

## High-speed radix-10 multiplication using partial shifter adder tree-based convertor

Utsav Kumar Malviya

Department of Electronics and telecommunication Engineering, Government Engineering College, Bilaspur, India

---

### Article Info

#### Article history:

Received Dec 14, 2019

Revised Oct 24, 2020

Accepted Nov 11, 2020

---

#### Keywords:

ASIC

Binary coded decimal

FPGA

ODDS

Partial shifter adder

Radix-10

---

### ABSTRACT

A radix-10 multiplication is the foremost frequent operations employed by several monetary business and user-oriented applications, decimal multiplier using in state of art digital systems are significantly good but can be upgraded with time delay and area optimization. This work is proposed a more area and time delay optimized new design of overloaded decimal digit set (ODDS) architecture-based radix-10 multiplier for signed numbers. Binary coded decimal (BCD) to binary followed by binary multiplication and finally binary to BCD conversion are 3 major modules employed in radix-10 multiplication. This paperwork presents a replacement technique for binary coded decimal (BCD) to binary and vice-versa convertors in radix-10 multiplication. A novel addition tree structure called as partial shifter adder (PSA) tree-based approach has been developed for BCD to binary conversion, and it is used to add partially generated products. To meet our major concern i.e. speed, we need particular high-speed multiplication, hence the proposed PSA based radix-10 multiplier is using vertical cross binary multiplication and concurrent shifter-based addition method. The design has been tested on 45nm technology-based Zynq-7 field programmable gate array (FPGA) devices with a 6-input lookup table (LUTs). A combinational implementation maps quite well into the slice structure of the Xilinx Zynq-7 families field programmable gate array. The synthesis results for a Zynq-7 device indicate that our design outperforms in terms of the area and time delay.

*This is an open access article under the [CC BY-SA](#) license.*



---

### Corresponding Author:

Utsav Kumar Malviya

Department of Electronics and telecommunication Engineering

Government Engineering College

Koni, Bilaspur, Chhattishgarh, India, 45001

Email: utsav00706@mail.com

---

## 1. INTRODUCTION

The binary-coded decimal (BCD) defines the encoding of decimal numbers in binary. Packed BCD uses four binary bits to represent a fixed decimal digit. IBM introduces code capable of representing alphanumeric information in the IBM card which was later accepted by many other manufacturers it was the first BCD code [1]. All IBM desktop and workstations include BCD arithmetic in their hardware [1]. Many microprocessors have decimal number support. In 80x86 family microprocessor DAX (decimal adjust) instructions used to convert binary output into binary coded decimal (BCD) output [2]. The Zilog Z80 series of microprocessors also includes additional hardware to support Intel x86 family decimal arithmetic instructions and BCD conversion [2]. For common decimal arithmetic two binary encodings were included in the 2008 IEEE 754r standard [3] and now the IEEE754r standard supports 16- and 34-digit decimals significands. Fujitsu Sparc X microprocessors [4], Z-System, and IBM microprocessors [3] based workstations

also use IEEE 754-2008 standard decimal number multipliers. The initial Motorola 6800 processor was also capable of performing BCD arithmetic's later Motorola designed new advanced versions of processors, and replace DA instruction with Coldfire instructions [5]. Different types of BCD or radix-10 multipliers have been designed by researchers to enhance BCD multiplication performance. signed digit radix-10 BCD multiplication using a multi-operand adder structure [6] offers improvements in performance in comparison with conventional BCD multiplier. Available binary-to-BCD converter uses for BCD digits multiplier (BDM) in [7], Gorgin's [7] multiplier design was based on the truth table for conversion of binary number into BCD digits and develops a new combinational design architecture for implementation on FPGA. Vazquez [8] designs a high-speed decimal multiplier using three stage parallel process. BCD multiplier using recoded multiplier digits as in conventional parallel multiplier design was represented in 2015 [9].

A parallel decimal multiplier [10, 11] uses partial product reduction (PPR) tree and overloaded decimal digit set (ODDS) architecture and BCD-4221/5211 codes. Cui's [10, 11] multiplier consists of a binary PPR tree block, a non-fixed size BCD-4221 counter block, and a BCD-4221/5211 PPR tree block and in their work decimal carry-save algorithm based on BCD-4221/5211 is used in the PPR tree to obtain high-performance radix-10 multipliers. In ODDS architecture, the 4-bit binary number converted into radix-10 (0,1,2..15). ODDS use a carry-free generation of decimal multiples also it performs multiplication of binary number and then converts it back into decimal. In contrast with normal BCD multiplication, in ODDS there is no requirement of include extra hardware for BCD numbers just requires convertors Binary to BCD and vice-versa [10, 11]. With increasing the need for the real-time computation, it is difficult to manage with the conventional radix-10 multiplication methods, however, with speed and area optimized design of radix-10 multiplier the demand for real-time computation can be satisfied. A fast radix-10 multiplication presented in [12-14]. Vazquez's multiplier [12-14] uses a new algorithm and architecture for the BCD parallel multiplier that utilizes some properties of two redundant BCD codes and speed-up computation time, it uses BCD excess-3 (XS-3) code along with an overloaded decimal digit set (ODDS) for signed BCD number multiplication. High-speed Vedic multiplication presented in 2012 [15] for high throughput and BCD multiplier design using the Vedic multiplication method was presented in 2013 [16]. Another combinational design of the radix-10 multiplier presented in [17] for floating-point numbers with IEEE Std 754-2008 [18] without ODDS architecture. Decimal multiplication of two 4-digit numbers with their new method of partial product generation presented in [19].

ODDS architecture and carry-save adder-based binary to BCD and vice versa converter was used in decimal multiplication in [20]. A look-up-table (LUT) based combinational design of high-speed and area-efficient BCD multiplier design described in [21]. Energy optimisations methods for microprocessor and ALU are defines in [22]. In [23] defines a base 1000 use for binary to BCD conversion and design embedded arithmetic blocks based decimal multiplier. In [8-14, 20] uses the Xilinx ISE [24] EDA tool for the design and verification of their BCD multiplier.

Multi-operand adder trees [6] implementation using state of art VLSI may results complex irregular layouts of BCD multiplier as compare with binary compressor trees and carry-save adders. PPR tree and ODDS base BCD multiplier [10,11] was limited for unsigned number multiplication only, with signed-digit and self-complementing codes theirs ODDS converters fail to convert the BCD to binary and vice-versa. Alvaro's multiplier [12-14] design resolve the issue of signed digit multiplication in ODDS by implementing XS-3 encoding and their radix-10 multiplier was good for both signed and unsigned number BCD multiplication, But the convertor was using a complex sequential architecture with parallel computation [12-14]. Overall, Alvaro's radix-10 multiplier [12, 14] was less area and speed optimized then Xiaoping's radix-10 multiplier [13, 14] but was capable of handling the signed number BCD multiplication in ODDS architecture. Vedic multiplier [15, 16] is best suited only when high speed is a major concern than area optimization.

This work represents an ODDS architecture [8-14] based BCD multiplier for signed number with optimized area and speed, there are two major modules in the radix-10 (BCD) multiplier which can be optimized to fulfill the need for high-speed real-time computations, first, Binary-BCD convertors and second, type of binary multiplier. This work is considering these optimization challenges, and develop a high speed modified adder tree structure partial shifter adder (PSA) for BCD to the binary converter in radix-10 multiplier for area optimization and also this work uses binary multiplication by Vedic multiplication method [15, 16] for fast output. These two changes increase the efficiency of hardware implementation in terms of area and speed. All modules of the proposed work design and verifies on vivado design suite [25] and validated on Zynq-7000 FPGA [26]. Xilinx vivado design suits [25] is the latest advanced tool of Xilinx's ISE [24].

## 2. RESEARCH METHOD

Radix-10 multiplication in the digital processing systems is done by binary coded decimal (BCD) number system. The decimal multiplier also called the BCD multiplier or radix-10 multiplier in the digital

*High-speed radix-10 multiplication using partial shifter adder tree-based convertor (Utsav Kumar Malviya)*

system. Figure 1 shows a flow for proposed work this work follows ODDS architecture, as it may be seen three major modules have been developed first, BCD to binary convertor, second, Binary multiplication, and third Binary to BCD converters. In Figure 1 A and B are the 4 digits (4 BCD digit is 16 binary bits) BCD inputs which requires to get multiply and an 8-digit expected BCD. This work produces final result following steps need to follow: -

Step 1: Convert A and B into a binary number individually. As A and B are of 4 BCD digit then expected output binary maximum up to 14 bit only because maximum 4-digit BCD number is 9999 requires only 14 binary bits. This BCD to binary conversion is done using a new proposed PSA tree adder.

Step 2: Multiply 14x14 binary numbers C and D using vertical cross multiply or Vedic multiplication [8] for fast results.

Step 3: convert the 27-bit binary answer into 8-digit BCD number.

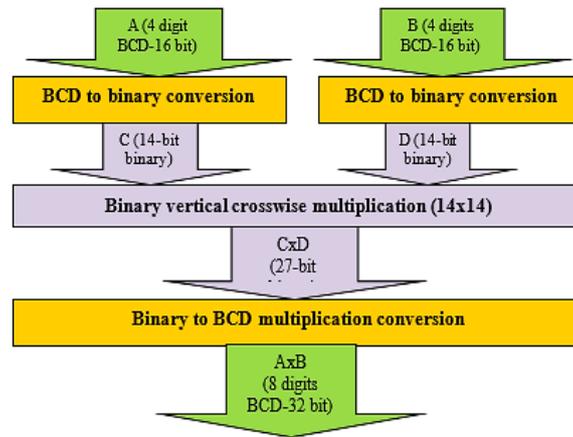


Figure 1. Proposed ODDS architecture base BCD multiplication

### 2.1. BCD to binary convertor

The proposed new architecture of 16-bit BCD to 14-bit binary convertor is shown in Figure 2. In proposed architecture, a new partial shifter adder (PSA) is used. This PSA based architecture also considers the sign bit and as per the sign bit, it converts the 4-digit BCD number into binary. PSA is combinational architecture which uses carry-free generation of decimal multiples and this makes the PSA significantly fast. The proposed method uses shifter operations, the left shift by 'n' bits is equaling to multiply a number with  $2^n$ . B15, B14 .....B0 is the 16-bit input BCD number and s13, s12.....s0 is 14-bit output binary, in (1) shows the binary generation from 16 bit BCD number. In (2) is the generation of 14 bit binary from 16-bit BCD using left shift operation. In (3) multiplication by 1024 is left shift by 10-bit, multiplication by 16 is left shift by 4 bit, multiplication by 8 is left shift by 3 bit, multiplication by 64 is left shift by 6 bit, multiplication by 32 is left shift by 5 bit, multiplication by 4 is left shift by 2 bit, multiplication by 2 is left shift by 1 bit. In this work a partial shifter adder tree where 2's complement addition used for subtraction, upper nibble of 16-bit BCD B15 B14 B13 B12 conceded with extra '0' at its MSB first, as shown in (4). In (5) obtained by replacing subtraction operations of (3) according to (4).

$$S13S12S11S10S9S8S7S6S5S4S3S2S1S0 = (B15B14B13B12) \times 1000 + (B11B10B9B8) \times 100 + (B7B6B5B4) \times 10 + (B3B2B1B0) \quad (1)$$

$$S13S12S11S10S9S8S7S6S5S4S3S2S1S0 = (B15B14B13B12) \times (1024 - 16 - 8) + (B11B10B9B8) \times (64 + 32 + 4) + (B7B6B5B4) \times (8 + 2) + (B3B2B1B0) \quad (2)$$

$$S13S12S11S10S9S8S7S6S5S4S3S2S1S0 = (B15B14B13B12) \times (1024) - (B15B14B13B12) \times (16) - (B15B14B13B12) \times (8) + (B11B10B9B8) \times (64) + (B11B10B9B8) \times (32) + (B11B10B9B8) \times (4) + (B7B6B5B4) \times (8) + (B7B6B5B4) \times (2) + (B3B2B1B0) \quad (3)$$

$$1 A15 A14 A13 A12 = \sim \{0 B15 B14 B13 B12\} \quad (4)$$



14-bit vertical cross multiplication implemented only and final answer is produced of 27 bit only which save another half adder need and reduce area requirement. The multiplication of two binary converted biggest 4 digits BCD's can be of 27-bit binary only which can be further converted into 8-digit BCD.

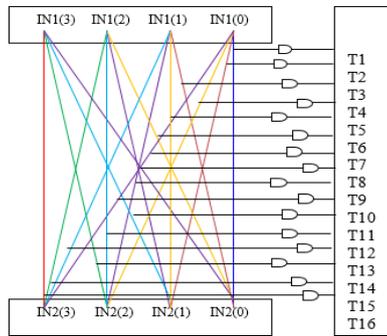


Figure 4. Cross components generation for 4-bit Vedic multiplier

		C9	C6				
	C11	C8	C5	C3			
C12	C10	C7	C4	C2	C1		
	T16	T14	T11	T7	T4	T2	T1
		T15	T12	T8	T5	T3	
			T13	T9	T6		
				T10			
S7	S6	S5	S4	S3	S2	S1	S0
0FA	1FA	2FA	2FA	2FA	1FA	0FA	0FA
0HA	0HA	0HA	1HA	1HA	1HA	1HA	0HA

Figure 5. Tree addition structure for 4-bit cross multiplication

**2.3. Binary to BCD converter**

Figure 6 shows the method of converting a 16-bit binary number to 4-digit BCD number, a similar binary to BCD converter design has been used for 28-bit binary number conversion into 8-digit BCD using VHDL while implementation. From Figure 6 leading one detection (LOD), subtraction and counting operation are used for conversion. this converter is sequential design so to improve throughput 3 stage parallel processing used between the generation of digits. BCD conversion form binary number is shown in the flow diagram of Figure 6 the process can be understood by an example elaborated in Table 1 consider any 12-bit binary number is (3485d)=(D82h)=(11011000010b). LOD is at 12. From Table 1 the process of binary to BCD conversion can be observed with an example, D3, D2..D0 is the final BCD output.

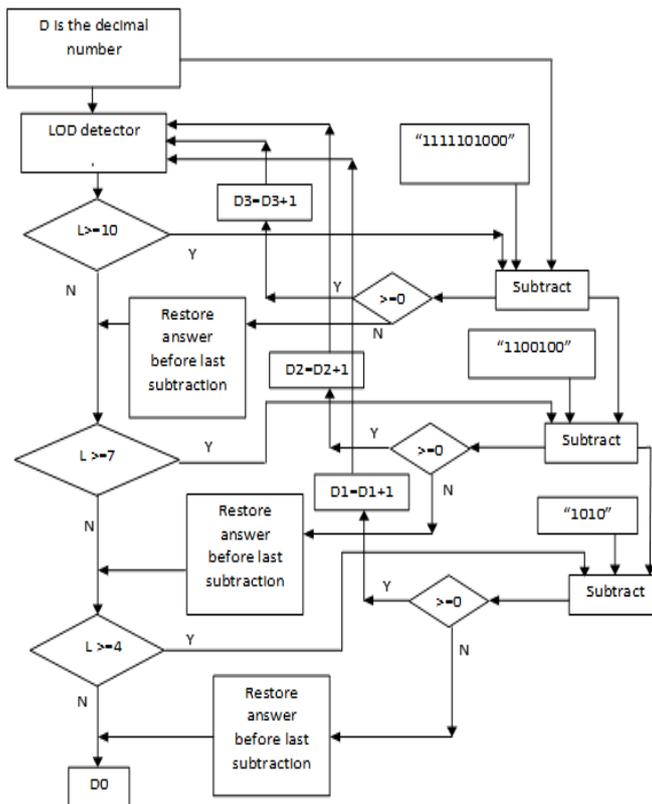


Figure 6. Binary to BCD conversion

Table 1. Binary to BCD conversion example

Binary I/P is 110110000010	LOD	D3	D2	D1	D0
110110000010-1111101000	12	1			
100110011010-1111101000	12	2			
10110110010-1111101000	11	3			
111001010-1100100	9		1		
101100110-1100100	9		2		
100000010-1100100	9		3		
1100100-1100100	7		4		
111010-1010	6			1	
110000-1010	6			2	
100110-1010	6			3	
11100-1010	5			4	
10010-1010	5			5	
1000	4				8
		3	4	5	8

3. RESULTS AND ANALYSIS

The register transfer logic entries are done using very high-speed integrated circuit hardware description language (VHDL) in the proposed design. Synthesis is done for combinational style modeling and the target device selected is Zynq-7000 FPGA. Figure 7 shows the simulation result obtain for this 16x16 radix-10 multiplier design; this result generated using the integrated simulation environment (ISE) of Xilinx Vivado [14]. A few other simulation observations are shown in Table 2. Row-1 of Table 2 shows the observation of the ISE simulator shown in Figure 7.

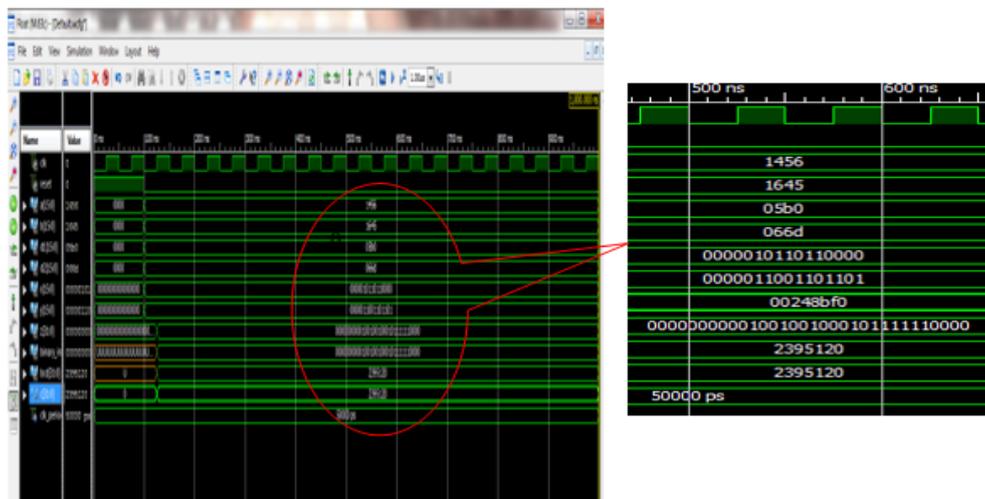


Figure 7. BCD multiplication simulation observed on Xilinx VIVADO-ISE

Table 2. The simulation results obtained on ISE

In1 (4-digit BCD)	In2 (4-digit BCD)	Bin1 (14-bit max)	Bin2 (14-bit max)	Binary multiplication output (27-bit max)	BCD output (8-digit BCD)	Outcome
1456	1645	5B0	66D	24BF0	2395120	Correct
2354	6369	932	18E1	E4C4F2	14992626	Correct
6002	5002	1772	138A	1CA1974	30022004	Correct
9999	9999	270F	270F	5F592E1	99980001	Correct

Table 3 shows the Synthesis results for the proposed 16x16 BCD multiplier design, observed using the Xilinx Vivado tool [14] and Zynq-7000 Field programmable gate array (FPGA) [14]. From Table 3 it may observe that numbers of Zynq-7000 FPGA slices used in this work are 482 and also the Maximum frequency obtain is 267.251 Mhz. The simulations are observed for a balanced Simulator setting in Xilinx Vivado. The area obtains for the proposed work design implementation is 0.04138 mm<sup>2</sup>. The number of universal gates two input NAND required for this design is 29362.

Table 4 shows the different methods used and by researchers and proposed method with the time delay results. The results are obtained using the HDL based FPGA implementation. This work is an application specific integrated circuit (ASIC) design and the target device FPGA technology must be the same for comparison, hence this work has selected FPGA of 45 nm technology same as of [1-3]. It is observed that the time delay for the proposed design of 16x16 BCD multiplication is lowest among other related works for the same platform. from the table, it may also observe that the area requirement of proposed work is less as compared with other work.

Table 3. Device utilization summary

Target FPGA Device: Zynq-7000	
Parameters	Used
The number for Slices:	482
The number for Slice Flip Flops:	105
The number for 4 input LUTs:	863
The number for bonded IOBs:	66
Time Delay	3.863ns
Max Freq.	267.251MHz
Area NAND2	29362
Area ( $mm^2$ )	0.04138
The number for GCLKs:	1

Table 4. Comparative Results obtain for target Zynq-7000 FPGA device (45nm technology)

Design	Method	Time Delay	No of slices	Area NAND2	Area ( $mm^2$ )
16x16 BCD multiplication	Proposed ODDS with PSA tree method	3.192 ns	482	29362	0.04138
	PPR tree with ODDS method [1]	3.21 ns	-	-	0.0422
	ODDS with XS-3 [2]	41.5 ns	-	120600	
	PPR tree with ODDS coding [3]	3.29 ns	-	32656	

#### 4. CONCLUSION

This work is a design of a decimal digit set (ODDS) architecture-based radix-10 multiplier for signed numbers. In this work, three modification methods incorporated, first, modified partial shifter adder (PSA) tree used for BCD to binary conversion, second, 14x14 Vedic binary multiplication used, and produce 27-bit binary output and third, BCD to binary conversion using sequential design with 7 stage pipelines. PSA tree is a combinational design with a minimum number of FA and HA requirements which reduce overall time delay. An addition tree in the Vedic multiplier also uses the PSA tree structure and produces 27-bit output instead of 28bit after the multiplication of two 14-bit input also reduces area. An additional register between the digits of the final output allows 7 stage pipeline which enhances overall throughput of output. The observed simulation results tested on the Xilinx Vivado tool and also verified on Zynq-7000 FPGA. Simulation is performed all possible test inputs and verified correctly. From the analysis of the results, the time delay, and area on IC found in this work for 16x16 radix-10 multiplication is less as compare with other similar work.

#### ACKNOWLEDGEMENTS

This work is carried out for the fulfillment of the Collaborative Research Scheme (CRS) of all India Council for Technical Education (AICTE) under TEQIP-III, New Delhi India.

#### REFERENCES

- [1] S. Mueller, S. Carlough, A. Collura, and M. Kroener, "The IBM Enterprise, Decimal Floating-Point Accelerator," *Proceedings of the 20th IEEE Symposium on Computer Arithmetic*, pp. 139-146, Jul. 2011.
- [2] H. Schmidt, "Decimal Computation," ISBN-047176180X, *John Wiley & Sons*, 1974.
- [3] IEEE Std 754xx-2008, "IEEE Standard for Floating-Point Arithmetic," *IEEE Computer Society*, 2008.
- [4] T. Maruyama, "SPARC64 X: Fujitsu's New Generation 16 Core Processor for the Next Generation UNIX Server," *Hot Chips 24 Conference*, 2012.
- [5] Motorola M68000 family programmer's reference manual, "CPU32 instructions," *Motorola inc.*, 1992
- [6] Kuan Jen Lin, Yu Chan Chiu, Tzu-Hao Lin, "Decimal Multiplication with Efficient Partial Product Generation," *18th IEEE/IFIP International Conference on VLSI and System-on-Chip*, pp. 27-29, 2010.
- [7] Gorgin, Sa'ed, Jaberipur, Ghassem, Hashemi Reza, "Efficient ASIC and FPGA Implementation for Binary-Coded Decimal Digit Multiplications," *Circuits Systems Signal Processing Academic Journal*, vol. 33, no. 12, pp. 3883-3899, 2014.

- [8] Alvaro Vazquez, Elisardo Antelo, P. Montuschi, "Improved Design of High-Performance Parallel Decimal Multipliers," *IEEE Transactions on Computers*, vol. 59, no. 5, pp. 679-693, 2010.
- [9] Sonam Negi, Pitchaiah Madduri, "Implementation of High-Speed Radix-10 Parallel Multiplication using Verilog," *19th International Symposium on VLSI Design and Test*, 2015
- [10] Xiaoping Cui, Wenwen Dong, Weiqiang Liu, Earl E. Swartzlander, Fabrizio Lombardi, "High-Performance Parallel Decimal Multipliers using Hybrid BCD Codes," *IEEE transactions on computers*, vol. 66, no. 12, 2017.
- [11] Xiaoping Cui, Weiqiang Liu, and Dong Wenwen Fabrizio Lombardi, "A Parallel Decimal Multiplication Using Hybrid Binary Coded Decimal (BCD) Codes," *23rd Symposium on Computer Arithmetic*, IEEE, 2016.
- [12] Alvaro Vazquez, Elisardo Antelo, Javier D. Bruguera, "Fast Radix-10 Multiplication Using Redundant BCD Codes," *IEEE transactions on computers*, vol. 63, no. 8, 2014.
- [13] Alvaro Vazquez, Elisardo Antelo, Paolo Montuschi, "A New Family of High-Performance Parallel Decimal Multipliers," *18th IEEE Symposium on Computer Arithmetic*, pp. 195-204, 2007.
- [14] Alvaro Vazquez, Elisardo Antelo " Multi-operand decimal addition by efficient reuse of a binary carry-save adder tree," *Conference Record of the Forty-Fourth Asilomar Conference on Signals, Systems and Computers*, IEEE, 2010.
- [15] Abhishek Gupta, Utsav Malviya, Vinod Kapse, "Design of Speed, Energy and Power Efficient Reversible Logic Based Vedic ALU for Digital Processors," *Nirma university international conference on engineering*, IEEE, 2012.
- [16] Arvind Kumar Mehta, Mukesh Gupta, Vipin Jain, Sudhir Kumar, "High-Performance Vedic BCD Multiplication and Modified Binary to BCD Converter," *Annual IEEE India Conference (INDICON)*, 2013.
- [17] T. Lang, A. Nannarelli, "A Radix-10 Combinational Multiplier," *Proceeding of 40th Asilomar Conference on Signals, Systems and Computers*, pp. 313-317, Oct. 2006.
- [18] IEEE, "IEEE Standard for Floating-Point Arithmetic," IEEE Std 754-2008.
- [19] M. A. Erle, M. J. Schulte, E. M. Schwarz, "Decimal Multiplication with Efficient Partial Product Generation," *17th IEEE Symposium on Computer Arithmetic*, IEEE Computer Society, pp. 21-28, 2005.
- [20] M. J. Schulte, M. A. Erle, "Decimal Multiplication Via Carry-Save Addition," *Proceeding of IEEE International Conference on Applications-Specific Systems, Architectures and Processors*, pp. 348-358, 2003.
- [21] Zarrin Tasnim Sworna, Mubin UI Haque, D. M. Anisuzzaman, "High-Speed and Area-Efficient LUT-Based BCD Multiplier Design," *IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2018
- [22] R. Krishnamurthy, V. G. Oklobdzija, "High-Performance Energy Efficient Microprocessor Design," *Springer*, 2006.
- [23] H. C. Neto, M. P. Vestas, "Decimal multiplier on FPGA using embedded binary multipliers," *Proceedings of International Conference of Field Programmable Logic and Applications FPL 2008*, pp. 197-202, 2008.
- [24] Xilinx inc., "ISE In-Depth Tutorial," UG695 (v14.1) April 24, 2012.
- [25] Xilinx inc., "Vivado Design Suite User Guide," UG910 (v2014.1) April 2, 2014.
- [26] Xilinx inc., "Zynq-7000 SoC Data Sheet: Overview," Product Specification, DS190 (v1.11.1) July 2, 2018.

## BIOGRAPHIES OF AUTHORS



**Utsav kumar Malviya** in 2007 received his B.Tech. degree in electronics engineering from IETE, New Delhi, India, in 2010 he received M.Tech. degree in embedded system program from DAVV University, Indore, India. He is currently working as an assistant professor in Government engineering college Bilaspur. Utsav malviya has 10 years of teaching experience and 1 year of industrial experience from ST Ericsson. His research interests include design and verification methodology of SoCs, ASIC design and Image processing.