## FPGA IMPLEMENTATION OF MULTIPLIER-ACCUMULATOR UNIT USING VEDIC MULTIPLIERS AND REVERSIBLE GATES

<sup>a</sup>Mangapathi Vinitha, <sup>b</sup>Kumarganesh.S<sup>\*</sup>

<sup>a</sup>PG Student, Department of ECE, *Knowledge institute of technology, Kakapalayam, Salem, Tamilnadu, India.* <sup>b</sup>Professor, Department of ECE, *Knowledge institute of technology, Kakapalayam, Salem, Tamilnadu, India.* 

## Abstract

The design of Multiplier-Accumulator (MAC) unit can be implemented by using the Vedic multiplier along with the reversible logic gates. The designing of Vedic multiplier is designed by using the new sutra called "Urdhava Triyagbhayam". The performance of the MAC operation depends on the multiplier unit and the adder units. Here the designing of a multiplier and an adder can be designed by using the reversible gates to get the high speed of operation and a Vedic multiplier is used for the higher performance, lesser area and to reduce the partial products. Nowadays reversible computing will take a preferable for low power dissipation, higher speed of operation. Here, we proposed an 8, 16, 32, 64-bit Vedic multiplier is designed by using the RCA based DKG adder and Vedic multiplier is designed by using the CSLA based DKG adder out of these the proposed DKG gate adder Based on CSLA is having the high speed of operation. The comparative analysis is carried out among the ripple carry adder (RCA), carry select adder. Finally, it has been proved that the proposed CSLA based DKG gate with Vedic multiplieradder is having the high speed of operation. The overall Simulation and synthesis process is carried out with Xilinx ISE14.7 and is dumped on the FPGA vertex7 board.

**Keywords:** Multiplier-Accumulator (MAC), Vedic multiplier, Ripple Carry Adder (RCA), CSLA

## I. INTRODUCTION

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.

The common multiplication method is "add and shift" algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, Vedic multiplier using carry look ahead adder is one of the most popular karastuba method. In this lecture we introduce the multiplication algorithms and architecture and compare them in terms of speed, area, power and combination of these metrics.

The binary multiplication also happens in same way of digit multiplication as shown in below example here by getting partial products and gates are used and we are using adder (half adder, full adder) adding the columns.

#### Array Multiplier

Although the method is simple as it can be seen from this example, the addition is done serially as well as in parallel. To improve on the delay and area the CRAs are replaced with Carry Save Adders, in which every carry and sum signal is passed to the adders of the next stage. Final product is obtained in a final adder by any fast adder (usually carry ripple adder). In array multiplication we need to add, as many partial products as there are multiplier bits.

In applications like multimedia signal processing and data mining which can tolerate error, exact computing units are not always necessary. They can be replaced with their approximate counterparts. Research on approximate computing for error tolerant applications is on the rise. Adders and multipliers form the key components in these applications. In, approximate full adders are proposed at transistor level and they are utilized in digital signal processing applications.

## WALLACE TREE Multiplier Using Adder

Ripple Carry Adder is the method used to add a greater number of additions to be performed with the carry in sand carry outs that is to be chained. Thus, multiple adders are used in ripple carry adder. It is possible to create a logical circuit using several full adders to add multiple-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is a ripple carry adder since each carry bit "ripples" to the next full adder. The proposed architecture of WALLACE multiplier algorithm using RCA is shown in Fig

Take any 3 values with the same weights and give them as input into a full adder. The result will be an output wire of the same weight.

- Partial product obtained after multiplication is taken at the first stage. The data are taken with 3 wires and added using adders and the carry of each stage is added with next two data in the same stage.
- Partial products reduced to two layers of full adders with same procedure.

At the final stage, same method of ripple carry adder method is performed and thus product terms p1 to p8 is obtained.

## **Carry Save Adder**

The carry-save adder reduces the addition of 3 numbers to the addition of 2 numbers. The propagation delay is 3 gates regardless of the number of bits. The carry-save unit consists of n full adders, each of which computes a single sum and carries bit based solely on the corresponding bits of the three input numbers. The entire sum can then be computed by shifting the carry sequence left by one place and appending a 0 to the front (most significant bit) of the partial sum sequence and adding this sequence with RCA produces the resulting n + 1-bit value. This process can be continued indefinitely, adding an input for each stage of full adders, without any intermediate

carry propagation. These stages can be arranged in a binary tree structure, with cumulative delay logarithmic in the number of inputs to be added, and invariant of the number of bits per input. The main application of carry save algorithm is, well known for multiplier architecture is used for efficient CMOS implementation of much wider variety of algorithms for high-speed digital signal processing. CSA applied in the partial product line of array multipliers will speed up the carry propagation in the array.

### II. LITERATURE VIEW

Vijay Kumar Reddy Modified High Speed Vedic Multiplier Design and Implementation The proposed research work specifies the modified version of binary Vedic multiplier using Vedic sutras of ancient Vedic mathematics. It provides modification in preliminarily implemented Vedic multiplier. The modified binary Vedic multiplier is preferable has shown improvement in the terms of the time delay and device utilization. The proposed technique was designed and implemented in Verilog HDL.For HDL simulation, modalism tool is used and for circuit synthesis, Xilinx is used. The simulation has been done for 4-bit, 8-bit,16-bit, multiplication operation. Only for 16-bit binary Vedic multiplier technique the simulation results are shown. This modified multiplication technique is extended for larger sizes. The outcomes of this multiplication technique are compared with existing Vedic multiplier techniques.

A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and Analysis of Approximate Compressors for Multiplication", Inexact (or approximate) computing is an attractive paradigm for digital processing at nanometric scales. Inexact computing is particularly interesting for computer arithmetic designs. This paper deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. These designs rely on different features of compression, such that imprecision in computation (as measured by the error rate and the so-called normalized error distance) can meet with respect to circuit-based figures of merit of a design (number of transistors, delay and power consumption). Four different schemes for utilizing the proposed approximate compressors are proposed and analyzed for a Dadda multiplier. Extensive simulation results are provided and an application of the multipliers to image processing is presented. The results show that the proposed designs accomplish significant reductions in power dissipation, delay and transistor count compared to an exact design; moreover, two of the proposed multiplier designs provide excellent capabilities for image multiplication with respect to average normalized error distance and peak signalto-noise ratio (more than 50 dB for the considered image examples).

C. Liu, J. Han, and F. Lombardi, "A Low-High-Performance Multiplier Power, with Configurable Partial Error Recovery", Proc. of IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), [Approximate circuits have been considered for error-tolerant applications that can tolerate some loss of accuracy with improved performance and energy efficiency. Multipliers are key arithmetic circuits in many such applications such as digital signal processing (DSP). In this paper, a novel multiplier with a lower power consumption and a shorter critical path than traditional multipliers are proposed for highperformance DSP applications. This multiplier leverages a newly designed approximate adder that limits its carry propagation to the nearest neighbors for fast partial product accumulation. Different levels of accuracy can be achieved through a configurable error recovery by using different numbers of most significant bits (MSBs) for error reduction. The multiplier has a low mean error distance, i.e., most of the errors are not significant in magnitude. Compared to the Wallace multiplier, a 16-bit multiplier implemented in a 28nm CMOS process shows a reduction in delay and power of 20% and up to 69%, respectively. It is shown that by utilizing an appropriate error recovery, the proposed multiplier achieves similar processing accuracy as traditional exact multipliers but with significant improvements in power and performance.

G. Zervakis, et al., "Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation" Approximate computing has received significant attention as a promising strategy to decrease power consumption of inherently error tolerant applications. In this paper, we focus on hardware-level approximation by introducing the partial product perforation technique for designing approximate multiplication circuits. We prove in a mathematically rigorous manner that in partial product perforation, the imposed errors are bounded and predictable, depending only on the input distribution. Through extensive experimental evaluation, we apply the partial product perforation method on different multiplier architectures and optimal architecture-perforation expose the configuration pairs for different error constraints. We show that, compared with the respective exact design, the partial product perforation delivers reductions of up to 50% in power consumption, 45% in area, and 35% in critical delay. In addition, the product perforation method is compared with the state-of-the-art approximation techniques, i.e., truncation, voltage over scaling, and logic approximation, showing that it outperforms them in terms of power dissipation and error.

T. Yang, T. Ukezono, and T. Sato "A Low-Power High-Speed Accuracy-Controllable Multiplier Design", Multiplication is a key fundamental function for many error-tolerant applications. Approximate multiplication is an efficient technique for trading off energy against performance and accuracy. This paper proposes an accuracy-controllable multiplier whose final product is generated by a carry-maskable adder. The proposed scheme can dynamically select the length of the carry propagation to satisfy the accuracy requirements flexibly. The partial product tree of the multiplier is approximated by the proposed tree compressor. An  $8 \times 8$  multiplier design is implemented by employing the carry maskable adder and the compressor. Compared with a conventional Wallace tree multiplier, the proposed multiplier reduced power consumption by between 47.3% and 56.2% and critical path delay by between 29.9% and 60.5%, depending on the required accuracy. Its silicon area was also 44.6% smaller. In addition, results from an image processing application demonstrate that the quality of the processed images can be controlled by the proposed multiplier design.

A. Cilardo, et al., "High-Speed Speculative Multipliers Based on Speculative Carry-Save Tree", Sacrificing exact calculations to improve digital circuit performance is at the foundation of approximate computing. In this paper, an approximate multiply-and-accumulate (MAC) unit is introduced. The MAC partial product terms are compressed by using simple OR gates as approximate counters; moreover, to further save energy, selected columns of the partial product terms are not formed. A compensation term is introduced in the proposed MAC, to reduce the overall approximation error. A MAC unit, specialized to perform 2D convolution, is designed following the proposed approach and implemented in TSMC 40nm technology in four different configurations. The proposed circuits achieve power savings more than 60%, compared to standard, exact MAC, with tolerable image quality degradation.

J. Liang, et al., "New Metrics for The Reliability Approximate and Probabilistic Adders", of Approximate/inexact computing has become an attractive approach for designing high performance and low power arithmetic circuits. Floating-point (FLP) arithmetic is required in many applications, such as digital signal processing, image processing and machine learning. Approximate FLP multipliers with variable accuracy are proposed in this paper; the accuracy and the circuit requirements of these designs are analyzed and assessed according to different metrics. It is shown that the proposed approximate FLP multiplier designs further reduce delay, area, power consumption and power-delay product (PDP) while incurring about half of the normalized mean error distance (NMED) compared with the previous designs. The proposed IFLPM24-15 is the most efficient design when considering both PDP and NMED. Case studies with three errortolerant applications show the validity of the proposed approximate designs.

### III. PROPOSED METHODOLOGY

## A. Design of a Mac by 64 X 64 Vedic Multiplier using DKG Adders

A MAC unit is a foreseeable element of a many digital signal processing (DSP) applications involving multiplications/accumulations. It is also used for performing the high-speed digital DSP systems. There are several applications in DSP including the convolution, filtering, and inner products. The discrete cosine transforms (DCT) or discrete wavelet transforms (DWT) are the nonlinear functions generally use in DSP methods. Because they are essentially accomplished by cyclic application of addition and multiplication, the overall speed of the addition and multiplication arithmetic calculations are determined by the speed of execution and the

entire calculation performance.



Figure1. MAC – Basic building block diagram

Multiplication-and-accumulate operations are distinctive for digital filters. Then, the basic functionality of the MAC unit which enables the high-speed filtering as well as other distinctive processing units for DSP applications, while the MAC unit operates totally independent of the CPU, it individually processes the data for reducing the load of the CPU. Optical communication system is one of the best applications is completely based on the DSP, which requires enormous data for fast processing of digital data. The multiplication and addition operations are also required for Fast Fourier Transform (FFT). The 64-bit MAC unit which can deal a large number of bits and it needs more amount of memory. Fundamentally it consists of the multiplier and an accumulator unit which contains the sum of previous successive product terms. The MAC unit inputs are getting from the corresponding memory location which is connected to the multiplier block.

#### **B. MAC Operation**

MAC operation is not only the key operations in DSP but also in the multi-media information processing applications and various other applications also. Already mentioned above that, the MAC [12] consists of multiplier, adder and register/accumulator. In this paper, we used a Vedic multiplier. The inputs of MAC are obtained from the corresponding memory location which is connected to the multiplier block. This is helpful for the 64-bit DSP.From the 64-bit memory location, we can connect the input. The 64-bit input is given to the multiplier, after successful computation of the multiplier it will give the output of 128-bit data; these multiplier outputs of 128-bit data is given as the input to any one of the adders which perform addition. Here we are using the three different types of adders carry save adder, kogge stone adder, and new DKG gate. Finally, it has proved that the DKG adder is having the highest speed of operation.

The MAC unit function is represented by the following equation:

$$F = \sum P_i Q_i \tag{1}$$

We can get the final output of the adder unit is 129bit that is the carry is another extra bit. The corresponding output data is connected to the accumulator register. The parallel in Parallel out (PIPO) register technique is used in the accumulator register. Because of this PIPO, the bits are enormous, and it produces the corresponding adder output values are generated in parallel, PIPO referred as the input bits are received in parallel and the corresponding out bits are also generated in parallel mode. The accumulator register output is getting from any one of the inputs to a corresponding adder. The Basic building block diagram of the MAC unit is shown in Figure 1.

	and the second	
\$78P.1	STEP 2	\$7617.8
	·····	
	····	·····
5787-4	STEP 5	STEP 6
		····
57577	STEP 0	STEP 3
	**** ****	*** *****
. alter	min	ret in
STCP 10	STOPIL	STEP 12
1000	********	
ann -	dim.	.A
\$76 <b>P</b> 18	SYGP 18	INP 15
		,
A	A	

## Figure 2. Graphical representation of Urdhwa Tiryakbhyam Sutra

### C. Reversible Gates

Reversible circuits provide a one-to-one relation between inputs and outputs; therefore, inputs can be recovered from outputs. This interesting feature results in significant power saving in digital circuits. Classical digital gates are not reversible, reversible gates should be designed as basic components to design logical reversible circuits. Well known reversible gates are Feynman, Peres and HNG. The Feynman or controlled not (CNOT) gate is frequently used in reversible circuits, since it can provide exclusive OR (XOR) as well as copy and complement of the input. Since reversible circuits do not take advantage of fan-out, this gate can be used to achieve two copies of the same input by setting the other input of the gate to the zero-logic level. Similarly, by setting the second input of the CNOT to one-logic level, we can achieve the complement of the other input.

## D. RCA based DKG Adder (Adder used in Existed Method)

Multiple DKG adder circuits can be cascaded in parallel to add an N-bit number. For an N- bit parallel adder, there must be N number of DKG circuits. A ripple carry adder is a logic circuit in which the carry-out of each DKG is the carry in of the succeeding next most significant DKG. It is called a ripple carry adder because each carry bit gets rippled into the next stage. In a ripple carry adder the sum and carry out bits of any half adder stage is not valid until the carry in of that stage occurs. Propagation delays inside the logic circuitry is the reason behind this. Propagation delay is time elapsed between the application of an input and occurrence of the corresponding output. Similarly, the carry propagation delay is the time elapsed between the application of the carry in signal and the occurrence of the carry out (Cout) signal. Circuit diagram of a 4-bit ripple carry adder is shown below. Sum out S0 and carry out Cout of the DKG1 is valid only after the propagation delay of DKG1. In the same way, Sum out S3 of the DKG4 is valid only after the joint propagation delays of DKG1 to DKG4. In simple words, the result of the ripple carry adder is valid only after the joint propagation delays of all DKG circuits inside it.



Figure 3. 4-bit RCA based DKG adder



Figure 4. 2x2 Vedic Multiplier

Figure 5 shows the 64-bit Vedic multiplier using DKG gate. With Reference to the above architecture, the realization of Vedic multiplier is very easy to be intended with the hierarchical method. To design of a 64-bit Vedic multiplier, it requires the 32-bit design of a lower-level multiplier. Further, this design requires the 16, 8, 4bit multiplier and 2- bit multiplier. Thus, it is very simple to expand the multiplier design to a high level by maintaining its own modularity.

The primary logic for the proposed operation of a 64-bit size issimply separated into the two half of 32-bit each. The lower halves of the numbers are the inputs of the 1<sup>st</sup> stage, by simply swapping the two halves of the data in 2<sup>nd</sup> and 3<sup>rd</sup> stage and finally, the upper half is for the 4<sup>th</sup> and final stage. The intermediate adder stage is also required for this design. Firstly, the adder stage is designed by using the new reversible DKG logic gate. The A and B are the two inputs which are applied in a transverse mode; it is a 64 bit of input size and themaximum of 128 bit of the result. It can add the 64-bit inputs and the sum will be saved to their corresponding registers, if any carry is generated it can be added to the next consecutive stage. To equate the total number of bits in the addition process, we can append the zeros as one of the inputs to the adder stage, then only these addition processes able to be performed without having an error. For inspecting the design of the 64-bitmultiplier we must rectify three DKG stages. At the last stage of adder, the carry is generated, whereas the final stage, the sum output is taken from the adder. From the first stage of the multiplier, we can get the LSB sum bits directly. The design and operation of a 32-bit and 64-bit multiplier arealmost the same except for the 32-bit design having the 32 input bits and for 64-bit having the 64 input bits



Figure 5. 64-bit Vedic multiplier using RCA Based DKG adder

# **3.4 CSLA based DKG Adder (adder is used in proposed design)**

The 4bit DKG adder was designed by carry select adder. The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripples carry adders), to perform the calculation twice, one time with the assumption of the carry-in being zero and the other assuming it will be one. After the two results are calculated, the correct sum, as well as the correct carry-out, is then selected with the multiplexer once the correct carry-in is known.



Figure 6. Proposed Vedic Multiplier

## IV. RESULTS AND DISCUSSIONS RTL Schematic

The RTL schematic is abbreviated as the register transfer level it denotes the blueprint of the architecture and is used to verify the designed architecture to the ideal architecture that we need development. The HDL language is used to convert the description or summery of the architecture to the working summery by use of the coding language i.e., Verilog, VHDL. The RTL schematic even specifies the internal connection blocks for better analyzing. The figure represented below shows the RTL schematic diagram of the designed architecture.





**Figure 7. (a)** MAC using RCA based DKG adder

(b) MAC using CSLA based DKG adder

The technology schematic makes the representation of the architecture in the LUT format, where the LUT is consider as the parameter of area that is used in VLSI to estimate the architecture design. the LUT is consider as an square unit the memory allocation of the code is represented in there LUT s in FPGA.





**Figure 8.** (a) MAC using RCA based DKG adder

(**b**) MAC using CSLA based DKG adder

The simulation is the process which is termed as the final verification in respect to its working whereas the schematic is the verification of the connections and blocks. The simulation window is launched as shifting from implantation to the simulation on the home screen of the tool, and the simulation window confines the output in the form of the wave forms. Here it has the flexibility of providing the different radix number systems.



Consider in VLSI the parameters treated are area, delay and power based on these parameters one can judge the one architecture to other. here the consideration of delay is considered the parameter is obtained by using the tool XILINX 14.7 and the HDL language is Verilog language.

**Table 1.** Comparison of PerformanceAnalysis of MAC

Mathadalaan	No. of	Delay	Power
Methodology	LUTs	(ns)	( <b>mW</b> )
MAC using CSLA	10830	37.81	15525 46
based DKG Adder	10650		15525.40
MAC using RCA	11507	56.23	15526.21
based DKG Adder	11507		15520.21
Vedic Multiplier	10757	49.32	15621.12
with Kogge Stone			
Adder			
Vedic Multiplier	10688	56.67	15546 56
and reversible logic	10000		15540.50

## V. CONCLUSION

The results are obtained from the proposed DKG adder gate design using a Vedic multiplier with reversible computing are relatively good. The proposed 64- bit MAC unit is successfully designed with Vedic multiplier using RCA and CSLA using DKG reversible logic. it has been proved that the design is optimized in terms of total delay. We are successfully designed all the 64-bit MAC architecture of fundamental analyzed for all the existing blocks. Hence, we can prove that the Urdhava Triyagbhayam sutra with 64-bit MAC Unitand the reversible logic concept is the finest in terms of total delay aspect is shown in table 1. The overall Simulation and synthesis process is successfully carried out with Xilinx ISE

14.7 The design parameters of any architecture completely depend on the basic building blocks. For our proposed design MAC, the basic building blocks are a multiplier and the adder. In future, these basic building block designs are designed highly optimized than our proposed design obviously, it leads to reduce the total delay in the MAC architecture.

## REFERENCES

- R. Anitha1, Neha Deshmukh, Sarat Kumar Sahoo, S. Prabhakar Karthikeyan," A 32 BIT MAC Unit Design Using Vedic Multiplier and Reversible Logic Gate" International Conference on Circuit, Power and Computing Technologies [ICCPCT].
- [2] Ramalatha, M. Dayalan, K D Dharani, P Priya, and S Deoborah, High Speed Energy Efficient ALU design using Vedic multiplication techniques, International.
- [3] Sree Nivas A and Kayalvizhi N. Article: Implementation of Power Efficient Vedic Multiplier. International Journal of Computer Applications 43(16):21-24, April 2012. Published by Foundation of Computer Science, New York, USA.
- [4] Vaijyanath Kunchigi, Linganagouda Kulkarni, Subhash Kulkarni, High Speed and Area Efficient Vedic Multiplier, International Conference on Devices, Circuits and Systems (ICDCS), 2012.
- [5] D.P. Vasudevan, P.K. Lala, J. Di, and J.P. Parkerson, "Reversible logic design with online testability", IEEE Trans. on Instrumentation and Measurement, vol.55, no.2, pp.406-414, April 2006.
- [6] Prabir Saha, Arindam Banerjee, Partha Bhattacharyya, AnupDandapat, High Speed ASIC Design of Complex Multiplier
- [7] Raghava Garipelly, P. Madhu Kiran, A. Santhosh Kumar A Review on Reversible Logic Gates and their Implementation International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 3, March 2013.
- [8] www.vedicmaths.org/

- [9] Asmita Haveliya, A Novel Design for High-Speed Multiplier for Digital Signal Processing Applications (Ancient Indian Vedic mathematics approach), International Journal of Technology and Engineering System (IJTES), Vol.2, No.1, Jan -March 2011.
- [10] Aniruddha Kanhe, Shishir Kumar Das and Kumar Singh, Ankit Design and Implementation of Low Power Multiplier Using Vedic Multiplication Technique, (IJCSC) International Journal of Computer Science and Communication Vol. 3, No. 1, January-June 2012. pp. 131-132 International Journal of Scientific and Research Publications, Volume 3, Issue 2, February 2013 ISSN 2250-315.
- [11] A. Abdelgawad, Magdy Bayoumi," High Speed and Area- Efficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications", IEEE Int. Symp. Circuits Syst. (2007)3199–3202.
- [12] Fatemeh Kashfi, S. Mehdi Fakhraie, Saeed Safari," Designing an ultra-high-speed multiply-accumulate structure", Microelectronics Journal 39 (2008) 1476– 1484.
- [13] P. A. Patil and C. Kulkarni, "A survey on multiply accumulate unit," in Proc. 4th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA), Pune, India, Aug. 2018, pp. 1– 5.
- P. F. Stelling and V. G. Oklobdzija,
  "Implementing multiply-accumulate operation in multiplication time," in Proc. 13th IEEE Sympsoium Comput. Arithmetic, Asilomar, CA, USA, Jul. 1997, pp. 99–106.
- [15] A. Abdelgawad and M. Bayoumi, "High speed and area-efficient multiply accumulate (MAC) unit for digital signal processing applications," in Proc. IEEE Int. Symp. Circuits Syst., New Orleans, LA, USA, May 2007, pp. 3199–3202.
- [16] M. D. Ercegovac and T. Lang, Digital Arithmetic. San Mateo, CA, USA: Morgan Kaufmann, 2003.
- [17] T. T. Hoang, M. Sjalander, and P. Larsson-Edefors, "A high-speed, energy efficient two-cycle multiply-accumulate (MAC) architecture and its application to a doublethroughput MAC unit," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 12,

pp. 3073-3081, Dec. 2010.

- [18] B. Liebig, J. Huthmann, and A. Koch, "Architecture exploration of high performance floating-point fused multiplyadd units and their automatic use in highlevel synthesis," in Proc. IEEE Int. Symp. Parallel Distribution. Process., Workshops PhD Forum, Cambridge, MA, USA, May 2013, pp. 134–143.
- [19] A. Wahba and H. Fahmy, "Area efficient and fast combined Binary/Decimal floating point fused multiply add unit," IEEE Trans. Comput., vol. 66, no. 2, pp. 226–239, Feb. 2017.
- [20] P. Aliparast, Z. D. Koozehkanani, and F. Nazari, "An ultra-high-speed digital 4-2 compressor in 65-nm CMOS," Int. J. Comput. Theory Eng., vol. 5, no. 4, pp. 593– 597, Aug. 2013.
- [21] C. P. Narendra and K. M. R. Kumar, "Low power compressor-based MAC architecture for DSP applications," in Proc. IEEE Int. Conf. Signal Process., Informa., Commun. Energy Syst. (SPICES), Kozhikode, India, Feb. 2015, pp. 1–5.
- [22] A. Rezai and P. Keshavarzi, "Highthroughput modular multiplication and exponentiation algorithms using Multibit-Scan–Multibit-Shift technique," IEEE Trans. Very Large-Scale Integer. (VLSI) Syst., vol. 23, no. 9, pp. 1710–1719, Sep. 2015.
- [23] A. Rezai and P. Keshavarzi, "Highperformance scalable architecture for modular multiplication using a new digitserial computation," Microelectron. J., vol. 55, pp. 169–178, Sep. 2016.
- [24] A. Rezai and P. Keshavarzi, "Compact SD: A new encoding algorithm and its application in multiplication," Int. J. Comput. Math., vol. 94, no. 3, pp. 554–569, Mar. 2017.
- [25] D. Nguyen, D. Kim, and J. Lee, "Double MAC: Doubling the performance of convolutional neural networks on modern FPGAs," in Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE), Lausanne, Switzerland, Mar. 2017, pp. 890–893.
- [26] M. Sjalander and P. Larsson-Edefors, "High-speed and low-power multipliers using the Baugh–Wooley algorithm and HPM reduction tree," in Proc. 15th IEEE

Int. Conf. Electron., Circuits Syst., St. Julien's, Malta, Aug. 2008, pp. 33–36.

- [27] L.-D. Van and J.-H. Tu, "Power-efficient pipelined reconfigurable fixed width Baugh–Wooley multipliers," IEEE Trans. Comput., vol. 58, no. 10, pp. 1346–1355, Oct. 2009.
- [28] K. Tsoumanis, S. Xydis, C. Efstathiou, N. Moschopoulos, and K. Pekmestzi, "An optimized modified booth recorder for efficient design of the add-multiply operator," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 4, pp. 1133–1143, Apr. 2014.
- [29] F. Moldovan, "Partitioning and mapping algorithms into fixed size systolic arrays," IEEE Trans. Comput., vol. C-35, no. 1, pp. 1–12, Jan. 1986.
- [30] N. Petkov, Systolic Parallel Processing. New York, NY, USA: Elsevier, 1992.