International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I10P201

Volume 70 Issue 10, 1-7, October 2022
© 2022 Seventh Sense Research Group®

Original Article

FPGA based Matched Filter Design using Modified Masking Signal Generator

Arunjyothi Eddla[1], Venkata Yasoda Jayasree Pappu[2]

[1, 2]*Department of ECE, GITAM Institute of Technology, Hyderabad, India.*

[1]*Corresponding Author : aeddla@gitam.edu*

Received: 05 July 2022 Revised: 25 September 2022 Accepted: 29 September 2022 Published: 19 October 2022

Abstract *- Nowadays, digital filters are broadly used in various signal and image processing applications due to their efficacy in filtering processes. However, the implementation of digital filters is suffered from various issues, such as high area and power consumption. An effective filter must be implemented to minimise the area while minimizing power consumption to overcome this. In this paper, the Modified Masking Signal Generator (MMSG) is proposed for designing the Matched Filter (MF) over Field-Programmable Gate Array (FPGA). The proposed MMSG uses only a smaller number of resources during the filtering processes, which helps to decrease the overall hardware resources. The performance of the MF-MMSG architecture is analyzed using slices, slice registers; Slice Look Up Tables (SLUTs), logical elements, flip flops, bonded Input/Output Block (IOB), power, delay and operating frequency. The existing research, namely Two-Dimensional MF (TDMF) and Matched Filtering Unit (MFU) with Synthetic Aperture Radar (SAR), are used to evaluate the MF-MMSG architecture. The SLUTs of the MF-MMSG designed in Zynq Ultrascale+ FPGA is 4120, less than the MFU-SAR.*

Keywords *- Digital Filters, Field-Programmable Gate Array, Matched Filter, Modified Masking Signal Generator, Hardware Utilization, Power Consumption.*

1. Introduction

The general operation of the filter is to remove the unwanted part of the signal or reduce the noise. Also, it is used to extract the relevant portions of that respective signal for further processes. Generally, filters are categorized as analog and digital. The merits of digital filters are higher efficiency, compact size and faster reconfiguration. These digital filters are highly accurate compared to analogical circuits [1] [2]. The digital filter output is the convolution between the distinct signal amplitude at input and impulse response in the time domain [3]. Digital signal filtering is used in different applications such as quality control in production, video surveillance systems, medicine, edge detection, smoothing, image enhancement, geolocation and so on [4] [5]. This research considers the matched filter for filtering applications because it is used to increase the output signal-noise ratio [21]. The output signal generally correlates with a filter to identify the environment's target existence [7] [8]. Very Large-Scale Integration (VLSI) is used for many standard digital implementations, and it mainly concentrates on the optimization of electronic circuits [9] [10].

The FPGA platforms are also used to design filters because they offer advanced analog resources, a quick time to market, flexibility, quick floating-point operations, enough embedded resources, and faster computing [11] [12]. This FPGA has interconnection blocks, I/O and reconfigurable logic, but it differs from Digital Signal Processing (DSP) processors and microcontrollers [13]. In the design of VLSI, optimization is performed from the device to the system level. However, there is some acceptable imprecision while using the optimization approaches at the system level. Subsequently, the area, power consumption and system performance must be updated because of the imprecision [14]. The large components cause performance degradation in VLSI architecture which causes the error in the filter design in the floating-point arithmetic step [15]. To overcome this, an effective filter design is developed with less hardware resources to improve the FPGA performances.

The main contribution of this research is given as follows:

- The MMSG-based MF is designed to improve filtering applications. The developed MMSG reduces the number of logical elements in the filter design, which helps minimise the MF's overall hardware resources.
- In the MF-MMSG architecture, fewer hardware resources are used to minimize power consumption and improve the operating frequency.

The remaining paper is arranged as follows: An existing FPGA-based filter design is given in Section 2. Clear details

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

about the MF-MMSG architecture are provided in Section 3. The outcomes of the MF-MMSG architecture are provided in Section 4, whereas the conclusion is presented in Section 5.

2. Related Work

By detecting an object's position in the correlation domain, the matched filtering technique typically uses a provided object with a known position as a template to find the position of a second object. Popular techniques for locating an object in the middle of noise and distortions include the classical matched filter and its modification. Due to their effectiveness in filtering operations, digital filters are widely used in various signal and image processing applications. However, several problems occur during the implementation, including high area and power requirements. As a result, a brief discussion about the previous works and their various filtering techniques is reviewed in detail below.

Wang, D et al. [16] developed the Block-matching and 3D filtering (BM3D) denoising approach due to the enhanced image processing quality. Here, a design hardware accelerator was developed to improve the BM3D with less power utilization. The fine-grained data-level parallelism of BM3D was used by developing the dedicated systolic-like array for accomplishing the parallel block-matching, which was used to save a huge amount of hardware resources. The Processing Element (PE) array consumed a huge number of resources of DSP due to the distance computation in the filter.

Magesh, and Duraipandian [17] implemented the higher-order matched Finite Impulse Response (FIR) filter by using the odd and even phases. The number of adders and multipliers required to design the filter was minimized then the conventional multiple constant multiplication FIR filters by using the odd and even phases of the FIR filter. Further, this higher-order matched FIR filter decreased the required area in the design. The developed higher-order matched FIR filter consumed higher resources than the conventional FIR due to adding extra components.

Abeer Chang et al. [18] developed the single-bit ternary matched filter in the DSP application. Here, the ternary quantizer is used to make the filter coefficients ternary $\{1,-1,0\}$. The second-order sigma-delta modulator was used to select the ternary coefficients. Further, the return waveform from the filter was similar to the input; hence it is defined that the single-bit ternary matched filter performed well in the filtering process. However, this work did not discuss the FPGA resources used by the matched filter.

Xiang et al. [19] developed the Two-Dimensional MF (TDMF) to enhance the rendering quality and auxiliary equipment’s speed in vein imaging systems. The developed TDMF was utilized to provide real-time response and was built into the handheld portable device. The key operation of the matched filter was the convolution among the image window’s pixels and convolution kernel template. This TDMF was efficient because of its simple hardware operation. Moreover, the increment in the convolution kernel window increased the usage of FPGA internal resources.

Choi Y et al. [20] designed real-time Synthetic Aperture Radar (SAR) imaging by using the Range-Doppler Algorithm (RDA). The developed RDA comprised Range Cell Migration Correction (RCMC), azimuth, and range compression. Here, real-time processing was needed in the RCMC Processing Unit (RPU) and Matched Filtering Unit (MFU). The RPU and mixed-radix multi-path delay commutator Fast Fourier Transform (FFT) was used in the MFU. MFU used decimation-in-frequency and time to minimize the memory requirements. But, the usage of more hardware resources affected the operating frequency.

3. MF-MMSG Architecture

Till now, there have been several problems with digital filters' implementation, including their high area and power requirements. An efficient filter that reduces the area while decreasing power usage must be used to get around this. Therefore, the Matched Filter (MF) over Field-Programmable Gate Array is proposed in this study using the Modified Masking Signal Generator (MMSG) (FPGA). The suggested MMSG requires fewer resources for the filtering operations, which reduces the total amount of hardware resources used. This MF-MMSG architecture is mainly proposed to develop a low-area architecture while achieving effective filtering performances. Since matched filter [19] is a template-matching algorithm, it increases the ratio of signal power to the average noise power in the output. The main enhancement accomplished in the control unit is optimizing the structure of the conventional Masking Signal Generator, which helps to decrease the number of hardware elements used in the overall architecture. Hence, the MF-MMSG architecture achieves low area, whereas this filter's power consumption and operating frequency also improved.

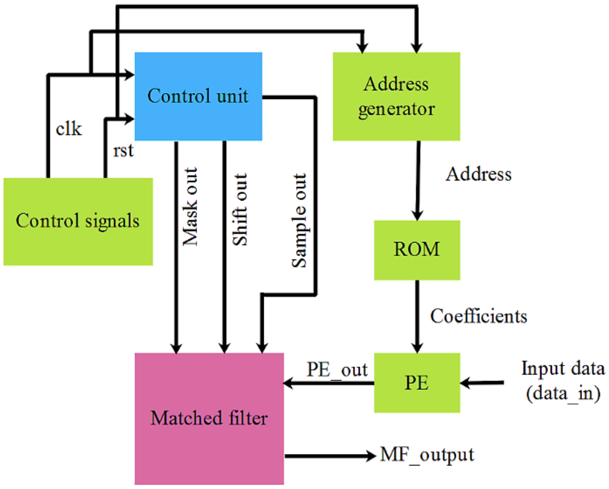


Fig. 1 Block diagram of the proposed architecture

The overall process of the proposed architecture is mentioned as follows:

- At first, the control signals of the clock $(clk)$and reset(rst) are given as input to the control unit. Since the control unit is designed with the MMSG, which returns three different outputs such as $Maskout, Shiftout$ and $Sampleout$.
- The address is generated using the address generator, and the coefficients are stored in ROM. According to the address, the stored coefficient is fetched from the ROM, and the fetched coefficient is given as input to the PE.
- On the other hand, the randomly generated input data $(data_in)$is also given as input to the PE, which is multiplied by the coefficients. These multiplied $data\_in$ values $(PE_out)$are given as input to the MF.
- Further, the output of MF, i.e. MF_output is computed by adding the PE_out with $Maskout, Shiftout\ and\ Sampleout.$

3.1. Control Unit

In MF-MMSG architecture, the latch clock $LCK_i$is provided by the control unit and masking signal from MMSG. The block diagram of the control unit is shown in Figure 2. The system clock only executes the clock pulse counter. In the Delay Line (DL), the counter's each bit and the delayed signal (other than LSB and MSB) are less frequent since the LSB is used as postponing clock at the delay unit.

Additionally, MSB is taken as an outcome to a receiver as LSK_0 it comprises the frequency f_c/L . The MSB given to the DL unit is then postponed in the sequence by the range of $1/N.f_c$. The tapping on the DL is used to obtain the remaining latch clocks LCK_i .

Fig. 3 shows the DL's slice for MF-MMSG architecture. Here, the hatched part's operating frequency is less and equation (1) shows the average clock frequency.

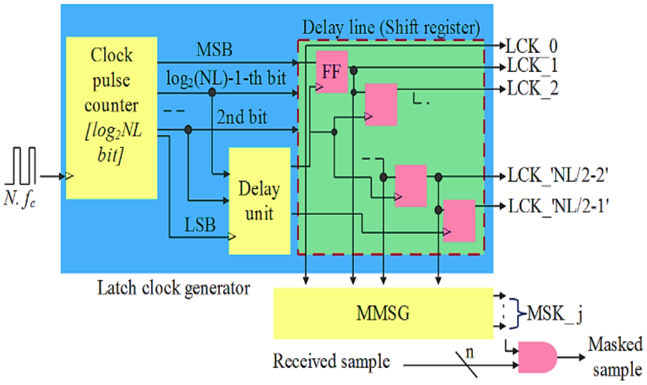
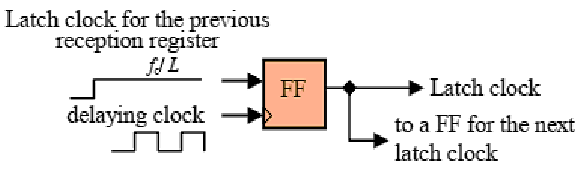


Fig. 2 Control unit architecture


Latch clock for the previous reception register
fc/ L
delaying clock
FF
Latch clock
to a FF for the next latch clock

a) Proposed

Fig. 3 Slice of delay line

$$Average\ clock\ frequency = N.f_c/8.\sum_{n=0}^{log_2 N.\ L-3}(1/4)^n \approx N.f_c/6 \quad (1)$$

Fig. 3 shows that the control unit requires less receiver registers, reducing the latch clock. Thus, the switching activity minimization is also used to decrease the power usage in the control unit. Moreover, data masking is proposed for generating the MSK_jbased on the signals of LSK_i which is detailed in the following section.

3.2. Data Masking using MMSG

The proposed MMSG block minimises the number of resources used in the overall filter architecture. Figure 4 illustrates the architecture of MMSG. The input load of FFs is increased when the collected signal bus is linked with the $N.\ L$ receiver registers. The bus's predictable switching frequency is high, and power distribution is minimized when the input load of the AND gates is less. Further, the designed MMSG generates the masking signals according to the LCK_i .

The steps processed in the MMSG are given as follows:

1. An 8-bit value is taken from 8 FFs and given as input to the MMSG-based data masking operation where the given 8-bit signals are divided into four 2-bit pairs $\{a(0,1), a(2,3), a(4,5), a(6,7)\}$.

2. The divided four 2-bit pairs are given as input to the four different logic gates: AND, OR, XOR, and XNOR, respectively. The operations processed in the logic gates are expressed in the following equations (2)-(5).

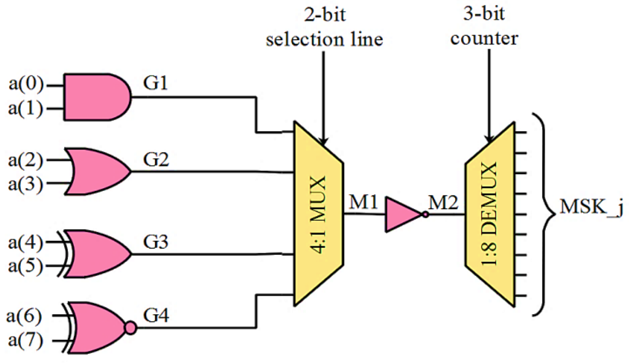


Fig. 4 Architecture of MMSG

Table 1. The output of 4:1 MUX

2-bit selection line	Output from logic gate
00	AND
01	OR
10	XOR
11	XNOR

$$G1 = a(0).a(1) \qquad (2)$$

$$G2 = a(2) + a(3) \qquad (3)$$

$$G3 = a(4) \oplus a(5) \qquad (4)$$

$$G4 = \overline{a(6) \oplus a(7)} \qquad (5)$$

Where $G1, G2, G3andG4$ are the outputs of AND, OR, XOR and XNOR logic gates, respectively.

3. The values $G1, G2, G3andG4$are input to the 4:1 MUX, which returns 1-bit output ($M1$) according to the 2-bit selection line. The following Table 1 shows the output of 4:1 MUX.

4. The outcome from the 4:1 MUX is again given to the NOT gate, as shown in equation (6).

$$M2 = \overline{M1} \qquad (6)$$

Where $M2$is the output from the NOT gate.

5. Further, the output from the NOT gate $M2$ is given as input to the 1:8 DEMUX, which provides the output MSK_jaccording to the 3-bit counter. The MSK_j is further processed with the received sample n to generate the $Sampleout$. Moreover, the LCK is the $Shiftout$ and MSK is the $Maskout$.

On the other hand, the is PE_outobtained by multiplying the $data_in$with the coefficients. Subsequently, the values $PE_{out}, Maskout, Shiftout\ and\ Sampleout$ are added together to generate an output of MF (MF_output), as shown in equation (7).

$$MF_{output} = PE\_out + Maskout + Shiftout + Sampleout \qquad (7)$$

4. Results and Discussion

The outcomes of the MF-MMSG architecture are provided in this section. The enhanced matched filter with the MMSG has been implemented in Xilinx ISE 14.2 software. The simulations of the MF-MMSG are accomplished based on Verilog HDL and ModelSim. The randomly generated input is utilized for analyzing the functions of the proposed MF architecture. Table 2 provides the specification parameters of the MF-MMSG architecture.

Table 2. Specifications parameters

Parameter	Value
Pulse width	10ns
Clock period	20ns
Clock frequency	50MHz
Duty cycle	50%

Table 3. Synthesis results for MF-MMSG with Cyclone IV FPGA

FPGA performances	Total Amount Of resources	Used resources	% of consumption
Slices	6100	10	1%
Slice registers	66235	5	1%
SLUTs	76028	5217	7%
Logical elements	76028	5302	7%
Flip Flops	66235	5	1%
Bonded IOB	300	12	4%

Table 4. Synthesis results for MF-MMSG with Zynq Ultrascale+ FPGA

FPGA performances	The total amount of resources	Used resources	% of consumption
Slices	5500	11	1%
Slice registers	72000	357	1%
SLUTs	72000	4120	5%
Logical elements	72000	150	1%
Flip Flops	72000	360	1%
Bonded IOB	200	6	3%

Table 5. Results of delay, power and operating frequency

FPGA devices	Delay (ns)	Power (W)	Operating frequency (MHz)
Cyclone IV	2.79	0.037	247.019
Zynq Ultrascale+	1.02	0.014	527.173

4.1. Performance analysis of MF-MMSG Architecture

The MF-MMSG is examined using various FPGA devices such as Cyclone IV EP4CE115F29C8 FPGA and Zynq Ultrascale+ FPGA. Here, the resource utilization of MF-MMSG architecture over the FPGA is analyzed using slices, slice registers, SLUTs, logical elements, flip flops and bonded IOB. The delay, power and operating frequency are also analysed for this MF-MMSG architecture.

Tables 3 and 4 provide the results of MF-MMSG designed in the Cyclone IV and Zynq Ultrascale+ devices, respectively. These results show the number of hardware resources used by the MF-MMSG in FPGA devices. The maximum amount of resources used by the MF-MMSG is 7% for Cyclone IV and 5% for Zynq Ultrascale+ FPGA devices. Further, the evaluation of delay, power, and

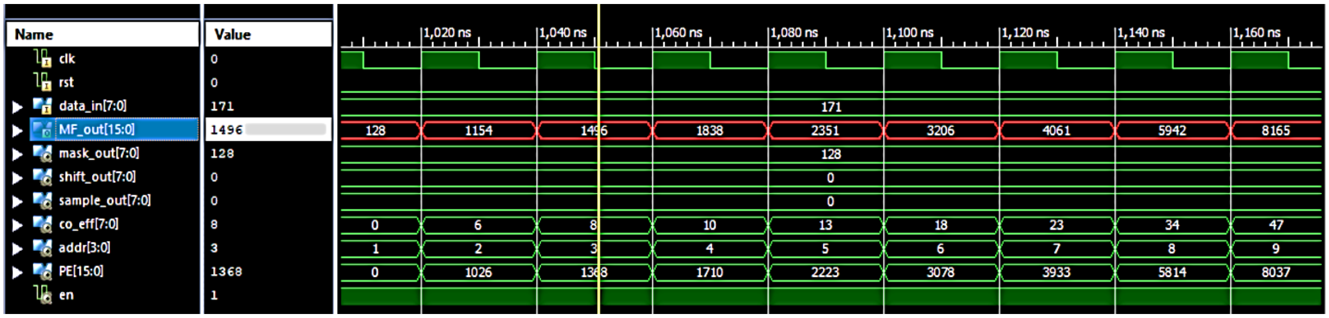


Fig. 5 Simulation waveform of MF-MMSG architecture

operating frequency are given in Table 5. The MF-MMSG designed in the Zynq Ultrascale+ FPGA achieves an operating frequency of 527.173 MHz, whereas the MF-MMSG with Cyclone IV achieves 247.019 MHz.

The overall simulation waveform of the MF-MMSG architecture is shown in Figure 5. The $clk, rst and en$ control signals are given to the control unit and address generator of MF-MMSG, whereas the address $(addr)$ stored in the ROM is 3. The 8-bit input data $(data\_in)$ 171 and 8-bit coefficient $(co_eff)$of 8 are given as input PE. In PE, the $data_in$and co_eff are multiplied together for generating 16-bit output$PE$of 1368. On the other hand, the control unit generates three 8-bit values, such as $Maskout$, $Shiftout$ and $Sampleout$. The values of $Maskout, Shiftout$ and $Maskout$, $Shiftout$ are equal to $128, 0\ and\ 0$, respectively.

Further, the outputs from PE and control unit are given as input to the MF. In MF, all the four values, such as $PE, Maskout, Shiftout$ and, $Sampleout$ are added together to generate the 16-bit output $MF_out$of 1496. Therefore, the given simulation waveform shows the effective performance of MF-MMSG architecture.

4.2. Comparative analysis

The comparative analysis of the MF-MMSG architecture is shown in this section. The existing FPGA-based filter designs, such as BM3D [16], TDMF [19] and MFU-SAR [20], are used to evaluate the MF-MMSG. The FPGA devices such as Cyclone IV and Zynq Ultrascale+ are considered for comparison purposes.

Table 6. Comparative analysis of MF-MMSG for Cyclone IV

Performances	BM3D [16]	TDMF [19]	MF-MMSG
Logical elements	-	50939	5302
Operating Frequency (MHz)	233	50	247.019

Table 7. Comparative analysis of MF-MMSG for Zynq Ultrascale+ FPGA

Performances	MFU-SAR [20]	MF-MMSG
SLUTs	51,542	4120
Operating Frequency (MHz)	300	527.173

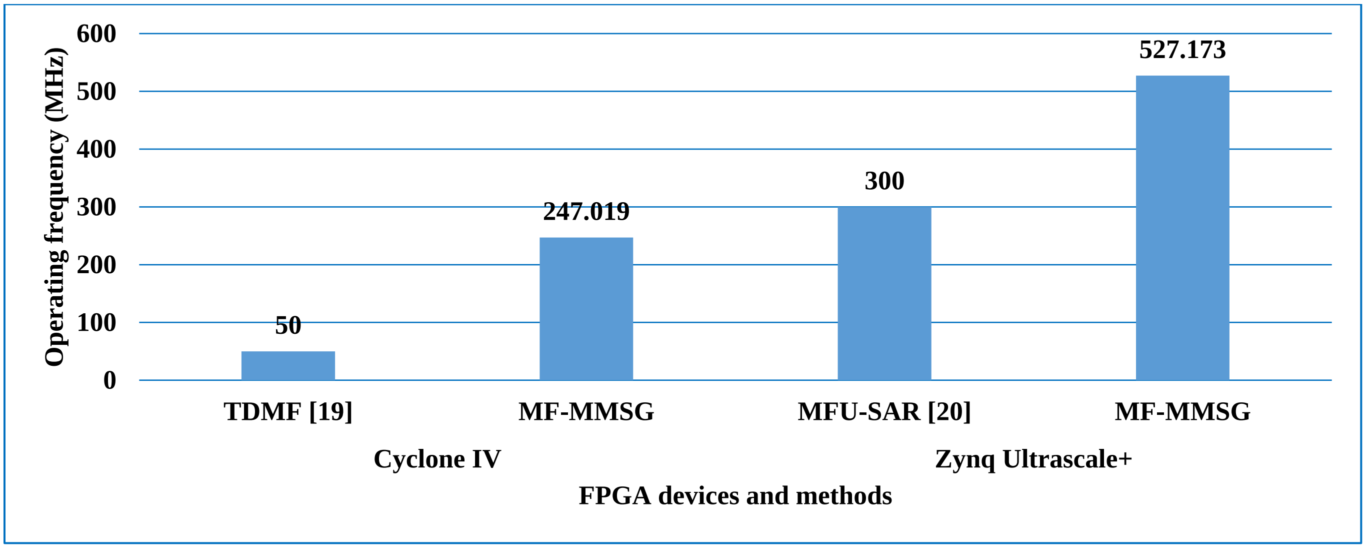


Fig. 6 Comparison of operating frequency

Tables 5 and 6 show the comparative analysis of the MF-MMSG for Cyclone IV and Zynq Ultrascale+ FPGA devices, respectively. Moreover, the graphical comparison of the operating frequency for MF-MMSG is shown in Figure 6. From the analysis, it is known that the proposed MF-MMSG achieves less hardware utilization and high operating frequency than the BM3D [16], TDMF [19] and MFU-SAR [20]. For example, the SLUTs for the MF-MMSG designed in Zynq Ultrascale+ FPGA is 4120, less than the MFU-SAR [20].

4.3. Discussion

The hardware resources of the MF-MMSG are reduced by lowering the number of logical components in the MMSG and by lowering the number of receiver registers needed in the control unit. The traditional Masking Signal Generator's structure was optimized as part of the control unit's major improvement, reducing the overall architecture's reliance on hardware components. As a result, the MF-MMSG architecture provides minimal area while improving this filter's power usage and operating frequency. Overall assessment demonstrates that the suggested MF-MMSG outperforms the existing BM3D [16], TDMF [19], and MFU-SAR [20] in terms of hardware utilization and operating frequency. When evaluated with the MFU-SAR, the SLUTs of the MF-MMSG developed on Zynq Ultrascale+ FPGA are 4120.

5. Conclusion

Digital filters are widely used to remove or extract certain features from the image/ signal. Since the filter design with large-size modules increases the overall size of the filters. This paper proposes the MF-MMSG architecture for designing the filter with fewer hardware resources. The developed MF increases the output signal-noise ratio during the filtering process. Specifically, the design of MMSG is used to reduce the number of hardware elements used in the overall MF-MMSG, leading to reduced power consumption. The results evaluation shows that the MF-MMSG outperforms well than the TDMF and MFU-SAR. The SLUTs of the MF-MMSG designed in Zynq Ultrascale+ FPGA is 4120, less than the MFU-SAR. In the future, an optimized infinite impulse response filter can be developed to improve performance.

References

[1] D. Datta, and H.S. Dutta, “High-Performance IIR Filter Implementation on FPGA,” *Journal of Electrical Systems and Information Technology,* vol. 8, pp. 1-9, 2021.

[2] S. Zahoor, and S. Naseem, “Design and Implementation of an Efficient Fir Digital Filter,” *Cogent Engineering,* vol. 4, pp. 1323373, 2017.

[3] S. Yadav, R. Yadav, A. Kumar, and M. Kumar, “A Novel Approach for the Optimal Design of Digital Fir Filter Using Grasshopper Optimization Algorithm,” *ISA Transactions,* vol. 108, pp. 196-206, 2021.

[4] M. Valueva, P. Lyakhov, G. Valuev, and N. Nagornov, “Digital Filter Architecture with Calculations in the Residue Number System By Winograd Method F (2× 2, 2× 2),” *IEEE Access,* vol. 9, pp. 143331-143340, 2021.

[5] A.K. Joginipelly, and D. Charalampidis, “An Efficient Circuit for Error Reduction in Logarithmic Multiplication for Filtering Applications,” *International Journal of Circuit Theory and Applications,* vol. 48, pp. 809-815, 2020.

[6] Ramya R, Madhura R, "Fpga Implementation of Optimized Bist Architecture for Testing of Logic Circuits," *SSRG International Journal of Vlsi & Signal Processing,* vol. 7, no. 2, pp. 36-42, 2020. Crossref, https://doi.org/10.14445/23942584/IJVSP-V7I2P106.

[7] M.M. Pishrow, and J. Abouei, “Joint Design of the Discrete Phase Transmit Sequence and Receive Filter In Radar Systems,” *IET Radar, Sonar & Navigation,* vol. 16, pp. 315-326, 2022.

[8] D. Koukounis, C. Ttofis, A. Papadopoulos, and T. Theocharides, “A High-Performance Hardware Architecture for Portable, Low-Power Retinal Vessel Segmentation,” *Integration,* vol. 47, pp. 377-386, 2014.

[9] S. Jadhav, B. Pooja, C. Asawari, and C. Namrata, “FPGA Based ECG Signal Noise Suppression Using Windowing Techniques,” *International Journal of Engineering Trends and Technology (IJETT),* vol. 47, no. 9, pp. 505-508, 2017. Crossref, https://doi.org/10.14445/22315381/IJETT-V47P283.

[10] S. Dagar, and G. Nijhawan, “Area Efficient Moving Object Detection Using Spatial and Temporal Method in FPGA,” *International Journal of Engineering Trends and Technology,* vol. 70, no. 9, pp. 138-147, 2022. Crossref, https://doi.org/10.14445/22315381/IJETT-V70I9P214.

[11] A. Kaur, and R. Mehra, “Da Algorithm Based Reconfigurable 16-Tap Fir Filter Design Analysis,” *International Journal of Engineering Trends and Technology (IJETT),* vol. 60, no. 1, 2018. Crossref, https://doi.org/10.14445/22315381/IJETT-V60P210.

[12] M. Maamoun, A. Hassani, S. Dahmani, H. Ait Saadi, G. Zerari, N. Chabini, and R. Beguenane, “Efficient FPGA Based Architecture for High Order Fir Filtering Using Simultaneous DSP and LUT Reduced Utilization,” *IET Circuits, Devices & Systems,* vol. 15, pp. 475-484, 2021.

[13] M.A. Kumar, and K.M. Chari, “Efficient Fpga-Based VLSI Architecture for Detecting R-Peaks in Electrocardiogram Signal by Combining Shannon Energy with Hilbert Transform,” *IET Signal Processing,* vol. 12, pp. 748-755, 2018.

[14] V.K. Odugu, "An Efficient Vlsi Architecture of 2-D Finite Impulse Response Filter using Enhanced Approximate Compressor Circuits," *International Journal of Circuit Theory and Applications*, vol. 49, pp. 3653-3668, 2021.

[15] T.M. John, and S. Chacko, "FPGA-Based Implementation of Floating-Point Processing Element for the Design of Efficient Fir Filters," *IET Computers & Digital Techniques*, vol. 15, pp. 296-301, 2021.

[16] D. Wang, J. Xu, and K. Xu, "An Fpga-Based Hardware Accelerator for Real-Time Block-Matching and 3d Filtering," *IEEE Access*, vol. 8, pp. 121987-121998, 2020.

[17] V. Magesh, and N. Duraipandian. "Design of Higher Order Matched Fir Filter using Odd and Even Phase Process," *Intelligent Automation and Soft Computing*, vol. 31, pp. 1499-1510, 2022.

[18] C. Abeer, T. Din Memon, Zahir, M. Hussain, I. Hussain Kalwar, and B. Shankar Chowdhry. "Design and Analysis of Single-Bit Ternary Matched Filter," *Springer Science+Business Media*, LLC, Part of Springer Nature, 2018. https://doi.org/10.1007/S11277-018-5729-Y.

[19] W. Xiang, D. Li, D. Sun, J. Liu, G. Zhou, Y. Gao, and X. Cui, "Fpga-Based Two-Dimensional Matched Filter Design for Vein Imaging Systems," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 9, pp. 1-10, 2021.

[20] Y. Choi, D. Jeong, M. Lee, W. Lee, and Y. Jung, "FPGA Implementation of the Range-Doppler Algorithm for Real-Time Synthetic Aperture Radar Imaging," *Electronics*, vol. 10, pp. 2133, 2021.

[21] W. Xiang, J. Sun, D. Li, J. Liu, G. Zhou, and X. Cui, "A FPGA Vein Imaging System Based on Matched Filter," *In 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, pp. 274-278, 2021.