ISSN 2063-5346

A COMPENDIOUS REVIEW ON DIFFERENT MAC ARCHITECTURES

S. Saranya[1], Dr.K. Lakshmi Joshitha[2]

Article History: Received: : 19.04.2023 Revised: 02.05.2023 Accepted: 10.06.2023

Abstract

The need for artificial neural networks (ANN) with high computational performance and real-time processing capabilities has increased due to the rapid advances in science and technology. One of the most crucial components that makes an ANN function successfully is the Multiply Accumulate (MAC) Unit. The system's final total performance is determined by the MAC unit's performance. The basic operations often include additions and multiplications. The bulk of a processor's time and hardware resources are consumed by multiplying, compared to all other arithmetic op-erations, like as adding or subtracting. An examination of several MAC units with various multi-pliers and adders is conducted in this work. Adders are used to gather the incomplete products that are produced by multipliers. The Effective Multiply Accumulate Unit (MAC) is crucial for real-time algorithms for processing that runs at peak speed. The increase in speed of MAC designs has been the major emphasis in recent years. This study analyzes several MAC units developed till date that are fast and energy-efficient.

Keywords: *ANN, MAC, Multiplier, Adder.*

[1,2] Assistant Professor, Department of Electronics and Communication Engineering, Sri Sairam Engineering

College, Chennai, India.

[1] saranya.ece@sairam.edu.in, [2] lakshmi.ece@sairam.edu.in

DOI: 10.48047/ecb/2023.12.si12.109

1. Introduction

A division of machine learning is artificial neural networks (ANN). Artificial intelligence is made up of neurons connecting to one another at different network levels [1]. The neural network is consistently used in categorization, target recognition, medical diagnosis, and financial forecasting. When developing an ANN, multipliers are given a lot of consideration since they have an impact on the network's functionality and overall strength. The input vectors and related weights are multiplied quickly and smoothly by ANN. In order for Artificial Neural Networks to perform the necessary mathematical operations and produce the appropriate output, multiplication and accumulation units turn out to be the crucial components.

Multiply Accumulate units are used to execute the two particular categories of arithmetic operations known as multiplication and accumulation. Developing MAC units for best purposes associated to neural networks is one of the constant and extraordinarily interesting topics in the contemporary smart electronics and digital era. And furthermore, lightning fast and limited electricity units are in immensely popular, focus on high end smart applications like multimedia and audio signal processing, data and analytics, deep learning (AI), deep learning, RADAR, Li-DAR, and SONAR, as well as other defensive line applications. [2].

Listed below is the order of the review work. The Basic construction of the MAC unit is contained in Section I. Different adders that are employed to stockpile incomplete products in MAC units are presented in Section II. Many such multipliers are discussed in section III. The numerous MAC units with varied multipliers are covered in Section IV. The conclusion on the functionality of various MAC units is represented in Section V.

2. MAC Unit and Operation

A multiplier, an adder, and an aggregate register constituent of the MAC module. The multiplier is generated by the mix up of partial creation of products, partial product elimination, and send disseminate addition. To maintain any outcome of the aggregation function safe, a parallel in parallel out register could be inserted into the accumulator register. The multiplication product that is at present being generated is mixed with the earlier result that has been saved in the accumulator generated by the adder.

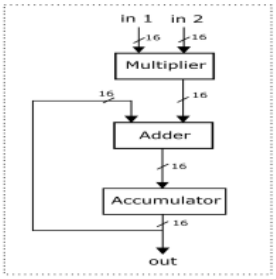

Figure 1: Basic Structure of MAC Unit

Applications for tackling multifaceted ideas in the manipulating, data processing, and among others, the MAC unit is essential. The adder receives the partial products that the multiplier generates and adds them together. The outcomes of multiplier will then be added to the total results that had been gathered. Figure 1 exemplifies the MAC unit's basic organizational structure.

The add-shift methodology, which takes up less space but has a slower processing speed and uses less power, is the foundation of the fundamental multiplication method. With low power multipliers, scientists could often reduce the space without sacrificing processing speed. Low power and fast processing are therefore required. Selecting an adder with a minimal delay, low power, and fast speed is crucial. Due to the quick development of technology, various adders have been created that offer swiftness, tiny size, reduced energy execution, or a variety of these aspects [3].

3. Adders

All adder architectures are built on the foundation of the Half Adder and Full Adder.

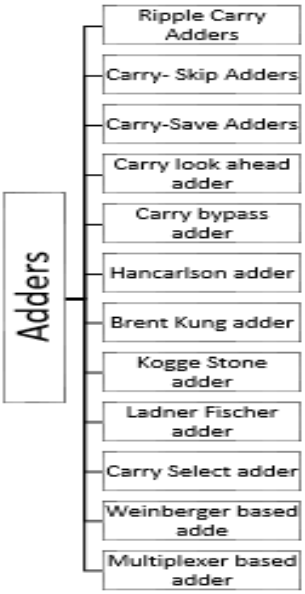

Figure 2: Types of Adders

Contingent on the rate, loss of energy, and region consumption [2,] differing adders are adopted. A major focus of VLSI design is minimizing computing time and power consumption [4]. In this research, parallel computing is used to create a Ladner Fischer-style approximation adder. In earlier technologies, arithmetic operations were performed redundantly using ripple carry adders, nonetheless, multipliers are a crucial part of any operating system that employs large adders repeatedly, therefore execution rates should be maintained to a minimum. VHDL was used to design the architecture in this article, and simulation tools from Cadence was deployed to test its performance. Then, utilizing 90 nm CMOS technology, it was constructed. The total array of cells covered by the 16-bit addition is 26%, and the dynamic power of the current design is 0.0187 W.

By leveraging parallel computing to build the carry, the Weinberger adder accelerates the addition operation. Recursion algorithm is used in the work. Recurrence is responsible for the final carry in the 32-bit Weinberger architecture. While the Weinberger adder just requires the recurrence carry to conduct calculation, typical adders require the computation of the preceding carry as well [5].

Instead of using the traditional full adder, perhaps another type of multiplexer (MUX) XOR gates are used in the progression of an optimal adder is utilized to develop an array multiplier. In this instance, a 2:1 MUX and two XOR gates are utilized. Area reduction and delays are the outcomes of this design [5].

The speed of the Russian Peasant Multiplier (RPM) increased when the Han Carlson Adder (HA) was utilized in the extension approach to enhance the performance of the RPM [6].The industry standard for high performance adders is the Kogge-stone adder, a bit parallel adder created using a carry look ahead structure with emphasis for design time. [7] [8].

By computing intermediate prefix bits together with huge prefixes, the Sklansky adder boasts a simple prefix design based on the notion of "split and conquer.". This delay is log2n steps. Latch, AND, OR, and XOR gates were used in the construction of the Sklansky adder [7] [8]. The Brent Kung adder has emerged as one of the parallel prefix adders. in accordance to the older logic, they evolve via carry propagate and carry produce signals. It has highlights like a lower expense and straightforward infrastructure. [(logn2)-2]For it, [7][8] gives the delay model.

The carry select adder (CSLA) is an adder that aids in accelerating the execution of a number of arithmetic operations. The concept of using several high speed adder logics within a conventional CSLA is used to further increase speed. The design of a hybrid Kogge Stone adder (CSLA) that also incorporates a look-ahead adder (CLA) is covered in this paper. It does so to increase speed. When compared to other CSLA structures that were already in place, the speed and power consumption were greatly improved by swapping out the RCA stages with a mix of fast adders such the kogge stone adder and CLA [9].

The latency is longest for ripple carry adder. When developing new adders, this was taken into account. The ripple carry adders have been updated by carry look ahead adders, carry bypass adders, and carry select adders. Comparing modified carry bypass adders to conventional adders, the modified carry bypass adders offer a 16.5% reduction in Xilinx-based delays and a 29.5% reduction in logical effort-based delays. Comparing the modified carry select adder to the conventional adder, the modified adders show a 45.42% reduction in Xilinx-based and a 20.4% reduction in logical effort-based delays [10].

With the use of this Distributive Arithmetic architecture, a changeable filter design may be accomplished. In this case, the distributive architecture's LUT (lookup tables) hold the total of the pipelined partial products, which are provided as input values. The design shrinks the total dimensions of the shift accumulation unit by opting for a carry-save adder in place of the adder. The establishment of the ripple carry adder greatly mitigates the magnitude of hurdles that should be triumph over with the intent to achieve. [11].

The adder is developed from reformed Boolean expressions that decrease critical route delay and do not do duplicate computations. The advised designs are created using a 65nm CMOS library and written in VHDL using the Synopsys Design Compiler. Synthesis findings reveal that The drafted 32-bit carry skip adder drops backlog, area, and battery life over the reputed carry skip adder by 10.2%, 13.6%, and 8%, respectively [12].

In Table 1 [25], the Adders Comparison is displayed. The chart demonstrates that a multiplexer-based adder is very effective in terms of region while a

Weinberger-based adder is optimal in criteria of holdup and energy that is dynamic.

Table 1: Comparison of Adders

Adder Units	No. of Slices	No. of LUTs	Delay (nS)	Dynamic Power(µW)
4-bit Modified Weinberger based	4	8	6.736	3.164
4-bit Multiplexer based	6	8	8.574	4.637
8-bit Modified Weinberger based	12	21	7.585	5.372
8-bit Multiplexer based	12	6	8.957	7.646
16-bit Modified Weinberger based	27	47	9.719	7.768
16-bit Multiplexer based	24	32	13.34	13.125
32-bit Modified Weinberger based	57	99	14.068	12.957
32-bit Multiplexer based	48	64	22.105	21.876

4. Multipliers

Array multipliers and timed or serial data multipliers are the two basic types of multipliers. Figure 3 shows the types of multipliers.

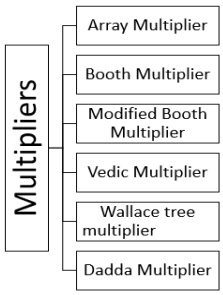

Figure 3: Types of Multipliers

Radix-4 serial booth multiplier, updated, is the proposed multiplier in [13]. By booth encoding, the recommended multiplier's multiplier section's bit count is reduced. The multiplier combines the partial items generated serially using a carry save adder. The last addition in the last phase was made with a kogge stone adder. The proposed multiplier promotes speed by 31.09% and saves power consumption by 5.31% with just a 0.25% rise in the area compared to the existing serial multiplier [13].

The goal of the most recent studies is to create and execute a 1-bit complete adder with field performance that functions as a fast, low-energy, and safe 4-bit array multiplier. The performance of the pitched 4-bit array multiplier is thoroughly examined for strength and hold up qualities at various voltages and for various scientific areas spanning from 16nm to 130nm. While mirror full-adder circuits consume 18.17 W, the proposed 16nm technological node 4-bit array multiplier costs a nominal power of 5.05 W at an activation voltage of 0.7 V. The 16nm practical node 4-bit array multiplier highlighted a small postponements of 6.37 ps at 1.2V when

deploying a full-adder based on a transmission gate [14].

The Wallace tree tactics of multiplication has incentives at the level of effectiveness." As technology evolves, more of an urge for sleek, extremely quickly wiring is being recognized. In this work, a brand-new Wallace tree multiplier structure is brought forward to boost the Wallace tree multiplier's rate without curbing its area parameter. The disclosed arrangement's final addition process for partial products is completed by parallel prefix adders. (PPAs). In the current work, five Wallace tree multiplier designs are illustrated utilizing different adders.

Using half adders and full adders, Dadda Multipliers are put to work for directing the addition of the partial output. In this case a logical AND gate has been utilized to bring about the incorrect outcomes. An additional tool for process optimization is a compressor. With a commendable speed that is significantly higher than the counterparts, this process uses relatively little power compared to other cutting-edge multipliers [16].

A well-liked 2's complement multiplier that effectively multiplies both positive and negative numbers is the Modified Baugh Wooley Multiplier. Most notably, modified Baugh Wooley multipliers have value to supervise sign bits. An updated 16-bit Baugh Wooley multiplier will be provided in this research. The provided circuit was assembled using full adders, tweaked ripple carry adders, as well as decreased power compressors. It is recreated using Cadence Virtuoso and gpdk 45nm technology. A model reveals its layout is better and uses less power than the standard wiring. The way additionally deploys a lesser amount of [17] in instead of this.

As the finest alternative to multiplying algorithms, this effort intends to construct a Vedic Multiplier utilizing Indian Vedic Mathematics technique. An element called a multiplier has a significant role in how well a high-speed CPU performs. To address these significant issues of complexity and time, we will employ the Vedic mathematics method in this project together with detector and compressor circuits. To increase speed, this paper concentrate on reducing the digital circuit's processing time. Additionally, minimizing switching activities will lower power usage. The algorithm employed is the "Urdhva-Tiryagbhyam Sutra" [18].

The intermediately developed partial products are shrunk by the radix-4 Booth technique. A larger representations radix might be employed to indicate a figure with lesser numbers. The Radix-4 technique supports in a two-fold minimization of the partial items when compared to the Radix-2 booth's method. The radix-4 technique is recommended because that conducts many factor performances more consistently than the upper radices, covering delay, area, and power [18].

A unique design for an 8-bit Radix4 modified Booth multiplier operating at 500 MHz with better lag and energy usage is exhibited in this [19] study. This design recommendations stipulates novel encoder for the first partial product collection and cuts down its wiring for the first partial product collection to only a 5-bit MUX 3:1 and simple parts than contemporary state-of-the-art designs with a 5-bit MUX 4:1. For binary inputs with LSB=0 and LSB=1, a tailored two's complement tackle is shown while unique circuits is generated for all of the cases requiring minimal valuable passageways. Through the optimal 6-bit Squareroot CS-Hybrid adders, tiny output GDI and CMOS logic-based squareroot Carry Select adders via boosted output oscillate for 8-bit products are practical.

5. MAC Models

An emphasis would be placed on the ideal building of a 32-bit Multiplier-Accumulator Unit dependent on Vedic Mathematics [20]. As a consequence of the findings, it is possible to draw the conclusion that, in comparison to the traditional multiplier, delay time may be decreased by using a multiplier built using the logic of Vedic Mathematics. So, an innovative methodology was employed in this investigation and the Vedic Multiplier was created using various adders. Subsequently, a 64-bit MAC unit was constructed using this creative Vedic Multiplier.

It is put forward to use a pipeline multiply-accumulate (MAC) architecture via quick acceleration and minimal electricity. Carry propagations of adds, incorporating additions in multiplications and additions into accumulations, often lead into substantial expenditures of energy and path lag in a normal MAC. To tackle this flaw, the partial product reduction method encountered only minimal progresses. More significance bits are not added to or collected until the PPR phase of the subsequent multiplication in the recommended MAC architecture. The overflow in the PPR process is accomplished by accumulating the overall amount of carries

through a small-size adder. Experimental findings demonstrate that the proposed MAC architecture, when used with the same timing constraint, can significantly reduce both power consumption and circuit area when compared to prior works [21].

The findings of this research specialized on the fixed-point multiply-accumulate unit's structure evaluation for rapid computation and low energy use. Several MAC blocks are combined, overlapped, and glued together to harness a 2D graphic convolution method. An augmented adder unit, a controller, and a sequence multiplier are accessories of the MAC that is brought about. The controller unit isolates the kernel pixels and enters image pixels based on similarities found, conserving power by avoiding unnecessary multiplications. The design incorporates a revolutionary concept for minimizing the additions in picture filtering procedures. Compared to the conventional approaches, the proposed MAC performed 28% more efficiently [22].

Under the time-multiplexed architecture, a hardware-efficient implementation of ANN designs employing approximate adders and multipliers is suggested. Additionally, examined to other approximate multipliers that were earlier described, the approximate multiplier that is utilized substantially diminishes the number of units of physical space and energy mandatory for the ANN design. When in contrast to ANN designs that extend specialized adders and multipliers, research findings highlight that utilizing of approximation adders and multipliers in ANN models greatly restricts computational sciences while only enhancing hardware dependability [23].

Using the Urdhva Tiryakbhyam sutra as a Vedic multiplier and the Modified Weinberger adder method as an efficient adder circuit, an unique 32-bit MAC unit has been implemented and is given in this study. Comparatively, MAC units developed employing modified Weinberger Adders and Vedic Multipliers are profitable in terms of energy and lag at the price of area [24]. Tables 2 and 3 provide a comparative analysis among many MAC models [26, 27]. The table 2 demonstrates that the Wallace tree multiplier and Carry save adder-based MAC model is efficient as far as of area, lag, and energy.

Table 2: Comparison of Different MAC Models

MAC		Cell Area(μm^2)	Delay(PS)	Power(mW)
Multiplier	Adders			
Modified Booth	Carry Look Ahead	7677	4995	1.812
Dadda	Carry Look Ahead	7904	4213	1.849
Wallace Tree	Carry Look Ahead	4398	3084	0.903
Modified Booth	Carry Select Adder	7418	4556	1.713
Dadda	Carry Select Adder	7637	4125	1.721
Wallace Tree	Carry Select Adder	5013	1890	0.657
Modified Booth	Carry Save Adder	6819	4545	1.517
Dadda	Carry Save Adder	7099	3890	1.508
Wallace Tree	Carry Save Adder	2947	1026	0.435

6. Conclusion

In this analysis, a synopsis of the diverse MAC unit designs has been delivered. It has been discovered that some multipliers were effective for both power and delay, while others were effective for both. Based on the application, Multipliers are preferred for designing MAC units. While some designs emphasize on multiplier performance, certain designs were geared at enhancing adder performance. The multiplier circuits as well as the adder stage have both been altered in some architectural designs. The work aims at identifying the design of the modified MAC unit with optimal and acceptable figures of power, delay and area for the image processing application

References

[1] Chhabra, A., & Dhanoa, J. (2020, December). A Design Approach for Mac Unit Using Vedic Multiplier. In 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE) (pp. 1-5). IEEE.

[2] Patil, P. A., & Kulkarni, C. (2018, August). A survey on multiply accumulate unit. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-5). IEEE.

[3] Jena, D., Das, S. N., Dash, T., & Ali, F. A. (2021, October). High-Speed MAC Using Integrated Vedic-Mathematics. In 2021 International Conference in Advances in Power, Signal, and Information Technology (APSIT) (pp. 1-4). IEEE.

[4] Pandey, R. (2020, November). Implementation of Approximate Adder circuit of Ladner Fischer Adder (16 bit). In 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 31-34). IEEE.

[5] Rakesh, H. M., & Sunitha, G. S. (2020, June). Design and implementation of Novel 32-bit MAC unit for DSP applications. In 2020 International Conference for Emerging Technology (INCET) (pp. 1-6). IEEE.

[6] Rao, E. J., Ramanjaneyulu, T., & Kumar, K. J. (2018, December). Advanced multiplier design and implementation using Hancarlson adder. In 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC) (pp. 1-5). IEEE.

[7] AM, A. P., Arjun, R. V., Deepaknath, K., & Gayathree, K. (2020, March). Implementation of optimized digital filter using sklansky adder and kogge stone adder. In

2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 661-664). IEEE.

[8] Akash, S., Ajeeth, M., & Radha, N. (2020, July). An Efficient Implementation of FIR Filter Using High Speed Adders for Signal Processing Applications. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 1047-1051). IEEE.

[9] Simson, A., & Deepak, S. (2021, February). Design and Implementation of High-Speed Hybrid Carry Select Adder. In 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT) (pp. 1-6). IEEE.

[10] Joy, M. C., Jimmy, A., Thomas, T. C., & Kollannur, M. I. (2020, November). Modified 16-bit Carry Select and Carry Bypass Adder Architectures for High-Speed Operations. In 2020 IEEE International Conference for Innovation in Technology (INOCON) (pp. 1-4). IEEE.

[11] Saritha, M., Radhika, C., Reddy, M. N., Lavanya, M., Karthik, A., Vijay, V., & Vallabhuni, R. R. (2021, December). Pipelined Distributive Arithmetic-based FIR Filter Using Carry Save and Ripple Carry Adder. In 2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4) (pp. 1-6). IEEE.

[12] Patel, S., Garg, B., Mahajan, A., & Rai, S. (2019, December). Area-delay efficient and low-power carry skip adder for high performance computing systems. In 2019 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS) (pp. 300-303). IEEE.

[13] Kumar, B. T., Chitiprolu, A., Reddy, G. H. K., & Agrawal, S. (2020, August). Analysis of High-Speed Radix-4 Serial Multiplier. In 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 498-503). IEEE.

[14] Praneeth, D., Varshitha, R., Prashanth, S. S., Mahesh, V., Thoutam, L. R., & Ajayan, J. (2022, April). Design and Development of Reliable Low Power High Speed 4-Bit Array Multiplier using High Performance 1-Bit Full Adders. In 2022 IEEE International Conference on Nanoelectronics, Nanophotonics, Nanomaterials, Nanobioscience & Nanotechnology (5NANO) (pp. 1-6). IEEE.

[15] Devi Ykuntam, Y., Pavani, K., & Saladi, K. (2020, July). Design and analysis of High speed wallace tree multiplier using parallel prefix adders for VLSI circuit designs. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[16] Chanda, S., Guha, K., Patra, S., Singh, L. M., Baishnab, K. L., & Paul, P. K. (2020, February). An energy efficient 32 Bit approximate Dadda multiplier. In 2020 IEEE Calcutta Conference (CALCON) (pp. 162-165). IEEE.

[17] Saha, M., & Dandapat, A. (2021, May). Modified Baugh Wooley Multiplier using Low Power Compressors. In 2021 2nd International Conference for Emerging Technology (INCET) (pp. 1-6). IEEE.

[18] Awasthy, A. (2022, February). VLSI Implementation of Multiplier and Adder Circuits with Vedic Algorithm Computation. In 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT) (pp. 1-4). IEEE.

[19] Rafiq, A., Chaudhry, S. M., Awan, K. S., & Usman, M. (2021, January). An efficient architecture of modified booth multiplier using hybrid adder. In 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST) (pp. 648-656). IEEE.

[20] Jena, D., Das, S. N., Dash, T., & Ali, F. A. (2021, October). High-Speed MAC Using Integrated Vedic-Mathematics. In 2021 International Conference in Advances in Power, Signal, and Information Technology (APSIT) (pp. 1-4). IEEE.

[21] Tung, C. W., & Huang, S. H. (2020). A high-performance multiply-accumulate unit by integrating additions and accumulations into partial product reduction process. IEEE Access, 8, 87367-87377.

[22] Samanth, R., Kedlaya, V., & Nayak, S. G. (2021). A Novel Approach to Develop Low Power MACs for 2D Image Filtering. IEEE Access, 9, 28421-28428.

[23] Nojehdeh, M. E., Aksoy, L., & Altun, M. (2020, July). Efficient hardware implementation of artificial neural networks using approximate multiply-accumulate blocks. In 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) (pp. 96-101). IEEE.

[24] Rakesh, H. M., & Sunitha, G. S. (2020, June). Design and implementation of Novel 32-bit MAC unit for DSP applications. In 2020 International Conference for Emerging Technology (INCET) (pp. 1-6). IEEE.

[25] Jagadeesh, P., Ravi, S., & Mallikarjun, K. H. (2013, March). Design of high-performance 64-bit MAC unit. In 2013 international conference on circuits, power and computing technologies (ICCPCT) (pp. 782-786). IEEE.

[26] Kumar, M. S., Kumar, D. A., & Samundiswary, P. (2014, March). Design and performance analysis of Multiply-Accumulate (MAC) unit. In 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014] (pp. 1084-1089). IEEE.

[27] Ahish, S., Kumar, Y. B. N., Sharma, D., & Vasantha, M. H. (2015, June). Design of high performance Multiply-Accumulate Computation unit. In 2015 IEEE International Advance Computing Conference (IACC) (pp. 915-918). IEEE.