Distributed Arithmetic Mechanization of Multiply and Accumulate Core for DSP Applications

Bharathi.M
Research Scholar, VTU & Assistant Professor, Department of ECE, Center for VLSI & Embedded Systems, School of Engineering and Technology, Mohan Babu University, Tirupati, Chittoor District, Andhra Pradesh, India, Email Id: bharathi.m@vidyanikethan.edu

Dr. Yasha Jyothi M Shirur
Professor of ECE, BNM Institute of Technology, Bangalore, VTU, Email Id: yashamallik@gmail.com

Abstract

This paper presents a new method for constructing a Multiply and Accumulate Unit core for Distributed Arithmetic applications in Digital Signal Processing. In applications involving digital signal processing, the MAC FIR filter core is crucial. DSP functionalities in FPGA and ASIC devices benefit from the high-speed advantages of distributed arithmetic-based MAC cores. It is an effective technique that is used to compute the inner products when creating incredibly effective MAC Cores. In this article, two models based on distributed arithmetic with seven different adders are given. These models are made for varying levels of delay and area parameters. The suggested designs are implemented in XILINX ISE and xc3s1200e-5fg320 FPGA devices based on Spartan 3E. The effectiveness of the various models is assessed and shown to be superior to the state-of-the-art of traditional technique for high speed applications. Additionally, it is demonstrated that the two models that are suggested, which use different adders, speed up processing compared to earlier studies while simultaneously achieving notable increases in power and number of slice resources.

Keywords: *DA-LUT (LUT based on a distributed algorithm), MAC (Multiply Accumulate Unit) Core, LUT-Less based DA.*

Introduction

A processor is an electronic gadget that forms certain sorts of operations. There are 3 types of processors.1) General purpose Processor(GPP) 2)Micro Processor 3)Digital signal Processor.General-purpose Processor(GPP) units are outlined to perform worldwide operations just like the CPU in our computer. A microprocessor is also an electronic device that performs arithmetic and logical operations based on the instruction set. These two are built based on von Neumann's Design. Digital Signal Processor is a special purpose electronic device that has a dedicated architecture suited for digital signal processing applications. It is utilized to analyze the real-time signals and to progress and optimize its execution based on the application.

The main features of DSPs includes dedicated arithmetic operations(Called multiply & Accumulate),DCT,IDCT,FFT,DFT.With the combination of DSP&FPGA innovation, the performance parameters can run for more validation cycles in less time. For example, FPGA Platforms to build for signal transmission in mobile communication for 4G/5G[4].At present, the trend is moving around SOC [3],[13](System on chip) which has a single platform that combines the entire system devices includes multi-core processors-CPU, DSP, Memory blocks & peripheral circuitry. Every SOC includes a dedicated DSP block to carry out software driver tasks of digital signals. The major applications of SOCs are in embedded systems- AI-based telemetry, Industrial IoT & edge computing Socs. Socs in wireless networks can perform mobile computing. One practical example is Qualcomm uses snapdragon SOCS for manufacturing Laptop computers and in many applications like medical wireless, communications which can suits for mixed signals too.

Inner product computation is one of the major tasks in DSP applications like convolution, correlation, and frequency transformations. All these applications involve computation between two signals ie., bit-serial in nature. Since DSPs suits for real-time processing applications[15], it needs fast computing hardware architectures. This paper's motto is to develop a platform to increase speed by increasing the performance of a Digital signal processor. Every DSP includes multiply and accumulate units. The most common applications in Digital Signal processor is multiplication &Accumulation.

$w(n)=p(n)+ W(n)$

where $w(n)$ is the output response,$p(n)$ is the input response & $W(n)$ be previous response.

Many researchers have developed several Distributed based MAC Architectures. In[1],[5] Mehendale et.al, proposed various techniques for optimization of area, delay & power concerning the number of bits, input bit precision, and coefficient of Distributed and offset binary coding Architectures. In[6] ye ulu et.al, adopted a new InSite error prevention to mitigate the effects on DA-based circuits from that they achieved 32% of power saving for 16 tap FIR Filter[20],[21]. DA-based resampling filter is shown in [7] in addition to that 2 real-time functions are derived in the FPGA platform for end-user application to increase efficiency. For example, Masadah[8] et.al, have designed approximate MAC to reduce power consumption by using an input aware conditional block for image processing applications.

In[9] Bhaweez Ahamed et.al, has identified that MAC operation is a backbone of LMS digital filters in a DSP core. In place of the multiplier component, they suggested a composite Distributed Arithmetic circuit architecture that can store partial products. Additionally, they developed the OBC approach and a half memory algorithm to cut the size of the memory by a factor of 4, while simultaneously sacrificing latency to increase power and throughput.Parallel DA-based designs for approximate inner product computing have been developed by Basant Kumar Mohanty et. al, in [10] which suits DSP Applications. It is based on approximate computing hence it gives less accuracy but the area-delay product can be improved. Additionally, they proposed three Parallel DA structures for the same accuracy which can be used for higher word lengths.

Low power fixed-point MAC is designed in [11],[14],[17] to increase speed and to reduce power. Their goal is to boost power by decreasing the computations for addition and taking into account multipliers, so that the architectures use less power and produce more effectively. It is noteworthy to add [12] Che-Wei tung et.al, introduced high speed and low power pipelined MAC [2],[19] for DSP applications. They integrate adder block in the multiplication process as a result delay & power can be reduced. The experimental result was shown for both signed and unsigned numbers for the alpha bit addition mechanism.

In any Digital Signal Processor, the basic building Block is a MAC Core. The name suggests it is made up of Multipliers and adders. The output of the adder is given to the accumulator. It generates partial products which consume more area to overcome this, DA will be preferred.

Existing Distributed Arithmetic

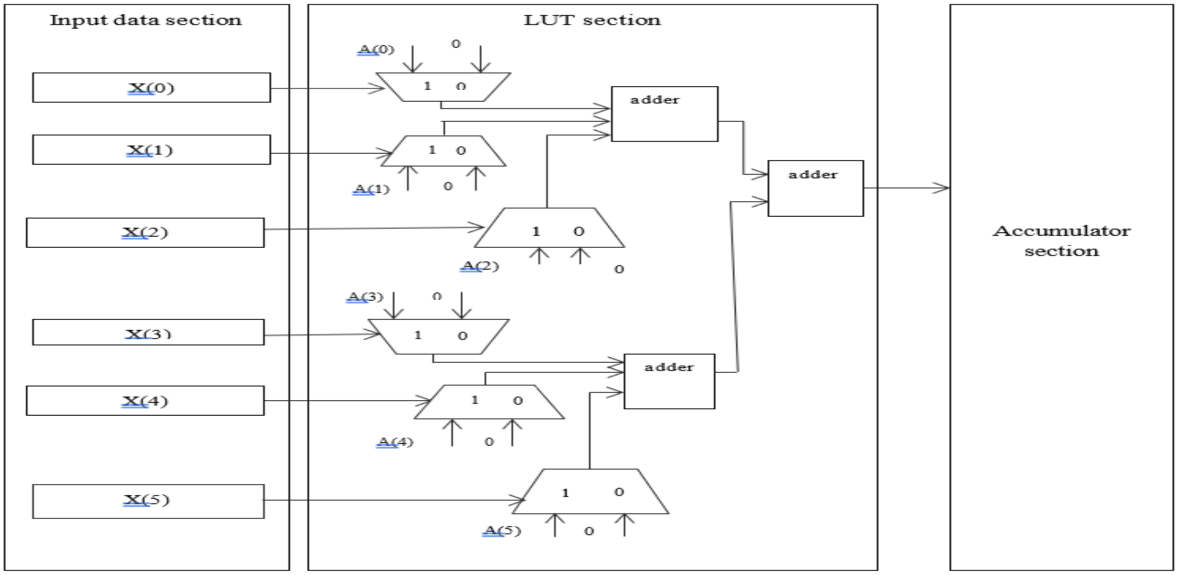
To know the proposed designs, it is necessary to present fundamental work. This section mainly discusses relevant algorithms of Distributed Arithmetic[16].

Distributed Arithmetic [18] is a key role in implementing digital signal processing functions. Distributed Arithmetic is a technique that performs the inner product between 2 vectors, mainly one of the vectors is constant and the other is a known value. First, it takes the values of input and addresses bits of the data section. LUT section generates the precomputed values based on the address bit. Here multiplication block is replaced with pre-computed LUT(Lookup table) instead of logic followed by adder and accumulator.

PROPOSED METHOD

The traditional (LUT Based) DA offers much less speed for this reason it's miles inadequate for Real-time applications. To increase speed the primary concept is to alter the adder with options together with Serial adder, Ripple Carry adder, Carry look ahead Adder, Carry select adder, carry skip adder Kogge-stone adder, and Ladner-Fischer adde. This section includes first the performance metrics hired to see the area delay product and power delay product are introduced for 16, 32 & 64 bits of Distributed Arithmetic and Offset Binary Coding architectures. Next, the speed evaluation of specific styles of Distributed architectures is designed for Single LUT-based & LUT-less-based architectures are shown: In Single LUT Architecture, the precomputations are done inside the LUT, and the output can be derived from from the contents of address bits. while in LUT-less eliminates the LUT Section and uses adders and multiplexers. The output of multiplexers is summed using an adder. Since there doesn’t have LUT they are referred to as LUT-Less Distributed architectures are also known as “Adder based DA”. Figure 1 indicates LUT-Less architecture is shown below:

Figure 1: 64 Bit Dedicated Adder based Distributed Arithmetic based MAC core



EXPERIMENTAL RESULTS:

To generate the precomputation values, the proposed LUT-less technique uses adders and multiplexers. Distributed arithmetic cores with no LUT are also known as LUT-less distributed arithmetic cores. The suggested was tested on a 64-bit MAC core to demonstrate this. Xilinx ISE of the target device XC3S1200E-5fg320 was used to simulate and synthesise the design. With an adderss of 63, created a 64 bit MAC Core with the following input set: 31,30,29,28,27,26. The identical set of inputs is used to replicate both existing (LUT) and proposed (LUT-less) Distributed arithmetic. The performance characteristics are compared to those of existing DA-based MAC cores, and it is found that the results are stable for the given set of inputs. In this paper the proposed approach is done for Offset Binary Coding Based Distributed Arithmetic cores for 16,32 & 64 bits to calculate area, power, delay,Area-Delay Product(ADP) & Power-Delay Product(PDP). The area of OBC is increased but delay and power is minimized. As we know that ,there is a trade-off between area and power, this can be used for high Speed & low power applications. The delay of 64 bit Mac core is 147.848ns where as the existing DA has 274.350ns.

In comparison to a traditional one, the speed is boosted by 45.97%. Table 1 compares the DA and OBC for LUT-less MAC cores with 16, 32, and 64 bits of memory. In comparison to LUT-based DA, the power consumption savings in LUT-less based DA is 16.59%. This method can be expanded to higher-order MAC cores, making it suitable for High-Speed Applications. For DHI, DCT, and DST transformations, this method can be used. In the future, more reconfigurable MAC cores could be added.

According to the findings, the PDP of OBC is significantly lower than that of DA. Figures 6,7, and 8 correspond to the evaluation of 16,32, and 64 bit DA and OBC based MAC Cores in terms of area, latency, and power.

Table 2 compares all designs in terms of area, delay, and power utilizing several adders, including the serial adder, ripple carry adder, carry look-ahead adder, carry choose adder, carry skip adder, kogge stone adder, and Ladner Fischer adder.

Higher-order computing relies heavily on prefix adders. The creation of customized ASIC and FPGA circuits for digital signal processing applications will benefit from the development of efficient Distributed MAC cores. Any prefix adder has three phases in general. Precomputation is dealt with in stage 1, a prefix is dealt with in stage 2, and post computation is dealt with in stage 3. Precomputation requires propagating and generating bits, while carry computation can be generated in the prefix stage, and overall sum and carry can be generated in the final stage. 2 prefix adders(Kogge stone and Ladner Fischer) are implemented in the adder block of DA based MAC cores in this paper. In the Ladner fisher adder, the area above the kogge stone is still reduced. All of these adders were designed in Verilog HDL and simulated on the XC3S1200E-5fg320 target hardware. All of the other 64-bit LUT-based architectures were also compared to LUT-less architectures. Ladner-Fischer LUT-less is the most area-efficient, occupying only 212 slices compared to 679 slices for LUT-based (68.77 percent reduction in area). In comparison to LUT-based systems, LUT-free architectures provide less delay. Delay reduction has been reported to be between 25% and 40%.

From table 1, it is evident that offset binary coding reduces delay compared with conventional DA. Table 2 gives two types of 64 bit-based DA cores that are designed and implemented. References [14-21] propose different adders.

Table 1: Implementation Results of 16,32 & 64 bit based MAC Cores using DA & OBC (for the Target device is xc3s1200e-5fg320)

Bits		Area (µm2)	Delay (ns)	Power(mW)	ADP (µm2 xns)	PDP (mw x ns)
DA	16	221	274.350	229	60,631.35	62,826.15
	32	325	274.350	231	89,163.75	63,374.85
	64	675	273.650	235	184,713.75	64,307.75
OBC	16	242	142.947	194	34,593.174	27,731.718
	32	423	135.649	197	57,379.527	26,722.853
	64	767	147.848	196	1,340,599.416	28,978.208

The proposed approach is extended and modelled for 64 bits of DA-LUT & DA-LUT-Less with various adders. Firstly, in Table:1 DA with single LUT for various adders have been presented. Secondly, LUT Vs LUT-Less models are compared in Table:2. Experimental work has been done to perform logic synthesis to minimize the delay is the most. The experimental data presented here is done with respect to simulation analysis. Table 1 tabulates the synthesis output of various MAC with different types of adders. The columns in table 1 denote area, delay, and power. Also calculated area delay product & power delay product.

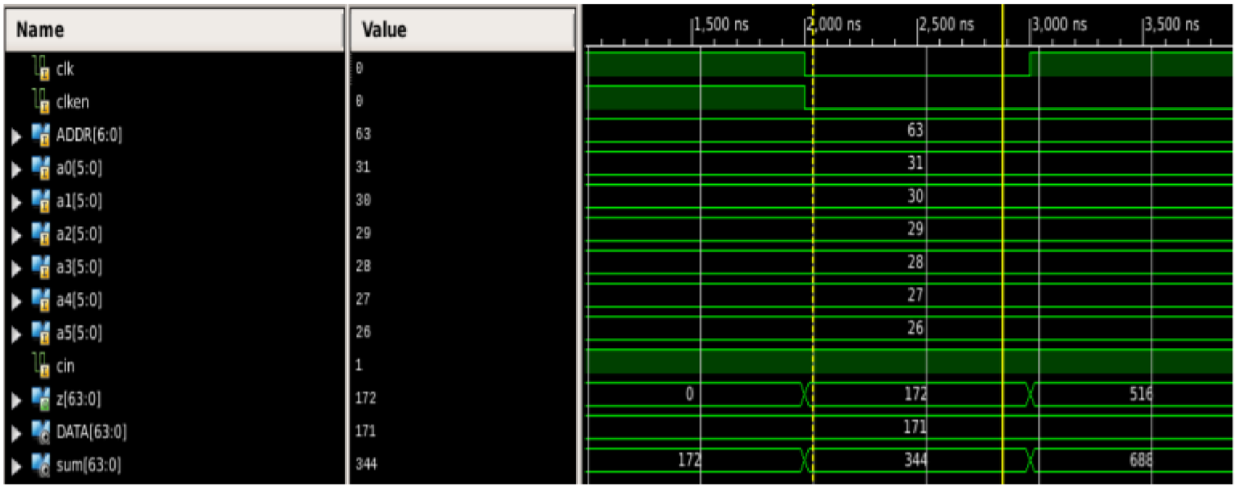
Table 2: Performance Metrics, ADP & PDP of proposed Single LUT and LUT-less based DA of various adders (for the Target device is xc3s1200e-5fg320)

Techniques		**Area (no. of slices)**	**Delay (ns)**	**Power(mW)**	
				Logic power	**Data Power**
DA-Single LUT	Serial adder	612	222.712	147	562
	Ripple carry adder	640	273.650	148	703
	Carry look ahead adder	672	273.165	150	688
	Carry skip adder	671	272.873	150	639
	Carry select adder	716	279.122	152	661
	Kogge stone adder	680	239	143	653
	Ladner Fischer adder	679	272	153	592
DA-LUTLESS	Serial adder	767	147.848	196	108
	Ripple carry adder	214	141.383	194	143

	Carry look ahead adder	185	134.621	194	138
	Carry skip adder	185	134.621	194	142
	Carry select adder	512	145.543	195	180
	Kogge stone adder	1204	134.661	202	118
	Ladner Fischer adder	212	141.962	193	102

The simulation findings below apply to LUT and LUT-Less-based 64 bit MAC cores. Created a 64 bit LUT based MAC Core with the following input set: 31,30,29,28,27,26 and an address of 63. sum=171 is the final result. If cin=1, the sum equals 172. The architecture was created in such a way that if clk (CLOCK) and clken (CLOCK ENABLE)=1, no accumulation occurs; otherwise, the output is accumulated with the prior result. In fig. 2, the prior output is 172, and the current output is 172, hence the MAC accumulates the previous value, i.e., 172+172=344.

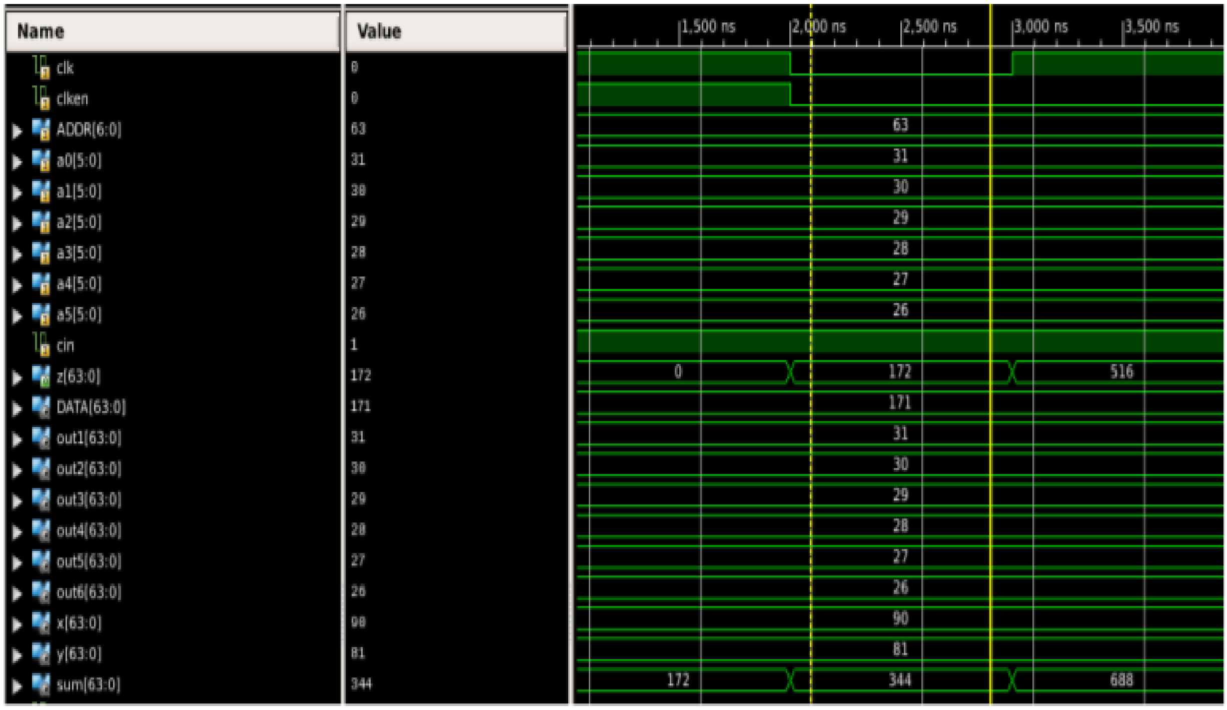
Figure 2: Simulation of DA-based implementation using single LUT.



Name	Value
clk	0
clken	0
ADDR[6:0]	63
a0[5:0]	31
a1[5:0]	30
a2[5:0]	29
a3[5:0]	28
a4[5:0]	27
a5[5:0]	26
cin	1
z[63:0]	172
DATA[63:0]	171
sum[63:0]	344

Figure 3 depicts simulation a 64-bit LUT-less MAC core with the same set of inputs as Figure 9: 31,30,29,28,27,26 and an address of 63. sum=171 is also used here. Figure 5 also features three adders. When adder 1(DATA1) and adder 2(DATA 2) are combined, the result is DATA DATA1 + DATA2 =90+81 =171, which is added together to obtain DATA DATA1 + DATA2 =171+1 = 172.

Figure 3: Simulation of DA-based implementation using LUT-Less.



Name	Value
clk	0
clken	0
ADDR[6:0]	63
a0[5:0]	31
a1[5:0]	30
a2[5:0]	29
a3[5:0]	28
a4[5:0]	27
a5[5:0]	26
cin	1
z[63:0]	172
DATA[63:0]	171
out1[63:0]	31
out2[63:0]	30
out3[63:0]	29
out4[63:0]	28
out5[63:0]	27
out6[63:0]	26
x[63:0]	90
y[63:0]	81
sum[63:0]	344

CONCLUSION:

In DSP applications, the MAC is the most widely utilized block. In ASIC and FPGA devices, distributed arithmetic and offset binary coding are critical for implementing DSP operations. Multiplication is done without the need for a multiplier, and partial products are used in lookup tables in DA. The fundamental disadvantage of DA is that with each additional input, the size of the ROM grows exponentially. In this case, OBC is reduced by 46% of delay and 16.6% of power compared with conventional DA. A novel approach of DA& OBC with multiple adders[20-28] is proposed in this study to boost the speed of FPGA implementation. It's worth noting that OBC outperforms conventional DA in terms of area, power, and delay for 16, 32, and 64 bit. LUT-based and LUT-less DA were also discussed. Area MAC methods, Power, and the trade-off for the target application. A case study of MAC with LUT and LUT-less adders with various adders shows that the design is well suited for high-speed applications, with delays reduced by 12.45 percent and 4.7 percent, respectively, as compared to conventional DA. Researchers have a lot of flexibility in developing the LUT implementation they want, and they can adjust the parameters as well. Low-power approaches can also be used to reduce power consumption while increasing the speed of different adders.

References

[1] B. K. Mohanty and P. K. Meher, An Efficient Parallel DA-Based Fixed-Width Design for Approximate Inner Product Computation, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 5,(2020) pp. 1221-1229, doi: 10.1109/TVLSI.2020.2972772.

[2] C. Tung and S. Huang, A High-Performance Multiply-Accumulate Unit by Integrating Additions and Accumulations Into Partial Product Reduction Process, in IEEE Access, vol. 8,

pp. 87367-87377, 2020, doi: 10.1109/ACCESS.2020.2992286.

[3] M. A. Akram, I. -C. Hwang and S. Ha, "Power Delivery Networks for Embedded Mobile SoCs: Architectural dvancements and Design Challenges," in IEEE Access, vol. 9, pp. 46573-46588, 2021, doi: 10.1109/ACCESS.2021.3067644.

[4] . Dakulagi, M. Alagirisamy and M. Singh, "Efficient Coherent Direction-of-Arrival Estimation and Realization Using Digital Signal Processor," in IEEE Transactions on Antennas and Propagation, vol. 68, no. 9, pp. 6675-6682, Sept. 2020, doi: 10.1109/TAP.2020.2986045.

[5] H. Belhadj Amor, C. Bernier and Z. Prikryl, "A RISC-V ISA Extension for Ultra-Low Power IoT Wireless Signal Processing," in IEEE Transactions on Computers, doi: 10.1109/TC.2021.3063027.

[6] S. Ahmad, S. G. Khawaja, N. Amjad and M. Usman, "A Novel Multiplier-Less LMS Adaptive Filter Design Based on Offset Binary Coded Distributed Arithmetic," in IEEE Access, vol. 9, pp. 78138-78152, 2021, doi:10.1109/ACCESS.2021.3083282.

[7] chauhan, D. S., Tomar, R., singh, S. A New Trans-admittance-Mode Biquad Filter Suitable for Low Voltage Operation. International Journal of Engineering, 2015; 28(12): 1738-1745.

[8] P. Lyakhov, M. Valueva, G. Valuev and N. Nagornov, "High-Performance Digital Filtering on Truncated Multiply-Accumulate Units in the Residue Number System," in IEEE Access, vol. 8, pp. 209181-209190, 2020, doi: 10.1109/ACCESS.2020.3038496.

[9] T. T. Hoang, M. Själander and P. Larsson-Edefors, "A High-Speed, Energy-Efficient Two-Cycle Multiply- Accumulate (MAC) Architecture and Its Application to a Double-Throughput MAC Unit," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 12, pp. 3073-3081, Dec. 2010, doi:10.1109/TCSI.2010.2091191.

[10] J. Garland and D. Gregg, "Low Complexity Multiply Accumulate Unit for Weight-Sharing Convolutional Neural Networks," in IEEE Computer Architecture Letters, vol. 16, no. 2, pp. 132-135, 1 July-Dec. 2017, doi: 10.1109/LCA.2017.2656880.

[11] D. Shin, W. Choi, J. Park and S. Ghosh, "Sensitivity-Based Error Resilient Techniques With Heterogeneous Multiply–Accumulate Unit for Voltage Scalable Deep Neural Network Accelerators," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 3, pp. 520- 531, Sept. 2019, doi: 10.1109/JETCAS.2019.2933862.

[12] M. T. Mahmud, M. O. Rahman, S. A. Alqahtani and M. M. Hassan, "Cooperation-Based Adaptive and Reliable MAC Design for Multichannel Directional Wireless IoT Networks," in IEEE Access, vol. 9, pp. 97518-97538, 2021, doi: 10.1109/ACCESS.2021.3093491

[13] Alexandru Amaricai, Oana Boncalo, Constantina-Elena Gavriliu“Low-precision DSP-based floating-point multiply-add fused for Field Programmable Gate Arrays” in IET Computers & Digital Techniques ,Vol 8,Issue 4,Pages 187-197,July 2014

[14] M. Masadeh, O. Hasan and S. Tahar,

"Input-Conscious Approximate Multiply-Accumulate (MAC) Unit for Energy-Efficiency," in IEEE Access, vol. 7, pp. 147129-147142, 2019, doi: 10.1109/ACCESS.2019.2946513.

[15] I. Kataeva, H. Engseth and A. Kidiyarova-Shevchenko, "Scalable Matrix Multiplication With Hybrid CMOS-RSFQ Digital Signal Processor," in IEEE Transactions on Applied Superconductivity, vol. 17, no. 2, pp.486-489, June 2007, doi: 10.1109/TASC.2007.901451.

[16] Lu, Y.; Duan, S.; Halak, B.; Kazmierski, T. A Variation-Aware Design Methodology for Distributed Arithmetic. Electronics 2019, 8, 108. https://doi.org/10.3390/electronics8010108

[17] P. Schober, M. H. Najafi and N. Taherinejad, "High-Accuracy Multiply-Accumulate (MAC) Technique for Unary Stochastic Computing," in IEEE Transactions on Computers, doi: 10.1109/TC.2021.3087027.

[18] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC Realizations of a DA-Based Reconfigurable FIR Digital Filter," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 7, pp. 511-515, July 2014, doi: 10.1109/TCSII.2014.2324418.

[19] D. J. M. Moss, D. Boland and P. H. W. Leong, "A Two-Speed, Radix-4, Serial–Parallel Multiplier," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 4, pp. 769-777, April 2019, doi:10.1109/TVLSI.2018.2883645.

[20] M Balaji, and N. Padmaja, “High-Speed Pipelined Architecture-Based Residue Number System for FIR Filter Design” Gongcheng Kexue Yu Jishu/Advanced Engineering Science, Vol. 54, Issue. 6, pp. 2056- 2066, August, 2022.

[21] Morasa, Balaji, and Padmaja Nimmagadda. 2022. “Low Power Residue Number System Using Lookup Table Decomposition and Finite State Machine Based Post Computation.” Indonesian Journal of Electrical Engineering and Computer Science 6(1): 127–34.