

Article

Advanced, Real-Time Programmable FPGA-Based Digital Filtering Unit for IR Detection Modules

Krzysztof Achtenberg ^{1,*}, Ryszard Szplet ² and Zbigniew Bielecki ¹¹ Institute of Optoelectronics, Military University of Technology, 2 Kaliskiego Str., 00-908 Warsaw, Poland² Faculty of Electronics, Military University of Technology, 2 Kaliskiego Str., 00-908 Warsaw, Poland

* Correspondence: krzysztof.achtenberg@wat.edu.pl

Abstract: This paper presents a programmable digital filtering unit dedicated to operating with signals from infrared (IR) detection modules. The designed device is quite useful for increasing the signal-to-noise ratio due to the reduction in noise and interference from detector–amplifier circuits or external radiation sources. Moreover, the developed device is flexible due to the possibility of programming the desired filter types and their responses. In the circuit, an advanced field-programmable gate array FPGA chip was used to ensure an adequate number of resources that are necessary to implement an effective filtration process. The proposed circuitry was assisted by a 32-bit microcontroller to perform controlling functions and could operate at frequency sampling of up to 40 MSa/s with 16-bit resolution. In addition, in our application, the sampling frequency decimation enabled obtaining relatively narrow passband characteristics also in the low frequency range. The filtered signal was available in real time at the digital-to-analog converter output. In the paper, we showed results of simulations and real measurements of filters implementation in the FPGA device. Moreover, we also presented a practical application of the proposed circuit in cooperation with an InAsSb mid-IR detector module, where its self-noise was effectively reduced. The presented device can be regarded as an attractive alternative to the lock-in technique, artificial intelligence algorithms, or wavelet transform in applications where their use is impossible or problematic. Comparing the presented device with the previous proposal, a higher signal-to-noise ratio improvement and wider bandwidth of operation were obtained.



Citation: Achtenberg, K.; Szplet, R.; Bielecki, Z. Advanced, Real-Time Programmable FPGA-Based Digital Filtering Unit for IR Detection Modules. *Electronics* **2024**, *13*, 4449. <https://doi.org/10.3390/electronics13224449>

Academic Editor: Alexander Barkalov

Received: 8 October 2024

Revised: 8 November 2024

Accepted: 11 November 2024

Published: 13 November 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/>).

Keywords: digital filter; FPGA; finite impulse response; FIR; infinite impulse response; IIR; IR detector; digital signal processing; DSP; IR detection module

1. Introduction

Field-programmable gate array (FPGA) devices have become a popular and effective platform for digital signal processing (DSP) implementations [1,2]. These specific and flexible architectures of integrated circuit (IC) resources can be used in a wide range of applications like image processing, neural network systems, measurement instrumentation, or fast acquisitions [3–5]. Furthermore, last year’s effective use of FPGAs in quantum systems and nuclear physics was also observed [6]. The high popularity of this type of IC is determined primarily by the possibility of simple adjustment of the desired function via proper configuration. The prepared hardware description language (HDL) code determines the setting of internal connections between individual logic elements [7]. In this way, building complete, advanced digital systems is achievable. Using FPGAs, it is also possible to obtain a substitute for the application-specific integrated circuit (ASIC) only by preparing the appropriate HDL code. This allows for lower prototyping or production costs than a typical ASIC [8]. The main limitation of such a method is that FPGA ICs are typically not equipped with advanced analog blocks, and external electronic elements are necessary for specific applications. In some advanced FPGA devices, built-in simple analog-to-digital converters (ADCs) can be found [9]. In the literature, we can find various functionalities

implemented using FPGAs: filtering, precision time interval measurement, arithmetic or cryptographic co-processor, fast transceivers or receivers, and many others [10–13]. This is mainly because FPGA systems are more efficient than typical processors. In many cases, they can perform complex mathematical operations in a one clock cycle. Authors in [10] presented the application of FPGA-based filtering to reduce the influence of noise and interferences in space satellite systems. In [13], the digital filter was designed to denoise electroencephalography signals from unwanted components.

Infrared (IR) photodetectors are devices used to convert infrared radiation energy into an electrical signal. Typically, a distinction is made between thermal and photonic groups depending on the physical phenomenon used (heating of a photosensitive element or generation of photocurrent due to the specific interaction of photons with electrons) [14]. The vast majority of IR detectors use semiconductor materials as a photosensitive element. For this reason, there has been significant development in their materials' engineering technology. Appropriate selection of material, doping, and architecture enables the construction of detectors with various parameters, including the operating spectral range [15].

Meaningful progress in recent decades has led to the development of detectors operating from short-wave IR (SWIR) to very-long-wave IR (VLWIR). The most popular semiconductor materials used for their production include mercury cadmium telluride (HgCdTe, MCT), indium arsenide (InAs), indium antimonide (InSb), indium arsenide antimonide (InAsSb), indium gallium arsenide (InGaAs), and also silicon (Si) [16–18]. In contrast to these materials, graphene is a promising optoelectronic material for ultra-broadband photodetectors due to its gapless band structure [19].

Directly using the signal from an IR detector is problematic due to its typical small value. Due to this fact, a specially designed amplifier is usually necessary to convert it into higher measurable amplitude values. The most popular of them is the transimpedance amplifier (TIA), but in some cases, voltage amplifiers (VAs) are also used [20]. The specific model of the amplifier depends on the type of its operation: photovoltaic or photoconductive mode. Several parameters can characterize each detector. The most important of them are normalized detectivity (D^*), current or voltage responsivity (R_i/R_v), noise equivalent power (NEP), and time constant (τ). These parameters depend on the temperature and bias voltage applied to the detector. The main limitation of IR detector operability is the noise generation phenomenon because of its direct influence on D^* and NEP values. The IR detector noise power spectrum density (PSD) usually rises along with the bias and temperature, so the cooling improves its parameters [21]. However, in order to obtain appropriate frequency parameters for the system related to the short τ , they require the use of an appropriate bias and the adequate type of amplifier, which unfortunately leads to an increase in noise in the output signal. This applies primarily to low-frequency noise of the $1/f$ type [22]. For this reason, certain procedures are usually observed to reduce the impact of noise on the parameters of the target system. One of them is the lock-in technique or appropriate filtration that can be performed in analog or digital processing traces [23,24].

IR detectors have found various effective applications in industry, consumers, and the military [25–27]. They can be found in temperature monitoring, communication, medical, security, and imaging systems [28–31]. In all applications, noise led to some limitations and parameters deteriorating.

DSP is one of the most effective techniques of signal manipulation. Appropriate algorithms enable the implementation of several functions, like converting, transforming, filtering, compression, and many others [32,33]. Some functions can also be provided in both time and frequency domains. In many cases, e.g., regarding filtration, the efficiency of the DSP system significantly exceeds the analog solutions, due to more flexibility in parameter adjusting. DSP systems can be based on processors (CPU), microcontrollers (μ C), system-on-chip (SoC), dedicated digital signal processors, or FPGAs [34–36]. Sometimes, application-specific ICs (ASICs) are equipped with special modules dedicated to implementing DSP functions, e.g., Fast Fourier Transform (FFT) [37].

This work combined the few above-described devices and techniques to construct a specialized unit that can improve the signal-to-noise ratio (SNR). The high-performance digital filter was implemented in an advanced FPGA IC using HDL to reject unwanted noises and external radiation signals from the output of the IR detection module.

The presented unit can be programmed to obtain the necessary form (low-pass, band-pass, etc.) and characteristics of the filter (e.g., frequency or phase response). Thanks to the large number of resources in the FPGA that can be used to construct it, an advanced, effective filtering unit capable of suppressing unwanted signals at a level exceeding several dozen dB can be obtained.

In the paper, we presented the hardware construction of the circuit, a result of the simulations of the example filters, and its practical implementation. The well-known techniques and parts were combined to construct an advanced FPGA-based system for real-time measurements, where high-order filters can be implemented. Moreover, we also conducted a practical experiment with the InAsSb mid-wave IR (MWIR) detection module, where noise is quite problematic when operating with weak signals. The obtained performances are significantly better than previous work [38]. The advantage of real-time operation can be especially useful in systems where quick reaction is necessary.

2. Hardware and Methods

2.1. Hardware Platform

Each DSP system is composed of an appropriate combination of hardware and software. Operating with analog signals usually requires the application of data converters, like analog-to-digital converters and/or digital-to-analog converters (DACs). The selection of their specific properties depends strictly on the target application. Analog front-ends are also often used in signal paths to prepare input–output signals by adapting them to converters (ADCs, DACs). The generalized hardware block diagram of a typical DSP system (that can be used for filtering) containing an ADC and DAC is presented in Figure 1.

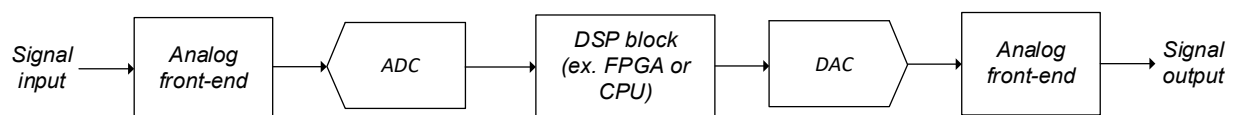


Figure 1. Generalized hardware block diagram of typical DSP system with ADC and DAC that can be used to implement digital filter.

The analog input signal is applied to the analog front-end. This block is usually based on properly selected operational amplifiers, which adjust the signal to the ADC input. It can be amplification, attenuation, filtering, or a change to a differential signal to drive ADC inputs properly. The sampled and digitized data are sent to the DSP block, where the adequate function is implemented. A processed digital signal can be read from this block or converted to the analog domain via DAC. At its output, an analog front-end can also be applied. It can perform similar functions, like in the case of the ADC front-end.

Figure 2 presents the block diagram of the digital filtering unit we developed in our work. The less important elements were intentionally omitted in this figure.

The input signal was connected to the anti-aliasing low-pass filter (LPF) to reject frequencies above half of the sampling frequency. This filter was needed to ensure it operated by the Nyquist sampling theorem and avoided the aliasing phenomenon [39]. In our circuit, we used the two stages of the Sallen–Key configuration with elements calculated to obtain Bessel-type characteristics [40]. The filter was characterized by a flat response to about 1 MHz with attenuation less than 0.1 dB, whereas its cut-off (−3 dB) frequency equaled 4.3 MHz. The −10 dB signal attenuation occurred at a frequency of 8.1 MHz. The oscillations observed in step response were reduced thanks to the Bessel character.

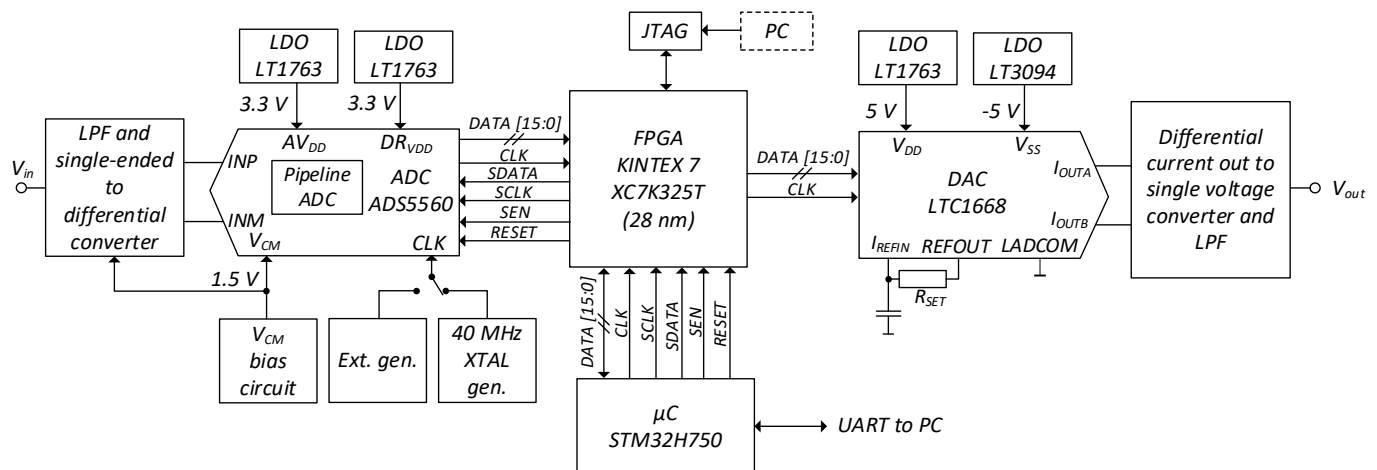


Figure 2. The block diagram of the hardware platform for the digital filtering unit.

The 16-bit CMOS parallel interface was used to exchange data with the FPGA. ADC can be clocked with a constant frequency of 40 MHz from a crystal generator or an externally connected generator using a dedicated SMA connector. According to the datasheet, this ADC had a high SNR of 84 dBFS, adjustable fine gain, internal voltage reference, and low-frequency suppression mode [41]. It was enclosed in a 48-VQFN surface-mount package. The internal registers of ADS5560 could be controlled via serial interface lines (RESET, SEN, SCLK, SDATA) using μ C (FPGA was working only as a simple signal bridge). They were responsible for the gain, output data interface, format, and sampling frequency range. In the case of a sampling frequency ≤ 25 MSa/s, the specific register bit needed be set. Both the ADC and differential amplifiers on the front end operated with a common-mode voltage of $V_{CM} = 1.5$ V. The main DSP operations were performed on the Xilinx series-7 KINTEX XC7K325T FPGA device [42]. The main device properties regarding the available resources are listed in Table 1. In this table, the properties of Xilinx ARTIX-7 XC7A35T used in similar previous work [38] were also included for comparison.

Table 1. Resources of XC7K325T and XC7A35T.

Resource Name	XC7K325T	XC7A35T Used in [38]
Logic Cells	326,080	33,280
Slices	50,950	5200
DSP Slices	840	90
Memory	16,020	1800
Max. I/O Pins	500	250

XC7K325T was characterized by about ten times more logic resources and functional blocks than XC7A35T. Regarding DSP-based filtering operations, the number of DSP slices and logic cells defined the final possibilities. This is because most operations were based on multiplication, summation, or accumulation. In the FPGA, typically specialized DSP slices (highlighted in Table 1) were dedicated to this operation, i.e., DSP48 in the KINTEX-7 family of FPGAs.

The digitally processed signal was sent to the DAC via a parallel interface. In our circuit, we used a 16-bit LTC1668 DAC. Its differential current outputs could be updated at a maximum 50 MSa/s rate. It was characterized by high spectral purity (87 dB SFDR at $f_{out} = 1$ MHz), a low 5 pV-s glitch impulse, and 20 ns of settling time. The DAC chip was enclosed in a 28-pin SSOP package. It could operate with TTL/CMOS 3.3 V or 5 V input signals [43]. Output differential current signals were converted into single-ended using an I-V converter based on a high-speed LT1819 op-amp. Then, the single-ended voltage signal was filtered using the same filter type as the one used for the ADC input. The role of the

DAC output filter was to reject clock and the above Nyquist frequency signal components. It also reduced the effect of zero-order keeping (visible in the time domain as steps) and was sometimes named a “restoration” filter [44].

The FPGA and ADC main features were managed using the STMicroelectronics STM32H750 microcontroller. It could also be further used to extract filtered data from FPGA internal memory registers. A universal asynchronous receiver–transceiver interface was applied to establish a connection with the PC. For practical realization, the overall unit was divided into three printed circuit boards (PCBs) boards, namely ADC/DAC, FPGA, and μ C. The supply and ground planes in the ADC/DAC 4-layer PCB were divided into digital and analog parts to avoid interferences. The required voltages were provided from low-noise LT1763 and LT3094 low-dropout (LDO) regulators. In the case of the FPGA and μ C, we used the evaluation core boards. The μ C module daughterboard was also projected to ensure proper header connectors with the FPGA evaluation board. Analog input and output signals could be connected via SMA connectors. A photo of the developed hardware platform for digital filtering is presented in Figure 3. In this figure, some additional connectors are also visible (i.e., bypassing analog front-end LPFs and additional clocks).

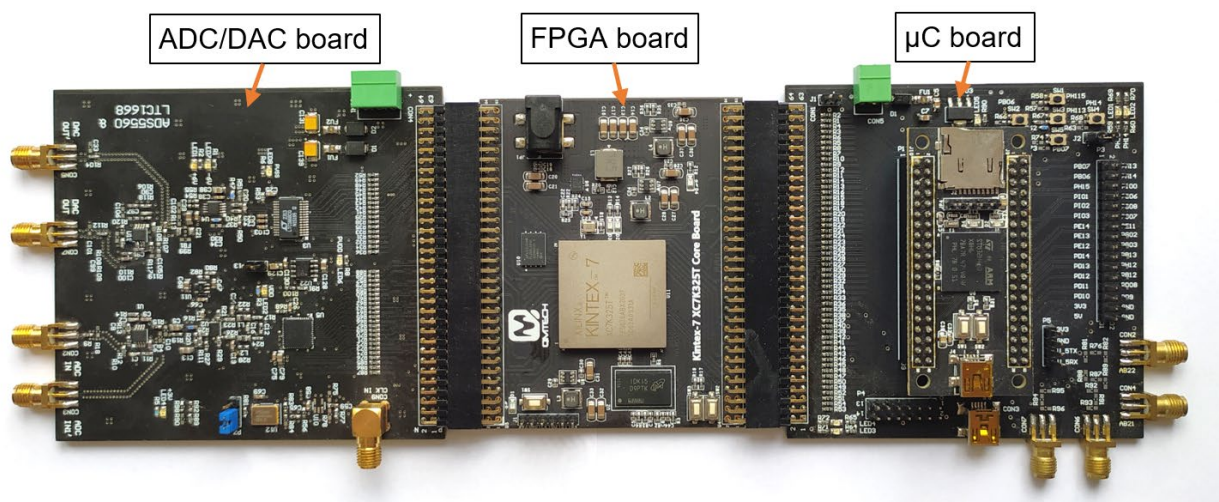


Figure 3. The photo of the hardware platform developed for the digital filtering unit.

Inside the FPGA, we applied a few blocks that were responsible for different functions. The input ADC data in two complement codes were connected to the decimator block. It was responsible for data decimation to obtain a lower sampling frequency than an ADC clock. This was especially important for sampling rates below 1 MSa/s because, according to the ADS5560 datasheet, it can properly operate only above this rate. The decimator can also be set to a decimation factor of 1 to bypass this operation. Then, the data with the needed rate were applied to the filter core that performed the overall filtering operation. An output coding converter was necessary to convert two complement codes into a straight binary, which was required by LTC1668 DAC inputs. Inside the FPGA, the additional memory block can also be applied to enable the possibility of extracting filtered data to μ C. The filter architecture and coefficients were constant and stored inside the filter core block. A dedicated VIVADO design environment was used to prepare, synthesize, and implement Very High-Speed Integrated Circuit Hardware Description Language (VHDL) codes in FPGA. The block diagram of the functionality of the internal module implementation in the FPGA device is presented in Figure 4.

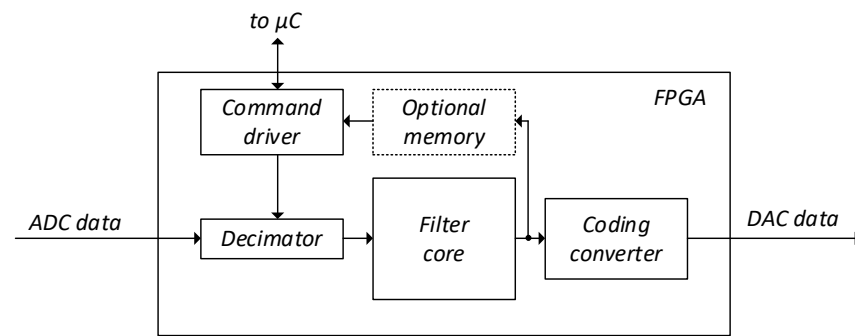


Figure 4. Block diagram of the internal functional module implemented in the FPGA device.

2.2. Digital Filters

Generally, the filter core can be implemented in various forms. Regarding impulse response length, digital filters can be divided into two categories: finite impulse response (FIR) and infinite impulse response (IIR) [45]. Moreover, its architecture types can also be different, like direct forms (i.e., biquad) or transformed forms [46]. All of their properties have an influence on the final behavior and implementation procedure. The example architectures of FIR (direct form) and IIR (direct form I-biquad) in the form of schematic blocks are presented in Figure 5.

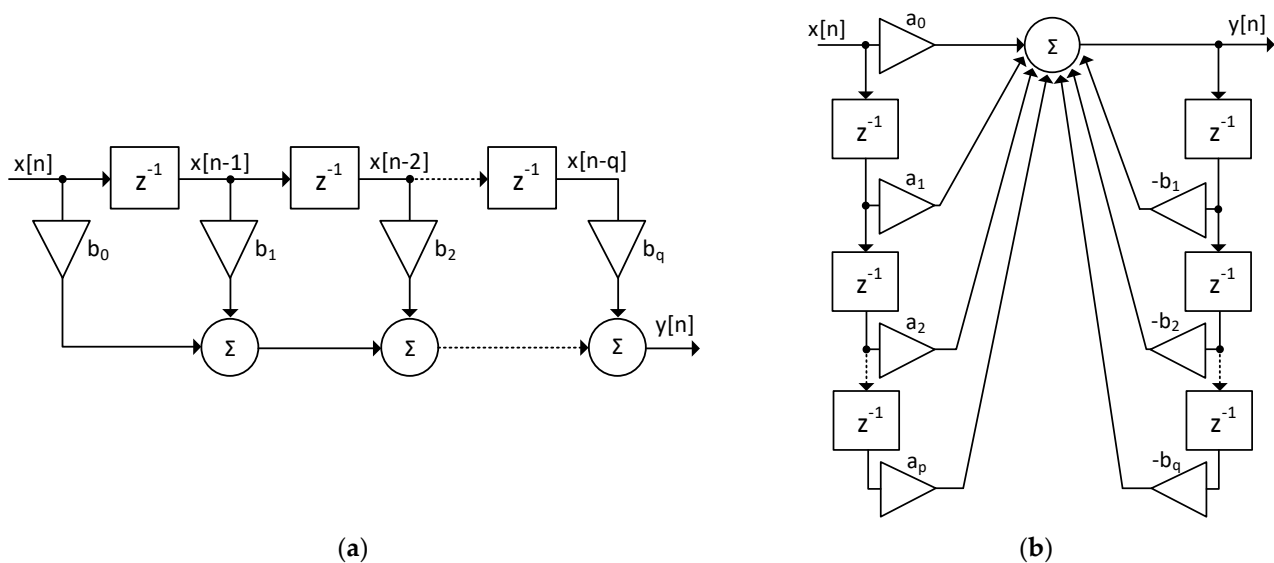


Figure 5. Example schematic block diagrams of direct form FIR (a) and direct form I-biquad IIR (b) filter implementations. Square blocks are delays, and triangular blocks are multipliers by constants.

The FIR structure was composed of a q -stage delay line with $q + 1$ taps. The output data from unit delays were multiplied by the b_i coefficient (impulse response). All multiplications were summed to produce an output result. It was a different situation in the case of the IIR, where the output $y[n]$ sample was given to create a feedback loop connected to the sum block. Such an operation created a situation of infinite impulse response because each output sample was looped back and had some influence on the next output signal sample.

When deciding between the FIR and IIR, parameters related to the phase response and resource availability in the target system are usually taken into account. Table 2 summarizes the main properties of both filter types.

Table 2. Main properties of FIR and IIR filters.

Property	FIR	IIR
Implementation cost	High	Low
Stability	Stable	Can be unstable
Latency	High	Low
Phase response	Linear	Non-linear
Analog equivalent	No	Yes
Customization and implementation	Easy customization and implementation	More difficult customization and implementation

Analyzing the properties of both filters, it should be highlighted that the main differences included implementation costs, phase response, and stability. Comparing implementation costs, it can be stated that regarding similar filters, the IIR filters required fewer hardware resources than the FIR. This is caused mainly by less multiplications being needed by the IIR architecture.

2.3. Filter Core Design, Simulation of Its Characteristics, and HDL Code Preparing for FPGA

The overall behavior of the filtering unit depends on the filter core and sampling frequency. From a mathematical point of view, digital filtering uses the operation of convolution of the input signal with the filter impulse response. Assuming operation on the input discrete signal $x[n]$, the output signal $y[n]$ of an FIR filter can be described as follows [47]:

$$y[n] = \sum_{i=0}^q b_i \cdot x[n-i], \quad (1)$$

where q is the filter order, and b_i is the impulse response value at the i -th instant for $0 \leq i \leq q$ of a q -th-order filter. For the direct form of the FIR, b_i is also a filter coefficient. In the case of the IIR, such an equation takes the following form:

$$y[n] = \frac{1}{a_0} \left[-\sum_{j=1}^p a_j \cdot y[n-j] + \sum_{i=0}^q b_i \cdot x[n-i] \right] \quad (2)$$

Our work used the MATLAB environment to design, simulate, and generate the HDL code. The Filter Designer Tool (FDT) built-in MATLAB R2024a software can prepare an overall filter core HDL code depending on the requirements. Using FDT, we can manage filter properties like type (low-pass, band-pass, etc.), response, design method, order, frequency, and magnitude responses. The calculated coefficients must be quantized and adapted to the desired representations since the FDT operates on floating-point numbers in either double-precision or single-precision format. In the case of the FPGA, we used the FDT built-in function of quantizing into 16-bit fixed-point representations. As the number of different types of filters that can be implemented is quite enormous, we can present only a few examples without a thorough analysis of individual characteristics and design methods in this paper. The fundamentals of Butterworth, Bessel, Chebyshev, or elliptic filters can be found in [48,49]. The considerations about design methods like equiripple or windowing are described in [50,51]. The main differences between these filters are related to some compromise between the passband and stopband behavior. For example, the Chebyshev filter is characterized by high attenuation in the stopband (in comparison with others) at the cost of oscillations of gain in the passband and also after step response. The Bessel type is free of these disadvantages at the expense of weaker attenuation in the stopband.

In the case of our application with the IR detection module, narrow-band filtering can most effectively improve the SNR by attenuating $1/f$ and wide-band white noises. Due to this fact, we decided to present the results focusing on band-pass filters. The example results of simulations in the FDT of FIR windowed-*sinc* band-pass filters (BPFs) for different window types are presented in Figure 6. In this figure, the frequency (Figure 6a), phase (Figure 6b), impulse (Figure 6c), and step (Figure 6d) responses were simulated

for a normalized 840-order (like the number of DSP blocks in the XC7K325T) filter. The passband was set from 0.01 to 0.02 of normalized frequency as some compromise between the narrowest possible band without significant attenuation at its center frequency.

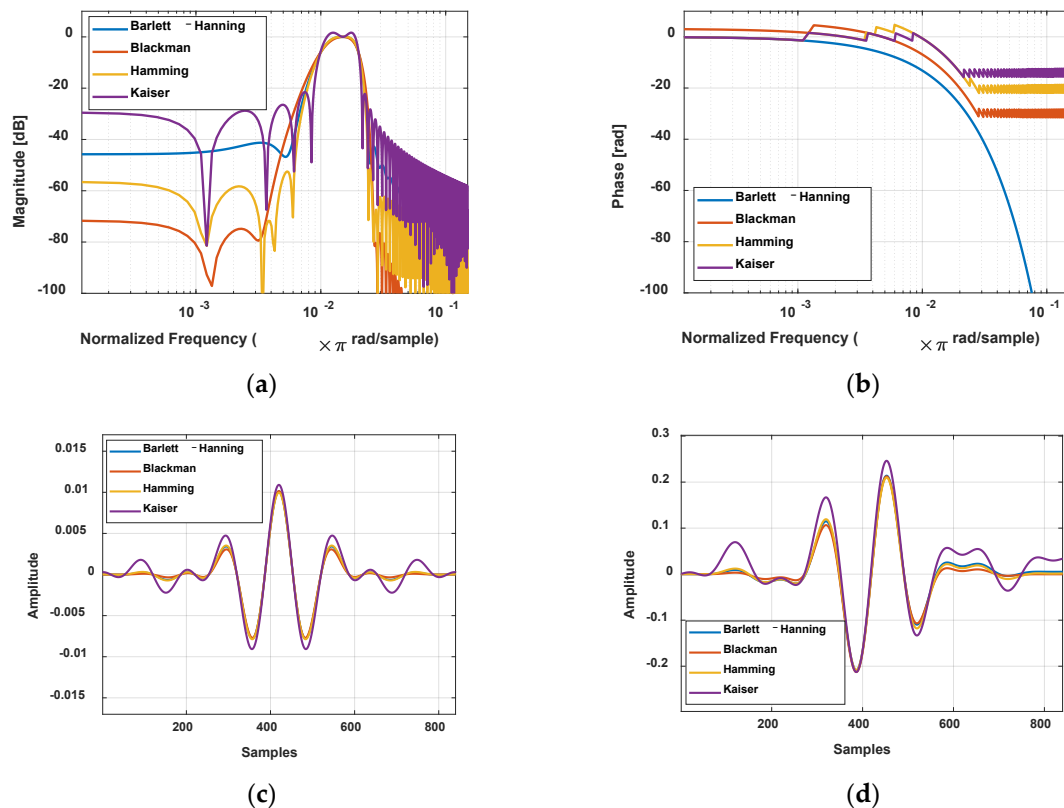


Figure 6. Frequency (a), phase (b), impulse (c), and step (d) responses for normalized 840-order windowed-sinc filter. The passband was set from 0.01 to 0.02 of normalized frequency.

Each of the applied windows caused a change in resulting characteristics. The highest attenuation in the stopband was observed in the case of the Blackman window. This explains the fact that this type of windowing is typically used. The lower attenuation in the stopband was observed for Kaiser and Barlett–Hanning windows at the expense of steeper slopes near the passband. The Barlett–Hanning window was characterized by the weakest oscillations at stopband frequencies and other phase responses (without accumulation). For other types of windows, the accumulation oscillating around a certain level was observed before and after the passband. In the passband range, the phase was linear, with some delay observed for all windows. Analyzing impulse and step responses, the *sinc*-like characteristics with slight differences were observed (the higher amplitude for Kaiser). In the case of pulsed signals, some types of filters can lead to significant distortions, including oscillations visible in the step response characteristics.

As we mentioned before, the performance of the implemented digital filter depends on the number of available resources. In practice, it is the number of multiplications and summations that can be performed in the required time. It must be highlighted that, theoretically, such a filter core can also operate at a rate other than sampling frequency. Then, the rate of signal processing should be much higher than the sampling frequency.

To directly show how the impulse response length (and also a number of coefficients) affects the BPF frequency response, we simulated the equiripple method-designed FIR. The low-pass stopband was set to 0.01, the passband from 0.02 to 0.03, and the high-pass stopband to 0.04 of normalized frequency. The simulation results are presented in Figure 7.

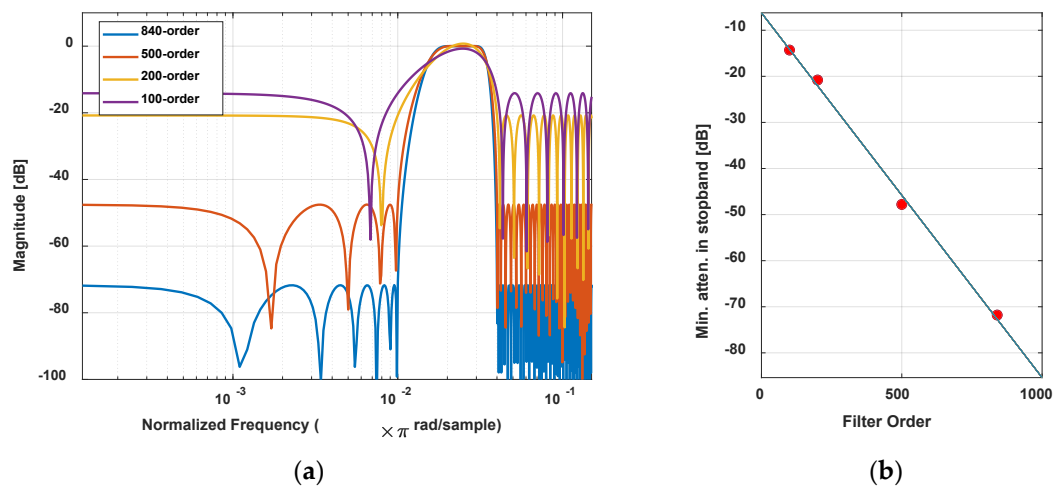


Figure 7. Simulated FIR BPF frequency responses designed using the equiripple algorithm for different orders (a). Dependence between the filter order and minimum attenuation in stopband (b). The low-pass stopband was set to 0.01, the passband from 0.02 to 0.03, and the high-pass stopband to 0.04 of normalized frequency.

Analyzing Figure 7b, it is evident that increasing the order of a digital filter significantly improved attenuation in the stopbands. The 100-order filter gave about -14 dB of attenuation, whereas the 840-order filter gave about -72 dB. It must be noted that some noticeable noise may be added when sending filtered data to the DAC. In many cases, the attenuation of the order of about a dozen dB may provide insufficient or essentially insignificant SNR improvement for many applications.

The disadvantages of high-order FIR filters are primarily a significant time delay, high hardware requirements (to obtain high-order filter), difficulties in dynamic tuning, and the Gibbs' effect (oscillations of the signal amplitude at the boundary between band-pass and stopband).

Similar simulations were conducted for the IIR, such as BPFs. In Figure 8, the frequency-normalized example simulation results of 50-order filters (in the case of elliptic, it was 20-order to maintain stability) are presented. The passband was set to the range from 0.01 to 0.02 of normalized sampling frequency to obtain one octave of the passband. As can be noticed, in comparison with the FIR, it is evident that relatively steeper slopes can be obtained with significantly lower filter orders. However, according to the theory, the phase was non-linear (Figure 8b), and the impulse (Figure 8c) response was very long. Using the IIR, very effective attenuation in the stopbands with relatively small resources (filter order) can be obtained. Due to these facts, such filters are commonly used in low-resource applications where the non-linear phase is not problematic (i.e., smart sensors, telecommunications).

Analyzing the step response, it must be highlighted that the filter types simulated in Figure 8 are typically characterized by relatively high oscillations in step response (Figure 8c) over a relatively long time interval (theoretically infinite, in practice to the level of, i.e., noise). Such phenomena can be quite difficult in the case of pulse or square-like signals. The behavior of the attenuation as a function of the order is, in general, the same as in the case of analog filters, so in many applications, using the tens of orders of IIR filters is unnecessary.

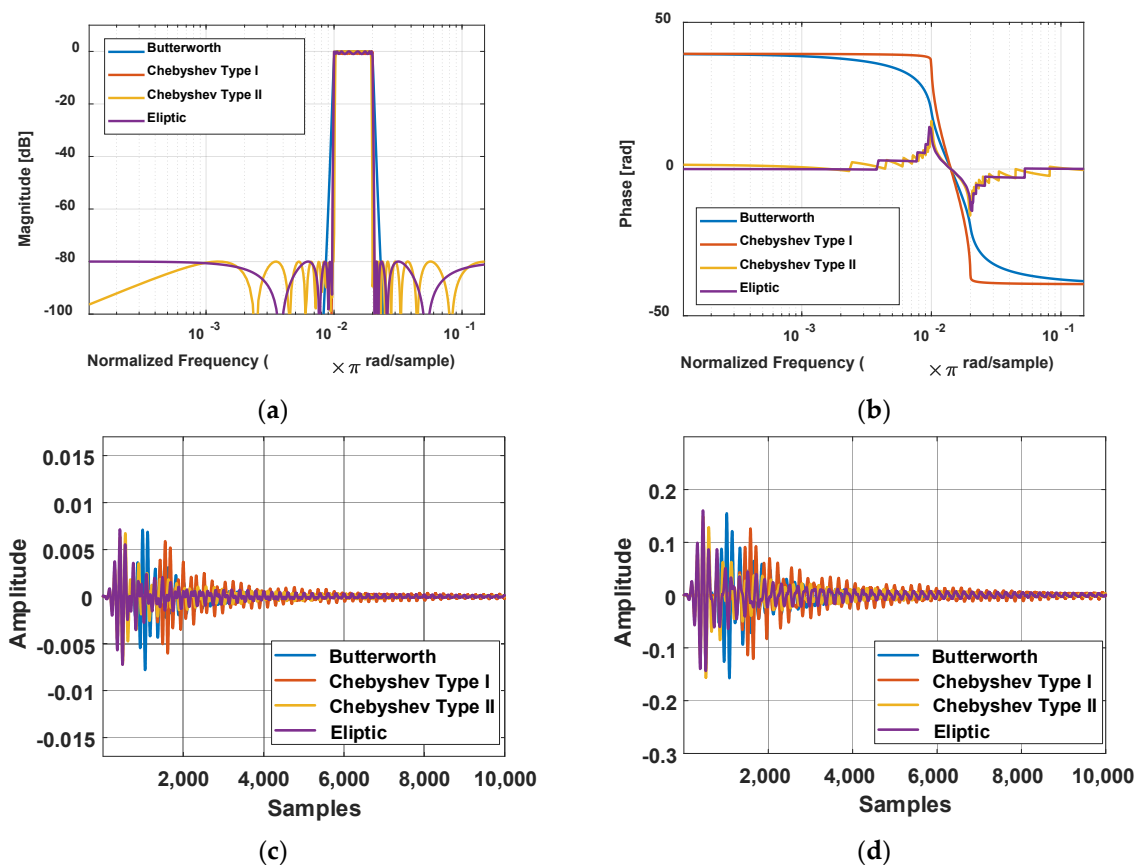


Figure 8. Frequency (a), phase (b), impulse (c), and step (d) responses simulated for the normalized 50-order (20-order for elliptic) IIR filter. The passband was set from 0.01 to 0.02 of normalized frequency.

The simulations of the Butterworth filter for a few orders and the passband set from 0.01 to 0.02 of normalized frequency are presented in Figure 9.

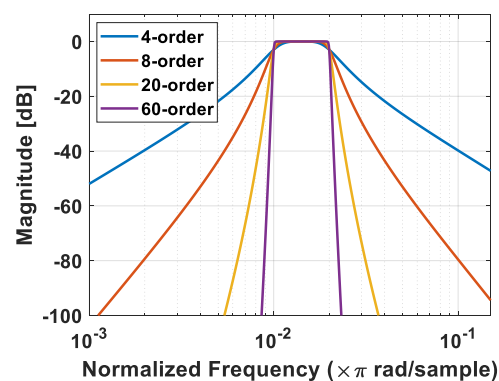


Figure 9. Simulated Butterworth IIR BPF frequency responses. The passband was set from 0.01 to 0.02 of normalized frequency.

2.4. IR Detectors and Amplifiers Noise

As this paper presents a proposition of noise reduction in IR detection modules, a little information about its origin was also included. IR detectors, like other semiconductor components or devices, are a source of noise. They always generate wideband thermal noise according to the Johnson–Nyquist formula ($4kT/R$ in the case of the current calculation). When operating with bias (i.e., to reduce its capacitance in high-speed applications), they also generate shot and low-frequency $1/f$ noise [52]. The latter can be the most problematic due to its significant power. The power spectral density of $1/f$ noise is inversely

proportional to frequency. It also usually rises with bias. In some cases, cooling systems are provided to reduce this noise. In Figure 10, an example of a typical detector–amplifier circuit with noise sources is presented.

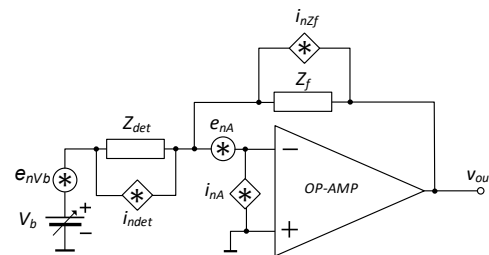


Figure 10. Detector–amplifier circuit with noise sources (IR detection module).

The transimpedance amplifier with voltage (e_{nA}) and current (i_{nA}) noise sources is represented by an operational amplifier with feedback impedance (Z_f). This impedance is also characterized by noise (i_{nzf}), i.e., thermal noise due to the real part (resistance). The IR detector (marked in the form of impedance Z_{det}) total noise is represented by i_{ndet} current noise source, which usually consists of thermal, shot, $1/f$, and generation–recombination noises. In the case of wideband operation, the thermal noise can also be considerable because its root mean square rises with the bandwidth. When operating with a bias voltage (V_b), its noise effect (e_{nVb}) should also be taken into consideration [53]. All these sources shape the characteristics of the power spectrum density, PSD, at the amplifier output and may constitute a significant limitation in applications where weak signals are processed. The circuit shown in Figure 10 can also be regarded as an IR detection module.

3. Results

3.1. Noise in InAsSb IR Detection Module

In the first stage of the work, the MWIR InAsSb IR detection module (Thorlabs PDA10PT-EC) was characterized in terms of noise properties. In Figure 11, the sampled noise output signal (with the detector covered) using an oscilloscope with a 16-bit ADC resolution is presented in Figure 11a. The PSD of this signal is shown in Figure 11b. The detector module output was loaded with an impedance of 50 Ω .

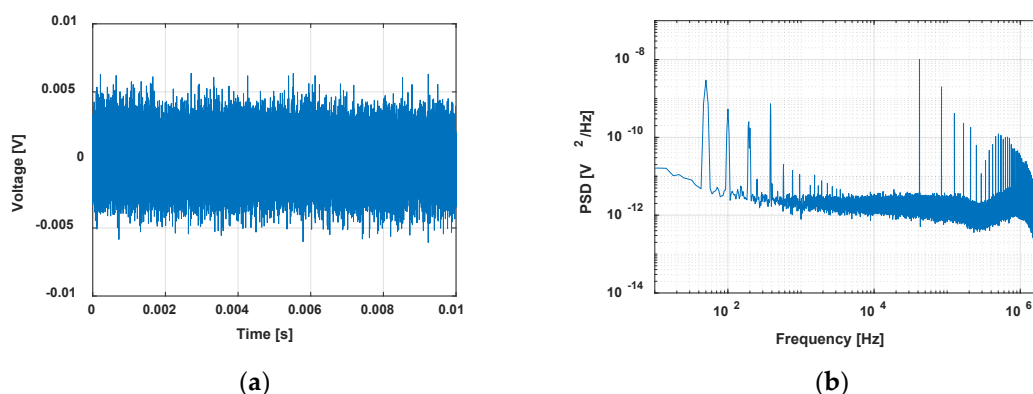


Figure 11. The InAsSb IR detection module output noise signal (a) and its PSD (b). The measurements were provided with a gain set to 40 dB and bandwidth set to 1.6 MHz.

Analyzing Figure 11a, it can be stated that this module was characterized by about 12 mV_{pp} of noise at the output (with the set parameters and 10 ms of observation). Due to this fact, the proper detection of a signal in the range of mV can be difficult or impossible [54]. Regarding the PSD characteristic (Figure 11b), a little more information can be extracted. First of all, it is evident that $1/f$ noise was visible at frequencies up to about 1 kHz. Above this frequency, the wideband white noise was registered at the level of about

1.5 $\mu\text{V}/\sqrt{\text{Hz}}$. Moreover, the high influence of some interferences in the form of single bins was observed. The most likely source of these spectral components were interferences generated by the switching power supply converters or external instrumentation. They were a common problem in sensitive amplifier circuits. Such signals were quite difficult to eliminate by simple methods. Above the frequency of 1 MHz, the detection module bandwidth limitation was visible.

3.2. Implementation and Verification of Sample Filters in the Proposed Unit

We proposed to implement the filter in a dedicated FPGA-based unit platform to reduce the influence of these undesirable signals. The analysis of its properties and effectiveness will be measured and analyzed in the following sections.

To verify the operation of the sample filters, the digital filter was practically implemented in the proposed digital filtering unit, as shown in Figure 3. During this test, two FIR filters and one IIR filter were verified using a 50 mV_{pp} sine sweep signal. The equivalent-type, 800-order FIR#1 filter was designed and simulated (in FDT) to obtain passbands from 2 kHz to 3 kHz and stopbands at 1 kHz and 4 kHz with a frequency of 250 kS/s (using a decimation block). The maximum attenuation in stopbands was set by filter order, whereas the amplitude uncertainty in the passband was set to a maximum of 1 dB. The results of frequency and phase responses simulation and measurement are presented in Figure 12. It should be noted that the measured phase response was rolled up (from $-\pi$ to π), whereas the simulated one was expanded (for all measurements).

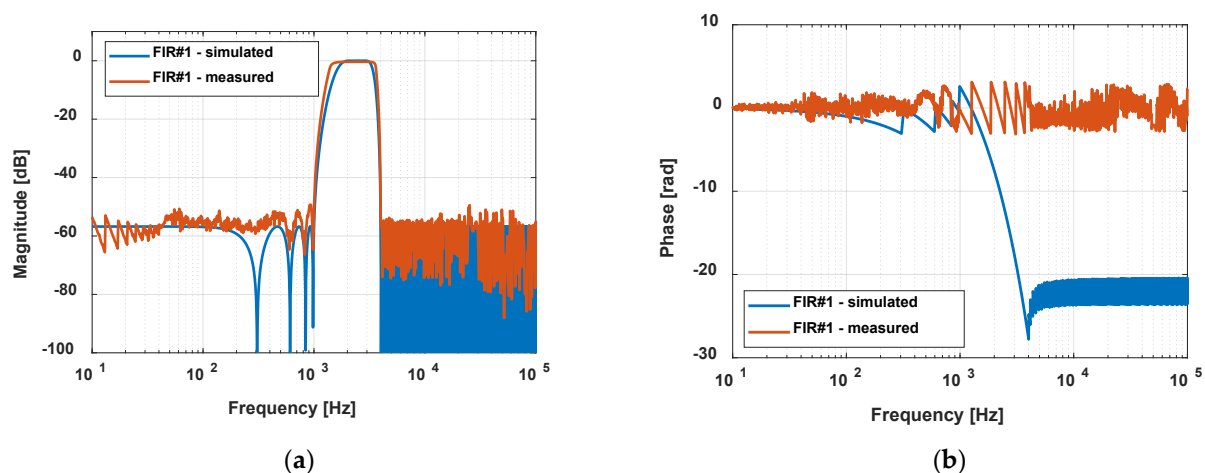


Figure 12. Frequency (a) and phase (b) responses were simulated and measured for the 800-order FIR BPF (FIR#1). The passband was set from 2 kHz to 3 kHz, stopbands at 1 kHz and 4 kHz, and sampling frequency to 250 kS/s. The measured phase was rolled up (from $-\pi$ to π rad).

After analyzing the results, it is clear that the frequency response characteristic corresponded quite well to that of the simulated one. Some differences were visible, probably caused by filter quantization from the floating point representation to fixed point. This procedure usually led to uncertainty due to the loss of precision in number representation. Our work used a 16-bit signed representation of coefficients in VHDL code. When analyzing the behavior of real filters in stopbands, some unusual behavior could be seen. It was caused by the DAC output noise, which directly limited the system's capabilities. In other words, the filter attenuation of the several dozen dB attenuated the input signals to the DAC and output circuit noise level. Regarding the phase response, it is visible that in the passband, the simulated phase corresponded to measures with some shift. This shift was caused by analog front-ends connected to the ADC input and DAC output. Nevertheless, we took into consideration that the log scale in frequency axis, so phase linearity could be noticed, which was in line with the theoretical behavior. Due to the high attenuation, the

phase response in the stopband was difficult or impossible to measure. It can be observed as some noise in the obtained characteristic (Figure 12b).

Similar results were obtained for the equiripple-type 800-order FIR#2 filter, which was designed and simulated to obtain passbands from 200 kHz to 300 kHz and stopbands at 100 kHz and 400 kHz with a sampling frequency of 40 MS/s. Its frequency and phase responses (simulated and measured) are presented in Figure 13.

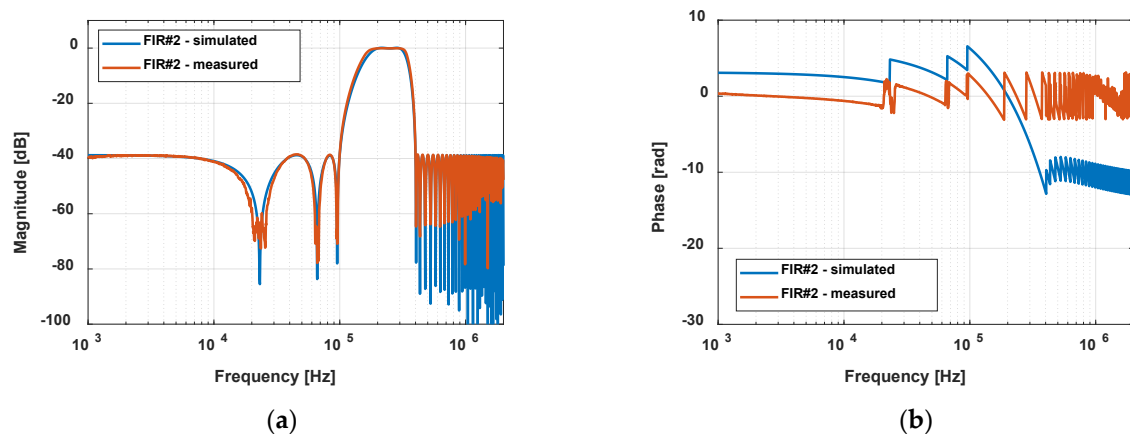


Figure 13. Frequency (a) and phase (b) responses were simulated and measured for 800-order FIR BPF (FIR#2). The passband was set from 200 kHz to 300 kHz, stopbands at 100 kHz and 400 kHz, and a sampling frequency of 40 MS/s. The measured phase was rolled up (from $-\pi$ to π rad). The log scale was used for the frequency axis.

Analyzing the behavior of FIR#2, it is evident that the frequency responses matched quite well. Moreover, a specific shape was also registered in stopbands. It was mainly caused by the fact that obtaining low-frequency narrow-passband filters operating with high sampling frequencies was difficult. This usually requires very long impulse responses (filter taps). Regarding the phase response, there was some shift due to the same reason as in the case of FIR#1.

Since a very effective and popular method of testing filters in the time domain is to use a chirp signal with a specific frequency sweep set to visualize passbands and stopbands, such an experiment was also provided using the RIGOL DG1062Z generator. The result of FIR#2, when the input sine signal was sweeping from 10 kHz to 1 MHz (chirp), is presented in Figure 14.

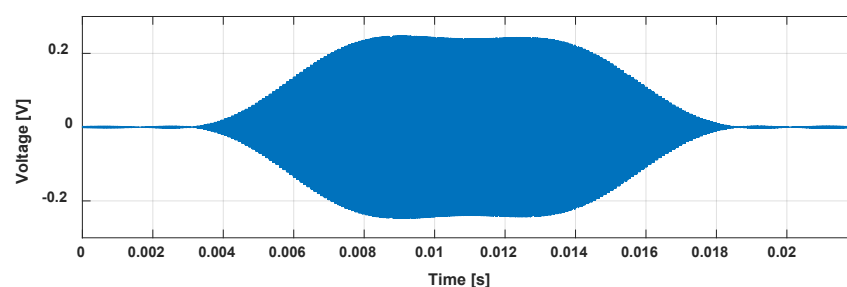


Figure 14. FIR#2 measures filter response to the chirp signal with a sweep from 10 kHz to 1 MHz.

Referring to the oscillogram in Figure 14, the FIR#2 filter worked properly with some insignificant passband gain uncertainty (not exceeding 1 dB). Both high-pass and low-pass edges behaved correctly without unexpected artifacts.

In Figure 15a, we present the FIR#2 filter implementation process in the XC7K325T FPGA chip and the utilization of resources as a percentage of the total resources available (Figure 15b).

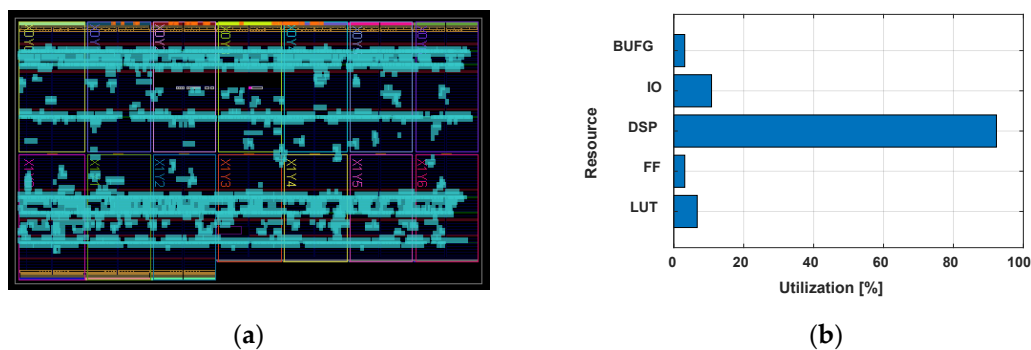


Figure 15. FIR#2 graphical schematic of implementation in FPGA chip (a); utilization of resources (b).

Looking at the location of the filter resources in the FPGA chip, it can be said that they were distributed over the entire surface of the IC. Due to the high utilization of DSP48 blocks located in a few rows from left to right, most of the filter logical components were located in their surroundings. Then, the entire filter was distributed over all the logical component banks and clock regions of the IC. Regarding the percentage utilization, it is evident that FIR#2 used more than 90% of the DSP48 blocks. Each of them included a multiplier of two signed values and several different adders that were perfect for implementing digital filtering operations based on multiplying and accumulating (MAC) [55].

Although IIR filters are not as widely used as FIR ones, we also tested one of these types of filters in practice. The results of an implementation of the IIR filter set up to obtain passbands from 3 kHz to 8 kHz and stopbands at 1 kHz and 10 kHz with frequency at 250 kS/s (using decimation block) are presented in Figure 16. During the design process, the Butterworth type was selected with attenuation in a stopband of 60 dB. Such value led to attenuating unwanted noises to the level of signal processing circuit noise. The filter was designed as a 38-order type II direct form filter with second-order sections.

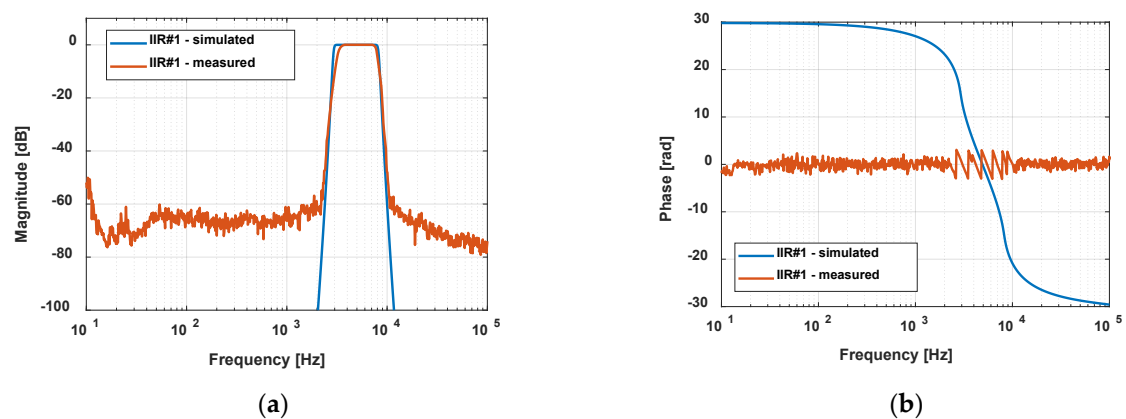


Figure 16. Frequency (a) and phase (b) responses were simulated and measured for the 38-order IIR BPF (IIR#1). The passband was set from 3 kHz to 8 kHz, stopbands at 1 kHz and 10 kHz, and sampling frequency at 250 kS/s. The measured phase was rolled up (from $-\pi$ to π rad). The log scale was used for the frequency axis.

The frequency response of the IIR#2 filter in the passband matched the simulated one, with some differences occurring in the passband edges. It can be caused by a quantization procedure. Moreover, the minimum attenuation in the stopband was limited by the DAC and the analog front-end noise, as well as the limitations of the impedance analyzer used for these tests. As for the phase response (Figure 16b), it can be stated that it was non-linear according to theory. This was confirmed by a straight line from $-\pi$ to π for the measured characteristics in the plot where the frequencies were on a logarithmic scale.

3.3. Analyzing the Filters' Effectiveness During Operation with the IR Detection Modules

The presented filtering unit was also tested with a real noisy signal from the IR detection module. These noise sources were generally discussed in Section 2.4. In this experiment, we used a RIGOL DG1062Z generator to drive the modulation input of a Newport 505B current driver with a *sine*-shape signal. In such a way, the IR LED (Vishay TSAL6200, $\lambda_p = 940$ nm) at the DC operating point set to 35 mA was modulated accordingly. The emitted IR beam was directed at the PDA10PT-EC InAsSb detection module. It operates from about 1 μm to 5.8 μm wavelength. The output signal was connected to the filtering unit and 16-bit scope based on a PCIE-1840L data acquisition (DAQ) card. In the proposed filtering unit, we intentionally implemented FIR#2 to present an operating commonly used FIR filter with 40 dB of attenuation in stopbands, operating at the maximum possible sampling frequency (40 MS/s). The unfiltered signal was also connected to the scope. The experiment measurement setup is presented in Figure 17.

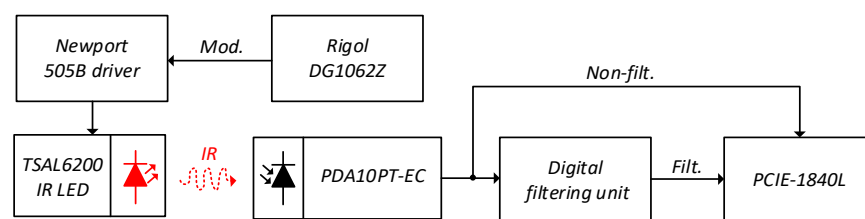


Figure 17. A measurement setup is used to verify the digital filtering unit's operation with a noisy signal.

The unfiltered input and filtered output signals in the time and frequency domain are presented in Figure 18a,b, respectively. The power of the incident IR LED radiation on the detector was deliberately adjusted to obtain a weak SNR (signal in the range of mV, significantly distorted by noise). The modulation signal frequency was set to half the passband (250 kHz).

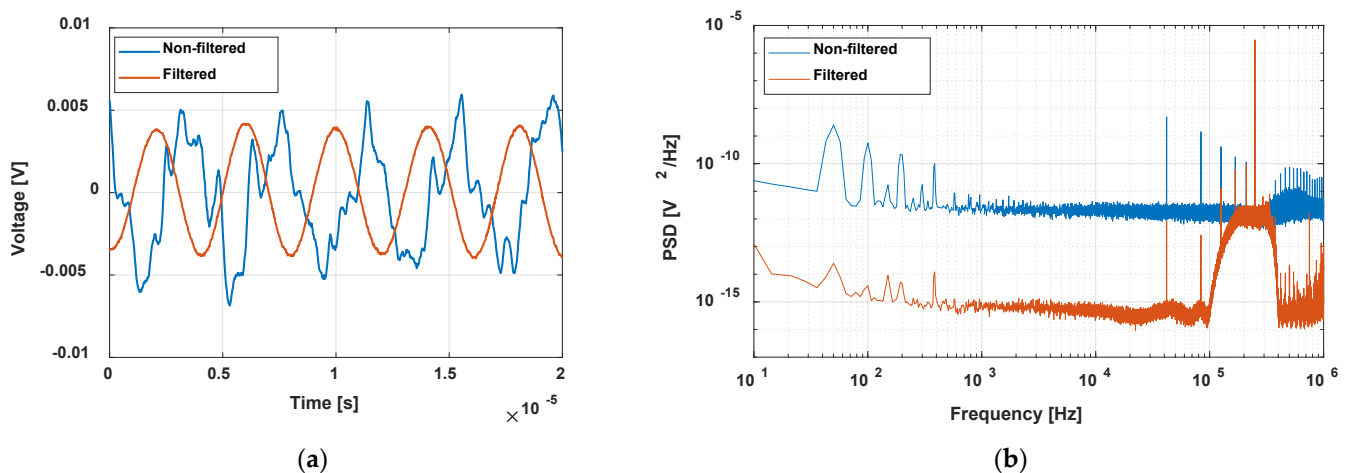


Figure 18. Results of filtering noisy signal from the InAsSb IR detection module using the proposed unit and FIR#2 in the time domain (a) and frequency domain (b).

Working with signals in the mV range was quite difficult due to the noise generated by the detector–amplifier circuit. Looking at the oscillograms in Figure 18a, it can be seen that the *sine*-modulated signal was quite strong and disturbed by additional noise. The filtering unit significantly improved the signal quality through its band selectivity. It was confirmed by both results in the time and frequency domains. Observing the spectrum at the presented frequencies (10 Hz to 1 MHz), the attenuation of unexpected noise of the order of several decades has been achieved. For $f = 10$ kHz, noise was reduced from about

10^{-12} V²/Hz to about 5×10^{-16} V²/Hz. The shape of the desired BPF can also be observed in the spectrum of the filtered signal, with the top of the passband overlapping with the original spectrum (for a non-filtered signal). The calculated improvement of SNR using the spectrum integration method was equal to about 48 dB. This can be considered a very satisfactory result. In describing the obtained spectra, it must be stated that, as before, some limitations were introduced by the noise of the DAC and its front-end circuit. The remaining single bins visible in the spectrum can occur due to specific interferences from switching converters (used to provide supply, i.e., FPGA boards). They can be further eliminated by replacing them with linear regulators. In turn, low-frequency performance can be improved by using a higher-resolution, lower-noise DAC, and an output circuit.

The following steps must be applied to implement the proposed device in the specific application. First, the hardware platform should be prepared by building ADC/DAC, FPGA, and microcontroller modules. Certain properties can be adapted to specific needs in this step by selecting data converters and their front-ends. Then, the requirements of filtering must be considered, like band-pass and stopband characteristics. In some cases, other properties, such as delays or impulse responses, should also be considered. The next step is to prepare the software, including the filter core code that includes the appropriate impulse response, as intended. After programming the microcontroller and configuring the FPGA device, the operation of the prepared filter system should be tested accordingly.

4. Summary

This paper presents the digital filtering unit in the context of working with IR detection modules. The proposed unit operates in real-time, which can be especially useful in systems where a quick response or reaction to a signal is important. Moreover, filtering operations lead to an improved SNR, which limits the performance of the detection systems. The hardware of the proposed filtering unit was built using high-speed and high-resolution 16-bit ADCs and DACs in cooperation with an advanced FPGA device. Thanks to the relatively large amount of available resources in the FPGA, long impulse responses can be implemented, requiring high attention in stopbands and steep slopes of frequency response characteristics. Such prepared hardware is more effective and flexible than previous work [38] regarding the maximum length of the impulse response and sampling frequency.

The very attractive attenuation and linear phase of high-order FIR filters are obtained and designed as the equiripple or Blackman windowed.

In the experimental part, we implemented two FIR filters and one IIR filter in the proposed unit. For the FIR filter, operations with sampling frequencies below 1 MS/s and a maximum of 40 MS/s were verified. The obtained frequency and phase response characteristics were similar to the simulated ones. Nevertheless, the main limitation of maximum attenuation was caused by the DAC and output front-end circuit noises.

The operation of the proposed filter unit has been verified during the test with the InAsSb IR detection module, where selective signal reception is desirable to reduce unwanted components. The results of the measurements confirmed the high effectiveness of the proposed device in the time and frequency domain, where about 20 dB of SNR improvement was obtained with an example FIR BPF.

In conclusion, it must be highlighted that the main purpose of this paper was to present a hardware platform that can be used to implement almost every filtering operation in the range of its properties. The resultant characteristics depend mainly on the implemented algorithm in the form of prepared software codes. On the pages of this paper, we were able to present only a few examples of filters. It must be remembered that other types of filters can also be applied. The presented device can be used in many applications where improvement of the SNR led to better performances of the overall specific system (i.e., detection expanded to a more advanced form of surveillance, communication, medicine, and security). The presented device can also be expanded to a more advanced form by preparing software according to individual needs. It can be an alternative to other methods,

such as lock-in, cross-correlation, or AI algorithms, which can sometimes return unexpected results when operating with unknown signals.

Author Contributions: Conceptualization, K.A.; methodology, K.A.; software, K.A.; validation, K.A., R.S. and Z.B.; formal analysis, Z.B.; investigation, K.A.; resources, K.A.; data curation, K.A.; writing—original draft preparation, K.A.; writing—review and editing, K.A., R.S. and Z.B.; visualization, K.A.; supervision, R.S. and Z.B.; project administration, K.A.; funding acquisition, K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Military University of Technology, grant no. UGB 22-871.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

Abbreviations

Full Term	Abbreviation	Full Term	Abbreviation
Field-Programmable Gate Array	FPGA	Analog-to-Digital Converter	ADC
Digital Signal Processing	DSP	Digital-to-Analog Converter	DAC
Integrated Circuit	IC	Printed Circuit Board	PCB
Hardware Description Language	HDL	Low-Dropout (voltage regulator)	LDO
Application-Specific Integrated Circuit	ASIC	Finite Impulse Response	FIR
Infrared	IR	Infinite Impulse Response	IIR
Short-Wave IR	SWIR	Filter Designer Tool	FDT
Mid-Wave IR	MWIR	Very High Speed Integrated Circuit Hardware Description Language	VHDL
Very-Long-Wave IR	VLWIR	Multiplying And Accumulating	MAC
Mercury Cadmium Telluride	MCT	Band-Pass Filter	BPF
Transimpedance Amplifier	TIA	System-On-Chip	SoC
Voltage Amplifier	VA	Application Specific ICs	ASICs
Noise Equivalent Power	NEP	Signal-to-Noise Ratio	SNR
Power Spectrum Density	PSD	Microcontroller	μC
Central Processing Unit (Processor)	CPU	Data Acquisition	DAQ
Fast Fourier Transform	FFT	Light-Emitting Diode	LED

References

1. Woods, R.; McAllister, J.; Lightbody, G.; Yi, Y. *FPGA-Based Implementation of Signal Processing Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2008. [\[CrossRef\]](#)
2. McAllister, J. FPGA-based DSP. In *Handbook of Signal Processing Systems*; Springer: Boston, MA, USA, 2010; pp. 363–392.
3. Lielāmurs, E.; Cvetkovs, A.; Novickis, R.; Ozols, K. Infrared Image Pre-Processing and IR/RGB Registration with FPGA Implementation. *Electronics* **2023**, *12*, 882. [\[CrossRef\]](#)
4. Leong, P.H. Recent Trends in FPGA Architectures and Applications. In Proceedings of the 4th IEEE International Symposium on Electronic Design, Test and Applications (DELTA 2008), Hong Kong, China, 23–25 January 2008; pp. 137–141. [\[CrossRef\]](#)
5. Rodríguez-Andina, J.J.; Valdes-Pena, M.D.; Moure, M.J. Advanced Features and Industrial Applications of FPGAs—A Review. *IEEE Trans. Ind. Inform.* **2015**, *11*, 853–864. [\[CrossRef\]](#)
6. Qin, X.; Zhang, W.; Wang, L.; Zhao, Y.; Tong, Y.; Rong, X.; Du, J. An FPGA-Based Hardware Platform for the Control of Spin-Based Quantum Systems. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1127–1139. [\[CrossRef\]](#)
7. Wolf, W. *FPGA-Based System Design*; Pearson Education: Upper Saddle River, NJ, USA, 2004.
8. Einspruch, N. (Ed.) *Application Specific Integrated Circuit (ASIC) Technology*; Academic Press: New York, NY, USA, 2012; Volume 23.
9. Wang, X.; Li, F.; Liu, S.; Miao, P.; Zhang, Z.; Geng, T.; Jin, G. A Scanning Test System of p/sFEB Based on FPGA XADC for the ATLAS Phase-I sTGC Upgrade. *IEEE Trans. Nucl. Sci.* **2019**, *66*, 1249–1253. [\[CrossRef\]](#)
10. Visser, S.J.; Dawood, A.S.; Williams, J.A. FPGA Based Real-Time Adaptive Filtering for Space Applications. In Proceedings of the 2002 IEEE International Conference on Field-Programmable Technology (FPT), Hong Kong, China, 16–18 December 2002; pp. 322–326. [\[CrossRef\]](#)
11. Wang, X.; He, X.; Zhu, X.; Zheng, F.; Zhang, J. Lightweight and Real-Time Infrared Image Processor Based on FPGA. *Sensors* **2024**, *24*, 1333. [\[CrossRef\]](#)

12. Kwiatkowski, P.; Szplet, R. Efficient Implementation of Multiple Time Coding Lines-Based TDC in an FPGA Device. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 7353–7364. [\[CrossRef\]](#)
13. Ozpolat, E.; Karakaya, B.; Kaya, T.; Gulten, A. FPGA-based digital Filter Design for Biomedical Signal. In Proceedings of the 2016 XII International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv-Polyana, Ukraine, 20–24 April 2016; IEEE: New York, NY, USA, 2016; pp. 70–73. [\[CrossRef\]](#)
14. Rogalski, A. Infrared Detectors: An Overview. *Infrared Phys. Technol.* **2002**, *43*, 187–210. [\[CrossRef\]](#)
15. Rogalski, A. Infrared Detectors: Status and Trends. *Prog. Quantum Electron.* **2003**, *27*, 59–210. [\[CrossRef\]](#)
16. Kinch, M.A. The Future of Infrared; III–Vs or HgCdTe? *J. Electron. Mater.* **2015**, *44*, 2969–2976. [\[CrossRef\]](#)
17. Pham, T.; Du, W.; Tran, H.; Margetis, J.; Tolle, J.; Sun, G.; Yu, S.Q. Systematic Study of Si-Based GeSn Photodiodes with 2.6 μm Detector Cutoff for Short-Wave Infrared Detection. *Opt. Express* **2016**, *24*, 4519–4531. [\[CrossRef\]](#)
18. Karim, A.; Andersson, J.Y. Infrared Detectors: Advances, Challenges and New Technologies. In Proceedings of the IOP Conference Series: Materials Science and Engineering, London, UK, 18 December 2013; Volume 51, p. 012001. [\[CrossRef\]](#)
19. Rogalski, A. Graphene-Based Materials in the Infrared and Terahertz Detector Families: A Tutorial. *Adv. Opt. Photonics* **2019**, *11*, 314–379. [\[CrossRef\]](#)
20. Menolfi, C.; Huang, Q. A Low-Noise CMOS Instrumentation Amplifier for Thermoelectric Infrared Detectors. *IEEE J. Solid State Circuits* **1997**, *32*, 968–976. [\[CrossRef\]](#)
21. Rogalski, A.; Bielecki, Z. *Detection of Optical Signals*; CRC Press: Boca Raton, FL, USA, 2022.
22. Ciura, Ł.; Kopytko, M.; Martyniuk, P. Low-Frequency Noise Limitations of InAsSb- and HgCdTe-Based Infrared Detectors. *Sens. Actuators A Phys.* **2020**, *305*, 111908. [\[CrossRef\]](#)
23. Huth, S.T.; Breitenstein, O.; Huber, A.; Dantz, D.; Lambert, U.; Altmann, F. Lock-in IR-Thermography—A Novel Tool for Material and Device Characterization. In *Diffusion and Defect Data Part B Solid State Phenomena*; Scitec Publications: Zurich, Switzerland, 2002; pp. 741–746. [\[CrossRef\]](#)
24. Hamamatsu. *Characteristics and Use of Infrared Detectors*; Technical Information SD-12; Hamamatsu: Hamamatsu, Japan, 2004.
25. Bhan, R.K.; Dhar, V. Recent Infrared Detector Technologies, Applications, Trends and Development of HgCdTe Based Cooled Infrared Focal Plane Arrays and Their Characterization. *Opto-Electron. Rev.* **2019**, *27*, 174–193. [\[CrossRef\]](#)
26. Rogalski, A. HgCdTe Infrared Detector Material: History, Status and Outlook. *Rep. Prog. Phys.* **2005**, *68*, 2267. [\[CrossRef\]](#)
27. Sharma, U.C. Infrared Detectors. In *Tech Credit Seminar Report*; Electronic Systems Group, IIT Bombay: Bombay, India, 2004.
28. Mikołajczyk, J.; Bielecki, Z.; Bugajski, M.; Piotrowski, J.; Wojtas, J.; Gawron, W.; Prokopiuk, A. Analysis of Free-Space Optics Development. *Metrol. Meas. Syst.* **2017**, *24*, 653–674. [\[CrossRef\]](#)
29. Porter, S.; Mansi, M.; Sumpter, N.; Galloway, L. Low Cost Smart Sensors Using Infrared Detector Arrays. *Sens. Rev.* **2001**, *21*, 283–287. [\[CrossRef\]](#)
30. Diakides, N.A.; Bronzino, J.D. Advances in Medical Infrared Imaging. In *Medical Infrared Imaging*; CRC Press: Boca Raton, FL, USA, 2007; pp. 19–32. [\[CrossRef\]](#)
31. Corsi, C. Infrared: A Key Technology for Security Systems. *Adv. Opt. Technol.* **2012**, *2012*, 838752. [\[CrossRef\]](#)
32. Antoniou, A. *Digital Signal Processing*; McGraw-Hill: New York, NY, USA, 2016.
33. Mulgrew, B.; Grant, P.; Thompson, J. *Digital Signal Processing: Concepts and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2002.
34. Chronopoulos, S.K. Flexible and Reconfigurable OFDM Implementation in DSP Platform for Various Purposes and Applications. *Sensors* **2024**, *24*, 2732. [\[CrossRef\]](#)
35. Dan, L.; Guowei, G. Design of Attitude and Heading Reference System Based on DSP and STM32. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 2733–2738. [\[CrossRef\]](#)
36. Fayyazi, M.; Movahedin, M.R.; Navabi, Z.; Riahi, P.A.; Dezfoli, A. An Efficient CPU Architecture for DSP Processor. In Proceedings of the Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, 17–19 May 1999; pp. 63–70.
37. Noor, S.M.; John, E.; Panday, M. Design and Implementation of an Ultralow-Energy FFT ASIC for Processing ECG in Cardiac Pacemakers. *IEEE Trans. Very Large Scale Integr. Syst.* **2018**, *27*, 983–987. [\[CrossRef\]](#)
38. Achtenberg, K.; Mikołajczyk, J.; Bielecki, Z. IR Detection Module with Integrated Real-Time FIR Filter Implemented in FPGA. *Int. J. Electron. Telecommun.* **2024**, *70*, 175–182. [\[CrossRef\]](#)
39. Baker, B.C. Designing an Anti-Aliasing Filter for ADCs in the Frequency Domain. *Analog. Appl.* **2015**. Available online: <https://www.ti.com/lit/an/slyt631/slyt631.pdf#page=7> (accessed on 21 August 2024).
40. Kugelstadt, T. Active Filter Design Techniques. In *Op Amps for Everyone*; Newnes: Oxford, UK, 2009; pp. 365–438.
41. Texas Instruments. *ADS556x 16-Bit, 40 and 80 MSPS ADCs with DDR LVDS and CMOS Outputs*; Datasheet; Texas Instruments: Dallas, TX, USA, 2016.
42. Xilinx. *7 Series FPGAs Data Sheet: Overview*; DS180; Xilinx: San Jose, CA, USA, 2020.
43. Linear Technology. *LTC1666/LTC1667/LTC1668 12-Bit, 14-Bit, 16-Bit, 50MSPS DACs*; Datasheet; Linear Technology: Milpitas, CA, USA, 2000.
44. Maxim Integrated. *Equalizing Techniques Flatten DAC Frequency Response*; Application Note 3853; Maxim Integrated: San Jose, CA, USA, 2012.
45. Litwin, L. FIR and IIR Digital Filters. *IEEE Potentials* **2000**, *19*, 28–31. [\[CrossRef\]](#)

46. Gettings, K.M.; Bolstad, A.K.; Ericson, M.N.; Wang, X. Biquad Implementation of an IIR Filter for IQ Mismatch Correction in an SoC RF Receiver. In Proceedings of the 2013 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 10–12 September 2013; pp. 1–5. [\[CrossRef\]](#)
47. D’Antona, G.; Ferrero, A. Finite Impulse Response (FIR) Filters. In *Digital Signal Processing for Measurement Systems: Theory and Applications*; Springer: Boston, MA, USA, 2006; pp. 179–220. [\[CrossRef\]](#)
48. Antoniou, A. *Digital Filters: Analysis, Design, and Signal Processing Applications*; McGraw-Hill Education: New York, NY, USA, 2018.
49. Horowitz, P.; Hill, W.; Robinson, I. *The Art of Electronics*, 2nd ed.; Cambridge University Press: Cambridge, UK, 1989.
50. Samueli, H. On the Design of Optimal Equiripple FIR Digital Filters for Data Transmission Applications. *IEEE Trans. Circuits Syst.* **1988**, *35*, 1542–1546. [\[CrossRef\]](#)
51. Xu, Y. Design of FIR Filter with Several Window Functions. In Proceedings of the 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC), Chongqing, China, 19–21 November 2021; pp. 638–642. [\[CrossRef\]](#)
52. Zhu, L.; Huang, J.; Xie, Z.; Deng, Z.; Chen, L.; Lin, C.; Chen, B. Low-Frequency Noise Spectroscopy Characterization of HgCdTe Infrared Detectors. *IEEE Trans. Electron. Devices* **2020**, *67*, 547–551. [\[CrossRef\]](#)
53. Achtenberg, K.; Scandurra, G.; Mikołajczyk, J.; Ciofi, C.; Bielecki, Z. Transimpedance Amplifier for Noise Measurements in Low-Resistance IR Photodetectors. *Appl. Sci.* **2023**, *13*, 9964. [\[CrossRef\]](#)
54. Bielecki, Z. Maximisation of Signal-to-Noise Ratio in Infrared Radiation Receivers. *Opto. Electron. Rev.* **2002**, *10*, 209–216.
55. Xilinx. *7 Series DSP48E1 Slice—User Guide*; UG479; Xilinx: San Jose, CA, USA, 2018.

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.