 2v3000ff1152-4 FPGA. This system provides adaptability with area and power reduction at architectural as well as filter levels.

This is how this paper is organized. An analysis of the FRM scheme is provided in Section 2. The framework for the devised Finite Impulse Response filter is explained in Section 3. The exploratory outcomes are displayed in Section 4. Our assessments are in Section 5.

2. Analysis of FRM Method

Here is a brief overview of the FRM method in [4] in this division. The fundamental idea of the FRM approach is to use multiple broad transition-band subfilters to frame the overall filter, which has a sharp transition-band. The prototype filter $H_a(z)$, which has odd-length (Na) as well, has a low-pass with linear phase as well as impulse response and is symmetrical, while its complementary filter $H_c(z)$ has the ensuing expressions:

$$H_c(z) = z^{-\frac{(N_a-1)}{2}} - H_a(z) \tag{1}$$

$H_a(z^M)$ and $H_c(z^M)$ are the system functions of filters that are created by changing M delays for every delay component of the modal filters, $H_a(z)$ and $H_c(z)$. Compared to $H_a(z)$, the widths of transition are more restricted by factor M for the filters $H_a(z^M)$ and $H_c(z^M)$. As illustrated in Figure 1, the FRM approach involves $H_{Ma}(z)$ along with $H_{Mc}(z)$, which are pair of masking filters cascading with $H_a(z^M)$ as well as $H_c(z^M)$, correspondingly. The overall filter with transfer function H(z) is provided through

$$H(z) = H_a\left(z^M\right)H_{Ma}(z) + H_c\left(z^M\right)H_{Mc}(z) \tag{2}$$

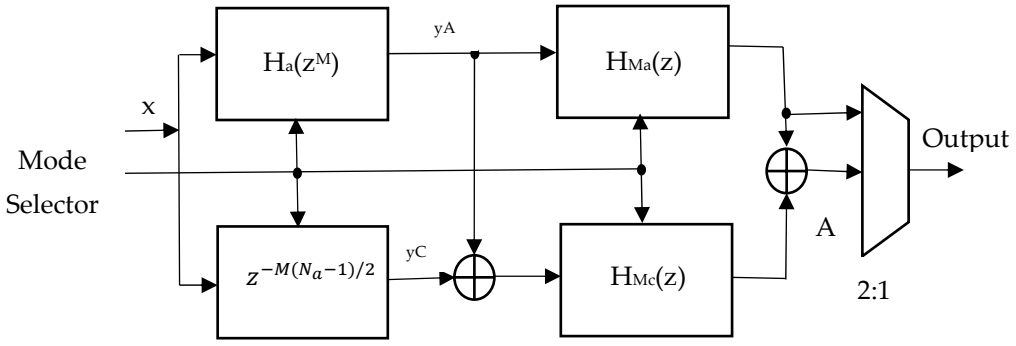


Figure 1. Design of FIR filter using the FRM technique.

Observe that $M(N_a - 1)$ must be an even number, and the collective delays of $H_{Ma}(z)$ and $H_{Mc}(z)$ are the same. Solving the expressions is a step in the creation process for the subfilters shown in Figure 1 [4].

$$m = \lfloor f_p M \rfloor, f_{ap} = f_p M - m, f_{as} = f_s M - m, f_{map} = f_p$$

$$f_{mas} = \frac{m + 1 - f_{as}}{M}, f_{mcp} = \frac{m - f_{ap}}{M} \text{ and } f_{mcs} = f_s \tag{3}$$

where the entire filter's edges of the passband is f_p and the stopband is f_s, $H_a(z)$ the modal filter with its edges of the passband is f_{ap} and the stopband is f_{as}, $H_{Ma}(z)$ the masking filter for interpolated filter $H_a(z^M)$ with its edges of the passband is f_{map} and the stopband is f_{mas}, $H_{Mc}(z)$ the masking filter for complimentary interpolated filter $H_c(z^M)$ with its edges of the passband is f_{mcp} and the stopband is f_{mcs}. $\lfloor x \rfloor$ represents the biggest integer smaller than x. The $H_a(z)$ up sampling rate is denoted by M. The realization of the exact transition-band

FIR filter is possible to determine simply by carefully choosing the edges of the passband as well as the stopband of the modal along with the masking filters. In Figure 2 [4], the FRM filter design process is visually represented.

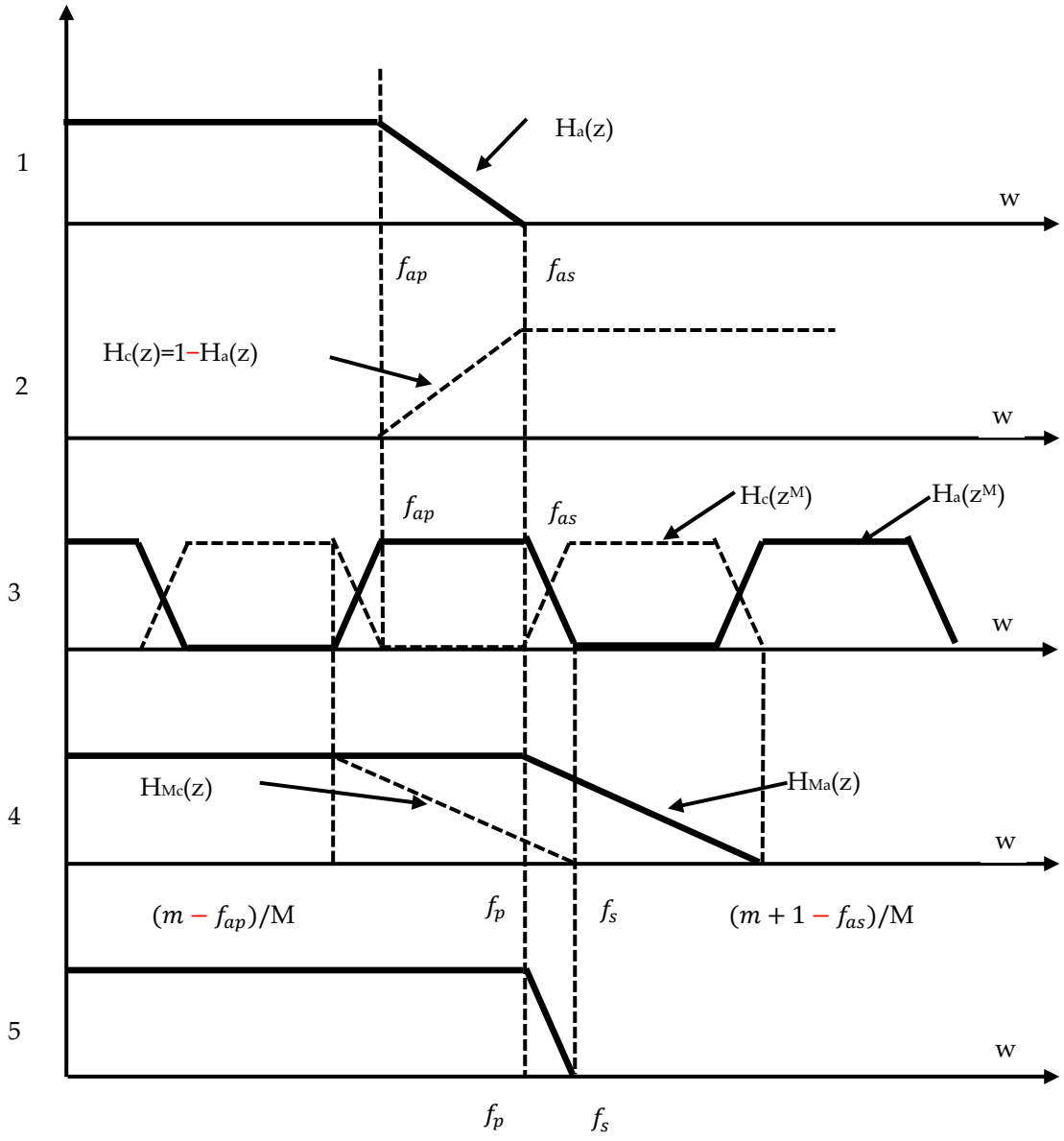


Figure 2. Illustration of the FRM approach's frequency response.

3. Proposed Channel Filter

Using the idea of the resource sharing principle, a time-division multiplexing (TDM)-based filter framework was propounded in this paper that uses an individual multiply and add section regardless of the number of channels as well as taps. The frequency with which the filter operates was increased and the resource sharing principle was implemented.

3.1. Efficient FIR Filter Architecture with a Single Channel

The primary components that make up the FIR filter's structure are delay blocks and multipliers, which take the shape of MAC structures. The speed of the multiplier, which establishes the critical path within the filter arrangement, is the only factor affecting the FIR filter's performance.

Let the input is X(n) and output is Y(n).

An N-tap FIR filter's general form can be expressed as

$$Y(n) = \sum_{k=0}^{N-1} h_k X(n-k) \quad (4)$$

where h_k denotes the filter impulse response's kth coefficient.

Multipliers are the primary source of the FIR filter's complexity. There are multipliers in proportion to the number of taps. These challenges are addressed by an effective architecture that increases the FIR filter's output sampling rate and uses a single multiplier and adder to operate the entire filter. To finish the filter's operation, N clock cycles are needed for the N-tap FIR filter. Let us say the rate at which samples come from the final filter is raised to four Mega Samples per Second (MSPS) for a FIR filter with four taps, with the rate of input sampling set at 1 MSPS. Data are fed into the multiplier to perform filter operations for each clock cycle via delay blocks, and the clock cycle needed to obtain the output is four. Figure 3 depicts a single-channel MAC-based FIR filter architecture. To choose the data from all of the registers and carry out the multiplier operation, utilize the multiplier.

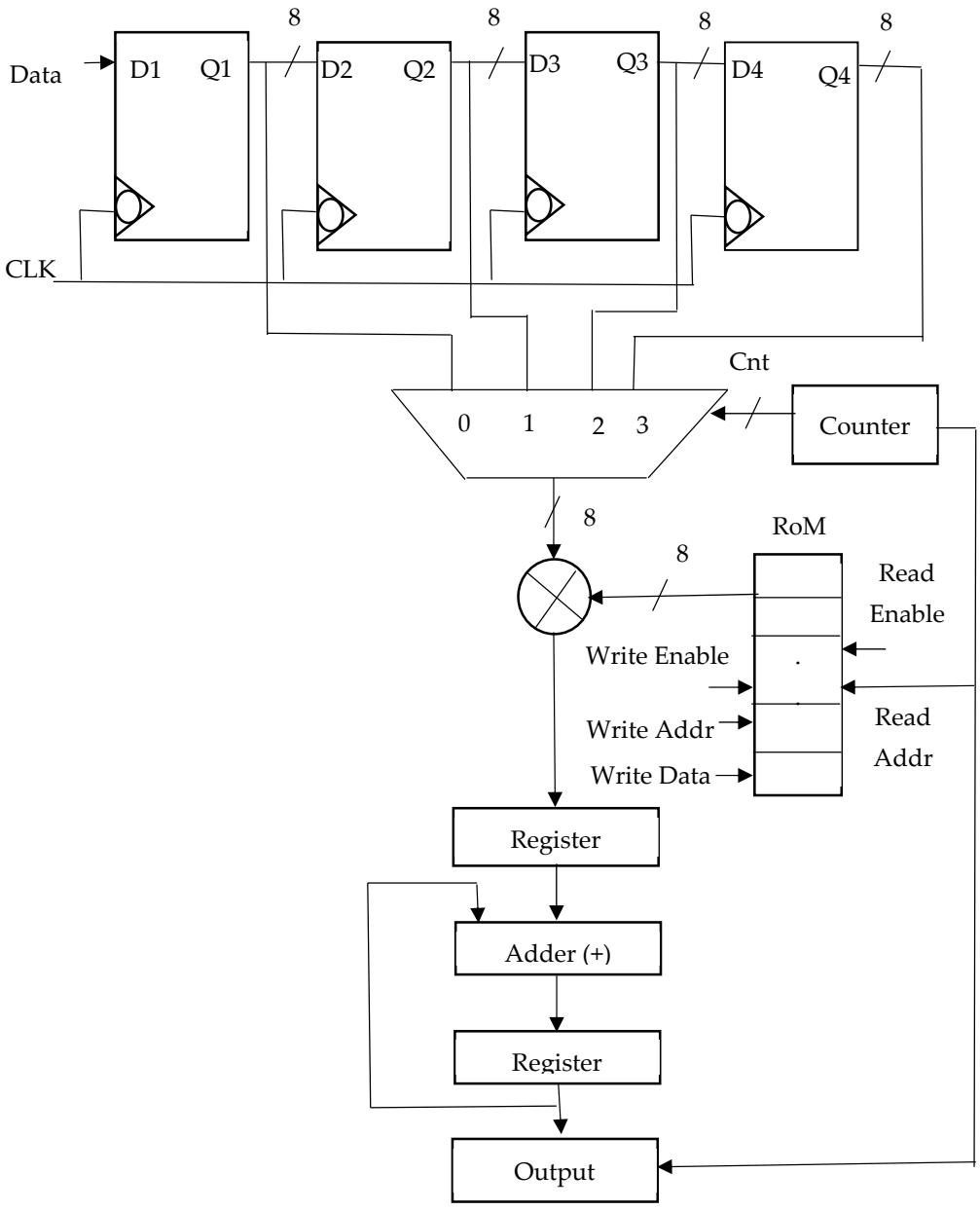


Figure 3. Single-channel MAC-based FIR filter.

The coefficients are saved in a 256 × 8 RoM (four bytes are used for the 4-tap filter), which is intended to minimize switching activity as opposed to storing the coefficients in registers. After four clock cycles, the multiplier outputs are accumulated in an accumulator block, which is then reset to zero. Only one generic counter selects the accumulator operation, coefficient memory address array, and multiplexer select lines. Comparably, a single multiplier and adder can be used to implement N taps by inserting registers, which raises the FIR filter's operating frequency. For this reason, an 8-tap FIR filter uses 8 MSPS with a sampling rate of 1 MSPS. To obtain the filter output of a FIR filter with N-tap, N clock cycles are required.

Figure 4 illustrates the FRM technique using the FIR filter that is based on a single channel MAC. The output of the FRM technique using a single channel MAC was produced by replacing the four sub-filters that are present in the FRM method, $H_a(z)$ which is a linear-phase low-pass filter, interpolated filter $H_a(z^M)$ for filter $H_a(z)$ which is low pass, masking filter $H_{Ma}(z)$ for interpolated filter $H_a(z^M)$ and masking filter $H_{Mc}(z)$ for complementary interpolated filter $H_c(z^M)$ with a single-channel MAC-based FIR filter. The complement filter, $H_c(z)$ is achieved by using $H_a(z)$ which uses a single-channel MAC-based FIR filter.

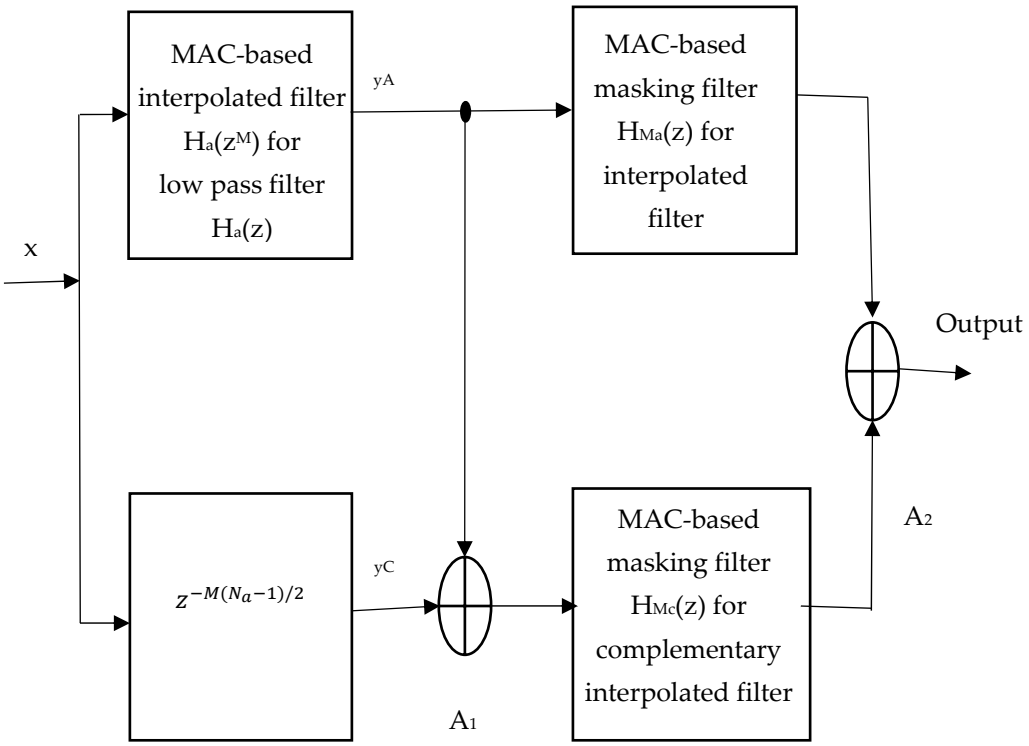


Figure 4. Design of the FIR filter using single MAC-based FRM technique.

3.2. Efficient FIR Filter Architecture with Multi-Channel

The expressions (3) can be used to demonstrate the channel filter's architectural reconfigurability. Although the proposed architecture can operate in multiple modes, a dual-mode channelizer was taken into consideration for simplicity of explanation. Assume that the channel filter operates in two modes of communication, first, with the passband and stopband frequencies of f_{p1} and f_{s1}, and second, with f_{p2} and f_{s2}.

Applying the similar subfilters shown in Figure 1 of the two modes will yield reconfigurability. The filters used for masking can be rearranged by switching the values in the Lookup Table (LUT), which is going to be covered subsequently. The variables f_{ap} and f_{as} are the same for the two standards. Consequently, we have

$$f_{p1}M_1 - |f_{p1}M_1| = f_{p2}M_2 - |f_{p2}M_2| \quad (5)$$

$$f_{s1}M_1 - |f_{s1}M_1| = f_{s2}M_2 - |f_{s2}M_2| \quad (6)$$

where M_1 is the interpolation factor for the first communication mode and M_2 is the interpolation factor for the second communication mode, which can be found by working through Equations (5) and (6).

Therefore, the same modal filter can be reconfigured to meet both standards by varying the delay components. By selecting the right number of delay units, the double mode channelizer can be expanded to include more modes. You can find the up-sampling factor (M) or the number of delays by changing the filter specification in expressions (5) and (6) to match the third standard.

For M = 2, 3, 4, and 5, the representation of channelizer provides a clear explanation of the architectural reconfigurability. Let $f_{ap} = 0.2$ and $f_{as} = 0.25$ be the modal filter's specifications. With M = 2 (two delays for each modal filter delay), we obtain $0.2 = 2f_p - [2f_p]$ from expression (5). We obtain $0.25 = 2f_s - [2f_s]$ from expression (6). We obtain $f_p = 0.1$ and $f_s = 0.125$ by solving the two expressions, and we will use y_1 to represent this channel. Here, $f_p < f_{ap}$ and $f_s < f_{as}$ can be obtained by the combination of masking filters $H_{Ma1}(z)$ with transition-band parameters of fmap = 0.1 and $f_{mas} = 0.375$, which are wider, as found in Equation (3) with the modal filter. This will give us the value of f_p is 0.1 and f_s is 0.125 which are the specification for the filter. The output is obtained from the adder's output or the masking filter's output, $H_{Ma1}(z)$, using the 2:1 multiplexer (MUX) In Figure 1. In this instance, $H_{Ma1}(z)$'s output yields the output y_1. The masking filter has very little complexity because of the wider nature of the transition band. The expression (7) proposed by Bellanger [15] can be used to determine the length of each filter, N, in Figure 1.

$$N = -\frac{2\log_{10}(10\delta_p\delta_s)}{3(f_s - f_p)} - 1 \quad (7)$$

where $(f_s - f_p)$ is the width of transition, which is normalized to 1, the peak stopband ripple is δ_p, and the peak passband ripple is δ_s. For each of the subfilters in Figure 1, let $\delta_p = 0.1$ dB and $\delta_s = -40$ dB be fixed. We can determine $N_{HMa1(z)}$, which the length of the filter used for masking $H_{Ma1}(z)$, as 6, and the modal filter length for all cases as $N_{modal} = 39$ by accordingly changing the values in (7). The expression (8) [4] provides the number of complementary delays.

$$N_{delays} = \frac{((N_{modal} - 1)M)}{2} \quad (8)$$

Take note that obtaining y_1 does not necessitate complementary delays. Following is the general process to obtain the values of $N_{HMa(z)}$, $N_{HMc(z)}$, and N_{delays}, along with the specifications of $H_{Ma}(z)$ and $H_{Mc}(z)$.

(Step 1) To obtain the specifications f_p and f_s, solve Equations (5) and (6) by replacing 0.2 as the values of f_{ap} and 0.25 as the value of f_{as} and M, which is the upsample factor.

(Step 2) Substitute the specifications of f_p and f_s from the first step to find the specifications of $H_{Ma}(z)$ and $H_{Mc}(z)$ utilizing (3).

(Step 3) $N_{HMa(z)}$, $N_{HMc(z)}$ values can be obtained by applying (7).

(Step 4) Utilizing (8), find the value of N_{delays}.

(Step 5) Using the 2:1 MUX and the design shown in Figure 1, extract the filter output with f_p and f_s specifications from the adder A_2.

The subfilter specifications for various values of M, calculated using the above procedure, are displayed in Table 1. For each of M, the modal filter's particularizations are $f_{ap} = 0.2$, $f_{as} = 0.25$, and $N_{modal} = 39$. As a result, it is clear that the modal filter is always fixed as well and the only variable is M, which is the quantity of delays. It is feasible to obtain filter specifications of $f_p = 0.1$ and $f_s = 0.125$, $f_p = 0.24$ and $f_s = 0.25$, $f_p = 0.3$ and $f_s = 0.312$, and $f_p = 0.4$ and $f_s = 0.415$ by using proper filters for masking and using the design in Figure 1. At the resulting filter, it is feasible to reduce the transition band by selecting a modal filter with the transition band much sharper. For instance, with 2 as the value of M, the final filter specification is $f_p = 0.1$ along with $f_s = 0.11$ if we select a

modal filter with f_{ap} = 0.2 and f_{as} = 0.22. It goes without saying that for extremely narrow transition-band specifications, the modal filter's complexity will rise.

Table 1. Subfilters specification of Figure 1 for various M.

M	H(z)		$H_{Ma}(z)$			$H_{Mc}(z)$			
	f_p	f_s	f_{map}	f_{mas}	$N_{HMa(z)}$	f_{mcp}	f_{mcs}	$N_{HMc(z)}$	N_{delays}
2	0.1	0.125	0.1	0.375	6	NA	NA	NA	NA
3	0.4	0.415	0.4	0.584	12	0.267	0.415	12	57
4	0.3	0.312	0.3	0.438	16	0.2	0.312	16	76
5	0.24	0.25	0.24	0.35	21	0.16	0.25	21	95

The multi-channel filter's fundamental framework is identical to that of the FRM network in Figure 1. In the case of y_1, the output is obtained from the filter used for masking $H_{Ma1}(z)$ using 2:1 MUX, and the outputs y_2, y_3, and y_4 were obtained from adder A_2 using 2:1 MUX. The output y will be selected from the values y_1, y_2 y_3, and y_4 to the input given at the mode selector in Figure 1. Figure 5 depicts the design of the modal filter and Figure 6 depicts the Structure of complementary delays. (Computed in Section 3.2) The selected length (N_{modal}) is 39 for the modal filter. The mode selector (MS) sets the number of delays (NOD). For instance, the MS is set to "00," "01," "10," and "11," respectively, to obtain the output channels y_1, y_2, y_3, and y_4. As a result, the NOD is modified from M = 2 to 5, accordingly. As seen in Figure 7, the complementary outputs y_{c2}, y_{c3}, and y_{c4} are obtained using the same scheme. The number of delays will be chosen as "0" for y_{c1} (since y_1 does not require complementary delays), "57" for y_{c2}, "76" for y_{c3}, and "95" for y_{c4} [as discussed in Section 3.2, and obtained from (8)] to mode selector.

The architecture of [11] is utilized in the implementation of the filters used for masking $H_{ma}(z)$ and $H_{mc}(z)$. The required set of coefficients is chosen for the respective channel after the four set coefficients required for the filters, which correspond to the four channels from y_1 to y_4, which are stored in the Lookup Table. The architecture also allows for customization by allowing the number of masking filter taps to be varied based on the mode selector. As mentioned in Section 3.2, the mask filter only requires 6 taps for M = 2, but 12 taps are needed for M = 3, and so on.

Dynamically, the number of taps varies based on the mode selector. To obtain y_4, for instance, a 21-tap masking filter is needed. All that is required if the intended channel turns into y_1 is a filter for masking with a number of taps of 6. In this instance, the dynamically changed quantity of taps is 6. Because the unused taps are unloaded, this filter length reconfiguration reduces the dynamic power consumption. Therefore, in order to acquire y_1, the MS sets its value to "00" such that the value of M is 2. Since y_1 is acquired by modal filter mask output y_{A1} with a filter $H_{ma1}(z)$, no complementary delays are needed. As a result, the LUT will be changed to coefficients that match $H_{ma1}(z)$. The output y_1 straight from the output of $H_{ma1}(z)$ can be extracted from 2:1 multiplex unit, as illustrated in Figure 1. The MS is set to "01" to obtain y_2, and N_{delays} = 57 is the number of complementary delays, meaning that M = 3. $H_{ma2}(z)$ and $H_{mc2}(z)$ coefficients are loaded into the LUT corresponding to $H_{ma}(z)$ and $H_{mc}(z)$, respectively. You can use the same method to obtain the outputs y_3 and y_4. This means that by adjusting the MS and Lookup Table data of $H_{ma}(z)$ and $H_{mc}(z)$, the similar filter model can offer four sets of channel particulars: f_p = 0.1 and f_s = 0.125, f_p = 0.24 and f_s = 0.25, f_p = 0.3 and f_s = 0.312, and f_p = 0.4 and f_s = 0.415. Since the architecture of the coefficient multiplier is similar to the one in [11], filters for any additional channel with different specifications can be implemented by simply altering the values of the coefficient in the Lookup Table to match the required filter details. Thus, it can be deduced that, as illustrated in Figure 1, various filter details, such as y_1 to y_4, can be acquired at the output by varying the value of the mode selector.

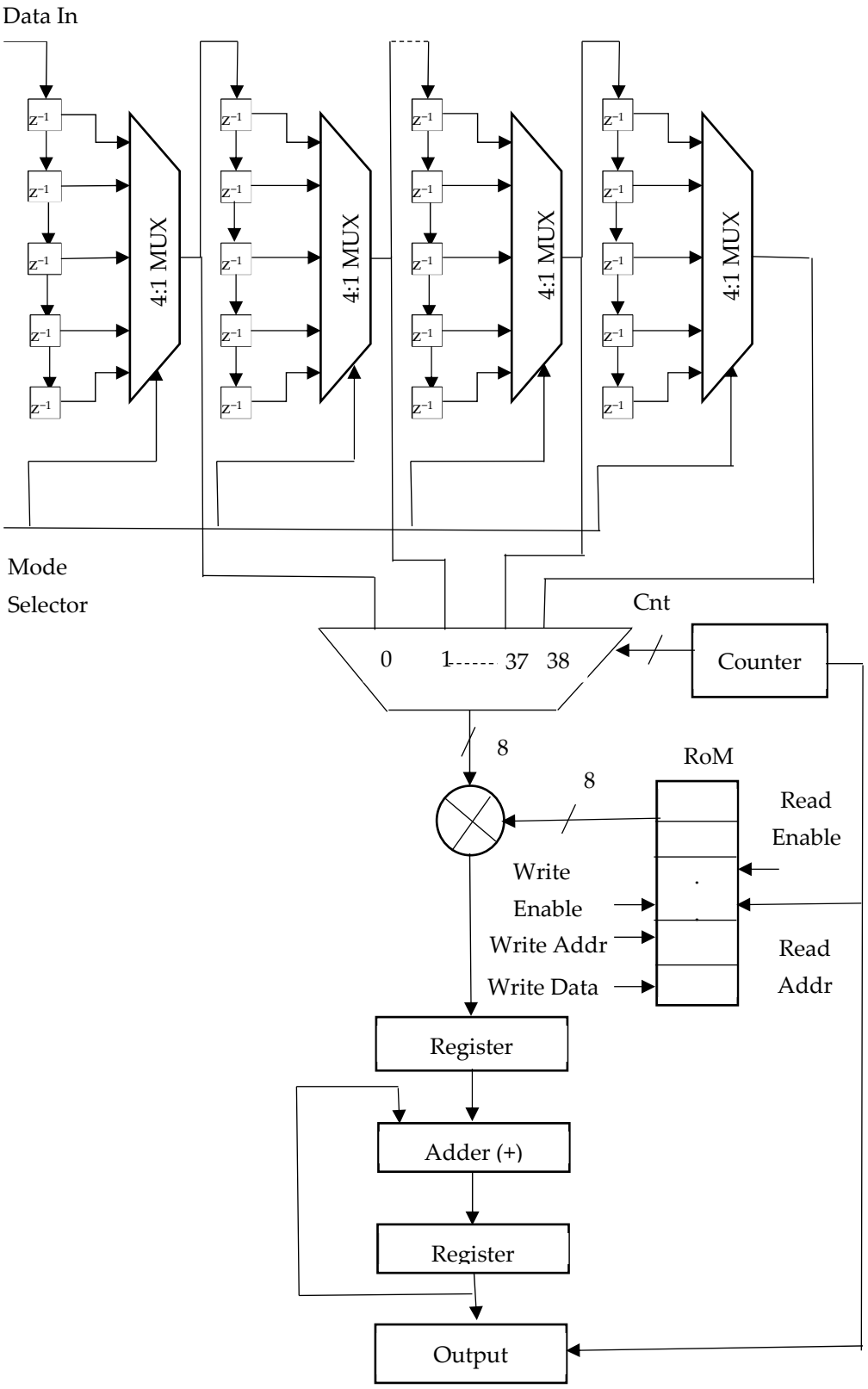


Figure 5. Multi-channel MAC-based FIR filter.

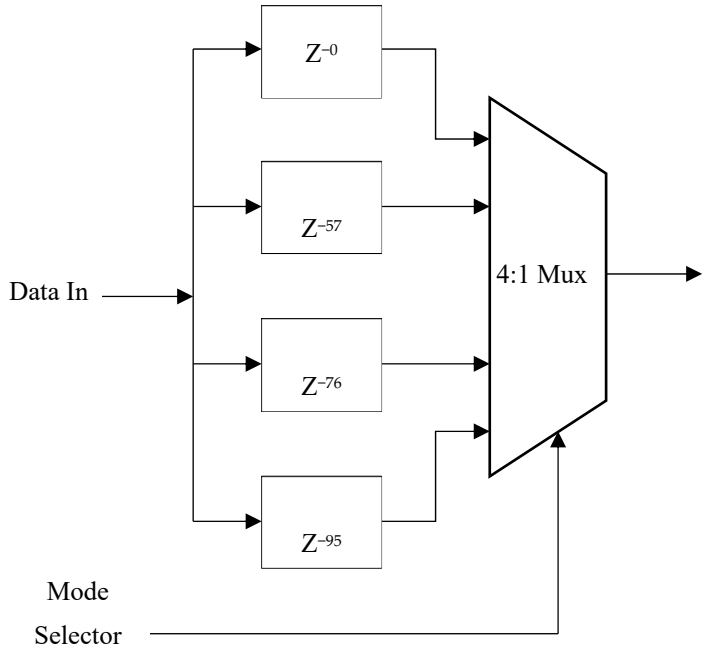


Figure 6. Structure of complementary delays.

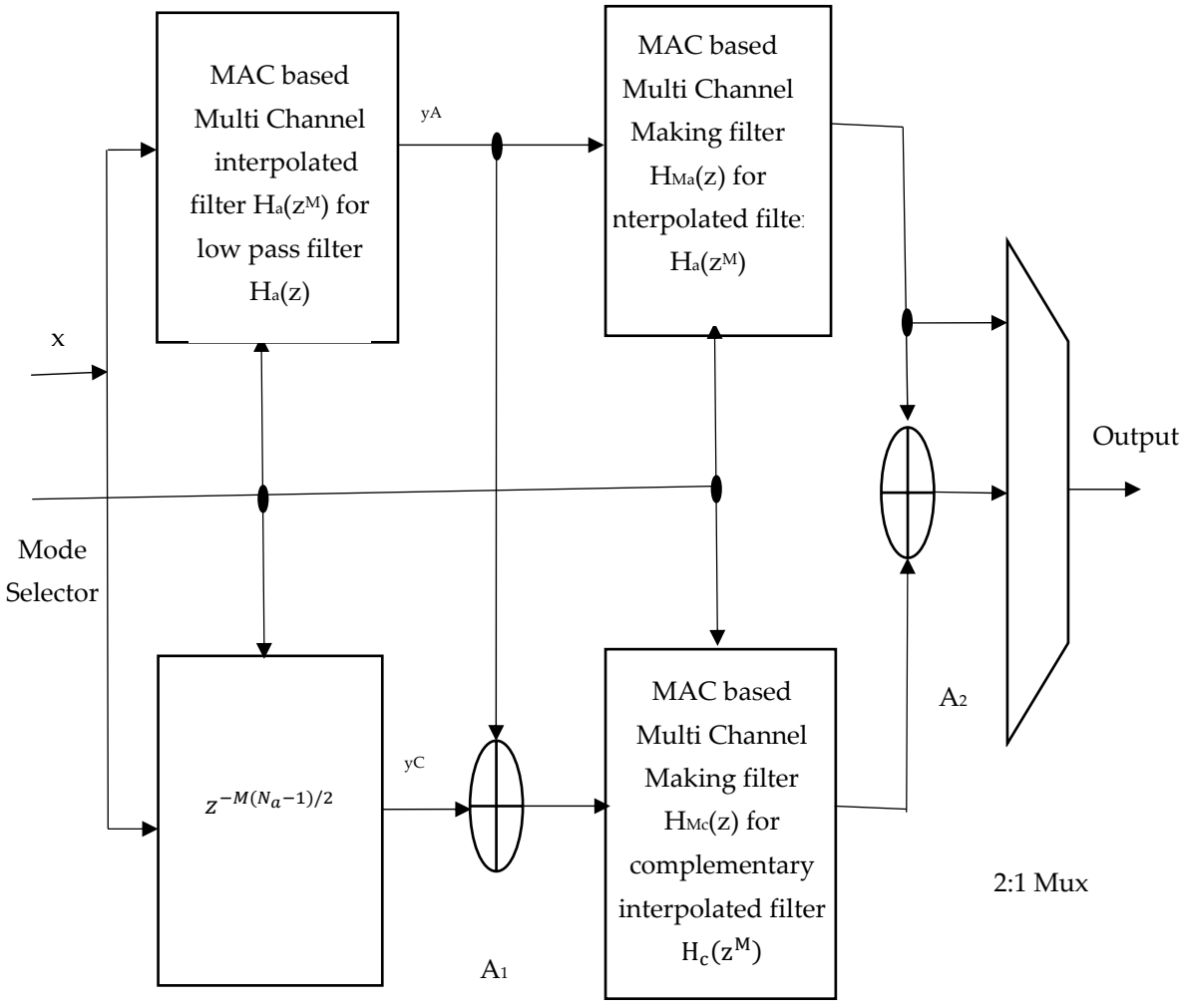


Figure 7. Proposed Design of FIR filter using Multi-Channel MAC-based FRM technique.

The expressions (2) provide an example of the channel filter's architectural reconfigurability. Note that the suggested design can operate in several modes; however, for simplicity, a dual-mode channelizer was taken into consideration. It is necessary to consider the channel filters' passband and stopband frequencies, which correspond to the two communication standards (or modes) of operation, respectively. Utilizing the identical filters in Figure 1 for a pair of standards will allow for reconfigurability. It is possible to adjust the LUT data to change the masking filters while maintaining the same parameters for both standards (the same modal filter is used for both standards). The implementation of a single mac FIR filter reduces the complexity of subfilters.

Therefore, we can achieve reconfiguration using the identical modal filter to function for two distinct standards by altering the number of delay components. Additional communication modes can be added to the dual-mode channelizer by selecting the right number of delay elements. Sharp transition-band FIR filters created with traditional techniques, however, would need considerably more taps (higher order), which would raise the cost of the hardware. Conversely, the FRM technique-designed sharp transition-band FIR filters have a significantly lower number of taps, which would result in a large reduction in both cost and power usage.

4. Experimental Results

The proposed channel filter architecture's design and synthesis outcomes are shown in this section.

Synthesis Findings

The proposed Multi-Channel MAC FRM architecture synthesis and analysis are presented in this section, along with a performance comparison with the traditional architectures. Topologies for FRM Multi-Channel MAC filters are synthesized with Altera and Xilinx platform. Figure 8 displays the simulation waveform of the FIR filter with multi-channel.

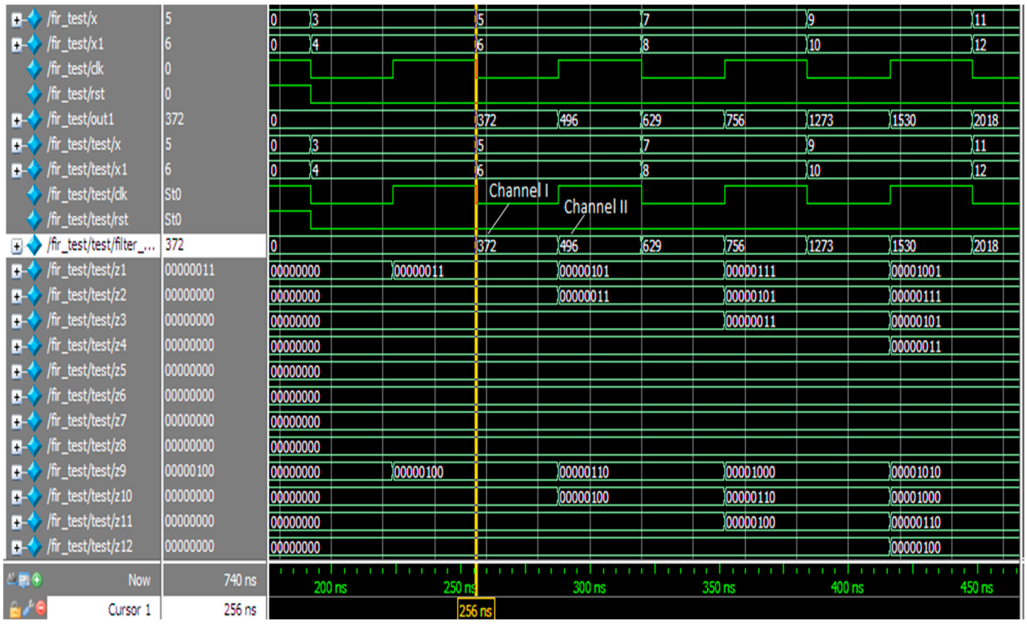


Figure 8. Waveform for simulation of multi-channel MAC FIR filter.

Figure 8 shows the simultaneous fetching of data_in_ch1 and data_in_ch2, with the outputs being obtained at data_out in a sequential manner. Because of the single MAC core architecture, the output (data_out) of channel 1 is available after 8 clocking. Channels 1 and 2 have sampling frequencies of fs and fs/2, respectively. In this case, the sample frequency is 16 times more than the input sample frequency; thus, 16 clocks are needed to obtain the

output of both channels. As a result, channel 1 and channel 2 data are retrieved once every sixteen clocks. After 2080 ns, the output (data_out) is acquired (1500 ns to process the 8-bit input and 580 ns for both channels' latency).

The netlist illustration of the filter with the single channel is displayed in Figure 9. The key components for calculating area in the multiplexer of FIR filter with Multi-Channel are the MAC and RAM. Using the Xilinx Virtex-5 XC5VSX95T-1FF113 FPGA apparatus, the propounded framework is harnessed and synthesized.

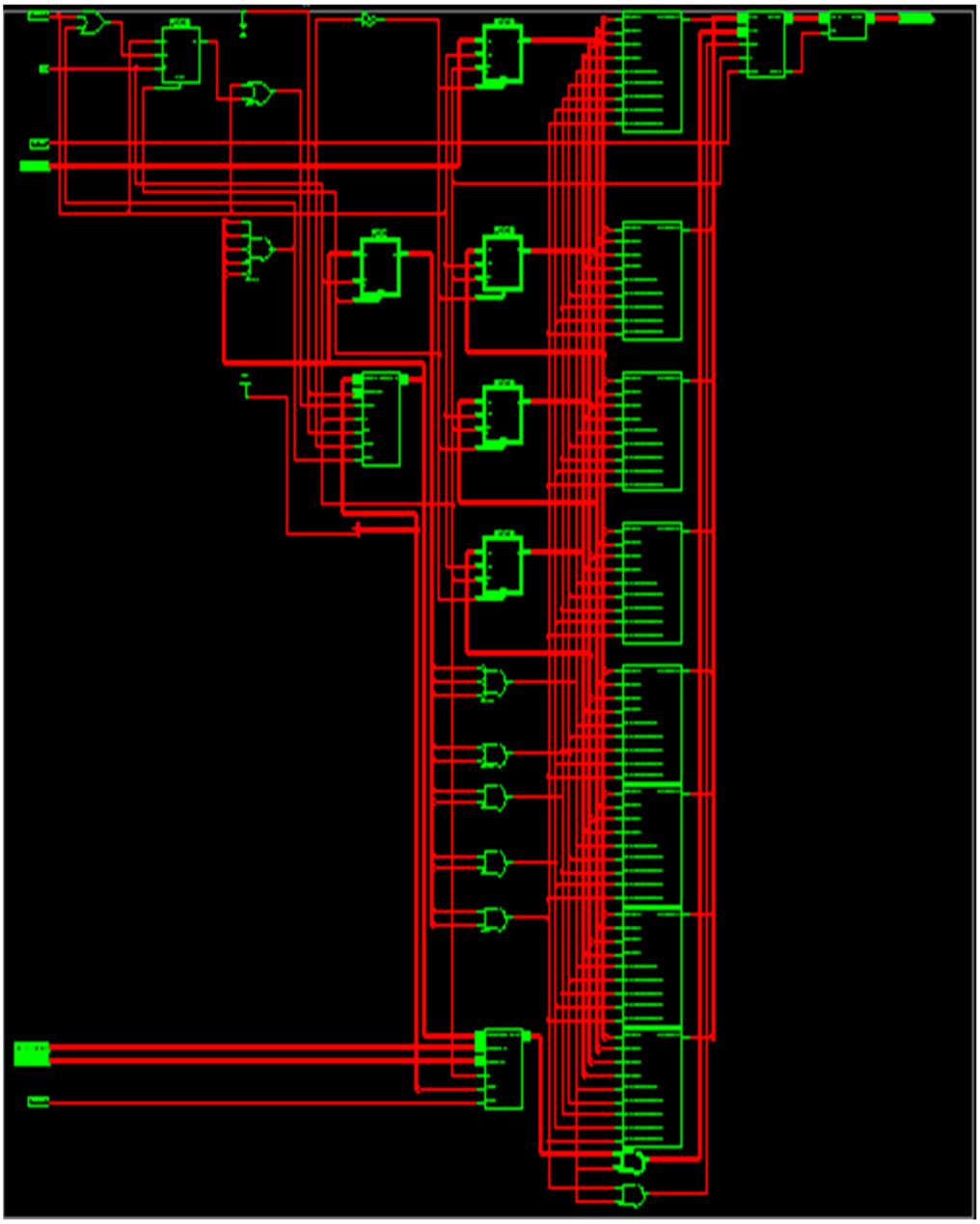


Figure 9. RTL representation of FIR filter with Multi-Channel MAC.

The technology used to synthesize the recommended reconfigurable filter is 0.18-µm CMOS and Table 2 tabulates the data. On comparing with the results produced in the architecture in [5,11,12], the synthesis result showed that the recommended channel filter implemented using FPGA came up with the mean area and power scaling down of 68.38% and 72.76% compared to CSD-CSM [5], 64.34% and 75.13% compared to CSD-PSM [5], 59.38% and 41.63% over BPSM [11], 61.82% and 65.95% over BCSM [11], 21.36% and 14.88%

over RFRM [12], respectively. It is very clear that the result shown by the recommended channel filter provides a compelling curtail of area and power compared to the techniques implemented in [5,11,12].

Table 2. Results for synthesis of reconfigurable channel filter.

Parameters	CSD-CSM [5]	CSD-PSM [5]	BPSM [11]	BCSM [11]	RFRM [12]	Proposed Filter
Taps	350	350	350	350	81	84
Area (mm^2)	5.82	5.16	4.53	4.82	2.34	1.84
Delay (ns)	13.5	15.34	9.87	9.08	5.98	4.95
Power (mW)	105	115	49	84	33.6	28.6

The proposed reconfigurable filter was put into practice and examined using Virtex-II 2v3000ff1152-4 FPGA and Table 3 tabulates the data. On comparing with the outcomes produced in the architecture in [11,12], the results showed that the recommended channel filter implemented using ASIC gives an average area and power scaling down of 84.12% and 61.76% over BPSM [11], 85.42% and 64.89% over BCSM [11], and 5.18% and 9.08% over RFRM [12], respectively. The result produced by the recommended channel filter gave a momentous diminishment of area and power compared to the procedure implemented in [11,12].

Table 3. Results for implementation of the proposed channel filter.

Parameters	BPSM [11]	BCSM [11]	RFRM [12]	Proposed Filter
Gate Count	449,382	489,632	75,248	71,348
Data Arrival Time (ns)	42.824	38.672	15.78	12.18
Power Dissipation (mW)	375.45	408.9	157.89	143.54

The proposed MAC designs and the existing reconfigurable architecture are compared in Table 4. In comparison with the reported architectures, they offer significant performance and space savings through the use of MAC techniques. The reason for the operation's increased speed is due to pipelining.

Table 4. Implementation results of filter structures (Device: Xilinx ZYNQ-XC7Z020-1CLG84C FPGA device).

Parameters	Sudipta Bose [17]	Proposed Single MAC FIR Filter
Number of Slices	1014	218
Delay (ns)	1.81	1.76
Slice Lut	936	842

The proposed Multi-Channel MAC-based FRM design is compared with the existing reconfigurable architecture from in Table 5. When compared to the reported architectures, for channel 1, the proposed design requires a single multiplier and adder to implement the model filter; it also requires one multiplier and adder for the masking filter, while channels 2 to 4 need one multiplier and adder each for the model filter and two multipliers and adders for the masking filter. Additionally, one adder is needed for adder 2 in the architecture. In total, the proposed filter can be designed using 11 multipliers and 12 adders.

Table 5. Hardware complexity comparison of proposed methods.

Parameters	FRM-VBW [19]	FRM-Farrow Structure [19]	Proposed Filter
No. of multipliers	1157	1226	11
No. of adders	2314	2452	12

5. Conclusions

In this paper, we have put forward a reconfiguration oriented FRM for the SR application by implementing Time Division Multiplexing (TDM)-based multi-channel MAC architecture. We used the principle of resource sharing to realize multiple sharp filter responses from a single prototype low-pass FIR filter with linear-phase in each sub-filter level. This paper uses an individual multiply and add module regardless of the number of channels as well as taps. The reconfigurable filter proposed was synthesized on technology that is based on 0.18-µm CMOS, and the trials were put into practice and carried out on Virtex-II 2v3000ff1152-4 field-programmable gate array. The outcomes showed that the proposed channel filter, which was synthesized using FPGA, provides 21.36% of the area and 14.88% of the power scaling down on average and put into practice using ASIC provides 5.18% of the area and 9.08% of power scaling down on average.

Author Contributions: Conceptualization, S.A. and B.P.J.; methodology, S.A.; software, S.A.; validation, M.-F.L.; formal analysis, B.P.J.; resources, M.-F.L.; writing—original draft preparation, S.A.; writing—review and editing, B.P.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mitola, J. *Software Radio Arch Itecture*; Wiley: New York, NY, USA, 2000.
2. Zangi, K.C.; Koilpillai, R.D. Software radio issues in cellular base stations. *IEEE J. Select. Areas Commun.* **1999**, *17*, 561–573. [CrossRef]
3. Hentschel, T. Channelization for software defined base-stations. *Ann. Telecomm.* **2002**, *57*, 386–420. [CrossRef]
4. Lim, Y.C. Frequency-response masking approach for the synthesis of sharp linear phase digital filters. *IEEE Trans. Circuits Syst.* **1986**, *CAS-33*, 357–364.
5. Hartley, R.I. Subexpression sharing in filters using canonic signed digit multipliers. *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.* **1996**, *43*, 677–688. [CrossRef]
6. Peiro, M.M.; Boemo, E.I.; Wanhammar, L. Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm. *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.* **2002**, *49*, 196–203. [CrossRef]
7. Vinod, A.P.; Lai, E.M.-K. On the implementation of efficient channel filters forwideband receivers by optimizing common subexpression elimination methods. *IEEE Trans. Comput. Aided Design. Integr. Circuits Syst.* **2005**, *24*, 295–304. [CrossRef]
8. Mahesh, R.; Vinod, A.P. A new common subexpression elimination algorithm for realizing low complexity higher order digital filters. *IEEE Trans. Comput. Aided Design. Integr. Circuits Syst.* **2008**, *27*, 217–229. [CrossRef]
9. Chen, K.H.; Chiueh, T.D. A low-power digit-based reconfigurable FIR filter. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2006**, *53*, 617–621. [CrossRef]
10. Zhangwen, T.; Zhang, J.; Min, H. A high-speed, programmable, CSD coefficient FIR filter. *IEEE Trans. Consum. Electron.* **2002**, *48*, 834–837. [CrossRef]
11. Mahesh, R.; Vinod, A.P. Reconfigurable low complexity FIR filters for software radio receivers. In Proceedings of the 2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, Helsinki, Finland, 11–14 September 2006; pp. 1–5.
12. Mahesh, R.; Vinod, A.P. Reconfigurable Frequency Response Masking Filters for Software Radio Channelization. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2008**, *55*, 21–25. [CrossRef]
13. Li, H.; Tong, X.; Li, Q. A Low Complexity FIR Filter with an Efficient Frequency Response Masking Technique. In Proceedings of the IEEE 3rd International Conference on Electronics Technology, Chengdu, China, 8–12 May 2020.
14. Alam, S.A.; Gustafsson, O. On the Implementation of Time-Multiplexed Frequency-Response Masking Filters. *IEEE Trans. Signal Process.* **2016**, *64*, 3933–3944. [CrossRef]

15. Bellanger, M.; Evci, C. On computational complexity in adaptive digital filters. In Proceedings of the ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing, Boston, MA, USA, 14–16 April 1983; pp. 45–48.
16. Zhang, L.; Rao, C.; Lou, X. Low-power reconfigurable FIR filter design based on common operation sharing. *IEEE Trans. Circuits Syst II Express Briefs* **2023**, *70*, 3169. [CrossRef]
17. Bose, S.; De, A. Area-Delay-Power Efficient VLSI Architecture of FIR Filter for Processing Seismic Signal. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 3451–3455. [CrossRef]
18. Goldsmith, J.; Crockett, L.H.; Stewart, R.W. A Natively Fixed-Point Run-Time Reconfigurable FIR Filter Design Method for FPGA Hardware. *IEEE Open J. Circuits Syst.* **2022**, *3*, 25–37. [CrossRef]
19. Haridas, N.; Elias, E. Reconfigurable Farrow Structure-Based FRM Filters for Wireless Communication Systems. *Circuits Syst. Signal Process.* **2016**, *36*, 315–338. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.