Article

A Method of Increasing Digital Filter Performance Based on Truncated Multiply-Accumulate Units

Pavel Lyakhov ¹ , Maria Valueva ^{1,*}, Georgii Valuev ¹ and Nikolai Nagornov ²

¹ Department of Mathematical Modeling, North-Caucasus Federal University, 355009 Stavropol, Russia; ljahov@mail.ru (P.L.); elasgreece92@mail.ru (G.V.)

² Department of Automation and Control Processes, St. Petersburg Electrotechnical University “LETI”, 197376 Saint Petersburg, Russia; sparta1392@mail.ru

* Correspondence: mriya.valueva@mail.ru; Tel.: +79-887-458-885

Received: 18 November 2020; Accepted: 15 December 2020; Published: 18 December 2020

Abstract: This paper proposes new digital filter architecture based on a modified multiply-accumulate (MAC) unit architecture called truncated MAC (TMAC), with the aim of increasing the performance of digital filtering. This paper provides a theoretical analysis of the proposed TMAC units and their hardware simulation. Theoretical analysis demonstrated that replacing conventional MAC units with modified TMAC units, as the basis for the implementation of digital filters, can theoretically reduce the filtering time by 29.86%. Hardware simulation showed that TMAC units increased the performance of digital filters by up to 10.89% compared to digital filters using conventional MAC units, but were associated with increased hardware costs. The results of this research can be used in the theory of digital signal processing to solve practical problems such as noise reduction, amplification and suppression of the frequency spectrum, interpolation, decimation, equalization and many others.

Keywords: digital signal processing; digital filter; multiply-accumulate unit

1. Introduction

Digital filtering is the core of digital signal processing since it is the foundation of the solution to most practical problems in this area: noise reduction [1], amplification and suppression of frequencies [2], interpolation [3], decimation [4], equalization [5] and many others. The tool for digital signal processing is a digital filter (DF), which is usually divided into filters with either finite impulse response (FIR) or infinite impulse response (IIR). In digital circuit design, there is a need to increase device performance. Usually, the two approaches for improving the performance of digital devices are distinguished as pipelining [6] and parallelization [7].

The implementation principle of IIR filters is similar to that of FIR filters in terms of arithmetic units. The difference is the recursive connections, which do not affect system performance in terms of frequency [8]. Therefore, in this article, we will consider the architecture of FIR filters.

FIR filters are usually components of complex digital signal processing systems; therefore, FIR filter performance affects the performance of the entire system. For instance, FIR filters were applied to image filtering [9]. In that paper, the authors propose a vectorizing pattern that accelerates FIR filters. The authors of [10] proposed a method of separable two-dimensional FIR filter design to increase device performance. In [11], an adaptive FIR filter was developed and the authors performed a simulation for the various device parameters. The authors of [12] proposed a constant multiplier based on the vertical-horizontal binary common sub-expression elimination algorithm and its application in FIR filter implementation. This multiplication technique allowed for the improvement in area, delay and power consumption of the device. Another way to improve the technical characteristics of FIR filters is

the use a residual number system, which allows for parallelization of the computation across multiple channels [13].

The main tool of digital filtering is the multiply-accumulate (MAC) unit [14,15]. In [16], the authors propose a MAC unit architecture based on truncated multipliers and approximate adders. The authors of [17] designed an approximate MAC unit based on input awareness, which consisted of approximate multipliers and input-aware conditional blocks. This approach allowed for the achievement of high energy efficiency. In [18], the authors proposed a modified MAC unit that had a novel partial product reduction block. The authors of [19] conducted a comparative analysis of different precision-scalable MAC unit architectures.

The form of number representation has a great influence on the performance of blocks. In [20], MAC units were proposed, in which the technique of generating partial products combined the multiplier and the adder. Two's complement arithmetic was used to perform operations on negative numbers in these units. The effectiveness of this approach for application in deep neural networks was considered. Another way to represent numbers is in posit format. In [21], the authors presented a generator of posit MAC units and their use in deep-learning applications. In [22], for the implementation of DFs, single precision floating point Vedic multipliers were used.

In order to reduce the latency and hardware cost of devices containing MAC units, researchers have used various compression techniques. The authors of [23] proposed a 4:2 compression technique to add partial products in MAC units. In [24], the authors proposed a low-latency MAC unit architecture using the column bit compression technique.

In the present study, a new modified MAC unit called truncated MAC (TMAC) was developed to increase the performance of FIR DFs. This paper contains a theoretical analysis and hardware simulation on field-programmable gate array (FPGA) FIR DFs containing the proposed modified TMAC units and a comparative analysis with FIR DFs using traditional MAC units.

The remainder of this paper is organized as follows: Section 2 discusses the structure of FIR DFs and presents the FIR filter architecture using the proposed TMAC units. Section 3 presents the theoretical analysis and hardware simulation results, and the conclusion of the paper is reported in Section 4.

2. Materials and Methods

2.1. Digital Filters

A sequence of signal samples is generated by an analog-to-digital converter or transmitted by a computing bus from a digital source. Then, the digital signal $X(N)$ is fed to the input of the FIR DF. An output signal $Y(N)$ is generated by the formula

$$Y(N) = \sum_{i=0}^K b_i X(N-i), \quad (1)$$

where b_i are filter coefficients and K is a filter order.

Figure 1 shows the FIR DF circuit. The z^{-1} denote the signal delay blocks for one sample, which are implemented using buffers in practice. In other words, when a signal arrives at the input of the z^{-1} block, the signal $X(N-1)$ is generated at the output of this block. The basis of the circuit shown in Figure 1 is the repeated execution of the multiplication operation and addition with some intermediate values. In modern digital signal processing, it is customary to combine these two operations into one MAC unit. Since no signal is already available for addition into the first MAC unit, 0 is fed to the input of the unit as a summand.

Figure 1. Finite impulse response (FIR) digital filter (DF) scheme of order K . MAC is multiply-accumulate unit.

2.2. Multiply-Accumulate Units

Consider the implementation of a MAC unit in the FIR DF node corresponding to the coefficient b_i . This unit performs calculations using the formula

$$Y_i = b_i X(N - i) + Y_{i-1}, \quad (2)$$

where Y_i is the result of the current MAC unit and Y_{i-1} is the result of the previous MAC unit.

To obtain a result, according to Formula (2), there is no need to perform a complete multiplication $b_i X(N - i)$. Instead, it is enough to use the generator of k partial products, where $k = \lceil \log_2 b_i \rceil$ is the bit width of filter coefficient b_i , and a carry-save adder (CSA) tree [25], without using the final addition of the Kogge–Stone adder (KSA) [26]. Instead, an additional term Y_{i-1} can be fed to the CSA tree, and the outputs A and B of this tree can be summed in the KSA.

The MAC unit operating according to this principle is shown in Figure 2. Using the notation $((k + 1):2)$, it can be shown that $(k + 1)$ terms are fed to the input of the CSA tree, and two terms are formed at the output.

Figure 2. Structure of the multiply-accumulate (MAC) unit. CSA is carry-save adder. KSA is Kogge–Stone adder.

The basic device for performing arithmetic operations is a full adder (FA) [25]. Bits α , β and the carry C_{in} are the inputs of the device, which are converted to output bits S and C_{out} using the formulas

$$\begin{aligned} S &= \alpha \oplus \beta \oplus C_{in}, \\ C_{out} &= (\alpha \&\beta) \vee (C_{in} \&(\alpha \oplus \beta)) \end{aligned} \quad (3)$$

where bit S is a sum, the output bit C_{out} is a carry obtained in the FA, $\oplus$ is an exclusive disjunction, $\&$ is a conjunction and $\vee$ is a disjunction.

The main idea of a CSA is to transform three input vectors of a device into two output vectors: sum and carry. At the same time, the amount of information for processing at the next step is reduced by 1.5 times.

Another modification of the adders is the Kogge–Stone parallel-prefix adders. Consider the addition of two k -bit numbers A and B . The idea of the parallel-prefix implementation is performed in three steps. At the first stage, the carry-generate bits G_i , the carry-propagate bits P_i and the half-sums H_i are pre-calculated for i , $0 \leq i \leq k-1$:

$$G_i = A_i \& B_i, \quad P_i = A_i \vee B_i, \quad H_i = A_i \oplus B_i \quad (4)$$

The second stage of addition, called the parallel-prefix network, computes the carry bits C_i , for $0 \leq i \leq k-1$, using G_i and P_i . For this, an operator $\circ$ is used that connects pairs of carry-generate and carry-propagate bits, and is defined as

$$(G, P) \circ (G', P') = (G \vee (P \& G'), P \& P'). \quad (5)$$

The sequential calculation of carry-generate and carry-propagate bit pairs (G, P) are denoted as $(G_{i:j}, P_{i:j})$, $i > j$, where the corresponding pair is calculated based on the bits $i, i-1, \dots, j$ in the following way:

$$(G_{i:j}, P_{i:j}) = (G_i, P_i) \circ (G_{i-1}, P_{i-1}) \circ \dots \circ (G_j, P_j). \quad (6)$$

Since the carry is $C_i = G_{i:0}$ for all $i > 0$, all carries can be calculated using only the operator $\circ$ [26]. At the third stage, the sum is calculated as

$$S_0 = H_0 \oplus C_{in}, \quad S_i = H_i \oplus C_{i-1}, \quad S_k = C_{k-1} \quad (7)$$

for $0 \leq i \leq k-1$.

Figure 3 shows the basic blocks for the parallel-prefix addition. Block 3a implements Formula (4). Block 3b implements Formula (5). No action takes place in block 3c. Block 3d implements Formula (7). Figure 4 shows the parallel-prefix adder scheme with the organization of a parallel-prefix network, according to the Kogge–Stone method.

For a theoretical analysis of the digital device parameters, we used an abstract model for calculating the delay and area of the very large-scale integration (VLSI), known as a unit-gate model [27]. If we denote the logical device delay calculated according to the specified model as U_{delay} and logical device area as U_{area} then logic gates are formulated in the following way:

$$U_{delay}(NOT) = 0, \quad U_{area}(NOT) = 0 \quad (8)$$

$$U_{delay}(AND) = 1, \quad U_{area}(AND) = 1 \quad (9)$$

$$U_{delay}(OR) = 1, \quad U_{area}(OR) = 1 \quad (10)$$

$$U_{delay}(XOR) = 2, \quad U_{area}(XOR) = 2 \quad (11)$$

$$U_{delay}(XNOR) = 2, \quad U_{area}(XNOR) = 2 \quad (12)$$

Figure 3. The structure of the basic blocks of the parallel-prefix adder: (a) the first stage block; (b,c) the second stage blocks; and (d) the third stage block.

Figure 4. The structure of the 8-bit parallel-prefix Kogge–Stone adder.

Then, taking into account Formulas (3) and (8)–(12), the delay and area FA can be written as

$$U_{\text{delay}}(\text{FA}) = 4, U_{\text{area}}(\text{FA}) = 7 \quad (13)$$

CSAs consist of FA blocks (Figure 3); therefore, the delay and area parameters are defined as follows:

$$U_{delay}(CSA) = U_{delay}(FA) = 4 \quad (14)$$

$$U_{area}(CSA) = k \cdot U_{area}(FA) = 7k \quad (15)$$

For the KSA, when condition $C_{in} = 0$ is satisfied, which does not require a logical operation $\oplus$ for calculating S_0 by Formula (7), the parameters of the delay and area are determined by the formulas

$$U_{delay}(KSA) = 2 + 2 \cdot \lceil \log_2 k \rceil + 2 \approx 2\log_2 k + 4 \quad (16)$$

$$U_{area}(KSA) = 4k + 3 \cdot (k \cdot \lceil \log_2 k \rceil - (2^{\lceil \log_2 k \rceil} - 1)) + 2(k - 1) \approx 3\log_2 k + 3k + 1 \quad (17)$$

The approximately equal sign in Formulas (16) and (17) means the assumption $\lceil \log_2 k \rceil \approx \log_2 k$ and does not introduce any error when considering the most common cases of addition in 8-bit, 16-bit, 32-bit numbers, etc.

Let us estimate the parameters of the delay and area of the MAC unit shown in Figure 2 for the worst case, where b_i is not known in advance. In this case, we have

$$U_{delay}(MAC) \approx 8,8\log_2 k + 5 \quad (18)$$

$$U_{area}(MAC) \approx 3k\log_2 k + 8k^2 - 4k + 1 \quad (19)$$

The delay and area of the computational part of the FIR DF shown in Figure 1 are equal to the sum of delays and areas of MAC units, respectively. If we denote the computational part of the K -th order FIR DF with k -bit coefficients based on MAC units by $FIR_{MAC}^{K,k}$, then

$$U_{delay}(FIR_{MAC}^{K,k}) = (K + 1) \cdot U_{delay}(MAC) \approx 8,8K\log_2 k + 8,8\log_2 k + 5K + 5 \quad (20)$$

$$\begin{aligned} U_{area}(FIR_{MAC}^{K,k}) &= (K + 1) \cdot U_{area}(MAC) \approx \\ &\approx 3kK\log_2 k + 3k\log_2 k + 8k^2K + 8k^2 - 4kK - 4k + K + 1. \end{aligned} \quad (21)$$

Analysis of the derivation of Formulas (20) and (21) shows that the main part of the delay and area $FIR_{MAC}^{K,k}$ is made up of KSAs.

2.3. Proposed FIR Filter Architecture Using Truncated MAC Units

The number of KSAs in the MAC unit can be reduced to one if we use the iteration of the circuit in Figure 1 and the operation principle of the MAC unit in Figure 2. The output of each internal MAC unit in Figure 1 is fed to the input of the CSA tree of the subsequent MAC unit. Instead, numbers A and B from the previous MAC unit can be fed to the input of the adder tree of the next MAC unit, without adding them by a KSA. We call this unit a truncated MAC (TMAC); its operation principle is shown in Figure 5.

The input of each TMAC unit receives signal $X(N - i)$, filter coefficient b_i and terms A_{i-1} and B_{i-1} from previous TMAC unit output. The output of the TMAC unit is a pair of numbers A_i and B_i , which are fed to the next TMAC unit input or are added in a KSA if this TMAC unit is the last one in the FIR DF. The main difference between TMAC and MAC units is the absence of a KSA, which requires the most delay and area, and a slightly wider CSA tree, which transforms one more term.

The FIR DF scheme based on TMAC units is shown in Figure 6. Two zero signals must be fed to the inputs of the first TMAC unit, and outputs A_K and B_K of the last TMAC block must be fed to the input of the KSA.

Figure 5. The proposed truncated MAC (TMAC) unit structure.

Figure 6. The proposed architecture of a K-th order FIR DF based on TMAC units.

To describe the device shown in Figure 6 in delay and area terms, we must first find parameters U_{delay} and U_{area} of the TMAC unit:

$$U_{delay}(TMAC) \approx 6,8 \log_2 k + 1 \quad (22)$$

$$U_{area}(TMAC) \approx 8k^2. \quad (23)$$

The delay and area of the FIR DF computational part shown in Figure 6 are equal to the sum of delays and areas of the TMAC units and the KSA, respectively. If we denote the K-th order FIR DF computational part with k-bit coefficients based on TMAC units by $FIR_{TMAC}^{K,k}$, then

$$U_{delay}(FIR_{TMAC}^{K,k}) = (K+1) \cdot U_{delay}(TMAC) + U_{delay}(KSA) \approx 6,8K \log_2 k + 8,8 \log_2 k + K + 5 \quad (24)$$

$$U_{area}(FIR_{TMAC}^{K,k}) = (K+1) \cdot U_{area}(TMAC) + U_{area}(KSA) \approx 3k \log_2 k + 8k^2 K + 8k^2 + 3k + 1. \quad (25)$$

A comparison of Formulas (20), (21) and (24), (25) shows that the proposed blocks can reduce the FIR delay by about $2K \log_2 k$ and reduce its area by about $3kK \log_2 k$

3. Results

3.1. Digital Filters Theoretical Comparative Analysis

For a comparative analysis of the technical characteristics of FIR DFs based on known MAC units [28] and DFs based on proposed TMAC units, we alternately fix parameters K and k. Let us first consider the case of a 15th-order filter (i.e., $K = 15$). For the considered case, we will change the

bit width k , sorting through the most popular data formats 8, 16, 32 and 64 bits. Table 1 shows the obtained values of the parameters U_{delay} and U_{area} for the corresponding devices. After that, we fix the capacity $k = 16$ bits, and we sort through the orders 3, 7, 15 and 31 for the FIR DF. Table 2 shows the obtained values of the parameters U_{delay} and U_{area} for the corresponding devices.

Table 1. Comparison of 15th-order FIR DFs with different bit width based on known architecture [28] and based on the proposed architecture.

Bit Width, k	U_{delay}			U_{area}		
	[22]	Proposed	Difference, %	[22]	Proposed	Difference, %
8	502	352	29.86	8848	8289	6.32
16	643	463	27.99	34,832	33,009	5.23
32	784	574	26.79	136,720	131,649	3.71
64	925	685	25.95	538,640	525,633	2.41

Table 2. Comparison of FIR DFs with different orders based on known architecture [28] and based on the proposed architecture.

Filter Order, K	U_{delay}			U_{area}		
	[22]	Proposed	Difference, %	[22]	Proposed	Difference, %
3	161	125	22.39	8708	8433	3.16
7	322	238	26.12	17,416	16,625	4.54
15	643	463	27.99	34,832	33,009	5.23
31	1286	914	28.92	69,664	65,777	5.58

The data analysis obtained in Tables 1 and 2 shows that the transition from MAC units to TMAC units as the basis for FIR DF implementation can theoretically reduce filtering time by 22.39–29.86% and reduce hardware costs by 2.41–6.32%, depending on the filter order and bit width of the processed data.

3.2. Hardware Simulation of Digital Filters

Hardware simulation was performed on FPGA Artix xc7a200tffg1156-3 in Xilinx Vivado 18.3 using the very-high-speed integrated circuit (VHSIC) hardware description language (VHDL).

The goal of the simulation was to compare the technical characteristics of FIR DFs containing TMAC units with FIR DFs using traditional MAC units [28].

Results of the hardware simulation of FIR DFs are shown in Figures 7 and 8, which demonstrate that using TMAC units in the implementation of FIR DFs allowed for an increase in the devices' maximum clock frequency by 4.41–10.89%, but at the same time, the hardware costs increased: the number of used look up tables (LUTs) by 0.63–18.63% and power consumption by 1.80–27.17%. The difference between theoretical and practical results is explained by the FPGA features and the weaknesses of the “unit-gate” model, which include ignoring the effects of the load outputs capacity of individual logic units and the circuit, generally.

Our approach allows for improvements in systems where performance is critical. The approach proposed in this paper may be applied in real-time systems or other systems where performance is the main characteristic, for example in medical image processing systems. Increasing the max clock frequency of a medical tomogram processing system would allow for an increase in its performance (i.e., the number of processed frames per second).

Figure 7. Hardware simulation results of 15th-order FIR DFs with different bit width based on known architecture [28] and based on the proposed architecture: (a) frequency; (b) number of look up tables (LUTs); (c) power consumption.

Figure 8. Hardware simulation results of FIR DFs with different orders based on known architecture [28] and based on the proposed architecture: (a) frequency; (b) number of LUTs; (c) power consumption.

4. Conclusions

In this work, we developed a new FIR DF architecture based on a modified MAC unit architecture called TMAC, with the aim of increasing the digital filtering performance. Theoretical analysis of digital filter parameters was performed using the abstract “unit-gate” model. According to the theoretical analysis, FIR DF implementation based on TMAC units can theoretically reduce filtering time by 22.39–29.86% and reduce hardware costs by 2.41–6.32%. The results of the hardware simulation on a FPGA show that the use of TMAC units increased the FIR DF performance up to 10.89% but required more hardware costs compared to traditional FIR DFs using traditional MAC units. The results of this research can be used in the digital signal processing theory and for solving practical problems such as machine learning, multimedia processing, noise reduction and many others.

In future works, we plan to study the application of the proposed approach for discrete wavelet transform of medical tomograms. This type of medical image usually uses 8-, 12- or 16-bit data representation, and the wavelet filter banks have various orders. Thus, many FIR configurations discussed in this article may be applied in practice to process medical tomograms using a discrete wavelet transform.

Author Contributions: Conceptualization, P.L.; methodology, M.V.; software, M.V.; validation, N.N.; formal analysis, G.V.; investigation, G.V.; resources, N.N.; data curation, N.N.; writing—original draft preparation, G.V.; writing—review and editing, M.V.; visualization, G.V.; supervision, P.L.; project administration, P.L.; funding acquisition, P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors are grateful to the North Caucasus Federal University for supporting the competition of scientific groups and individual scientists of the North Caucasus Federal University.

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

References

1. Bhaskar, P.C.; Uplane, M.D. FPGA based digital FIR multilevel filtering for ECG denoising. In Proceedings of the 2015 International Conference on Information Processing (ICIP), Pune, India, 16–19 December 2015; pp. 733–738.
2. Kurbel, T.; Göckler, H.G.; Alfsmann, D. Oversampling Complex-Modulated Digital Filter Bank Pairs Suitable for Extensive Subband-Signal Amplification. 2009, pp. 2658–2662. Available online: <https://ieeexplore.ieee.org/document/7077454> (accessed on 19 October 2020).
3. Porshnev, S.V.; Kusaykin, D.V.; Klevakin, M. On accuracy of periodic discrete finite-length signal reconstruction by means of a Whittaker-Kotelnikov-Shannon interpolation formula. In Proceedings of the 2018 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology, Yekaterinburg, Russia, 7–8 May 2018; pp. 165–168. [CrossRef]
4. Tang, F.; Wang, Z.; Xia, Y.; Liu, F.; Zhou, X.; Hu, S.; Lin, Z.; Bermak, A. An Area-Efficient Column-Parallel Digital Decimation Filter With Pre-BWI Topology for CMOS Image Sensor. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2018**, *65*, 2524–2533. [CrossRef]
5. Kiran, S.; Shafik, A.; Tabasy, E.Z.; Cai, S.; Lee, K.; Hoyos, S.; Palermo, S. Modeling of ADC-Based Serial Link Receivers with Embedded and Digital Equalization. *IEEE Trans. Components Packag. Manuf. Technol.* **2018**, *9*, 536–548. [CrossRef]
6. Lakkadi, A.; Debrunner, L.S. Radix-4 modular pipeline fast Fourier transform algorithm. In Proceedings of the 2017 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 29 October–1 November 2017; pp. 440–444.
7. Medus, L.D.; Iakymchuk, T.; Frances-Villora, J.V.; Bataller-Mompean, M.; Rosado-Muñoz, A. A Novel Systolic Parallel Hardware Architecture for the FPGA Acceleration of Feedforward Neural Networks. *IEEE Access* **2019**, *7*, 76084–76103. [CrossRef]
8. Tan, L.; Jiang, J. *Digital Signal Processing*, 3rd ed.; Elsevier: Amsterdam, The Netherlands, 2018; ISBN 9780128150726.
9. Wang, H. A New Separable Two-dimensional Finite Impulse Response Filter Design with Sparse Coefficients. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **2015**, *62*, 2864–2873. [CrossRef]
10. Jaiswal, M.; Sharma, S.; Sharma, A. Implementation of high-speed-low-power adaptive finite impulse response filter with novel architecture. *J. Eng.* **2015**, *2015*, 86–91. [CrossRef]

11. Hatai, I.; Chakrabarti, I.; Banerjee, S. An Efficient Constant Multiplier Architecture Based on Vertical-Horizontal Binary Common Sub-expression Elimination Algorithm for Reconfigurable FIR Filter Synthesis. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **2015**, *62*, 1–10. [[CrossRef](#)]
12. Maeda, Y.; Fukushima, N.; Matsuo, H. Taxonomy of Vectorization Patterns of Programming for FIR Image Filters Using Kernel Subsampling and New One. *Appl. Sci.* **2018**, *8*, 1235. [[CrossRef](#)]
13. Kaplun, D.; Butusov, D.N.; Ostrovskii, V.; Veligosha, A.; Gulvanskii, V. Optimization of the FIR Filter Structure in Finite Residue Field Algebra. *Electronics* **2018**, *7*, 372. [[CrossRef](#)]
14. Rakesh, H.; Sunitha, G.S. Design and Implementation of Novel 32-Bit MAC Unit for DSP Applications. In Proceedings of the 2020 International Conference for Emerging Technology, Belgaum, India, 5–7 June 2020; pp. 1–6.
15. Patil, P.A.; Kulkarni, C. Multiply Accumulate Unit Using Radix-4 Booth Encoding. In Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 14–15 June 2018; pp. 1076–1080.
16. Lahari, P.; Bharathi, M.; Shirur, Y.J. An Efficient Truncated MAC using Approximate Adders for Image and Video Processing Applications. In Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), Tirunelveli, India, 15–17 June 2020; pp. 1039–1043.
17. Masadeh, M.; Hasan, O.; Tahar, S. Input-Conscious Approximate Multiply-Accumulate (MAC) Unit for Energy-Efficiency. *IEEE Access* **2019**, *7*, 147129–147142. [[CrossRef](#)]
18. Ahish, S.; Kumar, Y.; Sharma, D.; Vasantha, M. Design of high performance Multiply-Accumulate Computation unit. In Proceedings of the 2015 IEEE International Advance Computing Conference (IACC), Bangalore, India, 12–13 June 2015; pp. 915–918.
19. Camus, V.; Enz, C.; Verhelst, M. Survey of Precision-Scalable Multiply-Accumulate Units for Neural-Network Processing. In Proceedings of the 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hsinchu, Taiwan, 18–20 March 2019; pp. 57–61.
20. Yang, T.; Sato, T.; Ukezono, T. An Approximate Multiply-Accumulate Unit with Low Power and Reduced Area. In Proceedings of the 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Miami, FL, USA, 15–17 July 2019; pp. 385–390.
21. Zhang, H.; He, J.; Ko, S.-B. Efficient Posit Multiply-Accumulate Unit Generator for Deep Learning Applications. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019; pp. 1–5.
22. Howal, P.S.; Upla, K.P.; Patel, M.C. HDL implementation of digital filters using floating point vedic multiplier. In Proceedings of the IEEE International Conference on Circuits and Systems, ICCS 2017, Bangalore, India, 12–13 June 2015; pp. 274–279.
23. Spoorthi, H.R.; Narendra, C.P.; Mohan, U.C. Low Power Datapath Architecture for Multiply—Accumulate (MAC) Unit. In Proceedings of the 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 17–18 May 2019; pp. 391–395.
24. Suguna, R.; Rathinasabapathy, V. A novel high speed Low Latency Column Bit Compressed MAC architecture for Wireless Sensor Network applications. *Comput. Commun.* **2020**, *150*, 739–746. [[CrossRef](#)]
25. Parhami, B. *Computer Arithmetic: Algorithms and Hardware Designs*; Oxford University Press: Oxford, UK, 2010; ISBN 9780195328486.
26. Kogge, P.M.; Stone, H.S. A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations. *IEEE Trans. Comput.* **1973**, *22*, 786–793. [[CrossRef](#)]
27. Zimmermann, R. *Binary adder architectures for cell-based VLSI and their synthesis*; Hartung-Gorre: Konstanz, Germany, 1998; ISBN 978-3896492890.
28. Tung, C.-W.; Huang, S.-H. A High-Performance Multiply-Accumulate Unit by Integrating Additions and Accumulations into Partial Product Reduction Process. *IEEE Access* **2020**, *8*, 87367–87377. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).