

DESIGN AND IMPLEMENTATION OF A MAC USING MODIFIED BOOTH RECORDER

CH.Bharathi

M.Tech., VLSI Design, Narasaraopeta Institute Of Technology, Narasaraopeta

ABSTRACT: Add-Multiply operator is an operating unit which performs addition and then add the result to the input of the multiplier. It increases both area and critical path delay of the circuit, since the adder of AM unit inserts delay in its critical path and increases area occupation and power consumption. Hence the AM operator is optimized by employing fusion techniques. For the Design optimization of Fused- Add Multiply (FAM) unit recoding techniques are introduced. These techniques are used to implement direct recoding of the sum of two numbers in its modified booth form. Modified booth is a prevalent form used in multiplication; it reduces the number of partial products into half. In order to improve the performance of FAM unit different schemes of recoding technique called sum to modified booth recoding technique(S-MB) are introduced. By comparing the proposed recoding schemes with existing one the proposed technique yields considerable reductions in terms of delay, hardware complexity and power consumption of the FAM unit.

Keywords: Computer arithmetic, multiplication by constants, common sub expressions sharing, Add-Multiply operation, arithmetic circuits, Modified Booth recoding, VLSI design.

I. INTRODUCTION

In this paper, we study the various parallel MAC architectures and then implement a design of parallel MAC based on some booth encodings such as radix-2 booth encoder and some final adders such as CLA, Kogge stone adder and then compare their performance characteristics. In general, a multiplier uses Booth algorithm and an array of full adders, this multiplier mainly consists of three parts Wallace tree, to add partial products, booth encoder and final adder. Recent research activities in the field of arithmetic optimization [1], [2] have shown that the design of arithmetic components combining operations which share data, can lead to significant performance improvements. Based on the observation that an addition can often be subsequent to a multiplication (e.g., in symmetric FIR filters), the Multiply-Accumulator (MAC) and Multiply-Add (MAD) units were introduced [3] leading to more efficient implementations of DSP algorithms compared to the conventional ones, which use only primitive resources [4]. Several architectures have been proposed to optimize the performance of the MAC operation in terms of area occupation, critical path delay or power consumption [5]–[7]. As noted in [8], MAC components increase the flexibility of DSP datapath synthesis as a large set of arithmetic operations can be efficiently mapped onto them. Except the

MAC/MAD operations, many DSP applications are based on Add-Multiply (AM) operations (e.g., FFT algorithm [9]). The straightforward design of the AM unit, by first allocating an adder and then driving its output to the input of a multiplier, increases significantly both area and critical path delay of the circuit. Targeting an optimized design of AM operators, fusion techniques [10]–[13], [23] are employed based on the direct recoding of the sum of two numbers (equivalently a number in carry-save representation [14]) in its Modified Booth (MB) form [15]. Thus, the carry-propagate (or carry-look-ahead) adder [16] of the conventional AM design is eliminated resulting in considerable gains of performance. Lyu and Matula [10] presented a signed-bit MB recoder which transforms redundant binary inputs to their MB recoding form. A special expansion of the preprocessing step of the recoder is needed in order to handle operands in carry-save representation. In [12], the author proposes a two-stage recoder which converts a number in carry-save form to its MB representation. The first stage transforms the carry-save form of the input number into signed-digit form which is then recoded in the second stage so that it matches the form that the MB digits request. Recently, the technique of [12] has been used for the design of high performance flexible coprocessor architectures targeting the computationally intensive DSP applications [17]. Zimmermann and Tran [13] present an optimized design of [10] which results in improvements in both area and critical path. In [23], the authors propose the recoding of a redundant input from its carry-save form to the corresponding borrow-save form keeping the critical path of the multiplication operation fixed. Although the direct recoding of the sum of two numbers in its MB form leads to a more efficient implementation of the fused Add-Multiply (FAM) unit compared to the conventional one, existing recoding schemes are based on complex manipulations in bit-level, which are implemented by dedicated circuits in gate-level. This work focuses on the efficient design of FAM operators, targeting the optimization of the recoding scheme for direct shaping of the MB form of the sum of two numbers (Sum to MB – S-MB). More specifically, we propose a new recoding technique which decreases the critical path delay and reduces area and power consumption. The proposed S-MB algorithm is structured, simple and can be easily modified in order to be applied either in signed (in 2's complement representation) or unsigned numbers, which comprise of odd or even number of bits. We explore three alternative schemes of the proposed S-MB approach using conventional and signed-bit Full Adders (FAs) and Half Adders (HAs) as building blocks. We

evaluated the performance of the proposed S-MB technique by comparing its three different schemes with the state-of-the-art recoding techniques [12], [13], [23]. Industrial tools for RTL synthesis [18] and power estimation [19] have been used to provide accurate measurements of area utilization, critical path delay and power dissipation regarding various bit-widths of the input numbers. We show that the adoption of the proposed recoding technique delivers optimized solutions for the FAM design enabling the targeted operator to be timing functional (no timing violations) for a larger range of frequencies. Also, under the same timing constraints, the proposed designs deliver improvements in both area occupation and power consumption, thus outperforming the existing S-MB recoding solutions.

II. DIFFERENT MULTIPLIERS

A. Binary Multiplication

In the binary number system the digits, called bits, are limited to the set. The result of multiplying any binary number by a single binary bit is either 0, or the original number. This makes forming the intermediate partial-products simple and efficient. Summing these partial-products is the time consuming task for binary multipliers. One logical approach is to form the partial-products one at a time and sum them as they are generated. Often implemented by software on processors that do not have a hardware multiplier, this technique works fine, but is slow because at least one machine cycle is required to sum each additional partial-product. For applications where this approach does not provide enough performance, multipliers can be implemented directly in hardware.

B. High-Speed Booth Encoded Parallel Multiplier Design:

Fast multipliers are essential parts of digital signal processing systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors today, especially since the media processing took off. In the past multiplication was generally implemented via a sequence of addition, subtraction, and shift operations. Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product. In most computers, the operand usually contains the same number of bits.

III. FUSED AM IMPLEMENTATION

A. Arithmetic Circuits

Arithmetic circuits are the most natural and standard model for computing polynomials. In this model the inputs are variables x_1, \dots, x_n , and the computation is performed using the arithmetic operations $+$, \times and may involve constants from a field F . The output of an arithmetic circuit is thus a polynomial (or a set of polynomials) in the input variables. The complexity measures associated with such circuits are size and depth which capture the number of operations and the maximal distance between an input and an output,

respectively.

B. Modified Booth Form

The modified-Booth algorithm is extensively used for high-speed multiplier circuits. Architecture of the modified Booth multiplier. It is possible to reduce the number of partial products by half, by using this technique of Radix 4 Booth encoding. The basic idea is that, instead of shifting and adding for every column of multiplier term and multiplying by $+1, +2, \text{ or } 0$, we only take every second column, and multiply by $+1, +2, \text{ or } 0$, to obtain the same results.

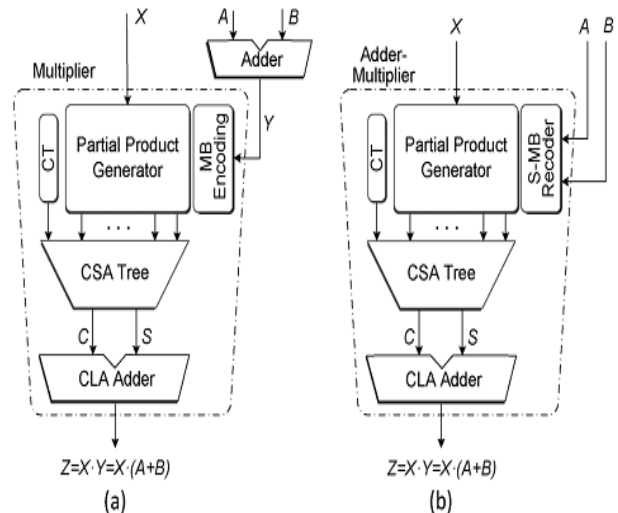


Fig. 1. AM operator based on the (a) conventional design and (b) fused design with direct recoding of the sum of A and in its MB representation. The multiplier is a basic parallel multiplier based on the MB algorithm. The terms CT, CSA Tree and CLA Adder are referred to the Correction Term, the Carry-Save Adder Tree and the final Carry-Look-Ahead Adder of the multiplier.

The multiplicand Y can be represented in MB form as:

$$Y = \langle y_{n-1} y_{n-2} \dots y_1 y_0 \rangle_{2^s} = -y_{2k-1} + i_{.2i} \langle y_{k-1} y_{k-2} \dots y_1 y_0 \rangle_{MB} = jMB. 22j = y_{2j+1} + y_{2j} + y_{2j-1} y_j^{MB} = (-1)^s * [one_j + 2 * two_j].$$

C. FAM Implementation:

Let us consider the product $X.Y$. The term $Y = \langle y_{n-1} y_{n-2} \dots y_1 y_0 \rangle_{2^s}$ is encoded based on the MB algorithm (Section II.B) and multiplied with Both X and Y consist of $n=2k$ bits and are in 2^s complement form. Equation (4) describes the generation of the k partial products:

$$PP_j = X \cdot y_j^{MB} = \bar{p}_{j,n} 2^{jn} + \sum_{i=0}^{n-1} p_{i,i} \cdot 2^i. \tag{4}$$

The generation of the i -th bit of the $P_{j,i}$ partial product PP_j is based on the next logical expression while Fig. 3 illustrates its implementation at gate level

TABLE I
 MODIFIED BOOTH ENCODING TABLE.

Binary			y_j^{MB}	MB Encoding			Input Carry $c_{in,j}$
y_{2j+1}	y_{2j}	y_{2j-1}		$sign=s_j$	$\times 1=one_j$	$\times 2=two_j$	
0	0	0	0	0	0	0	0
0	0	1	+1	0	1	0	0
0	1	0	+1	0	1	0	0
0	1	1	+2	0	0	1	0
1	0	0	-2	1	0	1	1
1	0	1	-1	1	1	0	1
1	1	0	-1	1	1	0	1
1	1	1	0	1	0	0	0

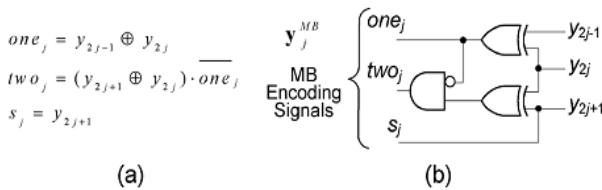


Fig. 2. (a) Boolean equations and (b) gate-level schematic for the implementation of the MB encoding signals.

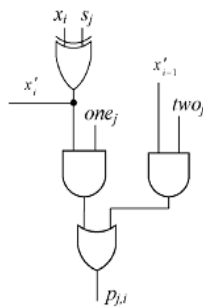


Fig. 3. Generation of the i -th bit $p_{j,i}$ of the partial product P^i_j for the conventional MB multiplier.

IV. SUM TO MODIFIED BOOTH RECODING TECHNIQUE(S-MB)

A. Signed-Bit Full Adders and Half Adders for Structured Signed Arithmetic

More specifically, in this work, we use two types of signed HAs which are referred as HA* and HA**. Tables II - IV are their truth tables and in Fig. 4 we present their corresponding Boolean equations. Considering that p, q are the binary inputs C and S , are the outputs (carry and sum respectively) of a HA* which implements the relation $2.C-S = p + q$. where the sum is considered negatively signed (Table II, Fig. 4(a)), the output takes one of the values $\{0,+1,+2\}$ In Table III, we also describe the dual implementation of HA* where we inverted the signs of all inputs and outputs and, consequently, changed the output values to $\{-2,-1,0\}$. Table IV and Fig. 4(b) show the operation and schematic of HA** which implements the relation $2.C-S = -p+q$ and manipulates a negative (p) and a positive (q) input resulting in the output value $\{-1,0,+1\}$. Also, we design two types of signed FAs which are presented in Table V and VI and Fig. 5. The schematics drawn in Fig. 5(a) and (b) show the relation of FA* and FA** with the conventional FA.

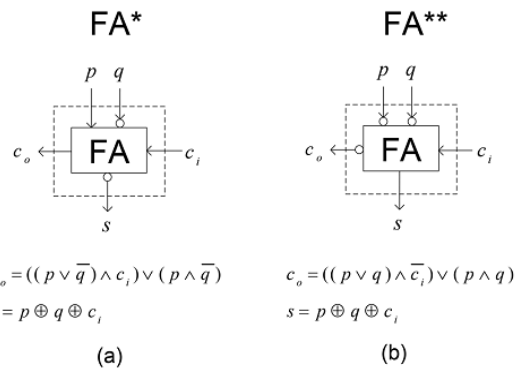


Fig.5. Boolean equations and schematics for signed (a) FA* and (b) FA**.

B. Proposed S-MB Recoding Techniques

In order to design and explore three new alternative schemes of the S-MB recoding technique. Each of the three schemes can be easily applied in either signed (2's complement representation) or unsigned numbers which consist of odd or even number of bits

1) S-MB1 Recoding Scheme: The first scheme of the proposed recoding technique is referred as S-MB1 and is illustrated in detail in Fig. 6 for both even (Fig. 6(a)) and odd (Fig. 6(b)) bit-width of input numbers. As can be seen in Fig. 6, the sum of A and B is given by the next relation:

$$Y = A + B = y_k \cdot 2^{2k} + \sum_{j=0}^{k-1} y_j^{MB} \cdot 2^{2j}$$

where $y_j^{MB} = -2s_{2j+1} + s_{2j} + c_{2j}$. (8)

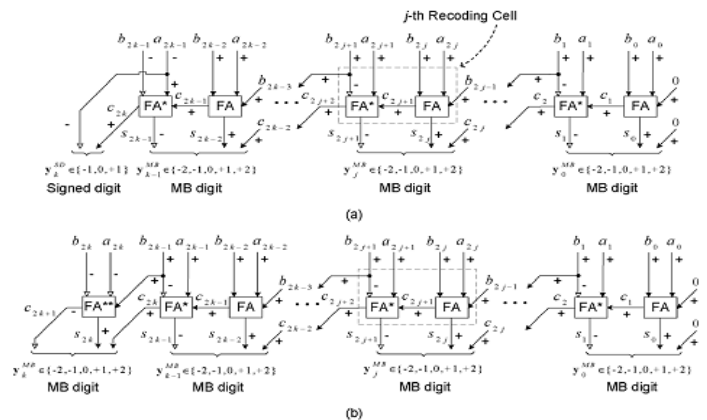


Fig. 6. S-MB1 recoding scheme for (a) even and (b) odd number of bits.

TABLE II
 HA* BASIC OPERATION.

Inputs		Output Value ¹	Outputs	
$p (+)$	$q (+)$		$c (+)$	$s (-)$
0	0	0	0	0
0	1	+1	1	1
1	0	+1	1	1
1	1	+2	1	0

¹ Output Value = $2 \cdot c - s = p + q$

TABLE III
 HA* DUAL OPERATION.

Inputs		Output	Outputs	
$p (-)$	$q (-)$	Value ²	$c (-)$	$s (+)$
0	0	0	0	0
0	1	-1	1	1
1	0	-1	1	1
1	1	-2	1	0

² Output Value = $-2 \cdot c + s = -p - q$

TABLE IV
 HA** OPERATION.

Inputs		Output	Outputs	
$p (-)$	$q (+)$	Value ³	$c (+)$	$s (-)$
0	0	0	0	0
0	1	+1	1	1
1	0	-1	0	1
1	1	0	0	0

³ Output Value = $2 \cdot c - s = -p + q$

TABLE V
 FA* OPERATION.

Inputs			Output	Outputs	
$p (+)$	$q (-)$	$c_i (+)$	Value ¹	$c_o (+)$	$s (-)$
0	0	0	0	0	0
0	0	1	+1	1	1
0	1	0	-1	0	1
0	1	1	0	0	0
1	0	0	+1	1	1
1	0	1	+2	1	0
1	1	0	0	0	0
1	1	1	+1	1	1

¹ Output Value = $2 \cdot c_o - s = p - q + c_i$

TABLE VI
 FA** OPERATION.

Inputs			Output	Outputs	
$p (-)$	$q (-)$	$c_i (+)$	Value ²	$c_o (-)$	$s (+)$
0	0	0	0	0	0
0	0	1	+1	0	1
0	1	0	-1	1	1
0	1	1	0	0	0
1	0	0	-1	1	1
1	0	1	0	0	0
1	1	0	-2	1	0
1	1	1	-1	1	1

² Output Value = $-2 \cdot c_o + s = -p - q + c_i$

2) SMB recoding scheme 2

The second approach of the proposed recoding technique, S-MB2, is described in Fig.5 for even and odd bit-width of input numbers. It consider the initial values $c_{0,1} = 0$ and $c_{0,2} = 0$. The digits, $0 \leq j \leq k-1$, are formed based on s_{2j+1} , s_{2j} and $c_{2j,2}$. As in the S-MB1 recoding scheme, it uses a conventional FA to produce the carry c_{2j+1} and the sum s_{2j} . The inputs of the FA are a_{2j} , b_{2j} and $c_{2j,1}$. The bit $c_{2j,1}$ is the output carry of a conventional HA which is part of the (j-1) recoding cell and has the bits a_{2j-1} , b_{2j-1} as inputs. The bit s_{2j+1} is the output sum of a HA* in which it drives c_{2j+1}

and the sum produced by a conventional HA with the bits a_{2j+1} , b_{2j+1} as inputs. The HA* is used in order to produce the negatively signed sum s_{2j+1} and its outputs

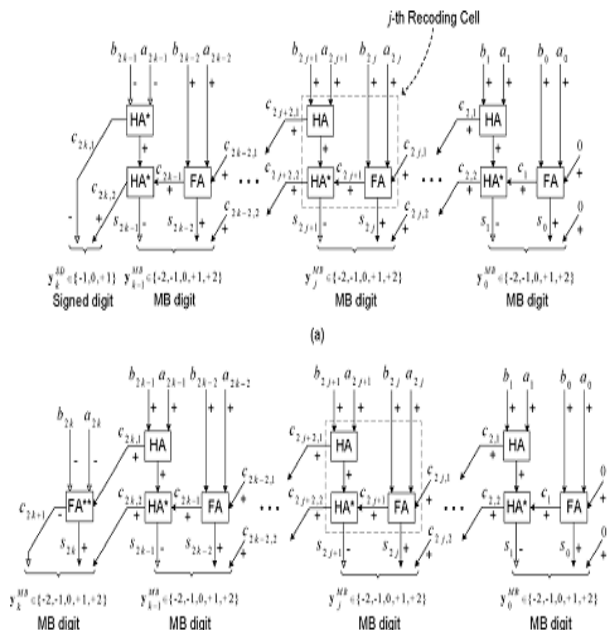


Fig. 7. S-MB2 recoding scheme for (a) even and (b) odd number of bits.

3) SMB 3 recoding scheme

The third scheme implementing the proposed recoding technique is S-MB3. It is illustrated in detail in for even and odd bit-width of input numbers. It consider that $c_{0,1} = 0$ and $c_{0,2} = 0$. It builds the digits, $0 \leq j \leq k-1$, based on s_{2j+1} , s_{2j} and $c_{2j,2}$. Once more, it uses a conventional FA to produce the carry c_{2j+1} and the sum s_{2j} . The bit $c_{2j,1}$ is now the output carry of a HA* which belongs to the (j-1) recoding cell and has the bits a_{2j-1} , b_{2j-1} as inputs. The negatively signed bit s_{2j+1} is produced by a HA** in which drive c_{2j+1} and the output sum (negatively signed) of the HA** of the recoding cell with the bits a_{2j+1} , b_{2j+1} as inputs.

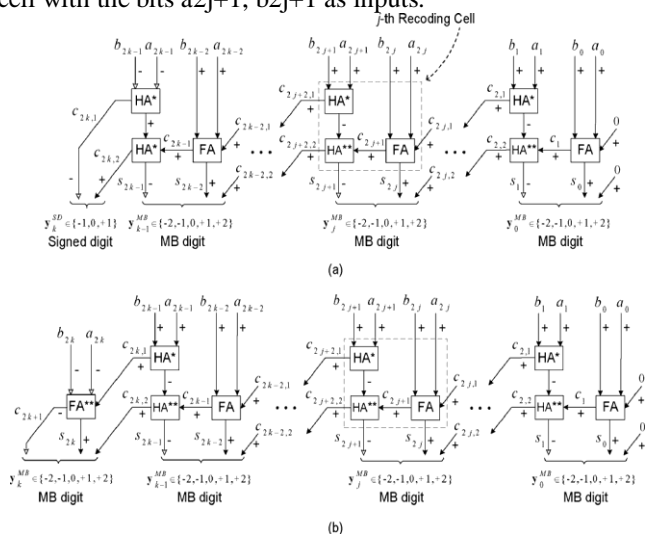


Fig. 8. S-MB3 recoding scheme for (a) even and (b) odd number of bits.

V. CONCLUSION

A 16x16 multiplier-accumulator (MAC) is presented in this work. A RADIX 4 Modified Booth multiplier circuit is used for MAC architecture. Compared to other circuits, the Booth multiplier has the highest operational speed and less hardware count. The basic building blocks for the MAC unit are identified and each of the blocks is analyzed for its performance. Power and delay is calculated for the blocks. 1-bit MAC unit is designed with enable to reduce the total power consumption based on block enable technique. Using this block, the N-bit MAC unit is constructed and the total power consumption is calculated for the MAC unit. The power reduction techniques adopted in this work. The MAC unit designed in this work can be used in filter realizations for High speed DSP applications.

REFERENCES

- [1] A. Amaricai, M. Vladutiu, and O. Boncalo, "Design issues and implementations for floating-point divide-add fused," IEEE Trans. Circuits Syst. II-Exp. Briefs, vol. 57, no. 4, pp. 295-299, Apr. 2010.
- [2] E. E. Swartzlander and H. H. M. Saleh, "FFT implementation with fused floating-point operations," IEEE Trans. Comput., vol. 61, no. 2, pp. 284-288, Feb. 2012.
- [3] J. J. F. Cavanagh, Digital Computer Arithmetic. New York: McGraw-Hill, 1984.
- [4] S. Nikolaidis, E. Karaolis, and E. D. Kyriakis-Bitzaros, "Estimation of signal transition activity in FIR filters implemented by a MAC architecture," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 19, no. 1, pp. 164-169, Jan. 2000.
- [5] O. Kwon, K. Nowka, and E. E. Swartzlander, "A 16-bit by 16-bit MAC design using fast 5:3 compressor cells," J. VLSI Signal Process. Syst., vol. 31, no. 2, pp. 77-89, Jun. 2002.
- [6] L.-H. Chen, O. T.-C. Chen, T.-Y. Wang, and Y.-C. Ma, "A multiplication-accumulation computation unit with optimized compressors and minimized switching activities," in Proc. IEEE Int. Symp. Circuits and Syst., Kobe, Japan, 2005, vol. 6, pp. 6118-6121.
- [7] Y.-H. Seo and D.-W. Kim, "A new VLSI architecture of parallel multiplier-accumulator based on Radix-2 modified Booth algorithm," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp. 201-208, Feb. 2010.
- [8] A. Peymandoust and G. de Micheli, "Using symbolic algebra in algorithmic level DSP synthesis," in Proc. Design Automation Conf., Las Vegas, NV, 2001, pp. 277-282.
- [9] W.-C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," IEEE Trans. Signal Process., vol. 51, no. 3, pp. 864-874, Mar. 2003.
- [10] C. N. Lyu and D. W. Matula, "Redundant binary Booth recoding," in Proc. 12th Symp. Comput. Arithmetic, 1995, pp. 50-57.
- [11] J. D. Bruguera and T. Lang, "Implementation of the FFT butterfly with redundant arithmetic," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 43, no. 10, pp. 717-723, Oct. 1996.
- [12] W.-C. Yeh, "Arithmetic Module Design and its Application to FFT," Ph.D. dissertation, Dept. Electron. Eng., National Chiao-Tung University, Chiao-Tung, 2001.
- [13] R. Zimmermann and D. Q. Tran, "Optimized synthesis of sum-of-products," in Proc. Asilomar Conf. Signals, Syst. Comput., Pacific Grove, Washington, DC, 2003, pp. 867-872.
- [14] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs. Oxford: Oxford Univ. Press, 2000.
- [15] O. L. Macsorley, "High-speed arithmetic in binary computers," Proc. IRE, vol. 49, no. 1, pp. 67-91, Jan. 1961.
- [16] R. Zimmermann and D. Q. Tran, "Optimized synthesis of sum-of-products," in Proc. Asilomar Conf. Signals, Syst. Comput., Pacific Grove, Washington, DC, 2003, pp. 867-872.
- [17] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs. Oxford: Oxford Univ. Press, 2000.
- [18] O. L. Macsorley, "High-speed arithmetic in binary computers," Proc. IRE, vol. 49, no. 1, pp. 67-91, Jan. 1961.
- [19] N. H. E. Weste and D. M. Harris, "Datapath subsystems," in CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Readington: Addison-Wesley, 2010, ch. 11.
- [20] S. Xydis, I. Triantafyllou, G. Economakos, and K. Pekmestzi, "Flexible datapath synthesis through arithmetically optimized operation chaining," in Proc. NASA/ESA Conf. Adaptive Hardware Syst., 2009, pp. 407-414.

Authors

- [1] Ch. Bharathi, has completed B.Tech (ECE) in 2012 from Vignan's nirula Institute of science & technology for women, Sattenapally and pursuing M.Tech (VLSIES) in Narasaraopet Institute of Technology, Narasaraopet, Guntur District, Andhra Pradesh, India. Her research area is VLSI.
- [2] D. Raghava Reddy, has his M.Tech (ES) from Veltech Engineering College, Chennai. Presently he is working as HOD in Narasaraopet Institute of Technology, Narasaraopet, Guntur District, Andhra Pradesh, India. He has over 7 years of experience in teaching. He has published and presented 2 research papers in respective International/National journals and Conferences