

ASIC design of low power-delay product carry pre-computation based multiplier

Chaitanya CVS¹, Sundaresan C², P R Venkateswaran³, Keerthana Prasad⁴

^{1,2,4}School of Information Sciences, Manipal Academy of Higher Education, Manipal, Karnataka, India.

³Bharat Heavy Electricals Limited, Tiruchurapalli, Tamil Nadu, India.

Article Info

Article history:

Received Oct 6, 2018

Revised Dec 07, 2018

Accepted Dec 21, 2018

Keywords:

Binary Multiplication
Carry Pre Computation
Multiplier Architecture
Operand Decomposition
Vedic Multiplier

ABSTRACT

High speed and efficient multipliers are essential components in today's computational circuits like digital signal processing, algorithms for cryptography and high performance processors. Invariably, almost all processing units will contain hardware multipliers based on some algorithm that fits the application requirement. Tremendous advances in VLSI technology over the past several years resulted in an increased need for high speed multipliers and compelled the designers to go for trade-offs among speed, power consumption and area. Amongst various methods of multiplication, Vedic multipliers are gaining ground due to their expected improvement in performance. A novel multiplier design for high speed VLSI applications using Urdhva-Tiryagbhyam sutra of Vedic Multiplication has been presented in this paper. The proposed architecture modeled using Verilog HDL, simulated using Cadence NCSIM and synthesized using Cadence RTL Compiler with 65nm TSMC library. The proposed multiplier architecture is compared with the existing multipliers and the results show significant improvement in speed and power dissipation.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Chaitanya CVS,
School of Information Sciences,
Manipal Academy of Higher Education,
Manipal 576104, Karnataka, India.
Email: chaitanya.cvs@manipal.edu

1. INTRODUCTION

Processors are important part of integrated circuits (IC). Large numbers of functionalities are packed in an IC thanks to tremendous growth in density of integration in recent times. As the number of functions increases, the need for computation also grows. With the advent of new process technologies, shrinking of feature size and availability of modern CAD tools, a development of complex integrated circuits for various applications is possible. Examples of such applications include digital signal processing [1,2], mobile computations and communications, multimedia applications and processing required for scientific computing and applications etc. The speed and efficiency of processor in such IC is very crucial for meeting the requirements of the applications supported by the IC. The speed of processor and efficiency of processor in-turn depends upon an arithmetic logic unit [3] which is considered as the main computational unit of the processor.

Moreover, the multiplier units [4] are the most important hardware structures in a complex arithmetic unit. The multiplier units are capable of performing operations on operands of various data types such as calculating running sum of products. As multiplication is a crucial arithmetic operation in processors [5] and digital computer systems, multipliers are the core building block for many algorithms in a wide variety of computing applications. Although multipliers are main arithmetic components used for processing scientific data, the excessive power consumption and delay attracts attention from the research

community. Usually, multiple arithmetic cores working in parallel are used so as to process large amounts of data with relatively low power and delay.

Various algorithms have been proposed for the hardware implementation of multipliers in the past. Add and Shift is the common algorithm used in designing of multiplier [6]. In parallel multipliers, the important parameter which is used to determine performance is the number of partial products which are needed to be added. One such algorithm is Modified Booth algorithm [7] which reduces the number of partial products during the multiplication which in turn increases the performance of the multiplier. Another algorithm is Wallace tree based algorithm which reduces number of adding stages and is used to improve the speed of multiplication. In some implementations, efficient multiplier architecture is designed by combining both Modified Booth algorithm and Wallace Tree algorithm. However, an increasing parallelism increases the number of shifts between intermediate sum and partial products which results in reduced speed, increased power consumption and also increased area because of irregular structure. Thus, in some cases, low power and compact multiplier architectures is implemented using serial multiplication algorithm. Serial multipliers [8] have better performance for power consumption and area with the delay tradeoff. Depending upon the application, either parallel or serial multipliers are selected to perform the operation.

However, in the high speed processors which are operating at higher clock frequencies, the existing multiplier takes more delay for execution of the instructions. The existing multiplier units that consume more power are not suitable to be incorporated in the processors which are used in wireless and portable devices. Thus, power savings is an important area for improvement.

In order to address the low power computation along with high performance, a new approach to multiplier design based on ancient Vedic Mathematics has been explored. The mathematical operations using Vedic mathematics are very fast and require less hardware. This aspect of Vedic mathematics can be utilized to increase the computational speed of multipliers. This paper describes the design and implementation of a Vedic multiplier based on Urdhva-Tiryagbhyam Sutra [9]-[11]. The number of steps required to perform a multiplication operation by using UrdhvaTiryagbhyam Sutra are considerably less compared to the conventional multiplication techniques. In this paper, we have further explored a novel method to enhance the speed of a Vedic multiplier by pre-computing the carries which are used during summation of partial products. The implementation of pre-computation logic using multiplexer based carry-look ahead logic and XOR logic resulted in reduction of delay. The proposed multiplier along with operand decomposition technique resulted in reduction of power consumption which in turn reduced the power-delay product of the multiplier.

The structure of the paper is divided as follows: The methodology and the architecture of the proposed multipliers are given in section 2. Results are presented in section 3. Finally, conclusion is given in section 4.

2. RESEARCH METHOD

2.1. Carry pre-computation based binary multiplier

An 8 bit Binary Vedic Multiplier has been proposed with A and B as inputs and P as the final 16-bit product. The block diagram for 8 bit multiplication is shown in Figure 1. In the proposed multiplier the operands A and B are divided into Higher and Lower parts with 4-bits each.

$$A = \{AH, AL\} \quad (1)$$

$$B = \{BH, BL\} \quad (2)$$

AL*BH	
AH*BH	AL*AL
AH*AL	
Product	

Figure 1. Block Diagram of 8-bit Multiplication

In this type of multiplier an 8 bit Binary multiplication is realized using 4-bit binary vedic multiplication using carry pre-computation logic shown in below Figure 2. where A3, A2, A1, A0 & B3, B2, B1, B0 are 4 bit binary inputs and P7, P6, P5, P4, P3, P2, P1, P0 are the binary output bits.

		A ₃	A ₂	A ₁	A ₀		
	X	B ₃	B ₂	B ₁	B ₀		
				pp ₄	pp ₃	pp ₂	pp ₁
			pp ₈	pp ₇	pp ₆	pp ₅	
		pp ₁₂	pp ₁₁	pp ₁₀	pp ₉		
	pp ₁₆	pp ₁₅	pp ₁₄	pp ₁₃			
			c ₃₂	c ₃₁	c ₂		
			c ₄₂	c ₄₁			
		c ₅₂	c ₅₁				
c ₆₂	c ₆₁						
c ₇₁							
P ₈	P ₇	P ₆	P ₅	P ₄	P ₃	P ₂	P ₁

Figure 2. Carry Pre-Computation Based Multiplier

The architecture of the 4-bit multiplier can be understood from the block diagram shown in Figure 3.

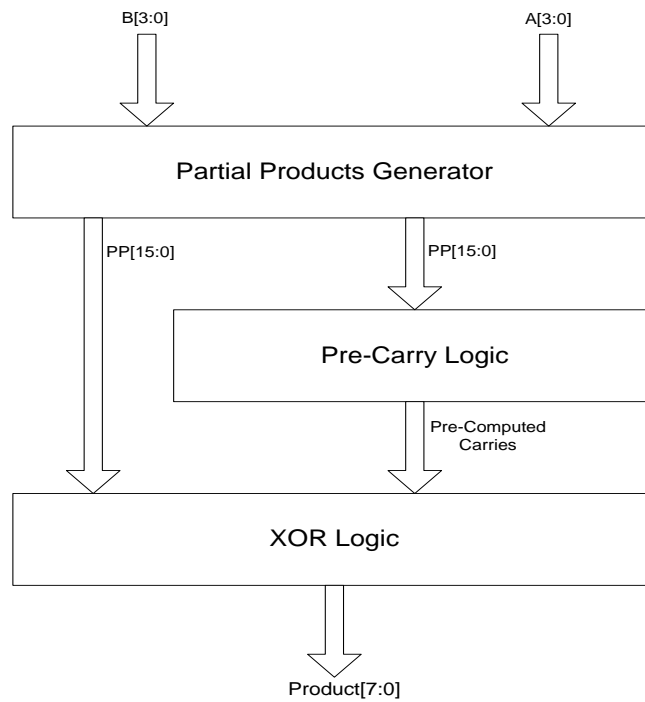


Figure 3. Architecture of Carry Pre-Computation based Multiplier

The partial product generator is the first block of the multiplier to which the 4 bit multiplicand and multiplier are given as inputs. At this juncture, the multiplication technique used is Urdhva-Tiryagbhyam. The 4 bit multiplication results in a total of 16 partial products (pp1-pp16). The result of multiplying any one binary bit with another is either a zero or a one which is simply the logic of ANDing of the two bits.

The products of AL*BL, AH*BL, AL*BH, AH*BH are determined using above 4-bit carry pre-computation based multiplier and the results of all sub multipliers are added to determine the final product. The block of the 8-bit multiplier is shown in Figure 4.

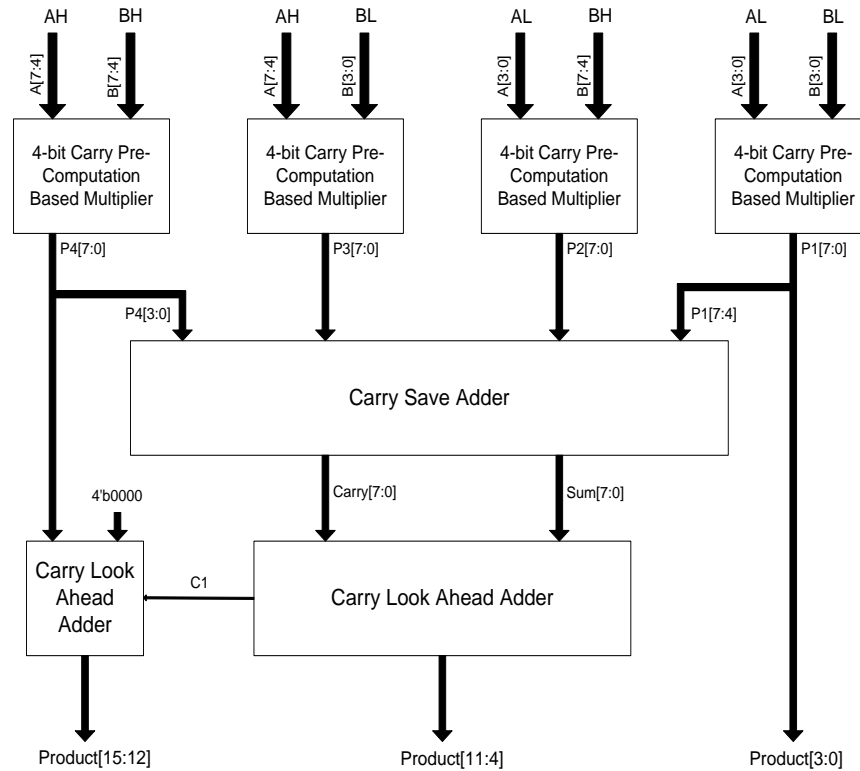


Figure 4. Block Diagram of 8-bit Multiplier Using 4-bit Carry Pre-Computation Based Multiplier

The second stage in the block diagram is the carry generation circuit. Here, we have integrated pre-computation logic along with the Urdhva-Tiryagbhyam multiplication technique. The carry equations are generated separately for each column of partial products and the inputs for these equations are taken from the previous column. The equations for pre-computed carries are given below.

$$c2 = pp5 \& pp2; \quad (3)$$

$$c3t1 = (pp6 \& pp3) \mid (pp9 \& (pp3 \mid pp6)); \quad (4)$$

$$c3t2 = (pp9 \& \sim pp6) \mid (pp3 \& \sim pp9) \mid (\sim pp3 \& pp6); \quad (5)$$

$$c31 = c2 ? c3t2 : c3t1; \quad (6)$$

$$c32 = pp2 \& pp5 \& pp3 \& pp6 \& pp9; \quad (7)$$

$$c41t1 = pp13 ? ((pp10 \& \sim pp7) \mid (pp4 \& \sim pp10) \mid (\sim pp4 \& pp7)) : ((pp7 \& pp4) \mid (pp10 \& (pp4 \mid pp7))); \quad (8)$$

$$c41t2 = pp13 ? ((\sim pp7 \& \sim pp4) \mid (\sim pp10 \& (\sim pp4 \mid \sim pp7))) : ((\sim pp7 \& pp4) \mid (pp10 \& \sim pp4) \mid (\sim pp10 \& pp7)); \quad (9)$$

$$c41 = c31 ? c41t2 : c41t1; \quad (10)$$

$$c42 = ((c31 \& pp13) \& ((pp10 \& (pp7 \mid pp4)) \mid (pp7 \& pp4))) \mid ((pp10 \& pp7 \& pp4) \& (c31 \mid pp13)); \quad (11)$$

$$c51t1 = c32 ? ((pp14 \& \sim pp11) \mid (pp8 \& \sim pp14) \mid (\sim pp8 \& pp11)) : ((pp11 \& pp8) \mid (pp14 \& (pp8 \mid pp11))); \quad (12)$$

$$c51t2 = c32 ? ((\sim pp11 \& \sim pp8) \mid (\sim pp14 \& (\sim pp8 \mid \sim pp11))) : ((\sim pp11 \& pp8) \mid (pp14 \& \sim pp8) \mid (\sim pp14 \& pp11)); \quad (13)$$

$$c51 = c41 ? c51t2 : c51t1; \quad (14)$$

$$c52 = ((c41 \& c32) \& ((pp14 \& (pp11 \mid pp8)) \mid (pp11 \& pp8))) \mid ((pp14 \& pp11 \& pp8) \& (c41 \mid c32)); \quad (15)$$

$$c6t1 = (pp12 \& pp15) \mid (c42 \& (pp12 \mid pp15)); \quad (16)$$

$$c6t2 = (c42 \& \sim pp15) \mid (pp12 \& \sim c42) \mid (pp15 \& \sim pp12); \quad (17)$$

$$c61 = c51 ? c6t2 : c6t1; \quad (18)$$

$$c62 = c51 \& c42 \& pp12 \& pp15; \quad (19)$$

$$c71 = (c52 \& pp16) \mid (c61 \& (c52 \mid pp16)); \quad (20)$$

The third stage in the block diagram involves the use of XOR logic for the partial products and carry generated in each column. The output of this stage gives the final 16 bit product which is obtained in a parallel mechanism instead of sequential mechanism.

2.2. Carry pre-computation based binary multiplier using operand decomposition

In operand decomposition [12], the operands X and Y are decomposed into four numbers A, B, C and D to reduce the number of ones in the partial products. The operands are decomposed in such a way that the number of zeros in decomposed operand will be more when compared to number of ones. As the number of zeros are more, the switching activity of the circuit will be reduced which in turn reduce the dynamic power consumption of the architecture.

Assuming that the two operands are X and Y have n bits,

$$\begin{aligned} X &= [X_{n-1} X_{n-2} \dots X_1 X_0], \text{ and} \\ Y &= [Y_{n-1} Y_{n-2} \dots Y_1 Y_0] \end{aligned} \quad (21)$$

The four decomposed operands are given in the following

$$\begin{aligned} A &= \sim X \wedge \sim Y, \\ B &= X \wedge Y, \\ C &= \sim X \wedge Y, \text{ and} \\ D &= X \wedge \sim Y \end{aligned} \quad (22)$$

Where, \wedge is and operation & \sim is two's complement
The final product is determined by using equation 23.

$$X * Y = (C * D) - (A * B); \quad (23)$$

The products of C*D and A*B are determined using 8-bit carry pre-computation based multiplier. Then the final partial sum and carry from both products can be combined carry save adder and carry look ahead adder. The block diagram for above multiplier is shown in Figure 5.

3. RESULTS AND ANALYSIS

The proposed architecture modeled using Verilog HDL, simulated using Cadence NCSIM and synthesized using Cadence RTL Compiler with 65nm TSMC library. Different implementation methodology have been taken and implemented in same technological environment and then compared the performance parameters. For the comparison point of view the ideas have been considered from the references and simulated and performance parameters was computed using the same MOSFET technology file. Input data was taken in a regular fashion for experimental purpose. The delay and the power measured using the worst-case pattern and from the output where the delay is maximum.

It is observed that the proposed carry pre-computation based multiplier and carry pre-computation based multiplier with operand decomposition offered substantial reduction of propagation delay and total power consumptions. From Table 1 and Table 2, it can be observed that the proposed carry pre-computation based multiplier design offered ~23%, ~64%, ~57%, ~83%, ~94% when compared with array multiplier, wallace multiplier, column based multiplier, Nikhila based and compressor based multipliers respectively, and carry pre-computation based multiplier with operand decomposition offered ~41%, ~72%, ~67%, ~87%,

~95% when compared with array multiplier, wallace multiplier, column based multiplier, Nikhilam based and compressor based multipliers respectively.

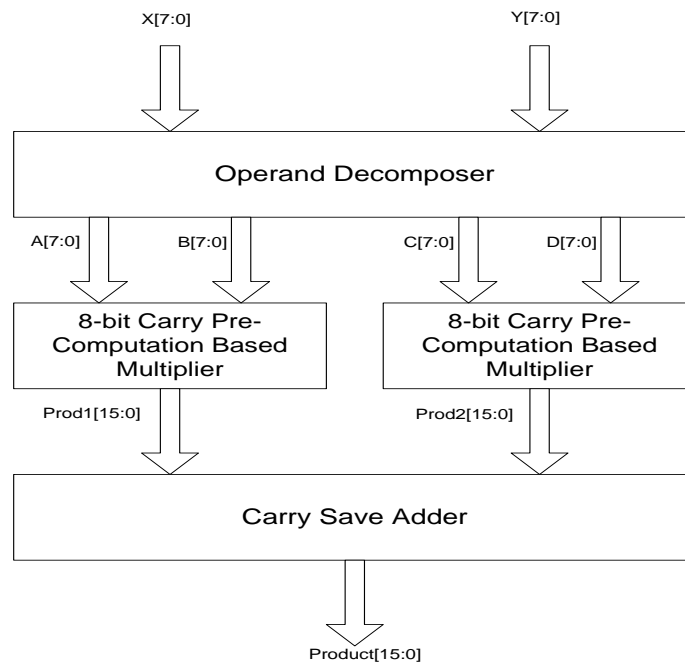


Figure 5. Carry Pre-Computation Based Multiplier Using Operand Decomposition

Table 1. Summary of Synthesis Results of 8-Bit Multiplier Architectures

S.No	Architecture (8-bit)	Delay (ns)	Dynamic Power (uW)	Static Power (uW)	Total Power (uW)	Power-Delay Product (pJ)
1	Array Based Multiplier [6]	1.5	15.09	6	21.09	31.63
2	Wallace Based Multiplier [2]	1.2	6.27	49.913	56.184	67.42
3	Column Based Multiplier [9]	1.95	26.74	2.8	29.54	57.6
4	Nikhilam Based Multiplier [10]	3.2	42.56	4.3	46.86	149.95
5	Compressor Based Multiplier [11]	4.02	95.2	6.79	101.99	410.92
6	Pre-Computation Based Multiplier	0.75	25.77	7.45	33.23	24.23
7	Pre Computation Based Multiplier with Operand Decomposition	1.02	3.36	14.808	18.172	18.5

Table 2. Summary of Synthesis Results of 16-Bit Multiplier Architectures

S.No	Architecture (16-bit)	Delay (ns)	Dynamic Power (uW)	Static Power (uW)	Total Power (uW)	Power-Delay Product (pJ)
1	Array Based Multiplier [6]	2.89	30.18	12	42.18	121.90
2	Wallace Based Multiplier [2]	2.46	12.54	99.826	112.366	276.42
3	Column Based Multiplier [9]	3.82	52.48	5.4	57.88	221.10
4	Nikhilam Based Multiplier [10]	5.96	80.65	8.1	88.75	528.95
5	Compressor Based Multiplier [11]	8.04	190.4	13.58	203.98	1639.99
6	Pre-Computation Based Multiplier	1.4	51.54	14.9	66.44	93.016
7	Pre Computation Based Multiplier with Operand Decomposition	1.96	6.72	29.616	36.336	71.218

From the Table 1 and Table 2, it can be observed that carry pre-computation based multiplier with operand decomposition consumes less power when compared to carry pre-computation based multiplier with the delay tradeoff. Proposed Carry pre-computation based multiplier with operand decomposition gave the better power-delay product when compared to proposed carry pre-computation based multiplier and existing multiplier from literature.



4. CONCLUSION



In this paper, a Vedic mathematics based multiplier has been proposed which uses Carry pre-computation and operand decomposition methodology. The proposed architecture combines the benefits of Vedic method, parallel pre-computation of carries, and operand decomposition thereby resulting in reduction of power-delay product. The propagation delay of carry pre-computation based multiplier for calculation of 8 bit and 16 bit multiplication was 0.75ns and 1.4ns while power consumption was 33.23 uW and 66.44 uW. The propagation delay of carry pre-computation based multiplier with operand decomposition for calculation of 8 bit and 16 bit multiplication was 1.02ns and 1.96ns while power consumption was 18.17 uW and 36.13 uW. The delay of multiplication was decreased by ~68% and power consumption was reduced by ~61% when compared to Nikhilam based Vedic multiplier.

REFERENCES

- [1] Xiangui Kang, AnjiePeng, XianyuXu, Xiaochun Cao, Performing Scalable Lossy Compression On Pixel Encrypted Images, EURASIP Journal on Image and Video Processing, (2013), pp. 1-6.
- [2] Nikolay Ponomarenko, Sergey Krivenko, Vladimir Lukin, Karen Egiazarian, Jaakko T, Astola, Lossy Compression of Noisy Images Based on Visual Quality: A Comprehensive Study, EURASIP Journal on Advances in Signal Processing, (2010), pp. 1-13.
- [3] L.-K. Wang, M. A. Erle, C. Tsen, E. M. Schwarz, and M. J. Schulte, A survey of hardware designs for decimal arithmetic, IBM Journal of Research and Development, 54 (2) (2010), pp. 8:1-8:15.
- [4] M. Jeevitha, R. Muthaiah, P. Swaminathan, Efficient Multiplier Architecture in VLSI Design, Journal of Theoretical and Applied Information Technology, 38 (2) (2012), pp. 196-201.2
- [5] J. R. Boddie, G. T. Daryanani, I. I. Eldumiaty, R. N. Gadenz, J. S. Thompson, S. M. Walters, Digital Signal Processor: Architecture and Performance, Bell System Technical Journal, 60 (7) (1981), pp. 1449-1462.
- [6] Ko-Chi Kuo, Chi-Wen Chou, Low Power And High Speed Multiplier Design With Row Bypassing And Parallel Architecture, Microelectronics Journal, 41 (2010), pp. 639-650.
- [7] Constantinos Efsthathiou, N. Moshopolous, N. Axelos, K. Pekmestzi, Efficient Modulo 2n+1 Multiply And Multiply-Add Units Based On Modified Booth Encoding, Integration, the VLSI Journal, 47 (2014), pp. 140-147.
- [8] Manas Ranjan Meher, Ching Chuen Jong, and Chip-Hong Chang, "A High Bit Rate Serial-Serial Multiplier With On-the-Fly Accumulation by Asynchronous Counters", IEEE trans. On VLSI systems, Vol. 19, No. 10, pp. 1733-1745, October, 2011.
- [9] BharatiKrsnaTirthaji, V. S Agrawala, "Vedic Mathematics", 13th Edition, Motilal Banarsidass, 2010.
- [10] P. Saha, A. Banerjee, A. Dandapat, and P. Bhattacharyya, "ASIC design of a high speed low power circuit for factorial calculation using ancient Vedic mathematics", ELSEVIER Microelectronics Journal, vol. 42, issue 12, pp. 1343-1352, December, 2011.
- [11] MD. Belal Rashid, Balaji B.S and Prof. M.B. Anandaraju, "VLSI Design and Implementation of Binary Multiplier based on UrdhvaTiryagbhyam Sutra with reduced Delay and Area", International Journal of Engineering Research and Technology, vol. 6, no. 2, pp. 269-278, March, 2013.
- [12] Rizwan Mudassir, Mohab Anis, and Javid Jaffari, "Switching Activity Reduction in Low Power booth Multiplier", IEEE Symposium on Circuits and Systems, Seattle, vol. 1, pp. 3306-3309, May, 2008.

BIOGRAPHIES OF AUTHORS

	<p>Chaitanya CVS received his Bachelor Degree in Electronics and Communication Engineering in 2006 from JNTU, Hyderabad and his MS degree in VLSI-CAD from Manipal University in 2007. In 2010, he started his career as Assistant Professor in School Of Information Sciences, Manipal. Currently, he is doing Ph.D at Manipal University. His research interest includes High Performance Computer Arithmetic, Advanced Computer Architecture, Low-power VLSI Design, Electronic Design Automation, and Parallel Algorithms/Architectures.</p>
	<p>Dr. C Sundaresan completed Bachelor degree in Electronics and Communication in 2000 from Madurai Kamaraj University and MS degree in VLSI CAD in 2003 from Manipal University and PhD in 2018 from Manipal Academy of Higher Education. He started his career as R & D engineer at Aplab Ltd. Currently he is working as Assistant Professor in School Of Information Sciences. His research interests includes Computer Arithmetic, Low-Power VLSI Design, Logic Synthesis, Static Timing Analysis.</p>

	<p>Dr. P. R. Venkateswaran obtained his bachelor's degree in Electronics and Instrumentation Engineering from National Engineering College, Kovilpatti in 1998 and Masters in Instrumentation and Control Engineering from Technical Teachers' Training Institute, Chandigarh in 2002. He completed his doctoral research in 2008 from Manipal University, Manipal. He started his career as teaching faculty at Sethu Institute of Technology, Madurai and continued his teaching career with Technical Teachers' Training Institute, Chandigarh and later at Manipal Institute of Technology, Manipal. Presently, he is working as Senior Engineer (Control and Instrumentation) at Welding Research Institute, BHEL, Tiruchirappalli and is associated in the areas of Welding Automation and Welding Power Sources. His areas of interest are linear Control theory, Electronic Instrumentation and Soft Computing Techniques. He has been a reviewer for journals like IEEE SMC, Elsevier, AMSE etc. He is a member of professional bodies of ISTE, IWS and IE.</p>
	<p>Dr. Keerthana Prasad is working as Professor in School of Information Sciences, a constituent institution of Manipal University. Her research interests are image analysis and its applications in medicine and high performance computing approach for image processing.</p>